



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ  
ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY  
FACULTY OF MECHANICAL ENGINEERING  
INSTITUTE OF SOLID MECHANICS, MECHATRONICS  
AND BIOMECHANICS

## VIZUALIZACE VEKTORŮ MAGNETICKÉHO POLE V REÁLNÉM ČASE

REAL-TIME VISUALIZATION OF MAGNETIC FIELD VECTORS

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

TOMÁŠ KAPLAN

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. MICHAL MATĚJÁSKO

BRNO 2016

## **Abstrakt**

Tato práce se zabývá návrhem a realizací zařízení pro vizualizaci vektorů magnetického pole v reálném čase. Informaci o magnetickém poli zprostředkovávají senzory magnetické indukce, jejichž výstupní data jsou zpracována v mikrokontroleru a následně odeslána do počítače, kde probíhá jejich vizualizace. Cílem je zobrazení aktuální magnetické indukce ve smysluplné vizuální podobě.

## **Abstract**

This thesis deals with the design and realization of a device for a real-time visualization of magnetic field vectors. The information about magnetic field is obtained by magnetic flux density sensors. Outcoming data from sensors are processed in microcontroller and then sent to a personal computer where they are visualized. The goal is to visualize the magnetic flux density in meaningful form.

## **Klíčová slova**

Vizualizace v reálném čase, magnetická indukce, MPU9150, MAG3110, dsPIC, I<sup>2</sup>C, mikrokontrolér, DC motor.

## **Keywords**

visualization in real-time, magnetic flux density, MPU9150, MAG3110, dsPIC, I<sup>2</sup>C, microcontroller, DC motor.

KAPLAN, T. *Vizualizace vektorů magnetického pole v reálném čase*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2016. 54 s. Vedoucí bakalářské práce Ing. Michal Matějáško.

## **Zadání bakalářské práce**

Ústav:	Ústav mechaniky těles, mechatroniky a biomechaniky
Student:	<b>Tomáš Kaplan</b>
Studijní program:	Aplikované vědy v inženýrství
Studijní obor:	Mechatronika
Vedoucí práce:	<b>Ing. Michal Matějáško</b>
Akademický rok:	2015/16

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

### **Vizualizace vektorů magnetického pole v reálném čase**

#### **Stručná charakteristika problematiky úkolu:**

Tato práce navazuje na předchozí projekt, v rámci kterého bylo vyvinuto zařízení pro určení směru otáčení palivových pump, na základě změny magnetické indukce v jejích okolí. Cílem je vyvinout zařízení, a uživatelské rozhraní v PC, obdobné, jenž bude schopno vektory magnetické indukce v reálném čase vizualizovat. Data budou měřena pomocí senzorů magnetické indukce, zpracovávána v mikrokontroleru a posílána do PC, v němž budou následně vizualizována. Práce bude zahrnovat návrh DPS s potřebnou elektronikou, tvorbu programu pro mikrokontroler včetně komunikace se senzorem a PC, a tvorbu GUI v Matlabu. Cílem je zobrazit uživateli aktuální měřené veličiny v smysluplné vizuální podobě.

**Cíle bakalářské práce:**

- 1) Použijte elektroniku dostupnou v Mechatronické Laboratoři a seznámte se s problematikou programování mikrokontrolérů pomocí automatického generování kódu z prostředí Simulink. K dané elektronice následně připojte dostupný senzor magnetické indukce, a pomocí mikrokontroleru z něj čtete data, která následně přepošlete do PC.
- 2) V prostředí Matlab vytvořte grafické uživatelské rozhraní (GUI), ve kterém bude moci uživatel zahájit čtení dat z mikrokontroleru. Tato data vykreslujte v podobě vektoru. Vykreslování provádějte s pravidelnou časovou periodou.
- 3) Navrhněte podobu zařízení, se kterým bude možno měřit magnetické pole v jedné rovině. Toto zařízení by se mělo skládat z mechanické konstrukce a potřebných senzorů a elektroniky, aby mohlo být propojeno s PC a mohlo s ním být komunikováno pomocí vytvořeného GUI.
- 4) Pro navržené zařízení vyvíjte firmware, který bude zajišťovat potřebnou funkcionalitu.
- 5) Upravte vytvořené GUI a firmware v navrženém zařízení tak, aby uživatel mohl zahájit čtení dat ze zařízení, jejich vykreslování a přepínat režimy (např. vykreslování nekompenzovaného vs. kompenzovaného mag. pole o jeho statickou hodnotu apod.).

**Seznam literatury:**

Matějáško, M.: Detekce směru otáčení palivové pumpy, BP VUT FSI, 2013

Herout, P.: Učebnice jazyka C, Kopp 2011

Záhlava, V.: Návrh a konstrukce DPS, BEN 2010

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2015/16

V Brně, dne

L. S.

---

prof. Ing. Jindřich Petruška, CSc.  
ředitel ústavu

---

doc. Ing. Jaroslav Katolický, Ph.D.  
děkan fakulty

Prohlašuji, že jsem bakalářskou práci vypracoval sám na základě svých znalostí, snahy, rad a pokynů vedoucího bakalářské práce. Veškeré použité podklady a literaturu jsem uvedl.

V Brně dne .....

.....

Tomáš Kaplan

Chtěl bych poděkovat vedoucímu bakalářské práce Ing. Michalu Matějáskovi za zprostředkování tématu, ochotu a cenné rady a zkušenosti, které mi v průběhu práce předal. Dále pak rodině za neustálou podporu v průběhu studia a také své přítelkyni Lucii za trpělivost při mé tvůrčí činnosti.

Tomáš Kaplan

# Obsah

<b>1</b>	<b>Úvod</b>	<b>7</b>
<b>2</b>	<b>Rešerše proveditelnosti</b>	<b>8</b>
2.1	Hallův jev	8
2.2	Senzor magnetické indukce MAG3110	8
2.3	Inerciální měřicí jednotka MPU9150	8
2.4	I <sup>2</sup> C sběrnice	9
2.4.1	Komunikace prostřednictvím I <sup>2</sup> C	10
2.5	Vývojová jednotka s mikroprocesorem dsPIC	10
2.5.1	Program pro čtení dat ze senzoru	11
2.5.2	UART sběrnice	13
2.6	Experiment pro srovnání senzorů	13
2.6.1	Výsledky experimentu	14
<b>3</b>	<b>Elektronika zařízení</b>	<b>15</b>
3.1	Blokové schéma zařízení	15
3.2	DPS s mikrokontrolérem dsPIC	16
3.3	Mikrokontrolér dsPIC 33EP512MC806	20
3.4	Řídící obvod DC motoru L293D	21
3.5	DPS řízení motoru	22
3.6	DPS kyvadla se senzory	24
3.7	DC motor	27
<b>4</b>	<b>Software zařízení</b>	<b>28</b>
4.1	MPLAB Device Blocks for Simulink toolbox	28
4.2	Simulink model programu pro mikrokontroler	29
4.2.1	Konfigurační subsystém	30
4.2.2	Subsystém řízení motoru	30
4.2.3	Subsystém READING & SENDING DATA	35
4.2.4	Subsystém UART receive	40
4.3	Skripty a funkce na straně počítače	41
4.3.1	Hlavní skript obsluhující chod měření	42
4.3.2	Skript pro zpracování příchozích zpráv	45
4.4	Grafické uživatelské rozhraní	46
4.5	Experiment s palivovou pumpou	47
4.6	Ovládání zařízení	48
<b>5</b>	<b>Konstrukce zařízení</b>	<b>49</b>
<b>6</b>	<b>Závěr</b>	<b>51</b>
<b>7</b>	<b>Seznam použitých zkratk a symbolů</b>	<b>53</b>

# 1. Úvod

Při návrhu zařízení obsahujícího prvky generující magnetické pole může nastat situace, kdy toto pole vykazuje stěží předvídatelné chování. Jedním z příkladů, kterého se tento problém týká, je spínání palivové pumpy. Ta může měnit směr i rychlost otáčení a tím může být chování magnetického pole nesnadno odhadnutelné.

Tato práce si klade za cíl vyvinout prototyp zařízení, které bude schopno zprostředkovat smysluplnou vizualizaci intenzity magnetického pole v okolí zkoumaného předmětu. Práce popisuje návrh zařízení v podobě elektrickým motorem poháněného kyvadla, na kterém jsou umístěny senzory magnetické indukce. Zařízení má za úkol nasnímat magnetickou indukci na ploše opsané kyvadlem při jeho kyvu a následně data ze senzorů zaslat do počítače ke smysluplné vizualizaci v reálném čase.

Díky vizualizaci naměřených dat nám prototyp nabízí představu o směru a velikosti vektorů magnetické indukce v blízkosti zkoumaného předmětu, jakým může být například zmíněná palivová pumpa.



## 2. Rešerše proveditelnosti

Začátek návrhu spočíval ve výběru vhodného senzoru magnetické indukce, který je dostupný na trhu a má dostačující citlivost. V mechatronické laboratoři (dále jen Mechlab) byly k dispozici 2 senzory pracující na principu Hallova jevu.

### 2.1. Hallův jev

Pokud umístíme vodič (např. tenká destička z mědi), kterým protéká proud, do magnetického pole, vznikne na jeho stěnách napětí [1]. Toto napětí se nazývá *Hallovo napětí* a je způsobené silou, která posouvá vodivostní elektrony ke kraji vodiče. Tento fyzikální jev se nazývá *Hallův jev*. Na jeho principu fungují dnešní senzory magnetické indukce.

### 2.2. Senzor magnetické indukce MAG3110

Pro počáteční testování měření byl vybrán trojosý digitální senzor magnetické indukce (dále jen magnetometr) *MAG3110* od výrobce *Freescall Semiconductor*, který se vyznačuje nízkou spotřebou, malými rozměry a dobrou citlivostí. Komunikace s tímto senzorem probíhá po sběrnici I<sup>2</sup>C (kap. 2.4).

Důležité parametry magnetometru *MAG3110* [2]:

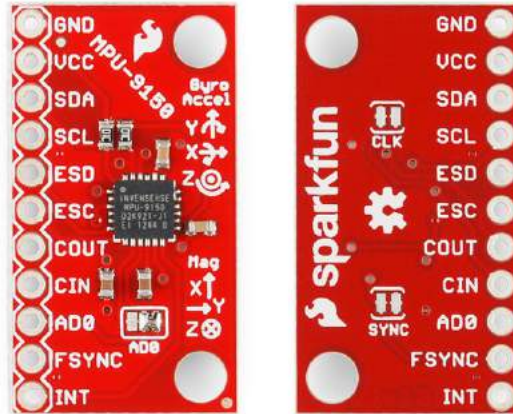
- Pouzdro součástky DFN (2x2x0,85 mm)
- Napájecí napětí(VDD) 1,95 V až 3,2 V
- Napájecí napětí logiky (VDDIO) 1,62 V až 3,2 V
- Citlivost senzoru 0.1  $\mu$ T s rozsahem od  $-1$  mT do 1 mT

### 2.3. Inerciální měřicí jednotka MPU9150

Druhá volba připadla na inerciální měřicí jednotku (dále jen IMU) *MPU9150* výrobce *Inven Sense*. Ta v sobě obsahuje mimo magnetometru i gyroskop a akcelerometr. Stejně jako u magnetometru kap.(2.2) probíhá komunikace *MPU9150* po sběrnici I<sup>2</sup>C.

Důležité parametry *MPU9150* [3]:

- Pouzdro LGA (4x4x1 mm)
- Napájecí napětí 2,375 V až 3,465 V
- Citlivost vnitřního magnetometru 0,3  $\mu$ T s rozsahem od -1,2 mT do 1,2 mT
- Možné nastavení rozsahů akcelerometru  $\pm 2$  g,  $\pm 4$  g,  $\pm 8$  g,  $\pm 16$  g
- Rozsah gyroskopu  $\pm 250$  °/s,  $\pm 500$  °/s,  $\pm 1000$  °/s,  $\pm 2000$  °/s



Obrázek 2.1: DPS s MPU9150 [4]

Zakoupená IMU jednotka přichází již napájená na DPS, kde jsou vyvedeny všechny její výstupy (obr. 2.1).

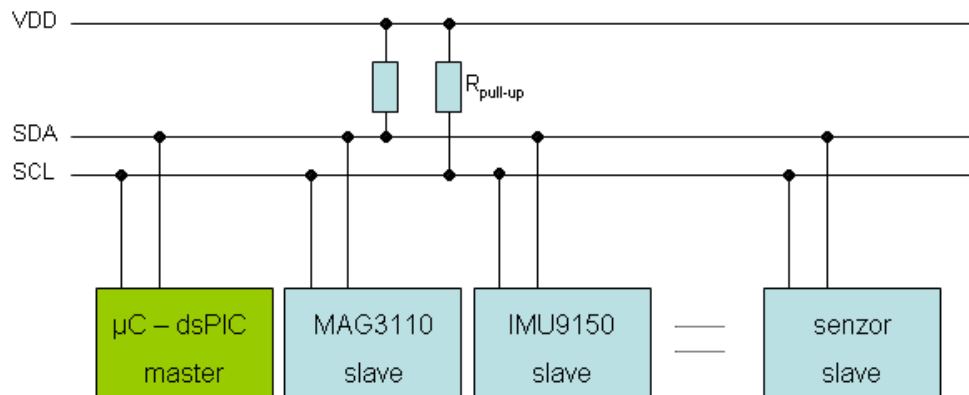
## 2.4. I<sup>2</sup>C sběrnice

I<sup>2</sup>C je sériová sběrnice vyvinutá firmou *Phillips*, která slouží pro připojení pomalých zařízení, jako jsou AD/DA převodníky, EEPROM paměti, nebo senzory [5].

Komunikace probíhá po dvou linkách, a to Serial Data Line (SDA) a Serial Clock Line (SCL). Sběrnice je typu *multimaster*, kde lze připojit několik master a slave zařízení zároveň.

Každé připojené slave zařízení (dále jen slave) má svou 7 bitovou nebo 10 bitovou adresu. Komunikace probíhá vždy s jedním zařízením, které je zaadresováno. Rozmezí taktu, na kterém sběrnice pracuje, je 100 kHz až 400 kHz, nově i do 10 MHz.

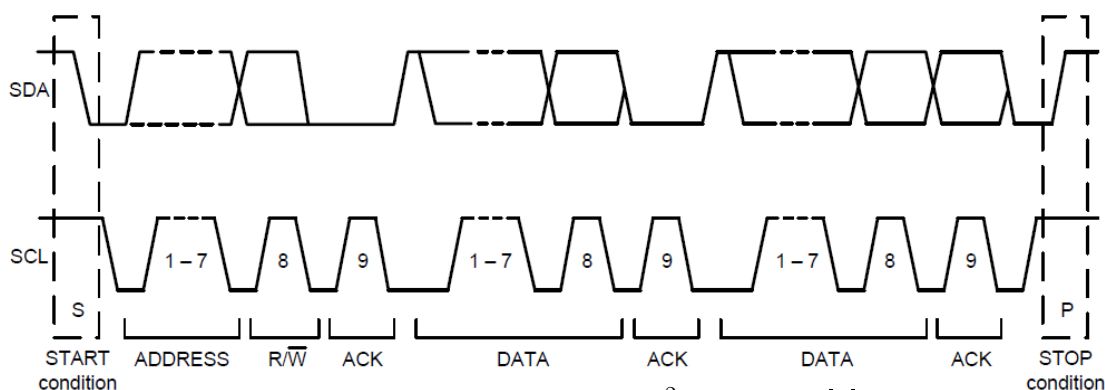
Na (obr. 2.2) je naznačeno schéma připojení zařízení na sběrnici. Zapojení je jednoduché v tom, že stačí 2 pull-up rezistory a 2 vodiče k připojení velkého množství zařízení.



Obrázek 2.2: Schéma připojení zařízení na sběrnici I<sup>2</sup>C

### 2.4.1. Komunikace prostřednictvím I<sup>2</sup>C

Na *obr. 2.3* je vyobrazen průběh komunikace po sběrnici s *MPU9150*. Po příchozí startovní podmínce (START condition) zašle master adresu (ADDRESS) slave, se kterým chce navázat komunikaci, kde její poslední bit určuje zápis nebo čtení (Read / Write). Poté se zašle potvrzující (ACK) bit a začne výměna dat (DATA). Po každém příchozím byte následuje ACK bit a započne odeslání dalšího byte. Po odeslání posledního byte a posledním ACK bitu se komunikace ukončí stop podmínkou (STOP condition).



Obrázek 2.3: komunikace po I<sup>2</sup>C sběrnici [2]

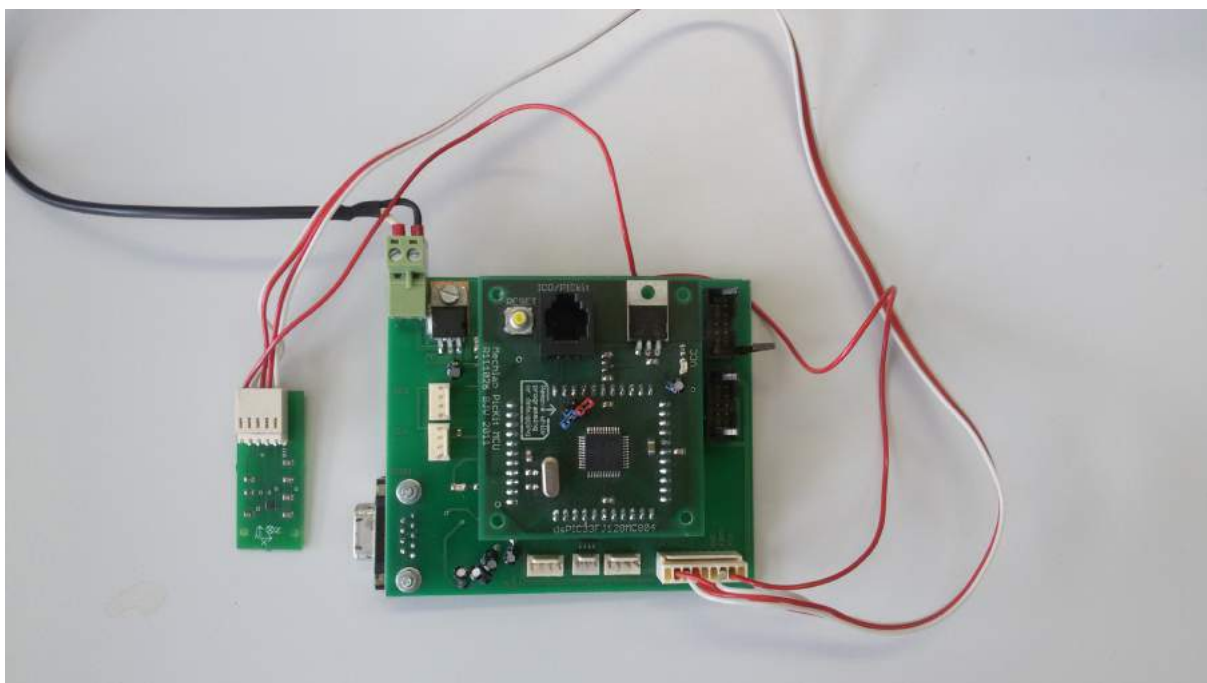
Pro správnou komunikaci se zařízením je nutné sestavit komunikační sekvenci. Pro komunikaci s různými zařízeními se tvary sekvence mohou lišit. Proto je nutné, při tvorbě sekvence, vycházet z dokumentu od výrobce (dále jen datasheet).

