

Robot sledující čáru

Line-following robot

Tomáš Spurný

Bakalářská práce
2011



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2010/2011

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Tomáš SPURNÝ**
Osobní číslo: **A08090**
Studijní program: **B 3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**

Téma práce: **Robot sledující čáru**

Zásady pro vypracování:

1. Seznamte se s existujícími konstrukcemi podobných robotů a navrhnete konstrukci jednoduchého a levného robota schopného sledovat čáru.
2. Navrhnete mechanické provedení robota a zapojení řídicího obvodu.
3. Realizujete funkční prototyp robota.
4. Vytvoříte základní programové vybavení.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. PINKER, Jiří. Mikroprocesory a Mikropočítače. Praha : BEN – technická literatura, 2004. 220 s. ISBN 80-7300-110-1.
2. MANN, Burkhard. C pro mikrokontroléry. Praha : BEN – technická literatura, 2004. 280 s. ISBN 80-7300-077-6.
3. CATSOULIS, John. Designing Embedded Hardware. OReilly Media, 2005. 400 s. ISBN 978-0-596-00755-3.
4. SKALICKÝ, Petr. Mikroprocesory řady 8051. Praha : BEN – technická literatura, 2002. 159 s. ISBN 80-86056-39-2.
5. NOVÁK, Petr. Mobilní roboty. Praha : BEN – technická literatura, 2005. 256 s. ISBN 80-7300-141-1.
6. KARGER, Adolf. KARGEROVÁ, Marie. Základy robotiky a prostorové kinematiky. Praha : BEN, ČVUT, 2008. 265 s. ISBN 80-01-02183-1.

Vedoucí bakalářské práce:

Ing. Jan Dolinay, Ph.D.

Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce:

25. února 2011

Termín odevzdání bakalářské práce:

7. června 2011

Ve Zlíně dne 25. února 2011

prof. Ing. Vladimír Vašek, CSc.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

ABSTRAKT

Cílem této práce je navrhnout konstrukci robota schopného sledovat čáru, kterou bude považovat za dráhu své jízdy. Návrh dále zahrnuje i vlastní zapojení řídicího modulu a základní programové vybavení pro vývojový kit Arduino Uno. Součástí bude také zhotovený funkční prototyp.

Klíčová slova: Robot, Arduino Uno, Modul, Mikropočítač, Časovač, Servomotor, Senzor

ABSTRACT

The main purpose of this work is to propose a design of the robot that will be able to follow the line. This line will be considered like a way of ride. The suggestion also includes its own connection control module and the basic software equipment for development kit Arduino Uno. An operable prototype is component of this work.

Keywords: Robot, Arduino Uno, Module, Microcomputer, Timer, Servomotor, Sensor

Zde bych chtěl poděkovat vedoucímu mé bakalářské práce panu Ing. Janu Dolinayovi za jeho rady a vstřícné jednání, které mi poskytoval během návrhu a vytváření tohoto projektu.

Motto:

“

Svoboda bez vzdělání je nebezpečná, vzdělání bez svobody je zbytečné.

”

John Fitzgerald Kennedy (*1917 - †1963)

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....

podpis diplomanta

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 ROBOTY	11
1.1 OBECNĚ.....	11
1.2 DĚLENÍ ROBOTŮ	11
1.3 MOBILNÍ ROBOTIKA.....	11
1.4 ROBOT SLEDUJÍCÍ ČÁRU.....	12
2 MIKROPOČÍTAČE	13
2.1 JEDNOČIPOVÝ POČÍTAČ.....	13
2.2 ARCHITEKTURA MIKROPROCESORŮ	15
2.3 INSTRUKČNÍ SOUBORY	15
3 ARDUINO UNO	16
3.1 SPECIFIKACE.....	17
3.2 PROGRAMOVÁNÍ.....	18
4 MODELÁŘSKÁ SERVA	19
4.1 POPIS SERVOMOTORU	19
4.2 OVLÁDÁNÍ SERVOMOTORU	20
5 OPTICKÉ SENZORY	21
5.1 INFRAČERVENÉ SENZORY	21
II PRAKTICKÁ ČÁST	22
6 POPIS POUŽITÝCH SOUČÁSTEK	23
6.1 ČASOVAČ NE556N	23
6.1.1 Popis časovače.....	23
6.1.2 Časovač jako komparátor	24
6.2 SERVO VTS-05A	24
6.2.1 Parametry.....	25
6.2.2 Úprava pro kontinuální otáčení	25
6.2.2.1 Odstranění zpětné vazby	26
6.2.2.2 Odstranění mechanického dorazu	26
6.2.2.3 Ovládání upraveného serva.....	26
6.3 SENZOR QRD1114.....	27

7	ELEKTRONICKÉ ZAPOJENÍ.....	28
7.1	POPIS KONEKTORŮ.....	29
7.2	NAPÁJENÍ ROBOTA	30
7.3	VYHODNOCOVÁNÍ SIGNÁLŮ ZE SENZORŮ	30
7.3.1	Dotykové senzory.....	30
7.3.2	Optické senzory.....	31
7.3.2.1	Digitální vyhodnocování.....	31
7.3.2.2	Analogové vyhodnocování	32
7.4	DIL SPÍNAČ	32
8	MECHANICKÉ ZPRACOVÁNÍ.....	33
8.1	TĚLO ROBOTA.....	33
8.2	PODVOZEK ROBOTA.....	34
8.3	UCHYCENÍ SENZORŮ.....	35
8.4	ZÁSUVNÝ MODUL PRO ARDUINO UNO.....	36
8.5	SLOŽENÍ ROBOTA.....	37
9	SOFTWAREVÉ VYBAVENÍ.....	39
9.1	SLEDOVÁNÍ ČÁRY	39
9.1.1	Rovný úsek.....	39
9.1.2	Zatáčka	40
9.1.3	Vyjetí robota z dráhy	40
9.2	HLAVNÍ PROGRAM.....	41
	ZÁVĚR	43
	ZÁVĚR V ANGLIČTINĚ.....	44
	SEZNAM POUŽITÉ LITERATURY.....	44
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	47
	SEZNAM OBRÁZKŮ	48
	SEZNAM PŘÍLOH.....	49

ÚVOD

V dnešní době, až na výjimky, prakticky neexistují elektronická zařízení, která by neřídil alespoň z části nějaký mikropočítač. Postupem času tak roste nejen počet vyrobených kusů, ale také možnosti jejich využití, kdy se dostávají téměř do všech průmyslových odvětví i našich domácností.

Proto není až tak od věci, vyzkoušet si přímo jejich praktické využití. K tomuto účelu jsem zvolil výrobu vlastního pojízdného robota.

Roboti jsou jednou z atraktivních možností, kde lze názorně takové využití mikropočítačů demonstrovat. Existuje také plno každoročních soutěží, ve kterých se konstruktéři a programátoři předhánějí, čím robot dokáže danou disciplínu absolvovat lépe. Disciplín a soutěží je několik a bývají rozděleny podle druhů robotů. Jsou například pohyboví roboti, kteří musí v co nejkratším čase najít cestu z labyrintu, roboti mající za úkol zdolávat náročný terén, překážky a nástrahy, nebo třeba zápasící roboti, jenž jsou v nejrůznějších disciplínách nasazováni proti sobě. Populární je také robot, který umí projíždět libovolnou dráhu tvořenou černou čarou na kontrastním povrchu.

Tato bakalářská práce se věnuje právě návrhu robota, který má sledovat černou čáru. Robot musí být konstrukčně navržen tak, aby se mohl pohybovat do všech stran a projet bez problému předem danou dráhu. Jeho pohyb zajistí dva servomotorů s diferenciálním řízením pohybu (princip podobný jako u tanku). Aby robot dokázal sledovat dráhu tvořenou například černou izolační páskou, je zapotřebí senzorů schopných černou čáru rozpoznat od podkladu, na kterém je umístěna. Tento úkol zastanou optické senzory, fungující na principu vysílání a zachycování odraženého infračerveného světla. Je obecně známo, že světlé povrchy světelné záření odrážejí, zatímco tmavé světlo pohlcují. Tohoto jevu lze potom při sledování černé čáry využít. O řízení robota se následně postará příslušná elektronika a naprogramovaný mikropočítač řídící a monitorující periferie robota.

Vytvořený robot sice může na běžného uživatele působit jen jako hračka, ale i přesto najde své využití například jako učební pomůcka při výuce odborných předmětů, nebo jako inspirace pro zájemce, jejichž zájmy se ubírají tímto směrem.

I. TEORETICKÁ ČÁST

1 ROBOTY

1.1 Obecně

V 17. Století bylo slovo *robota* spojeno výhradně s otrockou prací. Až později Karel Čapek ve své hře R.U.R. zmínil slovo *robot* ve významu stroje, když přemýšlel, jak pojmenovat umělou bytost [8].

Dnes se od robota požaduje, aby byl schopný v jisté míře vykonávat úkony samostatně podle předem stanovených instrukcí nebo interakčně v závislosti na okolí či pokynech uživatele. Robot může být také vybaven různými senzory a vysílacími moduly, které pro něj pracují jako smysly pro vnímání a poznávání okolního prostředí a vnějších vlivů. Složitější roboti jsou schopni se na základě získaných informací i něco naučit a vyvarovat se tak třeba do budoucna různým situacím a stavům, které jsou pro uživatele nežádoucí. Tomu se říká umělá inteligence.

1.2 Dělení robotů

Roboty lze dělit podle různých kritérií. Například podle pohybových možností, autonomie, způsobu jejich programování, podle jejich účelu, vzhledu, schopností, způsobu vzniku atp.

Podle schopnosti přemísťovat se, však dělíme roboty na:

- Stacionární
- Mobilní

Zatím co stacionární roboti se nemohou pohybovat z místa na místo a plní funkci tzv. manipulátorů, mobilní mají tu výhodu, že se mohou pohybovat po okolí, tak jak jim jejich konstrukce dovolí. Jedná se právě o skupinu, do které spadá i robot sledující čáru.

1.3 Mobilní robotika

Mobilní robotika se zabývá roboty schopnými se v daném prostředí a čase přemísťovat. Toto odvětví robotiky spojuje dohromady informatiku, elektroniku, mechaniku i fyziku. Mobilních robotů je mnoho druhů a to zejména podle jejich použití. V základu je dělíme na autonomní a dálkově řízené.

Autonomní roboti jsou schopni samostatně vykonávat zadanou úlohu, na kterou jsou zkonstruováni. Takový robot se rozhoduje na základě toho, jak byl naprogramovaný a jeho úkony jsou tak pevně dané. Dokáže například zmapovat prostředí pomocí senzorů a podle získaných informací se po okolí pohybovat.

Jak už název napovídá, dálkově řízení roboti jsou ovládání nějakým uživatelem nebo operátorem na dálku. I zde může být robot vybaven různými senzory. Ty však splňují funkci spíše informační. Tu vyhodnocuje většinou sám uživatel, protože převážná odpovědnost co se s robotem bude dít, jak a kam se bude pohybovat, spočívá výhradně na něm.

1.4 Robot sledující čáru

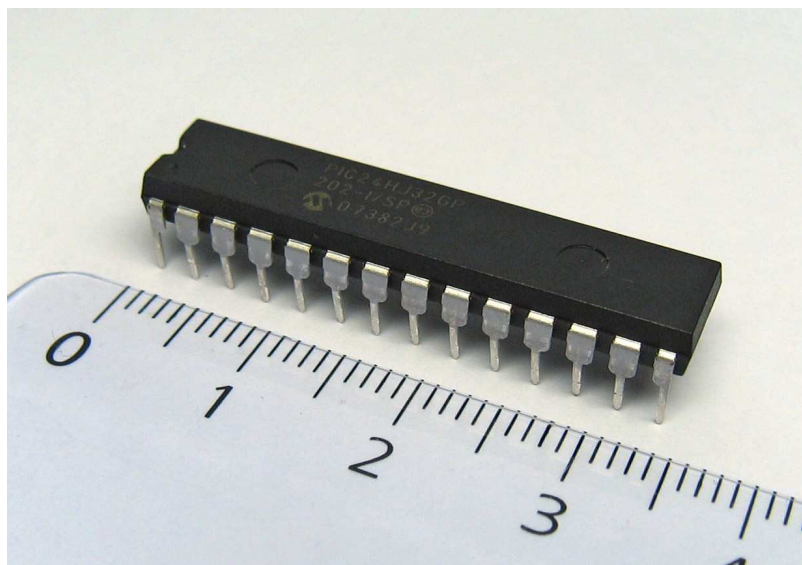
V případě robota sledujícího čáru, bude využit autonomní systém. Robot bude na základě infračervených senzorů zjišťovat, jestli se pohybuje po předem vytvořené dráze, či nikoliv. Dráhu bude tvořit černá páska na bílém podkladu. O vyhodnocování se stará jeho elektrické zapojení společně s programem, který bude mít uložený ve své paměti. Po mechanické stránce bude využita koncepce s dvoukolovým podvozkem s diferenciálním řízením pohybu, takže směr a rychlost pohybu bude řízena rychlostmi jednotlivých servomotory, které jsou ovládány nezávisle na sobě. Kvůli stabilitě je na podvozku přimontováno také všesměrové kolečko.

2 MIKROPOČÍTAČE

V 60. – 80. letech minulého století, se pod pojem mikropočítač, skrýval následník dřívějších velkých sálových počítačů. Později se mikropočítačem nazývaly zařízení, které byly výrazně menší, než stolní počítač. Dnes už je mikropočítač zpravidla představován jediným čipem.

