

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor č. 10 - Elektrotechnika, elektronika a telekomunikace

Bender – robot sledující čáru

Bender -The Line Following Robot

Autoři: Adam Heinrich, Zbyněk Moravec

Škola: Mendelovo gymnázium, Opava

Vedoucí práce: Ing. Magdalena Chmelařová, Ph. D.

Opava, 2010

Mobilní robot sledující čáru

Chtěli bychom poděkovat všem, bez kterých by tato práce nikdy nevznikla – především Ing. Magdaleně Chmelařové, Ph. D. za podporu a pomoc s opatřením součástek, Matěji Kubičkovi za poskytnutí schematické značky a footprintů konektoru pro Bluetooth modul a konzultace, firmě Spezial Electronic za poskytnutí Bluetooth modulu a Martinu Thomasovi za svolení se zveřejněním upravených zdrojových kódů bootladeru.

Prohlašujeme, že jsme tuto práci vypracovali samostatně za použití podkladů uvedených v příloženém seznamu.

V Opavě, dne 16. března 2010

Adam Heinrich

Zbyněk Moravec

Mobilní robot sledující čáru

Anotace

Práce popisuje konstrukci, elektroniku a programování mobilního robota vybaveného rozhraním Bluetooth, umožňujícím dálkové ovládání robota, bezdrátové programování a efektivní ladění algoritmů. Úkolem robota je projetí trasy vyznačené černou čarou na bílém podkladu a objíždění překážek.

Klíčová slova: Robotika, Sledování čáry, Autonomní chování

Abstract

This document describes the design of a mobile robot equipped with a Bluetooth interface, that allows remote control of the robot, wireless programming and effective wireless debugging. The robot's task is to follow a black line on a white background and obstacle avoidance.

Keywords: Robotics, Path following, Autonomous behavior

Mobilní robot sledující čáru

I Úvod	5
I.1 Robotika	5
I.2 Motivace	5
I.3 Způsob stavby	5
II Mechanická konstrukce	6
III Elektronika mobilního robota	7
III.1 Blokové schéma	7
III.2 Hlavní deska	8
III.2.1 Schéma zapojení	9
III.2.2 Mikrokontrolér	10
III.3 Bluetooth modul	10
III.3.1 Řízení motorů	11
III.3.2 Napájení	11
III.3.3 Další komponenty na desce	12
III.4 Senzory	13
III.4.1 Snímač čáry	13
III.4.2 Infračervený dálkoměr	15
IV Programování robota	16
IV.1 Programování mikrokontroléru	16
IV.1.1 Bootloader	16
IV.1.2 Základní rutiny	17
IV.1.2.1 USART	17
IV.1.2.2 A/D převodník	17
IV.1.2.3 Servomotor	18
IV.1.2.4 Řízení motorů	19
IV.1.3 Popis algoritmů	20
IV.1.4 Objížďení cihly	20
IV.1.5 Sledování čáry	22
IV.1.6 Monitorování z PC	23
IV.1.7 Odesílání na PC – sendInfo()	24
IV.1.8 Dálkové ovládání – Delphi a J2ME	25
V Závěr	26
VI Významy zkratk	27
VII Seznam příloh	28
VIII Seznam zdrojů	29
VIII.1.1 Publikace, články	29
VIII.1.2 Softwarové projekty	29
VIII.1.3 Dokumentace	29
VIII.1.4 Datasheety	29

I Úvod

I.1 Robotika

Mobilní robotika se stala oblíbenou proto, že jako odvětví robotiky spojuje elektroniku, informatiku a programování, mechaniku i fyziku. Mobilní roboti mohou mít mnoho podob, které se odvíjejí od jejich použití a mohou být dálkově řízení, semiautonomní či zcela autonomní. Takový robot se pak rozhoduje na základě programu uloženého v paměti.

Velice oblíbenou koncepcí malých robotů je malý dvoukolový podvozek s diferenciálním řízením pohybu (to znamená, že robot je vybaven dvěma motory a směr pohybu je řízen změnou rychlosti motorů – podobně jako tank) a relativně levnou elektronikou založenou na mikrokontroléru a dostupných senzorech.

V zámoří jsou pak často pořádány soutěže těchto robotů, jejichž úkolem je například vyhledání a vytlačení soupeře z kruhového ringu v disciplíně Minisumo (v nejčastější variantě soutěže mají roboti maximální rozměry 10×10 cm), vyhledání cesty v bludišti či sledování čáry na kontrastním podkladu, nejčastější kombinací je černá čára na bílém reflexivním podkladu.

V české republice a okolí je soutěží týkajících se robotiky hned několik, například mezinárodní soutěž **Eurobot**¹, ve které roboti plní rozličné úkoly, zápasy Minisumo, konané při soutěži Eurobot, soutěž **Robotour**², kde je úkolem robota projet zadanou trasu v parku, či slovenská soutěž **ISTROBOT**, která má hned několik kategorií – **Stopař**, **Myš v bludišti**, **Minisumo** a **Volná jízda**.

I.2 Motivace

Motivací pro konstrukci robota je kromě praktického vyzkoušení zajímavých algoritmů právě kategorie Stopař soutěže **ISTROBOT**, každoročně pořádaná FEI STU v Bratislavě. Úkolem robota zde je projet trasu určenou černou čarou šířky 15 mm na bílém podkladu v co nejkratším čase. Kromě sledování čáry na robota čeká několik překážek, jako například přerušení čáry v délce 7 cm, objetí překážky na čáře (typicky cihla), či průjezd tunelem. Pravidla se obměňují a jejich přesné znění je k vidění na WWW stránkách soutěže.³

Robota jsme navrhli tak, aby nebyl problém přidat další senzory a plnit i jiné úkoly než sledování čáry.

I.3 Způsob stavby

Robota stavíme v týmu dvou studentů Mendelova gymnázia v Opavě, kteří se delší dobu věnují převážně aplikačnímu programování. **Zbyněk Moravec** má na starosti především programování robota a aplikací pro počítač, **Adam Heinrich** se stará o konstrukci, správnou funkčnost elektroniky a programování základních rutin mikrokontroléru.

Jméno **Bender** robot převzal od jedné z hlavních postav kresleného seriálu Futurama, humanoidního robota Bendera.

1 Soutěž Eurobot – www.eurobot.cz

2 Soutěž Robotour – www.robotika.cz

3 WWW stránka soutěže ISTROBOT: <http://robotika.sk/contest/2010>

II Mechanická konstrukce

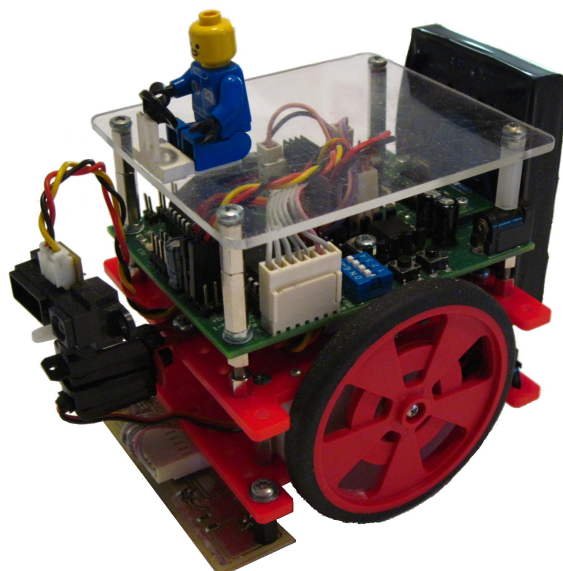
Jako základ byl použit **podvozek UMU-01** vyráběný firmou Hobbyrobot⁴. Jedná se o podvozek vybavený dvěma **motory s převodovkou GM8** výrobce Solarbotics a ocasní kuličkou jako třetím opěrným bodem.^{143: 1}

Tab. 1: Parametry motoru GM8 ⁵	
Převodový poměr	143: 1
Rychlost otáčení	70 ot/min při 5 V
Napájení	maximálně 6 V
Odběr	670 mA při 0 ot/min
Odběr bez zátěže	58 mA

Do desek podvozku jsou vyvrtány díry o průměru 3,2 mm pro šroubky M3 v rastru 10 mm tak, aby pasovaly do rastru stavebnice Merkur. Umístění montážních otvorů na deskách plošných spojů je navrženo obdobně. Kompatibilita se stavebnicí Merkur je žádoucí proto, že se jedná o ideální základ pro další konstrukční vylepšení.

V zadní části podvozku je uchycen držák 6ti AAA akumulátorů. Dále je podvozek doplněn o plexisklovou desku umístěnou nad řídicí deskou robota, která by měla elektroniku chránit, a o uchycení servomotoru a infračerveného dálkoměru.