## 2.5. Vývojová jednotka s mikroprocesorem dsPIC

Uvedené senzory (*kap. 2.2 a 2.3*) byly testovány na vývojové jednotce s mikroprocesorem dsPIC (dále jen *EduKIT*). *EduKIT* byl určen pro prvotní seznámení s programováním mikroprocesorů dsPIC v prostředí MATLAB/Simulink včetně doplňkové knihovny MPLAB Device Blocks for Simulink (*4.1*).

Důležité parametry *EduKITu*:

- Napájecí napětí od 5 V do 12 V
- Mikroprocesor dsPIC33FJ128MC804 od výrobce *Microchip*
- Vyvedené veškeré periferie
- Konektor na programování prostřednictvím externího programátoru *PicKit3*



Obrázek 2.4: EduKIT s připojeným magnetometrem

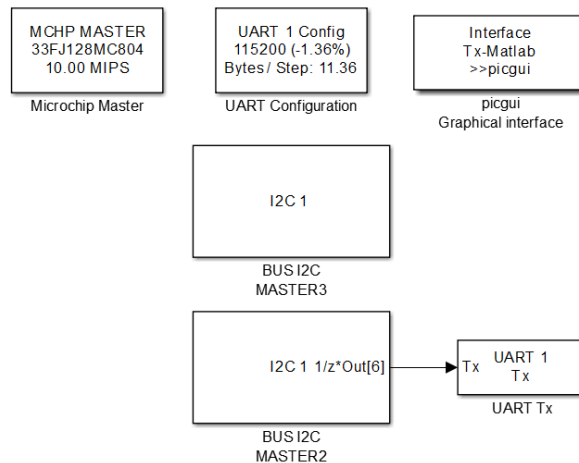
Parametry mikrokontroleru dsPIC33FJ128MC804 [6]:

- 16 bitová modifikovaná Harvardská architektura
- až 40 miliónů operací za sekundu (MIPS)
- pracovní rozsah teplot od  $-40^{\circ}\text{C}$  do  $150^{\circ}\text{C}$
- 2x UART, 1x I<sup>2</sup>C periferie, 8 PWM kanálů
- napájecí napětí 3 V až 3,6 V
- pouzdro TQFP, 44 vývodů

Senzor *MAG3110* je připojen na vývody mikroprocesoru, které obsluhují I<sup>2</sup>C komunikaci. Data jsou procesorem přečtena pomocí předepsané komunikační sekvence a následně jsou odeslána do počítače po sběrnici UART skrze převodník UART-USB.

### 2.5.1. Program pro čtení dat ze senzoru

Pro vyčtení dat ze senzoru byl vytvořen model v *Simulinku* (obr. 2.5) za pomoci setu bloků toolboxu *MPLAB Device Blocks for Simulink* [www stránka produktu]. Z tohoto modelu byl následně vygenerován kód, který se nahrál do mikrokontroleru pomocí externího programátoru *PicKit3*.



Obrázek 2.5: Simulink model pro čtení dat ze senzoru

Seznam bloků modelu a popis jejich funkce:

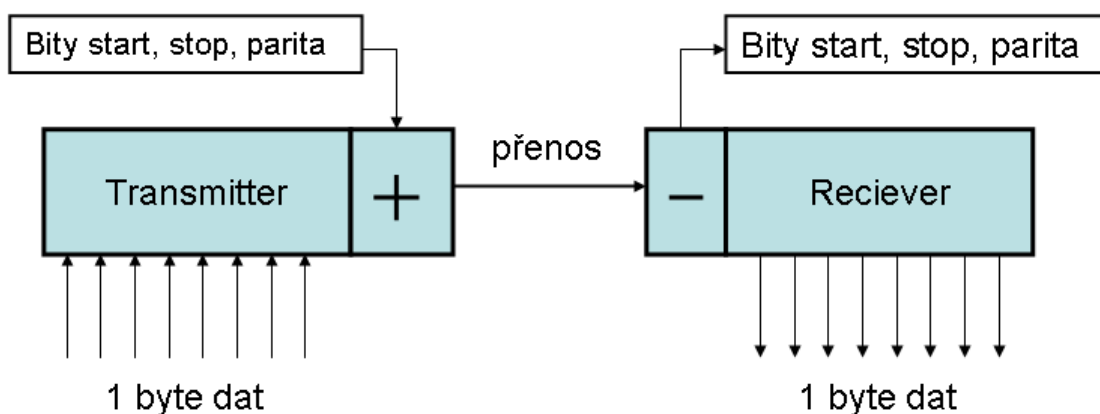
- **MICROCHIP MASTER** - blok slouží ke konfiguraci modelu pro daný mikrokontrolér, na který bude kód implemetován. Dále slouží k nastavení taktu, na kterém procesor pracuje, případně jeho předděličky a obsahuje mnoho dalších možností, které nebylo potřeba nastavovat.
- **UART Configuration** - tento blok obsahuje nastavení sběrnice UART. Zde bylo nutné nastavit piny na procesoru, které tuto sběrnici obsluhují, rychlost přenosu a velikosti kruhových bufferů.
- **BUS I2C MASTER** - tyto bloky slouží ke konfiguraci I<sup>2</sup>C sběrnice. Bylo nutné nastavit, na které piny procesoru jsou připojené linky *SDA* a *SCL*, rychlost přenosu a nakonfigurovat sekvenci komunikace po sběrnici.
- **UART Tx** - tento blok zprostředkovává odeslání naměřených dat ze senzoru do počítače po sběrnici UART. Tato data se následně uloží do struktury v Matlabu.
- **picgui Graphical interface** - poslední blok, díky kterému lze vizualizovat příchozí data bez nutnosti tvorby čtecího skriptu. Vychází ze struktury v *Matlabu*, která je uložena blokem *UART Tx*.

V prvním měření bylo zkoumáno pouze magnetické pole země za účelem ověření správnosti komunikační sekvence a nastavení konfiguračních bloků. Výsledky měření nás v tomto ujistily a bylo vše připraveno pro provedení experimentu pro srovnání uvedených senzorů. (viz. kap. 2.6).

### 2.5.2. UART sběrnice

Pro přenos naměřených dat ze senzorů do počítače je využito univerzální asynchronní sběrnice UART (Universal Asynchronous Reciever Transmitter). Při komunikaci skrze UART se k vysílaným datům připojí start a stop bit, které jsou po přijmutí zprávy odstraněny [7].

Komunikace umožňuje nastavení rychlosti přenosu dat (baudrate). Data jsou posílána po linkách RxD (Recieved Data) a TxD (Transmited data). Velmi často se komunikace prostřednictvím *UART* používá v kombinaci s některými integrovanými obvody pro převod na sběrnici USB a následné připojení do PC.



Obrázek 2.6: blokové schema sběrnice UART

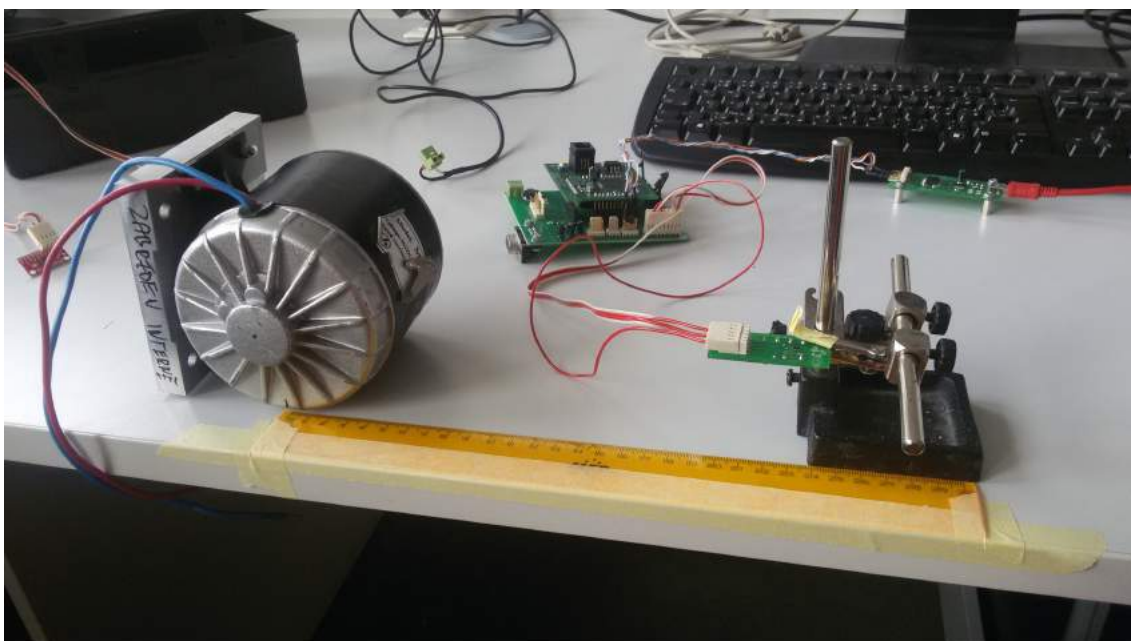
Před začátkem odeslání dat, transmitter (vysílač) připojí start, stop bity a případně i kontrolní paritní bit. Tyto bity se dále na straně recieveru (příjímač) od přenesených dat odpojí (obr. 2.6).

## 2.6. Experiment pro srovnání senzorů

Byl proveden jednoduchý experiment, jehož cílem bylo porovnat vysoce citlivý MAG3110 s méně citlivou IMU jednotkou MPU9150. Cílem bylo zjistit, zda-li je IMU jednotka schopna poskytnout smysluplná data a v jaké vzdálenosti od zkoumaného předmětu.

Na obr. 2.7 lze vidět sestavu experimentu. Skládala se ze zabržděného DC motoru jehož konstrukce byla zmagnetizována a sloužila jako zdroj statického magnetického pole. Dále z polohovatelného držáku, na kterém byl umístěn zkoumaný senzor, a z měřidla vzdálenosti, v tomto případě pravítka, pro odečítání vzdálenosti senzoru od DC motoru. Důležitým požadavkem na experiment byla jeho snadná reprodukovatelnost.

Experiment spočíval v odečítání hodnot magnetické indukce při přibližování senzoru k zabržděnému DC motoru po krocích 3 cm. Senzor byl umístěn ve výšce 5 cm nad podložkou a 4,5 cm od horizontální osy polohovatelného držáku. Dno držáku bylo opřené o pravítko sloužící jako vedení pro přibližování k DC motoru.



Obrázek 2.7: Sestava experimentu pro porovnání senzorů

### 2.6.1. Výsledky experimentu

Níže v *tab. 2.1* jsou uvedena naměřená data senzory MAG3110 a MPU9150. Rozsah vzdálenosti senzoru od motoru byl zvolen od 24 cm do 14 cm.

Tabulka 2.1: Naměřené hodnoty magnetické indukce s MAG3110 a MPU9150 senzory

x [cm]	14	15	16	17	18	19	20	21	22	23	24
<b>L [<math>\mu T</math>] MAG</b>	91,6	81,2	69,5	61,6	57,6	54,4	50,8	48,5	46,5	44,8	42,9
<b>L [<math>\mu T</math>] MPU</b>	98,8	83,9	72,1	63,6	56,4	52,2	47,9	45,8	43,4	42,1	41,2

V celém rozsahu vzdálenosti od DC motoru se naměřené hodnoty magnetické indukce lišily velmi málo. Při dalším přibližování senzorů se velikost a směr magnetické indukce rapidně měnily a již nebyly pro měření užitečné.

Pro použití v zařízení byla vybrána IMU jednotka MPU9150 ačkoliv má horší citlivost než MAG3110. V celém měřeném rozsahu vzdálenosti od zdroje magnetického pole měla uspokojivé výsledky měření. Při rozhodování převažoval fakt, že MPU9150 má integrovaný gyroskop a akcelerometr, které bylo v úmyslu později využít pro polohové řízení pohonu kyvadla.

---

## 3. Elektronika zařízení

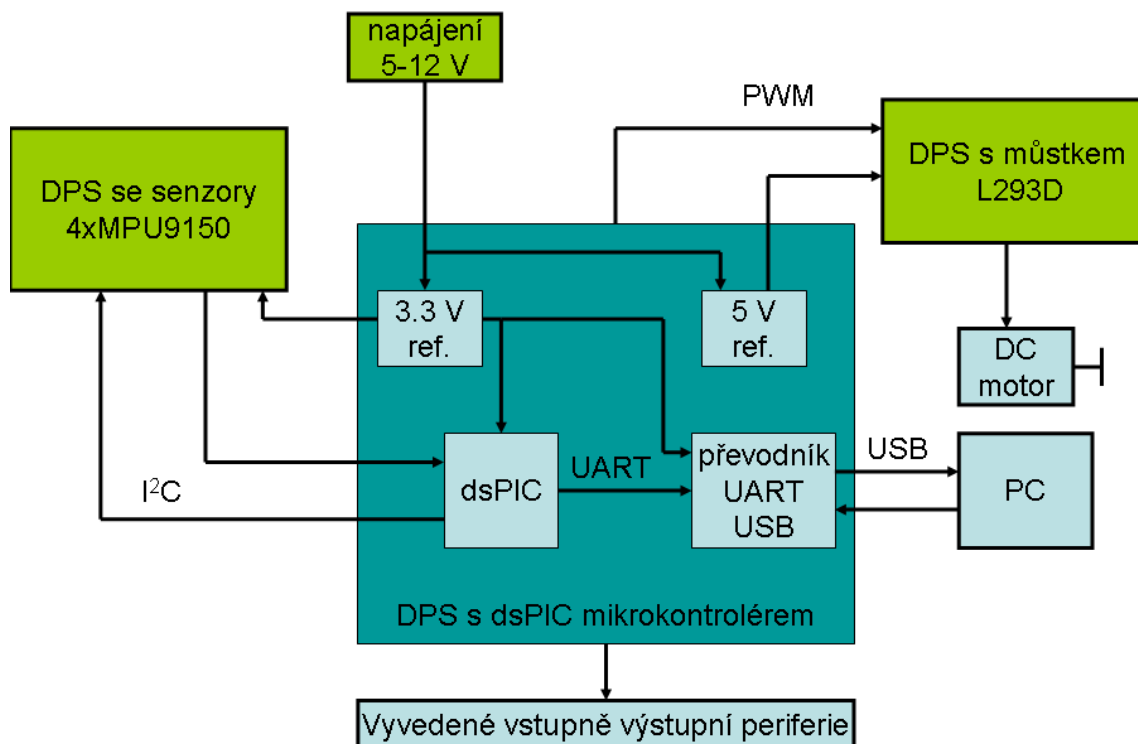
Pro požadanou funkčnost zařízení byly navrženy celkem 3 DPS. Jádro prototypu zařízení je DPS s mikrokontrolérem dsPIC firmy *Microchip*, která zprostředkovává veškerou komunikaci se senzory a s počítačem, nabízí vyvedené periferie procesoru na lištové konektory a napájí zbylé DPS. Ke správnému měření magnetické indukce v jedné ploše byla navržena druhá DPS kyvadla se čtyřmi magnetometry umístěnými v jedné ose pod sebou. DPS s magnetometry je připevněná na hřídeli stejnosměrného motoru jenž je spínán můstkem umístěným na samostatné, třetí DPS. Při návrhu byl využit CadSoft EAGLE PCB Design Software a byl kladen důraz na kompaktní rozměry všech DPS.

### 3.1. Blokové schéma zařízení

Na *obr. 3.1* je vidět blokové schéma zařízení dávající povědomí o propojení jednotlivých DPS a jejich vzájemné komunikaci.

K napájení lze využít jakýkoliv stejnosměrný zdroj s výstupním napětím v rozsahu od 5 V do 12 V Vstup napájení je situován na hlavní desce s mikroprocesorem. Po připojení napájení je napětí rozdělené do dvou napěťových regulátorů s výstupy 3,3 V a 5 V, kde první regulátor je využit pro napájení samotného mikroprocesoru a desky s magnetometry. Druhý regulátor napájí logické obvody řídicího můstku DC motoru.





Obrázek 3.1: Blokové schéma zařízení

Zařízení vyžaduje vzájemnou komunikaci vnitřních prvků. DPS s mikroprocesorem využívá sběrnice I<sup>2</sup>C ke čtení naměřených hodnot z magnetometrů. Tato data je nutné zpracovat a zaslat po sběrnici UART, přes převodník UART-USB, do počítače. Řešení s USB rozhráním je v dnešní době nejvhodnější, protože všechny osobní počítače mají k dispozici USB rozhraní, narozdíl od např. sériové linky. Dále kontrolér řídí spínací můstek DC motoru výstupním PWM signálem skrze dva digitální výstupní piny.

Pro případné rozšiřující prvky nebo pro diagnostiku zařízení slouží vyvedené vstupně výstupní piny kontroléru.

