



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV INTELIGENTNÍCH SYSTÉMŮ**

DEPARTMENT OF INTELLIGENT SYSTEMS

**VÝVOJOVÁ PLATFORMA ROBOTA**

ROBOT DEVELOPMENT PLATFORM

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**MAREK LÖRINC**

**VEDOUcí PRÁCE**

SUPERVISOR

**Ing. JAROSLAV ROZMAN, Ph.D.**

**BRNO 2020**

## Zadání bakalářské práce



Student: **Lörinc Marek**  
Program: Informační technologie  
Název: **Vývojová platforma robota**  
**Robot Development Platform**  
Kategorie: Umělá inteligence

### Zadání:

1. Nastudujte hardwarové vybavení potřebné pro tvorbu vlastního robota (motory, řízení, senzory, komunikaci, atd.). Jako procesor použijte Atmega32.
2. Na základě nastudovaných informací navrhnete potřebnou elektroniku pro řízení čtyřkolového robota. Robota vybavte senzory - IMU, enkodéry, sonary/IR senzory, atd. Navrhnete program pro řízení robota pomocí smartphone na systému Android.
3. Implementujte navrženou hardwarovou část robota a vytvořte program pro řízení robota pomocí smartphone.
4. Vytvořeného robota včetně správné funkčnosti senzorů otestujte.

### Literatura:

- McComb, G, Predko, M.: Robot Builder's Bonanza. McGraw-Hill, 2006, ISBN 0-07-146893-5
- Cook, D.: Intermediate Robot Building. Apress, 2004, ISBN 1-59059-373-1

Pro udělení zápočtu za první semestr je požadováno:

- První dva body zadání.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Rozman Jaroslav, Ing., Ph.D.**

Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.

Datum zadání: 1. listopadu 2019

Datum odevzdání: 28. května 2020

Datum schválení: 31. října 2019

## Abstrakt

Práca sa zaoberá návrhom a vytvorením vývojovej platformy pre robota určenej najmä pre začiatočníkov. K tomu je použitý mikrokontrolér AVR ATmega32, ku ktorému je navrhnutá a vyrobená doska plošného spoja. Ďalej sa práca venuje tvorbe robota na tejto platforme, ktorý obsahuje najpoužívanejšie komponenty v robotike. Tento robot poskytne začiatočníkom všetky potrebné informácie na stavbu vlastného robota. Posledná časť práce je návrh a tvorba aplikácie, ktorá posluží na ovládanie robota a získavanie potrebných informácií z robota cez Bluetooth. Aplikácia je navrhnutá tak, aby ju každý mohol použiť na ovládanie svojho robota.

## Abstract

This thesis deals with the design and creation of a development platform for a robot designed especially for beginners. For this purpose is used the microcontroller AVR ATmega32 for which is designed and manufactured circuit board. It also deals with creating a robot on this platform, which contains the most used components in robotics. This robot will provide beginners all the necessary information to build their own robot. The last part of the thesis is the design and creation of the application, which will serve to control the robot and obtain the necessary information from the robot via Bluetooth. The application is designed to everyone could use it to control their robot.

## Kľúčové slová

Robotika, vývojová platforma, AVR Atmega32, android aplikácia, jazyk C, tvorba robota

## Keywords

Robotics, development platform, AVR Atmega32, android application ,C language, robot building

## Citácia

LÖRINC, Marek. *Vývojová platforma robota*. Brno, 2020. Bakalárska práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jaroslav Rozman, Ph.D.

# Vývojová platforma robota

## Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Jaroslava Rozmana, Ph.D. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....  
Marek Lörinc  
28. mája 2020

## Podakovanie

Ďakujem vedúcemu práce, pánovi Ing. Jaroslavovi Rozmanovi, Ph.D. za poskytnuté rady pri tvorbe a písaní práce.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Hardvérové vybavenie</b>	<b>4</b>
2.1	Mikrokontrolér AVR ATmega32 . . . . .	4
2.1.1	Funkčné bloky . . . . .	6
2.1.2	Programovanie mikrokontroléra . . . . .	7
2.2	Elektromotory . . . . .	7
2.3	H-mostík . . . . .	8
2.4	Senzor na meranie vzdialenosti . . . . .	9
2.4.1	Ultrazvukový dialkomer . . . . .	9
2.4.2	Infračervený dialkomer . . . . .	10
2.5	Senzor na sledovanie čiary . . . . .	11
2.6	Jednotka IMU . . . . .	11
2.7	Servomotory . . . . .	12
2.8	Senzor na počítanie otáčok motora . . . . .	13
2.9	Obvod reálneho času . . . . .	14
2.10	Bezdrôtová sériová komunikácia . . . . .	15
2.11	Displej a I2C modul . . . . .	16
<b>3</b>	<b>Funkčné bloky mikrokontroléra a ich registre</b>	<b>18</b>
3.1	SPI . . . . .	18
3.2	Vonkajšie vstupy prerušenia . . . . .	19
3.3	A/D prevodník . . . . .	20
3.4	Generovanie signálu pomocou pulzne šírkovej modulácie . . . . .	22
3.5	Rozhranie TWI . . . . .	24
3.6	Jednotka USART . . . . .	25
3.7	Vstupné/Výstupné porty . . . . .	27
<b>4</b>	<b>Elektronika</b>	<b>28</b>
4.1	Analýza konkurencie a riešenie konkurenčných nedostatkov . . . . .	28
4.2	Hlavné prvky schémy zapojenia . . . . .	29
4.3	Návrh dosky plošného spoja . . . . .	30
4.4	Výroba dosky plošného spoja . . . . .	31
<b>5</b>	<b>Konštrukcia robota</b>	<b>32</b>
5.1	Hotové konštrukcie . . . . .	32
5.2	Súprava častí - Kit . . . . .	33
5.2.1	Robotický systém VEX . . . . .	33

5.2.2	Súprava Lynxmotion Servo Erector . . . . .	33
5.2.3	Bioid . . . . .	33
5.3	Vlastná konštrukcia . . . . .	34
5.3.1	Drevo . . . . .	34
5.3.2	Plast . . . . .	34
5.3.3	Kov . . . . .	35
5.3.4	Lahké materiály . . . . .	35
5.4	Použitá konštrukcia pre robota . . . . .	36
<b>6</b>	<b>Aplikácia na ovládanie robota</b>	<b>37</b>
6.1	Existujúce riešenia . . . . .	37
6.2	Grafické užívateľské prostredie . . . . .	37
6.3	Konfigurácia aplikácie . . . . .	38
<b>7</b>	<b>Testovanie</b>	<b>40</b>
7.1	Testovanie senzorov . . . . .	40
7.2	Testovanie podprogramov robota . . . . .	41
7.3	Testovanie ostatných hardvérových prvkov . . . . .	42
<b>8</b>	<b>Záver</b>	<b>44</b>
	<b>Literatúra</b>	<b>45</b>
<b>A</b>	<b>Schéma zapojenia</b>	<b>47</b>

# Kapitola 1

## Úvod

Cieľom práce je vytvorenie vývojovej platformy, ktorá bude najlepšou voľbou pre začiatočníka, ktorý si chce postaviť a naprogramovať robota podľa vlastných predstáv. V ideálnom prípade by užívateľ pripojil danú súčiastku (motor, senzor, atď.) k platforme, tú k počítaču a už len programoval. Tento spôsob sa snaží poskytovať najznámejšie riešenie v danom segmente a to Arduino, avšak má veľké množstvo nedostatkov. Za najväčší pokladám ten, že užívatelia začiatočníci takmer vôbec nevedia ako použiť senzory a súčiastky fungujú. To je spôsobené najmä tým, že ich mikrokontroléry sú zbytočne komplikované a tým pádom je pre začiatočníka veľmi ťažké niečo pochopiť z technickej dokumentácie. Tak sa pri programovaní používajú len hotové funkcie od výrobcu, prípadne tretích strán, čo znamená, že sa užívateľ takmer nič nenaučí. To je hlavný dôvod prečo som sa rozhodol vytvoriť vlastnú platformu. Užívateľ sa pri vytváraní programu toho veľa naučí o procesore a aj o zariadení, pretože bude vytvárať výsledný program pomocou registrov mikrokontroléra na základe technickej dokumentácie. Tú má procesor AVR Atmega32 veľmi dobre napísanú aj pre začiatočníkov. Tento spôsob vytvárania programu sa používa aj pri najmodernejších a najvýkonnejších procesoroch, preto potom následný prechod z dôvodu nepostačujúceho výkonu bude takmer bezproblémový.

Na ovládanie robota a získavanie potrebných informácií slúži aplikácia pre smartphome na operačnom systéme Android. Samotné prvky v aplikácii je možné si nakonfigurovať podľa vlastných predstáv alebo sa dá využiť aj preddefinovaná šablóna. Aplikácia je použiteľná aj pre akúkoľvek platformu, ktorá komunikuje so smartphonom cez Bluetooth.

Rovnako dôležitá časť tejto práce je aj vytvorenie robota, ktorý slúži na demonštráciu základných prvkov, ktoré by robot mohol obsahovať. Sú to najmä senzory, ktoré slúžia na monitorovanie okolia, ale aj súčiastky, ktoré slúžia na pohyb robota prípadne manipuláciu s predmetmi. Na tejto ukážke je dobre vidieť, že na vcelku zložitého robota, postačí aj jednoduchý 8-bitový mikrokontrolér, ktorý je podľa mňa ideálnou voľbou pre začiatočníka.

## Kapitola 2

# Hardvérové vybavenie

Pred samotným návrhom dosky plošného spoja pre vývojovú platformu je potrebné zvážiť, čo všetko môže robot obsahovať. Základné prvky, ktoré by mal robot obsahovať sú senzory na sledovanie okolia, motory na pohyb a modul na komunikáciu s robotom. Ďalším dôležitým aspektom je, aby boli tieto prvky kompatibilné s použitým mikrokontrolérom AVR ATmega32.

### 2.1 Mikrokontrolér AVR ATmega32

AVR ATmega32 je 8-bitový mikrokontrolér od spoločnosti Atmel. Jeho hlavnými prednosťami sú nízka energetická náročnosť, cena v pomere za výkon a je jednoduché pochopiť jeho operácie. Procesor je založený na Harwardskej architektúre s inštrukčnou sadou s 131 inštrukciami, čím sa približuje k CPU s architektúrou CISC. Avšak ostatné parametre ako je spracovanie inštrukcií v jednom hodinovom cykle, rovnaká bitová šírka inštrukcií sú charakteristické vlastnosti pre architektúru RISC. Čiže by sa dalo povedať, že sa využíva výhody oboch architektúr, pretože je možné udržať malú veľkosť programu a zároveň veľkú rýchlosť spracovania. Inštrukčný kód má dĺžku 16 bitov a je teda dostatočne veľký, aby mohol v jednom inštrukčnom slove pojať inštrukciu aj operand. Maximálna frekvencia procesora je až 16MHz [3].

Základné vlastnosti mikrokontroléra [12]:

- Inštrukčný súbor obsahuje 131 inštrukcií
- 32 registrov dĺžky 8 bitov
- Štyri 8bitové vstupno/výstupné porty (celkom teda 32 vstupov/výstupov)
- Hodinový takt 0 až 16MHz, maximálny výpočetný výkon je až 16 MIPS
- Pamäť programu je tvorená zabudovanou Flash, kapacita je 32 KB
- Dátová pamäť RAM kapacity 2 KB
- Dátová pamäť EEPROM kapacity 1024 B
- Flash a EEPROM sú programovateľné priamo v systéme pomocou rozhrania SPI alebo JTAG
- Dva 8bitové čítače/časovače a jeden 16 bitový



- Analógový komparátor, 10bitový A/D prevodník
- Jednotky USART, SPI a TWI
- Jednotky WDT a Power-on reset

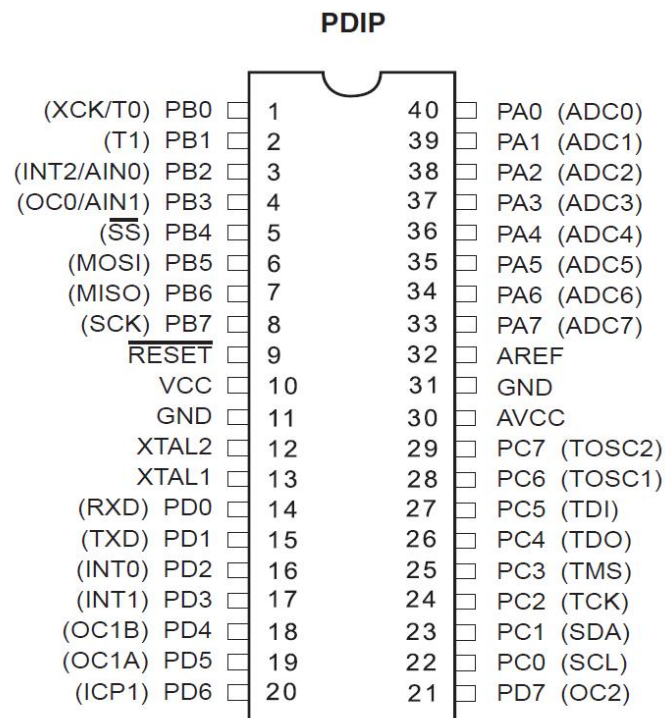
Interná pamäť mikrokontroléra je rozdelená na [12]:

- Flash pre program - 32KB
- EEPROM pre ukladanie dát - 1024KB
- SRAM pre registre - 2KB

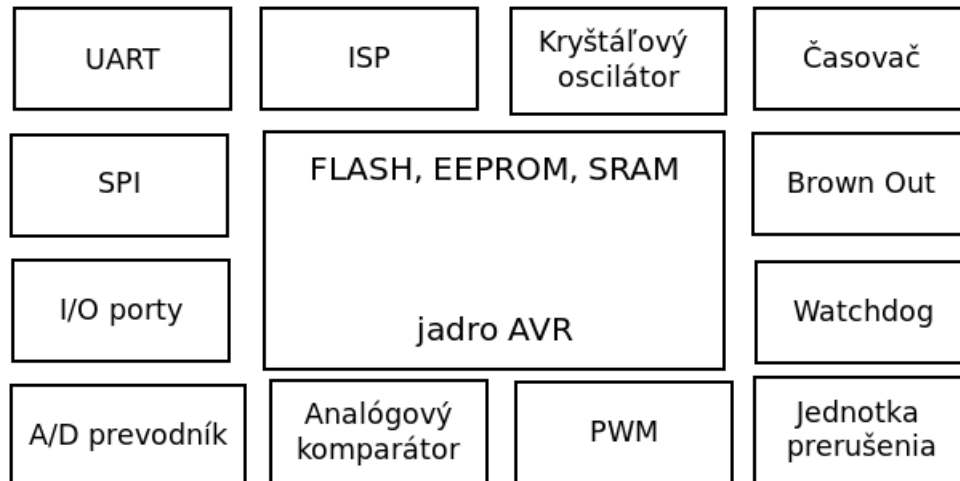
Adresný priestor pre dátovú pamäť je lineárny, navzdory funkčnému rozdeleniu.

Celkovo má mikrokontrolér ATmega32 32 identických 8-bitových registrov pre všeobecné použitie, ktoré sa všetky dajú použiť ako akumulátor. Všetky registre sú adresovateľné prostredníctvom štandardných pamäťových prístupov. S výnimkou štyroch 32 bitových inštrukcií majú všetky dĺžku 16 bitov [18].

Mikrokontrolér obsahuje piny pre napájanie, reset, externý kryštál, pre referenčné napätie pre A/D prevodníky a 32 I/O pinov z ktorých má každý vlastnú špecializáciu a zároveň všetky sa dajú použiť aj ako bežné digitálne I/O piny. Tieto piny sú rozdelené do 4 základných blokov, ktoré charakterizujú ich využitie a sú to bloky PORTA, PORTB, PORTC a PORTD. Každý blok má 8 pinov [3].



Obr. 2.1: Atmega32 pin Diagram <sup>1</sup>



Obr. 2.2: Atmega32 rozdelenie funkčných blokov <sup>2</sup>

### 2.1.1 Funkčné bloky

Základné bloky mikrokontroléra AVR Atmega32 sú [17]:

- USART - Univerzálny sériový prijímač / vysielateľ
 

Plne duplexný sériový kanál, umožňujúci komunikáciu v štandardnom 8 a 9 bitovom asynchrónnom režime. Obsahuje detekciu falošného start-bitu, detekciu nesprávneho znaku a pretečenie dátového registra, filtráciu šumu. Disponuje 3 vektormi prerušenia: vysielanie dokončené, vysielací register prázdny a príjem kompletný.
- ISP - In-system programming
 

Tento blok umožňuje programovanie mikrokontroléra cez jednoduché sériové rozhranie priamo v aplikácii
- Brown Out - Podpäťová ochrana
- Watchdog - Strážny obvod
 

Je časovač, ktorý je spustený po resete alebo neskôr z programu a po vypršaní časového limitu vyvolá automatické prerušenie, pokiaľ ho nejakým podnetom (signálom, hodnotou zapísanou do riadiaceho registra, špeciálnou inštrukciou) nevrátime do počiatočného stavu
- Čítač / časovač - môže mať niekoľko funkcií:
 

Output Compare - porovnáva hodnotu časovača/čítača s hodnotou v Output Compare registri a v prípade zhody môže vyvolať prerušenie

Input Compare - zachytáva stav časovača/čítača v okamžiku kedy sa zmení stav na pine ICP do aktívnej úrovne

PWM - tento blok slúži na generovanie signálu PWM (pulzne šírková modulácia)
- SPI - synchronný sériový port

<sup>1</sup><http://www.circuitstoday.com/wp-content/uploads/2012/01/ATmega8.jpg>

- Analógový komparátor - môže byť použitý ako menej presný A/D prevodník
- A/D prevodník - 8 bitový + 10 bitový s možnosťou obmedzenia šumu
- Prerušovací systém (Interrupts)

Sú to adresy vektorov prerušenia, ktoré sú usporiadané za sebou od začiatku pamäťového bloku.