Rozměry robota jsou asi 170×100 mm , výška je 120 mm .



Obr. 1: Fotografie hotového robota

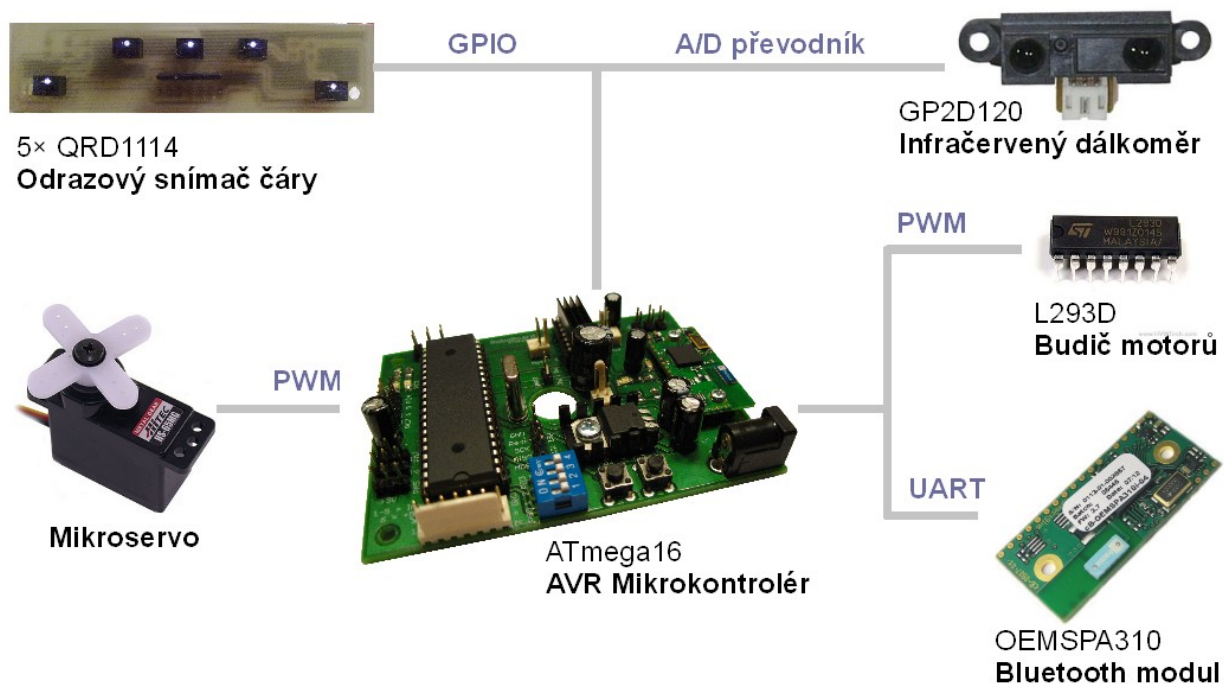
⁴ WWW stránka firmy Hobbyrobot: www.hobbyrobot.cz

⁵ WWW stránka výrobce motoru GM8: <http://www.robotshop.ca/solarbotics-gm8-gear-motor-8-offset.html>

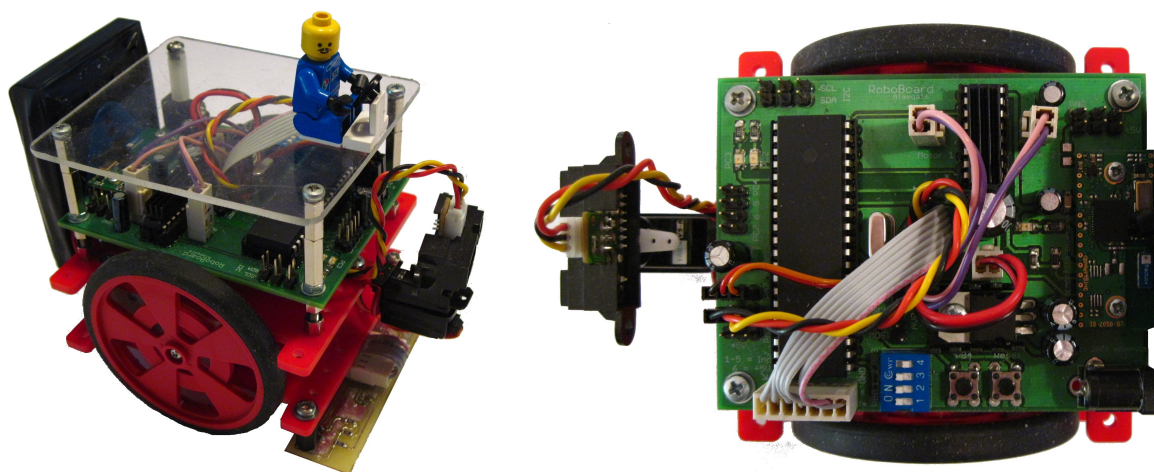
Mobilní robot sledující čáru

III Elektronika mobilního robota

III.1 Blokové schéma



Obr. 2: Blokové schéma zobrazující jednotlivé moduly robota

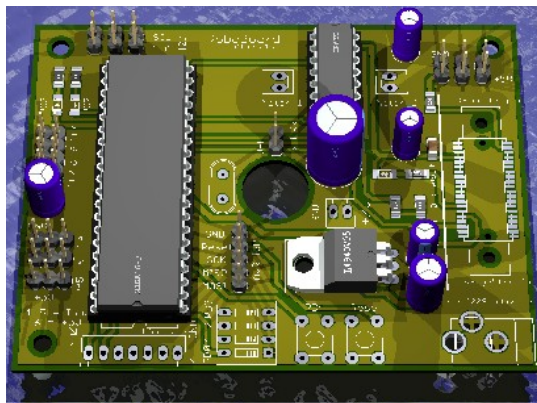


Obr. 3: Další fotografie robota

Mobilní robot sledující čáru

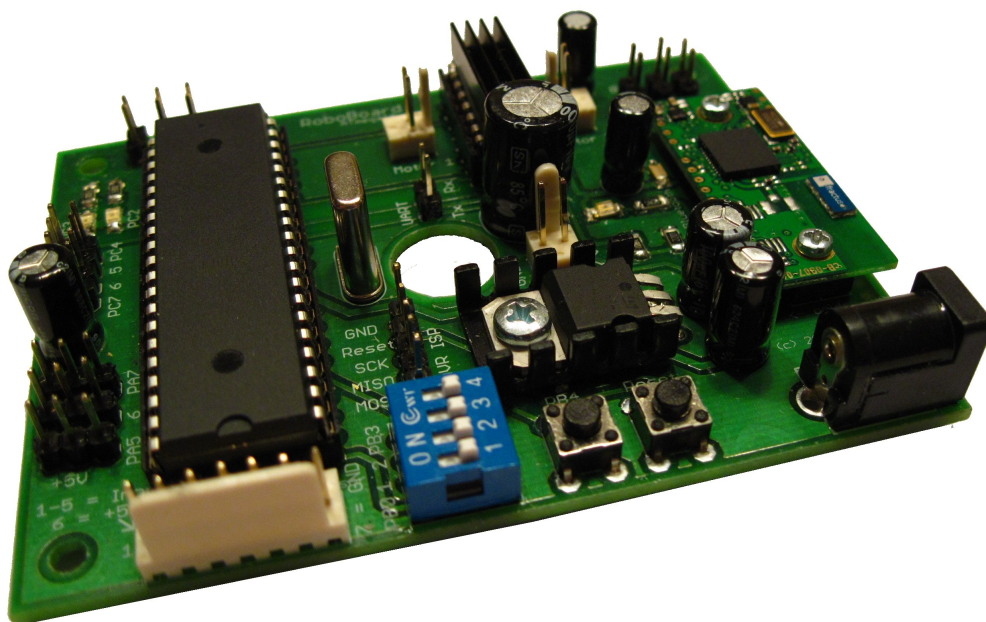
III.2 Hlavní deska

Deska byla navržena jako oboustranná s diskretními součástkami v pouzdech pro povrchovou montáž. Rezistory a LED diody jsou v pouzdech velikosti 1206. Integrované obvody jsou v pouzdru DIP a patiči pro případnou výměnu. Uprostřed je (stejně jako podvozek) vybavena dírou o průměru 10 mm pro průchod kablíků. Ta zabraňuje zamotávání kabelů do koleček. Montážní díry v rozích desky mají průměr 3,2 mm a jsou umístěny tak, aby pasovaly do rastru stovebnice Merkur (10 mm).



