

UNIVERZITA PALACKÉHO v OLOMOUCI

PŘÍRODOVĚDECKÁ FAKULTA

Společná laboratoř optiky

Bakalářská práce

Čtyřkolový dálkově řízený robot založený na STM32



Autor	Dominik Sobek
Vedoucí práce	RNDr. Jiří Pechoušek, Ph.D.
Studijní obor	Přístrojová fyzika
Forma studia	Prezenční
Rok	2014

Jméno a příjmení autora	Dominik Sobek
Název práce	Čtyřkolový dálkově řízený robot založený na STM32
Typ práce	Bakalářská
Pracoviště	Společná laboratoř optiky
Vedoucí práce	RNDr. Jiří Pechoušek, Ph.D.
Rok obhajoby práce	2014
Počet stran	52
Počet příloh	5
Jazyk	Český
Abstrakt	Bakalářská práce se zabývá návrhem a realizací čtyřkolového mobilního robota s diferenciálním řízením. Robot je řízen platformou STM32F4 Discovery. Pro komunikaci mezi obsluhou a robotem jsou použity bezdrátové XBee moduly. Chování robota je řešeno programově a je založeno na principu pohybu kolových vozidel. Práce obsahuje zpracovanou dokumentaci pro výrobu elektroniky umístěné na robotovi.
Klíčová slova	Robot, ATmega32, STM32F407 Discovery, joystick, XBee moduly, H-můstek, PWM, mikrokontrolér, odometrie

Author's first name and surname	Dominik Sobek
Title	Wheel remote controlled robot based on STM32
Type of thesis	Bachelor
Department	Joint laboratory of optics
Supervisor	RNDr. Jiří Pechoušek, Ph.D.
Year of presentation	2014
Number of pages	52
Number of appendices	5
Language	Czech
Abstract	<p>Focus of this Bachelor thesis is the design and realisation of a four-wheeled mobile robot with differential steering. The robot is controlled by the platform STM32F4 Discovery. The communication between an operator and the robot is mediated by wireless XBee modules. The behavior of the robot is solved programmatically and is based on the movement of wheeled vehicles. The thesis includes all the documents for the construction and a list of all electronic parts that have been used for completing the robot.</p>
Keywords	Robot, ATmega32A, STM32F407 Discovery, joystick, XBee modules, H-bridge, PWM , microcontroller, odometry

Prohlašuji, že jsem předloženou bakalářskou práci vypracoval samostatně pod vedením RNDr. Jiřího Pechouška, Ph.D., a že jsem použil zdrojů, které cituji a uvádím v seznamu použitých zdrojů.

V Olomouci dne

.....

podpis

Chtěl bych poděkovat RNDr. Jiřímu Pechouškovi, Ph.D., za vedení mé bakalářské práce, za cenné rady a připomínky.

Mé poděkování patří též Mgr. Michalu Dudkovi za věcné připomínky a vstřícnost při konzultacích při vypracování mé bakalářské práce.

Úvod

Slovo robot bylo poprvé použito v knize R.U.R. (Rassum Universal Robots, r. 1920) českého spisovatele Karla Čapka. V českém jazyce má slovo robot význam pracovníka nebo služebníka. Ke konkrétnější definici robota přistoupil americký robotický institut (1979): Robot je programovatelný, multifunkční manipulátor vyroben pro práci s materiálem, s jinými zařízeními, nebo je vyroben pro speciální druh činnosti [1]. Tato definice je jednou z mnoha dalších definic, které rozdělují roboty podle druhu činnosti, konstrukce, pohonu, mobility aj. kritérií. Po uplynutí více než 30 let vznikl v roce 1954 první průmyslový robot, vytvořený za účelem zjednodušení práce na výrobní lince. Ke zdokonalení robotů přispěl pokrok v elektronice, softwarovém inženýrství a ve výrobě nových materiálů. Od roku 1954 až do současnosti nahradili roboti některé druhy obtížných a nebezpečných prací. Roboti dnes již nejsou pouhým usnadněním práce jako v minulosti. Pronikli do dalších oborů lidské činnosti, například do zdravotnictví, kde robot dokáže rukou chirurga provést operační zákrok s obrovskou přesností. Pro armádní a civilní potřeby byli zkonstruováni roboti, kteří umí zneškodňovat trhavy a nástražné systémy. Velkým pokrokem prošel výzkum v polidštění robotů, jehož výsledkem je humanoidní robot Assimo. Rozvojem prochází také mobilní roboti. Mobilní roboty můžeme například rozdělit podle činnosti, pro kterou jsou určeni, nebo podle rychlosti, se kterou má být úkol splněn. Mobilní roboty dělíme mimo jiné na pozemní, vodní, létající a vesmírné [1]. Dále bych se chtěl zaměřit pouze na roboty pozemní, rozdělit je můžeme podle podvozku na pásové, kolové a kráčivé.

Každý mobilní robot se skládá z nosné konstrukce (šasi), na kterou se upevňují další periferní prvky. Další nezbytnou součástí je elektronika, která zajišťuje činnost senzorů a pohyb robota. Důležitou částí je zdroj napájení, který musí např. splňovat výkonové požadavky robota. Nejdůležitější částí je inteligence robota, které dosáhneme softwarově pomocí mikrokontroléru. Inteligence robota je mimo jiné také spojena s navigací robota. Dnes je možné robota ovládat vizuálně (kamery), pomocí senzorů (sonarů), inercializací (využití akcelerometrů a gyroskopů), odometrií nebo pomocí počítačích metod (s využitím enkodérů) aj.[1],[2].

Tato práce je zaměřena na konstrukci mobilního robota s využitím bezdrátové komunikace. K ovládní robota je použit joystick řízený obsluhou.

Obsah

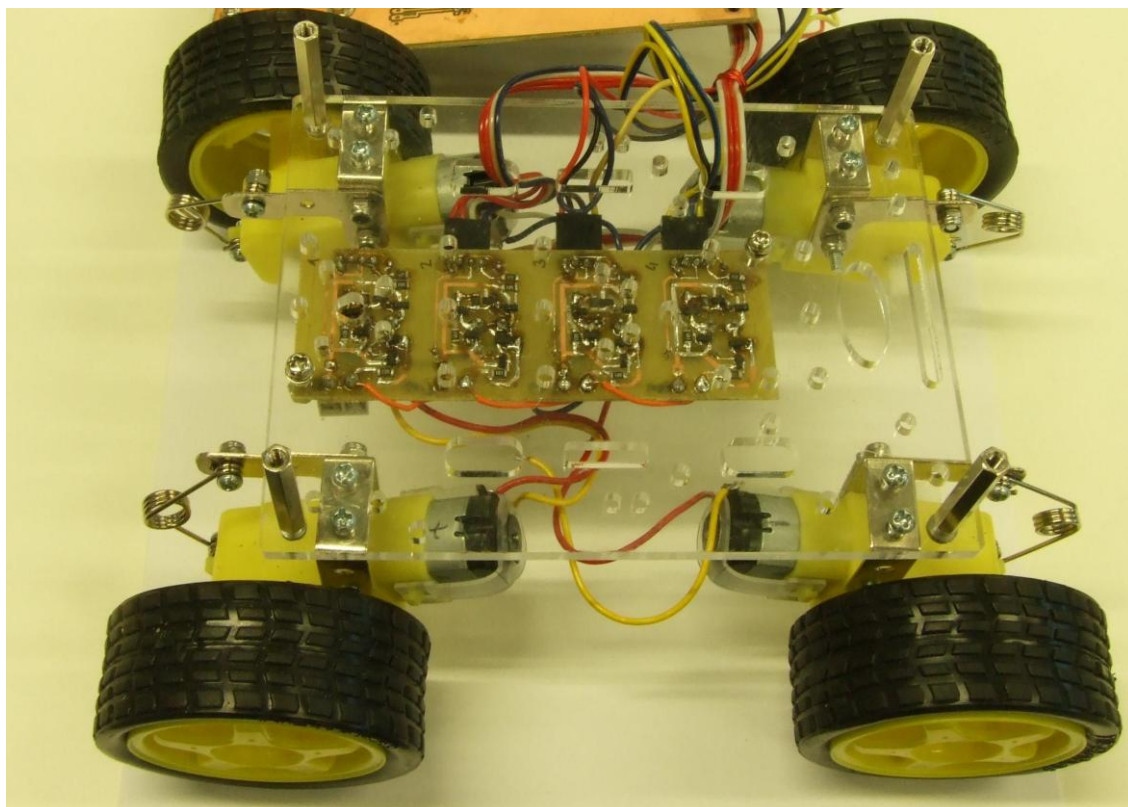
Úvod.....	5
1. Hardware pro výrobu dálkově řízeného robota.....	7
1.1 Šasi.....	7
1.2 Napětím řízené stejnosměrné motory	8
1.3 H-můstek.....	8
1.4 Akumulátory	10
1.5 Základní deska	12
1.6 XBee moduly	13
1.6.1 XBee Board.....	14
1.7 Mikrokontrolér STM32F407.....	15
1.7.1 Vnitřní schéma mikroprocesoru STM32F407VGT6	16
1.8 Joystick.....	18
1.9 Atmel AVR ATmega32A	18
1.10 Modul UM232.....	20
2. Programové vybavení robota	22
2.1 Přenos dat z joysticku do XBee modulu mikrokontrolérem ATmega32A	22
2.2 Konfigurace XBee modulů	23
2.2.1 Sériová komunikace	23
2.2.2 Připojení XBee modulu k počítači pomocí modulu UM232.....	24
2.2.3 Konfigurace XBee modulu pomocí programu X-CTU.....	25
2.3 Příjem dat z XBee modulu do mikrokontroléru STM32F407.....	28
2.3.1 Ochranný časovač	30
2.4 Řízení motorů pulsní šířkovou modulací	31
3. Řízení robota	33
3.1 Kinematika řízení.....	34
3.1.1 Určení souřadnicového systému.....	34
3.1.2 Kinematický model pro řízení robota.....	35
3.1.3 Živé řízení robota	39
Závěr	40
Seznam použitých zdrojů.....	41
Příloha A. Návrh DPS ATmega32A	43
Příloha B. Návrh DPS pro výrobu H-můstků.....	45
Příloha C. Návrh DPS pro výrobu XBee Boardu.....	47
Příloha D. Návrh DPS pro výrobu základní desky STM32F4 Discovery.....	49
Příloha E. Elektronická příloha	51

1. Hardware pro výrobu dálkově řízeného robota

V této kapitole jsou jednotlivě popsány základní části robota spolu s elektronickými schématy a návrhy plošných spojů. Spolu s fotodokumentací jsou zde uvedeny základní parametry použitých součástí. Některé části robota jsou zakoupené, jiné vyrobené. Všechny části jsou ale nezbytné pro chod a řízení robota. V úvodu kapitoly je popsána základní konstrukce robota, včetně základních desek a motorů. Dále je zde uvedena výroba H-můstek pro řízení stejnosměrných motorů, typ a zapojení baterie, popis použitých XBee modulů a mikrokontrolérů STM32F407 a AVR ATmega32A.

1.1 Šasi

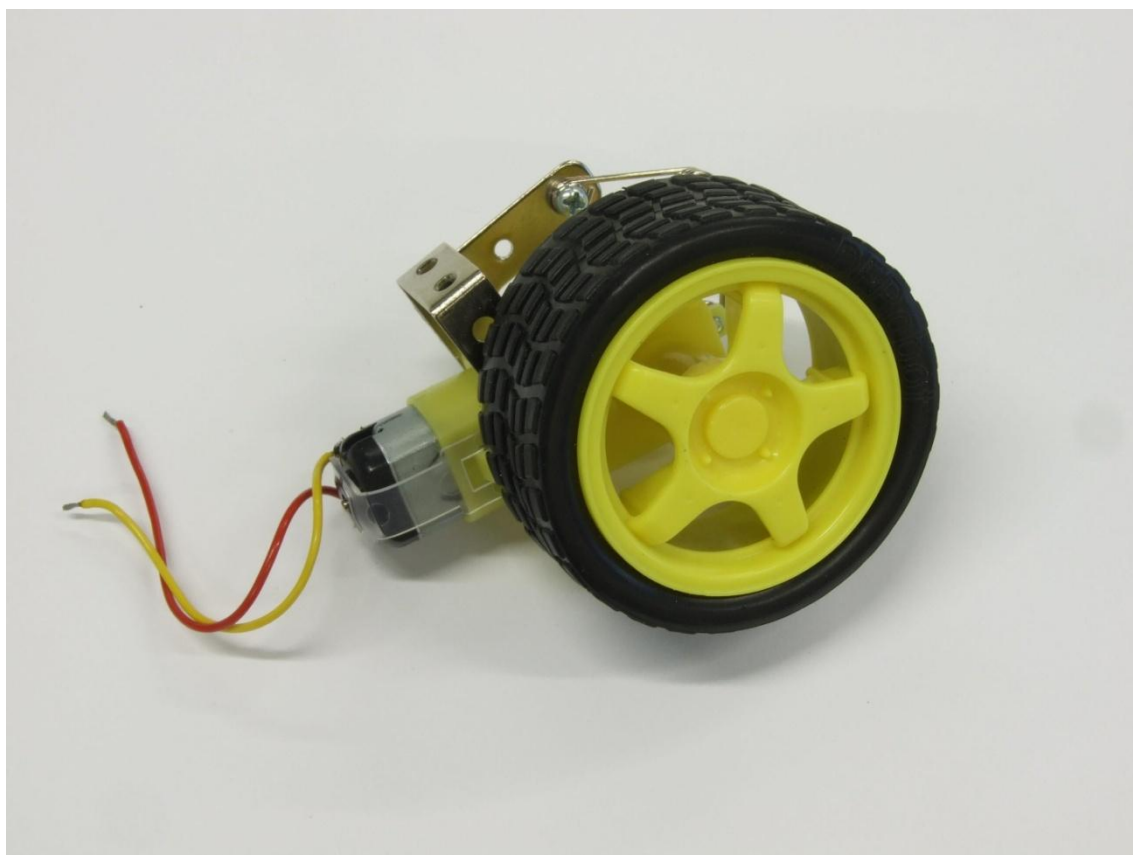
Šasi robota se skládá ze dvou hlavních částí. První částí je nosná konstrukce, kterou tvoří plastová základní deska. Druhou částí jsou motory s koly. Deska je opatřena předvrtanými otvory pro šrouby a vyfrézovanými otvory pro vodiče. Rozměry desky jsou 250 mm × 180 mm. Na obrázku 1.1 je šasi robota s plastovou základní deskou a koly.



Obrázek 1.1: Plastová základní deska s H-můstkou a čtyřmi koly.

1.2 Napětím řízené stejnosměrné motory

Pohon robota tvoří čtyři stejnosměrné napětím řízené motory, zapouzdřené do plastových krytů, ve kterých jsou převodové soustavy (viz obrázek 1.2). Z převodové soustavy je vyvedena hřídel, na které je nasazeno plastové kolo s gumou.

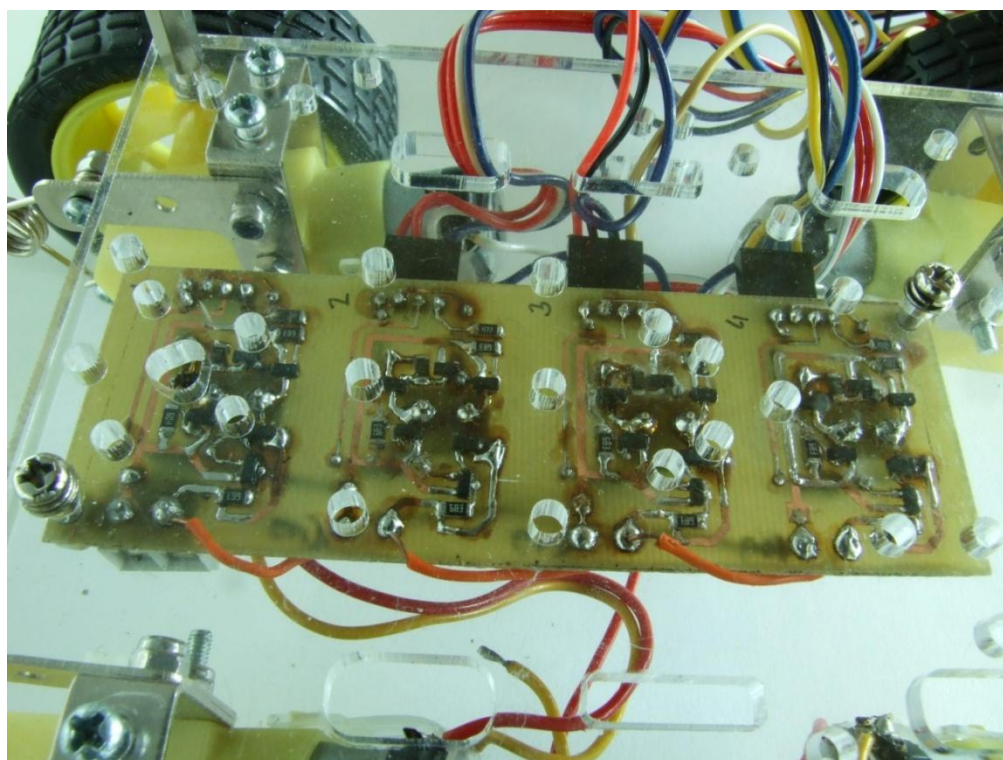


Obrázek 1.2: Stejnosměrný motor s kolem na hřídeli.

