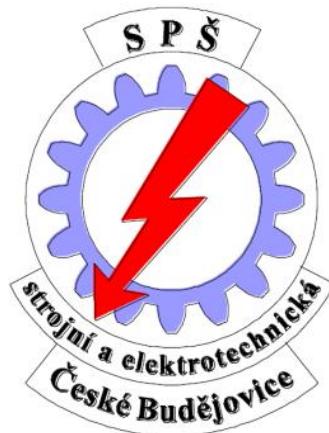


**STŘEDNÍ PRŮMYSLOVÁ ŠKOLA STROJNÍ A ELEKTROTECHNICKÁ
ČESKÉ BUDĚJOVICE, DUKELSKÁ 13**



**NÁVRH A REALIZACE ROBOTA SLEDUJÍCÍHO ČERNOU ČÁRU
MATURITNÍ PRÁCE**

Autor práce: **Lukáč Čížek**
Vedoucí práce: **Bc. Jakub Kákona**
Oponent: **Ing. Jan Janoud**

Školní rok: **2012/2013**

Prohlašuji, že jsem maturitní práci s názvem: "*Návrh a realizace robota sledujícího černou čáru*" vypracoval samostatně pod vedením Bc. Jakuba Kákony s použitím literatury uvedené na konci mé maturitní práce v seznamu použité literatury.

V Českých Budějovicích dne 26.února 2013

.....
Lukáš Čížek

Chtěl bych poděkovat všem, kteří mi umožnili realizovat tuto práci, zvláště vedoucímu mé maturitní práce Bc. Jakubu Kákonovi, dále pak mému oponentovi Ing. Janu Janoudovi, učiteli automatizace Ing. Oldřichu Smutnému, ale také mým přátelům, spolužákům a rodině za inspiraci i jejich trpělivost.

Konstrukce robota byla realizována z příspěvku Střední průmyslové školy strojní a elektrotechnické v Českých Budějovicích a z prostředků firmy Universal Scientific Technologies s.r.o. Za tyto příspěvky mnohokrát děkuji.

Zadání odborné maturitní práce

Zadání je vystaveno pro žáka:

Jméno a příjmení: **Lukáš Čížek**

Datum narození: **31. 1. 1994**

Žák třídy: **4.EA ve šk. r. 2012 - 2013**

Téma maturitní práce:

Návrh a realizace robota sledujícího černou čáru

Požadované dílčí body práce:

1. návrh a realizace modulu pro Li-Ion akumulátor, který zajistí jeho ochranu a dobíjení
2. vhodný výběr senzorů pro sledování čáry, návrh modulu a jeho realizace
3. návrh a realizace modulu H-můstků pro řízení motorů
4. návrh a realizace podvozku
5. výběr vhodného MCU pro robota
6. napsání programu pro vybraný procesor

Cíl maturitní práce:

Návrh, konstrukce, implementace programu pro sledování čáry a zpracování dokumentace. Zajištění kompatibility všech součástí a modulů s opensource modulární stavebnicí MLAB

Rozsah práce: minimálně 20 stran textu + elektrické schéma

Termín odevzdání práce: do 21. dubna 2013 ve dvou vyhotoveních

Vedoucí práce: Bc. Jakub Kákona

Oponent: Ing. Jan Janoud

Kritéria hodnocení: Původnost, technická úroveň návrhu, formální úroveň zprávy, úroveň ústní prezentace při obhajobě práce

V Českých Budějovicích 19. 12. 2013

Ing. Čestmír Tschauder,
ředitel školy

Abstrakt

Práce se zabývá rozborem problémů vznikajících při úkolu sledování černé čáry, jejich řešením a výslednou konstrukcí robota včetně napsání programu. Tato úloha je poměrně častou výukovou metodou senzoriky a autonomního řízení. Lze se s ní také setkat v různých robotických soutěžích po celém světě.

Klíčová slova: sledování čáry, senzorika, autonomní řízení, autonomní robot, rádkový senzor

OBSAH

1. ÚVOD	1
2. BLOKOVÉ SCHÉMA ROBOTA.....	2
3. ROZBOR PROBLÉMŮ A JEJICH ŘEŠENÍ	3
3.1. NAPÁJENÍ.....	3
3.1.1. AKUMULÁTORY.....	3
3.1.2. STABILIZACE NAPĚtí	4
3.2. BUZENÍ MOTORŮ.....	6
3.2.1. STEJNOSMĚRNÉ KOMUTÁTOROVÉ MOTORY.....	6
3.2.2. H - MŮSTKY	7
3.2.3. RUŠENÍ	9
3.3. SENZORY	9
3.3.1. ŘÁDKOVÝ SNÍMAČ	10
3.3.2. ODRAZOVÝ SENZOR.....	11
3.3.3. DOTYKOVÝ SENZOR.....	12
3.3.4. INFRAČERVENÝ DÁLKOMĚR.....	13
3.4. VÝBĚR PROCESORU.....	14
3.4.1. PROCESOR PIC16F887.....	14
4. ŘÍDÍCÍ PROGRAM	15
4.1. PROCESNÍ DIAGRAM	15
4.2. POPIS PROGRAMU	16
4.2.1. INDIKACE	16
4.2.2. ŘÍZENÍ MOTORŮ.....	17
4.2.3. ČTNÍ ANALOGOVÝCH SENZORŮ	18
4.2.4. PRÁCE S OPTICKÝM ŘÁDKOVÝM SNÍMAČEM	18
4.2.5. VYHODNOCENÍ.....	22
4.2.6. ŘÍDÍCÍ ČÁST PROGRAMU A DIAGNOSTIKA	24

5. KONSTRUKCE	25
5.1. POPIS MECHANICKÉ KONSTRUKCE	25
5.2. FOTOGRAFIE ROBOTA	25
6. ZÁVĚR.....	27
7. POUŽITÁ LITERATURA.....	28
8. OBSAH PŘILOŽENÉHO CD	30
9. PŘÍLOHA A - Dokumentace navržených modulů	32
10. PŘÍLOHA B - Výkresy plošných spojů navržených modulů	70

1. ÚVOD

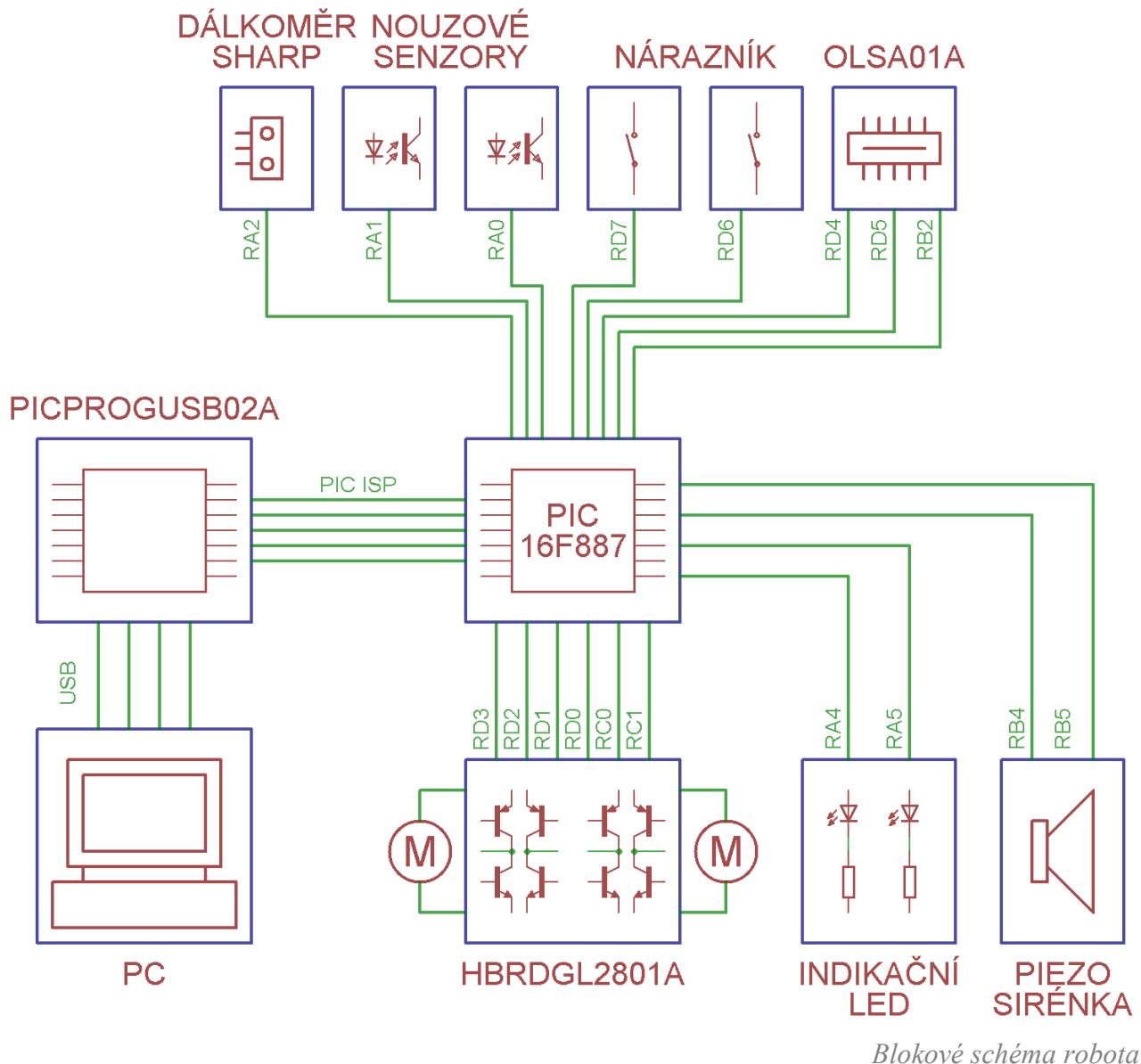
Robotické aplikace se stále více vyskytují v průmyslu, což je důvodem, proč se rozvíjí na středních odborných školách výuka automatizace, robotiky a mikroprocesorové techniky, a tak lze na středních i vysokých školách narazit krom teoretické výuky i na výuku praktickou, což znamená, že je nutné do škol poskytnout potřebné vyučovací pomůcky.

Právě robot, který bude sledovat černou čáru, je typickou výukovou aplikací z oblasti senzoriky a samočinného řízení a proto by mohl dobře posloužit i jako pomůcka pro výuku.

Tato práce si neklade za cíl vytváření nových poznatků z oblasti senzoriky a řízení, ale snaží se o výběr a aplikaci dosud známých způsobů měření a řízení tak, aby bylo možné na základě zkonztruovaného prototypu robota, dokumentace a získaných poznatků sestrojit spolehlivé a rychlé roboty pro sledování čáry.

2. BLOKOVÉ SCHÉMA ROBOTA

Níže lze vidět blokové schéma robota. Je zde zakresleno pouze propojení jednotlivých součástí s procesorem, není zde naznačeno napájení. Logické části obvodů jsou napájeny stabilizovaným zdrojem 5V, k jehož získání slouží modul LINREG01A. Silové části jsou napájeny přímo z baterii (zde se jedná o H-můstek HBRDGL29801A a vstupní napájení modulu stabilizovaného zdroje LINREG01A).



Jako analogové vstupy jsou využity vývody RA0 a RA1 pro modré odrazové senzory a RA2 pro infračervený dálkoměr. Počítač se k robotovi připojuje pouze při nahrávání programu do procesoru. Za tímto účelem je také nutné propojit PICPROGUSB20A s procesorem pomocí PIC ISP (využívá signály PGC, PGD, VDD, GND a MCLR).

Všechny vývody krom vývodů RB2, RD7 a RD6 jsou v programu nastaveny jako výstupní. Tyto piny slouží jako digitální vstupy. Piny RC0 a RC1 slouží pro přenos signálu PWM pro řízení rychlosti motorů.

3. ROZBOR PROBLÉMŮ A JEJICH ŘEŠENÍ

Základním problémem bylo vyřešit, jaký druh baterie zvolit pro napájení, jak zajistit její ochranu a dobíjení, a jakým způsobem realizovat stabilizaci napětí 5V pro napájení logiky, tedy procesoru a logické části H-můstku a senzorů.

3.1. NAPÁJENÍ

3.1.1. AKUMULÁTORY

V dnešní době lze na trhu zakoupit nepřeberné množství akumulátorů elektrické energie různých typů a odlišných provedení. Při výběru vhodného akumulátoru jsem se rozhodoval podle nominálního napětí článku, počtu nabíjecích cyklů článku, kapacity, ale vzhledem k tomu, že se jedná o pojízdného robota, rozhodovala při výběru i hmotnost a geometrické rozměry.

Vzhledem k dostupnosti jsem se nakonec rozhodoval mezi NiMh (Nikl - metal hybridový) a Li-Ion (Lithium - iontový). Jejich vlastnosti jsou stručně popsány níže.

Výhodnější se nakonec ukázalo použít Li-Ion akumulátor, byť je pro jeho použití potřeba navrhnut ochranné obvody, protože při nesprávném zacházení, jako je např. přebíjení nebo zkrat, může dojít k jeho vznícení a explozi.

NiMh akumulátory

NiMh články mají poměrně nízké jmenovité napětí 1,2 V. Na plně nabitém článku je napětí až 1,4 V, na vybitém 1,0 V. Životnost akumulátorů se pohybuje do 1000 nabíjecích cyklů. Na další stránce jsou uvedeny základní vlastnosti těchto akumulátorů

Výhody:

- bezpečnost provozu (*oproti Li-Ion akumulátorům*)
- jednoduchost aplikace (*pro většinu aplikací odpadá potřeba obvodů pro kontrolu a ochranu baterie*)

Nevýhody:

- malá kapacita (*proto články mají větší rozměry a hmotnost*)
- malé jmenovité napětí (1,2 V)
- samovybíjení (*až 30 % za měsíc*)
- nelze je vybíjet a nabíjet velkými proudy (*dochází k degradaci elektrod*)

Li-Ion akumulátory

Li-Ion akumulátory se dnes běžně používají ve spotřební elektronice, zejména v notebookech a mobilních telefonech. Jmenovité napětí je přibližně 3,7 V a vydrží až 2000 nabíjecích cyklů.

Výhody:

- vysoká hustota energie (*mají velkou kapacitu i při malé hmotnosti a rozměrech*)
- netrpí samovybíjením (*pod 5 % za měsíc*)
- nemají paměťový efekt (*způsobuje náhlý pokles napětí při zatížení, pokud akumulátor nebyl před nabítím plně vybit*)

Nevýhody:

- stárnutí (*baterie pomalu ztrácí kapacitu i bez ohledu, zda je používána či nikoliv*)
- nebezpečí výbuchu nebo vznícení při špatném zacházení (*nutnost ochranných obvodů*)
- nesmí se úplně vybit (*při napětí pod 3 V je velmi těžké je zpět oživit - ztratí kapacitu*)

Pro použití Li-Ion akumulátoru jsem provedl návrh modulu LION2CELL01A, jehož dokumentace je uvedena v kapitole 9. "Dokumentace navržených modulů".

3.1.2. STABILIZACE NAPĚTÍ

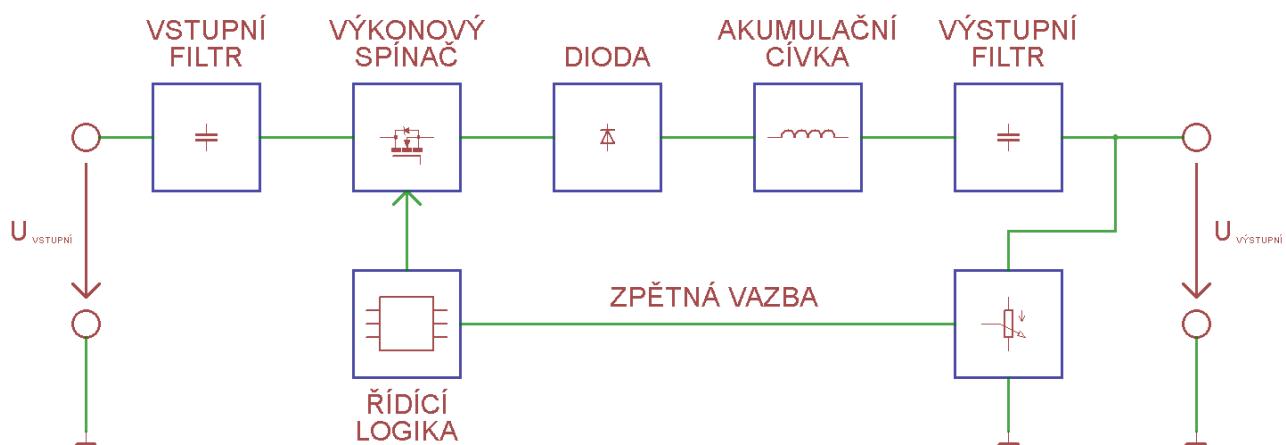
Pro logické části elektroniky je nutné získat odrušené a stabilizované napětí. Jeho hodnota se pohybuje do 5V. Toho můžeme nejlépe dosáhnout dvěma způsoby. Prvním je použití spínaného zdroje, tím druhým je použití lineárního stabilizátoru.

Jako výhodnější varianta se nakonec ukázal lineární stabilizátor, protože jej lze lehce zkonstruovat, nebude generovat rušení a pro malé odběry (na robotovi maximálně 500 mA) se nevyplatí návrh a konstrukce spínaného zdroje.

Spínaný zdroj

Spínané zdroje využívají cívek a kondenzátorů k akumulaci elektrické energie. Na vysokých frekvencích je připojují pomocí elektronického spínače (nejčastěji tranzistorů MOSFET) a odpojují k napájecímu napětí, čímž se v nich akumuluje energie, která je poté čerpána do zátěže. Elektronický spínač je řízen zpětnou vazbou, která snímá výstupní napětí, čímž se zajistí stabilní výstupní napětí.

Pro robota by bylo nejvýhodnější sestavit snižující měnič bez transformátoru, který by byl sestrojen podle blokového schématu, které je níže.



