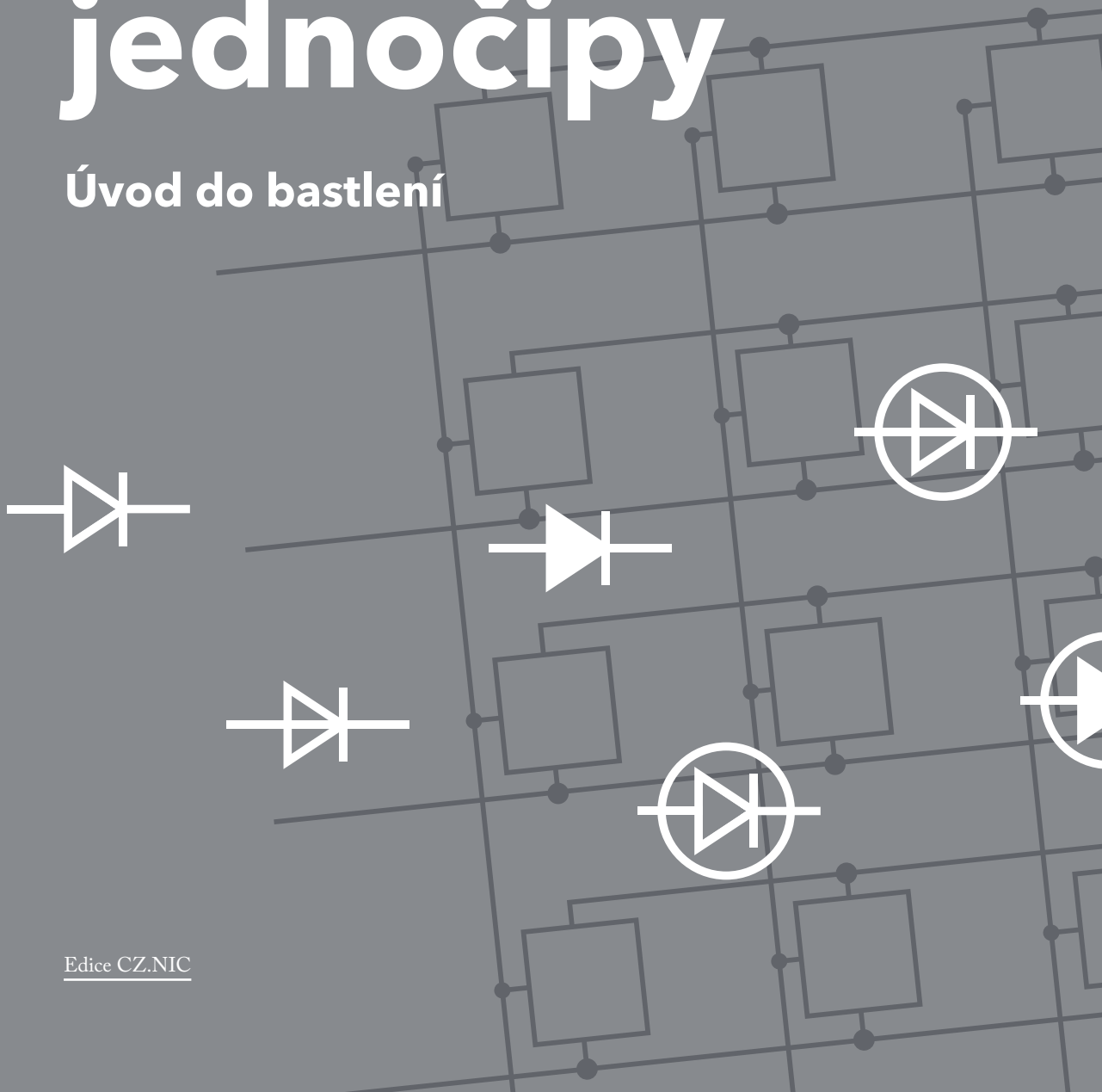
Martin Malý

Hradla, volty, jednočipy

Úvod do bastlení



Edice CZ.NIC

HRADLA, VOLTY, JEDNOČIPY
Úvod do bastlení

Martin Malý

Vydavatel:
CZ.NIC, z. s. p. o.
Milešovská 5, 130 00 Praha 3
Edice CZ.NIC
www.nic.cz

1. vydání, Praha 2017
Kniha vyšla jako 16. publikace v Edici CZ.NIC.
ISBN 978-80-88168-26-3

© 2017 Martin Malý
Toto autorské dílo podléhá licenci Creative Commons
(http://creativecommons.org/licenses/by-nd/3.0/cz/),
a to za předpokladu, že zůstane zachováno označení autora díla a prvního vydavatele díla, sdružení CZ.NIC, z. s. p. o. Dílo může být překládáno a následně šířeno v písemné či elektronické formě na území kteréhokoliv státu.

CZ.NIC | SPRÁVCE DOMÉNY CZ

ISBN 978-80-88168-26-3

— Martin Malý

Hradla, volty, jednočipy

Úvod do bastlení

— Edice CZ.NIC

Poděkování

Poděkování

Děkuji všem, kdo se na vzniku této knihy podíleli, ať už přímo, nebo nepřímo.

Děkuji lidem z Edice CZ.NIC, jejíž laskavou péčí kniha vyšla.

Děkuji Petrovi Koubskému, který mě před mnoha lety pošťouchl k tomu, že vůbec píšu.

Děkuji všem „betatesterům“, kteří četli rukopis a poslali mi k němu svoje připomínky, jmenovitě pánové Michal Valášek, Peter Kmet a Martin Javorek.

Děkuji Štěpánovi Bechynskému, Vladimíru Šuvarinovi, Jakubu Goldmannovi a dalším lidem, kteří mě při psaní podpořili radou, připomínkou a vlídným slovem.

Děkuji všem lidem, které na serveru Zonky.cz zaujal tenhle nápad a umožnili mi téměř půlroční tvůrčí prázdniny!

Děkuji svojí přítelkyni. Nebýt její tolerance, pochopení a neustálého povzbuzování, tak bych nic nenapsal!

Rád bych tuto knihu dedikoval člověku, který pro mne v mládí představoval opravdového hrdinu číslicového věku, člověku, který se zapsal do dějin československé výpočetní techniky jako konstruktér několika počítačů a jehož konstrukce a články mi byly vzorem a první učebnicí počítačové techniky – panu Eduardu Smutnému. Děkuju

Předmluva vydavatele

Předmluva vydavatele

Bývaly časy, kdy se pravý muž poznal podle skvrn od oleje na oblečení a schopnosti kdykoliv rozmontovat a často i zpět smontovat automobil. Tyhle časy minuly, vytlačila je doba řízeného vstřikování a dílů, které se mění kus za kus. A nejenom v automobilu.

A tak na tištěném spoji vidíte jeden spálený váleček a říkáte si, že by přeci mohlo stačit jej nahradit jiným podobným válečkem. Nebo, možná i jen přešlemovat pozici drátkem jako to dělával děda v pojistkové skříni, když něco v domácí elektrotechnice nefungovalo.

Záhy ověříte empiricky, nebo možná vezmete za dané moje tvrzení, že doba přeci jen už o něco pokročila a ne na vše stačí kus drátu. Přesně v rozhraní mezi dobou pokročilou a dobou minulou leží tato kniha. Z minulosti si bere tvůrčí zápal. Schopnost nerezignovat nad úkolem překračujícím pípnutí bezkontaktní platební karty v platebním terminálu. Pokus pochopit klubka drátů. A propojuje ji s dobou moderní, s mikroprocesory, řádky softwarového kódu, Vašimi nápady a samozřejmě se Sítí sítí, Internetem.

Touto knihou Vás autor vezme na procházku do světa nápadů, které střídá realizace. Přestanete být pouhým konzumentem, stanete se autorem, spolutvůrcem, nebo prostě jen napříště budete schopni opravit ten váleček, co se mu mimochodem říká odpor, i když se správně jmenuje rezistor.

Hezké snění a tvůrčí zápal přeje

Patrick Zandl, CZ.NIC
Praha, 20. listopadu 2017

Předmluva mentora

Předmluva mentora

Bývaly časy, kdy nejsnazší způsob, jak přijít k osobnímu počítači, byl postavit si ho. Bývaly časy, kdy u nás bylo víc bastlířů než hráčů počítačových her; napřed jste museli dobastlit, abyste měli na čem hrát.

Bastlit, pokud to slovo už nebo ještě neznáte, znamená: vymýšlet, stavět, drátovat a letovat, zapojovat a rozpojovat, měřit, přemýšlet a drbat se na hlavě a nadávat a nespat, kreslit a škrtat. Užitečný stručný výraz pro velmi komplexní činnost.

Samosebou to byly pitomé časy. Nedostávalo se v nich nejen osobních počítačů, ale taky svobody slova, toaletního papíru a mnoha dalších užitečných komodit. Udržet si čisté svědomí a čistý zadek vyžadovalo opatrnost a vynalézavost. Jen hlupák by chtěl, aby se ta doba vrátila. (Že je u nás takových hlupáků hodně, to by patřilo do předmluvy k nějaké úplně jiné knížce.)

Kdo ale bastlířskou érou prošel, získal důležitý poznatek: elektronika je jeden ze způsobů, jak se naučit myslet.

Co tím chci říci? Existují různé způsoby, jak pochopit logiku světa a odnést si to pochopení z jednoho oboru do druhého. Nechtějte, abych podrobně popsal, co to je logika světa, protože to nedovedu; určitě do ní patří základy skutečné formální logiky, ale také pochopení věcí jako zákony zachování (neboli: vždy je něco za něco), vězňovo dilema (neboli: z některých pastí se dostaneš jen tak, že je redefinuješ), zlaté pravidlo morálky (neboli: všechno se ti vrátí). Nikde se to nenaučíte jako celek, ale jsou obory, jejichž osvojení vám velmi pomůže, naučí vás logice, pečlivosti i pokoře. Patří k nim – podle mne – třeba matematika, jakýkoli cizí jazyk, výchova dětí, bojové sporty, horolezectví, fyzika a biologie, téměř jakákoli řemeslo, které se dělá rukama a vzniká při ní uspořádanější soustava, než jaká byla předtím. Nepatří k nim – opět podle mne – ekonomie, filozofie, míčové hry, marketing, literatura a kupodivu také většina programování. Tím ty druhé obory nesnižuji, jen upozorňuji na rozdíl.

Elektrotechnika, elektronika a číslicová technika v sobě šťastným způsobem spojuje přírodní vědy, matematiku, řemeslo a pokoru. K osvojení logiky světa je proto obzvlášť vhodná. Měli byste si ji vyzkoušet, i když ji k ničemu potřebovat a používat nebudete. Měli byste si ji vyzkoušet, protože je to pro vás dobré a zdravé. A když říkám vyzkoušet, míním tím: rukama. Nestačí číst, je třeba vzít skutečné součástky a začít skutečně stavět a měřit, dělat chyby a odhalovat je.

Chtěl jsem dodat, že je to dnes nesrovnatelně lehčí než v Éře Chybějícího Toaletního Papíru, ale to by nebylo přesné. Nesrovnatelně lehčí je začít, dojít na startovní čáru. Obchody jsou plné součástek a předpřipravených stavebnic. Existují skvělé věcičky jako Arduino a Raspberry Pi a Lego Robotics. Existují zázračné vynálezy jako nepájivé kontaktní pole, což je pomůcka, za kterou bych řekněme v roce 1980 dal téměř cokoli.

Ale tím to končí. Kirchhoffovy zákony jsou pořád stejné, právě tak Booleovu algebru neokecáte tehdy, dnes ani za tisíc let. Musíte projít stejnou cestou, kterou prošli všichni ostatní.

Je to skvělá cesta.

Knížka, kterou držíte v ruce, je turistickým (jen místy lezeckým) průvodcem po této cestě. Po dlouhé době někdo sepsal původní českou knihu o elektronice a číslicové technice, knihu dokonale uzpůsobenou současným podmínkám a možnostem, knihu informačně bohatou a přitom vtipnou a laskavou. Mám radost z každého dobrého výkladu jakékoli problematiky – radost vzácnou, protože moc často si ji člověk neužije. Kniha Martina Malého ji nabízí v míře vrchovaté. Jestli o elektronice nic nevíte, nezmeškejte tuhle výjimečnou příležitost.

A jestli o elektronice víte něco nebo i hodně, pak zas nezmeškejte příležitost podívat se na dobře známé věci jinýma očima – a možná se i seznámit s tím, co je nového, s čím vším si můžete hrát.

Protože elektronika je především ohromná zábava, nekonečná stavebnice, obrovský prostor na hraní a vymýšlení. Život je krátký a všichni si musíme vybírat, co si do něj pustíme a co ne. Já se velmi přimlouvám za elektroniku. Zkuste to.

Petr Koubský
9. listopadu 2017

Výmluvy místo předmluvy

Výmluvy místo předmluvy

Znáte Archimédův zákon?

„No jistě," řeknete, „to zná přeci každý, každý ví, že těleso ponořené do kapaliny je nadnášeno silou, která se rovná váze kapaliny tělesem vytlačené!" To pokud jste chodili do školy dřív.

Pokud jste mladší, znáte možná variantu „Těleso ponořené v kapalině je nadlehčováno hydrostatickou vztlakovou silou, jejíž velikost se rovná tíze kapaliny stejného objemu, jako je objem ponořené části."

Určitě si vzpomenete i na to, jak jste se ho učili: jako básničku. K tomu jste viděli pár malůvek na tabuli – obdélník je těleso, síla F sem, objem V tam, spočítat, vynásobit – bude to plavat? A pane Lorenc, uvezu to?

Kolik z vás si vybaví, že fyzikář přinesl do místnosti lavór s vodou a ukazoval, jak plave plechový hrníček a jak plave sklo? A jak se koule potopí a dutá ne, protože každá váží jinak? Kolik z vás přišlo domů a řeklo si: Aha, proto poloprázdná láhev od limonády plave ve vodě!

Většina dětí umí Archimédův zákon odříkat. Chápou i ty síly a objemy a vzorečky, protože se je naučí. Mají ale problém ho použít.

Nebojte, nejsme v tom sami. O tomtéž psal už Feynman ve svých slavných knížkách. V Brazílii se učila fyzika tak, že studenti dokázali odříkat, co je to Brewsterův úhel, ale nedokázali poučku použít na to, aby vyřešili prostý problém: kde najít v přírodě zdroj polarizovaného světla. I když se oknem dívali na to, jak se slunce odráží ve vodách zálivu.

Nebo to s těmi čtverci nad odvěsnami – já si jako dítě nedokázal představit, jak sčítám čtverce. Ale znám to v téhle podobě, snad jako všichni. Pak jsem se naučil, že jde o druhé mocniny. Věřte nebo ne, to grafické znázornění, s těmi čtverečky, jsem viděl až dlouho po škole, až na internetu.

Nebo Ohmův zákon. Všichni se ho učili, všichni věděli, že „ho musíte znát, i kdyby o půlnoci…" a tak dále. Ale k čemu vám kdy byl, ptám se? V té podobě, v jaké se učí, tedy že U rovná se I krát R, ho moc neužijete. Musíte si ho zažít.

Dnes se všude mluví o internetu věcí a lidé, chytří a vzdělaní, se pouští do elektronických projektů a vlastních konstrukcí, protože cítí buď příležitost, nebo jen obyčejné okouzlení a radost z toho, když něco vlastními rukami uděláte. A spousta z nich chodí za mnou a ptá se: „Jak to vlastně funguje?" Já jim vždycky říkám, že to je úplně jednoduché, když si to sami vyzkouší a nebojí se zeptat. Ukážu jim, jak jednoduché je zapojit LED, aby svítila, a pak se zeptám: „A co kdybych tu LED zapojil obráceně?" V tu chvíli se na mne podívají a řeknou:

„Nevím. Co by se stalo?“

Čekají, že jim to řeknu, ale já jim to nechci říkat. Já chci, aby to sami zjistili, a pak se mě zeptali: Proč to tak je? A proč LED vedle sebe svítí jinak než když jsou za sebou? Teprve teď, když se ptají „proč“, jim začnu vysvětlovat teorii, protože už narazili na otázku, na kterou jim teorie dá odpověď. Začít nejdřív teorií, to je jako začínat odpovědí…

Já věřím, že elektronika je nejen zábavná a tajemná a plná skvělých možností, ale taky snadná na pochopení, pokud se vykládá trochu jinak než od teorie ke vzorečkům. Věřím, že ji dokážu vysvětlit i vám, a že vám ji vysvětlím tak, že se v ní neztratíte. Vím, že můj postup nesedne každému, že někdo chce spíš ta skripta a sednout si na ně a naučit se to po svém a pochopit to vlastním způsobem. Pro takové lidi moje knížka není moc vhodná. A není ani vhodná pro malé děti, od toho jsou knihy jiné.

Věřím, že základy číslicové techniky a číslicové obvody jsou natolik zajímavá oblast, že se vyplatí investovat několik stokorun do součástek a začít experimentovat. Tato kniha vám může být decentním průvodcem, ale nepovede vás za ruku. Není to ani učebnice elektrotechniky se spoustou teorie. Snažil jsem se projít cestou od svícení LEDkou až někam k mikroprocesorům a návrhu počítačů. Cestou úmyslně opomíjím spoustu zajímavých oblastí, jako je audiotechnika nebo vysokofrekvenční technika, a z teorie vysvětluju pouhé nezbytné minimum. Nejde mi o to udělat z vás, čtenářů, elektroinženýry, ale povzbudit vás k vlastním pokusům, vzbudit zájem, nasměrovat… Pokud vás číslicová technika zaujme natolik, že budete chtít vědět víc a tato kniha vám už bude malá, budu jen rád.

Myslím si, že úplně nejlepší způsob, jak se něco učit, je zkusit si to. Vlastní zkušenost je k nezaplacení. Ale vím, že součástky něco stojí, navíc nepřijdou hned, tak jsem využil volně dostupného emulátoru elektronických obvodů, v němž si můžete zkusit probírané věci zapojit, nebo i vytvořit vlastní konstrukci. Nemusíte si kvůli tomu nic instalovat do počítače, stačí vám jen moderní internetový prohlížeč.

Ke knize jsem připravil i materiály pro další studium, dokumenty ke stažení a zdrojové kódy příkladů. Rozhodl jsem se nezaplácávat drahý papír v knize sáhodlouhými výpisy programů, které vy budete dlouze přepisovat. Místo toho naleznete vše potřebné, včetně výše zmíněného emulátoru, na https://elektrokniha.cz – a navíc s opravenými chybami.

—

Kdysi mi jeden známý říkal: „Já dokážu zapojit cokoli podle schématu, ale vůbec nevím, jak na to autor přišel, že to má být takhle! Co potřebuju vědět k tomu, abych to taky dokázal vymyslet?“

V téhle knížce si to povíme. Projdeme si spolu světem elektroniky a číslicové techniky. Nebude to úplně tak, jak to učí ve škole, a spoustu teorie vynecháme. Asi před vámi otevřu víc otázek, než položím odpovědí. Chci, abyste na konci mohli říct nejen „dokážu něco vymyslet a postavit“, ale hlavně „zajímá mě to“ a „mám chuť se do toho pustit!“

Máte chuť se do toho pustit?

Obsah

Poděkování **9**

Předmluva vydavatele **13**

Předmluva mentora **17**

Výmluvy místo předmluvy **21**

1 Budu velkým elektronikem a budu stavět hrozně cool obvody! **37**
1.1 Blikač 37
1.2 „Dílna" 38
1.3 Kde nakoupit součástky? 42
1.4 Nákupní seznam: Součástky pro blikač 44
1.4.1 LED 44
1.4.2 Rezistory 44
1.4.3 Kondenzátory 45
1.4.4 Integrované obvody 46

2 Postavte si blikač – teď už to snad půjde lépe **49**
2.1 Který rezistor je ten pravý? 52
2.2 Měření multimetrem 52
2.3 LED podrobněji 55

3 Hlava, koleno, zem… **61**
3.1 „Nemá to něco společného s atomy?" 62
3.2 Napětí 63
3.3 Proud 65
3.4 Vodič a nevodič 66
3.5 Odpor 66
3.6 Měření, měření! 67
3.7 Ohmův zákon 68
3.8 Výkon 69
3.9 … a malé opáčko 71
3.9.1 Násobky a podíly 71
3.10 Zkratky u značení 72
3.11 Vyvolená čísla 72
3.12 Pro lepší představu 74
3.13 Střídavý proud 75
3.14 Zkrat 76
3.15 Multimetr jako zkrat? 77
3.16 Elektromagnetická indukce 78

3.17 Značky pro schémata 79
3.17.1 Kroužek, nebo ne? 84

4 Zdroje napětí 87
4.1 Společná zem 89

5 Vedle sebe, za sebou 93
5.1 Svítilna s LEDkou 93
5.2 Sériové zapojení 94
5.3 Dělič napětí 95
5.4 Paralelní zapojení 96
5.5 Kirchhof 2 98
5.6 Baterie sériově – paralelně 99
5.7 Potenciometr 99
5.8 Úbytek napětí na LED 101
5.9 Co jsou vlastně ty diody zač? 103
5.10 Datasheet 105

6 Základní elektronické součástky 109
6.1 Rezistor 109
6.2 Kondenzátor 110
6.3 Cívka 112
6.4 Transformátor 113
6.4.1 DC měnič 114
6.4.2 Stabilizátor 114
6.4.3 7805 115

7 Polovodiče 119
7.1 P-N přechod 121
7.2 Dioda 122
7.3 Tranzistor 123
7.4 Rozsvítíme prstem LED! 125
7.4.1 Více světla! 128
7.5 Tranzistor řízený polem (FET) 128
7.6 Šoupejte nožkou… 130
7.7 MOSFET 131
7.7.1 Co je to CMOS? 133
7.8 A to je všechno s polovodiči? 133

8 Pouzdra elektronických součástek 137
8.1 Co je to SMT a THT 138

8.2 DIP, DIL 138
8.3 Co s těmi ostatními? 139
8.3.1 Praktické tipy 140

9 Blikač s Arduinem 143
9.1 Když se řekne Arduino 143
9.2 Programování Arduina 146
9.3 Blikání Arduinem 147
9.4 Krok zpět k drátům 148
9.5 Arduino a EduShield 151

10 Fotorezistor 155
10.1 Obrácená logika 156
10.2 Trimry 157
10.3 Lepší řešení detektoru tmy 158
10.4 Fotorezistor a Arduino 159

11 Termistor 165

12 LM35 169

13 „Jak naučit kámen počítat" 173
13.1 Stavebnice 173
13.2 Logické funkce 174
13.2.1 Digitální, nebo analogové? 174
13.2.2 Dvojková soustava 175
13.2.3 Šestnáctková soustava 176
13.2.4 Zpátky k technice 177
13.3 TTL a CMOS 177
13.3.1 Propojení CMOS a TTL 181
13.4 Operace s bity 181
13.5 Booleova algebra, výroková logika 181
13.6 Logika v číslicové technice 184
13.7 U-káz-ka! U-káz-ka! 186
13.8 Tlačítko a přepínač 188
13.9 Pull Up a Pull Down 190
13.10 Pomalé tlačítko 193
13.11 Schmittův obvod 194
13.12 Blokovací kondenzátor 196
13.13 Buzení z Arduina 196

14 Kombinační logika **201**
14.1 De Morganův zákon 202
14.2 XOR 205
14.3 Logické funkce dvou proměnných 206
14.4 Vícevstupová hradla 208
14.5 Mimochodem, když máme NAND, co ty ostatní? 208
14.6 Zjednodušování logických výrazů 210
14.7 AND-OR-INVERT 210
14.8 Multiplexor 211
14.9 Proč slučujeme přes OR? 213
14.10 Dekodér (demultiplexor) „1-z-N" 213
14.11 Vícebitové varianty 214
14.12 Otevřený kolektor, třetí stav, OE 214
14.13 Dekodéry 217
14.14 Pojďme, budeme už fakt něco počítat! 219
14.15 Aritmeticko-logická jednotka (ALU) 224

15 Sedmisegmentovky LED **229**
15.0.1 Mimochodem… 232
15.1 Víc sedmisegmentovek… 234

16 Jak vypadá hradlo uvnitř **239**
16.1 Proč zapojovat blokovací kondenzátory k napájení 242
16.2 Negované signály 242
16.3 MOS, CMOS 242

17 „Plnou parou vzad!" – „Ale jak daleko?" **247**
17.1 Ještě pípat! 249

18 Zpětná vazba **255**
18.1 Astabi-cože? 256
18.2 Blikač 257
18.3 Krystalový oscilátor DIL 259
18.4 Monostabilní klopný obvod 261
18.5 Detektor pohybu 264
18.6 Bistabilní klopný obvod R-S 266
18.6.1 Klopný obvod s hradly NAND 268
18.7 Zakázané kombinace, zpětná vazba, … 268
18.8 Hodiny 270
18.9 Synchronní / Asynchronní 271
18.10 Symbol pro klopný obvod 272

18.11 Reálný klopný obvod D: 7474 272
18.12 Reálný latch 7475 276

19 Panna, nebo orel? 279
19. 1 Náhoda? Nemyslím si… 280
19.1.1 Jaké hodnoty RC zvolit, když vím jen to, že součin má být XYZ? 281
19.2 Střída 281
19.3 PWM 283
19.4 Dělení kmitočtů 284
19.4.1 Násobení kmitočtu? 285
19.5 Klopný obvod T 286
19.6 Klopný obvod J-K 287

20 Čítače 291
20.1 Čítač s nulováním 292
20.2 Čítače v praxi 294
20.3 Hrací kostka 297
20.3.1 Reálná kostka 299
20.3.2 Vyjádření logických výrazů 301
20.3.3 Montážní OR 302
20.3.4 Zobrazovací obvod hrací kostky 303
20.3.5 Dekodér hrací kostky 304
20.4 Další čítače 305
20.5 Ještě nějaké čítače? 307
20.5.1 Johnsonův kód 308
20.5.2 Grayův kód 308
20.6 Rotační enkodér 310
20.7 Čítač s dekodérem 1-z-10 typu 744017 311
20.8 Počítadlo k autodráze 312

21 Posuvné registry 317

22 Paralelní a sériová rozhraní 321
22.1 Buzení displeje ze sedmisegmentovek 324
22.2 Posuvný řadič SIPO 74HCT595 326

23 Sériová komunikace 331
23.1 Sériová sběrnice SPI 331
23.2 Sériová sběrnice I²C 334
23.3 Prakticky… 336
23.4 EduShield a displej 338

23.5 RS-232, UART, Serial… 339
23.6 Převodník USB na sériové rozhraní 342
23.7 1-Wire 343

24 Paměti 347
24.1 7489 – 64 bitů RAM 348
24.2 Dynamická RAM 350
24.3 ROM, PROM a další 351
24.4 To nejlepší z obou světů 353
24.5 Několik tipů k pamětem 355
24.6 Jak se zapisuje do EEPROM či FLASH? 355
24.7 Sériové paměti 357

25 Sériová paměť prakticky 363

26 Hodiny reálného času 369

27 Paměťové karty 373

28 Logický analyzátor, logická sonda 377

29 Elektronika a svět kolem nás 381
29.1 Ovládáme přírodu elektronikou 381
29.1.1 Elektromagnety 381
29.1.2 Motory 382
29.1.3 Relé 385
29.1.4 Darlington, FET, Tyristor 386
29.1.5 Servo 387
29.1.6 Krokový motor 388
29.1.7 Světlo 388
29.1.8 Peltierův článek, topná spirála 389
29.1.9 Reproduktor 390
29.2 Příroda ovládá elektroniku 391
29.2.1 Tlačítka a klávesy 391
29.2.2 Dotyk 391
29.2.3 Světlo – fotorezistor, fotodioda, line tracking 392
29.2.4 Magnetismus 393
29.2.5 Otáčení, posun 393
29.2.6 Poloha, zrychlení 394
29.2.7 Zvuk 394
29.2.8 Teplota, vlhkost 395

29.2.9 Vzdálenost 395
29.2.10 Tlak 396
29.2.11 Plyn 396
29.2.12 GPS 396

30 Meteostanice 401
30.1 Výběr součástek 401
30.2 Špinavej trik 401
30.3 Stavíme z polotovarů 402

31 Bezdrátový přenos dat 409
31.1 Vysílání na 433 MHz 410
31.2 nRF24L01+ 411

32 Procesory, počítače, mikrořadiče 417
32.1 Mikroprocesor 8080A 418
32.1.1 Ready / Wait 420
32.1.2 Hold (DMA) 420
32.2 Přerušení 420
32.2.1 Nemaskovatelné přerušení 421
32.3 Periferie 421
32.4 Složitější periferie 422
32.5 Jednočipový mikropočítač 424
31.5.1 Harvard vs von Neumann 425
32.6 Atmel AVR 425
32.6.1 RISC 425
32.6.2 Vnitřní uspořádání ATmega328 426
32.7 Další mikrokontroléry 427
32.7.1 ARM 428
32.7.2 PIC 428
32.7.3 8051/8052 428
32.8 Tak málo nožiček… 428
32.9 Programování jednočipů 429

33 Displeje 433
33.1 Znakový displej 1602, 2004 433
33.2 Grafický displej 12864 435
33.3 Další displeje 436
33.4 Bezdrátový displej k naší meteostanici 436

34 Klávesnice **441**
34.1 Šetříme vývody 443
34.2 Připojujeme klávesnici od PC 444
34.3 Matice tlačítek 445
34.4 Postavte si třeba… kalkulačku? 446

35 Osm tlačítek na třech vodičích **453**
35.1 Multiplexior / Demultiplexor 453
35.2 PISO a SPI 453
35.3 Analogová cesta 455
35.4 R-2R 457

36 Joystick **461**

37 ESP8266 WiFi **465**
37.1 Moduly ESP8266 465
37.2 Převodník napěťových úrovní 466
37.3 WeMos D1 Mini, NodeMCU 468
37.4 Bezdrátový teploměr s WiFi 469
37.5 Instalace podpory ESP8266 do Arduino IDE 469
37.6 WiFi Manager 471
37.7 Klient / server? 472

38 Low Power **475**
38.1 Solární články 477

39 Sigfox **481**
39.1 Co je to Sigfox? 481
39.2 Cloudový teploměr se Sigfoxem 482
39.3 Co s daty v Sigfoxu? 485

40 Šťastnou cestu… **489**

Přílohy **493**
Nástroje a weby 493
Nákupní seznam začínajícího hobby elektronika 494
EduShield 497
Nahrání firmware do EduShieldu 499
Turris Omnia pro experimenty s elektronikou 502
Karnaughova mapa 504
„Dobré rady nad zlato" na jednom místě 508

1 Budu velkým elektronikem a budu stavět hrozně cool obvody!

1 Budu velkým elektronikem a budu stavět hrozně cool obvody!

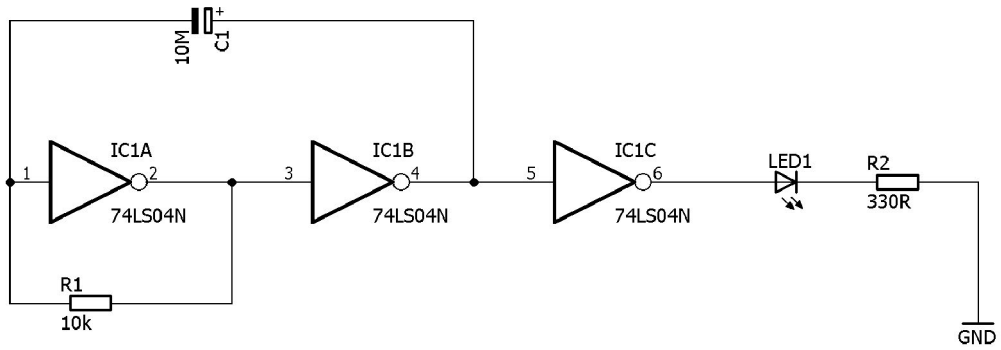
Naučit číslicovou techniku, říkáte? A základy elektroniky, říkáte? Však to není žádná složitá věc, to zvládneme. A ať se zbytečně nezdržujeme, tak rovnou začneme blikačem, ne?

Tuhle kapitolu můžete klidně vzít zlehka, letem světem, a později se k ní vrátit. Pokud nechápete, o čem je v ní řeč, nevadí. Zkuste si ji i tak alespoň přečíst, jak se říká, nasucho… Nebojte, hned za chvíli bude všechno jasnější!

Jako programátoři začínají známým „Hello world“, tak se v elektronice používá blikání světlem. To je takový základ, který by každý měl znát, a kterým se vždycky začíná. Pojďme si to postavit, je to snadné.

1.1 Blikač

Blikač můžeme postavit mnoha způsoby a použít k tomu nejrůznější součástky. Dřív se používaly tranzistory, my se k nim taky dostaneme, ale pro začátek zvolíme složitější součástku, totiž integrovaný obvod. Konkrétně to bude obvod 7404 – vyrábí ho celá řada výrobců v nejrůznějších obměnách, my použijeme libovolný z nich – tedy 74LS04, 7404, 74HC04. Obvod se skládá z šesti invertorů, využijeme z nich polovinu, tedy tři. K tomu připojíme rezistor, elektrolytický kondenzátor, a jako indikaci použijeme LED s rezistorem v sérii. Takto:



⌀ https://eknh.cz/ledblik

Zapojení je velmi jednoduché, s danými hodnotami kmitá zhruba na frekvenci 4,5 Hz, takže záblesky jsou dobře vidět prostým okem. Postavte si ho na nepájivém kontaktním poli…

Nebo něco není jasné?

Cože? Že **nic není jasné**? Že nevíte, co je integrovaný obvod, co znamenají ty značky, co je to to nepájivé kontaktní pole, co je rezistor, a vlastně z toho vysvětlení nechápete vůbec nic?

Tak to jste tu správně!

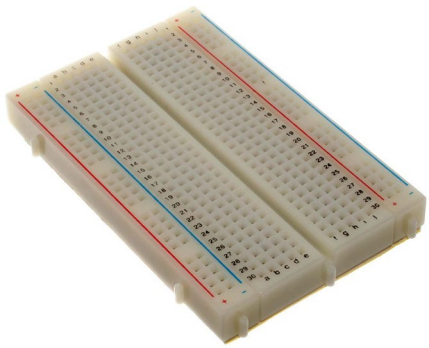
Začneme od počátku. Nebojte, ten blikač si postavíme za chviličku, jen si nejdřív řekneme, co je co a jak na to.

1.2 „Dílna"

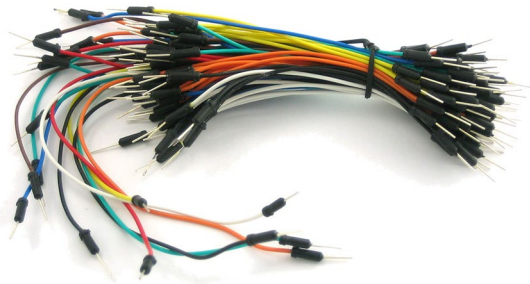
Být programátorem je velmi snadné. Stačí vám k tomu počítač a židle. Ani ten stůl není potřeba, zvládnete to totiž na koleni. Doslova. S elektronikou je to trochu těžší – minimálně ten stůl potřebujete, protože pracujete s malými součástkami a nechcete, aby vám padaly na koberec.

Neříkám, že potřebujete hned dílnu s ponkem a spoustou přístrojů, ale něco přeci jen mít musíte. V téhle knize vám budu ukazovat základy elektroniky, a budu se snažit, abyste se při tom neměli šanci ani zranit, ani zničit vybavení bytu. Bude vám stačit jen místo na stole, dostupná zásuvka, pořádné světlo a pár nástrojů a pomůcek. Něco z toho možná doma máte, pokud ne, tak to pořídíte v libovolném obchodě, ať už kamenném, nebo internetovém. Pár tipů najdete v další kapitole.

Co tedy nakoupit? V první řadě **nepájivé kontaktní pole**. Anglicky se tomu říká *breadboard*, a je to deska s maticí otvorů, které jsou vždy po pěti propojené. Do těchto otvorů zasunujete vývody součástek a spojujete je pomocí drátků, pokud nevyjdou vývody do správné pětice. Díky tomu nemusíte pracovat s páječkou, a zároveň je zapojený obvod snadno rozebiratelný, takže můžete zapojení sestavovat a opět rozmontovávat bez rizika, že součástky třeba nadměrnou tepelnou zátěží poškodíte.



K nepájivému kontaktnímu poli je vhodná i sada **propojovacích vodičů**. V zásadě můžete použít jakýkoli izolovaný drát s tloušťkou kolem 0,3 mm, ale doporučím koupit speciální vodiče pro nepájivá kontaktní pole. Tyto vodiče mají na koncích plastové koncovky, díky kterým se snáze zastrkávají i vytahují. V ČR jsou z nějakého důvodu nehorázně drahé, ale pokud je koupíte přes internet z e-shopů v zahraničí, můžete se dostat na zlomek ceny. Hledejte *jumper wire* nebo *dupont wire*.



Když už budete kupovat propojovací vodiče, kupte si nejen „male-male“, tedy „samec-samec“, ale i „male-female“ a „female-female“. Tedy vodiče, které mají na jednom či obou koncích dutinky. Budou se hodit pro připojování modulů k Arduinu.

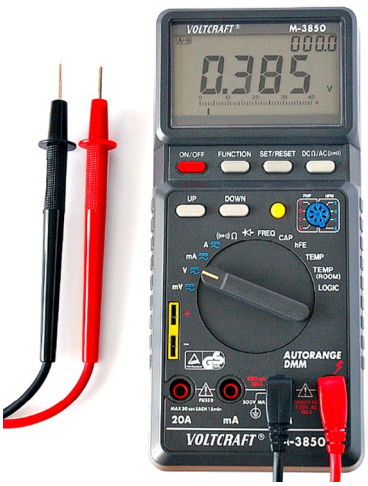


Budete potřebovat občas **malý šroubovák**, plochý i křížový – asi všichni máme nějaký takový doma, ale zde zdůrazňuju to „malý“. Nebo si rovnou vezměte celou sadu. Hodí se vám i **plochá pinzeta**. Nekupujte si nejlevnější z obyčejného plechu, u nich se snadno stane, že se ohnou a nedrží a jsou pak na vyhození. A pokud nemáte oči jako ostříž, oceníte **lupu**.



Další nezbytná věc jsou **malé kleště**, ideálně dvoje – jedny štípačky, jedny „obyčejné“. Po čase přijdete na to, že se vám budou hodit i kleště pro odizolování kabelů. Teď to ještě nezbytné není.

Co nezbytné je, to je **multimetr**. Kdysi se prodával krásný měřicí přístroj, jmenoval se Avomet a byl nehorázně drahý. Dnes si můžete koupit taky drahé meřicí přístroje, ale ze začátku vám naprosto bez problémů postačí digitální multimetr, který najdete v každém hobbymarketu. Multimetr je zařízení, které vám umožňuje měřit napětí, proud, odpor, a někdy i další veličiny. Multimetr je nezbytný, protože, jak se říká, bez měření není vědění! Sice existují způsoby, jak si podobná měřidla vyrobit či nahradit, ale do začátku je lepší si jedno prostě koupit.

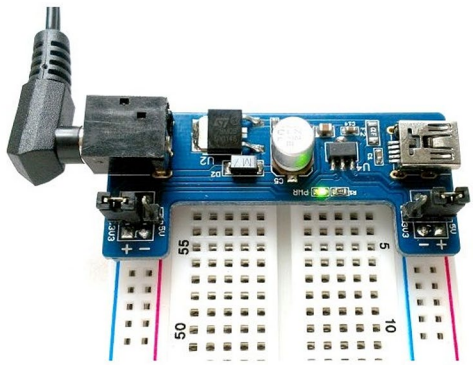


CC-BY-SA, autor André Karwath

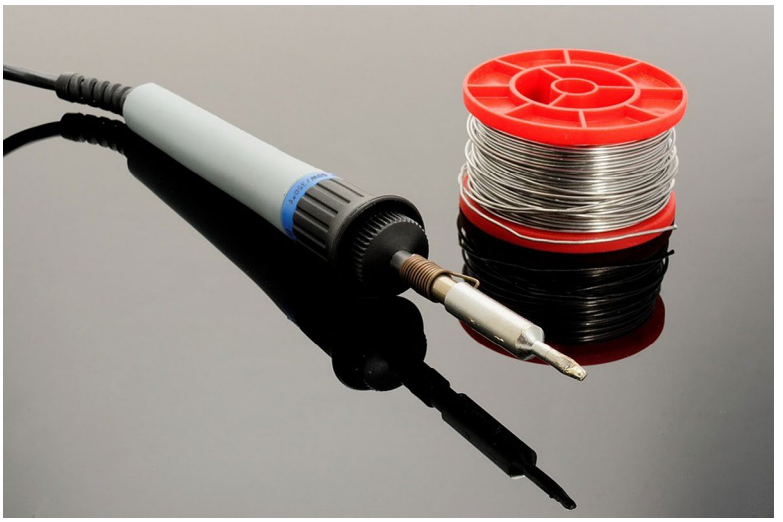
Když už budete v tom hobbymarketu, kupte si i takové **plastové krabičky**, které se prodávají na různé šroubky a drobné věci, nebo kufřík s přihrádkami. Do nich si můžete dávat součástky, které se vám nebudou válet všude možně, nebudou se vám ztrácet a nešlápnete na ně.



Aby bylo vůbec co měřit, potřebujete **zdroj**. My naštěstí budeme pracovat s elektronikou, která si vystačí s nízkým napětím pěti voltů. Je několik možných způsobů, jak toto napětí získat. Můžete si koupit laboratorní zdroj, ale ten stojí nemalé peníze. Můžete použít plochou baterii nebo tři monočlánky – ty dohromady dají jen 4,5 V, ale to většinou stačí. Na druhou stranu jsou tyto baterie velmi nepraktické, protože se po čase vybijí a můžete je zahodit. Nejvhodnější způsob, který vám doporučím, je použít nabíječku, která má konektor USB. USB totiž používá právě těch 5 voltů, takže nemusíme řešit stabilizaci napětí, zároveň dává i dostatečně velký proud, s nímž můžeme použít i některé náročné komponenty, jako displeje nebo bezdrátové komunikační moduly. Já používám přesně takový adaptér, a k němu speciální nástavec do nepájivého kontaktního pole. Další možnost, trochu nouzová, je použít Arduino nebo Raspberry a brát si napětí z něj.



Všimněte si, že jsem nezmínil **páječku**. Zatím ji nepotřebujete, základní zapojení si postavíte i bez ní. Až jednou přijde doba, kdy se bez ní neobejdete, tak vám doporučím – nekupujte si pistolovou trafopáječku, kupte si takovou, kde je stojánek a regulace teploty. Nemusí to stát majlant, nejlevnější stačí. A když už jsme u toho: ten přístroj je *páječka*, ačkoli tomu kdekdo říká *pájka*. *Pájka* je ve skutečnosti ta kovová slitina, kterou se letuje, čili pájí, a té se lidově říká *cín*, i když cín je jen jedna ze složek…



To je opravdu všechno. S tímhle vybavením se můžete pustit do objevování tajů elektřiny, elektroniky, mikroelektroniky, číslicové techniky – jak chcete… Teď už stačí jen nakoupit součástky.

1.3 Kde nakoupit součástky?

Tohle je docela podstatná kapitola. Kde nakoupit součástky a nástroje?

V České republice existují kamenné obchody se součástkami. Nejsou zdaleka v každém městě, ale naštěstí mívají i e-shop, takže si vybrané součástky můžete nechat poslat poštou. Velká výhoda těchto obchodů je, že vám součástky dojdou poměrně rychle – třeba do týdne. Další výhoda je při nákupu dražšího vybavení (pro tuto knihu dražší vybavení není potřeba, ale třeba jednou…) – máte ze zákona záruku na zakoupené zboží.

V ČR asi nejlépe fungují obchody GM Elektro, GES, TME a Tipa. Tyto obchody udržují široký sortiment, od objímek a drátů přes součástky až po nástroje, přístroje a další vybavení dílny.

Některé součástky lze nakoupit i ze zahraničních e-shopů, jako je Mouser a Digikey, ale tady je potřeba počítat s vyšším poštovným, které může být klidně i 1 500 Kč.

Specializované internetové obchody (Sparkfun, Adafruit, Seeedstudio) nabízejí spoustu velmi zajímavých součástek, senzorů a dalšího vybavení, ale problémem může být vyšší cena, vyšší poštovné, a občas i zdržení a další poplatky na celnici (clo + DPH).

Poslední možnost, kde nakoupit součástky ve velkém množství a levně, jsou obří internetové servery ebay.com a aliexpress.com, oba plné čínských prodejců součástek a různých klonů. Zde je potřeba určité obezřetnosti – někdy můžete koupit funkčně ekvivalentní věc, a jindy získáte klon, který bude špatně dílensky provedený, bude se přehřívat, nebude fungovat spolehlivě… Předem nelze říct. Na druhou stranu většina těchto obchodů má poštovné zdarma, a za cenu, za jakou koupíte v českém kamenném obchodě tři LED, máte z těchto obchodů celý sáček se stovkou kusů (*100 pcs*).

Další nevýhodou těchto obchodů je delší dodací lhůta – čekáte dva až šest týdnů. Já sám jsem si zde nakoupil většinu základních součástek, jako jsou rezistory, kondenzátory, LED, propojovací vodiče, nepájivá kontaktní pole, tlačítka… Upřímně – i kdybych měl z balíčku stovky tlačítek deset procent kazových, tak je s klidným svědomím vyhodím, protože celý balíček i s poštovným stál 25 Kč. Což je zhruba cena tří kusů týchž tlačítek v kamenném obchodě.

Ovšem samozřejmě můžete zapomenout na záruku. Vadné zboží zde těžko vyreklamujete (anebo se vám to při jeho ceně nevyplatí). U menších součástek však nebývá problém napsat obchodníkovi, ten nabídne raději refundaci peněz nebo zaslání nové sady, než aby přišel o zákazníka a dostal negativní hodnocení.

Na druhou stranu zažíváte až absurdní pocity, když si zde koupíte za cca dvacetikorunu svazek 75 propojovacích vodičů pro nepájivé kontaktní pole, a pak vidíte, že za tentýž svazek (i fotka odpovídá) chce nejmenovaný český internetový obchod okolo sto dvaceti korun. Podobné situace jsou s různými „sety“ a „začátečnickými kity“, kde je většinou taková všehochuť součástek a komponent – rezistory a kondenzátory různých hodnot, několik různobarevných LED atd. V českých obchodech není problém za takovou sadu dát i několik tisíc korun, z Číny vás to bude stát třeba okolo dvou set Kč. Ale zase platí: počkáte si delší čas. Pokud chcete koupit něco rychle, připlaťte si a objednejte v ČR.

A pokud vám aspoň trochu leží na srdci osud lidí, kteří se tu nějak snaží povzbuzovat kutilskou tvorbu, tak aspoň občas nakupte u nich (HW Kitchen, Hobbyrobot apod.) Ceny jsou vyšší, ale pokud tyto obchody nepřežijí, bude to škoda.

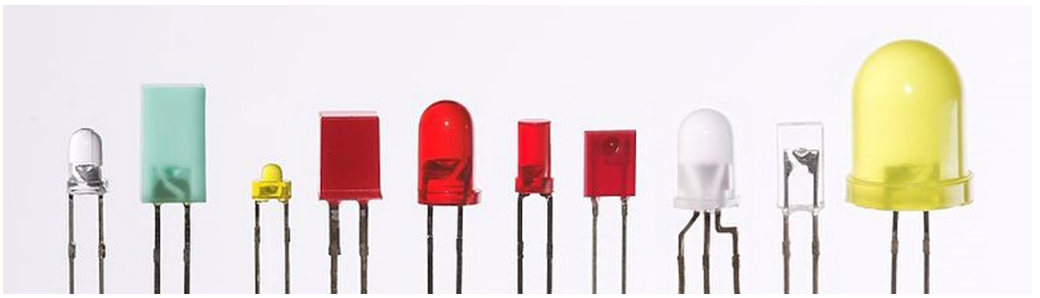
1.4 Nákupní seznam: Součástky pro blikač

Bez součástek se neobejdete. Bez součástek si ani nebliknete, ani nezapípáte, bez součástek neuděláte nic. Nebudete potřebovat žádné složité ani drahé ani příliš náročné. Do začátku vám jich stačí jen několik základních, které vás nepřijdou na víc než na několik desítek korun.

1.4.1 LED

Především jsou to **LED**. LED je akronym anglických slov Light Emitting Diode, tedy „světlo vyzařující dioda". Neříkejte proto prosím „LED dioda", protože říkáte vlastně „Světlo vyzařující dioda dioda". Někteří obhajují slovní spojení „LED dioda„ kvůli tomu, že slovo „dioda" můžete skloňovat: LED diodou, LED diody, … *Já s dovolením sklouznu k nespisovnému, ale používanému výrazu LEDka.*

Ale zpět k věci: LED jsou v nejrůznějších barvách, nejčastěji červená, žlutá, zelená, modrá a bílá. Jsou i vícebarevné diody, dvoubarevné, trojbarevné (RGB – červená, zelená, modrá) i čtyřbarevné (RGBW – jako předchozí, ale navíc s bílou). Jsou diody se zabudovaným řídicím obvodem, který umožňuje sofistikované řízení, např. v různých LED páskách. Ale ty potřebovat nebudete. Kupte si do začátku úplně obyčejné jednobarevné LED, třeba pět kusů od každé barvy. Jestli zvolíte průměr 5 mm, nebo 3 mm, to je jedno. Ty větší líp „sednou do ruky" při manipulaci, ty menší zase snáze dáte vedle sebe. Já používám pětimilimetrovou verzi…



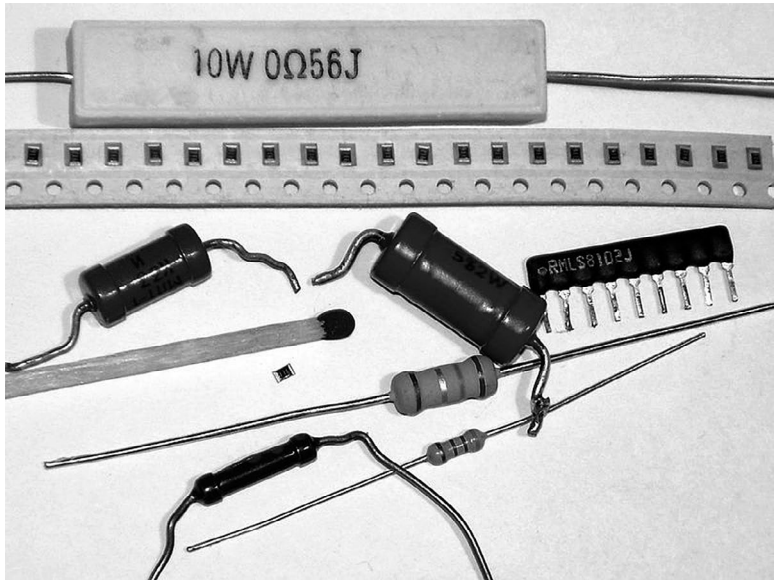
CC-BY-SA, autor Afrank99

1.4.2 Rezistory

Další součástky, bez nichž se neobejdete, jsou **rezistory**. Dříve se jim říkalo také „odpory". Jenže odpor je označení fyzikální veličiny (jednotka je ohm, značíme Ω), a pak to špatně vypadá, když říkáte „odpor s odporem 100 ohmů". Asi jako „dvoumetrový metr". Proto se používá označení „rezistor".

Rezistory se třídí do různých skupin – podle výkonu, podle přesnosti, podle výrobní technologie... Pro naše potřeby postačí ty nejmenší, nejobyčejnější a nejlevnější, takže něco jako „uhlíkový rezistor 0,25 W“ nebo „metalizovaný rezistor 0,25 W“. V kusovém množství to jsou korunové položky. Existuje ale možnost nakoupit celé sady rezistorů, třeba 100–200 kusů s různými hodnotami odporů. Při nákupu z Číny vás celá taková sada vyjde třeba na dvacet korun (ovšem dodací lhůta je nepoměrně delší).

Pořiďte si pro první pokusy rezistory o odporu 330 ohmů (označují se 330R), 1000 ohmů (1K) a 10 000 ohmů (10K). Kdo chce, pořídí si i 220R, 2K2, 4K7, 22K, ať můžeme experimentovat. Od každé velikosti tak deset kusů, ale víc se rozhodně neztratí. (O značení a jednotkách si za chvíli povíme víc.)



1.4.3 Kondenzátory

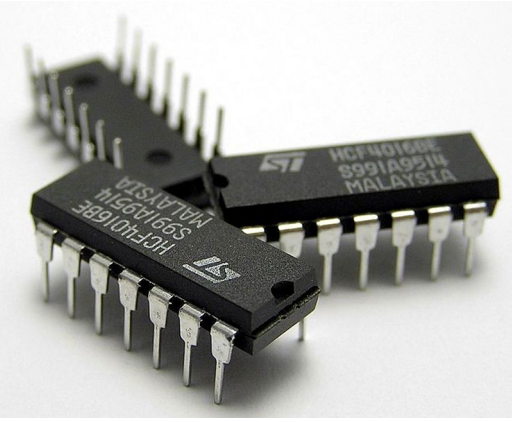
Kromě rezistorů budete potřebovat i **kondenzátory**. Základní vlastností kondenzátorů je jejich kapacita, která se měří ve faradech (značíme F). Ještě začátkem 90. let platilo, že nejčastější kapacity se udávají v pikofaradech, nanofaradech a mikrofaradech (bilióntina, miliardtina, milióntina), ovšem dnes jsou dostupné kondenzátory s kapacitou stovek faradů. V číslicové technice takové velké kapacity můžeme použít k napájení obvodů při výpadku, ale vy je teď potřebovat nebudete. Vystačíte si pro začátek s kapacitami 33 pikofaradů (pF), 100 nanofaradů (nF) a 10 mikrofaradů (μF; u tohoto se nenechte vylekat tím, že bude označený jako „elektrolytický“). Setkáte se taky

se starším označením 33, 100K a 10M. Opět vezměte ty nejlevnější, na experimenty budou stačit. Kapacity 33 pF a 100 nF vypadají jako malý polštářek, 10 µf vypadá jako váleček.



1.4.4 Integrované obvody

A když už jsme v tom, tak si kupte i dva integrované obvody. Jeden má označení 74HCT00, druhý 74HCT04. Pokud nebude k dispozici HCT, kupte klidně LS nebo ALS (74LS00, 74ALS04, …) Pozor! Vybírejte takové, které jsou v pouzdře DIL (někdy značeno jako DIP). Takové můžete zastrčit do nepájivého kontaktního pole. Vyhněte se pouzdrům SO, ty jsou určené pro povrchovou montáž pájením.



Později budete potřebovat i další součástky – tranzistory, diody, integrované obvody. Jejich seznam najdete na konci knihy, abyste nemuseli kupovat každou součástku zvlášť. Teď si vystačíme s tím, co je výše napsané. Vážně!

Pojďme si tedy postavit ten blikač...

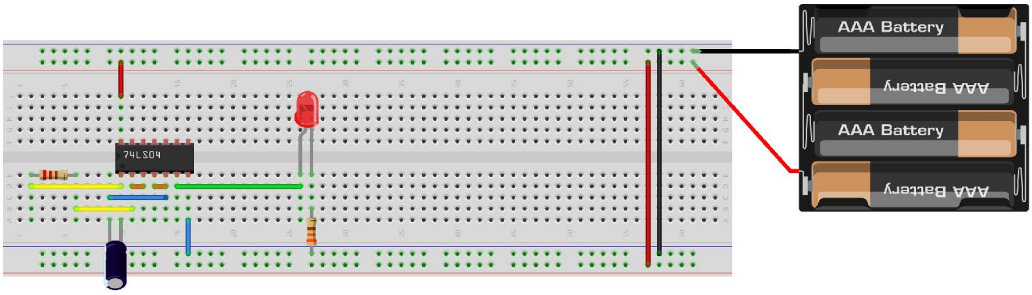
2 Postavte si blikač – teď už to snad půjde lépe

2 Postavte si blikač - teď už to snad půjde lépe

Budete potřebovat:

- Integrovaný obvod 7404 (v té variantě, jakou máte, tedy i 74LS04, 74ALS04, 74HCT04)
- Elektrolytický kondenzátor 10 µF
- Rezistor 330 Ω
- Rezistor 10 kΩ
- LED libovolné barvy
- Nepájivé kontaktní pole
- Napájecí obvod
- Propojovací kabely (Poslední tři položky budete potřebovat vždy, tak už je nebudu opakovat)

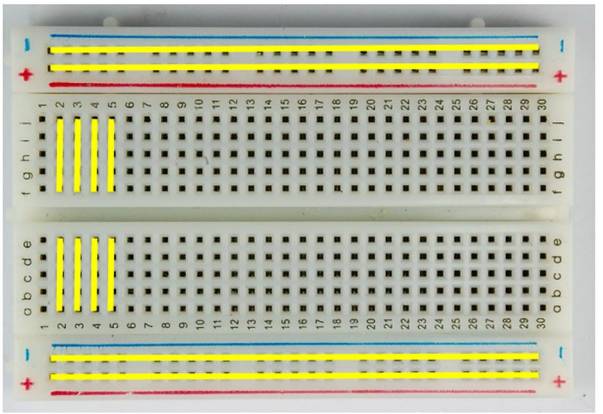
Bez dalšího vysvětlování si pojďte poskládat obvod na nepájivém kontaktním poli. Výsledek bude vypadat nějak takto:



Nejprve se podívejte na nepájivé kontaktní pole. Všimněte si, jak je toto pole uspořádané. Nahoře a dole (z tohoto pohledu) jsou dvě oddělené lišty, označené červeným a modrým proužkem. To jsou napájecí lišty, které slouží k rozvádění napájecího napětí do obvodu. Jsou uspořádané do pětic a jsou spolu uvnitř pole vodivě propojeny horizontálně, jak je naznačeno na následujícím obrázku.

Střední část pole je uspořádána jinak – zde je horní a dolní polovina, a v nich jsou vždy jednotlivé pětice otvorů propojeny svisle (a-b-c-d-e, f-g-h-i-j). Je to výhodné uspořádání pro součástky,

které mají dvě řady vývodů, jako náš integrovaný obvod 7404. Takový obvod přesně sedne doprostřed, a každý jeho vývod je připojen ke čtyřem dalším otvorům, takže máte dostatečný prostor na propojování.



Než vytáhnete integrovaný obvod z obalu, tak se nejprve uzemněte. Sáhněte třeba na radiátor. Statická elektřina je prevít a některé citlivé obvody můžete zničit tím, že na ně sáhnete. Pokud máte 74ALS04 nebo 74LS04, tak ty tak citlivé nejsou, ale některé řady (74HCT04) a bez výjimky všechny složitější obvody jsou na statickou elektřinu velmi citlivé. Proto se většinou prodávají zapíchané v proužku vodivé pěny, která udržuje všechny vývody propojené, a tudíž se nestane, že by na jednom vývodu bylo velmi vysoké napětí oproti ostatním.

Jste uzemněni? Tak se podívejte na integrovaný obvod. Jsou na něm nejčastěji různé značky výrobce (ST, TI, F), pak čísla výrobních sérií, země původu, a někde mezi tím bude skryto i to označení 7404. U mne to je konkrétně T74LS04B1. „T“ označuje výrobce, 74LS04 je typ obvodu a technologie, a B1 je upřesňující označení – teplotní rozsah, pouzdro atd.

Vývody integrovaného obvodu jsou očíslované od 1, jenže na pouzdru tato čísla nikde nejsou. Proto si pamatujte základní poučky, jak zjistit orientaci:

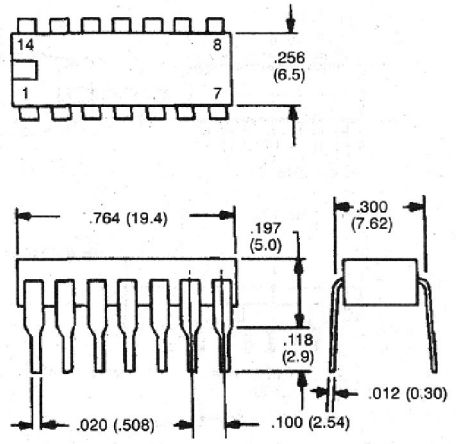
- Když si dáte obvod tak, aby označení bylo čitelné, tak většinou (ne vždy!) bývá vývod číslo 1 vlevo dole. Ale toto pravidlo není stoprocentně spolehlivé.

- Podívejte se na pouzdro. Na jedné z kratších stran bude značka – vlys, výbrus, malý žlábek, … Když držíte pouzdro tak, aby tato značka byla vlevo, tak je vývod číslo 1 zase vlevo dole.

- Někdy výrobce i na tuto značku kašle, a místo toho označí nějak vývod 1. Buď barevnou tečkou, nebo třeba miniaturním důlkem.

- Když všechno selže, podívejte se do datasheetu. Později si vysvětlíme, co to datasheet je.

Když víte, jak je obvod orientovaný, tak ho zasuňte do pole tak, jak je nakresleno na úvodním obrázku. Nepůjde to, vývody trochu nepasují – je to proto, že vývody jsou z výroby lehce „rozkročené". Proto je decentním tlakem z obou stran o trochu narovnejte, aby byly rovnoběžné. Já k tomu používám plochu stolu. Stačí opravdu maličko. Pak obvod půjde bez problémů zasunout do nepájivého kontaktního pole. Zasuňte ho tak, aby vývod 1 byl vlevo dole, třeba v otvoru „10e". Je to jedno, ale aspoň se v tom lépe vyznáte.



Máte-li vývod 1 vlevo dole, tak číslování pokračuje proti směru hodinových ručiček – napravo od 1 je 2, pak 3, 4, až k vývodu číslo 7. Naproti němu je vývod číslo 8 (vpravo nahoře), a číslování pokračuje až k vývodu 14 (vlevo nahoře). Obvod 7404 totiž používá čtrnáctivývodové pouzdro (značí se DIL14 nebo DIP14). Některé obvody mají pouzdra větší – 16 vývodů, 24 vývodů, 28, 40, někdy i víc. Ale obvykle platí, že od určitého počtu vývodů se obvody primárně dělají v pouzdrech pro povrchovou montáž (SMD – Surface Mount Device), které mívají přes sto miniaturních vývodů. Ty ale my používat nebudeme.

Máte zasunutý obvod, je to tak? Nejprve k němu připojíme napájení. Vývod 7 připojte k modré (záporné) napájecí liště, vývod 14 k červené (kladné). Proč zrovna tyto dva a takto, to se později dozvíte, teď prosím jen následujte můj popis.

Pomocí propojovacích vodičů spojte vývody 2 a 3. Dalším vodičem spojte vývody 4 a 5.

Zapojte rezistor 10 kΩ mezi vývody 1 a 2.

Aha? A který rezistor to je, že?

2.1 Který rezistor je ten pravý?

Na většině součástek je napsán alespoň typ, nebo základní hodnota. Rezistory, které používáme, ale vypadají jako takové miniaturní válečky s dvěma vývody. Bohužel, pokud máte miniaturní rezistory 0,25 W, tak na nich není moc prostoru k nějakému rozepisování. Místo toho na nich jsou barevné proužky, které udávají odpor a přesnost (toleranci). Pro odpor 10K to jsou hnědá, modrá a oranžová, takže – lupu do ruky, pořádné světlo, a hledat! Já nevím jak ostatní, ale já často na rezistor koukám, a přemýšlím, jestli to je hnědý proužek nebo červený, jestli ten vedle je modrý nebo černý, pak přemýšlím, jestli ho nedržím obráceně, a nakonec sáhnu pro multimetr a prostě jej změřím. Udělejte to taky tak.

Přemýšlel jsem, jestli sem tu tabulku dávat, a nakonec jsem ji nezařadil. Najdete ji kdekoli na internetu, stačí hledat „barevné značení rezistorů", nebo v jakékoli jiné knížce o elektronice. Já si říkám, že se bez přesné znalosti v číslicové technice docela dobře obejdete. Stačí vám vědět, že něco takového je, a kde to popřípadě najdete.

2.2 Měření multimetrem



CC-BY-SA, autor André Karwath

K multimetru se dodávají dvě sondy, nejčastěji červená a černá. Černou zasuňte do zdířky, která bývá označena COM, popřípadě symbolem pro uzemnění (na obrázku úplně dole). Červenou

zapojte do zdířky, která mívá označení VΩmA (na obrázku prostřední). Třetí zdířka je určená pro měření velkých napětí a proudů a je označená např. 10ADC, 10A MAX a podobně. Tu nebudete používat.

Multimetr má většinou uprostřed jeden otočný volič, kterým přepínáte měřenou veličinu (napětí, proud, odpor) a rozsah. U odporů (bývá značen velkým písmenem omega) to bývá 200, 2K, 20K, 200K a 2M. Začneme tím nejmenším, to je 200. Přepněte volič do pozice 200, a na displeji by se měla ukázat jednička úplně vlevo – to znamená „větší odpor, než je rozsah".

Teď zkuste na chviličku spojit oba hroty sond. Uvidíte, že číslo na displeji rychle klesne až skoro k nule. Právě jsme vytvořili zkrat – a ten má skoro nulový odpor (vlastně jen odpor samotných vodičů, a ten je minimální). Některé multimetry v takové chvíli taky pískají, to abyste věděli i bez sledování displeje, že jsou oba hroty vodivě spojeny.

Pro pořádek: symbolem „V" a vodorovnou čárou volíte měření napětí ve voltech, případně milivoltech



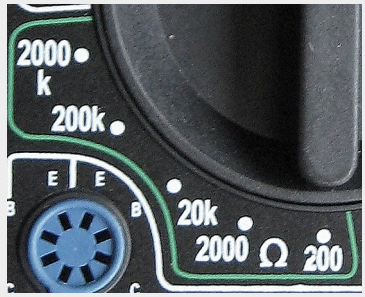
Symbolem V s vlnovkou vybíráte rozsah měření pro střídavé napětí



Symbol A s vodorovnou čárkou znamená měření stejnosměrného proudu (v mikro - a miliampérech).



Symbol „omega" (Ω) označuje rozsah pro měření odporů v ohmech a kiloohmech:



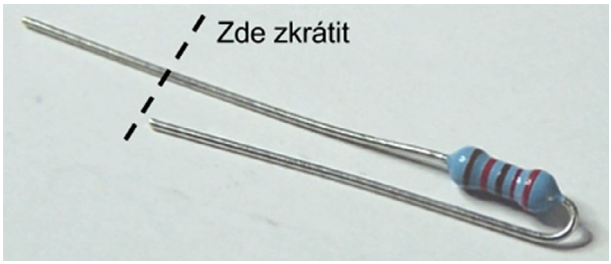
Jestli nevíte, jaký je přesně rozdíl mezi napětím a proudem, tak to teď nevadí, včas se to dozvíte. Teď jen přepněte multimetr na „200 Ω".

Pojďme měřit. Jeden vývod rezistoru připojte na černý hrot, druhý vývod na červený, a sledujte displej. Pravděpodobně tam bude stále „1" – tedy „odpor je větší než rozsah". Přepněte tedy rozsah na větší – 2K. Pokud se na displeji objeví třeba „330", tak to znamená, že měříte rezistor s odporem 330 ohmů. To není ten, co teď potřebujete (budete ho potřebovat až za chvilku). Přepněte na 20K – a pokud máte správný, tak se na displeji objeví hodnota okolo 10 – bývá to třeba 9,7, nebo 10,2 – to je stále v toleranci.

Pro zajímavost: Zkuste si teď rezistor otočit, prohodit červený a modrý hrot, a změřte si odpor tentokrát. Jaký je? Je stejný. Rezistor je součástka symetrická, a pokud jej otočíme, tak se nic nestane.

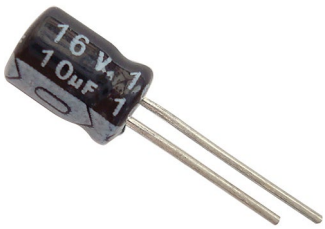
Zpátky k blikači. Zapojte rezistor 10K mezi vývody 1 a 2 integrovaného obvodu (IO). Na obrázku je rezistor nakreslený vlevo od IO a je připojen pomocí dvou vodičů. Je to proto, že vývody 1 a 2 jsou

hned vedle sebe, a rezistor se mezi ně nevejde. Další možnost je zapojit rezistor našikmo, jedním vývodem do otvoru „d“, druhým do otvoru „a“, no a poslední možnost je zapojit jej „nastojato“ – jeden vývod zkrátíte, na druhém utvoříte ohyb o 180 stupňů, a tím se oba vývody dostanou vedle sebe a můžete je zapojit do sousedních otvorů.



Jsme v polovině! Jupí...

Teď zapojte kondenzátor 10 µF mezi vývody 1 a 4. Na kondenzátory multimetr většinou nemá měřicí rozsah, ale naštěstí jsou kondenzátory větší a vejde se na ně popis. Ten váš hledaný bude vypadat jako černý nebo modrý váleček s dvěma vývody na jednom konci. U jednoho vývodu je po celé délce pouzdra bílý proužek se stylizovaným znakem „minus“. Ten vývod bývá také kratší (ale nespoléhejte na to, třeba ho někdo ucvaknul – spolehlivý je ten proužek na pouzdru). Elektrolytický kondenzátor totiž má „kladný“ a „záporný“ vývod. Zapojte jej kladným na vývod 1 integrovaného obvodu, záporným na vývod 4.



Kdybyste teď připojili napájecí napětí, tak na vývodu 6 získáte pravidelné pulsy. Jenže je neuvidíte, neuslyšíte, neucítíte… Takže vám nezbývá, než mi to věřit – nebo si to ověřit!

2.3 LED podrobněji

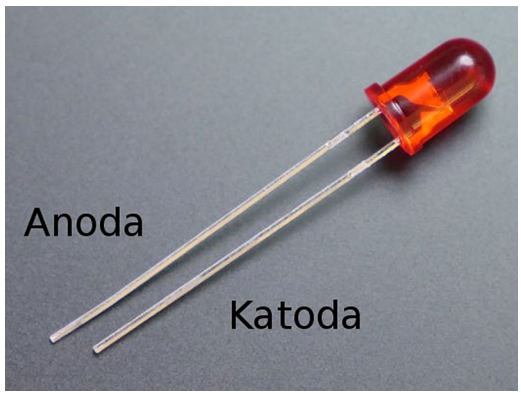
Nejjednodušší způsob, jak to udělat, je použít LED. LED je součástka, která za určitých podmínek mění procházející elektrický proud na světlo. Nejčastější typ LED vypadá jako takový malý klobouček s dvěma vývody.



Všimněte si několika věcí:

- Jeden vývod je delší (pokud si ho nezkrátíte…).
- Uvnitř diody je větší část s takovou miniaturní mističkou (už jsem říkal, že se vám bude hodit lupa?) a menší část. Větší část má kratší vývod, menší část delší (asi aby to bylo spravedlivé).
- Na spodním okraji kloboučku je zvýrazněná hrana, a ta je v jednom místě seříznutá. Je to u té větší části s kratším vývodem.

Dioda je součástka, která má dva vývody, a protože se bavíme jako elektronici, tak si budeme pamatovat, že vývody jsou připojené k elektrodám, které se jmenují *katoda* (což je záporná elektroda) a *anoda* (to je kladná elektroda – mnemotechnická pomůcka: *kladná* elektroda tam má to *ano*, a ti, co umí rusky, vědí, že *da* znamená *ano*). Seříznutá hrana označuje *katodu*. Katoda je ten větší útvar uvnitř diody. Ten druhý vývod, bez seříznuté hrany a s menším útvarem, to je kladná elektroda, anoda. (*Další mnemotechnická pomůcka: delší vývod si můžete zkrátit a ten ucvaknutý kousek přeložit napříč, abyste získali „+“*)



Zapojte anodu na výstup číslo 6 integrovaného obvodu. Asi není dobrý nápad ji strkat přímo na místo. Dejte si ji kousek stranou, do jiné řady vývodů, a anodu propojte s vývodem 6 nějakým vodičem.

Předposlední krok: Katodu připojte přes rezistor s odporem 330 ohmů (rozpoznávání hodnot viz předchozí text) na zápornou napájecí lištu. Pamatujte si: **LED vždy s rezistorem!** Jinak ji spálíte.

Teď připojte zdroj k nepájivému kontaktnímu poli. Modrou – zápornou – napájecí lištu na záporný pól (budeme mu říkat zem, proč, to si řekneme později), červenou na kladný (+ 5 V). Můžete si zdroj ještě proměřit na multimetru: nastavte rozsah 20 V, černou sondu připojte na zem (tedy na zápornou, modrou lištu), červenou na + 5 V, a pokud se na displeji ukáže něco okolo 5, je to v pořádku. Pokud se ukáže záporné číslo, máte prohozené plus a mínus.

Nezapomeňte propojit horní napájecí lištu se spodní dvěma vodiči tak, jak je naznačeno na obrázku. Lišty spolu nejsou vodivě spojené, o to se musíte postarat sami.

Máte? Ještě jednou si to prosím překontrolujte podle výše uvedeného postupu a obrázku. Pro jistotu stručné shrnutí:

1. Integrovaný obvod 74LS04 (74ALS04, 74HCT04) zasunutý tak, aby měl vývod číslo 1 vlevo dole – to znamená potisk čitelný, ne vzhůru nohama, a na levém okraji značku.

2. Vývod 14 (vlevo nahoře) připojený na +, vývod 7 (vpravo dole) připojený na -.

3. Spojené vývody 2 a 3 integrovaného obvodu (IO)

4. Spojené vývody 4 a 5 IO.

5. Mezi vývody 1 a 2 IO zapojený rezistor 10 K (před zapojením změřte!).

6. Mezi vývody 1 a 4 IO zapojený kondenzátor 10 µF. Vývod označený bílým proužkem na vývod 4 IO, druhý na vývod 1 IO.

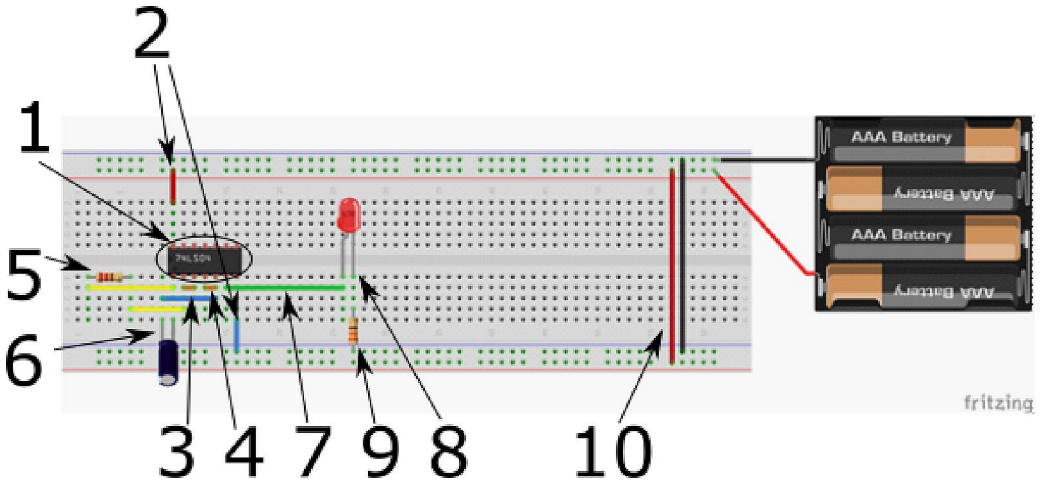
7. Vývod 6 IO připojený na anodu LED (anoda = menší útvar uvnitř, delší vývod, hladký lem).

8. Katoda LED (= větší útvar, kratší vývod, seříznutý lem) připojená na jeden vývod rezistoru 330.

9. Druhý vývod téhož rezistoru připojený na mínus (-).

10. Horní i dolní napájecí lišty propojené – plus na plus, mínus na mínus.

11. Zdroj 5 V zapojený správně, nepřepólovaný.



Zapněte napájení.

Bliká?

Pokud ano, tak hurá, sláva, haleluja, zvládli jste to. Pokud ne, nastává problém, a těch může být asi tak dvacet. Od těch banálních (na něco jste zapomněli, něco nemá správný kontakt, napájení nefunguje) přes závažnější (něco jste zapojili obráceně a je potřeba to otočit – nejčastěji LED, popřípadě jste použili nefunkční součástku) po fatální (něco jste zapojili obráceně, ono to potichu shořelo a přestalo fungovat a fungovat to nezačne, ani když to otočíte zpátky). Projděte si znovu celé zapojení, všechno si překontrolujte, změřte, jestli je na vývodech 7 a 14 napájecí napětí, zkuste otočit LED (prohodit vývody), jestli ji nemáte obráceně...

Pokud jste ale při sestavování postupovali pečlivě a přesně podle návodu, mělo by vše fungovat na první dobrou, takže se můžeme za intenzivního blikání naší první elektronické konstrukce podívat, co že se tam vlastně děje a proč. Protože cílem této knihy není ukázat vám, jak se postaví blikač, ale dát vám dostatek informací k tomu, abyste si ho dokázali navrhnout a postavit sami!

Ono to bohužel není tak úplně jednoduché a k tomu, abyste opravdu pochopili, proč blikač bliká, potřebujete spoustu teorie. Možná vás zklamu, ale nečekejte, že se ono tajemné mystérium blikače dozvíte hned v následující kapitole… Ale slibuju, že se k němu dostaneme – sice až v půlce knihy, ale věřte mi: to už budete vědět, jak funguje třeba kondenzátor a jakou má funkci, takže princip blikače bude naprosto zřejmý.

3 Hlava, koleno, zem...

3 Hlava, koleno, zem...

... zapikanej jsem, říkali jsme jako malí při rozpočítávání. Když jsem přemýšlel nad tím, jak vysvětlit základní pojmy, bez kterých se neobejdeme, totiž pojem napětí a proudu, tak mě napadlo právě toto říkadlo.

Pohodlně se usaďte, na chvíli si odpočineme od sestavování konstrukcí, beztak to byla příliš velká nálož novinek. Teď zapojíme místo diody vlastní hlavu...

Schválně, jak vysoko nad zemí je vaše hlava, když stojíte? U sebe to vím celkem přesně, jsou to dva metry. Koleno 60 centimetrů. Zem je od země přesně nula centimetrů – to jen pro pořádek. Když vezmu vajíčko a upustím ho na koberec – ne, prosím, tento pokus nemusíte nutně dělat, jen si to představte – z výšky svého kolene, tak vydrží. Když ho pustím z výšky svojí hlavy na tentýž koberec, rozkřápne se. Jak to, že totéž vejce puštěné z různých výšek dopadne pokaždé jinak?

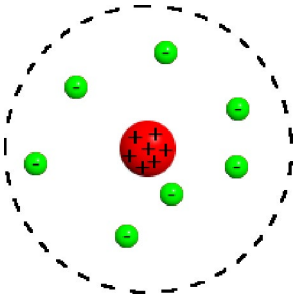
Odpověď je: „To je přeci jasné!“ No dobře, jasné to je, ale nějaké vysvětlení by tu být mohlo. Je zde, a jmenuje se „potenciální energie“ (též „polohová“). V našem případě je to potenciální energie v gravitačním poli Země. Tato energie je tím vyšší, čím výš nad zemí předmět, v našem případě vejce, je (samozřejmě s omezeními, protože v dostatečně velké vzdálenosti už bude gravitační síla, která to vajíčko přitahuje, tak malá, že nebude mít na vejce podstatný vliv). Dokud držím vejce v ruce, má potenciální energii. Jakmile ho upustím, začne se tato potenciální energie přeměňovat v energii kinetickou, pohybovou, a vajíčko bude padat k zemi. Čím níž vejce bude, tím vyšší bude jeho kinetická energie – a podle zákona o zachování energie bude součet kinetické a potenciální energie konstantní. Jakmile vejce doputuje k zemi, bude jeho potenciální energie nulová a kinetická bude rovna té původní potenciální. A tato kinetická energie se v okamžiku dopadu na zem opět přemění v jinou energii a vykoná práci (v našem případě destrukci skořápky).

Výška nad zemí tedy určuje potenciální energii – pozice „hlava“ má vyšší než pozice „koleno“, a pozice „koleno“ má vyšší než pozice „zem“. Pozice „zem“ má potenciální energii nulovou. Když vejce leží na zemi, nemá kam padat a nemá tedy nic, co by se proměnilo v kinetickou energii. Tohle všechno je empiricky pochopitelné, úplně jasné a každý si to dokážeme představit.

Proč v knížce o elektřině vysvětluju padání vajec na zem, navíc tak zdlouhavě? Protože nám to poslouží jako dobrá analogie s elektřinou. Všimněte si, že potenciální energie se neměří „sama od sebe“, vždycky se měří „mezi dvěma body“. Třeba mezi zemí a předmětem nad ní. Stejně tak se neříká, že „Sněžka je sto kilometrů daleko“ – je to úplně prázdné sdělení, pokud neřekneme odkud je to těch sto kilometrů.

3.1 „Nemá to něco společného s atomy?"

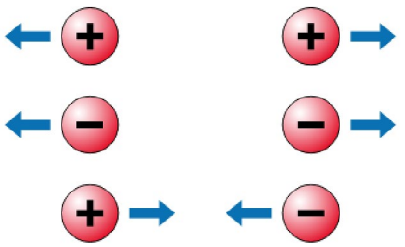
S elektřinou je to podobné, jako s gravitací. Jen místo gravitační síly fungují síly elektrické. Nositelem elektrické energie jsou subatomární částice, elektrony a protony. To si určitě pamatujete ze středoškolské fyziky: každá látka je složena z atomů, a v těchto atomech je jádro, sestávající z protonů a neutronů, a kolem jádra jsou elektrony. Když jsem já chodil do školy, tak se pro zjednodušení používal „planetární model", v němž elektrony vypadaly jako malé kuličky, co létají okolo jádra po různých drahách, podobně jako planety. Dnešní fyzikální výklad je mnohem subtilnější, ale pro pochopení elektrické energie nám stačí ten model planetární.



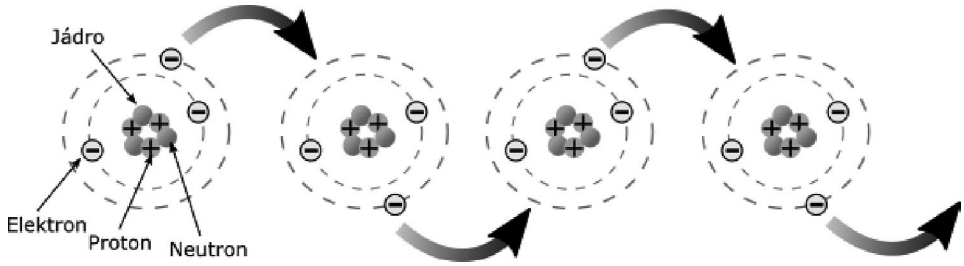
Elektrony i protony nesou malý náboj. Elektrony záporný, protony kladný – a v obou případech je stejně velký. Za ideálního stavu (*ideální stav je ten, který pozorujeme pouze na obrázcích v učebnicích středoškolské fyziky*) je v atomu přesně tolik elektronů, kolik je v něm protonů, jejich elektrické náboje jsou tedy stejně velké, a navenek má tedy atom nulový náboj.

Věci s kladným nábojem a věci se záporným nábojem se navzájem přitahují – podobně jako gravitace způsobuje, že se k sobě přitahují předměty. U gravitace ale nepozorujeme, že by fungovala obráceně. U elektřiny (a magnetismu) tomu tak je: stejně nabité částice se odpuzují.

Jak je tedy možné, že se kladně nabité protony mačkají spolu v jádru, a nerozletí se od sebe? Drží je u sebe jiná fyzikální síla, zvaná silná interakce – tento jev ale působí pouze na velmi malé vzdálenosti, takže s ním nemáme každodenní zkušenost.



Ve skutečnosti se atomy nevyskytují samy. Ve hmotě je jich spousta vedle sebe. Stává se, že elektrony, které jsou od jádra atomu nejdál, se vzdálí od toho svého jádra a bloumají meziatomárním prostorem. V původním atomu se tím poruší rovnováha, atom získá kladný náboj (protože v něm bude víc protonů než elektronů), a bloumající elektron ponese zase záporný náboj. Po čase tento volný elektron najde nějaký atom, kde elektron chybí, a stane se jeho elektronem – a toto se děje v hmotě neustále. Neustále se od atomů uvolňují elektrony a chytají se k jiným atomům. Některé látky, třeba kovy, mají atomy s velmi promiskuitními elektrony, které rády cestují prostorem – v každé chvíli je k dispozici velké množství volných elektronů. Jiné látky, třeba sklo, mají takových volných elektronů málo.



Elektrony přeskakují z jednoho atomu do druhého

3.2 Napětí

Určitě si vzpomínáte na ten školní pokus, kdy fyzikář třel ebonitovou tyč liščím ohonem a tvořil tak statickou elektřinu, kterou pak nechával vybít za působivého jiskření. Vzpomenete si na to vždycky, když se v oblečení z umělých vláken posadíte na křeslo z podobného materiálu, a při vstávání dostanete ránu, až vás zabrní ruka. „Jojo, statická elektřina," řeknete si. Ale jak vlastně vznikla?

Když se o sebe třou dva předměty z nevodivé látky, tak za určitých okolností přejdou volné elektrony z povrchových atomů jednoho předmětu na povrch druhého předmětu. Počet takto přešlých elektronů závisí na spoustě faktorů, od složení látek přes rychlost vzájemného tření až po vzdušnou vlhkost. Předmět, který elektrony získal, má teď elektronů víc než ten, který je ztratil. Ten, který elektrony přijal, má vůči tomu druhému záporný náboj (elektrony nosí záporný náboj), předmět, který elektrony ztratil, má stejně velký kladný náboj.

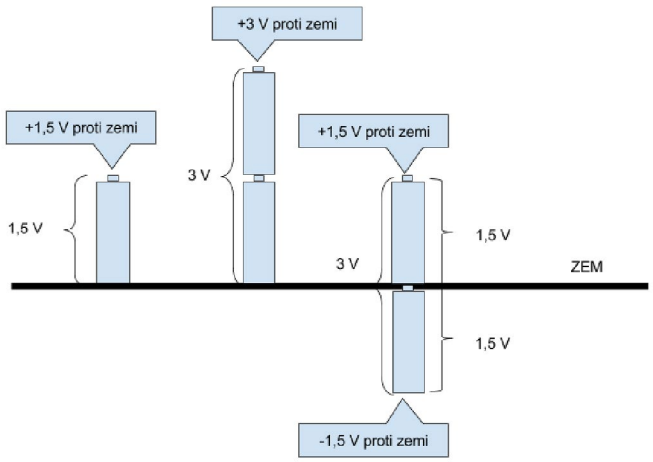
Takovýto elektrický náboj můžeme přirovnat k výše zmíněné potenciální energii. Čím větší je rozdíl v náboji, tím větší „potenciální energie" mezi předměty vznikla. Říkáme, že tyto předměty mají rozdílný elektrický potenciál. Velikost toho rozdílu se označuje elektrické napětí, označuje se U (v anglické literatuře někdy V jako Voltage) a měří se ve voltech (značka V).

Elektrické napětí je tedy rozdíl potenciálů mezi dvěma body. Jeden, ten, kde je elektronů méně, označujeme za kladný, ten, kde je elektronů víc, označujeme za záporný. Když se podíváte na obyčejnou baterii, najdete záporný a kladný pól. Všude v elektronice je záporný a kladný pól, a napětí je udávané mezi těmito dvěma póly. (A platí to i pro střídavé napětí, ačkoli tam se, jak si za chvíli řekneme, oba póly neustále prohazují, střídají.)

Aby se to celé nepletlo, tak se zavedl koncept nulového potenciálu, a dohoda zní, že tímto tělesem je Země. To je naše „vztažná soustava", to je bod 0.

Vrátím se na chvíli ke svému říkadlu a stvořím analogii: Hlava je u mne 200 centimetrů nad zemí, rozdíl „potenciálů" je tedy 200 (centimetrů). Koleno je 60 centimetrů nad zemí, rozdíl potenciálů koleno-zem je tedy 60 cm. Rozdíl hlava-koleno je tedy 140 centimetrů. Pokud budeme brát směr „nad zem" jako kladný, tak koleno má + 60, hlava + 200. Teď si představte, že mám zem měkkou jako těsto a zabořím se do ní až po kolena. Co se stane? Hlava bude mít proti zemi potenciál + 140, kolena budou ve stejné výšce jako Země, takže jejich potenciál bude 0, a paty budou mít potenciál - 60 (budou pod povrchem).

Proč o tom mluvím? Protože stejný princip platí i pro napětí. Představte si, že vezmu obyčejný tužkový monočlánek – typu AA, 1,5 V – a postavím ho na zem záporným pólem (to je ten plochý. Na kladném, to je ten s tím hrbolkem, by nestála, to dá rozum). Tím vznikne spojení mezi zemí a záporným pólem baterie. Kladný pól tedy bude mít potenciál 1,5 voltu proti zemi. Když vezmu druhou baterii stejného typu a postavím ji na tu první, tak potenciál jejího kladného pólu bude + 3 volty proti zemi. Kdybych to celé postavil obráceně, bude mít volný pól potenciál - 3 volty.



⌀ https://eknh.cz/tribat

3.3 Proud

Napětí a proud se lidem často pletou. Je to stejná dvojice, jako třeba Žebrák a Točník, nebo Laurel a Hardy – všichni vědí, že patří k sobě, ale lidé obvykle nevědí, který je který. My si zase pomůžeme analogií – tentokrát ale ne s vajíčkem, ale s nádrží plnou vody, která bude mít u dna výpust. Když takovou nádrž postavíte na zem, tak se nic nestane. Nádrž bude stát, vodní hladina se bude jemně otřásat (protože nad ní proudí vzduch, vedle v místnosti někdo jde, venku jezdí auta, zkrátka máme kolem sebe neustále nějaké zdroje vibrací, i když jsou mizivé), a nic se nebude dít. Analogická situace je u naší baterie: mezi jejími póly je napětí neustále, ale za normálního stavu se nic neděje.

Teď výpust otevřete. Voda začne téct, nejprve rychle a prudce, a jak bude hladina klesat, bude vodní proud slábnout a slábnout, až nakonec ustane. Dalo by se říct, že síla vodního proudu (což není fyzikální veličina, ale pro představu stačí) je závislá na výšce vodní hladiny nad zemí. Čím vyšší vodní hladina, tím silnější proud.

Stejný jev nastane, když vodivě spojíte kladný a záporný pól baterie. Elektrony začnou přecházet od záporného pólu ke kladnému, začnou proudit vodičem, a vzniká elektrický proud. Ten se označuje písmenem I (velké i, nikoli malé L – z francouzského intensité de courant – intenzita proudu) a měří se v ampérech (označení A).

Toto je důležitý rozdíl. Velmi důležitý! Proto to teď několikrát zopakuju v různých formulacích, aby bylo naprosto jasno.

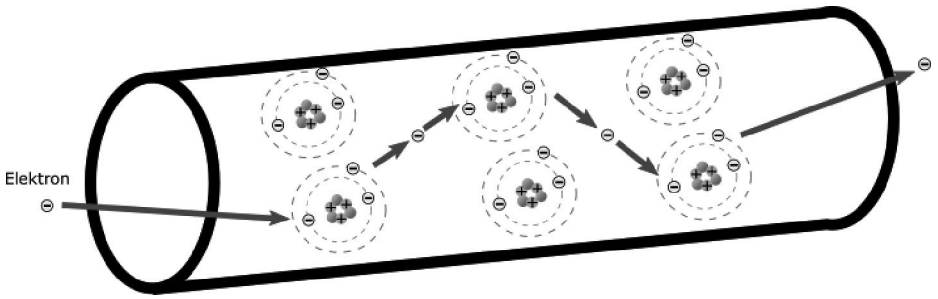
- Napětí je **statické**, proud je **dynamický**.
- Napětí **je**, proud **teče**.
- Proud teče vodičem mezi dvěma body, mezi nimiž je napětí.
- Napětí může existovat samo (viz třeba baterie, kde je napětí), aniž by tekl proud. Proud bez napětí ale nemůže existovat.

Aby to nebylo tak jednoduché, tak je tu jedna nelogická věc. Proud vzniká tehdy, pokud vodičem prochází elektrony. Elektrony se vždy pohybují od záporného pólu ke kladnému, ovšem když se bavíme o směru proudu, je přesně opačný. Říkáme, že **proud teče od kladného pólu k zápornému**. Je to čistě konvence, která bohužel vznikla dřív, než se přišlo na to, co přesně se pohybuje a že to jsou záporně nabité elektrony. Tak si to prosím pamatujte: **proud teče od + k -, ale elektrony se pohybují obráceně**!

3.4 Vodič a nevodič

Z běžného života víte, že některé látky elektřinu vedou – říkáme jim **vodiče**, jiné látky nevedou – těm se říká **nevodiče** nebo **izolanty**. Záleží na tom, kolik má která látka k dispozici volných elektronů. Pokud jich je hodně, jako třeba v kovech, tak takové látky elektřinu vedou, pokud je jich málo (dřevo, plast, sklo, suchý vzduch), tak elektřinu nevedou.

Jak vlastně ta elektřina vodičem putuje? Představme si běžný drát, třeba železný nebo měděný. V něm je velké množství volných elektronů, které jsou rovnoměrně rozptýlené. Teď připojíme vodič jedním koncem na záporný pól baterie. Jak víme, záporný pól má nadbytek elektronů. Ty se budou tlačit do vodiče, a protože se částice se stejným nábojem odpuzují, budou před sebou tlačit volné elektrony, co jsou uvnitř drátu. Na druhém konci vodiče tedy vznikne zase přebytek elektronů, a ten se stane „záporným". Pokud tento konec připojíme ke kladnému pólu a *uzavřeme tak obvod*, přejdou elektrony na kladný pól, kde jich je nedostatek.



Průtok elektronu vodičem. Každý urazí jen krátkou dráhu, ale rychlost šíření informace je obrovská.

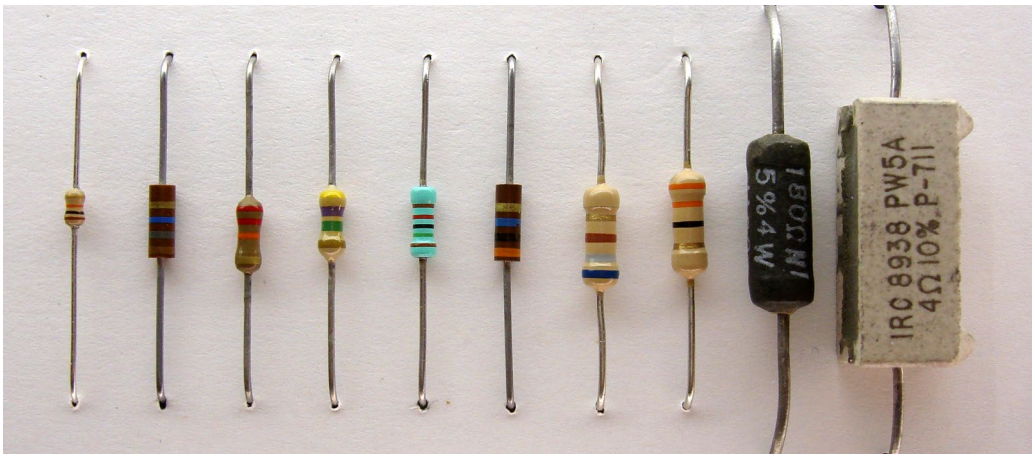
Analogicky si můžeme vodič představit jako vodovodní trubku, která vede z vodárny k vám domů. Je plná vody, a ta v ní stojí. Jakmile otočíte kohoutkem, tak voda začne vytékat. Není to ta, která v tu chvíli teče ve vodárně do trubky, ale ta, kterou před sebou tato čerstvá voda tlačí v potrubí. Stejné to je i s proudem ve vodiči. Elektrony samotné se pohybují velmi rychle, ale ve vodiči urazí jen několik centimetrů za sekundu. Ovšem „tlak" od záporného pólu dorazí na konec vodiče téměř rychlostí světla.

3.5 Odpor

Vraťme se k našemu vědru s vodou, akváriu, nádrži… Síla proudu, který vytéká výpustí, je dána nejen výškou hladiny (tedy v naší analogii napětím), ale i dalším faktorem, totiž velikostí výpusti (respektive jejím průřezem). Čím větší výpust, tím větší proud, a obráceně.

Ve světě elektřiny existuje analogická vlastnost, totiž **vodivost**. Vodivost vlastně udává, kolik volných elektronů má daná látka k dispozici. Vodiče, látky vodivé, mají vodivost velmi vysokou, látky nevodivé velmi nízkou (vakuum téměř nulovou). V elektronice ale používáme jinou veličinu, která je převrácenou hodnotou vodivosti, totiž **odpor**. Vodiče mají odpor velmi malý, nevodiče téměř nekonečný. Odpor měříme v ohmech (čteme [óm] a značíme Ω) Označení pro tuto veličinu je R (od slova „resistance“)

Odpor doplnil základní trojici elektrických veličin. Pardon za dlouhý výklad bez experimentů.



Teď hned jdeme něco změřit!

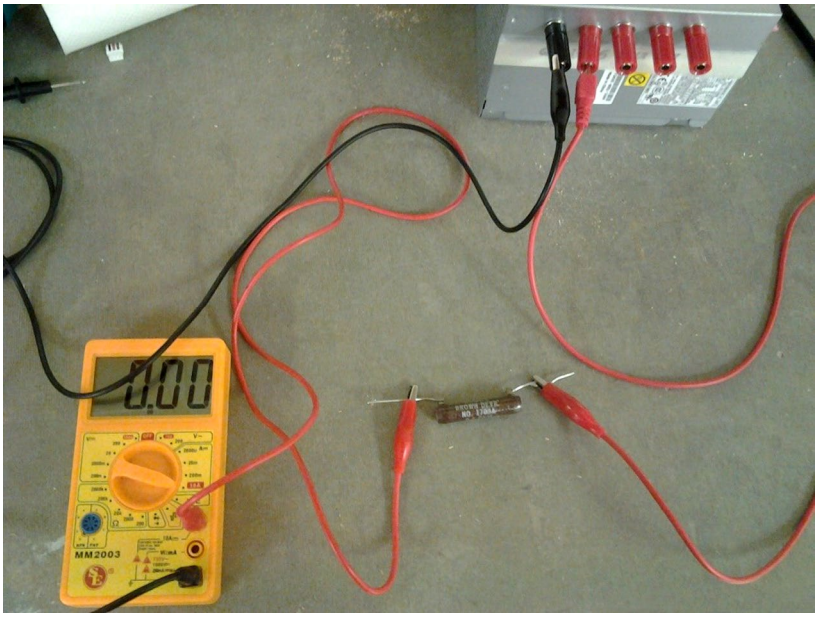
3.6 Měření, měření!

Máte ještě na nepájivém kontaktním poli ten blikač? Tak ho zase rozpojte, vytáhněte integrovaný obvod, vytáhněte kondenzátor, rezistory, odpojte všechny propojky, nechte jen napájení. Máte?

Tak, a teď vezměte multimetr, přepněte ho na měření napětí, zvolte rozsah 20 V a připojte jej na napájecí napětí. Černý kablík na záporný pól, červený na kladný. Pokud používáte zdroj a stabilizátor a držíte se návodu, budete tam mít 5 voltů. Plus mínus… Zapište si někam naměřenou hodnotu.

Přepněte multimetr na rozsah 20 mA. Vezměte rezistor s odporem 10 000 ohmů (10 kΩ, značeno jako 10K) a zapojte ho jedním vývodem ke kladnému napětí. Ke druhému vývodu přiložte červený hrot multimetru. Černý připojte na záporný pól napájecího zdroje. Na displeji uvidíte hodnotu, která se bude pohybovat někde okolo 0,5. Výsledek je v miliampérech, tedy 0,5 miliampéru, tedy 0,0005 ampéru.

Zkuste si pokus opakovat s jiným rezistorem, u kterého si změříte jeho odpor. Naměřené údaje si zapište do tabulky:



POZOR!!! Nikdy nepřipojujte multimetr v režimu „měření proudu" (tedy rozsahy 2 mA, 20 mA, 200 mA a podobně) přímo na napájecí zdroj! Vždy jen přes rezistor!

Napětí (V)	Odpor (Ω)	Proud (A)
5	10K (10000)	0,0005
5		
5		

⌀ https://eknh.cz/ohms

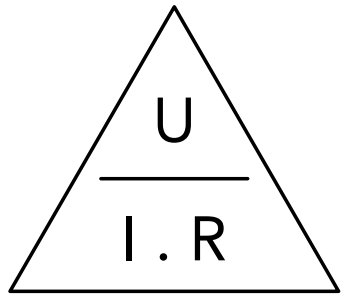
3.7 Ohmův zákon

Pokud jste měřili dobře, zjistíte, že pro výše uvedené hodnoty platí (s určitou mírou tolerance, danou nepřesností měření a použitých součástek) následující vztah:

proud = napětí / odpor (A; V, Ω)

Tomuto vztahu se říká **Ohmův zákon** a je to nejdůležitější pravidlo, které v elektrotechnice platí. Zapisuje se i ve tvaru **napětí = proud × odpor**, matematickým zápisem jako U = I . R

Pro snazší zapamatování se někdy používá zápis v trojúhelníku:



Stačí si pak jednu veličinu zakrýt, a zbývající dvě řeknou, jak ji spočítáme. U = I . R, I = U / R, R = U / I.

3.8 Výkon

Proud nám teče vodičem kolem dokola, to je moc hezké, ale sám o sobě nic užitečného nedělá. Musí se někde spotřebovávat. Nejjednodušší spotřebič je třeba stará dobrá žárovka – to je tenké kovové vlákno, které má poměrně vysoký odpor. Pokud proud překonává odpor, část elektrické energie se přeměňuje v tepelnou. Množství takto přeměněné energie se nazývá **výkon**, měří se ve wattech (W), označuje se P a jeho velikost je rovna součinu proudu a napětí, tedy **P = U . I**.

Elektrický výkon je pro nás velmi užitečná veličina, protože udává nejen to, jak silný spotřebič máme (žárovku, motor apod.), ale na druhé straně i to, kde elektřinu zbůhdarma pálíme. Pokud zapojíme rezistor 330 ohmů na baterii 1,5 V, poteče jím proud 4,54 mA, a to znamená, že výkon, který se na něm „protopí", bude 6,8 mW. To zvládnou i miniaturní rezistory. Ovšem někdy se stane, že navrhneme obvod, kterým poteče velký proud, a pak se může snadno stát, že spotřebovaný výkon bude tak velký, že nezbyde než použít pořádně velké součástky a intenzivně chladit! Hodně „topí" například stabilizátory napětí ve zdrojích, výkonové zesilovací prvky, nebo mikroprocesory.

Mimochodem, když si do předchozího výrazu P = U . I dosadíte podle Ohmova zákona (U = R . I), tak zjistíte, že lze výkon vyjádřit i pomocí odporu a proudu, jako P = R . I . I = R . I^2. Tedy že výkon je přímo úměrný odporu, který proud překonává, a druhé mocnině velikosti proudu. Což je důvod, proč se pro dálkový přenos elektřiny používá velmi vysoké napětí s malými proudy: snižují se tím ztráty ve vedení. Elektrické vedení o délce 100 kilometrů má nezanedbatelný odpor, a máme

jedinou možnost, jak snížit „propálený výkon“: snížit procházející proud. A k tomu, abychom mohli snížit proud, potřebujeme zvýšit napětí.



Zkuste si to spočítat:

V zásuvce máme 230 voltů, odebrat z ní můžeme 6 ampérů. Výkon, který nám zásuvka poskytne, je tedy 230 . 6 = 1380 wattů (1,38 kW). Kdybychom tuto energii přenášeli od elektrárny až do domu napětím 230 voltů, potřebujeme přenést vodičem zmíněných 6 ampérů. Kdyby mělo vedení odpor 0,3 ohmu na kilometr, což je přibližně reálná hodnota, bude jeho celkový odpor 30 ohmů, a výkon, který se po cestě spálí, bude $30 . 6^2 = 1080$ wattů!

Když zvýšíme pro přenos napětí tisícinásobně, tedy na 230 000 voltů (230 kV – ve skutečnosti se používá velmi blízká hodnota 220 kV, i vyšší), stačí nám tisíckrát menší proud, tedy 6 mA. Přenesený výkon stále odpovídá: 230000 . 0,006 = 1380. Jenže na stejném vedení o odporu 30 ohmů spálí proud 0,006 A pouze $30 . 0{,}006^2 = 0{,}00108$ wattu. Ano, vidíte dobře: 1,08 miliwattu, což je miliontina oproti předchozímu případu.

To je důvod, proč je v dálkovém vedení velmi vysoké napětí. Není to proto, aby bylo nebezpečné dotýkat se drátů i na zem spadlých, ale proto, že při vedení na dlouhé vzdálenosti je výhodnější vést vyšší napětí a menší proud.

⌀ https://eknh.cz/vedeni

Jazyková vsuvka

Je „ten ampér“, nikoli „ta ampéra“. Takže doma máte jistič (třeba) na 25 ampérů, nikoli na 25 ampér. Důvod je jednoduchý: Ampér, jako jednotka proudu, je pojmenován po francouzském matematikovi,

který se jmenoval André-Marie Ampère, a přestože tam může mást ta Marie, tak věřte, že to byl muž. Proto ten ampér a proud má velikost pět ampérů, nikoli pět ampér.

Volt problém nedělá – a přitom se fyzik, po kterém nese název jednotka napětí, jmenoval Alessandro Volta. Třetí do party je německý fyzik Georg Simon Ohm, po němž je pojmenována jednotka elektrického odporu. Jen pro jistotu – čte se to [óm].

Čtvrtý kolega je Skot James Watt, a s ním jsou taky obtíže. Leckdo je schopen říct, že „spotřeboval sto kilowatt" – jenže ona není žádná „kilowatta"! Tu si schovejte, až budete nakupovat obvazy. V elektrotechnice vždy a pouze „sto kilowattů". Navíc, a to dodávám na okraj, se spotřeba neměří ani v kilowattech, ani v kilowattách, ale v kilowatthodinách.

3.9 … a malé opáčko

Já bych se k tomu nevracel, ale ono je to fakt hodně důležité, tak mi připadá potřebné to znovu zmínit.

Napětí je statická veličina, která se vždy měří mezi dvěma body. Pokud jsou v těchto bodech různé elektrické potenciály, je mezi těmito body elektrické napětí.

Spojíte-li dva body, mezi nimiž je napětí, vodivým materiálem (vodičem), začne tímto vodičem protékat **proud**. Velikost tohoto proudu je přímo úměrná napětí a nepřímo úměrná **odporu** vodiče. Proud se proto měří v jednom bodě (vodiči), protože proud procházející vodičem je ve všech bodech tohoto vodiče stejný. V praxi tak, že jeden konec připojíte přes ampérmetr.

3.9.1 Násobky a podíly

Kolik ampérů je jeden miliampér a kolik ohmů má jeden kiloohm? Já myslím, že to všichni víte, ale pro jistotu to tu sepíšu.

	Předpona	**Napětí**	**Proud**	**Odpor**	**Kapacita**	**Výkon**
Základní jednotka	-	1 V (Volt)	1 A (Ampér)	1 Ω (Ohm)	1 F (Farad)	1 W (Watt)
× 1000 (10^3)	kilo-	1 kV	1 kA	1 kΩ	1 kF	1 kW
× 10^6	mega-	1 MV	1 MA	1 MΩ	1 MF	1 MW
× 10^9	giga-	1 GV	1 GA	1 GΩ	1 GF	1 GW
× 10^{-3}	mili-	1 mV	1 mA	1 mΩ	1 mF	1 mW
× 10^{-6}	mikro-	1 μV	1 μA	1 μΩ	1 μF	1 μW

	Předpona	Napětí	Proud	Odpor	Kapacita	Výkon
$\times 10^{-9}$	nano-	1 nV	1 nA	1 nΩ	1 nF	1 nW
$\times 10^{-12}$	piko-	1 pV	1 pA	1 pΩ	1 pF	1 pW

V tabulce jsem sepsal všechny kombinace jednotek a jejich násobků, i když některé jsou příliš velké nebo naopak příliš malé na to, abychom se s nimi v praxi běžně setkávali. Kilovolty potkáte třeba v rozvodné síti, mega- a gigavolty nejspíš nikde, totéž platí pro ampér. U napětí se setkáte nejčastěji s volty a milivolty, u proudu s jednotkami ampérů, a pak s mili- a mikroampéry. Odpor zase budete nejčastěji měřit v ohmech, kiloohmech a megaohmech, zlomky ohmů v praxi nepotkáte. U kapacity naopak budete pracovat s mikrofarady, nanofarady a pikofarady. Elektrický výkon nejčastěji potkáte v mikroelektronice jako watt a miliwatt, doma třeba i kilowatt, a když budete číst o elektrárnách, narazíte na mega- a gigawatty.

3.10 Zkratky u značení

Ujalo se nám mezi elektroniky takové zkracování. Třeba hodnotu 2,2 kiloohmu nepíšeme jako 2,2k (ohm vynecháme, u rezistoru je jasné, že jde o ohmy), ale jako 2k2. Prostě to písmenko, co označuje předponu, posuneme na místo desetinné tečky. Takto:

Hodnota	Značení
2,2	2R2
22	22R (nebo jen 22)
220	220R (nebo jen 220)
2 200	2K2
22 000	22K
220 000	220K (nebo M22)
2 200 000	2M2
22 000 000	22M

A jak je to třeba u kondenzátorů? Úplně stejně, ale za základ se bere 1 pF. Kondenzátor 22K = 22 000 pF = 22 nF. Kondenzátor 4M7 bude tedy $4{,}7 \times 10^6$ pF = 4,7 µF.

3.11 Vyvolená čísla

Dobrá, a teď prakticky: Co když mi výpočtem vyjde, že potřebuju rezistor 283 ohmů? Vyrábí někdo takovou hodnotu?

No, budete se možná divit, ale ne. Pro rezistory, kondenzátory a součástky se ujalo, že se vyrábí v určité řadě a toleranci, a tyto řady se značí E6, E12, E24, E48, … V každé řadě je mezi hodnotami 1 a 10 definováno 6 (12, 24, 48, …) hodnot pomocí takzvaných *vyvolených čísel*. To jsou čísla, která jsou zvolená tak, že poměr mezi sousedními členy je **přibližně** stejný.

Pro řadu E6 to jsou následující hodnoty: 1,0 – 1,5 – 2,2 – 3,3 – 4,7 – 6,8

Zkuste si to sami:

- $1 \cdot 1{,}5 = 1{,}5$
- $1{,}5 \cdot 1{,}5 \approx 2{,}2$
- $2{,}2 \cdot 1{,}5 \approx 3{,}3$
- $3{,}3 \cdot 1{,}5 \approx 4{,}7$
- $4{,}7 \cdot 1{,}5 \approx 6{,}8$

Řada E12 má prvky 1,0 – 1,2 – 1,5 – 1,8 – 2,2 – 2,7 – 3,3 – 3,9 – 4,7 – 5,6 – 6,8 – 8,2

Proto píšu o „rezistoru 2K2“ a ne třeba 2K9 – protože součástky se nevyrábějí ve „všech možných“ hodnotách, ale pouze v hodnotách z určitých řad.

Řada E6 zajišťuje, že pro každou zvolenou hodnotu existuje v této řadě *vyvolené číslo*, odlišné maximálně o 20 %. U řady E12 to je 10 %, u řady E24 pak 5 %

V amatérských konstrukcích s číslicovými obvody se nejčastěji setkáte s řadami E6 a E12, které jsou dostatečně přesné pro zamýšlené použití. Málokdy bývá hodnota naprosto kritická – snad s výjimkou časování.

Samozřejmě, dát místo rezistory 330R rezistor 3K3 už je rozdíl o řád, to už se může projevit negativně na funkci, ale nahradit jej například rezistorem 470R ve většině případů nejspíš neudělá problém. (Problém to udělá například v obvodu časovače...)

Každopádně doporučuju udělat si doma zásobu nejčastějších nejpoužívanějších hodnot rezistorů. Pro číslicovou techniku bych volil hodnoty mezi 100R a 100k – především 220R a 330R, které se hodí k LED, pak 1K, 2K2, 4K7, 6K8, 10K, 22K, 68K a 100K.

U kondenzátorů pak 22p, 33p, 100K (= 100 nF), 4M7, 10M (= 10 µF). To je naprostý a nezbytný základ.

3.12 Pro lepší představu

Pro mnoho začátečníků jsou jednotky příliš abstraktní, těžko si pod nimi něco představí. Jeden ampér – kolik to je? Je to hodně, nebo málo? No, jak se to vezme…

Pokud se budeme pohybovat v oblasti číslicové techniky, které je tato kniha primárně věnována, budeme pracovat nejčastěji s napětím 5 voltů, někdy i mírně vyšším – třeba 12 voltů. Proudy, které našimi obvody potečou, se budou měřit na miliampéry. 200 miliampérů (0,2 A) už pro nás bude poměrně velký proud. Třeba skrz LED budeme pouštět maximálně 20 mA (větší proud by ji pravděpodobně zničil).

Rozvodná a přenosová soustava pracuje s vyšším napětím – doma v zásuvkách máme 230 voltů, ale v „páteřní síti" se pracuje s napětím 400 kV, 220 kV či 110 kV.

Rezistory budeme používat v řádech stovek ohmů až desítek tisíc ohmů. Měděný vodič má odpor blízký nule. Záleží na jeho délce, průřezu a na teplotě. Suchý vzduch je považován za izolant, ale není ideální. Vakuum je téměř ideální izolant. U vody záleží na složení – tvrdá voda s velkým množstvím solí vede elektrický proud docela dobře, měkká voda, třeba dešťová, hůř. Destilovaná voda bez jakýchkoli příměsí proud nevede skoro vůbec. Ovšem od určité meze dochází ve vodě k jejímu rozkladu (elektrolýze) na vodík a kyslík, a pak proud protéká.

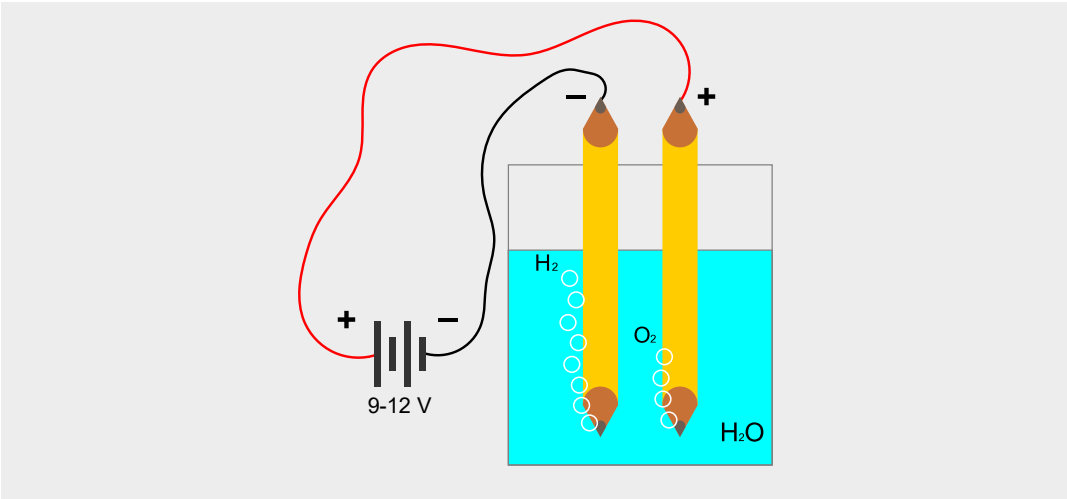
Nevěříte?

Vezměte si prosím dvě tuhy z tužky. Třeba náhradní náplně do verzatilky, ať nemusíte zbytečně ničit dřevěné tužky. Ideálně tak 10 centimetrů dlouhé. K jedné připojte kladný pól, ke druhé záporný pól zdroje. Do sklenice nalijte vodu a obě uhlíkové elektrody do ní ponořte tak, aby se nedotýkaly. Zapněte napájení a změřte protékající proud. V závislosti na čistotě a „měkkosti" vody naměříte nějaký malý procházející proud. Platí, že čím měkčí voda, tím menší proud.

Stále měřte, a zkuste do vody přidat třeba lžičku soli, nebo octa.

Proud začne procházet, a u elektrod se budou tvořit bublinky plynů. U záporné je to vodík, u kladné kyslík – ale to určitě znáte ze školy.

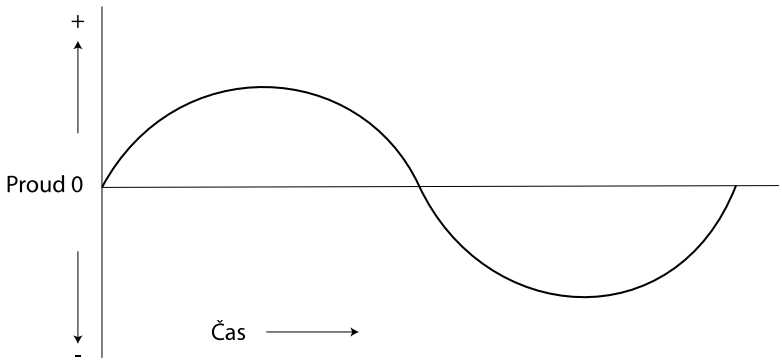
Pro člověka není nebezpečné ani tak napětí, jako proud, který prochází tělem. Navíc lidská kůže má tu vlastnost, že její odpor není konstantní, mění se v závislosti na napětí. Nízkému napětí klade vyšší odpor, jak napětí roste, tak se odpor snižuje. Za bezpečné bývá považované napětí do 24 voltů, které člověka sice zabrní, ale neuškodí. Napětí 230 voltů, které je v zásuvkách, už může být za jistých okolností smrtící. Některé elektrické paralyzéry vytvářejí napětí v řádech tisíců voltů, ale s malým proudem, takže rána je velmi citelná, člověk je na chvíli ochromen, může upadnout do bezvědomí, ale díky malému proudu není takový zásah pro zdravého jedince smrtící.



