

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

ČTYŘNOHÝ KRÁČEJÍCÍ ROBOT

Bc. Petr Černošlávek

Diplomová práce
2015

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Petr Černošlák**
Osobní číslo: **I13460**
Studijní program: **N2646 Informační technologie**
Studijní obor: **Řízení procesů**
Název tématu: **Čtyřnohý kráčejíci robot**
Zadávací katedra: **Katedra řízení procesů**

Z á s a d y p r o v y p r a c o v á n í :

Cílem práce je návrh a realizace čtyřnohého kráčejíciho robota. Ke konstrukci nohou robota a zajištění jejich pohybu lze využít komponenty robotické stavebnice Bioloid Premium. Mechanické díly robota budou navrženy ve zvoleném vizualizačním software a realizovány v součinnosti s externí firmou. Elektronická řídicí jednotka bude určena k řízení autonomního, stabilního pohybu robota. Základem řídicí jednotky bude mikropočítač fy Atmel, řady ATmega (např. ATmega128). Jejím hlavním úkolem bude zajistit stabilní chůzi robota, diagnostiku robota a komunikaci s nadřazeným systémem - osobním počítačem. Komunikace s osobním počítačem bude realizována rozhraním Wi-fi. Nadřazený řídicí systém osobního počítače bude umožňovat dálkové řízení robota a vizualizaci diagnostických dat. V teoretické části bude popis konstrukčního řešení kráčejíci robotů a principy jejich možného řízení. V praktické části bude popis vlastního technického řešení kráčejíciho robota. Firmware mikropočítače a ovládací software bude realizován ve zvoleném programovacím jazyce. Součástí práce bude kompletní dokumentace pro realizaci hardware robota a kompletní zdrojové kódy použitého software.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

MATOUŠEK, D. Práce s mikrokontroléry ATMEL AVR - 3. díl - edice uP a praxe 2. vydání, BEN - technická literatura, 2006, ISBN 80-7300-209-4

ZÁHLAVA, V. Návrh a konstrukce DPS, BEN-technická literatura, 2010, ISBN 978-80-7300-266-4

Stavebnice Bioloid Premium [online] [cit. 2014-10-10] Dostupné na: <http://www.trossenrobotics.com/p/bioloid-premium-robot-kit.aspx>

Vedoucí diplomové práce:

Ing. Libor Havlíček, Ph.D.

Katedra řízení procesů

Datum zadání diplomové práce:

30. října 2014

Termín odevzdání diplomové práce:

15. května 2015



L.S.

prof. Ing. Simeon Karamazov, Dr.
děkan

Ing. Daniel Honc, Ph.D.
vedoucí katedry

V Pardubicích dne 14. listopadu 2014

Prohlášení

Prohlašuji

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích 15. 5. 2015

Podpis

Poděkování

Úvodem bych chtěl poděkovat vedoucímu práce Ing. Liborovi Havlíčkovi, Ph.D., za věcné nápady a poskytnutí prostředků, nutných pro vývoj tohoto zařízení. Nesmím zapomenout na svého kolegu Michala Staška, DIS., který mi poskytl cenné rady při návrhu a osazování plošného spoje. Dále bych chtěl zmínit svou rodinu, které děkuji za psychickou a finanční podporu během celého studia. Rodina jsou lidé, kteří jsou pro nás v životě nejdůležitější, proto také děkuji své přítelkyni Barboře Štěpánkové, která za mnou vždy stála a je pro mě motivací a oporou. Nemáme zapomínat na lidi, kteří už nejsou mezi námi a v mém životě se objevil jeden člověk, kterému hodně vděčím za své úspěchy, za to jaký jsem, byla to moje babička, paní Marie Vlasáková. Říká se, že přátelé z vysokoškolských studií, zůstávají přáteli na celý život, proto děkuji svým kamarádům a spolubydlícím Bc. Milanovi Jičínskému a Bc. Radimovi Kyrzczovi za školní i mimoškolní zážitky, které jsme spolu za všechny roky studií prožili.

V Pardubicích 15. 5. 2015

Podpis

ANOTACE

Tato práce se obecně zabývá řídicími algoritmy chůze čtyřnohých robotů. Předmětem je také implementace na konkrétní reálné konstrukci. Dále je obecně rozebírána problematika servopohonů a komunikačních modulů, používaných pro dálkové ovládání. Praktická část popisuje návrh a výrobu dílů, které tvoří mechanickou konstrukci. Řídicí jednotka je navržena přímo pro toto řešení, včetně softwarové části.

KLÍČOVÁ SLOVA

Čtyřnohý robot, servopohon, Wi-Fi modul.

TITLE

Quadruped Robot

ANNOTATION

This thesis follow up of control algorithms the walking four-legged robots. The subject is also on the specific implementation of the real structure. It is also follow up of actuators and communication modules used for remote control. The practical part describes the design and manufacture of parts that make up the mechanical structure. The control unit is designed specifically for this solution, including software section.

KEYWORDS

Quadruped Robot, actuator, Wi-Fi module.

Obsah

Seznam zkratk	9
Seznam symbolů	11
Seznam obrázků	12
Seznam tabulek	14
ÚVOD	15
1 ŘÍDÍCÍ ALGORITMY CHODÍCÍCH ROBOTŮ	16
1.1 PROBLÉM INVERZNÍ KINEMATIKY	16
1.1.1 Transformace polohy bodu	17
1.2 SEKVENCE NAUČENÝCH POHYBŮ	20
1.2.1 Chůze dopředu a dozadu	20
1.2.2 Rotace vpravo a vlevo	21
2 SERVOPOHON	22
2.1 ELEKTRICKÝ SERVOPOHON	23
2.2 ELEKTROMOTORY POUŽÍVANÉ V SERVOPOHONECH	24
2.2.1 Stejnosměrný motor	25
2.2.2 Elektronicky komutovaný motor	26
2.2.3 Krokový motor	27
2.3 ZPŮSOBY MĚŘENÍ POLOHY	28
2.3.1 Absolutní snímače	28
2.3.2 Relativní snímače	29
3 MOŽNOSTI DÁLKOVÉHO OVLÁDÁNÍ	30
3.1 XBEE MODULY	30
3.2 BLUETOOTH MODULY	31
3.3 WI-FI MODULY	32
4 POPIS POUŽITÝCH HARDWAROVÝCH PROSTŘEDKŮ	33
4.1 POUŽITÝ MIKROPROCESOR	33
4.2 PŘEVODNÍK NAPĚŤOVÝCH ÚROVNÍ	34
4.3 MEMS AKCELEROMETR A GYROSKOP	35
4.4 WI-FI MODUL	36
4.5 SERVOPOHONY	38
4.6 SBĚRNICE UART	41
5 POPIS POUŽITÝCH SOFTWAREVÝCH PROSTŘEDKŮ	42

5.1 KOMUNIKAČNÍ PROTOKOL DYNAMIXEL	42
5.2 BLUEGIGA BGSCRIPT	44
6 KONSTRUKČNÍ ŘEŠENÍ.....	47
6.1 KONSTRUKCE TĚLA.....	47
6.2 KONSTRUKCE NOHY	47
7 HARDWAROVÉ ŘEŠENÍ ROBOTA	48
7.1 ZAPOJENÍ NAPÁJECÍCH OBVODŮ.....	49
7.2 ZAPOJENÍ MIKROPROCESORU	50
7.3 ZAPOJENÍ SBĚRNIC	51
7.4 NÁVRH PLOŠNÉHO SPOJE A OSAZENÝ PLOŠNÝ SPOJ	53
8 SOFTWAREVÉ ŘEŠENÍ ROBOTA.....	54
8.1 WI-FI MODUL	54
8.2 ŘÍDÍCÍ JEDNOTKA ROBOTA	56
8.3 OVLÁDACÍ A VIZUALIZAČNÍ SOFTWARE	59
ZÁVĚR	61
LITERATURA	62
Seznam příloh	64

Seznam zkratek

ACK	Acknowledge
A/D	Analog/Digital
AP	Access Point
CNC	Computer Numerical Control
CTS	Clear To Send
DHCP	Dynamic Host Configuration Protocol
DIL	Dual In-Line
DNS	Domain Name Service
DPAK	Discrete Packaging
DRC	Design Rules Control
EEPROM	Electrically Erasable Programmable Read Only Memory
FET	Field Effect Transistor
I/O	Input/Output
ISO/OSI	International Organization for Standardization/Open Systems Interconnection
FTP	File Transfer Protocol
I ² C	Inter-Integrated Circuit
HAL	Hot Air Levelling
HFP	Hands Free Profile
HID	Human Interface Device
HTTP	HyperText Transfer Protocol
ISP	In System Programming
JTAG	Joint Test Action Group
LED	Light Emitting Diode
MEMS	Micro Electro Mechanical System
MISO	Master In Slave Out
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
MOSI	Master Out Slave In
PCM	Pulse Code Modulation
PDI	Programming Data Input
PDO	Programming Data Output
PWM	Pulse Width Modulation
QFN/MLF	Quad Flat No-Lead/Micro Lead Frame Package

RAM	Random Access Memory
RICS	Reduced Instruction Set Computer
RTS	Request To Send
RXD	Receive Data
SCL	Serial Clock
SDA	Serial Data
SMD	Surface Mount Device
SMTP	Simple Mail Transfer Protocol
SPI	Serial Peripheral Interface
SPP	Serial Port Profile
SSID	Service Set Identifier
TCP/IP	Transmission Control Protocol/Internet Protocol
TQFP	Thin Quad Flat Package
TTL	Transistor-Transistor Logic
TWI	Two-Wire Serial Interface
TXD	Transmit Data
UART	Universal Asynchronous Receiver Transmitter
UFL	Ultra Small Surface Mount Coaxial Connector
USART	Universal Synchronous Asynchronous Receiver Transmitter
XML	eXtensible Markup Language

Seznam symbolů

C	konstanta motoru
e	regulační odchylka
I	elektrický proud
M	moment síly
p	pozice bodu v souřadném systému
R	matice rotace
r	vektor posuvu počátku souřadného systému
T	matice homogenní transformace
t	čas
U	elektrické napětí
u	akční zásah
w	žádaná hodnota
y	regulovaná veličina
Φ	magnetický tok
ω	úhlová rychlost

Seznam obrázků

Obrázek 1.1 – Pohyb referenčního bodu bez změny polohy nohou	17
Obrázek 1.2 – Kinematický model nohy robota	19
Obrázek 1.3 – Časový diagram chůze čtyřnohého robota	21
Obrázek 1.4 – Popis průběhu rotačního pohybu robota vpravo	22
Obrázek 2.1 – Obecné blokové schéma otevřeného regulačního obvodu	23
Obrázek 2.2 – Obecné blokové schéma uzavřeného regulačního obvodu	23
Obrázek 2.3 – Elektrický servopohon jako systém.....	24
Obrázek 2.4 – Řez stejnosměrným motorem.....	26
Obrázek 2.5 – Řez elektricky komutovaným motorem	27
Obrázek 2.6 – Řez hybridním krokovým motorem	28
Obrázek 2.7 – Absolutní snímač polohy s 8-bitovým výstupem	29
Obrázek 2.8 – Konstrukce relativního snímače polohy	30
Obrázek 3.1 – Komunikační modul XBee 802.15.4 PRO	31
Obrázek 3.2 – Komunikační modul Bluegiga BLE121RL.....	32
Obrázek 3.3 – Komunikační modul Bluegiga WF121	33
Obrázek 4.1 – Mikroprocesor ATmega128 v pouzdře TQFP	34
Obrázek 4.2 – Popis vývodů integrovaného obvodu	35
Obrázek 4.3 – Modul se senzorem LSM330DLC	36
Obrázek 4.4 – Blokové schéma Wi-Fi modulu.....	37
Obrázek 4.5 – Porovnání servopohonů Dynamixel	38
Obrázek 4.6 – Servopohon Dynamixel AX-12A.....	39
Obrázek 4.7 – Konstrukční řešení snímače polohy v servopohonu Dynamixel AX-12A	39
Obrázek 4.8 – Řídící jednotka a stejnosměrný motor servopohonu Dynamixel AX-12A	40
Obrázek 4.9 – Převodová soustava servopohonu Dynamixel AX-12A.....	41
Obrázek 4.10 – Zapojení konektorů servopohonu Dynamixel AX-12A	41
Obrázek 4.11 – Diagram přenosu rámce po UART sběrnici	42
Obrázek 5.1 – Bluegiga WF121 GUI	46
Obrázek 6.1 – Duralové díly tvořící tělo robota	47
Obrázek 6.2 – Konstrukce nohy robota	48
Obrázek 7.1 – Blokové schéma řídicí jednotky robota.....	48
Obrázek 7.2 – Schéma zapojení napájecích obvodů.....	49
Obrázek 7.3 – Schéma zapojení mikroprocesoru	50

Obrázek 7.4 – Schéma zapojení konektorů servopohonů.....	52
Obrázek 7.5 – Schéma zapojení převodníku úrovní SN74LVC1T45	52
Obrázek 7.6 – Návrh plošného spoje řídicí jednotky robota	53
Obrázek 7.7 – Osazený plošný spoj řídicí jednotky robota	54
Obrázek 7.8 – Návrh plošného spoje komunikační jednotky robota	54
Obrázek 7.9 – Osazený plošný spoj komunikační jednotky robota.....	54
Obrázek 8.1 – Vývojový diagram firmwaru Wi-Fi modulu	55
Obrázek 8.2 – Stavový diagram chůze dopředu a dozadu	57
Obrázek 8.3 – Vývojový diagram hlavní smyčky programu	58
Obrázek 8.4 – Umístění servopohonů dle adres	58
Obrázek 8.5 – Ovládací a vizualizační software.....	60

Seznam tabulek

Tabulka 4.1 – Pravdivostní tabulka integrovaného obvodu SN74LVC1T45	35
Tabulka 5.1 – Struktura instrukčního rámce komunikačního protokolu Dynamixel	42
Tabulka 5.2 – Tabulka nejpoužívanější instrukcí komunikačního protokolu Dynamixel	43
Tabulka 5.3 – Datový rámec instrukce SYNC_WRITE	44
Tabulka 5.4 – Rámec BGAPI pro vytvoření Access Pointu	46
Tabulka 5.5 – Událost BGAPI signalizující úspěšné vytvoření Access Point	46
Tabulka 8.1 – Tabulka hodnot povelů ovládajících pohyb robota	60

ÚVOD

Tato diplomová práce se zabývá problematikou chodících mobilních robotů od teoretických poznatků po praktické provedení konkrétní konstrukce a implementaci řídicího algoritmu. Slovo robot je v dnešním moderním světě pokládáno za samozřejmost, ale s ohledem na historii tomu tak vždy nebylo a nesmíme zapomínat na původ tohoto výrazu, který poprvé použil náš slavný spisovatel Karel Čapek. Již ve starověku se lidé snažili vyvinout stroj, který by dokázal vykonávat lidskou práci, což je jedno z možných vysvětlení tohoto slova. Dnes můžeme na roboty narazit snad ve všech odvětvích lidské činnosti od robotických manipulátorů přes automatizované výrobní podniky po aplikace v lékařství, bez kterých by spousta dnes už rutinních operačních zákroků nebylo možné vykonávat. Chodící robot je specifický v tom, že je napodobením přírody u čtyřnohých zvířat a jeho použití je zejména v náročném terénu, ve kterém se nachází překážky, jejichž zdolání je pro roboty jiných typů náročné či nemožné.

Obsah práce je dělen na teoretickou a praktickou část. Teoretická část rozebírá obecnou problematiku inverzní kinematiky, časovou synchronizaci pohybu nohou, servopohonů a komunikačních modulů, umožňujících dálkové ovládání. Dle tématického členění můžeme teoretickou část dělit na plně obecnou, jejíž obsah byl výše popsán a část přibližující se ke konkrétnímu řešení, která popisuje funkci a provedení použitých hardwarových a softwarových prostředků. Praktická část popisuje reálná schémata zapojení elektronických obvodů a implementaci řídicího algoritmu chůze do určených směrů. Součástí je také nasazení konkrétního komunikačního modulu s rozhraním Wi-Fi, který tvoří datovou cestu mezi ovládacím softwarem a samotným robotem. Vlastní obsah práce je zakončen popisem programového řešení ovládacího a vizualizačního softwaru.

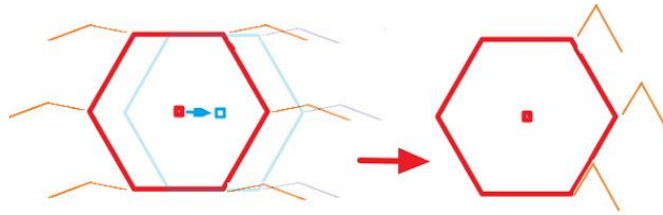
1 ŘÍDÍCÍ ALGORITMY CHODÍCÍCH ROBOTŮ

Při tvorbě řídicího algoritmu chodícího robota hledáme nejčastěji inspiraci v přírodě. Pokud se jedná o humanoidní roboty, je problém řízení neobtížnější, protože je značný problém s udržováním stability těla. Na druhou stranu je nám tento typ robotů nejbližší, protože je inspirován anatomii lidského těla. Tato práce se zabývá čtyřnohým robotem, kde je potřebné řídit a synchronizovat pohyb dvojnásobného počtu končetin, než u robotů humanoidních, ale podstatně se zde zjednoduší problém udržení stability těla. Řídicí algoritmus je ve většině případů řešen dvěma způsoby. První variantou je využití inverzní kinematiky, která slouží k výpočtu poloh jednotlivých servopohonů na základě znalosti požadovaných souřadnic nášlapných ploch nohou. Z hlediska inverzní kinematiky se jedná o koncový bod kinematického systému, který je tvořen jednotlivými nohama. Druhou variantou je implementace sekvencí naučených pohybů. Tato metoda lze použít pouze pro realizaci základního algoritmu chůze bez uvažování překážek. Využíváme zde faktu, že pohyb vychází vždy ze zvolené výchozí polohy nohou vzhledem k tělu. Značným zjednodušením je, že pohyb všech nohou je stejný pouze fázově posunutý a tyto fáze pohybu se opakují po celou dobu trvání pohybu příslušným směrem. Pro oba přístupy je nutná implementace časování, tedy časové osy, která určuje kdy která noha vykonává příslušný pohyb. Časový diagram pohybu bude později podrobně popsán na konkrétní realizaci algoritmu.

1.1 PROBLÉM INVERZNÍ KINEMATIKY

Při aplikaci tohoto problému na řídicí algoritmus chodícího robota bereme střed těla jako referenční bod a současně počátek kartézského souřadného systému. Při pohybu dochází ke změně polohy tohoto referenčního bodu a tato změna je převedena na změnu polohy nohou. Inverzní kinematika tedy slouží ke zjištění úhlů natočení jednotlivých servopohonů, které určují počet stupňů volnosti nohou. Tento princip je použit při řízení pohybu letadel, kde uvažujeme tři směry. Rotační pohyb trupu v horizontální rovině nazývaný Roll, náklon trupu ve vertikální rovině nazývaný Pitch a Yaw, což je rotace kolem referenčního bodu, kterým je střed trupu letadla. Pro správnou funkci algoritmu je třeba brát v úvahu změnu polohy referenčního bodu i v případě, že nedochází ke změnám absolutních poloh nohou. Při aplikaci problému inverzní kinematiky musíme tyto změny uvažovat, protože zjišťujeme aktuální absolutní polohy nohou, vzhledem k referenčnímu bodu, jehož poloha se změnila pouze vlivem změn úhlů natočení servopohonů, ne vlivem

změn absolutní polohy nášlapných ploch nohou. Tato situace je popsána na následujícím obrázku [LIANG, 2012].



Obrázek 1.1 – Pohyb referenčního bodu bez změny polohy nohou [LIANG, 2012]

1.1.1 Transformace polohy bodu

Výše bylo uvedeno, že referenčním bodem, ke kterému je vztahován výpočet úhlů natočení servopohonů pomocí inverzní kinematiky, je střed těla robota. Tento referenční bod se pohybuje, takže dochází k transformaci jeho polohy, která je většinou popsána tzv. homogenní transformací. Při popisu této transformace uvažujeme tuhé těleso umístěné v základním souřadném systému, což je v mém případě tělo robota a nějaký bod na tomto tělese, kterým je v této situaci střed. Při pohybu dochází k přemístění tohoto tělesa a předpokládáme, že dochází i ke změně počátku souřadného systému, který je pevně spojen s tělesem. Potom vznikají dva souřadné systémy, jeden z nich je základní, nezávislý na změně polohy a druhý pevně svázaný s tělesem, kopírující jeho trajektorii.