### 2.1.2 Programovanie mikrokontroléra

Na programovanie mikrokontrolérov radu AVR sa väčšinou používa softvér od výrobcu a to *Atmel Studio*. Na programovanie je možné použiť Assembler alebo jazyk C. Oba jazyky majú svoje výhody. Assembler produkuje vo výsledku menší kód a je rýchlejší. Výhody jazyka C sú najmä rýchlejšom písaní kódu a je jednoduchší na pochopenie. Použitím knižníc sa kód môže jednoducho znovu použiť a predpísané funkcie je možné používať len pridaním knižnice [9].

Assembler je vhodný najmä vtedy, ak je rýchlosť veľmi dôležitá. Ak je teda požadované, aby mikrokontrolér pracoval v reálnom čase napríklad čítaním dát na UART, spracovaním dát a zobrazením na SPI zbernici kým sa taktiež monitorujú iné funkcie. Takže najlepším riešením je v týchto kritických situáciách použiť assembler a inak jazyk C. Väčšina C prekladačov podporuje vloženie kódu jazyka Assembler do kódu jazyka C (vrátane prekladača softvéru Atmel Studio) [9].

Keďže robot neobsahuje žiadne časovo kritické úseky, je použitý iba jazyk C.

## 2.2 Elektromotory

Na napájanie robota bude použitá batéria, takže je vhodné použiť jednosmerné motory. Robot najčastejšie býva poháňaný jednosmerným motorom s permanentným magnetom alebo krokovým motorom.

Motor s permanentným magnetom je najjednoduchším motorom na jednosmerný prúd. Jeho stator je tvorený permanentným magnetom. Rotor tvorí elektromagnet s pólami. Elektrický prúd je do cievok rotora privádzaný cez komutátor, čo je vlastne rotačný prepínač. Jeho úlohou je meniť polaritu elektrického prúdu a tým aj polaritu magnetického poľa prechádzajúceho cievkami. Počet prepínacích plôšok komutátora zodpovedá počtu cievok (najmenej dve). Konštrukcia komutátora zaisťuje, že sily pôsobiace na póly rotora majú stále rovnaký smer. V okamihu prepnutia polaritu udržiava beh tohto motora v správnom smere zotrvačnosť rotora. Počet pólov rotora ovplyvňuje plynulosť chodu motora, a silu potrebnú na jeho rozbeh (záberový moment). Čím viac pólov, tým plynulejší chod, obvykle sú štyri. Vzhľadom na to, že výkon motora je závislý na veľkosti permanentného magnetu sa takáto konštrukcia používa len pre malé elektromotory. Využitie je na pohon ventilátorov používaných pri chladení v elektrotechnike, modelárske motorčeky pre pohon hračiek a podobne. Výhodou motora s permanentným magnetom je možnosť meniť smer otáčania zmenou polaritu napájania [5].

Krokový motor je v podstate špeciálny synchronný motor, ktorého funkčný pohyb, t.j. otáčanie hriadeľa, nie je plynulý ale je zložený z čiastkových rovnako veľkých pohybov - krokov. Krokové motory umožňujú prenášať riadiacu informáciu - impulzy. Impulzy sa privádzajú k jednotlivým sekciám statorového vinutia. Frekvencii impulzu zodpovedá rýchlosť

<sup>2</sup><http://noel.feld.cvut.cz/vyu/scs/prezentace2005/ATmega/ATmega.pdf>

natáčania motora. Hraničná frekvencia je 1 až 2 kHz, pri ktorej sa stroj vyradí z činnosti. Použitie krokových motorov je dnes už veľmi rozmanité. Používajú sa na pohon valcovacích stolíc, navíjačiek, zväračiek no veľmi často sa používajú aj na pohon rôznych periférnych zariadení počítačov a robotov [5].

Ovládanie krokového motora je zložitejšie a tým pádom zaberá na procesore viac výstupov na robotovi sú použité jednosmerné motory s permanentným magnetom, konkrétne motory od spoločnosti Elecrow. Maximálnu rýchlosť otáčania(500RPM) dosahujú pri napätí 13V a veľkosť prúdu závisí od zaťaženia (maximum 1A). Na riadenie smeru otáčania motorov je nutné použiť H-mostík, ktorý zaťaženie zvládne aspoň pre 2 motory.



Obr. 2.3: Použitý jednosmerný motor Elecrow <sup>3</sup>

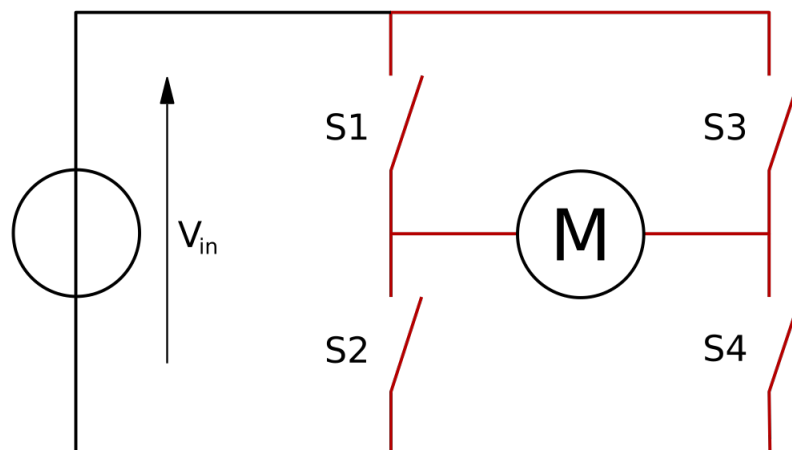
## 2.3 H-mostík

H mostík je elektronický obvod, ktorý mení polaritu napätia aplikovaného na záťaž. H-mostík môže byť realizovaný ako integrovaný obvod alebo poskladaný z jednotlivých komponentov. Termín H-mostík je odvodený od jeho grafického znázornenia(2.4). Keď sú prepínače S1 a S4 zatvorené a S2 a S3 otvorené, na motor bude aplikované pozitívne napätie. Otvorením spínačov S1 a S4 a zatvorením spínačov S2 a S3 je toto napätie obrátené, čo umožňuje spätnú prevádzku motora [2].

Vhodnou voľbou pre použité motory najmä z hľadiska ceny je H-mostík L298N. Slúži na riadenie 1 krokového motora alebo 2 jednosmerných motorov. Pre riadenie 1 jednosmerného motora je potrebné použiť tri piny z procesora. Jeden na určenie rýchlosti motora (PWM) a ďalšie 2 na určenie smeru otáčania prípadne zastavenie(postačia bežné I/O piny). H-mostík zvládne prúd až 2A a napájacie napätie je 5-12V.

<sup>3</sup><https://ae01.alicdn.com/kf/HTB1hMkwmXkoBKNjSZFEq6zrEVXaL/Elecrow-DIY-Smart-Car-For-Arduino-Robot-Education-Smart-Car-Encoder-Chassis-Front-wheel-Steering-Gear.jpg>

<sup>4</sup>[https://en.wikipedia.org/wiki/H\\_bridge](https://en.wikipedia.org/wiki/H_bridge)



Obr. 2.4: Štruktúra H-mostíka <sup>4</sup>

## 2.4 Senzor na meranie vzdialenosti

Na detekciu predmetov v okolí a zabráneniu prípadnej zrážky, je vhodné použiť senzor na meranie vzdialenosti. V robotike sa na to najčastejšie používajú 2 druhy senzorov a to ultrazvukový alebo infračervený senzor.

### 2.4.1 Ultrazvukový diaľkomer

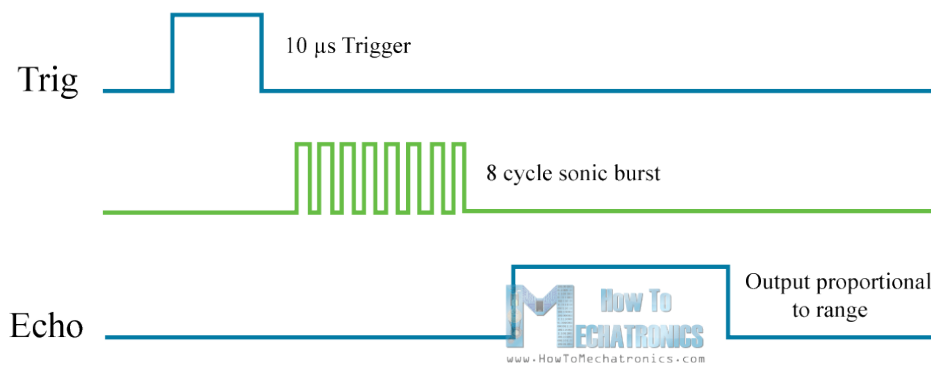
Ultrazvukový diaľkomer ako už názov naznačuje, meria vzdialenosť použitím ultrazvukových vln. Na pokyn hlava snímača vyšle ultrazvukovú vlnu o frekvencii 40kHz a ak mu stojí v ceste prekážka, tak sa zvuk odrazí od prekážky a vráti sa späť do senzora. Na základe rýchlosti zvuku a času kým sa zvuk vrátil späť do senzora, je možné určiť vzdialenosť od predmetu. Ultrazvukový diaľkomer potom meria vzdialenosť od cieľa meraním času medzi odoslaním a prijatím ultrazvukovej vlny. Výhodou ultrazvukového snímača je, že využíva jeden prvok aj na odoslanie aj na prijatie zvukovej vlny [7].

Na robotovi bude použitý ultrazvukový diaľkomer HC-SR04. Ten má 4 vývody a to:

- VCC - napájanie +5V
- GND - zem
- TRIGGER - do tohto pinu sa odosiela impulz na začatie merania
- ECHO - impulz, ktorého šírka určuje vzdialenosť od predmetu

Na generovanie ultrazvuku diaľkomerom je potrebné poslať na pin TRIG 10 us impulz. To vyšle 8-cyklový zvukový výbuch, ktorý sa bude pohybovať rýchlosťou zvuku a bude prijatý na pine Echo. Šírka impulzu na ECHO pine určuje čas, ktorý zvuková vlna cestovala. Vzdialenosť od predmetu je [7]:

$$340m/s * \text{širkaimpulzu}/2.$$

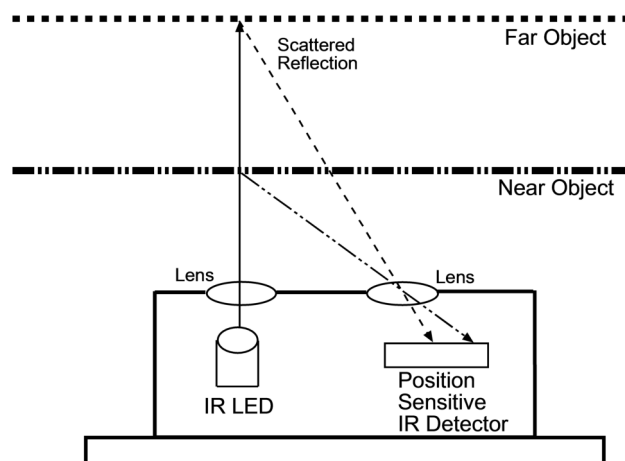


Obr. 2.5: Princíp funkcie ultrazvukového diaľkomeru <sup>5</sup>

### 2.4.2 Infračervený diaľkomer

Infračervený diaľkomer funguje na princípe odrazených svetelných vln. Odrazené svetlo je detekované senzorom a na základe množstva odrazeného svetla je vypočítaná vzdialenosť od predmetu.

Použitý infračervený diaľkomer Sharp GP2Y0A41SK0F sa vyznačuje najmä vysokou spoľahlivosťou a presnosťou. Diaľkomer odosiela vzdialenosť od predmetu vo forme napätia 0-5V, čiže je nutné použiť A/D prevodník a vzorec na prevedenie napätia na vzdialenosť. Keďže v technickej dokumentácii sa žiadny nenachádza, je potrebné vytvoriť si vlastný na základe zmeraných dát.



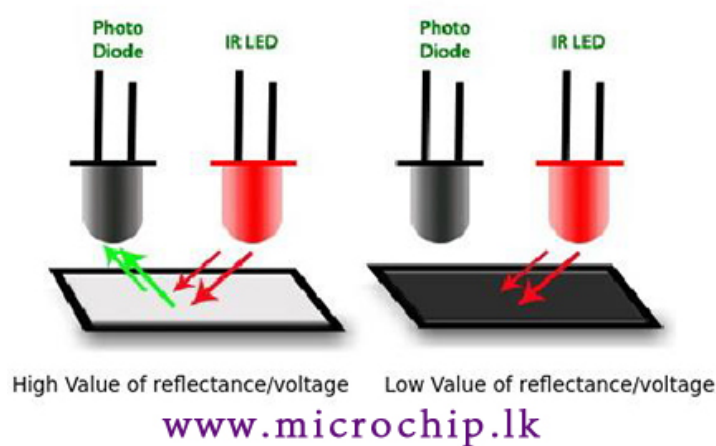
Obr. 2.6: Princíp funkcie infračerveného diaľkomeru <sup>6</sup>

<sup>5</sup><https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/>

## 2.5 Senzor na sledovanie čiary

Zvyčajne sa jedná o 3 alebo viac senzorov ktoré dokážu rozlíšiť farbu medzi sledovanou čiarou a okolím. Na základe pod ktorým sa čiara nachádza sa rozhodne o pohybe robota. Na rozlíšenie farby sa používajú infračervené senzory. Dióda vyžaruje infračervené žiarenie ktoré sa odráža od zeme a na základe množstva odrazeného žiarenia do fototranzistora je možné určiť farbu. Senzor odosiela farbu vo forme napätia v rozmedzí od čiernej, ktorá pohlcuje najviac svetla až po bielu, ktorá najmenej.

### IR Line Track Follower Sensor



Obr. 2.7: Princíp funkcie senzoru na sledovanie čiary <sup>7</sup>

## 2.6 Jednotka IMU

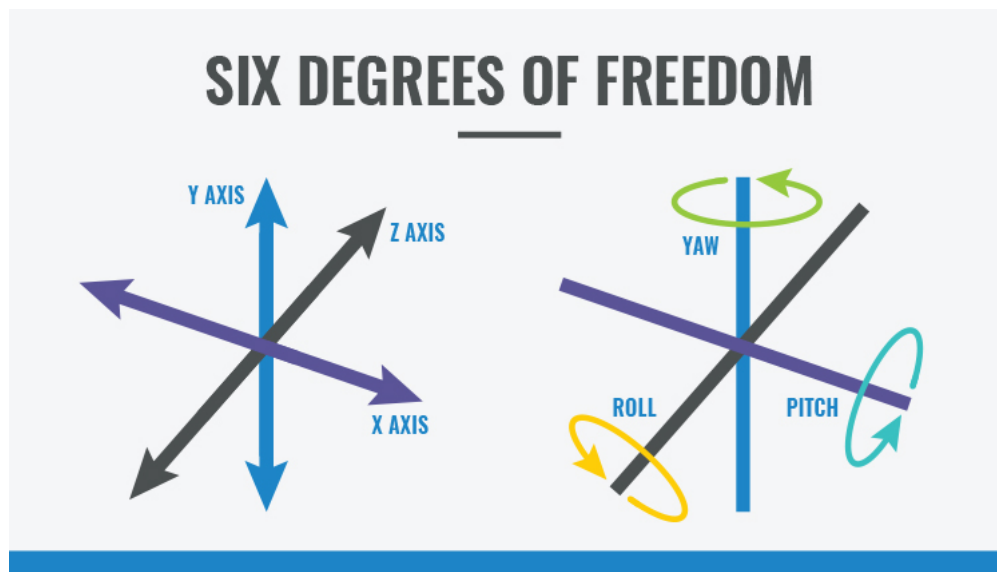
IMU (Inertial measurement unit) je špecifický typ senzora, ktorý meria uhlovú rýchlosť, silu a niekedy aj magnetické pole. Jednotka IMU sa bežne skladá z 3-osového akcelerometra a 3-osého gyroskopu. Takýto senzor by sa považoval za 6-osový IMU. Môže tiež obsahovať ďalší 3-osový magnetometer, kedy by vznikol 9-osový IMU. Technicky sa pojem „IMU“ týka iba senzora, ale IMU sú často spárované so softvérom na fúziu senzorov, ktorý kombinuje údaje z viacerých senzorov, aby poskytol opatrenia na orientáciu a smerovanie. Pri bežnom používaní sa termín „IMU“ môže používať na označenie kombinácie softvéru snímača a snímača fúzie [15].

IMU poskytuje 2 až 6 DOF (stupne slobody), čo predstavuje počet rôznych spôsobov, ktorými sa objekt dokáže pohybovať v 3D priestore. Maximálny možný počet je 6 DOF, ktorý by zahŕňal 3 stupne translačného (plochého) pohybu cez priamu rovinu / pozdĺž každej osi (predná / zadná, pravá / ľavá, nahor / nadol) a 3 stupne rotačného pohybu cez x, y osi / okolo každej osi [15].

Použitý senzor MPU6050 je 6-osový a obsahuje 3 osový gyroskop, 3 osový akcelerometer a senzor na meranie teploty. S mikrokontrolérom dokáže komunikovať pomocou I2C

<sup>6</sup><https://os.mbed.com/forum/mbed/topic/1048/?page=1>

<sup>7</sup><https://www.microchip.lk/product/tcrt5000-ir-infrared-line-track-follower-sensor-obstacle-avoidanc-module>



Obr. 2.8: Znázornenie 6 stupňov slobody (DOF) [15]

zbernice. Údaje z gyroskopu a akcelerometra pozdĺž osí X, Y a Z sú k dispozícii vo forme dvojkového doplnku. Odčítanie teploty je k dispozícii v celočíselnej forme. Gyroskop udáva údaje v stupňoch za sekundu, akcelerometer v jednotkách g a teplota je v stupňoch Celzia.

## 2.7 Servomotory

Na manipuláciu s predmetmi je vhodné použiť rameno. Na ovládanie ramena je potrebné použiť motory, najvhodnejšie servomotory. Servomotor je motor, u ktorého je možné na rozdiel od bežného motoru nastaviť presnú polohu natočenia osy. U RC modelov sa bežne používajú modelárske servomotory [1].

