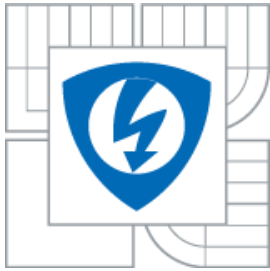




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ  
ÚSTAV AUTOMATIZACE A MĚŘÍCÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

# VYLEPŠENÍ ZPŮSOBU CHŮZE A OVLÁDÁNÍ 6- NOHÉHO KRÁČEJÍCÍHO ROBOTU ČERNÁ VDOVA

EMENDATION METHOD WALKING AND CONTROL 6-LEGGED WALKING ROBOT BLACK  
WIDOW

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

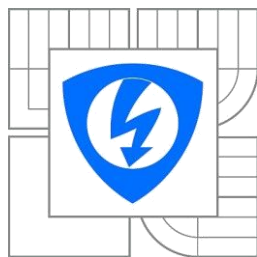
AUTOR PRÁCE  
AUTHOR

JAN KREJSA

VEDOUCÍ PRÁCE  
SUPERVISOR

ING. VLASTIMIL KRÍŽ

BRNO 2014



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav automatizace a měřicí techniky

# Bakalářská práce

bakalářský studijní obor  
Automatizační a měřicí technika

**Student:** Jan Krejsa

**ID:** 146042

**Ročník:** 3

**Akademický rok:** 2013/2014

## NÁZEV TÉMATU:

**Vylepšení způsobu chůze a ovládání 6-nohého kráčejičího robotu Černá vdova**

## POKYNY PRO VYPRACOVÁNÍ:

1. Proveďte rozbor možností chůze 6-nohých kráčejičích robotů.
2. Po dohodě s vedoucím implementujte vybrané způsoby chůze do předloženého robotu. Robot bude schopen chůze na nerovné ploše s překážkami do výšky 3 cm.
3. Vytvořte rozhraní a aplikaci pro bezdrátové ovládání a programování robotu.
4. Proveďte rešerši možností vzdáleného nahrání programu do řídicího mikrokontroléru robotu. Vybraný způsob implementujte.
5. V případě potřeby upravte hardware robotu.

## DOPORUČENÁ LITERATURA:

Hrouzek, M., Kráčejičí mikrorobot s modelářskými servomechanismy - mechanická konstrukce, Diplomová práce, VUT v Brně, 2002

Kruták, T., Kráčejičí mikrorobot s modelářskými servomechanismy - řídicí systém, Diplomová práce, VUT v Brně, 2002

**Termín zadání:** 10.2.2014

**Termín odevzdání:** 26.5.2014

**Vedoucí práce:** Ing. Vlastimil Kříž

**Konzultanti bakalářské práce:**

**doc. Ing. Václav Jirsík, CSc.**

*Předseda oborové rady*

## UPOZORNĚNÍ:

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

## **Abstrakt**

Bakalářská práce se zabývá problematikou řízení, ovládní a vzdáleného programování šestinožného kráčejiho robota. Je rozdělena do šesti kapitol. V úvodu se zabývá teorií vzdáleného programování. Druhá část rozebírání možné způsoby kráčení šestinožných podvozků. Dále se tato práce zabývá řešením překonávání překážek, tedy chůzi po nerovné ploše. Následující část rozebírání možnosti bezdrátové komunikace. Poslední dvě kapitoly popisují hardware a softwarové vybavení robota.

## **Klíčová slova**

Bezdrátová komunikace, bootloader, mikroprocesor, řízení servomotoru, snímač polohy, šestinožný kráčeji robot, vzdálené programování

## **Abstract**

This bachelor thesis deals with control, operating and remote programming of a six-legged walking robot. It is divided into six chapters. The introduction chapter is about remote programming. The second chapter describes possible walking ways of six-legged chassis. Furthermore this thesis deals with the overcoming of obstacles, thus to walking on uneven ground. The following chapter discusses the possibilities of wireless communication. The last two chapters describes the hardware and software of the robot.

## **Keywords**

Wireless communication, bootloader, microprocessor, steering servo, position sensor, six legged walking robot, remote programming

### **Bibliografická citace:**

KREJSA, J. *Vylepšení způsobu chůze 6-nohého kráčejičího robotu Černá vdova*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2014. 49s. Vedoucí bakalářské práce byl Ing. Vlastimil Kříž.

## **Prohlášení**

Prohlašuji, že svou bakalářskou práci na téma Vylepšení způsobu chůze a ovládnání 6-nohého kráčejičího robotu Černá vdova jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

V Brně dne: **26. května 2014**

.....  
podpis autora

## **Poděkování**

Děkuji vedoucímu bakalářské Ing. Vlastimilovi Křížovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne: **26. května 2014**

.....

podpis autora

# OBSAH

1	Úvod.....	9
2	Rešerše vzdáleného programování .....	10
3	Rozbor možností chůze 6-nohých krácejících robotů.....	13
3.1	Vlastnosti krácejících podvozků .....	13
3.2	Stabilita/nestabilita krácejících podvozků.....	13
3.2.1	Statická stabilita .....	13
3.2.2	Dynamická stabilita .....	13
3.3	Šestinohý krácející podvozek.....	14
3.3.1	Způsoby kráčení.....	14
4	Způsob chůze PO nerovné ploše.....	17
4.1	Princip .....	17
4.2	Snímače polohy .....	17
4.2.1	Mechanické kontaktní snímače polohy.....	18
4.2.2	Ultrazvukové snímače polohy .....	18
4.2.3	Indukčnostní snímače polohy s potlačeným magnetic-kým polem .....	19
4.2.4	Kapacitní bezkontaktní snímače polohy .....	20
4.3	Vlastní řešení chůze po nerovné ploše .....	20
5	Způsoby Bezdrátové komunikace.....	21
5.1	Bluetooth .....	21
5.2	WiFi.....	22
5.3	ZigBee .....	23
5.4	VF moduly .....	24
5.4.1	Komunikační modul BIM-433-F .....	24
6	Hardware robota .....	25
6.1	Konstrukce robota .....	25
6.2	Konstrukce končetin robota .....	26
6.3	Modelářské servomotory a jejich řízení.....	26
6.3.1	Princip řízení servomotorů.....	27
6.3.2	Vlastnosti použitých servomotorů .....	27
6.4	Pohyb přes překážky .....	28
6.4.1	Uchycení táhel končetin.....	29
6.4.2	Chodidla končetin .....	31
6.5	Detailní konstrukce nárazníků.....	32
6.6	Řídící elektronika .....	32
7	Software .....	35
7.1	Hlavní řídicí program .....	35
7.1.1	Důležité funkce a principy programu .....	35
7.1.2	Přerušení od čítače/časovače 0 .....	37

7.1.3	Přerušení od sériové linky 1.....	38
7.2	Uživatelské rozhraní.....	39
7.2.1	Popis aplikačního módu.....	40
7.2.2	Popis programovacího módu .....	40
7.2.3	Tvar posílané zprávy.....	41
7.3	Bootloader .....	41
7.3.1	Princip funkce bootloaeru .....	42
8	Závěr .....	43



# 1 ÚVOD

Kráčející roboty představují pouze malé procento dnes používaných mobilních robotů. Nejrozšířenějšími jsou kolové roboty, které disponují velice dobrými vlastnostmi, jako jsou například jednoduchost řízení a rychlost pohybu. Hlavním důvodem používání kráčejících podvozků i přes jejich často obtížné řízení je schopnost překonávat překážky. Velikost překonatelných překážek je určena hlavně hardwarovou konstrukcí daného robotu. V dnešní době se stále častěji objevují takzvané hybridní konstrukce. Jedná se o roboty, které jsou obvykle postaveny na kombinacích kolového, kráčejícího a pásového podvozku. Díky těmto kombinacím mají hybridní podvozky lepší vlastnosti, zejména prostupnost terénem.

Tato práce je rozdělena do šesti kapitol. Nejprve rozebírá možnosti vzdáleného programování. Takzvaný bootloader je software uložený v paměti programu mikroprocesoru umožňující vzdálené programování. Jedná se o program využívající možnost přepisovat paměť programu. Je také schopen přepsat či úplně odstranit sám sebe.

V druhé kapitole se práce zabývá způsoby kráčení šestinohých robotů a základními vlastnostmi jednotlivých způsobů. Nejdůležitější vlastnost, o které je nutné se zmínit, je stabilita. Čím více má robot končetin, tím snáze se docílí statické stability. U robotů s menším počtem končetin (tři a méně) nelze statické stability docílit. Tyto roboty využívají dynamické stability získané ze setrvačnosti pohybu. Dalšími parametry, o kterých se v této kapitole pojednává, jsou rychlost pohybu a nosnost robotu.

Dále je rozebrán princip chůze kráčejících robotů po nerovné ploše. S touto problematikou jsou úzce spjaty snímače polohy. Aby robot pohybující se v terénu nepřepadl a pohyboval se přirozeně, musí řídicí jednotka dostávat informace od končetin o jejich výšce, nebo o jejich kontaktu se zemí. Na chodidlech končetin jsou proto zpravidla umístěny snímače polohy.

V následující kapitole jsou popsány způsoby bezdrátové komunikace, včetně implementovaného způsobu pomocí rádiového VF modulu BIM-433-F od firmy Radiometrix.

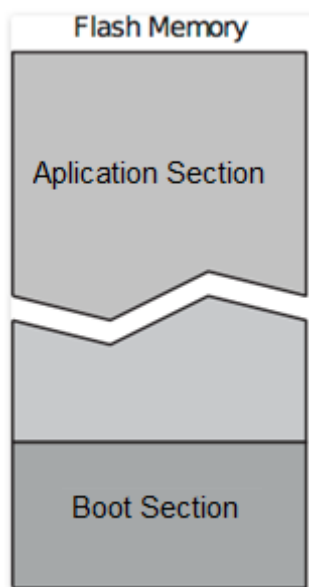
Kapitola nazvaná hardware robota se zabývá od mechanického provedení po elektroniku robota. Je zde detailně popsána konstrukce končetin včetně vypočítaného maximálního zatěžovacího momentu působícího na servomotor. Dále je v této kapitole popsán princip řízení servomotorů. Elektrické vybavení představují dvě desky plošných spojů. Jedna slouží pouze pro komunikaci s počítačem a druhá, hlavní deska, jako řídicí jednotka robota. Ta je osazena mikroprocesorem ATmega 162.

Poslední kapitola popisuje softwarové vybavení robota. Tím jsou tři programy. Bootloader sloužící pro vzdálené programování. Uživatelské rozhraní pro bezdrátové ovládání i programování a hlavní řídicí program starající se o pohyb robota.

## 2 REŠERŠE VZDÁLENÉHO PROGRAMOVÁNÍ

Bootloadery lze využít u všech mikrokontrolérů, které podporují self programming. Jedná se o funkci, která umožňuje programu uloženému v paměti programu přepisovat sám sebe [1].

Bootloader, v české literatuře také často označován jako zavaděč, je speciální program, který slouží k programování mikrokontrolerů bez nutnosti použití externích programovacích stanic, programátorů či jiného hardwaru. Bootloader se nachází v paměti programu (nejčastěji paměť typu flash) v takzvané Boot Section, která je nejčastěji umístěna na konci paměti programu. Paměť programu je tedy z pohledu bootloaderu rozdělena na dvě části. Část pro bootloader (Boot Section) a část pro vlastní aplikaci (Application Section) [1]. Tuto část může bootloader smazat a následně do ní zapsat novou aplikaci. Bootloadery také obvykle disponují funkcí aktualizace sama sebe, nebo úplného odstranění. Po odstranění bootloaderu je aplikaci přístupná celá paměť programu.



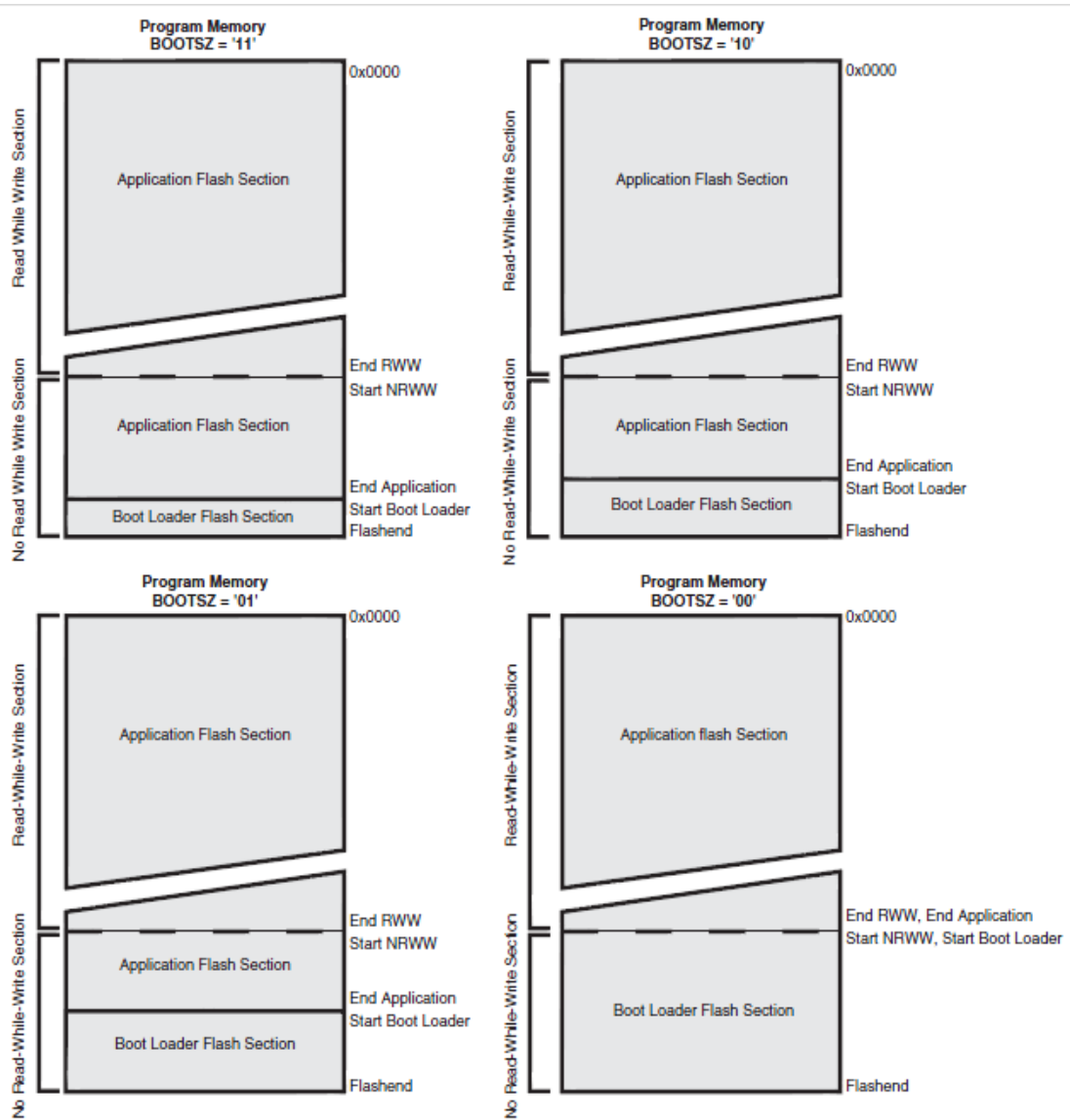
Obr. 2.1: Rozdělení paměti na Application a Boot Section [4]

Paměť programu je také rozdělena na RWW (Read - While - Write) a NRWW (Non - Read - While - Write) Section. Zásadní rozdíl mezi RWW a NRWW sekcemi je v chování mikroprocesoru při přepisování stránky, která je v jedné z těchto oblastí.

