



**České vysoké učení technické v Praze**

**Fakulta elektrotechnická**

**Katedra elektrických pohonů a trakce**

**Řízení stejnosměrného pohonu pomocí platformy  
TI**

**Control of DC drive with the Help of TI Board**

Bakalářská práce

Studijní program: Elektrotechnika, energetika a management

Studijní obor: Aplikovaná elektrotechnika

Vedoucí práce: Ing. Jan Bauer Ph.D.

**Tomáš Hošek**

**Praha 2017**

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Hošek** Jméno: **Tomáš** Osobní číslo: **420172**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra elektrických pohonů a trakce**  
Studijní program: **Elektrotechnika, energetika a management**  
Studijní obor: **Aplikovaná elektrotechnika**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Řízení stejnosměrného pohonu pomocí platformy TI**

Název bakalářské práce anglicky:

**Control of DC drive with the Help of TI Board**

Pokyny pro vypracování:

- 1) Popište metody řízení otáček DC motorků
- 2) Seznamte se vývojovým kitem od firmy TI a programovacím prostředím Code Composer
- 3) Navrhněte DSP pro realizaci řízení DC motorku pomocí vývojového kitu TI
- 4) V prostředí CodeComposer vytvořte program pro řízení otáček DC motorku a běžící text

Seznam doporučené literatury:

- [1] Lettl J., Cerovsky Z., Pavelka J. ? Vykonova elektronika, skripta CVUT FEL
- [2] Javurek J. ? Regulace moderních elektrických pohonů, Grada
- [3] Pavelka, J., Zdenek, J., - Elektrické pohony a jejich řízení. 1. vyd., Praha, CVUT, 2010.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Ing. Jan Bauer Ph.D., katedra elektrických pohonů a trakce FEL**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **09.05.2017**

Termín odevzdání bakalářské práce: **26.05.2017**

Platnost zadání bakalářské práce: **30.06.2018**



Podpis vedoucí(ho) práce



Podpis vedoucí(ho) ústavu/katedry



Podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.  
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

15.5.2017

Datum převzetí zadání



Podpis studenta

# Prohlášení

„Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.“

V Praze dne .....

.....

Podpis autora

# Poděkování

Děkuji především vedoucímu mé bakalářské práce panu Ing. Janu Bauerovi Ph.D. za námět bakalářské práce, za obstarání potřebných komponentů, cenné připomínky, rady a též za ochotnou pomoc při hledání nedostatků a jejich odstraňování.

Děkuji i všem ostatním, kteří mě při tvorbě závěrečné práce podporovali.



# Abstrakt

Tato bakalářská práce je zaměřena na konkrétní použití experimentální vývojové destičky C2000™ LaunchPad. Vývojový kit je použit pro řízení stejnosměrného motoru a světelných diod, které budou umístěny na rotující části sestavy a řízeným rozsvěcováním a zhasínáním těchto diod se docílí efektu běžícího textu. Práce popisuje volbu komponentů, návrh a realizaci elektrického propojení součástek s vývojovým kitem. V práci je zároveň popsán zdrojový kód pro mikroprocesor nacházející se na vývojové desce tvořený pomocí IDE Code Composer a návrh desky plošného spoje pomocí SW Formica 4.40 Layout.

## Klíčová slova

C2000™ LaunchPad, Code Composer, deska plošných spojů, Formica 4.40 Layout, řízení pohonu, stejnosměrný motor, světelná dioda

# Abstract

This bachelor thesis focuses on the application of the C2000™ LaunchPad experimental development board. The development kit is used to control the DC motor and the light diodes that will be placed on the rotating parts of the assembly. The effect of the running text will be created by controlled by turning on and off these diodes. The thesis describes component selection, design and realization of electrical connection of components with development kit. The thesis also describes the source code for the microprocessor developed using the IDE Code Composer and the design of the printed circuit board using SW Formica 4.40 Layout.

## Key words

C2000™ LaunchPad, Code Composer, PCB, Formica 4.40 Layout, control of drive, DC motor, light emitting diode

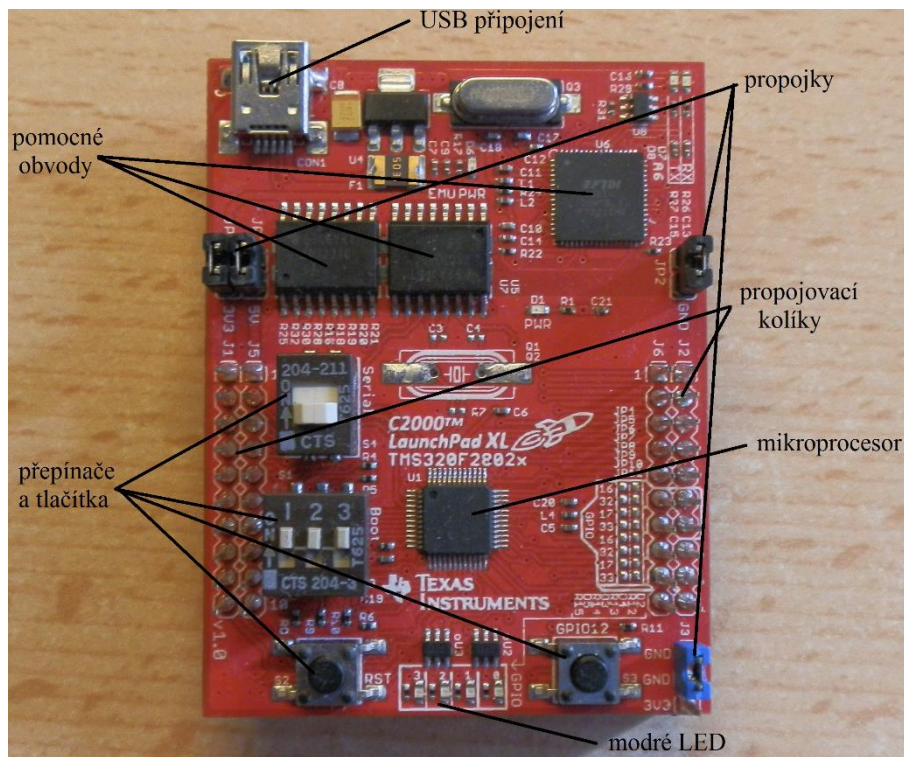
# Obsah

<b>1</b>	<b>Rozbor problematiky .....</b>	<b>1</b>
<b>2</b>	<b>Stejnoseměrný motor.....</b>	<b>2</b>
<b>2.1</b>	<b>Princip funkce stejnosměrného motoru .....</b>	<b>4</b>
<b>3</b>	<b>Metody řízení otáček DC motorků .....</b>	<b>7</b>
<b>3.1</b>	<b>Stejnoseměrný měnič napětí .....</b>	<b>9</b>
<b>3.1.1</b>	<b>Stejnoseměrný měnič napětí s odporovou zátěží .....</b>	<b>9</b>
<b>3.1.2</b>	<b>Stejnoseměrný měnič napětí s R-L zátěží .....</b>	<b>10</b>
<b>3.1.3</b>	<b>Stejnoseměrný měnič napětí s aktivní zátěží .....</b>	<b>11</b>
<b>3.1.4</b>	<b>Použití stejnosměrného měniče napětí .....</b>	<b>11</b>
<b>3.2</b>	<b>Obousměrné řízení otáček DC motorku .....</b>	<b>12</b>
<b>3.3</b>	<b>Elektronické spínače.....</b>	<b>14</b>
<b>4</b>	<b>Vývojová deska, vývojové prostředí, programovací jazyk.....</b>	<b>17</b>
<b>4.1</b>	<b>Vývojová deska od firmy Texas Instruments.....</b>	<b>17</b>
<b>4.2</b>	<b>Vývojové prostředí .....</b>	<b>18</b>
<b>4.3</b>	<b>Programovací jazyk C.....</b>	<b>20</b>
<b>5</b>	<b>Popis vnějšího obvodu.....</b>	<b>21</b>
<b>5.1</b>	<b>Dolní deska .....</b>	<b>21</b>
<b>5.2</b>	<b>Horní deska.....</b>	<b>22</b>
<b>5.3</b>	<b>Popis DC motorku.....</b>	<b>22</b>
<b>5.4</b>	<b>Popis součástí .....</b>	<b>23</b>
<b>6</b>	<b>Popis celé sestavy .....</b>	<b>28</b>
<b>7</b>	<b>Popis programu .....</b>	<b>31</b>
<b>7.1</b>	<b>Běžící text .....</b>	<b>33</b>
<b>8</b>	<b>Problémy a jejich řešení .....</b>	<b>35</b>
<b>9</b>	<b>Závěr .....</b>	<b>36</b>
	<b>Seznam obrázků .....</b>	<b>37</b>
	<b>Seznam tabulek .....</b>	<b>38</b>
	<b>Seznam použité literatury .....</b>	<b>39</b>
	<b>Příloha č. 1: Blokové schéma vývojové desky C2000 .....</b>	<b>a</b>
	<b>Příloha č. 2: Programový kód .....</b>	<b>b</b>



# 1 Rozbor problematiky

Jak z názvu vyplývá, cílem mé bakalářské práce bylo seznámit se s řízením stejnosměrného motorku pomocí naprogramovaného mikroprocesoru a s otáčkami motorku sesynchronizovat blikání světelných diod tak, aby výsledný efekt vytvořil běžící text. S vedoucím práce jsme se dohodli na použití vývojového přípravku LaunchPad C2000<sup>TM</sup> s integrovaným mikroprocesorem od firmy Texas Instruments. Vývojovou destičku, zobrazenou na Obr. 1, bylo potřebné propojit s DC<sup>1</sup> motorkem, LED<sup>2</sup> a pomocnými obvody. Ve výsledné fázi bude vše rotovat spolu s motorkem kolem osy motorku, a proto nebylo možné pomocné obvody vytvořit na nepájivém kontaktním poli. Z tohoto důvodu jsem navrhl dvě desky plošných spojů a nazval jsem je horní a dolní podle výsledného umístění. Propojení součástek a všech komponentů jsem musel nejprve vymyslet, poté pomocí programu Formica 4.40 Layout nakreslit rozmístění součástek a jejich propojení na desce plošných spojů a pak obě desky nechat vyrobít. Poté jsem sestavil základnu, na které bude rotační část umístěna a ze které bude také napájena. Dalším krokem bylo naprogramovat mikroprocesor, k čemuž jsem použil program CodeComposer. Některé části programu bylo možné odzkoušet mimo sestavu pomocí nepájivého kontaktního pole, například správnou funkci časovače, jednotky generující PWM<sup>3</sup> signál, vstupy a výstupy vývojového kitu. Další části programu již nešly zkusit mimo hotovou sestavu, protože řízení otáček DC motorku malého výkonu bylo ovlivněno setrvačností rotující části sestavy. Také řízené rozsvěcování a zhasínání LED bylo vhodnější zkusit spolu s řízením otáček pomocí zpětnovazebního obvodu.



Obr. 1: Vývojová deska LaunchPad C2000<sup>TM</sup>

<sup>1</sup> DC – označení pro stejnosměrné systémy

<sup>2</sup> LED = Light Emitting Diode, neboli světelná dioda

<sup>3</sup> PWM = Pulse Width Module, pulsně šířková modulace, způsob řízení spotřebičů

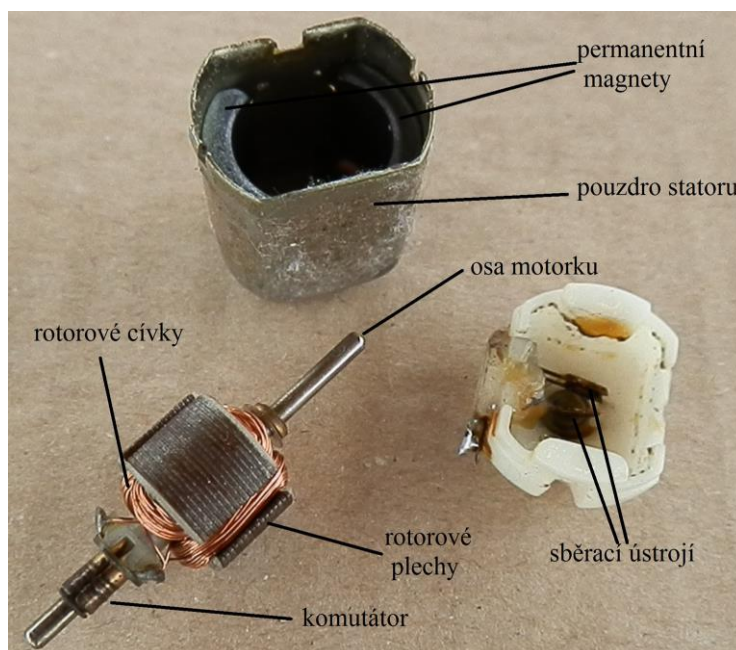
## 2 Stejnospměrný motor

Obecně: Elektrický stroj je zařízení přeměňující elektromechanicky energii na principu elektromagnetické indukce. Mezi elektrické stroje řadíme také motor, který přeměňuje elektrickou energii na mechanickou energii. [1]

Stejnospměrný motor (též DC motor) se skládá ze statoru a rotoru. Stator slouží k vytváření magnetického pole. Toto pole je vytvářeno buďto budicím vinutím umístěným na hlavních pólech statoru, nebo permanentními magnety. Permanentní magnety vytvářejí magnetické pole o konstantní velikosti a není možné ji jakkoliv měnit. Proto není možné stejnosměrný motor s tímto typem buzení odbuzovat ani přibuzovat. To je určitá nevýhoda, a proto se permanentní magnety používají pouze u stejnosměrných motorů menších výkonů. V mé sestavě je umístěn právě tento typ stejnosměrného motorku o velmi malém výkonu.

Na rotoru se nachází vinutí tvořené vodiči rozloženými na jeho vnějším obvodu. Rotor je tvořen transformátorovými plechy s drážkami, ve kterých se nacházejí vodiče cívek. Cívky jsou vinuty tak, aby rovina, na které leží, procházela osou rotoru. Na jedné rovině se proto nacházejí vodiče jedné cívky jdoucí jedním směrem rovnoběžně s osou rotoru a ve stejné vzdálenosti přes osu na téže rovině se nacházejí vodiče stejné cívky jdoucí opačným směrem. Konce cívky jsou vyvedeny na lamely komutátoru. Lamely jsou vzájemně izolovány. Na tyto lamely dosedají kartáče sběracího ústrojí. Sběrací ústrojí bývá spojeno se státorem, při chodu motoru je v klidu. Komutátor se otáčí kolem osy motoru spolu s rotorem. Tento systém je velmi důmyslný, avšak přináší určitá omezení a je zdrojem poruch. V rotoru se indukuje napětí, proto nazveme rotorový obvod kotva.

Výše popsaná konstrukce rotoru se používá pro stejnosměrné motory s cizím buzením v podobě cívek umístěných na pólech nad pólovými nástavci připevněných k vnitřní stěně statoru. Mezi pólovými nástavci a rotorem je vzduchová mezera. Tu se snaží výrobci minimalizovat. Se vzrůstající se velikostí vzduchové mezery klesá velikost magnetického toku a zhoršují se tak vlastnosti stroje. Vzduchová mezera se pohybuje mezi 1 až 15 mm podle toho, jak je stroj velký. [1]



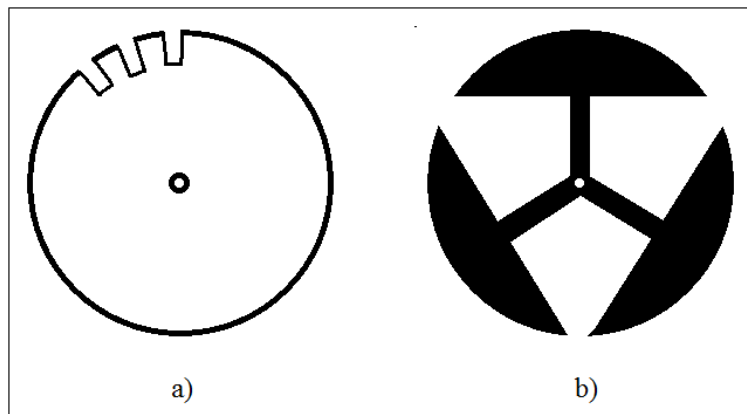
Obr. 2: Rozebraný stejnosměrný motorek malého výkonu

U motorků s permanentními magnety na statoru jsou cívky v rotoru uspořádány odlišně. Rozebraný elektromotorek s permanentními magnety je zobrazen na Obr. 2. Odlišný je také tvar elektrotechnických plechů. Plechy jsou navzájem izolovány, stejně jako u předchozího typu rotorů, avšak se v nich nenacházejí drážky pro uložení vodičů.

Tvary elektrotechnických plechů pro výrobu obou typů rotorů jsou znázorněny na Obr. 3.

Obrázek 3 a) znázorňuje rotorový plech s drážkami rozmístěnými po celém obvodu. V obrázku je drážkování naznačeno pouze na části obvodu.

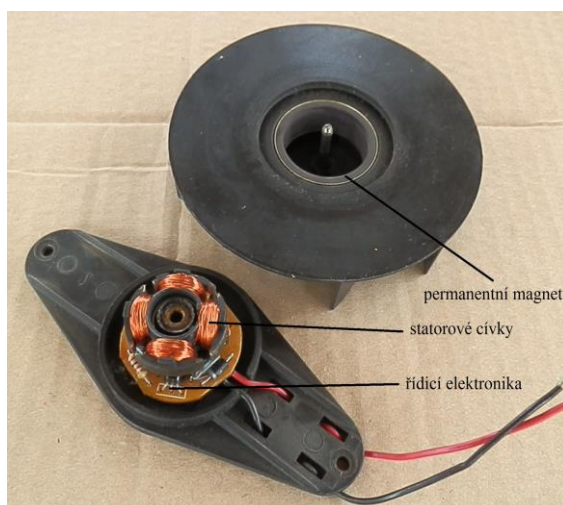
V druhém případě (obr. 3 b)) je zobrazen rotorový plech používaný u velmi malých DC motorků užívaných v nejrůznějších aplikacích například v hračkách. Cívky jsou navinuty mezi osou rotoru a pólovým nástavcem, kde jsou plechy zúžené. V zobrazeném typu plechu jsou 3 pólové nástavce a místa pro tři cívky. Pod každým pólovým nástavcem se tedy nachází jedna cívka. Cívky jsou vinuty z měděných lakovaných drátků, jejichž průměr, resp. průřez závisí na jmenovitém výkonu motorku. Oba vývody cívky jsou přivedeny na lamely komutátoru. Těch je v popisovaném motorku stejný počet, jako cívek. Každý konec cívky je vyveden nejen na lamelu, ale také spojen s koncem sousední cívky. Mnou použitý motorek odpovídá zde popsanému typu.



Obr. 3: Typy rotorových plechů

Speciálním typem DC motoru s malým výkonem je motorek, jenž má cívky ve statorové části a permanentní magnety rotují, poněvadž jsou umístěny na rotoru. Tento typ motorku nemá komutátor, a proto musí být cívky připínány na napájecí napětí pomocí elektronického obvodu, který se umísťuje na stator. Příkladem motorku s takto provedenou konstrukcí je motorek pohánějící ventilátor v počítači nebo notebooku.

Motorek je zabudován přímo do ventilátoru, aby nezabíral příliš mnoho místa. Pohled na rozebraný motorek je na Obr. 4.



Obr. 4: Ventilátor na stejnosměrné napětí malého výkonu

## 2.1 Princip funkce stejnosměrného motoru

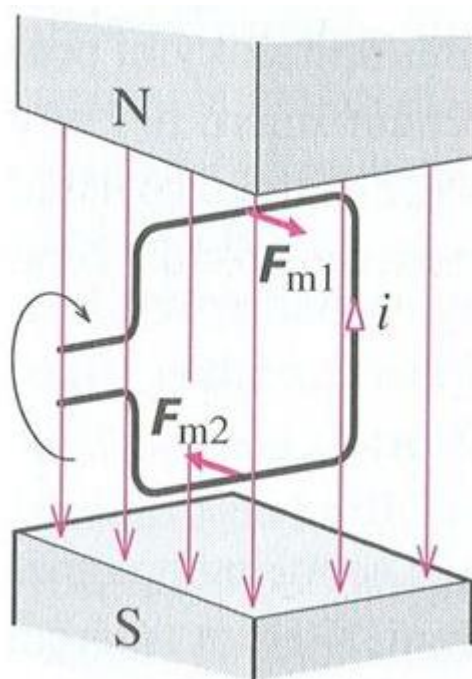
Přeměna elektrické energie na energii mechanickou. Jelikož jsem použil motorek s permanentními magnety, zaměřím se pouze na buzení s konstantní hodnotou. Stator vytváří magnetické pole procházející od severního pólu magnetu k jižnímu skrz rotor. Přes kartáče sběracího ústrojí a přes lamely komutátoru protéká proud vždy tou cívkou, která je v těsné blízkosti pólů magnetu, protože je zde magnetické pole nejsilnější. Tím, že se vodič nachází v magnetickém poli a je protékán elektrickým proudem, bude na něj působit síla. Směr působení síly určíme z pravidla levé ruky: „*Položíme-li levou ruku na vodič pohybující se v magnetickém poli tak, aby siločáry vstupovaly do dlaně a prsty ukazovaly směr proudu, pak palec ukazuje směr síly působící na vodič.*“ [1]

Velikost síly lze určit z následujícího vzorce.

$$F = B \cdot l \cdot I \quad (N; T, m, A) \quad (1)$$

kde  $F$  je síla působící na vodič,  $B$  je magnetická indukce v místě vodiče,  $l$  je délka vodiče umístěného v magnetickém poli a  $I$  je proud protékající vodičem.