3.13 Střídavý proud

Až dosud jsme mlčky předpokládali, že zdroj napětí má svůj kladný a záporný pól, elektrony proudí od záporného ke kladnému, proud od kladného k zápornému, a tak to je a v čase se to nemění. Proud má stále stejný směr, proto se mu říká stejnosměrný. Anglicky Direct Current, neboli DC.

Kromě stejnosměrného proudu se v elektrotechnice využívá i proud střídavý. Takový proud mění svůj směr, chvíli teče jedním směrem, chvíli druhým. Takový proud má své výhody – lze jej velmi dobře transformovat v transformátorech, proto se používá k rozvodu elektrické energie. Střídání směrů si ale nepředstavujte jako přepínání „dopředu – dozadu“. Ve skutečnosti se směry plynule mění. Když si průběh takového proudu nakreslíme v čase, dostaneme hezkou sinusovku:



Samozřejmě že střídavý proud může mít i jiný průběh – obdélníkový, trojúhelníkový, pilový, nepravidelný apod. To uvádím jen pro úplnost. Nejčastěji se pod pojmem „střídavý proud" myslí právě takovýto sinusový.

A teď otázka pro chytré hlavy: Jak u zdroje střídavého proudu poznáme, kde je plus a kde mínus? No, nepoznáme. U zdrojů střídavého proudu nemá takové dělení smysl. Proto se používá označení jiné: jeden z vývodů se bere jako neutrální – říká se mu **nulový** (též **nulák**), druhý vývod je fáze. Napětí mezi fází a nulovým vodičem se v čase mění, chvíli je kladné, chvíli záporné.

V zásuvkách máme 230 voltů (možná vás učili hodnotu 220, ale to už od 90. let minulého století neplatí). To ale neznamená, že maximální napětí (když je sinusovka v maximu nebo v minimu) je 230 voltů. To číslo označuje takzvané „efektivní napětí". Maximální je skoro jedenapůlkrát vyšší (asi 325 voltů) – přesněji je to *(odmocnina ze dvou) krát* (cca 1,4142). Efektivní napětí násobené proudem udává skutečný výkon střídavého proudu, proto se používá právě tato hodnota.

Směr proudu v rozvodné síti se změní padesátkrát za sekundu – tedy u nás. Třeba v USA je to jinak, tam je frekvence o něco vyšší – 60 změn za sekundu (kmitočet se udává v jednotkách „změn za sekundu", jednotka je Hertz, zkratka Hz, to jen pro pořádek). Navíc v USA nepoužívají 230 voltů, ale 120 voltů…

Rozdíl ve frekvencích nezpůsobuje jen potíže při dovozu elektroniky z USA (pokud není uzpůsobený napájecí obvod). Způsobil například i to, že evropské systémy televizí PAL / SECAM používaly obnovovací frekvenci 25 Hz, americký NTSC používal ~ 30 Hz (polovina síťového). Ačkoli dnes se už z frekvence sítě pracovní kmitočty neodvozují, tyto rozdíly zůstaly… Tak až si někdy všimnete při konvertování videa otázky na FPS (Frames Per Second) a budete přemýšlet, proč 25, 30 a 29,970… Těch 29,970 je proto, že při 30 docházelo k interferencím s nosným kmitočtem barvy, proto byla frekvence obrazu o něco snížena – ale jedovatá přezdívka NTSC jako „Never The Same Color" zůstala.

3.14 Zkrat

Též v tajné řeči bratří drátů „kraťas", neboli „krátké spojení", nastává, když spojíte kladný pól zdroje se záporným nějakým vodičem napřímo, bez rezistoru. Odpor vodiče je minimální, to znamená, že – podle Ohmova zákona – poteče obrovský proud, a to může způsobit několik věcí.

Zaprvé: zvedne odběr na maximum. Zdroj dá všechnu elektřinu co má, vymáčkne ze sebe ochotně každý elektron, takže už na další zařízení, která mají větší odpor, nezbude. Bohužel, takovéhle množství energie jen zbůhdarma ohřeje dráty a vyzáří se jako teplo bez jakéhokoli užitku.

Zadruhé: toho tepla bude hodně moc. Tak moc, že může zničit věci v okolí, roztavit plasty, zapálit je…

Zatřetí: zdroj to nemusí vydržet a zničí se.

Obrana proti zkratu je docela jednoduchá: Pojistky, jističe, ochrany proti velkým proudům. Když se vám náhodou doma podaří vytvořit zkrat – stačí šikovně navrtat kabely ve zdi, nebo zapojit do zásuvky spotřebič s uklepaným kablíkem – stane se přesně to, co jsem popisoval, ale naštěstí doma máte jističe (popřípadě snad ještě někde staré porcelánové pojistky). Jistič je zařízení, které měří protékající proud, a jakmile stoupne nad danou mez (třeba 6 ampérů u světel, 10 ampérů u zásuvek atd.), tak přeruší napájení. Páčkou je pak potřeba jej opět „nahodit".

U pojistek je tenký drátek, který se velkým proudem roztaví a tím přeruší přívod elektřiny. Pojistka se takzvaně „spálí" a je potřeba ji – samozřejmě až po odstranění příčiny – nahradit novou.

Třetí možnost, kterou jsou vybavené třeba laboratorní zdroje napětí, jsou proudové ochrany. Jakmile odběr překročí nastavenou mez, elektronika zasáhne, vypne napětí a rozsvítí varovný signál.

3.15 Multimetr jako zkrat?

Už jsem to tu taky psal, ale pro jistotu to ještě jednou zopakuju:

Multimetrem měříte **napětí mezi dvěma body**. Multimetr má, když je přepnutý na měření napětí, mezi sondami velmi velký odpor, takže skrz něj teče jen minimální proud. Čím větší *vnitřní odpor* má, tím méně ovlivňuje měřený obvod.



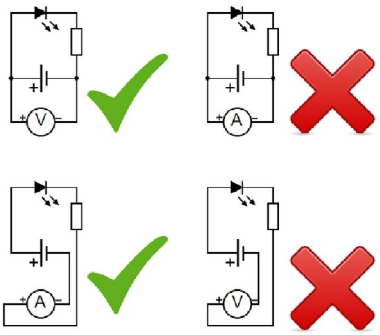


Když multimetrem měříte **proud**, tak vždy **v jednom bodě** tak, že proud teče **skrz multimetr**. Multimetr, přepnutý na měření proudu, má vnitřní odpor minimální – právě proto, aby skrz něj tekl proud bez problémů a byl měřením co nejméně ovlivněn.

Co se stane, když přepnete multimetr na měření proudu a připojíte jej na póly baterie? *Nedělejte to, fakt ne!* Bude mít téměř nulový odpor, a vy jím vlastně zkratujete baterii. Dále platí všechno to, co jsme si řekli o zkratu výš. V tomto případě bude pravděpodobným výsledkem zničení multimetru.

A teď ruku na srdce – co byste jako chtěli měřit? „Proud baterie"? Pokud si myslíte, že baterie má „proud", tak jsem jako lektor selhal. Zkuste se prosím vrátit ještě jednou na začátek této kapitoly, třeba jste to jen v roztržitosti zapomněli…

A protože jeden obrázek vydá za tisíc slov, tak zde máte… Vlevo správný způsob, vpravo nesprávný. Zapamatujte si to!



3.16 Elektromagnetická indukce

Já před vámi v téhle knize hodně věcí zatajím. Tedy přesněji řečeno: nebudu o nich psát. Třeba namátkou celou širokou oblast nízkofrekvenčních signálů, silovou elektřinu, audiotechniku, radiotechniku, magnetostrikci, elektrický oblouk i katodové paprsky, elektronky, Teslův transformátor, Van der Graafův generátor, varikap, transil, tříbodový souběh superhetu i Čebyševův filtr třetího řádu. Ale tuhle věc o elektřině byste měli vědět. Sice ji ze školy už určitě znáte a máte ji někde uloženou v hlavách, ale nejspíš jako zlomkovitou znalost – něco s pravou rukou tam bylo a s vodičem v magnetickém poli a tak.

Vodič, kterým protéká proud, okolo sebe vytváří magnetické pole. Můžete si to vyzkoušet, pokud máte kompas a drát. Když položíte kompas na drát a pustíte skrz něj proud, třeba do žárovky, tak se střelka kompasu vychýlí.

Nebo: Vezměte vodič, třeba ten, co máte k propojování na kontaktním poli, a větší hřebík. Vodič omotejte okolo hřebíku a na jeho konce připojte třeba plochou baterii. Hřebík začne fungovat jako magnet, což se dá ověřit tím, že se k němu chytají menší kovové předměty – maličké šroubky, kovové sponky na papír a tak. Díky tomu, že jste delší drát namotali na menší prostor, působí magnetická síla koncentrovaněji.

Funguje to i obráceně?

Odpojte baterii, místo ní připojte multimetr, přepnutý na měření milivoltů, a teď k hřebíku přibližujte magnet a zase ho vzdalujte. Sledujte, co se děje.

Funguje to? Funguje, ale trochu jinak.

Ve vodiči v magnetickém poli vzniká elektrické napětí, ovšem pouze ve chvíli, kdy se mění intenzita tohoto magnetického pole. Třeba když se vodič pohybuje, nebo když magnetické pole odstraníte nebo vytvoříte.

Když si k výše zmíněnému hřebíku namotáte podobně druhý vodič (klidně přes ten první), máte možnost zkusit, jak se to celé bude chovat. Jeden vodič použijte jako elektromagnet, tzn. připojte k němu baterii, k druhému vodiči připojte měřicí přístroj, a uvidíte, že se nic neděje. Děje se jen ve chvílích, kdy baterii připojujete a odpojujete... V takových chvílích se na měřicím přístroji objeví pulsy.

Kdybychom tedy na vstupu ten signál pravidelně přerušovali, tak... No, zkuste si to.

A tomu efektu, kdy změnou magnetického pole vzniká elektrické napětí, se říká **elektromagnetická indukce**.

Tohle jsem před vámi utajit nemohl. Ale věřte mi – chtěl jsem! Nakonec se ukázalo, že to nepůjde, protože indukce může za spoustu věcí, někdy špatných (rušení signálů), někdy dobrých (třeba výroba elektrické energie v dynamu).

Později na indukci ještě několikrát narazíme...

3.17 Značky pro schémata

Když se podíváte na zapojení našeho blikače na nepájivém kontaktním poli, zjistíte, že není moc přehledné. Není z toho jasné, co se tam děje a proč. Proto se používají schematické značky a nákresy – schémata. *Prosím, čtěte to [schéma], ne [séma]!* Schéma obsahuje všechny součástky a jejich spojení tak, aby bylo člověku, který obvod staví, jasné, co má kde použít a s čím propojit.

Schematické značky slouží, podobně jako třeba notový zápis nebo vývojový diagram, k tomu, aby bylo jasné, co a jak je v obvodu zapojené, i když to v reálu vypadá naprosto jinak.

Na schématu neřešíte velikosti, vzdálenosti, nic z toho. **Nejdůležitější je, aby schéma bylo dobře čitelné a srozumitelné.** Proto se vynechávají detaily, které nejsou podstatné – například u schématu blikače nakreslím jen někam značku „tady bude zem a tady + 5 voltů“, a neřeším, jak a kde se tam vezme. Když nakreslím zem na schématu někam vlevo a pak nakreslím zem někde vpravo, tak není potřeba explicitně udávat, že jsou ty vývody ve skutečnosti propojené, protože jsou oba vodivě připojeny do jednoho místa. To si každý konstruktér pořeší sám. Dál je dobrým zvykem postupovat stejně jako v textu, tedy kreslit schéma tak, že signál jde zleva doprava a shora dolů.

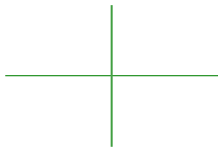
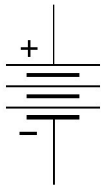
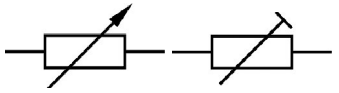
Na značky samozřejmě existují normy a výklady a vykladači, schopní se kdykoli pohádat o kanonickém tvaru té které značky, ale v praxi se setkáte třeba se schématy, které nakreslil někdo v USA a použil lehce jiné značky, než jaké známe my. Důležitější je proto srozumitelnost a pochopitelnost. Pokud je třeba jedno, jestli se zařízení napájí z baterie, nebo ze stabilizovaného zdroje, tak někam prostě jen nakreslím značku, k ní připíšu „+ 5 V“, a každému je jasné, že se do toho bodu musí nějak dostat + 5 voltů, a že tedy někde bude nějaký bod, označený jako „zem“ (silnější vodorovná čárka), a tam přijde druhý pól napájení.

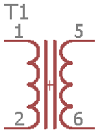
Základem jsou symboly pro vodiče a jejich spojení. To je prostá čára. Pokud se dvě čáry kříží, neznamená to, že jsou spojené. Pokud mají být spojené, je v místě křížení výrazná tečka.

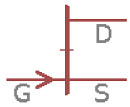
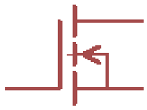
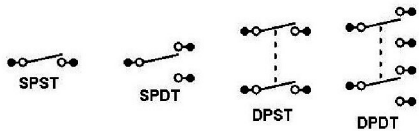
Projděte si nejčastější symboly, s nimiž se setkáte. Za chvíli si o součástkách, které představují, řekneme víc.

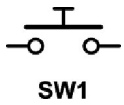
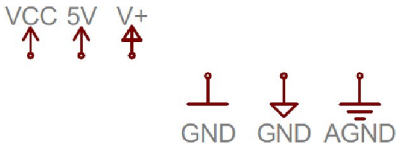
V tabulce uvádím značky tak, jak jsou v databázi programu Eagle – de facto standardu pro amatérskou tvorbu schémat a desek plošných spojů. A přidávám i anglické názvy, protože se s nimi často setkáte.

Značka	**Význam**
	Vodič (Wire)

Značka	Význam
	Křížení vodičů bez spojení (Wire crossing)
	Spojení vodičů (Junction)
	Monočlánek (Power cell)
	Baterie (Battery)
	Rezistor (vlevo evropský symbol, vpravo americký) (Resistor)
	Rezistor s proměnným odporem (potenciometr), trimr (Potentiometer, variable resistor)
	Kondenzátor (vlevo evropský symbol, vpravo americký) (Capacitor)

Značka	Význam
	Polarizovaný (elektrolytický) kondenzátor (vlevo evropský symbol, vpravo americký) (Electrolytic capacitor)
	Cívka (Coil)
	Transformátor (Transformer)
	Dioda (Diode)
	LED (Světlo vyzařující dioda)
	Schottkyho dioda
	Bipolární tranzistor NPN (Bipolar Junction Transistor – BJT, NPN type)

Značka	Význam
	Bipolární tranzistor PNP (Bipolar Junction Transistor – BJT, PNP type)
	Tranzistory v Darlingtonově zapojení (Darlington BJT)
	Tranzistor JFET-N (JFET N-type)
	Tranzistor MOSFET-N (MOSFET N-channel, N-type)
	Tranzistor MOSFET-P (MOSFET P-channel, P-type)
	Spínače a přepínače (Switch, Momentary Switch)

Značka	Význam
	Tlačítko (Push button)
	Napájení, uzemnění (Voltage, Power, Supply; Ground)

A to jsou všechny značky? Ale kdepak, kdepak, ještě jich bude spousta, nebojte… Včas si je ukážeme.

3.17.1 Kroužek, nebo ne?

Některé součástky (tranzistory, diody) mívají schematické značky uzavřené do kroužku. Kroužek symbolizuje pouzdro součástky. Pokud se kreslí tranzistory v integrovaném obvodu, měly by se kreslit bez kroužku. Tak hovoří norma. Jenže zase platí, že norma něco říká, ovšem v praxi narazíte na leccos. Pokud používáte program pro kreslení schémat, jste odkázáni na to, jak autoři tuto problematiku pojali, jestli připravili značky s kroužkem, nebo bez. *Máte dvě možnosti: buď se rozčilovat, že autoři neumí nakreslit značku tranzistoru s kroužkem podle normy, nebo se s tím smířit. Doporučuju druhý způsob, tedy přijmout fakt, že svět není dokonalý, a ušetřit čas, který byste strávili domalováváním kroužků do schémat, vytvořených v nějakém nástroji.*

4 Zdroje napětí

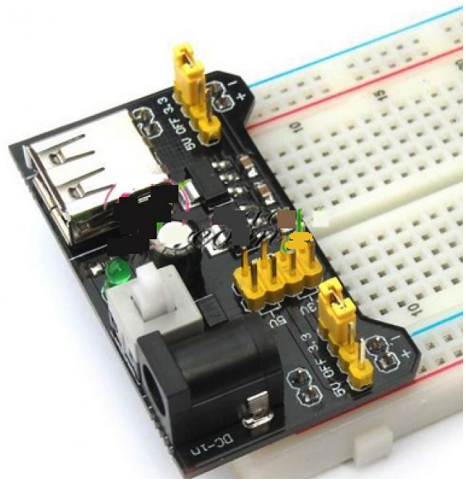
4 Zdroje napětí

Ono se řekne – zdroj napětí! Ale jaký zvolit?

V číslicové technice potřebujeme nejčastěji napětí 5 voltů, navíc poměrně stabilní, bez výrazných výkyvů. Při nižším napětí nemusí naše zapojení fungovat, při vyšším se bude přehřívat, a může se i zničit. Kde ale napájecí napětí sehnat?

Jedna z možností je využít počítač a USB, popřípadě využít nejrůznější nabíječky k mobilním telefonům s rozhraním USB. Jejich výhoda je, že dokáží dát docela velký proud a stabilní napětí. Drobná nevýhoda je, že konektor většinou nelze zapojit do nepájivého kontaktního pole (NKP), a je potřeba vše řešit nějakou redukcí.

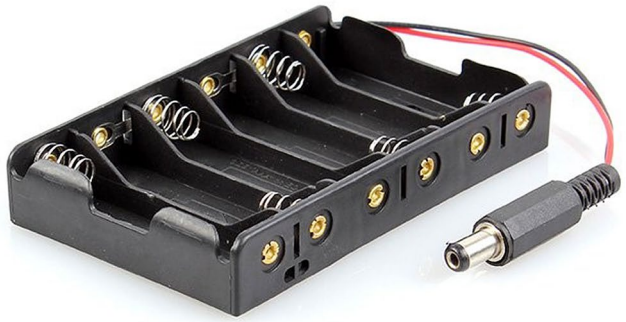
Pro NKP existují různé moduly, které lze koupit v kamenných obchodech i v e-shopech, a ty se starají o správné rozvedení napájecího napětí a jeho úpravu. Většinou obsahují stabilizátor, přepínač napájecího napětí 5 V / 3,3 V a napájecí konektor – většinou souosý pro různé adaptéry, a k tomu konektor USB.



Problém s těmito moduly je, že když do nich přivedete 5 voltů z USB, tak si stabilizátor vezme nemalou část a na výstupu, kde má být pět voltů, naměříte třeba 3,5 V. Což může být problém. Pokud chcete takový modul používat, pořiďte si k němu raději nějaký adaptér, který dá na výstupu okolo 7-9 voltů. Pokud použijete třeba 12 V adaptér, budou se stabilizátory intenzivně zahřívat.

Další možnost je využít baterie. Nejčastěji používané typy jsou AA / AAA monočlánky. Existují držáky, které spojují dva, tři či čtyři monočlánky sériově, takže výsledkem jsou 3 V, 4,5 V nebo 6 V.

Obvody, které používáme, by měly bez problémů fungovat i při 4,5 voltech. Nevýhoda baterií je ta, že jejich kapacita je omezená, a po několika dnech zkoumání, zkoušení a testování se zkrátka vybijí. Nabíjecí monočlánky pomohou jen trochu – po několika dnech zkoumání, zkoušení a testování se vybijí taky, ale můžete je nabít. A nezapomeňte, že tyto monočlánky mají o něco nižší napětí, jen cca 1,2 voltu. Čtyři takové, zapojené sériově (tedy za sebou) dají dohromady 4,8 voltů. Pokud budete experimentovat intenzivněji, doporučuju pořídit si nějaký zdroj, napájený ze sítě.

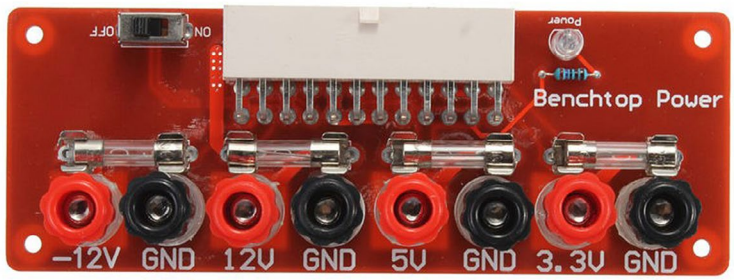


Některá zařízení, vybavená vhodným měničem napětí, lze napájet i z destičkové baterie, která dává 9 voltů. Takové baterie se používají často k napájení např. multimetrů. Existují k nim konektory, které lze připojit například k výše zmíněným napájecím modulům. Ale napájení z takových baterií je vysloveně nouzové, protože pokud nemáte zapojení, navržené speciálně na malý odběr, tak kapacita takové baterie vystačí jen na pár hodin provozu.



Pokud vás elektronika zaujme víc, určitě si pořiďte laboratorní zdroj, u kterého můžete regulovat napětí i proud. Nemusí to být hned profesionální zařízení za několik tisíc Kč, vystačíte i s levnější variantou. Dbejte na to, aby zdroj dokázal dát alespoň 15 voltů a proud 1 A.

Zajímavá možnost je využít zdrojů pro počítače PC s konektorem ATX. Existují adaptéry pro tyto konektory, ze kterých pak lze odebírat napětí 5 V, 12 V i - 5 V. **Dbejte na to, aby takový zdroj měl vždy nějaký minimální odběr – zapojte například na napájecí napětí žárovku. Pokud necháte ATX zdroj bez zátěže, přetíží se a shoří!** *No dobře, nebude hořet jasným plamenem, spíš tak jako zasmrdí a přestane fungovat. Důvodem je, že tyto zdroje jsou zapojeny jako takzvané „spínané zdroje", a v nich slouží cívka jako akumulátor energie, z níž se odebírá přesně dané napětí. Výhodou těchto zdrojů je jejich velmi dobrá účinnost, nevýhodou to, že vyžadují zátěž, která energii z cívky odvede a zpracuje. Pokud ji neodvede, naakumulovaná energie se začne, laicky řečeno, spalovat přímo v cívce.*



4.1 Společná zem

Představte si, že máte dva obvody. Jeden je napájený třeba z USB, například Arduino. Druhý obvod je napájený nějak jinak, třeba vlastní baterií, nebo adaptérem ze sítě – třeba ovladač dveřního zámku. A vy teď chcete, aby se nějaký signál z Arduina dostal k tomu ovladači, protože stavíte automatický kódový zámek… Jak to uděláte?

Tak samozřejmě, nejprve zjistíte, jestli jsou obě zařízení kompatibilní, jestli Arduino dá dost proudu na to, aby zámek sepnul, ale dejme tomu, že to je ověřené, že u zámku je napsáno „Arduino compatible, 5 V logic", takže to spojit můžete.

Tak spojíte Arduino a vstup zámku vodičem, a budete se moc divit, že to nefunguje. Proč? Odpověď je prostá:

Arduino na výstup pošle + 5 voltů, třebas. Jenže jak jsme si říkali: volty jsou vždy „mezi něčím a něčím jiným" – u Arduina to je + 5 voltů mezi výstupem a **jeho** zemí (GND). Elektronický zámek to má ale úplně stejně: očekává napětí 5 voltů mezi vstupem a **svojí** zemí.

Proto je potřeba zajistit, aby *má zem byla i tvá zem* (jak se říká vznešeně ve slavnostních řečech hned předtím, než vyhubí domorodce). Nejjednodušší způsob, jak to zajistit, je prostě to spojit. Spojte vodivě zem u Arduina se zemí u zámku, čímž zajistíte společnou neutrální úroveň a to,

že všechno, co bude + 5 voltů nad potenciálem země u Arduina bude i + 5 voltů nad potenciálem země u elektronického zámku. A v tu chvíli se všechno jako zázrakem zprovozní a začne fungovat.

Nezapomeňte: Pokud spolu mají komunikovat dva obvody, musí mít spojené „zemní" pól napájení. Tomuto spojení se říká **společná zem**, ještě na to několikrát narazíme, a je to docela důležité pravidlo, které si pamatujte!

S tím souvisí ještě jedno pravidlo, a to je to, že když spolu spojujete několik spotřebičů (například počítač a monitor), tak by měly být ve stejné zásuvce. Někdy se může stát, že dvě různé zásuvky mají různou fázi, což v důsledku vede k tomu, že jejich „země" nejsou totožné a je mezi nimi napětí (nebudu se pouštět do podrobností, ale je to tak). Když zapojíte počítač do jedné zásuvky, monitor do druhé, je to OK – ale pak spojíte monitor s počítačem. V tu chvíli propojíte zem u obou spotřebičů. Pokud má každá zásuvka jinou fázi, a tudíž i jiný zemní potenciál, začne téct docela velký proud z jedné zásuvky skrz počítač, videokabel a monitor do druhé. A máte problém.

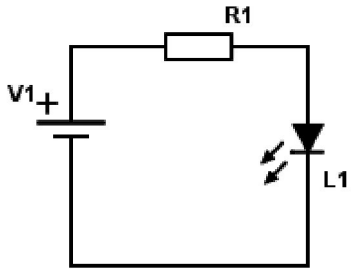
5 Vedle sebe, za sebou

5 Vedle sebe, za sebou

Pojďte, necháme teď chvíli značky značkami a teorii teorií, a zase si něco zapojíme a změříme.

5.1 Svítilna s LEDkou

Úplně nejjednodušší zapojení je taková obyčejná svítilna. Třeba právě s LEDkama. Jak funguje? No, je to jednoduché: máte zdroj energie, třeba baterii, a LEDku, a mezi tím rezistor. Nějak takhle:

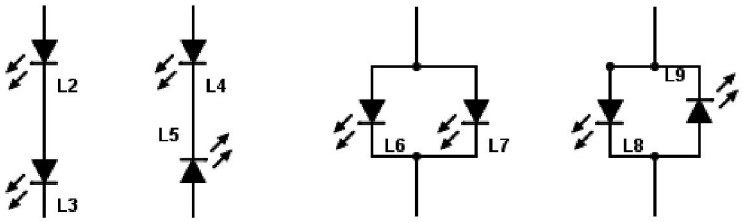


Zkuste si to zapojit. Prozradím vám předem, že když na místě baterie použijete monočlánek s napětím 1,5 voltu, tak vám to svítit nebude. Musíte použít alespoň dva za sebou, ještě lépe plochou baterii (v níž jsou tři za sebou), nebo například Arduino, z něhož můžete odebírat + 5 voltů.

Rezistor použijte třeba náš oblíbený 330R. U baterie 4,5 voltů propustí 4,5/330 = 13 mA, což je dostatečný proud na to, aby LED svítila, a zároveň bezpečný natolik, aby se nepřepálila. Když chcete, aby LED svítila víc, můžete zkusit snížit velikost odporu – můžete jít třeba na 220R, pak bude proud skoro 20 mA, a to je pro většinu LED maximum.

Jenže co když chcete svítit víc, ještě víc? Jak na to? No, co třeba zapojit víc, víc diod? Zkuste si to. Zapojte si dvě najednou…

Jak? No, v zásadě máte čtyři možnosti, jak místo jedné LED zapojit dvě:



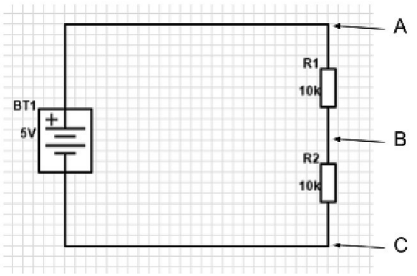
Možnosti 1 a 3 dávají smysl. Vyzkoušejte si je, při které z nich budou LED svítit nejvíc. Možnost 2 je nefunkční, takhle vám LED svítit nebudou. Možnost 4 je zajímavá v tom, že takhle vám bude svítit vždy alespoň jedna LED, i když přepólujete zdroj, přehodíte kladný a záporný pól, ale většího jasu nedosáhnete…

A co když zapojíte tři LED? Zvláštní – tři za sebou nebudou svítit vůbec, tři vedle sebe ano… Jak je to možné? Proč je v tom rozdíl? Přesně to se dozvíte v této kapitole.

⌀ https://eknh.cz/lsp

5.2 Sériové zapojení

Zkuste udělat následující obvod:



Vidíte dva rezistory, zapojené „za sebou“ – tedy sériově. Proud teče od kladného pólu baterie (taková je konvence, vzpomeňte si) horním vodičem do rezistoru R1, skrz něj do rezistoru R2, a z něj spodním vodičem k zápornému pólu baterie.

Když si takové zařízení postavíte na kontaktním poli, nebude nijak zajímavé. Nebude blikat, svítit, bzučet – jen bude vybíjet baterii, tak jej prosím nenechte zapojený dlouho! Elektřina z baterie se zbůhdarma mění v rezistorech na teplo. Ale přeci jen si ho zapojte – ne proto, abyste viděli, jak se vybíjí baterie, ale abyste si změřili pár údajů.

Začneme napětím, a změříme si nejprve napětí mezi body A a C. Pokud zanedbáme odpor vedení, a ten se většinou zanedbává, tak je situace stejná, jako bychom měřili přímo na baterii. Tedy měli bychom naměřit 5 voltů, popřípadě míň, podle zdroje, jaký používáte. Já používám stabilizovaný zdroj 5 V, takže jsem naměřil 4,997 V – a to je plně v toleranci.

Teď je otázka: jak to bude s napětím v bodu B? Jaké bude napětí vůči bodu A a jaké vůči bodu C? Nebudu nic prozrazovat, změřte si to.

Máte? A vyšlo vám pokaždé totéž? Mělo by, pokud jste použili rezistory se stejnou hodnotou odporu, tedy 10 kiloohmů. Napětí mezi body A-B (říká se taky „napětí na rezistoru R1“) a B-C („na rezistoru R2“) je pokaždé 2,5 V, tedy polovina napájecího napětí. Samozřejmě počítejte s nepřesností součástek a měření, ale princip je, myslím, jasný. Ale ještě si to ověřte – zkuste zapojit třetí rezistor stejné velikosti R3 za rezistor R2, a změřit napětí na jednotlivých rezistorech teď. Mělo by být zhruba 1,6 voltů, tedy třetina napájecího napětí.

Jako by na každém rezistoru ubyla část napájecího napětí. A ono to tak opravdu je. Proto někdy hovoříme o takzvaném *úbytku napětí* na rezistoru (nebo obecně na jakémkoli spotřebiči). Tento úbytek napětí se promění na jinou energii – v případě rezistoru na teplo.

Dalo by se z toho odvodit, jaký je výsledný odpor dvou (nebo obecně několika) rezistorů za sebou? Zatím ještě ne, na to potřebujeme (vzpomeňte si na Ohmův zákon) znát proud, který obvodem protéká.

Změřte tedy proud v bodech A, B a C. Nezapomeňte, že při měření proudu musíte obvod rozpojit a ampérmetr zapojit do cesty proudu. Máte?

Tentokrát se, překvapivě, nic nedělí, a ve všech třech bodech je proud stejný. Nevím jak u vás, ale u mne to s rezistory 10K a napájecím napětím 5 voltů bylo 0,25 mA. Což by znamenalo, že výsledný odpor je 20 kΩ, tedy vlastně R1 + R2.

Prověřte to. Zkuste si místo jednoho rezistoru 10K dát třeba rezistor 4K7, nebo 1K. Zkuste si zapojit za sebe tři rezistory a měřte vždy protékající proud. Z něj spočítáte velikost celkového odporu. A mělo by vám vyjít, že **výsledný odpor R sériově zapojených rezistorů R+, R2, … je roven součtu všech odporů R1 + R2 + … + Rn**

⌀ https://eknh.cz/delnap

5.3 Dělič napětí

Když už máme ty rezistory takto zapojené, tak si zkuste změřit opět úbytek napětí na obou rezistorech (tedy mezi body A-B a B-C), pokud bude jejich hodnota různá. Zapište si výsledky:

Odpor R1	Odpor R2	Napětí A-B	Napětí B-C	Napětí A-C	Proud
10000	10000	2,5	2,5	5	0,00025
1000	10000	0,45	4,55	5	0,00045
1000	1000	2,5	2,5	5	0,0025

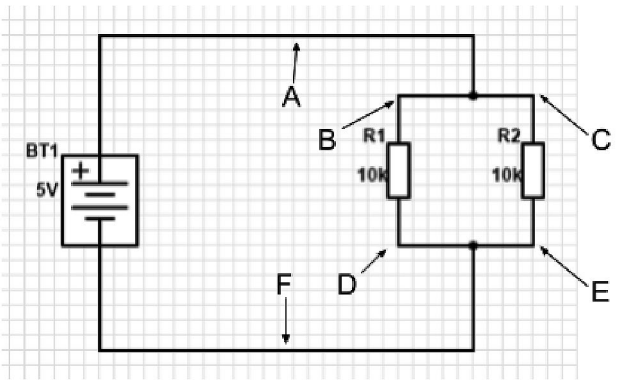
Co z naměřeného vyplývá? Že pokud jsou odpory obou rezistorů stejné, tak je napětí na nich rovněž stejné. Pokud se liší, je na jednom z nich menší úbytek, na druhém větší. V jakém poměru? Šlo by to spočítat…

Mějme sériové zapojení dvou rezistorů. Už víme, že výsledný odpor je roven součtu odporů obou rezistorů, tedy R1 + R2. Označme si tento součet Rx. Proud, který teče obvodem, bude tedy I = U / (R1 + R2) => I = U / Rx. No a napětí na jednotlivých rezistorech (označme si je U1 a U2) bude zase podle Ohmova zákona U = I . R. Tedy: U1 = I . R1 a U2 = I . R2. Vidíme, že větší úbytek napětí bude na rezistoru s větším odporem.

Když vzorec rozepíšeme, dostaneme podobu U1 = U . (R1 / Rx), U2 = U . (R2 / Rx). Vzorec jde zobecnit pro víc rezistorů: Un = U . (Rn / Rx)

5.4 Paralelní zapojení

Uděláme jen drobnou změnu. Místo „za sebou“ zapojíme rezistory „vedle sebe“ – tedy místo sériového zapojení zvolíme paralelní. Takto.



Na takové zapojení můžeme opět pohlížet jako na zapojení jednoho rezistoru s celkovým odporem Rx. Jaká bude jeho velikost? Schválně, zkuste hádat:

a. Bude stejná jako u sériového zapojení

b. Bude jiná

Logická úvaha: No, stejná asi nebude, to bych se na to neptal a nedělal s tím takové štráchy, že… Ale co když jen tak blufuju? Nebude lepší to zase změřit?

Tak jo, pojďme to změřit. Napájecí napětí je 5 V, napětí na rezistoru R1 (mezi body B-D) je… aha, a na rezistoru R2, tedy mezi body C-E…? Aha aha! Ono je to vlastně jasné – z hlediska měření napětí jsou si body A, B a C rovny – jsou spojeny vodičem, a ten má, jak jsme si řekli, pro nás zanedbatelný odpor. Totéž platí pro body D, E a F. Tedy je logické, že napětí mezi bodem A (nebo B, či C) a bodem F (nebo D, E) bude stále stejné. Tedy těch 5 voltů. Paralelní zapojení tedy nefunguje jako dělička napětí; na všech rezistorech je stejný úbytek napětí.

Úvaha: Bude na nich stejné napětí, i když bude mít jeden 10K a druhý jen 1K? Když se nad tím zamyslíte a uvědomíte si to, že jsou body A-B-C a D-E-F spojeny, tak by na nich mělo být stále stejné napětí. Mezi body B a C není nic, kde by mohlo dojít k úbytku, stejně tak mezi body D a E, takže není důvod, proč by se napětí B-D mělo lišit od C-E. Ale zkuste si to sami proměřit.

Než to změříte, tak jen připomenu, že vodiče samozřejmě svůj odpor mají, ovšem je tak nízký, že jej můžeme pro naše konstrukce zanedbat, a taky se to tak dělá. Pokud bychom ale přemýšleli nad vedením na velké vzdálenosti, musíme samozřejmě odpor vodičů započítat, protože už z jednotek ohmů naroste na vyšší hodnoty. Pro nás v elektronice jsou tyto hodnoty zanedbatelné, protože většina vodičů bude mít několik málo centimetrů. Ale kdyby vás napadlo, že zapojíte čidlo přes desetimetrový vodič, už se může snadno stát, že úbytek bude tak velký, že s ním je potřeba počítat.

Změřením ověřeno? Je to tak, jak jsem psal? Tak to se mi ulevilo. Pokud je tedy napětí na obou rezistorech stejné, jaké to bude s proudem? Při odpovědi můžete použít dva postupy. Buď změříte, nebo se zamyslíte. Já doporučuju zkombinovat obojí, tedy buď vymyslet a měřením ověřit, nebo změřit a úvahou podepřít naměřené.

⌀ https://eknh.cz/delpro

Každopádně úvaha by se měla brát zhruba tudy: Když je na rezistoru Rn napětí U, tak víme, že skrz něj poteče proud In = U / Rn. Nějaký proud teče celým obvodem. Teď je otázka, jestli poteče oběma rezistory stejný proud, jako celým obvodem, nebo jestli se tam někde nahoře mezi body B a C rozdělí na dva, a pak zase spojí. Voda to tak dělá… Dělá to i proud?

Dělá! Odborně se tomu říká **první Kirchhofův zákon** a zní tak, že součet proudů do uzlu vstupujících je stejný jako součet proudů z uzlu vystupujících. Uzel zní hrozně odborně, ale pro nás je to prostě to místo vodivého spojení, tedy bod mezi body B a C, kam přichází proud z baterie a kde se dělí do dvou větví. Vstupuje do něj proud z baterie (I), a vystupují z něj dva proudy I1 (do rezistoru R1) a I2 (do rezistoru R2). Mělo by tedy platit, že I (vstupující) = I1 + I2 (součet vystupujících).

Stejná situace se odehrává mezi body D a E, v místě, kde se obě větve opět spojují do jedné. Tam zase proudy I1 a I2 vstupují, vystupuje proud I.

Teď to změříme. V bodě A a F bychom měli naměřit hodnotu proudu I (v obou stejnou), v bodech B a D proud I1, v bodech C a E proud I2. Platí, že proud, který do rezistoru vstupuje, je stejný jako proud, který z něj vystupuje.

Pojďme zase na začátek úvah. Víme, jaký proud teče jakým rezistorem. Rezistor R1 pouští I1 = U / R1, rezistorem R2 prochází I2 = U / R2. Celkový proud I by tedy měl být I1 + I2. A ono to tak je. Pro napětí 5 V bude proud I1 a I2 roven 0,5 mA a celkový proud 1 mA. A pokud je celkový proud 1 mA, tak vychází, že celkový odpor bude 5 / 0,001 = 5000 ohmů, tedy 5k. Tedy poloviční.

Odvodíme si vzorec. Rx = U / I, kde I je celkový protékající proud. Ten je, jak víme, součtem obou proudů v obou větvích, tedy I1 + I2. Po dosazení:

$I = U / R1 + U / R2$

$Rx = U / (U / R1 + U / R2)$

$Rx = (R1 \times R2) / (R1 + R2)$

Nebo též $1 / Rx = 1 / R1 + 1 / R2$

Tedy **převrácená hodnota celkového odporu je rovna součtu převrácených hodnot jednotlivých odporů**.

Dalo by se říct, že paralelní zapojení funguje jako „dělička proudu". Při různých hodnotách dílčích odporů se dělí v různých poměrech, při stejných pak rovnoměrně 1:1. Sice se říká, že „proud vždy teče cestou nejmenšího odporu", ale není to tak úplně pravda. Správně bychom měli říct, že „oč je odpor menší, o to víc proudu jím protéká".

Všimněte si, že **u sériového zapojení je výsledný odpor vždy větší než největší z dílčích, při paralelním je celkový odpor vždy menší než nejmenší z nich**.

5.5 Kirchhof 2

Zmínil jsem **první Kirchhofův zákon**. Je snad nějaký další? Samozřejmě že je, a nepřekvapivě se nazývá **druhý Kirchhofův zákon**. Ten říká, že součet úbytků napětí na (sériově zapojených) spotřebičích je roven součtu napětí (sériově zapojených) zdrojů. Čili řečeno polopaticky: pokud zdroje zapojíte tak, že budou dávat 5 voltů a připojíte k nim iks spotřebičů sériově, tak ty se mezi sebou budou muset o těch pět voltů podělit. A to tak, že celkový součet napětí pro každý z nich bude přesně těch pět voltů, nikam se nic neztratí ani nikde nic nevznikne.

Mimochodem, oba zákony se jmenují podle německého vědce Gustava Kirchhoffa, profesora fyziky, který mimo těchto zákonů objevil například záření černého tělesa.

Mimochodem, jak je to se zdroji a jejich spojováním?

5.6 Baterie sériově – paralelně

To je dobrá otázka, a bude se to hodit. Už jsme si ukázali, že dva monočlánky 1,5 V, zapojené sériově, tedy „za sebou“, dávají dohromady 3 V. Tedy Ux = U1 + U2. Co když je zapojíme vedle sebe, paralelně? Výsledné napětí bude stejné, tedy 1,5 V, ale zdvojnásobí se velikost proudu, který je schopna baterie dodat.

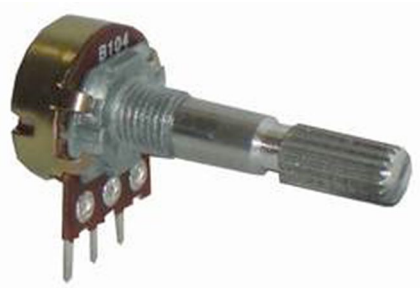
Baterie totiž, nepřekvapivě, nedokáže dodat nekonečný proud. Pokud budeme odebírat stále víc a víc proudu, tak začne klesat její napětí. Pokud bychom, teoreticky, zvedali odběr do nekonečna, klesne její napětí na nulu. Ovšem v praxi je potřeba dodržovat nějakou minimální hodnotu napětí, proto není možné brát libovolné množství proudu.

Co když jednu z baterií otočíme? U sériového zapojení to povede k tomu, že výsledkem bude nula voltů, u paralelního jsme právě stvořili automatický vybíječ baterií.

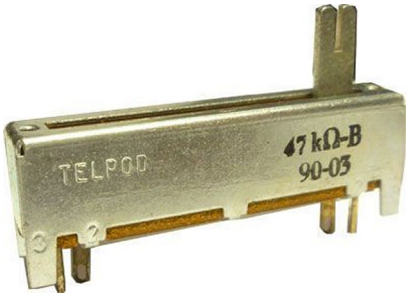
5.7 Potenciometr

Potrenciometr je vlastně proměnný dělič napětí v hmatatelné podobě. Součástka, kterou všichni známe z každodenní zkušenosti. Kdo z nás alespoň jednou nezesílil či nezeslabil rádio pomocí otočného knoflíku? Tak ta součástka uvnitř, která zařizuje, že vlevo to nehraje vůbec a vpravo nejvíc, se jmenuje právě potenciometr.

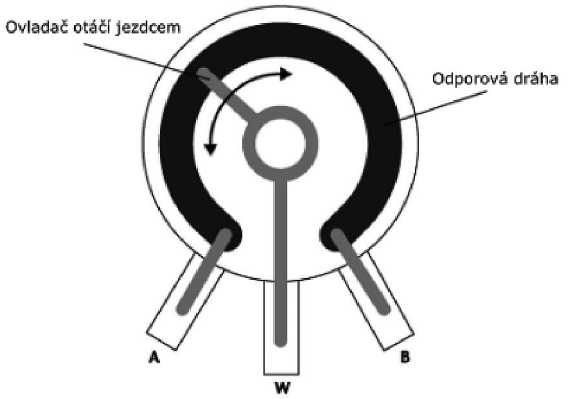
Nejčastější typ je otočný potenciometr:



Ale existují i jiné – třeba tahové



Potenciometr je vlastně rezistor určité velikosti, například 10k, ale proti obyčejnému rezistoru má vyvedený ještě třetí vývod, který je spojený právě s ovládacím mechanismem. Uvnitř se tento vývod („jezdec“) dotýká odporové dráhy. Tím, jak jej přibližujeme k jednomu či druhému krajnímu vývodu, se mění odpor mezi jezdcem a oběma vývody.



Dá se tedy říct, že jde vlastně o známé sériové zapojení dvou rezistorů, které má vyvedený prostřední vývod, a které má vždy celkový součet odporu stejný. Když je jezdec (W) otočený úplně doleva (k vývodu A), je odpor A-W (téměř) nulový, a odpor W-B je rovný celkovému odporu (A-B). Když je jezdec uprostřed, je odpor A-W poloviční proti odporu A-B (a stejný jako W-B). Když je jezdec ve čtvrtině dráhy, je odpor A-W roven jedné čtvrtině, W-B pak třem čtvrtinám celkového odporu...

Pozor – tak to je pouze u takzvaných lineárních potenciometrů. V audiotechnice, pro zesilování signálů a podobně se používá jiný typ, takzvaný logaritmický. Tam neplatí, že uprostřed je hodnota odporu poloviční.



Vlevo americký symbol, vpravo evropský.

Nám se potenciometr v číslicové technice moc nehodí, ale hodí se nám pro experimenty s elektřinou. Ukážeme si s ním jednoduchý pokus.

5.8 Úbytek napětí na LED

Zapojte si LED s rezistorem tak, aby LED svítila. Vyberte si třeba červenou LED a rezistor 220 ohmů. Napájecí napětí bude 5 V. Víme, že proud, protékající rezistorem, bude U / R = 5 / 220 = 0,022 A, tedy 22 mA. To je pro většinu LED proud na hranici bezpečného provozu. Pokud byste zapojili LED bez omezujícího odporu, poteče jí plný proud, který může zdroj dát, a LED spálíte! Pozor na to!

Je pravda, že ve mne teď hlodá zlomyslný červíček, který vám chce doporučit, abyste si to zkusili. Fakt, vzít zdroj, třeba nabíječku k telefonu, připojit k němu holou LED, a sledovat to ***prsk****, kterým se s vámi LEDka rozloučí. Abyste si to pamatovali. Ale myslím, že se vám to určitě brzy podaří neúmyslně…*

Změřte proud, který obvodem teče, a změřte, jaký je úbytek napětí na LED a na rezistoru. Pokud máte červenou LED, tak úbytek napětí na ní bude zhruba 1,8 voltů, na rezistoru pak ten zbytek do celkového napájecího napětí (při 5 V to bude zhruba těch 3,2 voltů).

Všimněte si, že jsem v předchozím textu nenapsal, jestli máte zapojit rezistor mezi katodu a zem, nebo mezi anodu a kladné napětí. Je v tom nějaký rozdíl? Neptejte se, zkuste si to, přinejhorším si zničíte diodu.

Máte? Teď už asi víte, že jste si diodu nezničili, a že je úplně jedno, jestli rezistor zapojíte před diodu, nebo za diodu...

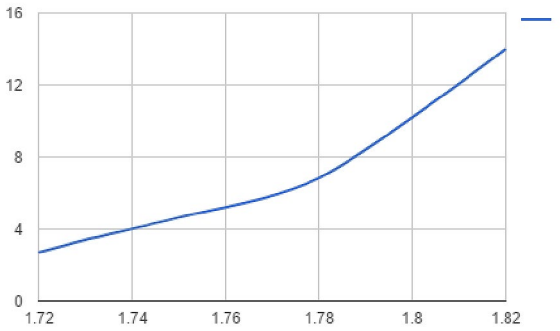
Zkuste zapojit druhou diodu do série s tou první. Tedy třeba mezi katodu a záporný pól připojte druhou LED tak, že anodu zapojíte na katodu předchozí diody, a katodu diody 2 zapojíte na zem. Svítí? Svítí. A co třetí dioda do série, svítí i ta? Za chviličku si řekneme víc...

Teď *do série* zapojíme potenciometr – třeba 1k, ale klidně i s větším odporem. Využijeme jeden z krajních vývodů a prostřední vývod, druhý krajní necháme nezapojený. Když budeme teď potenciometrem otáčet, uvidíme, že se LED rozsvěcí a zhasíná podle toho, jak otáčíme. K pevnému rezistoru 220 ohmů se přičítá proměnný odpor rezistoru, a podle natočení kolísá od 0 do Rmax.

⌀ https://eknh.cz/ledpot

Máte? Teď si zkuste změřit proud, který teče LEDkou, a napětí, jaké na ní je, a to pro různá nastavení jezdce. Můžete zapojit i výpočetní techniku, zanést si hodnoty do tabulky a udělat si graf, kde na vodorovné ose bude úbytek napětí na diodě, na svislé pak proud.

Pokud si takový graf nakreslíte, získáte tak něco, čemu se říká „voltampérová charakteristika“. Mně to vyšlo nějak takto:



(Vodorovně volty, svisle miliampéry)

Když si tedy spočítám, jaký je odpor LED (R = U / I), tak zjistím, že se mění podle velikosti proudu. Čím větší proud, tím menší odpor. Součástky, jako jsou diody (tedy i LED, což je „Light Emitting Diode“), se nazývají **nelineární**, protože jejich odpor není konstatntní a proud, který jimi prochází, není čistě závislý na napětí. Rezistory jsou **lineární** (pokud tedy zanedbáme změny odporu v důsledku zahřívání procházejícím proudem).

Když budeme proud, tekoucí skrz LED, snižovat, bude její odpor růst, naopak při zvyšování proudu odpor klesá a při překročení určité meze dojde k nevratnému poškození.

Všimněte si, že úbytek napětí na LED je víceméně konstantní a pohybuje se okolo 1,8 voltů. Toto napětí je dáno fyzikálními vlastnostmi – u červených LED se pohybuje pod hranicí dvou voltů,

u infračervených (např. v dálkových ovladačích TV) je okolo 1,5 V, u zelených nad 2 V, u modrých pak i přes 3 volty. Obecně se dá říct, že čím kratší vlnová délka světla, tím vyšší úbytek napětí.

U normálních polovodičových diod, tedy těch, co nesvítí, je tento úbytek zhruba 0,7 voltu (opět se mění podle proudu, který diodou teče). Toto napětí se může lišit podle technologie, u starších a dnes už méně běžných germaniových diod to bývá okolo 0,2 voltu, u takzvaných Schottkyho diod to bývá mezi 0,15 – 0,4 voltu. Úbytek napětí se v literatuře označuje V_F (z anglického Voltage Forward).

Vlnová délka (nm)	**Světlo**	**V_F při proudu 20 mA**	**Materiál**
940	Infračervené	1,5 V	GaAlAs / GaAs
635	Červené	2 V	GaAsP / GaP
570	Zelené	2 V	InGaAlP
430	Modré	3,8 V	SiC / GaN

5.9 Co jsou vlastně ty diody zač?

Dioda je dnes většinou polovodičová součástka (dříve i v podobě elektronky), jejíž nejvýraznější vlastností je, že vede proud pouze jedním směrem. Má dva vývody, anodu a katodu, a vede proud pouze ve směru od anody ke katodě.

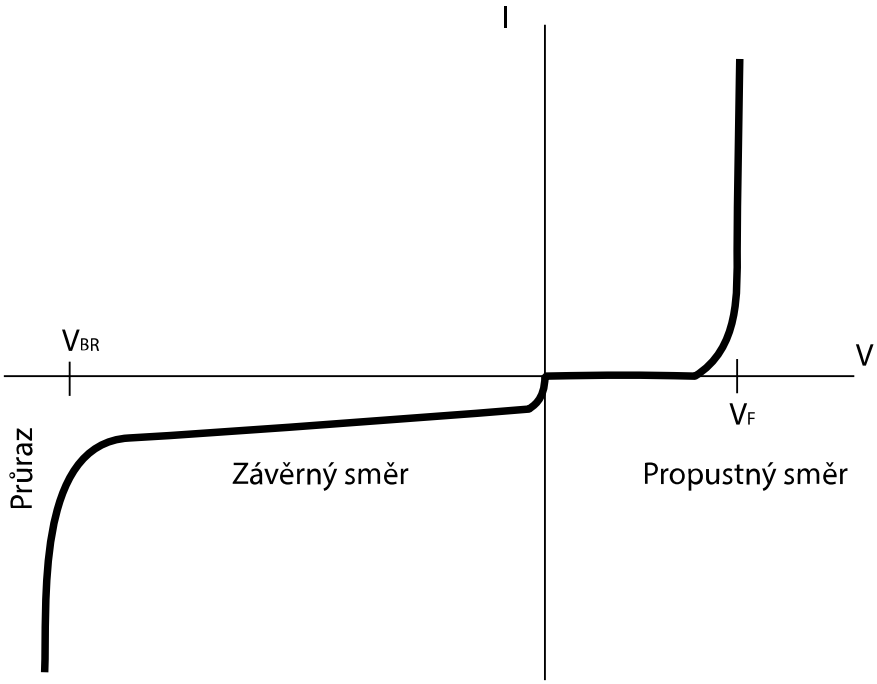
O diodách se ještě budeme bavit později, zde jen stručně…

Napětí určité výše je potřeba, aby se dioda vůbec „otevřela“, aby vedla proud. Takový je princip polovodičových přechodů, což si vysvětlíme později. Pokud je na diodě napětí menší, je zavřená – tedy její odpor je velmi vysoký a neteče skrz ni žádný proud, v žádném směru. Jakmile je na diodě napětí vyšší, dioda se skokově otevře (její odpor se velmi výrazně sníží) a proud může procházet – ovšem pouze v jednom směru.

Však si to sami zkuste. Nechte LED svítit, a pak ji otočte o 180 stupňů, prohoďte její vývody. Svítit nebude. V druhém směru („závěrném“) je dioda zavřená. Zase až do určité velikosti napětí, tentokrát zvaného **průrazné**. Pokud zapojíme vyšší napětí v závěrném směru, dioda se rychle zničí. Viz následující obrázek, který ukazuje závislost proudu na napětí, připojeném k (obecné) diodě.

V pravé části grafu je dioda zapojená v dopředném – průchozím směru. Proud teče velmi malý, a až u hranice V_F se dioda otvírá a vede proud velmi dobře, téměř s nulovým odporem.

V závěrném směru, tedy vlevo od svislé osy, prochází velmi malý proud. V bodě V_{BR} (Breaking Voltage, průrazné napětí) se dioda zničí, „prorazí“, a přestane fungovat. Dopředné napětí bývá nízké, okolo desetin až nízkých jednotek voltů, průrazné je o řády větší – podle typu třeba 50 voltů, ale i 1000.



Speciální variantou diody je Zenerova dioda. Ta se při určitém napětí v závěrném směru „prorazí“, ale vratně. Proto se používá ke stabilizaci či přepěťové ochraně. Jakmile napětí překročí danou mez, dioda se otevře, vede proud, a tím snižuje napětí.

⌀ https://eknh.cz/dioda

V číslicové technice se s diodami setkáme poměrně často. Používají se například k ochraně obvodů před přepólováním. Dříve se diody používaly v těch nejjednodušších přijímačích – krystalkách, což byly vděčné obvody pro amatéry, s nimiž jste pomocí několika součástek mohli poslouchat rozhlasové vysílání v pásmech AM. Diody se také používají k usměrňování střídavého napětí. V této knize se ale těmto oblastem nebudu věnovat šířeji a zájemce odkážu na jinou literaturu.

Ale myslím, že je právě teď vhodná chvíle udělat opět malou odbočku. Podíváme se, kde zjistit ta čísla, kterými tu šermuju, jako 0,7 V a 50 V. To jsem tak chytrý a Aštar Šeran mi ta čísla vnukl skrz alobalovou čepičku? Ale kdepak, podíval jsem se do datasheetu.

Do čeho že?

Do **datasheetu**!

5.10 Datasheet

Co je recept pro kuchaře, co jsou noty pro houslistu, to je datasheet pro elektronika. Datasheet se česky nazývá též „katalogový list". Dřív to platilo doslova – výrobce vydával katalogy součástek (my starší si pamatujeme katalog Tesly Eltos), kde každé součástce, kterou vyráběl, byla věnována jedna či více stran, kde bylo napsané všechno podstatné, co pro práci se součástkama potřebujete. Tedy – teoreticky. Někdy jste byli rádi, když platilo aspoň zapojení vývodů.

Dnes je situace díky internetu o mnoho jednodušší – k téměř libovolné součástce najdete její datasheet (klíčová slova: Google, název součástky, „datasheet"). Na stránkách výrobců bývají i různé vyhledávače a srovnávače, takže když například hledáte vhodnou součástku, můžete si z portfolia vybrat tu nejvhodnější na základě požadavků – třeba napájecího napětí, frekvence, pouzdra, funkce, ceny, …

Každopádně datasheet je klíčová věc, kterou budete potřebovat téměř vždy.

Zkuste si teď cvičně vyhledat pár datasheetů. Pro diodu 1N4004, pro tranzistor BC547, pro integrovaný obvod 74HCT04…

Tyto součástky vyrábí vícero výrobců, proto je můžete nalézt v různých provedeních a s různým obsahem. Některé speciální obvody vyrábí jen jeden výrobce, tam jste odkázáni pouze na jeho dokumentaci.

Co ale v datasheetu zjistíte vždy:

1. Přesné označení typu. Hlavně u integrovaných obvodů je to docela podstatné – 65C02 je něco jiného než 6502C, i když je obojí vlastně jen mírně vylepšené 6502. Součástí značení je i informace o pouzdru – 74HCT04N je funkčně shodný s obvodem 74HCT04D, ale jeden z nich je v pouzdru DIL (DIP), druhý v pouzdru SO. Pro nás to je podstatný rozdíl, protože zatímco ten první můžeme použít přímo v nepájivém kontaktním poli, ten druhý je určený pro povrchovou montáž pájením, má proto krátké nožičky a do kontaktního pole jej nedostaneme.

2. Informace o tom, který vývod u jakého pouzdra má jakou funkci.

3. Informace o napájecím napětí, o odběru a dalších charakteristikách obvodů a součástek.

4. Důležité informace o maximálních přípustných hodnotách – především napětí, proudu a teploty.

5. U integrovaných obvodů je popsaná jejich funkce, u těch nejsložitějších, jako jsou jednočipy a procesory, může být i informace o programování.

6. U diskrétních součástek bývají k dispozici různé grafy, například změna odporu v závislosti na teplotě apod.

7. Je popsáno chování za určitých standardních hodnot.

8. Pokud je více variant jednoho typu, je popsáno, čím se od sebe liší. Například tranzistor BC547 má tři varianty, A, B a C, které se liší maximálními přípustnými hodnotami napětí a proudů.

9. U složitých obvodů, například klopných obvodů, pamětí či některých digitálních senzorů, je popsán i komunikační protokol a průběhy signálů, ze kterých vyčteme třeba to, v jakém pořadí musí přijít řídicí signály, aby obvod fungoval správně.

Z dobře napsaného datasheetu zkrátka zjistíte vše, co o součástce potřebujete vědět. Třeba:

- Jaký je úbytek napětí na diodě 1N4007?
- Jaký je maximální proud kolektor-emitor u tranzistoru BC547B?
- Je u tranzistoru BC547 uprostřed báze, nebo kolektor?
- Kam se připojuje napájecí napětí u obvodu 74HCT00? Jak velké má být?

V datasheetech najdete i ta čísla, kterými jsem tu v minulých kapitolách šermoval. Pamatujete? 0,7 voltu na diodě, 10 mA proud LEDkou atd.

U některých dostatečně obecných součástek, jako jsou rezistory, keramické kondenzátory nebo obyčejné LED, často datasheet ani nenajdeme, protože nezjistíme typ součástky. Jde vlastně o takovou „bižuterii“. Přesto – zkuste si najít datasheet pro „červenou LED“.

6 Základní elektronické součástky

6 Základní elektronické součástky

Už jsme se seznámili s několika různými součástkami, tak je skoro načase udělat si takové drobné shrnutí, a při té příležitosti si ukážeme i pár součástek dalších.

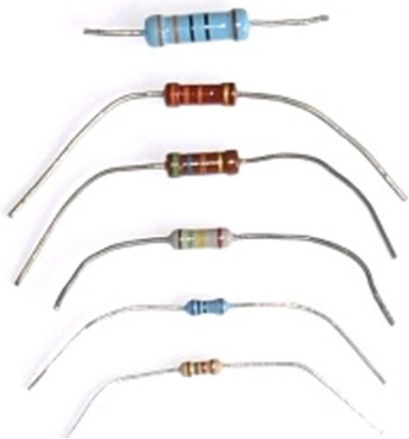
Součástky, s nimiž jsme se zatím podrobněji seznámili, se nazývají **diskrétní** – dělají nějakou jednu elementární funkci, například omezují procházející proud. Ale už jsme použili i jeden **integrovaný obvod** – to je komplexní součástka, sestavená z mnoha jednoduchých součástek, které jsou uzavřeny v jednom pouzdru. S integrovanými obvody se budeme setkávat v celém zbytku knihy, takže ještě bude hodně příležitostí si je popsat podrobněji.

Mezi diskrétní součástky, s nimiž jsme se už seznámili, patří rezistory, potenciometry a světlo emitující diody (LED). V prvním zapojení (blikač) jsme použili i kondenzátor, o kterém ještě nebyla řeč.

6.1 Rezistor

Rezistor spolu s kondenzátorem a cívkou tvoří základní trojici součástek. V číslicové technice budeme používat nejčastěji rezistory, kondenzátory jen okrajově a cívky ještě méně, ale je na místě se s nimi seznámit.

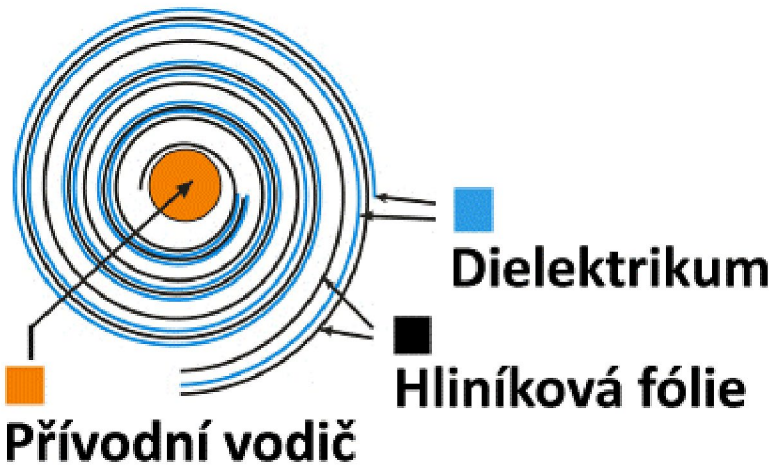
Rezistor je diskrétní součástka se dvěma vývody, jejíž zásadní vlastností je, že klade odpor procházejícímu proudu. Rezistory se vyrábějí v různých velikostech, pro naše účely postačí ty nejmenší. Pokud je třeba, aby rezistorem procházely vyšší proudy (a tím byly vyšší tepelné ztráty), musíte použít rezistory s velkým *ztrátovým výkonem*.



6.2 Kondenzátor

Kondenzátor je součástka, která umí uchovat elektrický náboj. Pokud připojíme kondenzátor k baterii, nabije se a náboj si udrží i po odpojení. Když pak připojíme k vývodům spotřebič, náboj do něj může přejít. Množství náboje, které kondenzátor dokáže pojmout, určuje jeho charakteristika, zvaná **kapacita**. Jednotkou kapacity je farad, v praxi se ale používají jednotky v řádech miliontin (mikrofarad) až biliontin (pikofarad). Dlouho platilo, že reálně dostupné kapacity jsou tisíce mikrofaradů (milifarady), ale v posledních letech s rozvojem nových materiálů začaly být dostupné i superkondenzátory s kapacitou v řádech stovek faradů, které dokáží napájet obvody po dlouhou dobu, vyrovnávat proudové odběry atd.

Kondenzátor v té nejjednodušší podobě tvoří dvě destičky, které jsou velmi blízko sebe, ale mezi nimi je izolant. V minulosti byly hojně používané např. svitkové kondenzátory, což byly vlastně dvě pásky z tenké hliníkové fólie, proložené izolační fólií a stočené do ruličky.

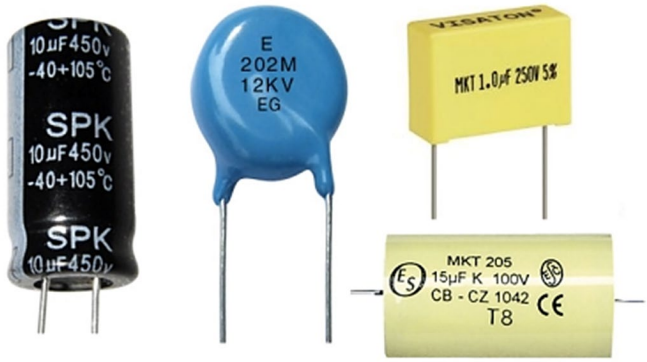


Dnes se nejčastěji v číslicové technice setkáme s kondenzátory keramickými a elektrolytickými. Keramické kondenzátory se hodí pro menší kapacity v řádech nano- a pikofaradů. Pro větší kapacity v mikrofaradech se používají kondenzátory elektrolytické, v nichž je v roli izolantu speciální roztok – elektrolyt. Tyto kondenzátory mají několik nectností. V první řadě jsou citlivé na přepólování, proto mají označený kladný a záporný vývod a je potřeba dodržet orientaci. Druhá nectnost těchto kondenzátorů je, že časem degradují, vysychají a ztrácejí svoje vlastnosti. Pokud jde o levnější kusy, mohou přestat fungovat už po několika letech. Elektrolytické kondenzátory bývají nejčastější příčinou toho, že i relativně nová elektronika přestane fungovat. Používají se hlavně v napájecích obvodech, a to proto, že dokážou udržet velký náboj a pokrýt okamžiky zvýšeného odběru. Jakmile ale takový kondenzátor zestárne, ztratí schopnost udržet náboj a zařízení

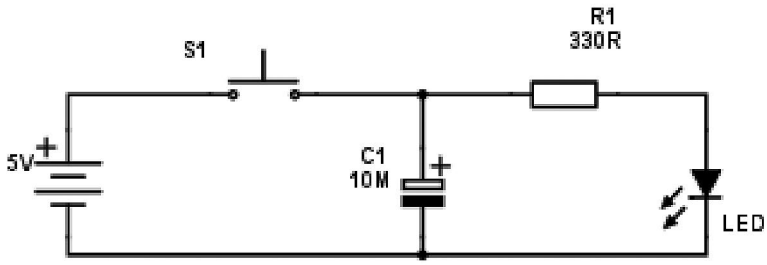
pak funguje s chybami, vypíná se nebo nefunguje vůbec. Leckdy stačí pouze tyto kondenzátory vyměnit a vše bude zase fungovat.

Když kondenzátor zapojíme sériově do obvodu, například s LED, kde prochází stejnosměrný proud, zjistíme, že proud prochází jen velmi krátkou dobu po zapnutí, pak přestane. Kondenzátor stejnosměrný proud nevede – ostatně je uvnitř izolant. Ovšem vlivem elektrické indukce vyvolá změna napětí (změna!) na jednom vývodu změnu náboje na vývodu druhém. Při rychlých změnách napětí na jednom pólu může tímto způsobem proud skrz kondenzátor procházet. Čím větší kapacita, tím pomalejší změny (tedy *nižší frekvence*) skrz kondenzátor projdou.

Proto se kondenzátor používá jako filtr. Tam, kde potřebujete například odfiltrovat příliš rychlé změny ze signálu (protože víte, že to je třeba rušení), připojíte paralelně kondenzátor. Rychlé změny projdou skrz něj (do zkratu).

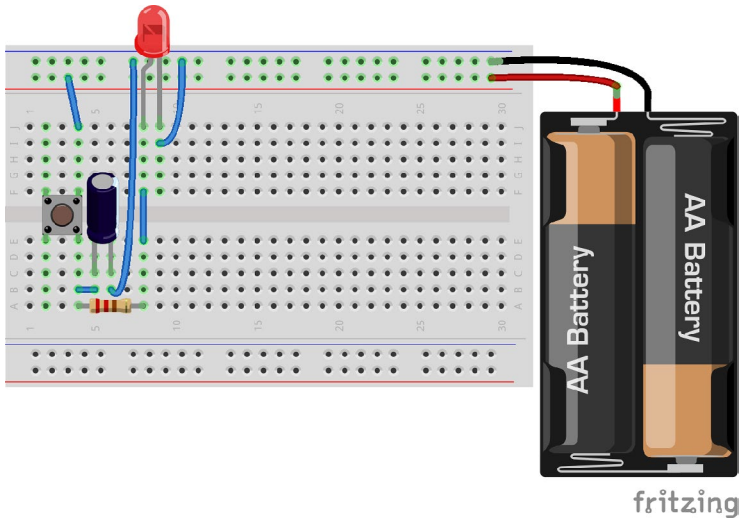


Minipokus: Zapojte si takovýto obvod:



⌀ https://eknh.cz/cap

Co se bude dít? Jakmile stisknete tlačítko, začne svítit dioda, a zároveň se začne nabíjet kondenzátor C1 (a část proudu poteče skrz něj, ovšem jak se bude kondenzátor nabíjet, bude tento proud klesat k nule). Když tlačítko po chvíli pustíte, přestane do LED téct proud z baterie, ale LED bude dál svítit. Bude ji pohánět náboj z kondenzátoru, který se pomalu vybíjí přes rezistor R1 do diody. Poznáme to podle toho, že dioda nezhasne hned, ale bude postupně zhasínat, zhasínat, až zhasne. To se stane ve chvíli, kdy napětí na kondenzátoru poklesne pod prahové napětí, nutné k rozsvícení a otevření diody.



Zkuste si pokus opakovat s různými kondenzátory a různými velikostmi rezistoru. Když budete měřit, jak dlouho to trvá, než LEDka přestane svítit, zjistíte, že tato hodnota je přímo úměrná kapacitě i odporu; čím větší je odpor a čím větší je kapacita, tím pomaleji se kondenzátor vybíjí a tím déle bude svítit. Samozřejmě, jakmile bude rezistor příliš velký, bude proud LEDkou tak malý, že LED bude svítit nepozorovatelně.

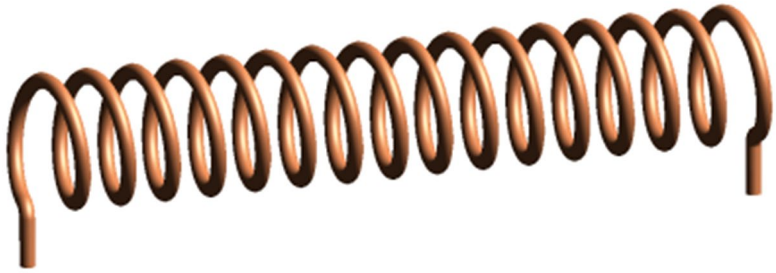
6.3 Cívka

Cívka je součástka, bez níž se neobejdete v radiotechnice a příbuzných oborech. Jde o vodič, navinutý okolo jádra (které může být kovové, z nějaké slitiny, nebo i takzvaně „vzduchové“). Její základní vlastností je impedance, což je v podstatě odpor, kladený střídavému proudu – obecněji jakýmkoli rychlejším změnám proudu.

Cívka při průchodu elektrického proudu vytváří kolem sebe magnetické pole (jedním z oblastí využití cívek jsou elektromagnety), které při svém vzniku nebo zániku vytváří v cívce proud.

Tento proud působí vždy proti směru změny. Jakmile zapnete zdroj, cívkou začne protékat proud, vzniká magnetické pole, to zase zpětně vytváří proud, ovšem opačného směru. Výsledkem je, že ze začátku teče ze zdroje proud malý (indukce v cívce jej snižuje) a postupně se zvýší až na plnou velikost. Když zdroj odpojíte, zase se změní magnetické pole (zmizí), a tato změna opět indukuje proud opačného směru, který postupně slábne.

Cívkou tedy bez problémů prochází stálý proud, ale proud, který se rychle mění, cívkou neprochází, protože rychlé změny magnetického pole pracují proti němu. Čím vyšší frekvence změn, tím větší odpor proudu cívka klade. Impedance cívky závisí na několika faktorech, ale především na počtu závitů, průměru závitů, materiálu, z něhož je vytvořeno jádro, ...



Cívky a kondenzátory se používají ke generování pravidelných kmitů, bez nichž se ve spoustě aplikací neobejdete. V číslicové technice ale nejsou moc používány, jejich nejčastější využití je v napájecím zdroji, kde slouží k tomu, aby propouštěly stálý proud, bez výkyvů a případných změn, popřípadě v tzv. spínaných zdrojích, kde fungují jako akumulátor pro elektrický náboj.

Z této „svaté trojice“ elektronických součástek budete nejčastěji používat rezistory, a v některých případech i kondenzátory. Cívkám se blíže věnovat nebudeme.

6.4 Transformátor

Transformátor je známá součástka, se kterou máme všichni každodenní zkušenost. Transformátor slouží k tomu, abychom určitou úroveň napětí – třeba 230 voltů v zásuvce – snížili na nižší hodnotu, nebo naopak na vyšší. Transformátor je fyzicky tvořen dvěma cívkami na společném jádru. Změna proudu v jedné z těchto cívek vyvolá změnu magnetického pole, a tato změna v druhé cívce vyvolá indukci proudu. Poměr počtu závitů obou cívek udává poměr, v němž se transformuje proud, a tedy i napětí. Pokud transformujeme napětí 230 V přes transformátor s poměrem závitů např. 1000:10000, získáme napětí 23 voltů, tedy desetinové, ovšem procházející proud bude desetinásobný.



Toho se využívá například při indukčním ohřevu – pod indukční deskou je cívka s mnoha závity, druhou cívku tvoří dno hrnce – které můžeme považovat za jeden závit. V něm se indukcí vytváří velmi nízké napětí, ovšem prochází jím obrovský proud, který ohřívá vodič – v našem případě dno.

6.4.1 DC měnič

Transformátor dokáže velmi efektivně měnit úroveň napětí, ale pouze tehdy, pokud na něj připojíme střídavé napětí. Stejnosměrné napětí transformátorem neprochází, protože nevyvolává změny magnetického pole. Pokud potřebujete změnit stejnosměrné napětí na jiné (třeba 5 voltů na 3,3, nebo na 12), použijte takzvané DC měniče (DC converters), což jsou zapojení, které buď pomocí kondenzátoru (nábojová pumpa – charge pump) nebo cívky (spínaný zdroj – switched-mode power supply) dokáží přeměnit stejnosměrné napětí na jiné. Buď na vyšší (a pak se jim říká step-up nebo boost convertor), nebo na nižší (step-down, popřípadě buck convertor).

⌀ https://eknh.cz/cpump

6.4.2 Stabilizátor

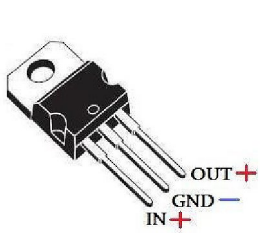
Pokud potřebujeme přesnou velikost napětí – a to v číslicové technice potřebujeme vždy! – musíme napětí, které přivádíme do obvodu, stabilizovat. V praxi se postupuje tak, že pomocí transformátoru snížíme síťové napětí 230 V střídavého proudu na nižší hodnotu, ovšem o něco vyšší, než je požadované. Například na 10 voltů. Toto napětí usměrníme pomocí diod, což jej

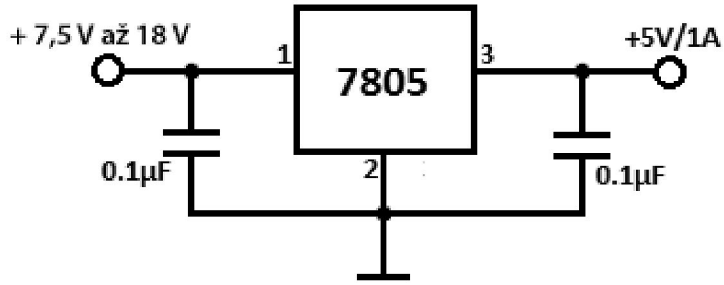
opět o něco sníží, a výsledné stejnosměrné napětí filtrujeme pomocí kondenzátorů. Máme na výstupu například 8 voltů, a musíme použít ještě speciální součástku, zvanou stabilizátor, která dokáže udržet předem danou hodnotu napětí. Nejčastější typ stabilizátoru je takový, který přebytečné napětí „spálí" – promění jej v teplo. Proto takové zdroje intenzivně hřejí a potřebují chlazení. Pro náročnější zařízení se používají stabilizátory s nízkým úbytkem, kterým stačí na vstupu napětí třeba jen o jeden volt vyšší než je požadované, a tím pracují efektivněji a nemají tak velké ztráty.

V napájecích obvodech u PC se používá kombinace transformátoru a spínaného zdroje. Dobré na této kombinaci je, že má relativně malé ztráty a vysokou efektivitu, a že dokáže dát velké a stabilní proudy ve stabilních úrovních napětí. Nevýhodou je, že potřebují stálý minimální odběr, a pokud je zapojíme „naprázdno", tak během několika sekund shoří, protože v jejich cívkách budou vznikat velké proudy, které nebudou mít kam odejít...

6.4.3 7805

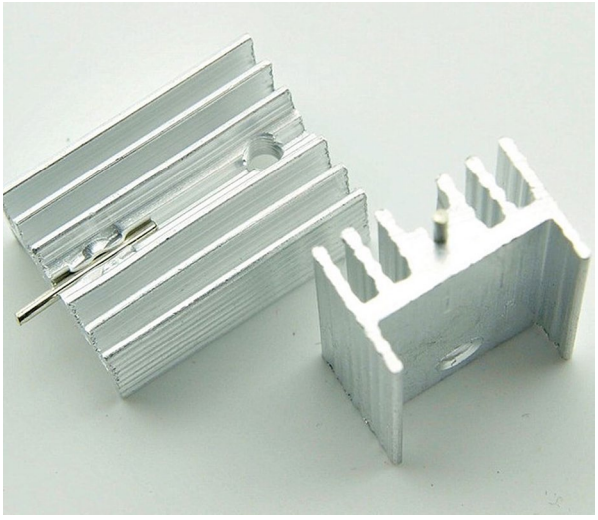
Označení 7805 nese součástka, kterou vyráběla už Tesla za socialismu. Jde o napěťový stabilizátor, který dokáže snížit napětí v rozsahu 7,5 V – 35 V na napětí 5 voltů. Jeho zapojení je velmi jednoduché: je to součástka se třemi vývody. Prostřední je společná zem, a dva krajní jsou vstup a výstup. Zapojuje se spolu se dvěma kondenzátory:





Někde se setkáte i s doporučením použít elektrolytické kondenzátory s řádově vyšší kapacitou, třeba i 10 µF. To může pomoci s vykrýváním odběrových špiček, ale zase má zdroj pomalejší náběh.

Takto zapojený stabilizátor dá dostatek proudu pro napájení číslicového obvodu. Pokud je potřeba větší proud, počítejte s tím, že se stabilizátor bude zahřívat, a přidejte k němu chladič. Chladič je tvarovaný hliníkový profil, který má velkou plochu a odvádí dobře teplo. Uprostřed je otvor pro šroub, stejný, jako má obvod 7805.



Obvod 7805 je levný, dobře dostupný a jednoduchý k použití. Bohužel má i své nevýhody, především vysokou vlastní spotřebu (ale k tomu se ještě dostaneme v kapitole o nízkopříkonových zařízeních). Pro stabilizaci napětí z baterií proto není příliš vhodný, ale pro zapojení, poháněné transformátorem (třeba starým napájecím adaptérem pro mobilní telefony nebo notebooky), jde o řešení použitelné.

7 Polovodiče

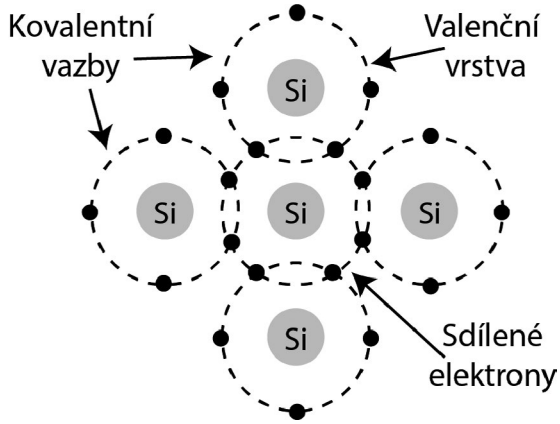
7 Polovodiče

Zatímco všechny dosavadní součástky se skládaly z vodičů, tedy látek, které vedou proud, tak teď přichází na scénu látky, které jej vedou „tak trochu“, nebo přesněji řečeno „za jistých podmínek“.

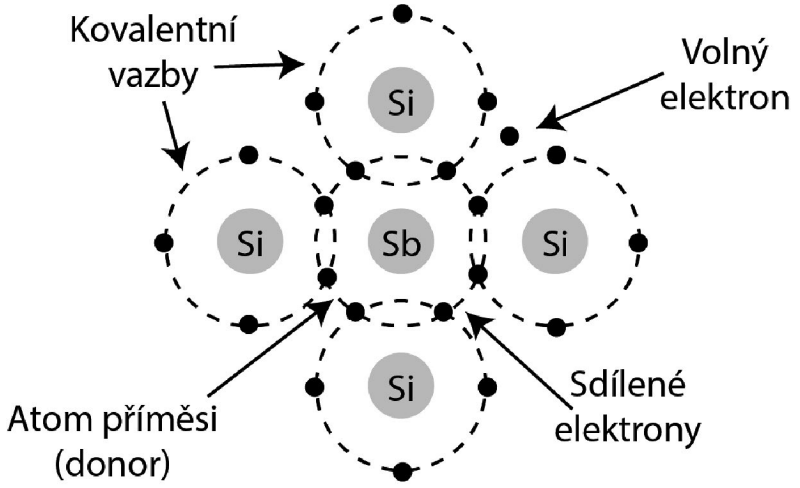
Polovodič si nepředstavujte, prosím, tak, že když máte vedle sebe vodič a polovodič, tak že polovodič vede míň proudu, kdepak. Polovodiče jsou látky, které vedou proud za jistých okolností, a ty jsou dané jednak typem polovodiče, a jednak elektrickými poměry, které na něm panují.

Polovodiče, třeba křemík nebo germanium, nemají moc volných elektronů. V jejich krystalové mřížce jsou atomy pravidelně uspořádány, každý atom sousedí se čtyřmi dalšími, a sdílí s nimi své čtyři valenční elektrony (tedy ty na nejvzdálenějších drahách). Což je poměrně stabilní uspořádání. Není v něm moc volných elektronů, takže látka proud nevede.

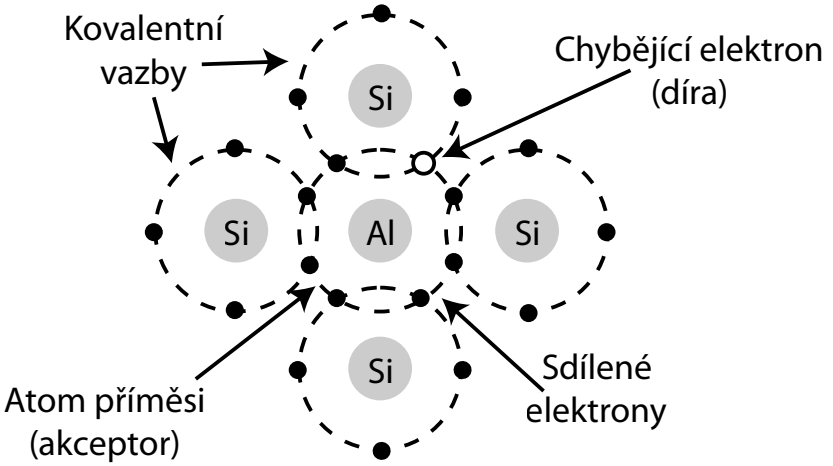
Stabilitu ovšem může narušit přidání energie zvenčí – kupříkladu světla či tepla. Dodaná energie excituje atomy, dojde k porušení těchto vazeb mezi atomy, a elektrony začnou opět nám známým způsobem putovat po látce. U původního atomu tak vznikne „díra“. Elektron nese záporný náboj, díra kladný. Čím víc energie dodáváme, tím víc podobných párů „elektron-díra“ vzniká. Většina z nich ale opět zanikne – nějaký volný elektron zaplní díru, ne nutně u stejného atomu.



Pokud ale do takové struktury zaneseme příměs, která má víc valenčních elektronů (například fosfor, arsen, antimon), stanou se atomy této látky součástí mřížky. Čtyři elektrony vytvoří páry se sousedními atomy, ale pátý elektron bude „volný“, a začne se potulovat hmotou. Vzniklá hmota bude vodivá, a protože vodivost zajišťují elektrony s negativním nábojem, říkáme, že se jedná o polovodič typu N.



Když do krystalu polovodiče naopak přidáme příměs takového prvku, který má pouze tři valenční elektrony (bór, hliník, galium, indium), bude v mřížce jeden elektron chybět, a vznikne tak (kladně nabitá) díra. Do ní může přeskočit elektron ze sousedního atomu – a tak se díra vlastně přesune o atom dál. Takto pozměněný polovodič bude rovněž vodivý, ovšem bude to polovodič typu P – protože jeho vodivost je založena na dírách, které mají kladný (= pozitivní) náboj.

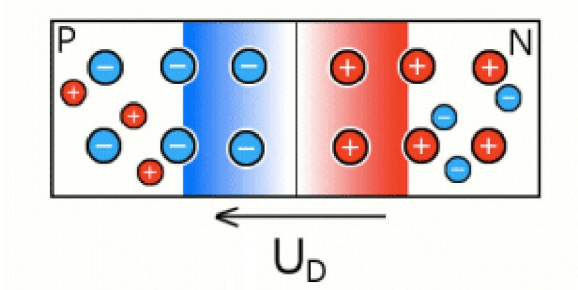


Zatím to je nuda. Polovodič vede občas, a můžeme do něj přidat příměsi, co z něj udělají buď P, nebo N. A dál?

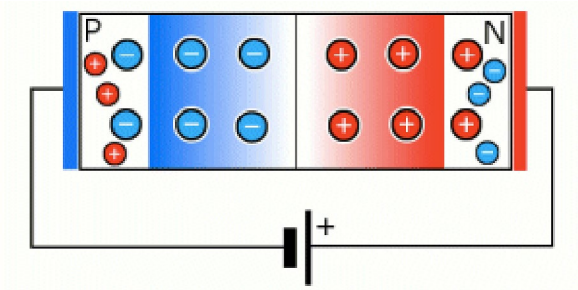
7.1 P-N přechod

No, teď si představte, že do cesty elektrickému proudu postavíte oba polovodiče za sebou, že vytvoříte takzvanou **oblast P-N přechodu**.

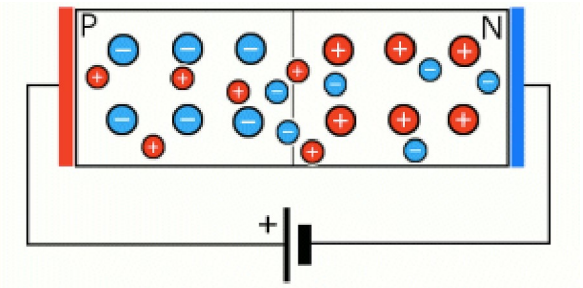
Co se vlastně stane, když k sobě připojíme polovodiče P a N? Víme, že polovodič N má nadbytek elektronů. Ty, co jsou blízko přechodu, přeskočí do polovodiče P, tam zaplní prázdné díry, a výsledek je, že v bezprostředním okolí přechodu vznikne vyprázdněná oblast bez volných nosičů. Ve zbytku polovodiče N tak po odchodu některých volných elektronů bude převaha kladného náboje, v polovodiči P po zaplnění některých děr bude převaha záporného náboje, a na přechodu tak vznikne difúzní napětí Ud (Jazykový koutek: difúze se píše s čárkou, nikoli s kroužkem.)



Teď na přechod přivedeme napětí – kladný pól na N, záporný na P. Co se stane? Kladný pól přitáhne k sobě volné elektrony, záporný jako by přitáhne volné díry (ve skutečnosti odtlačí blízké elektrony do vzdálených volných děr), a tím to hasne.



Když napětí přepólujete, dojde k opačnému jevu. Kladný pól, připojený k polovodiči P, odtlačí volné díry a přitáhne elektrony, které přešly přes přechod z polovodiče N. Záporné napětí, připojené k polovodiči N, bude naopak odtlačovat elektrony směrem k přechodu a do polovodiče P. Pokud bude napětí větší než difuzní, tak se vyprázdněná oblast opět zaplní. Tím se z nevodivého přechodu stane přechod vodivý.



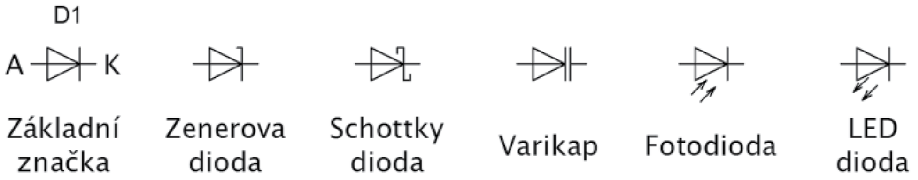
To je důležitý jev, a já s dovolením zdůrazním: **Přechodem P-N prochází proud pouze jedním směrem!**

Pokud si vzpomenete na předchozí kapitolu, tak zazněla zmínka o tom, že jedna ze součástek má přesně stejnou funkci…

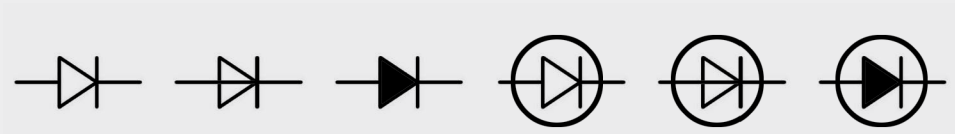
7.2 Dioda

Ano, polovodičová dioda je ve skutečnosti přesně toto – jsou to dva maličké kousky polovodičů P a N, spojených dohromady tak, že tvoří přechod. K těmto polovodičům jsou připojeny vývody – u P je to anoda, u N katoda.

Teď už je jasné, jak je možné, že diodou prochází proud jen jedním směrem. Ostatně – dobrou pomůckou je už samotný symbol diody, kde trojúhelník tvoří malou šipku. Ta ukazuje směr, kterým diodou teče proud – od anody ke katodě.



Mimochodem, když už je řeč o schematických značkách – víte, jaký je rozdíl mezi následujícími symboly?



Všechny symboly označují totéž: diodu. Jen se v různých dobách kreslily různě. Dřív se hodně dbalo na ten kroužek okolo symbolu (kroužek symbolizuje pouzdro součástky). Setkáte se ale při čtení schématů se všemi. Hlavní rozdíl je, že jen jeden z nich je podle platné normy (je to ten druhý zleva). Měli byste tedy používat ten normovaný. Co ale naděláte, když zrovna ten váš program, který ke kreslení použijete, bude mít symbol pro diodu jiný? Upřímně řečeno: neřešil bych to. Pokud bude poznat, že to je dioda, tak v pořádku. (Což je zase rada strýčka bastlíře.)

Když se místo klasického křemíku použijí jiné látky, například galiumarsenid (GaAs), bude energie, která se v diodě ztrácí při „zapadávání elektronů do děr" (laicky řečeno) vyzářena jako viditelné světlo. Vzniká tak LED, neboli světlo emitující dioda (Light Emitting Diode). Různé látky vydávají různé barvy světla, a tak máme k dispozici celou škálu barev LED. Nejstarší byly červené, pak přišly žluté a zelené, a když byla objevena látka, která vydává modré světlo, bylo možné sestrojit RGB LED a vytvářet libovolnou barvu světla. Existují i LED, které vydávají světlo ultrafialové a infračervené (infračervené LED se používají u dálkových ovladačů televizí apod.)

P-N přechod je samozřejmě stále citlivý na dodávání energie zvenčí – proto jsou polovodičové součástky většinou v neprůsvitném obalu, aby je neovlivňovalo světlo. I tak je ale může ovlivňovat teplo a měnit jejich charakteristiku. Se vzrůstající teplotou klesá jejich odpor a roste šum.

Někdy je ale citlivost na světlo vítaná – existují speciální diody, které mají průsvitné pouzdro, a u nich se právě citlivost na světlo a snižování odporu v závěrném směru používá k detekci světla. Proto se jim říká fotodiody.

Samotný P-N přechod, pokud je osvětlen, dokonce elektrické napětí vytváří přímo ze světla. Jestli tipujete, že to je princip solárních článků a panelů, jste na správné stopě.

Jak se říká – kdo nevěří, ať tam běží. Zkuste si to sami: Vezměte LED, připojte k jejím vývodům voltmetr, přepnutý na hodně nízký rozsah, a posviťte na ni jasným světlem…

Můžete si zkusit různobarevné LED, a uvidíte, pro kterou barvu se vám podaří získat největší napětí.

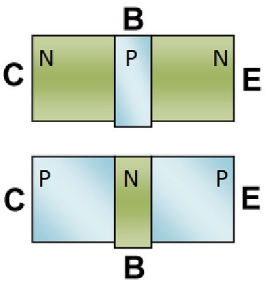
Diod existuje víc typů – už jsem zmínil Zenerovu diodu, která slouží ke stabilizaci napětí. Zajímavým typem diod jsou Schottkyho diody – zde je na místě jedné elektrody použit místo polovodiče kov. Taková dioda má nižší dopředné napětí a je rychlejší – rychleji se uzavírá a otevírá.

7.3 Tranzistor

Moje babička tohle slovo znala. Vždycky na mne volala: „Zeslab ten tranzistor, huláká to tu jak na lesy" nebo „Vypni ten tranzistor a pojď jíst"! Nemyslela tím elektronickou součástku, ale tranzistorové rádio. To byl velký hit 60. až 80. let – radiopřijímač, kde se několik tranzistorů staralo

o to, aby signál byl hezky hlasitý, a navíc to fungovalo na baterky, takže to bylo přenosné.

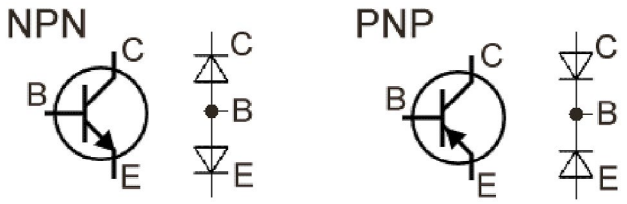
Tranzistor ale není radiopřijímač. Tranzistor je druhý typ polovodičové součástky, s nímž se seznámíme. Místo dvou elektrod, připojených na polovodiče P a N a jednoho P-N přechodu má hned tři elektrody a dva P-N přechody. Podle typu uspořádání rozlišujeme tranzistory NPN a PNP.



Elektrody se jmenují Kolektor (C), Báze (B) a Emitor (E). Tranzistor má tu vlastnost, **že proud, tekoucí mezi bází a emitorem, zároveň otvírá cestu mnohem větším proudům, které tečou mezi kolektorem a emitorem.** Proud mezi kolektorem a emitorem je úměrný proudu mezi bází a emitorem, a tak se tranzistor používá jako jednoduchý zesilovač. Nejdůležitější vlastnost tranzistoru se nazývá **proudový zesilovací činitel** (značí se h_{21e}, někdy taky *beta*, β nebo h_{FE}, a v anglických materiálech se setkáte s označením DC Current Gain) a udává, kolikrát je proud kolektorem vyšší než proud bází.

Ovšem audiotechnika, do níž zesilovače signálu patří, je sice krásná oblast elektroniky, ale mimo záběr této knihy, a proto se tu omezím pouze na konstatování, že tomu tak je. V číslicové technice, kde nezáleží ani tak na velikosti napětí a proudu, jako spíš na tom, jestli proud teče či neteče a jestli napětí je, nebo není, se tranzistory používají jiným způsobem – ne jako zesilovače, ale jako spínače.

Pokud připojíte na bázi kladné napětí vůči emitoru, tranzistor se otevře a vede proud z kolektoru do emitoru (u typu NPN; pro PNP platí totéž, ale obráceně – ovládá se záporným napětím a proud teče z emitoru do kolektoru). Když je na bázi napětí menší než 0,7 voltu (u křemíkových tranzistorů), tak je tranzistor zavřený a proud mezi kolektorem a emitorem neteče.



Ostatně je to vidět i na schematické značce – emitor je vždy označen šipkou, a ta udává směr proudu.

A opět mnemotechnické pravidlo: NPN (čtěte samozřejmě [en pé en]) – šipka ven!

Kromě těchto tranzistorů NPN a PNP, které jsou označovány jako **bipolární**, existují i tranzistory typu FET (tranzistory řízené polem – míněno elektrickým): JFET a MOSFET.

V číslicové technice se tranzistory používají nejčastěji jako budiče proudově náročných zařízení – relé, motorů, běžných žárovek atd. Číslicové integrované obvody totiž dávají na výstupu poměrně malý proud, který třeba pro LED stačí, ale pro sepnutí relé už je nedostatečný. Tranzistor nabízí jednoduchý způsob, jak tento výstup „posílit". Ale o tom se pobavíme později.

Já vím, relé je opravdu stará součástka a ne každý ji pamatuje. Proto vězte, že relé je „elektromagnetický spínač", kde je elektromagnet (cívka) a dva kovové kontakty. Když cívkou prochází proud, tvoří se okolo ní magnetické pole, a to přitáhne kontakty, které se tak spojí (nebo rozpojí, podle konstrukce). Tak lze nízkým napětím (5 V) spínat třeba síťových 230 voltů. Ale o relé ještě bude řeč.

7.4 Rozsvítíme prstem LED!

Pamatujete se na staré televize, které měly přepínání kanálů pomocí senzorů? Ne, teď nemyslím senzor jako že snímač, myslím senzor jako dotykové tlačítko. Vypadalo nějak takto:

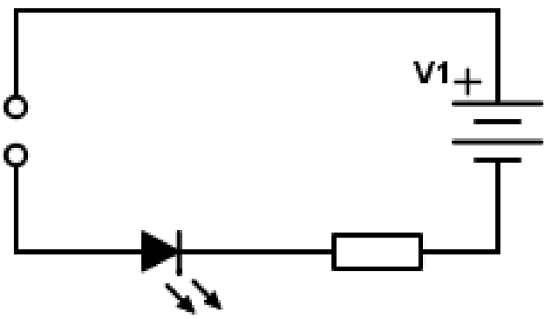


Senzor v tehdejším pojetí byly dva plíšky, oddělené nevodivou mezerou. Když jste se těchto dvou plíšků dotknuli prstem, spojili jste je (kůže je trochu vodivá), a řídicí elektronika to vyhodnotila jako stisknutí tlačítka.



Pojďme si to zkusit zapojit. Žádné složitosti, jen senzor, klidně nasimulovaný dvěma dráty vedle sebe, a dotykem rozsvítíte LED.

Co třeba takhle?



Ty dvě kolečka vlevo symbolizují dvě dotykové plošky, které budeme prstem spojovat. Vzhledem k tomu, že kůže má poměrně velký odpor, je potřeba, aby byly co nejblíž u sebe, třeba milimetr, dva…

Zkuste se dotknout – a co? Nesvítí, že? A když ty plošky spojíte třeba hřebíkem, tak to funguje. Stvořili jsme dotykový senzor pro kyborgy s kovovými prsty, ovšem pro lidi nic moc.

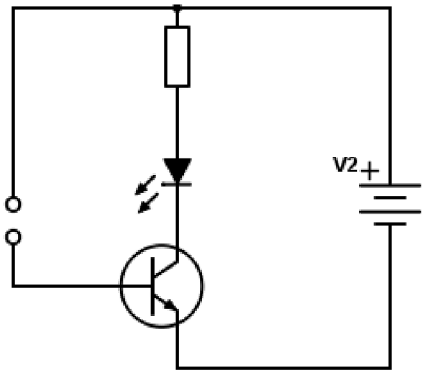
V čem je problém? Prozradím vám to: lidská kůže má fakt hodně velký odpor, na těch dvou milimetrech vzdálenosti klidně několik megaohmů. *Ostatně zkuste si to sami změřit!*

Pokud baterie dává 5 voltů, tak z Ohmova zákona vyplývá, že skrz prst (ne, to není jazykolam) poteče nějakých pár μA. Jenže to je na LED málo… Můžete si zkusit prst naslinit, odpor klesne třeba desetkrát, ale to stále není žádná výhra.

Kéž bychom měli součástku, která malým proudem dokáže sepnout velký, že?

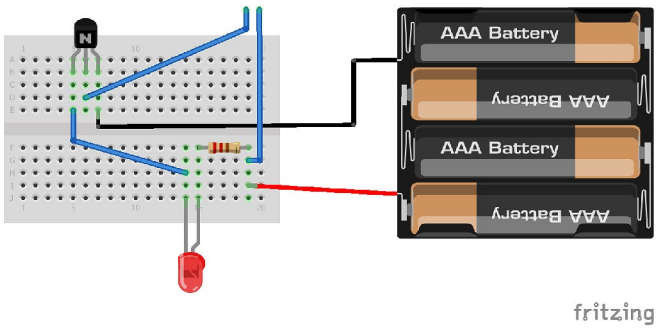
Nebudu vás napínat – právě jsem jinými slovy zopakoval to, co jste četli o několik stránek dřív: tranzistor malým proudem (mezi bází a emitorem) spíná velký proud (mezi kolektorem a emitorem).

Velký proud, to bude to, co poteče LEDkou, malý proud bude to, co teče senzorem. Co třeba takhle?



„Velký proud“ teče – pokud tedy teče! – z kladného pólu baterie přes rezistor a LED do kolektoru u tranzistoru, skrz tranzistor emitorem ven a do záporného pólu baterie. „Malý proud“ teče z kladného pólu téže baterie do senzoru, tam protéká vaším prstem, zpátky do druhého pólu senzoru, odtamtud do báze tranzistoru, a přes emitor ven a do baterie. Což samozřejmě platí pro situaci, kdy je prst přiložen. Když prst přiložen není, tak žádný malý proud neteče, a tím pádem neteče ani velký proud.

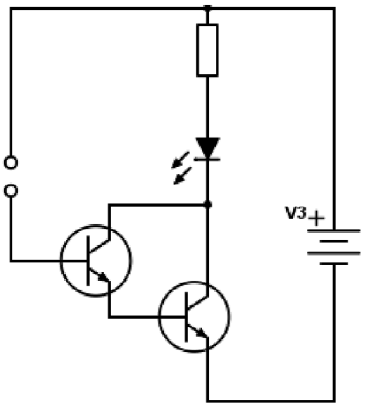
Jako tranzistor použijte třeba BC548C, ten má hodně velký proudový zesilovací činitel. Co je „hodně velký“? No, podle datasheetu je to nějakých 500 (čeho? Jablek? Voltů? Ohmů? Ničeho; zesilovací činitel je bezrozměrný, nemá jednotku) při proudu kolektorem 2 mA. Na takový proud stačí tedy 500 × menší proud bází – tedy 4 µA. To by mohlo fungovat, i když svit LED při dvou mA nebude nijak přehnaný…



7.4.1 Více světla!

Co když i ten „velký proud“ je pro nás stále malý? Co kdybychom chtěli do LED pouštět víc, třeba 10 mA. Nebo kdybychom chtěli zapojit žárovku, co si bere desítky až stovky miliampérů? Nebo když máme obzvlášť suché prsty a „malý proud“ je příliš malý?

Prozradím vám trik zvaný *Darlingtonovo zapojení*. V něm použijeme dva tranzistory. První udělá z malého proudu střední, druhý ze středního velký.



7.5 Tranzistor řízený polem (FET)

Tranzistory NPN a PNP jsou historicky první polovodičové tranzistory. Velmi hezky zesilují signál, ale mají jednu drobnou nevýhodu: zesilují **proud**. I když tranzistor zesílí proud třeba stonásobně, znamená to, že pokud má na výstupu dát třeba 100 mA, musí do vstupu téct 1 mA, což není zanedbatelný proud. Když chcete na výstupu spínat proud třeba 2 A, musel by takový tranzistor na vstupu odebírat 20 mA. Řešením by bylo mít tranzistor s *vyšším proudovým zesilovacím činitelem*, popřípadě zapojit dva tranzistory za sebou v Darlingtonově zapojení – první zesílí z 0,2 mA na 20 mA, druhý z 20 mA na 2 A. Problém je, že se tím zvýší spotřeba, a tím i ztráty.

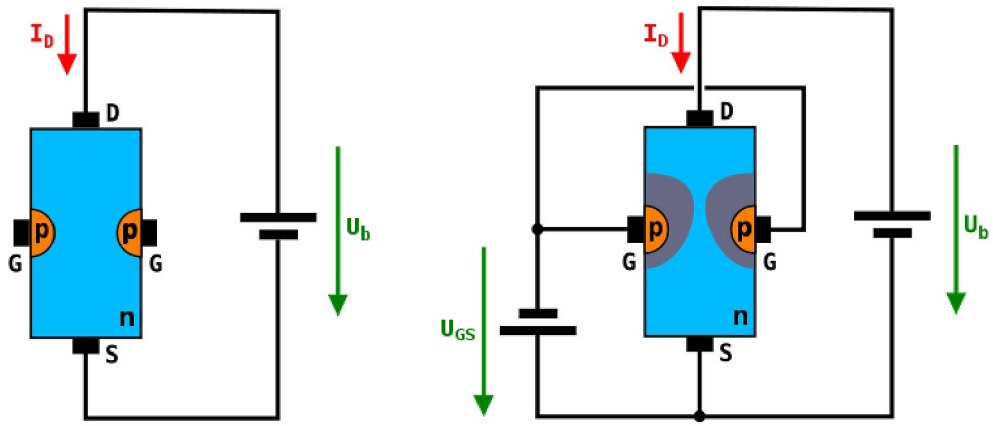
Existují ale i tranzistory, které na řídicí elektrodě nepotřebují téměř žádný proud a řídí se pouze napětím. Nazývají se *unipolární*, též *řízené polem*, v angličtině Field-Effect Transistor (FET). Nejjednodušší tranzistor FET je JFET (Junction FET).

Představte si dlouhý polovodič typu N („dlouhý“ na poměry elektronických součástek, v praxi řádově milimetry a menší), na jehož protilehlé konce jsou připojeny elektrody. Nazvěme je analogicky s emitorem (vysílač) a kolektorem (sběrač): Source (zdroj) a Drain (jímka). Polovodič

typu N má dostatek volných elektronů, takže vede proud mezi elektrodami S a D. Zatím to moc zajímavé není.

Teď si ale představte, že tento kanál omezíte. Někde v polovině vytvoříte oblast s vodivostí typu P, a k ní připojíte další elektrodu. Pokud bude na této elektrodě menší napětí proti elektrodě S (bude tedy „zápornější“), vytvoří se okolo přechodu ochuzená oblast (vzpomeňte si na diodu…), která nevede proud. Díky tomu se zmenší průřez kanálu, a tím vzroste jeho odpor pro proud mezi S a D. Tato elektroda tedy řídí tok proudu, a proto se jí říká řídicí, anglicky Gate (G).

Když bude na G kladnější napětí než na S, tranzistor bude otevřený.

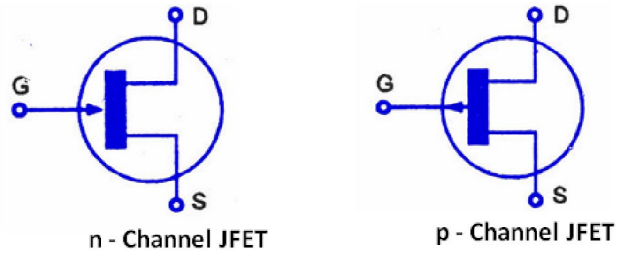


CC-BY-SA, autor Chautube, Wikimedia

Všimněte si, že se tranzistor řídí napětím v závěrném směru, tedy ve směru, kdy P-N přechodem mezi G a S neprochází proud. Proudy, tekoucí skrz Gate, jsou opravdu mizivé, v řádech nanoampérů.

Napětí, při kterém se tranzistor JFET zcela uzavře, bývá mezi - 0,5 V a - 10 V. Toto napětí velmi kolísá, nejen podle typu tranzistoru ale dokonce i u dvou tranzistorů stejného typu může být rozdílné.

Když zaměníte polovodiče P a N, stvoříte JFET typu P (dosud jsme si popisovali JFET-N). Jeho funkce bude stejná, jen logika obrácená: tranzistor se bude zavírat kladnějším napětím na elektrodě G.



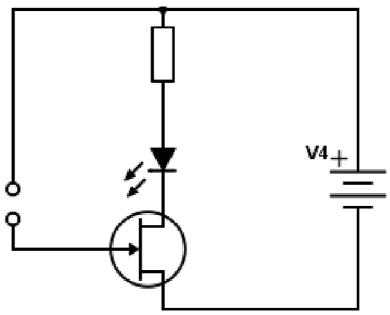
Všimněte si, že z popisu funkce vyplývá, že vlastně není rozdíl mezi elektrodami S a D. A ono to tak je: když tranzistor zapojíte obráceně (prohodíte S a D), tak *teoreticky* bude fungovat úplně stejně. U některých JFET to tak funguje doopravdy. V praxi ale mezi těmito elektrodami bývá rozdíl, například ve velikosti a tloušťce, ale některé typy mají opravdu S a D zaměnitelné.

7.6 Šoupejte nožkou…

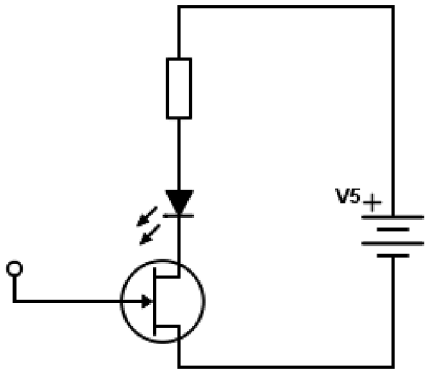
Sedíte? A máte koberec? A šoupete po něm mimoděk nohama? To nedělejte, to není dobře, tím se nabíjíte statickou elektřinou, ta putuje po vašem těle a nejen že můžete dostat ránu, když se dotknete něčeho uzemněného, ale můžete taky naprosto tiše a bez jakýchkoli vnějších projevů pouhým dotykem zničit drahou elektroniku.

O statické elektřině jsme si říkali, že má potenciál klidně v řádech kilovoltů, ale náboj není tak velký, aby dokázal vytvořit dostatečně nebezpečný proud. Když si to spojíte s těmi tranzistory: u bipolárních (NPN, PNP) potřebujete k jejich otevření proud; ty se statickou elektřinou neotevřou. JFET je řízený napětím, tam by to šlo…

Vezměte náš senzor s tranzistorem z předchozí kapitoly a tranzistor NPN nahraďte tranzistorem JFET – já použil třeba BF245. V datasheetu jsem si našel zapojení vývodů (G-S-D), a zapojil jsem vše analogicky: G místo báze, S místo emitoru, D místo kolektoru.



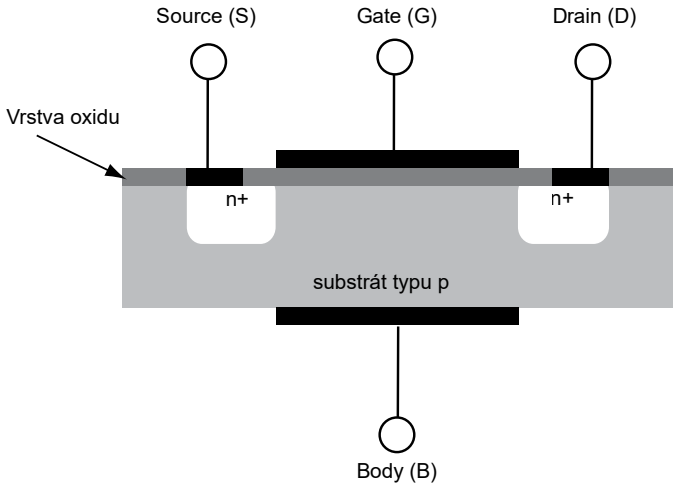
Všimněte si, že tentokrát není potřeba se prstem dotýkat. Stačí ho přiblížit dostatečně blízko. Dokonce si, věřte nebo ne, vystačíme s jediným pólem:



Klidně použijte vodič s izolací, ani nemusí být „holý“. LED bude změnou jasu reagovat i na pouhé přiblížení ruky.

7.7 MOSFET

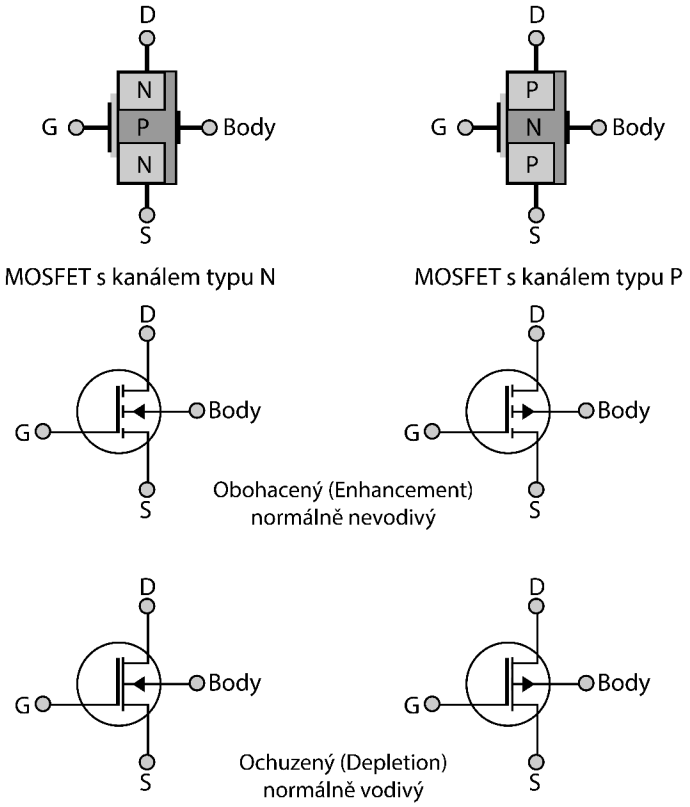
U tranzistoru JFET je řídicí elektroda součástí P-N přechodu a v propustném směru jí může téct proud, stejně jako skrz jakoukoli diodu. Pokud ale řídicí elektrodu izolujete tenkou vrstvou nevodiče, například oxidu křemičitého, můžete vytvořit další typ tranzistoru, řízeného polem.



Elektrody S a D jsou připojené k polovodičům typu N, které jsou zapuštěné v bloku polovodiče P (substrát, též tělo – body). Řídicí elektroda G je kovová a od polovodiče je izolovaná nevodivou vrstvou oxidu. Z tohoto uspořádání pochází i označení MOS: Metal-Oxid-Semiconductor.

Za normálního stavu mezi S a D neteče žádný proud. Brání tomu dvojice přechodů N-P-N mezi elektrodami a substrátem. Pokud ale na řídicí elektrodu G přivedete kladné napětí, „odtlačí" toto napětí volné díry v polovodiči typu P na druhou stranu, a pod ní vznikne úzký vodivý kanál, který se chová jako slabý polovodič typu N, a tranzistor začne vést proud.

Tranzistorů MOSFET je velké množství s nejrůznějšími typy struktur a vodivostí. Dva základní typy jsou NMOS (elektrody typu N, substrát typu P) a PMOS (elektrody P, substrát N). Oba typy mohou být vytvořené buď s indukovaným kanálem (enhanced – tento typ jsme si popsali), nebo se zabudovaným kanálem (depletion). U „depletion" typů je vytvořen vodivý kanál z výroby pomocí ochuzeného polovodiče stejného typu, tranzistor tedy za normálního stavu vede, a záporné řídicí napětí naopak vodivost snižuje.



Tranzistory typu MOS mají několik vynikajících vlastností – téměř nulový proud řídicí elektrodou, velké proudy mezi Source a Drain, nízké ztráty apod., ale mají jednu výraznou nevýhodu: vrstvička izolantu mezi elektrodou G a substrátem je opravdu velmi tenká, takže ji lze prorazit i poměrně nízkým napětím, třeba okolo 20 voltů. V reálných systémech bývají proto tyto elektrody chráněny rychlými diodami před průrazem. Ovšem největší nebezpečí představuje pro tyto tranzistory statická elektřina. Proto se při manipulaci s tranzistory MOS a dalšími součástkami, v nichž je tato technologie použita, doporučuje zvýšená opatrnost, měli byste s nimi manipulovat vždy důkladně uzemněni (používají se například vodivé náramky), a dokud to je možné, tak by měly mít tyto součástky všechny své vývody vodivě propojené (používá se buď vodivá pěna, nebo například alobal). Toto vodivé spojení by mělo být odstraněné až po montáži součástky do obvodu.