Obr. 4: 3D model hlavní desky

Schéma a motivy plošných spojů jsou navrženy ve Freeware verzi programu Eagle⁶. Plošný spoj byl vyroben firmou Pragoboard⁷, je vybaven prokovy, HAL, nepájivou maskou a potiskem vrchní strany desky.



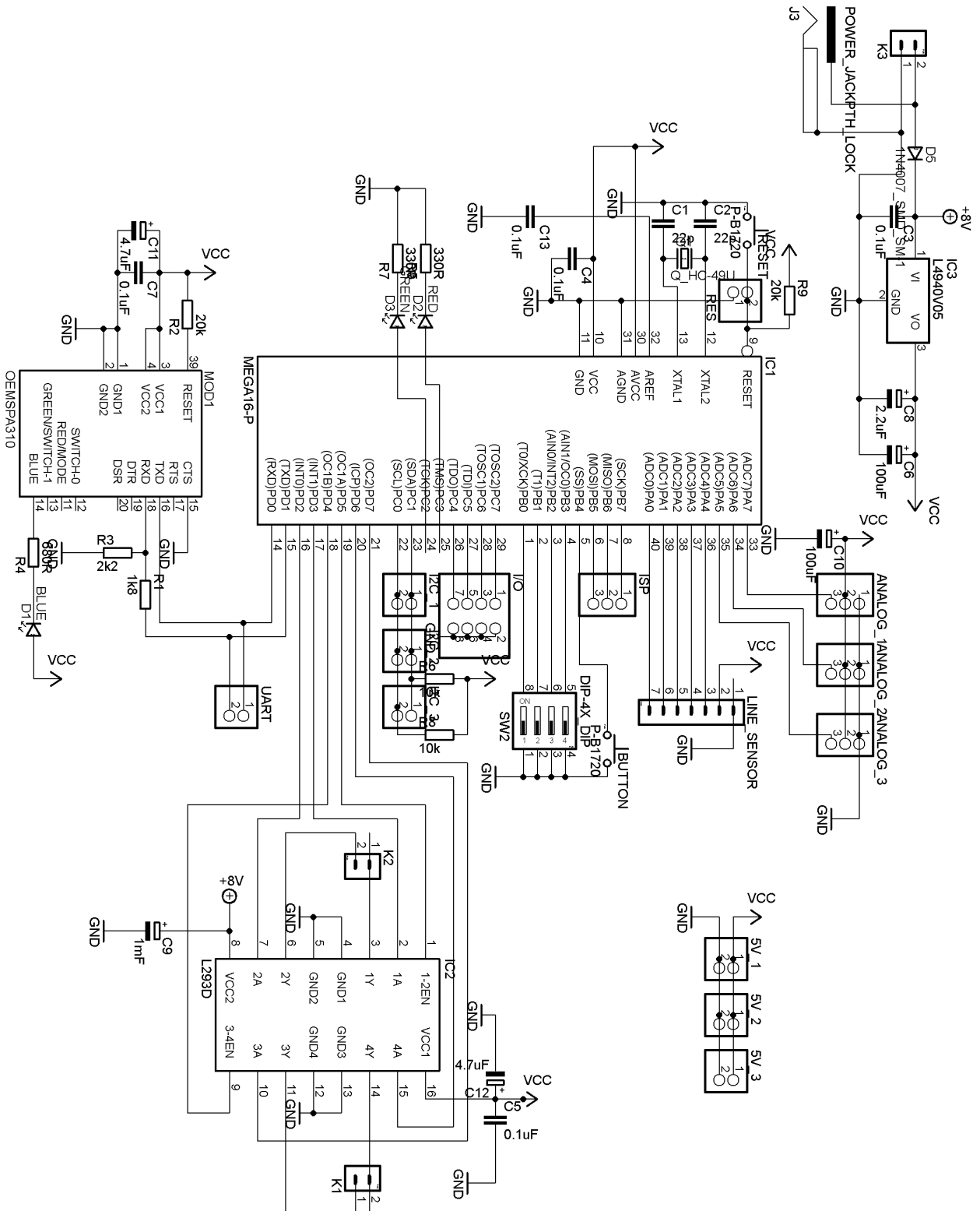
Obr. 5: Fotografie hotové desky

⁶ WWW stránka programu Eagle: www.cadsoft.de/

⁷ WWW stránka firmy Pragoboard: www.pragoboard.cz

Mobilní robot sledující čáru

III.2.1 Schéma zapojení



Mobilní robot sledující čáru

III.2.2 Mikrokontrolér

O řízení robota se stará osmibitový mikrokontrolér **Atmega16** [M16]⁸ s mikroprocesorem řady AVR, vyráběný firmou Atmel. Mikrokontrolér obsahuje 16 kB paměti typu Flash pro uložení programu, 512 B paměti EEPROM a 1 kB paměti RAM. Dále je vybaven jednotkou USART, TWI (ekvivalent pro I2C), 10ti bitovým A/D převodníkem či několika čítači/časovači. Zvolen byl kvůli dostatečnému množství vstupně/výstupních pinů.

III.3 Bluetooth modul

Pro ladění programů, ovládání robota z mobilního telefonu či počítače a bezdrátové programování je použit Bluetooth modul **OEMSPA310** [BT] vyráběný firmou ConnectBlue. Jedná se o „Serial Port Adapter“ (protokol BTSP), to znamená, že využívá profilu SPP a v počítači se hlásí jako běžný sériový port. Použitý modul je nastaven na komunikaci rychlostí 19200 baudů, 8 datových bitů, jeden stop bit bez použití hardwarového řízení toku dat.



Obr. 6: Fotografie modulu OEMSPA310



Obr. 7: Fotografie konektoru

Modul OEMSPA310 je možné připojit přímým připájením k plošnému spoji, připájením kabelu či prostřednictvím speciálního B2B (board-to-board) konektoru SAMTEC⁹, určeného pro povrchovou montáž (SMT). Zvolili jsme právě tento konektor. Je vybaven dvěma otvory se závitem pro šroubky M2, sloužící k přichycení modulu. Výrobce používá stejný konektor pro více typů modulů, není tedy problém použít jiný typ Bluetooth modulu s odlišnými vlastnostmi, nebo modul využívající rozhraní WiFi. Součástku v knihovně pro program Eagle vytvořil a poskytl Matěj Kubička.

Napájení modulu je 3-6 V. Na konektor je přivedeno napájení 5 V a vyvedeny jsou linky Rx a Tx. Linka CTS je uzemněna a na vývod BLUE je připojena LED dioda signalizující přenos dat. Protože Bluetooth modul operuje v 3,3 V úrovních, je vstup Rx s výstupem Tx mikrokontroléru (který používá 5 V úrovně) oddělen odporovým děličem 1k8/2k2. Logické úrovně linky Tx Bluetooth modulu není třeba posilovat, mikrokontrolér je vyhodnocuje správně.

Bluetooth modul OEMSPA310 poskytla firma Spezial Electronic¹⁰.

⁸ Několikapísmenné zkratky v hranatých závorkách, použité v textu (například [M16]), jsou odkazy na citace v Seznamu použité literatury.

⁹ Konkrétně typ *FSI-120-03-G-D-M-AB*

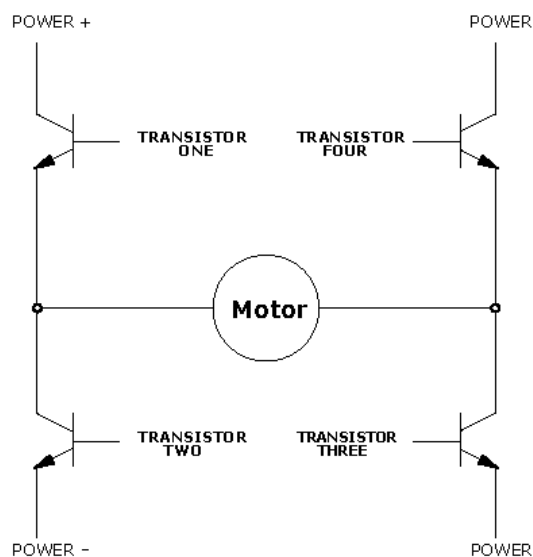
¹⁰ WWW stránka Spezial Electronic: www.spezial.cz

Mobilní robot sledující čáru

III.3.1 Řízení motorů

Pro spínání dvou DC motorů je použit integrovaný H-můstek (H-bridge) **L293D** [STM]. Ten umožňuje řízení rychlosti otáčení pulzně-šířkovou modulací (PWM) a k řízení využívá celkem 6 výstupních pinů mikrokontroléru. Čtyři jsou použity pro nastavení směru otáčení, dva jsou připojeny na PWM výstup 16ti bitového čítače/časovače TCNT2.