Blokové schéma spínaného snižujícího měniče bez transformátoru

Výhody:

- malé ztráty
- při velkých výkonech malé rozměry

Nevýhody:

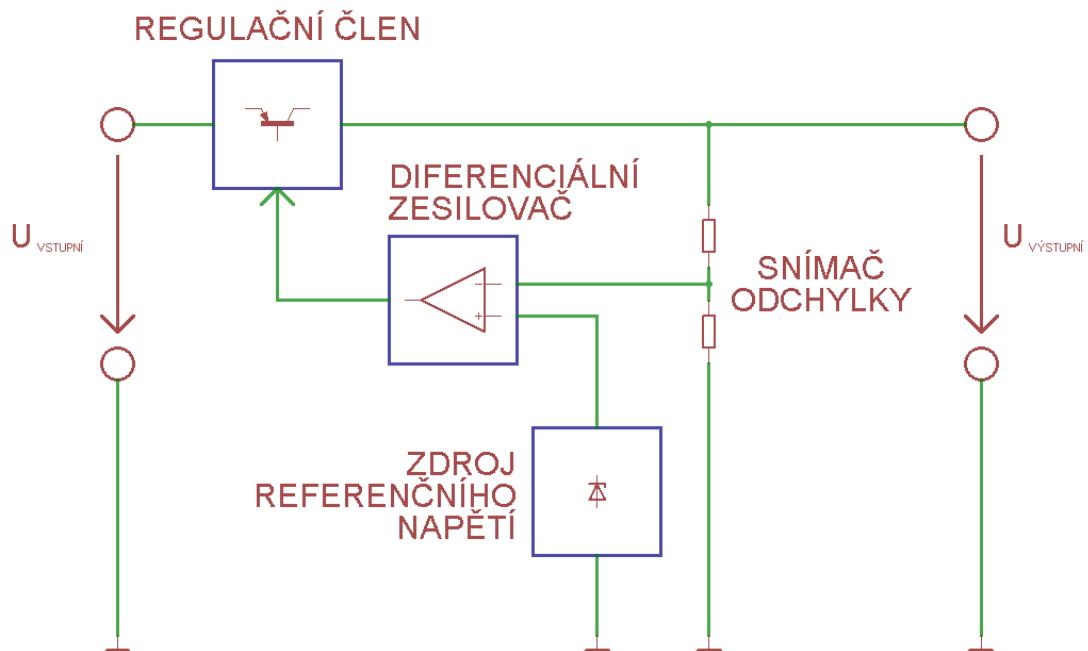
- rušení, které vzniká při spínání
- složité zapojení (oproti lineárním stabilizátorům)

Lineární stabilizátor

Lineární stabilizátory pracují na jednoduchém principu. Skládají se z akčního prvku (nejčastěji bipolárního tranzistoru) a z řídící logiky (diferenciálního zesilovače a zdroje referenčního napětí).

Diferenciální zesilovač sleduje odchylku výstupního napětí a napětí referenčního. Výstup zesilovače budí regulační tranzistor. Dojde-li ke snížení výstupního napětí (např. vlivem zvýšení odběru), zvýší se odchylka a diferenciální zesilovač otevře více tranzistor, čímž se sníží úbytek napětí, který na něm vzniká a výstupní napětí vzroste.

Pevné lineární stabilizátory se dělají např. v třísvorkovém provedení s označením 78XX nebo 79XX (XX označuje výstupní napětí). Nastavitelné lineární stabilizátory jsou označovány např. jako LM317.



Blokové schéma lineárního stabilizátoru

Výhody:

- jednoduchost zapojení
- nevzniká rušení

Nevýhody:

- velké ztráty
- pro velké výkony je nutno chlazení

Pro použití lineárního stabilizátoru jsem navrhl modul LINSTAB01A, jehož dokumentace je uvedena v kapitole 9. "Dokumentace navržených modulů".

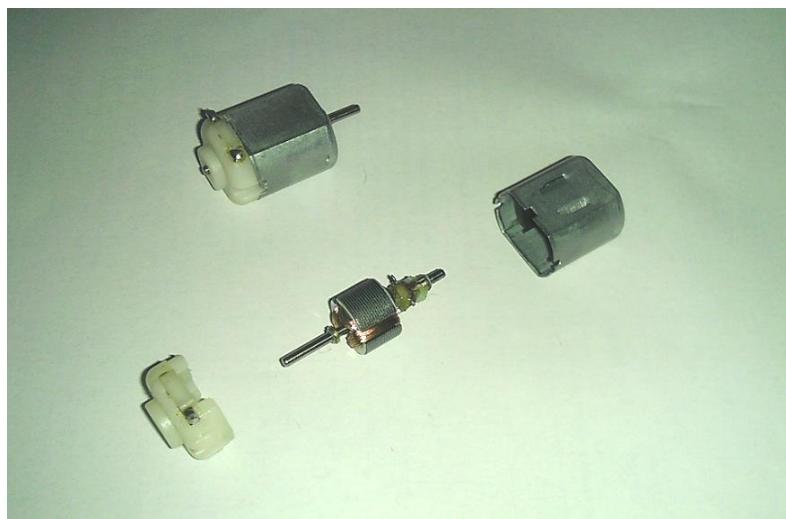
3.2. BUZENÍ MOTORŮ

K tomu, aby se robot mohl pohybovat jsem se rozhodl použít modelářské stejnosměrné motory s převodovkou. Vybral jsem je proto, protože je poměrně jednoduché je řídit, ale také je důležitým faktem, že mají nízkou hmotnost, což umožní konstrukci robota menší hmotnosti, který poté může dosahovat vyšší rychlosti.

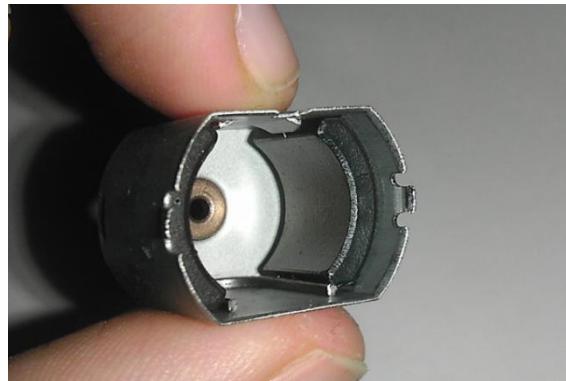
3.2.1. STEJNOSMĚRNÉ KOMUTÁTOROVÉ MOTORY

Stejnosměrný motor se skládá ze dvou důležitých částí. Tou první je stator a druhou rotor. Na statoru jsou umístěny dva opačně orientované magnety, mezi něž je vložen rotor, na kterém je navinuto několik vinutí (u našeho motoru jsou tři). Velmi důležitou částí rotoru je komutátor, který zajišťuje přepínání cívek při chodu motoru.

Pro zmenšení tření je osička motoru umístěna v bronzových kluzných ložiscích, která jsou namazána vazelinou.



Na obrázku výše lze vidět složený a rozložený motorek, které jsou použity v převodovce robota. Na obrázcích na další straně je detail rotoru (vlevo) a statoru (vpravo).



FUNKCE STEJNOSMĚRNÉHO KOMUTÁTOROVÉHO MOTORU

Na vinutí rotoru motoru je přivedeno přes komutátor napětí, které způsobí to, že cívku (představující část vinutí) začne protékat proud, který vytvoří magnetické pole. Poté začne být tato cívka přitahována směrem k magnetu s opačnou polaritou, což způsobí rotační pohyb rotoru. Před momentem, kdy se od magnetu začne cívka oddalovat (vlivem setrvačnosti), přepojí komutátor napětí na další cívku, aby nedocházelo k brzdění pohybu (vlivem magnetické síly, která by zpět přitahovala cívku k magnetu).

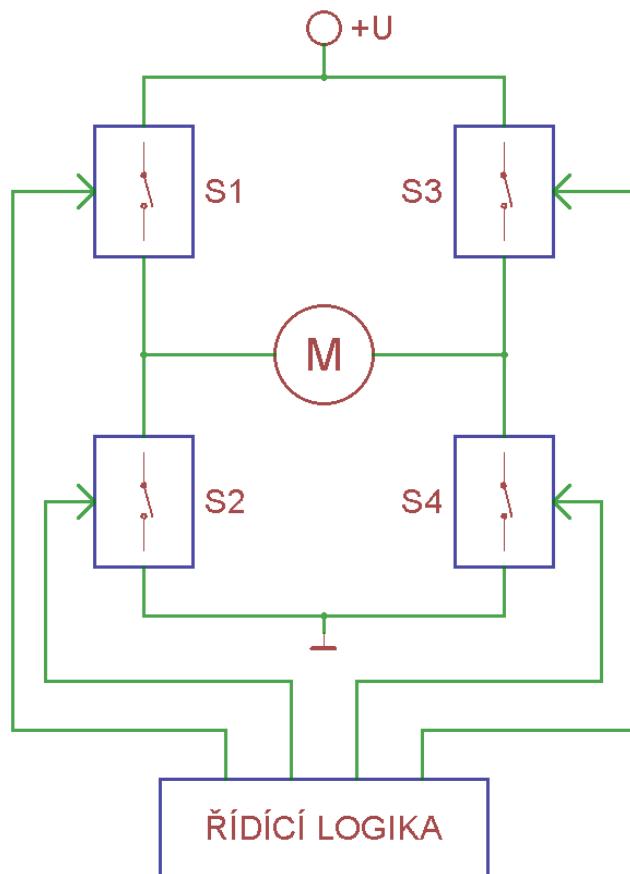
Tento děj se neustále opakuje a to způsobí otáčení hřídelky, na níž je připojena převodovka.

3.2.2. H - MŮSTKY

Vzhledem k tomu, že motory nelze budit přímo z výstupu mikroprocesoru, musíme pro jeho spínání použít minimálně jeden tranzistor. Tímto způsobem však nelze měnit směr otáčení motoru. K tomu slouží tzv. H-můstek.

H-můstek je elektronický obvod, který slouží k přepínání pólů (a tedy i ke změně směru rotace) u stejnosměrných motorů. Dnes se lze setkat s H-můstky realizovaných pomocí bipolárních nebo unipolárních tranzistorů, popř. může být celý můstek (nebo i více můstků) v jednom integrovaném obvodu.

H můstek si můžeme představit realizovaný pomocí čtyř elektronických spínačů, jak jsem naznačil na obrázku níže.



Principielní schéma H-můstku

H-můstky většinou pracují tak, že se spínají vždy dva spínače v jedné diagonále (v našem případě spínače S1 a S4 nebo S2 a S3). Při spínání motoru může dojít k pěti stavům:

1. Motor je zcela odpojen (*žádný ze spínačů není sepnutý*)
 2. Motor se točí jedním směrem (*jsou sepnuté spínače S1 a S4*)
 3. Motor se točí druhým směrem (*jsou sepnuté spínače S2 a S3*)
 4. Motor je brzděn (*spínače S2 a S4*)
 - z motoru je odebírána elektrická energie (*motor je zkratován*)
 5. H-Můstek zkratován (*všechny spínače jsou sepnuté*)
 - nežádoucí stav, spínacími tranzistory tečou velké proudy a může dojít ke zničení H-můstku

ŘÍZENÍ RYCHLOSTI OTÁČENÍ STEJNOSMĚRNÝCH KOMUTÁTOROVÝCH MOTORŮ

Řízení rychlosti motorů lze dosáhnout pomocí pulzní šířkové modulace (PWM = pulse width modulation), která mění střídu signálu.

Lze si ji představit tak, že v průběhu jedné periody motor přepneme ze stavu zapnuto na vypnuto a naopak, přičemž se mění poměr mezi dobou, kdy je motor zapnutý a kdy vypnuty. Vzhledem k vysoké frekvenci těchto impulzů se motor točí plynule. Čím delší dobu je motor v jedné periodě sepnut, tím větší jsou otáčky motoru.

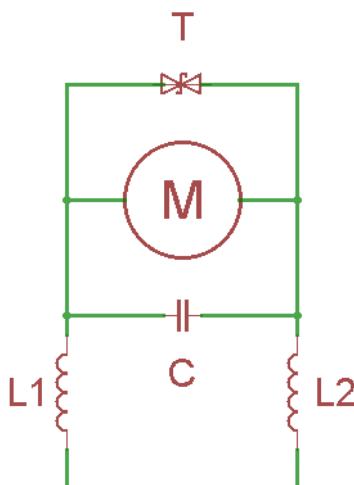
Pro H-můstek jsem navrhl modul HBRDGL29801A, jehož dokumentace je uvedena v kapitole 9. *"Dokumentace navržených modulů"*.

3.2.3. RUŠENÍ

Jelikož je motor indukční zátěž, vzniká při chodu motoru rušení, které může mít za následek buď špatnou funkci řídící jednotky, ovlivnit funkci senzorů a dalších zařízení, ale dokonce může mít za následek i zničení některých prvků.

Při chodu motoru dochází ke vzniku napěťových a proudových špiček, za což může přepínání cívek vinutí komutátorem. Pro jejich odstranění jsem k motoru připojil obousměrný transil, blokovací kondenzátor a dvě cívky.

Transil tvoří přepěťovou ochranu, protože při překročení napětí mezi jeho svorkami dojde k jeho sepnutí a zkratování motoru. Blokovací kondenzátor blokuje menší napěťové špičky, na které ještě transil nereaguje. Dvojice cívek zapojených do série s motorem potlačují vznik proudových špiček.



Odrušení stejnosměrného komutátorového motoru

3.3. SENZORY

Senzory slouží robotovi k tomu, aby získal alespoň základní přehled o okolním prostředí nebo aby sledoval potřebné parametry. Při detekci a měření různých veličin se však setkáváme s problémy, které je potřeba řešit. V našem případě se toto týkalo zejména černé čáry, po níž má robot jezdit.

Pokusy jsem zjistil, že různé materiály různých barev se v infračerveném spektru jeví jinak, než je tomu ve spektru, které vidí člověk. Z tohoto důvodu jsem se rozhodl použít ke sledování čáry optické odrazové senzory, které reagují na viditelné spektrum, místo původních infračervených senzorů.

Toto řešení však přineslo několik komplikací. Je nutné pořídit filtry barev nebo přímo snímací prvky (nejčastěji fototranzistory a fotodiody), které reagují pouze na vhodnou barvu, aby se zamezilo rušení snímání vlivem okolního světla. O řešení jednotlivých senzorů pro snímání čáry se rozepíší v dalších podkapitolách.

Krom detekce čáry je potřeba také zajistit alespoň minimální interakci se člověkem. Z tohoto důvodu jsem na robota přidělal nárazník. K čemu na robotovi je a jak funguje, jsem více rozepsal v podkapitole 3.3.3. "Dotykový senzor".

Použité senzory:	optický řádkový snímač dva odrazové senzory reagující na modré světlo infračervený dálkoměr dva dotykové senzory tvořící nárazník
-------------------------	--

3.3.1. ŘÁDKOVÝ SNÍMAČ

Ke snímání čáry je nejlepší mít k dispozici co nejvíce snímacích bodů. Nejprve jsem chtěl použít fototranzistory s filtry světla, ale ukázalo se, že integrovaný AD převodník v procesoru je velmi pomalý, protože by přečtení deseti senzorů trvalo přibližně 100 ms, což je poměrně dlouhá doba a k tomu je potřeba ještě implementovat vyhodnocení a řízení motorů, a tak by nakonec jeden cyklus programu mohl trvat i 110 ms. Chci upozornit, že se jedná o přibližné hodnoty pro procesor PIC16F877 s taktem 8MHz.

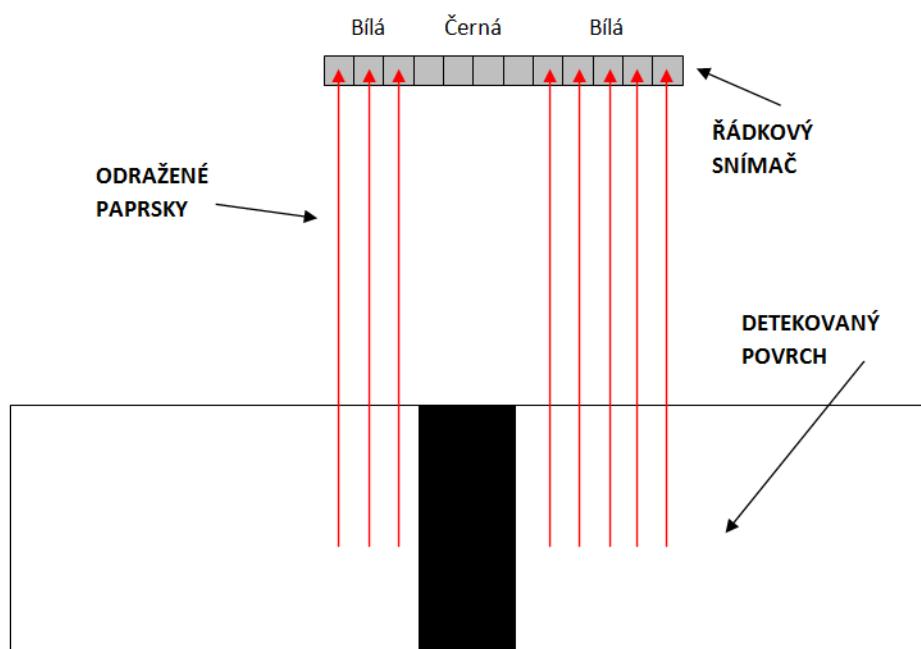
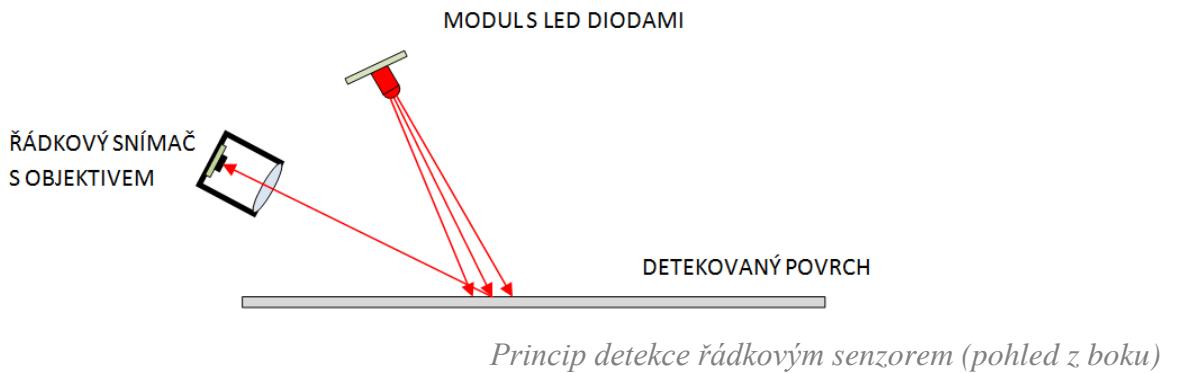
Při prohlížení různých typů senzorů jsem narazil na optické řádkové snímače (linear optical sensor array), který jsem se rozhodl použít. Jedná se v podstatě o zjednodušenou verzi kamery (má méně pixelů a pouze v jedné řadě), kterou jsem již dříve viděl na robotech na sledování čáry používat.

Pro detekci čáry jsem nakonec vybral senzor s označením TSL3301. Jedná se o řádkový senzor, který se skládá ze 102 fotodiód, vyhodnocovací logiky a interního AD převodníku. Senzor pracuje s napájecím napětím 3,0 V - 5,5 V a frekvence hodin může být až 10 MHz. Poté se přečtení celého senzoru včetně zpracování pohybuje v jednotkách ms. Senzor TSL3301 je i s veškerou vyhodnocovací logikou v integrovaném obvodu v pouzdru CL8.

Senzor reaguje na světlo o vlnové délce 300 nm až 1100 nm, ale největší citlivost je na světlo o vlnové délce 700 nm, což odpovídá červenému světlu. Proto byl k tomuto senzoru navržen i modul na osvětlení LEDSENBAR01A. Bohužel se mi nepodařilo sehnat LED diody s vlnovou délkou 700 nm, a tak jsem použil nejbližší hodnotu - 660 nm.

Vzhledem k tomu, že osvětlení snímaného prostoru řeší sám robot, umožňuje to jízdu a správnou funkčnost i za zhoršených optických podmínek, např. v noci při zhasnutém světle.

Při použití tohoto senzoru bylo nutné vyřešit i přenos obrazu na fotodiody. Senzor jsem proto musel opatřit objektivem. K dispozici jsem měl objektiv ze scanneru čárových kódů Genius scanner 4500 A. Bylo nutné zajistit, aby při manipulaci s objektivem a modulem pro senzor TSL3301 nedošlo ke vniku prachu nebo jiných nečistot do prostoru objektivu. Senzor musí být v objektivu také uzavřen tak, aby dovnitř nevnikalo žádné světlo z okolí.