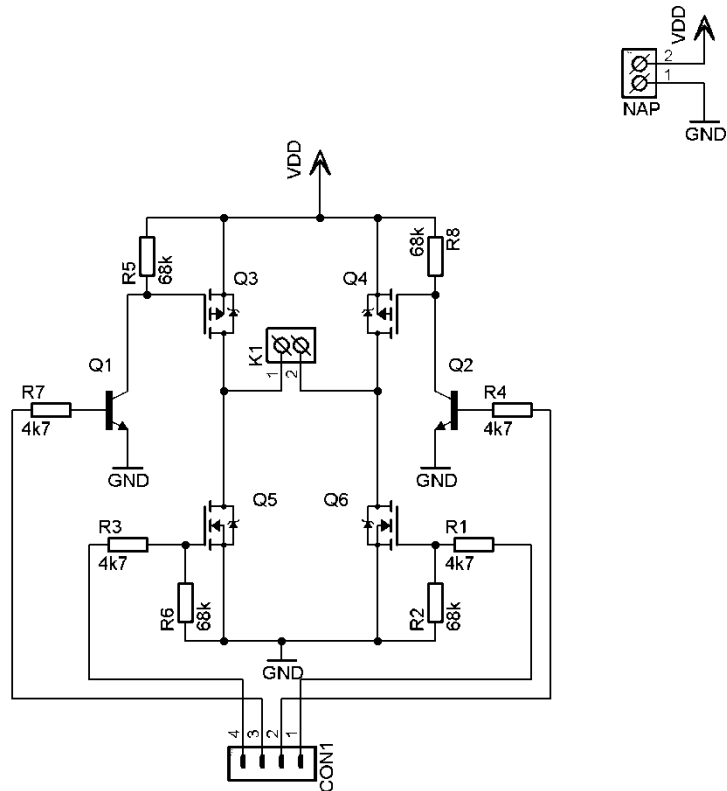
1.3 H-můstek

H-můstek je elektronický obvod, který dovolí přivést napětí obou polarit na zátěž [3]. H-můstky mohou být vyrobeny v podobě integrovaného obvodu (např. DRV8833), nebo sestaveny z diskrétních součástek. Tyto obvody se používají v robotice, pro řízení motorů, ale i v dalších oborech elektroniky. H-můstky jsou použity pro řízení stejnosměrných motorů a jsou napájeny z baterie. Motor je připojen na baterii s napětím $U = 12,6 \text{ V}$, tranzistory proto musí splňovat minimální napětí $U_{DS} = 15 \text{ V}$ a proud $I_{DS} = 1,5 \text{ A}$. Po pečlivém prozkoumání všech dostupných integrovaných obvodů se bohužel nepodařilo vybrat vhodný integrovaný obvod s H-můstkem, který by splňoval napěťové a proudové požadavky.

Z toho důvodu bylo rozhodnuto zkonstruovat H-můstky z diskretních součástek (viz obrázek 1.3). Pro konstrukci byly zvoleny tranzistory MOSFET s P a N vodivostí. Tranzistory Q3 a Q4 s vodivostí P jsou označeny jako IRLML6402. Tranzistory s vodivostí P pracují s maximálním napětím $U_{DS} = 20\text{ V}$ a proudem $I_{DS} = -2,2\text{ A}$. Tranzistory Q5, Q6 s vodivostí typu N jsou označeny jako IRLML2502. Pracují s maximálním napětím $U_{DS} = 20\text{ V}$ a proudem $I_{DS} = 3,4\text{ A}$. Bipolární tranzistory (BC817) Q1 a Q2 slouží k ovládání tranzistorů Q3 a Q4. Rezistory R5 a R8 slouží k zavírání tranzistorů Q3 a Q4. Analogicky fungují i rezistory R2 a R6 u tranzistorů Q5 a Q6. H-můstek může přivádět na svorky K1 napětí ve dvou směrech. Kladná polarita je přivedena na pin 1 a záporná polarita na pin 2. Otevřeny jsou tranzistory Q3 a Q6. Pro pohyb v opačném směru je přivedena kladná polarita na pin 2 a záporná polarita na pin 1. Otevřeny jsou tranzistory Q4 a Q5. Zavřením všech čtyř tranzistorů nepřivádí můstek na svorky K1 žádné napětí (motor se protáčí). Programem musí být ošetřena možnost otevření tranzistorů v nesprávné kombinaci (Q3 a Q5, popřípadě Q4 a Q6), v opačném případě by došlo ke zkratu napájecího zdroje. Tranzistory s vodivostí typu N a P obsahují komutační diodu, která zajistí vybití indukční zátěže (motoru). Návrh schématu je vytvořen v programu Eagle 6.4.0 Light (viz obrázek 1.4).



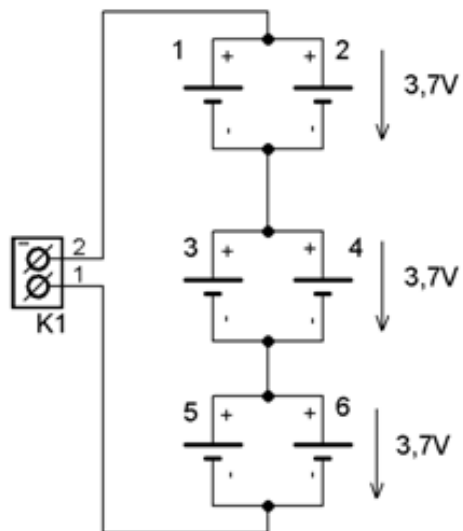
Obrázek 1.3: Pohled na H-můstky, uchycené na vnitřní straně základní konstrukce.



Obrázek 1.4: Schéma zapojení H-můstku sestaveného z diskrétních součástek.

1.4 Akumulátory

Jako zdroj elektrické energie byl vybrán nabíjecí akumulátor Sony 18650GR. Nominální napětí akumulátoru je $U = 3,7$ V. Kapacita akumulátoru je 2400 mAh. Akumulátor je válcového tvaru, má rozměry 18,25 mm × 65,0 mm a váhu 48 g. Každá dvojice je zapojena paralelně. Při paralelním zapojení je napětí jednoho článku 3,7 V, kapacita dvojice vzroste na dvojnásobnou hodnotu 4800 mAh. Schéma zapojení je na obrázku 1.5. Výsledné nominální napětí serioparalelního zapojení je 11,1 V s kapacitou 4800 mAh, obrázek 1.6.



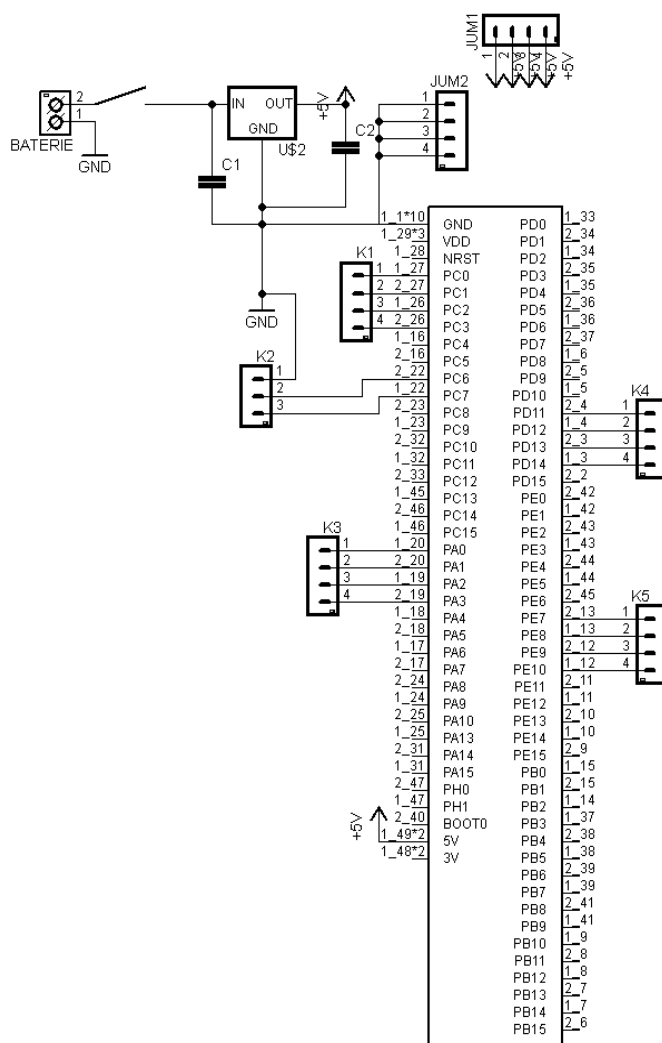
Obrázek 1.5: Schéma zapojení akumulátorů s nominálním napětím 11,1 V.



Obrázek 1.6: Akumulátor v sérioparalelním zapojení.

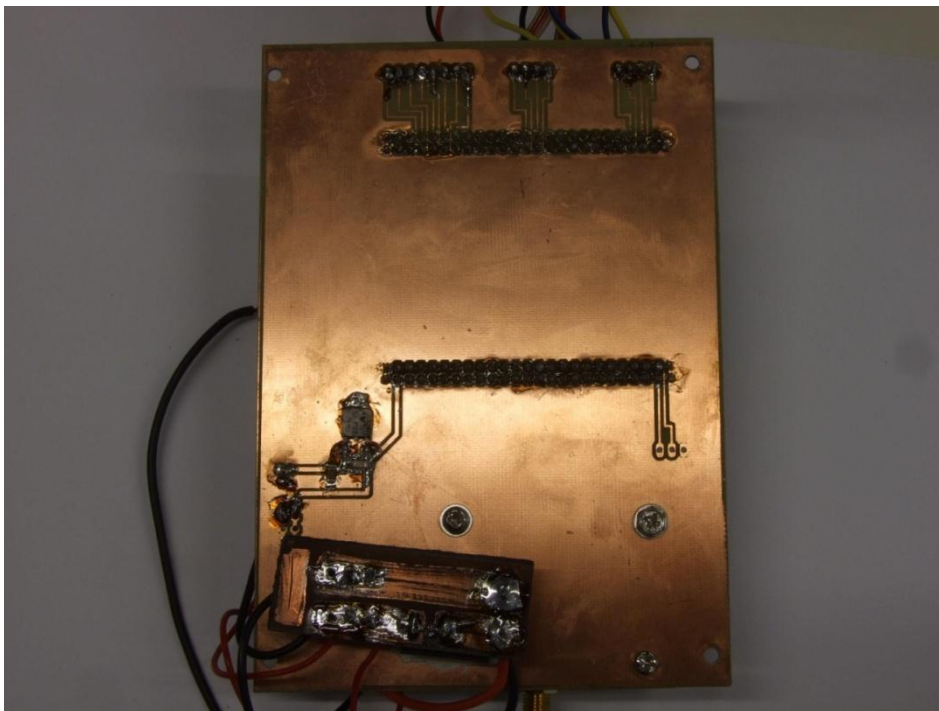
1.5 Základní deska

Základní deska je vyrobena pro uchycení vývojové desky STM32F4 Discovery na šasi, ke kterému je přimontována pomocí čtyř distančních šroubů. Deska je osazena dvěma páry konektorů, do kterých se zasouvá vývojová deska STM32F4 Discovery. Z vývojové desky jsou dále vyvedeny signálové piny pro pohon motorů na H-můstky. Označení pinů je následující: PA0 - PA3, PC0 - PC3, PD11 - PD14, PE7 - PE10. Dále se na desce nachází lineární regulátor LM1117, který snižuje napájecí napětí baterie na 5 V. Elektronické schéma zapojení je na obrázku 1.7.



Obrázek 1.7: Schéma zapojení základní desky s mikrokontrolérem STM32F4.

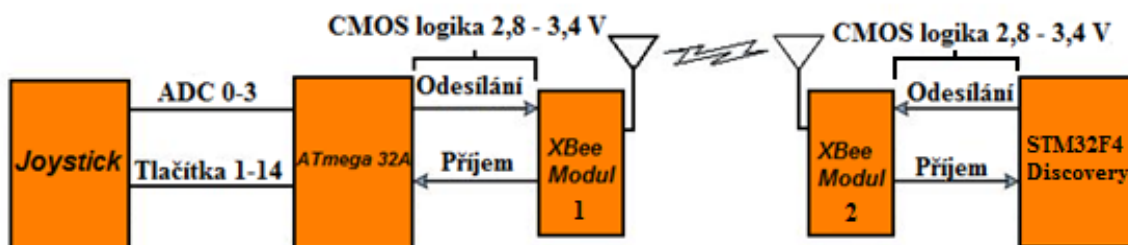
Na základní desce je umístěn kolébkový vypínač, který galvanicky odděluje akumulátor, základní desku, H-můstky a rozvody o napětí 5 V. Na obrázku 1.8, je zobrazena spodní strana základní desky s kolébkovým vypínačem.



Obrázek 1.8: Základní deska.

1.6 XBee moduly

Xbee-PRO jsou vysokofrekvenční bezdrátové moduly, vyrobeny společností Digi. XBee moduly mezi sebou komunikují vysokofrekvenčním bezdrátovým přenosem. S ostatními moduly (např. UM232) komunikují pomocí protokolu UART. Pro komunikaci mezi mikrokontroléry pomocí XBee modulů, je důležité splnit minimální přenosovou konfiguraci. Na obrázku 1.9 je zobrazena komunikace, která začíná vysíláním dat z joysticku, která jsou dále zpracována mikrokontrolérem ATmega32A a pomocí dvou datových linek dopravena do vysílací části XBee modulu 1. Druhá část vysokofrekvenčního přenosu, tedy XBee modul 2 přijme data a předá je mikrokontroléru STM32F407 [4].

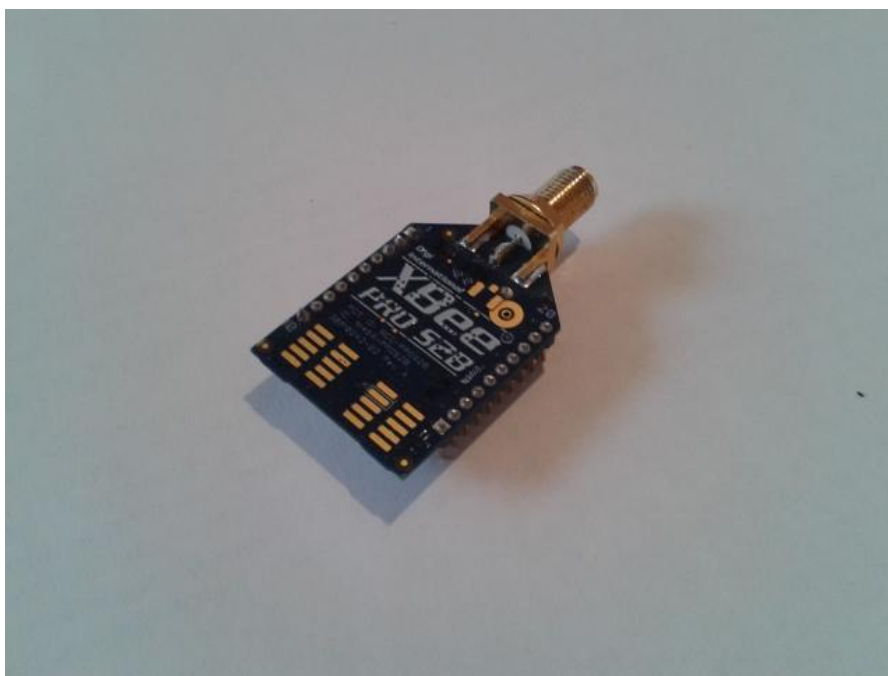


Obrázek 1.9: Schéma zapojení mikrokontroléru a XBee modulů, upraveno [4].

Vlastnosti XBee modulů

- komunikační vzdálenost uvnitř budovy, kanceláře nebo tovární haly je 60 m - 90 m
- v otevřeném prostranství je dosah komunikace od 750 m - 1600 m
- maximální vysílací výkon je 63 mW
- rychlost přenosu dat je až 250 000 bitů za sekundu
- vysílací maximální proud je 250 mA
- přijímací maximální proud je -55 mA
- komunikace ve vysokofrekvenčním pásmu 2,4 GHz

Na obrázku 1.10 je zobrazen použitý vysokofrekvenční XBee modul.



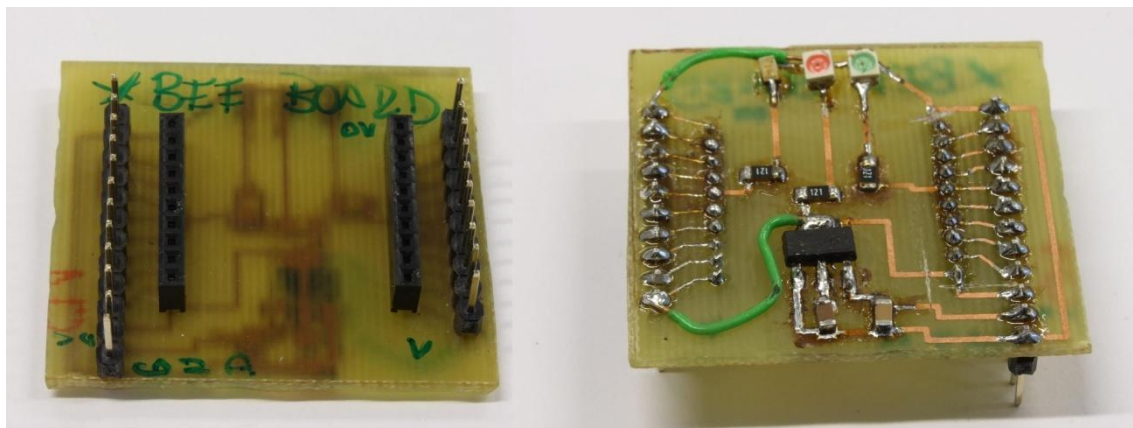
Obrázek 1.10: Bezdrátový vysokofrekvenční XBee modul.

