

Univerzální infračervený přijímač s diskretním výstupem

The universal infrared receiver with discrete output

Bc. Jakub Henzelý



ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jakub HENZELÝ**
Osobní číslo: **A09422**
Studijní program: **N 3902 Inženýrská informatika**
Studijní obor: **Počítačové a komunikační systémy**

Téma práce: **Univerzální infračervený přijímač s diskretním výstupem**

Zásady pro vypracování:

1. Nastudujte a popište používané standardy pro dálkové ovládání domácích spotřebičů pomocí infračerveného záření.
2. Navrhněte hardwarové i softwarové řešení přijímače IR signálu schopného pracovat s co nejširší množinou běžně používaných dálkových ovladačů, s možností ovládat výstupní relé pomocí uživatelsky zvoleného tlačítka ovladače.
3. Navrhněte a realizujte zapojení přijímače.
4. Vytvořte programové vybavení a ověřte jeho funkci.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **MANN, Burkhard.** C pro mikrokontroléry. Praha : BEN – technická literatura, 2004. 280 s. ISBN 80-7300-077-6.
2. **PINKER, Jiří.** Mikroprocesory a Mikropočítače. Praha : BEN technická literatura, 2004. 220 s. ISBN 80-7300-110-1.
3. **MATOUŠEK, David.** Číslicová technika. Praha: BEN technická literatura, 2002. 208 s. ISBN 80-7300-025-3.
4. **MARTÍNEK, Radislav.** Senzory v průmyslové praxi. Praha: BEN technická literatura, 2004. 200 s. ISBN 80-7300-114-4.
5. **NEVRIVA, Pavel.** Analýza signálů a soustav. Praha: BEN technická literatura, 2002. 670 s. ISBN 80-7300-004-0.
6. **CATSULIS, John.** Designing Embedded Hardware. OReilly Media, 2005. 400 s. ISBN 978-0-596-00755-3.
7. **ŠÁRA, Zdeněk.** Normy pro IR datový přenos [online]. 1999 [cit. 2011-01-26]. Dostupný z WWW: http://hw.cz/docs/ir_prenos/ir_protokoly.html
8. **8-bit Microcontrollers** [online]. Freescale Semiconductor, 2004 [cit. 2011-01-25]. Dostupný z WWW: <http://www.freescale.com/webapp/sps/site/homepage.jsp?code=8BITMCU>

Vedoucí diplomové práce:

Ing. Jan Dolinay, Ph.D.

Ústav automatizace a řídicí techniky

Datum zadání diplomové práce:

25. února 2011

Termín odevzdání diplomové práce:

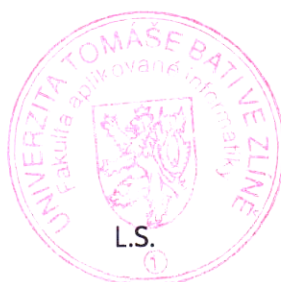
13. června 2011

Ve Zlíně dne 25. února 2011



prof. Ing. Vladimír Vašek, CSc.

děkan



prof. Ing. Karel Vlček, CSc.

ředitel ústavu

ABSTRAKT

Tato diplomová práce se zabývá softwarovým a hardwarovým návrhem a realizací zařízení, které bude sloužit jako univerzální infračervený přijímač. Pomocí tohoto zařízení bude možné pracovat s nejširší množinou běžně používaných dálkových ovladačů. V teoretické části se práce zabývá rozбором infračerveného signálu a jeho standarty a normy. V praktické části se pak zaměřuje na návrh a realizaci zařízení, popis použitého mikropočítače a vytvořením programu pro mikropočítač. Na závěr se práce zabývá nalezením optimálního nastavení zařízení pro co největší množinu dálkových ovládání.

Klíčová slova:

IR přijímač, IR přenos, infračervené záření, RC-5, mikropočítač, dálkové ovládání

ABSTRACT

The subject of this thesis is hardware and software design and implementation of a device, which will be able to function as a universal infrared receiver. Thanks to this device, it will be possible to operate with the widest set of commonly used controllers. The theoretical part of my thesis will be an analysis of infrared signal with its standards and norms. The practical part is focused on the design and production of the receiver itself, along with a description of the used microcomputer and its operational program. In the conclusion I will try to find the optimal setup for the widest possible set of remotes.

Keywords:

IR receiver, IR transmission, infrared radiation, RC-5, microcomputer, remote control

Na tomto místě bych rád poděkoval svému vedoucímu diplomové práce, panu Ing. Janu Dolinayovi, Ph.D., za vedení práce, vstřícný a profesionální přístup. Dále děkuji rodině a přátelům za podporu v průběhu studia.

Motto:

Maximální lenost – koupit si dálkový ovladač na dálkový ovladač.

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD.....	9
I TEORETICKÁ ČÁST	10
1 TEORIE IR PŘENOSU	11
1.1 HISTORIE DÁLKOVÉHO OVLÁDÁNÍ	11
1.2 HISTORIE IR PŘENOSU	13
1.3 IrDA.....	14
2 DRUHY MODULACE IR ZÁŘENÍ.....	15
2.1 PULSNÍ MODULACE.....	15
2.2 FSK MODULACE	16
2.3 BI - PHASE MODULACE	16
2.4 PULSNĚ ŠÍŘKOVÁ MODULACE	17
3 PŘENOSOVÉ PROTOKOLY	18
3.1 MÉNĚ POUŽÍVANÉ PŘENOSOVÉ PROTOKOLY	19
3.1.1 ITT protokol	19
3.1.2 JVC protokol	20
3.1.3 Nokia NRC17	21
3.1.4 Sharp protokol	22
3.1.5 Sony SIRC protokol	23
3.2 NEC PROTOKOL.....	24
3.3 RC-5 PROTOKOL	25
3.3.1 RC-6 protokol.....	27
4 JEDNOČIPOVÝ MIKROPOČÍTAČ.....	28
4.1 ARCHITEKTURA.....	29
4.2 INSTRUKČNÍ SOUBOR.....	29
4.2.1 CISC	29
4.2.2 RISC	29
4.2.3 DSP	30
II PRAKTICKÁ ČÁST	31
5 POPIS ČÁSTÍ ZAŘÍZENÍ	32
5.1 MIKROPOČÍTAČ FREESCALE S08SH16.....	32
5.1.1 Vlastnosti S08SH	33
5.1.2 Výhody S08SH.....	34
5.1.3 Sada funkcí S08SH16	34
5.1.4 Hardwarový popis S08SH16	34
5.2 IR PŘÍJÍMAČ SFH5110.....	35
5.2.1 Vlastnosti.....	35
5.2.2 Blokový diagram SFH5110-36	36

6	HARDWAROVÁ ČÁST	37
6.1	NÁVRH A VÝROBA TIŠTĚNÉHO SPOJE	37
6.1.1	Napájecí napětí a jeho stabilizace	37
6.1.2	Zapojení mikropočítače a programovacího portu	37
6.1.3	Zapojení IR přijímače.....	38
6.1.4	Vstupy a výstupy	39
6.1.5	Schéma zapojení.....	40
6.1.6	Konstrukce, problémy	41
6.2	PARAMETRY ZAŘÍZENÍ	42
7	SOFTWAREVÁ ČÁST	43
7.1	UVEDENÍ ZAŘÍZENÍ DO PROVOZU	43
7.2	ZÁKLADNÍ NASTAVENÍ.....	44
7.2.1	Nastavení vstupů a výstupů.....	44
7.2.2	Nastavení přerušení od IR přijímače	44
7.2.3	Nastavení přerušení od časovače.....	45
7.3	PROGRAM PRO PROTOKOL RC-5	48
7.3.1	Rozkódování RC-5.....	48
7.3.2	Funkce programu pro protokol RC-5	49
7.4	UNIVERZÁLNÍ PROGRAM.....	53
7.4.1	Teoretická myšlenka.....	53
7.4.2	Konstanty	53
7.4.3	Funkce univerzálního programu.....	54
7.5	TESTOVÁNÍ UNIVERZÁLNÍHO PROGRAMU	58
7.5.1	Ovladač LG – MKJ42519618	59
7.5.2	Ovladač Optibox Alligator.....	59
7.5.3	Ovladač VRC Panasonic	60
7.5.4	Ovladač HiFi Philips	60
7.5.5	Ovladač LG - 6711R1P070B	61
7.5.6	Ovladač UET MC-7	62
7.5.7	Ovladač HUMAX RM-106.....	62
7.5.8	Vyhodnocení	63
7.5.9	Automatická volba konstant.....	64
	ZÁVĚR	65
	CONCLUSION	67
	SEZNAM POUŽITÉ LITERATURY.....	69
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	71
	SEZNAM OBRÁZKŮ	72
	SEZNAM TABULEK.....	74
	SEZNAM PŘÍLOH.....	75

ÚVOD

Hlavním tématem této diplomové práce je dálkové ovládání domácích elektrospotřebičů pomocí infračerveného záření. V teoretické části práce popisuje používané standardy a normy pro infračervený přenos signálu. Život bez dálkových ovladačů si dnes ani nedokážeme představit. Tento fenomén se dostal přes veškeré domácí zařízení, počínaje ovládání televizoru, videorekordéru, DVD přehrávače, až po dnešní moderní věci, jako ovládání světel v domě, náklony satelitů apod.

Různá zařízení od různých firem komunikují různými protokoly, různými přenosy. Jedním z úkolů diplomové práce je vybrat ten nepoužívanější na základě průzkumů a vytvořit univerzální přijímač infračerveného signálu tak, aby byl schopen pracovat s co nejširší množinou běžně používaných dálkových ovladačů, s možností ovládat výstupní relé pomocí uživatelsky zvoleného tlačítka ovladače.

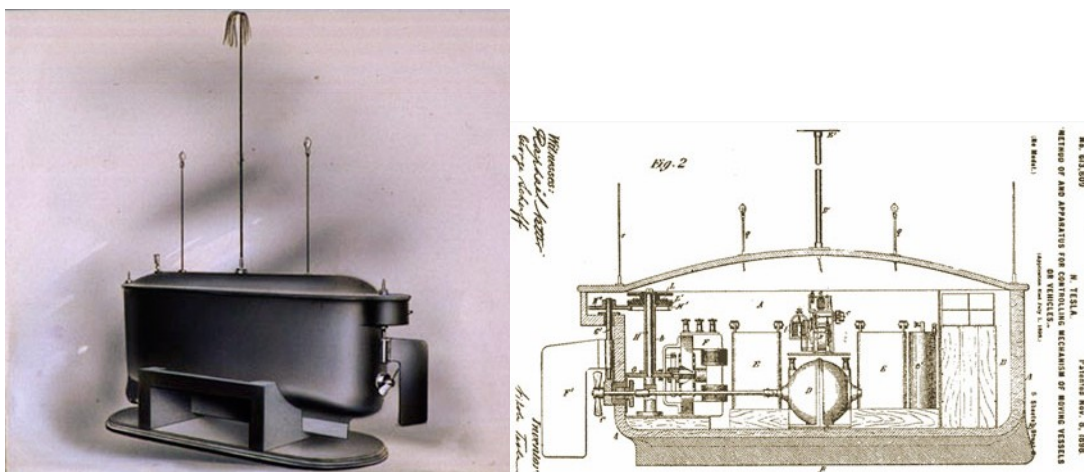
Univerzální IR přijímač bude hardwarově navrhován a sestavován. Pomocí mikropočítače, který bude součástí hardwaru, bude vytvořeno softwarové vybavení a ověřena jeho funkce. Software by měl být co nejvíce univerzální, tudíž bude testován na řadě různých ovladačů. Budou zhodnoceny výsledky testování a navrhnu to nejvhodnější nastavení.

I. TEORETICKÁ ČÁST

1 TEORIE IR PŘENOSU

1.1 HISTORIE DÁLKOVÉHO OVLÁDÁNÍ

Myšlenka dálkového ovládání libovolného přístroje se zrodila již před více než 100 lety. V roce 1893 vyvinul Nikola Tesla (1856 – 1943) první přístroj, který pojmenoval „Method of an Apparatus for Controlling Mechanism of Moving Vehicle or Vehicles“, který je popsán v patentu U. S. Patent 613809. Dnešní moderní vojenské technologie využívají směr, který Tesla téměř kompletně představil na přelomu 19. a 20. století.

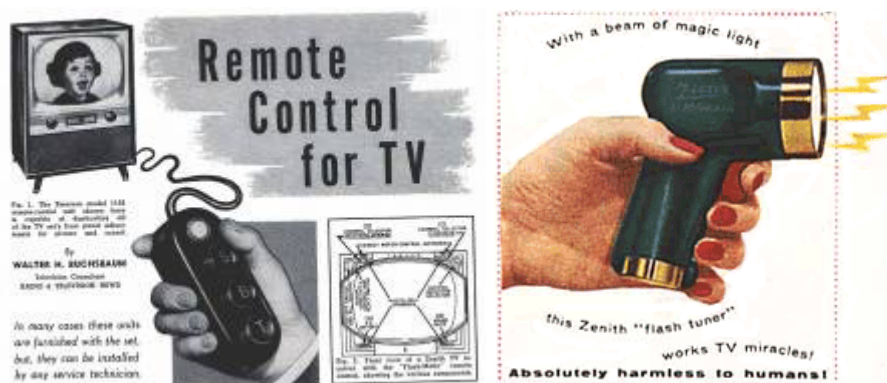


Obrázek 1 - Teslův první dálkový ovladač zvaný „Teleautomatons“

První dálkově ovládaný model letadla vzlétl v roce 1932. Technologie dálkového ovladače pro vojenské účely pak byla intenzivněji zpracována během druhé světové války.

Prvotní myšlenka dálkového ovladače pro ovládání televizorů, jak jej známe dnes, vznikla v roce 1948 ve firmě Zenith. První dálkový ovladač zkonstruovali v roce 1950 pod názvem „Lazy Bones“ čili „Líné kosti“. K televizoru byl ovladač připojen pomocí kabelu a nabízel pouze základní funkce jako přepínání kanálu a ovládání zvuku. (Obrázek 2)

V roce 1955 byl představen první bezdrátový dálkový ovladač zvaný „Flashmatic“. Ovladač pracoval na principu přenosu světla a jeho vynálezcem byl inženýr Eugene Polley. Bohužel přijímač v televizoru nedokázal rozlišit správné světlo přicházející od ovladače a jiných zdrojů, proto se stávalo, že se televize samy vypínaly nebo přepínaly programy.



Obrázek 2 - Ovladač Lazy Bones (vlevo) a Flashmatic (vpravo)

Dalším pokrokem byl ovladač „Zenith Space Command“ vyvinutý v roce 1956 Robertem Adlerem. Jednalo se o mechanické zařízení využívající ultrazvuk pro změnu programu a hlasitosti. Když uživatel stiskl tlačítko na dálkovém ovladači, udeřil tak do hliníkové tyčinky a ta vydala zvuk. Každé tlačítko mělo svoji tyčinku, která byla jinak dlouhá a podle frekvence zvuku vydaného tyčinkou televize rozeznala, o které tlačítko se jedná. Tato frekvence byla nad horním prahem lidského slyšení. Problémy byly v přijímači, který mohl přijmout i náhodný šum a také někteří lidé, obzvláště mladé ženy, mohli slyšet ostré ultrazvukové signály.



Obrázek 3 - Zenith Space Command

V 60. letech po objevu tranzistorů byl zvuk vytvářen elektronicky a ovladače na tomto principu se používaly až do začátku 80. let.

Do konce 70. let se používaly ovladače pouze se základními funkcemi: předchozí stanice, následující stanice, zvýšení a snížení hlasitosti. To se ovšem změnilo vývojem teletextové služby zvané „Cfax“. Čtyři základní funkce nestačily, protože se vyhledávání stránek skládalo ze tří číslic. Aby mohl uživatel načíst požadovanou stranu, potřeboval mít na ovladači tlačítka pro každé číslo, od nuly až po devítku. Také byly přidány pro přepínání mezi teletextem a normálním obrazem. Tohle by bylo ultrazvukově velmi náročné, proto se začaly opět používat drátové ovladače. Brzy se ale ukázalo, že je potřeba bezdrátového zařízení.

Kolem let 1977-78 se ukázaly první prototypy dálkových ovladačů pracujícím na principu infračerveného záření. ITT byla jednou ze společností, které se podílely na vývoji, a později dala jméno ITT protokolu infračervené komunikace.

