

SLOVENSKÁ POĽNOHOSPODÁRSKA UNIVERZITA
V NITRE
TECHNICKÁ FAKULTA

NÁVRH RIADENIA BEZKOMUTÁTOROVÝCH JEDNOSMERNÝCH
MOTOROV NA VYUŽITIE
V AUTONÓMNYCH MOBILNÝCH ROBOTOCH

Diplomová práca

Študijný program:	Informačná a automatizačná technika v kvalite produkcie
Študijný odbor:	5. 2. 57 Kvalita produkcie
Školiace pracovisko:	Katedra elektrotechniky, automatizácie a informatiky
Školiteľ:	prof. Ing. Dušan Hrubý, PhD.

Nitra 2010

Peter Roško, Bc.

Zadávací protokol

Čestné vyhlásenie

Podpísaný Peter Roško vyhlasujem, že som diplomovú prácu na tému „Návrh riadenia bezkomutátorových motorov na využitie v autonómnych mobilných robotoch“ vypracoval samostatne s použitím uvedenej literatúry.

Som si vedomý zákonných dôsledkov v prípade, ak uvedené údaje nie sú pravdivé.

V Nitre dňa 28.4.2010

.....

Pod'akovanie

Na tomto mieste by som sa rád poďakoval školiteľovi diplomovej práce, prof. Ing. Dušanovi Hrubému, PhD., za poskytnutý čas, cenné rady a pripomienky počas riešenia danej témy diplomovej práce.

Abstrakt

V súčasnosti sa zvyšujú požiadavky na efektívne využívanie elektrickej energie, čo vedie k zámeru vyrábať ekonomické zariadenia. Je dôležité efektívne riadiť dodávanú energiu. V diplomovej práci je opísaný PWM modul riadenia BLDC motorov s možnosťou riadenia výstupného výkonu motora. Modul obsahuje mikrokontrolér založený na jadre 8051. Pomocou softvéru generuje impulzne – šírkoovo modulované signály a riadi motor pomocou výkonových prvkov. S používateľom komunikuje pomocou tlačidiel, LED diód a USB rozhrania, cez ktoré vysiela informácie o stave riadenia. Modul je určený pre implementáciu do iných zariadení, prípadne ako učebná pomôcka pre študentov.

Kľúčové slová: impulzne – šírková modulácia, BLDC motor, mikrokontrolér, riadenie, programové vybavenie

Abstract

At present increase demands on effective exploitation of electrical power, what increase pretension to manufacture economical devices. It's necessary to control sourcing energy effectively. In this diploma work has been described PWM control modul of BLDC motors with ability to regulate output power . This modul contains an operating microcontroler based at 8051 kernel. It generate a pulse – width modulated signals by operatig software and drives the motor using a power components. It communicate with user by using pushbuttons, LED diodes and USB interface, which is used for transmit information about the state of control. This modul is appointed for implementation into other devices, pertinently as a perceptual tool for students.

Key words: pulse – width modulation, BLDC motor, microcontroler, control, software

Obsah

Zoznam použitých označení a skratiek	9
Úvod	11
1. Prehľad o súčasnom stave riešenej problematiky	12
1.1 BLDC motory	12
1.1.1 BLDC motor - základné časti.....	12
1.1.2 BLDC motor – matematický opis	14
1.1.2.1 Opis elektrickej časti BLDC motora	14
1.1.2.2 Opis mechanickej časti BLDC motora.....	15
1.2 Pulse width modulation – Impulzne šírková modulácia	16
1.2.1 Matematický princíp	16
1.2.2 Princípy generovania.....	17
1.2.3 Typy PWM modulácie	17
1.2.4 Ochranná dióda	18
1.3 Riadenie BLDC motorov	19
1.3.1 Riadenie BLDC motora PWM signálom v H – moste.....	20
1.3.2 BEMF – meranie, využitie na riadenie	23
1.3.3 Princíp a použitie Hallových snímačov na riadenie motora	24
1.4 Jednočipové mikropočítače.....	26
1.4.1 Štruktúra jednočipových mikropočítačov	26
1.4.2 Mikrokontroléry s generátorom PWM signálu	28
1.5 Programovanie jednočipových mikropočítačov.....	28
1.5.1 Programovanie v JSA.....	29
1.5.1.1 Formát zdrojového programu v JSA	30
1.5.1.2 Štruktúra zdrojového programu	31
1.5.2 Programovanie vo vyššom programovacom jazyku C	31
1.5.2.1 Konštrukcia programu.....	32
1.5.2.2 Štruktúra programu	33
1.5.3 Zhrnutie z teórie programovania jednočipových mikropočítačov	33

2 Cieľ práce	34
3 Metodika práce.....	35
3.1 Charakteristika pracoviska.....	35
3.2 Požiadavky kladené na vyrobený modul	35
3.2.1 Požiadavky na riadenie motora.....	35
3.2.2 Požiadavky na zobrazenie informácií o stave motora.....	36
3.3 Voľba typu riadiaceho mikrokontroléra a BLDC motora.....	36
3.3.1 Voľba typu riadiaceho mikrokontroléra a vývojovej DPS.....	36
3.3.2 Voľba typu BLDC motora	36
3.4 Charakteristika technického a programového vybavenia	37
3.4.1 Použité prístroje	37
3.4.2 Použité elektronické súčiastky	37
3.4.3 Programové vybavenie.....	37
3.5 Vykonanie funkčných skúšok.....	37
3.6 Návrhy na využitie výsledkov.....	38
4 Výsledky práce	39
4.1 Opis hardvéru	39
4.1.1 Opis mikropočítača C8051F310	42
4.1.1.1 Parametre C8051F310.....	42
4.1.2 Opis operačného zosilňovača LMV358.....	43
4.1.3 Opis USB – UART prevodníka CP2101.....	43
4.1.4 Opis stabilizátora LM2937-33	43
4.1.5 Opis radiča TC4428 pre MOSFET tranzistory	44
4.1.6 Opis MOSFET tranzistorových dvojíc F7309	44
4.2 Opis softvéru	45
4.2.1 Výpis hlavných častí programu.....	49
4.2.2 Opis inicializačných funkcií.....	52
4.2.2.1 SYSCLK_Init ()	52
4.2.2.2 PORT_Init()	53
4.2.2.3 PCA0_Init().....	54
4.2.2.4 T0_Init () a T2_Init ().....	55
4.2.2.5 UART0_Init ();	56

4.2.2.6 ADC_Init()	57
4.2.3 Vývojový diagram softvéru	57
4.3 Dosiiahnuté výsledky	58
5 Diskusia	60
6 Návrhy na využitie výsledkov	62
7 Záver	64
8 Zoznam použitej literatúry	65

Zoznam použitých označení a skratiek

A/D (D/A)	Analógovo/digitálny (digitálno/analógový) prevodník
ALU	Aritmetical and logical unit
BEMF	Back electromotion force
BIOS	Basic input output system
BLDC	Brushless DC
CMOS	Complementary metal oxide semiconductor
COM	komunikačný (port)
CPU	Central procesing unit
Č/Č	počítadlo/časovač
DC	jednosmerný
DI	Dekodér inštrukcií
DPS	Doska plošných spojov
EPROM	Druh pamäte
f_{osc}	Frekvencia oscilátora
HDD	Harddisc
HEX	označenie šestnástkovej sústavy
I/O	Input/output
ISR	Interrupt service routine
JSA	Jazyk symbolických adries
ksp	jednotka frekvencie vzorkovania
LED	Light eluminated diode
LQFP-32	druh puzdra pre elektronické súčiastky
MHz	megahertz, jednotka frekvencie (10^6 Hz)
MIPS	Million instruction per second (jednotka rýchlosti spracovania dát)
MOSFET	Metal oxide semiconductor field effect transistor
PC	Program counter
PMSM	Permanent magnet synchronous motor

PWM	Pulse – width modulation
RAM	druh pamäte
RI	Register inštrukcií
ROM	druh pamäte
RS232	sériová linka, štandard prenášania dát
RST	Reset
RWM	druh pamäte
SFR	Special function register
SMD	technológia výroby puzdier súčiastok
SO8	druh puzdra pre elektronické súčiastky
SPI	druh komunikácie medzi elektronickými zariadeniami
TTL	Logické bipolárne obvody
UART	Universal asynchronous receiver/tranceiver (iné označenie RS232)
USB	Universal serial bus
V/V	vstup/výstup
V	Volt, jednotka elektrického napätia
W	Watt, jednotka elektrického výkonu

Úvod

Človek sa už od dávnych čias snaží dosiahnuť čo najväčší komfort vo svojom živote. Táto dlhotrvajúca snaha vyústila práve k vytvoreniu jedného z najkomplexnejších a najzložitejších systémov, bez ktorých si už človek v dnešnej dobe ani nevie predstaviť život. Týmito systémami sú počítače, ktoré prešli najrýchlejším vývojom za posledné desaťročia. Sú obľúbené z rôznych dôvodov, najmä preto, že zvládajú široký sortiment úloh, ktoré sú náročné na výpočtový výkon a dávajú tiež možnosť pripojenia periférií ako sú monitor, klávesnica či tlačiareň, ale taktiež umožňujú používateľovi pohodlný spôsob ovládania iných zložitých systémov, ktoré by bez riadiaceho PC nebolo možné ovládať.

Nie vždy sú však počítače vhodnou variantou pre riešenie zložitých aplikácií. Ako príklad uvediem veľmi rýchlo sa rozvíjajúce systémy ako napr. mobilný telefón, alebo tiež mnohé iné podobné zariadenia. Tento systém potrebuje tiež k svojej činnosti výpočtovú jednotku, ktorá však nemusí v sebe zahŕňať veľké množstvo operácií, ale sa špecializuje na určitú úlohu ako napríklad sluchová či písomná komunikácia. Táto výpočtová jednotka je súčasťou komplexnejšieho systému a tak sa dostávame práve k pojmu vstavané systémy. Tieto systémy sú schopné bez použitia počítača vykonávať špecializované úlohy a reagovať na rôzne vonkajšie udalosti. Údaje, ktoré pritom spracovávajú, nie sú priamo viditeľné pre užívateľa ako pri PC na monitore, ale to pri daných zariadeniach nie je podstatné.

Využitie vstavaných systémov si vyžaduje špeciálnu jednotku, ktorú nazývame mikrokontrolér. Mikrokontroléry sa od klasických mikroprocesorov odlišujú tým, že ich čip je okrem základnej procesorovej jednotky CPU vybavený aj rôznymi druhmi periférií ako napr. V/V obvody, A/D a D/A prevodníkmi, a iné. Poznáme rôzne druhy mikrokontrolérov od rôznych výrobcov ako Silabs, Atmel, Microchip, Motorola a iné. Človek venuje stále viac pozornosti rozvoju vstavaných systémov, či už vo sfére vedy, techniky alebo aj zábavy. Veď ako príklad môžeme uviesť aj zdravotnícke zariadenia, číslicové meradlá a iné. Tieto vstavané systémy tiež nachádzajú svoje miesto aj v oblasti mobilnej robotiky a práve tou sa zaoberám v diplomovej práci.

1. Prehľad o súčasnom stave riešenej problematiky

1.1 BLDC motory

Bezkomutátorové jednosmerné motory, z anglického brushless DC motors, patria do skupiny synchronných trojfázových motorov. Ich rotor je zložený z permanentných magnetov a stator tvorí trojica budiacich cievok vzájomne posunutých o 120° .

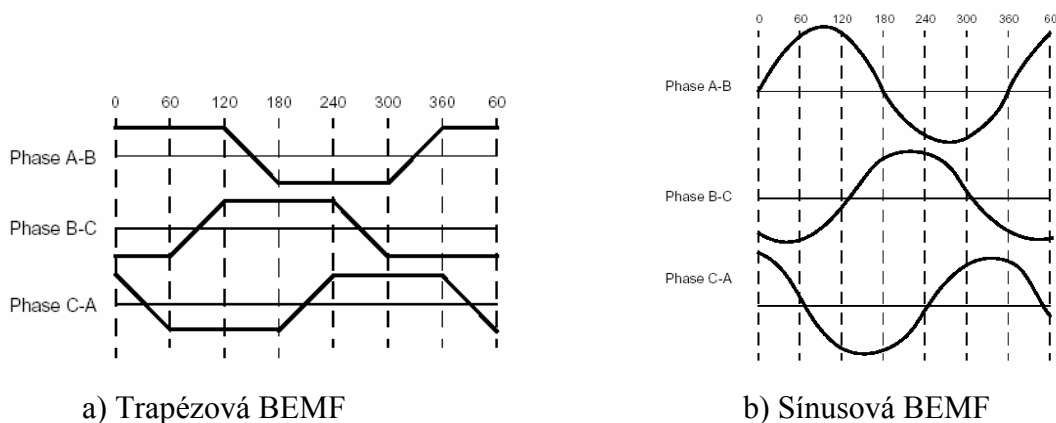
BLDC motory s permanentnými magnetmi nahrádzajú bežné komutátorové motory v mnohých aplikáciách, pričom poskytujú vyššiu účinnosť, životnosť, výkonnosť a nižšiu hlučnosť, ktorá je pri bežných jednosmerných motoroch spôsobovaná mechanickým komutátorom.

1.1.1 BLDC motor - základné časti

▪ **Stator** – je zložený z tenkých, navzájom izolovaných plechov s drážkami po obvode, v ktorých je uložené statorové vinutie. Prevedenie je rovnaké ako pri indukčných motoroch, ibaže vinutia sú uložené v drážkach v odlišnej konfigurácii. Väčšina BLDC motorov má tri statorové vinutia zapojené do hviezdy. Tie sú po obvode statora rozdelené v drážkach tak, aby sa dosiahla čo najväčšia efektívnosť motora. Existujú dva typy statorových vinutí:

- trapézové
- sínusové

Táto diferenciácia je založená na rôznych prepojeniach cievok, kedy vznikajú rôzne priebehy spätnej indukcie elektromotorických napätí zobrazených na obr.1.

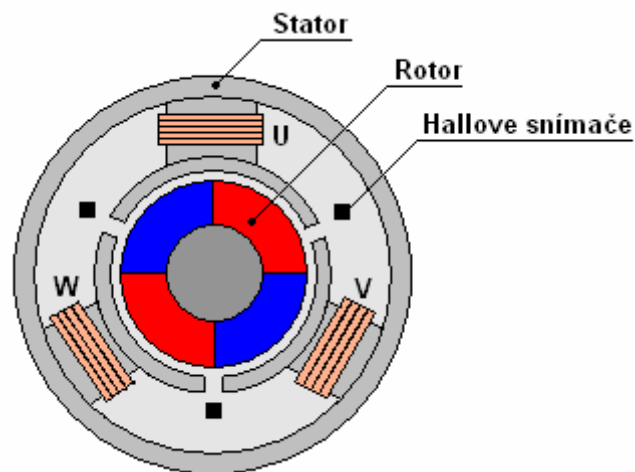


Obr.1 Typy BEMF

Podľa typu vinutia sa potom volí tiež priebeh prúdu prechádzajúceho vinutím. Výsledkom toho je rôzny krútiaci moment motora, pričom pri sínusovom zapojení cievok dostávame hladší priebeh. Preto sa tieto motory označujú aj ako PMSM (Permanent magnet synchronous motor). Používajú sa na presné servomechanizmy riadené na konštantné otáčky prípadne polohu. Ich nevýhoda je vyššia nadobúdacía cena a výpočtová náročnosť riadiaceho algoritmu než u BLDC motorov, ktoré majú trapézové rozloženie BEMF, z čoho vyplýva aj nevhodnosť použitia pre aplikácie s presným polohovaním rotora.

- **Rotor** – je vyrobený z permanentných magnetov tak, aby tvoril dva až osem pólových dvojíc. Ako materiály pre výrobu týchto magnetov sa používajú ferity (lacnejšie), neodymium (Nd), samáriumkobalt (SmCo), zliatiny neodymia, feritu a bóru (NdFeB) a niektoré druhy prírodných magnetov.

- **Hallové senzory** - pretože BLDC motory sú riadené elektronicky je potrebné poznať polohu rotora na spustenie riadiacej sekvencie - komutácie. Na to slúžia Hallové senzory zabudované v statore. Väčšina týchto motorov má tri senzory, ktoré sa aktivujú pri priblížení niektorého z pólov rotora a podľa pólu N alebo S indikujú vysokú alebo nízku úroveň signálu. Podľa kombinácie signálov z troch sensorov možno teda určiť presnú polohu rotora potrebnú na spustenie riadiacej sekvencie. Na zistenie polohy rotora však môže slúžiť aj meranie BEMF, ktoré bude samostatne popísané neskôr. Na obr. 2 je zobrazená štruktúra BLDC motora.



Obr. 2 Štruktúra BLDC motora

1.1.2 BLDC motor – matematický opis

Matematický opis sa dá rozdeliť na elektrickú a mechanickú časť.