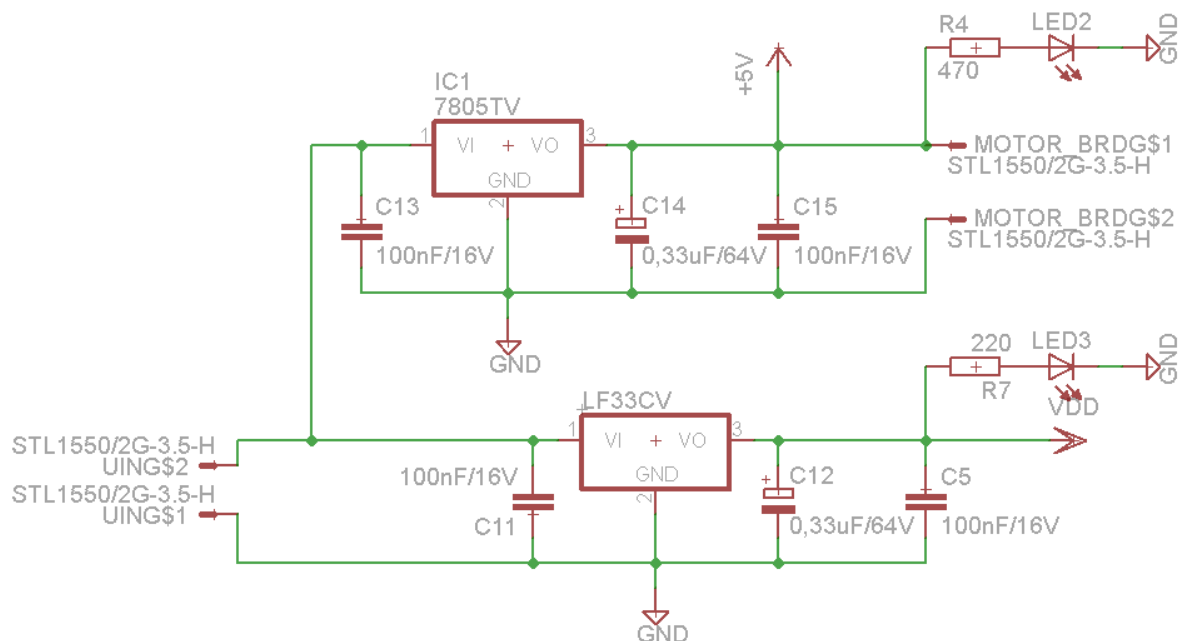
V následujících kapitolách je popsána konstrukce a funkce jednotlivých DPS.

## 3.2. DPS s mikrokontrolérem dsPIC

DPS s mikrokontrolérem dsPIC tvoří hlavní část celého zařízení. Na *obr. 3.4* je fotografie přední strany desky a na *obr. 3.5* je zadní strana desky.

Kontrolér je umístěn uprostřed desky a kolem něj jsou situovány stěžejní prvky pro funkci zařízení.

Pro napájení desky je v horní části umístěn konektor typu STL1550 pro síťový transformátor, který napájí napěťové regulátory (*obr. 3.2*) LF33CV pro 3,3 V referenci a L7805 pro 5 V referenci. Oba regulátory jsou vyrobeny v pouzdře TO220.



Obrázek 3.2: Schéma zapojení regulátorů vstupního napětí

Výstupní konektor pro napájení logických obvodů řídicího můstku DC motoru je umístěn vedle vstupního konektoru. Regulátory jsou umístěné na zadní straně desky, protože došlo k pochybení při výběru pouzdra součástky v průběhu návrhu. Na funkci tato chyba nemá žádný vliv a jde pouze o estetickou záležitost.

Regulátory byly zapojeny dle obvyklého zapojení uvedeného v datasheetech [8] a [9], kde je potřeba na vstup a výstup umístit blokovací kondenzátory. Při oživování této DPS nastalo, z důvodu zvoleného vyhlazovacího elektrolytického kondenzátoru s příliš malou kapacitou, že napájení začalo kmitat a to se projevilo do výsledků měření ze senzorů. Chyba byla identifikována pomocí osciloskopu a následně byl kondenzátor vyměněn za jiný, s vyšší kapacitou.

Dále je na DPS umístěn převodník *UART-USB* (obr. 3.3), který se skládá ze dvou integrovaných obvodů a podpůrných součástek potřebných pro jeho funkci. Integrovaný obvod ISO7242 slouží k oddělení vstupních a výstupních signálů, napájení a zemí DPS a počítače (USB rozhraní pracuje s napětím 5 V, kdežto DPS s mikrokontrolérem je napájena napětím 3,3 V). Druhý integrovaný obvod FT231X slouží jako řídicí člen komunikace, který zajišťuje viditelnost zařízení v počítači a osbluhu převodu dvou sběrnic. V zapojení jsou přítomny dvě zelené SMD LED pro signalizaci přenosu dat do počítače a z počítače.

Pro připojení k počítači byl zvolen konektor *miniUSB*.

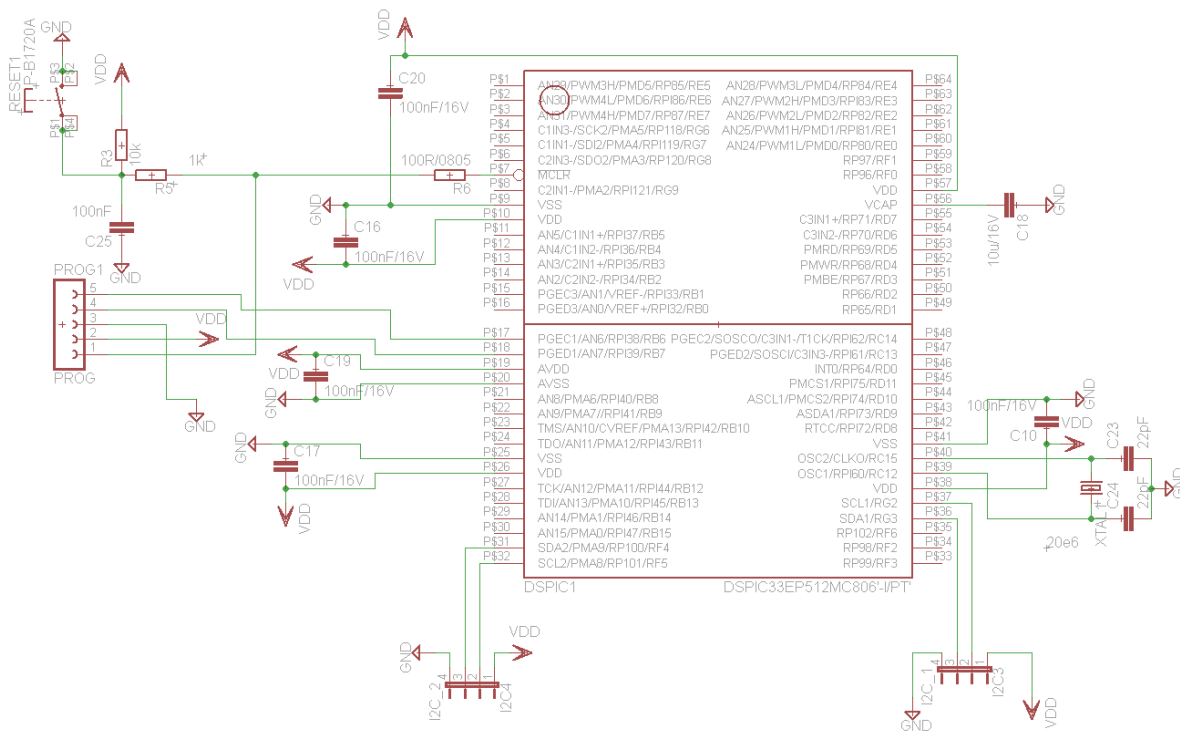




Obrázek 3.5: Zadní strana DPS s mikrokontrolérem

Při zapojování mikroprocesoru bylo potřeba striktně dodržet pravidla popsána v datasheetu kontroleru [10]. Kritické je umístění blokačních keramických kondenzátorů co nejbližší k pinům napájení procesoru. To bylo zajištěno jejich umístěním na zadní stranu desky přímo pod kontrolér a jejich připojením skrze prokvy v DPS (*viz obr. 3.5*).

Pro možnost připojení externího programátoru byla na DPS vyvedena konektorová lišta (*viz schéma 3.6*), na kterou se programátor nasadí. Dále bylo ke kontroleru zapojeno resetovací tlačítko, po jehož stisku se nahraný program v procesoru začne vykonávat od začátku. Je to důležitá věc při lazení chodu programů a při diagnostice. Stejného efektu by se dosáhlo odpojením a připojením napájení, ale to je nepraktické.



Obrázek 3.6: Schéma základního zapojení dsPIC kontroléru

### 3.3. Mikrokontrolér dsPIC 33EP512MC806

Pro zařízení bylo v úmyslu vybrat mikrokontrolér, který by měl k dispozici 2 periferie  $I^2C$  pro čtení dat ze 4 senzorů, periferii obsluhující  $UART$  sběrnici pro zasílání naměřených a zpracovaných dat do počítače a alespoň 2 kanály PWM generátoru pro řízení spínacího můstku DC motoru. Dalším, neméně důležitým kritériem, byla podpora ze strany *MPLAB Device Blocks for Microchip* toolboxu v *Simulinku*, abychom mohli využít automatického generování kódu modelu v prostředí *Simulink*.

Byl vybrán kontrolér *dsPIC 33EP512MC806*, který splňoval všechny zmíněné požadavky a nabízel dostatečnou rezervu vstupně výstupních pinů pro případná další rozšíření zařízení.

Důležité parametry mikrokontroléru dsPIC 33EP512MC806 [10]:

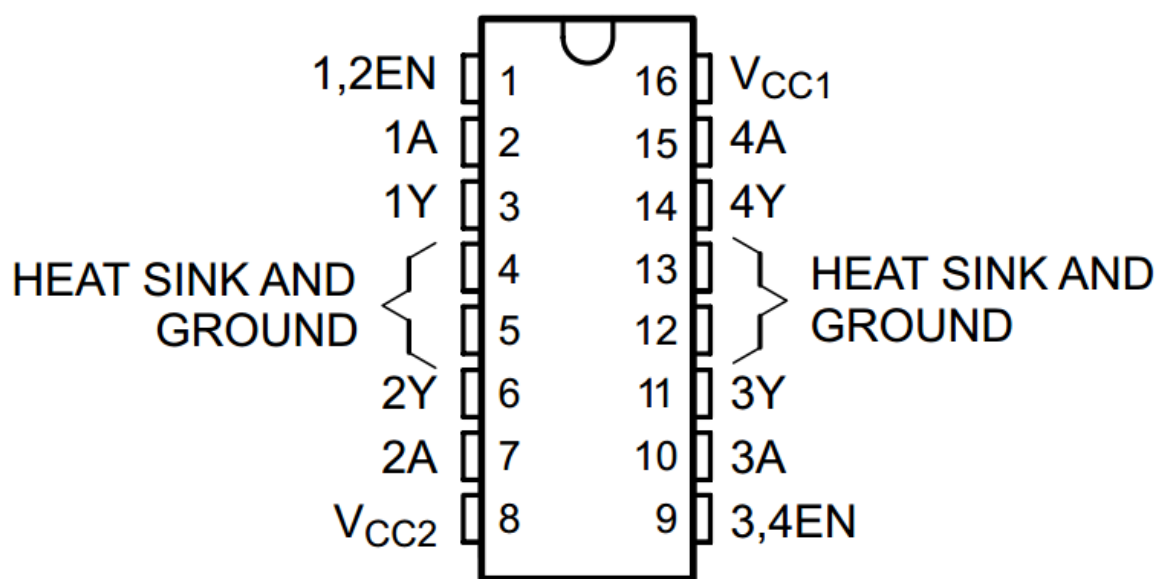
- 16 bitová modifikovaná Harvardská architektura
- až 70 MIPS
- teplotní rozsah od  $-40^{\circ}\text{C}$  do  $85^{\circ}\text{C}$  při nastavení 70 MIPS
- 4x  $UART$ , 2x  $I^2C$  periferie, 8 PWM kanálů
- Napájecí napětí 3 V až 3,6 V
- pouzdro TQFP, 64 vývodů

K procesoru se připojily vývody I<sup>2</sup>C komunikace z DPS se senzory, datové vodiče UART sběrnice se vyvedly na vstup převodníku UART-USB a všechny důležité vstupně výstupní piny byly vyvedeny na lištové konektory.

### 3.4. Řídící obvod DC motoru L293D

K účelu spínání DC motoru, jenž je použit k pohonu kyvadla, byl vybrán velmi rozšířený obvod *L293D* od výrobce *Texas Instruments*.

Jedná se o poloviční H můstek, který je schopen obousměrně ovládat DC motory při špičkovém proudu 2 A [11]. Poskytuje možnost připojení externího napájení pro motor a sám pro napájení svých logických obvodů potřebuje 5 V. Pouzdro obvodu je na obr. 3.7.



Obrázek 3.7: Pouzdro obvodu L293D (převzato z [11])

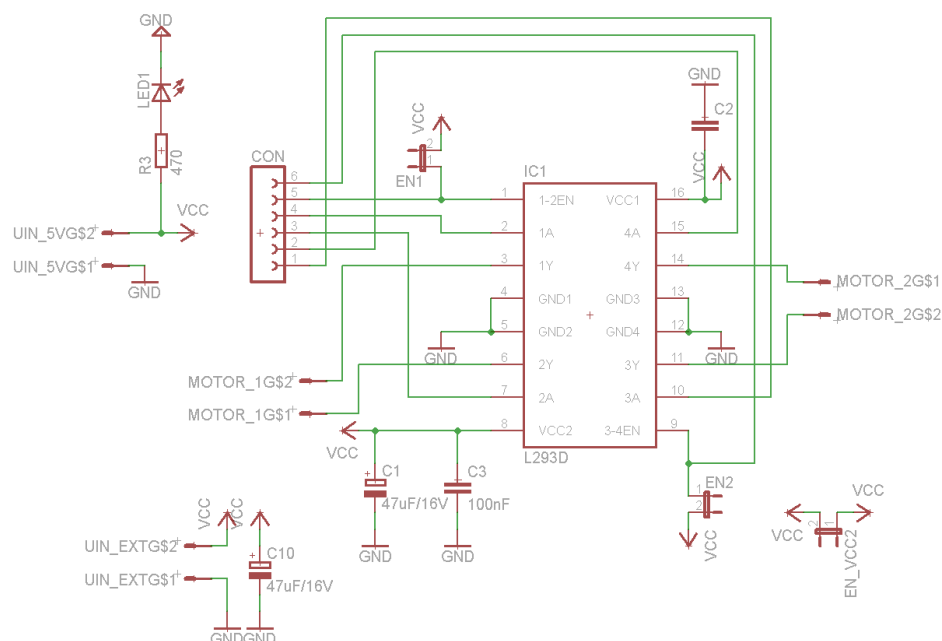
Popis funkcí jednotlivých vývodů:

- V<sub>CC1</sub> - napájení logických obvodů můstku
- V<sub>CC2</sub> - externí napájení motorů
- 1-4EN - povolení spínání motorů
- 1A, 2A - vstupy PWM signálů pro motor č. 1
- 1Y, 3Y - výstup pro připojení motoru č. 1
- 3A, 4A - vstupy PWM signálů pro motor č. 2
- 3Y, 4Y - výstup pro připojení motoru č. 2

Podstatou řízení DC motoru s tímto integrovaným obvodem je povolení spínání daného motoru přes piny 1,2 EN, vygenerování dvou PWM signálů ze strany procesoru, které se dle dané logiky mění a následné připojení těchto signálů na vstupy např. 1A a 2A pro spínání motoru č.1.

### 3.5. DPS řízení motoru

Pro můstek zmiňovaný v předchozí kapitole byla navržena samostatná DPS. Přední stranu lze vidět na *obr. 3.9*. DPS je kompaktních rozměrů a při případné poruše můstku se dá snadno odmontovat a můstek vyměnit bez nutnosti zásahu do hlavní řídicí DPS s mikroprocesorem. Schéma zapojení můstku lze vidět na *obr. 3.8*.



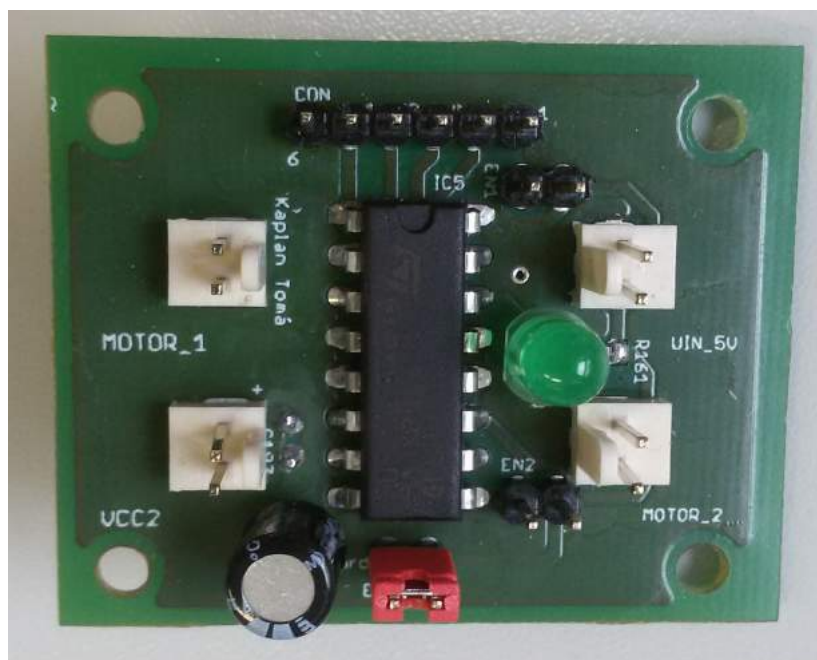
Obrázek 3.8: Schema zapojení řídicího můstku L293D

Návrh tohoto zapojení vycházel ze znalosti funkce můstku (*kap. 3.4*). Stejně jako u mikrokontroléru je potřeba dodržet co nejmenší vzdálenost blokovacích kondenzátorů od vývodů napájení součástky. Proto jsou umístěny na zadní straně desky přímo pod můstkem. Jelikož můstek spíná motor, což je indukční zátěž, jsou potřeba na napájecích vývodech i elektrolytické kondenzátory pro potlačení propadnutí napájení při proudových špičkách, které motor odebírá. Jeden z elektrolytických kondenzátorů je umístěn na zadní straně DPS. Bylo to způsobeno výběrem kondenzátoru se stejnou kapacitou, ale s jinými rozměry, než bylo plánováno. Opět jde o estetickou záležitost, která nemá na funkci žádný vliv.