2.1 Jednočipový počítač

Jedná se o mikrokontrolér (MCU), který je většinou zapouzdřený ve formě integrovaného obvodu. Díky své spolehlivosti, malým rozměrům a poměrně univerzálnímu využití, se stal hojně používanou součástí pro řídicí hardware, který se využívá převážně pro jednorúčelové aplikace se specifickým zaměřením. MCU najdeme téměř v každé spotřební elektronice nebo v systémech, co na starost regulaci, řízení, monitorování apod. [9][10].

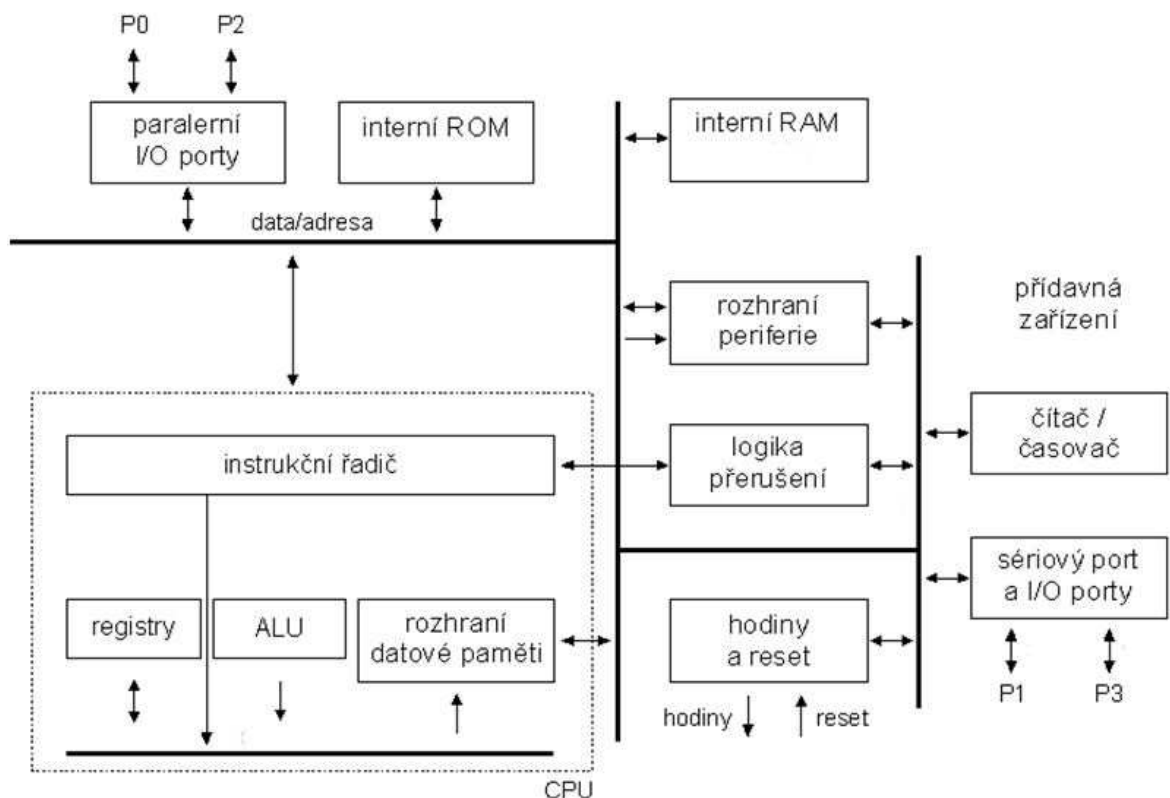


Obrázek 1. Mikrokontrolér řady PIC

Integrovaný čip obsahuje vlastní mikroprocesor, operační paměť (RAM) pro ukládání dat a pevnou paměť sloužící pro uložení vlastního programu (FLASH, ROM, PROM, EPROM, EEPROM). Dále v něm najdeme obvody, jenž slouží pro řízení různých periférií. Jedná se zpravidla o sériové rozhraní, logické vstupy/výstupy, komunikační linky, řadič přerušení, čítače, časovače, A/D převodníky, D/A převodníky, obvody podporující PWM apod. Díky

tomuto rozhraní zastane mikrokontrolér plno funkcí, které jsou potřebné pro chod mnoha aplikací.

Mikrokontrolérů existuje celá řada, lišící se od sebe, rozměrem, počtem vstupů/výstupů, komunikačními kanály a nejrůznějšími parametry. Obecně však existuje vnitřní struktura, která je analogicky stejná ve všech mikropočítačích. Tato struktura je znázorněna na obrázku 2.



Obrázek 2. Obecné blokové schéma mikrokontroléru

Nejdůležitějším blokem každého mikropočítače je centrální procesorová jednotka (CPU). Ta obsahuje aritmeticko-logickou jednotku (ALU), která provádí všechny aritmetické nebo logické operace. Příslun instrukcí, které má ALU vykonávat, zajišťuje instrukční řadič, jenž uchovává operační kód příslušné instrukce do doby, než se tato instrukce vykoná. Další součástí centrální procesorové jednotky jsou registry, do kterých se ukládají operandy a výsledky početních operací. Rozhraní datové paměti má pak za úkol komunikovat s pamětí, v které jsou data nezbytná pro chod vlastního programu včetně uživatelských dat.

2.2 Architektura mikroprocesorů

Architektura mikroprocesorů se dělí do dvou skupin na základě toho, jak je nakládáno s pamětí pro vlastní data a pamětí určenou pro program.

Von Neumannova architektura má paměť určenou pro data a program společnou. Výhodou je, že je použita jen jedna datová a adresová sběrnice a netřeba tak rozlišovat přístup do paměti programu a dat. Nevýhodou je omezení rychlosti pro práci s instrukcemi, která je u dnešních mikroprocesorů daleko vyšší, než rychlost potřebná pro práci s pamětí. Vzhledem ke sdílené paměti, je u této architektury rovněž nebezpečí přepsání vlastního zdrojového kódu daty.

Hardvardská architektura díky oddělené paměti programu a dat, zajišťuje vyšší rychlost pro zpracování instrukcí a je tak schopna pokrýt dnešní nároky, které se na mikroprocesory kladou. Rovněž je výhodou, že mikroprocesor je schopný číst instrukce i potřebná data ve stejný okamžik. Nevýhodou je naopak technologická náročnost, kdy je potřeba zajistit dvě sběrnice.

2.3 Instrukční soubory

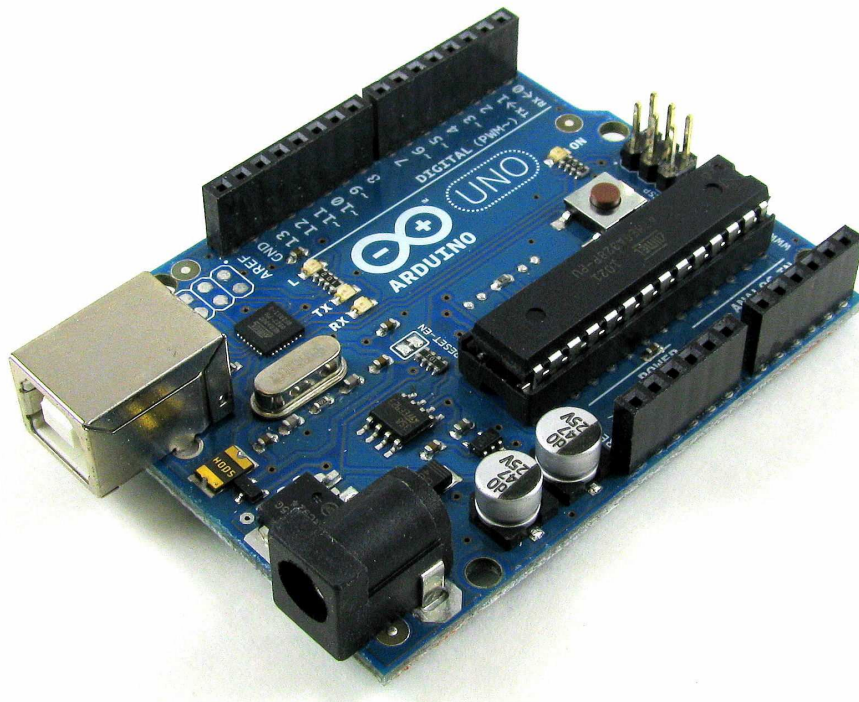
Podle používaných instrukcí, jenž obsahuje zdrojový kód, který tvoří program pro mikroprocesor, rozdělujeme instrukční soubory do dvou skupin.

CISC je označení pro mikroprocesor obsahující poměrně složitý instrukční soubor. Výhodou je, že na mnoho nejrůznějších operací a úkonů existuje vlastní instrukce a díky tomu se zdrojový kód programu stává přehlednějším a také zabere v paměti méně místa. Nevýhoda spočívá v použití složitějšího dekodéru instrukcí a tím je dáno i pomalejší zpracovávání instrukcí. Pro programátora to také znamená, že si musí pamatovat poměrně vysoký počet instrukcí, aby byl schopen napsat efektivní program.

RISC nabízí jednodušší kódování a také menší počet instrukcí. Díky tomu není potřeba používat složitý dekodér instrukcí a roste tak rychlost, jakou jsou instrukce zpracovávány. Jednodušší je potom i vlastní programování. Protože jsou však instrukce RISC jednodušší, je jich potřeba daleko více pro vykonání určité operace, než by tomu bylo u CISC. Napsaný program tak zabere více místa a má větší nároky na spotřebu paměti, ve které je uložený.

3 ARDUINO UNO

Arduino Uno je vývojová deska (kit), která kromě mikropočítače obsahuje vlastní rozhraní k naprogramování obvodu, řadu portů pro komunikaci s okolím, signalizační prvky a jiné vymoženosti. Označením Uno, spadá kit do kategorie několika verzí fyzické počítačové platformy Arduino. Kit má open-source licenci a jsou tak k dispozici zdrojové soubory a detailní schéma zapojení s podrobnou dokumentací. Na základě toho je možné tuto platformu různě modifikovat, rozšiřovat, vylepšovat a také k ní vyrábět řadu doplňků a klonů. Rozšířené jsou především periferie známé jako *štíty* (shields), které mají identické rozložení konektorů jako Arduino. V případě využití průchozích konektorů je možné na sebe naskládat hned několik desek a pokrýt tak širší využití kitu v praxi [11][12].



Obrázek 3. Vývojový kit Arduino Uno

Kit obsahuje jednoduchou desku disponující základními I/O porty a mikrokontrolér ATmega328 od firmy Atmel. Tento mikroprocesor využívá architekturu AVR (osmibitový RISC, hardvarská architektura). Komunikace s PC je prováděna přes USB rozhraní a díky implementovanému převodníkovému čipu USB-to-RS232, je zajištěna virtuální sériová linka, pro kterou jsou k dispozici ovladače na všechny platformy.

3.1 Specifikace

Mikroprocesor

- Atmel ATmega328
- 16MHz
- 32kB flash, 2kB SRAM, 1kB EEPROM
- Atmel AVR architektura

I/O piny

- **14 digitálních portů**
 - 0, 1 – sériová linka
 - 2, 3 – externí přerušení
 - 3, 5, 6, 9, 10, 11 – PWM
 - 10, 11, 12, 13 – SPI
 - 4, 5 – I2C
- **6 analogových vstupů**

Připojení k PC

- USB
- RS-232 za použití převodníku úrovní

Napájení

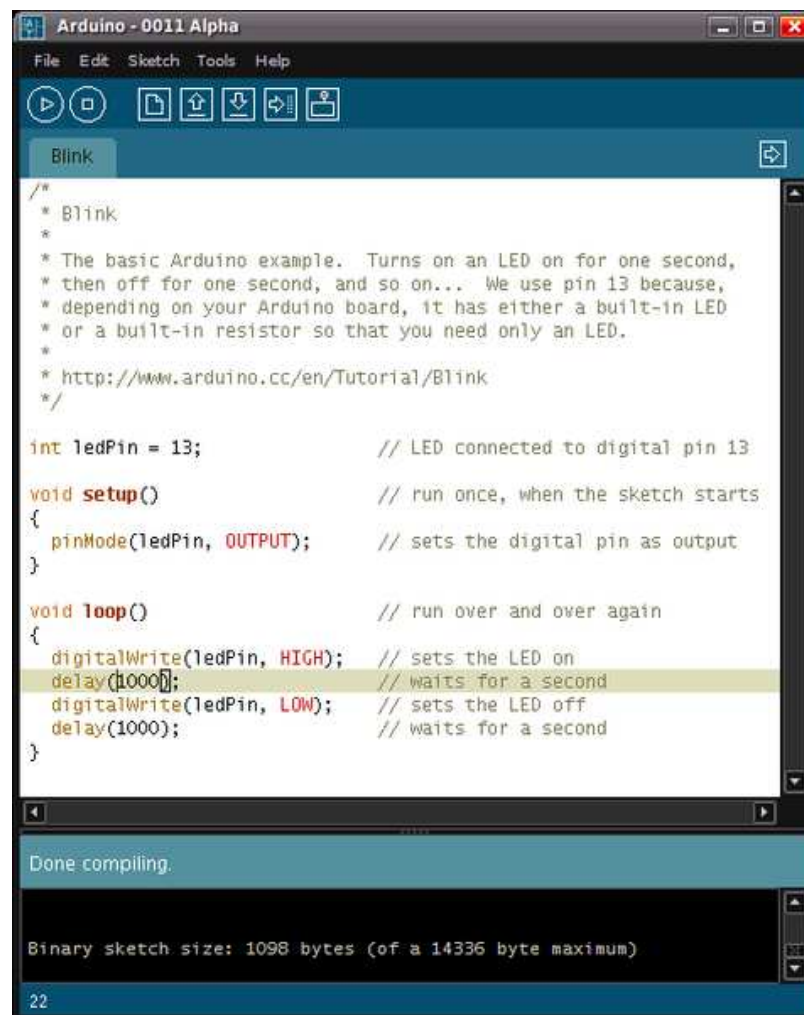
- 5V USB
- 7-12V adaptér

Programovací jazyk

- Wiring s Processing IDE
- Multiplatformní vývojové prostředí (WIN/Linux/Mac/...)