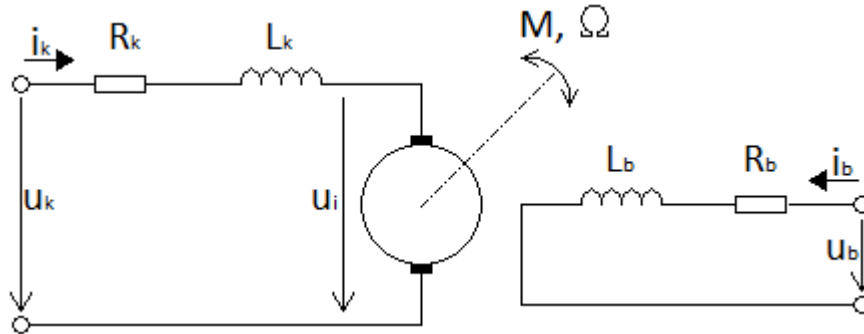
Princip otáčející se cívky, kterou v magnetickém poli prochází elektrický proud, přibližuje Obr. 5. [32]



Obr. 5: Princip otáčející se cívky umístěné v magnetickém poli

Síla působí na vodič cívky v blízkosti severního pólu magnetu jedním směrem a u jižního pólu opačným směrem, protože elektrický proud protéká opačným směrem a zároveň magnetická síla působí v rotoru pouze jedním směrem. Tím pádem na vodič tvořící cívku protékající elektrickým proudem a umístěnou v magnetickém poli působí dvojnásobná síla oproti přímému vodiči. Síla se vynásobí počtem závitů cívky a dostaneme výslednou sílu, která otáčí rotorem. Při pootočení se přerušuje obvod aktivní cívky a vytvoří se nový obvod obsahující následující cívku. Toto je způsobeno zmiňovaným komutátorem, který zajišťuje plynulé přepínání jednotlivých cívek. Komutátor je tedy jakýsi automatický přepínač, který spíná vždy tu cívku, která je v nejvýhodnější pozici pro vyváření elektromagnetického momentu. [1]

Každý elektrický stroj lze popsat rovnicemi matematického modelu vycházejícími z obvodového schématu. Pro stejnosměrný cize buzený motor je obvodové schéma znázorněno na Obr. 6.



Obr. 6: Obvodové schéma stejnosměrného cize buzeného motoru

Pro matematický model stejnosměrného cize buzeného motoru platí následující rovnice.

$$u_k = R_k \cdot i_k + L_k \cdot \frac{di_k}{dt} + u_i \quad (V; \Omega, A, H, A, s, V) \quad (2)$$

$$u_b = R_b \cdot i_b + L_b \cdot \frac{di_b}{dt} \quad (V; \Omega, A, H, A, s) \quad (3)$$

$$u_i = k_1 \cdot \Phi \cdot \Omega \quad (V; V \cdot s \cdot Wb^{-1}, Wb, s^{-1}) \quad (4)$$

$$M = k_2 \cdot \Phi \cdot i_k \quad (Nm; Nm \cdot A^{-1} \cdot Wb^{-1}, Wb, A) \quad (5)$$

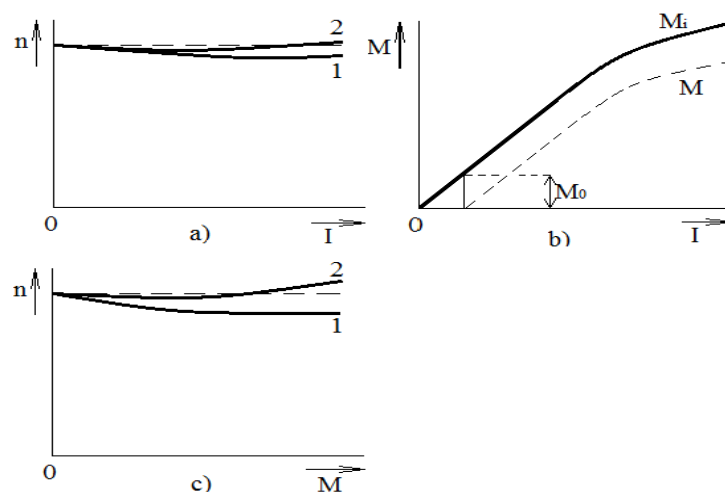
$$\Phi = f(i_b) \quad (Wb, A) \quad (6)$$

$$M - M_z = J_c \cdot \frac{d\Omega}{dt} \quad (Nm, Nm; kg \cdot m^{-2}, s^{-1}, s) \quad (7)$$

Kde  $u_k, i_k, R_k, L_k, u_i$  jsou elektrické veličiny kotevního obvodu,  $u_b, i_b, L_b, R_b$  jsou elektrické veličiny budícího obvodu,  $M$  je elektromagnetický moment a  $\Omega$  je otáčivá rychlost rotoru. Znak  $\Phi$  reprezentuje magnetický tok, který je vybuzen proudem procházejícím budící cívkou  $L_b$ .  $J_c$  je moment setrvačnosti soustavy, jednotkou je  $kg \cdot m^{-2}$ . [2]

Pro stejnosměrný motor s buzením permanentními magnety lze vynechat rovnici (3) a (6), neboť takovýto motor postrádá budící obvod. V rovnici (4) lze součin konstanty úměrnosti  $k_1$  a magnetického toku  $\Phi$  nahradit novou konstantou úměrnosti  $k$  a indukované napětí  $u_i$  bude přímo úměrné otáčivé rychlosti  $\Omega$ . Podobnou úvahu lze napsat pro rovnici 5, kdy hnací moment  $M$  bude přímo úměrný kotevnímu proudu  $i_k$ .

Následující obrázek zachycuje průběhy typických charakteristik stejnosměrného motoru. [1]



Obr. 7: Zatěžovací charakteristiky stejnosměrného cize buzeného motoru

Pro všechny tři charakteristiky uvažujeme proud tekoucí budícím obvodem za konstantní  $I_b = konst.$ , tím bude konstantní i magnetický tok  $\Phi$  a lze říci, že charakteristiky pro stejnosměrný motorek s buzením permanentními magnety budou přibližně stejné. Křivky charakteristik se mohou mírně lišit, protože DC motorek s permanentními magnety má menší výkon, rozměry, nižší napájecí napětí a menší proud. Součin  $k^* \Phi$  lze opět nahradit jedinou konstantou označenou  $k$ .

Na obrázku 7 a) je rychlostní charakteristika, což je závislost otáček  $n$  na proudu  $I_k$ , který teče kotevním obvodem. Rychlostní charakteristiku vyjádříme vztahem (8).

$$n = \frac{U_k - R \cdot I_k}{k} (s^{-1}; V, \Omega, I, s) \quad (8)$$

Označení  $U_k$  a  $I_k$  odpovídá veličinám  $u_k$  a  $i_k$  výše popsaným. Značení velkými písmeny se používá, je-li veličina v čase stálá (konstantní). Obdobně pro další veličiny vyskytující se v mé práci tento předpoklad platí.

Z obrázku vyplývá, že s rostoucím proudem dojde nejprve k mírnému snížení otáček, to je způsobeno úbytkem napětí na kotvě, poté se otáčky začnou mírně zvyšovat (křivka 1). Křivka 2 znázorňuje průběh charakteristiky při vyšších proudech v kotevním obvodu, kdy demagnetizační účinek reakce kotvy neboli zeslabení magnetického pole může převažovat nad vlivem úbytku napětí na kotevním obvodu. Otáčky se s rostoucím proudem začnou zvyšovat a mohou narušit stabilitu chodu motoru. [1]

Na obrázku 7 b) je znázorněna křivka momentové charakteristiky, závislosti momentu  $M$  na proudu  $I$ . V tomto případě uvažujeme neměnné napětí na kotvě, tedy  $U_k = konst.$  Závislost momentu na proudu je až do oblasti jmenovitého proudu lineární. K zakřivení přímky směrem k ose  $x$ , na níž vynášíme velikost proudu, dochází při proudech větších než jmenovitém proudu. Opět je to způsobeno zeslabením magnetického pole. Křivka tvořená plnou čarou a označená  $M_i$  je charakteristika vnitřního elektromagnetického momentu  $M_i$  motoru. Čárkovaně je zobrazena křivka momentové charakteristiky skutečného mechanického momentu na hřídeli  $M$ , který je menší o moment ztrát naprázdno  $M_0$ . [1]

Poslední obrázek 7 c) znázorňuje mechanickou charakteristiku. Závislost otáček  $n$  na momentu  $M$  je taktéž důležitý parametr stejnosměrných motorů. Taktéž je křivka tvořena pro konstantní kotevní napětí  $U_k$ . Mechanická charakteristika se velmi podobá rychlostní charakteristice, protože závislost momentu  $M$  na proudu  $I$  je lineární, jak je patrné z momentové charakteristiky. [1]

### 3 Metody řízení otáček DC motorků

Budeme-li uvažovat DC motorek s permanentními magnety na statoru, vytvářejícími konstantní magnetické pole, je možné měnit otáčky pouze změnou některé veličiny rotoru. Síla vytvářející otáčení rotoru závisí na velikosti elektrického proudu, protékajícího vodičem cívky. Vodič cívky se chová jako odpor, proto je procházející proud podle Ohmova zákona přímo úměrný napětí na svorkách cívky.

Proto se řízení otáček DC motoru provádí regulací napětí na cívkách rotoru. Toho lze dosáhnout různými způsoby.

- 1) napájení zdrojem s nastavitelným výstupním napětím,
- 2) napájení ze zdroje s konstantním výstupním napětím, které snižujeme zapojováním různých odporů do série s kotvou motoru
- 3) napájení konstantním napětím, v obvodu kotvy se nachází proměnný odpor
- 4) napájení časově proměnným napětím.

Nyní si jednotlivé typy řízení DC motorku rozebereme.

ad 1) Napájíme zdrojem s nastavitelným výstupním napětím, tím měníme velikost napětí na svorkách motoru. Tento způsob lze uskutečnit například tak, že budeme mít k dispozici transformátor s možností přepínání odboček na sekundárním vinutím a tento výstup usměrníme pomocí diod v Graetzově můstkovém zapojení. Zvlnění signálu na výstupu nebude pro správný chod motoru nežádoucí. [3]

ad 2) Napájíme zdrojem konstantního napětí. Napětí na svorkách zdroje lze snižovat sériovým zapojením odporu do obvodu kotvy. Odpor může být jeden, nebo se jich do obvodu kotvy připojí více a pro zvyšování otáčivé rychlosti se postupně vyřazují. Napětí na svorkách motoru se bude měnit skokově, proto se někdy používá lepší varianta popsána v následujícím bodě.

ad 3) Napájíme zdrojem konstantního napětí, do obvodu kotvy přidáme proměnný odpor. Tímto proměnným odporem bývá nejčastěji otočný potenciometr. Protože potenciometr je zapojený v sérii s kotevním obvodem, protéká jím veškerý výkon, který motorek požaduje. Potenciometr musí být na takový výkon určený. V případě DC motorků o malém výkonu to nebývá problém. Musíme si dále uvědomit, že průchodem proudu  $I$  odporem  $R$  se maří výkon závislý na velikosti odporu a velikosti procházejícího proudu. Elektrický výkon  $P_z$  přeměněný uvnitř odporu na teplo určíme ze vztahu (9). [4]

$$P_z = R \cdot I^2 \quad (W; \Omega, A) \quad (9)$$

Tento vztah platí i pro bod č. 2.

V jednoduchých aplikacích, ve kterých jsou použity DC motorky a je zapotřebí regulovat otáčky co nejjednodušeji, se tento způsob užívá, například u hraček (autíčka na autodráze).

ad 4) Způsobem nejvíce užívaným je řízení časově proměnným napájecím napětím. S tímto způsobem se setkáváme u řízení stejnosměrných motorů nejrůznějších výkonů. Kromě řízení otáček točivých strojů se tento způsob napájení používá i pro regulaci osvětlení.

Princip: Napájecí napětí zdroje má v čase neproměnnou velikost, je konstantní, označme jej  $U_{jm}$ . Tímto napětím budeme napájet spotřebič, v našem případě stejnosměrný motorek přes spínač V. (Proud se chová jako kapalina – „teče“ vodičem, proto se spínač značí V jako ventil.) Spínač zajistí, že napětí na svorkách motorku bude ve vypnutém stavu nulové  $U_{vyp} = 0$  a v zapnutém stavu bude totožné s napětím na svorkách napájecího zdroje  $U_{zap} = U_{jm}$ . Motorek tedy napájíme napětím dvou velikostí:  $0V$  a  $U_{jm}$ .

Periodickým spínáním a rozpínáním obvodu vytvoříme obdélníkový průběh (Obr. 8). Tento průběh lze nahradit konstantním napětím o velikosti střední hodnoty obdélníkového průběhu. [5]

Pro velikost střední hodnoty výstupního napětí platí vztah:

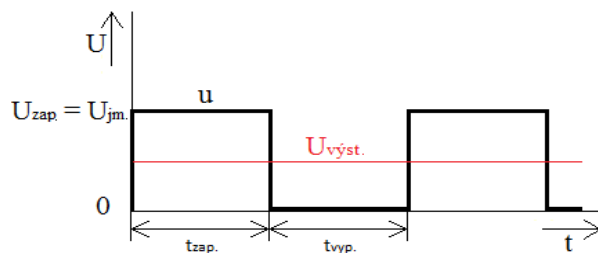
$$U_{výst.} = U_{zap.} \cdot \frac{t_{zap.}}{T} (V; V, s, s) \quad (10)$$

Kde  $t_{zap.}$  je doba zapnutí,  $T$  je perioda, pro kterou platí:

$$T = t_{zap.} + t_{vyp.} (s; s, s) \quad (11)$$

$T_{vyp.}$  Je doba, po kterou je na svorkách motorku nulové napětí.

Podíl doby zapnutí a periody lze nahradit tzv. zatěžovatelem  $z$ , nebo také střídou  $s$ . Bude-li  $s = 0$ , výstupní napětí bude nulové  $U_{výst.} = 0$ , pro  $s = 1$  bude výstupní napětí stejné jako napájecí. Při regulaci napětí se střída bude pohybovat mezi těmito krajními body. [5]



Obr. 8: Průběh napětí na zátěži při řízení podle bodu 4

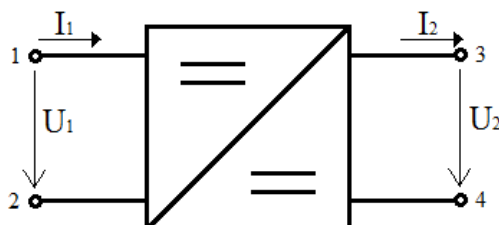
Jelikož jsou elektrické vstupní veličiny tohoto obvodu stejnosměrné a výstupní elektrické veličiny taktéž stejnosměrné, ale jiných hodnot, nazveme tento obvod stejnosměrným měničem napětí. Tyto měniče se užívají tam, kde je k dispozici napájecí zdroj stejnosměrného napětí o konstantní hodnotě, ale pro napájení spotřebiče je třeba velikost napětí v čase měnit. S těmito měniči se setkáváme především v trakci. [5]

Zajímavým příkladem může být lokomotiva řady 110, původně E 458.0, která byla postavena na začátku 70. let minulého století. Byla určena pro provoz na elektrifikovaných tratích se stejnosměrným napětím 3 kV v troleji. Z výroby byla tato řada vybavena odporovou regulací výkonu. Rozjezdové odporníky byly zapojeny sériově s trakčními motory a při rozjezdu se postupně z obvodu vyřazovaly. Tento způsob řízení byl nevhodný, a proto se uvažovalo o napájení lokomotivy proměnlivým napětím. K dálkovému ovládní velikosti napětí v napájecí stanici sloužila pomocná trolej a pomocný sběrač umístěný na boku střechy vozidla. Tento způsob řízení byl aplikován v železniční stanici Valašské Meziříčí. Systém pomocné troleje se však nerozšířil. Až po rozvoji polovodičové techniky byly v průběhu sedmdesátých let odporové regulátory, popř. boční sběrače nahrazeny polovodičovými měniči pro napájení trakčních motorů. [6]



### 3.1 Stejnosemny mēnič napětí

Jak již bylo uvedeno, tento elektronický obvod přeměňuje stejnosměrnou energii. Zapojení obvodu tvoří dvojbran, na vstupu jsou dvě svorky, na Obr. 9 označené 1 a 2, které se navenek jeví jako spotřebič. Orientace vstupních elektrických veličin, tedy elektrického proudu a napětí je shodná. Proud teče od svorky 1 ke svorce 2, napětí má orientaci od svorky 1 ke svorce 2. Vstupní veličiny označíme  $U_1$  a  $I_1$ . Napětí  $U_1$  uvažujeme v čase neproměnné, proud  $I_1$  závisí na velikosti připojené zátěže na výstupní svorky dvojbranu. Výstupní svorky označené 3 a 4 se jeví jako svorky zdroje, neboť elektrické veličiny zde mají zdrojovou orientaci. Výstupní proud  $I_2$  pomyslně teče od svorky 4 ke svorce 3 skrz vnitřní obvod, kdežto výstupní napětí  $U_2$  má orientaci od svorky 3 ke svorce 4. [7], [4]



Obr. 9: Stejnosemny mēnič napětí jako dvojbran

Chování mēniče napětí je různé podle typu zátěže. Nejčastějšími typy zátěží je čistě odporová zátěž, zátěž R-L neboli sériové spojení rezistoru a cívky a aktivní zátěž.

#### 3.1.1 Stejnosemny mēnič napětí s odporovou zátěží

Nejjednodušším typem zátěže je rezistor. Dvojbran mēniče lze nahradit vodičem propojujícím svorky 2 a 4 a spínačem  $V$  umístěným mezi svorky 1 a 3. Spínač nahrazuje funkci polovodičového prvku. Při demonstraci funkce mēniče toto provést lze. Pokud bychom sledovali mēnič z hlediska účinnosti, nebo rychlosti spínání, museli bychom počítat s použitým polovodičovým prvkem, který má konkrétní fyzikální a elektrické vlastnosti. O používaných typech polovodičových spínačů se zmíním v samostatném odstavci. Mezi výstupní svorky dvojbranu vložíme rezistor s hodnotou odporu  $R$ . Stav mēniče bude záviset na poloze kontaktu - buď bude spínač sepnut (stav 1), nebo rozepnut (stav 2). V rozepnutém stavu obvodem neteče proud, na vstupních svorkách dvojbranu je napětí  $U_1$ , na výstupních svorkách je napětí  $U_2 = 0$ , protože rezistorem neprotéká proud a není na něm úbytek napětí.

Po sepnutí spínače začne obvodem téct proud, který teče přes rezistor. Protékajícím proudem se na něm vytvoří napětí, jehož velikost bude stejná jako výstupní napětí  $U_2$ . Hodnoty elektrických veličin považujeme za stálé, v čase neproměnné a můžeme je značit velkými písmeny. Při sepnutém spínači platí rovnice

$$U_2 = U_1 = R \cdot I_1 = R \cdot I_2 \quad (V; V; \Omega, A; \Omega, A) \quad (12)$$

Celý obvod z hlediska ekvivalentnosti lze nahradit zdrojem napětí  $U_1$  a ekvivalentním rezistorem  $R_e$ , jehož velikost je dána velikostí skutečného rezistoru a poměrnou dobou zapnutí  $z$  podle vztahu

$$R_e = \frac{R}{z} \quad (\Omega; \Omega, -) \quad (13)$$

Změnou zatěžovatele lze měnit  $R_e$  od velikosti hodnoty rezistoru  $R$  až po nekonečně veliký odpor při  $z=0$ .

### 3.1.2 Stejnoseměrný měnič napětí s R-L zátěží

Když k odporové zátěži přidáme do série indukční zátěž, vznikne R-L zátěž. Indukčnost je vlastnost cívek, které považujeme za akumulační obvodový prvek. [7]

Při rozpínání obvodu, ve kterém se nachází cívka, se na jejích vývodech indukuje napětí  $u$ . Velikost napětí je přímo úměrná rychlosti změny proudu  $i$  v čase  $t$ . Je-li indukčnost cívky  $L$ , můžeme velikost indukovaného napětí určit ze vztahu

$$u = L \cdot \frac{\Delta i}{\Delta t} (V; H, A, s) \quad (14)$$

Velikost indukovaného napětí může být velká i při malých indukčnostech. [7]

Indukované napětí se projeví jako přepětí na kontaktech spínače  $V$ , popř. na polovodičové součástce. Proto musíme zabránit vzniku přepětí. Toho se zbavíme jen tehdy, když zamezíme rychlé změně protékajícího proudu cívkou. Po rozepnutí kontaktu spínače se musí proud uzavřít přes pomocný prvek v obvodu s cívkou, kterým je dioda zapojená v závěrném směru vůči napájecímu zdroji. Katoda je připojena k pólu  $+$  a anoda k pólu  $-$ . Při sepnutém spínači je napětí na diodě v závěrném směru a neprochází jí žádný proud. Po rozepnutí spínače dojde k přepólování svorek cívky tak, že se ze spotřebiče stane zdroj, na diodě se objeví napětí v propustném směru a začne skrz ni protékat proud procházející cívkou. Proud bude exponenciálně klesat k nule. Použitou součástku nazveme nulová dioda  $V_0$ . [5]

Spínač má opět dva stavy. V sepnutém stavu platí následující vztahy:

$$u_2 = U_1 = R \cdot i_2 + L \cdot \frac{di_2}{dt} (V; V; \Omega, A, H, A, s) \quad (15)$$

$$i_1 = i_2 (A; A) \quad (16)$$

$$u_{V_0} = -U_1 (V; V) \quad (17)$$

$$i_{V_0} = 0 (A) \quad (18)$$

Elektrické veličiny výstupní napětí  $u_2$ , vstupní  $i_1$  a výstupní proud  $i_2$  se již v čase mění, proto je značíme malými písmeny.  $u_{V_0}$  je napětí na nulové diodě a  $i_{V_0}$  je proud protékající tímto prvkem.

