



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

MALÝ ROBOT PRO SLEDOVÁNÍ ČÁRY

SMALL PATH FOLLOWER ROBOT

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

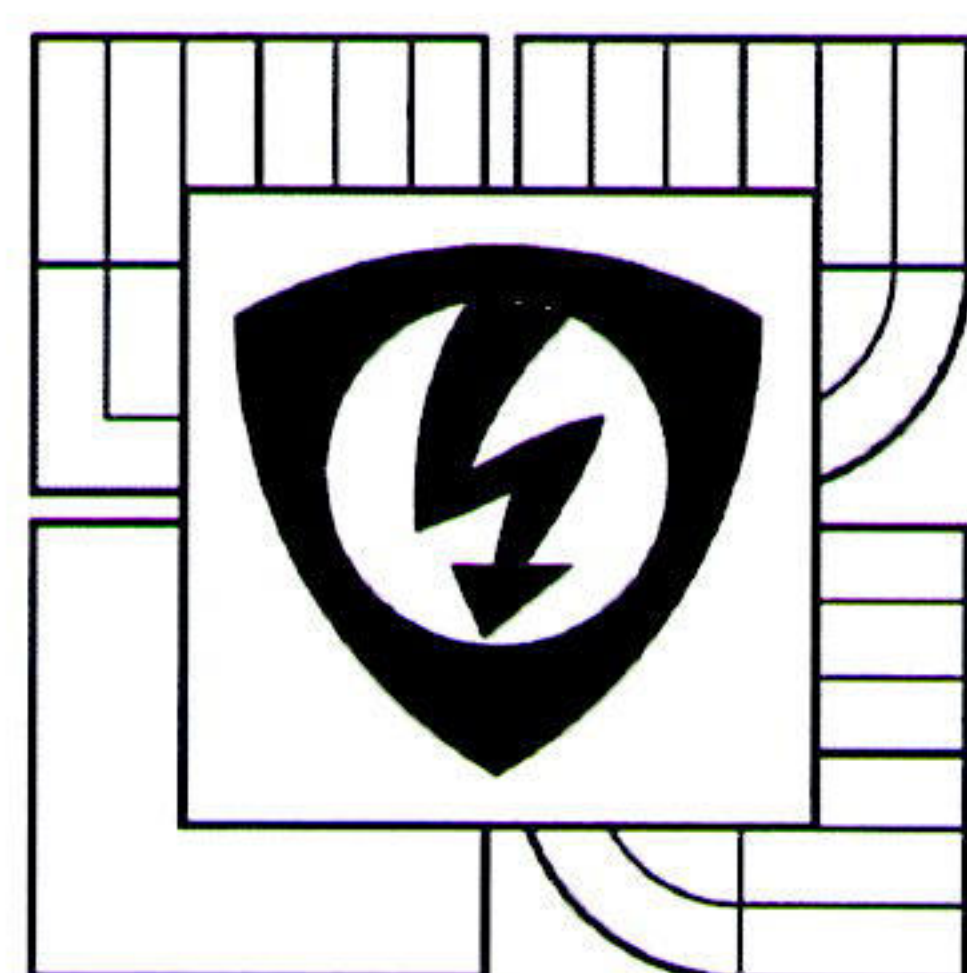
MICHAL ZACPAL

VEDOUcí PRÁCE

SUPERVISOR

Ing. LUKÁŠ KOPEČNÝ, Ph.D.

BRNO 2013



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student: Michal Zaccpal

Ročník: 3

ID: 134663

Akademický rok: 2012/13

NÁZEV TÉMATU:

Malý robot pro sledování čáry

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a sestavte řídicí systém malého robotu pro sledování čáry. S použitím modulů vzniklých v předchozích projektech zkompletujte a otestujte celý robot.

DOPORUČENÁ LITERATURA:

Magazín světa robotiky [online]. [cit. 2012-09-14]. <<http://robotrevue.cz>>.

Robotika.cz [online]. [cit. 2012-09-14]. <<http://robotika.cz>>.

Termín zadání: 11.2.2013

Termín odevzdání: 27.5.2013

Vedoucí práce: Ing. Lukáš Kopečný, Ph.D.

Konzultanti bakalářské práce:

doc. Ing. Václav Jirsík, CSc.

předseda oborové rady



UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Abstrakt

Tato práce se zabývá návrhem a následnou realizací řídicího systému malého robota pro sledování čáry. Na začátku jsou představeny platformy, na kterých je systém postaven. V další části jsou postupně představeny: modul pro sledování čáry (HW,SW), modul pro řízení motorů (HW,SW) a řídicí jednotka (HW,SW). Dále je představena konstrukce, na které byl systém testován. Nakonec byly implementovány dva algoritmy pro sledování čáry. Celý systém včetně zásuvných modulů je postaven na mikrokontrolérech firmy Atmel, a to ATmega8 a ATmega328. Ty jsou založeny na open - source platformě Arduino tím, že obsahují Arduino bootloader. Mikrokontroléry využívají ke komunikaci mezi sebou I2C sběrnici. Ke komunikaci s uživatelem využívá řídicí jednotka LCD displej a čtyři tlačítka. Pro snímání čáry je použito pět reflexních optočlenů CNY70.

Klíčová slova

Robot, I2C, Arduino, ATmega8, ATmega328, mikrokontrolér, H-můstek, sledování čáry, reflexní optočlen, LCD displej.

Abstract

This thesis describes the design and subsequent implementation of the control system for small path follower robot. On the introduction are presents the platform's on which the system was built. In the next section are gradually introduced: A module for following the line (HW, SW), modul for control motors (HW, SW) and the control unit (HW, SW). Then is presents construction on which the system was tested. Finally, there were implemented two algorithms for tracking lines. Whole system, including the plug moduls is built on Atmel microcontrollers, and this ATmega8 and ATmega328. They are based on open - source platform Arduino, because they contain Arduino bootloader. Microcontrollers using I2C bus to communicate each other. Control unit is using to communicate with users LCD display and four buttons. For scanning a line is using five reflective optocoupler CNY70.

Keywords

Robot, I2C, Arduino, ATmega8, ATmega328, microcontroller, H-bridge, line following, reflective optocoupler, LCD display.

Bibliografická citace:

ZACPAL, M. *Malý robot pro sledování čáry*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 57s. Vedoucí bakalářské práce byl Ing. Lukáš Kopečný, Ph.D..

Prohlášení

„Prohlašuji, že svou bakalářskou práci na téma Malý robot pro sledování čáry jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **24. května 2013**

.....
podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Lukáši Kopečnému, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne: **24. května 2013**

.....
podpis autora

Obsah

1	Úvod.....	11
1.1	Cíle práce.....	11
2	Koncepce práce, použité platformy	12
2.1	Uspořádání konektoru určeného pro připojení k řídicí jednotce	12
2.2	Použité platformy.....	13
2.2.1	Mikrokontrolér, mikroprocesor	13
2.2.2	Arduino.....	13
2.2.3	Komunikační sběrnice I2C	15
3	Modul pro sledování čáry (Line follow modul)	18
3.1	Popis modulu	18
3.2	Elektronika.....	19
3.2.1	Řídicí mikrokontrolér	19
3.2.2	Princip snímání čáry	20
3.3	Řídicí algoritmus	22
3.3.1	Komunikace s modulem.....	23
4	Modul pro řízení motorů (Driver modul)	25
4.1	Popis modulu	25
4.2	Elektronika.....	26
4.2.1	Řídicí mikrokontrolér	26
4.2.2	Ovládání motorů.....	27
4.3	Řídicí algoritmus	33
5	Řídicí jednotka (Robot controll unit)	36
5.1	Elektronika.....	36
5.1.1	Napájení řídicí jednotky	39
5.2	Softwarové vybavení jednotky	39
5.2.1	Komunikace s moduly.....	39
5.2.2	Obsluhující algoritmus řídicí jednotky	41
5.2.3	Algoritmus sledování čáry.....	42
6	Konstrukce robota.....	47
7	Závěr.....	49
	Literatura.....	51
	Seznam zkratk	52
	Seznam příloh.....	53

Seznam obrázků

Obr. 2.1. Uspořádání konektorů na desce zásuvného modulu pro připojení k řídicí jednotce (pohled shora)	12
Obr. 2.2. Ukázka vývojového prostředí Arduino IDE a deska Arduino UNO.....	14
Obr. 2.3. Schéma připojení zařízení na sběrnici I2C	16
Obr. 3.1. Blokové schéma modulu pro sledování čáry	18
Obr. 3.2. Uspořádání konektorů modulu pro sledování čáry	19
Obr. 3.3. Reflexní optočlen CNY70 (převzato z [7]).....	20
Obr. 3.4. Princip funkce reflexního optočlenu CNY70.....	21
Obr. 3.5. Zapojení snímače čáry (reflexního optočlenu CNY70)	21
Obr. 3.6. Uspořádání reflexních optočlenů CNY70 na desce pro snímání čáry (Pohled shora – snímače pod deskou).....	22
Obr. 3.7. Osazená řídicí deska modulu pro sledování čáry.....	22
Obr. 3.8. Vývojový diagram znázorňující algoritmus modulu pro sledování čáry	23
Obr. 3.9. Vývojový diagram znázorňující průběh funkce <i>receive_event()</i> ;	23
Obr. 3.10. Vývojový diagram znázorňující průběh funkce <i>request_event()</i> ;.....	24
Obr. 4.1. Blokové schéma modulu pro řízení motorů.....	25
Obr. 4.2. Uspořádání konektorů modulu pro řízení motorů.....	26
Obr. 4.3. Kvadranty řízení motoru.....	27
Obr. 4.4. Budič pracující v kvadrantu Q1	28
Obr. 4.5. Budič pracující v kvadrantu Q2	28
Obr. 4.6. Budič pracující v kvadrantech Q1 a Q4.....	29
Obr. 4.7. Budič pracující v kvadrantech Q1 a Q2	29
Obr. 4.8. Čtyřkvadrantový budič.....	30
Obr. 4.9. Princip pulzně šířkové modulace (PWM).....	31
Obr. 4.10. Osazený modul pro řízení motorů.....	33
Obr. 4.11. Vývojový diagram znázorňující průběh funkce <i>receive_event()</i> ;.....	34
Obr. 4.12. Vývojový diagram znázorňující algoritmus modulu pro řízení motorů.....	35
Obr. 5.1. Konstrukce řídicí jednotky s popisem součástí.....	37
Obr. 5.2. Blokové schéma řídicí jednotky.....	38
Obr. 5.3. DC-DC měnič TRACO TSR 1-2450 umístěný na robotovi	39
Obr. 5.4. Koeficienty přiřazené jednotlivým čidlům snímače čáry.....	42
Obr. 5.5. Vývojový diagram funkce pro sledování čáru	43

Obr. 5.6. Vývojový diagram funkce scan_I2Cbus	44
Obr. 5.7. Vývojový diagram hlavní smyčky	45
Obr. 5.8. Vývojový diagram funkce show_menu(), která slouží k výpisu menu na displej	46
Obr. 5.9. Vývojový diagram funkce show_menu2(), která provádí aktivaci funkcí řídicí jednotky.....	46
Obr. 6.1. Detail podpůrné nožky	47
Obr. 6.2. Detail držáku snímače čáry	48
Obr. 6.3. Kostra robota se snímačem čáry - celkový pohled	48

Seznam tabulek

Tab. 2.1. Přehledný výpis funkcí knihovny Wire.h s informací kde lze použít	17
Tab. 3.1. Popis zapojení pinů mikrokontroléru modulu pro sledování čáry	20
Tab. 4.1. Popis zapojení pinů mikrokontroléru modulu pro řízení motorů.....	27
Tab. 4.2. Tabulka znázorňující ovládání H – můstku – převzato z [1]	31
Tab. 5.1. Popis zapojení pinů mikrokontroléru řídicí jednotky	38

1 ÚVOD

Sledování čáry se řadí mezi jednu ze základních disciplín na většině robotických soutěžích. Úkolem robota sledujícího čáru je v co nejkratším čase projet dráhu tvořenou černou čarou na bílém podkladu. Dráha může obsahovat překážky, kterým se robot musí vyhnout. To vše musí provést autonomně bez jakéhokoliv zásahu uživatele. Využití sledování čáry nemá uplatnění pouze v robotických soutěžích, může se uplatnit v různých odvětvích, kde je potřeba sledovat dráhu v podobě černé čáry, magnetické pásky či jiného média.

Práce je rozdělena do několika částí. Nejprve jsou představeny a popsány prvky a platformy použité při konstrukci. Mezi ty hlavní patří mikrokontroléry firmy Atmel pracující na platformě Arduino a důležitá část celého projektu I2C sběrnice, na které závisí veškerá vnitřní komunikace robota. Poté následuje popis dvou modulů, a to modulu pro sledování čáry a modulu pro řízení motorů vytvořeného v semestrálním projektu [1]. Vždy je nejprve popsána elektrická stránka a poté softwarová, kde jsou popsány implementované algoritmy a způsob komunikace s modulem. Následně je představena řídicí jednotka a nakonec konstrukce robota, na které byl celý systém provozován.

1.1 Cíle práce

Cílem této práce je návrh a konstrukce řídicího systému pro malého robota sledujícího čáru. Tento systém bude tvořen jednotkou řízenou mikrokontrolérem schopnou komunikace s uživatelem prostřednictvím LCD displeje a tlačítka nebo za pomoci aplikace v PC přes sériové rozhraní. Jednotka bude s potřebnými moduly komunikovat přes vnitřní sběrnici I2C. Tyto moduly budou vytvořeny jako zásuvné moduly a k jednotce připojitelné přes předem dané uspořádání konektorů. Jednotku tak bude možné rozšířit o dodatečné funkce, které jsou v tomto případě představovány sledováním čáry a řízením motorů. Nakonec bude jednotka doplněna o algoritmus sledování čáry.

2 KONCEPCE PRÁCE, POUŽITÉ PLATFORMY

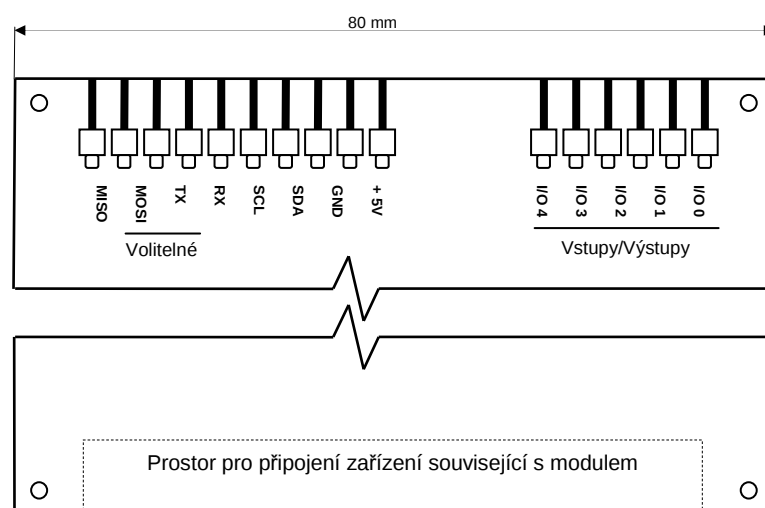
Vzhledem k tomu, že dále v textu se budou objevovat často se opakující pojmy, ať už více či méně důležité k pochopení práce (záleží na znalostech čtenáře), vytvořil jsem tuto kapitolu k jejich souhrnu a popisu. Budu se na ní dále odkazovat.

2.1 Uspořádání konektoru určeného pro připojení k řídicí jednotce

Zásuvné rozšiřující moduly jsou k řídicí jednotce připojovány přes uspořádání konektoru zobrazeném na Obr.2.1. Při pohledu shora na modul slouží pravá část k propojení vstupů/výstupů s deskou řídicí jednotky, která sdružuje vstupy/výstupy ze všech modulů a řídicí jednotky do jednoho konektoru. Levá část slouží k připojení napájecího napětí, připojení na I2C sběrnici pomocí vodičů SDA,SCL, sériové rozhraní (RX/TX), dále může mít modul připojeny piny pro ISP rozhraní (MISO,MOSI,SCK). To ovšem v jednotce není využito, a je určeno čistě jen k programování samostatného modulu (primárně jsou k programování využity piny RX,TX a RESET) a nakonec reset sloužící k resetování mikrokontroléru.

Mechanicky je konektor tvořen řadou zahnutých pozlacených kolíků viz. foto některého z modulů.

Řídicí programy zásuvných modulů byly nejprve odladěny mimo řídicí jednotku a poté byla zprovozněna komunikace s jednotkou.



Obr. 2.1. Uspořádání konektorů na desce zásuvného modulu pro připojení k řídicí jednotce (pohled shora)

2.2 Použité platformy

K vývoji a konstrukci robota byly využity tyto programy: Veškeré návrhy elektronických schémat a desek plošných spojů byly provedeny v návrhovém programu Eagle 5.7.0. Ke tvorbě zdrojových kódů a programování bylo využito Microsoft Visual Studio 2010 s Arduino pluginem (doplňkovým modulem). Odstavce níže stručně popisují použité mikrokontroléry a jejich souvislost s použitou platformou Arduino dále sběrnici I2C použitou ke komunikaci a rovněž její souvislost s Arduinem.

2.2.1 Mikrokontrolér, mikroprocesor

Mikroprocesor je složitý sekvenční obvod vykonávající sled aritmetických a logických operací uložených v paměti. Mikrokontrolér, také označovaný jako jednočipový mikropočítač, je elektronická součástka v podobě integrovaného obvodu, který obsahuje kompletní mikropočítač na jednom čipu. To znamená mikroprocesor společně s různými perifériemi, jako paměť programu, operační paměť, I/O (vstupně/výstupní) vývody, A/D převodníky (převádějí analogový signál na digitální) a další rozhraní. Díky své spolehlivosti a kompaktnosti jsou nejčastěji využívány v jednoúčelových aplikacích a často jsou součástí vestavěných (embedded) systémů.

2.2.1.1 Použité mikrokontroléry

V této práci jsou použity dva typy mikrokontroléru od firmy Atmel a to ATmega8 a ATmega328. Druhý jmenovaný vychází z ATmega8 a je její novější verzí doplňující ji o jistá vylepšení. Tyto mikrokontroléry jsou založeny na harvardské architektuře, to znamená, že mají oddělenou paměť pro program a data.

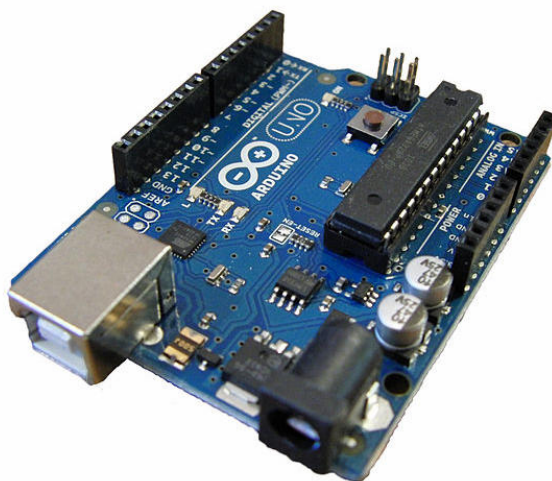
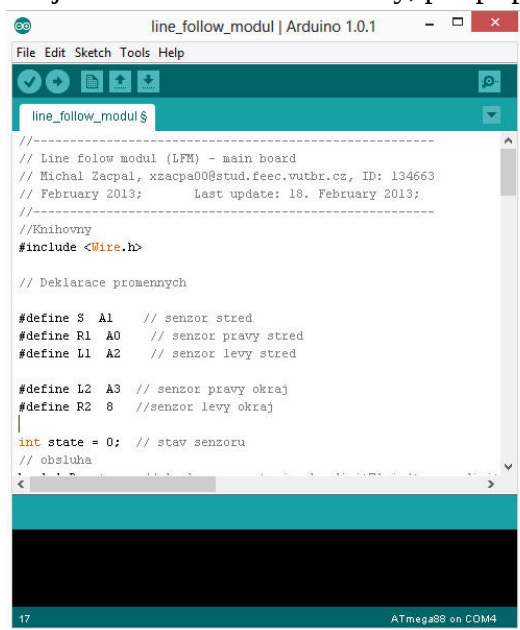
ATmega8 obsahuje osmibitový mikroprocesor AVR s RISC architekturou. Architektura RISC označuje mikroprocesory s redukovanou instrukční sadou. Kde méně často se vyskytující instrukce je potřeba realizovat za pomoci několika jednoduchých instrukcí na rozdíl od mikroprocesorů s architekturou CISC obsahujících komplexní instrukční sadu. Mikroprocesor ATmega8 je schopen pracovat na frekvenci až 16 MHz, které je možné dosáhnout externím oscilátorem, jinak pracuje na frekvenci 1 – 8 MHz (vnitřní oscilátor). Mikrokontrolér má Flash paměť velikosti 8kB pro nahrání programu, 1kB SRAM pro provozní data a 512B EEPROM. Dále obsahuje jeden šestnáctibitový a dva osmibitové čítače/časovače, tři PWM kanály.

Vylepšení ATmega328 spočívají ve větší paměti Flash, a to 32kB, 2kB SRAM a 1kB EEPROM. Je schopná pracovat na frekvenci až 20 MHz při použití externího oscilátoru a obsahuje šest PWM kanálů.

2.2.2 Arduino

Arduino je open-source platforma využívající mikrokontrolérů ATmega od firmy Atmel. Jedná se o desku obsahující mikrokontrolér, konektory pro připojení různých zařízení na digitální nebo analogové vstupy/výstupy, USB konektor pro komunikaci

s PC, konektor pro připojení externího napájení a další podpůrné obvody. Těchto desek je několik druhů, rozlišují se větším či menším počtem vstupně/výstupních pinů a použitým typem mikrokontroléru. Jednotlivé verze jsou rozlišovány názvy, např. Arduino UNO, Arduino Duemilanove, Arduino DUE více na: <http://www.arduino.cc/en/Main/Products>. K nahrávání programu do mikrokontroléru se používá vývojové prostředí Arduino IDE viz. Obr.2.2. ukazující jak desku, tak vývojové prostředí. Deska se k počítači připojuje přes USB konektor umístěný taktéž na desce. V PC se potom Arduino jeví jako standardní sériová linka. Komunikace a nahrávání programu do mikrokontroléru pak probíhá přes softwarově simulovanou sériovou linku. K tomu, aby PC viděl Arduino jako sériovou linku, slouží ve starších verzích FTDI čip, v novějších to je mikrokontrolér ATmega8U2, který k tomu má předprogramovaný firmware. Mikrokontrolér obsahuje Arduino bootloader což je program, který se při startu mikrokontroléru postará o jeho základní nastavení registrů, USART rozhraní a další. Pokud chceme použít mikrokontrolér s tímto bootloaderem ve vlastním navrhnutém zařízení a nechceme kupovat celou Arduino desku, lze zakoupit i samotný mikrokontrolér, a to ATmega 328. V tomto projektu je ale použit i mikrokontrolér ATmega8, do kterého musel být bootloader nahrán. Vzhledem k tomu, že se jedná o open-source platformu, je tento bootloader volně dostupný, stejně tak jako schéma Arduino desky, pro případ kdybychom si ji chtěli vyrobit sami.