3.2 Programování

K programování slouží vývojové prostředí, u kterého se zdrojový kód píše v Arduino Programmable Language, jenž je založený na jazyce Wiring. Je to v podstatě jazyk vzniklý z C, který je modifikovaný na programování jednočipových zařízení. Vývojové prostředí je JAVA aplikace a umožňuje pro psaní programů využívat různé dodávané knihovny. V tomto prostředí jde sice sledovat třeba sériová linka, ale díky své jednoduchosti už neobsahuje širší možnost ladění samotného programu.



```
Arduino - 0011 Alpha
File Edit Sketch Tools Help
Blink
/*
 * Blink
 *
 * The basic Arduino example. Turns on an LED on for one second,
 * then off for one second, and so on... We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

int ledPin = 13;           // LED connected to digital pin 13

void setup()              // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop()              // run over and over again
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);               // waits for a second
  digitalWrite(ledPin, LOW);  // sets the LED off
  delay(1000);               // waits for a second
}

Done compiling.

Binary sketch size: 1098 bytes (of a 14336 byte maximum)

22
```

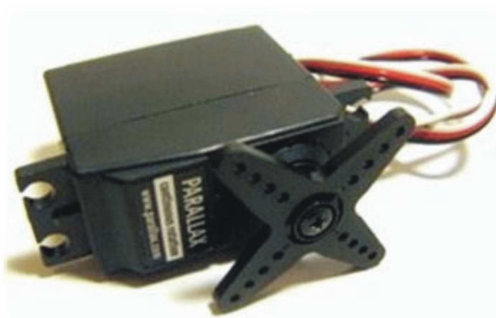
Obrázek 4. Vývojové prostředí pro Arduino

4 MODELÁŘSKÁ SERVA

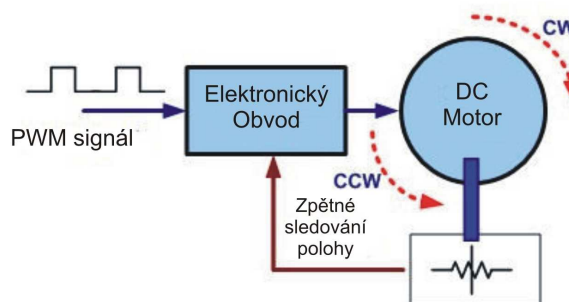
Tyto serva mají na dnešní robotiku poměrně velký vliv a využívají se především v dálkově řízených RC modelech. Díky malým rozměrům lze serva šikovně zakomponovat i do míst, kam by se samotný stejnosměrný elektromotor jen těžko montoval. Zmíněná mechanicko-elektrická součástka má oproti obyčejným motorkům nespornou výhodu v možnosti jejich řízení, kdy můžeme za pomoci pulzně šířkové modulace (PWM) přesně nastavit polohu natočení osy.

4.1 Popis servomotoru

Servo se skládá ze tří základních prvků. Nezbytnou část tedy tvoří malý stejnosměrný motor, většinou umělohmotná převodovka a řídicí elektronika se zpětnovazebním potenciometrem.



Typický servomotor



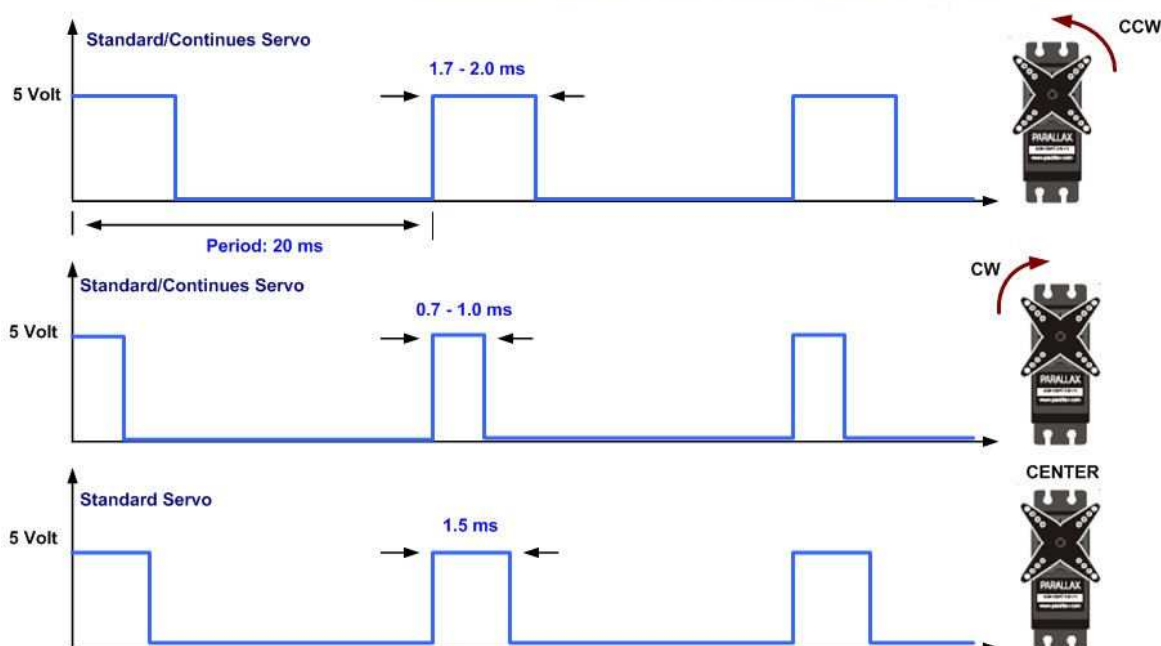
Blokové schéma servomotoru

Obrázek 5. Servomotor a blokové schéma řízení

Z každého serva vede tři žilový kablík, jenž má na konci konektor, který je možné napojit na vývody určené pro jumpery. Jednotlivé dráty jsou od sebe barevně odlišeny, kde zpravidla černá barva představuje GND, červený drát napájení Vcc a oranžový (nebo jinak barevný) značí vstup pro PWM. Na tento přívod je v pravidelném intervalu poslán kladný pulz o frekvenci 50Hz, na jehož délce (šířce) závisí právě natočení serva. Rotor tvoří ozubené kolo, na které je možné připevnit umělohmotné násady, dodávané většinou společně se servomotorem. Uvnitř je na rotor napojený potenciometr, který jako zpětná vazba snímá aktuální natočení a jeho napětí je přiváděno na vyhodnocovací elektroniku.

4.2 Ovládání servomotoru

Běžně dostupné serva jsou schopny natočit rotor od 0° do maximálně 180° . Úhel natočení je dán šířkou pulzu, které jsou většinou dlouhé 1-2ms s konstantní frekvencí 50Hz. Z toho vyplývá, že pokud budeme chtít servomotor nastavit do střední polohy (90°), bude potřeba generovat pulz o šířce 1,5ms. Amplituda pulzů bývá stejná jako velikost napájení serva. Tato hodnota se pohybuje kolem 5V ($\pm 0,2V$). Změna směru otáčení se však neovládá změnou polaritý napájení. To je potřeba striktně dodržovat, jinak se může poškodit vnitřní elektronika. Pootočení serva doleva nebo doprava (např. ze střední polohy), tak závisí na tom, jestli budou následné pulzy větší nebo menší 1,5ms. Každý servomotor má jiné krajní hodnoty a není dobré je překračovat. Většina jich totiž má na koncích rozsahu pohybu mechanický doraz. Pokud bychom tuto hranici chtěli překročit, rotor bude blokován, tím by se zvýšil odebíraný proud a vnitřní řídicí elektronika by se mohla poškodit [13][14].



Obrázek 6. Možné průběhy řídicího PWM signálu

Poměrně častý je také mechanický zásah, kdy po rozdělení serva a následné úpravě zajistíme plynulé otáčení o 360° , vhodné pro pohyb některých robotů. Popis úpravy je uveden v praktické části.

5 OPTICKÉ SENZORY

Optické senzory mají oproti dotykovým sensorům (například tlačítko nebo spínač) výhodu v tom, že jsou schopné detekovat vnější vlivy ještě dřív, než dojde k fyzickému kontaktu.

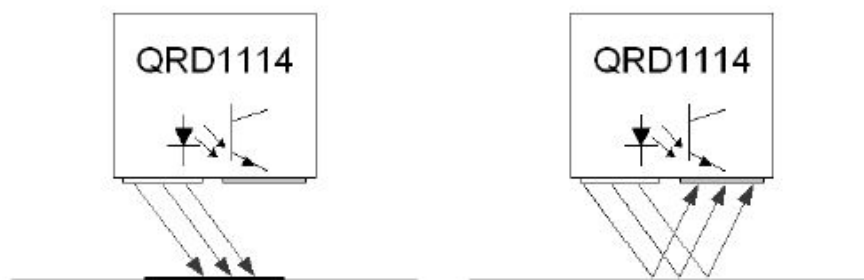
Optické senzory dělíme na aktivní a pasivní.

- Aktivní - vyhodnocují vlastní odražené záření
- Pasivní - vyhodnocují výhradně záření přijaté z okolí

Velmi rozšířené jsou infračervené senzory, které se využívají v dálkovém ovládání.

5.1 Infračervené senzory

Aktivní IR senzory se skládají ze dvou částí - emitor a detektor. Emitorem je IR LED dioda, která vyzařuje infračervené světlo o vlnové délce kolem 940nm. Druhou nezbytnou částí je fototranzistor, který má jako detektor snímat, jestli došlo k odražení IR paprsku od povrchu nebo překážky.



Obrázek 7. Princip odrazu IR světla u QRD1114

V případě že se paprsek odrazí, fototranzistor zmenší svůj vnitřní odpor a protéká přes něj větší proud. Nejedná se o skokovou změnu, takže díky IR senzoru můžeme vyhodnocovat i množství odraženého paprsku. Na senzor má však negativní vliv i sluneční záření, nebo osvětlení žárovkami. Takže se může stát, že fototranzistor bude detekovat odraz i když k žádnému nedojde. Tyto vlastnosti lze do jisté míry eliminovat použitím filtrů nebo kalibrací senzorů ve vlastním zapojení. Množství odraženého světla závisí také na barvě povrchu. Je obecně známo, že černá barva záření pohlcuje, zatímco bílá odráží. Tohoto extrému lze využít pro robota, který bude sledovat černou čáru na bílém povrchu.

II. PRAKTICKÁ ČÁST

6 POPIS POUŽITÝCH SOUČÁSTEK

Tato kapitola věnuje větší pozornost časovači NE556N, použitým servomotorům a senzoru QRD1114.

6.1 Časovač NE556N

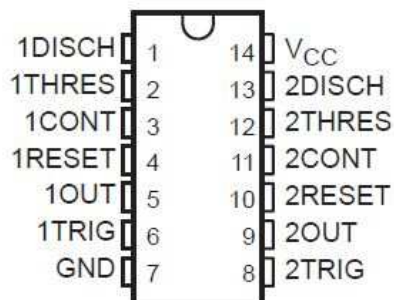
NE556N je velmi rozšířený dvojitý časovač, který je tvořen monolitickým, čtrnácti pinovým integrovaným obvodem, obsahující spojení invertujícího zesilovače, nulovatelného klopného obvodu a komparátoru. Tento obvod se velmi často využívá pro zapojení, kdy chceme zajistit periodický signál. Jedná se například o zapojení pro blikání LED diod v různých režimech, akustickou signalizaci nebo využití komparátorů na sledování vstupního napětí.



Obrázek 8. Časovač NE556N

6.1.1 Popis časovače

Dvojitý časovač obsahuje uvnitř dvě naprosto identické části, které mají nezávisle na sobě stejné vnitřní zapojení a stejné vlastnosti.



Obrázek 9. Rozmístění pinů obvodu NE556N

Časovač je schopný pracovat jako a stabilní a monostabilní multivibrátor. Jeho maximální operační frekvence je větší než 500kHz. Délka jednoho impulsu se pohybuje od μs až po řádově hodiny. Z výstupu je možno získat proud o hodnotě 200mA, kdy je zajištěna plná kompatibilita s TTL obvody. Vnitřní zapojení integrovaného obvodu je tvořeno zhruba 40 tranzistory a 20 rezistory. Více informací lze vyčíst z datasheetu [15].

6.1.2 Časovač jako komparátor

Pro využití časovače jako komparátoru, je nezbytné vycházet z funkční tabulky. Ta popisuje úroveň výstupního napětí v závislosti na kombinaci vstupních napětí na jednotlivých pinech.

FUNCTION TABLE				
RESET	TRIGGER VOLTAGE†	THRESHOLD VOLTAGE†	OUTPUT	DISCHARGE SWITCH
Low	Irrelevant	Irrelevant	Low	On
High	$< 1/3 V_{DD}$	Irrelevant	High	Off
High	$> 1/3 V_{DD}$	$> 2/3 V_{DD}$	Low	On
High	$> 1/3 V_{DD}$	$> 2/3 V_{DD}$	As previously established	

† Voltage levels shown are nominal.

Obrázek 10. Funkční tabulka časovače NE556N

6.2 Servo VTS-05A

Malý servomotor je tvořený umělohmotným krytem, obsahující umělohmotné převody společně s řídicí elektronikou a miniaturním elektrickým motorkem. Napájení včetně řízení serva je zajištěno přívodním kablíkem. Ten je zakončený konektorem, který je možné napojit na kontakty určené pro jumpery.