1.1.2.1 Opis elektrickej časti BLDC motora

$$\frac{d}{dt}i_a = \frac{1}{3L_s}(2v_{ab} + v_{bc} - 3R_s i_a + \lambda p \omega_r (-2\Phi_a' + \Phi_b' + \Phi_c')) \quad (1)$$

$$\frac{d}{dt}i_b = \frac{1}{3L_s}(-v_{ab} + v_{bc} - 3R_s i_b + \lambda p \omega_r (\Phi_a' - 2\Phi_b' + \Phi_c')) \quad (2)$$

$$\frac{d}{dt}i_c = -\frac{d}{dt}i_a - \frac{d}{dt}i_b \quad (3)$$

$$T_e = p\lambda(\Phi_a' i_a + \Phi_b' i_b + \Phi_c' i_c) \quad (4)$$

kde:

- i_a, i_b, i_c – prúdy tečúce jednotlivými cievkami statora
- v_{ab}, v_{bc} – rozdielové napätia na jednotlivých statorových vinutiach
- R_s – odpor cievky statorového vinutia
- L_s - indukčnosť cievky statorového vinutia
- $\Phi_a', \Phi_b', \Phi_c'$ - rozloženie spätnej indukcie elektromotorického napätia rotorom do statorového vinutia (BEMF)
- λ – magnetický tok permanentného magnetu
- ω_r – uhlová rýchlosť rotora
- p - počet pólových dvojíc
- T_e – elektromagnetický krútiaci moment

Rovnice tejto časti motora opisujú chovanie zmien prúdov tečúcich vinutiami cievok statora na aplikovanom svorkovom napätí, odčítanú o hodnotu napätia z parazitných odporov cievok a BEMF - napätia spätne indukovaného rotorom motora. Pretože zapojenia cievok BLDC motora sú do hviezdy stačí k popisu tretieho prúdu vzťah $i_c = -i_a - i_b$ (1. Kirchhoffov zákon). Časová konštanta nábehu prúdu cievky pri

pripojení napätia nadobúda hodnotu $\frac{R}{L}$. Táto hodnota je slúži pri nastavovaní spínacej frekvencie PWM generátora. Rozloženie BEMF je funkciou elektrického uhlu natočenia rotora $\Phi_{a,b,c} = f(\varphi_e)$.

Hodnota elektromagnetického krútiaceho momentu závisí na počte pólových dvojíc motora, na hodnote magnetického toku rotora, jeho polohe a prúdov tečúcich cievkami statora.

1.1.2.2 Opis mechanickej časti BLDC motora

$$\frac{d}{dt}\omega_r = \frac{1}{J}(T_e - F\omega_r - T_m) \quad (5)$$

$$\frac{d\varphi}{dt} = \omega_r \quad (6)$$

$$\varphi_e = p \cdot \varphi \quad (7)$$

kde:

- ω_r - uhlová rýchlosť rotora
- φ - uhol natočenia rotora
- φ_e - uhol elektrického natočenia rotora
- J – kombinovaná zotrvačnosť statora a záťaže
- p – počet pólových dvojíc
- T_e – elektromagnetický krútiaci moment
- T_m – mechanický krútiaci moment záťaže
- F – viskózne trenie

Opis mechanickej časti udáva informáciu o dynamických vlastnostiach motora. Derivácia uhlovej rýchlosti rotora je priamo úmerná krútiacemu momentu vyvolanom cievkami statora, záťažou a viskóznym trením. Derivácia uhlovej rýchlosti je nepriamo úmerná kombinovanej zotrvačnosti rotora a záťaže. Zmena uhlu natočenia rotora je závislá na uhlovej rýchlosti rotora.

1.2 Pulse width modulation – Impulzne šírková modulácia

Impulzne šírková modulácia signálu je technika generovania impulzného priebehu s konštantnou periódou a premennou dobou pracovného cyklu. Tento druh signálu sa používa predovšetkým na prenos informácií cez komunikačné kanály, vytváranie zvukových efektov alebo na kontrolu množstva výkonu posielaného do záťaže, čo je pre nás podstatné.

1.2.1 Matematický princíp

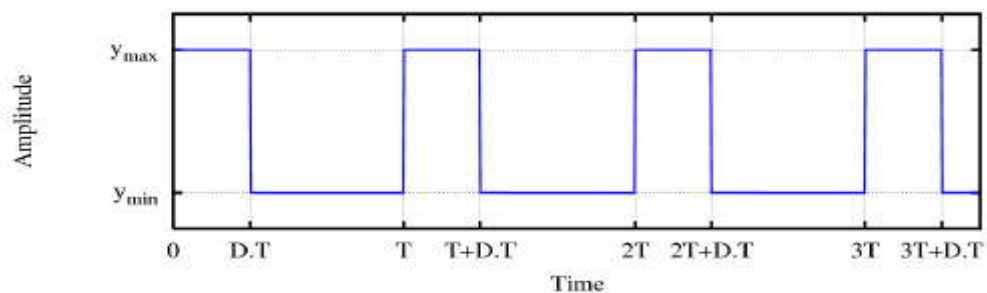
Impulzne šírková modulácia používa obdĺžnikový signál, ktorého doba pracovného cyklu je modulovaná na základe zmeny priemernej hodnoty priebehu. Ak vezmeme do úvahy obdĺžnikový priebeh $f(t)$ s najnižšou hodnotou y_{\min} , s najvyššou hodnotou y_{\max} a s dobou pracovného cyklu D (obr. 3), tak priemerná hodnota priebehu je daná ako:

$$\bar{y} = \frac{1}{T} \int_0^T f(t) dt \quad (1)$$

Keďže $f(t)$ má obdĺžnikový priebeh tak hodnota y_{\max} dosahuje svoje hodnoty pre interval $0 < t < D.T$ a y_{\min} pre interval $D.T < t < T$. Po dosadení do rovnice (1) v tejto kapitole dostávame:

$$\bar{y} = \frac{1}{T} \left(\int_0^{D.T} y_{\max} dt + \int_{D.T}^T y_{\min} dt \right) = \frac{D.T y_{\max} + T(1-D)y_{\min}}{T} = D y_{\max} + (1-D)y_{\min} \quad (2)$$

Výraz (2) môže byť značne zjednodušený v prípade, že $y_{\min} = 0$ a potom $\bar{y} = D y_{\max}$. Z toho je zrejmé, že priemerná hodnota signálu \bar{y} je priamoúmerne závislá na dobe pracovného cyklu D . (http://en.wikipedia.org/wiki/Pulse-width_modulation).



Obr.3 Priebeh PWM signálu

1.2.2 Princípy generovania

Existujú 4 spôsoby generovania impulzne šírkovy modulovaného signálu, a to:

- *Pretínajúci* – je najjednoduchším možným spôsobom generovania PWM signálu, ktorý vyžaduje iba pilovitý alebo trojuholníkový priebeh ľahko generovateľný obyčajným oscilátorom a komparátorom. Keď hodnota referenčného signálu je väčšia ako hodnota modulovaného priebehu, tak PWM signál je vo vysokej hladine, v opačnom prípade je v hladine nízkej.

- *Delta* – výstupný signál je porovnávaný s medznými hodnotami, ktoré sa zhodujú s referenčným signálom o konštantnú odchýlku. Zakaždým, keď výstupný signál dosiahne jednu z medzných hodnôt, PWM signál zmení stav

- *Sigma-Delta* – výstupný signál je odčítaný z referenčného signálu na tvar signálu chyby. Táto chyba je integrovaná a keď integrál chyby prekročí medzné hodnoty, výstup zmení svoj stav

- *Číslicový (digitálny)* – mnoho číslicových obvodov ako napr. aj mikrokontroléry majú PWM výstup. Tento fakt má v našom prípade veľký význam práve kvôli možnosti ovládania DC motora. Spravidla na to používajú časovač, ktorý sa periodicky inkrementuje (je napojený priamo či nepriamo na zdroj hodinového signálu) a je resetovaný pri každej perióde PWM signálu. Keď je hodnota časovača väčšia ako referenčná hodnota, PWM výstup zmení svoj stav z vysokej hladiny na nízku a naopak.

1.2.3 Typy PWM modulácie

Existujú tri možné spôsoby PWM modulácie:

- a) Stred impulzu je sústredený v strede časového rámca a k zmene šírky impulzu dochádza sťahovaním a rozťahovaním obidvoch hrán impulzu
- b) Nástupná hrana je držaná v čele časového rámca a zostupná hrana je modulovaná
- c) Zostupná hrana je držaná a nástupná je modulovaná

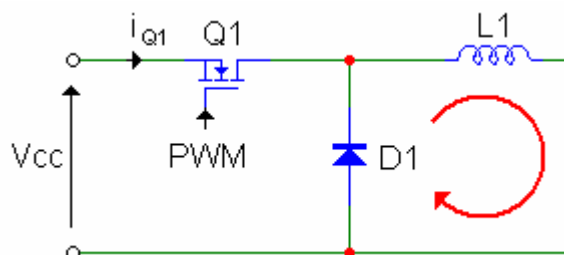
Výhodou PWM je, že výkonový tranzistor pracuje v spínacom režime. Ak je zopnutý, prechádza ním síce veľký prúd, ale úbytok na ňom je veľmi malý, rádovo niekoľko desiatín V. Ak je naopak rozpojený, prechádza ním len malý zbytkový prúd. V oboch prípadoch je výkonová strata pomerne malá. Tranzistor je však zaťažovaný pri prechodoch medzi jednotlivými stavmi. Prechody neprebiehajú okamžite, ale potrebujú určitý konečný čas (rádovo μs , pričom vypnutie je niekoľkonásobne dlhšie ako zopnutie) a behom nich sa tranzistor zahrieva. Stratový výkon a tiež teda zahrievanie sa preto zväčšujú s rastúcou frekvenciou ovládacieho PWM signálu. Preto treba vhodne voliť výkonový člen podľa katalógového listu. (Hlava, 2000 s.76).

Vzhľadom k malému prúdovému zosilňovaciemu činiteľovi výkonových tranzistorov sa na spínanie obvykle používa Darlingtonovo zapojenie tranzistorov. Bežne sa takéto zapojenie vyrába v jednom puzdre ako Darlingtonov tranzistor. Tiež je možné použiť integrované výkonové obvody na riadenie motorov, poprípade tranzistory MOSFET a IGBT, ktoré sa tiež využívajú s veľkou obľubou v takýchto zapojeniach.

1.2.4 Ochranná dióda

Motor predstavuje indukčnú záťaž, a preto sa v zapojení vyskytuje ochranná dióda. Na jednoduchom príklade si vysvetlíme jej funkciu. Indukcia poľa budenia je znázornená na obr. 4 ako L1. Rezistivita vinutí nie je v tomto prípade dôležitá, a preto nie je zakreslená.

Q1 je MOSFET tranzistor a pri jeho otvorení ním prechádza prúd i_{Q1} vinutím. V prípade vypnutia Q1 nezanikne prúd vinutím okamžite lebo vytvorené magnetické pole v okolí statora (jeho zmena), ho tlačí ďalej vinutím v tom istom smere ako zobrazuje červená šípka na obr. 4. Okruh sa uzavrie cez diodu D1, ktorá bráni poškodeniu Q1, lebo takýmto spôsobom sa na vinutí indukuje vysoké napätie (pri výkonnejších motoroch aj niekoľko jednotiek kV).

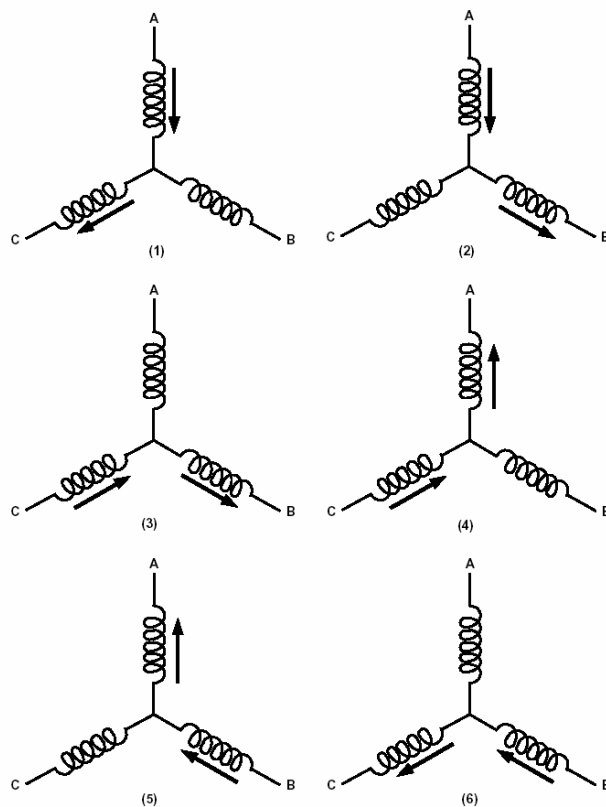


Obr. 4 Zapojenie ochrannej diody

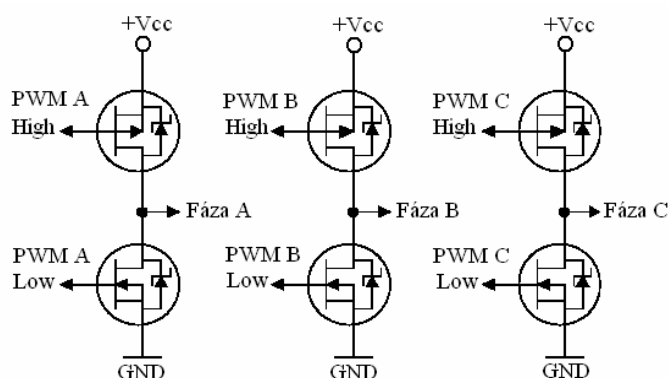
1.3 Riadenie BLDC motorov

Na riadenie trojfázových bezkomutátorových motorov sa používa trojpolomostíková „invertorová“ topológia – trojfázový H mostík, kde riadiacim signálom je tzv. six-step signál modulovaný pomocou PWM modulácie. Pri elektronickej komutácii jednotlivých fáz motora treba tiež rešpektovať aj synchronizáciu medzi statorovým magnetickým tokom a permanentnými magnetmi rotora, ktorú zabezpečí riadiaca elektronika na základe informácie získanej z Hallových senzorov alebo pomocným obvodom na detekciu BEMF.

Na obr. 5 je znázornený tzv. six-step signál, ktorý je potrebný na vytvorenie rotujúceho magnetického poľa. Šípky znázorňujú tok prúdu statorovými vinutiami v jednotlivých krokoch sekvencie. Z uvedeného obrázka teda vyplýva spínanie jednotlivých výkonových prvkov v 3 fázovom H – moste, ktorý je uvedený na obr. 6.



Obr. 5 Vytvorenie rotujúceho magnetického poľa
signálom six-step



Obr. 6 Trojfázový H-mostík

1.3.1 Riadenie BLDC motora PWM signálom v H – moste

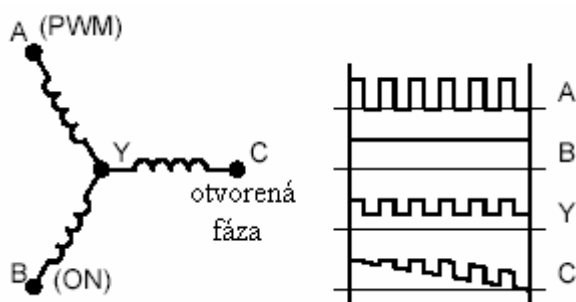
Tento druh riadenia možno rozdeliť podľa použitej komutácie na :

a) *Low-side PWM komutácia* - PWM signálom sú riadené iba tranzistory označené ako „Low“ (obr. 6) . V závislosti na pozícii rotora, ktorá je vydedukovaná z kódu Hallových snímačov , prípadne z BEMF merania, sú vygenerované impulzno-šírkovo modulované signály, ktorými sú spínané „Low“ tranzistory. V tomto prípade tranzistory označené ako „High“ sú zopnuté v stave log. 1. Tento druh komutácie sa nazýva „ Low-side PWM komutácia“ a je zobrazená v tab. 1.

Tab. 1 Low-side PWM komutácia

Krok	„High“ tranzistor			„Low“ tranzistor			Otvorená fáza
	A	B	C	A	B	C	
1.		ON		PWM			C
2.			ON	PWM			B
3.			ON		PWM		A
4.	ON				PWM		C
5.	ON					PWM	B
6.		ON				PWM	A

V súvislosti s fázami motora je teda jedna fáza v stave log.1, ďalšia fáza je impulzne-šírkovo modulovaná a tretia fáza je otvorená. Tretia fáza sa využíva na meranie BEMF pri takto riadených motoroch. Ide o napätie medzi otvorenou fázou a nulovým, stredným vodičom. Fázy a napätia na jednotlivých fázach sú zobrazené na obr. 7.