Přepisuje-li se stránka z RWW oblasti, mikroprocesor může během této akce spolehlivě vykonávat program z NRWW oblasti. Je však zakázáno vykonávat program z RWW oblasti, jelikož by mohlo dojít ke kolizi zápisu (mazání) a čtení. Proto je také nutné zakázat přerušení, která jsou mapována právě do RWW oblasti [2] [4].

Při aktualizaci stránek v NRW oblasti je chod mikroprocesoru automaticky zastaven [2] [4].

Z datasheetu pro daný mikroprocesor lze zjistit, kolik tyto oblasti obsahují stránek (pages) a tedy kolik místa zabírají v paměti. Sekce RWW slouží pro ukládání vlastních aplikací, které mohou, ale nemusí zasahovat i do sekce NRW. Sekce NRW je vyčleněna pro bootloader, z čehož vyplývá, že samotný bootloader nikdy nesmí svojí velikostí přesáhnout místo vyhrazené pro NRW sekci. Ta tedy mimo jiné udává maximální velikost bootloderu [1]. Oblast RWW vždy začíná na adrese 0x0000 a končí tam, kde začíná oblast NRW. Ta nemá pevný začátek. Jelikož NRW oblast končí na konci paměti, její začátek se určuje podle toho, jak jsou nastaveny bity BOOTSZ0 a BOOTSZ1.



Obr. 2.2: Dělení paměti na RWW a NRW sekce [4]

Po připojení napájecího napětí nebo vyvolání restartu může mikroprocesor začít zpracovávat paměť programu ze dvou míst. Výběh určují fuses, někdy také označované jako pojistky. Jedná se o konfigurační bity určující vlastnosti bootloADERu, jako jsou jeho velikost (BOOTSZ0 a BOOTSZ1) a reset vektor (BOOTRST). Pokud je BOOTRST nastaven do úrovně logické jedničky jedná se o výchozí nastavení a program se začíná vykonávat od adresy 0x0000 (reset vektor). To je začátek aplikační sekce a začne se tedy provádět námi naprogramovaná aplikace. Pokud je však BOOTRST nastaven na úroveň logické nuly reset vektor je nastaven na začátek bootloADERu. Očekává se tedy změna programu v aplikační sekci [2] [4].

Při zpracovávání může sám bootloADER přeprogramovat pomocí SPM instrukcí paměť programu novým softwarem, který přijímá přes sériovou linku neboli UART. BootloADERy nejsou však omezeny pouze používáním sériové linky, ale mohou přijímat nový software jakýmkoli jiným rozhraním, které daný mikrokontrolér podporuje [1].

SPM instrukce jsou instrukce, které slouží pouze pro self programing a nelze s nimi tedy pracovat v jiných částech programu než v samotném bootloADERu uloženém právě v Boot Section.

SPM instrukce pracují ve třech základních krocích. K práci potřebují speciální paměť o velikosti jedné stránky nazývanou page buffer, registr SPMCR, který specifikuje činnost SPM instrukce a registry Z, R0 a R1 [1] [3].

V prvním kroku se provede smazání stránky. Nastaví se adresa stránky do registru Z, který se skládá ze dvou částí. První část pro nastavení adresy slova v page bufferu a druhá část pro nastavení adresy stránky. Následným nastavením bitů PERS, SPEN v SPMCR registru a vykonáním instrukce SPM dojde k smazání dané stránky [1].

Druhým krokem je zápis slova do page bufferu. Slovo zapisujeme do registrů R1:R0 a adresu do registru Z. Nastavením bitu SPEN v SPMCR registru a vykonáním instrukce SPM se slovo zapíše do page bufferu [1].

Ve třetím kroku se stránka zapisuje do paměti programu. To lze pouze, pokud jsou již v page bufferu kompletní data (kompletní stránka připravená pro zápis do fyzické paměti). Adresa stránky se uloží do registru Z, v SPMCR registru se nastaví bity PGWRT a SPEN a vykonáním instrukce SPM se stránka uloží do paměti programu [1].

BootloADER pokračuje stejnou posloupností kroků, dokud nebudou všechny stránky v paměti programu nahrazeny stránkami nové aplikace. Poté co bootloADER dokončí programování flash paměti novým programem, pokračuje se první instrukcí tohoto programu, která leží na adrese reset vektoru.

Bit	7	6	5	4	3	2	1	0	
	SPMIE	RWWSB	–	RWWSRE	BLBSET	PGWRT	PGERS	SPEN	SPMCR
Read/Write	R/W	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Obr. 2.3: Registr SPMCR [3]

# 3 ROZBOR MOŽNOSTÍ CHŮZE 6-NOHÝCH KRÁČEJÍCÍCH ROBOTŮ

## 3.1 Vlastnosti kráčejších podvozků

Mezi hlavní důvody používání kráčejších podvozků, patří jejich schopnost překonávat překážky, která dělá problémy jinak nepoužívanějším kolovým podvozkům v mobilní robotice. Dále pak podobnost pohybu kráčejších robotů s pohyby živočichů či člověka. Kráčejší podvozky mají také velké množství možností, co se týče počtu končetin, nebo stupňů volnosti končetin. Aby byl vůbec pohyb kráčejšího podvozku možný je nutné, aby končetiny disponovaly alespoň dvěma stupni volnosti.

Hlavními nevýhodami těchto podvozků jsou často pomalejší pohyb a nestabilita, která způsobuje největší problémy u podvozků s menším počtem končetin. Rychlost pohybu je ovlivněna množstvím akčních členů, které je potřeba obsloužit a také způsobem řešení stability robotu. U vícenohých podvozků bývá právě stabilita jedním z hlavních důvodů jejich používání.

## 3.2 Stabilita/nestabilita kráčejších podvozků

Stabilita kráčejších podvozků je nezbytně nutná vlastnost k vykonání pohybu. Rozeznáváme dva druhy stability - statickou a dynamickou.

### 3.2.1 Statická stabilita

Statickou stabilitou se rozumí, že robot musí být v každém časovém okamžiku ve stabilní poloze. Toho jsou schopny docílit podvozky se čtyřmi nebo více končetinami. Těžiště celého robotu musí být uvnitř polygonu, který tvoří končetiny, na nichž robot právě stojí. Těžiště je možné přesouvat nakláněním robota, nebo přesouváním hmotné zátěže. Tou je nejčastěji akumulátor. Abychom docílili staticky stabilního robota bez přesouvání hmotné zátěže je nutné, aby robot měl alespoň čtyři končetiny a v jeden časový okamžik byla zdvihnutá pouze jedna z nich. Tři končetiny budou vždy udržovat těžiště robota ve vzniklém trojúhelníku a čtvrtou nohou bude robot kráčet [5] [6].

### 3.2.2 Dynamická stabilita

Robot je dynamicky stabilní, pokud je staticky stabilní pouze v určitých okamžicích svého pohybu. To znamená, že v některých fázích pohybu se dostává do nestabilní polohy. Příkladem takového pohybu je běh nebo skákání. Dynamická stabilita spoléhá na dynamické jevy a setrvačnost pohybu, proto požaduje určitou minimální rychlost pohybu. Výhodou je možnost dosažení větších rychlostí, než je tomu u statické stability.

Nevýhodou je však daleko obtížnější řízení, a také nemožnost zastavit v každém okamžiku pohybu tak, aby robot zůstal ve stabilní poloze [5] [6].

### 3.3 Šestinohý kráčející podvozek

Šestinohé podvozky disponují šesti končetinami nejčastěji na šestiúhelníkovém či obdélníkovém podvozku, kde každá končetina má alespoň dva stupně volnosti. Hlavní výhodou těchto podvozků je lehce dosažitelná statická stabilita, které lze docílit vhodným způsobem chůze a také vhodným rozmístěním nejtěžších prvků do hlavních os robotu (nejčastěji akumulátor). Šestinohé kráčející podvozky jsou z hlediska statické stability nuceny v každém okamžiku stát alespoň na třech končetinách.

#### 3.3.1 Způsoby kráčení

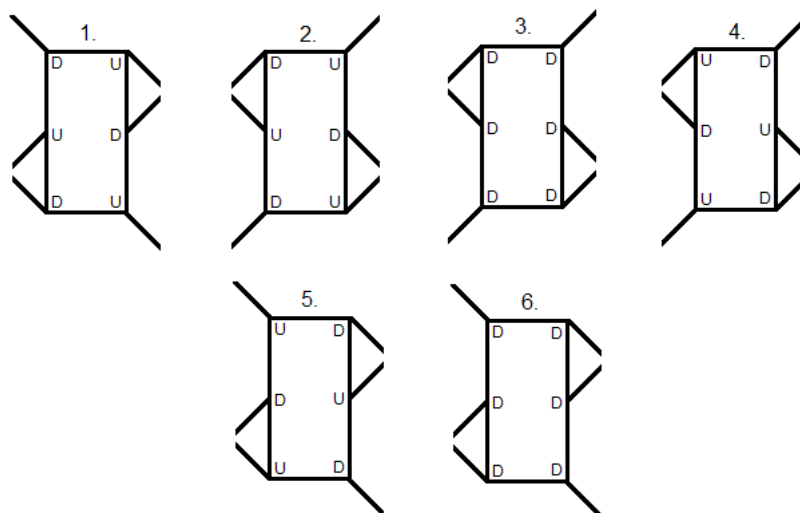
Každý ze způsobů kráčení je založen na určité posloupnosti příkazů pro jednotlivé končetiny. Proto je potřeba před začátkem cyklického opakování jednoho ze způsobů kráčení provést inicializaci. To znamená, že robot si přednastaví končetiny buď do stejných například prostředních poloh, nebo do pozice, která je prvním krokem odpovídajícího způsobu kráčení.

##### 3.3.1.1 Tři nohy v záběru

Tento způsob je nejjednodušší, co se týče obtížnosti řízení. Servomotory zajišťující pohyb robota, mají pouze dva stavy vychýlení do jedné a druhé krajní polohy. V každém okamžiku pohybu robot stojí minimálně na třech nohách. Je to nejméně nohou, na nichž může robot stát a docílit tak statické stability. Nosnost v tomto režimu je nejnižší ze zmiňovaných způsobů kráčení.

Algoritmus přímočaré chůze způsobem tři nohy v záběru (Obr. 3.1):

- Zdvih 2., 4. a 6. nohy
- Všech šest nohou se přesune do opačné polohy. Nohy 1, 3 a 5 svým přesunem vykonají pohyb robota. Zdvížené nohy 2, 4 a 6 se pouze připraví do takové polohy, aby ve stavu č. 4 zajistili pohyb robota.
- Položení nohou 2, 4 a 6
- Zdvih 1., 3. a 5. nohy
- Opět se všech šest nohou přesune do opačné polohy. Nyní však pohyb vykonávají nohy 2, 4 a 6 připravené z kroku č. 2. Nohy 1, 3 a 5 se připravili pro zajištění pohybu v dalším cyklu.
- Položením nohou 1, 3 a 5 se robot dostal do počátečního stavu a těchto šest stavů se již cyklicky opakuje.



Obr. 3.1: Algoritmus přímého kráčení způsobem tři nohy v záběru

### 3.3.1.2 Čtyři nohy v záběru

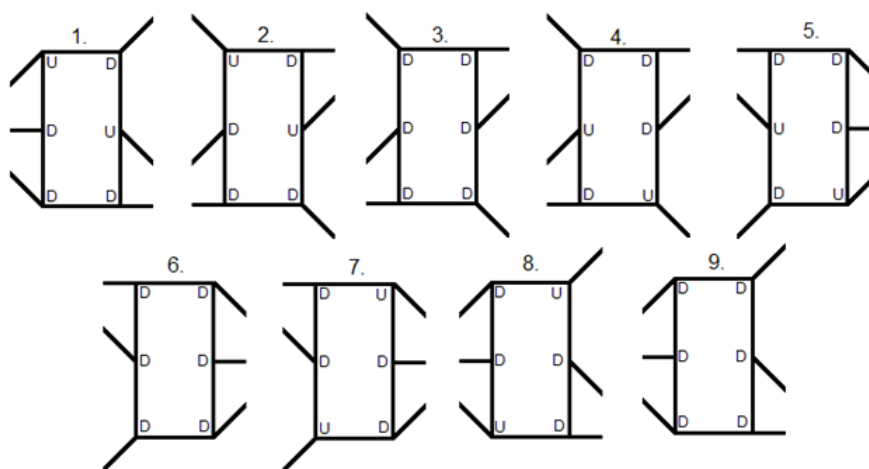
Tímto způsobem kráčení se robot pohybuje po čtyřech končetinách. Tím se změní jeho vlastnosti oproti způsobu s třemi nohami v záběru. Stabilita a nosnost se o jednu nohu zlepší, ale každá noha má nyní tři polohy natočení, tudíž bude i obtížnější řízení. Algoritmus kráčení bude mít devět stavů v jednom cyklu. Tím se také přibližně 1,5x zpomalí rychlost pohybu oproti způsobu se třemi nohami v záběru.