Na desce jsou 4 stejné konektory se zámkem, které slouží pro připojení napájecího napětí logických obvodů můstku (*UIN\_5V*), externího napájení motorů (*UIN\_EXT*) a vyvedení dvou výstupních PWM signálů pro spínání motorů (*MOTOR\_1*, *MOTOR\_2*).

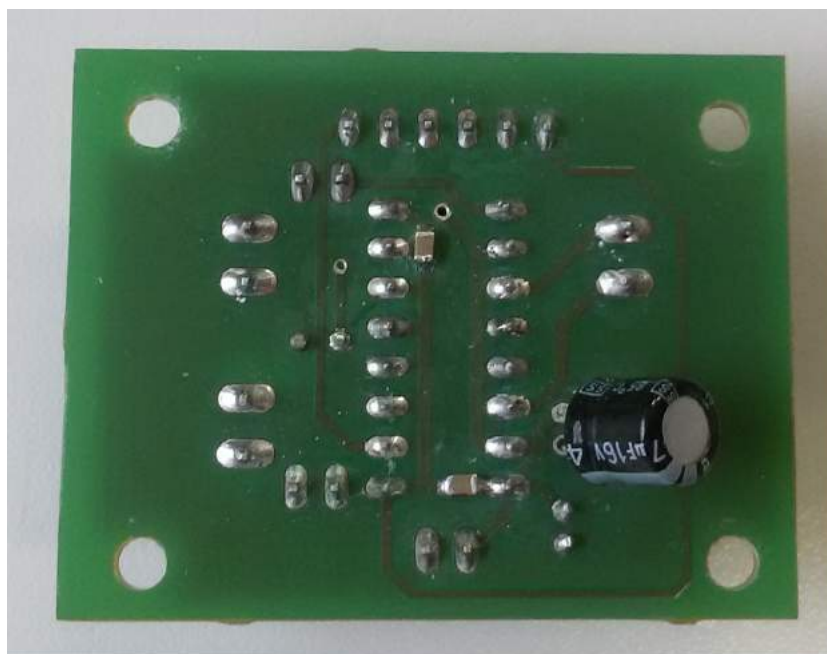
Pro volbu napájení slouží jumper konektor (*EN\_VCC2*), který v propojeném stavu přivádí 5 V z konektoru UIN\_5V na motory. V rozpojeném stavu je na motory přivedeno externí napájení. V našem případě je DC motor napájen externím napětím o velikosti 12 V.

Ovládání motorů lze povolit dvěma způsoby. Při propojení jumper konektoru (*EN\_1*) nebo (*EN\_2*) je přivedeno 5 V na vývod můstku, který povolení ovládá. Při rozpojení těchto jumperů je povolování ovládáno externě pomocí digitálních výstupů z hlavní desky s mikrokontrolérem.



Obrázek 3.9: Přední strana DPS s řídicím můstkem L293D



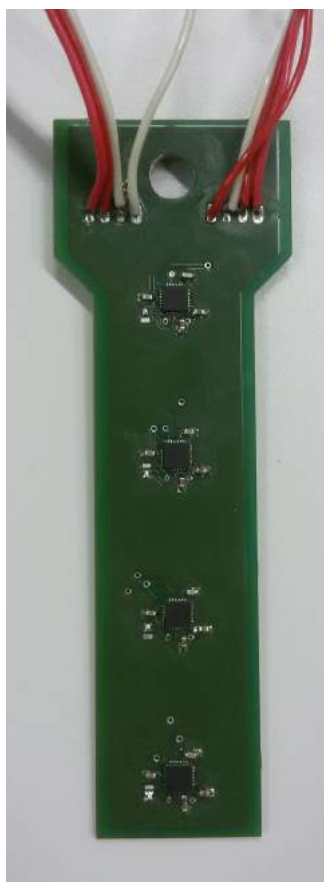


Obrázek 3.10: Zadní strana DPS s řídicím můstkem L293D

### 3.6. DPS kyvadla se senzory

K účelu měření byla navržena třetí DPS (*obr. 3.11*), na které jsou rozmístěny 4 IMU jednotky MPU9150 v jedné ose pod sebou. Rozestup mezi nimi činí 1 palec (1 palec = 2,54 cm).

Každé dvě jednotky jsou připojeny na jednu I<sup>2</sup>C sběrnici a ta je vyvedena vodiči ven k připojení na hlavní DPS s mikrokontrolérem, kde probíhá čtení naměřených dat. Vodiče jsou umístěny blízko osy rotace kyvadla, která se nachází v horní části desky (otvor pro hřídel motoru), aby nedocházelo ke zbytečnému namáhání pájených spojů, kterými jsou připojeny k desce.



Obrázek 3.11: DPS kyvadla z přední strany

Zapojení každé z IMU jednotek vychází z [3] a je vyobrazeno na *obr. 3.12*. Pro provozování sběrnice na vyšších frekvencích (cca 400 kHz) jsou potřebné pull-up rezistory o velikosti do  $2,2\text{ k}\Omega$ . Ty jsou umístěny ze zadní strany DPS blízko výstupním vodičům. Na DPS jsou přítomny také pájecí jumper plošky, které slouží k nastavení 7 bitové adresy IMU jednotky. Pokud je adresní vývod IMU jednotky připojen na napájecí napětí 3,3 V, je jeho adresa 69. Pokud je připojen na zem, má adresu 68. Je to z toho důvodu, že máme na mikrokontroleru pouze 2 periferie obsluhující komunikaci I<sup>2</sup>C.



Pro osazení pouzdra IMU jednotky na desku, které má pájecí plošky zespoda, byla použita horkovzdušná pistole. Postup pájení byl nejprve natrénován na nefunkčních součástkách se stejným pouzdem, aby se snížilo riziko zničení funkčních součástek. Osazení dále bylo považováno za jeden z velkých úspěchů při tvorbě prototypu zařízení.

---

### 3.7. DC motor

Jako pohon kyvadla byl nejprve vybrán DC motor výrobce *Magneti Marelli*, který dříve sloužil jako motor pro pohon elektrického stahování oken v automobilu.

Po několika marných pokusech řídit motor polohově se zjistilo, že má motor velké vnitřní tření rotoru a není tedy vhodný pro tuto aplikaci. Proto se zakoupil nový (*obr. 3.13*), který vyhovoval svými rozměry, napájecím napětím a průměrem hřídele.

Nový motor disponuje vysokým kroutícím momentem při maximu  $100 \text{ ot.min}^{-1}$  díky své čelní převodovce.



Obrázek 3.13: Stejnoseměrný motor pro pohon kyvadla

---

## 4. Software zařízení

Většinou část práce tvoří software, který zajišťuje požadovanou funkčnost a chování zařízení. V tomto případě bylo využito toolboxu *MPLAB Devices for Simulink*, který umožňuje automatické generování kódu z modelu v Simulinku, vyhnutí se programování kontroléru v textových jazycích a tím zrychlit celý proces vývoje.

Dalším nástrojem využitým při tvorbě software byl toolbox Simulinku Stateflow, který umožňuje tvořit stavové algoritmy potřebné pro řízení chodu zařízení a jeho ovládání z GUI v Matlabu.

### 4.1. MPLAB Device Blocks for Simulink toolbox

Jedná se o přídatnou knihovnu Simulinku, která obsahuje funkční bloky pro konfiguraci a programování 16 bitových mikrokontrolérů výrobce *Microchip* a tím umožňuje efektivní vývoj zařízení s nimi [12].

Důležité bloky knihovny:

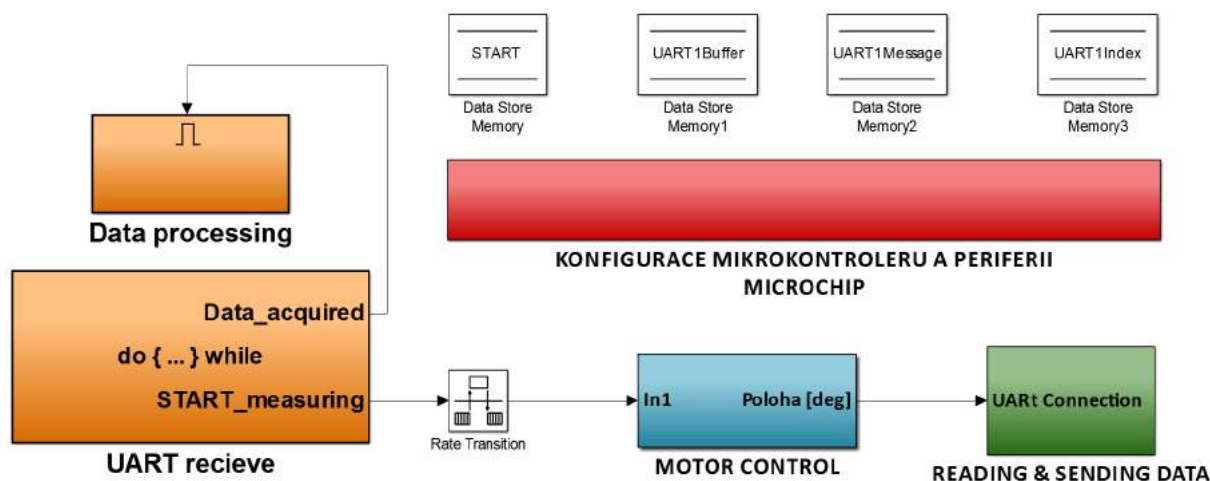
- **Microchip Master** - konfigurační blok, který je nutný pro každou vyvíjenou aplikaci. V něm probíhá nastavení typu mikrokontroleru, oscilátoru a jeho frekvence, počtu instrukcí, které se mají provést za jednu vteřinu, a mnoha dalších položek.
- **BUS I2C MASTER** - konfigurační blok sloužící pro konfiguraci I<sup>2</sup>C sběrnice. V něm probíhá sestavení komunikační sekvence se senzory, nastavení obslužných pinů periferie na mikrokontroleru a přenosová rychlost sběrnice.
- **UART Configuration** - další konfigurační blok sloužící k nastavení sběrnice UART, která je využita pro přenos naměřených a zpracovaných dat ze senzorů. Probíhá zde nastavení pinů periferie na mikrokontroléru a přenosové rychlosti sběrnice.
- **UART Rx / Tx** - dva funkční bloky reprezentující odesílání a příjem dat do mikrokontroleru nebo z něj. Do bloku UART Tx vstupují data, která jsou zaslána do počítače a z bloku UART Rx vystupují data přečtená z počítače.
- **UART Tx Matlab** - bloček sloužící k propojení Matlabu s mikrokontrolerem skrze sériovou linku. Je vhodný zejména pro diagnostiku vnitřních stavů software při vývoji, přičemž lze nastavit až 16 přenosových kanálů pro zaslání do Matlabu, kde je možnost jejich vykreslování pomocí následujícího bločku.
- **picgui Graphical interface** - tento blok slouží pro konfiguraci výše uvedeného propojení mikrokontroleru s Matlabem. Probíhá zde nastavení konkrétního portu, skrze který komunikace probíhá, přenosové rychlosti a je zde přítomna možnost vykreslení příchozích dat.
- **Output Compare HW** - bloček reprezentující vnitřní periferii komparátoru, který se v našem případě využívá pro generování PWM signálů pro řízení pohonu kyvadla. Probíhá zde nastavení vývodů mikrokontroleru a perioda výstupního signálu.

- **Data store memory** - datový blok sloužící k deklaraci globální proměnné daného datového typu. V případě zařízení jsou bloky využity k deklaraci proměnných pracujících s příchozími příkazy ze strany počítače.

Po vytvoření modelu v Simulinku za pomoci této knihovny lze použít automatické generování kódu do jazyka C, který se následně v programu *MPLAB* zkompile kompilátorem a výsledný strojový kód nahraje externím programátorem *PicKit 3* do mikrokontroleru.

## 4.2. Simulink model programu pro mikrokontroler

Největší částí softwaru je model vytvořený v Simulinku (obr. 4.1) z něhož je následně generován kód do mikrokontroleru. Program zajišťuje komunikaci se čtyřmi senzory, čtení jimi naměřených dat, dále obsluhuje zpracování naměřených dat a jejich odeslání do počítače k vykreslení a zároveň polohově řídí pohon kyvadla.



Obrázek 4.1: Model programu mikrokontroleru v simulinku rozdělený do podsystémů

Model obsahuje tyto subsystémy:

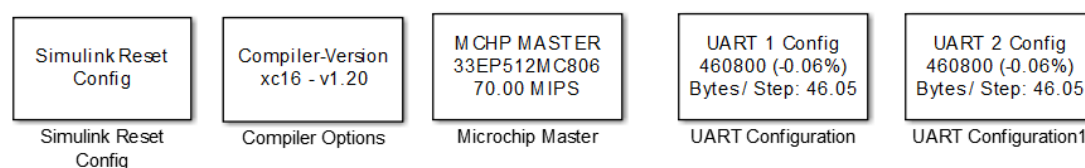
- **KONFIGURACE MIKROKONTROLERU A ...** - nejdůležitější subsystém sloužící ke konfiguraci kontroléru, kompilátoru a UART periférií.
- **MOTOR CONTROL** - subsystém provádějící polohové řízení pohonu kyvadla.
- **READING & SENDING DATA** - v tomto subsystému probíhá konfigurace komunikace se senzory, čtení naměřených dat, jejich následné zpracování a zaslání do počítače po sběrnici UART.
- **UART recieve** - subsystém, který slouží ke čtení příchozích příkazů ze strany počítače. Jím přečtená data z UART sběrnice putují do subsystému **Data processing**, který příchozí zprávu zpracuje a do globální proměnné uloží kód příkazu, který má zařízení vykonat.

---

Jelikož je model komplexní, je v Simulinku rozdělen do subsystémů, které jsou vzájemně propojeny. Popis modelu bude zaměřen na jednotlivé subsystémy.

### 4.2.1. Konfigurační subsystém

Jak již bylo zmíněno, tento subsystém (obr. 4.2) je důležitý, protože obsahuje bloky pro konfiguraci mikrokontroléru prototypu, k nastavení kompilátoru pro generování kódu z modelu a nastavení UART periférií pro přenos dat do počítače.



Obrázek 4.2: Obsah konfiguračního subsystému

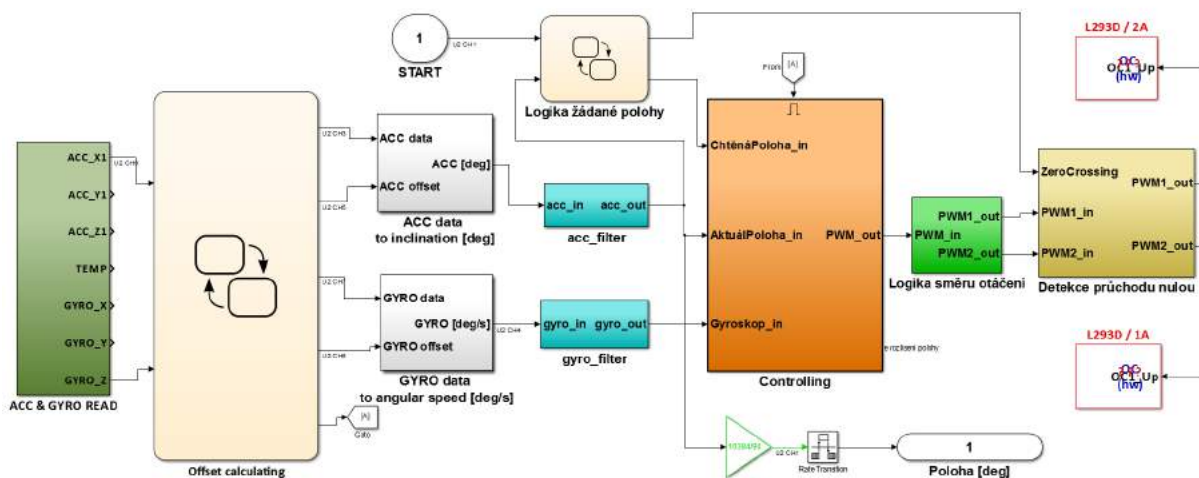
Na obrázku výše lze vidět, že Microchip master blok je nakonfigurován pro procesor dsPIC 33EP512MC806 s taktem 70 MIPS, dvě UART sběrnice s nastavenou baudrate 460800, kde UART2 sloužila pro diagnostiku při vývoji software a UART1 slouží pro komunikaci mezi zařízením a PC.

Tento subsystém má nastavenou nejvyšší prioritu, aby se potřebná konfigurace provedla jako první a až poté může následovat běh samotného programu.

### 4.2.2. Subsystém řízení motoru

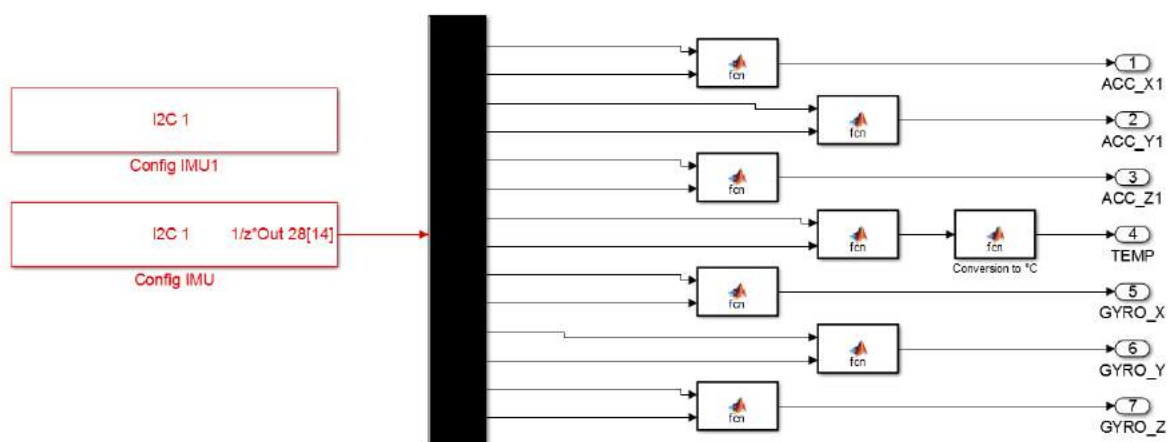
Pro uvedení kyvadla do kývavého pohybu byla vytvořena polohová regulace se zpětnou vazbou od *gyroskopu* a *akcelerometru*, které jsou součástí IMU jednotek na DPS kyvadla. Obsah subsystému lze vidět na obrázku níže.