1.6.1 XBee Board

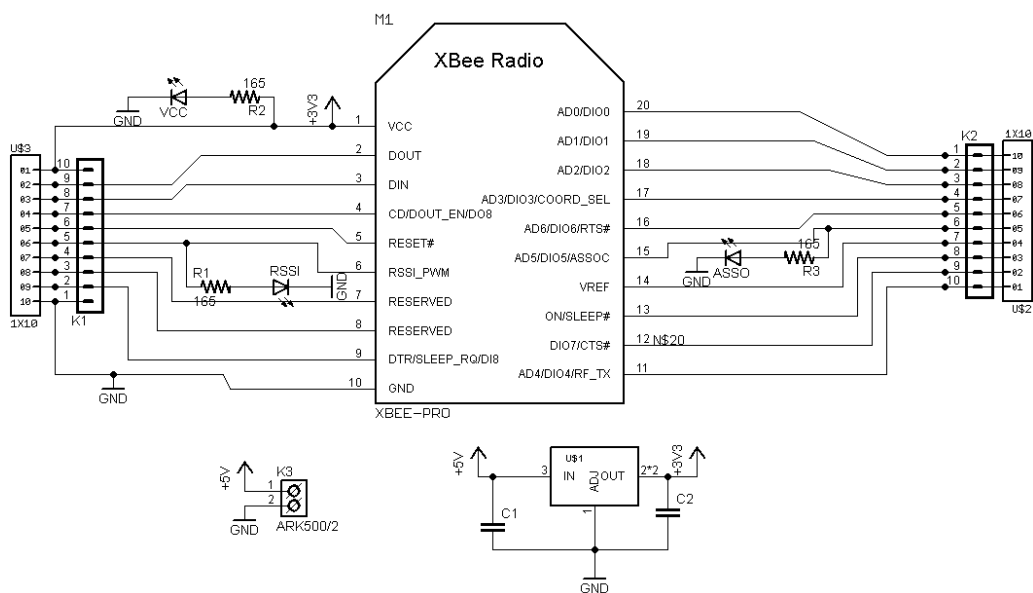
Pro práci s XBee modulem byla vytvořena deska (viz obrázek 1.11). Deska je navržena v programu Eagle 6.4.0 Light a osazena diskrétními součástkami. Příkladem může být lineární regulátor LM 1117, který snižuje napájecí napětí z 5 V na 3,3 V (pracovní napětí XBee modulu). Pro upevnění XBee modulu slouží dva páry hřebíkových konektorů s roztečí 2 mm a 2,54 mm. K indikaci napájení je použita LED dioda, označena jako VCC. O intenzitě přichozícího signálu informuje LED dioda RSSI a poslední LED dioda ASSO zpravuje obsluhu robota o úspěšném spojení XBee modulů.

XBee moduly jsou doplněny o dvě vnitřní všesměrové antény TL-ANT2405CL. Antény pracují v pásmu 2,4 GHz - 2,4835 GHz a poskytují zisk 5 dBi. Antény disponují konektory RP-SMA, které jsou kompatibilní s XBee moduly.

Elektronické schéma zapojení XBee Boardu je na obrázku 1.12.



Obrázek 1.11: Deska plošného spoje pro XBee Board.



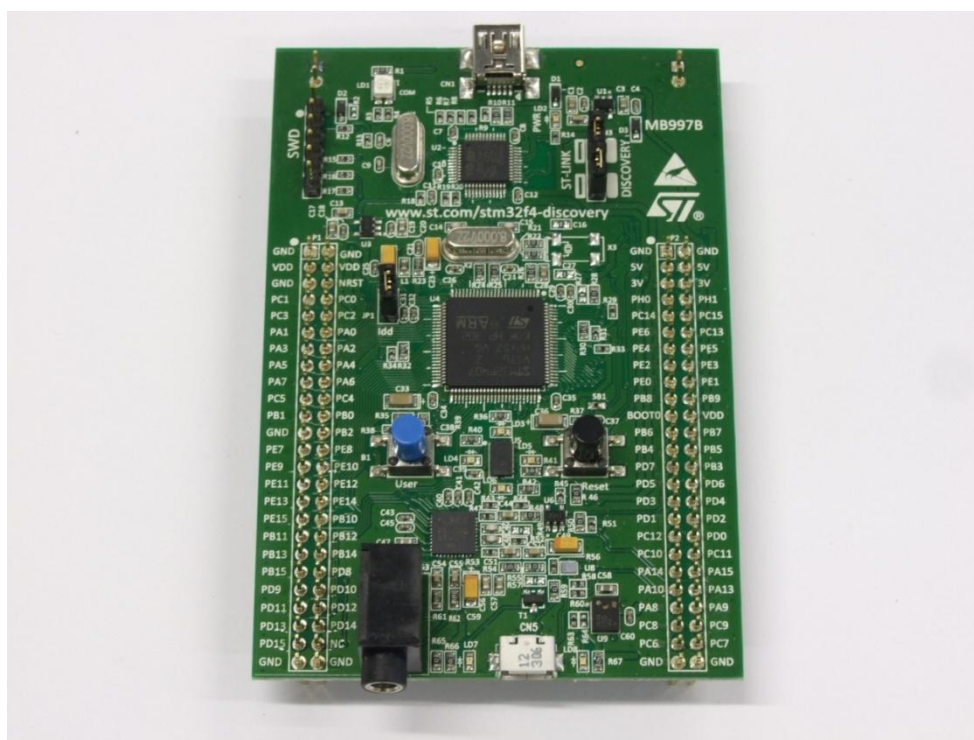
Obrázek 1.12: Elektronické schéma a zapojení XBee boardu.

1.7 Mikrokontrolér STM32F407

STM32F4 Discovery je vývojová deska, založena na procesoru STM32F407VGT6 s jádrem ARM® Cortex-M4 (viz obrázek 1.13). Vývojová deska obsahuje MEMS digitální akcelerometr, mikrofon, audio DAC převodník třídy D s integrovanými ovladači, LED diody, tlačítka a mikro USB konektor. Procesor může pracovat na frekvenci až 168 MHz. Hlavní předností procesoru je schopnost pracovat s čísly s plovoucí desetinnou čárkou (floating point unit).

Mikrokontrolér STM32F407 disponuje výkonnou pamětí flash o kapacitě 1 MB a statickou pamětí RAM s kapacitou 192 kB. V procesoru se nacházejí tři 12-bitové A/D převodníky s dobou převodu 0,41 μ s, dva D/A převodníky, dvanáct 16-bitových časovačů se dvěma rychlými 168 MHz PWM časovači pro řízení motorů, dva 32-bitové časovače a jeden 32-bitový generátor náhodných čísel. Procesor dále nabízí pokročilé komunikační rozhraní, mezi které patří například USB, I²C, I²S, USART (10,5 Mbits/s) a UART, SPI (37,5 Mbits/s), Ethernet, CAN, SDIO/MMC. Vývojová deska je opatřena rozhraním ST-LINK/V2, které slouží jako nástroj k programování a ladění aplikace na vývojové desce [5],[6],[7].

Díky nadstandardnímu vybavení je mikrokontrolér STM32F407 použitelný pro velké množství aplikací (např. k řízení motorů, technologických procesů, má využití pro zdravotní a průmyslové aplikace, PLC systémy, tiskárny, skenery, alarmy aj.).

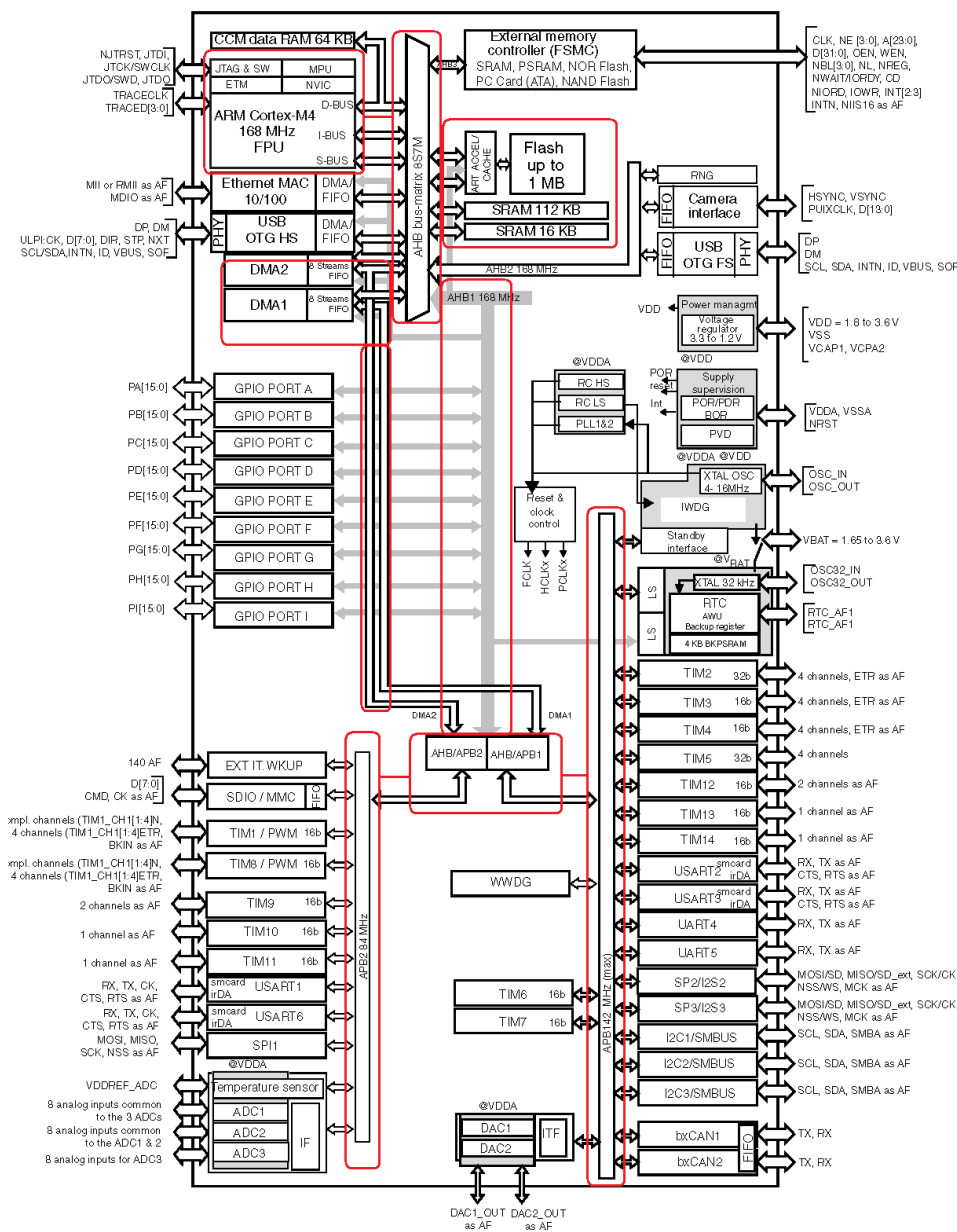


Obrázek 1.13: Vývojová deska STM32F4 Discovery s mikrokontrolérem STM32F407.

1.7.1 Vnitřní schéma mikroprocesoru STM32F407VGT6

Pro lepší pochopení struktury mikroprocesoru je na obrázku 1.14 uvedeno obecné schéma všech použitelných periférií. Všechny důležité části procesoru jsou označeny v červených rámečcích.

V levém horním rohu je zobrazeno jádro Cortex-M4 CPU, na které navazuje JTAG rozhraní pro programování flash paměti a NVIC (Nested Vector Interrupt Controller). Procesor je spojen s flash pamětí a s SRAM pamětí pomocí maticové sběrnice (Matrix Bus). Maticová sběrnice je napojena na hlavní sběrnici AHB1 s maximální frekvencí 168 MHz. Na sběrnici AHB1 jsou připojeny všechny porty A-I. Sběrnice AHB1 je dále spojena s dvěma dalšími sběrnici APB1 a APB2. Sběrnice APB1 může pracovat s maximální frekvencí 142 MHz a sběrnice APB2 s frekvencí 84 MHz. Rychlejší sběrnice APB1 má větší množství periférií než sběrnice APB2. Obě tyto sběrnice mají schopnost dynamického přístupu do paměti pod zkratkou DMA (Dynamic Memory Access) [7].



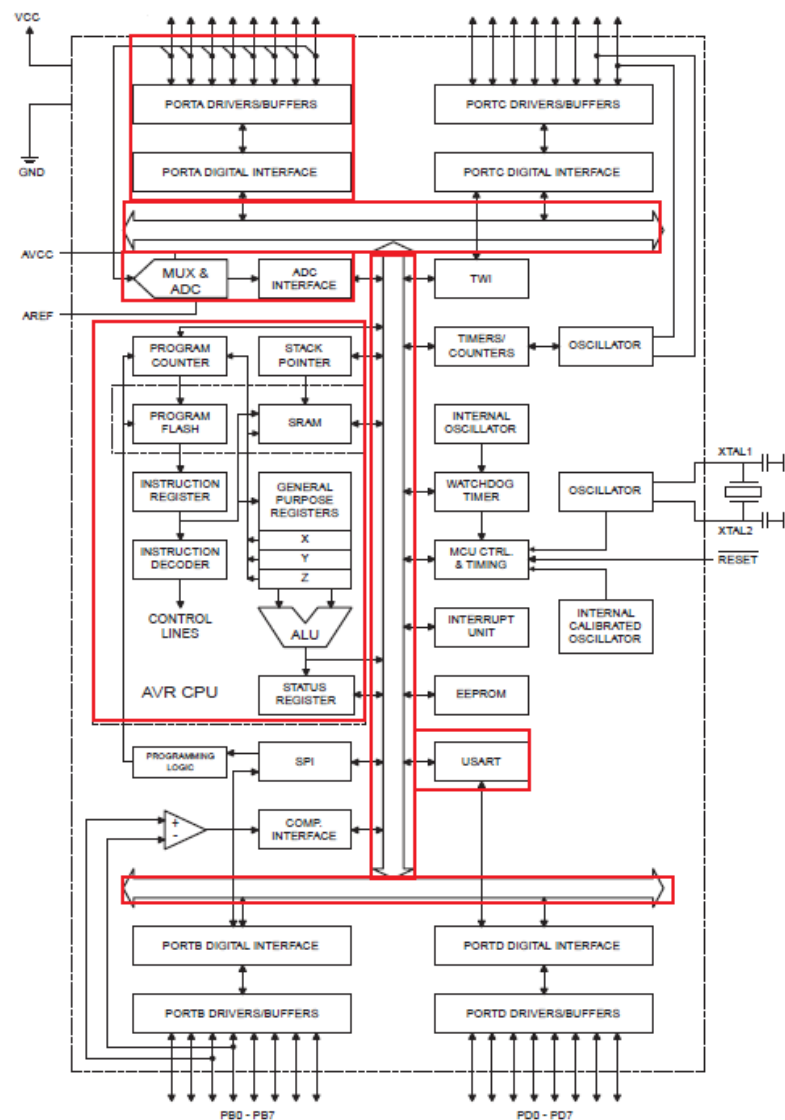
Obrázek 1.14: Schéma struktury mikrokontroléru STM32F407VGT6 s jádrem Cortex-M4, upraveno[7].

1.8 Joystick

Pro řízení robota byl vybrán čtyřosý joystick firmy Trust. Joystickem lze pohybovat ve třech stupních volnosti. Osy x a y popisují horizontální a vertikální část pohybu joysticku. Osa z je kolmá na osy x a y a opisuje kruhovou dráhu. Další polohový prvek je posuvný jezdec a třináct tlačítek. Osy x, y, z a posuvný jezdec jsou realizovány pomocí potenciometrů. Tlačítka jsou spínané proti zemi. Pro zpracování aktuální pozice páky joysticku jsou využity A/D převodníky mikrokontroléru ATmega32A.