Modelársky servomotor sa skladá z troch základných prvkov:

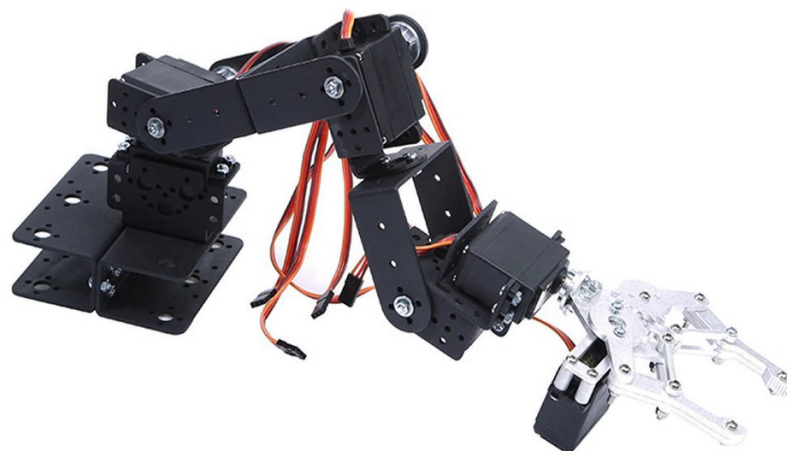
- jednosmerného motoru
- prevodovky
- riadiacej elektroniky

Modelárske servomotory sa delia na dva druhy. Analógové a digitálne. Riadiaca elektronika analógových servomotorov spracováva vstupný PWM signál, z ktorého sa potom vyhodnocuje požadované natočenie. Riadiaci signál u kvalitných servomotoroch je od 1ms do 2ms pre natočenie od  $0^\circ$  po  $180^\circ$ . Pre ideálny vstupný signál je frekvencia opakovania impulzu 50 Hz [14].

Digitálny servomotor je vybavený digitálnou riadiacou elektronikou, ale využíva rovnaký vstupný signál PWM. Digitálne servomotory dokážu pracovať s vyššou frekvenciou, sú presnejšie ale majú vyššiu spotrebu a cenu. Niektoré digitálne servomotory sú programovateľné. Je možné u nich nastaviť neutrálnu polohu, koncové body a rýchlosť pohybu [14].

Z dôvodu, že rameno sa skladá až zo 6 servomotorov, tak cena a spotreba sú veľmi dôležité. Preto je lepšou voľbou použiť analógové servomotory, konkrétne sú použité MG995, ktorých aj presnosť je na postačujúcej úrovni.





Obr. 2.9: Použité rameno polohovateľné 6 servomotormi<sup>8</sup>

## 2.8 Senzor na počítanie otáčok motora

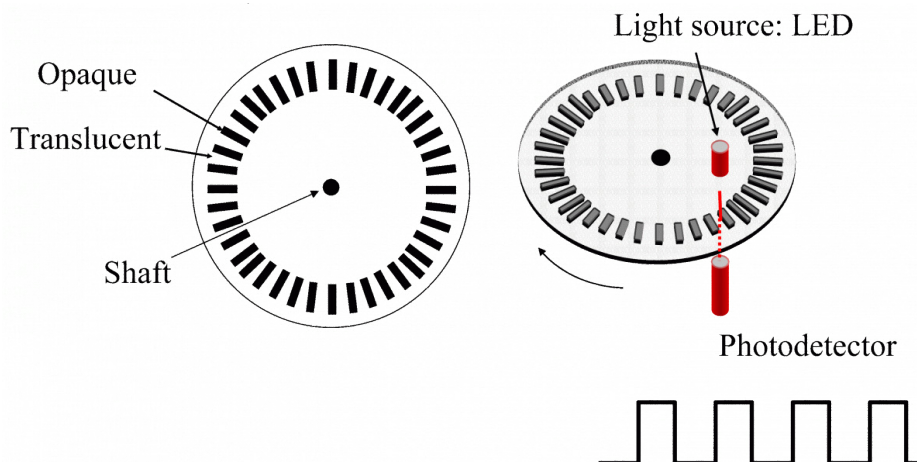
Na počítanie otáčok motora sa používajú enkodéry, konkrétne rotačné optické enkodéry. Enkodér je elektromechanický prevodník, ktorý prevádza rotačný pohyb na sekvencie elektrických digitálnych impulzov. V prípade popisovaného rotačného enkodéru, je celý systém zložený z nasledujúcich hlavných častí [19]:

- disk s postupnosťou priehľadných a nepriehľadných otvorov
- optického vysielača - najčastejšie LED, ktorá vysiela svetlo vo viditeľnom alebo infračervenom spektre
- Optickým prijímačom - fototranzistor alebo fotodióda

Princíp funkcie je zhruba nasledovný: Disk je spojený s otočným spojovacím hriadeľom enkodéru (shaft), ktorý je potom v praxi pripevnený k hriadeľu motora. Hriadeľ sa teda otáča súčasne s diskom. Svetlo vysielačným zdrojom (diódou LED), ktorý je umiestnený pred diskom, môže prechádzať diskom len priehľadnými okienkami (Translucent), zatiaľ čo zostávajúca časť disku je nepriehľadná (Opaque) a pohlcuje svetlo. Svetelné impulzy vytvárané otáčaním disku aktivujú optický snímač, ktorý ich prevádza na elektrické impulzy alebo obdĺžnikový signál. Aby bolo u snímača zaistené generovanie obdĺžnikových impulzov bez rušenia, musí byť elektrický signál zosilnený a elektronicky spracovaný. Pre zlepšenie kvality a stability výstupných signálov sa môže snímať v diferenciálnom režime, kedy sa porovnávajú dva takmer totožné signály s opačnou fázou (tj. Fázový rozdiel 180 elektrických stupňov). Snímanie rozdielu dvoch signálov odstraňuje súhlasné rušenie, pretože rovnaké signály ľubovoľného tvaru sa vždy prekryjú [19].

<sup>8</sup>[https://www.aliexpress.com/wholesale?catId=0&initiative\\_id=SB\\_20191103154705&SearchText=6+DOF+arm](https://www.aliexpress.com/wholesale?catId=0&initiative_id=SB_20191103154705&SearchText=6+DOF+arm)

Použité motory obsahujú rotačný optický enkodér, takže stačí len daný výstup z motora pripojiť na 1 z pinov procesora INT0-INT2 a z technickej dokumentácii motora zistiť počet impulzov pre 1 otočenie.



Obr. 2.10: Princíp funkcie rotačného optického enkodéra <sup>9</sup>

## 2.9 Obvod reálneho času

Obvod reálneho času (RTC) sú počítačové hodiny, zvyčajne vo forme integrovaného obvodu, ktorý je určený výlučne na uchovávanie času. Počíta samozrejme, hodiny, minúty, sekundy, mesiace, dni a dokonca roky. RTC sa nachádzajú v osobných počítačoch, vstavaných systémoch a serveroch a v akomkoľvek elektronickom zariadení, ktoré môže vyžadovať presné dodržiavanie času. Schopnosť fungovať nezávisle od hlavnej energie systému, je zásadný. RTC musia presne udržiavať čas, aj keď je zariadenie vypnuté, pretože sa často používajú ako spúšťač na zapnutie zariadenia alebo na spustenie udalostí, ako sú napríklad budíky. Integrované obvody RTC pracujú s alternatívnym napájacím zdrojom, ktorý umožňuje nepretržitú prevádzku pri nízkom príkone alebo dokonca aj pri vypnutom počítači. IO na starších systémoch využívajú lítiové batérie, zatiaľ čo novšie systémy využívajú pomocné batérie alebo superkondenzátory. Integrované obvody RTC, ktoré používajú superkondenzátory, sú nabíjateľné a dajú sa spájať. Ale vo väčšine základných dosiek pre spotrebiteľov je RTC napájaný jedinou batériou, ktorá po odstránení resetuje RTC na počiatkový bod [8].

Výhody použitia tohto obvodu sú najmä:

- Presnejšia metóda od ostatných ako napríklad meranie času pomocou časovača mikrokontroléra
- Po odpojení hlavného napájania dokáže uchovať aktuálny čas vďaka jeho zálohovacej batérii
- Nízka spotreba a vysoká frekvenčná stabilita
- Nezafážuje hlavný systém výpočtami pre aktuálny čas

<sup>9</sup><https://automatizace.hw.cz/rotacni-ekodery-pro-rizeni-pohonu-a-pohybu.html>

Robot používa obvod reálneho času DS1307, ktorého prednosti sú najmä pre jeho presnosť a cena. Je riadený cez zbernicu I2C a pred prvým použitím je potrebné zapísať aktuálny čas. Aktuálny čas sa potom ukladá do EEPROM, čo je elektricky programovateľná pamäť, ktorá uchováva dáta aj po odpojení napájania. Všetky informácie o čase je možné nájsť v 8-bitových registroch pre každú jednotku (sekundy, minúty, hodiny, dni, mesiace, roky). Pamäť EEPROM obsahuje aj mnoho ďalších registrov pre uloženie vlastných dát.

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION	RANGE
00h	CH	10 Seconds			Seconds			Seconds	00-59	
01h	0	10 Minutes			Minutes			Minutes	00-59	
02h	0	12	10 Hour	10 Hour	Hours			Hours	1-12 +AM/PM 00-23	
		24	PM/ AM							
03h	0	0	0	0	0	DAY		Day	01-07	
04h	0	0	10 Date		Date			Date	01-31	
05h	0	0	0	10 Month	Month			Month	01-12	
06h	10 Year			Year			Year	00-99		
07h	OUT	0	0	SQWE	0	0	RS1	RS0	Control	—
08h-3Fh								RAM 56 x 8	00h-FFh	

Obr. 2.11: Význam jednotlivých bitov registrov uchovávajúcich čas v obvode reálneho času DS1307<sup>11</sup>

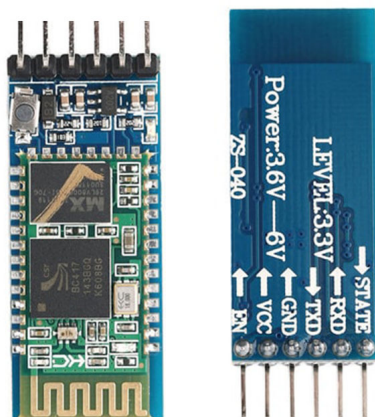
## 2.10 Bezdrôtová sériová komunikácia

Pre komunikáciu s mikrokontrolérom je veľmi bežná bezdrôtová komunikácia. Na procesore AVR ATMEGA32 sa táto komunikácia realizuje prostredníctvom USART, ktorý slúži k sériovému prenosu. Pre bezdrôtovú komunikáciu sú 2 možnosti a to buď pomocou wi-fi alebo Bluetooth. Wi-fi komunikácia má lepší dosah a aj rýchlosť prenosu a pri pripojení wi-fi robota napríklad do domácej siete je možné komunikovať s robotom kdekoľvek, kde je internetové pripojenie. Avšak správna konfigurácia wi-fi modulu je pomerne zložitá pre začiatočníka a sú potrebné aj základné znalosti sietí. V prospech Bluetooth modulu hrá aj cena a spotreba, ktorá je veľmi dôležitá pri napájaní z batérie [10].

Preto robot na komunikáciu používa Bluetooth, konkrétne modul HC-05, ktorý je veľmi obľúbený v robotike pre začiatočníkov, pre jeho jednoduchosť. Pre rýchlu a bezchybnú komunikáciu sa odporúča baudrate (rýchlosť prenosu) minimálne 9600 bd, čo procesor pri jeho základnej frekvencii nezvládne. Preto je použitý externý kryštál s frekvenciou 16MHZ. Pred prvým použitím je potrebné Bluetooth modul správne nakonfigurovať cez sériovú linku. Nastaviť je požadované meno, heslo a samozrejme baudrate na ktorej bude s procesorom komunikovať.

<sup>11</sup><https://datasheets.maximintegrated.com/en/ds/DS1307.pdf>

<sup>12</sup><https://simple-circuit.com/wp-content/uploads/2018/06/pic-microcontroller-i2c-lcd-circuit-pcf8574.png>



Obr. 2.12: Bluetooth modul HC05<sup>12</sup>

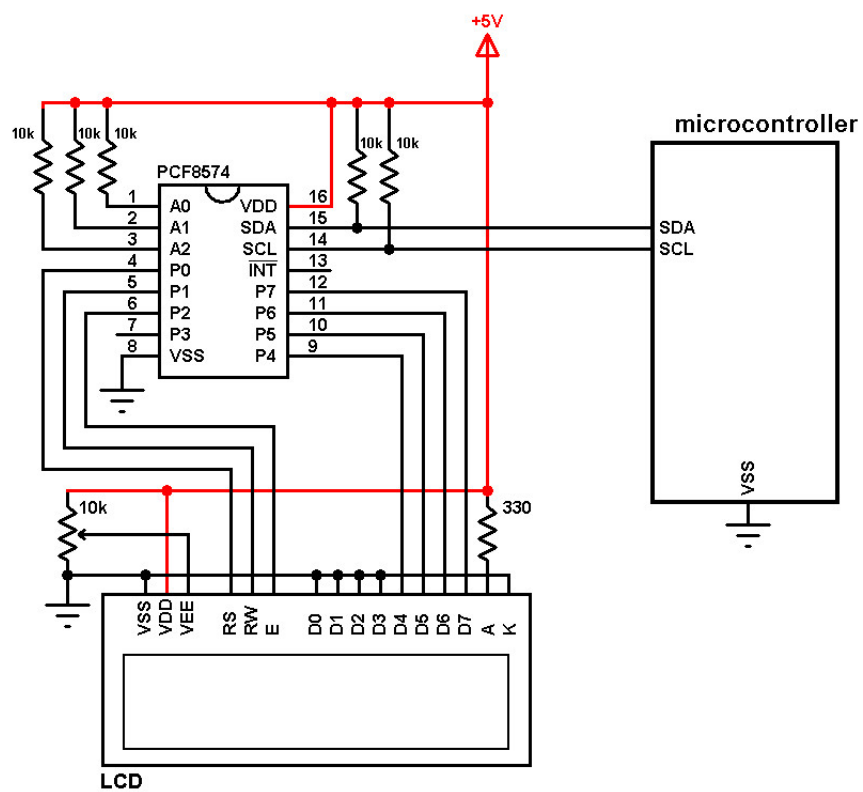
## 2.11 Displej a I2C modul

Na zobrazenie potrebných informácií o okolí a robotovi je vhodné použiť displej. Najpoužívanějšíe sú dva druhy dotykový a LCD. Dotykový displej má výhodu, že má lepšie rozlíšenie, zobrazuje viac informácií a ovláda sa dotykom. Avšak nevýhodou je, že zaberá veľké množstvo vývodov na mikrokontroléri a zároveň pre začiatočníkov je až príliš zložitý.

Preto je použitý LCD displej, konkrétne 4x20. čo znamená, že obsahuje 4 riadky a na každom riadku je možné napísať až 20 znakov. Displej má až 16 vývodov z toho najdôležitejšie sú 8 dátových, 1 pre určenie či sa znaky čítajú alebo zapisujú, 1 na určenie či sa bude čítať horných 8 dátových bitov alebo dolných a 2 na napájanie.

Pre správne fungovanie je použitých minimálne 6 I/O registrov na procesore. Avšak je možné použiť I2C modul PCF8574T, ktorý celú komunikáciu zoserializuje a tým pádom displej nezaberie na procesore ani jeden vývod ale iba jednu adresu v I2C zbernici.

<sup>13</sup><https://simple-circuit.com/wp-content/uploads/2018/06/pic-microcontroller-i2c-lcd-circuit-pcf8574.png>



Obr. 2.13: Zapojenie I2C modulu PCF8574 k displeju a následne k mikrokontroléru<sup>13</sup>

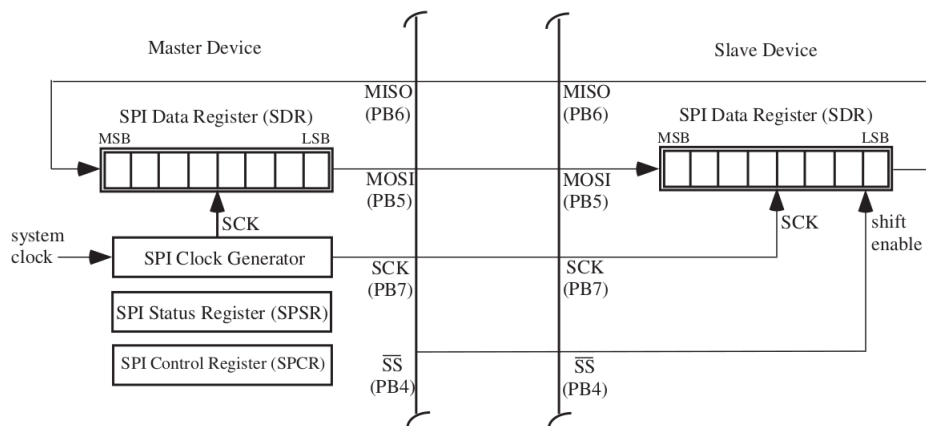
## Kapitola 3

# Funkčné bloky mikrokontroléra a ich registre

V tejto časti sú detailne popísané aké registre sú nevyhnutné pre prácu s danými funkčnými blokmi mikrokontroléra Atmega32. Jedná sa iba o funkčné bloky, ktoré využíva hardvérové vybavenie popísané v kapitole 2. V každej podkapitole je najskôr funkčný blok a jeho registre popísané z teoretického hľadiska a potom ako sa tieto registre používajú na naprogramovanie daného hardvérového prvku.

### 3.1 SPI

SPI (Serial Peripheral Interface) poskytuje obojsmernú komunikáciu medzi vysielačom a prijímačom. V SPI vysielač a prijímač zdieľajú hlavný zdroj hodín. To vyžaduje prídavnú linku so zdrojom hodín medzi vysielač a prijímač, avšak to umožní rýchlejší a efektívnejší prenos dát v porovnaní s USART.