Do zpětné vazby polohového řízení byl vybrán akcelerometr a gyroskop z IMU jednotky na druhé pozici od osy rotace kyvadla. Bylo tak učiněno proto, že naměřená data obsahovala nejmenší hladinu šumu.



Obrázek 4.3: Obsah subsystému MOTOR CONTROL

Přečtení dat z akcelerometru a gyroskopu obsluhuje další subystém ACC & GYRO READ (na obrázku výše je vybarven tmavě zelenou barvou), jehož obsah je na *obr. 4.4*.

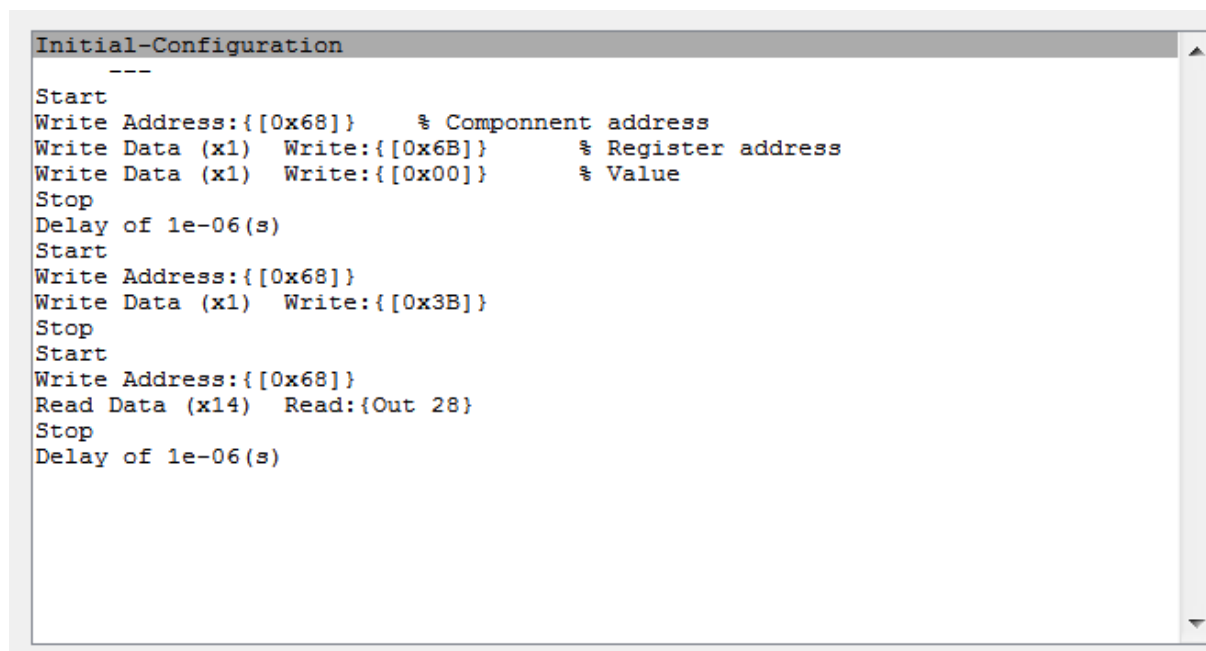


Obrázek 4.4: Model pro čtení dat z akcelerometru a gyroskopu



Nejprve bylo nutné sestavit komunikační sekvenci (obr. 4.5) pro přečtení žádaných hodnot. Struktura sekvence byla popsána v kapitole 2.4.1.

K řízení pohonu je využita osa X akcelerometru a osa Z gyroskopu. Vychází to ze souřadného systému IMU jednotky na DPS kyvadla a z toho, že potřebujeme znát úhlovou rychlost kyvadla a směr gravitačního zrychlení, které působí na kyvadlo. V krajních polohách míří gravitační zrychlení (osa x) na kyvadlo směrem dolů a tyto krajní polohy tím lze detekovat. Úhlová rychlost v průběhu rotace kyvadla směřuje ven nebo dovnitř kyvadla (osa Z) což vyplýne z použití pravidla pravé ruky.

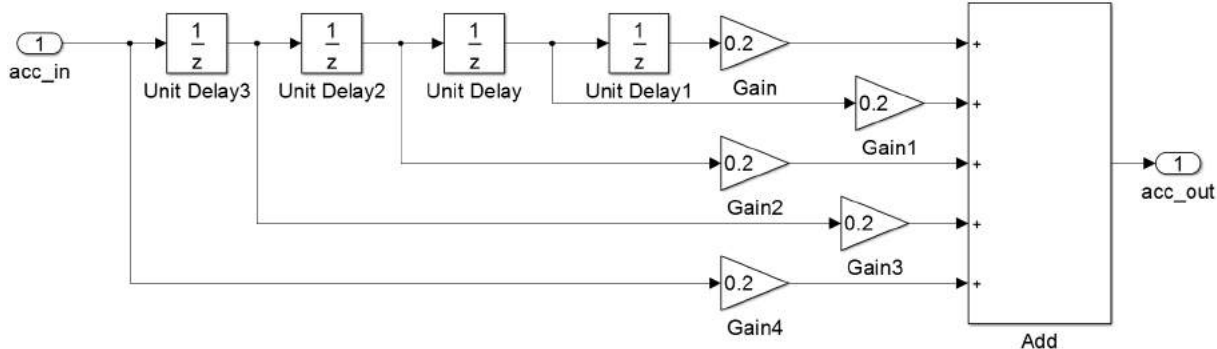


```
Initial-Configuration
---
Start
Write Address:{[0x68]}    % Component address
Write Data (x1)  Write:{[0x6B]}    % Register address
Write Data (x1)  Write:{[0x00]}    % Value
Stop
Delay of 1e-06(s)
Start
Write Address:{[0x68]}
Write Data (x1)  Write:{[0x3B]}
Stop
Start
Write Address:{[0x68]}
Read Data (x14)  Read:{Out 28}
Stop
Delay of 1e-06(s)
```

Obrázek 4.5: Komunikační sekvence pro čtení dat akcelerometru a gyroskopu

Přečtená data z akcelerometru a gyroskopu vstoupí do Stateflow funkce *Offset calculating*, kde dojde ke spočtení jejich offsetu. To je realizováno přečtením prvního sta hodnot a vypočtením jejich průměru. Vypočtený offset je nadále v blocích *ACC data to inclination* a *GYRO data to angular speed* odečten od příslušné naměřené veličiny. Výstupem prvního zmíněného bloku v tomto odstavci je již **úhlové natočení kyvadla** ve stupních a výstupem druhého bloku je **úhlová rychlost kyvadla** ve stupních za sekundu.

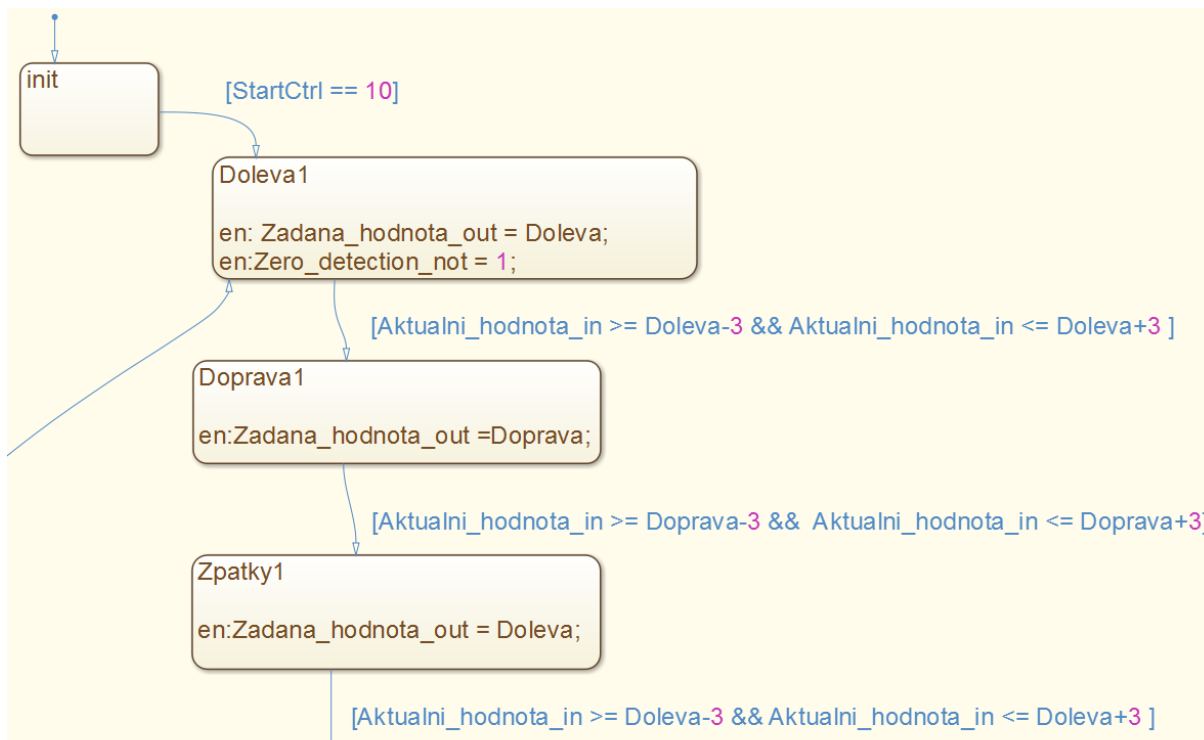
V těchto aplikacích se běžně využívá komplementárního filtru, do kterého vstupují nefiltrovaná data z akcelerometru a gyroskopu. Jelikož data z akcelerometru trpí větší hladinou šumu oproti datům z gyroskopu, přiřazuje se mu mnohem menší váha, čímž má na polohovou regulaci menší vliv než gyroskop. V našem případě byla cesta komplementárním filtrem vyzkoušena, ale výsledky nebyly uspokojivé a to z toho důvodu, že odezva řízení byla značně pomalá. Ačkoliv data z akcelerometru trpí šumem, pro naší aplikaci stačí.



Obrázek 4.6: Filtrování příchozích hodnot z akcelerometru

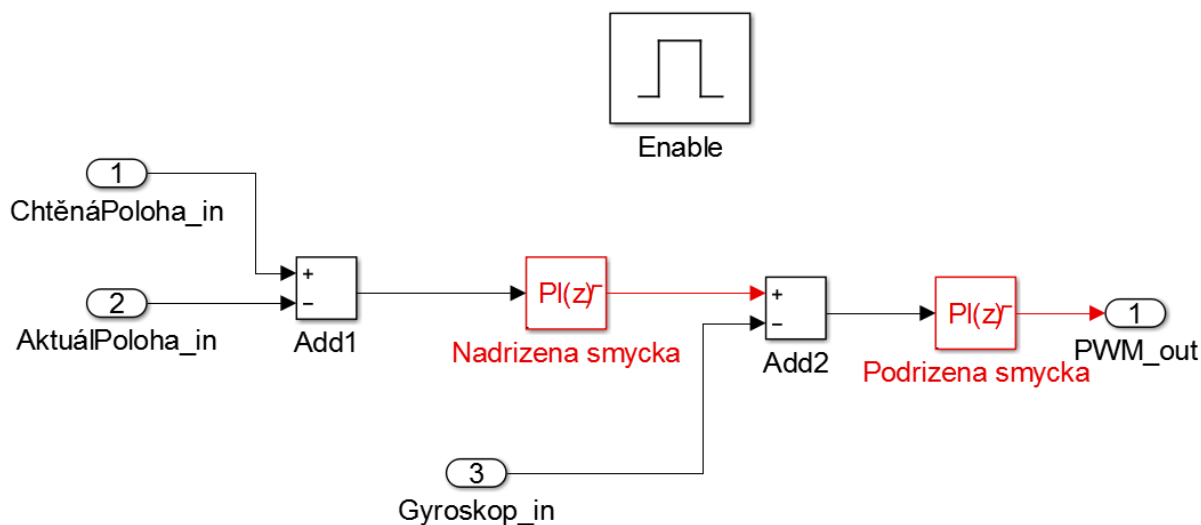
Pro zlepšení kvality výstupních dat byly vytvořeny jednoduché filtry *acc filter* a *gyro filter* (obr. 4.6), které fungují na principu spočtení průměrné hodnoty z pěti po sobě jdoucích přečtených hodnot.

Na obr. 4.7 lze vidět část Stateflow mechanismu, který po příchozí zprávě k odstartování měření přepíná žádané hodnoty polohy kyvadla. Výchozím stavem kyvadla je svisle dolů, kde se před začátkem měření spočte offset, a jakmile se zadá příkaz k měření, dostane regulátor žádanou polohu Doleva. Po tom, co se kyvadlo konstantní rychlostí přesune do této polohy, uvedená Stateflow logika žádanou polohu překllopí a kyvadlo se začne pohybovat na druhou stranu.



Obrázek 4.7: Část stateflow funkce pro ovládání žádané polohy kyvadla

Proměnná žádaná hodnota vstupuje do do subsystému *Controlling* s kaskádně řazenými diskrétními PID regulátory (obr. 4.8).

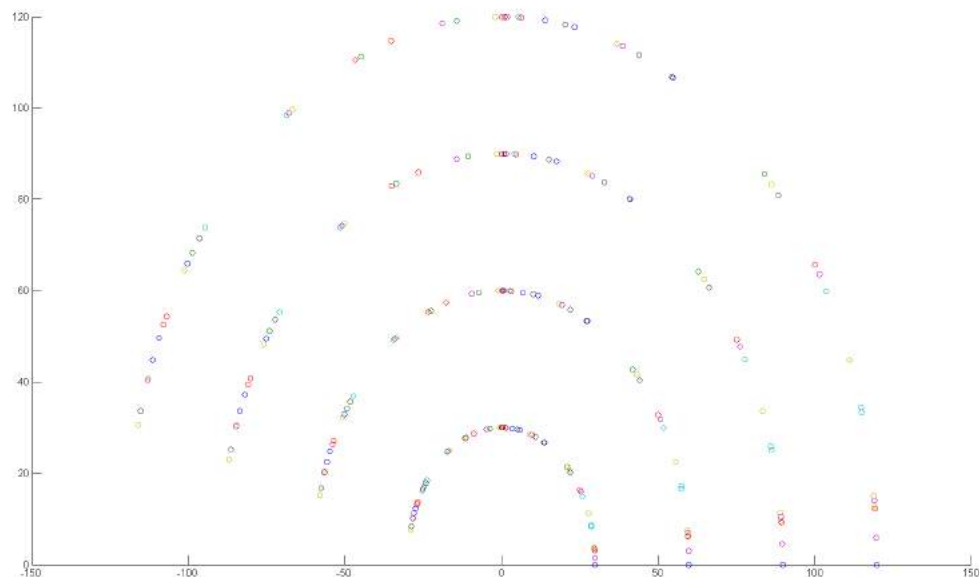


Obrázek 4.8: Diskrétní PID regulátory pro řízení polohy kyvadla

Nejprve byla vytvořena rychlostní vnitřní smyčka regulátoru a po dosažení konstantní rychlosti otáčení motoru se nad rychlostní smyčku nadřadila polohová smyčka, do níž vstupuje aktuální natočení kyvadla z výstupu akcelerometru a žádaná hodnota z výstupu zmíněné Stateflow logiky.

Výstupní signál z regulátoru je přes jednoduchou rozhodující logiku rozděleno na dva signály ovládající hardwareové periferie Output Compare. Z datasheetu spínacího můstku L293D (kap. 3.4) lze vyčíst, že pro řízení motoru oběma směry je nutné spínat řídicí vstupy můstku proti sobě. To znamená, že pro jeden směr otáčení je nutný PWM signál na jednom vstupu a na druhém logická nula a pro druhý směr otáčení naopak.

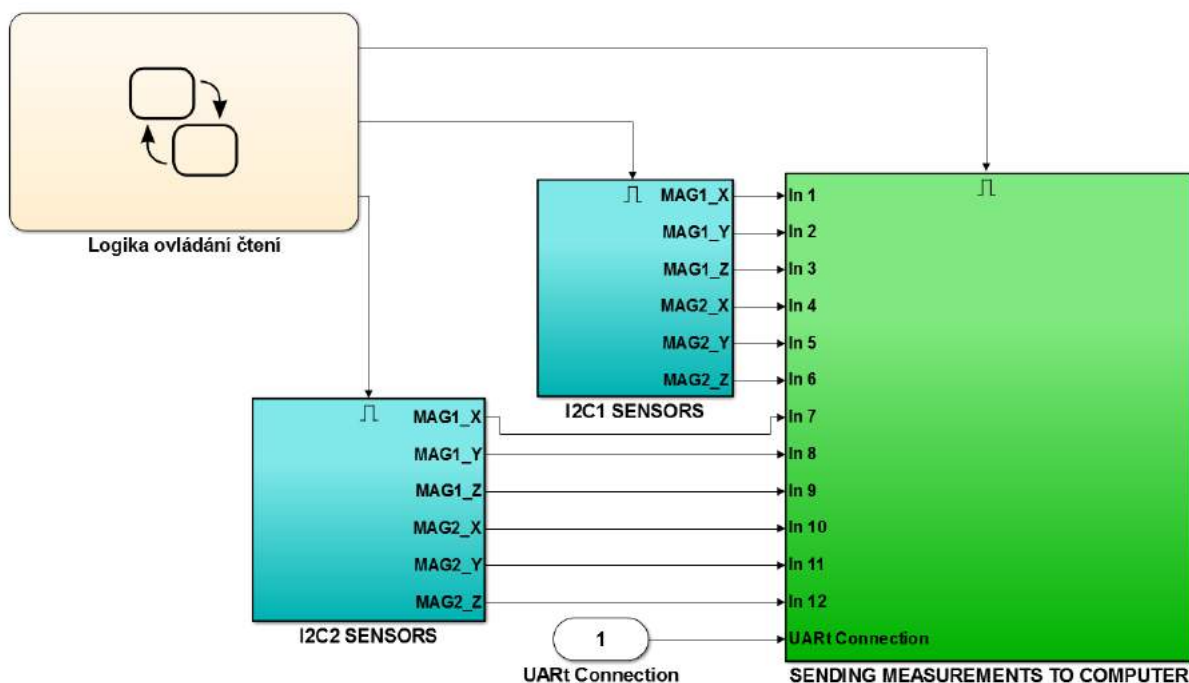
Na obrázku níže je vykreslený graf, který sloužil pro ověření funkce snímání polohy pomocí akcelerometru a gyroskopu. Každá z malých kružnic reprezentuje polohu snímače v ploše při jednom kyvu a cílem bylo vykreslit výseče kružnic, které opisují senzory při rotačním pohybu kyvadla.