7.7.1 Co je to CMOS?

Patří to sem? Ano, trochu ano. CMOS je technologie, která kombinuje NMOS a PMOS tak, že se oba tranzistory doplňují (proto CMOS – Complementary MOS). V současné době je to nejpoužívanější technologie pro výrobu integrovaných obvodů: je dostatečně rychlá, s velmi nízkou spotřebou…

Pamětníci si vzpomenou, že měli v počítačích PC něco, čemu se říkalo taky CMOS, a byla tam baterie. To byl čip, který fungoval jako paměť konfigurace (jak velký disk je použitý, jakou má strukturu, kolik paměti je nainstalováno). Byl vyráběn právě technologií CMOS a díky tomu byla jeho spotřeba velmi nízká a pokryla ji po mnoho let baterie. Když se pak baterie vybila, začala „cé moska“ zapomínat.

A možná znáte zkratku CMOS z recenzí různých fotoaparátů: „Typ snímacího čipu: CMOS“. Ano, jde zase o stejnou technologii. Vzpomeňte si, jak jste svítili na LED: polovodičový přechod je citlivý na světlo. A když poskládáte strukturu MOS tranzistorů tak, že je z nich matice, na kterou může světlo dopadat, bude vám je dopadající světlo zavírat a otvírat…

7.8 A to je všechno s polovodiči?

Ani náhodou. Dioda a tranzistor jsou nejjednodušší polovodičové součástky s jedním, resp. dvěma P-N přechody. Existují i součástky s více přechody, jako tyristory, triaky a diaky, používané především pro spínání velkých proudů a regulaci výkonu, ale těmi se zabývat nebudeme.

Naopak se budeme zabývat jinými polovodičovými součástkami, totiž integrovanými obvody. Integrované obvody pomocí sloučení mnoha tranzistorů, diod a rezistorů na jedné křemíkové destičce nabízejí velmi složité funkce a komplexní obvody.

Integrované obvody dělíme do dvou základních skupin: analogové a digitální. Zjednodušeně se dá říct, že analogové obvody nějak pracují s proudy či napětím – zesilují, zpracovávají apod., zatímco u digitálních se pracuje pouze s logickými signály, nejčastěji definovanými jako: je napětí/není napětí. Zapnuto/vypnuto. Mezi nimi jsou obvody hybridní, které kombinují oba světy, jako například analogově-digitální a digitálně-analogové převodníky, časovače, Hallovy sondy s digitálním výstupem a podobně.

My budeme používat převážně digitální obvody v pouzdrech DIL. Což mě přivádí k nápadu, že je vhodná chvíle pobavit se o tom, jak jsou součástky vlastně „zabalené“ a jak se s nimi pracuje.

8 Pouzdra elektronických součástek

8 Pouzdra elektronických součástek

Rezistory se nejčastěji vyskytují v podobě malých válečků s axiálně (tj. v ose) umístěnými vývody. Vypadají jako „oteklý drát“, jak se říká. Hodnota bývá značena buď barevnými proužky, nebo (na větších rezistorech) napsána číslem.

Kondenzátory jsou nejčastěji menší či větší válečky s oběma vývody na jednom konci (u elektrolytických kondenzátorů), popřípadě vypadají jako velké rezistory (svitkové) nebo jako malé polštářky. U elektrolytických kondenzátorů je dost místa na označení hodnoty a maximálního napětí, u menších keramických bývá trojčíslí, kde první dvě číslice znamenají hodnotu a poslední počet nul, které je potřeba dopsat, abychom dostali kapacitu v pikofaradech. Příklad:

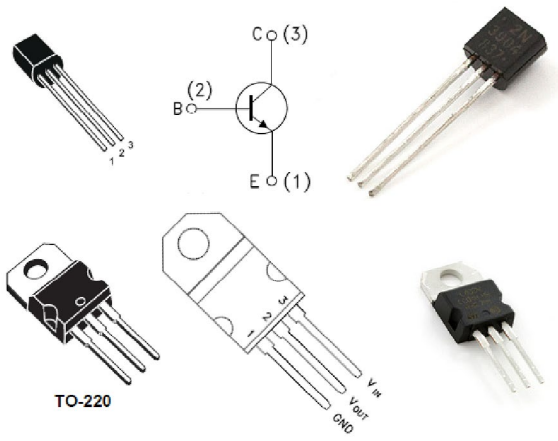
330 = 33, žádná nula, tedy 33 pF

104 = 10 + 0000, tedy 100 000 pF = 100 nF

472 = 47 + 00, tedy 4 700 pF = 4,7 nF

Diody jsou podobné rezistorům – mívají pouzdro, na kterém je napsán typ diody, a u jednoho vývodu barevný pruh. Ten označuje katodu.

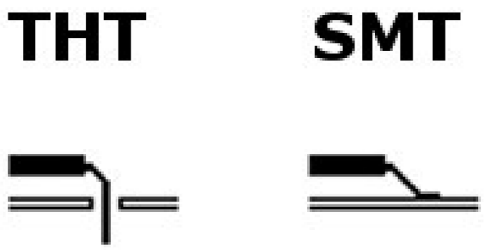
Tranzistory bývají nejčastěji v pouzdrech, které se označují TO-92 nebo TO-220. TO-220 jsou pouzdra pro výkonové tranzistory, k nimž se připojuje chladič.



To, který vývod je báze, který kolektor a který emitor, se u jednotlivých typů liší, proto je vždy důležité podívat se do datasheetu.

8.1 Co je to SMT a THT

Kdysi dávno se obvody montovaly ze součástek s drátovými vývody, které se k sobě letovaly. Později přišly takzvané plošné spoje – to je deska, na níž je nanesena vodivá vrstva, nejčastěji mědi, a pomocí ní jsou vytvořené vodivé spoje dle potřeby. Pro každou součástku bylo vyhrazené místo, kam byla připájena, a nebylo potřeba propojovat vývody ručně s rizikem toho, že „se na něco zapomene". Později, s rozvojem integrace, přišly vícevrstvé plošné spoje. Pak se ukázalo, že vrtání děr je drahé, proto se přešlo k součástkám v pouzdrech, které jsou přizpůsobené pro montáž na povrchu, tedy není nutné prostrkovat přívody vrtanými otvory. Této technologii se říká Surface Mounted Technology – SMT, tedy „technologie povrchové montáže". Pouzdra, která jsou vhodná pro tuto montáž, se označují SMD (Surface Mounting Design). Starší způsob, s otvory a vývody na prostrčení, dostal název THT – Through Hole Technology.



Pro začátečníky je vhodnější technologie THT. Má to hned několik důvodů. Pouzdra pro THT mají dlouhé vývody s většími roztečemi (nejčastěji v rastru 2,54 mm = 0,1 palce) a dobře se s nimi pracuje. Dají se použít v nepájivých polích, dají se i dobře pájet v ruce… Součástky v SMD pouzdrech mají vývody krátké, někdy ani vývody nemají a mají jen pájecí plošky, a tak je musíte (skoro) vždy pájet. Navíc mají menší mezery mezi vývody, a tak jejich ruční pájení vyžaduje lepší vybavení a hlavně cvik.

V SMD pouzdrech se vyrábějí snad všechny diskrétní součástky – rezistory, kondenzátory, diody, tranzistory, LED – i integrované obvody. Některé integrované obvody se vyrábějí pouze v SMD pouzdrech. Platí to zejména pro velmi složité obvody, jako jsou moderní procesory, programovatelná logická pole, miniaturní senzory a další podobné. Výrobci dnes už málokdy navrhují moderní obvody pro THT, a ani je v takových pouzdrech nenabízí. Výjimkou jsou snad jen některé sériové paměti.

8.2 DIP, DIL

U integrovaných obvodů, s nimiž budeme pracovat, budeme nejčastěji používat pouzdra typu DIL či DIP. Jaký je mezi nimi rozdíl?

DIL znamená Dual In-Line. Tedy že vývody jsou uspořádány do dvou (dual) řad (line). Mezi sebou mají rozestup 2,54 mm, řady jsou od sebe vzdáleny trojnásobek či šestinásobek této hodnoty.

DIP znamená Dual In-Line Package. Tedy naprosto totéž jako DIL, jen je tam přidáno slůvko *Package*, neboli pouzdro. Můžete se setkat i s označením PDIP a CDIP. PDIP znamená Plastic DIP, CDIP je Ceramic DIP. Dnes se většinou používají první, tedy plastová pouzdra, většinou černé barvy. CDIP, keramická pouzdra, bývala kdysi populární, nejčastěji měla bílou barvu, ale dnes už se moc nepoužívají – jsou křehká a někteří uživatelé hovoří o tom, že po mnoha letech přestávají součástky fungovat, protože se pouzdra občas poruší.

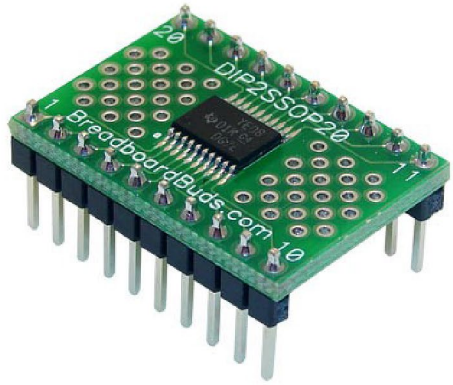
Číslovka za označením znamená počet vývodů. DIP14, DIP16, DIP24, DIP28, DIP32 a DIP40 (popřípadě „DILxx“) udávají, že součástka je v pouzdře se 14, 16 atd. vývody.

Pokud se dostanete k pájení, velmi rychle zjistíte, že integrované obvody jsou náchylné na přehřátí. Proto je dobré dávat je do takzvaných patic (či objímek) Sice to někdy sníží spolehlivost, ale zjednoduší to výrazně výměnu poškozených součástek. Integrované obvody s mnoha vývody se totiž snadno zapájí, ale podstatně hůř vypájí.

8.3 Co s těmi ostatními?

Jak jsem už psal – u moderních součástek bývá časté, že je výrobce nenabízí v jiném pouzdru, než pro povrchovou montáž. Ale naštěstí spousta dalších firem nabízí takzvané „breakout moduly“ – což jsou v podstatě malé destičky, na nichž je připájena daná součástka, a mají vyvedené vývody v roztečích, které jsou vhodné pro nepájivá pole.

Mnohé součástky, především senzory, jsou díky tomu dostupné i pro domácí kutily.



8.3.1 Praktické tipy

Pro svoje pokusy budete potřebovat mít sadu základních součástek, nebudete pokaždé běhat do obchodu, když budete potřebovat tři rezistory.

Už jsem to psal: Udělejte si zásobu součástek. Pro číslicovou techniku bych doporučil mít základní sadu – sepsal jsem ji na konci knihy, v příloze „nákupní seznam". S takovou sadou součástek budete moct sestavit prototyp libovolného zařízení, nemusíte čekat, až dojdou speciální obvody, můžete vesele testovat, a až pošta doručí vybrané součástky, tak můžete postavit konečně hotové zařízení.

9 Blikač s Arduinem

9 Blikač s Arduinem

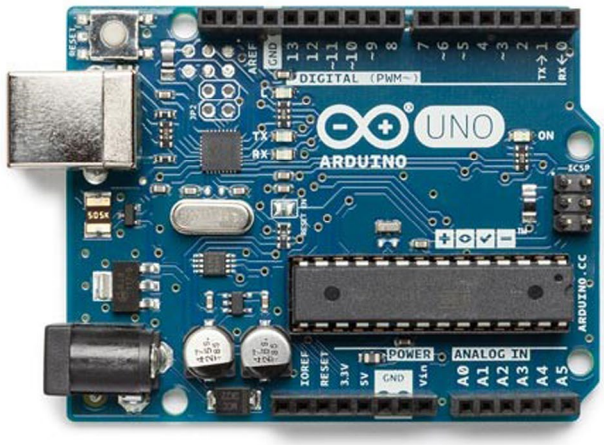
Dobře, pojďme si ukázat o něco jednodušší způsob. Arduino jistě znáte…

Neznáte? Tak to rychle napravíme, protože Arduino je fajn pomůcka i nástroj, se kterým ušetříte spoustu součástek, a navíc si zaprogramujete. Ne, vážně, bez znalosti programování v nějakém jazyku, podobném C, Arduino nevyužijete. Na druhou stranu – pokud umíte programovat v čemkoli, co má podobnou syntax, tedy v Javě, JavaScriptu, C#, Lua a dalších jazycích, tak zvládnete i jazyk pro Arduino (říká se mu Wiring).

Knih a výukových materiálů k Arduinu je spousta, takže vás s podrobnostmi odkážu raději na ně (tip: arduino.cz má zajímavou knihu jako e-book), tady si řekneme jen pár základních věcí.

9.1 Když se řekne Arduino

tak „se“ má na mysli nejspíš Arduino Uno. Je to dnes nejrozšířenější typ, a díky tomu, že je open-source, tak mohlo vzniknout nepřeberně klonů, které jsou lépe vybavené, levnější, popřípadě obojí.



Originální Arduino obsahuje *jednočip* ATmega328 – to je to velké černé vpravo dole s mnoha vývody. Tento obvod provádí program, který do něj nahrajete, a podle něj umí řídit jednotlivé výstupy (horní a dolní okraj). Dále obsahuje komunikační rozhraní pro sběrnici USB – kovový konektor vlevo nahoře a malý čtvercový obvod napravo od něj. Vlevo dole pak jsou součástky, které se starají o napájení toho všeho. A to je vše.

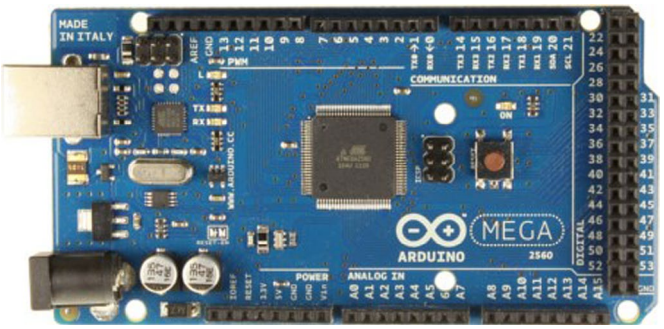
Neoriginální Arduino může vypadat třeba takhle:



Trénované oko vidí, že procesor je v jiném pouzdru (černý čtverec zhruba uprostřed), a že obvod pro řízení USB je jiný (obdélník místo čtverce).

Pokud Arduino ještě nemáte a přemýšlíte, jaké pořídit, tak mám jeden tip: **pořiďte si v českém obchodě originální**. Je sice dražší, ale máte mnohem vyšší naději, že bude v pořádku. A máte i záruční lhůtu na případnou opravu. Klon vás vyjde třeba na pětinovou cenu, nebo i nižší, ale může se vám stát, že budete pozorovat „podivné" chyby v chování – protože třeba výrobce něco nepřipájel, nebo osadil ne zcela perfektní obvody. Mně se podobné chování objevilo asi tak u dvou procent Arduin, ale nikdy nevíte…

Kromě základní verze Uno existuje i Arduino Mega s mnohem větším procesorem (a taky s více vývody):

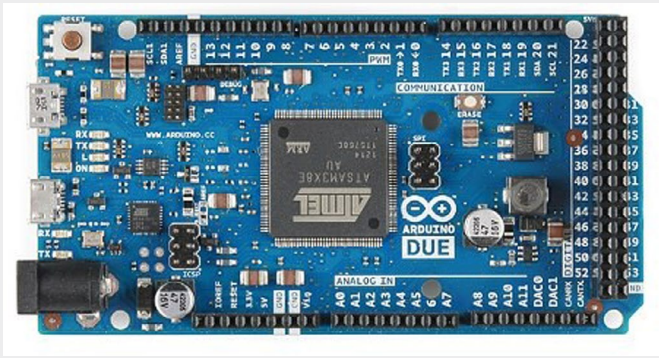


Vidíte, že zleva je to velmi podobné obyčejnému Arduinu, a vpravo je jaksi „navíc“ spousta dalších vývodů a konektorů.

A kromě těchto Arduin je i spousta dalších desek a kitů, které se jmenují všelijak – Seeduino, DCDuino, Funduino, … Společné mají to, že je lze programovat stejně jako základní Arduino, ale liší se ve schopnostech, někdy i v konektorech…

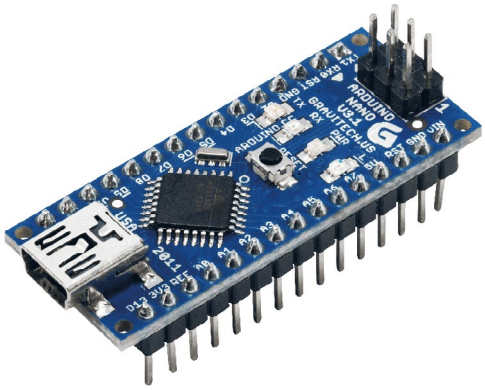
Existují i Arduina, která jsou osazena jiným procesorem, než je Atmel AVR – například procesorem z rodiny ARM:

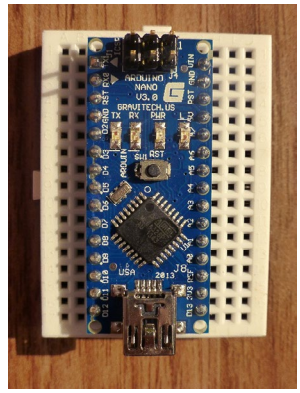




Tato Arduina mají sice stejné vývody, ale vyžadují jiné napětí – ne 5 voltů, ale 3,3 voltu. Později si vysvětlíme, co to znamená, teď si pamatujte: ***pro začátek tato Arduina nepoužívejte, použijte Uno!***

Zajímavá varianta Una je miniaturní Nano. Výhoda tohoto Arduina je, že ho lze zasunout do nepájivého kontaktního pole:

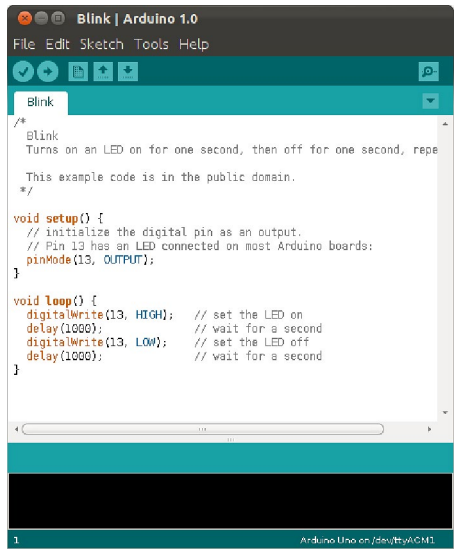




9.2 Programování Arduina

Aby Arduino k něčemu bylo, musí v něm být nějaký program. K programování se používá nástroj, který se trochu honosně jmenuje Arduino IDE – ve skutečnosti jde o textový editor, k němuž je přibalený kompiler pro procesory Atmel AVR a knihovny. Na druhou stranu stačí stáhnout si tento nástroj, propojit Arduino s počítačem přes kabel – a fungujete.

Arduino IDE si stáhnete ze stránek Arduino.cc pro všechny tři hlavní OS – Windows, Linux i macOS.



Udělejte to.

Ne, zcela vážně: **Vezměte Arduino Uno a propojte ho s počítačem. Hned teď.** Originální Arduino nebude vyžadovat žádné ovladače, respektive si je stáhne. Problém může být s neoriginálními Arduiny, ty vyžadují na macOS nebo na starších Windows (7, 8, 8.1) instalaci ovladačů.

Máte? **Stáhněte si Arduino IDE, rozbalte (nemusí se instalovat) a spusťte.**

Nejdřív jděte do menu Tools, česky Nástroje, tam vyberte podmenu Board – Vývojová deska, a v ní zvolte položku „Arduino / Genuino Uno". Tedy samozřejmě předpokládám, že máte Arduino Uno, nebo jeho klon. Druhý krok je opět v menu Tools – Nástroje, položka Port. Pokud máte Arduino připojené, měli byste jej vidět v nabídce. Na Windows bude mít označení COMxx (xx je nějaké číslo, klidně COM2, ale i COM37, je to opravdu libovolné). Na Linuxu to bude /dev/tty*něconěco*. Pozor, na Linuxu musí být uživatel, pod nímž je spuštěno Arduino IDE, členem skupiny „dialout". Na Macu to bude něco jako /dev/ttyusb*něco*.

V menu File je podmenu Examples (česky Soubor – Příklady), a tam naleznete adresář 01 Basics, a v něm příklad, nazvaný Blink. Klikněte na něj, otevře se takzvaný „sketch" – vlastně zdroják pro Arduino s příponou .ino

9.3 Blikání Arduinem

Vidíte kód, který je hodně jednoduchý, programátoři už určitě větří… Vidíte dvě funkce, setup() a loop(). První se provede po spuštění celého zařízení, a když doběhne, tak se donekonečna volá ta druhá. Nějak takhle – představte si, že tohle je funkce na pozadí:

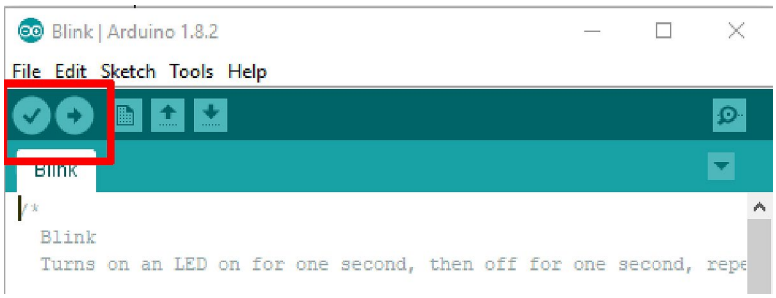
```
void main() {
    setup();
    while(1) loop();
}
```

Zpátky k příkladu blink. Ve funkci setup() je volaná nějaké funkce pinMode(), která říká, že nějaký výstup má být OUTPUT, a v druhé se pak střídá digitalWrite HIGH a digitalWrite LOW, proložené zpožděním (delay). (Na konci knihy, v přílohách, najdete stručný přehled základních funkcí Arduina).

```
void setup() {
    pinMode(LED_BUILTIN, OUTPUT);
}
```

```
void loop() {
    digitalWrite(LED_BUILTIN, HIGH);
    delay(1000);
    digitalWrite(LED_BUILTIN, LOW);
    delay(1000);
}
```

V horní liště, hned pod menu, je několik tlačítek. To úplně první vlevo zkontroluje, jestli jsou v kódu nějaké chyby. To druhé, vedle něj, s ikonkou šipky, spustí překlad, a po překladu se přeložený kód nahraje do Arduina po USB kabelu.



Předpokládám, že máte vše nastavené a propojené, na Arduinu svítí LED, která signalizuje napájení – pojďme na to! Klikněte směle na překlad a nahrání.

Bude to chvilku trvat, zhruba tak dlouho, jak budete číst tento odstavec, a až bude všechno přeloženo, tak se v tom zeleném pruhu pod editorem objeví nápis „Done upload“. V černém poli pod ním pak uvidíte buď lakonické sdělení o tom, kolik program zabírá paměti, nebo nějaké oranžové hlášky, které znamenají chybu.

Nejčastější chyba, věřte tomu nebo ne, je ta, že zapomenete vybrat port a typ desky. Viz výše.

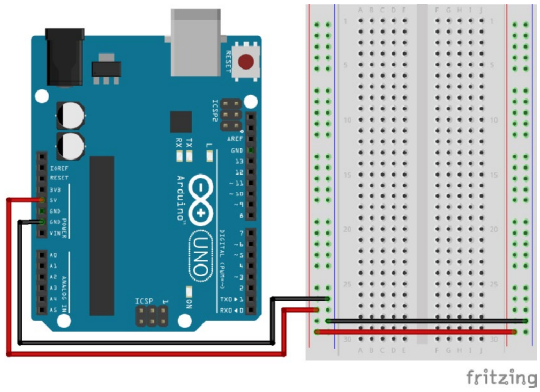
Pokud to všechno proběhlo, tak si všimněte, že na desce bliká LEDka.

To bylo jednoduché, co? Žádný kondenzátor, žádný integrovaný obvod, žádná LED, žádný rezistor… Jenže takhle se nic nedozvíme a nenaučíme. Takže uděláme jeden krok…

9.4 Krok zpět k drátům

Připravte si zase nepájivá kontaktní pole. Připravte si propojovací vodiče. Tentokrát odpojte napájení, vezmete si potřebné napětí přímo z Arduina. Všimněte si na Arduinu dole dvou vývodů,

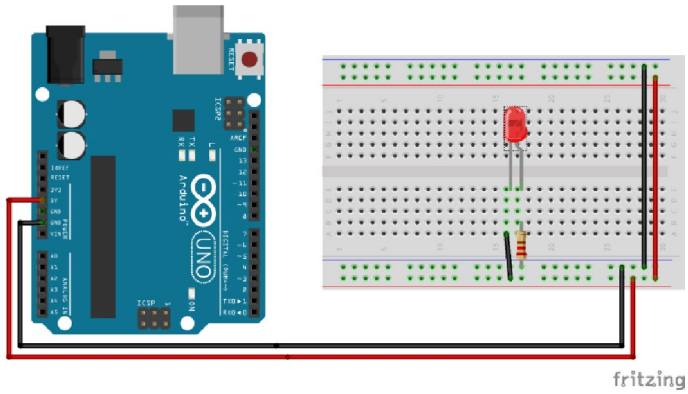
označených GND (ten je zdvojený) a 5 V. GND je i v horní řadě vývodů – je úplně jedno, který použijete, protože uvnitř jsou stejně propojené. A teď připojte tyto dva vývody k napájecím lištám na kontaktním poli. Nějak takto:



Tak, a teď pokaždé, když Arduino poběží, budete mít na kontaktním poli i potřebné napájecí napětí. Jupí.

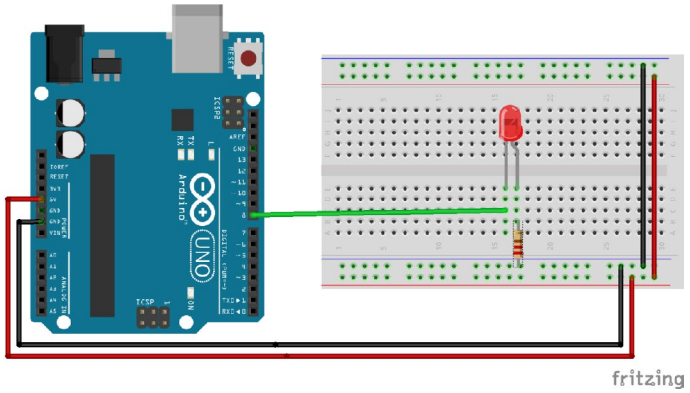
Zapojte si LED.

Z minulé kapitoly si jistě pamatujete, že LED má dva vývody a zapojuje se k ní rezistor. Udělejte to podle obrázku:



Připojte Arduino k počítači – ani v něm nemusí být nic nahraného. LED by měla svítit. Pokud nesvítí, otočte ji (a tím otočením myslím, že prohodíte její vývody, ne že ji třeba… co já vím… vzhůru nohama, nebo tak něco).

Svítí? Jupí. Teď ji propojte s Arduinem:



Všimněte si, co přesně jsem nakreslil – vodič, který propojoval LED a kladný pól, jsem odstranil, a místo kladného pólu jsem daný vývod LED zapojil do Arduina, na pin číslo 8. Proč? No protože jsem se rozhodl, že budu LED ovládat Arduinem. A vy taky!

Máte stále otevřený ten příklad Blink? Tak prosím všude, kde je LED_BUILTIN, napište 8. Jako že osmičku. Je to na třech místech. Výsledek bude vypadat nějak takhle:

```
// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin LED_BUILTIN as an output.
    pinMode(8, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(8, HIGH);        // turn the LED on (HIGH is the voltage level)
    delay(1000);                          // wait for a second
    digitalWrite(8, LOW);         // turn the LED off by making the voltage LOW
    delay(1000);                          // wait for a second
}
```

Teď přeložte a nahrajte. Pokud vše půjde bez problémů, stane se co? Správně, bude blikat ta naše LED!

Ptáte se, proč právě osmičku? Všimněte si, že v té delší řadě vývodů jsou vývody očíslované od 0 do 13. Je tu tedy 14 vývodů – pinů, k nimž můžeme připojit téměř cokoli. V praxi se moc

nepoužívají vývody 0 a 1, protože přes ně Arduino komunikuje se světem, ale ten zbytek je volný. A já jsem LED připojil do vývodu číslo 8. Co kdybych ji zapojil do vývodu číslo 9, co by se stalo?

Neptejte se mě, zkuste si to – přepojte vodič z osmičky na devítku, uvidíte sami. Možná bude potřeba ještě něco v programu změnit… ne?

Máte?

Dobře, a teď si zapojte dvě LEDky. Úplně stejné zapojení, druhá LED taky potřebuje rezistor, tak na to nezapomeňte, no a jednu LED připojte k Arduinu na výstup 8, druhou na výstup 9, a zkuste si, aby blikaly na střídačku.

Máte i toto?

Právě jste vytvořili svůj první programovatelný číslicový obvod! Jaký to je pocit?

(Chvilka na opájení se vlastní šikovností…)

Pojďme teď – nebo ne, ještě chvilku se chvalte, jak jste šikovní! Zasloužíte si to!

(Ještě krátká chvilka!)

A teď dojde na lámání chleba. *Hello world* už máme, tak teď následují ty protivné prozaické otázky: Proč to svítí? Proč to obráceně nesvítí? Co je to těch 5 voltů? Proč zrovna pět? A jak je možné, že já tady napíšu nějaká písmena, z toho se stane nějaký kód, a ten se nějak nahraje do té součástky? A jak se tam jako nahraje? Kam se tam nahraje? A jak ta součástka pozná, že má zrovna blikat LEDkou?

Ale ještě, než si to řekneme, zkuste si cvičení pro bystré hlavy: Zapojte dvě LED na jeden vývod Arduina tak, aby při stavu HIGH svítila jedna, a při stavu LOW svítila druhá! Napadá vás, jak to udělat?

9.5 Arduino a EduShield

Arduino je skvělá věc k demonstrování základů číslicové techniky. Někteří vyučující postupují jako já v této knize: dají lidem do ruky hrst součástek a nechají je zapojovat a zkoumat. Což je fajn, když je dost času. Pokud ale děláte třeba půldenní workshop s dvaceti lidmi, tak brzy zjistíte, že někdo prohodí vodiče, otočí součástku, vytvoří zkrat, součástku tak zničí a pak se diví, že mu to nefunguje tak, jak mu od tabule vyprávíte…

Na kurzech Arduina jsme proto sáhli k jednoduchému řešení: vytvořili jsme destičku s několika základními součástkami, které jsou ozkoušené, zapojené a připravené k testům. Tuto destičku

jsme nazvali EduShield. Vyrábí a dodává ji CZ.NIC. Podrobnosti, včetně příkladů a firmware naleznete na GitHubu:

https://github.com/arduino-edushield

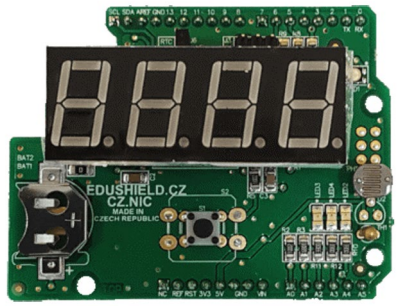
EduShield můžete použít i vy k ověření některých zapojení.

EduShield obsahuje následující periferie:

- Tlačítko, připojené na pin 2,
- RGB LED, připojenou na piny 5 (zelená), 6 (modrá) a 9 (červená),
- Tři barevné LED: zelená (pin 13), červená (pin 16) a oranžová (pin 17),
- Fotorezistor na vstupu A0,
- Termistor na vstupu A1,
- Hodiny reálného času a displej, obojí připojené na sběrnici I^2C.

Nebojte se, že zatím některým termínům nerozumíte. Během dalších kapitol pochopíte.

Já budu EduShield občas zmiňovat, když narazíme na nějaký příklad, který se na něm dá dobře demonstrovat. K pochopení výkladu ale není nezbytně nutné, abyste jej měli.



10 Fotorezistor

10 Fotorezistor

Už jsme si ukázali svítící diody, tedy součástky, které mění elektrický proud na světlo. Nastal čas podívat se na součástku s opačnou funkcí, tedy takovou, která reaguje na světlo. Už vás asi nepřekvapí, že takových je celá řada. Já začnu tou nejjednodušší, **fotorezistorem**.

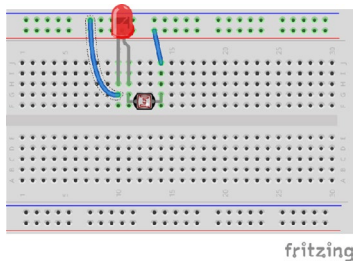


Jak název napovídá, půjde o nějaké spojení světla a rezistoru. Fotorezistor funguje velmi podobně jako rezistor, jen s tím rozdílem, že se jeho odpor mění podle toho, kolik světla na něj dopadá. Čím víc světla, tím nižší odpor.

Všimněte si, že fotorezistor má průhledný kryt – logicky, protože musí propouštět světlo – pod kterým je vidět struktura, připomínající vlnovku. Ta je tvořena polovodičem. Světlo, dopadající na polovodič, zvyšuje množství volných elektronů, které tak mohou sloužit jako přenosové médium. Větší množství světla znamená víc volných elektronů, a tedy vyšší vodivost.

U fotorezistorů se udává nejčastěji odpor při osvětlení 10 luxů a teplotě 25 °C (označuje se R_{10lx}) a u těch nejčastějších a nejlevnějších, kterých koupíte za pětikorunu celý sáček, to bývá okolo 10 kΩ. Bez dalších znalostí, například přesného typu nebo kalibrace, se moc nehodí na přesné měření osvitu, ale to ve spoustě aplikací nevadí. Třeba pro automatické spínání osvětlení nebývá potřeba přesně stanovit, že se má spustit při 28 luxech a vypnout při 37 luxech. Stejně tak například u optických závor (zařízení, které reaguje na přerušení paprsku) nám stačí vědět, jestli světlo dopadá, nebo nedopadá (a tedy jestli je odpor nízký / vysoký).

Pojďte si to zkusit, jestli to funguje tak, jak si představujete. Co když zapojíte takový fotorezistor místo obyčejného rezistoru do série s LED?



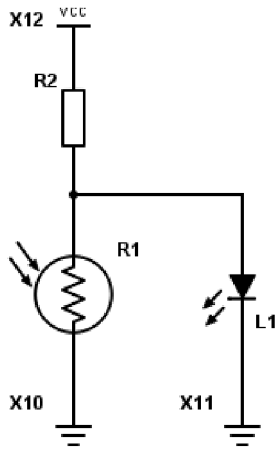
LED bude svítit, ovšem když zastíníte fotorezistor prstem, její jas se sníží!

10.1 Obrácená logika

Je hezké, že LED svítí, když je fotorezistor osvícen, ale co když to chcete obráceně, tedy aby se LED rozsvítila tím víc, čím je větší tma? To bude užitečnější, že?

Jak na to? Vlastně potřebujete, aby větší množství světla znamenalo menší napětí na diodě… Použijte k tomu známý rezistorový dělič:

Ve tmě je odpor fotorezistoru R1 velký, takže proud teče přes rezistor R2 do LED a ta svítí. Pokud na fotorezistor posvítíte, jeho odpor klesne, víc proudu poteče přes něj do země, napětí na LED tak poklesne…



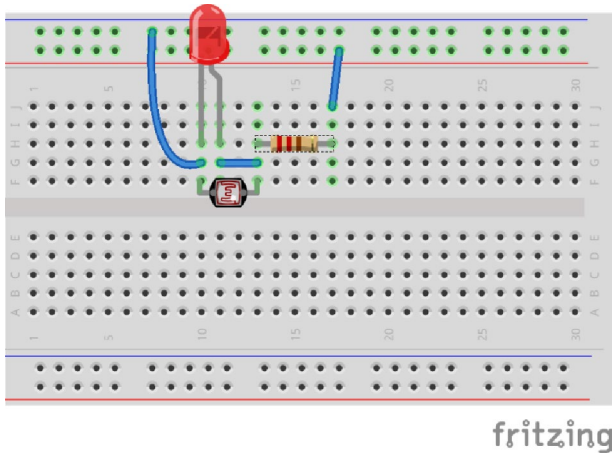
Jak velký by měl být rezistor R2? To si můžete spočítat podle odporu fotorezistoru R1 v závislosti na osvětlení. Nejjednodušší je změřit si odpor fotorezistoru ve tmě a za osvětlení. U mého to bylo zhruba 20 kΩ ve tmě a 200 Ω při přímém osvětlení. Pak jsem si nasimuloval „šero“, tedy úroveň, při níž bych chtěl, aby se LED rozsvítila. Bylo to okolo 10 kΩ.

Teď už stačí spočítat dělič napětí tak, aby při „šeru“ byly na diodě právě 2 volty (to je hranice rozsvěcení pro červené LED, pamatujete?) Pokud to celé napájím pěti volty, tak musím zvolit R2 tak velký, aby při R1 = 10 000 bylo napětí 2 V.

Vzorec si jistě pamatujete – U1 = U . (R1 / Rx), kde Rx = R1 + R2. Po dosazení: 2 = 5 . (10 000 / (10 000 + R2))

Po úpravách: R2 = (U × R1 / U1) – R1

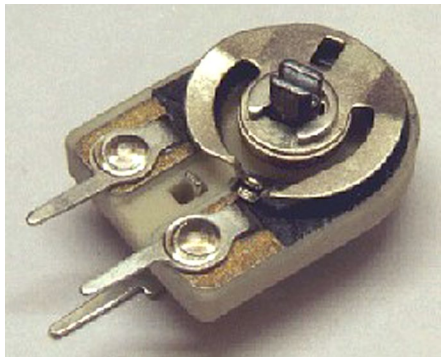
Vychází mi 15 kΩ. Vám taky? S rezistorem R2 o velikosti 15k tedy budou „za šera“ na LED dva volty, a LED začne svítit. Teda – teoreticky. Můžete si zkusit přijít na důvod, proč to zpochybňuju, za chvíli se k tomu vrátím.



V praxi bych udělal ještě jednu úpravu: na místě R2 bych použil nastavitelný rezistor – tedy potenciometr, nebo menší variantu: trimr.

10.2 Trimry

Celým jménem „potenciometrické trimry“ jsou vlastně potenciometry, tedy rezistory s proměnným odporem, ovšem bez té osy, kterou se dá točit prstem. Místo toho mají jen malý výřez.



Takový trimr se nastavuje při oživování zařízení pomocí malého šroubováku. Nastaví se tak, aby zařízení správně fungovalo, a pak se s ním už nehýbe.

U našeho „spínače osvětlení" bych tuto součástku použil, protože pomocí ní snadno nastavím úroveň osvětlení, při níž má obvod sepnout, a navíc, když si někdy v budoucnu usmyslím, že by mi víc vyhovovala jiná úroveň osvětlení, tak můžu obvod snadno přenastavit, nemusím měnit rezistor.

10.3 Lepší řešení detektoru tmy

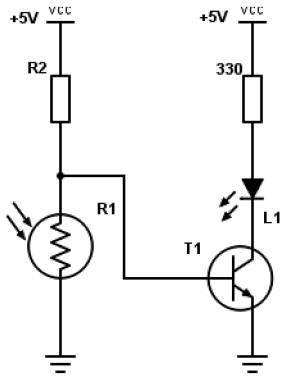
Všimněte si, že u výše uvedeného zapojení jste limitováni odporem R2 – jeho velikost 15 kΩ znamená, že i když bude absolutní tma, tak do diody může téct maximálně 0,33 mA. Což je hodně málo. Když odpor zmenšíte, bude zase LED svítit i při větším osvětlení, takže se nakonec dostanete do stavu, kdy LED ve tmě svítí víc, a na světle svítí míň. Což tedy není nic moc.

Kéž by tak byl způsob, jak malým proudem, co proteče z R2, sepnout velký proud pro LED, že?

Moment, co že jsem to právě napsal? *Malým proudem sepnout větší proud*? Kde jsem tohle už psal? To bylo – no ano, to bylo v kapitole o tranzistoru.

Připomenu teorii: Tranzistor sepne tehdy, když je na jeho přechodu báze-emitor napětí větší než 0,7 voltu. Pak stačí malý proud k tomu, aby vedl násobně větší proud ve směru kolektor-emitor.

Upravme naše zapojení tak, že využije tranzistor ke spínání LED. Budeme muset upravit hodnotu R2 tak, aby při našem limitu „šero" bylo na bázi 0,7 voltu (místo 2 V) – vychází to na cca 62 kΩ.



Teoretický výsledek je fajn, ale nejlepší bude na místě R2 použít trimr 100k, a ten přesně nastavit podle požadované úrovně osvětlení.

Za naprosté a absolutní tmy (nereálné, já vím) poteče rezistorem R2 proud 0,08 mA (uvažuju hodnotu R2 rovnou 62k). Pokud použijeme např. tranzistor BC547B s proudovým zesilovacím činitelem okolo 300, může při 0,08 mA bází téct kolektorem (a tedy LEDkou) až 24 mA, což je dost (ve skutečnosti bude proud omezen rezistorem 330R na 15 mA).

Tady bych chtěl upozornit na důležitou věc: **Takovéhle výpočty jsou orientační.** Sice pro většinu aplikací stačí, ale pokud se dostanete blíž k limitním hodnotám, například k velkým proudům skrz kolektor, posunou se i jiné hodnoty mimo „standardně uvažované hodnoty". Pro takové to domácí kutění je to dostatečné, ale snažte se vždy pohybovat v rozumných mezích a nejít, jak se říká, *až na krev*. Pak se součástky mohou chovat výrazně jinak. Vždy se proto podívejte do datasheetu, jaké jsou mezní hodnoty – nejčastěji pro protékající proud a připojené napětí.

10.4 Fotorezistor a Arduino

Jeden z nejčastěji připojovaných senzorů k Arduinu je právě fotorezistor. Je levný, jednoduchý na obsluhu, tak co by ne, že?

Pro takovéhle typy snímačů, které nemají digitální výstup (třeba jako tlačítko), ale (nejčastěji) mění svůj odpor, má Arduino speciální vstupy – analogové A0 až A5. Funkce analogRead() dokáže zjistit, jaké napětí je na tomto vstupu, a převést ho na škálu hodnot 0 až 1023.

Nula samozřejmě znamená, že daný vstup má úroveň země. Ale kolik je maximum, tedy těch 1023?

To je správná otázka. Odpověď zní: 1023 je hodnota, která je naměřená tehdy, když je na analogovém vstupu takzvané referenční napětí!

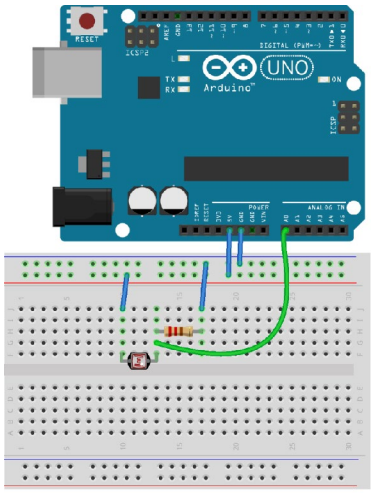
Odpověď jak od chytré horákyně, že? A kolik je tedy to referenční napětí?

Většinou, pokud není určeno jinak, to je napájecí napětí, tedy zhruba 5 voltů. Zhruba píšu proto, že toto napětí může kolísat, a u některých typů Arduin může být třeba 3,3 voltů. U Arduina UNO ale bude zhruba těch 5 voltů.

Můžete ale použít vnitřní zdroj referenčního napětí 1,1 voltu.

No a konečně můžete říct, že si referenční napětí, ne vyšší než 5 V, vytvoříte nějak sami a přivedete ho na vstup AREF. Jak si ho vytvoříte? Buď nějakým stabilizátorem, nebo specializovanou součástkou – generátorem referenčního napětí. Dělají se například přesné generátory 1,023 voltu – pak naměřená hodnota odpovídá velmi přesně milivoltům.

Ale to jsem trochu odbočil. Pokud k Arduinu připojujete třeba fotorezistor nebo podobnou součástku (například termistor, což je rezistor, jehož odpor je závislý na teplotě), používá se většinou některý z analogových vstupů, a součástku zapojujete v sérii s rezistorem, jako napěťový dělič. Analogový vstup totiž měří *napětí*, nikoli *proud*. Kdybyste zapojili třeba jen fotorezistor na + 5 voltů, tak by se se změnou odporu měnil proud, ale napětí by bylo stále 5 voltů.

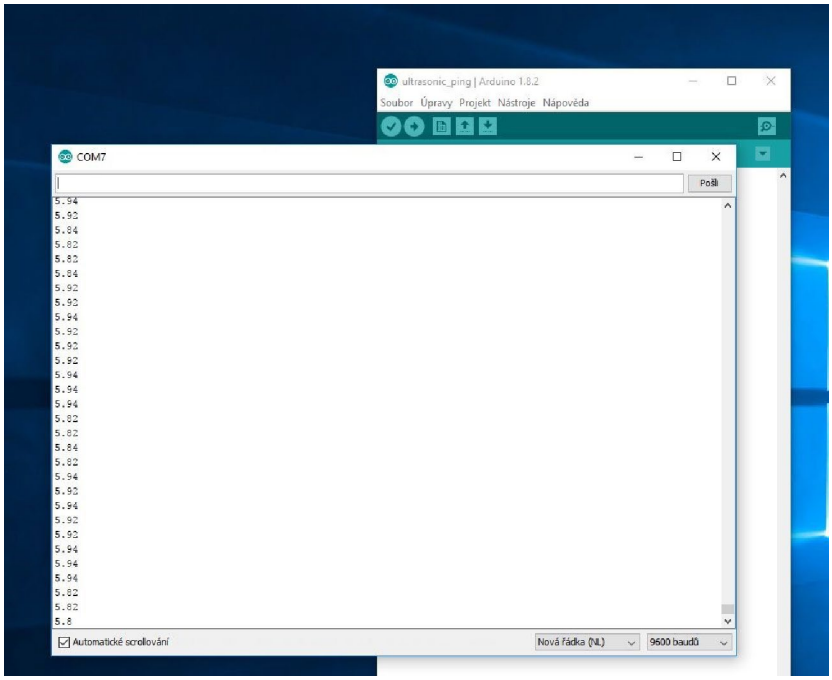


Když si fotorezistor takto zapojíte, zjistíte, že při velmi silném osvětlení funkce analogRead(A0) vrátí téměř 1023, ve tmě klesne – nikoli na nulu, protože odpor fotorezistoru nebude nekonečný, takže napětí neklesne k ideální nule, ale klesne hodně nízko.

```
void setup () {
    Serial.begin(9600);
}
void loop() {
    int a = analogRead(A0);
    Serial.println(a);
}
```

Když tento program napíšete do Arduino IDE a nahrajete do Arduina, bude měřit napětí na vstupu A0 a vypisovat ho na sériovou konzoli. Sériová konzole je způsob, jakým Arduino může posílat údaje do připojeného počítače. Princip popíšu později, teď stačí, abyste věděli, že v Arduino IDE je v menu Nástroje (Tools) položka Sériový monitor (Serial Monitor). Když ji vyberete, otevře se nové okno, a v něm se vypisuje to, co Arduino pošle funkcí Serial.println().

Všimněte si, že ve funkci Setup volám Serial.begin(9600). Tato funkce zajistí zapnutí této komunikace a nastavení přenosové rychlosti. Toto číslo je velmi důležité – přenosová rychlost, s jakou



Arduino pracuje, musí být shodná s přenosovou rychlostí, jakou komunikuje počítač.

Ta se nastavuje v Sériovém monitoru vpravo dole – všimněte si hodnoty „9600 baudů“.

Postavte si teď přesně takový detektor tmy, jako předtím, ale tentokrát s Arduinem. Vlastně moment – vy jste si ho už postavili, to na předchozím obrázku, to je přesně on! Použijete LED, která je v Arduinu zapojena na výstupu 13, a máte to! Teď stačí v nekonečné smyčce loop() opakovat stále dokola totéž: Přečíst analogový stav na vstupu A0, a podle načtené hodnoty LED buď rozsvítit, nebo zhasnout. Buď metodou pokus-omyl, nebo vypsáním naměřených hodnot zjistíte, která hodnota je pro vás už „dostatečná tma“, a pokud načtete víc, LED zhasnete, pokud míň, LED rozsvítíte. Takhle jednoduché to bude.

Když použijete EduShield, tak nemusíte ani nic zapojovat. EduShield má totiž na sobě fotorezistor zapojený přesně tak, jak jsem právě popsal.

11 Termistor

11 Termistor

Už několikrát jsem tu zmínil v nejrůznějších obměnách větu „… a odpor závisí i na teplotě“. U většiny součástek je to vlastnost nevítaná, ale někdy se hodí – třeba pro měření teploty. Vyrábí se k tomu speciální součástky – rezistory s odporem, závislým na teplotě, zvané **termistory**.



Termistory jsou dvojího typu – jedny, ty běžnější, fungují tak, že s rostoucí teplotou jejich odpor klesá (označují se jako NTC), druhé, označované PTC, svůj odpor s rostoucí teplotou zvyšují.

Pokud máte po ruce termistor, zapojte si jej místo fotorezistoru k Arduinu. Zbytek zůstane tak, jak byl. Nechte si vypisovat naměřenou hodnotu a zkuste termistor zahřát – například tak, že ho chytnete do prstů.

V EduShieldu je také termistor, ale zapojený na vstup A1.

Po načtení získáte celé číslo v rozmezí 0 až 1023. Jak přepočítat toto číslo na skutečnou teplotu? No, není to úplně triviální. Vzorec (jmenuje se Steinhartův-Hartův, Steinhart-Hart Equation) vypadá takto:

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{B} \ln\left(\frac{R}{R_0}\right)$$

„Magické“ hodnoty B, R_0 a T_0 naleznete v datasheetu k danému termistoru. T_0 je referenční pokojová teplota 25 °C, R_0 je odpor rezistoru při této teplotě (například 10 000 ohmů) a B je teplotní koeficient (například u termistoru, který je v EduShieldu, je to 3950).

Pro přepočet na stupně Celsia je zapotřebí nejprve z naměřené úrovně napětí spočítat odpor termistoru (musíme znát odpor druhého rezistoru v děliči napětí), a dosadit tento odpor do výše

uvedeného vzorce. Což s Arduinem samozřejmě lze, ale je to docela náročný výpočet, který používá desetinná čísla atd.

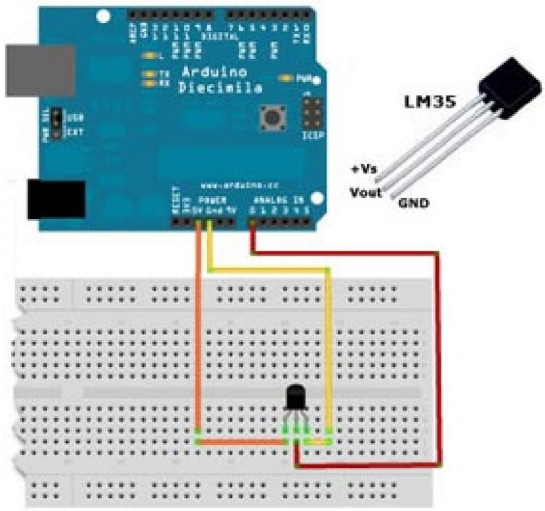
Proto se používá podobný fígl, jako u měření světla – většinou potřebujete něco udělat při nějaké hodnotě teploty, tak stačí zjistit, že při dané teplotě je naměřená hodnota například 700, takže i bez přepočtu můžete říct: Pokud překročí hodnota 700, tak se něco stane.

Pokud ale chcete třeba vytvořit teploměr, který ukazuje teplotu, nezbyde vám, než přepočítávat naměřenou hodnotu výše uvedeným vzorcem na stupně. Anebo jít jinou cestou, třeba použít sofistikovanější součástku…

12 LM35

12 LM35

LM35 je přesný teploměr, který vypadá trochu jako tranzistor. Dva krajní vývody jsou zem a napájecí napětí, prostřední vývod je měřicí. Napětí na něm je přímo úměrné teplotě – jednomu stupni Celsia odpovídá 10 milivoltů. Při teplotě 0 °C je na tomto výstupu 0, při 100 °C je na vývodu přesně 1 volt.



Teplotu pak spočítáte jednoduše, pomocí vzorce:

```
temp = (5.0 * analogRead(tempPin) * 100.0) / 1024;
```

Magické konstanty 5.0 a 1024 jsou maximum (5 V) a hodnota, kterou při tomto maximu naměříte (1024).

V praxi těžko budete měřit teploty nad 100 °C, takže se dá předpokládat, že napětí na měřicím výstupu nepřesáhne 1 volt. Můžete tedy přepnout referenční napětí pro převodník (pamatujete?) na vnitřní zdroj 1,1 V, pomocí funkce analogReference. Zvýšíte tím i přesnost měření.

Vzorec se tedy lehce změní. Maximum (1023) není 5 voltů, ale 1,1. Když 1,1 podělíte 1024 (to je rozsah měření), zjistíte, že nárust hodnoty o 1 odpovídá nárustu napětí o 1,0742 mV (u pěti voltů to bylo 4,88 mV). Pokud jednomu stupni Celsia odpovídá 10 mV, znamená to, že odpovídá hodnotě zhruba 9,31 (10 / 1,0742). Pokud naměřená hodnota vzroste o 9,31, znamená to, že teplota vzrostla o jeden stupeň. Zkuste si to sami:

```
float tempC;
int reading;
int tempPin = A0;

void setup() {
    analogReference(INTERNAL);
    Serial.begin(9600);
}

void loop(){
    reading = analogRead(tempPin);
    tempC = reading / 9.31;
    Serial.println(tempC);
    delay(1000);
}
```

13 „Jak naučit kámen počítat“

13 „Jak naučit kámen počítat”

Seriál s tímto názvem vycházel v časopise Věda a technika mládeži někdy v polovině 80. let minulého století. Postupně vysvětloval, jak je možné, že z kusu křemenu může postupně vzniknout celý počítač. Já si název vypůjčil pro tuto kapitolu.

Když jsem přemýšlel nad tím, kdo je vlastně modelový čtenář této knihy, měl jsem před očima velmi přesný obraz: na kurzech a školeních za mnou chodili programátoři a říkali: „Já bych tak rád zkusil něco s elektronikou, ale nevím, nerozumím tomu, nechápu, bojím se…“

Jeden z nich mi položil i skvělou otázku, která stála vlastně na začátku celých úvah o této knize. Programoval hry a říkal: „Já vím, jak to naprogramuju, pak vím, že jsou nějaké knihovny, tomu pořád rozumím, překladačem to přeložím – a co se děje pak? Jsou to nějaká čísla? Jak se uloží? Kam? A jak ten procesor ví, co má udělat? To vůbec netuším… jen vím, že pak někde úplně dole jsou dráty a elektrony…“

A tahle kniha je vlastně obšírnější odpovědí na tento dotaz. Jen jsme začali od konce, od těch elektronů a drátů. Teď už to bude jen jednodušší, slibuju.

13.1 Stavebnice

Vždycky zájemcům říkám, že není čeho se bát. Že to všechno, co mají ve svém počítači, jsou vlastně tytéž tranzistory a rezistory, jen poskládané do větších a větších funkčních celků.

Z několika tranzistorů je poskládané takzvané **logické hradlo**. To je součástka, která dělá právě jednu **logickou funkci**. K nim se dostaneme vzápětí. Několik logických hradel dohromady dává větší celek – kombinační obvody či sekvenční obvody. Multiplexory, demultiplexory, dekodéry, registry, čítače, klopné obvody, sčítačky… Nebojte, za chvíli si vysvětlíme, co jednotlivé výrazy znamenají. Tyto obvody se opět dají kombinovat do větších celků – mnoho registrů spolu s datovými multiplexory a adresními dekodéry tvoří paměť. Několik sčítaček a hradel spolu s multiplexory vytvoří aritmeticko-logickou jednotku. Čítače zapojené za sebou dají třeba „programový čítač“ (PC). Když zkombinujeme registry, čítače, aritmeticko-logickou jednotku a dobře navržené dekodéry, získáme mikroprocesor. Jednoduché registry s posíleným výstupem vytvoří obvody pro vstup a výstup… Z toho všeho pak můžeme poskládat vlastní počítač. A věřte tomu, že váš domácí počítač vypadá uvnitř přesně tak, jak jsem popsal – každý z těch obvodů, který v něm vidíte, když ho rozmontujete, se uvnitř skládá z logických celků, které se skládají z jednodušších obvodů, které se skládají z elementárních hradel, které se skládají z tranzistorů… Vážně!

Pojďme tedy nejprve na ta hradla a logické funkce.

13.2 Logické funkce

Programátoři vědí, nebo by alespoň měli vědět, že máme tři základní logické funkce, totiž AND, OR a NOT.

Moment, moment, neotočil tu někdo víc stran najednou? Ještě v minulé kapitole tranzistory a volty, a najednou logické funkce? Neuteklo mi něco? Pojďme radši popořadě…

13.2.1 Digitální, nebo analogové?

Analogová elektronika se zabývá *spojitými veličinami*, digitální (též číslicová) veličinami *nespojitými*. Což je hezká školní definice, ale – co si pod tím mohu představit?

Typická analogová veličina je třeba, řekněme, teplota. Teplota plynule roste a klesá, a záleží jen na přesnosti teploměru, jak přesně tuto změnu změříme. Můžeme vidět, že teplota během dopoledne roste: 13 °C, 14, 15, 16, … Pak vezmeme přesnější teploměr, a sledujeme mnohem rychlejší změny: 13,0… 13,1… 13,2… a tak dál. Když si vezmeme ještě přesnější teploměr, můžeme změnu měřit po setinách, ale vidíme, že teplota se mění spojitě, že v přírodě není žádný „daný krok teploty“. (No, ve skutečnosti je, ale hodně malý, takže to můžeme zanedbat a brát teplotu jako spojitou veličinu.)

Naproti tomu například počet nějakých věcí, dejme tomu vajíček, je typická nespojitá veličina. Vajíčko můžete mít jedno, dvě, tři, nebo třeba žádné, ale nemáte 2,7153 vajíčka. Máte buď dvě, nebo tři. Počet, který vyjadřujeme *přirozenými čísly*, je typická nespojitá veličina. Nemění se plynule, ale jakoby „skokově“.

V elektronice můžeme také využívat oba přístupy. Typickou oblastí, kde dlouhou dobu hrála prim analogová technika, je například zpracování zvuku. Všechny ty reprosoustavy, zesilovače, ekvalizéry, efekty, snímače, mikrofony, to všechno pracuje se zvukem, který je ve své podstatě spojitý. (Dnes už to tak úplně neplatí, dnes už se hojně pro zpracování využívají digitální postupy, ale minimálně od zesilovače dál jde o analogovou techniku.)

Do analogové techniky patří také velmi zajímavá oblast, zabývající se nízkofrekvenčními a vysokofrekvenčními signály, jejich zpracováním, filtrováním, patří do ní součástky, zvané operační zesilovače a komparátory, a je to poměrně zajímavá oblast – ovšem mimo záběr naší knihy.

Představte si, že byste chtěli využít nějak téhle techniky k tomu, abyste něco spočítali. Třeba součet, nebo i logaritmus. Dejme tomu, že bychom řekli: Nula je 0 voltů, hodnota 10000 je 10 voltů, tedy jednička bude 1 mV. Dáme dva potenciometry, kterými budeme moci měnit „čísla“ – tedy napětí na výstupu. Pak mějme nějakou součástku (a taková opravdu existuje), která

zjistí, že na jednom vstupu je 18 mV, na druhém 278 mV, obě napětí sloučí a na výstupu poskytne 296 milivoltů. Tedy 18 + 278. Postavit to lze, ale není to tak úplně přímočaré – stačí když se někde nějaký vodič zahřeje víc, jiný méně, změní se odpory ve vedení, a vyjde vám, že 18 + 278 je tak nějak 280, plus mínus deset procent.

Že by takovou techniku nikdo nepoužíval? Ve skutečnosti používaná byla, a tato omezení se musela nějakým způsobem kompenzovat. Stálo to za to, protože analogové počítače, jak se takovým strojům říkalo, dokázaly téměř v reálném čase velmi složité operace, jako derivace a integrace signálu, které se numerickými metodami řešily velmi zdlouhavě. Programovaly se spojováním základních funkčních celků a dokázaly už někdy v 60. letech řešit složité lineární a nelineární diferenciální rovnice rychleji, než tehdejší číslicové počítače.

Číslicové počítače jsou založené na jiném principu. Nepracují se spojitými signály, ale s takzvanou **binární logikou**. Binární znamená, že má pouze dvě hodnoty: 0 a 1. Pak si třeba řekneme, že 0 bude vyjádřené nula volty a 1 vyjádříme pěti volty a že povolíme nějakou toleranci, tak je celkem jedno, jestli na vstupu bude přesně 5 V, nebo třeba 4,75 – výsledek bude stále hodnota „1“.

Jenže pokud budeme mít jen dvě hodnoty, jak třeba můžeme pracovat s přirozenými čísly? Naštěstí přišla matematika se svými **číselnými soustavami**. Dovolte jen rychlé opáčko: Jsme zvyklí počítat s desítkovou soustavou. Deset proto, že se rodíme s deseti prsty na rukou – a právě prsty na rukou jsou první počítadlo, které máme.

Mimochodem, víte, jak se řekne latinsky prst? Digitus. Prsty byly první počítadlo přirozených čísel. Od toho „digitální“, neboli „číslicové“.

13.2.2 Dvojková soustava

V desítkové soustavě používáme deset základních symbolů pro jednotlivé číslice – totiž 0, 1, 2, 3, 4, 5, 6, 7, 8 a 9. Pokud potřebujeme zapsat větší číslo, desítku, použijeme k tomu poziční zápis a zapíšeme vedle sebe symbol pro desítky a symbol pro jednotky: 10. A můžeme pokračovat dál, přidávat další jednotky a počítat: 11, 12, 13… až do 19. Když přidáme další jednotku, tak nám počítadlo jednotek opět „přeteče“, a my tedy o jedničku zvýšíme číslici na pozici desítek: 20. Pozice tedy zprava stoupají – jednotky, desítky, stovky, tisíce… Každý řád je desetinásobek následujícího.

Já vím, je to elementární, umíme to všichni a denně to používáme, takže nám to ani nepřijde. Ovšem zopakoval jsem to proto, že si na tom můžeme ukázat přepočítávání do jiných číselných soustav. Když jsem chodil do školy, tak jsme někdy v páté třídě přepočítávali mezi různými soustavami, a byla to ta nejdivnější činnost – nikdo z nás netušil, proč to děláme a k čemu to může být dobré. Pokud to ani vy doteď netušíte, tak se to právě teď dozvíte.

Řekli jsme si, že v binární logice máme jen dvě hodnoty: 0 a 1. Jak pomocí těchto dvou symbolů zapíšeme nějaká čísla? Princip je stejný jako v té naší desítkové soustavě. Dívejte:

Nulu zapíšeme jako 0. Jasné.

Jedničku zapíšeme jako 1. Taky jasné.

Dvojku zapíšeme jak? No, další symbol za 1 už nemáme, tak jedeme opět od 0, a použijeme další řád: 10

Trojku tedy zapíšeme jako 11.

Na poslední pozici jsou zase jednotky. Na předposlední nejsou desítky, ale „dvojky“, na pozici před nimi jsou „čtyřky“, před nimi „osmičky“ – tedy řada jde po dvojnásobcích: 1, 2, 4, 8, 16, 32, 64, 128, …

Čtyřka bude tedy 100. Pětka 101, šestka 110, sedmička 111.

Osmička bude 1000, devítka 1001, desítka 1010…

K zapsání desítky potřebujeme tedy čtyři symboly. Jaké je nejvyšší číslo, které můžeme zapsat čtyřmi symboly? Je to 15, které zapíšeme jako 1111.

K zapsání čísla v rozsahu 0 až 15 potřebujeme tedy čtyři binární číslice. Z anglického výrazu pro binární číslici „BInary digiT“ vzniklo zkratkové slovo „bit“. Čtyři binární číslice, čtyři bity nám tedy stačí pro zápis šestnácti hodnot (proč 16, když nejvyšší je 15? Protože první je nula, druhá jednička, třetí dvojka… a šestnáctá hodnota je 15).

Digitální technika tedy pracuje se signály, uspořádanými do N-tic bitů. Čtveřice bitů pojme pohodlně číslice od 0 do 9, a k tomu ještě šest navíc. Osmice bitů může vyjádřit číslo od 0 do 255. Dvě osmice, tedy 16 bitů, dokáže vyjádřit celá čísla od 0 do 65535. Matematicky zapsáno je možných kombinací 2N.

Nejčastěji se budete setkávat s obvody čtyřbitovými a osmibitovými. Vícebitové součástky samozřejmě existují také, ale více bitů znamená více vývodů ze součástky, tedy buď větší pouzdro, nebo menší vývody… Pro amatérské konstrukce jsou vhodné právě ty méněbitové součástky.

13.2.3 Šestnáctková soustava

Programátoři ji jistě znají, říká se jí také někdy hexadecimální, méně správně „hexa“. Je vhodná právě pro práci s číslicovou technikou, protože dokáže jedním znakem vyjádřit stav čtyř bitů.

Jak už víme, čtyři bity mohou mít 16 různých kombinací. K jejich zápisu použijeme číslice 0 až 9 a pro hodnoty 10, 11 atd. použijeme písmena A, B, C, D, E, F.

Osmibitové číslo vyjádříme pouhými dvěma znaky: 00, 2A, 3F, DC, 12. Aby se čísla nepletla, zapisují se buď s postfixem „h“, nebo s prefixem „0x“. Tedy 0x2A, 0x12, popř. 3Fh, 56h a podobně.

Pokud to stále ještě neumíte, naučte se zpaměti převody mezi hexadecimálními, desítkovými a dvojkovými čísly, alespoň do těch 16. Vážně.

13.2.4 Zpátky k technice

Když jsme si zopakovali, jak pomocí dvojkové soustavy můžeme přenést diskrétní informaci, pojďme se podívat, jak je to vlastně realizované fyzicky. Už jsem to tu naznačil, tak to pojďme dokončit.

Nejjednodušší způsob, jak vyjádřit logickou informaci pomocí elektřiny, je ten, že jedné logické úrovni bude odpovídat nějaké napětí, jiné logické úrovni napětí jiné. Nabízí se, že logická nula bude 0 voltů, stav bez napětí, a logická jednička pak nějaké konkrétní dohodnuté napětí – třeba + 5 voltů. No a tak tomu je v podstatě doposud.

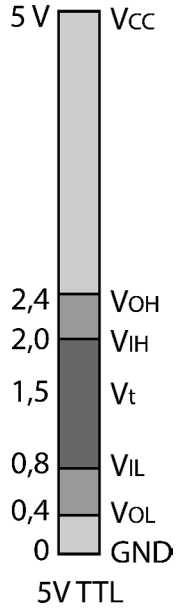
Ne že by neexistovaly jiné způsoby. Například u jednoho sériového rozhraní je to tak, že logická 0 je definováno jako napětí + 15 voltů proti zemi, logická 1 je napětí - 15 voltů proti zemi. Některé logické obvody byly dřív stavěny tak, že logická 1 byla definována jako záporné napětí, ještě jiné pracovaly s obrácenou logikou, ale nakonec se jako nejpoužívanější uchytily dvě technologie, které zároveň definují onu otázku: „Kolik voltů je logická 1?“

13.3 TTL a CMOS

Jako první přišly číslicové obvody typu RTL (Resistor-Transistor Logic) a DTL (Diode-Transistor Logic). Z nejrůznějších důvodů, jako je třeba „vysoká spotřeba“ a „nízká rychlost“ se moc neuchytily. I když – jak se to vezme. Řídicí počítač pro lunární misi Apollo používal právě obvody RTL. Ale brzy přišly obvody lepší, rychlejší...

Až další evoluční krok, nazvaný TTL (Transistor-Transistor Logic), vyšel. S touto technologií bylo navrženo a vyrobeno velké množství typů číslicových obvodů, a dá se s nimi potkat dodnes.

Technologie TTL stanovovala výše zmíněné úrovně: 0 voltů pro logickou 0, + 5 voltů pro logickou 1. Díky technologii, která byla použitá, nejsou hodnoty naprosto striktní. Běžný TTL obvod vezme jako logickou 0 jakékoli napětí mezi 0 a 0,8 voltu a jako logickou 1 napětí mezi 2 a 5 volty. Tyto hranice se označují jako V_{IL} a V_{IH}. Proč?



Je takový zvyk označovat logickou nulu písmenem L (Low, nízký, nepravda, false, ...) a logickou jedničku písmenem H (High, vysoký, pravda, true, ...) Používá se to tam, kde by se mohlo plést označení 0 a 1 s nějakými jinými číselnými hodnotami. Ale zvykněte si, že veškerá literatura nakládá s tímto označením naprosto volně, v jednom datasheetu se setkáte s H, L, v druhém s 1, 0… Mělo by platit, že 0 a 1 se používá tam, kde se hovoří o logické hodnotě, L a H tam, kde jde o konkrétní vyjádření těchto hodnot, jenže naprosto běžně uslyšíte mluvit dva techniky, že „mají ten vstup v jedničce“. Takže doporučuju se s tím smířit.

Označení V_{IL} tedy znamená *Voltage on Input in Low* a V_{IH} je *Voltage on Input in High*. Tedy (nejvyšší) napětí na vstupu, které se ještě považuje za 0 (L), a (nejnižší) napětí na vstupu, které se už považuje za 1 (H). Logická nula je tedy od 0 voltů do V_{IL}, logická jednička od V_{IH} do Vcc.

Samozřejmě že existují dva obdobné údaje pro výstup: V_{OL} a V_{OH}. Tedy maximální napětí na výstupu pro log. 0 a minimální napětí na výstupu pro log. 1. U TTL to je 0,4 voltu pro L a 2,4 pro H.

Logická otázka: **Co se stane, pokud je na vstupu třeba 1,5 voltu?** To je správná otázka, a odpověď zní: Je průšvih! Integrovaný obvod neví, jestli se má otevřít, nebo zavřít, tak je v takovém divném mezistavu, ani ryba, ani rak, je polootevřený, teče skrz něj velké množství energie a jeho stav nelze předvídat ani definovat. Proto se snažíme těmto hodnotám vyhnout a dodržet je co nejblíž maximům, tj. 0 a 5 V.

Totéž platí pro vstup, který není s ničím propojený a, jak se říká, „visí ve vzduchu“ (někdy doslova). Takový vstup způsobuje jen a pouze problémy, protože velmi rád chytá indukcí rušení z okolí.

Postupem času se původní technologie TTL zlepšovala a přišly novější. Jako první přišla řada „TTL-S“ (Schottky), která byla rychlejší, ale měla vyšší spotřebu. K ní komplementárně přišla technologie TTL-L (Low Power), která měla menší spotřebu, zato byla pomalejší. Až po letech se podařilo zkombinovat obě do technologie TTL-LS, která má menší spotřebu, a zároveň je rychlejší. Ta byla pak opět vylepšena do podoby TTL-ALS (Advanced Low Power Schottky).

Obvody TTL zpočátku vyráběli výrobci stylem „každý pes jiná ves“, ale brzy se jako standard uchytila řada 74xx od Texas Instruments. Ostatní výrobci začali vyrábět stejné obvody, označené stejným číslem, jen jiným kódem výrobce. Například originál od Texas Instruments měl označení SN7404. Československá Tesla vyráběla ekvivalentní obvod se stejnou funkcí a stejným zapojením pod označením MH7404, Polsko jej označovalo UCY7404, tehdejší východní Německo vyrábělo ekvivalentní D104, maďarský Tungsram dodával týž obvod jako 7404APC. Americký National Semiconductors používal písmena DM: DM7404. A tak dále.

Platí, že bez větších problémů můžeme nahradit třeba obvod SN7404 obvodem MH7404 nebo jiným 7404. Měl by mít stejnou spotřebu i rychlost. Do jisté míry můžeme i obyčejný obvod 7404 nahradit obvodem z řady L, S, LS, ALS – například SN74ALS04. Pokud nejde o nějaké zapojení, kde je kritická rychlost obvodu nebo jeho zisk (vysvětlím později), nebude s tím problém.

Existovaly i řady 54xx a 84xx, které byly určené pro náročnější výrobky, třeba pro takové, co musely pracovat v extrémních teplotách apod. Jinak se od verze 74xx neliší (snad krom ceny).

Časem se ukázalo, že technologie TTL má dvě velké nectnosti – velkou spotřebu, která se projevovala v tom, že obvody hřály, a že zabírají hodně místa, takže se na křemíkový čip vešlo jen omezené množství tranzistorů.

Druhý problém řešila jiná technologie, nazvaná MOS. Ta místo tranzistorů bipolárních používala tranzistory MOSFET. Nejprve jako PMOS, později NMOS, a nakonec CMOS – ta je tvořena dvojicí tranzistorů PMOS a NMOS, které jsou zapojeny „proti sobě“. CMOS má tu výhodu, že má proti TTL nepatrný odběr. Vzpomeňte si: bipolárními tranzistory musí téct proud mezi bází a emitorem, aby se otevřely, u tranzistorů MOS stačí připojené napětí na řídicí elektrodu Gate, a proud skrz ně teče minimální. K přepnutí proto není zapotřebí, aby obvodem tekly velké proudy, stačí jen napětí.

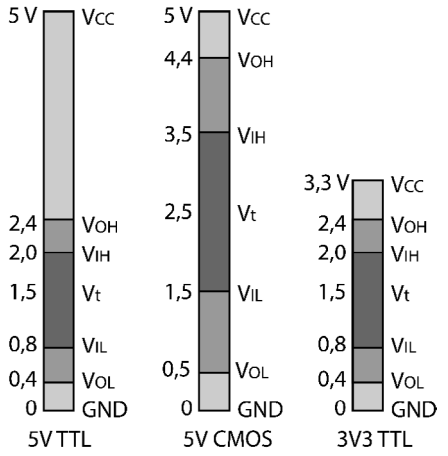
Technologie CMOS měla také podobnou základní řadu jako byla 74xx. U CMOS to byla řada 40xx, ovšem obvody nebyly navzájem kompatibilní a číslování bylo taky jiné (například ekvivalent 7400 byl v CMOS řadě 4011 apod.)

CMOS řada má navíc ještě některé výhody, například široký rozptyl napájecího napětí (až k 15 voltům). Ovšem problém nastal při vzájemném spojování obvodů CMOS a TTL. CMOS mají totiž jiné prahové hodnoty V_{IL}, V_{IH}, V_{OL} a V_{IOH}. U CMOS je při napájení 5 V logická nula na vstupu zaručena od 0 do 1,5 V, logická 1 od 3,5 do 5 V. A tady nastává problém – na výstupu TTL může být při log. 1 pouhých 2,4 V, a to pro CMOS nestačí. Pokud bylo potřeba zapojit výstup TTL na vstup CMOS, používaly se budiče, třeba tranzistory, nebo specializované obvody (převodníky), což celou konstrukci prodražovalo.

Nakonec přišla technologie HCT, která zkombinovala rychlé, levné a nízkopříkonové obvody CMOS a napěťové úrovně kompatibilní s TTL. V řadě 74xx teď můžeme koupit obvody 74HCTxx, které jsou sice technologií výroby CMOS, ale úrovněmi kompatibilní se starými obvody z řady 74xx. Podobné jsou řady AC, ACT, AHC, AHCT, …

Technologie CMOS a její vylepšené varianty je v současnosti nejpoužívanější technologií pro výrobu číslicových integrovaných obvodů. Technologie TTL ale nezůstala zapomenuta, dodneška se obvody v řadě 74xx vyrábějí, a hlavně: dodneška se používá „pětivoltová logika“ TTL.

Postupem času ale bylo potřeba řešit i ten první problém, totiž spotřebu a ztrátový výkon, proměněný na teplo. Logická cesta vedla přes snižování napájecího napětí. Nejprve na 3,3 voltů, později na 2,5 V, pak na 1,8 V, dneska už i 1,2 V a 1 V. Některé součástky, používající 3,3 voltovou logiku (též označovanou 3V3), jsou takzvaně „5 V-tolerantní“, což znamená, že když na jejich vstupy přivedete signál z 5 V obvodu, který bude mít třeba i těch 5 voltů, tak je nespálí. Jiné obvody (například procesor ve známém Raspberry Pi – tak na to pozor, prosím) nejsou 5 V-tolerantní, to znamená, že je vyšší napětí může poškodit. Výhodou 3V3 logiky je to, že má úrovně V_{IL}, VIH, V_{OL} a V_{IOH} shodné s 5 V, takže není problém takové součástky navzájem propojovat (vždy je ale třeba dbát na to, jestli jsou 3V3 součástky na vstupu 5 V-tolerantní). Víc naznačí obrázek:



13.3.1 Propojení CMOS a TTL

Pokud u CMOS dodržíte napájecí napětí stejné jako u TTL, tedy 5 voltů, je s trochou dobré vůle možné tyto dvě technologie kombinovat. Co obnáší ta „dobrá vůle“? V zásadě dvě hlavní pravidla:

- Výstup CMOS můžete připojit na vstup TTL. Ale CMOS mají malé výstupní proudy, takže jedním výstupem CMOS můžete budit vždy jen jeden vstup TTL.

- Výstup TTL může mít v log. 1 úroveň napětí nižší, než je minimální napětí pro CMOS (3,5 V). Nejjednodušší řešení je „zvednout“ výstup TTL pomocí pull-up rezistoru (pojem pull-up si vysvětlíme později) o velikosti cca 10K.

13.4 Operace s bity

Už víme, jak budeme předávat informace. Když budeme chtít do vodiče poslat logickou 1, tedy vysokou úroveň (H), bude to pro nás 5 voltů (přesněji 2 až 5, ale držme se raději té vyšší hodnoty). Logická 0 bude reprezentována stavem bez napětí, popř. nižším než 0,4 V. Tak.

A co s těmi informacemi budeme dělat? Možná vás to překvapí, ale škála možností není zas tak velká. Jediné, co máme k dispozici, jsou takzvané logické funkce. Vše složitější si ale naštěstí můžeme postavit z nich.

Možná je znáte z matematiky, možná je znáte z programování, možná odjinud. Někde se tomu říká i Booleova algebra, a pokud vám to připomíná typ Boolean, jste na správné stopě. Dovolte drobné opakování.

13.5 Booleova algebra, výroková logika

Booleova algebra (kterou nám připomínají typy Boolean nebo Bool z některých programovacích jazyků) je širší pojem, který se v číslicové technice scvrkává na operace s *logickými hodnotami*. Logické hodnoty, též *pravdivostní*, jsou hodnoty logických *výroků*, které nabývají dvou různých stavů: *Pravda* a *Nepravda*. Anglicky True a False. Typicky:

- **Venku prší.** Ano, nebo ne? Pravda, nebo nepravda?

- **Mám deštník**. Mám, nebo nemám? Pravda? Nepravda?

- **Přines mi jídlo.** Tohle není výrok, u kterého můžeme rozhodnout o jeho pravdivosti. Pardon, to jen na ukázku, že ne všechno je výrok.

- **Přinesl jsi mi jídlo.** Tady už rozhodnout lze. Přinesl? Nepřinesl? Ano, nebo ne?

Takovéto výroky jsou výroky *jednoduché*. Jazyk (a každodenní zkušenost“ nás učí, že takovéto jednoduché výroky můžeme spojovat do složitějších složených výroků pomocí spojek, jako je „a“ či „nebo“. *Venku prší a mám deštník*.

Funkci spojek zastávají ve výrokové logice takzvané *logické operátory*. Představme si je podobně jako v matematice – třeba operátor sčítání, operátor násobení, … Můžeme je převést i do podoby *logické funkce* a místo A AND B zapisovat AND (A, B).

Pojďme si dát příklad:

Čtyři sběratelé, Adam, Bořek, Cyril a David (a je jen náhoda, že si je můžu označit A, B, C a D), sbírají různé věci, a to takto:

- Adam sbírá známky a krabičky od zápalek.
- Bořek sbírá mince.
- Cyril sbírá známky i mince.
- David sbírá mince a krabičky od zápalek.

Ke každému pánovi můžeme vytvořit sadu výroků: „Sbírá mince“, „sbírá známky“, „sbírá krabičky od zápalek“, a o každém výroku můžeme říct, zda je pravdivý nebo ne. Tedy: „A sbírá mince“ je výrok nepravdivý, „A sbírá známky“ je výrok pravdivý, a tak dále. Zapíšeme si to do takzvané „pravdivostní tabulky“:

	Sbírá mince	**Sbírá známky**	**Sbírá krabičky**
Adam	ne	ano	ano
Bořek	ano	ne	ne
Cyril	ano	ano	ne
David	ano	ne	ano

Teď otázka: Je tu někdo, kdo sbírá mince a známky zároveň? O kom můžeme říct „Ten sbírá známky i mince“? Jsou to vlastně dva výroky, „sbírá známky“ a „sbírá mince“, a výsledek je pravdivý pouze tehdy, když oba výroky platí. Tedy musí platit, že „sbírá známky = ano“ a „sbírá mince = ano“. Koukneme do tabulky a vidíme: **Cyril je ten pán!**

A teď otázka číslo dva: Je mezi pány někdo, kdo sbírá známky nebo krabičky od zápalek? Jsou to opět dva výroky, „sbírá známky“ a „sbírá krabičky“, ale tentokrát nechceme, aby dělal nutně obojí naráz, stačí nám *alespoň jedna varianta*. Z tabulky vyplývá, že podmínku splňují pánové Adam, Cyril a David. Bořek ne, ten sbírá mince.

Otázka tři: Je mezi pány někdo, kdo nesbírá mince? Tedy o kom platí, že „sbírá mince je nepravda“? Ano, máme tu takového pána, Adama.

Ta první otázka ukázala logický operátor AND (tedy „a“, „i“). Výsledek je pravda tehdy a jen tehdy, pokud jsou jednotlivá tvrzení pravdivá.

Druhá otázka ukazovala operátor OR (tedy „nebo“ ve slučovacím tvaru). Výsledek je pravda tehdy, když je pravda alespoň jedno tvrzení. Pozor, nepleťte si to s tím druhým tvarem, vylučovacím, tedy buď – anebo! Tomu říkáme „vylučovací NEBO“ a značíme ho XOR. K němu se ještě dostaneme. *Mimochodem, víte, že v češtině se mezi těmito tvary rozlišuje? Když napíšu „dáte si brambory, nebo rýži?“ – s čárkou před „nebo“ – znamená to, že platí jen jedna varianta. Buď brambory, nebo rýže. To je „vylučovací, exkluzivní OR“ (XOR). Když ale napíšu „podíváte se se mnou na rezistory, kondenzátory nebo diody“, znamená to, že mohou nastat všechny varianty, a to je to naše „obyčejné“ OR.*

Třetí otázka ukazovala operátor NOT. Jeho výsledek je pravda, pokud je tvrzení nepravdivé.

Programátorům toto určitě připadá jako jasné a samozřejmé. Ale pro jistotu jsem to rozepsal.

AND se nazývá taky „logický součin“, OR „logický součet“ a NOT je „negace“. Čímž se odkazujeme k matematice: AND nahradíme násobením, OR sčítáním, místo „pravda“ použijeme 1, místo „nepravda“ použijeme 0. Někdy se můžete setkat i se zápisem pomocí matematických značek $\vee$ a $\wedge$ (NEBO, A).

Uděláme si tabulku pro dvě proměnné, A a B, a výsledky funkcí AND a OR:

A	**B**	**A OR B**	**A AND B**
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

Máme čtyři možné kombinace, do jakých se vstupy mohou (legitimně) dostat. Z tabulky je vidět, že obě funkce jsou komutativní, to znamená, že když jim prohodíte hodnoty A a B, bude to jedno.

Třetí logická funkce do party je NOT, též logická negace (značí se ¬ před parametrem, nebo vodorovnou čárou nad výrazem) – je to funkce s jedním parametrem a jednou hodnotou, a pokud je parametr 0, je hodnota 1 a obráceně:

A	NOT A
0	1
1	0

Negované signály se značí různými způsoby. Nejčastěji tak, že nad názvem signálu je vodorovná čárka: Q a $\overline{Q}$ Nadtržení se ale zapisuje velmi nepohodlně, proto se někdy setkáte s tím, že negovaný signál bývá označen lomítkem: /Q. Tohoto zvyku se přidržím i já. Další možnost je označit takový signál písmenkem „n“ – nQ, nREADY, nBUSY…

13.6 Logika v číslicové technice

Už jsme si říkali, že v digitální technice používáme jen dva stavy, Pravda a Nepravda, tedy True a False, a řešíme to tak, že buď je někde napájecí napětí (a pak to je 1), nebo tam není, je to spojené se zemí, a pak to je 0.

Neexistuje žádná hodnota „0.5“, třeba že bychom pustili „ne úplně napájecí napětí, ale jen trochu“. Ne, buď 0, nebo 1! Pokud bude na vstupu nejasná hodnota, bude výsledek podivný a chybný.

Asi už tušíte, že jsem o logických funkcích nepsal jen tak zbůhdarma, a máte pravdu. Tyto základní logické funkce mají totiž svou fyzickou podobu – existují číslicové obvody, které dělají přesně tyto operace. Vstupy i výstupy jsou signálové vodiče (opravdové fyzické spoje), a součástka dělá AND, OR, NOT, NAND, NOR, ...

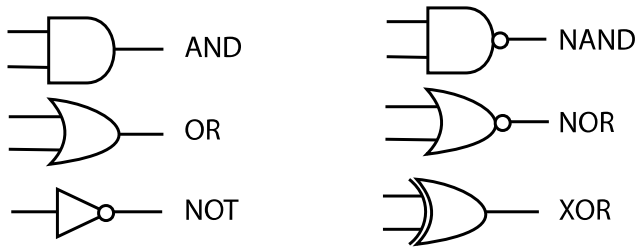
Moment, NAND? NOR?

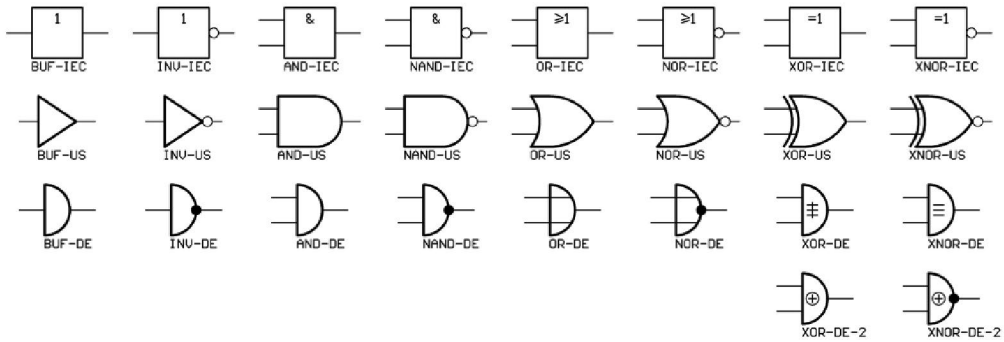
Ano, ve světě číslicové techniky jsme si rozšířili základní typy logických operací pro dva vstupy na 4. K AND existuje NAND, tedy „negovaný AND“, a k OR existuje NOR, čili „negovaný OR“. Zkrátka za výstupem je ještě zapojený invertor, takže tabulka hodnot vypadá takto:

A	**B**	**A OR B**	**A NOR B**	**A ANDB**	**A NAND B**
0	0	0	1	0	1
0	1	1	0	0	1

A	B	A OR B	A NOR B	A AND B	A NAND B
1	0	1	0	0	1
1	1	1	0	1	0

Každá logická funkce (hradlo) má i vlastní schematickou značku. Ty jsou shrnuty v obrázku (včetně hradla XOR, k němuž se ještě dostaneme). Schematické značky, co používám, jsou podle normy ANSI. V ČSSR se koncem osmdesátých let začala prosazovat evropská norma IEC, kde všechny vypadají jako obdélníčky se symbolem; setkáte se s oběma typy schematických značek. Jedno mají společné: Negovaný vstup nebo výstup se značí kroužkem:





Ta první funkce, označená jako „BUF“ (z anglického Buffer, česky se pro tuto součástku používá slovo „budič“), je vlastně něco-jako-invertor, který neinvertuje. Když je na vstupu 0, na výstupu je taky 0. Když je na vstupu 1, na výstupu je taky 1. Možná si říkáte, že taková součástka je úplně na... zbytečná, ale přeci jen se někdy hodí: v případě, že potřebujeme signál přivést k více dalším zařízením, hodí se posilovač – budič. Takové součástky většinou dokáží spolehlivě dodat dostatek proudu k tomu, aby signál mohl být přiveden do několika různých obvodů najednou. Pro klasické obvody TTL platí, že k jejich výstupu můžete připojit maximálně 10 vstupů dalších obvodů TTL. Pro CMOS je tento počet mnohem nižší – a tam se hodí právě budič.

Mimochodem, tomu číslu, které říká, kolik vstupů může být připojeno na jeden výstup, se říká logický zisk a je dobré ho znát a vzpomenout si na něj pokaždé, když jeden výstup budete připojovat na víc vstupů. Je to vlastně poměr mezi proudem, který může téct výstupem jednoho hradla, a proudem, který musí téct vstupy dalších hradel. Pokud je výstupní proud desetkrát větší než vstupní, můžete připojit deset vstupů na jeden výstup a máte logický zisk 10.

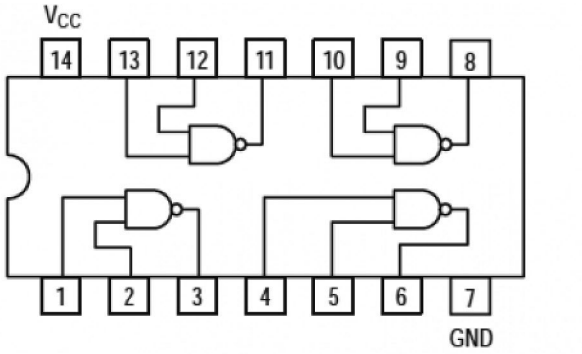
13.7 U-káz-ka! U-káz-ka!

Dobrá, pojďte si zazapojovat.

Vezměte si obvod 7400. Teď už víte, že jeho označení bude třeba SN74ALS00, nebo SN74HCT00, nebo klidně i MH74LS00, nebo jen 74HC00, nebo … v podstatě cokoli z kombinace *xx74xx00xx*. Prostě nějaký takový obvod vezměte a zapojte ho do nepájivého pole stejně, jako jste to dělali v první kapitole. Pamatujete? Pěkně doprostřed, a podívejte se na něj – klíč, ta značka na pouzdru, ať je hezky vlevo, ať máme vývod číslo 1 vlevo dole.

Pokud si nepamatujete, nalistujte si první kapitolu, zapojování blikače – *orientaci obvodu určíte podle popisku a výlisku na pouzdru, pokud máme popisky správně orientované a výlisek vlevo, tak vlevo dole je vývod 1, vedle něj 2, a číslování pokračuje ve spodní řadě doprava. Na konci pokračuje proti směru hodinových ručiček nahoře, zprava doleva.*

Tak. Obvod bychom měli. Co s ním? Jak zjistíte, co máte kam připojit? No, schválně, kdo si to pamatuje? Správně, podíváte se do datasheetu. Ale protože jsem na vás hodný, tak vám sem ten nejdůležitější obrázek zkopíruju. To je on:



Vidíte, jak je uspořádaný obvod uvnitř: jsou tam čtyři hradla NAND, zapojená na vývody 1-2-3, 4-5-6, 13-12-11 a 10-9-8. Vývod 7, označený GND, je zem (Ground), tedy pro nás záporný pól napájecího zdroje, vývod 14 je Vcc, neboli napájecí napětí.

Jak poznáte, který vstup je A a který B? Nepoznáte. Je to přeci úplně jedno! U hradla NAND můžete vstupy zaměnit.

Pamatujte si: Napájecí napětí se ve schématech značí buď Vcc, nebo Vdd, nebo prostě „+ 5 V“; zem je GND, Vss, Vee, nebo „0“. Proč? No, protože kladný pól, tedy napájecí napětí, bývá spojeno s kolektorem tranzistorů u obvodů TTL (odtud Vcc – Voltage Collector), případně s elektrodou Drain u MOS (Vdd – Voltage Drain). Analogicky Vee bude souviset s emitorem (TTL), Vss s elektrodou Source (MOS). Ale pro nás je Vcc totéž jako Vdd, a Vss budiž totožné s Vee i s GND.

Napájecí napětí se tedy připojuje stejně jako u blikače. Podíváme se na funkci jednoho hradla. Třeba toho, jak má vstupy 1 a 2 a výstup 3. Na výstup (nožička číslo 3) si připojte LEDku (samozřejmě přes rezistor 330 ohmů!). Vstupy (1 a 2) propojte se zemí (se záporným pólem napájení). Oba.

Co s těmi ostatními hradly v pouzdru? Teď nic, ale v reálných konstrukcích je dobré připojit jejich vstupy k některému pólu napájení, ideálně oba k + 5 V. Vývody mohou zůstat nezapojené, pokud dané hradlo nepoužíváte. Proč? No, pokud ty nepoužité necháte úplně nezapojené, mohou se vlivem rušení náhodně překlápět, čímž budou zvyšovat odběr a přenášet rušení do napájecího vedení.

Když teď zapnete napájení, měla by LED svítit. Oba vstupy spojené se zemí znamenají, že na vstupech jsou hodnoty 0, a výsledek 0 NAND 0 = 1.

Když teď oba vstupy připojíte na kladný pól napájení, LED zhasne.

Otázka: Můžete to dělat při zapnutém napájení? Odpověď správná: Ne, nikdy to nedělejte! Odpověď realistická: Sice to někteří někdy dělají a nic se nestane, ale nezvykejte si na to a fakt to radši nedělejte, protože občas se holt něco stát může, a hezké to nebude…

Zkuste si přepojováním vstupů 1 a 2 nasimulovat všechny možné kombinace vstupních hodnot a podívat se, jaký bude výstup.

Pokud vám LED nesvítí vůbec, máte problém. Zkuste ji nejprve přepojit místo na vývod číslo 3 na napájecí napětí. Samozřejmě přes ten rezistor. Pokud stále nesvítí, může to mít tři příčiny: Máte LED obráceně (vzpomeňte si na kapitolu o diodách), máte LED spálenou, nebo vám nefunguje napájení.

Pokud vám LED svítí naopak pořád, i když zapojíte vstupy 1 a 2 na napájecí napětí, máte pravděpodobně vadný obvod 7400. Zkuste to popřepojovat jinak, třeba na vstupy 4, 5 a výstup 6.

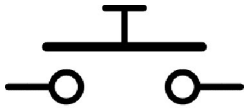
Jde ty vstupy udělat jinak, než pomocí drátků?

Jasně že to jde. Jako vstup můžeme použít třeba tlačítko, nebo přepínač.

13.8 Tlačítko a přepínač

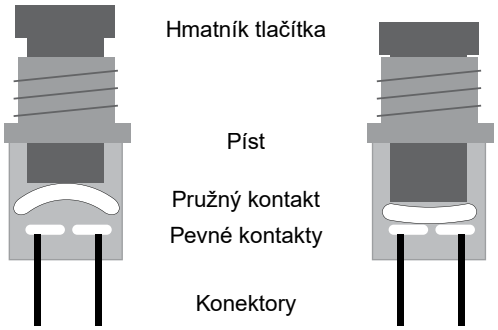
Tlačítka, přepínače, vypínače, spínače a jiné zapínače a odpínače patří hned vedle zásuvek a žárovek k základním elektrotechnickým komponentám, které zná úplně každý z každodenní zkušenosti. Všechny tyto součástky slouží k tomu, aby buď přerušily nebo naopak spojily obvod, a to dočasně, po dobu stisknutí tlačítka, nebo trvaleji, do dalšího přecvaknutí přepínače…

Hodně o jejich funkci napoví schematické značky:



SW1

Toto je schematická značka tlačítka. Všimněte si, že vodič je přerušený, a je nad ním ploška. Přesně tak vypadá tlačítko uvnitř: jsou tam dva póly a nad nimi pružný plíšek, na který tlačí plastový hmatník. Jakmile zatlačíte prstem, tak se plíšek při určitém tlaku prohne (s typickým cvaknutím) a oba póly propojí. Když tlak přestane, plíšek se opět prohne na druhou stranu a obvod zase rozpojí.



Tlačítka, která budete používat nejčastěji, jsou malá mikrotlačítka, která mají čtyři vývody. Vždy dva a dva jsou spojeny dohromady, a mezi těmito dvěma páry je právě samotný mechanismus tlačítka.

Pokud si nejste jistí, které dva vývody jsou spolu propojené a které ne, tak je nejjednodušší způsob si to prostě změřit. Vážně, doporučuju…

Tlačítko tedy spíná pouze po tu dobu, kdy ho někdo tiskne, nespisovně mačká. Znáte třeba z domovního zvonku. Druhá možnost je, že tlačítko za normálních okolností vede, při stisku

se naopak rozpojí, ale taková tlačítka jsou méně častá. První typ se označuje NO (Normally Open), druhý typ NC (Normally Closed), podle toho, jestli je tlačítko samo o sobě, tj. bez vnější síly, otevřené a rozpojené (NO), nebo zavřené, propojené (NC).

Pro naše experimenty s LEDkami je tlačítko dostatečně dobré zařízení, ale jakmile začnete stavět něco s rychlými procesory, zjistíte, že s tlačítkem může být problém. Totiž – člověk jednou zmáčkne, a procesor zachytí třeba dva nebo tři po sobě jdoucí stisknutí.

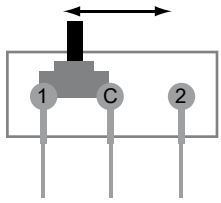
Tomuto efektu se říká **zákmit**, a je způsoben právě tím, jak je mechanicky tlačítko uspořádané. Plíšek má totiž nějakou svou pružnost, a když se „přecvakne“ do pozice *stisknuto*, tak si ještě několikrát zavibruje, a tím způsobí několik rozpojení a spojení. U pomalých obvodů to nevadí, ale rychlý jednočip napočítá klidně i několik stisknutí, a to může přinášet problémy.

Zákmity tlačítek lze řešit několika způsoby. Mechanicky – tedy použitím bezzákmitových tlačítek. Ale ta jsou velmi drahá. Pak elektricky – přidáním kondenzátoru, který rychlé pulsy odfiltruje, a Schmittova obvodu (k tomu se ještě dostaneme), který upraví signál. Nebo pomocí monostabilního klopného obvodu. No a nakonec softwarově – nejjednodušší řešení je při každém stisknutí tlačítka počkat třeba tři milisekundy (to je empiricky zjištěná hodnota pro běžná mikrotlačítka), zkontrolovat, zda je tlačítko stále stisknuté, a teprve poté dělat nějakou akci. Anglicky se těmto postupům říká **debouncing** – občas se s tím setkáte, tak abyste věděli…

Přepínač funguje podobně, ale „pamatuje si“ poslední stav.

Uvnitř bývá mechanická propojka, která vždy spojí prostřední vývod (společný, C – Common) s jedním z krajních vývodů. Mechanismus je navržen tak, aby k přepnutí došlo co nejrychleji, skokově, aby se minimalizovala doba, kdy jsou oba vývody odpojené a aby se vyloučilo, že budou zapojené oba naráz. (Existují i přepínače se třemi polohami, kdy uprostřed jsou oba vývody odpojené.)

Přepínač nemusí být jen páčkový, jak jej známe ze starších elektronických výrobků. Může být třeba posuvný.





Tento přepínač se dá, při troše dobré vůle, použít i do nepájivého kontaktního pole. Bohužel, rozteč vývodů u některých laciných přepínačů může být pouze 2,5 mm, ne 2,54, a tak lehce nesedí, což je škoda. Snažte se proto sehnat takové, které mají rozteč 2,54 mm.

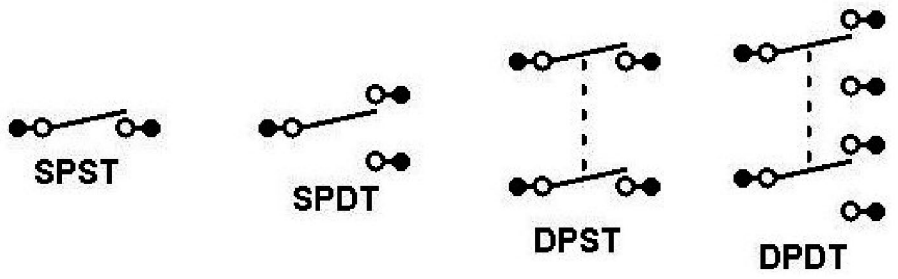
Pokud u přepínače jeden krajní vývod vynecháme, dostaneme spínač (popřípadě vypínač, ono je to totéž). Velmi časté jsou spínače a přepínače **dvoupólové**, tedy takové, kde jsou vlastně dva spínače / přepínače v jednom pouzdru, jen páčka je společná. Abychom se v tom trochu vyznali, tak jsou zavedené zkratky:

SPST – Single pole (jednopólový), single throw (s jedním vývodem, tedy spínač)

SPDT – Single pole (jednopólový), dual throw (s dvěma vývody, tedy přepínač)

DPST – Dual pole (dvoupólový), single throw (s jedním vývodem, tedy spínač)

DPDT – Dual pole (dvoupólový), dual throw (s dvěma vývody, tedy přepínač)

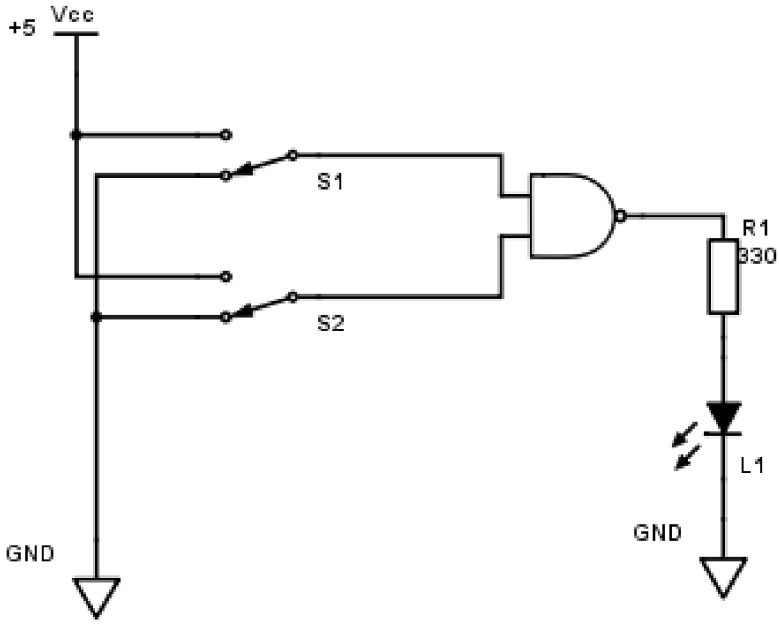


13.9 Pull Up a Pull Down

Fajn, už víte, jaké možnosti máte. A teď tedy zpátky k našemu problému, totiž vyřešit nějak nastavování jedniček a nul: co zapojit? A jak?

Klasické přepínače a spínače by šly použít, ale museli byste si k nim připájet vývody, co jdou zapojit do nepájivého kontaktního pole (NKP).

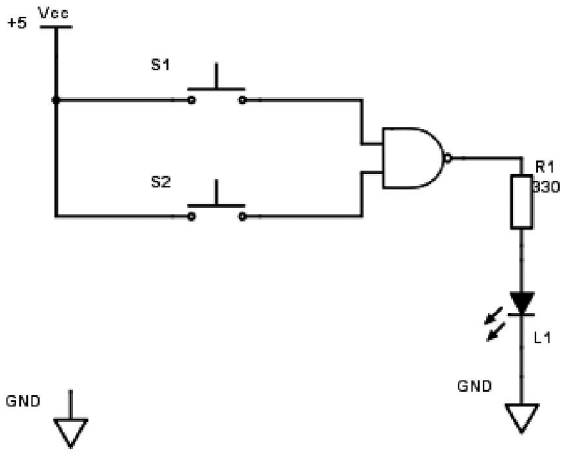
Můžete použít tahový přepínač, který jde sice do kontaktního pole zasunout, ale nedrží tam moc dobře, protože mívají krátké nožičky. Ale šlo by to: Na vstup hradla připojíte prostřední vývod (C), jeden krajní zapojíte na zem, druhý na Vcc, a bude to. Podle toho, jak přepnete, bude vývod C přepínače spojen buď se zemí, nebo s napájením, a bude na něm tedy buď log. 0, nebo log. 1. A když budete přepínat rychle, tak minimalizujete i okamžiky, kdy na vstupu nebude ani zem, ani napájecí napětí, což je pro nás *zakázaný stav*.



Nebo můžete použít mikrotlačítko, které do NKP sedne docela dobře, jenže neumí přepnout mezi dvěma póly. (Jsou taková tlačítka, spíš „přepínače s pružinou“, která se vždy vrací do výchozí polohy, ale ty teď neuvažujme.) Jak to uděláte?

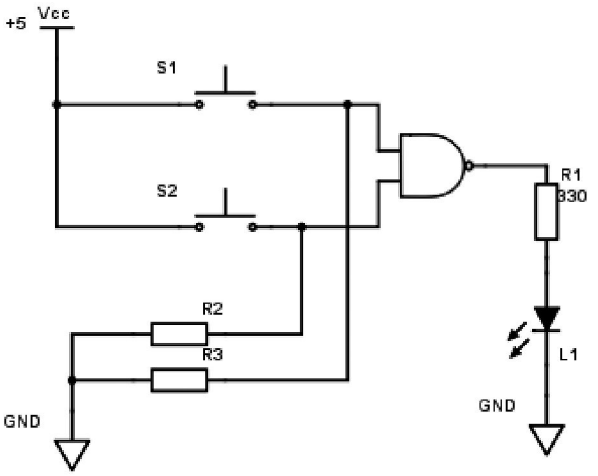
Tak, v první řadě – zapojíte tlačítko mezi vstup hradla a napájecí napětí. Pokud bude tlačítko stisknuté, propojí vstup s Vcc a na vstupu bude tedy logická 1. Pokud bude tlačítko rozpojené, tak… Co? No, v takovém případě nebude na vstupu nic konkrétního. Záleží na technologii obvodu, jak si s takovým stavem poradí, ale upřímně říkám, že třeba u obvodů CMOS stačí přiblížit ruku a stav se změní (vzpomeňte si na pokus s tranzistorem JFET)… Takže by bylo fajn tam mít po zbytek času zem.

A jestli mi nevěříte, zkuste si to sami. Použijte třeba 74HC00 (tady je to HC důležité, s obvodem LS, ALS apod. to fungovat nebude), zapojte LED na výstup hradla, jeden vstup připojte na + 5 V, druhý nechte nezapojený a podívejte se co se stane, když budete kolem toho nezapojeného šátrat prstem. Když k nezapojenému vstupu připojíte jeden konec vodiče a druhý necháte bimbat prostorem, stvoříte navíc anténu, citlivou i na statickou elektřinu, a nebudete se stačit divit, jak se bude LED chovat. Tedy až do té chvíle, kdy se zapomenete, poškrábete se zamyšleně na triku, tím si vytvoříte větší statický náboj, a když pak ruku přiblížíte k téhle anténě, tak se ozve suché zapraskání, které vám zároveň oznámí, že jste pravděpodobně právě zničili celý integrovaný obvod.



Říkáte si *dobře, tak to propojím se zemí, aby byl klid a žádné rušení?* To není moc dobrý nápad. Co by se stalo? Dokud není tlačítko stisknuté, tak je vstup propojen se zemí a je na něm log. 0. Jakmile tlačítko zmáčknete, tak se propojí napájecí napětí se zemí, vytvoří se zkrat, kterým poteče tolik proudu, kolik jen zdroj dokáže dát a kolik vodiče snesou, ostatní obvody zůstanou bez proudu (většina poteče zkratem) a je zle. Takhle tedy ne!

Dělá se to tedy tak, že mezi vstup a zemi zapojíte rezistor, třeba 10k. Pokud je tlačítko rozpojené, tak je vstup obvodu skrz něj připojen k zemi, a je tam tedy nula. Když stisknete tlačítko, tak sice vznikne zkrat, kterým poteče I = U / R = 5 / 10000 = 0,5 mA, ale zároveň dostanete na vstup požadovaných (skoro) 5 V.



⌀ https://eknh.cz/7400

Této technice se říká **pull-down rezistor** – jako že „táhne vstup dolů, k zemi, k nule“. Když to zapojíte obráceně, tedy tlačítko na zem a rezistor na napájecí napětí, stvoříte **pull-up**, tedy zapojení, kde rezistor „táhne vstup nahoru“ – myšleno k logické 1. Tento model je velmi častý a setkáte se s ním u spousty zapojení. Ovšem funkce tlačítka pak bude obrácená: při stisknutí spojí vývod na log. 0, při puštění log. 1.

13.10 Pomalé tlačítko

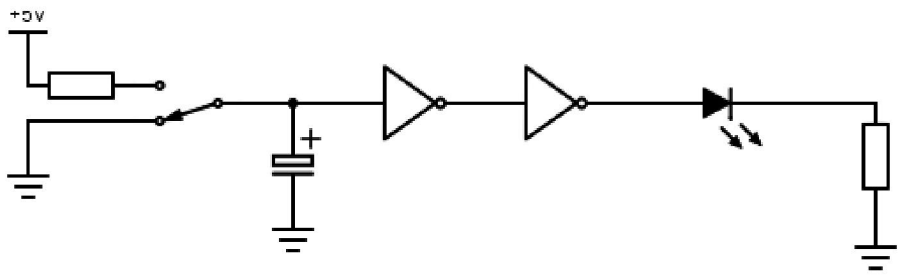
Představte si situaci, kdy máte nějaký obvod, startovaný přepínačem, a chcete, aby nastartoval ne hned, ale až když je přepínač přepnutý nějakou chvíli, třeba dvě sekundy. Potřebujete tak eliminovat náhodné neúmyslné stisknutí. Obsluha musí přepínač přepnout a držet nějakou dobu přepnuté.

Jak na to? V první řadě přemýšlejte: jaká součástka má nějakou souvislost s časem? Co můžeme použít, když potřebujeme na něco počkat?

Odpověď je: **Kondenzátor**. Ten se nabíjí, a napětí na něm pomalu roste s tím, jak se nabíjí. Takže jej necháme pomalu přes rezistor nabíjet, a jakmile hodnota napětí překročí 2 volty (což je rozhodovací hranice pro logickou 1), tak sepneme obvod – pro náš test třeba LED.

Jak zajistíte to přepnutí? Dáte za sebe dva invertory, to je takový jednoduchý trik. Na výstupu bude stejná logická hodnota jako na vstupu, ale „ošetřená“ – tedy logická 1 bude téměř napájecí napětí, ne usmolené 2 volty.

No a přepnutí přepínače do druhé polohy vybije kondenzátor k zemi. Nějak takto.



Čím větší odpor na vstupu a kapacita kondenzátoru, tím delší bude prodleva. Když zvolíte rezistor kolem 100 kΩ a kondenzátor třeba 10 μF, tak bude prodleva necelou sekundu. Jenže!

Jenže záleží hodně na typu invertoru. 74ALS04 se bude chovat jinak než třeba 74HCT04, a svými parametry bude ovlivňovat čas. Doporučuju opět metodu pokus-omyl.

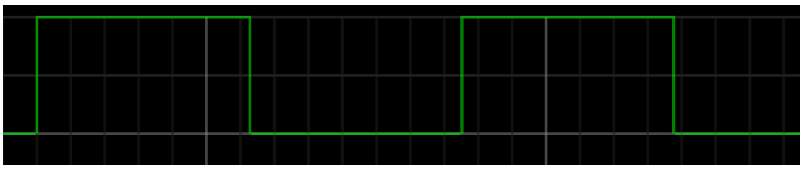
Ale ještě počkejte a zamyslete se… Když píšu, že se kondenzátor pomalu nabíjí od 0, co to znamená? No, že napětí roste plynule, a v určitou chvíli se ocitne v *zakázaném pásmu* mezi 0,8 a 2 V. V takovou chvíli může invertor dělat naprosto cokoli, klidně se může rozkmitat, nebo se může rovnou přepnout. Lze to nějak vyřešit?

No, kondenzátor těžko přesvědčíte, aby se po 0,8 V nabil hned na 2 V. Musíte to vyřešit na straně toho invertoru. Naštěstí je řešení docela jednoduché, a říká se mu

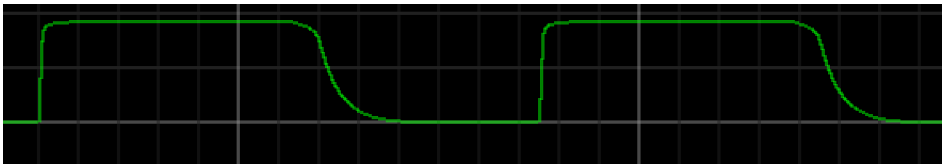
13.11 Schmittův obvod

O kousek dřív jsem řešil, že číslicové obvody potřebují jasnou hodnotu: méně než V_{IL} je nula, víc než V_{IH} je jednička, a cokoli mezi tím je zakázané. To se prostě nesmí!

Jenže – co když se to stane? Jak? No, je hned několik možností. Nejčastěji se stane, že někde na vstupu je nějaká kapacita. A tím nemyslím odborníka kapacitu, ale kapacitu – kondenzátor. Někdy záměrně, někdy náhodou – třeba jsou vodiče, co vedou signál, příliš dlouhé a příliš blízko u sebe, takže mezi nimi vznikne takzvaná *parazitní kapacita*. Z experimentu s LEDkou a kondenzátorem si určitě pamatujete, že kondenzátor v obvodu zpomalí vypínání a zapínání. Z dokonale čistého signálu „0 nebo 1“, který vypadá nějak takto (říkáme, že má „strmé hrany“)



se stane signál, který bude pomalu nabíhat a pomalu klesat:



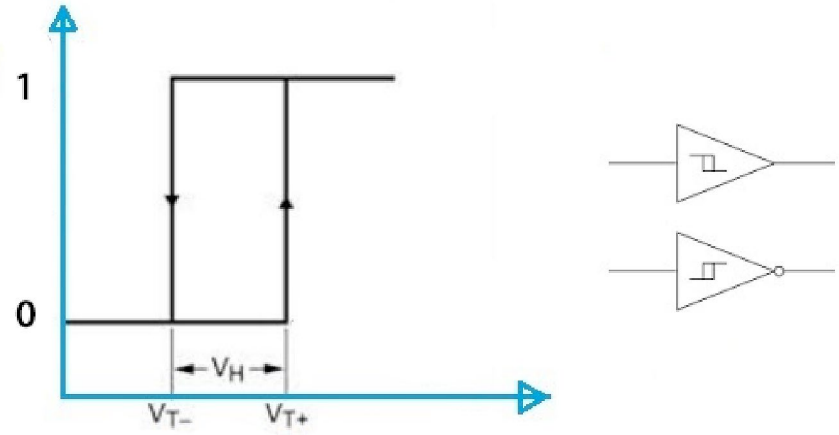
Všimněte si, že v tomto případě signál stráví významnou dobu někde uprostřed mezi minimem a maximem. A to není dobře. To je přímo problém, protože během té doby obvod nedokáže vyhodnotit, jestli je na vstupu 0, nebo 1, a jeho chování je nejisté. On si s tím nějak nakonec poradí,

ale může se také stát, že bude třeba „pootevřený“, takže jím poteče zbytečně velký proud, na jeho výstupu bude zase zakázaná úroveň, další obvod to taky vyhodnotí špatně...

Zkrátka – číslicové obvody vyžadují, aby to přepnutí z jednoho stavu do druhého bylo co nejrychlejší, hrany co nejstrmější. Pak funguje dobře. Ale co když to prostě nelze zajistit?

Pak musíme použít součástku, která má na vstupu takzvaný **Schmittův klopný obvod**. To je speciální úprava vstupu obvodu, která má rozhodovací úrovně mezi 0 a 1 posunuté tak, že se překrývají. To znamená, že když napětí na vstupu pomalu roste, tak třeba až do 3 V je považované za logickou 0, a pak je to logická 1. Ovšem pokud napětí klesá od 5 V dolů, tak je stále považované za logickou 1, až když klesne třeba pod 2 V, tak je považované za logickou 0.

To, že úroveň pro přepnutí z 0 na 1 je vyšší, než pro přepnutí z 1 na 0, způsobí, že obvod nebude při žádné úrovni napětí zmatený, ale vždy v přesně definovaném stavu, v závislosti na předchozím stavu. Této vlastnosti, kdy pro přepnutí v jednom směru platí jiné podmínky, než pro přepnutí ve směru opačném, se říká **hystereze**. Když si to nakreslím do grafu, vznikne typický obrazec, kterému se říká **hysterezní smyčka**. Hysterezní smyčka má tak charakteristický tvar, že se u obvodů, které mají takto ošetřené vstupy, maluje i do schematické značky.