Obr. 3.1: Pripojenie master a slave zariadenia [4]

SPI môže byť chápaný ako synchronný 16-bitový posuvný register s 8 bitmi nachádzajúcimi sa vo vysielači a so zostávajúcimi 8 bitmi vo prijímači. Vysielač je určený ako master, pretože poskytuje synchronizačný zdroj hodín. Prijímač je určený ako slave. Prijímač je

zvolený pripojením linky slave select (SS). SPI prenos je inicializovaný načítaním jedného bajtu dát do registra konfigurovaného ako SPI Data Register (SPDR). V čase ako generátor hodín poskytuje hodinové impulzy do master zariadenia, sú zároveň impulzy odosielané do slave zariadenia cez SCK pin. Bit je posunutý von z master zariadenia pomocou posuvného registru mikrokontroléra Master Out Slave In (MOSI) každý SCK impulz. Dáta sú prijaté na MOSI pine slave zariadenia. V rovnakom čase je jeden bit posunutý von z MISO (Master In Slave Out) pinu slave zariadenia na pin MISO master zariadenia. Po ôsmych SCK hodinových impulzoch je prenesený jeden bajt dát. Ukončenie prenosu dát v master zariadení a prijatie v slave zariadení je signalizované príznakom SPIF (SPI INterrupt flag) v obidvoch zariadeniach. Tento príznak sa nachádza v registri SPSR obidvoch zariadení [4].

Registre použité pre prácu s SPI sú [4]:

- SPCR - SPI kontrolný register pozostáva z :
  - SPE - vypínač zapnúť/vypnúť SPI systém
  - DORD - umožňuje posuv smeru dát od master do slave zariadenia
  - MSTR - bit určuje či SPI systém bude master alebo slave
  - CPOL - určuje nečinný stav SCK pinu
  - CPHA - určuje či dáta budú vzorkované na nábežnej alebo zostupnej hrane signálu SCK
- SPIF - Status Register. Tento príznak určuje kedy budú dáta prenesené od master do slave zariadenia
- SPDR - SPI dátový register. Obsahuje jeden bajt dát.

Prenos dát pomocou SPI sa používa pri programovaní mikrokontroléra Atmega32. Programátor je implementovaný na doske plošného spoja. Jeho hlavný prvok, pomocou ktorého komunikuje s mikrokontrolérom Atmega32 je mikrokontrolér AVR ATmega8. Atmega32 vystupuje ako slave zariadenie a Atmega8 ako master.

## 3.2 Vonkajšie vstupy prerušenia

vonkajšie prerušenia sú spustené vstupmi INT0 až INT2. Tieto vstupy sú ovládané pomocou registrov GICR (masky prerušenia) MCUCR (voľba aktivácie) a GIFR (príznačky prerušenia).

Pomocou maskovacích bitov registra GICR je možné povoľovať alebo zakazovať prerušenie pomocou vonkajších vstupov INT0 až INT2. Napríklad bit INT0 povoľuje(1) alebo zakazuje (0) aktiváciu prerušenia pomocou vstupu INT0. Pomocou príznakových bitov registra GIFR sa udržiavajú požiadavky aktivácie vonkajších prerušovacích vstupov v okamžiku, kedy je prerušenie zakázané buď odstavením masky (GICR) alebo globálnym zákazom prerušenia. Pomocou bitov registra MCUCR je možné konfigurovať podmienku, ktorá aktivuje prerušenie vonkajším vstupom. S ohľadom na prerušenia majú v registri MCUCR význam iba bity ISC00,ISC01,ISC10 a ISC11, ktoré nastavujú INT0 a INT1 [11].

Pre prerušenie na vstupe INT2 sa nastavuje bit ISC2 v registri MCUCSR. AK ISC2 = 0, tak vstup je citlivý na vzostupnú hranu a pre ISC2 = 1 je vstup INT2 citlivý na nábežnú hranu. Prerušenia pomocou vonkajších vstupov využíva enkodér a ultrazvukový dialkomer.

Pre enkodér je na vývojovej platforme vyvedený výstup INT2 a tým pádom sa nastaví tento bit v registri GICR na logickú 1. Enkodér aktivuje prerušenie vždy na nábežnú hranu.

ISC01	ISC00	Popis
0	0	vstup INTO sa aktivuje log. 0 (úrovňovo citlivý vstup)
0	1	akákoľvek zmena stavu INTO generuje prerušenie
1	0	vstup INTO sa aktivuje zostupnou hranou
1	1	vstup INTO sa aktivuje nábežnou hranou

Tabuľka 3.1: Citlivosť vstupu INTO [11]

Takže v MCUCSR sa nastaví ISC2 na logickú 1. V obsluhu prerušenia potom postačí vždy pri jeho vyvolaní zvyšovať počítadlo. Keďže k použitému motoru a teda aj jeho enkodéru je málo informácií, tak prevod na otáčky bol robený vlastnými meraniami. Z meraní sa zistilo, že 1 otočenie motora spôsobí 400 prerušení.

Pre ultrazvukový senzor sa bit INTO nastaví na logickú 1 v registri GICR. Pri ultrazvukovom senzore je potrebné detegovať nábežnú a aj zostupnú hranu, čiže akákoľvek zmena stavu. V MCUCR sa nastaví ISC00 na logickú 1 a ISC01 na logickú 0. V obsluhu prerušenia potom sa potom pri nábežnej hrane spustí časovač a pri zostupnej hrane sa zastaví. Potom z čísla v časovači sa zistí šírka impulzu a teda aj vzdialenosť od predmetu.

### 3.3 A/D prevodník

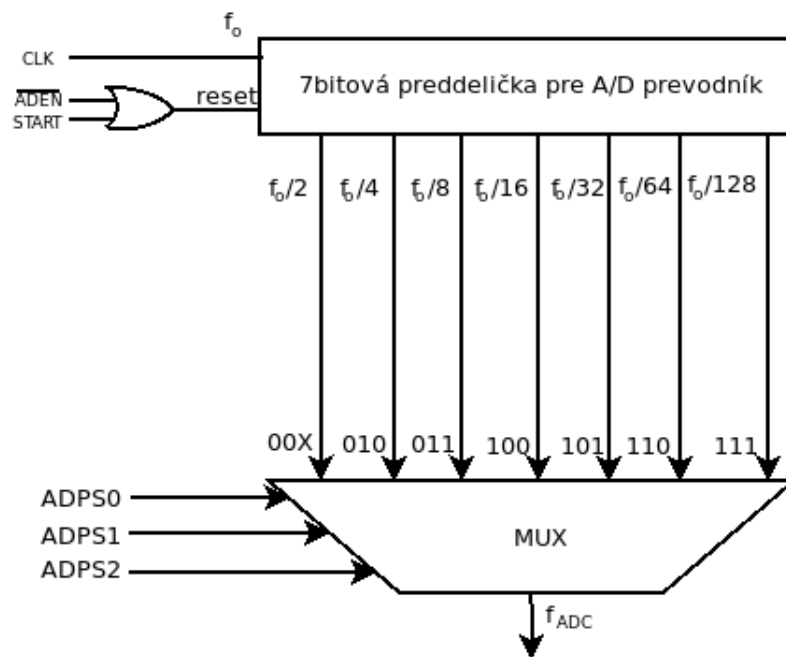
A/D prevodník prevádza vstupné analógové napätie na 10bitovú číslicovú hodnotu algoritmom postupnej aproximácie. Minimum je reprezentované AGND a maximum AREF. Je možné voliť medzi referenčným napätím AREF z vonkajšku, to je napätím vývodu AUcc a vnútorným referenčným napätím 2,56 V. Voľba sa urobí pomocou bitov REFS0 a REFS1, ktoré sú v registri ADMUX. Vstupné analógové kanály a zisk sa volí zápisom bitov MUX0 až MUX4 do registru ADMUX. Činnosť A/D prevodníka sa povoľuje bitom ADEN z registru ADCSRA. Voľba výstupného kanálu a referenčného zdroja nemá vplyv na ADEN. A/D prevodník generuje 10bitový výsledok, ktorý sa ukladá do registrového páru ADCH:ADCL [11].

A/D prevodník môže byť spustený viacerými spôsobmi [11]:

- Jednoduchý prevod - každý prevod musí byť odštartovaný užívateľom zápisom 1 do bitu ADSC z registru ADCSRA. Tento bit zostane nastavený tak dlho, pokiaľ sa prevod nedokončí.
- Automatické spustenie rozličnými zdrojmi prerušení - autospúšťanie je povolené nastavením bitu ADATE z registru ADCSRA. Spúšťač zdroj je možné vybrať pomocou bitov ADTS0 až ADTS2 z registru SFIOR.
- Voľný beh - ako zdroj automatickej aktivácie je možné tiež zvoliť príznak ADIF (indikuje koniec prevodu). V tomto prípade dôjde k spusteniu nového prevodu hneď ako sa dokončí predchádzajúci. Prvý prevod sa musí odštartovať ručne, nastavením bitu ADSC z registru ADCSRA.

Postupná aproximácia potrebuje pre dosiahnutie maximálnej presnosti vstupný hodinový kmitočet v rozsahu 50 až 200 kHz. K výberu hodinového kmitočtu FADC sa používajú 3 spodné bity ADPS0 - ADPS2 z registra ADCSRA. Preddelička začne čítať od chvíle kedy je A/D prevodník zapnutý nastavením bitu ADEN z registra ADCSR. Preddelička zostáva bežať tak dlho, pokiaľ sa nevynuluje bit ADEN [11].





Obr. 3.2: Preddelička A/D prevodníka [11]

A/D prevodníky sa používajú na získanie dát z infračerveného diaľkomera, senzora na sledovanie čiary a napätia na batérii. Inicializácia A/D prevodníka prebehne tak, že do registra ADMUX sa na bit REFS0 zapíše logická 1, čím sa vyberie referenčné napätie ako vonkajšie. Pre dosiahnutie vhodnej presnosti pre postupnú aproximáciu je potrebné použiť preddeličku. Zdrojová frekvencia hodín je 16 MHz, takže vhodná preddelička je 128, tým sa FADC zníži na 125kHz. Na nastavenie tejto preddeličky je potrebné zapísať do registra ADCSRA logickú 1 na bity ADEN, ADPS0, ADPS1 a ADPS2.

Pre infračervený diaľkometer je na vývojovej platforme vyhradený kanál 5 (PINA5). Príslušný kanál sa vyberie zápisom logickej 1 na bity MUX2 a MUX0 registra ADMUX. Pre aktivovanie čítania hodnoty na tomto kanále sa zapíše do registra ADCSRA logická 1 na bit ADSC a potom už len stačí počkať kým sa na tomto bite objaví logická 0, čím končí úspešne čítanie. 10bitová hodnota sa potom nachádza v registri ADC. Tú už stačí previesť na požadovanú vzdialenosť.

Pre senzor na sledovanie čiary sú použité kanály 0 až 2. Hodnoty na príslušných kanáloch je potrebné čítať po jednom, zápisom do registra ADMUX kombináciou bitov MUX0 až MUX2. Ďalej aktivácia A/D prevodníka a získanie hodnoty je taká istá ako v prípade infračerveného diaľkomera. Ďalej už stačí len vyhodnotiť čísla a na základe týchto čísiel rozhodnúť, kde sa nachádza čierna čiara.

Pre meranie napätia na batérii je vyhradený kanál 6. Výber kanálu a aktivácia je taká istá ako v prechádzajúcich prípadoch. Z dôvodu použitia napäťového deliča, napätie na danom pине neodpovedá skutočnému napätiu. Napäťový delič znížil toto napätie presne o 3 násobok, preto treba previesť 10bitovú hodnotu na napätie a potom vynásobiť číslom 3.

### 3.4 Generovanie signálu pomocou pulzne šírkovej modulácie

Signál pomocou pulzne šírková modulácie (PWM) sa vytvára cez časovače/čítače. Mikrokontrolér AVR Atmega32 obsahuje 3 časovače/čítače a to 2 8bitové a 1 16bitový. Pre generovanie čo najpresnejších impulzov je použitý 16bitový časovač/čítač [16].

Najdôležitejšie registre pre prácu s vybraným časovačom/čítačom sú [11]:

- Riadiaci register TCCR1A obsahuje bity:
  - COM1A1, COM1A0, COM1B1, COM1B0 - riadia chovanie vývodov OC1A alebo OC1B. Ak je jeden z týchto bitov nastavený, je prekrytá normálna funkcia vstupno/výstupného vývodu výstupom OC1A alebo OC1B.
  - FOC1A, FOC1B – vynútenie OC1A, OC1B výstupu. Pri PWM režime musí byť FOC1X nastavené na logickú 0
  - WGM11, WGM10 – režim generovaného priebehu. Spolu s bitmi WGM12 a WGM13 riadi čítanú sekvenciu, volí maximálnu čítanú hodnotu a typ priebehu generovaného signálu. Jedná sa o normálny režim, CTC režim, rýchly PWM režim a fázovo korigovaný PWM režim.

COM1A1	COM1A0	Popis
0	0	Časovač/Čítač odpojený od vývodu OC1A
0	1	Pri rýchlom PWM režime s vrcholom v OCR1A nastane negácia pri zhode, inak je vývod odpojený
1	0	OC1A = 0 po zhode, OC1A = 1 pri dosiahnutí vrcholu
1	1	OC1A = 1 po zhode, OC1A = 0 pri dosiahnutí vrcholu

Tabuľka 3.2: Význam bitov COM1A1 a COM1A0 v rýchlom PWM režime [11]

- Riadiaci register TCCR1B obsahuje bity:
  - ICNC1 - potlačovač šumu pre jednotku IC
  - ICES1 - výber hrany pre jednotku IC. Pre ICES1 = 0 sa pre záchyt použije zostupná hrana ICP1, inak sa použije nábežná hrana ICP1.
  - WGM13, WGM12 - režim generovaného priebehu
  - CS12 až CS10 - výber hodín. Je možné vybrať hodinový signál odvodený od kmitočtu mikrokontroléru alebo hodinový signál privedený na vývod T1.
- Register TCNT1 - Jedná sa o 16bitový aktuálny stav časovača/čítača. Zápis do registru TCNT1 blokuje udalosť zhody TCNT1 = OCR1X (X = A/B) na jeden hodinový cyklus čítača. Zmena obsahu TCNT1 v priebehu čítania je teda riskantné a môže viesť ku strate udalosti zhody TCNT1 = OCR1X.
- Register ICR1 – register jednotky IC. Jedná sa o 16bitový záchytný register.
- Registre OCR1A, OCR1B - jedná sa o 16bitové komparačné registre.
- Register TIMSK – masky prerušenia. Významné bity sú:
  - TICIE1 – povolenie prerušenia pri záchyte obsahu

CS12	CS11	CS10	Popis
0	0	0	stop, časovač/čítač 1 je zastavený
0	0	1	$f_{clk}$ (hodiny mikrokontroléra)
0	1	0	$f_{clk}/8$
0	1	1	$f_{clk}/64$
1	0	0	$f_{clk}/256$
1	0	1	$f_{clk}/1024$
1	1	0	zostupná hrana T1
1	1	1	nábežná hrana T1

Tabuľka 3.3: Význam bitov CS12 až CS10 v registri TCCR1B [11]

- OCIE1X – povolenie prerušenia po zhode
- TOIE1 – povolenie prerušenia pri pretečení obsahu TCNT1

Najpoužívanejší spôsob na generovanie PWM je riadiť chovanie výstupov OC1A/OC1B na základe zhody registra TCNT1 s registrom ICR1. Avšak v tom prípade by PWM signál mohol byť generovaný len na 2 výstupoch avšak je potrebných až 9 PWM výstupov - 6 pre rameno, 2 pre riadenie rýchlosti motorov a 1 pre zatáčanie predných kolies. Preto sa použije metóda, ktorou je možné generovať PWM signál na akomkoľvek vstupno/výstupnom pine.

Pre generovanie je zvolený rýchly PWM režim a vrchol závisí od obsahu záchytného registra ICR1, ktorý určuje periódu signálu. Takže sa nastaví WGM11, WGM12 v registri TCCR1A a WGM13 v registri TCCR1B. V prípade regulovania rýchlosti pre motory, dĺžka periódy nie je až taká podstatná, avšak pri zatáčaní kolies a pri ramene sa jedná o servomotor, pri ktorých musí byť perióda nastavená presne na 20 milisekúnd. Číslo, ktoré má byť uložené do záchytného registra ICR1 sa vypočíta nasledovne:  $ICR1 = f_{CPU} * perioda - 1$ . Výpočtom vyjde číslo 319999. Keďže register ICR1 je 16bitový, najväčšie možné číslo, ktoré sa dá uložiť do tohto registra je 65535. Preto sa použije preddelička, ktorá zníži frekvenciu časovača 8násobne. Na to je potrebné zapísať logickú 1 na bit CS11 v registri TCCR1B. Ďalej je potrebné povoliť prerušenie vždy pri dosiahnutí čísla ICR1. To sa docieli zápisom bitu TICIE1 do registra TIMSK.

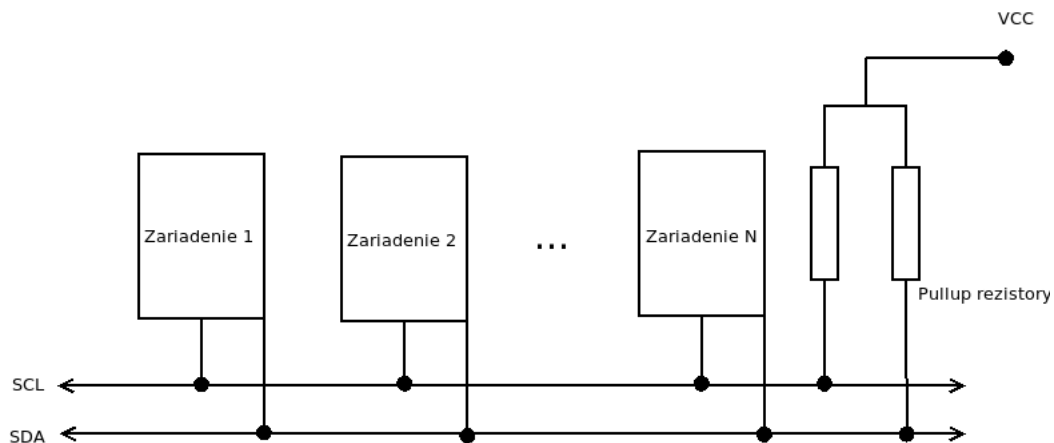
Princíp generovania PWM je nasledovný:

1. Na začiatku sa zapíše logická 1 na vybraný pin.
2. V momente ako dosiahne register TCNT1 požadované číslo (0 – 0V 39999 – 5V) sa tento bit prepíše na logickú 0.
3. ndenurate Pri dosiahnutí periódy (číslo 39999) sa vyvolá prerušenie
4. Vynuluje sa register TCNT1 a zapíše sa znova logická 1 na daný bit. Pokračuje sa bodom 2.