Jak je vidět z obrázku výše, chová se řádkový senzor obdobně jako mnoho bodových senzorů v jedné linii. Pro tento senzor byl navržen modul OLSA01A. Jeho dokumentaci spolu s dokumentací k modulu LEDSENBAR01A najdete v kapitole 9. "Dokumentace navržených modulů".

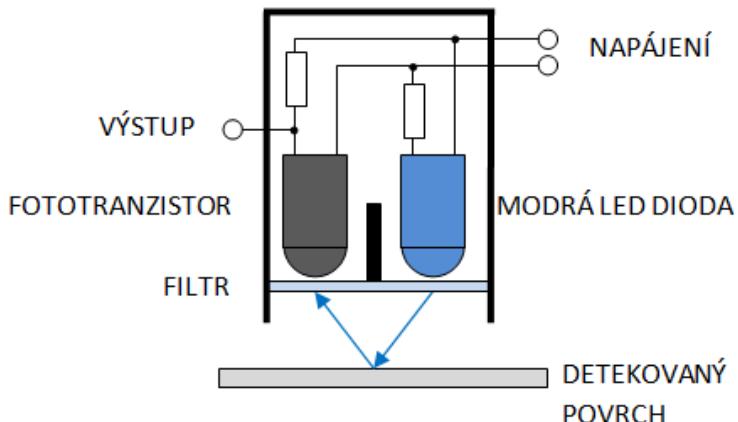
3.3.2. ODRAZOVÝ SENZOR

Sice jsem v podkapitole 3.3.1. "Řádkový senzor" psal o nevýhodnosti použití integrovaného AD převodníku procesoru v kombinaci s použitím odrazových senzorů s analogovým výstupem.

Nicméně i přes to jsem se rozhodl použít dva odrazové senzory s analogovým výstupem na robotovi. Ty slouží v případě, že robot ztratí na hlavním senzoru - TSL3301 obraz nebo nebude vidět čáru. Jedná se v podstatě o záchranný mechanizmus, který robotovi umožní opětovně najít čáru nebo v případě nefunkčnosti optického čárového senzoru, čáru sledovat.

Na robotovi se tedy nacházejí dva záchranné senzory. Pochází z inkoustové tiskárny, kde se používají ke zjištění kvality papíru. K tomu využívají viditelného světla modré barvy. Mechanická konstrukce senzoru je velmi jednoduchá a samotný senzor se skládá pouze z fototranzistoru, odporu, modré LED diody a filtru modrého světla s čočkami. Senzory jsou obaleny černou samolepící páskou, aby se zamezilo pronikání okolního světla k fototranzistoru. Zároveň izolep slouží jako stínítko pro detekční prostor.

Detekce probíhá tak, že se měří napětí na děliči tvořeným z fototranzistoru a odporu. Mechanické uspořádání a princip detekce je znázorněný na obrázku dole.



Konstrukce a princip detekce záchranného odrazového senzoru

3.3.3. DOTYKOVÝ SENZOR

Dotykový senzor je na robotovi implementován ze dvou důvodů. Prvním je detekce nízkých překážek. Druhým důvodem je interakce s člověkem, protože při zapnutí robota si lze vybrat režim sledování čáry nebo diagnostiku pro ověření funkčnosti senzorů a motorů.

Jeden programový cyklus spustí diagnostiku, kdy dochází k výpisu všech senzorů na USB a lze pomocí toho zjistit, zda je vše v pořádku a na základě těchto dat odhalit např. vzniklou závadu. Druhý programový cyklus spustí proceduru sledování čáry. O těchto cyklech se podrobněji rozepíší v kapitole 4. "Řídící program".

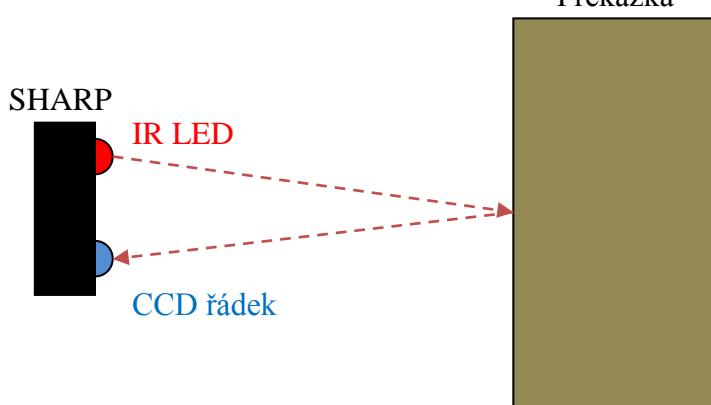
Dotykový senzor je proveden jako nárazník v přední části robota. Skládá se ze dvou mikrospínačů, z nichž každý je na jedné straně robota, a dvou pull-up rezistorů, které přivádí na senzor napětí 5 V. Spínací tlačítka jsou spojeny s plechovým profilem, který tvoří nárazník. Tato konstrukce umožňuje zjistit, zda se jedná o čelní náraz nebo zda robot pouze o něco zavadil na jedné ze strany.

Pokud spínač není sepnutý (tj. není stisknutý nárazník), je na výstupu senzoru 0 V - logická 0 (potenciál GND). V případě, že dojde k sepnutí, přivede se přes odpór na výstup napětí 5 V - logické 1. Výstup je tedy kompatibilní s logikou TTL i CMOS. Výstup ze senzoru je přímo připojen na digitální vstup mikroprocesoru.

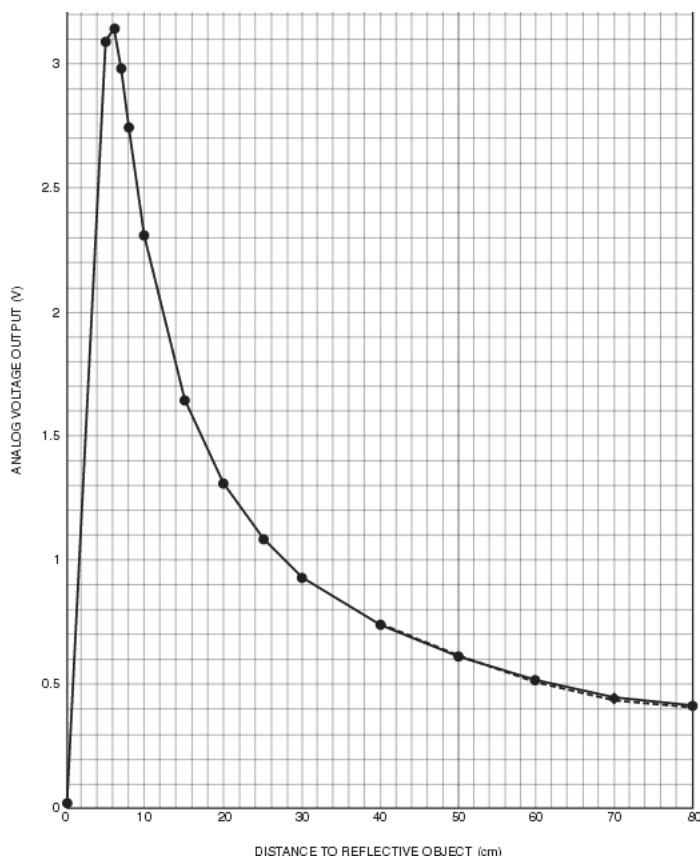
3.3.4. INFRAČERVENÝ DÁLKOMĚR

Před robotem se mohou na dráze objevit různé překážky, které je nutné včas detektovat, aby robot nenarazil a nepoškodil se. K tomuto účelu jsem vybral senzor SHARP GP2Y0A21YK.

Senzor se skládá z infračervené LED diody, CCD řádky a řídící a vyhodnocovací logiky. Infračervená dioda slouží jako vysílač infračervených paprsků, které se poté odrážejí od překážky a dopadají na CCD řádku, jak je znázorněno na obrázku níže.



Vyhodnocovací logika poté na základě triangulace přepočte změrenou vzdálenost na napětí. Proto má senzor pouze tři vývody, z nichž dva jsou napájecí a třetí je analogový, kde výstupní napětí odpovídá vzdálenosti překážky a pohybuje se v mezích 0 - 3,3 V. Závislost výstupního napětí není lineární, ale má tvar exponenciální.



Výrobce udává, že se senzorem lze měřit ve vzdálenosti 4 - 30 cm, ale praktické pokusy ukázaly, že vzdálenost, kde senzor pro naše účely ještě spolehlivě měří, je přibližně dvojnásobná (50 - 60 cm).

3.4. VÝBĚR PROCESORU

Po dokončení návrhu jednotlivých modulů robota bylo potřebné zvolit vhodný procesor. Plně dostačující bylo použití 8 bitového mikrokontroleru. V dnešní době na tomto poli dominuje firma Atmel s procesory ATmega a firma Microchip s procesory PIC.

K dispozici jsem měl procesor PIC16F887 (*modul PIC16F87xTQ4401B*) [18] s USB programátorem PicKit2 (*modul PICPROGUSB02A*) [17] a procesor ATmega 328 (*modul ATmegaTQ3201A*) [19] s převodníkem USB na RS232 (*modul USB232R01B*) [20].

Předem podotykám, že ani jeden z výše zmíněných modulů v kapitole 3.4. "Výběr procesoru" není mým návrhem a pouze jsem použil součásti ze stavebnice MLAB, na jejichž stránkách lze najít kompletní dokumentaci k těmto modulům.

Oba procesory jsou si velmi podobné, jak po stránce paměti, tak rychlosti. Nakonec jsem zvolil procesor PIC16F887, protože mám k dispozici kvalitní vývojové prostředí PCW PICC Compiler, jemuž nemůže žádné z vývojových prostředí pro ATmega konkurovat.

Sice je jistou alternativou kit Arduino [22], který lze pomocí modulu procesoru a USB/RS232 převodníku lehce realizovat [21]. Poté by se procesor mohl programovat v jazyce Wiring [22], který vychází z jazyka C, jeho nevýhodou však je obrovské plýtvání prostředky procesoru (oproti čistému C) a tak i výsledná menší rychlosť chodu programu.

3.4.1. PROCESOR PIC16F887

Procesor PIC16F887 [16] se doporučuje napájet napětím 3,3 až 5,0 V. Pro taktování lze použít nastavitelný interní oscilátor (v méém případě je nastaven na 8 MHz), ale lze připojit i externí oscilátor s frekvencí až 20 MHz. Cena tohoto procesoru se pohybuje kolem 90 Kč a celý modul vyjde přibližně na 200 Kč.

Procesor má 44 vývodů, z nichž 4 slouží jako napájení a další 4 nejsou použity. Pro uživatele tedy zbývá 36 vývodů. Jak tomu bývá u mikrokontrolerů, je řada těchto vývodů použita pro několik různých funkcí (mikrokontrolery v pouzdru mají krom procesoru i další pomocné obvody, jako je např. AD převodník, časovač a komparátor).

Procesor je založen na harvardské architektuře, která předpokládá oddělené prostory paměti pro program a data. Proto se paměť skládá z paměti typu FLASH (112 KB), EEPROM (256B) a RAM (368B), z něhož paměť FLASH a EEPROM jsou datové oblasti pro uložení programu a nelze k nim přistupovat přímo, ale pouze pomocí speciálních registrů, což slouží jako ochrana proti nechtěnému přepsání programu.

PIC16F887 patří mezi procesory architektury RISC, což znamená, že mají redukovanou instrukční sadu. Návrh této instrukční sady poté počítá s jednoduchými strojovými instrukcemi (např. neexistuje strojová instrukce pro násobení nebo dělení), které jsou však implementovány hardwarově, což značně zvyšuje rychlosť provádění těchto instrukcí.

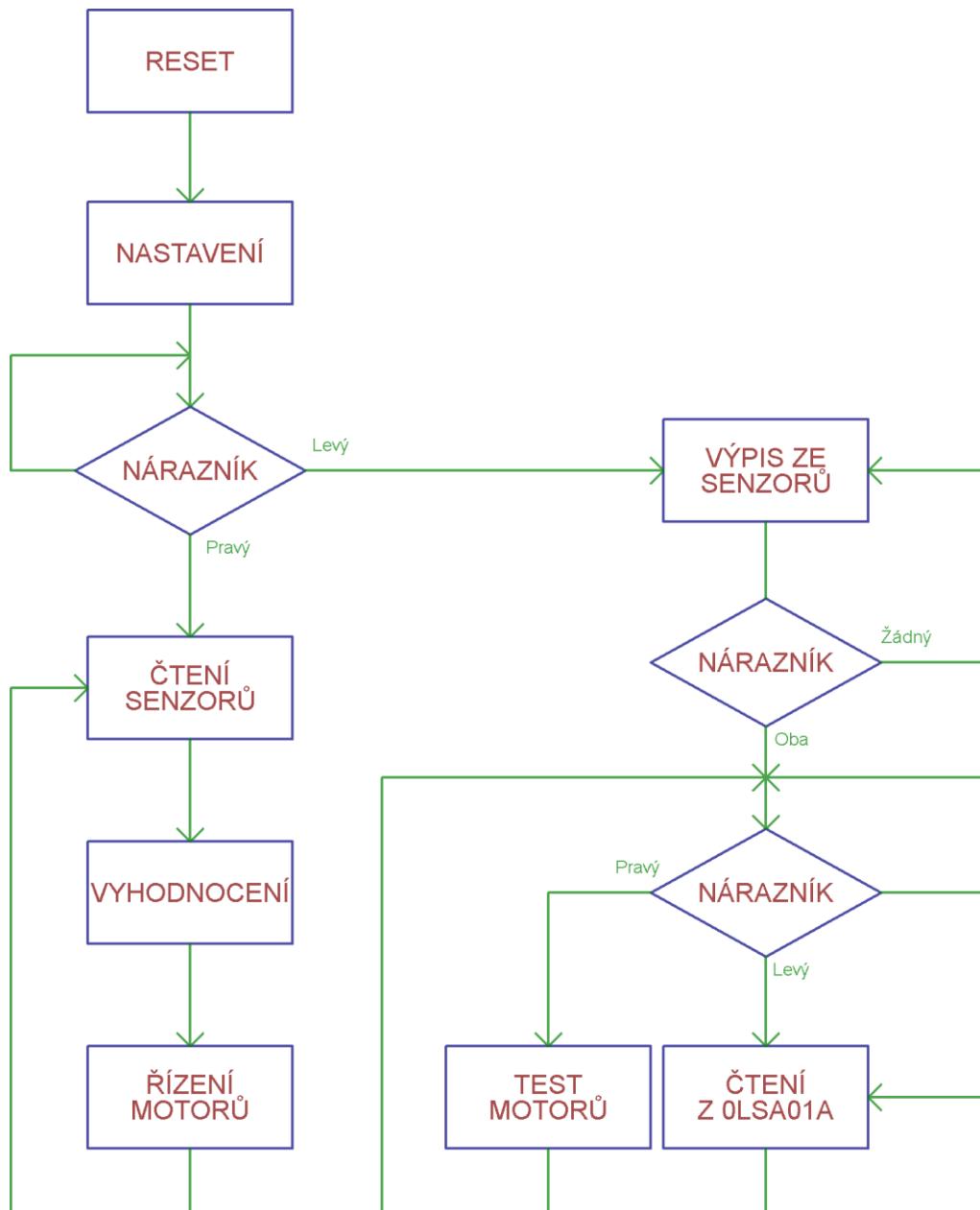
Program pro tento procesor byl napsán ve vývojovém prostředí PIC C Compiler od firmy CCS. Program je v jazyce C. Podrobnější popis a rozbor samotného programu je v kapitole 4.2. "Program". Po zkompilování programu se program do procesoru nahrál přes programátor PICPROGUSB02A a softwaru od firmy Microchip PicKit2.

4. ŘÍDÍCÍ PROGRAM

Program byl napsán v jazyce C ve vývojovém prostředí PIC C Compiler od firmy PCW. Zkomplikovaný program jsem do robota nahrál pomocí programátoru PICPROGUSB02A, kompatibilním s programátorem PICkit2 firmy Microchip.

4.1. PROCESNÍ DIAGRAM

Níže vidíte procesní diagram programu. Nezobrazuje chod jednotlivých kroků programu, ale pohlíží na program jako takový a ukazuje jeho základní strukturu. Podrobnější popis obsluhy senzorů a jejich vyhodnocení a řízení motorů bude v dalších kapitolách.



Jak je vidět z vývojového diagramu výše, po resetování procesoru dojde ke nastavení registrů procesoru, ale i jeho periferií. Poté se čeká na výběr režimu.

Při nastavení se povolují interní AD převodníky a definují analogové vstupy. Dále se nastavují časovače a režim CCP jednotky umožňující generování PWM signálu pro motory. Zakazuje se zde použití komunikace přes SPI a použití interních komparátorů. Po sdělení všech údajů procesoru se provádí reset a nastavení optického řádkového senzoru, vypnou se oba motory a zhasnou indikační diody. Poté robot pípne a pošle zprávu po RS232.

Prvním režimem je diagnostika, která se vyvolá stisknutím levého nárazníčku. V tomto režimu procesor čte údaje ze senzorů, vyhodnocuje pozici čáry a hodnoty posílá na sériovou linku, ale zároveň kontroluje i stav nárazníku. Po stisknutí obou tlačítek nárazníku si lze vybrat mezi testem motorů nebo úplným výpisem ze senzoru OLSA01A. Z praktických důvodů není možné diagnostiku opustit, a tak se změna režimu provádí resetováním procesoru.

Test motorů zajistí, že každý motor sám začne postupně zrychlovat na maximální hodnotu a poté postupně zpomalí. Nejdříve se motor točí dopředu, poté dozadu. Jako první se testuje levý motor a jako druhý motor pravý. Po ukončení lze spustit opět test motorů nebo čtení ze senzoru OLSA01A.

Čtení ze senzoru OLSA01A zajistí, že se do paměti procesoru uloží naměřené hodnoty všech 102 pixelů. Ty se následně vypíší v hexadecimálním formátu na sériovou linku. V praxi se ukázalo lepší, když procesor vypisuje hodnoty kontinuálně a co nejrychleji. Proto v tomto režimu není možné provést jinou akci a pro opětovné spuštění diagnostiky je potřeba procesor resetovat.

V režimu sledování čáry procesor kontroluje všechny senzory, provádí jejich vyhodnocení a podle získaných údajů nastavuje motory tak, aby se černá čára držela uprostřed čtené řádky, tedy, čím více se čára odkládá od středu, tím více robot zatačí. Aby při sledování čáry nedošlo ke spuštění diagnostiky, je potřeba pro změnu režimu resetovat procesor.

4.2. POPIS PROGRAMU

Nyní rozeberu nejdůležitější části programu, které se týkají čtení senzorů a jejich vyhodnocení, indikátorů a řízení motorů. Celý program je uložen na přiloženém CD.

4.2.1. INDIKACE

Na robotovi je umístěno několik indikačních prvků. Ty slouží k jednoduchému zjištění zvoleného režimu či upozornění. Jedná se o dvě červené SMD LED diody a piezo sirénku.

LED diody

Led diody se obsluhují stejným způsobem jako digitální výstupy, jen je potřeba mít na paměti, že pokud nastavím výstup do stavu logické 0, bude dioda svítit a pokud do logické 1, dioda bude zhasnutá, protože je připojena anodou na zdroj napětí a katodou na výstup procesoru.