Obr. 2.2. Ukázka vývojového prostředí Arduino IDE a deska Arduino UNO

V této práci obsahují všechny mikrokontroléry Arduino bootloader. A k nahrávání programu do nich byl použit UASB/UART převodník, který se zapojí na piny RX, TX, GND a RESET mikrokontroléru. Programovací prostředí nazývané Arduino IDE je napsáno v jazyce Java. Díky tomu je možné je spustit jak na operačním systému Windows, tak i Linux či MAC OS X. Programovací jazyk na tvorbu programů se nazývá Wiring a je založený na C/C++. Pro nahrání programu do mikrokontroléru je nejprve nutné zvolit verzi Arduina a poté zmáčknutím jednoho tlačítka dojde

k přeložení programu do strojového kódu a nahrání do mikrokontroléru. Existuje i Arduino plugin do Microsoft Visual Studia a programy lze tak psát i nahrávat přímo v něm.

Tuto platformu jsem zvolil, protože s ní již delší dobu pracuji a je uživatelsky přívětivá. Odpadá zde potřeba nastavování registrů, stačí pouze zavolat funkci, která provede vše za nás. Například přivedení log.1 na *pin1* (PD1 – ATmega 328) se zapíše následovně:

```
digitalWrite(pin1,HIGH);
```

nebo pro přečtení v jaké úrovni se nachází *pin2* (PD2 – ATmega 328):

```
digitalRead(pin2);
```

Na začátku je třeba definovat zda, se jedná o pin vstupní či výstupní, což se provede následovně:

```
pin1 = pinMode(OUTPUT);           // výstupní
```

```
pin2 = pinMode(INPUT);            // vstupní
```

Někdy se ale stejně nastavování registrů nevyhneme, například v případě nastavení frekvence PWM.

2.2.3 Komunikační sběrnice I2C

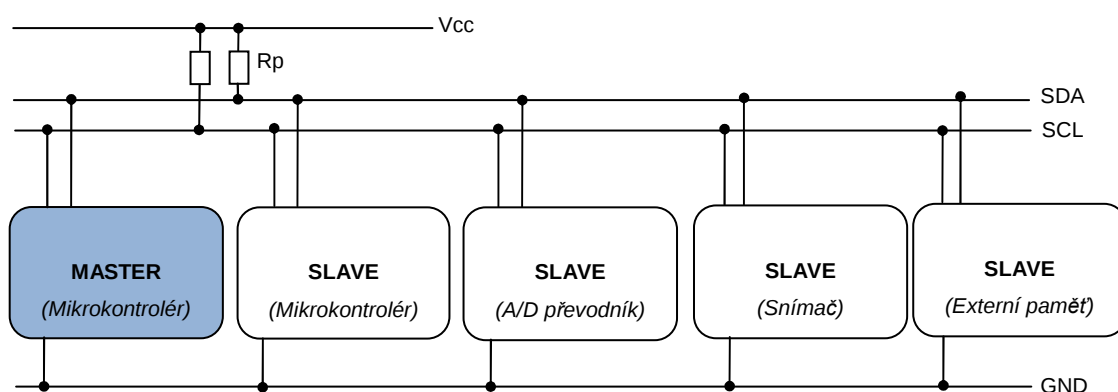
I2C (bus) sběrnice také označovaná jako IIC bus je zkratkou pro Internal-integrated-circuit bus. Jde o interní datovou sběrnici vyvinutou firmou Philips ke komunikaci a přenosu dat mezi integrovanými obvody, jako jsou mikrokontroléry, A/D , D/A převodníky, paměti a další. Sběrnice propojuje zařízení typu master (řídící) a zařízení typu slave (posluchač, otrok), ale umožňuje i propojení typu master – master. Ke komunikaci mezi jednotlivými zařízeními využívá sériového přenosu dat, k čemuž jí stačí pouze dva vodiče, a to SDA – sloužící k přenosu dat a SCL – udávající hodinový signál, který udává master. Oba vodiče jsou připojeny přes pull up rezistor ke kladnému napětí z důvodu zajištění vysoké úrovně v klidovém stavu sběrnice. Každé zařízení připojené ke sběrnici má danou 7 – bitovou adresu (některá zařízení 10 - bitovou) z toho plyne, že ke sběrnici je možno připojit maximálně 128 (1024) zařízení. Ve skutečnosti jsou tato čísla nižší. V původní verzi byla frekvence hodin 100 kHz, v dnešní verzi je to až 5 MHz, rychlost celé sběrnice závisí na nejpomalejším článku. Princip propojení sběrnice je uveden na Obr. 2.3.

Princip přenosu dat je následující: v klidovém stavu jsou vodiče SDA i SCL ve vysoké úrovni. Přenos dat je zahájen START bitem tj. SDA přejde do nízké úrovně zatím co SCL je stále ve vysoké. Vysílač (master nebo slave) přivede na datový vodič 8 datových bitů, ty jsou hodinovými impulsy, generovanými masterem na vodiči SCL posouvány dále, tedy při každém impulsu na SCL je přenesen jeden bit na SDA. Přenos začíná od bitu s nejvyšší váhou. Ukončení je provedeno STOP bitem tj. SDA přejde do vysoké úrovně zatím co SCL je také ve vysoké úrovni. Při přenosu je po START bitu nejprve vyslána 7 – bitová adresa, následně je vyslán bit R/W (read/write) indikující

čtení nebo zápis. Vyslanou adresu připojená zařízení porovnají se svou adresou. Pokud dojde ke shodě s adresovaným zařízením, potvrdí přijetí adresy bitem ACK, pak začne být přenášena informace rozdělená na byty. Úspěšný přenos jednoho bytu je potvrzen znovu bitem ACK a je možné přenést další byte. Pokud je přenos potvrzen bitem NACK znamená to, že poslední přenášený byte byl poslední a dojde k ukončení komunikace (bitem STOP).

Aby při komunikaci nedocházelo ke kolizím, využívá metoda I2C metodu s detekcí kolize. Každé připojené zařízení může zahájit přenos pouze pokud je sběrnice v klidovém stavu tj. když je na vodičích SDA a SCL vysoká úroveň.

Protože na I2C sběrnici vlastní patent firma Philips, bylo vyvinuto několik ekvivalentních sběrnic pracujících na stejném principu. Jednou takovou je sběrnice TWI (Two-wire interface) od firmy Atmel. Vzhledem k tomu, že jsou v práci využívány procesory této firmy (popsané výše) je zde použita i tato sběrnice, protože se ale jedná pouze o ekvivalent jiné firmy bude dále v textu používán rozšířenější výraz I2C.



R_p – Pull up rezistory
Vcc – Napájecí napětí
GND – Signálová zem
SDA – Přenos dat
SCL – Hodinový signál

Obr. 2.3. Schéma připojení zařízení na sběrnici I2C

2.2.3.1 Arduino a I2C

Pro komunikaci po I2C / TWI sběrnici využívá platforma arduino knihovnu Wire.h obsahující třídu Wire. Tato část se zabývá popisem funkcí z této knihovny použitých v práci. Tab.2.1. obsahuje výpis těchto funkcí a informaci zda se používá u zařízení typu master nebo slave.

První funkcí je *Wire.begin()*; ta slouží k nastavení zařízení jako master nebo slave do I2C sběrnice. Pokud je volána bez parametru, zařízení je nastaveno jako zařízení typu master, pokud s parametrem (7-bitová adresa) je zařízení nastaveno jako zařízení typu slave s adresou danou parametrem.

Jako další je *Wire.requestFrom()*; používaná v zařízení typu master k vyžádání dat od zařízení typu slave. Její syntaxe je následující:

```
Wire.requestFrom(adress, quantity, stop);
```

Parametr *adress* udává adresu slave zařízení, *quantity* počet požadovaných bitů a *stop*, pokud je true, tak master uvolní sběrnici po vyslání žádosti, pokud false master bude udržovat aktivní spojení. Poslední zmíněný parametr je nepovinný a může být vynechán. Návratovou hodnotou funkce je počet přijatých bitů od slave zařízení.

K zahájení komunikace a jejímu ukončení slouží funkce *Wire.beginTransmission()*; (parametrem je 7-bitová adresa zařízení typu slave se kterým chceme komunikovat) a *Wire.endTransmission()*; (volitelným parametrem může být proměnná typu bool, pokud je true, tak master uvolní sběrnici po vyslání žádosti, pokud false master bude udržovat aktivní spojení) návratovou hodnotou je byte udávající informaci o přenosu: 0 – úspěšný přenos, 1 – příliš dlouhý údaj, který se nevejde do bufferu, 2 – přijetí bytu NACK (no acknowledge) při přenosu adresy, 3 – přijetí bytu NACK (no acknowledge) při přenosu dat, 4 – jiná chyba .

K odeslání dat slouží funkce *Wire.write()*; jejím parametrem jsou posílaná data, případně nepovinným parametrem počet bytů udávající délku posílaných dat. Vkládá se mezi funkce *Wire.beginTransmission()*; a *Wire.endTransmission()*;. Následující příklad uvádí poslání proměnné s názvem *data* o délce 5 bytů na zařízení s adresou 3.

```
Wire.beginTransmission(3);
Wire.write(data,5);
Wire.endTransmission();
```

Funkce *Wire.available()*; vrací počet bytů dostupných na sběrnici k přečtení. Ke čtení příchozích dat slouží *Wire.read()*;

Funkce *Wire.onReceive()*; definuje název funkce, která bude provedena pokud zařízení typu slave obdrží příchozí komunikaci od mastera. Parametrem je název volané funkce.

Funkce *Wire.onRequest()*; definuje název funkce, která bude provedena pokud zařízení typu master požádá zařízení slave o data. Parametrem je název volané funkce.

Tab. 2.1. Přehledný výpis funkcí knihovny Wire.h s informací kde lze použít

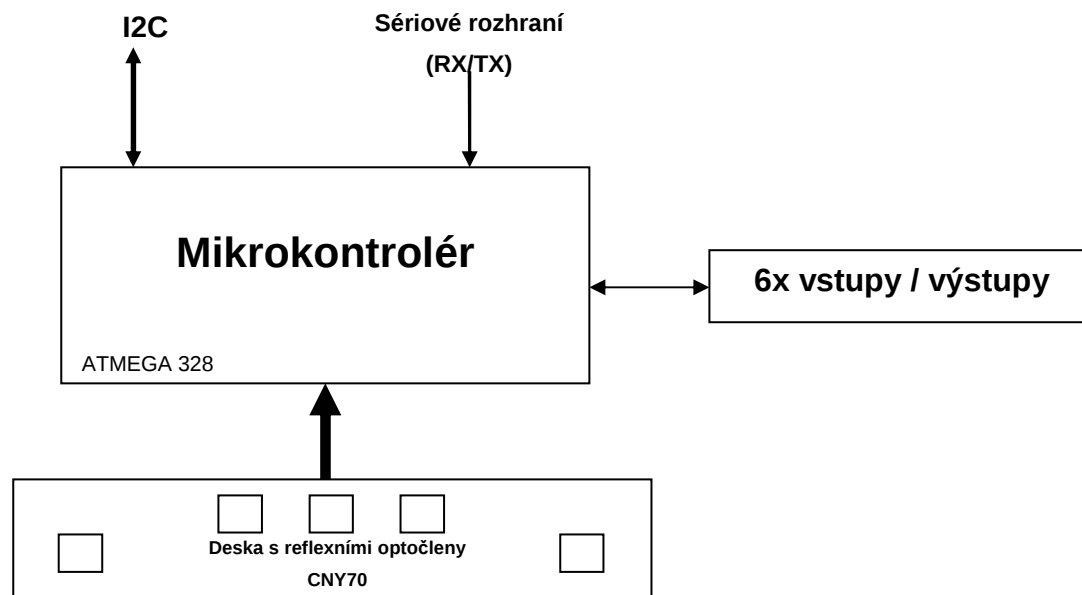
Funkce	master/slave
Wire.begin	Master i slave
Wire.requestFrom	Master
Wire.beginTransmission	Master
Wire.endTransmission	Master
Wire.write	Master i slave
Wire.available	Master i slave
Wire.read	Master i slave
Wire.onReceive	Slave
Wire.onRequest	Slave

3 MODUL PRO SLEDOVÁNÍ ČÁRY (LINE FOLLOW MODUL)

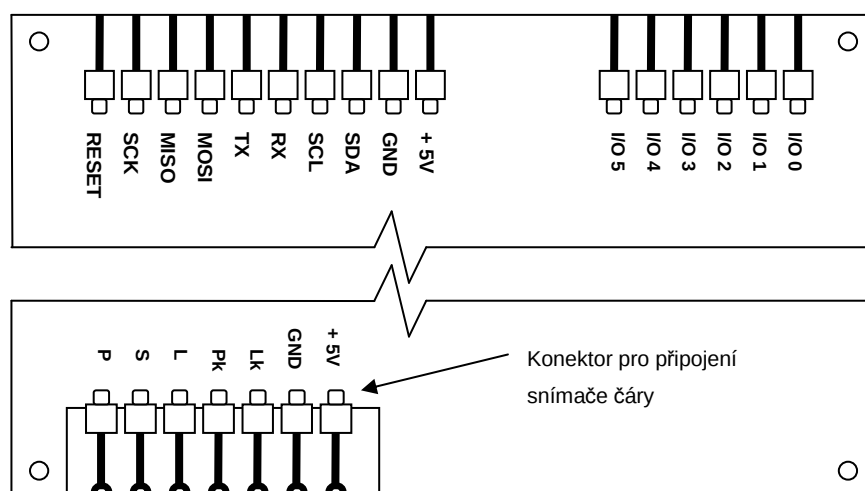
Tento modul je určen ke zjištění polohy čáry. Stejně tak jako modul pro řízení motorů je univerzálně použitelný díky své schopnosti komunikace přes vnitřní I2C sběrnici robota. Tato kapitola se zabývá popisem jeho hardwarového a softwarového vybavení.

3.1 Popis modulu

Blokové schéma je na Obr.3.1. jeho funkce je následující: Mikrokontrolér sbírá údaje ze snímače čáry (desky s reflexními optočleny), jak k tomu dochází je popsáno níže v kapitole Princip snímání čáry. Tyto údaje následně zpracuje a pokud o to nadřazený mikrokontrolér (master) požádá, pošle mu je přes I2C sběrnici. Master si může požádat o informaci, pod kterým optočlenem se čára právě nachází nebo zda se pod některým optočlenem čára nachází či nikoliv. Komunikace přes I2C modulu zajišťuje univerzální použití a možnost jeho zařazení do většího celku jako slave zařízení. Sériové rozhraní (RX,TX piny) slouží k programování modulu. Dále má vyvedeno šest vstupně výstupních pinů pro případné připojení dalších snímačů. Uspořádání konektorů odpovídá tomu, popsaném, v kapitole 2 a je uvedeno na Obr. 3.2



Obr. 3.1. Blokové schéma modulu pro sledování čáry



Obr. 3.2. Uspořádání konektorů modulu pro sledování čáry

3.2 Elektronika

Tento modul se skládá ze dvou částí, a to hlavní řídicí desky (Obr. 3.6.) a snímače čáry (desky s reflexními optočleny) (Obr. 3.7.). Obě desky jsou propojeny sedmižilovým plochým kabelem. Schéma zapojení je uvedeno v příloze 2 list č.1 (řídicí deska modulu) a v příloze 2 list č.2 (snímač čáry).

Pro připojení k hlavní řídicí jednotce modul využívá uspořádání konektorů uvedeném na Obr. 3.2 a popsaném v kapitole 2. Modul má na rozdíl od modulu pro řízení motorů vyvedeny piny pro programování přes ISP rozhraní. Led dioda označená jako PWRLED1 slouží k indikaci napájecího napětí na řídicí desce a PWRLED k indikaci napájecího napětí na snímači čáry. Tlačítko RESET je určeno k restartování mikrokontroléru.

3.2.1 Řídicí mikrokontrolér

Řídicím mikrokontrolérem je ATmega8, její stručný popis je uveden v kapitole 2. Je taktována na 16 MHz. Jejím úkolem je čtení hodnot ze snímače čáry (reflexních optočlenů) a předání této informace přes I2C sběrnici nadřazenému zařízení typu master, a to v podobě informace o poloze čáry pod snímačem čáry.

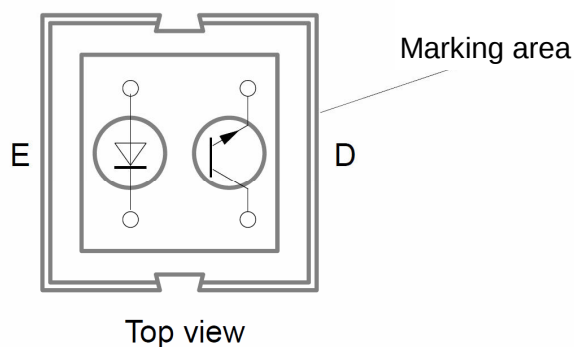
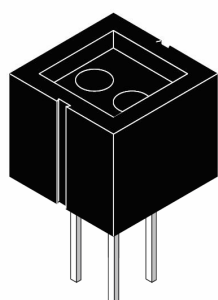
Mikrokontrolér je možné naprogramovat pomocí programátoru k tomu určenému. Aby nebyla potřeba přenášet čip mezi deskou modulu a programátoru, má vyvedeny piny pro ISP rozhraní (MISO, MOSI, SCK, RESET). Nicméně mikrokontrolér obsahuje arduino bootloader, díky tomu je možné jej programovat za použití programovacího prostředí Arduino a USB/UART převodníku, který se připojí k pinům RX, TX a RESET. Tento způsob programování je preferován, více v kapitole 2. Tab. 3.1. uvádí přehledný popis využitých pinů mikrokontroléru.

Tab. 3.1. Popis zapojení pinů mikrokontroléru modulu pro sledování čáry

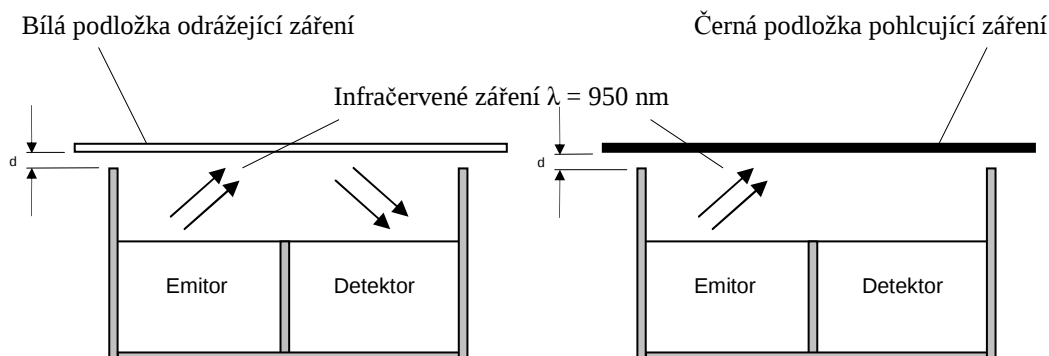
Pin	Označení pinu	Funkce
PD 0	RX	sériové rozhraní
PD 1	TX	sériové rozhraní
PD 2	DIO0	Digitální vstup/výstup
PD 3	DIO1	Digitální vstup/výstup
PD 4	DIO2	Digitální vstup/výstup
PD 5	DIO3	Digitální vstup/výstup
PD 6	DIO4	Digitální vstup/výstup
PD 7	DIO5	Digitální vstup/výstup
PB 0	Nepřipojen	-
PB 1	CNY70	Pravý krajní optočlen
PB 2	Nepřipojen	-
PB 3	MOSI	ISP
PB 4	MISO	ISP
PB 5	SCK	ISP
PC 0	CNY70	Pravý optočlen
PC 1	CNY70	Střed optočlen
PC 2	CNY70	Levý optočlen
PC 3	CNY70	Levý krajní optočlen
PC 4	I2C (SDA)	I2C
PC 5	I2C (SCL)	I2C
PC 6	Reset	Reset

3.2.2 Princip snímání čáry

Ke snímání čáry je použit reflexní optočlen CNY70. Jedná se o infračervenou (IR) led diodu (Emitter) a fototranzistor (Detektor) v jednom pouzdře viz. Obr. 3.3. Tyto prvky pracují s infračerveným zářením o vlnové délce 950nm. Optočlen je možné použít ve vzdálenosti 0 - 5 mm od podkladu. Princip funkce je následující: IR led emituje infračervené záření, podle podkladu, na který toto záření dopadne, se paprsky buď odrazí a nebo jsou podkladem pohlceny. Podklad černé barvy záření pohltí, kdežto podklad bílé barvy záření odrazí. Odražené paprsky pak dopadnou na fototranzistor, jehož fotocitlivá vrstva je bází tranzistoru, tím dojde k jeho sepnutí. Pokud se paprsky neodrazí tranzistor zůstane rozepnutý. Princip funkce je znázorněn na Obr 3.4.



Obr. 3.3. Reflexní optočlen CNY70 (převzato z [7])



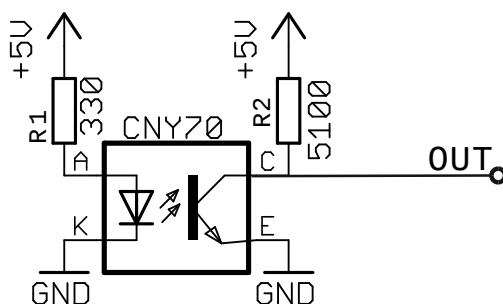
- d – vzdálenost mezi podložkou a reflexním optočlenem CNY70

Obr. 3.4. Princip funkce reflexního optočlenu CNY70

Zapojení reflexního optočlenu je na Obr. 3.5. Odpor R1 slouží k omezení proudu do IR led diody. Odpor R2 slouží jako PULL UP rezistor. Pokud je fototranzistor sepnutý (tj. optočlen se nachází nad podkladem bílé barvy a na fotocitlivou vrstvu fototranzistoru dopadá záření) je vstup mikrokontroléru připojen na zem, což je detekováno mikrokontrolérem jako log.0. V případě, že je fototranzistor rozepnutý (optočlen se nachází nad podkladem černé barvy a na fotocitlivou vrstvu fototranzistoru žádné záření nedopadá) je vstup mikrokontroléru připojen přes pull up rezistor na +5V, to je detekováno jako log.1. V případě, že by pull up rezistor chyběl a vstup mikrokontroléru by byl zapojen přímo na kolektor fototranzistoru, tak v rozepnutém stavu by na vstupu byla neurčitá úroveň (v sepnutém stavu by vstup byl opět připojen na zem a byla by detekována log.0). Výpočet pull up rezistoru je následující a postačí k němu pouze znalost ohmova zákona. Z katalogového listu [7] byl zjištěn maximální proud tekoucí fototranzistorem v sepnutém stavu a to $I_c = 1\text{mA}$. Při napájecím napětí $U = 5\text{V}$ se pull up odpor vypočte následně:

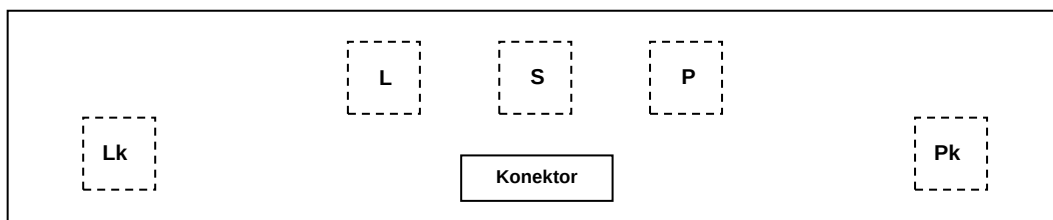
$$R = \frac{U}{I_c} = \frac{5}{1 \cdot 10^{-3}} = 5000\Omega. \quad (3.1)$$

Z řady dostupných odporů je této hodnotě nejbližší 5100Ω ten byl také zvolen do zapojení.