Od roku 2000 množství elektronických přístrojů ve většině domácností prudce stoupl. Podle asociace spotřební elektroniky, má každý průměrný Američan doma čtyři ovladače. Většinou však více: jeden pro kabelovou televizi nebo satelit, videorekordér nebo HDD rekordér, DVD přehrávač, televize atd. V dnešní době je odhadem pouze v německých domácnostech kolem 120 miliónů ovladačů a nejvíce používaný přenos je pomocí infračerveného záření. [9]

1.2 HISTORIE IR PŘENOSU

Infračervené záření je elektromagnetické záření s vlnovou délkou větší než viditelné světlo, to znamená mezi 760nm do 1000nm. Využití tohoto záření pro přenos signálu je velmi výhodné, protože se během let velmi rozšířilo a jeho cena je nízká. Pro využití infračerveného světla pro přenos signálu muselo být navrženo několik způsobů, jak signál modulovat, abychom vyloučili nežádoucí chyby a dosáhli tak spolehlivosti přenosu.

V roce 1993 byl představen standart nazvaný IrDA – Infrared Data Association. Podílely se na něm firmy HP, IBM, Sharp a další. Hlavními výhodami IrDA jsou snadný přenos mezi různými aplikacemi i různým hardwarem, jednoduchá a poměrně levná implementace, nízký příkon, tudíž se dá s výhodou využít v mobilních zařízeních.

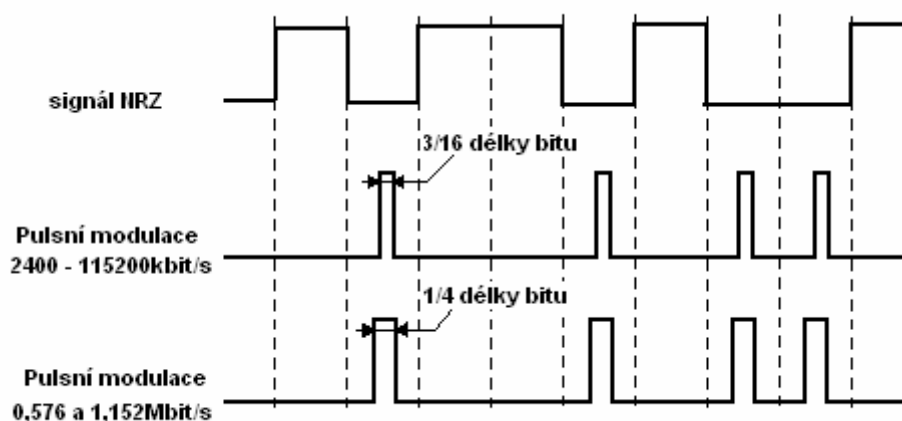
1.3 IrDA

IrDA zařízení vysílá a přijímá modulované infračervené záření o vlnové délce 875nm. K přenosu využívá infračervené LED diody. IrDA je dnes součástí notebooků, PDA, mobilních telefonů, i když je vytlačováno přenosem Bluetooth a to kvůli nevýhodě infračerveného přenosu – přímé viditelnosti.

První standart označený IrDA 1.0, popisuje přenos dat rychlostmi od 2400bit/s do 115200bit/s (2400bit/s, 9600bit/s, 19,2kbit/s, 38,4kbit/s, 57,6kbit/s, 115,2kbit/s). Prvotní komunikace probíhá vždy rychlostí 9600bit/s, a proto je podpora této rychlosti povinná. Používá se pulsní modulace a doba trvání pulsu je 3/16 doby trvání bitu. Metoda se občas značí jako SIR (Serial Infrared).

Druhým standardem je IrDA 1.1, který navíc definuje rychlosti 0,576Mbit/s a 1,152Mbit/s s pulsní modulací o době trvání pulsu 1/4 doby trvání bitu. Metoda se občas značí jako MIR (Medium Infrared).

Obě verze pracují do vzdálenosti 1 metr. Formát dat je jako při běžném sériovém přenosu se start bitem. Pulsní modulace se používá, aby infračervené LED diody mohly využívat maximální výkon, kterým by ovšem nemohly svítit trvale, proto pulsy.



Obrázek 4 - Doby trvání pulsů

2 DRUHY MODULACE IR ZÁŘENÍ

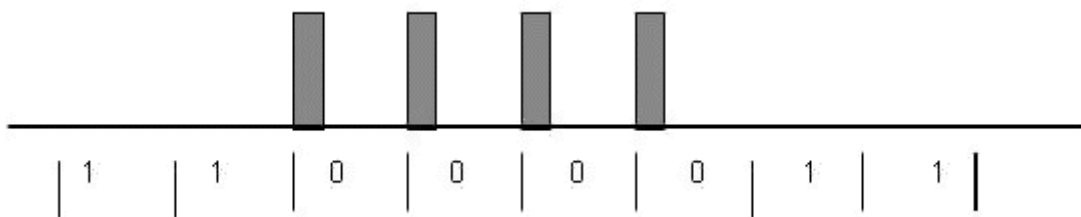
Základním požadavkem na kódování přenášených dat je zajištění dostatečné bezpečnosti přenosu. Data mohou být vysílána mnoha způsoby. V současné době se využívá výhradně digitální přenos, délka přenášených rámců se liší v závislosti na použité aplikaci. Některé modulace byly přijaty jako standart. Mezi nejčastěji používané modulace IR záření patří:

- Pulsní modulace
- FSK modulace
- Bi – phase modulace
- Pulsně šířková modulace

2.1 PULSNÍ MODULACE

Pulsní modulace je jedním z nejjednodušších způsobů kódování dat přenášených infračerveným zářením. Princip této modulace spočívá v rozdělení signálu na stejně velká tzv. časová okna. V tomto okně se buď vyskytne, nebo nevyskytne impuls konstantní délky. Tento impuls je podstatně kratší, než je délka časového okna. Logickou nulou poté rozumíme stav, kdy se impuls v okně vyskytne. Pokud impuls v okně není, uvažujeme logickou jedničku.

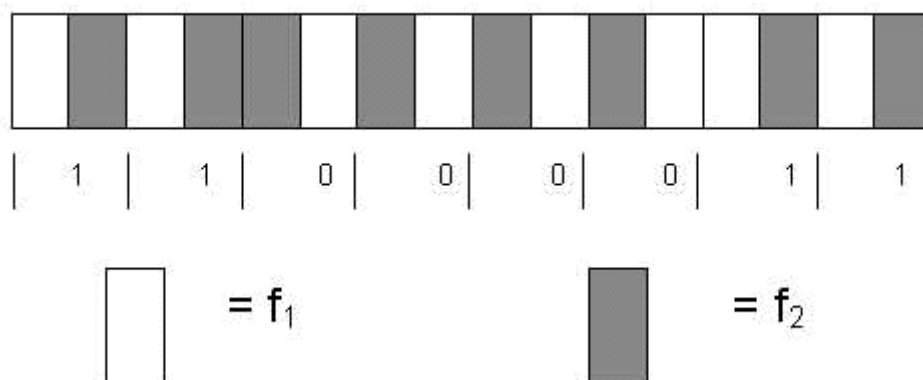
Důležitá je synchronizace hodinových impulsů s hranou přijímaného signálu. Při přenosu většího bloku jedničkových bitů, kdy nejsou vysílány žádné impulsy, by mohl přijímač vypadnout ze synchronismu. Této chybě předcházíme přidáním tzv. bit-stuffingu, kdy po určitém počtu jedničkových bitů je vyslán navíc jeden nulový. Tento bit je pak třeba na přijímací straně odstranit.



Obrázek 5 - Princip pulsní modulace

2.2 FSK MODULACE

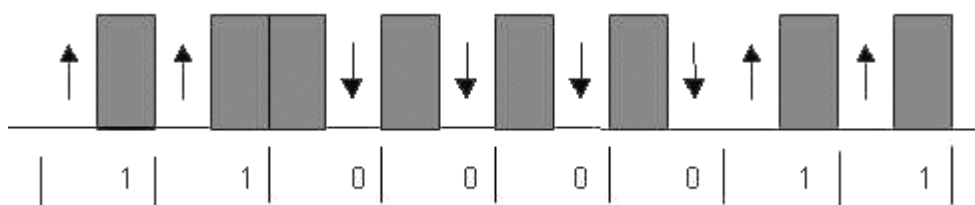
Zkratka FSK značí „frequency shift keying“. Jedná se o velice bezpečnou metodu přenosu. Princip spočívá v modulování bitů pomocí dvou frekvencí, jak je znázorněno na obrázku 2-2. Nevýhodou je vyšší cena zařízení, větší energetická náročnost a pomalejší datový přenos. Z těchto důvodů se FSK modulace používá jen výjimečně a to v systémech s požadavky na vysokou bezpečnost.



Obrázek 6 - Princip FSK modulace

2.3 BI - PHASE MODULACE

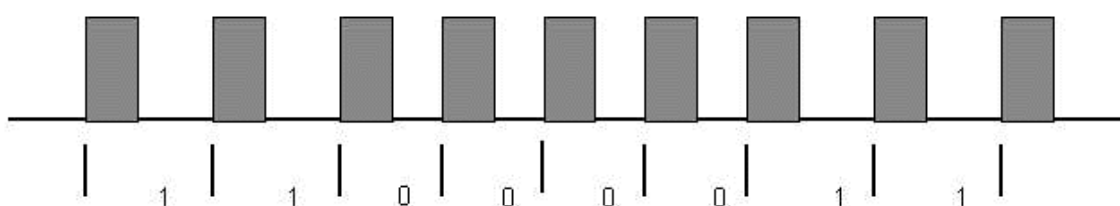
Princip bi-phase modulace spočívá, stejně jako u pulsní modulace, v rozdělení signálu na časová okna. Tato časová okna mají stejnou velikost a pro detekci bitu se bere změna úrovně signálu uvnitř tohoto okna. Pokud je změna kladná, tj. z log. 0 na log. 1, je bit brán jako jednička, pokud je změna záporná, uvažujeme nulu. Toto je nejčastěji používaný typ modulace. Využívá jej například kód RC-5, který je hojně používaný u spotřební elektroniky evropské produkce.



Obrázek 7 - Princip bi – phase modulace

2.4 PULSNĚ ŠÍŘKOVÁ MODULACE

Principem pulsně šířkové modulace, neboli PWM „Pulse Width Modulation“, je opět rozdělení signálu do tzv. časových oken. Tato okna ovšem nejsou u pulsně šířkové modulace stejně dlouhá, tak jak to bylo v předešlých případech. Impuls se v jednotlivých časových oknech nachází vždy, ale podle délky časového okna určujeme jeho hodnotu. Delšímu časovému oknu odpovídá jednička a kratšímu nula. Je ovšem opět důležité brát zřetel na synchronizaci, proto jsou jednotlivá časová okna synchronizována náběžnou hranou.



Obrázek 8 - Princip pulsně šířkové modulace

3 PŘENOSOVÉ PROTOKOLY

Přenosový protokol je určitý předpis, podle kterého se uskutečňuje spojení, spolupráce vysílače a přijímače a zabezpečení dat. Přenosových protokolů je celá řada. Někteří výrobci spotřební elektroniky a počítačů si vytvořili vlastní protokoly, některé jsou zanedbatelné a některé se začaly hojně využívat. Použití přenosových protokolů se také rozděluje podle lokality. Zmíníme si všechny důležité:

- ITT protokol (z historického hlediska)
- JVC protokol
- Nokia NRC17
- Sharp protokol
- Sony SIRC
- NEC protokol
- Philips RC-5

Ovšem dopodrobna se budeme zabývat těmi nejpoužívanějšími, a to jsou NEC a RC-5. Přenosové protokoly předepisují ve většině případů přenos informace pomocí datových rámců. U každého protokolu je tento rámec složen z různého počtu bitů a různých délek. V rámci můžeme najít také rozdílné bity.

- Start-bit (může značit začátek datového rámce)
- Toggle-bit (zabezpečení přenosu, dodržení synchronizace)
- Adresové bity (určují, jaká spotřební elektronika bude ovládána – TV, DVD,...)
- Datové bity (někdy uváděny jako příkazové bity, určují, jaký příkaz provádíme – zesílení hlasitosti, přepnutí programu,...)

Jednotlivých bitů může být mnohem více. Výrobci spotřební elektroniky a také amatérští modeláři volí protokol nejen podle ceny součástek, ale i podle náročnosti či složitosti protokolu.

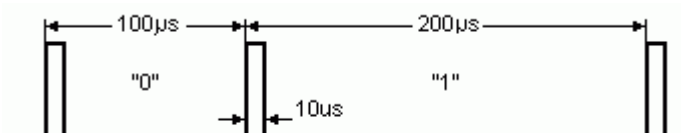
3.1 MÉNĚ POUŽÍVANÉ PŘENOSOVÉ PROTOKOLY

3.1.1 ITT protokol

ITT protokol patří mezi nejstarší protokoly pro přenos infračerveného záření vůbec. Od ostatních protokolů se liší tím, že nepoužívá modulovanou nosnou frekvenci. Jedním příkazem je přenášeno 14 krátkých pulsů s délkou pulsu $10\mu\text{s}$. Kódování probíhá tak, že se mění vzdálenost mezi pulsy. Pro přenos pomocí ITT protokolu je potřeba velmi malá energie, tudíž zajišťuje dlouhou životnost baterie. Velkou nevýhodou tohoto protokolu je občasné vyvolávání falešných příkazů. Mnoho výrobců spotřební elektroniky využívalo ITT protokol, např. ITT, Greatz, Luxor ale třeba i známý výrobce mobilních telefonů Nokia.

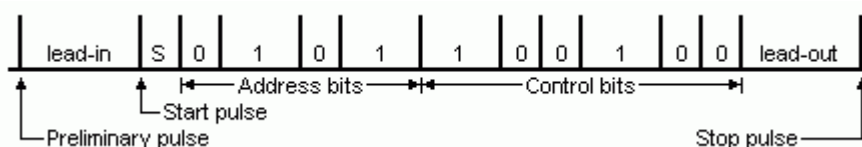
Vlastnosti ITT

- Přenášeno 14 krátkých pulsů s délkou pulsu $10\mu\text{s}$
- Kódování probíhá změnou vzdálenosti mezi pulsy
- Zaručuje dlouhou výdrž baterie
- Délka adresy je 4 bity, délka dat je 6 bitů
- Mezera $100\mu\text{s}$ značí logickou nulu a mezera $200\mu\text{s}$ značí logickou jedničku



Obrázek 9 - ITT protokol – rozpoznávání logické úrovně

- Na začátku a na konci zprávy je tzv. „lead-in“ a „lead-out“, který nám říká, že je začátek, respektive konec zprávy. Jeho délka je $300\mu\text{s}$.



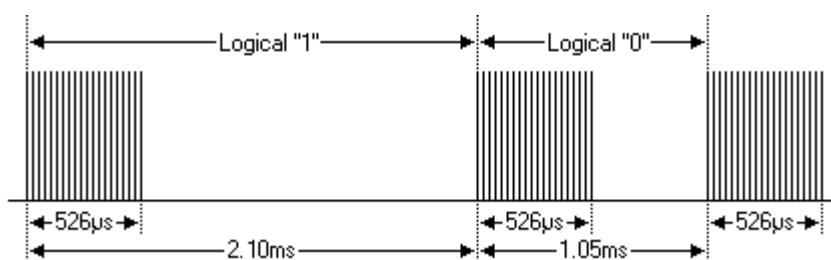
Obrázek 10 - ITT protokol – ukázka signálu

3.1.2 JVC protokol

Značka JVC má také svůj vlastní protokol pro komunikaci infračerveným zářením, ale ve svých výrobcích používají mnohdy různé protokoly. Pro modulaci používá délky časových oken. Ve kterých se vždy vyskytují pulsy o konstantní délce $526\mu\text{s}$.