Algoritmus přímočaré chůze způsobem čtyři nohy v záběru (Obr. 3.2):

Zdvih prvního páru nohou

- Natočení čtveřice zatížených nohou do následující polohy. Současně se natočí i nezatížené nohy do první polohy.
- Položení prvního páru nohou.

Tento postup se provede pro všechny tři páry nohou. Po položení posledního páru je cyklus kráčení u konce a začne se opakovat.



Obr. 3.2: Algoritmus přímého kráčení způsobem čtyři nohy v záběru

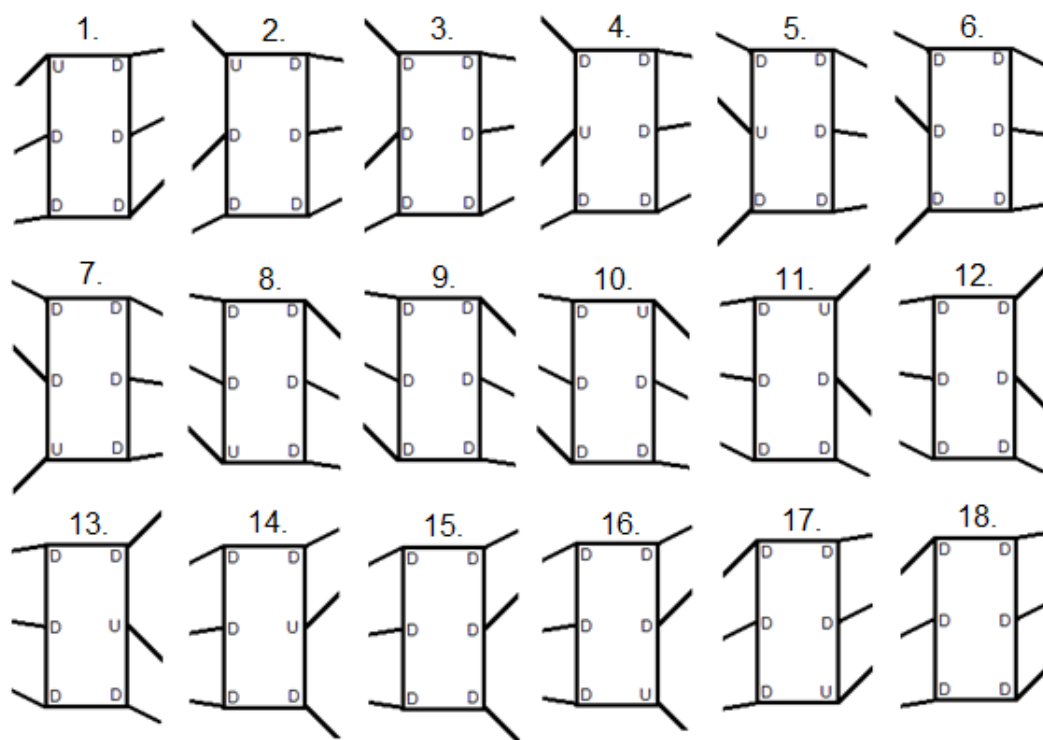
### 3.3.1.3 Pět noh v záběru

Z hlediska řízení se jedná o nejsložitější způsob kráčení. Každá noha má šest poloh natočení. Při pohybu robot stojí na pěti nohách. Tím je zajištěna maximální stabilita a nosnost robota. Algoritmus kráčení má nyní osmnáct stavů, a proto je tento způsob pomalejší než ostatní způsoby s menším počtem nohou v záběru.

Algoritmus přímočaré chůze způsobem pět nohou v záběru (Obr. 3.3):

- Zdvih jedné nohy.
- Všech pět nohou, které jsou v kontaktu s povrchem, se přesune o jednu šestinu plného vychýlení servomotoru. Zbývající noha se přesune na první polohu.
- Položení zdvižené nohy.

Tento postup se provede pro všech šest noh robota. Po položení šesté nohy byl dokončen jeden cyklus kráčení, který se dále cyklicky opakuje.



Obr. 3.3: Algoritmus přímého kráčení způsobem pět noh v záběru

### 3.3.1.4 Šest noh v záběru

Jelikož se jedná o šestinohý kráčející podvozek, nabízí se zde i způsob kráčení šest noh v záběru. Jelikož robot v určitých časových okamžicích stojí pouze na pěti nohách, je název tohoto způsobu značně matoucí a jedná se v podstatě o způsob pět noh v záběru. Jediným rozdílem je, že při vykonávání pohybu robota je v kontaktu se zemí všech šest noh na rozdíl od předchozího způsobu, kdy bylo v kontaktu se zemí pět noh. Stabilita ani nosnost se nezlepšují, zato je opět pomalejší rychlost pohybu. Tento "způsob kráčení" tedy nepřináší žádné zlepšení vlastností.



## 4 ZPŮSOB CHŮZE PO NEROVNÉ PLOŠE

### 4.1 Princip

Aby byl robot schopen chůze po nerovné ploše a překonávat tak překážky, je nutné znát v každém okamžiku pohybu polohu končetin. Je tedy žádoucí, aby robot zastavil akční členy (servomotory) pohybující s končetinami právě v čase, kdy dojde k jejich kontaktu se zemí a tyto pozice byly dále drženy do další změny pohybu. Jinak by nebylo možné říci, že robot dokáže bezpečně překonávat překážky. To znamená, že by při pohybu docházelo k jeho neočekávanému naklánění nebo pádům.

K získání informace o vertikální poloze končetin se používají spojitě nebo nespojitě snímače polohy. Pozice servomotorů jsou většinou získávány pomocí zpětné vazby. Přímo v servomotech je zabudován enkodér vyhodnocující aktuální polohu, která se vrací zpět do řídicí jednotky. Servomotory se zpětnou vazbou bývají nákladnější, a proto se často využívá jiného řešení. Tím je vhodné softwarové řízení servomotorů.

### 4.2 Snímače polohy

Existuje mnoho snímačů polohy, ale pro problematiku chůze krácejících robotů po nerovné ploše je vhodné použít pouze malé procento z nich. U menších konstrukcí robota se nejčastěji používají mechanické snímače, které mění svůj stav na základě přítomnosti/nepřítomnosti překážky, v našem případě země (podlahy) nebo nepřekonatelné překážky. Tu detekují přední a zadní nárazníky, které jsou osazeny těmito snímači. Jejich hlavními výhodami jsou snadné zpracování informace od snímače a minimální hmotnost. U větších konstrukcí se kromě mechanických kontaktních snímačů také používají ultrazvukové snímače polohy. Jejich využití má však smysl spíše v mapování prostoru, v němž se robot pohybuje.

Nespojitě snímače polohy vyhodnocují pouze dva stavy. Buď je sepnuto, nebo rozepnuto. Tomu odpovídají napěťové úrovně pro logickou „1“ a „0“. Ty jsou následně logickými obvody nebo mikroprocesorem zpracovány a vyhodnocovány. Kromě jejich četného využívání v průmyslu jsou také velice často používány na chodidlech končetin krácejících robotů právě jako snímače dotyku končetin se zemí. Hlavním důvodem používání nespojitých snímačů polohy je jejich jednoduchost.

Naopak spojitě snímače polohy jsou schopny udávat aktuální polohu. Tyto snímače mohou poskytovat informace o aktuální výšce nohy nad zemí, nebo o poloze robota v prostoru (vzdálenosti od stěn, detekce nepřekonatelných překážek). Použitím spojitých snímačů polohy je možné efektivněji mapovat okolí robota.

### 4.2.1 Mechanické kontaktní snímače polohy

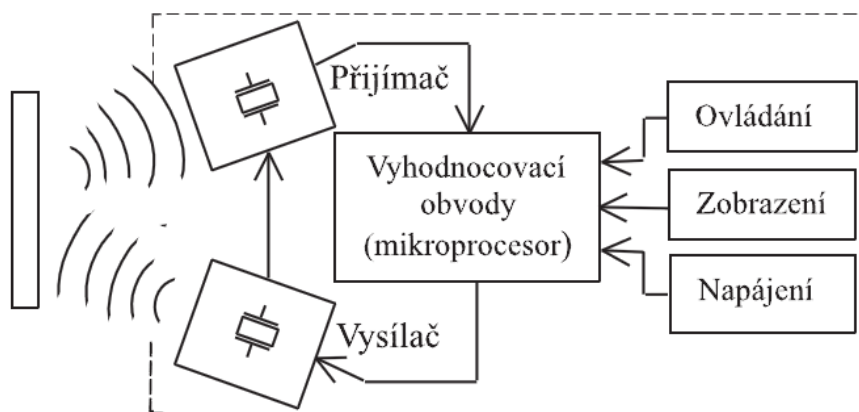
Působením neelektrické veličiny, tedy mechanického tlaku na snímač, dojde ke skokové změně odporu sepnutím nebo rozepnutím kontaktu. Nejčastěji se používá jako takzvaný koncový spínač. Výstupní signál mechanických kontaktních snímačů nabývá pouze dvou logických úrovní „0“ a „1“. Díky jejich vlastnostem, hlavně ceny, jsou často používány v průmyslu. V mém případě jsou použity při řešení problematiky s kráčením po nerovné ploše.



Obr. 4. 1: Mechanický snímač polohy (mikrospínač) [15]

### 4.2.2 Ultrazvukové snímače polohy

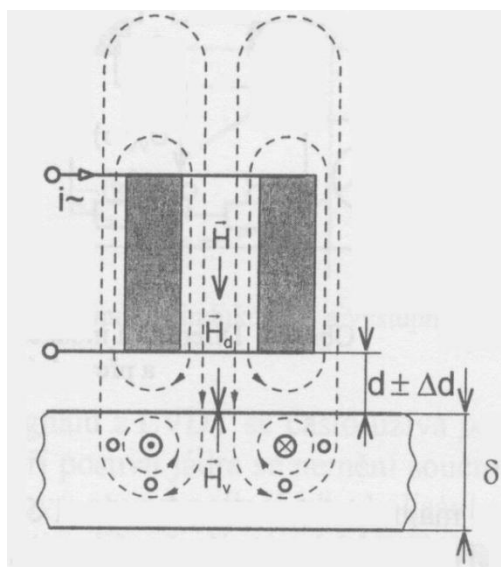
Ultrazvukové snímače polohy pracují na principu měření doby letu ultrazvukového signálu. Vysílač vyšle krátký ultrazvukový pulz. Tím je několik period ultrazvukového signálu o rezonančním kmitočtu měniče a zároveň se začne měřit čas. Po odražení signálu od překážky se vrací zpět k přijímači. Po přijetí odraženého pulzu je měření času zastaveno. Následně dochází k vyhodnocení, zda je přijatý signál opravdu echo vyslaného signálu odraženého od překážky. Pokud ano, je z rychlosti šíření zvuku v daném prostředí a doby letu signálu vyhodnocena vzdálenost překážky. Je tedy zřejmé, že přesnost je dána vlastnostmi prostředí, kterým se ultrazvuk šíří (vlhkost, teplota) a jasnosti odezvy [16].



Obr. 4. 2: Blokové schéma ultrazvukového snímače polohy [16]

### 4.2.3 Indukční snímače polohy s potlačeným magnetickým polem

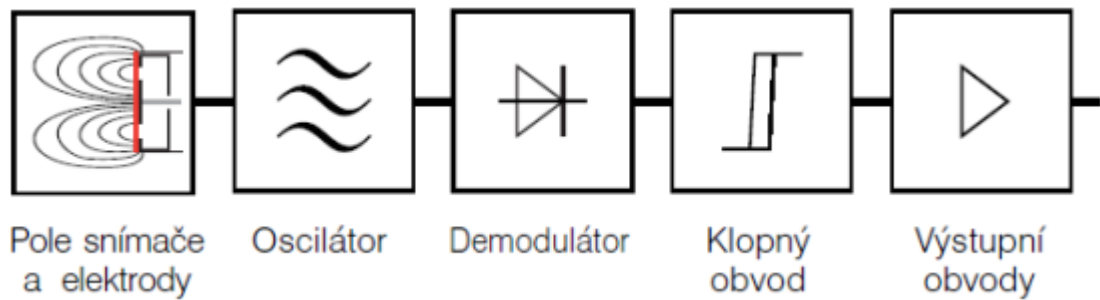
Indukční snímače s potlačeným magnetickým polem pracují na principu vzniku vířivých proudů. Ty vznikají přiložením elektricky vodivého materiálu do magnetického pole snímače  $H_{SN}$  a způsobují vznik magnetického pole  $H_{VÍŘ}$ . Směry těchto magnetických polí působí proti sobě (Lenzův zákon). Tím se zmenšuje intenzita původního magnetického pole. Tak vznikl název indukční snímače s potlačeným magnetickým polem [16].



Obr. 4. 3: Principiální schéma indukčního snímače polohy s potlačeným magnetickým polem [16]

#### 4.2.4 Kapacitní bezkontaktní snímače polohy

Kapacitní snímače pracují na principu deskového kondenzátoru, jehož jednou deskou je detekční plocha snímače a druhou je v našem případě podlaha. Na rozdíl od indukčních snímačů snímače kapacitní detekují i nevodivé materiály, je však nutné znát permitivitu tohoto materiálu.