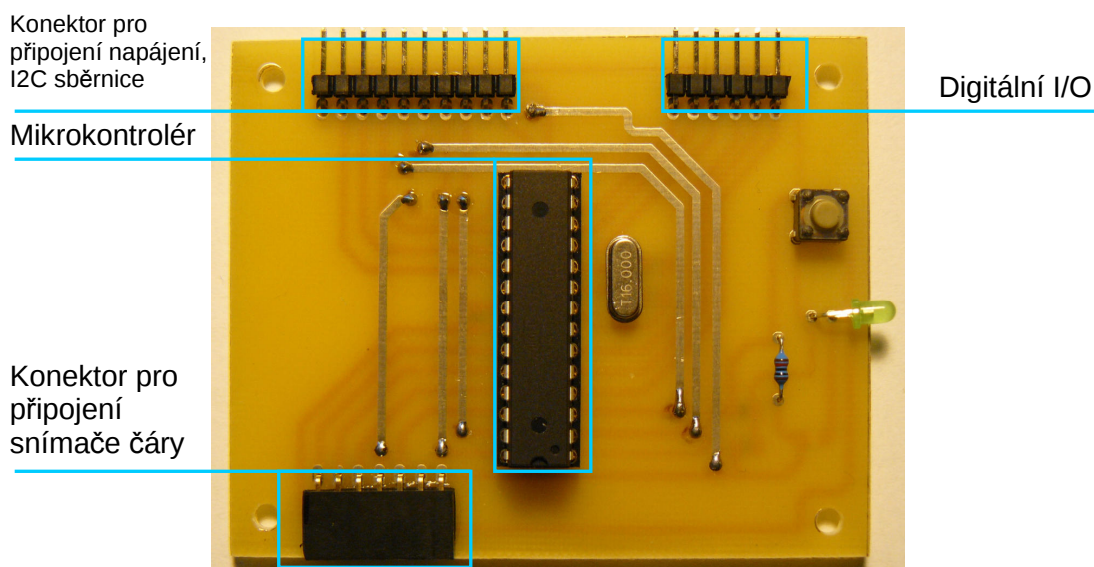


Obr. 3.5. Zapojení snímače čáry (reflexního optočlenu CNY70)

Snímač čáry je vybaven pěti reflexními optočleny CNY70, jejich osazení je zobrazeno na Obr. 3.6. Pro hlavní sledování čáry slouží tři prostřední (L – levý, S – středové, P - pravý), dva krajní (Pk – pravý krajní a Lk – levý krajní) jsou využívány převážně jako záchytné při prudkých zatáčkách.



Obr. 3.6. Uspořádání reflexních optočlenů CNY70 na desce pro snímání čáry (Pohled shora – snímače pod deskou)



Obr. 3.7. Osazená řídicí deska modulu pro sledování čáry

3.3 Řídicí algoritmus

Modul pro sledování čáry se na sběrnici I2C chová jako zařízení typu slave. Vývojový diagram znázorňující řídicí algoritmus je na Obr. 3.8. Na počátku proběhne blok inicializace, kdy dojde k nastavení adresy slave zařízení, nastavení vstupních a výstupních pinů a počáteční inicializace proměnných. Dále pokračuje hlavní smyčka algoritmu, kde dochází ke čtení údajů ze snímačů a jejich následnému zpracování.

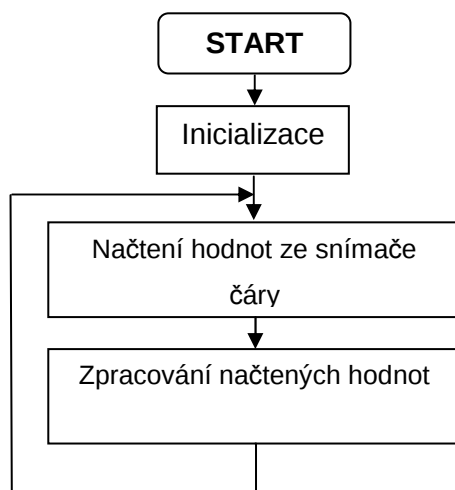
Komunikace je zajištěna za pomoci dvou funkcí, a to *WireOnReceive()*; a *WireOnRequest()*; z knihovny Arduina Wire.h. Obě jsou blíže popsány v kapitole 2. Na Obr. 3.9. je vývojový diagram znázorňující průběh funkce *WireOnReceive()*; v tomto případě nazvané jako *receive_event()*; Tato funkce je zavolána pokud modul přijímá data, jak je vidět z vývojového diagramu, v tomto případě je pouze uloží. Obr. 3.10.

zobrazuje vývojový diagram znázorňující průběh funkce *WireOnRequest()*; nazvané jako *request_event()*. Tato funkce je zavolána pokud master požaduje data od modulu. Průběh je následující: Pokud byl posledně přijatý znak ‚A‘, tak odešle masterovi údaj o poloze čáry, pokud byl posledně přijatý znak ‚B‘, odešle informaci zda se čára nachází pod robotem či nikoliv. V případě, že přijatý znak nebyl ani jeden z předešlých neodešle žádnou informaci.

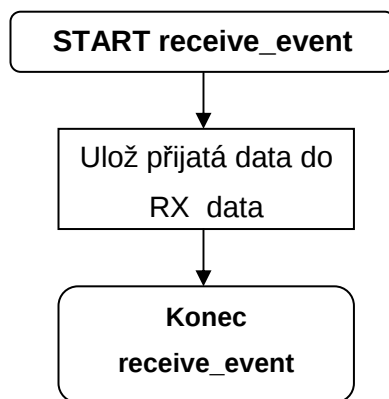
Data jsou odesílána jako 1 byte v případě, že chceme informaci zda se čára pod snímačem nachází či ne; dostaneme buď 1 (pokud ano) nebo 0 (ne). V případě informace o poloze čáry dostaneme hodnotu 0 – čára není pod žádným čidlem. 1 – čára se nachází pod levým krajním čidlem (00001b) 2 – levým čidlem (00010b), 4 – středním čidlem (00100b), 8 – pravým čidlem (01000b), 16 – pravým krajním čidlem (10000b).

3.3.1 Komunikace s modulem

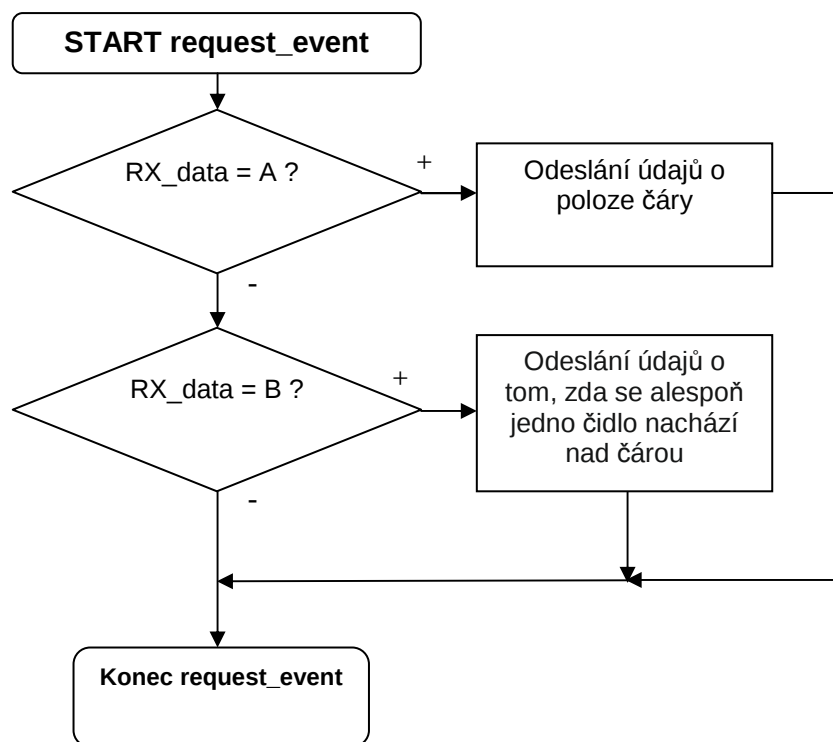
Postup komunikace s modulem je následující: Zařízení typu master vyšle znak odpovídající požadované informaci, počká a poté si od zařízení typu slave (modul) vyžádá informaci. Tu mu modul předá za předpokladu, že poslal platný znak.



Obr. 3.8. Vývojový diagram znázorňující algoritmus modulu pro sledování čáry



Obr. 3.9. Vývojový diagram znázorňující průběh funkce *receive_event()*;



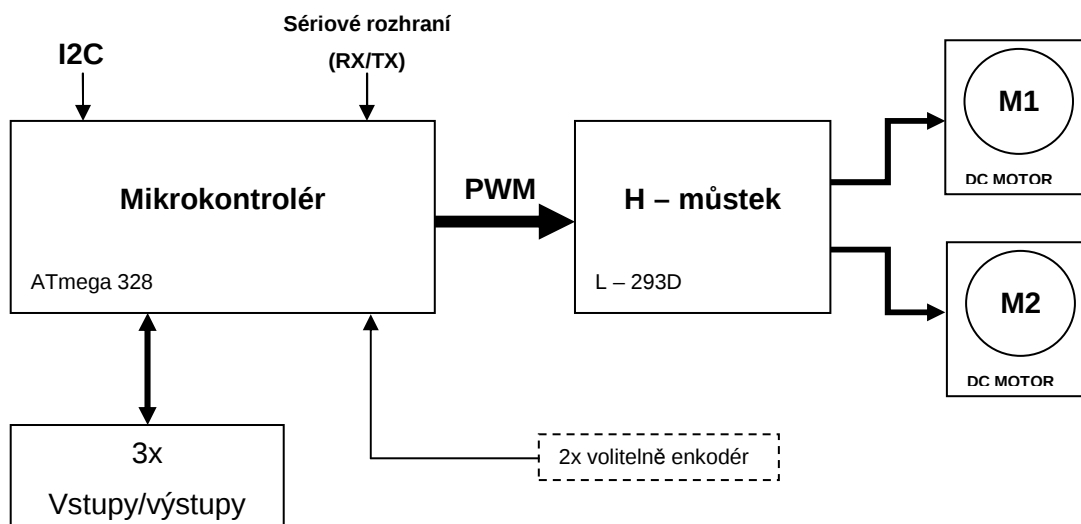
Obr. 3.10. Vývojový diagram znázorňující průběh funkce `request_event()`;

4 MODUL PRO ŘÍZENÍ MOTORŮ (DRIVER MODUL)

Tento modul byl navrhnout a vytvořen v rámci semestrální práce [1]. Pro tuto práci prošel modul drobnými změnami popsány níže. V této kapitole je uveden jeho popis a představeno hardwarové a softwarové vybavení.

4.1 Popis modulu

Modul byl navržen tak, aby byl univerzálně použitelný pro řízení dvou stejnosměrných motorků s maximálním odběrem až 600mA na každý motor. Je možné jej použít samostatně k řízení robota nebo ho zařadit do většího celku, což je případ této práce. Ke komunikaci s okolními periferiemi využívá především I2C sběrnice (popsané v kapitole 2), ale po úpravě řídicího algoritmu je také možné jej ovládat pomocí sériového rozhraní (RX, TX piny). To bylo využito převážně v semestrálním projektu, kdy byl modul řízen za pomoci bluetooth.



Obr. 4.1. Blokové schéma modulu pro řízení motorů

Blokové schéma je uvedeno na Obr. 4.1. Mikrokontrolér ATmega 328, který je v tomto případě ovládán přes I2C řídí činnost H-můstku L293D, který obstarává chod dvou stejnosměrných motorků.

Aby bylo možné modul připojit k hlavní řídicí jednotce za pomoci uspořádání konektorů popsaném v kapitole 2, bylo pozměněno jeho zapojení desky plošných spojů oproti původní verzi ze semestrální práce. Byly odebrány konektory (se standardním servo zapojením) pro připojení externích zařízení a byly ponechány pouze tři vstupy/výstupy. Dále byl odebrán DC-DC měnič Traco TSR 1-2450. Napájení je nyní odebíráno ze zdroje řídicí jednotky, stejně tak napájení pro motory.

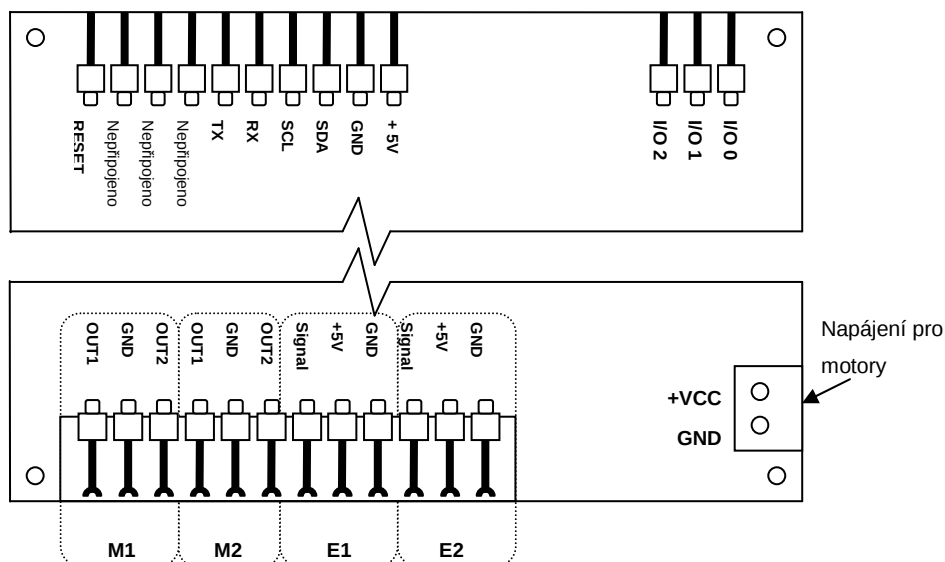
4.2 Elektronika

Schéma zapojení je uvedeno v příloze 3. Řídicím mikrokontrolérem je ATmega 328, ke které je připojen H-můstek L293D. K tomu jsou připojeny dva stejnosměrné motorky, každý za pomoci tří pinů. Dva krajní jsou připojeny ke kontaktům motorků a jeden prostřední spojuje kovový obal motoru a nulový potenciál napájení modulu.

K indikaci napájecího napětí +5V pro napájení řídicích prvků slouží led dioda označená ve schématu jako PWR+5V k indikaci napájecího napětí pro motory zase led dioda označená jako PWR+VCC.

Na Obr 4.2 je uvedeno uspořádání konektorů pro připojení k řídicí jednotce a také pro připojení motorů a enkodérů. Modul obsahuje konektory pro připojení komunikace přes I2C, ale také sériové rozhraní (RX,TX piny) a má vyveden pin pro reset. Uspořádání konektorů k připojení enkodérů odpovídá standardnímu servokabelu (signál / napájení 5V / GND).

Tlačítko RESET je určeno k resetování mikrokontroléru.



Obr. 4.2. Uspořádání konektorů modulu pro řízení motorů

4.2.1 Řídicí mikrokontrolér

Řídicím mikrokontrolérem je již zmiňovaná ATmega 328P, jejíž stručný popis je uveden v kapitole 2. V tomto případě je taktován na 16 MHz. Jeho úkol je příjem příkazů přes I2C sběrnici a jejich následné zpracování. Ovládá H-můstek a pokud jsou připojeny enkodéry, tak zpracovává údaje z nich (po úpravě řídicího algoritmu). V Tab. 4.1. je uveden přehledný popis zapojení pinů mikrokontroléru.

Jako v případě modulu pro sledování čáry mikrokontrolér obsahuje arduino bootloader. Je tedy možné jej programovat za použití USB/UART převodníku, který se připojí k pinům RX,TX a RESET podrobnější popis je uveden opět v kapitole 2. Na

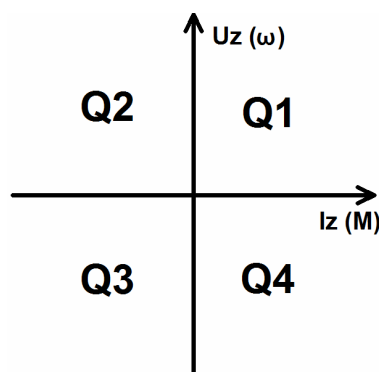
rozdíl od modulu pro sledovní čáry nemá modul vyvedeny piny pro ISP rozhraní (MISO, MOSI, SCK), a to z důvodu nevyužívání rozhraní. Tím také bylo ušetřeno místo na desce plošných spojů.

Tab. 4.1. Popis zapojení pinů mikrokontroléru modulu pro řízení motorů

Pin	Označení pinu	Funkce
PD 0	RX	sériové rozhraní
PD 1	TX	sériové rozhraní
PD 2	Enkodér 2	Externí přerušení
PD 3	Enkodér 1	Externí přerušení
PD 4	Nepřipojen	-
PD 5	Nepřipojen	-
PD 6	H-Mústek	PWM
PD 7	H-Mústek	Digitální výstup
PB 0	H-Mústek	Digitální výstup
PB 1	H-Mústek	PWM
PB 2	H-Mústek	Blokování motorů
PB 3	Nepřipojen	-
PB 4	Nepřipojen	-
PB 5	Nepřipojen	-
PC 0	H-Mústek	Blokování motorů
PC 1	AIO0	Analogový vstup/výstup
PC 2	AIO1	Analogový vstup/výstup
PC 3	AIO2	Analogový vstup/výstup
PC 4	I2C (SDA)	I2C
PC 5	I2C (SCL)	I2C
PC 6	Reset	Reset

4.2.2 Ovládání motorů

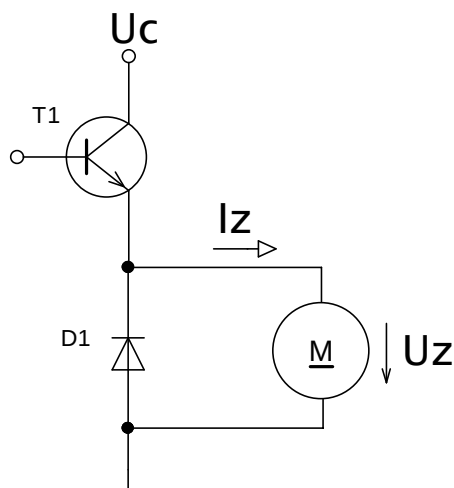
Pro ovládání motorů není možné je připojit na výstupy mikrokontroléru, protože z výstupu jednoho pinu je možné odebírat maximálně 40 mA. Motory je tedy potřeba řídit za pomoci výkonového stupně. Motory je možné řídit v různém počtu kvadrantů viz. Obr 4.3.



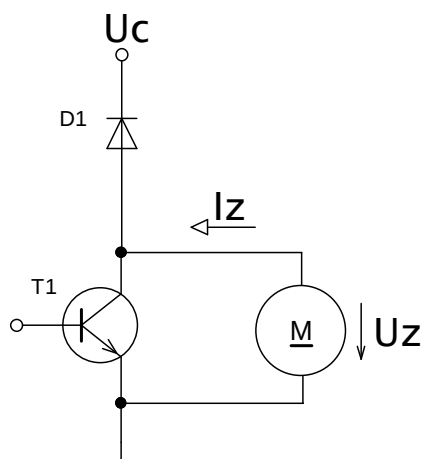
Obr. 4.3. Kvadranty řízení motoru

4.2.2.1 Jednokvadrantové řízení

Za pomoci jednokvadrantového budiče je možné motor ovládat pouze v jednom směru a není možné jej brzdit. Z jeho zapojení je vidět, že se jedná o elektronický spínač. Dioda slouží k ochraně tranzistoru před indukovanými napěťovými špičkami vznikajícími při běhu motoru a také přebírá proud zátěže při vypnutí tranzistoru. Budič pracuje pouze v jednom kvadrantu, a to v Q1 nebo Q2.



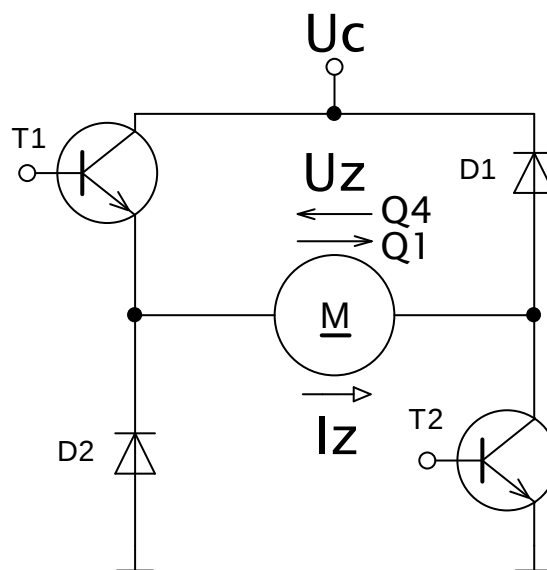
Obr. 4.4. Budič pracující v kvadrantu Q1



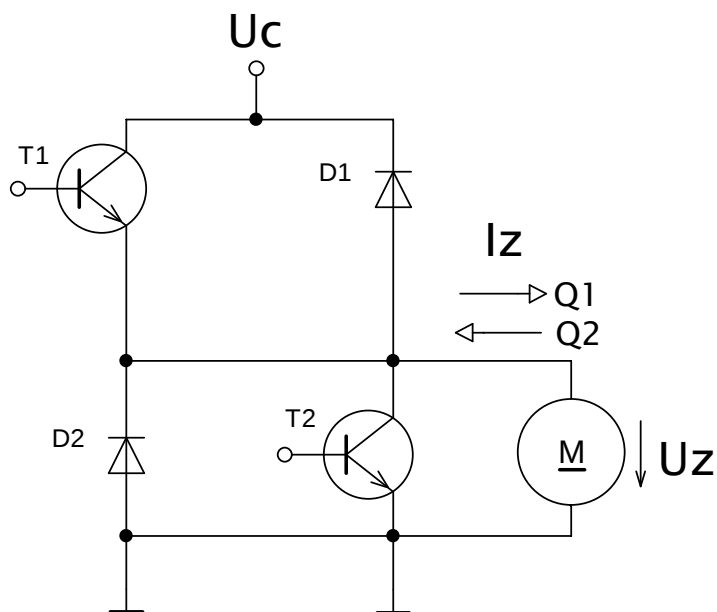
Obr. 4.5. Budič pracující v kvadrantu Q2

4.2.2.2 Dvoukvadrantové řízení

Tento budič umožňuje řízení motoru v kvadrantech Q1 a Q4 (Obr.4.6) a v kvadrantech Q1 a Q2 (Obr.4.7). Je tedy možné za pomoci něj ovládat motor v jednom směru a brzdit jej. Diody opět slouží k ochraně tranzistoru převzetí proudu při vypnutí jako v předešlém případě.