Zavedeme označení:

p_0 – pozice bodu p v souřadném systému tělesa, která se nemění

p_1 – pozice bodu p v základním souřadném systému, která se mění

r_{01} – vektor posunu počátku souřadného systému

Potom je transformace polohy bodu popsána vztahem, kde R_{01} je maticí rotace:

$$p_0 = r_{01} + R_{01}p_1 \quad (1.1)$$

Homogenní transformaci je možné zapsat pomocí rozšířeného vektoru polohy:

$$\bar{p} = \begin{bmatrix} p \\ 1 \end{bmatrix} \quad (1.2)$$

Tím získáme zjednodušený zápis, kde \mathbf{T}_{01} je maticí homogenní transformace:

$$\begin{bmatrix} \bar{p}_0 \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{01} & \mathbf{r}_{01} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_1 \\ 1 \end{bmatrix} = \mathbf{T}_{01} \bar{p}_1 \quad (1.3)$$

Pokud je ve zjednodušeném případě uvažována rovina, můžeme rotaci vektoru o úhel α popsat následujícím vztahem:

$$\begin{aligned} x' &= x \cos \alpha - y \sin \alpha \\ y' &= x \sin \alpha + y \cos \alpha \end{aligned} \quad (1.4)$$

Začneme-li uvažovat trojrozměrný prostor, můžeme stejnou rotaci popsat maticí rotace \mathbf{R}_z , která odpovídá dílčí rotaci kolem osy z. Uvažujeme ovšem všechny tři rozměry, takže musíme definovat matice rotací pro všechny tři osy:

$$\mathbf{R}_z(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.5)$$

$$\mathbf{R}_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \quad (1.6)$$

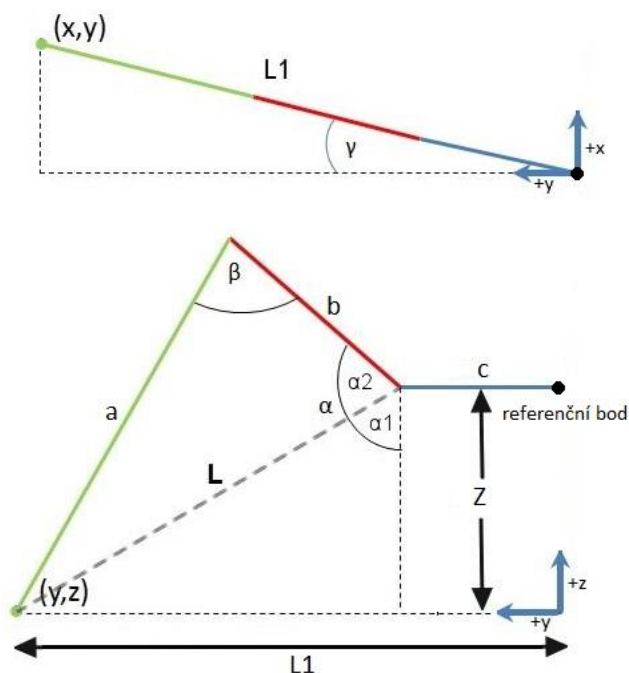
$$\mathbf{R}_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{bmatrix} \quad (1.7)$$

Pro realizaci algoritmu inverzní kinematiky je potřebná celková matice rotace, popisující změnu polohy ve všech třech osách. Existuje více variant pro její výpočet, ale nejčastěji jsou použity tzv. Eulerovy ZYX úhly, odpovídající výše popisovanému řízení pohybu letadel (Roll, Pitch, Yaw).

Vztah pro získání celkové matice rotace je potom:

$$\mathbf{R}_{zyx}(\alpha, \beta, \gamma) = \mathbf{R}_z(\alpha) \mathbf{R}_y(\beta) \mathbf{R}_x(\gamma) \quad (1.8)$$

Výstupem algoritmu inverzní kinematiky jsou tedy úhly natočení jednotlivých servopohonů závislé na zadané poloze koncového bodu. Každá noha robota je nezávislým kinematickým systémem, který má tři stupně volnosti dané počtem kloubů v systému, v tomto případě počtem servopohonů. Na dalším obrázku je popsán kinematický systém jedné nohy [CVEJN, 2014].



Obrázek 1.2 – Kinematický model nohy robota

Příklad jednoduchého výpočtu úhlů natočení, ze znalostí souřadnic koncového bodu s použitím inverzní kinematiky:

$$\frac{x}{y} = \tan \gamma \Rightarrow \gamma = \arctan\left(\frac{x}{y}\right) \quad (1.9)$$

$$L = \sqrt{z^2 + (L_1 - c)^2}$$

Úhel α je pro zjednodušení výpočtu rozdělen na dvě části:

$$\alpha_1 = \arccos\left(\frac{z}{L}\right)$$

$$a^2 = b^2 + L^2 - 2bL \cos \alpha_2 \Rightarrow \alpha_2 = \arccos\left(\frac{a^2 - b^2 - L^2}{-2bL}\right)$$

$$\alpha = \alpha_1 + \alpha_2 = \arccos\left(\frac{z}{L}\right) + \arccos\left(\frac{a^2 - b^2 - L^2}{-2bL}\right) \quad (1.10)$$

$$L^2 = a^2 + b^2 - 2ab \cos \beta \Rightarrow \beta = \arccos\left(\frac{L^2 - a^2 - b^2}{-2ab}\right) \quad (1.11)$$

Získáváme úhly natočení servopohonů α , β , γ , odpovídající koncovému bodu s polohou o souřadnicích x , y , z .

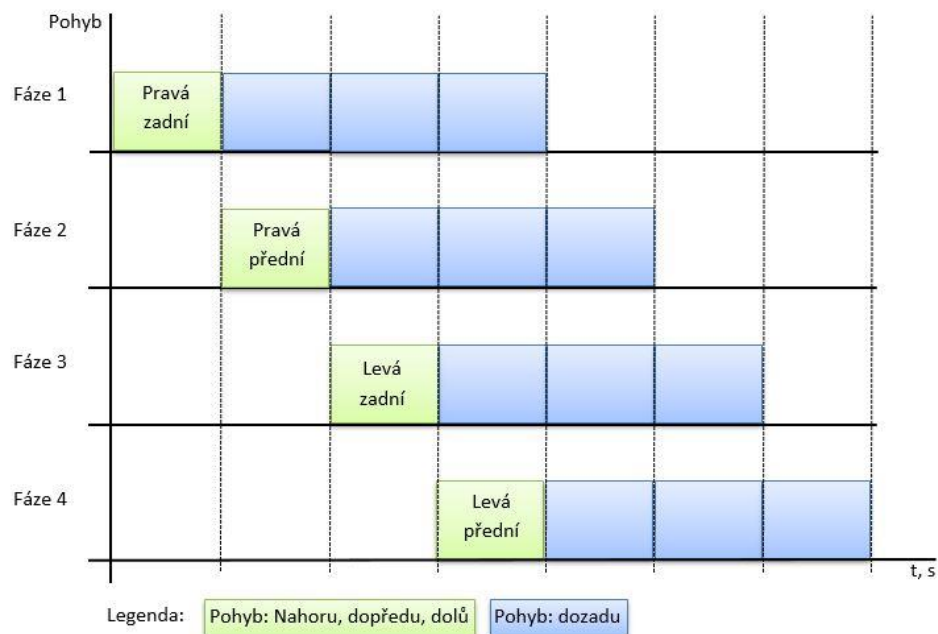
1.2 SEKVENCE NAUČENÝCH POHYBŮ

Na rozdíl od algoritmů řešících problém inverzní kinematiky je tento přístup přímočařejší a jednodušší na pochopení i implementaci. Musíme však počítat s omezeními, která s sebou nese toto řešení. Základním předpokladem řízení chůze je rovná podložka bez překážek, po které se robot pohybuje. Toto omezení vyplývá z toho faktu, že výchylky servopohonů jsou ve všech fázích chůze stejné, nehledě na okolní terén. Tento algoritmus je v případě, že nemáme o okolním prostředí žádné informace, zcela dostačujícím řešením. Funkce je založena na implementaci sekvence pohybů jednotlivých nohou a načasování jejich vzájemného pohybu, probíhajícího v pevně daných fázích, které se opakují. Tímto způsobem je možné implementovat chůzi ve čtyřech základních směrech. Směry dopředu a dozadu jsou prakticky totožné, sekvence pohybů začíná pouze od jiné nohy, podobně je to i s rotací vpravo a vlevo, kde se mění pouze smysl výchylek servopohonů, které zajišťují pohyb nohou do strany. Po implementaci těchto čtyř sekvencí je možný libovolný pohyb robota po rovné podložce. Realizace problému inverzní kinematiky má tedy význam pouze v případě, že máme možnost získat informace o proporcích okolního terénu. Tyto informace se získávají pomocí tzv. exteroceptivních senzorů. Jedná se zejména o tlaková čidla v nášlapných ploškách, která dávají řídicí jednotce informaci o tom, jestli se pohybující noha dotýká pevné podložky. Ve spojení těchto informací a algoritmu inverzní kinematiky je možné řešit chůzi i za podmínky pohybu po nerovném terénu.

1.2.1 Chůze dopředu a dozadu

Tato sekvence pohybů má své kořeny v přírodě u čtyřnohých zvířat, kterými se můžeme při realizaci inspirovat. Při stabilní chůzi musí být dodržen jeden základní předpoklad, kterým je tzv. třínohá stabilita. To znamená, že v každém okamžiku se musí tři nohy dotýkat pevné podložky a pouze jedna noha může být ve vzduchu, čímž je zaručena stabilita. Chůze má čtyři fáze a začíná vždy od zadní nohy. Jako počáteční byla zvolena pravá zadní noha, která vykoná v první fázi pohyb nahoru, dopředu a dolů. Po

dokončení popsaného pohybu, dochází ve stejném okamžiku k vykonání totožného pohybu pravou přední nohou a pravá zadní noha se začíná vracet do původní polohy, vykoná tedy pohyb dozadu. Pohyb dozadu je vykonáván takovou rychlostí, aby bylo dodrženo, že čas jeho vykonávání bude stejně dlouhý jako tři fáze pohybu. Tato časová posloupnost musí být dodržena u všech čtyř nohou v pořadí pravá zadní, pravá přední, levá zadní a levá přední. Pohyb všech čtyř nohou je tedy naprosto totožný, mění se pouze fáze, ve které je vykonáván. Tato sekvence se opakuje po celou dobu trvání pohybu dopředu nebo dozadu. Pohyb dozadu se liší pouze v tom, že invertujeme fáze pohybu, počáteční nohou je tedy levá přední, následuje levá zadní, pravá přední a pravá zadní. Průběh všech fází pohybu je popsán v následujícím časovém diagramu [LIANG, 2013].

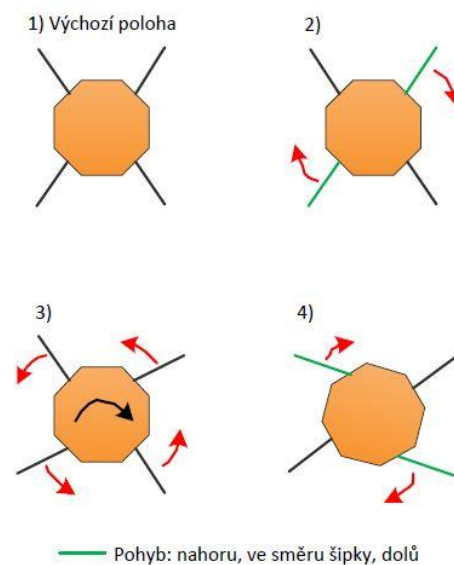


Obrázek 1.3 – Časový diagram chůze čtyřnohého robota

1.2.2 Rotace vpravo a vlevo

K rotačnímu pohybu robota je potřeba zabezpečit rotaci celého těla. Musí být dodržen předpoklad, že je na začátku této sekvence robot v klidu, ve výchozí poloze. Zde je vykonáván pohyb dvou, křížem proti sobě ležících nohou. Pro popis algoritmu je vybrána rotace vpravo s počátečním pohybem pravou zadní a levou přední nohou. Sekvence začíná současným pohybem těchto nohou nahoru, pravá zadní dopředu a levá přední dozadu, poté následuje současný pohyb dolů. K zajištění rotace celého těla je potřebný současný pohyb všech čtyř nohou, pravá zadní a levá přední se tedy vrací do výchozí polohy, dvě zbývající nohy kopírují předcházející pohyb stranou stejnohých

nohou, pravá přední dopředu a levá zadní dozadu. Těmito pohyby byla způsobena rotace celého těla o úhel rovný výchylce servopohonů při pohybu dopředu nebo dozadu. Dokončením této sekvence je vrácení pravé přední a levé zadní nohy do výchozí polohy, což je zabezpečeno pohybem nahoru, stranou do výchozí polohy a dolů, čímž se nohy robota dostanou do stejné polohy vůči tělu jako před začátkem rotace. Tato sekvence je opakována, dokud robot nedosáhne rotace o požadovaný úhel. Při změně smyslu otáčení se invertují pouze pohyby dopředu a dozadu, ostatní jsou totožné. Rotace těla robota je popsána na následujícím obrázku.

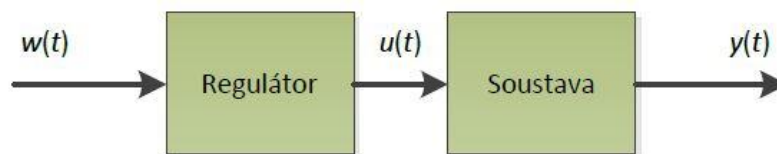


Obrázek 1.4 – Popis průběhu rotačního pohybu robota vpravo

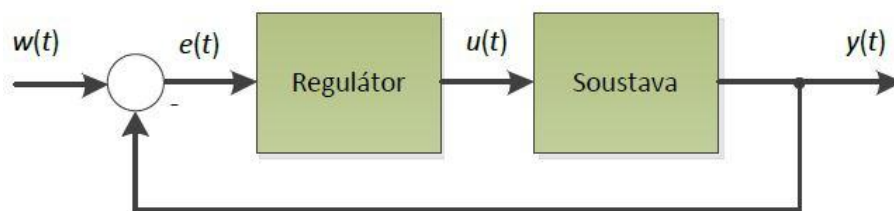
2 SERVOPOHON

Servopohon je obecně zařízení, které mění tlakovou nebo elektrickou energii na mechanickou, pracující ve zpětnovazebné regulační smyčce. Pokud dochází k přeměně tlakové energie, jedná se o hydraulické systémy, kde je přenosovým médiem olej, nebo systémy pneumatické, kde se tlaková síla přenáší pomocí vzduchu. Tyto dvě skupiny mají své opodstatnění hlavně ve výbušném prostředí, protože při jejich činnosti nedochází k jiskření. Hydraulické systémy jsou význačné hlavně kvůli přestavné síle, kterou dokážou vyvinout, proto jsou často využívány u bagrů a důlní techniky. Nevýhodou jsou náklady na údržbu a nutnost zásobníku na hydraulický olej, který je nutno doplňovat a po nějaké době měnit, protože s časem mění své vlastnosti. Touto nevýhodou pneumatické systémy netrpí, neboť zdrojem natlakovaného vzduchu je kompresor. Klasické pohony mohou být ovládány prostřednictvím otevřeného

regulačního obvodu tzv. ovládání, kde regulátor nepracuje s výstupní veličinou, tedy rychlostí či polohou rotoru nebo posuvné části u lineárních pohonů. Servopohon se vždy vyznačuje použitím zpětné vazby, tedy uzavřeného regulačního obvodu. Zpětná vazba je rychlostní a ve většině případů i polohová. Dominantními vlastnostmi, které nás u servopohonů zajímají, jsou regulační parametry, což je zejména přesnost a rychlost regulace a v případě průmyslového použití spolehlivost. Úkolem rychlostní zpětné vazby je udržování úhlové rychlosti rotoru na žádané hodnotě a odregulování nežádoucích poruch. Polohová zpětná vazba má totožnou úlohu pouze pro jinou veličinu.



Obrázek 2.1 – Obecné blokové schéma otevřeného regulačního obvodu

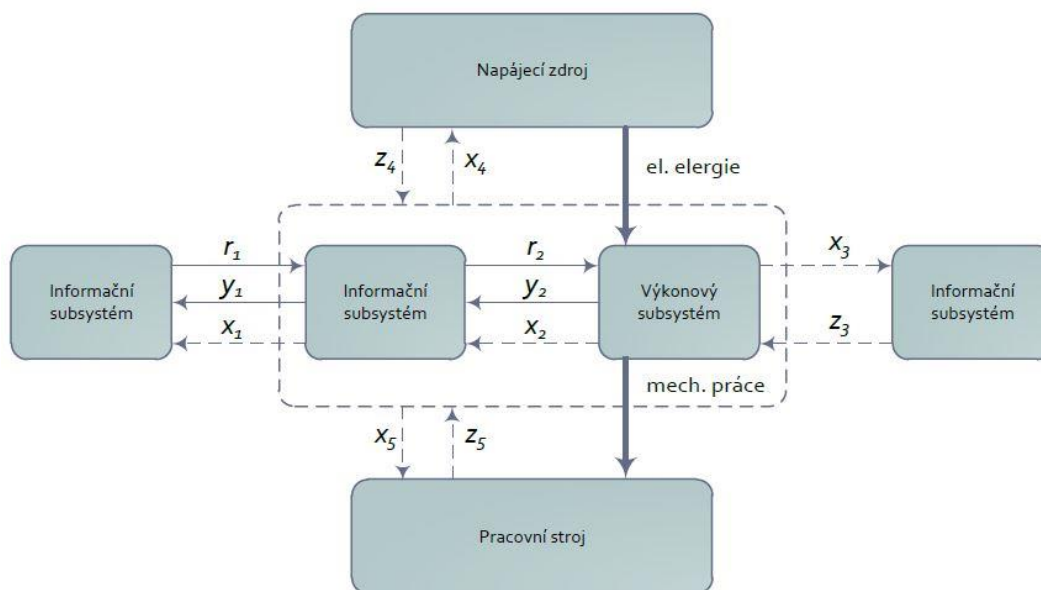


Obrázek 2.2 – Obecné blokové schéma uzavřeného regulačního obvodu

$w(t)$ – žádaná hodnota, $e(t)$ – regulační odchylka, $u(t)$ – akční zásah, $y(t)$ – regulovaná veličina

2.1 ELEKTRICKÝ SERVOPOHON

Zdrojem mechanické energie těchto servopohonů je elektrický motor. Typy nejčastěji používaných elektromotorů budou popsány v této kapitole. Nejprve je však servopohon jako celek rozdělen na dva subsystemy, výkonový a informační. Na servopohon můžeme nahlížet jako na část mechatronického systému, který podléhá ať už pozitivním či negativním vlivům okolního prostředí. Výkonový subsystem tvoří samotný elektromotor a jeho napájecí výkonové měniče. Informační subsystem se skládá z řídicí elektroniky a regulačních obvodů. Okolí servopohonu zahrnuje elektrický napájecí zdroj, poháněné pracovní zařízení, pracovní prostředí a řídicí subsystem celého mechatronického systému. Oba subsystemy provádějí s okolím interakci, popsanou na následujícím obrázku, způsobenou vnitřními a vnějšími vazbami, které mohou být užitečné, ale i rušivé.



Obrázek 2.3 – Elektrický servopohon jako systém

V blokovém schématu vidíme vzájemné vazby mezi servopohonem a ostatními subsystemy tvořícími mechatronický systém. Mezi užitečné vazby můžeme zařadit řídicí signály a žádané hodnoty regulované veličiny označené jako r_1 , r_2 , kterými mohou být např. okamžitá rychlost či poloha. Zpětnovazební signály y_1 , y_2 jsou výstupy ze senzorů různých elektrických i neelektrických veličin, jako jsou proud, napětí, otáčky, poloha apod. Rušivé vazby lze rozdělit podle toho, zda jsou způsobeny samotným servopohonem na vnitřní nebo vnější, způsobené okolím systému. Vnitřním rušením x_1 , x_2 mohou být vysokofrekvenční napětí a magnetická pole, nebo vlivy x_3 , x_5 ovlivňující pracovní prostředí jako chvění, hluk, tepelné ztráty. Odběr energie může způsobit nežádoucí vlivy na napájecí zdroj, jako je například zkreslení vyššími harmonickými x_4 . Vnějšími vazbami majícími negativní účinky na servopohon může být teplota a vlhkost okolí, voda, agresivní páry z_3 . Musíme sem také započítat negativní vlivy napájecího zdroje, jako je pokles napětí akumulátoru, přepětí v síti nebo rušení vyššími harmonickými z_4 . Nesmíme zapomenout na změny zátěžového momentu způsobované mechanickým subsystémem a jeho rezonanční kmitočty z_5 [SKALICKÝ, 2001].

2.2 ELEKTROMOTORY POUŽÍVANÉ V SERVOPOHONECH

Tato práce se zabývá mobilní robotikou, proto budou dále popsány pouze elektromotory, které jsou použity v servopohonech určených pro tyto aplikace.

2.2.1 Stejnoseměrný motor

Jedná se o jeden z nejpoužívanějších elektromotorů v servopohonech, zejména z důvodu jednoduchosti řízení otáček a volby směru otáčení. V nejjednodušším případě můžeme využít přímé úměrnosti mezi přiváděným napětím a otáčkami rotoru. Tato možnost je ovšem energeticky neúsporná, proto je ve většině případů k regulaci otáček použit PWM signál, kde platí nepřímá úměrnost mezi střídou a otáčkami. Volba směru otáčení je možná změnou polarit napájecího napětí, k čemuž jsou ve většině případů použity tranzistorové H-můstky, které je možné sestavit z diskrétních součástek nebo využít specializovaných integrovaných obvodů. Nejčastější konstrukci je stejnosměrný motor s permanentními magnety, které tvoří stator. Magnety jsou vyrobeny z magneticky tvrdých materiálů, většinou na bázi feritu a mají za úkol vytvářet magnetické pole ve vzduchové mezeře mezi státorem a vinutím rotoru. Rotor je tvořen jednotlivými cívkami, uloženými v drážkách a připojenými na komutátor, zabezpečující přepólování, které je nezbytné pro generování otáček. Napájecí napětí je na komutátor přivedeno prostřednictvím kartáčů, tvořících tzv. sběrací ústrojí, které je zdrojem jiskření, jehož nežádoucí vlivy jsou částečně eliminovány odrušovacími kondenzátory. Pro dosažení většího výkonu existuje i varianta stejnosměrného motoru s cizím buzením, kde je magnetické pole statoru vytvářené pomocí vinutí. Nevýhodou jsou větší ztráty v budícím vinutí. Pro stejnosměrný motor platí následující rovnice.