Pri servomotoroch je dĺžka striedy 1ms až 2ms. Z toho vyplýva, že v bode 2 sa bude pracovať z číslami 1999 až 3999.

### 3.5 Rozhranie TWI

Sériové rozhranie TWI je ideálne použiteľné pre komunikáciu s mikrokontrolérom a až 128 rôznymi zariadeniami za použitia iba 2 liniek zbernice SCL (hodiny) SDA (dáta). Dané rozhranie je plne kompatibilné so známejšou zbernicou I2C. Vonkajšie zariadenie musí pripojiť iba jeden pull-up rezistor na každú z oboch liniek. Všetky zariadenia pripojené na zbernici majú individuálne adresy, mechanizmus pripojovania ku zbernici je definovaný samotným TWI protokolom. Každý dátový bit prenášaný po TWI zbernici je spojený s jedným impulzom hodín. Úroveň na dátovej linke SDA musí byť stabilná v okamžiku kedy SCL = 1. Toto pravidlo platí okrem prípadu, že sa generuje tzv. START alebo STOP stav [11].



Obr. 3.3: Pripojovanie zariadení na TWI (I2C) zbernicu

Všetky adresné pakety sú na TWI zbernici o dĺžke 9 bitov. Skladajú sa zo: 7 bitovej adresy, riadiaceho bitu R/W a potvrdzovacieho bitu ACK. Ak R/W = 1, jedná sa o čítanie a ak R/W = 0, jedná sa o zápis. Keď slave rozpozná, že je adresovaný, mal by stiahnuť SDA linku v okamihu potvrdzovacieho bitu ACK. Ak je adresovaný slave zaneprázdnený alebo keď žiadne zariadenie neopovedá, je linka SDA ponechaná v logickej 1. Master potom môže vyslať STOP stav alebo opakovať START stav po inicializácii nového prenosu [12].

Dátové pakety sa po TWI zbernici vysielajú tiež v dĺžke 9 bitov. Skladajú sa z 8bitového dátového údajá a potvrdzovacieho bitu ACK. V 8bitových dátach sa ako prvý posiela najviac významný bit. Každý úspešne prijatý bajt je zo strany prijímača potvrdený stiahnutím linky SDA v 9. cykle hodín (ACK) [12].

Registre spojené so zbernicou TWI sú [11]:

- TWBR - register prenosovej rýchlosti TWI. Tento register riadi periódu hodín SCL v prípade režimu master
- TWCR - riadiaci register TWI. Používa sa na riadenie operácií na TWI. Je použitý pre povolenie funkcie TWI, inicializáciu master prístupu START, generuje ACK prijímača, STOP stav a riadi zastavenie zbernicových operácií, keď dáta pre zápis na zbernici sú zapisované do registra TWDR. Bity použité v tomto registri sú:
  - TWINT - príznak prerušenia TWI
  - TWEA - povolenie potvrdzovania (ACK) na TWI
  - TWISTA - START stav

- TWISTO - STOP stav
  - TWWC - príznak kolízie zápisu
  - TWEN - povolenie funkcie TWI
  - TWIE - povolenie prerušenia TWI
- TWSR – stavový register. Obsahuje 5 bitov, ktoré odrážajú aktuálny stav TWI logiky a zbernice a 2 bity preddeličky prenosovej rýchlosti.

Operácia	Symbol	Stavový kód
START stav	START	\$08
vysielanie slave adresy, ACK prijatý	MT_SLA_ACK	\$18
vysielanie slave adresy, NACK prijatý	MT_SLA_NACK	\$20
vysielanie dát, ACK prijatý	MT_DATA_ACK	\$28
vysielanie dát, NACK prijatý	MT_DATA_NACK	\$30
strata dát	MT_LOSE	\$38

Tabuľka 3.4: Stavové kódy pre master vysieláč [11]

- TWDR - dátový register TWI. Vo vysielacom režime obsahuje ďalší bajt pre odoslanie. V prijímacom režime obsahuje posledný prijatý bajt. Dáta v tomto registri zostávajú stabilné tak dlho, pokiaľ TWINT = 1.
- TWAR - adresový register. Tento register obsahuje 7bitovú adresu slave obvodu, na ktorú bude TWI odpovedať pri naprogramovaní ako slave. Register ešte obsahuje bit TWGCE, ktorý sa používa na rozlíšenie broadcast adresy. Pripojený adresný komparátor generuje pri zhode adresy prerušenie.

Na inicializáciu TWI rozhrania je potrebné nastaviť register TWSR na 0, čím sa vyberie, že sa nepoužije žiadna preddelička a vhodná prenosová rýchlosť sa nastaví do registra TWBR. Na odoslanie alebo získanie informácií z I2C zariadenia sa najskôr musí poslať START stav, počká sa kým je prenos kompletný a skontroluje sa, či sa podarilo úspešne pripojiť k zariadeniu. Potom sa odošle adresa zariadenia, ku ktorému sa pristupuje, počká sa na úspešné odoslanie a overí sa či všetko prebehlo úspešne. Celý tento proces odráža funkcia `i2c_start`. Na získanie dát slúžia funkcie `i2c_readNak`, ktorá získa 1 bajt a ukončí získavanie dát a funkcia `i2c_readAck`, ktorá tiež získa 1 bajt, avšak požaduje potom ďalšie dáta. Získaný bajt je uložený v registri TWDR. Na zápis dát slúži funkcia `i2c_write`. Dáta sa taktiež zapisujú do registra TWDR.

Získavanie dát pomocou funkcie `i2c_readAck` sa používa pri obvode reálneho času a senzore IMU. Naopak zápis dát pomocou funkcie `i2c_write` sa používa na riadenie displeja. Podrobný popis ako sa získavajú a zapisujú dáta z týchto sensorov a displeja, je knižnica pre daný modul.

### 3.6 Jednotka USART

Univerzálny synchronný a asynchronný sériový prijímač a vysieláč je vysoko flexibilné zariadenie pre sériovú komunikáciu. Kľúčové vlastnosti sú [11]:

- Plný duplex

- Synchronný alebo asynchronný režim
- Pri synchronnom režime môže pracovať ako master
- Podpora dĺžky 5 až 9 dátových bitov a 1 až 2 stop bity
- Generátor párnej/nepárnej parity pre vysieláč a hardvérové testovanie parity pre prijímač
- Detekcia straty znaku
- Detekcia chyby rámca
- Možnosť zdvojnásobiť prenosovú rýchlosť v asynchronnom režime

Jednotka USART zakomponovaná do mikrokontroléru ATmega32 je plne kompatibilná s jednotkami UART, ktoré boli k dispozícii v starších modeloch mikrokontrolérov AVR. Je zachované umiestnenie bitov, generátor prenosovej rýchlosti, operácie vysieláča a prijímača. Registre potrebné pre prácu s USART sú [11]:

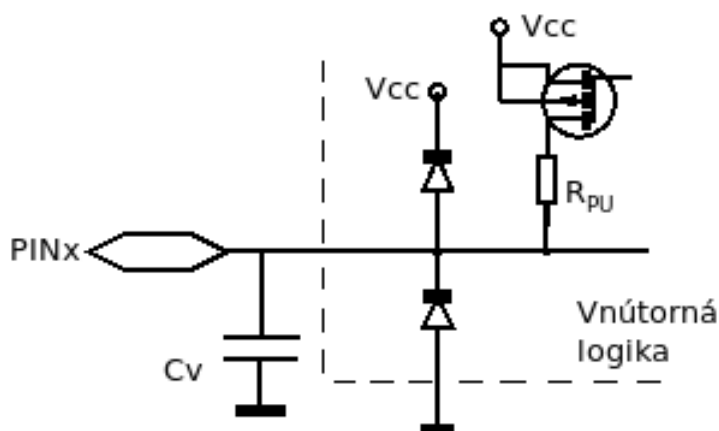
- UDR – v/v dátový register. Dátový buffer prijímača a vysieláča zdieľa rovnakú adresu označenú ako register UDR. Pre prípad 5,6 a 7bitového znaku sú najvyššie bity vysieláčom ignorované a prijímačom vynulované. Do vysieláča je možné zapisovať iba ak je nastavený príznak UDRE z registra UCSRA.
- UCSRA – riadiaci a stavový register A. Register UCSRA obsahuje predovšetkým príznaky komunikácie a chyby príjmu.
- UCSRB – riadiaci a stavový register B. Tento register obsahuje predovšetkým bity povoľujúce prerušenia alebo činnosť rôznych blokov jednotky USART.
- UCSRC – riadiaci a stavový register C. Register UCSRC obsahuje bity pre voľbu režimu jednotky USART.
- UBRRH, UBRRH – register prenosovej rýchlosti. Tento pár registrov slúži pre nastavenie prenosovej rýchlosti.

Jednotka USART sa používa pre komunikáciu medzi robotom a smartphonom pomocou Bluetooth modulu HC-05. Ako prvé sa v inicializácii nastaví prenosová rýchlosť. V Bluetooth module je prednastavená na 9600 Bd, čo je vhodná pre frekvenciu procesora. Prenosová rýchlosť sa zapisuje do registra UBRR a vypočíta sa ako:  $UBRR = f_{CPU}/(16 * Baudrate) - 1$ . Ďalej je potrebné nastaviť pin Tx ako výstupný a Rx ako vstupný. V registri UCSRB sa povolí vysieláč TXEN, prijímač RXEN a prerušenia pri prijímaní RXCIE. Potom už sa len zvolí počet dátových bitov rámca pre prijímač a vysieláč, ktorých bude 8. To sa zvolí zápisom bitov UCSZ0 a UCSZ1 do registra UCSRC.

Pre odoslanie znaku (8 bitov) do Bluetooth modulu sa najskôr musí počkať kým bude buffer prázdny a potom je potrebné zapísať tento znak do registra UDR. Pri prijatí znaku sa vždy vyvolá prerušenie, počká sa kým je prenesený celý bajt, ktorý sa potom nachádza v registri UDR. Na základe znaku alebo viacerých znakov sa rozhodne o pohybe robota, prípadne ramena. Jednotlivé príkazy pre robota je potrebné oddeliť znakom nového riadku.

### 3.7 Vstupné/Výstupné porty

Vstupnými/Výstupnými portami sa rozumie skupina vývodov, ktorá je z hľadiska mikrokontrolérov chápaná ako celok. Mikrokontrolér ATmega32 disponuje štyrmi 8bitovými portami. Každý port obsahuje 8 vstupno/výstupných portov a všetky majú rovnaké vlastnosti, avšak môžu sa odlišovať napojením na rôzne zabudované periférie. Porty majú tzv. skutočnú Read - Modify - Write funkcionality, v prípade, že sú použité ako obecné číslicové vstupy/výstupy. To znamená, že smer jedného vývodu portu je možné meniť bez nežiaducej zmeny smeru ostatných vývodov. Podobne je možné meniť vybudovaný stav alebo pripojenie/odporenie pull-up rezistoru. Všetky vývody majú individuálne aktivovateľný pull-up rezistor a ochrannú diódu. Mnoho vývodov má zavedený multiplex s alternatívnymi funkciami vývodov (napríklad pre použitie vonkajšieho prerušenia, SPI, TWI apod.) [11].



Obr. 3.4: Vnútorná logika vstupného pinu [11]

Vstupné piny sa používajú na identifikáciu stlačenia tlačidla. Na robotovi sa nachádzajú 4 tlačidlá a tieto piny sú konfigurované ako vstupné a stlačením sa na pine objaví logická 0. Z toho vyplýva, že ak tlačidlo nie je stlačené musí byť na tomto pine stála hodnota a to logická 1. To sa zaisťuje pripojením zabudovaného pull-up rezistora na daný pin.

Piny konfigurované ako výstupné sa používajú na riadenie elektromotorov. Pre každý motor sa používajú 2 piny. Tieto piny sú zapojené na H-mostík a ich kombináciou sa volí smer motora (viď tabuľka).

PINC4	PINC5	Stav motora
0	0	STOP
0	1	VPRED
1	0	VZAD
1	1	STOP

Tabuľka 3.5: Stav motora na základe stavu výstupných pinov

# Kapitola 4

## Elektronika

Táto kapitola zahŕňa návrh schémy zapojenia a následne na základe toho samotný návrh dosky plošného spoja a jej výroba. Pred návrhom schémy je potrebné určiť, čo všetky by mala obsahovať. To sa určí na základe hardvérového vybavenia 2, čiže doska musí obsahovať jednoduchú možnosť pripojenia týchto zariadení, najlepšie bez použitia kontaktného pola alebo iných nadstavbových prvkov. Analýza konkurenčných riešení a zistenie ich výhod a nedostatkov je tiež veľmi dôležité. Výsledkom celého tohto procesu je vývojová platforma, na ktorej bude bežať robot.

### 4.1 Analýza konkurencie a riešenie konkurenčných nedostatkov

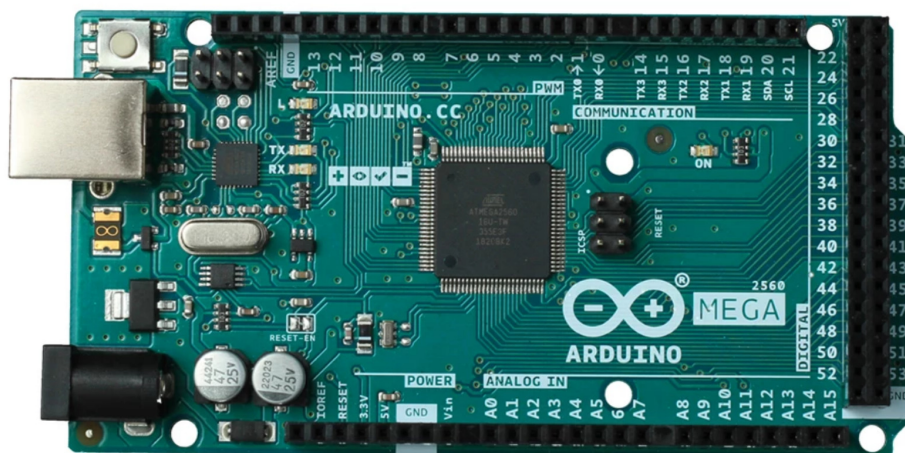
Najznámejšie konkurenčné riešenia sú od spoločností Arduino a NXP. Detailnejšie boli otestované modely Arduino Mega2560 ale FRDM KL-25Z od NXP. Tieto konkurenčné riešenia ponúkajú mikrokontrolér vo forme SMD. Jeho výhodou je, že zaberá menej miesta a ponúka vyšší výkon. Nevýhodou je o niečo vyššia cena a pri poškodení mikrokontroléra je nevyhnutné vymeniť celú dosku. Preto je na vývojovej platforme použitý mikrokontrolér Atmega32, ktorý je osadený do päťice. Ďalším dôvodom prečo je použité toto riešenie je, že vyrobiť a osadiť dosku s SMD procesorom je v domácich podmienkach takmer nemožné. Preto by bolo potrebné k tomu využiť firmu, ktorá sa na to špecializuje, ale tým by sa celá práca výrazne predražila.

V prípade konkurenčných riešení je veľkým nedostatkom príliš málo vývodov pre napájanie, väčšinou maximálne 3, čo znamená že pri použití viac ako 3 zariadení je užívateľ nútený použiť kontaktné prepojovacie pole, prípadne použiť univerzálnu dosku plošného spoja. Preto je tento problém vyriešený tak, že každý vývod alebo skupina vývodov na ktoré sa dá pripojiť nejaké zariadenie, bude mať vlastné napájanie.

Ďalším nedostatkom konkurenčných riešení je, že pri použití napájania z batérie je k dispozícii iba napájanie 3,3V s nízkym maximálnym prúdom z dôvodu slabého stabilizátora, čo je napríklad pre servomotory veľmi málo. Pri napájaní z USB portu je k dispozícii aj 5V napájanie, čo už je dostačujúce, avšak počítač poskytuje prúd, ktorý by postačil na napájanie maximálne 2 servomotorov. Toto riešenie je zároveň nepraktické, pretože je nutné mať stále pripojený kábel k počítaču. Čiže zostáva jediné riešenie a to opäť použiť kontaktné pole alebo DPS a vlastný stabilizátor. Preto sa k doske bude stabilizátor pripojovať externe a maximálny prúd závisí len na použítom stabilizátore. Nevýhodou je, že stabilizátor musí



byť na napätie 5V, pretože sa z neho napája aj mikrokontrolér. Vhodné je použiť stabilizátor aspoň 5V/10A, ktorý by mal zvládnuť aj viac zariadení s väčším odberom prúdu.



Obr. 4.1: Najznámejšie konkurenčné riešenie Arduino Mega 2560 Rev3<sup>1</sup>

## 4.2 Hlavné prvky schémy zapojenia

Pre ovládanie až 2 motorov je vyvedených 6 pinov, ktoré budú výhradne slúžiť na pripojenie externého H-mostíka. Z týchto 6 pinov slúžia 2 na ovládanie rýchlosti otáčok motorov (PWM), ostatné slúžia na určenie smeru otáčania (bežné I/O piny).

Pre pripojenie senzoru na sledovanie čiary je vyvedených 5 vývodov (A/D prevodníky), čiže je možné pripojiť až 5 senzorov, ktoré dokážu určiť farbu okolia.

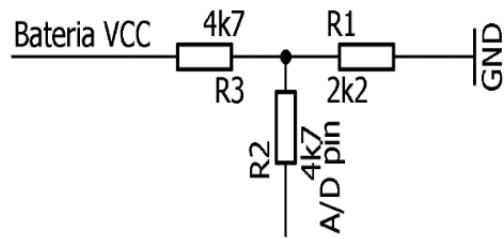
Pre ostatné senzory a súčiastky popísané v sekcii 2, sú vyvedené vývody pre sériovú komunikáciu (UART), I2C, A/D prevodníky a bežné I/O piny. I/O piny sú použité aj na pripojenie tlačidiel prípadne softvérovou úpravou sa dajú použiť aj ako PWM.

Z dôvodu, že základná frekvencia procesora je len 1MHz, čo je pre rýchlu a spoľahlivú bezdrôtovú komunikáciu veľmi málo je ešte použitý externý kryštál s frekvenciou 16MHz. Pred prvým použitím procesora je preto nutné zmeniť jeho poistky podľa návodu popísaného v technickej dokumentácii procesora.