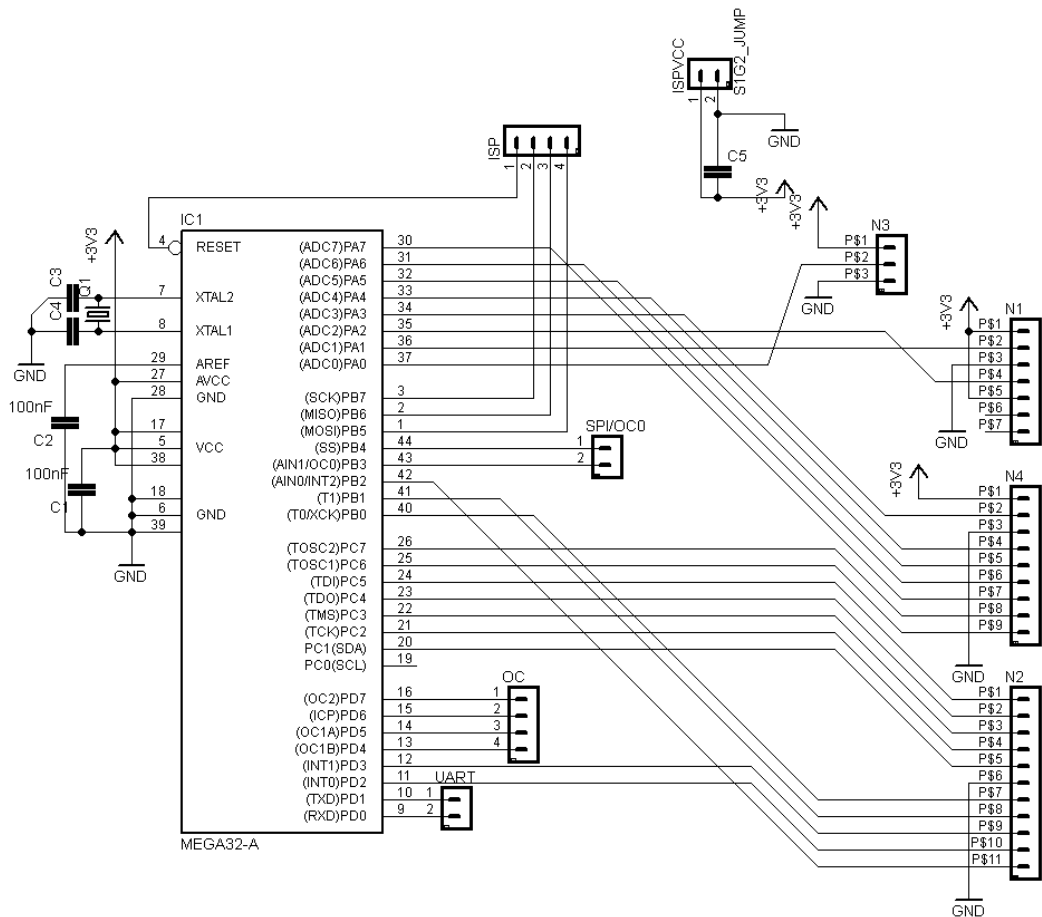
1.9 Atmel AVR ATmega32A

ATmega32A je výkonný, 8-bitový mikrokontrolér od firmy Atmel. Mikrokontrolér disponuje 2kB pamětí SRAM, 1024 bitovou pamětí EEPROM a 32kB programovatelnou pamětí flash. Podporuje ISP programování a JTAG rozhraní. Mezi hlavní periferie patří dva 8-bitové časovače a jeden 16-bitový časovač s porovnávacími módy (compare modes), interní a externí přerušení, čtyři PWM kanály, 8-kanálový 10-bitový A/D převodník, USART a SPI sériový port, Watchdog časovač a analogový komparátor. Mikrokontrolér je v pouzdře TQFP40, z důvodu úspory místa v joysticku. Na obrázku 1.15 je vnitřní schéma zapojení mikrokontroléru ATmega32A, v červených rámečcích je vyznačena sběrnice, A/D převodník společně s portem A, na kterém jsou vyvedeny jednotlivé kanály A/D převodníku, a sériová komunikace UART [8].



Obrázek 1.15: Vnitřní schéma mikrokontroléru ATmega32A, upraveno[8].

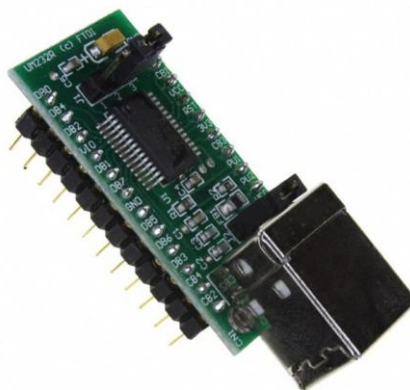
Mikrokontrolér je použit k převodu analogového signálu z os joysticku na digitální signál. Všechny výstupy z potenciometrů jsou přivedeny na vstupy 10-bitových A/D převodníků mikrokontroléru. Tlačítka joysticku jsou přivedena na ostatní výstupy mikrokontroléru. Z mikrokontroléru je vyvedena UART komunikace pro přenos dat sériovou linkou, SPI rozhraní a ISP rozhraní. ISP je zkratkou pro In System Programming. Toto rozhraní slouží k nahrávání programu do mikrokontroléru. Mikrokontrolér je osazen krystalem, který pracuje na frekvenci 8 MHz a je napájen na 3,3 V, aby mohl bezpečně komunikovat s XBee modulem, který rovněž pracuje na napětí 3,3 V. Na obrázku 1.16 je elektronické schéma zapojení mikrokontroléru ATmega32A a joysticku.



Obrázek 1.16: Schéma zapojení joysticku a mikrokontroléru ATmega32A.

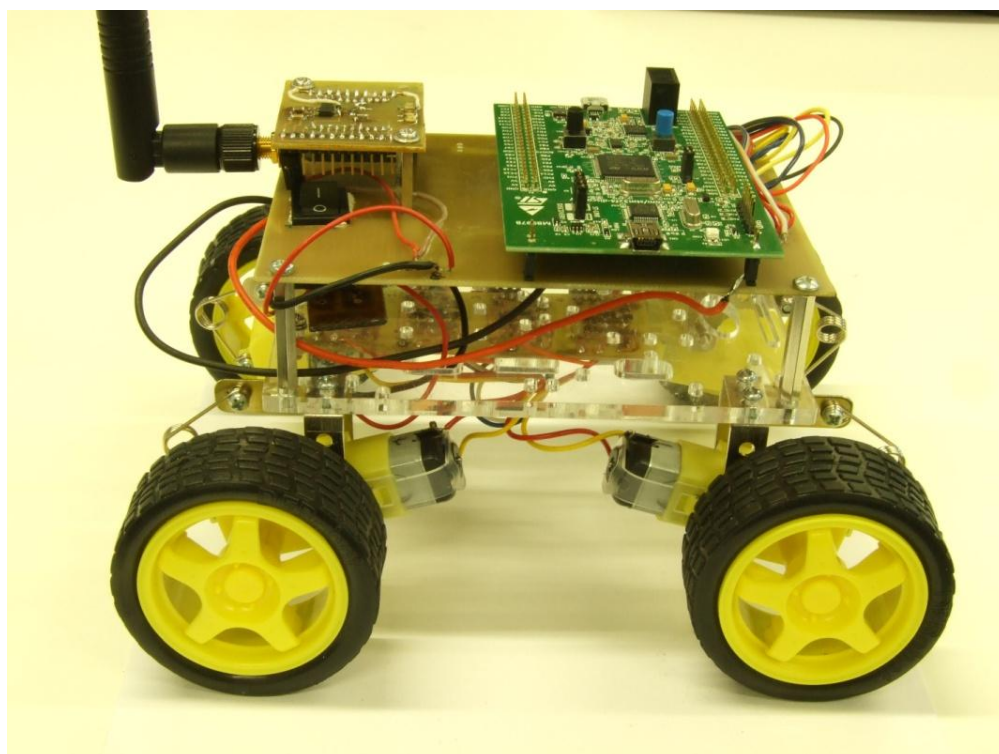
1.10 Modul UM232

UM232 je vývojový modul, který používá FT232RL integrovaný obvod (viz obrázek 1.17). UM232 je navržen tak, aby zapadl do kontaktního pole, nebo standardního 24 pinového pouzdra. Základní vlastností je UART rozhraní, podporující 7 nebo 8-bitovou komunikaci se stop bity a paritou. Mezi typické aplikace modulu UM232 patří zápis a čtení flash karty, nebo může být použit jako rozhraní k digitální kameře [9].



Obrázek 1.17: Vývojový modul UM232 s FTDI čipem.

Celková konstrukce robota se nachází na obrázku 1.18.

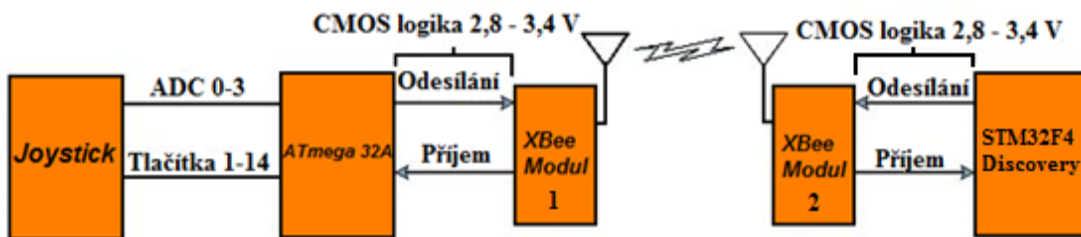


Obrázek 1.18: Celková konstrukce robota.

2. Programové vybavení robota

Robot se skládá z jednotlivých hardwarových částí, které jsou uvedeny v kapitole 1. Správná činnost robota závisí na propojení a programovém nastavení dílčích částí, které jsou uvedeny v blokovém schématu na obrázku 2.1. Blokové schéma můžeme rozdělit na dvě části. Do první části můžeme zahrnout řízení robota pomocí joysticku, převod signálu z joysticku pomocí mikrokontroléru ATmega32A a přenos dat do vysílací části XBee modulu. Druhá část blokového schématu obsahuje příjem dat pomocí XBee modulu umístěného na robotovi a následný přenos dat z XBee modulu do mikrokontroléru STM32F407 pomocí sériové linky UART.

První kapitola popisuje část programu, která převede aktuální polohu páky joysticku na digitální signál pomocí A/D převodníku. Následující kapitola se zabývá konfigurací XBee modulů pomocí programu X-CTU, v podkapitole je uveden princip přenosu sériové komunikace. V podkapitole je také vysvětlen důvod použití modulu UM232 s FTDI čipem. Poslední část kapitoly se zabývá příjmem dat z XBee modulu do mikrokontroléru STM32F407 a generováním PWM signálu pro řízení motorů.



Obrázek 2.1: Blokové schéma sériové komunikace mezi XBee moduly a mikrokontrolérem, upraveno [4].

2.1 Přenos dat z joysticku do XBee modulu mikrokontrolérem ATmega32A

Na obrázku 2.2 se nachází část zdrojového kódu, který popisuje převod signálu z joysticku a následný přenos do XBee modulu. Zdrojový kód byl vytvořen v programu CodeVisionAVR Evaluation, který slouží k programování mikrokontrolérů firmy Atmel.

Napěťový signál z joysticku je přiveden na vstup A/D převodníku s postupnou aproximací, který převede signál na digitální hodnotu. V další části zdrojového kódu je uveden princip odesílání dat sériovou linkou. Data jsou poté přijata XBee modulem. Nejprve je odeslán identifikátor osy. Příjemní strana rozeznává identifikátory osy x, y, z, p a q. Po skončení převodu je do vytvořené proměnné uloženo číslo datového typu Word. Jelikož je komunikace nastavena na 8-bitový datový rámeček, je zapotřebí rozdělit 10-bitové číslo na dvě části. Nejprve program pošle 8-bitové číslo s nejméně

významným bitem. Dále je poslána zbývající část 10-bitového převodu s nevýznamnějším bitem. Nakonec je poslán identifikátor, který informuje o dokončení převodu.

```
x=read_adc(2); //převod x-ové osy A/D převodníkem
USART_Transmit('x'); //odeslání znaku pro rozpoznání osy
xlsb=(char)(x & 0xFF); //první byte slova word
USART_Transmit(xlsb); //odeslání první části slova word
xmsb=(char)(x >> 8); //druhý byte slova word
USART_Transmit(xmsb); //odeslání druhé části slova word
USART_Transmit('q'); //odeslání znaku pro dokončení převodu
```

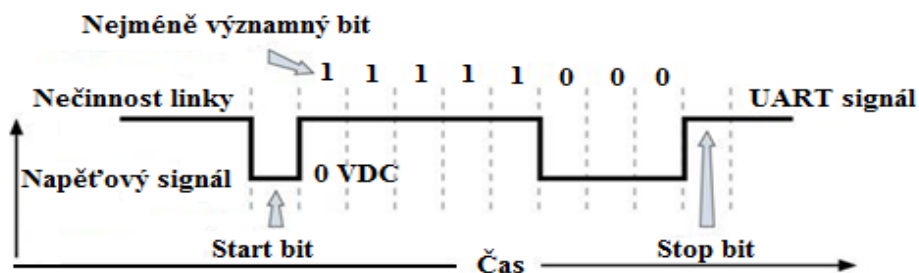
Obrázek 2.2: Zdrojový kód převodu a posílání dat.

2.2 Konfigurace XBee modulů

Konfigurace XBee modulů probíhá ve dvou fázích. V první fázi je XBee modul připojen k počítači pomocí modulu UM232 s FTDI čipem, který zajistí správnou komunikaci s počítačem pomocí UART protokolu. V další fázi je uveden popis nastavení XBee modulů pomocí programu X-CTU. V následující podkapitole je uveden princip sériové komunikace.

2.2.1 Sériová komunikace

Sekvence přenosu asynchronní komunikace (UART) se skládá ze start bitu, 5 - 8 datových bitů, stop bitu a paritních bitů. Start bit zahájí komunikaci změnou hodnoty datové linky z logické jedničky na logickou nulu. Po start bitu následuje sekvence osmi přenášených datových bitů. První z 5 nebo 8 datových bitů má význam nejméně významného (LSB) bitu a poslední z datových bitů má hodnotu nejméně významnějšího (MSB) bitu. Posledním bitem je stop bit, který ukončuje komunikaci a vrací hodnotu datové linky na logickou jedničku. Poté datová linka čeká na další sekvenci dat [4]. Schéma přenosu sériové komunikace je uvedeno na obrázku 2.3.



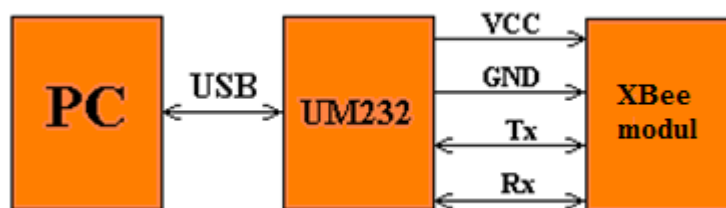
Obrázek 2.3: Princip přenosu 8-bitové sériové komunikace.

Komunikace je využita k přenosu dat mezi mikrokontrolérem ATmega32A a XBee modulem. Dále jsou pomocí protokolu UART nastaveny XBee moduly. Sériová linka je nastavena i na mikrokontroléru STM32F407, který přijímá data z XBee modulu. Nastavení sériové komunikace:

- přenosová rychlost 9600 baudů za sekundu
- 8 datových bitů
- 1 stop bit
- žádná parita

2.2.2 Připojení XBee modulu k počítači pomocí modulu UM232

XBee modul nemá standardní komunikační rozhraní pro připojení k počítači (USB, RS232). Komunikuje pouze pomocí UART protokolu, který je vyveden na piny XBee modulu. Z tohoto důvodu je k nastavení XBee modulu použito jiné zařízení, které komunikuje pomocí UART protokolu. K nastavení komunikace byl vybrán modul UM232, který převede pomocí FTDI čipu UART protokol z XBee modulu do počítače. Blokové schéma zapojení počítače, XBee modulu a UM232 je na obrázku 2.4.



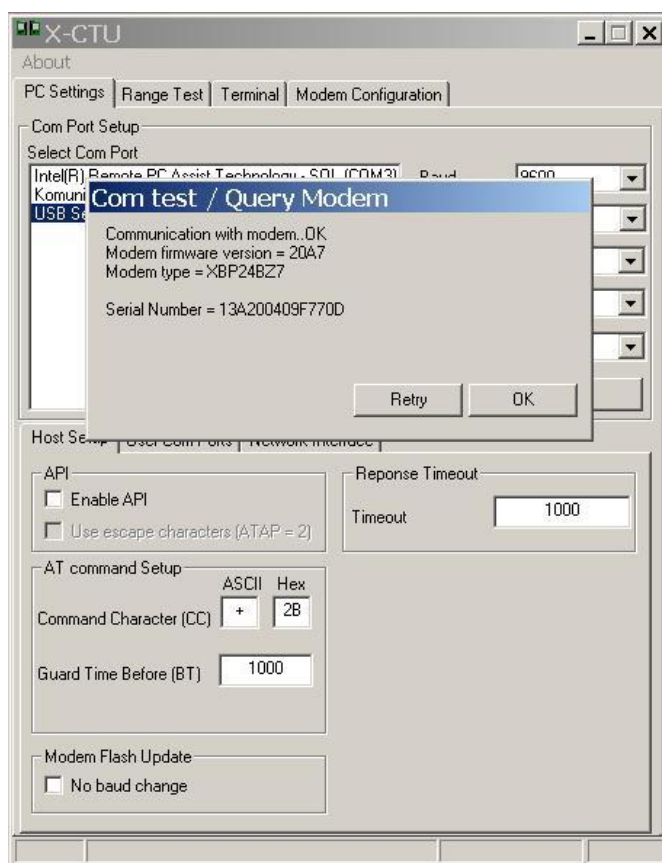
Obrázek 2.4: Schéma připojení XBee modulu pomocí modulu UM232 k počítači.