Obrázek 4.9: Vykreslené polohy senzorů v průběhu kyvu kyvadla

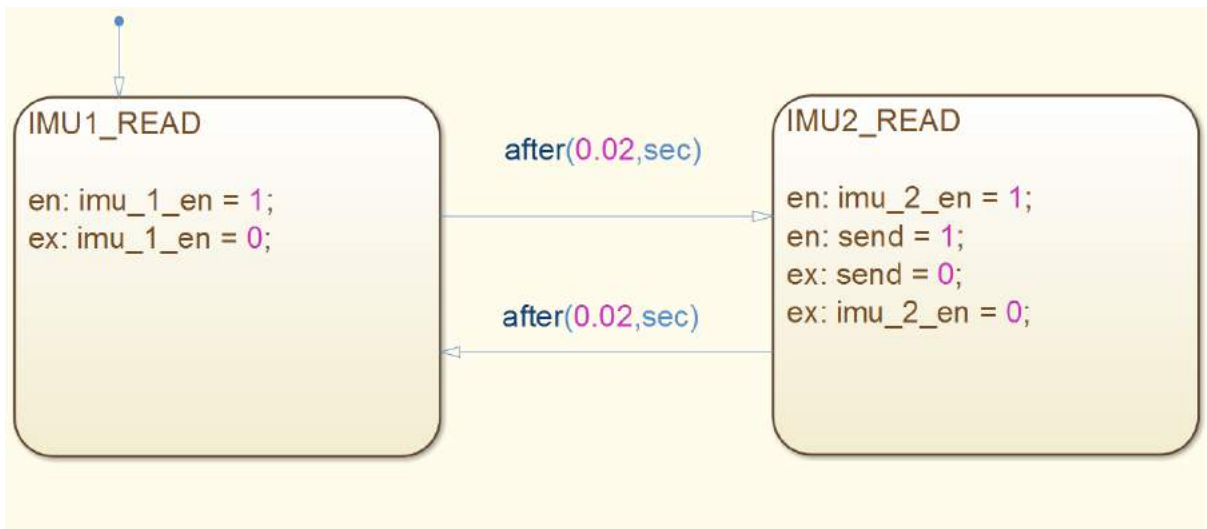
### 4.2.3. Subsystem READING & SENDING DATA

Tento subsystém obsluhuje čtení hodnot magnetické indukce z magnetometrů po sběrnici I<sup>2</sup>C a jejich následný přenos, prostřednictvím komunikačního protokolu, do osobního počítače.



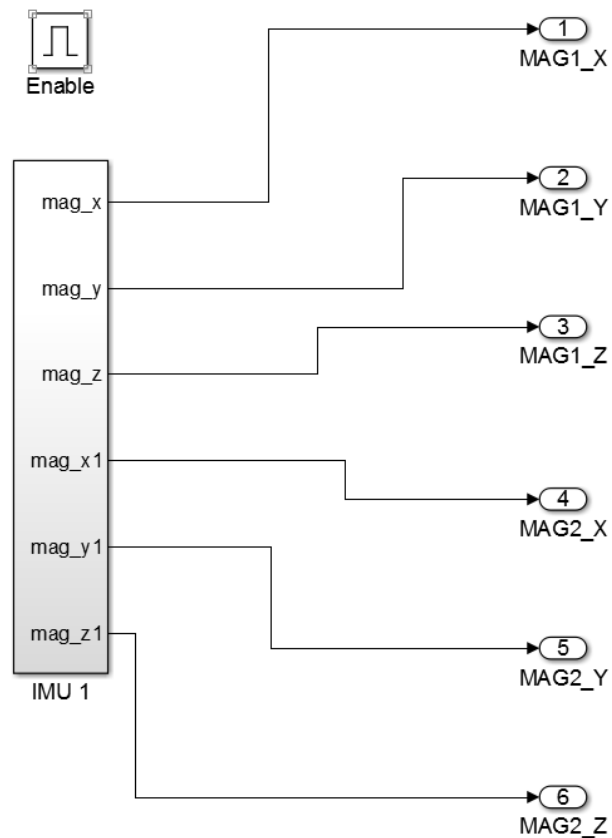
Obrázek 4.10: Subsystému uvnitř READING & SENDING DATA

Na *obr. 4.10* lze vidět rozdělení do dalších subsystémů uvnitř popisovaného subsystému. Takováto struktura je z toho důvodu, že při komunikaci se senzory docházelo k ovlivňování jedné vnitřní periferie mikrokontroleru obsluhující I<sup>2</sup>C s druhou. Chyba se projevovala nesprávně přečtenými daty, která ale měla zvláštní pravidelnost. Až zkoumání sběrnic osciloskopem ukázalo, že se komunikace zastavuje v případě, že je komunikace v chodu na obou linkách zároveň. Bylo proto tedy nutné vytvořit Stateflow logiku (*obr. 4.11*), která bude čtení z jedné dvojice senzorů provádět pouze ve chvíli, kdy bude dokončené čtení z druhé dvojice magnetometrů a následně odešle data v moment, kdy se nečte ani z jedné dvojice senzorů. Toto zvláštní chování je pravděpodobně způsobeno nedostatkem nástroje MPLAB Device Blocks for Simulink.



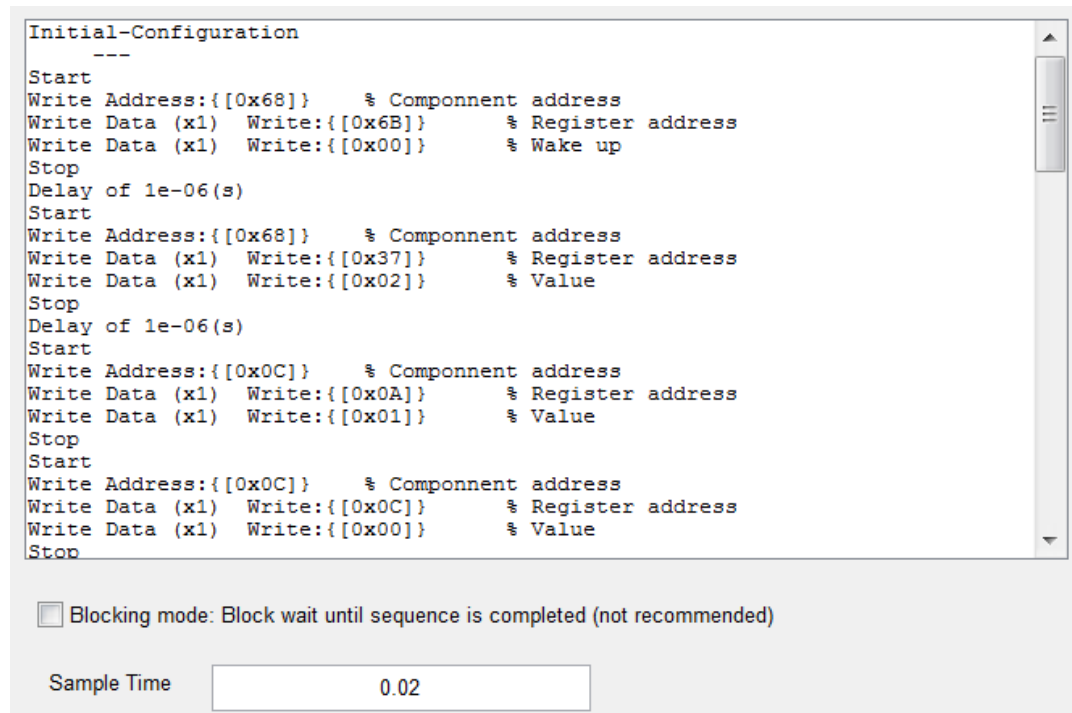
Obrázek 4.11: Stateflow logika pro kontrolu taktu čtení a zasílání dat z magnetometrů

Pro komunikaci s magnetometry na každé z I<sup>2</sup>C linek byl vytvořen model (*obr. 4.12*) (nachází se uvnitř modře podbarvených subsystémů *I2C1 sensors*, *I2C2 sensors*), který je velmi podobný s *obr. 4.4*.

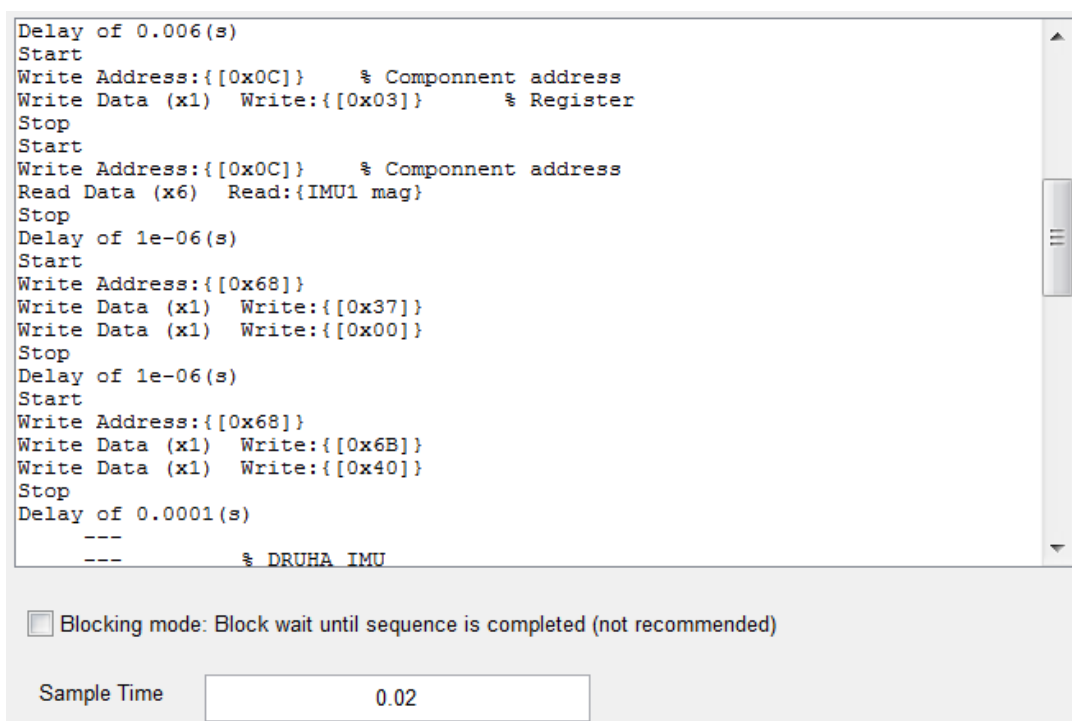


Obrázek 4.12: Model pro vyčtení naměřených hodnot z magnetometrů na jedné I2C lince

Uvnitř modelu pro čtení dat z magnetometru se nachází sekvence komunikace dvou magnetometrů na jedné I<sup>2</sup>C lince. První část komunikace je vyobrazena na *obr. 4.13* a druhá část na *obr. 4.14*.



Obrázek 4.13: První část sekvence komunikace s magnetometrem po I<sup>2</sup>C sběrnici

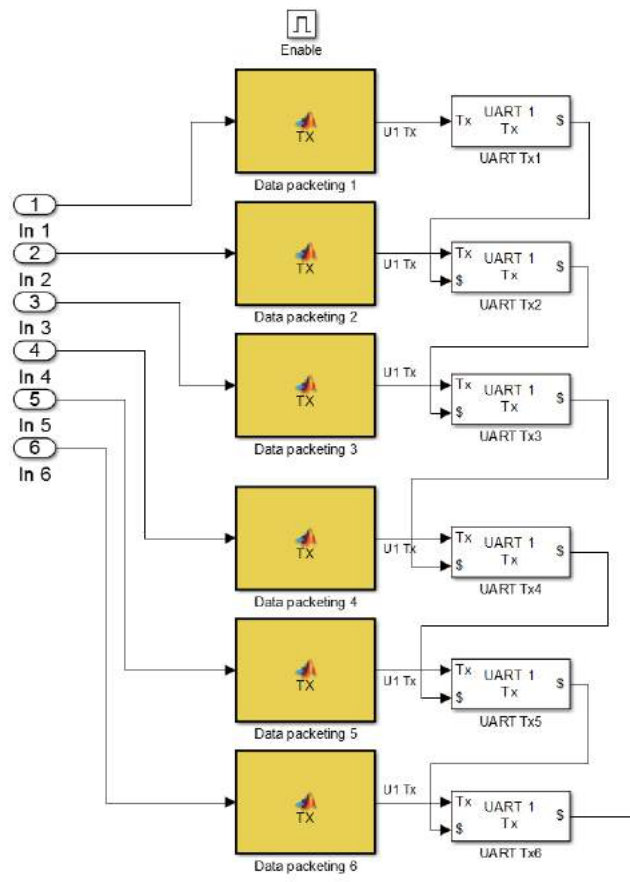


Obrázek 4.14: Druhá část sekvence komunikace s magnetometrem po I<sup>2</sup>C sběrnici

Na začátku komunikační sekvence je zaslání adresy slave zařízení (magnetometru). Po příjmu ACK bitu se zašle adresa konfiguračního registru pro nastavení režimu *bypass*,

který je nutný pro přístup k magnetometru, jelikož je umístěn na individuálním čipu v součástce. Po nastavení režimu bypass následuje další konfigurace chodu čtení. Po konfiguraci je nutné nastavit sběrnici pauzu. Není známo z jakého důvodu, ale bez ní čtení dat neprobíhalo vůbec. Po pauze se již zašle adresa datového registru, ze kterého je míněno číst a následuje přečtení naměřených dat. Ty se objeví na výstupu bloku I<sup>2</sup>C komunikace. Následuje vypnutí bypass režimu a usnutí daného, aktuálně čteného, magnetometru.

Tímto způsobem proběhne nejprve čtení prvních dvou magnetometrů, poté čtení druhé dvojice magnetometrů a následně se vykoná subsystém *SENDING MEASUREMENTS TO COMPUTER*, který je zvýrazněn zelenou barvou a polovina jeho obsahu, z důvodu velikosti, je zobrazena na *obr. 4.15*.



Obrázek 4.15: Část subsystému obsluhující zaslání dat do počítače

Úkolem výše zmíněného modelu je zaslat příchozí naměřená data ze všech čtyřech magnetometrů do počítače po sběrnici UART s využitím komunikačního protokolu.

Komunikační protokol je realizován Matlab funkcí, která se volá v Simulinku bločky *Data packeting*. Tato funkce zabalí příchozí naměřená data do zprávy dlouhé 13 bytů. Protokol na naměřená data aplikuje kontrolní součtový algoritmus *Fletcher*, který z užitečných dat vytvoří čtyři kontrolní byty, které se dále na straně počítače kontrolují, zda-li příchozí data jsou ta, co se odeslala. Tvar zprávy je v *tab. 4.1* a popis jednotlivých bytů pod ní.



Tabulka 4.1: Tvar zprávy komunikačního protokolu

35	70	50	10	D1	D2	D3	D4	F1	F2	F3	F4	36
----	----	----	----	----	----	----	----	----	----	----	----	----

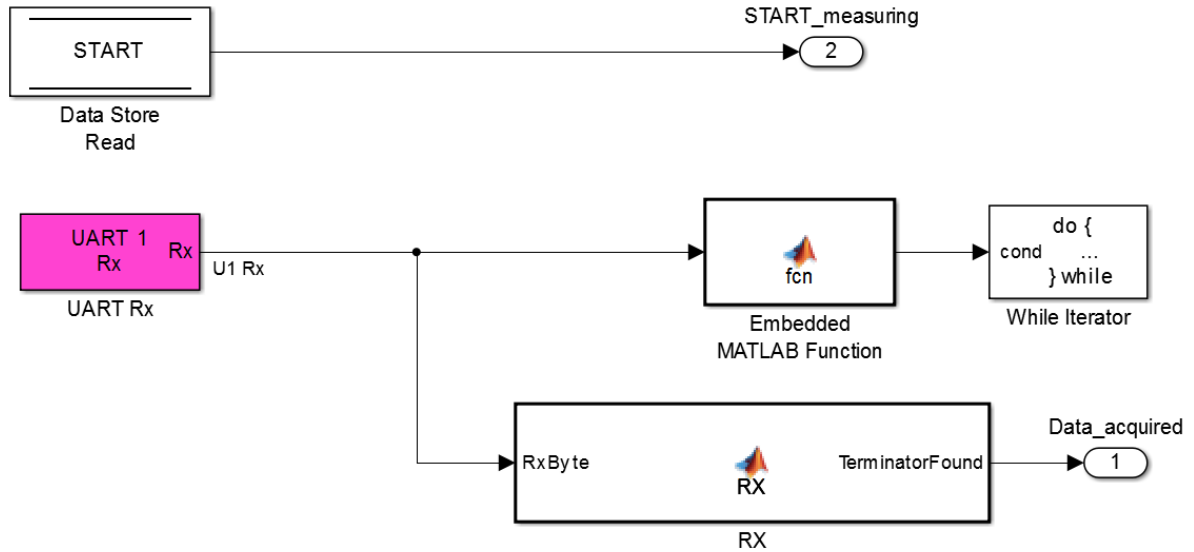
Popis bytů komunikační zprávy:

- **35** - první startovní byte s nemennou hodnotou 35. Tento byte značí začátek zprávy.
- **70 50 10** - trojice identifikačních bytů, které jsou unikátní pro každou z měřených os magnetometru. Díky těmto bytům se rozeznává, co v danou chvíli přišlo za naměřená data.
- **D1 až D4** - čtyři byty nesoucí užitečná data.
- **F1 až F4** - čtyři kontrolní byty *fletcher* algoritmu, které na straně počítače slouží pro test správnosti příchozích dat.
- **36** - poslední koncový byte značící konec zprávy.