Při otáčení úhlovou rychlostí ω se ve vinutí kotvy indukuje napětí U_i :

$$U_i = C\Phi\omega \quad (2.1)$$

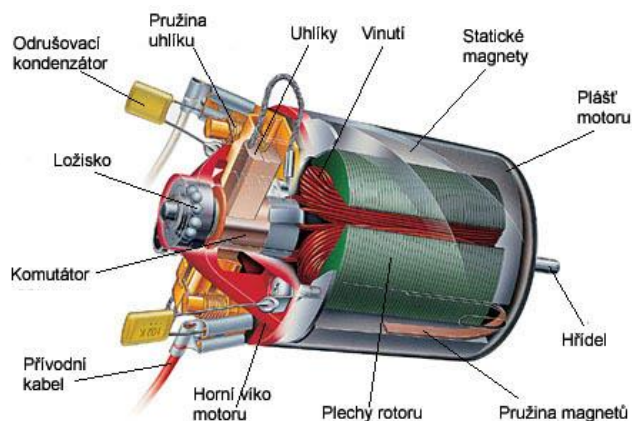
Působením proudu a magnetického toku je vytvářen moment:

$$M = C\Phi I \quad (2.2)$$

kde C – konstanta motoru

Φ – magnetický tok

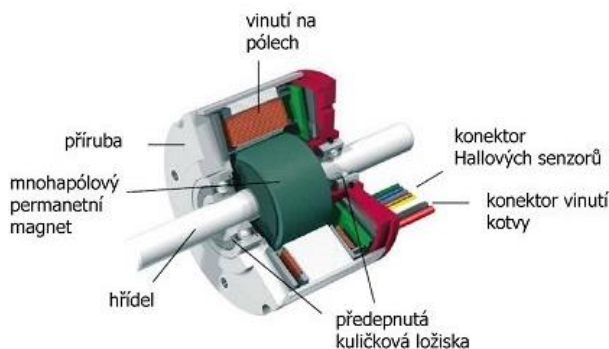
I – proud rotoru



Obrázek 2.4 – Řez stejnosměrným motorem [Offroad RC, 2015]

2.2.2 Elektronicky komutovaný motor

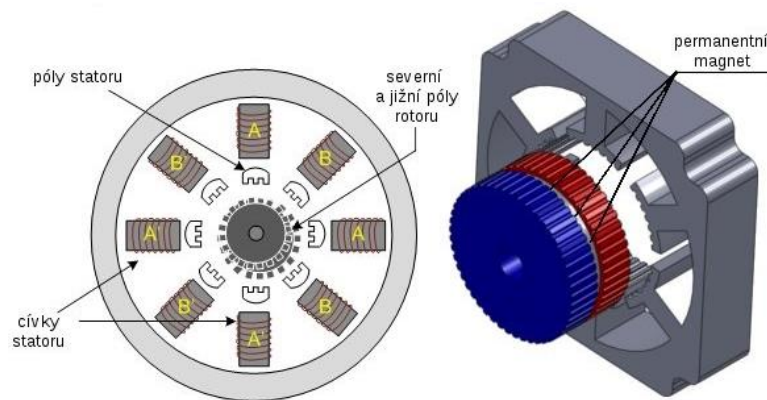
U klasických stejnosměrných motorů s permanentními magnety je problémovou součástí mechanický komutátor. Postupem času dochází k jeho opotřebovávání a ulpívání nečistot, které vyžadují pravidelnou údržbu čištěním nebo zabrušováním kartáčů, či celková výměna sběracího ústrojí. Elektronicky komutovaný motor těmito problémy netrpí, neboť je mechanický komutátor nahrazen snímači polohy a řídicí elektronikou, která přepíná proud do jednotlivých statorových vinutí. Mechanická konstrukce je opačná než u předchozích typů motorů. Rotor je tedy tvořen permanentními magnety, uloženými buď na povrchu, nebo uvnitř a stator má v drážkách uložené třífázové vinutí, podobné jako u standardních asynchronních nebo synchronních strojů. Pro komutaci budícího proudu třífázového vinutí tranzistorovým měničem je potřeba měření polohy rotoru. Informace o poloze je měřena diskrétně po 120° za pomoci fotoelektrických snímačů nebo Hallových sond. Při použití v servopohonech je potřeba zpětnovazební informace, kterou jsou v tomto případě otáčky. Pro dodržení koncepce bezkontaktního provedení je jako snímač použito elektronicky komutované tachodynamo, a tak je zajištěn provoz bez nutnosti údržby systému jako celku. Z pohledu regulace otáček se elektronicky komutovaný motor chová stejně jako klasický stejnosměrný motor, otáčky jsou tedy přímo úměrné velikosti napájecího napětí a je zachována i přímá úměra mezi momentem a proudem. Tvar budícího proudu statorového vinutí je obdélníkový.



Obrázek 2.5 – Řez elektricky komutovaným motorem [TechPark, 2015]

2.2.3 Krokový motor

Rotor krokového motoru může dosahovat pouze diskrétního počtu poloh, který je dán jeho mechanickou konstrukcí. Rychlost otáčení je úměrná frekvenci číslicového signálu, kterým motor řídíme. Přivedením jednoho impulsu řídicího signálu dojde k vykonání pohybu rotoru o jeden krok. Směr otáčení je závislý na polaritě budícího napětí, která je řízená stejně jako u stejnosměrných motorů pomocí H-můstku. Pohyb rotoru je zprostředkován postupným napájením pólových dvojic stejnosměrným proudem. Servopohony s krokovými motory jsou jedinou variantou, u které není třeba snímání polohy, tedy zpětné vazby, pokud jsme schopni zabezpečit vykonání pohybu rotoru o příslušný počet kroků. Krokové motory můžeme dle konstrukčního provedení rozdělit do tří skupin. První z nich jsou reluktanční krokové motory, u kterých jsou vytvořeny póly pomocí zubů na statoru a rotoru, jejichž počet musí být rozdílný. Stator a rotor jsou vyrobeny z magnetických materiálů s výrazně rozdílnou reluktancí v příčné a podélné ose. Druhou skupinou jsou krokové motory s permanentními magnety, které tvoří póly rotoru. Vykazují se výrazným reluktančním momentem, který zabezpečuje klidovou polohu rotoru v nenapájeném stavu statorového vinutí. Poslední skupinou je kombinace popsaných konstrukčních řešení krokových motorů, tzv. hybridní.



Obrázek 2.6 – Řez hybridním krokovým motorem [Servo Drive, 2015]

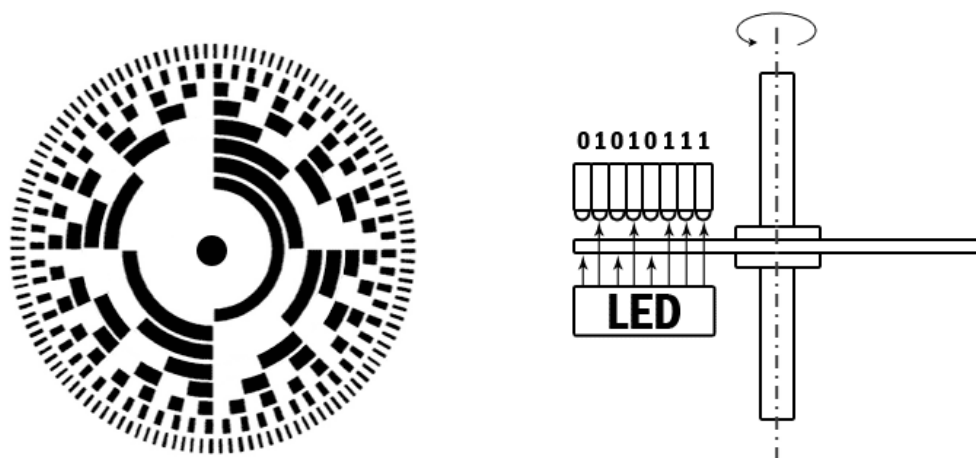
2.3 ZPŮSOBY MĚŘENÍ POLOHY

Zpětnovazební informace o poloze rotoru servopohonu mohou být buď absolutní, nebo relativní. Absolutní poloha znamená, že pro každou pozici rotoru existuje jedinečný výstup ze senzoru, proto je možné získat informaci o poloze bezprostředně po startu systému. Relativní poloha se měří vzhledem k nějakému referenčnímu bodu, takže existuje výchozí poloha, na kterou systém po startu najede a dále je informace o poloze zjišťována od této vztažné hodnoty.

2.3.1 Absolutní snímače

Absolutní snímače polohy jsou většinou na optickém principu. Konstrukční řešení se skládá z optického vysílače např. infračervené LED diody a optického přijímače, kterým je ve většině případů fototranzistor. Mezi vysílačem a přijímačem je umístěn kódovací kotouč, který je rozdělený na několik řad, podle toho jakou bitovou délku má výstupní hodnota polohy ze snímače. Každá řada se skládá z průhledných a neprůhledných míst a má vlastní optický snímač. Políčka na kódovacím kotouči většinou zobrazují Grayův kód, který má výhodu změny sousedních hodnot pouze o jeden bit, čímž zajistíme, že v každém okamžiku bude při současném čtení všech optických senzorů výstupní hodnota polohy ze snímače aktuální. Výhodou je, že hodnota polohy s sebou nese i informaci o smyslu otáčení. Na následujícím obrázku je možná varianta kódovacího kotouče se znázorněným výstupem. Jednou z možností absolutního snímání polohy u méně sofistikovaných servopohonů je kruhová odporová dráha. Jedná se v podstatě o odporový trimr, jehož jezdec se pohybuje společně s hřídelí pohonu. Výstupní odpor je poté možné měřit pomocí můstku a A/D převodníku pro další digitální

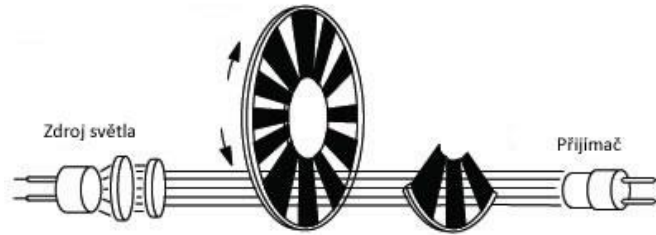
zpracování. Nevýhodou tohoto řešení je mechanické opotřebovávání odporové dráhy, která s postupem času mění své vlastnosti.



Obrázek 2.7 – Absolutní snímač polohy s 8-bitovým výstupem [Senzory, 2015]

2.3.2 Relativní snímače

Tato konstrukce snímačů polohy je od předchozí skupiny podstatně jednodušší, což ovšem přináší podstatná omezení. Jak už bylo řečeno, relativní snímače potřebují nějaký vztažný bod, od kterého je zjišťována hodnota polohy. Tento typ snímačů pracuje nejčastěji na optickém principu, existují ovšem i varianty s použitím Hallovy sondy na magnetickém principu. Konstrukce je podobná jako u absolutních snímačů, jedná se o optický přijímač a vysílač, mezi kterými je umístěn kotouč. V tomto případě se na kotouči střídají pouze průhledná a neprůhledná políčka. Pohybem kotouče tedy dochází k přerušování paprsku mezi vysílačem a přijímačem, čímž vznikají na výstupu impulzy. Změna polohy je potom přímo úměrná počtu impulzů od referenčního bodu, ve kterém musí být vždy rotor na začátku měření. Varianta s Hallovou sondou je odlišná pouze ve způsobu snímání impulzů a konstrukci kotouče, na který jsou připevněny permanentní magnety. Výhodou je jednoduchá konstrukce a informace o poloze v digitální formě, která je pro zpracování v moderních servopohonech nezbytná. Popsaná konstrukce trpí kromě nutnosti výchozí polohy ještě jednou zásadní nevýhodou, a to je nemožnost detekce směru pohybu. Tento nedostatek je možné vyřešit přidáním druhého, posunutého měřicího ústrojí, které generuje fázově posunuté impulzy a poté je možné detekovat směr pohybu podle smyslu fázového posunu.



Obrázek 2.8 – Konstrukce relativního snímače polohy [Pohonná technika, 2015]

3 MOŽNOSTI DÁLKOVÉHO OVLÁDÁNÍ

Výběr komunikačního modulu v mobilní robotice se řídí několika kritérii. V první řadě nás zajímá požadovaný dosah komunikace a s ním spojené další parametry, jako je vysílací výkon, se kterým přímo souvisí spotřeba, která musí být uvažována při návrhu napájecích obvodů, jejichž jádrem je v mobilní robotice ve většině případů akumulátor. Dalším pohledem na výběr komunikačního modulu je jeho celkové využití, to v nejjednodušším případě znamená pouze jednosměrný přenos povelů z ovládacího zařízení do řídicí jednotky robota. Komunikace může být také využívána k účelům diagnostiky nebo přenosu informací z různých senzorů robota opačným směrem. Potom je vhodné uvažovat další parametr komunikačního modulu, což je přenosová rychlost, jejíž význam roste s objemem přenášených dat.

3.1 XBEE MODULY

Tyto komunikační moduly jsou častou volbou z důvodů možností, jednoduchosti komunikace a nízké spotřeby. Komunikace je založena na standardu IEEE 802.15.4 v kmitočtovém pásmu 2,4 GHz, který umožňuje topologii komunikace Point-to-point, tedy mezi ovládacím zařízením a robotem, která je pro tento účel nejpoužitelnější. Další možností je topologie Point-to-multipoint, kterou by bylo možné využít např. při synchronizovaném pohybu více robotů, protože zde vzniká možnost komunikovat mezi ovládacím zařízením a roboty navzájem. V praxi se často používá varianta XBee modulu od firmy Digi International s označením XBee 802.15.4 a XBee-PRO 802.15.4. Tyto dvě varianty se vzájemně liší z hlediska vysílacích výkonů a dalších návazných parametrů. Pro zvýšení citlivosti analogové části je zde možnost připojení antény pomocí konektoru UFL. Komunikace mezi modulem a řídicím mikroprocesorem je řešena obousměrným rozhraním UART s přenosovou rychlostí 1200 bps – 250 kbps. Nastavování a ovládání

funkcí je umožněno pomocí příkazů. Je zde i možnost využití interních digitálních vstupů/výstupů a analogových vstupů, řešených 10-ti bitovými A/D převodníky.



Obrázek 3.1 – Komunikační modul XBee 802.15.4 PRO [GraviTech, 2015]

3.2 BLUETOOTH MODULY

Zvláště přenosná zařízení, jako jsou mobilní telefony a notebooky, disponují dnes tímto komunikačním rozhraním. Přenosový protokol je definován standardem IEEE 802.15.1 v kmitočtovém pásmu 2,4 GHz. Jedná se o velice flexibilní komunikační rozhraní, které umožňuje klasický přenos dat prostřednictvím simulace sériové linky RS232, kterou zabezpečuje protokol s názvem SPP. Bluetooth je často používán i ke komunikaci bezdrátových periférií s počítačem, kde se jedná o protokol HID. V neposlední řadě toto komunikační rozhraní také umožňuje přenos zvukového streamu, který je nejčastěji modulován signálem PCM a výsledná zvuková data jsou přenášena pomocí protokolu HFP do cílového zařízení. Dosah klasických komunikačních modulů se obvykle pohybuje v okruhu 20 m na přímou viditelnost. Nejnovější verze Bluetooth rozhraní má označení 4.0 a existují dvě verze, které se liší zejména hardwarovým řešením koncového stupně. První variantou je Single Mode, který má obchodní označení Low Energy, kde už název napovídá, že se jedná o komunikační moduly s velmi nízkou spotřebou. Toto řešení ovšem trpí zásadním nedostatkem, tedy nekompatibilitou s předchozími verzemi Bluetooth. Naopak výhodou je možnost definice vlastních protokolů pomocí značkovacího jazyka XML. Nekompatibilita je odstraněna druhou variantou s názvem Dual Mode, která v sobě slučuje funkce klasického Bluetooth a Bluetooth Low Energy. Z konkrétních vyráběných modulů je zvláště zajímavý produkt společnosti Bluegiga s označením BLE121RL Long Range. Z obchodního označení je zřejmé, že se jedná o Bluetooth verze 4.0 Low Energy, která je založena na integrovaném obvodu CC2541 společnosti Texas Instruments, který má implementovány funkce tohoto komunikačního rozhraní. Jedná se o mikroprocesor, nad jehož instrukční sadou vytvořila společnost Bluegiga svůj programovací jazyk s názvem BGScript, který podstatně

usnadňuje programování samotného komunikačního modulu bez znalosti instrukční sady pomocí jednoduchých funkcí. Význačnou vlastností je podstatné zvětšení dosahu tohoto modulu, který je udáván až 450 m na přímou viditelnost při maximálním odběru 36 mA. Výhodou je také přítomnost periférií, jako jsou externí přerušeni, čítače, A/D převodníky a komunikační sběrnice UART a SPI, které se dají jednoduše ovládat zmiňovanými funkcemi jazyka BGScript.

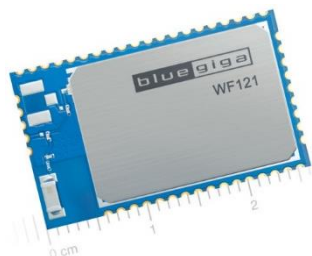


Obrázek 3.2 – Komunikační modul Bluegiga BLE121RL [Bluegiga, 2015]

3.3 WI-FI MODULY

Bezdrátové komunikační rozhraní Wi-Fi je v dnešní době jedno z nejrozšířenějších především z důvodu, že se ve spojení s protokolovou sadou TCP/IP používá jako bod pro bezdrátové připojení k internetu prakticky veškerých mobilních zařízeních. Opět se jedná o kmitočtové pásmo 2,4 GHz s přenosovým protokolem definovaným standardy IEEE 802.11 pro fyzickou a spodní část spojové vrstvy ISO/OSI komunikačního modelu. Spojová vrstva má tedy dvě části, zmiňovanou spodní část nazýváme Media Access Control, ta zabezpečuje identifikaci koncového zařízení v lokální síti dle fyzické adresy MAC. Vrchní část spojové vrstvy je nazývána Logical Link Control, definovaná standardem IEEE 802.2 a stará se o identifikaci protokolu síťové vrstvy, který bude použit k dalšímu zpracování přijatého datového rámce. Pokud nám jde pouze o bezdrátový přenos informace, stačí komunikační modul, který dokáže obsluhovat pouze výše popsané vrstvy. Použití Wi-Fi je většinou spojeno se síťovou komunikací, je zde nutné implementovat další dvě vrstvy ISO/OSI modelu, tedy síťovou a transportní. Implementace těchto vrstev je často označována jako TCP/IP stack. Běžně vyráběné Wi-Fi moduly existují ve dvou variantách, buď mají implementovány pouze tři spodní vrstvy, zajišťující bezdrátový přenos informace mezi dvěma body, nebo moduly, které obsahují i samotnou implementaci TCP/IP stacku a jsou plnohodnotnými přístupovými body, či klienty v počítačových sítích. Existuje i kombinace těchto dvou variant. Modul, zabezpečující komunikaci pouze v prvních třech vrstvách, s možností

vlastní implementace TCP/IP stacku do mikroprocesoru, který obsahuje Wi-Fi modul k těmto účelům. Tato možnost je ovšem značně komplikovaná, ale při některých konkrétních aplikacích nezbytná. Z komerčních Wi-Fi modulů je zajímavý produkt společnosti Bluegiga s označením WF121, který bude dále podrobně popsán v praktické části této práce.