H-můstek je zapojení umožňující spínání motoru s možností změny směru otáčení. Zjednodušené schéma je zobrazeno na následujícím obrázku¹¹:



Obr. 8: Spínání DC motoru H-můstkem

Maximální proud, který je H-můstku jednomu motoru schopen dodat, je 0,5 A. Obvod má čtyři vývody využité k chlazení, ty jsou připojeny na rozlitou měď na plošném spoji. Kromě toho byl použit i hliníkový chladič připevněný na pouzdro obvodu.

Motory jsou napájeny z nestabilizovaného napájení desky. Při použití 12 V síťového adaptéru je ale nutné snížit rychlost motorů, které jsou stavěny na napětí 6 V (to je řešeno DIP přepínačem a změnou plnění PWM).

III.3.2 Napájení

Pro získání napětí 5 V k napájení elektroniky je použit integrovaný nízkoúbytkový (Low-Dropout, LDO) stabilizátor **L4940V05** [STM99] v pouzdru TO-220. Napětí poskytované 6ti AAA akumulátory Eneloop je 7,2 V, elektronika robota také umožňuje napájení z 12 V adaptéru. Z těchto důvodů je stabilizátor vybaven hliníkovým chladičem.

Deska je opatřena několika elektrolytickými kondenzátory, pokrývajícími poklesy napětí například při rozjezdu motorů. Kromě toho jsou integrované obvody vybaveny blokovacími kondenzátory o velikosti 0,1 μF .

11 Zdroj obrázku: <http://www.legolab.daimi.au.dk/CSaEA/RCX/Manual.dir/Actuators.dir/Actuators.html>

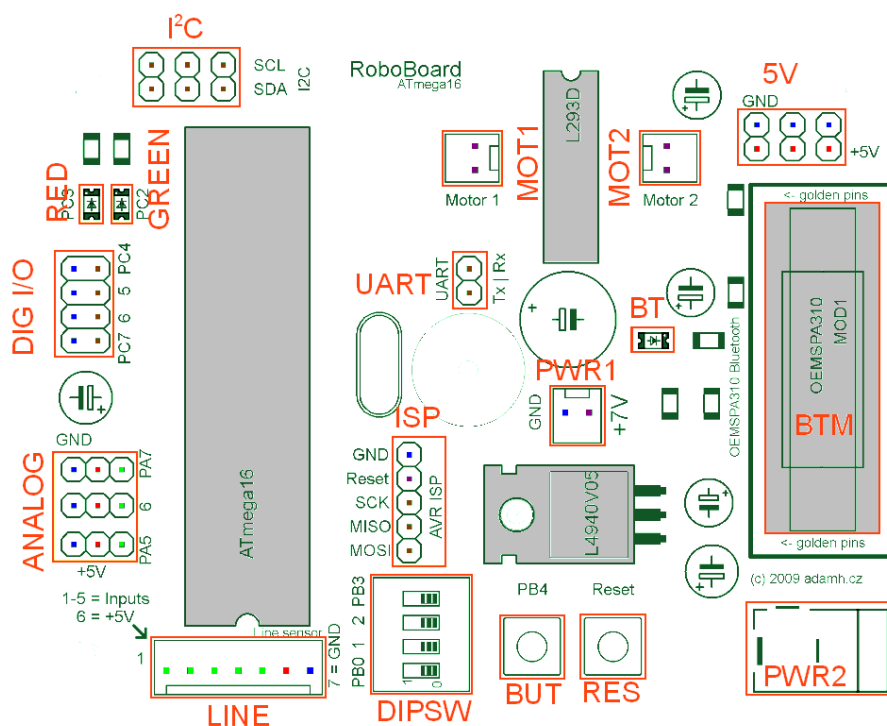
Mobilní robot sledující čáru

III.3.3 Další komponenty na desce

Deska je vybavena červenou a zelenou LED diodou připojenou k mikrokontroléru, zelenou LED signalizující činnost Bluetooth modulu, čtyřbitovým DIP přepínačem sloužícím k nastavování programu, tlačítkem připojeným na *RESET* vývod mikrokontroléru a druhým tlačítkem připojeným na digitální vstup mikrokontroléru.

Na desce se nachází několik konektorů:

- **PWR1** pro připojení akumulátoru a **PWR2** pro připojení síťového adaptéru.
- **ISP** pro připojení ISP programátoru. ISP programátor je použit pouze při nahrání bootloaderu, proto je možné tři piny na konektoru (MOSI, MISO, SCK) využít jako vstupně/výstupní.
- **DIG I/O** obsahující 4 vstupně/výstupní piny a GND.
- 3 konektory **ADC** obsahující vstupy A/D převodníku či digitální vstupy/výstupy, napájení 5 V a GND pro připojení analogových senzorů či servomotorů.
- **LINE** pro připojení senzoru čáry (5 vstupů AD převodníku či digitálních vstupů a napájení).
- **I²C** obsahující linky SDA a SCL pro připojení I²C zařízení. Linky SDA a SCL jsou vybaveny zvedacími 10k zvedacími odpory.
- **5V** obsahující napájení 5V pro připojení dalších periférií
- **MOT1** a **MOT2** pro připojení motorů.
- **BT** pro připojení Bluetooth modulu



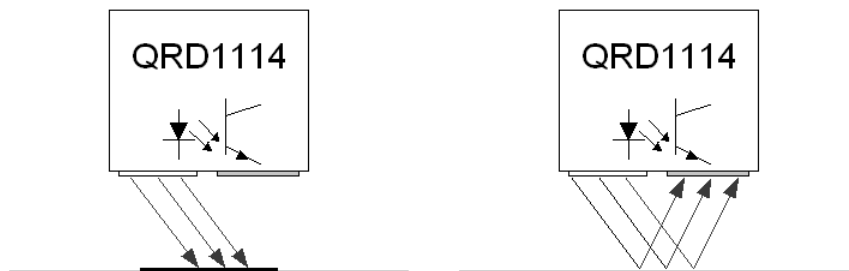
Obr. 9: Popisky konektorů a LED diod na desce

Mobilní robot sledující čáru

III.4 Senzory

III.4.1 Snímač čáry

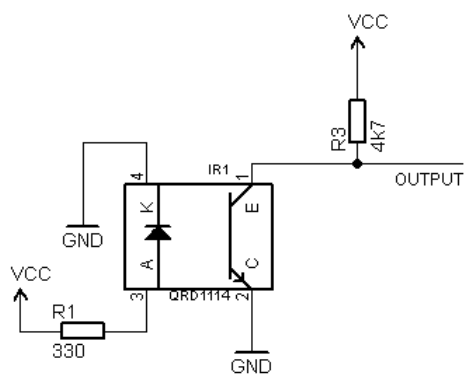
Ke snímání čáry slouží pětice snímačů **QRD1114** [QRD] vyráběných firmou Fairchild Semiconductors. Jedná se o infračervenou LED diodu a fototranzistor umístěné ve společném pouzdru.



Obr. 10: Princip funkce čidla QRD1114

Černá čára, tvořená izolační páskou, infračervené světlo pohltí, takže paprsek na fototranzistor nedopadne. Na výstupu je pak, jak je patrné ze zapojení čidla, vysoká úroveň – tedy **log. 1**.

Pokud se pod čidlem objeví **bílá plocha**, odrazí se paprsek do fototranzistoru a na výstupu čidla se objeví nízká úroveň (**log. 0**).



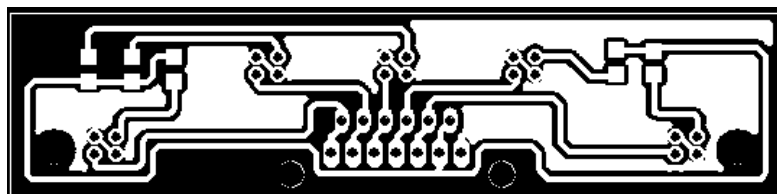
Obr. 11: Zapojení jednoho z pěti čidel

Mobilní robot sledující čáru

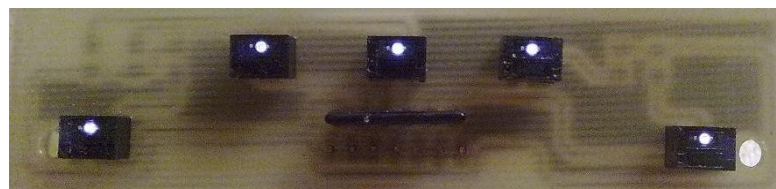
Výstupní signály pěti čidel jsou přivedeny na digitální vstupy mikrokontroléru. Aby bylo možné modul použít na zařízeních¹², kde není možné zapnout interní pull-up rezistory, obsahuje deska 5 zvedacích rezistorů o velikosti $4k7 \Omega$ ve společném pouzdru – pro použití na tomto konkrétním robotovi s mikrokontrolérem Atmega16 ale zvedací rezistory zapotřebí nejsou. Pro připojení k hlavní desce slouží sedmižilný kablík opatřený zámkovými konektory a stav senzorů lze číst buď jako digitální informaci nebo A/D převodníkem, který využívá piny, na které je senzor čáry připojen. Montážní otvory jsou opět navrženy tak, aby pasovaly do rastru stavebnice Merkur. Výstup v analogové podobě má tu výhodu, že můžeme nastavit prahové hodnoty kalibrací, aby robot mohl jezdit i na jiných barvách podkladu.