Meranie napätia na batérii nie je možné realizovať jednoduchým pripojením batérie na A/D prevodník, pretože použitá batéria má pri plnom nabití napätie až 12,1V, čo je pre A/D prevodník procesora príliš vysoké napätie. Povolené maximum je 5V. Preto je potrebné použiť dvojicu, prípadne trojicu odporov pre zníženie napätia na požadovaný rozsah. Pri použití 2 odporov 4K7 a jedného 2K2 zapojené podľa obrázka 4.2, sa zníži napätie približne o trojnásobok, čo už je možné odmerať A/D prevodníkom aj pri plnom nabití batérie.

Pre programovanie procesora sú 2 možnosti. Vyviest vývody potrebné na programovanie ku ktorým sa bude pripájať programátor alebo implementovať programátor už do DPS. Doska obsahuje druhé riešenie, pretože prvé je veľmi nepraktické, pretože užívateľ si musí zakúpiť programátor a dobre nastudovať ako ho pripojiť k doske. Na schéme zapojenia je implementovaný programátor USBASP s procesorom AVR ATMEGA8 podľa návodu na stránke: <https://www.fischl.de/usbasp/>, takže dosku s počítačom stačí prepojiť dátovým

<sup>1</sup><https://amperka.com/products/arduino-mega-2560>



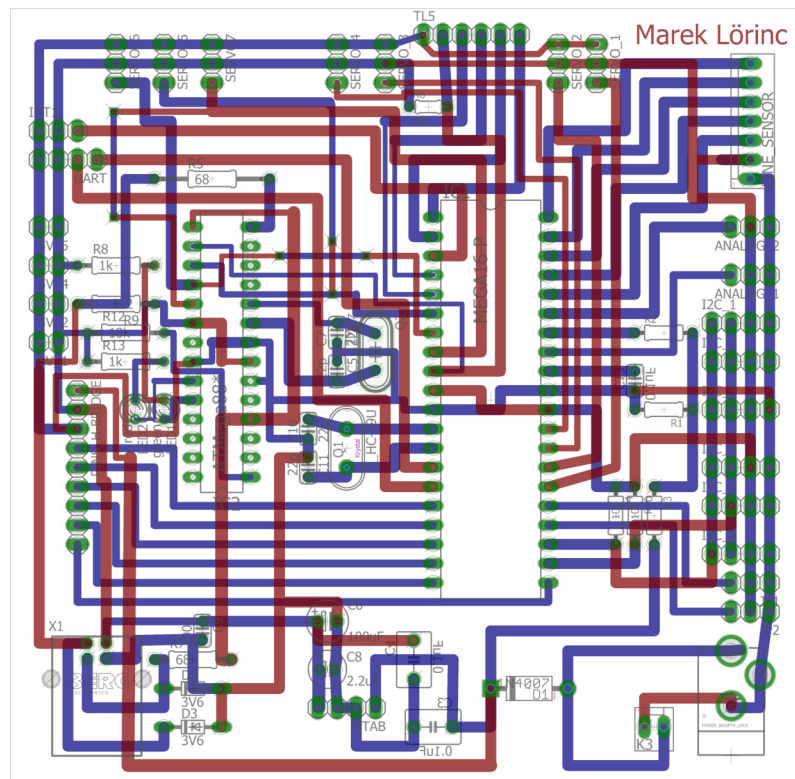
Obr. 4.2: Zapojenie rezistorov použité na zmeranie napätia na batérii

káblom USB A-B a následne na naprogramovanie procesora stačí už len použiť príslušný softvér (ExtremeBurner, Atmel Studio).

Návrh schémy zapojenia prebiehal v programe Cadsoft EAGLE 7.6.0. Výhodou tohto programu je, že sa dá použiť aj na návrh dosky plošného spoja, ktorý už obsahuje potrebné prepojenia na základe schémy, čím sa výrazne znižuje možnosť chyby a aj urýchli celú prácu. Výsledná schéma zapojenia sa nachádza v prílohe A.

### 4.3 Návrh dosky plošného spoja

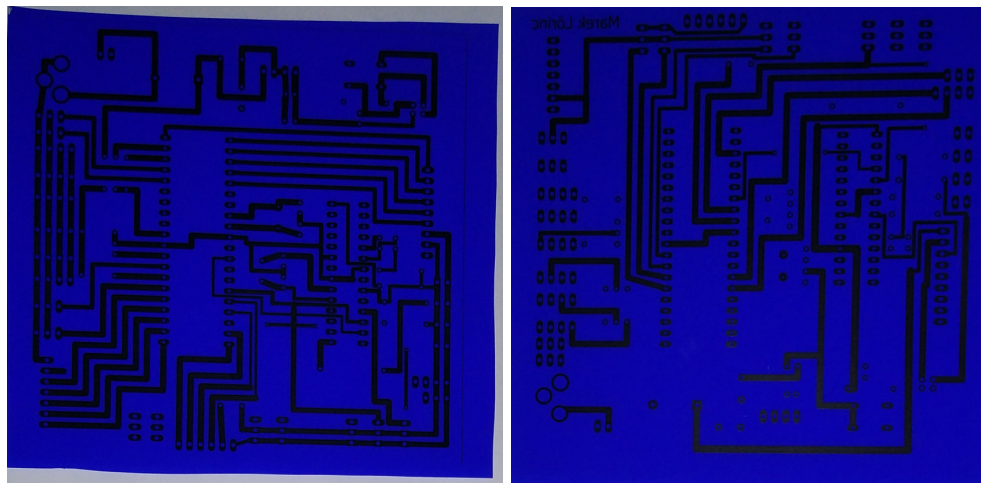
Návrh dosky plošného spoja sa vytváral na základe schémy zapojenia, čiže najdôležitejšou časťou je správne rozloženie súčiastok a vytvorenie prepojení tak, aby dosku bolo možné vytvoriť v domácich podmienkach. Aby výsledná doska bola čo najmenšia, bol použitý obojstranný návrh dosky. Výsledne rozmery dosky sú 12cm x 12cm.



Obr. 4.3: Obojstranný návrh dosky plošného spoja

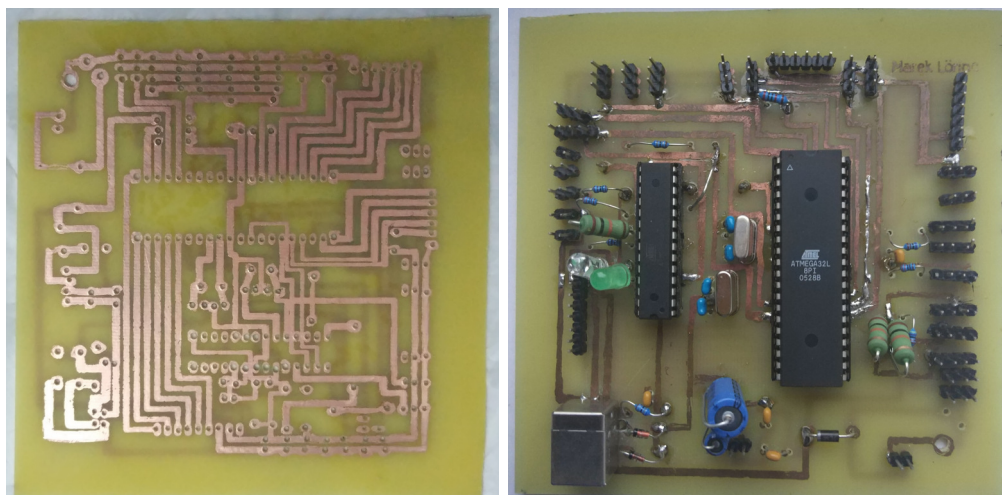
## 4.4 Výroba dosky plošného spoja

Výroba dosky plošného spoja prebiehala pomocou nažehlovacej metódy. Pre zvýšenie úspešnosti tejto metódy bola miesto bežného papiera použitá modrá fólia zvaná B-foil určená presne pre tieto účely. Na túto fóliu sa vytlačí požadovaný návrh prednej aj zadnej strany dosky plošného spoja pomocou laserovej tlačiarne. Následne je tento návrh prenesený na dosku plošného spoja pomocou žehličky, ktorá musí mať nastavenú optimálnu teplotu.



Obr. 4.4: Vytlačený návrh dosky plošného spoja na fólii B-foil

Po prenesení toneru na dosku bola doska vyleptaná pomocou chloridu železitého, čím sa odstránila všetka meď okrem tej, ktorá sa nachádza pod preneseným tonerom. Následne bol toner odstránený technickým benzínom, čím zostali iba vodivé cesty. Potom prebehla ešte kontrola týchto vodivých ciest či neprišlo k spojeniu alebo podleptaniu niektorých ciest. Vzhľadom k tomu, že niektoré cesty sú veľmi blízko seba a niektoré sú veľmi úzke bolo potrebných dosť korekcií. Nasledovalo už iba osadenie všetkých súčiastok pomocou mikrosťapkačky.



Obr. 4.5: Zadná strana dosky plošného spoja po vyleptaní a predná strana po osadení súčiastok

## Kapitola 5

# Konštrukcia robota

Po uvážení čo má robot vykonávať a aké všetky súčiastky bude robot obsahovať treba zvážiť, aká konštrukcia je pre robota vhodná.

Existujú 3 základné typy konštrukcií vhodné pre robota:

- Hotové konštrukcie
- Súprava častí (Kity)
- Vlastná konštrukcia

### 5.1 Hotové konštrukcie

Hotové konštrukcie sú vhodné najmä pre tých, ktorí sa nezaujímajú o tvorbu konštrukcie. Pri týchto konštrukciách sa nevyžaduje žiadna montáž. S hotovou konštrukciou je možné všetok čas stráviť pripojením senzorov a hľadaním lepších spôsobov programovania robota. Dobrým príkladom je konštrukcia *iRobot Create*. Vyzerá presne ako robotický vysávač *iRobot Roomba*, ale bez všetkých senzorov a programu na vysávanie [13].

Tieto typy konštrukcií už obsahujú mikrokontrolér a väčšinou aj programovacie prostredie od výrobcu. Z toho vyplýva, že väčšinou treba pracovať s vývojovou platformou, ktorú hotová konštrukcia obsahuje.



Obr. 5.1: Hotová konštrukcia iRobot Create 2<sup>1</sup>

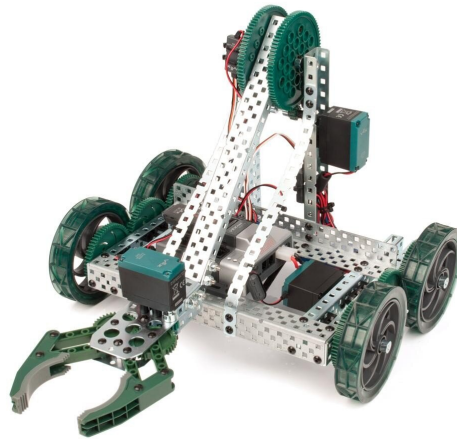
<sup>1</sup><https://store.irobot.com/default/create-programmable-programmable-robot-irobot-create-2/RC65099.html>

## 5.2 Súprava častí - Kit

Tento typ konštrukcií by sa dal prirovnať k hračkáckym stavebnciam LEGO, ktoré poskytujú hardvér pre tvorbu akejkoľvek budovy, mostu auta alebo inej mechanickej veci. Veľa spoločností, ktoré sa špecializujú na robotiku sa inšpirovali týmto nápadom a posunuli ho o krok ďalej. Ich konštrukčné kity môžu byť použité na vytvorenie plne funkčného robota alebo jednotlivé kúsky je možné si prispôbiť na vytvorenie vlastného dizajnu [6].

### 5.2.1 Robotický systém VEX

Tento systém sa sústreďuje na vzdelávací ale aj hobby trh. Je založený na systéme nosníkov a spojovacích kusov. Väčšina častí je pripevnená pomocou bežnej techniky strojných skrutiek a matíc. Čo odlišuje VEX od konkurencie je najmä to, že tieto kusy sú navrhnuté špeciálne pre malých robotov. Súpravy sa dodávajú s motormi vyrobenými tak, aby sa zmestili na nosníkové časti, ozubené kolesá, kolesá a mnoho ďalších častí hardvéru [13].



Obr. 5.2: VEX Clawbot Kit <sup>2</sup>

### 5.2.2 Súprava Lynxmotion Servo Erector

Sada servo erektorov od spoločnosti Lynxmotion sa skladá z rôznych konzol a iného hardvéru pre účely spájania servo motorov štandardnej veľkosti R/C. Tieto konzoly sa dajú pripojiť na tradičnú robotickú základňu, aby mohol byť zostrojený kolesový alebo kráčací stroj alebo je ich možné pripevniť do rúrok, rozbočovačov a konektorov, aby sa vytvorila podoba podľa vlastných predstáv [13].

### 5.2.3 Bioloid

Robotické stavebnice Bioloid sú navrhnuté pre pokročilú stavbu robotov. Použitie zmesi kovových a plastových častí umožňuje, aby sa kusy bioloidu mohli kombinovať tak, aby sa vytvorili kolesové a aj chodiace roboty, ako aj plne funkčné uchopovače a ramená. Súpravy obsahujú špeciálne digitálne motory s vysokým krútiacim momentom a centrálny mikrokontrolér na ovládanie všetkých elektronických komponentov [13].

<sup>2</sup><https://www.vexrobotics.com/276-2600.html>

## 5.3 Vlastná konštrukcia

Pri vlastnej konštrukcii je základom výber správneho materiálu. Robot môže byť skonštruovaný z dreva, plastu, kovu prípadne z tvrdej lepenky. Závisí to najmä od dizajnu robota - aký veľký bude a na čo sa bude používať. Môže to byť jednoduchý kus dreva alebo aj amalgám z kvalitného plastu.

Rozpočet, konštrukčné znalosti a dostupnosť náradia tiež ovplyvnia výber materiálu. Je oveľa jednoduchšie rezať prípadne vŕtať do plastu a lepenky ako do hliníku alebo ocele.

### 5.3.1 Drevo

Drevo je materiál s ktorým sa ľahko pracuje, je možné ho brúsiť a píliť do akéhokoľvek tvaru a nie je vodivé (zabraňuje skratom), pokiaľ nie je mokré. Drevo je ľahko dostupné dokonca za nízku cenu. Je veľmi odolné a silne odporúča sa pre väčšinu robotických základní. Nevýhody dreva sú, že nie je také pevné ako plast alebo kov a môžu na ňom vznikáť praskliny [13].



Obr. 5.3: Robotické rameno vyrobené z dreva <sup>3</sup>

### 5.3.2 Plast

Existuje vyše tisíc druhov plastov avšak pre konštrukciu robota je vhodných a dostupných iba pár typov. Najpoužívanejšie sú [13]:

- Akryl -používa sa primárne na dekoratívne účely ako sú napríklad rámy obrazov. Treba si dávať pozor na praskanie spôsobené príliš veľkou hmotnosťou.
- Polykarbonát vyzerá podobne ako akryl, ale je výrazne pevnejší. Tento plast je aj bežná náhrada za okenné sklo. kvôli jeho zvýšenej hustote je veľmi ťažké pracovať s ním a je drahší ako akryl.
- Špeciálny typ PVC zvaný "expanded PVC" je ideálne pre konštrukciu malých a stredne veľkých robotov. Je ľahšie s ním pracovať ako s akrylom a polykarbonátom.

<sup>3</sup><https://www.instructables.com/id/How-to-Use-a-Few-Pieces-of-Wood-to-Assemble-Into-a/>

### 5.3.3 Kov

Typický materiál pre konštrukciu robota je kov. Kov patrí medzi najdrahšie materiály pre robotov z hľadiska nákladov a váhy. S kovom je veľmi ťažké pracovať pokiaľ nie je použité vhodné náradie a zručnosti.

Pre robotov sú hliník a oceľ najpoužívanejšie kovy. Hliník je mäkkší kov, takže je ľahšie s ním pracovať. Oceľ je niekoľkokrát pevnejšia a ľahšie sa zvára ak sa plánuje stavba veľkého robota.

Roboti vyrobené z dreva, plastu alebo aj iných materiálov však môžu používať kov (typicky hliník) v ich konštrukcii. Jedná sa hlavne o kovové časti ako sú konzoly na držanie kusov, matice, skrutky a iné spojovacie prostriedky [13].



Obr. 5.4: Robot s ocelovou konštrukciou <sup>4</sup>

### 5.3.4 Ľahké materiály

Ľahké materiály sa používajú najmä pri technike zvanej rýchle prototypovanie. Tieto materiály môžu byť rezané bežnými ručnými nástrojmi ako nože alebo hobby píly. Dokonca sa dajú použiť klince pre vysekávanie dier. Vďaka rýchlemu prototypovaniu sa dá vyrobiť robot za kratší čas a menej peňazí. Je to dobrá cesta ako si otestovať dizajn pred postavením robota zo silnejších materiálov [13].