Obr. 4. 4: Blokové schéma kapacitních bezkontaktních snímačů polohy [17]

Je tedy zřejmé, že použití indukčních a kapacitních snímačů polohy je pro kráčející roboty nevhodné. Obvykle neznáme materiál, na kterém se robot pohybuje.

#### 4.3 Vlastní řešení chůze po nerovné ploše

Robot Černá vdova má na každé z šesti končetin zabudován mechanický snímač polohy (mikrospínač). Ten slouží k detekci dotyku se zemí a posílá do řídicího mikroprocesoru informaci, zda je či není sepnut. Nejdůležitější je však změna stavu, kdy dojde k sepnutí mikrospínače. Tuto změnu stavu mikroprocesor zpracuje a přestane zvětšovat šířku pulzu PWM signálu, díky čemuž se končetina pokládala. Dále je poslán do servomotoru tak široký pulz, který odpovídá aktuální výšce končetiny při dotyku kontaktu se zemí. Podrobněji je princip řízení servomotorů popsán v kapitole 6.3 Modelářské servomotory a jejich řízení.

## 5 ZPŮSOBY BEZDRÁTOVÉ KOMUNIKACE

### 5.1 Bluetooth

Jedná se o bezdrátovou radiovou komunikaci. Její hlavní vlastnosti jsou malá velikost zařízení, nízká cena, robustní spojení a malá spotřeba energie. S tím je však úzce spjat i dosah komunikace. Bluetooth pro komunikaci používá ISM (Industrial Scientific and Medicine) pásmo na frekvenčním rozsahu 2,4 - 2,483 GHz [8].

Podle maximálního výstupního výkonu jsou definovány 3 třídy zařízení (Tab. 5.1: Třídy zařízení Bluetooth). Ty umožňují dosah spojení 10 až 100 metrů. Takto velkého dosahu spojení je však možné docílit pouze ve volném prostoru. Pokud jsou mezi komunikujícími zařízeními překážky, roste počet chybně přenesených paketů. Je tedy zřejmé, že dosah spojení v takovémto prostředí klesá [8].

Třída	Maximální výstupní výkon	Dosah
I.	10 mW (20 dBm)	cca 100 metrů
II.	2,5 mW (4 dBm)	cca 50 metrů
III. (standardní specifikace)	1 mW (0 dBm)	cca 10 metrů

Tab. 5.1: Třídy zařízení bluetooth [8]

Bluetooth zajišťuje rychlost přenosu až 433,9 kb/s v symetrickém přenosu a 723,2 kb/s v asymetrickém přenosu. Komunikace je možná jak mezi dvěma zařízeními (point-to-point), tak v síti s více zařízeními (point-to-multipoint). V takovéto síti se jedná o master-slave komunikaci. Jedno zařízení je vždy nadřazené ostatním [8].



Obr. 5.1: Bluetooth modul [9]

## 5.2 WiFi

Wifi je moderní způsob bezdrátové komunikace založený na standardu IEEE 802.11. Její hlavní výhodou je rychlost přenosu dat a flexibilita. Wifi pracuje stejně jako Bluetooth v bezlicenčním rádiovém ISM pásmu na kmitočtu 2,4 GHz a 5 GHz. Standard IEEE 802.11 obsahuje řadu specifikací, které definují kromě způsobu modulace radiového signálu také dosah spojení, rychlost přenosu dat nebo šifrování dat (Tab. 5.2: Specifikace standardu IEEE 802.11).

Standard	Rok vydání	Pásmo [GHz]	Maximální rychlost [Mbi/s]	Fyzická vrstva
původní IEEE 802.11	1997	2.4	2	
IEEE 802.11a	1999	5	54	OFDM
IEEE 802.11b	1999	2.4	11	DSSS
IEEE 802.11g	2003	2.4	54	OFDM
IEEE 802.11n	2009	2.4 nebo 5	600	MIMO OFDM
IEEE 802.11y	2008	3.7	54	
IEEE 802.11ac	2013	5	1000	MU-MIMO OFDM
IEEE 802.11ad	2014	2.4, 5 a 60	7000	

Tab. 5.2: Specifikace standardu IEEE 802.11 [10]

Základem bezdrátových sítí Wifi jsou přístupové body označovány jako AP (Access Point), které často vysílají svá SSID (Service Set Identifier). SSID slouží jako identifikátor jednotlivých sítí. Podle nich si klient může vybrat ke které síti se připojí [10].



Obr. 5.2: Wifi modul [11]

### 5.3 ZigBee

Standard označený, jako IEEE 502.15.4. Tato zařízení pracují stejně jako Bluetooth a Wifi ve frekvenčním pásmu 2,4 GHz. ZigBee moduly se používají pro jednoduchou bezdrátovou komunikaci s požadavky na nízkou spotřebu energie. Proto jsou často využívány v automatizaci u bateriově poháněných zařízení a robotů. Nízká spotřeba je vykoupena nízkou přenosovou rychlostí. Tou se však zlepšuje odolnost vůči rušení a díky tomu jsou ZigBee často využívány i v průmyslu, kde není kladen velký důraz na rychlost komunikace. Použití složitější a dražší Wifi komunikace je tedy zbytečné. ZigBee mají dobře vyřešený způsob adresování. Díky tomu je možné tyto moduly použít i pro rozsáhlejší sítě. Porovnání základních parametrů standardů používaných pro bezdrátovou komunikaci je v tabulce Tab. 5.3 [13].

Obchodní jméno Standard	GPRS/GSM 1xRTT/CDMA	Wi-Fi™ 802.11b	Bluetooth™ 802.15.1	ZigBee™ 802.15.4
Aplikační zaměření	Široké oblasti Hlas & Data	Web, Email, Video	Náhrada za kabel	Monitorování & Řízení
Systémové zdroje (paměť)	16MB a více	1MB a více	250KB a více	4KB - 32KB
Životnost baterií (dny)	1-7	0.5 - 5	1 - 7	100 – 1 000 i více
Max. velikost sítě (počet uzlů/sít')	1	32	7	65 000 ( příp. až 2 <sup>64</sup> )
Přenosová rychlost (Kb/s)	64 – 128	11 000	720	20 - 250
Komunikační dosah (m)	1 000 i více	1 - 100	1 - 10	1 – 100
Výhody	Dosažitelnost, Kvalita	Rychlost, Flexibilita	Cena, Jednoduchost	Spolehlivost, Výkon/Cena

Tab. 5.3: Porovnání základních parametrů standardů používaných pro bezdrátovou komunikaci [13]



Obr. 5.3: ZigBee modul [14]

## 5.4 VF moduly

U VF modulů pro bezdrátovou komunikaci se používají dva typy modulace signálu. Amplitudová a frekvenční. Při použití amplitudové modulace je komunikace méně odolná proti rušení. Výhodou těchto modulů je nízká cena. Nejčastěji se používají pro přenos digitálních dat logické „0“ a „1“ [12].

Frekvenčně modulovaný signál je odolnější vůči rušení, a proto dosahuje lepší spolehlivosti přenosu. Přenosové rychlosti jsou řádově vyšší než u amplitudově modulovaného signálu [12].

### 5.4.1 Komunikační modul BIM-433-F

Jedná se o miniaturní UHF (Ultra high frequency) rádiový modul. Dvojice těchto modulů společně komunikují polovičním duplexem. To znamená, že v každý časový okamžik může probíhat přenos pouze jedním směrem. Buď se data přijímají, nebo vysílají. Příjem i vysílání pracují na odlišných frekvencích. Zařízení spolu komunikují rychlostí až 40 kb/s do vzdálenosti až 120ti metrů v otevřeném prostoru. Ve vnitřních prostorech se dosah snižuje na 30 metrů [7].



Obr. 5.4: Komunikační modul BIM-433-F

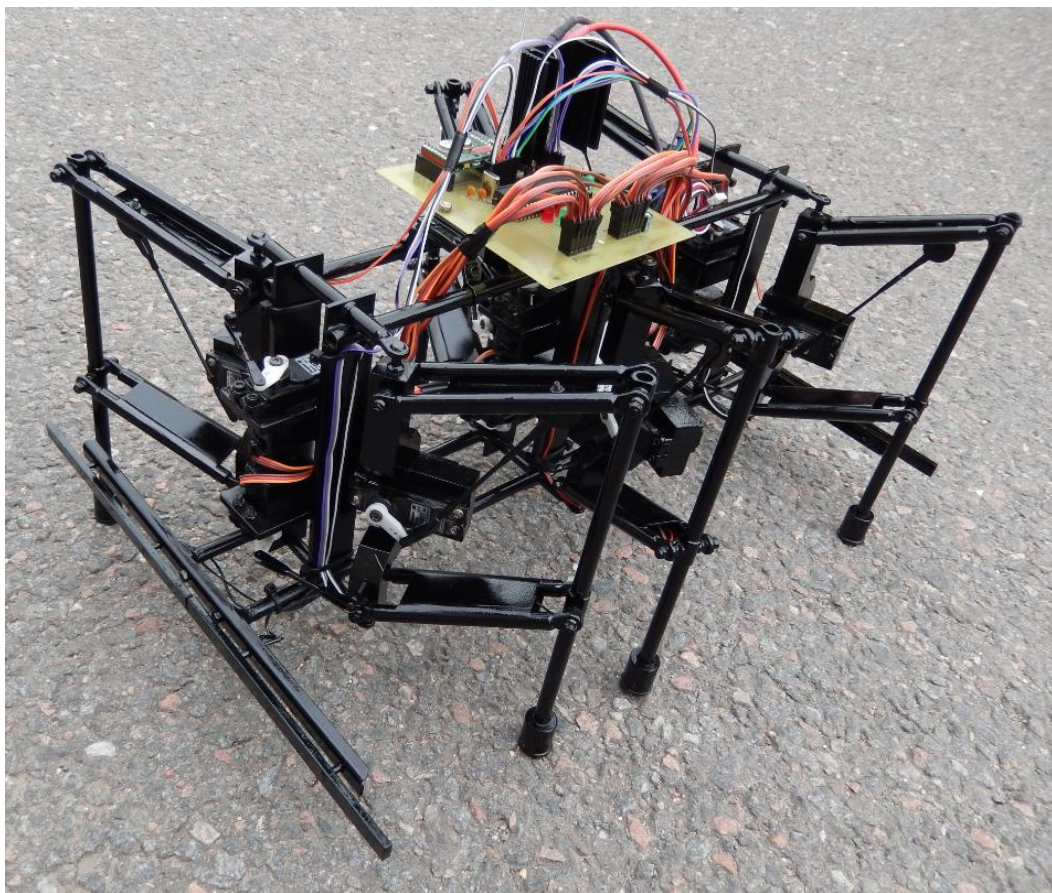


## 6 HARDWARE ROBOTA

### 6.1 Konstrukce robota

Robot Černá vdova má šest končetin se dvěma stupni volnosti, kde každý stupeň volnosti obstarává jeden servomotor. Robot je tedy osazen dvanácti servomotory. Každá končetina má uvnitř své konstrukce vedeny signálový a zemnicí vodič k mikrospínači na chodidle končetiny. Tyto mikrospínače slouží jako zpětná vazba pro řídicí mikroprocesor o dotyku se zemí. Robot je dále vybaven párem nárazníků pro detekci nepřekonatelných překážek.

Napájení zajišťuje dvoučlánková Li-pol baterie s výstupním napětím 8,4 V (při plném nabití) o kapacitě 3200 mAh. Ta je usazena do hliníkového držáku a umístěna co nejbližší těžiště robotu tak, aby byla její váha co nejlépe rozložena mezi všech šest končetin.



Obr. 6.1: Konstrukce 6-nohého kráčejiho robota Černá vdova

## 6.2 Konstrukce končetin robota

Každou z končetin řídí dva servomotory. Jeden vždy přímo na konstrukci končetiny zajišťující horizontální pohyb a druhý na rámu robota pro vertikální pohyb končetiny. Rotační pohyb hřídelí servomotorů se pomocí vhodně uchycených táhel přenese na pohyb končetiny. Dalším důležitým prvkem každé končetiny je mikrospínač zabudovaný v chodidle. Tyto mikrospínače detekují dotek chodidla se zemí. Slouží jako zpětná vazba pro řízení servomotorů vykonávajících horizontální pohyb končetin.

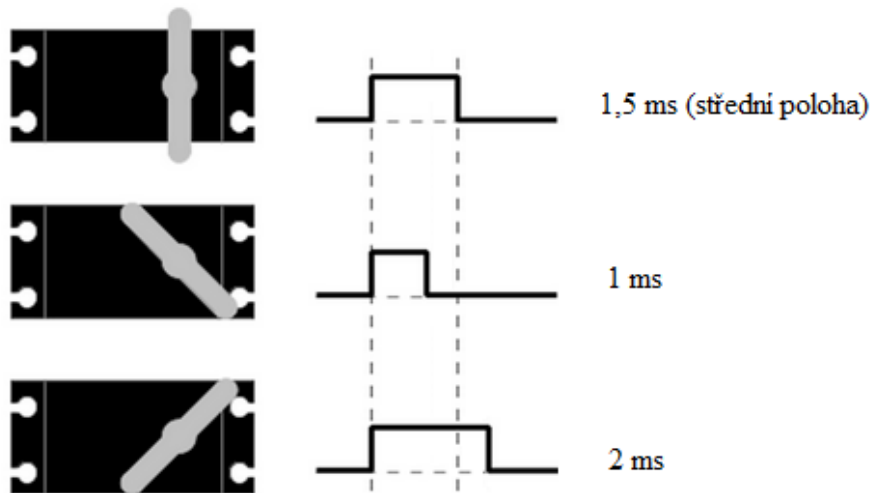


Obr. 6.5: Konstrukce končetiny

## 6.3 Modelářské servomotory a jejich řízení

Dnešní servomotory se skládají z elektromotoru, převodovky a silové elektroniky. Vstupem servomotoru je PWM (pulse width modulation) signál vytvořený například mikroprocesorem. Dále je nutné připojit napájecí (nejčastěji +5 V) a zemnicí vodič. Výstupem je rotační pohyb hřídele.

