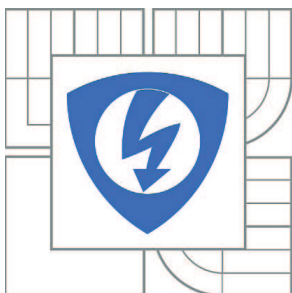


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

MODUL ŘÍZENÍ LED OSVĚTLENÍ

LED DRIVER

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

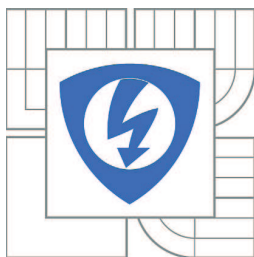
MICHAL HROMÁDKA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAROSLAV ŠEMBERA

BRNO 2011



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student: Michal Hromádka

ID: 120780

Ročník: 3

Akademický rok: 2010/2011

NÁZEV TÉMATU:

Modul řízení LED osvětlení

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce studenta je navrhnout a realizovat univerzální modul řízení několika světelných okruhů LED. Ovládání bude probíhat pomocí IR dálkového ovládání a sériového rozhraní. Modul musí být schopen "naučit se" příkazy různých dálkových ovladačů.

DOPORUČENÁ LITERATURA:

Hrbáček, J.: Komunikace mikrokontroléru s okolím 1, BEN, 2002, ISBN: 80-86056-42-2

Termín zadání: 7.2.2011

Termín odevzdání: 30.5.2011

Vedoucí práce: Ing. Jaroslav Šembera

prof. Ing. Pavel Jura, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Tato práce se zabývá návrhem a realizací modulu pro ovládání LED osvětlení. Modul je určen k ovládání jasu čtyř LED okruhů. Ovládání je řešeno pomocí IR dálkového ovladače nebo z PC přes USB rozhraní. Modul disponuje režimem „učení“, který umožní ovládání z jakéhokoliv ovladače, který využívá protokol RC5 nebo NEC.

Práce se dále zabývá popisem použitých IR protokolů, rozбором vybraných komponentů a vytvořeného softwaru pro mikrokontrolér ATmega16 a PC.

Klíčová slova

LED, USB, FT232, ATmega16, AVR, dálkový ovladač, IR přijímač, režim učení

Abstract

This thesis describes the design and implementation module for LED lighting. The module is designed to control brightness of four LED circuits. Control is done via IR remote control or from PC via USB interface. The module has a mode of "learning", which allows control from any controller that uses the RC5 or NEC protocol. The thesis also deals with the description of used IR protocols, analysis of selected components and software developed for the ATmega16 microcontroller and PC.

Keywords

LED, USB, FT232, ATmega16, AVR, remote control, IR reciever, learning mode

Bibliografická citace:

HROMÁDKA, M. *Modul řízení LED osvětlení*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 54s. Vedoucí bakalářské práce byl Ing. Jaroslav Šembera

Prohlášení

„Prohlašuji, že svou bakalářskou práci na téma Modul řízení LED osvětlení jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **27. května 2011**

.....
podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Jaroslavu Šemberovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne: **27. května 2011**

.....
podpis autora

OBSAH

1	Úvod	11
2	LED osvětlení	13
2.1	Parametry používané u osvětlení	13
2.2	Rozdělení osvětlení.....	15
3	Použité principy při návrhu	16
3.1	Pulzně šířková modulace – PWM.....	16
3.2	Komunikace po sériové lince.....	17
3.2.1	Synchronní přenos dat	17
3.2.2	Asynchronní přenos dat	17
3.3	Ovládání pomocí IR komunikace	18
3.3.1	Protokol RC-5	18
3.3.2	Protokol NEC.....	19
4	Vlastní návrh	21
4.1	Vlastní koncepce.....	21
4.2	Použité součástky.....	23
4.2.1	Mikrokontrolér (ATmega16)	23
4.2.2	Převodník USB - UART (FT232RL).....	23
4.2.3	IR demodulátor (TSOP4836).....	23
4.3	Výkonové tranzistory (IRLL2705)	24
4.4	Návrh DPS	24
5	Softwarová část	25
5.1	Řídící program pro MCU ATmega16.....	25
5.1.1	Paměť EEPROM.....	26
5.1.2	Linearizace jasu LED kanálů	27
5.1.3	Režim spánku.....	28
5.1.4	Softwarová pulzní šířková modulace - PWM.....	29
5.1.5	Dekódování IR signálu	31
5.1.6	Režim „učení“.....	34
5.1.7	Sériová komunikace po UART	35
5.1.8	Zavaděč (bootloader)	36

5.2	Ovládací program pro PC	37
6	Závěr	39
	Literatura	41
	Seznam symbolů, veličin a zkratk	42
	Seznam příloh	43

SEZNAM OBRÁZKŮ

Obr. 2.1 Teplota chromatičnosti [7].....	14
Obr. 3.1 Příklady průběhů PWM modulace [9].....	16
Obr. 3.2 Průběh synchronního přenosu dat.....	17
Obr. 3.3 Průběh asynchronního přenosu dat.....	18
Obr. 3.4 RC-5 protokol s příkladem datového rámce RC-5 [13]	19
Obr. 3.5 Dvoufázová modulace logických úrovní [10]	19
Obr. 3.6 Pulsně šířková modulace signálu IR [10]	20
Obr. 3.7 Datový rámec protokolu NEC [10]	20
Obr. 3.8 Opakovací impulzy protokolu NEC [10].....	20
Obr. 4.1 Blokové schéma modulu	22
Obr. 4.2 Vnitřní a doporučené zapojení IR demodulátoru TSOP4836 [11]	24
Obr. 5.1 Vývojový diagram hlavní části programu	26
Obr. 5.2 Graf závislosti vstupních a výstupních hodnot linearizace jasu LED	28
Obr. 5.3 Průběh generování PWM pro 2 kanály.....	31
Obr. 5.4 Vývojový diagram pro jeden PWM.....	31
Obr. 5.5 Vývojový diagram IR dekodování	33
Obr. 5.6 Vývojový diagram režimu učení	35
Obr. 5.7 Formát datové zprávy	35
Obr. 5.8 Vzhled programu MegaLoad V:7.0:.....	37
Obr. 5.9 Vzhled aplikace pro ovládání LED okruhů z PC	38

SEZNAM TABULEK

Tab. 1 Orientační srovnání výkonů světelných zdrojů podle světelného toku [16]	14
Tab. 2 Příklady teploty chromatičnosti [7]	14
Tab. 3 Použité porty a piny	22
Tab. 4 Mapa paměti EEPROM	27
Tab. 5 Naměřené hodnoty odběru modulu	29
Tab. 6 Pořadí tlačítek při učení	34

1 ÚVOD

Žijeme v době, kdy se pokrok nedá zastavit a téměř každý den vznikají nové a nové technologie, pomocí kterých se snažíme náš život zpříjemnit nebo zkrátit. Tato doba si však žádá čím dál více energie, kterou ovšem nemůžeme z našich zdrojů čerpat neomezeně. A proto se hledají možnosti, jak čerpat energii z jiných zdrojů, které jsou obnovitelné, a mohli bychom je tedy čerpat neomezeně. Kromě větrných, geotermálních, přílivových a jiných elektráren je to především asi nejrychleji se rozvíjející energie solární, která by se měla stát energetickým zdrojem budoucnosti.

Ovšem, můžeme se vydat i opačným směrem. A to, že se nebudeme snažit vyrobit více elektrické energie, ale budeme se snažit jí méně spotřebovat. To má pro nás příznivý vliv, jak z ekologické stránky, tak i z finanční. A proto se v dnešní době častěji hledají technologie, které snižují spotřebu elektrické energie.

Mezi jednu z nejvýznamnějších technologií patří i technologie světlo vyzařujících diod tzv. LED diod. Ty dnes pomalu začínají nahrazovat jak klasické, tak i úsporné žárovky či zářivky nebo CFL trubice v LCD obrazovkách. Tato technologie tedy rychle přestává být jen módním trendem, ale spíše ekonomickým. To je dáno především množstvím výhod, kterými disponuje.

Výhody LED osvětlení:

- dlouhá životnost až 100 000 hodin
- nízké požadavky na napětí a proud
- možnost častého spínání či regulace jasu
- vysoká energetická účinnost
- žádné IR nebo UV záření
- neobsahuje rtuť ani jiné těžké prvky
- malé rozměry
- odolnost vůči nárazům
- nízká provozní teplota
- rychlý start

Největší nevýhodou LED technologie je pořizovací cena, což v současnosti hraje nejspíše hlavní roli v tom, proč není tolik používána. Spousta lidí ji zatím kvůli ceně odsuzuje, ovšem s přibývajícími požadavky na ekologii bude tento krok nevyhnutelný. Tato technologie už postupně pronikla například do automobilového průmyslu a bude nás obklopotvat čím dál více. Má tedy ty nejlepší předpoklady stát se světelným zdrojem budoucnosti.

Tato práce se zabývá návrhem a řešením univerzálního modulu, který slouží k ovládání čtyř světelných okruhů výkonových LED. Samotné ovládání okruhů by mělo být řešeno pomocí IR dálkového ovladače a PC. Modul by měl také disponovat režimem „učení“, ve kterém je schopen se naučit příkazy jakéhokoliv dálkového ovladače s protokoly NEC, RC5 a kompatibilními. Komunikace mezi PC a modulem by měla být zajištěna pomocí sériového rozhraní. Napájecí napětí modulu by se mělo pohybovat v rozsahu 12-24V. Dále by měl být vytvořena aplikace, která by byla schopna ovládat modul z PC a software pro mikrokontrolér, který bude celý modul řídit. U modulu se předpokládá větší využití ovládání pomocí dálkového ovládání, a proto bude kladen větší důraz na jeho zpracování.

2 LED OSVĚTLENÍ

2.1 Parametry používané u osvětlení

Zdroj [14].

LED technologie se stala v současnosti dostatečně výkonným světelným zdrojem, aby plnohodnotně nahradila běžné světelné zdroje, jako je žárovka či zářivka. Při výběru vhodného LED světla, ať už se jedná o ekvivalent v podobě žárovky, zářivky, výbojky nebo jiný, musíme rozlišit několik základních parametrů:

Svítivost

Ta se dá obecně charakterizovat jako hustota světelných paprsků. Svítivost se zpravidla uvádí jako jedna hodnota, která platí pro určitý vyzařovaný úhel např. 45° nebo v podobě grafu ve kterém jsou zaznamenány hodnoty svítivosti ve všech úhlech.

Jednotka: cd (candela)

Světelný tok

Světelný tok nebo taky světelný výkon vyjadřuje celkový světelný výkon vyzářený do všech směrů. Tato jednotka je pro posouzení celkového výkonu nejvhodnější a také nejsrozumitelnější z hlediska porovnávání výkonu z hlediska klasické vláknové žárovky a LED světla. Příklad orientačního porovnání různých světelných zdrojů při zhruba stejném světelném toku je v tabulce č.1.

Jednotka: lm (lumen)

Osvětlení

Jedná se o celkový světelný výkon, který dopadne na jednotku plochy. Pokud je tedy třeba osvětlit konkrétní místo jako je například pracovní stůl, je vhodné porovnávat jednotlivé zdroje světla podle hodnoty osvětlení, které jsou schopné poskytnout na dané ploše.

Jednotka: lux

Světelná účinnost

Udává hodnotu celkového světelného výkonu na 1Watt spotřebované energie. Pomocí této hodnoty můžeme velice rychle porovnat energetická účinnost různých typů LED zdrojů. U běžných vláknových žárovek je hodnota světelné účinnosti v rozmezí 10-18 lm/W avšak u LED světél je tato hodnota několikrát vyšší zhruba 50-150 lm/W.

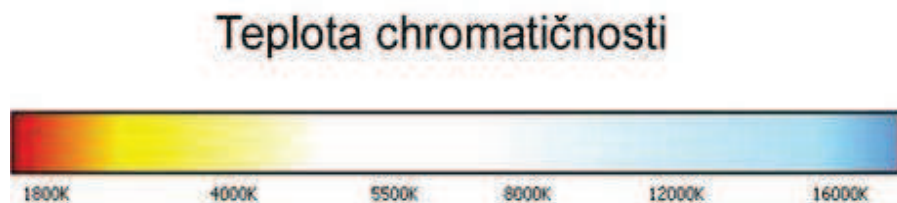
Jednotka: lm/W

Barevná teplota

Barevná teplota neboli teplota chromatičnosti je důležitá z hlediska vnímání kvality

světla člověkem. Tato hodnota je odvozena od barvy černého tělesa zahřátého na tuto teplotu. Výhodou LED světel je, že během změny jasu je teplota chromatičnosti konstantní, zatímco u klasických žárovek, pokud jas snižujeme, teplota chromatičnosti prudce klesá a může být žlutá až červená. Příklady teploty chromatičnosti jsou uvedeny v tabulce č. 1.

Jednotka: K [Kelvin]



Obr. 2.1 Teplota chromatičnosti [7]

Světelný tok (lm)	90	220	430	730	960	1380
klasická žárovka (W)	15	25	40	60	75	100
halogenová žárovka (W)	-	18	28	42	51	70
úsporná žárovka (W)	4-5	5-7	8-10	14-15	18	23
LED žárovka (W)	2-3	3-4	4-5	5-6	7-9	12-14

Tab. 1 Orientační srovnání výkonů světelných zdrojů podle světelného toku [16]

Teplota chromatičnosti[K]	příklad
1200	plamen svíčky
2800	žárovka, slunce při východu a západu
3000	studiové osvětlení
3200	halogenová žárovka
5000	obvyklé denní světlo, zářivky
5500	výbojky, fotografické blesky
6000	jasné polední světlo
6500	standardizované denní světlo
7000	lehce zamračená obloha
8000	oblačno, mlhavo (mraky zabarvují světlo do modra)
9300	analogová TV obrazovka
10 000	silně zamračená obloha nebo jen modré nebe bez Slunce

Tab. 2 Příklady teploty chromatičnosti [7]

2.2 Rozdělení osvětlení

Zdroj [7].