Modul UM232 je k počítači připojen pomocí USB kabelu a je vsazen do kontaktního pole. Modul UM232 je opatřen interním 3,3 V regulátorem, který je využit k napájení XBee Boardu. XBee modul je připojen k modulu UM232 čtyřmi vodiči. Dva vodiče VCC a GND jsou využity jako zdroj napájení. Zbývající dva vodiče slouží ke komunikaci. Vodiče pro komunikaci mají označení Tx (vysílání) a Rx (přijímání). Pin Tx z XBee modulu musí být přiveden na pin Rx UM232 modulu. Analogicky je připojen pin Rx. Jestliže je XBee modul správně připojen, je počítačem chápán jako virtuální port (COM). Pomocí programu X-CTU je XBee modul připraven ke konfiguraci.

2.2.3 Konfigurace XBee modulu pomocí programu X-CTU

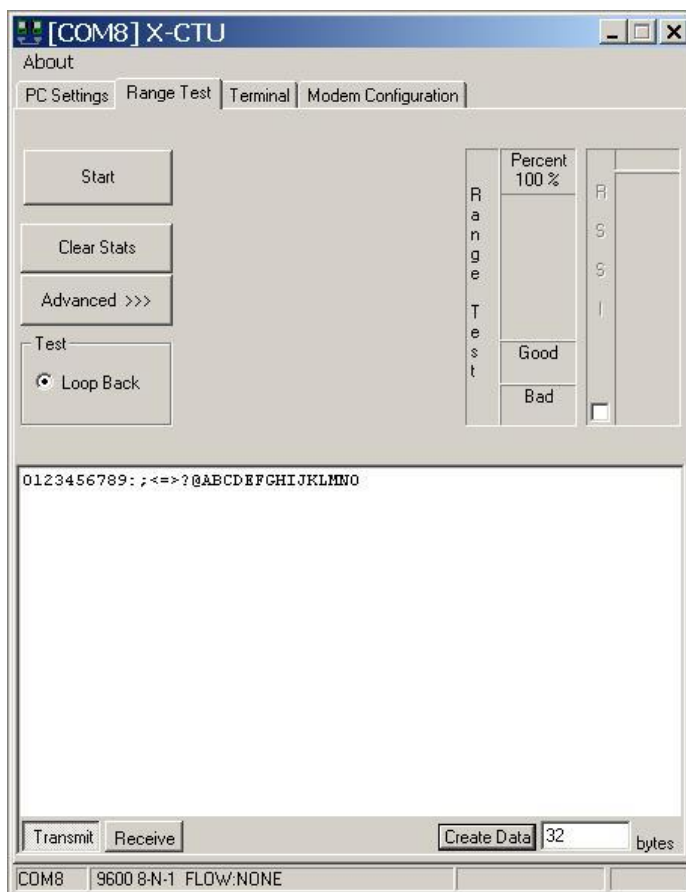
X-CTU je program vyvinut společností Digi a je určen pro systémy Windows. Byl navržen, aby komunikoval s firmwarem, který se nachází na vysokofrekvenčních výrobcích společnosti Digi. Program X-CTU má grafické uživatelské prostředí s jednoduchým ovládáním. Program má 4 záložky: PC settings, range test, terminal a modem configuration [10].

Záložkou PC settings nastavuje obsluha počítače COM port, na kterém je pomocí modulu UM232 XBee modul připojen. První záložka také nastavuje sériovou komunikaci (přenosovou rychlost, počet datových bitů, paritu, řízení toku a počet stop bitů). Jestliže je XBee modul připojen na správný COM port a komunikace je správně nastavena, nabízí tlačítko test/query možnost přihlášení XBee modulu. Po úspěšném připojení XBee modulu se zobrazí oznamovací tabulka, ve které je uvedena verze firmwaru, identifikační číslo modulu a sériové číslo. Program s nastavením komunikace a oznamovací tabulkou je uveden na obrázku 2.5 [10].



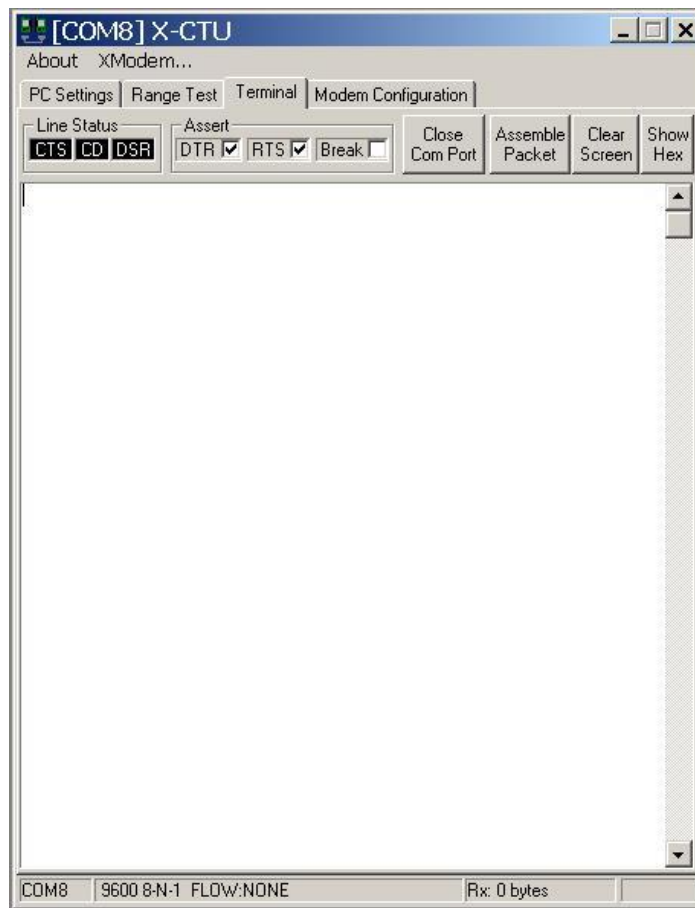
Obrázek 2.5: Záložka PC Settings v programu X-CTU, upraveno [10].

V další záložce je možnost testování rozsahu. Výsledek testu rozsahu je určen vzdáleností XBee modulu. Testování probíhá odesláním paketu dat a jeho přijetím v definovaném čase. Tabulka informuje v indikačním boxu (Range Test) o procentuální úspěšnosti testu. Záložka pro testování rozsahu je na obrázku 2.6.



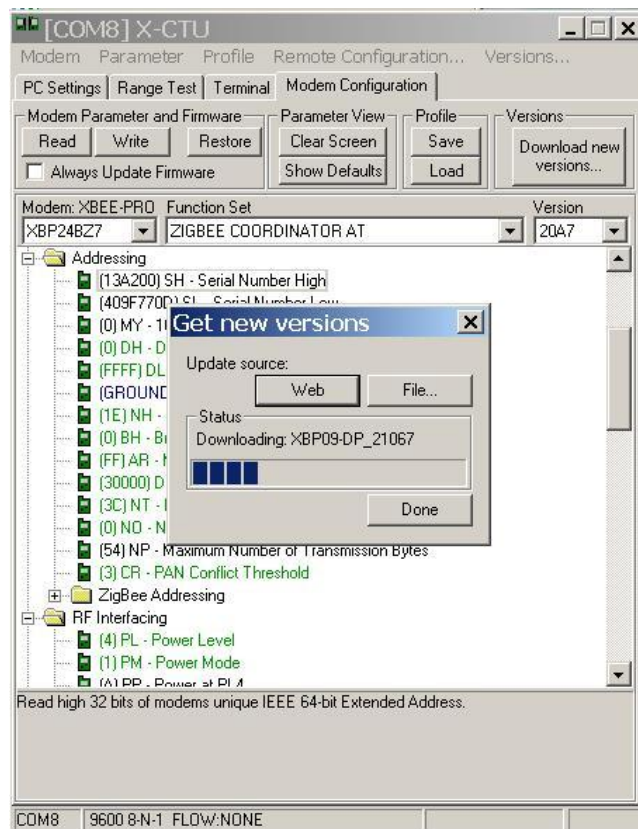
Obrázek 2.6: Testování rozsahu komunikace, upraveno [10].

Záložka terminál umožňuje komunikovat a nastavovat XBee moduly pomocí tzv. AT příkazů (viz obrázek 2.7). Terminál funguje i jako emulátor, tedy jako podpora pro jiné uživatelské platformy. Všechny AT příkazy, na které XBee moduly reagují, jsou popsány v jejich datasheetu [4]. Každý AT příkaz má svůj význam, na vyslaný příkaz XBee moduly reagují a odpovídají. Odeslané příkazy mají modrou barvu a přijaté příkazy barvu červenou. AT příkazy jsou zobrazeny na terminálu, aby si obsluha mohla zkontrolovat správné nastavení zařízení.



Obrázek 2.7: Záložka terminálu bez přenosu dat, upraveno [10].

Poslední záložka je konfigurace modemu (XBee modulu). Mezi její hlavní součásti patří grafické rozhraní, čtení a zapisování firmwaru, stahování nového firmwaru a jeho uložení. Pro zobrazení aktuálního nastavení XBee modulu, je nutné spustit čtení firmwaru v záložce modem Parameter and Firmware. Po úspěšném dokončení čtení se zobrazí typ zařízení a jeho funkce. Funkce je zobrazena v roletkovém menu. Pro správnou činnost komunikace mezi XBee moduly, musí být jeden XBee modul nastaven jako koordinátor a druhý XBee modul jako koncové zařízení. Záložka konfigurace modemu umožňuje nastavit mnoho parametrů (viz obrázek 2.8) [4].



Obrázek 2.8: Grafická konfigurace zařízení se stahováním nové verze firmwaru, upraveno [10].

2.3 Příjem dat z XBee modulu do mikrokontroléru STM32F407

Data přijatá XBee modulem jsou pomocí signálových vodičů předána mikrokontroléru. V mikrokontroléru STM32F407 je důležité nastavit komunikaci a obsluhu dat. Jelikož má mikrokontrolér velké množství periferních prvků, mohou některé piny zastávat více funkcí (tzv. alternate function). Proto je důležité v inicializaci komunikace vybrat správné piny, které slouží k přijímání a vysílání dat. V mikrokontroléru byla zvolena periférie UART5, která se stará o sériový přenos dat. Pro vysílání dat (Tx) je nastaven pin 12 na portu C a pro příjem dat (Rx) pin 2 na portu D. Dále je v komunikaci zapotřebí nastavit obsluhu přerušení. Obsluha přerušení je speciální asynchronní druh činnosti procesoru, která je vyvolána NVIC kontrolérem. Procesor, který vykonává posloupnost přesně definovaných příkazů v hlavní smyčce, může při určité události (např. při příjmu dat) vyvolat obsluhu přerušení. Procesor může během své práce reagovat na více než jedno přerušení, proto se obvykle v obsluhách přerušení nastavuje tzv. prioritizace přerušení. Nastavením priority přerušení je zaručeno, že procesor vykoná důležitější přerušení před méně důležitým. Po skončení důležitějšího

přerušení se vrátí na původní místo, ze kterého bylo přerušení vyvoláno, tedy do přerušení s menší prioritou, a poté se vrátí zpět do hlavní smyčky.

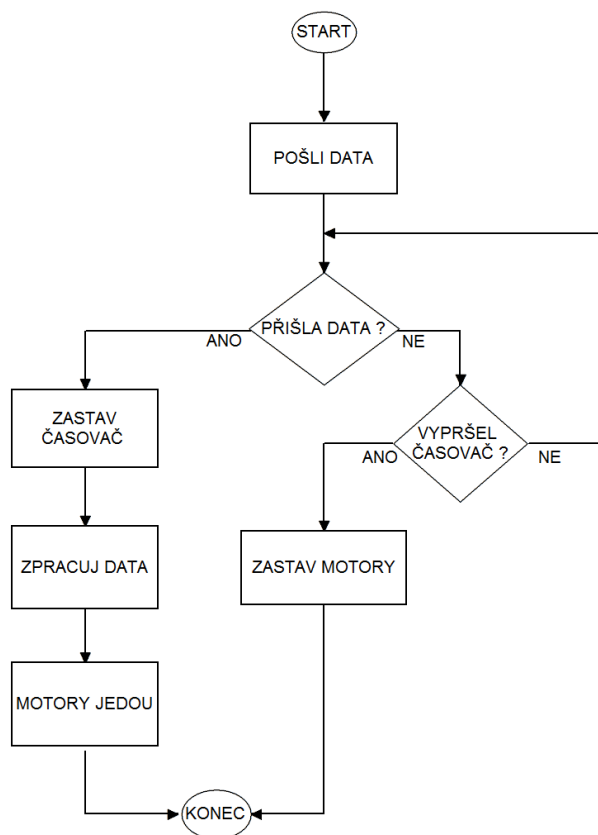
Obsluha přerušení je vyvolána NVIC kontrolérem na základě příjmu nových dat. Příjem dat je vykonán pouze v obsluze přerušení, která je umístěna mimo hlavní program. Na obrázku 2.9 je zdrojový kód obsluhy přerušení. Obsluha přerušení je vykonána, jestliže mikrokontrolér přijme data. Obsluha zkontroluje, jestli přišlo přerušení kvůli příjmu nových dat. Pokud přišlo přerušení z důvodu příjmu nových dat, jsou přijatá data uložena do proměnné, kde čekají na další zpracování. Jestliže jsou přijata všechna data, speciální proměnná informuje hlavní řídicí program. Hlavní program data převezme a zpracuje. Nakonec jsou přijatá data v proměnné přemazána a obsluha přerušení je ukončena.

```
void UART5_IRQHandler(void)
{
    static char received_string[MAXZNAK]={0,0,0,0}; // pole pro příjem dat
    int i = 0; //proměnná pro počítání znaků
    if(USART_GetITStatus(UART5,USART_IT_RXNE) != RESET)//informace-příjem
    dat
    {
        static uint8_t cnt = 0;
        char t = USART_ReceiveData(UART5); //proměnná t ukládá přijatá data
        if(t!='\0' && cnt<MAXZNAK)//čekání na ukončovací znak nebo na maximální
    počet znaků
        {
            received_string[cnt] = t; //zápis příchozího bytu do pole
            cnt++; //posun na další pozici
        }
        else
        {
            cnt = 0;
            nova_data=1; //oznámení hlavnímu programu-přijatá data
            strncpy (command,received_string, sizeof(received_string));//data jsou
    uložena do proměnné comamnd
            for (i = 0; i <= MAXZNAK-1; i++) //přemazání pole received string
                received_string[i] = '\0';
        }}}
}
```

Obrázek 2.9: Zdrojový kód, který popisuje příjem dat v obsluze přerušení.

2.3.1 Ochranný časovač

V průběhu komunikace mezi joystickem a mikrokontrolérem STM32F407 může dojít k výpadku nebo ke ztrátě signálu. Hlavním důvodem výpadku komunikace je ztráta viditelnosti XBee modulu. XBee moduly mají omezenou komunikační vzdálenost v uzavřených prostorech, naopak v otevřeném terénu vzroste komunikační vzdálenost několikanásobně. Ztráta signálu v uzavřeném prostoru je způsobena především kvůli kovovému a betonovému charakteru budovy. Signál může být nepřímo rušen dalšími zařízeními, které pracují blízko komunikační frekvence XBee modulů. Z tohoto důvodu je do komunikace zahrnut bezpečnostní prvek ve formě ochranného časovače. Časovač má za úkol zastavit činnost motorů, jestliže robot v definovaném čase neobdrží data z joysticku. Časovač, který stále čítá impulsy je při příjmu nových dat resetován. Jestliže mikrokontrolér neobdrží nová data z joysticku do doby, než časovač dopočítá čas do předem nastavené maximální hodnoty, dojde k zastavení motorů. Na obrázku 2.10 je vývojový diagram, který schématicky popisuje část programu, která zastaví motory.

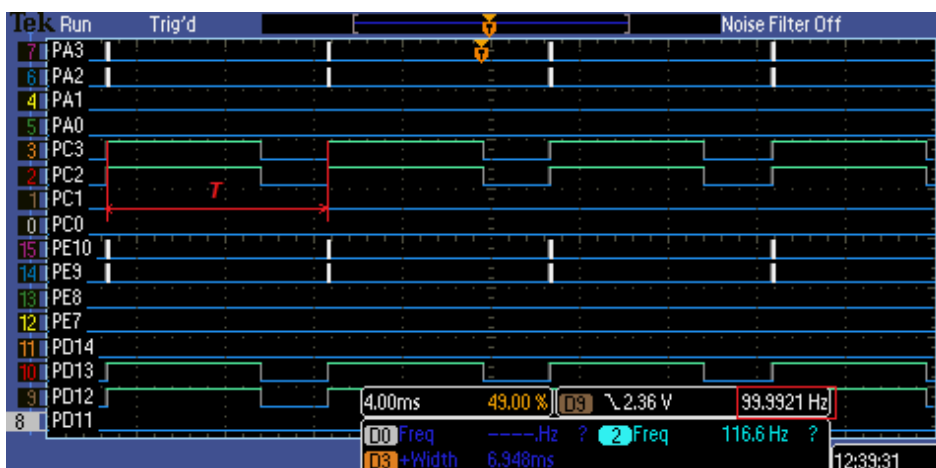


Obrázek 2.10: Schématický popis programu, který při ztrátě dat zastaví motory.