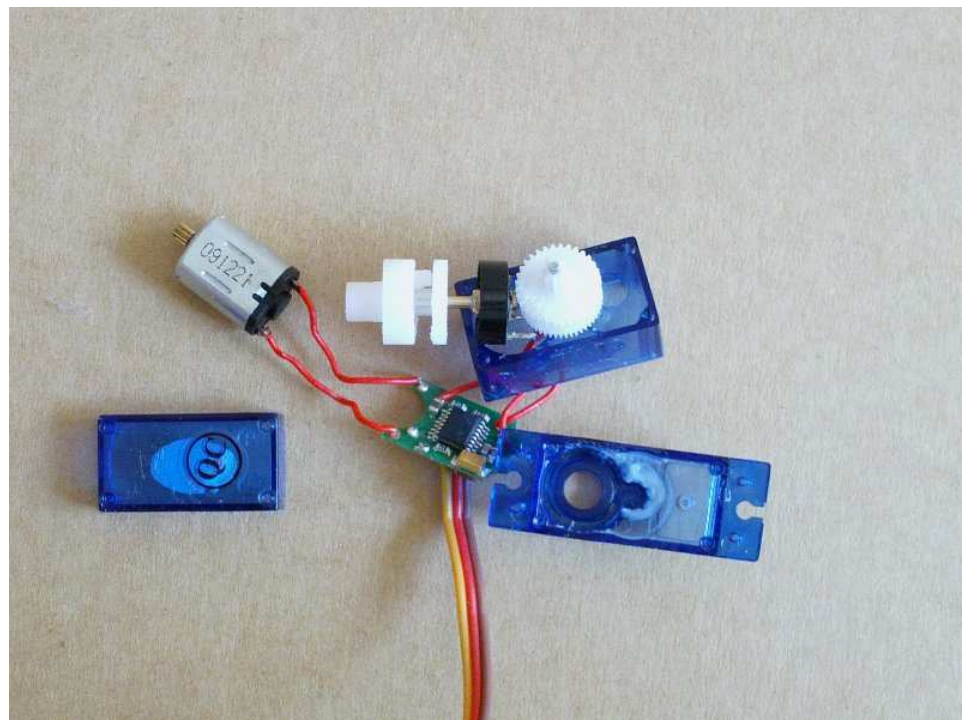
Obrázek 11. VIGOR VTS-05A

6.2.1 Parametry

- Napájení - 4,8-6V
- Šířka řídicího pulzu - 1-2ms
- Frekvence pulzů - 50Hz
- Kroutící moment - 10N.cm při 4,8V
- Rychlost - 0,17s/60° při 4,8V
- Hmotnost - 8g
- Rozměry - 22,5x12,8x23,7mm

6.2.2 Úprava pro kontinuální otáčení

Úprava serva vyžaduje zrušení zpětné vazby, kterou představuje potenciometr a odstranění mechanického dorazu bránícího rotoru v protáčení. Samotná demontáž servomotoru není nijak složitá operace, pokud máme k dispozici správný šroubovák na miniaturní šroubky.



Obrázek 12. Rozdělaný servomotor (nejedná se o VTS-05A)

6.2.2.1 Odstranění zpětné vazby

První možností, jak zrušit zpětnou vazbu sledující natočení rotoru, je odstranění potenciometru. Místo něj se zapojí odporový dělič (dva stejné rezistory v sérii o doporučené hodnotě 2k7). Druhá varianta spočívá v ponechání potenciometru, ale je potřeba mechanicky zajistit jeho střední polohu. K tomu může posloužit lepidlo, jistější je však zatavení. Při zajišťování se musí dát pozor na to, aby se jezdec při samotném úkonu nepohnul. Následně se ještě musí nějakým způsobem zlikvidovat mechanický kontakt potenciometru, jenž je spojený s ozubeným kolem. V tomto případě záleží na druhu koupěného serva, ale někdy postačí i obyčejné štípačky s pilníkem.

6.2.2.2 Odstranění mechanického dorazu

Mechanický doraz, bránící natočení o více než 180°, se nachází většinou na výstupním ozubeném kole serva, který se kontaktuje s dorazem na horním víku servomotoru. Můžeme si vybrat, který výstupek odstraníme. Tuto drobnou umělohmotnou část je možné odstranit buď ostrým nožičkem, nebo miniaturní bruskou.

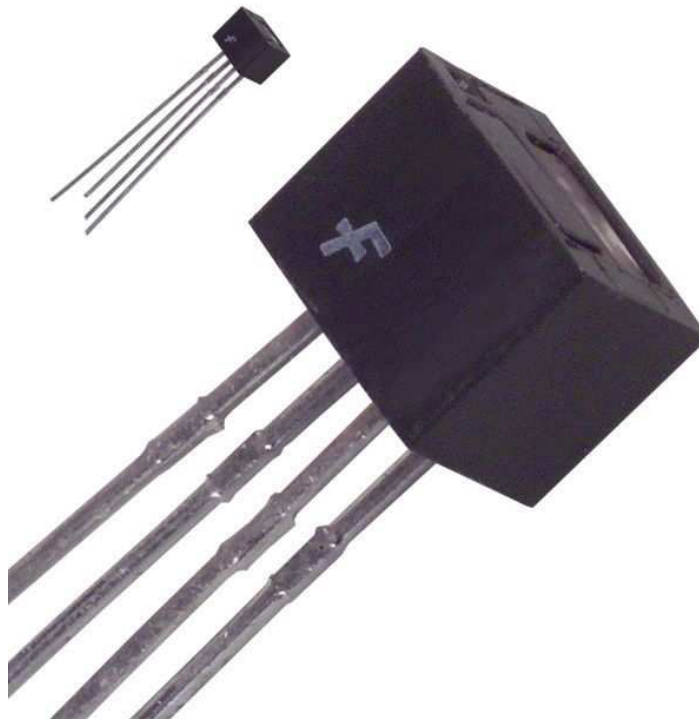
6.2.2.3 Ovládání upraveného serva

Po dokončení úpravy je možné zase všechno poskládat dohromady. Dbát by se mělo na správné uložení převodů a neuškodí ani mírné promazání například vazelínou. Pokud byl takto provedený zásah úspěšný, servo je nyní nastavené pro kontinuální otáčení. Jeho ovládání přes PWM zůstává stále stejné, rozdíl však nastává v chování serva.

Nyní můžeme ovládat plynule otáčky serva, včetně směru otáčení. Zatímco předtím znamenala délka pulzu 1,5 ms natočení ozubeného kola do střední polohy, nyní je to klidový stav, kdy se servomotor netočí. Za předpokladu, že budeme délku pulzu zvyšovat v intervalu od 1,5 do 2ms, dosáhneme otáčení rotoru směrem doprava. Čím vyšší bude délka pulzu, tím vyšších otáček dosáhneme. Interval od 1 do 1,5ms pak znamená otáčení rotoru doleva, kdy při 1ms dosáhneme maximální rychlosti.

6.3 Senzor QRD1114

Součástka spadající do kategorie odrazových infrasenzorů. Svými parametry a svojí konstrukcí je vhodná právě pro roboty, mající za úkol sledovat čáru. Senzor tvoří IR LED dioda a fototranzistor, jenž jsou společně v umělohmotném krytu.



Obrázek 13. Senzor QRD1114

Parametry

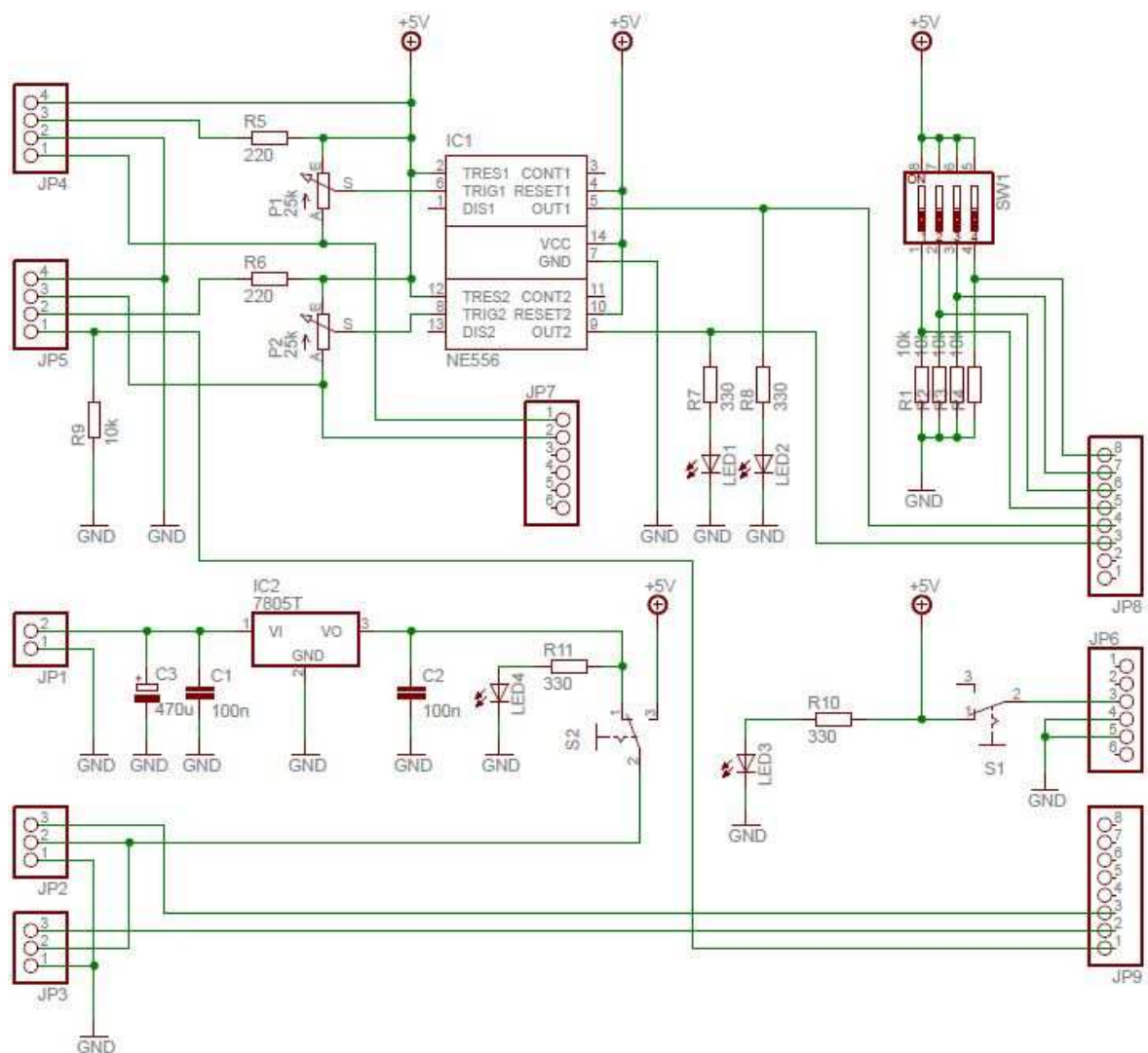
- Detekce na vzdálenost - $< 6\text{mm}$
- Optimální vzdálenost od povrchu - 0,25 až 1,5mm
- Rozměry - 4,4x6,1x4,7mm

Podrobnější parametry obsahuje datasheet [16].

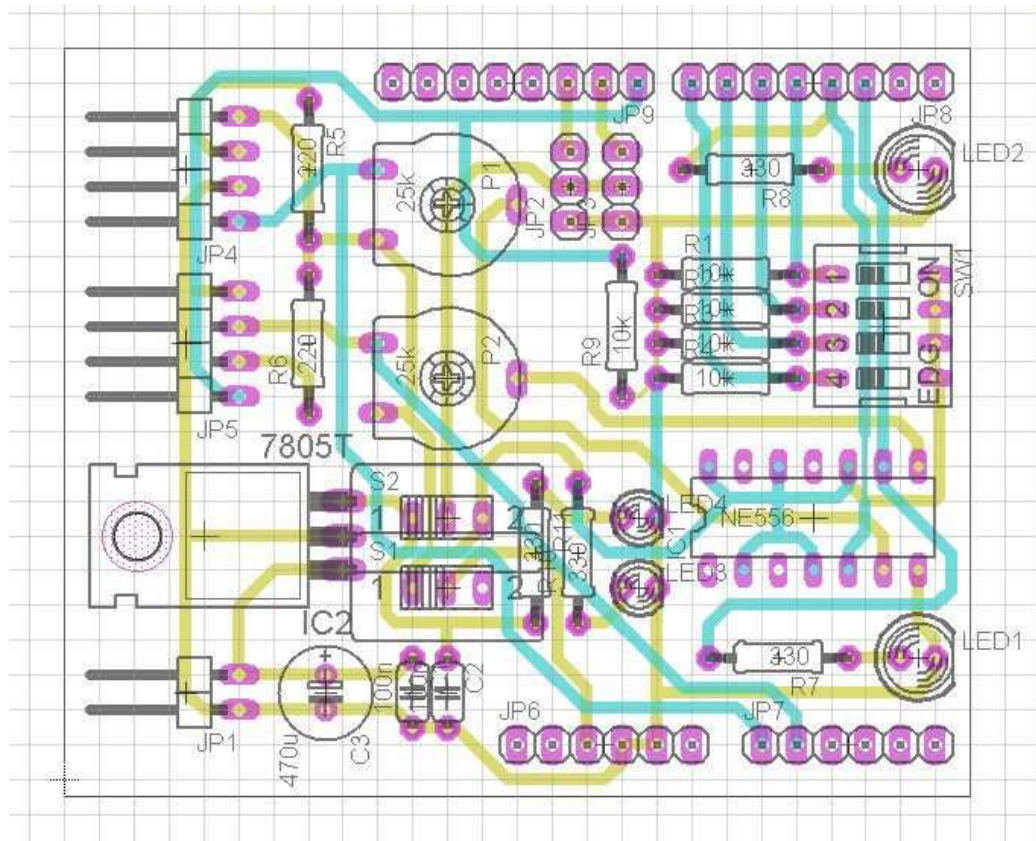
7 ELEKTRONICKÉ ZAPOJENÍ

Kapitola obsahuje schéma zapojení pro zásuvný modul a desky, na které jsou umístěny senzory. Schémata byly kresleny v programu Eagle. Součástí kapitoly je rovněž popis důležitých částí zapojení.