Obrázek 3.3 – Komunikační modul Bluegiga WF121 [Bluegiga, 2015]

4 POPIS POUŽITÝCH HARDWAROVÝCH PROSTŘEDKŮ

4.1 POUŽITÝ MIKROPROCESOR

Jedná se o hlavní část řídicí jednotky robota, proto byl na výběr mikroprocesoru kladen největší důraz. Z důvodu paměťových nároků na řídicí software a uvažování možnosti připojení různých senzorů, byl vybrán 8-bitový mikroprocesor společnosti Atmel s označením ATmega128A. Jedná se o mikroprocesor z rodiny AVR Harvardské architektury, což znamená, že paměť pro uložení programu je oddělená od paměti pro uložení dat. Z pohledu instrukční sady se jedná o mikroprocesory typu RISC, jejichž instrukční sada je redukována na malý počet jednoduchých instrukcí. Pro zajištění chodu programu obsahuje mikroprocesor 32 pracovních registrů o šířce slova 8 bitů s paměťovými místy v paměti RAM. Do těchto registrů je možné pomocí určených instrukcí ukládat operandy, se kterými se nadále pracuje prostřednictvím instrukcí určených k vykonávání různých operací, jejichž výsledky jsou také ukládány do registrů k možnosti dalšího zpracování. Paměť RAM kromě registrů obsahuje i zásobník, který v případě programování s použitím assembleru vytváří programátor. Pokud je k programování použit vyšší programovací jazyk, obvykle C nebo C++, dochází k vytvoření zásobníku automaticky kompilátorem. Zásobník slouží k uložení návratových adres při volání podprogramů, nebo při skoku programu do obslužné rutiny přerušení. Velikost paměti RAM je 4 KB. Pro uložení programu slouží paměť typu flash o velikosti 128 KB, organizovaná po 16-bitových slovech. Pokud je potřeba trvalé uložení dat mimo programovou paměť, existuje možnost využití EEPROM o velikosti 4 KB. Označení

mega mají mikroprocesory z rodiny AVR, které jsou vybaveny velkým množstvím periférií. Jejich použití je tedy univerzální, což je také jeden z důvodů jeho výběru. Mezi zmiňované periferie patří čítače/časovače, generátory PWM signálu, externí přerušeni, A/D převodníky a hlavně široká škála sériových sběrnic. Mikroprocesor obsahuje dvě nezávislá rozhraní USART, sběrnici SPI a I²C, která je společností Atmel označovaná TWI, ale jedná se pouze o rozdílný název a sběrnice jsou plně kompatibilní. Jako binární vstupy/výstupy je možné použít sedm 8-pinových portů s označením A – G. Zdrojem hodinového signálu je interní kalibrovaný RC oscilátor, nebo externí krystalový rezonátor, zajišťující vyšší přesnost, která je potřeba zejména v aplikacích, které vyžadují přesné časování často při použití časovačů a A/D převodníků. Maximální taktovací kmitočet je 16 MHz. Písmeno A v označení mikroprocesoru znamená rozšířený rozsah napájecího napětí od 2,7 V do 5,5 V, což zabezpečuje kompatibilitu s dalšími integrovanými obvody, jak pro logiku 3,3 V, tak pro klasickou TTL logiku 5 V. Mikroprocesor ATmega128A je vyráběn ve dvou variantách pouzder pro povrchovou montáž, TQFP a QFN/MLF se 64 vývody.



Obrázek 4.1 – Mikroprocesor ATmega128 v pouzdře TQFP [Dunia Elektronika, 2015]

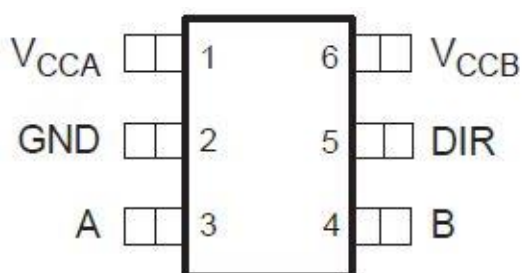
4.2 PŘEVODNÍK NAPĚŤOVÝCH ÚROVNÍ

Integrované obvody tohoto typu používáme v aplikacích, kde se v jednom obvodu vyskytují zařízení, pracující s různou velikostí logických napěťových úrovní. Pokud je signálů, u kterých potřebujeme změnit logiku malé množství, můžeme tento problém snadno vyřešit vhodným zapojením např. FET tranzistorů. Pokud chceme co nejvíce zjednodušit plošný spoj, je vhodnější použít integrovaný obvod určený pro tyto účely. Jedním ze zastupitelů obvodů tohoto typu je produkt od společnosti Texas Instruments s označením SN74LVC1T45, který obsahuje jeden kanál, ve kterém se mohou obousměrně měnit logické úrovně mezi napájecím napětím integrovaného obvodu převodníku a zvolenou napěťovou úrovní. Rozsah napájecího napětí je v intervalu

1,65 V – 5,5 V, takže je tento převodník vhodný pro přechod mezi logickými úrovněmi 1,8 V nebo 3,3 V a 5 V. Jedná se o pouzdro se 6 vývody, kde V_{CCA} a V_{CCB} jsou úrovně napětí, mezi kterými se přechází. Směr převodu je řízen signálem DIR, který se ovládá logickou úrovní V_{CCA} a A, B je samotný komunikační kanál. Tento obvod se často používá jako převodník pro sběrnice, proto je významným parametrem přenosová rychlost, která může být až 420 Mbps, při převodu logických úrovní mezi 3,3 V a 5 V. Vyrábí se varianty, které umožňují vícekanálový převod.

Tabulka 4.1 – Pravdivostní tabulka integrovaného obvodu SN74LVC1T45

<i>Logická úroveň na vývodu DIR</i>	<i>Směr převodu</i>
0	z B do A
1	z A do B

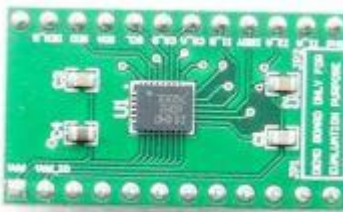


Obrázek 4.2 – Popis vývodů integrovaného obvodu [TEXAS INSTRUMENTS, 2004]

4.3 MEMS AKCELEROMETR A GYROSKOP

Senzory tohoto typu, vyráběné technologií MEMS, jsou v dnešní době velmi oblíbené z důvodu svých vlastností a velikostí pouzder integrovaných obvodů. Senzor od společnosti ST Microelectronics s označením LSM330DLC, v sobě slučuje tříosý akcelerometr a gyroskop. Jedná se o MEMS technologii, snímačem měřené veličiny je tedy kondenzátor, jehož kapacita se mění v závislosti na dynamickém zrychlení, působícím v příslušné ose. Pro každou osu je samostatný kondenzátor, který se při měření zrychlení v příslušné ose přepne pomocí multiplexoru na vstup společné analogové části. Kapacita kondenzátoru, odpovídající zrychlení, je převedena na napětí, které je následně zesíleno a přivedeno na vstup A/D převodníku. Komunikace je možná pomocí SPI a I²C sběrnice. Senzor obsahuje paměťové registry, které slouží k nastavování funkcí a také k uložení změřených hodnot veličin. Podobně můžeme senzor

pomocí sběrnice jednoduše nastavovat zápisem do příslušných registrů a naopak získávat požadované změřené hodnoty čtením příslušných registrů. Tento integrovaný obvod dokáže měřit zrychlení až $\pm 16 \text{ g}$ a úhlovou rychlost až $\pm 2000 \text{ dps}$. Informace o změřených veličinách má 16-bitovou přesnost. Zajímavá je také funkce generování přerušení na různé podněty, např. volný pád nebo pohyb. Napájení je oddělené, pro analogovou část by se mělo pohybovat v rozsahu mezi $2,4 \text{ V} - 3,6 \text{ V}$ a pro digitální část dle použité logiky $1,8 \text{ V}$ nebo $3,3 \text{ V}$. Plošný spoj řídicí jednotky robota byl navržen pro možnost ručního pájení, které není možné s pouzdem LGA-28L, ve kterém je vyráběn tento senzor, proto byl zvolen hotový modul o velikosti vývodového pouzdra DIL24. Tento modul obsahuje pouze integrovaný obvod a několik externích součástek zapojených dle datasheetu.

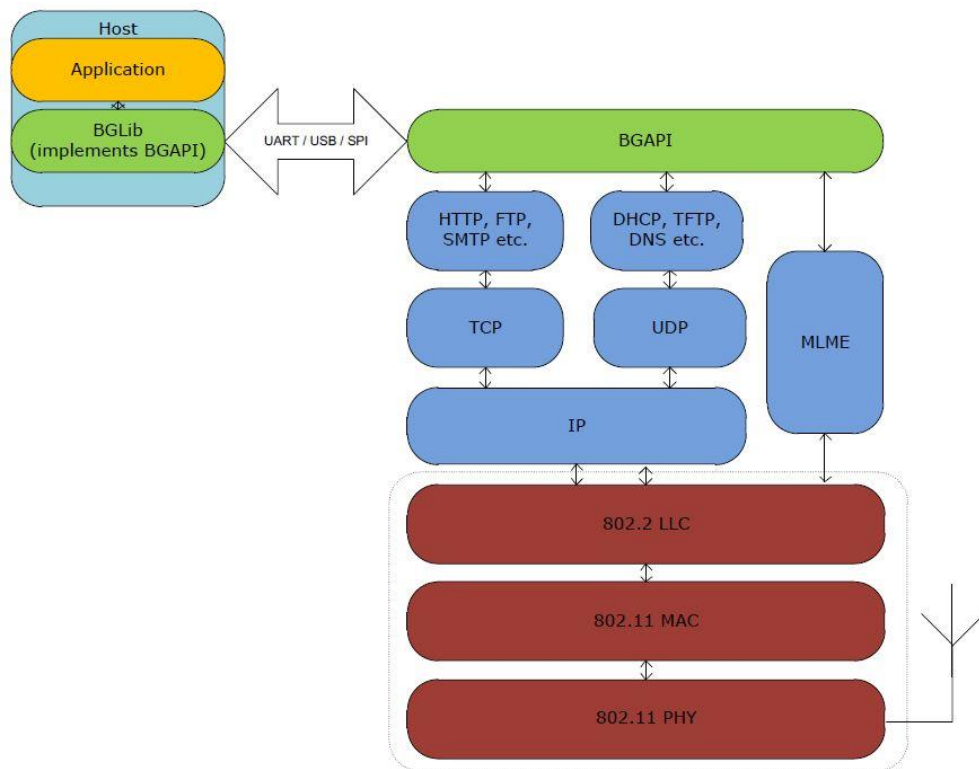


Obrázek 4.3 – Modul se senzorem LSM330DLC [MOUSER ELECTRONICS, 2013]

4.4 WI-FI MODUL

Použitý Wi-Fi modul obsahuje implementaci fyzické a spojové vrstvy ISO/OSI modelu, která zajišťuje bezdrátový přenos informace mezi dvěma body. Protože je komunikace mezi ovládacím softwarem a robotem založena na protokolové sadě TCP/IP, nechybí u tohoto komunikačního modulu ani implementace těchto potřebných protokolů síťové a transportní vrstvy. Tato aplikace to nevyžaduje, ale nechybí ani podpora některých protokolů aplikační vrstvy, jako je HTTP, SMTP, FTP a DNS. Vnitřní architektura Wi-Fi modulu je tvořena dvěma integrovanými obvody. První obvod zajišťuje bezdrátovou komunikaci dle standardů IEEE 802.11 a 802.2, které popisují spodní dvě vrstvy ISO/OSI modelu pro komunikační rozhraní Wi-Fi. Druhým integrovaným obvodem je mikroprocesor od společnosti Microchip s označením PIC32MX695H. Jedná se o architekturu pracující s šířkou slova 32 bitů s jádrem MIPS M4K a maximálním taktovacím kmitočtem 80 MHz . Pro uložení programu slouží paměť typu FLASH o velikosti 512 KB . Paměť typu RAM, využívaná pro chod programu, má velikost 128 KB . Tento mikroprocesor má v architektuře Wi-Fi modulu

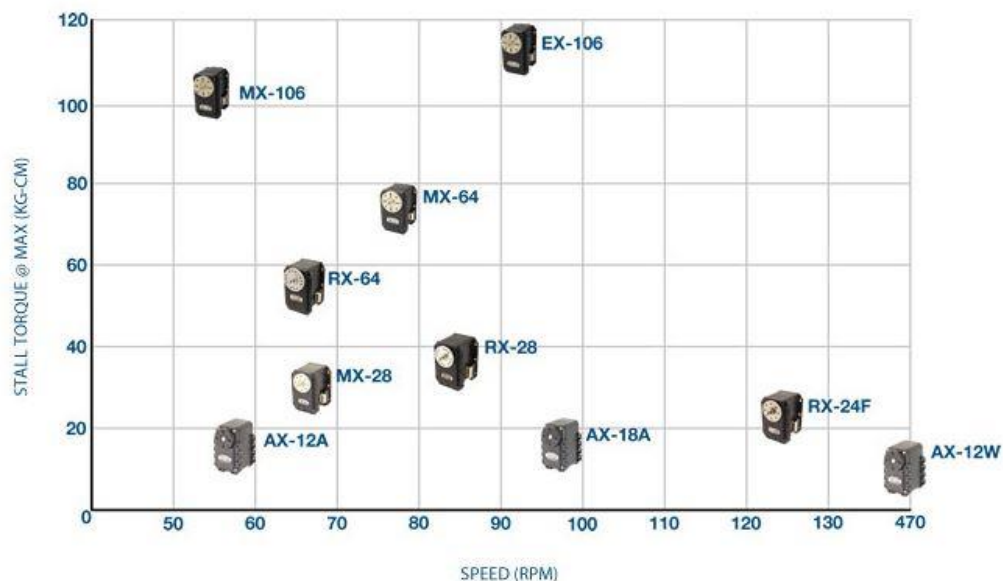
funkci obsluhy prvního integrovaného obvodu a je v něm umístěná implementace TCP/IP stacku a ostatních protokolů aplikační vrstvy. Firmware Wi-Fi modulu je tedy uložen v paměti tohoto obvodu. Upgrade firmwaru je možný dvěma způsoby, a to pomocí přímého programování mikroprocesoru PIC32MX695H, jehož programovací signály jsou vyvedeny do pouzdra samotného Wi-Fi modulu. Druhou možností je upgrade pomocí rozhraní UART za použití bootloADERu, který umožňuje firmware po skončení datového přenosu nastartovat. Ovládání a nastavování funkcí Wi-Fi modulu je řešeno pomocí tzv. BGAPI, což je ovládací rozhraní, které si uživatel povolí v příslušném XML souboru na zvolené sběrnici UART. Potom je možné ovládat funkce komunikačního modulu prostřednictvím příkazových rámců, dle dokumentace k BGAPI, nebo používat BGLib, což je obsluha API, implementovaná pro hostitelské mikroprocesory. Další možností je programování funkcí Wi-Fi modulu prostřednictvím jazyka BGScript, který bude později popsán v této práci. Pro jednoduché a rychlé seznámení s funkcemi modulu je výhodné využít software Bluegiga WF121 GUI, který generuje příkazové rámce pro obsluhu BGAPI na stisk funkčních tlačítek. Napájení Wi-Fi modulu je 3,3 V a některé vývody tolerují napěťové úrovně klasické TTL logiky, tedy 5 V. Význačnou vlastností je možnost využití periférií jako jsou časovače, A/D převodníky a vstupně/výstupní porty.



Obrázek 4.4 – Blokové schéma Wi-Fi modulu [BLUEGIGA TECHNOLOGIES, 2014]

4.5 SERVOPOHONY

Pro aplikaci chodícího robota je kladem důraz na rychlost pohybu nohou, proto nás při volbě servopohonů zajímá zvláště tento parametr, definovaný většinou jako počet otáček za minutu. Vysokou rychlostí pohybu rotoru se vyznačují servopohony od společnosti Dynamixel, které budou dále popsány. Druhým parametrem, který nás při volbě pohonů zajímá, je kroutivý moment, který definuje sílu závislou na vzdálenosti od rotoru, kterou dokáže servopohon vyvinout. Tento parametr není v tomto případě důležitý, protože při konstrukci robota není uvažována větší zátěž, pro kterou by byl moment pohonů Dynamixel nedostatečný. Od zmiňovaného výrobce byl vybrán konkrétní model s označením AX-12A, který je z nabízených produktů výrobce jedním z nejzákladnějších, ale pro tuto aplikaci přesto plně vyhovuje v obou zmiňovaných parametrech. Porovnání všech nabízených variant servopohonů z hlediska rychlosti a kroutivého momentu je na následujícím obrázku.



Obrázek 4.5 – Porovnání servopohonů Dynamixel [Trossen Robotics, 2015]

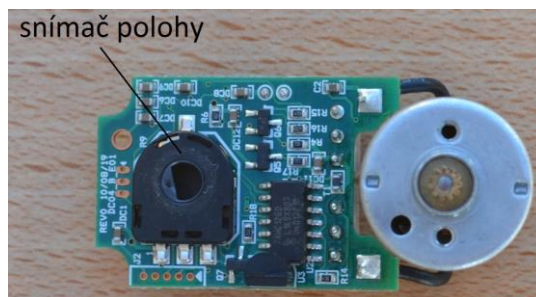
Varianta AX-12A má rozlišení pohybu rotoru $0,29^\circ$ v režimu servopohonu, který umožňuje volbu pozice v rozsahu 300° mezi krajními polohami. Existuje zde i možnost rotačního režimu, ve kterém se rotor otáčí zvolenou rychlostí a směrem o celých 360° . Komunikace je řešená prostřednictvím poloduplexního rozhraní UART v 5 V TTL logice. Nad klasickou sériovou linkou je vytvořena nadstavba komunikačním protokolem, umožňujícím adresaci jednotlivých servopohonů, které je možné paralelně připojovat na

společnou sběrnici. Šířka adresy je 8 bitů, což umožňuje obsluhovat až 254 zařízení na sběrnici, poslední adresa z rozsahu je tzv. broadcast, který umožňuje použití hromadného povelu pro více připojených pohonů. Komunikační rychlost na sběrnici je až 1 Mbps. Napájecí napětí se může pohybovat v rozsahu 9 – 12 V a je možné ho číst přímo z jednotlivých pohonů, stejně jako vnitřní teplotu systému a polohu rotoru. Celkový odběr v pohotovostním stavu je 50 mA, při pohybu rotoru je přírůstek odběru úměrný rychlosti a zatížení.



Obrázek 4.6 – Servopohon Dynamixel AX-12A [Trossen Robotics, 2015]

Jak už bylo řešeno dříve, servopohon se vyznačuje zpětnou vazbou, která slouží k regulaci rychlosti nebo polohy. Informace o poloze rotoru je v tomto případě získávána pomocí odporového trimru, jehož jezdec se pohybuje společně s rotorem pohonu. Hodnota polohy je tedy měřena absolutně, což nevyžaduje žádnou výchozí pozici při připojení napájení.



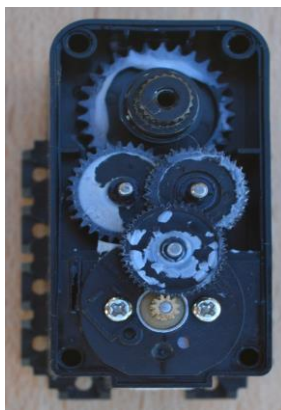
Obrázek 4.7 – Konstrukční řešení snímače polohy v servopohonu Dynamixel AX-12A

Jádrem řídicí elektroniky je mikroprocesor od společnosti Atmel s označením ATmega8, který měří a vyhodnocuje informace ze snímače polohy a realizuje zpětnovazební smyčku a regulátor při změně hodnoty polohy na žádanou hodnotu. Další důležitou funkcí mikroprocesoru je implementace komunikačního protokolu a zajištění komunikace s uživatelem nebo hostitelským zařízením. Pro možnost výchozího nastavení

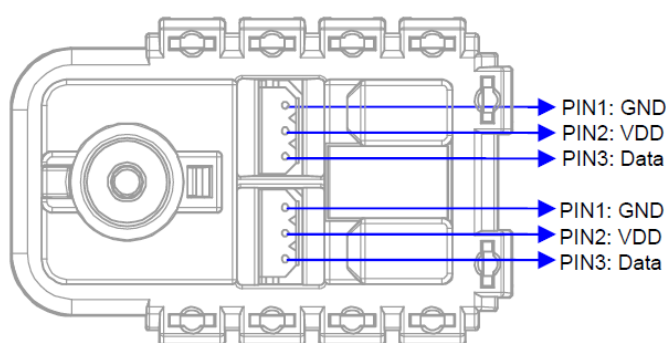
servopohonu je využita paměť EEPROM, ve které jsou uloženy informace o počáteční konfiguraci vybraných parametrů, jako je přenosová rychlost, adresa na sběrnici a různá omezení jak pozic rotoru, tak hraničních hodnot napájecího napětí, či provozní teploty. Paměť typu RAM je využita pro parametry, které se nastavují uživatelem, tím je myšlena žádaná hodnota pozice rotoru a rychlost pohybu. Dále jsou tu uloženy hodnoty veličin, které pohon v provozním stavu pravidelně vzorkuje, jedná se zejména na napájecí napětí, provozní teplotu a aktuální pozici rotoru. Obě dvě paměti mají společný adresní prostor a instrukce komunikačního protokolu umožňují z těchto paměťových míst číst, nebo do nich naopak zapisovat. Některá paměťová místa podporují obě instrukce, některá jsou pouze pro čtení. Použitý mikroprocesor má pouze duplexní rozhraní UART, proto je využit integrovaný obvod 74HC126, což je sběrniceový driver, který umožňuje sloučení datových signálů RXD a TXD mikroprocesoru na jeden společný signál vyvedený na konektor servopohonu. Použitým motorem je stejnosměrný motor s permanentními magnety, jehož otáčky jsou ovládány PWM signálem, který je generován mikroprocesorem. Směr otáčení tohoto typu elektromotorů je řízen polaritou napájecího napětí, což zabezpečuje integrovaný obvod SUF2001. Jedná se v podstatě o dva MOSFET tranzistory v jednom pouzdře, které slouží k otáčení polarity napájecího napětí a zesilování generovaného PWM signálu. Z toho vyplývá, že je tento obvod zapojený jako klasický H-můstek. Převodovou soustavu tvoří pět ozubených kol, která převádějí vysoké otáčky stejnosměrného motoru na vysoký kroutivý moment hřídele servopohonu.