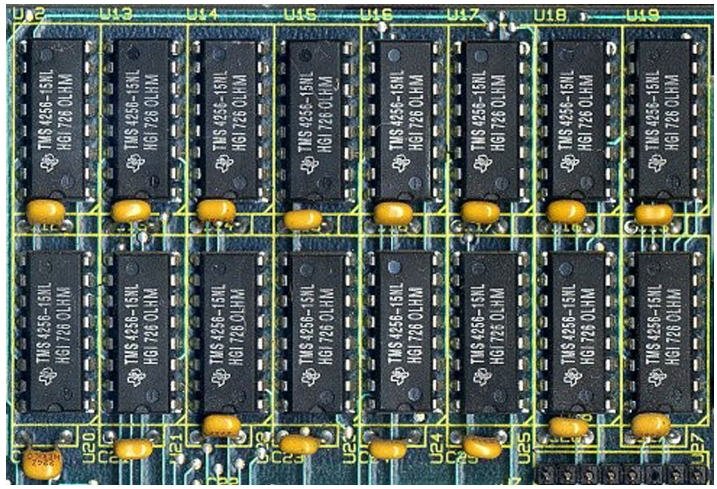
Rozdíl mezi prahovým napětím V_{T+} a V_{T-} udává takzvanou šířku hysterezní smyčky. Čím širší, tím bude obvod odolnější proti náhodným výkyvům, ale zase bude vyžadovat jednoznačnější úrovně užitečného signálu.

⌀ https://eknh.cz/schm

13.12 Blokovací kondenzátor

Přeci jen je ale jedno místo, kam se kondenzátory zapojují schválně a záměrně a vždy.

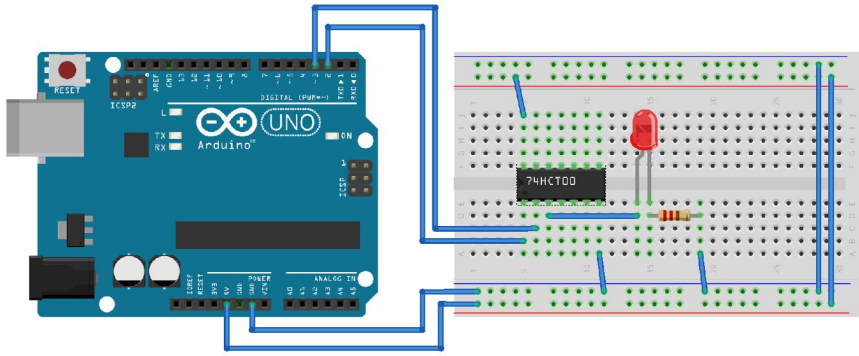
Už víte, že když se číslicový obvod přepíná z jednoho stavu do druhého, může se stát, že přepnutí není ideální. Že zkrátka na malý okamžik mohou být otevřeny obvody pro logickou 0 i pro logickou 1, a po ten malý okamžik teče skrz obvod velký proud z napájení do země. Projeví se to prudkým skokovým zvýšením spotřeby. Čím větší frekvence přepínání, tím častěji se takové proudové špičky objeví. To znamená, že na napájecím napětí vznikají nevítané pulsy, které způsobují rušení, a to se přenáší do dalších obvodů. Proto se paralelně k napájecím vývodům připojuje kondenzátor o dostatečné kapacitě, který podobné špičky dokáže pokrýt a zachytit. Používá se nejčastěji kondenzátor o kapacitě 100 nF. Zapojuje se co nejblíž integrovanému obvodu, ideálně „hned vedle“. Když se podíváte na desky plošných spojů u starších počítačů, uvidíte to názorně:



U každého obvodu je co nejblíž zapojen blokovací kondenzátor – na fotografii jsou hned pod integrovanými obvody.

13.13 Buzení z Arduina

Jen takový tip… Když se vám nechce zapojovat všechny ty přepínače, tlačítka a rezistory, tak můžete použít zase Arduino. Vstupy obvodu si připojíte na nějaké z digitálních výstupů v Arduinu, napájecí napětí vezmete taky z Arduina, a nastavování logických úrovní necháte na programu…



A pokud jste opravdový programátor, tak vynecháte i LEDku, místo toho zapojíte výstup z hradla na volný digitální vstup Arduina, a otestujete si fungování úplně čistě Arduinem.

— 13 „Jak naučit kámen počítat“

14 Kombinační logika

14 Kombinační logika

Je fajn, že nám jedno hradlo dělá jednu funkci, ale to není moc zajímavé. Zajímavé to začne být, až když si ty vývody začnete všelijak spojovat. U obvodů 74xx platí, že k jednomu výstupu můžete připojit až deset vstupů. Výstup ale nikdy nesmíte připojit přímo k jinému výstupu. Pokud chcete dva výstupy nějak spojit dohromady, (skoro) vždycky musíte použít hradlo.

Proč nesmíte spojit výstupy přímo spolu? Ještě se k tomu vrátím, ale teď dovolte aspoň stručné vysvětlení: Když budou oba v logické 1 nebo oba v logické 0, tak by to teoreticky nemuselo vadit. Jenže co když výstup jednoho hradla bude logická 1 a výstup druhého logická 0? Jaký bude výsledek? „Logická polovina" neexistuje… Ve skutečnosti bude výsledek takový, že zbůhdarma poteče proud skrz jedno hradlo do druhého a výsledek bude nejasný.

Hradlo je součástka, která *realizuje* logickou funkci. Základní logické funkce jsou, znovu připomínám, AND, OR a NOT. Jejich vhodnou kombinací lze získat všechny logické funkce. V praxi se používá hradlo NAND (popřípadě NOR), z nějž lze poskládat všechny ostatní. Vážně, hned si to ukážeme.

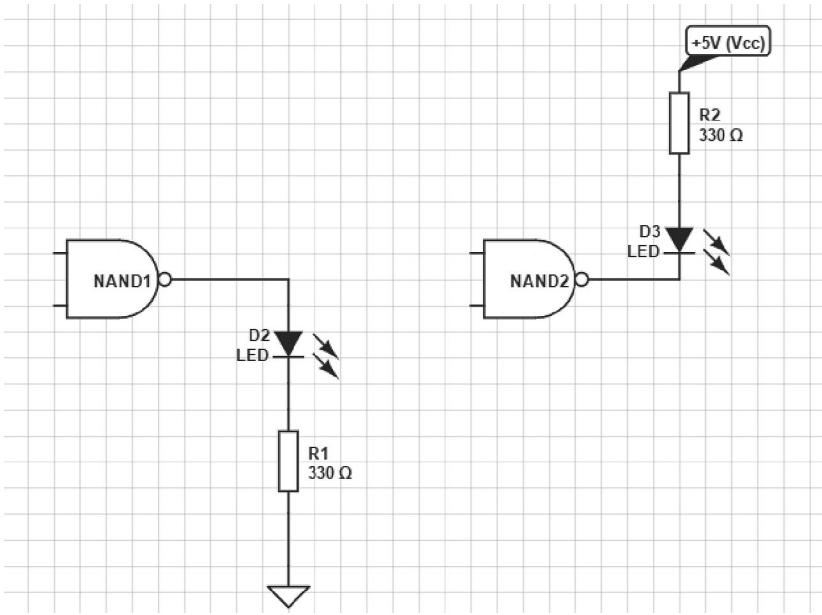
Proč *hradlo*? Hradlo je pojem z železnice. Do jeho přesné definice se pouštět nebudu, to by mě případní železničáři mezi vámi utloukli. Zůstanu u jednoduchého lidového popisu: Hradlo je *ten semafór*, který říká v zásadě buď „volno", nebo „stůj". Obvod s funkcí AND vlastně může hrát roli takového „hradla" pro logický signál. Na jeden vstup, nazvu ho datový, přivedu kýžený signál, a druhým řídím. Když je řídicí vstup 0, je na výstupu taky nula. Když je na řídicím vstupu jednička, je na výstupu to, co je na datovém vstupu.

Zkusme si to na nepájivém poli:

Jedno tlačítko pro nás bude „signál", druhé tlačítko „povolení". Ale musíme počítat s tím, že výsledek je negovaný, takže pokud je „povolení" nula, bude na výstupu vždy 1, pokud je „povolení" 1, bude na výstupu negovaný signál.

Chcete poradit trik, jak diodou zobrazovat negovaný výstup? Zapojte ji ne k zemi, ale k napájecímu napětí! Samozřejmě je nezbytné, aby byla správně polarizovaná… Pak bude svítit, když na výstupu hradla bude logická 0.

Zkuste si to takto přepojit, uvidíte, že teď bude LED svítit přesně opačně, tedy ve stavu 0.



A když už v tom budete, podívejte se, co se stane, když použijete jen jedno tlačítko, a to připojíte:

a. na oba vstupy NAND najednou

b. na jeden vstup NAND, a na druhý zapojíte napevno zem (GND)

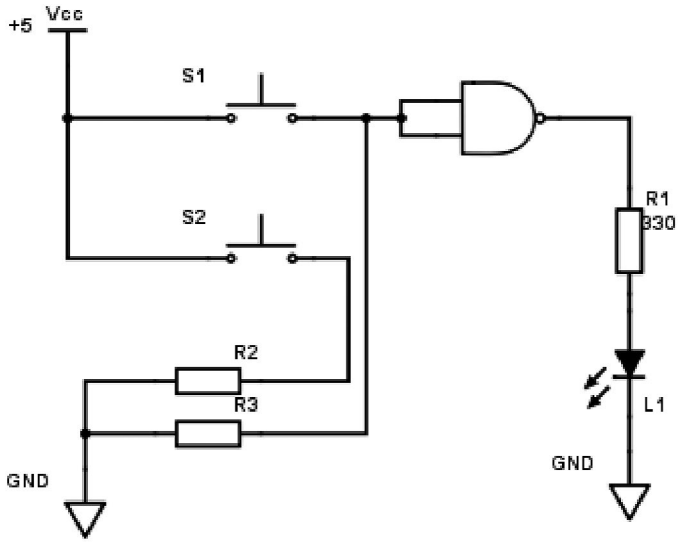
c. na jeden vstup NAND, a na druhý zapojíte napevno napájecí napětí (Vcc)

14.1 De Morganův zákon

Ve skutečnosti, věřte mi nebo ne, si v celé číslicové technice vystačíte s jedním jediným typem hradla! Vše ostatní si z něj dokážete poskládat. Tímto hradlem je nejčastěji NAND (mohl by teoreticky být i NOR). Všimněte si tabulky obvodu NAND a přemýšlejte: Co se stane, když spojíte oba vstupy dohromady a na oba přivedete stejnou hodnotu? Jak se bude obvod chovat? Podívejte se na první a poslední řádek tabulky – nepřipomíná vám to nic?

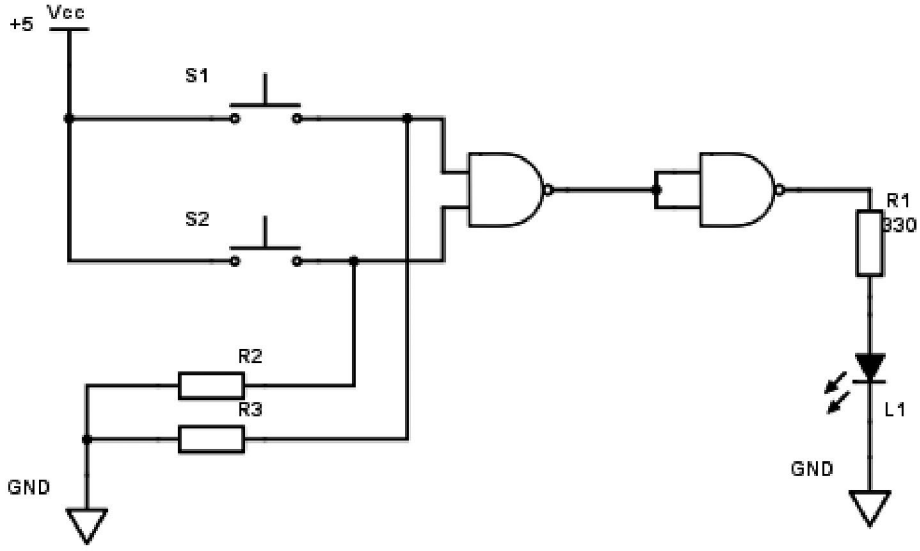
No jasně, získáte tak invertor!

Zkuste si to:



Tlačítko S2 je teď úplně „plonkové“, bez funkce (jen zahřívá rezistor R2). Tlačítko S1 obsluhuje oba vstupy.

A když zapojíte takový invertor na výstup hradla NAND? Tedy použijete dvě hradla…

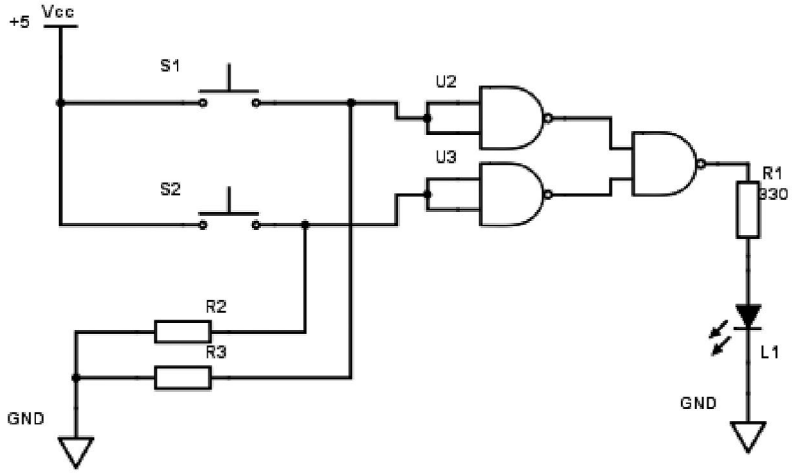


Zapište si zase výsledky do tabulky:

A	B	Y
0	0	
0	1	
1	0	
1	1	

Jistě už víte, co to je za funkci.

A co když invertujete vstupy hradla NAND? Jaký bude výsledek? (Napovím: všimněte si v tabulce podobnosti mezi sloupcem NAND a dalšími sloupci... A všimněte si, že při invertování A a B dostanete tutéž tabulku, ale pozpátku...)



A naměřené výsledky hned do tabulky:

A	B	Y
0	0	
0	1	
1	0	
1	1	

⌀ https://eknh.cz/mrgn

Tomu, co jste si teď prakticky zkusili, zapsali a zjistili, se vznešeněji říká *de Morganovy zákony*, a formulují se takto: Negace součinu je rovna součtu negací, negace součtu je rovna součinu negací. Když se vrátíme k matematickému zápisu:

¬ A ∧ ¬ B = ¬ (A ∨ B) – - NOT A AND NOT B = NOT (A OR B)

¬ A ∨ ¬ B = ¬ (A ∧ B) – - NOT A OR NOT B = NOT (A AND B)

My si z toho můžeme vzít jednoduché ponaučení: z NAND se invertováním vstupů stane OR, z NOR se invertováním vstupů stane AND.

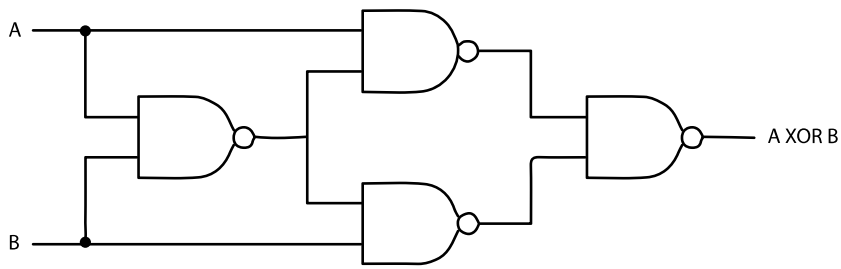
14.2 XOR

A kde máme tu zmíněnou funkci „exkluzivního OR", neboli XOR? No, dala by se poskládat z hradel NAND. Schválně…

Nejprve si udělejme tabulku. Slovní definice zní: Výstup je v log. 1, pokud jsou hodnoty na vstupech různé (tedy buď A, nebo B jsou v log. 1, a ten druhý v log. 0). Pokud platí, že A = B, tak je na výstupu 0. Tabulka bude vypadat takto:

A	**B**	**Y**
0	0	0
0	1	1
1	0	1
1	1	0

Zapojení je takovéhle:



Po pravdě říkám, že bych ho dokázal odvodit, ale raději si to zapamatuju. A vy si to zapamatujte taky. Máme tu čtyři hradla NAND. První zleva má na výstupu 0, pokud jsou na vstupu A = B = 1. Tedy A NAND B. Označme si tuto mezihodnotu jako C.

Horní a dolní hradlo NAND pak počítá hodnoty A NAND C a B NAND C. Označme si je třeba AX a BX.

Poslední hradlo pak udělá AX NAND BX.

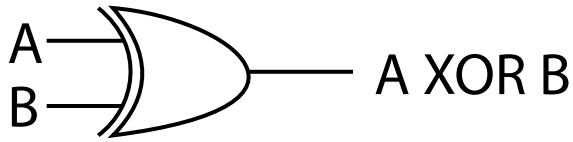
Zkuste si to zapojit. Tlačítka máte, v obvodu 7400 máte i čtyři hradla NAND, zapojte, zkuste, uvidíte sami!

⌀ https://eknh.cz/xor

Kompletní tabulka:

A	B	A NAND B (C)	A NAND C (AX)	B NAND C (BX)	AX NAND BX (Y)
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	0

Výsledek si můžeme „schovat“ zase do jednoho symbolu:



Když se nad tím tak zamyslíte – kolik takových funkcí pro dvě proměnné může být?

14.3 Logické funkce dvou proměnných

Dvě vstupní proměnné mohou nabývat celkem čtyř kombinací: 00, 01, 10 a 11. Každé téhle kombinaci odpovídá nějaká hodnota na výstupu, kterou můžeme zapsat jako čtyřbitové slovo – jako bychom četli poslední sloupec pravdivostní tabulky shora dolů.

Pro OR to je 0111, pro NOR 1000, pro AND 0001, pro NAND zase 1110...

Máme tedy 16 možných výsledků, od 0000 (vždy 0) po 1111 (vždy 1). Pojďme si je zase zapsat do tabulky.

AB				**Funkce**	**Poznámka**
00	**01**	**10**	**11**		
0	0	0	0	Y = 0	Kontradikce, vždy 0
0	0	0	1	Y = A AND B	Součin (konjunkce)
0	0	1	0	Y = NOT (A –> B)	Nonimplikace
0	0	1	1	Y = A	Projekce A
0	1	0	0	Y = NOT (A <– B)	Obrácená nonimplikace
0	1	0	1	Y = B	Projekce B
0	1	1	0	Y = A XOR B	Nonekvivalence
0	1	1	1	Y = A OR B	Disjunkce
1	0	0	0	Y = NOT (A OR B)	NOR
1	0	0	1	Y = A XNOR B	Ekvivalence
1	0	1	0	Y = NOT B	Negace B
1	0	1	1	Y = A <– B	Obrácená implikace
1	1	0	0	Y = NOT A	Negace A
1	1	0	1	Y = A –> B	Implikace
1	1	1	0	Y = NOT (A AND B)	NAND
1	1	1	1	Y = 1	Tautologie, vždy 1

Není jich víc? Co?

No, opravdu není. Tahle tabulka vyčerpala všechny možné kombinace výsledků pro všechny možné kombinace dvou vstupních hodnot.

Všimněte si, že v tabulce jsou naši staří známí (AND, NAND, OR, NOR, XOR a jeho negovaný bratr XNOR), jsou tam i funkce jedné proměnné (negace a projekce), jsou tam funkce, které jsou vždy 0, nebo vždy 1 (a přátelé, taková funkce, to snad ani není funkce!), a jsou tam ještě čtyři funkce implikace (normální, obrácená, a dvě jejich negované varianty). Funkce implikace jako jediné porušují pravidlo komutativnosti, tedy když zaměníme vývody A a B, dostaneme jinou funkci. Po pravdě řečeno nevím o tom, že by někdo „implikační“ hradla vyráběl jako integrovaný obvod. Pokud už někdo potřebuje třeba implikaci (1101), tak použije například hradlo OR, a na vstup A připojí invertor. Schválně se podívejte, jestli implikace není totéž co (NOT A) OR B.

14.4 Vícevstupová hradla

Ještě jsme si neřekli, i když si to mnozí z vás asi odvodili, že existují i vícevstupová hradla – v praxi se používají třívstupová, čtyřvstupová a osmivstupová, ale při syntéze obvodů v logických polích si klidně nadefinujeme třináctivstupové AND. Funkce je podobná těm dvouvstupovým, jen si slovo „oba" nahraďte slovy „všechny" (AND). Tedy: **Výsledkem AND je 1, pokud všechny vstupy jsou 1, jinak 0. OR je 1, pokud alespoň jeden vstup je 1, jinak 0**.

Schválně, co udělá třívstupový XOR? Třívstupový XOR je v logické 1 tehdy, pokud je na vstupu právě jedna logická 1, nebo pokud tam jsou tři. V logické 0 je, pokud na vstupu nejsou žádné jedničky, nebo pokud tam jsou dvě. Obvod XOR tak vlastně počítá **lichou paritu**.

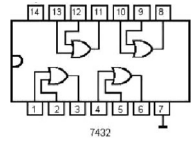
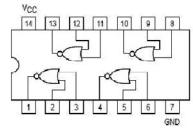
Lichá parita (Parity Odd) nějakého vícebitového signálu je jednobitový údaj, který je logická 1 tehdy a jen tehdy, pokud je počet logických 1 v signálu lichý. Paritní bit se používá pro jednoduchou kontrolu správnosti dat: pokud je jeden bit poškozen (má opačnou hodnotu), parita nesedí. Na více bitů bohužel jednoduchá parita nesedí. A ano, je i sudá parita (Parity Even), která má přesně opačnou funkci.

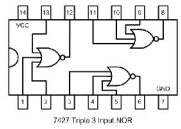
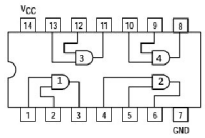
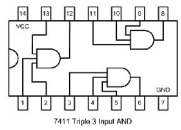
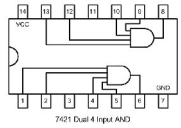
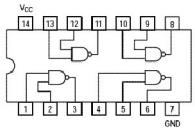
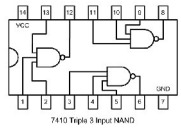
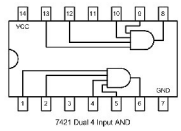
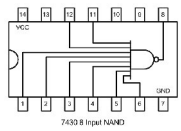
14.5 Mimochodem, když máme NAND, co ty ostatní?

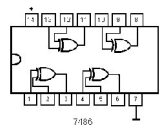
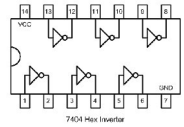
Už víme, že 7400 je čtveřice hradel NAND. V první kapitole jsme používali i obvod 7404, což je šestice invertorů.

Co ta ostatní hradla? Máme někde invertory, hradla NOR, AND, XOR? Máme někde třívstupová hradla? Čtyřvstupová?

Máme! Nejlépe je najdete v různých přehledech rodiny 74xx, ale protože jsou tak různě na přeskáčku, tak jsem si řekl, že bude dobře, když si je tu shrneme a ukážeme si i zapojení. S dovolením tedy – velký přehled hradel!

Funkce	Varianta	Základní	Schmitt	OC	Vývody základní verze
OR	4 dvouvstupová hradla	7432			
NOR	4 dvouvstupová hradla	7402	74232	7433	

Funkce	Varianta	Základní	Schmitt	OC	Vývody základní verze
NOR	3 třívstupová hradla	7427			
AND	4 dvouvstupová hradla	7408		7409	
	3 třívstupová hradla	7411		7415	
	2 čtyřvstupová hradla	7421			
NAND	4 dvouvstupová hradla	7400	7424	7401	
	3 třívstupová hradla	7410		7412	
	2 čtyřvstupová hradla	7420	7413	7422	
	1 osmivstupové hradlo	7430			

Funkce	Varianta	Základní	Schmitt	OC	Vývody základní verze
XOR	4 dvouvstupová hradla	7486		74136	
NOT	6 invertorů	7404	7414	7405	

V tabulce jsou kromě základních typů uvedené i typy se Schmittovým obvodem na vstupu. O něm jsme si už říkali – slouží k ochraně obvodů před signály, které nejsou dostatečně strmé. Další sloupec ukazuje obvody s otevřeným kolektorem na výstupu. V tuto chvíli to nechme stranou, později v této kapitole si o nich povíme víc.

14.6 Zjednodušování logických výrazů

Na logické výrazy můžeme s trochou dobré vůle nahlížet podobně jako na aritmetické – i zde existují například operace sčítání a násobení (OR, AND), i tyto operace mají neutrální prvek 0, resp. 1, i tyto operace jsou komutativní (lze prohodit operandy) a asociativní, i zde lze krátit a dělat podobné operace. Nejčastější situace, kdy je potřeba pracovat s těmito funkcemi, je tehdy, když máte daný „black box", tedy nějaký obvod s určitým počtem vstupů a výstupů, k tomu dostanete pravdivostní tabulku, tedy seznam požadovaných hodnot při určitých vstupních kombinacích, a vy máte navrhnout z hradel ekvivalentní zapojení.

V číslicové technice platí, že libovolný výraz, zadaný pomocí tabulky hodnot, můžeme převést na součet součinů (popřípadě součin součtů), a to nad vstupními signály a jejich negacemi. Tedy třeba (A AND B) OR (NOT B AND C) OR (C AND NOT D AND E)... V praxi se k takovému zjednodušení používají různé metody – Karnaughova mapa, algoritmus Quine-McCluskey...

14.7 AND-OR-INVERT

Přiznám se, že když jsem se seznamoval s číslicovou technikou, tak jsem moc nerozuměl smyslu tohoto obvodu. V katalogu TESLA jich bylo několik (mezi 7450 a 7470), a mně vrtalo hlavou, k čemu takový nesmysl může být. Vám ušetřím tápání: AND-OR je ten výše zmíněný „součet součinů", takže tyto obvody slouží právě ke skládání složitých logických funkcí.

No a druhá věc, která je na AND-OR hezká, je, že mohou přímo sloužit k vytvoření obvodu, který se jmenuje

14.8 Multiplexor

V anglické literatuře se tentýž obvod jmenuje **multiplexer**, a česky jsem slyšel oba tvary. Ačkoli se většinou přimlouvám za jednotnou terminologii, tak zrovna u tohoto slova používám oba tvary. Hanba mi!

Multiplexor je „digitálně řízený přepínač". V základní podobě má několik signálových vstupů (třeba 2, označme si je A a B), jeden výstup (Q) a jeden řídicí vstup (S), který říká, který ze vstupů má být připojen na výstup. U našeho dvouvstupového multiplexoru stačí jeden řídicí bit. Funkce je pak v závislosti na hodnotě S následující:

S	**Y**
0	hodnota ze vstupu A
1	hodnota ze vstupu B

Jasné? Pokud ne, tak si to celé ještě rozepíšeme:

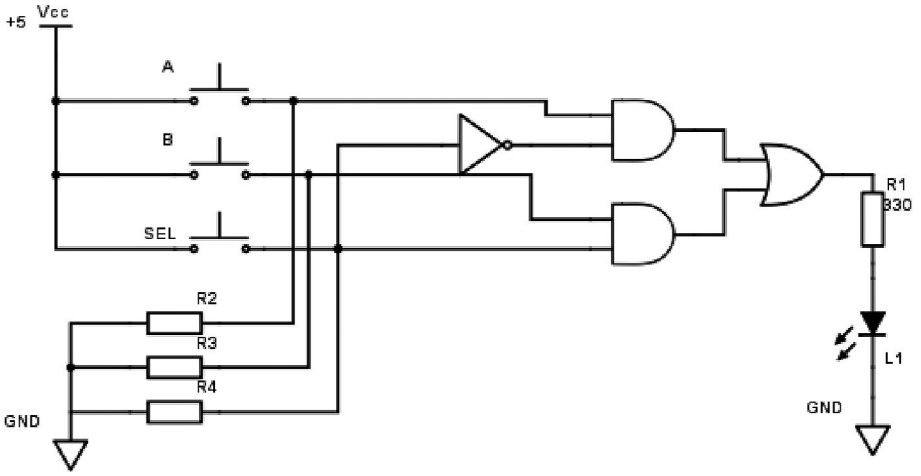
S	**A**	**B**	**Y**
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Všimněte si, že funkce odpovídá slovnímu popisu výše: pokud je S = 0, je na výstupu Y hodnota, která je na vstupu A, a vstup B se ignoruje. Pokud je S = 1, je na výstupu hodnota ze vstupu B bez ohledu na vstup A.

Můžeme navrhnout různá zapojení, ale nejjednodušší získáme, když si uvědomíme to, co jsme si řekli výše o hradlech: vstup S bude povolovat buď signál A, nebo signál B. Signál B bude povolovat, pokud bude roven 1. Takže „povolený signál B" bude vlastně „B AND S". Signál A je povolen

v nule, takže „povolený signál A“ bude „A AND NOT S“. No a protože víme, že „zakázaný signál“ je 0, a nula je zároveň neutrální hodnota pro OR, můžeme výsledek zapsat jako „povolené A OR povolené B“ – po dosazení tedy „(A AND NOT S) OR (B AND S)“.

A jak řekli, tak udělali…



⌀ https://eknh.cz/muxao

Náš multiplexor byste tedy mohli poskládat z jednoho obvodu OR (7432), z jednoho invertoru (7404) a ze dvou hradel AND (7408). Jenže: z OR byste využili jednu čtvrtinu obvodu 7432 (1 hradlo ze čtyř), z invertorů jednu šestinu a z AND pouhou polovinu obvodu (2 hradla ze čtyř). A to se nevyplatí.

Naštěstí to dělat nemusíte, protože většinu složitějších obvodů už za nás někdo poskládal a udělal je jako samostatné součástky. I multiplexory a další. Ale teď neřešme, jestli *to už existuje*, ale přemýšlejme, *jak to udělat*? Protože cílem této knihy je i to, abyste uměli vymyslet vlastní složitější konstrukci z jednodušších komponent.

Pomocí hradla AND-OR jsme stvořili multiplexor. Tedy teoreticky, nijak jsme si ho nezapojovali – nanejvýš v emulátoru. Jeho funkce spočívá v propojení vybraného vstupu s výstupem. Vstup se vybírá pomocí řídicího vstupu S. V naší nejjednodušší podobě má řídicí vstup S jeden bit, proto může vybrat jeden ze dvou vstupů. Pokud použijeme dva řídicí bity S1 a S2, můžeme vybrat jeden ze čtyř vstupů. Při třech řídicích bitech jeden z osmi vstupů. V praxi jsou nejčastěji používány právě tyto tři multiplexory, totiž se dvěma, čtyřmi a osmi vstupy.

Zkusme si cvičně vytvořit soustavu rovnic pro čtyřvstupový multiplexor:

Y = (A AND NOT S1 AND NOT S2) OR

(B AND S1 AND NOT S2) OR

(C AND NOT S1 AND S2) OR

(D AND S1 AND S2)

Na každém řádku je zapsaný logický součin pro jeden vstup. NOT S1 AND NOT S2 je splněno pro S1 = S2 = 0, S1 AND NOT S2 je splněno pro S1 = 1, S2 = 0 a tak dále. Takto vlastně „hradlujeme" čtyři vstupy, a nakonec je prostě sloučíme do jednoho.

14.9 Proč slučujeme přes OR?

Ano, správná otázka. To bychom nemohli ty vývody prostě jen tak spojit dohromady drátem? No, mohli. Taky bychom mohli zkratovat baterii hřebíkem, ale neděláme to. Důvod je, že by to nefungovalo tak, jak si představujeme. U logických obvodů totiž platí, že proud může téci z výstupu i do výstupu (za chvíli si řekneme, proč to tak je). A pokud bychom spojili „jen tak drátem" vývody dvou hradel, a na jednom by byla výstupní hodnota 0, na druhém 1, tekl by proud z jednoho výstupu do výstupu druhého hradla a jen by se darmo pálila elektřina. Pokud chceme nějak „sloučit" výstupy hradel, musíme vždy použít logickou funkci (pravděpodobně OR). Existují výjimky, ke kterým se dostanu za chvilku.

14.10 Dekodér (demultiplexor) „1-z-N"

Demultiplexor je součástka s opačnou funkcí, než má multiplexor. Má jeden signálový vstup, několik výstupů, a řídicí vstup, který říká, na jaký výstup bude vstup připojen. Ostatní výstupy budou v logické nule, případně ve třetím stavu (za chvíli si o něm něco povíme).

Speciálním případem demultiplexoru je dekodér – ten nemá signálový vstup, jen výstupy a řídicí vstupy. Podle toho, jaká kombinace je na řídicích vstupech, je na zvoleném výstupu jednička, na ostatních nula (nebo obráceně).

Dekodéry se používají nejčastěji 1-ze-4, 1-z-8, 1-z-10 a 1-z-16. Mají tedy dva, tři nebo čtyři řídicí vstupy a odpovídající počet výstupů. Logická funkce je podobná té u multiplexoru:

Y1 = NOT S1 AND NOT S2

Y2 = S1 AND NOT S2

Y3 = NOT S1 AND S2

Y4 = S1 AND S2

S1	S2	Y1	Y2	Y3	Y4
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

14.11 Vícebitové varianty

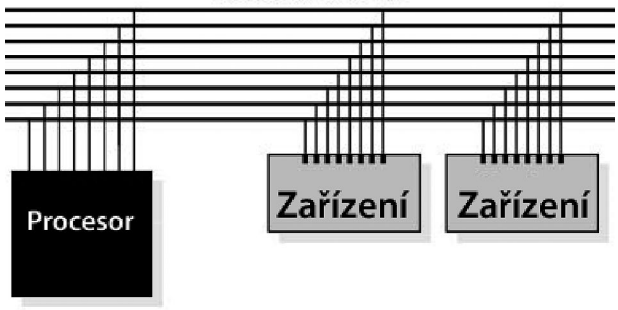
Jak multiplexory, tak demultiplexory se používají nejen jako jednobitové, ale i jako vícebitové – fyzicky to je pak několik (de)multiplexorů se spojenými řídicími vstupy. Hovoříme pak například o čtyřkanálovém čtyřbitovém multiplexoru (tedy 2 řídicí vstupy, 4 datové vstupy, každý po 4 bitech, 1 výstup po 4 bitech), nebo dvoukanálovém (1 řídicí bit) osmibitovém (vstupy i výstupy mají osm vodičů, A0-A7, B0-B7 ,Y0-Y7) multiplexoru. Tyto obvody se používají velmi často, a to pro směřování signálů takovou cestou, jakou bychom si představovali. Například pokud znáte strojový kód procesoru 8080 nebo Z80, tak víte, že v instrukčním slovu u operace MOV a dalších je číslo registru ve třech bitech. Fyzicky v procesoru existuje osmikanálový multiplexor, který má tyto tři bity připojeny na svoje řídicí vstupy, na datových vstupech má připojeny jednotlivé registry, a podle hodnoty pak na vnitřní datovou sběrnici připojí jeden z těchto registrů.

14.12 Otevřený kolektor, třetí stav, OE

Průběžně tu na to narážím, tak je načase vysvětlit, oč jde.

Už jsem tu říkal, že nesmíme jen tak spojit výstupy hradel TTL „nakrátko“. Nejen že by tekly obvodem úplně zbytečné proudy, ale navíc by ani nebylo jasné, jaká hodnota je na výstupu – jednotlivá hradla by se o vedení přetahovala a vyhrálo by to nejsilnější. Proto: ne, nikdy, nikdy, nikdy!

Jenže! Občas potřebujeme, aby se na jedno společné vedení připojovalo víc funkčních jednotek. Třeba na datové sběrnici procesoru je připojená paměť RAM, paměť ROM a různé periferie, a podle adresy se vybírá konkrétní obvod, s nímž se komunikuje. Až dosud dobrý, ale teď mi řekněte: Co mají dělat ostatní obvody, připojené tamtéž, se svými výstupy? Nemůžou být ani v log. 1, ani v log. 0, tím by totiž rušily přenos dat.

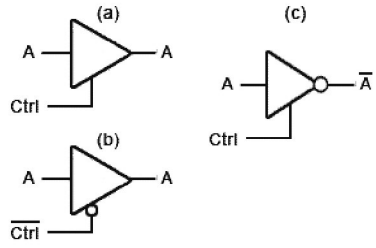


Co mají udělat? Jedna z možností je použít obří multiplexor, a připojovat vždy pouze jediné zařízení. Jenže když si představíte, jak by to vypadalo u osmibitového počítače: máme třeba pět zařízení (RAM, ROM, časovač, paralelní port, sériový port), každé po osmi bitech, to je 40 bitů, dalších 8 bitů výstup, 3 bity řídicí – jen ten multiplexor by měl pouzdro větší než celý procesor!

V praxi se na to jde jinak. Existuje totiž něco, čemu se říká třetí stav. Označuje se Z, říká se mu taky „stav vysoké impedance" a znamená to, že obvod má výstup odpojený. Zkrátka neposílá na něj ani logickou 0, ani logickou 1, zkrátka nic. Fyzicky se uvnitř obvodu odpojí jak od země, tak od napájecího napětí.

Ne každý obvod má třístavové výstupy. Hradla, o nichž jsme se až dosud bavili, to většinou neumožňují. Složitější součástky, třeba dnes popsané demultiplexory, nebo třeba paměti či procesory, ale mají možnost, jak své výstupy odpojit. U pamětí, dekodérů, demultiplexorů a podobných najdeme vstup, který se jmenuje OE, tedy „output enable". Pokud je 1, obvod funguje normálně, pokud je 0, jsou všechny vývody nastavené do stavu Z, tedy odpojené.

Díky tomu může být na jedné datové sběrnici připojena paměť RAM i ROM i periferie. Všechny tyto obvody mají totiž výstupy ve stavu Z, a procesor si určí, ze kterého chce číst. Tomu pak povolí přístup na sběrnici tím, že mu pošle signál OE.



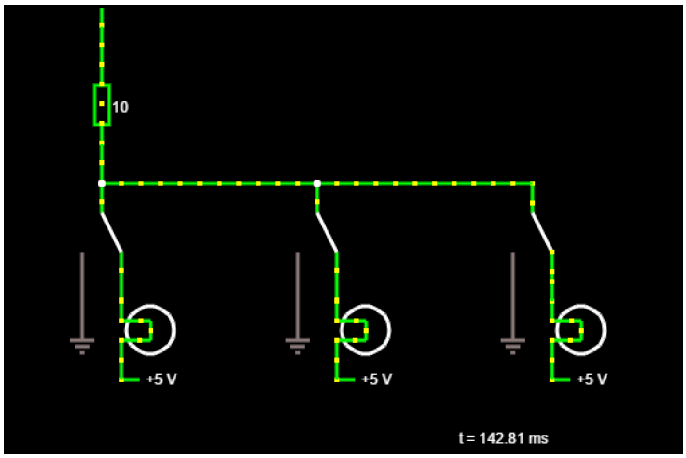
U jednoduchých obvodů máte k dispozici buď budiče s třístavovým výstupem (a), popřípadě takové, kde je řídicí vstup negovaný (b), anebo invertory s třístavovým výstupem (c).

Speciálním typem třístavových zařízení, s nimiž se setkáte i u hradel, je takzvaný výstup s **otevřeným kolektorem**. Setkáte se s ním i dnes poměrně běžně – třeba sběrnice I²C, používaná pro připojování senzorů (v dalším textu si ji představíme), má právě tento druh výstupů. Většina lidí si vystačí s tím, že si pamatují, že „k takovému výstupu musí přidat pull-up rezistor", ale my si teď řekneme i důvod, proč tomu tak je.

Výstup s otevřeným kolektorem má dva stavy: logickou 0, při níž je výstup spojen se zemí, a logickou 1, při níž je výstup odpojený (stav Z). Vzpomeňte si na naše zapojování tlačítka – tohle je obdobný stav. Pokud spolu spojíme vodičem výstupy s otevřeným kolektorem, máme dvě možnosti: Buď jsou všechny výstupy v logické 1, jsou tudíž odpojené a na spojovacím vodiči je v tu chvíli „nic", nebo je některý z výstupů v logické 0, a spojovací vodič je tedy spojen se zemí.

Na takto spojené výstupy s otevřeným kolektorem se připojuje přes dostatečně velký rezistor, třeba 10k, napájecí napětí. V klidovém stavu, když jsou výstupy odpojené, je tak na výstupech přes tento rezistor napájecí napětí, a tedy logická 1. Pokud některý z výstupů sepne k zemi, bude na vedení logická 0. Samozřejmě si to můžeme nasimulovat pomocí přepínačů a žárovek, princip je stejný.

Řekněme, že máme tři stanoviště, kde může vzniknout poplach, a chceme, aby vždy všechna stanoviště viděla, že na některém vznikl poplach. Máme opět spoustu možností, ale když zvolíme „otevřený kolektor", vystačíme si s jediným signálovým vodičem.



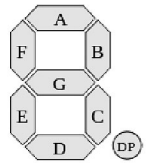
Za normálního stavu, tedy v klidu, je signálový vodič připojen k žárovkám na jednotlivých stanovištích, a zároveň přes rezistor k napájecímu napětí. Je na něm tedy logická 1. Jakmile se na

některém ze stanovišť přepne přepínač k zemi, tedy k logické 0, uzavře se obvod tak, že poteče proud přes žárovky na stanovištích přes tento přepínač k zemi. Žárovky se rozsvítí, a o to nám šlo!

U velmi často používané sběrnice I^2C se jedná o podobný postup (stejně jako u 1-Wire – obě sběrnice vám v pravý čas představím). Zařízení jsou datovými vodiči s otevřeným kolektorem připojeny k signálu SDA, a v klidu je tento signál „zvedán" k log. 1 rezistorem. Pokud je třeba vyslat data, tak buď procesor, nebo zařízení (podle toho, jak se předem domluví), posílá data tak, že buď „stahuje výstup k nule", nebo jej nechává odpojený (=log. 1).

14.13 Dekodéry

Z kombinačních obvodů už nám toho k probrání moc nezbývá. Užitečné jsou dekodéry, které berou vstupní informaci, většinou vícebitovou, a převádí ji na nějaký jiný vícebitový kód. Asi nejpoužívanější jsou dekodéry, určené pro připojení sedmisegmentových displejů. Sedmisegmentovky pomocí rozsvěcení určitých segmentů ukazují číslice 0-9 (a s trochou dobré vůle i některá písmena). Jejich segmenty jsou označeny A až G, a to takto:



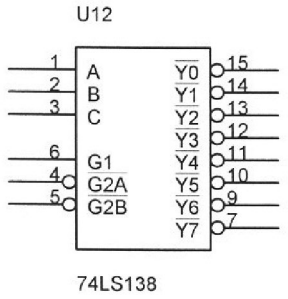
Pokud pečlivě počítáte, zjistíte, že segmentů je osm, a tím osmým je desetinná tečka (DP). Tu ale můžeme teď vypustit a nestarat se o ni.

Pro přenos hodnot 0 až 9 potřebujeme čtyři bity. Čtyři bity dokážou přenést hodnoty 0 až 15. Otázka je, co s hodnotami 10 až 15. Nejjednodušší způsob je prohlásit je za zakázané. Výsledný čtyřbitový kód se pak nazývá kód BCD – Binary Coded Decimal, tedy dvojkově kódované desítkové číslo. Dekodér BCD-to-7segment pak pracuje podle následující tabulky:

s	BCD	G	F	E	D	C	B	A
0	0000	0	1	1	1	1	1	1
1	0001	0	0	0	0	1	1	0
2	0010	1	0	1	1	0	1	1
3	0011	1	0	0	1	1	1	1
4	0100	1	1	0	0	1	1	0
5	0101	1	1	0	1	1	0	1
6	0110	1	1	1	1	1	0	1
7	0111	0	0	0	0	1	1	1
8	1000	1	1	1	1	1	1	1
9	1001	1	1	0	1	1	1	1

Takové obvody fakt existovaly a existují, například 7447 (za socialismu se vyráběl v NDR pod označením D147D a představoval nejjednodušší způsob, jak vytvořit nějaký displej se sedmisegmentovkami). Ale sedmisegmentovkám se ještě budeme věnovat podrobněji.

Další oblíbené dekodéry jsou 1-z-N, tedy například 1-z-8. Tento obvod má tři vstupy a osm výstupů (proto se taky někdy označuje jako 3-na-8 – čteme „jeden z osmi“ a „tři na osm“), a aktivní je vždy jen jeden výstup, podle toho, jaká kombinace je na vstupech. Jako příklad nám může posloužit obvod 74LS138.



Jeho pravdivostní tabulka je následující:

Vstupy						**Výstupy**							
Výběr			Řízení										
C	B	A	G1	/G2A	/G2B	/Y0	/Y1	/Y2	/Y3	/Y4	/Y5	/Y6	/Y7
0	0	0	1	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	0	1	0	1	1	1	1	1	1
0	1	0	1	0	0	1	1	0	1	1	1	1	1
0	1	1	1	0	0	1	1	1	0	1	1	1	1
1	0	0	1	0	0	1	1	1	1	0	1	1	1
1	0	1	1	0	0	1	1	1	1	1	0	1	1
1	1	0	1	0	0	1	1	1	1	1	1	0	1
1	1	1	1	0	0	1	1	1	1	1	1	1	0
X	X	X	0	X	X	1	1	1	1	1	1	1	1
X	X	X	X	1	X	1	1	1	1	1	1	1	1
X	X	X	X	X	1	1	1	1	1	1	1	1	1

Můžeme brát vstupy A, B, C jako tříbitovou adresu v rozsahu 0-7 (tedy binárně 000 až 111). Binární „hodnota" na těchto třech vstupech určuje, který výstup bude aktivní.

Obvod má tři řídicí vstupy G, jeden normální (G1) a dva negované (/G2A a /G2B). Obvod pracuje tehdy, když je výsledek G = G1 AND NOT /G2A AND NOT /G2B = 1, tedy pouze tehdy, když G1 = 1, /G2A = 0 a /G2B = 0. Pokud tato podmínka splněna není, bude na všech vývodech logická 1, tedy „neaktivní stav" (vývody jsou negované). Takovou situaci ukazují poslední tři řádky tabulky.

Pokud je na vstupech G správná kombinace, tak funkci obvodu můžeme popsat nejlépe slovním popisem: **Všechny vývody jsou neaktivní (log.1) s výjimkou vývodu, jehož číslo odpovídá kombinaci bitů na vstupech C, B, A. Například pro kombinaci CBA = 110, což je binární podoba číslice 6, je aktivní (=log. 0) vývod /Y6.**

Tento obvod se velmi často používal v osmibitových počítačích na adresování periferií a pamětí. Ze tří adresových vodičů dokázal vytvořit osm řídicích signálů, které pak přímo ovládaly jednotlivé periferie.

Mimochodem, ve schématech starších počítačů najdete někdy adresový dekodér s označením 3205. To byl obvod s totožnou funkcí, ale z jiné produktové řady, a protože byl v ČSSR dostupnější než 74138, používal se ten. I když měl vyšší spotřebu.

Existují i další typy dekodérů: 2-na-4 (tedy 1-z-4), 4-na-16 (tedy 1-z-16), nebo i 4-na-10 (1-z-10).

Takovéto dekodéry jsou vlastně *degradované multiplexory* – funkce je stejná (posílá se signál na jeden z N daných výstupů), ovšem signál je pevně daný (v případě obvodu 74138 to je logická 0).

14.14 Pojďme, budeme už fakt něco počítat!

Ne, vážně – zatím tu děláme samé OR a AND a přepínáme signály zleva doprava, ale ještě jsme nespočítali ani kolik je 6 + 3. A ne že by to nešlo. Jde to. A jak?

No, budete počítat s dvojkovými čísly. Budete sčítat 0110 (což je 6) a 0011 (což jsou 3). Jste na tom stále líp než staří Římané, kteří sčítali VI + III – to je hrozně nešikovné. Díky pozičnímu zápisu číslic v číslech můžete zapsat pod sebe:

```
 0110
+0011
-----
=????
```

Pamatujete se, jak jsme sčítali pod sebou víceciferné sčítance ve škole? Začnete zprava a budete sčítat jednotky s jednotkami:

0 + 1 = 1

Poslední číslice výsledku je tedy 1. Přesuňte se k vyššímu řádu – v našem případě ke „dvojkám" (v desítkové soustavě k desítkám). Sčítejte opět pod sebou: 1 + 1 = ? No, v desítkové soustavě to jsou 2, ve dvojkové nemáme dvojku, tak výsledek bude „10". Jak ho zapíšete? No stejně jako při obyčejném sčítání: zapíšete si nulu, a tu jedničku si přenesete do vyššího řádu (tady „do čtyřek")

Předposlední číslice je 0 a posouváte se o řád výš, tedy opět doleva. Tady sčítáte 1 + 0, a k tomu musíte připočítat tu jedničku, co jste si přenesli v předchozím kroku: 1 + 0 + 1 = ? Opět tedy ve dvojkové soustavě 10. A stejný postup – zapíšete nulu, jedničku si přeneste o řád výš.

Třetí od konce tedy je nula, jděte o řád výš, na „osmičky", a sčítejte: 0 + 0 + 1 (to je přenos z předchozího kroku). Tedy výsledek: 1. Zapíšete…

Dostali jste tedy 1001 – což je binárně 9 (1 × osmička, 1 × jednička).

Zkuste si:

a. 0110 + 0111

b. 1000 + 1001

c. 1010 + 1111

Máte? Čistě pro kontrolu:

a: 1101 (6 + 7 = 13), b: 10001 (8 + 9 = 17), c: 11001 (10 + 15 = 25)

Tak. A teď si pojďme říct, jak to tedy počítáme pro jednotlivé bity.

Mějme dvě čísla, A a B. Každé z nich bude čtyřbitové. Jednotlivé bity si označíte A3-A0 a B3-B0. jednotky budou A0/B0, dvojky A1/B1, čtyřky A2/B2 atd.

Na nejnižším řádu sčítáte A0 + B0. Výsledek si označíte S0, přenos z nejnižšího řádu C0 (C nebo CY = Carry, tedy přenos). Jak bude vypadat tabulka?

A0	B0	C0	S0
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

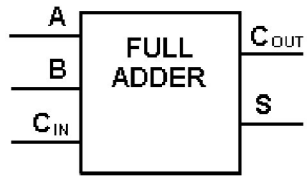
Schválně se podívejte na sloupce C0 a S0, mělo by vás to do očí uhodit... Už víte?

No jasně! C je přeci obyčejné A AND B, S je A XOR B. Takto vypadá obvod, kterému se říká **poloviční sčítačka** (anglicky *half adder*).

⌀ https://eknh.cz/had

U nejnižšího řádu nám stačí, ale už u řádu dvojek budete potřebovat přičíst ještě přenos z nižšího řádu – C0. Musíte tedy použít plnou sčítačku (anglicky *full adder*). Bude mít tři vstupní signály – An, Bn a vstupní přenos CIn – a dva výstupní: Sn a COn. (CI jako Carry In, CO jako Carry Out). Její obecná funkce se dá popsat následující tabulkou:

An	Bn	CIn	COn	Sn
0	0	0	0	0
0	1	0	0	1
1	0	0	0	1
1	1	0	1	0
0	0	1	0	1
0	1	1	1	0
1	0	1	1	0
1	1	1	1	1



Když se budete na výstupní dvojici CO a S dívat jako na dvoubitové číslo, zjistíte, že jeho hodnota (0 – 3) udává počet jedniček na vstupech A, B, CI. To je zajímavý poznatek, ale vy ho potřebujete

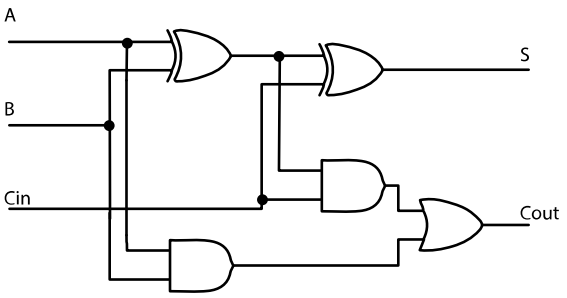
převést nějak do řeči hradel. Můžete na to jít od lesa, prohlásit, že Y je vlastně (A XOR B), a pokud je CI = 1, tak je to negované. Schválně, máme takovou funkci, která by dělala něco takového, jako že „S = A pokud B=0, a pokud B=1, tak S = NOT A“? Máme?

Máme, a je to, světe div se, XOR. Podívejte se znovu na její tabulku hodnot – vidíte to tam? Takže u plné sčítačky můžete říct, že S = (A XOR B) XOR CI

Co s CO? Tam je to trošku složitější. Pokud CI = 0, tak jde o A AND B, pokud CI = 1, tak jde o A OR B. To moc nepomáhá. Co si to takhle popsat slovy: CO je 1, pokud je zároveň A a B nebo A a C nebo B a C. Výrazem: CO = (A AND B) OR (A AND CI) OR (B AND CI)

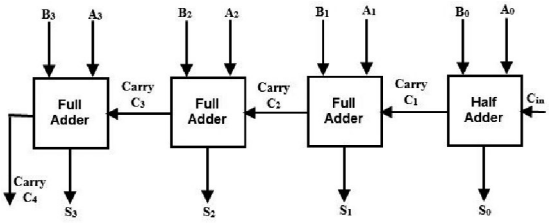
Mimochodem, víte, jak jsem psal, že OR se taky označuje jako logický součet a AND jako logický součin? Mohli bychom přepsat výše uvedený výraz do podoby CO = A × B + B × CI + A × CI

Jediný problém je, jak udělat OR se třemi vstupy. Kupodivu docela jednoduše: A + B + C = (A + B) + C.



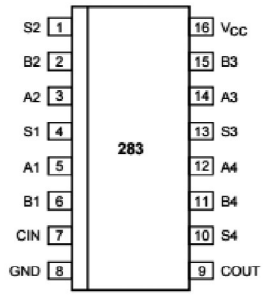
⌀ https://eknh.cz/fad

Tak, a teď je potřeba tohle zopakovat čtyřikrát a získat tak čtyřbitovou sčítačku. Do každé sčítačky zavedete příslušné bity z čísel A a B, a k nim přenos z předchozího řádu. U nejnižšího bude přenos nula, přenos z nejvyššího řádu udělá pátý (nejvyšší) bit výsledku.



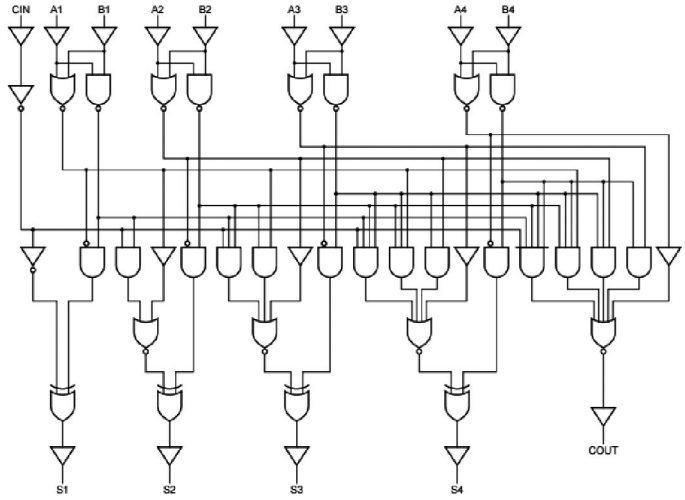
Všimněte si, že v rámci zjednodušení jsem výše uvedené zapojení sčítačky z elementárních hradel proměnil v „black box" – prostě obdélníček, na něm je napsáno „toto je sčítačka", a k tomu popsané vývody. Uvnitř je to ale pořád to původní schéma.

Výsledný obvod je tak složitý, že se nikomu nevyplatí něco takového stavět. Lepší je koupit to už hotové. Co myslíte, udělal to někdo? Odpověď zní: Ano, a má to označení 74283. Je v pouzdru se šestnácti vývody, a zapojeno je to takto:



Cin je vstupní přenos do nejnižšího řádu, Cout výstup do řádů vyšších. Díky tomu je možné dvě sčítačky zapojit paralelně tak, aby vytvořily osmibitovou sčítačku, nebo klidně i vícebitovou. Jediný rozdíl je, že výrobce řády nečísluje od nuly, ale od jedničky (A1-A4). Na funkci to ale nic nemění.

Věřte nebo ne, ale uvnitř je to přesně tak, jak jsme si popsali výš:



Můžeme si z toho něco postavit? No jasně, co třeba binární sčítačku?

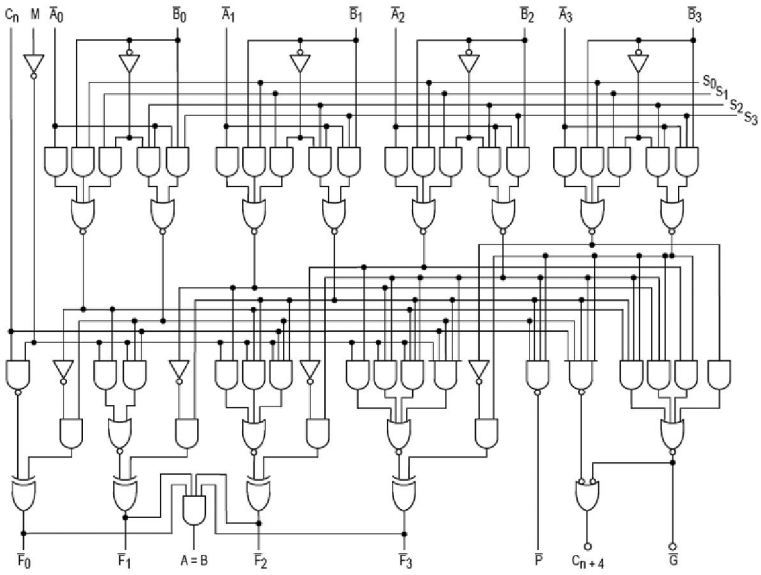
Vezmeme osm přepínačů, uděláme z nich dvě skupiny po čtyřech, výsledek nám bude zobrazovat pět LEDek… Máte-li po ruce obvod 74283, můžete si to vyzkoušet na nepájivém poli, ale nemáte-li jej, nezoufejte, můžete použít online emulátor.

⌀ https://eknh.cz/add4

14.15 Aritmeticko-logická jednotka (ALU)

Asi nejsložitější obvody, které můžete stvořit kombinační technikou, jsou násobičky a aritmeticko-logické jednotky (ALU). Takové obvody mívají většinou dva vícebitové vstupy, jeden vícebitový výstup, a k tomu sadu řídicích vstupů, které určují, jakou operaci obvod právě dělá. Většinou mívá i jakési „vedlejší výstupy", třeba přenos nebo informaci o tom, že jsou si čísla rovna.

Reálným příkladem takového obvodu je obvod 74181. Tento obvod zpracovává dva čtyřbitové vstupy (A0-A3 a B0-B3) a nabízí sadu funkcí pro sčítání, odčítání, logické operace či posuny, a to s přenosem i bez přenosu. Funkce se vybírají pomocí čtyřbitového řídicího vstupu S0-S3. Pro zajímavost se podívejme na schéma:



CC BY-SA 3.0, Link

Sčítání dvou čísel zabralo tomuto obvodu 22 nanosekund. Verze „Schottky" 74S181 sčítala dvě čísla 11 nanosekund, rychlá verze 74F181 sedm nanosekund. Rychlost byla dána rychlostí přepínání hradel a tvořila technologický limit – nějaké „zvyšování kmitočtu" nepomohlo. Důvod je ten, že sčítačka sčítá všechny pozice najednou, a jediné, co ji brzdí, je právě zmíněný technologický limit, který se projeví jako zpoždění při přenosu z nižšího řádu do vyššího.

Funkce se volily jednak pomocí výběru S0-S3 a dále pak vstupem, který přepínal logické a aritmetické instrukce (M). Obvod pracoval buď v normálním režimu, nebo v negovaném. Za normálního režimu dokázal následující operace (Cn je přenos z nižšího řádu):

MODE SELECT INPUTS				ACTIVE HIGH INPUTS AND OUTPUTS	
S_3	S_2	S_1	S_0	LOGIC (M=H)	ARITHMETIC(2) (M=L; C_n=H)
L	L	L	L	$\overline{A}$	A
L	L	L	H	$\overline{A + B}$	A + B
L	L	H	L	$\overline{A}B$	$A + \overline{B}$
L	L	H	H	logical 0	minus 1
L	H	L	L	$\overline{AB}$	A plus $A\overline{B}$
L	H	L	H	$\overline{B}$	(A + B) plus $A\overline{B}$
L	H	H	L	$A \oplus B$	A minus B minus 1
L	H	H	H	$A\overline{B}$	$A\overline{B}$ minus 1
H	L	L	L	$\overline{A} + B$	A plus AB
H	L	L	H	$\overline{A \oplus B}$	A plus B
H	L	H	L	B	(A + $\overline{B}$) plus AB
H	L	H	H	AB	AB minus 1
H	H	L	L	logical 1	A plus A(1)
H	H	L	H	$A + \overline{B}$	(A + B) plus A
H	H	H	L	A + B	(A + $\overline{B}$) plus A
H	H	H	H	A	A minus 1

Násobení nevedeme... Násobilo se buď postupně, nebo existovala hardwarová násobička – velmi specializovaná součástka. Pro vícebitové operace se zapojovalo několik těchto obvodů vedle sebe a propojovaly se pomocí přenosu Cn. Protože zpoždění signálu mohlo ovlivnit výsledek, používal se obvod 74182 pro „předvídání přenosu".

Předvídání přenosu je postup, kdy pomocí jednoduché kombinační logiky rychle tipneme, zda výsledek „přeteče", a dál s tímto odhadem pracujeme, takže pro většinu případů získáme výsledek rychleji.

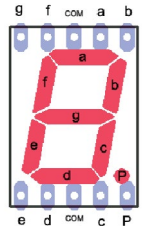
Aritmeticko-logická jednotka je základem každého moderního procesoru. Dnes už je integrovaná přímo na čipu, ale v dřevních dobách let 60. a 70. se opravdu skládala z hradel, později z takovýchto málobitových jednotek vedle sebe.

15 Sedmisegmentovky LED

15 Sedmisegmentovky LED

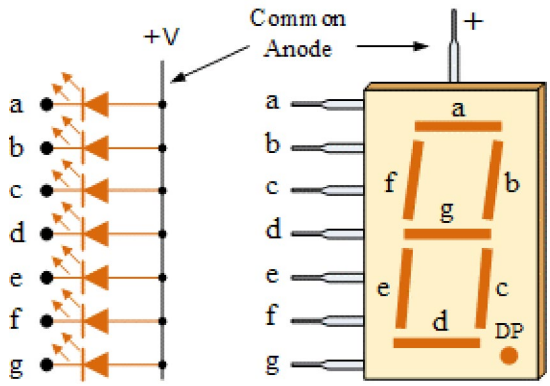
Odpočiňme si na chvíli od kombinačních obvodů a podívejme se na něco mnohem atraktivnějšího. Podívejme se na známé *digitální osmičky*, ze kterých se dělaly displeje. A ačkoli vypadají jako osmičky, říká se jim *sedmisegmentovky*. Správně vlastně „sedmisegmentová zobrazovací jednotka", ale než to řeknete, tak posluchači usnou.

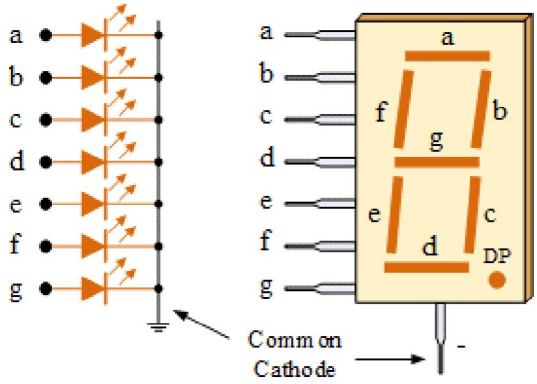
Bývaly doby, kdy sedmisegmentovka představovala ultramoderní techniku. Když chtěli ve filmu tvůrci ukázat, jak doba kvačí a budoucnost je na dosah, vrzli tam něco, co zobrazovalo na displeji ze sedmisegmentovek nějaké číslice. Sedmisegmentovky jsme pak všichni měli v kapesních kalkulačkách a digitálních hodinkách.



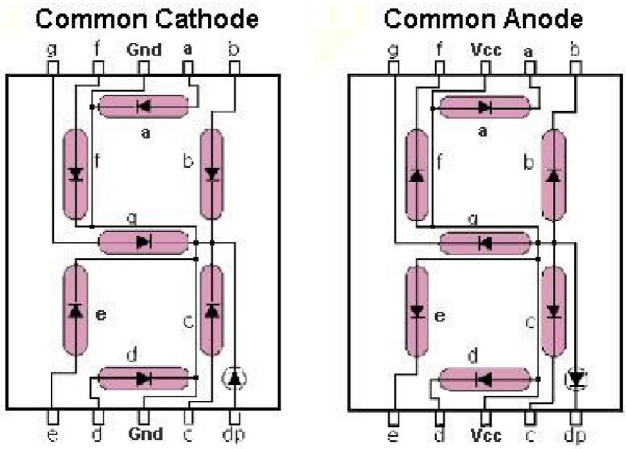
Sedmisegmentovka má opravdu sedm segmentů. Označují se tradičně písmeny A až G, od horního vodorovného po směru hodinových ručiček, prostřední vodorovný je pak poslední. Jenže naprostá většina sedmisegmentovek má vstupů osm: poslední je desetinná tečka, nebo dvojtečka. Sedmisegmentovek je spousta různých druhů, dělají se sdružené displeje po dvou číslicích, po čtyřech, po osmi, dělají se speciální „hodinové" displeje, které mají jeden a půl pozice…

Sedmisegmentovky se skládají ze sedmi (s tečkou z osmi) svítících diod (LED), které jsou paralelně spojené. Ovšem je otázka, jak jsou spojené, jestli katodami, nebo anodami.





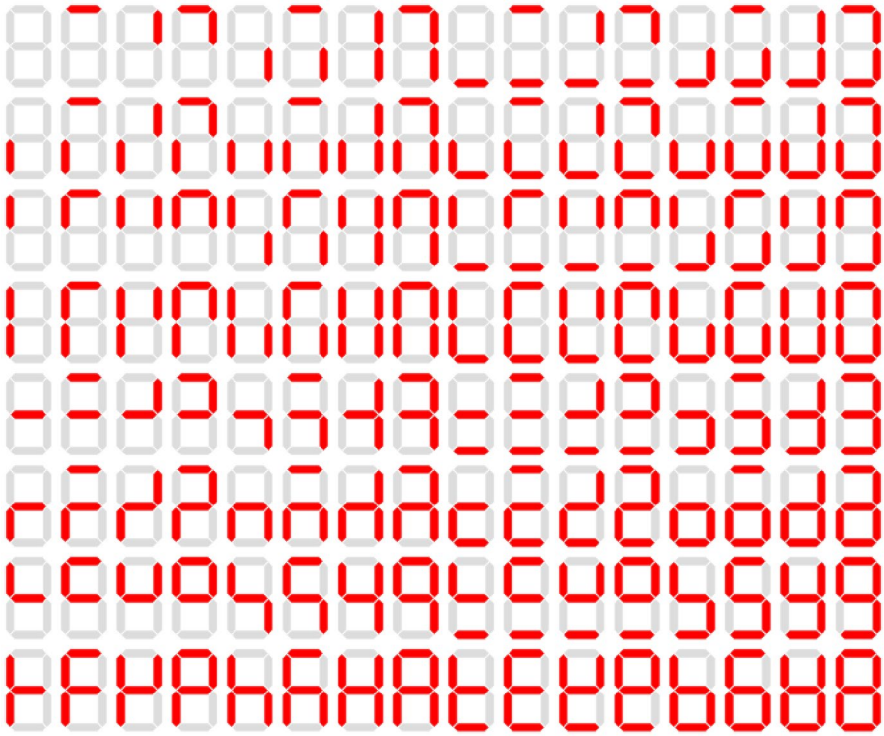
Podle toho, jestli je sedmisegmentovka se společnou katodou (CC, Common Cathode), nebo se společnou anodou (CA, Common Anode), se bude lišit i práce s ní. U sedmisegmentovky se společnou katodou se společný vývod zapojuje na zem, a segmenty budou svítit tehdy, když bude na příslušný vstup přivedeno kladné napětí (tedy log. 1). U společné anody tomu bude obráceně – společná anoda je zapojena na kladné napájecí napětí, a segment, který má svítit, musí mít vstup v log. 0, neboli uzemněný.



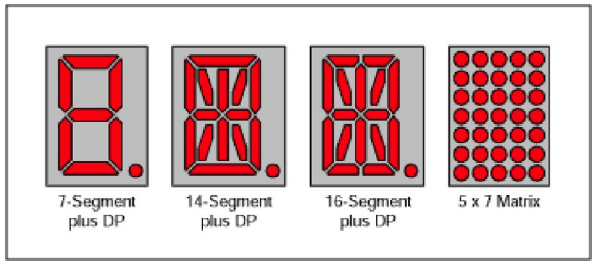
Samosebou nesmíme zapomenout na rezistory ke každému segmentovému vstupu. Uvnitř jsou úplně stejné LED jako známe, tak je potřeba se k nim i stejně chovat.

Poznámka: Nechme teď stranou displeje z kapalných krystalů (LCD). Samozřejmě existují i v podobě sedmisegmentovek, ale jejich princip není tak jednoduchý a vyžadují trošku specifičtější zacházení.

Pomocí sedmisegmentovky můžeme zobrazit 127 různých znaků, které ukazuje následující obrázek.



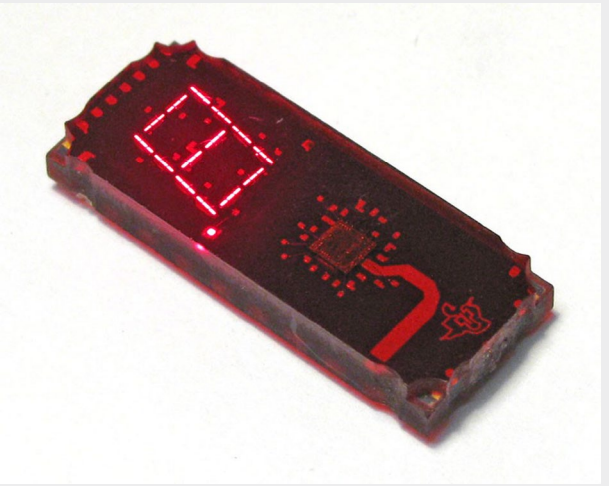
Většinou se používají číslice 0 až 9 a některá písmena. Pokud chcete lepší ztvárnění písmen i číslic, můžete použít zobrazovače se 14 nebo 16 segmenty, nebo pak matici bodů.



Pro displeje ze sedmisegmentovek existují speciální obvody – dekodéry, které převádějí (většinou čtyřbitovou) informaci na sedm bitů pro sedmisegmentovku. O nich jsem se už zmiňoval v kapitole o dekodérech a kombinační logice.

15.0.1 Mimochodem...

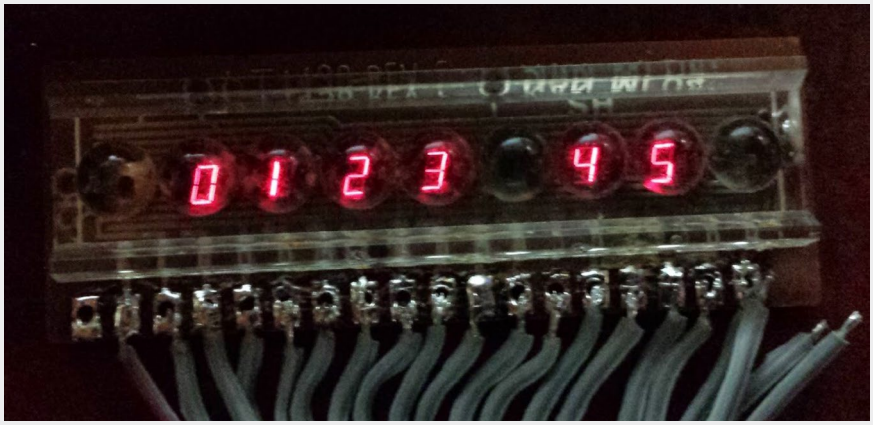
Sedmisegmentovky jsou dnes už velmi levné, ale bývaly to poměrně drahé součástky. Dnes už i ty staré, původní, seženete defacto „na váhu", za pár korun. S některými výjimkami, jako jsou obvody TIL od Texas Instruments. Šlo o jedny z prvních sedmisegmentovek, a výrobce Texas Instruments třeba do modelu TIL 306 či TIL307 zamontoval i integrovaný obvod, který fungoval jako čítač a jako dekodér.



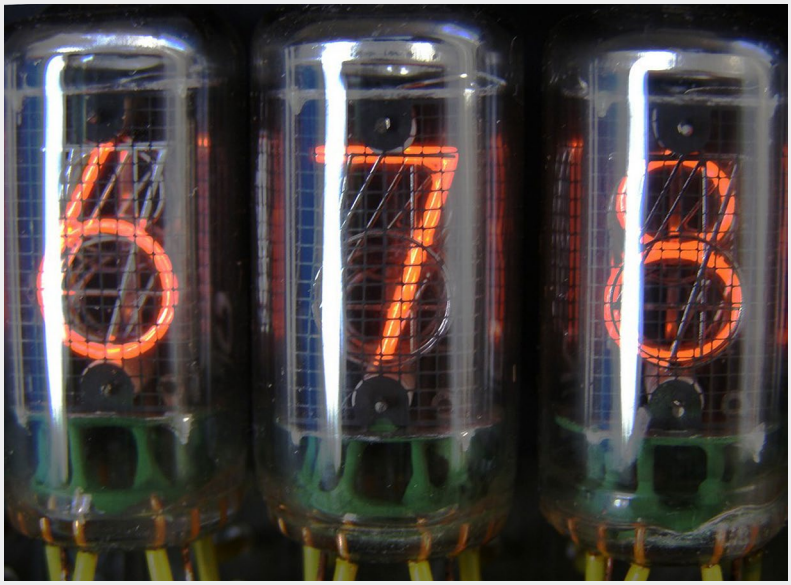
Taková sedmisegmentovka je dnes skoro malý poklad, protože její ceny na aukčních serverech jsou okolo dvou set korun za jeden kus. Podobně i jejich klony z bývalého NDR, jako je například VQC10. Ovšem pokud stavíte něco, co má mít „retro vzhled", budou tyto displeje naprostým hitem.



Další oblíbený displej byl takzvaný „bublinkový“. Šlo o několik sedmisegmentovek, spojených do jednoho bloku (většinou po 4, 8 nebo 9). Nad každou sedmisegmentovkou byla plastová „bublinka“, která fungovala jako čočka a zvětšovala titěrné symboly tak, aby byly čitelné. Podobné displeje se používaly převážně v kalkulačkách.



No a když už jsme v tom retrovzpomínání, tak se dodneška můžeme potkat s takzvanými „nixie“ displeji – vypadaly jako elektronky, a nebyly zapojené po segmentech, ale měly speciálně tvarované vlákno pro každou číslici. Vyráběly se i v Sovětském svazu, kde se jim říkalo Itrony.



15.1 Víc sedmisegmentovek...

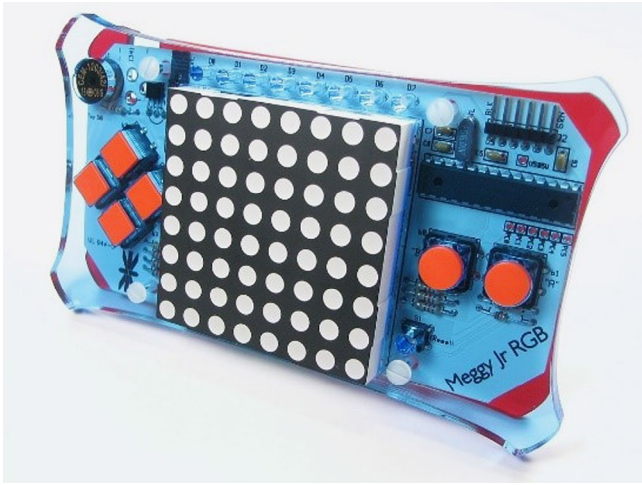
Na Aliexpressu nebo eBay najdete mnoho hotových modulů se sedmisegmentovkami. Většinou používají čínské obvody TM16xx, popřípadě obvody Maxim MAX72xx. Jeden z oblíbených typů je osmimístný displej s obvodem MAX7219. Tento obvod v zásadě funguje podobně jako naše oblíbené posuvné registry, jen místo osmi segmentů dokáže budit až 64 segmentů (8 pozic po osmi segmentech). Pomocí jednoduchého sériového rozhraní (CLK a MOSI) pošlete do obvodu 16bitové řídicí slovo, které nastavuje segmenty, zapíná a vypíná dekodér znaků nebo řídí intenzitu, a pulsem na vstupu LOAD jej nahrajete do registrů (a tedy provedete). Výhodou je, že obvod MAX7219 je průchozí, a je tedy možné naskládat takových displejů za sebe několik.



Když může takový obvod budit 64 LED segmentů, tak kde je psáno, že to musí být jen sedmisegmentovky? Může to být klidně i matice diod:



Takové matice můžete zase pospojovat za sebe a vytvořit z nich například displeje, podobné těm, co v MHD ukazují názvy stanic. Dokonce existují i velmi jednoduché herní konzole s takovými displeji (ovšem s RGB LED):



Málokdy budete s takovými obvody pracovat napřímo, většinou budou skryté v nějakém modulu spolu s LED a omezovacími rezistory. Když je o tom řeč – nezapomeňte, že LED jsou sice málo náročné na proud, ale když jich je 64, tak se to nasčítá…

16 Jak vypadá hradlo uvnitř

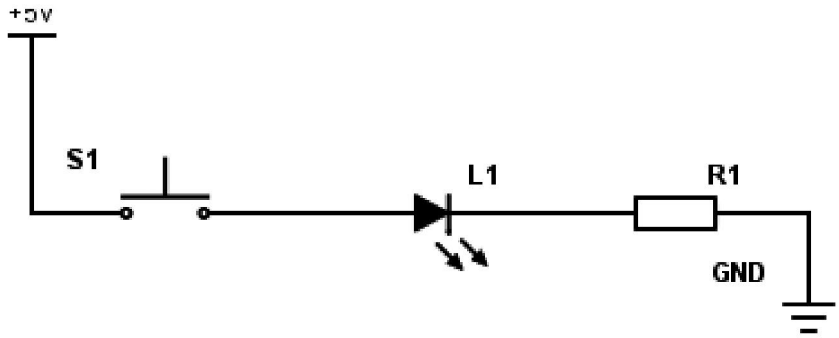
16 Jak vypadá hradlo uvnitř

Tahle kapitola není povinná. Nemusíte ji znát, ale je dobré, když se k ní třeba za čas vrátíte. V ní totiž zjistíte, jak jsou hradla udělaná „vevnitř". Jak je poskládané z těch tranzistorů či čeho, jaký je rozdíl mezi TTL a CMOS, co se děje, když se něco děje, proč se řídicí signály často používají negované, proč se k obvodům zapojuje odrušovací kondenzátor…

Tranzistor je prvek, který je vlastně základním stavebním kamenem celé číslicové techniky. Na rozdíl od analogové, kde máme tranzistory rádi, protože zesilují slabé signály, tak v číslicové spíš využíváme toho, že tranzistor se za jistých podmínek sepne pro procházející proud, a to podle proudu, který prochází bází.

Základní prvek je invertor. Musíme nějak dokázat, aby na výstupu byl proud, pokud na vstupu není, a obráceně.

Začneme úplně od lesa. Zapojte si do série tlačítko, LED a rezistor tak, aby LED svítila, když zmáčknete tlačítko. Máte? Nějak takhle to bude:



A teď výzva: Zapojte to tak, aby to fungovalo obráceně, totiž aby LED ZHASLA, když zmáčknete tlačítko.

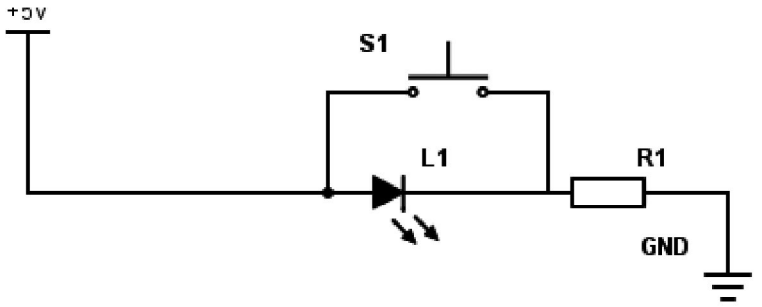
Chvíle pro přemýšlení...

Ještě chvíle pro přemýšlení...

Máte to?

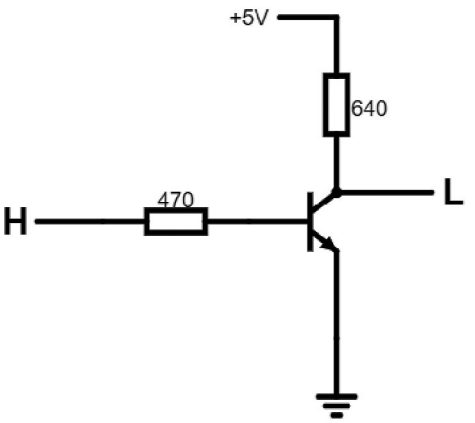
Napovím: Proud jde cestou menšího odporu. A co má menší odpor – tlačítko, nebo dioda? Navíc jsme to probírali už u fotorezistoru a detektoru tmy!

Dobře, tak takhle:



Když tlačítko pustíte, jde proud přes rezistor do diody. Když tlačítko zmáčknete, teče místo toho většina tím tlačítkem.

No, a teď využijeme toho, že tranzistor je vlastně takový „proudem řízený spínač“. Postavte si toto:



Princip je jednoduchý: Když je na vstupu logická 0 (L), tedy vstup je spojen (více méně) se zemí, je tranzistor zavřený, a proud teče přes rezistor 640 ohmů z napájecího napětí na výstup. Jakmile na vstup přivedeme log. 1 (H), tranzistor se otevře a spojí výstup se zemí. Na výstupu bude tedy logická 0.

⌀ https://eknh.cz/rtlinverter

Hradlo NAND bude na podobném principu:

⌀ https://eknh.cz/rtlnand

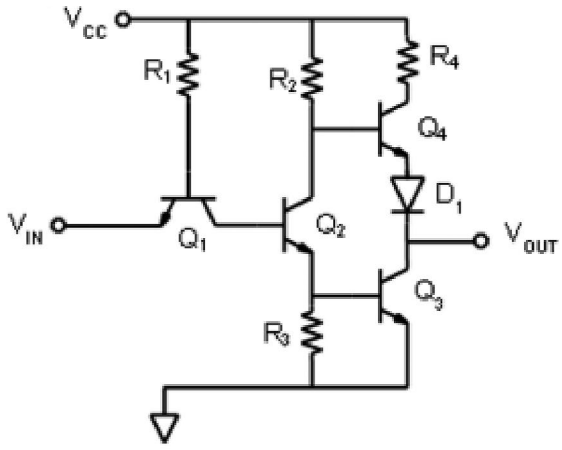
Pokud tranzistory zapojíme paralelně, namísto sériově, získáme hradlo NOR.

⌀ https://eknh.cz/rtlnor

Takovéhle logické elementy opravdu existovaly. Říkalo se jim RTL – Resistor-Transistor Logic – a už jsem psal, že byly použité například v naváděcím počítači Apolla. Měly ale obrovskou spotřebu (podívejte se, jaké proudy tečou hradlem NAND, když jsou oba tranzistory otevřené), nebyly moc odolné proti rušení atd. Vývoj proto přinesl další technologii, a tou byla TTL, neboli transistor-transistor logic.

Výhodou TTL proti RTL je nižší spotřeba a větší rychlost. Nevýhodou pak to, že potřebujete výrazně vyšší počet tranzistorů na logickou funkci. Na výše uvedeném invertoru je to vidět – tranzistor úplně vlevo, připojený emitorem ke vstupu (Q1), slouží ke spínání centrálního tranzistoru (Q2). Pokud je vstup na nízké logické úrovni, teče proud skrz první tranzistor ven (ano, teče proud ze vstupu…) Při napájecím napětí 5 V a omezovacím odporu R1 o velikosti 100k jde o necelých 50 uA. Centrální tranzistor Q2 je tak uzavřen, a proud teče přes rezistor R2 do báze tranzistoru Q4. Ten je tak otevřen, a na výstup je přes R4 a D1 přivedeno napájecí napětí. Q3 je uzavřen díky odporu R3 v bázi.

Pokud na vstup přivedeme logickou 1 (H), dostane se toto napětí na bázi Q2, který se otevře. Proud nyní putuje přes R2 a Q2 do báze Q3. Q4 je tedy zavřený, Q3 otevřený, a výstup je propojen přes něj se zemí.



Toto řešení výstupu pomocí dvou tranzistorů, rezistoru a diody se nazývá též *totem*. Připomíná Darlingtonovo zapojení, které dovoluje spínat velké proudy. Díky tomu lze výstup zatížit poměrně velkou zátěží. U standardní řady TTL lze na jeden výstup hradla připojit až deset vstupů.

16.1 Proč zapojovat blokovací kondenzátory k napájení

Nevýhodou totemu je, že spoléháme na nekonečnou rychlost přepínání tranzistoru Q2. V ideálním případě je tento tranzistor buď plně otevřen (a Q4 je tedy zavřený, Q3 otevřený), nebo plně zavřen (Q4 otevře, Q3 zavře). Jenže reálné obvody vykazují pomalejší přechod, daný převážně fyzikálními vlastnostmi. Na krátký okamžik se tak při přepínání stane, že budou otevřeny oba tranzistory v totemu, jak Q3, tak Q4, a poteče přes ně proud, omezený jen rezistorem R4. V praxi se to projeví rušivými pulsy, kdy je zvýšený odběr z napájení. Proto se připojuje k napájení obvodů TTL paralelně kondenzátor o kapacitě ~ 100 nF, který tyto špičky pokryje. Zároveň z toho vyplývá, že *čím větší je přepínací frekvence, tím vyšší je odběr obvodu*.

16.2 Negované signály

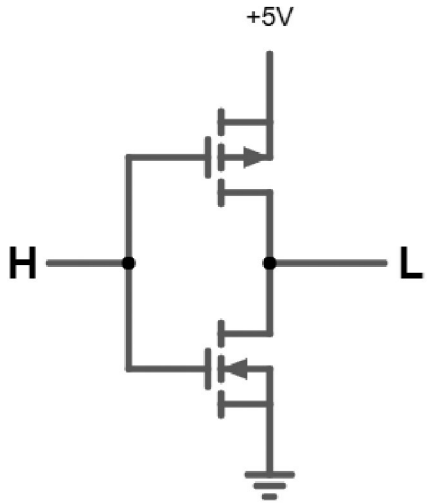
Možná jste si říkali, proč jsou některé signály v číslicové technice negované. Typicky třeba signál RESET a signály řízení sběrnice bývají navržené tak, že jsou aktivní v log. 0, a v klidovém stavu jsou v log. 1. Důvodem je rušení. Představte si, že se ve vedení indukuje rušivé napětí, a tak místo 5 V tam je vlivem rušení třeba o 1,5 V méně, tedy 3,5 V. Pokud je signál navržený jako invertovaný, nevadí to, stále se vejdeme do dovoleného pásma (2 V-5 V). Pokud bychom ale měli signál v klidu na log. 0, tak by 1,5 V rušivého napětí znamenalo, že se signál dostane mimo bezpečnou oblast (0 V-0,8 V), a obvod takový stav může vyhodnotit jako aktivní impuls. Zkrátka díky tomu, že logická 1 má mnohem větší pracovní pásmo, je výhodnější u signálů, které mají být po většinu doby v klidu a sepnout jen někdy, použít invertovanou hodnotu a nadefinovat klidový stav jako 1.

16.3 MOS, CMOS

Až do této chvíle jsme si popisovali obvody, nazývané jako „bipolární". To proto, že používají takzvané *bipolární tranzistory* NPN. Místo bipolárních tranzistorů můžeme ale použít tranzistory MOSFET. Už jsem je představoval, tak to jen bleskem zopakuju: FET znamená Field-Effect Transistor, MOS odkazuje na jeho uspořádání: Metal-Oxid-Semiconductor. Na rozdíl od bipolárních tranzistorů, řízených protékajícím proudem, jsou tranzistory MOS řízeny napětím. V praxi to znamená, že vstupem takového obvodu teče téměř nulový proud a chová se tedy jako by měl (téměř) nekonečný odpor.

N-MOS se spíná pozitivním napětím řídicí elektrody (gate) proti společné (source). Technologie P-MOS to má přesně obráceně. Technologie MOS má mnoho výhod, a jednou z nich je to, že se na stejnou plochu křemíkového čipu vejde víc logiky. Proto se obvody TTL dělaly nanejvýš ve středním stupni integrace (MSI – čítače, dekodéry, multiplexery). Vyšší stupeň integrace (LSI, VLSI), potřebný pro složitější obvody včetně mikroprocesorů, vznikaly technologií MOS. Nejprve šlo o P-MOS, ale později přišla rychlejší NMOS, bohužel s vyšším odběrem.

Pokud se v logickém prvku zkombinovaly obě tyto technologie, vznikl CMOS (Complementary MOS).



Vidíte, že díky použití dvou rozdílných tranzistorů, které jsou zapojené „proti sobě“, získáváme téměř ideální invertor, kterým v klidovém stavu neteče téměř žádný proud. Spotřeba je tedy nula nula nic. Ovšem – jen teoreticky. V praxi je vlivem technických nedokonalostí a parazitních kapacit situace spíš takováto:

Při přepínání krátkodobě vyskočí proud protékající obvodem. Na druhou stranu oproti hradlům TTL je to vysloveně nic. Obvody CMOS (řady 40xx) mají proti TTL mizivou spotřebu, mohou pracovat v širokém rozmezí napájecího napětí, ale na druhou stranu jsou pomalejší a jejich úrovně jsou nekompatibilní s TTL.

Pokud obvody CMOS napájíme 5 V, je pro ně logická 0 na vstupu v rozmezí 0 V až 1,3 V. Logická 1 je 3,7 V až 5 V. Na výstupu je v log. 0 maximálně 0,2 V, v log. 1 pak minimálně 3,7 V. Vstupy CMOS mají hodně velké „zakázané pásmo“ (1,3 V až 3,7 V), a střední hodnota (prahové napětí pro přepnutí) je okolo 2,5 V – tedy už v oblasti, kde je TTL logická 1.

Naštěstí v dnešní době technologie pokročila dopředu., a dnešní moderní CMOS jsou mnohem rychlejší, a třeba řada HCT má napěťové úrovně kompatibilní s TTL a představuje tak reálně „to nejlepší z obou světů“.

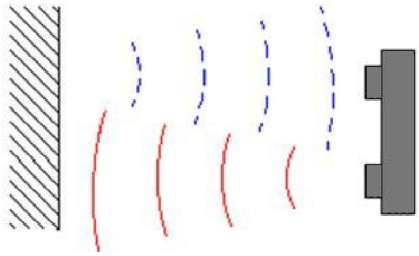
17 „Plnou parou vzad!“ – „Ale jak daleko?“

17 „Plnou parou vzad!“ - „Ale jak daleko?“

Ne, nezbláznil jsem se, nebojte… Chtěl jsem vás jen nečekaným titulkem připravit na další pokus. Na chvíli si zase odpočineme od kombinačních obvodů a postavíme si něco s Arduinem.

Asi víte, jak funguje takzvaný **parkovací asistent** v autě. Je to takové nenápadné zařízení na palubní desce, které ukazuje, jak daleko za zadním nárazníkem máte překážku – tedy jiné auto, zeď, chodce, …

Fungování takového zařízení je jednoduché: v nárazníku jsou namontované reproduktory, které vysílají zvukový signál. Ten se odrazí od překážky a vrátí se zpět. Reproduktor v tu chvíli funguje jako mikrofon, odraženou vlnu zachytí a převede ji na impuls. Řídicí elektronika jen počítá čas, který uplyne od vyslání impulsu („pípnutí“) do jeho příchodu zpátky.



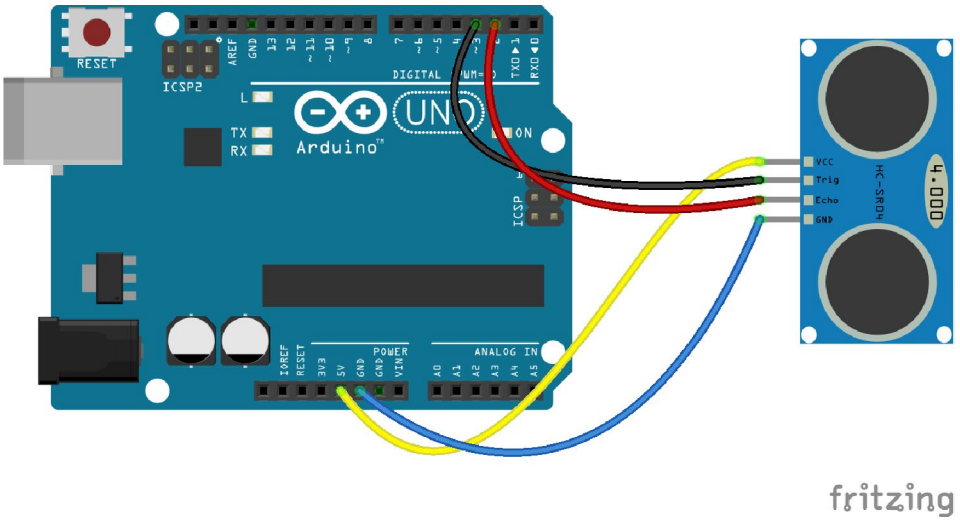
Říkáte si, že zvuk letí hodně rychle? No, jak se to vezme. Rychlost šíření zvuku ve vzduchu záleží na mnoha faktorech, ale zhruba lze říct, že za jednu sekundu uletí 340 metrů. Což se zdá jako velká vzdálenost, ale když si uvědomíte, že v elektronice pracujeme běžně s mnohem kratšími časy, třeba milisekundami, a že takové Arduino udělá 16 operací každou mikrosekundu, tak to zas tak závratná rychlost není, že? Za jednu milisekundu zvuk urazí tedy pouhých 34 centimetrů. A když si uvědomíte, že musí letět k překážce a zpátky, tedy urazit dvě cesty, tak vám vyjde, že pokud se zvuk vrátí za jednu milisekundu, tak je překážka vzdálená pouhých 17 centimetrů.

Parkovací asistent většinou používá čtyři takové senzory, které slouží zároveň jako reproduktor i mikrofon. Pro amatérské konstrukce jsou dobře dostupné **senzory HC-SR04**, které používají dvojici reproduktor-mikrofon.



Tyto levné moduly jsou uzpůsobené pro napájení 5 V (vstupy GND a VCC), a mají dva další vývody: Trig, kterým se spouští ono „písknutí“, a Echo, které oznamuje, že mikrofon přijal signál. Délka pulsu na výstupu Echo odpovídá naměřené vzdálenosti. Modul pracuje s ultrazvukem na frekvenci 40 kHz, což je vysoko nad tím, co dokáže slyšet lidské ucho (to slyší tóny zhruba v rozsahu 20 Hz až 20 kHz).

Postavit si měřič vzdálenosti s takovým modulem a Arduinem je velmi snadné – stačí propojit vývody Trig a Echo s datovými piny a zapojit napájení podle obrázku.



Já zapojil Echo na pin 2 a Trig na pin 3.

Obslužný program má jen pár řádků – ve funkci Setup je potřeba nastavit správně vstupy a výstupy, a ve smyčce pak posílat pulsy, dlouhé 10 mikrosekund na vstup Trig (pin 3) a čekat na odpověď na pinu 2. K tomu slouží funkce pulseIn(), která měří délku pulsu (v mikrosekundách).

```
#define ECHOPIN 2 // Echo pin z HC-SC04 na pin 2
#define TRIGPIN 3 // Trig pin z HC-SC04 na pin 3
void setup() {
    //Nastaví sériovou komunikaci
    Serial.begin(9600);
    //Nastaví pin 2 jako vstupní
    pinMode(ECHOPIN, INPUT);
    //Nastaví pin 3 jako výstupní
    pinMode(TRIGPIN, OUTPUT);
```

```
    digitalWrite(TRIGPIN, LOW);
}

void loop() {
    // Vyšle impuls do modulu HC-SR04
    digitalWrite(TRIGPIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIGPIN, LOW);
    // Spočítá vzdálenost
    float distance = pulseIn(ECHOPIN, HIGH);
    distance= distance*0.017315f;
    // odešle informace na sériový port
    Serial.print(distance);
    Serial.print(„\n„);
    //počká 1 sekundu
    delay(1000);
}
```

17.1 Ještě pípat!

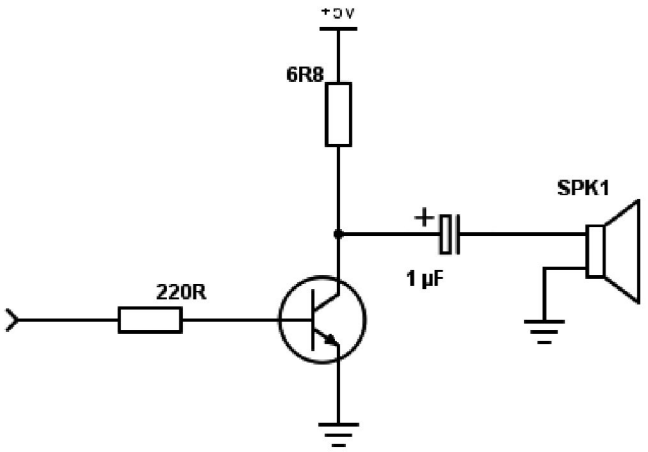
Takový parkovací senzor ale většinou taky pípá, to aby řidič nemusel sledovat displej a mohl se soustředit na couvání. Otázka tedy zní: Jak udělat zvuk? Jak udělat to pípání?

Asi víte, že zvuk je vlnění, přenášené hmotou, že podle počtu kmitů za sekundu člověk rozlišuje výšku tónu. Možná nevíte, že tón o dvojnásobné frekvenci vnímáme jako o „oktávu vyšší“, ale můžete si to sami vyzkoušet s plastovým pravítkem a stolem.

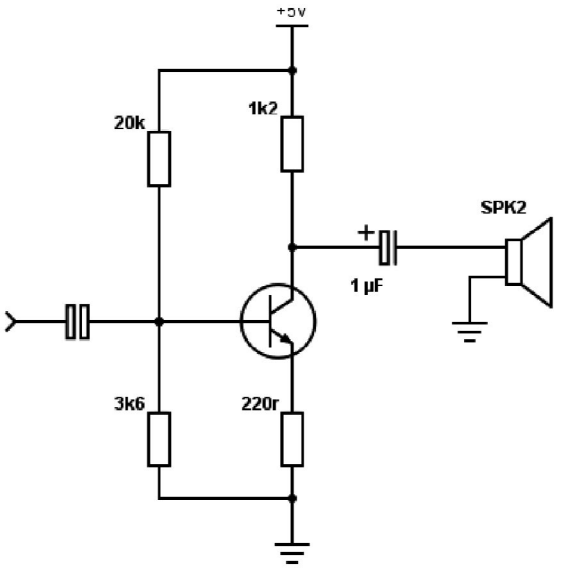
Máte? A jste v tiché kanceláři, kde to vzbudí pořádnou nevoli u kolegů? Pokud ano, tak jděte raději vedle. Položte pravítko na stůl tak, aby bylo ve vzduchu třeba svou polovinou. A drnkněte. Slyšíte ten rachot? Tak, a když teď posunete pravítko tak, aby přesahovala jen čtvrtina, uslyšíte rachot o oktávu vyšší. Kytaristi to znají – když stisknou strunu E přesně uprostřed, bude hrát zase E, ale o oktávu výš.

No dobře, a co tedy my s elektronikou? Jak můžeme udělat zvuk? Máme na výběr několik možností. Jedna z možností je reproduktor se zesilovačem. Bez zesilovače to není moc dobrý nápad, protože reproduktor potřebuje velký proud (má malý odpor).

Samozřejmě existuje jednoduché zapojení tranzistorového zesilovače pro reproduktor. Na webu naleznete desítky jednoduchých zapojení, která budou vypadat třeba nějak takto:

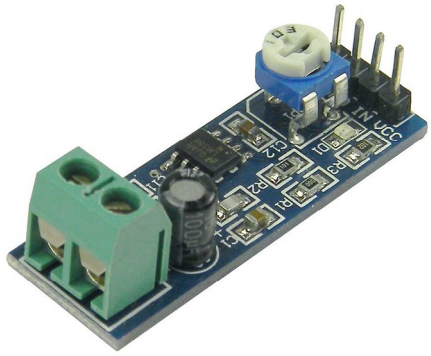


Zkušený elektronik v nich odhalí hned několik nectností (například chybějící oddělovací kondenzátor na vstupu) a z hlavy nakreslí mnohem vhodnější zapojení:



Takovéhle zapojení má mnohem menší zkreslení, teplotní stálost, nižší šum a je „čistší“ než to předchozí, ale ty hodnoty rezistorů nespadly odněkud shůry, ty si musíte poměrně zdlouhavě spočítat podle napájecího napětí a typu tranzistoru…

Hodnoty rezistorů jsem se jako mladý elektrotechnik učil počítat, a zařekl jsem se, že to v téhle knize nebudu dělat. Já vás nechci zmučit počítáním něčeho, co si můžete koupit hotové v různých obchodech s elektronickými moduly. Hledejte „audio amplifier module“, třeba s obvodem LM386:



Do zelených svorek zapojíte reproduktor. Čtyři piny na opačné straně jsou označené „GND – GND – IN – VCC“. Krajní dva (GND a Vcc) slouží pro napájecí napětí reproduktoru (5 až 12 voltů), prostřední dva (GND a IN) jsou audio vstup. GND připojte na společnou zem s Arduinem, IN na některý z datových pinů v Arduinu. *Vynechte piny 0 a 1, ty slouží ke komunikaci, a kdybyste připojili zesilovač na ně, asi byste se při programování nestačili divit. Ale klidně si to zkuste…*

Tak, tohle by bylo. Dejme tomu, že jste zapojili zesilovač na pin 4. Jak teď udělat pípnutí?

No, asi vás to nepřekvapí: použijte zase stejný kód jako pro blikání LEDkou, jen místo delay(1000) použijte třeba delay(1) – parametr funkce delay() říká, kolik milisekund se má čekat, pokud budete před každým přepnutím čekat 1 ms, bude celková doba jednoho taktu rovna (zhruba) 2 ms, tedy frekvence bude okolo 500 Hz (f {Hz} = 1 / T {s}).

Nechte cyklus proběhnout třeba pětsetkrát – tím získáte jednosekundové pípnutí tónem o výšce 500 Hz, což je tón, zhruba odpovídající komornímu H (komorní A je tón s frekvencí 440 Hz, ten náš je o kousek vyšší).

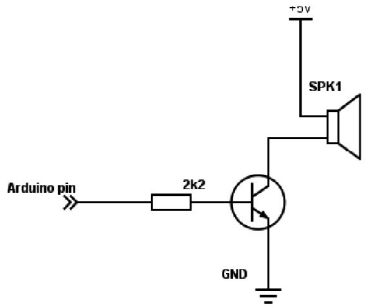
Druhá možnost je použití piezobzučáku. Vypadají jako takové malé válečky s otvorem a dvěma nožičkami.



Takových bzučáků je mnoho typů – některé vyžadují vyšší napětí, třeba 12 voltů, další se spokojí s pěti volty. Některé z nich při zapojení stejnosměrného napětí pískají samy na nějaké frekvenci, dané konstrukcí, jiné fungují jako maličké reproduktorky. Velké množství konstrukcí je připojuje přímo k Arduinu na digitální piny.

Já bych tu byl trochu opatrnější a zvolil bych způsob, při kterém se bude spínat větší proud, který… aha, už víte? Že už je to pro vás obnošená vesta? To rád slyším. Tak jak to bude?

No, pro spínání použijeme tranzistor (správně), zvolíme běžný NPN, protože ten se spíná kladným napětím proti zemi, a použijeme to nejjednodušší zapojení, kde tranzistor nezesiluje proud, jen funguje jako spínač, protože to potřebujeme. Tedy nějak takto:



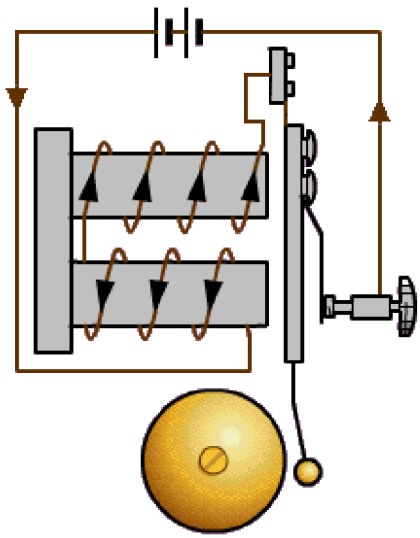
Mezi Arduinem a tranzistorem je ten rezistor proč? Aha? No protože Arduino pustí na výstup ve stavu logické 1 svých 5 voltů, a bez vhodného omezení by tekl maximální proud. S rezistorem jeho velikost omezíte – rezistor 2K2 pustí při 5 voltech něco málo přes 2 mA, což je proud bezpečný a Arduino to ustojí!

Pamatujte si, že jediným pinem by neměl téct proud větší než 20 mA, a že není rozumné pouštět do mnoha pinů najednou velký proud, protože snadno překročíte interní maximální proudy a Arduino se vám bude zbytečně přehřívat, nebo se dokonce spálí.

18 Zpětná vazba

18 Zpětná vazba

Když už jsme tu nakousli to pípání a bzučení – pamatujete se na staré elektrické domovní zvonky? Tam byl elektromagnet a kovová kotva, která břinkala do toho kovového zvonivého objektu. Jakmile se pustil proud, elektromagnet přitáhnul kotvu – ale tím se zároveň přerušil obvod, takže elektromagnet pustil, kotva se vrátila zpět, ale tím opět sepnula obvod, takže elektromagnet přitáhnul...

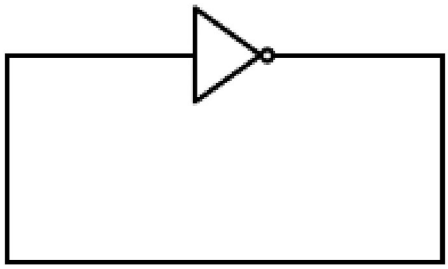


Výsledek tedy byl ten, že s určitou frekvencí, danou mechanickými vlastnostmi té kotvy, především její setrvačností, střídalo zařízení dva stavy.

⌀ https://eknh.cz/zvnk

A teď myšlenkový pokus, ano? Máme invertor. Víme, že když má na vstupu 0, má na výstupu 1. Když má na vstupu 1, má na výstupu 0. Zkrátka je vždycky v opozici, asi jako ten elektromagnet s kotvou u zvonku... Co kdybychom ho donutili, aby byl v opozici se sebou samotným?! Vytvořili bychom paradox? Zakázaný stav? Nestvořili bychom časoprostorovou trhlinu? Nebo by to fungovalo jako ten zvonek?

No, schválně si to představte. Na vstupu je 0. Tedy na výstupu je 1. Tu přivedeme na vstup, takže na výstupu bude 0. Tu přivedeme na vstup, takže na výstupu bude 1. Tu přivedeme na vstup, takže... Takže se to celé buď ustálí v nějakém stavu „ani ryba, ani rak, logická hodnota 0.5“, nebo se to rozkmitá, že?



Bé je správně! Invertor má totiž taky určité dynamické vlastnosti, stejně jako ta kotva, a nepřepíná úplně hned, chvilku mu to trvá. Jakou chvilku? No, u toho starého dobrého TTL invertoru 7400 to bylo okolo 10 ns. Řada 74ALS má okolo 4 ns, řada 74S okolo 3 ns, „rychlá" řada Fast (74F) přepne za 2.5 ns. Nízkoodběrová řada 74L měla zpoždění 33 ns (používám minulý čas, protože i tato řada je dnes minulostí). Ve skutečnosti to bylo ještě složitější, protože zpoždění se lišilo podle toho, jestli se přepíná z 0 do 1, nebo z 1 do 0, ale to můžeme v našem případě zanedbat.

Většinou nevadí, když do zapojení dáte místo obvodu 74LS00 třeba obvod 74ALS00 nebo 74L00. Nějak to fungovat bude. Ale u složitějších (a rychlých) obvodů může pomalejší kus způsobovat velmi výrazné změny funkce (většinou směrem k horšímu, jak jinak). Někdy to platí i obráceně – že zapojení, které stavíte, buď schválně, nebo v důsledku „race condition", bude fungovat pouze s pomalejší řadou, protože autorovi zpoždění někde k něčemu pomohlo.

Pokud je zpoždění 10 nanosekund, bude frekvence, na které to celé kmitá, rovna 1/t, tedy 100 MHz. Což je hodně moc. Já vím, že s 2.4 GHz rádiem a třígigovými procesory v notebooku vám to nepřipadá, ale v číslicové technice, kde se budeme pohybovat my, je vrcholem tak těch 24 MHz pro procesor 8052, ale ještě častěji spíš 16 MHz v Arduinu. Staré procesory běží na 2 MHz, popřípadě 4 MHz. Takže pokud se něco rozkmitá na 100 MHz, věští to problémy.

18.1 Astabi-cože?

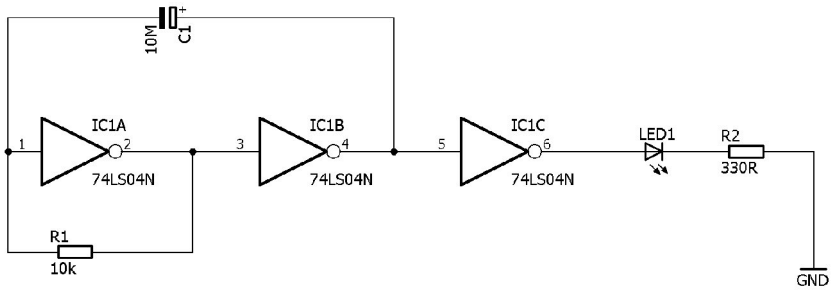
Stvořili jsme astabilní klopný obvod. **Klopné obvody** jsou obvody, které definovaným způsobem mohou měnit svůj stav v čase. Rozlišujeme tři základní druhy. Klopný obvod *astabilní* nemá žádný pevný, stabilní stav. Střídá na výstupu 1 a 0 a ignoruje přitom okolí. Druhý typ obvodů je *monostabilní*. Má jeden stálý stav (třeba log. 0), ale na základě nějakého vnějšího vlivu může přejít do druhého stavu (třeba log. 1), z něhož se ale po čase vrátí zpět do toho stálého. No a konečně třetí typ jsou *bistabilní* klopné obvody. Ty mohou být ve dvou různých stavech, a jakmile v jednom z nich jsou, jsou v něm neomezeně dlouho – až dokud okolí nezpůsobí jejich přepnutí.

Bistabilní obvod bych přirovnal třeba k vypínači od lustru: je buď zapnutý, nebo vypnutý, a setrvává v tom stavu až do doby, než ho vnější vliv nepřiměje stav změnit. Monostabilní obvod v téže analogii funguje jako schodišťové osvětlení: stisknutím rozsvítíte, a po čase samo zhasne.

Astabilní obvody (oscilátory) se používají v číslicové technice často, a to ke generování časových pulsů o nějaké frekvenci. Příklad: chcete si postavit digitální hodiny. Začnete tedy tím, že si postavíte astabilní klopný obvod, který bude generovat pulzy s frekvencí 1 Hz a víte, že co puls, to sekunda. *Ve skutečnosti ale použijete obvod, který bude kmitat na frekvenci vyšší (třeba 32,768 kHz) a několikerým dělením získáte kýžený 1 Hz. Proč? Protože pro 32,768 kHz seženete přesnou součástku, zvanou „krystal", která se o frekvenci postará.*

Ještě častěji se potkáte s bistabilními obvody. Důvodem je to, že si pamatují svůj stav, což se hodí pro nejrůznější účely, od prostého zapamatování hodnoty po konstrukci složitějších obvodů, jako jsou děličky frekvencí, čítače, posuvné registry a spousta dalších zábavných komponent.

18.2 Blikač



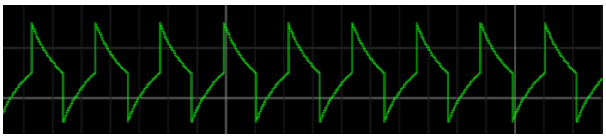
Vzpomínáte si na blikač z první kapitoly? Jak ten vypadal?

Co se tu děje? Na vstupu 1, úplně vlevo, je po zapnutí dejme tomu logická 0. Na výstupu 2 je tedy logická 1, ta je i na vstupu 3. Na výstupu 4 je zase logická 0, no a invertor IC1C z ní dělá logickou 1 na výstupu celého zapojení. Důvod, proč je tu použitý vlastně na první pohled „zbytečný" invertor, je ten, že kondenzátor C1 by se mohl nabíjet či vybíjet přes obvody, co jsou zapojené na výstup – tady třeba přes LED. Tím, že zařadíte další invertor, vlastně oddělíte vnitřní logiku blikače od ovládaných součástek.

Značení IC1A, IC1B atd. napovídá, že se jedná stále o jeden integrovaný obvod, konkrétně číslo 1, a v něm používáme hradla A, B, C. IC je z anglického Integrated Circuit. Někdy se v českých schématech setkáte s označením IO („integrovaný obvod"), zahraniční časopisy občas používají „U" („unit")…

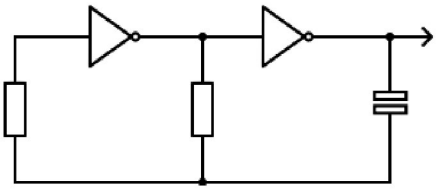
Rezistor přenáší napětí z výstupu 2 – tam je logická 1 – zpátky na vstup. V předchozím zapojení, bez kondenzátoru, by se invertor IC1A rovnou přepnul. Jenže tady část proudu sebere kondenzátor. Bude se pomalu nabíjet, napětí poroste, a v určitém okamžiku překročí rozhodovací hranici. V tu chvíli IC1A přepne, na výstupu bude 0, na výstupu druhého invertoru tedy jednička, na výstupu třetího nula.

A celý proces pojede obráceně. Teď se bude naopak kondenzátor vybíjet. A jakmile napětí klesne pod danou mez, zase se celý obvod přepne. Průběh napětí na vstupu 1 bude vypadat nějak takto:



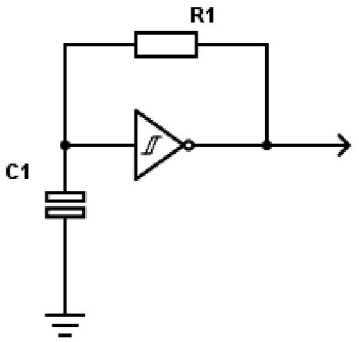
Teď už vám je jasné, že mnohem lepší a čistší by bylo použít invertory, které mají na vstupech Schmittův obvod, tedy typ 7414.

Samozřejmě, že toto není jediný typ astabilního obvodu. Existují i další zapojení:

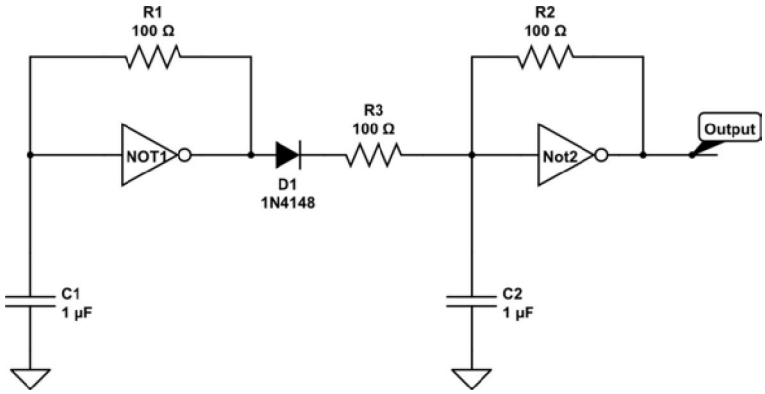


Frekvence kmitání takovýchto oscilátorů je zhruba 1 / (2,2.R.C) (Hz; Ω, F)

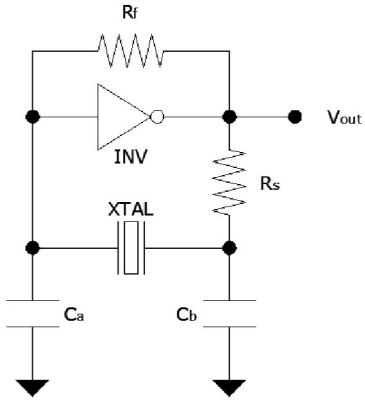
Existují i „oscilátory chudého muže“ s jedním invertorem:



nebo s dvěma kondenzátory



Ovšem v praxi se používá buď přesnější časovač 555 (NE555), nebo krystalový oscilátor, kde je kondenzátor nahrazen součástkou, zvanou „krystal“, což je opravdu kousek křemíkového krystalu, který mění svou kapacitu s určitou frekvencí, která je velmi přesná a stabilní.



Krystalové oscilátory se používají všude tam, kde je potřeba mít přesný zdroj pulsů.