Obr. 7 Napätia na jednotlivých fázach pri „Low-side“ komutácii

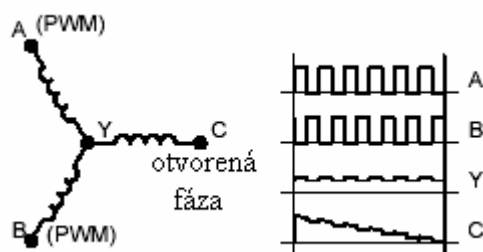
Problémom je, že v bode Y nie je konštantné DC napätie. Za predpokladu, že rezistivita a indukčnosť vinutí A a B sú rovnaké, bude napätie v bode Y polovičné ako napätie medzi týmito fázami. Napätie v bode Y bude teda polovičné z napájacieho napätia ak je fáza A v stave log.0. Ak je fáza A v stave log.1 bude napätie v tomto bode blízke napájaciemu napätiu. Napätie na fáze C je teda súčtom napätia v bode Y a BEMF napätia a je zároveň uzatvárané cez ochrannú diodu MOSFET tranzistora. Výsledkom je pulzne modulovaný tvar vlny, ktorej minimálne napätie je úmerné BEMF motora. Filtrácia PWM signálu alebo vzorky napätia pri takejto komutácii je príliš zložitá a preto sa oveľa častejšie používa symetrická PWM komutácia.

b) *Symetrická PWM komutácia* – ide o komutáciu, kedy sú spínané tranzistory označené ako „High“ aj „Low“ podľa PWM súčasne. Je teda značne podobná Low-side komutácii a je zobrazená v tab. 2.

Tab. 2 Symetrická PWM komutácia

Krok	„High“ tranzistor			„Low“ tranzistor			Otvorená fáza
	A	B	C	A	B	C	
1.		PWM		PWM			C
2.			PWM	PWM			B
3.			PWM		PWM		A
4.	PWM				PWM		C
5.	PWM					PWM	B
6.		PWM				PWM	A

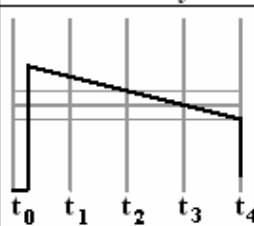
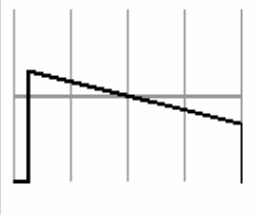
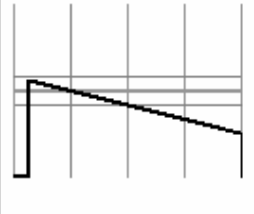
Počas jednotlivých krokov sú na oba typy tranzistorov, „High“ aj „Low“, privádzané identické PWM signály, ktoré spínajú jednotlivé fázy. V prvom kroku je výsledkom, že fáza A prechádza do stavu log.0 v čase keď fáza B prechádza do stavu log.1. Ak rezistivita a indukčnosť vinutí A a B sú rovnaké, bude napätie v bode C totožné s BEMF napätím. V tomto bode sa tiež môže vyskytnúť zvyškový PWM šum, prípadne rušivé napätia spôsobené nevyváženosťou vinutí alebo parazitnou kapacitou. Tieto signály sú zanedbateľné oproti žiadanému signálu BEMF, ako zobrazuje obr. 8.



Obr. 8 Napätie na jednotlivých fázach pri symetrickej komutácii

1.3.2 BEMF – meranie, využitie na riadenie

Väčšina bezsenzorových BLDC riadiacich systémov využíva meranie času pri prechode BEMF nulou v uzavretej slučke a tento čas vyhodnocuje ako riadiacu premennú. Z obr. 9 vyplýva princíp tohto merania. Pribeh BEMF vlny je rozdelený do časových úsekov t_0 až t_4 s presne stanovenou dobou trvania. Aby bola dosiahnutá správna rýchlosť a správne napätie, je nutné, aby BEMF vlna prechádzala nulov práve v časovom okamihu t_2 . V ostatných prípadoch je nutné doregulovať napätie aj rýchlosť.

Tvar vlny	Rýchlosť	Napätie
	vysoká	nízke
	správna	správne
	pomalá	vysoké

Obr. 9 Princíp merania BEMF

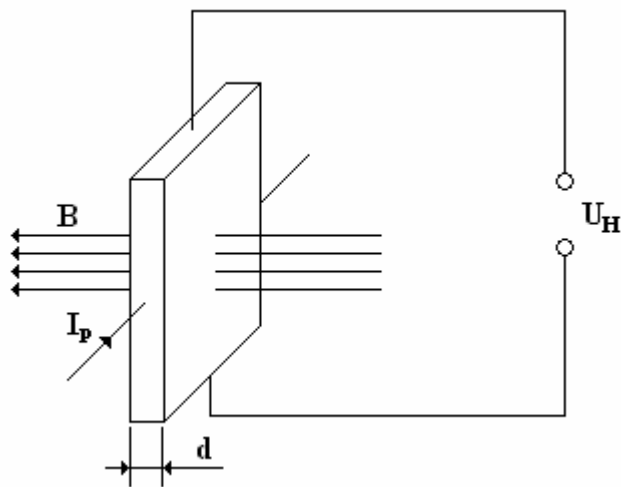
Okrem merania času prechodu BEMF nulou je možné tiež merať napäťové vzorky v jednotlivých časových úsekoch. Tento spôsob ponúka vyššie rozlíšenie a nie je až tak náročný na výpočet. Nehodí sa však na motory veľmi malého výkonu, pretože potrebuje určitú minimálnu hodnotu BEMF napätia.

Z uvedeného vyplýva, že na zistenie presnej polohy rotora pre spustenie komutácie je nutné vinutia najskôr vybudíť určitým napätím a až potom je možné spustiť komutáciu v správnom kroku. Toto je možné obísť použitím Hallových

snímačov magnetického poľa, ktoré sú v niektorých motoroch zabudované a samotné riadenie sa tak stáva jednoduchšie. Obstarávacia cena takýchto motorov je však vyššia.

1.3.3 Princíp a použitie Hallových snímačov na riadenie motora

Hallove snímače využívajú princíp Hallovho javu, kedy magnetické pole s indukciou B pôsobí na polovodičový kváder s hrúbkou d , ktorou prechádza prúd I_p tak, že na vedľajších stranách kvádra sa generuje Hallovo napätie. Tento princíp vidieť na obr. 10.



Obr. 10 Hallov jav

Pre výpočet Hallovho napätia platí vzťah:

$$U_H = R_H \frac{I_p \cdot B}{d} \quad (1)$$

kde Hallova konštanta R_H sa vypočíta:

$$R_H = \frac{3}{8 \cdot n_q \cdot q} \quad (2)$$

kde: n_q – hustota nosičov náboja

q – elementárny náboj

d – hrúbka polovodiča

Z uvedeného vyplýva, že na Hallov senzor je nutné najskôr priviesť určitý prúd I_p a až potom je možné pri priblížení magnetických pólov rotora detekovať jeho natočenie. V tomto prípade však nedochádza ku kmitavému pohybu rotora pri zisťovaní jeho natočenia ako pri meraní BEMF. Preto sa tento systém lepšie uplatní pri konštrukciách presných servomechanizmov. Tieto senzory sú umiestnené na kostre motora tak, aby mohli detekovať prítomnosť magnetickej indukcie pólov rotora, pričom sú vzájomne posunuté o 120° .

1.4 Jednočipové mikropočítače

1.4.1 Štruktúra jednočipových mikropočítačov

Jednočipový mikropočítač, v priemysle nazývaný aj mikrokontrolér alebo mikroradič, je taký integrovaný obvod, ktorý na jednom čipe združuje mikroprocesor, generátor hodinového impulzu, vstupno/výstupné brány a pamäť pre dáta aj program v takom rozsahu, že je schopný samostatne pracovať. Vyznačujú sa nízkou spotrebou a dostatočnou hardvérovou vybavenosťou pre riadenie jednoduchých aplikácií. Prvý mikrokontrolér bol vyrobený v roku 1977 od firmy INTEL a mal označenie 8048 (bol 8-bitový). Mikroradiče sa stali postupne štandardom a v súčasnosti sa už vyrábajú aj 32-bitové. Medzi najznámejšie výrobné firmy patrí INTEL – rada MCS, ATMEL, Microchip PIC. Ich základnou vlastnosťou je ich programovateľnosť.

Mikrokontroléry sa používajú na jednoduchšie aplikácie v meracej, riadiacej, regulačnej a spotrebnej technike. Sú určené na také operácie kde by iné mikroprocesory neboli využité a boli by cenovo nevýhodné. Väčšinou sa používajú na riadenie strojov v reálnom čase (obchodné váhy, výtahy, šijacie stroje, meracie prístroje, univerzálne regulátory,.....). Od univerzálnych mikroprocesorov, ktoré sú konštruované podľa klasickej Von Neumanovej štruktúry počítača (jedna pamäť pre dáta aj program) sa mikroradiče zásadne líšia tým, že majú od seba oddelenú pamäť na vloženie programu a pamäť na uloženie údajov – Harwardská štruktúra. Na čipe býva pamäť programu typu ROM alebo EPROM, pamäť údajov typu RAM alebo RWM.

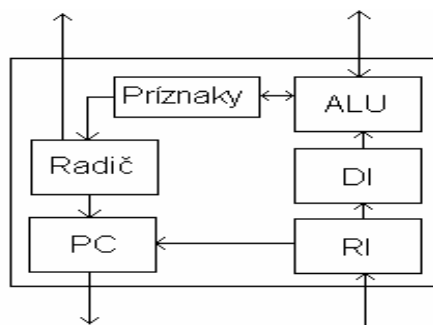
Štruktúru monolitického mikroradiča možno rozdeliť na tri časti na:

- 1.pamäte
- 2.procesorová časť
- 3.vstupno/výstupné obvody (brány)

Tieto časti sú prepojené zbernicami, ktoré delíme na dátovú, adresnú a riadiacu.

V procesorovej časti, obr.11, radič riadi vykonávanie operácií a chod celého procesora podľa inštrukcií programu. Pre svoju činnosť potrebuje register inštrukcií RI, ktorý uchováva operačný kód práve vykonávanej inštrukcie po dobu jej vykonania. Výstupy z tohto registra sú pripojené do dekodéra inštrukcií DI, ktorý obsah dekóduje a s ostatnými pomocnými obvodmi generuje riadiace signály pre radič a ALU (aritmeticko-logická jednotka).

PC – programový čítač obsahuje adresu nasledujúcej inštrukcie a po vykonaní inštrukcie sa PC automaticky zvyšuje. Príznakové bity sa nastavujú podľa vykonania operácie a využívajú inštrukcie pre vetvenie.

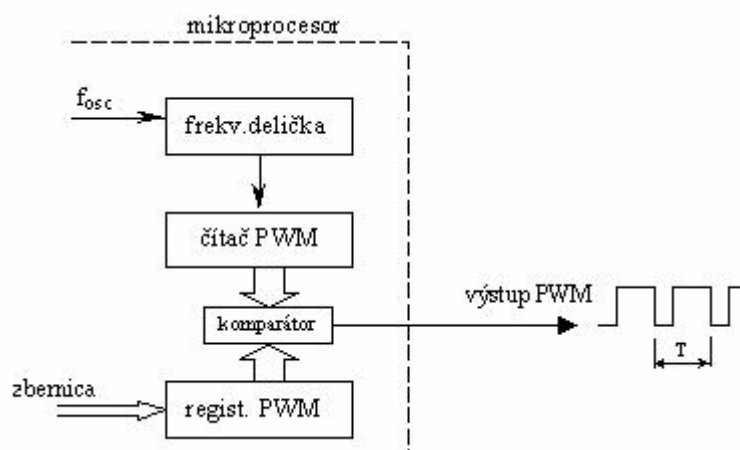


Obr.11 Bloková schéma procesorovej časti

Popísaná štruktúra a blokové schémy sú spoločným základom pre všetky mikropočítače. S pokrokom doby sa už iba zvyšujú nároky na jednotlivé obvody najmä pamäťové a sú vybavené rôznymi doplnkovými funkciami. Každý rad mikroradičov je preto osobitný s danými funkciami a možnosťami, ktoré sa dajú jednoducho zistiť z firemných literatúr ako sú napr. datasheets (katalógové listy). Datasheet tvorí neoddeliteľnú súčasť každého mikropočítača. Obsahuje technické údaje, spôsob programovania, funkcie a mnoho iných popisov, ktoré sú potrebné pre samotný návrh konštrukcie.

1.4.2 Mikrokontroléry s generátorom PWM signálu

V dnešnej dobe už existujú aj mikrokontroléry, ktoré majú zabudovaný generátor PWM signálu priamo na čipe a generovanie tohto signálu je znázornené na obr.12.



Obr.12 Štruktúra generátora PWM

Do frekvenčnej deličky sa privádza hodinový signál f_{osc} , ktorý môže ostať nezmenený alebo sa delí konštantou. Deliacu konštantu možno zvoliť nastavením príslušného riadiaceho registra. Čítač PWM je voľne bežiace počítadlo, ktorý počíta hodinový signál z frekvenčnej deličky. Obsah tohoto počítadla je komparátorom porovnávaný s registrom PWM, do ktorého zadávame požadovanú hodnotu. Na výstupe komparátora dostávame PWM signál – ak je obsah registra PWM a čítača PWM rovnaký je na výstupe log. 0, ak sú rôzne je na výstupe log.1. Strieda výstupného signálu je úmerná obsahu registra PWM a perióda je daná pevne počítaným signálom a módom počítadla.

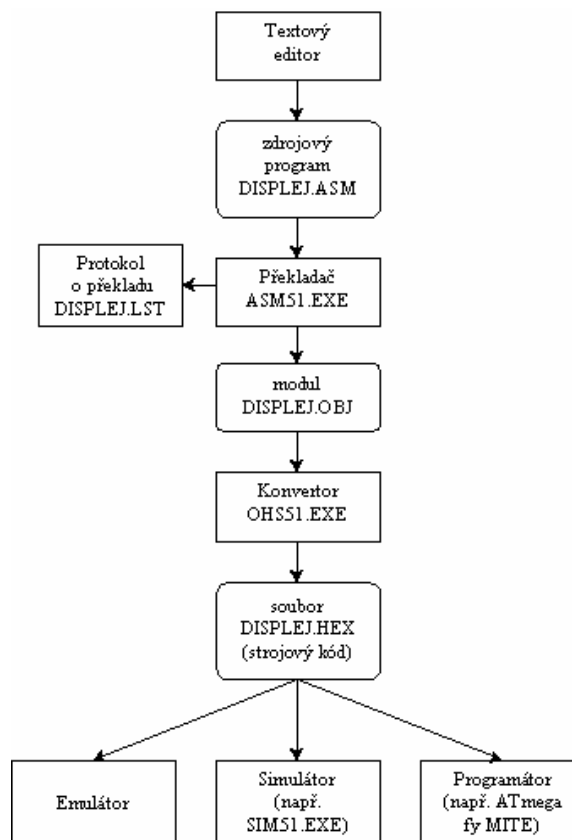
1.5 Programovanie jednočipových mikropočítačov

Pri vývoji programu pre ktorýkoľvek typ mikrokontroléra je potrebné poznať nielen jeho architektúru, ale aj možnosti jeho inštrukčného súboru. Programovanie tvorí kapitolu samú o sebe, tak sa pokúsím zhrnúť informácie o najčastejšie využívaných spôsoboch.

1.5.1 Programovanie v JSA

Jazyk symbolických adres má najbližšie k strojovému kódu mikroprocesora, preto pri programovaní v ňom dosiahneme najefektívnejšieho programu – bude rýchly a pamäťovo najmenej náročný.

Zdrojový program je možné písať v ľubovlnom textovom editore, ktorý do výsledného súboru nepridáva svoje vlastné formátovacie alebo riadiace informácie. Možno teda použiť Poznámkový blok z Windows 95/98/Me/NT/2000/XP. Ďalším krokom je preloženie programu, čo umožňujú rôzne prekladače. Tie zaisťujú kontrolu správnosti zápisu zdrojového programu a vytvárajú pomocné súbory, ktoré slúžia na vytvorenie konečného súboru vo formáte HEX, čo je aj jeho prípona. Takto vytvorený súbor zo zdrojového programu môže byť nahraný do mikropočítača pomocou programátora, čo je aj posledným krokom naprogramovania mikropočítača. Celý proces vývoja programu je však názornejší z obr.13



Obr.13 Proces vývoja programu

1.5.1.1 Formát zdrojového programu v JSA

Program v JSA zapisujeme po riadkoch, kde na jednom riadku môže byť len jedna inštrukcia alebo pseudoinštrukcia. Celý riadok sa skladá zo 4 častí, ktoré je dobré od seba pre lepšiu prehľadnosť oddeľovať aspoň jednou medzerou. Týmito štyrmi časťami sú:

1. návěstie alebo symbolické meno
2. inštrukcia alebo pseudoinštrukcia
3. operandy
4. komentár

Príklad pre inštrukciu				Príklad pre pseudoinštrukciu			
1	2	3	4	1	2	3	4
START:	MOV	A,P1	;komentár	STOVKA	EQU	64H	;komentár

Obr.14 Príklady

Ako je z príkladov vidieť, návěstie sa od symbolického mena odlišuje tým, že má na konci dvojbodku. Inštrukcia môže mať 0 až 3 operandy, pseudoinštrukcia má vždy aspoň 1 operand.