V ustáleném stavu budou hodnoty všech veličin konstantní. [5]

Při rozepnutí spínače platí následující vztahy. [5]

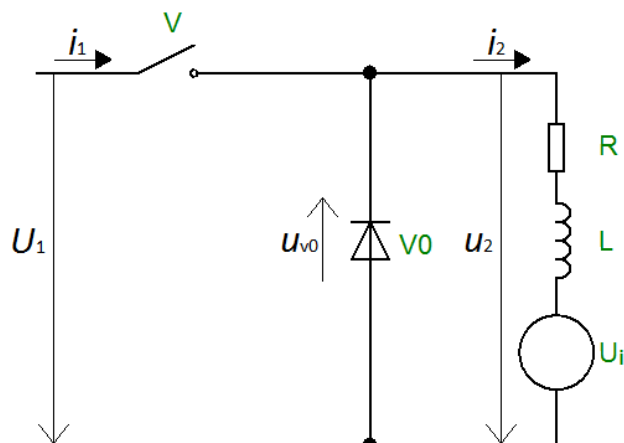
$$i_1 = 0 (A) \quad (19)$$

$$i_{V_0} = i_2 (A; A) \quad (20)$$

$$u_2 = R \cdot i_2 + L \cdot \frac{di_2}{dt} = 0 (V; \Omega, A, H, A, s) \quad (21)$$

### 3.1.3 Stejnosemřný měnič napětí s aktivní zátěží

Je-li zátěž tvořena stejnosmřným motorem, mluvíme o stejnosmřném měnič napětí s aktivní zátěží. Kromě spínače  $V$  se v obvodu nachází zátěž tvořená odporem  $R$ , indukčností  $L$  a zdrojem napětí  $U_i$ . Stejně jako v předešlém případě bude k zátěži paralelně připojena nulová dioda  $V_0$ . Vinutí kotvy tvoří cívku v sérii s rezistorem.[5]



Obr. 10: Obvodové schéma stejnosmřného měniče napětí s aktivní zátěží

Stejnosemřný měnič napětí s aktivní zátěží podle Obr. 10 má dva stavy: 1) spínač  $V$  sepnut, 2) spínač  $V$  rozepnut. Oba stavy si popíšeme.

1) spínač  $V$  je sepnut: Spínačem  $V$  teče proud  $i_1$ , který protéká zátěží, na zátěži je napětí zdroje  $U_i$ , na nulové diodě je napětí  $U_1$  v závěrném směru, neprotéká jí proud. Vše lze zapsat rovnicemi:

$$u_2 = R \cdot i_2 + L \cdot \frac{di_2}{dt} + U_i = U_1 \quad (V; \Omega, A, H, A, s, V; V) \quad (22)$$

$$i_1 = i_2 \quad (A; A) \quad (23)$$

$$u_{V_0} = -U_1 \quad (V; V) \quad (24)$$

$$i_{V_0} = 0 \quad (A) \quad (25)$$

2) spínač  $V$  je rozepnut: V okamžiku rozpojení spínače se odepne napětí  $U_1$ , na nulové diodě již nebude napětí v závěrném směru a diodou poteče proud zátěže  $i_2$ . [5]

$$u_2 = R \cdot i_2 + L \cdot \frac{di_2}{dt} + U_i = 0 \quad (V; \Omega, A, H, A, s, V) \quad (26)$$

$$i_1 = 0 \quad (A) \quad (27)$$

$$i_{V_0} = i_2 \quad (A; A) \quad (28)$$

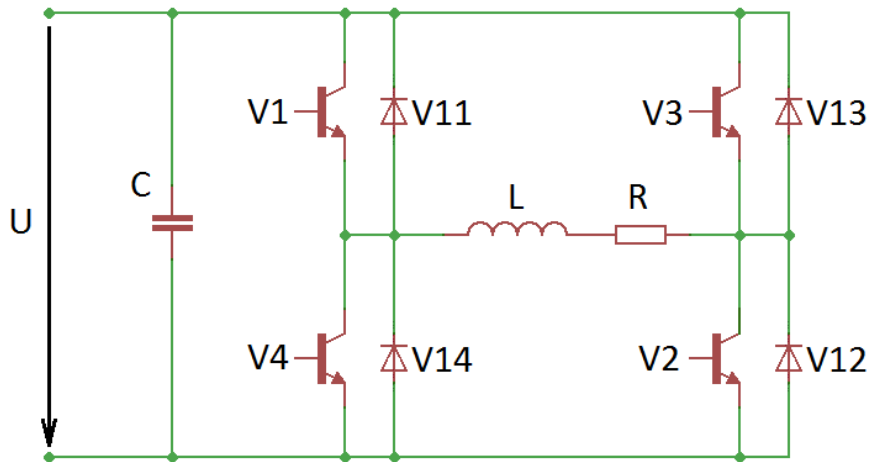
Proud  $i_2$  nezaniká exponenciálně k nule. Ustálí se na hodnotě  $-U_i/R$ . Jestliže je doba poklesu kratší než doba vypnutí  $t_{vyp}$  spínače, přestane proud procházet nulovou diodou, ta se vypne a na zátěži bude napětí  $U_i$ . V takovém případě hovoříme o přerušovaném proudu.

### 3.1.4 Použití stejnosmřného měniče napětí

Stejnosemřný měnič napětí lze použít pro regulaci elektrického pohonu se stejnosmřným motorem. Pokud budeme požadovat regulaci otáček pro oba směry, musíme zajistit, aby na zátěži bylo napětí opačné polarity a proud tek l druhým směrem, než proud  $i_2$  v obrázku 10. Potřebovali bychom tedy použitý měnič napětí odpojit a připojit druhý měnič napětí, který by toto zajistil. Vhodnějším způsobem pro řízení otáček stejnosmřného motoru změnou polarity napětí na kotvě je použití napěťového střídače, o kterém pojednává další podkapitola.

### 3.2 Obousměrné řízení otáček DC motorku

V provozu je velmi často požadováno, aby byl pohon nejen regulován, ale také aby bylo možné měnit směr otáček. Přitom soustrojí je napájeno konstantním stejnosměrným napětím  $U$ . Při použití polovodičových spínacích prvků lze vytvořit obvod tvořící jednofázový střídač. Jak z názvu vyplývá, jedná se o obvod, který mění některou elektrickou veličinu (proud nebo napětí) ze stejnosměrné hodnoty na střídavou. Zaměříme se nyní na napěťový střídač. Schéma jednofázového napěťového střídače je na Obr. 11. [5]



Obr. 11: Jednofázový napěťový střídač s RL zátěží

Napájení takového střídače zajistíme zdrojem konstantního napětí  $U$ , které se v ideálním případě nebude měnit se zatížením. Pro z odolnění zdroje napětí proti energetickým rázům přicházejícím od zátěže, připojíme na vstup napěťového střídače kondenzátor o velké kapacitě  $C$  paralelně k napájecím svorkám zdroje. K tomuto kondenzátoru připojíme dvě paralelní větve obsahující dva sériově spojené spínače. V první větvi budou za sebou spínače označené  $V1$  a  $V4$ , v druhé větvi pak  $V3$  a  $V2$ . Zátěž je připojena mezi uzly součástek  $V1$ ,  $V4$  a  $V3$ ,  $V2$ . Toto zapojení připomíná schéma jednofázového můstku, přičemž zatěžovací obvod, tvořený rezistorem  $R$  a cívkou  $L$ , tvoří diagonálu obvodu. Dále se ke každému spínači připojí antiparalelně dioda. Tyto diody, ve schématu označené  $V11$ ,  $V12$ ,  $V13$  a  $V14$  jsou v obvodu nezbytné, je-li v zatěžovacím obvodu indukčnost. V mém případě je zátěž tvořena vinutím rotoru DC motorku, které se chová jako RL zátěž. Kdybychom vypnuli proud tekoucí RL zátěží a v obvodu by se nenacházely tyto tzv. nulové diody, proud by se neměl kudy uzavírat a na indukčnosti  $L$  by se objevilo přepětí. Toto přepětí by bylo nebezpečné zejména pro spínače. Proud musí při vypnutí vypínače zkomutovat na jiný obvod, který tvoří právě nulové diody. Komutací rozumíme změnu smyslu proudu. Teče-li například proud  $i_z$  přes vypínače  $V1$  a  $V2$ , teče zároveň i přes zátěž. Po vypnutí obou spínačů, poteče proud  $i_z$  diodami  $V13$  a  $V14$ . Současně se na svorkách spotřebiče objeví napětí opačné polarity. [5]

Spínače bývají realizovány vypínatelnými tyristory, např. GTO<sup>4</sup>, nebo tranzistory. Pro podrobnější vysvětlení funkce nulových, či zpětných diod budeme uvažovat použití tranzistorů. Přivedením zapínacího pulzu na dobu, kdy má být tranzistor sepnut, bude jím protékat proud. Sepneme-li spínač *V1*, bude protékán proudem  $i_z$ . Vypnutím této součástky proud  $i_z$  zkomutuje na diodu *V14*. Na zátěži se tak objeví opačná polarita napětí. Bude-li smysl proudu opačný a zajistíme sepnutí spínače *V1*, proud poteče diodou antiparalelně připojenou k tomuto spínači, v našem případě se jedná o diodu *V11*. Když nyní sepneme vypínač *V4*, proud zátěže, tekoucí diodou *V11* zkomutuje na tento spínač. Toto platí stejně i pro ostatní spínače a příslušné diody. Z toho lze usoudit, že: „Svorka spotřebiče je bez ohledu na smysl proudu připojena vždy k té polaritě zdroje k níž je připojen spínač, mající na svém hradlu spínací impuls. Vyslovený závěr je platný bez ohledu na to, jestli proud teče dotyčným spínačem nebo jeho antiparalelní diodou. V případě, že u vstupní svorky spotřebiče nemá žádný z obou spínačů řídicí impuls je tato svorka v případě kladného smyslu proudu připojena na zápornou polaritu napájecího napětí a v případě záporného proudu připojena na kladnou polaritu napětí. Obrácené platí pro výstupní svorku.“ [5]

Napětíové jednofázové střídače sice slouží pro přeměnu stejnosměrného napětí na střídavé, lze je však použít i pro řízení DC motorků. Hlavní rozdíl je v tom, že v případě přeměny stejnosměrného napětí na střídavé postupně spínáme všechny čtyři spínače. Při řízení motorku spínáme pouze dva spínače pro každý směr. Stav spínačů napětíového jednofázového střídače pro řízení směru otáček motorku ukazuje následující Tab. 1. [5]

Tab. 1: Stav spínačů a vliv na obvod

Stav zátěže s ohledem na stav spínačů				
	V1	V2	V3	V4
otáčky směr 1	sepnut	sepnut	vypnut	vypnut
otáčky směr 2	vypnut	vypnut	sepnut	sepnut
brzdění	vypnut	sepnut	vypnut	sepnut
samovolné dotočení	vypnut	vypnut	vypnut	vypnut

Budeme-li požadovat regulaci otáček, použijeme periodické spínání příslušných spínačů. Pro podrobnější popis principu řízení otáček pomocí napětíového jednofázového střídače zvolíme otáčení hřídele motorku *směrem 1*. V tomto případě budeme spínat spínače *V1* a *V2*.

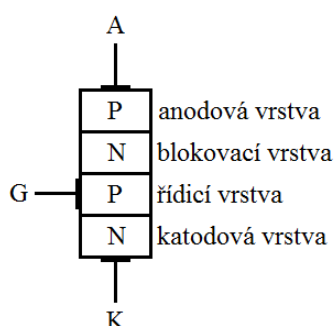
První možností je sepnout jeden spínač trvale a druhý spínač periodicky spínat a rozpínat. Zde se jeví jako vhodnější použít ke spínání obvodu tranzistor, protože ten ovládáme jediným řídicím signálem. Řídicí signál je přiveden na tranzistor po celou dobu, kdy má být tranzistor sepnut. Oproti tomu vypínatelný tyristor potřebuje pro zapnutí řídicí pulz o jedné polaritě napětí a vypínací pulz opačné polaritě. Při řízení otáček motorku spínáním jednoho tranzistoru bude potřebný program pro řízení tranzistoru jednodušší, než v případě druhé možnosti, kdy by byly řízené oba spínače. V mém projektu jsem nepoužil napětíový střídač, ale obvod tvořený budičem **L293B** a zpětnými diodami. Ovšem toto zapojení má naprosto stejnou funkci, jako napětíový jednofázový střídač pro řízení otáček motorku jedním směrem. [5]

<sup>4</sup> GTO = Gate Turn Off Thyristor, vypínatelný tyristor

### 3.3 Elektronické spínače

Elektronické součástky, které se používají ke spínání elektrických obvodů, dělíme podle jejich vlastností do tří skupin.

V první skupině jsou součástky, u kterých nemůžeme vnějším zásahem změnit jejich stav. Nazýváme je neřízené elektronické součástky a patří sem dioda. Dioda obsahuje jeden PN<sup>5</sup> přechod. Její hlavní vlastností je usměrňovací schopnost. V závislosti na přiloženém napětí na její vývody umožňuje protékání proudu skrz PN přechod. V propustném směru, kdy je na anodě kladný potenciál a na katodě záporný, má VA<sup>6</sup> charakteristika exponenciální průběh. V závěrném směru propouští PN přechod jen zanedbatelně malý proud (u křemíkové diody v řádu desítek nA). Dojde-li k překročení maximální velikosti napětí v závěrném směru, PN přechod se zničí, dojde k jeho průrazu, proto se v aplikacích musí používat diody, které jsou konstruovány na požadované velikosti napětí. Diody s PN přechody jsou užívány pro nízké frekvence, nejčastěji do 400 Hz. Potřebujeme-li spínat při vyšších kmitočtech, lze užít tzv. Schottkyho diody. O těchto diodách se ještě zmíním, protože jsem je ve své práci použil a bylo by dobré vysvětlit, proč jsem se rozhodl použít právě tyto diody. [5], [3]



Obr. 12: Struktura tyristoru

Do druhé skupiny elektronických součástek používaných ve výkonové elektronice řadíme polořízené součástky. Můžeme je vnějším zásahem zapínat, proto je označujeme jako součástky s řízeným zapínáním. Řídicím impulzem ji můžeme zapínat, ale už ji nelze řídicím impulzem vypnout. Do této skupiny patří triodové zpětně závěrné tyristory, častěji označované jako tyristory. Jsou to čtyřvrstvé diody doplněné o jednu elektrodu, takže jim říkáme čtyřvrstvé triody. Dvě vrstvy tvoří polovodič typu P a dvě vrstvy typu N vzájemně proložené, takže vytvářejí tři PN přechody. Strukturu tyristoru si lze představit jako dva spojené tranzistory, jeden typu PNP a druhý typu NPN tak, že NP přechod prvního tranzistoru je zároveň NP přechodem druhého tranzistoru. Situaci pro lepší představu dokumentuje Obr. 12. Oproti diodě má tato součástka tři vývody. První dva slouží ke stejnému účelu jako diody, nazýváme je proto stejně - anoda a katoda a třetí vývod slouží k přivedení zapínacího impulsu. Řídicí elektrodu nazýváme hradlo a značíme G, z anglického slova Gate. Zapínací impuls vyšleme na součástku, když je na anodě kladný potenciál a na katodě záporný potenciál. Zapínací impuls je proudového charakteru a proud teče z hradla do katody. Zároveň začne protékat proud tyristorem od anody ke katodě a součástka tak začne vést elektrický proud. Je sepnutá. Pro její vypnutí musí dojít k poklesu protékajícího proudu na nulu. Poté se obnoví závěrně polarizované přechody a součástka je připravena k dalšímu sepnutí proudovým impulzem. Proudový impuls musí být patřičně velký a mít dostatečnou strmost proudu  $di/dt$ . Tyristorů je celá řada, patří sem rychlé tyristory, frekvenční tyristory, asymetrické tyristory, fototyristory atd. U tyristorů používaných ve výkonové elektronice dosahuje nejvyšší okamžitá hodnota jakéhokoliv opakovatelného přechodného závěrného napětí označeného  $U_{RRM}$  řádu jednotek kV.

<sup>5</sup> PN – přechod dvou polovodičů, jeden má nadbytek děr (typ P) a druhý nadbytek elektronů (typ N). [3]

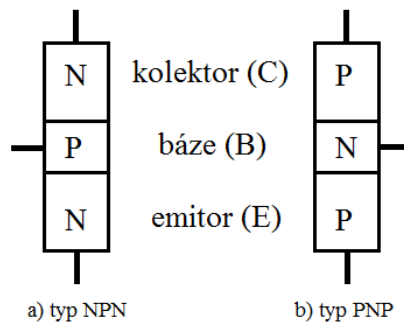
<sup>6</sup> VA charakteristika neboli volt ampérová charakteristika

Například podle katalogového listu součástky označené **TV 989-2700-28** výrobce Polovodiče a.s. dosahuje stálá hodnota procházejícího proudu  $I_{TRMS}$  4 193 A při teplotě pouzdra součástky 70°C, teplotě přechodu 125°C a frekvenci 50 Hz. Uvedený tyristor má pastilkové pouzdro.[5], [3], [8]

Třetí skupinou součástek jsou plně řízené polovodičové prvky. Sem patří tranzistory, vypínatelné tyristory, IGBT<sup>7</sup> a IGCT<sup>8</sup>. Vypínatelné součástky mají tři elektrody. Jejich označení záleží na typu součástky. Protože se tyto součástky hojně využívají v dnešních elektronických výkonových aplikacích, rozebereme si je podrobněji. Jelikož jsem se v předchozím odstavci zmínil o tyristorech, zmíním se nejprve o tyristorech, které lze zapínat i vypínat. Vypínací tyristory označené GTO z anglické zkratky slov Gate Turn Off Thyristor, tedy tyristor, na jehož elektrodu - hradlo se vysílá zapínací, ale i vypínací impuls. Průběh spínání je shodný se spínáním klasického tyristoru. Při potřebě vypnutí tyristoru bez poklesu proudu k nule, je na hradlo vyslán záporný impuls. U novějších typů vypínatelných tyristorů dosahuje hodnota průměrného propustného proudu  $I_{TAV}$  jednotek kA. Typové napětí v propustném směru, při kterém nepřejde tyristor do vodivého stavu označené  $U_{FD}$  dosahuje jednotek kV. S vypínacími GTO tyristory se setkáme především v trakci, kde se instalované výkony pohybují v řádu stovek kW. Pouzdro těchto součástek je ve tvaru pastilky, odvod tepla probíhá přes čelní kruhové plochy pouzder, na které těsně doléhá tělo chladiče.[5], [3]

Dalším typem vypínatelných tyristorů je vypínací tyristor IGCT. Zkratka je opět vytvořena z anglického názvu součástky: Integrated Gate-Commutated Thyristor. Jelikož u vypínatelných součástek dochází k velkým výkonovým ztrátám v průběhu zapínání a pak především během vypínání, bylo v průběhu vývoje výkonových vypínatelných součástek cílem snížení těchto ztrát zkrácením doby vypnutí. IGCT tyristor má hradlo uzpůsobené pro rychlé spínání, navíc k tyristoru není třeba připojovat ochranné obvody, které jsou u GTO tyristorů nutné. Vypínací tyristory se vyrábějí pro chlazení kapalinou i vzduchem. Při užití druhé varianty se musí zákazník spokojit s většími rozměry součástky. [3]

Mezi plně řízené součástky patří tranzistory. Jsou to aktivní součástky se třemi vývody, které dělíme do dvou skupin. Prvním typem je bipolární tranzistor (Obr. 13). Jedná se o součástku se dvěma PN přechody. Ty jsou vytvořeny umístěním polovodiče jednoho typu mezi dva polovodiče druhého typu. Elektrody součástky se označují písmeny E (emitor), C (kolektor) a B (báze). Přechody označujeme, podle jejich umístění, kolektorový a emitorový. Bipolární tranzistor se označuje také zkratkou BJT<sup>9</sup>. Podle typu přechodů dělíme bipolární tranzistory na typy NPN a PNP. U tranzistoru NPN je báze tvořena tenkou vrstvou polovodiče typu P. Napětí  $U_{CE}$  polarizuje přechod mezi kolektorem a bází závěrně, kdežto druhý přechod je polarizován v propustném směru. Tenká vrstva P polovodiče umožňuje vzájemné ovlivňování se obou přechodů. Přivedeme-li napětí  $U_{BE}$  mezi bází a emitor tak, že kladný potenciál bude na bází a záporný na emitoru, dojde k pronikání elektronů z emitorové oblasti do polovodiče typu P. Část z nich přejde na elektrodu báze, ale většina z nich je interakcí vtažena do kolektorové oblasti a proud se uzavře přes smyčku se zdrojem napětí  $U_{CE}$ . Velikost proudu tekoucího touto smyčkou se reguluje velikostí bázevého proudu  $I_B$  ze zdroje  $U_{BE}$ . Tranzistor PNP funguje obdobně. [3]



Obr. 13: Struktura bipolárního tranzistoru

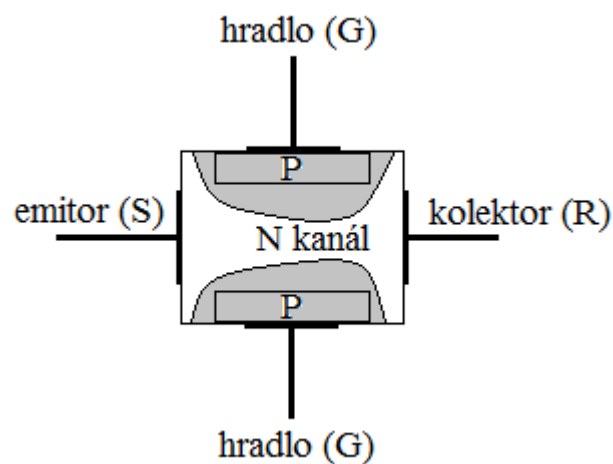
<sup>7</sup> IGBT = Insulated Gate Bipolar Transistor, bipolární tranzistor s izolovaným hradlem. [3]

<sup>8</sup> IGCT – označení pro GCT (Gate Commutated Thyristor, vypínatelný tyristor), který je dodáván s řídicí jednotkou. [5]

<sup>9</sup> BJT = Bipolar Junction Transistor, označení pro bipolární tranzistory. [3]

Druhým typem tranzistoru je unipolární tranzistor označený FET<sup>10</sup>. U tohoto typu tranzistoru je průchod většinových nábojů řízen velikostí elektrického pole, které je vytvořeno napětím přiloženým na elektrodu označenou G. Tímto písmenem značíme hradlo, neboli gate tvořící řídicí elektrodu. Hradlo je izolováno od polovodičové destičky vytvářející kanál. Kanálem se pohybují od emitoru označeného S (z anglického Source, česky zdroj) ke kolektoru D (počáteční písmeno anglického slova Drain). Struktura popsaného typu tranzistoru je stručně zobrazena Obr. 14. Velikost procházejícího elektrického náboje skrz kanál je ovlivněna napětím přiloženým na hradlo a také velikostí vnitřního odporu kanálu. Ve výkonové elektronice se užívá tranzistorů značených MOSFET<sup>11</sup>, které mají izolované hradlo. Běžně se u nich dosahuje hodnoty napětí 500 V mezi kolektorem a emitorem  $U_{DS}$  a proudem  $I_D=50 A$ .