Rozložení senzorů bylo zvoleno na základě informací o jiných robotech sledujících čáru na webu a jeví se pro daný počet senzorů jako ideální. Krajní čidla jsou od tří prostředních umístěna dále, aby byla schopna zachytit čáru v prudkých zatáčkách a fungují jako jakási záchranná brzda.

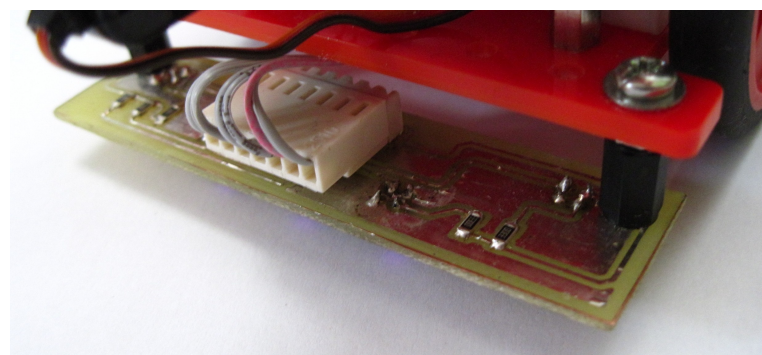
Celý snímač jsme popsali v článku pro časopis Robot Revue¹³ v rámci několikadílného seriálu „Programování jBrainu¹⁴ v Javě“ (Robot Revue číslo 03/2010) a na webové stránce <http://projects.adamh.cz/LineSensor>.



Obr. 12: Deska plošných spojů snímače čáry



Obr. 13: Snímač čáry ze strany senzorů



Obr. 14: Fotografie hotového snímače

12 Například robot jBot – <http://projects.adamh.cz/jBot>

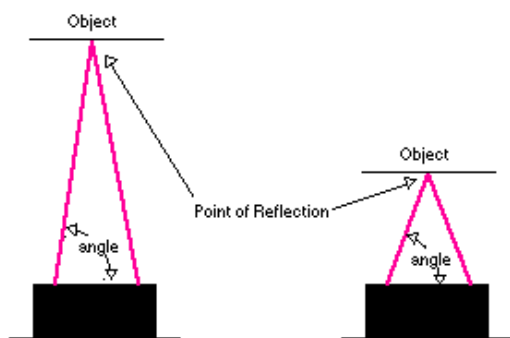
13 Robot Revue – časopis o robotech: <http://robotrevue.cz>

14 jBotBrain – web výrobce: <http://c-bot.eu>

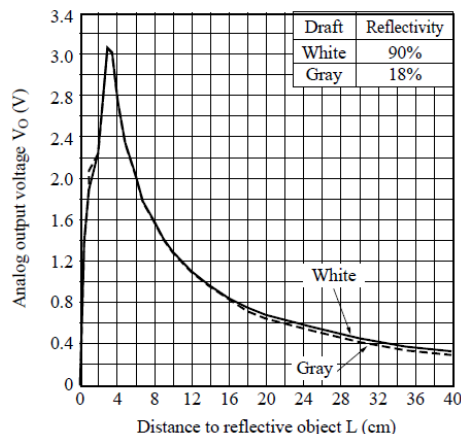
Mobilní robot sledující čáru

III.4.2 Infračervený dálkoměr

Infračervený dálkoměr **GP2D120** [GP2D], vyráběný firmou Sharp, je schopen změřit vzdálenost k překážce v rozsahu 4-30 cm na principu triangulace. Výstupní napětí je nepřímo úměrné vzdálenosti k překážce v rozsahu 4-30 cm, v praxi je možné překážku detekovat i na vzdálenost větší. Na robotovi je senzor připevněn k modelářskému servomotoru, který umožňuje natočení v rozsahu 0-180°.



Obr. 16: Princip měření vzdálenosti [FED05]



Obr. 15: Závislost výstupního napětí na vzdálenosti k překážce [GP2D]

Informaci o vzdálenosti získává mikrokontrolér prostřednictvím A/D převodníku. Protože je vhodné pracovat s naměřenou hodnotou v centimetrech, je potřeba hodnotu převést. To lze buďto za pomoci tabulky uložené v paměti nebo jednoduchým vztahem popsáním v [ACR], který převede výstupní napětí (U) na přibližnou vzdálenost v centimetrech (R). Pro dané potřeby je přesnost výpočtu zcela dostačující:

$$R = (2914 / (U + 5)) - 1$$

Překážka umístěná ve vzdálenosti menší než 4 cm se jeví jako překážka vzdálenější a měření je tudíž nepoužitelné (viz graf). Řešením tohoto problému by bylo ignorování vzdálenosti menší než přibližně 8 cm, nicméně v praxi (podmínky dané soutěží Istrobot, zmíněné v úvodu) nás problém s překážkou ve vzdálenosti menší než 4 cm trápit nemusí.

IV Programování robota

IV.1 Programování mikrokontroléru

Mikrokontrolér je programován v jazyce C za použití toolchainu **WinAvr**¹⁵ (obsahuje například kompilátor avr-gcc, knihovnu avr-libc nebo nástro avrdude pro nahrání programu do paměti mikrokontroléru).

IV.1.1 Bootloader

V paměti mikrokontroléru je uložen bootloader podle aplikační poznámky **AVR109** [AVR109]. Bootloader slouží k bezdrátovému programování pomocí rozhraní Bluetooth. Program vychází z upraveného bootladeru [THO08]. Aktivovaný bootloader se projeví rozsvícením červené LED. Jednoduchá úprava spočívala v kontrole příznaku v EEPROM, bootloader lze tedy aktivovat dvěma způsoby:

- 1) **Stiskem tlačítka PB4** při zapnutí mikrokontroléru (dá se vyvolat také současným stiskem tlačítka PB4 a RESET). Poté bootloader čeká, dokud mu počítač nezačne posílat program.
- 2) **Nalezením příznaku (flag) v EEPROM**. Po zapnutí mikrokontroléru bootloader zkontroluje první bajt (adresa `0x00`) EEPROM a pokud je tento nastaven na hodnotu `0x01`, je přepsán na původní hodnotu `0xFF` a bootloader je aktivován.

V opačném případě bootloader vrátí používané periferie do původního stavu a skočí na začátek programu.

Před samotným programováním nástrojem AVRDUDE¹⁶ je mikrokontroléru zaslán příkaz (zde je to řetězec „@@\n“), který se postará o zapsání příznaku do EEPROM, aktivaci watchdogu a reset mikrokontroléru. Následující kód ukazuje výřez podprogramu přerušení při příjmu bajtu z jednotky USART:

```
switch (command1) {
    // ... ostatní příkazy
    case '@':
        // Příznak do EEPROM
        eeprom_write_byte((unsigned char*)0x00, 0x01);
        wdt_enable(WDTO_30MS); // Zapnutí watchdogu
        while(1); // Nekonečná smyčka
    break;
}
```

O zaslání příkazu pro reset se stará jednoduchý program ComSend (použití: ComSend.exe <COMn> <baudrate> <text>), napsaný v ObjectPascalu (Delphi), který je spuštěn těsně před programem AVRDUDE.

¹⁵ WinAvr – <http://winavr.sourceforge.net/>

¹⁶ AVRDUDE – <http://www.bsdhome.com/avrdude/>

Mobilní robot sledující čáru

IV.1.2 Základní rutiny

Triviální rutiny pro obsluhu periférií byly implementovány již dříve pro použití v jiných projektech¹⁷ na základě informací z datasheetu ATmega8 [M8] a překompilovány, případně lehce upraveny pro mikrokontrolér ATmega16 [M16].

O Inicializaci a nastavení periférií (načtení knihoven, namapování pinů, nastavení rychlostí) se stará soubor *board.h*. V něm jsou také uložena makra například pro čtení stavu tlačítek nebo ovládání LED diod, které zjednodušují programování.