Modelářské servomotory se nejčastěji ovládají pulzy širokými 0,5 až 2,5 ms při kmitočtu 50 Hz. Šířka pulzu je přímo úměrná poloze vychýlení hřídele. 1,5 ms představuje střední polohu, 0,5 a 2,5 ms pak krajní polohy vychýlení (Obr. 6.2: Řízení modelářských servomotorů).

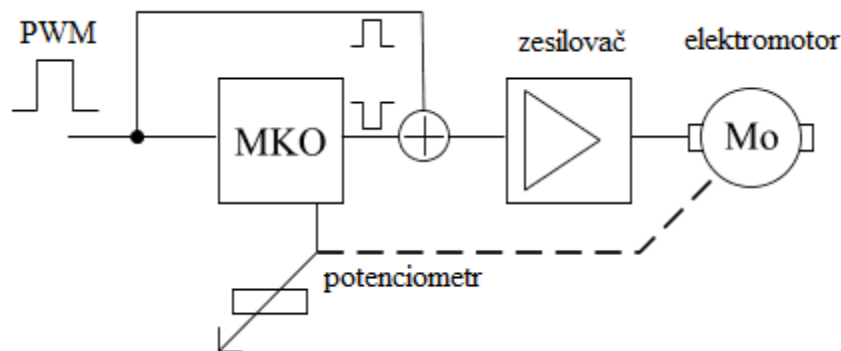


Obr. 6.2: Řízení modelářských servomotorů [18]

### 6.3.1 Princip řízení servomotorů

Řídicí elektronika servomotorů je především založena na zpětné vazbě od elektromotoru, která je realizována potenciometrem. Ten je připojen k monostabilnímu klopnému obvodu (MKO), do kterého je přiveden i vstupní řídicí signál. Podle obou těchto signálů se v MKO vytváří nový signál, který se v porovnávacím členu porovnává se vstupním signálem. Elektromotor se otáčí, dokud nejsou oba tyto signály stejné [18].

Pokud je již hřídel v požadované poloze a do servomotoru jsou neustále posílány stejně dlouhé impulzy odpovídající této poloze, elektromotor tuto polohu udržuje. Je tedy možné servomotor zatížit, aniž by došlo k pootočení hřídele.



Obr. 6.3: Blokové schéma vnitřní elektroniky servomotorů [18]

### 6.3.2 Vlastnosti použitých servomotorů

Na robotu je použito celkem dvanáct servomotorů typu Graupner C5077 jehož parametry jsou uvedeny v Tab. 6.1: Parametry servomotoru.



Obr. 6.4: Servomotor Graupner C 5077 [19]

Typ	C 5077
Rozsah napájecího napětí	4,8 - 6 V
Odebíraný proud bez zatížení	5 mA
Maximální odebíraný proud při 6 V	1010 mA
Jmenovitý moment při 6 V	51 Ncm
Úhlové rozpětí	2 x 45°
Rychlost při 6 V	0,12 s/40°
Váha	40 g
Rozměry	38 x 19 x 37 mm

Tab. 6.1: Parametry servomotoru [19]

## 6.4 Pohyb přes překážky

Ze zadání plyne, že robot musí být schopen chůze po nerovné ploše s překážkami do výšky tří centimetrů. Toho lze docílit třemi způsoby.

Přesunutím táhel od servomotorů do takové pozice, ve které bude zajištěn větší zdvih končetin při maximálním vychýlení servomotorů.

Druhý způsob řešení je zvětšit délku otočného ramene (páky) na servomotech. To lze pouze minimálně (přibližně o dva milimetry) jelikož by větším ramenům překážela hlavní konstrukce končetin a úhel možného vychýlení servomotoru by se snížil na polovinu, tedy i zdvih končetiny by byl nedostatečný.

Tato řešení vychází z předpokladu, že servomotory zajišťující zdvih robota budou na rovném terénu ve střední pozici vychýlení a při maximálním vychýlení bude změna polohy končetiny oproti střední poloze alespoň tři centimetry. To znamená, že robot bude schopen zdotat jak překážku, tak i díru do výšky/hloubky tří centimetrů. Těmito

způsoby se také zvýší odběr proudu, jelikož na servomotory bude působit větší zatěžovací moment.

Posledním možným řešením je softwarové. Robot by mohl udržovat končetiny v takové poloze, ve které by byl schopen vždy položit nohu na překážku vysokou až tři centimetry. Pokud by robot narazil na díru, musel by se přikrčit tak, aby mohl končetinu v dalším pokusu položit do hloubky až tři centimetrů a zjistit tak, jestli dokáže dírou bezpečně projít. Toto řešení nevyžaduje žádné hardwarové úpravy, ale značně by se ztížilo řízení robota a v hodně členitém terénu by došlo i k značnému zpomalení pohybu.

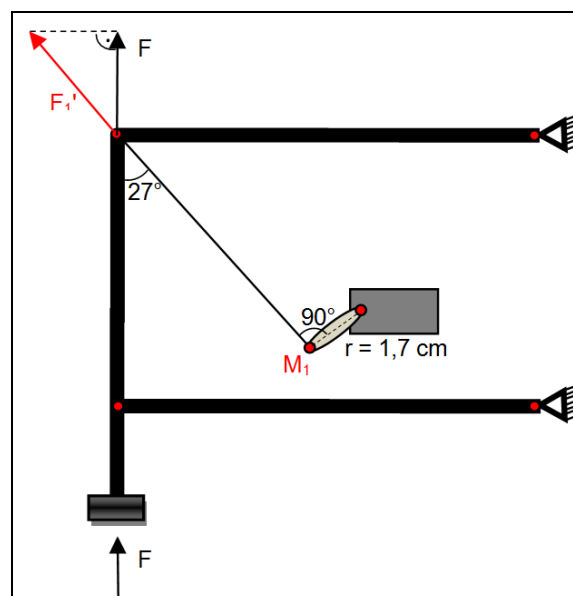
Ze dvou možných způsobů realizace chůze přes překážky vyplívajících z této kapitoly jsem se rozhodl pro změnu umístění uchycení táhel.

### 6.4.1 Uchycení táhel končetin

Při původní konstrukci končetin byl robot schopen docílit maximálního zdvihu 5 cm mezi krajními polohami vychýlení hřídelí servomotorů. Z toho vyplývá, že pro bezpečné překonání jak překážky vysoké až 3 cm, tak i díry do hloubky až 3 cm by bylo nutné softwarové řešení.

Po úpravě uchycení táhel končetin se nepatrně změnily krajní polohy vychýlení hřídelí servomotorů, zvětšil se zatěžovací moment jednotlivých servomotorů, ale bylo docíleno zdvihu až 7 cm. To již vyhovuje pro překonání požadované výšky překážek i bez nutnosti složitějšího softwarového řešení.

#### Výpočet maximálního zatěžovacího momentu působícího na servomotory při původní konstrukci končetin



Obr. 6.5: Původní konstrukce končetin

Z hmotnosti a gravitačního zrychlení jsem určil celkovou sílu  $F_{celková}$ , která působí na všechny končetiny. Hmotnost robota je 2 kg.

$$F_{celková} = m \cdot g = 2 \text{ kg} \cdot 9,81 \text{ ms}^{-2} = 19,62 \text{ N} \quad (1)$$

Největší zatížení servomotorů je při způsobu chůze tři nohy v záběru. Maximální moment se tedy bude počítat právě pro tento způsob kráčení. Z kapitoly rozbor možností chůze 6-nohých kráčejících robotů vyplývá, že robot bude v určitých okamžicích stát pouze na třech končetinách. Proto je maximální možná síla působící na jednu končetinu  $F$  třetina síly celkové  $F_{celková}$ .

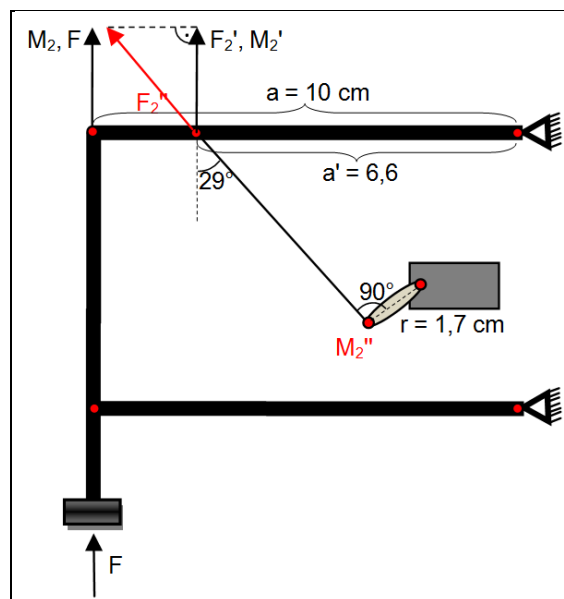
$$F = \frac{F_{celková}}{3} = 6,54 \text{ N} \quad (2)$$

Nyní je nutné přepočítat sílu  $F$  rozkladem sil na sílu  $F'_1$ , pomocí které se následně určí výsledný zatěžovací moment  $M_1$  (Obr. 6.5: Původní konstrukce končetin).

$$F'_1 = \frac{F}{\cos(\varphi)} = \frac{6,54 \text{ N}}{\cos(27^\circ)} = 7,34 \text{ N} \quad (3)$$

$$M_1 = F'_1 \cdot r \cdot \sin(\alpha) = 7,34 \text{ N} \cdot 1,7 \text{ cm} \cdot \sin(90^\circ) = 12,48 \text{ Ncm} \quad (4)$$

### Výpočet maximálního zatěžovacího momentu působícího na servomotory po upravení konstrukce končetin



Obr. 6.6: Upravená konstrukce končetin

Přesunutím táhla je výpočet o trochu složitější. Způsob kráčení se nemění a proto je i síla  $F$  stejná jako v předchozím případě. Nejprve je nutné sílu  $F$  přepočítat pomocí momentové věty na sílu  $F'_2$ . Tu lze již přepočítat pomocí rozkladu sil na sílu  $F''_2$ , která působí přes táhlo na páku servomotoru. Posledním krokem je výpočet výsledného zatěžovacího momentu  $M''_2$  (Obr. 6.6: Upravená konstrukce končetin).

$$M_2 = M'_2 \quad (5)$$

$$F \cdot a = F'_2 \cdot a' \quad (6)$$

$$F'_2 = F \cdot \frac{a}{a'} = 6,54 \text{ N} \cdot \frac{10 \text{ cm}}{6,6 \text{ cm}} = 9,91 \text{ N} \quad (7)$$

$$F''_2 = \frac{F'_2}{\cos(\varphi)} = \frac{9,91 \text{ N}}{\cos(29^\circ)} = 11,33 \text{ N} \quad (8)$$

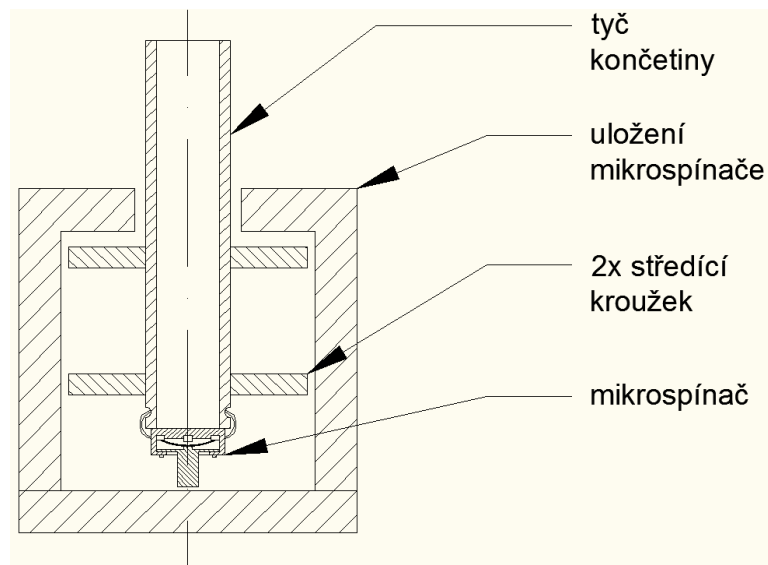
$$M''_2 = F''_2 \cdot r \cdot \sin(\alpha) = 11,33 \text{ N} \cdot 1,7 \text{ cm} \cdot \sin(90^\circ) = 19,26 \text{ Ncm} \quad (9)$$

Jmenovitý zatěžovací moment použitých servomotorů je 51 Ncm. Po úpravě vyšel maximální zatěžovací moment 19,26 Ncm. Je tedy zřejmé, že po změně uchycení táhel nedojde k překročení jmenovitého momentu a servomotory toto zatížení zvládnou. Úprava se projeví pouze zvýšením odebíraného proudu. Ten při měření nepřesáhl hodnotu vyšší než 3,5 A, což je také v toleranci.

## 6.4.2 Chodidla končetin

Jak už bylo řečeno, v chodidlech končetin jsou zabudovány mikrospínače pro detekci doteku se zemí. Pro výrobu uložení mikrospínačů byla vybrána silikonová tyč, a to zejména kvůli malé hmotnosti a dostatečné poddajnosti materiálu. Chodidlo se skládá z mikrospínače, jeho uložení a dvou kroužků. Spodní kroužek slouží zejména jako středící, aby se chodidlo nelámalo. Horní definuje vůli v chodidle. Při zdvižené noze zajistí, že uložení chodidla, klesne pouze na pozici, v níž je mikrospínač rozepnut.