2.4 Řízení motorů pulsní šířkovou modulací

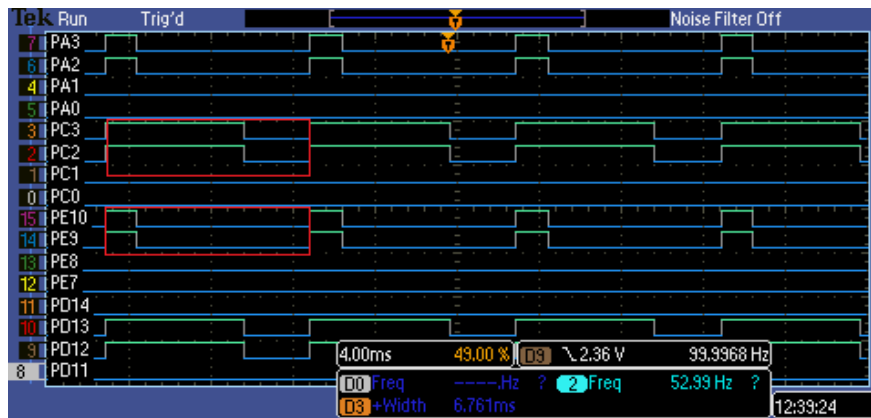
Pulsní šířková modulace, dále jen PWM, je technika pro kódování dat, která může přenášet data mezi dvěma zařízeními [3],[11]. PWM modulaci můžeme použít k řízení analogových obvodů pomocí digitálních výstupů mikrokontroléru. Digitální signál, který generuje mikrokontrolér pomocí časovače, má obdélníkový průběh. Signál může být pouze ve dvou stavech, v logické jedničce (+3,3 V), nebo v logické nule (0 V). Důležitou vlastností PWM signálu je střída, která určuje poměr délky pulsu logické jedničky vůči periodě, obrázky 2.11 a 2.12. PWM signál je použit pro širokou škálu aplikací, především pro měření, komunikaci nebo řízení výkonu elektrických zařízení.

Obrázky jsou pořízené z digitálního osciloskopu MSO2024B, který podporuje sledování 16 datových linek. Signál je využit pro řízení motorů. Aktivní porty mikrokontroléru STM32F407 (PA0 - PA3, PC0 - PC3, PD11 - PD14, PE7 - PE10), na kterých jsou vyvedeny PWM signály, jsou přivedeny na 4 pinové vstupy H-můstku, obrázek 1.3. PWM signál, který je generován časovačem, ovládá H-můstky podle kap. 1.3, otáčky motorů jsou úměrné střídě PWM signálu.



Obrázek 2.11: PWM signál generovaný časovačem s vyznačenou konstantní periodou.

Na obrázku 2.11 je vyznačena perioda signálu T s konstantní frekvencí $f = 100 \text{ Hz}$. Na dalším obrázku 2.12 jsou v rámečcích dva signály s odlišnou střídou. Střída vyznačená v dolním rámečku má přibližně čtvrtinový výkon oproti střídě v horním rámečku.

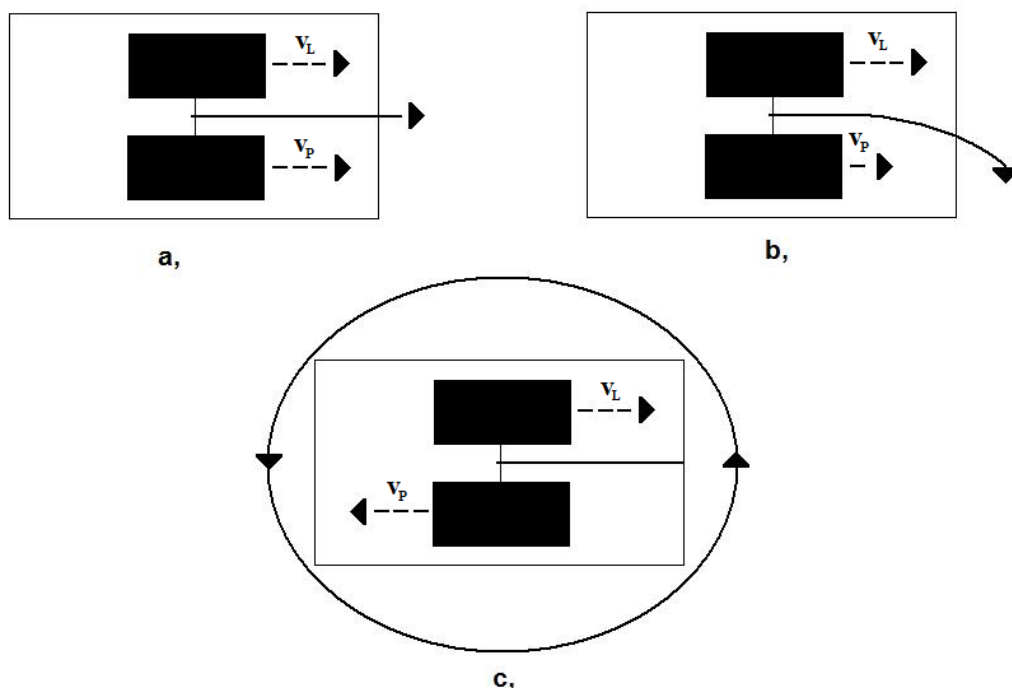


Obrázek 2.12: PWM signál generovaný časovačem s odlišnými střídami.

3. Řízení robota

Tato kapitola je zaměřena na řízení sestaveného robota. Mobilní roboty je možno řídit různými způsoby. Modely řízení kolových vozidel můžeme rozdělit podle pohybů, kterých jsou koloví roboti schopni. Do třech hlavních kategorií jsou rozděleni mobilní roboti, kteří mají všesměrové podvozky, diferenciální podvozky (tank), nebo pohyblivou přední nápravu (auto). Jeden z možných přístupů, který je využit pro řízení robota, je řízení pomocí odometrie. Odometrie je založena na geometrické znalosti modelu robota [12]. Sestavený robot byl vytvořen na principu diferenciálního podvozku, který podléhá tzv. diferenciálnímu řízení. Základem diferenciálního podvozku jsou čtyři motory, které jsou pevně namontovány na konstrukci robota. Důležitou vlastností motorů je jejich nezávislé řízení. Diferenciální řízení je mechanicky jednodušší než pohon jednoosého kola nebo celé nápravy. Naopak ovládání diferenciálního řízení je o něco komplikovanější než ovládání jednoosého kola, protože vyžaduje koordinaci všech poháněných kol [3].

Princip diferenciálního řízení je založen na určení středu otáčení robota. Jestliže je určen střed otáčení a využít princip nezávislých motorů, může se robot pohybovat třemi způsoby. Na obrázku 3.1 je zobrazen zjednodušený model diferenciálně řízeného robota.



Obrázek 3.1: Tři způsoby pohybu diferenciálně řízeného robota, upraveno [3].

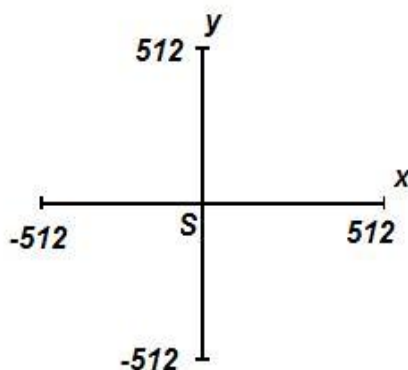
Na zjednodušeném obrázku jsou vyznačeny černým obdélníkem vždy dvojice kol. Rychlosti motorů jsou na levé straně označeny v_L a na pravé straně jsou označeny v_P . Při jízdě dopředu musí mít všechna kola stejnou rychlost $v_L = v_P$. Robot se pohybuje přímým směrem (viz obrázek 3.1a). Na obrázku 3.1b jsou zobrazeny rozdílné rychlosti obou dvojic motorů $v_L > v_P$ nebo $v_L < v_P$, poloměr otáčení je nenulový a robot zatáčí s určitým poloměrem. Poslední obrázek 3.1c zobrazuje rotaci robota kolem středu otáčení $v_L = -v_P$, nebo $v_P = -v_L$.

3.1 Kinematika řízení

Pro správné ovládnání robota pomocí diferenciálního řízení je potřeba zvolit kinematický model, který určí směr a rychlost robota. Základní kinematický model, který je popsán v knize Thomase Bräundla [3], vychází ze známé polohy robota. Poloha robota je určena enkodéry. Enkodéry posílají data do řídicí jednotky, která se stará o řízení robota. Enkodéry jsou v tomto případě nahrazeny daty, která posílá joystick. Joystick posílá data ze dvou os, které je zapotřebí zpracovat a správně interpretovat. Stávající souřadnicový systém joysticku je nevhodný pro řízení robota a je nutné ho upravit. Je důležité změnit kinematický model řízení, který přijatá data zpracuje a vytvoří funkci, podle které bude robot řízen [3].

3.1.1 Určení souřadnicového systému

Kartézský souřadnicový systém zobrazuje čísla v pravoúhlé soustavě s počátkem osového kříže ve středu $S[0,0]$. Bod o souřadnicích $P[x,y]$ reprezentuje aktuální pozici joysticku v rozsahu od 0 do 1023 bodů (bitů). Jestliže se tedy střed joysticku nachází uprostřed osového kříže, má aktuální hodnota joysticku střed v souřadnicích $S[512,512]$. Odečtením hodnoty 512 od x-ové a y-ové osy joysticku, je střed osového kříže joysticku upraven do pozice $S [0,0]$. Na obrázku 3.2 je zobrazen upravený střed osového kříže joysticku se středem $S[0,0]$.

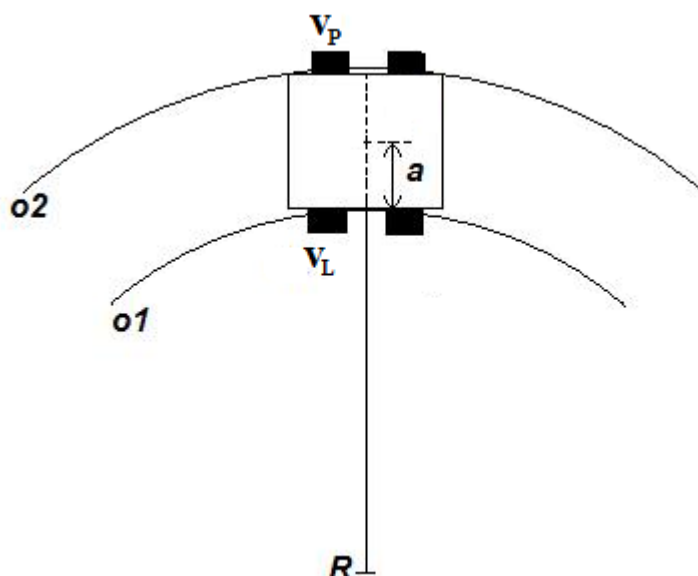


Obrázek 3.2: Souřadnicový systém pro řízení robota se středem $S[0,0]$.

Během testovacích jízd byl objeven problém se stálostí středu joysticku. Napínací pružiny, které zajišťují pozici páky ve středu joysticku, mají odchylky. Odchylky jsou závislé na napínání pružin a jejich schopnosti vrátit se do původní pozice joysticku. Z tohoto důvodu byla do řízení zahrnuta mrtvá zóna, která eliminuje mírné odchylky nedokonalé středové pozice joysticku. Mrtvá zóna je stanovena programově a její hodnota je ± 30 bitů.

3.1.2 Kinematický model pro řízení robota

Model řízení robota je založen na pohybu robota po kruhové dráze s poloměrem R . V závislosti na různé rychlosti levé a pravé strany motorů, se mění rychlost a poloměr zatačení robota. Na obrázku 3.3 je vyznačena rychlost levých a pravých motorů v_L a v_R , pološířka robota a a část kružnic, po kterých se budou motory pohybovat v závislosti na poloměru kružnic.



Obrázek 3.3: Kinematický model pro řízení robota, upraveno [3].

Vzdálenost, kterou motory ujedou po kruhové dráze je uváděna v metrech. Vztah (1) vypočítá obvody kružnic. Výchozím bodem pro stanovení poloměru R , podle kterého se budou motory pohybovat, je střed robota. Obvod kružnic je také určen přičtením nebo odečtením robotovy pološířky a :

$$\begin{aligned} o_1 &= 2 \times \pi \times (R - a) \\ o_2 &= 2 \times \pi \times (R + a). \end{aligned} \quad (1)$$

Z rovnice (2) jsou získány dva obvody kružnic, které musí kola robota opsat, aby se dostala do počátečního bodu. Výsledkem odvození je získat poměr mezi rychlostmi motorů v závislosti na poloměru R a pološířce robota. Z tohoto důvodu je možno zkrátit všechny konstanty a transformovat rovnici (1):

$$\begin{aligned} o_1 &= v_1 \times t \\ o_2 &= v_2 \times t. \end{aligned} \quad (2)$$

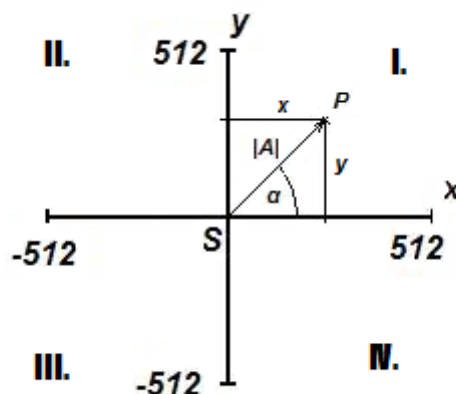
Vztah (2) předpokládá, že motory s různými rychlostmi ujedou své dráhy za stejný čas. Protože má čas v těchto dvou rovnicích charakter konstanty, je v rovnicích zkrácen. Poté lze vytvořit poměr rychlostí v závislosti na poměru obvodů kružnic, vztah (3):

$$\frac{v_2}{v_1} = \frac{o_2}{o_1}. \quad (3)$$

Vztah (3) je upraven pomocí rovnice (1), po vykrácení konstant je získán vztah (4):

$$\frac{v_2}{v_1} = \frac{R - a}{R + a}. \quad (4)$$

Z výsledného vztahu (4) je zapotřebí získat poloměr R . Poloměr obdržíme prostřednictvím matematických nástrojů z aktuální polohy joysticku. Bod určený v kartézské soustavě je pomocí matematických funkcí převeden do polárního souřadnicového systému. Bod je po převedení určen pozicí osy x a y a úhlem α (viz obrázek 3.4). Pro výpočet rychlostí motorů, je důležité znát velikost úsečky $|SP|$.



Obrázek 3.4: Souřadnicový systém s aktuální pozicí joysticku v bodě P .

Úhel je získán pomocí cyklometrické funkce arkus tangens, vztah (5) popisuje výpočet úhlu pomocí osy x a y.

$$\alpha = \operatorname{arctg} \frac{y}{x}. \quad (5)$$

Poloměr otáčení R je vypočítán pomocí vztahu (6):

$$R = \tan \alpha \quad (6)$$

Pro pohyb robota po kružnici s poloměrem R je důležité zjistit, jakou rychlostí pojedou motory. Pro výpočet rychlostí motorů je důležité určit velikost vektoru, určeného úsečkou |SP|. Každá úsečka má počátek ve středu S[0,0] a konec v bodě P obrázek 3.4. Velikost úsečky, je určena pomocí vztahu (7):

$$|A| = \sqrt{x^2 + y^2}. \quad (7)$$

Vypočtené hodnoty poloměru R a rychlosti motorů, stačí pouze pro jízdu v prvním kvadrantu souřadnicového pole. Aby se mohl robot pohybovat ve všech kvadrantech, tedy po celém souřadnicovém poli joysticku, je zapotřebí rozpoznat, ve kterém kvadrantu se páka joysticku nachází. Kvadranty jsou na obrázku 3.4 očíslované proti směru hodinových ručiček.