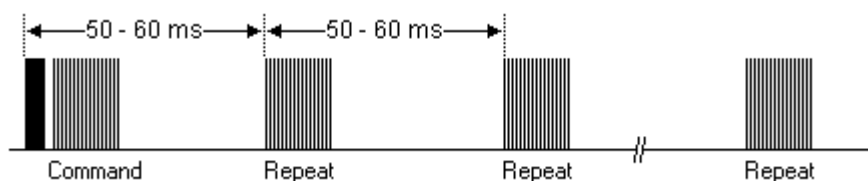
Vlastnosti protokolu JVC

- Délka adresy je 8 bitů a data zabírají také 8 bitů
- Používá pulsně poziční modulaci
- Délka pulsu je $526\mu\text{s}$
- Nosná frekvence je 38kHz
- Délka bitu je 1,05ms, to značí logickou 0 a logická jednička je 2,10ms



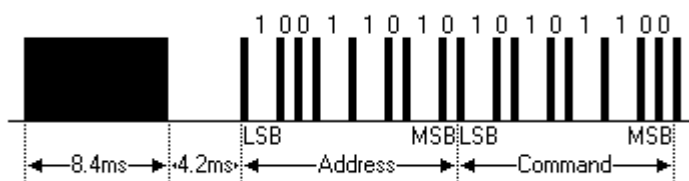
Obrázek 11 - JVC protokol – rozpoznávání logické úrovně

- Data jsou přenášena po 50 – 60ms blocích tak dlouho, dokud je stisknuto tlačítko na dálkovém ovladači.



Obrázek 12 - JVC protokol – ukázka stále stisknutého tlačítka

- Adresovým a datovým bitům předchází 8,4ms dlouhý pre-puls, a poté 4,2ms dlouhá pauza.



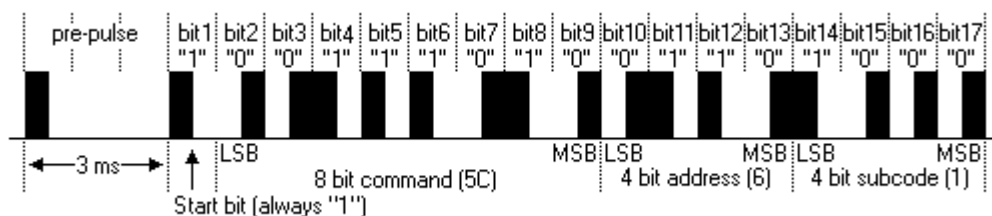
Obrázek 13 - JVC protokol – ukázka signálu

3.1.3 Nokia NRC17

Název protokolu je složen z počátečních písmen „Nokia Remote Control“, a protože používá 17 bitů pro přenos, zní název NRC17. Tento protokol nepoužívala jen Nokia, ale i další firmy zabývající se spotřební elektronikou jako Finlux nebo Salora. V současné době se tento protokol používá především v satelitní technice – přijímače a set-top boxy Nokia.

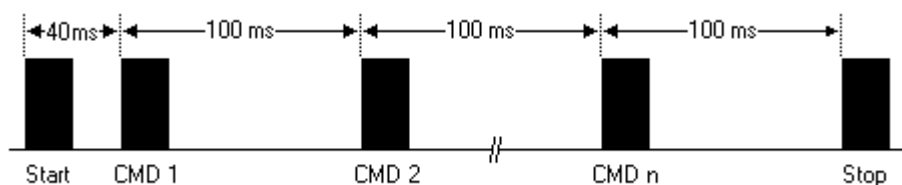
Vlastnosti NRC17

- 4 adresné bity, 8 datových bitů a 4 bity pro rozšíření adresných bitů
- Používá bi – fázovou modulaci
- Nosná frekvence je 38kHz
- Délka jednoho bitu je 1ms
- Poměr impuls / mezera je $\frac{1}{4}$, čímž snižujeme spotřebu a zvyšujeme výdrž baterie
- Pre-puls má 0,5ms a následuje ho 2,5ms dlouhá pauza, poté následuje start bit, který je vždy logická jednička a může sloužit jako synchronizační bit.



Obrázek 14 - Nokia NRC17 – ukázka signálu

- Zpráva se skládá ze 3ms pre-pulsu a pauzy a následuje 17 bitů po 1ms. Tudiž je délka zprávy 20ms.
- Při stisku tlačítka je vždy poslán blok a první opakování přichází po 40ms, další opakování jsou již vždy po 100ms.



Obrázek 15 - Nokia NRC17 – opakování zprávy

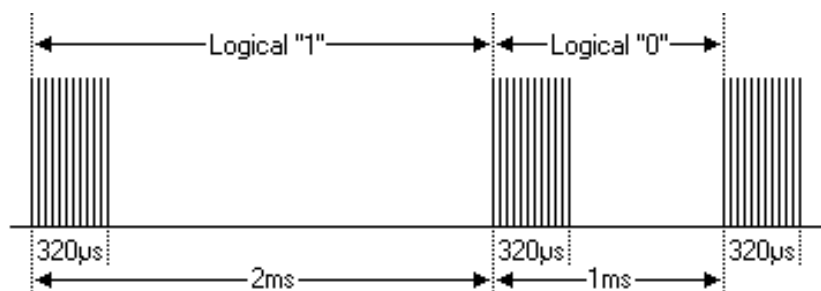
- NRC17 poskytuje možnost sdělit přijímači, že je baterie vybitá. Pokud se tak stane, pošle místo 3ms dlouhého pre-pulsu 4ms dlouhý pre-puls.

3.1.4 Sharp protokol

Jedná se o protokol, který používají videorekordéry značky Sharp.

Vlastnosti Sharp protokolu

- 8 datových bitů, 5 adresových bitů
- Používá pulsně poziční modulaci
- Nosná frekvence je 38kHz
- Délka jednoho bitu je 1ms, pokud se jedná o logickou nulu nebo 2ms pokud se jedná o logickou jedničku
- Každý puls má 320μs, a poměr impulz / mezera je 1/4 nebo 1/3



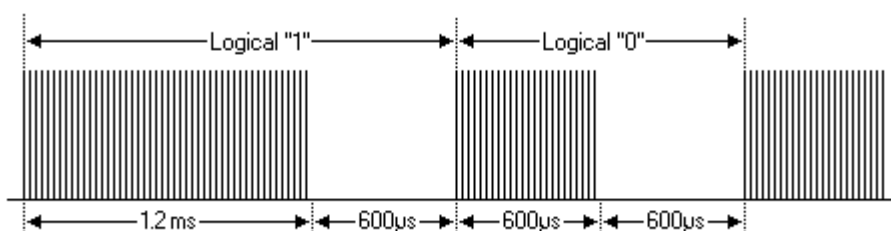
Obrázek 16 - Protokol Sharp – rozpoznávání logické úrovně

3.1.5 Sony SIRC protokol

SIRC značí „Seriál InfraRed Control“. Tohoto protokolu existují 3 verze. 12bitová, 15bitová a 20bitová. Níže popíšu 12bitovou verzi.

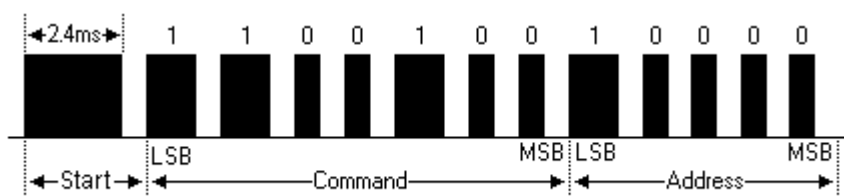
Vlastnosti Sony SIRC protokolu

- 5 adresových bitů, 7 datových bitů
- Používá pulsně šířkovou modulaci (PWM)
- Nosná frekvence je 40kHz
- Bitový čas je 1,2ms pro logickou jedničku a 0,6ms pro logickou nulu
- Mezi každým bitem je mezera 0,6ms
- Poměr impuls / mezera je $\frac{1}{4}$ nebo $\frac{1}{3}$



Obrázek 17 - Sony SIRC protokol – rozpoznávání logické úrovně

- Na začátku zprávy je vždy 2,4ms dlouhý pre-puls, po němž následuje klasická 0,6ms dlouhá mezera. Data jsou přenášena každých 45ms tak dlouho, dokud je stisknuto tlačítko ovladače.



Obrázek 18 - Sony SIRC protokol – ukázka signálu

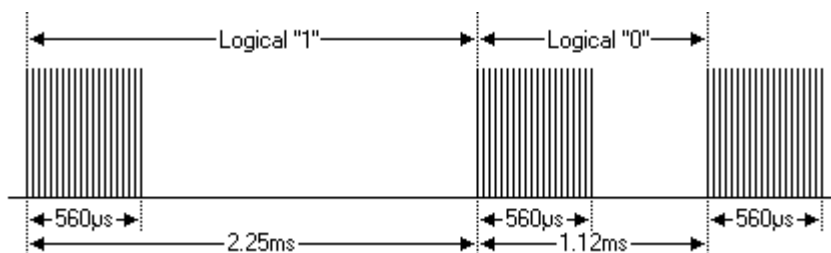
3.2 NEC PROTOKOL

Tento protokol byl vyroben firmou NEC Electronics a je využíván ve většině výrobků pocházejících od japonských firem – Pioneer, Hitachi, Teac, Yamaha. Mezi jeho výhody patří možnost kontroly dat proti chybám.

Vlastnosti NEC protokolu

- Nosná frekvence je 38kHz
- Používá pulsně poziciční modulaci
- 8 bitů je datových a 8 adresových
- Adresa a data jsou kvůli kontrole přenášeny dvakrát

Pulsy jsou u tohoto protokolu vždy stejně dlouhé a to 560 μ s. Pokud je puls dlouhý 2,25ms tak se jedná o logickou jedničku. Pokud je dlouhý 1,125ms (tedy přesně polovina), tak se jedná o logickou nulu.

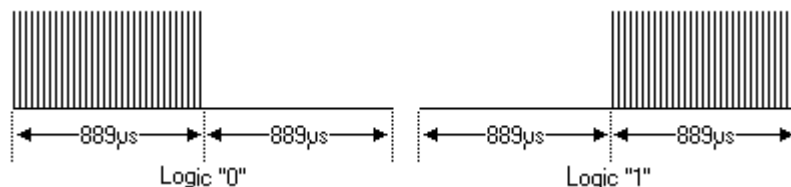


Obrázek 19 - Protokol NEC – rozpoznávání logické úrovně

V hlavičce protokolu se vždy na prvním místě nachází 9ms dlouhý pre-puls, který je následován 4,5ms dlouhou mezerou. Délka této mezery určuje typ rámce. Novější specifikace protokolu totiž umožňuje zaslání prvního celého rámce a při zjištění, že je tlačítko stále zmáčkuto, je dále vysílána pouze krátká informace o opakování rámce. Tímto snížíme spotřebu až o jednu třetinu. Pokud je tedy doba mezery dlouhá 2,5ms, jedná se o opakovací kód a kromě 560 μ s dlouhého ukončovacího pulsu se další data nevyšlou. Pokud je doba klasických 4,5ms, tak se vyšle znova celý rámec o délce 32 bitů.

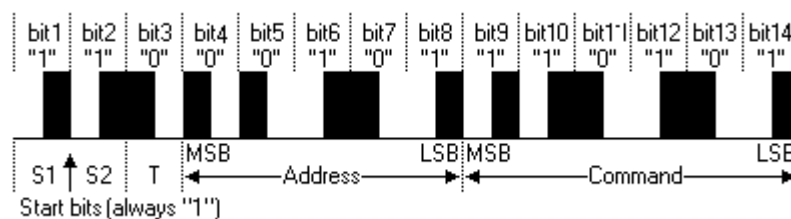
Toggle-bit nemá stejnou logickou úroveň. Jeho úroveň se mění na opačnou při každém novém stisku tlačítka. Díky tomu můžeme na přijímací straně rozlišit nový stisk tlačítka, nebo tlačítko trvale stisknuté.

Protokol RC-5 používá bi-fázovou modulaci a pracuje na frekvenci 36kHz.



Obrázek 23 - RC-5 protokol – rozlišení logické úrovně

Bitový interval má konstantní délku rovnou 64 periodám nosné frekvence, tudíž lze spočítat $64/36\text{kHz} = 1778\mu\text{s}$. Jestliže je datový rámeček dlouhý 14 bitů, pak jednoduše spočítáme dobu přenosu a to $14 * 1778\mu\text{s} = 25\text{ms}$. Doba značky či mezery je složena z 32 period nosné, tudíž 889µs.



Obrázek 24 - RC-5 protokol – datový rámeček

Philips vytvořil seznam standardizovaných adres a příkazů. Tím zajistíme vzájemnou kompatibilitu mezi jinými výrobci, kteří používají protokol RC-5. Philips myslel i na možnost, kdy se na sebe položí dvě zařízení používající protokol RC-5. Tím by došlo k nežádoucímu přepínání obou zařízení zároveň. Proto je většina z často používaných elektronických zařízení v seznamu adres uvedena dvakrát.

RC-5 Adresa	Zařízení
\$ 00 - 0	TV1
\$ 01 - 1	TV2
\$ 02 - 2	Teletext
\$ 03 - 3	Video
\$ 04 - 4	LV1
\$ 05 - 5	VCR1
\$ 06 - 6	VCR2
\$ 07 - 7	Experimentální
\$ 08 - 8	Sat1
\$ 09 - 9	Kamera
\$ 0A - 10	SAT2
\$ 0B - 11	
\$ 0C - 12	CDV
\$ 0D - 13	Videokamera
\$ 0E - 14	
\$ 0F - 15	
\$ 10 - 16	Pre-amp
\$ 11 - 17	Ladička
\$ 12 - 18	Rekordér1
\$ 13 - 19	Pre-amp
\$ 14 - 20	CD přehrávač
\$ 15 - 21	Phono
\$ 16 - 22	Sata
\$ 17 - 23	Recorder2
\$ 18 - 24	
\$ 19 - 25	
\$ 1A - 26	CDR
\$ 1B - 27	
\$ 1C - 28	
\$ 1D - 29	Osvětlení
\$ 1E - 30	Osvětlení
\$ 1F - 31	Telefon

RC-5 Příkaz	TV zabezpečení	Příkaz VCR
\$ 00 - 0	0	0
\$ 01 - 1	1	1
\$ 02 - 2	2	2
\$ 03 - 3	3	3
\$ 04 - 4	4	4
\$ 05 - 5	5	5
\$ 06 - 6	6	6
\$ 07 - 7	7	7
\$ 08 - 8	8	8
\$ 09 - 9	9	9
\$ 0A - 10	-/--	-/--
\$ 0C - 12	Pohotovostní	Pohotovostní
\$ 0D - 13	Němý	
\$ 10 - 16	Volume +	
\$ 11 - 17	Hlasitost -	
\$ 12 - 18	Jasu +	
\$ 13 - 19	Jas -	
\$ 20 - 32	Program +	Program +
\$ 21 až 33	Program -	Program -
\$ 32 - 50		Vzad
\$ 34 až 52		Rychlý posun vpřed
\$ 35 - 53		Hrát
\$ 36 - 54		Stop
\$ 37 - 55		Záznam

Tabulka 1 – Adresy a příkazy RC-5 protokolu

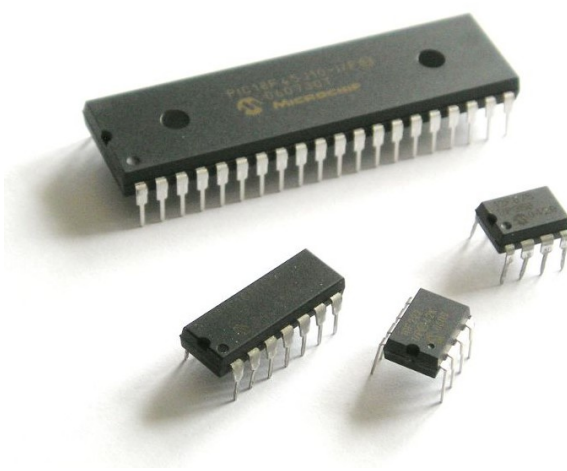
3.3.1 RC-6 protokol

Tento protokol byl navržený firmou Philips jako nástupce RC-5. Zatím se příliš nerozšířil. Jedná se o velmi univerzální protokol, tudíž jeho realizace a dokumentace je velmi obtížná. Také používá bi-fázovou modulaci a jeho nosná frekvence je 36kHz.

4 JEDNOČIPOVÝ MIKROPOČÍTAČ

Jedná se o monolitické integrované obvody, které obsahují všechny prvky mikropočítače (procesor, vstupy, výstupy, operační paměť, časovač, řadič, čítač,...). Bývají používány jednoúčelově, ovšem většina z nich je schopna zvládat mnoho operací. Jejich použití je spolehlivé a kompaktní. Historie sahá až do 60. let 20. století. Tehdy se ovšem nejednalo o tak miniaturní obvody, jak je známe dnes.