Pokud vybíráme LED osvětlení, musíme zohlednit, k čemu bude sloužit. Podle využití můžeme osvětlení rozdělit na:

Hlavní osvětlení

V případě hlavního osvětlení jde o adekvátní náhradu světelných zdrojů, kdy poskytnou světlo podobné či stejné jako dosavadní zdroje, avšak s podstatně nižší spotřebou (viz tabulka č.1). V případě náhrady LED technologií však nesmíme zohledňovat pouze výkon osvětlení, ale stejně důležitým parametrem je i jeho barevná teplota – teplota chromatičnosti.

Pro hlavní osvětlení se používají především dvě barvy světla: teplá bílá a studená bílá.

Teplá bílá je velice podobná klasickým žárovkám a její teplota chromatičnosti je v rozsahu 2800-3500K. Nazývá se teplou, a proto by měla být umístěna do místností vyhrazených a odpočinku a relaxaci, neboť žlutá barva dává tělu impuls přejít právě do režimu odpočinku. Podobným efektem se vyznačuje například i oheň v krbu. Tato barva je lidmi hojně využívána, aniž by tušili, že lidské oko se v prostředí, kde se tato barva vyskytuje, kazí 7 krát rychleji než venku na denním světle. Při čtení a psaní v tomto světle násobek ještě stoupá.

Naopak studená bílá s barevností okolo 6000K se blíží dennímu světlu pod mrakem, které eliminuje únavu očí a oči se v ní kazí podstatně méně. Je tedy vhodná především pro čtení, psaní a práci vůbec.

Vedlejší osvětlení - orientační

Co se týká vedlejšího osvětlení, u něj je kladen důraz na co nejnižší spotřebu, jelikož se nejčastěji používá jako orientační bod ve tmě nebo pro nasvícení schodišť. Výkon LED diod použitých pro toto osvětlení se tedy pohybuje v rozmezí 0,4-1,5W.

Dekoratивní osvětlení

Dekoratивní osvětlení je především záležitostí designu a celkového efektu jakým má působit a zaujmout. V tomto případě se většinou používají jednobarevné nebo tříbarevné RGB LED, které po připojení na řídicí jednotku mohou vytvořit jakoukoli barvu.

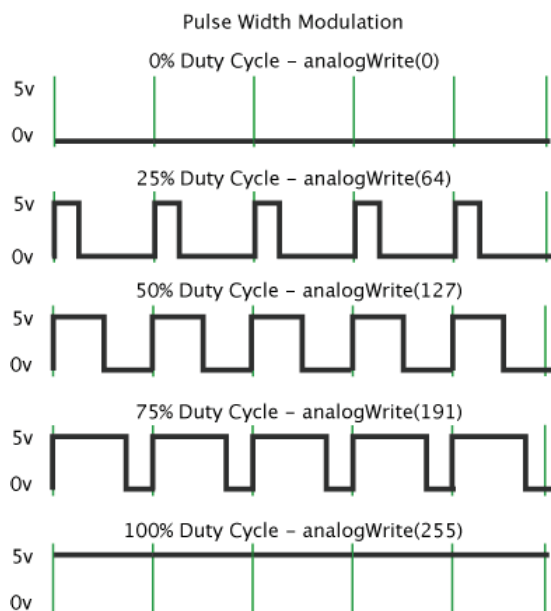
3 POUŽITÉ PRINCIPY PŘI NÁVRHU

V této kapitole jsou shrnuty teoretické poznatky použité při návrhu ovládacího modulu.

3.1 Pulzně šířková modulace – PWM

Pulzně šířková modulace je diskretní modulace, jejíž princip je založen na změně střidy binárního (dvouhodnotového) signálu (viz obr. 3.1). K demodulaci takového signálu pak postačí pouze filtr typu dolní propust. Tato modulace se využívá především ve výkonové elektronice k řízení napětím či proudem. Výhodou je její jednoduchost, modulace a demodulace, přenos signálu s nízkou náchylností na okolní vlivy či podpora téměř ve všech dnes dostupných mikrokontrolérech.

Jde tedy o signál s konstantní periodou T , ve kterém se mění pouze střída napětí (tj. poměr délky impulsu k délce mezery uvažovaný v jedné periodě). Tato hodnota se často uvádí jako poměr (1:1, 2:1, 1:5 atd.) nebo jako procentuální (100%, 50%, 0.1% atd.), kde 100% představuje poměr 1:0, 50% poměr 1:1 atd. Poměr délky impulsu k délce mezery bývá v zahraniční literatuře nazýván Duty Cycle.



Obr. 3.1 Příklady průběhů PWM modulace [9]

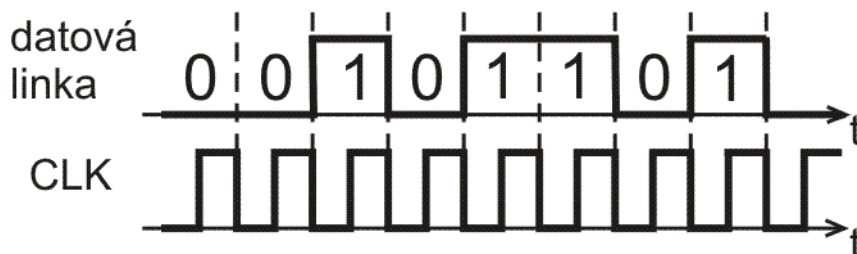
3.2 Komunikace po sériové lince

Jako sériový přenos dat považujeme takový, u kterého data běží sekvenčně tj. za sebou. Sériová komunikace se začala používat především tam, kde bylo zapotřebí odeslat větší objem dat na velkou vzdálenost. Její největší nevýhodou byl nízký datový tok. S příchodem nových technologií se datový tok zrychluje a sériová komunikace se používá i na malé vzdálenosti (např. S-ATA u PC). Je to především proto, že u paralelní komunikace se musí ošetřit problémy s přeslechy mezi jednotlivými linkami, dále se synchronizací hodinového signálu mezi vodiči a taky parazitní kapacity.

Pokud chceme, aby přijímač správně vyhodnotil data zasláné vysílačem, musí být spolu nějakým způsobem časově synchronizovány. Potom můžeme rozdělit sériovou komunikaci na synchronní a asynchronní.

3.2.1 Synchronní přenos dat

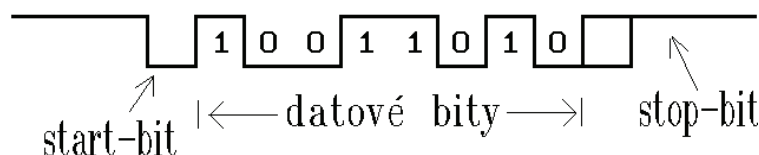
U tohoto typu přenosu jsou datové linky podporovány hodinovým signálem (CLK), který je vedený zvlášť nebo ukryt v datové lince a udává takt přenosu (viz obr. 3.2). Nevýhodou tohoto přenosu je, že se musí stále udržovat synchronizace mezi přijímačem a vysílačem. Výhodou přenosu je vyšší rychlost oproti asynchronnímu režimu, jelikož přenos není zatížen režii start a stop bitů. Proto se tento typ přenosu využívá spíše pro bitově orientované pakety a větší objemy dat.



Obr. 3.2 Průběh synchronního přenosu dat

3.2.2 Asynchronní přenos dat

Tento typ přenosu dat postrádá hodinový signál. Ten je však nahrazen režii start bitů a stop bitů, které se přenášejí spolu s daty (viz obr. 3.3). To způsobí, že se vysílač a přijímač vzájemně synchronizují jen fázově, než se začnou vysílat data (pomocí start bitu). Kromě start a stop bitu, může být k datům přidán paritní bit, který zvyšuje bezpečnost přenosu. Takový přenos je sice pomalejší, ale zato spolu mohou spolupracovat jakákoliv zařízení využívající stejný přenosový protokol. K tomu, aby došlo k fungujícímu spojení, musí mít spojená zařízení stejnou režii.



Obr. 3.3 Průběh asynchronního přenosu dat

3.3 Ovládání pomocí IR komunikace

Komunikace pomocí infračerveného světla (IR) je založena na principu generování signálu pomocí infračerveného záření, jinak zvaného tepelného. Toto záření je generováno infračervenou LED diodou o vlnové délce kolem 950nm, která je součástí dálkového ovladače. Na druhé straně je dekodér s PIN diodou, který signál dekóduje a pošle dál (např. do MCU). Jelikož je posílání informace z ovladače do dekodéru založeno na rozsvěcování a zhasínání LED diody, je velká pravděpodobnost příchodu rušivého signálu ze světelných zdrojů, jako jsou lampičky, slunce atd. Rušení se částečně odstraňuje za pomoci tmavých pouzder přijímacích IR dekodérů (nebo červenou plastovou clonou před ním) a dále pak modulací vysílaného signálu.

Při modulování se LED budí obdélníkovým nosným signálem o dané frekvenci, nejběžněji v rozsahu od 30 do 50kHz. Modulací, které se k tomu používají je několik typů (pulzní, pulzní šířková, FSK, dvoufázová). K tomu, aby měla modulace smysl, musíme jí nastavit určitou režii, což se provádí pomocí protokolů. Mezi nejčastěji používané protokoly u dálkových ovladačů spotřební elektroniky patří RC-5, RC-6, NEC, SONY SIRC. Nyní si dva nejdůležitější popíšeme.

3.3.1 Protokol RC-5

Protokol RC-5 je speciální protokol vytvořený pro dálkové ovladače a patří mezi nejrozšířenější ve spotřební elektronice. Datový rámec protokolu obsahuje 14bitů: 2 start bity, toggle bit, 5 adresových a 6 datových bitů.

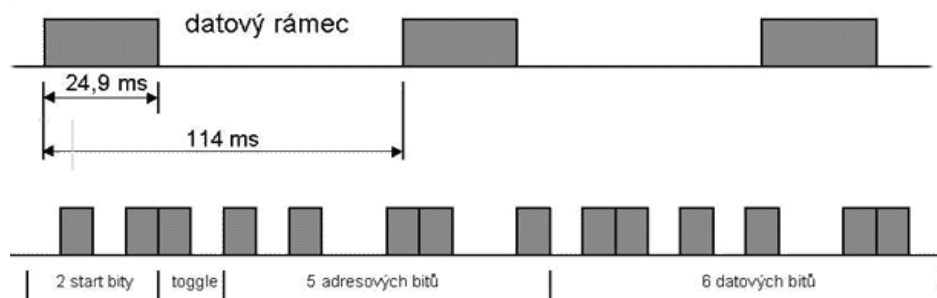
Start bity - slouží k synchronizaci datového rámce.

Toggle bit – nese v sobě informaci o stavu tlačítka. V případě, že tlačítko na DO držíme, hodnota toggle bitu se nemění. Pokud tlačítko pustíme a opět zmáčkeme, jeho hodnota se neguje.

Adresové bity – obsahují adresu cílového zařízení, aby se zabránilo ovládání jiných zařízení.

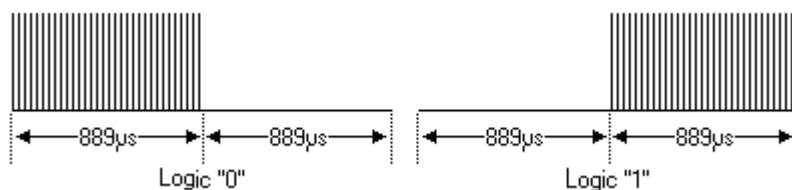
Datové bity – hodnota stisknutého tlačítka.

Celková délka datového rámce je 114ms, z toho 24,9ms obsahuje data a zbytek tvoří mezera (viz. obr. 3.4). Pokud je tlačítko na ovladači po odeslání prvního datového rámce stále aktivní, následuje odeslání dalšího rámce, který je stejný jako předchozí s výjimkou toggle bitu.



Obr. 3.4 RC-5 protokol s příkladem datového rámce RC-5 [13]

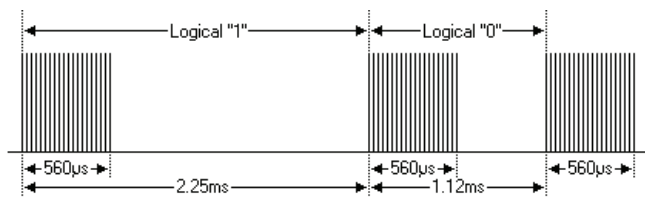
K modulování se využívá dvoufázová (bi-phase) modulace na nosné frekvenci $f=36\text{kHz}$. Její princip spočívá v rozdělení signálu na časové úseky – okna (viz obr. 3.5). Ty mají konstantní časovou délku a bit se vyhodnocuje vzhledem ke změně úrovně signálu uvnitř okna. Pokud se signál uvnitř okna změní z log.0 na log.1, je logikou vyhodnocen jako 1, při přechodu z log.1 na log.0 jako 0. Každý vyslaný bit obsahuje 32 impulzů o celkové délce $889\mu\text{s}$.



Obr. 3.5 Dvoufázová modulace logických úrovní [10]

3.3.2 Protokol NEC

Protokol NEC je stejně jako předchozí protokol určen výhradně pro dálkové ovladače. Signál je také přenášen na modulovací frekvenci $f=36\text{kHz}$, ale ke kódování se používá pulsně šířková modulace – PWM (viz obr. 3.6). Výhodou tohoto protokolu je nízká chybovost. Toho bylo dosaženo pomocí konstantního datového rámce s PWM modulací. Toto samotné spojení by však konstantní datový rámec nemohlo tvořit. Proto se data i adresa nejdříve vysílají v daném tvaru a poté ve tvaru negovaném. To způsobí, že se tyto dva datové úseky navzájem spojí v jeden konstantní datový rámec.