Obr. 4.6. Budič pracující v kvadrantech Q1 a Q4



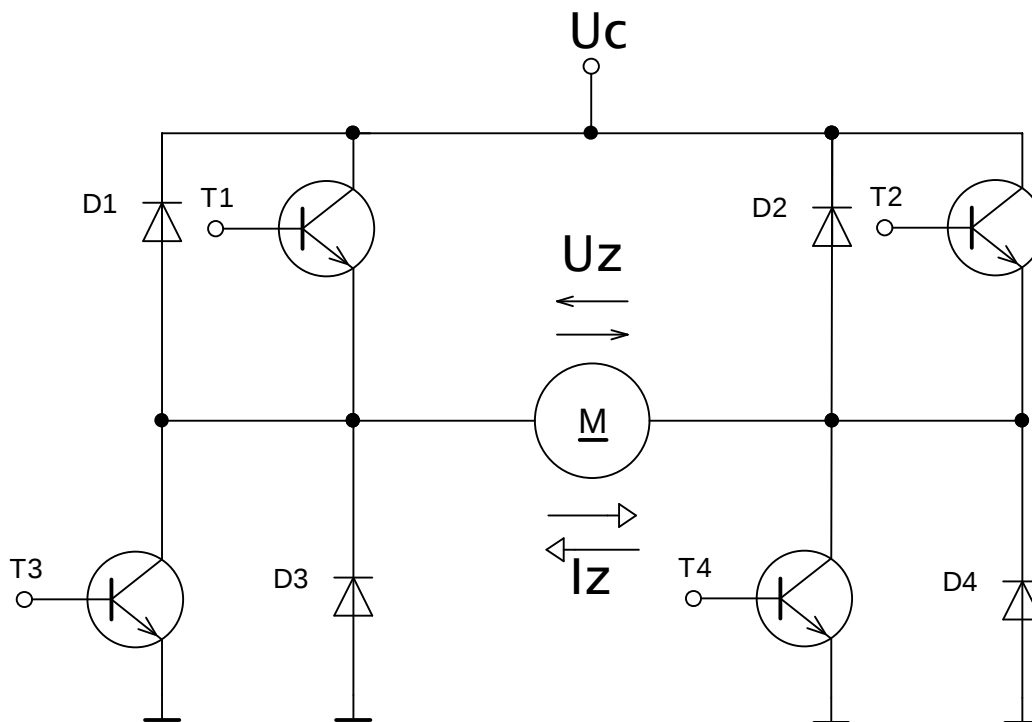
Obr. 4.7. Budič pracující v kvadrantech Q1 a Q2

4.2.2.3 Čtyřkvadrantové řízení

Čtyřkvadrantovým budičem je možné řídit motor ve všech kvadrantech. Také se mu říká H-můstek, tento název je odvozen od jeho typické grafické reprezentace elektrického obvodu. Zapojení čtyřkvadrantového budiče je na Obr. 4.8. A do modulu pro řízení motoru byl zvolen právě tento typ.

Zjednodušeně je H-můstek možné popsat jako elektronický obvod umožňující proudu téct jedním anebo druhým směrem. Jeho funkce je znázorněna na Obr. 4.8. Pokud sepne tranzistory T1 a T4 motor se bude točit určitým směrem, pokud

sepneme tranzistory T3 a T2 motor se bude točit obráceně. Je patrné, že musíme vždy sepnout tranzistory do kříže. Pokud bychom sepnuli tranzistory pod sebou, zkratovali bychom napájecí zdroj. Sepnutím tranzistorů vedle sebe motor zabrzdíme. Diody v zapojení opět slouží k ochraně tranzistoru před indukovanými napěťovými špičkami vznikajícími při běhu motoru a převzetí proudu zátěže při vypnutí motoru.



Obr. 4.8. Čtyřkvadrantový budič

4.2.2.4 H - můstek L293D

V modulu je použit integrovaný obvod L293D, ten v sobě obsahuje dva plnohodnotné H-můstky, z nichž každý lze ovládat samostatně. Pro spínání používá bipolární tranzistory. D na konci označení znamená, že tato verze má již integrované ochranné diody. H-můstek je schopný každému motoru dodat 600 mA, to je 1200 mA na oba motory. Všechny vstupy tohoto obvodu jsou kompatibilní s TTL logikou a je tedy možné je připojit k výstupům mikrokontroléru.

K mikrokontroléru je H-můstek připojen za pomoci šesti pinů. Dva jsou určeny k blokování motorů a čtyři slouží k určení směru otáčení, dva z těchto čtyř jsou pak připojeny k PWM výstupům mikrokontroléru (popsáno v Tab 4.1.). Určení směru otáčení motorků je prováděno přivedením odpovídajících logických úrovní, příslušejících požadovanému směru, na vstupy obvodu L293D. Kombinace logických úrovní s odpovídajícími směry jsou uvedeny v Tab. 4.2. Pro tuto aplikaci nebyla potřeba k pouzdru součástky připevňovat chladič. Tento obvod byl zvolen, protože výkonově postačuje motorům, které jsou nejčastěji používány v malých robotech.

Tab. 4.2. Tabulka znázorňující ovládání H – můstku – převzato z [1]

1A (3A) [PWM]	2A (4A)	EN1-2 (EN3-4)	Funkce
L	L	H	Motor zkratován (brzda)
L	H	H	Otáčení vlevo
H	L	H	Otáčení vpravo
H	H	H	Motor zkratován (brzda)
X	X	L	Motor odpojen

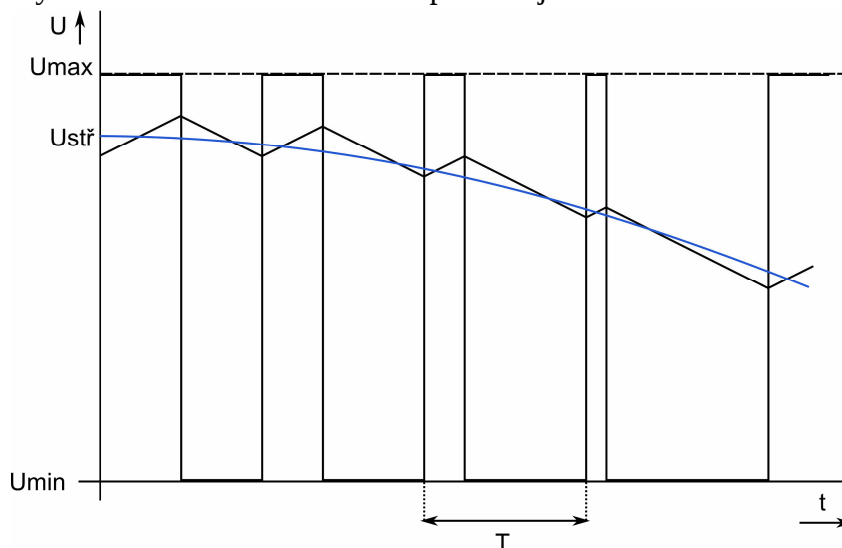
H = vysoká úroveň (1), L = nízká úroveň (0), X = libovolná úroveň A1, A2 – Vstupy pro ovládání motoru 1, A3, A4 – Vstupy pro ovládání motoru 2, EN1-2 (EN3-4) – Vstupy pro blokování.

Pro tvorbu elektronických schémat budičů motorů byl využit program ProfiCAD.

4.2.2.5 Ovládání rychlosti

Rychlost otáčení motorků je realizována pomocí pulzně – šířkové modulace (PWM) přiváděné na vstup H-můstku L293D , který je schopný zpracovat frekvenci PWM až do 5kHz, což je maximální spínací frekvence bipolárních tranzistorů.

PWM (pulse width modulation) neboli pulzně šířková modulace je změna střídavy S (změna poměru doby zapnutí k době vypnutí v periodě) obdélníkové signálu. Následkem této změny je změna střední hodnoty výstupního napětí ($U_{stř} = U_{max} * S$) a tím i změna rychlosti otáčení motoru. Princip PWM je uveden na Obr. 4.9.



Obr. 4.9. Princip pulzně šířkové modulace (PWM)

Pro správný chod motoru je potřeba určit vhodnou frekvenci PWM, proto potřebujeme nejprve zjistit časovou konstantu motoru. Použité stejnosměrné motory jsou rozměrů 15 x 10 x 12 mm. Jelikož výrobce neudává indukčnost vinutí motoru ani odpor vinutí, byli tyto hodnoty stanoveny podle téměř totožného motoru (tvarem i velikostí) jiného výrobce. Odpor vinutí $R_a = 5,3 \, \Omega$, indukčnost vinutí motoru $L_a = 1,5 \, \text{mH}$. Časová konstanta motoru se vypočte dle [6]:

$$\tau_a = \frac{La}{Ra} \quad (4.1)$$

A pro použitý motor byla stanovena na:

$$\tau_a = \frac{La}{Ra} = \frac{1,5 \cdot 10^{-3}}{5,3} = 283,02 \mu s \quad (4.2)$$

Frekvence PWM by měla být větší nebo rovna:

$$f_{PWM} \geq \frac{1}{\tau_a} \quad (4.3)$$

Tedy větší nebo rovno než:

$$f_{PWM} \geq \frac{1}{\tau_a} \geq \frac{1}{283,02 \cdot 10^{-6}} \geq 3533,44 Hz \quad (4.4)$$

Frekvence PWM by tedy měla být vyšší než anebo rovna 3,533 kHz a nižší než 5 kHz (z důvodu omezení H-můstku) z řady možných frekvencí PWM, které je možno nastavit na mikrokontroléru tomuto odpovídá jediná, a to 3,906 kHz. Dále si zvolíme maximální zvlnění střední hodnoty proudu $\Delta I_{max} = 10\%$. Pomocí vzorce z [6]:

$$\Delta I_{max} = \frac{Ud}{4 \cdot f_{PWM} \cdot L} \quad (4.5)$$

- Ud – napájecí napětí budiče. V našem případě $Ud = 5V$.

vypočteme indukčnost při $f_{PWM} = 3,906 kHz$.

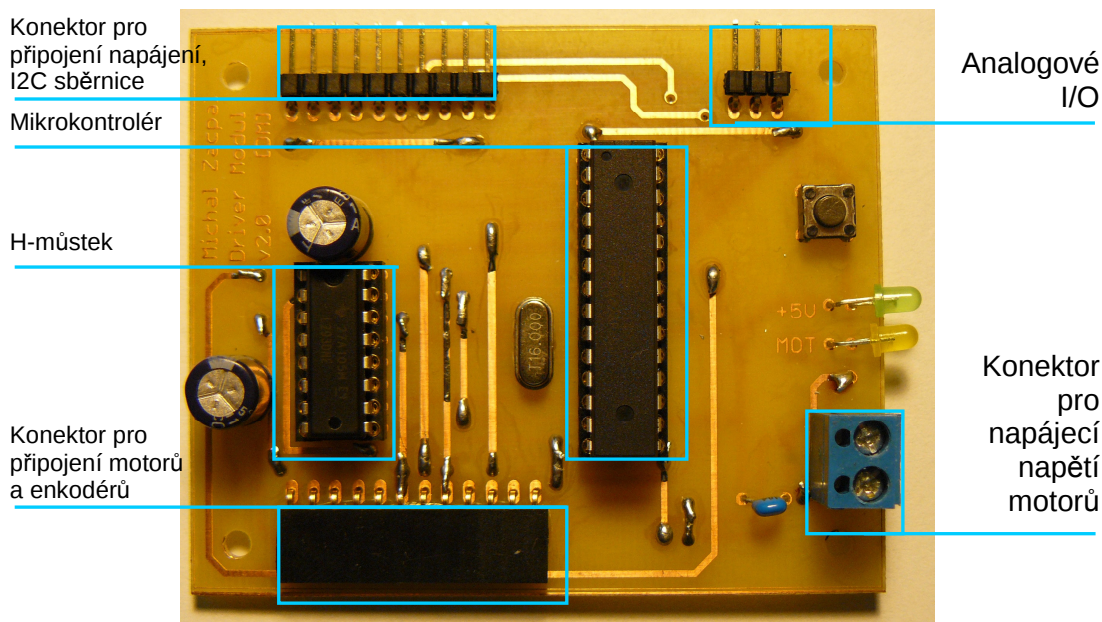
$$L = \frac{Ud}{4 \cdot f_{PWM} \cdot \Delta I_{max}} = \frac{5}{4 \cdot 3906 \cdot 0,1} = 3,2 mH \quad (4.6)$$

Tím jsme získali celkovou indukčnost, pomocí které lze vypočíst hodnotu předřadné tlumivky, kterou přiřadíme k motoru do série, abychom dosáhli požadovaného zvlnění ΔI_{max} .

$$L = La + Lp \Rightarrow Lp = L - La = 3,2 \cdot 10^{-3} - 1,5 \cdot 10^{-3} = 1,7 mH \quad (4.7)$$

Z řady dostupných tlumivek jsou této hodnotě nejbližší 1,5 mH a 1,8 mH. Po dosazení těchto hodnot do vztahu (4.5) vychází maximální zvlnění proudu $\Delta I_{max} = 10,67\%$ pro

tlumivku s indukčností 1,5mH a $\Delta I_{max} = 9,70\%$ pro tlumivku s indukčností 1,8mH. Z důvodu menšího zvlnění volíme tlumivku s indukčností 1,8 mH.

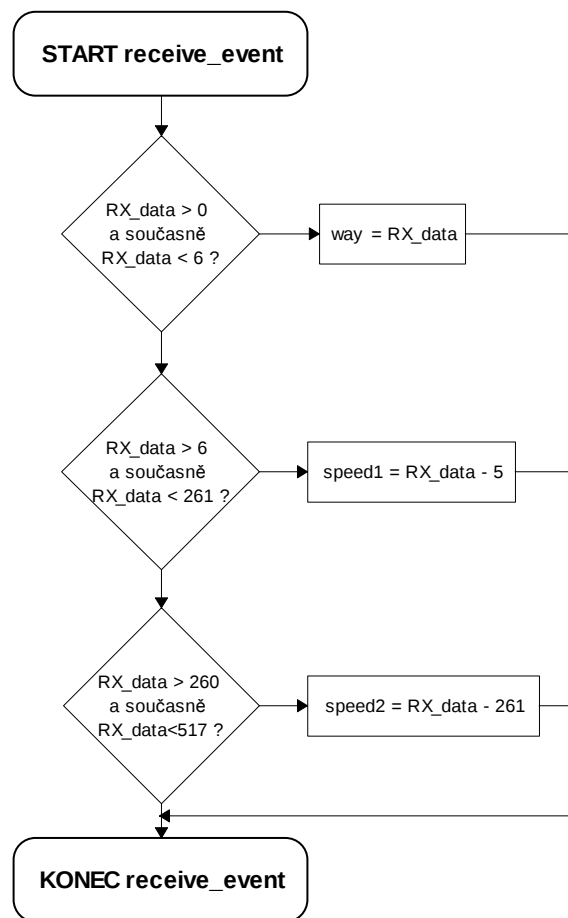


Obr. 4.10. Osazený modul pro řízení motorů

4.3 Řídicí algoritmus

Pro ovládání motorů byla vytvořena již v semestrálním projektu [1] knihovna obsahující funkce pro řízení každého motoru zvlášť a řízení celého podvozku (řízení obou motorů dohromady). Vývojový diagram znázorňující řídicí algoritmus je na Obr. 4.12. Na počátku proběhne blok inicializace, kdy dojde k nastavení adresy slave zařízení, nastavení vstupních a výstupních pinů, nastavení frekvence PWM a počáteční inicializace proměnných. Dále pokračuje hlavní smyčka algoritmu kde dochází k řízení robota za pomoci přijatých dat.

Komunikace je zajištěna za pomoci funkce `WireOnReceive()`; z knihovny Arduina `Wire.h` popsané v kapitole 2. Modul se tedy chová pouze jako zařízení typu slave pouze se schopností přijímat. Funkce `WireOnReceive()`; (`receive_event()`;) jejíž vývojový diagram je znázorněn na Obr.4.11. Slouží ke zpracování přijatých dat (`RX_data`), která ukládá do 2 proměnných s názvem `way` a `speed`. Proměnná s názvem `way` určuje směr jízdy a nabývá hodnot z intervalu `<1;5>`. 1 – vpřed, 2 – vzad, 3 – doprava, 4 – doleva, 5 – stop. Proměnné `Speed1` a `Speed2` udávají rychlost otáčení motoru 1 a motoru 2 a nabývají hodnot z intervalu `<0;255>`. Aby mikrokontrolér rozpoznal zda přijatá data znamenají směr nebo rychlost, jsou odesílána ve tvaru celočíselných hodnot v intervalu `<1;5>` pro určení směru, `<5;260>` pro určení rychlosti motoru 1 a `<261;516>` pro určení rychlosti motoru 2. Při uložení rychlosti do příslušné proměnné se pak přijatá hodnota sníží o 5 a o 261 aby odpovídala intervalu `<0;255>`.



Obr. 4.11. Vývojový diagram znázorňující průběh funkce receive_event();

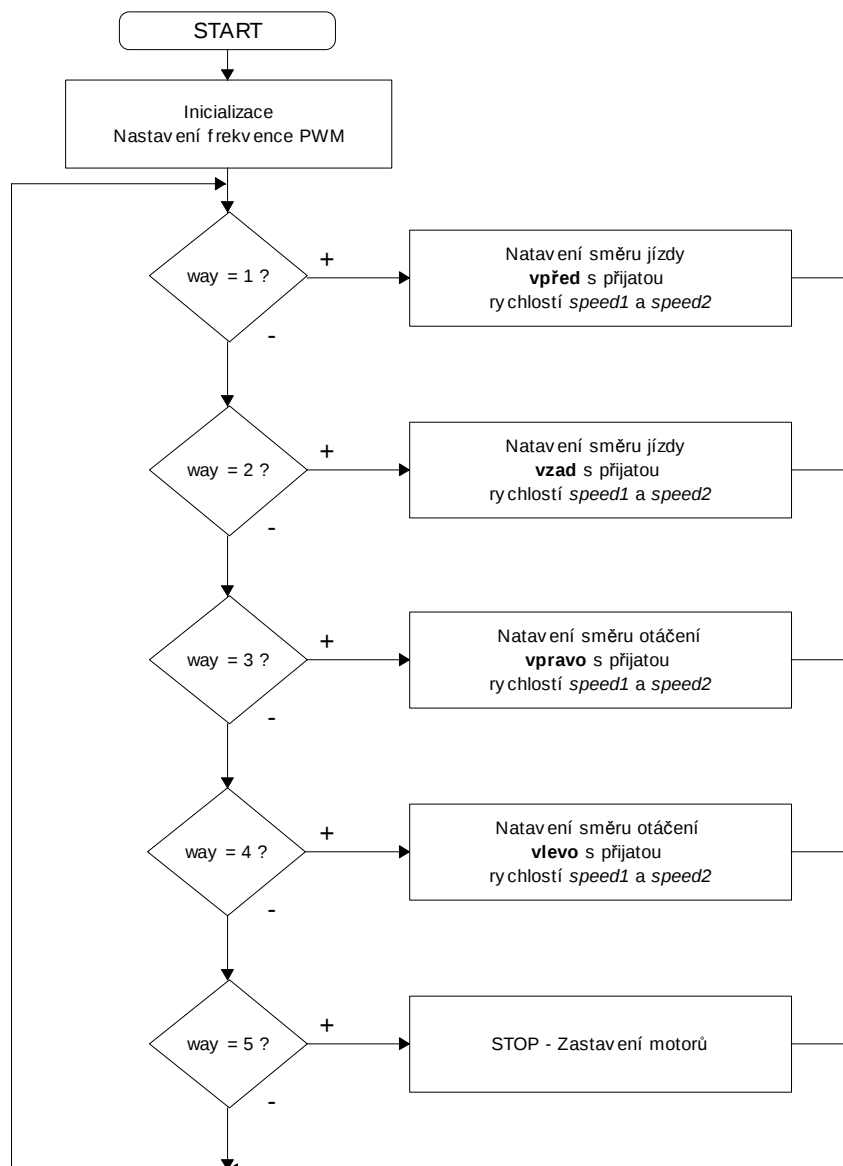
Popis proměnných:

RX_data – přijatá data

way – obsahuje informaci o směru v podobě čísla 1 – 5

Speed1 – obsahuje informaci o rychlosti v podobě čísla 0 – 255 pro motor1

Speed2 – obsahuje informaci o rychlosti v podobě čísla 0 – 255 pro motor2



Obr. 4.12. Vývojový diagram znázorňující algoritmus modulu pro řízení motorů

5 ŘÍDICÍ JEDNOTKA (ROBOT CONTROLL UNIT)

Řídicí jednotka slouží k celkovému ovládání robota, komunikaci s moduly a okolním prostředím. Tato kapitola se zabývá nejprve hardwarovou stránkou jednotky, propojením jednotlivých komponent a popisem konstrukce. Následně se zabývá softwarovou stránkou, popisem komunikace s moduly z pohledu zařízení typu master, obslužným programem pro komunikaci s uživatelem a nakonec algoritmem pro sledování čáry.