Jednočipové mikropočítače mají tak jako stolní počítače různé druhy architektur a instrukčních sad.



Obrázek 25 - Jednočipové počítače

Mezi výrobce mikropočítačů patří firma Freescale Semiconductor, Inc. Tento americký výrobce polovodičových součástek vznikl oddělením polovodičové divize od společnosti Motorola v roce 2004. Hlavní činností společnosti je výroba součástek pro automobilový a telekomunikační průmysl, mikrokontrolérů a mikromechanických senzorů. Freescale Semiconductor je jednou z dvaceti největších firem v polovodičovém průmyslu na světě.

Další výrobci mikropočítačů:

- NEC Electronics
- NXP Semiconductors
- Micron Technology
- Infineon

- STMicroelectronics
- Texas Instruments

4.1 ARCHITEKTURA

Existují dvě základní architektury mikroprocesorů – von Neumannova a Harvardská, každá má svoje výhody i nevýhody. Při současném stupni integrace se zřejmě častěji využívá Harvardská architektura, vysoký stupeň integrace dovoluje připojit různé bloky paměti pomocí vlastních sběrnic. Dělení na „Harvardskou“ a „Von Neumannovu“ architekturu je však při dnešním stupni integrace již poněkud akademické. U moderních architektur se často uživateli adresový prostor jeví navenek jako lineární (Von Neumannovský), zatímco fyzicky jsou paměti k jádru připojeny pomocí několika nezávislých sběrnic (např. jedna sběrnice pro FLASH/ROM (paměť programu), druhá pro uživatelskou vnitřní RAM a zásobník, třetí pro připojení integrovaných paměťově mapovaných periférií, další pro připojení externí RAM). [11]

4.2 INSTRUKČNÍ SOUBOR

Další rozdělení mikrokontrolérů je podle použitého instrukčního souboru. V oblasti jednočipových počítačů se běžně používají instrukční soubory typu CISC, RISC i DSP.

4.2.1 CISC

CISC označuje procesor se „složitým instrukčním souborem“. Procesor podporuje mnoho formátů a druhů instrukcí. Na jednu stranu to znamená úsporu místa v programové paměti (vyšší hustotu kódu), na druhé straně to však znamená komplikovanější dekodér instrukcí ve vlastním mikrokontroléru a pomalejší zpracování instrukcí.

4.2.2 RISC

RISC označuje procesor s redukováným instrukčním souborem. Základní myšlenkou je omezení počtu a zjednodušení kódování instrukcí, což vede ke zjednodušení instrukčního dekodéru. Hlavní výhodou tohoto přístupu je rychlost a jednoduchost, na stejné ploše čipu může být místo 16bitového procesoru CISC 32bitový procesor RISC.

Nevýhodou je, že pro zakódování instrukce je potřeba více místa, někdy musíme použít dvě instrukce místo jedné, takže klesá hustota kódu.

4.2.3 DSP

Digitální signálový procesor nebo také digitální signální procesor (zkratka DSP) je mikroprocesor, jehož návrh je optimalizován pro algoritmy používané při zpracování digitálně reprezentovaných signálů. Hlavním nárokem na systém bývá průběžné zpracování velkého množství dat „protékajících“ procesorem. [12]

II. PRAKTICKÁ ČÁST

5 POPIS ČÁSTÍ ZAŘÍZENÍ

Zde budou popsány nejdůležitější části, ze kterých se zařízení skládá, včetně popisu vytvoření zařízení, schémat apod.

5.1 MIKROPOČÍTAČ FREESCALE S08SH16

Firma Freescale Semiconductors má ve svém portfoliu celou řadu mikropočítačů. Kromě 8bitových zde najdete také 16bitové, 32bitové digitální signálové procesory atd. My se ale budeme zabývat těmi prvními, tedy 8bitovými. K těmto mikropočítačům nabízí Freescale celou řadu podpůrných nástrojů - hardwarové a softwarové nástroje, demo desky, ladící nástroje a také integrované vývojové prostředí (IDE) CodeWarrior.

Samozřejmě volba toho správného mikropočítače může probíhat podle mnoha kritérií - cena, výkon, velikost operační paměti, napájení atd. Pro tuto diplomovou práci byl vybrán takový, který splňoval účel, pro který chceme mikropočítač používat. Proto byl vybírán velmi dostupný a paměťově dostačující mikropočítač S08SH16.

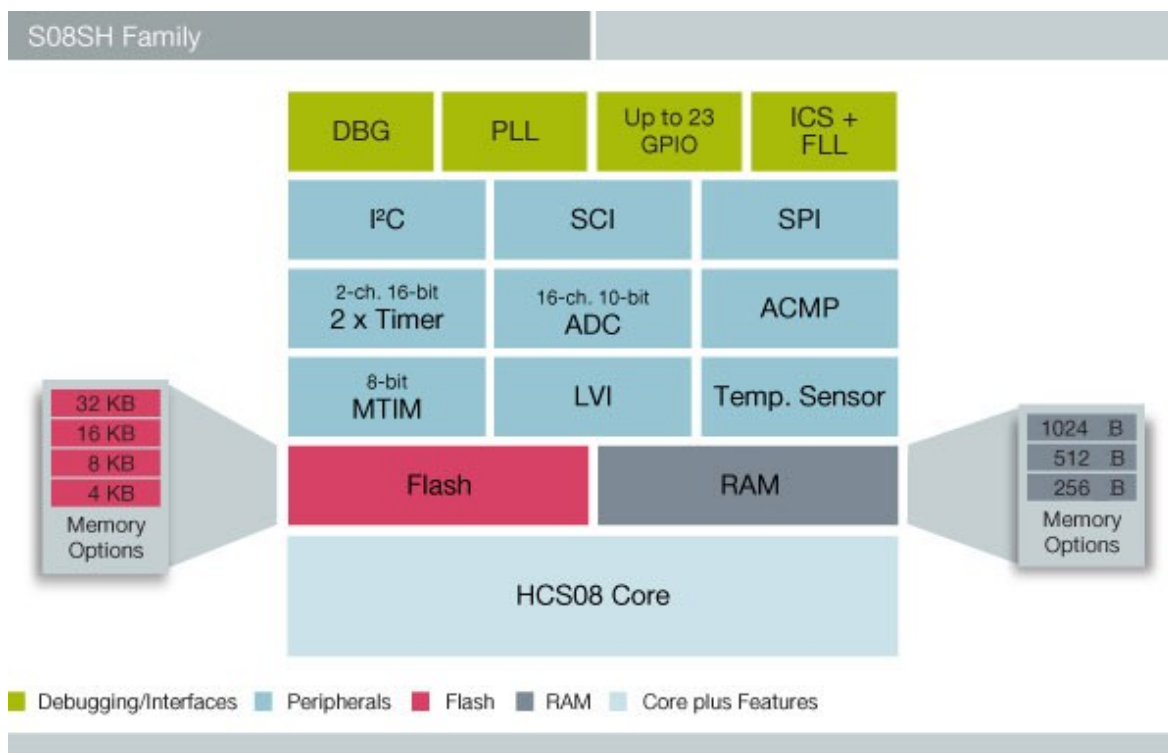
Série mikropočítačů S08SH je zaměřena na aplikace v rozsahu do 5V. Dělí se podle velikosti vnitřní paměti od 4kB do 32kB a také podle velikosti paměti RAM od 256B do 1024B. Každý z nich je také k dostání v různých pouzdrech (podle počtu vývodů). Záleží na použití – kolik je potřeba časovačů, vstupů, výstupů, apod.



Obrázek 26 - Mikropočítač S08SH16 (SOIC 28)

5.1.1 Vlastnosti S08SH

- Frekvence jádra je 40MHz
- Frekvence sběrnice je 20MHz
- Více možností pro zdroje hodinového signálu
 - 20MHz vnitřní zdroj hodinového signálu
 - XOSC pro externí zdroj hodinového signálu
- 2 x 2 kanálový 16bitový časovač (TPM) a 8bitový modulo časovač (MTIM)
- 16kanálový 10bitový analogově-digitální převodník (ADC), vestavěný teplotní senzor
- Analogový komparátor ACMP
- Detekce nízkého napětí
- Watchdog
- Univerzální Input/Output (GPIO)



Obrázek 27 - Popis částí mikropočítačů S08SH

5.1.2 Výhody S08SH

- Velké možnosti analogového převodníku (16ch, 10bit ADC)
- Široká nabídka periférií (SCI, SPI, I2C, ADC, analogový komparátor, teplotní čidlo)
- Velká FLASH paměť

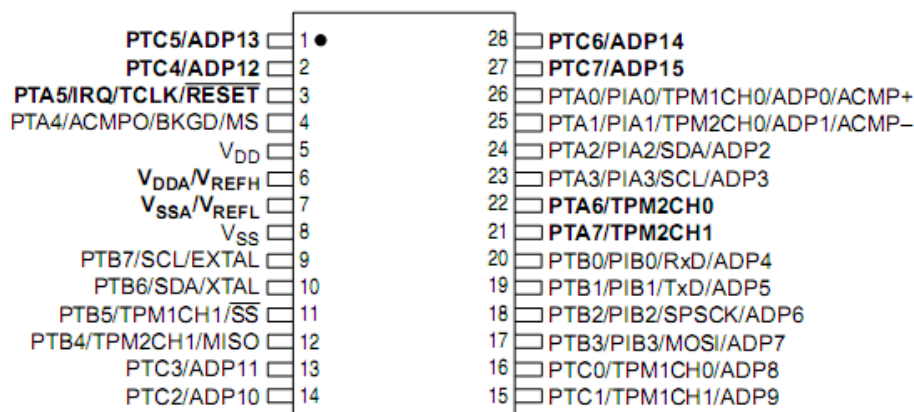
5.1.3 Sada funkcí S08SH16

FLASH size (bytes)	16 384		
RAM size (bytes)	1024		
Pin quantity	28	20	16
ACMP	Ano		
ADC channels	16	12	8
DBG	Ano		
ICS	Ano		
IIC	Ano		
IRQ	Ano		
MTIM	Ano		
Pin Interrupts	8		
Pin I/O 1	23	17	13
RTC	Ano		
SCI	Ano		
SPI	Ano		
TPM1 channels	2		
TPM2 channels	2		
XOSC	Ano		

Tabulka 2 – Sada funkcí S08SH16

5.1.4 Hardwarový popis S08SH16

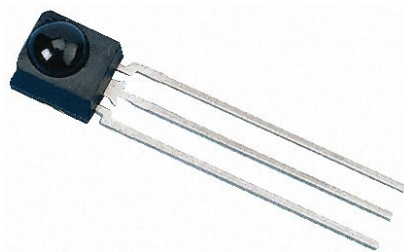
Pro IR zařízení byl použit 28vývodový S08SH16. Popis jeho vstupů a výstupů vidíte na obrázku níže. Kromě klasického napájení a země (vývody 5,6 a 7,8), je také důležitý programovací port. Ten se skládá z vývodu 3 (RESET) a také vývodu 4 (BKGD).



Obrázek 28 - Pouzdro S08SH16

5.2 IR PŘIJÍMAČ SFH5110

SFH5110 je infračervený přijímač, který detekuje IR záření od infračerveného dálkového ovládání. Zařízení obsahuje fotodiodu, předzesilovač, automatické řízení zisku, pásmové propusti a demodulátor. Vyrábí se ve více provedeních, odlišných nosnou frekvencí signálu a to od 30 do 40kHz. To udává poslední dvojčíslí v názvu SFH5110-XX.

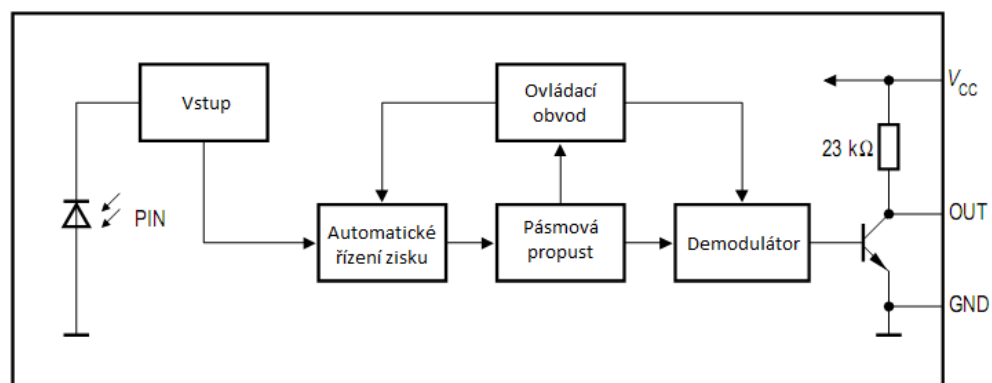


Obrázek 29 - IR přijímač SFH5110

5.2.1 Vlastnosti

- Jedná se o monolitické zařízení (všechny části v jednom čipu)
- Vhodný pro použití při vlnové délce 940nm
- Vysoce citlivý
- Různé frekvence pásmové propusti
- Kompatibilní CMOS i TTL
- Nejsou potřeba žádné externí komponenty

5.2.2 Blokový diagram SFH5110-36



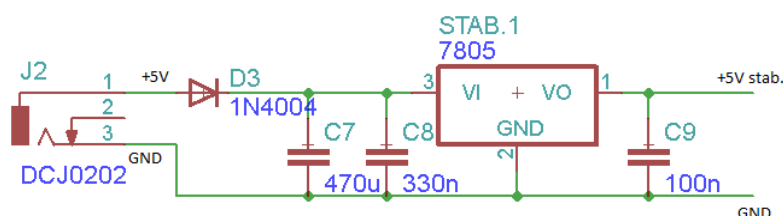
Obrázek 30 - Blokový diagram SFH5110-36

6 HARDWAROVÁ ČÁST

6.1 NÁVRH A VÝROBA TIŠTĚNÉHO SPOJE

6.1.1 Napájecí napětí a jeho stabilizace

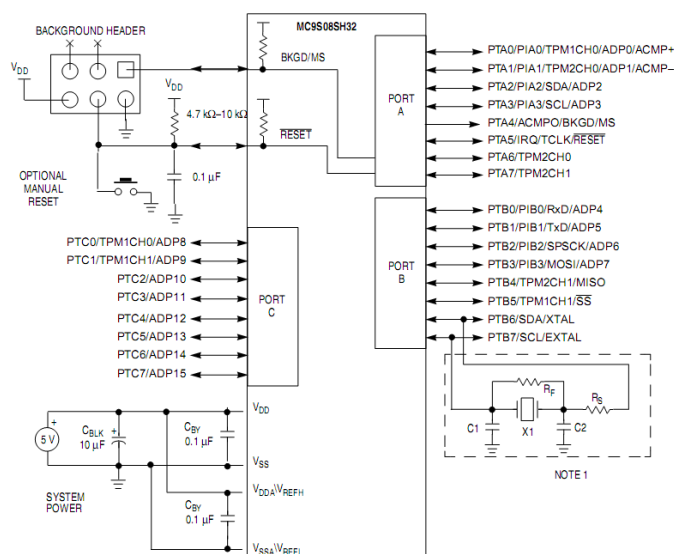
Před začátkem návrhu je potřeba rozvrhnout vstupy, výstupy apod. Mikropočítač S08SH16 je napájen 5V. Stejné napětí je potřebné pro relé a IR přijímač. Zdroj napětí 5 - 12V (u zařízení je použit zdroj s napětím 9V) je přiveden do zařízení přes klasický napájecí konektor. Aby bylo napětí pro mikropočítač +5V, je potřeba napájení stabilizovat. K tomu je použit stabilizátor 7805, který spolu s kondenzátory a ochrannou diodou proti přepólování tvoří stabilní výstup +5V a GND.