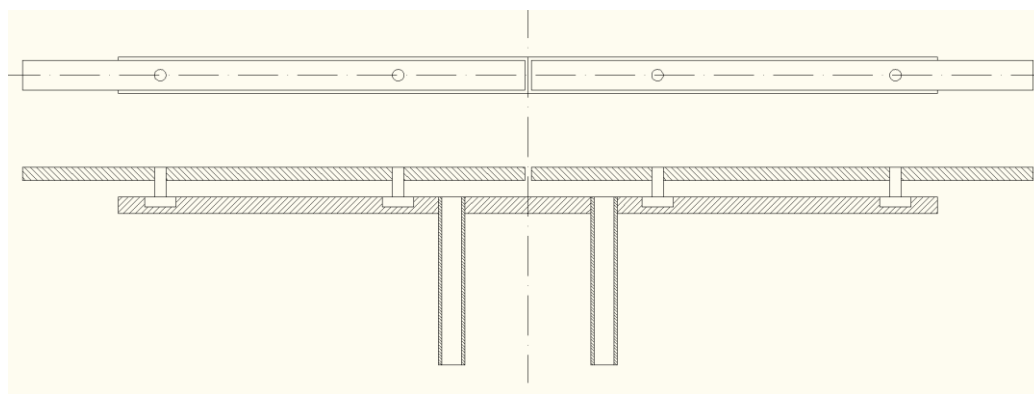




Obr. 6.7: Řez chodidla

## 6.5 Detailní konstrukce nárazníků

Oba nárazníky nesou čtyři mikropínače, které posílají do řídicího mikroprocesoru informaci o nárazu do překážky. Hmatníky mikropínačů jsou propojeny tak, že robot detekuje i náraz do tenké překážky. Propojeny jsou vždy po dvojicích. Je tedy možné rozlišit, zda na překážku narazila levá či pravá polovina robota. To by mohlo vést k dokonalejšímu mapování terénu. Nosná část nárazníku je vyrobena z hliníkového plechu ohnutého do tvaru písmene U. K té jsou mikropínače přilepeny.



Obr. 6.8: Řez nárazníku

## 6.6 Řídicí elektronika

Elektronika robota se skládá ze dvou DPS (deska plošných spojů). Obě jsou vybaveny komunikačními moduly BIM-433-F od výrobce Radiometrix zajišťující bezdrátovou komunikaci s robotem.

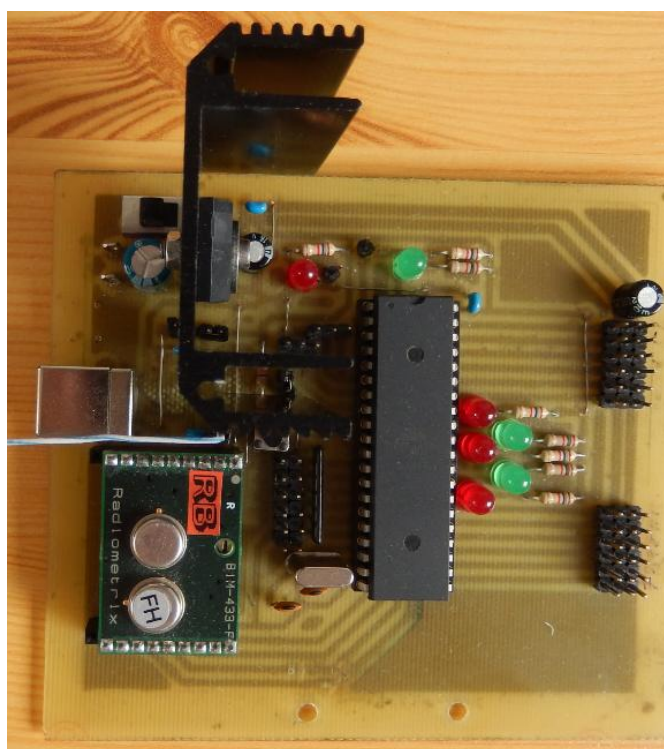


První deska plní funkci pouze přijímače/vysílače pro bezdrátovou komunikaci a je přes USB propojena přímo s počítačem (Obr. 6.9: DPS bezdrátové komunikace).



Obr. 6.9: DPS bezdrátové komunikace

Druhá deska, představující řídicí jednotku robota, je již poněkud složitější. Mozkem celého zařízení je mikroprocesor ATmega 162 pracující na kmitočtu 16 MHz. Ten řídí všech dvanáct servomotorů a přijímá informace celkem od čtrnácti snímačů. Dále zpracovává a vyhodnocuje povely přijímané po sériové lince.



Obr. 6.10: Řídicí jednotka robota

Při výběru procesoru byly kladeny požadavky na dostatečný počet vstupně/výstupních portů rychlost procesoru, velikost programovatelné paměti flash

a hlavně přítomnost dvou sériových linek. Jedna sloužící pro bezdrátovou komunikaci a druhá pro programování přes USB s využitím bootloaderu. Bylo by však vhodné nahradit stávající mikroprocesor jiným s větší pamětí programu, pro aplikaci více způsobů kráčení.

Napájení zajišťuje dvoučlánková Li-Pol baterie poskytující napětí 8,4 V. To je pomocí lineárního stabilizátoru sníženo na 5 V a dále filtrováno elektrolytickými a keramickými kondenzátory.

### Ztráty na stabilizátoru

Vstupní napětí:  $U_{vstup} = 8,4 V$

Výstupní napětí:  $U_{výstup} = 5 V$

Maximální zatěžovací proud:  $I_{zat} = 5 A$

$$P = (U_{vstup} - U_{výstup}) \cdot I = (8,4 V - 5 V) \cdot 5 A = 17 W \quad (10)$$

Na takto velký zatěžovací proud je stavěn lineární stabilizátor. Ztráty 17 W jsou tedy spočítány v mezním případě. Zařízení ve skutečnosti odebírá méně proudu, a proto jsou ztráty menší. Jsou však stále dost velké a proto byl stabilizátor vybaven hliníkovým chladičem s  $R_{th} = 6,5 K/W$ .



Obr. 6.11: Hliníkový chladič V7132

## 7 SOFTWARE

Celkové softwarové vybavení robota představují tři programy. Prvním je program pro řízení robota. Dále uživatelské rozhraní spouštěné z PC pro ovládání a programování robota. Posledním je bootloader uložený na konci paměti programu v mikroprocesoru sloužící k jeho přeprogramování.

### 7.1 Hlavní řídicí program

Z důvodu omezené velikosti paměti je implementován pouze jeden z možných způsobů kráčení. Tím je kráčení způsobem pět noh v záběru. Program využívá dvou přerušení. Jedním je přerušeni od sériové linky, ve kterém jsou přijímány instrukce o změnách směru pohybu. Druhým je přerušeni od čítače/časovače. To slouží k vytváření PWM signálu, kterým se řídí servomotory. V hlavní smyčce programu je vytvořen algoritmus kráčení pro různé směry pohybu.

#### 7.1.1 Důležité funkce a principy programu

##### Určení směru pohybu z přijaté zprávy

Po vybrání směru pohybu v uživatelském rozhraní, je z PC odeslán odpovídající kód (kombinace tří znaků). Příjem tohoto kódu se provede v obsluze přerušeni od sériové linky. Jakmile je zpráva kompletní, dojde k zavolání funkce *porovnali*. Ta vrací celočíselnou hodnotu, která je v hlavní smyčce programu přiřazena určitému směru pohybu.

- hlavička funkce *porovnali*: `int porovnali(void);`

##### Horizontální pohyb končetin

Horizontální pohyb končetin zajišťuje funkce *Zdvih*. V té se nastavují nové hodnoty pro servomotory. Funkce má šest vstupních parametrů typu `char` (*SH1* až *SH6*). Každá odpovídá požadovanému vychýlení určité končetiny. Předávány jsou hodnoty v rozsahu od 16 do 48. Hodnotě 16 odpovídá délka PWM pulzu 0,8 ms a hodnotě 48 2,4 ms. Rozlišení délky pulzu je tedy 50  $\mu$ s.

- hlavička funkce *Zdvih*: `void Zdvih(char SH1, char SH2, char SH3, char SH4, char SH5, char SH6);`

##### Vertikální pohyb končetin

Tato funkce je velice podobná funkci *Zdvih*. Má stejný počet vstupních parametrů. Ty nyní odpovídají skupině servomotorů vykonávající posuvný pohyb robota. Jediným rozdílem je rozsah možného vychýlení servomotorů. Ten může nabývat hodnot od 22

do 38. Tomu odpovídají pulzy široké 1,1 ms až 1,9 ms. Větší rozsah nedovoluje hardwarová konstrukce robota.

- hlavička funkce *Posun*: `void Posun(char SH7, char SH8, char SH9, char SH10, char SH11, char SH12);`

### **Zajištění opakovatelného překonávání překážek**

Schopnost překonávat i více po sobě jdoucích překážek zajišťuje funkce pojmenovaná jako *Straightening*. Tato funkce nemá žádné vstupní ani výstupní parametry. Pracuje přímo s hodnotami uloženými v proměnných *SERVOI-12* určující zdvih a vychýlení jednotlivých končetin. Výsledná úprava polohy končetin je určena z nejnižší a nejvyšší položené končetiny. Mohou nastat tři případy. První a druhý případ jsou velice podobné. Pokud jsou obě uvažované končetiny pod, nebo nad střední hodnotou, výsledný zdvih se určí z rozdílu bližší končetiny k střední poloze se střední hodnotou (střední poloha odpovídá hodnotě 32). Třetí případ nastane, pokud je nejnižší končetina pod střední polohou a nejvyšší nad střední polohou. V tomto případě se výsledná úprava polohy všech končetin provede tak, aby rozdíl mezi oběma končetinami a střední hodnotou byl stejný. Díky této funkci se robot po překonání překážky dostane do pozice, ze které je znovu schopen překonávat novou překážku.

- hlavička funkce *Straightening*: `void Straightening(void);`

### **Příprava k pohybu**

Přípravu k pohybu obstarává funkce *Inicialization*. Při každé změně pohybu se postará o optimální nastavení vertikální polohy končetin, aby při prvním kroku opravdu došlo k požadovanému směru pohybu. Je také volána po resetu procesoru ještě před vniknutím do hlavní smyčky programu. Po resetu je nejpravděpodobnější chůze dopředu. Proto jsou končetiny přednastaveny na chůzi vpřed.

- hlavička funkce *Inicialization*: `void Inicialization(char X);`

### **Zamezení naklápění robota**

K naklápění dochází nepřesným určením polohy, ve které došlo ke kontaktu se zemí. Pro zamezení naklápění robota z důvodů malého rozlišení vychýlení servomotorů a chybné detekce dotyku (zvláště u předních, méně zatížených končetin), jsou použity následující principy.

Jelikož při pokládání končetiny není jasné, kde přesně mezi dvěma stupni vychýlení došlo ke kontaktu se zemí, servomotor v prvním kroku udržuje hodnotu, ve které již končetina dolehla na zem. V následujícím kroku, kdy je daná končetina opět pokládána se použije poslední hodnota vychýlení, ve které ještě nedošlo ke kontaktu se zemí.

Druhá chyba, je způsobena samotnou konstrukcí robota. Jelikož má každá končetina pouze dva stupně volnosti, chodidlo opisuje trajektorii kružnice. Při doteku vzniká nežádoucího tření mezi středními kroužky a uložením chodidla, a je zapotřebí

větší síly k promáčknutí mikrospínače v chodidle. Větší síla je získána přizvednutím celého robota. Proto je detekce kontaktu se zemí prováděna dvěma způsoby. Jedním je již zmiňované promáčknutí mikrospínače na právě pokládané končetině. Druhým způsobem se vyhodnotí jako kontakt se zemí, pokud došlo k rozepnutí některého z mikrospínačů na ostatních chodidlech.

Samotná tato řešení však zcela nevyřešila problém s naklápěním. Proto jsem vytvořil funkci *Korekce*, která porovná předchozí a aktuální hodnotu vychýlení při kontaktu se zemí. Pokud je jejich rozdíl menší než tři použije se předchozí hodnota vychýlení. To zamezí naklápění robota, ale také zhoršilo rozlišení výšky překážek, které lze detekovat.

- hlavička funkce *Korekce*: `void Korekce(int servo);`

## 7.1.2 Přerušeni od čítače/časovače 0

### Nastavení registrů

Pro správnou funkci přerušeni od čítače/časovače 0 a jeho vyvolání každých 50  $\mu$ s je potřeba přednastavit registry TCCR0 a TCNT0. Do registru TCCR0 je na bit označený jako CS01 zapsána jednička. To znamená, že čítač/časovač 0 bude pracovat s předděličkou osmi. Díky tomu je možné nastavit hodnotu registru TCNT0 tak, aby vznik přerušeni měl potřebnou periodu. Hodnota registru TCNT0 (počáteční stav čítání) se vypočítá podle vzorce (11).

$$TCNT0 = 2^R - \frac{f_{osc} \cdot t}{n} \quad (11)$$

- n ..... použitá předdělička (1, 8, 64, 256, 1024)  
R ..... kolika bitový je čítač (8 b, 16 b)  
 $f_{osc}$  ..... kmitočet oscilátoru (16 MHz)  
t ..... požadovaná perioda vzniku přerušeni (50  $\mu$ s)

$$TCNT0 = 2^8 - \frac{16 \cdot 10^6 \cdot 50 \cdot 10^{-6}}{8} = 156 \quad (12)$$

Nakonec je nutné v registru TIMSK nastavit bit TOIE0, kterým se povolí přerušeni od čítače/časovače 0.

### Obslužná rutina přerušeni

Vždy při vzniku přerušeni je inkrementována proměnná *cc*. Jakmile proměnná *cc* odpovídá hodnotě 400, uběhlo právě 20 ms což je perioda, na které pracují servomotory. V tu chvíli jsou nastaveny řídicí piny všech servomotorů na „1“ a *cc* je vynulováno.

V případě, že se nohy zvedají nebo dochází k pohybu robota, je pouze hodnota v proměnné *cc* porovnávána s proměnnými *SERVO1-12*. Pokud dojde ke shodě je vynulován řídicí pin odpovídajícího servomotoru. Do servomotoru je tedy vyslán pulz, který vychyluje hřídel do požadované polohy. Na vzdálenosti počáteční a cílové polohy hřídele závisí počet pulzů potřebných k odpovídajícímu vychýlení.