Své místo v nejrůznějších aplikacích nacházejí tranzistorové moduly IGBT s teplotou přechodu  $T_{jmax}=150^{\circ}C$ . Užívají se ve výkonové elektronice pro stejnosměrné a střídavé pohony, svářečky, rezonanční a spínané napájecí zdroje. Hodnoty několika jednotek kV dosahuje napětí mezi kolektorem a emitorem  $U_{CE}$  vysoce výkonných IGBT tranzistorů s kolektorovým proudem přes 1 kA. [3], [5], [9]



Obr. 14: Struktura unipolárního tranzistoru

<sup>10</sup> FET = Field Effect Transistor, označení pro polem řízené tranzistory. [3]

<sup>11</sup> MOSFET = Metal-Oxid-Semiconductor FET, zkratka pro tranzistor FET s kovovým hradlem a izolantem z oxidu křemičitého. [3]



## 4 Vývojová deska, vývojové prostředí, programovací jazyk

Pro uskutečnění řízení DC motorku a naprogramovaného rozsvěcování a zhasínání LED bylo třeba vybrat vhodné hardware (HW) a software (SW). Vedoucí práce mě seznámil s hardware v podobě vývojové destičky od firmy Texas Instruments a prostředím, ve kterém lze vytvářet příslušný software. Na následujících řádcích popíšu obě části nezbytně nutné pro úspěšnou realizaci praktické části bakalářské práce.

### 4.1 Vývojová deska od firmy Texas Instruments

Název této vývojové desky (, zobrazené na Obr. 1,) je C2000<sup>TM</sup> Piccolo<sup>TM</sup> LaunchPad. Osazena je programovatelným mikroprocesorem **TMS320F28027PTT**. Každý výrobce se v zájmu přehlednosti a informovanosti snaží své výrobky pojmenovat tak, aby z jejich nejčastěji písmeno-číslíkového označení šlo získat základní technické parametry výrobku. Ne jinak je tomu u mikroprocesoru firmy Texas Instruments. První tři písmena **TMS** značí, že se jedná o kvalifikované, odzkoušené zařízení (*TMS = qualified device*), číslo **320** označuje skupinu zařízení, další písmeno **F** informuje o použité technologii (*F = Flash*), následující pětičíslí **28027** je evidenční číslo zařízení, písmena **PT** se vztahují k pouzdru mikroprocesoru, zahrnují informaci o počtu vývodů, kterých je 48 a o typu pouzdra, které je nízkoprofilové se čtvercovým půdorysem, neboli *Low-Profile Quad Flatpack (LQFP)*. Posledním znakem je v mém případě písmeno **T**, které udává teplotní rozsah. Mikroprocesor lze provozovat při teplotě  $-40^{\circ}\text{C}$  až  $+105^{\circ}\text{C}$ . Poslední čtyři znaky však výrobce na vývojové desce neuvádí, číslice **7** je nahrazena písmenem **x** a zbylé znaky chybí úplně. Jsou však k nalezení na pouzdře mikroprocesoru, kde se nachází ještě několik dalších údajů. [10]

Kromě mikroprocesoru se na desce nacházejí pomocné obvody, napájecí obvod se stabilizací napětí na hodnotu 3,3 V, externí oscilátor, 4 modře svítící LED pro zobrazování probíhajících dějů, 2 tlačítka, jedno lze naprogramovat a použít v aplikaci, druhé s funkcí RESET<sup>12</sup>, posuvné přepínače *S1* a *S4* a celkem 43 propojovacích kolíků pro připojení vnějších obvodů k vývojové desce. Kromě zmiňovaných LED se na přípravku nacházejí pomocné obvody s LED indikujícími různé stavy:

- *D1*, zelená LED, informuje o napájení přípravku,
- *D6*, červená LED, informuje o napájení obvodu emulátoru,
- *D7* a *D8*, modré LED, informují o sériové komunikaci mezi počítačem a vývojovým přípravkem přes USB<sup>13</sup> kabel, nebo o komunikaci mezi mikroprocesorem a vnitřní pamětí.

Použitý mikroprocesor TMS320F28027 je 32 bitový mikrokontrolér s délkou instrukčního cyklu 16,67 ns ( $f = 60$  MHz). Mikrokontrolér má oddělené sběrnice pro čtení instrukcí a zapisování dat, protože je užito architektury typu Harvard. Slabiková organizace paměti je typu Little endian, slova se ukládají po slabikách tak, že jako první se uloží poslední slabika slova, teprve potom ta předchozí (první). Napájení je 3,3 V. Na čipu modulu se nachází teplotní čidlo umožňující sledovat a měřit teplotu čipu. V mikroprocesoru se nacházejí 3 časovače s frekvencemi 60 MHz, 50MHz a 40 MHz, 4 jednotky generující PWM signál, porovnávací jednotku - komparátor, analogově digitální převodník (ADC), záchytnou jednotku, 3 externí přerušeni 20 digitálních vstupů/výstupů (GPIO<sup>14</sup>) a 6 analogových vstupů/výstupů (AIO<sup>15</sup>). [10]

<sup>12</sup> RESET – označení pro funkci, která „něco“ vrací do přednastaveného stavu

<sup>13</sup> USB = Universal Serial Bus, označení pro univerzální sériovou sběrnici a pro rozhraní v počítači

<sup>14</sup> GPIO = General Purpose Input Output, označení použité v kódu programu pro nespojitou komunikaci s okolním světem

<sup>15</sup> AIO = Analog Input Output, označení použité v kódu programu pro spojitou komunikaci s okolním světem

Blokové schéma použitého mikroprocesoru je znázorněno v příloze č. 1. [10]  
Napájení vývojové desky napětím 5 V je realizováno přes USB kabel z PC.  
Napájecí napětí lze přivést na vývojovou desku také prostřednictvím propojů ze vnějšího obvodu.  
Propojky *JP1*, *JP2* a *JP3* slouží ke vzájemnému rozpojení některých částí vývojové desky.

## 4.2 Vývojové prostředí

K programování mikroprocesoru je zapotřebí programu *Code Composer Studio* a *controlSUITE*. Po instalaci se připojí přípravek *C2000* a automaticky dojde k nainstalování software mikroprocesoru.

Program lze psát jazykem C nebo C++. [11]

Vývojové prostředí *Code Composer Studio* umožňuje vytvářet program, který se následně přeloží a nahraje do paměti mikroprocesoru. Psaní programu je poměrně jednoduché, programové prostředí při psaní příkazu nabízí možné varianty, různé podbarvení znaků rozčleňuje strukturu a prostředí automaticky zarovnává text, aby byl napsaný kód přehledný.

Po napsání programu je třeba program přeložit. V případě, že je program napsán správně, je přeložen a převeden na soubor obsahující instrukce v podobě jedniček a nul, který je nahrán do paměti mikroprocesoru. Vyskytuje-li se v programu po jeho napsání chyba, nedojde k přeložení programu a uživatel je informován o stavu. Podle typu chyby je upozorněn na vzniklou situaci buďto přímo komentářem s konkrétní chybou, nebo pouze upozorněním, že je v programu chyba. Příkladem konkrétní chyby může být chybějící znaménko ";", kdy se projekt sice vytvoří po kliknutí na ikonu 7 dle Obr. 15, ale nedojde k přeložení. V případě, že je mikroprocesor připojen a kliknutím na ikonu 8 se projekt vytvoří, ale před spuštěním se objeví okno, které informuje uživatele o existenci chyby v projektu a nabídne možnost zrušení akce, nebo pokračování ve spuštění. Program by tedy bylo možné spustit i s chybou, ale mohlo by to mít následky. Proto je na místě zabránit spuštění programu, chybu vyhledat a odstranit. Vedle čísla řádku, ve kterém se chyba nachází, se objeví znak označující chybu v podobě bílého křížku v červeném kolečku a po najetí myši na tuto ikonku se zobrazí komentář *expected a ";"*. V některých případech se při překládání chyba sice odhalí, ale nezobrazí se stejný symbol jako v předchozím příkladu, ale symbol černého vykřičníku ve žlutém trojúhelníčku. Pokud zapomene uživatel napsat do kulatých závorek podmínky při porovnávání znaků dvojité rovnítko, objeví se po vytvoření projektu vedle řádku s chybou tento znak a komentář obsahuje informaci, že by zde měl být znak "==" namísto "=". Program tedy mnoho chyb při překládání odhalí a navrhne vhodnou úpravu.

Špatná struktura příkazového řádku se většinou projeví ihned po jeho napsání změnou podbarvení. Zde jako příklad uvedu nastavení pinu *J6/3* na funkci GPIO, tedy umožním, aby se prostřednictvím tohoto pinu komunikovalo s vnějším světem. Napíšu příkaz

```
GpioCtrlRegs.GPAMUX1.bit.GPIO2 = 0;
```

a vynechám některé písmenko. Ihned se modrá část řádku přebarví na černou. Pokud si tohoto uživatel nevšimne, zobrazí se mu po vytvoření projektu symbol červeného kolečka s bílým křížkem a komentář jej upozorní na chybnou strukturu. V případě, že chyba není syntaktická, nýbrž programátorská, může a nemusí být při překládání objevena. Pokud není objevena, musí na ni uživatel přijít sám. S tímto typem chyby jsem se také setkal.

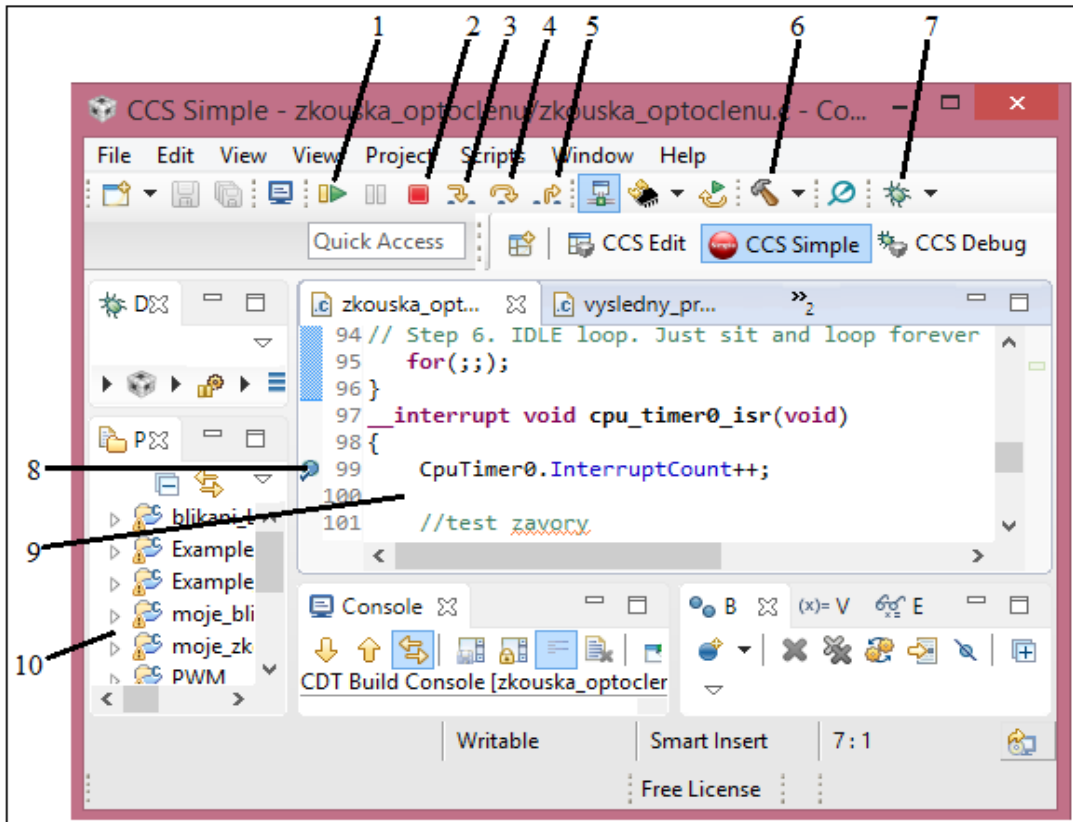
Dalším typem chyb jsou chyby, které nebrání spuštění programu, ale neumožňují jeho plnou funkci. Při tomto druhu chyb se vedle čísla řádku objeví jiný, než výše popsany symbol. Modrý vykřičník se zobrazil poté, co jsem v podmínce porovnával dvě proměnné různých datových typů, přičemž jeden byl znaménkový a druhý neznaménkový. Žlutý trojúhelníček se může objevit například poté, co se překročí rozsah datového typu při definování proměnné.

Výhodou programu *Code Composer* je možnost kontroly správné činnosti programu krokováním, kdy se program nahraje do mikroprocesoru, ve vývojovém prostředí se vytvoří značky tzv. breakpointy. Program se poté spustí, ale skončí v místě prvního breakpointu nacházejícího se v „cestě“ sledu vykonávaných instrukcí.

Pomocí kláves *F5*, *F6* a *F7* nebo kliknutím na ikonky se šipkami v horní liště prostředí programu (na Obr. 15 označené čísla 4, 5 a 6) se programem postupuje k dalším příkazům.

Kdykoli během tohoto krokování je možné nahlédnout do jakékoliv proměnné nacházející se v programu. Můžeme tak zjistit, jestli daná proměnná nenabývá hodnot, které jsou mimo námi požadovaný rozsah, nebo zdali se program k takovýmto hodnotám vůbec dokáže dostat. Dá se tedy snadno vypátrat jakákoliv chyba bránící správnému chodu programu.

Hodnotu proměnných lze také změnit v režimu krokování, kdy program „dokrokuje“ k určitému breakpointu, zastaví se jeho chod, uživatel změní některou proměnnou, která má za následek například vykonání konkrétní podmínky a po spuštění programu symbolem označeným č. 1 v Obr. 15 se vykonají příkazy uvedené mezi složenými závorkami za podmínkou.



Obr. 15: Otevřená okna programu Code Composer. 1) spuštění programu, 2), 3) zastavení programu, 4), 5) a 6) ikonky pro krokování programu, 7), 8) tlačítka kontroly programu, následného přeložení a spuštění, 9) breakpoint, 10) okno editoru, kde se program píše, 11) okno vytvořených programů

Do programu však lze zasahovat mnohem výrazněji. Po přepnutí do režimu *CCS Debug* vpravo v horní liště a do okna *Registers*, neboli registry, je možné upravovat hodnoty registrů za chodu programu. Při své práci v tomto programu jsem často využil možnosti měnit velikost periody použitého časovače při spuštění programu pomocí registru *TIMER<sub>X</sub>PRDH* po rozkliknutí registru *CPUTIMER*, přičemž místo *X* je číslo 1, 2, nebo 3, podle zvoleného časovače. Změnu periody jsem následně pozoroval na změně rychlosti blikání modrých LED na destičce. Možné je též změnit logickou úroveň jakéhokoliv výstupního pinu a ověřit tak funkci některého vnějšího obvodu jako reakci na programovou instrukci. Při režimu krokování je však vhodnější vnější digitální výstupy nahradit uvnitř kulatých závorek podmínkou pomocnou proměnnou, protože u té lze měnit její hodnotu přímo, bez použití registrů.

Celý program je koncipován tak, aby se jeho ovládání naučil každý uživatel velmi rychle a bez sáhodlouhého pročítání manuálu. Uživatel však musí znát vlastnosti a parametry mikroprocesoru, které získá v datovém listu mikroprocesoru.

V dalších manuálech se programátor dozví, jak kterou část programu napsat, objevují se zde konkrétní příklady a popis jejich funkcí, takže záleží především na programátorovi, podle kterého příkladu vytvoří vlastní program. Veškeré údaje o vývojové desce, manuály a datové listy daného mikroprocesoru jsou k nalezení na internetových stránkách výrobce. [12]

Ještě zmíním rozdílnost funkcí některých ikonek v horní liště programu. Ikonka s „kladívkem“ označená číslem 7 slouží k vytvoření programu (anglicky Build = vytvořit). Nemusí být přítom k počítači připojena vývojová deska. Pro vytvoření projektu však dojde k tzv. „odvšivení“ (z angl. Debug) programu, ale ne jeho nahrání do paměti mikroprocesoru. Ikonka se „švábem“<sup>16</sup> v obr. 13 pod číslem 7 plně nahrazuje funkci tlačítka 6 v případě propojení SW s HW (programového prostředí s mikroprocesorem). Projekt se vytvoří a nahraje do paměti mikroprocesoru, nenachází-li se v programu závažná chyba. Ikonky označené v obrázku čísly 2 a 3 slouží k pauze, respektive ukončení programu. Tlačítka 2 i 3 mají však hlubší smysl pouze v režimu krokování, kdy pozastavení programu použijeme tehdy, když se program z nějaké příčiny nedostane k dalšímu breakpointu a nepozastaví se tak sám. Pak lze použít zmíněné tlačítka a po pozastavení programu odstranit chybu bránící očekávanému sledu operací. Tlačítkem 3 se program ukončí. V případě, že je program spuštěn a vykonává nějaký cyklus, po kliknutí na ikonku 3 se program sice ukončí a ukončí se také komunikace s mikroprocesorem, ale nahraný program běží dále. Chceme-li program nahraný do mikroprocesoru ukončit bez resetování odpojením od napájení, klikneme na ikonku 2 a poté teprve na 3. Pro opětovné spuštění programu, musíme soubor do mikroprocesoru znovu nahrát kliknutím na ikonu 8. [13]

### 4.3 Programovací jazyk C

V předchozím odstavci jsem popsal vývojové prostředí, které používá programovací jazyk C, proto se krátce zmíním i o programovacím jazyce. Jazyk C patří k jazykům nižší úrovně, je univerzální, strukturovaný, obsahuje moderní datové struktury. Při zpracování programu se nejprve napíše programovacím jazykem programový kód v editoru, který vytvoří zdrojový soubor s koncovkou **.c**, následně se zdrojový soubor upraví v **preprocesoru**, který je součástí překladače - **compileru**. Compiler přeloží zdrojový soubor převzatý od preprocesoru do objektového kódu počítače. **Linker** sestaví program přidělením absolutních adres relativnímu kódu a výsledkem je program, který již lze spustit. Posléze vytvořený program zkontrolujeme pomocí ladícího programu - **debuggeru**, který má za úkol vyhledat chyby vyskytující se v programu. Výsledkem je hotový spustitelný program. [13]

---

<sup>16</sup> Slangové označení integrovaných obvodů v pouzdech s dvěma řadami vývodů

## 5 Popis vnějšího obvodu

Vnější obvod sestává ze dvou částí: dolní desky plošných spojů a horní desky plošných spojů (dále jen deska). Dolní deska (, na vyrobené desce označena jako „Spodní deska“,) slouží k napájení DC motorku, pomocných vnějších obvodů a vývojové desky C2000. Nacházejí se na ní dva stabilizátory napětí, elektrolytický kondenzátor, řídicí obvod pro DC motorek, který je k této desce přišroubován a senzor pro indikaci otáček. Na horní desce se nacházejí pomocné obvody pro napájení LED. Obě desky plošných spojů (DPS) jsou jednostranné, vodivý obrazec se vytváří na jedné straně opatřené měděnou fólií. Nosným materiálem je FR4<sup>17</sup>. DPS jsou osazeny součástkami typu THT<sup>18</sup>, což znamená, že vývody součástek procházejí otvory v desce a jsou připájeny na straně s vyleptaným obrazcem, součástky jsou na protější straně. Samotné LED diody jsou připevněny na plastové destičce přišroubované k horní desce.

### 5.1 Dolní deska

Rozpis součástek a prvků umístěných na dolní desce přináší následující tabulka. Dolní deska má rozměry 122 mm x 122 mm x 2 mm. Na horní straně desky se nacházejí součástky a stejnosměrný motorek, na protější straně jsou zapájeny vývody součástek a nachází se zde kruhové měděné plochy sloužící k přivedení elektrické energie do sestavy.