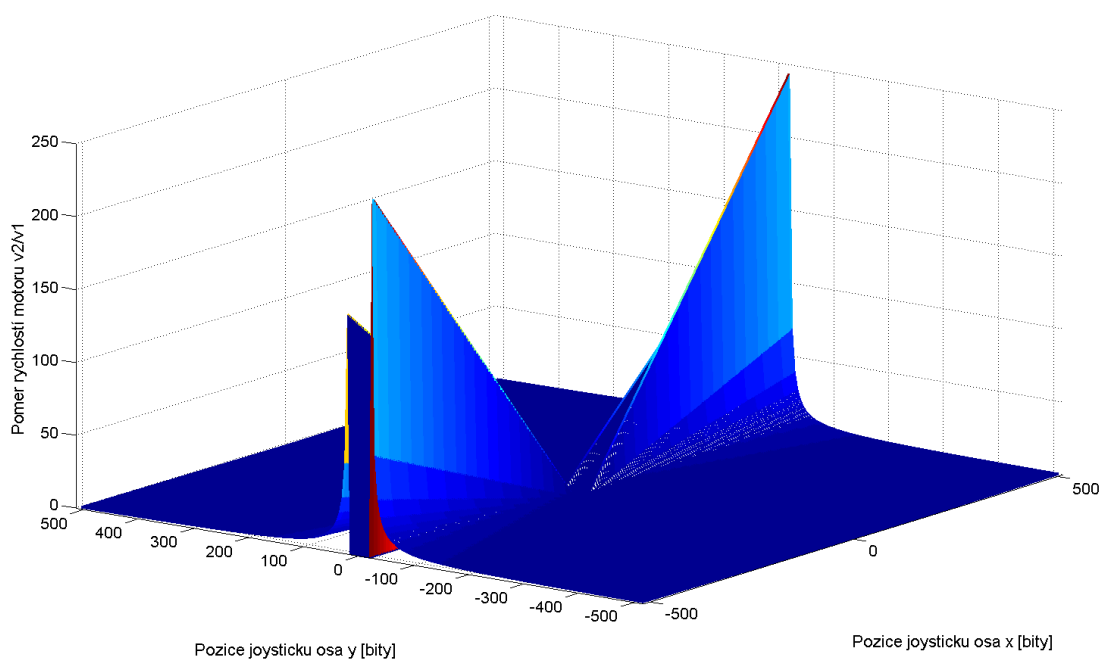
Nyní je pohyb robota určen poloměrem otáčení R, rychlostmi obou motorů a orientací v souřadnicovém systému pomocí kvadrantů. Dalším úkolem, který musí robot zvládnout je jízda dopředu a dozadu. Určení jízdy vpřed a vzad je jednoduché a vychází ze znalosti kvadrantu. Robot se pohybuje dopředu ve kvadrantech I a II a dozadu ve kvadrantech III a IV. Vztah (4) počítá pouze s poměry dvou rychlostí, není v něm zahrnut případ, kdy se poloměr nachází ve středu robota nebo jeho blízkosti.

Z tohoto důvodu jsou stanoveny dvě kritéria, jestli je $R >$ šířka robota, nebo je $R <$ šířka robota. Pokud platí první podmínka, pro poměr rychlostí platí vztah (4), jestliže je splněno druhé kritérium, platí pro poměr rychlostí vztah (8):

$$\frac{v_2}{v_1} = \frac{R}{a} - 1. \quad (8)$$

V praxi to znamená, že se bude robot otáčet kolem své osy. Aby se mohl robot otáčet kolem své osy, musí být jasné, která kola se budou pohybovat vpřed a která vzad. V závislosti na kvadrantu, ve kterém se joystick nachází, je určeno, zda se bude robot otáčet po směru, nebo proti směru hodinových ručiček. Jestliže se joystick nachází v prvním kvadrantu s poloměrem otáčení blízkým pološířce robota nebo dokonce ve středu robota, bude se levá strana motorů pohybovat vpřed a pravá strana motorů vzad. Robot se bude otáčet na místě. Pro ostatní kvadranty je situace obdobná.

Trojrozměrný graf byl vytvořen programem MATLAB R2007b (viz obrázek 3.6) a popisuje všechny akce joysticku, které jsou potřebné k řízení robota. V grafu jsou přímky, které znázorňují tah motorů. Dále jsou v grafu zobrazeny poloměry zatáčení R a je zde také vyznačena tzv. mrtvá zóna, ve které se robot nepohybuje. Rozdílné výšky píku (maximální rychlosti motorů) jsou ve skutečnosti stejné. Tento nedostatek je způsoben hrubým vykreslením grafu.



Obrázek 3.6: Grafický model řízení robota.

3.1.3 Živé řízení robota

Pro řízení robota byly získány všechny potřebné veličiny, které se budou podle upravené teorie podílet na řízení robota. Závěrečná fáze obsahuje zápis hodnot do příslušného registru časovače pro generování PWM signálu. Celý zdrojový kód obsahuje více než 150 řádků, které popisují řízení robota, a do kterých je přepsána teorie řízení. Z tohoto důvodu je uvedena jenom část zdrojového kódu. Celý zdrojový kód je přiložen na CD, které je součástí této práce.

Každý motor, pro který je generován PWM signál, je ovládán jedním speciálním registrem časovače, ve kterém může být pouze jedna hodnota, tedy údaj o rychlosti pro daný motor. Motory se podle teorie pohybují v tandemu, proto stačí zapsat pouze dvě rychlosti. Rychlost pro pravou stranu motorů a pro levou stranu motorů. Výsledné rychlosti levé a pravé strany motorů (TL a TP) jsou přetypovány na celočíselnou hodnotu a vynásobeny periodou časovače. Poslední čtyři řádky zapisují rychlosti motorů do registrů časovače. Časovač poté vygeneruje příslušný PWM signál pro motory.

```
TL=(uint16_t)(tah1*10000);  
TP=(uint16_t)(tahp*10000);  
TIM2->CCR1=TL;  
TIM2->CCR2=TL;  
TIM2->CCR3=TP;  
TIM2->CCR4=TP;
```

Obrázek 3.7: Výpočet rychlostí pro levý a pravý motor a zápis do registrů časovače.

Závěr

Cílem bakalářské práce bylo navrhnout a sestavit čtyřkolového dálkově řízeného mobilního robota. Práce je rozdělena do třech hlavních částí.

První část se zabývá hardwarem, přesněji řečeno návrhem všech elektronických částí, které jsou nezbytné pro chod robota a které nebyly zakoupeny jako elektronický celek. Prvním problémem byla volba správných H-můstků. H-můstky v integrovaných obvodech nesplňovaly stanovené požadavky, bylo nutné umístit, navrhnout a sestavit nové H-můstky pro řízení stejnosměrných motorů. Dále bylo zapotřebí uchytit na robota platformu STM32F4 Discovery. Za tímto účelem byla vytvořena základní deska, která obsahuje elektroniku potřebnou k chodu platformy a konektory pro její upevnění. K základní desce byl později připevněn jeden ze dvou bezdrátových XBee modulů. XBee moduly zajišťují komunikaci mezi obsluhou a robotem. Jelikož mají moduly nestandardní konektorové připojení, byly zakoupeny XBee Boardy. XBee Boardy obsahovaly elektroniku pro práci XBee modulů a konektory pro jejich upevnění. Ukázalo se, že zakoupené XBee Boardy byly nekvalitní a nespolehlivé. Z tohoto důvodu byly navrženy a sestaveny nové XBee Boardy. Důležitým článkem pro řízení robota je jeho ovládání, které je realizováno pomocí joysticku. Joystick byl pro tento účel upraven a doplněn mikrokontrolérem ATmega32A .

Druhá část práce se zabývá nastavením jednotlivých částí do funkčního celku. Je zde uveden převod dat z joysticku a následný přenos dat sériovou linkou do vysílací strany XBee modulu pomocí mikrokontroléru ATmega32A. Dále je v kapitole popsáno připojení XBee modulů k počítači pomocí modulu UM232 a jejich konfigurace programem X-CTU. V práci je uveden princip zpracování přijatých dat z XBee modulů platformou STM32F4 Discovery. Dále se kapitola zabývá ovládáním stejnosměrných motorů a je zde vyřešen problém při ztrátě spojení.

V poslední kapitole je nastíněn a realizován možný přístup k řízení robota. Teorie je založena na řízení robota pomocí joysticku. V kapitole je pomocí vzorců odvozen výchozí vztah, který je důležitý pro řízení robota a pro který bylo nutné získaná data z joysticku zpracovat. Dále byl vytvořen model řízení robota. V závěru kapitoly je uvedeno, jakým způsobem je model řízení v robotovi implementován.

Práce je doplněna množstvím fotografií a dokumentací s příloženými návrhy elektronických schémat a tištěných spojů. Na příloženém CD jsou umístěny zdrojové kódy, fotografie a návrhy schémat, které byly použity v této práci.

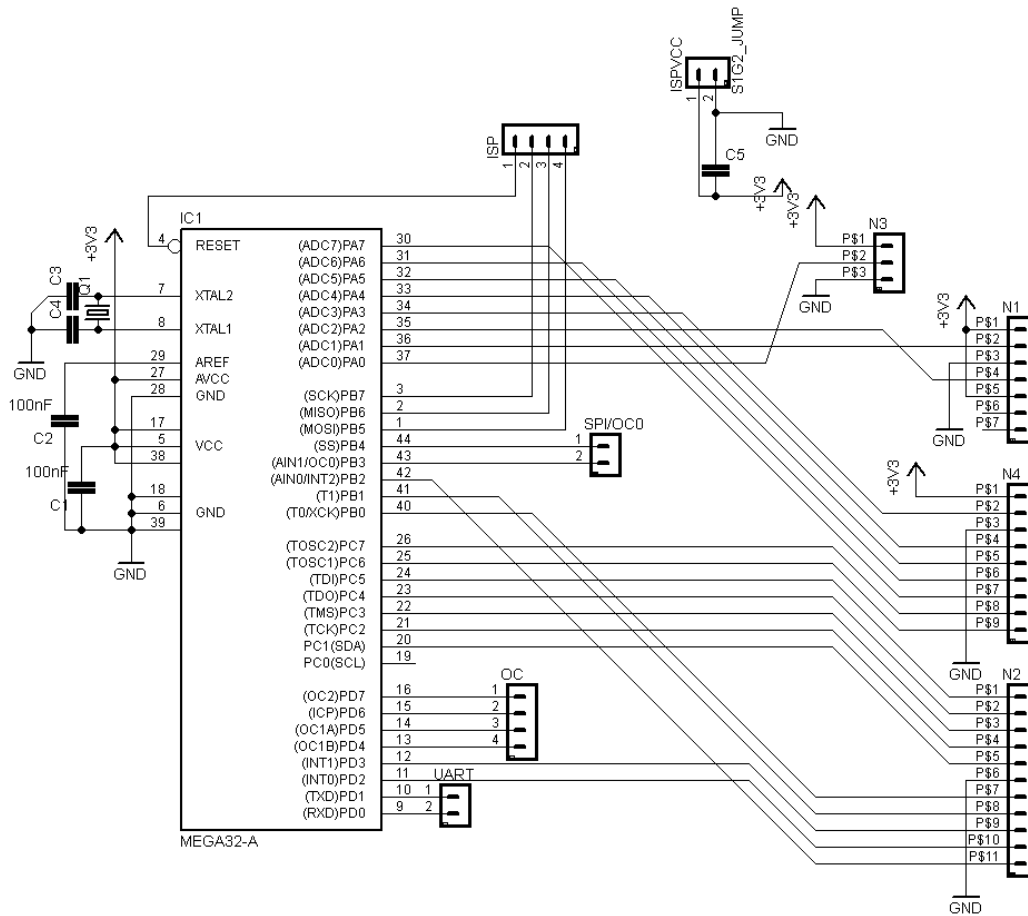
Seznam použitých zdrojů

- [1] KAPILA, Vikram. POLYTECHNIK INSTITUTE OF NYU. *Introduction to Robotics* [online]. New York [cit. 2014-05-10]. Dostupné z: http://mechatronics.poly.edu/smart/Archive/intro_to_rob/Intro2Robotics.pdf
- [2] LIMA, Pedro a Maria Isabel RIBERIO. INSTITUTO SUPERIOR TÉCNICO. *MOBILE ROBOTICS* [online]. Lisabon, 2002 [cit. 2014-05-10]. Dostupné z: <http://users.isr.ist.utl.pt/~mir/cadeiras/robmoveel/Introduction.pdf>
- [3] BRÄUNL. THE UNIVERSITY OF WESTERN AUSTRALIA. *Embedded Robotics* [online]. Australia: Springer-Verlag Berlin Heidelberg, 2006 [cit. 2014-05-10]. ISBN 10-3-540-034361. Dostupné z: <http://newplans.net/RDB/Embedded%20Robotics%20-%20Thomas%20Braunl.pdf>
- [4] DIGI INTERNATIONAL. *XBee®/XBee-PRO® RF Modules* [online]. 2009 [cit. 2014-05-10]. Dostupné z: <https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Datasheet.pdf>
- [5] *UM1472* [online]. 2014 [cit. 2014-05-10]. Dostupné z: http://www.st.com/st-web-ui/static/active/en/resource/technical/document/user_manual/DM00039084.pdf
- [6] MARTIN, Trevor. HITECH. *The Insider's Guide To The STM32* [online]. 2008 [cit. 2014-05-10]. ISBN 0-9549988 8. Dostupné z: http://www.gaw.ru/pdf/The_Insiders_Guide_to_the_STM32_ARM-based_Microcontroller_Hitex.pdf
- [7] *STM32F405xx STM32F407xx* [online]. 2013 [cit. 2014-05-10]. Dostupné z: <http://www.st.com/web/en/resource/technical/document/datasheet/DM00037051.pdf>
- [8] ATMEL CORPORATION. *8-bit Microcontroller with 32K Bytes of In-System Programmable Flash* [online]. 2003 [cit. 2014-05-10]. Dostupné z: <http://www.atmel.com/Images/doc1457.pdf>
- [9] FUTURE TECHNOLOGY DEVICES INTERNATIONAL LTD. *UM232R USB - Serial UART Development Module Datasheet* [online]. Glasgow, United Kingdom, 2011 [cit. 2014-05-10]. Dostupné z: http://www.ftdichip.com/Support/Documents/DataSheets/Modules/DS_UM232R.pdf

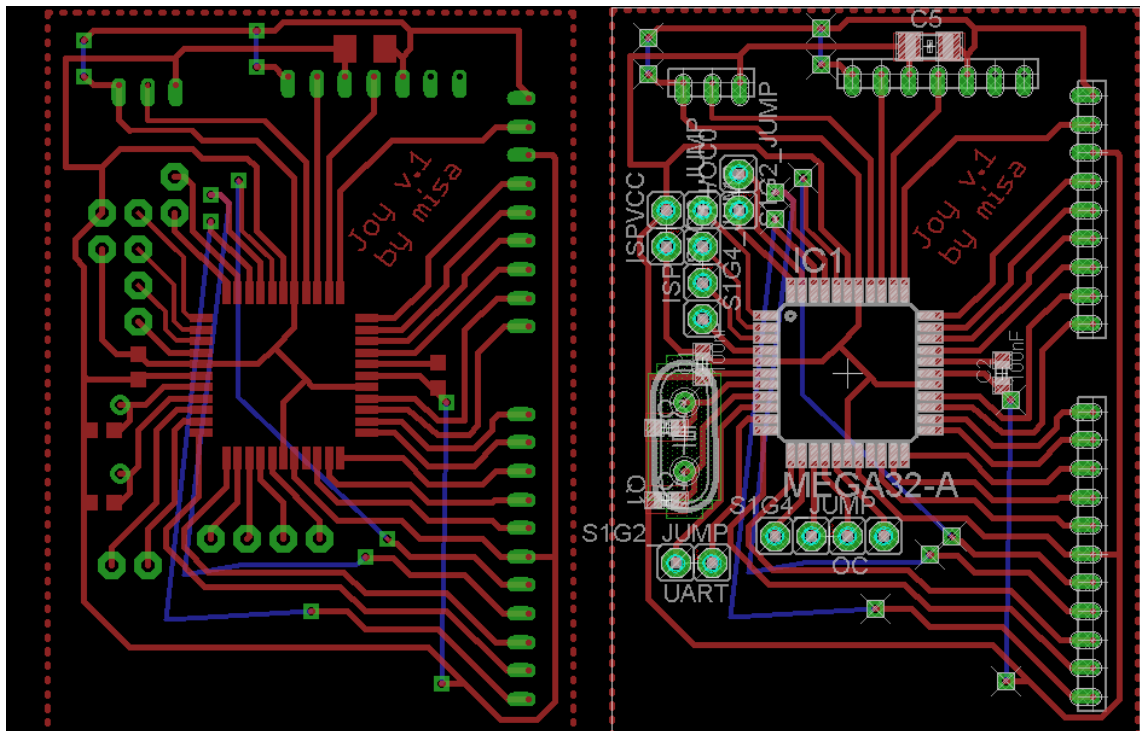
- [10] *X-CTU Configuration & Test Utility Software: User's Guide* [online]. 2008
[cit. 2014-05-10]. Dostupné z:
http://ftp1.digi.com/support/documentation/90001003_A.pdf
- [11] Pulsně šířková modulace. *DH servis* [online]. 2013 [cit. 2014-05-10]. Dostupné z: <http://www.dhservis.cz/psm.htm>
- [12] WINKLER, Zdeněk. Odometrie. *Robotika.cz* [online]. 2005 [cit. 2014-05-10]. Dostupné z: <http://robotika.cz/guide/odometry/en>

Příloha A. Návrh DPS ATmega32A

Příloha obsahuje elektronické schéma, návrh desky plošného spoje pro převod napětí z os joysticku a seznam použitých součástek.



Obrázek A.1: Návrh DPS pro mikrokontrolér ATmega32A a joystick.



Obrázek A.2: Návrh DPS pro mikrokontrolér ATmega32A a joystick.