Obsluha vývodů poté probíhá následovně:

```
output_high(LED1);           // zhasne LED1
output_low(LED2);            // rozsvíti LED2
```

Piezo sirénka

Aby sirénka udělala delší zvuk - pípnutí, je potřebné, aby na ní bylo přivedeno proměnné napětí. K tomuto účelu jsem vytvořil funkci *beep*, kterou lze vidět na další straně.

```
void beep(int16 period, int16 length)
{
    int16 bp;                                // promenna pro nastaveni delky
    for(bp=length;bp>0;bp--)                  // prepina vystupy tolikrat, jakou jsme zadali delku
    {
        output_high(SOUND_HI);
        output_low(SOUND_LO);
        delay_us(period);
        output_high(SOUND_LO);
        output_low(SOUND_HI);
        delay_us(period);
    }
}
```

Jako parametr funkce slouží dvě proměnné. Proměnná *period* určuje výslednou frekvenci tónu a proměnná *length* délku pípnutí (určuje, kolikrát proběhne cyklus). Cyklus *for* opakuje sled instrukcí, které ve výsledku mění polaritu napětí přivedeného na sirénku.

4.2.2. ŘÍZENÍ MOTORŮ

Obsluha motorů vyžaduje nastavit směr otáčení a rychlosť. K tomu slouží příkazy s nastavením logické úrovně na výstupu (*output_low(PIN)* / *output_high(PIN)*). Rychlosť se nastavuje pomocí PWM, k čemuž je nutné nastavit časovač číslo 2, PWM povolit a nastavení hodnoty použít příkaz *set_pwm_duty(RYCHLOST)*.

```
void l_motor_fwd(int8 speedl)           // Levy motor dopredu
{
    output_high(LMF);
    output_low(LMB);
    set_pwm2_duty(speedl);
}

void l_motor_bwd(int8 speedl)           // Levy motor dozadu
{
    output_high(LMB);
    output_low(LMF);
    set_pwm2_duty(speedl);
}

void r_motor_fwd(int8 speedr)           // pravy motor dopredu
{
    output_high(RMF);
    output_low(RMB);
    set_pwm1_duty(speedr);
}

void r_motor_bwd(int8 speedr)           // pravy motor dozadu
{
    output_high(RMB);
    output_low(RMF);
    set_pwm1_duty(speedr);
}
```

Krom těchto čtyř funkcí jsou nadefinovány i funkce *l_motor_off()* a *r_motor_off()*. Ty nastaví všechny řídící signály do hodnoty logické 0 (včetně PWM), čímž se zajistí zastavení motorů.

4.2.3. ČTENÍ ANALOGOVÝCH SENZORŮ

Pro používání analogových vstupů je potřeba je nastavit. Musí se definovat, které vývody budou sloužit jako analogové vstupy a povolit interní oscilátor pro interní AD převodník. Tato deklarace probíhá v nastavení.

```
setup_adc_ports(sAN0-sAN1-sAN2);      // aktivní analogové vstupy RA0, RA1 a RA2
setup_adc(ADC_CLOCK_INTERNAL);         // interní hodiny pro AD prevodník
```

Nyní lze začít číst ze senzorů hodnoty. To jsem udělal tak, že jsem vytvořil funkci, do níž jsem vložil příkazy, které uloží výstup z interního AD převodníku do globální proměnné *line_l* nebo *line_r*. Konstanty *LINEL* a *LINER* označují číslo analogového vstupu. Pauzy 10 us zajišťují, že proběhne celý převod.

```
void read_blue_sensors()           // cteni nouzovych senzoru
{
    set_adc_channel(LINEL);        // cti Levy nouzovy senzor
    delay_us(10);
    line_l=read_adc();            // cti pravy nouzovy senzor
    set_adc_channel(LINER);
    delay_us(10);
    line_r=read_adc();
}
```

Pokud robot stojí na černé, je výstupní hodnota blízká 0, pokud na bílé, je v proměnné uloženo číslo blízké 255. Následné vyhodnocení probíhá velmi jednoduše pomocí podmínky *if*, která porovnává konstantu *TRESHOLD* (rozhodující hodnota) s hodnotou, která se uložila do proměnných.

Pokud přejede robot černou čáru, dojde k prudkému zatočení, aby se čára dostala zpět mezi záchranné senzory (a tedy i do zorného úhlu rádkového snímače) a zapamatuje si pozici, kde byla čára viděna naposledy. Řídící program rozeberu později.

4.2.4. PRÁCE S OPTICKÝM ŘÁDKOVÝM SNÍMAČEM

Pro komunikaci se rádkovým senzorem se používá sériová linka. Tu má sice procesor integrovanou hardwarově, ale je zde sloučen sériový vstup a hodinový signál, tedy, procesor neumí pracovat se sériovou synchronní linkou, kterou optický rádkový senzor vyžaduje.

Proto jsem musel synchronní sériový přenos naprogramovat. Slouží k němu tři vývody, v programu pojmenované jako *SCLK*, *SDIN* a *SDOUT*. Toto označení je shodné s označením vývodu modulu OLSA01A, a tak *SDIN* slouží jako výstup (z pohledu procesoru) a *SDOUT* jako vstup dat (z pohledu procesoru). *SCLK* je hodinový signál, který generuje procesor na základě programu.

Informace, které procesor posílá, jsou uloženy do polí. V každém poli je uloženo osm bitů řídící instrukce. Pro odesílání dat po sériové lince bylo vytvořeno několik funkcí. Jedná se o funkce *olsa_pulse()*, *olsa_pulses(POČET)* a *olsa_send(INFORMACE)*. Pro čtení ze senzoru jsem napsal funkci *read_olsa()*.

OLSA_PULSE()

Jedná se o velmi jednoduchou funkci. Jejím cílem je pouze zpřehlednit program komunikace. Vytváří jeden impulz na hodinovém výstupu SCLK.

```
void olsa_pulse()           // vytvorí jeden impulz
{
    output_high(SCLK);
    output_low(SCLK);
}
```

OLSA_PULSES(POČET)

Tato funkce je velmi podobná funkci OLSA_PULSE(). Jejím cílem je vytvořit na výstupu hodinového signálu SCLK počet impulzů, který je uveden v závorce. Byla vytvořena proto, že některé příkazy po zaslání potřebují více impulzů, než se provedou. Jedná se např. o příkazy *reset*, *start_integration* a *stop_integration*.

Pro provedení správného počtu impulzů slouží v této funkci cyklus *for*.

```
void olsa_pulses(int count)           // vytvorí impulzy pro ridici logiku
{
    int8 ct;
    for(ct=0;ct<=count;ct++)
    {
        output_high(SCLK);
        output_low(SCLK);
    }
}
```

OLSA_SEND(PŘÍKAZ)

Funkce *olsa_send(příkaz)* je jedna ze složitějších funkcí pro obsluhu senzoru. Zajišťuje odeslání zprávy do senzoru. Zpráva je uložena v osmibitovém poli, které je celé prohlédnuto a podle hodnoty, které jsou uloženy v buňkách (0 a 1) je nastaven výstup procesoru. Jako první se odesílá nejméně významný bit.

Zprávy a obsah, které lze do senzoru posílat určuje výrobce v dokumentaci k integrovanému obvodu TSL3301. Podrobněji jsem se o práci s tímto obvodem rozepsal v jeho dokumentaci.

Krom prohlédnutí pole zajišťuje i odeslání start bitu a stop bitu a synchronizaci (po každém odeslání osmibitové zprávy je důležité nastavit výstup *SDIN* na logickou jedničku a hodinový výstup *SCLK* na logickou nulu).

```
void olsa_send(int8 info[8])           // USART komunikace s modulem OLSA01A - poslani zpravy
{
    int *ip;                           // ukazatel na pole s informaci
    int1 i;                            // pomocna promenna pro nastaveni 0 nebo 1 na SDIN
    output_low(SDIN);                 // start bit
    olsa_pulse();
    for(ip=0;ip<8;ip++)
    {
        i=info[ip];                  // ziskani hodnoty z pole
        if(i==1)                      // vyhodnoceni obsahu informace - nastav 1
        {
            output_high(SDIN);
        }
        else                          // vyhodnoceni obsahu informace - nastav 0
        {
            output_low(SDIN);
        }
        olsa_pulse();
    }
    output_high(SDIN);                // stop bit
    olsa_pulse();
}
```

OLSA_RESET()

Tato funkce se provede pouze jednou v nastavovací části programu. Jejím úkolem je vyresetovat senzor a připravit jej k použití poté, co bylo připojeno napájení, protože mohlo dojít k hazardním stavům v senzoru a čtení z něj by nemuselo korektně fungovat.

Vyresetovaní se provede tak, že se vstup senzoru *SDIN* vynuluje a pošle se třicet hodinových impulzů na vstup *SCLK*. Poté je vhodné provést synchronizaci start bitu. Ta se uskuteční tak, že se vstup *SDIN* nastaví na logickou 1 a pošle se deset hodinových impulzů na hodinový vstup *SCLK*.

Dále je potřeba vynulovat vnitřní registry. To se provede odesláním patřičné zprávy do senzoru.

```
void olsa_reset()                                // hlavní RESET - provádí se po zapnutí
{
    output_low(SDIN);
    output_low(SCLK);
    olsa_pulses(30);                            // reset radikového senzoru
    output_high(SDIN);
    olsa_pulses(10);                            // start bit - synchronizace
    olsa_send(MAIN_RESET);
    olsa_pulses(5);                            // vycistí mode registr
    olsa_send(SET_MODE_RG);                     // vycistí mode registr
    olsa_send(CLEAR_MODE_RG);
}
```

OLSA_SETUP()

Po provedení resetu je potřeba opět nastavit všechny registry a zisk senzoru, který určuje citlivost - délku expozice. Hodnoty, které lze do registrů zapsat opět uvádí výrobce v dokumentaci k senzoru. Nastavení se provede tak, že se do senzoru odešle sled patřičných zpráv, jak můžete vidět níže.

Tato funkce se v programu volá pouze v nastavovací části, neboť pro mé účely není potřeba při běhu programu měnit nastavení senzoru. Praktické využití by však bylo možné, pokud by robot měl ještě senzor na detekci světla okolního prostředí, kdy by mírou expozice mohl kompenzovat úbytek světla (např. v tunelu, který bývá častou překážkou v soutěžích nebo při poruše osvětlovacího modulu).

```
void olsa_setup()                                // kompletní nastavení, provádí se po resetu
{
    olsa_send(LEFT_OFFSET);                      // nastavení levého segmentu (offset a zisk)
    olsa_send(OFFSET);
    olsa_send(LEFT_GAIN);
    olsa_send(GAIN);
    olsa_send(MID_OFFSET);                      // nastavení středního segmentu (offset a zisk)
    olsa_send(OFFSET);
    olsa_send(MID_GAIN);
    olsa_send(GAIN);
    olsa_send(RIGHT_OFFSET);                     // nastavení pravého segmentu (offset a zisk)
    olsa_send(OFFSET);
    olsa_send(RIGHT_GAIN);
    olsa_send(GAIN);
}
```

OLSA_INTEGRATION()

Funkce *olsa_integration()* krom odeslání příkazu o začátek měření - expozice, zasílá i požadavek o jeho ukončení. Jak lze vidět v dokumentaci od výrobce, je potřeba po poslání příkazu *start_integration* poslat 22 hodinových impulzů, kdy senzor měří a zapisuje hodnoty do registru. Po odeslání příkazu *stop_integration* je potřeba poslat 5 hodinových impulzů, kdy senzor připraví data k odeslání.

```
void olsa_integration()           // snimani pixelu
{
    olsa_send(START_INT);        // zacatek integrace senzoru
    olsa_pulses(22);
    olsa_send(STOP_INT);         // konec integrace senzoru
    olsa_pulses(5);
}
```

READ_OLSA()

Na začátku funkce odešle příkaz o naměření *olsa_integration()* a posléze příkaz na odeslání naměřených hodnot *olsa_send(readout)*. Dále bude zajišťovat příjem dat a jejich uložení.

Cyklus *do* zajišťuje, že se přečtou a uloží všechny hodnoty pixelů. Aby nedošlo ke čtení v momentě, kdy ještě nejsou na vstupu *SDOUT* platná data, je zde podmínka čekání na start bit a pokud nepřichází, odešle se jeden impulz na hodinový výstup *SCLK*.

Dále následuje cyklus *for*, který zajistí přečtení všech osmi bitů. Jednotlivé byty se uloží do bytu tak, že se provede operace logického součtu *or*, podle přijatého bitu buď s hodnotou 1 nebo 0 a následně posunutí pixelu.

Po získání celého bytu se jeho hodnota zapíše do pole hodnot. V závislosti na pořadí pixelu (0 až 101) se zapíše buď do pole pro levou nebo pravou část řádky, kde v každém může být 52 hodnot.

Toto řešení bylo zvoleno proto, protože maximální adresovatelná velikost u procesoru PIC16F887 je 96 bytů, což je nedostatečné pro uložení celé řádky. Sice by byla možnost vypustit několik pixelů, např. tři pixely z každé strany, ale tím by jsme se v podstatě ochudili o šest senzorů, představíme-li si řádku jako 102 bodových senzorů.

Po této funkci může následovat vyhodnocení polohy čáry a nastavení - řízení chodu motorů. Program funkce *read_olsa()* je na další straně.

```

void read_olsa()
{
    int8 cpixel;                                // pocet prectenych pixelu
    int8 cbit;                                  // pocet prectenych bitu
    int8 pixel;                                 // hodnota precteneho pixelu
    cpixel=0;
    lp=0;
    rp=0;
    olsa_integration();
    olsa_send(READOUT);
    do                                         // precte 102 pixelu
    {
        if(!SDOUT)                            // zacatek prenosu - zachycen start bit
        {
            pixel=0;
            for(cbit=0;cbit<8;cbit++) // cte jednotlive bity (8 bitu - 0 az 7)
            {
                olsa_pulse();                  // impulz pro generovani dalsiho bitu
                if(SDOUT)                      // zachycena 1
                {
                    pixel|=1;                 // zapise do bitu pixelu 1 - OR
                }
                else                           // zachycena 0
                {
                    pixel|=0;                 // zapise do bitu pixelu 0 - OR
                }
                pixel<<=1;                   // posune pixel
            }
            olsa_pulse();                  // generuje stop bit
            if(cpixel<52)                // ulozeni do pole
            {
                olsa_lseg[lp]=pixel;       // Leva polovina radky - leve pole
                lp++;
            }
            else
            {
                olsa_rseg[rp]=pixel;       // prava polovina cary - prave pole
                rp++;
            }
            cpixel++;
        }
        else
        {
            olsa_pulse();              // generuje start bit, nebyl-li poslan
        }
    }
    while(cpixel<102);                         // precte 102 pixelu
}

```

4.2.5. VYHODNOCENÍ

Po přípravení všech potřebných funkcí a vyzkoušení funkčnosti senzorů, jsem začal psát program pro vyhodnocení polohy čáry. Jakožto primární senzor na snímání slouží právě OLSA01A, a tak dva bodové senzory používám pouze jako ujištění, že robot nepřejel čáru, nebo v případě, že se ztratila čára z optického rádkového senzoru.

OLSA_POSITION()

Úkolem této funkce je uložit do globální proměnné *position* hodnotu, která odpovídá poloze čáry na senzoru. To je provedeno tak, že funkce prohlíží dvě pole, přičemž v jednom jsou uloženy hodnoty levé poloviny řádky a ve druhém poloviny pravé.

Vyhodnocení je celkem jednoduché. Pokud funkce narazí na jistý počet po sobě jdoucích pixelů, jejichž hodnota odpovídá černé, lze s jistotou tvrdit, že jsme našli čáru. Pokud se pozici čáry nepovede z hodnot vyhodnotit, např. pokud čára zmizela ze zorného úhlu, uloží se do globální proměnné *position* hodnota 0.

Podle potřeby lze měnit v části programu, kde jsou uloženy konstanty, rozhodovací úroveň odpovídající černé barvě (uložena pod názvem *OLSALEV*) nebo počet po sobě jdoucích pixelů, které slouží k detekci čáry (konstanta *LINEPX*). V mém případě používám *OLSALEV=0x60* a *LINEPX = 4*.

```
void olsa_position() // vyhodnoti pozici cary
{
    int8 searchp; // ukazatel na pole
    int8 search; // ulozeni prectene hodnoty
    int8 protect_count; // opravdu vidime caru
    position=0; // nuluje pozici, pokud cara není, ulozena 0
    for(searchp=0;searchp<52;searchp++)
    {
        search=olsa_lseg[searchp]; // vybira pixel
        if(search==OLSALEV) // cerna nebo bila?
        {
            protect_count++; // pokud nasleduje cerna, pricte 1 k poctu cernych pixelu
        }
        else
        {
            protect_count=0; // pokud nasleduje bila, pocet cernych pixelu vynuluje
        }
        if(protect_count>LINEPX) // vidim caru
        {
            position=searchp; // zapis presnou pozici
            line_sector=LEFT; // cara je v leve polovine
            searchp=55; // ukonci hledani
        }
    }
    for(searchp=0;searchp<52;searchp++) // prohlizi pravou cast cary
    {
        search=olsa_rseg[searchp]; // vybira pixel
        if(search==OLSALEV) // pokud nasleduje cerna, pricte 1 k poctu cernych pixelu
        {
            protect_count++; // pokud nasleduje bila, pocet cernych pixelu vynuluje
        }
        else
        {
            protect_count=0; // pokud nasleduje bila, pocet cernych pixelu vynuluje
        }
        if(protect_count>LINEPX) // vidim caru
        {
            position=(searchp+50); // zapis presnou pozici
            line_sector=RIGHT; // cara je v prave polovine
            searchp=55; // ukonci hledani
        }
    }
}
```

4.2.6. ŘÍDÍCÍ ČÁST PROGRAMU A DIAGNOSTIKA

Na rozdíl od jednotlivých funkcí jsem se rozhodl, že zde nebudu uvádět stejně podrobně diagnostiku ani řídící program, neboť by zabral několik stran, čímž by se tato práce znepřehlednila. Celý program přikládám na CD nosiči, kde je podrobně okomentován, a tak umožní zorientování v jeho funkci.

Řídící program i diagnostika přesně odpovídá vývojovému diagramu a stručnému popisu v kapitole 4.1 "*Procesní diagram*".

Na začátku programu se do paměti uloží předchozí pozice robota. V paměti je uloženo posledních pět poloh robota, které se použijí pro výpočet PID regulátoru. Poté následuje čtení všech senzorů a jejich vyhodnocení. Ze současné pozice čáry se spočítá chyba, která se použije pro výpočet v PID regulátoru.

PID regulátor má dominantní složku P. Zavedení složky I zajistilo plynulejší jízdu po rovině a zavedení složky D se projevilo rychlejším průjezdem zatáček a lepší odezvou na rychlé změny polohy čáry vůči robotovi. Regulátor ukládá do globální proměnné *reg_out* absolutní hodnotu, tedy nerozlišuje rozdíl mezi regulací, pokud je čára v levé nebo v pravé polovině rádkového senzoru. Tento způsob se ukázal jako výhodnější.

Pro nastavení motorů je proto nutné reagovat i na pozici čáry. Pokud se čára bude nacházet vlevo, musí pravý motor mít vyšší otáčky a naopak, pokud čára bude napravo, musí mít vyšší otáčky levý motor. To se zajistí porovnáním polohy čáry s konstantou 50, která odpovídá středu.