IV.1.2.1 USART

Jednotka USART (*Universal Synchronous/Asynchronous Receiver and Transmitter*) je použita pro dálkové ovládání, monitorování a ladění programu v robotovi. Pro ladění v terminálu (například Hyperterminál nebo Bray Terminal¹⁸) jsou implementovány funkce pro odesílání řetězců. Rutiny pro práci s jednotkou USART jsou uloženy v souboru *lib/uart.c*.

- `void uartInit (int ubrr)` – zapnutí USARTu a přerušení, parametr `ubrr` slouží k nastavení baudrate
- `void uartPutc (unsigned char c)` – odeslání znaku
- `unsigned char uartReceive ()` – příjem znaku
- `void uartPuts (char *txt)` – odeslání řetězce
- `void uartPuti (int i)` – odeslání celého čísla ve formě řetězce (pro ladění v terminálu)
- `void uartPutInt(int i)` – odeslání čísla ve třech bajtech

IV.1.2.2 A/D převodník

A/D převodník je použit pro měření vzdálenosti infračerveným dálkoměrem **GP2D120**. Inicializace je v souboru *board.h*, o čtení napětí se stará funkce `int readAdc (unsigned char channel)` uložená v souboru *lib/adc.c*. Jako reference je použito napětí 5 V na pinu AVCC.

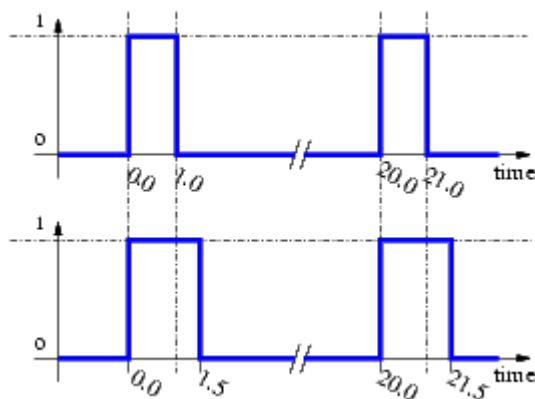
17 Například na robotovi popsaném v práci „*Konstrukce mobilního robota schopného pohybu ve venkovním prostředí*“ (SOČ 2009, Adam Heinrich) – <http://projects.adamh.cz/robot2>

18 Bray terminal – <http://sites.google.com/site/braypp/terminal>

Mobilní robot sledující čáru

IV.1.2.3 Servomotor

Modelářský servomotor je motor s převodovkou a řídicí elektronikou, která se stará o jeho natočení na zadaný úhel v rozsahu 180° (standardně by měl být rozsah natočení rozsah pouze 90° , v praxi ale servomotory nabízejí rozsahy větší). Velikost natočení je získávána zpětnovazebným potenciometrem umístěným na výstupní ose serva. Natočení je nastaveno pulzy o frekvenci 50 Hz. Šířka pulzu je přímo úměrná natočení servomotoru a je v rozsahu 0,5 – 2,5 ms. Hodnota 1,5 ms odpovídá střední poloze. Ukázky signálu jsou zobrazeny na následujícím grafu¹⁹:



Obr. 17: Graf řídicího signálu

Signál je generován programově v přerušení čítače/časovače TCNT2:

```
void handleServo() {  
    SERVO_TIMER = 10;  
    servoPhase++;  
  
    if (servoPhase == servoDirection) {// aktuální natočení serva  
        servoOff();  
    }  
  
    if (servoPhase == 1280) {// 20 ms - konec jedné periody  
        servoOn();  
        servoPhase = 0;  
    }  
}
```

K nastavení výstupní polohy servomotoru slouží makro **turnServo** (*int direction*). Krajní hodnoty natočení byly pro daný servomotor určeny experimentálně.

¹⁹ Obrázek pochází z <http://robotika.cz/guide/servo/cs>

Mobilní robot sledující čáru

IV.1.2.4 Řízení motorů

O řízení motorů se stará soubor *lib/motors.c*. Důležité jsou tyto funkce makra:

- void **leftSpeed** (*int speed*) – nastavení rychlosti levého motoru. Rychlost 0 slouží k zastavení motoru. Maximální rychlosti odpovídá hodnota 255. Pokud je hodnota záporná, motor se otáčí dozadu.
- void **rightSpeed** (*int speed*) – nastavení rychlosti pravého motoru.
- **motorsInit** () – nastavení výstupních pinů pro řízení směru otáčení, inicializace PWM.

Řízení rychlosti motorů je realizováno **pulzně-šířkovou modulací (PWM)**. Signál PWM je generován na pinech OC1A a OC1B díky čítači/časovači **TCNT1**. Ten je nastaven následovně:

- Phase Correct PWM
- Bez předděličky (No prescaler)
- TOP = OCR1A nebo OCR1B
- nastavení pinů OC1A nebo OC1B při dosažení hodnoty TOP

V souboru *board.h* jsou uložena dvě důležitá makra, **setHighSpeed** () a **setLowSpeed** (). Slouží k přepnutí rychlosti při použití různých druhů napájení. Robota je totiž kromě šesti AAA baterií možné napájet i 12 V síťovým adaptérem. Napětí 12 V by mohlo motory poškodit, proto musíme při použití adaptéru rychlost podstatně snížit. K tomu slouží 3. bit na DIP přepínači.

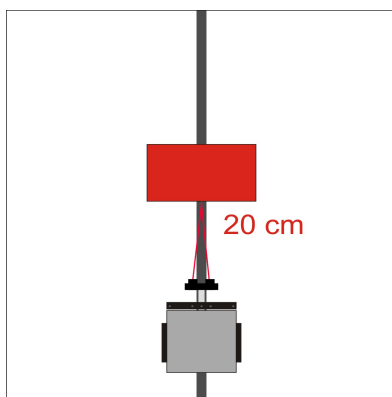
Mobilní robot sledující čáru

IV.1.3 Popis algoritmů

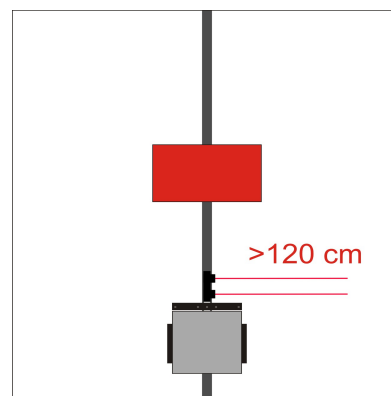
Robot sleduje čáru IR senzory na spodní straně robota. Celý algoritmus sledování čáry funguje na principu PID regulátoru. V průběhu sledování čáry sleduje vzdálenost na IR dálkoměru – pokud je získaná vzdálenost menší než 10 cm, začne algoritmus vykonávat dílčí procesy (např. Objíždění překážky – cihly).

IV.1.4 Objíždění cihly

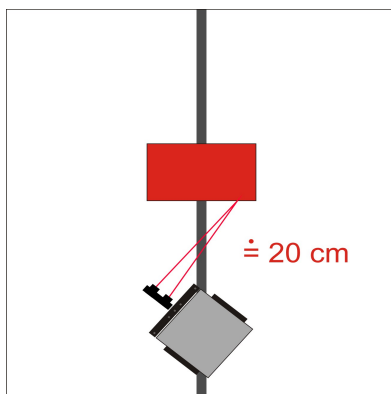
Postup objetí překážky v devíti krocích:



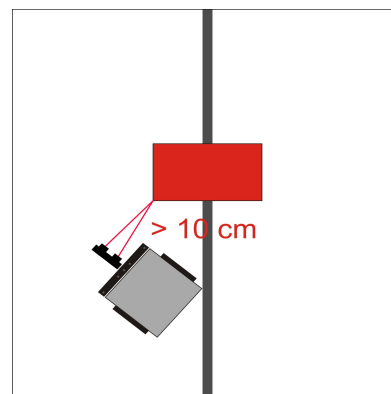
Krok 1: Dálkoměr detekuje překážku.



Krok 2: Dálkoměr se vytočí napravo, do volného prostoru.

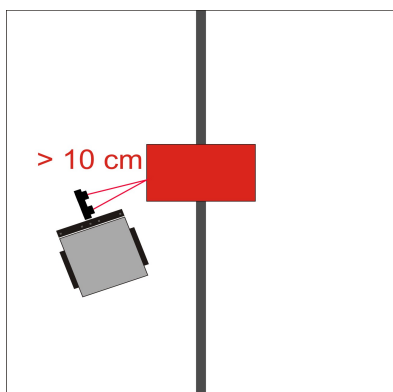


Krok 3: Robot se začne otáčet doleva, dokud neměří hodnotu podobnou 20 cm.