Seznam použitých součástek

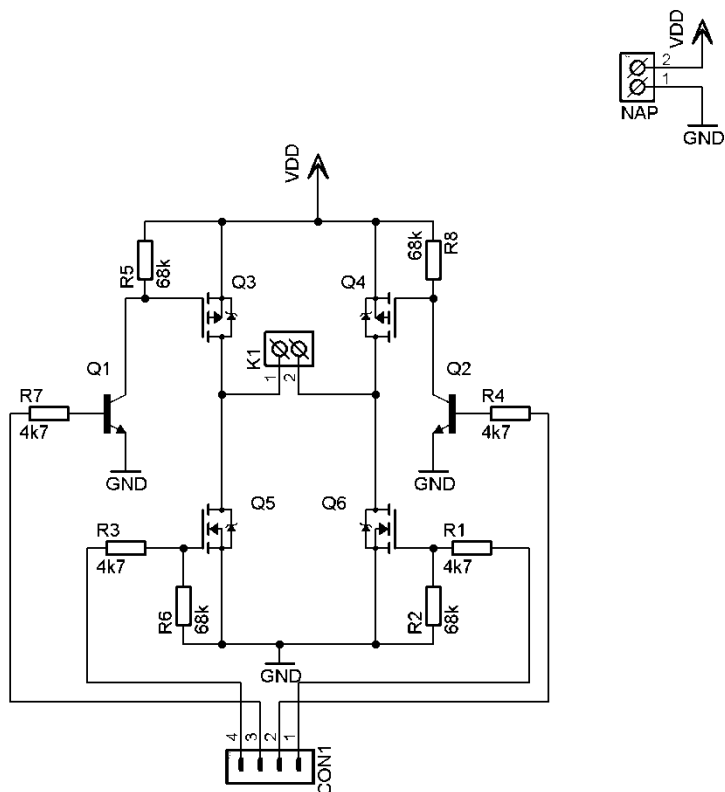
Všechny použité součástky jsou uvedeny v tabulce A.3.

Označení	Popis	Pouzdro
IC1	Atmega32A	TQFP 44A
Q1	Krystal 8MHz	U4
C1	CK0805 100nF	SMD
C2	CK0805 100nF	SMD
C3	CK0805 15pF	SMD
C4	CK0805 15pF	SMD
C5	CK1206 100nF	SMD
UART_JUMP	konektor S1G40	-
OC_JUMP	konektor S1G41	-
ISP_VCC	konektor S1G42	-
SPI/OCO	konektor S1G43	-
ISP	konektor S1G44	-
N1	konektor NX1001-7	-
N2	konektor NX1001-11	-
N3	konektor NX1001-3	-
N4	konektor NX1001-9	-

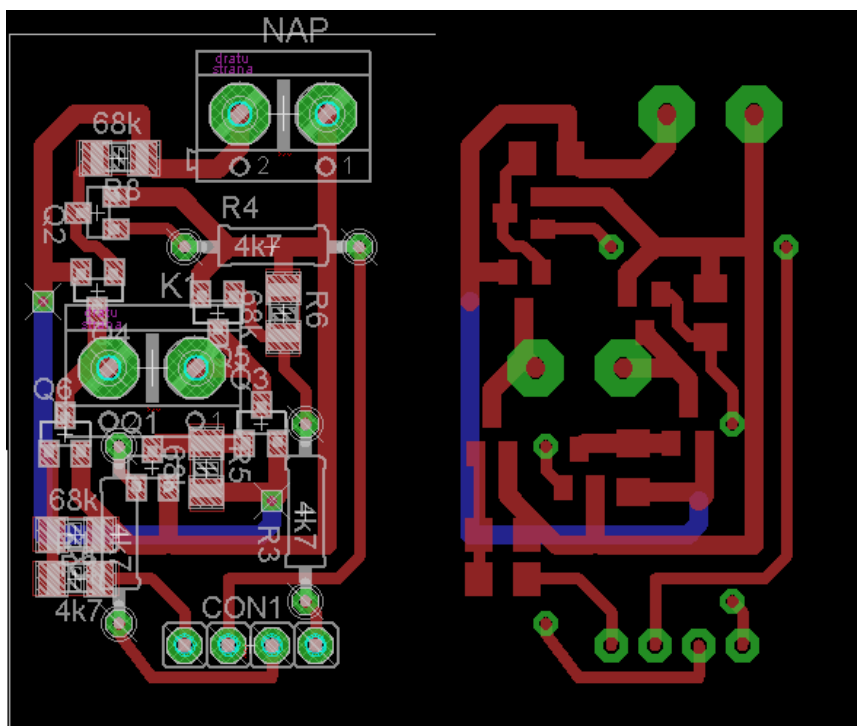
Tabulka A.3: Seznam použitých součástek.

Příloha B. Návrh DPS pro výrobu H-můstků

Příloha obsahuje elektronické schéma, návrh desky plošného spoje H-můstku a seznam použitých součástek. Deska je vytvořena podle schéma na obrázku B. 1.



Obrázek B. 1: Elektronické schéma zapojení H-můstku.



Obrázek B. 2: Návrh DPS pro konstrukci H-můstků.

Seznam použitých součástek

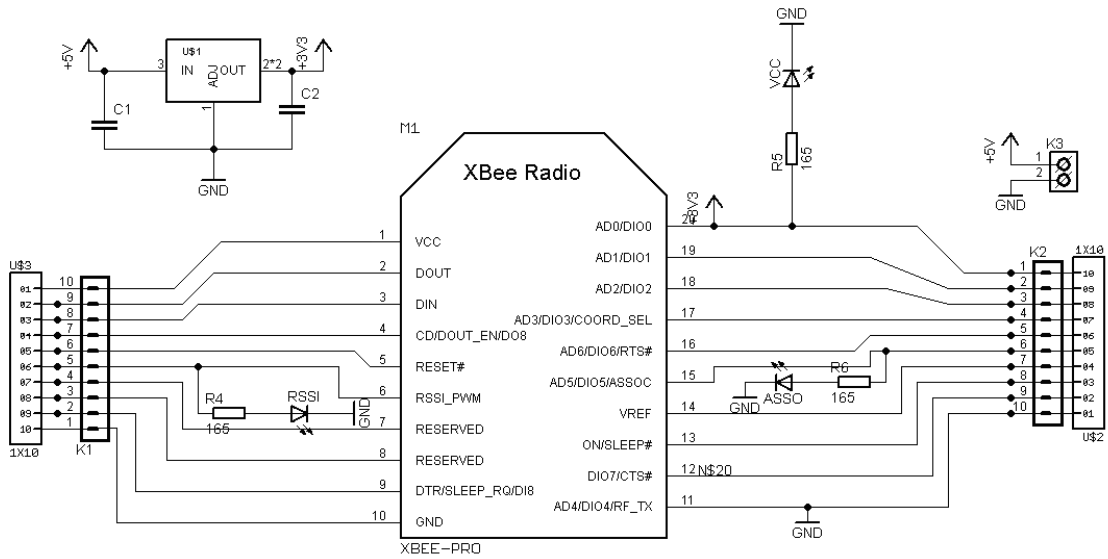
Všechny použité součástky jsou uvedeny v tabulce **B.3**.

Označení	Popis	Pouzdro
Q1	BC817	SMD
Q2	BC817	SMD
Q3	IRLM2505	SMD
Q4	IRLML6402	SMD
Q5	IRLM2505	SMD
Q6	IRLML6402	SMD
CON1	Konektor S1G40	-
NAP	ARK500/2S	-
R1	4k7	THT
R2	68k	SMD
R3	4k7	THT
R4	68k	SMD
R5	68k	SMD
R6	68k	SMD
R7	4k7	THT

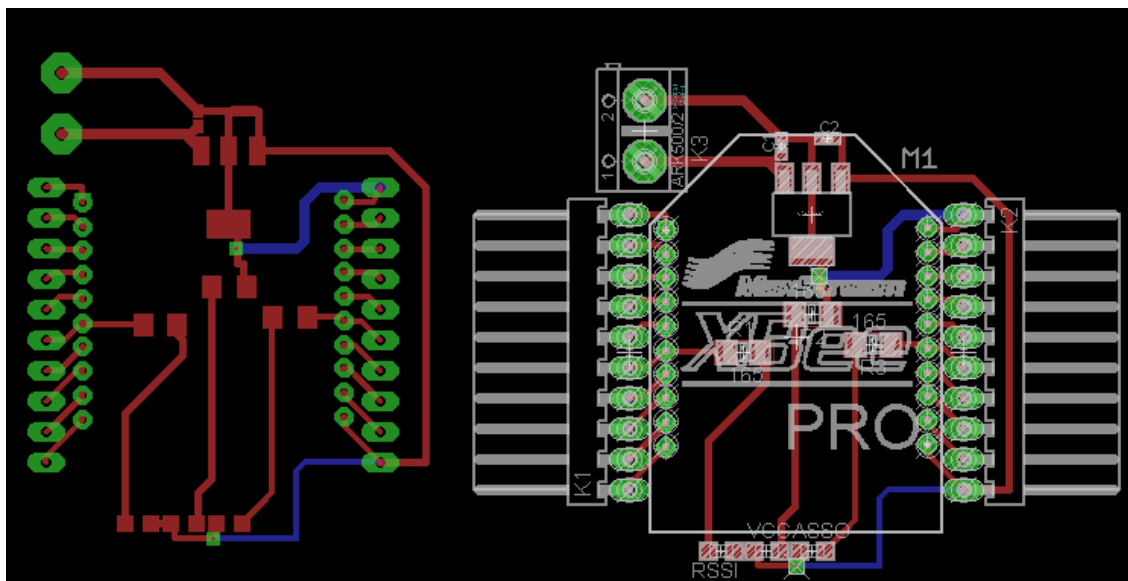
Tabulka B. 3: Seznam použitých součástek.

Příloha C. Návrh DPS pro výrobu XBee Boardu

Příloha obsahuje elektronické schéma, návrh desky plošného spoje XBee Boardu a seznam použitých součástek. Deska byla vytvořena podle schémata na obrázku C.1.



Obrázek C. 1: Elektronické schéma XBee Boardu.



Obrázek C. 2: Návrh DPS pro výrobu XBee Boardu.

Seznam použitých součástek

Všechny použité součástky jsou uvedeny v tabulce C.3.

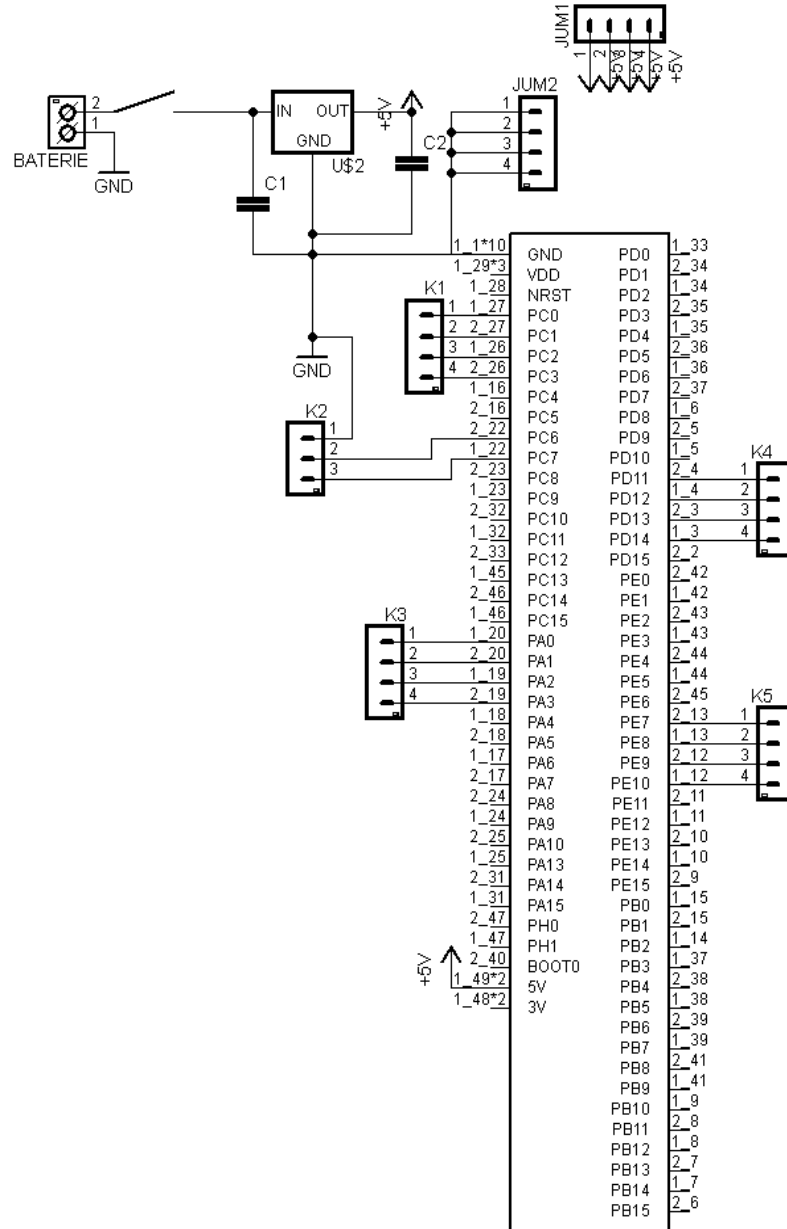
Označení	Popis	Pouzdro
K1	konektor ZL265-10SG	-
K2	konektor ZL265-10SG	-
K3	konektor S1G40	-
U1	LM1117	-
U2	konektor S1G40	-
U3	konektor S1G40	-
R4	165	SMD
R5	165	SMD
R6	165	SMD
RSSI	LED KA3528	SMD
VCC	LED KA3528	SMD
ASSO	LED KA3528	SMD
C1	CK0805 100nF	SMD
C2	CK0805 10nF	SMD
M1	XBee-PRO	-

Tabulka C.3: Seznam použitých součástek.

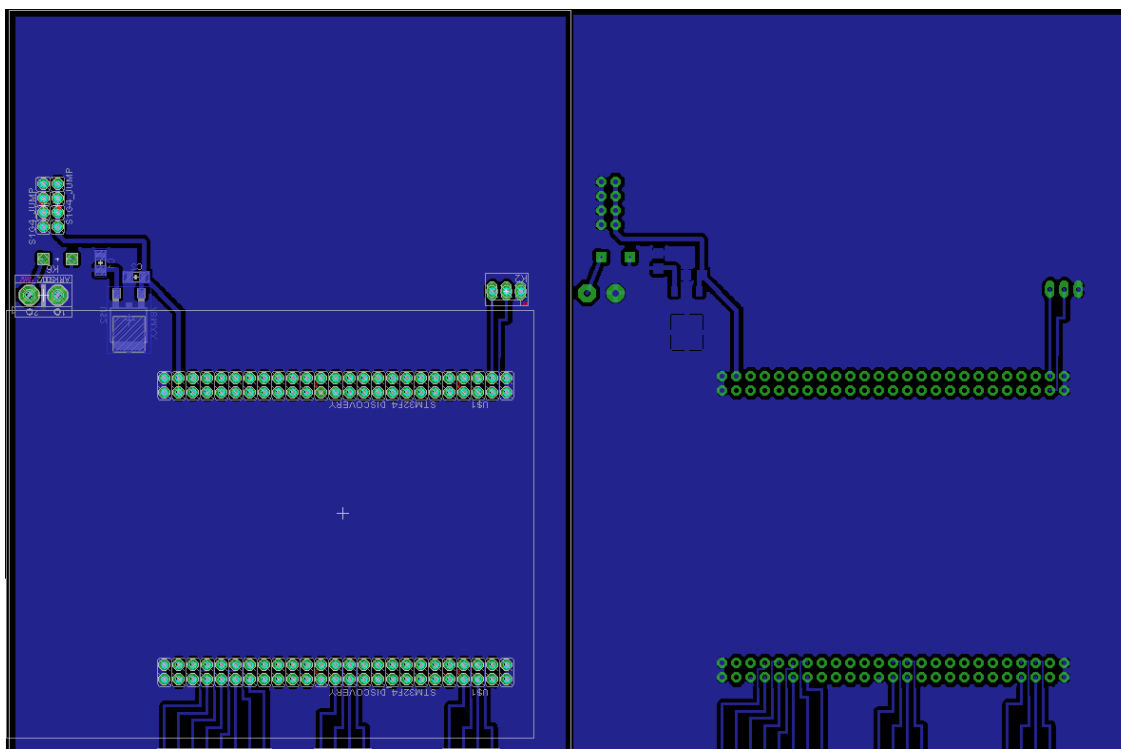
Příloha D. Návrh DPS pro výrobu základní desky STM32F4 Discovery

Discovery

Příloha obsahuje návrh základní desky plošného spoje, který je určen pro uchycení platformy STM32F4 Discovery, s vypínačem.



Obrázek D. 1: Schéma zapojení základní desky pro STM32F4 Discovery.



Obrázek D. 2: Návrh DPS pro výrobu základní desky STM32F4 Discovery.

Seznam použitých součástek

Všechny použité součástky jsou uvedeny v tabulce D.3.

Označení	Popis	Pouzdro
U2	LM1117	SOT-223
C1	CK0805 10uF	SMD
C2	CK0805 22uF	SMD
Baterie	Spínač 2 polohy SSK3A	-

Tabulka D. 3: Seznam použitých součástek.

Příloha E. Elektronická příloha

Příložené CD obsahuje tyto soubory.

Nákresy

- Návrhy elektronických schémat a desek ve formátu EAGLE

Programy

- Projekty a programy napsané v jazyce C
- Programy napsané v softwaru MATLAB

Fotodokumentace

- Stavba robota

Bakalářská práce

- Práce ve formátu PDF
- Použité obrázky ve formátu PNG