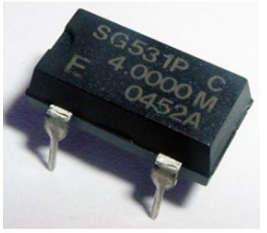
18.3 Krystalový oscilátor DIL

Většinou platí, že když je něco v elektronice šikovné a užitečné, tak se objeví někdo, kdo to bude vyrábět. Totéž platí i pro krystalové oscilátory.

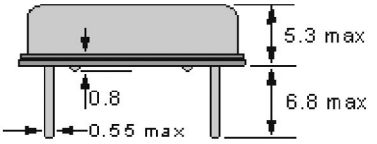
Krystal potřebuje ke své práci invertor a několik dalších součástek. Proto se, logicky, nabízí varianta „uzavřít to všechno do jednoho pouzdra“. Nějak takto:



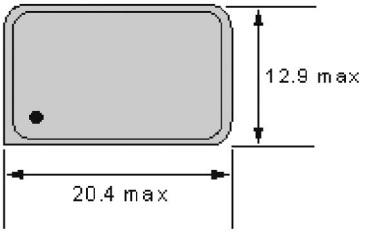
Nebo takto:

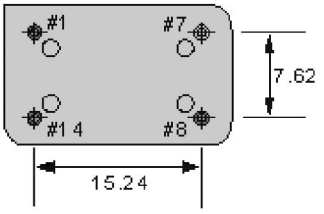


Nejčastěji tedy v podobě pouzder DIL8 nebo DIL14, ačkoli jsou osazené jen čtyři. Vývody 7 a 14 bývají většinou napájecí, stejně jako třeba u 7400, vývod 8 (křížem naproti zvýrazněnému vývodu 1) bývá výstup, vývod 1 není zapojen, nebo funguje jako „povolovací“ (1 = oscilátor běží).



PIN	CONNECTION
1	NC (OE)
7	GND
8	OUTPUT
14	V_{DD}



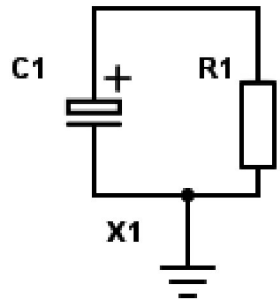


Takovéto oscilátory se vyrábějí v širokém rozmezí frekvencí, od 1 MHz po stovky MHz. Vyrábějí se i v provedení „low voltage“, v provedení pro povrchovou montáž a v dalších.

18.4 Monostabilní klopný obvod

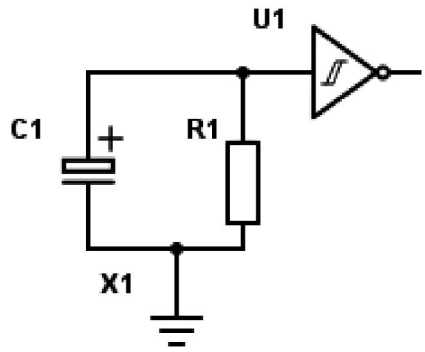
Pojďme si zase něco vymyslet… Třeba… Pamatujete se na to zapojení s tlačítkem, kondenzátorem a LEDkou, jak se tlačítkem nabil kondenzátor a pak se pomalu vybíjel přes rezistor a svítila nám LEDka? Dalo by se to nějak zapojit s integrovaným obvodem?

Co třeba takhle… Bude tam kondenzátor a rezistor. Kondenzátor se nabije, nějak, to teď nebudeme řešit. No a pak se bude pomalu vybíjet přes ten rezistor, a až se vybije pod určitou mez, tak to nějak přepne výstup.



Tak, a teď dvě otázky: jak zajistit ten výstup, a jak zajistit to úvodní nabití?

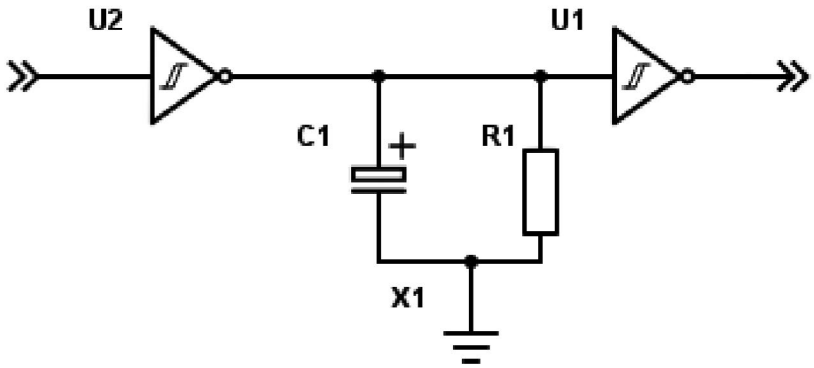
Výstup bude jednodušší: prostě na tu horní část připojíme invertor, ideálně se Schmittovým vstupem. Takto:



Na výstupu bude tedy v klidu logická 1, když bude kondenzátor nabitý, bude na výstupu 0, a jak bude napětí klesat, tak v určitém bodě se invertor opět přepne do 1.

To bychom měli. Teď jak zajistit to nabití? Co třeba stejně, tedy přivést do téhož místa výstup z invertoru. Ten dodá dostatečný proud pro nabití kondenzátoru. Je důležité, aby ten startovací puls měl nějakou minimální délku, aby se kondenzátor stihl nabít. Čím větší kondenzátor, tím delší čas pro nabití.

Zkusíme to tedy takhle:



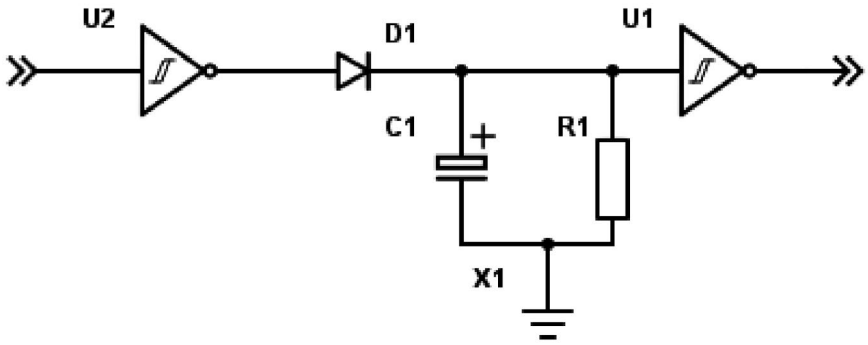
Otázka za sto bodů: proč jsem použil na vstupu invertor se Schmittovým obvodem, když by tam mohl být klidně obyčejný? Nemá to žádnou komplikovanou příčinu, prostě jen vím, že invertory se dělají v pouzdrech po šesti, a když použiju obvod 7414, mám v něm šestici invertorů se Schmittovým obvodem na vstupu. Tak ho rovnou použiju, když je v témže pouzdru. Proč bych tam dával další pouzdro, že?

Zkusíme to, ale nebude to fungovat. Fakt, klidně si to zkuste, ať vidíte sami, že to nic nedělá. Proč?

Na vstupu by měla být 1, na výstupu 0. To je klidový stav. Pak na vstup přivedete 0. Na výstupu bude 1, kondenzátor se nabije… oukej… na vstup zase vrátíme 1, na výstupu bude tedy nula, a… Aha! Už jsem to říkal, ale zopakujme si: **Když je na výstupu 0, znamená to, že uvnitř obvodu je výstup spojen se zemí.** Takže jakmile se výstup levého invertoru přepne do 0, tak se kondenzátor bleskurychle vybije právě přes ten výstup.

Chtělo by to nějakou součástku. Nějakou takovou, která by pustila proud jen z výstupu ven, ale už ne zpátky dovnitř. Nějakou, co pouští proud jen jedním směrem…

A zatímco přemýšlíte, jaká součástka by to mohla být, tak já si tu něco jen tak nakreslím:



Bude to fungovat? No, zkuste si.

⌀ https://eknh.cz/mono

Celé téhle báječné věci se říká „monostabilní klopný obvod". Jako že má jeden stabilní stav (1), vnější puls jej na chvíli přepne do druhého stavu (0), a obvod se po čase vrátí zpátky do stabilního. Proto mono-stabilní.

Délku trvání výstupního pulsu nastavíte kombinací kapacity C1 a odporu R1. Délka pulsu je přibližně 0,8.R.C (dosaďte ohmy, farady a výsledek vyjde v sekundách). Přesněji řečeno: doba, o kterou je opožděno přepnutí obvodu proti konci pulsu na vstupu. Dejme tomu: kondenzátor 10 μF, rezistor 100 kΩ, čas by tedy měl být zhruba 0,8 sekundy. Jakmile přivedete na vstup 0, výstup se taky přepne do nuly. Dokud bude na vstupu nula, na výstupu bude taky nula. Od okamžiku, kdy se vstup vrátí do 1, to bude trvat 0,8 sekundy, než se do stavu 1 vrátí i výstup.

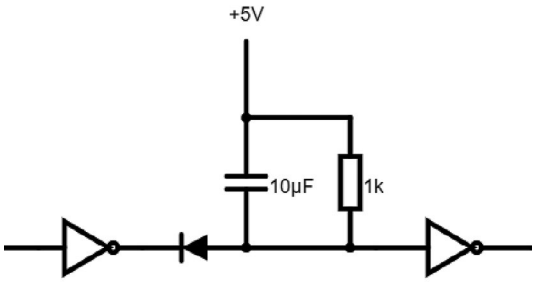
Pokud někdy během té 0,8 sekundy přijde další impuls, začne se počítat až od jeho konce. Když tedy budete posílat na vstup pulsy třeba každých 0,5 sekundy, obvod se bude znovu a znovu spouštět, a výstup bude stále 0. Takovým monostabilním klopným obvodům se říká *znovuspustitelné* (retriggerable) – vstupní puls jakoby znovu spustil odpočet. Existují i takové MKO, které v aktivním stavu ignorují vstupní signály, a je možné je znovu aktivovat až poté, co se přepnou zpět do klidového stavu (nonretriggerable).

Zkuste si, pro zajímavost, vymyslet, jak zařídit, aby se náš monostabilní klopný obvod stal *non-retriggerable*, tedy aby vždy zareagoval na puls pouze v klidovém stavu. Není to tak těžké, stačí si uvědomit, že potřebujeme vlastně jen to, aby ve stavu 0 (aktivní) neprošly na vstup žádné pulsy… Svoje řešení si můžete porovnat s tím mým:

⌀ https://eknh.cz/monotrig

A co kdybychom chtěli, aby byl klidový stav klopného obvodu v 0, a aktivní v 1? Šlo by to? Šlo, jen je třeba zajistit opačné fungování, tedy že impuls na vstupu (1, po inverzi 0) nabije kondenzátor. Jak může „nula“ nabít kondenzátor? No tak, že jej zapojíme proti napájecímu napětí. I vybíjení tak bude probíhat přes rezistor na napájecí napětí…

⌀ https://eknh.cz/monopos



18.5 Detektor pohybu

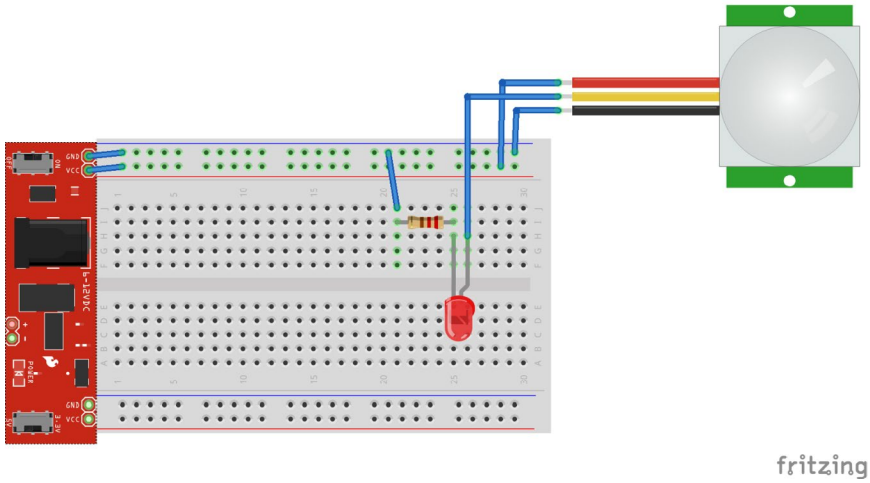
Jdeme stavět. Chcete? Já vím, že ano. Postavíme si něco s čidlem PIR.

PIR jsou pasivní infračervená čidla (proto ta zkratka: Passive Infra-Red), která zachycují pohyb objektů o specifické teplotě. Člověk mívá povrchovou teplotu těla někde mezi 35-37 °C, a jak známo: každý teplý objekt vysílá infračervené záření určité vlnové délky, která je nepřímo úměrná teplotě objektu.

Čidlo PIR má čip, citlivý na infračervené záření, před ním filtr, který odfiltruje (alespoň nějaké) cizí zdroje tepla, a okolo je takzvaná Fresnelova čočka, která koncentruje záření z okolí do jednoho bodu. Zároveň dává PIR čidlu charakteristický vzhled:



Takovéto moduly jsou dobře dostupné a velmi jednoduché na zapojení. Mají jen tři vývody – zem, napájení (+ 5 voltů) a výstup informace o tom, že se něco děje (Output). Zapojte si nějaké testovací zapojení, třeba takto:



A můžete testovat chování. LED normálně nesvítí. Jakmile před čidlem mávnete rukou, nebo se výrazněji pohnete, LED se rozsvítí asi na 1 sekundu. To je celé. Jednoduché, že?

Znáte takové ty automaty, co rozsvítí světlo na chodbě, když se někdo pohne? Zvládnete ho postavit? Já věřím, že ano… Schválně, napadá vás vhodný způsob?

Bude tam PIR, to je jasné. PIR dá při pohybu na výstupu krátký puls. Ten nestačí, to bychom udělali krok a světlo by zhaslo. Je potřeba z krátkého pulsu udělat delší…

Stačí monostabilní klopný obvod. Takový, jako jsme udělali na konci předchozí kapitoly, spínaný do logické 1. Jen ten čas by měl být delší…

Zkusme tedy dát co největší kapacitu a co největší odpor. Mně se v šuplíku válel kondenzátor 47 µF. K němu jsem našel rezistor 1M5, tedy 1,5 MΩ. Když si doplníte hodnoty do výše uvedeného vztahu, zjistíte, že čas sepnutí bude zhruba jedna minuta. A to je přijatelné.

Co dál? PIR čidlo zajistí detekci pohybu, monostabilní KO prodlouží tento impuls na minutu, a ještě nějak musíme zařídit to slíbené světlo. Pokud stačí jedna LED, není co řešit, připojte ji na výstup MKO. Ovšem případů, kdy na osvětlení chodby stačí jedna LED není zas tolik (krom domečků pro panenky), takže bude potřeba sepnout větší světelný zdroj.

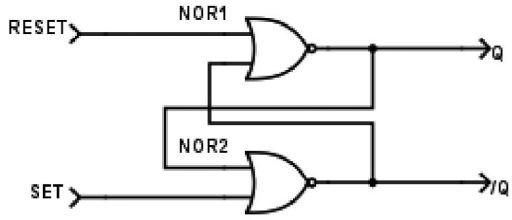
Už jsem to tu psal tolikrát, že to bude možná vypadat, že mám sklerózu a zapomínám, co jsem už napsal, popřípadě že trpím neschopností myšlenku opustit, ale zopakuju to znovu: Velký světelný zdroj = velký proud. Výstup invertoru = malý proud. **Musíte použít něco, co z malého proudu udělá velký.** Pokud budete rozsvěcovat světlo, napájené třeba pěti volty, použijte tranzistor. Pokud budete spínat třeba 230 V žárovku (na střídavý proud), použijte relé (a tranzistor k jeho sepnutí).

A na tomto místě vám velmi silně doporučím nepoužívat 230 V žárovku! Vážně! Nebo jinak: **Zakazuju vám připojovat cokoli se síťovým napětím do konstrukcí, co stavíte!** Ne že by to nešlo, jde to, teoretickou výbavu máte, ale síťové napětí, nezlobte se, tam jde o zdraví, bezpečnost a o lidské životy (bez nadsázky). Nedělejte to, **a alespoň ze začátku si k tomu vždy přizvěte někoho zkušeného, znalého techniky a oprávněného takové věci dělat**!

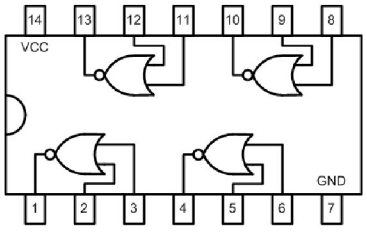
Vážně. Děkuju.

18.6 Bistabilní klopný obvod R-S

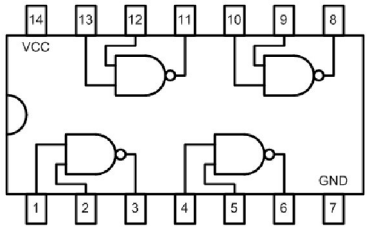
Můžeme v našich experimentech se zaváděním zpětné vazby pokračovat. Zkusíme to s dvouvstupovým hradlem, kdy do jednoho vstupu zapojíme výstup hradla, a na druhý budeme přivádět log. 1. Moc zajímavé to není. Co si takhle vzít dvě hradla a zapojit je křížem? Nějak takhle



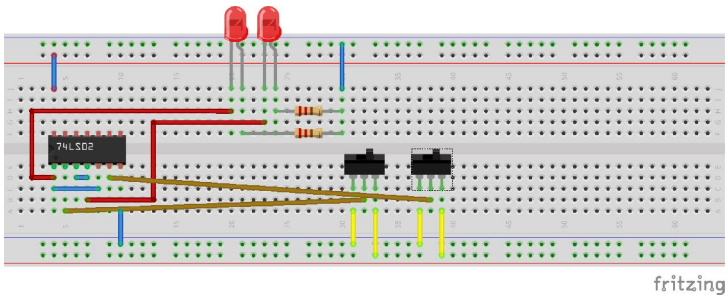
Zapojte si obvod 7402 podle tohoto schématu. **Pozor – obvod má jinak vstupy a výstupy hradel, než má 7400!** Podívejte se do datasheetu nebo na následující obrázek:



7402 Quad 2 Input NOR



7400 Quad 2 Input NAND



Nyní pozor, opět bude přemýšlení. Zkusíme si tak nějak z hlavy zjistit stav na výstupu hradla NOR1. Dejme tomu, že vstup, označený Reset, je v log. 0. Co bude na výstupu z hradla? To záleží na hodnotě druhého vstupu, a ten je připojený na výstup hradla NOR2. Jeho hodnota záleží na tom, co je na vstupu Set (dejme tomu, že i tam je 0) a na tom, co je na druhém vstupu... který je připojený na výstup NOR1... který... a jsme v nekonečné smyčce…

Ale dejme tomu (*dejme tomu!*), že na výstupu NOR2 je logická 0. Na výstupu NOR1 bude tedy 1 (0 NOR 0 = 1). Tato jednička je přivedena na vstup NOR2. Společně s nulou na vstupu Set dají dohromady nulu na výstupu NOR2 (1 NOR 0 = 0). Obvod je takto stabilní. Q je 1, not Q (označil jsme ho /Q) je 0, oba vstupy jsou nulové.

Co se stane, když na vstup Reset přivedu 1? Na vstupech NOR1 teď bude 1 a 0, výsledek bude tedy 0 a Q se změní na 0. Tato nula se ale přenese i na vstup hradla NOR2, kde spolu s nulou od vstupu Set vytvoří jedničku na výstupu NOR2 (0 NOR 0 = 1). Na výstupu /Q bude tedy 1. Tato jednička se přenese na NOR1, s jedničkou ze vstupu RESET vytvoří zase nulu, takže se na výstupu nic nemění, a tento stav bude trvat po celou dobu, co bude na vstupu Reset logická 1.

A co se stane, když se Reset teď vrátí do logické 0? Ve světě kombinačních obvodů by platilo, že pro určitou kombinaci vstupních hodnot (Reset = Set = 0) je vždy jen jedna platná kombinace hodnot výstupních. Se zpětnou vazbou to tak úplně neplatí. Chvilku se nad tím zamyslete, třeba si i namalujte, kde jsou ty jedničky a nuly... Už to vidíte? Já to radši zvýrazním:

Obvod zůstává ve stavu, v jakém byl předtím!

Všimněte si, že u obvodů se zpětnou vazbou, tedy takových, kde je na vstup nějakým způsobem zapojený některý z výstupů, musíme už brát v úvahu něco, čemu se říká *předchozí stav*. Zatímco dosud jsme se setkávali s obvody, kde se změna neuvažovala, respektive bralo se, že je okamžitá a vždy dopředná, v obvodech se zpětnou vazbou může mít obvod cosi jako „vnitřní stav". To kombinační obvody, které jsme probíral v minulé kapitole, nemají. Kombinační obvod sám o sobě nemá žádný vlastní stav, a jeho výstup je vždy závislý pouze na stavu vstupů v daném okamžiku.

Zkuste si chování takového obvodu prozkoumat. Co se stane, když přivedete 1 na Set? Co když budou na obou vstupech nuly? Co když budou na obou vstupech jedničky?

Reset	**Set**	**Q**	**/Q**
0	0	Qn-1	/Qn-1
0	1	1	0
1	0	0	1
1	1	0(X)	0(X)

Symbolem „Qn-1“ se označuje „stav předtím„. Pokud je reset i set roven 0, udržuje obvod předchozí stav. Vstup Set nastaví jedničku na výstupu Q, vstup Reset nastaví na výstupu 0. Výstup /Q je „negovaný Q“. Co se stane, když aktivujete oba vstupy, Set i Reset? Obvod se dostane do takzvaného „zakázaného stavu“, též hazardního. Proč hazardní? Představte si, že ze stavu „Set = Reset = 1“ přejdou oba vstupy naráz do stavu 0. Co se stane s výstupy? Správně, začnou kmitat. V reálu ale kmitat nezačnou, v reálu se vzhledem k různým nedokonalostem a asymetriím překlopí jeden výstup do 1 a druhý do 0. Který? No, to je právě to, co nelze předvídat, proto „hazardní stav“. Konstruktéři se proto snaží důmyslnými zapojeními předejít tomu, aby se na vstupech takového klopného obvodu objevily zakázané hodnoty.

18.6.1 Klopný obvod s hradly NAND

Co se stane, když hradla NOR nahradíte hradly NAND? Zkuste si nahradit obvod 7402 obvodem 7400. A pozor, musíte jej pozapojovat jinak, v obvodu 00 jsou hradla „otočená“!

Jak označíte vstupy u takového obvodu?

Po troše zkoumání přijdete na to, že takový obvod funguje, jako by měl invertované vstupní signály – klidový stav je 1, aktivní 0, takže by bylo fér vstupy označit jako /R a /S

18.7 Zakázané kombinace, zpětná vazba, ...

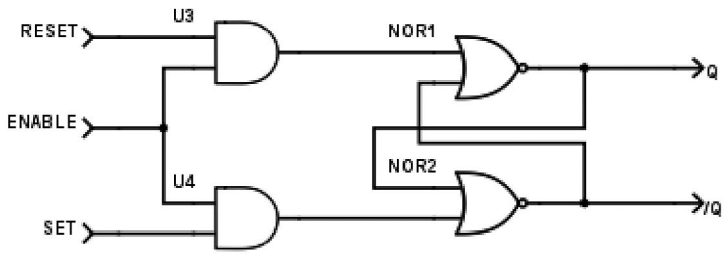
Pojďme si říct na rovinu, že zpětná vazba je dobrý sluha, ale zlý pán. Může snadno způsobit to, že se celý obvod rozkmitá, začne fungovat úplně podivně, popřípadě přestane fungovat, a výsledek nebude předvídatelný. Zpětná vazba totiž do krásně deterministického světa kombinačních obvodů, kde je stav výstupů závislý jen a pouze na stavu vstupů, zavádí prvek zpětného působení výstupů na vstupy, a pokud vytvoříme něco jako „smyčku“, vznikne nám obvod, jehož stav na výstupech není závislý jen na samotných vstupech, ale i na předchozím stavu obvodu.

Ovšem tahle vlastnost není vždy jen negativní.

Zpětná vazba je v elektronice, v číslicové technice i ve spoustě dalších oborů velmi užitečná. Na jednu stranu způsobí třeba to, že mikrofon strčený před zesilovač vygeneruje hnusné hlasité pískání, na druhou stranu když je vhodně zkrocená, tak může generovat pravidelné (i nepravidelné) impulsy a můžeme s ní navrhnout prvek, který si něco pamatuje. *(Ano, tušíte správně, základ všech polovodičových pamětí je zde!)*

Na to, abychom takové obvody drželi zkrátka a aby fungovaly tak, jak si představujeme, je potřeba dodržet určitá pravidla sebekázně. Minule jsme si ukázali, že u klopného obvodu R-S (Reset / Set) existuje takzvaný „hazadrní stav", totiž takový, v němž jsou oba vstupy aktivované a oba výstupy mají stejnou hodnotu. Pokud z takového stavu přepneme naráz oba vstupy do neaktivního stavu, nelze říct, v jakém stavu se bude obvod nacházet.

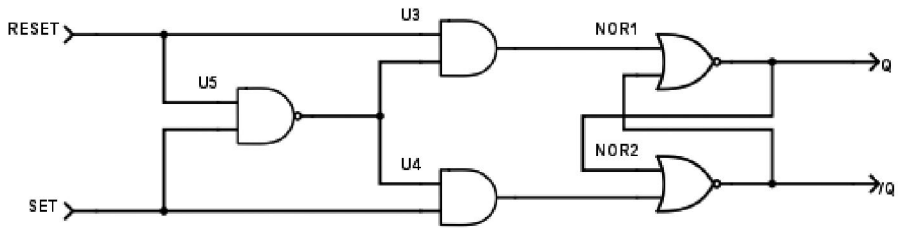
Nejprve si rozšíříme náš klopný obvod R-S o takzvaný **povolovací vstup** E (Enable). Uděláme to jednoduše – na vstupy R a S připojíme hradla, která propustí řídicí signály R a S pouze v případě, že vstupní signál E bude log. 1:



Vidíte sami – do vlastního klopného obvodu se nedostane nic, pokud je E rovno 0. Pokud nastavíme E na 1, funguje obvod jako vždycky.

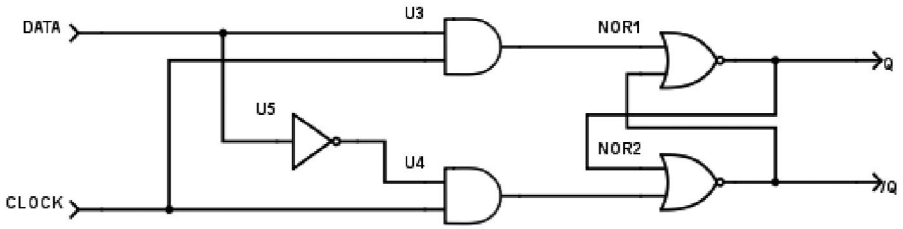
TIP: Tento postup si zapamatujte. Vždy, když potřebujete nějaký vstup, aktivní v logické 1, ošetřit tak, aby fungoval „jen někdy", použijte předřazené hradlo AND, a do něj zaveďte daný signál a řídicí (povolovací) signál E.

Máme teď hradlo R-S s povolovacím vstupem, ale to nijak neřeší hazardní stav (R = S = 1). Zkuste přemýšlet – jak zařídit, aby nikdy nenastal stav, že R i S budou zároveň v logické 1? Tak samozřejmě, můžeme zapojit mezi oba vstupy hradlo AND, jehož výstup bude zároveň ovládat vstup E tak, že pokud nastane R = S = 1, tak vstup E zavře... Nějak takto:



Což je řešení, které téměř nemá chybu. Ve skutečnosti má chybu zásadní, a to tu, že řídicí signál vzniká až v hradle NAND. A hradlo má, jak už víme, nějaké zpoždění, takže hazardní stav R = S = 1 projde přes obě hradla o chviličku dřív, než je stihne nula na výstupu E „zabouchnout".

Použijeme tedy jiný trik: zrušíme dva vstupy R a S, a budeme používat jen jeden. Ten připojíme na oba vstupy – na vstup S přímo, na vstup R přes invertor. Takový vstup nazveme D (jako že „data"). Pokud bude D = 0, bude S taky rovno 0 a R rovno 1. Pokud bude D = 1, bude S = 1 a R = 0. A takto vygenerované signály pošleme do obvodu spolu s povolovacím vstupem E, jak jsme si ho ukázali výše. Výsledkem je obvod, který se označuje jako **latch** – česky se překládá jako „závora", ale tenhle pojem se moc neujal, a i v češtině se říká latch, ale vyslovuje se to [leč].



18.8 Hodiny

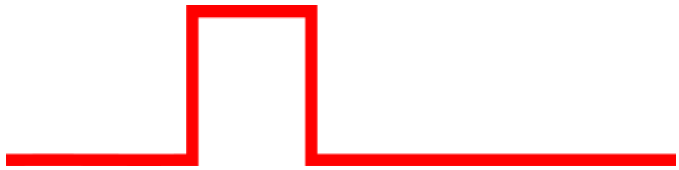
Když píšu „hodiny", nepředstavujte si prosím nic sofistikovaného, natož švýcarského natahovacího se strojkem. Je to prostě jen jeden vodič, na kterém se plus mínus pravidelně střídají 0 a 1. Tento signál podobně jako srdeční tep řídí pak celý obvod. V zásadě má funkci takovou, že říká, kdy nějaký obvod smí změnit svůj stav. A protože u většiny zapojení bývá opravdu pravidelný a přesný a určuje rytmus vnitřních dějů, říká se mu „hodinový puls„.

Výchozí stav je takový, že vstup E je 0. Výstupy zůstávají stále ve stejné úrovni. Jakmile je E = 1, nastaví se výstupy podle vstupu D. Pokud je nula, bude Q = 0 a /Q = 1, pokud je jedna, bude Q = 1 a /Q = 0. Dokud bude E = 1, budou výstupy reagovat na vstup D. Jakmile přepneme vstup

E zpátky na 0, zůstane obvod v posledním nastaveném stavu. Proto ten český název „závora“ – jakmile *spadne*, zůstane výsledek *zamknutý* až do doby, než se opět zvedne.

Modifikací tohoto obvodu dostaneme zapojení, které se přepíná pouze v jednom jediném okamžiku, totiž když se mění stav hodinového vstupu. Nejčastěji reaguje na změnu z 0 do 1 – tomu okamžiku se říká *náběžná hrana hodinového pulsu*.

Výsledek se nazývá **„klopný obvod D“**. Můžeme si to představit jako paměťovou buňku pro jeden bit informace. Přivedeme tento bit na vstup D (buď 1, nebo 0), a „zapamatujeme“ si jej tak, že na vstup Clock přivedeme krátký impuls. Impulsem je míněn přechod z 0 do 1 a opět zpátky na 0. Nějak takto:



Zapamatovaná hodnota je na výstupu Q, na výstupu /Q (NOT Q) je jeho negovaná hodnota.

18.9 Synchronní / Asynchronní

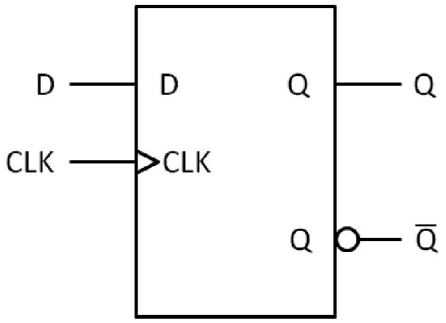
U takového zapojení hovoříme o **synchronním** vstupu D. Slovo „synchronní“ znamená, že se jeho změny projeví v obvodu až poté, co nastane nějaká událost – zde příchod hodinového pulsu. Pouze pokud přijde hodinový puls, může se vstup nějak projevit. Naproti tomu vstupy R, S u dříve zmíněného klopného obvodu R-S jsou vstupy **asynchronní** – změna na těchto vstupech se okamžitě projeví na výstupech.

Jak jsem psal už na začátku kapitoly: pokud je ve hře zpětná vazba, je potřeba ji zkrotit a dát ji zcela jasné mantinely. Takovým mantinelem jsou v číslicových systémech právě hodinové pulsy. Hodinové pulsy *synchronizují* celý obvod a zajišťují, že se něco stane až po něčem jiném, že změny nastanou ve stejný okamžik, že jedna část obvodu se nezmění jindy než jiná, a pokud ano, tak že to nebude mít vliv na výstup, a tak dále...

Proto u složitějších obvodů vždy všechno řídíme hodinovými pulsy. Nechceme, aby se cokoli měnilo halabala, protože pak můžou nastat hazardní stavy. Vždy všechno důsledně řízené hodinami a synchronní. Asynchronní vstup znamená vždy výjimečnou událost, a jako takový by měl být používaný. RESET či přerušení je asynchronní událost, vše ostatní by mělo být synchronní. Jakmile ztratíme synchronizaci, rozsype se funkce celého obvodu, a to znamená třeba ztrátu dat při přenosu, vznik falešných dat a podobné chyby funkčnosti.

18.10 Symbol pro klopný obvod

Už jste si mohli všimnout, že jakmile poskládáte složitější obvod, tak někdo (no dobře, jsem to já) přiběhne a udělá z něj „black box„, obdélník, kde popíše, jaké jsou vstupy (vlevo) a jaké jsou výstupy, a je to.



Stalo se to i s vaším klopným obvodem D. Všimněte si, že vstupy D a CLK jsou vlevo, a u CLK je malá šipka. Ta označuje, že vstup má funkci hodinového vstupu, synchronizace. Dokonce podle směru šipky (do obvodu / z obvodu) můžeme poznat, jestli jde o hodinový vstup spouštěný vzestupnou hranou nebo sestupnou hranou. Kroužek u vstupu nebo výstupu značí negovaný signál, tedy takový který je aktivní v logické 0.

Po pravdě: v technických normách naleznete spoustu speciálních symbolů, které dokáží označit, že daný vstup je například asynchronní a má Schmittův obvod. V praxi, a zejména v amatérských schématech, se setkáte s tak zjednodušenými značkami, že jste vůbec rádi, když autor vyznačí kolečkem invertovaný vývod, a šipka u hodin bývá vrchol snahy. Sice to vzbuzuje úšklebek u odborníků, kteří jsou zvyklí na jednoznačnost a u zkoušek na škole vás za špatně nakreslený symbol vyhodí, ale – život je takový, a vy máte dvě možnosti: buď budete nadávat na toho, kdo to kreslil, nebo přijmete fakt, že nic a nikdo není dokonalé. Doporučuju druhý přístup.

18.11 Reálný klopný obvod D: 7474

Samozřejmě že máme k dispozici popsaný klopný obvod typu D jako součástku. A ano, dodneška se používá. Nejznámější obvod je 7474 (a my už víme, že to může být i 74LS74, nebo 74HCT74). Tento obvod obsahuje dvojici klopných obvodů typu D. Jeho datasheet naleznete třeba zde: http://www.ti.com/lit/ds/symlink/sn74s74.pdf. V popisu je napsáno, že jde o „Dual D-type Positive-Edge-Triggered Flip-Flops with Preset and Clear“. Pojďme si to v rámci cvičení přeložit.

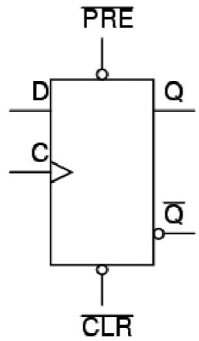
- „Dual“ naznačuje, že v jednom pouzdru jsou dva klopné obvody stejného typu. Česky „dvojitý“.

- „D-type Flip-Flop“ říká, že jde o klopný obvod (flip-flop – vážně, takhle se v anglické literatuře říká klopným obvodům) typu D.

- „Positive edge triggered“ si rozluštíme jako „spouštěný náběžnou hranou“. Trigger je spoušť, tady to symbolizuje onu akci, která se provede – třeba „zapamatování údajů“. Positive edge je takzvaná „náběžná hrana“. Když se podíváte na vizualizaci hodinového pulsu, tak ta svislá čára vlevo, jak stoupá zespoda nahoru, tomu se říká „náběžná“ (též „vzestupná“) hrana hodinového pulsu. Ta druhá, přechod shora dolů, je sestupná hrana, anglicky pak „negative edge“. Někdy se setkáte i s termíny „falling edge“ (sestupná) a „lead edge“ (náběžná).

- „with preset and clear“ říká, že jde o klopný obvod D, který má nastavovací a nulovací vstup. Někdy se dodává, aby to bylo jasnější, i slovo „asynchronous preset and clear“. Jde v podstatě o naše vstupy R a S, jen s tím rozdílem, že jsou připojeny přímo do klopného obvodu a nečekají až na hodinový impuls.

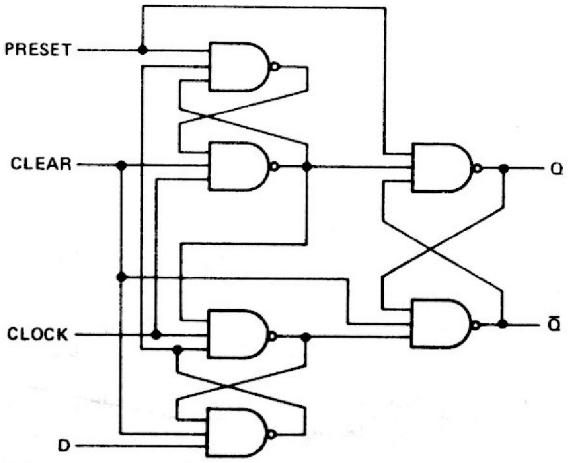
Schematická značka to celé hezky ilustruje:



Vidíte dva vstupy D a C (ten má funkci hodin), vstupy PRE a CLR (negované) a dva výstupy, Q a /Q.

*Všimněte si: u podobných obvodů se **nekreslí napájení**. Samozřejmě, že obvod musí být připojený na napájecí napětí, ale to se **bere samo sebou, automaticky**, takže ho není nutné kreslit.*

Schéma vnitřního zapojení je ve skutečnosti o něco složitější než je výše uvedený „latch“ – to proto, aby obvod reagoval opravdu jen na přepnutí z 0 do 1, a ne na jiné události:



'L74—DUAL D WITH CLEAR AND PRESET

Naštěstí se zapojením nemusíte zabývat. Vám stačí znát funkční tabulku:

Vstupy				**Výstupy**	
/PRE	**/CLR**	**CLK**	**D**	**Q**	**/Q**
0	1	X	X	1	0
1	0	X	X	0	1
0	0	X	X	1*	1*
1	1	^	1	1	0
1	1	^	0	0	1
1	1	0	X	Q0	/Q0

Nerozumíte? Vysvětlení je zde:

Vstupy /PRE a /CLR jsou zmiňované „preset“ (=„nastav na 1“) a „nuluj“ (=„nastav na 0“). Lomítko před názvem znamená, jak už jsme si řekli, negaci, takže klidový stav je 1, a požadovanou akci spustíme hodnotou 0.

Zatím jsme se setkávali s pozitivními signály, tedy takovými, kdy 0 znamenala „klid“ a 1 nějakou akci. U negovaných vstupů to je obráceně. Některé signály se v číslicové technice používají právě v takovéto negované podobě. Má to své důvody, o nichž jsme se už bavili, tak jen připomenu: logická 1 má širší rozsah napětí, tak je odolnější proti rušení.

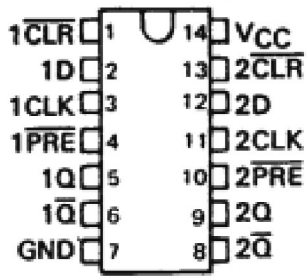
První řádek ukazuje stav, kdy je aktivní signál PRESET (na vstup /PRE je přivedena logická 0, na vstupu /CLR je logická 1, tedy „klid“). Vstupy CLK a D mohou být v libovolné úrovni, což značíme symbolem X. V takovém případě se obvod „nastaví“, na výstupu bude 1, na negovaném výstupu bude tedy 0.

Druhý řádek. Je aktivní signál CLEAR (tedy /CLR je 0, /PRE je 1, CLK a D libovolné). Situace je analogická předchozí, ale obráceně: obvod se „vynuluje“, na výstupu bude 0, na negovaném výstupu 1.

Třetí řádek označuje hazardní stav. Oba asynchronní vstupy PRE i CLR jsou aktivní (tedy v úrovni 0), oba výstupy jsou v logické 1, a v datasheetu je ještě u toho poznámka, že nelze zaručit správnou úroveň napětí na výstupu, takže do takového stavu by se neměl obvod dostávat.

Poslední tři řádky popisují situaci, kdy jsou asynchronní vstupy v klidu (=log. 1). V takové situaci obvod funguje tak, jak jsme si popsali výše. Pokud je CLK rovno 0, tak bez ohledu na stav vstupu D je na výstupech to, co tam bylo předtím (poslední řádek tabulky). Změna probíhá s náběžnou hranou signálu CLK (symbolizováno stříškou ^) – pokud je D = 1, bude i Q = 1 (a /Q logicky 0), pokud je D = 0, bude Q = 0 (a /Q = 1).

Reálná součástka je v pouzdře DIL se 14 vývody, které jsou zapojeny takto:

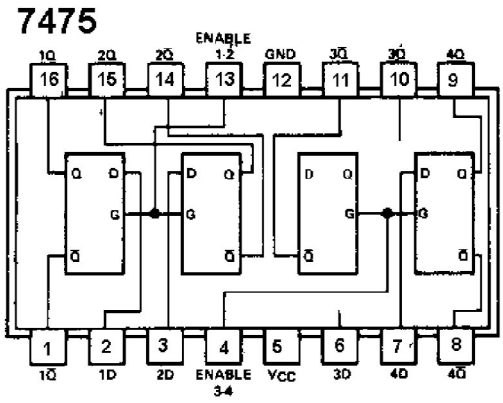


A protože jsou uvnitř dva nezávislé klopné obvody, jsou i všechny vstupy a výstupy v obvodu dvakrát. Při pohledu shora vidíte vlevo vývody obvodu 1 postupně pod sebou: /CLR, D, CLK, /PRE, Q a /Q. Vpravo je totéž pro obvod 2.

Možná vás mate, že hodiny se tu značí CLK, zatímco já vám výš tvrdil, že se značí C. Víte, ve skutečnosti je to jedno – CLK nebo C, obojí je přijatelné, když je jasné, o čem je řeč. Někdy se setkáte i s označením CK a někdy ještě jinak, ale vždy je někde napsáno, že se jedná o hodinový vstup (clock).

18.12 Reálný latch 7475

Myslím, že vás nepřekvapí, když vám prozradím, že i latch existuje v podobě integrovaného obvodu. Ten nejznámější má označení 7475 a obsahuje dvě dvojice latchů, vždy se společným vývodem E pro každou dvojici.



Tento obvod má o dva vývody víc, než ty, s nimiž jsme se až doteď setkávali (má pouzdro DIP16, předchozí obvody měly DIP14), a taky nedodržuje pravidlo pro napájení. Ostatní obvody, s nimiž jsme se až dosud potkávali, měly napájení „vpravo dole zem“ a „vlevo nahoře napájení“ – tento obvod to má jinak! Zem je na vývodu 12, napájení na vývodu 5, tedy zhruba uprostřed pouzdra.

Proč to výrobce takto udělal? Snad proto, aby nás zmátl? Ne, tak to není. Ve skutečnosti je ten důvod ukrytý ve vnitřní struktuře čipu, kdy se při návrhu ukázalo, že je mnohem snazší navrhnout tento čip tak, že bude mít napájení nikoli v rozích, ale po stranách…

Jsou situace, kdy bude použití 7475 vhodnější pro daný účel. Ovšem obecně platí, že klopné obvody, které jsou řízené úrovní (*latche*), mohou do systému při nevhodném použití zanést nejednoznačnosti a hazardní stavy, a proto je bezpečnější používat klopné obvody řízené hranou.

19 Panna, nebo orel?

19 Panna, nebo orel?

To takhle sedíte u počítače a přemýšlíte: Mám napsat teď kapitolu o posuvných registrech, nebo raději nějakou praktickou konstrukci? Jak se rozhodnout? Co třeba hodit mincí?

Problém je, pokud jste moderní člověk s bezhotovostními penězi. To si můžete tak akorát hodit platební kartou, a to, uznejte sami, není ono. Můžete si taky pustit počítač a naprogramovat si házení mincí… Anebo si můžete postavit elektronickou házecí minci.

Jako správní nadšenci do číslicové techniky se pustíme do té poslední varianty. Racionální hlas uvnitř hlavy ironicky podotýká, že je to varianta sice nejdražší, zato zabere nejvíc času. Nadšenecký hlas ale říká: *No a co? Aspoň se u toho něco naučím!*

Zapojení bude jednoduché. Budou tam dvě LED, a bude svítit vždy jen jedna z nich. Pokaždé, když obsluha zmáčkne tlačítko, tak se jedna z nich rozsvítí. Která? No, to by mělo být náhodné!

Zařízení samozřejmě nemusíme používat nezbytně nutně jen na házení mincí. Hodí se třeba i pro chovatele Schrödingerovy kočky, kde poměrně kvalitně simuluje otvírání krabice, a pro spoustu dalších zásadních životních rozhodnutí. Sice nedokáže simulovat stav „mince padla na hranu“, „mince zapadla mezi prkna v podlaze“ a „mince zůstala viset ve vzduchu“, ale na hrubou náhradu to jistě stačí.



Tak. Princip máme. Jak to teď celé poskládat? Vyjdeme od konce, tedy od těch zobrazovacích LE-Dek. Je potřeba, aby zůstávaly v jednom ze dvou stavů. Tady by se hodilo mít třeba něco jako – jo, něco jako klopný obvod D, který má dva výstupy, Q a /Q, které jsou vždy takříkajíc „v opozici“. No a k nim připojíme ty dvě LED, jednu z nich označíme „panna“ a druhou „orel“ (ti, co si staví Schrödingerovu krabici, si LED označí „živá číča“ a „mrtvá číča“).

Zobrazování bychom měli taky. Teď co s tím tlačítkem?

Vzpomeňte si na kapitolu, kde jsme si představovali obvod 7474 – dvojici klopných obvodů typu D. Mají dva synchronní vstupy, totiž D a CLK. A vstup CLK funguje tak, že propustí do obvodu stav na vstupu D, a to **ve chvíli, kdy se na tomto vstupu změní úroveň z log. 0 do log. 1**, což je přesně to, co potřebujeme. Přesně tohle totiž udělá naše tlačítko, pokud ho zapojíme na + 5 V – jakmile ho obsluha stiskne, změní se stav z 0 V na + 5 V. To je ta „vzestupná hrana", která spustí požadovanou akci.

Teď ještě tu náhodu. Jak to zařídit?

19. 1 Náhoda? Nemyslím si...

Můžeme použít tu nejlepší náhodu, co známe, a měřit rozpad radionuklidů. Což je samozřejmě řešení první volby, pokud máte doma radionuklidy a detektory částic. Ti, kteří nemají, pokud se snad takoví najdou, musí řešit problém jinak.

Další možnost je vzorkování šumu. Ale řekněme si upřímně a na rovinu: Zas tak matematicky dokonale náhodné to být nemusí. My totiž můžeme využít toho, že *naprosto náhodný* je okamžik, kdy člověk to tlačítko stiskne. To nelze předem odhadnout. Takže stačí na vstup D pustit pravidelně se měnící signál 0 – 1 – 0 – 1, takový, jaký máme například z našeho blikače, a je to.

No jo, zní ale námitka – co když obsluha zná frekvenci a naučí se mačkat pravidelně v témže rytmu? To pak budou padat samé hlavy, nebo samí orli?

Co s tím? No, zase využijeme toho, že člověk není stroj, a zvýšíme frekvenci kmitů. Když jich bude třeba tisíc za sekundu, tak mačkat tlačítko v tak přesných intervalech, aby to vždycky vyšlo ve stejné chvíli, zvládne jen Chuck Norris. Pro nás ostatní to bude stačit.

Použijeme tedy zapojení blikače, které lehce upravíme tak, aby blikalo rychleji. Máme tři možnosti:

1. Zmenšit kapacitu kondenzátoru.

2. Zmenšit odpor rezistoru.

3. Udělat obojí.

V kapitole o astabilních klopných obvodech jsem ukazoval vzorec f = 1 / (2,2RC). Pokud tedy chci blikání okolo 1 kHz, musí platit, že 1 / (2,2RC) = 1000, tedy 2,2RC = 0,001, tedy RC = 0,00045.

19.1.1 Jaké hodnoty RC zvolit, když vím jen to, že součin má být XYZ?

To je dobrá otázka. Něco podobného vyprávěl Feynman ve své knize vzpomínek, když líčil, jak kombinoval ozubená kola, aby dostal určitý poměr. Starší zkušený technik mu dal radu: „Vezmi katalog ozubených kol, vynech kola s příliš velkým počtem zubů a s příliš malým počtem zubů, a z těch středních hodnot to nějak nakombinuj!"

Dělejte to úplně stejně. Vyhněte se hodnotám příliš nízkým i příliš vysokým, a pokud to jde, použijte „zlatý střed". A vždy začněte tím, co máte doma. Já bych sáhnul po kondenzátoru 100 nF, protože těch mám doma dost, a hlavně proto, že mám výrazně méně hodnot kondenzátorů (vlastně jen 100n, 33p a pak elektrolyty) než rezistorů (těch mám asi 30 základních hodnot).

Jak to vychází pro 100 nF?

R = 0,00045 / 0,0000001 = 4500. Tedy odpor 4k5. Ten nemám, sáhnu tedy po odporu 4k7. Bude to nějak moc vadit?

f = 1 / (2,2 × 4700 × 0,0000001) = 967 Hz (zhruba).

Což je hodnota přijatelná, v toleranci – ono je jedno, jestli to bude 1 kHz, nebo 0,967 kHz, důležité je, že to je „dost rychle na to, aby člověk nedokázal vychytat frekvenci".

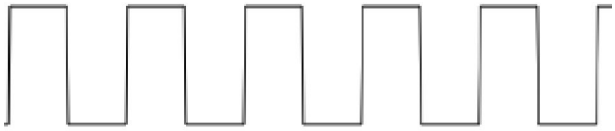
Kdybych neměl 4k7, dám tam 3k3, výsledná frekvence bude f = 1 / (2,2 × 3300 × 0,0000001) = 1377 Hz, tedy necelých 1,4 kHz.

Dobrá. Tohle všechno je OK, ale co kdybych chtěl tu frekvenci fakt přesnější? Měl bych hledat přesné rezistory a kondenzátory? Mohl bych, ale byla by to marná práce, protože oscilátor, založený na kondenzátoru a rezistoru, se bude velmi pravděpodobně rozlaďovat. Jistější by bylo použít obvod 555, který má díky teplotním kompenzacím přesnější chod, a úplně nejpřesnější by bylo vzít krystalový oscilátor. S největší pravděpodobností byste frekvenci 1,000 kHz nesehnali, tak by nezbývalo, než použít frekvenci vyšší a pak ji podělit.

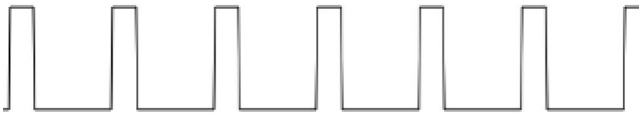
⌀ https://eknh.cz/pnor

19.2 Střída

Do tohoto okamžiku jsem tak nějak mlčky předpokládal, že když má oscilátor frekvenci 1 Hz, tak že to znamená nejen to, že každý kmit (tj. logická 0 a logická 1) trvá jednu sekundu, ale i to, že jsou obě hodnoty stejně dlouhé, totiž že je půl sekundy logická 0 a půl sekundy logická 1. Nějak takhle:



Jenže ona to nemusí být pravda. Co když to bude vypadat třeba takhle? Čtvrtinu sekundy logická 1, tři čtvrtiny 0. Stále platí, že frekvence je 1 Hz – náběžné hrany pulsů přicházejí po sekundě jako předtím, ale výsledek vypadá jinak:



A co teprve takovýhle signál?



Taky náběžná hrana každou sekundu, ale signál je v logické jedničce jen mizivý zlomek sekundy. Kdybyste svůj generátor „panna-orel" krmili takovýmto signálem, co by tak asi vyšlo? Správně, padala by pořád jen jedna hodnota.

Faktoru, který popisuje, jestli má pravidelný signál poměr trvání jedniček a nul stejný, nebo různý (a jak moc různý) se říká střída (anglicky duty cycle, někdy se to překládá i jako „plnění"). Když je signál tak hezky rovnoměrný jako na prvním obrázku, říkáme, že má střídu 1:1, nebo plnění 50 %. Ten pod ním bude mít 25 % (nebo 1:3), ten na posledním obrázku třeba 1 %.

Za chvíli si řekneme, jak zajistit střídu 1:1, ale nejdřív se pojďte podívat, jak to vypadá a co se děje, když se mění střída.

Sice bychom mohli experimentovat s obvodem 555 a sluchátkem a poslouchat, jak se zvuk mění z takového nijakého bučení (1:1) na ostrý „techno" zvuk u střídy okolo 1:4 a pak na jemné bzučení. Jenže to už bychom hodně odbočili od číslicové techniky.

Můžete si to přesto zkusit: ⌀ https://eknh.cz/pwma *– jsou tu čtyři generátory se stejnou frekvencí, ale různou střídou. Nechte emulátor chvilku běžet, aby měly audiovýstupy dostatek dat k přehrávání (naplní se ukazatel u „audio out"), a pak si pomocí tlačítek Play vpravo přehrajte jednotlivé zvuky.*

Teď použijeme světlo a Arduino. Arduino Uno má LED na pinu 13, už jste si s ní blikali, a teď se k blikání vrátíme. Ve smyčce loop() je dvakrát volána funkce delay(1000). Co když to změníme tak, že u prvního bude delay (100), u druhého delay (1900)? Celkové zpoždění zůstane stejné, 2000 milisekund, ale změní se právě střída. LED bude buď „probleskovat", nebo „pohasínat".

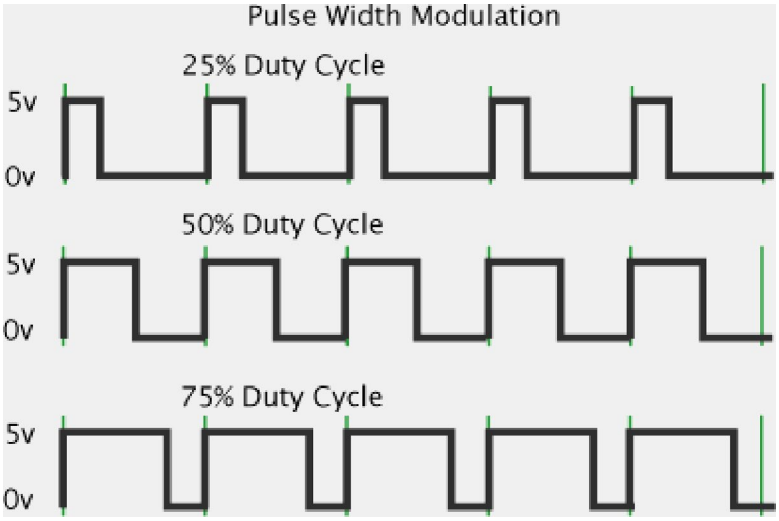
Fajn, a teď zvýšíme frekvenci – tedy snížíme hodnoty u delay. Obě na desetinu. Třeba na 10 a 190. Dioda probleskuje velmi rychle, a to s frekvencí 5 Hz. Jdeme dál, ještě o desetinu. Budeme mít 1 a 19. Dioda teď bliká tak rychle, že už oko nevidí jednotlivé záblesky, ale zdá se, jako by slabě svítila pořád. Zkusme teď změnit střídu na 50 % – tedy 10 a 10. LED bude blikat se stejnou frekvencí, ale bude mnohem více času trávit ve stavu „svítí". Oku se tedy bude zdát, že svítí s větším jasem.

Když poměry dob změníte třeba na 19:1, bude LED zase blikat, ale bude 19 milisekund ve stavu „svítí" a 1 milisekundu ve stavu „nesvítí". Výsledek bude vypadat jako ještě větší jas!

Právě jste si vyzkoušeli, jakým způsobem měnit jas LED, i když máte k dispozici jen hodnotu 0 nebo 1!

19.3 PWM

Je to sice velký objev pro člověka, ale malý krok pro lidstvo. Lidstvo totiž tuhle techniku dávno zná, a nazývá ji PWM, což je z anglického Pulse Width Modulation, neboli **Pulsně-šířková modulace**. Jako že modulujeme (měníme) šířku pulsu. (Zmíněné Arduino má tuhle vlastnost zabudovanou, má na to funkci analogWrite.)



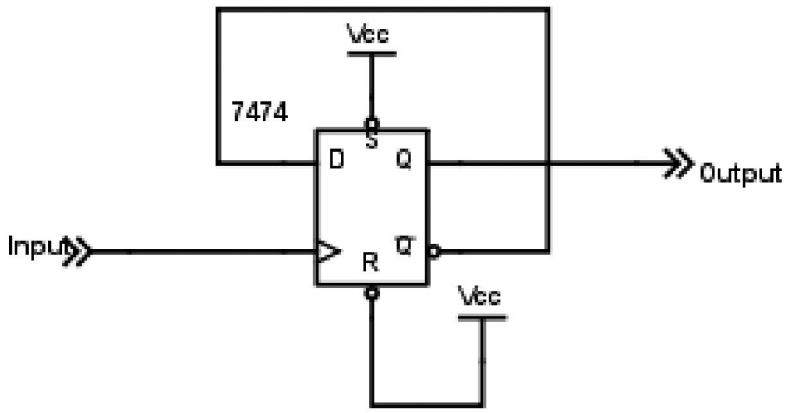
PWM umožňuje pomocí změny střídy dávkovat množství energie, které do zařízení přiteče. Pomocí PWM se dá dobře měnit jas žárovky či LED (ano, stmívání světel funguje přesně na tomto principu), pomocí PWM se mění výkon topné spirály v elektrickém sporáku i výkon ohřevu v mikrovlnce (všimněte si, že při různých výkonech mikrovlnka střídavě hučí a nehučí – to vždycky na určitou část pracovního cyklu řídicí jednotka zapíná a vypíná mikrovlnnou lampu.) Jde v podstatě o nejjednodušší způsob, jak z digitálního světa udělat „alespoň trochu" analogový výstup. Jsou samozřejmě i lepší způsoby, ale k nim se dostaneme později.

19.4 Dělení kmitočtů

Už jsem tu zmínil, že existuje jednoduchá technika, díky níž můžeme jakýkoli pravidelný signál s libovolnou střídou změnit na signál se střídou 1:1. Jak na to?

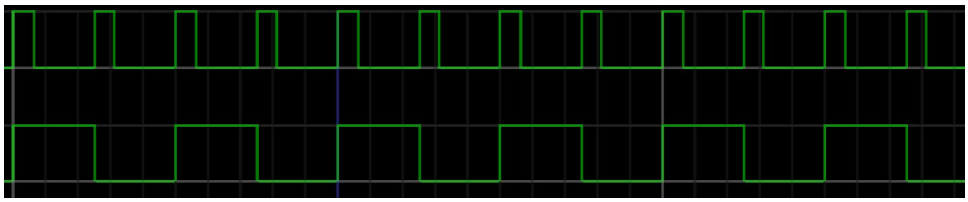
Tak, jedna možnost je měřit, za jak dlouho po sobě přijdou dvě náběžné hrany, a pak na výstup pouštět signál se sestupnou hranou posunutou přesně do poloviny tohoto intervalu. Což je teoreticky správné řešení, ale takhle se to zkrátka nedělá, protože to je řešení náročné. Existuje totiž jednodušší cesta.

Opět si vzpomeňte na náš klopný obvod typu D a na jeho vlastnost, popsanou slovy „změna stavu proběhne při změně hodinového vstupu z 0 na 1" (tedy při *náběžné hraně hodinového pulsu*). A teď si představte, že na vstup D přivedeme výstup /Q – tedy negovaný. Vlastně vytvoříme zpětnou vazbu na druhou. Na začátku je třeba obvod ve stavu 0. Na výstupu Q bude 0, na /Q 1. Ta je zároveň připojená na vstup D. Při náběžné hraně vstupu CLK se obvod překlopí do log. 1, na výstupu



Q bude 1, na /Q bude 0… Při další náběžné hraně se obvod opět překlopí do 0…

Vidíte, že je úplně jedno, jestli je střída vstupního signálu 1:1 nebo 1:3 – pokud přicházejí náběžné hrany pravidelně, překlápí se klopný obvod **s poloviční frekvencí proti vstupu**. Pokud máte na vstupu 10 kHz, na výstupu bude 5 kHz, a vedlejší efekt bude ten, že dosáhnete přesné střídy 1:1. A proto se často používá tento postup při získávání časových pulsů v číslicové technice. Některá zapojení jsou totiž náročná na dodržení přesné střídy, a proto se používají oscilátory, které kmitají na vyšší frekvenci, která je pak dělena na požadovanou hodnotu.



A jako vedlejší efekt jsme si právě ukázali jedno ze zásadních zapojení, totiž **dělič kmitočtu**.

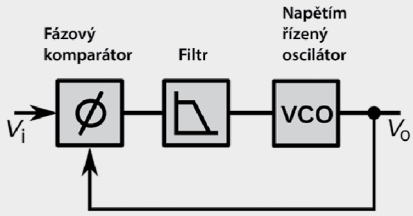
⌀ https://eknh.cz/del2

Můžete druhou půlku obvodu 7474 zapojit jako dělič kmitočtu u vašeho hodomincomatu. Sice se frekvence přepínání sníží na polovinu, zato budou oba stavy trvat stejně dlouho, a při stisknutí tlačítka budou pravděpodobnosti obou stavů stejné.

19.4.1 Násobení kmitočtu?

Dělení kmitočtu bychom měli pořešené, a co takhle násobení? Šlo by to nějak? Šlo. Sice to sami dělat nebudeme, ale určitě se s tím časem někdy setkáte, tak si to ve stručnosti popíšeme, ať víte…

*Existuje složitá součástka, která se jmenuje **fázový závěs**. Anglicky se to označuje Phase-Locked Loop a zkracuje se to na **PLL**. Fázový závěs dokáže generovat signál tak, aby měl fázi shodnou se vstupním signálem (zjednodušeně řečeno aby chodily vzestupné i sestupné hrany ve stejný okamžik). Pokud se vstupní (též **referenční**) a výstupní signál „rozladí", změní frekvenci výstupu tak, aby jej opět synchronizoval se vstupem.*



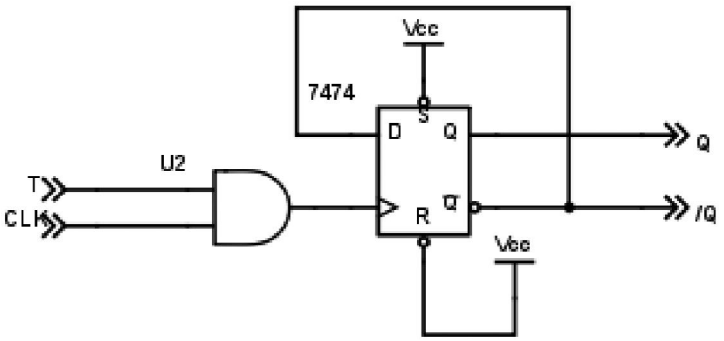
Představte si teď, že do zpětné vazby zařadíme třeba děličku 1:64. Znamená to, že PLL bude generovat kmitočet 64 × vyšší než je referenční.

PLL se používají u některých jednočipů, kde slouží ke generování vysokého pracovního kmitočtu, kontrolovaného relativně pomalým krystalem. U složitých logických polí (FPGA) bývá hned několik PLL, jimiž můžeme ze vstupního kmitočtu, třeba 50 MHz, vygenerovat celou škálu hodinových signálů.

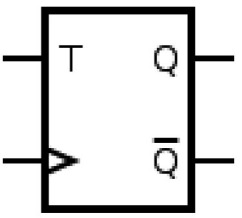
Výhoda takového řešení je, že můžete přepínáním děličů a násobičů změnit pracovní frekvenci obvodu bez změny krystalu (snížení frekvence se používá ke snižování odběru). Takto získaný kmitočet je navíc dostatečně přesný, takže není potřeba krystalů s vysokými frekvencemi, které se obtížně vyrábějí.

19.5 Klopný obvod T

Když u předchozího obvodu do cesty hodinovému pulsu zapojíme hradlo AND, získáme tak klopný obvod typu T (Toggle).



Pravdivostní tabulka takového obvodu je prostá: Pokud je T = 0, zůstává klopný obvod stále ve stejném stavu. Pokud je T = 1, tak se s každou náběžnou hranou hodin přepne.

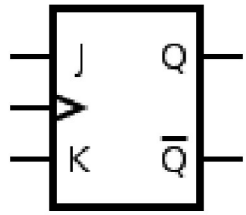


T	Q_n	Q_{n+1}	Funkce při náběžné hraně Clk
0	0	0	Drží hodnotu
0	1	1	Drží hodnotu
1	0	1	Přepíná
1	1	0	Přepíná

Klopný obvod T se používá pro konstrukci čítačů a děliček frekvence, jak si hned za chvíli ukážeme.

19.6 Klopný obvod J-K

Nebojte, tento je poslední, a zmíním se o něm jen pro úplnost. Zas tak často se totiž v praxi nepoužívá… Dřív se z těchto obvodů skládaly například čítače, ale dnes už jsou takové obvody dostupné v integrované podobě, takže se s ryzím obvodem J-K skoro nepotkáte.



Klopný obvod J-K je vlastně kombinací jednoduchého klopného obvodu R-S a klopného obvodu typu T. Pokud jsou J a K v logické 0, drží obvod svůj stav. Pokud je J = 0 a K = 1, nastaví se obvod do logické 0. Pokud je J = 1 a K = 0, nastaví se obvod do logické 1. No a konečně pokud jsou oba vstupy v log. 1, přepne se s každým hodinovým pulsem obvod do opačného stavu, než byl předtím.

I tento klopný obvod najdete v řadě 74xx – nejznámější jsou typy 7470 a 7472. U těchto obvodů jsou vstupy J a K dokonce trojité – /J0, J1, J2 a /K0, K1, K2. Jeden je negovaný, dva obyčejné, a uvnitř se skládají pomocí hradla AND: K = /K0 AND K1 AND K2. K tomu, aby byl vstup K aktivní, musí nastat kombinace K0 = 0, K1 = 1, K2 = 1. Obdobně pro vstup J.

Klopné obvody tvoří základ veškerých složitějších číslicových strojů. Jejich zásadní vlastností je, že si dokáží pamatovat, a tím do systému vnáší prvek času. „Něco" se stane v pravidelných intervalech, „něco" se stane až poté, co se stalo něco jiného…

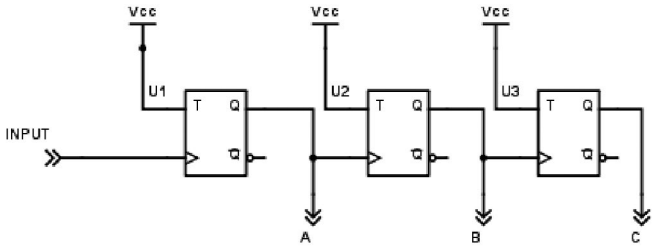
— 19 Panna, nebo orel?

20 Čítače

20 Čítače

Musím se vám k něčemu přiznat. Když jsem jako desetiletý chlapec dostal do ruky katalog integrovaných obvodů Tesla a viděl jsem obvody, které se jmenují „čítače“, tak jsem si říkal: „Asi udělali chybu, asi tam mělo být napsáno *počítače*,“ a fakt jsem se těšil, že *tohle už je to ono*. No, není. Trochu mě to mrzelo, ale jen chvilku, brzo jsem totiž zjistil, že čítače jsou opravdu šikovné součástky.

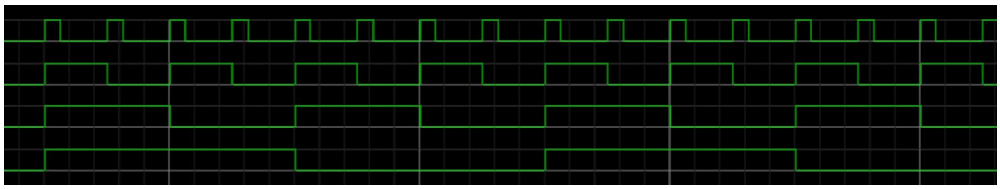
Představte si, že vezmete několik „děliček kmitočtů“ a zapojíte je za sebe. Nějak takto:



⌀ https://eknh.cz/cit3

Vstupy T jsou připojené na Vcc – napájecí napětí, takže jsou ve stavu log. 1 a obvod funguje jako dělička kmitočtu.

Na vstup pak přivedeme hodinový signál. Co se stane?



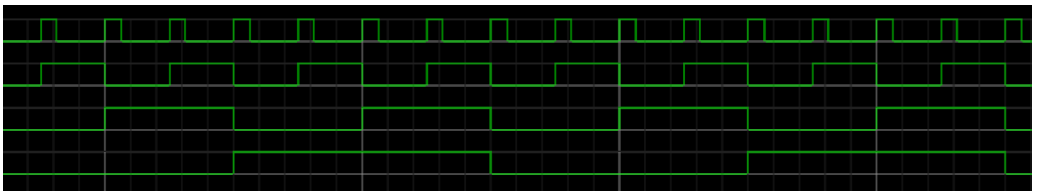
Na výstupu A bude hodinový signál s poloviční kmitočtem, na výstupu B se čtvrtinovým, na výstupu C s osminovým… A když se podíváte na stav na vývodech CBA v čase, uvidíte opakující se vzorec:

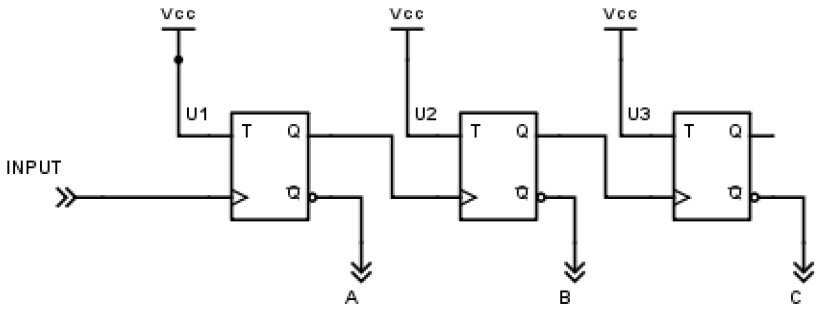
111–110–101–100–011–010–001–000–111

…

Tedy v dvojkovém kódu 7 – 6 – 5 – 4 – 3 – 2 – 1 – 0 – 7 – 6 atd.

Když vezmete data z negovaných vývodů, dostanete vzestupnou sekvenci:





⌀ https://eknh.cz/citn

Právě jste stvořili **dvojkový (binární) tříbitový asynchronní čítač**. Pokud budete jeho tři vývody považovat za tři bity informace, bude se s každým pulsem na vstupu hodnota na výstupu zvyšovat o 1 a bude postupně střídat hodnoty 0 až 7 stále dokola.

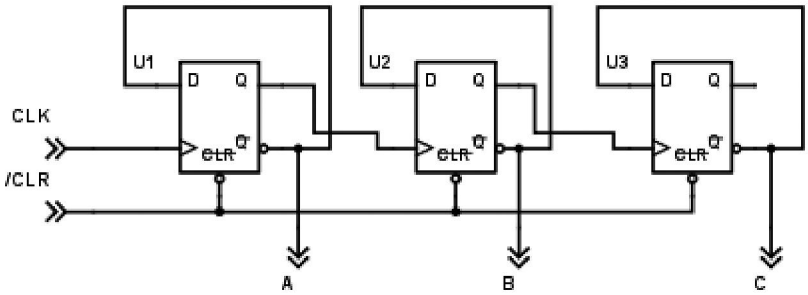
Když přidáte další klopný obvod do řady, získáte čtyřbitový čítač, který bude mít hodnoty 0 – 15. Takhle můžete skládat čítače za sebe, každý další klopný obvod zvýší počet hodnot na dvojnásobek. Patnáctibitový čítač tak bude nabývat hodnot 0 až 32767…

K čemu to může být dobré kromě dělení frekvence? Co třeba k počítání nějakých událostí? Třeba když budete počítat pulsy s frekvencí 1 Hz, získáme hodiny.

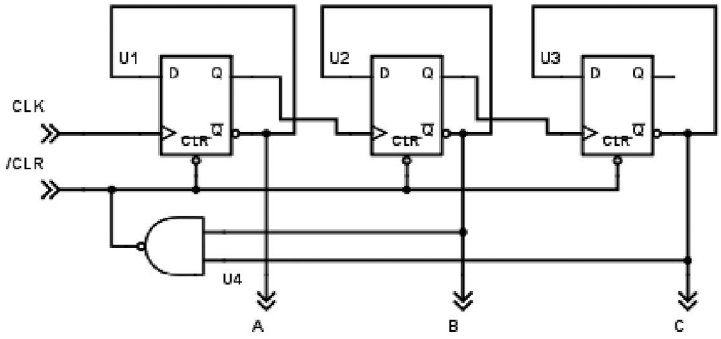
Nojo, říkáte si, ale hodiny se přeci nepočítají do osmi, ale do šedesáti sekund, šedesáti minut a 24 hodin. Co s tím?

20.1 Čítač s nulováním

Pokud u klopných obvodů vyvedete vstup /CLR (tedy nulovací), získáte možnost kdykoli celý čítač „vynulovat“ a počítat opět 0 – 1 – 2 – 3…



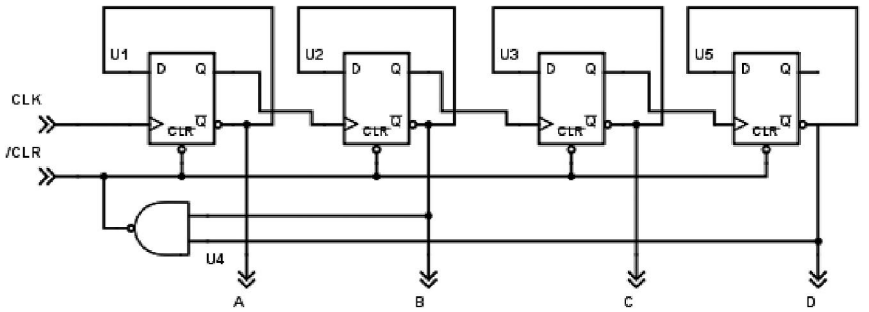
Co kdybychom u našeho čítače zapojili nulovací vstup tak, že se nastaví ve chvíli, kdy vývody B a C budou 1? Prostě přes hradlo AND – respektive NAND, protože vstup /CLR je aktivní v 0:



Co se stane? Čítač bude postupně na vývodech CBA procházet stavy 000, 001, 010, 011, 100, 101, a pak vstoupí do stavu 110. V tu chvíli se na vývodu hradla NAND objeví log. 0 (B NAND C), a tím se uvede do stavu 000. Takovému zapojení se říká **čítač se zkráceným cyklem** – v tomto případě jsme zkrátili cyklus na šest kroků, 000 až 101, tedy desítkově 0 až 5.

Technicky samozřejmě platí, že se na vývodech objeví i stav 110, ale ten je tam jen velmi krátkou dobu, která odpovídá rychlosti hradla NAND a nulovacího vstupu. U běžných zapojení můžete takový stav ignorovat a zanedbat.

Máme tedy čítač, který čítá od 0 do 5. Stejným způsobem můžete zkrátit čtyřbitový čítač tak, aby počítal 0 – 9:

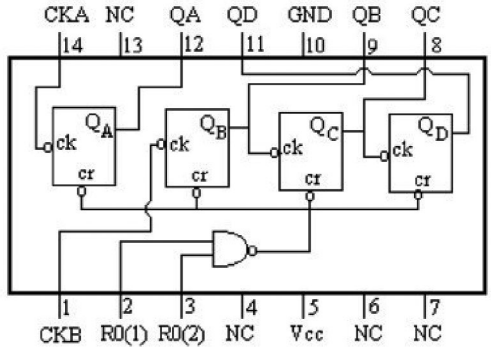


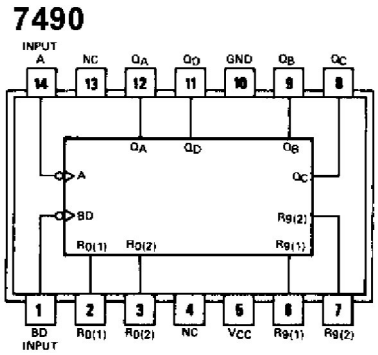
Výborně, máte tím pádem pořešené dělení desíti a dělení šesti. Když zapojíte za sebe děličku 10 a děličku 6, získáte děličku 60… Představte si teď, že na vstup zapojíte ten zmiňovaný hodinový signál s frekvencí 1 Hz. Co získáme? Ano, z prvního děliče budou jednotky sekund (0-9), z druhého desítky (0-5). Po hodnotě 59 bude následovat 00…

⌀ https://eknh.cz/hod

20.2 Čítače v praxi

Čítače se samozřejmě, jak jinak, vyrábí i jako integrované obvody v řadě 74xx. Dva nejběžnější jsou 7490 a 7493. Jeden z nich je čtyřbitový asynchronní binární, druhý čtyřbitový asynchronní desítkový (to jako že počítá 0-9). Který je který? Mnemotechnická pomůcka: 7490 má na konci 0, stejně jako 10 má na konci nulu, takže 7490 je desítkový, 7493 je binární (vlastně „šestnáctkový“).





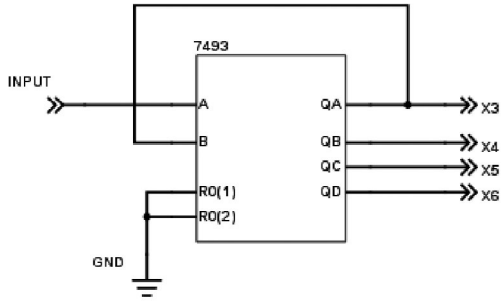
Oba obvody mají stejné zapojení vývodů, jen u typu 7490 je zapojený vnitřní resetovací obvod pro stav 1010 (= 10 v desítkové soustavě), čímž se zkracuje cyklus, a má navíc vstupy R9.

Všimněte si několika věcí: **nulovací vstup není jeden, jsou dva**, a jsou uvnitř zapojeny s hradlem NAND, takže k vynulování je potřeba stav R0(1) = R0(2) = 1. Což se nám hodí, viz výše – až budeme zkracovat cyklus na 6 hodnot, tak nemusíme zapojovat už nic navíc.

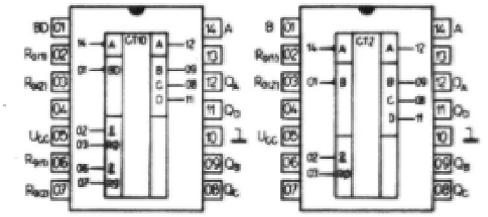
U čítače 7490 jsou navíc dva vstupy, označené R9(1) a R9(2). Funkce je obdobná vstupům R0: pokud jsou oba v log. 1, tak se čítač nastaví do stavu 9 (1001).

Další věc je zase **zapojení napájecích vývodů**. Tyto obvody opět nedodržují tradiční napájení, podobně jako obvod 7475, tak je na to třeba dát pozor.

No a poslední věc, na kterou musím upozornit, je ta, že **čítače nejsou plně čtyřbitové. Jsou uvnitř vlastně dva**, jeden jednobitový, jeden tříbitový. První má hodinový vstup CKA (někdy jen A) a výstup QA, druhý má vstup CKB (někdy značený jen B) a výstupy QB, QC a QD. Pokud chceme, aby obvod fungoval tak, jak jsem ho popsal, tedy jako čtyřbitový, zapojuje se výstup QA na vstup CKB.



Pro zajímavost – takhle vypadaly katalogové informace z katalogu Tesla v 80. letech:



FUNKČNÍ TABULKY

MH . . 90A

VSTUP	VÝSTUPY			
A	Q_A	Q_B	Q_C	Q_D
0	L	L	L	L
1	H	L	L	L
2	L	H	L	L
3	H	H	L	L
4	L	L	H	L
5	H	L	H	L
6	L	H	H	L
7	H	H	H	L
8	L	L	L	H
9	H	L	L	H

MH . . 90A

NASTAVOVACÍ VSTUPY				VÝSTUPY			
$R_{0(1)}$	$R_{0(2)}$	$R_{9(1)}$	$R_{9(2)}$	Q_A	Q_B	Q_C	Q_D
H	H	L	X	L	L	L	L
H	H	X	L	L	L	L	L
X	X	H	H	H	L	L	H
X	L	X	L	ČÍTÁ			
L	X	L	X				
L	X	X	L				
X	L	L	X				

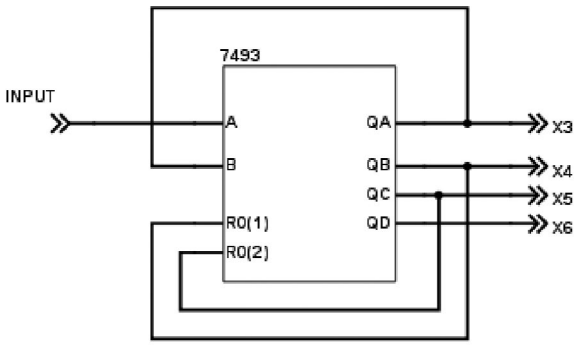
MH7493A

NASTAVOVACÍ VSTUPY		VÝSTUPY			
$R_{0(1)}$	$R_{0(2)}$	Q_A	Q_B	Q_C	Q_D
H	H	L	L	L	L
L	X	ČÍTÁ			
X	L				

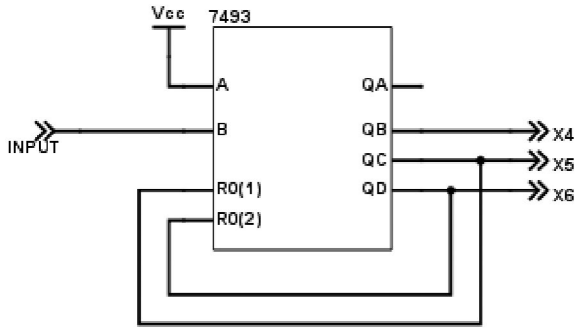
MH . . 93A

VSTUP	VÝSTUPY			
A	Q_A	Q_B	Q_C	Q_D
0	L	L	L	L
1	H	L	L	L
2	L	H	L	L
3	H	H	L	L
4	L	L	H	L
5	H	L	H	L
6	L	H	H	L
7	H	H	H	L
8	L	L	L	H
9	H	L	L	H
10	L	H	L	H
11	H	H	L	H
12	L	L	H	H
13	H	L	H	H
14	L	H	H	H
15	H	H	H	H

Můžete si za sebe zapojit desítkový čítač 7490 a za něj binární čítač 7493 se zkráceným cyklem na 6. Využijte k tomu oba vstupy R0:



Na hodnoty 0 až 5 ale stačí jen tři bity, můžete proto využít jen tříbitový čítač B a výstupy QB, QC a QD:



Všimněte si, že jsem nevyužitý vstup A připojil k napájecímu napětí. Už jsem to zmiňoval, a připomenu to znovu: **Nevyužité vstupy nikdy nenechávejte „bimbat“ jen tak ve vzduchu, vždy je připojte buď k zemi, nebo k napájecímu napětí.**

20.3 Hrací kostka

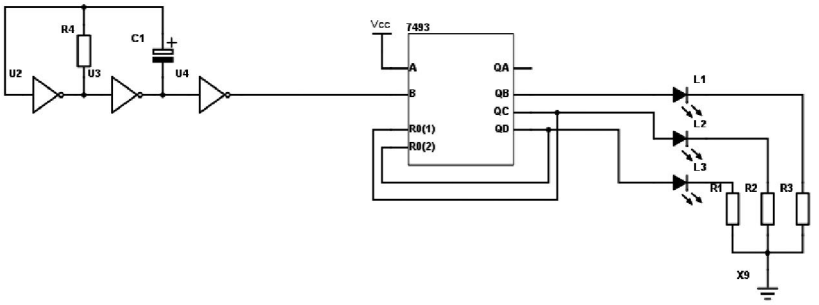
Máme ten čítač 7493, a víme, jak ho zapojit tak, aby dělil šesti. Co tak bychom mohli…?

No jasně: Můžeme rozšířit náš *mincovrhostroj* a udělat z něj hrací kostku. Takovou tu klasickou z Člověče, nezlob se.

Na vstup čítače pustíme ten generátor pulsů 1 kHz, na výstupu se bude rychle měnit 0 – 1 – 2 – 3 – 4 – 5, a my podle toho zobrazíme odpovídající počet teček. Jak na to?

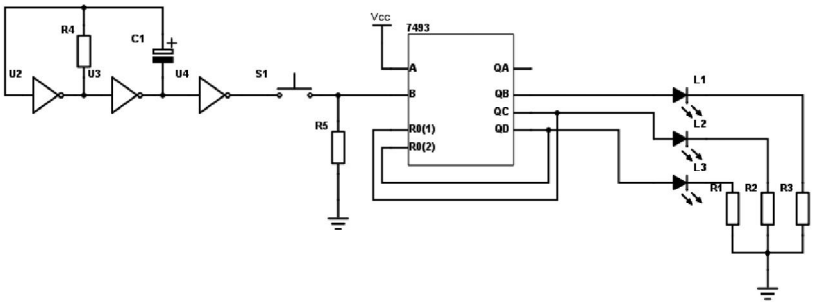
Pojďme si nejdřív postavit „binární kostku“. Výstup budeme zobrazovat pomocí tří LED. Na vstup připojíme náš generátor kmitů, ale pro začátek ho necháme kmitat pomaleji, ať můžeme zkontrolovat, že čítač dělá, co dělat má.

Tip: Nemusíte vždy stavět oscilátor z obvodu 7404, rezistoru a kondenzátoru. Pro tyto testy můžete použít Arduino a jednoduchý příklad Blink. Z pinu 13 (nebo z kteréhokoli jiného) si můžete odebírat potřebný signál, jehož frekvenci i střídu nastavíte pomocí konstant delay(). V takovém případě je nejlepší brát napájecí napětí přímo z Arduina. Pokud budete zbytek obvodu napájet jiným zdrojem, nezapomeňte spojit vodivě zemní napájecí vodiče pro oba obvody!



Použijte obvod 7404 a obvod 7493. Když zvolíte velkou kapacitu C1 a velký odpor R4, bude oscilátor kmitat dost pomalu na to, aby bylo vidět, jak se přepínají stavy na LEDkách. Není to úplně naprosto dokonalá kostka, protože výsledek musíte číst binárně, ale základ máte položený!

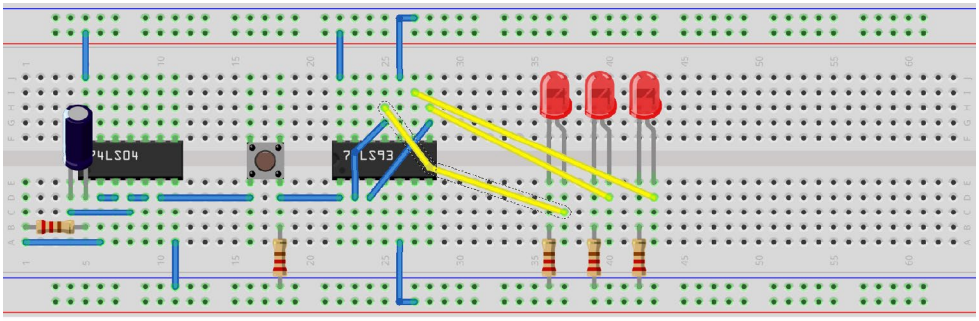
Teď můžete zrychlit a zapojit tlačítko. Pokud bude stisknuté, bude čítač rychle střídat hodnoty 0-5. Jakmile ho pustíte, zůstane na čítači poslední hodnota.



Do zapojení jsem přidal i rezistor R5, protože – kdo to ví? Ano? Přesně tak! Bez rezistoru R5 by byl vstup B „ve vzduchu", pokud by tlačítko nebylo stisknuté. Proto je tam rezistor R5, který v tu dobu na vstup B přivede logickou 0. Jakmile je tlačítko stisknuté, tak „slabou nulu" z R5 přebije výstup z oscilátoru. Jak velký ten rezistor má být? No, můžeme to spočítat z výstupních proudů 7404, ale takové „bastličské pravidlo" říká: *Na místo pullup nebo pulldown dej rezistor 10k, když to bude moc, uber, když málo, přidej.* A věřte nebo ne – pro většinu zapojení to je naprosto vhodná hodnota.

Je to podobné empirické pravidlo, jako je u „rezistoru k LED". Pokud to nepotřebujete nějak extra přesně, nezáleží vám na tom, jak jasný svit bude, a pokud zapojujete LEDku k číslicovým obvodům, dejte 330 ohmů. To je „bezpečný rezistor", který bude pro 99 % LED fungovat.

Tip: *Pokud tuto knihu čtete během studia na technické škole, tak prosím svému vyučujícímu neříkejte, že „tam dáte 10k, protože jste to četli", pracujte tak, jak po vás vyučující chce, a správnou hodnotu si spočítejte! Totéž platí, pokud budete navrhovat a oživovat složitější zapojení. Ovšem pro jednoduché projekty toho typu, jaké si předvádíme my, si vystačíte s výše zmíněnými empirickými pravidly.*

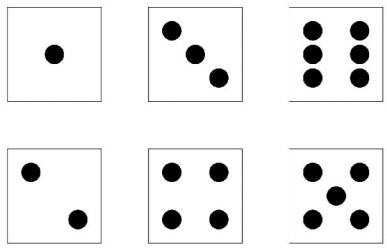


Takhle nějak by mohlo vaše zapojení „binární kostky" na nepájivém poli vypadat. Dokud držíte tlačítko, budou lehce svítit všechny LED (ta vpravo víc). Jakmile tlačítko pustíte, uvidíte jeden z šesti možných stavů.

Proč ta vpravo svítí víc? Odpověď se skrývá v binárních číslech. Střídají se stavy 000, 001, 010, 011, 100 a 101. Během těchto šesti taktů svítí LED vpravo v plné polovině stavů (001, 011, 101). Prostřední LED svítí třetinu doby (010,011), levá taky (100,101). Vzpomeňte si na kapitolu o PWM. Tady je pravá LED buzena polovinu doby, zbývající jsou buzeny třetinu doby.

20.3.1 Reálná kostka

Opravdová hrací kostka není binární. Číslo se ukazuje pomocí kombinace 1 až 6 teček ve známém obrazci.



Označíme si jednotlivé tečky (je jich sedm) pomocí písmen, takto:

A E
B D F
C G

Uděláme si další tabulku, aby bylo jasno, co kdy svítí. V tabulce jsem zapsal i binární vyjádření, tedy stavy, které jsou na výstupu čítače 7493.

Číslo	**Binárně**			**A**	**B**	**C**	**D**	**E**	**F**	**G**
	QD	QC	QB							
1	0	0	0	0	0	0	1	0	0	0
2	0	0	1	1	0	0	0	0	0	1
3	0	1	0	1	0	0	1	0	0	1
4	0	1	1	1	0	1	0	1	0	1
5	1	0	0	1	0	1	1	1	0	1
6	1	0	1	1	1	1	0	1	1	1

Všimněte si jedné zajímavé věci. Některé tečky jsou spolu spojené. Třeba vždy, když svítí A, tak svítí i G. Když svítí B, tak svítí i F. No a C svítí vždy společně s E.

Prostřední tečka (D) se objevuje u kombinací 000, 010 a 100.

D = not QB

Diagonální LED A a G se rozsvěcí ve všech případech, kromě jedničky (což je pro nás stav 000). Můžeme tedy jejich stav zapsat jako logický součet (OR) všech tří vstupních bitů:

A = G = QB or QC or QD

Vodorovné LED (B a F) se rozsvěcí pouze v jednom jediném případě, totiž když je hodnota 6 (pro nás stav 101). Zobrazíme je tedy tehdy, když bude vstup QD i QB v logické 1.

B = F = QB and QD

Druhá diagonála, tvořená diodami C a E, je aktivní pouze pro čísla 4, 5 a 6. Pro nás to jsou hodnoty 011, 100 a 101. Jak bude vypadat výraz pro tuto kombinaci?

20.3.2 Vyjádření logických výrazů

Víme, že C (= E) má být 1 pro kombinace 011, 100 a 101. Pro kombinace 000, 001 a 010 má být 0. Pro ostatní kombinace je nám to jedno, protože ty nenastanou. Jak takovéhle zadání převést na logický výraz?

Vidíme, že C je 1 tehdy, když je QD (nejvyšší bit) v logické 1. Toto pravidlo pokrývá kombinace 100 a 101. Zapíšeme si:

C = E = QD ...

a pokračujeme dál. Musíme totiž pokrýt kombinaci 011.

Tu bychom mohli vyjářit jako „NOT QD AND QC AND QB" – platí, pokud je QD 0, QC 1 a QB taky 1. Jenže my nepotřebujeme úplně přesně odlišit 011 a 111 (protože ta druhá nenastane). Můžeme tedy říct, že je nám stav QD ukradený. Pomůže to nějak? Pomůže! Kombinaci něco-1-1 můžeme vyjádřit jako „pokud nastane QB a zároveň QC". Spojka „a" znamená funkci AND. QB AND QC bude platit pro kombinace 011 a 111, ale to nám nevadí, protože 111 nenastane.

No a výsledek tedy je: C je 1, pokud:

je QD rovno 1

nebo

je QC=1 a QB=1

Tedy:

C = E = QD OR (QC AND QB)

Ptáte se, jestli existuje způsob, jak z tabulky zjistit logický výraz? Existuje jich několik. Jeden z nich, asi nejjednodušší pro funkce tří nebo čtyř bitů, je takzvaná Karnaughova mapa. Do té se teď nechci pouštět, ale myslím, že se k ní ještě vrátím, alespoň v příloze.

Máme tedy sérii výrazů. První vyjádříme pomocí hradla NOT, druhý pomocí třívstupového OR, třetí pomocí dvouvstupového AND, a čtvrtý pomocí dvojice AND-OR.

To znamená, že budeme potřebovat dvouvstupové AND (jeden obvod), třívstupové OR (druhý obvod), dvojici AND-OR (buď třetí obvod, nebo využijeme jedno hradlo AND z prvního obvodu a jedno hradlo OR z druhého obvodu), no a invertor. A ten už tam naštěstí máme.

Trošku problém bude u toho hradla OR. V základní řadě 74xx nejsou třívstupová hradla OR. Existují v řadě 7440xx, což jsou takové TTL náhrady za CMOS obvody. Tam je k dispozici obvod 744075 – trojice třívstupových hradel OR.

Ovšem naštěstí existuje takový technický hack, který nám ušetří hradla OR, a kterému se říká

20.3.3 Montážní OR

Vůbec nevím, jestli vám to prozrazovat, protože to je opravdu ošklivé řešení, ale na druhou stranu: někdy se hodí, a určitě na něj v nějakém zapojení narazíte, tak je dobře ho znát.

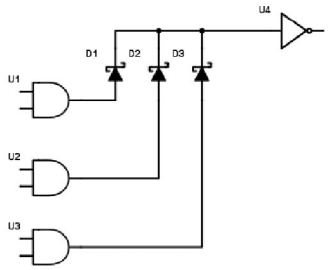
Montážní OR je jeden ze dvou případů, kdy u číslicových obvodů spojíme vývody k sobě. Ale neuděláme to jako Pat a Mat, že bychom je humpolácky spojili dohromady, to ne. Půjdeme na to jinak.

Připomeňme si znovu fungování TTL hradel. Pokud je na výstupu logická 1, je výstup připojený uvnitř k napájení (Vcc) a proud teče ven. Pokud je výstup v logické 0, je tentýž výstup uvnitř hradla spojený se zemí, a *proud teče dovnitř*. Já vím, že to může znít divně, ale je to tak.

Pro vstupy platí analogická situace: pokud je vstup nějak propojen s napájením, teče proud do vstupu a tvoří logickou 1. Pokud je vstup spojen nějak se zemí, teče proud uvnitř obvodu z Vcc přes tranzistor do vstupu a vstupem ven.

Pokud bychom natvrdo spojili dva výstupy, nic by se nestalo – pokud by byly oba ve stejné úrovni. Pokud by ale jeden byl v 1 a druhý v 0, tak by z toho v 1 tekl proud do toho v 0, a výsledek by byl nesmyslný.

Naštěstí máme součástku, která umí zařídit, aby proud tekl jen jedním směrem. Pamatujete? Ano, je to dioda. Takže když mezi výstupy hradel a „bod spojení“ zapojíme diody, zařídíme tím, že proud poteče jen jedním směrem.



Máme tu tři hradla U1 až U3, a jejich vývody jsme spojili přes diody. Výsledek je připojen na vstup hradla U4. Co se teď stane?

Pokud jsou všechny tři výstupy hradel U1-U3 v logické 0, proud by rád tekl z U4 do vstupů těchto hradel, ale nemůže, brání mu v tom diody. Takže na vstupu U4 je logická 0.

Když některé z hradel, třeba U1, přepne do log. 1, proud se z výstupu dostane přes diodu D1 do bodu spojení a na vstup hradla U4, ale už nemůže odtékat zpátky přes výstupy hradel U2 a U3, protože tomu brání diody D2 a D3.

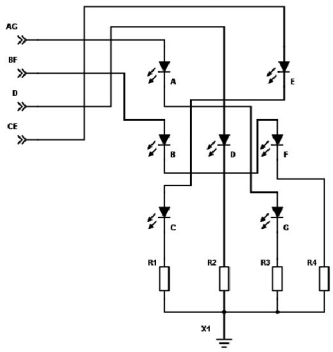
Výsledná funkce takového zapojení je „log. 1, pokud je alespoň jeden ze vstupů v logické 1" – tedy funkce OR.

Výhoda takového zapojení je jediná: můžeme jednoduše vytvořit OR s několika vstupy jen pomocí diody, nemusíme k tomu vyplýtvat integrovaný obvod.

Nevýhodou takového zapojení může být třeba to, že na diodách vzniká úbytek napětí, který může posunout hodnoty napětí do zakázaného pásma. Tomu můžeme předejít tím, že použijeme diody s nízkým úbytkem (např. Schottkyho). Jsou i další nevýhody, například to, že dioda má nějakou nenulovou kapacitu, takže může ovlivnit vysokofrekvenční signály atd. Ale někdy holt výhoda jednoduchosti převýší nad nevýhodami, a proto se s „montážním OR" čas od času setkáte.

20.3.4 Zobrazovací obvod hrací kostky

Zapojme si tedy sedm LED tak, aby dávaly dohromady stejné obrazce, jako jsou na hrací kostce. Využijeme toho, že některé jsou vždy ve stejném stavu (A a G, B a F, C a E) a zapojíme to třeba takto:



A teď otázka: **Mohu to takto zapojit?** Mohu zapojit dvě LED za sebe (sériově)? Brání mi v tom něco?

Může mi v tom teoreticky zabránit úbytek napětí na diodě. Když zapojím dvě za sebe, a na každé vznikne úbytek 2 volty, tak bude celkový úbytek 4 volty, což je ještě OK. Třetí LED by vyžadovala už zvýšení napájecího napětí, s ní bychom se do pěti voltů nevešli.

(A co když je zapojím vedle sebe, co se stane?)

20.3.5 Dekodér hrací kostky

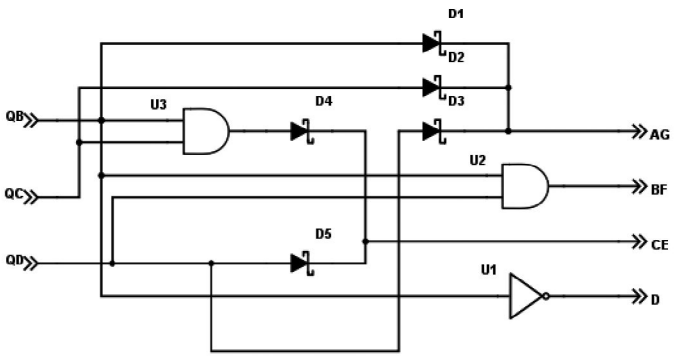
Chybí teď ta podstatná část, totiž jak ze signálů QB, QC a QD z čítače udělat požadované signály AG, BF, CE a D.

U D to mám nejjednodušší, ten připojím přes invertor U1 (mám ho třeba v oscilátoru 7404) na QB.

Signál AG vytvořím třeba montážním OR pomocí tří Schottkyho diod (D1, D2, D3) ze signálů QB, QC a QD.

Pro signál BF musím použít hradlo AND (U2), třeba obvod 7408. Sloučím signály QB a QD.

Signál CE vygeneruju tak, že pomocí hradla AND (U3) spojím signály QB a QC, a výsledek pomocí montážního OR spojím se signálem QD (diody D4, D5).



Zde je opravdu na místě použít Schottkyho diody, které mají malý úbytek napětí. Nezapomeňte na to, že kvůli zapojení LED do série potřebujeme na vstupu alespoň 4 volty. Kdybychom použili normální diody s úbytkem 0,7 voltu, bylo by to už dost na hraně.

Existuje řešení problému s napětím? Co třeba zapojit dvojice LED ne sériově, za sebe, ale paralelně, vedle sebe? Snížili bychom potřebné napětí na polovinu, ale vzrostl by zase potřebný proud. Najděte si v datasheetu obvodu 7493, kolik proudu je schopen dát na vývodech při stavu log. 1 – bude se to lišit pro obvody 74LS93, 7493, 74ALS93 a třeba 74HCT93. Většinou platí, že u obvodů HC(T) bývá maximální proud okolo 20 mA. U typu 74LS93 to je, pro zajímavost, 16 mA. Ale není dobré součástky napínat na mezní hodnotu. Když navrhneme rezistor tak, abychom odebírali 10 mA, bude na každou diodu připadat 5 mA, což je dostatečný proud. Rezistor tak bude mít velikost R = U / I, tedy 5 / 0,01 = 500 ohmů. Použijte rezistor nejbližší vhodné hodnoty, tedy 470 ohmů...

Zkuste si zapojit takovou kostku na nepájivém kontaktním poli. A až si dostatečně zaházíte, podíváme se na další čítače.

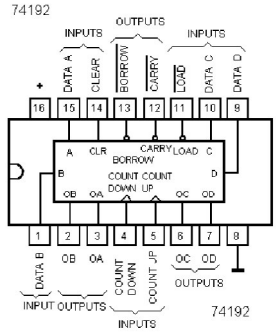
20.4 Další čítače

Ukázali jsme si zatím takzvané asynchronní čítače – na vstup je přivedený signál, a ten se postupně šíří vnitřními klopnými obvody z jednoho do druhého. Takže se hodnota mění sice rychle, ale přesto postupně, což může někdy vadit. Proto se v takových situacích používají takzvané **synchronní čítače** – ty mají zavedený jednotný časový signál pro všechny stupně, takže se všechny výstupy mění naráz.

U obvodu 7490 jsme viděli kromě nulovacích vstupů R0 i nastavovací vstupy R9. Obecnější verze čítačů umožňuje nastavit libovolnou hodnotu – říká se tomu „čítač s přednastavením“ (preset).

Některé čítače umí čítat nahoru (0 – 1 – 2 – 3 – 4 …) i dolů (7 – 6 – 5 – 4…).

Kombinací všech zmíněných funkcí oplývají čítače 74192 a 74193. Opět jde o čtyřbitové čítače – typ 193 je binární, typ 192 desítkový.

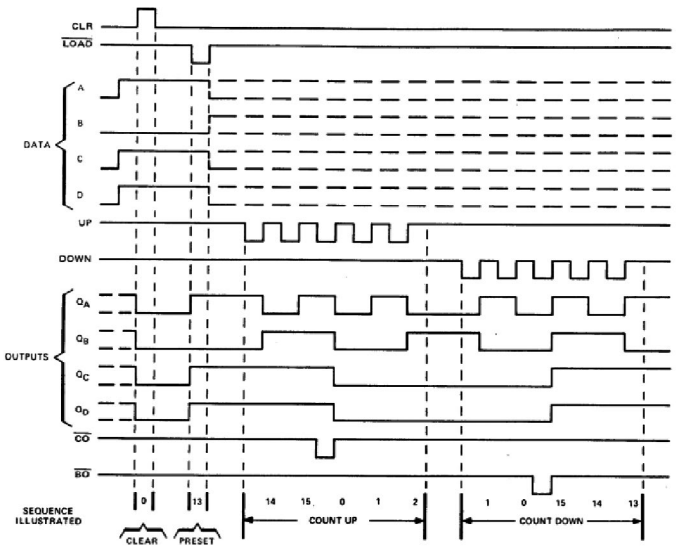


Obvod má dva hodinové vstupy, CountUp a CountDown. První funguje jak jsme zvyklí a vyvolává čítání vzhůru. CountDown vyvolává čítání dolů. Čítače mají čtyři výstupy: QA, QB, QC, QD. Navíc mají i výstupy „/Carry" a „/Borrow" – /Carry označuje, že čítač příštím pulsem přejde ze stavu 15 do stavu 0 (respektive ze stavu 10 do stavu 0 u desítkového). Signál je negovaný, takže tuto skutečnost oznámí, logicky, nulou. Podobnou funkci má signál /Borrow – oznamuje, že čítač je ve stavu 0, a při dalším pulsu CountDown přejde na nejvyšší hodnotu (15, resp. 10). Tyto výstupy se používají k řetězení čítačů za sebe, pokud chcete získat vícebitový čítač.

Další vstup je známý nulovací vstup, zde označený CLR – Clear. Vynuluje celý čítač.

Poslední pětice vstupů jsou vstupy A, B, C, D a /LOAD. Signál /LOAD říká, že se do čítače má zapsat hodnota ze vstupů A, B, C, D. A protože je /LOAD negovaný, děje se tak při změně úrovně z 1 (klidový stav) do 0 (aktivní stav).

Vím, že někdy obrázek vydá i za 273 slov, tak se podívejme, jak funkci obvodu 74193 popisuje graficky datasheet:



No dobře, uznávám, tento obrázek není právě krystalickou ukázkou, ten v hlavě začátečníka vydá sice za 273 slov, ale většina z nich je „cože?" Ale protože takových obrázků jsou plné datasheety, pojďme tomu dát trochu práce.

V grafu jsou vyznačeny časové průběhy signálů. Vlevo jsou signály popsané – máme tam CLR, /LOAD, datové vstupy, výstupy, signály UP a DOWN a výstupy přenosu. Graf zachycuje průběhy

ve významných okamžicích – neznamená to, že to takto musí jít za sebou, to ne!

Jako první je událost CLEAR. Vidíte, že nastala příchodem pulsu na vstup CLR a měla vliv na výstupy QA až QD. Ty předtím mohly mít jakýkoli stav (naznačeno přerušovanou čárou v log. 1 i log. 0), ale jakmile přišla vzestupná hrana CLR, tak se na těchto výstupech nastavila 0.

Vedle je popsána událost PRESET. Všimněte si, že CLR je v log. 0 a přišla sestupná hrana na vstupu /LOAD. S ní se výstupy QA až QD nastavily podle stavu na vstupech A až D. (Tady je to trošku nejednoznačné, protože stejné úrovně jsou nakreslené na vstupech i pro událost CLEAR; správně by měly být čárkované, protože na nich při nulování nezáleží.)

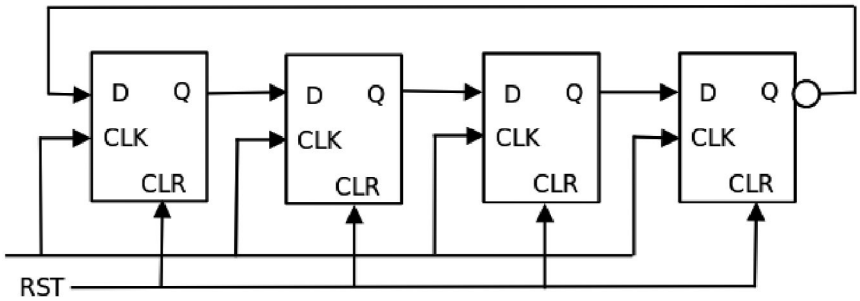
Další události, jaké jsou znázorněné, probíhají při CLR = 0 a /LOAD = 1. Na hodnotě vstupů A – D nezáleží. Nejprve se ukazuje, co se děje při pulsech na vstupu UP (= čítání nahodu, count up), poté co dělají pulsy na vstupu DOWN. Sledujte, jak se mění hodnoty na výstupech, všimněte si hlavně *kdy* se mění (při vzestupné hraně signálů UP a DOWN), všimněte si, *jak* reagují signály /CO a /BO (přenos, resp. výpůjčka)…

20.5 Ještě nějaké čítače?

Samozřejmě, že jsou na skladě. Třeba takový Johnsonův čítač.

Představme si řetěz obvodů typu D, podobně jako u výše zmíněných čítačů. Každý vstup D je připojen na výstup Q předchozího obvodu. Vstup D prvního obvodu v řadě je zapojen na **invertovaný výstup /Q** posledního obvodu.

Všechny klopné obvody mají spojený vstup CLK, a všechny mají vyvedený vstup CLR (Reset).



Začíná se od stavu 0000, a čítač nabývá postupně hodnot 0000, 0001, 0011, 0111, 1111, 1110, 1100, 1000 – a pak zase od 0000. Počítá tedy v takzvaném „Johnsonově kódu“…

20.5.1 Johnsonův kód

Klasický binární kód má jednu nevýhodu – někdy se mění hodnota více bitů najednou, třeba při přechodu ze stavu 0011 do stavu 0100 se mění rovnou tři bity. Pokud změna probíhá pomaleji a ne (ideálně) najednou, objeví se postupně sekvence velmi rychlých změn, např. 0011 -> 0010 -> 0000 -> 0100. Pokud je další zařízení dostatečně rychlé, může tyto změny zaznamenat a fungování obvodů bude narušené. Někdy to lze vyřešit změnou obvodového řešení, ale někdy taková možnost není. Proto se používají i kódy, kde se při přechodu mezi stavy mění vždy hodnota jednoho jediného bitu. Johnsonův kód je příklad jednoho takového kódu. Pro N bitů může nabývat 2N kombinací, jak jsme si ukázali výš.

Čtyřbitový Johnsonův kód tvoří postupně tyto hodnoty: 0000, 0001, 0011, 0111, 1111, 1110, 1100, 1000.

20.5.2 Grayův kód

Další podobný kód je kód **Grayův**. Vychází z binárního kódu, ale čísla jsou přeuspořádaná tak, že mezi dvěma po sobě následujícími stavy se také mění hodnota právě jednoho jediného bitu:

Hodnota	Binárně	Grayův kód
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Grayův kód se používá místo binárního tam, kde by případná změna více bitů nemusela proběhnout přesně ve stejný čas. Navíc se poměrně snadno převádí na binární a zpět, pomocí operace XOR. Pokud si bity binárního kódu označíme jako b3-b0 a Grayova kódu jako g3-g0, tak převodní vzorce jsou jednoduché:

g0 = b0 XOR b1

g1 = b1 XOR b2

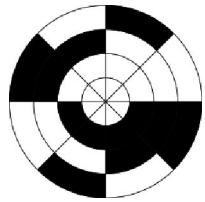
g2 = b2 XOR b3

g3 = b3 XOR 0 = b3

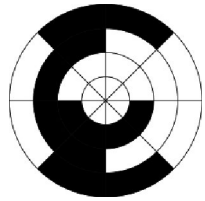
Tedy vždy jako XOR daného bitu a bitu o řád vyššího (u nejvyššího bereme nulu).

Představte si, že máte cosi, co se otáčí, a vy chcete vědět, jakým směrem je to natočené. Pro jednoduchost řekněme, že vám to stačí znát s přesností plus minus 45 stupňů. Připevníte k otáčivé věci kolo, to si rozdělíte na osm segmentů, a protože osm hodnot zakódujete do tří bitů, tak si uděláte tři soustředné kružnice. A v nich si některé segmenty vybarvíte a jiné ne, takže pak budete moci prostým způsobem, třeba optickým snímačem, určit, jak je kolo natočeno.

První pokus bude vypadat nějak takto:



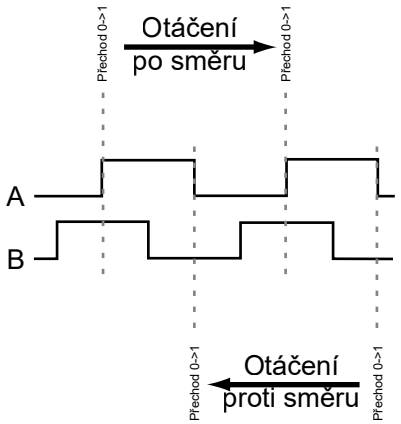
Jenže stačí drobná nepřesnost, a hned v pozici „nahoře“ vám vznikne přechod 001 – 011 – 010, popřípadě 001 – 000 – 010. Když použijeme Grayův kód, nic takového nehrozí:

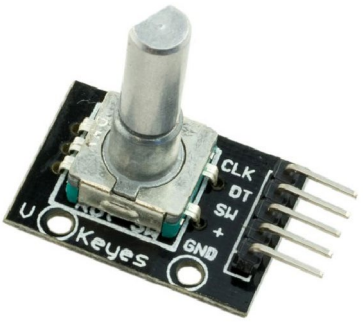


20.6 Rotační enkodér

Samozřejmě, že se takovéhle věci používají i v praxi. Těm součástkám se říká **rotační enkodéry** *(rotary encoder)*, a často se používají pro ovládání různých zařízení. Vypadají jako potenciometry, ale mohou se točit „donekonečna“, a většinou cítíte jemné vrčení, když jimi otáčíte.

Taková součástka má uvnitř podobný kotouček s Grayovým kódem. Má dva vývody, na kterých se střídají hodnoty 00 – 01 – 11 – 10, nebo 00 – 10 – 11 – 01, to podle směru otáčení. Na jednu otáčku mívají typicky 20 kroků, a navíc bývají kombinovány s tlačítkem, takže lze zařízení ovládat pomocí otáčení do dvou směrů a mačkání tlačítka.





Jak takové signály číst? No, jsou dva možné způsoby. Buď budete vzorkovat oba signály a podle přechodové funkce zkoušet, jestli se ovladač otáčí doleva nebo doprava. No a nebo se podíváte na graf průběhů výše a dojde vám to…

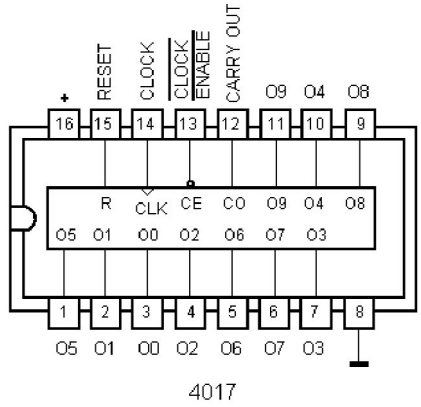
Stačí signál A nazvat „hodinovým“ (CLK) a hlídat, kdy se změní z 0 na 1 (viz obrázek). V ten okamžik přečtěte stav na vstupu B (nazvu ho datovým, DT). Pokud je 1, znamená to otáčení jedním směrem, pokud 0, znamená to otáčení druhým směrem. *Za domácí úkol si zkuste, co se stane, když vstupy A a B prohodíte.*

Já vím, tohle je trošku divoké na představivost, ale to půjde… Sledujte okem první řádek, signál A, a hledejte vzestupnou hranu – tedy změnu 0–1. Jakmile ji najdete, podívejte se, jakou hodnotu má právě v tu chvíli signál B. Když jedete okem zleva doprava, tak vzestupné hrany přicházejí vždy ve chvíli, když je na vstupu B logická 1. Když to zkusíte v protisměru a pojedete okem zprava doleva, přijde změna 0->1 vždy ve chvíli, kdy je vstup B v logické 0.

20.7 Čítač s dekodérem 1-z-10 typu 744017

Už jsem tu zmínil, že kromě TTL řady 74xx existovala i řada CMOS 40xx, v níž bylo několik velmi zajímavých obvodů. Protože konstruktéři tyto obvody používali a museli řešit přizpůsobení úrovní mezi CMOS a TTL, sáhli výrobci k řešení: vyrobili funkčně a vývodově shodné obvody.

Příkladem takového obvodu může být 4075, což je trojice třívstupových hradel OR, a její TTL ekvivalent 744075 (74HCT4075). Zajímavý obvod je 4017, v TTL verzi jako 744017 – např. 74HCT4017. Tento obvod kombinuje pětibitový Johnsonův čítač (tedy s deseti hodnotami) a dekodér 1-z-10, takže má deset výstupů, pro každou hodnotu jeden. Což se může hodit v nejrůznějších situacích.



Obvod má nulovací vstup R, hodinový vstup CLK a povolovací vstup /CE – pokud je neaktivní (=1), čítač ignoruje vstup CLK, pokud je aktivní (=0), vstup CLK funguje. Z výstupů Q0 až Q9 je aktivní právě jeden, podle aktuálního stavu. Negovaný výstup /CO (Carry Out) je neaktivní (=1) pro hodnoty 0 až 4, pro hodnoty 5-9 je aktivní (=0).

U tohoto čítače můžeme zkrátit periodu velmi snadno – chceme-li například omezit hodnoty na 0-5, propojíme výstup Q6 s resetovacím vstupem.