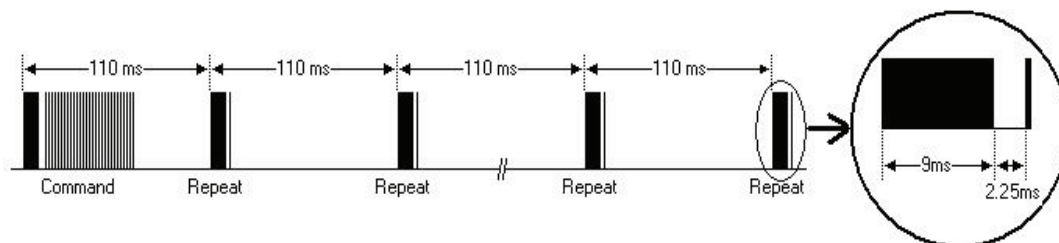


Obr. 3.6 Pulsně šířková modulace signálu IR [10]

Protokol NEC obsahuje zaváděcí kód, 2x8 adresových bitů a 2x8 datových bitů. Délka zaváděcího kódu je 9ms v log.1, poté následuje mezera o délce 4,5ms. Po zaváděcím kódu následuje úsek s adresou v normálním a negovaném tvaru a úsek s daty taktéž v normálním a negovaném tvaru (viz obr. 3.7). Celý datový rámec i s mezerou čítá asi 110ms. V této podobě je při stisku tlačítka vyslán pouze jednou. Poté se v případě držení tlačítka vysílá opakovací impuls (viz obr. 3.8), který se skládá z impulsu o délce 9ms, mezery o délce 2,25ms a krátkého pulzu. Opakování těchto pulzů probíhá opět po 110ms.



Obr. 3.7 Datový rámec protokolu NEC [10]



Obr. 3.8 Opakovací impulzy protokolu NEC [10]

4 VLASTNÍ NÁVRH

Kapitola 3 popisuje postupy použité při vlastním návrhu. Při návrhu se dbalo především na splnění požadavků jako je schopnost dekodovat IR signál, komunikovat s PC pomocí sériové linky. Hotový modul řízení bude testován na čtyřech párech LED modulů s napájecím napětím 12V. LED moduly budou zapojeny sériově tak, aby řídicí modul byl napájen napětím 24V. Testováno bude 6 modulů, každý s příkonem 0,24W a 2 s příkonem 2,4W.

4.1 Vlastní koncepce

Celý modul byl nejprve navržen a částečně odladěn na vývojovém kitu EvB 4.1 fy AND-TECH. Poté byl realizován tak, aby splňoval dané požadavky. Modul je navržen tak, aby byl schopen komunikovat s PC přes sběrnici USB a přijímal příkazy z dálkového ovladače. Je možno jej napájet stejnosměrným napětím 7-30V. Napájecí napětí je zároveň určeno k napájení všech LED kanálů (okruhů). Maximální a minimální hodnoty napájení jsou dány vlastnostmi 5V stabilizátoru, který napájí řídicí část obvodu.

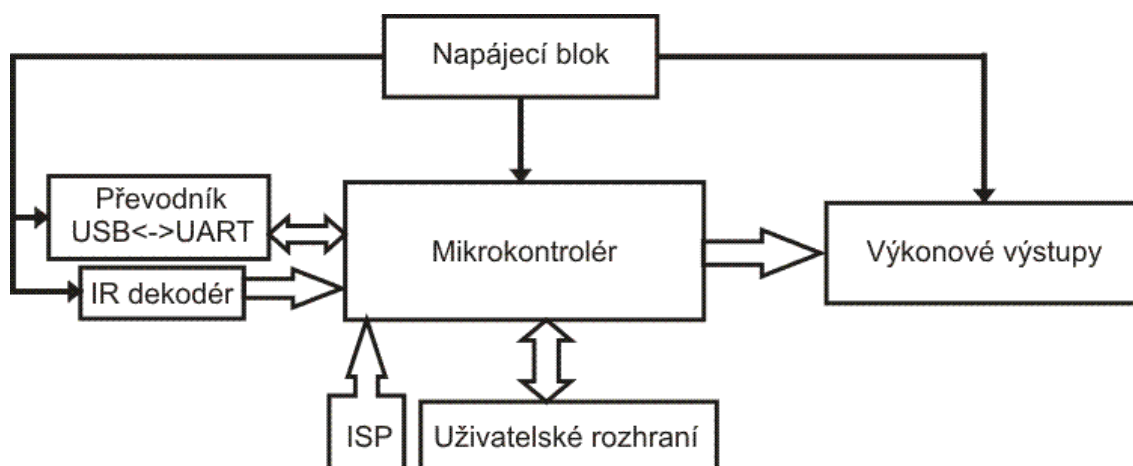
Jelikož MCU ATmega16 použitý k řízení modulu obsahuje pouze 4 hardwarově podporované PWM kanály z nichž 2 jsou znehodnoceny pro běh programu, musely být nahrazeny softwarovou formou PWM. Ukázalo se, že postačující počet kanálů je 4. Každý z kanálů je řízen výkonovým SMD tranzistorem typu MOSFET a výstupy jsou přivedeny na společnou násuvnou šroubovací svorkovnici. Maximální proud jedním kanálem je cca 3,2A při buzení tranzistoru maximálním napětím 5V.

K univerzálnosti modulu přispěly také uživatelské vstupy a výstupy, které jsou vyvedeny na konektorech JUM2 a JUM1 (viz. příloha A.1). Tyto vstupy a výstupy jsou určeny pro připojení tlačítek a indikačních LED. Uživatelské rozhraní je nyní tvořeno čtyřmi LED diodami a dvěma tlačítky. LED diody na předním panelu slouží k informování uživatele o aktuálním stavu modulu. Zelená LED signalizuje běh programu. Pokud program běží, dioda svítí. Po přepnutí do režimu spánku zhasne. Žlutá LED označená jako „IR“ signalizuje přijetí IR datového rámce a svítí po dobu jeho dekodování. Červená LED označená jako „LRN“ svítí, pokud je aktivní režim učení, po jeho ukončení zhasne. Červená LED „USB“ svítí po propojení s PC. Tato LED je nezávislá na MCU, protože je přímo napájena z PC. Červené tlačítko umístěné na předním panelu slouží ke spuštění režimu učení. Černé tlačítko „Reset“ umístěné na zadní straně modulu slouží k resetování programu v modulu nebo jako spouštěcí signál pro programování pomocí zavaděče. Toto tlačítko je ale připojeno na piny konektoru JP1 (zem a reset), který je určen pro programování pomocí ISP rozhraní. Pomocí tohoto

rozhraní může být program v MCU změněn přímo za běhu modulu, bez nutnosti vyjmutí procesoru.

Konektory JUM3 a JUM4 jsou určeny k napájení uživatelských vstupů a výstupů. A konektor JUM5 je určen pro připojení externího IR přijímače (demodulátoru), který je umístěn na kabelu.

Použití konkrétních portů a pinů MCU ATmega 16 je uvedeno v tabulce č.3.



Obr. 4.1 Blokové schéma modulu

Port	Použití	Port	Použití	Port	Použití
PA0	LED LRN	PB2	PWM kanál 2	PD0	TXD (UART)
PA1	LED IR	PB1	PWM kanál 3	PD1	RXD (UART)
PA2	LED PWR	PB0	PWM kanál 4	PD3	ext. přerušení
PB3	PWM kanál 1	PC1	Tlačítko "učení"		

Tab. 3 Použité porty a piny

Parametry modulu:

Napájecí napětí:	7 - 30V
Max. proud kanálem:	3,2 A
Počet LED kanálů:	4
Spotřeba v režimu spánku (bez USB):	0,4 W
Spotřeba v režimu spánku (s USB):	0,6 W
Podporované protokoly DO:	RC5, NEC
Počet programovatelných tlačítek:	14
Připojení k PC:	USB 1.1 a kompatibilní
Rozměry:	130x100x30mm

4.2 Použité součástky

Hlavní části modulu tvoří: mikrokontrolér, převodník USB – UART, IR demodulátor a výkonové tranzistory. Následující podkapitoly popisují důvody výběru jednotlivých komponentů.

4.2.1 Mikrokontrolér (ATmega16)

Jedná se o 8-mi bitový mikrokontrolér firmy Atmel používající architekturu RISC. Je zde použit jako hlavní prvek celého zařízení. Slouží ke komunikaci s PC pomocí sériové linky (přes FT232RL), ke komunikaci s IR dekodérem a dále pak ke generování PWM signálu, potřebného k řízení jasu LED okruhů. Důvodů proč vybrat tento mikrokontrolér, bylo hned několik.

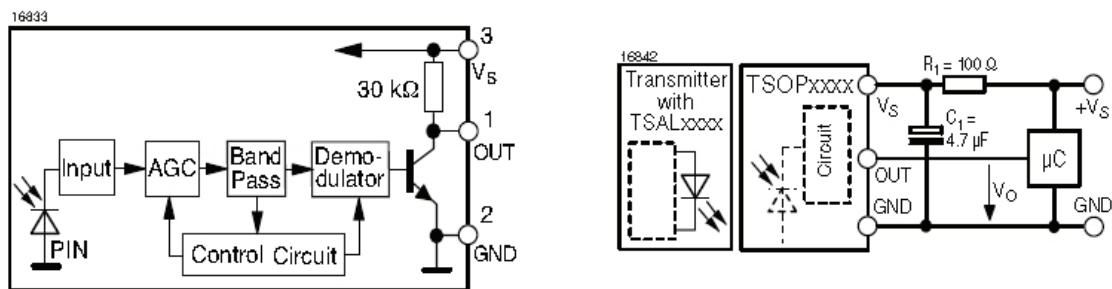
Především to byla dobrá znalost mikrokontroléru a možnost vývoje aplikací na vývojovém kitu EvB 4.1. Jako programovací jazyky lze zvolit jazyk symbolických adres nebo dnes asi nejpoužívanější jazyk C. Co se týká samotného mikrokontroléru, jeho parametry jsou postačující. Velikost flash paměti 16kB, vysoká taktovací frekvence 16MHz, množství vstupů/výstupů, velké pouzdro, spousta periférií. Snad jen drobnou nevýhodou je, na dnešní dobu nepříznivý poměr cena/výkon a malý počet PWM kanálů.

4.2.2 Převodník USB - UART (FT232RL)

Jelikož se sériové porty v dnešních počítačích vyskytují už jen zřídka, bylo zapotřebí najít vhodnou alternativu. Jednou z nejpoužívanějších a nejpropracovanějších je integrovaný obvod fy FTDI s označením FT232RL. Jde o obousměrný převodník USB - UART. V dnešní době, kdy má téměř každý počítač či notebook port USB, se tato variant ukázala jako nejvhodnější. Pro plnou podporu tohoto zařízení musíme do PC nainstalovat VCP (Virtual COM port) ovladače, které vytvoří virtuální COM port a po propojení modulu s PC budou spolu komunikovat, jako by byli propojeni přes dnes již ustupující COM port.

4.2.3 IR demodulátor (TSOP4836)

K tomu, abychom mohli IR signál z dálkového ovladače přijímat je zapotřebí použít přijímací diodu. Ta by sama o sobě generovala modulovaný signál na nosné frekvenci, a proto by jej bylo zapotřebí dále demodulovat a filtrovat. Jednodušší variantou je použití součástky, která těmito funkcemi disponuje. Jedná se o IR demodulátor typu TSOP4836 fy Vishay, což je demodulátor s digitálním výstupem pracující se signálem s nosnou frekvencí 36KHz. Tato frekvence je u dálkových ovladačů nejběžnější. Demodulátor je určen pro napájecí napětí 2.7 - 5.5V a podporuje všechny potřebné protokoly, jako jsou RC5/6, NEC, SONY, Sharp atd. Je tedy vhodný pro použití do modulu.



Obr. 4.2 Vnitřní a doporučené zapojení IR demodulátoru TSOP4836 [11]

4.3 Výkonové tranzistory (IRLL2705)

Samotný výstup MCU by k vybuzení výkonových LED diod nestačil a proto je zapotřebí použít nějaký výkonový prvek. Pokud řídíme jas pomocí PWM modulace je vhodné využít vlastností tranzistorů MOSFET. Jejich výhoda oproti bipolárním tranzistorům spočívá především v nižší spotřebě. Požadavky na tranzistor byly vyšší napětí než 30V, proud kolem 2A a co nejnižší odpor R_{ds} .

Jako nejvhodnější se ukázal tranzistor IRLL2705 typu HEXFET fy International Rectifier. Svými vlastnostmi splnil všechny požadavky. Jeho maximální napětí $V_{ds}=55V$, proud $I_d=3,8A$ a odpor $R_{ds}=0,051\Omega$. Nízká hodnota odporu R_{ds} je velmi důležitá, protože čím je menší, tím menší výkon se na tranzistoru ztrácí. Výhodou bylo i provedení v SMD pouzdře SOT-223.

4.4 Návrh DPS

DPS byla navrhována pomocí editoru plošných spojů Eagle 5.7.0. Kvůli minimalizaci desky byla většina součástek zvolena v pouzdru SMD. Napájení, USB konektor a výkonové výstupy vyvedené na šroubovací svorky jsou osazeny na jedné straně kvůli lepšímu přístupu po zabudování do krabice. Zbývající vstupy/výstupy a IR vstup jsou osazeny na protější straně, pro jednodušší vyvedení na přední panel. Po celé desce byla v závěru návrhu „rozlita“ měď - zem, díky níž by se mělo eliminovat působení vnějších rušivých signálů. Taktéž to usnadňuje leptání desky, jelikož se zmenší plocha mědi určená k odleptání. K uchycení desky do krabice slouží 4 otvory o velikosti 3mm v každém rohu.

5 SOFTWAREVÁ ČÁST

5.1 Řídící program pro MCU ATmega16

Řídící program pro mikrokontrolér ATmega16 byl napsán ve vývojovém prostředí AVR Studio 4.16 s podporou WinAVR v jazyce ANSI-C. Program byl nejprve odladěn na vývojovém kitu EvB 4.1 a následně implementován do samotného modulu. Je sestaven tak, aby zvládl obsluhu jak z počítače, tak i z dálkového ovladače.