Zapojení zásuvného modulu:



Obrázek 14. Schéma zapojení zásuvného modulu



Obrázek 15. Navržená deska modulu v programu Eagle

7.1 Popis konektorů

Modul disponuje devíti konektory o různém počtu pinů. Ty slouží pro komunikaci s kitem, senzory, servy a napájením.

- JP1 – přídavné napájení na provoz servomotorů
- JP2 – servo pro pravé kolo
- JP3 – servo pro levé kolo
- JP4 – senzor monitorující vybočení doleva
- JP5 – senzor monitorující vybočení doprava
- JP6 – napájecí část kitu
- JP7 – A/D převodník kitu
- JP8 – digitální I/O porty kitu
- JP9 – digitální I/O porty kitu + PWM

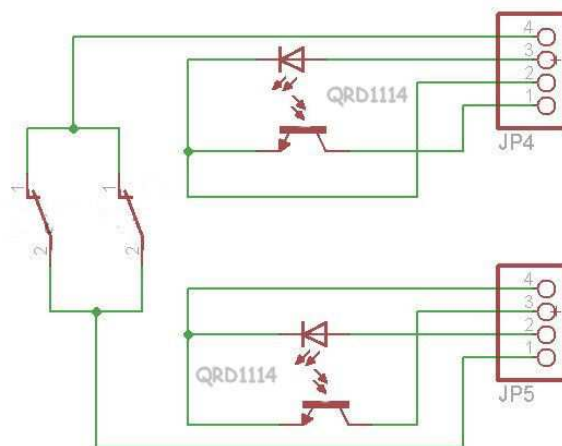
7.2 Napájení robota

Arduino Uno je napájeno jednou 9V alkalickou baterií. Napojený zásuvný modul tak odebírá napětí přímo z kytu přes konektor JP6, jeho hodnota je 5V. Přes spínač S1 je možné modul vypnout nebo zapnout. Signalizace zapnutí je indikována pomocí zelené LED diody LED3.

Sevomotory mohou být napájeny buď z modulu, kdy se jedná o napětí brané z kytu Arduino Uno, nebo z přídatné druhé 9V baterie. V prvním případě je celý robot poháněn jen jednou baterií a jedná se o poměrně velký proudový odběr. Robot by moc dlouho v chodu nevydržel. Proto je lepší, aby serva byly napájené vlastní baterií, jejíž napětí je pomocí vhodného zapojení stabilizátoru L7805CV, stabilizované na 5V. Jestli chceme použít společné nebo oddělené napájení serv, můžeme ovlivnit přepínačem S2. Pokud je připojena přídatná 9V baterie, žlutá LED dioda LED4 trvale svítí.

7.3 Vyhodnocování signálů ze senzorů

Dotykové i optické senzory jsou umístěny na vlastní desce a konektory JP4 a JP5 propojeny se zásuvným modulem.



Obrázek 16. Schéma zapojení desky obsahující senzory

7.3.1 Dotykové senzory

Díky dotykovým sensorům, je robot schopný rozpoznat čelní náraz do překážky nebo zdi. Senzor tvoří dva paralelně zapojené mikrosplínače, které jsou napojené na digitální I/O

port JP9. Pokud není některý z mikrospínačů stisknutý, na pinu 1 je log. 0, uzemněná přes 10k rezistor. Při nárazu na překážku se na pin 1 přes mikrospínač dostane napětí 5V a dostáváme tak log. 1. Tyto stavy vyhodnocuje následně samotný program, který monitoruje tento pin.

7.3.2 Optické senzory

7.3.2.1 Digitální vyhodnocování

IR senzory, mající za úkol sledovat černou čáru, využívají pro svůj chod časovač NE556N, který pracuje jako komparátor. Za předpokladu, že využijeme funkční tabulku časovače (Obrázek 10), můžeme na výstupu časovače očekávat log. 1 nebo log. 0 v závislosti na tom, jestli se aktuální čidlo nachází nad černou čarou či nikoliv.

Podmínky pro vyhodnocení odrazu IR světla

- **OUT = LOW:** RESET = HIGH, TRIGGER > $1/3 V_{cc}$, THRESHOLD > $2/3 V_{cc}$
- **OUT = HIGH:** RESET = HIGH, TRIGGER < $1/3 V_{cc}$, THRESHOLD > $2/3 V_{cc}$

Optický senzor infračervené světlo od černého povrchu neodráží, takže téměř žádné světlo na fototranzistor nedopadá a protéká jím minimální proud. V tomto případě je většinou napětí na jezdcí trimru větší než $1/3 V_{cc}$ (což je třetina z 5V). Toto napětí přivedené na pin TRIG je komparátorem porovnáváno s napájecím napětím a poté je na výstupu OUT log. 0. Při vyjetí z dráhy se IR světlo odrazí a vnitřní odpor fototranzistoru se zmenší, což způsobí pokles napětí na jezdcí trimru, který je přivedený na pin TRIG. Pokud na něm bude napětí menší než $1/3 V_{cc}$, na výstupu OUT bude log. 1. Červená signalizační LED dioda se rozsvítí a mikroprocesor má možnost zaznamenat vyjetí z dráhy.

Dvojitý časovač má dva výstupy – OUT1 a OUT2. Každý sleduje jedno čidlo, aby bylo možné rozhodnout, zda robot vyjel zprava nebo zleva. Výstupy OUT jsou přivedeny na konektor JP8.

Fototranzistor je do jisté míry háklivý i na okolní osvětlení. Aby mohl komparátor správně rozhodnout o odrazu, je potřeba provést kalibrace. K tomu slouží trimry. Pro zpětnou vazbu, zda jsme citlivost správně nastavili, slouží signalizační LED diody LED1 a LED2.

7.3.2.2 Analogové vyhodnocování

U každého IR senzoru je navíc ještě vyveden kolektor fototranzistoru na konektor JP7. Odtud je možné pomocí A/D převodníku, který mikroprocesor obsahuje, vyhodnotit míru dopadajícího světla na fototranzistor.

7.4 DIL spínač

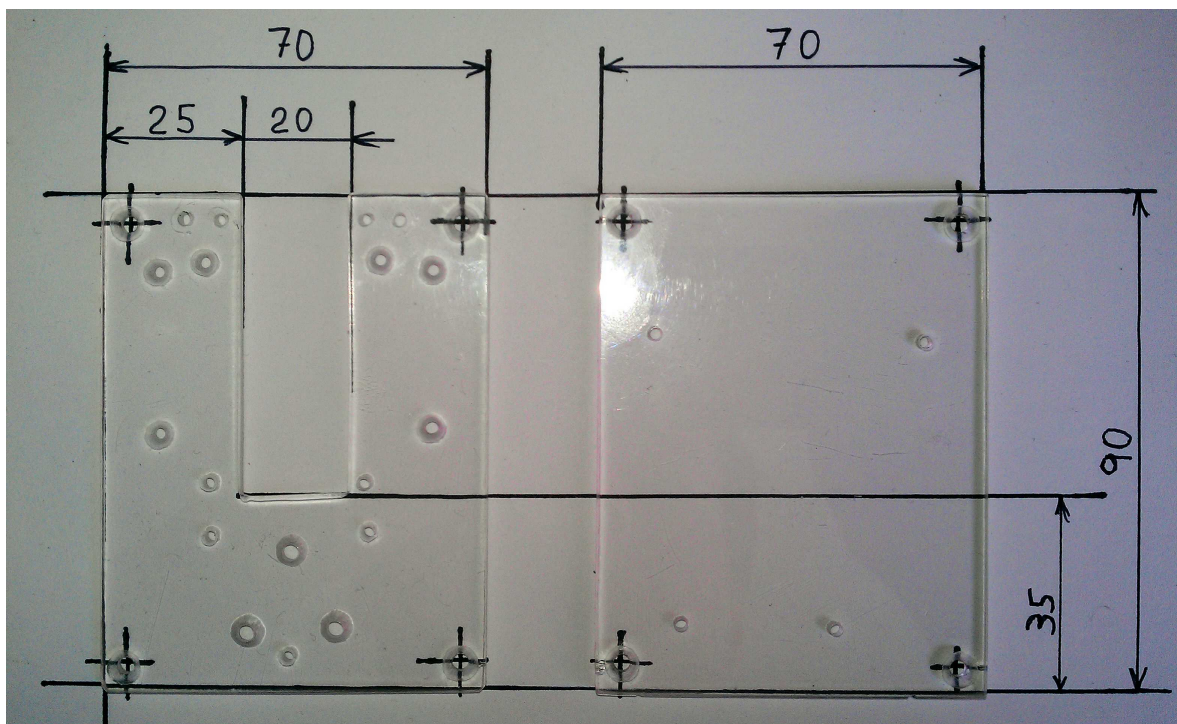
Součástí modulu je i SDIP04, který má v sobě zabudované 4 spínače. Jejich sepnutí/rozepnutí pak odpovídá tomu, jestli se na pinech konektoru JP8 objeví log. 1 nebo log 0. Tyto stavy je možné programem libovolně vyhodnocovat a zajistit tak různé režimy pro chod robota.

8 MECHANICKÉ ZPRACOVÁNÍ

Při návrhu robota bylo potřeba vycházet především z rozměrů kitu Arduino Uno. Ten musí být navíc dobře dostupný, aby na něj bylo možné napojit vyrobený modul a vyvést konektory na příslušné periferie, které tvoří robota. Důležitý je taktéž prostor pro baterie, protože ty zaberou podstatnou část místa. Snažil jsem se o to, aby samotná konstrukce měla co možná nejmenší rozměry. To ovlivnilo také výběr servomotorků, protože prioritou byla malá velikost.

8.1 Tělo robota

Základnu robota tvoří čiré plexisklo o rozměrech 90x70x3mm. Svým tvarem připomíná písmeno U. Výsek široký 20mm a zapuštěný 55mm je nezbytný pro senzory. Ty potřebují bezproblémový přístup k povrchu, po kterém se bude robot pohybovat. Na spodní stranu základny jsou přišroubovány servomotorky společně se všesměrovou kuličkou, která zajišťuje stabilitu při pohybu. Z druhé strany je umístěn tištěný spoj se senzory a také dva doky na devíti voltové baterie.



Obrázek 17. Dvě plexiskla tvořící tělo robota

K základně je pomocí čtyř distančních sloupků (25mm) přichycená horní část. Ta je tvořena rovněž čirým plexisklem o rozměrech 90x70x3mm. Na rozdíl od základny má ale relativně málo děr, protože slouží pouze k uchycení vývojového kitu.

8.2 Podvozek robota

Na podvozek jsou přichyceny dvě serva VTS-05A a všesměrové kolečko TM70114 pro zajištění stability. Servomotorky bylo nejprve potřeba upravit na kontinuální otáčení. Robot bude využívat diferenciální řízení pohybu.

Aby bylo možné serva na plexisklo vůbec přichytit, bylo zapotřebí vyrobit čtyři přesné miniaturní uhlíky o rozměrech 6x10mm. Teprve potom se dají serva pomocí šroubků namontovat.

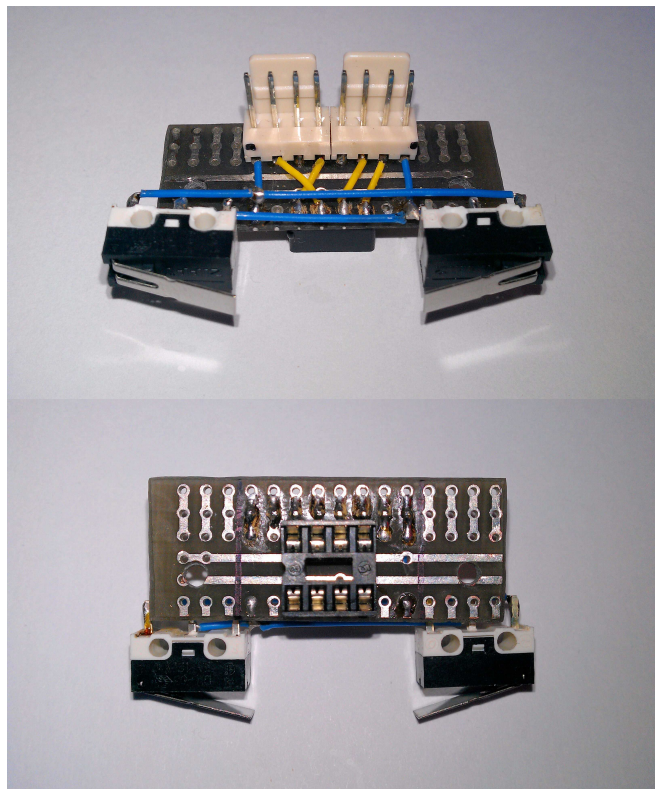


Obrázek 18. Serva se všesměrovým kolečkem připravené k namontování

Na výstupní ozubené kolo každého servomotorku je přidělaná umělohmotná násada, ke které je rozleptáním obou umělých hmot přichycené černé kolečko Pololu1087. Kolečko má po obvodu gumový povrch, který zabraňuje prokluzování.