Obrázek 31 - Stabilizace napětí

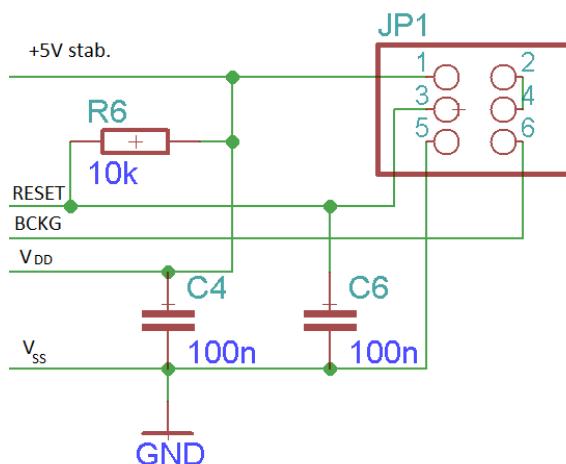
6.1.2 Zapojení mikropočítače a programovacího portu

U zapojení mikropočítače bylo dodržováno optimální zapojení dle manuálu výrobce.



Obrázek 32 - Zapojení mikropočítače

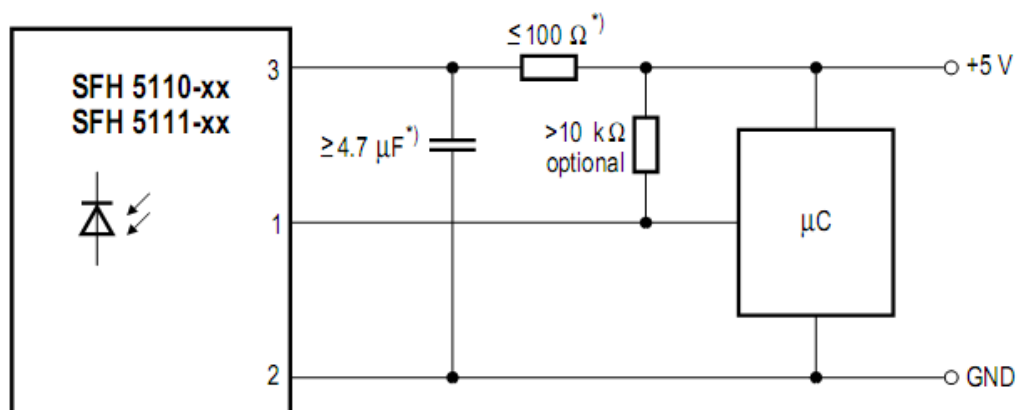
Do napájení mikropočítače je přivedeno stabilizované napájení +5V. Mezi V_{DD} (napájení mikropočítače) a V_{SS} (zem mikropočítače) byl vložen kondenzátor $0,1\mu\text{F}$ dle manuálu. Důležité je ovšem také připojení programovacího konektoru. Ten má kromě přivedených +5V a GND také ochranný kondenzátor a odpor. Mezi signál RESET a GND je zapojen kondenzátor $0,1\mu\text{F}$ a mezi RESET a +5V odpor $10\text{k}\Omega$.



Obrázek 33 - Zapojení programovacího portu

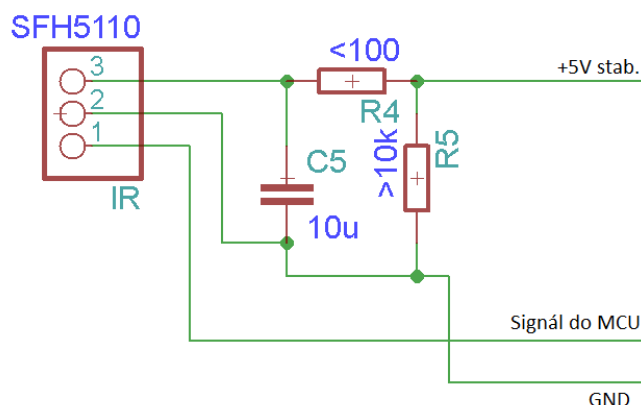
6.1.3 Zapojení IR přijímače

Pro zapojení IR přijímače bylo také dodrženo zapojení dle jeho manuálu.



Obrázek 34 - Zapojení IR přijímače SFH 5110-36

Vývod přijímače č. 1 je veden do mikropočítače na vývod 16, který nese časovač TPM1CH0. IR přijímač je opatřen demodulátorem, takže nám z něj bude přicházet obdélníkový signál do mikropočítače.

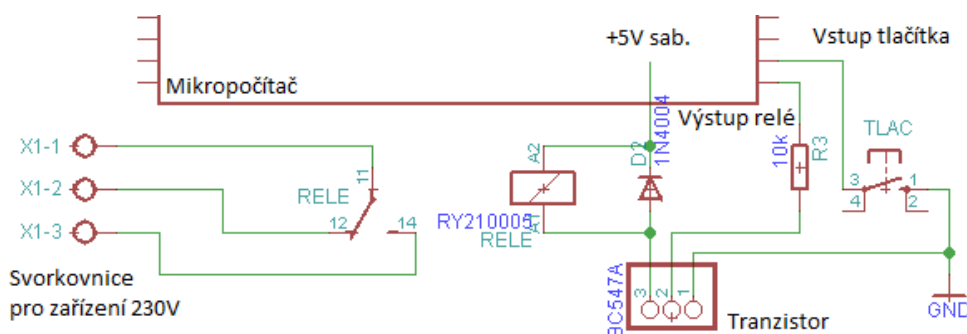


Obrázek 35 - Reálné zapojení IR přijímače

6.1.4 Vstupy a výstupy mikropočítače

Nejdůležitější je vstup z IR přijímače, o kterém byla zmínka výše. Dalším vstupem je signál z tlačítka, který bude spouštět pokyn učení. Při stisku tlačítka zařízení čeká na příchozí IR signál od ovladače, aby se mohlo naučit jeho kombinaci. Tím je zařízení nastaveno na určitý signál.

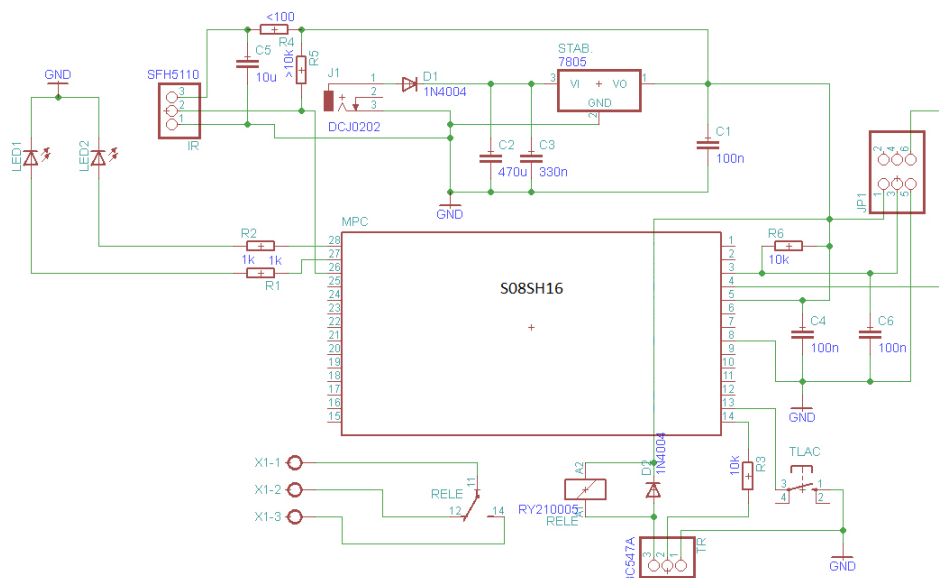
Výstupy máme tři. Ten nejdůležitější je výstup na relé. To je řešeno pomocí NPN tranzistoru. Výstup z mikropočítače je veden na bázi tranzistoru, který při otevření pustí stabilizovaný 5V signál na relé, které se přepne. Na relé je možno připojit dvě jakákoliv zařízení do 230V. Další dva výstupy jsou totožné. Jedná se o LED diody, ty jsou přes odpory 1k Ω vedeny na zem. Diody slouží jako indikace učení, popřípadě příchodu signálu na zařízení.



Obrázek 36 - Ukázka některých vstupů a výstupů

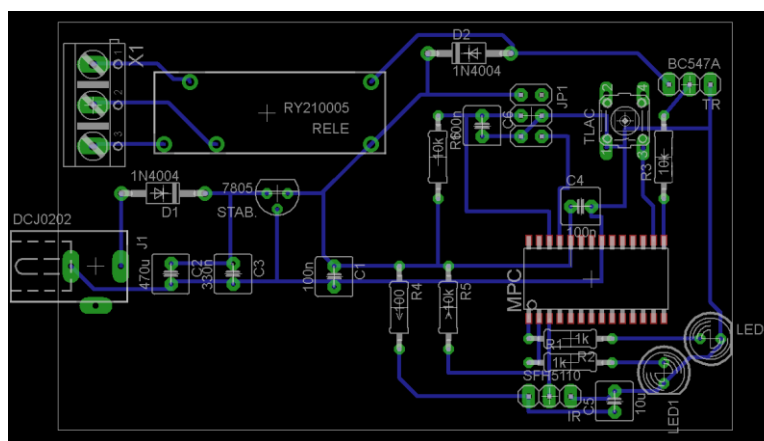
6.1.5 Schéma zapojení

Na schématu níže vidíme všechny výše popsané části. (J1 - konektor pro napájení, JP1 - programovací port, TR - NPN tranzistor, SFH5110 - infračervený přijímač a S08SH16 – mikropočítač)



Obrázek 37 - Návrh plošného spoje v programu Eagle 5.1.0

Při návrhu schématu, se v programu Eagle vytvářelo reálné zapojení tak, jak bude skutečně vypadat. Součástky bylo nutné na plochu poskládat a propojit vodivé cesty. Je snaha zařízení minimalizovat, proto jsme se snažili využít co nejvíce prostoru desky.



Obrázek 38 - Návrh desky v programu Eagle

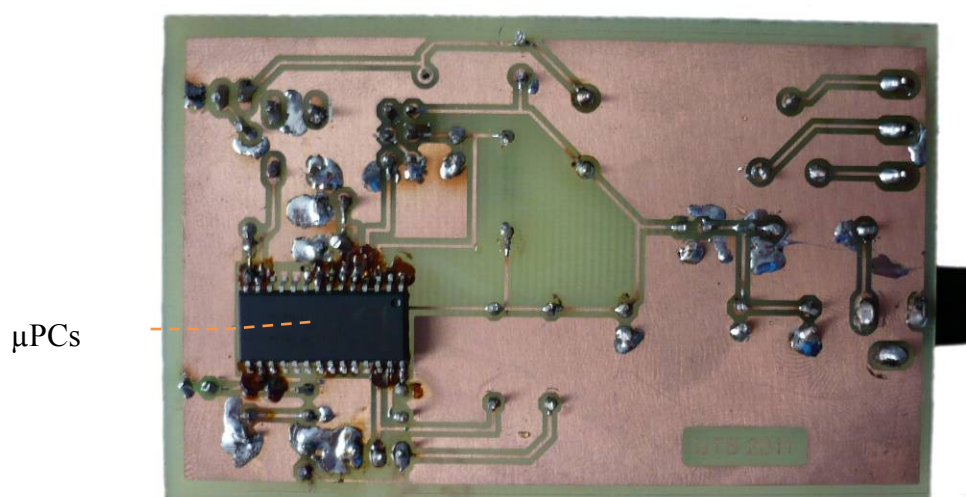
6.1.6 Konstrukce, problémy

Při konstrukci zařízení bylo zjištěno hned několik problémů. Některé se týkaly uspořádání součástek, jiné zapojení jednotlivých součástek.



Obrázek 39 - IR přijímač ze strany součástek

Pájení mikropočítače v pouzdře SOIC je poměrně obtížné, proto je nutné dbát na velkou přesnost.



Obrázek 40 - IR přijímač ze strany spojů

6.2 PARAMETRY ZAŘÍZENÍ

Délka: 85mm

Šířka: 53mm

Výška: 27mm

Napájecí napětí: 5 – 12V

Pozn.: Při napájení 12V a plné zátěži je nutné počítat s větším zahříváním stabilizátoru. Možno opatřit pasivním chlazením.

LED diody jsou na zařízení dvě. Kvůli variabilitě má jedna zelenou barvu a druhá barvu červenou. Za stabilizátorem je přesně 4,89V, což je dostatečné napětí pro provoz mikropočítače i všech dalších součástí zařízení.

7 SOFTWAREVÁ ČÁST

7.1 UVEDENÍ ZAŘÍZENÍ DO PROVOZU

Jelikož na zařízení není vypínač, stává se zařízení provozuschopné, jakmile k němu připojíme napájení. Pak samozřejmě záleží na programu, který je nahrán v paměti mikropočítače. Pokud je paměť prázdná, zařízení nedělá nic.

Pro samotné programování byl použit program CodeWarrior 5.9 od firmy Freescale, tudíž od výrobce mikropočítače. Aby byla možnost IR přijímač propojit s počítačem, je potřeba použít zařízení, které převede programovací konektor na USB. Zde je použit BDM programátor od firmy P&E Microcomputer Systems, který je vytvořen přímo pro programování mikropočítačů firmy Freescale. Zařízení není nutno instalovat. Program CodeWarrior bude automaticky nabízet možnost propojení mikropočítače pomocí USB Multilink Interface.



Obrázek 41 - Zařízení pro propojení s počítačem

Programování mikropočítače je možno provádět buď v jazyce symbolických adres, nebo jazyce C. Vzhledem k náročnosti softwaru, který se chystáme programovat, použijeme samozřejmě jazyk C. Softwarová část se bude skládat ze dvou kapitol, první se bude zabývat rozkódováním a uvedením zařízení do provozu pomocí protokolu RC-5. Druhá část bude řešit problém univerzálnosti. Výstupem tudíž budou dva programy.

7.2 ZÁKLADNÍ NASTAVENÍ

Na začátku programu byl vložen kód pro nastavení pevné frekvence sběrnice a to proto, abychom byli schopni na časovači nastavit určitou potřebnou hodnotu. Frekvence byla nastavena na 16MHz.

7.2.1 Nastavení vstupů a výstupů

Nastavení vstupů a výstupů je pro obě části stejné. Jelikož zařízení má na sobě výstupy – LED diody a relé, a vstupy – tlačítko a IR přijímač, musí se nastavit příslušné registry. Nastavení se provádí pomocí příkazu PTxDD, kde x značí port. Diody, relé a tlačítko je připojeno na port C. Proto do portu PTCDD musíme nastavit jako výstupy 1 a jako vstupy 0. Diody jsou na pinu 6 a 7, tlačítko je na pinu 3 a relé na pinu 2.

PIN	7	6	5	4	3	2	1	0
HODNOTA	1	1	0	0	0	1	0	0

Tabulka 3 – Nastavení vstupů a výstupů na portu C

Tudíž příkaz, který provedeme, bude: PTCDD=0b11000100;

Protože pin 3 je tlačítko, je potřeba na něm aktivovat Pull-up rezistory.

Příkaz: PTCPE_PTAPE0=1;

Na portu A je připojený jediný prvek, a to IR přijímač. Ten je připojen na časovači TPM1, který se nachází na pinu 0. Proto pin 0 nastavíme jako vstup.

Příkaz: PTADD_PTAD0=0;

7.2.2 Nastavení přerušení od IR přijímače

V první fázi bude nastaveno přerušení od IR přijímače. V manuálu k mikropočítači vyhledáme adresu, na které je přerušení od časovače 1 kanálu 0 (TPM1CH0), na který máme připojen IR přijímač. Adresa je 0xFFD6. Zvolíme si název funkce, která bude ukazovat na adresu vektoru přerušení. Tato funkce bude vyvolána jako obsluha přerušení.

Příkaz: void(*const obsluha) (void) @0xFFD6=pina0int;

Je nutné nastavit také, jestli se nám přerušení vyvolá sestupnou nebo vzestupnou hranou. Tato hodnota se nastavuje v registru PTAES na pinu 0 (PorT A Edge Select Register). Vzestupná hrana by byla 1, sestupná 0. Jelikož v klidném stavu máme na vstupu logickou 1, čekáme tedy na hranu sestupnou, proto do tohoto registru zapíšeme logickou 0.

Příkaz: PTAES_PTAES0=0;

Přerušení od IR přijímače se dá také velmi jednoduše při běhu programu programově povolit, nebo zakázat. K tomu slouží registr PTAPS (PorT A Pin Select Register). Pokud chceme přerušení povolit, nastavíme nultý bit registru PTAPS na jedničku.

Příkaz: PTAPS_PTAPS0=1;

Kdybychom chtěli přerušení zakázat, použijeme příkaz stejně, pouze změníme zapisovanou hodnotu na 0.

