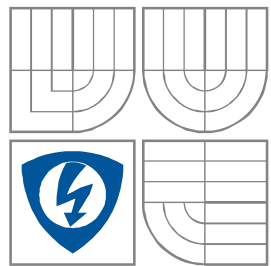


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

ŘÍDICÍ SYSTÉM PRO MALÝ MOBILNÍ ROBOT

CONTROL SYSTEM FOR A SMALL MOBILE ROBOT

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

VEDOUCÍ PRÁCE

SUPERVISOR

Bc. IMRICH SÁRKÖZY

doc. Ing. LUDĚK ŽALUD, Ph.D.

BRNO 2011



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Diplomová práce

magisterský navazující studijní obor
Kybernetika, automatizace a měření

Student: Bc. Imrich Sárközy

Ročník: 2

ID: 98030

Akademický rok: 2010/11

NÁZEV TÉMATU:

Řídicí systém pro malý mobilní robot

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte mikroprocesorový řídicí systém pro mobilní platformu na bázi procesoru Atmel xMega. Navrhněte schéma systému, rozšířte systém o snímače neelektrických veličin. Navržený systém realizujte a po konzultaci s vedoucím práce ověřte jeho funkci.

DOPORUČENÁ LITERATURA:

H.R. Everett, Sensors for Mobile Robots - Theory and Application, A K Peters, 1995, ISBN 1-56881-048-2

Termín zadání: 7.2.2011

Termín odevzdání: 23.5.2011

Vedoucí práce: doc. Ing. Luděk Žalud, Ph.D.

Konzultanti diplomové práce:


prof. Ing. Pavel Jura, CSc.
předseda oborové rady



UPOZORNĚNÍ:

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

Abstrakt

Cílem projektu bylo navrhnout a realizovat obvod s mikrokontrolérom XMega, který bude mít za úkol řídit malého mobilního robota s pásovým podvozkem.

První část projektu zahrnovala návrh schématu řídicí desky, zkonstruování a odladění DPS.

V další fázi se připojili snímače neelektrických veličin a navrhli se algoritmy pro jednoduchou navigaci. V poslední fázi se ladilo uživatelské rozhraní a nastavoval co nejmenší datový tok mezi mobilní platformou a řídicím počítačem.

Klíčová slova

AVR, XMega16 A4, PDI, RS 232, USART, řídicí systém, mobilní platforma, diferenciální podvozek, mikroprocesor, SRF08, SRF02, SRF10, ACC 7260, CMPS03, ADNS-5030, Sabertooth, pohon, mřížka obsazenosti

Abstract

The aim of this work was to design a control system based on the XMega microcontroller, for a small mobile robot with an tracked undercarriage.

At first, my work was to develop an electrical scheme and circuit boards for the control system.

The next step was to connect sensors for non electric quantity and modify the control algorithms for a simple navigation. In the last phase I tuned the user interface and the whole control system for minimal data flow between the mobile platform and the user interface.

Keywords

AVR, XMega16 A4, PDI, RS 232, USART, control system, mobile platform, differential gear, microprocessor, control, SRF08, SRF02, SRF10, ACC 7260, CMPS03, ADNS-5030, Sabertooth, actuator, Occupancy grid

Bibliografická citace:

SÁRKÖZY, I. Řídicí systém pro malý mobilní robot. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 71 s. Vedoucí diplomové práce doc. Ing. Luděk Žalud, Ph.D.

Prohlášení

„Prohlašuji, že svou diplomovou práci na téma Řídicí systém pro malý mobilní robot jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **20. května 2011**

.....

podpis autora

Poděkování

Děkuji vedoucímu diplomové práce doc. Ing. Ludřkovi Žaludovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne: **20. května 2011**