Medzi najdôležitejšie pseudoinštrukcie patria:

ORG – nastavuje pri preklade čítač inštrukcií PC na danú hodnotu. Ak nie je použitá preklad začína automaticky v programovej časti od adresy 000H. Pred ORG nesmie byť návěstie.

END – indikuje koniec zdrojového textu a končí prácu prekladača

EQU – priraduje výraz ku zvolenému symbolickému menu. Výrazom môže byť konštanta, konkrétna adresa, vyhradený symbol prekladača (R0,R1,...)

BIT, DATA, IDATA, XDATA, CODE – podobne ako EQU

DB, DW – slúžia k uchovaniu konštánt v programovej pamäti.

1.5.1.2 Štruktúra zdrojového programu

Hlavička programu

```
;*****názov*****           ; len ako informácia pre programátora  
;**popis činnosti**
```

Popisná časť – určuje priradenie adries premenným, hodnoty konštant,...

```
P1      EQU  90H           ; port P1 priradí na adresu 90H  
LED     BIT   P1.06H      ; symbolickú LED priradí na 6.pin portu P1  
...                                           ; ďalšie pseudoinštrukcie  
...
```

Hlavný program

```
ORG     000H           ; začiatok programu  
....           ; jednotlivé inštrukcie alebo pseudoinštrukcie  
....  
....  
END           ; koniec programu
```

Takto písané inštrukcie platia pre verzie založené na jadre 8051. Od ostatných verzií sa môže sada inštrukcií líšiť.

1.5.2 Programovanie vo vyššom programovacom jazyku C

Jazyk C je jedným zo základných kameňov vyšších programovacích jazykov. Je to univerzálny kompilačný programovací jazyk vyššej úrovne s jazykovými prostriedkami, ktoré dovoľujú manipuláciu s údajmi aj na nižšej úrovni. Univerzálnosť tohto jazyka je v tom že jeho použitie nie je obmedzené na úzku aplikačnú oblasť.

Je vhodný pre:

- systémové programovanie
- vedecko-technické výpočty spracovanie hromadných údajov
- riadenie v reálnom čase

Tento jazyk nie je viazaný na konkrétny operačný systém. Väzby sa realizujú pomocou systémových knižníc, preto je veľká prenositeľnosť tohto jazyka.

Väzby na operačný systém, ktoré sú realizované knižnicami :

- vstup/výstup práca s takýmito zariadeniami
- prístup k súborom
- pridelovanie pamäte
- multiprogramovanie
- vytváranie a synchronizácia procesov

Verzie jazyka C :

- Definícia podľa Kernighana a Richieho (K&R C)
- ANSI C
- Verzie podporované rozličnými kompilátormi (Turbo C, Borland C ...)

1.5.2.1 Konštrukcia programu

Konštrukcia programu je veľmi jednoduchá, lebo tento jazyk je štruktúrovaný do prehľadných blokov. Základný blok, ktorému sú všetky ostatné bloky podriadené, je funkčný blok, označovaný ako funkcia. Celý program sa potom skladá z množiny funkcií, ktoré obsahujú a spracovávajú dáta a premenné a prostredníctvom nich vzájomne komunikujú. Použitie funkcií sa uskutočňuje prostredníctvom podprogramov, procedúr atď. Pri mikrokontroléroch sa pred vyvolaním programu spúšťa počiatočný (*startup*) modul väčšinou napísaný v JSA. Prostredníctvom neho sa uskutočňuje základná inicializácia rôznych registrov mikrokontroléra, oblastí pamäte, ukazovateľa zásobníka a vektorov prerušení. Po tomto spúšťačom kóde začína chod programu C vždy funkciou *main()*, ktorá je vyvolaná *startup* modulom, preto je vloženie tejto funkcie povinné. Po spustení programu prevezme funkcia *main()* riadenie programu, spracováva postupne príkazy podľa zdrojového kódu a volá ďalšie funkcie.

Ak je funkciou vrátená hodnota, je pred menom deklarácie a definície uvedený typ návratovej hodnoty. *Deklarácia* – pred tým ako je funkcia vyvolaná, musia byť známe

jej vlastnosti. *Definícia* – vloženie, poprípade kód C funkcie. Za menom funkcie sú v okrúhlych zátvorkách uvedené formálne parametre použité pri volaní funkcie spolu s príslušným typom. Pokiaľ sa nevracia žiadna hodnota, alebo sa pri volaní nepoužije žiadny parameter, musí sa to označiť slovom void na príslušnej pozícii. Slovo void je možné preložiť ak neobsadený, vyjadruje teda, že neexistujú žiadne parametre. Každá definícia funkcie má funkčný blok vymedzený zloženými zátvorkami, ktorý obsahuje vlastné príkazy C (tzv. telo funkcie). Príkazy v C je veľmi vhodné popisovať príslušným komentárom.

1.5.2.2 Štruktúra programu

```
void main(void)
{
    if(...)
    {
        for(...)
        {
            ...           // jednoriadkový komentár
        }
    }
    else
    {
        ...           /* viacriadkový komentár */
    }
}
```

1.5.3 Zhrnutie z teórie programovania jednočipových mikropočítačov

Pre programátora je vytváranie programu v jazyku C jednoduchšie a prehľadnejšie. Tento luxus je ale za cenu toho, že takýto program bude o niečo pomalší a pamäťovo neoptimalizovaný, ako keby bol napísaný v JSA. Ide o to, že program napísaný v C musí byť taktiež preložený do JSA automaticky pomocou programu nazývaného kompilér. "Inteligencia" prevodu do JSA je tak daná vyspelosťou daného kompilera a programátor ju sám obvykle nemôže ovplyvniť.

2 Cieľ práce

Táto práca má pomôcť pri vývoji pohonného systému robotického zariadenia vyvíjaného na Katedre elektrotechniky, automatizácie a informatiky TF SPU v Nitre. Výsledkom práce bude návrh elektronických obvodov s použitím mikrokontroléra a základné programové vybavenie na moduláciu výkonu motora.

Medzi najvyužívanejšie typy motorov v mobilnej robotike patria jednosmerné motory, bezkomutátorové - BLDC motory a krokové motory. Každý z nich potrebuje pre svoju efektívnu činnosť účinný regulátor, ktorý riadi rýchlosť otáčania výstupného hriadeľa, smer jeho otáčania a chráni motor pred preťažením, a jeho následným zničením.

So zameraním sa na BLDC motory preto treba vytvoriť modul PWM riadenia ich výkonu, ktorého základ bude tvoriť jednočipový mikropočítač a iné elektronické obvody. Zariadenie má pritom umožňovať reguláciu od 0 do 100 % v jednopercentnom rozlíšení.

3 Metodika práce

Keďže ide o prototypové zariadenie, u ktorého sa predpokladá ďalší vývoj, samotnú prácu sme sa rozhodli riešiť pomocou vývojovej dosky od firmy Silicon Laboratories.

3.1 Charakteristika pracoviska

Návrhy boli riešené a odobrené spolu so školiteľom diplomovej práce na Katedre elektrotechniky, automatizácie a informatiky TF SPU v Nitre. Realizácia sa uskutočnila v domácich podmienkach. Overenie funkčnosti zariadenia a merania boli vykonané znovu na katedre.

3.2 Požiadavky kladené na vyrobený modul

3.2.1 Požiadavky na riadenie motora

Všeobecnou požiadavkou pri riadení motorov je plynulá zmena otáčok z nulových do maximálnych, alebo naopak. To zabezpečí riadiaci mikropočítač, ktorý bude vytvárať potrebné PWM signály. Keďže sú potrebné 3 PWM signály nie je vhodné uvažovať o ich softvérovom generovaní. Mikropočítač by mal vytvárať tieto signály na pinoch vstupno/výstupných brán určených pre výstup PWM. Treba brať do úvahy, že riadiaca úroveň jednočipových mikropočítačov je vo väčšine prípadov log. 0. Od toho sa bude odvíjať celé konečné zapojenie modulu. V neposlednom rade treba dbať aj na problematiku výkonov. Ani jeden z vyrábaných mikroradičov určite nie je schopný vybudieť motor priamo, ale pomocou tranzistorov alebo integrovaných obvodov na to určených. Taktiež ide o napájanie mikrokontrolérov, ktoré je väčšinou +5V alebo +3,3V a napájanie týchto motorov sa pohybuje od +9V z hľadiska efektivity výkonu. Preto treba už pri návrhu poznať aký druh mikropočítača a motora bude použitý, a podľa toho prispôbiť aj zapojenie.

3.2.2 Požiadavky na zobrazenie informácií o stave motora

Úlohou zobrazovača je poskytnúť užívateľovi základné informácie o stave motora, teda o jeho aktuálnych otáčkach, odoberanom prúde atď. Keďže tieto informácie sú potrebné najmä pri oživovaní zariadenia a jeho nastavovaní, bude nám postačovať dočasná komunikácia napr. pomocou sériovej linky alebo USB s osobným počítačom a jednoduchým terminálom. Aby však nebola komunikácia s užívateľom uskutočňovaná len pomocou osobného počítača je vhodné modul vybaviť aspoň LED diódami, ktoré by reprezentovali napr. štart a stop riadiacej sekvencie, poprípade preťaženie zariadenia. Tým sa poskytnú aspoň základné informácie o stave riadenia, aby nedošlo k poškodeniu modulu alebo motora.

3.3 Voľba typu riadiaceho mikrokontroléra a BLDC motora

3.3.1 Voľba typu riadiaceho mikrokontroléra a vývojovej DPS

Z popísaných požiadaviek sme zvolili pre stavbu mikropočítač od firmy Silicon Laboratories C8051F310 a vývojovú dosku označenú pod názvom Sensorless BLDC Motor Drive. Tento mikropočítač je 8 bitový procesor so 16 kB flash pamäťou, A/D prevodníkmi, čítačmi/časovačmi a inými prednosťami potrebnými pre stavbu zariadenia. Je vyrobený v 32-pinovom mikropúzdre LQFP-32, ktoré znižuje nároky na veľkosť plochy DPS.

3.3.2 Voľba typu BLDC motora

Kvôli cene nám pre overenie funkčnosti zariadenia úplne postačí malý bezkomutátorový DC motor získaný z nefunkčnej HDD mechaniky, i keď modul bude schopný dodať vyšší výkon.

Parametre motora: Typ: JVC FF 59-004037-000C

Napájanie: +12 V

3.4 Charakteristika technického a programového vybavenia

3.4.1 Použité prístroje

- zdroj napätia - batéria 12V /2Ah Typ:AB1202 - 0
- digitálny multimeter MY – 64
- USB Debug Adapter – firma Silicon laboratories.
- Logický analyzátor
- Laser Revolution Counter DT-10L, laserový otáčkomer

3.4.2 Použité elektronické súčiastky

Zoznam použitých súčiastok je uvedený v prílohe č. 6.

3.4.3 Programové vybavenie

- Microsoft Word – program pre spracovanie textov
- Eagle – program pre kreslenie elektrotechnických schém a návrh DPS
- Keil C51 μ Vision 3 – softvér pre tvorbu a zápis programu do mikropočítača
- Corel Draw – program pre kreslenie a úpravu obrázkov
- Driver pre USB Debug Adapter – firma Silicon laboratories
- Driver pre USB-UART prevodník s CP210x – firma Silicon laboratories
- Terminál pre sériovú komunikáciu

3.5 Vykonanie funkčných skúšok

Na zostavenú vývojovú dosku pripojíme BLDC motor, USB Debug Adapter a USB konektorom prepojíme vývojovú dosku s PC. Po dôkladnej kontrole pripojíme napájacie napätie 12V z batérie a mikrokontrolér naprogramujeme vytvoreným a doladeným programom pre použitý motor. Potom môžeme spustiť zariadenie a sledujeme správanie sa a chod motora pri zmenách otáčok, a či nedochádza k zahrievaniu súčastí systému. Na jednotlivých fázach môžeme pomocou logického analyzátoru zobrazíť priebeh napätí a skontrolovať tak správne generovanie PWM signálov. Pomocou laserového otáčkomera tiež môžeme zmerať otáčky rotora a porovnať ich s vysielanou informáciou o stave riadenia.

3.6 Návrhy na využitie výsledkov

Modul PWM riadenia BLDC motorov bude navrhovaný ako súčasť robotického zariadenia vyvíjaného na Katedre elektrotechniky, automatizácie a informatiky TF SPU v Nitre. Má slúžiť ako kontrolný modul na riadenie pohonu robota, ktorý je schopný komunikovať s nadriadenými systémami. Tiež môže slúžiť ako výučbová pomôcka pre študentov.

4 Výsledky práce

4.1 Opis hardvéru

Schéma zapojenia je zobrazená na obr. 15. Obvod pozostáva z mikrokontroléra C8051F310, trojfázového výkonového MOSFET mostíka, troch dvojitých radičov pre MOSFET tranzistory, napät'ových deličov pre kontrolu napätia, prúdového zosilňovača, potenciometra pre nastavenie rýchlosti, dvoch funkčných spínačov, USB-UART prevodníka a napät'ového regulátora..

Piny portu P1 sú nakonfigurované ako push-pull výstupy a sú pripojené k trom dvojitým radičom MOSFET tranzistorov označeným ako U3, U4, U5. Tieto piny striedavo spínajú pomocou radičov výkonové tranzistory označené ako Q1A-Q3A a Q1B-Q3B v zmysle komutačnej sekvencie, a tým privádzajú pracovné napätie na jednotlivé cievky motora.

Ku každému z troch výstupov fáz je pripojený jednoduchý rezistorový delič tvorený rezistormi R11 a R21, R12 a R22, R13 a R23. Tento delič rozdeľuje napätie privádzané na fázy šiestimi. Tento pomer dovoľuje presné ADC meranie do 19,8V. Tolerancie vstupných napätí na pinoch tohto portu ponúkajú ochranu zariadenia pred napätiami na fázach až 30V. Nominálne napájanie motora a teda aj zariadenia je 12V.

Kondenzátor pripojený k nižšiemu rezistoru každého napät'ového deliča fázy tak vytvára spolu s ním jednoduchý dolnopriepustný RC filter. Kondenzátory sú označené ako C18 - C20. Každý filter je nastavený na frekvenciu o dekádu vyššiu než je frekvencia PWM. Testovacie body sú na schéme zobrazené ako VA, VB, VC. Štvrtý napät'ový delič je použitý na zisťovanie napájacieho napätia motora. Rezistory sú v pomer 1:12 a sú označené ako R20 a R15. K rezistoru R20 je tiež pripojený paralelne kondenzátor C17. Testovací bod je na schéme zobrazený ako VM. Pri napájanom napätí 12V bude teda na fázových deličoch napätia na príslušnom ADC vstupe napätie 2V zatiaľ čo v bode označenom ako VM bude napätie 1V. Výsledkom ADC prevodu bude 16 – bitové číslo so znamienkom zodpovedajúce strednej hodnote.

Zapojenie ďalej obsahuje snímací rezistor prúdu R25, na ktorý je pripojený prúdový zosilňovač U7A. Rezistor má hodnotu $20\text{m}\Omega$ a je v púzde 1812. Tento rezistor slúži na ochranu motora a výkonových spínacích prvkov pred preťažením a následným zničením. Jeho maximálna výkonová strata je 500mW a maximálny prúd, ktorý ním môže tiecť je 5A . Strata 100mV na rezistore si vyžaduje také zosilnenie, aby sa dosiahlo viac ako 5 bitové rozlíšenie z 10 bitového ADC.