Pokud jej použijeme pro naši hrací kostku, nebudeme potřebovat další integrovaný obvod, namísto toho použijeme pouze montážní OR, protože:

D = Q0 or Q2 or Q4

AG = Q1 or Q2 or Q3 or Q4 or Q5

BF = Q5

CE = Q3 or Q4 or Q5

20.8 Počítadlo k autodráze

Neejn k autodráze, samozřejmě. Můžet ho použít kdekoli, kde je potřeba počítat, kolikrát se něco stane.

U autodráhy třeba průjezd autíčka. Když už se závodí, tak ať je jasné, kdo projel kolik kol. Jak na to?

No, bude potřeba zjistit, že autíčko projelo nějakým měřicím bodem. Možností je, jako vždy, několik, ale nejjednodušší bude spolehnout se na staré dobré světlo. Šel by sice použít třeba mechanický spínač, ale předpokládám, že by autíčko mohlo mít trošku problém a ve vysoké rychlosti by mohlo vyletět z dráhy. Světlo bude lepší.

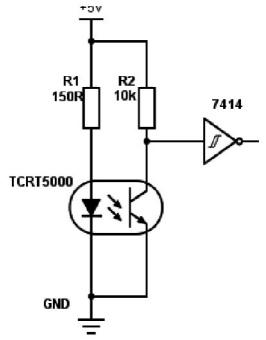
Když světlem, tak jak? Buď můžete vzít LED, tu dát na jednu stranu dráhy, na druhou naproti dát fotorezistor, a pak sledovat, kdy dojde k zatmění. Pardon, k zastínění. Pro zvýšení citlivosti můžete dát na obě součástky malé plastové trubičky a nasměrovat je přesně proti sobě, aby je tolik neovlivňovalo okolní světlo.

Druhá možnost je použít takzvané reflexní snímače, známé též pod označením „detektory přiblížení“ (anglicky „optical reflective sensor“, případně „line tracking sensor“ – třeba typ TCRT5000). Technicky to je LED a fototranzistor v jednom bloku:



Pokud nemáte autíčka úplně černá, tak fototranzistor zachytí světlo, odražené od jeho povrchu, a dokáže tak detekovat průjezd.

Zapojení takového snímače je jednoduché: LED připojte přes rezistor k napájecímu napětí. Fototranzistor zapojte podobně jako tlačítko, tedy kolektor přes pull-up rezistor cca 10 kΩ k napájecímu napětí, emitor k zemi, a výsledek odebírejte z kolektoru. Na výstup dejte, kvůli snazšímu zpracování signálu, Schmittův klopný obvod (třeba v invertoru 7414).



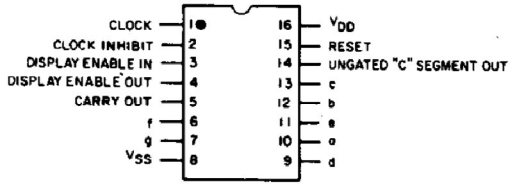
Za normálního stavu LED svítí, ale na fototranzistor nic nedopadá. Fototranzistor je tedy zavřený, na vstup invertoru jde přes rezistor 10k napájecí napětí, tedy log. 1, a na výstupu je logická 0. Jakmile se před senzor dostane autíčko, LED osvítí jeho povrch, odražené světlo sepne fototranzistor, tím se na vstup Schmittova obvodu dostane zem, a na výstupu bude 1.

S touto informací už můžete dál pracovat. Nejjednodušší je zase připojit Arduino s displejem, počítat pulsy a zobrazovat je na displeji.

Pojďme ale zvolit „staré dobré obvodové řešení“. Existuje totiž obvod, který v sobě kombinuje desítkový čítač, dekodér pro sedmisegmentovky a budič. Jeho označení je SN74143. Když se podíváte do datasheetu, zjistíte, že je to přesně to, co potřebujeme.

Plni nadšení ho jdete koupit, a zjistíte, že ho pravděpodobně neseženete. V katalogu už je označen jako „Obsolete“ a už se ani nevyrábí.

Podobný obvod se ale stále vyrábí v CMOS řadě 40xx, tentokrát s označením CD4026. V pouzdru DIP16 je desítkový čítač, dekodér a budič.



Vstup 1 slouží k přivedení vstupních pulsů. Vstup 2 (clock inhibit) slouží k přerušení čítání. Připojte ho na zem (nechceme přerušovat). Vstup 3 (display enable) slouží ke zhasínání displeje. Nechceme, proto jej připojte na napájecí napětí. Výstup 4 nechte být (odpovídá hodnotě na vstupu 3), výstup 5 taky (pokud nebudete dávat dva čítače za sebe, abyste získali vícemístné počítadlo). Vývody 6, 7, 9, 10, 11, 12, 13 odpovídají jednotlivým segmentům LED sedmisegmentovky, Vss je zem a Vdd je napájení (CMOS můžete napájet až 15 volty). Vstup 15 je RESET, ten si připojte přes tlačítko na napájecí napětí, a k němu pulldown rezistor. Získáte tak možnost vynulovat počítadlo. Vývod 14 ponechte být.

Sedmisegmentovku (zvolte typ „se společnou katodou“) připojte samozřejmě přes oddělovací rezistory. Zapojte si to takto na kontaktním poli, jako vstup dejte zase tlačítko s pulldown rezistorem, a zkoušejte, jestli vše počítá jak má.

Proč se společnou katodou? Protože pak má sedmisegmentovka vyvedené anody od segmentů a katody spojené do jednoho vývodu. Jednotlivé segmenty svítí, když se na ně přivede napájení. Sedmisegmentovky se společnou anodou to mají obráceně, ty svítí, když se vývod segmentu propojí se zemí. Šly by použít, ale museli byste buď zapojit invertory, nebo tranzistory.

Když se začtete do datasheetu, zjistíte, že vstupy CLOCK a CLOCK INHIBIT mají na vstupu Schmittův klopný obvod, takže můžeme ten invertor vynechat. Hurá, o jeden obvod méně.

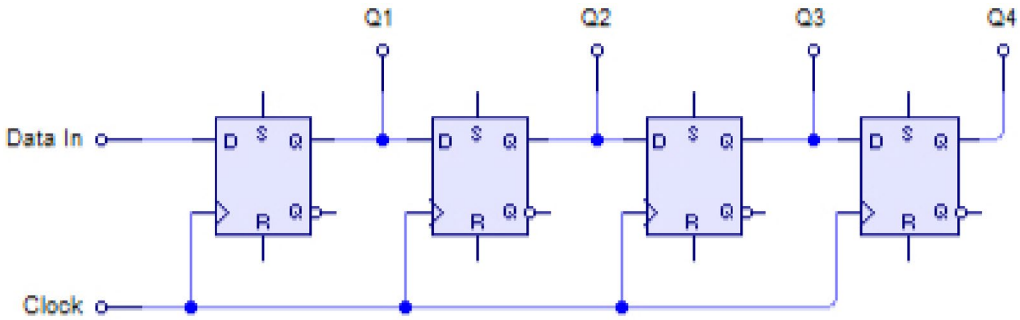
Problém je, že teď naše čidlo má na výstupu 1 a průjezd autíčka způsobí 0, zatímco obvod 4026 očekává naopak vzestupné pulsy. Ale lze to vyřešit velmi elegantně: v datasheetu naleznete vnitřní schéma obvodu. V něm uvidíte, že vstupy CLOCK a CLOCK INHIBIT jsou vlastně stejné, jen INHIBIT je negovaný. Uvnitř jsou pak spojené hradlem AND. Takže můžeme připojit výstup senzoru na CLOCK INHIBIT, a CLOCK zapojit na napájecí napětí. Za normálního stavu tak bude na vnitřních hodinách 0 (protože CLOCK INHIBIT=1), jakmile projede autíčko, vypne se CLOCK INHIBIT (=0), a tím se uvnitř obvodu vytvoří vzestupná hrana.

Pokud zjistíte, že jeden průjezd autíčka vyvolá vícero impulsů, můžete zkusit buď umístit senzor jinam, nebo třeba polepit autíčko stříbrnou fólií, popřípadě na výstup z čidla dát kondenzátor, který odfiltruje rychlé pulsy a sleje je do jednoho. Čítači to, díky Schmittovu obvodu, vadit nebude.

21 Posuvné registry

21 Posuvné registry

Vraťme se teď myšlenkami o kousek zpátky. Johnsonův čítač… Klopné obvody typu D, zapojené za sebou… máte? Co to trochu zobecnit?



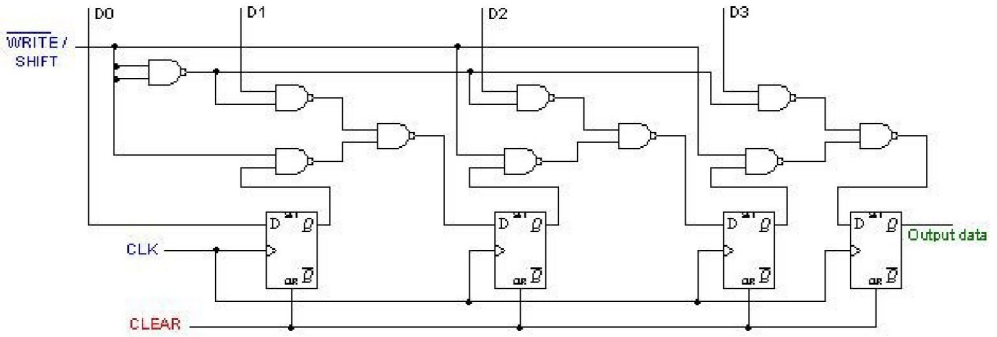
Představte si takovéhle zapojení. S každým pulsem hodin se do levého klopného obvodu zapíše informace ze vstupu Data In, do druhého obvodu se zapíše předchozí informace z prvního, do třetího se zapíše informace z druhého… No a nakonec po čtyřech pulsech hodin se zapíšou čtyři bity hezky postupně do všech klopných obvodů. Nějak takhle – představte si, že po každém řádku přijde puls na vstup Clock:

Data In	Q1	Q2	Q3	Q4
0	x	x	x	x
1	0	x	x	x
0	1	0	x	x
1	0	1	0	x
1	1	0	1	0
0	1	1	0	1
x	0	1	1	0

Všimněte si, že vlastně takto bereme bity, které jdou za sebou, sériově, po jednom vodiči Data In, a převádíme je do paralelní podoby.

Takovému uspořádání se říká **posuvný registr** – důvod je asi jasný: každý puls hodin posouvá informaci uvnitř o jednu pozici dál. Tento typ se označuje jako SIPO: **Serial In**, **Parallel Out**, tedy *sériově dovnitř, paralelně ven*.

Samozřejmě existují i opačné registry PISO. Mají několik datových vstupů (většinou 4 nebo 8), pak signál, kterým se zapíše tato informace do vnitřních klopných obvodů, a pak hodiny, které posílají jednotlivé bity na výstup Q. Jeho zapojení je ale o něco složitější.



Kombinací obého dostáváme hybridní registry SISO-PISO-SIPO, tedy takové, které mohou mít vstupy i výstupy obojího typu. Častá kombinace je SISO-SIPO, tedy registry, které mají sériový vstup i výstup, a k tomu paralelní výstup. Lze je totiž snadno zapojit za sebe a poskládat tak třeba dvacetibitové převodníky.

K čemu je taková věc dobrá? No, třeba když potřebujete přenést signál na delší vzdálenost, tak je dobré převést jej nejprve na sériovou podobu, tedy na proud bitů, přenést je hezky po jednom, a pak opět poskládat dohromady do celých slov.

Proč?

Víte co, pojďme si zatím říct něco o komunikačních rozhraních, pak to bude třeba jasnější.

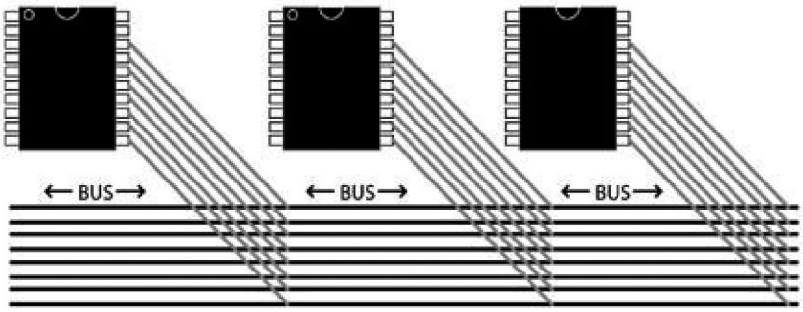
22 Paralelní a sériová rozhraní

22 Paralelní a sériová rozhraní

Začnu zase jazykovou vsuvkou. Za socialismu se politika promítala do všeho, a i naprosto nenápadné věci byly vlastně politické. Například v technice se soudruhům nelíbilo velké množství anglických výrazů v názvosloví, a snažili se je nahrazovat českými. Tak vzniklo třeba sousloví „stykové rozhraní". Kdybyste tápali, co to je – je to „interface". Anglické slovo je složené ze dvou výrazů, „inter", tedy mezi, a „face", tedy obličej. S trochou obrazotvornosti vidíte přenos informací, který se odehrává mezi dvěma obličeji. Stykové rozhraní zní spíš jako něco ze sexuologie, ale je to totéž. Některé výrazy byly ještě mnohem horší, třeba takový magnetoskop – složenina podle vzoru „magnetofon", která měla nahradit prohnilý západní termín „video" – a po právy vymřely. Ze stykového rozhraní nám zůstalo alespoň to rozhraní, i když se běžně používá i počeštělá verze anglického výrazu interfejs. Jazyk si s tím poradí, přidá koncovky, zavede flexi, a tak můžeme běžně číst o interfejsech a pracovat s interfejsem.

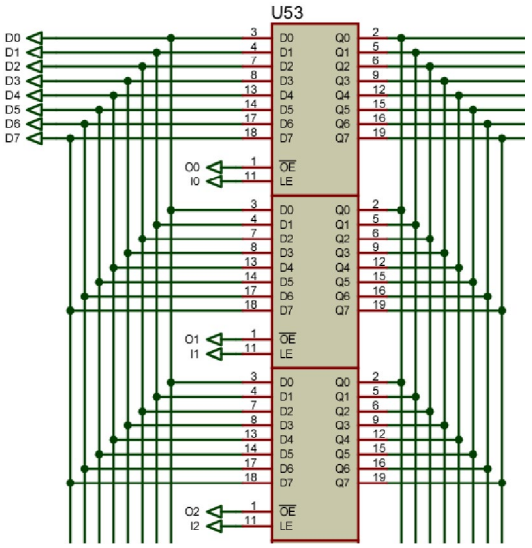
Ale to jen na okraj.

Od počátku číslicové techniky se ukázala potřeba nějak přenášet vícebitová data. Představme si osmibitový mikroprocesor – musí nějak komunikovat s pamětí a dalšími obvody. Nejjednodušší je přenášet osmibitová data po osmici vodičů. A přesně to se dělalo.

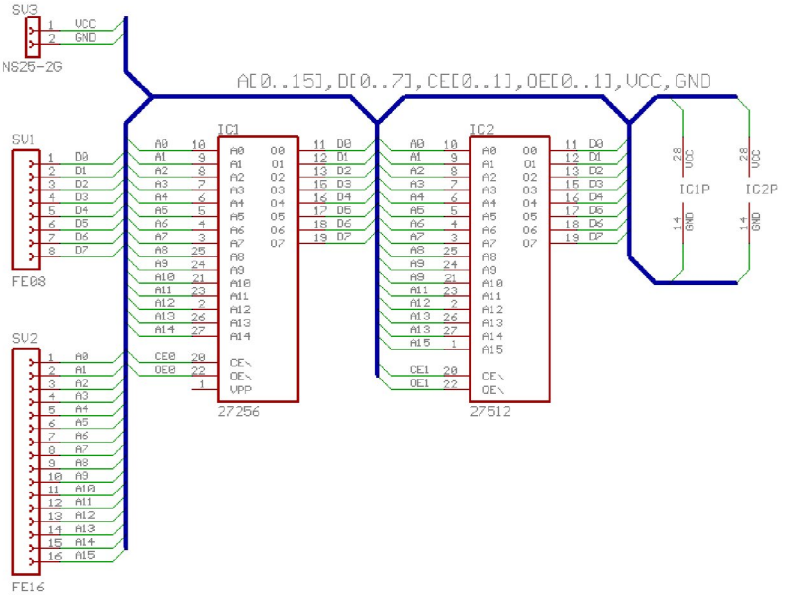


Datové vodiče se jmenovaly D0 až D7, a všechno, co mělo nějak přebírat data z procesoru, se posílalo po těchto osmi vodičích.

Dohromady se jim říkalo (a říká stále ještě) **datová sběrnice**. Proč sběrnice? Protože se k těmto vodičům připojovalo více zařízení (paměť RAM, paměť ROM, vstupně-výstupní obvody atd.) najednou, a ty samotné vodiče jako by *sbíraly* data z různých zařízení. Samozřejmě – aby se obvody nepraly o to, kdo bude na datové vodiče posílat informace (vzpomeňte si, co jsme si říkali o připojování výstupů logických obvodů k sobě), tak jsou všechny vybavené třístavovým výstupem, tedy takovým, který umožňuje odpojení vývodu, a procesor si vždy řekne, které zařízení má komunikovat.



Abychom se z toho ve schématech nezbláznili, jak taháme osm drátů kolem dokola, tak namalujeme jednu tlustou čáru, řekneme, že pro nás představuje třeba osm vodičů D0-D7, a pak jen malujeme součástky a říkáme, jak se mají napojit:



Protože po takovéhle sběrnici jde těch osm bitů vedle sebe osmi vodiči, tak říkáme, že se jedná o **paralelní přenos dat**.

Má své výhody – třeba že přenášíme osm bitů najednou. Má ale i své nevýhody: musíme opravdu fyzicky natáhnout osm spojů, což je někdy protivná činnost a snadno se na nějaký zapomene. Druhá, větší nevýhoda je, že takto můžeme přenášet data jen na krátké vzdálenosti. Pomocí obyčejných vodičů tak na desítky centimetrů, pomocí speciálních kabelů třeba i na metr a půl, ale na delší vzdálenosti se přenos rapidně horší – dochází k chybám, k rušení, k velkému útlumu, navíc všechny nectnosti rostou s tím, jak roste rychlost přenášených dat (tedy frekvence změn). Pak už ani pořádně stíněné kabely nepomohou.

Co v takovém případě? Pak je lepší sáhnout po sériovém přenosu dat. K němu vám stačí jen pár vodičů – dva, tři, čtyři, podle typu přenosu. Se sériovým přenosem sice přenášíte data pomaleji než paralelně (máte typicky jen jeden datový vodič místo osmi), ale lépe se stíní, je odolnější proti rušení, můžete jej proto vést na delší vzdálenost…

Typickým příkladem bylo připojování tiskáren – kdysi se používalo paralelní rozhraní, známé jako Centronics. V počítačích PC bylo skryté pod označením LPT – starší počítače, ještě tak z roku 2005, ho mají na základní desce, později se už nepoužívalo. Kabely k paralelnímu rozhraní byly tlusté, krátké a drahé.

Některé tiskárny proto používaly sériové rozhraní RS-232 což je známý COM port. Takové rozhraní si v principu vystačí se třemi vodiči: data do zařízení, data ze zařízení a uzemnění. Kabel pro sériové rozhraní mohl mít třeba pět metrů, dražší a kvalitnější i víc.

Sériové rozhraní se používá dodneška pro připojení většiny periferních zařízení k počítači. Nemýlíte se, známé USB znamená *Universal Serial Bus*, a je to přesně stejný princip přenosu dat, jen vylepšený tak, aby bylo možno připojit víc zařízení naráz a používat delší kabely.

Sériový přenos se používá i v průmyslu – možná jste se setkali s rozhraním RS-485. Dokonce i přenos videosignálu po kabelu HDMI, kterým máte pravděpodobně připojený monitor k počítači nebo třeba herní konzoli k televizi, probíhá, jak jinak, sériově.

I počítačové sítě dnes většinou fungují na principu sériového přenosu dat –Ethernet, tedy „ten kabel na sítě“, přenáší data po dvou dvojicích vodičů (další dvě dvojice jsou nepoužité).

Aby se zlepšila odolnost proti rušení indukcí, používá se u sériových rozhraní, která přenášejí data na větší vzdálenost, takzvaný „diferenciální přenos“ – jeden signál není přenášen jedním vodičem, ale kroucenou dvojicí vodičů, z nichž jeden přenáší signál, druhý jeho negaci. Přijímač pak oba vodiče porovná, a z nich dekóduje zpět přenášený signál. Vychází se z předpokladu, že případné rušení zasáhne oba vodiče stejně – oběma buď přidá, nebo ubere – ale rozdíl zůstane stejný.

Paralelní rozhraní se používá tam, kde je potřeba obrovská rychlost a stačí krátké vzdálenosti – typicky mezi procesorem a pamětí. Dříve se používalo i pro připojení dalších komponent, jako grafické karty nebo disků. Dnes se postupně přešlo i u disků a karet na sériové rozhraní (u disků SATA – Serial ATA, u karet PCI-Express, což je taky sériové rozhraní). Důvodem je zvyšování přenosové rychlosti – u sériové sběrnice je možné dosáhnout řádově vyšších rychlostí a frekvencí bez přeslechů a problémů s časováním, což trochu kompenzuje fakt, že přenášíme míň bitů najednou.

Vlastnost	**Paralelní**	**Sériové**
Počet najednou přenášených bitů	více (typicky 8, 16)	Většinou jen jeden
Přenosová rychlost	Nižší	Vyšší
Vzdálenost, na jakou je možno přenášet data	Menší	Větší
Odolnost k rušení	Malá	Větší

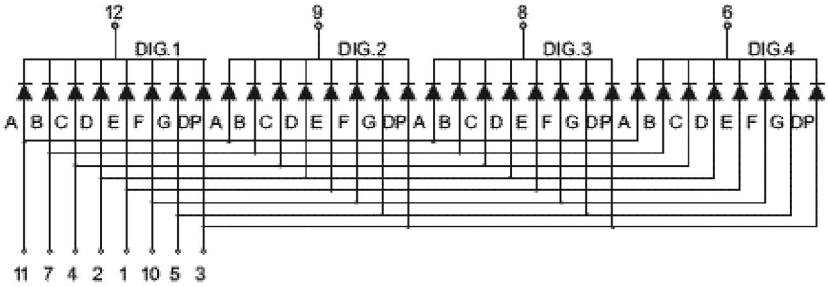
V elektronické praxi se potkáte nejčastěji se sériovými sběrnicemi SPI, I²C, RS-232 (někdy označované jako UART nebo jen Serial) a 1-Wire. Paralelní rozhraní mají z často používaných komponent snad jen některé displeje. Většina ostatní techniky – senzory, aktuátory, paměti, paměťové karty atd. – používají sériová rozhraní. Důvody jsou nejen ty výše zmíněné, ale třeba i to, že řídicí obvody – jednočipy, mikrokontroléry – mohou snadno pomocí dvou či tří vývodů připojit hned několik zařízení. Pro paralelní rozhraní by bylo potřeba vývodů osm, či spíš devět (osm datových plus jeden, který říká „teď něco udělej!“)

Tak, a teď vážně: Kde bychom tak mohli použít posuvný registr? Něco mě napadá…

22.1 Buzení displeje ze sedmisegmentovek

Když to vezmete kolem a kolem, je takový displej ze sedmisegmentovek velice náročná záležitost. Představte si čtyřmístný displej. Každá sedmisegmentovka má osm vývodů, to máte 32 vývodů, ani nemrknete…

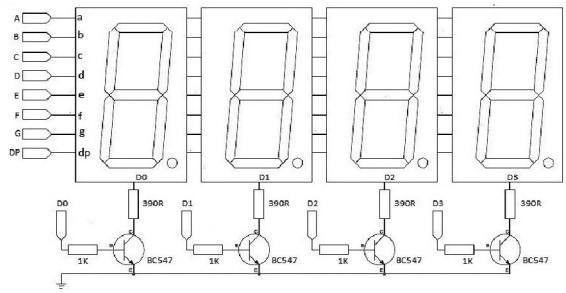
Ve skutečnosti se ale takto vícemístné sedmisegmentovky nezapojují. Zapojují se tak, že mají spojené všechny segmentové vstupy (a-g a tečku), tedy 8 vývodů, a pak jsou vyvedené společné anody / katody pro každou pozici. Nějak takto:



Pro N pozic stačí tedy 8 + N vývodů – pro čtyřmístný displej 12.

Na používání takového displeje musí být určitý trik. Je jasné, že nemůžeme zobrazovat všechno najednou – nemáme jak. Používá se proto řešení jiné: přivedeme na segmentové vstupy kombinaci pro první pozici, a aktivujeme společný vývod pro pozici 1. Necháme ji chvilku svítit, a pak totéž opakujeme pro druhou, třetí, … pozici. Říká se tomu *multiplexované řízení* displeje. Pokud takhle blikáme dostatečně rychle, lidské oko nevidí, že vždy svítí jen jedna sedmisegmentovka a zdá se mu, že svítí celý displej. „Dostatečně rychle" znamená alespoň padesátkrát za sekundu. K tomu se opět používají buď specializované obvody, nebo se to nechá na jednočipu, který má dostatečný výkon, aby se tímhle zabýval. Bohužel, takový displej sebere hodně vývodů (třeba těch 12), a když máte jednočip s dvaceti vývody, moc vám jich už nezbývá.

Technická poznámka: pro buzení takových displejů je vhodné použít tranzistorový budič. Pokud každý segment chce proud třeba 5 mA , tak vám při rozsvícené „osmičce" poteče společným vývodem 35 mA, a to je víc, než je většina jednočipů schopna a ochotna poskytnout.



Co dělat v situaci, kdy:

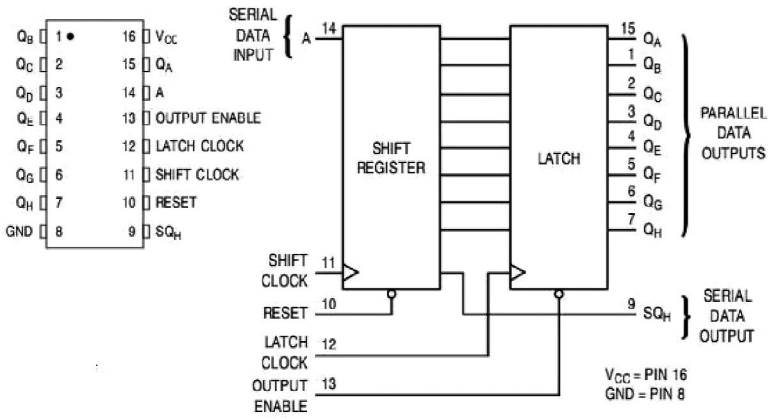
- blikání displeje, byť rychlé, je na závadu (třeba je snímané kamerou)
- nemáte sdružený displej, ale opravdu čtyři sedmisegmentovky

- chcete ušetřit počet nutných signálů?

22.2 Posuvný řadič SIPO 74HCT595

Jak už z mezititulku tipujete, půjde asi zas o nějaký skvělý integrovaný obvod. A přesně tak to je!

74595 je obvod, který se skládá z osmibitového posuvného registru a osmibitového klopného obvodu typu D.

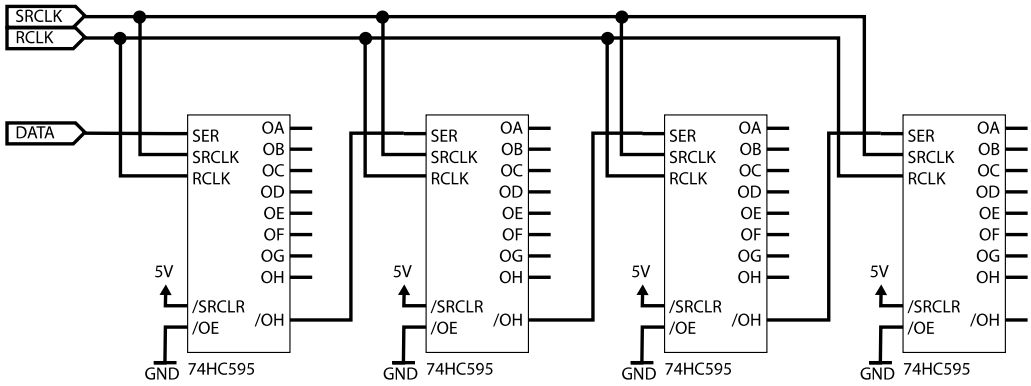


Všimněte si, že posuvný registr má sériový vstup A, sériový výstup SQh, hodinový vstup Shift Clock, též SRCLK (náběžná hrana posouvá informace v registru) a nulovací vstup /RESET. Díky tomu můžeme po dvou signálech (Serial In a Clock) poslat do posuvného registru osm bitů tak, jak potřebujeme. Náběžnou hranou na vstupu Load (Latch Clock, RCLK – každý výrobce značí vývody jinak, ale význam je stejný) přeneseme stav posuvného registru do klopných obvodů D. Ty budou tento stav pouštět na výstup a zároveň jej budou držet až do příští náběžné hrany na RCLK.

Obvod má ještě vstup /OE, kterým můžeme vývody odpojit, pokud je to potřeba.

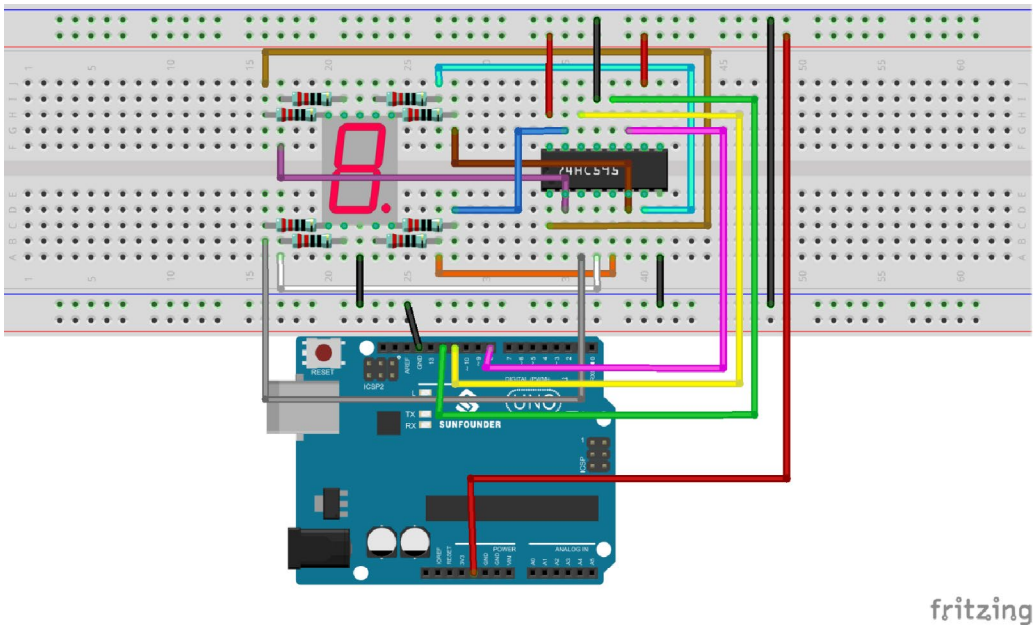
Díky tomu, že posuvný registr má i sériový výstup, můžeme těchto obvodů zapojit za sebe několik. Vstupy SRCLK a RCLK můžeme propojit navzájem, a výstup jednoho sériového registru zavedeme na sériový vstup druhého.

Pomocí tří vodičů (Serial In, SRCLK a RCLK) tak můžeme ovládat teoreticky neomezené množství registrů. Každý registr má k dispozici osm výstupů…



Ke každému registru můžeme (přes rezistory) připojit sedmisegmentovku. Výstupy mají budiče a jsou schopné dodávat 6 mA. Celý obvod je navíc poměrně rychlý a dokáže pracovat na frekvenci 25 MHz.

Pojďte si to zapojit. Jen obvod 74595 a sedmisegmentovku. Bude se to celé budit z Arduina. Aspoň se pocvičíte v práci se sériovými daty.



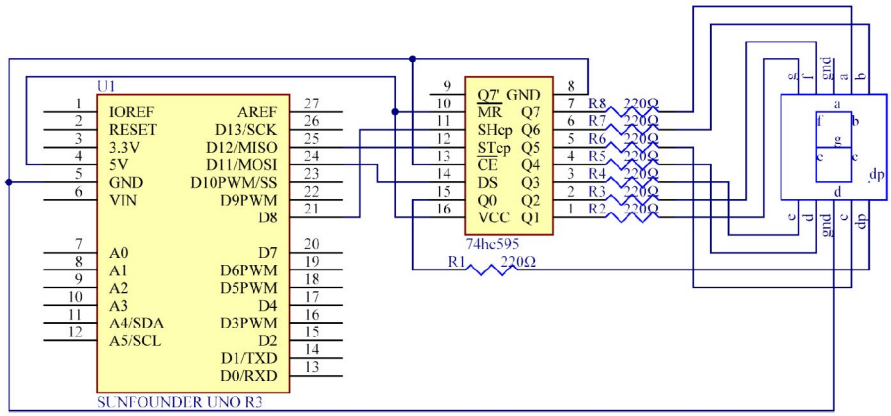


Schéma a zdrojový kód najdete na 🔗 https://eknh.cz/595

A teď pojďte, zapojte si ještě jednu sedmisegmentovku hned vedle té, co jste si zapojili teď. Použijte další obvod 74595, zase osm rezistorů, propojte sériový výstup prvního registru se vstupem druhého, spojte SRCLK a RCLK, upravte kód…

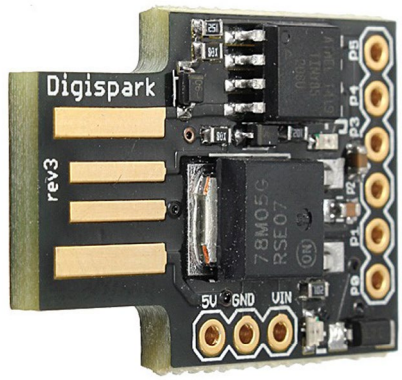
Pamatujte: pokaždé, když bojujete s malým počtem dostupných pinů, zvažte použití podobného řešení. Pokud budete potřebovat vícebitové řešení, popř. obsloužit i vstupy, zvažte použití obvodů MCP23xxx od Microchip, např. MCP23008 (8 bitů, I²C), MCP23S08 (8 bitů, SPI), MCP23017/MCP23S17 (16 bitů) atd.

23 Sériová komunikace

23 Sériová komunikace

Co jsem v předchozí kapitole nakousl, to teď rozvedu. Sériová komunikace je totiž velmi používaná věc a téměř všechny moderní elektronické komponenty používají právě tento způsob komunikace, kdy jsou vícebitové informace (většinou osmibitové) přenášeny po bitech postupně za sebou.

Sériová komunikace si proto vystačí s jedním, dvěma, třemi či čtyřmi vodiči pro data, plus jedním vodičem pro zem. Společná zem je podstatná věc pro spojování zařízení, bez ní nám jednak chybí společný vztažný bod pro měření napětí, jednak při spojení tečou úplně zbytečné proudy, které mohou přetěžovat vstupní obvody. Mimochodem, všimli jste si někdy, že například u konektorů USB jsou některé kontakty delší?



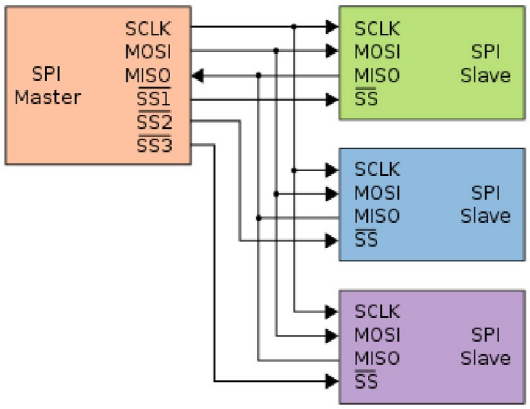
Důvod je prostý – delší vývody jsou napájení, takže při zasouvání se nejprve vodivě propojí napájecí obvody, a teprve potom se připojí datové signály.

Sériových rozhraní je několik druhů. Liší se od sebe nejen počtem vodičů, ale i schopnostmi a rychlostí.

23.1 Sériová sběrnice SPI

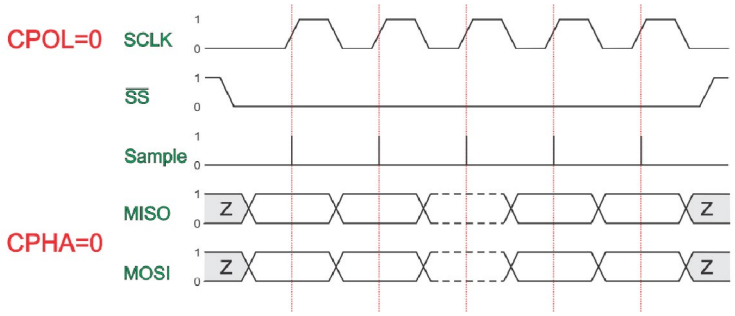
Asi nejjednodušší na pochopení je sběrnice SPI (z anglického Serial Peripheral Interface). Tímto rozhraním bývají vybavené různé paměti nebo některé senzory.

U rozhraní SPI je vždy jeden řídicí obvod (Master) a jeden či více periferních obvodů (Slave). Master řídí celou komunikaci a pomocí signálů /SlaveSelect určuje, se kterým periferním zařízením se právě pracuje.



Obecně vypadá komunikace tak, že master aktivuje signál /SlaveSelect pro ten obvod, s nímž chce komunikovat, a uvede jej do log. 0 (je to invertovaný signál).

Poté, co je obvod vybraný, začne Master na výstupu SCLK (Serial Clock) posílat hodinové pulsy, a zároveň na výstupu MOSI posílá data od nejnižšího bitu.



Jak na straně masteru, tak na straně slave zařízení jsou použity posuvné registry. Master pošle do zařízení požadované informace, nejčastěji příkaz a nějaké parametry, a jakmile skončí přenos, uvede vstup /SlaveSelect opět do neaktivního stavu.

Sběrnice SPI je plně duplexní, to znamená, že v ten samý čas mohou jít data jak ve směru MASTER -> SLAVE, tak i v opačném, tedy z periferie do řídicího zařízení. K tomu slouží druhý vodič, MISO.

Vodiče MISO a MOSI se mohou začátečníkům plést, proto si pamatujte, co to znamená: MISO je Master In, Slave Out, MOSI je Master Out, Slave In.

Stačí nám tedy tři komunikační vodiče – SCLK, MOSI a MISO, a k tomu jeden „výběrový“ signál pro každé zařízení.

Ale aby to nebylo tak jednoduché, tak si můžeme vybrat, jestli se bity čtou se vzestupnou nebo sestupnou hranou, a jestli jsou hodiny v klidovém stavu v log. 1, nebo v log. 0. Protokol se nastavuje většinou pomocí konfigurace SPI rozhraní (např. v procesoru). Zařízení mívají dané, jaký typ komunikace používají.

Konfigurační bit CPOL udává polaritu hodin (Clock Polarity). 0 znamená, že hodiny jsou v klidu ve stavu log. 0, CPOL = 1 znamená, že hodiny jsou invertované (ve stavu 0). Konfigurační informace CPHA udává, při které hraně se čtou informace (Clock Phase). Pokud je to 0, čte se informace při přechodu z neaktivního stavu hodin do aktivního (což je vzestupná hrana u CPOL 0, sestupná u CPOL 1). Pokud je CPHA rovno jedné, je to ta druhá hrana. Rozlišujeme tedy čtyři různé módy SPI.

SPI mode	CPOL	CPHA	Význam
0	0	0	Hodiny jsou v klidu v 0, vzorkuje se při přechodu 0->1
1	0	1	Hodiny jsou v klidu v 1, vzorkuje se při přechodu 1->0
2	1	0	Hodiny jsou v klidu v 0, vzorkuje se při přechodu 1->0
3	1	1	Hodiny jsou v klidu v 1, vzorkuje se při přechodu 0->1

Pokud budete mít štěstí, nebudete muset nic z toho řešit a vystačíte si s módem 0.

Některá zařízení pro sběrnici SPI neposílají žádná data, proto u nich nemusíme zapojovat MISO. Některá zase mají možnost vyvolat přerušení – tedy upozornit na to, že se něco stalo, a že by tomu měl master věnovat pozornost.

Takováto sběrnice může přenášet data velmi rychle, až v řádech megabajtů za sekundu. Samozřejmě jsme limitováni kvalitou vedení a jeho délkou, ale teoreticky lze dosáhnout velmi velkých rychlostí. Některá zařízení, převážně moderní sériové paměti, používají mód, kdy dokážou přenášet během jednoho hodinového pulsu dva nebo čtyři bity najednou (samosebou je potřeba víc vývodů MOSI/MISO). Takový přenos se označuje jako Dual SPI, popřípadě Quad SPI (QPI, nebo také SQI).

Sběrnici SPI používá celá řada periferií – především už zmíněné sériové paměti RAM / EEPROM / FLASH. Pomocí SPI lze pracovat s paměťovými kartami typu SD. Dále SPI používají některé ovladače displejů, nebo různé senzory. SPI definuje pouze to, jakým způsobem probíhá komunikace fyzicky, neříká, jaké informace se mají jak posílat. To je na obslužném software. Naštěstí to najdeme v datasheetu.

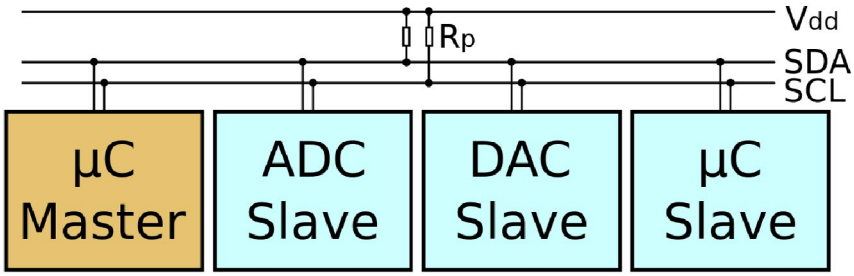
SPI je standard, který definovala Motorola na konci 80. let, ale používá se dodnes. Někteří výrobci v datasheetu uvádí, že jejich zařízení umí SPI, jiní jsou kryptičtější, a snad v obavách z případných sporů nazývají toto rozhraní jinak – třeba 4-wire Serial.

23.2 Sériová sběrnice I²C

I²C je z anglického *Inter-Integrated Circuit* a mělo by se to správně číst *I squared C*, nesprávně *I two C*. V našich luzích a hájích běžně uslyšíte „í-dva-cé“ („ájsquérdsí“ ani „í-na-druhou-cé“ se moc neuchytilo). Navíc je I²C ochranná známka společnosti Philips, která tento standard vyvinula, takže mnozí výrobci sice používají naprosto stejný typ sběrnice, se stejnými parametry a stejným ovládáním, ale cudně ji označují jinak, aby se vyhnuli použití chráněné značky. Kupříkladu Atmel označuje ve svých dokumentech tuto sběrnici jako TWI – Two Wire Interface.

Na první pohled vypadá I²C velmi podobně jako SPI, jen má míň vodičů. Kromě země si vystačí s pouhými dvěma vodiči, datovým SDA a hodinovým SCL.

Opět platí, že máme jeden „master“ obvod (může jich být i víc, ale nejčastěji je jeden) a k němu připojené periferní obvody. Každý periferní obvod má, na rozdíl od SPI, vlastní sedmibitovou adresu, tedy v rozsahu 0 až 127. Adresy nejsou úplně libovolné, většinou jsou pevně dané výrobcem – či alespoň jejich část. Pokud chceme zapojit dva obvody stejného typu naráz, musí podporovat možnost zapojit je na různé adresy. Jak vidno, sedm bitů není příliš mnoho, navíc některé kombinace jsou vyhrazené pro speciální použití, a tak se pomalu prosazuje novější standard s desetibitovými adresami.



Ve všech zařízeních jsou vývody SDA a SCL navrženy „s otevřeným kolektorem“. Už jsme se s tím setkali, vzpomeňte si: takový výstup je buď ve stavu 0, nebo ve stavu „odpojeno“. Výhodou je, že můžeme takové vývody spojovat dohromady, nevýhoda je, že potřebuje pull-up rezistor, aby mohla být na vedení za klidového stavu logická 1. Další výhoda takového zapojení je, že vývod může být zároveň výstup i vstup. Pokud funguje jako výstup, je buď 0, nebo „odpojen“. Pokud má fungovat jako vstup, nechává se „odpojen“, a obvod „naslouchá“, jestli je na vedení 0, nebo 1.

Komunikaci opět zahajuje master, a to tím, že na vývod SDA pošle logickou 0 (říkáme, že vývod „stáhne k zemi“), a poté (to je důležité!) stáhne i hodinovou linku SCL. Tím vytvoří takzvaný „startovní signál“ (Start Condition). Následně master začne vysílat hodiny, a zároveň nastavuje na vodič SDA data (postupně od nejvyššího bitu k nejnižšímu – zde je rozdíl proti SPI).

Přenos dat vždy začíná tím, že je přenesena sedmibitová adresa (A6 – A0) a bit R/W, který říká, zda bude následovat čtení ze zařízení (1), nebo zápis do zařízení (0). Během té doby všechna zařízení naslouchají a čtou adresu. Pokud je adresa jiná, než jejich, opět se odpojí a čekají na STOP signál (stop condition), kdy se při SCL=1 změní SDA z 0 na 1. Pokud je adresa stejná, zařízení se připraví na přenos dat.

Poté nechá master datovou sběrnici odpojenou, začne naslouchat a pošle ještě jeden hodinový puls. Zařízení, které rozpoznalo svou adresu, stáhne SDA k zemi, tedy vyšle log. 0, a tím potvrdí připravenost komunikovat (říkáme tomu puls ACK – z Acknowledge, neboli potvrzení). Master tento bit sleduje, a pokud přijme 0, ví, že je vše OK. Pokud ale během devátého hodinového pulsu najde na SDA hodnotu 1, znamená to, že žádné zařízení nerozpoznalo svou adresu.

Po této výměně následuje vlastní přenos dat, a to buď z masteru do periferie, nebo opačným směrem. V obou případech ale časování řídí master. Když je přeneseno vše, co přeneseno mělo být, uvolní master sběrnici tím, že nejprve „pustí“ SCL (a pullup rezistor jej vytáhne k log. 1), a poté uvolní i SDA. Tím všechna zařízení na sběrnici poznají, že nastal konec přenosu.

Sběrnice I²C umožňuje i některé zajímavé činnosti, například poslat desetibitovou adresu, pokud ji zařízení podporují. Master nemusí přenos ukončovat „stop stavem“, ale může znovu zahájit vysílání dalším startem a opětovným vysláním adresy. Existuje i možnost „natažení času“ – pokud je periferní zařízení příliš pomalé a trvá mu delší dobu, než má k dispozici data, udělá takový trik: samo stáhne hodinový signál k 0, čímž naznačí masteru, že má chvíli počkat. Když je pak připravené, tak opět hodinovou linku uvolní, master zjistí, že je volno, a pokračuje v přenosu.

Přenos dat probíhá po této sběrnici základní rychlostí 100 kHz, tedy 100 kbit / sec. Většina zařízení umí pracovat i s rychlostí 400 kHz. Zařízení standardu High Speed mohou komunikovat s frekvencí 3,4 MHz. Nejnovější ultra fast zařízení používají, stejně jako SPI, vedení bez pull-up rezistorů, ovšem komunikují pouze jednosměrně.

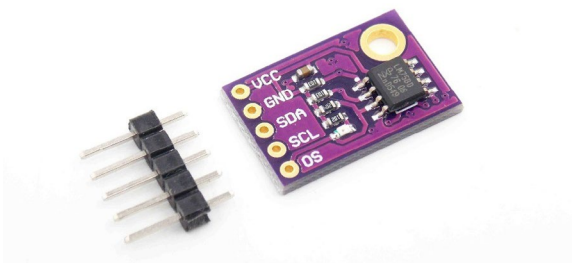
Vlastnost	**SPI**	**I²C**
Rychlost	Závisí na zařízení, teoreticky neomezená, prakticky omezená fyzikálními limity (délka vedení, jeho kvalita apod.)	100 kbit/s 400 kbit/s 3,4 Mbit/s (pouze nové obvody)

Počet vodičů	1 × hodinový signál 2 × datový signál 1 řídicí vodič pro každé zařízení Společná zem	1x hodinový signál 1x datový signál Společná zem
Počet připojených zařízení	Teoreticky neomezený, prakticky omezený potřebou mít pro každé zařízení jeden řídicí signál	Teoreticky neomezený, prakticky omezený adresováním zařízení. V případě připojení více obvodů stejného typu je třeba vyřešit, aby každý měl svou adresu. Pokud to obvod neumožňuje, nelze jich připojit víc.

23.3 Prakticky...

Pojďte si zase něco připojit! Jako první si připojíme k Arduinu nějaký senzor – třeba digitální teploměr LM75A. Používá sběrnici I²C. Nemusíme nějak složitě řešit přenosový protokol, protože naštěstí Arduino, respektive použitý jednočip ATmega, má specializované obvody, které se postarají o správný přenos dat, posílání start / stop signálů i hodin.

Zásadní otázka je: Kde LM75A sehnat? Trochu hloupé je, že tento obvod máme k dispozici pouze v pouzdrech pro povrchovou montáž. Ale naštěstí některé e-shopy nabízejí takzvaný „breakout board" (hledejte „LM75 Breakout"). Vlastně jde o miniaturní destičku, kde je připájený tento obvod, a jsou zde vývody vyvedené tak, aby je bylo možné připojit k nepájivému kontaktnímu poli či propojovacím vodičům.



Všimněte si popisků: VCC a GND je napájení, GND zem, tedy záporný pól, VCC kladný pól. SDA a SCL jsou signály sběrnice I²C. Na destičce jsou připojené i pullup rezistory mezi VCC a SDA / VCC a SCL. Což je dobré, nemusíte se o ně starat. Poslední vývod je OS. Co to je?

Podíváme se do datasheetu, kde najdeme nejen odpověď na tuto otázku (je to signál „Overtemperature Shutdown" – signál, který je aktivní, pokud teplota překročí nastavenou mez), ale i jiné důležité věci.

Například že obvod lze napájet pěti volty, že v rozsahu - 20 °C až + 100 °C je jeho přesnost +/- 2 °C, a že jeho sedmibitová adresa je „1001xxx". Přesněji 1 – 0 – 0 – 1 – A2 – A1 – A0. Hodnoty A2, A1 a A0 záleží na to, jako jsou zapojené tři vstupy se stejným označením (piny 7, 6 a 5). U většiny breakoutů budou připojené k 0, ale ještě si to ověřte u svého kousku. Pokud je tomu opravdu tak, je adresa 1001000, neboli 0x48 hexadecimálně.

Zároveň se dozvíme, jak do zařízení zapisovat a jak z něj číst. Tady je dobré poznamenat, že většina zařízení se sběrnicí I²C se tváří jako svého druhu „paměť" – jako by byly uvnitř rozdělené na paměťové buňky, ke kterým se přistupuje, čte se z nich a zapisuje do nich.

Obvod LM75 má čtyři registry, kterými se nastavuje jeho funkce, a ve kterých jsou připravená data ke čtení. Adresy jsou následující:

Adresa	Jméno	R/W	Funkce
0	Temp	Pouze pro čtení	Naměřená teplota (16 bitů)
1	Conf	Čtení i zápis	Konfigurace obvodu (8 bitů)
2	Thyst	Čtení i zápis	Hystereze pro teplotní poplach (Overtemperature Shutdown) (16 bitů)
3	Tos	Čtení i zápis	Teplota pro teplotní poplach (16 bitů)

Zápis dat probíhá tak, že master vyšle START, poté adresu obvodu (0x48), poté číslo registru, se kterým chce komunikovat, a pak už následují data pro zápis. Při čtení se postupuje o něco složitěji: master vyšle start, pak adresu obvodu 0x48, pak číslo registru, ze kterého chce číst, nato následuje opět start, adresa obvodu (ale s nastaveným příznakem „čtení"), a pak master naslouchá a obvod posílá data.

V Arduinu je vyvedená sběrnice I²C přímo na konektorech, buď nahoře u dat, nebo se používají vývody A4 (SDA) a A5 (SCL) u Arduina Uno. U Arduina Mega to jsou vývody 20 (SDA) a 21 (SCL). Zkontrolujte si, pokud máte jiné Arduino, jak je tomu u vás.

Propojte čtyřmi vodiči breakout modul s Arduinem – vývody GND, Vcc (na + 5 V), SDA a SCL.

Arduino má pro sběrnici I²C speciální knihovnu, která se jmenuje Wire. V dokumentaci naleznete přesný popis, zde jen odkážu: https://www.arduino.cc/en/reference/wire

Základní funkce jsou begin(), která připraví celý subsystém TWI. Komunikace začíná zavoláním

funkce beginTransmission(addr) – vlastně jde o vyslání signálu Start a adresy – a ukončuje se endTransmission(). Pokud požadujete data, používá se funkce requestFrom(addr).

Pro zápis se používá následující sekvence:

```
Wire.begin();
…
Wire.beginTransmission(adresa);
Wire.write(číslo registru);
Wire.write(data);
…
Wire.endTransmission();
```

Pro čtení se používá modifikovaná sekvence:

```
Wire.begin();
…
Wire.beginTransmission(adresa);
Wire.write(číslo registru);
Wire.endTransmission();
Wire.requestFrom(adresa, počet bajtů které budeme číst);
data = Wire.read();
…
```

Pro práci se senzorem LM75 najdete několik knihoven pro Arduino, ale doporučuju, abyste si zkusili nejprve jak vypadá samotná komunikace. Zkuste třeba přečíst dva bajty z registru 0, tedy aktuální teplotu. V datasheetu najdete, jak se posílá informace o teplotě, v jakých jednotkách a s jakou přesností.

Schéma a zdrojový kód najdete na ⌀ https://eknh.cz/lm75

23.4 EduShield a displej

EduShield obsahuje rovnou dvě zařízení, připojená na sběrnici I^2C. Jedno z nich jsou hodiny reálného času DS1307 – k nim se za chvíli dostaneme. Druhé zařízení je budič sedmisegmentového displeje. Jestli se ptáte, jaký to je typ, že byste si to našli v datasheetech, tak vás musím rovnou upozornit, že neuspějete – místo konkrétního obvodu je tam naprogramovaný *jednočip* ATtiny2313, který funguje jako I^2C slave. Poslouchá na adrese 0x27 a jediné, co s ním můžete dělat, je zapsat jeden byte do jedné z adres 0, 1, 2, 3.

Komunikační protokol je extrémně jednoduchý: pošlete dva bajty. První je adresa (0 – 3) která zároveň odpovídá pozici na displeji (0 vlevo, 3 vpravo). Druhý je bajt, který říká, co má na dané pozici svítit. Není použitý žádný kód, žádné sofistikované převody, jednotlivé bity zkrátka říkají, jaké segmenty mají svítit. Nejvyšší bit je desetinná tečka, nejnižší segment A. Tedy hodnota 0x00 celou pozici zhasne, hodnota 0x7F rozsvítí všech sedm segmentů („osmička"), hodnota 0xFF udělá totéž i s desetinnou tečkou, hodnota 0x06 rozsvítí segmenty B a C (a ty dohromady dají znak „jednička")…

Schéma a zdrojový kód najdete na ⌀ https://eknh.cz/edudis.

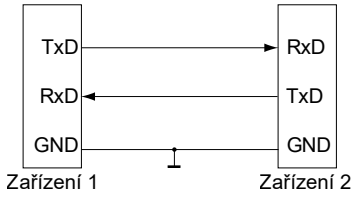
23.5 RS-232, UART, Serial…

Před mnoha lety jsem psal do jednoho časopisu článek, kde jsem zmínil tento standard pouze v souvislosti s formátem posílaných dat. Dostal jsem velmi rozhořčený mail od jednoho čtenáře, který mě plísnil za terminologickou nedbalost, protože RS-232 zahrnuje mnohem víc věcí, než jen formát dat, a že třeba takový hnusný novotvar, jako je „RS-232 TTL" je naprostý terminologický nesmysl. „Ať se ti pisálci, kteří to takto používají, podívají do normy, aby viděli, že jen matou lidi!"

Já vám tedy na úvod hlásím, že nic takového, jako „RS-232 TTL" de iure ani „de technická norma" neexistuje, ale *pisálci nám to drze ignorují a používají to*, a proto vás seznámím s existencí podobných pojmů a při té příležitosti zopakuju, že se můžete, až budete hledat něco na webu, dočíst leccos – a nebude to vždy podle norem.

RS-232 je starý standard sériového přenosu dat, používaný už v 60. letech (poslední varianta RS-232C pochází z roku 1969). Oproti výše zmíněným sběrnicím spolu mohou komunikovat vždy pouze dvě rovnocenná zařízení (žádné není master a komunikaci může zahájit libovolné z nich). Dříve se toto rozhraní hojně používalo i v počítačích, po roce 2000 ustoupilo výkonnějšímu USB. V průmyslu se s ním ale setkáte nadále, navíc stejný princip přenosu se používá dodnes, i když v modifikované variantě, i v číslicové technice. Třeba ve vašem smartfonu je zařízení, které se jmenuje GSM modem, a s ním si, věřte nebo ne, interně procesor povídá přesně tímto způsobem.

Principiálně šlo o obousměrnou komunikaci po dvou vodičích. Na každém zařízení byl vývod RxD (pro příjem, receive) a TxD (pro vysílání dat, transmit). Výstup TxD jednoho zařízení se propojuje se vstupem RxD druhého zařízení, a opačně. Třetím nutným vodičem byla, jako všude, zem (GND).



Kromě těchto datových vodičů se používaly i řídicí vodiče, pomocí nichž signalizovalo jedno zařízení druhému, že požaduje data, nebo že je připraveno vysílat data. Šlo o signály RTS (Request to Send – počítač oznamuje, že bude posílat data), CTS (Clear to Send – zařízení potvrzuje, že počítač může poslat data), DSR (Data Set Ready – zařízení je připraveno), DTR (Data Terminal Ready – počítač je připraven), RI (Ring Indicator) a DCD (Data Carrier Detected). Poslední dva se používaly především u telefonních modemů, kde signalizovaly příchozí spojení a navázání komunikace. Dnes se tyto signály používají převážně u některých zařízení v průmyslu; většina komunikace se odehrává jen po datových vodičích. Někdy jsou řídicí signály používané k úplně jiným účelům (například u Arduina je signál DTR použit ke vzdálenému RESETu).

Standard RS-232 definoval nejen to, jak má komunikace vypadat, ale i jakému napětí odpovídá jaká hodnota. Vězte, že pro datové signály platilo, že logická 0 představuje napětí + 3 až + 15 voltů proti zemi, logická 1 bylo napětí - 3 až - 15 voltů. Pro řídicí signály platily údaje přesně opačné.

U počítačů se používaly hodnoty + 12 V a - 12 V, především proto, že toto napětí bývá k dispozici. Firma Maxim vyráběla (a vyrábí) převodníky RS-232 na úrovně TTL (tedy 0 a 5 V), kde jsou zabudované nábojové pumpy a invertory, které si vyrábějí z 5 V napájecího napětí potřebné hodnoty 10 V a -10 V.

RS-232 může pracovat buď v synchronním, nebo v asynchronním módu. U synchronních se posílají i hodinové pulsy, u asynchronních, což bývá nejčastější případ, musí být obě zařízení nastavena tak, aby používala stejnou rychlost. Datový přenos pak vypadá tak, že vysílací zařízení pošle bit 0 (START bit), pak osm datových bitů od nejnižšího k nejvyššímu, a pak bit 1 (STOP). Linka je v klidu v log. 1.

Jenže nic není tak snadné a jednoduché. RS-232 může používat kromě osmi datových bitů i sedm. Může, a nemusí, používat paritní bit, který doplní počet jedniček v přeneseném bajtu na sudý počet. Nebo na lichý. Někdy se mohou poslat dva STOP bity (oba log. 1). A teď otázka za sto bodů: Jak přijímač pozná, jestli vysílač posílá paritu, jaká to je, kolik bitů se přenáší a kolik STOP bitů bude?

Schválně, tipnete si?

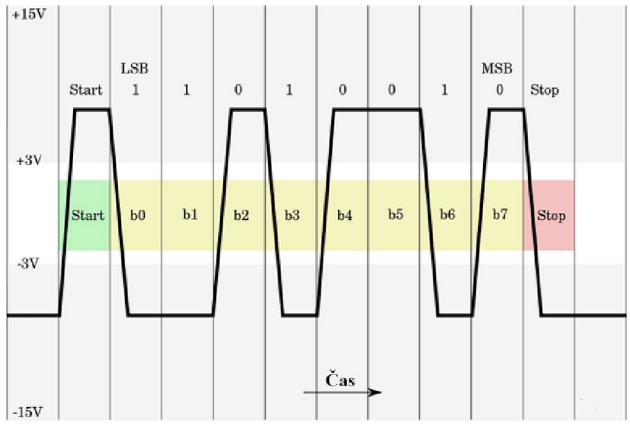
Správně: **Nijak**! Tyto parametry musí být nastavené předem. Proto třeba v dokumentaci zařízení naleznete, že komunikuje rychlostí 9600 Bd formátem 8-N-1. 8 znamená 8 datových bitů, N značí No parity (tedy neposílá se paritní bit) a 1 je STOP bit.

Rychlost udává, s jakou frekvencí se posílají jednotlivé bity. Teoreticky může být libovolná, ale v praxi se používají dodnes násobky čísla 150. Tedy 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 a 115200. Rychlost 150 znamená, že se přenášelo 150 bitů za sekundu, tedy maximální frekvence signálu (pokud se vysílalo „10101010“) odpovídala 150 Hz. Počet bitů

za sekundu (bps, bits per second) tedy odpovídá takzvané modulační rychlosti, která se udává v baudech (Bd). Mluvíme proto o přenosu rychlostí (např.) 9600 Bd.

Ovšem pozor, neznamená to, že se za sekundu přenese 9600/8 = 1200 bytů! Nezapomeňte, že na 8 datových bitů připadá jeden START bit (0) a alespoň jeden STOP bit (1), takže maximální přenosová rychlost je ve skutečnosti 9600/10 = 960 bytů za sekundu.

Dnes se v praxi používají rychlosti od 2400 výše. Nejčastěji to bývá 9600 u pomalých zařízení, kde nedává smysl vysoká přenosová rychlost. U rychlejších se setkáme nejčastěji s rychlostmi 19200 a 115200.



V mezititulku jsem zmínil i zkratku UART. Je to zkratka z Universal Asynchronous Receiver / Transmitter, což bylo označení součástek, které v počítačových systémech měly na starosti sériovou komunikaci po rozhraní RS-232. Dokázaly poslat a přijmout data, a procesor je mohl nakonfigurovat podle potřeby na různou délku dat, na různou paritu i na různou rychlost. Existovaly i součástky USART (Universal Synchronous / Asynchronous Receiver / Transmitter), které dokázaly používat i synchronní přenos, tj. s hodinovým signálem. Nám dodneška zůstalo toto označení, které se významově rozšířilo a označuje obecně výše popsaný způsob komunikace.

Uvnitř systémů by se nevyplácelo měnit 0 – 5 voltů na - 12 / + 12 voltů, a proto se používá standardní TTL konvence (0-5). Tomuto způsobu se někdy říká UART TTL, popřípadě nesprávně RS-232 TTL. Není to nic víc ani míň, než výše popsaný způsob komunikace, ovšem upravený tak, že 0 je 0 V a 1 je 5 V.

Kde se tato komunikace používá? No, na to, jak je stará, tak je stále při síle a hodně častá. Výhodou je její jednoduchost a nenáročnost. Hodně pomohlo i to, že USB definuje standard CDC, kde

je možné tento způsob přenosu dat „zabalit“ do USB. Pokud USB zařízení implementuje tento standard, vidí ho počítač jako „virtuální sériový port“.

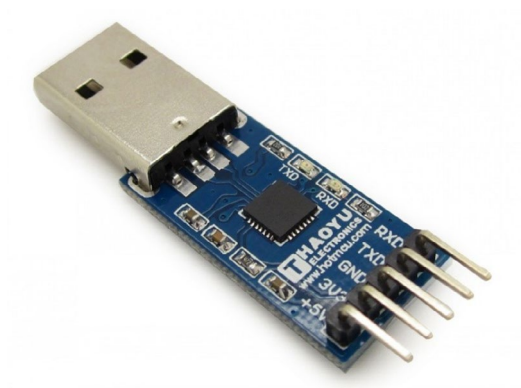
Známé jsou USB převodníky od firmy FTDI, např. FT232. Podobný obvod je použitý i v Arduinu, je připojen na piny 0 a 1 (všimněte si nápisů RX a TX) a slouží k programování Arduina přes Arduino IDE.

Po tomto sériovém rozhraní komunikují například GPS jednotky, komunikují takto GSM modemy, některé WiFi moduly, Ethernetová rozhraní nebo třeba moduly pro síť Sigfox používají právě tento protokol… Proto má většina mikrokontrolérů zabudovaný obvod UART. Ostatně pokaždé, když v Arduinu napíšete Serial.write, tak na pozadí proběhne výše popsaný proces, z jednočipu po vývodu TxD odejdou data, přijdou na vstup RxD u převodníku UART/USB a jsou poslána po USB do počítače.

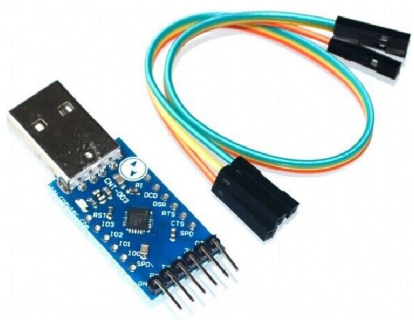
23.6 Převodník USB na sériové rozhraní

Mimochodem, když už jsem to nakousnul – takové převodníky USB na sériové rozhraní jsou velmi užitečná věc, a klidně si jeden, dva pořiďte. Samozřejmě, nejlepší jsou ty s originálními obvody FTDI a s vyvedenými signály DTR a RTS, ovšem jejich cena nebývá moc příznivá. Naštěstí existují levnější alternativy s čipy Silabs CP210x nebo Prolific PL2303.

V nejjednodušší podobě má takový převodník vyvedenou zem, napětí 5 V (z USB) a 3,3 V (stabilizovaných), a vývody TxD a RxD.



Já ale doporučuju dát si trochu víc práce a najít takový, který má vyvedené i další signály. Bude třeba o dolar dražší, ale využijete ho například pro programování některých zařízení, které potřebují signál RESET. Klíčová slova: *USB to TTL converter DTR*.



23.7 1-Wire

Co myslíte, šlo by ještě nějaké vodiče ušetřit? Určitě ano! Co třeba po jednom jediném vodiči?

Výrobce Dallas Semiconductors přesně takovou sběrnici navrhl, a nazval ji 1-Wire. Ve skutečnosti tedy používá vodiče dva (ten druhý je samozřejmě zem), ale data, data jdou jen po jednom drátu. Princip je podobný jako u sběrnice I^2C – každé zařízení má vlastní adresu a komunikace využívá principu otevřeného kolektoru – tedy pomocí rezistorů se komunikační linka udržuje ve stavu 1, a zařízení, popřípadě „master", ji stahuje k 0. Na rozdíl od I^2C se zde nepoužívá hodinový signál. Místo něj je použito přesné časování datových pulsů.

Dallas Semiconductors (DS) je snad jediný výrobce, který pro tuto sběrnici vyrábí periferní zařízení, a tak není tato sběrnice moc rozšířená. Mikrokontroléry pro ni nemají specializovaný komunikační obvod, komunikace se řeší softwarově, ale přesto je dobré se o této sběrnici zmínit, a to ze dvou důvodů.

Jednak proto, že pro tuto sběrnici existují zajímavá zařízení – kromě malých pamětí nebo teploměrů i speciální „iButtony", které se používají v zabezpečovacích zařízeních, pro kontrolu přístupu apod.



iButton může obsahovat buď unikátní číslo (ID), nebo například paměť pro data či pro šifrovací klíč.

Zajímavé na iButtonu je, že má pouhé dva vývody. Zem a data. Jak je to možné a kde je napájení? Je použitý takový fígl, a to je druhý důvod, proč se o 1-Wire zmínit. Fíglu se říká „parazitní napájení". Obvod využívá toho, že komunikační linka je tažena pomocí rezistoru k napájecímu napětí, a čerpá z ní napájecí napětí, které si ukládá do kondenzátoru. Když se pak se zařízením komunikuje, pracuje zařízení právě na tuto nastřádanou energii. Řídicí procesor může takovým zařízením pomoci tím, že je ve volných chvílích připojí natvrdo na napájecí napětí, tedy nikoli přes pull up rezistor, a tím zrychlí jejich nabíjení.

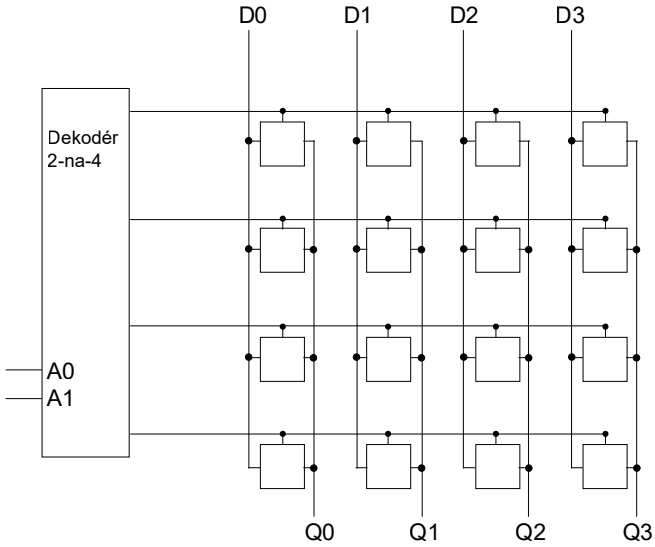
24 Paměti

24 Paměti

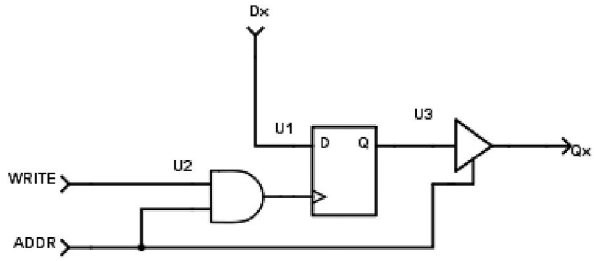
Od komunikačních technologií se zase na chvíli vrátíme k součástkám. Vzpomínáte na klopný obvod D? Klopný obvod D umožňuje dělat spoustu zajímavých věcí. Už jsme ho viděli jako dělič kmitočtu, viděli jsme ho jako posuvný registr i jako čítač, no a jeho další funkci si řekneme právě teď.

Víme, že si klopný obvod D pamatuje poslední zapsanou hodnotu a drží ji na svém výstupu až do chvíle, než vzestupná hrana na hodinovém vstupu nezmění vnitřní stav.

Představme si, že takových obvodů zapojíme šestnáct do matice 4 × 4. Nějak takhle:



Nahoře jsou čtyři datové vstupy D0-D3, dole jsou čtyři datové výstupy Q0-Q3, a vlevo je velký dekodér (demultiplexer) 2-na-4 (nebo 1-z-4, jak chcete). Každá buňka je zapojená takto:



Podle kombinace na vstupech A0, A1 je vybrán jeden z **adresních řádků** Y0, Y1, Y2, Y3. Tyto řádky jsou připojené na vstup ADDR každé buňky v řádku. Vstup ADDR řídí třístavový budič U3 na výstupu. V řádku, který má nastavený ADDR na 1, se tak výstup Q každého klopného obvodu připojí na odpovídající vývod Qx (výstupy v ostatních řádcích budou odpojené). Na vývodech Q0-Q3 tak bude k dispozici kombinace z příslušných klopných obvodů. Říkáme, že tím „čteme z paměti".

Pokud přivedeme na vstup WRITE puls, tak se na vybraném řádku (ADDR = 1) dostane přes hradlo AND na hodinový vstup registru D. Do tohoto obvodu se tedy zapíše stav na příslušném vstupu Dx. Říkáme, že tím „zapisujeme do paměti".

To, co jsme teď stvořili, se jmenuje „statická paměť RAM".

Dovolte mi malé opáčko: Paměti dělíme v zásadě na dva druhy: **RAM** (Random Access Memory, což je historický a nepřesný název, přesněji jde o RWM – Read Write Memory), tedy paměti, ze kterých lze číst a do kterých lze i zapisovat, a **ROM** (Read Only Memory), tedy paměti, ze kterých lze jen číst. Ta první zhruba odpovídá pracovní paměti ve vašem počítači, ta druhá svou funkcí připomíná třeba vylisovaná datová CD.

Paměti ROM udržují informaci i poté, co je obvod vypnutý, na rozdíl od (většiny) pamětí RAM. Paměti RAM obvykle o svůj obsah přijdou ve chvíli, kdy je odpojeno napájecí napětí.

Z klopných obvodů a dekodérů můžeme stvořit paměť RAM, která si pamatuje, co je do ní zapsáno, ale jen dokud je přivedené napájecí napětí. Jakmile se vypne a znovu zapne, bude stav náhodný.

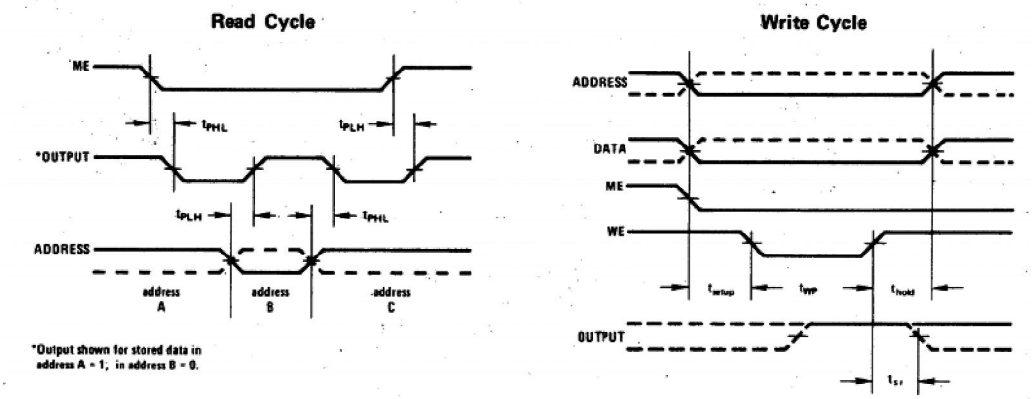
V řadě TTL se vyráběly obvody 7481 a 7484, které fungovaly plus mínus stejně, jako to výše popsané zapojení. Dokázaly uložit 16 bitů, ale moc se s nimi nesetkáte. Častěji se používal například obvod 7489.

24.1 7489 - 64 bitů RAM

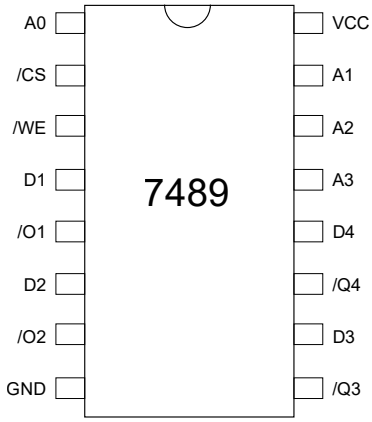
Šestnáctipinové pouzdro ukrývá statickou paměť RAM s kapacitou 64 bitů, uspořádanou do 16 slov po 4 bitech (říkáme, že je organizovaná „16 × 4"). To znamená, že bude mít čtyři datové vstupy, čtyři výstupy (podobně jako náš obvod na předchozích stránkách), čtyři adresové vstupy (A0-A3, dohromady dokážou adresovat 16 pozic), vstup /CS (Chip Select) a /WE (Write Enable). /CS povoluje práci s pamětí – pokud je neaktivní, tedy log. 1, pamatuje si paměť zapsaná data, ale má odpojené výstupy a ignoruje datové i adresové vstupy. Pokud je /CS aktivní (=0), je na výstupech hodnota, zapsaná v paměti na adrese, dané kombinací adresových vstupů A0-A3. *Tady je rozdíl proti našemu zapojení – výstupy jsou negované, takže pokud zapíšete na adresu 0 hodnotu 0011, přečtete 1100.*

Zápis probíhá tak, že na adresové vstupy přivedeme požadovanou adresu, na datových vstupech nastavíme požadovaná data, povolíme práci s pamětí nastavením /CS do aktivního stavu, a pak pošleme puls na vstup /WE – povolení zápisu.

Tyto informace jsem si nevymyslel ani mi je neprozradili staří mudrci při tajném rituálu – jsou napsané v datasheetu, dokonce graficky, aby to bylo úplně jasné. Pojďte se podívat (výrobce značí vstup CS jako ME – Memory Enable):



Vlevo vidíme čtení, vpravo zápis, graficky vyjádřené, co a jak se mění, a navíc najdete vyznačené i některé časy. Například jak dlouho to trvá od změny adresy do okamžiku, kdy jsou na výstupu data. U typu 7489 to trvá maximálně 60 nanosekund (t_{PHL}, t_{PLH}). Dozvíte se třeba to, za jakou dobu po CS (respektive ME) můžete posílat zapisovací puls (t_{SETUP} – 10 nanosekund min.) a jak dlouhý ten puls musí být (t_{WP} – 20 nanosekund).



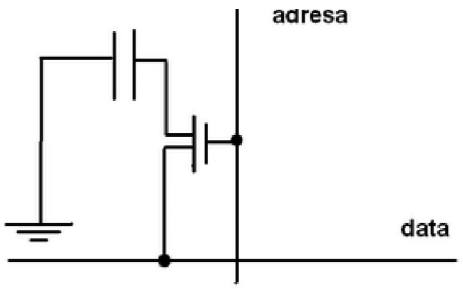
Obvod 7489 se používal v různých řadičích s mikroprogramem apod. V prvních osmibitových počítačích se využívala statická paměť 2114 s organizací 1024 × 4 – tedy dvě takové paměti vedle sebe měly kapacitu 1 kilobajt (1024 × 8bit). Byla podobná předchozím obvodům, ale měla jednu významnou změnu: Datové vývody byly zároveň vstupy i výstupy.

24.2 Dynamická RAM

V osmibitových počítačích byly běžné kapacity pamětí v řádech desítek kilobajtů. Dnes už není problém sehnat statické RAM, které mají kapacitu klidně 128 kB (128k × 8) nebo větší, ovšem na začátku 80. let to tak snadné nebylo.

Statické paměti, tedy takové, které ukládají data v klopných obvodech, jsou sice hodně rychlé, ale mají zásadní nevýhodu: jsou drahé. Na plochu čipu se vejde poměrně malá kapacita, takže tehdejší technologie byly omezené. Výrobci hledali způsob, jak kapacitu polovodičové paměti zvýšit, a nakonec přišli s dynamickou RAM.

Dynamická RAM nevyužívá klopný obvod, vystačí si místo toho s jediným tranzistorem MOSFET. Informace je uložena v kondenzátoru. Ten vzniká jako vedlejší efekt izolované řídicí elektrody tranzistoru a jeho kapacita je extrémně malá.



Výhoda je jasná: na plochu, kam se vejde jeden klopný obvod statické paměti RAM, se u dynamických vejde násobně víc paměťových buněk.

Nevýhody to má dvě. Zaprvé: při každém čtení se data vymažou. To řeší řídicí obvod paměti, který při čtení uloží data do vyrovnávací paměti a z ní je rovnou zapíše zpět. Zadruhé: náboj se samovolně vybíjí.

Druhý problém je podstatnější. Řeší se tak, že se neustále celá paměť postupně čte. Při čtení se zapisují data zpět, a tím se náboj obnovuje (Refresh).

Výsledkem tohoto tanečku je, že paměť potřebuje speciální obvody, které se starají o to, aby byla neustále čtena, a že v důsledku toho všeho je o něco pomalejší než statická RAM.

Asi by takovou věc nikdo nepoužíval, nebýt toho, že takové paměti jsou v porovnání cena / kapacita mnohem výhodnější než SRAM.

Běžné osmibitové počítače v 80. letech používaly právě dynamické RAM – díky jejich nízké ceně mohly být stroje vybavené klidně kapacitou 64 kB, i víc. Nižší rychlost nevadila, ony tehdejší procesory nebyly taky bůhvíjak rychlé, tak rychlost pamětí stačila.

V moderních počítačích se používají vylepšené paměti DRAM – SDRAM a DDR SDRAM, které mají gigabitové kapacity a většinou mívají i obvod automatického obnovování (autorefresh). Problém s rychlostí se řeší pomocí kombinace DRAM a rychlé statické RAM jako cache.

24.3 ROM, PROM a další

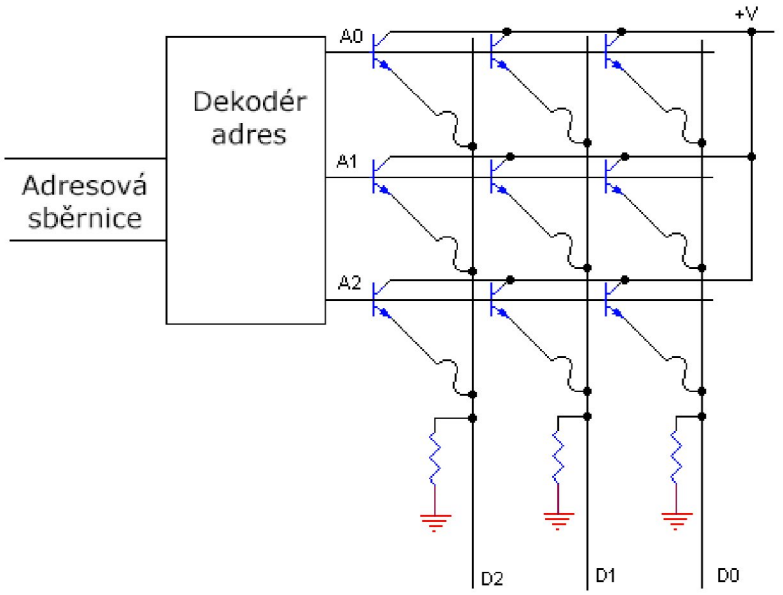
Občas je zapotřebí mít paměť, která udržuje nějaká pevně daná data. Může to být tabulka konstant, může to být i nějaký napevno zadaný program. Takové paměti se říká ROM, z anglického Read Only Memory, neboli *paměť pouze pro čtení*.

Úplně nejjednodušší paměť ROM dostanete, když v tom výše uvedeném schématu nahradíte klopné obvody, které si pamatují data, prostými propojkami, buď na log. 1, nebo na log. 0. Zbytek, tedy adresový dekodér a budiče výstupu, zůstane stejný. Signál pro zápis a datové vstupy ale budou zbytečné.

Logická otázka je: **Jak se tam ta data zapíšou?** Ne, vážně: kde se tam vezmou ty propojky? Odpověď zní: Udělá je tam výrobce při výrobě. Pokud si takový obvod chcete objednat, musíte dát výrobci data, jaká tam má nahrát. Samozřejmě se s tím výrobce nebude mazat, když po něm budete chtít dva obvody. Ale pokud jich budete chtít desetitisícové série, jistě se dohodnete.

Proto se paměti ROM moc neprodávaly – každý chtěl jiná data atd. Snad jediná výjimka byly paměti ROM, které obsahovaly ASCII znaky v matici 5 × 7, a používaly se v alfanumerických displejích.

Později výrobci nabídli paměti, které nazývali **PROM – Programmable ROM**. Ty obsahovaly jemné drátové propojky, které bylo možné určitým postupem a za použití vyššího napětí (většinou 12 voltů) přepálit, a tím naprogramovat jejich obsah. Jednou naprogramovaná paměť PROM už tedy nešla smazat a přeprogramovat znovu. (Funkce je tedy obdobná, jako u zapisovatelných CD a někdy se označuje jako paměť WORM – Write Once, Read Many.)

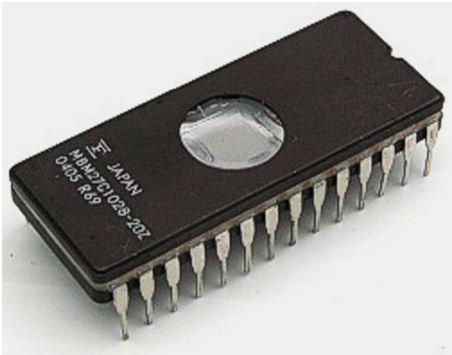


Paměť PROM byla velmi populární nejen pro ukládání řídicích programů, ale i pro vytváření logických kombinačních obvodů. Nevěříte? Sledujte!

Velmi populární paměť PROM měla označení 74188 a byla organizovaná jako 32 × 8 (32 byte). Má pět adresových vstupů a osm adresových výstupů. Představte si, že jste potřebovali navrhnout složitější logickou funkci. Buď jste mohli kombinovat hradla, anebo – v případě, že kombinační obvod využíval max. 5 vstupů a max. 8 výstupů – jste vzali tuto paměť, a prostě jste všechny možné kombinace do takové paměti zapsali, „vypálili“, a pak jste ji použili jako náhradu klidně desítky hradel.

Tato technika se používala až do doby, než přišly specializované obvody – programovatelná logická pole. Ty umí o něco sofistikovanější operace, můžete je i přeprogramovat (typ GAL), a postupně přišly velmi komplikované obvody typu CPLD a FPGA – v nich můžete vytvořit velmi složité obvody, doslova celé systémy na čipu (SoC – System on a Chip).

PROM tedy koupíte prázdnou a naprogramujete si ji sami. Bohužel jen jednou. Jakmile je naprogramovaná, už s ní nehnete, protože jednou přepálenou propojku už nescelíte. Tuto nevýhodu odstranily až paměti EPROM (Erasable PROM), což jsou paměti, které bylo možno nejen naprogramovat, jako PROM, ale poté i smazat pomocí ultrafialového záření a naprogramovat znovu. K tomu, aby bylo možné tyto paměti mazat, byly vybaveny skleněným okénkem nad čipem.



Po pamětech EPROM přišly **EEPROM**, tedy „**Electrically Erasable PROM**“. Technicky obdoba EPROM, ale obsah nebylo nutné mazat ultrafialovým světlem. Pomocí speciální sekvence se nastaví, jaká oblast má být smazána, a obsah se smaže přímo za chodu zařízení, není potřeba paměti vyjímat a mazat pod UV lampou.

Další generaci po EEPROM známe jako **FLASH**. Technologie FLASH dále vylepšuje EEPROM – jsou rychlejší, mají menší spotřebu, vyšší kapacitu, ale nevýhodou je, že po určitém počtu zápisů, řádově nižším než u EEPROM, ztrácí svou funkci. Navíc nelze zapisovat úplně libovolně; při zápisu je vždy smazán celistvý blok paměti.

Nevýhodou mazatelných PROM (EPROM, EEPROM, FLASH) je i to, že časem zapomínají. Ten čas se měří sice na roky a desítky let, ale i tak – když dnes dostanete do ruky třeba starý počítač z 80. let minulého století, ve kterém byly použity méně kvalitní paměti EPROM, může se stát, že jejich obsah už bude poškozený. Za nějaký čas to čeká i dnešní FLASH.

24.4 To nejlepší z obou světů

Máme tedy řadu pamětí ROM, do nichž lze data s určitými omezeními zapisovat, jako do paměti RAM. Jejich nevýhoda je, že jsou pomalejší než RAM. Z druhé strany máme paměti RAM, které jsou rychlé, ale zase po výpadku napájení zapomenou uložená data (jsou volatilní). Z tohoto směru zase přišla snaha výrobců udržet výhody RAM a nějak zajistit, aby výpadek napájení nesmazal zapsaná data, tedy vytvořit takzvanou **nonvolatilní RAM (NVRAM)**. Nejjednodušší způsob je zajistit, aby napájení nikdy nevypadlo – třeba výrobce pamětí Dallas Semiconductors integroval do pouzdra i malou baterii, která fungovala jako záložní. Nevýhoda je jasná – každá baterie se po čase vybije, takže ani tyto paměti neudrží svůj obsah výrazně dlouho.

V posledních letech se na trh dostaly ve větší míře paměti, které využívají pro uložení dat namísto elektrického náboje jiný princip. Paměti **FRAM (Ferroelectric RAM)** využívají ferroelektrickou vrstvu (feroelektrikum lze, zjednodušeně řečeno, přeměnit na permanentní magnet pomocí elektrického pole). FRAM využívají této vlastnosti a ukládají data tím, že mění polaritu takových látek. Při čtení se nastavuje buňka paměti na 0. Pokud už předtím byla 0, neděje se nic, pokud předtím byla 1, vznikne krátký puls, který logika vyhodnotí. Další paměti na podobném principu jsou **MRAM (Magnetoresistive RAM)**, kde se při čtení využívá měření elektrického odporu.

Tyto paměti jsou prozatím „to nejlepší z obou světů". Životnost dat v nich uložených je vyšší než u FLASH a EEPROM, udržují si data i po vypnutí napájecího napětí, zároveň jsou rychlé jako RAM, jsou odolnější vůči radiaci a lze je přepisovat téměř donekonečna (výrobce Texas Instruments udává u svých pamětí řádově 100-1000 bilionů zápisových cyklů). Na druhou stranu mají zatím stále menší kapacitu než FLASH (míněno „na trhu dostupné čipy") a jsou dražší.

Když si shrneme výhody a nevýhody jednotlivých typů:

FLASH: levné, nonvolatilní (pamatují si i bez napájení), zápis poměrně pomalý, čtení rychlé, nízká spotřeba, ale velmi omezený počet záznamových cyklů (typicky 10 000 – pracuje se ale na vylepšení technologie)

EEPROM: levné, nonvolatilní, zápis pomalejší než u FLASH, čtení rychlé, spotřeba vyšší než FLASH, vyšší počet záznamových cyklů (okolo 100 000).

RAM: levné, volatilní (potřebují napájení, aby si udržely data), zápis i čtení velmi rychlé, spotřeba nižší než u FLASH, počet zápisů neomezený.

MRAM, FRAM: dražší, nonvolatilní, zápis zhruba stejně rychlý jako u RAM, čtení také, spotřeba se pohybuje mezi RAM a FLASH, počet záznamových cyklů v řádech stovek bilionů.

Praktický tip: Co tedy použít, pokud ve svém zařízení potřebujete ukládat někam třeba naměřená data? Nejlevnější řešení s největší kapacitou nabízí FLASH (tuto technologii využívá třeba SD karta). Na druhou stranu musíte počítat s možnými chybami a ošetřit je, navíc při častějším přepisování dat paměť brzy odejde. Pokud potřebujete často zapisovat a mazat, zvažte použití EEPROM. Nevýhodou je zase vysoká spotřeba a pomalý zápis, takže u zařízení, která chrlí rychle data, budete stejně potřebovat nějakou verzi rychlé paměti RAM, do níž si uložíte data, a pak je najednou zapíšete do trvalejší paměti. Pokud potřebujete často a rychle zapisovat a mít nízkou spotřebu, využijte paměť RAM se záložním napájením. Riziko je, že po čase se i záložní akumulátor vybije a vy na to přijdete třeba až ve chvíli, kdy paměť zapomene data. Pokud potřebujete spojit všechny výhody, tj. nízkou spotřebu, vysokou rychlost zápisu a čtení a dlouhodobé uchování dat bez napájení, zvolte FRAM nebo MRAM. Nevýhodou je cena a malá dostupná kapacita těchto pamětí.

24.5 Několik tipů k pamětem

Pokud budete používat polovodičové paměti, hodí se vám pár tipů.

Tak například: ve světě pamětí a podobných specializovaných obvodů už většinou opustíme naši oblíbenou řadu 74xx a obvody TTL. Každý výrobce si vyrábí vlastní obvody v různých konfiguracích. Pro svoje konstrukce pravděpodobně nebudete používat třeba SDRAM a podobné ultramoderní kousky. S největší pravděpodobností sáhnete po sériových pamětech, které si představíme za chvilku. Ale kdybyste náhodou někdy stavěli něco se staršími procesory, tak si pamatujte:

PROM se dají stále sehnat, možná by šlo jimi nahradit složitější kombinační obvody, ale jednou naprogramovaná PROM se už moc opravit nedá. Jednodušší a lepší bude použít obvody GAL.

EPROM jsou už hodně výběhové paměti. Můžete je sehnat ve výprodejích, ale počítejte s tím, že nemusí fungovat, a taky s tím, že pro jejich mazání potřebujete ultrafialovou výbojku.

EEPROM jsou lepší volba, už jen proto, že se stále vyrábějí. Pro amatérské konstrukce jsou vhodné typy 28C64 (8k × 8), 28C256 (32k × 8) a 28C010 (128k × 8). Všechny tyto typy se vyrábějí i v pouzdrech DIP.

FLASH jsou vyráběny v řadě 39F – např. 39F010 (39SF010 – S označuje standardní napájení 5 V, L nízké 3,3 V) s kapacitou 128k × 8, 39F020 a 39F040 (256k × 8 a 512k × 8).

Paměti SRAM si výrobci značí ledasjak, ale vždy lze najít alternativu. Jen je potřeba spíš sledovat parametry, než hledat nějakou logiku v číslování. Oblíbené paměti jsou např. 6264 (8k × 8), 62256 (32k × 8) a pak 128k × 8 (např. AS6C1008), 256k × 8 (AS6C2008) a 512k × 8 (AS6C4008)

Ve starých PC AT se daly najít rychlé cache paměti SRAM, většinou organizované jako 32k × 8 apod. Jejich výhodou byla velká rychlost, nevýhodou velká spotřeba.

Zajímavé jsou takzvané „dual port“ paměti. Mají dvě sady adresních, datových i řídicích vývodů, takže se tváří jako dvě paměti, ovšem sdílejí společnou paměťovou strukturu. Používají se například jako video RAM – z jedné strany k nim může přistupovat procesor a zapisovat, a z druhé strany může nezávisle videoprocesor číst a zobrazovat data.

24.6 Jak se zapisuje do EEPROM či FLASH?

Paměti EEPROM (a také FLASH) jsou „elektricky mazatelné a přepisovatelné“. Jak se to dělá?

Paměť EEPROM má, podobně jako RAM, signál /WE – Write Enable, který slouží k zápisu dat. Můžete zapsat až 64 bajtů naráz. Paměť si je uloží do vnitřních registrů, a pak je začne zapisovat do paměťových buněk. Zápis je poměrně pomalý, může trvat podle typu paměti až 10 milisekund. Když budeme zapisovat vždy plný počet 64 bajtů naráz, a každý zápis bude trvat těch maximálních 10 ms, zapíšeme za jednu sekundu něco přes jeden kilobajt. U osmi kilobytů (28C64) to bude ještě snesitelné, ale u 128k paměti (28C010) budeme zapisovat přes 10 sekund (tato paměť dokáže zapsat v jednom cyklu až 128 bajtů).

Na druhou stranu dlouhá zápisová doba nevadí, protože paměť by neměla být často přepisována. Ostatně výrobce uvádí zaručený počet přepisů 100 000 – pokud ji přepíšete víckrát, není zaručeno, že ještě bude fungovat.

Ovšem představte si, co by se stalo, kdyby nějaký program „zdivočel“ a začal přepisovat paměť EEPROM. To by nebylo vůbec hezké. Proto mají EEPROM možnost ochrany paměti, a to jak hardwarovou, tak softwarovou. Pomocí speciální posloupnosti zápisů, která je u každé paměti jiná, se paměť „zablokuje“ nebo „odblokuje“ pro zápis.

Příklad: U EEPROM AT28C64B (8k × 8) se softwarová ochrana zapne takto:

1. zapíšete 0xAA na adresu 0x1555

2. zapíšete 0x55 na adresu 0x0AAA

3. zapíšete 0xA0 na adresu 0x1555

Poté máte možnost zapsat až 64 bajtů, a poté je paměť „uzamčena“. Víc podrobností najdete, jak jinak, v datasheetu.

U pamětí FLASH je situace obdobná, ovšem zápis je výrazně rychlejší. Paměť FLASH je typicky rozdělena do sektorů, např. po 4 kB (vezměme si jako příklad paměti 39F010 / 39F020, 39F040), a vždy před tím, než do daného sektoru začnete zapisovat, musíte si jej předem smazat. Smazání sektoru zabere zhruba 18 milisekund, smazání celé paměti zabere 70 ms.

Jak se takové smazání zařídí? Samozřejmě vás asi nepřekvapí, že se zapisují nějaké dané hodnoty na určité adresy. Pokud takové zápisy přijdou v přesně daném pořadí, spustí se daná operace. Například smazání celé paměti spustí šestikroková sekvence zápisů:

1. 0xAA na 0x5555

2. 0x55 na 0x2AAA

3. 0x80 na 0x5555
4. 0xAA na 0x5555
5. 0x55 na 0x2AAA
6. 0x10 na 0x5555

FLASH mívají ještě menší počet cyklů smazání/zápis než EEPROM. Proto je potřeba u aplikací, které intenzivně zapisují, s tímto faktorem počítat a přizpůsobit mu styl zápisu, aby se rovnoměrně využívaly všechny buňky (např. FAT nebude moc vhodný způsob, protože neúměrně často přepisuje první sektory). U některých aplikací bude lépe z téhož důvodu sáhnout po NVRAM – tedy po takových pamětech, které jsou typu SRAM a jsou zálohované baterií.

24.7 Sériové paměti

Až dosud jsem hovořil o klasických pamětech RAM, ROM atd. Takové paměti mají N adresových vstupů A0-An, M datových vývodů D0-Dm, kapacitu 2^N adres krát 2^M bitů, a pro větší kapacitu logicky potřebujete větší pouzdro s větším počtem vývodů. Tyto paměti mají totiž *paralelní rozhraní*. Jestli si pamatujete, co jsme si říkali v kapitole o rozhraních, tak totéž v bledě modrém platí i o těchto pamětech: nesnesou příliš dlouhé datové/adresové vedení, rychlost přenosu dat je teoreticky neomezená, a lze je připojovat snadno přímo na sběrnici procesoru.

To je také mimochodem odpověď na to, proč mají paměťové moduly do PC tolik pinů – kvůli rychlosti používají paralelní přenos, takže třeba moduly DDR3 DIMM používají 240 vývodů…

Pro malé systémy s jednočipovými řadiči se jako vhodná alternativa ukázaly sériové paměti. Technicky jde o stejnou paměť RAM, EEPROM, FLASH, ale používá sériové rozhraní, takže se vejde klidně do pouzdra s osmi vývody.

Dovolte tabulku:

Rozhraní / Typ	**RAM**	**EEPROM**	**FLASH**
SPI	23xx	25xx	25Fxx
I^2C	47xx (SRAM+EEPROM)	24xx	
Microwire		93xx	

Shrnul jsem sériové paměti podle typu a rozhraní. Třetí do sbírky, Microwire, je vlastně podmnožina SPI (umožňuje pouze SPI Mode 0 – viz popis SPI).

Asi největší portfolio těchto pamětí má v době psaní knihy společnost Microchip. Podržím se tedy jejich rozdělení a značení.

Sériové paměti se značí dvojčíslím, které udává typ paměti a rozhraní, viz tabulka, pak jedním či více písmeny, která udávají podtyp, popř. napájecí napětí, a opět číslem, které udává kapacitu. Kapacita se nejčastěji udává v kilo*bitech*, takže si nezapomeňte číslo podělit osmi, abyste dostali kapacitu v *kilobajtech*.

Písmena jsou nejčastěji L, C nebo LC (standardní paměti s napájením 2,5 – 5 V), AA (nízkonapěťové 1,8 – 5 V), A (s napětím 1,7 – 2,2 V). U pamětí pro I^2C se setkáte i s písmenem F, které znamená, že paměť umí pracovat i s přenosovou frekvencí 1 MHz (bez F používají většinou 400 kHz).

Čísla udávající kapacitu mají tento význam (u EEPROM a SRAM):

Kapacita (kilobit)	**Kapacita (kilobajt)**	**Kód**
1	0,125 (128 byte)	xx01
2	0,25 (256 byte)	xx02
4	0,5 (512 byte)	xx04
8	1	xx08
16	2	xx16
32	4	xx32
64	8	xx64
128	16	xx128
256	32	xx256
512	64	xx512
1024	128	xx1024,xx1025,xx1026,xxM01

U sériových FLASH se pohybujeme v trochu jiných dimenzích:

Kapacita (kilobit)	**Kapacita (kilobajt)**	**Kódh**
512	64	xxF512
1024	128	xxF010
2048	256	xxF020
4096	512	xxF040
8196	1024 (1 MB)	xxF080

16384	2048 (2 MB)	xxF016
32768	4096 (4 MB)	xxF032
65536	8192 (8 MB)	xxF064

Paměti s větší kapacitou a rozhraním SPI mají většinou i možnost využít dvou nebo čtyř datových vodičů najednou (SDI, SQI) – kvůli zrychlení přenosu.

Sériové paměti jsou zároveň poměrně levné, podle typu a kapacity okolo 50-90 Kč. Díky pouzdru s osmi vývody je většinou nebývá problém sehnat i v provedení DIP, tedy naprosto ideální pro amatérské použití.

Zásadní nevýhoda je, že to nejsou „jednoduché paměti", kam přivedete adresu po sběrnici a dostanete „v reálném čase" data. Vždy musíte udělat několik kroků, a i v tom nejjednodušším případě čtení dat musíte poslat do paměti požadovanou adresu, a pak přečíst data. Vhodné zařízení pro komunikaci je tedy mikrokontrolér…

Ale dosti planých řečí, pojďme si to zkusit zapojit.

25 Sériová paměť prakticky

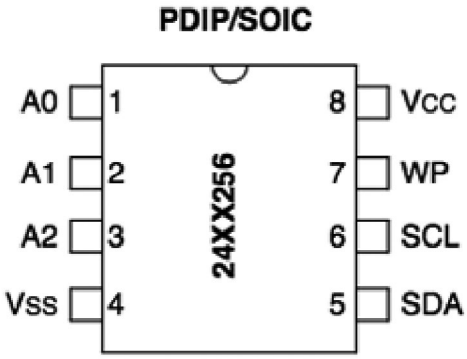
25 Sériová paměť prakticky

Pojďme si připojit sériovou paměť a *něco* s ní dělat. Co? Pojďme si třeba ukládat naměřená data do sériové paměti EEPROM, připojené přes sběrnici I²C.

Arduino Uno má samozřejmě vyvedené signály SDA a SCL pro sběrnici I²C. Překladač pro Arduino navíc nabízí knihovnu Wire, která právě se sběrnicí I²C pracuje. Navíc umí zapnout interní pull-up rezistory, takže není potřeba připojovat ehterní. Připojení sériové paměti je tak otázka jen několika propojovacích vodičů.

Použijte paměť z řady 24LC – třeba 24LC01, ale klidně i jakoukoli jinou s větší kapacitou. Budeme si ukazovat princip, ne konkrétní řešení.

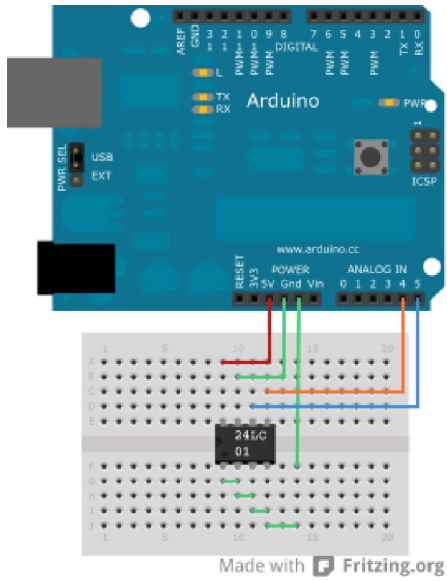
Paměti 24LCxx mají následující zapojení vývodů:



Vývody 1, 2 a 3 nastavují adresu. Už jsem se zmiňoval, že zařízení na sběrnici I²C mají svou sedmibitovou adresu, a že u některých ji lze změnit. Sériové paměti EEPROM mívají jako základní adresu 0x50, a tři nejnižší bity lze nastavit pomocí vstupů A0-A2. Konkrétní čip tedy může být nastaven tak, aby se hlásil na jedné ze sedmi adres 0x50 – 0x57. Pokud máte na jedné sběrnici jen jednu paměť, tak připojte všechny vstupy A na zem, paměť tak bude mít adresu 0x50.

Na zem připojte i vývod 7 – WP (Write protect). Pokud je připojen k zemi, může se do paměti zapisovat, pokud je spojen s napájecím napětím, je zápis zakázán.

Teď zbývá jen připojit napájecí napětí a signály pro I²C, tedy SDA a SCL. Na konci knihy, v přílohách, najdete takzvaný „pinout“, neboli rozložení vývodů Arduina včetně speciálních funkcí. A tam zjistíte, že SCL je vyveden na pin A5 a SDA na pin A4. propojení pak vypadá nějak takto:



Pro zápis do této paměti stačí poslat po sběrnici I²C nejprve adresu obvodu (tedy výše zmíněných 0x50), poté adresu v paměti (dva bajty, nejprve vyšší), a nakonec požadovaný bajt k zápisu. Nezapomeňte po zápisu počkat dostatečně dlouhou dobu – EEPROM zapisují pomalu.

```
void writeEEPROM(unsigned int address, uint8_t data)
{
    Wire.beginTransmission(0x50);
    Wire.send((int)(address >> 8)); // MSB
    Wire.send((int)(address & 0xFF)); // LSB
    Wire.send(data);
    Wire.endTransmission();
    delay(5);
}
```

Při čtení pošlete adresu stejně jako při zápisu, ale pak namísto poslání bajtu k zápisu jeden bajt přečtěte:

```
uint8_t readEEPROM(unsigned int address)
{
    uint8_t data = 0xFF;
    Wire.beginTransmission(0x50);
    Wire.send((int)(address >> 8));    // MSB
```

```
    Wire.send((int)(address & 0xFF)); // LSB
    Wire.endTransmission();
    Wire.requestFrom(0x50,1);
    if (Wire.available()) data = Wire.receive();
    return data;
}
```

Schéma a zdrojový kód najdete na ⌀ https://eknh.cz/eep

26 Hodiny reálneho času

26 Hodiny reálného času

V EduShieldu je použitý speciální druh paměti DS1307 – má pouhých 64 bajtů, a navíc se v prvních osmi bajtech údaje mění… Což takhle zní jako pěkná hloupost, že? Ve skutečnosti je to tak, že v prvních osmi bajtech je uložena informace o aktuálním čase a datu…

Obvod DS1307 je označován jako RTC – Real Time Clock (hodiny reálného času). Obvod se připojuje k zařízení opět pomocí sběrnice I^2C a funguje úplně stejně jako výše zmíněná paměť – nabízí 64 adres pro uložení dat. Ovšem prvních osm bajtů má speciální význam:

Adresa	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	
00	CH	Desítky sekund			Jednotky sekund				00–59
01	0	Desítky minut			Jednotky minut				00–59
02	0	12/24	Desítky nebo AM/PM	Desítky hodin	Jednotky hodin				
03	0	0	0	0	0	Den v týdnu			1–7
04	0	Desítky dne v měsíci			Jednotky dne v měsíci				01–31
05	0	0	0	Desítky měsíce	Jednotky měsíce				01–12
06	Desetiletí				Roky				00–99
07	OUT	0	0	SQWE	0	0	RS1	RS0	

Při čtení z adresy 00 přečtete vždy sekundy. Nejvyšší bit na adrese 0 se jmenuje Clock Halt, a pokud do něj zapíšete 1, hodiny budou stát. Jakmile do něj zapíšete 0, hodiny se rozběhnou.

Na adrese 1 jsou minuty, na adrese 2 hodiny (bit 6 přepíná mezi 12 a 24hodinovým režimem).

Adresa 3 říká, jaký je den v týdnu, adresa 4 udává den v měsíci, adresa 5 měsíc a na adrese 6 jsou roky. Adresa 7 obsahuje řídicí slovo, kterým můžete nastavit funkci budíku. Bližší informace najdete, jak jinak, v datasheetu.

Zvídavé mozky už chystají otázku: *A jak to, že ten obvod ví, kolikátého dneska je?*

Odpověď je jednoduchá: Při prvním zapojení do něj zapíšete aktuální datum a čas. Takhle jednoduché to je.

Ovšem, ptá se zvídavec dál: *Jak je možné, že si to ten obvod pamatuje i když vypnu proud? Nebo to musím při každém zapnutí nastavit?*

Milé zvídavé uklidním: Žádný zázrak se neděje. K obvodu lze připojit malý článek, baterii, která vydrží takový obvod napájet několik let, a zároveň je u něj připojený krystal, který kmitá na frekvenci 32,768 kHz, a tím udržuje přesný čas. Tedy, zas tak úplně přesný není, může se s reálným rozejít i o několik minut za měsíc, ale na většinu běžných aplikací to stačí. Pokud potřebujete vyšší přesnost, použijte obvod DS3231, který se posune maximálně o několik minut ročně, a popřípadě můžete ještě hodiny pravidelně seřizovat vnějším signálem, třeba z GPS nebo z internetu.

Schéma a zdrojový kód najdete na ⌀ https://eknh.cz/rtc

27 Paměťové karty

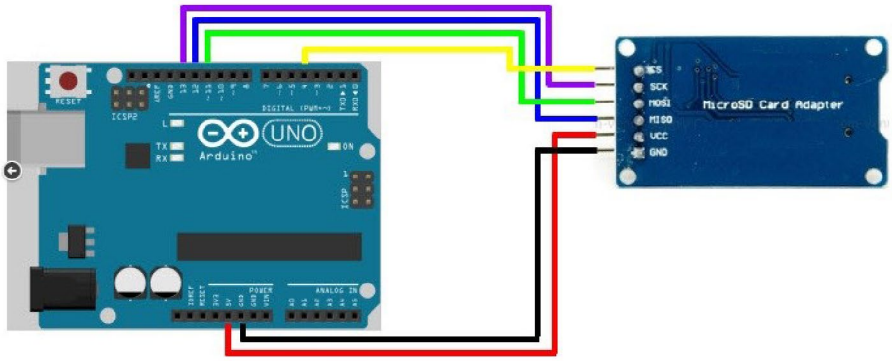
27 Paměťové karty

Arduino se často používá jako takzvaný *data logger*, tedy zařízení, které někde leží, sbírá data ze senzorů a zaznamenává si je pro další zpracování.

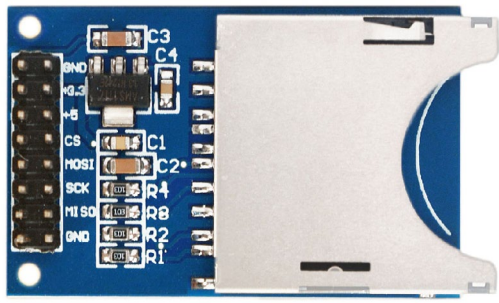
Otázka je: Kam si je zaznamenává?

Samozřejmě, může si je zaznamenávat do vnitřní paměti, ale tam se jich moc nevejde. Druhá možnost je použít externí paměť EEPROM. No a třetí možnost je využít standardní paměťovou kartu typu SD, naformátovanou na souborový systém FAT (FAT16 / FAT32).

Taková paměťová karta se k Arduinu připojujue jako zařízení na sběrnici SPI – používá tedy signály MOSI, MISO, SCK a Slave Select. MOSI, MISO a SCK mají v Arduinu pevná místa, SS připojte třeba na pin 4 – to je častý způsob.



Mimochodem – SD karty vyžadují napájení 3,3 voltu, ale naštěstí různé adaptéry „SD Card for Arduino“ obsahují převodníky úrovní a stabilizátor:



Pro Arduino existuje knihovna SD, která výrazně zjednodušuje práci s těmito kartami. Při inicializaci stačí zavolat metodu SD.begin(cs), kde cs je číslo pinu, kam jste připojili Slave Select pro SD kartu.

Pomocí metody SD.open pak otevřete soubor pro zápis nebo čtení, stejně jako u normálního PC, a pomocí funkcí print, println, read a write zapisujete a čtete data.

Pokud bude vaše zařízení intenzivně zapisovat na SD kartu, tak se souborovému systému FAT vyhněte (při každé změně zapisuje do několika málo sektorů) a použijte kvalitní karty „průmyslového typu" (které mají garantovaný vyšší počet zápisových cyklů, ovšem také výrazně vyšší cenu). S „obyčejnými" SD kartami při nevhodně vyřešeném zápisu např. každou sekundu narazíte na limit bezpečně po několika dnech.

Schéma a zdrojový kód najdete na ⌀ https://eknh.cz/sdcard

28 Logický analyzátor, logická sonda

28 Logický analyzátor, logická sonda

V této knize nechci jen ukazovat zapojení a vysvětlovat, jak fungují, ale také dát pár tipů na zajímavé součástky a nástroje, které se vám budou hodit.

Jedním takovým nástrojem je logická sonda (*logical probe*). Můžete ji sehnat za cenu okolo 200 Kč. Vypadá zhruba takto:



Červený a černý přívod zapojíte na napájení zkoumaného obvodu (černý je zem), a pomocí hrotu se můžete dotýkat různých míst v obvodu a zjišťovat, jestli je v tom kterém místě logická 0, nebo logická 1. Sonda dokáže rozpoznat i to, že se napětí mění, tedy že se pravděpodobně dotýkáte místa, kde probíhají nějaké hodinové pulsy nebo signál. Složitější logické sondy mají i akustickou signalizaci, přepínání mezi TTL a CMOS nebo detekci náběžných a sestupných hran.

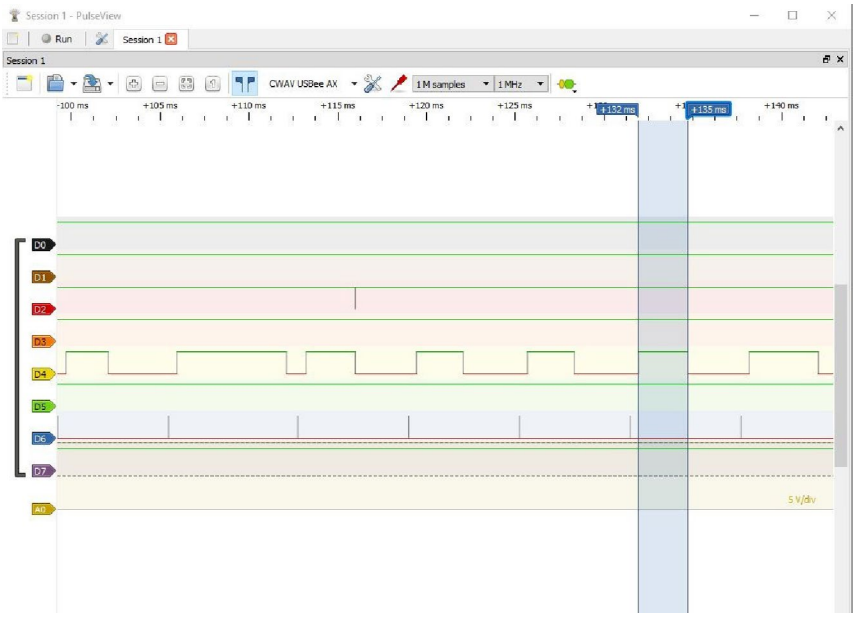
Druhý nástroj, který se může hodit, je logický analyzátor. Můžete ho sehnat za mnoho desítek tisíc jako samostatné zařízení s displejem a spoustou funkcí, ale pro běžné použití naleznete i jednoduchou verzi, připojenou přes USB k počítači – vypadá jako na následujícím obrázku:



Takový analyzátor stojí zhruba sto korun. Připojíte jej k počítači přes USB. K ovládání můžete použít například open source programy Sigrok / PulseView (je potřeba správně nastavit ovladače, viz dokumentace).

Pro vlastní měření propojte zem analyzátoru se zemí měřeného obvodu. Můžete měřit až osm signálů najednou (CH0 až CH7). Připojte vývody do míst, která chcete měřit, a spuštěním měření (Capture) logický analyzátor začne sledovat průběhy v těchto bodech.

Logický analyzátor je velmi užitečná pomůcka pro zkoumání, zda v obvodu funguje vše jak má, popřípadě proč to nefunguje. Na displeji vidíte průběhy signálů v čase. Můžete si například změřit šířku pulsů, podívat se, zda následují po sobě tak, jak mají, lepší software umí například pracovat se sběrnicemi I²C nebo SPI a rovnou ukázat, jaká data po jakém vodiči šla…



29 Elektronika a svět kolem nás

29 Elektronika a svět kolem nás

V této prakticky zaměřené kapitole shrnu nejrůznější způsoby, jakými elektronické konstrukce komunikují se světem. Zmíním jak dosud neprobrané způsoby, tak i ty, kterými jsme se už zabývali. Primárně se zaměřím na způsoby, jak taková zařízení připojit k číslicovým obvodům.

29.1 Ovládáme přírodu elektronikou

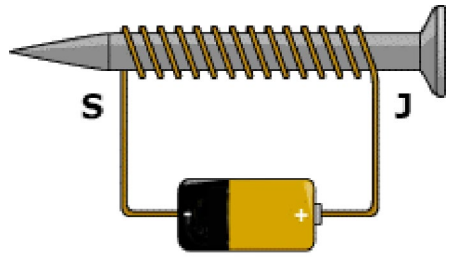
Do této podkapitoly patří situace, kdy potřebujeme elektronikou nějak působit na svět okolo sebe, ať už mechanicky, nebo jinak.