Typické ľahké materiály sú [13]:

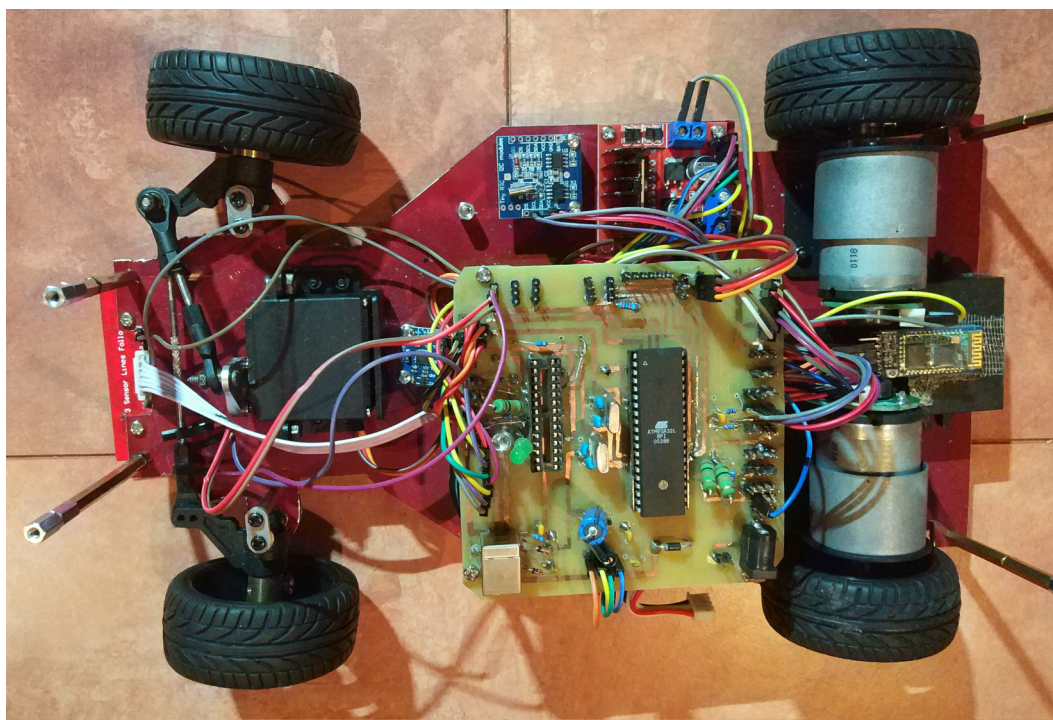
- Vysoko odolný kartón - je prekvapivo silný a ľahko sa dá rezať aj vrtáť.
- Vrstvené kompozitné materiály - patrí sem napríklad penová doska, ktorá je zložená z kusu plastovej peny, ktorá je vložená medzi 2 kusy ťažkého papiera. Iné typy vrstvených materiálov môžu používať kombinácie dreva, papiera, plastu a dokonca tenký kov.
- Vlnený plast je ako kartón, ale je vyrobený z platu. Je vhodný pre malých a ľahkých robotov. Tento materiál sa dá krájať nožom.

<sup>4</sup><http://www.gloriousmakeupartist.ca/Kids-Toys-tank-Robot-DIY-Chassis-Smart-track-with-two-carbon-brush-motor-for-Arduino-Stainless-steel-tanks-fourwheel-drive-climbing-u02kDy8t-p-2859.html>

## 5.4 Použitá konštrukcia pre robota

Na konštruovanie robota bola použitá vlastná konštrukcia. Jedná sa o materiál zvaný Alupanel, ktorý sa zaraďuje medzi vrstvené kompozitné materiály. Alupanel sa skladá z dvoch hliníkových plechov o šírke 0,3 mm a sendvičovej vrstve polyetylénuového jadra. Jeho výhody sú najmä kvalita pevnosť a tvrdosť podobná oceli a zároveň je tento materiál oveľa ľahší ako oceľ. Dokonca hliníkové vrstvy sú pokryté špeciálnou farbou, takže materiál je nevodivý a zabraňuje vzniku skratov. Nevýhodou tohto materiálu je najmä horšia dostupnosť a cena.

Na vyrezanie materiálu do potrebného tvaru bola použitá píla na železo a detaily boli potom dobrúsené pomocou pilníka na železo. Čiže s materiálom sa pracovalo ako s kovovým. Motory aj s kolesami boli uchytené pomocou skrutiek a matíc. Takisto pomocou skrutiek a matíc bol uchytený aj mechanizmus na zatáčanie predných kolies, vrátane servomotora. Keďže Bluetooth modul, stabilizátor napätia a batéria neobsahovali diery na uchytenie a pri vŕtaní do týchto súčiastok by mohlo dôjsť k ich poškodeniu, tak sú uchytené pomocou kvalitnej obojstrannej pásky. Ostatné súčiastky na spodnej časti konštrukcie (doska plošného spoja, senzory, H-mostík) sú uchytené pomocou dištančných stĺpikov a skrutiek.



Obr. 5.5: Spodná časť konštrukcie vyrobená z Alupanelu aj s osadenými súčiastkami

Vrchná a spodná časť konštrukcie je takisto prepojená pomocou dištančných stĺpikov a tiež aj displej a tlačidlá s vrchnou časťou konštrukcie. V strede vrchnej konštrukcie sa nachádza diera, cez ktorú vedú káblíky od týchto súčiastok do dosky plošného spoja. Na vrchnej časti konštrukcie sa ešte nachádzajú dva senzory na meranie vzdialenosti, ktoré sú uchytené pomocou malých plechov ohnutých do pravého uhla, aby senzor sledoval vzdialenosť od predmetu pred sebou, respektíve za sebou.



## Kapitola 6

# Aplikácia na ovládanie robota

Aplikácia na ovládanie robota je určená pre smartphoney s operačným systémom Android. Jej cieľom je najmä univerzálnosť, aby si ju mohol každý užívateľ nakonfigurovať podľa seba, na základe toho akého má robota. Aplikácia nebude určená len pre danú vývojovú platformu, ale pre každého, kto ovláda svojho robota cez Bluetooth.

Na tvorbu aplikácie bolo použité najpoužívanejšie rozhranie IDE a to Android Studio. K vytvoreniu spojenia medzi smartphoney a robotom je použitá voľne dostupná knižnica *Kotlin-Bluetooth*<sup>1</sup>, ktorá používa programovací jazyk Kotlin.

### 6.1 Existujúce riešenia

Existujúcich riešení na ovládanie robota pre operačný systém Android je veľké množstvo a dali by sa rozdeliť do 3 kategórií:

- Nemeniteľné prostredie a aj príkazy (vopred určené na daného robota)
- Nemeniteľné prostredie a konfigurovateľné príkazy
- Konfigurovateľné príkazy a aj prostredie

Ideálne riešenie v tomto prípade je aspoň 1 preddefinované prostredie a aj s preddefinovanými príkazmi pre tých, ktorí sa detailnejšie nechcú zaoberať aplikáciou a ostatné prostredia by boli plne konfigurovateľné. Najbližšie k tomuto riešeniu mala aplikácia *RoboRemo* avšak tá poskytuje iba konfiguráciu vlastných prostredí a dokonca v bezplatnej verzii maximálne 5 položiek na 1 rozhranie. Aplikácia je dosť rozsiahla a nie je určená len na ovládanie robota cez Bluetooth, a preto si ju správne nakonfigurovať pre svojho robota je dosť náročné, najmä pre začiatočníka. Dokonca niektoré veci je potrebné si nastudovať z technickej dokumentácie k aplikácii.

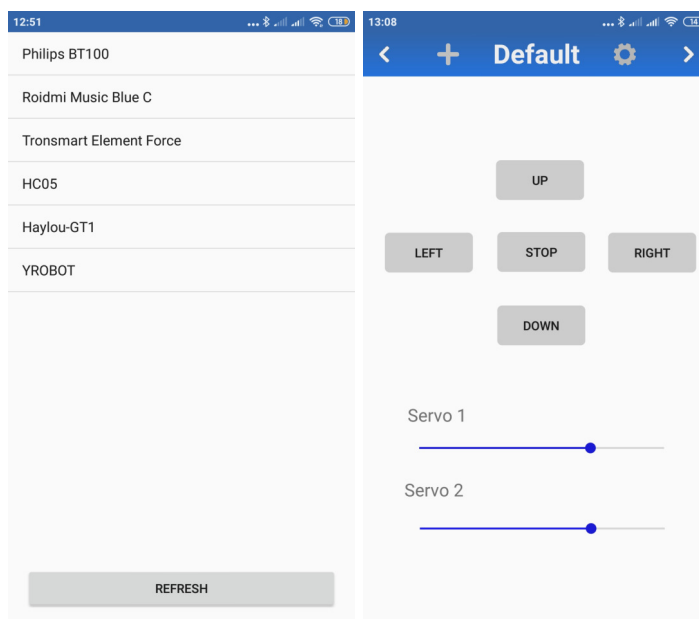
### 6.2 Grafické užívateľské prostredie

Grafické užívateľské prostredie je navrhnuté tak, aby bolo jednoduché a zároveň intuitívne. Takže obsahuje len prvky nevyhnutné na konfiguráciu prostredia a ovládanie robota.

Po spustení aplikácie sa zobrazí výzva o zapnutie Bluetooth na smartphoney a po zapnutí sa zobrazia všetky zariadenia spárované so smartphoney. Po úspešnom pripojení k robotovi

<sup>1</sup><https://github.com/appsinthesky/Kotlin-Bluetooth>

sa zobrazí preddefinované prostredie na ovládanie robota, ktoré si môže užívateľ upraviť, prípadne vytvoriť úplne nové. Preddefinované prostredie obsahuje 5 tlačidiel na ovládanie robota a 2 slidery na ovládanie servomotorov alebo prípadne rýchlosti motorov. Jednotlivé príkazy, ktoré budú odoslané do robota sú nakonfigurované tak, že sa zhodujú s názvom daného tlačidla, prípadne slidera. V prípade sliderov je zadaný aj rozsah a to od 0 po 100. Po kliknutí na tlačidlo, prípadne posunutie slidera sa v dolnej časti aplikácie zobrazí aký príkaz bol odoslaný do robota. Používateľ môže mať vytvorených aj viac prostredí a kedykoľvek medzi nimi prepínať. Na tvorbu grafického užívateľského prostredia bol použitý



Obr. 6.1: Pripojenie sa k robotovi a preddefinované rozhranie

jazyk XML.

### 6.3 Konfigurácia aplikácie

V hornej časti aplikácie sa nachádzajú šípky na zmenu rozhrania, prípadne vytvorenie nového, tlačidlo na pridanie prvkov do rozhrania(+) a tlačidlo na úpravu (ozubené koliesko). Po kliknutí na "+" sa zobrazí možnosť pridať do rozhrania nové tlačidlo, slider alebo text . Po kliknutí na tlačidlo pre úpravu, sa zobrazí rozhranie v režime úpravy a vtedy je možné meniť polohu prvkov kliknutím na daný prvok a jeho posúvaním. V režime úprav po kliknutí na prvok rozhrania bude možné nakonfigurovať daný prvok. Pri tlačidle to bude:

- Názov tlačidla
- Príkaz, ktorý sa pošle do robota po jeho stlačení
- Šírka a výška
- Zmazať tlačidlo

Pri slideri:

- Minimum - číslo

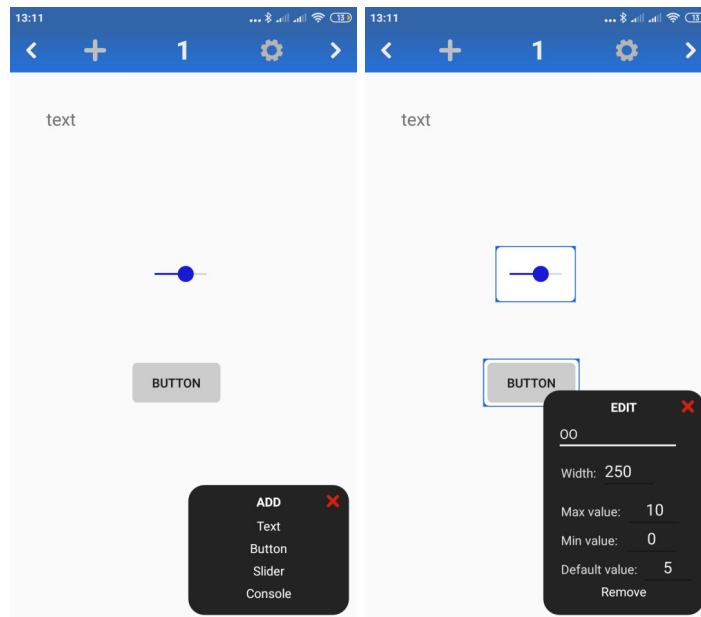
- Maximum - číslo
- ID tlačidla, ktoré bude odoslané do robota s aktuálnou hodnotou slidera
- Šírka
- Základná hodnota
- Odstránenie slidera

Pri texte:

- Veľkosť textu
- Šírka a výška
- Odstránenie textu

Väčšina Bluetooth modulov prijíma jednotlivé príkazy po znakoch. Aby bolo možné vytvárať aj viac-znakové príkazy, tak je vždy na koniec pridaný znak nového riadku, ktorý slúži na oddelenie jednotlivých príkazov od seba.

Pri slideroch sa posiela a aktuálne číslo na slideri. Takže ak je príkaz „OO“ a na slideri aktuálne číslo 125, tak sa pošle príkaz „OO125\n“.



Obr. 6.2: Úprava rozhrania a konfigurácia slidera

# Kapitola 7

## Testovanie

Cieľom testovania bolo odhaliť funkčnosť hardvérových prvkov, nájsť chyby v programe a odhaliť odchýlku od skutočných hodnôt. Všetky testy prebehli v domácich podmienkach. Testovanie bolo rozdelené na 3 kategórie a to:

- Testovanie podprogramov
- Testovanie senzorov
- Testovanie ostatných hardvérových prvkov

Videa z testov sa nachádzajú na pamäťovom nosiči v zložke **Videos**.

### 7.1 Testovanie senzorov

Na robotovi sa nachádzajú 4 senzory na otestovanie infračervený dialkomer, ultrazvukový dialkomer, senzory na sledovanie čiary a senzor počítanie otáčok motora.

Dialkomery boli otestované tak, že sa pred nich postavil predmet a najskôr sa odmerala vzdialenosť pomocou dialkomeru, ktorá sa zobrazuje na displeji a potom bola vzdialenosť od predmetu premeraná pomocou bežného metra. Pri infračervenom dialkomery, ktorý meria vzdialenosť od 4cm do 30cm boli merania veľmi podobné a líšili sa maximálne do 5 milimetrov. Pri ultrazvukovom dialkomery, ktorý meria vzdialenosť od 4cm do 400cm sa odchýlka mierne zvyšovala podľa vzdialenosti od predmetu. Pri vzdialenosti do 80cm bola odchýlka maximálne 5 milimetrov, avšak pri vyšších vzdialenostiach sa zobrazovalo na displeji viacero čísel medzi ktorými bol rozdiel až do 3 centimetrov. Tieto čísla boli potom spriemerované a vyšli prekvapivo presné výsledky aj pri väčších vzdialenostiach.

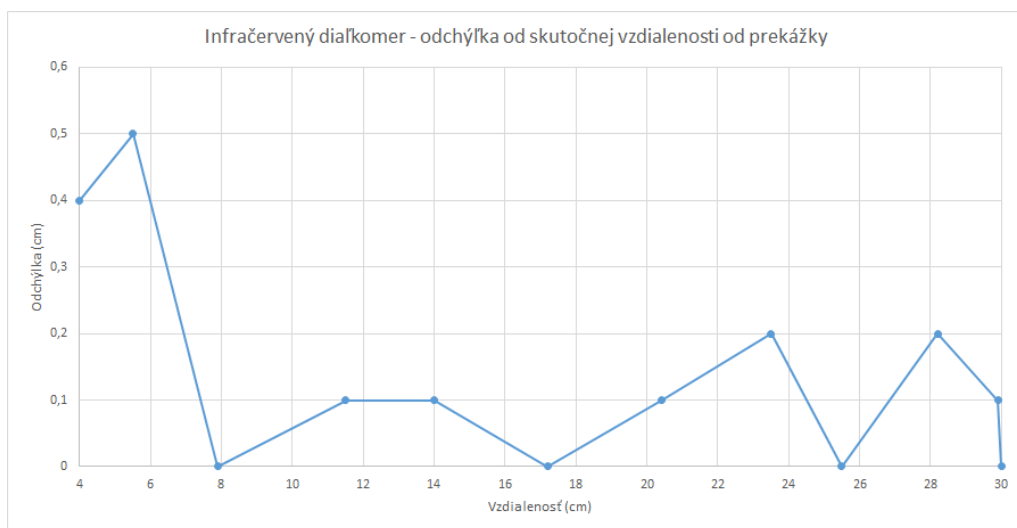
U senzoroch na sledovanie čiary bola testovaná najmä správna funkčnosť. Testovaná bola veľkosť napätia, keď sa pod senzorom nachádza čierny a biely podklad. Testy prebehli pomocou A/D prevodníkov a čísla boli zobrazované na displeji. Na čiernom podklade boli tieto čísla v rozmedzí od 950 do 1000 (4,6V - 4,9V). Pri bielom podklade sú to čísla od 600 do 900 (3,9V - 4,4V). Pre biely podklad sú tieto čísla dosť vysoké a mali by sa blížiť k nule. Preto boli tieto senzory premerané aj digitálnym multimetrom a čísla sa zhodovali. To naznačuje že použité senzory sú dosť nekvalitné, avšak na rozlíšenie bielej a čiernej to bude postačovať. Pri iných farbách by to však mohol byť problém. Podrobnejšie budú tieto senzory otestované pri programe na sledovanie čiary.

Senzor na počítanie otáčok motora, ktorý slúži na zistenie prejdenej vzdialenosti robotom bol testovaný podobným princípom ako dialkomery. Prejdená vzdialenosť z enkodéra je zobrazovaná na displeji a následné porovnaná so vzdialenosťou nameranou metrom.

Boli merané vzdialenosti do 5m, aby bolo možné túto vzdialenosť zmerať bežným metrom. Vzdialenosti sa líšili maximálne do 10 centimetrov.



Obr. 7.1: Graf určujúci odchýlku medzi vzdialenosťou nameranou ultrazvukovým diaľkomerom a metrom



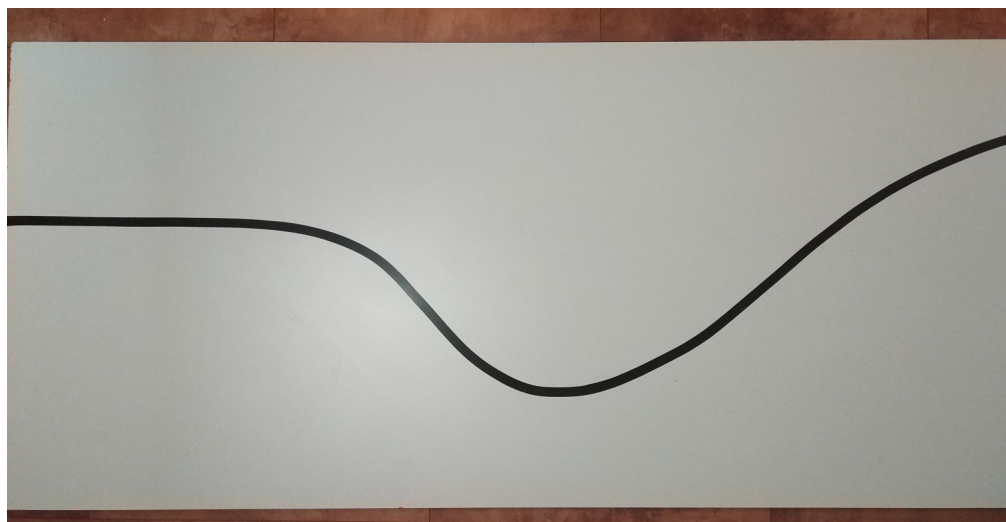
Obr. 7.2: Graf určujúci odchýlku medzi vzdialenosťou nameranou infračerveným diaľkomerom a metrom

## 7.2 Testovanie podprogramov robota

Robot obsahuje 2 podprogramy a to program na sledovanie čiary a program na bezdrôtové ovládanie robota cez Bluetooth.