8.3 Uchycení senzorů

Robot je vybaven dotykovými a bezdotykovými senzory, které jsou v párech přichyceny na tištěnou desku. Ta je vystřižená z většího univerzálního plošného spoje a má rozměry 38x16mm. Propojení senzorů s kontaktními vidlicemi je tak zajištěno pomocí tenkých drátů.



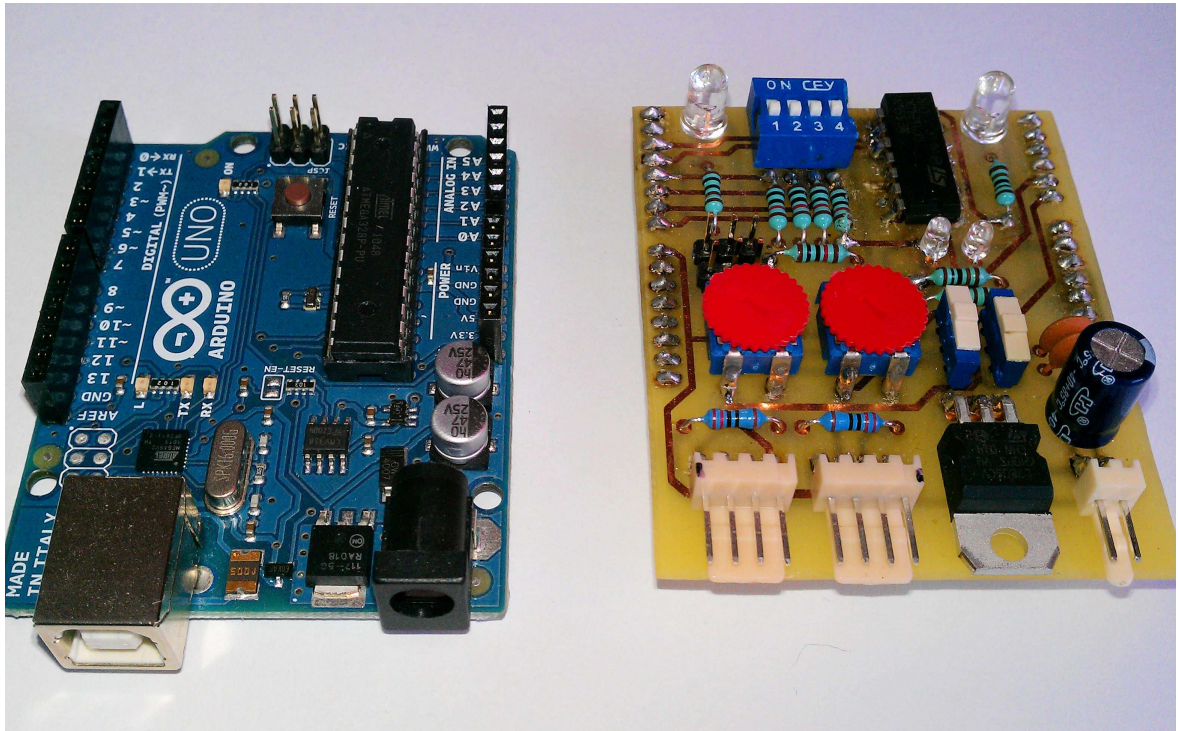
Obrázek 19. Osazený plošný spoj pro zajištění senzorů

Zatímco mechanické senzory tvořeny mikrospínači jsou napájené k tištěné desce napevno, optické IR senzory je možné kdykoliv zakontaktovat do patice PAT08 DIL ST.

Každý mikrospínač je k plexisklu přidělán pomocí dvou tenkých šroubků. Samotný osazený tištěný spoj je potom přidělán také dvěma šroubky. Díky výseku v plexiskle je patice dobře přístupná a není problém senzory QRD1114 zapojit tak, aby měly dobrý přístup k snímané čáře.

8.4 Zásuvný modul pro Arduino Uno

Navržený modul je nezbytnou součástí robota. Výrazně usnadňuje komunikaci mikropočítače se servomotory a senzory. Zároveň se také stará o vlastní napájení serv a umožňuje kalibrovat citlivost senzorů v závislosti na okolním osvětlení.



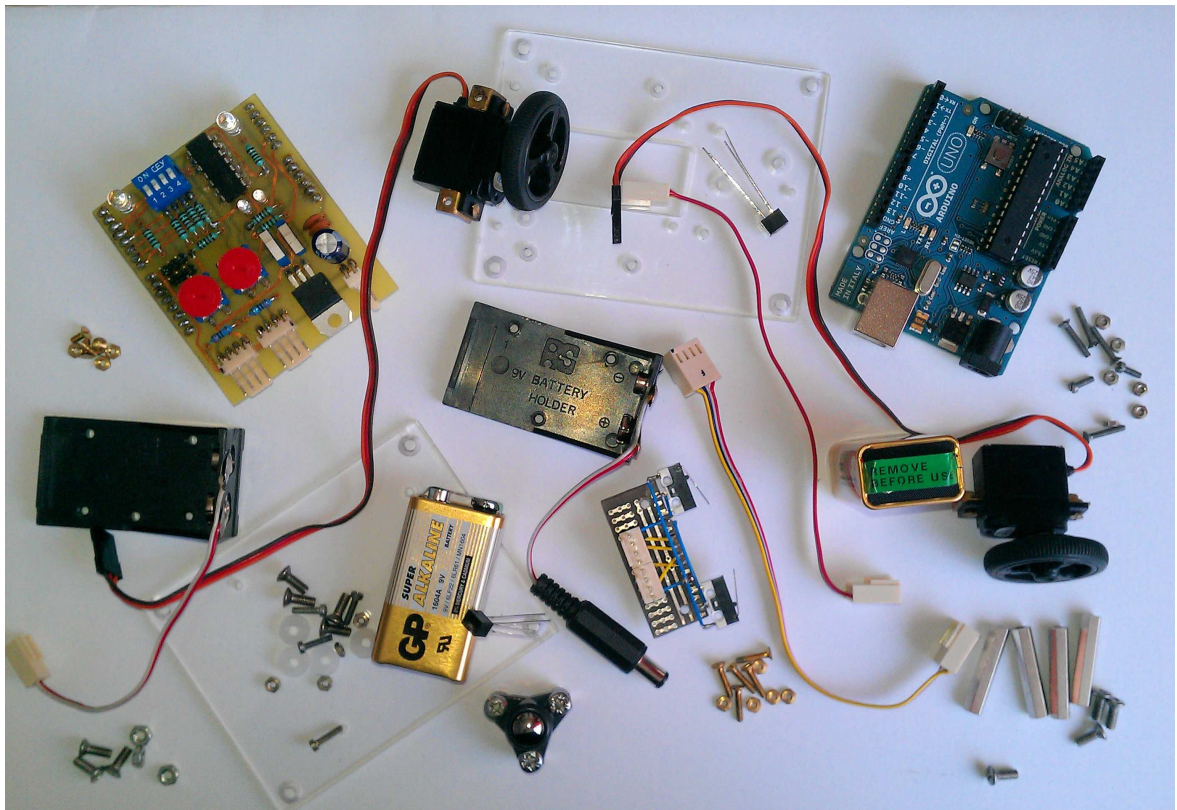
Obrázek 20. Modul v porovnání s Arduino Uno

Modul má na spodní části stejnou rozteč pinů, jako vývojový kit, což zajišťuje bezproblémové a pevné kontaktování. Na straně druhé jsou dvě čtyřpinové zahnuté vidlice, které jsou propojeny pomocí vyrobených kablíčků s deskou, na niž jsou umístěny senzory. Na zbývající dvoupinovou vidlici je přivedeno napájení z devíti voltové baterie. K modulu se také připojují obě dvě serva, jež jsou napojeny na piny, umístěné kousek od trimrů.

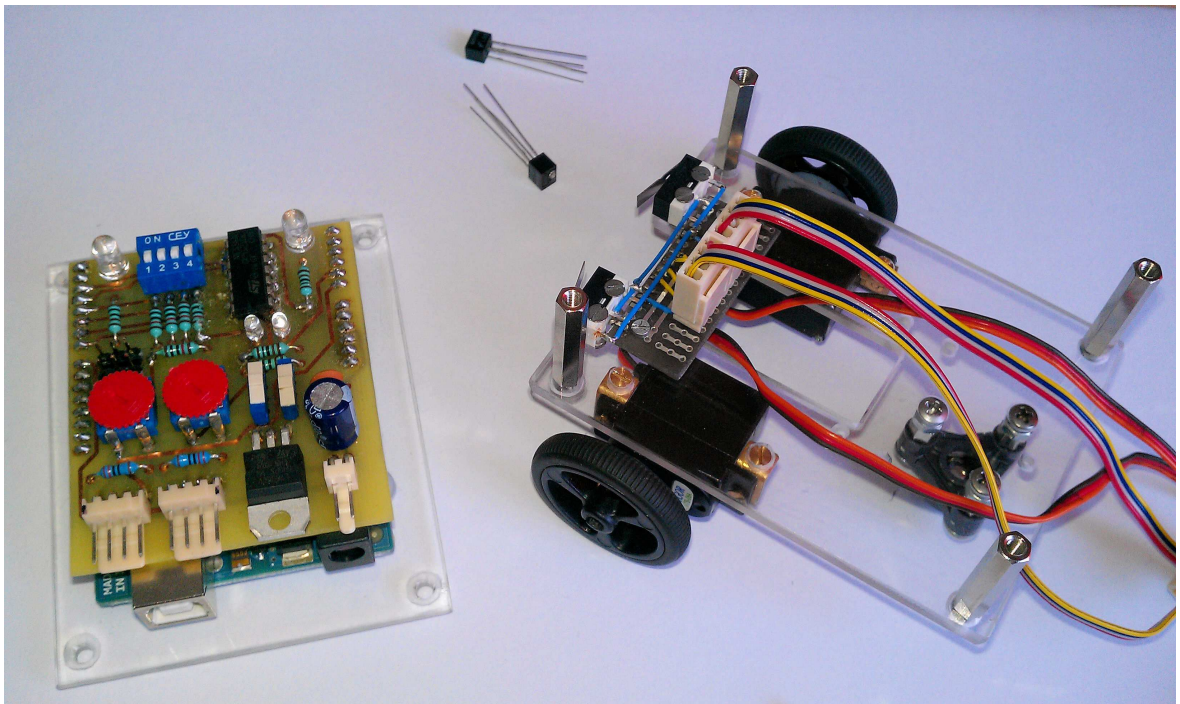
Oboustranný tištěný spoj byl navrhován v programu Eagle. Po následném osvětlení UV zářením a vyleptání spojů, bylo možné desku osadit součástkami. Jediný problém byl, že se na amatérské úrovni nedají zhotovit prokovené díry, a proto bylo následné pájení některých součástek komplikované.

8.5 Složení robota

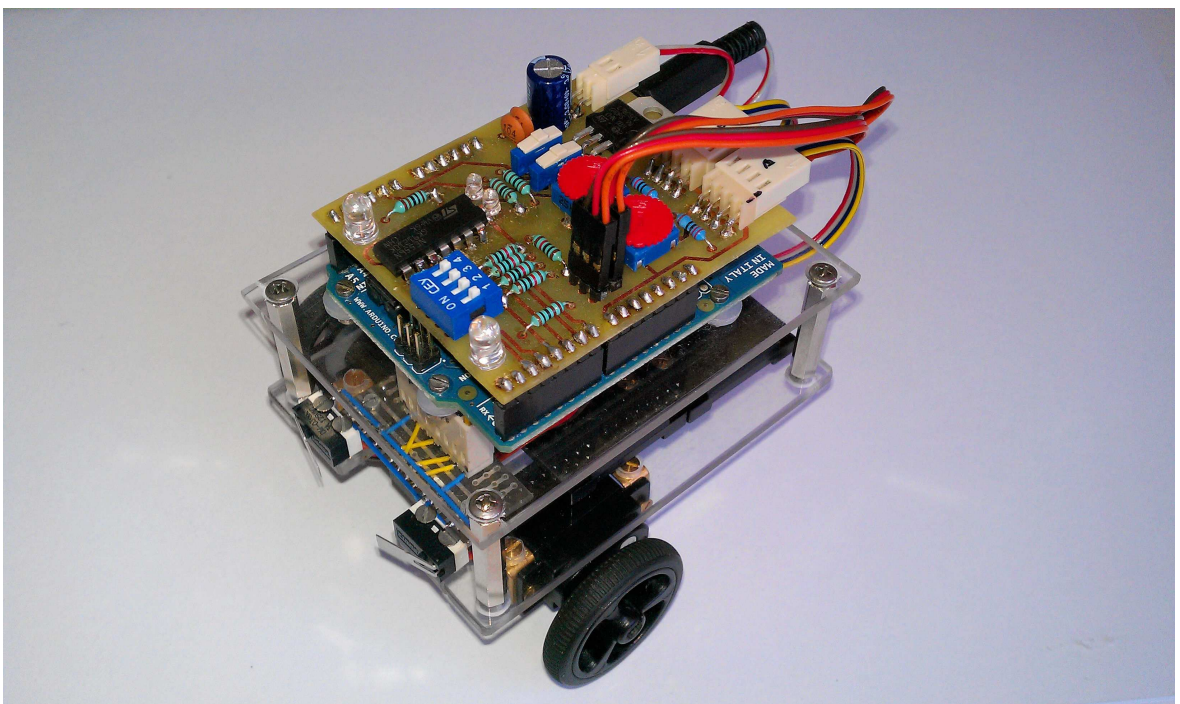
Po zhotovení a zakoupení všech nezbytných věcí, je možné robota konečně poskládat dohromady. Většina dílů je spojená pomocí šroubků a maticek. To zajišťuje pevnou a odolnou konstrukci.