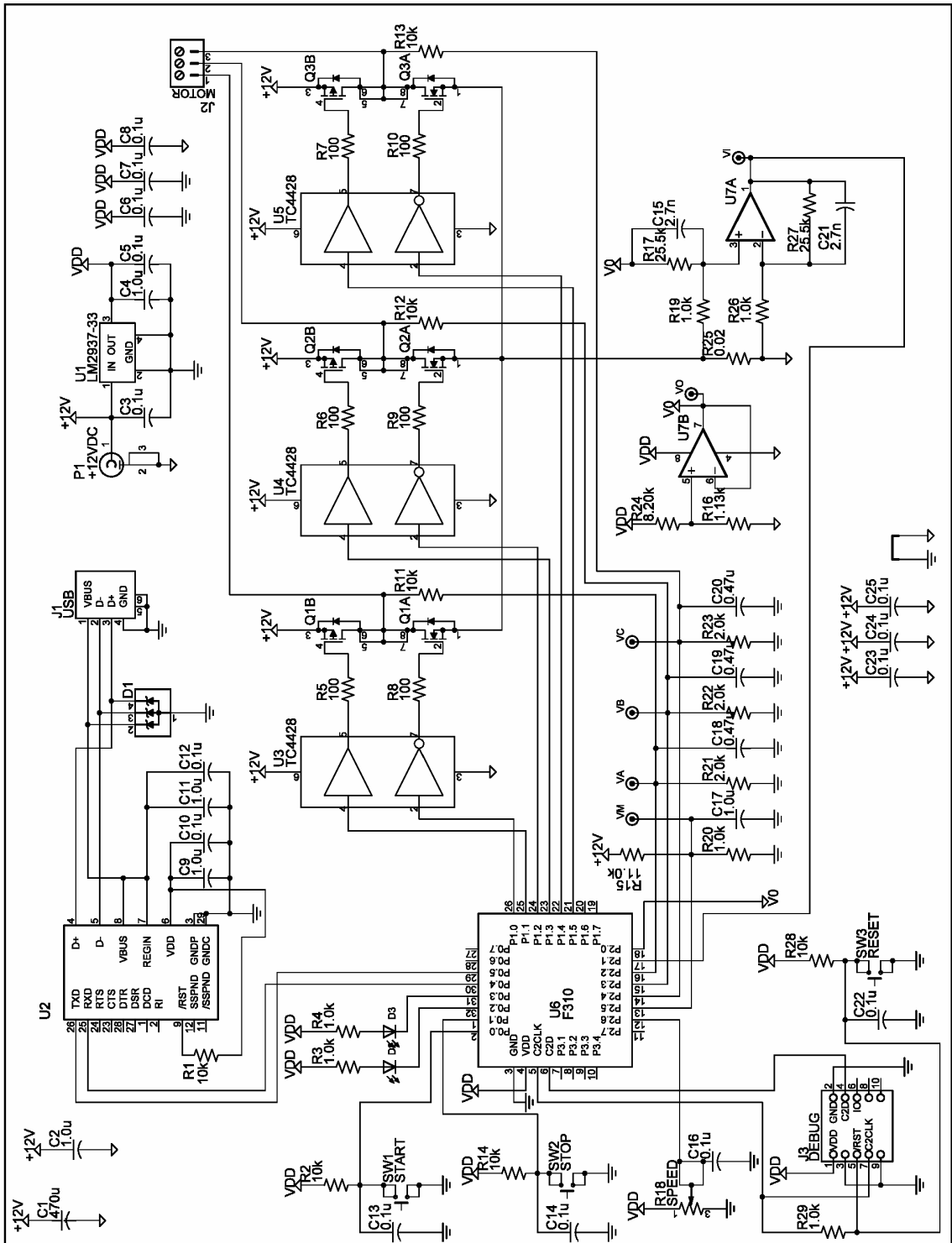
Prúdový zosilňovač je zapojený ako klasický rozdielový zosilňovač. V tomto zapojení je rezistor R17 pripojený na 400mV napätovú referenciu namiesto toho, aby bol pripojený na napätovú zem. Táto referencia kompenzuje vstupnú napätovú nesymetriu operačného zosilňovača a taktiež udržuje výstup v odporúčanom napätovom rozsahu.

Pre túto funkciu bol vybraný operačný zosilňovač LMV358. Je to nízkonapätová verzia štandardne vyrábaného LM358, dvojitého CMOS operačného zosilňovača.

Druhý operačný zosilňovač U7B v tomto púzde je použitý na vytvorenie spomínanej napätovej referencie 400mV .

Ako USB – UART prevodník je zvolený obvod CP2101 označený ako U2. Tento obvod uskutočňuje USB komunikáciu použitím virtuálneho COM portu na používateľovom PC. Mikrokontrolér komunikuje s CP2101 použitím konvenčnej sériovej linky RS232.

Ako stabilizátor napätia pre napájanie mikrokontroléra v príslušnom rozsahu $2,7\text{V}$ až $3,6\text{V}$ sme zvolili obvod LM2937-33. Jeho vstup je pripojený priamo na nominálne napájanie zariadenia 12V a na výstupe generuje potrebné stabilizované napájanie $3,3\text{V}$ pre mikrokontrolér a obvod U7.



Obr. 15 Schéma zapojenia

4.1.1 Opis mikropočítača C8051F310

Ako základ modulu je zvolený mikropočítač C8051F310, ktorý svojimi možnosťami dokonale postačí na jeho realizáciu. Prvým krokom pre realizáciu akéhokoľvek zapojenia s jednočipovým mikropočítačom je nastudovanie si informácií z jeho katalógového listu (datasheet). Blokový diagram C8051F310 uvádzam v prílohe č.1 a konfiguráciu pinov puzdra v prílohe č.2.

4.1.1.1 Parametre C8051F310

- Vysokorýchlostné jadro kompatibilné s 8051 schopné pracovať s rýchlosťami do 25MIPS
- Zabudovaný debug prevodník na čipe
- 10 – bitový, 200ksps, 25 kanálový ADC s analógovým multiplexorom
- 16 kB flash pamäte na čipe
- 1280 bytov RAM pamäte na čipe
- SMBus/I2C, UART, SPI sériové prevodníky implementované do hardvéru
- Štyri 16 – bitové časovače
- Programovateľné C/T pole s piatimi Capture/Compare modulmi a Watchdog časovačom
- Power – on reset na čipe, VDD monitor a teplotný senzor
- Dva napäťové comparátory na čipe

S Power-on resetom, VDD monitorm, Watchdog časovačom a hodinovým oscilátorom je C8051F310 skutočne samostatným systémom. Flash pamäť môže byť dokonca preprogramovaná priamo v obvode bez straty dát. Používateľov softvér má celkovú kontrolu nad perifériami a môže individuálne vypínať tieto periférie pre prípadnú potrebu úspory energie. Mikrokontrolér je určený pre napájanie od 2,7V do 3,6V a pre teplotný rozsah od -45°C do +80°C. Porty I/O a RST pin majú toleranciu vstupných signálov do 5V.

4.1.2 Opis operačného zosilňovača LMV358

Tento operačný zosilňovač patrí medzi nízkonapäťové (2,7V - 5,5V) verzie dvojitých operačných zosilňovačov. Je vhodné ho použiť v aplikáciách pre operácie s nízkymi napätiami a vďaka jeho mikropuzdru je možné ušetriť miesto na doske plošných spojov. Spoľahlivo pracuje v šírke pásma do 1MHz a rýchlosť nábehu napätia je 1V/ μ s s nízkou spotrebou napájacieho prúdu. Teplotný rozsah použitia je -40°C až +85°C. Konfigurácia pinov puzdra je uvedená v prílohe č.4.

4.1.3 Opis USB – UART prevodníka CP2101

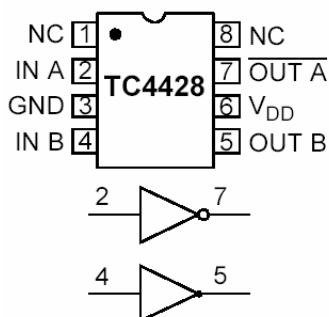
Tento obvod je plne integrovaný USB – UART prevodník, poskytujúci jednoduché rozšírenie modulov komunikujúcich pomocou sériovej linky RS232, na zariadenie prevádzkujúce USB komunikáciu s minimom externých súčiastok. Obsahuje USB 2.0 plne rýchlostný funkčný kontrolér, USB vysielateľ, oscilátor, pamäť EEPROM a asynchrónny UART. EEPROM môže byť použitá na uloženie dát ako sú sériové číslo produktu, popis produktu, ID produktu a iné. Komunikácia sa uskutočňuje pomocou virtuálneho COM portu a terminálu pre komunikáciu pomocou linky RS232. Rozsah napájacieho napätia je od 3,0V – 3,6V a zariadenie je možné použiť v rozsahu teplôt -40°C až +85°C. Konfiguráciu pinov puzdra uvádzam v prílohe č.3.

4.1.4 Opis stabilizátora LM2937-33

Ide o pozitívny napäťový stabilizátor s možnosťou dodania prúdu do záťaže o veľkosti 500mA. Pokojový odber prúdu nepresahuje 10mA. Obsahuje ochranu proti skratu na výstupe, tepelnú poistku a ochranu proti prepólovaniu batérie.. Umožňuje pripojenie filtračného kondenzátora s nízkym ESR (0,01 Ω - 5 Ω). Tolerancia stabilizovaného napätia je v rozsahu 5%. Puzdro je zobrazené v prílohe č.5.

4.1.5 Opis radiča TC4428 pre MOSFET tranzistory

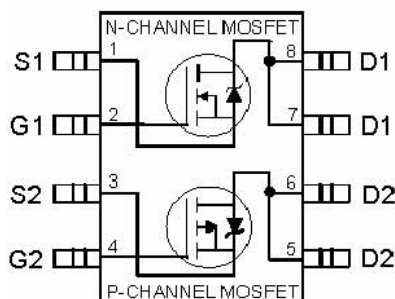
Ide o rýchly radič, ktorý dokáže nabíjať/vybíjať kapacitu až 1000pF G – elektród MOSFET tranzistorov do 30 ns. Napájacie napätie je do 4,5V – 18V. Je odolný proti elektrostatickým výbojom do 4kV. Výstupná impedancia je 7 Ω . Vstupy sú vysokoimpedančné, vyrobené technológiou TTL/CMOS. Puzdro je zobrazené na obr.16.



Obr. 16 Puzdro radiča TC4428

4.1.6 Opis MOSFET tranzistorových dvojíc F7309

MOSFET tranzistorová dvojica je integrovaná v puzdre SO8 a je zložená z N – kanálového a P – kanálového MOSFET tranzistora. Napájanie je $\pm 20V$ a odpor v zopnutom stave sa pohybuje od 0,05-0,1 Ω . Výkonová strata na tomto prvku môže byť 1,4 W. Pracovný teplotný rozsah je od -55°C až 150°C. Každý tranzistor má paralelne pripojenú ochrannú diódu. Puzdro je zobrazené na obr. 17.



Obr. 17 Puzdro tranzistorovej dvojice F7309

4.2 Opis softvéru

Kód v jazyku C pre bezsenzorové riadenie BLDC motora je rozdelený do štyroch súborov:

- *slbdc.h*
- *slbdc.c*
- *T0_ISR.c*
- *T2_ISR.c*

Súbor označený ako *slbdc.h* obsahuje všetky preprocesorové makrá, parametre motora a typové definície. Hardvér môže byť použitý s akýmkoľvek 12V BLDC motorom, pod podmienkou, že sa parametre daného motora prepíšu v tomto kóde na príslušné hodnoty. Používateľ špecifikuje tieto hodnoty v makrách, ktorými sú MAXRPM – maximálne otáčky, POLES – počet pólov daného motora, VMOTOR – napájacie napätie motora, IMOTOR – prúd tečúci motorom pri plnej záťaži a MILIOHMS – elektrický odpor cievky jednej fázy v miliohmoch.. Všetky ostatné potrebné hodnoty pre riadenie motora sú vypočítavané z týchto hodnôt. Používateľ teda potrebuje len prepísať tieto hodnoty, a znovu skompilovať a zapísať softvér do mikrokontroléra.

Súbor *slbdc.c* obsahuje *main()* funkciu a všetky funkcie, ktoré sú ňou volané. Táto funkcia implementuje inicializáciu a komunikáciu s užívateľom. Všetky časovo náročné úlohy sú ovládané pomocou podprogramov prerušenia označených ako ISR (Interrupt Service Routines).

Premenné v týchto podprogramoch, ktoré sú inicializované alebo sprístupnené pre hlavný program *main()* sú deklarované v ISR súboroch a tiež sú naopak *externe* deklarované v súbore *slbdc.c* .

USB prevodník môže byť použitý samostatne s prídavným modulom pre zobrazenie informácií, alebo môže byť pripojený na osobný počítač pomocou USB konektora a jednoduchého hyperterminálu.

Dve spínacie tlačidlá, označené ako START a STOP, spúšťajú a zastavujú chod programu a teda aj samotného motora. Stlačením tlačidla START sa hlavný program odvoláva na funkciu *StartMotor()*. Samotný štart a chod motora teda preberá táto funkcia. Stlačením tlačidla STOP sa podobne zavolá funkcia na zastavenie motora až kým nie je znovu reštartovaný.

Pomocou osobného počítača a nainštalovaného hyperterminálu sa zobrazujú informácie o aktuálnom stave riadenia a tak možno upravovať hodnoty PI konštánt. Po prvom stlačení tlačidla START mikrokontrolér vyšle informáciu cez USB do používateľovho PC a v hyperterminálovom okne sa vypíšu nasledovné informácie podľa zmeny stavu riadenia:

Aligning... (vybudenie motora a zistenie natočenia rotora)
Starting... (štart riadiacej sekvencie)
Running... (chod motora)

Keď je motor v prevádzke, používateľ môže zobrazit' v hyperterminálovom okne stav riadenia motora jednoduchým zapísaním znaku „s“ (status) z klávesnice a jeho vyslaním na komunikačný port. Mikrokontrolér spracuje požiadavku a vyšle dané informácie. Hodnoty riadiacich PI konštánt môžu byť tiež zmenené jednoduchými znakovými príkazmi. Zapísaním a vyslaním znaku „p“ možno zmeniť proporcionálnu konštantu KP v rozmedzí desiatkových čísel. Podobne tiež možno modifikovať integračnú konštantu KI zapísaním znaku „i“. Možnosť zmeny PI konštánt je veľmi účelná pri oživovaní zariadenia s pripojeným motorom s konkrétnymi parametrami.

Všetky riadiace a časovacie cykly sú generované pomocou časovača T0. Tento je nakonfigurovaný ako 16-bitový časovací modul. Pri pretečení časovača je ním vyvolané prerušenie, ktoré vyvolá vykonanie funkcie *T0_ISR()*. Prerušenie od časovača T0 má nastavenú najvyššiu prioritu, takže vykonanie tejto funkcie má prednosť pred inými funkciami, ktoré vyvolávajú prerušenia s nižšou prioritou. Funkcia *T0_ISR()* vyvoláva mnoho iných potrebných funkcií, ktoré sú uložené v súbore *T0_ISR.c*.

Funkcia *StartMotor()* je vyvolaná hlavným programom *main()* na inicializáciu T0 a plánuje prvé prerušenie. Táto funkcia sa tiež nachádza v súbore T0_ISR.c a je externe deklarovaná z hlavného programu *main()*.

Základom funkcie *T0_ISR()* je stavová premenná označená ako *Status* a v závislosti na hodnote uloženej v tejto premennej, funkcia *T0_ISR()* vyvoláva príslušné funkcie, ktorými sú *stop()*, *align()*, *start()* a *run()*. Poslednú úlohu, ktorú táto funkcia vykonáva pred návratom do hlavného programu je aktualizácia registrov časovača T0 pre začiatok ďalšej periódy *NextT*. To je spoločné pre všetky stavy a preto je to obsiahnuté v tejto funkcii.

Funkcia *stop()* je veľmi jednoduchá. Pozastavuje činnosť PWM signálov, preruší od časovača T0 a zastavuje časovač. Motor sa zastaví bezpečne, pričom sú všetky tranzistory uzatvorené. Vyvolanie tejto funkcie je možné stlačením tlačidla STOP alebo ju vyvoláva funkcia *Status* pri detekovaní nadprúdu.

Funkcia *align()* vyvoláva periodické prerušenia každých 10ms. Používa dve statické počítadlá označené ako „v“ (voltage ramp) a „d“ (delay). Vždy keď je vyvolaná táto funkcia inkrementuje sa obsah počítadla „v“ až kým nie je dosiahnuté maximum. Pri dosiahnutí maxima následné vyvolanie funkcie inkrementuje obsah počítadla „d“. Pri dosiahnutí predpísaného oneskorenia daného hodnotou počítadla „d“, je premenná vo funkcii *Status* inkrementovaná a premenné vo funkcii *start()* sú inicializované. Znamená to, že natočenie rotora je v správnej polohe pre spustenie riadiacej sekvencie. Následne už *T0_ISR()* vyvoláva funkciu *start()* namiesto volania funkcie *align()*.

Funkcia *start()* využíva riadenie rozdelené do 8 časových modulov. Ku komutácii dochádza len v prvom časovom module. Tento systém dovoľuje dlhšie trvanie periódy počas štartovacej periódy bez zmeny časovacích hodín.

Použitie sú časové programové moduly 0 a 7 a sú vyjadrené funkciami. Funkcia *start0()* je vyvolaná v časovom module 0 a vypočítava nasledujúcu periódu *NextT*. Táto funkcia je realizovaná 192 krát, vždy pri vstupe do akceleračnej tabuľky. Hodnota

v tejto tabuľke je použitá pre výpočet nasledujúcej periódy a výstupného napätia. Pritom sa tiež inkrementuje hodnota uložená v *AccIndex*.