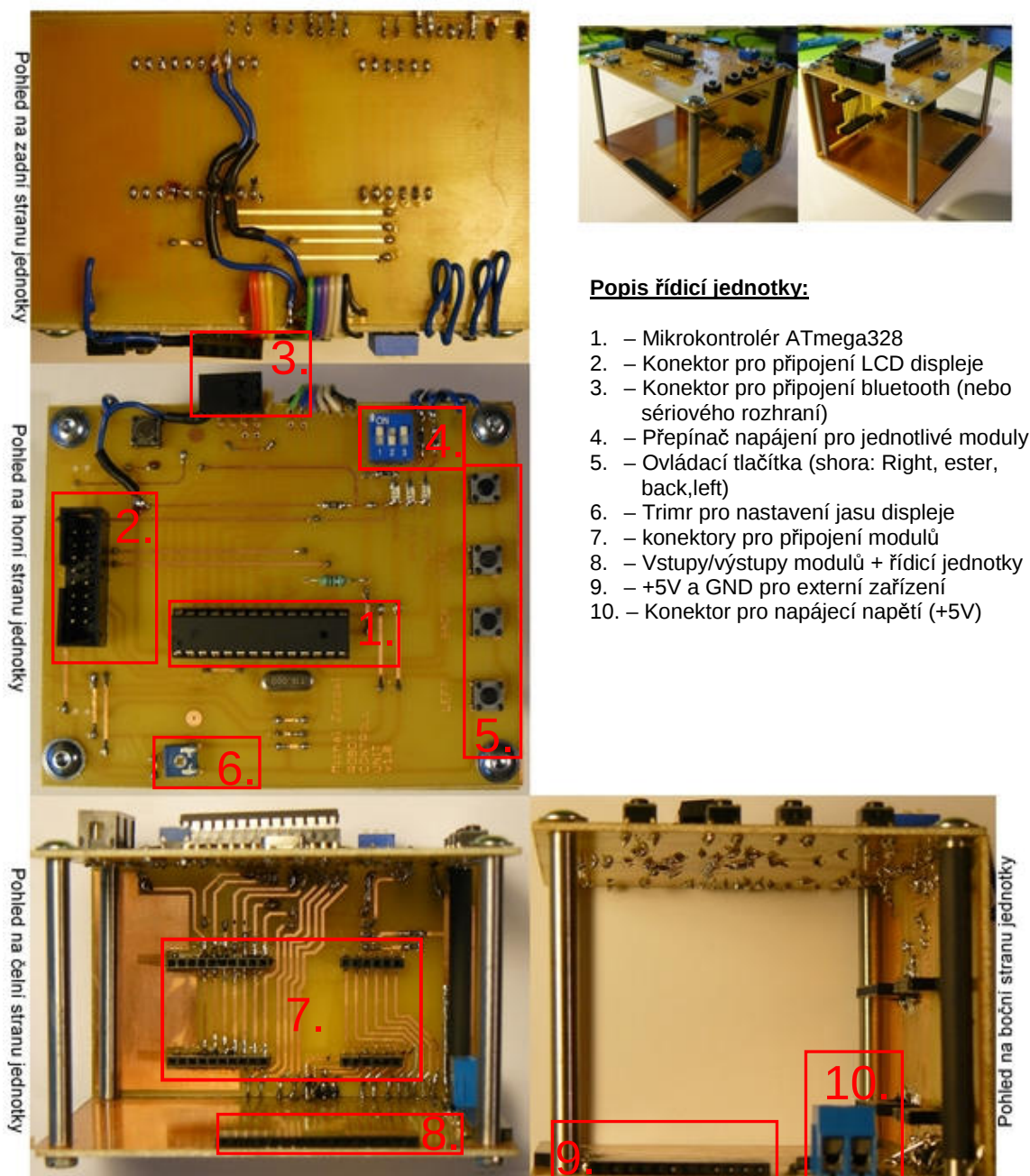
5.1 Elektronika

Schéma zapojení je uvedeno v příloze 4 a blokové schéma jednotky je na Obr.5.2. Pro komunikaci s uživatelem používá dvouřádkového šestnácti znakového displeje a čtyř tlačítek. Obsahuje ale i konektor pro připojení bluetooth modulu, a tím i schopnost komunikace přes sériové rozhraní za pomoci aplikace v PC. Moduly jsou k jednotce připojovány dle daného rozmístění konektorů viz. kapitola 2.1. Uspořádání konektoru určeného pro připojení k řídicí jednotce. Celkem je k jednotce možné připojit 2 moduly, které jsou z ní napájeny. Napájení jednotlivých modulů a jednotky lze zapínat i vypínat přes DIP spínač SW1. Toto napájecí napětí je poté indikováno příslušnými led diodami.

Na Obr.5.1 je konstrukce jednotky s popisem součástí. Veškerá elektronika se nachází na horní desce, boční deska obsahuje konektory pro připojení modulů a spodní deska obsahuje konektor pro vstupní napětí +5V a dále konektor sdružující všechny vstupy/výstupy ze zásuvných modulů a 4 analogové vstupy/výstupy řídicí jednotky. Dále také obsahuje napájecí konektory pro připojení externích zařízení. Horní deska s elektronikou je se zbytkem propojena pomocí drátových propojek a je možné ji odklopit kvůli lepšímu přístupu k modulům, kdyby to bylo zapotřebí.

Jednotka dále obsahuje konektor pro připojení dalšího zařízení schopného komunikace přes I2C a nebo sériové rozhraní. Rezistory R2 a R3 slouží jako pull up rezistory připojující I2C vodiče (SDA,SCL) ke kladnému napětí pro zajištění vysoké úrovně na nich v klidovém stavu sběrnice. ERROR_LED slouží jako indikační pokud v systému nastala nějaká chyba například nepřítomnost některého z modulů nutného pro chod zvolené funkce. Tlačítko RESET slouží k restartování řídicí jednotky i připojených modulů.

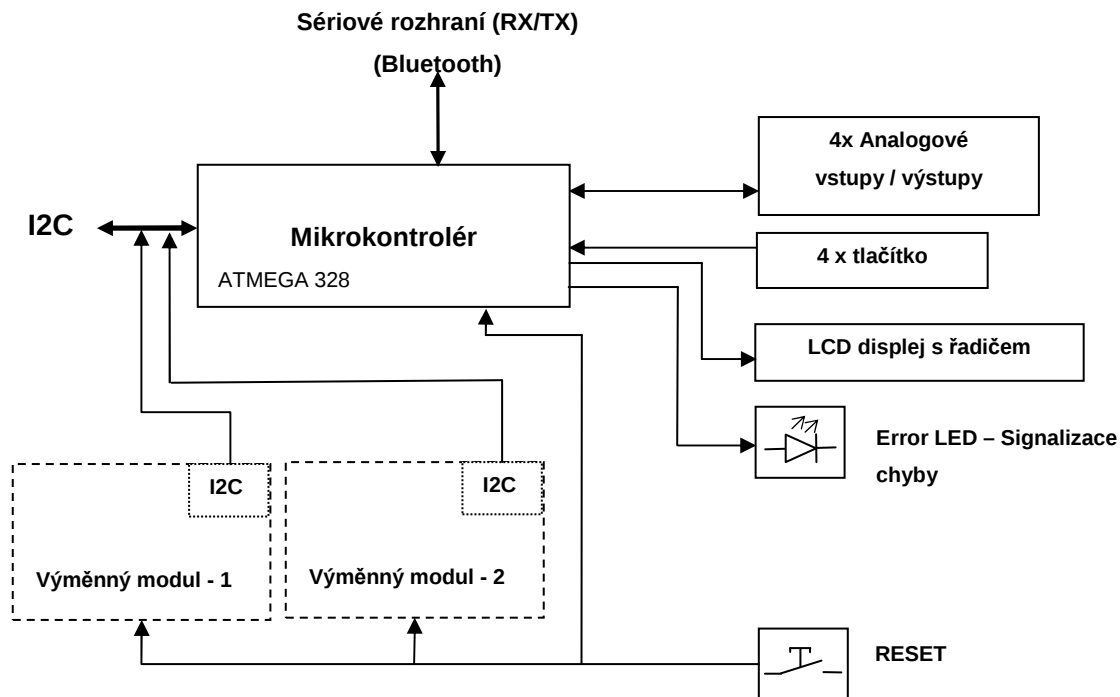
Řídicím mikrokontrolérem je ATmega328 popsaná v kapitole 2. Ta řídí chod celé jednotky. Je taktována na 16 MHz. Tlačítka jsou připojena na digitální vstupy, které mají aktivovány interní pull up rezistory zajišťující, že v případě kdy tlačítko není sepnuto, tak se na něm neobjeví náhodná úroveň. V případě stisknutí tlačítka je digitální vstup propojen na zem, což je detekováno mikrokontrolérem jako log.0.



Obr. 5.1. Konstrukce řídicí jednotky s popisem součástí

LCD displej je k jednotce připojen přes konektor a v případě potřeby je možné jej oddělat. Je umístěn v plexiskle, do kterého je vsazen. Plexisklo je k jednotce připevněno přes distanční sloupky délky 30 mm.

V Tab. 5.1. je uveden přehledný popis zapojení pinů mikrokontroléru a jejich funkce.



Obr. 5.2. Blokové schéma řídicí jednotky

Tab. 5.1. Popis zapojení pinů mikrokontroléru řídicí jednotky

Pin	Označení pinu	Funkce
PD 0	RX	sériové rozhraní
PD 1	TX	sériové rozhraní
PD 2	DB7	Ovládání displeje
PD 3	DB6	Ovládání displeje
PD 4	DB5	Ovládání displeje
PD 5	DB4	Ovládání displeje
PD 6	LEFT_BUT	Tlačítko LEFT
PD 7	BACK_BUT	Tlačítko BACK
PB 0	ENTER_BUT	Tlačítko ENTER
PB 1	RIGHT_BUT	Tlačítko RIGHT
PB 2	ERROR_LED	LED signalizující chybu
PB 3	E	Ovládání displeje
PB 4	RS	Ovládání displeje
PB 5	Nepřipojen	-
PC 0	AIO0	Analogový vstup/vystup
PC 1	AIO1	Analogový vstup/vystup
PC 2	AIO2	Analogový vstup/vystup
PC 3	AIO3	Analogový vstup/vystup
PC 4	I2C (SDA)	I2C
PC 5	I2C (SCL)	I2C
PC 6	Reset	Reset

5.1.1 Napájení řídicí jednotky

Řídicí jednotka pracuje s napájecím napětím +5V pro jeho připojení je vybavena konektorem viz Obr.5.1 odkaz 10. Toto napětí je získáno za pomoci DC-DC měniče Traco TSR 1-2450 (Obr. 5.3), který se vyznačuje velkou účinností a to až 94% bez potřeby dalšího chladiče. Jeho vstupní napětí je v rozsahu 6,5 – 36V DC a na výstupu dává stabilizovaných 5,0V DC maximální proud na výstupu je 1,0A. Současně je toto napětí využito i jako napájecí napětí pro motory. Na vstup měniče jsou připojeny dvě 9V baterie zapojené do série.



Obr. 5.3. DC-DC měnič TRACO TSR 1-2450 umístěný na robotovi

5.2 Softwarové vybavení jednotky

Jednotka ke své komunikaci s uživatelem využívá LCD displeje s řadičem. Ten je k mikrokontroléru připojen pomocí šesti pinů. K jeho ovládání je využita knihovna Arduina LiquidCrystal.h. V následujícím odstavci budou popsány pouze využité funkce v práci, jinak jich knihovna obsahuje více.

LCD.begin(); je určena k nastavení velikosti použitého displeje, je volána s parametry a to počtem řádků a sloupců displeje. (*LCD.begin(sloupce,řádky);*). Další funkcí je *LCD.clear();* - volaná bez parametru, sloužící k vymazání obsahu displeje. *LCD.print();* - volaná s parametrem, kde parametr je text, který požadujeme vypsát na displej. *LCD.setCursor();* - slouží k nastavení kurzoru na dané souřadnice, odkud poté bude vypisován text, volá se s parametry souřadnic (*LCD.setCursor(sloupec,řádek);*). *LiquidCrystal(rs,enable,d4,d5,d6,d7)* – slouží k určení pinů, které jsou připojeny k displeji. Parametry rs, enable, d4, d5, d6, d7 jsou nahrazeny číselným označením pinu mikrokontroléru, na které jsou připojeny kontakty LCD displeje (rs, enable, d4, d5, d6, d7).

5.2.1 Komunikace s moduly

Ke komunikaci jednotka využívá funkce z knihovny Wire.h a to *Wire.beginTransmission();* k zahájení komunikace, *Wire.endTransmission();* k ukončení komunikace. *Wire.write();* k odeslání dat a *Wire.requestFrom();* k vyžádání informací.

Tyto funkce byly pro zpřehlednění zdrojového kódu zahrnuty do funkce *Send.data()*; jejíž definice je následující (zelený text značí komentáře):

```
void Send_data(int device, int data1, int data2, int data3)
{
    if (data2 == 0)
    {
        Wire.beginTransmission(device);
        Wire.write(data1);
        error=Wire.endTransmission();
    }
    else
    {
        Wire.beginTransmission(device);
        Wire.write(data1);
        Wire.write(data2+5);
        Wire.write(data3+262);
        error=Wire.endTransmission();
    }
    if(error == 4)                                     //Chyba při komunikaci
    {
        chyba(true);
        lcd.clear();
        lcd.print("Chyba I2C prenos");
    }
}
```

Funkce nejprve zjistí zda bude muset odeslat druhý parametr *data2*, pokud je nulový tak ne a odešle pouze parametr *data1*, adresa zařízení je dána proměnnou *device*. Pro modul řízení motorů slouží: *data1* pro určení směru, *data2* jako určení rychlosti motoru 1 (která je vždy zvětšena o 5 z důvodu rozlišení na přijímací straně, z toho plyne, že nikdy nemůže být nulová), *data3* jako určení rychlosti pro motor 2 (zvětšená o 262 z důvodu rozlišení na přijímací straně). Uživatel pro *data2* i *data3* zadává hodnotu v rozsahu <0-255> funkce sama zvětší interval. Pro modul sledování čáry slouží *data1* jako odeslání toho co od modulu požadujeme a *data2* i *data3* se zadají jako nulové. Nakonec pokud při komunikaci nastane nějaká chyba, funkce o tom uživatele upozorní zprávou na displeji. Při komunikaci s modulem sledování čáry je nejprve nutné se dotázat a poté přijmout odpověď. Za pomoci funkce *Wire.read()*; vše ukazuje následující část zdrojového kódu.

```
Send_data(2, 'A', 0);
delay(1);
Wire.requestFrom(2, 1);
if (Wire.available())
{
    RXI2C = Wire.read();
}
// časová prodleva 1ms
// Požádání o data
// Pokud je příchozí komunikace
// Přečtení dat a uložení do
// proměnné RXI2C
```

5.2.1.1 Přítomnost modulů

Ke zjištění přítomnosti modulů byla vytvořena funkce *scan_I2Cbus()*; Ta proskenuje I2C sběrnici a zjistí počet připojených zařízení. Rozpoznání modulu pak závisí na znalosti jeho adresy. Modul pro snímání čáry má přiřazenou adresu: 2 a modul pro řízení motorů adresu: 3. Funkce nejprve zahájí přenos pomocí funkce

`Wire.beginTransmission()`; a poté jej ukončí pomocí `Wire.endTransmission()`; a přijme byt o úspěšnosti přenosu, který uloží do proměnné `err`. Pokud je byt roven 0 přenos proběhl úspěšně a na dané adrese se nachází zařízení a je schopno komunikovat, pokud je aktuální adresa rovna 2 nastaví proměnnou `LFM` (modul snímání čáry) typu bool jako true (značící přítomnost modulu), pokud je aktuální adresa rovna 3 nastaví proměnnou `DM` (Modul řízení motorů), také typu bool, jako true. Pokud je proměnná `err` ale rovna 4 (chyba přenosu) znamená to, že na dané adrese není nic připojeno, pokud se jedná o adresy modulů proměnné `LFM` nebo `DM` se nastaví jako false (značící nepřítomnost modulu). Celý cyklus se provede 128-krát což je maximální počet adres. Nakonec na displej vypíše počet připojených zařízení. Vývojový diagram je na Obr.5.6.

Pro výpis na displej, který z modulů je přítomen, slouží funkce `show_modul()`, která provede scan sběrnice pomocí funkce `scan_I2Cbus()` a výsledek vypíše na displej. Její definice je následující(zeleně jsou uvedeny komentáře):

```
void show_modul()
{
    scan_I2Cbus();           // Proveďte scan sběrnice
    delay(500);              // Počka 500 ms
    lcd.clear();             // Vycisti displej
    lcd.write("Připojene moduly:"); // vypis na displej 1. radek
    lcd.setCursor(0,1);      // nastavení kursoru na 2.radek
    lcd.write("LFM: ");lcd.print(LFM); // Vypis na displej 2.radek
    lcd.write(" DM: ");lcd.print(DM);  // Vypis na displej 2.radek
}
```

5.2.2 Obsluhující algoritmus řídicí jednotky

Obsluhujícím algoritmem je myšlen algoritmus starající se o chod jednotky. Tedy hlavní smyčku, která snímá vstupy s připojenými tlačítky, ovládá displej a zásuvné moduly. Tlačítka slouží k procházení menu a výběru akce. Je možné spustit scan připojených zařízení do I2C sběrnice, sledování čáry, aktivovat sériové rozhraní, zobrazit připojené moduly a potvrdit chybu. Vývojový diagram je na Obr.5.7.

Program postupně zjišťuje zda některé je tlačítko stisknuté či nikoliv, pokud je, potom provede příslušnou akci. Tlačítka označená jako left a right slouží pro pohyb v menu do stran. Tlačítko označené jako enter slouží ke spuštění vybrané akce. Tlačítko back slouží k ukončení akce a návratu do menu. Pro výpis menu na displej slouží funkce `show_menu()` jejíž vývojový diagram je na Obr.5.8. Tlačítka left a right při stisknutí inkrementují nebo dekrementují proměnnou `menu` určující, kde v menu se právě nacházíme a na jejímž základě je potom položka menu vypsána na displej. Při aktivaci příslušné akce tlačítkem enter dojde k jejímu spuštění a signalizaci, o což se stará funkce `show_menu2()` jejíž vývojový diagram je na Obr.5.9. Proměnná `menuH` je typu bool a slouží k rozlišení, zda se program nachází právě v menu nebo je prováděna nějaká akce.

Pokud v systému dojde k nějaké chybě, například chceme sledovat čáru a chybí nám některý z modulů je aktivována funkce `chyba(err)`, kde parametr `err` je typu bool.

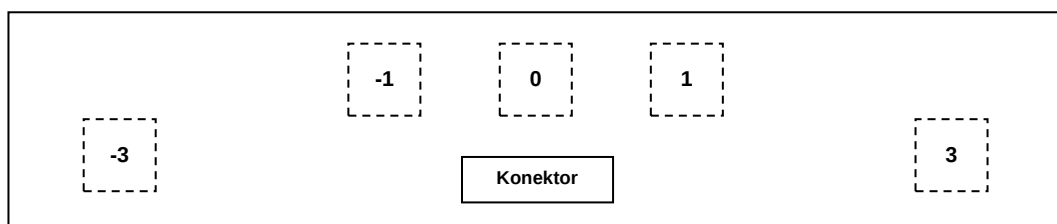
Pokud je *err = true*, pak je chyba aktivní, dojde k signalizaci chyby rozsvícením led diody ERROR_LED a zastavení motorů, pokud jsou v pohybu. Při potvrzení chyby je funkce volána s parametrem *err = false* a dojde ke zhasnutí ERROR_LED a uživatel musí chybu vzniklou v systému spravit.

5.2.3 Algoritmus sledování čáry

Funkce pro sledování čáry se nazývá *follow()*; a je aktivována z menu. Její vývojový diagram je na Obr 5.5. Nejprve dojde k ověření přítomnosti modulů pomocí proměnných udávajících jejich přítomnost. Pokud některý z modulů chybí je aktivována funkce *chyba()* s parametrem *true* (je rozsvícena LED dioda signalizující chybu a zastavení motorů pokud běží) a uživatel musí potvrdit chybu v menu. Pokud jsou oba moduly přítomny, je následně vyžádána informace o poloze čáry, ta je přijata a na jejím základě je vypočítána nebo určena rychlost motorů. Cílem je udržet prostřední čidlo nad čarou.

Pro ověření funkčnosti celého systému byly do řídicí jednotky implementovány a ozkoušeny dva druhy algoritmů pro sledování čáry. První algoritmus byl založený na stavovém automatu, kdy pro každé čidlo je předem definovaná rychlost a směr motorů. Dle polohy čáry se za pomoci switch/case přepíná rychlost motorů. Pokud se čára nachází uprostřed robot jede rovně, pokud pod pravým čidlem začne pomalu zatáčet doleva, pokud pod levým čidlem tak naopak. Pokud se čára nachází pod pravým krajním čidlem začne prudce zatáčet doleva (levý motor zastaven), pro levé krajní čidlo je tomu zase naopak. Tento algoritmus je jednoduchý na implementaci, ale za to je složitější na nastavení a následné odladění jednotlivých rychlostí motorů, pokud požadujeme aby robot plynule sledoval čáru.

V dalším algoritmu je rychlost vypočítávána v průběhu sledování na základě aktuální a předchozí polohy čáry. Algoritmus je založený na obdobě PID regulátoru. Jednotlivým čidlům jsou přiřazeny různé koeficienty viz Obr.5.4. a cílem je, aby se čára opět nacházela pod prostředním s hodnotou 0.