Bodové senzory, které jsou nasvíceny modře, se aktivují až v případě, že se několikrát po sobě po vyhodnocení polohy z optického rádkového senzoru vrátila hodnota 0, která oznamuje, že se čára ztratila ze zorného úhlu senzoru. V tomto případě, pokud robot najede na některý ze dvou bodových senzorů, bude pozice, která odpovídá nule nahrazena imaginární pozicí, tedy, pokud robot najede na čáru levým senzorem, bude pozice rovna jedné (maximální výchylka vlevo), a pokud najede na čáru senzorem pravým, bude pozice rovna hodnotě 99 (maximální výchylka vpravo).

Odezva PID regulátoru a část s nastavením motorů poté zajistí, že se jejich rychlosť nastaví tak, aby se čára vrátila zpět do zorného úhlu senzoru.

Pokud však robot dálkoměrem zachytí překážku, mikroprocesor bude ignorovat všechny ostatní senzory a motory zastaví. Poté začne překážku objíždět. Vzhledem k tomu, že je robot konstruován tak, aby využíval pravidlům soutěže Istrombot, předpokládá se jako překážka cihla. Samotné objíždění je udělané velmi jednoduše.

Nastaví se rychlosť motorů a čas, po který tak má být. Výsledný program poté vypadá tak, že se robot otočí doprava o 90° , popojede kousek rovně, otočí se o 90° doleva, popojede rovně, otočí se o 90° doleva a poté jede tak dlouho rovně, dokud nenarazí opět na čáru, při jejímž zaznamenání jede doprava. Tím se zajistí, aby se náhodně neotočil doleva a nenarazil na překážku nebo se nezačal vracet v protisměru.

5. KONSTRUKCE

5.1. POPIS MECHANICKÉ KONSTRUKCE

Pro robota jsem zvolil dvoukolový podvozek s třecím bodem, který je pro zmenšení třecího odporu realizován kuličkou umístěnou v pouzdře s ložisky. Obě kola jsou nezávisle poháněny vlastním elektrickým komutátorovým motorem. Toto řešení umožňuje zatačet na velmi malém prostoru, a tak by robotovi nemělo činit problém ani projetí ostrých zatáček.

Nosná část robota je zkonztruována ze dvou základních desek MLAB ALBASE1115, což umožňuje osadit robota v podstatě jakýmkoliv MLAB kompatibilním zařízením. Zároveň děrování desek v rastru 400 mils odlehčuje celkovou konstrukci. Desky jsou k sobě spojeny závitovými tyčemi, čímž se vytvořili dvě patra. Do prvního jsem umístil motory, odrušovací cívky a modul LION2CELL01A. Ve druhém patře se nachází zbytek elektroniky.

Na konstrukci jsem použil také díly ze stavebnice Merkur. Ty slouží k uchycení rádkového senzoru a osvětlení, k upevnění motorů a také jsem z nich vytvořil nárazník.

Důležitou částí konstrukce, kterou bylo potřeba vyřešit byl právě úchop optického rádkového senzoru OLSA01A a modulu LEDSENBAR01A. Rozhodující byl úhel, pod kterým snímal OLSA01A čáru. Pokud se úhel snímání blížil 90° , začínalo být snímaní nespolehlivé kvůli stínění. Naopak, pokud byl úhel příliš malý, uplatňovaly se při snímání odlesky povrchu, které opět snižovaly spolehlivost snímání.

Modul s LED diodami LEDSENBAR01A jsem umístil do takové vzdálenosti, aby se světlo z jednotlivých LED diod na snímaném povrchu slévalo, což také pomáhá zvýšit spolehlivost snímání.

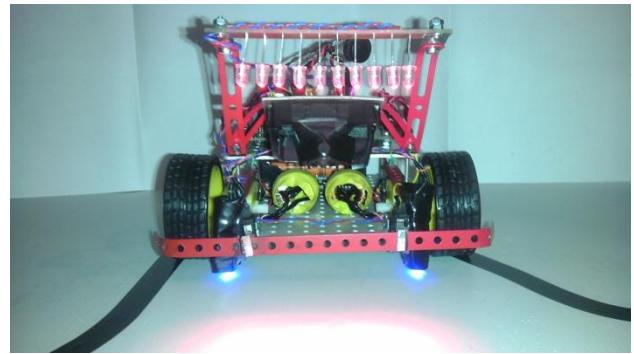
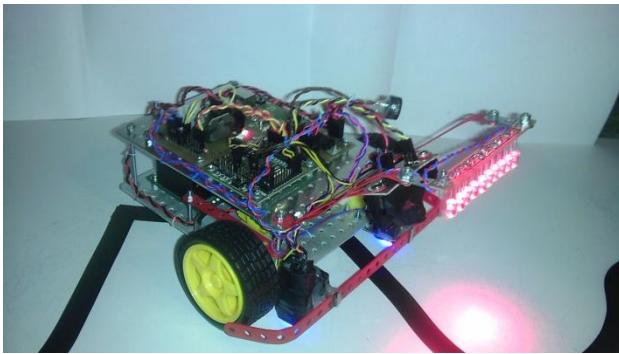
Robot je 20 cm široký a vysoký je 14 cm i s konstrukcí potřebnou pro rádkový senzor a modul s osvětlením. Váha i s bateriemi je 950g.

5.2. FOTOGRAFIE ROBOTA

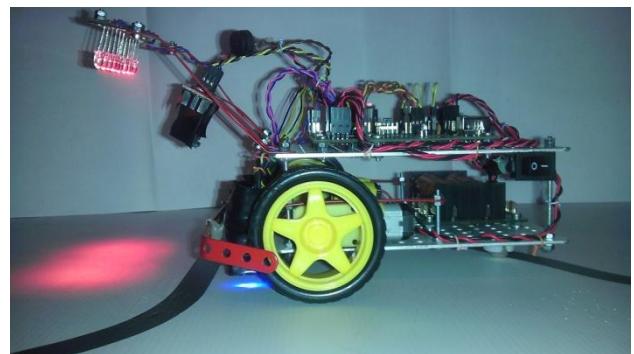
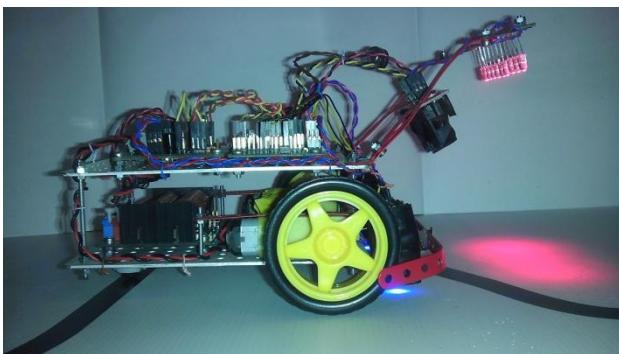
Po dobu vývoje jsem prováděl fotodokumentaci svých prací. Vybrané fotografie a videa najdete na přiloženém CD nosiči. Na další straně si můžete prohlédnout na fotografií finální podobu robota.

Jak lze vidět z fotografií, ve velkých mezích se měnila výška robota. To bylo způsobené jednak změnou velikosti bateriového modulu, ale také hledáním správného úhlu a výšky optického rádkového senzoru.

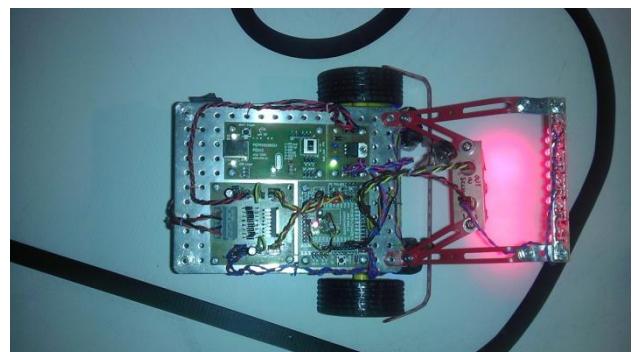
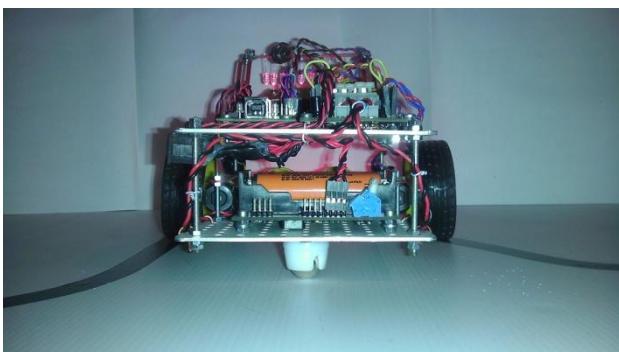
Na posledním páru fotografií na poslední straně jsou vidět snímky, kdy probíhalo testování optického rádkového snímače. Jak si lze všimnout, robot byl v té době napájen z provizorního zdroje elektrické energie - NiMh akumulátoru do airsoftových zbraní. Na průběhu z testování je patrné, že senzor odesílá data. Žlutá barva znázorňuje průběh dat, zelená hodinový signál, který jsem kvůli větší názornosti zobrazil s poloviční amplitudou.



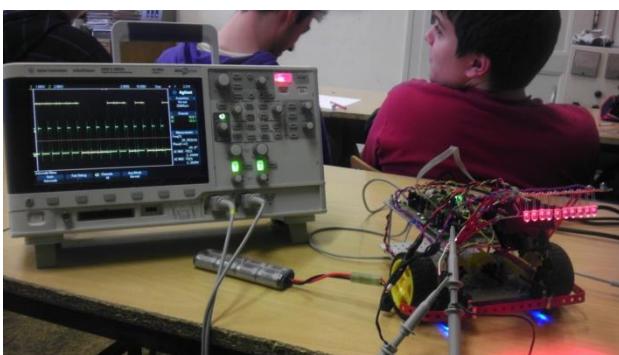
Celkový pohled na robota (vlevo) a pohled zepředu (vpravo)



Pravá strana robota (vlevo) a levá strana robota (vpravo)



Pohled zezadu (vlevo) a pohled shora (vpravo)



Test funkčnosti optického řádkového senzoru (vlevo) a detail průběhu (vpravo)

6. ZÁVĚR

Při řešení maturitní práce jsem prozkoumal problematiku týkající se senzoriky, řízení a konstrukce robota sledujícího černou čáru na kontrastním - bílém podkladě. Zaobíral jsem se zejména nad problematikou užití optického rádkového senzoru, který se jeví jako sofistikovanější řešení pro snímání černé čáry, než jsou jednoduché bodové senzory.

Výsledkem této práce je prototyp robota sledujícího černou čáru. Tento robot všemi parametry odpovídá zadání a požadavkům na něj. Vyzdvihl bych zejména kompatibilitu se stavebnicí MLAB, která umožnuje použít robota v kombinaci s dalšími moduly, a tak jej rozšířit o mnoho dalších funkcí, nebo použít moduly navržené pro tohoto robota i k jiným účelům.

Podle dokumentace přiložené k této práci lze sestavit shodný typ robota a použít jej například k výuce na středních i vysokých školách nebo na mezinárodní soutěže, jako je např. Robotchallenge ve Vídni nebo Istrombot v Bratislavě.

Přínosem této práce je vytvoření důležitých modulů do stavebnice MLAB, které mohou sloužit při sestrojování mobilních robotů (např. OLSA01A a HBRDGL29801A) i dalších elektronických zařízení. (např. LION2CELL01A).

7. POUŽITÁ LITERATURA

Při realizaci své maturitní práce jsem čerpal z těchto knih, katalogových listů a webových stránek:

- [1] PRŮMYSLOVÁ ELEKTRONIKA A INFORMAČNÍ TECHNOLOGIE
Heinz Haberle a kolektiv
- [2] UČÍME SE PROGRAMOVAT V JAZYCE C D
Václav Kadlec
- [3] ELEKTRONIKA - SOUČÁSTKY A OBVODY, PRINCIPY A PŘÍKLADY G
Jan Vobecký, Vít Záhlava
- [4] DIODA, TRANZISTOR A TYRISTOR NÁZORNĚ
Siemens Aktiengesellschaft, SNTL
- [5] STAVEBNICE MLAB
<http://www.mlab.cz>
- [6] H-MŮSTKY A ELEKTRONICKÉ SPÍNAČE
<http://www.robotrevue.cz>
- [7] NAPÁJECÍ ZDROJE, BATERIE A AKUMULÁTORY
<http://www.wikipedie.org>
- [8] L298 - DUAL FULL-BRIDGE DRIVER
<http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00000240.pdf>
- [9] LM78XX - 3-TERMINAL 1A POSITIVE VOLTAGE REGULATOR
<http://www.fairchildsemi.com/ds/LM/LM7805.pdf>
- [10] BQ24103 - SYNCHRONOUS SWITCHMODE LI-ION & LI-POL CHARGE MNGMT IC
<http://www.ti.com/lit/ds/symlink/bq24103.pdf>
- [11] SN74HC164 - 8-BIT PARALLEL-OUT SERIAL SHIFT REGISTERS
<http://www.ti.com/lit/ds/symlink/sn74hc164.pdf>
- [12] BQ34Z100 - WIDE RANGE FUEL GAUGE WITH IMPEDANCE TRACK
<http://www.ti.com/lit/ds/symlink/bq34z100.pdf>
- [13] BQ29209 - VOLTAGE PROTECTION WITH AUTOMATIC CELL BALANCE
<http://www.ti.com/lit/ds/symlink/bq29209.pdf>
- [14] TSL3301 - LINEAR OPTICAL SENSOR ARRAY WITH AD CONVERTER
<http://www.eureca.de/datasheets/01.xx.xxxx/01.06.xxxx/01.06.0015/TSL3301.pdf>
- [15] USING THE TSL3301 WITH A MICROCONTROLLER
<http://www.ams.com/eng/content/download/251600/993363/version/1>

- [16] PIC16F887 - 8-BIT CMOS MICROCONTROLLERS
<http://ww1.microchip.com/downloads/en/DeviceDoc/41291G.pdf>
- [17] PICPROGUSB02A - PROGRAMÁTOR PRO PROCESORY PIC
<http://www.mlab.cz/Modules/PIC/PICPROGUSB02A/DOC/PICPROGUSB02A.cs.pdf>
- [18] PIC16F87XTQ4401B - MODUL PRO PROCESORY PIC16F87X V POUZDRU TQFP44
<http://www.mlab.cz/Modules/PIC/PIC16F87xTQ4401B/DOC/PIC16F87xTQ4401B.cs.pdf>
- [19] ATMEGA801B - MODUL PROCESORU ATMEGA8
<http://www.mlab.cz/Modules/AVR/ATmegaTQ3201A/DOC/ATmegaTQ3201A.cs.pdf>
- [20] USBRS23201B - PŘEVODNÍK USB NA RS232
<http://www.mlab.cz/Modules/CommSerial/USB232R01B/DOC/USB232R01B.cs.pdf>
- [21] LABDUINO - HOW TO
<http://www.mlab.cz/Designs/LABduino/DOC/LABduino.cs.pdf>
- [22] PROJEKT ARDUINO
<http://www.arduino.cc/>

8. OBSAH PŘILOŽENÉHO CD

Součástí této práce je také přiložené CD, kde lze nalézt celou práci v elektronické podobě i s dokumentací ke všem použitým modulům. Dále se na CD nachází i program v jazyce C včetně všech souborů potřebných pro překlad ve vývojovém prostředí PCW PIC C Compiler, ale i přeložený soubor pro procesor PIC16F887.

Na CD jsou umístěny i videa a fotografie robota, jak z finální verze, tak z verze testovací.

ADRESÁŘOVÁ STRUKTURA CD:

/DOC/MODULES	elektronická verze dokumentace všech použitých modulů
/DOC/MP	elektronická verze této práce
/PHOTOS	fotografie robota
/SW	soubory programu pro robota
/VIDEOS	videa robota

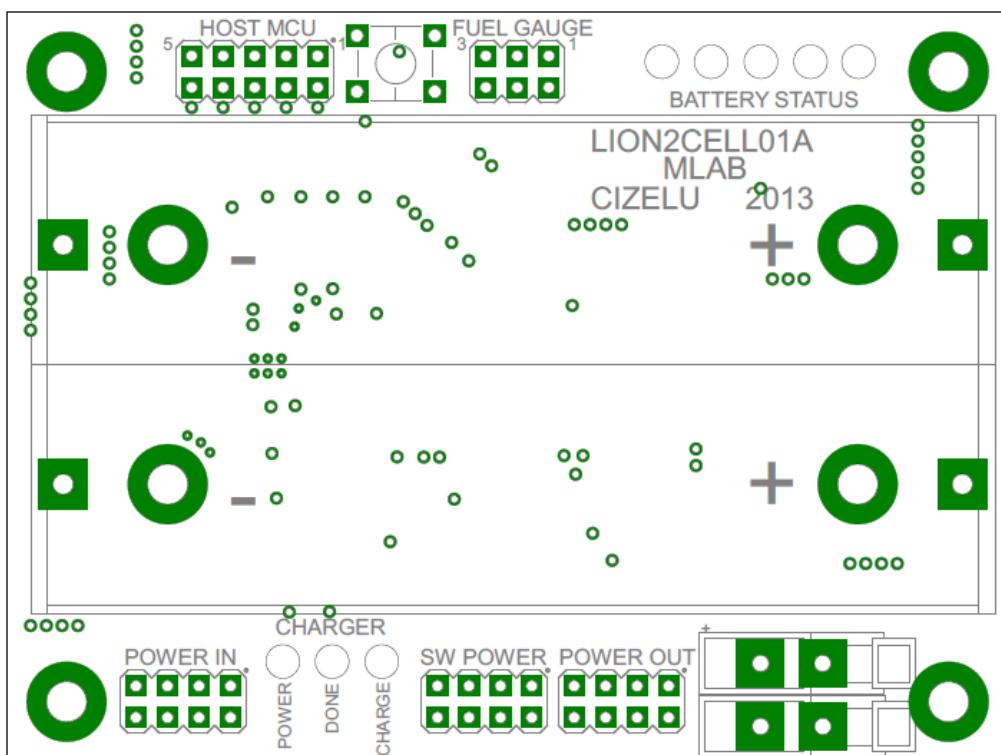
9. PŘÍLOHA A

DOKUMENTACE NAVRŽENÝCH MODULŮ

Napájecí modul se dvěma Li-Ion články

Lukáš Čížek

Modul slouží jako držák dvou Li-Ion článků zapojených v sérii, který zároveň zajišťuje jejich ochranu i dobíjení.



1. Technické parametry

Parametr	Hodnota	Poznámka
Vstupní napětí	8,4 - 16 V	Pro dobíjení článků
Vstupní proud	Až 2,2 A	Pro dobíjení článků
Výstupní napětí	Přibližně 7,4 V	Z obou článků v sérii
Výstupní proud	Maximálně 5 A	Dle použité pojistky
Rozměry	81,3x60x30	Výška nad základnou je dána SMD cívkou

2. Popis konstrukce

2.1. Úvodem

Li-Ion (Lithium-iontové) akumulátory jsou perspektivním zdrojem elektrické energie. Dnes jsou běžnou součástí spotřební elektroniky, používají se běžně např. v notebookech a mobilních telefonech.