Funkcia *start7()* je vyvolaná v poslednom časovom module funkcie *start()*. Táto funkcia testuje hodnotu uloženú v *AccIndex* na určenie, či sa štartovacia fáza kompletne uskutočnila. Ak táto funkcia vyhodnotí, že štart motora prebehol bez problémov, hodnota v premennej *Status* sa nastaví na *Run* – chod motora.

Funkcia *run()* využíva riadenie rozdelené do 4 časových modulov. Motor je pritom komutovaný len v časovom module 0. Každý z týchto časových modulov má funkciu s prislúchajúcim názvom *run0()*, *run1()*, *run2()*, *run3()*. Riadenie v týchto štyroch časových modulloch ako v celku uľahčuje plánovanie úloh v rozličných časových programových modulloch.

V každom časovom slote funkcie *run()* je prečítaná hodnota v ADC, uložená v príslušnom registri a ADC je nakonfigurovaný na nasledujúci časový slot.

BEMF je vzorkovaná v časových slotoch 1, 2 a 3. ADC je nakonfigurovaný na spustenie konverzie pri pretečení časovača T0. Toto zaistí, že prvá vzorka sa objaví v presnom čase pri rešpektovaní komutačnej sekvencie. ADC je znovu nakonfigurovaný pre ďalšiu vzorku na konci každého časového slotu, takže vzorky BEMF sú spracovávané na konci časových slotov 0,1 a 2.

Na zlepšenie rozlíšenia a redukciu šumu vytváraného PWM signálmi je BEMF vzorkovaná ôsmimi vzorkami. Iniciačná vzorka je synchronizovaná s časovačom T0 a zvyšných 7 vzoriek sa vzorkuje maximálnou rýchlosťou ADC.

Prúd prechádzajúci motorom a informácia z potenciometra sa vzorkujú počas časového slotu 0, pokiaľ v tomto slote nie je vzorkovaná BEMF. Výpočty sú rozdelené do troch časových slotov, aby sa vyrovnala dĺžka trvania každej funkcie. Priemerná hodnota a hodnota chybového napätia sú vypočítavané časovom slote 2. Výpočty súvisiace s detekciou stavu sú uskutočňované v časovom slote 3.

Časovač T2 je nakonfigurovaný ako 16-bitový s funkciou autoreload, generujúci periodické prerušenia každú 1 ms. Funkcia *T2_ISR()* je určená pre prácu s PI konštantami a na riadenie zrýchlenia rotora.

4.2.1 Výpis hlavných častí programu

```
void main (void)
{
    char theChar;

    PCA0MD &= ~0x40;           // nepovolený watchdog časovač

    SYSCLK_Init ();           // inicializačné funkcie
    PORT_Init();
    PCA0_Init();
    T0_Init ();
    T2_Init ();
    UART0_Init ();
    ADC_Init();
    EA = 1;                   // povolenie prerušení
    printf("\r\nreset...\r\n");
    while(1)
    {
        printf("push start button...\r\n");
        while(Start==OFF);    // čaká na štart
        while(Start==ON);     // čaká na spustenie
        StartMotor();         // štart motora
        printf("aligning...\r\n");
        while(Status!=START); // zisťuje stav riadenia
        printf("starting...\r\n");
        while(Status!=RUN);   // zisťuje stav riadenia
    }
}
```

```

printf("running...\r\n");
while(Status==RUN)
{
    if (RI0) // overuje sa vstup
    {
        theChar = getkey();
        while (theChar<'a' || theChar>'z') // ignoruje znaky mimo rozsahu
            theChar = getkey();
        switch(theChar) // rozbor znaku
        {
            case 'p': // nastavenie Kp
                printf("Kp?\r\n");
                scanf("%d",&Kp);
                printf("\r\nKp=%d\r\n",Kp);
                break;
            case 'i': // nastavenie Ki
                printf("Ki?\r\n");
                scanf("%d",&Ki);
                printf("\r\nKi=%d\r\n",Ki);
                break;
            case 's': // zobrazenie stavu riadenia
                printf("Speed=%u\r\n",SpeedRPM);
                printf("Vout=%u\r\n",Vout);
                printf("Verror=%d\r\n",Verror);
                printf("Imotor=%u\r\n",Imotor);
                printf("Vemf=%d\r\n",Vemf);
                printf("Vpi=%d\r\n",Vpi);
                printf("Vpot=%u\r\n",Vpot);
                printf("Kp=%d\r\n",Kp);

```

```

        printf("Ki=%d\r\n",Ki);
        printf("\r\n");
        break;
    default:
        printf("Error\r\n");
    }
    // koniec rozboru
}
// koniec cyklu if
if(Stall)
    //overuje príznakový bit rozbehu motora
    {
        printf("Stall!\r\n");
        Stoji = 1;
        // motor sa zastavil vplyvom preťaženia

    }

if(Stop==ON)
    // overuje stlačenie Stop tlačidla
    {
        Status = STOP;
        printf("Stopping Motor!\r\n");
        while(Stop==ON);
    }
}

if(OverCurrent)
    // overuje príznakový bit pri meraní nadprúdu
    {
        printf("OverCurrent!\r\n");
    }

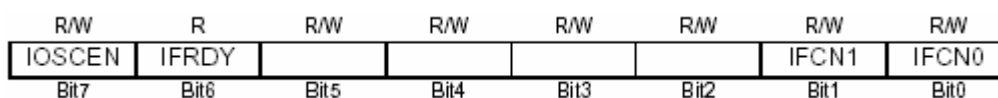
}
}

```

4.2.2 Opis inicializačných funkcií

4.2.2.1 SYSCLK_Init ()

Mikrokontrolér C8051F310 obsahuje na svojom čipe programovateľný vnútorný oscilátor a riadiaci obvod potrebný pre externý oscilátor. Vnútorný oscilátor môže byť povolený/nepovolený a kalibrovaný pomocou registrov OSCICN a OSCICL. Register OSCICL určuje periódu vnútorného oscilátora, ktorá je vo výrobe nakalibrovaná na frekvenciu 24,5 MHz. Registrom OSCICN je možné túto frekvenciu deliť 1, 2, 4 a 8. V tomto zariadení je register OSCICN nakonfigurovaný podľa obr. 18 príkazom OSCICN = 0x83, teda zariadenie pracuje na frekvencii 24,5 MHz.



Obr.č. 18 Konfigurácia registra OSCICN

Bit7: IOSCEN: Povoľujúci bit vnútorného oscilátora

0: Vnútorný oscilátor nie je povolený

1: Vnútorný oscilátor je povolený

Bit6: IFRDY: Kontrolný bit

0: Vnútorný oscilátor nepracuje na naprogramovanej frekvencii

1: Vnútorný oscilátor pracuje na naprogramovanej frekvencii

Bity5–2: nepoužité

Bity1–0: IFCN1-0: Nastavovacie bity

00: nakalibrovaná frekvencia 24,5 MHz je delená 8

01: nakalibrovaná frekvencia 24,5 MHz je delená 4

10: nakalibrovaná frekvencia 24,5 MHz je delená 2

11: nakalibrovaná frekvencia 24,5 MHz je delená 1

4.2.2.2 PORT_Init()

Inicializácia I/O portov pozostáva z nasledujúcich krokov:

- výber vstupného módu pre všetky piny daného portu (analógový alebo digitálny) použitím registra PnMDIN.
- výber výstupného módu pre všetky piny daného portu (otvorený kolektor alebo zapojenie push-pull) použitím registra PnMDOUT.
- vybrať piny ktoré bude vynechané I/O Crossbar dekodérom použitím registra PnSKIP
- priradiť piny portu na periférie
- povoliť Crossbar dekodér, register XBRn

Podľa postupu sa nastaví potrebné registre nasledujúcimi príkazmi:

```
XBR0 = 0x01;           // povolený UART na Crossbar
XBR1 = 0x02;           // povolené CEX0,CEX1 na Crossbar
P0MDOUT = 0x1C;       // P0.2, P0.3 a P0.4 sú výstupy
P1MDOUT = 077;        // povolené výstupy motora
P2MDIN = 0x00;        // všetky piny portu P2 sú analógové vstupy
P0SKIP = ~0x30;       // vynechané všetky bity, okrem UART
P1SKIP = 071;         // inicializačný PSKIP vzor
P2SKIP = 0x0F;        // vynechané všetky bity portu P2
XBR1 |= 0x40;         // povolený Crossbar dekodér
P1 = 0xff;            // P1 nastavený na log.1
```

Všetky piny príslušného portu musia byť nakonfigurované buď ako analógové alebo digitálne. Piny, ktoré budú použité ako komparátor alebo ADC vstupy majú byť nastavené ako analógové vstupy. Navyše by tieto analógové vstupy mali byť nastavené tak, aby boli Crossbar dekodérom ignorované, a to nastavením registrov PnSKIP. Vstupný mód je nastavený v registroch PnMDIN, kde „1“ indikuje digitálny vstup a „0“ analógový vstup. Výstupný mód je nastavovaný v registroch PnMDOUT. Každý výstupný radič môže byť nakonfigurovaný ako otvorený kolektor alebo ako push-pull.

Registre XBR0 a XBR1 musia byť naplnené vhodnými hodnotami vyžadované návrhom. Nastavením XBARE bitu v registry XBR1 na „log.1“ povolujeme funkciu Crossbar dekodéra. Pokiaľ je povolená jeho funkcia piny zostávajú v štandardnom I/O móde aj napriek nastaveniam XBRn registrov.

4.2.2.3 PCA0_Init()

Programovateľné pole PCA0 poskytuje lepšie využitie časových modulov, pričom nezaťažuje CPU ako štandardné C/T moduly. Pozostáva zo 16-bitového C/T a päť 16-bitových C/C (capture/compare) modulov. Každý z týchto modulov má vlastnú I/O linku (CEXn), ktorá je riadená I/O Crossbar dekodérom. C/T je riadený programovateľnou časovou základňou, ktorá môže byť vybraná zo šiestich zdrojov hodinového signálu, ktorými sú systémové hodiny, systémové hodiny delené 4 alebo 12, externý oscilátor delený 8, pretečenie T0 a externý hodinový signál na vstupnom pine ECI. C/C moduly môžu pracovať nezávisle ako softvérový časovač, frekvenčný výstup, 8-bitový PWM výstup, 16-bitový PWM výstup. Registre sú nakonfigurované nasledovne:

```
PCA0MD = 0x02;           // PCA používa sysclk/4
PCA0CPL0 = 0x00;        //zápis nuluje ECOMn bit
PCA0CPL1 = 0x00;        //zápis nuluje ECOMn bit
PCA0L = 0x00;           // reset časovača
PCA0H = 0x00;           // reset časovača
PCA0CPH0 = 0x00;        // inicializácia na 0%
PCA0CPH1 = 0x00;        // inicializácia na 0%
CR = 1;                 // štart PCA0 časovača
```

4.2.2.4 T0_Init () a T2_Init ()

Mikrokontrolér obsahuje štyri počítadlá/časovače, ktoré môžu byť nakonfigurované v rôznych módoch, ktoré zobrazuje tab. 3.

Tab. 3 Módy časovačov

Módy časovača T0 a T1	Módy časovača T2	Módy časovača T3
13-bitové počítadlo/časovač	16-bitový časovač s funkciou autoreload	16-bitový časovač s funkciou autoreload
16-bitové počítadlo/časovač		
8-bitové počítadlo/časovač s funkciou autoreload	Dva 8-bitové časovače s funkciou autoreload	Dva 8-bitové časovače s funkciou autoreload
Dve 8-bitové počítadlá/časovače (len T0)		

Časovač T0 je implementovaný ako 16-bitový register s prístupom po dvoch oddelených bytoch označených ako nižší byte (TL0) a vyšší byte (TH0). Riadiaci register TCON je použitý na povolenie tohto časovača a tiež na indikáciu stavu. Povolenie prerušenia od časovača sa nastavuje v registri IE nastavením bitu ET0. Nastavenie módu časovača sa uskutočňuje v registri TMOD. Nastavenie tohto časovača sa prevedie nasledovnými príkazmi:

```

TMOD &= ~0x03;           // vynulovanie módu časovača T0
TMOD |= 0x01;            // T0 mód 1
CKCON &= ~0x07;         // vynulovanie bitov SCAx a T0M
CKCON |= 0x02;          // T0 používa SYSCLK/48
IP |= 0x02;             // T0 vysoká priorita

```

Časovač T2 je 16-bitový register tvorený dvomi 8-bitovými špeciálnymi funkčnými registrami TMR2L (nižší byte) a TMR2H (vyšší byte). Tento časovač môže pracovať ako 16-bitový s funkciou autoreload alebo ako rozdelený na dva 8-bitové časovače tiež s funkciou autoreload. Tretí bit v registri TMR2CN určuje tento mód. Na nastavenie sa použijú príkazy:

```
CKCON |= 0x10;           // T2 používa SYSCLK
TMR2RLL = T2_RELOAD_L;  // Časovač T2 nižší byte
TMR2RLH = T2_RELOAD_H;  // Časovač T2 vyšší byte
```

4.2.2.5 UART0_Init ();

UART je asynchrónny, plne duplexný sériový port umožňujúci sériový prenos. Obsahuje dva špeciálne funkčné registre SCON0 a SBUF0. Po vyslaní informácie je vyvolané prerušenie od UART a bit TI0 v registri je nastavený na „log. 1“, tak ako je pri prijatí informácie nastavený bit RI0 na „log.1“ v tomto registri. Nulovanie týchto bitov sa prevádza softvérovo. Nastavenie UART sa prevedie nasledujúcimi príkazmi:

```
SCON0 = 0x10;           // povolenie prijímania
TMOD &= ~0x30;         // vynulovanie módu časovača T1
TMOD |= 0x20;          // časovač T1 mód 2
CKCON |= 0x08;         // T1 používa SYSCLK
TH1 = 0x96;            // pevná prenosová rýchlosť 115200 Baud
TL1 = TH1;             // inicializácia časovača T1
TR1 = 1;               // štart časovača T1
TI0 = 1;               // indikácia vyslania dát TX0
```

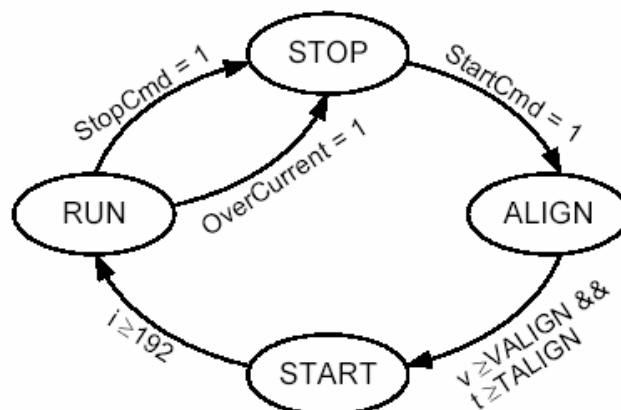

4.2.2.6 ADC_Init()

Subsystem ADC0 pozostáva z dvoch analógových multiplexorov (spoločne označených ako AMUX0) s 25 vstupnými selekciami a 200 kps, 10-bitovým postupným aproximačným registrom (SAR) ADC. Módy konverzie dát a AMUX0 je možné nakonfigurovať softvérovo pomocou SFR. ADC0 subsystem je povolený len keď AD0EN bit v kontrolnom registri ADC0CN je nastavený na „log.1“. Ak je nastavený na „log. 0“ subsystem pracuje v nízkopríkonovom režime. Registre sa inicializujú nasledovnými príkazmi:

```
AMX0P = 0x09;           // konfigurácia pre prúd tečúci motorom
AMX0N = 0xFF;           // single ended mód
ADC0CF = 0x40;          // nastavenie hodín SARCLK 272222,
ADC0CN = 0x80;          // inicializácia na AD0BUSY
REF0CN = 0x08;          // použitie Vdd ako referencie pre ADC
                        // oscilátor
```

4.2.3 Vývojový diagram softvéru

Popísaný softvér je názornejšie vidieť na obr. 19.