Obr. 5.4. Koeficienty přiřazené jednotlivým čidlům snímače čáry

Výpočet rychlosti pro motory probíhá následovně:

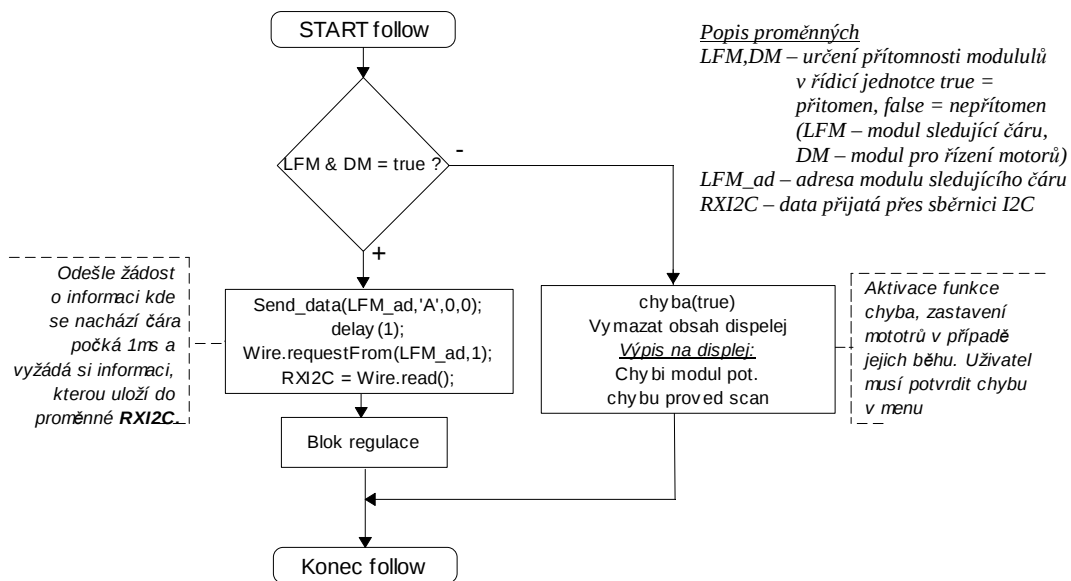
```
dif = err-prev_err1;
prev_err=err;
sum_err +=err;
pid_speed = kP * err + kD*dif + kI * sum_e;
```

proměnná *err1* je dána koeficientem čidla, pod kterým se právě nachází čára, *prev_err* je hodnota zjištěná v přechozím kroku, *dif* je rozdílem těchto proměnných a *sum_err* je akumulovaná odchylka. Proměnné *kP*, *kD*, *kI* jsou koeficienty, které byly nastavovány experimentálně. Proměnná *pid_speed* je vypočtená hodnota rychlosti, která se před posláním na motory normalizuje, aby nepřesahovala z rozsahu hodnot <-255;255> a poté je poslána na motory, což ukazuje následující část kódu(zelně jsou uvedeny komentáře):

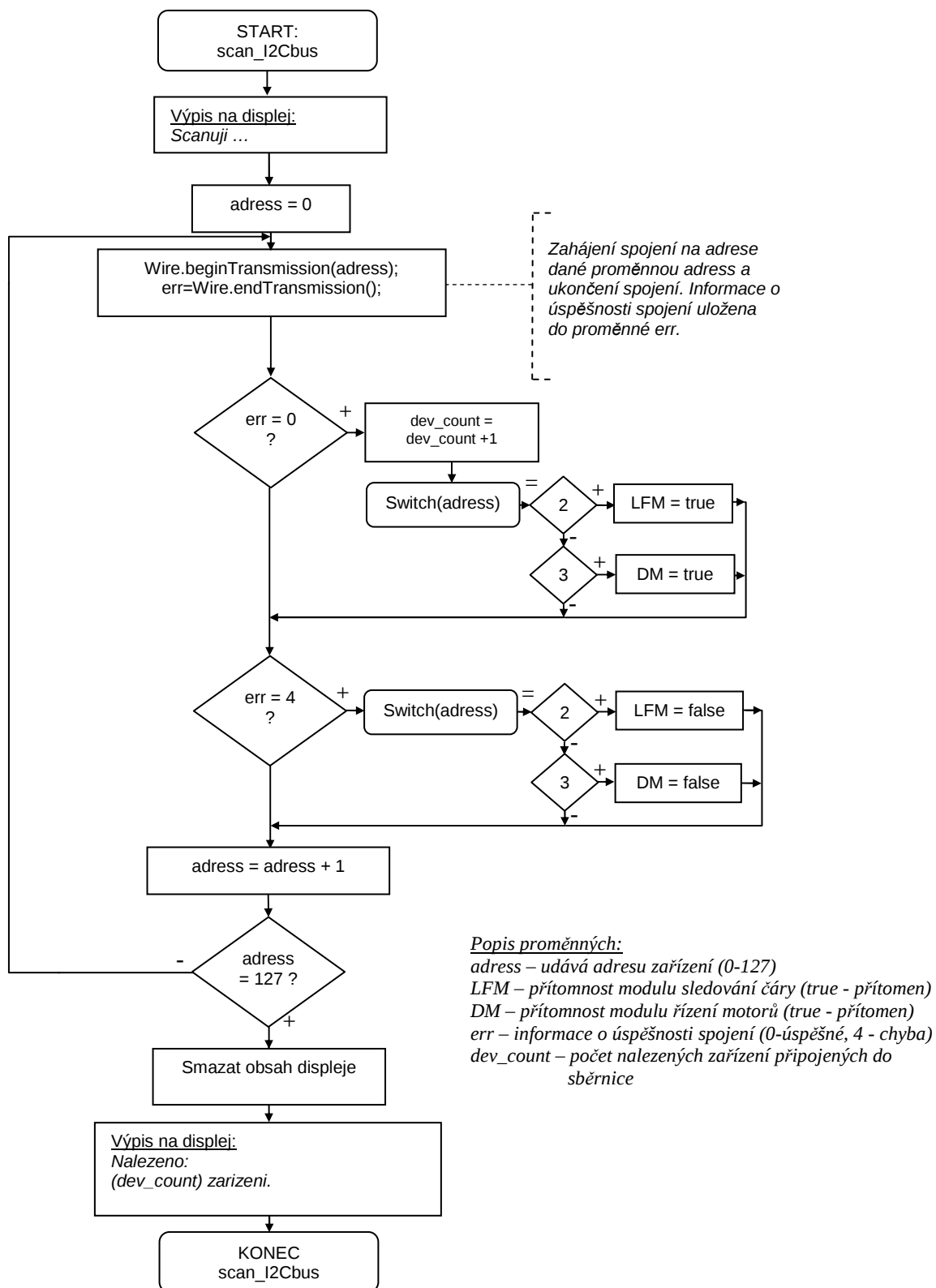
```
if (pid_speed > 0)           // pokud je větší musím zpomalit pravý motor
{
    Send_data(DM_ad,1,255,(255-pid_speed));
}
if (pid_speed < 0)           // pokud je menší musím zpomalit levý motor
{
    Send_data(DM_ad,1,(255+pid_speed),255);
}
```

Pokud při prvním spuštění robot není na čáře rozjede se rovně.

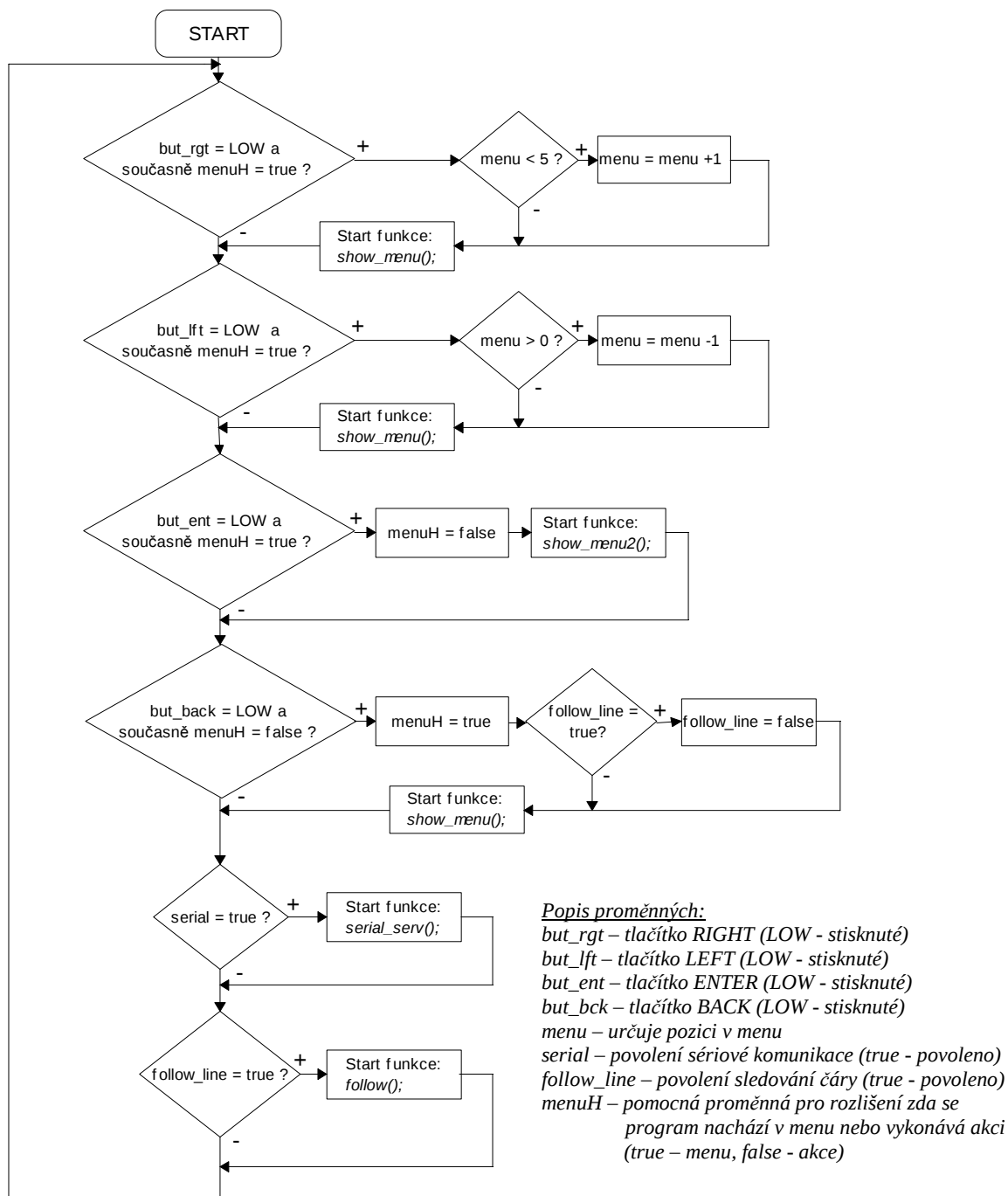
S druhým algoritmem bylo sledování čáry plynulejší. Testování bylo prováděno na černé čáře o šířce cca 15 mm ta se používá na většině robotických soutěží. Ovšem snímač čáry se se svým rozložením čidel hodí spíše pro sledování čáry širší. Pro takto úzkou čáru je jeho rozlišení velké a pohyb při sledování není úplně plynulý.



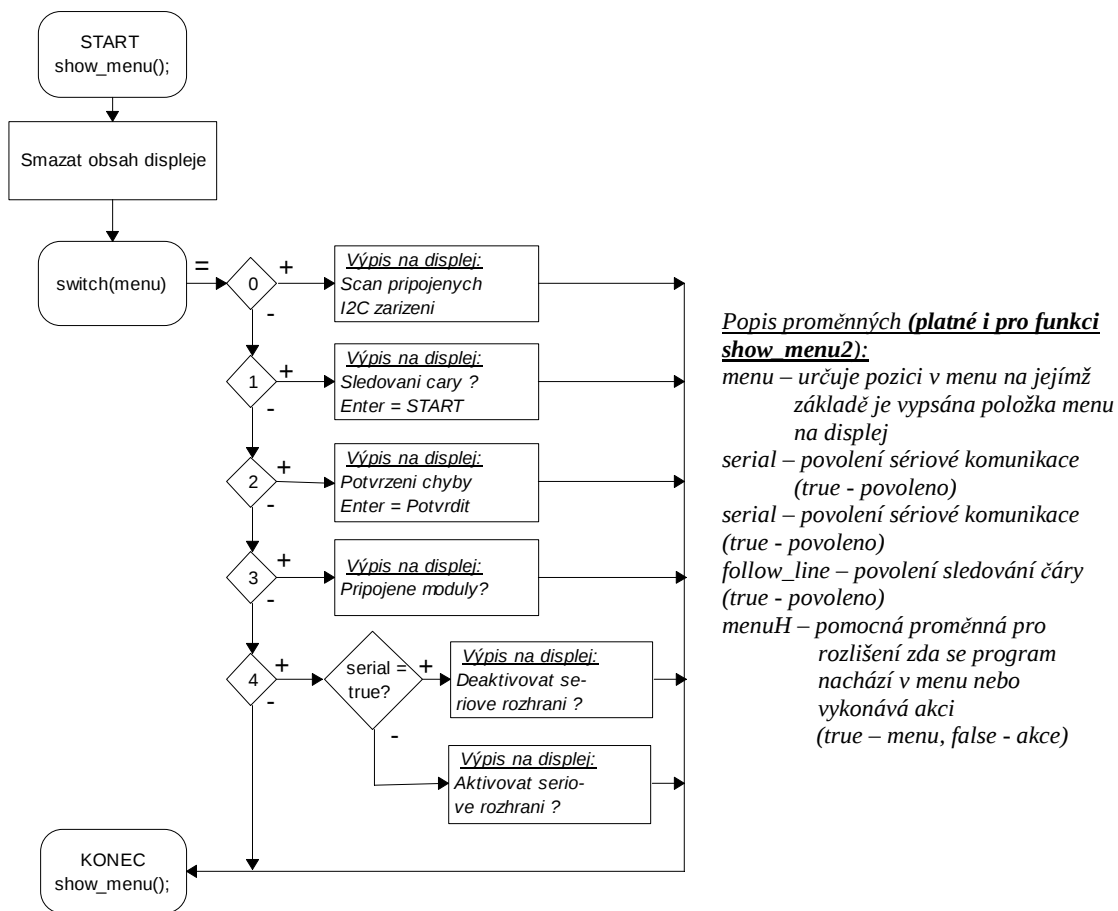
Obr. 5.5. Vývojový diagram funkce pro sledování čáry



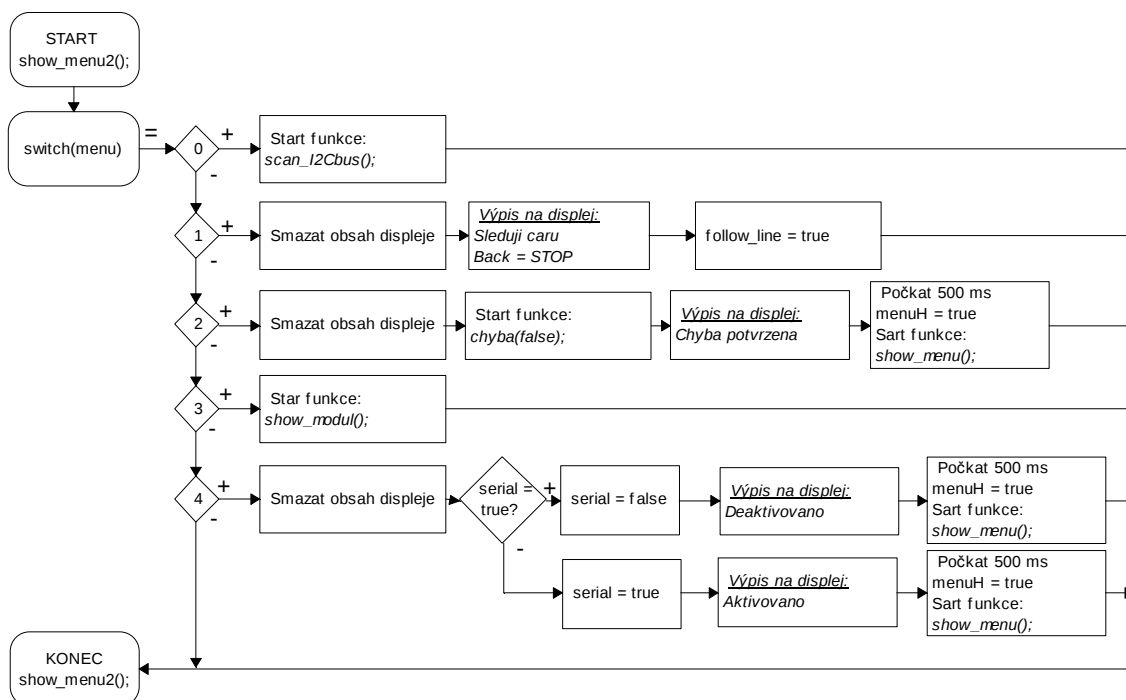
Obr. 5.6. Vývojový diagram funkce scan_I2Cbus



Obr. 5.7. Vývojový diagram hlavní smyčky



Obr. 5.8. Vývojový diagram funkce `show_menu()`, která slouží k výpisu menu na displej



Obr. 5.9. Vývojový diagram funkce `show_menu2()`, která provádí aktivaci funkcí řídicí jednotky

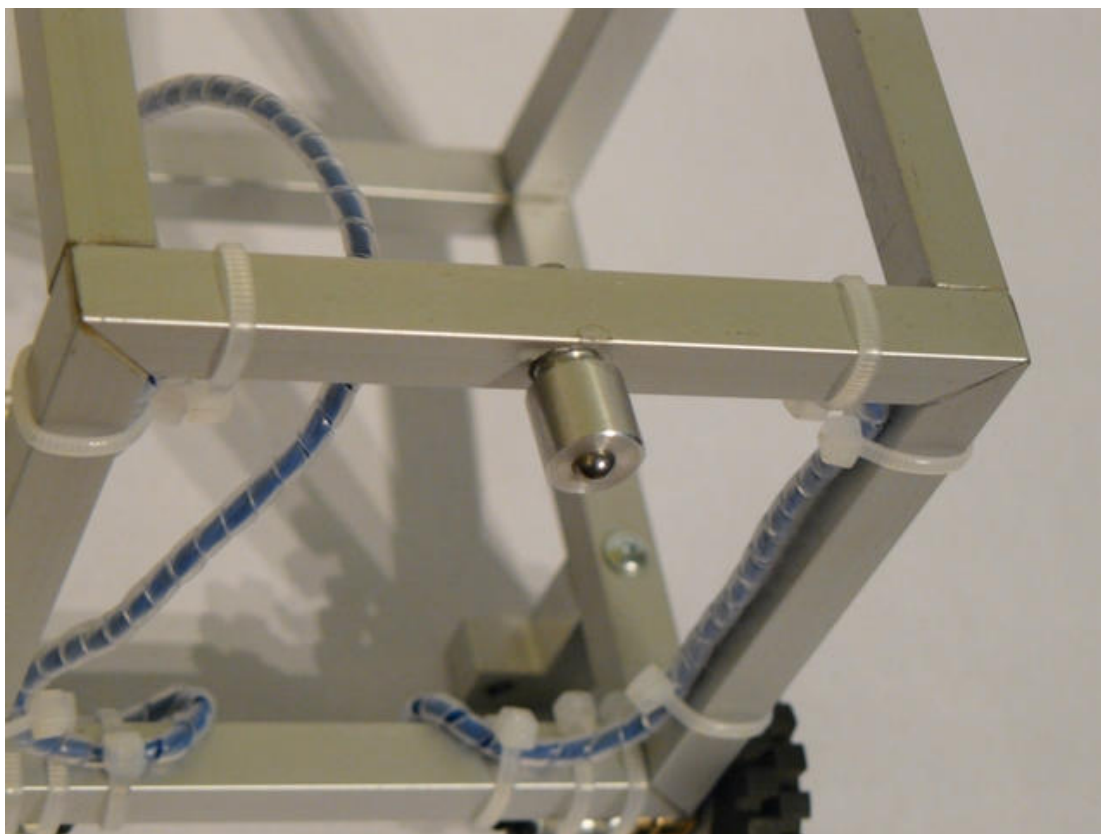
6 KONSTRUKCE ROBOTA

Aby mohla být řídicí jednotka otestována, byla k tomuto účelu vytvořena konstrukce robota, ve vložené příloze 6 na papíru formátu A3 jsou její technické nákresy.

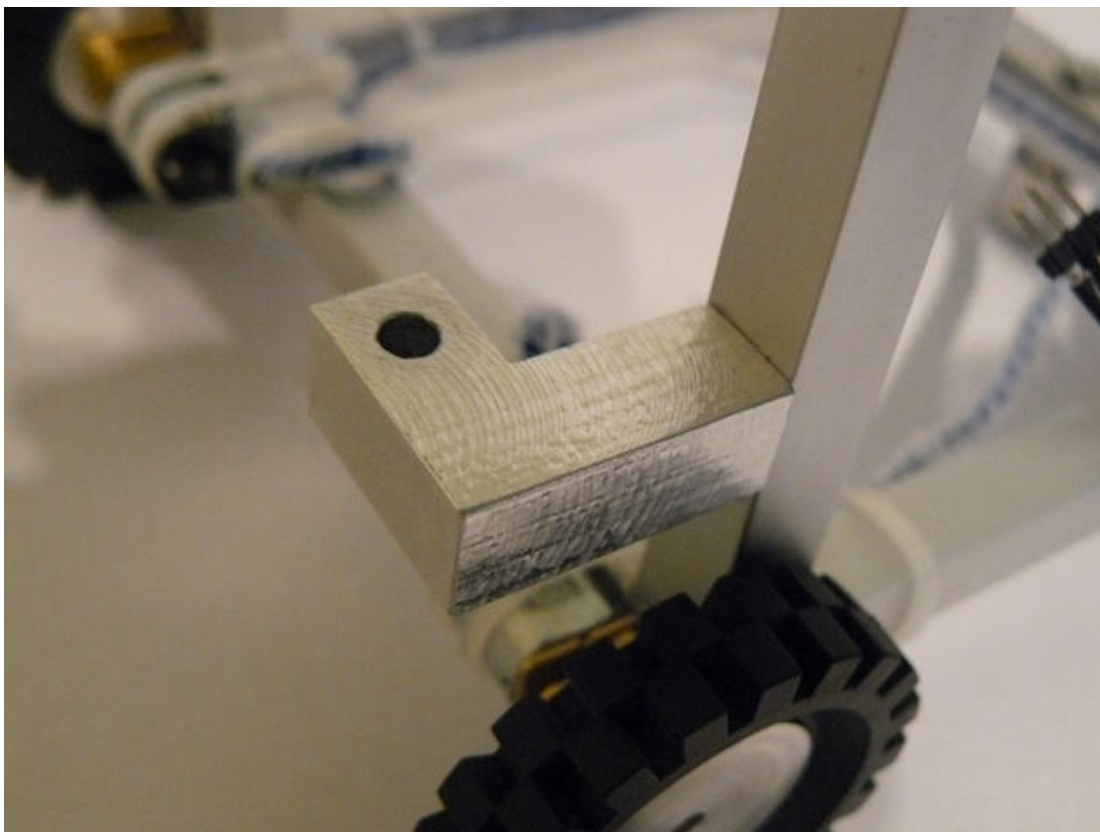
Kostra je vytvořena z hliníkových profilů čtvercový jekl 10 x 10 mm. Tyto profily byly nařezány na patřičný rozměr a slepeny lepidlem na kov. Profily pak tvoří konstrukci tvaru krychle o délce hrany 110 mm, kde jedna hrana je vynechána pro lepší přístup k vnitřním komponentům robota.

Motory jsou přilepeny lepidlem a pojištěny stahovacími páskami – tato varianta upevnění se ukázala jako zcela dostačující pro robota této velikosti. Nad motory se nachází držáky snímače čáry, jehož vzdálenost od podložky je nastavitelná pomocí matek. Podpurná nožka (Obr. 6.1.) je tvořena dvěma ložiskovými kuličkami (dvěma pro snížení třecích ploch) umístěnými ve válečku a přitažených šroubkem. Váleček má na jedné straně zúžený otvor pro zabránění vypadnutí kuliček. Řídicí jednotka je s konstrukcí robota spojena přes plexisklo upevněné pomocí 4 šroubků.