Hlavní program (viz obr. 5.1) je rozdělen na 2 části: inicializační a cyklickou. V inicializační části se provede nastavení registrů portů, časovačů 0 a 2, jednotky UART a ext. přerušení následně:

Nastavení portů:

```
PORTA = 0x00;      // port A - 0 na výstupu (zhasnou led)
DDRA  = 0xFF;      // port A - Výstupy indikačních LED
PORTB = 0x00;      // port B - LED
DDRB  = 0xFF;      // port B -vystup výkonových LED Kanálu
PORTC = 0xFF;      // port C - 1 na výstupu (pro tlačítka)
DDRC  = 0x00;      // port C - Vstupy tlačítek
```

Nastavení INT1 - externí přerušení pro IR demodulátor na PD3:

```
MCUCR = 0x08;      // INT1 reaguje na sestupnou hranu
GICR  = 0x80;      // povolení přerušení od INT1
```

Nastavení TIMER0 a přerušení (Timer0 zajišťuje softwarové PWM):

```
TCNT0 = REG_TCNT0; // nastavení registru čítače
TIMSK |= (1<<TOIE0); // přerušení při přetečení
TCCR0 |= (1<<CS10); // normální režim, dělička 0
```

Nastavení TIMER2 a přerušení (Timer2 zajišťuje dekodování IR):

```
TCNT2 = 0;          //vynulování registru čítače
TIMSK |= (1<<TOIE2); //přerušení při přetečení
TCCR2 |= (1<<CS22) | (1<<CS21); //normální režim, dělička
/256 (1 tik za 16us, přetečení po 4,096ms)
```

Nastavení sériové linky USART pro komunikaci s PC:

```
UBRRH = (unsigned char) (MY_UBRR>>8); // Nastaví rychlost
komunikace - horní byte (9600Bd)
```

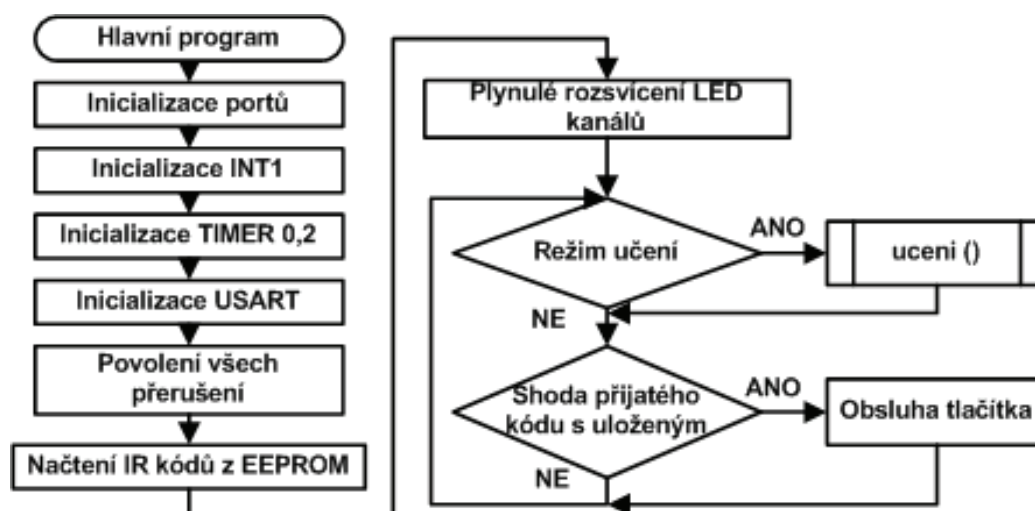
```

UBRR1 = (unsigned char) MY_UBRR;          // Nastaví rychlost
komunikace - dolní byte (9600Bd)
UCSRB = (1<<RXEN)|(1<<TXEN)|(1<<RXCIE); // Zapne vysílač a
přijímač, povolí přerušeni při příjmu a dokončení vysílání
UCSRC = (1<<URSEL)|(1<<USBS)|(1<<UCSZ0)|(1<<UCSZ1);

```

Po nastavení periferií se provede povolení všech přerušeni příkazem `sei()`, načtení hodnot uložených kódů dálkových ovladačů z paměti EEPROM do pole `pCode[14]` a plynulé rozsvícení LED kanálů na poslední uložené nastavení.

Program se poté dostane do cyklu, kde nejprve testuje stav tlačítka pro spuštění režimu učení. Pokud dojde ke stisku tlačítka, program skočí do funkce `uceni()` a po jejím ukončení se vrátí zpět do cyklu. Poté se kontroluje příchod signálu od dálkového ovladače. V případě, že je přijat signál a dekodován porovnává se přijatý kód v proměnné `code2` s kódy uloženými v poli `pCode[14]`. Po shodě přijatého kódu s uloženým se provede příslušná operace.



Obr. 5.1 Vývojový diagram hlavní části programu

Další části programu jsou popsány v následujících podkapitolách:

5.1.1 Paměť EEPROM

Je paměť, která je schopna uchovávat data i po odpojení napájecího napájení. Zde je tato paměť použita pro uložení kódů dálkového ovladače a jasu LED diod. Životnost paměti EEPROM v MCU ATmega16 je garantována na 100 000 cyklů. To znamená, že při 10-ti zápisech denně by paměť měla vydržet asi 27 let, což je více než postačující. V případě, že by ovšem tato hodnota nestačila, existují speciální algoritmy, které změnu

místa uložení v paměti a tím zajistí její delší životnost. Toho lze dosáhnout pouze, pokud paměť disponuje volnými místy.

Pro práci s pamětí je určena knihovna `eeeprom.h`. K ukládání a načítání hodnot z paměti jsou použity 4 funkce:

`eeeprom_write_byte()`, `eeeprom_read_byte()`, `eeeprom_write_word()`, `eeeprom_read_word()`.

Ukládání po slovech (word), která obsahují 2 bajty, je určeno pro kódy z dálkového ovladače uložené v poli `pCode[14]`. Hodnoty se ukládají v bajtové orientované paměti od adresy 0 až po adresu 27. Ukládání po bajtech je určeno pro hodnoty jasů. Zde je paměť rozdělena na 3 části. Adresy 30 až 33 obsahují nelinearizované hodnoty jasů `Kanal[4]` a adresy 35 až 38 linearizované hodnoty jasů `linKanal[4]` ukládané při přechodu do režimu spánku nebo učení. Tyto hodnoty se ukládají vždy všechny najednou a také se načítají po připojení napájení. Poslední část zabírají adresy 40 až 43. Jsou v nich ukládány linearizované hodnoty jasů `linKanal[4]` při vypnutí jednotlivých kanálů. To znamená, že se zde ukládá vždy jen jedna požadovaná hodnota.

Adresa paměti	obsah paměti	Adresa paměti	obsah paměti	Adresa paměti	obsah paměti
0	1. IR kód	30	Kanal [0]	39	volné
1		31	Kanal [1]	40	linKanal [0]
2	2. IR kód	32	Kanal [2]	41	linKanal [1]
3		33	Kanal [3]	42	linKanal [2]
4 až 25		34	volné	43	linKanal [3]
26	14. IR kód	35	linKanal [0]	44	volné
27		36	linKanal [1]		
28	volné	37	linKanal [2]		
29	volné	38	linKanal [3]		

Tab. 4 Mapa paměti EEPROM

5.1.2 Linearizace jasu LED kanálů

Linearizace je zde použita kvůli snížení počtu kroků nastavení jasu a lepšímu odstupňování jasu. Je to dáno především kvůli charakteristice LED diody, která není lineární. Samotná linearizace se skládá ze 4 křivek (obr. 5.2), které mají postupně stoupající strmost (viz obr. 5.2). Vstupem do funkce je hodnota v rozsahu 0 až 31 a výstupem je zlinearizovaná hodnota v rozmezí 0 až 255, která je dále zpracována PWM modulací. Pokud je na vstupu hodnota vyšší než 31, na výstupu je vždy hodnota 255.

Pro ovládání pomocí DO musela být tato hodnota snížena na 9 hodnot, aby bylo

dosaženo rychlejšího a efektivnějšího ovládání jasu. Snížení rozlišení bylo dosaženo skokem po 4 hodnotách, takže nyní jsou vstupní hodnoty 0, 4, 8, 12, 16, 20, 24, 28 a 30. V případě ovládání z PC bylo ponecháno rozlišení na rozsah 0 až 31. Linearizace je převzata a upravena z [12].

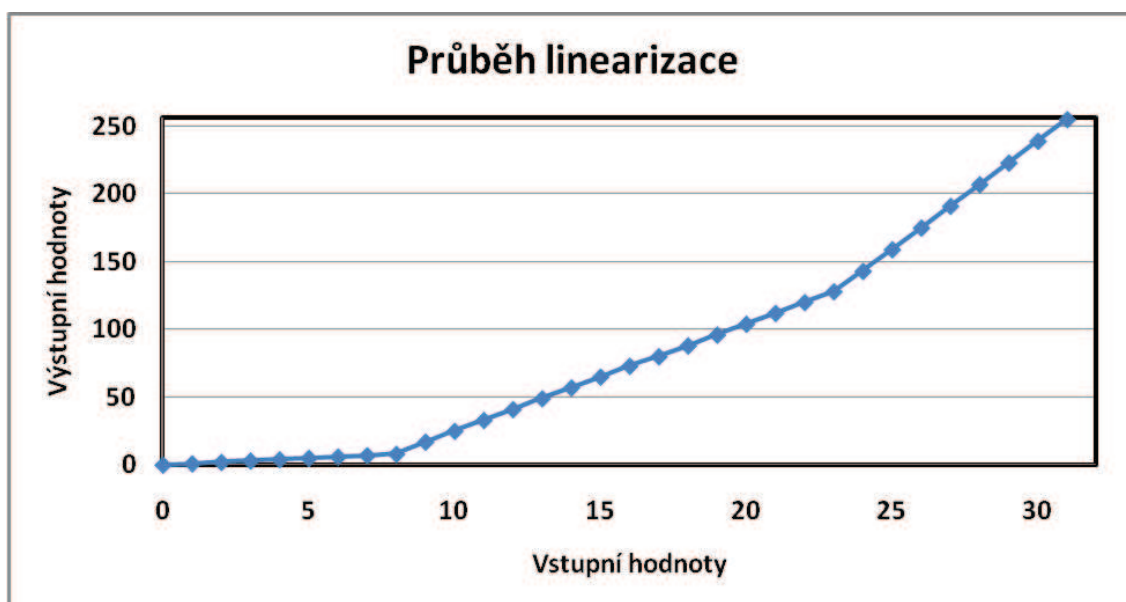
Linearizační křivky mají rovnice v následujícím tvaru:

$$y = x \quad \text{pro } D(f)=\langle 0,7 \rangle \text{ a } H(f)=\langle 0,7 \rangle$$

$$y = x * 8 - 55 \quad \text{pro } D(f)=\langle 8,15 \rangle \text{ a } H(f)=\langle 9,65 \rangle$$

$$y = x * 8 - 56 \quad \text{pro } D(f)=\langle 16,23 \rangle \text{ a } H(f)=\langle 72,128 \rangle$$

$$y = x * 16 - 241 \quad \text{pro } D(f)=\langle 24,31 \rangle \text{ a } H(f)=\langle 143,255 \rangle$$



Obr. 5.2 Graf závislosti vstupních a výstupních hodnot linearizace jasu LED

5.1.3 Režim spánku

Režim spánku, který se také nazývá režimem snížené spotřeby, se využívá především u bateriemi napájených zařízení nebo zařízení, která jsou stále připojena k síti. Režim spánku vede ke snížení spotřeby, čímž se například může prodloužit životnost baterií. Tato funkce však musí být podporována MCU. Jelikož se očekává, že modul bude přes adaptér stále připojen k síti, můžeme výhody této funkce využít.

MCU ATmega16 má 6 různých režimů spánku. V programu je použit režim nečinnosti (Idle mode), který se jevil jako nevhodnější. Při jeho aktivaci dochází k zastavení činnosti jádra procesoru, ale zároveň zůstávají aktivní čítače/časovače, jednotka UART a přerušovací systém. To je důležité pro zpětné probuzení, kde se tyto periferie používají.

Pro práci s tímto režimem je určena knihovna `sleep.h`. Před samotným uspaním procesoru se zhasne LED indikující běh programu, plynule pohasne všechny LED kanály, vypne se časovač č.0 generující PWM a nastaví se typ režimu spánku. Poté program přejde do cyklu, ve kterém se provede samotné uspaní jádra procesoru a program se zastaví. K probuzení dojde jen tehdy, je-li přijat správný kód z dálkového ovladače nebo dojde-li ke zmáčknutí tlačítka režimu „učení“ a zároveň je přijat jakýkoliv kód z ovladače. Po probuzení a přijetí správného kódu se procesor přepne do normálního režimu, program vyskočí z cyklu, zapne čítač č.0, plynule rozsvítí všechny LED kanály a zapne LED indikující běh programu. Pokud je ovšem přijatý kód špatný, program zůstane v cyklu a procesor se zpět uspí.

Abychom zjistili, zda-li je režim spánku opravdu funkční, bylo provedeno měření spotřeby modulu. Měření bylo prováděno pomocí ampérmetru a voltmetru připojenými mezi adaptér a modul. Měřilo se ve dvou stavech. Jednou s odpojenými LED diodami, aby měl modul co nejnižší odběr a podruhé po uvedení do režimu spánku. Jelikož bylo zjištěno, že odběr modulu se liší po připojení k PC, byly tyto hodnoty naměřeny ještě pro stavy s připojením a bez připojení USB. Výsledky měření jsou uvedeny v tabulce č.5. Z výsledků lze usoudit, že režim spánku má vliv na spotřebu modulu.

Stav modulu	U [V]	I [mA]	P [mW]
odpojené LED, bez USB	24,52	27,00	662
režim spánku, bez USB	24,52	15,93	391
odpojené LED, s USB	24,52	35,45	869
režim spánku, s USB	24,52	24,40	598

Tab. 5 Naměřené hodnoty odběru modulu

5.1.4 Softwarová pulzní šířková modulace - PWM

PWM modulace je zde použita pro změnu jasu LED kanálů. MCU ATmega16 obsahuje čtyři čítače/časovače a mezi jednu z jejich funkcí patří i generování signálu PWM. Obsluha takto hardwarově vytvořeného PWM signálu je pak velice snadná. Spočívá pouze ve změně hodnoty v jediném 8-mi bitovém registru, což má za následek změnu střídání na výstupním pinu čítače/časovače. Jeho nevýhodou je však pevně nastavený výstupní pin. U modulu je potřeba neustále generovat čtyři nezávislé kanály PWM a to znamená využití čtyř čítačů/časovačů. Bohužel jeden takový je potřebný i pro dekodování IR signálu a proto musíme zvolit jinou cestu.