Tab. 2: Součástky umístěné na dolní desce

součástka	označení	počet
stabilizátor napětí	7806	1 ks
stabilizátor napětí	78S05	1 ks
budič DC motorku	L293B	1 ks
Schottkyho dioda	1N5819	4 ks
světelná závora	TCST1103	1 ks
rezistor	2R	2 ks
rezistor	280R	1 ks
kondenzátor	330 nF	2 ks
kondenzátor	100 nF	2 ks
patice	DIL16	2 ks
oboustranný kolík	ASS11030G	4 ks
propojka	JP2	1 ks
DC motorek	RF-500TB-18280	1 ks

<sup>17</sup> FR4 – označení pro materiál používaný pro výrobu desek plošných spojů, je to skelná tkanina pojená epoxidovou pryskyřicí. [31]

<sup>18</sup> THT = Through Hole Technology, vsazovaná montáž, technologie užitá při osazování DPS součástkami s vývody. [31]

## 5.2 Horní deska

Horní deska je oproti dolní desce menší, její rozměry jsou 65 mm x 64 mm x 2 mm. Součástky se nacházejí na spodní straně spolu s propojovacími kolíky, na horní straně, straně vývodů se nacházejí na pomocném plastovém držáku LED. Přehled součástek přináší Tab. 3

Tab. 3: Součástky umístěné na horní desce

součástka	označení	počet
Schmittův invertor	74AC14	2 ks
patice	DIL14	2 ks
dutinková lišta	BL820GD	2 ks
propojka	JP3	1 ks
světelná dioda	OSY5LW3134A	10 ks
rezistor	1K2	10 ks

## 5.3 Popis DC motorku

Motorek na stejnosměrné napájecí napětí s označením RF-500TB-18280 má ve satorové části permanentní magnety. Rozměry motorku jsou na Obr. 16. [14]

Technické údaje [14]:

rozsah napětí: 3,0 VDC - 6,0 VDC

jmenovité napětí: 6,0 VDC

otáčky naprázdno: 5150 ot·min<sup>-1</sup>

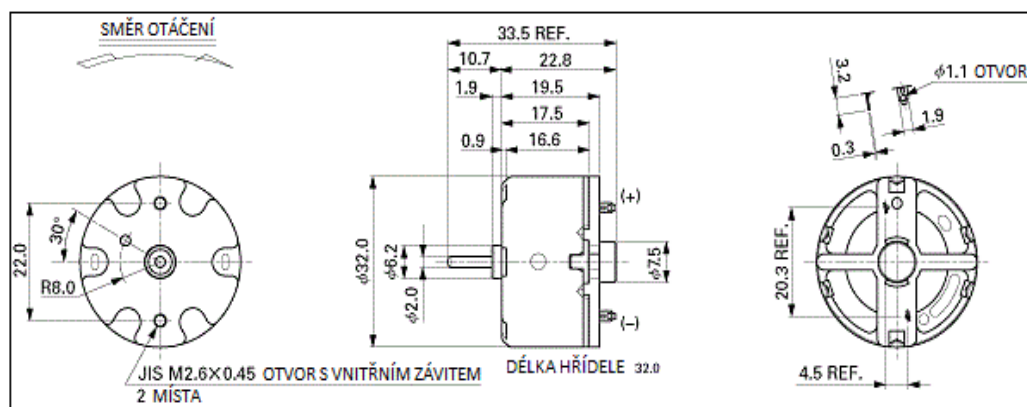
proud při otáčkách naprázdno: 0,05 A

jmenovité otáčky: 4300 ot·min<sup>-1</sup>

maximální proud při zatížení: 0,26 A

točivý moment: 2,06 mNm

výkon: 0,27 W



Obr. 16: Rozměry motorku

## 5.4 Popis součástek

### Stabilizátor napětí 7806

Jedná se o integrovaný obvod v pouzdru *TO220* se třemi vývody. Na vývody 1 a 2 je přivedeno napájecí napětí, které potřebujeme stabilizovat. Mezi vývody 3 a 2 je výstupní napětí, které je nižší než vstupní napětí a je stabilizované. Při zapojování je nutné brát v úvahu polaritu obou napětí. Prostřední vývod (č. 2) tvoří společnou zem, pól -, který se velmi často označuje GND<sup>19</sup>. Obrázek pouzdra zachycuje Obr. 17. Na tuto součástku lze nahlížet jako na dvojbran, který má uvnitř pouzdra spojenou jednu vstupní a jednu výstupní svorku. Tím je docíleno menších rozměrů celé součástky. Z označení vyplývají tyto technické údaje: první dvojčíslí 78 udává typ stabilizátoru - stabilizátor s pevným napětím, kladný (záporný pól je společný); druhé dvojčíslí udává hodnotu výstupního napětí ve voltech  $6\text{ V}$ . Mezi prvním a druhým dvojčíslím bývá také písmeno, udávající proudové zatížení stabilizátoru. Pokud písmeno chybí, lze ze stabilizátoru odebírat maximální proud

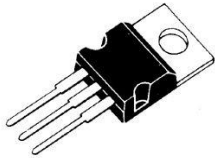
$$I_{max.} = 1\text{ A. [15]}$$

Pouzdro je na zadní straně opatřeno kovovou ploškou, skrze kterou se odvádí ze součástky vznikající teplo. Otvor v kovové plošce slouží k přišroubování hliníkového chladiče pro zvýšení odvodu tepla. V mé aplikaci jsem žádný chladič nepoužil, neboť součástka bude dostatečně chlazená okolním vzduchem.

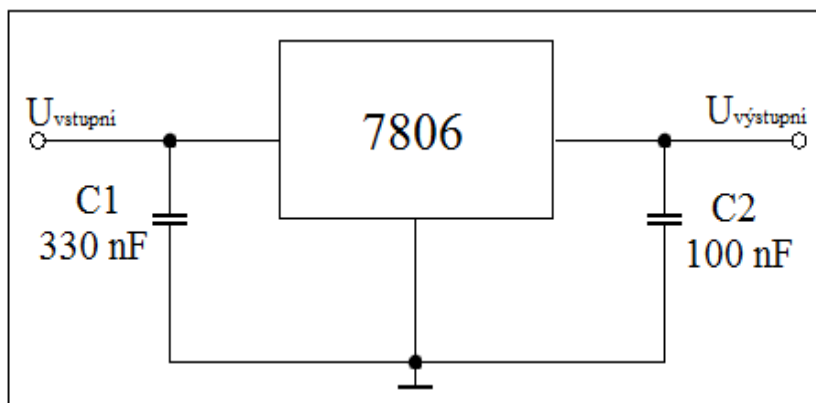
Základní technické údaje a obrázek pouzdra jsou uvedeny v Tab. 4.

Pro zajištění správné funkce stabilizátoru se v datovém listu součástky [15] doporučuje do obvodu připojit na vstupní i na výstupní stranu stabilizátoru kondenzátor. Ten musí být umístěn co nejbližší k vývodům součástky. Obvod stabilizátoru a kondenzátorů příslušných hodnot dokumentuje Obr. 18.

Tab. 4: Základní technické údaje a pouzdro součástky 7806 [15], [33]

Základní technické údaje			
technický údaj	zkratka	hodnota	
maximální vstupní napětí	$U_{vstup.}$	35 V	
výstupní napětí	$U_{výst.}$	6 V 1), 5 V 2)	
maximální výstupní proud	$I_{výst.}$	1 A 1), 2 A 2)	
provozní teplotní rozsah	$T_{ptr}$	-55°C až +150°C	
tepelný odpor mezi přechodem a základnou	$R_{thjc}$	5°C/W	
tepelný odpor mezi přechodem a okolím	$R_{thja}$	50°C/W	

1) pro 7806, 2) pro 78S05



Obr. 18: Zapojení stabilizátoru do obvodu s kondenzátory

<sup>19</sup> GND – označení pro svorku, která je propojená se záporným pólem napájecího zdroje

### Stabilizátor napětí 78S05

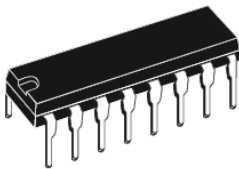
Integrovaný obvod v pouzdře *TO220* se stejnými vlastnostmi jako 7806. Pouze výstupní napětí má hodnotu  $5\text{ V}$  a proudové zatížení  $I_{max.} = 2\text{ A}$ . [15]

### Čtyřkanálový budič pro řízení motorku L293B

Tento integrovaný obvod v pouzdru *DIP16* je v mé práci užit pro řízení jednoho DC motorku pro jeden směr otáčení. Blokové schéma je znázorněno na Obr. 20. Obvod zesiluje PWM signál přivedený na vývody č. 2 a 7 z vývojové desky, přičemž na vývodu č. 2 je trvale  $0\text{ V}$ . Zesílený signál je prostřednictvím vývodů č. 3 a 6 přiveden na napájecí svorky DC motorku. Tento integrovaný obvod má dvě napájecí napětí. Jedno, označené v datovém listu jako  $V_s$ , slouží k silovému napájení motorku, druhé, označené jako  $V_{ss}$  slouží k napájení logických obvodů uvnitř součástky. Záporný pól obou napětí je společný, je nutné jej přivést na vývody č. 4, 5, 12 a 13. V mém případě je  $V_s = 6\text{ V}$  a  $V_{ss} = 5\text{ V}$ , což splňuje podmínku  $36\text{ V} \geq V_s \geq V_{ss} \geq 4,5\text{ V}$ . Do obvodu mezi tento integrovaný obvod a svorky motorku jsou připojeny 4 nulové diody. Princip řízení DC motorku integrovaným obvodem L293 a diod bude v samostatné kapitole připodobněn s obvodem napěťového střídače. Vybrané technické údaje a obrázek pouzdra obsahuje tabulka 5. [16]

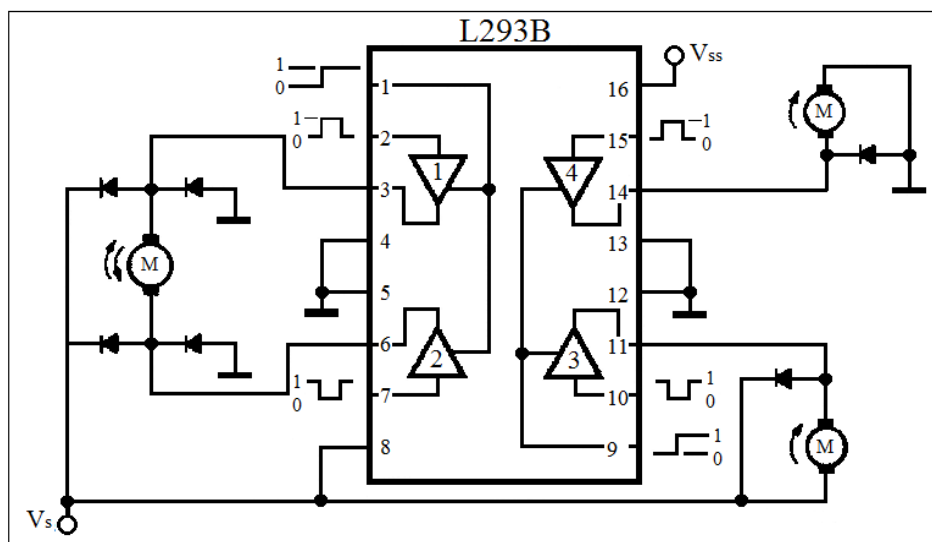
Tab. 5: Technické údaje a obrázek budiče L293B

Základní technické údaje		
technický údaj	zkratka	hodnota
maximální napájecí napětí	$V_s$	36 V
max. napájecí napětí logických obvodů	$V_{ss}$	36 V
max. výstupní proud 5ms impulz bez opakování	$I_{výst.}$	2 A
tepelný odpor mezi přechodem a základnou	$R_{thjc}$	$14^\circ\text{C/W}$
tepelný odpor mezi přechodem a okolím	$R_{thja}$	$80^\circ\text{C/W}$



Obr. 19: Pouzdro typu DIP16 [16]

Na vývodech součástky č. 4, 5, 12 a 13 je stejný potenciál (GND) a jsou umístěny tak, aby mohly být vzájemně propojeny přímo na desce plošných spojů a tato plocha mohla být co největší, protože přes tyto vývody se ze součástky odvádí teplo. Pokud tomu není uzpůsobeno rozmístění vodivých cest na DPS, je možné nasadit na pouzdro součástky speciální typ chladiče. V mé aplikaci není použit ani jeden výše uvedený typ chlazení. Opět postačuje chlazení součástky proudícím vzduchem, navíc proudové zatížení je velmi malé. [16]



Obr. 20: Blokové schéma L293B



### Světelná závora TCST1103

Původně jsem plánoval použít pro detekci otáček senzor s odrazem paprsků CNY70, který jsem zakoupil, zvolil vhodné zapojení a odzkoušel. Tento senzor funguje na podobném principu jako optron TCST1103 s tím rozdílem, že paprsek vysílaný IR<sup>20</sup> diodou se musí od něčeho odrazit a teprve po odrazu dopadá na bázi fototranzistoru. Počítal jsem s tím, že vyzařovaný paprsek se bude odrážet pouze od reflexních vrstev. Proto jsem odraznou plochu vytvořil z reflexní flexibilní odrazky hojně využívané jako bezpečnostní prvky pro cyklisty. Po umístění reflexního prvku vodorovně k hornímu povrchu součástky dopadalo záření vyzařované IR LED na fototranzistor i při vzdálenosti několika centimetrů obou předmětů od sebe. Později jsem však přišel na to, že ať umístím nad optočlen jakýkoliv předmět s rovným povrchem, bude se paprsek odrážet a dopadat na fototranzistor. Tudíž by docházelo k odražení paprsku od odrazného předmětu i od základny sestavy a otáčky by nebylo možné detekovat. Proto jsem se rozhodl přejít na vhodnější typ součástky pracující na podobném principu. Jako lépe vyhovující jsem vybral optron pro indikaci přerušení paprsku TCST1103. [17]

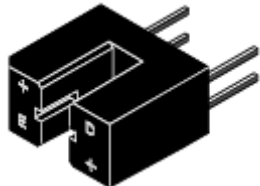
Tento dvojbran (viz. Obr. 21) funguje na principu přenášení 1bitové informace prostřednictvím optického záření. Zdrojem monochromatického záření o vlnové délce 950 nm je IR LED. Přijímač je tvořen fototranzistorem. Tato součástka se řadí mezi optoelektrické prvky. Použití optoelektrických prvků je v nejrůznějších senzorech. V zapnutém stavu emituje IR LED optické záření, které dopadá na fototranzistor v oblasti báze. Ten může být typu NPN i PNP. V aplikaci užitý senzor obsahuje fototranzistor typu NPN. Pohlcením fotonu v oblasti báze je generován pár elektron-díra, které jsou oddělovány kolektorovým a emitorovým přechodem. Tím se mění potenciál kolektorového a emitorového přechodu. [18]

Při přerušení paprsku nedopadá záření na fototranzistor, negenerují se nosiče elektrického proudu a obvodem neteče proud. Součástka je navíc vytvořena tak, aby na fototranzistor nedopadalo příliš mnoho světelného záření z okolí, poněvadž by mohlo negativně ovlivnit funkci optronu.

V tabulce 6 jsou základní technické údaje a obrázek součástky. [17]

Tab. 6: Technické údaje použitého optronu a obrázek pouzdra

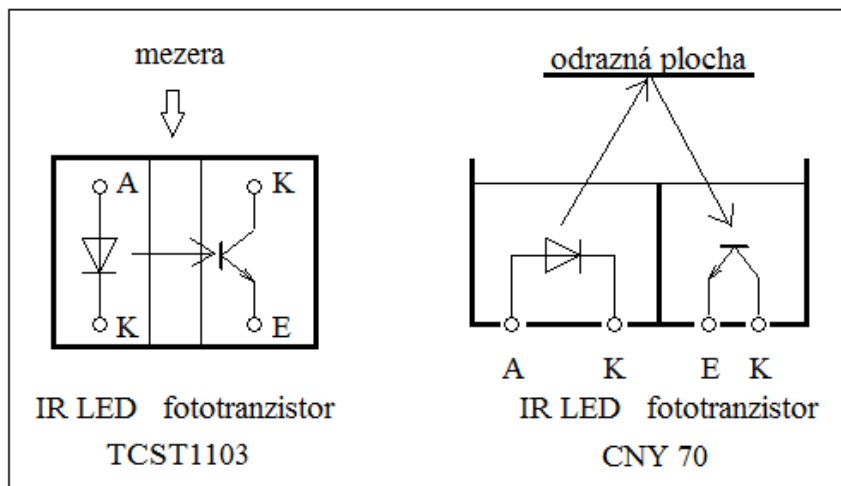
Základní technické údaje		
IR LED		
technický údaj	zkratka	hodnota
max. závěrné napětí	$U_{Rmax}$	6 V
max. propustný proud	$I_{Fmax}$	60 mA
fototranzistor		
technický údaj	zkratka	hodnota
max. napětí mezi kolektorem a emitorem	$U_{CEmax}$	70 V
max. napětí mezi emitorem a kolektorem	$U_{ECmax}$	7 V
max. kolektorový proud	$I_{Cmax}$	100 mA
vzdálenost mezi IR LED a fototranzistorem		7,6 mm



Obr. 21: Pouzdro součástky TCST1103

<sup>20</sup> IR LED = Infra Red LED, označení pro světelnou diodu, vyzařující elektromagnetické vlnění o vlnové délce odpovídající infračervenému spektru záření

Obr. 22 porovnáva principy zmiňovaných senzorů. [17], [19]



Obr. 22: Porovnání principu senzorů


### Schottkyho dioda 1N5819

Tato dioda má oproti klasické diodě s PN přechodem určité výhody, kterých jsem využil. Přechod Schottkyho diody je tvořen polovodičem a kovem. Proud tekoucí tímto přechodem je podmíněn majoritními (většinovými) nosiči náboje, kdežto u přechodu PN je dán minoritními (menšinovými) nosiči. Oproti křemíkovým diodám mají díky tomu schopnost propouštět větší proudy a také mají menší úbytek napětí v propustném směru. Úbytek napětí na přechodu je přibližně 3 V. Odpor substrátu  $N^{+21}$  a kapacita přechodu jsou velmi malé, proto má přechod malou časovou konstantu  $\tau = R \cdot C$ . Tím pádem jsou Schottkyho diody použitelné pro rychlé spínání. Používají se ve vysokofrekvenčních aplikacích. Také teplotní koeficient napětí na přechodu v propustném směru při konstantním proudu mají tyto diody menší než u křemíkových PN diod.

Princip přechodu kov-polovodič: Spojením kovu s výstupní prací  $A_k$  s polovodičem, který má výstupní práci  $A_p$ , dojde po jejich styku k vyrovnání náboje na přechodu. Elektronů přejdou z materiálu s menší výstupní prací do materiálu s vyšší výstupní prací. Bude-li výstupní práce polovodiče menší než výstupní práce kovu, vytvoří se v polovodiči oblast neobsahující volné elektrony. Proto v polovodiči vznikne vrstva s kladným nábojem a v kovu vznikne v těsné blízkosti přechodu tenká vrstva se záporným nábojem stejné velikosti. Tím vznikne elektrické pole zabírající přechodu dalších elektronů do kovu. [3]

Použitá součástka je v pouzdře *DO41*, základní technické údaje a obrázek součástky s označením vývodů obsahuje tabulka 7. [20], [21]

Tab. 7: Technické údaje a obrázek Schottkyho diody


Základní technické údaje			 <p>Obr. 23: Obrázek diody 1N5819</p>
technický údaj	zkratka	hodnota	
max. opakovatelné napětí, impulz	$U_{RRM}$	40 V	
maximální trvalé závěrné napětí	$U_{RMS}$	28 V	
max. neopakovatelné napětí, impulz	$U_{RSM}$	48 V	
max. propustný proud, střední hodnota	$I_{AV}$	1 A	
max. okamžité napětí v propustném směru	$U_F$	0,6 V	

<sup>21</sup>  $N^+$  - označení pro silně dotované polovodiče obsahující donory – dárce elektrického náboje. [3]

### 74AC14

Integrovaný obvod obsahuje 6 na sobě nezávislých invertorů Schmittova typu. Tato 14ti vývodová součástka v provedení *DIP* je použita pro snížení proudového zatížení výstupů na LED z vývojové destičky. Povolený proud procházející vstupními/výstupními piny vývojové destičky je 4 mA. Použitý typ LED by mohl výstupy přetížit. Proto je přivedený logický signál zesílen. Výstup tohoto integrovaného obvodu lze zatížit proudem 20 mA. Jelikož se jedná o invertor, výstupní stav je vždy opačný než vstupní. Základní technické údaje a obrázek pouzdra ukazují tab. 8. [22]

Tab. 8: Technické údaje a obrázek pouzdra 74AC14


Základní technické údaje			 Obr. 24: Pouzdro součástky 74AC14
technický údaj	zkratka	hodnota	
napájecí napětí	$V_{cc}$	0,5 V až 7 V	
vstupní napětí	$U_{vstup}$	0 V až $V_{cc}$	
výstupní napětí	$U_{výstup}$	0 V až $V_{cc}$	
proudové zatížení	$I$	20 mA	

### OSY5LW3134A

Světelná dioda o průměru kulatého pouzdra 3 mm s dominantní vlnovou délkou typově 590 nm odpovídající žluté barvě světla a typickou svítivostí 150 mcd je použita v počtu 10 kusů na vytvoření běžícího textu.

Tabulka 9 obsahuje nejzákladnější technické údaje a obrázek LED. [23]

Tab. 9: Technické údaje a obrázek LED

Základní technické údaje			 Obr. 25: Obrázek LED
technický údaj	zkratka	hodnota	
napájecí napětí	$U_f$	2,1 V	
propustný proud	$I_f$	30 mA	
typická svítivost*	$I_{\gamma\zeta}$	150 mcd	
úhel vyzařování	$\Omega_l$	60°	

\* typická svítivost při  $I_f = 20$  mA

Oboustranné kolíky a dutinkové lišty slouží k elektrickému i mechanickému propojení dolní a horní desky s vývojovým přípravkem. Součástky 74AC14 a L293B nejsou připájené k DPS, ale jsou umístěny v patičích. Tyto patice slouží ke snadné demontáži součástky z desky plošných spojů, například při hledání závad. Vývody patice kopírují rozmístění vývodů součástky a jsou zapájeny do desky plošných spojů. Stejně jako u součástek v pouzdech typu *DIP* je na patičích vylisovaná, nebo jinak provedená značka u vývodu č. 1, aby nedošlo k obrácení součástky.