Po zabalení příchozích naměřených dat do zpráv putují zprávy k odeslání do bloků UART Tx, které jsou řazené tak, aby se zprávy odesílaly postupně za sebou. V jednom měřícím cyklu se po sběrnici UART přenesou celkem 338 bytů, které obsahují dvě měření z magnetometrů a informaci o úhlovém natočení kyvadla.

#### 4.2.4. Subsystem UART receive

Dalším subsystemem v modelu programu pro mikrokontrolér je *UART receive*, který je na *obr. 4.1* vybarven oranžovou barvou. Slouží pro příjem příchozího příkazu ze strany počítače, na základě kterého se započne měřící cyklus, který obnáší odstartování kývání kyvadla, měření dat z magnetometrů a jejich následné zaslání do počítače k vykreslování v reálném čase. Obsah subsystemu lze vidět na *obr. 4.16*.



Obrázek 4.16: Subsystém obsluhující příjem příkazu z počítače

Příkaz k odstartování měření se provede stiskem tlačítka *START* v GUI na počítači (více v kap. 4.4). Tím se odešle po sběrnici UART zpráva stejného tvaru, jako v případě odesílání naměřených dat z magnetometrů do počítače (obr. 4.1).

Jakmile zprávu mikrokontrolér přijme bločkem *UART Rx*, uloží se do globální proměnné *UART message*. Následně se zavolá funkce *Zpracování zprávy*, která je uvnitř subsystému *Data processing*, jenž je propojen s popisovaným subsystémem. Skript je popsán v následující kapitole.

Po vykonání funkce se do další globální proměnné *START* uloží kód příkazu, který přišel ze strany počítače a na základě další logiky se odstartuje měření.

Pro obsluhu více druhů příchozích příkazů stačí upravit funkci pro zpracování zprávy, kde se na základě identifikačních bytů příchozí zprávy uloží výstup do jiné proměnné.

### 4.3. Skripty a funkce na straně počítače

Pro požadované zpracování příchozích zpráv a jejich vykreslení bylo vytvořeno několik skriptů jejichž popis je v následujících kapitolách. Bylo dále vytvořeno grafické uživatelské rozhraní pro ovládání měřicího cyklu a pro zobrazení vykreslených výsledků. Bylo využito přehledného prostředí Matlab, které je vhodné pro tvorbu programové obsluhy pro vyvíjený prototyp na úkor rychlosti běhu programu.

Je nutné říci, že veškeré skripty jsou umístěny na CD nosiči připojeném jako příloha.

### 4.3.1. Hlavní skript obsluhující chod měření

Celá sekvence příkazů pro čtení, zpracování a vykreslení příchozích naměřených hodnot je realizována jedním skriptem *COM\_PORT\_GUI\_READING*, ze kterého se volají skripty další. Toto řešení je výhodné v tom, že celý kód není umístěn v jednom skriptu, je přehlednější a případné modifikace jsou snadněji realizovatelné.

Prvním úkolem tohoto skriptu je zaslání zprávy mikrokontroléru pro začátek měření. To se stane ihned po tom, co se v GUI stiskne tlačítko *START*. Kód pro zaslání příkazu *START* vypadá následovně:

```
1 %% Zaslani prikazu START mikrokontroleru
2 Packet = int16(10)
3 StartMessage = Packeting(Packet)
4
5 %% Vytvoreni objektu serioveho portu
6 %% Konfigurace baudrate
7 SerialPort1 = serial(port, 'Baudrate',460800, 'terminator', 36);
8
9 %% otevreni serioveho portu
10 fopen(SerialPort1);
11
12 %% zaslani 3 identickych zprav pro Start mereni
13 fwrite(SerialPort1,StartMessage);
14 fwrite(SerialPort1,StartMessage);
15 fwrite(SerialPort1,StartMessage);
16
17 %% uzavreni serioveho portu
18 fclose(SerialPort1)
```

Listing 4.1: Zaslání příkazu pro start měřícího cyklu

Příkaz ke startu programu je zaslán třikrát za sebou, aby bylo zajištěno spolehlivého přečtení příkazu. Mohlo by se stát, že by program v mikrokontroléru nezareagoval na zprávu a nezačal měřící cyklus.

Po odstartování měřícího cyklu se otevře čtecí sériový port, stejně nakonfigurovaný jako ve výše uvedeném případě, a započne čtení příchozích zpráv. Příchozí zprávy se uloží do proměnné *Packets\_in* ve formě sloupcového vektoru. Část kódu, která zařizuje tuto funkci je zobrazena níže.

```

1 %% cteci smycka while o definovanem poctu mereni
2 while(i<PocetMereni+1)
3
4 SerialPort = serial(port, 'Baudrate',460800, 'terminator', 36);
5 SerialPort.InputBufferSize = 1024000;
6
7 %%otevreni seriového portu
8 fopen(SerialPort);
9 %% precist balik 13 bytu dlohou
10 Packets_in = fread(SerialPort,[338], 'uint8');
11
12 %% sorting & testing of incoming data packages
13 Sorted_messages = Sorting_Testing_function338(Packets_in);

```

Listing 4.2: část kódu skriptu pro čtení příchozích zpráv

Jakmile jsou data přečtena, je nutné je zpracovat a získat z nich užitečná data. Prvním krokem je volání funkce *Sorting\_Testing\_function338*, která má za úkol roztrždit příchozí data na základě jejich startovních, identifikačních a koncových bytů. Výstupem této funkce je matice rozměrů 13x13, kde 12 řádků zpráv jsou data z magnetometrů a jeden řádek je zpráva nesoucí informaci o úhlovém natočení kyvadla.

Následně jsou příchozí, již roztržené zprávy, zpracovány skriptem, který je popsán v *kap. 4.3.2*. Výstupem tohoto skriptu jsou již čistě naměřené hodnoty ze všech tří os každého magnetometru a informace o úhlovém natočení kyvadla. Tato data jsou připravena k vykreslení.

Vykreslení probíhá ihned po přečtení a přijmutí naměřených dat. Nejprve se vykreslí okamžité polohy magnetometrů, které jsou reprezentovány malými kružnicemi a následně se tyto polohy využijí jako počátky pro vykreslení 3D vektorů, které mají složky X, Y a Z. Pro vykreslování bodů polohy magnetometrů je využita Matlab funkce *scatter* a pro vykreslení vektorů *quiver3*. Část kódu, osluhující vykreslování vektoru magnetické indukce z jednoho senzoru, je uvedena níže.

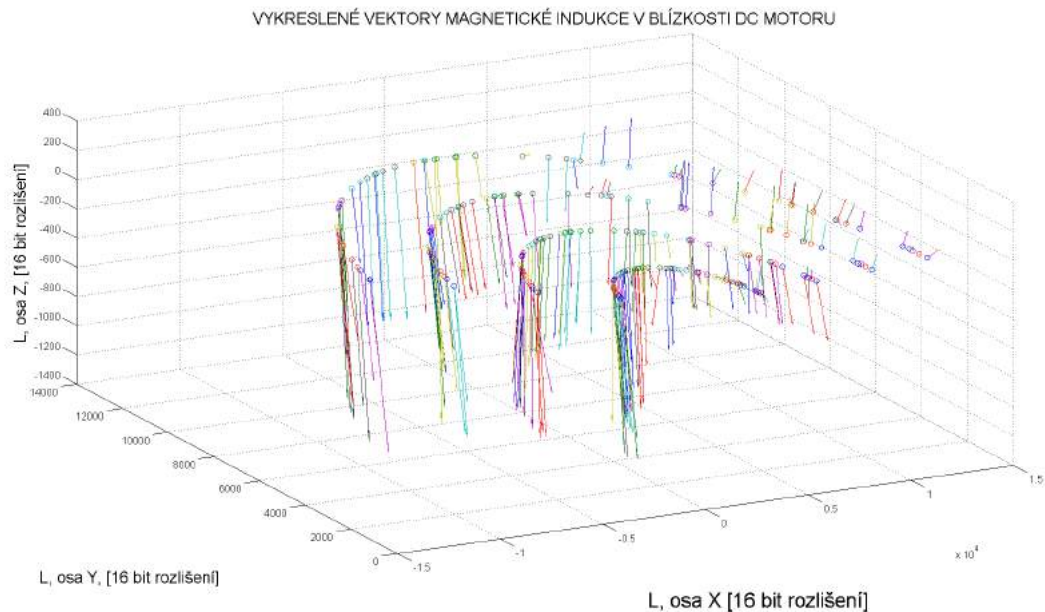
```

1 %vypocet souradnic polohy kyvadla
2 x_buffer = [3000*sind(Fi_natoceni) 6000*sind(Fi_natoceni) 9000*sind(
   Fi_natoceni) 12000*sind(Fi_natoceni)];
3
4
5 y_buffer = [3000*cosd(Fi_natoceni) 6000*cosd(Fi_natoceni) 9000*cosd(
   Fi_natoceni) 12000*cosd(Fi_natoceni)];
6
7 %vykresleni poloh senzoru
8 scatter3(x_buffer, y_buffer, [0 0 0 0])
9
10 %test, zda-li nejsou data nulova
11 if(A(1) == 1)
12 hold on
13 % vykresleni vektoru magneticke indukce
14 quiver3(x_buffer(1), y_buffer(1), 0, ((Results_buffer(1,1)-offset_1x)), ((
   Results_buffer(1,2)-offset_1y)), ((Results_buffer(1,3)-offset_1z)), 0)

```

Listing 4.3: část kódu skriptu obsluhující vykreslování vektorů magnetické indukce

Na *obr. 4.17* je vyobrazen výsledek vykreslení vektorů magnetické indukce v blízkosti zmagnetizované konstrukce DC motoru. Lze na něm vidět, že vektory míří stejným směrem a mají stejnou velikost. Pouze v pravé části vykresleného obrazu se vektory stáčí na opačnou stranu. To indikuje změnu magnetického pole v daném místě.



Obrázek 4.17: Vykreslené vektory magnetické indukce v blízkosti zmagnetizované konstrukce DC motoru

### 4.3.2. Skript pro zpracování příchozích zpráv

Tento skript slouží ke zpracování příchozích přečtených zpráv od mikrokontroleru po komunikačním protokolu sběrnice. V prototypu je využit jak na straně počítače, tak i na straně mikrokontroleru. V počítači slouží ke zpracování příchozích zpráv s naměřenými hodnotami z IMU jednotek kyvadla. Na straně mikrokontroleru zpracovává příchozí příkazy z počítače.

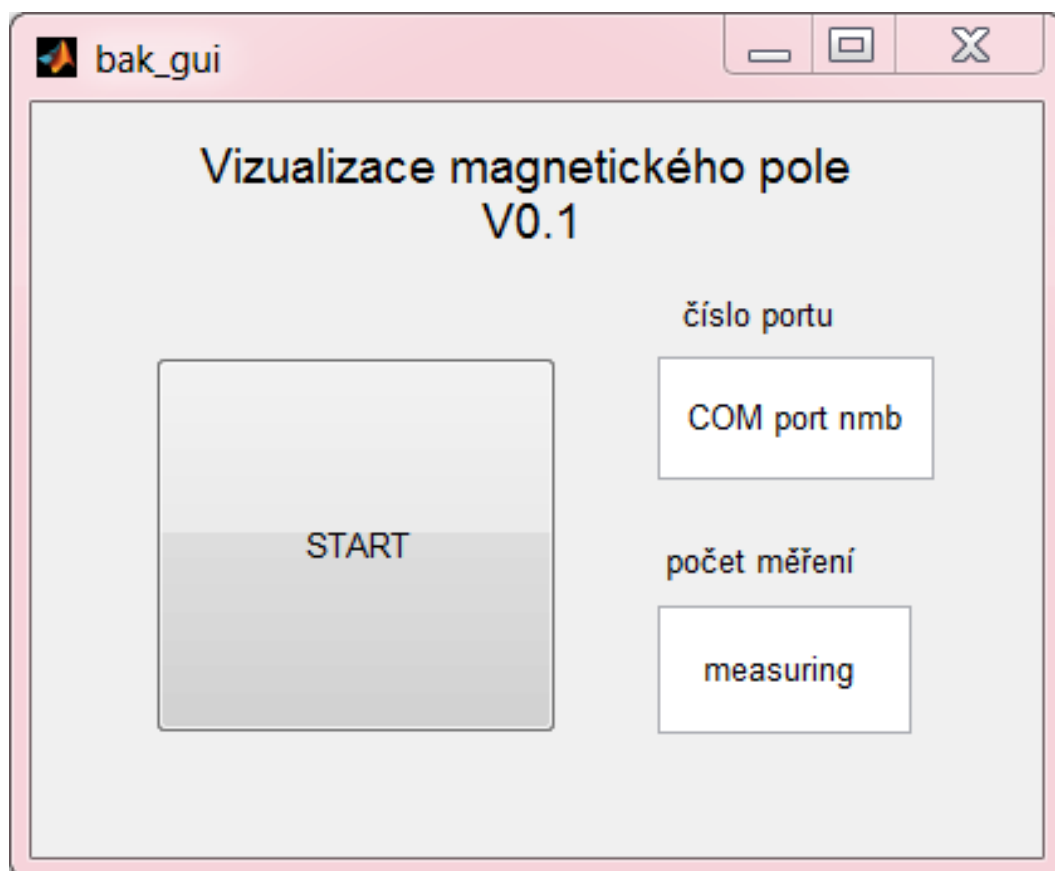
Níže lze vidět část kódu funkce zpracování zprávy. Z příchozí zprávy se vyextrahují data a kontrolní Fletcherovy byty. Po kontrole správnosti příchozích dat pomocí zmíněných Fletcherových bytů se z identifikačních bytů spočítá ID zprávy, které se následně *switch-case* logikou testuje, zda-li jsou příchozí data naměřené hodnoty nějaké osy magnetometru nebo úhlová poloha kyvadla. Výsledky se uloží do odpovídajících globálních proměnných, aby k nim byl přístup z ostatních skriptů.

```
1 function ZpracovaniZpravy()
2
3 global UART1Message;
4 global UART_OSA_X;
5 global UART_OSA_Y;
6 global UART_OSA_Z;
7 global POLOHA;
8
9 %% vyjmuti identifikacnich bytu ze zpravy
10 CommandID = int16(UART1Message(2:4));
11
12 %% vyjmuti dat a Fletcher hodnot ze zpravy
13 DATAH = ASCII2Dec(UART1Message(5), UART1Message(6));
14 DATAL = ASCII2Dec(UART1Message(7), UART1Message(8));
15 FCH1 = ASCII2Dec(UART1Message(9), UART1Message(10));
16 FCH2 = ASCII2Dec(UART1Message(11), UART1Message(12));
17
18 %% checksum – 16bit Fletchers checksum
19 [CH1, CH2] = Fletcher([DATAH, DATAL], 1, 2);
20 ID = command_come(CommandID);
21
22 %% kontrola Fletcher algoritmem
23 if ((FCH1 == CH1) && (FCH2 == CH2))
24
25     switch ID
26     case 390 % osaX
27         UART_OSA_X = typecast([DATAL DATAH], 'int16');
28     case 460 % osaY
29         UART_OSA_Y = typecast([DATAL DATAH], 'int16');
30     case 530 % osaZ
31         UART_OSA_Z = typecast([DATAL DATAH], 'int16');
32     case 210
33         POLOHA = typecast([DATAL DATAH], 'int16');
34     end
35 end
36 end
```

Listing 4.4: Část skriptu zpracování zprávy

## 4.4. Grafické uživatelské rozhraní

K ovládání měření z počítače bylo vytvořeno primitivní grafické uživatelské rozhraní, které obsahuje jedno tlačítko START a dvě vstupní pole. GUI je vyobrazeno na *obr. 4.18*



Obrázek 4.18: Primitivní grafické uživatelské rozhraní

Do vstupních polí uživatel zadá číslo portu, ke kterému je v počítači připojen náš prototyp a dále počet měření.

Po odstartování měření tlačítkem START se otevře individuální okno, kde se postupně vykreslují vektory magnetické indukce v průběhu kývání.

---

## 4.5. Experiment s palivovou pumpou

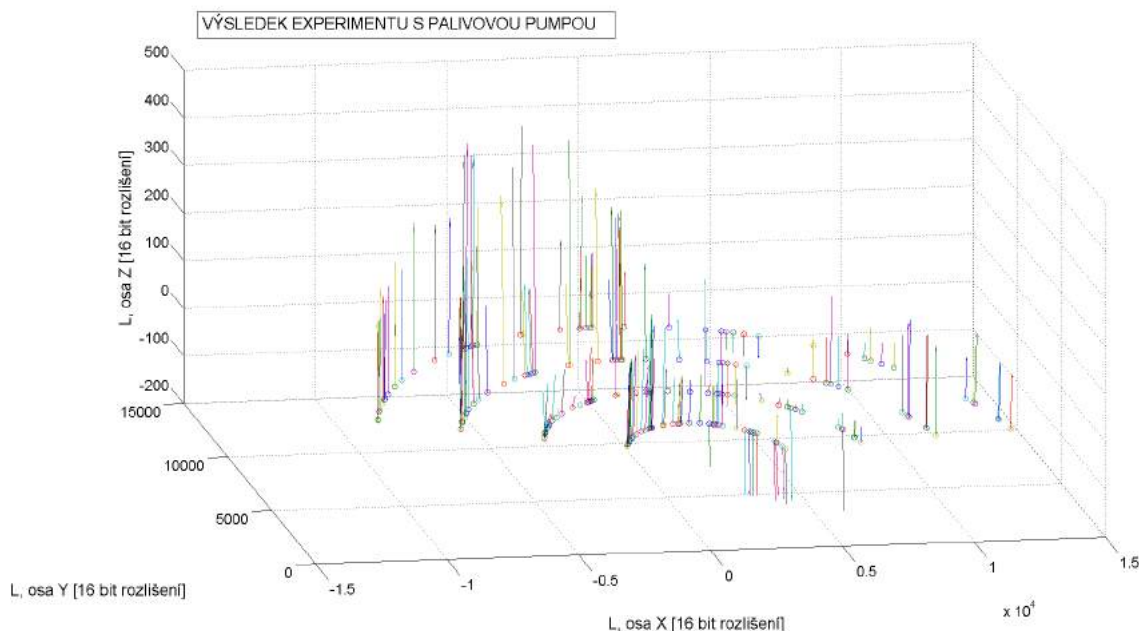
K ověření funkčnosti prototypu byl proveden experiment měření magnetické indukce v blízkosti palivové pumpy, kterou lze vidět na *obr. 4.19*.