Obrázek 4.8 – Řídicí jednotka a stejnosměrný motor servopohonu Dynamixel AX-12A



Obrázek 4.9 – Převodová soustava servopohonu Dynamixel AX-12A

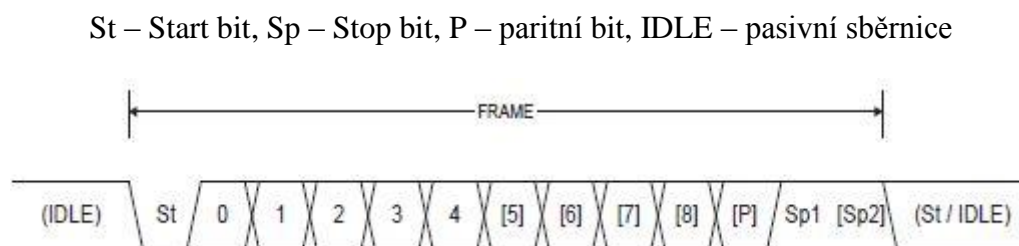


Obrázek 4.10 – Zapojení konektorů servopohonu Dynamixel AX-12A [Mbed, 2015]

4.6 SBĚRNICE UART

„Jedná se o sériovou, asynchronní, duplexní sběrnici, což znamená, že nedochází k přenosu hodinového signálu mezi komunikujícími zařízeními a je možné v každém okamžiku přenášet data oběma směry. Přenos dat probíhá prostřednictvím dvou vodičů připojených na piny RXD a TXD „kříženým“ způsobem. Základní přenášenou jednotkou je rámec, který může mít 5 až 9 bitů, dle nastavení sběrnice. Přenos rámce je zahájen tzv. Start bitem, poté následuje přenášený rámec, jehož přenos je signalizován jedním nebo dvěma Stop bity. Přenosová rychlost baudrate určuje počet přenesených bitů za sekundu a je konfigurovatelná. U asynchronních přenosů je velmi důležité časování, jehož přesnost je závislá na zdroji hodinového signálu, který řídí generátor přenosových rychlostí. Při použití UART sběrnice ke komunikaci s mikroprocesorem je výhodnější zvolit zdrojem hodinového signálu externí krystal, který je přesnější než interní RC oscilátor. Volba kmitočtu tohoto krystalu je závislá na zvolené přenosové rychlosti, protože pokud není tento kmitočet beze zbytku dělitelný přenosovou rychlostí, vzniká určitá chyba, která se při přenosu každého bitu sčítá a to může způsobit problémy při

přenosu většího bloku dat. Sběrnice UART má prostředky, které umožňují kontrolu integrity dat, jedná se o tzv. paritu, kdy při sudém počtu logických jedniček dochází k doplnění jedničky na místo paritního bitu, potom se jedná o lichou paritu, analogicky funguje sudá parita. Následující obrázek popisuje přenos jednoho rámce“ [ČERNOHLÁVEK, 2013].



Obrázek 4.11 – Diagram přenosu rámce po UART sběrnici [ČERNOHLÁVEK, 2013]

5 POPIS POUŽITÝCH SOFTWAREVÝCH PROSTŘEDKŮ

5.1 KOMUNIKAČNÍ PROTOKOL DYNAMIXEL

Komunikace s použitými servopohony probíhá prostřednictvím sériové sběrnice UART se standardem RS232. Jednotlivá zařízení jsou na sběrnici připojena paralelně, to znamená, že veškerou komunikaci mohou přijímat všechny připojené servopohony. Sběrnice je poloduplexní, proto je komunikační protokol implementován tak, že komunikaci může zahájit pouze hostitelské zařízení a příslušný servopohon pouze odpovídá na danou instrukci. Odpověď nese informaci o úspěšném provedení, nebo v opačném případě chybovou hlášku. Samotný standard RS232 neumožňuje adresaci zařízení na sběrnici, která je v tomto případě nezbytná. Komunikační protokol Dynamixel tedy definuje strukturu komunikačního rámce, která tuto funkci umožňuje. Adresa servopohonu má šířku 8 bitů, na jedné sběrnici může být tedy připojeno až 254 zařízení v adresním rozsahu 0x00 – 0xFD, adresa 0xFE je definována jako broadcast, jehož význam bude dále popsán. Instrukční rámec má pevně danou strukturu, kterou popisuje následující tabulka.

Tabulka 5.1 – Struktura instrukčního rámce komunikačního protokolu Dynamixel

0xFF	0xFF	Adresa	Délka	Instrukce	Parametr 1 – Parametr N	Kontrolní součet
------	------	--------	-------	-----------	-------------------------	------------------

První dva byty jsou konstantní a označují začátek rámce. Adresa je hodnota ve výše definovaném rozsahu a délka je počtem parametrů $N + 2$. Instrukce určuje, o jakou akci se jedná. Seznam nejpoužívanějších je uveden v následující tabulce. Význam a počet parametrů je závislý na konkrétní instrukci a kontrolní součet je definován jako nejméně významný byte z hodnoty získané následujícím vztahem.

$$KS = \text{inverze}(\text{adresa} + \text{délka} + \text{instrukce} + \text{parametr } 1 + \dots + \text{parametr } N) \quad (5.1)$$

Tabulka 5.2 – Tabulka nejpoužívanějších instrukcí komunikačního protokolu Dynamixel

<i>Instrukce</i>	<i>Funkce</i>	<i>Hodnota</i>	<i>Počet parametrů</i>
PING	Zkouška komunikace	0x01	0
READ_DATA	Zápis do paměti	0x02	2
WRITE_DATA	Čtení z paměti	0x03	2 a více
SYNC_WRITE	Zápis do paměti více servopohonů	0x83	4 a více

Instrukce PING slouží pouze ke zkoušce komunikace s příslušným servopohonem nebo ke zjištění, zda je dané zařízení připojené na sběrnici, či v aktivním stavu. Hostitelské zařízení odešle rámec s instrukcí PING a čeká na odpověď. Nastavování parametrů a ovládání servopohonu je řešeno zápisem do paměti typů EEPROM a RAM, které mají společný adresní prostor, kde má každá adresa konkrétní význam. Naopak informace o servopohonu můžeme získat čtením z této paměti, k čemuž slouží instrukce READ_DATA. Prvním parametrem této instrukce je počáteční adresa, od které začíná čtení a druhým délka požadovaných dat, která je vrácena v odpovědi od příslušného servopohonu. Zápis do paměti umožňuje instrukce WRITE_DATA, která má obdobnou strukturu komunikačního rámce. Počátečním parametrem je opět adresa paměťového místa, do kterého je započat zápis, a dalšími parametry jsou konkrétní hodnoty zapsané do paměti. Šířka paměťového místa je 8 bitů. Pokud ovládáme současně více servopohonů na sběrnici, je zbytečně komplikované a neefektivní ovládat jednotlivé servopohony samostatnými instrukcemi. Tento problém řeší instrukce s názvem SYNC_WRITE, která umožňuje s použitím broadcast adresy ovládat libovolný počet servopohonů jedním datovým rámcem, což je efektivnější jak pro komunikaci na sběrnici, tak pro práci hostitelského zařízení. Příklad sestavení datového rámce s touto instrukcí je v následující tabulce [Trossen Robotics, 2015].

Tabulka 5.3 – Datový rámec instrukce SYNC_WRITE

0	1	2	3	4	5	6	7	8	9	10	11	12	13
FF	FF	FE	0E	83	01	1E	02	FF	03	02	00	00	B4

Komunikační rámec má celkem 13 bytů, zde je popsán jejich jednotlivý význam:

Byte 0 a 1 – označení počátku rámce

Byte 2 – broadcast adresa

Byte 3 – délka dat v rámci

Byte 4 – volba instrukce SYNC_WRITE

Byte 5 – adresa prvního servopohonu

Byte 6 – počáteční adresa zápisu dat (nastavení pozice)

Byte 7 – délka zapisovaných dat

Byte 8 – první datový byte (L byte pozice)

Byte 9 – druhý datový byte (H byte pozice)

Byte 10 – adresa druhého servopohonu

Byte 11 – první datový byte (L byte pozice)

Byte 12 – druhý datový byte (H byte pozice)

Byte 13 – kontrolní součet, dle uvedeného vztahu

Takto definovaný rámec nastaví pozici servopohonu s adresou 0x01 na hodnotu 1023 a pozici servopohonu s adresou 0x02 na hodnotu 0, jedním zápisem na sběrnici.

5.2 BLUEGIGA BGSCRIPT

Jak už bylo řečeno dříve při hardwarovém popisu použitého Wi-Fi rozhraní, tak je zde možnost programování mikroprocesoru, který je součástí samotného komunikačního modulu. Jedná se o 32-bitovou architekturu, u které je programování značně komplikované, což řeší nástroj od společnosti Bluegiga s názvem BGScript. Tato nadstavba nad jazykem C je jednoduchou možností, jak může uživatel programovat funkce Wi-Fi modulu bez hostitelského zařízení a dokonce bez znalostí hardwaru a instrukční sady samotného mikroprocesoru. BGScript má vlastní kompilátor, který generuje dva typy souborů, klasický .hex a .dfu. Jedná se o mikroprocesor od společnosti Microchip, proto je možné soubor .hex nahrát do paměti pomocí programátoru PicKit 3. Soubor .dfu slouží k nahrávání firmwaru pomocí sběrnice UART, kde dochází k datovému přenosu firmwaru a poté k jeho naboťování s použitím bootloaderu. Tento

programovací jazyk má dva základní typy programových konstrukcí. Prvním typem je konstrukce s názvem call, což je příkaz, jaká akce se má v těle s příslušnými parametry vykonat. Druhá programová konstrukce je event, která signalizuje, že došlo k vykonání požadované akce, nebo naopak došlo k nějaké chybě, která akci znemožnila. Tímto způsobem je možné zajistit samostatnou funkci Wi-Fi modulu bez účasti hostitelského zařízení. BGScript umožňuje nastavení vstupně/výstupních pinů, interruptů, časovačů a dalších periferních zařízení Wi-Fi modulu. Zde je uveden příklad příkazu, který vytváří Access Point s příslušných SSID a události signalizující úspěšné vytvoření.

Příkaz k vytvoření Access Pointu s SSID Quadruped Robot:

```
call sme_start_ap_mode(channel, security, ssid_len, ssid_data)
call sme_start_ap_mode(3, 0, 15, "Quadruped Robot")
```

channel – kanál, na kterém je vytvořen Access Point (kanál 3)

security – zabezpečení sítě (bez zabezpečení)

ssid_len – délka řetězce SSID

ssid_data – řetězec SSID

Událost úspěšného vytvoření Access Pointu:

```
event sme_ap_mode_started(hw_interface)
```

hw_interface – 00 označující rozhraní Wi-Fi

BGScript není jedinou možností ovládní funkcí Wi-Fi modulu, je zde i druhá možnost, kterou je rozhraní s názvem BGApi. Toto rozhraní je možné spustit pomocí konfiguračního souboru XML na zvolené sběrnici, kterou může být UART, I²C nebo SPI. Jedná se o jasně definované rámce, které spouští funkce Wi-Fi modulu, ekvivalentní s příkazy v BGScriptu. Tímto způsobem je možné ovládat komunikační modul prostřednictvím hostitelského zařízení. V následujících tabulkách jsou uvedeny příklady rámců pro vytvoření Access Pointu a příslušné události pro stejné parametry jako v BGScriptu.

Tabulka 5.4 – Rámec BGAPI pro vytvoření Access Pointu

0	1	2	3	4	5	6
08	03	03	0B	kanál	zabezpečení	SSID (ASCII)
08	03	03	0B	03	00	51756164727570656420526F626F74

Byte 0 – Typ zprávy: příkaz

Byte 1 – Minimální délka rámce

Byte 2 – Třída zprávy: ovládání Wi-Fi

Byte 3 – ID zprávy

Tabulka 5.5 – Událost BGAPI signalizující úspěšné vytvoření Access Pointu

0	1	2	3	4
88	01	03	0A	Identifikátor Wi-Fi rozhraní
88	01	03	0A	00

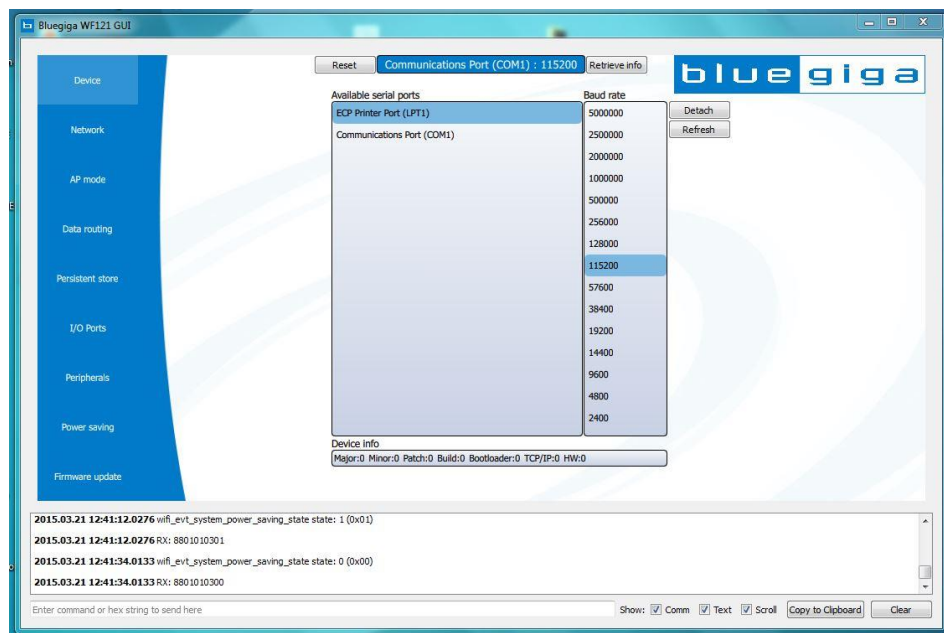
Byte 0 – Typ zprávy: událost

Byte 1 – Minimální délka rámce

Byte 2 – Třída zprávy: ovládání Wi-Fi

Byte 3 – ID zprávy

Při použití BGAPI na rozhraní UART je možné využít GUI od společnosti Bluegiga, které generuje příslušné rámce a umožňuje ovládat funkce Wi-Fi modulu přes sériový port počítače [BLUEGIGA TECHNOLOGIES, 2014].



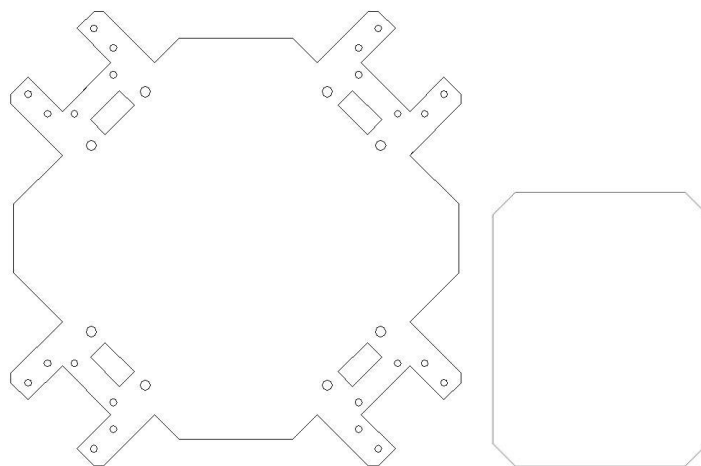
Obrázek 5.1 – Bluegiga WF121 GUI

6 KONSTRUKČNÍ ŘEŠENÍ

Konstrukční řešení můžeme rozdělit na tělo a nohy, které jsou z podstatné části tvořeny samotnými servopohony. Konstrukce robota je tvořena převážně plastovými a na zakázku vyrobenými díly z duralu. Tyto díly byly navrženy a nakresleny v softwaru AutoCAD 2013. Vytvořený návrh byl předán externí firmě, která díly vyrobila prostřednictvím CNC stroje z duralového plechu o tloušťce 1,5 mm.

6.1 KONSTRUKCE TĚLA

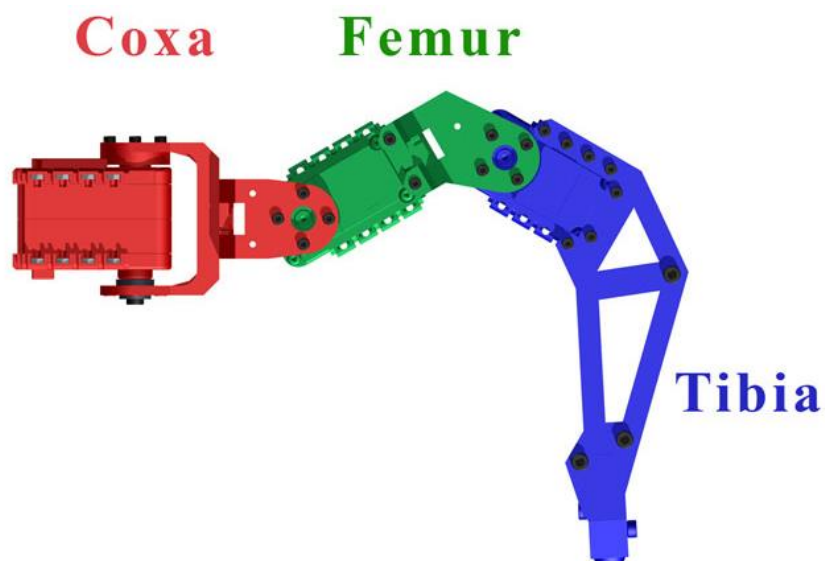
Tělo robota je tvořeno dvěma totožnými díly z duralu, které jsou spojeny samotnými servopohony a ke zpevnění konstrukce také distančními sloupky. Mezi těmito díly je umístěn akumulátor a vrchní díl nese plošný spoj řídicí jednotky, který je chráněn duralovou destičkou proti mechanickému poškození.



Obrázek 6.1 – Duralové díly tvořící tělo robota

6.2 KONSTRUKCE NOHY

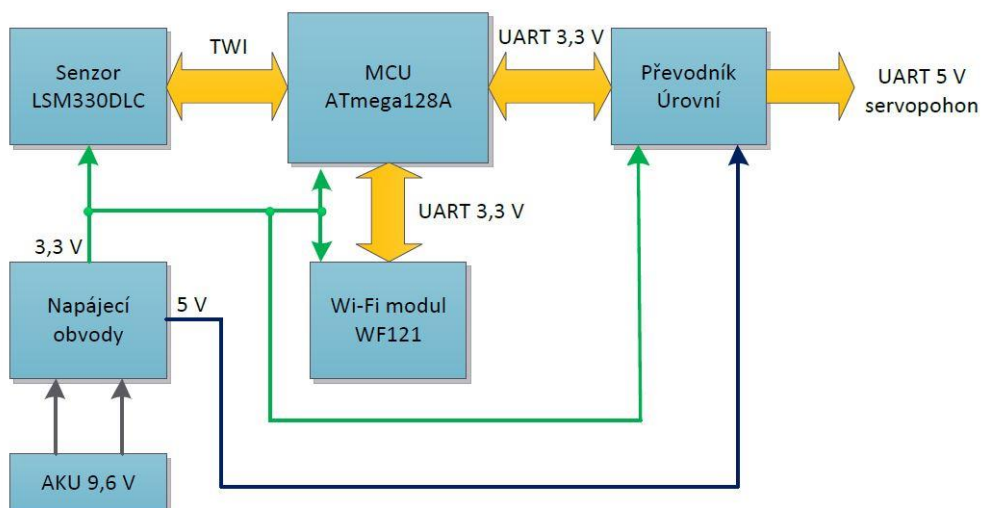
Noha má pochopitelně podstatně složitější konstrukci než tělo. Na obrázku níže jsou popsány jednotlivé části, které jsou přirovnány k anatomii čtyřnohého zvířete. Část Coxa je analogií stehenního kloubu, umožňující pohyb dopředu a dozadu. Femur je stehenní kost a Tibia holenní kost. Femur je konstrukčně řešen samotnými servopohony, spojenými pomocí plastových rámců, které jsou dodávány výrobcem. Tibia je tvořena dvěma díly z duralu spojenými pomocí distančních sloupků pro zajištění větší mechanické pevnosti. Každá noha obsahuje tři klouby, dostáváme tedy kinematický systém se třemi stupni volnosti.



Obrázek 6.2 – Konstrukce nohy robota [Trossen Robotics, 2015]

7 HARDWAROVÉ ŘEŠENÍ ROBOTA

Ústředním členem řídicí jednotky robota je mikroprocesor ATmega128A, napájený napětím 3,3 V. V obvodu se vyskytují dvě hladiny logických úrovní, tedy 3,3 V, pro zmiňovaný mikroprocesor, akcelerometr a Wi-Fi modul. Použité servopohony mají ovšem klasickou TTL logiku 5 V, proto je nutný vzájemný převod napěťových úrovní poloduplexního rozhraní UART. Pro ten účel je využit jednoboký integrovaný převodník úrovní SN74LVC1T45.