## 6 Popis celé sestavy

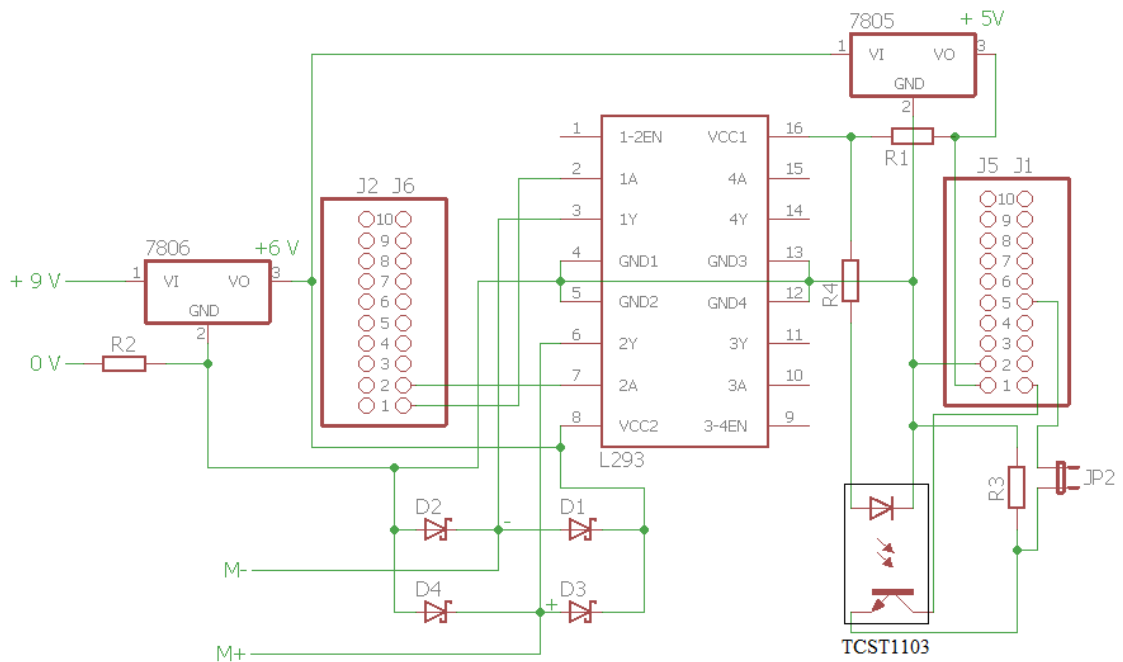
Sestava - praktická část mé bakalářské práce se skládá z nepohyblivé základny a rotující části. Nepohyblivá základna slouží k napájení rotující části a k její mechanické stabilizaci. Základna je tvořena plastovou čtyřdílnou krabičkou označení KP 3 B [24], ve které jsou umístěny tyto komponenty: napájecí konektor HEBL 21 [25], páčkový přepínač MTS 500APC [26], objímka pro 10mm LED SMB1149 [27] s oranžovou LED označení L-813ED [28] na přední stěně, držák pojistky PTF 5030 [29] s pojistkou na jmenovitý proud  $1,25 A$  na zadní stěně, hliníkový držák pro osu motorku a dvě hliníkové průchodky pro kartáčové vývody na horní stěně. Napájení je zajištěno externím zdrojem napětí  $9 VDC$ ;  $1,5 A$ . Díky stabilizátoru na dolní desce není problém napájet sestavu napětím i vyšším. Během testování funkce programu jsem užíval k napájení sestavy stejnosměrný napájecí zdroj o jmenovitém napětí  $12 V$ .

Rotující část je tvořena dolní deskou, vývojovou deskou s mikroprocesorem, horní deskou a plastovým dílem pro uchycení LED. Napájecí napětí je přivedeno na rotační část pomocí kartáčových vývodů umístěných na horní straně základny na soustředné kruhové plochy vytvořené na spodní straně dolní desky stejně jako vodivé cesty. Ve středu obou kružnic se nachází osa motorku, která je připevněna k základně prostřednictvím hliníkového držáku.

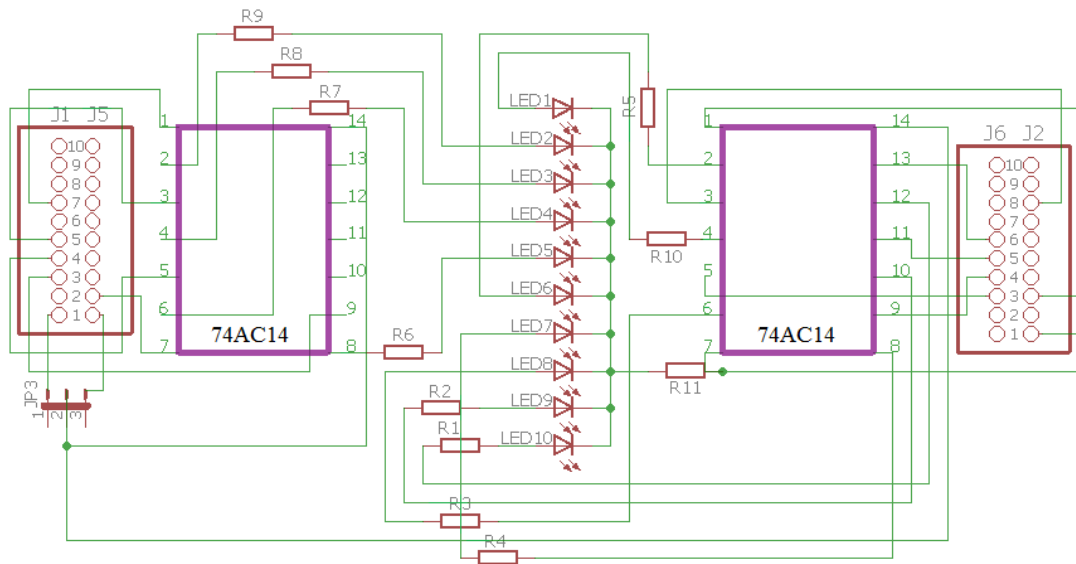
Bylo možné přistoupit k jednoduššímu způsobu provedení celého výtvaru tak, že by rotovaly pouze LED na destičce pevně spojené s osou motorku. V takovém případě by však bylo zapotřebí přivést na rotující část elektrické signály pro každou LED zvlášť, což by bylo více náročné. Zvýšil by se počet pólů sběračů a muselo by se zajistit mnohem kvalitnější elektrické propojení, aby nedocházelo ke ztrátám informací elektrického signálu. Dalším v úvahu přicházejícím řešením by bylo napájení LED malým akumulátorem umístěným na rotující destičce a řídicí signál by byl přiveden bezdrátově. Vyžadovalo by to vysílač na nepohyblivé části a přijímač na rotující části. Ovšem náročnost práce by se tím znásobila. Proto jsem přistoupil k jinému řešení, kdy potřebuji na rotující část přivést pouze napájecí napětí, tedy dva póly. Zároveň kvalita kontaktu kartáč- sběrač může být nižší. Užití k napájení motorku a celé sestavy akumulátor umístěný na pohyblivé části nepřichází v úvahu. Parametry akumulátoru s patřičnou kapacitou by negativně přispěly k rozměrům a hmotnosti rotující části. Výkon použitého motorku by tím pádem nemusel postačovat.

Napájecí napětí je přivedeno na stabilizátor napětí 7806, který snižuje napájecí napětí na hodnotu  $6 V$ . Tímto napětím je napájen budič L293B, světelná závora (IR LED), napětí  $6 V$  pól + je také spojen s dvěma Schottkyho diody a stabilizátorem napětí 78S05. Výstupní napětí z tohoto stabilizátoru je  $5 V$ . To slouží k napájení vývojové desky, logickému obvodu budiče L293B a přes propojovací kolíky na vývojové desce je připojeno na integrované obvody 74AC14 umístěné na horní desce. Zbylé dvě Schottkyho diody jsou připojeny na GND, tedy záporný pól napájení označený -. Druhý vývod těchto diod je spojen s napájením motorku stejně jako u prvních dvou diod. Každá Schottkyho dioda je tedy připojena k různým prvkům. Schottkyho diody jsem pro přehlednost zjednodušeně označil  $D1$  až  $D4$ . Jak vyplývá z obrázku 24 dolní desky, diody vytvářejí můstek. Anoda diody  $D1$  je spojena s výstupem stabilizátoru napětí 7806, tudíž je na napětí  $+ 6 V$ , katoda je propojena se záporným pólem motorku, stejně jako katoda diody  $D2$ . Anoda diody  $D2$  je spojena se pólem  $0 V$  (GND). Ke kladnému pólu motorku je připojena anoda diody  $D3$  a katoda diody  $D4$ . Katoda diody  $D3$  je propojena s anodou diody  $D1$ . Anoda diody  $D4$  je na napětí  $0 V$ .

Na následujících dvou obrázcích (Obr. 26 a Obr. 27) je znázorněné schéma elektrického obvodu dolní a horní desky.

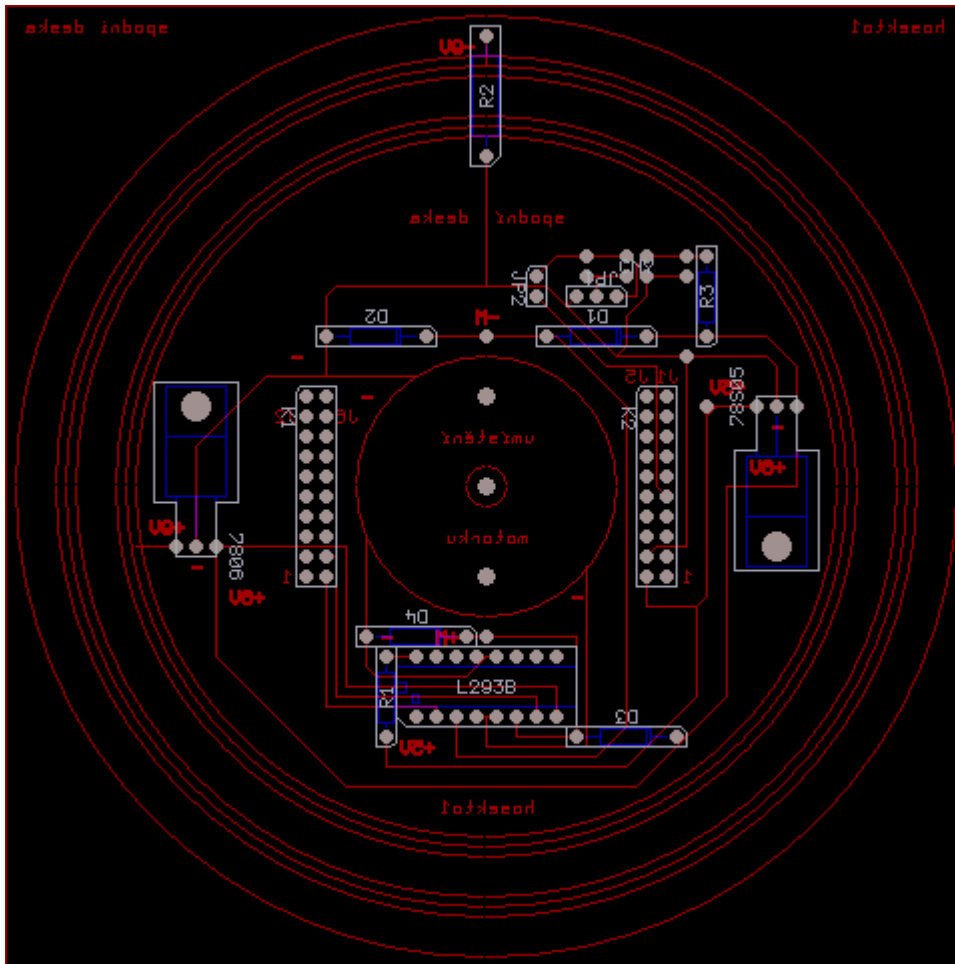


Obr. 26: Schéma zapojení dolní desky

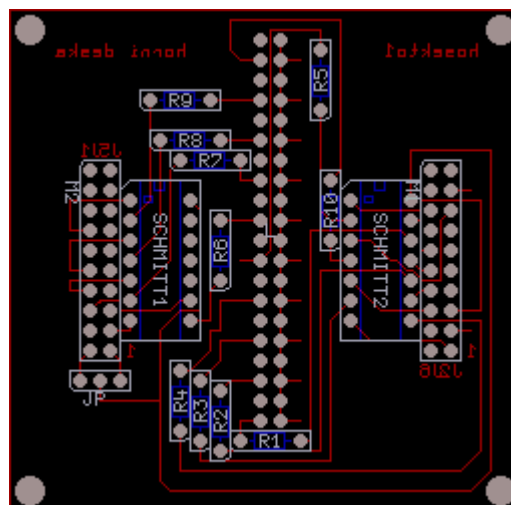


Obr. 27: Schéma zapojení horní desky

Na Obr. 28 a Obr. 29 jsou zobrazeny obrazce vodivých spojů na obou deskách spolu s rozmístěním součástek.



Obr. 28: Dolní deska v programu Formica 4.40 Layout



Obr. 29: Horní deska v programu Formica 4.40 Layout

## 7 Popis programu

V sedmé části mé bakalářské práce popíšu postup, který jsem praktikoval při tvorbě programového kódu.

Programově je třeba nastavit, jaký časovač bude použit, dále jsem nastavil jednotku generující PWM signál a také jsem nastavil, které piny budou použity jako vstupní a které jako výstupní. Nevyužité piny jsem pro jistotu nastavil jako výstupní, aby nesloužily jako přijímač rušivých signálů zvenčí a neovlivňovaly běh programu.

Jelikož jsem potřeboval, aby přerušení nastávalo v pravidelných intervalech, použil jsem jediný časovač `CpuTimer1` který má frekvenci 50 MHz. Ten jsem nastavil tak, aby k přerušení došlo každých 50  $\mu$ s. Veškeré řízení otáček motorku a LED je napsáno v podprogramu použitého časovače. Tento podprogram je rozdělen do tří hlavních částí. V první části je naprogramován rozběh motorku pomocí PWM signálu na jmenovité otáčky. V druhé části je aktivována světelná závora použitá jako senzor otáček. V této části podprogramu je řešeno zapínání a vypínání světelných diod, k čemuž se vrátím v samostatném odstavci (viz. odstavec 7.1 Běžící text). V poslední části podprogramu je motorek opět pomocí PWM signálu zpomalen na mez, při které se ještě dokáže točit a v tomto okamžiku je motorek odpojen a celá rotační část se setrvačností dotáhčí. Styk kartáče s povrchem měděné vrstvy vytváří třecí sílu, která přispívá ke snižování rychlosti otáčení. Kromě této síly působí i odpor vzduchu a též magnetické síly uvnitř motorku.

Výše popsáný děj bude trvat přibližně 90 sekund a po této době se spustí řízené blikání LED, jehož smyslem je upozornění na ukončení předvedení výsledku mé bakalářské práce.

Pro sestavení funkčního programu jsem vybral z nabídky knihovny jednoduchý vzor obsahující řízení jedné LED pomocí časovače. Poté jsem nepotřebné části šablony odstranil a dopsal si vlastní část kódu programu. Budu-li postupovat v programu od prvního řádku k poslednímu, nejprve jsem vytvořil proměnné a přiřadil jim konkrétní hodnotu (deklarace a inicializace proměnných) a podprogram přerušení od časovače, posléze jsem využil již hotové nastavení registrů časovače, dále jsem nastavil parametry časovače, jimiž jsou název, pracovní frekvence a doba, po které dojde k vyvolání přerušení. Původně jsem si myslel, že třetí údaj odpovídá počtu period, ale experimentálně jsem toto vyloučil. Údaj bude odpovídat počtu period po vynásobení číslem  $k = 41\frac{2}{3}$ . Po ujasnění si této situace již nebyl žádný problém s nastavením požadované periody přerušení.

Následně jsem nastavil požadované vstupy a výstupy. Užil jsem pouze digitálních vstupů a výstupů, které jsem nastavil takto:

```
GpioCtrlRegs.GPAMUX1.bit.GPIO2 = 1;  
GpioCtrlRegs.GPADIR.bit.GPIO2 = 1;
```

První řádek jsem nastavil podle tabulky *Table 6-49*. z datového listu mikroprocesoru tak, aby pin č.3 v řadě *J1* byl nastaven na digitální vstup nebo výstup. Druhý řádek potom rozhoduje o tom, zdali se jedná o výstup, nebo o vstup. Za rovnítkem v druhém řádku je znak 1 odpovídající nastavení digitálního výstupu. Obdobně jsem postupoval u zbývajících pinů propojených s LED.

Čísla pinů a jejich přiřazení shrnuje následující tabulka. [10]

Tab. 10: Nastavení pinů na vývojové desce

č. pinu	funkce	připojeno
J6/2	EPWM1B	signál pulsně šířkové modulace, L293B
J6/3	GPIO2	LED
J6/4	GPIO3	LED
J6/5	GPIO4	LED
J6/6	GPIO5	LED
J2/1	GND	pól -
J2/3	GPIO12	LED
J1/5	GPIO34	LED
J2/8	GPIO6	LED
J2/9	GPIO7	TCST1103
J1/1	+3,3 V	napěťová reference pro indikaci otáček
J1/3	GPIO28	LED
J1/4	GPIO29	LED
J1/7	GPIO18	LED
J5/1	+5 V	napájení vývojové desky a 74AC14
J5/2	GND	napájení vývojové desky

U pinu č. 9 v řadě *J2* jsem v druhém řádku uvedl nulu, jelikož se jedná o vstup a to o jediný digitální vstup použitý v mé práci. Piny *J1/1*, *J2/1*, *J5/1* a *J5/2* jsem nastavovat nemusel. Jejich připojení do obvodu nelze měnit. U posledního použitého pinu *J6/2* jsem za obě rovnítka napsal *I*, aby byla připojena jednotka generující PWM signál a aby se opět jednalo o digitální výstup. Shodně jsem nastavil i pin *J6/1*, ale jeho použití nebylo zapotřebí, neboť jsem druhý digitální vstup čtyřkanálového budiče L293B propojil se svorkou GND. Způsob řešení ničemu nebrání, otáčení je realizováno jen na jednu stranu. Jednotka generující PWM signál však umožňuje vytvářet obdélníkové průběhy pro oba dva výstupy (piny) různé, takže lze lépe modulovat napětí mezi oběma piny. Výstupní napětí se pak nebude pohybovat v mezích  $0\text{ V}$  až  $+U_{max}$ , ale v rozmezí  $-U_{max}$  až  $+U_{max}$ . Maximální výstupní napětí pak bude mít dvojnásobnou velikost, než při užití jednoho výstupu.

PWM jednotku jsem nastavil podle manuálu jednotky umístěného na webových stránkách výrobce vývojové destičky, ve kterém se nachází mnoho příkladů užití a jsou zde napsány úryvky kódu pro nastavení jednotky. [30]

Vývojový diagram popisující běh programu je znázorněn na Obr. 30.

Programový kód, ze kterého lze vyčíst instrukce s přidanými komentáři obsahuje příloha č. 2.



## 7.1 Běžící text

Světelné diody jsou umístěny v jedné řadě v počtu 10 ks. Tato řada LED leží na polopřímce vycházející z pomyslného bodu na desce, kterým prochází osa otáčení. Světelné diody označíme pro jejich případnou identifikaci *LED1* až *LED10*. Text tak bude možné zobrazovat na ploše mezikruží. Poloměr vnější kružnice tvoří *LED1* a poloměr menší vnitřní kružnice tvoří *LED10*.

V potaz jsem musel brát skutečnost, že vytvořený text pomocí všech LED bude značně rozbíhavý kvůli rozdílné dráze jednotlivých diod. Obvod trajektorie nejvnitřnější světelné diody je skoro 6 krát menší oproti *LED1*. Abych docílil rovnoběžnosti písmenek, musel bych uskutečnit jednu z těchto možností:

- zvětšit poloměry kružnic a nejlépe umístit LED těsně vedle sebe (ideálně zbroušením jejich pouzdra),
- zajistit 100% stabilitu otáček (, aby se text zobrazoval na stálém místě) a vhodně přepsat instrukce pro každou LED zvlášť, protože nyní by již nebyly rozsvěcovány současně např. při vytváření znaku „I“,
- nebo umístit světelné diody na stěnu pomyslného válce do jedné řady, přičemž válec by rotoval kolem osy procházející středy jeho podstav a všechny LED by se tím pádem pohybovaly po stejné velké trajektorii.

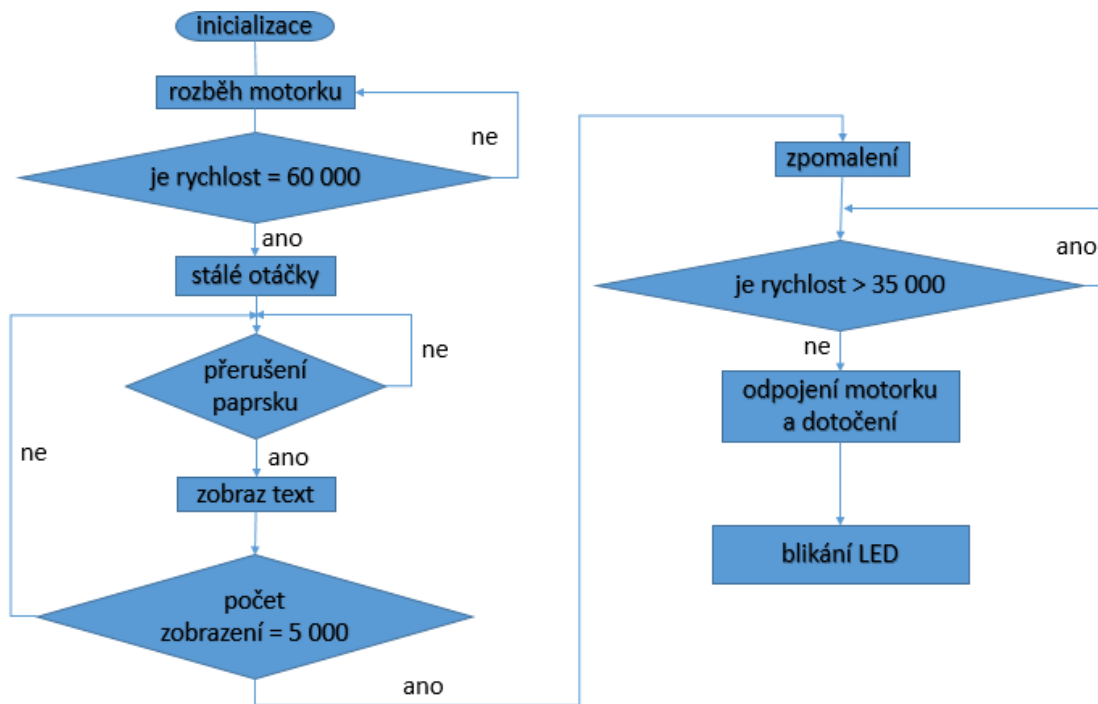
Možnost a) jsem zamítnul z důvodu, že bych musel sestavenou horní desku předělávat, zvětšily by se její rozměry (, nebo alespoň rozměry držáku diod), zvýšila by se celková hmotnost a vznikly by s tím spojené problémy.

Druhá možnost se jevila jako uskutečnitelná a původně jsem s tímto způsobem realizace běžícího textu počítal. Problémy spojené se zjišťováním skutečné rychlosti otáčení soustavy uskutečnění znemožnily.

Třetí možnost nebyla realizovatelná vůbec, napadla mě příliš pozdě. Může být však inspirací pro někoho dalšího.