Jejich výhodou je velká hustota energie, která umožňuje dosáhnout velké kapacity akumulátorů i při malé hmotnosti a malých rozměrech.

Při nesprávném zacházení, například při zkratování nebo přebíjení akumulátoru, může dojít k jejich vznícení a explozi. Proto je potřeba zajistit jejich ochranu.

Tento modul vznikl proto, aby bylo možné používal Li-On články i na mobilních robotech a dalších přenosných systémech.

2.2. Zapojení modulu

Modul je navržen tak, aby obsahoval jak nabíječku Li-On akumulátorů, tak poskytoval jejich ochranu. Lze ho sestavit ve třech provedeních.

V kompletním provedení modul poskytuje jak nabíječku, měříč zbývající kapacity, tak i ochranné obvody. Tato varianta však může být pro některé účely zbytečná a tedy i nákladná. Proto je modul navržen tak, aby ho bylo možné používat jako samostatnou nabíječku i jako ochranu pro dva Li-On články s měřením zbývající kapacity.

2.2.1. Dobíjení článků

Nabíječka je realizována s pomocí specializovaného obvodu BQ24103, ale lze osadit i integrovaný obvod BQ24113, který informace o dobíjení nezobrazuje na trojici LED diod, ale je uzpůsoben pro komunikaci s hostitelským procesorem.

Oba obvody se chovají jako spínané zdroje. Integrovaný obvod při nabíjení měří nabíjecí proud, napětí na baterii a teplotu pouzdra akumulátoru (pomocí 10k NTC termistoru R11). Podle těchto hodnot nastavuje velikost dobíjecího proudu (je dobré vědět, že Li-Ion články se nabíjejí proudovým zdrojem).

Záměna BQ24103 za BQ24113 se provede tak, že se neosadí LED diody (D3, D4 a D5) a rezistory (R1, R2, R3, R4 a R5), ale hřebínek označený jako SV4.

Pokud nebudem chtít nabíječku použít, stačí část s nabíječkou pouze neosadit (není potřeba dělat jakékoli další úpravy).

2.2.2. Ochrana článků

Modul je chráněn pomocí SMD tavné pojistky, která je zapojena v sérii s Li-On články. Pokud proud pojistikou překročí hodnotu 5A (záleží na použité pojistce - lze osadit i hodnotu menší), pojistka se přepálí a odpojí akumulátor.

Dále je na modulu osazen integrovaný obvod BQ29209, který měří napětí na obou článcích a v případě, že by se blížilo jejich zničení vlivem nabíjení (při přebití může dojít k explozi), chvilkově zkratuje SMD tranzistorem akumulátor, který způsobí přepálení tavné pojistky, čímž akumulátor odpojí.

Integrovaný obvod BQ29209 krom ochrany akumulátoru přepálením pojistky také zajišťuje balancování jednotlivých článků malými proudy. To pomáhá chránit články před podvybitím nebo přebitím, protože jak měřící obvod BQ34Z100, tak nabíječka s obvodem BQ24103 měří součet napětí obou článků (jsou zapojeny v sérii) proti zemi, a tak to, že jeden z článků má větší nebo menší napětí (čímž může dojít ke zničení jednoho ze dvou článků), nezjistí.

2.2.3. Měření a zobrazování stavu akumulátoru

Měření zajišťuje specializovaný obvod BQ34Z100. Z důvodu zmenšení vlastní spotřeby probíhá pouze při stisknutí tlačítka SW1. Kvůli rozsahu interního AD převodníku probíhá měření napětí akumulátoru přes spínaný dělič napětí. K tomuto obvodu doporučují připojit NTC termistor, který je umístěn přímo na plošném spoji. Měřící obvod jím měří teplotu akumulátorů, která je důležitá pro správnost výpočtu zbývající energie ve článcích.

Obvod BQ34Z100 komunikuje s hostitelským procesorem pomocí sériového standardu I2C nebo HDQ. Tato komunikace je obousměrná, proto měřící obvod může zasílat informace o akumulátoru a hostitelský procesor ho např. může přepínat do různých režimů (podrobněji rozepsáno níže).

Měřící obvod má tři režimy (NORMAL, SLEEP a FULL SLEEP), což pomáhá výrazně snížit vlastní spotřebu (přibližně 100 uA). V režimu NORMAL je obvod připojen ke zdroji napájení a může vykonat jakýkoliv požadovaný úkol. V režimu SLEEP, který je nastaven automaticky, má obvod sníženou spotřebu a periodicky provádí měření. Pokud dojde k překročení nastavených hodnot proudu tekoucího měřícím odporem, dojde k přepnutí do režimu NORMAL. Pokud se obvod nachází v režimu FULL SLEEP, vypne se i interní oscilátor a obvod pouze čeká na požadavek o přepnutí do režimu SLEEP.

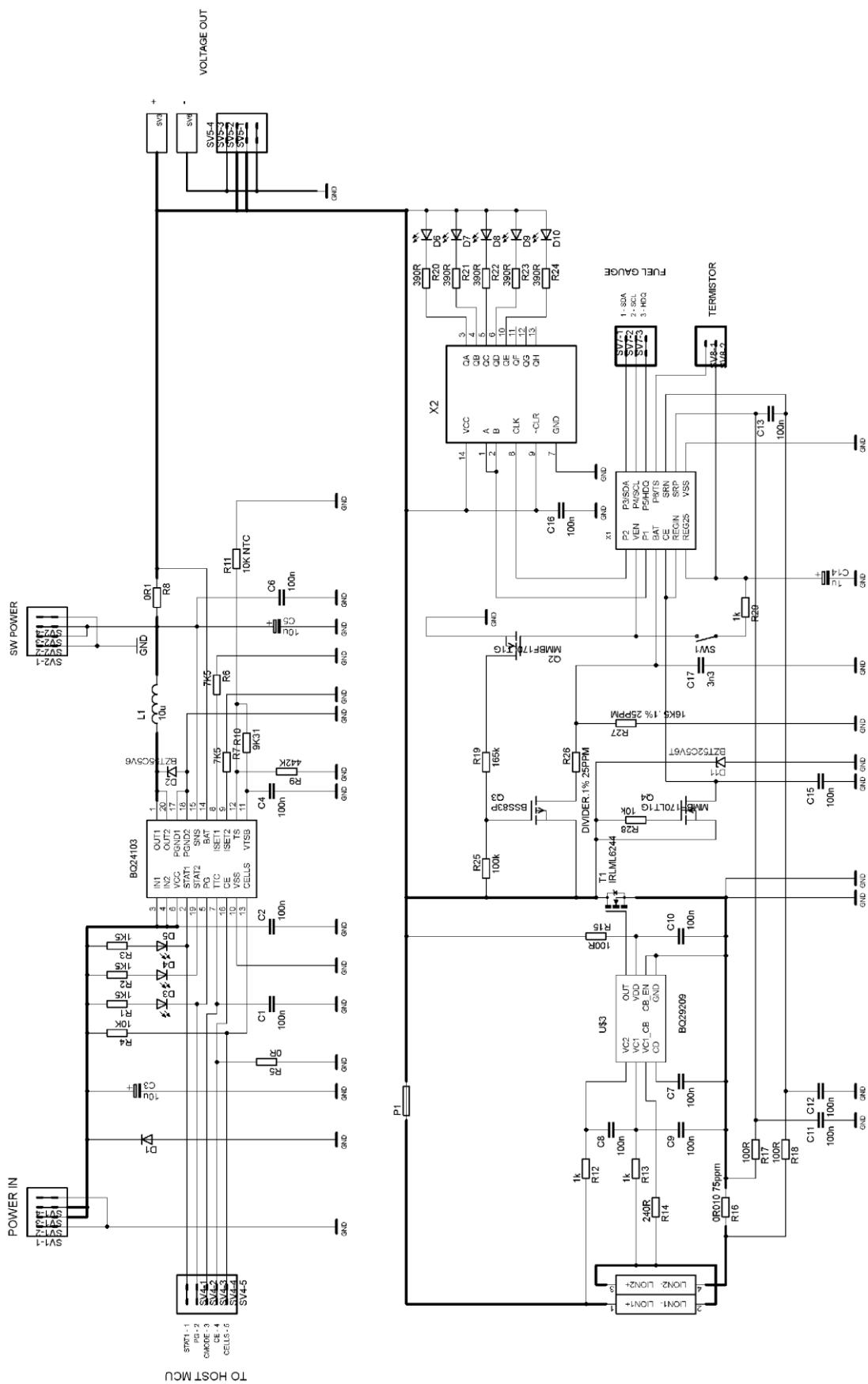
Zobrazení stavu akumulátoru probíhá na modulu pomocí 8-bitového registru se sérovým vstupem, který má paralelní výstup, k němuž je připojeno přes odpor 5 LED diod.

V případě, že budeme modul chtít využít pouze jako nabíječku, stačí neosazovat ochranný a měřící obvod a registr. V tomto případě však doporučují použít ji pro nabíjení dvoučlánkových akumulátorů, které v pouzdru obsahují i ochranné obvody, aby nedošlo k jejich poškození nebo explozi.

Na další straně najeznete zapojení celého modulu. Pro lepší orientaci je v databázi stavebnice MLAB uloženo ve formátu PDF určeným pro tisk na papír velikosti A3.

LION2CELL01A

MLAB



2.3. Práce s modulem

Při dobíjení je nutné připojit zdroj na hřebínek SV1. Nabíjecí obvod lze použít i v případě provozu zařízení. K tomu slouží hřebínek SV2.

Pokud chceme zobrazit na LED diodách zbývající kapacitu akumulátoru, musíme stisknout tlačítko SW1, které způsobí aktivaci měřícího obvodu, spínánoho napěťového děliče a dojde k odeslání zprávy do posuvného registru, který zajistí rozsvícení LED diod.

Podle odebíraného proudu připojíme zátěž bud' na hřebínek SV5 nebo na WAGO svorky (SV3 - kladný pól, SV4 - zem).

2.3.1. I2C komunikace

I2C komunikace probíhá pomocí dvou vývodů P3 - SDA (data) a P4 - SCL (hodinový signál). Maximální hodnota frekvence pro hodinový signál je 400 kHz.

Pro seznam příkazů, jejich popis a použití si prostudujte katalog součástky BQ34Z100 ze stránek Texas Instruments.

2.3.2. HDQ komunikace

Kromě komunikace pomocí standardu I2C obvod BQ34Z100 umožnuje i komunikaci přes rozhraní HDQ. Tento typ komunikace vyžaduje pouze jeden vodič. Ten je připojen k pinu P5 - HDQ. Pro správnou funkci komunikace je důležité připojit externí pullup nebo pulldown rezistor.

Pro seznam příkazů, jejich popis a použití si prostudujte katalog součástky BQ34Z100 ze stránek Texas Instruments. Pro způsob komunikace HDQ a jeho popis si prostudujte pdf "*HDQ Communication Basics for TI's Battery Monitor ICs*" ze stránek Texas Instruments.

2.3.3. Popis programu pro čtení měřiče kapacity

Adresa obvodu má sedm bitů. Osmý bit určuje, zda z obvodu chceme číst nebo do něj zapisovat. Pro čtení se tedy používá adresa 0xAB a pro zápis 0xAA. Hodnota, kterou pošleme na adresu 0xAA určuje, jaká data obvod připraví na adresu 0xAB.

Na následujícím příkladu názorně ukážu, jak zjistit napětí článků pomocí sběrnice I2C.

```
i2c zapiš 0xAA;      // komunikace s obvodem BQ34Z100
i2c zapiš 0x08;      // určení typu dat, které obvod připraví - 0x08 pro prvních osm bitů napětí
i2c zapiš 0xAB;      // komunikace s obvodem BQ34Z100
napětí_1 = čti i2c; // napětí článků lze přečíst ze sběrnice I2C
i2c zapiš 0xAA;      // komunikace s obvodem BQ34Z100
i2c zapiš 0x08;      // určení typu dat, které obvod připraví - 0x09 pro druhých osm bitů napětí
i2c zapiš 0xAB;      // komunikace s obvodem BQ34Z100
napětí_2 = čti i2c; // napětí článků lze přečíst ze sběrnice I2C
```

Na další straně uvádím tabulku, kde je uveden seznam instrukcí pro vyčítání různých parametrů z obvodu BQ34Z100. Jak je vidět z předchozího příkladu i uvedené tabulky, některé veličiny mají šestnáctibitovou hodnotu, a tak je nutné je číst nadvakrát a v řídící jednotce dopočítat.

Seznam příkazů:

NAME		COMMAND CODE	UNITS	SEALED ACCESS	UNSEALED ACCESS
<i>Control()</i>	CNTL	0x00 / 0x01	N/A	R/W	R/W
<i>StateOfCharge()</i>	SOC	0x02 / 0x03	%	R	R
<i>RemainingCapacity()</i>	RM	0x04 / 0x05	mAh	R	R
<i>FullChargeCapacity()</i>	FCC	0x06 / 0x07	mAh	R	R
<i>Voltage()</i>	VOLT	0x08 / 0x09	mV	R	R
<i>AverageCurrent()</i>	AI	0x0a / 0x0b	mA	R	R
<i>Temperature()</i>	TEMP	0x0c / 0x0d	0.1K	R	R
<i>Flags()</i>	FLAGS	0x0e / 0x0f	N/A	R	R
<i>Mfr Date</i>	DATE	0x6B / 0x6c	N/A	R	R
<i>Mfr Name Length</i>	NAMEL	0x6d	N/A	R	R
<i>Mfr Name</i>	NAME	0x6e – 0x78	N/A	R	R
<i>Device Chemistry Length</i>	CHEML	0x79	N/A	R	R
<i>Device Chemistry</i>	CHEM	0x7a – 0x7d	N/A	R	R
<i>Serial Number</i>	SERNUM	0x7e / 0x7f	N/A	R	R

Pomocí příkazu *Control()* lze obvod nastavovat a řídit. V další tabulce naleznete jejich seznam.

Příkazy podmíněné odesláním příkazu *Control()*:

CNTL FUNCTION	CNTL DATA	SEALED ACCESS	DESCRIPTION
CONTROL_STATUS	0x0000	Yes	Reports the status of DF Checksum, IT, for example.
DEVICE_TYPE	0x0001	Yes	Reports the device type of 0x0541 (indicating bq34z100)
FW_VERSION	0x0002	Yes	Reports the firmware version on the device type
HW_VERSION	0x0003	Yes	Reports the hardware version of the device type
RESET_DATA	0x0005	No	Returns reset data
PREV_MACWRITE	0x0007	No	Returns previous MAC command code
CHEM_ID	0x0008	Yes	Reports the chemical identifier of the Impedance Track configuration
BOARD_OFFSET	0x0009	No	Forces the device to measure and store the board offset
CC_OFFSET	0x000A	No	Forces the device to measure the internal CC offset
CC_OFFSET_SAVE	0x000B	No	Forces the device to store the internal CC offset
DF_VERSION	0x000C	Yes	Reports the data flash version on the device
SET_FULLSLEEP	0x0010	No	Set the [FULLSLEEP] bit in the control register to 1
STATIC_CHEM_CHKSUM	0x0017	Yes	Calculates chemistry checksum
CURRENT	0x0018	Yes	Returns the instantaneous current measured by the gauge
SEALED	0x0020	No	Places the device in SEALED access mode
IT_ENABLE	0x0021	No	Enables the Impedance Track algorithm
CAL_ENABLE	0x002D	No	Toggle calibration mode
RESET	0x0041	No	Forces a full reset of the bq34z100
EXIT_CAL	0x0080	No	Exit calibration mode
ENTER_CAL	0x0081	No	Enter calibration mode
OFFSET_CAL	0x0082	No	Reports internal CC offset in calibration mode

Pokud budeme chtít využít příkazu *Control()*, probíhá komunikace podobně, jako bylo uvedeno výše, se ještě po jeho odeslání musí poslat příkaz z předchozí tabulky a teprve poté lze začít s vyčítáním.

2.4. Mechanická konstrukce

Vzhledem ke složitosti zapojení je modul realizován na oboustranném plošném spoji. Silové cesty jsou vedeny co nejtlustšími spojemi. Obvod je kvůli zamezení zkratu z obou stran lakován. Kvůli cívce pro dobíjecí obvod je modul umístěn poměrně vysoko nad základnou.

Důležitá je provedení prokovů pod dobíjecím obvodem, které umožňují lepší odvod tepla (jako chladič se použije i druhá strana plošného spoje), čímž se zamezí poškození obvodu teplem vznikajícím při dobíjení.

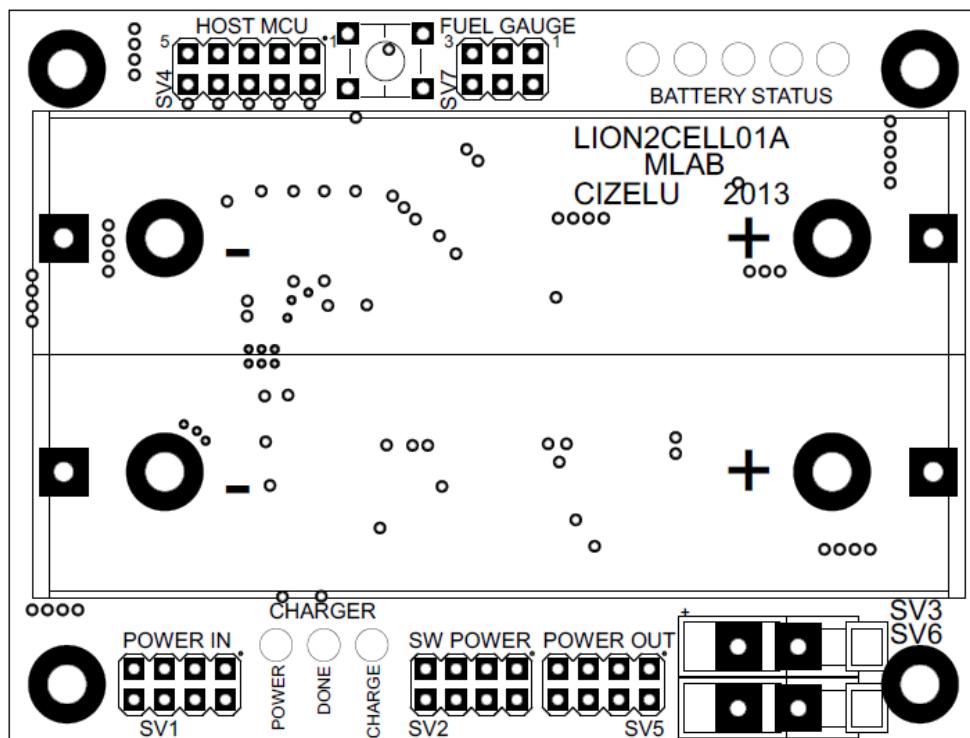
V rozích má modul šrouby M3, které v tomto případě nejsou spojeny se zemí.