Část kódu, který bude volán jako obsluha přerušení, vypíšeme do funkce pina0int. Důležité při deklaraci funkce je slůvko interrupt, které musíme napsat před funkcí.

Příkaz: interrupt void pina0int (void) {}

V této funkci musíme také volat potvrzení přerušení, které se volá v registru PTASC (PorT A Select and Control Register) na bitu, který se jmenuje PTAACK (PorT A Acknowledge).

Příkaz: PTASC_PTAACK=1;

7.2.3 Nastavení přerušení od časovače

V manuálu mikropočítače vyhledáme adresu přerušení od časovače. Ta se nachází na adrese 0xFFE8. Zvolíme si název funkce, která bude ukazovat na adresu vektoru přerušení. Tato funkce bude vyvolána jako obsluha přerušení od časovače.

Příkaz: void(*const tobsluha) (void) @0xFFE8=timer_int;

Pro vyhodnocení signálu z IR přijímače budeme potřebovat sbírat vzorky po určitém čase, který bude velmi krátký. Časovač obsahuje nastavitelný prescaler. Ten umožňuje nastavit dělicí poměr hodinového kmitočtu. Dále máme 16bitový registr a komparátor čítače, který řídí časování intervalů. Do čítačového modulu registru můžeme zapsat hodnotu, která je porovnávána s hodnotou čítače. Při shodě hodnoty čítače

s hodnotou v modulo registru, je nastaven příznak přetečení TOF (Timer Overflow Flag) nebo je vyvoláno přerušení.

Základní složky časovače:

- Zdroj hodinového signálu
- Prescaler
- 16bitový čítač
- Časovací modulo registr
- Komparátor

V registru TPM1SC (Timer Pulse-width Modulator – Status and Control register) nastavíme jednotlivé bity. Nastavením prvních tří bitů (PS0-PS2) nastavíme prescaler. Ten je možno nastavit na sedm dělicích poměrů a tím generovat sedm různých frekvencí vstupních hodin čítače.

PS2	PS1	PS0	Délka 1 cyklu	Pro $f_{bus}=16\text{MHz}$
0	0	0	1	62,5ns
0	0	1	2	125ns
0	1	0	4	250ns
0	1	1	8	500ns
1	0	0	16	1 μ s
1	0	1	32	2 μ s
1	1	0	64	3 μ s
1	1	1	128	4 μ s

Tabulka 4 – Nastavení prescaleru časovače

Bity číslo 3 a 4 jsou CLKS_B:CLKS_A (CLOCK Source select). Jedná se o dvojici bitů použitých k zakázání časovače nebo výběru jednoho ze tří hodinových zdrojů pro prescaler.

CLKS _B :CLKS _A	Nastavení pro prescaler
0:0	Časovač zakázán
0:1	Frekvence sběrnice
1:0	Frekvence PLL a FLL
1:1	Externí zdroj (krystal)

Tabulka 5 – Nastavení vstupní frekvence

Další bit je CPWMS (Center-Aligned PWM Select). Pokud je rovný 1 jedná se PWM se zarovnáním na střed. Pokud je rovný 0, jedná se o zarovnání na hranu.

Šestým bitem je TOIF (Timer Overflow Interrupt Enable). Bit pro povolení žádosti při přetečení časovače (nastaven na 1), po resetu vždy nastaven na nulu.

Sedmý bit je TOF (Time Overflow Flag). Detekuje přetečení od čítače. TOF=1, čítač dosáhl hodnotu v modulo registru.

Registr TPM1SC tedy nastavíme celý jako binární hodnotu a to TOF na nula, TOIF na jedna, CPWMS na nula, dvojbit CLKSB:CLKSA na 0:1, tudíž frekvence sběrnice a PS2:PS1:PS0 na hodnotu 0:0:0, takže prescaler na 1.

PIN	TOF	TOIF	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
HODNOTA	0	1	0	0	1	0	0	0

Tabulka 6 – Nastavení registru TPM1SC

Příkaz: TPM1SC = 0b01001000;

Další důležitý registr je TPM1MOD (TPM 1 counter MODulo registr). Jedná se o 16bitový registr časovače. Obsahuje hodnotu pro čítač, které když čítač dosáhne, nastaví se příznak přetečení (TOF) a čítač začne znova čítat od nuly.

Vzorec pro hodnotu modula:

$$Modulo = \frac{t_{TOF} \cdot f_{SOURCE}}{prescaler}$$

Modulo – údaj, který zapíšeme do modulo registru

t_{TOF} – požadovaný čas do přetečení časovače [s]

f_{SOURCE} – frekvence zdroje hodinového kmitočtu

prescaler – nastavení vstupní frekvenční děličky

Příklad:

Pro protokol RC-5 budeme potřebovat čtvrtinu délky bitu. Délka jednoho bitu je 1,778ms.

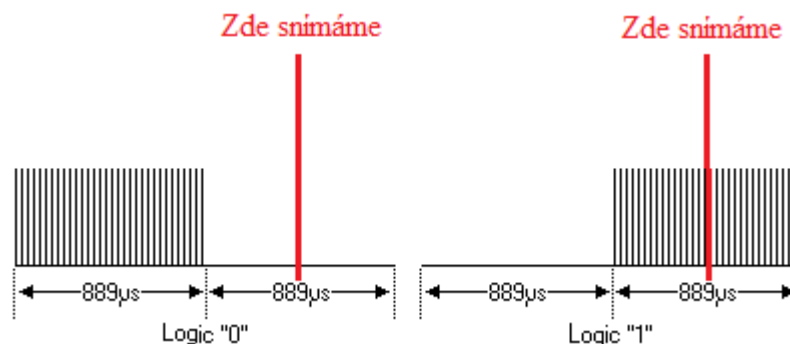
Tudíž je požadovaný čas 444,5μs.

$$Modulo = \frac{0,000\,444\,5 \cdot 16\,000\,000}{1} = 7112$$

7.3 PROGRAM PRO PROTOKOL RC-5

7.3.1 Rozkódování RC-5

Základní myšlenka rozkódování RC-5 je ve snímání hodnoty na IR přijímači ve $\frac{3}{4}$ času bitu. Jelikož vyhodnocení logické 0 a 1 u protokolu RC-5 spočívá ve změně v půlce času jednoho bitu, je zřejmé, že nám postačí snímat ve $\frac{3}{4}$ času bitu.



Obrázek 42 - Snímání RC-5

Jak jde vidět, $\frac{3}{4}$ času bitu v podstatě odpovídá logické úrovni jednotlivého celého bitu. Abychom snímali ve správném čase, počítáme modulo dle ukázky v kapitole 7.2.

Je potřeba si uvědomit, že za klidného stavu je logické úroveň 1 na výstupu IR přijímače a tudíž na vstupu mikropočítače. Takže vyhodnocování musíme invertovat.

Délka celého datového rámce RC-5 je 14 bitů po 1,778ms. První dva bity jsou start bity. Další je toggle bit, který se mění, mačkáme-li stejné tlačítko. Následuje pět datových bitů a nakonec šest příkazových bitů. Všechny tyto bity budeme snímat a ukládat do pole.

Snímání dle času začneme, jakmile nám přijde přerušení od sestupné hrany. První vyhodnocení musí přijít po $\frac{1}{4}$ času, protože první bit je vždy jedna a tudíž má již za sebou půlku logické 1. Poté snímáme vždy po celém bitu, protože se již nacházíme v $\frac{3}{4}$ bitu. Snímáme po 4 taktech a ukládáme do pole. Snímání se ukončí, jakmile se index pole rovná 13, tudíž máme sejmuto 14 hodnot. Jelikož je doba celého rámce velmi krátká, tak každý bit snímáme 5krát. Pokud převažuje počet jedniček, je bit uznán jako jednička. Pokud převažuje počet nul, je bit uznán jako nula.

7.3.2 Funkce programu pro protokol RC-5

Po spuštění je program v nekonečné smyčce, kde čeká na přerušení od IR přijímače, časovač běží a při dosažení čítače hodnoty modulo (7112) se vynuluje. Přerušení přijde při příchodu sestupné hrany na IR přijímač (stisk tlačítka na dálkovém ovladači). To indikuje bliknutí červené LED diody. Spustí se obsluha přerušení od sestupné hrany, ve které zakážeme přerušení od sestupné hrany. Potřebujeme totiž, aby se po dobu načítání datového rámce obsluha přerušení od sestupné hrany nevyvolávala. Povolí se načítání do pole. Když je pole plné, volá se funkce `cislo()` a povoluje se přerušení od sestupné hrany. V této funkci se hodnota příkazových bitů převede na dekadické číslo a uloží do proměnné `data`. Je také možnost nahlédnout v průběhu programu do proměnné `data`, ve které se nám dekadické číslo ukáže. Pokud stiskneme tlačítko 0 – 9 na ovladači, dekadická hodnota převedeného binárního čísla odpovídá hodnotě stisknutého tlačítka.

Pokud chceme tlačítko dálkového ovladače uložit, abychom mohli později porovnávat, potřebujeme tlačítko na zařízení. Při stisku tlačítka a příjmu sestupné hrany se provede vše výše uvedené a navíc se hodnota dekadického čísla uloží do proměnné `data_save`. V ní zůstane uložena až do doby, kdy budeme chtít uloženou hodnotu změnit. Uložení hodnoty indikuje rozsvícení obou diod na krátkou dobu.

Při každém dalším příjmu signálu od ovladače, se nám provádí porovnání uložené proměnné `data_save` s novou hodnotou proměnné `data`. Jakmile se hodnoty rovnají, je volána funkce `blikani()`, kde nám třikrát zablikají obě led diody a přepne se relé z polohy 1 do polohy 2, nebo naopak.

Jelikož nastavení přerušení, časovače, vstupů a výstupů jsme si ukázali výše, uvedeme pouze části volaných funkcí.

main() – hlavní funkce, která se volá na začátku programu, je v ní mimo jiné nastavení všech registrů, porovnání signálů, volání funkcí

```
void main(void) {  
    ⋮  
    TPM1MOD=7112;                //nastavení modula na hodnotu 7112  
    EnableInterrupts;            //povolení všech přerušení  
    for(;;) {                    //nekonecny cyklus for  
        if (ok==1) {             //jestliže je přijat platný příkaz  
            cekej();              //volá se funkce cekej()  
        }  
    }  
}
```

```

    cislo(); //volá se funkce cislo()
    if (PTCD_PTCD3==0){ //jestliže je stisknuto tlačítko
        data_save=data; //do proměnné data_save uložíme data
        data=0; //vynuluje data
        PTCD_PTCD6=1; //rozsvítí zelenou diodu
        PTCD_PTCD7=1; //rozsvítí červenou diodu
        cekej(); //volá funkci cekej()
        PTCD_PTCD6=0; //zhasne zelenou diodu
        PTCD_PTCD7=0; //zhasne červenou diodu
    }
    povol_preruseni(); //povolí přerušení od IR přijímače
    ok=0; //nastav proměnnou ok na 0
    if (data==data_save){ //jestliže je platný výrok v závorce
        blikani(); //volá funkci blikani()
    }
    PTCD_PTCD7=0; //zhasne červenou diodu
    PTCD_PTCD6=0; //zhasne zelenou diodu
}
__RESET_WATCHDOG(); //reset watchdog
}
}

```

pina0int() – obsluha přerušení od sestupné hrany

```

interrupt void pina0int (void) {
    if (test()==0){ //jestliže má funkce test návratovou
                    //hodnotu 0
        PTCD_PTCD7=0; //zhasni červenou diodu
    } else { //jinak
        PTCD_PTCD7=1; //rozsvítí červenou diodu
    }
    prvni=1; //do proměnné první přiřadí 1
    o=0; // do proměnné o přiřadí 0
    inc=0; // do proměnné inc přiřadí 0
    TPM1SC&=casovac; //vynuluje čítač
    PTASC_PTAACK=1; //potvrzení přerušení
    zakaz_preruseni(); //zakáže přerušení od IR přijímače
}

```

timer_int() – obsluha přerušení od časovače, obsahuje plnění pole o indexu inc

```

interrupt void timer_int (void) {
    o++;                                //inkrementuju proměnnou o
    if(o==1&&prvni==1){                 //jestliže jsou splněny podmínky
        if (test()==1){                 //jestliže funkce test vrazí 1
            save[inc]=1;                 //do pole save o indexu inc ulož 1
        } else {                         //jinak
            save[inc]=0;                 //do pole save o indexu inc ulož 0
        };
        inc++;                           //inkrementuju inc
        o=0;                             //do proměnné o přiřadí 0
        prvni=0;                         //do proměnné první přiřadí 0
    };
    if(inc<15&&inc>0){                  //jestliže jsou splněny podmínky
        if(o==4){                       //jestliže je splněna i tato podmínka
            if (test()==1){              //jestliže funkce test vrací 1
                save[inc]=1;              //do pole save o indexu inc ulož 1
            } else {                     //jinak
                save[inc]=0;              //do pole save o indexu inc ulož 0
            };
            inc++;                       //inkrementuj inc
            o=0;                         //do proměnné o přiřadí 0
        };
    };
    if(inc==14){                         //jestliže je splněna podmínka
        inc=0;                           //do proměnné inc přiřadí 0
        ok=1;                            //do proměnné ok přiřadí 1
    }
    TPM1SC&=casovac;                    //nuluje příznak přerušení
}

```

test() – funkce, ve které se vyhodnocuje převaha jedniček, popřípadě nul, pokud převažují jedničky, funkce má návratovou hodnotu 1 a naopak

```

char test (void) {
    char k;                             //proměnná k typu char
    char l=0;                           //proměnná l typu char
    for (k=5;k>0;k--){                  //pro k=5 do k>0 odečítej 1

```

```
    if (PTAD_PTAD0==0){                //jestliže je na pinu od IR přijímače 0
        l++;                            //inkrementuj proměnnou l o 1
    };
};
if (l>2){                              //jestliže je splněna podmínka
    return 1;                          //vrať 1
} else {                               //jinak
    return 0;                          //vrať 0
};
}
```

cislo() – převede binární číslo na dekadické

```
void cislo (void){
    int maska;                          //proměnná maska typu int
    char j;                             //proměnná j typu char
    maska = 0b00000001;                 //nastavíme masku
    data=0;                             //vynulujeme proměnnou data
    for (j=13;j>7;j--){                //pro j=13 do j>7 odečítej od j
        if (save[j]==1){                //jestliže je v save o indexu j 1
            data|=maska;                //proved logický OR s maskou
        }
        maska=maska<<1;                //posuň masku vlevo
    }
}
```

Další funkce jsou velmi jednoduché, a proto si uvedeme pouze jejich názvy a vlastnosti.

blikani() – funkce, která zabliká LED diodami a přepne relé

cekej() – funkce, která čeká 32000 taktů – asi 2ms

7.4 UNIVERZÁLNÍ PROGRAM

7.4.1 Teoretická myšlenka

Nejrozšířenějším protokolem v Evropě je RC-5, proto jsme naše zařízení osadili přijímačem pro nosné frekvence 36kHz. Ovšem ani příjem frekvencí od jiných protokolů než RC-5 nebude problém. Příjem je stále dostatečně kvalitní pro rozkódování, uložení znaků a porovnávání.

Jelikož chceme udělat zařízení co nejvíce univerzální, musíme počítat s tím, že nám můžou přijít různé signály. V první řadě je potřeba určit, jak dlouhý čas od příjmu první sestupné hrany budeme přijímat.

7.4.2 Konstanty

Do programu implementujeme volitelné nastavení konstant programátorem. Jedná se o část kódu, kde si bude moct programátor určit několik voleb. Tou první je rychlost přerušení. Vše je přednastaveno jako definice, proto bude stačit, když si programátor na začátku kódu zvolí jednu z předem nastavených definic. Jedná se vlastně o nastavení hodnoty modula dle kapitoly 7.2.3. Nastavení přerušení od časovače.

Definice vzorkování:

```
#define vzorkovani_100us TPM1MOD=1600           //vzorkovani 100us
#define vzorkovani_200us TPM1MOD=3200           //vzorkovani 200us
#define vzorkovani_400us TPM1MOD=6400           //vzorkovani 400us
#define vzorkovani_600us TPM1MOD=9600           //vzorkovani 600us
#define vzorkovani_800us TPM1MOD=12800          //vzorkovani 800us
#define vzorkovani_1ms TPM1MOD=16000            //vzorkovani 1ms
#define vzorkovani_2ms TPM1MOD=32000            //vzorkovani 2ms
```

Programátor poté jen do kódu přidá např.:

```
vzorkovani_1ms;
```

Velikost pole, do kterého si budeme ukládat logické hodnoty v čase přerušení, bude 100. To znamená, že při vzorkování 2ms uložíme do pole 0,2 sekundy. V pozdější kapitole se budeme zabývat nejuniverzálnějším rozložením přerušení.