Jiná cesta může například spočívat v softwarovém generování tohoto signálu. K tomu zde stačí použít jeden časovač, v našem případě časovač č.0, který udává základní periodu T_z . Po uplynutí každé této periody je vyvoláno přerušování, vzniklé přetečením tohoto časovače, a v něm se kontroluje nastavení střídání PWM signálu

jednotlivých kanálů. Pro vytvoření 8-bitového rozlišení modulace musí v jedné periodě signálu vzniknout těchto přerušení 256.

Základní perioda je dána nastavením časovače č.0. Zde je použit normální režim bez předděličky s povolením generovat přerušení při přetečení. Vypočítat ji můžeme následně:

$$T_z = \frac{1}{F_{cpu}} = \frac{1}{\frac{16000000}{256}} = 16 \cdot 10^{-6} s = 16 \mu s \quad (1)$$

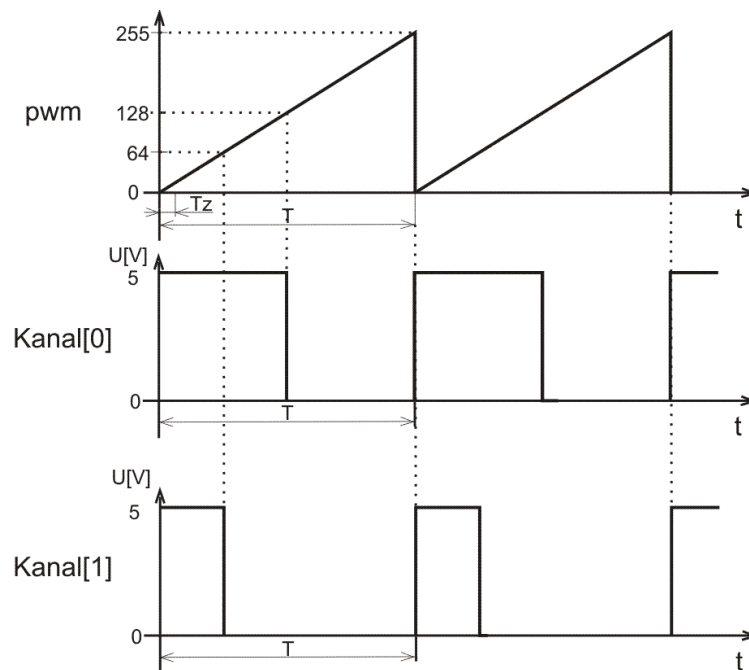
V každé obsluze přerušení je inkrementována proměnná pwm a dochází ke kontrole jednotlivých kanálů (viz obr. 5.4). Každý kanál má svoji 8mi bitovou proměnnou a ty jsou postupně porovnány s proměnnou pwm . Pokud $pwm=0$ nastaví se výstupy do log.1, to znamená začátek každé nové periody PWM signálu. Pokud dále dojde ke shodě proměnné kanálu a pwm , dojde k jeho nastavení do log.0. Po 256 krocích proměnná pwm přeteče, tím se vynuluje a vše se opakuje. Výslednou periodu T PWM signálu získáme se vzorce (2).

$$T = T_z * 2^8 = 16 \cdot 10^{-6} * 256 = 4,096 ms \quad (2)$$

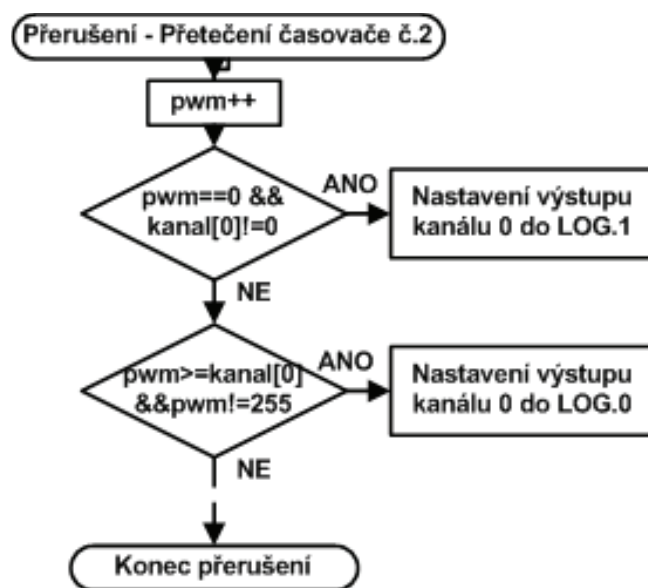
Z toho frekvence:

$$F_{pwm} = \frac{1}{T} = \frac{1}{4,096 * 10^{-3}} = 244,1 Hz \quad (3)$$

Hodnota 244Hz je postačující pro modulování a nedochází při ní k blikání LED diod. Tato hodnota byla ověřena měřením na osciloskopu.



Obr. 5.3 Průběh generování PWM pro 2 kanály



Obr. 5.4 Vývojový diagram pro jeden PWM

5.1.5 Dekódování IR signálu

Dekódování signálu od IR demodulátoru je nejdůležitější a nejsložitější částí celého programu. Slouží k přeměně signálu od demodulátoru na kód, který by měl být specifický pro každé tlačítko. Zde je požadováno, aby byl algoritmus funkční pro nejpoužívanější protokoly dálkových ovladačů RC-5 a NEC. Algoritmus by měl být co nejjednodušší, aby mikrokontrolér zbytečně nezatežoval, ale zároveň by měl být

efektivní a účinný. Jedna z možností dekodování tohoto signálu je rozpoznání protokolu podle startovací sekvence a jeho následné dekodování dle zvoleného algoritmu. V těchto případech má každý protokol svůj algoritmus, podle kterého je celý dekodován (včetně adresových bitů). Tento způsob je nejpřesnější, ale bohužel nejvíce náročný na výkon a paměť MCU.

První myšlenkou bylo vzorkovat příchozí signál po určitých periodách. Testování bylo prováděno s periodou kolem 1ms a 30-ti vzorky. Tento způsob byl funkční, ale bohužel jen u protokolů RC5, kde nedochází ke změně délky datového rámce. U protokolu jako je NEC, který využívá PWM modulaci, je jiná délka signálu pro log.1 a log.0 a proto by bylo nutné pro přesné načtení datového rámce zvýšit počet vzorků a zkrátit periodu vzorkování. To by si ovšem opět vyžadovalo vyšší výkon a větší paměť MCU.

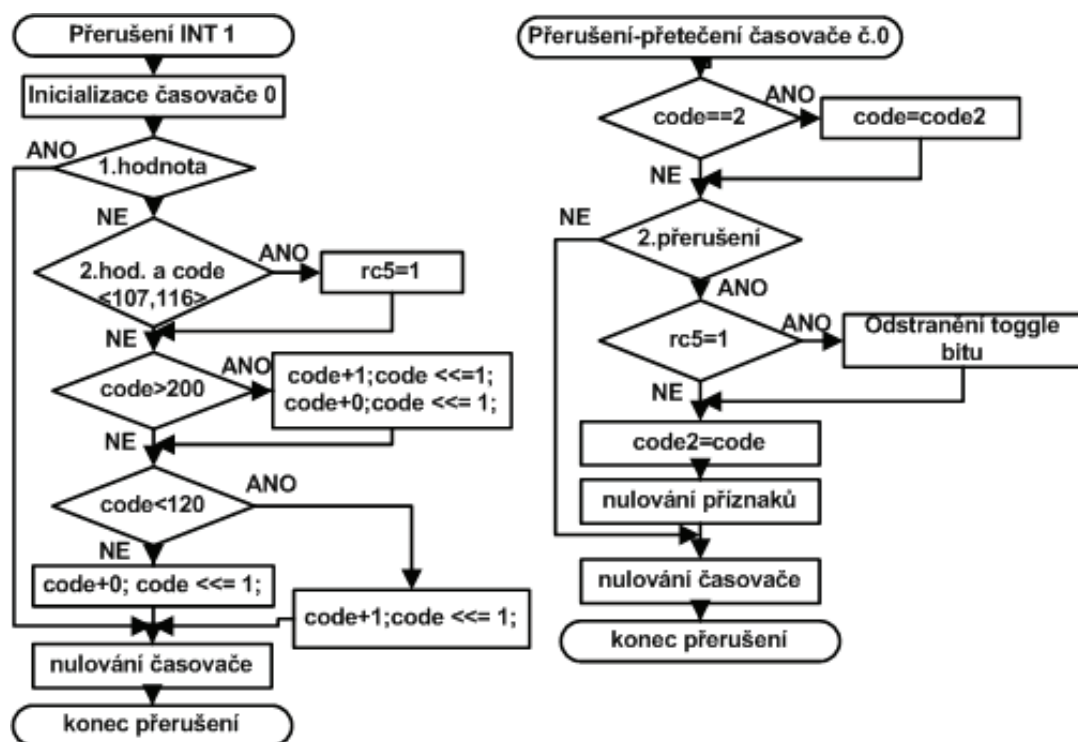
Základní myšlenkou navrženého algoritmu je využití PWM modulace a její rozdílné délky logických úrovní, které se snažíme měřit pomocí časovače č.2 a externího přerušení (viz obr. 5.5). Zvolený časovač č.2 je nastaven do normálního režimu, s děličkou nastavenou na dělicí poměr 256 a povoleným přerušením od přetečení časovače. To znamená, že jeden tik časovače přichází každých 16us a k přetečení dochází po 4,096ms. Externí přerušení na pinu PD3 je nastaveno, aby vyvolalo přerušení při každé sestupné hraně. Drobnou nevýhodou tohoto přerušení je, že nedokáže detekovat první startovací bit přijatého datového rámce. To ovšem v tomto případě nevádí, jelikož i tak jsou další bity svou délkou odlišné.

Po příchodu sestupné hrany od demodulátoru program skočí do obsluhy přerušení ext. přerušení. Pokud přijde první bit, jedná se o neplatnou hodnotu a časovač se vynuluje a znovu spustí. Po příchodu další sestupné hrany přerušení přečte délku předchozího impulsu a vyhodnotí ji. Vyhodnocení probíhá na základě rozsahů načtených hodnot a ukládá se do proměnné `code`. Experimentálně byla zjištěna hodnota 120 tiků časovače (přibližně 2ms), která rozděluje příchozí logické úrovně. V případě, pokud přijde impuls o délce menší než 120, do proměnné `code` se zapíše 0 a celá se posune rotací o jedno místo vlevo. Pro hodnotu větší než 120 se provede stejný postup, pouze se zapíše 1. Poté se časovač vynuluje a opět čeká na sestupnou hranu. Dekodování je ukončeno až poté, kdy dojde dvakrát za sebou k přetečení časovače, což značí mezeru delší než 8,192ms. Při ukončení se provede zkopírování výsledné hodnoty v proměnné `code` do proměnné `code2`, se kterou se dále pracuje v hlavní smyčce programu.

Použitý algoritmus byl funkční pouze pro protokoly s PWM modulací jako je NEC a proto byla do něj přidána část dekodující protokol RC-5. První problém, který u tohoto algoritmu nastal, byl s toggle bitem protokolu RC-5. Tento bit se mění při klikání na tlačítko, při držení tlačítka se nemění. To znamená, že každé tlačítko generuje 2 kódy, jeden s toggle bitem v log.0 druhý v log.1. Program ovšem počítá pouze s jedním kódem na tlačítko a proto tento stav musel být ošetřen. Toho bylo dosaženo

odstraněním 9-ti horních bitů výsledného kódu, který byl dekodován jako protokol RC5. Mezi horní bity patří i toggle bit, takže po odstranění zůstanou pouze bity charakterizující jednotlivá tlačítka. Druhým problémem bylo dekodování, jelikož příchozí zpráva po změření času nabývala tří hodnot a proto už nešlo jednoznačně určit log.0 a log.1, proto byla do algoritmu vnesena podmínka, která v případě, že je příchozí impuls delší než 200 tiků časovače, uloží do proměnné `code` dvě hodnoty najednou a to nejprve log.1 a poté log.0.

Jako další problém při dekodování se ukázal rozdílný přístup protokolů ovladačů k odesílání zprávy. Rozdíly se projevují v opakování zasílání zprávy při držení tlačítka na ovladači. Některé ovladače odesílají zprávu pouze jedenkrát, bez opakování, jiné ji opakují s určitou periodou po dobu držení tlačítka. V případě opakování signálu se můžeme setkat s tím, že je opakována stále stejná zpráva (viz. RC5), nebo je po odeslání zpráva odeslána a za ní se v určitých periodách odesílají opakující impulsy indikující držení tlačítka (viz. NEC). Stav, kdy jsou přijaty pouze opakující pulzy, je ošetřen v přerušení od časovače. Opakovací impulsy jsou dekodovány jako číslo 2, takže pokud je výsledný kód právě 2, do proměnné `code` se zapíše předchozí hodnota, tedy hodnota z proměnné `code2`. Při testování se objevily mezi ovladači i takové, které opakování nepodporovaly a tudíž při každém zmáčknutí byla zpráva vygenerována jen jednou. Naopak jeden ovladač při zmáčknutí odesílal za sebou hned dvě tyto zprávy.