Na několika následujících řádcích nastíním vytváření textu. Popíšu vznik písmena „T“ pomocí všech deseti diod. Vodorovná čára vznikne rozsvícením diody s největší vzdáleností od osy otáčení na určitý čas. Bude-li například doba jedné otáčky 25 ms a zobrazený znak zabere jednu pětinu využitelné plochy, bude použita LED svítit 5 ms. Svislou čáru vytvoříme společným rozsvícením všech LED na velmi krátkou dobu, například 0,02 ms. Cílem je, aby bliknutí diod bylo bodové a ne liniové jako u vodorovné čáry. V čase  $t = 0$  s by se tedy rozsvítila *LED1*, v čase  $t = 2,49$  ms by se rozsvítily i zbývající LED a svítily by do  $t = 2,51$  ms. V tomto okamžiku by zhasly s výjimkou *LED1*, která by zhasla až v čase  $t = 5$  ms.

Následující obrázek znázorňuje běh programu pomocí vývojového diagramu.



Obr. 30: Vývojový diagram vytvořeného programu

## 8 Problémy a jejich řešení

Během vytváření sestavy a jejího zprovoznování nastaly různé komplikace, které bylo nutné odstranit. Zde uvedu výčet všech zásadních závad, vysvětlím jejich příčinu, pokud je známa a proč bylo nutné je odstranit, způsob odstranění.

1) Nevhodná realizace dolní desky. Dolní deska plošných spojů měla z výroby pokovené zlatem pouze pájecí body, které byly z výroby provrtané a otvor pokoven, ovšem vše ostatní bylo překryto finální maskou. Na desce se přitom nachází vodivá plocha ve tvaru soustředných kružnic, které slouží k napájení sestavy a měly být pozlacené. Odstranění tohoto problému šlo pouze zčásti, odbroušením masky v těchto místech. Důsledkem je zhoršení elektrických vlastností sběracího ústrojí, poněvadž ničím nechráněný povrch mědi na vzduchu oxiduje. Vrstva sloučenin mědi a vzdušného kyslíku je přitom nevodivá. Tu je nutné po zhoršení elektrických vlastností nad určitou mez odstranit odbroušením. Není to dlouhodobé řešení, protože postupně přicházíme o vrstvu mědi na desce. Pro tento projekt to ovšem postačuje.

2) Použití nevhodného senzoru pro indikaci otáček. Plánovaný senzor s odrazem paprsků CNY70 se neosvědčil, proto byl nahrazen optronem pro indikaci přerušení paprsku TCST1103. Zároveň bylo nutné pozměnit umístění součástky na dolní desce a propojení s obvodem.

3) Špatný kontakt mezi LED na horní desce a dutinkovou lištou. Dutinková lišta se nacházela na horní desce na straně vývodů, přičemž je to prvek typu THT. Nedokonalým zapájením vývodů, které jsem ohnul pod úhlem  $90^\circ$ , abych vytvořil součástku typu SMT<sup>22</sup>, vznikly studené spoje a LED nesvítily. Též umístění jejich vývodů do dutinkové lišty nebylo vyhovující. Současně LED nebyly rozmístěny tak, aby při rotaci vytvářely mezikružní, poněvadž neležely mimo osu motorku, ale přímo v ose ležela jedna z prostředních LED. Efekt jednotlivých LED nacházejících se ve stejné vzdálenosti od osy motorku, by se překrýval. Proto jsem se rozhodl dutinkovou lištu vyjmout a nahradit ji plastovým držákem, ve kterém se nacházejí otvory pro LED. Propojení světelných diod s horní deskou je uskutečněno krátkými odizolovanými drátky připájenými ke zkráceným vývodům LED a na druhé straně jsou zapájeny do otvorů v desce. Nevýhodou je možnost výměny LED pouze odpájením.

4) Nerovnoměrné rozmístění hmotnosti vývojové destičky k ose otáčení. Při rotaci dochází k vibraci. Částečně jsem tento problém odstranil umístěním dodatečných součástek (optron, elektrolytický kondenzátor - viz. další bod) na opačnou stranu, odstředivé síly se kompenzují.

5) Pokles napájecího napětí při rotaci nedokonalým kontaktem mezi kartáči sběrače a napájecími plochami. Souvisí s bodem 1). Během rotace došlo při zkoušení řízení motorku na osazené desce k opakovanému resetování programu, kdy při určité rychlosti otáček sestavy nebyla vývojová deska napájena potřebnou velikostí napětí. Nedostatek byl odstraněn elektrolytickým kondenzátorem s kapacitou  $C = 1000 \mu F$  pro maximální napětí  $16 V$ . Kondenzátor jsem připojil paralelně k napájecím plochám a umístil jej na dolní desku tak, aby se snížila nerovnoměrnost rozložení hmotnosti rotující části k ose otáčení.

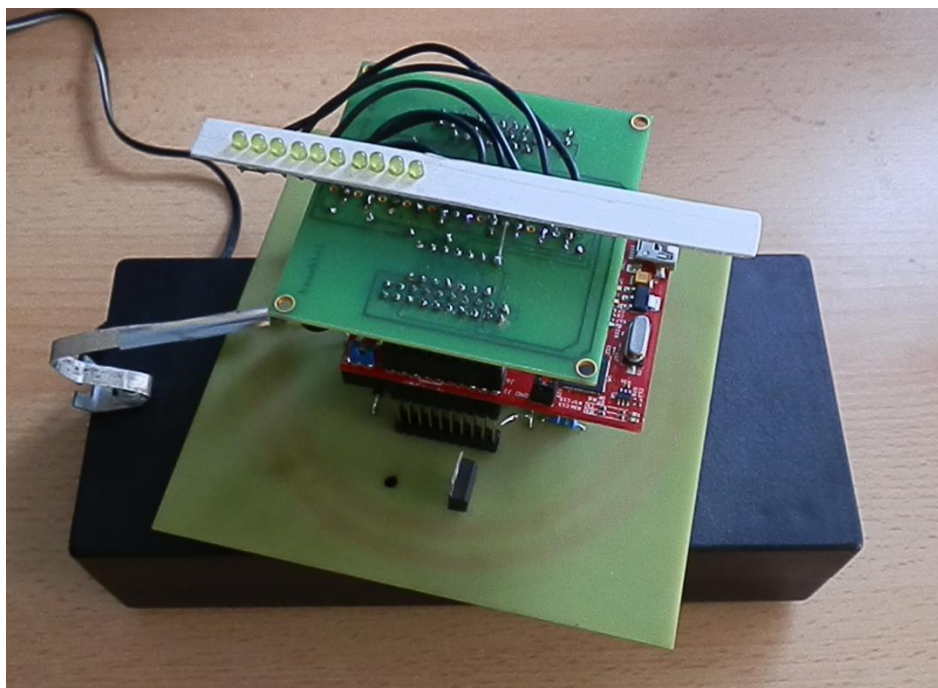
6) Chybějící propojení na horní desce mezi katodami LED a GND. Vyřešeno napájením propojovacího drátku. Katody jsou vzájemně propojeny přímo na desce plošných spojů.

---

<sup>22</sup> SMT = Surface Mount Technology, označení pro součástky, které jsou připájeny k DPS na straně vodivých obrazců, není třeba vrtat otvory jako u THT. [31]

## 9 Závěr

Po několikaměsíčním úsilí se mi podařilo realizovat nápad vedoucího práce. Teoretické předpoklady probrané na začátku semestru se mi dařilo postupně uskutečňovat, vytvářet sestavu, na které by se následně ověřilo, zdali byl nápad proveditelný a do jaké míry se mi to povedlo. Během realizace nastávaly problémy spojené s konstrukcí sestavy i s tvorbou programového kódu, kdy některé teoretické předpoklady nevedly ke zdárnému výsledku. Řešením problémů a nedostatků docházelo v průběhu vývoje sestavy k technickým úpravám, změnám použitých součástek a jinému nahlédnutí na strukturu programu v případě, že napsaný programový kód neověřil předpokládané chování fyzického objektu (sestavy). Naučil jsem se pracovat v programovém prostředí Code Composer, navrhnout desku plošných spojů a nakreslit za pomoci vhodného nástroje schéma elektrického obvodu. Řešením problematiky jsem si osvojil postupy, umožňující dosažení realizace praktické části bakalářské práce.



Obr. 31: Hotová sestava

Spolu s vedoucím práce jsme zjistili, že užití výše popsaného sběracího ústrojí nebylo šťastným nápadem, protože jeho správná funkce nebyla vždy zajištěna. Pokud bych se rozhodl na základě prototypu a získaných zkušeností sestavit novou sestavu, určitě bych zvolil jiný způsob napájení rotující části. Na trhu jsou k dispozici sběrací ústrojí, jejichž pohyblivé kontakty jsou pokoveny zlatem a třecí síly nejsou závislé na provedené montáži, jako v mém případě, kdy má na funkci vliv i připevnění pohyblivé části na základnu. Dále bych se pokusil určit skutečné otáčky sestavy jinak, než jak jsem se o to neúspěšně pokoušel mnoho dní. Výsledkem je, že nelze řídit motorek v uzavřené smyčce. Poslední zásadní změnou by bylo navržení vhodnějších rozměrů horní i dolní desky a zvětšení poloměru trajektorie nejnvnitřnější světelné diody, aby nebyl zobrazovaný text příliš deformovaný.

I přes popsané nedostatky si myslím, že mnou vytvořená sestava splnila očekávání vedoucího mé bakalářské práce a lze ji považovat za úspěch.

# Seznam obrázků

Obr. 1: Vývojová deska LaunchPad C2000™ .....	1
Obr. 2: Rozebraný stejnosměrný motorek malého výkonu .....	2
Obr. 3: Typy rotorových plechů .....	3
Obr. 4: Ventilátor na stejnosměrné napětí malého výkonu.....	3
Obr. 5: Princip otáčející se cívky umístěné v magnetickém poli .....	4
Obr. 6: Obvodové schéma stejnosměrného cize buzeného motoru.....	5
Obr. 7: Zatěžovací charakteristiky stejnosměrného cize buzeného motoru .....	5
Obr. 8: Průběh napětí na zátěži při řízení podle bodu 4 .....	8
Obr. 9: Stejnosměrný měnič napětí jako dvojbran .....	9
Obr. 10: Obvodové schéma stejnosměrného měniče napětí s aktivní zátěží .....	11
Obr. 11: Jednofázový napěťový střídač s RL zátěží .....	12
Obr. 12: Struktura tyristoru .....	14
Obr. 13: Struktura bipolárního tranzistoru .....	15
Obr. 14: Struktura unipolárního tranzistoru .....	16
Obr. 15: Otevřená okna programu Code Composer .....	19
Obr. 16: Rozměry motorku .....	22
Obr. 17: Pouzdro TO220 .....	23
Obr. 18: Zapojení stabilizátoru do obvodu s kondenzátory .....	23
Obr. 19: Pouzdro typu DIP16.....	24
Obr. 20: Blokové schéma L293B .....	24
Obr. 21: Pouzdro součástky TCST1103 .....	25
Obr. 22: Porovnání principu senzorů .....	26
Obr. 23: Obrázek diody 1N5819 .....	26
Obr. 24: Pouzdro součástky 74AC14 .....	27
Obr. 25: Obrázek LED .....	27
Obr. 26: Schéma zapojení dolní desky .....	29
Obr. 27: Schéma zapojení horní desky.....	29
Obr. 28: Dolní deska v programu Formica 4.40 Layout .....	30
Obr. 29: Horní deska v programu Formica 4.40 Layout .....	30
Obr. 30: Vývojový diagram vytvořeného programu .....	34
Obr. 31: Hotová sestava .....	36

## Seznam tabulek

Tab. 1: Stav spínačů a vliv na obvod.....	13
Tab. 2: Součástky umístěné na dolní desce .....	21
Tab. 3: Součástky umístěné na horní desce .....	22
Tab. 4: Základní technické údaje a pouzdro součástky 7806 .....	23
Tab. 5: Technické údaje a obrázek budiče L293B.....	24
Tab. 6: Technické údaje použitého optronu a obrázek pouzdra.....	25
Tab. 7: Technické údaje a obrázek Schottkyho diody .....	26
Tab. 8: Technické údaje a obrázek pouzdra 74AC14 .....	27
Tab. 9: Technické údaje a obrázek LED.....	27
Tab. 10: Nastavení pinů na vývojové desce .....	32

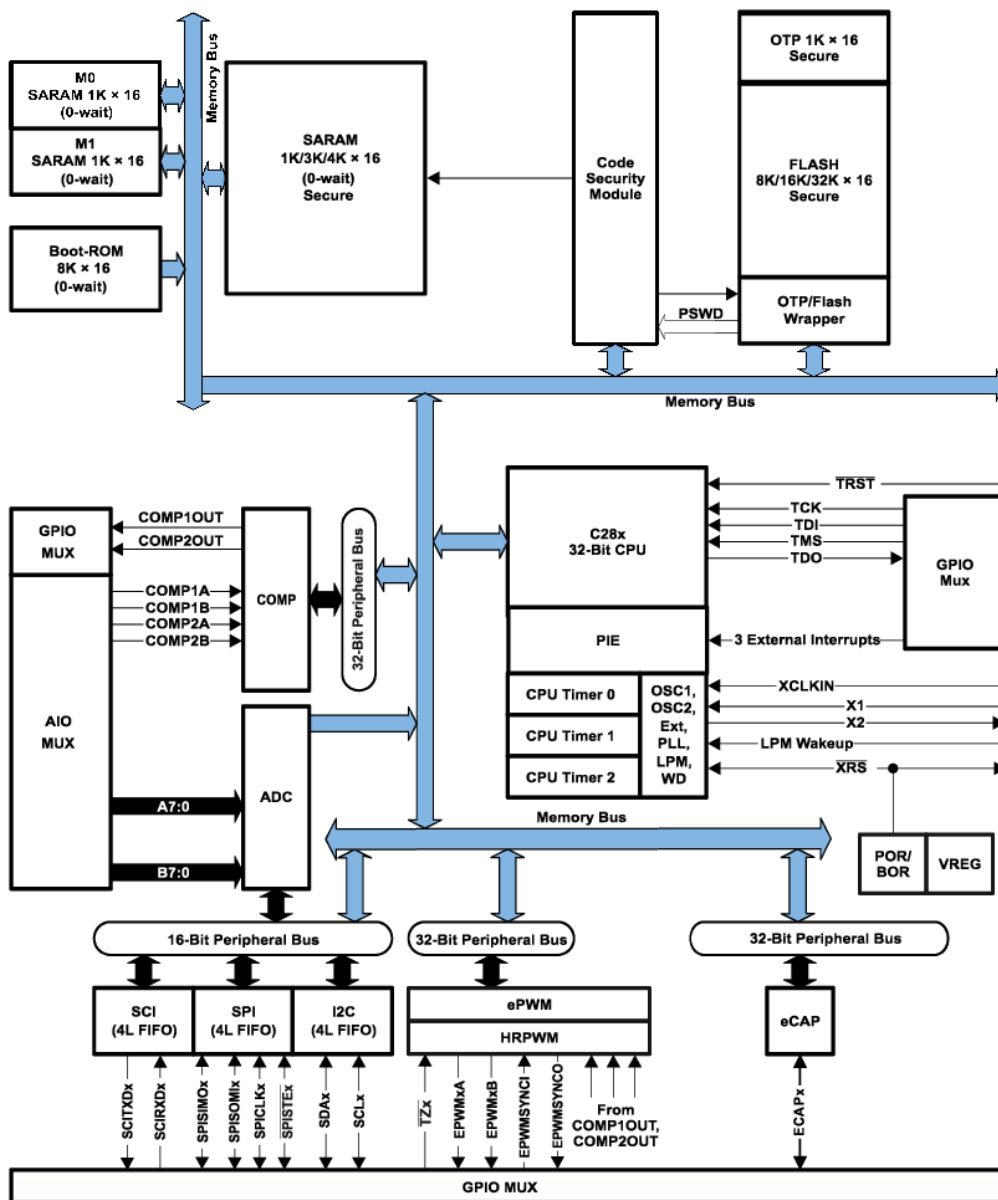
# Seznam použité literatury

- [1] VOŽENÍLEK, P., NOVOTNÝ V., MINDL P. *Elektromechanické měniče*. 2. vyd. Praha: České vysoké učení technické v Praze, 2011. ISBN 978-80-01-04875-7.
- [2] KOBRLE, P., PAVELKA, J. *Elektrické pohony a jejich řízení*. 3. přepracované vydání. V Praze: České vysoké učení technické, 2016. ISBN 978-80-01-06007-0.
- [3] DOLEČEK, J. *Moderní učebnice elektroniky*. Praha: BEN - technická literatura, 2005. ISBN 80-7300-161-6.
- [4] HAVLÍČEK, V., POKORNÝ, M., ZEMÁNEK, I. *Elektrické obvody*. 2. vyd. Praha: České vysoké učení technické v Praze, 2014-. ISBN 978-80-01-05492-5.
- [5] PAVELKA, J., ČEŘOVSKÝ, Z., LETTL, J. *Výkonová elektronika*. Vyd. 3., přeprac. Praha: Nakladatelství ČVUT, 2007. ISBN 978-80-01-03626-6.
- [6] CHOMJAK, M., KUNC, L., BELAY, M., CHOMJAK, P. Súnrak nad rušňami radu 110 na Slovensku. *Dráha*. Praha: NADATUR, 2016, **XXIII**(5), 12-18. ISSN 1211-1260.
- [7] DOLEČEK, J. *Moderní učebnice elektroniky*. Praha: BEN - technická literatura, 2005. ISBN 80-7300-146-2.
- [8] Datový list tyristoru TV 989-2700-28 [online]. [vid. 14. 5. 2017]. Dostupné z: [http://web.archive.org/web/20071218125215/http://www.polovodice.cz/\\_files/specifikace/tv989-2700-28.pdf](http://web.archive.org/web/20071218125215/http://www.polovodice.cz/_files/specifikace/tv989-2700-28.pdf)
- [9] Katalog polovodičů ČKD [online]. [vid. 14. 5. 2017]. Dostupné z: <http://danyk.cz/ckd.html>
- [10] TMS320F2802x Picolo™ Microcontrollers [online]. Texas Instruments Incorporated, 2017. [vid. 14. 5. 2017]. Dostupné z: <http://www.ti.com/lit/ds/sprs523k/sprs523k.pdf>
- [11] LAUNCHXL-F28027 C200 Piccolo LaunchPad Experimenter Kit, User's Guide [online]. Texas Instruments Incorporated, 2014. [vid. 14. 5. 2017]. Dostupné z: <http://www.ti.com/lit/ug/spruhh2a/spruhh2a.pdf>
- [12] C2000 Piccolo LaunchPad [online]. Texas Instruments Incorporated, 2016. [vid. 14. 5. 2017]. Dostupné z: <http://www.ti.com/tool/launchxl-f28027?DCMP=c2x-launchpad&HQS=c2000-launchpad>
- [13] HEROUT, P. *Učebnice jazyka C*. Praha: [Středisko pro podporu studentů se specifickými potřebami ELSA ČVUT], 2015. ISBN 978-80-7232-383-8.
- [14] Data Sheet DC Motor [online]. Kysan Electronics. [vid. 14. 5. 2017]. Dostupné z: <http://www.kysanelectronics.com/graphics/RF-500TB-1.pdf>
- [15] Positive voltage regulator ICs [online]. STMicroelectronics 2012. [vid. 14. 5. 2017]. Dostupné z: <https://www.gme.cz/data/attachments/dsh.330-002.1.pdf>
- [16] L293, L293D. Quadruple Half-H Drivers [online]. Texas Instruments Incorporated, 2016. [vid. 14. 5. 2017]. Dostupné z: <https://www.gme.cz/data/attachments/dsh.399-010.2.pdf>
- [17] Transmissive Optical Sensor with Phototransistor Output [online]. Vishay Semiconductor GmbH. [vid. 14. 5. 2017]. Dostupné z: <https://www.gme.cz/data/attachments/dsh.523-174.1.pdf>
- [18] DOLEČEK, J. *Moderní učebnice elektroniky*. Praha: BEN - technická literatura, 2005. ISBN 80-7300-184-5.
- [19] Reflective Optical Sensor with Transistor Output [online]. Vishay Semiconductor GmbH. [vid. 14. 5. 2017]. Dostupné z: <https://www.gme.cz/data/attachments/dsh.532-006.1.pdf>
- [20] Schottky Barrier Rectifier [online]. General Semiconductor. [vid. 14. 5. 2017]. Dostupné z: <https://www.gme.cz/data/attachments/dsh.223-001.1.pdf>
- [21] Schottkyho dioda 1N5819. In: *GMElectronics* [online]. GM elektronika, spol. s.r.o. © 2017. [vid. 14. 5. 2017]. Dostupné z: <https://www.gme.cz/1n5819>
- [22] Hex Inverter with Schmitt Trigger Input [online]. Fairchild Semiconductor Corporation, 2005. [vid. 14. 5. 2017]. Dostupné z: <https://www.gme.cz/data/attachments/dsh.429-013.1.pdf>
- [23] 3mm Round Yellow LED [online]. OptoSupply. [vid. 14. 5. 2017]. Dostupné z: <https://www.gme.cz/data/attachments/dsh.518-240.1.pdf>