3. Osazení a oživení

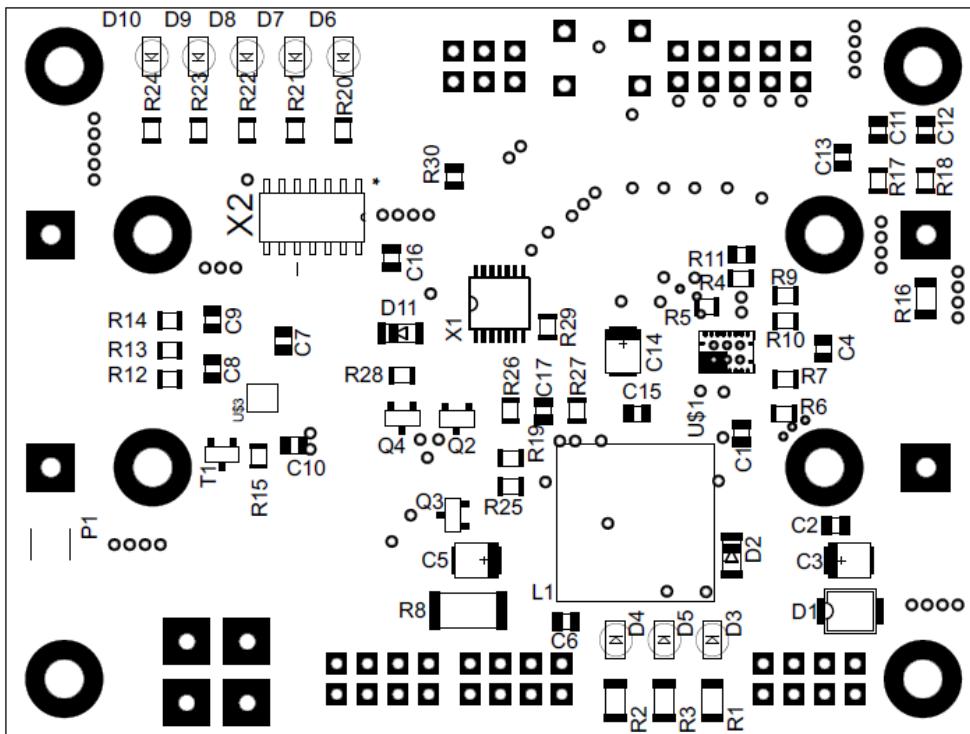
3.1. Osazení a seznam součástek

Všechny součástky, až na hřebínky, svorky a držák baterie jsou umístěny na horní straně plošného spoje. SMD LED diody doporučuji osadit tak, aby mohly svítit otvory vyvrstanými skrz desku plošného spoje. Tato konstrukce lépe chrání diody před mechanickým poškozením, přičemž není potřeba pro kontrolu stavu modul otáčet.

Na další straně je osazovací plánek plošného spoje a dále názvy součástek a jejich hodnoty.



Osazení modulu LION2CELL1A - vrchní strana



Osazení modulu LION2CELL1A - spodní strana

C1, C2, C4, C6, C7, C8, C9, C10, C11, C12, C13, C15, C16		
	100n	C0805
C3, C5	10u	SMB
C14	1u	SMB
C17	3n3	C0805
D1	1N4007	SMB
D2	BZT55C5V6	MINIMELF
D3, D4, D5, D6, D7, D8, D9, D10		
	LED	1206
D11	BZT52C5V6T	MINIMELF
L1	10u	DE1205-10
Q2	MMBF170LT1G	SOT23
Q3	BSS83P	SOT23
Q4	MMBF170LT1G	SOT23
R1, R2, R3		
	1K5M	1206
R4	10K	M0805
R5	0R	M0805
R6, R7	7K5	M0805
R8	0R1	R2512
R9	442K	M0805
R10	9K31	M0805
R11	10K NTC	R0805
R12,R13	1k	M0805
R14	240R	M0805
R15,R17,R18		
	100R	M0805
R16	0R010 75ppm	1206
R19	165k	M0805
R20, R21, R22, R23, R24		
	390R	M0805
R25	100k	M0805
R26	DIV.1% 25PPM	M0805
R27	16K5 .1% 25PPM	M0805
R28	10k	M0805
R29	1k	M0805
SW1	switch	TACT-64K
T1	IRLML6244	SOT23
IO1	BQ24103	RHL_R-VQFN-N20
IO2	BQ29209	DRB-N8
IO3	BQ34Z100	TSSOP14
IO4	SN74HC164D	SOIC127P600X175-14N

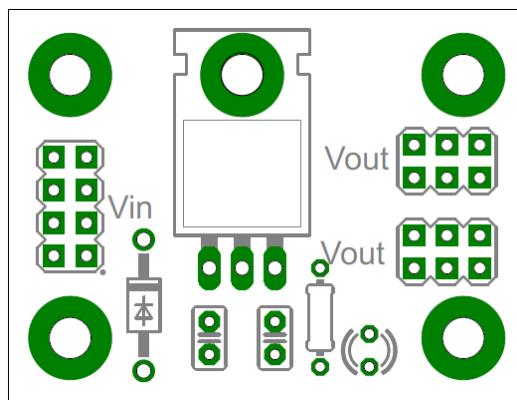
3.2. Oživení

Na modulu není potřeba nic oživovat, vše by mělo fungovat hned po zapojení. Pro zprovoznění komunikace měřícího obvodu přes I2C nebo HDQ je nutné jej nastavit.

Modul jednoduchého stabilizovaného zdroje

Lukáš Čížek

Modul je malý napájecí zdroj postavený na bázi třísvorkových lineárních stabilizátorů 78XX v pouzdru TO-220. Výstupní napětí je dáno typem použitého stabilizátoru.



1. Technické parametry

Parametr	Hodnota	Poznámka
Vstupní napětí	Maximálně 35 V	Pro stabilizátory s výstupním napětím do 18V
Výstupní napětí	5 V	Podle použitého stabilizátoru
Výstupní proud	Maximálně 1 A	V závislosti na chlazení
Rozměry	105x10x15 mm	

2. Popis konstrukce

2.1. Úvodem

Konstrukce vznikla jako základní jednoduchý zdroj pro napájení aplikací s malými odběry, např. procesorů na robotických aplikacích. Proto bylo zvoleno stabilizátoru řady 78XX.

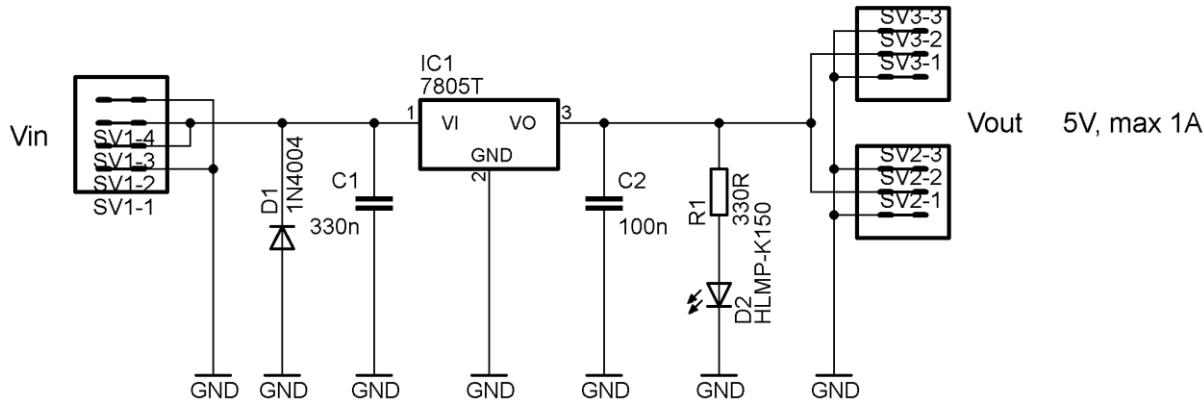
Pozor, tyto stabilizátory nejsou kompatibilní s třísvorkovými nastavitelnými stabilizátory (např. LM317) a negativní stabilizátory řady 79XX.

Nevýhodou stabilizátorů řady 78XX je to, že pokud se vstupní napětí bude blížit předpokládané hodnotě výstupního napětí, bude výstupní napětí nižší (až o 1 V).

2.2. Zapojení modulu

Zapojení modulu je velmi jednoduché. Kromě odrušovacích kondenzátorů obsahuje pouze indikační LED diodu a ochrannou diodu D1, která chrání stabilizátor před zničením přepólováním vstupního napětí.

Na obrázku je vidět zapojení pro stabilizátor 7805. Pro jiné typy je potřeba zvýšit odpor R1, aby nedošlo ke zničení LED diody.



Zapojení modulu LINREG01A

Chladící ploška stabilizátorů řady 78XX je připojena k jejich prostřednímu vývodu (GND). Proto je možné stabilizátor přišroubovat dírou pro šroub M3 k plošnému spoji. Tím se připojí na měděnou vrstvu s potenciálem GND, čímž se zlepší chlazení. Pro velké proudy však doporučují přidat i chladič.

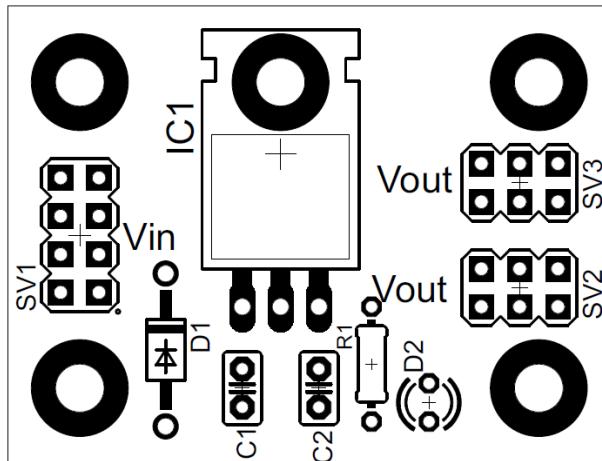
2.3. Mechanická konstrukce

Modul je upevněn pomocí čtyř šroubků M3, které jsou v rozích modulu. Ke stabilizátoru lze připevnit chladič k desce pomocí šroubku M3.

3. Osazení a oživení

3.1. Osazení a seznam součástek

Modul je konstruován na jednostranném plošném spoji. Všechny součástky jsou v THT provedení. Osazovací plánek se seznamem součástek je na následující straně.

*Osazení modulu LINREG01A*

- IC1 78XX (v našem případě 7805)
- D1 1N4004
- D2 LED 3mm
- C1 330 nF
- C2 100 nF
- R1 330 R (pro jiné stabilizátory je nutné změnit jeho hodnotu)

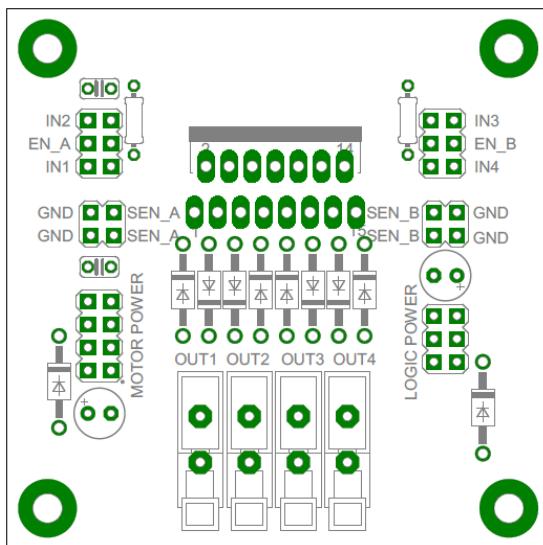
3.2. Oživení

Na modulu není potřeba nic oživovat, vše by mělo fungovat hned po zapojení. Při prvním spuštění doporučuji proměřit výstupní napětí při regulátoru zapojeném na zdroj s proudovým omezením.

Dvojitý H-můstek s integrovaným obvodem L298

Lukáš Čížek

Modul slouží jako výkonový budič pro dva komutátorové motory, které lze řídit nezávisle na sobě.



1. Technické parametry

Parametr	Hodnota	Poznámka
Napájení logiky	Maximálně 7V	
Napájení motorů	Maximálně 50V	
Zatížitelnost	Trvale až 2 x 2A	Špičkově až 3 A
Rozměry	60x60x15mm	Bez chladiče

2. Popis konstrukce

2.1. Úvodem

Modul je určený jako levný a jednoduchý výstup pro ovládání motorů a popř. i jiných výkonových periferií, které není možné připojit přímo na vstup mikroprocesoru.

Základem modulu je integrovaný obvod L298, který v sobě obsahuje dva úplné mosty, což nám umožňuje např. na sobě nezávisle řídit dva motory. Obvod L298 je kompatibilní s TTL. Ovládání motorů je dáno pravdivostní tabulkou (viz níže).

Vstupy			Funkce
$V_{EN} = H$	$IN_A = H$	$IN_B = L$	Vpřed
	$IN_A = L$	$IN_B = H$	Vzad
	$IN_A = IN_B$		Brzdění
$V_{EN} = L$	$IN_A = Z$	$IN_B = Z$	H-můstek odpojen

Pravdivostní tabulka pro ovládání jednoho plného mostu

2.2. Zapojení modulu

Na modulu jsou pomocí hřebíneků vyvedeny výstupy SEN_A a SEN_B. Jedná se o spodní konce obou H-můstků. Tato konstrukce nám umožňuje další ochranu H-můstku proti zkratu. Mezi výstup SEN a zem lze připojít např. pojistku (tavnou nebo vratnou polovodičovou) nebo měřící odpor a tím měřit proud tekoucí H-můstkom a v případě překročení rozumné hodnoty proudu H-můstek odpojit.

Jak je patrné z pravdivostní tabulky, směr otáčení se mění pomocí změny logických stavů na vstupech H-můstku. Pro změnu rychlosti lze využít PWM - pulzní šířkovou modulaci. Signál PWM je výhodné připojit na vstup EN_A nebo EN_B, ale lze proto také použít jeden ze vstupů pro směr otáčení (IN1 až IN4).

Obvod L298 doporučuje pro větší výkony opatřit chladičem. Jeho plocha by měla být nejméně 10 cm². Ke chlazení lze použít i kostru zařízení, protože chladící křídélko integrovaného obvodu je připojena k potenciálu GND. Pracovní teplota obvodu nesmí přesáhnout 70°C.

Schéma modulu je na další straně.

LOGIC POWER

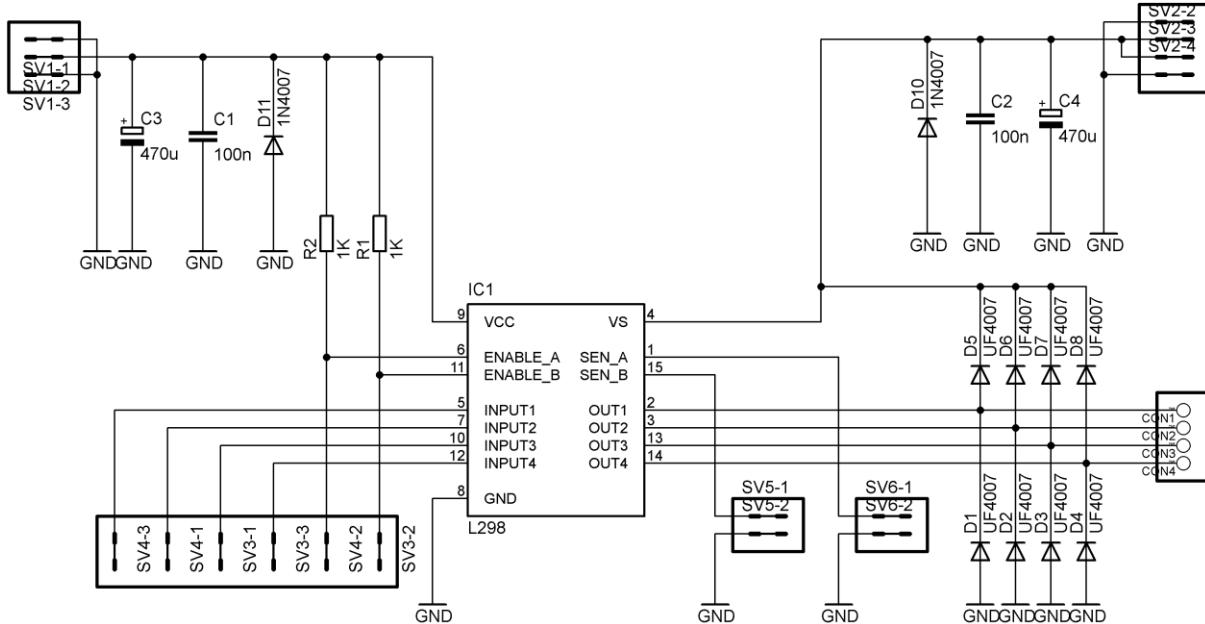
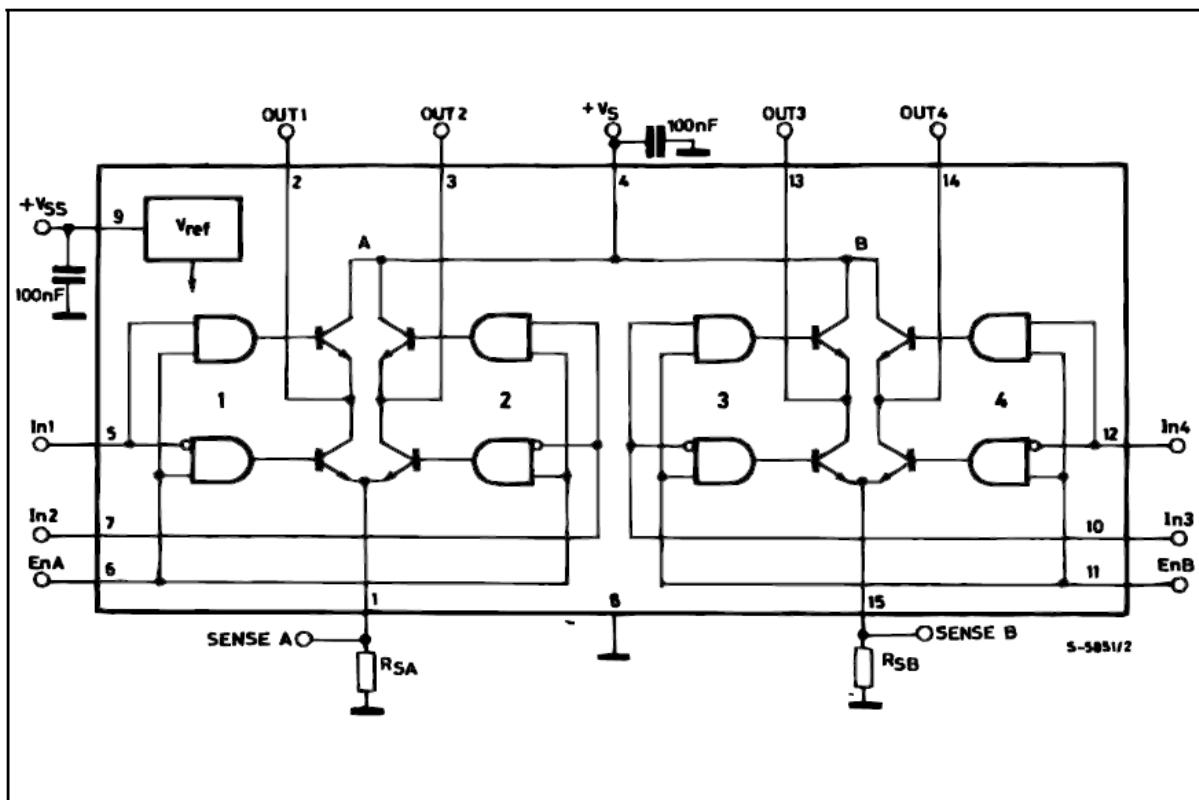


Schéma modulu HBRDGL29801A

2.3. Vnitřní zapojení obvodu L298

Obvod L298 obsahuje osm spínacích výkonových bipolárních tranzistorů v jednom můstku, kdy jsou vždy čtyři použity pro spínání jednoho můstku.