Obr. 5.5 Vývojový diagram IR dekodování

5.1.6 Režim „učení“

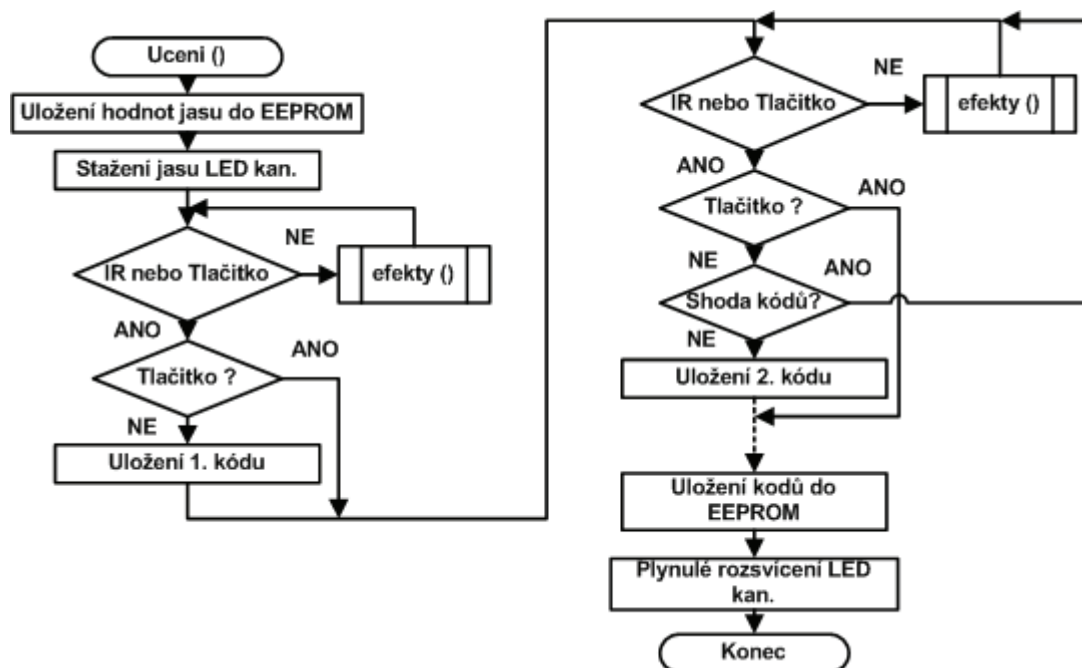
Režim učení slouží pro uložení nových kódů ovládacích tlačítek dálkového ovladače. Výrazně tedy rozšiřuje ovládání modulu, protože není vázán pouze na jeden konkrétní ovladač, ale kdykoli může být k ovládání použit jakýkoliv jiný ovladač. Modul je schopný rozpoznávat a tedy i naučit se rozpoznat ovladače s přenosovým protokolem NEC a RC5. U ostatních protokolů není zaručena funkčnost, pokud bychom chtěli ovládat všechny funkce.

Režim učení se spouští červeným tlačítkem na předním panelu. Pokud je spuštěn, rozsvítí se indikační LED na předním panelu, uloží se aktuální nastavení jasu LED kanálů a poté dojde k plynulému snížení jasu všech kanálů. Pak se rozbliká 1. kanál a program čeká na příjem IR kódu. Blikání se provádí pomocí funkce `efekty()`, která má jako vstupní parametry číslo kanálu a číslo stylu blikání (1- plynulé rozsvěcování, 2- plynulé pohasínání, 3- blikání). Styl blikání určuje, jakou funkci se tlačítko právě učí. Pokud LED diody pulsují směrem vzhůru, jde o tlačítko pro zvyšování jasu. Pokud pulsují směrem dolů, jde o tlačítko pro snižování jasu. Posledním případem je problikávání, které znázorňuje tlačítko pro vypínání a zapínání kanálu. Pořadí v jakém po sobě tlačítka jdou, je uvedeno v tabulce č.6. Po jeho přijetí dojde k zastavení blikání a k uložení do pole s kódy. Dále následuje menší prodleva a začne další blikání, které znázorňuje čekání na přijetí dalšího kódu. V tomto případě se může stát, že stiskneme některé z předchozích tlačítek. Pokud dojde k tomuto stavu, krátce probliknou všechny LED kanály a program se vrátí zpět a čeká na nový kód.

pořadí	funkce tlačítka	styl blikání v režimu učení
1.	snížení jasu kanálu 1	pulsování kanálu 1 směrem vzhůru
2.	zvýšení jasu kanálu 1	pulsování kanálu 1 směrem dolů
3.	vypnutí/zapnutí kanálu 1	blikání kanálu 1
4.	snížení jasu kanálu 2	pulsování kanálu 2 směrem vzhůru
5.	zvýšení jasu kanálu 2	pulsování kanálu 2 směrem dolů
6.	vypnutí/zapnutí kanálu 2	blikání kanálu 2
7.	snížení jasu kanálu 3	pulsování kanálu 3 směrem vzhůru
8.	zvýšení jasu kanálu 3	pulsování kanálu 3 směrem dolů
9.	vypnutí/zapnutí kanálu 3	blikání kanálu 3
10.	snížení jasu kanálu 4	pulsování kanálu 4 směrem vzhůru
11.	zvýšení jasu kanálu 4	pulsování kanálu 4 směrem dolů
12.	vypnutí/zapnutí kanálu 4	blikání kanálu 4
13.	min./max. jas všech kanálů	blikání všech kanálů
14.	uvedení do režimu spánku	pulsování všech kanálů

Tab. 6 Pořadí tlačítek při učení

V případě, že bychom chtěli některé kroky učení přeskočit, stačí zmáčknout červené tlačítko učení. Po ukončení učení posledního kódu se provede uložení všech kódu do paměti EEPROM, nastavení jasu LED kanálů do stavu před režimem učení, zhasnutí indikační LED a návrat do hlavního cyklu programu.



Obr. 5.6 Vývojový diagram režimu učení

5.1.7 Sériová komunikace po UART

K sériové komunikaci s PC je zde použita komunikační jednotka UART spojená s převodníkem FT232, který komunikuje s počítačem pomocí rozhraní USB. Komunikace mezi modulem a PC je duplexní, takže je možné posílat data v obou směrech.

Datová zpráva je tvořena začínajícím znakem, ukončovacím znakem a čtyřmi hodnotami jasu (viz obr. 5.7). Hodnoty jasu jsou odesílány jako hodnoty 0 (nejnižší jas) až 31 (nejvyšší jas). Znaky 'a' a 'z' jsou zvoleny tak, aby nekorespondovaly s těmito hodnotami. Jejich hodnoty v ASCII tabulce v hexadecimálním formátu jsou 61h a 7Ah.

Formát datové zprávy:



Obr. 5.7 Formát datové zprávy

Příklad datové zprávy v hexadecimálním formátu:

0x61|0x00|0x01|0x02|0x03|0x7A

Znaky na začátku a konci zprávy zabráňují jejímu nekorektnímu načtení. Po vyvolání přerušení program nejdříve uloží příchozí hodnoty do pole `Pole[6]` a zkontroluje jestli 1.byte v poli je shodný se znakem 'a' a 6.byte se znakem 'z'. Pokud ano, uloží hodnoty jasů v 2. až 5.bytu do pole `linKanal[4]`, pokud ke shodě nedojde, program uložení přeskočí a obsluha přerušení se ukončí.

Aby komunikace mezi PC a modulem fungovala, musí být konfigurace sériové linky shodná na obou stranách.

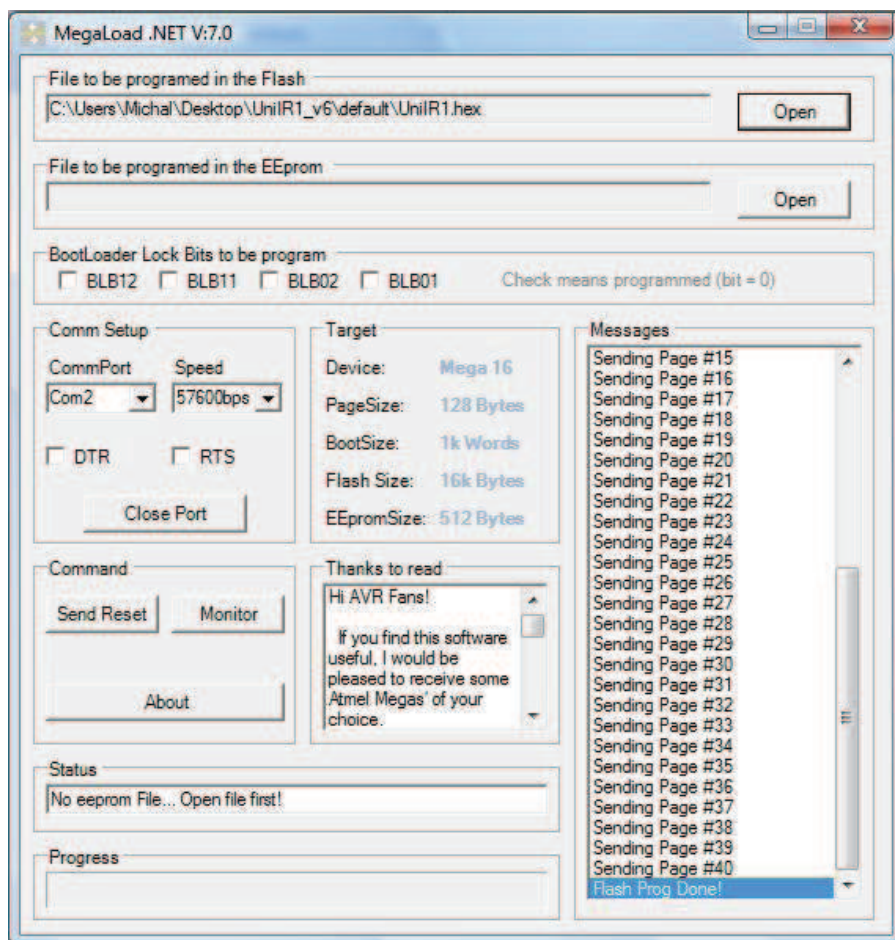
Parametry sériové komunikace:

Baudrate: 9600Bd
Datových bitů: 8
Parita: není
Stop bitů: 1
Handshaking: není

5.1.8 Zavaděč (bootloader)

Zavaděč je speciální kód, který slouží k naprogramování MCU i přes jiná, než programovací rozhraní (např. I2C, UART, SPI). Je uložen v horní části paměti a má za úkol přijímat kód a ukládat ho do dolní části programové paměti (flash). Z toho vyplývá, že není součástí navrženého softwaru a musíme ho do programové paměti nejprve nahrát pomocí programovacího rozhraní (v našem případě ISP). Aby nedošlo k jeho smazání, musí se použít nastavení pojistek – fuse bitů. Pro MCU ATmega16 je toto nastavení 0xD0 pro horní byte a 0xFF pro dolní byte. V našem případě jsme využili připojení přes USB, které slouží k ovládání LED kanálů. Výhodou tedy je, že zařízení nemusíme rozdělovat v případě, že bychom chtěli změnit program. Použitý zavaděč je dostupný na [<http://shop.onpa.cz/?podpora,28>]

Programování přes zavaděč se provádí pomocí programu MegaLoad.NET V:7.0, který je zdarma dostupný na [<http://www.and-tech.pl/evb43>]. Po propojení modulu s počítačem spustíme program MegaLoad.NET. Nastavíme parametry komunikace, které zahrnují nastavení komunikačního portu COM a přenosové rychlosti, která je pevně nastavena na 57600bps. Poté připojíme zdrojový kód s příponou .hex v kolonce flash a provedeme restart modulu tlačítkem umístěným v zadní části modulu. Tím dáme povel k nahrání zdrojového kódu do modulu. Pokud vše proběhne úspěšně, v pravé části programu s názvem Messages bude výpis ukončen zprávou: „Flash Prog Done!“. V případě, že se tak nestane, nahrávání opakujeme.



Obr. 5.8 Vzhled programu MegaLoad V:7.0:

5.2 Ovládací program pro PC

Jednoduchá aplikace pro ovládání modulu byla napsána v prostředí Microsoft Visual Studio 2010 v jazyku C#.

Aplikace je rozdělena na dvě části. První slouží ke komunikaci s modulem a druhá k nastavení jasu jednotlivých kanálů. Po inicializaci všech komponentů následuje nastavení vlastností sériového portu jako je komunikační rychlost a počet datových bitů. Poté se provede načtení volných portů pomocí metody `GetPortNames()`. V této chvíli se program zastaví a čeká na příchod některé z událostí:

`private void bt_connect_Click(object sender, EventArgs e)` – tato událost je spuštěna kliknutím na tlačítko „Connect“ nebo „Disconnect“. Pokud je port otevřený a dojde ke stisku tlačítka, port se uzavře, zakáže se pohyb táhel, povolí se volba portu a změní se název tlačítka na „Disconnect“. V opačném případě dojde k připojení k vybranému portu, a pokud se port otevře, povolí se pohyb táhel, zakáže se změna portu, vypíše se aktuální stav do historie a změní se název tlačítka na „Connect“.

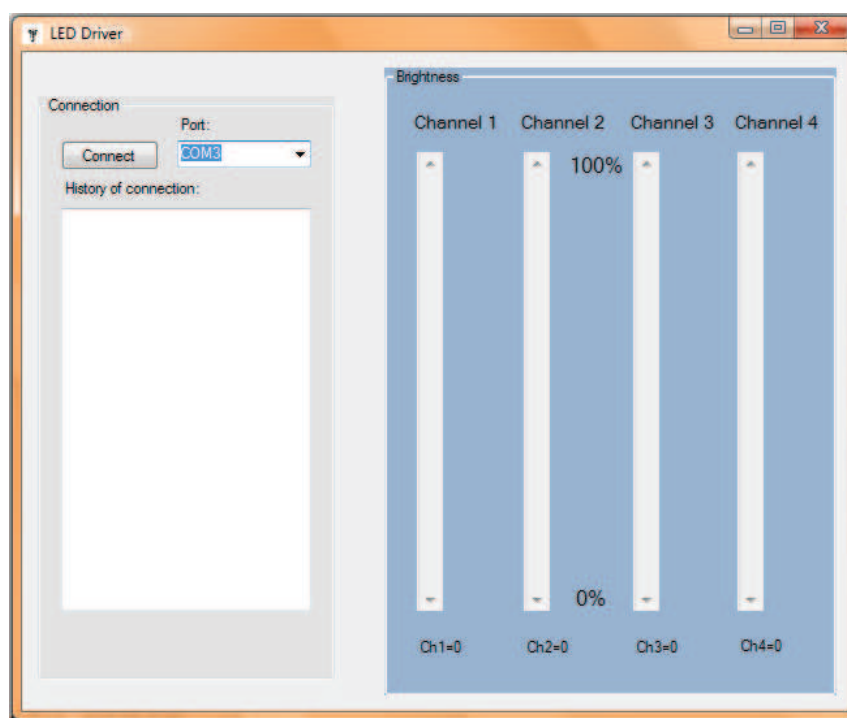
```
private void vSB_Kanal1_Scroll(object sender, ScrollEventArgs e)
```

- tato událost je spuštěna pohybem táhla pro určený kanál. Pokud je port otevřen aplikace pošle po sériové lince znak 'a', čtyři aktuální nastavení táhel a nakonec znak 'z' (viz. obr. 5.7). Tato událost je pro všechny čtyři kanály stejná.

```
private void vSB_Kanal1_ValueChanged(object sender, EventArgs e)
```

- událost slouží k výpisu aktuální hodnoty táhla do příslušného popisku umístěného pod táhlem. Událost se zavolá vždy, když se vyskytne změna hodnoty na příslušném táhlu.