Při pokládání končetin je nutné zastavit pokládání ve chvíli, kdy dojde ke kontaktu chodidla se zemí. Do servomotorů pokládajících končetiny je proto vždy posílán 3x stejný pulz, který je o 50  $\mu$ s delší než předchozí trojice pulzů a provádí se testy na detekci kontaktu se zemí. Pokud ke kontaktu nedošlo, je v dalších třech periodách posílán do servomotoru opět pulz o 50  $\mu$ s delší. Takto se pokračuje, dokud nedojde k doteku chodidla se zemí. V tu chvíli se pulz pro servomotor pokládající danou končetinu přestane zvětšovat. Pokud se noha dostane až do krajní polohy aniž by došlo ke kontaktu, narazil robot na příliš hlubokou díru a jeho pohyb je zastaven.

Zároveň je každých 20 ms prováděn test nárazníků. Pokud je nárazníky detekována překážka je pohyb robota také zastaven.

### 7.1.3 Přerušování od sériové linky 1

#### Nastavení registrů

Pro navázání komunikace po sériové lince 1 je nutné nastavit registry UCSR1B, UCSR1C a UBRR1L. Pro povolení vysílání i přijímání dat jsou v registru UCSR1B nastaveny bity označené jako RXEN1 a TXEN1. Dále je nutné nastavit formát dat. Ten se standardně nastavuje na 8 b. To se provede nastavením bitů UCSZ10, UCSZ11 a bitem URSEL1 v registru UCSR1C. Nakonec je nutné nastavit registr UBRR1L podle požadované rychlosti přenosu. Výpočet se provede podle rovnice (13).

$$UBRR1L = \frac{f_{osc}}{Baudrate \cdot 16} - 1 = \frac{16 \cdot 10^6}{9600 \cdot 16} - 1 = 103 \quad (13)$$

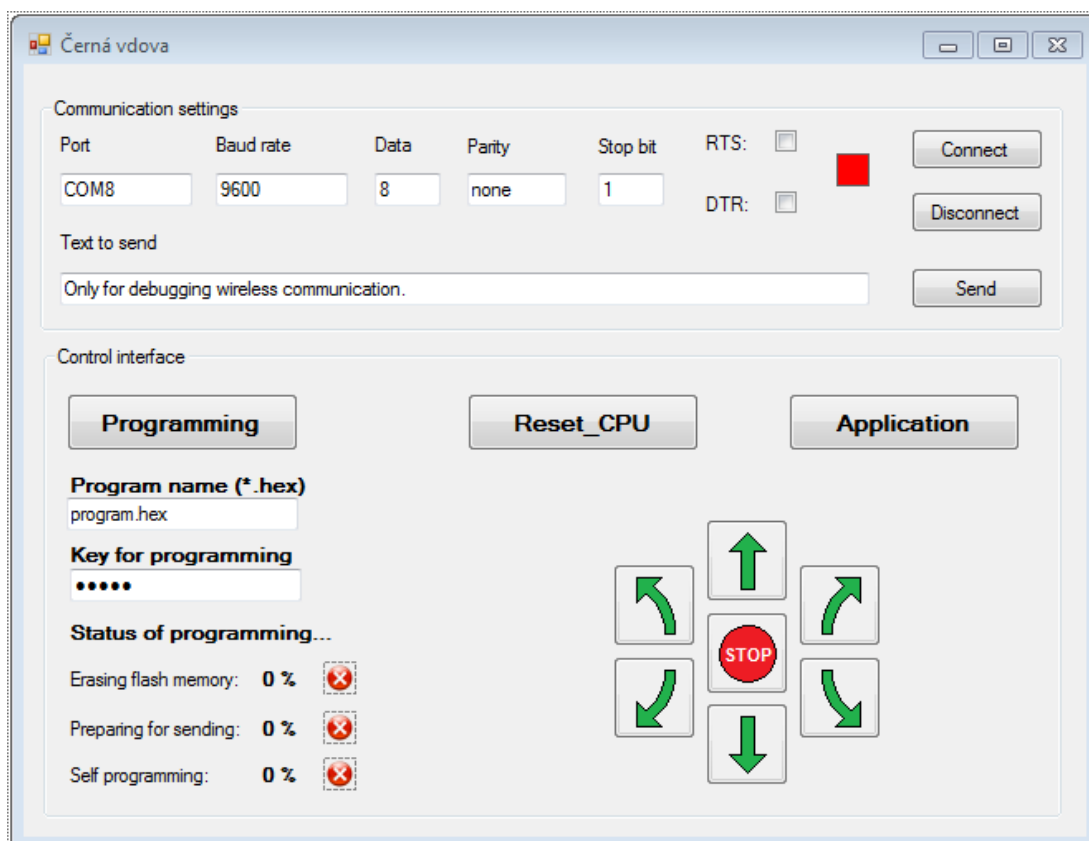
Posledním krokem je povolení přerušování od sériové linky 1. To se provede nastavením bitu RSCIE1 v registru UCSR1B.

#### Obslužná rutina přerušování

V obsluze přerušování se vždy čeká na příchod znaku 'd'. Po něm následují další dva znaky. V těchto třech znacích je zakódován směr pohybu. Po přijetí posledního znaku je zavolána funkce *porovnaní*, ve které je kód rozkódován a do proměnné *vysledek* se uloží celočíselná hodnota odpovídající jednomu ze směrů pohybu. Před zavoláním funkce *porovnaní* je aktuální hodnota v proměnné *vysledek* uložena do proměnné *last*. Ta je v hlavní smyčce programu využívána k zamezení nadbytečného používání funkce *Inicializace*. Například při změně pohybu z chůze vpřed na chůzi vzad je inicializace zbytečná, protože by končetiny vůbec nezměnily svoji polohu.

## 7.2 Uživatelské rozhraní

Uživatelské rozhraní je vytvořeno ve vývojovém prostředí Visual Studio 2010 v jazyce C++.



Obr. 7.1:Uživatelské rozhraní

Z pohledu uživatele je rozděleno do dvou částí. První část nazvaná *Communication settings* slouží pro nastavení komunikace po sériové lince včetně připojení i odpojení od sériového portu. Jelikož jsou pro bezdrátovou komunikaci použity moduly, které umí komunikovat pouze polovičním duplexem, je pomocí RTS signálu přepínán komunikační modul z přijímače na vysílač a naopak. Text box *Text to send*, tlačítka *Send* a check boxy RTS a DTR byly používány pouze pro ladění komunikace a ve výsledném programu nejsou využity. Po stisknutí tlačítka *Connect* se uživatelské rozhraní snaží otevřít port se jménem zapsaným do text boxu *Port*. Pokud je tento port otevřen (v mém případě COM8) došlo k navázání komunikace mezi PC a deskou plošného spoje připojenou k počítači. Signalizace umístěna nalevo od tlačítek *Connect* a *Disconnect* změní svou barvu z červené na zelenou.

Pokud k navázání komunikace nedojde, signalizace zůstane červená a dojde k vyvolání výjimky. Zobrazí se message box s upozorněním na vzniklou chybu.

Druhá část nazvaná *Control interface* slouží k řízení i bezdrátovému programování robota. Pro výběr mezi programovacím a aplikačním módem jsou zde tlačítka *Programming* a *Application*. Dále je zde nezbytně nutné tlačítko *Reset\_CPU*. Jelikož reset vektor je v mikroprocesoru nastaven na začátek bootloADERU (adresa 0x3800) je rozhodování zda se má programovat nebo spustit program implementováno v bootloADERU. V levé části, která je věnována programovacímu módu jsou dva text boxy pojmenované jako *Program name* a *Key for programming*. Ty slouží pro zápis jména programu, který chceme naprogramovat do robota a vstupní klíč, který programování povoluje. Přístupový klíč je nastaven na pět nul (00000).

### 7.2.1 Popis aplikačního módu

Pokud signalizace svítí zeleně je vše připraveno ke komunikaci. Kliknutím na tlačítko *Application* dojde k opuštění bootloADERU a robot je připraven přijímat povely o směru chůze pomocí šesti směrových tlačítek. Každé tlačítko má svůj specifický kód, který je po stisknutí poslán do robota. V robotu dojde k vyhodnocení tohoto kódu a určí se směr kráčení. Na stejném principu funguje i tlačítko *STOP*. Po jeho stisknutí dojde k zastavení pohybu.

### 7.2.2 Popis programovacího módu

Stisknutím tlačítka *Programming* se ověří klíč v text boxu *Key for programming*. Pokud je správný, odešle se do robota zpráva obsahující pět nul. Tu robot zpracuje a postoupí se hlouběji do bootloADERU. Pokud je klíč špatný, zobrazí se message box, který na chybu klíče upozorní uživatele.

Prvním krokem je mazání aktuálního obsahu paměti. To se provede posláním zprávy obsahující pět dvojek. Po dokončení mazání je z robotu poslána odpověď, že se může pokračovat dalším krokem programování. Odpověď je reprezentována třemi znaky '\r'.

Ve druhém kroku se připravují data k odeslání. Z text boxu *Program name* je vyčteno jméno programu a tento soubor je otevřen. Program je ve formátu hex. V tomto formátu každý řádek začíná dvojtečkou a za ní následují dva znaky určující počet bytů na jednom řádku. Cyklem přes všechny řádky je spočítán skutečný počet bytů, ze kterého je následně určen počet stránek. Dále je vytvořeno pole znaků pojmenované jako *program*. Do něho jsou uloženy všechny datové byty z načteného souboru.

Následuje cyklus přes všechny stránky programu. Ten se opakuje, dokud nejsou všechny stránky v pořádku přijaty bootloADEREM. Odesláním zprávy obsahující pět trojek se v bootloADERU přistoupí do programovací části. Před samotným posláním stránky je ještě poslána zpráva nesoucí informace o stránce. Těmito informacemi jsou pořadí stránky, ze kterého se v bootloADERU vypočítá adresa začátku stránky a počet posílaných bytů. Dále je vytvořeno pole znaků pojmenované jako *stranka*. Ta je naplněna daty a na konci cyklu odeslána. Po odeslání se čeká na odpověď od



bootloaderu, zda stránka dorazila v pořádku. Pokud ano, pokračuje se další stránkou v pořadí. V opačném případě se stránka odesílá znovu.

Pokud jsou již všechny stránky bezchybně přijaty, je odeslána zpráva obsahující čtyři čtyřky. Po jejím přijetí se opouští bootloader a začne se vykonávat nově nahraná aplikace.

Vývojový diagram programovacího módu je v příloze A.

### 7.2.3 Tvar posílané zprávy

Každá zpráva, ať už jde o kód stisku tlačítka nebo stránky programu musí obsahovat určitý rámec pro seřízení komparátoru na přijímacím modulu BIM-433-F. Rámec je tvořen dvěma byty, ve kterých se pravidelně střídá jednička s nulou v binárním tvaru (například znak 'U' = 01010101b). Díky nim dojde k seřízení komparátoru na přijímači. Dále jsou tyto dva byty doplněny o dva byty nul a dva byty jedniček, které zajistí synchronizaci. Pro zajištění správné funkce komunikace je dále nutné aby čtyři po sobě přijaté byty měly vyvážený počet nul a jedniček. Proto je po každém bytu dat posílán inverzní byte, který slouží jednak pro správnou funkci komunikace, ale také je využit při kontrole správnosti přijatých dat. Příklad posílané zprávy obsahující slovo DOG je na Obr. 7.2.

'U'	'U'	0	0	255	255	'D'	255-'D'	'O'	255-'O'	'G'	255-'G'
-----	-----	---	---	-----	-----	-----	---------	-----	---------	-----	---------

Obr. 7.2: Příklad zprávy obsahující slovo DOG

## 7.3 Bootloader

Bootloader sloužící k přeprogramování paměti programu. Byl použit originální bootloader vytvořený společností Atmel Corporation, který jsem si podle svých potřeb upravil.

Program je rozdělen do pěti na sobě závislých souborů:

- serial.h - hlavičkový soubor pro serial.c
- serial.c - soubor se zdrojovými kódy pro komunikaci po sériové lince
- flash.h - hlavičkový soubor definující všechny potřebné funkce pro práci s pamětí flash
- defines.h - hlavičkový soubor definující procesor, velikost stránek, velikost vyhrazeného místa pro bootloader, registry sériové linky a rychlost komunikace
- main.c - hlavní kód celého bootloderu, využívá všechny předchozí soubory

### 7.3.1 Princip funkce bootloaderu

Mezi nejdůležitější prvky programu patří proměnné *priznak* a *address* s funkcemi *rechcar()*, *sendchar()* a *BlockLoad()*. Pomocí proměnné *priznak* se určuje, která část programu se bude vykonávat. Do proměnné *address* se ukládá začátek stránky v paměti programu, kam se má přijatá stránka uložit. Funkce *rechcar()* slouží pro příjem všech bytů ze sériové linky a funkce *sendchar()* pro odesílání bytů po sériové lince. Ve funkci *BlockLoad()* se přijímají data, která jsou následně nahrána do paměti programu.

Vyvoláním resetu mikroprocesoru pomocí tlačítka na desce plošného spoje, nebo pomocí watchdog timeru, který se spustí z uživatelského rozhraní, dojde ke skoku na reset vektor. Zde začíná bootloader. Po vstupu do bootloderu se čeká na příjem nové instrukce. Tou může být v tuto chvíli pouze kombinace pěti po sobě jdoucích jedniček nebo nul. Pokud bude přijato pět jedniček, nastaví se proměnná *priznak* na hodnotu 1 a tím dojde k opuštění bootloderu a skoku na adresu 0x0000, kde začíná aplikační část paměti programu. Pokud přijde kombinace pěti nul, znamená to, že se chystá přeprogramování paměti.

Prvním krokem při přepisování paměti je mazání aktuálního obsahu. To se provede, pokud je proměnná *priznak* nastavena na hodnotu dva. K nastavení dojde opět přijetím kombinace pěti znaků, nyní dvojek. Po dokončení mazání je zpět odeslána zpráva, že mazání je dokončeno.