2.4. Mechanická konstrukce

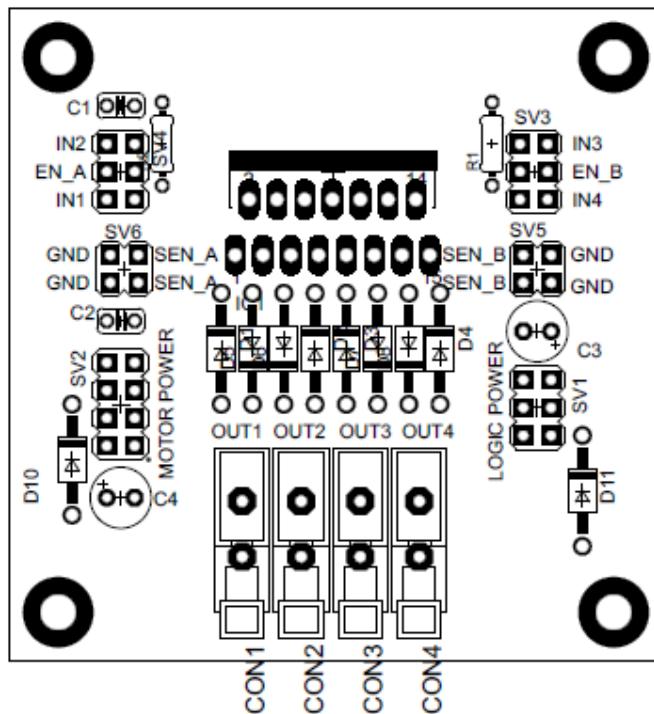
Jedná se o standardní modul stavebnice MLAB se čtyřmi upevňovacími šroubkami v rozích plošného spoje. Na plošném spoji je dostatek místa k připevnění chladiče.

Do plošného spoje nelze provrtat díru a obvod připevnit k plošnému spoji, došlo by ke zničení plošného spoje a vlivem zkrátů i ke zničení integrovaného obvodu!

3. Osazení a oživení

3.1. Osazení a seznam součástek

Modul je postaven na jednostranném plošném spoji, kdy jsou součástky umístěny na vrchní straně a spoje na spodní. Při osazování doporučuji osadit nejdříve diody, poté hřebínky a diody a integrovaný obvod a wago svorkovnice až nakonec.



Osazení modulu HBRDGL29801A

C1, C2	100nF
C3, C4	100uF
R1, R2	1k
D1 - D8	UF4007
D10, D11	1N4007
CON1 - CON4	svorkovnice WAGO256
SV1 - SV6	standardní hřebínky

3.2. Oživení

Pokud jsou použity fungující součástky a spoje jsou správně provedené bude H-můstek fungovat napoprvé. Pro jistotu je vhodné při prvním zapnutí použít regulovatelný zdroj a při postupném zvyšování napájecího napětí zkонтrolovat, zda je spotřeba naprázdno v rozumných mezích.

Poté stačí připojit malý motorek a vyzkoušet funkčnost na základě pravdivostní tabulky.

Modul s optickým řádkovým snímačem

Lukáš Čížek

Modul je založený na integrovaném obvodu TSL3301, což je řádkový snímač s 8-bit AD převodníkem.



1. Technické parametry

Parametr	Hodnota	Poznámka
Napájení	3,0 - 5,5 V	
Rozlišení senzoru	300 DPI	Na senzoru je 102 pixelů
Výstup	Digitální	USART - 8 bitů informace
Frekvence	Až 10 MHz	Na vstupu SCLK
Rozměry	50 x 15 mm	

2. Popis konstrukce

2.1. Úvodem

Tento modul vznikl pro potřeby konstrukce robota, který sleduje čáru na kontrastním podkladě.

Na tomto modulu je počítáno s použitím senzoru TSL3301, ale existuje poměrně velké množství řádkových senzorů. Na tento modul lze umístit pouze senzory se stejným rozložením vývodů (a ve stejném pouzdru), jako má TSL3301.

2.2. Zapojení modulu

Zapojení modulu je velmi jednoduché. Kromě samotného senzoru obsahuje ještě ochrannou diodu (ochrana proti přepólování zdroje) a blokovací kondenzátor.

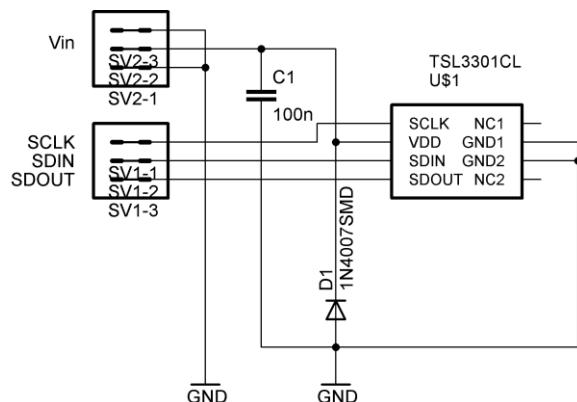
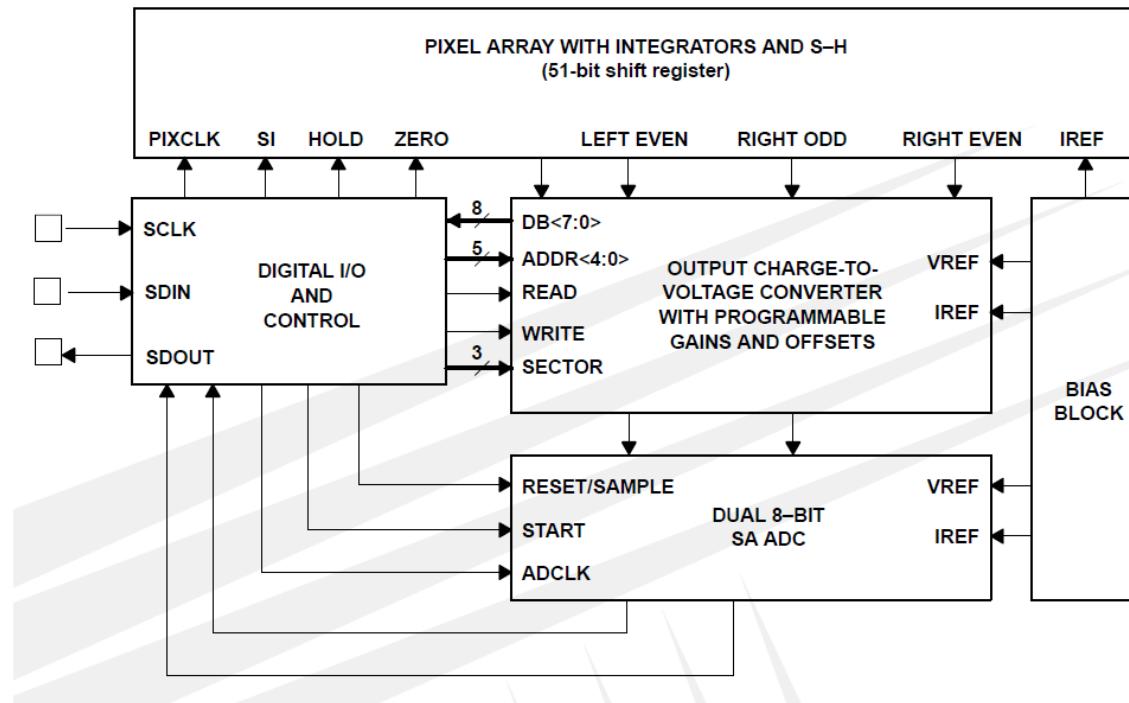


Schéma modulu OLSA01A

2.3. Vnitřní zapojení obvodu TSL3301



Jak je patrné z blokového schématu, obsahuje v sobě integrovaný obvod zajišťující sériovou komunikaci, čtení řádky diod, vyhodnocovací obvod, 8 bitový AD převodník a referenční zdroj.

Pro správnou funkci je důležité vědět, jak se senzorem komunikovat. Komunikaci si rozebereme v podkapitole "2.4. Práce se senzorem".

2.4. Práce se senzorem

Senzor musíme připojit ke zdroji napětí. K tomu na modulu slouží standardní hřebínek se 3 piny.

Pro práci se senzorem jsou k dispozici piny SCLK, SDIN a SDOUT. SCLK je hodinový signál pro sériový vstup i výstup a pro interní logiku senzoru. SDIN je sériový vstup dat a SDOUT je sériový výstup dat. Sériová komunikace odpovídá formátu USART.

Důležité je vědět, že řádka je rozdělena do tří segmentů, pro něž lze nastavit offset a zisk (gain) zvlášť.

2.4.1. Komunikace se senzorem

Po připojení napájení je potřeba senzor resetovat a nastavit vnitřní registry. Po provedení této operace lze senzoru zadat příkaz, aby začal měřit pixely. Poté lze postupně přečíst hodnotu všech pixelů. Doporučuji přečíst dokumentaci k integrovanému obvod TSL3301 a příručku "Using the TSL3301 with a Microcontroller" ze stránek TAOS.

Po zaslání každého příkazu je nutné, aby pin SCLK byl vynulován (nastaven logickou 0) a pin SDIN nastaven na logickou 1, protože poklesem na nulu se rozpozná, že začala komunikace - start bit.

Při posílání dat se ihned po start bitu posílá nejméně významný bit 8 bitové instrukce. Neposílá se paritní bit, ale rovnou po odeslání instrukce se odešle stop bit (alespoň jeden). Poté lze odeslat další instrukci. Doporučuji nastudovat formát komunikace přes rozhraní USART.

Níže je naznačeno, jak senzor správně obsloužit.

RESET: // resetování senzoru

Vynulovat SCLK

Vynulovat SDIN

Poslat 30 impulzů na pin SCLK // tvrdý reset

Nastavit SDIN

Poslat 10 impulzů na pin SCLK // synchronizace start bitu

Poslat instrukci 0x1B // příkaz pro reset

Poslat instrukci 0x5F // zápis do MODE registru

Poslat instrukci 0x00 // vyčistit MODE registr

SETUP: // počáteční nastavení senzoru

Poslat instrukci 0x40 // Offset levého segmentu

Poslat instrukci 0x81 // - 1

Poslat instrukci 0x41 // Zisk levého segmentu

Poslat instrukci 0x05 // 5

Poslat instrukci 0x42 // Offset středního segmentu

Poslat instrukci 0x81 // - 1

Poslat instrukci 0x43 // Zisk středního segmentu

Poslat instrukci 0x05 // 5

Poslat instrukci 0x44 // Offset pravého segmentu

Poslat instrukci 0x81 // - 1

Poslat instrukci 0x45 // Zisk pravého segmentu

Poslat instrukci 0x05 // 5

READ:

Poslat instrukci 0x08	// čtení ze senzoru
Poslat 22 impulzů na pin SCLK	// začátek integrace
Poslat instrukci 0x10	// konec integrace
Poslat 5 impulzů na pin SCLK	
Poslat instrukci 0x02	// začátek odesílání senzorů
Čekat na pokles na pinu SDOUT	// senzor začíná komunikovat
Přečíst 102 pixelů	// USART komunikace, 8 bitů jeden pixel

2.5. Důležité poznámky k použití

Při použití ke snímání čáry je nutné použít vhodný objektiv, kterým přivedeme obraz na senzor.

Při výrobě modulu a uzavírání senzoru do objektivu si musíme dát pozor na prach a další nečistoty, které by později mohly způsobit chyby ve snímání, pokud by zůstaly na povrchu senzoru nebo uvnitř objektivu.

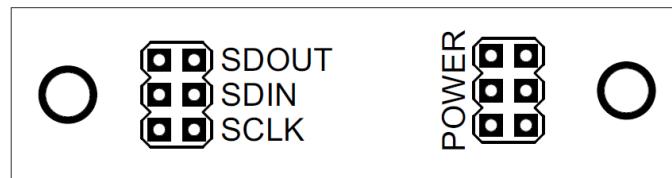
Z charakteristik senzoru vyplývá, že největší citlivost je na světlo s vlnovou délkou 700 nm, což odpovídá červené barvě. Proto je vhodné snímanou plochu nasvítit LED diodami. Běžně jsou dostupné LED diody, které vyzařují světlo s vlnovou délkou 660 nm, což by mělo být pro robota sledujícího čáru zcela dostatečné.

2.6. Mechanická konstrukce

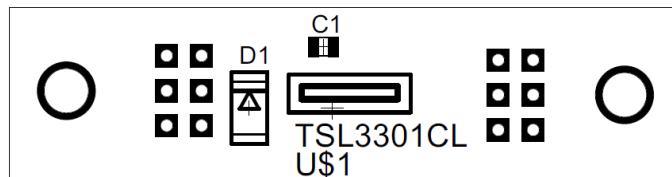
Modul je realizován na jednostranném plošném spoji obsahuje po stranách dva šrouby M3 na stranách desky.

3. Osazení a seznam součástek

Modul je postaven na jednostranném plošném spoji, kdy jsou součástky umístěny na vrchní straně a spoje na spodní. Při osazování doporučují osadit nejdříve diody, poté hřebínky a až nakonec samotný senzor,



Osazení modulu OLSA01A - svrchní strana



Osazení modulu OLSA01A - spodní strana

D1 dioda 1N4007 (DO213AB)

C1 100nF (0805)

U1 TSL3301

Je důležité dát si pozor při osazování senzoru TSL3301, protože je citlivý na teplotu pájení (může se zničit). Také nesmí být mimo ochranné balení déle jak 24 hodin, protože by se do pouzdra dostala vlhkost a při pájení by mohlo prasknout. Proto je nejlepší ho osadit hned, jak bude rozbalen.

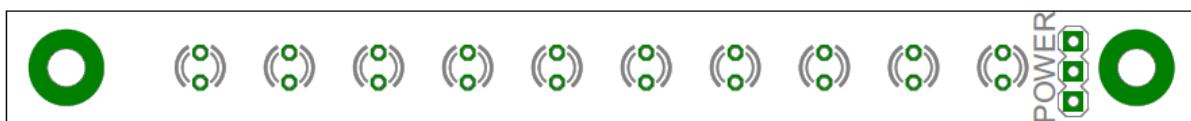
3.1. Oživení

Pokud jsou použity fungující součástky a spoje jsou správně provedené, stačí senzor resetovat a nastavit. Poté z něj můžeme vyčítat hodnoty.

Modul s deseti LED diodami

Lukáš Čížek

Modul slouží k připevnění deseti LED diod. Je určen primárně k osvětlení dráhy pro optické snímače, jako je např. kamera nebo řádkový senzor.



1. Technické parametry

Parametr	Hodnota	Poznámka
Napájení	5 V	Podle osazených odporů a použitých LED diod
Proud	200 mA	Podle použitých LED diod
Rozměry	105x10x10 mm	

2. Popis konstrukce

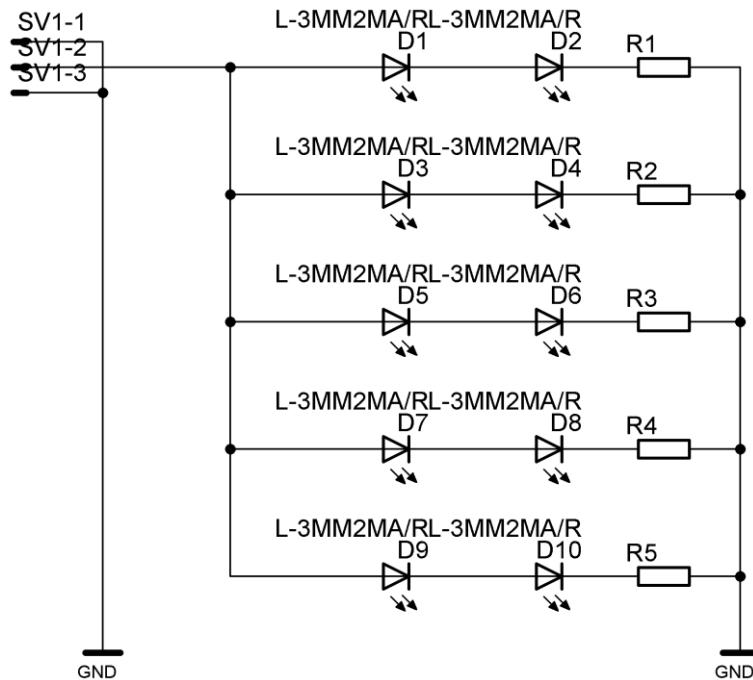
2.1. Úvodem

Tento modul konstrukčně navazuje na modul LEDPANEL01A. Rozdílný je v tom, že má menší rozměry, čímž se více hodí pro montáž na základní desku ALBASE1115.

2.2. Zapojení modulu

Zapojení modulu je opravdu jednoduché. Vždy dvě z deseti diod jsou v sérii s odporem. Tím se dosáhne celkově menšího odběru, než kdyby byly všechny LED paralelně. Kvůli vstupnímu napětí také všechny LED nemohou být v sérii.

Schéma modulu je na druhé straně.



Zapojení modulu LEDSENBAR01A

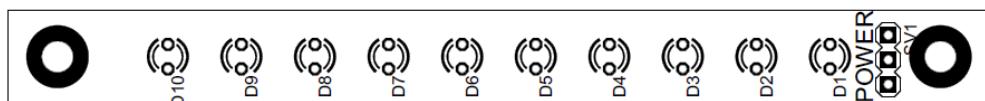
2.3. Mechanická konstrukce

Modul je upevněn pomocí dvou šroubků M3. Led diody je vhodné neumisťovat přímo k plošnému spoji, ale nechat je cca 1 cm nad deskou, čímž se omezí vliv stínění hranami plošného spoje.

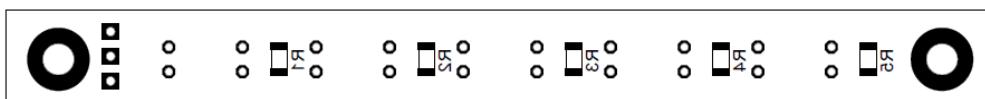
3. Osazení a oživení

3.1. Osazení a seznam součástek

Modul je konstruován na jednostranném plošném spoji. Na horní straně jsou umístěny LED diody, na druhé SMD odpory v pouzdrou 1206. Je možné osadit 3mm i 5mm LED diody.



Osazení modulu LEDSENBAR01A - vrchní strana



Osazení modulu LEDSENBAR01A - spodní strana

D1 - D10 LED diody 3mm/5mm
R1 - R5 rezistor v pouzdru 1206 - velikost zvolte podle přiloženého napětí a LED diod

LED	Napětí UAK	Proud IA	Odpor pro 5 V
Modrá	3,5 V	20 mA	Nízké napětí
Zelená	2,2 V	10 mA	68 R
Žlutá	2,1 V	10 mA	82 R
Červená	2 V	20 mA	51 R
Infračervená	1,5 V	50 mA	43 R
Bílá	3,6 V	20 mA	Nízké napětí

Orientační hodnoty odporů pro různé typy LED diod

3.2. Oživení

Pokud jsou použity fungující součástky a spoje jsou správně provedené rozsvítí se všechny LED diody po přivedení napájení.

10. PŘÍLOHA B

VÝKRESY PLOŠNÝCH SPOJŮ NAVRŽENÝCH MODULŮ