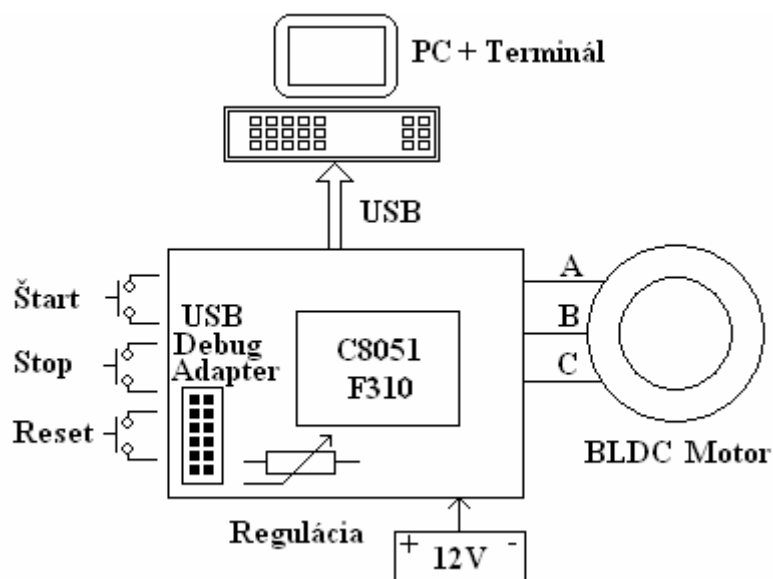


Obr. 19 Vývojový diagram softvéru

4.3 Dosiiahnuté výsledky

Požiadavky kladené na vyrobený modul sa nám podarilo dosiahnuť, až na požiadavku regulácie rýchlosti otáčania výstupného hriadeľa priamo od 0% . Pri experimentovaní a doladovaní programu sa nám nepodarilo dosiahnuť nízke hodnoty výkonu, pretože pri veľmi nízkych otáčkach vykazoval motor pulzný priebeh výstupného krútiaceho momentu. To bolo spôsobené tým, že tieto druhy motorov nie sú stavané na prevádzku pri takýchto nízkych otáčkach. Na aplikácie, pri ktorých sú požadované veľmi nízke otáčky, sú vhodnejšie krokové motory. Táto nevýhoda však nemala negatívny dôsledok na implementáciu do robotického zariadenia, pretože jeho pohon je riešený pomocou prevodového mechanizmu a teda aj pri určitých minimálnych otáčkach motora, pri ktorých sa už neprejavuje pulzujúci účinok, je robotické zariadenie schopné pohybovať sa tak, aby bolo dobre ovládateľné aj pri pomalom pohybe.

Zariadenie je zhotovené na vývojovej doske s rozmermi 81 x 63 mm. Obsahuje konektor pre pripojenie BLDC motora, napájania, USB Debug adaptéra a USB konektor pre komunikáciu s PC. Taktiež obsahuje regulačný potenciometer, dve signalizačné LED diódy a ovládacie tlačidlá. Modul spolu s pripojeným motorom je napájaný z 12V batérie. Komunikácia s užívateľom sa uskutočňuje pomocou USB pripojenej k PC , na ktorom je nainštalovaný terminál. Čiastočné informácie tiež ponúkajú signalizačné LED diódy. Bloková schéma celého zariadenia je zobrazená na obr. 20.



Obr. 20 Bloková schéma zostaveného zariadenia

PC a terminál slúžia na komunikáciu s užívateľom. Po zapojení zariadenia podľa blokovej schémy sa na obrazovke PC zobrazí nápis „ push start button ...“ a potom môže používateľ stlačiť Štart tlačidlo. Následne sa spustí štartovacia sekvencia motora a na obrazovke sa postupne vypisuje stav riadenia. Nápis „ aligning ...“ znamená, že poloha rotora sa nastavuje do potrebnej polohy na spustenie PWM komutácie. Potom nasleduje nápis „ starting ...“, kedy sa spúšťa PWM komutácia a motor sa rozbieha. Podľa požiadavky regulačného potenciometra dosiahne motor potrebné otáčky a na obrazovke sa vypíše nápis „ running ...“, kedy sú dosiahnuté požadované otáčky. Správanie motora je možné ovplyvniť nastavením konštánt K_p a K_i a to vyslaním príslušného znaku „p“ alebo „i“ a zadaním ich číselnej hodnoty. V prípade vyslania znaku „s“ je možné zistiť celkový stav riadenia, kedy sú na obrazovku vypísané nasledovné údaje: rýchlosť (SpeedRPM), prúd prechádzajúci motorom (I_{motor}), napätie BEMF (V_{emf}), napätie na potenciometri (V_{pot}), konštanty K_p a K_i , a iné.

V prípade rýchleho nadprúdu je motor automaticky zastavený, aby sa nezničil a na obrazovke sa vypíše nápis „ OverCurrent!“. Používateľ môže zastaviť motor stlačením tlačidla Stop, kedy sa na obrazovke vypíše nápis „Stopping Motor!“. Stlačením tlačidla Reset je na obrazovke vypísané „nreset...“ a celý systém sa reštartuje. V tab. 4 sú zobrazené niektoré informácie o stave riadenia, pri rôznych otáčkach rotora.

Tab. 4 Niektoré zobrazované informácie o stave riadenia pri rôznych otáčkach

Otáčky zmerané s DT10-L	540 min ⁻¹	1555 min ⁻¹	2110 min ⁻¹	2560 min ⁻¹
SpeedRPM	527 min ⁻¹	1520 min ⁻¹	2050 min ⁻¹	2510 min ⁻¹
I_{motor}	303 mA	332 mA	355 mA	364 mA
V_{emf}	11 mV	51 mV	68 mV	76 mV
V_{pot}	0 mV	116 mV	191 mV	255 mV
K_p	20	20	20	20
K_i	0	0	0	0

Z merania rýchlosti otáčania z tab. 4 vyplýva, že otáčky možno regulovať v krajných polohách potenciometra plynulo približne od 500 min⁻¹ po 2500 min⁻¹, čomu treba prispôbiť aj prevodový mechanizmus robota.

5 Diskusia

Pre navrhnutú aplikáciu bol zvolený mikropočítač, ktorý obsahuje generátor PWM signálu, pretože softvérové generovanie troch časovo na sebe závislých PWM signálov by bolo príliš zložité a procesor by pre náročnosť tohto podprogramu nebol schopný pracovať s inými dátami. Princíp generátora PWM signálu na čipe popisuje Fuksa (2001). Takéto jednočipové mikropočítače obsahujú register, v ktorom sa priamo navolí hodnota PWM a signál je vytváraný priamo, a nezávisle na prebiehajúcim programe. Generovanie PWM tiež neovplyvňuje prebiehajúci program a nedochádza tak k strate údajov. Vďaka tomu sme sa mohli zamerať na komunikáciu s používateľom, na riadenie samotného motora a zisťovanie stavu riadenia.

Veľkou výhodou modulu je možnosť pripojenia ľubovoľného 12V BLDC motora, samozrejme podľa možností výkonovej časti regulátora. Používateľ len potrebuje poznať parametre daného motora, ktoré sú spomínané v odseku 4.2 a prepísať ich v príslušnej časti obslužného programu. Potom už len stačí zapísať takto upravený program do pamäte mikrokontroléra.

Ďalšou výhodou je možnosť komunikácie s používateľom pomocou PC, kde môže používateľ priamo navoliť niektoré parametre ovplyvňujúce chod motora bez prepisovania obslužného programu. Používateľ je tiež informovaný o stave riadenia zadaním jednoduchého príkazu, kedy sa na obrazovke vypíšu číselné hodnoty riadenia. Zariadenie je teda schopné komunikovať vo vysokej miere nielen s používateľom, ale aj s inými zariadeniami, čo je veľmi potrebné pre plánovanú implementáciu do robotického zariadenia.

Pri testovaní zhotoveného modulu sa nám však nepodarilo motor primerane regulovať v nízkych otáčkach, lebo krútiaci moment vytváraný na hriadeli motora vykazoval pulzný priebeh a robotické zariadenie by nebolo dobre ovládateľné pri nízkych rýchlostiach. Tento problém bol vyriešený prevodovým mechanizmom z hriadeľa motora na hnacie kolesá, i keď použitý motor by bol schopný pohybovať robotom bez prevodovky. Tým sa dosiahlo, že pri potrebe pomalého pohybu robota,

najmä pri zisťovaní rozmerov prekážok, sú otáčky samotného BLDC motora vyššie a nedochádza tak k pulznému priebehu výstupného krútiaceho momentu, ako by to bolo pri priamom spojení hnacích kolies s motorom. Tento problém by sa dal tiež čiastočne odstrániť použitím BLDC motora s viacerými magnetickými pólmi, prípadne použitím krokového motora. S tým by sa však znížila aj maximálna možná rýchlosť pohybu robota.

Robot, v ktorom je navrhnuté zariadenie s úpravami implementované, je napájaný z batérie a vykonáva množstvo ďalších funkcií. Preto je použitie impulznej šírkovej modulácie na reguláciu výkonu motora veľmi výhodné z hľadiska úspory energie. Pri tomto type regulácie dochádza k pomerne malým stratám na výkonových prvkoch, ktoré popisuje Hlava (2000, s.76 až 79). V spojení s BLDC motorom, kde nedochádza k stratám ani na mechanickom komutátore ako pri jednosmerných motoroch, je preto takýto systém pohonu robota veľmi efektívny a úsporný.

6 Návrhy na využitie výsledkov

Bezkomutátorové motory sa nachádzajú v mnohých robotických aplikáciách pre ich bezúdržbovú prevádzku a vysokú účinnosť. Mojou úlohou bolo vytvoriť modul PWM riadenia týchto motorov, ktorý bude implementovaný do robotického zariadenia vyvíjaného na katedre. Vyrobené zariadenie na vývojovej doske výborne poslúžilo pri experimentovaní a testoch s pohonom robotického zariadenia, a prispelo tak ku konečnému návrhu pohonného modulu zariadenia, kde sa využíva senzorové riadenie bezkomutátorových jednosmerných motorov.

Tento modul riadenia, spolu s bezkomutátorovým motorom, by sa tiež uplatnil v aplikáciách, ktoré sú napájané z batérie a vzniká tak potreba šetrenie dodávanej energie. Ide najmä o modelárske pohonné jednotky, u ktorých je požadovaná vysoká účinnosť v oblasti spotreby energie. Zariadenia, u ktorých sa používajú bežné jednosmerné motory nemôžu dosahovať vysokú účinnosť, pretože veľké množstvo dodávanej energie sa pretransformuje na mechanickom komutátore iskrením na neúčinné teplo.

Iskrenie na mechanickom komutátore jednosmerných motorov vytvára elektromagnetické rušenie, ktoré je veľmi nežiadúce. Použitím BLDC motora je toto rušenie odstránené a dochádza už len k rušeniu vytváranému spínaním výkonových prvkov. Tento druh rušenia je však možné jednoduchšie eliminovať použitím filtrov. Preto sa BLDC motory používajú v HDD, CD a DVD mechanikách, ventilátoroch PC zdrojov a ako súčasť chladičových ventilátorov procesorov.

Navrhnuté zariadenie po prerobení výkonovej časti by sa tiež mohlo použiť ako regulátor ventilátorov na odsávanie škodlivých výparov z výrobných častí podnikov. Tu by sa využila najmä možnosť centrálného riadenia pomocou kontrolného PC, ktorý by vyhodnocoval koncentráciu škodlivých plynov na danom pracovisku a spúšťal by tak príslušný ventilačný systém. Určite by týmto spôsobom došlo k úsporám energie, pretože by sa tým nahradili indukčné motory pracujúce na plný výkon počas celej pracovnej doby.

BLDC motory sú bezúdržbové podobne ako trojfázové indukčné motory. Riadenie trojfázových indukčných motorov je však oveľa zložitejšie a k pomeru svojej hmotnosti nedosahujú taký výkon ako BLDC motory. Preto nahradením indukčných motorov BLDC motormi v rôznych aplikáciách určite dospejeme k mnohým výhodám.

7 Záver

V dnešnej dobe vzrastá potreba automatizácie a robotizácie výroby, čím sa zaručuje menšia chybovosť produkovaných výrobkov a zároveň štandardizuje ich kvalita, ktorá je vnímaná u konečného zákazníka. Neustále nároky na zvyšovanie konečnej kvality výrobku, nútia konštruktérov vyvíjať stále účinnejšie stroje, ktoré sa podieľajú na procesoch produkcie. Vysoká účinnosť strojov spočíva v ich možnostiach vyrábať veľké množstvá kvalitných produktov, pričom spotreba energií nutných pre produkciu ostáva na veľmi nízkej úrovni. Preto stroje, v ktorých sa pre pohon mechanizmov využívajú BLDC motory, možno označiť pojmom „účinné“, pretože pri ich riadení nedochádza k zbytočným stratám elektrickej energie. Vďaka pomeru hmotnosti motora k jeho výkonu tiež dochádza k úsporám materiálu, ktorý je potrebný na jeho výrobu a tým sa šetrí ďalšia energia. Používateľ týchto strojov chce mať tiež prehľad o spotrebovanej energii a tak nastáva nutnosť schopnosti komunikácie strojov s nadriadenými systémami, ktorými sú PC. Používateľ prostredníctvom nich získava informácie o stave riadenia a v prípade potreby môže jednoducho vykonať korekcie systému. Schopnosť komunikácie a vysokú účinnosť možno teda považovať za základné črty v súčasnosti vyrábaných strojov.

V diplomovej práci je opísaný modul PWM riadenia BLDC motorov, ktorý bol po ďalšom vývoji implementovaný do robotického zariadenia vyvíjaného na Katedre elektrotechniky, automatizácie a informatiky TF SPU v Nitre. Danú úlohu sa mi podarilo vyriešiť pomocou vývojovej dosky od firmy Silicon Laboratories. Základ modulu tvorí mikropočítač C8051F310, ktorý riadi otáčky motora a komunikuje s používateľom pomocou USB a PC. Časť informácií poskytuje aj pomocou LED diód implementovaných na vývojovej doske. K navrhnutému regulátoru je možné pripojiť akýkoľvek 12V BLDC motor so známymi parametrami v rámci možností výkonovej časti regulátora.

Verím, že táto práca pomôže i pri ďalších návrhoch mikroprocesorových systémov, zvlášť v oblasti riešenia problémov s pohonom robotických zariadení, u ktorých je potreba zisťovania presného stavu riadenia pohonu nevyhnutná na zabezpečenie ich spoľahlivosti.

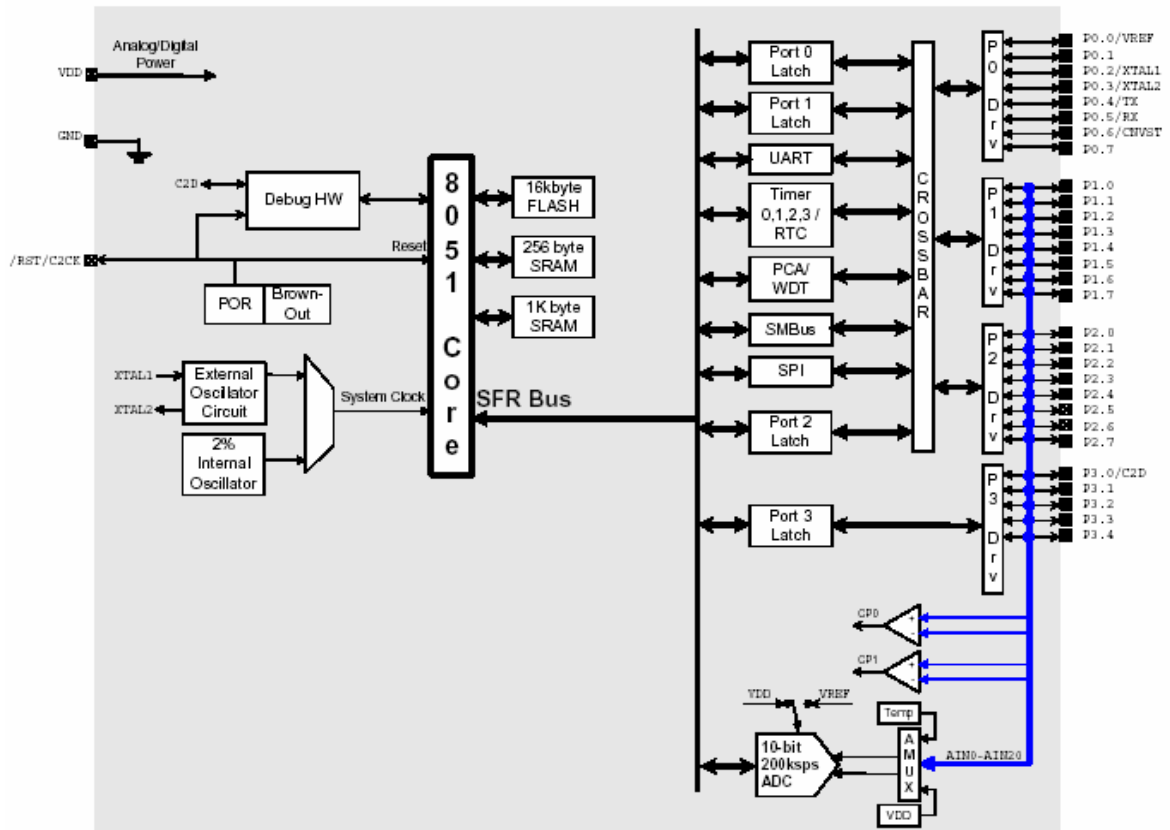
8 Zoznam použitej literatúry

1. 80C51 Family Programmer's Guide and Instruction Set. Dostupné na:
<http://mikroprocesory.sk/sites/default/files/programmers_guide.pdf>
2. AN1946. Sensorless BLDC Motor Control and BEMF Sampling Methods.
Dostupné na: <<http://www.st.com/stonline/books/pdf/docs/10401.pdf>>
3. Atmel – firemná literatúra:
AVR194. Brushless DC Motor Control using Atmega32M1. Dostupné na:
<http://www.atmel.com/dyn/resources/prod_documents/doc8012.pdf>
4. Boršč, M. 2005. Riadenie mechatronických systémov. AT&P journal, roč.6, 2005,
s. 99-103
5. Burkhard, M. 2003. C pro mikrokontroléry, μ C & praxe. BEN – technická
literatura. Praha 2003.
6. C8051F020 C Programming. Dostupné na:
<<http://www.massey.ac.nz/~chmessom/Chapter%206%20C%20ProgrammingFinal.pdf>>
7. Dal Y. Ohm – Jae H. Park. About Comutation and Current Control Methods for
Brushless Motors. Dostupné na: <<http://pdf-search-engine.com/brushless-motor-pdf.html>>
8. Electrical Energy Technology. Brushless DC Motor – Chapter 14. Dostupné na:
<http://services.eng.uts.edu.au/cempe/subjects_JGZ/eet/eet_ch14.pdf>
9. Fuksa, M. 2001. Pulzne šírková modulace ve spojení s 8051. Dostupné na
internetu: <<http://www.volny.cz/fuksan/povidani/pwm/pwm.htm>>
10. Hankovec, D. 2008. Vývoj a výroba elektroniky na zakázku. Jak se naučit
programovat? (nejen jednočipy). 1. až 6. díl. Dostupné na internetu:
<http://www.dhservis.cz/dalsi_1/obsah-popis.htm>
11. Hlava, J. 2000. Prostředky automatického řízení. 2.vyd. Praha: vydavatel'stvo
ČVUT, 2000. Regulační vlastnosti elektrických pohonů a výkonových členů, s.
72-83
12. Hrubý, D. – Pokorný, K. 2001. Aplikácie jednočipových mikropočítačov na
automatizáciu merania. Monografia.Nitra: SPU, 2001. 100 s. ISBN 80-137- 892-5
13. Jovankovič, J.- Žalman, M. 2006. Technika pohonov. AT&P journal, roč.3, 2006,
str.73 – 74