.....
podpis autora

Obsah

| | | |
|-------|--|----|
| 1 | Úvod..... | 9 |
| 2 | Seznámení s mikrokontroléry..... | 10 |
| 2.1 | ATMEL AVR XMega16 A4 | 11 |
| 2.2 | USART | 13 |
| 2.3 | A/D převodník mikrokontroléru [5] | 15 |
| 2.4 | TWI mikrokontroléru (I ² C) [2]..... | 18 |
| 2.5 | SPI mikrokontroléru [4]..... | 20 |
| 3 | Popis a realizace | 21 |
| 3.1 | Blokové schéma řídicího systému | 21 |
| 3.2 | Rozhraní RS 232..... | 21 |
| 3.3 | Převodník úrovní pro sběrnici I ² C | 23 |
| 3.4 | Pohon robota..... | 24 |
| 3.5 | Sabertooth 2x10..... | 24 |
| 3.6 | Snímače neelektrických veličin | 26 |
| 3.6.1 | Ultrazvukové snímače vzdálenosti typu SRFxx..... | 26 |
| 3.6.2 | Modul kompasu CMPS03 | 30 |
| 3.6.3 | Trojosý akcelerometr ACC7260 | 32 |
| 3.6.4 | Modul pro odometrii podvozku..... | 34 |
| 3.7 | Rozložení snímačů..... | 37 |
| 3.8 | Výpočet naklonění | 38 |
| 3.9 | Programátor USBTiny MkII..... | 40 |
| 3.10 | Návrh DPS řídicí jednotky..... | 41 |
| 3.11 | Napájení obvodů..... | 43 |
| 3.12 | Popis zdrojového kódu mikrokontroléru | 43 |
| 3.13 | Komunikační protokol..... | 48 |
| 3.14 | Mřížka obsazenosti | 49 |
| 3.15 | Popis řídicího programu REMOTE | 51 |
| 4 | Závěr..... | 55 |

1 ÚVOD

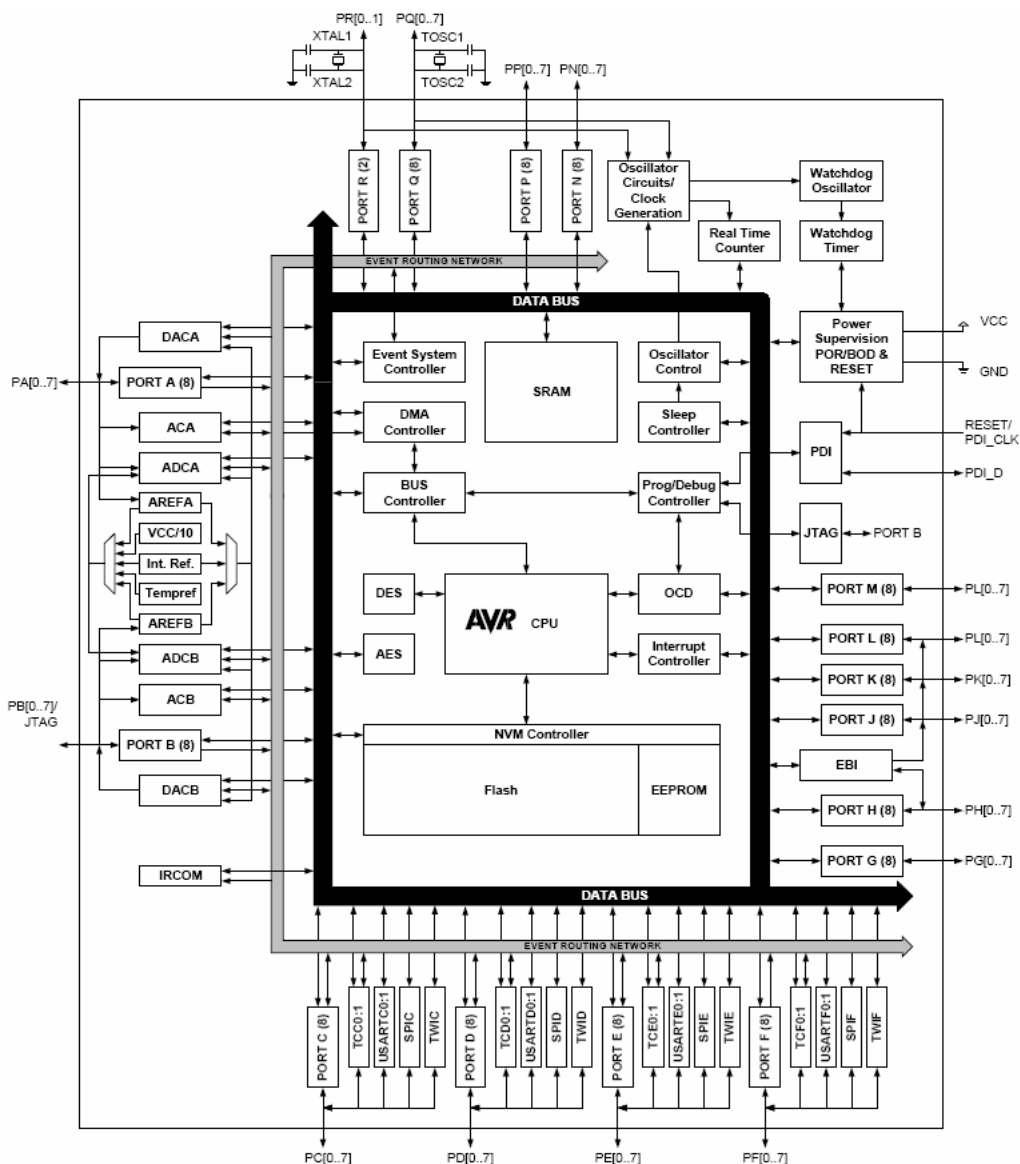
Cílem této diplomové práce bylo seznámit se s mikrokontroléry ATMEL AVR XMEGA16 A4, vytvořit návrh řídicího systému pro malého mobilního robota s pásovým podvozkem – navrhnout schéma zapojení a desku plošných spojů v CAD systému EAGLE. Projekt dále zahrnoval konstrukci desky plošných spojů, její osazení vhodnými součástkami a odladění řídicí desky. Dále jsem navrhnul řídicí algoritmy a procedury tohoto mobilního robota pro jednoduchou navigaci uvnitř budov.