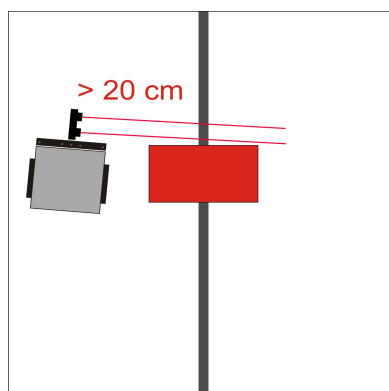


Krok 4: Robot se dále přibližuje k cihle, jede rovně.

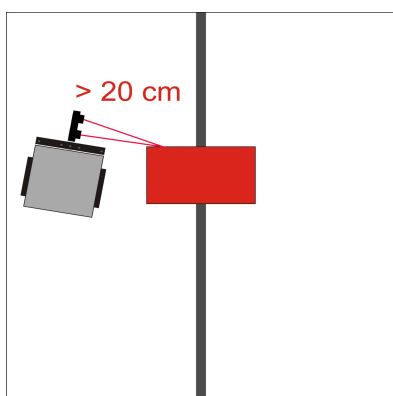
Mobilní robot sledující čáru



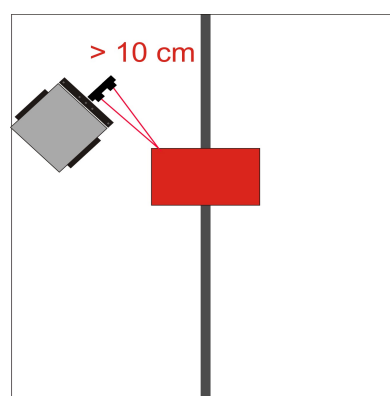
Krok 5: Robot se snaží držet si od cihly odstup 10 cm pomocí regulátoru.



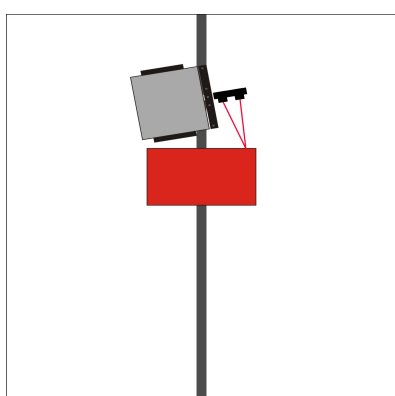
Krok 6: Pokračuje v objíždění cihly, dokud čidlo nenaměří vysokou hodnotu (konec překážky).



Krok 7: Robot začne zatáčet, dokud se napřiblíží k cihle.



Krok 8: A opět pokračuje v jízdě podél cihly.



Krok 9: Pokud narazí na čáru, objíždění cihly se ukončí a robot pokračuje dále po čáře.

Mobilní robot sledující čáru

IV.1.5 Sledování čáry

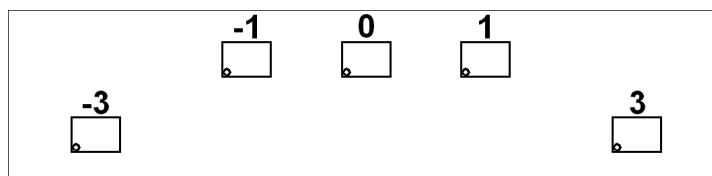
Bez využití regulátoru by trajektorie byla zbytečně dlouhá. Robot by dělal někdy příliš velké nebo příliš malé zatáčky a jeho dráha by vypadala asi takto:



PID regulátor [WIN07] reguluje rychlost otáček dvou motorů, aby bylo sledování čáry plynulejší:



Cílem při sledování čáry je, aby se čára nacházela pod čidlem s hodnotou 0. Pro každé čidlo je určena jiná hodnota odchylky, ta je uložena v proměnné **error**.



```
diff = error-prev_error;  
prev_error = error;  
sum_e += error;  
pid = kp*error + ki*sum_e + kd*diff;
```

- **error** je odchylka určená na základě pozice čáry (**P – Proporcionální složka regulátoru**).
- **prev_error** je odchylka naměřená v předchozí iteraci.
- **diff** je rozdíl odchylky a minulé odchylky (**D – Diferenční složka regulátoru**).
- **sum_e** je akumulovaná odchylka (**I – Integrační nebo sumační složka regulátoru**).
- **kp**, **ki** a **kd** jsou koeficienty určující parametry PID regulátoru.

Mobilní robot sledující čáru

Nastavení koeficientů určujících parametry PID regulátoru je klíčové a neobejde se bez ladění, které probíhá díky rozhraní Bluetooth bezdrátově. Integrovaná (sumační) složka se v prvních fázích ladění vůbec neuvažuje (tedy $KI = 0$), její koeficient se nastavuje jako poslední a je velmi malá.

Podle velikosti proměnné `pid` se pak reguluje směr otáčení motorů:

```
if (pid > 0) {
    leftSpeed(255 - pid);
    rightSpeed(255);
} else {
    leftSpeed(255);
    rightSpeed(255 + pid);
}
```

IV.1.6 Monitorování z PC

Pomocí počítače jsme schopni kontrolovat stav jednotlivých senzorů na čáru, pozici servomotoru, vzdálenost objektu a rychlost jednotlivých motorů. Celá komunikace probíhá pomocí Bluetooth modulu. S tímto rozhraním se pracuje stejně, jako se standardním sériovým portem (COM).

Odesílání informací z robotu do počítače je pomocí funkce `sendInfo()`.

„Balíček“ (packet) informací se skládá z 11 bajtů:

1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.
Začátek	Vzdálenost	IR čidla	Natočení serva	Levý motor		Pravý motor		Konec		

Mobilní robot sledující čáru

IV.1.7 Odesílání na PC – sendInfo()

```
void sendInfo() { // Send info to PC

    int cidla = 0; // Nastavení bitů čidel
    cidla |= (LINE1 << 0);
    cidla |= (LINE2 << 1);
    cidla |= (LINE3 << 2);
    cidla |= (LINE4 << 3);
    cidla |= (LINE5 << 4);

    // Odesílání informací do PC
    uartPutc(1);
    uartPutc(range());
    uartPutc(cidla);
    uartPutc(servoDirection);
    uartPutInt(motor_l);
    uartPutInt(motor_r);
    uartPutc('\n');
    delay(10);
}
```

Rychlost motorů se odesílá v hodnotě PWM signálu, která se následně přepočte na procentuální. Jednotlivé bity se uloží do bajtu a jsou odeslány do počítače, který si je následně zpět dekoduje.

Mobilní robot sledující čáru

IV.1.8 Dálkové ovládání – Delphi a J2ME

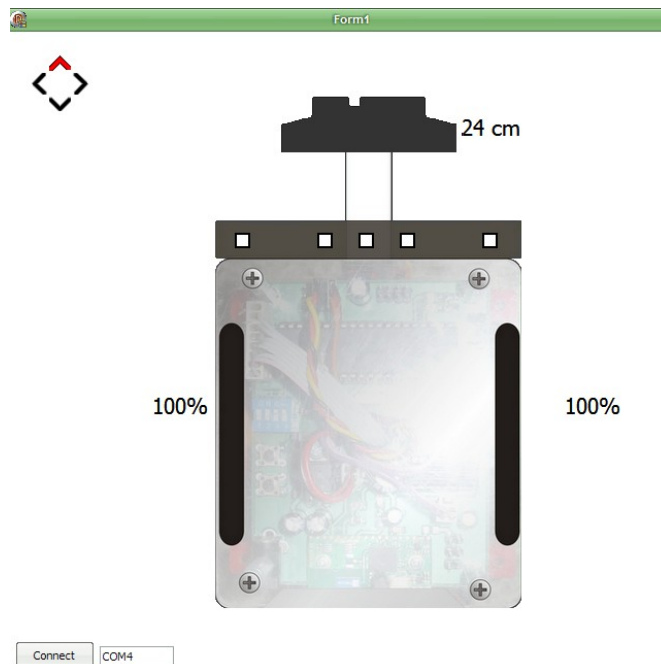
Robota je díky Bluetooth rozhraní možné ovládat dálkově jednoduchým tříbajtovým protokolem:

DIR1	DIR2	STOP
------	------	------

- **DIR1** a **DIR2** udávají směr levého a pravého motoru:
 - 'F': otáčení dopředu (*F jako forward*)
 - 'B': otáčení dozadu (*B jako backward*)
 - 'S': zastavení motoru (*S jako stop*)
- **STOP** je stop bajt, tedy '\n' (konec řádku). Konec řádku byl zvolen proto, aby bylo možné příkazy odesílat i v terminálovém programu.