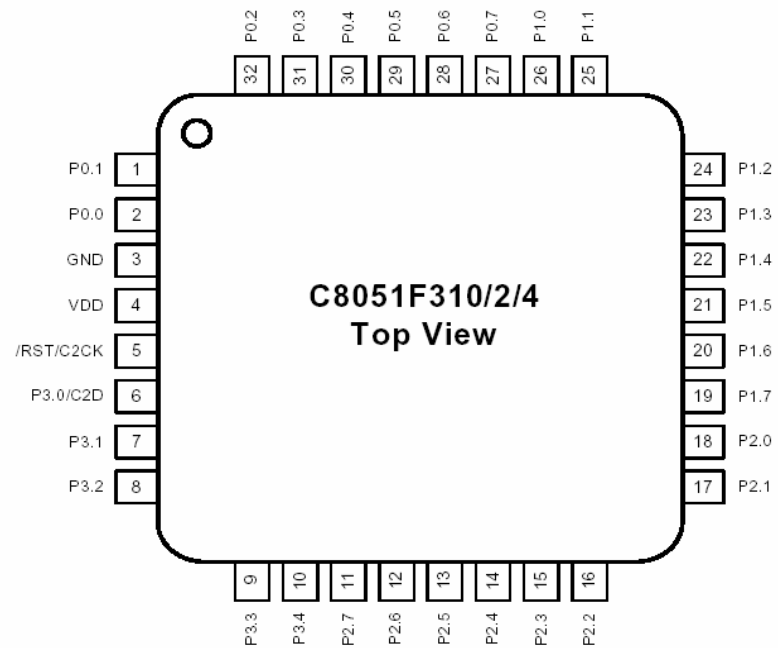
14. Silicon Laboratories – firemná literatúra:
AN208. Sensorless Brushless DC Motor Reference Design.
Datasheet C8051F310
Dostupné na: <<http://www.silabs.com>>
15. Microchip – firemná literatúra: AN957. Sensored BLDC Motor Control Using dsPIC30F2010. Dostupné na:
<http://www.jimfranklin.info/microchipdatasheets/00885a.pdf>
AN901. Using the dsPIC30F for Sensorless BLDC Control. Dostupné na:
<<http://www.microchip.com/downloads/en/AppNotes/Sensorless%20BLDC%200901a.pdf>>
AN885. Brushless DC (BLDC) Motor Fundamentals. Dostupné na:
<<http://www.microchip.com/downloads/en/AppNotes/00885a.pdf>>
16. Motion&Control. Motor technologies. Dostupné na:
<<http://www.allegromicro.com/en/Products/Design/compumot/a17a19.pdf>>
17. Pulse-width modulation. Dostupné na internete:
<http://en.wikipedia.org/wiki/Pulse-width_modulation>

Prílohy

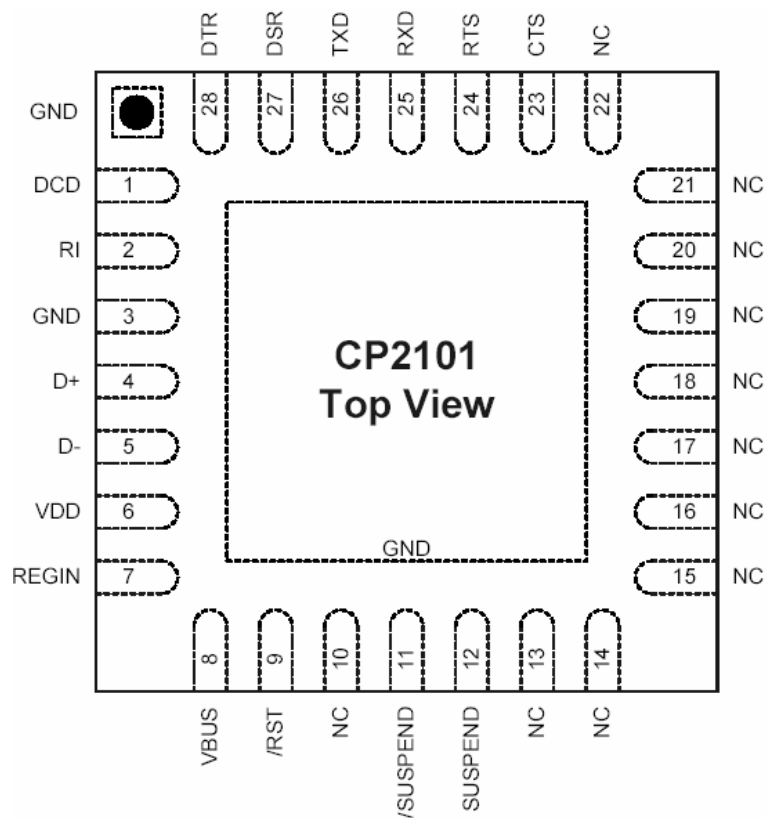
Príloha č.1 Blokový diagram C8051F310



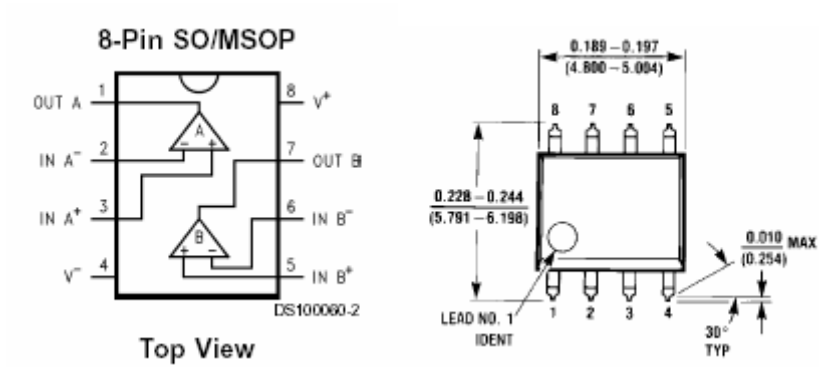
Príloha č.2 Konfigurácia pinov puzdra C8051F310



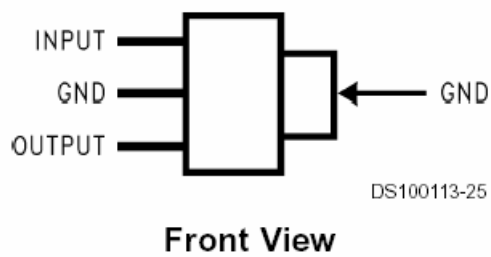
Príloha č. 3 Konfigurácia pinov puzdra USB – UART prevodníka CP2101



Príloha č. 4: Rozmiestnenie pinov puzdra LMV358



Príloha č.5: Puzdro stabilizátora LM2937 – 33



Príloha č.6: Súpis súčiastok

Kondenzátory:

C1	470u/25V elektrolytický 10x16mm
C2, C4, C9, C17	1u keramický SMD/1206
C3, C5, C6, C7, C8, C10, C11, C12, C13, C14, C16, C22, C23, C24, C25	100n keramický SMD/805
C18, C19, C20	470n keramický SMD/805
C21, C15	2,7n keramický SMD/805

Rezistory:

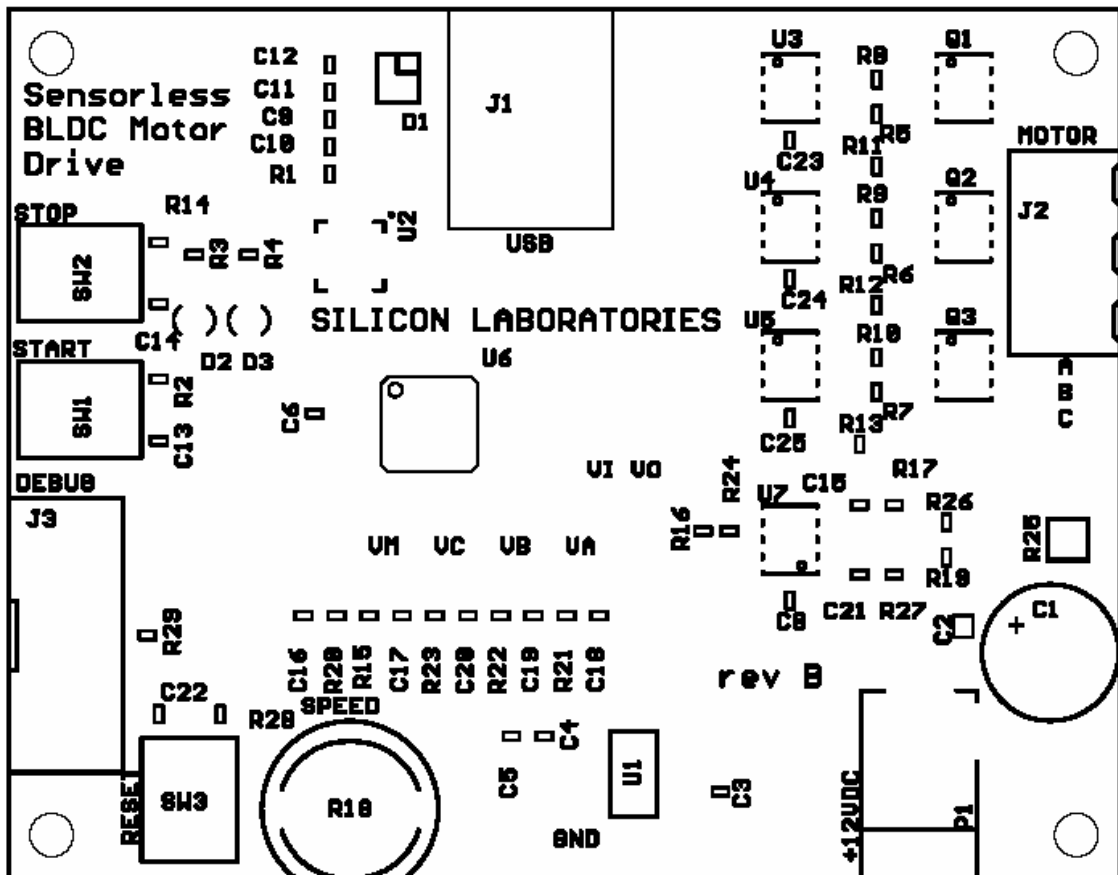
R1, R2, R11, R12, R13, R14, R28	10k SMD/805
R3, R4, R19, R20, R26, R29	1k SMD/805
R5, R6, R7, R8, R9, R10	100 SMD/805
R15	11.0k SMD/805
R16	1.13k SMD/805
R17, R27	25.5k SMD/805
R18	potenciometer 150
R21, R22, R23	2K SMD
R24	8.2k SMD
R25	0.02 SMD/1815

Konektory a mikrotlačidlá:

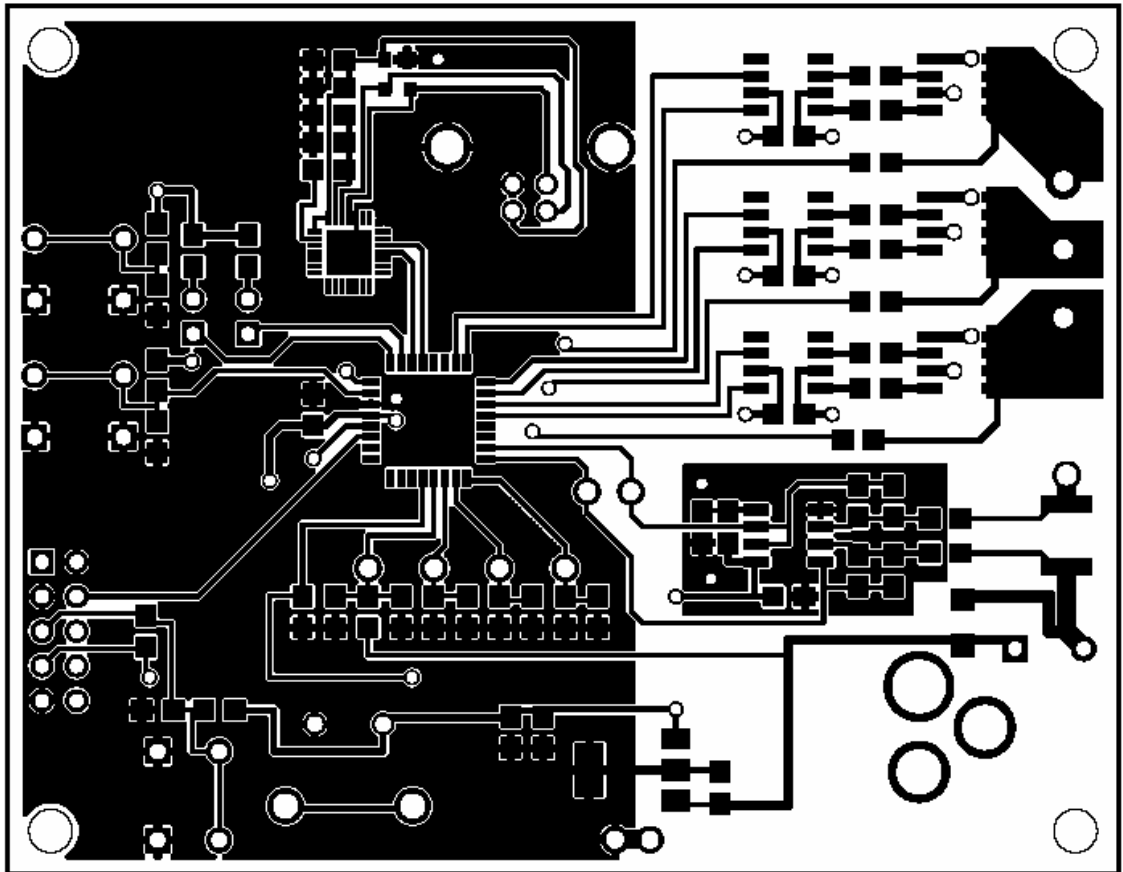
SW1, SW2, SW3	mikrotlačidlo 6mm
P1	napájací konektor 2.1mm
J1	USB konektor
J2	konektor pre pripojenie motora 3x5mm
J3	DEBUG konektor 2x5x10mm

Integrované obvody a polovodiče:

U1	LM2937-33/SOT223
U2	CP2101/MLP20
U3, U4, U5	TC4428/SO8
Q1A – Q3A, Q1B – Q3B	IRF7309/SO8
U6	C8051F310/QFN32
U7	LMV358/SO8
D1	ochranná trojdióda pre USB
D2, D3	LED dióda 3mm, červená



Príloha č.8: DPS (pohľad z vrchnej strany)



Príloha č.9: DPS (pohľad zo spodnej strany)