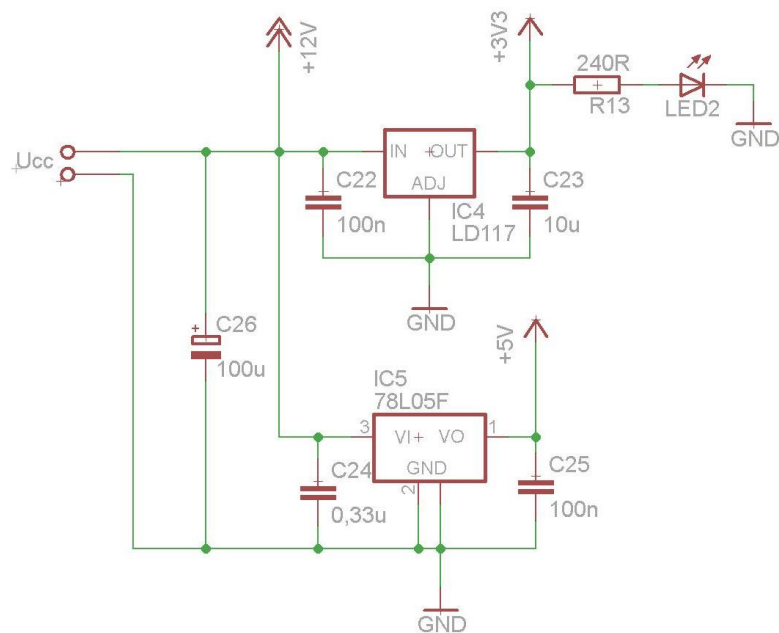


Obrázek 7.1 – Blokové schéma řídicí jednotky robota

Řídicí jednotka je z důvodu zmenšení plochy plošného spoje rozdělena na dvě části. Hlavní část nese mikroprocesor a s ním sdružené součástky, akcelerometr, napájecí obvody a v neposlední řadě konektory pro připojení servopohonů paralelně na sběrnici. Druhá část nese pouze komunikační modul a s ním sdružené součástky, včetně signalizačních LED. Mechanické propojení obou plošných spojů je řešeno pomocí kolíkové a dutinkové lišty. Propojena je komunikační sběrnice UART, přerušeni Wi-Fi modulu a samozřejmě napájení.

7.1 ZAPOJENÍ NAPÁJECÍCH OBVODŮ

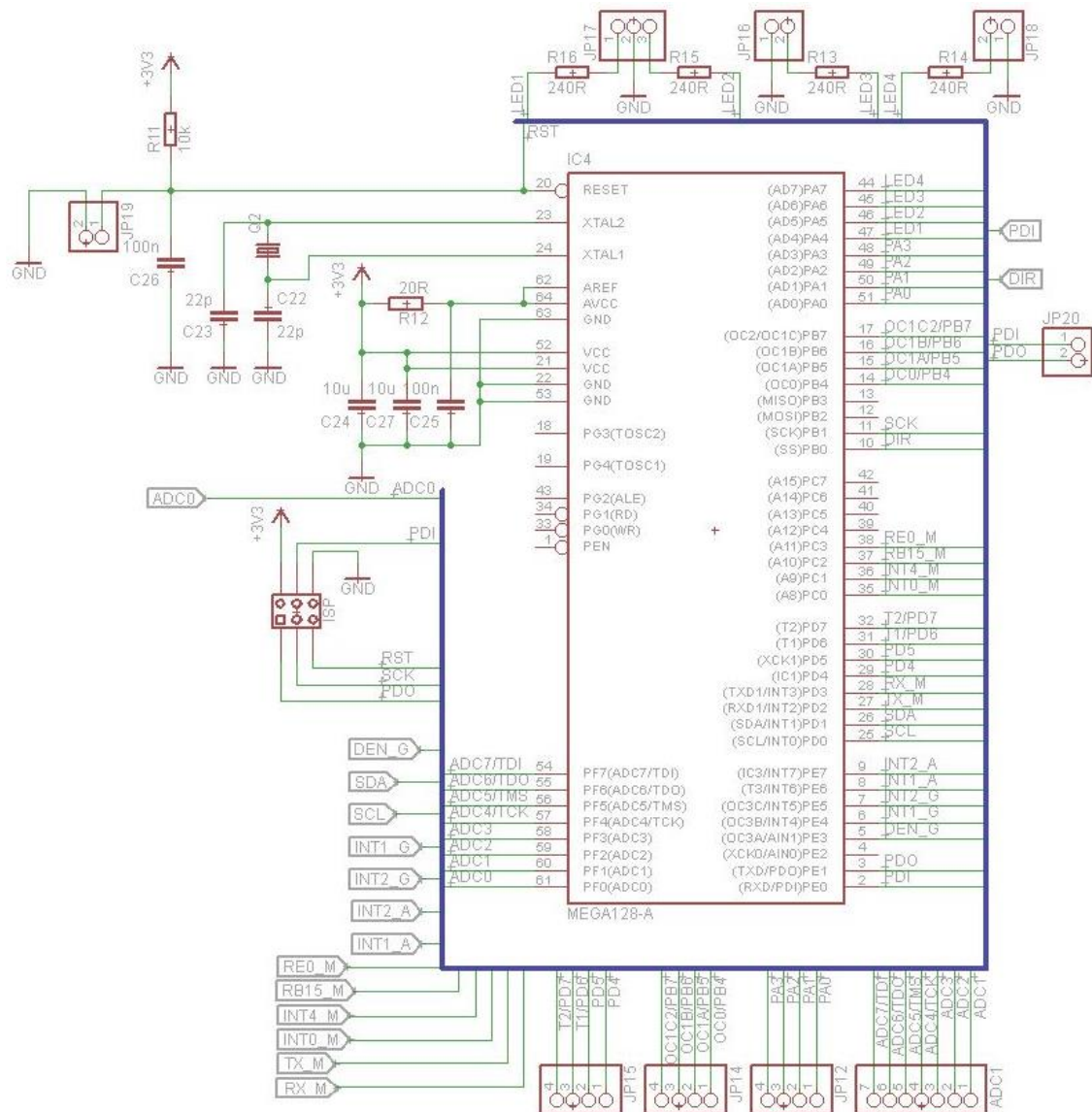
Napájení jednotlivých integrovaných obvodů je řešeno pomocí dvou samostatných stabilizátorů napětí. První stabilizátor s výstupním napětím 3,3 V napájí většinu součástek celého obvodu, proto je zvoleno provedení v pouzdře DPAK, které je robustní z pohledu proudového odběru, který může dosahovat až 1 A. Tento stabilizátor s označením LD1117 byl zvolen hlavně kvůli napájení Wi-Fi modulu, který dosahuje v AP módu špičkového odběru až 600 mA. Podstatnou část proudu tedy spotřebuje Wi-Fi modul. Napájení mikroprocesoru a akcelerometru je v řádu desítek mA, takže je proudová rezerva napájení dostatečná. Druhý stabilizátor s výstupním napětím 5 V slouží pouze jako zdroj referenčního napětí pro převodník úrovní sběrnice UART, zde tedy postačil klasický integrovaný obvod 78L05 s maximálním proudovým odběrem 100 mA. Napájení servopohonů je řešeno přímo z akumulátoru, připojeného na svorkovnici U_{cc} .



Obrázek 7.2 – Schéma zapojení napájecích obvodů

7.2 ZAPOJENÍ MIKROPROCESORU

Jak už bylo řečeno dříve, mikroprocesor ATmega128A je schopný pracovat při napájecím napětí 2,7 – 5,5 V bez omezení velikosti kmitočtu hodinového signálu. Z důvodu náročnosti výpočtu algoritmu, řídicího pohyb nohou robota při chůzi byl zvolen maximální kmitočet, a to je 16 MHz. Zdrojem hodinového signálu je externí krystalový rezonátor se dvěma paralelními kondenzátory, zapojenými dle datasheetu. Tyto kondenzátory, konkrétně C7 a C8, slouží jako externí součástky, které umožňují obvodu interního oscilátoru generovat signál.



Obrázek 7.3 – Schéma zapojení mikroprocesoru

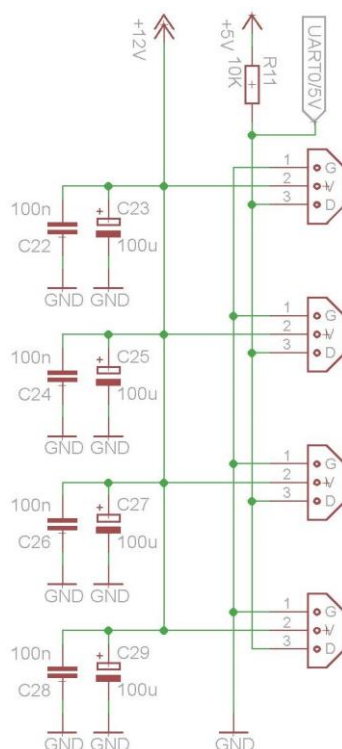
K programování mikroprocesoru je použito rozhraní ISP, jehož signály jsou vyvedeny na 6-pinový konektor, pro připojení programátoru. Připojení je odlišné od

ostatních mikroprocesorů z rodiny AVR, protože nejsou pro přenos dat použity signály MOSI a MISO, ale nahrazují je signály PDI a PDO. Piny pro připojení programovacího rozhraní JTAG jsou vyvedeny na kolíkovou lištu pro možnost ladění programu za chodu, při výskytu problémů s vývojem řídicího softwaru. Připojení napájení analogové částí a referenčního napětí pro A/D převodník je řešeno pomocí RC článku, který slouží k filtraci případných zákmitů. Jednotlivá napájení jsou blokována kondenzátory C9 a C10. Reset mikroprocesoru je aktivní v logické 0, proto je v obvodu zapojený odpor R2, který má funkci PULL UP, to znamená, že drží v provozním stavu na resetovacím pinu logickou 1 a připojením jumperu je úroveň změněna na logickou 0. Zbylé piny mikroprocesoru jsou vyvedeny na kolíkové lišty pro možnost připojení dalším periferních zařízení např. senzorů.

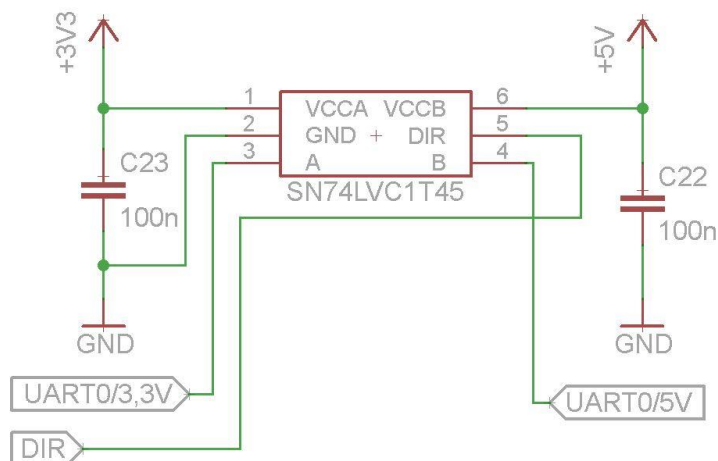
7.3 ZAPOJENÍ SBĚRNIC

Použitý mikroprocesor obsahuje dvě nezávislá rozhraní USART. Sběrnice s indexem 0 má to omezení, že jsou signály RXD a TXD společné pro účely programování rozhraním ISP, což může v některých aplikacích způsobit problémy. V továrním nastavení je možné USART1 používat pouze v synchronním režimu, protože mikroprocesor ATmega128A má volitelnou funkci kompatibility s mikroprocesorem ATmega103. Po vypnutí režimu kompatibility je možné využívat i toto rozhraní v plném funkčním rozsahu a naprosto nezávisle na druhém USARTU. Protože většina periferních zařízení, které komunikují s mikroprocesorem, pracuje s logickými úrovněmi 3,3 V, bylo toto napájecí napětí zvoleno jako ústřední. Problém ovšem přináší použité servopohony, se kterými se komunikuje prostřednictvím sběrnice UART s 5 V TTL logikou. Z tohoto důvodu byl použit integrovaný obvod SN74LVC1T45, který je podrobně popsán výše. Komunikace se servopohony je poloduplexní, proto stačí pouze jednokanálový převod logické úrovně, který zabezpečuje tento obvod, k jehož zapojení stačí pouze dva externí kondenzátory C12, C13 a signál řídicí směr komunikace s označením DIR. Všechny servopohony jsou připojeny do společné sběrnice pomocí standardního konektoru, který má 3 piny, a to samotnou sběrnici, kladné napájení 11,1 V z akumulátoru a samozřejmě zem. Tyto konektory jsou na plošném spoji celkem 4 pro samostatné připojení nohou robota. To znamená, že jsou na jeden konektor připojeny 3 servopohony, jejichž napájení je filtrováno přes elektrolytický kondenzátor o kapacitě 100 μ F a blokováno kondenzátorem 100 nF. Komunikace se senzorem LSM330DLC je zprostředkována

sběrnici TWI, která se tvoří signály SDA a SCL. Pokud je sběrnice neaktivní, tak je úroveň na těchto signálech v logické 1, což zabezpečují PULL UP odpory. Tyto odpory, které zakončují sběrnici, není nutné připojovat jako externí součástky, ale je možné využít integrovaných PULL UP odporů, které jsou součástí vstupně/výstupních obvodů mikroprocesoru a jejich aktivace je softwarová. Komunikace s Wi-Fi modulem je řešena prostřednictvím rozhraní UART s logickými úrovněmi 3,3 V pomocí signálů RXD a TXD. Přenosová rychlost komunikace je 9600 baud, proto postačují pouze datové vodiče a není potřeba hardwarová synchronizace pomocí signálů CTS a RTS.



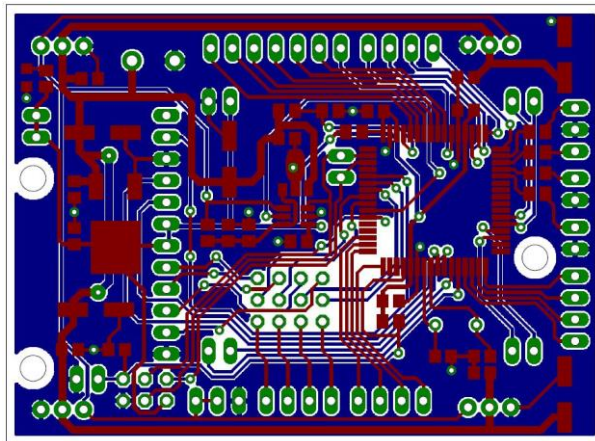
Obrázek 7.4 – Schéma zapojení konektorů servopohonů



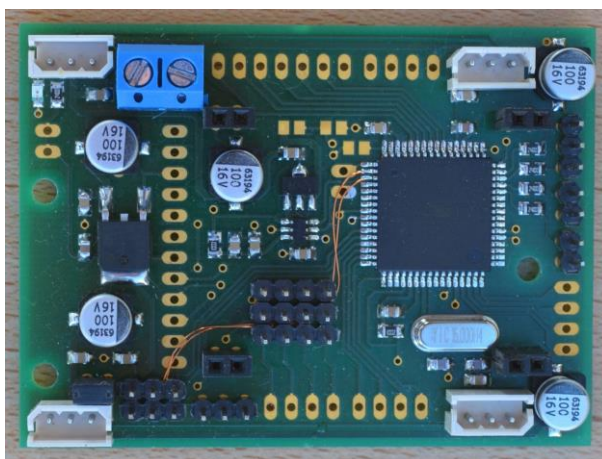
Obrázek 7.5 – Schéma zapojení převodníku úrovní SN74LVC1T45

7.4 NÁVRH PLOŠNÉHO SPOJE A OSAZENÝ PLOŠNÝ SPOJ

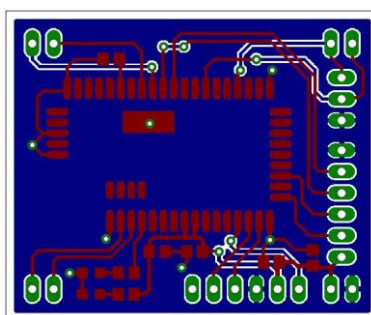
Pro návrh plošného spoje byl použit software Eagle, jehož funkce jsou dostatečné a umožňují generovat výstupní data, která plně popisují plošný spoj při samotné výrobě. Důležitá je funkce automatické kontroly DRC, která dle nastavené výrobní technologie zkontroluje velikost izolačních mezer, vzdálenosti vrtaných otvorů a cest, či součástek apod. Plošný spoj byl vyroben se základní tloušťkou měděné fólie 18 μm , kde je nutné dodržet izolační mezery alespoň 0,12 mm. Stejná vzdálenost platí i pro vrtané otvory s jinými objekty. Při ručním osazování plošných spojů SMD součástkami je lepší rovný povrch plošek, proto byla zvolena povrchová úprava chemickým zlatem, které je v tomto ohledu lepší než povrchová úprava bezolovnatým cínem, často nazývaná zkratkou HAL. Použitá součástková řada je SMD 0805, která je vhodná pro ruční pájení s použitím tavidla. Cesty pro signály byly kresleny tloušťkou čáry 0,3 mm, napájení integrovaných obvodů 0,4 mm a připojení napájení pro servopohony 0,8 mm, kvůli celkovému odběru servopohonů, který dosahuje až 5 A při pohybu robota. Plošné spoje jsou dvouvrstvé s rozlitou zemí ve spodní vrstvě. Řídící jednotka má rozměry 70×52 mm, druhá část nesoucí komunikační modul má rozměry 44×36 mm.



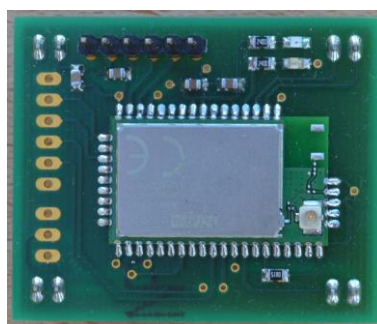
Obrázek 7.6 – Návrh plošného spoje řídicí jednotky robota



Obrázek 7.7 – Osazený plošný spoj řídicí jednotky robota



Obrázek 7.8 – Návrh plošného spoje komunikační jednotky robota



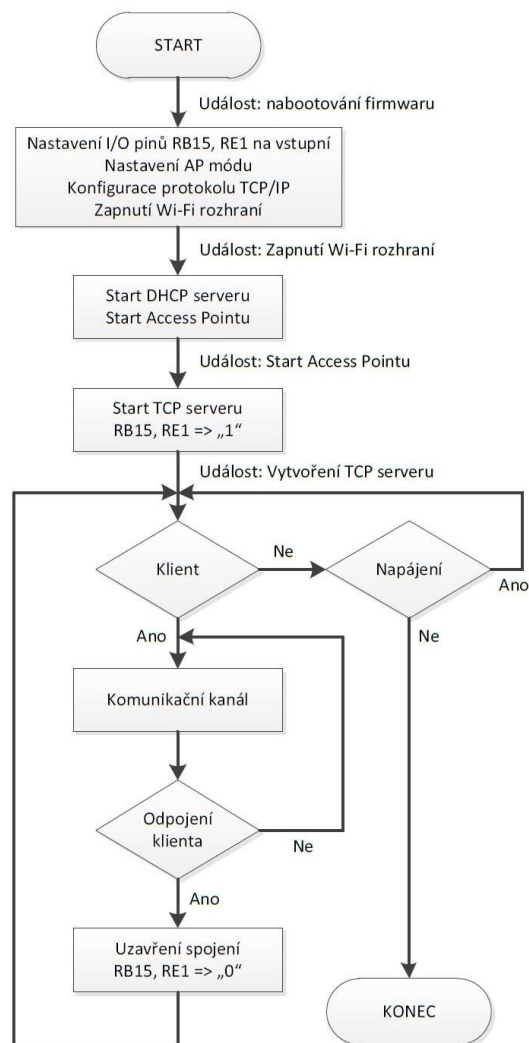
Obrázek 7.9 – Osazený plošný spoj komunikační jednotky robota

8 SOFTWAREVÉ ŘEŠENÍ ROBOTA

8.1 WI-FI MODUL

Dříve byly popsány možnosti uživatelského programování Wi-Fi modulu bez účasti hostitelského zařízení, které jsou v aplikaci robota výhodně využity. Přenos povelů a diagnostických dat mezi robotem a ovládacím softwarem probíhá prostřednictvím TCP komunikace. Z hlediska topologie Wi-Fi je robot Access point s TCP serverem, ke

kterému se připojuje Wi-Fi adaptér a TCP klient ovládacího softwaru. Komunikace mezi modulem a mikroprocesorem řídicí jednotky robota je zprostředkována sběrnici UART s přenosovou rychlostí 9600 baud a jedním vstupně/výstupním pinem, který indikuje, zda je připojen TCP klient, tedy ovládací software.



Obrázek 8.1 – Vývojový diagram firmwaru Wi-Fi modulu

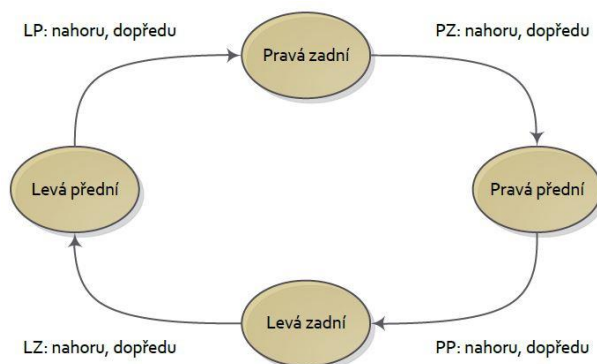
Algoritmus naprogramovaný v BGScriptu zajistí bez účasti mikroprocesoru řídicí jednotky vytvoření komunikační cesty. Po připojení napájení Wi-Fi modulu je automaticky vytvořen Access Point a DHCP server, který má za úkol přidělení IP adresy klientskému zařízení a TCP server čekající na připojení klienta. Při připojení ovládacího softwaru, tedy TCP klienta, dochází ke změně logické úrovně na pinu RB15, který signalizuje aktivní ovládání robota, tedy vytvoření datového kanálu pro obousměrný přenos informací. Aktivace ovládání je také signalizovaná červenou LED, která je řízena pinem RE1. Po odhlášení klientského softwaru nastává změna logické úrovně na pinech

RB15 a RE1 na logickou nulu. Tato signalizace umožňuje robotovi přecházet mezi výchozími polohami, pro chůzi při připojení ovládacího softwaru a pro polohu, do které se robot nastaví při připojení napájení nebo odpojení ovládacího softwaru. S využitím BGScriptu je Wi-Fi modul naprogramován jako samostatný subsystém, který je naprosto nezávislý na funkci hostitelského mikroprocesoru.