Jednoduchá aplikace pro mobilní telefon je napsaná v jazyce **Java (J2ME)** ve vývojovém prostředí **NetBeans**²⁰. Práce s rozhraním Bluetooth probíhá díky rozhraní **JSR 82 [JSR82]** za použití profilu **BTSPP**.

Monitorovací a ovládací program pro PC je napsán v **Delphi**, s využitím knihovny *Cportlib*. Grafické části jsou vykreslovány technologií DirectX s využitím knihovny *DelphiX*²¹. Program umožňuje dálkové ovládání výše popsaným protokolem a zobrazení stavu senzorů a rychlostí motorů na obrazovce počítače (v originále má pozadí černou barvu, z důvodu tisku byla barva změněna na bílou).



20 NetBeans IDE – www.netbeans.org

21 DelphiX - <http://www.micrel.cz/Dx/>

Mobilní robot sledující čáru

V Závěr

Povedlo se nám zkonstruovat a naprogramovat plně funkčního mobilního robota, schopného účasti na některých robotických soutěžích. Stavba nám pomohla zdokonalit se v elektronice a programování, a to jak mikrokontrolérů, tak i mobilních telefonů a aplikací pro osobní počítače.

V prosinci 2009 jsme se s robotem zúčastnili soutěže středoškolských robotů **Kyber robot 2009**²², na které nám byla udělena cena časopisu AUTOMATIZACE.



Obr. 18: Prezentace na soutěži Kyber robot 2009

Informace o robotovi byly zveřejněny na WWW stránce <http://projects.adamh.cz/LineFollower> tak, aby byly přístupné i dalším zájemcům o robotiku. Zveřejněny jsou zdrojové kódy některých ukázkových programů, jednoduchá ovládací aplikace pro PC, fotografie, schémata i výkresy desek plošných spojů.

Na robotovi budeme pracovat i nadále – nutné budou například malé mechanické úpravy pro hladké absolvování dráhy na soutěži Istrobot, která obsahuje most (naštěstí s mírným stoupáním). Také bychom se chtěli zabývat řešením dalších úloh, například hledáním cesty v bludišti.

22 WWW stránka soutěže Kyber robot 2009: <http://www.fm.tul.cz/cs/kyberrobot09>

VI Významy zkratek

- **LED** Light-Emitting diode - *světlo vyzařující dioda*
- **DIP** Dual In-Line Package – *pouzdro, ve kterém se dodávají integrované obvody*
- **Tx** Transmit - *přenos*
- **Rx** Receive - *příjem*
- **DC** Direct Current - *stejnoseměrný proud*
- **PWM** Pulse Width Modulation - *pulzně šířková modulace*
- **ISP** In-System Programming – *programování v systému*
- **PID** Proportional, Integral, Derivative – *typ regulátoru*
- **ADC** Analog to Digital Converter – *analogově/digitální převodník*
- **AVR** Advanced Risc Machine – *mikroprocesorová řada výrobce Atmel*
- **J2ME** Java 2 Platfom, Micro Edition – *Java Virtual Machine pro mobilní zařízení*

VII Seznam příloh

Přílohy jsou umístěny jak v ZIP archivu přiloženém spolu s prací, tak i volně dostupné na WWW stránkách robota²³. Obsahuje následující adresáře:

- *boards* obsahuje schémata a motivy DPS ve formátu Eagle.
- *code-bootloader* obsahuje zdrojový kód upraveného bootloADERu.
- *robot-control* obsahuje zdrojový kód jednoduché ovládací aplikace v Delphi.
- *robot-example1* obsahuje zdrojový kód pro dálkové ovládaní robota z PC.
- *robot-linefollower* obsahuje zdrojový kód jednoduchého sledování čáry.

Protože v době psaní práce robota dále aktivně programujeme abychom stihli účast na soutěži Istrobot, nejsou v přílohách umístěny kompletní zdrojové kódy finální verze robota sledujícího čáru pomocí PID regulátoru, nýbrž pouze jedny z prvních programů, které sloužily k otestování hotového hardware a k publikaci pro ostatní zájemce na webu.

²³ WWW stránka projektu: <http://projects.adamh.cz/LineFollower>

VIII Seznam zdrojů

VIII.1.1 Publikace, články

- [KRY08] KRYŠ, Václav; KOT, Tomáš; BABJAK, Ján; MOSTÍN, Vladimír. Testing and calibration of IR proximity sensors. *Acta Mechanica Slovaca*. 2008, ISSN: 1335-2393
- [NOV05] NOVÁK, Petr. *Mobilní roboty*. 1. vyd., nakladatelství BEN Praha, 2005. 247 stran. ISBN: 80-7300-141-1
- [WIN05] WINKLER, Zbyněk. *Řízení serva* [online], 2005 [cit. 08. 02. 2010]. Dostupné na: <http://robotika.cz/guide/servo/cs>
- [WIN07] WINKLER, Zbyněk. *Řízení pohybu* [online], 2007 [cit. 08. 02. 2010]. Dostupné na: <http://robotika.cz/guide/control/cs>
- [CAR06] CARRICK, Micah. *AVR Controlled GP2D120 Distance Sensor* [online], 2006 [cit. 08. 02. 2010]. Dostupné na: <http://www.micahcarrick.com/05-11-2006/avr-controlled-gp2d120-distance-sensor.html>
- [SOC09] SOCIETY OF ROBOTS. *Sensors – Sharp IR rangefinder* [online], 2009 [cit. 08. 02. 2010]. Dostupné na: http://www.societyofrobots.com/sensors_sharpirrange.shtml
- [ACR] ACRONAME, Inc.. *Linearizing Sharp Ranger Data* [online], [cit. 21. 02. 2010]. Dostupné na: <http://acroname.com/robotics/info/articles/irlinear/irlinear.html>
- [FED05] FEDER, Paul. *The Sharp GP2D120* [online], 2005 [cit. 21. 02. 2010]. Dostupné na: <http://itp.nyu.edu/physcomp/sensors/Reports/GP2D120>

VIII.1.2 Softwarové projekty

- [THO08] THOMASS, Martin. *ATMEL AVR UART bootloader for AVR-GCC/avr-libc* [online], 2008 [cit. 08. 02. 2010]. Dostupné na: http://www.siwawi.arubi.uni-kl.de/avr_projects/#avrprog_boot

VIII.1.3 Dokumentace

- [LIBC] FREE SOFTWARE FOUNDATION. *AVR Libc Home Page* [online], 2008 [cit. 08. 02. 2010]. Dostupné na: <http://www.gnu.org/savannah-checkouts/non-gnu/avr-libc/>
- [AVR109] ATMEL. *AVR109: Self Programming* [online], 2004 [cit. 08. 02. 2010]. Dostupné na: http://www.atmel.com/dyn/resources/prod_documents/doc1644.pdf
- [JSR82] SUN. *JSR 82 Bluetooth API and OBEX API* [online], 2006 [cit. 14. 3. 2010]. Dostupné na: <http://java.sun.com/javame/reference/apis/jsr082/>

VIII.1.4 Datasheety

- [QRD] FAIRCHILD SEMICONDUCTORS. *Datasheet QRD114* [online], 2005 [cit. 01. 02. 2010]. Dostupné na: <http://www.fairchildsemi.com/ds/QR/QRD114.pdf>
- [GP2D] SHARP. *Datasheet GP2D120* [online], 2006 [cit. 21. 02. 2010]. Dostupné na: <http://www.sharpsma.com/Page.aspx/americas/en/part/GP2D120/>
- [M8] ATMEL. *Datasheet ATmega8* [online], 2007 [cit. 08. 02. 2010]. Dostupné na: http://atmel.com/dyn/resources/prod_documents/doc2486.pdf
- [M16] ATMEL. *Datasheet ATmega16* [online], 2009 [cit. 08. 02. 2010]. Dostupné na: <http://www.atmel.com/atmel/acrobat/doc2466.pdf>
- [BT] CONNECTBLUE. *Datasheet OEMSPA310* [online], 2008 [cit. 08. 02. 2010]. Dostupné na: http://www.spezial.com/doc/conblu/em_ds_oemspa_310.pdf
- [STM] STMICROELECTRONICS. *Datasheet L293D* [online], [cit. 08. 02. 2010]. Dostupné na: <http://www.st.com/stonline/books/pdf/docs/1330.pdf>
- [STM99] STMICROELECTRONICS. *Datasheet L4940V05* [online], 1999 [cit. 08. 02. 2010]. Dostupné na: http://www.jklir.net/jklir_soubory2/odkazy/l4940v05.pdf