Jelikož budeme odebírat vzorky v podstatě náhodně, nebudeme schopni rozpoznat, o jaký kód se jedná. To pro nás ovšem není vůbec podstatné. Důležité je, aby se uložený kód co nejvíce rovnal kódu přijatému. Porovnávání probíhá ve funkci `porovnej()`, kde se uložené pole bitů porovnává s právě přijatým polem bitů.

Protože některé kódy jsou opatřeny bity, které se mění při delším držení tlačítka, je potřeba, abychom byli schopni tyto bity z porovnávání odstranit. Proto zavádíme další volbu konstant. Jedná se o dvě hodnoty mezi – dolní mez a horní mez. Tyto meze ukazují funkci `porovnej()`, od kterého bitu a po který bit se budou pole porovnávat.

Příklad volby mezí:

```
mez1=0;                //volba dolni meze
mez2=99;                //volba horni meze
```

I přes veškerá opatření se nám stane, že kód uložený a porovnávaný nesouhlasí. Je to dané také drobnou chybou při přenosu, jinou nosnou frekvencí, příliš dlouhými, nebo naopak krátkými vzorky. Proto opatříme kód další volbou konstanty, a to akceptací chyby. Tou nastavujeme, při kolika chybných porovnáních přijmeme kód jako stejný. Čím větší tato akceptace je, tím nám bude kód přijímat i jiná tlačítka na ovladači.

Příklad volby akceptace chyby:

```
rozliseni = 3;          //volba akceptace chyby
```

Programátor by samozřejmě neměl měnit nastavení prescaleru, protože poté by časy přerušení definované na začátku neplatily.

7.4.3 Funkce univerzálního programu

Po spuštění je program také v nekonečné smyčce, kde čeká na přerušení od IR přijímače, časovač běží a při dosažení čítače hodnoty definované ve volbě konstant, se vynuluje. Přerušení přijde při příchodu sestupné hrany na IR přijímač (stisk tlačítka na dálkovém ovladači). To indikuje bliknutí červené LED diody. Spustí se obsluha přerušení od sestupné hrany, ve které zakážeme přerušení od sestupné hrany. Potřebujeme totiž, aby se po dobu načítání datového rámce obsluha přerušení od sestupné hrany nevyvolávala. Povolí se načítání do pole. Když je pole plné, volá se funkce `porovnej()` a povoluje se přerušení od sestupné hrany.

Pokud chceme tlačítko dálkového ovladače uložit, abychom mohli později porovnávat, potřebujeme tlačítko na zařízení. Při stisku tlačítka a příjmu sestupné hrany se provede vše výše uvedené a navíc se pole[] uloží do proměnné save[]. V něm zůstane uloženo až do doby, kdy budeme chtít změnit uložené tlačítko na ovladači. Uložení pole indikuje rozsvícení obou diod na krátkou dobu.

Při každém dalším příjmu signálu od ovladače, se nám provádí porovnání uloženého pole save[] s novým polem pole[]. Samozřejmě s ohledem na meze, nastavené programátorem. Jakmile se pole rovnají (s ohledem na akceptaci chyby) je volána funkce blikani(), kde nám třikrát zablikají obě LED diody a přepne se relé z polohy 1 do polohy 2, nebo naopak.

Jelikož nastavení programátorského rozhraní, přerušení, časovače, vstupů a výstupů jsme si ukázali výše, uvedeme pouze části volaných funkcí.

main() – hlavní funkce, která se volá na začátku programu, je v ní mimo jiné nastavení všech registrů, porovnání signálů, volání funkcí

```
void main(void) {  
    :  
    vzorkovani_800us;                //volba konstanty  
    rozliseni = 3;                  //volba konstanty  
    mez1 = 0;                       //volba konstanty  
    mez2 = 99;                      //volba konstanty  
    EnableInterrupts;               //povolení všech přerušení  
    for(;;) {                       //nekonečný cyklus for  
        if (ok==1) {               //jestliže je splněna podmínka ok=1  
            if (PTCD_PTCD3==0){    //a jestliže je stisknuto tlačítko  
                PTCD_PTCD6=1;      //rozsvítí zelenou diodu  
                PTCD_PTCD7=1;      //rozsvítí červenou diodu  
                cekej();            //volání funkce cekej()  
                PTCD_PTCD6=0;       //zhasne zelenou diodu  
                PTCD_PTCD7=0;       //zhasne červenou diodu  
            }  
            povol_preruseni();      //povolí přerušení od IR přijímače  
            ok=0;                  //do proměnné ok přiřadí 0  
            for (q=5;q>0;q--){     //pro q rovno 5, do q rovno nule, odečítá  
                porovnej();        //volá funkci porovnej
```

```

        if (por1>=(mez2-mez1-rozliseni)){           //jestliže je por1 >= naší volbě chyby
            rozhodni++;                             //inkrementuje rozhodni
        }
    }
    if (rozhodni>3){                                 //je-li splněna podmínka
        blikani();                                   //volejá funkci blikani()
        rozhodni=0;                                  //vynuluje rozhodni
    } else{                                          //jinak
        rozhodni=0;                                  //vynuluje rozhodni
    }
}
__RESET_WATCHDOG();                               //reset watchdog
}
}

```

pina0int() – obsluha přerušení od sestupné hrany

```

interrupt void pina0int (void) {
    if (test()==0){                                 //jestliže funkce test vrátí 0
        PTCD_PTCD7=0;                               //zhasne červenou diodu
    } else {                                         //jinak
        PTCD_PTCD7=1;                               //rozsvítí červenou diodu
    }
    prvni=1;                                         // do proměnné prvni přiřadí 1
    o=0;                                             // do proměnné o přiřadí 0
    inc=0;                                           // do proměnné inc přiřadí 0
    TPM1CNT=0;                                       //vynuluje čítač
    PTASC_PTAACK=1;                                  //potvrzení přerušení
    zakaz_preruseni();                              //zakáže přerušení od IR přijímače
}

```

timer_int() – obsluha přerušení od časovače, obsahuje plnění pole o indexu inc

```

interrupt void timer_int (void) {
    o++;                                             //inkrementuj o
    if(o<100&&prvni==1&&PTCD_PTCD3==1){           //jestliže je inc<100, prvni=1,tlačítko=1
        if (test()==1){                             //jestliže je test vrátí 1
            save[inc]=1;                              //do pole save o indexu inc uloží 1
        } else {                                     //jinak
            save[inc]=0;                              //do pole save o indexu inc uloží 0
        }
    }
}

```

```

};
    inc++;                                //inkrementuj inc
};
if(o<100&&prvni==1&&PTCD_PTCD3==0){      //jestliže je inc<100, prvni=1,tlačítko=0
    if (test()==1){                      //jestliže test vrátí 1
        pole[inc]=1;                    //do pole o indexu inc uloží 1
    } else {                             //jinak
        pole[inc]=0;                    //do pole o indexu inc uloží 0
    };
    inc++;                                //inkrementuj inc
};
if(inc==99){                             //jestliže je splněna podmínka
    inc=0;                              //vynuluje inc
    ok=1;                                //do proměnné ok přiřadí 1
    prvni=0;                             //do proměnné prvni přiřadí 0
}
TPM1SC&=casovac;                         //nuluje příznak přerušení
}

```

test () – funkce test je stejná jako u programu pro protokol RC-5

porovnej () – porovnává dvě pole od dolní meze po horní mez, pokud se rovnají, inkrementuje proměnnou por1, kterou vyhodnocujeme

```

void porovnej (void){
    char j;                              //proměnná j typu char
    por1=0;                              //do proměnné por1 přiřadí 0
    for (j=mez1;j<mez2;j++){             //pro j=mez1 do j<mez2, přičítá 1
        if(save[j]==pole[j]){            //jestliže se pole rovnají
            por1++;                       //inkrementuje por1
        }
    }
}

```

Další funkce jsou velmi jednoduché, a proto si uvedeme pouze jejich názvy a vlastnosti.

blikani() – funkce, která zabliká LED diodami a přepne relé

cekej() – funkce, která čeká 32000 taktů – asi 2ms

7.5 TESTOVÁNÍ UNIVERZÁLNÍHO PROGRAMU

Abychom správně nastavili program (volba konstant), budeme naše zařízení testovat na několika dálkových ovladačích různých výrobců. Pokud to bude možné, tak se pokusíme odhadnout, o jaký protokol se jedná. Nakonec se pokusíme navrhnout nejuniverzálnější řešení.

Základní nastavení, na které budeme testovat ovladače, bude:

- Vzorkování: 400 μ s
- Akceptace chyby: 5
- Meze: (0,99)

Druhé nastavení, na které budeme testovat ovladače, bude:

- Vzorkování: 800 μ s
- Akceptace chyby: 5
- Meze: (20,99)

Třetí nastavení, na které budeme testovat ovladače, bude:

- Vzorkování: 800 μ s
- Akceptace chyby: 3
- Meze: (0,99)

7.5.1 Ovladač LG – MKJ42519618



Obrázek 43 - Ovladač LGMKJ42519618

Jedná se o ovladač k LCD televizoru LG. Při prvním příjmu signálu vidíme, že se celý datový rámec nevejde do pole. Dále se ukazuje, že při akceptaci chyby 5, akceptuje zařízení téměř všechna tlačítka ovladače.

Při druhém nastavení ovladač funguje spolehlivě. Akceptuje pouze naučené tlačítko. Při držení stejného tlačítka je využito asi 15% pole, takže se jedná o nějaký opakovací signál. Vzhledem k délce rámce a opakovacího rámce odhadujeme, že se jedná o protokol NEC.

Při třetím nastavení se akceptuje pouze tlačítko uložené v paměti, někdy ovšem díky malé akceptaci nepřijme ani toto tlačítko.

7.5.2 Ovladač Optibox Alligator



Obrázek 44 - Ovladač Optibox Alligator

Jedná se o ovladač od set-top boxu Alligator. Kromě ovládání set-top boxu, se dá nastavit také jako univerzální ovladač na několik televizorů. Při základním nastavení programátorského rozhraní vidíme, že se datový rámec nevešel do pole.

Při druhém nastavení se také objevuje opakovací signál, takže je velká pravděpodobnost, že se jedná opět o protokol NEC. Zařízení ovšem akceptuje i jiná tlačítka, než naučené.

U třetího nastavení už ovladač akceptuje pouze tlačítko uložené v paměti.

7.5.3 Ovladač VRC Panasonic



Obrázek 45 - Ovladač VRC Panasonic

Ovladač je od videorekordéru Panasonic. Zdá se, že signál do zařízení přichází hůře, než u ostatních ovladačů, takže můžeme předpokládat, že se jedná o jinou nosnou frekvenci. Datový rámec se ale ani tady nevejde do pole.

Při druhém nastavení ovladač funguje spolehlivěji. Akceptace projde pouze u naučeného tlačítka a při asi třech dalších. Ty ale musíme podržet déle. Při delším držení se nám do pole uložil také opakovací signál, který ale nezačíná sledem jedničkových bitů, takže můžeme vyloučit NEC i RC-5. Při uložení do pole jde vidět, že se nám uložily čtyři buňky jako jedničky, což značí délku kolem 3ms. Tudíž můžeme odhadnout, že se jedná o protokol Sony SIRC.

Při třetím nastavení se ovladač chová nejspolehlivěji. Akceptace je možná příliš natěsno.

7.5.4 Ovladač HiFi Philips



Obrázek 46 - Ovladač HiFi Philips

Jelikož se jedná o ovladač od HiFi soustavy Philips, je dost pravděpodobné, že bude komunikovat protokolem RC-5. Podíváme-li se na datový rámeček, vidíme, že se zaplnil zhruba z 60%, což značí čas kolem 23ms. Jelikož víme, že datový rámeček u RC-5 by měl být zhruba 24ms, je téměř jasné, že se jedná o protokol RC-5. Ovladač navíc se zařízením komunikuje dobře, při základním nastavení ovladač funguje, naučené tlačítko spouští blikání i relé a ostatní tlačítka neakceptuje.

Druhé nastavení funguje hůře než první. Datový rámeček se nám uložil zhruba do 30% pole, což je už poměrně málo. Navíc dolní mez je někde u 20%, takže se nám vyhodnocuje zhruba 10 buněk. Tím pádem při akceptaci 5, projde většina tlačítek, což je nežádoucí.

Třetí nastavení s ovladačem funguje spolehlivě. Někdy není přijato uložené tlačítko, ale to pouze, pokud ho stiskneme na velmi krátkou dobu.

7.5.5 Ovladač LG - 6711R1P070B



Obrázek 47 - Ovladač LG – 6711R1P070B

Jedná se o ovladač od DVD přehrávače LG. Datový rámeček se do pole nevejde, ale zdá se, že ovladač relativně funguje. Při akceptaci jiných tlačítek, než naučeného, se chová tak, že uzná tlačítka, pouze pokud se drží delší dobu. Tudíž hraje roli náhoda.

Při druhém nastavení funguje ovladač trochu lépe. Stále ovšem akceptuje tlačítka, která nejsou uložena. Pole je zaplněno do 75%, takže vyloučíme protokol RC-5. Na začátku pole vidíme sled jedniček, ovšem jen krátký. Tudíž můžeme vyloučit i protokol NEC.

Třetí nastavení vylučuje akceptaci tlačítek jiných než uloženého, ovšem i uložené někdy není přijato. Pravděpodobně se v poli mění nějaký bit, který převrací hodnoty více než tří buněk v poli.

7.5.6 Ovladač UET MC-7



Obrázek 48 - Ovladač UET MC-7

Ovladač UET MC-7 je univerzální ovladač, který se dá nastavit až na sedm zařízení. My použijeme nastavení TV, protože víme, že se jedná o RC-5. Datový rámec dosáhl zhruba 60% pole. Akceptace funguje dobře, u ostatních tlačítek se objevuje akceptování pouze při delším držení, a to ještě pouze u asi dvou tlačítek.

Jelikož se jedná o RC-5, ovladač při druhém nastavení spíše nefunguje. Akceptace proběhne u většiny tlačítek.

Třetí nastavení pomohlo vylepšit funkčnost, a to díky menší akceptaci chyby. Samozřejmě je ale pole stále využito jen z 30%, takže zařízení akceptuje více tlačítek než jen uložené.

7.5.7 Ovladač HUMAX RM-106



Obrázek 49 - Ovladač HUMAX RM-106

HUMAX RM-106 je ovladač od set-top boxu HUMAX. Při načtení do pole se opět rámec nevešel. Akceptace vypadá dobře, ale není příliš spolehlivá. Občas se přijme i jiné tlačítko, než naučené.

Při druhém nastavení je datový rámec uložen celý. Zařízení ovšem akceptuje i další tlačítka než je naučené. Při držení stejného tlačítka vidíme využito asi 15% pole, takže se jedná o nějaký opakovací signál. Vzhledem k délce rámce a opakovacího rámce odhadujeme, že se jedná o protokol NEC.

Třetí nastavení je pro ovladač optimální. Díky snížení akceptace se občas stane, že ovladač nepřijme stisk uloženého tlačítka, ale jedná se o výjimky. Ostatní tlačítka už zařízení neakceptuje vůbec.

7.5.8 Vyhodnocení

Provedeme vyhodnocení jednotlivých nastavení. Nastavení budeme bodovat od jednoho do tří bodů, podle toho, jak se v určitém nastavení ovladač choval. Nejlepší nastavení bude ohodnoceno třemi body, nejhorší jedním bodem.

Název	První nastavení	Druhé nastavení	Třetí nastavení	Odhad protokolu
LG – MKJ42519618	1	2	3	NEC
Optibox Alligator	1	2	3	NEC
VRC Panasonic	1	2	3	SONY SIRC
HiFi Philips	3	1	2	RC-5
LG - 6711R1P070B	1	2	3	SONY SIRC
UET MC-7	3	1	2	RC-5
HUMAX RM-106	2	1	3	NEC
Součet bodů	12	11	19	

Tabulka 7 – Porovnání nastavení zařízení

Jak je vidět, nejlépe se IR přijímač chová při třetím nastavení. Jako optimální možnost se tedy zdá zvýšit akceptaci na 4 a zbytek nechat podle třetího nastavení.