8.2 ŘÍDÍCÍ JEDNOTKA ROBOTA

Řídící jednotka robota je programována pomocí jazyka C ve vývojovém prostředí Atmel Studio 6. Hlavním úkolem je implementace sekvencí pohybů nohou, které jsou spouštěny na základě povelů přijatých z ovládacího softwaru. Za účelem přehledného programování pohybových sekvencí byla vytvořena knihovna implementující komunikaci se servopohony a knihovna obsahující algoritmy pohybových sekvencí pro pohyb v určeném směru. Knihovna realizující komunikaci se servopohony využívá dříve popsaného příkazu SYNC_WRITE, který umožňuje pomocí broadcastové adresy ovládat více servopohonů jedním datovým rámcem. V jednom okamžiku je potřeba ovládat maximálně čtyři servopohony, proto jsou implementovány procedury, které sestaví komunikační rámec pro změnu rychlosti a polohy jednoho, dvou, tří nebo čtyř servopohonů, protože pouze v některých případech dochází k ovládní více servopohonů v jednom okamžiku. Rozdělením procedur na počet ovládaných zařízení dochází k optimalizaci komunikace na sběrnici, protože je vyvolána pouze s těmi servopohony, u kterých je potřeba změnit úhel natočení. Komunikační rychlost na sběrnici je nastavena na 1 000 000 baud. Pohybové sekvence jsou naprogramovány dle popisu uvedeného v kapitole této práce, zabývající se problematikou algoritmu sekvencí naučených pohybů. Samotná implementace pohybu do příslušného směru je realizována tak, že dojde k vygenerování komunikačního rámce, který se odešle přes rozhraní UART do příslušných servopohonů, určených pomocí adres. Po přijetí a zpracování požadavků v servopohonech, dochází k jejich vykonání, to znamená zahájení změny úhlu natočení na příslušnou pozici. Tento pohyb má nějakou konstantní dobu vykonávání, po kterou musí mikroprocesor řídicí jednotky čekat, než dojde k odeslání dalšího požadavku na změnu pozice stejných servopohonů, aby mohl být předchozí pohyb vykonán do požadované pozice, protože servopohon vždy zpracovává aktuální požadavek. Zpoždění pro odesílání dalších příkazů po dokončení změny požadované příkazem předešlým bylo určeno experimentálně, protože vztah mezi rychlostí pohybu a časem jeho vykonávání je

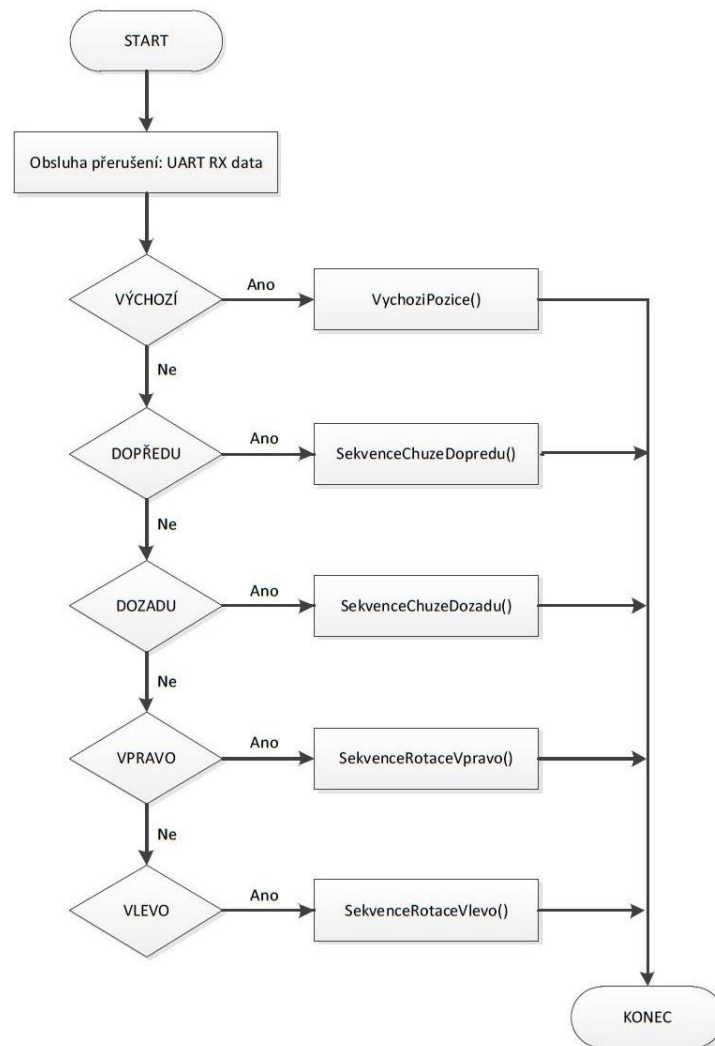
přibližně lineární. Tento lineární vztah znamená, že když rychlost pohybu zvýšíme x -krát, tak doba jeho vykonání bude x -krát menší. Požadovaná zpoždění byla určena experimentálně, kde byl algoritmus chůze testován při nejpomalejší rychlosti servopohonů, což umožnilo zjištění požadovaných časových prodlev. Při zvýšení rychlosti pohybu byl použit výše popsáný lineární vztah, kterým byl umožněn výpočet časů pro příslušnou rychlost pohybu robota. Rychlost chůze je možné nastavovat ve třech stupních, maximální, střední a nejpomalejší rychlost, při které nedochází k nestabilitě těla robota, způsobené převážením vlastní vahou. Pro všechny stupně rychlosti jsou implementovány samostatné procedury, vykonávající sekvenci pohybů do příslušného směru.



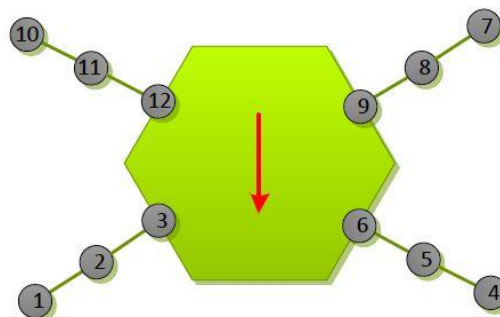
Obrázek 8.2 – Stavový diagram chůze dopředu a dozadu

Algoritmus chůze je realizován jako stavový automat, kde je v současném stavu možné přejít pouze do předem definovaného následujícího stavu, což znamená, že po vykonání pohybu jednou nohou je přesně definována následující noha, která vykoná totožný pohyb, pouze s časově posunutým počátkem. Po připojení napájení dochází k inicializaci vstupně/výstupních pinů, nastavení sběrnic, časovačů a nastavení polohy nohou do výchozí pozice, ze které robot zvedá tělo z podložky. Poté algoritmus čeká na připojení ovládacího softwaru, při kterém dochází ke změně logické úrovně na komunikačním pinu s Wi-Fi modulem, který signalizuje událost připojení klienta. Při aktivaci ovládání dochází k nastavení výchozí polohy nohou, ze které začíná pohyb jakýmkoli směrem. Povel, které určují rychlost a směr pohybu, jsou přijímány z Wi-Fi modulu prostřednictvím sběrnice UART. Přijetím dat je generováno přerušení, v jehož obsluze dochází k testování, zda se jedná o povel rychlosti nebo směru pohybu. Pokud se jedná o nastavení rychlosti, dochází pouze k přepsání hodnoty proměnné, určující rychlost pohybu, který se vykoná až po přijetí povelu nesoucího směr. Samotné servopohony umožňují čtení teploty a napájecího napětí. Tyto informace jsou

prostřednictvím Wi-Fi modulu přenášeny do ovládacího softwaru, který umožňuje jejich vizualizaci. K měření těchto veličin dochází po vykonání pohybu příslušným směrem, aby nedocházelo ke špatnému časování pohybové sekvence.



Obrázek 8.3 – Vývojový diagram hlavní smyčky programu



Obrázek 8.4 – Umístění servopohonů dle adres

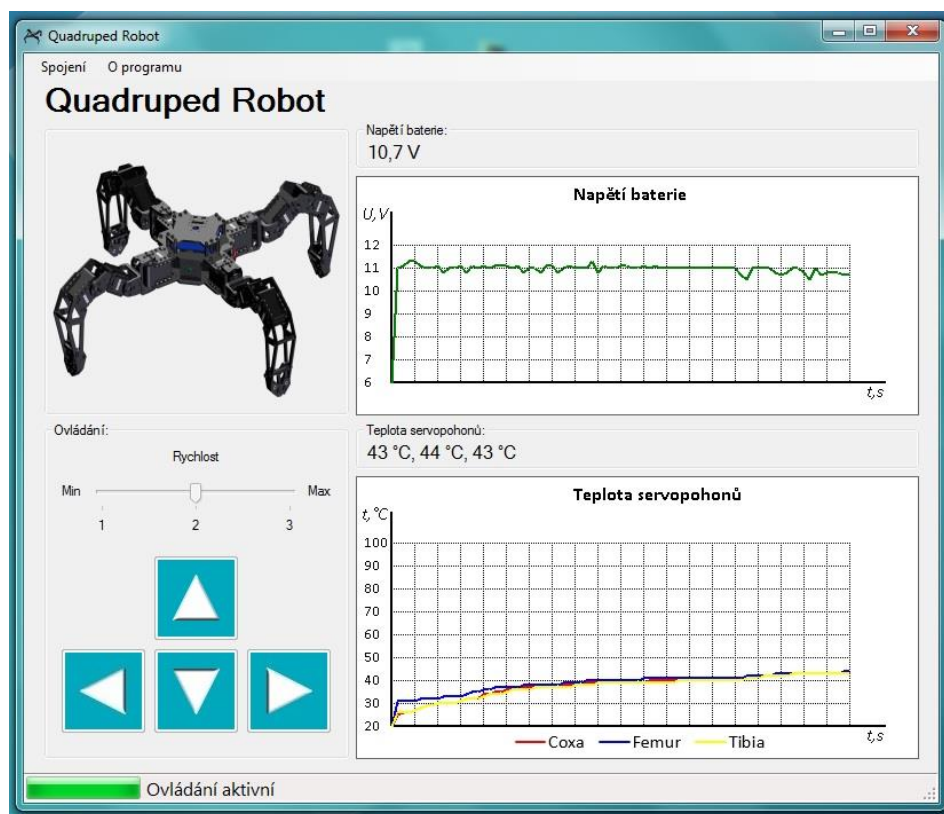
8.3 OVLÁDACÍ A VIZUALIZAČNÍ SOFTWARE

Software pro ovládání pohybu robota a zobrazování diagnostických dat byl naprogramován jazykem C# ve vývojovém prostředí Microsoft Visual Studio 2012. Jak už bylo řečeno dříve, komunikace je po fyzické vrstvě zprostředkována pomocí Wi-Fi adaptéru počítače a Wi-Fi modulu robota. Samotný přenos dat je zabezpečen TCP protokolem transportní vrstvy na portu číslo 2014. Ovládací software je v topologii počítačové sítě v roli TCP klienta. Bezdrátová komunikace je plně duplexní, proto musí tento TCP klient pracovat asynchronně, což znamená, že může vysílat i přijímat data v jednom okamžiku. Programovací jazyk C# má pro tyto účely předdefinované funkce, které se nachází v knihovně System.Net a System.Net.Sockets. Pro jednoduchou obsluhu má ovládací software funkci automatického připojení k Wi-Fi síti definované pomocí SSID. Po stisknutí tlačítka vyhledávání v kartě spojení, dochází ke skenování všech dostupných Wi-Fi sítí každé 2 s. Pokud je v seznamu dostupných sítí přítomné SSID Quadruped Robot, dochází k automatickému připojení. Po úspěšném připojení k Wi-Fi síti dochází k připojení TCP klienta k TCP serveru Wi-Fi modulu robota. Pro obsluhu Wi-Fi adaptéru počítače je využita knihovna s názvem NativeWifi, která obsahuje potřebné funkce pro interakci mezi ovládacím softwarem a hardwarem adaptéru. Po připojení TCP klienta k serveru Wi-Fi modulu robota dochází k odesílání diagnostických dat do ovládacího softwaru. Na straně TCP klienta dochází k přijímání 4 bytů. První tři byty obsahují teplotu servopohonů z jednotlivých částí konstrukce nohou robota. Poslední byte reprezentuje napájecí napětí. Diagnostická data jsou vykreslována do grafů. Vykreslování do grafů probíhá v reálném čase, to znamená hned po jejich přijetí a zpracování, což je pro většinu knihoven v jazyce C# problém. Z tohoto důvodu byly grafy vytvořeny pomocí funkcí počítačové grafiky, které umožňují kreslení bitmapových obrázků. Každý graf je tedy bitmapou, do které je postupně přikreslován průběh měřených veličin. Za tímto účelem byla implementována funkce pro vytvoření grafu, která vykreslí osy, mřížku, nadpis a popisy os. K zobrazování naměřených hodnot slouží funkce vykreslující přímky na dané souřadnice, dle rozsahu os grafu. Ovládání směru pohybu robota je možné ve čtyřech směrech, pomocí šipek na klávesnici počítače. Pro odchyťávání stisknutých kláves slouží události jazyka C# s názvem Key.Down a Key.Up. Při stisknutí příslušné směrové klávesy dochází k odeslání hodnoty povelu, která spouští v řídicí jednotce robota proceduru, která zabezpečuje pohyb ve vybraném směru. Příslušné sekvence pohybů jsou vykonávány po celou dobu stisku směrové klávesy. Po

jejím uvolnění, dochází k odeslání hodnoty povelu, který ukončí probíhající sekvenci pohybu a uvede nohy robota do výchozí pozice pro chůzi či rotaci. Nastavování rychlosti pohybu robota probíhá stejným způsobem. Pro snížení rychlosti je použita klávesa Page Down, ke zvýšení klávesa Page Up. Hodnoty reprezentující implementované povelu jsou uvedeny v následující tabulce.

Tabulka 8.1 – Tabulka hodnot povelů ovládajících pohyb robota

<i>Klávesa</i>	<i>Povel</i>	<i>Hodnota</i>
Uvolnění směrové klávesy	Výchozí poloha	0xAA
Page Up <=> Page Down	Rychlost „pomalá“	0xA1
Page Up <=> Page Down	Rychlost „střední“	0xA2
Page Up <=> Page Down	Rychlost „nejrychlejší“	0xA3
Nahoru	Pohyb dopředu	0x08
Dolů	Pohyb dozadu	0x02
Vpravo	Rotace vpravo	0x06
Vlevo	Rotace vlevo	0x04



Obrázek 8.5 – Ovládací a vizualizační software

ZÁVĚR

Tato práce se zabývala problematikou řídicích algoritmů chodících robotů, servopohonů a komunikačních modulů. Předpokladem pro chůzi robota byla rovná podložka. Za tohoto předpokladu se podařilo implementovat funkční algoritmus založený na sekvencích naučených pohybů, který umožňuje pohyb robota v libovolném směru.

Řídicí jednotka byla navržena pro možnost připojení akcelerometru, který by umožňoval detekci převrácení těla robota a neznemožnil by jeho pohyb v této situaci. Z důvodu zjednodušení byla zvolena vhodnější varianta konstrukce, která chůzi v opačné orientaci neumožňuje. Akcelerometr by v této situaci neměl své opodstatnění, a proto nebyl osazen. Nicméně je řídicí jednotka navržena univerzálně, což neznemožňuje další inovace, které jsou možné zejména v doplnění senzorů, které by v kombinaci s aplikací problému inverzní kinematiky umožňovaly pohyb robota přes překážky.

Klíčovým problémem při implementaci řídicího algoritmu bylo spolehlivé ovládání servopohonů. Pro tento účel byly vytvořeny procedury, které na základě parametrů adresy, rychlosti a požadované pozice sestaví komunikační rámec protokolu servopohonů společnosti Dynamixel. Aplikaci samotného algoritmu chůze značně zjednodušil lineární vztah mezi rychlostí a dobou nastavení servopohonu do požadované pozice. Rotace těla byla vyzorována z videonahrávky pohybu obdobné konstrukce robota a provedení její implementace bylo experimentální.

Původním záměrem bylo měření napájecího napětí a teploty servopohonů s konstantní vzorkovací periodou, která měla být řešena pomocí přerušení od časovače mikroprocesoru. V tomto řešení nastal problém synchronizace nohou při chůzi, protože spuštění této sekvence má náhodný charakter a měření zmiňovaných veličin nemá konstantní dobu získání informace. Z tohoto důvodu docházelo ke vzniku nepřijatelného zpoždění, které znemožnilo vykonávat pohyb nohou dle určeného časového diagramu. Tento problém byl vyřešen měřením veličin pouze po skončení sekvence pohybů. Uživatelský ovládací software umožňuje intuitivní a pohodlné ovládání a čtení diagnostických dat, vykreslovaných do přehledných grafů.

LITERATURA

- BLE121RL Bluetooth Smart Long Range Module, Bluegiga [online]. [cit. 2015-03-24].
Dostupné z: https://bluegiga.studio.crasman.fi/file/dl/c=product_big/RqWTVQ/g30L18kUHB0aw4_K73S18Q/BLE121LR_L_RGB.jpg
- WF121 Wi-Fi Module, Bluegiga [online]. [cit. 2015-03-24].
Dostupné z: https://bluegiga.studio.crasman.fi/file/dl/c=product_big/Oq7aHw/aS_Hxne6Pm7CEtRFqf7ww/WF121_M_RGB_front.jpg
- BLUEGIGA TECHNOLOGIES. Bluegiga WF121 Api Reference 1.7. Espoo, 2014.
- BLUEGIGA TECHNOLOGIES. Bluegiga WF121 Datasheet 1.5.2. Espoo, 2014.
- CVEJN, Jan. Průmyslové roboty. Pardubice, 2014.
- ČERNOHLÁVEK, Petr. Elektronická zabezpečovací a monitorovací jednotka. Pardubice, 2013. Bakalářská práce. Univerzita Pardubice.
- Mikrokontroler ATmega128, Dunia Elektronika [online]. [cit. 2015-03-24]. Dostupné z: <http://duniaelektronika.net/wp-content/uploads/2013/06/ATmega128.jpg>
- XBee PRO 802.15.4, GraviTech [online]. [cit. 2015-03-24]. Dostupné z: http://ep.yimg.com/ca/I/yhst-27389313707334_2252_64098223
- LIANG, Oscar. Inverse Kinematics for Hexapod and Quadruped Robots. In: OscarLiang.net [online]. 2012 [cit. 2015-03-24].
Dostupné z: <http://blog.oscarliang.net/inverse-kinematics-implementation-hexapod-robots/>
- LIANG, Oscar. Quadruped Robot Gait Study. In: OscarLiang.net [online]. 2013 [cit. 2015-03-24]. Dostupné z: <http://blog.oscarliang.net/quadruped-robot-gait-study/>
- Dynamixel AX-12A Servo, Mbed [online]. [cit. 2015-03-24]. Dostupné z: <http://developer.mbed.org/media/uploads/chris/310x165xax12pinout.png.pagespeed.ic.Fjnp1I49SM.png>
- MOUSER ELECTRONICS. Steval-MKI122V1. 2013. Dostupné z: <http://www.mouser.com/ds/2/389/DM00052733-492061.pdf>
- Volba pohonu, Offroad RC. [online]. [cit. 2015-03-24]. Dostupné z: http://www.offroad-rc.info/image/zaklady_valasek/emotor.jpg
- Princip – Encoder, Pohonná technika [online]. [cit. 2015-03-24]. Dostupné z: <http://www.pohonnatechnika.cz/shared/upload/editor/skola%20encoder%204.jpg>
- Optoelektronický senzor s binárním kódem, Senzory [online]. [cit. 2015-03-24].
Dostupné z: http://senzory.aspone.cz/obrazky_v_textu/OIS_2.png
- Speciální krokové motory na míru, Servo Drive. [online]. [cit. 2015-03-24].
Dostupné z: http://www.servo-drive.com/pictures/custom_steppers/two_phase_hybrid_step_per_motor_construction_cs.jpg
- SKALICKÝ, Jiří. Elektrické servopohony. In: SKALICKÝ, Jiří. Elektrické servopohony [online]. 2. vyd. Brno: Vysoké učení technické, 2001 [cit. 2015-03-24]. ISBN 80-214-1978-4. Dostupné z: http://www.vutbr.cz/www_base/priloha.php?dpid=33400
- Akční členy s EC motory, TechPark. [online]. [cit. 2015-03-24].
Dostupné z: http://www.techpark.sk/application_data/techpark/uploads/Image/technika_9_2009/8.jpg

TEXAS INSTRUMENTS. SN74LVC1T45 Datasheet. Dallas, Texas, 2004. Dostupné z:
<http://www.mouser.com/ds/2/405/sces515k-521641.pdf>

Dynamixel AX-12A, Trossen Robotics [online]. [cit. 2015-03-24]. Dostupné z:
<http://www.trossenrobotics.com/dynamixel-ax-12-robot-actuator.aspx>