29.1.1 Elektromagnety

Elektromagnet vznikne tím, že cívce dáme jádro z magneticky měkkého kovu. Princip je jednoduchý: Prochází-li proud, je okolo cívky magnetické pole. Když proud vypneme, pole zaniká. Elektromagnetem lze ovládat kovové mechanické části, různé závory, zámky, ventily atd.

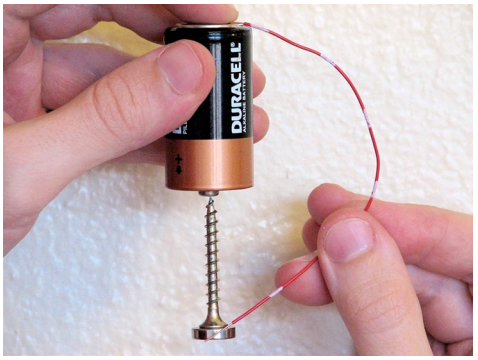
Pro elektromagnety platí vše, co jsem výše napsal o elektromotorech: připojujeme je vždy přes budič a s ochrannou diodou.

Nemáte elektromagnet? Snadná pomoc, stačí kousek izolovaného drátu a hřebík. Omotejte drát okolo hřebíku, a na jeho konce připojte napětí. Hřebík začne fungovat jako magnet.



Když celou tuhle sestavu postavíte na výšku tak, že hřebík bude zasunutý třebas jen do třetiny cívky a zapnete proud, cívka vtáhne hřebík dovnitř. Pokud má malou sílu, znásobte počet závitů (třeba naviňte druhou vrstvu přes první.

Mimochodem, pokud máte k dispozici silný neodymový magnet – bývají to takové malé kovové válečky – můžete si pomocí hřebíku (nebo šroubu), monočlánku a kousku drátu udělat jednoduchý motor. Viz obrázek.



Jak to funguje? Procházející proud vyvolá magnetické pole, které se bude, obrazně řečeno, „přetlačovat" s magnetickým polem magnetu. A protože je celá konstrukce lehká a má jen minimální tření, bude i malé magnetické pole stačit k tomu, aby se vše otáčelo.

29.1.2 Motory

Pokud potřebujete něčím otočit nebo něco posunout, zvolíte nejspíš elektromotor. Elektromotorů je obrovská škála, od těch nejmenších modelářských až po obrovské elektromotory v lokomotivách.

Elektromotor se skládá ze dvou částí, ze **statoru** a **rotoru**. Jsou různé typy a různá uspořádání těchto dvou částí. Nejčastěji se používá uspořádání, v němž je pevná část (stator) tvořena permanentním magnetem a otáčivá část (rotor) cívkou. Proud, procházející cívkou, vytváří okolo této cívky také magnetické pole, které cívkou točí tak, aby její severní pól byl co nejblíž jižnímu pólu pevného magnetu. Pokud v tu chvíli změníme směr proudu, tekoucího cívkou, její magnetické pole se otočí, přepóluje, a pevný magnet ji odpuzuje. Cívka se tedy opět pootočí…

V praxi bývá na rotoru cívek víc, většinou tři nebo šest, ale může jich být i víc.

U stejnosměrného motoru se používá speciální část, nazvaná komutátor, která se stará právě o přepojování napětí tak, aby se pravidelně měnil proud cívkami. Rychlost otáčení takového motoru je úměrná velikosti proudu, který jím protéká.

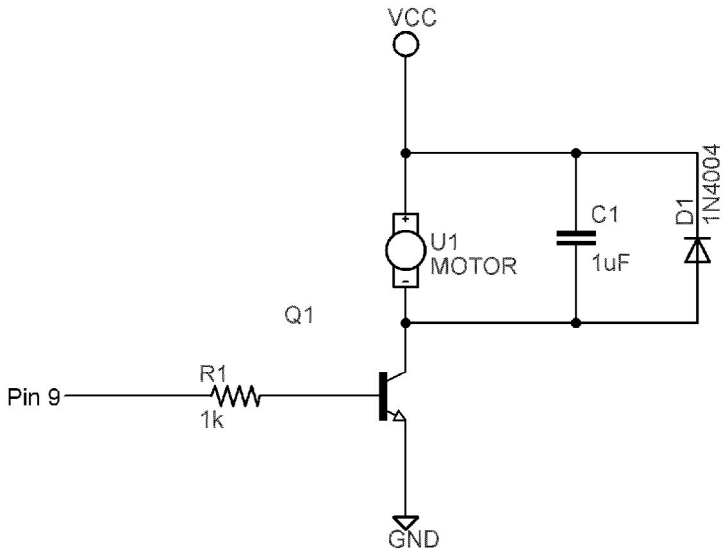
Existují i motory bez této součásti (bezkomutátorové, brushless), u nichž musí vnější elektronika řídit proud jednotlivými cívkami, a podle frekvence přepínání těchto cívek lze přesně řídit rychlost motorku. S takovými motory se setkáme tam, kde je na závadu případné jiskření, které hrozí u komutátorů, popřípadě tam, kde je dobré mít možnost ovládání otáček – například ve větráčcích u počítačů.

Motor nemusí být jen stejnosměrný, může fungovat i na střídavý proud, ovšem pak bývá jinak zapojen. Může mít uspořádání obrácené, tj. stator složený z cívek a rotor s permanentním magnetem. Místo permanentního magnetu může být rovněž použita cívka… Typů je nepřeberné množství.

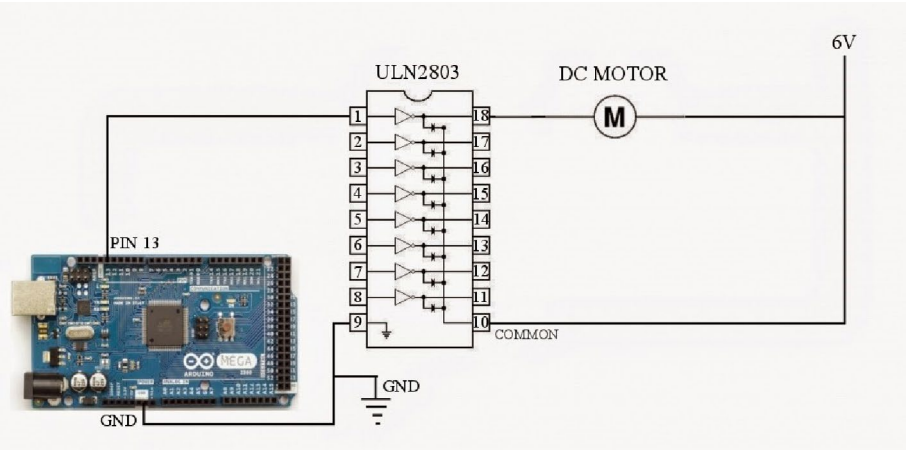
Pro nás je důležité, že motor je zařízení, které používá cívky. Cívka má tu vlastnost, že při odpojení proudu v ní zaniká magnetické pole, a zanikající magnetické pole v ní zpětně naindukuje proud opačného směru. Pokud bychom připojili cívku přímo k vývodu číslicového obvodu, tak při jejím vypnutí by tekl velký proud opačné polarity do vývodu číslicového obvodu, a tím by ho téměř jistě zničil.

Proto se motory (ale i další zařízení s cívkami) připojují k číslicovým obvodům **zásadně** s paralelně připojenou ochrannou diodou v závěrném směru. Navíc platí, že motory potřebují poměrně velké proudy, které číslicové obvody nejsou schopné poskytnout, takže se zapojují přes speciální budiče. A taky se snažte vyhnout tomu, abyste zapojovali motory přes nepájivá kontaktní pole. Ta nejsou stavěná pro velké proudy, a *mohlo by dojít k poškození*, tak prosím opatrně.

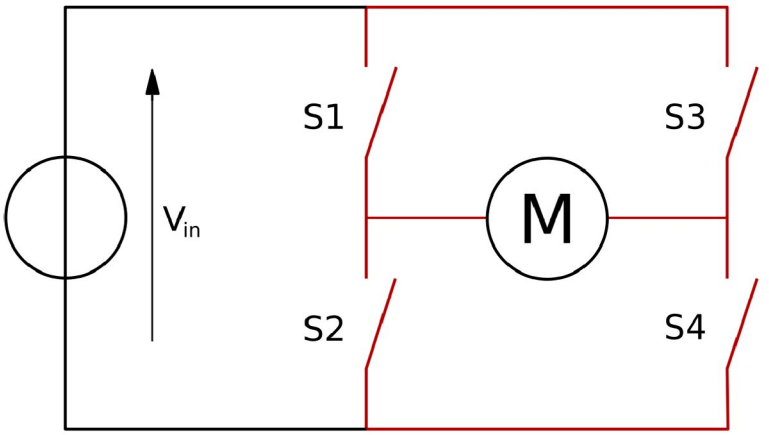
Nejjednodušší budič je obyčejný tranzistor, o němž víme, že malým proudem dokáže spínat velký proud.



Další možnost je použít specializovaný obvod (například ULN2803), který obsahuje osm budičů a osm ochranných diod, takže můžete jedním takovým obvodem spínat až osm motorů.

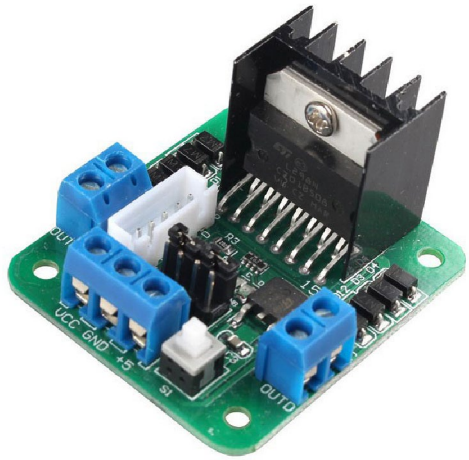


Těmito způsoby můžete motor zapnout a vypnout. Pokud jej potřebujete ovládat sofistikovaněji, třeba zapínat v obou směrech otáčení, použijete takzvaný **H můstek** (H bridge).



Toto zapojení umožní připojit motor v obou směrech – buď kombinací S1 + S4, nebo kombinací S2 + S3. Pokud jsou všechny spínače rozpojené, je motor navolno, pokud jsou sepnuté např. S2 a S4, je motor zabrzděn.

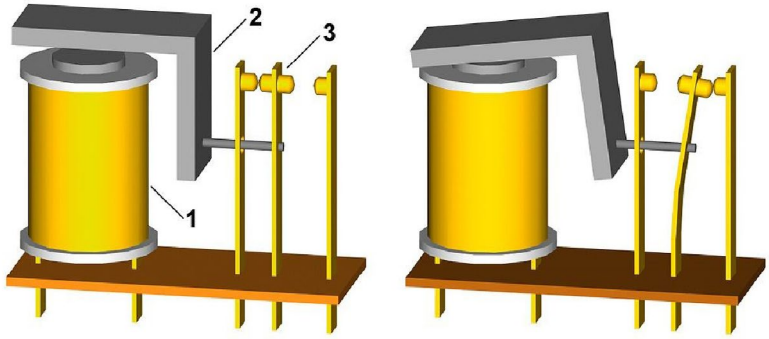
Samozřejmě místo spínačů mohou být použité tranzistory, nebo opět specializovaná součástka – integrovaný H můstek. Můžete ho koupit už jako celou desku, i s konektory.



Na obrázku vidíte takový integrovaný H můstek, řídicí vstupy od elektroniky a výkonové konektory pro zapojení motorů. U těchto můstků je časté, že vlastní spínací element (integrovaný obvod nebo výkonový tranzistor) má zabudované chladiče.

29.1.3 Relé

Relé je zase cívka, k níž je přes mechanismus připojena trojice vývodů. Když je cívka (1) bez proudu, jsou spojeny dva z nich (společný a NC). Když cívkou protéká proud, přitáhne k sobě mechanickou kotvu (2), a ta přepne vývody (3) tak, že jsou spojené jiné dva (společný a NO).



CC-BY-SA, autor Teslaton

Relé je velmi stará elektrická součástka, ale používá se dodnes. Umožňuje jednak spínat velké proudy i napětí, včetně střídavého (není problém pomocí 5 V relé spínat 230 V síťové napětí), a zároveň elektricky odděluje řídicí část od spínané. Pokud chcete elektronikou spínat například lampičku do zásuvky, použijte relé, protože tím fyzicky oddělíte vlastní elektroniku od silového napětí.

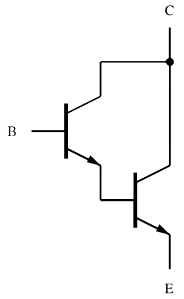
Vzhledem k tomu, že jde o součástku elektromechanickou, musíme počítat s tím, že přepnutí nějakou dobu trvá. K nějakému spínání vysokou frekvencí, v intervalech sekundových a kratších, není relé vhodné.

A protože řídicím elementem je opět cívka, tak při připojování k elektronice nesmíme co? No nesmíme ji připojit přímo, vždy přes budič a s ochrannou diodu. Viz výše.

29.1.4 Darlington, FET, Tyristor

Další spínací prvky, podobné relé, ovšem bez výhody elektrického oddělení obou zařízení, zato s výhodou přímého připojení na výstupy číslicových obvodů, jsou například výkonové tranzistory, tyristory a triaky.

Často používaným zapojením tranzistorů pro spínání velkých proudů je takzvané Darlingtonovo zapojení. Už jsme si jej představili, pro připomenutí schéma:



Takto zapojená dvojice tranzistorů se chová jako jeden tranzistor s mnohem větším proudovým zesilovacím činitelem, tedy dokáže velmi malými proudy spínat velmi velký proud. Výše zmíněný obvod ULN2803 používá vnitřně právě takovéto tranzistory v Darlingtonově zapojení.

Ke spínání velkých proudů můžeme použít i výkonové tranzistory MOSFET. U těchto tranzistorů se, jak víme, nespíná procházejícím proudem, ale přivedeným napětím. Spínací proud je v řádech mikroampérů, spínaný proud i několik ampérů.

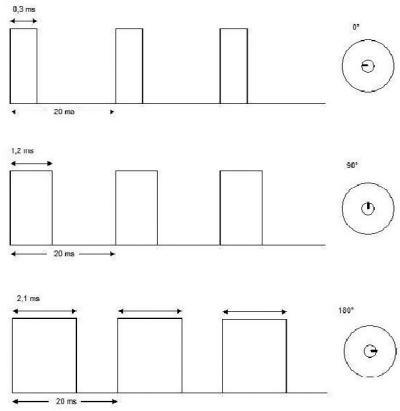
Dřív se často používaly i spínací prvky tyristory a triaky. Principiálně jsou podobné tranzistorům, ale mají víc N-P přechodů. Dodneška se používají ve specializovaných oblastech, ale v amatérské mikroelektronice se častěji setkáte s výkonovým MOSFETem či relé.

29.1.5 Servo

Servo je speciálně upravený elektromotor se zpětnou vazbou a soustavou převodů, které se otáčí v rozsahu 240° (někdy 180°, někdy i 360°). Často se s nimi potkáte v různých hobby robotech, kde pohybují rameny, popřípadě v modelech, kde nastavují například křidélka nebo kormidlo u lodi. Výhodou serva je možnost přesně nastavit natočení osy.



Servo mívá tři vstupy – zem, napájecí napětí a řídicí vstup. Serva se řídí pomocí PWM signálu, kde šířka pulsu určuje natočení osy:



K ovládání serv se používají specializované obvody, které mají rozhraní (např.) I²C a dokážou ovládat až 16 serv. Opět platí, že takové obvody často seženete jako hotové moduly i s konektory.

29.1.6 Krokový motor

Krokový motor je vlastně obyčejný elektromotor, který má vyvedené jednotlivé cívky. Ovládací elektronika postupně spíná jednu po druhé a tím otáčí osou po malých krocích.

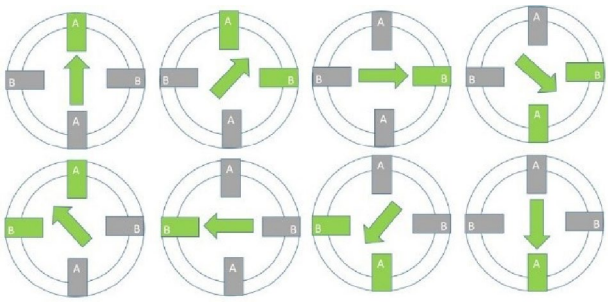


Fig3 - One-two phase on - half step

Ve skutečnosti mívají takové motory třeba 200 kroků na jednu otáčku. Krokové motory se používaly třeba v tiskárnách pro přesný posun vozíku. Dnes se používají ve 3D tiskárnách, kde zaručují totéž: přesný posun předmětu a trysky.

Krokový motor je opět zařízení s cívkou, takže zase platí: nepřipojovat napřímo, jsou na to speciální obvody. Hledejte „stepper motor driver“.

29.1.7 Světlo

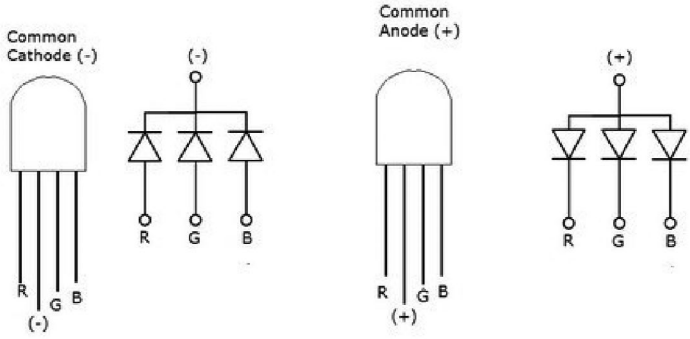
Jako zdroj světla jsme si dlouhá léta vystačili s žárovkou. V číslicové technice se žárovka moc nehodí, protože potřebuje poměrně velký proud, a to i když je malička. Tam ji nahradily LED.

Když ale přesto budete potřebovat ovládat žárovku, použijte k tomu buď tranzistor (žárovky do 12 voltů), nebo relé (pro žárovky s velkým výkonem).

Dnes jsou dostupné i výkonové LED. Počítejte ale s tím, že takové LED potřebují vyšší napětí a přesně daný proud. Pro některé účely můžete použít i laserové LED – podobné těm, co jsou v laserových ukazovátkách. Ovšem u takových LED je potřeba dodržet proud i napětí; obyčejné připojení na 5 V je pravděpodobně zničí.

U výkonnějších zdrojů světla počítejte s nutností chlazení. I když jsou LED… Minimálně pasivním chladičem (což jsou různě žebrované hliníkové profily), pokud je to nedostatečné, budete muset použít ventilátor.

Zajímavá součástka, která je i v EduShieldu, je RGB dioda. Jde vlastně o tři barevné LED (červená, zelená, modrá) v jednom pouzdru, buď se společnou anodou, nebo se společnou katodou. Taková součástka může měnit barvu vyzařovaného světla v širokém rozmezí – například smícháním červeného a zeleného světla získáte světlo žluté apod. RGB LED se používají v různých barevných displejích nebo v osvětlovacích páskách.



V EduShieldu je použito zapojení se společnou anodou. Znamená to, že společný vývod je připojen k napájecímu napětí a k tomu, aby se dioda rozsvítila, je zapotřebí připojit vstupy R, G, B k zemi (logická úroveň 0).

29.1.8 Peltierův článek, topná spirála

Zajímavá součástka je Peltierův článek. Ten využívá takzvaného Peltierova jevu: když prochází proud rozdílnými vodiči, zapojenými sériově, tak se jedna z jejich styčných ploch zahřívá, druhá ochlazuje. Peltierovy články se vyrábějí nejčastěji jako čtvercové keramické destičky s rozměry 10 × 10 až 60 × 60 milimetrů, většinou kolem 3 milimetrů silné, s dvěma vývody. Pokud skrz takový článek necháte procházet proud, např. z 12 V zdroje, jedna jeho plocha se zahřívá, druhá ochlazuje. Až to budete zkoušet, tak doporučuju nesahat na článek přímo – já držel dva články při testu v ruce, a byla to opravdu hloupost, protože ten rozdíl teplot mezi chladnou a teplou částí je i u malých článků třeba 40 °C, a navíc je téměř okamžitý po zapnutí napětí.

Když budete teplou část chladit větrákem, můžete na chladné snadno dosáhnout teplot pod bodem mrazu, a obráceně – když budete zahřívat chladnou část, dosáhnete na teplé snadno bodu varu. Peltierův článek se označuje též TEC – Thermoelectric Cooler.

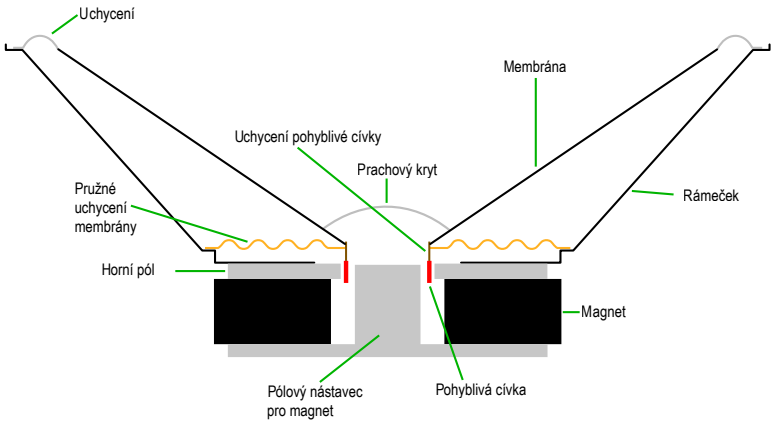


Topit můžete ale i jinak. Většinou tam, kde to nepotřebujete a nechcete (integrované obvody, tranzistory, rezistory). Takové teplo ale není moc užitečné. Pokud potřebujete něco ohřívat, zvolte topnou spirálu. Ano, stejnou, jaká je v elektrických vařičích, rychlovarných konvicích a podobných spotřebičích. Topná spirála je obyčejný drát z kovové slitiny s daným odporem, stočený do spirály (někdy i vícenásobně) a izolovaný. Pokud tímto rezistorem prochází proud, mění se v teplo (vzpomeňte na Ohmův zákon a vzorec pro výpočet výkonu).

Pokud budete něco z toho ovládat číslicovými obvody, tak vždy přes relé.

29.1.9 Reproduktor

Reproduktor slouží k přeměně elektrického proudu na zvuk. Zvuk, jak asi víte, je mechanické vlnění, které se šíří hmotou, včetně vzduchu. K tomu, aby vznikl zvuk, je zapotřebí mechanického chvění nějakého předmětu – třeba struny, blány (u bubnu), jazýčku u pískacích hraček, popřípadě vzduchového sloupce (píšťaly). U reproduktoru se chvěje tenká papírová membrána. Chvění vzniká tak, že k membráně je připevněná cívka, která je umístěna mezi silné magnety. Když se mění proud cívkou, mění se její magnetické pole, cívka je více či méně zatahována do magnetu, a vzniká tak chvění, které se přenáší na membránu, a membránou do vzduchu.



CC-BY-SA, autor Iain

Můžete vynechat magnet, cívku udělat stabilní a membránu kovovou. Při vhodně zvolené pružnosti bude proud cívkou přitahovat membránu, a ta bude zase rozechvívat vzduch. Takto pracují **sluchátka**.

Zase platí: když chcete připojit sluchátko nebo reproduktor, musíte počítat s tím, že obsahují cívku, podobně jako u motorů a elektromagnetů. Reproduktory navíc mají malý odpor, takže potřebují velké proudy. Sluchátka mají odpor větší, ale i tak je dobrý zvyk zapojit je přes tranzistor a ještě odfiltrovat stejnosměrné napětí kondenzátorem. Jde to i bez toho, ale výsledný zvuk je pak zkreslený.

Funguje to i obráceně? Funguje. Když budete mluvit do sluchátka nebo reproduktoru, bude se v cívce indukovat napětí, které můžete zpracovat a převést na digitální informaci. Toto je princip elektromagnetického **mikrofonu**. Existují i jiné typy – uhlíkový mikrofon mění svůj odpor, kondenzátorový mění svou kapacitu, piezoelektrický generuje napětí mechanickou deformací krystalů…

29.2 Příroda ovládá elektroniku

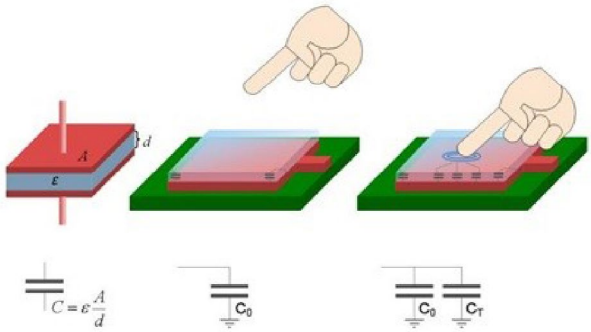
Stejně jako je někdy potřeba, aby elektronika ovládala svět okolo nás, tak bývá potřeba, aby naopak nějak reagovala ona na to, co se kolem ní děje. Třeba na to, že uživatel stisknul tlačítko nebo že vzrostla teplota.

29.2.1 Tlačítka a klávesy

Tyto součástky jsem už popisoval, tak jen pro pořádek připomenu: Jedná se o mechanické součástky, které při stisknutí buď vodivě sepnou dva vývody, nebo je naopak rozepnou, popřípadě přepnou. Můžete uspořádat větší množství tlačítek do podoby klávesnice, buď se samostatnými tlačítky, nebo (častěji) do matice.

29.2.2 Dotyk

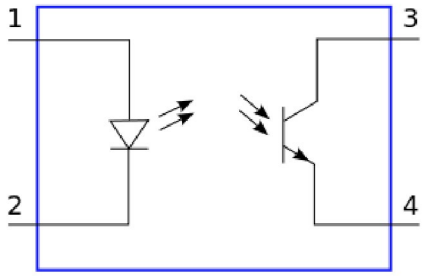
Tlačítko potřebuje, aby člověk vyvinul nějaký tlak prstem a mechanicky tak posunul spínací element do požadované polohy. Existují ale obvody, které reagují na přiblížení či dotyk. Nejčastěji jsou postavené jako kondenzátor, a princip jejich funkce je ten, že přiblížení prstu vyvolá změnu kapacity.



29.2.3 Světlo – fotorezistor, fotodioda, line tracking

Princip fotorezistoru jsem už probíral. Citlivější a rychlejší snímání světla umožňují polovodičové fotodiody a fototranzistory. Opět se využívá citlivosti polovodiče na světlo.

Pokud potřebujete z nějakého důvodu oddělit dva obvody od sebe, můžete využít takzvaný *optocoupler*. Česky se používá termín *optočlen*. Jde o součástku, která v jednom pouzdru integruje LED a fototranzistor. LED funguje tak, jak jste zvyklí, a intenzita jejího světla otvírá nebo zavírá fototranzistor.



Optické oddělovače se používají například u dlouhých datových vedení, kde není zajištěno, že budou obvody na obou koncích připojené ke stejné úrovni země, a při prostém propojení by hrozilo zničení velkým proudem.

Dvojice LED – Fototranzistor se používá třeba i v detektorech přiblížení nebo ve světelných závorách. U snímačů přiblížení se obě součástky umístí vedle sebe. LED vysílá pravidelné pulsy, a pokud se přiblíží nějaký předmět, který dostatečně odráží světlo (třeba i prst), odrazí se puls i do fototranzistoru.



Samozřejmě takový detektor nebude moc fungovat, pokud k němu přiblížíte černé těleso, které, jak známo, moc světla neodráží (proto je ostatně černé). Tento efekt se používá například u robotů, kteří dokážou sledovat nakreslenou čáru. Robot svítí před sebe na zem, a několik fototranzistorů snímá odražené světlo. Pokud se neodráží žádné, znamená to, že u daného fototranzistoru je pravděpodobně nakreslená černá vodicí čára.

29.2.4 Magnetismus

Pro snímání magnetického pole může sloužit cívka – ovšem ta, jak víte, indukuje napětí pouze tehdy, když se magnetické pole mění. Pro použití v číslicové technice je vhodnější Hallova sonda. Ta je tvořena polovodičovou destičkou, skrz níž prochází proud. Když tuto sondu vložíte do magnetického pole, indukuje se na ní napětí, které je možné měřit.

Složitější součástky (magnetometry, digitální kompasy) dokáží měřit i zemský magnetismus ve třech osách a dát tak informaci o tom, v jaké pozici se zařízení nachází. Nemusíte se ale bát – i takovéhle součástky jsou dostupné pro běžnou amatérskou praxi. Většinou používají rozhraní I^2C. Příkladem může být obvod HMC5883L.

29.2.5 Otáčení, posun

Pro snímání mechanického pohybu existuje celá řada řešení, podle toho, co se má vlastně detekovat. Buď se snímá posun na nějaké dráze, nebo otáčení, popřípadě kombinace obojího.

Nejjednodušší měření otáčení a posunu představují otočné a tahové potenciometry. Jejich nevýhoda je, že mají omezený rozsah (u posunu pár centimetrů, u otáčení většinou necelou jednu otáčku, nanejvýš jen několik málo otáček). Výhoda je, že víte, v jaké pozici se nachází jezdec, jestli to je na kraji nebo třeba uprostřed.

Při snímání otáček můžete použít rotační enkodér. Vzpomeňte si: otáčet můžete donekonečna, a dvojice spínačů říká, že se osa pootočila o nějakou definovanou část, a jakým směrem. Výhoda je to, že můžete točitstále dokola, nevýhoda je, že nevíte, jak je právě nastavená osa. Víte jen, že se otáčí.

Mechanické snímání rotačním enkodérem má i další nevýhodu – časem se opotřebují kontakty. Tuto nevýhodu odstraňují jiné metody snímání otáček: optické (na hřídeli jsou světlé a tmavé části, a pomocí LED a fototranzistoru snímáte, jestli je k vám natočena světlá nebo tmavá část), nebo třeba magnetické (v hřídeli jsou magnety a snímá se magnetické pole Hallovou sondou). Pokud máte jen jeden měřicí element (jednu sondu, jeden fototranzistor), dokážete snímat rychlost, ale nezjistíte směr. Pokud použijete dva vedle sebe, zjistíte i směr (podle toho, v jakém pořadí budou aktivní). Pokud je potřeba určit i úhel natočení, použije se rozdělení na mnoho segmentů (pro zvýšení přesnosti), a k tomu jeden další prvek, který vždy oznámí, když se sledovaný objekt natočí do úhlu 0.

Pro snímání posunu lze použít stejného principu – třeba nakreslit na dráhu pravidelně se střídající černé a bílé proužky a pomocí LED a fototranzistorů odečítat pulsy.

29.2.6 Poloha, zrychlení

Z mobilních telefonů jistě víte, že tato zařízení dokážou zjistit, v jaké jsou pozici – jestli leží na stole, nebo jestli je někdo zdvihnul, jestli je držíte na výšku nebo na šířku a spoustu dalších věcí. K tomu slouží součástky, zvané **akcelerometry** a **gyroskopy**.

Akcelerometr měří zrychlení ve třech osách (X, Y, Z). Pokud je zařízení v klidu, působí na něj tíhové zrychlení Země. Akcelerometr dokáže toto zrychlení detekovat. Když zařízením otočíte, změní se hodnoty tíhového zrychlení pro jednotlivé osy.

Vlastností zrychlení je, že se skládá jako vektor, takže samotný akcelerometr nedokáže rozpoznat, jestli síla, která na něj působí, je zemská tíže, nebo zrychlení při pohybu. Případný pohyb tak může zaměnit za náklon. Tuto nectnost odstraňuje druhá součástka, gyroskop. Ta zase neměří zrychlení, pouze určuje, jestli se změnila orientace zařízení v prostoru – tedy jestli jste s ním otočili.

V moderních součástkách se často kombinuje akcelerometr, gyroskop a kompas, a taková součástka dokáže podat hodně komplexní informaci o tom, v jaké pozici se zařízení nachází. Příkladem takové součástky je MPU9150 – je sice složitá a komplexní, ale s okolím komunikuje přes sběrnici I^2C. Práce s ní, například v Arduinu, je tedy velmi snadná.

29.2.7 Zvuk

K detekci zvuků se používají téměř výlučně mikrofony. Nejjednodušší použití jsou „hlukoměry“, které nedetekují, o jaký zvuk jde, ale jen to, že vznikl nějaký hluk – třeba tlesknutím. Složitější obvody s vlastními procesory dokážou rozpoznávat vzorky slov a implementovat tak třeba jednoduché hlasové ovládání. Takové obvody pak komunikují se zařízením pomocí nějakého standardizovaného rozhraní (sériový port např.)

Druhá možnost zpracování zvuku je, že signál z mikrofonu vzorkujete pomocí rychlého analogově-digitálního převodníku (ADC) a zvuk tak *samplujete*. Pokud je rychlost čtení vzorků dostatečně vysoká (třeba hudební CD používají 44,1 kHz) a přesnost alespoň 12 bitů, získáte velmi slušný záznam zvuku k dalšímu zpracování.

29.2.8 Teplota, vlhkost

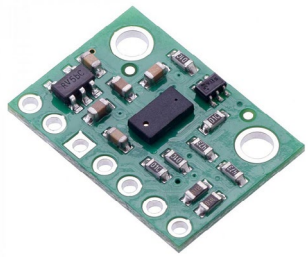
Termistor i obvod LM35 jsme si už představili. Představili jsme si i digitální teploměr LM75 s rozhraním I^2C. Moc víc vám toho stran měření teploty neukážu – samozřejmě jsou i další součástky, které mají vyšší přesnost, ale princip zůstává stejný.

Zajímavý doplněk jsou senzory relativní vlhkosti. Jejich princip je jednoduchý – obsahují substrát, který pohlcuje vlhkost, a k němu jsou připojené dvě elektrody. Čím je substrát vlhčí, tím menší je jeho odpor, takže měřením odporu se pak lehce zjistí vlhkost. Odpor samozřejmě záleží i na teplotě, a tak bývají často vlhkoměry kombinované v jednom pouzdru i s teploměrem.

Často používané senzory jsou DHT11 a DHT22 – jsou levné a dostupné a jejich připojení k Arduinu je extrémně snadné. Jediný problém je s jejich ovládáním. Nepoužívají totiž standardní sběrnice, ale přenáší data po jednom vodiči, podobně jako sběrnice 1-Wire (ale ne úplně stejně). Naštěstí to řešit nemusíte, protože existují hotové a vyzkoušené knihovny.

29.2.9 Vzdálenost

Jednu z možností, jak měřit vzdálenost, jsme si už ukazovali – bylo to měření vzdálenosti pomocí ultrazvuku. Na podobném principu pracují i takzvané Time-of-Flight (ToF) senzory, ovšem místo (ultra)zvuku používají světelný paprsek. Jenže světlo letí mnohem rychleji, tak i přesnost měření musí být o hodně vyšší. Naštěstí zase existují součástky, které vše potřebné zařídí za vás.



ToF senzor je ta černá obdélníková součástka uprostřed, ta se dvěma malými otvory.

29.2.10 Tlak

Takovým „hello world" pro konstrukce s Arduinem bývá domácí meteostanice. To takhle vezmete digitální teploměr, digitální vlhkoměr a digitální tlakoměr, připojíte to k Arduinu a během deseti minut měříte všechno možné. Nejčastěji používaným tlakoměrem je BMP180 (nebo přesnější BMP085), který integruje měření tlaku a teploty. Asi vás ani nepřekvapí, že opět komunikuje přes rozhraní I^2C.

Měření tlaku vyžaduje korekci podle nadmořské výšky, protože tlak vzduchu, jak známo, s rostoucí nadmořskou výškou klesá. Tlak, který naměříte, se proto přepočítává k univerzálnímu vztažnému bodu, totiž na „hladinu moře". Anebo to můžete celé udělat naopak, a ze znalosti skutečného tlaku a naměřeného určit nadmořskou výšku.

29.2.11 Plyn

Zajímavé senzory jsou takové, které detekují přítomnost plynů či prachu ve vzduchu. Princip prachových snímačů je prostý – vzduch prochází komorou se stanoveným objemem. Do této komory bliká intenzivní světlo z LED a měří se, jaká část projde skrz a jaká část se rozptýlí na prachových částicích.

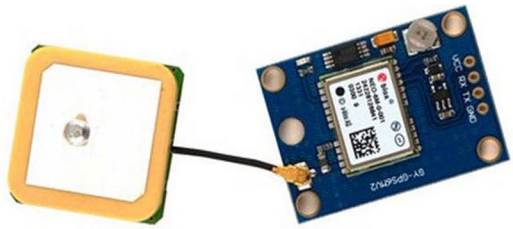
Senzory plynů se nejčastěji používají elektrochemické – obsahují destičku z látky, která reaguje s určitými plyny, jako je metan, propan, butan, páry etanolu, oxid uhelnatý, vodík, ozon, čpavek atd. Při reakci vzniká elektrické napětí, které se měří a podle jeho velikosti lze usuzovat na koncentraci daného plynu.

Jiný způsob měření používá technologii NDIR – Non-Dispersive Infra Red Sensors. Tyto senzory využívají toho, že určité plyny pohlcují záření určitých vlnových délek. Opět tedy používají komoru, do které svítí jasným světlem a měří úbytek světla přesně dané vlnové délky.

29.2.12 GPS

Co bylo před dvaceti lety fantazií ze stránek populárně-naučných magazínů, to je dnes leckdy běžně dostupnou realitou. Jako příklad můžu jmenovat třeba GPS – systém geolokace pomocí družicového signálu. Vlastně extrémně složité zařízení, které přijímá signál z družic, u nichž jsou známy přesně jejich pozice, a z posunu signálu dopočítává přesnou pozici přijímače, tedy zeměpisnou délku, šířku i nadmořskou výšku.

Takové přijímače nejsou úplně nejlevnější, stojí i několik set korun, ale jsou bez problémů dostupné i pro amatérské konstrukce. Navíc mnohé z nich dokáží využít kromě amerického systému GPS i ruský Glonass a evropský Galileo.



Připojení takových přijímačů je navíc extrémně jednoduché, protože naprostá většina z nich používá jednosměrné sériové rozhraní UART s rychlostí přenosu 4800 Bd. Přijímač posílá neustále informace o pozici a dalších hodnotách pomocí takzvaného protokolu NMEA. Jedná se o velmi jednoduchý protokol, co řádek, to informace. Přijímač posílá různé typy informací, ta, která vás asi bude zajímat nejvíc, začíná znaky „$GPRMC“ a za nimi je 11 hodnot, oddělených čárkami. Například takto:

$GPRMC,225446,A,4916.45,N,01511.12,E,000.5,054.7,090717,020.3,E*68

- 225446 je čas měření – 22:54:46 UTC
- A informuje, jestli je měření platné – A = Valid position, V = Warning
- 4916.45,N Šířka 49 deg. 16.45 min. North (severní)
- 01511.12,E Délka 15 deg. 11.12 min. East (východní)
- 000.5 Rychlost vůči Zemi (v uzlech – knots, 1 knot = 1.852 km/h)
- 054.7 Azimut pohybu
- 090711 Datum měření (UTC)
- 020.3,E Magnetická odchylka 20.3 stupňů východně
- *68 Kontrolní součet

Kromě vět GPRMC posílá přijímač i další (GPGSA – seznam dostupných družic, GPGSV – informace o pozici viditelných družic apod.), které ale asi nebudete využívat.

29.2.13 Pohyb osob (PIR)

Zajímavé čidlo, kterým tento přehled uzavřu, je pasivní infračervené čidlo pohybu (Passive Infra Red – PIR). Čidla PIR jsou nápadná svým zakrytováním – ve skutečnosti jde o Fresnelovu čočku, která zaostřuje infračervené záření z okolí na snímací prvek. Vlastní element je zase polovodičová součástka, která je citlivá na infračervené světlo. To vyvolává změnu napětí, ta je detekována citlivým tranzistorem typu FET a dále zpracovávána.



PIR se používají pro detekci toho, že je nablízku člověk. Lidské tělo totiž vyzařuje, jako každý teplý objekt, infračervené záření. U lidského těla má toto záření vlnovou délku 9,4 μm. jsou tedy často používané pro detekci pohybu osob, ať už vítaného (např. u spínání osvětlení v garáži), nebo nevítaného (poplachové systémy).

Nevýhodou může být přílišná citlivost – pokud je příliš velká, zachytí čidlo i menší živočichy, například kočku či psa. Hodně záleží i na místě instalace čidla – případné proudění teplého vzduchu může čidlo zmást.

30 Meteostanice

30 Meteostanice

Jako bych to neříkal! Stačí málo, neopatrné slovo, a už se staví meteostanice. Tak pojďme na to, ať to máme z krku...

Použijeme Arduino. Ne že by to nešlo bez něj, ale s ním je to výrazně, asi tak 83 ×, jednodušší.

Co se bude měřit? Navrhuju teplotu, tlak a vlhkost. Výšku a rosný bod necháme stranou.

Naměřené hodnoty budeme zatím posílat po sériovém rozhraní do počítače. Později si upravíte a vylepšíte…

30.1 Výběr součástek

Víte, že každý projekt by měl začínat pečlivým zvážením použitých součástek. Projděte si tedy prosím nabídku výrobců senzorů a vyberte vhodné typy senzorů, které splní všechny požadavky, jaké na meteostanici budete mít. Až najdete vhodné typy, tak pečlivě prozkoumejte, jak se připojují a jak se s nimi pracuje. Pak sestavte prototyp – už máte dostatek informací k tomu, abyste věděli, co si počít s teploměrem se sběrnicí I^2C například…

Ano, takto vypadá správný postup. Až budete vyrábět meteostanice ve velkém, postupujte prosím takto. A taky si k tomu nastudujte mnohem víc věcí, než vám v téhle knize prozradím já. Všechno to spočítejte a udělejte to pořádně.

Ale já vám prozradím něco, za co mě odborníci svorně proklejí. Prozradím vám totiž

30.2 Špinavej trik

Zastávám názor, že vynaložené úsilí by mělo být úměrné očekávanému výsledku, a že věci by měly být funkční a *dobré dostatečně pro zamýšlené použití*. Jinými slovy – když stavím zeď, nebudu si brát mikrometr, abych všechno důkladně změřil. Když budu měřit venkovní teplotu, nebudu kvůli tomu pořizovat ultrapřesný teploměr s přesností 0,01 ˚C. A když si stavím pro radost, tak použiju to, co:

- vyhovuje mým potřebám,
- dá se to dobře koupit,
- už to někde někdo zkusil, použil a napsal o tom.

Tedy v případě meteostanice budu chvíli googlit a zjistím, že lidé nejčastěji používají součástky, co se jmenují DHT11 nebo DHT22 (měří vlhkost a teplotu) a BMP180 (měří teplotu a atmosférický tlak). Zjistím, že pro ně jsou desítky příkladů, jak je zapojit k Arduinu, a taky to, že jsou velmi snadno dostupné v e-shopech jako moduly, které stačí propojit vodičem s Arduinem, a dají se i naskládat na nepájivé kontaktní pole.

Jasně, profesionální zařízení bych takto nestavěl – ale nikdo tu nemluvil o profesionálním zařízení. Od začátku je tu řeč o nadšeneckém kutění pro radost.

Tento přístup vyvolává v některých opravdových odbornících záchvaty agresivního odporu: „je to čuňárna, to není elektronika, to je takové náhodné patlání něčeho, co najdu na webu, ten člověk si říká elektronik, ale ve skutečnosti tomu nerozumí…" Na druhou stranu můžete díky tomuto přístupu rychle vyzkoušet nějakou myšlenku, nápad, ověřit, jestli to bude fungovat, postavit prototyp, a celé to zvládnete za jedno odpoledne.

Důležitou poznámku tu ale udělám: Když studujete na webu nějaké cizí zapojení či kód, asi začnete tím, že to zkopírujete a zkusíte spustit. Ale pak následuje velmi důležitá část, a tu nepřeskakujte: **Snažte se pochopit, co se tam děje a jak to funguje!**

Když autor použije někde tranzistor, ptejte se sami sebe: jakou tam má funkci? Co by se stalo, kdyby… Mohl bych použít nějaký jiný? A když tady čeká 10 milisekund, proč to tak je? Jaký měl důvod, že čeká? A můžu to zapojit na jiné piny?

Klást si otázky a odpovídat na ně je to nejcennější, co můžete s cizími příklady udělat. Tupě okopírovat to dokáže i cvičená opice. Já vás ale prosím, apeluju, naléhám na vás a žádám vás: **Nedělejte to!** Vždy berte nějaký hotový příklad jako inspiraci pro vlastní experimenty. Do cizích knihoven se podívejte, abyste viděli, co se děje pod kapotou. Držte se tohoto přístupu, a věřte mi: naučíte se spoustu věcí, a bude to zábavné učení.

Ostatně, i sami profesionálové staví také z ověřeného – když je potřeba zesílit audiosignál, použijí „zapojení první volby", tedy tranzistor se společným emitorem, vyzkouší, ověří, a pokud nevyhovuje, tak teprve vymýšlejí jiná důmyslná zapojení.

30.3 Stavíme z polotovarů

To byl takový morální apel, a teď se vraťme k meteostanici.

DHT22 i BMP180 jsou součástky, které jsou dostatečně popsané, zdokumentované, je k nim dostatek knihoven… Vy obvykle začínejte prosím s tím, že si zkusíte připojit jednu z nich, podívat se na příklady, vyzkoušíte, že vám to funguje, popřípadě proč ne, pak totéž s druhou…

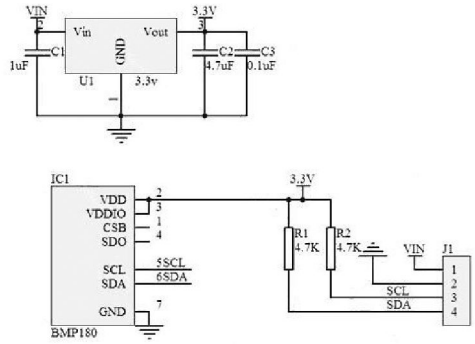
Teď si to zkusme ale jinak. Třeba s tím barometrem BMP180. Když se na modul s touto součástkou podíváte, najdete většinou čtyři vývody: Vcc, GND, SDA a SCL. Co to znamená?

Vcc a GND je jasné, to je napájení. SCL a SDA napovídá, že se bude zařízení připojovat přes sběrnici I²C. Najděte si datasheet pro BMP180 a ověřte si, že to je opravdu tak.

V datasheetu si všimněte, že napájecí napětí je 3,3 voltů. To by mohl být problém – jenže naštěstí moduly s BMP180 obsahují stabilizátor napětí – je to ta miniaturní černá součástka se třemi vývody:



Letmým googlením najdete i schéma takového modulu:



Vidíte, že je tam stabilizátor, který sníží napájecí napětí 5 V z Arduina na požadované 3,3 V. Pak vidíte, že jsou použité i pull-up rezistory na 3,3 V pro sběrnici I²C. Budeme ji moci připojit k Arduinu, když víme, že to používá 5 V?

No, vzpomeňte si na popis sběrnice I²C: logická 0 je zajištěná tím, že zařízení připojují jednotlivé linky k zemi, logická 1 je zajištěná právě pomocí pull-up rezistorů. Taky víte, že 3,3 V je dostatečné napětí pro logickou 1 i pro zařízení, pracující s pěti volty, takže v tomto ohledu je to bezpečné.

Může se tam někudy dostat + 5 voltů? Teoreticky ano, pokud bude na téže sběrnici připojené zařízení s pull-upy na + 5 V, nebo pokud budou zapojené interní pull-upy. Jinak ne.

V datasheetu je dále popsána přesně funkce celého obvodu – jak se adresuje, kde jsou jaké informace, jak se zadávají kalibrační údaje a spousta dalších věcí. A to je přesně okamžik, kde vám doporučím nevynalézat kolo a použít hotovou knihovnu. Vhodné jsou knihovny od výrobců takových modulů – nejčastěji Sparkfun nebo Adafruit.

Pokud najdete knihovnu BMP085, nebojte se – to je starší verze téhož, a v popisu většinou najdete, že umí i BMP180

Součástí těchto knihoven bývá i příklad měření. Podívejte se do něj – zjistíte, že se tam pracuje s aktuální nadmořskou výškou – tedy s tou vaší. Proč? No protože změřený tlak je ten reálný, co je okolo vás, ale aby se tlaky snáze porovnávaly, přepočítávají se na hladinu moře. Přepočet je jednoduchý – počítá se, že na každý metr nadmořské výšky připadá pokles tlaku o 8 Pa (jen pro jistotu: standardní tlak 1013,25 hPa = 101,325 kPa = 101325 Pa).

Například knihovna Adafruit BMP085 Library obsahuje ukázku kódu:

```
#include <Wire.h>
#include <Adafruit_BMP085.h>
Adafruit_BMP085 bmp;
void setup() {
    Serial.begin(9600);
    if (!bmp.begin()) {
        Serial.println(„Could not find a valid BMP085 sensor, check wiring!“);
        while (1) {}
    }
}

void loop() {
    Serial.print(„Temperature = “);
    Serial.print(bmp.readTemperature());
    Serial.println(„ *C“);

    Serial.print(„Pressure = “);
    Serial.print(bmp.readPressure());
    Serial.println(„ Pa“);
    delay(500);
}
```

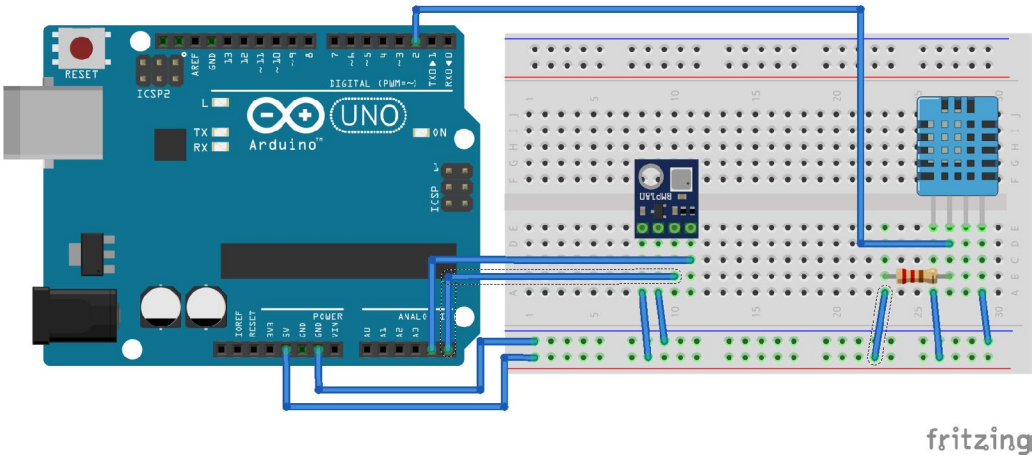
V příkladu není použitý přepočet na hladinu moře, proto se nedivte, když výsledky budou výrazně nižší, než očekáváte. Můžete si výsledek přepočítat pomocí výše uvedeného vzorce, nebo použít funkci bmp.readSealevelPressure(altitude), kde altitude je nadmořská výška v metrech.

Kde zjistit svou nadmořskou výšku? Můžete použít třeba GPS, ale nejjednodušší způsob je ten, že se podíváte na mapu, tam bude u vaší pozice uvedena i nadmořská výška! Nebo zkuste https://api.mapy.cz/view?page=altitude

Senzor dokáže změřit i nadmořskou výšku. Chcete vědět, jak to dělá? No, změří tlak, a předpokládá, že na hladině moře je standardních 101325 Pa. Což bývá málokdy, protože tlak se mění s počasím, rozdíl může být třeba 3 kPa během několika dní, a takový rozdíl představuje přes 350 metrů výškového rozdílu.

U pevné meteostanice bude lepší nastavit výšku odečtem z mapy natvrdo do kódu.

Druhý polotovar, DHT11 nebo DHT22, je nejjednodušší připojit podle standardního zapojení a využít hotovou knihovnu. Když se podíváte do datasheetů, zjistíte, že komunikace s těmito senzory není úplně jednoduchá, a pokud se vám zrovna nechce znovuvynalézat druhé kolo (a komu by se chtělo, když může ušetřený čas věnovat něčemu zajímavějšímu, než je opakování cizích chyb?), použijte ji.



Možné zapojení celé meteostanice ukazuje obrázek. Obvod DHT je připojen podle příkladu z knihovny DHT Simple na pin 2, snímač BMP180 je zapojen na sběrnici I²C. U DHT11 je zapojený i pull-up rezistor 10k.

Když oba příklady spojíme do jednoho, získáme tak kód, který bude vypisovat tlak, vlhkost a dvě různé teploty (jedna je z BMP180, jedna z DHT11). Můžete si vybrat, které budete spíš věřit, nebo je prostě zprůměrovat (anebo použít samostatný teploměr).

Mimochodem, měření teploty není tak triviální, jak vypadá. Teplotu ovlivňuje spousta věcí – například nezapomínejme, že součástky také „topí". U Arduina to není tak extrémní, ale třeba ESP8266, o němž se ještě budeme bavit, dokáže velmi pěkně ohřívat okolí, i o několik stupňů. Teploměr by samozřejmě neměl být na přímém slunci, měl by být daleko od ploch, které mohou sálat, měl by okolo něho proudit volně vzduch, …

No a když máte data, co s nimi? Zatím si je můžete zpracovat v počítači, nebo je můžete ukládat na SD kartu (o tom jsme se už bavili…) Později si tohle celé vylepšíme o posílání dat na server…

Schéma a zdrojový kód najdete na ⌀ https://eknh.cz/meteo

31 Bezdrátový přenos dat

31 Bezdrátový přenos dat

Bezdrátový přenos můžete zařídit několika různými způsoby. Většinou se pod pojmem „bezdrátový" myslí nějaký přenos pomocí radiových vln, i když technicky vzato do bezdrátového přenosu dat patří i třeba infračervený (znáte z vašich televizí) nebo ultrazvukový. Ale teď se chci věnovat radiovému přenosu.

Já se v této knize záměrně nechci věnovat vysokofrekvenční elektrotechnice – to je tak složitý a komplexní obor, že ho rád nechám stranou. To by ta kniha měla dvojnásobný rozsah…

Naštěstí můžete používat i věci, u nichž neznáte princip fungování. A pokud vám někdo říká, že ne, nevěřte mu, protože to byste nemohli ani zapnout automatickou pračku. Pro začátek stačí vědět, jak to použít a jaké jsou pravidla a limity.

Rádiové vlny patří, stejně jako třeba rentgenové paprsky, mikrovlny v troubě, světlo nebo ultrafialové záření mezi elektromagnetické vlnění. Možná vás to překvapí, ale princip rádiových vln a třeba světla je obdobný, liší se pouze svou frekvencí.

Někdy se setkáte také s označením „dlouhá vlna, krátká vlna" apod. Má to spojitost s šířením elektromagnetických vln a jejich frekvencí. Délka vlny je technicky vzdálenost, kterou elektromagnetické vlnění urazí během jednoho „kmitu". Čím vyšší kmitočet, tedy čím víc kmitů za sekundu, tím je kratší vlna. Elektromagnetické vlnění s frekvencí 299 MHz má tedy délku vlny zhruba 1 metr, rádiové vysílání v pásmu FM (87 – 108 MHz) má vlny délky (přibližně) 2,7 až 4 metry.

Viditelné světlo má frekvenci tak vysokou, že by bylo už nepraktické udávat ji v hertzech (přesněji v petahertzech, 10^{15}), místo toho se používají praktičtější nanometry (zhruba 390-760 nm je pásmo viditelného světla, červené má vlnu nejdelší, tedy frekvenci nejmenší, směrem k fialové délka klesá, frekvence roste).

Zůstaneme ale u rádiových vln. Udělat vysílač, který bude vysílat rádiové vlny, není těžké. Stačí k tomu kousek drátu a přesně vyladěný oscilátor. Jenže vysílání není jen tak. Protože rádiové vlny využívá televize, rozhlas, záchranáři, armáda a kdoví kdo ještě, jsou pečlivě rozdělené do pásem a je určeno, kdo smí jaké frekvence používat. Pokud chcete provozovat rádiový vysílač, musíte mít přidělenou frekvenci. Tím je zajištěno, že nebudete nikoho rušit a nikdo nebude rušit vás. Když budete vysílat bez přidělené frekvence, poměrně brzy si na vás patřičné úřady posvítí.

Naštěstí existuje několik frekvenčních pásem, v nichž může používat rádiové vysílání každý, i bez patřičné licence, pokud dodrží některá pravidla, především maximální vysílací výkon. Říká se jim „rádiové kmitočty vymezené všeobecným oprávněním", a vy si prosím pamatujte, že to jsou hlavně tato pásma:

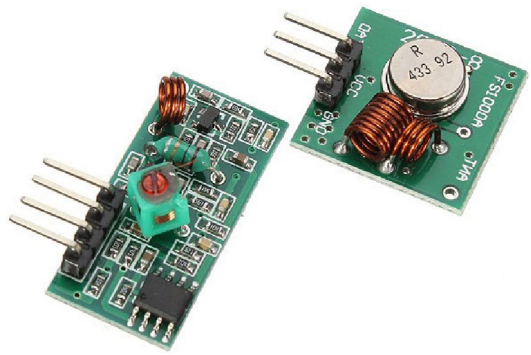
- 27 MHz pro „CB stanice“ (Citizen Band – občanské pásmo),
- 433 MHz pro přenos dat,
- 868 MHz pro přenos dat (v USA se používá 911 MHz, pozor na to!),
- 2,4 GHz a 5 GHz pro přenos dat (např. WiFi).

Na těchto frekvencích byste neměli mít s homologovaným přístrojem, to podtrhuju, žádný problém. Frekvence 433 a 868 jsou nejčastější frekvence, které můžete pro vlastní konstrukce použít. Doporučuju nevynalézat kolo a nesnažit se navrhnout vlastní vysílací a přijímací modul! Kupte si hotové a existující, které dodržují frekvenci i maximální výkon.

Počítejte s tím, že povolený výkon vám postačí pro pokrytí oblasti v řádech desítek metrů na volném prostranství. Je to proto, abyste svým provozem nerušili ostatní. Rádiová frekvence je holt sdílený statek, něco jako společná místnost: hlasitý hovor dvou lidí bude rušit váš hovor s někým dalším. Proto se tato pásma dělí do přesnějších *kanálů*… Například v pásmu 433 MHz má kanál 1 frekvenci 433,075 MHz, kanál 2 má frekvenci 433,100 MHz, kanál 3 má 433,125 MHz atd., až kanál 69 má frekvenci 434,775 MHz. Teoreticky byste měli mít možnost naladit vysílač i přijímač na přesný kanál v rámci daného pásma a snížit tak pravděpodobnost, že budete kolidovat například s bezdrátovým zvonkem u sousedů, nebo s ovladačem jejich garážových vrat. Praxe je bohužel ale zase mnohem pestřejší než teorie, protože velké množství hlavně levných zařízení vysílá „jak se dá“, a hlavně v městech jsou tato pásma velmi zarušená vším možným, co na těchto frekvencích vysílá. Kdejaké bezdrátové teploměry, zvonky, ovladače kdečeho…

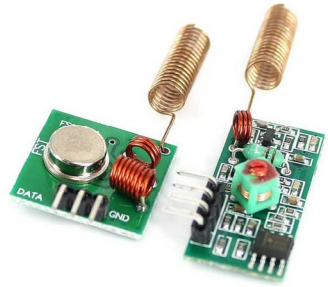
31.1 Vysílání na 433 MHz

A teď do toho přispějete i vy. Dvojice modulů vysílač-přijímač na 433 MHz za pár korun vypadá takto:



Vlevo přijímač (4 piny), vpravo vysílač (3 piny). Ještě k nim doporučím připojit anténu – bez ní bude spojení velmi špatné a dosah jen několik centimetrů. Staré radioamatérské přísloví říká, že dobrá anténa je lepší než sebesilnější vysílač – a je to tak.

Anténu si můžete buď vyrobit z kusu drátu (měla by mít délku 17,3 centimetrů, nebo 34,6 centimetrů – čtvrtina, respektive polovina vlnové délky), anebo ji můžete koupit. Popřípadě rovnou shánějte moduly i s anténou.



Oba moduly mají klasické napájecí piny GND a Vcc, a jeden datový (u přijímače jsou to ty dva prostřední piny – jsou uvnitř spojené, takže jako by to byl jeden).

V ideálním případě by to vše mělo fungovat tak, že když přivedete na vstup DATA vysílače logickou 1, vysílač začne vysílat radiové vlny 433 MHz. Přijímač je přijme a pošle na svůj datový výstup taky 1. Tak tedy zní teorie. V praxi to funguje tak, že přijímač chytá spoustu signálů na stejné frekvenci, je velmi zarušený, a vy v tom šumu musíte najít svoje data. Dražší a lepší přijímače spoustu té černé práce odvedou za vás; bez nich vám nezbývá nic jiného, než navrhnout dobře přenosový protokol, tak, aby minimalizoval chyby. Tedy se synchronizačními pulsy, s kontrolními součty, po malých dávkách atd.

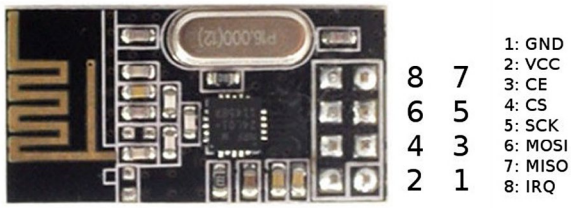
Můžete si zkusit takový protokol navrhnout a otestovat. Nebo můžete zkusit použít nějaký existující a ověřený – doporučím LightWaveRF, pro který jsou knihovny pro Arduino… Ale pravděpodobně zvolíte třetí cestu…

Schéma a zdrojový kód najdete na ⌀ https://eknh.cz/433

31.2 nRF24L01+

Kryptická zkratka označuje čip od Nordic Semiconductor, který v sobě integruje radiovou část na 2,4 GHz a základní logiku řízení síťového provozu. Připojuje se k řídicímu procesoru (například k Arduinu) přes sběrnici SPI a funguje jako transceiver. Toto slovo je složeno ze dvou slov:

transmitter (vysílač) a receiver (přijímač). *Ani nechci vymýšlet český překlad – vysjímač?* Zkrátka umí data posílat i přijímat.

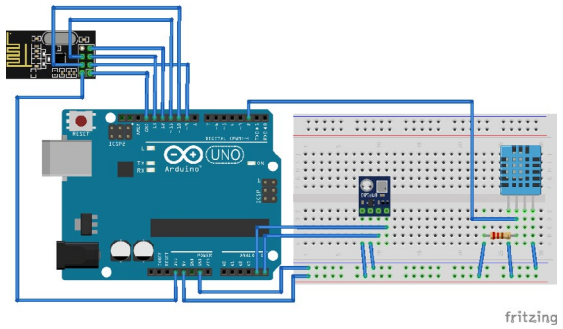


Opět existuje celá řada čínských klonů, které obsahují jiný čip – nejen s mnohdy horšími parametry, ale především i s jiným komunikačním protokolem. Naštěstí existují spousty knihoven, které tyto starosti vyřeší za vás (doporučím RF24). Jen asi nebudou fungovat dohromady dva různé klony, popřípadě klon a originál. Proto doporučím si připlatit za originál.

Každý modul může mít v jeden čas otevřeno až šest komunikačních kanálů (rour – „pipes“) a komunikovat s šesti dalšími moduly. Každá roura má svoje jméno – to má 5 bajtů. Všechny roury v jednom modulu musí mít jména, která se v prvních čtyřech bajtech shodují. Tedy například 0xDEADBEEF00, 0xDEADBEEF01 atd. Na jméno roury můžeme hledět jako na adresu spojení mezi dvěma moduly. K této adrese se odkazujete při volání vysílacích funkcí apod.

Při komunikaci si v jednom modulu otevřete rouru pro zápis se jménem (třeba) 0xDEADBEEF01, v druhém modulu (s druhým Arduinem, samozřejmě) si otevřete rouru pro čtení se stejným jménem. Na přijímači použijete funkci startListening(). Funkce isAvailable() vám řekne, jestli jsou dostupná data, a pokud ano, přečtete je metodou read(). Na vysílači použijete funkci write(). Pokud nenastavíte menší hodnotu, můžete jedním vysíláním přenést až 32 bajtů dat.

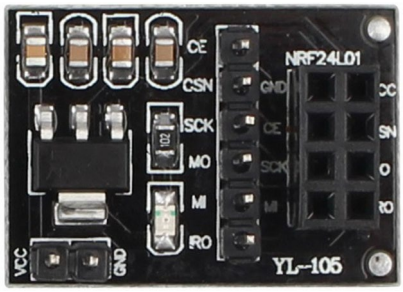
Což je dobrá zpráva. Meteostanice s dvaatřiceti bajty dat udělá nemálo parády, tam se vejde velké množství informací. Vlastně stačí přidat jen nRF24L01+ k našemu „meteoArduinu“…



Připojení je trošku nepřehledné, tak si pamatujte:

CE (3)	Pin 9
CS (4)	Pin 10
SCK (5)	Pin 13
MOSI (6)	Pin 11
MISO (7)	Pin 12

Tip: Pro nRF24L01 se prodává malý adaptér, na němž je stabilizátor 3,3 V a omezovací rezistory pro datové signály. Doporučuju jej použít, pokud připojujete modul k Arduinu.



Na protější straně může být další Arduino a nRF24L01, které bude data číst a posílat po sériové lince do PC. Můžete použít klidně i Raspberry Pi nebo Turris Omnia, nRF24L01 připojit k němu a data rovnou zpracovávat. Později, v kapitole o displejích, si ukážeme zapojení jednoduchého grafického displeje k Arduinu – pokud připojíte i nRF24L01, získáte tak docela slušný displej pro meteostanici.

Schéma a zdrojový kód najdete na ⌀ https://eknh.cz/rf24

32 Procesory, počítače, mikrořadiče

32 Procesory, počítače, mikrořadiče

Od hradel jsme prošli přes kombinační obvody a klopné obvody až k čítačům, registrům a pamětem. Teď se dostáváme k poslední velké oblasti číslicové techniky, k procesorům a počítačům.

Po zkušenosti z předchozích kapitol už jistě tušíte, že ani procesory nebudou nic mystického a tajuplného. A ono taky ne – jde jen o kombinaci toho, co už známe. Nevěříte? To byste měli. Existují totiž lidé, kteří si opravdu postavili svůj procesor z těch integrovaných obvodů, s nimiž jsme se seznámili. Ostatně, před vznikem mikroprocesoru ani jiná cesta nebyla.

Co to je vlastně procesor? Obecně jde o součást systému, která umí několik věcí:

1. Přečte z připojené paměti nějakou binární hodnotu.

2. Podle této binární hodnoty vykoná instrukci, která spočívá buď v přesunu dat odněkud někam, nebo v jejich zpracování v aritmeticko-logické jednotce.

3. Posune se k další adrese v paměti programu.

4. Řídí provádění těchto instrukcí a ovládá připojená zařízení.

K tomu potřebuje několik základních částí:

- Aritmeticko-logická jednotka (ALU). S ní jsme se už seznámili v kapitole o kombinační logice.

- Registry pro udržení dat, s nimiž se pracuje. Asi vás nepřekvapí, že jde o naše staré dobré známé klopné obvody D.

- Čítač instrukcí (PC – Program Counter). Technicky jde opravdu o jeden mnohabitový čítač s možností nulování a přednastavení hodnoty, tedy něco podobného, jako umí obvody 74193.

- Dekodér instrukcí. To je zase jen kombinační obvod, který podle kódu instrukce aktivuje požadované součásti v požadovaném pořadí.

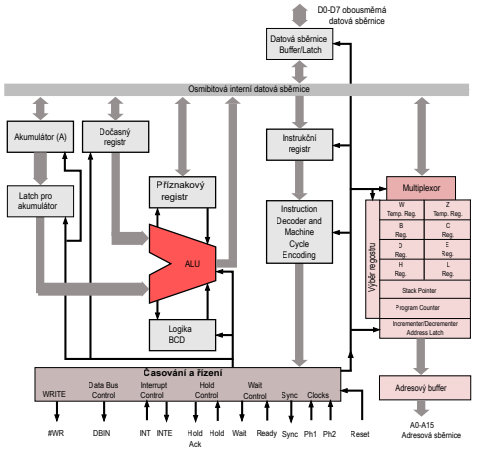
Procesory se původně stavěly z mnoha desek, plných tranzistorů, později z méně desek, plných integrovaných obvodů, později, jak se technologie zlepšovala, klesal počet nutných integrovaných obvodů, až počátkem 70. let 20. století pokročila technika natolik, že umožnila naprostou většinu funkce procesoru integrovat do jednoho nebo do několika málo pouzder. Zrodil se mikroprocesor.

Za úplně první mikroprocesor je považován obvod 4004 firmy Intel. Byl vyvinut pro kalkulátory. Brzy po něm přišla rozšířená verze 8008, a po něm první úspěšný mikroprocesor Intel 8080. U něj

se zastavíme podrobněji. Jednak proto, že z něj se postupnou evolucí vyvinuly šestnáctibitové procesory řady 80 × 86 (a z nich pak Pentia a Core a další dodneška používané mikroprocesory), jednak proto, že se na něm dají dobře ukázat základní principy funkce těchto obvodů, a v neposlední řadě i proto, že v osmdesátých letech byl tento procesor používaný ve většině tehdejších československých mikropočítačů (protože jeho klon vyráběla československá TESLA pod označením MHB8080A).

32.1 Mikroprocesor 8080A

Původní 8080 byl záhy nahrazen vylepšeným typem 8080A, který byl rychlejší. Jeho vnitřní uspořádání ukazuje následující schéma:



CC-BY-SA, autor Appaloosa

Uprostřed vidíte aritmeticko-logickou jednotku (ALU). Nalevo od ní (samozřejmě že je řeč o tom schematickém náčrtku) je pracovní registr – akumulátor a registr pro ukládání mezivýsledků. Nad ALU je blok příznakových bitů (Flag Flip Flops), pod ní kombinační logika pro převod mezi binárními a decimálními hodnotami. Napravo je registr pro instrukci a instrukční dekodér. Pravá část schématu obsahuje blok pracovních registrů, čítač instrukcí PC a další registry. Kromě těchto částí obsahuje procesor ještě obvody pro řízení datové sběrnice, pro řízení adresové sběrnice a obvody časování a řízení běhu procesoru.

Procesor je osmibitový, to znamená, že pracuje s údaji o šířce 8 bitů. Navenek komunikuje obousměrnou datovou sběrnicí D0-D7. Směr komunikace určuje procesor podle toho, jestli potřebuje

číst data, nebo je zapisovat. Procesor dokáže adresovat 64 kB paměti (k tomu sloužila adresová sběrnice s šestnácti signály A0-A15).

Z hlediska programátora jsou zajímavé právě pracovní registry B, C, D, E, H a L, registr A (zvaný akumulátor, to protože se v něm „akumulují" výsledky), ve kterém se provádějí všechny aritmetické i logické instrukce, ukazatel zásobníku a registr příznaků, který říká, jak dopadla poslední operace, a který slouží k podmíněným skokům (například „skoč, pokud výsledkem předchozí operace je 0").

Obvod 8080A se nejčastěji používal s dvojicí podpůrných obvodů 8224 a 8228. První jmenovaný se staral o správné vytváření hodinových pulsů (8080A vyžaduje dva nepřekrývající se hodinové pulsy) a synchronizaci signálů, druhý jmenovaný se staral o řízení sběrnice a vytváření základních řídicích signálů /MEMR, /MEMW, /IOR a /IOW.

Signály /MEMR a /MEMW říkají, že procesor chce číst z paměti (MEMory Read) nebo do ní zapisovat (MEMory Write). Signály IOR a IOW oznamují, že procesor chce číst nebo zapisovat do obvodů pro vstup a výstup. Takovéto rozdělení umožňovalo využít celý prostor 64 kB pro paměť, a další prostor 256 adres pro různé periferie (k nim se ještě dostaneme).

Po spuštění procesoru se většinou vnější obvody postaraly, aby ze všeho nejdřív přišel signál RESET. Tento signál uvedl procesor do výchozího stavu, což technicky znamenalo především nastavit čítač instrukcí (PC) do stavu 0x0000. Pak začal procesor provádět instrukce.

První hodinový puls se nazýval FETCH a jeho úkolem bylo vyzvednout kód instrukce z paměti. Řídicí logika v tomto cyklu přepne datovou sběrnici na vstup, na adresovou sběrnici pustí hodnotu čítače instrukcí PC a aktivuje signál /MEMR – čtení z paměti. Kdesi venku, mimo procesor, musí být připojená paměť, většinou EPROM nebo RAM, a v ní uložený program. Paměť je připojena na adresovou i datovou sběrnici, a povolovací vstup /CE je připojen právě na /MEMR. Když procesor tento signál aktivuje, paměť na výstup vystaví hodnotu, kterou má uloženou na dané adrese – v tomto případě na adrese 0000. Technicky je to prostě jen osmibitová hodnota, například 0xD3 – binárně 11010011.

Procesor tuto hodnotu vidí na datových vstupech a uloží si ho do registru instrukcí. To je zase náš známý osmibitový klopný obvod typu D. Na jeho výstupu je připojen velmi složitý kombinační obvod, který z těchto osmi bitů pozná, co má udělat, a podle toho v dalších cyklech aktivuje jednotlivé vnitřní komponenty. Například na vstupy ALU připojí akumulátor a některý z pracovních registrů, na řídicí vstup ALU nastaví operaci „sečti", a v dalším kroku uloží výsledek opět do akumulátoru. U některých instrukcí si může vyžádat další čtení z paměti nebo zápis do ní. Když vykoná procesor vše, co daná instrukce předepisuje, zvýší hodnotu PC o 1 a celý cyklus se opakuje. U některých instrukcí (skoky) se PC nezvyšuje, místo toho je do tohoto registru zapsána nová hodnota.

32.1.1 Ready / Wait

Některé periferie, popřípadě pomalé paměti, vyžadují, aby procesor chvilku počkal, protože nemají požadovaná data hned. V takovém případě uvedou vstup READY – tedy „připraveno" – do 0. Procesor z toho pozná, že data nejsou připravena, a čeká. Čeká do té doby, než bude signál READY zase 1. Během čekání uvede svůj výstup WAIT do 1, a tím oznamuje, že čeká na data. Jakmile má periferie data připravená, pošle je na datovou sběrnici a uvolní signál READY zpět do log. 1. Procesor tím vystoupí z čekací smyčky, přečte data a pokračuje dál v provádění instrukce.

32.1.2 Hold (DMA)

Procesor umožňuje některým periferním obvodům, aby převzaly kontrolu nad sběrnicí. Slouží k tomu signál HOLD. Jakmile je aktivovaný, dokončí procesor aktuálně prováděný cyklus a vstoupí do stavu HOLD. V tomto stavu se odpojí od adresové i datové sběrnice a aktivuje výstup HLDA (HOLD Acknowledge). Jakmile je HLDA aktivní, může periferní obvod přebrat řízení sběrnice.

Nejčastější scénář použití stavu HOLD je při takzvaném přímém přístupu do paměti (Direct Memory Access, DMA). Používá se ve chvíli, kdy některá rychlá periferie (nejčastěji to bývaly disky, zvukové obvody nebo videoobvody) potřebuje zapisovat nebo číst data. Speciální obvod (DMA Controller) obdržel např. informaci o tom, že periferie, například disk, má připravená data k přenosu. Požádal procesor o přístup na sběrnici (HOLD), a jakmile procesor potvrdil, že je odpojen, začal tento obvod číst data z periferie a zapisovat je do paměti mnohem větší rychlostí, než by to dokázal samotný procesor.

Práce s takovým obvodem pak pro programátora vypadala tak, že do DMA Controlleru nastavil adresu, kam se budou data zapisovat nebo odkud se budou číst, řekl kolik dat očekává, a oznámil periferii, že požaduje např. data z nějakého diskového sektoru. Pak pokračoval v práci. Ve chvíli, kdy periferie měla data připravená, přenesl je výše popsaným způsobem DMA Controller do paměti, a pak dal procesoru vědět, že je hotovo.

Jakým způsobem? Třeba přerušením!

32.2 Přerušení

Přerušení je velmi podstatný koncept v procesorovém světě. Jde o způsob, jakým dává okolí vědět procesoru, že se stalo něco, na co je třeba bezprostředně zareagovat. Může to být například informace o tom, že na klávesnici někdo stiskl klávesu, může to být signál, že přišla data po sériové lince, nebo může jít o pravidelný hodinový signál. Může jít o naprogramovaný signál typu „za 10 milisekund", nebo může jít třeba o výše zmíněný signál od obvodu DMA.

U procesoru 8080 slouží k vyvolání přerušení vstup INT. Na konci každé instrukce procesor testuje stav tohoto vstupu. Pokud je aktivní, pozná, že jde o přerušení. Pokud není přerušení zakázáno (což může programátor zařídit speciální instrukcí), udělá to, že přečte jednu instrukci z datové sběrnice, zakáže další přerušení a instrukci vykoná.

Tou jednou instrukcí bývala instrukce RST0 až RST7 – speciální obvod se staral o to, aby různé požadavky na přerušení měly různé instrukce. Instrukce RST jsou vlastně zkrácené instrukce volání podprogramu na adresách 0, 0x8, 0x10, 0x18, … 0x38.

Pro jednoduché systémy šlo zařídit, aby vždy přerušení vyvolalo instrukci RST7, tedy volání podprogramu na adrese 0x0038. Po příchodu požadavku na přerušení tedy procesor skočil do podprogramu na adrese 0x0038, tam musel programátor požadavek vhodně obsloužit, pak opět přerušení povolit a vrátit se zpět do hlavního programu.

32.2.1 Nemaskovatelné přerušení

Jiné procesory, třeba známý Z80, mají dva různé typy přerušení – maskovatelné a nemaskovatelné. Maskovatelné přerušení odpovídá tomu, co známe z procesoru 8080 (v procesoru Z80 máme k dispozici dokonce tři módy, což teď není podstatné).

Maskovatelné přerušení může programátor zakázat – „zamaskovat" – pomocí speciální instrukce DI (Disable Interrupt). Tato instrukce nastaví vnitřní příznak v procesoru (technicky: zase jeden klopný obvod). Vstup /IRQ (Interrupt Request) je tímto příznakem „maskován" – pokud je přerušení zakázáno, signál se nedostane dál a požadavek je ignorován. „Odmaskovat" přerušení lze instrukcí EI (Enable Interrupt).

Kromě maskovatelného přerušení /IRQ existuje i signál nemaskovaného přerušení /NMI (Non Maskable interrupt). Tento signál vždy vyvolá přerušení, bez ohledu na instrukce DI/EI.

32.3 Periferie

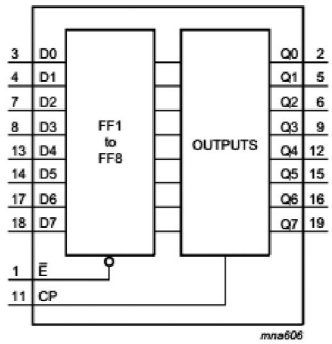
Už několikrát to tu padlo – periferie. Co to vlastně je? Možná napoví český výraz **vstupně/výstupní obvody** (ostatně i v angličtině se označují jako I/O, tedy Input/Output). Technicky to jsou obvody, které jsou připojené na sběrnici procesoru, a které může procesor adresovat, zapsat do nich požadovaná data anebo naopak data přečíst.

Na rozdíl od paměti, která slouží jako velké úložiště, plné buněk, kam se data ukládají a z nichž se čtou, tak periferie mají podstatně pestřejší funkce. Když chcete k počítači připojit klávesnici, je to periferie. Periferie je třeba sériové rozhraní, nebo externí paměť… Periferie jsou všechny čidla

a senzory. Zkrátka vše, co není procesor a pracovní paměť.

Technicky jsou tyto periferie připojené přes oddělovací obvody, které se postarají, aby na výstupech byla správná data, nebo naopak aby ta vstupní šla do systému pouze tehdy, když si o ně procesor řekne.

V roli takového nejjednoduššího obvodu si můžeme představit třeba osmici klopných obvodů typu D, například 74HCT377. Tento obvod obsahuje osm registrů D se společným hodinovým vstupem CP a společným povolovacím vstupem /E. Pokud je /E neaktivní (=1), zůstává obvod stále ve stejném stavu a na výstupech udržuje předchozí zapsanou hodnotu. Pokud je /E aktivní (=0), tak náběžná hrana na hodinovém vstupu CP zapíše data ze vstupů D0-D7 do vnitřních klopných registrů. Odtamtud se pak objeví na výstupech.



32.4 Složitější periferie

V našem příkladu jsme použili jednoduchý obvod a stvořili jsme tak primitivní paralelní výstup. Na ten bychom mohli připojit, dejme tomu, nějaké řízení, nebo sedmisegmentovku, nebo něco takového.

Podobným způsobem, ale pomocí obvodu s třístavovými výstupy, bychom mohli udělat vstupní periferii (například s obvodem 74HCT244). A opravdu se tak dělaly. Ovšem bylo to velmi náročné na obvody i na konstruktéra.

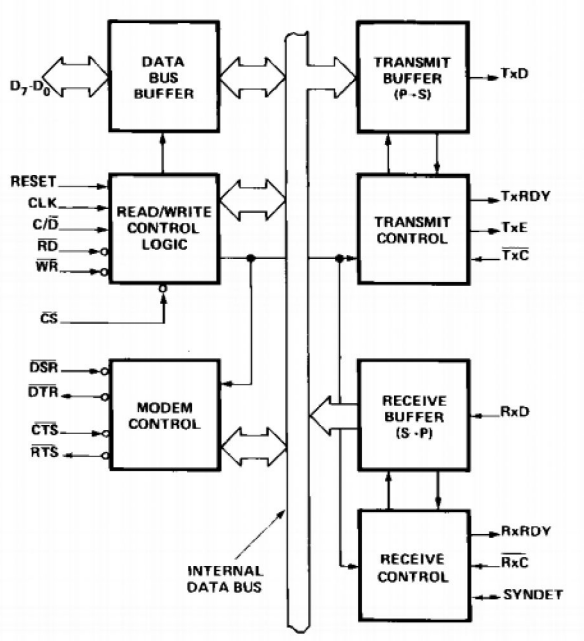
Postupem času přišly složitější periferie, většinou programovatelné. Příkladem mohou být dva obvody, 8255 a 8251.

Obvod 8255 obsahuje tři obousměrné paralelní osmibitové porty PA, PB a PC, proto se označuje jako PIO – Parallel Input and Output. K procesoru se připojuje přes datovou sběrnici a dva

adresové vstupy. Dále obsahuje vstup RESET a vstup /CS – známý Chip Select, který připojuje tento obvod ke sběrnici.

Ta nejzajímavější vlastnost obvodu 8255 je, že je konfigurovatelný. Podle kombinace na vstupech A0 a A1 procesor komunikuje buď s registry jednotlivých portů PA, PB, PC, nebo zapisuje a čte do/z osmibitového řídicího registru CWR. Pomocí zápisu do tohoto registru lze jednak ovládat přímo jednotlivé bity portu PC, ale také nastavit mód práce. Obvod totiž může fungovat jako tři nezávislé datové porty, ale také může rozdělit porty do dvou skupin (PA + polovina PC, PB + druhá polovina PC) a rozdělené signály portu PC využívat jako řídicí signály pro přenos dat, např. pro oznámení, že data byla přijata nebo naopak že je požadováno jejich převzetí apod.

Obvod 8251 se používal v osmibitových počítačích spolu s obvodem 8255. Zatímco 8255 nabízí paralelní rozhraní, 8251 nabízí sériové rozhraní. V kapitole o rozhraních jsme si říkali, že se pro rozhraní RS-232 používaly obvody UART/USART – no a 8251 je právě USART, neboli Univerzální Synchronní / Asynchronní Přijímač a Vysílač (rozuměj: sériových dat).



S procesorem komunikuje tento obvod opět přes datovou sběrnici a pomocí řídicích signálů /RD, /WR. Obvod se připojuje na sběrnici ve chvíli, kdy je aktivní vstup /CE. Vstup C/D určoval, jestli se zapisují nebo čtou data (=0), nebo řídicí / stavové informace (=1). V praxi se připojoval k některému z adresních signálů, např. k A0. Pokud byl obvod adresován například tak, že byl

aktivní na adrese 0x20, a vstup C/D byl zapojen na A0, tak se adresou 0x20 přistupovalo do datového registru, adresou 0x21 do řídicího.

Ve schématu jsou vidět i všechny signály, o nichž jsme mluvili v kapitole o sériové komunikaci. Všechny ty DSR, DTR, CTS, RTS a podobné, které řídí, kdy se může vysílat a kdy ne. Vlastní datový přenos probíhal po linkách TxD, RxD – u synchronního přenosu se používaly i hodiny TxC, RxC.

Dalšími používanými periferiemi byly například čítače a časovače 8253 (nebo Z80-CTC). Tyto obvody obsahují programovatelné čítače, které dokázaly počítat nějaké vnější události, např. hodinové impulsy, a při splnění určité podmínky buď změnily signál na výstupu, nebo např. vyvolaly přerušení.

Mezi periferie se počítaly i zobrazovací jednotky. Některé displeje fungovaly naprosto transparentně, tj. zobrazovaly data, která byla někde v paměti, bez účasti procesoru, jiné bylo potřeba předem přepnout, nastavit do nějakého vhodného pracovního módu – tak fungovaly například videořadiče od Motoroly MC6845 (používané i v grafických kartách MDA a Hercules u IBM PC), nebo např. TMS9918 (výrobce Texas Instruments).

Další periferie, například vnější paměti apod., většinou nemívaly vlastní obvod a připojovaly se přes standardní rozhraní. Výjimkou byly floppy disky (diskety), které měly vlastní řadiče – známé typy Intel 8272, WD2797 nebo WD1793, které se používaly např. v disketových jednotkách, vyráběných pro počítač ZX Spectrum…

32.5 Jednočipový mikropočítač

Sedíte, držíte v ruce Arduino a říkáte si: *Jak ono to funguje? Je tam taky nějaký procesor.*

Jenže, jak ten procesor ví, co má dělat? Musí tam být někde uložený program, v nějaké paměti? A kde je ta paměť?

Že by to bylo všechno v tom jednom obvodu? ATmega328P je tam napsáno… Někde jsem četl, že se tomu říká ***jednočip*** *– co to vůbec znamená?*

Vězte, přátelé, že jste na správné cestě. V minulých kapitolách jsme si ukázali, jak funguje číslicový počítač. Je tam procesor, paměti, periferie… Logický vývoj tedy směřoval k tomu, že výrobci tyto části spojili, vytvořili na jednom čipu a ten strčili do jednoho pouzdra. A protože jsou základní části počítače, tj. procesor, paměti a periferie, na jednom čipu, říká se tomu „jednočipový mikropočítač". Familiárně pak *jednočip*. Když budete hledat materiály v angličtině, nenechte se zmást: neříká se tomu singlechip, tohle slovo je vyhrazené pro něco trošku jiného, ale označení je

microcontroller, v počeštěné podobě pak *mikrokontrolér* či *mikrořadič*.

Mikrokontroléry vznikly jen pár let po prvních mikroprocesorech. Většinou integrovaly existující procesor v nějaké modifikované podobě spolu s pamětí ROM a malou pamětí RAM, a k tomu i několik periferií, nejčastěji paralelní a sériové rozhraní a čítače / časovače. Ze začátku se vyráběly hlavně mikrokontroléry s paměťmi ROM a PROM, později i EPROM. Logicky s vývojem technologie přišly jednočipy s pamětí FLASH, a ty jsou dnes asi nejrozšířenější.

31.5.1 Harvard vs von Neumann

Tady je dobré místo na odbočku. Když jsme se bavili o tom, že paměť obsahuje program, který procesor vykonává, není to tak úplně jednoznačné. Existují dvě hlavní architektury, totiž von Neumannova, kde je operační paměť společná pro program a data, a harvardská, kde jsou paměti pro data a pro program oddělené. U jednočipů je častější architektura harvardská, u počítačů pro všeobecné použití zase von Neumannova.

U jednočipu, třeba toho arduinského ATmega328, je paměť RAM tedy určena pro ukládání zpracovávaných dat a zásobník, kód je v paměti FLASH (tedy de facto ROM). Vzhledem k tomu, že mikrokontroléry jsou primárně určené k tomu, aby dělaly jednu jedinou úlohu, která se po dobu jejich „života" v daném zařízení nijak výrazněji nemění, tak je harvardská architektura docela vhodná. S touto architekturou můžete pro data a pro program zvolit také rozdílné typy pamětí – třeba mikrokontroléry PIC od Microchipu ze základní a střední řady (PIC10 – PIC16) zpracovávají osmibitová data, ale instrukce mají 12, resp. 14 bitů. Paměť FLASH je menší při stejné kapacitě než paměť SRAM, proto je ekonomičtější a výrobci tím snižují náklady.

32.6 Atmel AVR

Mikrokontrolér ATmega328 patří do rodiny mikrokontrolérů ATmega výrobce Atmel (v době psaní knihy už Microchip). Spolu s mikrokontroléry ATtiny, kde jsou mikrokontroléry s menší pamětí, patří do velké rodiny, označované AVR.

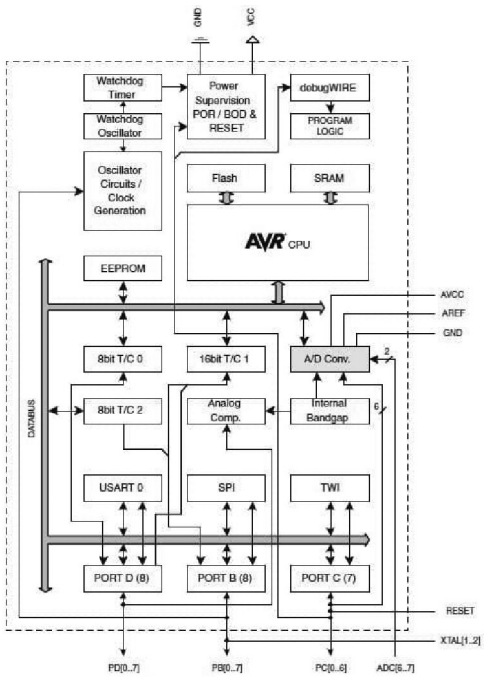
32.6.1 RISC

Jen pokud byste nevěděli – RISC znamená „Reduced Instruction Set Computing", a vychází z poznatku, že většina programů využívá jen část bohaté instrukční sady standardních procesorů (CISC – Complex Instruction Set Computers). RISC proto redukují velké množství instrukcí, ponechávají jen ty základní, a snaží se je provádět co nejrychleji. RISCové stroje zpravidla ani nemívají některé registry vyhrazené pro konkrétní operace, ale bývají ortogonální, tj. jakákoli instrukce může použít jakýkoli registr.

AVR jsou osmibitové procesory s RISCovými rysy – mají omezenou 16bitovou instrukční sadu, 32 osmibitových registrů, s nimiž instrukce pracují, a většina instrukcí se provádí velmi rychle.

Když se podíváme konkrétně na typ, který je použit v Arduinu UNO: kromě procesoru AVR má 32 kilobyte FLASH, 2 kilobyte RAM a spoustu periferií, které se ovšem neadresují speciálním způsobem, ale tváří se, jako by to byly paměťové buňky.

32.6.2 Vnitřní uspořádání ATmega328



Zdroj: Atmel

Na schématu vidíte vnitřní uspořádání čipu ATmega328. Uprostřed je samotný mikroprocesor AVR, spolu s pamětí FLASH a statickou RAM. Ve spodní polovině schématu jsou nakreslené periferie: tři porty (PD, PB a PC), obvod pro sériovou komunikaci USART, obvod, který komunikuje po sběrnici SPI a obvod TWI (Two-Wire Interface), který ovládá sběrnici I^2C (toto označení se nepoužívá, kvůli tomu, že jde o registrovanou známku jiného výrobce). Všechny tyto periferie jsou nám už známé.

Nad nimi se nachází tři čítače / časovače (Timer / Counter), které může programátor využít k nejrůznějším účelům, od počítání událostí po úlohy typu „vyvolej přerušení za X časových pulsů“. Dále je zde připojena malá paměť EEPROM, určená pro ukládání údajů, které mají být k dispozici i po případném výpadku napájení. Sadu periferií uzavírá analogově-digitální převodník, což je komponenta, která umí převést vstupní napětí v určitém rozsahu na desetibitové číslo.

V horní části se nachází obvody Watchdog a BOD. Obvod Watchdog slouží k tomu, aby se jednočip nedostal do nekonečné smyčky, ze které by neměl úniku. Programátor může tento obvod zapnout, a pak musí na různá místa programu dát instrukce, které řeknou watchdogu, že program běží, jak má. Pokud po určený čas nepřijme watchdog takové potvrzení, má se zato, že program zhavaroval, a vyvolá se reset.

BOD (někde zvaný i BOR) je obvod, který hlídá takzvaný Brown-Out stav (Brown Out Detection / Brown Out Reset). Pokud napájecí napětí klesne pod určitou mez, výrobce nezaručí, že obvod bude fungovat správně. Proto obvod BOD takový stav hlídá, a pokud klesne napájecí napětí pod výrobcem stanovenou bezpečnou mez, procesor vynuluje. Zároveň se stará o reset po zapnutí napájecího napětí.

Poslední částí mikrokontroléru ATmega328 je debugWIRE – je to rozhraní, které umožňuje pomocí SPI a signálu RESET uvést procesor do speciálního stavu, kdy je možné zapisovat do interní FLASH. K tomu je potřeba externí obvod – programátor.

U Arduina tento obvod není. Místo toho je využita schopnost zapisovat do FLASH programově. Od výrobce Arduina je v paměti FLASH nahraný krátký program, zvaný **bootloader**. Tento program se spustí po každém resetu, a jeho úkol je prostý: čeká, jestli po sériovém portu nepřicházejí nějaká data. Pokud během určeného časového intervalu nepřijdou, předá řízení nahranému programu. Pokud data začnou chodit, zapisuje je do vnitřní FLASH, a tím vlastně obvod přeprogramuje.

32.7 Další mikrokontroléry

Atmel AVR je velmi široká rodina osmibitových mikrokontrolérů, která zahrnuje maličké obvody ATtiny s pár kilobajty FLASH, často s pouhými osmi vývody, a na druhé straně spektra jsou obrovské čipy s 256 kB FLASH, 8 kB RAM a až 86 datovými piny. Řada AVR má i pokračovatele, AVR32, což je 32bitová architektura, ovšem ta už není tak rozšířená.

Důvodem je to, že v oblasti 32bitových mikrokontrolérů naprosto dominují procesory a mikrokontroléry, založené na jádru ARM.

32.7.1 ARM

Firma Acorn, která se na začátku 80. let proslavila díky výrobě počítače BBC Micro, zachytila včas vlnu inovací, a pro svůj nový počítač Archimedes použila vlastní 32bitový procesor, který pojmenovala Acorn RISC Machine, tedy ARM. Na tehdejší dobu šlo o velmi rychlé a výkonné procesory s architekturou RISC.

Procesory ARM nakonec nezůstaly jen doménou jedné značky jednoho výrobce. Vývoj tohoto procesoru převzal holding ARM Holdings, který zvolil šťastný obchodní model: Soustředí se na vývoj, a vyvinuté modely prodává výrobcům jako duševní vlastnictví. Výrobci tedy nemusí vyvíjet vlastní jádra, namísto toho se mohou soustředit na jiné věci, a mají jistotu, že jejich zařízení „s jádrem ARM" bude mít dostatečnou podporu ze strany softwarových vývojářů. V roce 2013 byl ARM nejpočetnější architekturou na světě a zhruba 60 % všech mobilních zařízení obsahovalo nějaký čip s jádrem ARM.

V současnosti se používají nejčastěji architektury ARMv6, ARMv7, ARMv8 s jádry Cortex.

Tato jádra lze nalézt i v mikrokontrolerech. Typickým zástupcem mohou být mikrokontroléry z řady STM32 od výrobce ST (pro amatérské použití jsou dostupné kity Nucleo) nebo NXP čipy LPC.

32.7.2 PIC

Velmi bohatou historii jednočipů tvoří firma Microchip se svou řadou PIC. Tyto mikrokontroléry byly dlouhá léta nejdostupnějšími jednočipy a prakticky až do nástupu rodiny AVR dominovaly v amatérských konstrukcích. Technicky jsou jejich možnosti srovnatelné s obvody ARM, ovšem programátoři je nemají moc rádi, protože jejich instrukční sada je někdy zvláštní.

32.7.3 8051/8052

Ani Intel nemohl stát stranou. Poté, co vyvinul a úspěšně nabídl řadu 8048 (hned několik modelů, co se lišily podle velikosti a typu paměti) přišel s řadou 8051/8052. I tato řada dodneška žije a používá se v nejrůznějších mikrokontrolérech. Jádro 8051 se u různých výrobců dočkalo vylepšení či zrychlení, a i přes některé podivnosti a nectnosti, jaké toto jádro má, zůstává dodnes používané. Jeho velkou výhodou totiž je, že jej zná velké množství vývojářů a existuje dostatek vývojářských nástrojů.

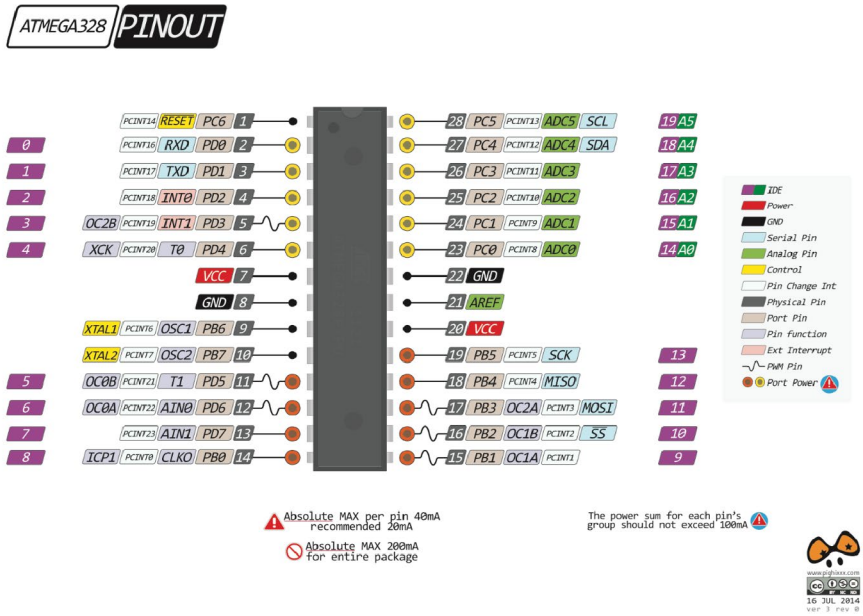
32.8 Tak málo nožiček...

a přitom tolik funkcí, že? Když se díváte na Arduino, tak si říkáte: Převodníky, tři porty, sériové rozhraní, kde to má všechno vývody?

Téměř všechny mikrokontroléry totiž používají jeden trik: jejich vývody mají většinou několik funkcí, a programátor spolu s návrhářem vybírají, jaké funkce použijí. Obvykle platí, že periferie typu sériové rozhraní mají pevně dané vývody, a pokud nejsou použité, tak jejich vývody fungují jako normální I/O porty.

Příkladem mohou být piny 27 a 28 u ATmega328. Na Arduinu jsou připojeny jako analogové vstupy 4 a 5, ale zároveň mohou sloužit jako komunikační piny SCL a SDA pro sběrnici I²C. Pokud ale nepoužíváme I²C a zrovna neměříme napětí na analogových pinech, fungují tyto dva vývody jako datové vývody portu PC, konkrétně PC4 a PC5. Podobné to je s piny 2 a 3 – buď po nich probíhá komunikace přes UART (TxD, RxD), nebo fungují jako piny PD0 a PD1 portu PD.

Obyčejné datové piny mohou pracovat v několika režimech – výstup, vstup, a protože spousta obvodů pracuje s otevřeným kolektorem, kdy je potřeba zapojit pull-up, tak bývá tento rezistor zapojený už přímo v jednočipu a lze ho volitelně zapnout (input pullup).



32.9 Programování jednočipů

Víme, že v jednočipu je paměť FLASH a v ní je uložený program. Víme, že některé konstrukce (Arduino třeba) a některé jednočipy mají k dispozici program, který umožní po startu zkontrolovat sériovou linku, jestli náhodou někdo nepožaduje zápis. (Podobný systém používaly například

mikrokontroléry Dallas Semiconductors, které měly jádro 8051, a kromě standardní paměti měly i zabudovaný program „monitor“, který umožňoval jednoduchou práci po sériové lince.)

Pokud jednočip takovou možnost nemá, je potřeba jej naprogramovat jinak. Možná vás to překvapí, ale třeba u AVR i PIC se tak děje způsobem, který je velmi podobný zápisu do sériové FLASH po sběrnici SPI. Podrobnosti jsou uvedeny vždy v datasheetu, ale většinou je nějakým způsobem využit signál /RESET, který uvede čip do stavu, kdy naslouchá na sběrnici SPI a pokud přijdou správná data, započne programování.

Starší jednočipy potřebovaly, podobně jako paměti EPROM, vyšší napětí, typicky 12 voltů. Toto vysoké napětí se přivedlo na signál RESET a signalizovalo, že bude následovat programování.

Novější jednočipy od tohoto způsobu ustoupily. Důvod byl jednoduchý: díky tomu, že se nepřivádělo napětí 12 V, ale pouze 5 V, mohly čipy za určitých okolností zůstat zapojené v obvodu a být naprogramovány přímo na místě. Tomuto programování se někdy říká In-Place Programming nebo In-Circuit Serial Programming (ICSP / ISP).

Arduino lze dokonce přeprogramovat speciálním firmware tak, aby fungovalo jako programátor pro jednočipy AVR, a to právě v režimu ICSP.

Jiné jednočipy, především ty s jádrem ARM, mají takzvané „JTAG“ obvody. Toto rozhraní funguje podobně jako výše zmíněné ICSP. Navíc mívá toto rozhraní schopnost ovládat procesor za běhu, přerušit jeho práci, přečíst vnitřní stav registrů a zjistit stav na vstupech a výstupech, takže se používá i pro ladění programů.

Další výrobci navrhují další a další způsoby, jak programovat jednočipy a jak ladit programy. Tyto způsoby jsou většinou nekompatibilní, takže pokud budete používat více rodin jednočipů, budete mít zároveň i několik programovacích rozhraní a programátorů, většinou poměrně drahých.

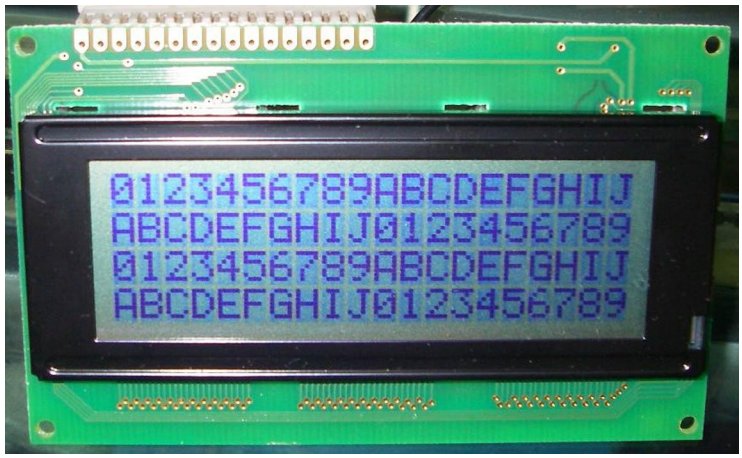
33 Displeje

33 Displeje

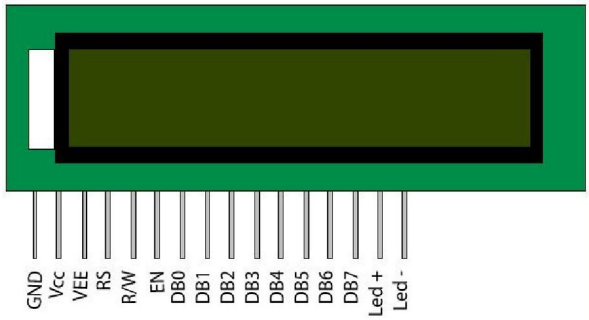
Vlastně až dosud jste používali pro výstup dat buď prostou LED, nebo jednoduchý sedmisegmentový displej, popřípadě jste posílali data na sériové rozhraní. Teď si pojďme ukázat některé vyspělejší displeje.

33.1 Znakový displej 1602, 2004

Dobře známý je displej, pro který se vžilo označení 1602 – 16 znaků, 2 řádky. Určitě jste ho už někde viděli, a když ne tento, tak jeho většího bratříčka 2004 (20 znaků, 4 řádky).



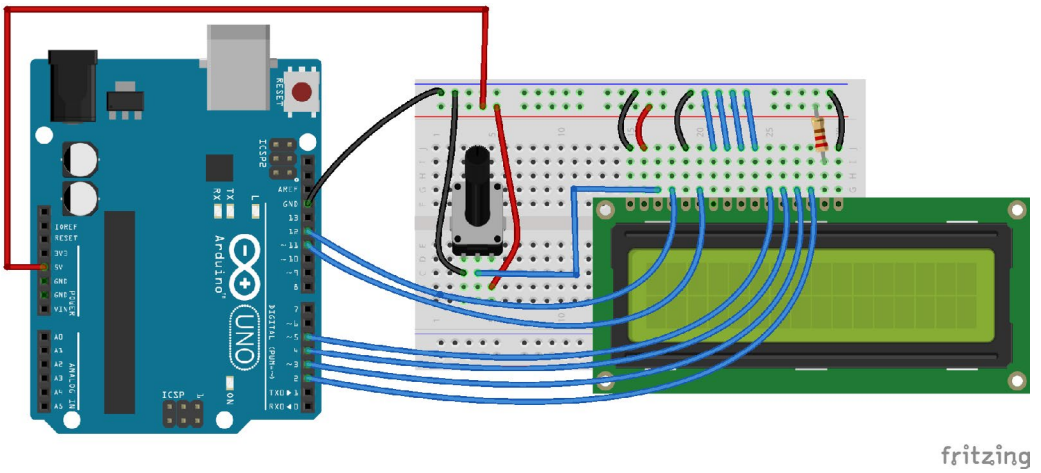
Vyrábí se v různých barevných provedeních, nejčastěji jako černé znaky na žlutozeleném pozadí, nebo jako bílé znaky na modrém pozadí, popř. modré na bílém pozadí. Tyto displeje obsahují řadič typu Hitachi HD44780, pro který existuje řada knihoven.



Skvělé na těchto displejích je, že jsou velmi jednoduché na ovládání – stačí osmibitová datová sběrnice (DB0-DB7), několik řídicích signálů (RS, R/W, EN) a jeden potenciometr pro nastavení jasu (Vee).

Což se lehce napíše a hůř provede, protože Arduino zas tolik digitálních výstupů nemá, a každý přijde vhod. Naštěstí mají tyto displeje možnost pracovat ve čtyřbitovém režimu. Čtyři nižší bity se připojí k zemi, a komunikuje se pouze přes horní čtyři bity (D4-D7). Navíc lze vynechat i řídicí vstup R/W, který říká, zda se z displeje čte (1), nebo do něj zapisuje (0), a napevno určit, že do řídicího obvodu displeje se bude jen zapisovat.

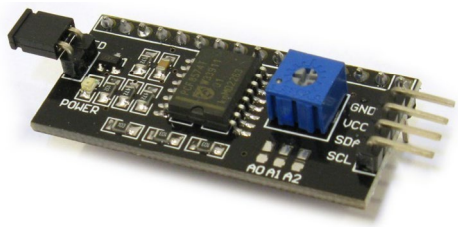
Pak už zbývá jen šest vodičů: čtyři datové, jeden povolovací (EN) a jeden, kterým se říká, jestli se posílá řídicí instrukce (RS=0), nebo znaky ke zobrazení na displeji (RS=1).



Obvod je potřeba nejprve přepnout do čtyřbitové komunikace – pomocí sekvence speciálních příkazů. Jakmile je displej správně nastaven, stačí jen posílat ASCII kódy znaků, které se mají zobrazit, a posílat je po čtyřech bitech, nejprve vyšší, pak nižší. Pomocí některých řídicích kódů lze nastavit, kam se má další znak zapsat, popřípadě vytvořit několik vlastních znaků a používat je (symboly, znaky s diakritikou, …)

Arduino má naštěstí knihovnu, která se jmenuje LiquidCrystal, a ta se postará o vše potřebné.

A kromě knihovny vám může pomoci ještě jeden fígl – hledejte „1602 I²C expander“. Vypadá to nějak takto:



Tento modul je uzpůsobený pro připojení k displeji 1602 nebo 2004. Postará se o správné řízení datových i řídicích signálů, obsahuje trimr pro nastavení jasu, omezovací rezistor pro podsvícení, zkrátka vše, co je k těmto displejům potřeba. A to nejlepší nakonec: displej bude fungovat jako každá jiná periferie na sběrnici I²C! (A samozřejmě i pro tuto variantu existuje knihovna, LiquidCrystal_I2C).

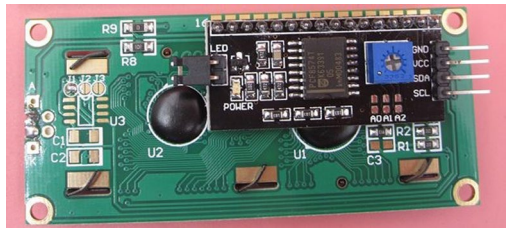
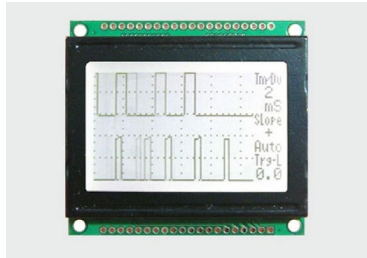


Schéma a zdrojový kód najdete na ⌀ https://eknh.cz/1602

33.2 Grafický displej 12864

Předchozí displeje byly znakové. Sice interně používají matici bodů pro každý znak, ale kromě možností nadefinovat si několik málo znaků „po svém“ nemáte jak zobrazit třeba graf.

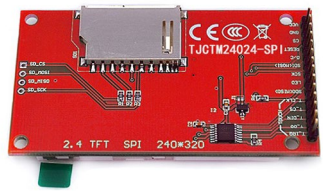
Tuto nevýhodu odstraňují grafické displeje. Pokračovatelem výše uvedených displejů je displej 12864 – číslo udává, že má rozlišení 128 × 64 bodů.



Tyto displeje jsou opět jednobarevné, nejčastěji černá kresba na pozadí podsvíceném bíle nebo zeleně, popřípadě bílá kresba na modrém pozadí… Rozhraní je většinou podobné předchozím modelům, ale některé displeje mají možnost přepnout se do sériového rozhraní (SCK + MOSI). Samozřejmě existují i expandéry – moduly, které tyto displeje připojí na sběrnici I²C.

33.3 Další displeje

Před lety se objevily ve výprodejích velmi levné displeje z mobilních telefonů, například ze starých Nokií 5110 atd. Tyto displeje mívají vlastní řídicí obvod, s nímž se komunikuje buď přes SPI, nebo po I²C. Nadšenci začali tyto displeje připojovat k jednočipům, protože byly levné a snadno řiditelné. Dnes je k dispozici velké množství modulů s takovými displeji, většinou za pár desítek korun (v českých e-shopech za jednotky stokorun), a se širokou škálou možností – s různým rozlišením, s různým rozhraním, barevné i monochromatické, LCD i OLED, někdy i s dotykovou vrstvou… Často je na stejném modulu například i slot pro SD kartu se SPI rozhraním.



K těmto displejům je vždy třeba přistupovat opatrně. Naprostý „noname“ bude asi těžké oživit, pokud nevíte, jaký řídicí obvod používají. Nejjistější je držet se takových, pro které si najdete dostatek literatury, a nejlépe hotových knihoven. Mohu doporučit hledat displeje s řadiči ILI9340 a ILI9341, pro ně je dostatek knihoven, jak pro ovládání, tak i pro kreslení grafických prvků, psaní znaků atd.

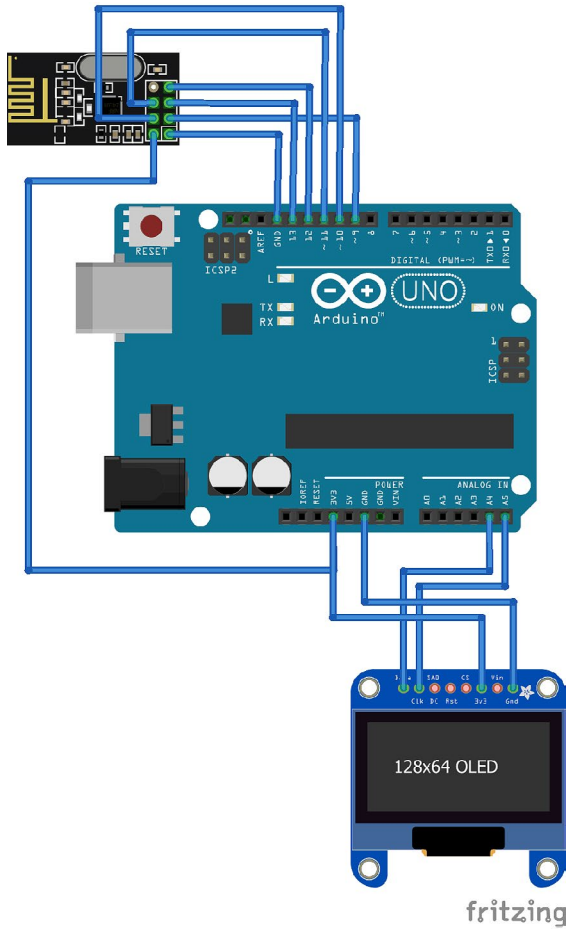
33.4 Bezdrátový displej k naší meteostanici

V kapitole o bezdrátovém spojení jsem slíbil, že si ukážeme, jak připojit malý grafický displej k Arduinu spolu s modulem nRF24L01. Použijeme malý OLED displej s rozlišením 128 × 64 a s rozhraním I²C. V ideálním případě má takový modul jen čtyři vývody:



Někdy mívají vyvedené i signály pro sběrnici SPI. Vy ale klidně použijte I²C.

Zapojení asi nemůže být jednodušší:



A zbytek je už, jak se říká, *obyčejná programátorská nádeničina*. Přečíst data a aktualizovat displej, ke všemu jsou knihovny, no a kdybyste už fakt nevěděli, tak nezapomeňte:

Schéma a zdrojový kód najdete na 🔗 https://eknh.cz/rfdis

34 Klávesnice

34 Klávesnice

Pojďme se podívat zase na jedno praktičtější téma.

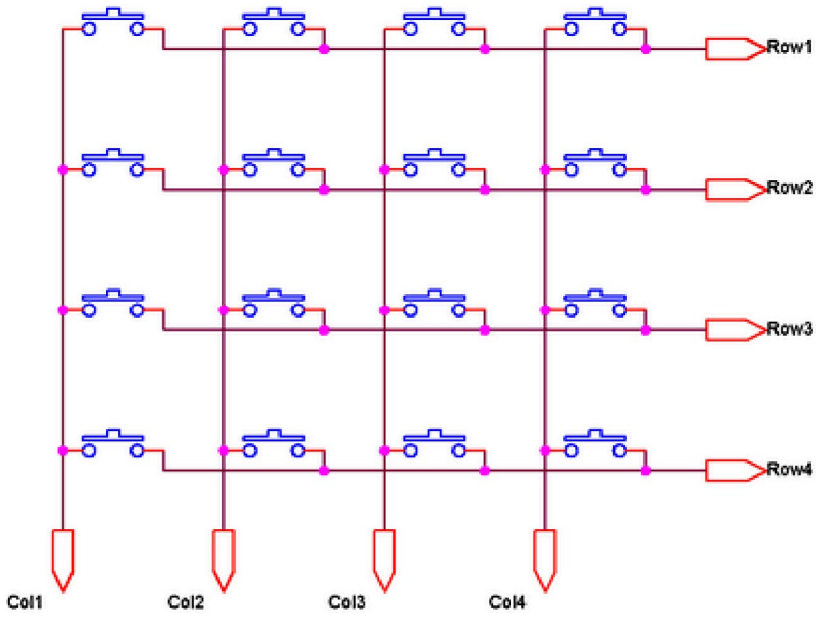
Už víme, jak jednočip naprogramovat, aby něco dělal. Už jsme si i různé informace četli a zobrazovali. Ale co když chceme, aby ho někdo ovládal?

Možností je několik. Pokud vám postačí tři základní pokyny (nahoru, dolů, potvrdit), můžete použít rotační enkodér (už jsme si ho představovali).

Ovšem sázka takříkajíc „na jistotu" je tlačítko. Pokud je potřeba víc pokynů, pak tedy víc tlačítek, respektive rovnou celou klávesnici.

Za chvíli vám ukážu různé způsoby připojení více tlačítek, ale teď probereme ten nejrozšířenější způsob zapojení – „do matice". Používal se u domácích počítačů, používal se u osobních kalkulaček a používá se dodnes i v klávesnicích pro PC.