Na otestovanie programu na sledovanie čiary bola použitá biela doska o rozmeroch 2m x 1m. Sledovaná čiara bola vytvorená pomocou čiernej izolačnej pásky so šírkou 1

cm. Počas testovania bol zistený jeden problém. Motory sa točia príliš rýchlo na to aby predný servomotor stihol včas zatočiť. Preto bola rýchlosť znížená pri zatáčaní kolies. Okrem samotného sledovania čiary má tento podprogram za úlohu detekovať predmet a zabrániť kolízii. Pri testovaní bolo zistené, že tento predmet musí mať minimálne 12cm výšku a 8cm šírku, aby ho diaľkomer zaznamenal a včas zastavil robota.



Obr. 7.3: Testovacia dráha na sledovanie čiary

Testovaním ovládania robota sa zároveň otestovala aj aplikácia na ovládanie robota. Testované boli príkazy odosielané do robota na pohyb motorov, rýchlosť motorov a servomotorov. Príkazy robot zvládol spracovávať bez problémov, avšak bolo potrebné nastaviť krajné polohy servomotorov, pretože pri odoslaní príliš vysokého čísla na otočenie došlo k poškodeniu a servomotor musel byť vymenený. Ďalej bola testovaná ovládateľnosť robota. K tomu poslúžili 3 respondenti, z ktorých 2 mali skúsenosti s ovládaním robota a 1 nie. Respondenti dostali 2 úlohy. Prvou úlohou bolo prejsť určenú dráhu. S touto úlohou nemal nikto problém. Druhou úlohou bolo presunúť predmet z bodu A do bodu B. Táto úloha bola pre každého náročná a dokonca respondent, ktorý nemal skúsenosti s ovládaním robota ju ani nezvládol. Všetci sa zhodli na tom, že ovládanie celého ramena pomocou sliderov je dosť náročné a vhodnejší by bol samostatný ovládač, čo je dobrý nápad na budúce rozšírenie robota.

### 7.3 Testovanie ostatných hardvérových prvkov

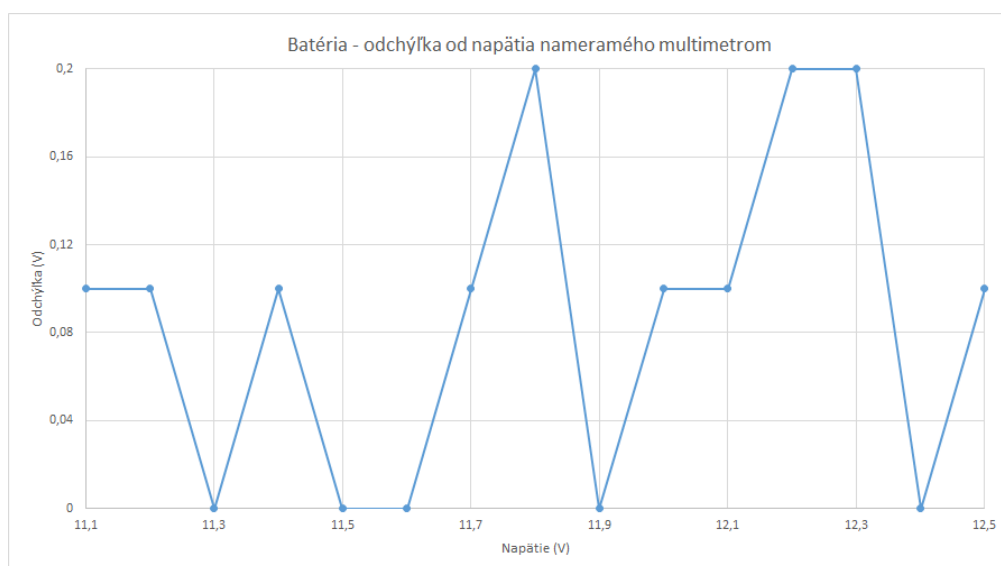
Medzi ostatné hardvérové prvky patrí obvod reálneho času, tlačidlá a batéria.

Na testovanie obvodu reálneho času bol potrebný dlhší interval a to 4 týždne. Na začiatku bol zapísaný do tohto obvodu aktuálny čas a po 4 týždňoch prebehla kontrola a porovnanie s aktuálnym časom. Čas zobrazený na displeji z obvodu reálneho času sa s aktuálnym časom líšil o 2 sekundy. Otestovať správnu funkčnosť sa podarilo až na druhýkrát, pretože počas prvého testovania po prvom týždni prebehol skrat, čím sa čas úplne zasekol a bolo potrebné ho znova nastaviť. Príčina skratu bola následne tiež odstránená.

Pri tlačidlách bola testovaná ich funkčnosť a spoľahlivosť. Test spočíval v tom, že každé tlačidlo bolo stlačené 40krát a na ich stlačenie mal zareagovať displej vypísaním, ktoré tlačidlo bolo stlačené. Najskôr bol problém s 1 tlačidlom, pretože dištančný stĺpik tlačidiel

bol spojený s medenou cestou, ktorá vedie z tlačidla do mikrokontroléra a ten reagoval na neho ako uzemnenie, čiže logickú 0. To znamená, že mikrokontrolér reagoval na to ako keby bolo toto tlačidlo stlačené. Medzi stĺpik a cestičku bol vložený nevodivý materiál, čo celý problém vyriešilo. Potom už testovanie prebehlo bez problémov a každé tlačidlo zareagovalo správne na 40 stlačení zo 40.

Pri báterii bola testované či zvládne maximálnu možnú záťaž, čas do vybitia pri bežnom používaní a napätie zobrazované na displeji. Zvládnutie maximálnej možnej záťaže bolo otestované tak, že robot musel udržať predmet zároveň bol zapnutý displej a motory bežali na plnej rýchlosti. Pôvodná batéria (7,4V 2700mAh) musela byť vymenená, pretože pri motoroch na plnej rýchlosti dochádzalo k reštartu mikrokontroléra spôsobené nízkym napätím na jeho vstupe. Preto bola použitá silnejšia batéria (11,1V 2700mAh) a tá už zvládla aj plnú záťaž. Použitá batéria je typu lítium polymér, ktoré sú známe tým, že zvládnu vysoký prúd, čiže maximálna záťaž by mala závisieť od použitého stabilizátora (5V/10A). Pri používaní robota čas do vybitia najviac ovplyvňovali používanie ramena a motory. Pri ich výraznom používaní batéria v priemere vydržala približne 30 minút avšak pri ich výraznom obmedzení sa tento čas predĺžil až na vyše 3 hodiny. Napätie na batérii zmerané pomocou A/D prevodníka a zobrazované na displeji, bolo porovnané s napätím nameraným pomocou digitálneho multimetra. Boli realizované 2 merania a to pri plnom nabití batérie a tesne pred vybitím. Napätia sa líšili v rozmedzí 0,1V.



Obr. 7.4: Napätie batérie - rozdiel medzi hodnotou nameranou multimetrom a A/D prevodníkom mikrokontroléra

## Kapitola 8

# Záver

Cieľom práce bolo vytvoriť vlastnú vývojovú platformu pre tvorbu robota, vhodnú pre začiatočníkov. Potom na tejto platforme vytvoriť vlastného robota a nakoniec aj aplikáciu na ovládanie tohto robota, čo bolo aj splnené.

V teoretickej časti sú popísané všetky hardvérové prvky, ktoré sú použité na robotovi a následne v ďalšej kapitole sú popísané funkčné bloky a registre mikrokotroléra, ktoré sú potrebné pre prácu s nimi. K mikrokontréru Atmega32 bola vytvorená vývojová platforma, aby mohli byť všetky hardvérové prvky pripojené a riadené pomocou mikrokotroléra. Vývojová platforma je navrhnutá, tak aby ju mohol každý použiť pre tvorbu svojho robota. Podobne univerzálne je aj navrhnutá aplikácia na ovládanie robota, ktorú si každý môže nakonfigurovať podľa vlastných predstáv. Vlastná konštrukcia robota je vyrobená z kvalitného materiálu, aby čo najmenej zaťažovala robota, ale zároveň bola dostatočne pevná. Nakoniec bol robot riadne otestovaný, kde sa našli hlavne menšie ale aj zásadne chyby, ktoré boli nakoniec odstránené.

Za veľmi náročnú časť práce považujem výrobu dosky plošného spoja. Keďže sa jedná o obojstranný návrh a niektoré cestičky sú užšie ako 1 milimeter, vzniká pri domácej výrobe veľké množstvo problémov ako sú skraty a neskôr aj studené spoje. Odstránenie všetkých týchto problémov bolo veľmi zdĺhavé a náročné, dokonca z času na čas sa objavil nový problém s doskou. I keď som sa pri odhaľovaní týchto problémov veľa nového naučil, nabudúce by som si dal takto náročnú dosku radšej vyrobiť špecializovanej firme.

V práci plánujem pokračovať tak, že vývojovú platformu predstavím na nejakej škole, prípadne pár ľuďom, ktorý majú záujem vytvoriť si vlastného robota. Robot poslúži ako veľmi dobrá ukážka toho, čo môžu dosiahnuť. Ak by práca zaujala týchto ľudí mohlo by byť vyrobených aj viac vývojových platforiem, prípadne neskôr by mohla vzniknúť aj nová verzia na základe poznatkov od týchto ľudí.



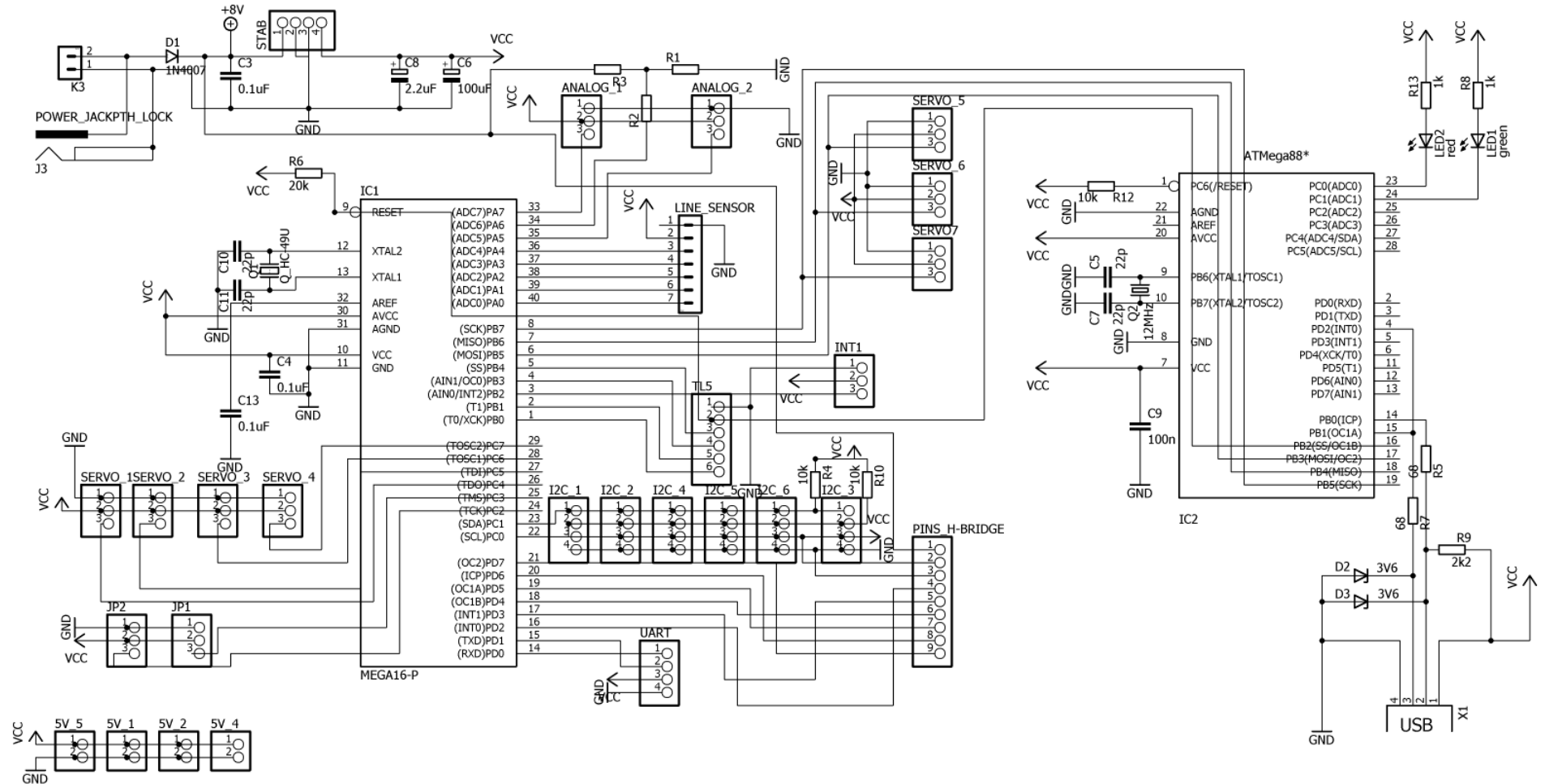
# Literatúra

- [1] *Servomotor* [online]. Wikipedia, september 2005 [cit. 2019-11-01]. Dostupné z: <https://cs.wikipedia.org/wiki/Servomotor>.
- [2] *H bridge* [online]. Wikipedia, november 2006 [cit. 2019-10-27]. Dostupné z: [https://en.wikipedia.org/wiki/H\\_bridge](https://en.wikipedia.org/wiki/H_bridge).
- [3] *Programming With AVR Microcontroller* [online]. 2014 [cit. 2019-12-14]. Dostupné z: <https://researchdesignlab.com/projects/AVR%20BOOK.pdf>.
- [4] BARRETT, S. F. a PACK, D. J. *Atmel AVR Microcontroller Primer: Programming and Interfacing*. Morgan & Claypool, 2008. ISBN 159829542X.
- [5] BRANDYS, F. *Jednosmerný motor a jeho riadenie* [online]. Posterus, december 2009 [cit. 2019-11-10]. Dostupné z: <https://www.posterus.sk/?p=3784>.
- [6] COOK, D. *Intermediate Robot Building*. 2. vyd. Apress, 2010. ISBN 1-4602-2755-7.
- [7] DEJAN. *Ultrasonic Sensor HC-SR04 and Arduino Tutorial* [online]. [cit. 2019-11-01]. Dostupné z: <https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/>.
- [8] JANSSEN, D. *Real-Time Clock (RTC)* [online]. [cit. 2020-05-05]. Dostupné z: <https://www.techopedia.com/definition/2273/real-time-clock-rtc/>.
- [9] KANDA. *AVR Microcontroller Coding* [online]. Apríl 2012 [cit. 2019-11-10]. Dostupné z: <https://www.kanda.com/blog/microcontrollers/avr-microcontrollers/avr-microcontroller-coding/>.
- [10] KATE, T., POONAM, S., SEHGAL, P. a JASUJA, N. *Bluetooth vs Wi-Fi* [online]. Diffen LLC [cit. 2020-05-03]. Dostupné z: [https://www.diffen.com/difference/Bluetooth\\_vs\\_Wifi](https://www.diffen.com/difference/Bluetooth_vs_Wifi).
- [11] MATOUŠEK, D. *Práce s mikrokontroléry ATMEL AVR ATmega16*. 1. vyd. BEN, 2006. ISBN 80-7300-174-8.
- [12] MAZIDI, M. A., NAIMI, S. a NAIMI, S. *AVR Microcontroller and Embedded Systems*. Pearson Education, Inc, 2011. ISBN 0-13-800331-9.
- [13] MCCOMB, G. *Robot Builder's Bonanza*. 4. vyd. The McGraw-Hill Companies, 2011. ISBN 0-07-175035-6.
- [14] OASHI. *Modelářské servo* [online]. Wikipedia, november 2010 [cit. 2019-10-27]. Dostupné z: <https://cs.wikipedia.org/wiki/Model>.

- [15] PAO, C. *What is an IMU Sensor?* [online]. [cit. 2020-05-05]. Dostupné z:  
<https://www.ceva-dsp.com/ourblog/what-is-an-imu-sensor/>.
- [16] PARDUE, J. *C Programming for Microcontrollers AVR*. Smiley Micros, 2005. ISBN 0-9766822-0-6.
- [17] POKORNÝ, M. *Úvod do mikrokontrolů ATMEL AVR Konkrétně klonů řady ATmega* [online]. 2004 [cit. 2019-12-15]. Dostupné z:  
<http://noel.feld.cvut.cz/vyu/scs/prezentace2005/ATmega/ATmega.pdf>.
- [18] VALEČKA. *Mikrokontrolér ATMEL ATmega 32* [online]. [cit. 2020-01-17]. Dostupné z:  
[https://www.souepl.cz/wp-content/ucitele/valecka/at\\_mega\\_32\\_popis%20%280praveno%29.htm](https://www.souepl.cz/wp-content/ucitele/valecka/at_mega_32_popis%20%280praveno%29.htm).
- [19] VOJÁČEK, A. *Rotační enkodéry pro řízení pohonů a pohybu* [online]. HW server s.r.o., apríl 2017 [cit. 2019-10-29]. Dostupné z:  
<https://automatizace.hw.cz/rotacni-enkodery-pro-rizeni-pohonu-a-pohybu.html>.

**Príloha A**

**Schéma zapojenia**



Obr. A.1: Schéma zapojenia vývojovej platformy