PhantomX Quadraped Assembly Guide, Trossen Robotics [online]. [cit. 2015-03-24].
Dostupné z: <http://learn.trossenrobotics.com/images/assembly/phantomXHexapod/legdiagram.jpg>

Přílohy

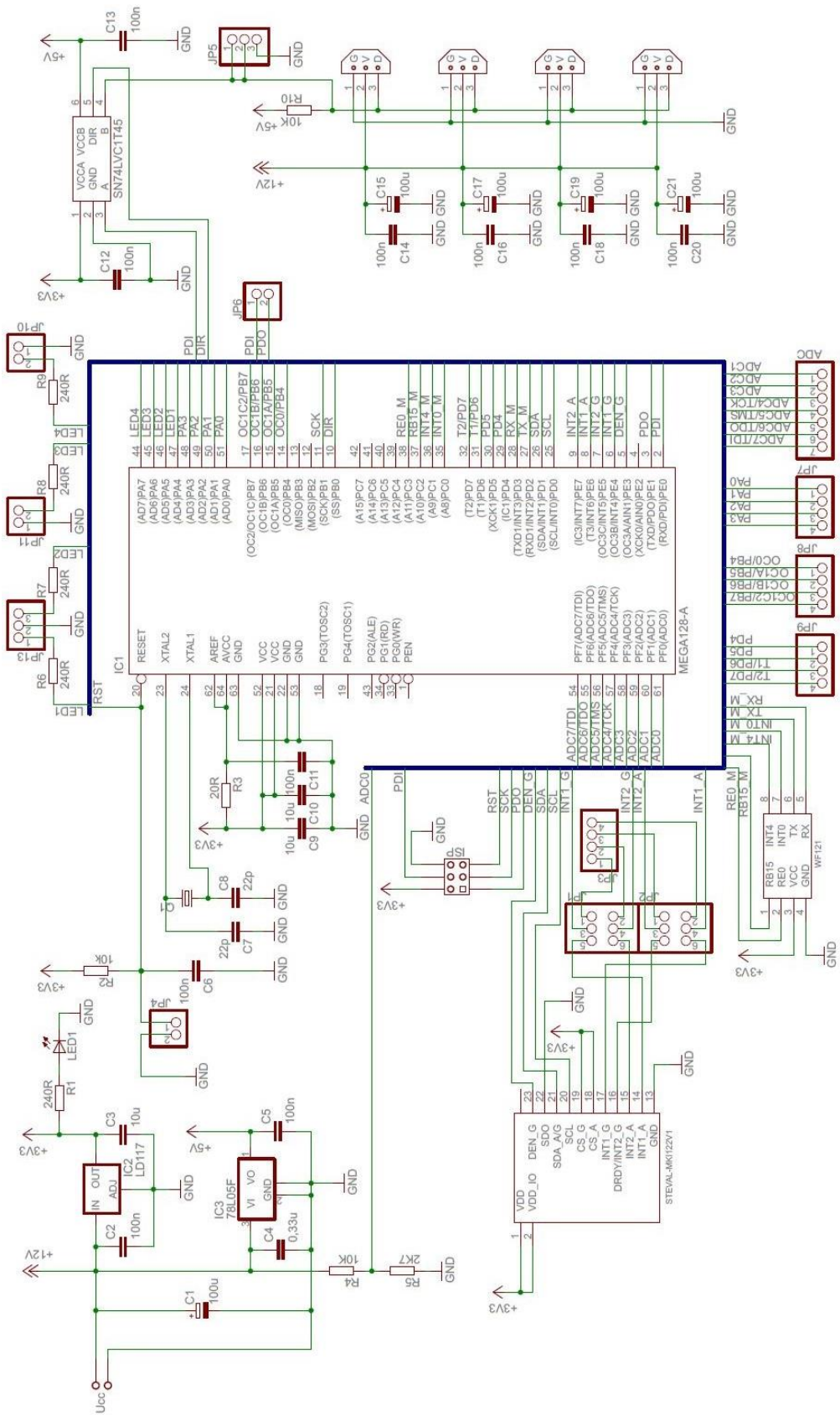
Příloha A: Schémata zapojení.....	A – 1
Příloha B: Návrhy plošných spojů	B – 1
Příloha C: Mechanická konstrukce	C – 1

Příloha k diplomové práci

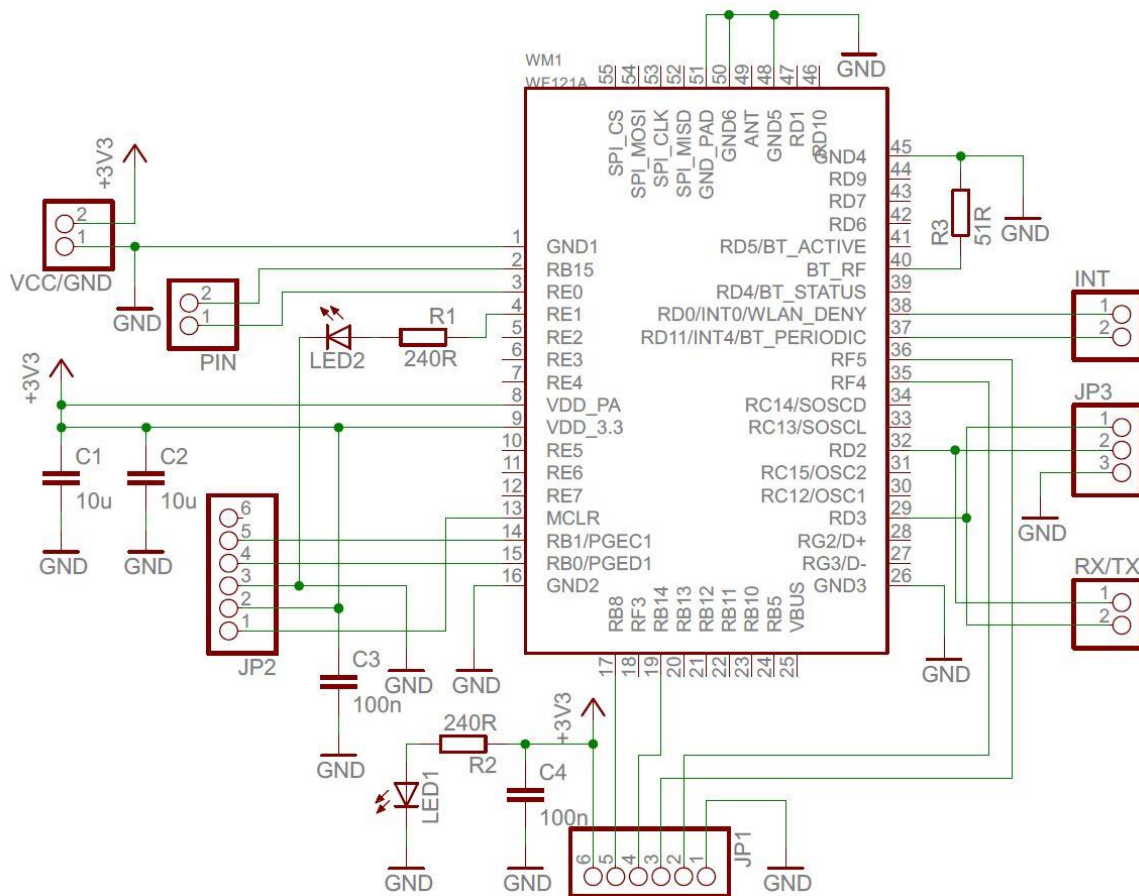
Čtyřnohý krácející robot

Bc. Petr Černohlávek

SCHÉMATA ZAPOJENÍ



Obrázek 1 – Schéma zapojení řídicí jednotky robota



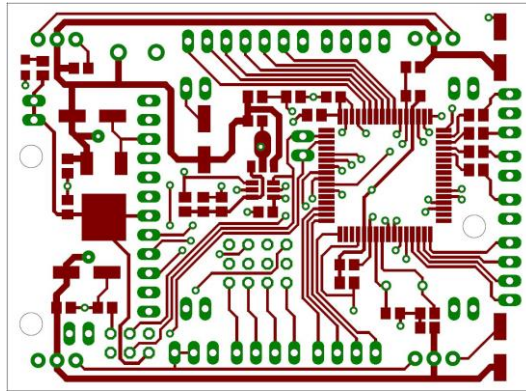
Obrázek 2 – Schéma zapojení komunikačního modulu

Příloha k diplomové práci

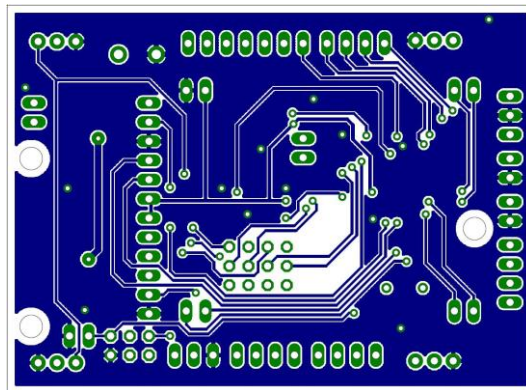
Čtyřnohý krácející robot

Bc. Petr Černohlávek

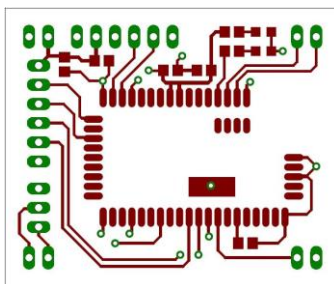
NÁVRHY PLOŠNÝCH SPOJŮ



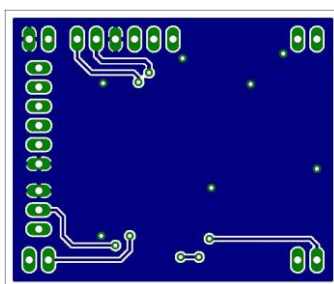
Obrázek 1 – Návrh plošného spoje řídicí jednotky robota (TOP)



Obrázek 2 – Návrh plošného spoje řídicí jednotky robota (BOTTOM)



Obrázek 3 – Návrh plošného spoje komunikační jednotky robota (TOP)



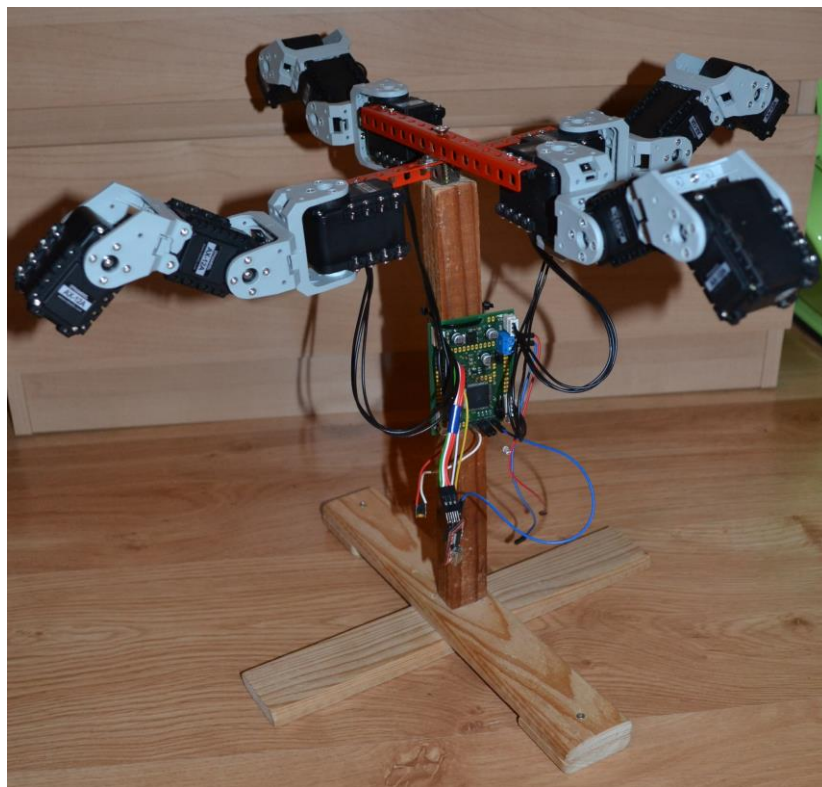
Obrázek 4 – Návrh plošného spoje komunikační jednotky robota (BOTTOM)

Příloha k diplomové práci

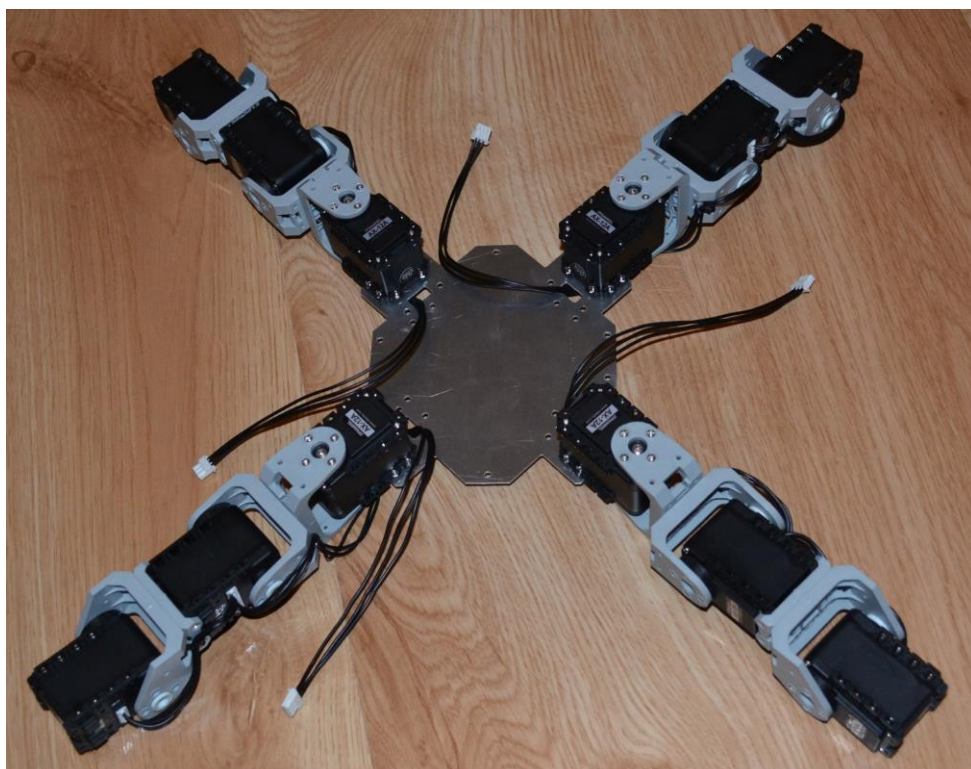
Čtyřnohý krácející robot

Bc. Petr Černohlávek

MECHANICKÁ KONSTRUKCE



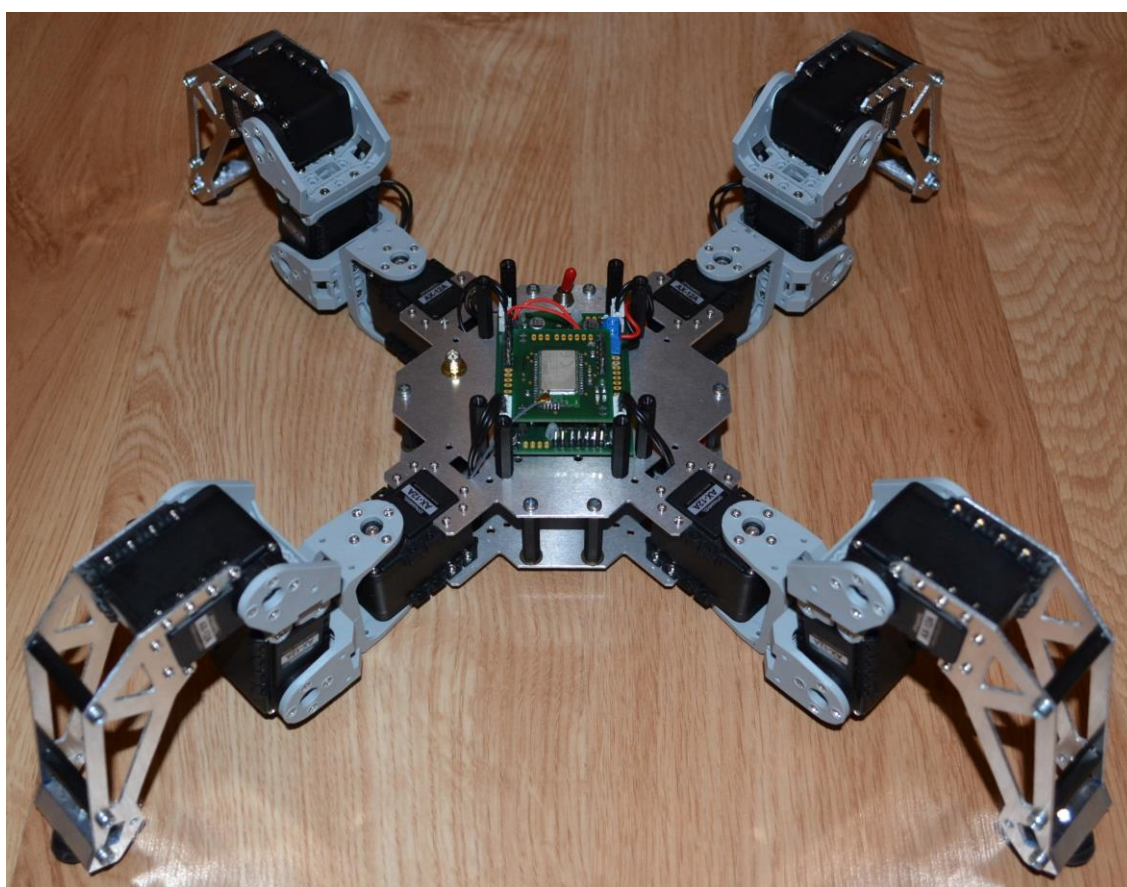
Obrázek 1 – Stojan k testovacím účelům servopohonů a sekvencí chůze



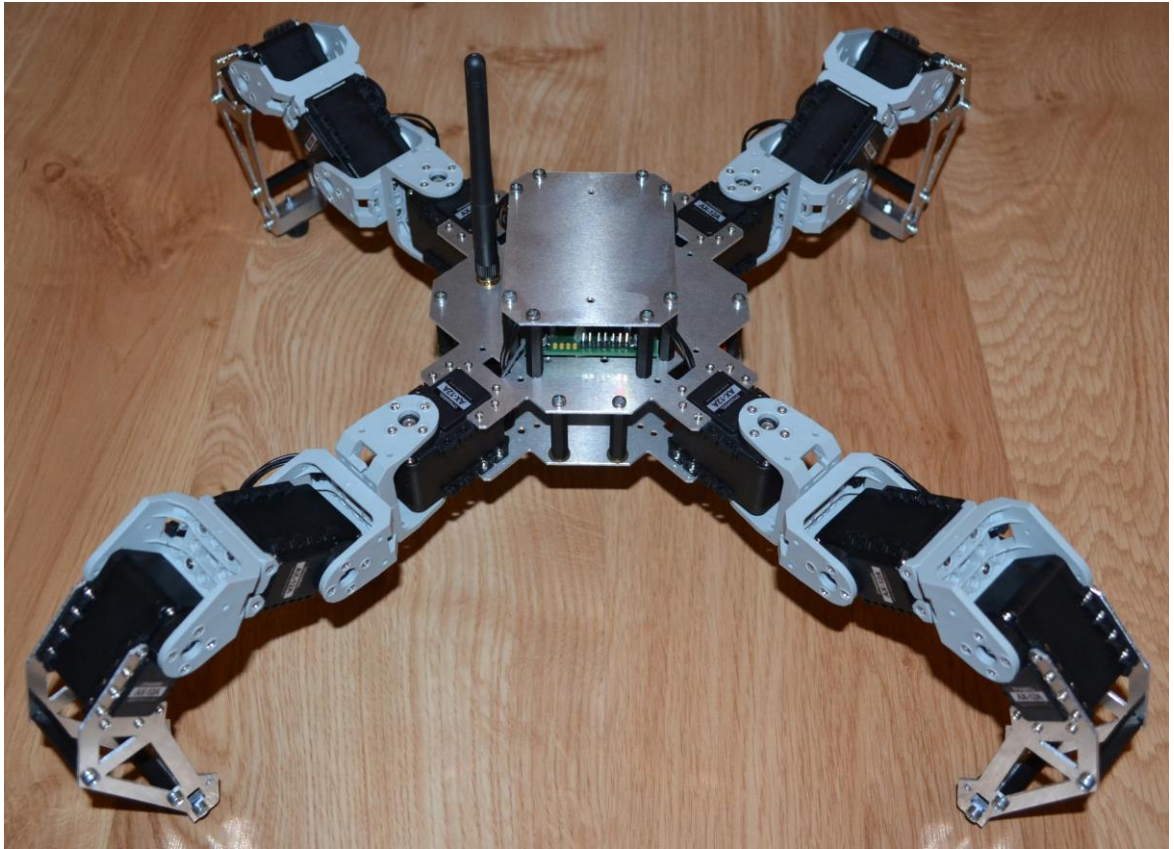
Obrázek 2 – Konstrukce robota: nohy a spodní část těla



Obrázek 3 – Konstrukce robota: uložení akumulátoru



Obrázek 4 – Konstrukce robota: uložení plošného spoje



Obrázek 5 – Kompletní konstrukce robota