Obrázek 21. Všechny potřebné díly na jednom místě



Obrázek 22. Postupné skládání robota



Obrázek 23. Hotový a sestavený robot

Robot po sestavení ještě není schopný pohybu. Nezbytnou součástí je program robota, kterým se bude řídit a černá čára na bílém podkladu, po které bude jezdit. Tu bude tvořit černá matná izolepa na výkresovém papíru.

9 SOFTWAREVÉ VYBAVENÍ

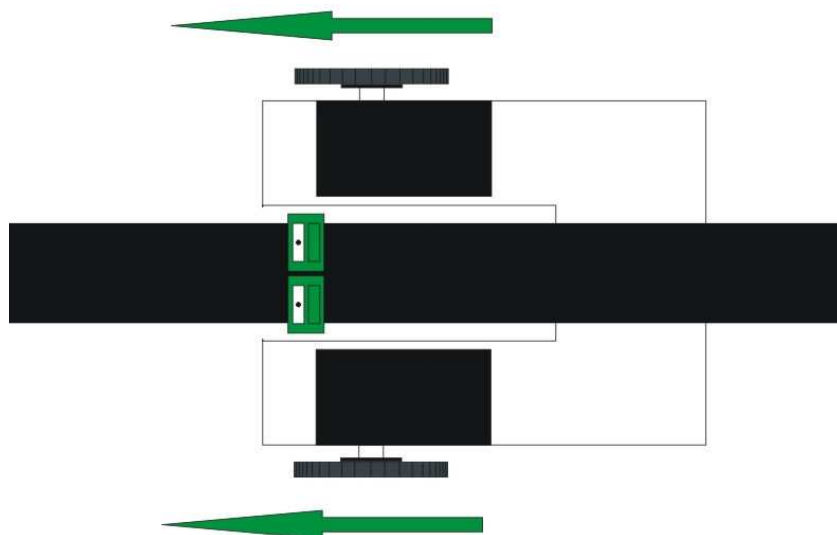
Robot, pohybující se po černé čáře, by měl v ideálním případě docílit takového stavu, aby měl nad čarou vždy minimálně jeden senzor. Efektivita projetí vyznačené dráhy je stoprocentní tehdy, když strávený čas senzorů nad čarou podělíme časem, za který robot projede dráhu a výjde nám číslo jedna. Kvalita robota je potom dána maximální rychlostí, v které je schopný se této podmínce přiblížit. Velkou roli hraje celková konstrukce robota, kvalita použitých součástek, návrh ovládání a především program, který bude robota řídit.

9.1 Sledování čáry

Při sledování čáry mohou nastat tři různé situace. Sledování rovné čáry, kdy se obě serva točí stejnou rychlostí, pravotočivá nebo levotočivá zatáčka a vyjetí robota z dráhy.

9.1.1 Rovný úsek

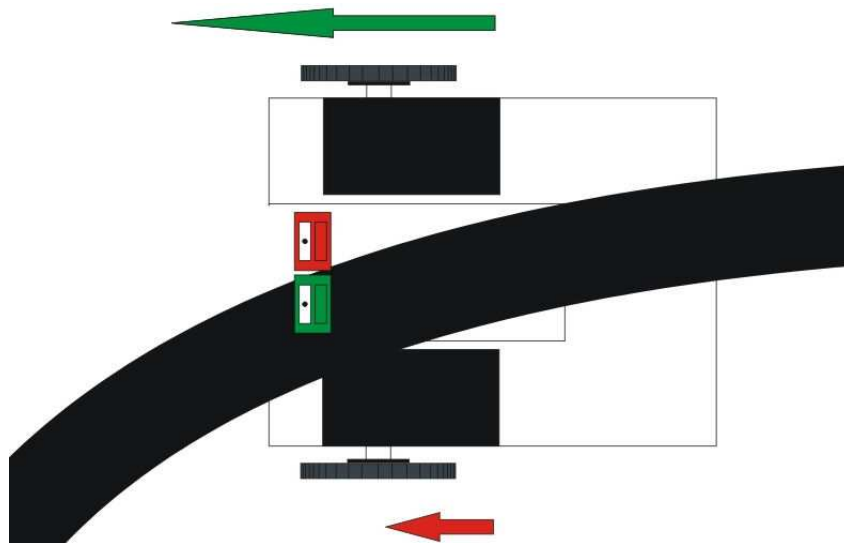
Pokud se robot pohybuje po rovné čáře, oba senzory nezaznamenávají odraz a serva se tak točí stejnou rychlostí. To je ideální případ, v praxi může dojít ovšem k mírné kolizi, kdy se robot potřebuje i na rovném úseku občas vyrovnat.



Obrázek 24. Sledování rovné čáry

9.1.2 Zatáčka

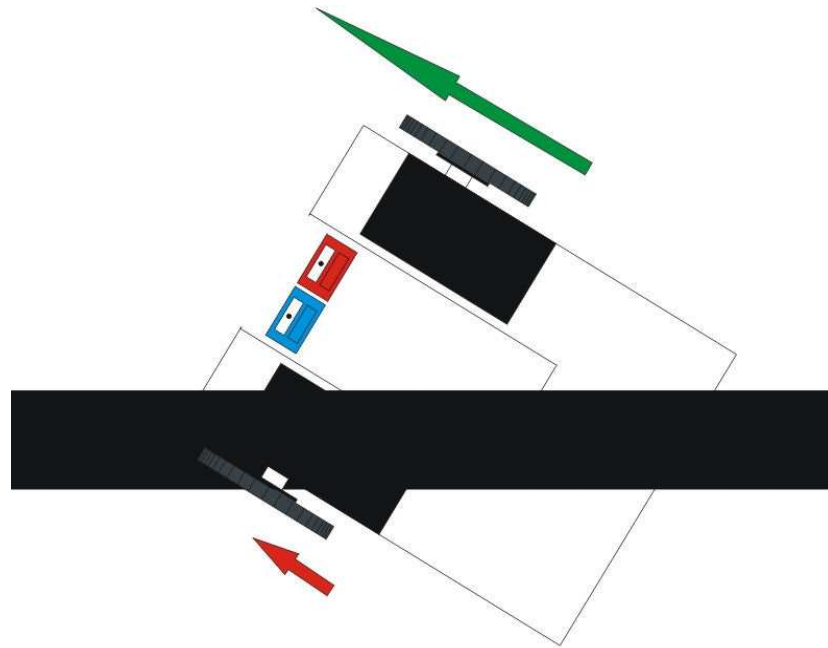
V případě, že robot vjede do pravotočivé nebo levotočivé zatáčky, jeden senzor opustí černou čáru. Senzor zaznamená odraz, který mikropočítač vyhodnotí a následně jedno servo zpomalí, nebo na chvíli úplně vyřadí z provozu – záleží na charakteru zatáčky. Program se snaží dostat robota do vyváženého stavu, aby ani jeden senzor nehlásil odraz.



Obrázek 25. Najetí do levotočivé zatáčky

9.1.3 Vyjetí robota z dráhy

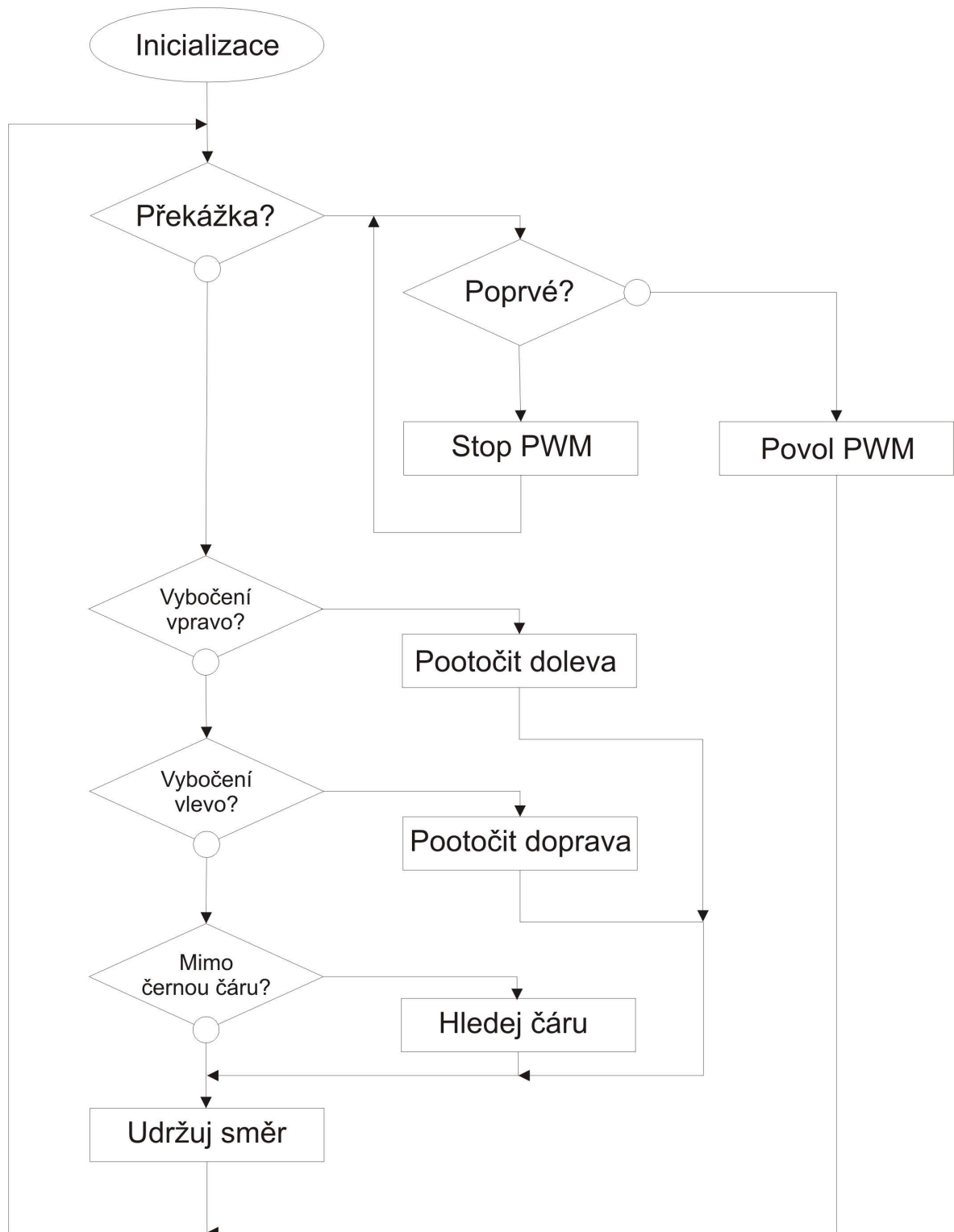
Jedná se o situaci, kterou by ideální robot neměl vůbec řešit. Bohužel v praxi tato situace není až tak ojedinělá, a proto je potřeba ji ošetřit. Do této fáze se může robot dostat ze tří různých příčin. Pokud je rychlost robota neúměrně rychlá blížící se zatáčce, může se stát, že program nestihne zkorigovat otáčky serv a oba senzory vyjedou z dráhy. Dalším případem je projíždění atypické zatáčky o malém průměru. Pokud konstrukce robota brání tuto zatáčku opsat, opět se dostanou oba senzory mimo dráhu. Třetím důvodem je okolní osvětlení. Senzory sice jdou kalibrovat, ale není problém, aby se robot po dráze dostal do míst, kde panují jiné světelné podmínky. Senzory pak nejsou schopny pohotově nebo spolehlivě detekovat odraz a robot tak sjede z cesty. V takových případech si program musí pamatovat, které čidlo opustilo dráhu jako poslední a tím směrem se vydat. Postupně projíždí určitou plochu, dokud nenatrefí zase na čáru.



Obrázek 26. Vybočení z dráhy

9.2 Hlavní program

Po příslušné inicializaci program nejdříve testuje, zda nenarazil na překážku. Pokud ano, robot se zastaví a dál ve své jízdě nepokračuje. Do provozu ho přivede opětovné stisknutí mikrospínače. Dále program neustále zjišťuje, jestli nevybočil z dráhy nebo neztratil kontakt s černou čarou. Pokud některá ze situací nastane, dojde k příslušnému vybavení instrukcí, majících za úkol robota správně nasměrovat. Jestliže jede robot po rovné čáře, stačí udržovat pouze aktuální směr. Celý program je ve smyčce, protože je potřeba neustále kontrolovat a korigovat směr jízdy robota.



Obrázek 27. Vývojový diagram hlavního programu

ZÁVĚR

Cílem této bakalářské práce bylo navrhnout a zkonstruovat funkčního robota, jenž má za úkol sledovat černou čáru. Nejprve bylo potřeba vybrat mikrokontrolér, který bude robota řídit. K těmto účelům se hodil ATmega328 od firmy Atmel, který je dodáváný i s vývojovým kitem Arduino Uno, jehož rozměry a specifické vlastnosti odpovídaly mým požadavkům. Nezbytnou součástí jsou dva servomotory, zajišťující pohyb robota. Serva VTS-05A byly vybrány díky své malé velikosti. Mechanický návrh robota se tak musel přizpůsobit právě velikostem serv a desce Arduino Uno. Ohled se musel brát i na umístění baterií a na infračervené senzory, jenž potřebují přístup k povrchu, po kterém se robot pohybuje.

Aby mohl robot fungovat, bylo zapotřebí vyrobit také rozšiřující zařízení ve formě zásuvného modulu, které musí být kompatibilní s kitem Arduino Uno. Díky němu může mikroprocesor přes příslušné porty snadněji ovládat serva a vyhodnocovat signál přicházející ze senzorů. Nezbytná je také samotná kalibrace senzorů, kterou modul zajišťuje. Tištěná deska modulu byla navržena v programu Eagle.