Před připojením k modulu se musí zadat port, ke kterému se má aplikace připojit. Samotné připojení se provádí pomocí tlačítka „Connect“. To se po zmáčknutí změní na „Disconnect“ a tím samym tlačítkem se může připojení zrušit. V průběhu připojení nelze měnit nastavení portu. Pod nastavením portu se nachází bílé okno, které zobrazuje aktuální stav připojení (stav, port, rychlost přenosu [Bd], typ přenosu - počet bitů a parit) a jeho historii. Napravo se nachází blok sloužící k nastavování jasu jednotlivých kanálů. K tomu slouží čtyři posuvná táhla. Nad každým z nich se nachází popisek a pod nimi zkratka s číslem kanálu a aktuální nastavenou hodnotou, která se pohybuje v rozsahu 0 až 31. Tato hodnota odpovídá plnému rozsahu jasu.



Obr. 5.9 Vzhled aplikace pro ovládání LED okruhů z PC

6 ZÁVĚR

Cílem bakalářské práce bylo navrhnout a realizovat modul, který pomocí sériové linky a IR dálkového ovládání řídí několik světelných okruhů výkonových LED a navíc obsahuje funkci pro „učení se“ dálkových ovladačů.

Návrh byl o to složitější, že měl být doprovázen realizací. Při realizaci práce byl navržen a vytvořen funkční prototyp modulu. Ten může být napájen napětím od 7V do 30V, s výstupem na 4 LED kanály a maximálním odběrem 3,2A na kanál.

Po hardwarové stránce byla snaha udělat modul co nejvíce univerzální, aby jej bylo možné později modifikovat (např. přidání ovládacích prvků, kontrolních LED diod atd.). Sériová linka byla doplněna o převodník USB, což je bráno jako standart dnešních počítačů a proto by konektivita s nimi neměla být problémem.

Softwarová stránka práce se týká především ovládání IR dálkovým ovladačem. Zde byl vytvořen algoritmus, který dekoduje přenosové protokoly NEC a RC5. Ty dekoduje téměř bez chyby, pouze u protokolu RC5 generuje pro 2 tlačítka stejný kód. Další drobný problém nastane v případě, že nemíříme na IR přijímač. Může se stát, že je kód vlivem odrazu deformován a poté i špatně dekodován. To může způsobit při shodě s jinými kódy například změnu jasu jiného kanálu, než očekáváme. Modul je dále opatřen režimem, který zajistí naučení všech 14-ti ovládacích tlačítek, režimem spánku kvůli snížení spotřeby a plynulým pohasínáním a rozsvěcováním LED kanálů při vypínání.

Pro ovládání modulu z PC byla vytvořena aplikaci v jazyce C#, která pomocí táhel mění jas jednotlivých kanálů. Zde nebyly kladeny žádné speciální požadavky na provedení. Program se umí pouze připojit, odpojit a nastavit jas všem kanálům. Byla snaha i o plnou synchronizaci s dálkovým ovladačem, ale aplikace bohužel vždy po přijetí dat z modulu „spadla“. Proto je zajištěna pouze vzájemná kompatibilita tak ,aby se tyto dva druhy ovládání nerušily, protože každý z nich pracuje s jiným počtem kroků.

Nedostatky modulu, které se během návrhu ukázaly, jsou především chybějící ochrana výstupů proti zkratu nebo pojistka napájení modulu. Tyto nedostatky nemají přímý vliv na funkci modulu, avšak při neodborné manipulaci může dojít např. ke zničení výkonových tranzistorů.

Tato práce byla realizována v souladu se zadáním práce a modul i software jemu vyhovují. Drobné nedostatky nemají výrazný vliv na chod celého zařízení.

Práce byla zajímavá a tak se předpokládá její pokračování a vylepšování stávajících parametrů. V pokračování práce by byly použity dosavadní poznatky doplněné o některé vylepšení. Jednalo by se především o změnu procesoru, který by měl být mnohem výkonnější. To by mělo zaručit více kanálů PWM a rychlejší frekvenci alespoň okolo 1KHz. Dále by bylo vhodné zkusit komunikaci s PC pomocí některých

moderních rozhraní bluetooth, Wi-Fi nebo LAN. Nedílnou součástí další verze by měly být i ochrany, případně doplněné měřením parametrů na výstupu jako je proud a napětí.

LITERATURA

- [1] MATOUŠEK, David. *Práce s mikrokontroléry ATMEL AVR - ATmega16* : 4. díl - edice uP a praxe. 1.vydání. Praha : BEN, 2006. 320 s. ISBN 80-7300-174-8.
- [2] MATOUŠEK, David. *USB prakticky s obvody FTDI* - 1.díl : měření, řízení a regulace pomocí několika jednoduchých přípravků. Praha : BEN, 2003. 272 s. ISBN 80-7300-103-9.
- [3] SEDGEWICK, Robert . *Algoritmy v C, části 1-4 : základy datové struktury třídění vyhledávání* . [s.l.] : Softpress, 2003. 688 s. ISBN 80-86497-56-9.
- [4] VÁŇA, Vladimír. *Mikrokontroléry ATMEL AVR Programování v jazyce C*. 1.vydání. Praha : BEN, 2003. 215 s. ISBN 80-7300-102-0.
- [5] BURKHARD, Mann. *C pro mikrokontroléry : ANSI-C, kompilátory, linkery, příklady, nástroje, tipy a triky*,... 1. české vydání. Praha : BEN, 2003. 280 s. ISBN 80-7300-077-6.
- [6] HRBÁČEK, Jiří . *Komunikace mikrokontroléru s okolím* - 1.díl. 1. vydání. Praha : BEN, 1999. 159 s. ISBN 80-86056-42-2.
- [7] *Porovnání s klasickými žárovkami* [online]. 2010 [cit. 2011-05-26]. Dostupné z: <<http://www.led-zarovky-svetla.cz/>>
- [8] HANKOVEC, David. *Příjem a dekódování telegramu infračervených dálkových ovládačů* [online]. 2002 [cit. 2010-12-03]. Dostupné z: <<http://www.dhservis.cz/>>
- [9] HIRZEL, Timothy. *PWM* [online]. 2009 [cit. 2011-05-26]. Dostupné z: <<http://www.arduino.cc/en/Tutorial/PWM>>
- [10] *IR remote controls* [online]. 2004 [cit. 2010-12-03]. Dostupné z: <<http://www.sbprojects.com/>>
- [11] *Datasheet TSOP4836* [online]. 2003 [cit. 2010-12-03]. Dostupný z: <<http://www.alldatasheet.com/datasheet-pdf/pdf/252285/VISHAY/TSOP4836.html>>
- [12] ČAPEK, Lukáš. *Relaxační LED lampička s převodem HSV do RGB* [online]. 2008 [cit. 2011-05-26]. Dostupné z: <<http://hw.cz/teorieapraxe/konstrukce/art2400-relaxacni-led-lampicka-s-prevodem-hsv-do-rgb.html>>
- [13] *Normy pro IR datový přenos* [online]. 1999 [cit. 2011-05-26]. Dostupné z: <http://hw.cz/docs/ir_prenos/ir_protokoly.html>
- [14] *LED žárovky* [online]. 2010 [cit. 2011-05-26]. Dostupné z: <<http://www.deramax.cz/led-zarovky/t-117/>>
- [15] *Datasheet ATmega16*. Atmel [online]. Dostupné z: <http://www.atmel.com/dyn/resources/prod_documents/doc2466.pdf>
- [16] *Podle čeho vybírat LED žárovky*. [online]. 2010 [cit. 2011-05-26]. Dostupné z: <<http://www.led-zarovky-osvetleni.cz/>>

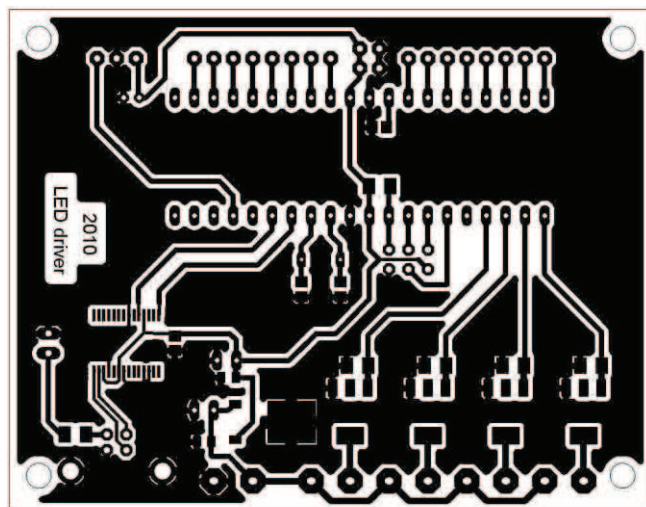
SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

IR	Infra Red, infra červené
ISP	In System Programming, programování za běhu
LED	Light-emitting diode, světlo vyzařující dioda
PC	Personal Computer, osobní počítač
PWM	Pulse Width Modulation, pulzně šířková modulace
SMD	Surface Mount Device, součástka pro povrchovou montáž
MCU	Micro Controller, mikrokontrolér
USART	Universal Synchronous and Asynchronous Receiver and Transmitter, univerzální synchronní a asynchronní přijímač a vysílač
USB	Universal Serial Bus, univerzální sériová sběrnice
FSK	Frequency - Shift Keying, frekvenční klíčování
CFL	Compact fluorescent lamp, kompaktní fluorescenční lampa
LCD	Liquid crystal display, Displej z tekutých krystalů
DO	Dálkové ovládání
DPS	Deska plošných spojů
Bd	Baudrate- rychlost přenosu sériové linky (1Bd=1bit/sec)

SEZNAM PŘÍLOH

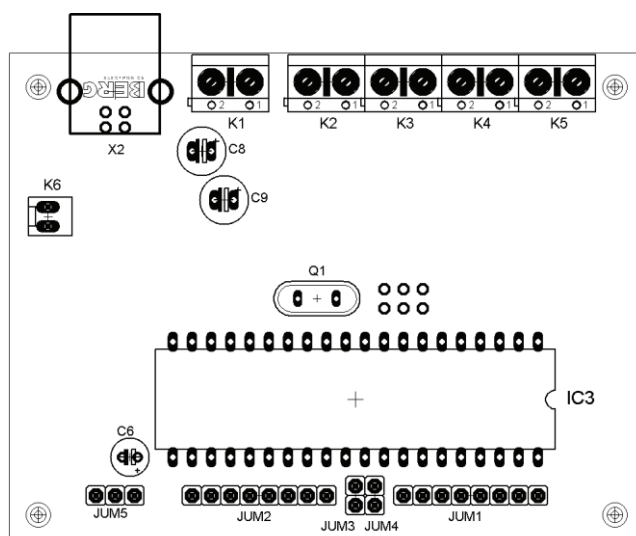
A	Návrh zařízení	44
A.1	Obvodové zapojení	44
A.2	Deska plošného spoje – bottom (strana spojů)	45
A.3	Osazovací schéma – top (strana součástek)	45
A.4	Osazovací schéma – bottom (strana spojů)	46
B	Seznam součástek	47
C	Fotografie modulu	48
D	Návod na obsluhu modulu	49
E	Obsah přiloženého CD	54

A.2 Deska plošného spoje – bottom (strana spojů)



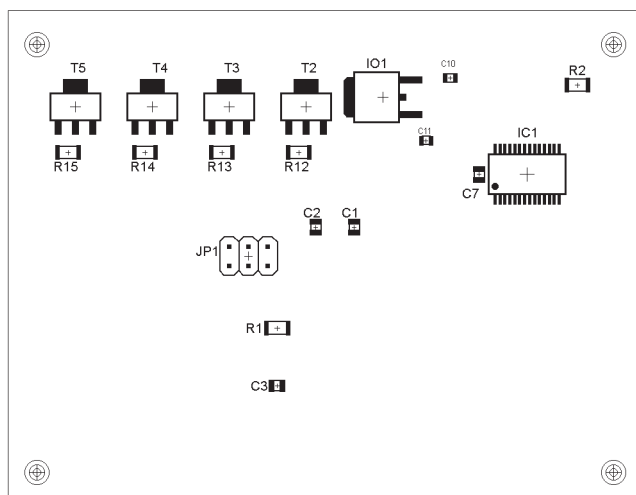
Rozměr desky 82 x 65 [mm], měřítko M1:1

A.3 Osazovací schéma – top (strana součástek)



Rozměr desky 82 x 65 [mm], měřítko M1:1

A.4 Osazovací schéma – bottom (strana spojů)