- [24] KP3B [online]. GES-Electronics, a.s. 2017. [vid. 14. 5. 2017].  
Dostupné z: <https://www.ges.cz/cz/kp-3-b-GES07203772.html>
- [25] HEBL 21 [online]. GES-Electronics, a.s. 2017. [vid. 14. 5. 2017].  
Dostupné z: <https://www.ges.cz/cz/hebl-21-GES06601472.html>
- [26] MTS 500APC [online]. GES-Electronics, a.s. 2017. [vid. 14. 5. 2017].  
Dostupné z: <https://www.ges.cz/cz/mts-500apc-GES06500133.html>
- [27] Objímka pro 10mm LED Signal Construct SMB1149, M14 [online]. Conrad.  
[vid. 14. 5. 2017]. Dostupné z: <http://www.conrad.cz/objimka-pro-10mm-led-signal-construct-smb1149-m14.k180998>
- [28] LED 10mm oranžová 40-150mcd 60° Čelo: vypouklé. [online]. Svět žárovek, 2017.  
[vid. 14. 5. 2017]. Dostupné z: <http://www.svetzarovek.eu/p-807/k-1208/led-10mm-oranzova-40-150mcd-60-celo-vypoukle/>
- [29] PTF 5030 [online]. GES-Electronics, a.s. 2017. [vid. 14. 5. 2017].  
Dostupné z: <https://www.ges.cz/cz/ptf-5030-GES07900032.html>
- [30] TMS320x2802x, 2803x, Piccolo Enhanced Pulse Width Modulator (ePWM) Module, Reference Guide [online]. Texas Instruments Incorporated, 2011. [vid. 14. 5. 2017]. Dostupné z: <http://www.ti.com/sitesearch/docs/universalsearch.tsp?searchTerm=spruge9e.pdf#linkId=1&src=top>
- [31] MACH, P., URBÁNEK, J., SKOČIL, V. *Montáž v elektronice: pouzdření aktivních součástek, plošné spoje*. Praha: Vydavatelství ČVUT, 2001. ISBN 80-01-02392-3.
- [32] Vodič ve tvaru smyčky v magnetickém poli. In: *SlidePlayer* [online]. SlidePlayer © 2017.  
[vid. 16. 5. 2017]. Dostupné z: <http://slideplayer.cz/slide/2763327/10/images/13/Vodi%C4%8D+ve+tvaru+smy%C4%8Dky+v+magnetick%C3%A9m+poli.jpg>
- [33] Stabilizátor pevného napětí STMicroelectronics 7805CV-DG-STM. In: *GMElectronics* [online]. GM elektronik, spol. s.r.o. © 2017. [vid. 16. 5. 2017]. Dostupné z: <https://www.gme.cz/stabilizator-pevneho-napeti-st-microelectronics-thomson-7805cv-stm>



Příloha č. 1: Blokové schéma vývojové desky C2000 [10]



## Příloha č. 2: Programový kód

```
//#####  
//  
// FILE:    Example_2802xLedBlink.c  
//  
// TITLE:   f2802x LED Blink Getting Started Program.  
//  
// ASSUMPTIONS:  
//  
// This program requires the f2802x header files.  
//  
//  
// As supplied, this project is configured for "boot to SARAM"  
// operation. The 2802x Boot Mode table is shown below.  
//  
// $Boot_Table  
// While an emulator is connected to your device, the TRSTn pin = 1,  
// which sets the device into EMU_BOOT boot mode. In this mode, the  
// peripheral boot modes are as follows:  
//  
//      Boot Mode:  EMU_KEY          EMU_BMODE  
//                  (0xD00)         (0xD01)  
// -----  
// Wait           !=0x55AA          X  
// I/O             0x55AA            0x0000  
// SCI             0x55AA            0x0001  
// Wait           0x55AA            0x0002  
// Get_Mode       0x55AA            0x0003  
// SPI            0x55AA            0x0004  
// I2C            0x55AA            0x0005  
// OTP            0x55AA            0x0006  
// Wait           0x55AA            0x0007  
// Wait           0x55AA            0x0008  
// SARAM          0x55AA            0x000A    <-- "Boot to  
SARAM"  
// Flash          0x55AA            0x000B  
// Wait           0x55AA            Other  
//  
// Write EMU_KEY to 0xD00 and EMU_BMODE to 0xD01 via the debugger  
// according to the Boot Mode Table above. Build/Load project,  
// Reset the device, and Run example  
//  
// $End_Boot_Table  
//  
// DESCRIPTION:  
//  
// This example configures CPU Timer0 for a 500 msec period, and toggles  
// the  
// GPIO34 LED once per interrupt. For testing purposes, this example  
// also increments a counter each time the timer asserts an interrupt.  
//  
// Watch Variables:  
//      CpuTimer0.InterruptCount  
//  
// Monitor the GPIO34 LED blink on (for 500 msec) and off (for 500  
msec)  
// on the 2802x control card.  
//
```

```

#####
// $TI Release: F2802x Support Library v230 $
// $Release Date: Fri May 8 07:43:05 CDT 2015 $
// $Copyright: Copyright (C) 2008-2015 Texas Instruments Incorporated -
// http://www.ti.com/ ALL RIGHTS RESERVED $
#####

#include "DSP28x_Project.h" // Device Headerfile and Examples Include
File

// Prototype statements for functions found within this file.
__interrupt void cpu_timer1_isr(void);
long n=0; //promenna slouzi pro rizeni programu rizeni
motorku
int p=1;
int q=1;
int p_1=0;
Uint16 s_1=1; //s=speed, otacky; promenna slouzici k
urcovani rychlosti zvysovani otacek motorku
int d=1; //d=delay, zpozdeni; promenna slouzi ke
spusteni rizeni motorku se zpozdenim po zapnuti
Uint16 s=1; //s=speed, otacky, promenna ridi primo otacky
motorku, slouzi pro jednotku generujici PWM signal
long a_p=1; //ap=auxiliary period, promenna slouzici k
rizenemu blikani na konci programu
int a_v=0; //av=auxiliary variable (for blinking),
zavisla na velikosti a_p, urcuje kdy ktera LED sviti
int s_2=0; //s=speed, promenna slouzici ke zpomalení
narustu otacek v urcitem kroce
void main(void)
{
// WARNING: Always ensure you call memcpy before running any functions from
RAM
// InitSysCtrl includes a call to a RAM based function and without a call to
// memcpy first, the processor will go "into the weeds"
#ifdef _FLASH
memcpy(&RamfuncsRunStart, &RamfuncsLoadStart,
(size_t)&RamfuncsLoadSize);
#endif

// Step 1. Initialize System Control:
// PLL, WatchDog, enable Peripheral Clocks
// This example function is found in the f2802x_SysCtrl.c file.
InitSysCtrl();

// Step 2. Initialize GPIO:
// This example function is found in the f2802x_Gpio.c file and
// illustrates how to set the GPIO to it's default state.
// InitGpio(); // Skipped for this example

// Step 3. Clear all interrupts and initialize PIE vector table:
// Disable CPU interrupts
DINT;

// Initialize the PIE control registers to their default state.
// The default state is all PIE interrupts disabled and flags
// are cleared.
// This function is found in the f2802x_PieCtrl.c file.
InitPieCtrl();

```

```

// Disable CPU interrupts and clear all CPU interrupt flags:
IER = 0x0000;
IFR = 0x0000;

// Initialize the PIE vector table with pointers to the shell Interrupt
// Service Routines (ISR).
// This will populate the entire table, even if the interrupt
// is not used in this example. This is useful for debug purposes.
// The shell ISR routines are found in f2802x_DefaultIsr.c.
// This function is found in f2802x_PieVect.c.
InitPieVectTable();

// Interrupts that are used in this example are re-mapped to
// ISR functions found within this file.
EALLOW; // This is needed to write to EALLOW protected registers
PieVectTable.TINT1 = &cpu_timer1_isr;
EDIS; // This is needed to disable write to EALLOW protected registers

// Step 4. Initialize the Device Peripheral. This function can be
// found in f2802x_CpuTimers.c
InitCpuTimers(); // For this example, only initialize the Cpu Timers
// Configure CPU-Timer 0 to interrupt every 500 milliseconds:
// 60MHz CPU Freq, 50 millisecond Period (in uSeconds)
ConfigCpuTimer(&CpuTimer1, 50, 60); //preruseni kazdych 50us

// To ensure precise timing, use write-only instructions to write to the
// entire register. Therefore, if any of the configuration bits are changed
// in ConfigCpuTimer and InitCpuTimers (in f2802x_CpuTimers.h), the
// below settings must also be updated.
CpuTimer1Regs.TCR.all = 0x4001; // Use write-only instruction to set TSS
bit = 0
// Step 5. User specific code, enable interrupts:

EALLOW;
//PWM
GpioCtrlRegs.GPAMUX1.bit.GPIO0 = 1; //nastaveni podle tabulky 6-49
GpioCtrlRegs.GPADIR.bit.GPIO0 = 1; //zajisti odruseni od okoli
GpioCtrlRegs.GPAMUX1.bit.GPIO1 = 1;
GpioCtrlRegs.GPADIR.bit.GPIO1 = 1;
//zavora
GpioCtrlRegs.GPAMUX1.bit.GPIO7 = 0;
GpioCtrlRegs.GPADIR.bit.GPIO7 = 0;
//LED 1
GpioCtrlRegs.GPAMUX1.bit.GPIO2 = 0;
GpioCtrlRegs.GPADIR.bit.GPIO2 = 1;
//LED 2
GpioCtrlRegs.GPAMUX1.bit.GPIO3 = 0;
GpioCtrlRegs.GPADIR.bit.GPIO3 = 1;
//LED 3
GpioCtrlRegs.GPAMUX1.bit.GPIO4 = 0;
GpioCtrlRegs.GPADIR.bit.GPIO4 = 1;
//LED 4
GpioCtrlRegs.GPAMUX1.bit.GPIO5 = 0;
GpioCtrlRegs.GPADIR.bit.GPIO5 = 1;
//LED 5
GpioCtrlRegs.GPAMUX1.bit.GPIO6 = 0;
GpioCtrlRegs.GPADIR.bit.GPIO6 = 1;
//LED 6

```

```

    GpioCtrlRegs.GPAMUX1.bit.GPIO12 = 0;
    GpioCtrlRegs.GPADIR.bit.GPIO12 = 1;
    //LED 7
    GpioCtrlRegs.GPAMUX2.bit.GPIO18 = 0;
    GpioCtrlRegs.GPADIR.bit.GPIO18 = 1;
    //LED 8
    GpioCtrlRegs.GPAMUX2.bit.GPIO28 = 0;
    GpioCtrlRegs.GPADIR.bit.GPIO28 = 1;
    //LED 9
    GpioCtrlRegs.GPAMUX2.bit.GPIO29 = 0;
    GpioCtrlRegs.GPADIR.bit.GPIO29 = 1;
    //LED 10
    GpioCtrlRegs.GPBMUX1.bit.GPIO34 = 0;
    GpioCtrlRegs.GPBDIR.bit.GPIO34 = 1;
    EDIS;

// Enable CPU INT1 which is connected to CPU-Timer 0:
//IER |= M_INT1;
IER |= M_INT13;

// Enable TINT0 in the PIE: Group 1 interrupt 7
PieCtrlRegs.PIEIER1.bit.INTx7 = 1;

// Enable global Interrupts and higher priority real-time debug events:
EINT; // Enable Global interrupt INTM
ERTM; // Enable Global realtime interrupt DBGEM

// Initialization Time
// = = = = =
EPwm1Regs.TBPRD = 60000; // Period = 601 TBCLK counts
EPwm1Regs.CMPA.half.CMPA = 350; // Compare A = 350 TBCLK counts
EPwm1Regs.CMPB = 200; // Compare B = 200 TBCLK counts
EPwm1Regs.TBPHS.all = 0; // Set Phase register to zero
EPwm1Regs.TBCTR = 0; // clear TB counter
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UP;
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Phase loading disabled
EPwm1Regs.TBCTL.bit.PRDLN = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCSEL = TB_SYNC_DISABLE;
EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1; // TBCLK = SYSCLKOUT
EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV1;
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on TBCTR = Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on TBCTR = Zero
EPwm1Regs.AQCTLA.bit.PRD = AQ_CLEAR;
EPwm1Regs.AQCTLA.bit.CAU = AQ_SET;
EPwm1Regs.AQCTLB.bit.PRD = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.CBU = AQ_SET;
// Run Time
// = = = = =
EPwm1Regs.CMPA.half.CMPA = 00; // adjust duty for output EPWM1A
EPwm1Regs.CMPB = 0; // adjust duty for output EPWM1B

// Step 6. IDLE loop. Just sit and loop forever (optional):
// ZACATEK PROGRAMU
for(;;);
}
__interrupt void cpu_timer1_isr(void)
{

```

```

CpuTimer0.InterruptCount++;

EPwm1Regs.CMPA.half.CMPA = 0; // Compare A = 350 TBCLK counts
EPwm1Regs.CMPB = (60000-s); // Compare B = 200 TBCLK counts

// ZACATEK 1. CASTI
if (p==1){
    if(d<5){
        if(n==20001)          {n=1;}
        if((n % 200)==0)      {s_1++;}
        if((s_1 % 100)==0) {d++;s_1++;}
    }
    if(s==60000){p=2;}
    if(d==5){
        if(n>20000)          {n=1;}
        if((n % 1000)==0)    {s_1++;}
        if(s_1<600)         {s=36000;} //zde se
nastavuje poc.rychlost
    }
    else if(s_1<1200){
        s_2++;
        if(s_2==6){
            s++;
            s_2=0;
        }
    }
}
}
// ZACATEK 2. CASTI
if(p==2){
    if((GpioDataRegs.GPADAT.bit.GPIO7 == 0) && (q==1)) {
        q++; //zpusobi, ze se indikuje otacka pouze jednou behem
preruseni paprsku
        p_1++; //inkrementace kazdou otacku, podle poctu otacek
pak nastavuju, kdy se prepne do treti casti programu
    }
    GpioDataRegs.GPASET.bit.GPIO18 = 1; //zapis pro zhasnuti LED
    GpioDataRegs.GPASET.bit.GPIO29 = 1;
    GpioDataRegs.GPBSET.bit.GPIO34 = 1;
    if((p_1 > 500)&&(p_1 < 2500)){ //napis TH
// VYTVORENI PISMENA T
        if(((n > 100)&&(n < 156))){
            GpioDataRegs.GPACLEAR.bit.GPIO5 = 1; //zapis pro
rozsviceni LED
        }
        if(((n > 155)&&(n < 175))){
            GpioDataRegs.GPACLEAR.bit.GPIO2 = 1;
            GpioDataRegs.GPACLEAR.bit.GPIO3 = 1;
            GpioDataRegs.GPACLEAR.bit.GPIO4 = 1;
            GpioDataRegs.GPACLEAR.bit.GPIO5 = 1;
            GpioDataRegs.GPACLEAR.bit.GPIO6 = 1;
            GpioDataRegs.GPACLEAR.bit.GPIO12 = 1;
            GpioDataRegs.GPACLEAR.bit.GPIO28 = 1;
        }
        if(((n > 174)&&(n < 230))){
            GpioDataRegs.GPACLEAR.bit.GPIO5 = 1;
        }
    }
// VYTVORENI PISMENA H
    if(((n > 270)&&(n < 290))){
        GpioDataRegs.GPACLEAR.bit.GPIO2 = 1;
    }
}

```

```

        GpioDataRegs.GPACLEAR.bit.GPIO3 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO4 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO5 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO6 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO12 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO28 = 1;
    }
    if(((n > 289)&&(n < 381))){
        GpioDataRegs.GPACLEAR.bit.GPIO3 = 1;
    }
    if(((n > 380)&&(n < 400))){
        GpioDataRegs.GPACLEAR.bit.GPIO2 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO3 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO4 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO5 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO6 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO12 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO28 = 1;
    }
}
if((p_1 > 2000)&&(p_1 < 4500)){ //napis FEL
// VYTVORENI PISMENA F
    if((n > 480)&&(n < 500)){
        GpioDataRegs.GPACLEAR.bit.GPIO2 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO3 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO4 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO5 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO6 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO12 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO28 = 1;
    }
    if((n > 499)&&(n < 591)){
        GpioDataRegs.GPACLEAR.bit.GPIO3 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO5 = 1;
    }
    if((n > 589)&&(n < 610)){
        GpioDataRegs.GPACLEAR.bit.GPIO5 = 1;
    }
}
// VYTVORENI PISMENA E
    if((n > 650)&&(n < 670)){
        GpioDataRegs.GPACLEAR.bit.GPIO2 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO3 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO4 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO5 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO6 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO12 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO28 = 1;
    }
    if((n > 669)&&(n < 761)){
        GpioDataRegs.GPACLEAR.bit.GPIO3 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO5 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO6 = 1;
    }
    if((n > 759)&&(n < 780)){
        GpioDataRegs.GPACLEAR.bit.GPIO3 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO5 = 1;
    }
}
// VYTVORENI PISMENA L
    if((n > 820)&&(n < 840)){

```

```

        GpioDataRegs.GPACLEAR.bit.GPIO2 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO3 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO4 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO5 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO6 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO12 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO28 = 1;
    }
    if((n > 839)&&(n < 950)){
        GpioDataRegs.GPACLEAR.bit.GPIO6 = 1;
    }
}
if(p_1 > 4500){// napis OK
// VYTVORENI PISMENA O
    if((n > 100)&&(n < 121)){
        GpioDataRegs.GPACLEAR.bit.GPIO2 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO3 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO12 = 1;
    }
    if((n > 119)&&(n < 141)){
        GpioDataRegs.GPACLEAR.bit.GPIO4 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO28 = 1;
    }
    if((n > 139)&&(n < 191)){
        GpioDataRegs.GPACLEAR.bit.GPIO5 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO6 = 1;
    }
    if((n > 189)&&(n < 211)){
        GpioDataRegs.GPACLEAR.bit.GPIO4 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO28 = 1;
    }
    if((n > 209)&&(n < 230)){
        GpioDataRegs.GPACLEAR.bit.GPIO2 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO3 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO12 = 1;
    }
}
// VYTVORENI PISMENA K
    if((n > 270)&&(n < 291)){
        GpioDataRegs.GPACLEAR.bit.GPIO2 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO3 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO4 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO5 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO6 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO12 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO28 = 1;
    }
    if((n > 306)&&(n < 328)){
        GpioDataRegs.GPACLEAR.bit.GPIO2 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO12 = 1;
    }
    if((n > 343)&&(n < 365)){
        GpioDataRegs.GPACLEAR.bit.GPIO4 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO28 = 1;
    }
    if((n > 380)&&(n < 402)){
        GpioDataRegs.GPACLEAR.bit.GPIO5 = 1;
        GpioDataRegs.GPACLEAR.bit.GPIO6 = 1;
    }
}
}

```



```

        if(n==1181){n=1;} //zpusobi, ze promenna v teto casti programu
nabyva hodnot <1, 1201>
        if(p_1>4800){p=3;} //po indikaci 6000 otacek dojde k prepnuti do
3. casti programu
        else {
            q=1; //aby doslo k nastaveni podminky pro
detekci otacky

            GpioDataRegs.GPASET.bit.GPIO2 = 1;//LED zhasnuty
            GpioDataRegs.GPASET.bit.GPIO3 = 1;
            GpioDataRegs.GPASET.bit.GPIO4 = 1;
            GpioDataRegs.GPASET.bit.GPIO5 = 1;
            GpioDataRegs.GPASET.bit.GPIO6 = 1;
            GpioDataRegs.GPASET.bit.GPIO12 = 1;
            GpioDataRegs.GPASET.bit.GPIO18 = 1;
            GpioDataRegs.GPASET.bit.GPIO28 = 1;
            GpioDataRegs.GPASET.bit.GPIO29 = 1;
            GpioDataRegs.GPBSET.bit.GPIO34 = 1;
        }
    }
// ZACATEK 3. CASTI
    if(p==3){
        if(n==20001) {n=1;} //zpusobi, ze promenna v teto casti programu
nabyva hodnot <1, 20001>
        if((n % 10)==0) {s--;}
        //zpomalovani otacek kazdy 10. cyklus
        if(s<=35000) {s=0;p=0;} //pri dane mire zpomaleni,
odpoji motorek a vyskoci z 3. casti programu
        GpioDataRegs.GPASET.bit.GPIO2 = 1; //LED zhasnuty
        GpioDataRegs.GPASET.bit.GPIO3 = 1;
        GpioDataRegs.GPASET.bit.GPIO4 = 1;
        GpioDataRegs.GPASET.bit.GPIO5 = 1;
        GpioDataRegs.GPASET.bit.GPIO6 = 1;
        GpioDataRegs.GPASET.bit.GPIO12 = 1;
        GpioDataRegs.GPASET.bit.GPIO18 = 1;
        GpioDataRegs.GPASET.bit.GPIO28 = 1;
        GpioDataRegs.GPASET.bit.GPIO29 = 1;
        GpioDataRegs.GPBSET.bit.GPIO34 = 1;
    }
// KONEC RIZENI OTACEK
    if(n<80000){n++;} //timto zpusobem cekam na dotoceni rotacni casti
// ZAVERECNE BLIKANI LED
    if(n==80000){ //po zastaveni zacnou problikavat LED
        if(a_p==4000){ //vytvoreni periody pomoci promenne citani
            a_v++;
            a_p=0;
        }
        if(a_v==1){
            GpioDataRegs.GPATOGGLE.bit.GPIO5 = 1; //rozsviti LED
        }
        if(a_v==2){
            GpioDataRegs.GPADAT.bit.GPIO5 = 1; //zhasne LED
            GpioDataRegs.GPATOGGLE.bit.GPIO4 = 1;
        }
        if(a_v==3){
            GpioDataRegs.GPADAT.bit.GPIO4 = 1;
            GpioDataRegs.GPATOGGLE.bit.GPIO2 = 1;
        }
        if(a_v==4){
            GpioDataRegs.GPADAT.bit.GPIO2 = 1;

```

```

        GpioDataRegs.GPATOGGLE.bit.GPIO3 = 1;
    }
    if(a_v==5){
        GpioDataRegs.GPADAT.bit.GPIO3 = 1;
        GpioDataRegs.GPATOGGLE.bit.GPIO12 = 1;
    }
    if(a_v==6){
        GpioDataRegs.GPADAT.bit.GPIO12 = 1;
        GpioDataRegs.GPATOGGLE.bit.GPIO28 = 1;
    }
    if(a_v==7){
        GpioDataRegs.GPADAT.bit.GPIO28 = 1;
        GpioDataRegs.GPATOGGLE.bit.GPIO6 = 1;
    }
    if(a_v==8){
        GpioDataRegs.GPADAT.bit.GPIO6 = 1;
        GpioDataRegs.GPATOGGLE.bit.GPIO18 = 1;
    }
    if(a_v==9){
        GpioDataRegs.GPADAT.bit.GPIO18 = 1;
        GpioDataRegs.GPBTOGGLE.bit.GPIO34 = 1;
    }
    if(a_v==10){
        GpioDataRegs.GPBDAT.bit.GPIO34 = 1;
        GpioDataRegs.GPATOGGLE.bit.GPIO29 = 1;
    }
    if(a_v==11){
        GpioDataRegs.GPADAT.bit.GPIO29 = 1;
        a_v=0;
    }
    a_p++;
}

PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
}
// KONEC PROGRAMU

```