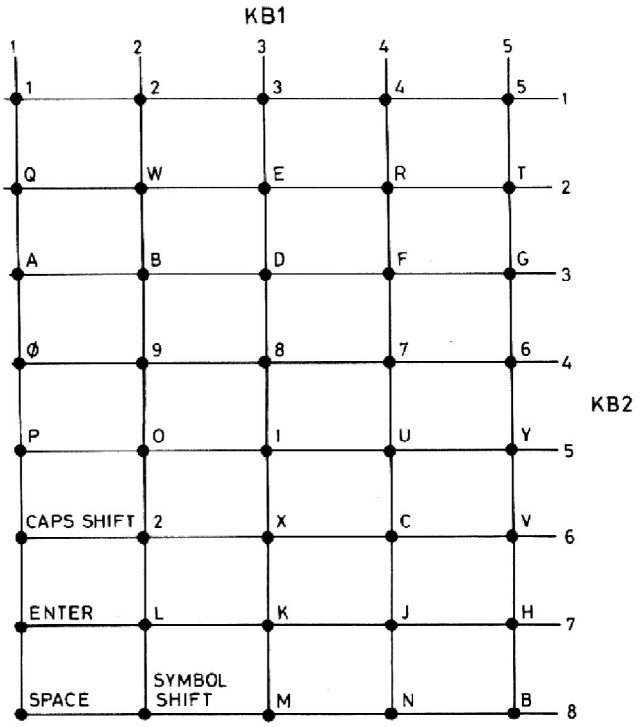
Představme si, že máme N vodičů svisle a M vodičů vodorovně. Třeba v matici 4 × 4 budeme mít čtyři sloupce (Col1-Col4) a čtyři řádky (Row1 – Row4), nějak takto:



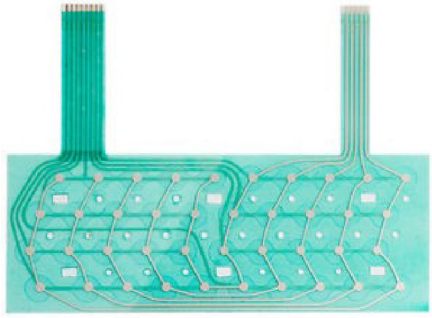
Do míst, kde se sloupce a řádky kříží, připojíme tlačítka. A dál?

Jednotlivé řádky a sloupce připojíme na osm pinů mikrokontroléru. Dejme tomu, že řádky (Row) zapojíme jako vstupy s pull-up rezistory, sloupce (Col) jako výstupy. Klidový stav je takový, že na všech výstupech Col jsou log. 1, tlačítka jsou rozpojená a na vstupech Row je také log. 1 (díky pull-upu). Procesor či jednočip musí takovou klávesnici pravidelně „scannovat" tím, že postupně jeden sloupec po druhém nastavuje do logické 0 a čte stav na vstupech Row. Pokud je to 1111, znamená to, že v daném sloupci žádné tlačítko není stisknuté. Pokud bude některé stisknuté, objeví se na příslušném řádkovém vstupu taky logická 0. Díky tomu program přesně zjistí, která tlačítka jsou stisknutá.

Pro 16 tlačítek potřebujeme 8 vodičů. To možná nevypadá jako velká výhra, ale třeba u počítače ZX Spectrum, které mělo 40 tlačítek, stačilo k jejich obsluze pouhých 13 vodičů (8 řádků × 5 sloupců, ovšem fyzicky uspořádané jako 4 × 10).



Výhoda takového uspořádání je i ta, že nemusíte na každý bod dávat mechanické tlačítko, které je poměrně drahé. Můžete použít dvě membrány, na nichž jsou nakreslené vodivé spoje, křížené v místech tlačítek, a mezi nimi je distanční vložka, která drží v klidu obě vrstvy kousek od sebe. Teprve když v místě křížení zatlačíme, spojí se obě cesty…



Tomuto typu klávesnice se říká membránová. I když třeba v ZX Spectru byl na ní položený gumový hmatník, byla to uvnitř jen a pouze membrána. Dokonce i většina levnějších PC klávesnic má uvnitř membránu. Nevěříte?



Pouze ty dražší mají v sobě opravdová tlačítka.

34.1 Šetříme vývody

Maticové uspořádání je fajn, především díky tomu, že maticově uspořádaná tlačítka můžeme snadno číst přímo jednočipem, bez dalších obvodů.

Pokud ale chceme ušetřit ještě nějaké vývody, nezbude než sáhnout po podpůrných obvodech. Třeba náš oblíbený 74LS138, neboli dekodér 1-z-8. Našich zmiňovaných 16 tlačítek bychom mohli zapojit do matice 2 × 8, řádek by se vybíral binární kombinací na vstupech A0-A2, a četly

by se dva sloupce. Dohromady pět vodičů. V případě klávesnice ZX Spectra (5 × 8) by nám stačilo 8 vodičů celkem (3 pro výběr řádku, 5 pro čtení sloupce).

Můžeme ušetřit ještě víc – existovaly obvody, zvané *prioritní enkodér*, které dokázaly obsluhovat celé pole tlačítek a na výstupu přímo oznamovaly kód, jaké tlačítko bylo stisknuto. Nevýhodou bylo, že tyto obvody byly drahé, málo dostupné a neřešily situaci, kdy uživatel stisknul několik tlačítek naráz (Ctrl Alt Del by asi neprošlo).

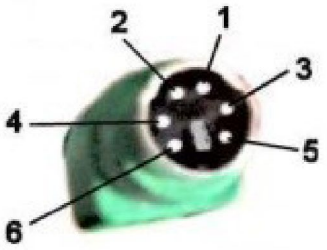
Když se podíváte na klávesnici PC, vidíte přes sto tlačítek, a přitom jen několik vodičů. Je to jednoduché: v klávesnici je speciální obvod, který neustále kontroluje matici tlačítek a po sériové lince posílá informaci o změnách – které tlačítko bylo stisknuto, které puštěno. V prvních PC klávesnicích byl v této roli použit jednočip 8048...

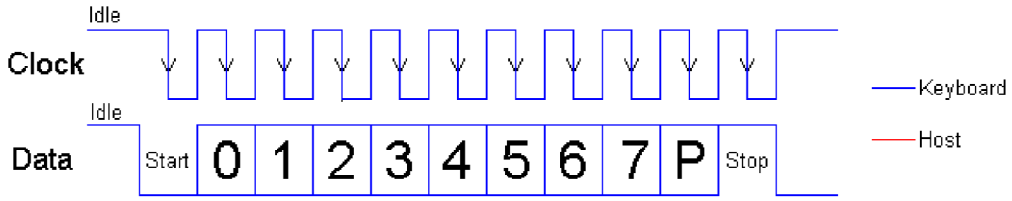
34.2 Připojujeme klávesnici od PC

Dnes už se setkáváme převážně s klávesnicemi, připojenými přes USB, ale dají se koupit i starší typy, které používaly rozhraní PS/2. Tyto klávesnice jsou vhodné i pro amatérské použití s jednočipy, protože mají jednoduchý komunikační protokol: po sériové lince klávesnice posílá kódy stisknutých a puštěných kláves. Jde v podstatě o synchronní sériový přenos s paritním bitem. Používá k tomu dva signály – hodiny a data (oba jsou s otevřeným kolektorem, takže jsou nutné pull-upy). Trochu nepraktické je, že hodinové pulsy generuje sama klávesnice, takže obsluhující procesor nemá pod kontrolou, kdy se přenáší data. Pokud mikrokontrolér neumí pracovat se synchronním sériovým rozhraním s paritou (USART), což pravděpodobně nebude umět, musíte klávesnici zapojit tak, aby její hodinový výstup KBDCLK dokázal vyvolat přerušení, v jeho obsluze přečíst stav na datovém pinu a poskládat si jednotlivé bity zpět do celých bajtů.

PS/2 keyboard connector (MINI-DIN6)

Connector Pin #	Purpose
Pin 1	KBDAT (data)
Pin 2	not used
Pin 3	GND
Pin 4	VCC (+5V)
Pin 5	KBDCLK (clock)
Pin 6	not used





Až budete připojovat PS/2 klávesnici k mikrokontroléru, bude se vám hodit podobný adaptér – jde jen o PS/2 konektor s vyvedenými vývody.

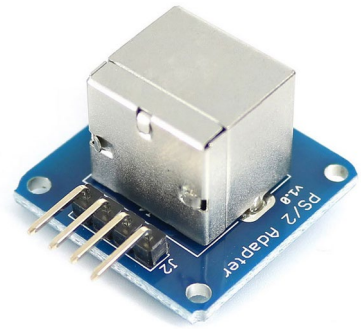


Schéma a zdrojový kód najdete na 🔗 https://eknh.cz/ps2.

34.3 Matice tlačítek

Ovšem ne vždy je potřeba plná klávesnice. Někdy stačí i míň kláves, například číslice a několik funkčních tlačítek. Dají se sehnat například membránové klávesnice 5 × 4 či 5 × 5, lze sehnat i malé klávesničky 4 × 4, 5 × 4, a některé e-shopy nabízejí například i sadu 16 tlačítek a 8 sedmisegmentovek – a navíc s čipem, který se stará o obsluhu klávesnice i zobrazovače, takže k mikrokontroléru připojujete celý blok přes SPI.

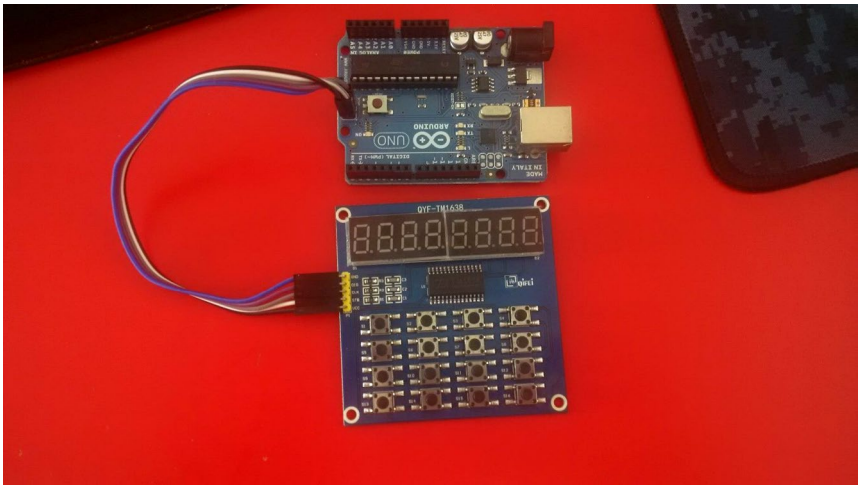
Historické okénko: U počítačů, zvaných „jednodeskové“, se často používaly maticové klávesnice a sedmisegmentové displeje. Některé takové počítače využívaly toho, že princip multiplexovaného displeje, kdy je potřeba vybírat pozice jednu po druhé, se dobře doplňuje s obsluhou klávesnice, kde se taky kontrolují řádky jeden po druhém, a tak často spojovaly výběr pozice na displeji s řádkem klávesnice, a během zobrazování zároveň četly klávesnici.

34.4 Postavte si třeba… kalkulačku?

Našel jsem hezké moduly s šestnácti tlačítky a osmimístným displejem, a když jsem přemýšlel, co z nich postavit, napadla mě, logicky, kalkulačka. Dnes existují dva hlavní způsoby stavby kalkulačky… Ne, pojďme si to říct jinak: Existují dva způsoby, jak přijít ke kalkulačce.

Zaprvé: Buď použijete tu, co máte v počítači, v mobilu, v lednici nebo kdekoli jinde, popřípadě si koupíte některou z 572 modelů, co mají v obchodech, vietnamskými večerkami počínaje. Anebo zadruhé – nějakou si postavíte.

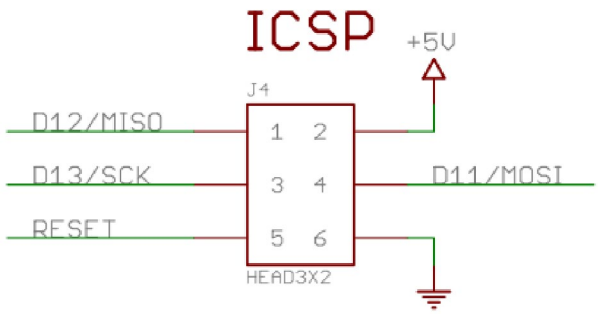
My budeme stavět, protože nás baví elektronika. Dnes existují dva hlavní způsoby stavby kalkulačky. Jeden je opravdu retro, při něm použijete staré kalkulačkové obvody (MHB7001, MC14007), druhý je modernistický, pod heslem *pokud to můžu poskládat z hotových modulů z eBay za dolar až pět včetně poštovného, tak to poskládám z modulů!*



Já si takhle vyhlédl modul na fotografii. Je tam osm osmisegmentovek a šestnáct tlačítek a připojuje se to přes SPI, tedy tři dráty (+ 2 na napájení). *Až ho budete hledat na eBay či AliExpressu, hledejte „keyboard display TM1638“.*

Všimněte si, jak jsem ho k Arduinu připojil.

Na Arduinu najdete speciální konektor s šesti piny (2 × 3), označený ICSP. Je tam + 5 V a zem, jsou tam tři datové vývody (11, 12 a 13) a RESET. Pokud zrovna nehodláte programovat jednočip, tak jsou volné a jsou natvrdo připojené na piny, kde je vyvedené rozhraní SPI.



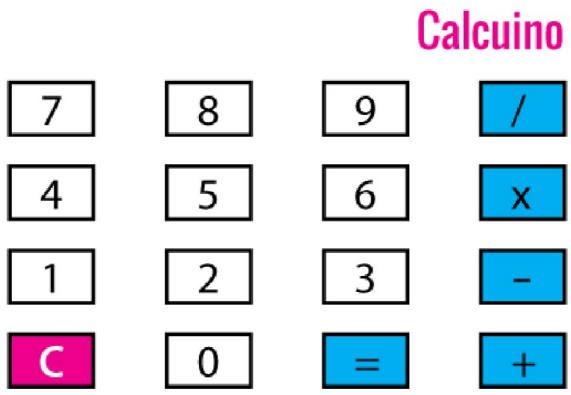
Zmíněný modul používá k řízení tlačítek a displeje čínský obvod TM1638. Existuje datasheet, ale v podobné situaci je nejjednodušší podívat se, jestli už není hotová knihovna pro Arduino. Většinou už ji někdo udělal. Stejně jako v tomto případě.

Všechno propojte, z knihovny otevřete příklad tm1638qyf, změňte nastavení vývodů tak, aby odpovídalo našemu zapojení, a spusťte. Nelekněte se, pokud začne displej zuřivě blikat – to je naprogramované demo!

Práce s klávesnicí je jednoduchá: knihovna vrátí šestnáctibitové číslo, kde co bit, to tlačítko. Nejnižší bit odpovídá tlačítku 1, které je „vlevo nahoře“…

Kalkulačka potřebuje 10 tlačítek pro číslice. Zbývá jich 6. Jedno tlačítko zabere „=“, zbývá jich pět. Takže máme prostor pro čtyři základní matematické operace, a to poslední tlačítko bude „C“ – tedy tlačítko na smazání chyby.

Namapujeme si je takto:



První pokus mi ukázal, že obvod by sice měl odstranit zákmity, ale nějak se mu to nedařilo, takže jsem si je ošetřil sám.

Druhý krok byla jednoduchá funkce: Když zmáčknu tlačítko s číslicí, objeví se na displeji a přidá se k už napsanému číslu. Je to jednoduchá operace: Z = 10 × Z + n. Jestli nevěříte, zkuste si to: Mám na displeji 123, zmáčknu čtyřku, výsledek by měl být 1234, a to je přeci těch původních 123 × 10 + (zmáčknutá) 4! Do téže funkce přidám omezení počtu míst (na sedm, osmou pozici nechám volnou pro symbol „mínus“).

Tlačítko C prostě jen vynuluje pracovní registr (a na displeji se objeví 0).

Až budete implementovat vlastní matematiku, bude se vám hodit pár tipů. Není to úplně triviální, jak by to na první pohled mohlo vypadat. Nezapomeňte, že kalkulačky pracují s infixovou notací, to znamená že 2 + 3 se zadává jako série stisknutí tlačítek 2, +, 3 a =. Algoritmus je tedy takový:

- Pokud uživatel stisknul číslici, přidej ji jako nejnižší řád k tomu, co je na displeji.
- Pokud uživatel stisknul C, nastav displej na 0.
- Pokud uživatel stisknul nějakou operaci, poznamenej si ji. Pak proveď předchozí uloženou operaci, ulož si mezivýsledek do registru X a zobraz ho. Zároveň při dalším stisknutí číslice nejprve vynuluj displej.
- Pokud uživatel stisknul „=“, tak proveď poslední uloženou operaci, výsledek si ulož do registru X a zobraz ho.

K tomu jen poznámka: Kalkulačka nezohledňuje (v téhle verzi) prioritu operátorů, to si můžete vyřešit za domácí úkol.

Kalkulačka potřebuje několik pracovních registrů: jednak registr X na uložení mezivýsledku, pak registr „op“ na uložení zvolené operace, a konečně registr Z, kde je obsah displeje.

Takhle z popisu by to mohlo vypadat, že při stisku tlačítka s nějakou operací se vlastně vykonává ta předchozí, a když se nad tím zamyslíte, tak to tak opravdu je. Ta aktuální se ukládá „na příště“ a provádí se ta předchozí. Tlačítko „=“ je v podstatě stejné jako jakákoli jiná operace, jen s tím rozdílem, že „na příště“ se uloží operace „nedělej nic, jen zkopíruj displej do mezivýsledku!“

Čistě pro formu si zkuste pár cvičení: upravit kalkulačku tak, aby zvládala prioritu operátorů, upravit ji tak, aby umožňovala zadat záporná čísla, upravit ji na RPN, …

RPN (Reversed Polish Notation, obrácená polská notace) je takový zápis matematických operací, kde se nejprve zadávají čísla, a teprve poté operátory. Například 2 + 3 × 4 se s RPN zadá jako 2 3 4 × +. Čísla se průběžně ukládají na zásobník, operátory si ze zásobníku berou argumenty a vrací tam výsledky.

Schéma a zdrojový kód najdete na ⌀ https://eknh.cz/calcuino

35 Osm tlačítek na třech vodičích

35 Osm tlačítek na třech vodičích

A nejen to. Proč? No, máme v ruce Arduino, to má 14 datových vývodů, 6 analogových. To není moc. Zapojíte k tomu dvě čidla, SD kartu, displej – a kam připojíte tlačítka?

Maticové uspořádání může hodně ušetřit – pro 16 tlačítek potřebujete 8 vodičů, pro 40 tlačítek 13 vodičů, ale to může být stále hodně. Naštěstí je pár triků…

35.1 Multiplexior / Demultiplexor

Pamatujete na kapitolu o číslicové technice? Multiplexor vybírá jeden z mnoha vstupů, a ten posílá na výstup, demultiplexor (dekodér) zase aktivuje jeden z několika výstupů.

Představme si tedy matici 8 × 8 tlačítek. Buzení řádků zařídí náš oblíbený 74LS138 – stručně zopakuju: obvod má tříbitový vstup a osm výstupů, které jsou v log. 1 kromě toho, jehož číslo odpovídá tříbitovému číslu na vstupech.

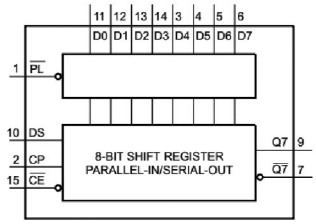
Osm sloupců bude zase připojeno na multiplexor, třeba 74HCT151. Tento obvod má pro změnu osm datových vstupů, tři vstupy adresy a dva výstupy, Y a /Y. Na výstup je připojen ten vstup, jehož číslo… a tak dál…

Takže potřebujete tři vývody na výběr řádku, tři vývody na výběr sloupce, a jeden vstup, který řekne, jestli dané tlačítko je nebo není stisknuté. 64 tlačítek tak obsloužíte pomocí 3 + 3 + 1 = 7 signálů. Na poloviční počet tlačítek stačí 6 vodičů…

A co na to jít ještě jinak?

35.2 PISO a SPI

V kapitole o posuvných registrech jsme si představili obvody PISO – jde o posuvné registry s paralelním vstupem a sériovým výstupem. Jako příklad si vezměme obvod 74HCT165. Jde o osmibitový posuvný registr PISO.

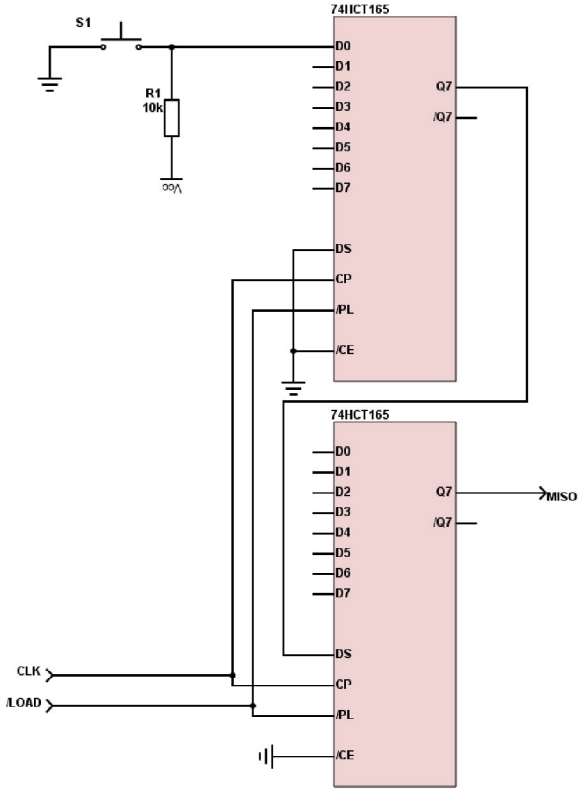


Datové vstupy jsou označené D0 až D7. Vstup /PL nahraje do registru aktuální stav na těchto datových vstupech. Výstup Q7 odpovídá nejvyššímu bitu posuvného registru (tedy po nahrání D7), /Q7 je, nepřekvapivě, jeho negace. Náběžná hrana na vstupu CP (Clock Pulse) posune obsah registrů ve směru 0->1->2->3->4->5->6->7, nejvyšší bit je tedy zahozen, na jeho místo přichází nižší… (Programátorsky řečeno: jde o posun doleva.)

Vstup /CE povoluje hodinový vstup (Clock Enable). Pokud je aktivní, tedy log. 0, tak vše funguje tak, jak jsem popsal, pokud je neaktivní (log. 1), tak jsou hodinové pulsy CP ignorované.

Konečně poslední vstup, DS (Data Serial input) slouží k sériovému zápisu do registru. Díky tomu můžete tyto obvody snadno řetězit – výstup Q7 jednoho obvodu připojíte na vstup DS druhého…

Když potřebujete obsloužit například 16 tlačítek, zřetězíte dva obvody 74HCT165 a na datové vstupy připojíte tlačítka s pull-upy, třeba takto (nakreslil jsem jen jedno tlačítko…)



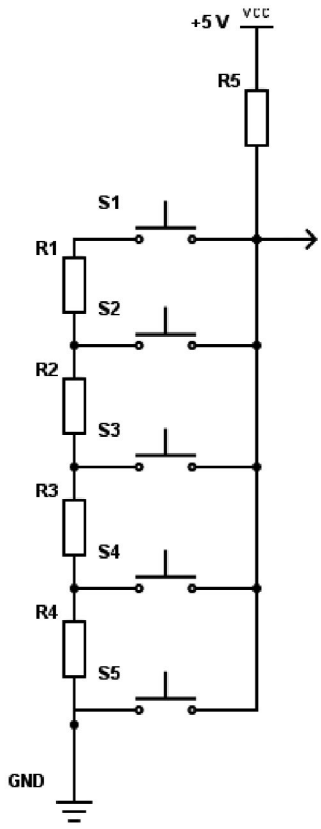
Stačí pouhé tři vodiče. Signálem /LOAD načtete stav tlačítek do registrů. Na výstupu MISO je k dispozici stav tlačítka 7 u spodního obvodu, postupnými pulsy na vstupu CLK se na výstup MISO dostávají další tlačítka…

Nevýhodou je, že každé tlačítko potřebuje vlastní pull-up rezistor a že načítání nějakou dobu trvá.

Co myslíte – jde to dotáhnout ještě do většího extrému?

35.3 Analogová cesta

Jde! Co by nešlo. Představte si takový rezistorový dělič, třeba ze čtyř rezistorů se stejnou hodnotou, k němuž jsou připojená tlačítka do společného bodu, který je přes pátý rezistor stejné hodnoty připojený k napájecímu napětí. Nějak takto:



Za normálního stavu, pokud není žádné tlačítko stisknuté, je na výstupu napájecí napětí 5 V přes rezistor R5.

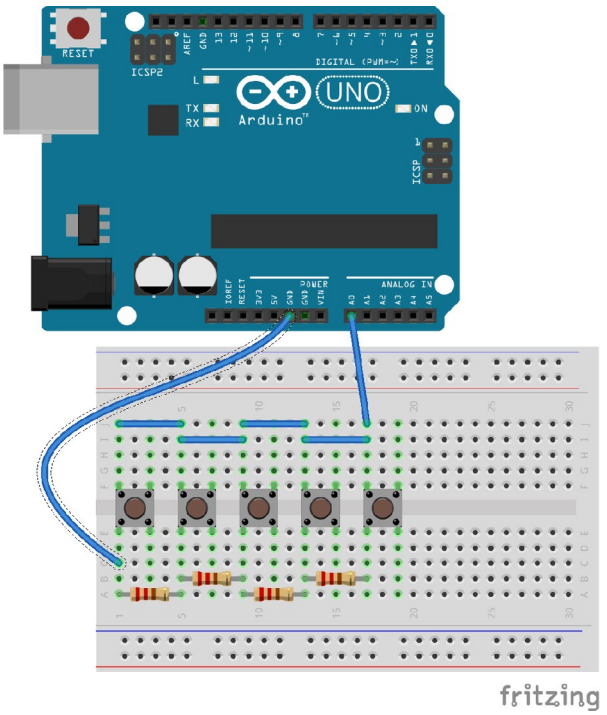
Když stisknete tlačítko S5, bude výstup připojen přímo na zem, a bude tam tedy rovných 0 voltů.

Když stisknete tlačítko S4, půjde proud přes rezistor R5, tlačítko S4 a rezistor R4 k zemi. Na výstupu bude 2,5 voltů, protože stiskem S4 vznikne dělič R : R, a jelikož jsou oba odpory stejně velké, bude výsledné napětí polovina napájecího.

Analogicky při stisku S3 bude dělič v poměru R:2R (R5 proti R3 + R4), tedy na výstupu bude cca 3,33 V.

Stisk S2 znamená dělení v poměru R:3R, tedy cca 3,75 V. Tlačítko S1 pak nastaví dělič na poměr R:4R, a výsledkem by mělo být napětí 4 volty.

No, a když výstup připojíte k analogovému vstupu Arduina, tak pomocí analogRead můžete číst přímo hodnoty stisknutých tlačítek. Dokonce můžete vynechat i R5 a použít interní pull-up – pinMode(A0, INPUT_PULLUP).

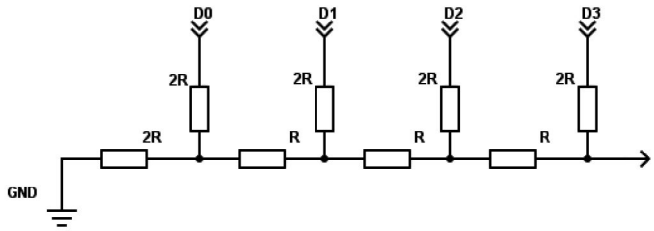


Takže nakonec obsluhujete pět tlačítek pouhým jedním vstupem! Samozřejmě, takovéhle řešení má svoje limity: analogový převodník není nekonečně přesný, rezistory také nebývají přesné, navíc se „rozlaďují“ teplem, takže pět tlačítek je takové rozumné maximum.

Ale když už jsme na to narazili – existuje takové hezké zapojení, kdy pomocí rezistorů převádíte digitální signál na analogový. Samozřejmě existují velmi přesné součástky, digitálně-analogové převodníky (DAC), které tuhle práci dělají za vás. Dokonce jsou i v podobě malých obvodů, řízených třeba přes I²C nebo SPI. Ale když není zbytí, lze použít i jednoduché zapojení, které se nazývá R-2R.

35.4 R-2R

Označení vyjadřuje, že se zapojení skládá z rezistorů dvou typů: s nějakým odporem R, a s dvojnásobným odporem 2R. V praxi se volí třeba 10k a 20k (ten není v základní řadě E6 ani E12, ale v řadě E24 už ano). Popřípadě se zapojují dva rezistory s hodnotou R do série.

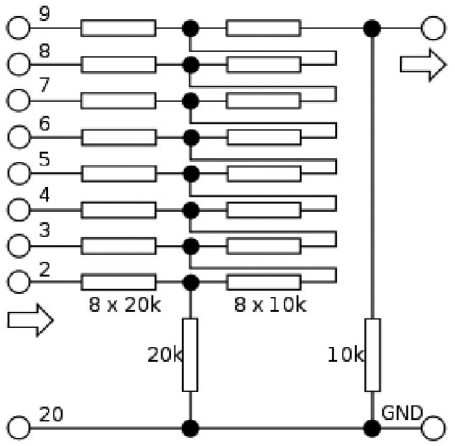


Vidíte, že jsou zde čtyři digitální vstupy D0-D3, připojené přes rezistory 2R k rezistorovému děliči z rezistorů R, který je připojen k zemi u nejnižšího bitu rezistorem 2R. Není ale problém tento „žebřík z rezistorů“ (tak se tomu zapojení také říká: R-2R ladder) nastavit o další bity…

Dá se spočítat, jak takové zapojení funguje, ale já se omezím na tvrzení, že každý digitální vstup, který je v logické 1, přispěje určitou částí k výslednému napětí. D3 přispěje napětím U/2, D2 napětím U/4, D1 napětím U/8 a D0 napětím U/16. Což je výhodné, protože výsledné napětí se pohybuje od 0 do 15/16 napájecího napětí a přesně odpovídá binárnímu vyjádření na digitálních vstupech.

Pokud naskládáte další R-2R členy, zvětšíte tím počet bitů, a tím i přesnost převodu.

Kdysi, v dřevních dobách PC, se tímto způsobem vytvářely zvukové výstupy: osmibitový R-2R převodník, připojený na paralelní port, stačil na přehrávání monofonní hudby. Tyto převodníky byly známé pod názvem Covox, a jeho schéma bylo velmi jednoduché:



Zpátky k našim tlačítkům: pokud bychom je připojili k R-2R „žebříku“, mohli bychom teoreticky zvýšit přesnost převodu a z ADC v Arduinu číst přímo hodnoty tlačítek jako bity převedeného čísla. V praxi bych do takového řešení ale nešel – u osmi tlačítek by stisku každého odpovídaly čtyři hodnoty, a to bych se bál, že rušení a nepřesné hodnoty součástek způsobí nespolehlivé fungování.

Každopádně vidíte, že fantazii se meze nekladou, a kdyby bylo nejhůř, tak i těch několik tlačítek můžete připojit na jeden vodič.

36 Joystick

36 Joystick

V dobách osmibitových počítačů byl joystick velmi důležitou periferií, bez níž se neobešel žádný hráč. Technicky jde o rukojeť, k níž jsou připojena čtyři tlačítka do kříže, a jedno, dvě či tři „akční“ tlačítka.



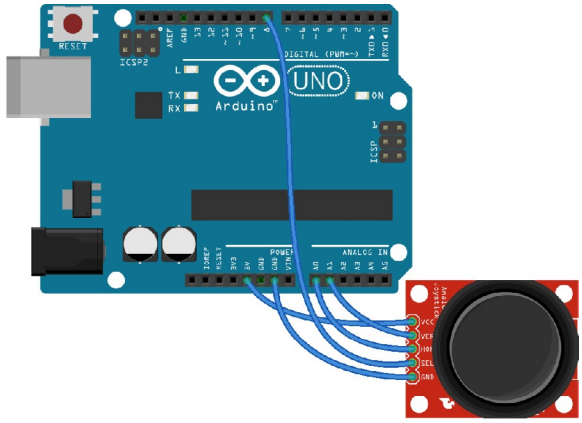
CC-BY-SA, autor Kshade

Při pohybu rukojetí se spínaly jednotlivé spínače pro směry nahoru, dolů, doleva a doprava. Při šikmých pohybech se sepnuly dva najednou. Při puštění vrátil systém pružinek rukojeť do středové polohy. Práce s takovým joystickem byla extrémně jednoduchá – stačilo jen číst stav pěti, šesti či sedmi spínačů, a podle toho programem reagovat.

Modernější joysticky neobsahují spínače, ale dvojici potenciometrů. Jeden snímá naklonění páky v ose X, druhý v ose Y. Výstup tedy není digitální (nahoru / dolů), ale analogový – lze změřit velikost odchylky od středové pozice.

Tyto joysticky znáte například z ovladačů pro herní konzole. Dají se ale sehnat i v podobě malých modulů – jen destička a joystick. Takové moduly pak mívají pět vývodů: vstupy GND a Vcc pro připojení země a napájecího napětí, analogové výstupy X a Y, které jsou připojené k jednotlivým potenciometrům, a digitální výstup SW, který je připojen k tlačítku uvnitř joysticku (klobouček lze totiž zmáčknout, a tím vyvolat nějakou funkci).

Připojení k Arduinu je opět prosté – analogové výstupy připojíme k analogovým vstupům, tlačítko k některému digitálnímu vstupu… Před použitím je dobré změřit, jaké hodnoty Arduino přečte na pinech A0 a A1 při maximální výchylce všemi směry, a jaká hodnota je pro joystick v klidu, a podle toho přepočítat naměřená data na použitelné hodnoty (říká se tomu *kalibrace joysticku*).



Zkuste si spojit toto zapojení třeba s už dříve uvedeným grafickým displejem. Pomocí pohybu joysticku můžete kreslit po displeji, případně ovládat kurzor. Ovšem naprogramování už nechám na vás…

Jen pro jistotu připomínám, že tlačítko je jen obyčejný spínač, který spíná k zemi, a že je nezbytně nutné mít tedy co? Ano, je nezbytně nutné mít pull-up rezistor! Buď externí, přímo připojený, nebo povolit interní pull-up rezistory v Arduinu.

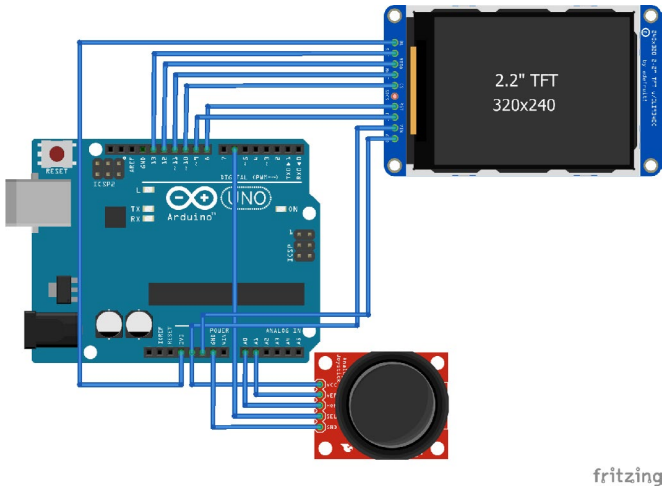


Schéma a zdrojový kód najdete na ⌀ https://eknh.cz/joydis

37 ESP8266 WiFi

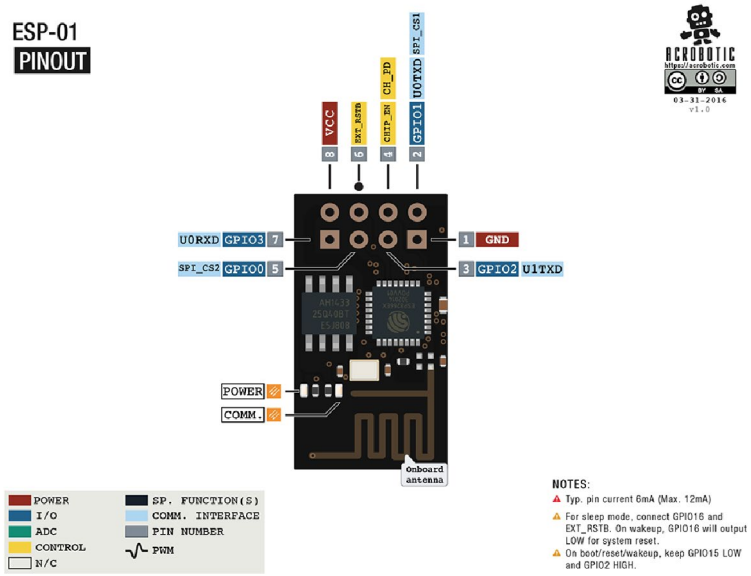
37 ESP8266 WiFi

V původní osnově této knihy, kterou jsem napsal před třemi lety, byla v tomto místě kapitola o připojování Arduina k internetu přes kabel (Ethernet). Ne že by to nešlo – jde to samozřejmě stále, ale mezitím přišly nové součástky, které umožnily za nižší cenu udělat připojení bezdrátové, pomocí WiFi.

Průkopníkem těchto součástek byl obvod ESP8266 – asi tím nejdůležitějším faktorem, který napomohl jejich rozšíření, bylo, že moduly stojí okolo dvou, tří dolarů, což je neuvěřitelně nízká cena. Poměrně brzy se objevily i překladače pro tento čip, další programovací jazyky (Lua, JavaScript, Python, BASIC, …) a další moduly

37.1 Moduly ESP8266

Moduly s tímto obvodem se brzy objevily v několika variantách. První byl modul, dnes označovaný jako ESP-01, připojovaný přes osm pinů:



ESP-01 byl nejjednodušší, nejlevnější a používal se především jako periferie – připojuje se přes sériové rozhraní (většinou rychlostí 115200 Bd) a veškerá komunikace s internetem se odehrává pomocí speciálních příkazů („AT příkazy“).

Například když pošlete do ESP8266 sekvenci znaků „AT+CWLAP\n“ (bez uvozovek, velká písmena), oscannuje si dostupné WiFi sítě a vrátí seznam jejich identifikátorů.

K WiFi síti se připojíte pomocí AT+CWJAP = „moje-sit“,„moje-heslo“

Po úspěšném připojení k WiFi AP můžete zkusit navázat spojení se serverem: AT+CIPSTART = „TCP“,„192.168.0.1“,„80“

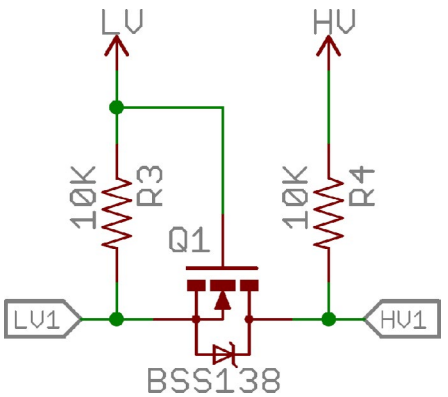
Pak můžete přenášet data: AT+CIPSEND = délka dat, a samotná data…

Naštěstí není potřeba používat tento způsob, existují knihovny, které AT příkazy zapouzdřují a navenek dovolují pohodlnou práci pomocí funkcí vyšší úrovně.

Připojení tohoto modulu přímo k Arduinu nedoporučuju – modul je určen pro napětí 3,3 voltů, a snadno jej spálíte. Použijte buď převodník úrovní, nebo specializovaný shield s tímto modulem, kde je převodník už zabudovaný.

37.2 Převodník napěťových úrovní

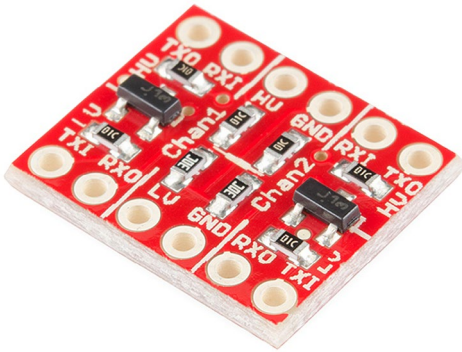
Velmi užitečná věc ve chvíli, kdy potřebujete propojit obvody s různou logikou – nejčastěji třeba 5 V Arduino a 3,3 V senzory. Existuje jednoduché zapojení, které dokáže konvertovat logické signály z jedné úrovně do druhé, a potřebuje k tomu pouze jeden tranzistor MOSFET:

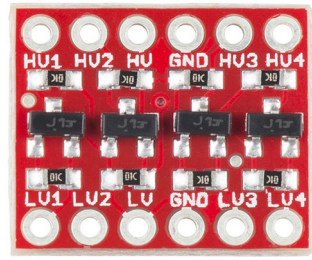


LV a HV je nízké (low) a vysoké (high) napájecí napětí, LV1, respektive HV1 jsou vstupy / výstupy v patřičných úrovních. Převodník funguje obousměrně.

⌀ https://eknh.cz/lcnv

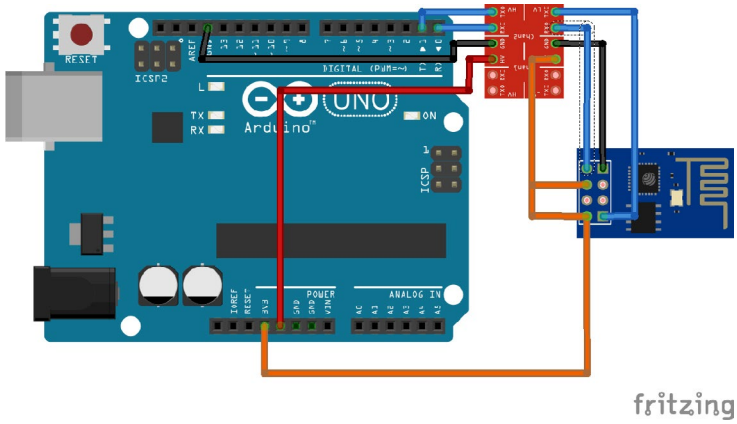
Naštěstí nemusíte sestavovat a pájet převodníky, existují jako hotové moduly, nejčastěji pro čtyři signály, nebo pro osm signálů:





Na jedné straně, tam, kde je LV, se připojují signály obvodu 3,3 V, na straně HV se připojují 5 V. LV a GND (resp. HV a GND) slouží k připojení napájecího napětí.

Připojení ESP8266 přes takový převodník je opravdu hračka:



Všimněte si, že modul je zapojený tak, jak byste čekali, tedy RxD na TxD a vice versa, jen je mezi Arduino a ESP vložen převodník úrovní.

Upozornění: Když budete takto zapojené Arduino programovat, pravděpodobně nastane chyba. Důvodem je to, že programování Arduina probíhá přes sériový port, a pokud je na tomtéž sériovém portu připojen jiný obvod, který má tendenci komunikovat (a to ESP8266 má!), tak ruší přenos dat, a programování skončí s chybou. Řešení je v takovémto případě před programováním modul ESP8266 odpojit!

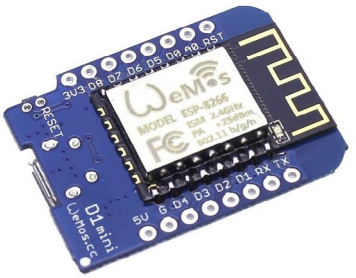
Pro ovládání ESP8266 opět existuje celá řada knihoven, které zapouzdřují příkazy tak, aby práce byla co nejjednodušší.

37.3 WeMos D1 Mini, NodeMCU

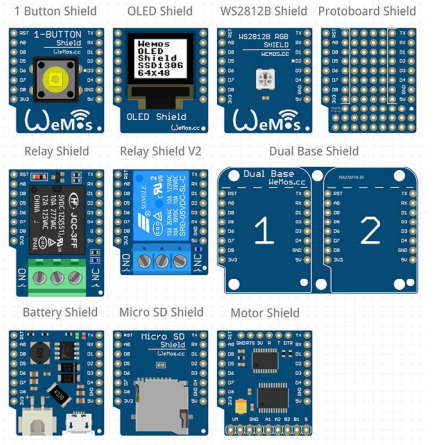
Samotný procesor ESP8266 je poměrně výkonný a nabízí i několik datových pinů. Logicky se tedy nabízí možnost použít přímo tento modul místo Arduina. Jde to, s drobnými omezeními (např. ESP8266 má pouze jeden A/D převodník).

Asi nejlepší pro takovou práci jsou moduly NodeMCU a WeMos D1 Mini.





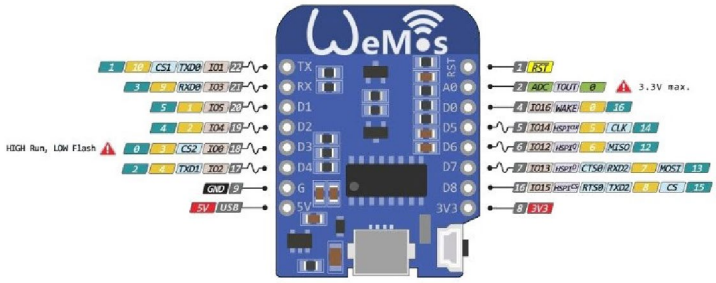
Oba tyto moduly obsahují základní modul ESP8266 a několik součástek, které slouží pro připojení k počítači přes USB. K těmto modulům existuje široká škála nástrojů a doplňků – například pro modul WeMos D1 Mini existují shieldy s displejem OLED, s barevlou LED, s teploměrem, pro SD kartu, shield s relé, s obvody pro napájení z baterií, pro ovládání motorů a další…

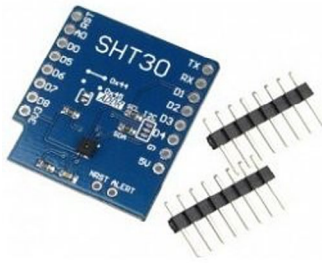


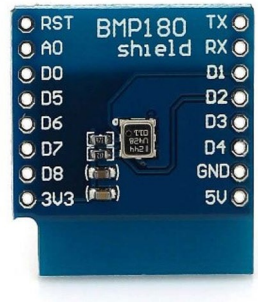
Výhoda je, že nejste odkázáni jen na Arduino – na webu naleznete návody, jak do těchto modulů nahrát například firmware pro Python nebo jazyk Lua. Vytvoření nějakého toho „internetu věcí“ je pak otázka doslova pár minut.

37.4 Bezdrátový teploměr s WiFi

Tentokrát opustíme bezpečný svět Arduina UNO a použijeme výše zmíněný modul WeMos D1 Mini spolu s moduly BMP180 (ten už známe) a SHT30 – tento modul měří teplotu a relativní vlhkost a připojuje se přes sběrnici I²C. Modul SHT30 je o něco přesnější než DHT11/DHT22, ale především používá standardní sběrnici I²C.







Poskládání takového teploměru je otázka několika minut, napsání softwaru také. Naštěstí existuje způsob, jak takovouto sestavu přímo programovat přes Arduino. Dovolte odbočku:

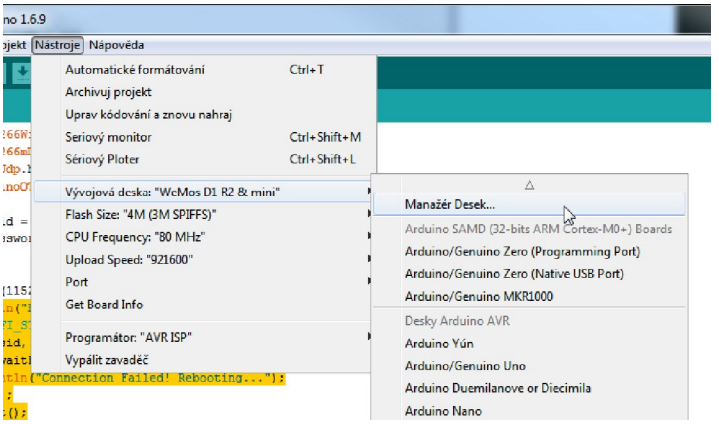
37.5 Instalace podpory ESP8266 do Arduino IDE

1. Nainstalujte ovladače pro USBtoUART převodník, který je v kitech WeMos a jejich klonech – bývá to CH340.

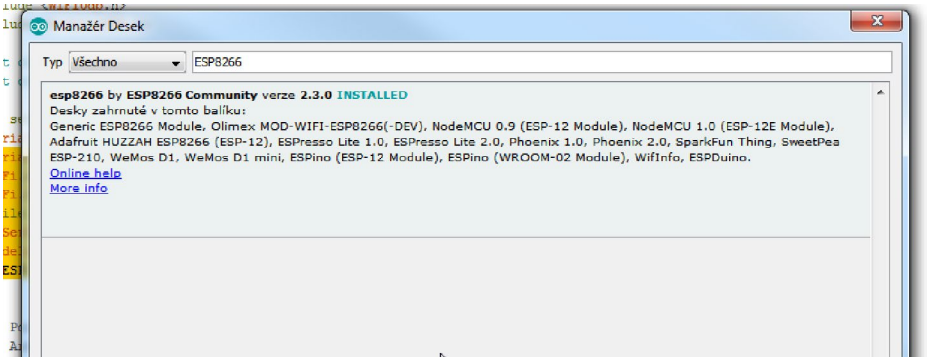
2. V Arduino IDE vyberte z menu Soubor – Vlastnosti, a do pole „Správce dalších desek URL“ přidejte adresu **http://arduino.esp8266.com/stable/package_esp8266com_index.json** (Pokud tam už nějaké záznamy máte, přidejte tento a oddělte jej čárkou)



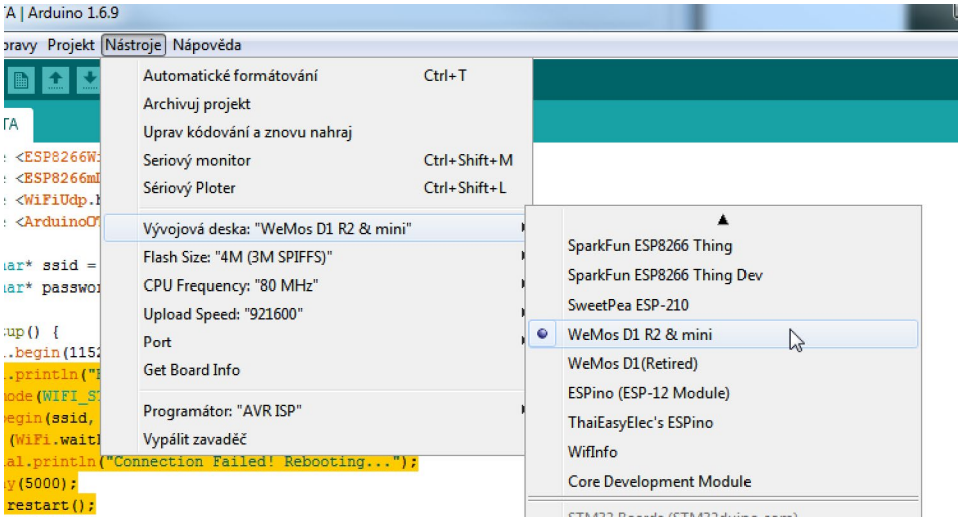
3. V menu Nástroje – Vývojová deska vyberte Manažér desek



4. Vyberte desku ESP8266 (můžete použít filtrovací pole nahoře)



5. Po nainstalování vyberte desku WeMos D1 Mini



Příklady jsou v sekci Příklady z knihoven, klíčové slovo ESP8266. Dokumentaci naleznete na http://esp8266.github.io/Arduino/versions/2.3.0/

37.6 WiFi Manager

Podpora pro WiFi je u desek D1 Mini v prostředí Arduino naprosto excelentní. Stačí zadat jméno WiFi sítě, heslo, připojit se… Jenže co když nechcete nechávat heslo ke své wifi síti v kódu, nebo budete instalovat zařízení někde, kde je síť, ke které zatím nemáte údaje? Snadná pomoc, použijte skvělou knihovnu WiFi Manager (https://github.com/tzapu/WiFiManager)

Tato knihovna nabízí, mimo jiné, funkci autoConnect(ssid); Při prvním zapnutí se ESP8266 zapne jako access point, a vy se k němu můžete připojit jako k jakémukoli jinému access pointu – třeba přes smartfon. Pak stačí otevřít prohlížeč a zadat adresu 192.168.4.1 a otevře se stránka s nastavením wifi manageru, kde nastavíte jméno sítě a heslo. Po úspěšném nastavení si WiFiManager tyto údaje uloží k dalšímu použití, a celé zařízení restartuje.

Pokud knihovna po startu objeví uložené údaje, zkusí se k takové síti připojit, už jako klient. Uspěje-li, spustí se hlavní program. Neuspěje-li, opět se přepne do AP módu a funguje jako server s nastavovací stránkou.

37.7 Klient / server?

Zásadní otázka u našeho teploměru je: Klient, nebo server? Odpověď záleží na tom, jestli:

- chcete data nějak zpracovávat, třeba na serveru
- máte dostatek volných WiFi kanálů, tedy bydlíte někde na samotě
- se chcete jen lokálně podívat, třeba z mobilu, kolik je stupňů.

Pokud pro vás platí poslední případ, a zároveň nejste někde v hustě osídlené oblasti se spoustou aktivních WiFi, zvolte způsob „server". Naprogramujte ESP8266 jako server, napište jednoduchou webovou stránku a v ní aktualizujte hodnoty teploty, vlhkosti a tlaku. Pak se k takovému teploměru připojíte jako k libovolnému access pointu a pomocí prohlížeče sledujete data.

V opačných případech zvolte postup „client". Ovšem znamená to, že někde (u vás doma, v routeru, v domácím serveru, nebo v internetu kdesi) musí běžet server, k němuž se teploměr bude pravidelně připojovat a hlásit naměřené údaje. Do podrobností se tu nebudu pouštět, to záleží asi na vkusu každého z vás, jaké řešení zvolí.

Existují služby (asi nejznámější je Thingspeak), které po registraci nabízejí jednoduché serverové API pro zasílání hodnot. Na webu můžete data prohlížet, zobrazovat v grafech, nebo exportovat pro další zpracování.

Další možností je vlastní server, buď s HTTP REST API, nebo (lépe) s MQTT brokerem. MQTT je de facto standard pro posílání zpráv ve světě Internetu věcí (IoT). Jde o jednoduchý a datově nenáročný protokol pro posílání zpráv a jejich přijímání, založený na architektuře pub-sub (Publisher – Subscriber). Podrobnější popis je mimo záběr této knihy, zájemce odkážu například na svůj web Iotta.cz, kde se MQTT věnuji.

Schéma a zdrojový kód najdete na ⌀ https://eknh.cz/esp.

38 Low Power

38 Low Power

S rozvojem technologií, vznešeně nazývaných „internet věcí“ (IoT, Internet of Things) se do popředí zájmu dostala spousta aspektů. Například bezpečnost, ergonomie, nové technologie pro komunikaci, ale především dvě slůvka: Low Power. Česky hovoříme o nízkoenergetických zařízeních, o zařízeních s malou spotřebou atd. Ale internetem věcí to nezačalo, kdepak. Na počátku těchto snah byly různá přenosná zařízení, od kalkulaček přes rádia až k notebookům, tabletům a smartfonům.

U stolního počítače se s tím lidé nějak smířili. 300 nebo 400 wattů, to už nějak překousnou. Mimochodem, víte, co se s většinou té energie stane? Ano, promění se v teplo. Pokud jste někdy byli v serverovně, tak to znáte. Ostatně i jeden stolní počítač dokáže v malé místnosti slušně zatopit.

Jenže co projde stolnímu počítači, to neprojde notebooku nebo tabletu. Proto přišly na řadu jednak technologie pro zvýšení kapacity baterií, ale také postupy na snížení odběru.

Takových postupů je v zásadě několik. Zaprvé: používat pokud možno CMOS technologii, která má v klidu (to podtrhuju) velmi nízký odběr. Zadruhé: snížit napájecí napětí. 3,3 voltů už je standard, ale lze jít ještě níž. Zatřetí: zpomalit práci.

Jistě si pamatujete, že nejvíc proudu teče obvodem tehdy, když mění svůj stav, když se tranzistory překlápějí. Vyšší hodinový kmitočet znamená tedy i vyšší spotřebu. Moderní procesory i mikrořadiče proto umožňují přepnout hodinovou frekvenci na naprosté minimum, třeba v řádech kilohertzů, nebo se zastavit úplně – takzvané *deep sleep* módy. Z takového stavu se procesor probudí buď po uplynutí nějaké doby, nebo po příchodu vnějšího impulsu (dotyk displeje, stisknutí tlačítka, příchod dat, …) Poté se procesor přepne na pracovní frekvenci, událost rychle obslouží a zase se uspí.

Technikou uspávání lze odběr snížit i o několik řádů. Pokud třeba měříte venkovní teplotu, asi není potřeba skutečný „real time“ přístup, stačí změřit teplotu jednou za několik minut a údaj někam poslat. Takto může zařízení běžet na jednu baterii i několik let, pokud zvolíte vhodný energeticky nenáročný způsob přenosu dat. *Ne, WiFi není energeticky nenáročný způsob!*

Samozřejmě, uspávání nestačí, když vám teče proud jinudy. Nejčastější příčinou rychlého vybíjení jsou zapomenuté LED – sice mají nízký odběr ve srovnání s jinými zdroji světla, ale i tak je to stále hodně. U low power zařízení je potřeba se s odběrem dostat někam k mikroampérům, a 10 mA, tekoucích LEDkou, je vlastně „hrozně moc“ proudu.

Takové Arduino Micro je docela slušné zařízení, které by se dalo použít pro low power zapojení, jenže obsahuje „power LED“, tedy LEDku, která indikuje, že je zařízení napájené. Pokud tuto LED odstraníte, ušetříte většinu spotřeby.

Dalším místem, kudy protékají cenné miliampéry proudu úplně zbůhdarma, jsou stabilizátory napětí. Pokud nejsou LVDO (Low Voltage Drop Out), popřípadě Ultra LVDO, tak pro svou vlastní činnost spotřebují velké množství energie i v případě, že se žádná energie neodebírá (takzvaný klidový proud).

Klasický stabilizátor 7805, který se používá už desítky let pro stabilizaci napájecího napětí na 5 V, potřebuje na vstupu alespoň 7 voltů a sebere pro svou vlastní potřebu 5 až 8 miliampérů (hledejte v datasheetu parametr, nazvaný Quiescent Current, I_Q). Stabilizátor MCP1703 se spokojí s napěťovým rozdílem do 1 V a pro svou potřebu si vezme dva mikroampéry. Stabilizátory z řady TPS783xx mají úbytek napětí okolo 150 mV a vlastní spotřebu 500 nanoampérů.

Samotná spotřeba není jediným faktorem, co ovlivňuje výdrž zařízení na baterie. Důležitým faktorem je i maximální odebíraný proud. Některé bezdrátové technologie a náročnější senzory se sice umí uspat k téměř nulovému odběru, ale po probuzení potřebují velmi velký proud. GSM modul, který se ve vašem mobilním telefonu připojuje k mobilní síti, si v klidu vystačí s mikroampéry. Při komunikaci potřebuje řádově desítky až stovky miliampérů, ale při úvodním připojování, vyhledávání sítě a přihlašování se do ní si vezme až 2 A. Sice jen na krátký okamžik, ale baterie musí být schopna tento prudce zvýšený odběr pokrýt.

Když je baterie starší, tak už nebývá schopna takhle velké proudy dodat. Proto se občas stane, že při navazování spojení nebo při přehlašování k jiné základně vzroste odebíraný proud tolik, že vezme vše, co baterie dává, tím pádem poklesne napětí pod mez, která je pro procesor bezpečná, a zařízení se restartuje. Bohužel se po restartování opět telefon chce přihlásit k síti, opět vzroste spotřeba, a telefon se dostane do série neustálých resetů.

To mimochodem vysvětluje i příčinu toho, proč se úplně vybitý telefon často musí nejprve nějakou dobu nechat nabíjet, než ho lze zapnout: protože nabíječky, zejména ty levnější, nejsou schopné dodat tak velké proudy.

Takové situace se řeší buď používáním úspornějších telekomunikačních metod (bezdrátové spojení v bezlicenčním pásmu 433 / 868 MHz, Sigfox), nebo pomocí kondenzátorů s velkou kapacitou, které jsou schopné krátkodobě dodat velmi veký proud. Ovšem kondenzátory se samovolně vybíjejí, a tak mohou představovat další místo, kudy vám poteče proud a bude zvyšovat odběr.

Zajímavé technologie, které mohou pomoci při napájení nízkopříkonových zařízení, se skrývají pod označením *energy harvesting*. Slouží k získávání energie ze zdrojů, jako je okolní teplota, proudění kapalin, rádiový signál, světlo, otřesy a podobně. Některá zařízení, určená pro voperování do lidského těla, dokáží získávat energii i z krevního cukru. Speciální obvody dokáží i velmi malé množství takto získané energie efektivně zpracovat a uložit do baterií či kondenzátorů.

38.1 Solární články

Boom solárních elektráren přinesl kromě negativních dopadů i některé pozitivní, a jedním z nich je výrazné zlevnění solárních – přesněji fotovoltaických – článků. Se zařízeními „na sluneční baterie" se můžete setkat v nejrůznějších situacích, od smysluplných (napájení inteligentních autobusových zastávek nebo autonomních komunikačních zařízeních v místech, kam nelze moc dobře dovést elektřinu) až po samoúčelné, a někdy i vysloveně podivné. Díky rapidnímu poklesu cen si můžete i poměrně výkonné a velké fotovoltaické panely pořídit za ceny v řádech desítek korun.

Fotovoltaické články jsou v podstatě velkoplošné polovodičové diody, kde vlivem energie zvenčí vzniká elektrické napětí. Vzpomeňte si na jeden z prvních experimentů v této knize, kdy jste svítili na LED jasným světlem a měřili vznikající napětí. Tak fotovoltaický článek je založený na stejném principu, jen na větší ploše.

Nevýhodou FV článků je poměrně malá účinnost – u těch, které za levno koupíte, to bude v jednotkách procent, průmyslové mívají okolo 15 %, experimentální laboratorní články dosahují účinnosti 30-40 procent. Ale i tak lze malé solární panely použít k napájení low power zařízení.

Fotovoltaický článek funguje vlastně jako baterie – má kladný a záporný pól, a když na něj dopadá světlo, tvoří se mezi nimi napětí. Problém je, že účinnost záleží i na dalších faktorech, jako je úhel dopadajícího světla nebo teplota článku. Pokud vás třeba napadlo zvýšit účinnost tím, že před článek předřadíte lupu, která koncentruje větší množství dopadajícího světla, je to nápad v zásadě dobrý, ale myslete na to, že spolu se slunečním svitem koncentruje čočka i teplo, a článek se snadno přehřeje, popřípadě spálí.

Napájení z fotovoltaických článků vyžaduje trochu víc přemýšlení, než jen: semhle plus, semhle mínus… Intenzita světla, a tedy i množství dodané energie, kolísá v průběhu dne – vlivem oblačnosti, nebo i vlivem toho, jak Slunce putuje po obloze a světlo tak dopadá z různých úhlů (samozřejmě je možnost vybavit solární panel zařízením, které jej bude natáčet podle směru svitu, ale to spotřebuje rovněž nějakou energii…) Navíc v noci nebude produkovat fotovoltaický článek téměř žádnou energii, protože v noci bývá většinou v našich zeměpisných šířkách tma.

Proto je dobré za FV článek zapojit nějaké zařízení, které dokáže jímat vyrobenou energii – například NiMH baterii (přes Schottkyho diodu, aby proud netekl zpátky do článků), Li-Ion baterii (přes specializované moduly, určené k nabíjení takových baterií), nebo takzvaný Supercap (superkondenzátor, anglicky supercapacitor, s kapacitami v desítkách až stovkách Faradů).

Pozor na Li-Ion baterie – například oblíbeného typu 18650! Mají sice velkou kapacitu a spoustu dalších dobrých vlastností, ale jsou velmi citlivé na správné zacházení a na přesné nabíjení specifikovaným napětím a proudem. Pokud překročíte povolená nabíjecí napětí a proudy, začnou v bateriích probíhat chemické reakce, které mohou vyústit až ve zničení článku, popřípadě k jeho prasknutí, požáru a explozi – navíc lithium a jeho sloučeniny jsou poměrně agresivní.

39 Sigfox

39 Sigfox

Přemýšlel jsem, čím tuto knihu ukončit. Neříkám uzavřít – tato kniha nemá nic uzavřít, ale naopak spoustu věcí otevřít, naťuknout a nechat vás, abyste se sami rozhlédli, kam vás to nejvíc táhne.

Původní plán byl takový, že kniha skončí tím, že si postavíte vlastní počítač s procesorem Zilog Z80. Ale v průběhu psaní jsem na vás, čtenáře, myslel, a říkal jsem si: Proč vlastně tuhle knihu čtete? Chcete si stavět vlastní historické počítače (jako já; mě to velmi baví), nebo chcete stavět, co já vím, domácí automatizaci? A pak jsem si přiznal, že správně bude asi ta druhá možnost.

Samozřejmě se nevzdávám. Už teď si v hlavě sumíruju druhý díl, pracovně nazvaný „od breadbordu k VHDL", a v něm zájemcům prozradím, jak si navrhnout vlastní mikroprocesor, jak postavit vlastní počítač s osmibitovým procesorem, jak ho naprogramovat, jak ho emulovat… Ale taknějak cítím, že to asi nebude masová záležitost!

Takže mi bylo jasné, že někde u procesorů nabereme trochu jiný směr, spíš do té automatizace a IoT-iky. A v tom případě bude nejlíp ukončit kapitolu nějakou žhavou novinkou, něčím, co opravdu frčí… (A doufám, že se čtenáři v roce 2027 nebudou na adresu mého výběru poťouchle smát.)

Volba padla na komunikační technologii Sigfox.

39.1 Co je to Sigfox?

Sigfox je bezdrátová technologie pro posílání dat, původem z Francie, ovšem s čím dál větším pokrytím po celém světě. Na rozdíl od GSM a podobných má určité specifické rysy. Zaprvé: má dlouhý dosah. Český operátor SimpleCell staví ve spolupráci s T-Mobile základnové stanice, v současnosti (srpen 2017) mají pokrytou celou Českou republiku a pokrývají část Slovenska. Díky dlouhému dosahu na pokrytí vystačí poměrně málo stanic. V testovacím provozu byly k dispozici čtyři, umístěné v Praze, a pokrytá byla většina Prahy a část Středočeského kraje, ovšem podle testů bylo možné za příznivých okolností poslat zprávu až ze Sněžky. Operátor udává dosah 200 kilometrů na přímou viditelnost, 50 kilometrů v terénu a 2-5 kilometrů ve městě, samozřejmě v závislosti na kvalitě antény.

Dalším rysem je nízký výkon. Sigfox využívá pásmo 868 MHz, kde je povolený nejvyšší vysílací výkon 25 mW, což je zhruba výkon dálkového zamykání od aut. Přenosová rychlost je 100 bitů za sekundu a maximální velikost přenášeného paketu dat je 12 bajtů. Odvysílání jedné zprávy trvá 4-6 sekund a denně můžete poslat až 140 takových zpráv.

Poměrně kuriozní omezení vyplývá z faktu, že Sigfox využívá při přenosu velmi přesné časování (velmi úzké pásmo), takže se může stát, že pokud by se zařízení pohybovalo větší rychlostí (např. auto za jízdy po dálnici, vlak apod.), tak rušení, způsobené pohybem vůči přijímači, způsobí, že zpráva nebude doručena. Proto se například v zařízeních pro sledování polohy používá postup, kdy se vysílá pouze tehdy, pokud je zařízení v klidu.

Z popisu je jasné, že technologie není vhodná pro použití v oblastech, kde je potřeba neustále přenášet velké objemy dat. Vhodná je spíš pro statická zařízení, typu senzorů nebo zabezpečovacích zařízení, které v pravidelných intervalech posílají zprávy o stavu, popřípadě alerty o výjimečné situaci. Sigfox umožňuje kromě vysílání ze zařízení do sítě (upstream) i příjem zprávy (downstream). Příjem zprávy je možný pouze na vyžádání a proběhne vždy po vyslání zprávy. Není tak možné třeba na dálku „probudit" zařízení vysíláním nějaké zprávy. Downstream se používá typicky pro přenos nějakých konfiguračních informací. Přenos dat není nijak zaručen – dal by se přirovnat k síťovému protokolu UDP.

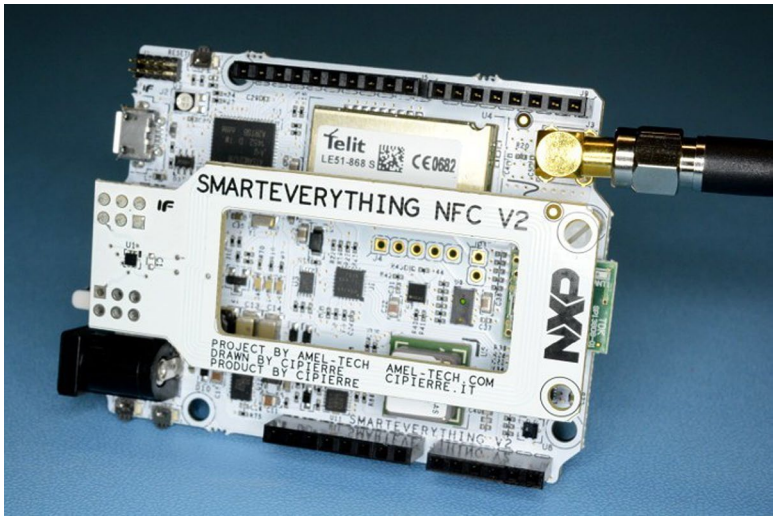
Každá zpráva je vyslána třikrát na třech náhodných frekvencích a putuje k vysílačům, které ji přijmou a zašlou na servery provozovatele. Ty ověří její validitu (např. podle toho, kolika vysílačům se podařilo zprávu zachytit) a zpřístupní zprávu klientovi. Zpráva je dostupná přes API a můžete si nastavit její předání na server klienta pomocí HTTP callbacku. Co se dál bude dít, je už plně na vás.

Sigfox má i řadu výhod. V první řadě je to nízká cena: moduly stojí řádově jednotky dolarů, samotná služba vyjde řádově na jednotky až desítky korun měsíčně. Druhou výhodou je velmi nízký odběr zařízení – na jednu baterii by mělo vydržet mnoho let. Navíc je spousta oblastí, kde není potřeba přenášet velké objemy dat. Ale i tam, kde se využívá jiná metoda bezdrátového přenosu, může být výhodné implementovat Sigfox jako záložní komunikační kanál, po kterém lze minimálně předat informaci, že se něco děje. V neposlední řadě je výhodou „globální roaming v ceně" – pokud se vaše zařízení ocitne v jiné zemi, kde má Sigfox pokrytí, funguje zcela bez problémů.

K dispozici jsou čipy i vývojové moduly (http://makers.sigfox.com/), zajímavou možnost představují moduly SiPy od výrobce Pycom (https://www.pycom.io/product/sipy/), různé shieldy pro Arduino, popřípadě vývojová deska Smart Everything, která je kompatibilní s Arduinem a obsahuje kromě modemu Sigfox i spoustu senzorů.

39.2 Cloudový teploměr se Sigfoxem

Pojďme tu vaši meteostanici posunout o další úroveň výš. Udělejme z ní meteostanici, připojenou do sítě Sigfox.



Můžete k tomu zvolit hned několik postupů. Můžete použít třeba desku SmartEverything (https://www.smarteverything.it/), která je vybavená snímačem teploty, tlaku, vlhkosti, osvětlení i modulem pro Sigfox. (Kromě toho má třeba i akcelerometry, gyroskop, Bluetooth nebo NFC, ale to nebudeme potřebovat).

K desce je dostupná celkem slušná dokumentace, a co je nejdůležitější – podpora pro Arduino IDE a spousta příkladů.

Začněte tím nejjednodušším příkladem, a tím je poslání obligátního Hello přes Sigfox. Jenže než to uděláte, je potřeba zaregistrovat vaše zařízení v rozhraní Sigfox (backend.sigfox.com). Postup není úplně přímočarý, naštěstí existují manuály a postupy (pokusím se jeden udržovat na webu elektrokniha.cz).

Po registraci na backendu Sigfox můžete zkusit poslat testovací zprávu, obligátní „Hello, world" – a nezapomeňte, že jedna zpráva Sigfox je dlouhá 12 bajtů, takže žádné velké vypisování se nekoná.

12 bajtů představuje 96 bitů informace, a je tedy jasné, že je na místě přemýšlet nad nějakou vhodnou kompresí dat. Posílat třeba teplotu jako posloupnost znaků „-25.3°C" znamená velkou neefektivitu.

Pro Sigfox je potřeba množství přenášených dat osekat na co nejmenší počet bitů. Ideálně převést na celá čísla a snížit možný rozsah. Například u tlaku – pokud budete posílat tlak v hPa, tak můžete buď posílat celá čísla v rozsahu 970 až 1060, což zabere 11 bitů, nebo můžete naměřenou hodnotu snížit např. o 950, čímž dostanete rozsah 20 až 110, s nímž se vejdete do sedmi bitů.

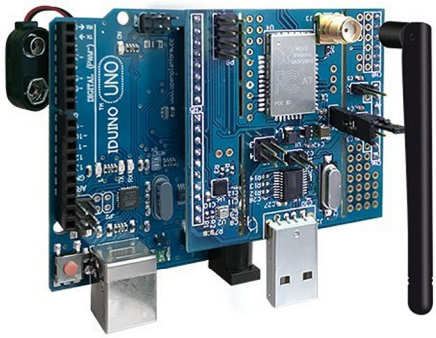
Kde jsem vzal čísla 970 a 1060? To jsou minimální a maximální hodnoty tlaku, naměřené na území Česka a Slovenska za celou dobu měření.

Totéž platí pro další veličiny. Například u teploty je rozumné zaokrouhlit ji na desetiny (a i tak pravděpodobně budete mimo garantovanou přesnost), tedy vynásobit hodnotu deseti a zahodit desetinnou část. A opět se hodí buď prostý posun, třeba udělat + 60, čímž se hodnota 0 stane teplotou - 60 °C. Další možnost je ušetřit ještě pár bitů a udělat kompresi adaptivní, například pro rozsahy 0 až 30 s přesností na jedno desetinné místo, pro teploty mimo tento rozsah třeba s přesností ± 2 °C – ale vždy záleží na konkrétní aplikaci a na tom, co můžete zanedbat a co je naopak nutné přenést.

Po přenesení naměřených hodnot nezapomeňte procesor i desku uspat pomocí příslušných funkcí – nemá smysl, aby celý systém běžel v době, kdy se neměří. Vzhledem k tomu, že se Sigfoxem můžete poslat jednu zprávu každých 10 minut, tak desku klidně uspěte na celých deset minut, po probuzení změřte všechny hodnoty, odešlete, a zase nechte desku spát.

Další možnost, místo poměrně drahé desky SmartEverything, je použít třeba devkit od Thinxtra:

Tento devkit můžete připojit k různým kitům, nejen k Arduinu, a dokonce i přímo k PC přes



USB. Pokud ho připojíte ke své meteostanici na Arduinu (pamatujete? DHT22 a BMP180), můžete snadno místo Serial.println() posílat data via Sigfox na server k dalšímu zpracování. Ale nezapomeňte opět na dlouhý a vydatný spánek, alespoň těch 10 minut.

Sigfox jako takový umí být opravdu low power. Pokud minimalizujete množství energie, které se propálí zbytečně, například ve stabilizátorech nebo v LED, pokud budete uspávat tak, jak máte, a probouzet jen na nezbytně nutnou dobu, získáte zařízení, schopné práce na baterii po dobu několika desítek měsíců.

39.3 Co s daty v Sigfoxu?

Veškerá data, co přes Sigfox pošlete, skončí v backendu této služby – je to centralizovaná cloudová služba, a všechna data protékají skrz server provozovatele. Za normálních okolností tam skončí, a vy si je můžete přečíst nebo exportovat. Ovšem backend Sigfox nabízí i možnost zprávy přeposlat – používá k tomu zase jednoduché http(s) volání GET s možností předat data surová, nebo jednoduše parsovaná. K tomu získáte i informace o pozici vysílače, o síle signálu, přesném čase vyslání zprávy atd.

Sigfox umožňuje i zpětný přenos dat, ale nikde není zaručeno, že se data na zařízení opravdu dostanou. Zpětný přenos funguje pouze jako doplněk vysílání (de facto odpověď na vysílání). Je vhodný maximálně pro nějaké servisní informace, přepínání módů apod. Rozhodně nemůže fungovat jako regulerní zpětný kanál, např. pro ovládání zařízení.

Důležité u sítě Sigfox je, že není garantováno doručení zprávy. Zpráva se vysílá několikrát, aby se zvýšila pravděpodobnost, že bude zachycena, ale pokud se tak nestane, nikdo se to nedozví – ani zařízení, ani příjemce.

Schéma a zdrojový kód najdete na ⌀ https://eknh.cz/sigfox

40 Šťastnou cestu...

40 Šťastnou cestu…

Naše společná cesta světem mikroelektroniky se tady rozchází, ovšem doufám, že nekončí. Věřím, že pro vás to byly jen první nesmělé krůčky, a snad to nebyl ztracený čas. Že teď už víte, jak ty cool elektronické obvody stavět – a nejen to! Už umíte číst schémata, umíte je i nakreslit, umíte si poskládat složitější obvody ze základních součástek, a hlavně: nebojíte se, že něco zkazíte.

Samozřejmě, že něco zkazíte. Ale každý, když se učil a začínal, spoustu věcí zkazil, bez toho to nejde. Nesmíte se tím nechat zviklat, a nesmíte si nechat svou snahu znechutit od lidí, co si tímtéž prošli tak dávno, že už zapomněli, jaké byly jejich vlastní začátky, a dnes se na začátečníky dívají skrz prsty.

Jestli vás číslicová technika zaujala, tak tato kniha splnila svůj účel. Teď už je to na vás a na vašem dalším zájmu. Ještě toho musíte spoustu nastudovat, spoustu si toho zjistíte sami, a třeba se číslicová technika stane vaším koníčkem. Dobrým začátkem mohou být webové stránky k této knize – https://www.elektrokniha.cz.

Já vám přeju, ať toho po cestě pokazíte co nejmíň. A k tomu vám dopomáhej následující desatero:

- **Pozor na zkraty!** Někdy to jen tak lehce zajiskří, nebo se jen zahřeje rezistor, a největší škoda je jen vysoký odběr nebo vybitá baterie. Ovšem většinou bývá hůř, něco se začne pálit, a pak může snadno dojít k neštěstí. A to platí i pro „neškodných pět voltů".

- **LED vždy s rezistorem!** Ano, existují situace, kdy není nutný, ale vy ho prosím pro jistotu vždy zapojte, nic tím nezkazíte.

- **Než něco v obvodu změníte, vždy ho nejdřív vypněte!** Přepojování „za živa" může velmi snadno vyústit v problém a zničení součástek.

- **Pozor na kondenzátory!** Když vypnete obvod, neznamená to vždy stoprocentně, že v něm není nikde žádné napětí. Pokud jsou v obvodech kondenzátory, může být napětí přítomno ještě několik sekund, někdy i desítek sekund, po vypnutí napájení.

- **Pořiďte si multimetr a měřte!** Sice jsem zažil starého ostříleného elektrikáře, který měl fázovku za zbytečnost a zkoušel přítomnost napětí ve vodiči krátkým dotekem prstu („*Když to uděláš rychle, tak to jen zabrní*" – ale on taky docela dost pil…), ale vy to fakt nedělejte. Změřením snadno poznáte, jestli je všude to, co tam má být, nebo se někdy dozvíte i úplně zajímavé věci o obvodech. Navíc změřit rezistor bývá rychlejší než dešifrovat proužkový kód. No a v neposlední řadě ta chvilka přemýšlení nad naměřenou hodnotou může zachránit cenný obvod.

- **Uzemněte se a vývodů citlivých součástek se nedotýkejte prsty**. Dvacetkrát se nic nestane, po jednadvacáté zapojíte tranzistor MOSFET, a nic, obvod nefunguje. Dlouho pak zkoumáte, čím to je, až zjistíte, že je tranzistor zničený.

- **Kreslete schémata jednoduchá a přehledná.** Nejde o evangelium a přesné dodržování kánonů – když dává smysl, aby nějaký signál postupoval zprava doleva a nahoru, tak to tak nakreslete, ale pouze když to dává smysl! Přehlednost schématu je důležitější věc, než rigidní lpění na poučkách. U číslicové techniky používejte i schematické značky pro sběrnice, ušetří hodně místa ve schématu.

- **Motory, relé, elektromagnety, serva a další součástky, v nichž jsou cívky, připojujte k číslicovým obvodům vždy s ochrannou diodou a přes tranzistorový budič.** K tomu asi není zapotřebí nic dodávat.

- **Pozor na součástky pro 3,3voltovou logiku!** Některé komponenty či moduly jsou určené pro napájecí napětí 3,3 voltu. Pokud je zbytek obvodu navržen pro 5 V napájení, budete muset pravděpodobně použít převodníky úrovní. Výjimkou jsou takzvané „5 V tolerantní" součástky.

- **Pozor na společnou zem!** Když spojujete dva obvody, dbejte na to, abyste vždy spojili jejich „zem" – tedy většinou záporný pól napájení. Bez tohoto propojení nebudou mít obvody jasno v tom, co je dohodnutá nulová úroveň napětí, a spojení nebude fungovat!

Přílohy

Přílohy

Nástroje a weby

Pro „experimentování bez součástek" se mohou hodit emulátory, jednak CircuitJS, jednak SimCir (pro digitální obvody). Odkazy na oba emulátory najdete na

https://elektrokniha.github.io/

Tamtéž najdete i další tipy a pomůcky, včetně obrázků z knihy, odkazů na zdrojové kódy a dalších užitečných věcí.

Až se dostanete s elektronikou dál a budete přemýšlet nad vhodným nástrojem pro kreslení schémat, máte na výběr z mnoha variant. Nejlepším řešením pro offline práci jsou programy Eagle či Altium. Pro online navrhování, tedy přímo v prohlížeči, můžete vyzkoušet například http://www.schematics.com, https://easyeda.com/ nebo sadu nástrojů od Autodesku: https://circuits.io, kde naleznete i emulátor (Lab) a nástroje pro tvorbu desek plošných spojů (PCB).

Pro to nejjednodušší kreslení obvodů lze použít i program Fritzing. Jeho výhodou je, že v jeho knihovnách naleznete naprostou většinu amatérsky použitelných modulů a součástek. Navíc s ním můžete kreslit nejen schémata, ale i „zapojovací náčrtky", kterými jsem doprovodil tuto knihu.

Nákupní seznam začínajícího hobby elektronika

- **Nepájivé kontaktní pole** (Breadboard)
- **Propojovací vodiče** (Dupont wires, male-to-male wires, breadboard wires, jumper wires). Kromě variant „M-M„ (na obou koncích vodiče) nakupte i variantu M-F (male-female, na jednom konci zdířka, na druhém vodič).
- **Napájecí modul pro kontaktní pole.**
- **Napájecí zdroj 5 V / 2 A s USB výstupem.**
- **Napájecí zdroj 7-12 V / 1,5 A.** Pokud koupíte s možností přepínání napětí, bude to lepší.
- **Měřicí přístroj.** Do začátku stačí ten nejlevnější univerzální, ale je to nutnost! Bez měření není vědění!
- **Nářadí:** Šroubovák křížový i plochý. Pinzeta. Ostrý nůž (ulamovací je dobrý, já si pořídil lékařský skalpel – surgical scalpel). Kleště štípačky.
- **LED.** Aspoň deset, různé barvy, budou se hodit. Kupujte ty s průměrem pouzdra 5 mm.
- **Sedmisegmentovky** (7-segment LED display). Kupte si jich třeba pět, zvolte typ „se společnou katodou„ (common cathode). Na barvě ani velikosti nezáleží. Pro pokusy zvolte spíš sedmisegmentovky samostatné, k Arduinu a jiné elektronice kupte spíš hotový modul (LED display module driver, LED display module SPI apod.).
- **Rezistory** (Resistor). Do začátku si kupte klidně ty nejlevnější. Kupte hodnoty 220R, 330R, 470R, 1K, 2K2, 3K3, 4K7, 6K8, 10K, 22K, 68K, 100K a 1M. Od těch nižších hodnot doporučuju třeba 10 kusů od každé (220R vezměte klidně 20), od 10K výše vezměte pět kousků. Později si pořiďte třeba kompletní řadu E6.
- **Kondenzátory** (Capacitor). Vezměte keramické 33 pF a 100 nF, od každé hodnoty deset kousků. Vezměte elektrolytické 1μF, 4,7 μF a 10 μF.
- **Diody** (Diode). 1N4004 / 1N4007 je taková „vanilla diode„, standardní křemíková usměrňovací dioda, kterou použijete do různých konstrukcí, protože snese i velká napětí. Dále 1N4148, což je „dioda první volby„ pro zpracování signálů (je rychlá), a k tomu i 1N5817 (nebo 1N5818, 1N5819) – rychlá Schottkyho dioda s nízkým úbytkem napětí.

- **Tranzistory** (Transistor). Kupte si jeden NPN a jeden PNP typ, třeba BC337 (NPN) a BC557 (PNP), od každého třeba 10 kusů. Popřípadě můžete koupit typ 2N3904 (NPN) / 2N3906 (PNP). Pro buzení zátěží si kupte nějaký výkonový, třeba TIP120, a výkonový MOSFET, např. FQP30N06L.

- **Integrované obvody** (IC, integrated circuit):

    - **Standardní hradla TTL** (TTL gates): 7400 (4x NAND), 7402 (4x NOR), 7486 (4x XOR), 7404 (6x invertor), 7414 (6x invertor se Schmittovým klopným obvodem), 7405 (6x invertor s otevřeným kolektorem), 7408 (4x AND). Od každého typu vezměte tak pět kousků. Volte řadu HCT (74HCT00, …) popřípadě ALS.

    - **Komplexnější obvody TTL:** 7474 (2x klopný obvod D), 7475 (2x 2bit latch), 7490 (4bitový desítkový čítač), 7493 (4bitový binární čítač), 74138 (dekodér 1-z-8), 74595 (8bitový posuvný registr/buffer SIPO), 74162 (8bitový posuvný registr PISO).

    - **Sériové paměti:** 24C01 (sériová EEPROM 128 byte, I^2C), 23LC512 (SPI sériová SRAM).

- **Arduino Uno.** Pro začátečníka nezbytnost, a doporučím: pokud si kupujete své první, kupte si originální od distributora. Některé kusy levných klonů vykazují podivné chyby, které by vás mátly.

- **Moduly, breakouty.** Tady nezbyde než sáhnout po čínských prodejcích a online obchodech, v českých kamenných nejspíš štěstí mít nebudete. Na druhou stranu právě tyto moduly jsou často velmi zajímavé. Nemusíte kupovat všechno, ale něco z toho pro vás bude určitě užitečné.

    - **HC-SR04.** Ultrazvukový měřič vzdálenosti. (Ultrasonic range detector).

    - **Senzor vlhkosti půdy** (Moisture soil sensor).

    - **PIR senzor** (PIR motion detector).

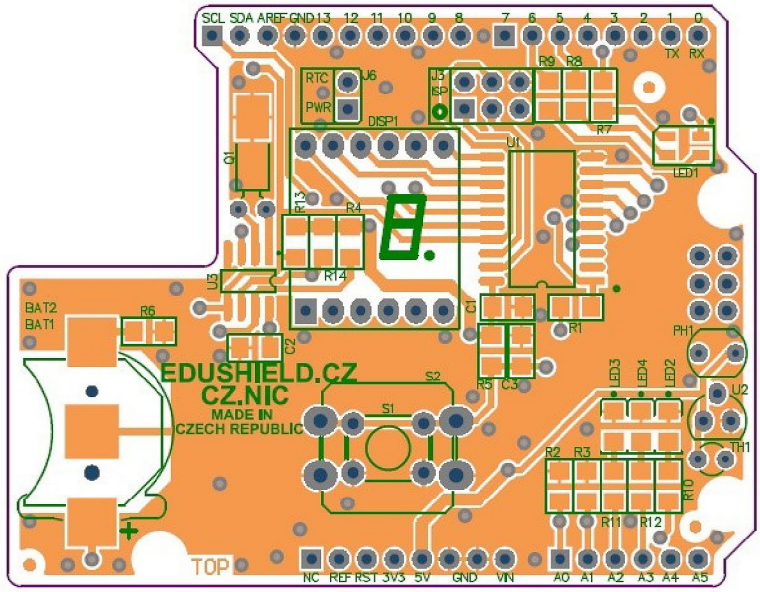
    - **Sedmisegmentový displej pro Arduino** (7segment LED display Arduino module). Zkuste klíčová slova MAX7219 nebo TM1638.

    - **Teploměr LM75A** (LM75A module, LM75A breakout).

    - **Vlhkoměr DHT22** (DHT22 module, DHT22 breakout).

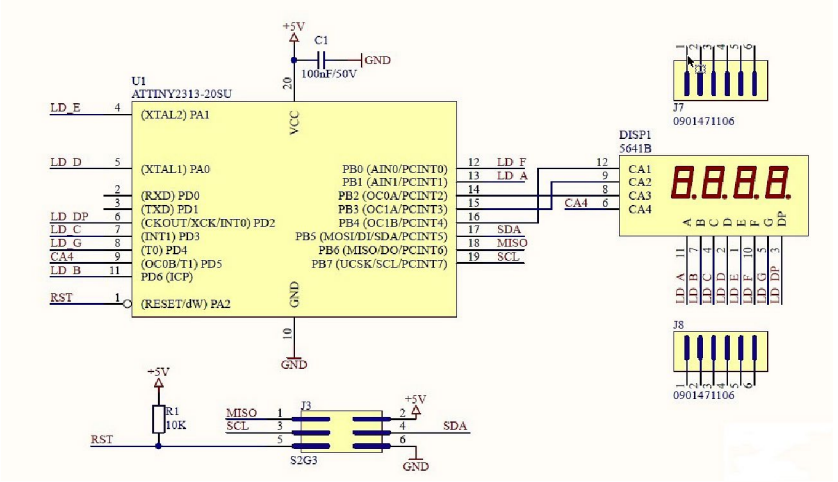
- **Vlhkoměr SHT30** (SHT30 module) – o něco dražší než předchozí, ale přesnější a se standardním rozhraním.
- **Barometr BMP180** (BMP180 module).
- **Bezdrátové moduly 433 MHz** (433 MHz transmitter, 433 MHz receiver; with antenna).
- **Bezdrátový komunikační modul nRF24L01+.** Alespoň dva. Kupte k nim i redukci na 5 V (nrf24l01 adapter).
- **Alfanumerický displej 16 znaků / 2 řádky** (1602 display LED).
- **Redukci k tomuto displeji na I²C** (1602 I²C expander).
- **OLED displej 128 × 64** (OLED display module 128 × 64).
- **Joystick** (Joystick module).
- **Rotační enkodér** (rotary encoder module).
- **NodeMCU** To pokud budete chtít trošku vážněji experimentovat s WiFi. Je to deska koncepčně podobná Arduinu, můžete k ní připojovat leccos, cena není vysoká, a programovat to lze jak z Arduino IDE, tak třeba v MicroPythonu.

EduShield



Použité piny na Arduino

Pin	Funkce	Pin	Funkce
D2 (INT)	Button	A0(D14)	Fotorezistor
D3 (INT)	RTC SQW out	A1(D15)	Termistor
D4	---	A2(D16)	Red LED
D5 (PWM)	RGB LED – G	A3(D17)	Orange LED
D6 (PWM)	RGB LED – B	A4(D18)	I²C SDA pin
D7	---	A5(D19)	I²C SCK pin
D8	---		
D9 (PWM)	RGB LED – R		
D10	---		
D11	---		
D12	---		
D13	Green LED		



Přiřazení pinů v ATtiny2313 k vývodům na konektoru pro displej (pin 1 je vlevo dole):

AVR

PB4	**PB1**	**PB0**	**PB3**	**PB2**	**PD6**
PA1	PA0	PD2	PD3	PD4	PD5

Arduino

D13	**D10**	**D9**	**D12**	**D11**	**D8**
D2	D3	D4	D5	D6	D7

LED (CA označují pozice, SEG jednotlivé segmenty)

CA1	**SEGA**	**SEGF**	**CA2**	**CA3**	**SEGB**
SEGE	SEGD	SEGH(DP)	SEGC	SEGG	CA4

Nahrání firmware do EduShieldu

V EduShieldu je displej řízen pomocí ATtiny2313. V něm je firmware, napsaný ve Wiring. Tento firmware se dá jednoduše nahrát pomocí Arduina a Arduino IDE.

1. Aktualizujte Arduino IDE na nejnovější verzi.

2. Stáhněte si aktuální software pro EduShield (https://github.com/arduino-edushield/edushield) – můžete použít i Správce knihoven v Arduino IDE.

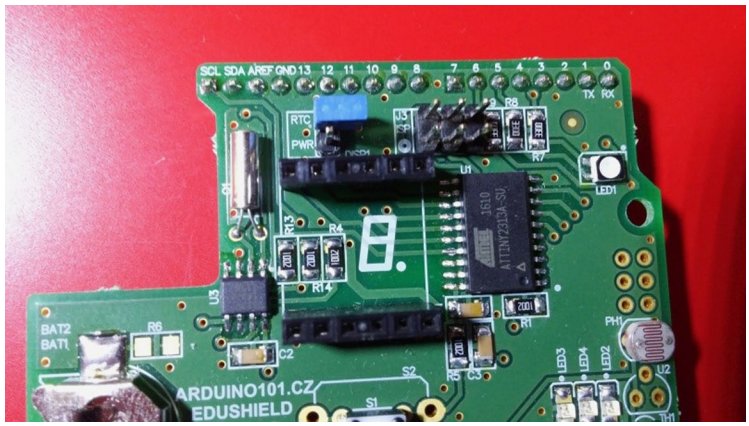
3. Ve složce _firmware naleznete podsložky „hardware“, „libraries“ a „tiny2313“. Obsah složek „hardware“ a „libraries“ zkopírujte do pracovního adresáře Arduina (nejčastěji v domovském adresáři, podsložka Arduino). Složka „libraries“ bude pravděpodobně existovat, „hardware“ možná ne, tak jej vytvořte.

4. Spusťte Arduino IDE a připojte Arduino Uno, kterým budete programovat. **Bez EduShieldu!**

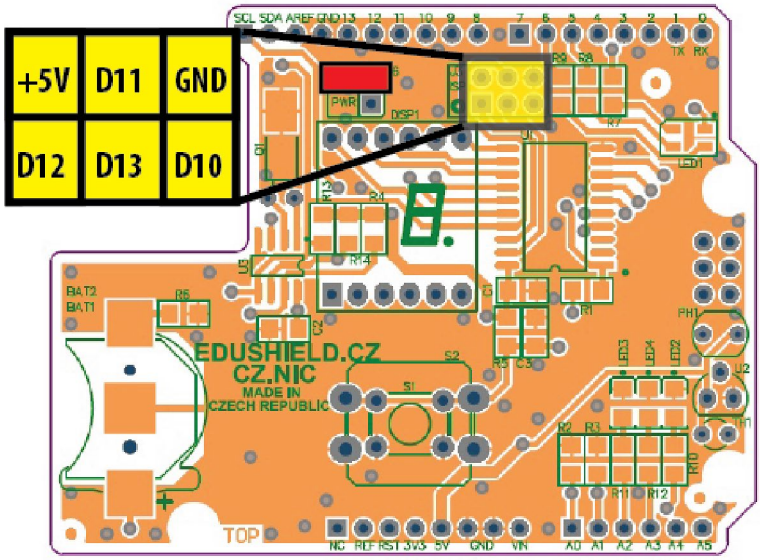
5. Z menu „Soubor – příklady“ vyberte „Arduino ISP“ a běžným způsobem jej nahrajte do Arduina.

6. Z EduShieldu sundejte displej.

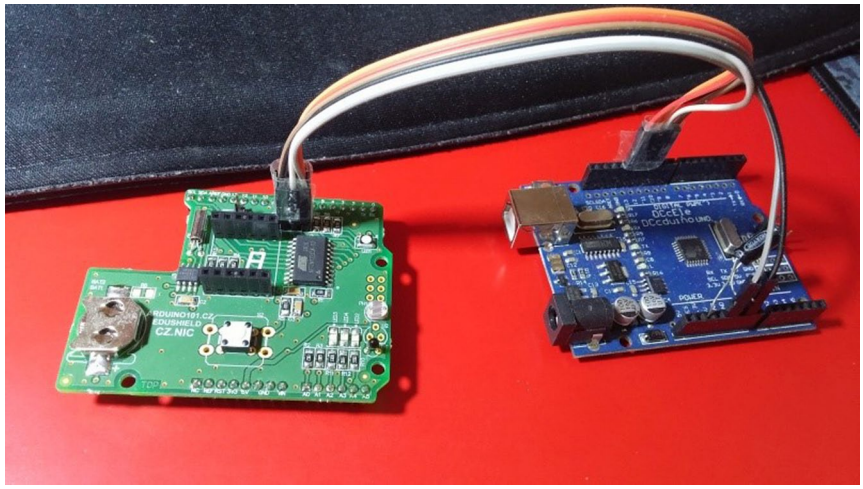
7. Switch J6 nad displejem rozpojte, viz obrázek (to je to modré nahoře pod piny 12, 11, označené RTC PWR):



8. Propojte pomocí šesti propojovacích vodičů EduShield (šestivývodový konektor označený J3 ISP) s Arduinem (+ 5 V, GND, datové piny 10, 11, 12 a 13). Správné propojení je naznačeno na následujícím obrázku:

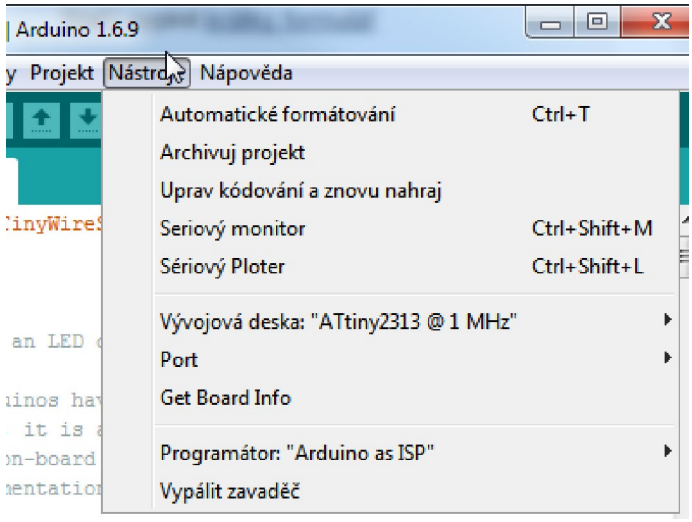


9. Propojené komponenty by měly vypadat zhruba takto:



10. Spusťte Arduino IDE a otevřete sketch Tiny2313 ze složky _firmware.

11. Vyberte jako desku „ATtiny2313 @ 1 MHz“ a jako programátor „Arduino as ISP“, viz screenshot:



12. Přeložte a spusťte nahrávání.

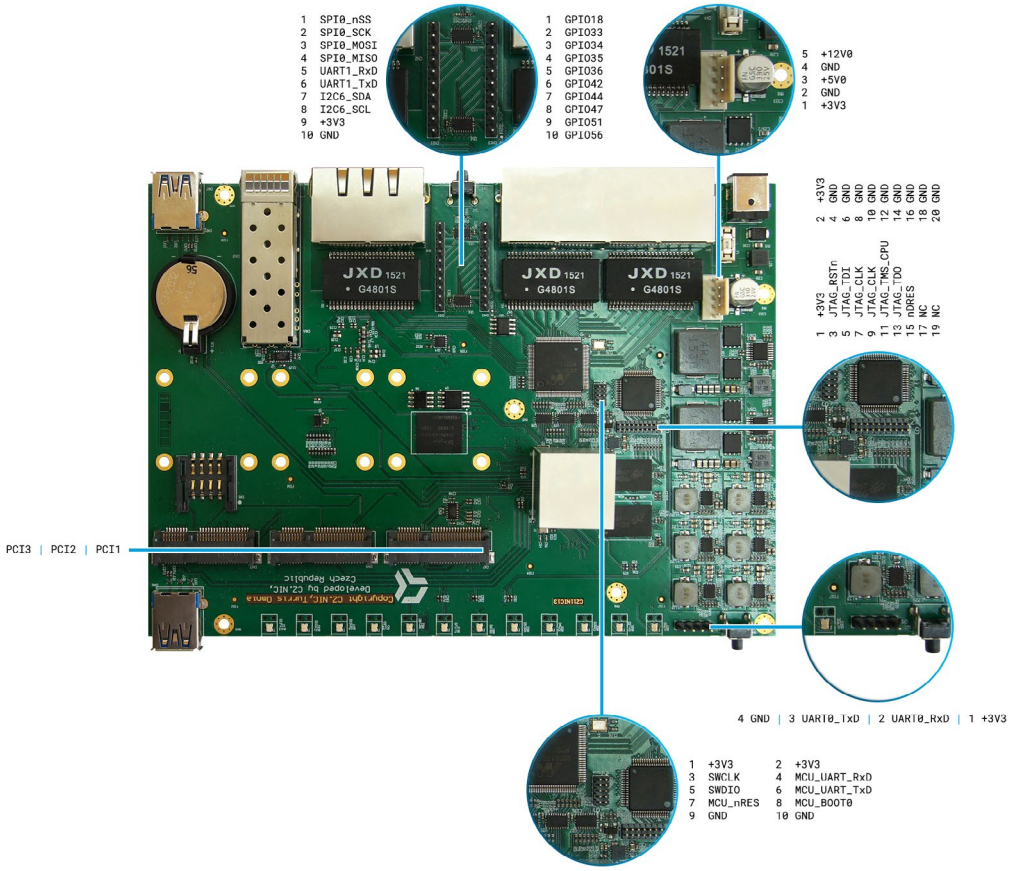
13. Po úspěšném nahrání odpojte EduShield od Arduina.

14. Vraťte zpátky switch J6 (musí spojovat oba vývody) a nasaďte displej.

Turris Omnia pro experimenty s elektronikou

Sdružení CZ.NIC, vydavatel této knihy, vyrábí a dodává routery s označením Turris Omnia. Tyto routery jsou navrženy jako jednodeskové počítače s procesorem ARM, na kterých běží distribuce Linuxu, založená na OpenWrt. Na rozdíl od většiny routerů, které jsou koncipovány jako uzavřené zařízení, jsou routery Turris otevřené. Díky této otevřenosti s nimi můžete pracovat jako třeba s Raspberry Pi – pomocí SSH konzoly se k nim připojíte jako k jinému počítači s Linuxem, a můžete instalovat balíčky, spouštět programy, nastavovat konfiguraci (ale nezapomeňte, že při tom všem musí router ještě vyřizovat veškerý internetový provoz)…

Otevřená koncepce vedla konstruktéry k tomu, že u první várky Omnií, připravené především pro kampaň na IndieGoGo, vyvedli některé signály z procesoru na speciální aplikační konektor (viz https://www.turris.cz/doc/_media/omnia-pinout.png):



Těmito výstupy jsou vybaveny 1 GB modely, v době psaní knihy jsou stále ještě k dispozici. Do budoucna se budou dodávat pouze 2 GB verze.

Vidíte, že máte k dispozici napájecí napětí 3,3 V (dejte pozor a nepřipojte omylem zařízení, které posílá 5 voltů), 10 univerzálních pinů (GPIO), sběrnici SPI, sběrnici I2C a sériové rozhraní UART.

Nejjednodušší použití těchto portů nabízí programovací jazyk Python. Viz dokumentace (https://www.turris.cz/doc/cs/howto/gpio). Pokud chcete začít klasickým „LED BLINK" příkladem, můžete postupovat podle tohoto návodu.

Připojte LED mezi pin GPIO18 a zem. Nezapomeňte na omezovací rezistor! U Turrisu Omnia můžete odebírat proud z GPIO výstupu max. 5 mA, takže rezistor by měl mít minimálně 660 ohmů. Zvolte ale vyšší, třeba 2k2.

Pomocí kódu v Pythonu pak můžete bliknout:

```
import turris_gpio as gpio
gpio.setmode(gpio.BCM)
gpio.setup(18, gpio.OUT)
gpio.output(18, True)
time.sleep(10)
gpio.output(18, False)
gpio.cleanup()
```

Další příklady naleznete v dokumentaci.

K Omnii lze připojit například jeden bezdrátový modul nRF24L01 (přes rozhraní SPI) a vytvořit si tak „domácí bezdrátový hub", s nímž budou komunikovat další čidla, používající tentýž typ modulů.

Bohužel je ve firmware, který je dostupný v době psaní knihy, zakázané rozhraní SPI. K jeho použití je zapotřebí upravený soubor /boot/dtb. Do budoucna by to mělo být jinak, proto doporučuju intenzivně sledovat aktuální verzi dokumentace. Existuje i řešení, které nasimuluje SPI rozhraní na volných GPIO portech (http://blog.jfila.cz/2014/11/zprovozneni-spi-na-openwrt.html)

Turris Omnia tedy umožňuje připojení různých senzorů a komunikačních zařízení „napřímo". Pomocí vlastních obslužných programů pak s těmito senzory můžete pracovat. Podpora a dokumentace jsou ale v době psaní této knihy poměrně skoupé na informace a pro úplného začátečníka bych proto Omnii jako platformu na experimenty nedoporučoval.

Karnaughova mapa

Myslím si, že Karnaughova mapa je užitečná pomůcka, když se snažíte přijít na to, jak poskládat ze základních hradel AND, OR a NOT nějakou logickou funkci, kterou máte zadanou například ve formě tabulky, někdy i neúplné, a není to na první pohled patrné. Ideální jsou Karnaughovy mapy pro hledání funkcí tří nebo čtyř proměnných.

Představte si, že máte zadanou funkci čtyř proměnných pomocí tabulky:

D	C	B	A	Y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

Možná vidíte pravidelnost ve výsledcích a dokážete si logickou funkci odvodit sami. Ale pokud ne (a pravděpodobně bez cviku nevidíte nic), tak nastává čas pro kreslení mapy.

Mapa má tolik políček, aby pokryla všechny kombinace – tedy pro čtyři proměnné to je 16 polí v matici 4 × 4. Sloupce budou řízené podle proměnných A a B, řádky podle C a D, ovšem platí jedno důležité pravidlo: každý řádek (sloupec) se liší od svých sousedních řádků (sloupců) liší pouze v hodnotě jedné proměnné: 00 – 01 – 11 – 10

		A	0	1	1	0
		B	0	0	1	1
D	C					
0	0		1	1	1	1
0	1		0	1	1	0
1	1		0	1	1	0
1	0		1	1	1	1

Do mapy zapíšete požadované hodnoty – viz tabulka nahoře. Teď musíte najít takzvané „smyčky" – oblasti se samými jedničkami, které zabírají dvě, čtyři nebo osm sousedících buněk. Ano, trochu to připomíná Tetris, ale ty tvary jsou jen „úsečka" a „obdélník". Snažte se najít co největší plochy. Smyčky se mohou překrývat.

Při hledání smyček můžete přecházet i přes hrany tabulky – jako by se jednalo o nekonečné opakování, takže například v hořejší tabulce můžete udělat smyčku osmi prvků, která zahrnuje první a poslední řádek.

Když máte smyčky hotové a pokryté všechny jedničky, můžete každé smyčce přiřadit její výraz. Například pro smyčku, tvořenou prvním řádkem (CD = 00) by to bylo „NOT C AND NOT D" – tedy pokud je C = 0 a zároveň D = 0, je výsledek 1.

1	1	1	1
0	1	1	0
0	1	1	0
1	1	1	1

Smyčka čtyř buněk v posledním řádku je D = 1 AND C = 0, tedy D AND NOT C.

1	1	1	1
0	1	1	0
0	1	1	0
1	1	1	1

Smyčka tvořená prostředními dvěma sloupci platí pro stavy AB = 10 a 11. Protože se B mění, můžeme ho zanedbat a říct jen, že výsledek bude 1, když je A = 1.

1	1	1	1
0	1	1	0
0	1	1	0
1	1	1	1

Je výhodné spojit první a poslední řádek do smyčky. Pak platí, že smyčka je aktivní pro C = 0 (D se mění, ignorujeme ho). Pro takovou smyčku tedy platí výraz NOT C.

1	1	1	1
0	1	1	0
0	1	1	0
1	1	1	1

No a teď můžeme jednotlivé výrazy spojit pomocí OR: A OR NOT C

Jednoduché to bylo? Dobře, zkuste složitější:

1	1	1	1
0	1	1	0
0	1	1	0
1	1	0	0

Máte? Správnou odpověď najdete níže.

Karnaughova mapa se hodí i v situacích, kdy funkce není zadána úplně, respektive zadavateli je jedno, jestli při určité kombinaci bude na výstupu 0, nebo 1. Pak takové nezadané hodnoty zapíšete jako X a můžete si vybrat: Když se vám hodí takové políčko do smyčky, tak ho zahrňte a považujte za jedničku, když se nikam nehodí, považujte ho za 0 a ignorujte ho.

X	1	X	1
0	1	1	0
0	0	1	0
X	0	1	0

V takovémto případě zvolím jednu smyčku v prvním řádku (NOT C AND NOT D), jednu smyčku ve třetím sloupci (A AND B) a jednu smyčku 2 × 2 v horní části tabulky (NOT D AND A)

A správná odpověď na testovací příklad? Zvolil bych tyto smyčky: NOT C AND NOT B, NOT C AND NOT D a C AND A

„Dobré rady nad zlato" na jednom místě

Lékaři mají něco, čemu se říká „lék první volby". Když přijdete s příznaky třeba bakteriální infekce, tak lékař dřív, než začne zkoumat, který že bakteriální kmen vás nakazil, nasadí takzvané „antibiotikum první volby", což je takové, které je účinné proti naprosté většině možných bakterií, a velmi pravděpodobně pomůže. Pokud ne, pak je potřeba hledat jiné, specifické řešení. Totéž platí i pro mnoho dalších nemocí: jsou na ně „léky první volby", které může lékař předepsat, protože je velká pravděpodobnost, že na většinu případů pomůžou.

Zkusil jsem sepsat několik „elektronických léků první volby". Není to nijak exaktní, jsou to „babské rady", ale pro návrh prototypu budou stačit.

- Rezistor k LED: pro 5 V logiku dejte 330R, tím nic nezkazíte.
- Pull-up rezistor: dejte 10k, to by mělo pro běžné použití stačit.
- Spínání většího proudu, například pro žárovku: použijte MOSFET-N nebo tranzistory v Darlingtonově zapojení.
- Kde je cívka (motor, elektromagnet), tam je potřeba ochranná dioda!
- Spínání většího napětí: Použijte relé. Nezapomeňte na ochrannou diodu (je tam cívka!) a na buzení přes tranzistor (potřebuje velký proud).
- Pro spínání motorů se bude hodit H-můstek.
- Spousta zajímavých součástek se dělá v malých pouzdrech, která je nutné pájet. Naštěstí existují „breakout" moduly, kde jsou tyhle součástky s nezbytnou bižuterií už připravené.
- Hradlo OR: můžete udělat montážní s diodami, ideálně Schottkyho.
- Chlazení: Chladit je potřeba zátěž, kterou teče velký proud, popřípadě složité obvody, které pracují na vysokých frekvencích. Empirické pravidlo: když na tom udržíte prst, chladit nemusíte. Typicky bude chlazení potřebovat výkonný stabilizátor, spínací tranzistory pro velké proudy, popřípadě procesor s vysokou frekvencí.
- K napájení integrovaných obvodů připojujte paralelně blokovací kondenzátor 100n, co nejblíž k pouzdru.
- Čím delší vodič, tím nižší frekvence! 10 cm vodič bez problémů přenese i megahertz, 30 cm spíš ne.

- Testujte: obvody skládejte po menších částech, u každé si ověřte, že funguje. Arduino se vám bude hodit jako „testbench“ – snadno nastavíte signály a jejich průběhy a zjistíte výsledek.

- Po zapojení té nejdůležitější části najednou přestává pracovat zbytek? Když má sepnout motor, tak se to celé resetuje? Pravděpodobně málo proudu! Dejte silnější zdroj.

- Některé nárazové odběry můžete pokrýt velkým kondenzátorem (desítky μF) paralelně ke zdroji.

- Měření tepla: pokud tam máte Arduino, tak nejjednodušší je měřit čidlem DS18B20, je levné a je ho všude dost.

- Termistor: Naměřenou teplotu nemusíte přepočítávat na stupně, pokud potřebujete nastavit jen nějaký práh sepnutí. Prostě jen nastavte stejnou teplotu a změřte jeho odpor, resp. napětí na něm.

HRADLA, VOLTY, JEDNOČIPY
Úvod do bastlení

Martin Malý

Vydavatel:
CZ.NIC, z. s. p. o.
Milešovská 5, 130 00 Praha 3
Edice CZ.NIC
www.nic.cz

1. vydání, Praha 2017
Kniha vyšla jako 16. publikace v Edici CZ.NIC.
Sazba: Vojtěch Cejnek (FRVR)

© 2017 Martin Malý
Toto autorské dílo podléhá licenci Creative Commons
(http://creativecommons.org/licenses/by-nd/3.0/cz/),
a to za předpokladu, že zůstane zachováno označení autora díla a prvního vydavatele díla, sdružení CZ.NIC, z. s. p. o. Dílo může být překládáno a následně šířeno v písemné či elektronické formě na území kteréhokoliv státu.

ISBN 978-80-88168-23-2 (tištěná verze)
ISBN 978-80-88168-24-9 (ve formátu EPUB)
ISBN 978-80-88168-25-6 (ve formátu MOBI)
ISBN 978-80-88168-26-3 (ve formátu PDF)

cz.nic | SPRÁVCE DOMÉNY CZ

Edice CZ.NIC

O knize Kniha je určena pro zájemce o elektroniku, číslicovou techniku a vlastní stavbu zařízení pro Internet věcí. U čtenáře předpokládá běžné zkušenosti s počítačovou technikou. Znalost programování není nutná, i když ji v některých příkladech čtenář využije. Výklad pokrývá oblast od naprostých základů (napětí, proud, odpor, Ohmův zákon) přes základní elektronické součástky a řadu číslicových obvodů TTL (74xx) až po specializované součástky, jako jsou čidla, polovodičové paměti a jednočipové počítače. Kniha se nevyhýbá ani tématům a příkladům, spojeným s bezdrátovou komunikací a Internetem věcí. Čtenář tak získá rychlý vhled do problematiky z praktické stránky, a k němu i nezbytnou porci teorie.

Kniha vznikla jako odpověď na otázky, které padaly při školení Arduino a EduShield. „A proč to tlačítko má u sebe rezistor?“ - „Proč nemůžeme zapojit motor přímo k jednočipu? Já vím, že nemůžu, ale je nějaká jednoduchá odpověď na otázku proč?“ - „Jak bych měl zapojit servo?“ Nejčastější typ tazatele je programátor, co vzápětí dodá: „Víte, já to dokážu naprogramovat, to chápu, ale nerozumím tomu, proč mi nesvítí žárovka, když ji tam připojím, i když LEDka svítí. Já si to dokážu poskládat podle návodu, ale nedokážu to vymyslet, nevím, jak ten autor přišel na to, že zrovna tady má být odpor 10 k...“

Cílem knihy není udělat ze čtenáře odborníka na návrh elektronických konstrukcí, ale poskytnout mu základní informace o tom, jak tato technika vlastně funguje, poodkrýt oponu tajemství, která kolem číslicové techniky panuje, a ukázat, že i doma na stole lze sestavit zařízení, které možná nesnese srovnání s průmyslově vyráběnou elektronikou, ale přinese svému staviteli mnohem větší radost a potěšení.

Pro čtenáře této knihy vznikl i web elektrokniha.cz, kde jsou materiály ke stažení, aktualizované příklady a zdrojové kódy, nebo simulátor elektronických zapojení, kde si mohou zájemci vyzkoušet nejrůznější zapojení a schémata.

O autorovi **Martin Malý** (známý též pod přezdívkou Arthur Dent či Adent), je český programátor, bloger, publicista a komentátor. V roce 2003 naprogramoval a spustil veřejnou blogovací službu Bloguje.cz, od roku 2007 psal pravidelný sloupek pro Digiweb - součást webu iHNed.cz. Později pracoval na pozici šéfredaktora webového magazínu Zdroják (vydávala společnost Internet Info, s.r.o.). V březnu 2013 nastoupil do vydavatelství Economia na pozici vedoucího týmu redakčních vývojářů iHNed.cz. Od roku 2015 popularizuje Internet věcí a DIY elektroniku, přednáší o těchto oborech a školí zájemce o platformu Arduino. Za svou popularizační činnost byl v roce 2016 nominován v anketě Křišťálová Lupa, v kategorii One (wo)man show. Elektronika je jeho koníčkem už od dětství. Nejraději má staré osmibitové počítače z osmdesátých let 20. století - vlastní jich několik desítek a stále je jimi fascinován natolik, že si staví jejich repliky a programuje jejich emulátory. Ve spolupráci se sdružením CZ.NIC přednáší v kurzu „Arduino pro učitele“.

O edici Edice CZ.NIC je jedním z osvětových projektů správce české domény nejvyšší úrovně. Cílem tohoto projektu je vydávat odborné, ale i populární publikace spojené s Internetem a jeho technologiemi. Kromě tištěných verzí vychází v této edici současně i elektronická podoba knih. Ty je možné najít na stránkách knihy.nic.cz.

knihy.nic.cz

ISBN 978-80-88168-26-3

9 788088 168263 >