Obrázek 4.19: Palivová pumpa jakožto zkoumaný předmět experimentu

Experiment spočíval v umístění palivové pumpy do blízkosti měřicího zařízení, spuštění měření a sepnutí palivové pumpy ve chvíli, kdy projde kyvadlo svislou referenční pozicí. Poté, co začal vinutím pohonu palivové pumpy procházet proud, začalo zařízení zaznamenávat změny v magnetickém poli. Změny se projevily ve velikosti a směru vykreslených vektorů magnetické indukce. Výsledek experimentu lze vidět na obrázku níže.





Obrázek 4.20: Výsledek experimentu s palivovou pumpou

Z vykresleného obrazu lze vidět, jak sepnutí pumpy ovlivnilo velikost a směr magnetické indukce. Při sepnutí tekla palivovou pumpou proud o velikosti 2 A a jelikož nebyla ponořena v palivu, což v jejím běžném provozu je, nesměli jsme jí nechat sepnutou na dlouhou dobu. Palivo slouží jako chladicí médium a tímto zacházením bychom jí mohli poškodit.

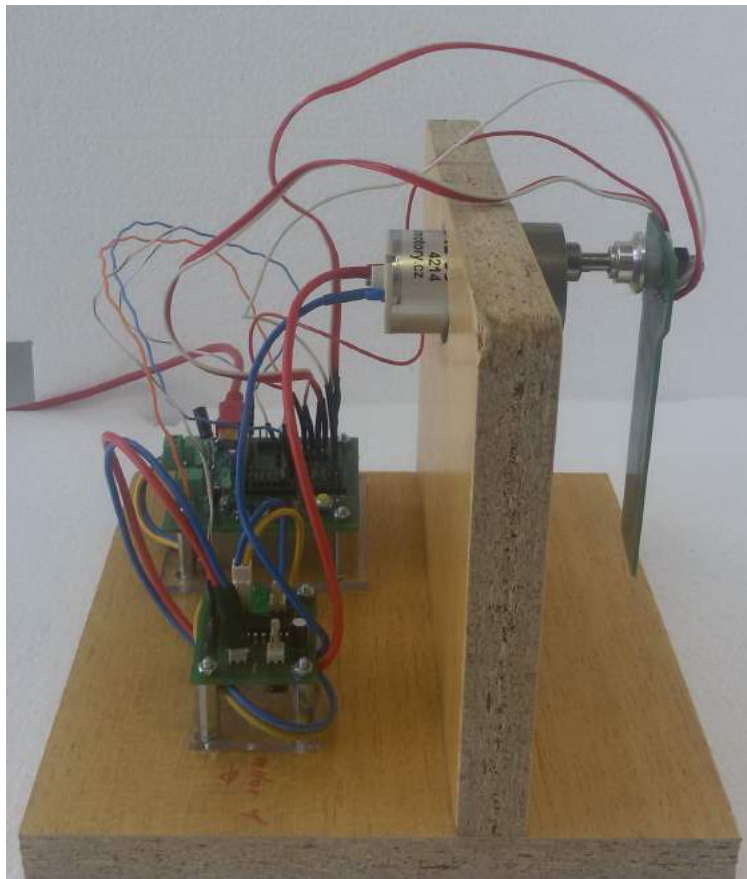
## 4.6. Ovládání zařízení

V první řadě je nutné připojit napájení řídicí elektroniky a následně napájení motoru. Poté zapnout počítač, na kterém je nainstalovaný software Matlab a mít umístěné potřebné soubory (nahrané na přiloženém CD nosiči) v hlavním pracovním adresáři Matlabu. Pomocí souboru *bak\_gui.fig* dojde ke spuštění řídicího GUI, kde je nutné zadat port počítače, ke kterému je zařízení připojeno a počet měření, které chce uživatel provést. Po těchto úkonech se může přejít k umístění zařízení do blízkosti zkoumaného předmětu a následně stiskem tlačítka START spustit měření. Nejprve dojde k výpočtu offsetů magnetometrů a zpětnovazebního akcelerometru a gyroskopu pro správné polohové řízení a poté se otevře samostatné okno, kde probíhá postupné vykreslování vektorů magnetické indukce v reálném čase.

---

## 5. Konstrukce zařízení

Konstrukce prototypu se skládá ze dvou na sebe kolmých desek dřevotřísky, které jsou k sobě přišroubovány vruty. Pohled na prototyp z boku je na *obr. 5.1*. Důvodem pro výběr takto silných desek dřevotřísky bylo, že je prototyp citlivým měřicím zařízením, jehož výsledky jsou snadno ovlivněny vibracemi, nebo nestabilitou kvůli upevněnému těžkému DC motoru a rotací kyvadla.

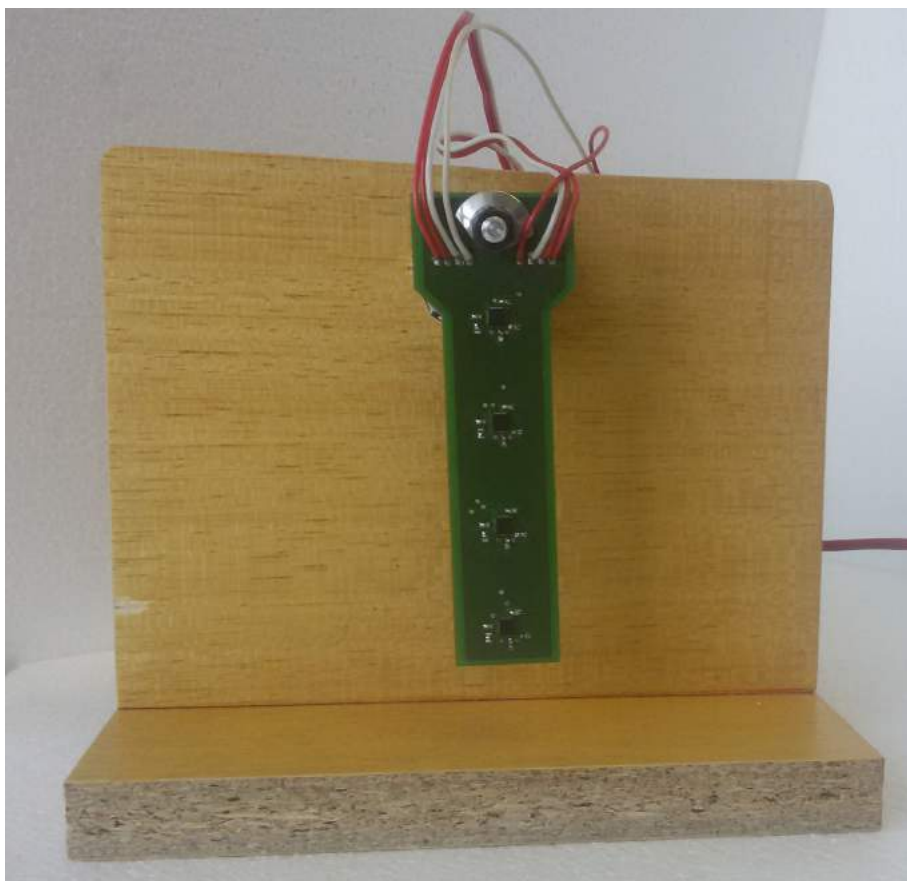


Obrázek 5.1: Pohled na prototyp z boku

DPS s řídicím mikrokontrolérem a DPS s řídicím můstkem motoru jsou připevněny k distančním sloupkům, které jsou přišroubované k plátku plexiskla. Plátky plexiskla jsou následně přišroubovány k dřevotřískové základně.

DPS kyvadla se senzory je připevněna k hřídeli motoru pomocí unašeče vrtule, který se využívá spíše v modelářství v konstrukcích RC modelů letadel. Pro naši aplikaci to ale byl efektivní způsob řešení připevnění DPS k hřídeli.

Na fotce níže lze vidět prototyp s upevněným kyvadlem se senzory zepředu.



Obrázek 5.2: Pohled na prototyp zepředu

## 6. Závěr

Cílem této práce byl návrh a realizace zařízení určeného pro vizualizaci vektorů magnetické indukce v reálném čase, což bylo zdárně splněno.

Návrh započal výběrem vhodného senzoru magnetické indukce. Byla vybrána inerciální měřicí jednotka MPU9150, která obsahuje magnetometr, gyroskop i akcelerometr.

Na vývojové jednotce EduKit proběhly první testy měření magnetické indukce, které zároveň sloužily k seznámení se s automatickým generováním kódu do mikrokontroléru z prostředí Simulink díky nástavbové knihovně MPLAB Device Blocks for Simulink. V rámci těchto měření bylo úspěšně zvládnuto zaslání naměřených dat do počítače, kde bylo vytvořeno jednoduché grafické uživatelské rozhraní v Matlabu pro vizualizaci naměřených dat v podobě vektorů.

Zařízení dostalo podobu stejnosměrným motorem poháněného kyvadla, na němž jsou umístěny čtyři IMU jednotky.

Realizace výsledné podoby zařízení obnášela návrh tří desek plošných spojů. První byla navržena hlavní řídicí DPS s mikrokontrolérem dsPIC, která obsluhuje řízení můstku stejnosměrného motoru, čtení naměřených dat ze senzorů, jejich zpracování a následné odeslání do počítače k vizualizaci ve smysluplné podobě. Dále byla navržena DPS s můstkem pro spínání stejnosměrného motoru, na jehož hřídeli je umístěna třetí navržená DPS kyvadla, na které jsou umístěny čtyři IMU jednotky v jedné ose pod sebou. Díky pohonu je kyvadlo uvedeno do kývavého pohybu, čímž bylo docíleno nasnímání magnetické indukce v jedné ploše.

Pro požadovanou funkčnost zařízení byl vyvinut komplexní software mikrokontroléru v Simulinku, Matlab skriptů obsluhujících čtení a vizualizaci příchozích dat na straně počítače a grafické uživatelské rozhraní pro ovládání zařízení.

Naskytly se problémy s polohovým řízením stejnosměrného motoru, kde je namísto konvenčního enkodéru využito akcelerometru a gyroskopu. Nejprve bylo zjištěno, že první vybraný motor není kvůli vnitřnímu tření rotoru pro danou aplikaci vhodný a bylo nutné zakoupit nový. Dále se naskytl problém s komplementárním filtrem dat akcelerometru a gyroskopu, jehož odezva nebyla dostatečně rychlá. Nakonec postačily přímé výstupy z akcelerometru a gyroskopu filtrované jednoduchým filtrem. Dalším problémem byla komunikace se senzory kyvadla po I<sup>2</sup>C sběrnici, kde docházelo k zamrzání komunikace při čtení dat ze všech senzorů najednou. Problém byl diagnostikován osciloskopem přímo na datových linkách sběrnice a odstraněn střídavým přepínáním čtení a odesílání dat, aby žádná z těchto akcí neprobíhala zaráz. Tyto problémy zabraly velké množství času. Při lépe zvoleném časovém harmonogramu práce v průběhu semestru by zmíněné problémy neměly tolik vliv na výslednou kvalitu prototypu zařízení.

Zařízení je ovládáno z grafického uživatelského rozhraní počítače a je schopno naměřit vektory magnetické indukce v jedné ploše, které jsou následně vizualizovány v počítači ve smysluplné podobě.

# Literatura

- [1] REICHL J., VŠETIČKA M., *Encyklopedie fyziky*. [online]. 2010-02-16.  
Dostupné z: <http://fyzika.jreichl.com/main.article/view/294-halluv-jev>
- [2] FREESCALE, *MAG3110 ThreeAxis, Digital Magnetometer*. Datasheet, rev. 9.1, 10/2012. [online].  
Dostupné z: [https://www.nxp.com/files/sensors/doc/data\\_sheet/MAG3110.pdf](https://www.nxp.com/files/sensors/doc/data_sheet/MAG3110.pdf)
- [3] InvenSense Inc., *Motion Processing Unit<sup>TM</sup> - MPU9150*. Datasheet, rev. 4.3, 9/2013. [online]. Dostupné z:  
<https://www.invensense.com/products/motion-tracking/9axis/mpu-9150/>
- [4] SPARKFUN ELECTRONICS, *SparkFun 9 Degrees of Freedom Breakout - MPU-9150*. [online].  
Dostupné z: <https://www.sparkfun.com/products/retired/11486>
- [5] MATOUŠEK, D., *Udělejte si z PC - generátor, čítač, převodník, programátor-*. Praha: BEN - technická literatura, 2001. PC & elektronika. s.78. ISBN 80-7300-036-9.
- [6] MICROCHIP, *dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04 and dsPIC33FJ128MCX02/X04*. Datasheet, rev. 04/2012. [online].  
Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/70291G.pdf>
- [7] SPARKFUN ELECTRONICS *Serial Communication*. [online]. 2010-02-16.  
Dostupné z: <https://learn.sparkfun.com/tutorials/serial-communication>
- [8] TEXAS INSTRUMENTS, *μA7800 SERIES POSITIVE-VOLTAGE REGULATORS*. Datasheet, rev. 05/2003. [online]. Dostupné z:  
<https://www.sparkfun.com/datasheets/Components/LM7805.pdf>
- [9] STMICROELECTRONICS, *LF00 SERIES VERY LOW DROP VOLTAGE POSITIVE REGULATORS WITH INHIBIT*.  
Datasheet, rev. podzim/2003. [online]. Dostupné z:  
<https://www.sparkfun.com/datasheets/Components/LM7805.pdf>
- [10] MICROCHIP, *dsPIC33EPXXX(GP/MC/MU)806/810/814 and PIC24EPXXX(GP/GU)810/814*. Datasheet, rev. podzim/2003. [online]. Dostupné z:  
<http://www.microchip.com/wwwproducts/en/dsPIC33EP512MC806>
- [11] TEXAS INSTRUMENTS, *L293x Quadruple Half-H Drivers*. Datasheet, rev. 01/2016. [online]. Dostupné z:  
<http://www.ti.com/lit/ds/symlink/l293.pdf>
- [12] MICROCHIP, *MATLAB®/Simulink® Device Blocksets for dsPIC® DSCsink*. Datasheet, rev. 2011. [online]. Dostupné z:  
[http://ww1.microchip.com/downloads/en/DeviceDoc/MATLAB\\_Device\\_Blocks\\_DS-51771E.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/MATLAB_Device_Blocks_DS-51771E.pdf)

## 7. Seznam použitých zkratek a symbolů

PC	Osobní počítač
I <sup>2</sup> C	Inter-integrated circuit
UART	Univerzální asynchronní vysílač a přijímač
MIPS	Milion operací za sekundu

---

# Seznam obrázků

2.1	DPS s MPU9150 [4]	9
2.2	Schéma připojení zařízení na sběrnici I <sup>2</sup> C	9
2.3	komunikace po I <sup>2</sup> C sběrnici [2]	10
2.4	EduKIT s připojeným magnetometrem	11
2.5	Simulink model pro čtení dat ze senzoru	12
2.6	blokové schéma sběrnice UART	13
2.7	Sestava experimentu pro porovnání senzorů	14
3.1	Blokové schéma zařízení	16
3.2	Schéma zapojení regulátorů vstupního napětí	17
3.3	Schéma převodníku UART-USB	18
3.4	Přední strana DPS s mikrokontrolérem	18
3.5	Zadní strana DPS s mikrokontrolérem	19
3.6	Schéma základního zapojení dsPIC kontroléru	20
3.7	Pouzdro obvodu L293D (převzato z [11])	21
3.8	Schema zapojení řídicího můstku L293D	22
3.9	Přední strana DPS s řídicím můstkem L293D	23
3.10	Zadní strana DPS s řídicím můstkem L293D	24
3.11	DPS kyvadla z přední strany	25
3.12	Schéma zapojení IMU jednotky na DPS se senzory	26
3.13	Stejnoseměrný motor pro pohon kyvadla	27
4.1	Model programu mikrokontroleru v simulinku rozdělený do podsystémů	29
4.2	Obsah konfiguračního subsystému	30
4.3	Obsah subsystému MOTOR CONTROL	31
4.4	Model pro čtení dat z akcelerometru a gyroskopu	31
4.5	Komunikační sekvence pro čtení dat akcelerometru a gyroskopu	32
4.6	Filtrování příchozích hodnot z akcelerometru	33
4.7	Část stateflow funkce pro ovládání žádané polohy kyvadla	33
4.8	Diskrétní PID regulátory pro řízení polohy kyvadla	34
4.9	Vykreslené polohy senzorů v průběhu kyvu kyvadla	35
4.10	Subsystému uvnitř READING & SENDING DATA	35
4.11	Stateflow logika pro kontrolu taktu čtení a zasílání dat z magnetometrů	36
4.12	Model pro vyčtení naměřených hodnot z magnetometrů na jedné I <sup>2</sup> C lince	37
4.13	První část sekvence komunikace s magnetometrem po I <sup>2</sup> C sběrnici	38
4.14	Druhá část sekvence komunikace s magnetometrem po I <sup>2</sup> C sběrnici	38
4.15	Část subsystému obsluhující zaslání dat do počítače	39
4.16	Subsystém obsluhující příjem příkazu z počítače	41
4.17	Vykreslené vektory magnetické indukce v blízkosti zmagnetizované konstrukce DC motoru	44
4.18	Primitivní grafické uživatelské rozhraní	46
4.19	Palivová pumpa jakožto zkoumaný předmět experimentu	47
4.20	Výsledek experimentu s palivovou pumpou	48
5.1	Pohled na prototyp z boku	49
5.2	Pohled na prototyp zepředu	50