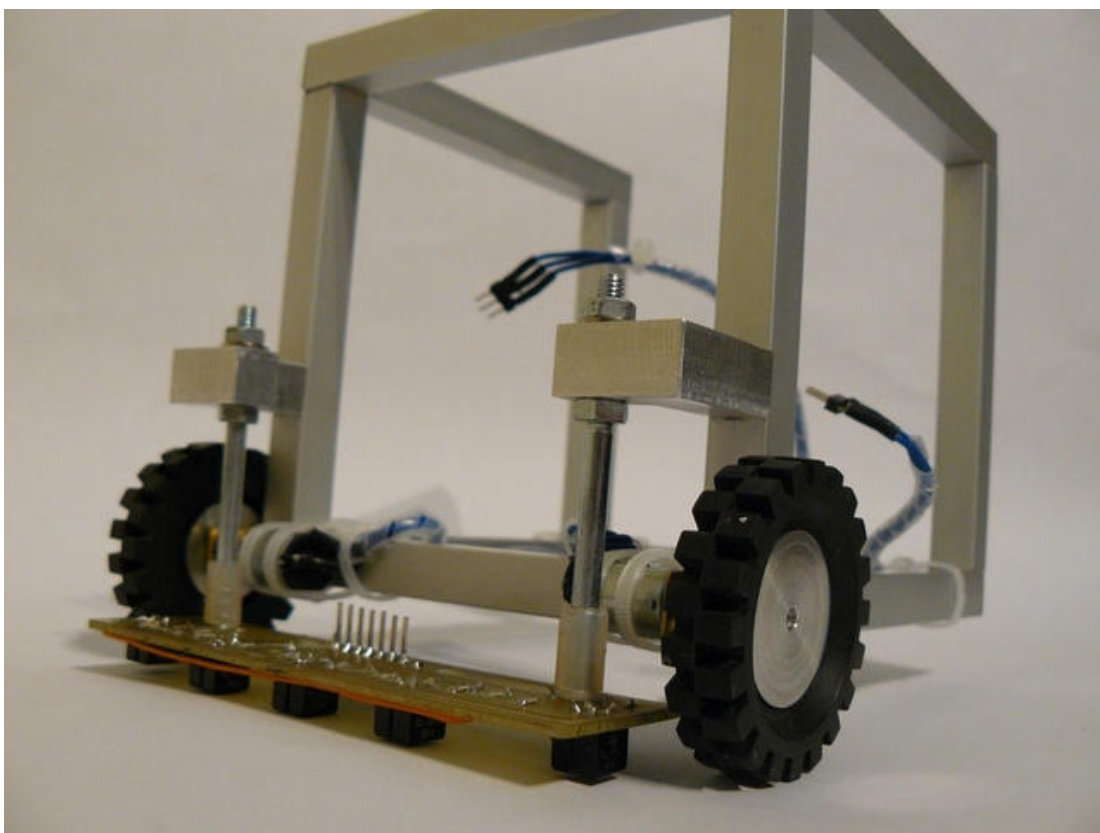
Vše ukazují přiložené fotografie na následujících obrázcích. Více jich je v příloze 1.



Obr. 6.1. Detail podpurné nožky



Obr. 6.2. Detail držáku snímače čáry



Obr. 6.3. Kostra robota se snímačem čáry - celkový pohled

7 ZÁVĚR

Výsledkem práce jsou dva moduly, a to modul pro sledování čáry a modul pro řízení motorů, který byl již vytvořen v rámci semestrálního projektu a nyní byl pouze předělán do podoby zásuvného modulu připojitelného k řídicí jednotce. Dalším výstupem práce je řídicí jednotka schopná připojení dvou zásuvných modulů. Dále byla vytvořena zcela nová konstrukce robota pro otestování celého systému.

V průběhu návrhu řídicího systému byl kladen důraz na další možnou rozšiřitelnost o jiné moduly než o dva výše zmíněné. Bylo tedy vytvořeno rozvržení konektorů, kterému musí odpovídat zásuvný modul připojitelný k této řídicí jednotce (ovšem pomocí drátových propojek je možné připojit jakýkoliv jiný). Řídicí jednotka je tedy modulární systém schopný rozšiřitelnosti pomocí modulů. To jí činí použitelnou nejen v robotech pro sledování čáry, ale i jinde. A věřím, že by se také mohla uplatnit v nějakém druhu laboratorní úlohy (např. pro výuku I2C sběrnice), kde by si studenti mohli tvořit vlastní zásuvné moduly. Momentálně je jednotku možno rozšířit o dva moduly, tento počet lze ale změnit, a to výměnou spodní a boční desky řídicí jednotky. Zde nastal problém při návrhu tyto dvě desky měly mít prohozené vrstvy pro lepší pájení konektorů bohužel desky byly v době odhalení problému již vyrobeny a kontakty konektorů, které se nepodařilo dobře zapájet musely být nahrazeny drátovými propojkami.

Celý systém komunikuje přes I2C sběrnici. S uživatelem pak pomocí LCD displeje, čtyř tlačítek a LED diody signalizující chybu. Díky I2C sběrnici je systém schopný zjistit kolik je k ní připojeno zařízení a identifikovat zda se jedná o některý z modulů. K tomu slouží funkce `scanI2C_bus()`; která identifikuje modul na základě znalosti jeho adresy, pokud by se v systému nacházelo jiné zařízení se stejnou adresou jako některý z modulů, systém si bude myslet že je připojený daný modul. Možné vylepšení bych tedy viděl v této funkci, kdy by se řídicí jednotka ještě dotázala zda se opravdu jedná o daný modul.

S tvorbou mechanické konstrukce nebyly žádné problémy a za pomoc s tvorbou panelu z plexiskla skla bych chtěl poděkovat vedoucímu práce Ing. Lukáši Kopečnému, Ph.D. Ostatní konstrukci a mechanické práce jsem si prováděl sám.

Nakonec byly pro ověření funkčnosti implementovány dva algoritmy sledující čáru. Jeden založený na předem známých rychlostech odpovídajícím poloze čidel a jeden vypočítávající rychlost pro motory na základě znalosti aktuální a předchozí polohy čáry. Druhý algoritmus se jevil jako lepší, nastavení konstant bylo časově méně náročnější než u prvního a i pohyb po čáře byl méně trhaný, ale pořád to nebylo to pravé. Důvodem neplynulé jízdy robota je rozlišení snímače čáry. Jednotlivá čidla (reflexní optočleny) jsou rozmístěna příliš daleko od sebe. Pro pokrytí celé šířky robota (110 mm) bych využil zbylých šest volných vstupů modulu pro sledování čáry a počet optočlenů tak zvýšil na jedenáct. V tomto bych viděl další vylepšení robota. Nutno podotknout, že na většině robotických soutěží se používá šířka čáry cca 15 mm (na této šířce byl robot také testován), pro kterou by toto rozlišení bylo dostačující. Pokud

bychom chtěli zachovat počet optočlenů stačilo by dát levý a pravý optočlen blíže středu a sledování by bylo opět plynulejší. Ovšem za cenu pokrytí menší oblasti. Pokud jsem robota nechal sledovat čáru o šířce cca 30 mm plynulost jeho jízdy se logicky zlepšila. Z toho tedy vyplývá, že daný snímač čáry je vhodný ke sledování čáry o šířce cca 30mm. Ani jeden z algoritmů neřeší situaci, kdy se čáry kříží. Pokud by k této situaci došlo, nelze říci jakým směrem by se robot pohyboval. Jeho trajektorie by záležela na tom, které čidlo detekuje křížící čáru jako první. V případě že je čára přerušena bude robot pokračovat v posledním nastaveném směru.

Jako další vylepšení řídicí jednotky by bylo dobré ji opatřit sledováním stavu nabití baterie a indikací jejího vybití.

Na závěr bych ještě doplnil, že robota je možné po aktivaci sériového rozhraní ovládat pomocí aplikace, vytvořené v rámci semestrálního projektu, z PC přes bluetooth rozhraní.

LITERATURA

- [1] ZACPAL, Michal. *Malý robot pro sledování čáry: Semestrální práce*. Brno, 2013. Semestrální práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Vedoucí práce Ing. Lukáš Kopečný, Ph.D.
- [2] Jednočipový počítač. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, [2006], aktualizováno 10.3.2013 [cit. 2013-03-14]. Dostupné z: http://cs.wikipedia.org/wiki/Jedno%C4%8Dipov%C3%BD_po%C4%8D%C3%ADta%C4%8D
- [3] Pulse-width modulation. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, [2002], last modified 25 March 2013 [cit. 2013-03-27]. Dostupné z: http://en.wikipedia.org/wiki/Pulse-width_modulation
- [4] RISC. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, [2005], aktualizováno 17.3.2013 [cit. 2013-04-02]. Dostupné z: <http://cs.wikipedia.org/wiki/RISC>
- [5] I2C. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, [2002], last modified 24 April 2013 [cit. 2013-04-25]. Dostupné z: <http://en.wikipedia.org/wiki/I%C2%B2C>
- [6] SKALICKÝ, J. *Elektrické regulované pohony*. Brno: FEKT VUT v Brně, 2007, 123s.
- [7] VISHAY SEMICONDUCTORS. *Reflective Optical Sensor with Transistor Output: CNY70*. 2012, 11 s. Dostupné z: <http://www.vishay.com/docs/83751/cny70.pdf>
- [8] OLEJÁR, Martin. Stručný popis sběrnice I2C a její praktické využití k připojení externí eeprom 24LC256 k mikrokontroléru PIC16F877. *HW.cz | Vše o elektronice a programování* [online]. 20.5.2000 [cit. 2013-05-19]. Dostupné z: <http://www.hw.cz/navrh-obvodu/strucny-popis-sbernice-i2c-a-jeji-prakticke-vyuziti-k-pripojeni-externi-eeprom-24lc256>
- [9] I2C - o co jde a jak pracuje. *DH servis Vývoj a výroba elektroniky na zakázku* [online]. © 2002 - 2013 [cit. 2013-05-19]. Dostupné z: <http://www.dhservis.cz/iic.htm>
- [10] VRAJESH, Davé a Phil LUU. NXP SEMICONDUCTORS, LeCroy Corporation. *How to debug I2C devices using an oscilloscope*. 30.7.2008, 90 s. Dostupné z: <http://ics.nxp.com/literature/presentations/interface/pdf/debug.i2c.oscilloscope.pdf>

SEZNAM ZKRATEK

ACK	Acknowledged – <i>bit značící úspěšné přijetí dat</i>
DC	Direct current – <i>stejnoseměrný proud</i>
DM	Driver modul – <i>Modul řízení motorů</i>
EEPROM	Electrically erasable programable read-only memory – <i>elektronicky mazatelná paměť</i>
I2C	Inter-Integrated Circuit – <i>sběrnice</i>
LCD	Liquid crystal display - <i>displej z tekutých krystalů</i>
LFM	Line follow modul – <i>modul sledování čáry</i>
NACK	No acknowledge – <i>bit ukončující komunikaci</i>
PWM	Pulse width modulation – <i>pulzně šířková modulace</i>
RCU	Robot controll unit - <i>řídící jednotka robota</i>
RX	Receiver - <i>přijímač</i>
SRAM	Static random access memory – <i>statická paměť</i>
TWI	Two wire interface – <i>ekvivalent I2C sběrnice od Firmy Atmel</i>
TX	Transceiver - <i>vysílač</i>

SEZNAM PŘÍLOH

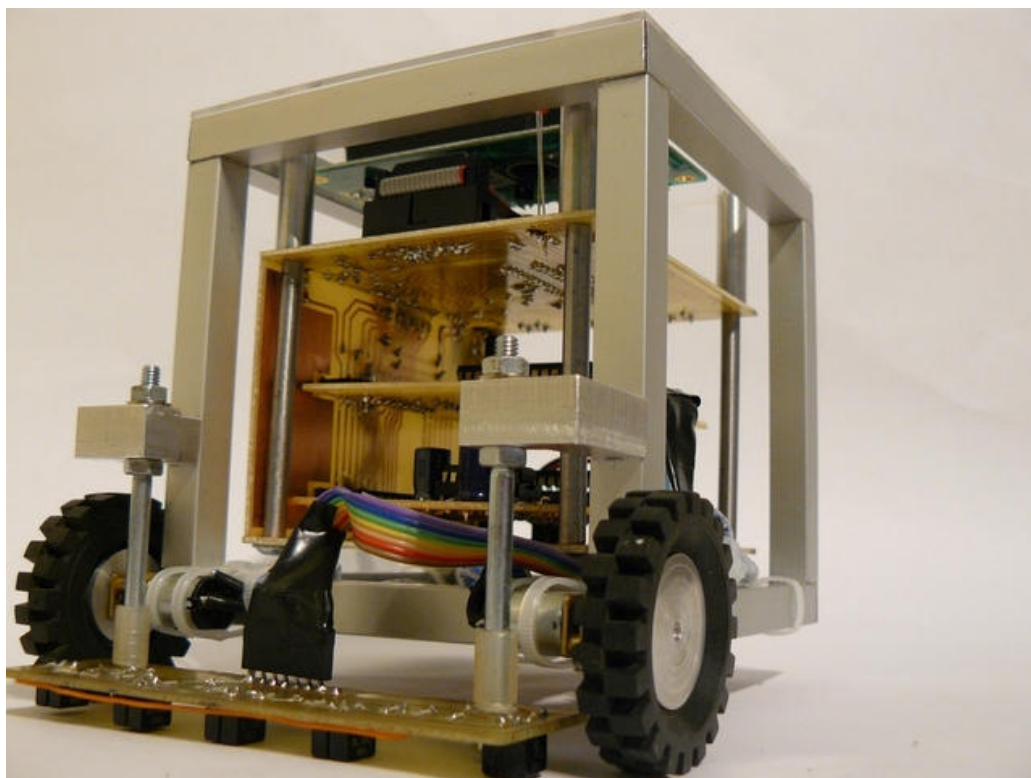
Elektrická schémata a technický výkres jsou vloženy na volných listech.

Příloha 1. Obrazová fotodokumentace	
Příloha 2. Elektrické schéma zapojení modulu pro sledování čáry	(2 listy)
Příloha 3. Elektrické schéma zapojení modulu pro řízení motorů	(1 list)
Příloha 4. Elektrické schéma zapojení řídicí jednotky	(1 list)
Příloha 5. Elektrické schéma zapojení robota	(1 list)
Příloha 6. Technický výkres konstrukce robota	(1 list formátu A3)
Příloha 7. CD	

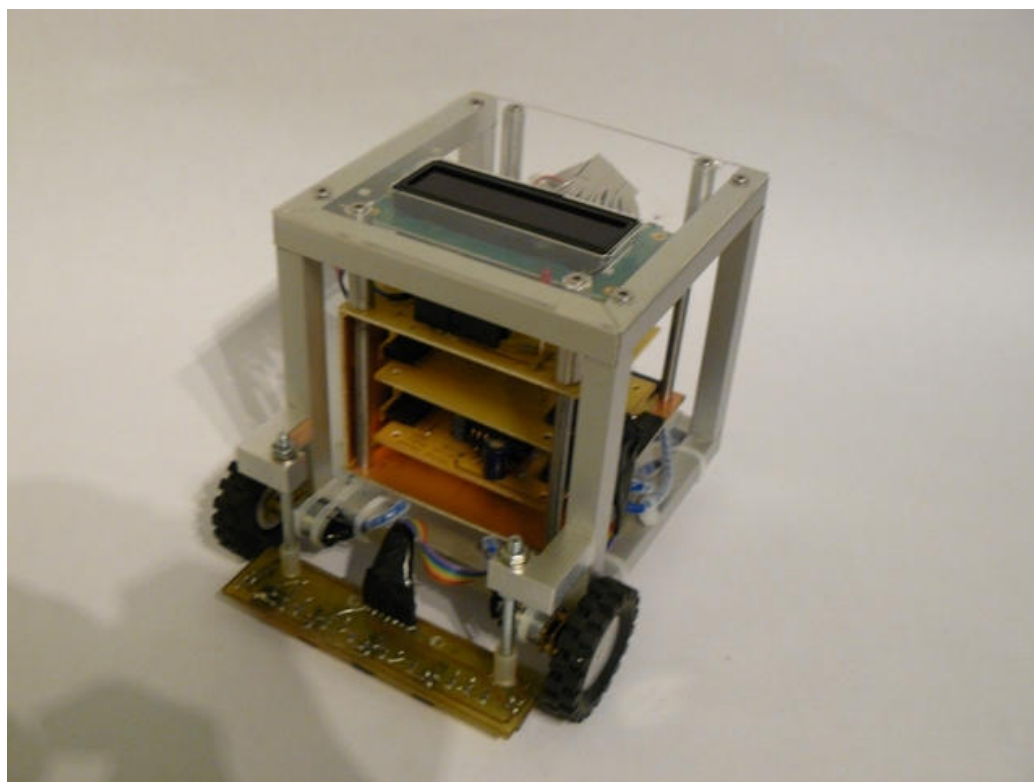
Obsah přiloženého CD:

- Bakalářská práce v elektronické podobě
- Zdrojový kód programu modulu pro sledování čáry
- Zdrojový kód programu modulu pro řízení motorů
- Zdrojový kód programu řídicího systému
- Zdrojové kódy knihovny *mot.h*
- Soubory programu Eagle (*.sch, *.brd): schéma zapojení a návrh desky plošných spojů modulu pro sledování čáry
- Soubory programu Eagle (*.sch, *.brd): schéma zapojení a návrh desky plošných spojů modulu pro řízení motorů
- Soubory programu Eagle (*.sch, *.brd): schéma zapojení a návrh desky plošných spojů řídicí jednotky