Optimální nastavení:

- Vzorkování: 800 μ s
- Akceptace chyby: 4
- Meze: (0,99)

7.5.9 Automatická volba konstant

Aby byl program ještě více univerzální, byla by možnost automatické volby konstant. V tuhle chvíli, musí programátor sám konstanty nastavit. Kdyby program po nesprávném přijetí znaků sám upravoval nastavení konstant, možná bychom se dostali k ještě přesnějším výsledkům. Je ovšem nutné brát v úvahu, že pokud by program testoval pravděpodobnost přijatého signálu, asi by se stávalo, že by při stisku velkého počtu tlačítek neuložených v paměti přeměnil konstanty natolik, že by poté přijímal i signály z jiných tlačítek než z tlačítka uloženého.

Asi jedinou reálnou možností by byla volba modula. Jestliže by se například signál načetl jen do poloviny pole, program by mohl automaticky snížit hodnotu modula a tím zvýšit rychlost odebírání vzorků z IR přijímače. Pokud by ale signál zaplňoval pole, museli bychom ukládat i vzorky mimo pole, a na základě počtu přetečených vzorků modulo upravovat.

ZÁVĚR

Tato diplomová práce měla za cíl vytvořit zařízení, které bude sloužit jako univerzální přijímač infračerveného záření z co největší množiny běžně používaných dálkových ovladačů. Jedním z nejdůležitějších prvků zařízení je mikropočítač firmy Freescale Semiconductors S08SH16. Díky jeho velké vybavenosti byla možnost na jeho vstupy a výstupy umístit další součásti jako třeba IR přijímač. Ten je opatřen dekodérem, takže nám do mikropočítače přichází logické hodnoty (jedna, nula). Zařízení je také opatřeno výstupními prvky jako led diody a relé.

V teoretické části se práce zabývá historií dálkového ovládání a IR přenosu, druhy modulace IR záření a nepoužívanějšími přenosovými protokoly. Jak bylo řečeno výše, nejdůležitější součástí je mikropočítač S08SH16, proto je v teoretické části kapitola o jednočipových počítačích a firmě Freescale Semiconductor.

Praktická část je rozdělena do tří kapitol. První je popis částí zařízení. V ní je popis použitého mikropočítače, jeho optimálním zapojení a řešení zapojení IR přijímače. Hardwarová kapitola teoretické části diplomové práce se zabývá návrhem a výrobou tištěného spoje. Asi nejdůležitější kapitolou je část softwarová. Ta je rozdělena do několika podkapitol. Jedna z prvních je věnována rozkódování protokolu RC-5. To je řešeno uložením datového rámce do pole, a následného porovnání polí. Tato metoda se zdá být poměrně provozuschopná. Nastávají ovšem drobné problémy při různých protokolech. Některé mají velmi krátký datový rámec, takže se pole vyplní jen z několika procent, některé mají datové rámce naopak velmi dlouhé, takže pole ani nestačí. Tento problém byl odladen díky volbě konstant. Programátor má možnost zvolit rychlost snímání vzorků signálu a tím dobře vyplnit pole. Dále je volitelný počet chybných porovnání a meze, ve kterých buňkách se má porovnání provést.

Velmi důležitou částí diplomové práce je testování zařízení na dálkových ovladačích od různých výrobců, tudíž různých i protokolů. Sedm různých dálkových ovladačů, bylo testováno na několika možných nastaveních zařízení. Výsledky funkčnosti jednotlivých nastavení bylo porovnáno a tím zjištěno optimální nastavení. Při časování 0,8ms, akceptaci 4 chyb a porovnávání v celém poli se zdá být jako nejlepší varianta. Při vyhodnocování byly odhadovány protokoly, kterými dálkové ovládání komunikují. Výsledky byly poměrně překvapivé. Podle některých zdrojů, nejrozšířenější Evropský

protokol RC-5 se ocitl až na druhém místě. Nejčastější byl protokol NEC a na třetím místě se umístil protokol SONY SIRC. Vzhledem k tomu, že ovladačů bylo použito pouze sedm, nelze protokol RC-5 sesadit z trůnu, ovšem zjištění, že v běžné domácnosti je více ovladačů komunikující protokolem NEC, je určitě zajímavé.

Na přiloženém CD se nachází text diplomové práce, zdrojové kódy k programu pro RC-5 protokol a pro univerzální program.

CONCLUSION

The goal of this thesis was to create a device that will be used as a universal infrared signal receiver that can be used with the widest possible selection of commonly used remote controls. One of the most important parts of the device is a microcomputer from the company Freescale Semiconductors S08H16. Thanks to its high amount of inputs and outputs, I was able to directly connect it with an infrared receiver. This receiver is equipped with a decoder, thus providing the microcomputer with logical values of 1 and 0. The device has also other output accessories such as LEDs or a relay.

In the theoretical part, I firstly focused on the history of remote controls and infrared transmission. Secondly I mentioned types of IR emission and the most commonly used transmission protocols. As I have already mentioned, the most important part of the device is the microcomputer, therefore I have included information about single-chip computers and the company Freescale Semiconductor.

The practical part of my thesis is separated into three chapters. The first is a description of the parts that make up our device. Here we find information about the used microcomputer, his optimal integration and the solution for connecting the IR receiver and relay. The chapter about hardware in this dissertation focuses on the design and production of a printed circuit. Probably the most important chapter is, however, about the software. After successfully putting the device into operation, solving problems and testing the function of the in/outputs, we started to program our device for the RC-5 protocol. We had to find out how to set the right cutoff from the IR receiver and timer. After setting the cutoff, our work on the device started to go in the right direction with positive results. After finding a way of decrypting the RC-5, we began to explore the possibilities that this device has to offer. We chose the path of saving a data frame into an array and then comparing these arrays. This method seems rather functional. The problems came when we started using different types of protocols, some of which had a very small data frame, leaving a high percentage of free space and others with a data frame that was too long and therefore our array did not have enough capacity. This problem was solved using a programming interface, which we programmed. The user has the ability to choose the speed of reading out samples, providing effective use of the array. One can also set the amount of allowed false comparisons and circumferences, within the comparison is to be made.

A very important segment is the one focused on testing devices on remotes from different manufacturers, therefore testing different types of protocols. We took seven remotes, which we tested with several types of settings. A comparison of the results helped us find our optimal settings. Timing 0,8ms, and acceptance of 4 mistakes and comparison throughout the whole array was found to be the best solution. While evaluating, we also tried to predict, which communication protocol is used and the results were quite interesting. The RC-5 protocol, according to many sources the most commonly used in Europe, came second. The most common in our research was the NEC protocol and in third place came SONY SIRC. Due to the fact that we tested only seven types of remotes, we cannot say for sure which one is really the most used, but the discovery, that the most used household protocol is NEC, was surely compelling.

On the enclosed CD you will find the text of my dissertation and source codes for the RC-5 and universal program.

SEZNAM POUŽITÉ LITERATURY

- [1] MANN, Burkhard. *C pro mikrokontroléry*. Praha : BEN - technická literatura, 2004. 280 s. ISBN 80-7300-077-6.
- [2] PINKER, Jiří. *Mikroprocesory a Mikropočítače*. Praha : BEN technická literatura, 2004. 220 s. ISBN 80-7300-110-1.
- [3] MATOUŠEK, David. *Číslicová technika*. Praha: BEN technická literatura, 2002. 208 s. ISBN 80-7300-025-3.
- [4] MARTÍNEK, Radislav. *Senzory v průmyslové praxi*. Praha: BEN technická literatura, 2004. 200 s. ISBN 80-7300-114-4.
- [5] NEVŘIVA, Pavel. *Analýza signálů a soustav*. Praha: BEN technická literatura, 2002. 670 s. ISBN 80-7300-004-0.
- [6] CATSOULIS, John. *Designing Embedded Hardware*. O'Reilly Media, 2005. 400 s. ISBN 978-0-596-00755-3.
- [7] ŠÁRA, Zdeněk. *Normy pro IR datový přenos* [Online]. 1999 Dostupný z WWW: http://hw.cz/docs/ir_prenos/ir_protokoly.html
- [8] *8-bit Microcontrollers* [Online]. Freescale Semiconductor, 2004 Dostupný z WWW: <http://www.freescale.com/webapp/sps/site/homepage.jsp?code=8BITMCU>
- [9] *Z historie dálkových ovladačů*. [Online] [Citace: 18. 4. 2011] Dostupný z WWW: <http://www.emerx.cz/z-historie-dalkovych-ovladacu.html>.
- [10] *Historie dálkového ovladače*. [Online] [Citace: 18. 4. 2011] Dostupný z WWW: <http://www.shopservis.cz/historie-dalkoveho-ovladace.html>.
- [11] *Jednočipový počítač*. [Online] [Citace: 19. 4. 2011] Dostupný z WWW: http://cs.wikipedia.org/wiki/Jednočipový_počítač.
- [12] *Digitální signálový procesor*. [Online] [Citace 19. 4. 2011] Dostupný z WWW: http://cs.wikipedia.org/wiki/Digitální_signálový_procesor.
- [13] MYSLÍK, Vladimír. *IrDA – Kompletní popis*. [Online] Dostupný z WWW: http://hw.cz/docs/irda/irda_uvod.html.

- [14] *Infra Red.* [Online] Dostupný z WWW:
<http://www.hifi-remote.com/infrared/index.shtml>.
- [15] *Přijem a dekódování telegramu infračervených dálkových ovladačů.* [Online]
Dostupný z WWW:
http://www.dhservis.cz/dalsi_1/stmivac_do_soubory/infra.htm#rc.
- [16] *SB-Project.* [Online] Dostupný z WWW:
<http://www.sbprojects.com/knowledge/ir/rc5.php>.
- [17] *Učíci se IR spínač.* [Online] Dostupný z WWW:
<http://zajacik.kvalitne.cz/Uirsp/uirsp.html>.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

IR	Infra Red – infračervený
RC	Remote Control – dálkové ovládání
DVD	Digital Video Disk – digitální video disk
HDD	Hard Disk Drive – pevný disk
IrDA	Infrared Data Association – název standardu
LED	Light Emitting Diode – Světelná dioda
PDA	Personal Digital Assistant – kapesní počítač
TV	Televize
ROM	Read Only Memory – paměť pouze pro čtení
RAM	Random Access Memory – paměť s náhodným zápisem
CISC	Complex Instruction Set Computer – počítač s úplnou instrukční sadou
RISC	Reduce Instruction Set Computer – počítač s redukovanou sadou instrukcí
CMOS	Complementary Metal Oxide Semiconductor – technologie výroby polovodičů
TTL	Transistor Transistor Logic – technologie tranzistorů
GND	Ground – uzemění
SOIC	Small Outline Integrated Circuit
USB	Universal Serial Bus
LCD	Liquid Crystal Display – Displej z tekutých krystalů
μPCs	Mikropočítač

SEZNAM OBRÁZKŮ

<i>Obrázek 1 - Teslův první dálkový ovladač zvaný „Teleautomatons“</i>	11
<i>Obrázek 2 - Ovladač Lazy Bones (vlevo) a Flashmatic (vpravo)</i>	12
<i>Obrázek 3 - Zenith Space Command</i>	12
<i>Obrázek 4 - Doby trvání pulsů</i>	14
<i>Obrázek 5 - Princip pulsní modulace</i>	15
<i>Obrázek 6 - Princip FSK modulace</i>	16
<i>Obrázek 7 - Princip bi – phase modulace</i>	16
<i>Obrázek 8 - Princip pulsně šířkové modulace</i>	17
<i>Obrázek 9 - ITT protokol – rozpoznávání logické úrovně</i>	19
<i>Obrázek 10 - ITT protokol – ukázka signálu</i>	19
<i>Obrázek 11 - JVC protokol – rozpoznávání logické úrovně</i>	20
<i>Obrázek 12 - JVC protokol – ukázka stále stisknutého tlačítka</i>	20
<i>Obrázek 13 - JVC protokol – ukázka signálu</i>	21
<i>Obrázek 14 - Nokia NRC17 – ukázka signálu</i>	21
<i>Obrázek 15 - Nokia NRC17 – opakování zprávy</i>	22
<i>Obrázek 16 - Protokol Sharp – rozpoznávání logické úrovně</i>	22
<i>Obrázek 17 - Sony SIRC protokol – rozpoznávání logické úrovně</i>	23
<i>Obrázek 18 - Sony SIRC protokol – ukázka signálu</i>	23
<i>Obrázek 19 - Protokol NEC – rozpoznávání logické úrovně</i>	24
<i>Obrázek 20 - Protokol NEC – První rámeček s mezerou 4,5ms</i>	25
<i>Obrázek 21 - Protokol NEC – Opakovací kód s mezerou 2,5ms</i>	25
<i>Obrázek 22 - Protokol NEC – ukázka kódu při stisknutém tlačítku</i>	25
<i>Obrázek 23 - RC-5 protokol – rozlišení logické úrovně</i>	26
<i>Obrázek 24 - RC-5 protokol – datový rámeček</i>	26
<i>Obrázek 25 - Jednočipové počítače</i>	28
<i>Obrázek 26 - Mikropočítač S08SH16 (SOIC 28)</i>	32
<i>Obrázek 27 - Popis částí mikropočítačů S08SH</i>	33
<i>Obrázek 28 - Pouzdro S08SH16</i>	35
<i>Obrázek 29 - IR přijímač SFH5110</i>	35
<i>Obrázek 30 - Blokový diagram SFH5110-36</i>	36
<i>Obrázek 31 - Stabilizace napětí</i>	37

<i>Obrázek 32 - Zapojení mikropočítače.....</i>	<i>37</i>
<i>Obrázek 33 - Zapojení programovacího portu</i>	<i>38</i>
<i>Obrázek 34 - Zapojení IR přijímače SFH 5110-36.....</i>	<i>38</i>
<i>Obrázek 35 - Reálné zapojení IR přijímače</i>	<i>39</i>
<i>Obrázek 36 - Ukázka některých vstupů a výstupů.....</i>	<i>39</i>
<i>Obrázek 37 - Návrh plošného spoje v programu Eagle 5.1.0.....</i>	<i>40</i>
<i>Obrázek 38 - Návrh desky v programu Eagle.....</i>	<i>40</i>
<i>Obrázek 39 - IR přijímač ze strany součástek</i>	<i>41</i>
<i>Obrázek 40 - IR přijímač ze strany spojů</i>	<i>41</i>
<i>Obrázek 41 - Zařízení pro propojení s počítačem</i>	<i>43</i>
<i>Obrázek 42 - Snímání RC-5</i>	<i>48</i>
<i>Obrázek 43 - Ovladač LGMKJ42519618</i>	<i>59</i>
<i>Obrázek 44 - Ovladač Optibox Alligator.....</i>	<i>59</i>
<i>Obrázek 45 - Ovladač VRC Panasonic.....</i>	<i>60</i>
<i>Obrázek 46 - Ovladač HiFi Philips</i>	<i>60</i>
<i>Obrázek 47 - Ovladač LG – 6711R1P070B.....</i>	<i>61</i>
<i>Obrázek 48 - Ovladač UET MC-7</i>	<i>62</i>
<i>Obrázek 49 - Ovladač HUMAX RM-106</i>	<i>62</i>

SEZNAM TABULEK

<i>Tabulka 1 – Adresy a příkazy RC-5 protokolu</i>	<i>27</i>
<i>Tabulka 2 – Sada funkcí S08SH16.....</i>	<i>34</i>
<i>Tabulka 3 – Nastavení vstupů a výstupů na portu C</i>	<i>44</i>
<i>Tabulka 4 – Nastavení prescaleru časovače.....</i>	<i>46</i>
<i>Tabulka 5 – Nastavení vstupní frekvence.....</i>	<i>46</i>
<i>Tabulka 6 – Nastavení registru TPMISC</i>	<i>47</i>
<i>Tabulka 7 – Porovnání nastavení zařízení</i>	<i>63</i>

SEZNAM PŘÍLOH

PŘÍLOHA 1: DOKUMENTAČNÍ CD

Na přiloženém cd se nachází digitální forma této práce, zdrojové kódy k programům.