Druhým krokem je příjem zprávy, která nese informace nutné pro programování jedné stránky. K příjmu této zprávy je nutné nejprve přistoupit do této části programu. To se provede opět pomocí proměnné *priznak*, která musí být nyní nastavena na hodnotu tři. Nastavení se provede stejným způsobem jako v předchozích případech, přijetím kombinace pěti znaků (trojek). Přijatá zpráva nese informace o velikosti posílaného bloku bytů a adrese, od které se mají přijímaná data ukládat do paměti. Následuje zavolání funkce *BlockLoad()* s těmito parametry.

Ve funkci *BlockLoad()* se již přijímá skutečný program. Byty jsou ve smyčce přijímány a ukládány do bufferu. Každý přijatý byte dat je v zápětí doplněn inverzním bytem. Tyto dva byty se porovnají, a pokud se první nerovná inverzi druhému, je to vyhodnoceno jako chyba v komunikaci. Po odeslání celé stránky se buď stránka pomocí SPM instrukcí nahraje do paměti programu a pokračuje se opět druhým krokem s následující stránkou, nebo došlo k chybě v komunikaci a je odeslána žádost o opětovné zaslání stejné stránky. Takto se pokračuje, dokud nejsou naprogramovány všechny stránky.

Posledním krokem je nastavení proměnné *priznak* na hodnotu čtyři, což znamená opuštění bootloderu a skok na začátek aplikační části v paměti programu.

## 8 ZÁVĚR

Výsledkem této práce je 6-nohý kráčejíci robot Černá vdova schopný překonávat překážky do výšky 3 cm, využívající způsob kráčení pět noh v záběru. Součástí je také bootloader, pomocí kterého je možné robota bezdrátově programovat. Dále je vytvořeno uživatelské rozhraní pro bezdrátové ovládání i programování.

Z provedeného rozboru jsou pro šestinohé kráčejíci roboty možné tři způsoby kráčení. Jsou jimi tři, čtyři a pět noh v záběru, kde s větším počtem noh v záběru roste nosnost a stabilita na úkor složitosti řízení a rychlosti pohybu. Ta se snižuje, jelikož algoritmus řízení má více stavů. Po dohodě s vedoucím práce byl do robota Černá vdova implementován způsob kráčení pět noh v záběru.

Pro pohyb přes překážky vysoké až 3 cm bylo nutné provést některé hardwarové úpravy. Byla přesunuta uchycení táhel od servomotorů tak, aby se zvýšil maximální zdvih končetin z původních 5 cm na 7 cm. Díky tomu bylo možné jednodušší řízení robota. Další zásadní úpravou bylo vyrobení nových chodidel s uloženými mikropřínači pro kontrolu kontaktu končetin se zemí a nárazníků pro detekci nepřekonatelných překážek.

Pro pohyb robota byl vytvořen řídicí program. Ten využívá dvou přerušení. Přerušení od sériové linky pro příjem zpráv, v nichž jsou zakódované informace o změnách směru pohybu robota. Druhé přerušení je od čítače/časovače. Jedná se nejdůležitější část programu. Vzniká každých 50 us a slouží k vytváření PWM signálu, kterým se řídí servomotory. V obsluze přerušení se také vyšetřuje kontakt jednotlivých chodidel se zemí a náraz na překážky. Jelikož byl problém s naklápěním robota, jsou při pokládání jednotlivých končetin porovnávány hodnoty při doteku se zemí s hodnotami z předchozího kroku. Pokud se tyto hodnoty příliš neliší (jejich rozdíl hodnot je menší než 3), je použita původní hodnota. Tím bylo zamezeno naklápění, ale také se snížilo rozlišení detekovatelných překážek.

Dále byla vytvořena aplikace (uživatelské rozhraní) pro bezdrátové ovládání a programování robota. Tato aplikace pracuje ve dvou módech. Aplikační mód slouží pro ovládání směru pohybu robota. Programovací mód je úzce spjat s bootloaderem. Jelikož je bezdrátová komunikace založena na VF modulu BIM-433-F, který vyžaduje v rámci čtyř bytů vyrovnaný počet nul jako jedniček, je každý byte dat hned doplněn inverzním bytem viz. obrázek 7.2. Tato kombinace je také využita pro kontrolu správnosti přijatých dat. Pokud bootloader při programování zachytí dva po sobě jdoucí byty, které nejsou vzájemně inverzní, vyhodnotí to jako chybu v komunikaci. Po příjmu celé stránky odešle požadavek o její opětovné poslání. Toto opatření však nestačí k dostatečné robustnosti bezdrátové komunikace. Bylo by vhodné přidat ošetření proti ztrátě bytů, například pomocí časovače, který přeteče při nečekaně dlouhém čekání na data. V tu chvíli by bootloader poslal zpět hlášení o chybě a přenos by se opakoval.

# LITERATURA

- [1] SLINTÁK, Vlastimil. Bootloader v mikrokontrolérech AVR. In: *MArt.cz* [online]. 2012 [cit. 2014-05-17]. Dostupné z: <http://uart.cz/721/bootloader-v-avr/>
- [2] VÁGNER, Martin. *Vzdálená správa jednočipových systémů*. Brno, 2010. Diplomová práce. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Ing. Pavel Kučera, Ph.D.
- [3] ATMEL CORPORATION. *8-bit AVR Instruction Set*. San Jose (USA), 2010. Dostupné z: <http://www.atmel.com/Images/doc0856.pdf>
- [4] ATMEL CORPORATION. *8-bit AVR Microcontroller with 16K Bytes In-System Programmable Flash*. San Jose (USA), 2013. Dostupné z: <http://www.atmel.com/devices/ATMEGA162.aspx>
- [5] ŠOLC, František a Luděk ŽALUD. *ROBOTIKA* [online]. Brno, 1. 10. 2002 [cit. 2014-05-17]. Dostupné z: <http://www.uloz.to/xXdgUCT/robotika-solc-zalud-vut-pdf#download>
- [6] KOŠNAR, Karel. *Mobilní robotika* [online]. České Vysoké Učení Technické v Praze, 2007 [cit. 2014-05-17]. Dostupné z: <http://www.roznovskastredni.cz/dwnl/pel2007/06/Kosnar.pdf>
- [7] RADIOMETRIX LTD. *Low Power UHF Data Transceiver Module*. England, 1997. Dostupné z: <http://www.datasheetarchive.com/radiometrix%20%20BIM%20433%20F-datasheet.html>
- [8] ŠKOPEK, Pavel. Techbox: Bluetooth sjednotilo bezdrátovou komunikaci. In: *Mobilnet.cz* [online]. 2013, 24. 05. [cit. 2014-05-17]. Dostupné z: <http://mobilenet.cz/clanky/techbox-bluetooth-sjednotilo-bezdratovou-komunikaci-12085>
- [9] *Sparkfun* [online]. [cit. 2014-05-17]. Dostupné z: <https://www.sparkfun.com>
- [10] Wi-Fi. In: *Wikipedie Otevřená encyklopedie* [online]. 2014, 18. 2. [cit. 2014-05-17]. Dostupné z: <http://cs.wikipedia.org/wiki/Wi-Fi>
- [11] ROVINGNETWORKS. *RN-171-XV 802.11 b/g Wireless LAN Module*. Los Gatos, 2012. Dostupné z: [http://www.tme.eu/cz/details/rn171xvs\\_rm/komunikacni-moduly-rf/microchip-technology/#](http://www.tme.eu/cz/details/rn171xvs_rm/komunikacni-moduly-rf/microchip-technology/#)
- [12] VELEBA, Tomáš. *Čtyřnohý krácející robot*. Brno, 2008. Diplomová práce. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Ing. TOMÁŠ NEUŽIL.

- [13] ZigBee - novinka na poli bezdrátové komunikace. In: *Hw.cz* [online]. 2005 [cit. 2014-05-17]. Dostupné z: <http://www.hw.cz/navrh-obvodu/rozhrani/zigbee-novinka-na-poli-bezdratove-komunikace.html>
- [14] ATMEL CORPORATION. *ZigBit™ 2.4 GHz Wireless Modules*. © 2009. Dostupné z: <http://www.tme.eu/cz/details/atzb-24-a2r/komunikacni-moduly-rf/atmel/#>
- [15] *GM electronic* [online]. [cit. 2014-05-17]. Dostupné z: <http://www.gme.cz/>
- [16] BENEŠ, Pavel, Jan CHLEBNÝ, Jindřich KRÁL, Josef LANGER a Marie MARTINÁSKOVÁ. *Automatizace a automatizační technika 3: Prostředky automatizační techniky*. Brno: CP Books, 2005, viii, 280 s. ISBN 80-251-0795-7.
- [17] BENEŠ, Petr. *Přednáška: Snímače polohy, rychlosti a zrychlení*. Brno, 2013.
- [18] Řízení modelářského servomotoru. In: *Narrow freeweb* [online]. 2012 [cit. 2014-05-17]. Dostupné z: <http://narrow.vzap.eu/elektrotechnika/navody/58-rizeni-modelerskeho-servomotoru>
- [19] *Graupner* [online]. [cit. 2014-05-17]. Dostupné z: <http://www.graupner.de>

## SEZNAM ZKRATEK A SYMBOLŮ

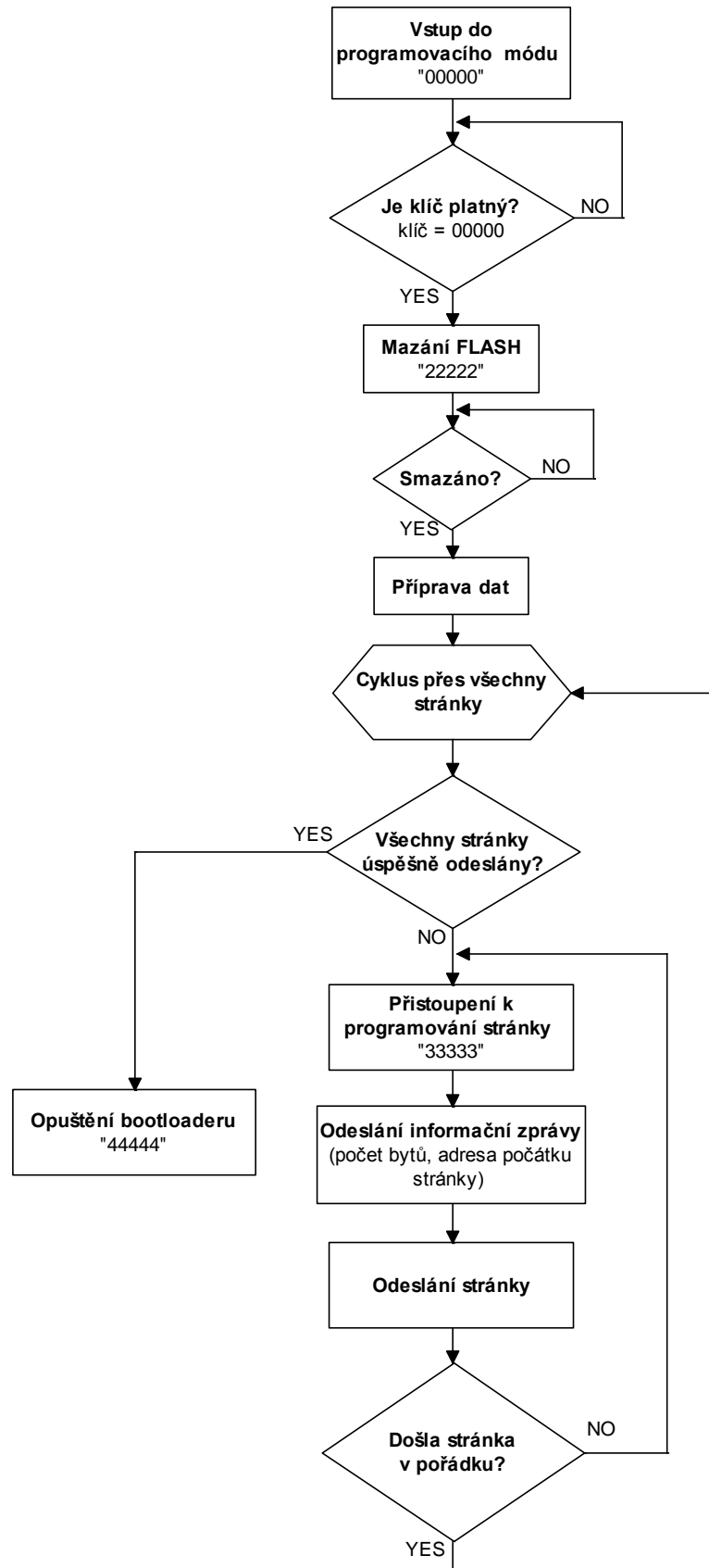
VF	Vysoká frekvence
RWW	Read-While-Write
NRWW	Non-Read-While-Write
SPM	Store Program Memory
UART	Universal Asynchronous Receiver/Transmitter
PWM	Pulse width modulation
ISM	Industrial Scientific and Medicine
IEEE	Institute of Electrical and Electronics Engineers
SSID	Service Set Identifier
AP	Access point
UHF	Ultra High Frequency
MKO	Monostabilní klopný obvod
DPS	Deska plošného spoje
USB	Universal Serial Bus
PC	Personal computer
TCCR0	Timer Counter Control Register 0
TIMSK	Timer Interrupt Mask Register
TOIE0	Timer Overflow Interrupt Enable
UBRR1L	USART Baud Rate Register Low Byte
RTS	Request to Send
DTR	Data Terminal Ready

# SEZNAM PŘÍLOH

Příloha A - Vývojový diagram programovacího módu

Příloha B - CD-ROM

# Příloha A





## **Příloha B**

Příloha obsahuje:

- Elektronickou verzi tohoto dokumentu: BP\_Krejsa.pdf
- Řídící program do mikroprocesoru: Robot.rar
- Uživatelské rozhraní pro ovládání a programování robota: Komunikace.rar
- Bootloader pro možnost bezdrátového programování: Bootloader.rar