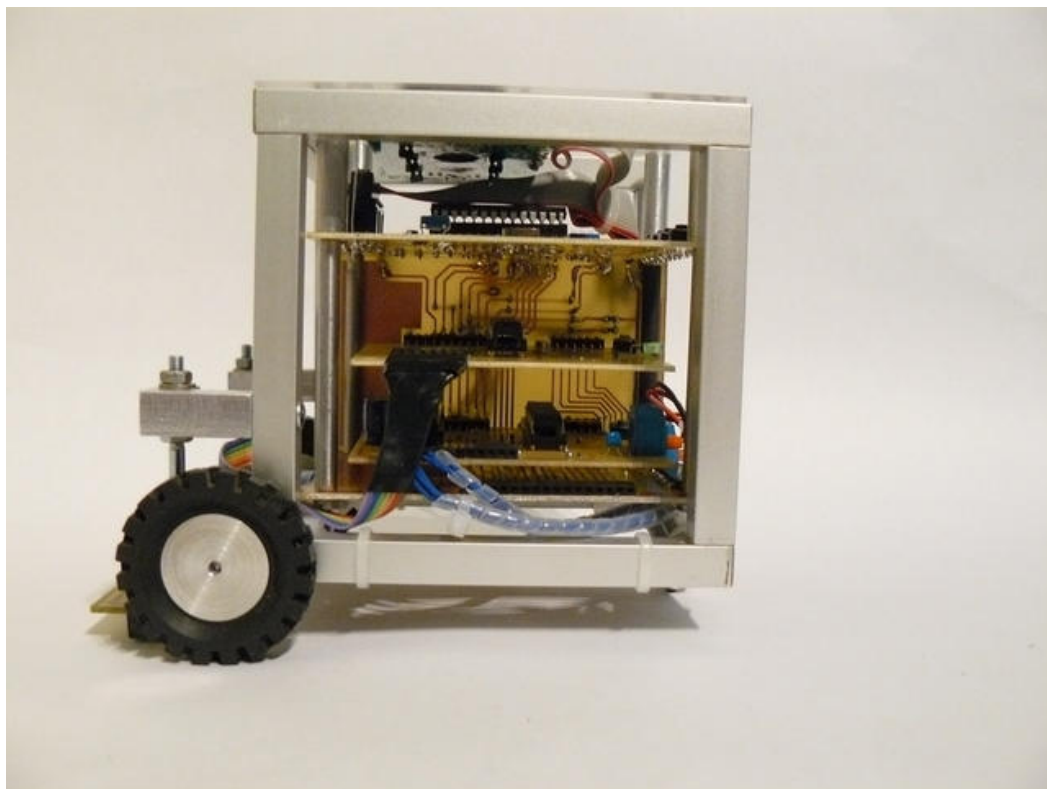
Příloha 1. Obrazová dokumentace



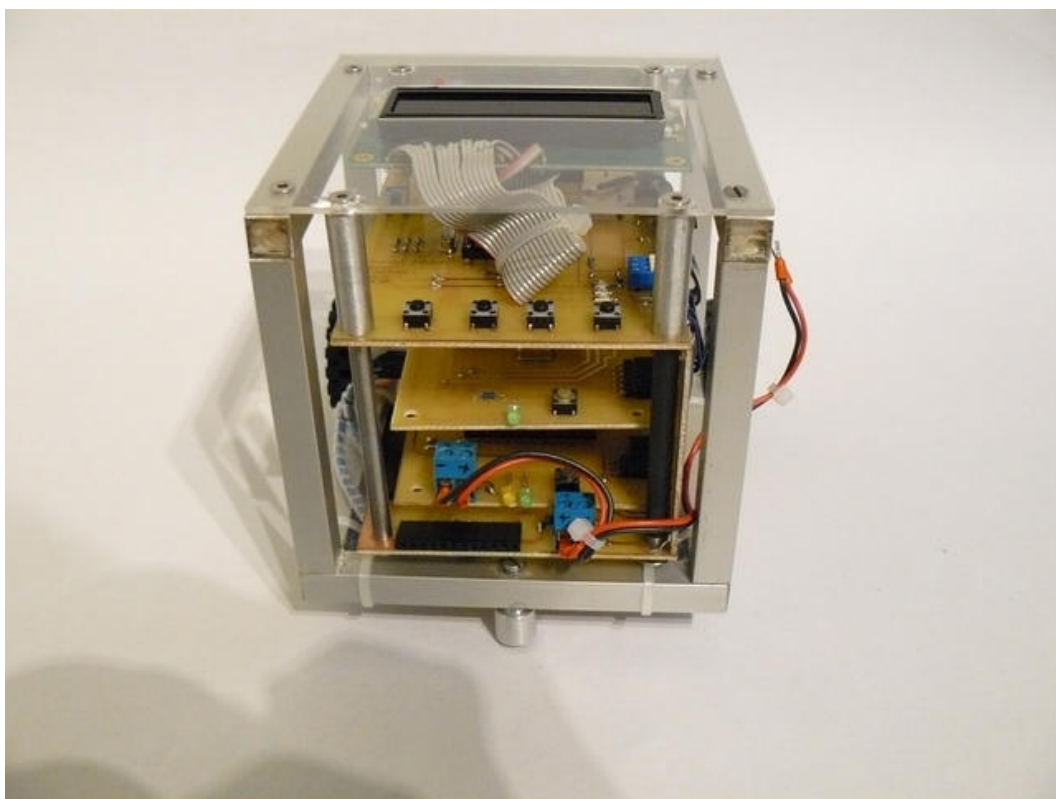
Obrázek 1: Pohled na robota 1



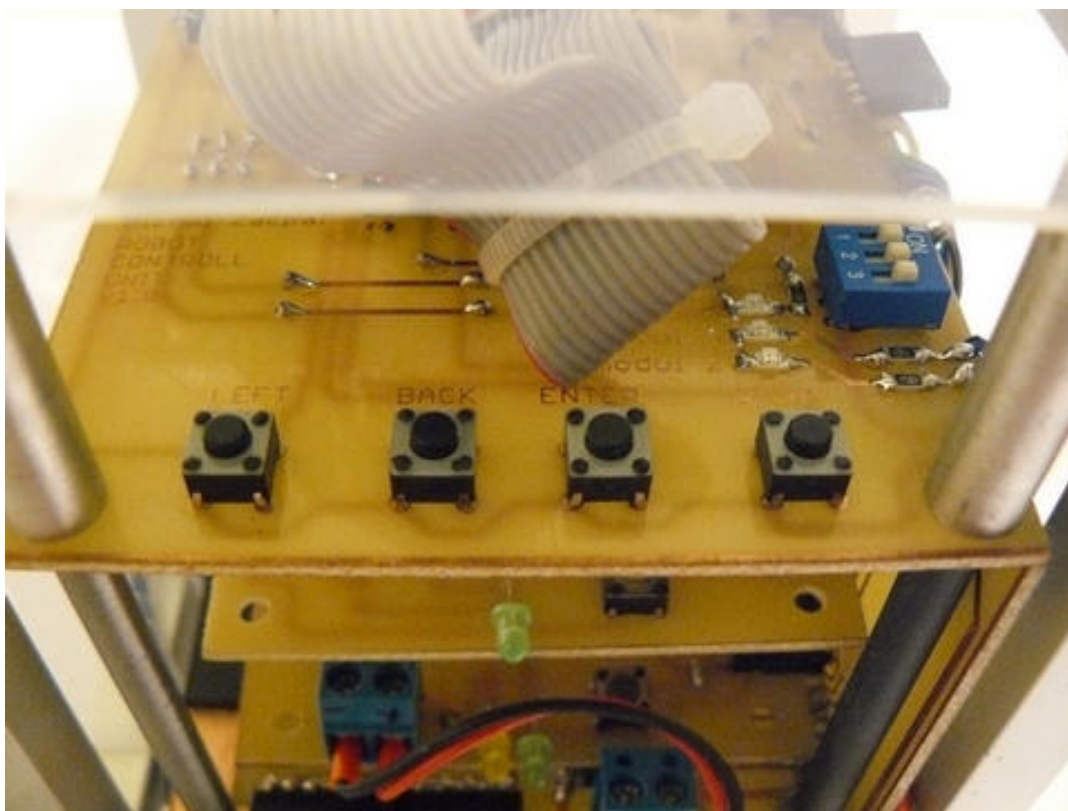
Obrázek 2: Pohled na robota 2



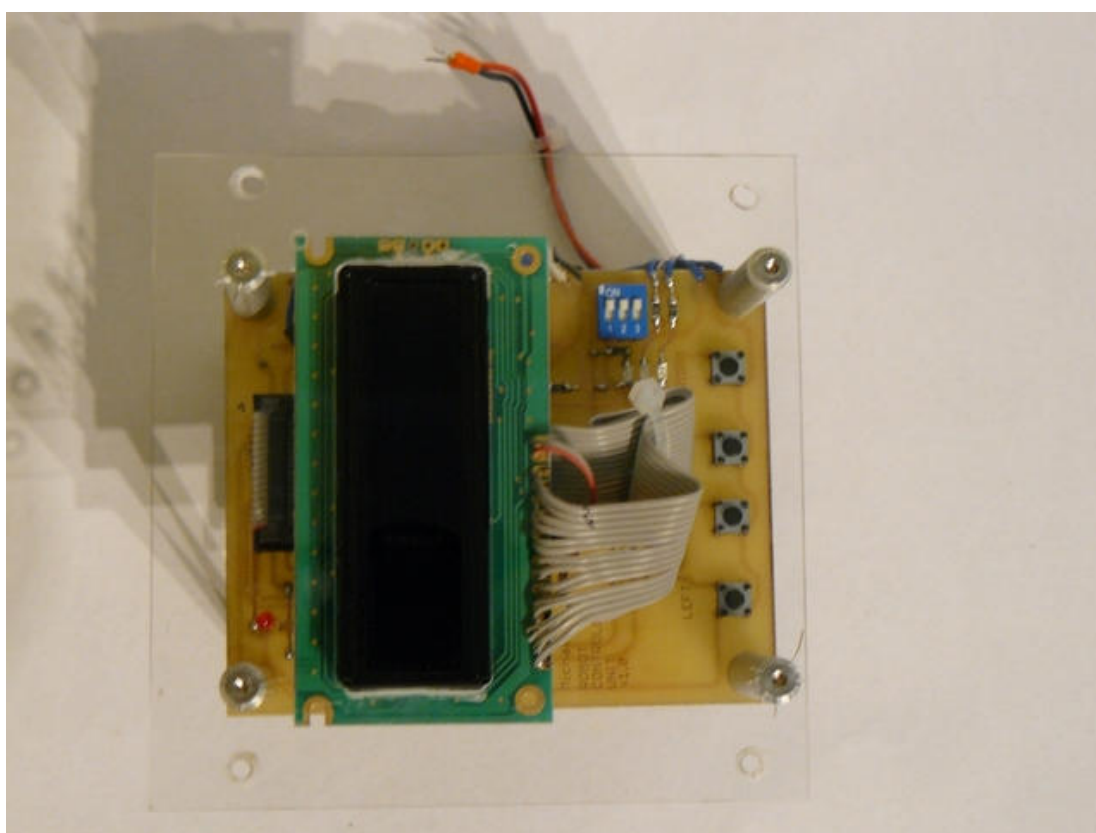
Obrázek 3: Pohled na robota ze strany



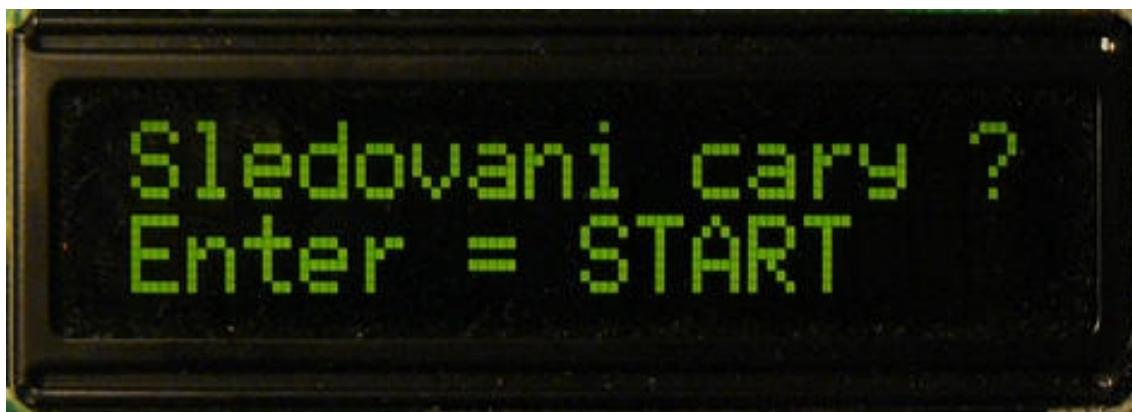
Obrázek 4: Pohled na robota ze zadní strany



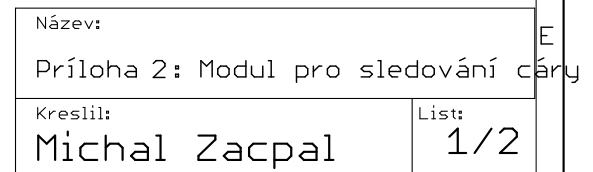
Obrázek 5: Pohled na ovládací tlačítka řídicí jednotky

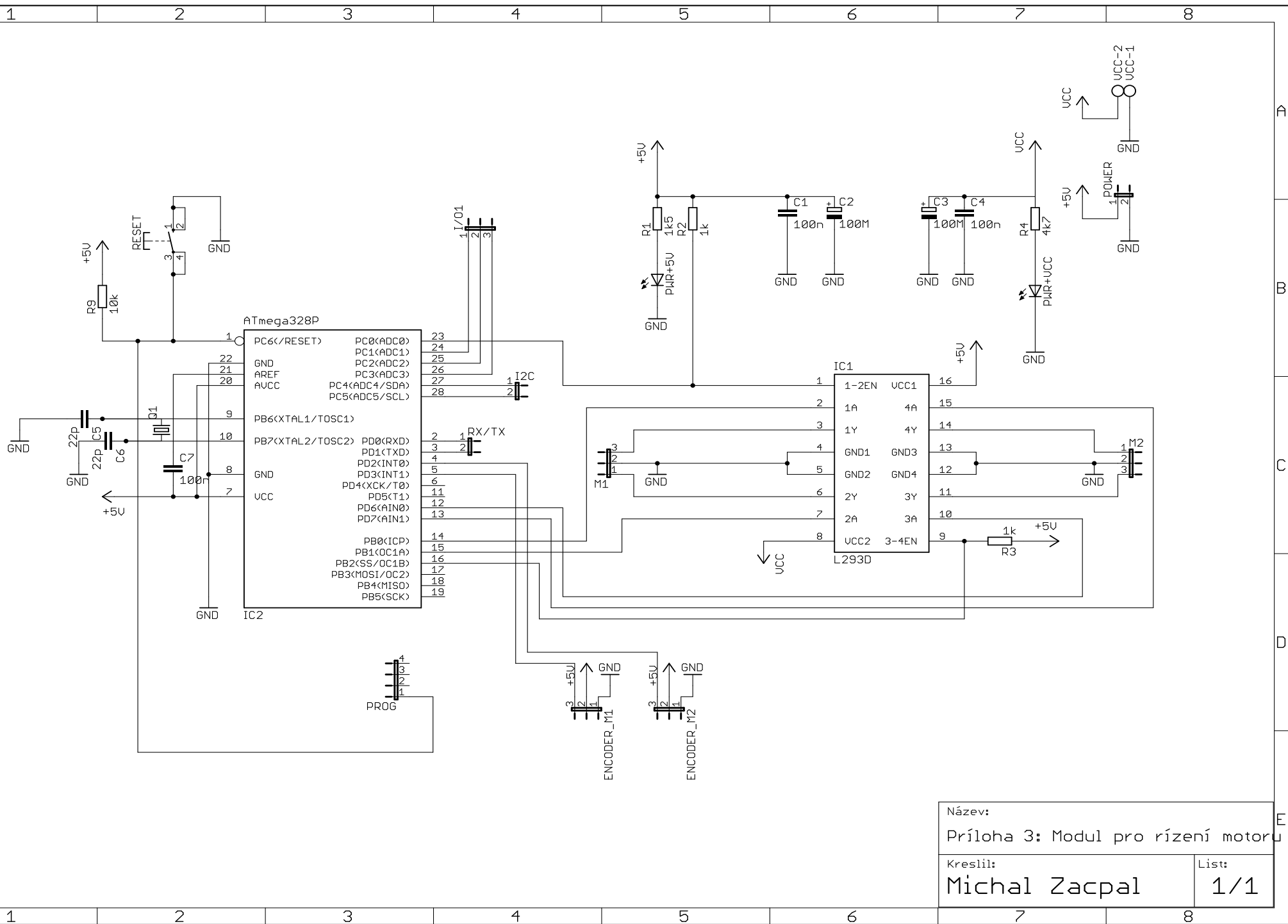


Obrázek 6: Osazená řídicí jednotka s displejem a plexi, pohled shora

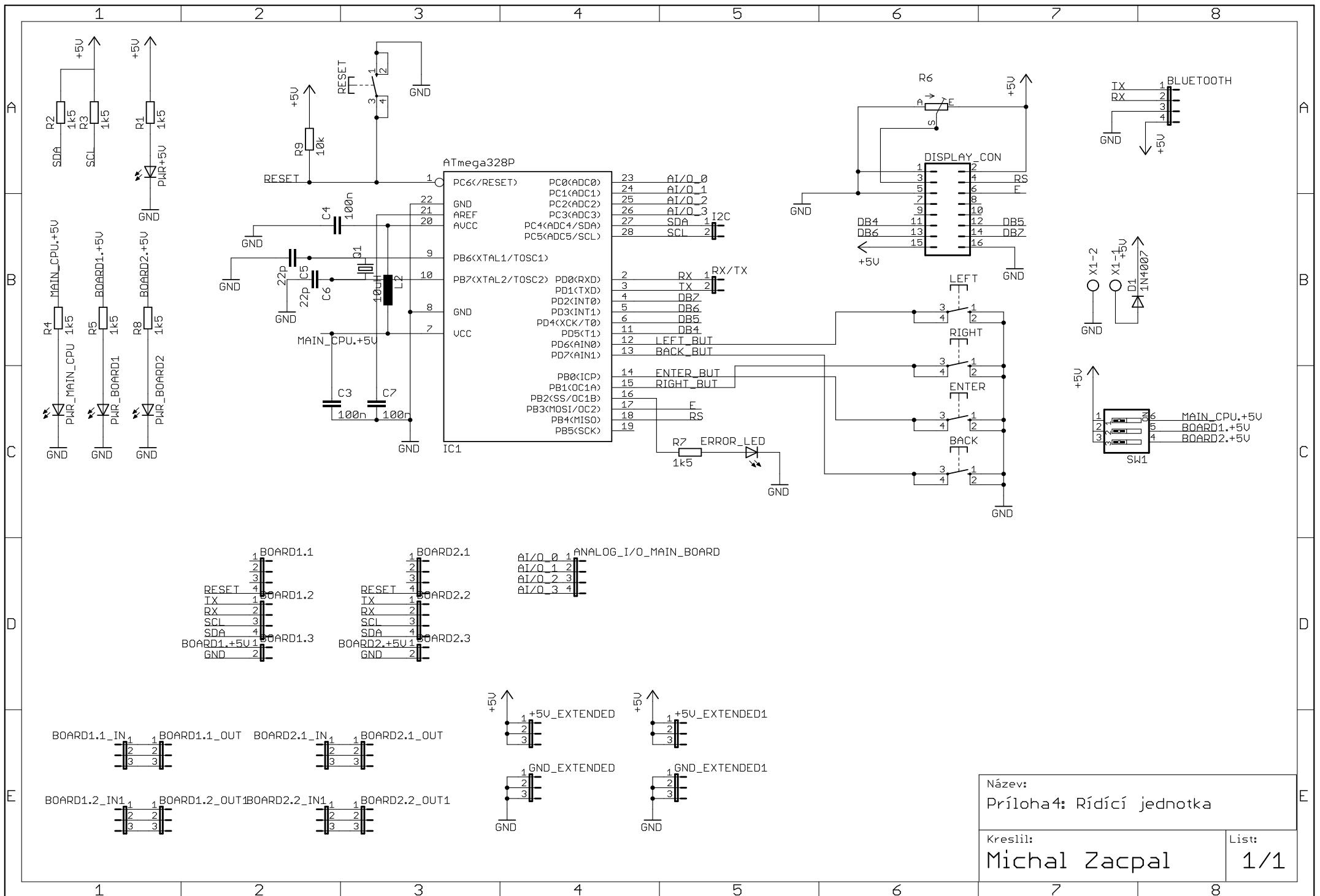


Obrázek 7: Ukázka výpisu LCD displeje

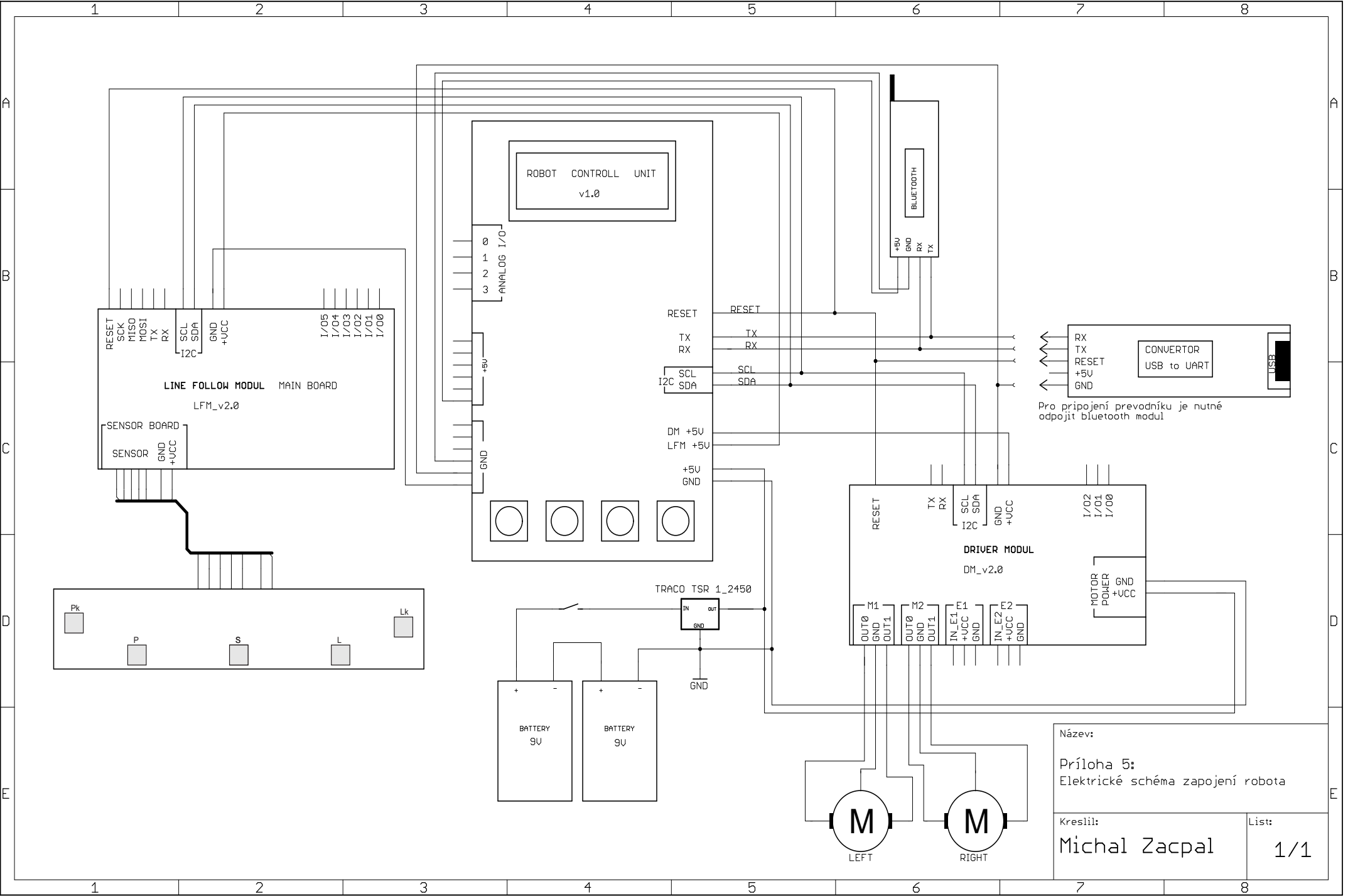


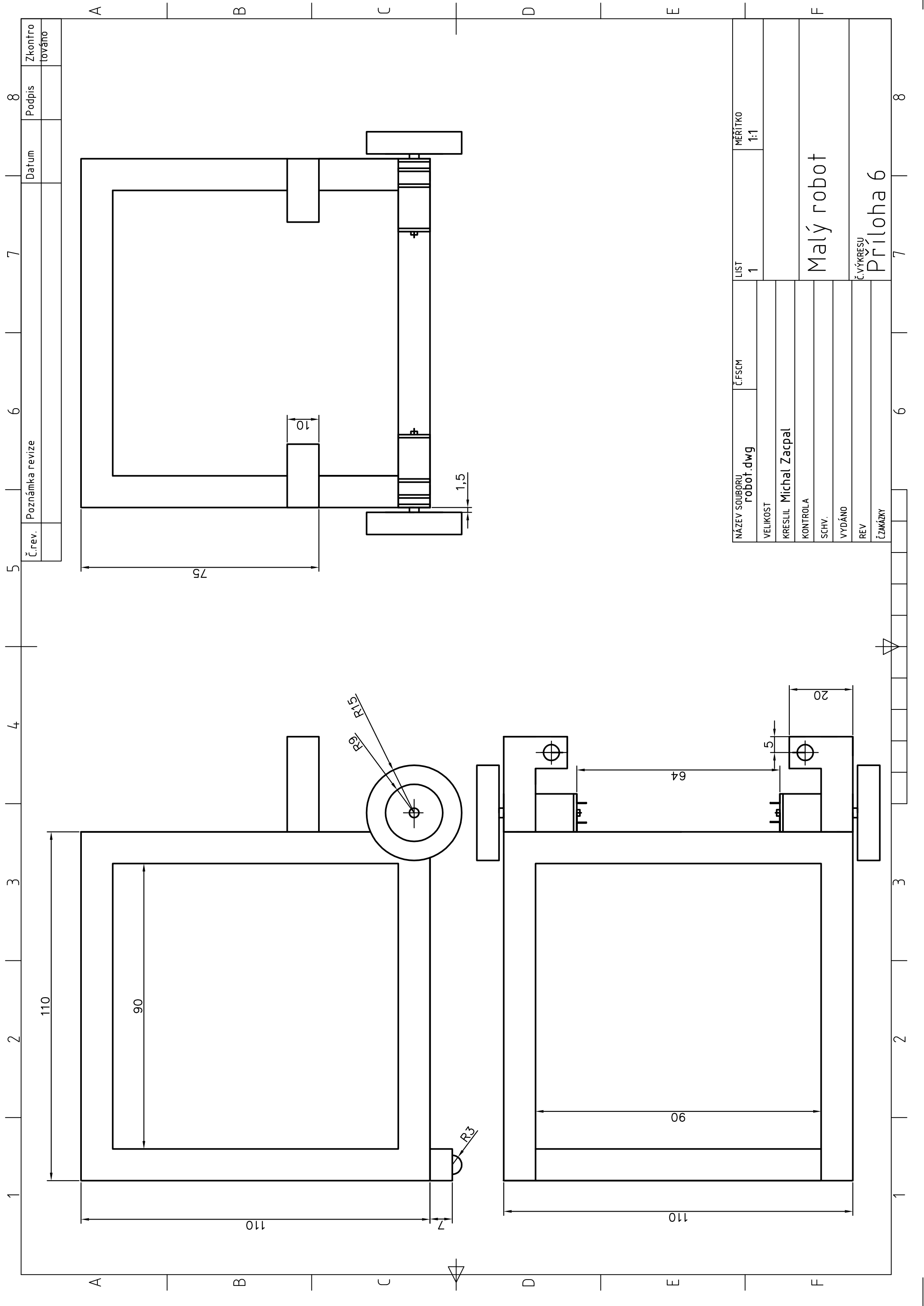


Název:	
Príloha 3: Modul pro řízení motoru	
Kreslil:	List:
Michal Zacpal	1/1



Název:	
Príloha 4: Rídící jednotka	
Kreslil:	List:
Michal Zacpal	1/1





Č.rev.	Poznámka revize	Datum	Podpis	Zkontrolováno

NÁZEV SOUBORU robot.dwg		Č.FSCM	LIST 1	1	MĚŘÍTKO 1:1
VELIKOST					
KRESLIL Michal Zápál					
KONTROLA					
SCHV.					
VYDÁNO					
REV					
Č. VÝKRESU					
Č. ZAKÁZKY					

Malý robot

Príloha 6