Po zapojení a následném vytvoření softwarového vybavení je robot schopný rozpoznat černou čáru na světlém povrchu a pohybovat se po ní jako po jízdni dráze. Takto vytvořený funkční robot dosahuje maximální rychlosti 0,1 m/s. Rychlost je omezená především konstrukcí serv, jenž jsou původně určeny pro polohové natáčení v rozmezí 0 až 180 stupňů. Aby se mohly serva kontinuálně otáčet, bylo zapotřebí zásahu do vnitřního zapojení.

Při výrobě robota bylo časově náročné především mechanické zpracování včetně návržení a výroby modulu. Také potřebné testování zabralo nějaký čas. Až na menší problémy s časovačem NE556N, mající na starost monitorování senzorů, se při vytváření robota nevyskytly žádné problémy.

Díky této bakalářské práci jsem si přímo v praxi vyzkoušel zacházení s mikropočítačem a také si rozšířil obzor v oblasti robotiky.

Součástí je přiložené CD, na kterém se nachází text bakalářské práce a zdrojový kód, potřebný pro funkční chod robota.

ZÁVĚR V ANGLIČTINĚ

The aim of this thesis was to design and construct a functional robot which is supposed to follow a black line. At first, there was a need to choose a proper microcontroller which will control the robot. For these purposes was most suitable the ATmega328 from Atmel which comes with development kit Arduino Uno. Its parameters and specific characteristics matched the requirements. An essential part is the two servo-motors VTS-05A providing mobility. These servo-motors were selected because of their small size. Mechanical design of the robot had to adapt to the size of servo-motors and Arduino Uno motherboard. Also the location of the battery and infrared sensors (which need access to the surface after which the robot moves) had to be taken into consideration.

For proper functionality of the robot was required to manufacture an expansion device in form of the plug-in module which had to be compatible with the Arduino Uno kit and which allows the microprocessor to control servo-motors and evaluate the signal coming from the servo-motors more easily. Module also provides necessary calibration of sensors. The printed circuit board for module was designed in software Eagle.

After the proper wiring and programming the robot is able to detect a black line on a bright surface and follow it. Functional robot reaches a maximum speed of 0.1 m/s. This speed is especially limited by the construction of servo-motors which are originally designed for positioning between 0 and 180 degrees. In order to allow the servo-motors to rotate continuously the internal construction had to be adjusted.

The most time-consuming part was especially mechanical fabrication including design and module manufacturing. Necessary testing also took some time. Aside from minor problems with the NE556N timer (which is monitoring the infrared sensors), there were no further troubles with construction or programming.

Due to this thesis I've experienced practical work with microcomputer and also improved my knowledge in the field of robotics.

Enclosed CD contains text of this thesis and a source code which is essential for proper functionality of the robot.

SEZNAM POUŽITÉ LITERATURY

- [1] PINKER, Jiří. *Mikroprocesory a Mikropočítače*. Praha : BEN – technická literatura, 2004. 220 s. ISBN 80-7300-110-1.
- [2] MANN, Burkhard. *C pro mikrokontroléry*. Praha : BEN - technická literatura, 2004. 280 s. ISBN 80-7300-077-6.
- [3] CATSOULIS, John. *Designing Embedded Hardware*. O'Reilly Media, 2005. 400 s. ISBN 978-0-596-00755-3.
- [4] ATmega8 Datasheet [online]. Atmel Corporation, 2009 [cit. 2011-01-17]. Dostupný z WWW: <http://www.atmel.com/atmel/acrobat/doc2486.pdf>
- [5] SKALICKÝ, Petr. *Mikroprocesory řady 8051*. Praha : BEN – technická literatura, 2002. 159 s. ISBN 80-86056-39-2.
- [6] NOVÁK, Petr. *Mobilní roboty*. Praha : BEN – technická literatura, 2005. 256 s. ISBN 80-7300-141-1.
- [7] KARGER, Adolf. KARGEROVÁ, Marie. *Základy robotiky a prostorové kinematiky*. Praha : BEN – ČVUT, 2008. 265 s. ISBN 80-01-02183-1
- [8] Wikipedie: Robot [online]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Robot>.
- [9] Wikipedie: Jednočipový počítač [online]. Dostupné z WWW: http://cs.wikipedia.org/wiki/Jednočipový_počítač.
- [10] Mikropočítače [online]. Dostupné z WWW: <http://lucy.troja.mff.cuni.cz/~tichy/elektronika/kap9/mikropo.html>.
- [11] Oficiální stránky Arduino [online]. Dostupné z WWW: <http://arduino.cc/>.
- [12] ARUINO: Vývojový kit pro hrátky s hardware [online]. Dostupné z WWW: <http://www.root.cz/clanky/arduino-vyvojovy-kit-pro-hratky-s-hardware/>.
- [13] Řízení serva – teorie [online]. Dostupné z WWW: <http://www.serva.cz/rizeni-serva-teorie/>.
- [14] Úprava serva na 360° otáčení [online]. Dostupné z WWW: <http://www.serva.cz/uprava%20serva%20na%20360%C2%B0%20otaceni/>.

- [15] Datasheet: NE556, SA556, SE556 DUAL PRECISION TIMERS [online].
Dostupné z WWW:
<http://www.datasheetcatalog.org/datasheet2/6/0eiwi9tqtr4e4tcjw5xdzdrh0wy.pdf>
- [16] Datasheet: QRD1113, QRD1114 Reflective Object Sensor [online]. Dostupé z
WWW: < <http://www.fairchildsemi.com/ds/QR/QRD1114.pdf> >.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

MCU	Jednočipový počítač (Multipoint Control Unit)
RAM	Paměť s libovolným přístupem (Random Access Memory)
ROM	Paměť pouze pro čtení (Read Only Memory)
PROM	Elektricky jednorázově programovatelná paměť typu ROM
EPROM	Programovatelná (UV světlem mazatelná) paměť typu ROM-RAM
EEPROM	Elektricky mazatelná, programovatelná paměť typu ROM-RAM
A/D	Analogově digitální
D/A	Digitálně analogový
PWM	Pulsně šířková modulace (Pulse Width Modulation)
CPU	Procesor (Central Processing Unit)
ALU	Aritmeticko-logická jednotka (Arithmetic Logic Unit)
CISC	Complex Instruction Set Computer
RISC	Reduced Instruction Set Computer
I/O	Vstup/Výstup (Input/Output)
AVR	Označení Atmelovské architektury mikrokontrolerů
PC	Osobní počítač (Personal Computer)
USB	Univerzální sériová sběrnice (Universal Serial Bus)
GND	Elektrické uzemnění (Ground)
IR	Infračervené záření (Infrared)
LED	Elektroluminiscenční dioda (Light Emitted Diode)
TTL	Tranzistorově tranzistorová logika (Transistor Transistor Logic)
UV	Ultrafialové záření (Ultraviolet)

SEZNAM OBRÁZKŮ

<i>Obrázek 1. Mikrokontrolér řady PIC.....</i>	13
<i>Obrázek 2. Obecné blokové schéma mikrokontroléru</i>	14
<i>Obrázek 3. Vývojový kit Arduino Uno</i>	16
<i>Obrázek 4. Vývojové prostředí pro Arduino</i>	18
<i>Obrázek 5. Servomotor a blokové schéma řízení.....</i>	19
<i>Obrázek 6. Možné průběhy řídicího PWM signálu.....</i>	20
<i>Obrázek 7. Princip odrazu IR světla u QRD1114.....</i>	21
<i>Obrázek 8. Časovač NE556N</i>	23
<i>Obrázek 9. Rozmístění pinů obvodu NE556N.....</i>	23
<i>Obrázek 10. Funkční tabulka časovače NE556N</i>	24
<i>Obrázek 11. VIGOR VTS-05A.....</i>	24
<i>Obrázek 12. Rozdělaný servomotor (nejedná se o VTS-05A).....</i>	25
<i>Obrázek 13. Senzor QRD1114</i>	27
<i>Obrázek 14. Schéma zapojení zásuvného modulu</i>	28
<i>Obrázek 15. Navržená deska modulu v programu Eagle</i>	29
<i>Obrázek 16. Schéma zapojení desky obsahující senzory</i>	30
<i>Obrázek 17. Dvě plexiskla tvořící tělo robota</i>	33
<i>Obrázek 18. Serva se všesměrovým kolečkem připravené k namontování</i>	34
<i>Obrázek 19. Osazený plošný spoj pro zajištění senzorů.....</i>	35
<i>Obrázek 20. Modul v porovnání s Arduino Uno.....</i>	36
<i>Obrázek 21. Všechny potřebné díly na jednom místě</i>	37
<i>Obrázek 22. Postupné skládání robota.....</i>	38
<i>Obrázek 23. Hotový a sestavený robot</i>	38
<i>Obrázek 24. Sledování rovné čáry</i>	39
<i>Obrázek 25. Najetí do levotočivé zatáčky</i>	40
<i>Obrázek 26. Vybočení z dráhy</i>	41
<i>Obrázek 27. Vývojový diagram hlavního programu.....</i>	42

SEZNAM PŘÍLOH

P I Seznam použitých součástí

P II Zdrojový kód

P III CD disk s bakalářskou prací a souborem obsahující zdrojový kód

PŘÍLOHA P I: SEZNAM POUŽITÝCH SOUČÁSTEK

	typ	hodnota	ks
Rezistory	R0207	220	2
	R0207	330	4
	R0207	10k	5
Odporové trimry	CA9MV	25k	2
Kondenzátory	TK.../50V	100n	2
	ELRA-A 470M/16V	470μ	1
LED Diody	L5RED4200	/	2
	L934SGC	/	1
	L934SYC	/	1
Držáky baterií	SN9V1	/	2
Nap. Konektory	NAK 2.5 Z K 9.5x5.5	/	1
Konektory	PFH02-02P	/	1
	PFH02-04P	/	4
	PSH02-02W	/	1
	PSH02-04P	/	2
	PSH02-04W	/	2
Spínače	DM03S1P	/	2
Přepínače	SDIP04	/	1
	ESP101	/	2
Patice	PAT08 DIL ST	/	1
Senzory	QRD1114	/	2
Stabilizátory	L7805CV	/	1
Časovače	NE556N	/	1
Serva	VTS-05A	/	2
Kit	Arduino Uno	/	1
Mikroprocesor	ATMega328	/	1


```

pinMode(LED, OUTPUT);

pinMode(Sensor_L, INPUT);

pinMode(Sensor_R, INPUT);

pinMode(DIL_1, INPUT);

pinMode(DIL_2, INPUT);

pinMode(DIL_3, INPUT);

pinMode(DIL_4, INPUT);

pinMode(Impact, INPUT);

}

//=====
//=====

void loop()
{
    CurrentTime = millis();    //Ukládá aktuální dobu od restartu (ms)

    Signal_L = digitalRead(Sensor_L); //Monitorování vstupů

    Signal_R = digitalRead(Sensor_R); //...

    SW1 = digitalRead(DIL_1);    //..

    SW2 = digitalRead(DIL_2);    //.

    SW3 = digitalRead(DIL_3);

    SW4 = digitalRead(DIL_4);

    Stop = digitalRead(Impact);

//-----

    if(Stop == HIGH){          //Ovládá režim nárazových senzorů

        Flag_1 = 1;

        Go = 0;

        delay(200);

        if(Flag_2 == 1){

            Go = 1;

            Flag_1 = 0;

            Flag_2 = 0;

        }

    }

    if(Stop == LOW && Flag_1 == 1)

        Flag_2 = 1;

//-----

```

```

if(Go == 1){           //Robot má povolen pohyb

  if(SW4 == HIGH && SW1 == LOW){ //Přímý pohyb bez sledování čáry

    analogWrite(Servo_L, 245);

    analogWrite(Servo_R, 1);

  }

  //-----

  if(SW2 == HIGH){     //Nastavování citlivosti pro zatáčky

    if(SW3 == HIGH)

      Interval = 250;

    else

      Interval = 500;

  }

  else

    Interval = 60000;

  //-----

  if(SW1 == HIGH){     //Režim sledování černé čáry

    if(Signal_L == LOW && Signal_R == LOW){ //Na rovné čáře

      analogWrite(Servo_L, 245);

      analogWrite(Servo_R, 1);

      Deadline = 0;

      Synchro = 0;

    }

    //-----

    if(Signal_L == HIGH && Signal_R == LOW){ //Pravotočivá zatáčka

      if(Deadline == 0){

        analogWrite(Servo_R, 0);

        analogWrite(Servo_L, 245);

      }

      if(Synchro == 0){

        PreviousTime = CurrentTime;

        Synchro = 1;

      }

      if((CurrentTime - PreviousTime) >= Interval){

        analogWrite(Servo_R, 245);

        analogWrite(Servo_L, 245);

      }

    }

  }

}

```

```

    Deadline = 1;
}
}
//-----
if(Signal_R == HIGH && Signal_L == LOW){ //Levotočivá zatáčka
    if(Deadline == 0){
        digitalWrite(LED, HIGH);
        analogWrite(Servo_L, 0);
        analogWrite(Servo_R, 1);
    }
    if(Synchro == 0){
        PreviousTime = CurrentTime;
        Synchro = 1;
    }
    if((CurrentTime - PreviousTime) >= Interval){
        analogWrite(Servo_R, 1);
        analogWrite(Servo_L, 1);
        Deadline = 1;
    }
}
}
else{
    if(SW4 == LOW){
        analogWrite(Servo_L, 0);
        analogWrite(Servo_R, 0);
    }
}
else{
    analogWrite(Servo_L, 0);
    analogWrite(Servo_R, 0);
}
}

```