Rozměr desky 82 x 65 [mm], měřítko M1:1

B SEZNAM SOUČÁSTEK

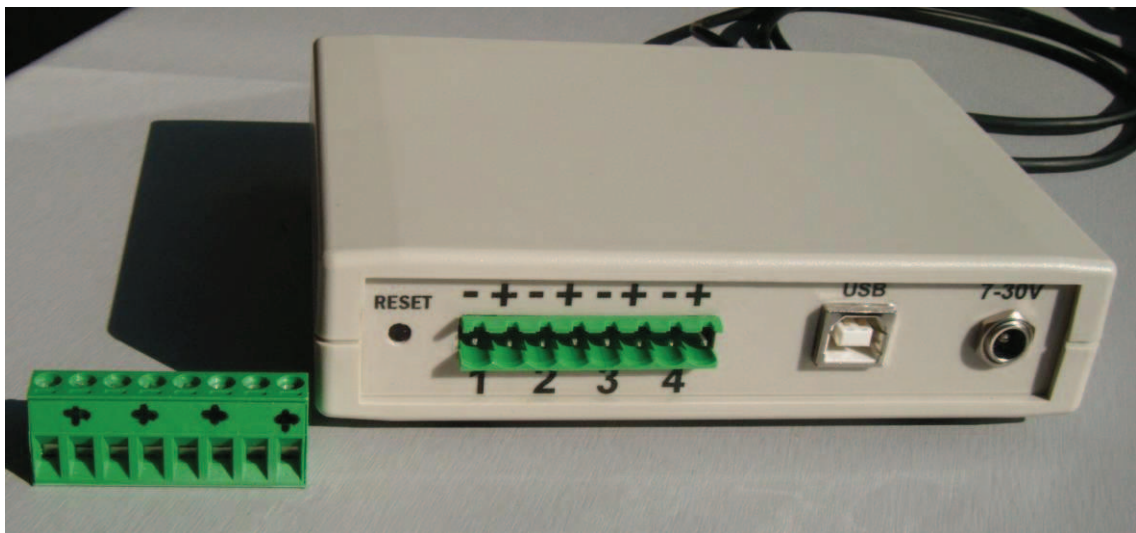
Označení	Hodnota	Pouzdro	Popis
C1	22pF	C0805	Keramický kondenzátor SMD
C2	22pF	C0805	Keramický kondenzátor SMD
C3	100nF	C0805	Keramický kondenzátor SMD
C6	4,7uF	E2-5	Elektrolytický kondenzátor
C7	22pF	C0805	Keramický kondenzátor SMD
C8	10uF	C-EL_2,5	Elektrolytický kondenzátor
C9	100uF	C-EL_2,5	Elektrolytický kondenzátor
C10	100nF	C0805	Keramický kondenzátor SMD
C11	100nF	C0805	Keramický kondenzátor SMD
IC1	FT232RL	SSOP28	Převodník USB <->UART
IC3	MEGA16-P	DIL40	8-bitový mikrokontrolér
IO1	7805 SMD	D2PAK	5V stabilizátor napětí SMD
JP1	ISP	S2G3_JUM	Konektor 2X3
JUM1	S1G8_JUMP	S1G8_JUM	Konektor 1x8
JUM2	S1G8_JUMP	S1G8_JUM	Konektor 1x8
JUM3	ext. nap. +	S1G2_JUM	Konektor 1x2
JUM4	ext. nap. GND	S1G2_JUM	Konektor 1x2
JUM5	IR	S1G3_JUM	Konektor 1x3
K1	ARK500/2	ARK500/2	Napájecí konektor
K2	LED okruh	ARK500/2	Svorkovnice
K3	LED okruh	ARK500/2	Svorkovnice
K4	LED okruh	ARK500/2	Svorkovnice
K5	LED okruh	ARK500/2	Svorkovnice
K6	USB POWER	PSH02-02P	Konektor
Q1	16MHz	QS	Krystal
R1	10k	M1206	Rezistor SMD
R2	2k2	M1206	Rezistor SMD
R12	2k2	M1206	Rezistor SMD
R13	2k2	M1206	Rezistor SMD
R14	2k2	M1206	Rezistor SMD
R15	2k2	M1206	Rezistor SMD
T2	IRLL2705	SOT223	Výkonový tranzistor
T3	IRLL2705	SOT223	Výkonový tranzistor
T4	IRLL2705	SOT223	Výkonový tranzistor
T5	IRLL2705	SOT223	Výkonový tranzistor
X2	PN61729-S	PN61729-S	USB konektor

C FOTOGRAFIE MODULU

Pohled zepředu:



Pohled zezadu:



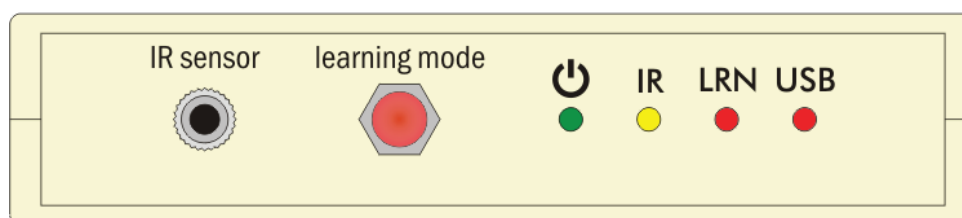
D NÁVOD NA OBSLUHU MODULU

Popis modulu:

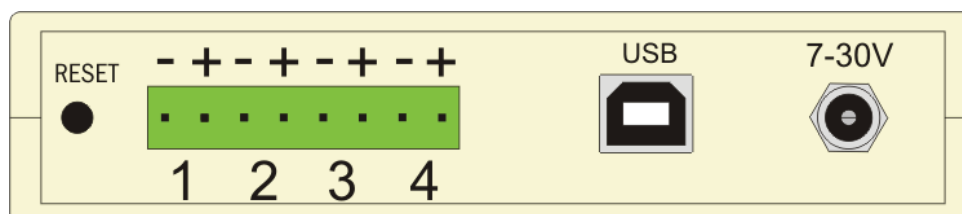
Tento modul je určen k řízení jasu 4 světelných okruhů výkonových LED diod. Řízení je možné provádět pomocí IR dálkového ovladače (např. od televize, radia, videa) nebo počítačem vybaveným USB konektorem. Modul disponuje funkcí „učení“ příkazů od dálkového ovladače. Tato funkce značně rozšiřuje jeho možnosti, jelikož může být ovládán téměř jakýmkoliv ovladačem.

K modulu je možné připojit pouze LED okruhy s napájením totožným s napájením modulu, tedy 7-30V a maximálním proudem 3,2 A na okruh.

Přední panel:



Zadní panel:



Parametry modulu:

Napájecí napětí:	7 - 30V
Max. proud kanálem:	3,2 A
Počet LED kanálů:	4
Spotřeba v režimu spánku (bez USB):	0,4 W
Spotřeba v režimu spánku (s USB):	0,6 W
Podporované protokoly DO:	RC5, NEC
Počet programovatelných tlačítek:	14
Připojení k PC:	USB 1.1 a kompatibilní
Podporované verze OS WIN:	Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2
Rozměry:	130x100x30mm

Připojení a obsluha modulu

1. Do zadní části modulu připojíme na zelené svorky LED kanály s napájecím napětím shodným jako je napájecí napětí modulu. POZOR! Zkontrolujeme správné zapojení polarity LED kanálů.
2. Do zdířky označené „IR sensor“ na předním panelu připojíme IR přijímač umístěný na kabelu.
3. Napájecí adaptér připojíme do zásuvky a poté výstupní konektor zapojíme do zadní části modulu.
4. Na předním panelu se rozsvítí zelená indikační LED dioda, která značí běh modulu. Pokud se tak stane, je modul připraven k použití.
5. Chceme-li, aby byl modul ovládán novým ovladačem, stiskneme červené tlačítko na předním panelu označené nápisem „learning mode“. Modul přejde do módu „učení“ a rozsvítí červenou indikační LED diodu „LRN“, všechny LED kanály pohasnou a 1. LED kanál začne pulzovat.
6. Nyní vezmeme ovladač, který chceme použít a namíříme ho na IR přijímač. Vybereme tlačítko pro funkci a krátce stiskneme. Pokud vše proběhlo v pořádku, LED kanál změní styl blikání. Poté vybereme další tlačítko a opět krátce stiskneme. Tento postup opakujeme, dokud se modul nenaučí všechny tlačítka.

Pořadí tlačítek při učení		
pořadí	funkce tlačítka	styl blikání v režimu učení
1.	snížení jasu kanálu 1	pulsování kanálu 1 směrem vzhůru
2.	zvýšení jasu kanálu 1	pulsování kanálu 1 směrem dolů
3.	vypnutí/zapnutí kanálu 1	blikání kanálu 1
4.	snížení jasu kanálu 2	pulsování kanálu 2 směrem vzhůru
5.	zvýšení jasu kanálu 2	pulsování kanálu 2 směrem dolů
6.	vypnutí/zapnutí kanálu 2	blikání kanálu 2
7.	snížení jasu kanálu 3	pulsování kanálu 3 směrem vzhůru
8.	zvýšení jasu kanálu 3	pulsování kanálu 3 směrem dolů
9.	vypnutí/zapnutí kanálu 3	blikání kanálu 3
10.	snížení jasu kanálu 4	pulsování kanálu 4 směrem vzhůru
11.	zvýšení jasu kanálu 4	pulsování kanálu 4 směrem dolů
12.	vypnutí/zapnutí kanálu 4	blikání kanálu 4
13.	min./max. jas všech kanálů	blikání všech kanálů
14.	uvedení do režimu spánku	pulsování všech kanálů

Pozn.: Pokud dojde ke shodě kódů nebo ke zmáčknutí stejného tlačítka, všechny kanály krátce probliknou. To znamená, že musíme použít jiné tlačítko.

Pozn.2: Pokud chceme změnit ovládání jen některých kanálů, můžeme kroky v učení přeskakovat zmáčknutím červeného tlačítka na přední straně modulu.

7. Po úspěšném naučení všech 14-ti tlačítek zhasne červená LED dioda a LED kanály se rozsvítí do původního nastavení jasu použitým před režimem učení a jsou připraveny k ovládání pomocí dálkového ovladače.

Ovládání modulu pomocí PC

1. Ovládací modul připojíme k PC pomocí USB kabelu (typ A-B). Pokud se rozsvítí červená LED „USB“ na předním panelu modulu, je modul k PC připojen.
2. Nainstalujeme ovladače převodníku USB (ovladače jsou dostupné na CD nebo adrese: www.ftdichip.com/Drivers/VCP.htm).
3. Spustíme aplikaci pro ovládání modulu.
4. Zjistíme komunikační port (Tento počítač → Vlastnosti → Správce zařízení → Porty (COM a LPT) → USB Serial Port(hledaný port např. COM2)).
5. V aplikaci vybereme nalezený komunikační port a připojíme se k modulu pomocí tlačítka Connect.
6. Nyní můžeme pomocí čtyř táhel ovládat jas jednotlivých LED kanálů.
7. Před ukončením aplikace se zrušíme komunikaci s modulem pomocí tlačítka „Disconnect“. Poté aplikaci zavřeme.

Změna programu modulu

Změnit program modulu můžeme jednoduše pomocí připojení USB a zavaděče (bootloaderu), kterým disponuje mikrokontrolér uvnitř modulu.

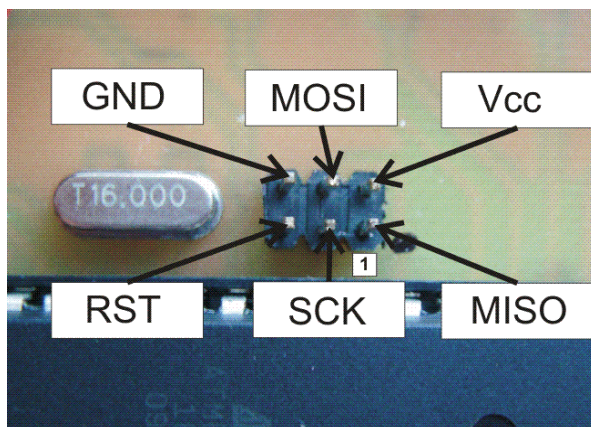
1. Pokud nejsme propojeni s modulem, provedeme připojení s PC pomocí USB kabelu (typ A-B). Pokud jsme připojeni a požíváme program pro řízení jasu, odpojíme se od sériové linky pomocí tlačítka „Disconnect“.
2. Spustíme program Megaload V7.0 (program je dostupný na CD nebo adrese: <http://www.and-tech.pl/evb43>).
3. Nastavíme komunikační port (předchozí kapitola bod č.4) a rychlost komunikace na 57600Bd.
4. Připojíme soubor s příponou ‘.hex ‘.

5. Stiskneme tlačítko „Restart“ na zadní straně modulu.
6. Pokud vše proběhlo správně, v pravé části programu se vypíše počet zapsaných paketů zakončený zprávou „Flash Prog Done!“.

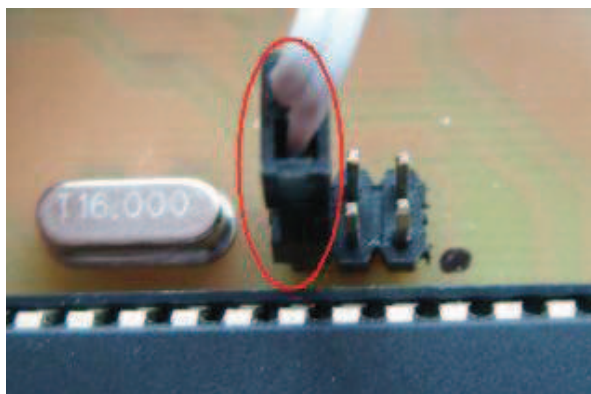
Servisní manuál – Programování přes ISP

V případě, že bude potřeba vyměnit mikrokontrolér nebo přehrát bootloader, bude nutné nejdříve krabičku rozšroubovat. Poté máme 2 možnosti:

1. Pokud nevlastníte programovací zařízení s rozhraním ISP (In System Programming), vyjměte mikrokontrolér z patice a vyměňte nebo přeprogramujte v externím zařízení.
2. Vlastníte-li zařízení k programování mikrokontrolérů s rozhraním ISP můžete využít vyvedeného ISP konektoru na desce modulu. K tomu, aby jste konektor mohli využít, musíte nejdříve od něj odpojit kabel resetovacího tlačítka. Poté připojte programovací zařízení na piny (viz. obrázek 1) a proveďte přeprogramování. Po úspěšné úpravě programu vraťte piny resetovacího tlačítka zpět na původní místo (viz, obrázek 2) a krabičku uzavřete.



Obrázek 1. Popis pinů konektoru ISP



Obrázek 2. Umístění konektoru resetovacího tlačítka na konektoru ISP

Kontaktní údaje:

e-mail: Michal.Hromadka@seznam.cz

telefon: +420 604 513 560

E OBSAH PŘILOŽENÉHO CD

PC:

- Program AVR Studio 4.18
- Program MegaLoad .NET V:7.0
- Ovladače VCP (Virtual COM port) pro FT232RL
- Ovládací program pro PC v C#

MCU:

- Software pro ATmega16
- Bootloader pro ATmega16
- Návod na programování pomocí bootloADERu

Modul:

- Podklady DPS v Eagle 5.7.0

Dokumentace:

- Dokumentace
- Návod k obsluze modulu