Řídicí systém slouží k ovládání výkonového člena pro čtyři stejnosměrné motory - pohon mobilní platformy, zpracovává naměřené údaje z ultrazvukových snímačů, akcelerometru, kompasu a odometrie. Celý řídicí systém je ovládaný aplikací REMOTE z operátorského počítače, která zpracovává a vyhodnocuje naměřené údaje ze snímačů neelektrických veličin. Vstupním ovládacím prvkem aplikace je joystick a výstupem aplikace je vytvořená mapa prostředí. Komunikace mezi malým mobilním robotem a řídicím počítačem je bezdrátová. V době realizace byl jako bezdrátová komunikační jednotka využit Bluetooth modul. Konstrukce je však navržena tak, že daný Bluetooth modul je snadno nahraditelný jiným modulem, například ZigBee.

2 SEZNÁMENÍ S MIKROKONTROLÉRY

Mikroprocesor je poměrně složitý číslicový obvod, který za pomoci nahreného programu vykonává sled aritmetických a logických operací. Mikrokontrolér (jednočipový mikrořadič) představuje spojení mikroprocesoru s pamětí a obvody rozhraní. Čip je opatřen příslušnými vývody, které se mohou používat k přímému řízení připojených zařízení.

Jádem celé řídicí jednotky je mikrokontrolér řady ATMEL AVR. Jsou to nízkopříkonové 8 bitové mikrokontroléry založené na AVR RISC architektuře.



Obr.2.1: Blokové schéma mikrokontroléru X Mega [1]

2.1 ATMEL AVR XMega16 A4

Mikrokontroléry řady XMega [1] tvoří novou řadou firmy Atmel. Atmel AVR XMEGA obsahují výkonné 8/16bitové MCU postavené na technologii picoPower druhé generace a disponující unikátním Event Systémem. Od jejich předchůdců AtMega se liší vyšší pracovní frekvencí a větší rozmanitostí svých periférií.

Zajímavou vlastností, kterou rodina XMega disponuje, je tzv. Event Systém[1], který umožňuje meziperiferijní komunikaci bez nutnosti zatěžování CPU nebo DMA. To zajistí stoprocentně předvídatelnou a krátkou dobu odezvy. XMEGA umožňuje použití až osmi simultánních událostí (event) nebo podmínek přerušení na perifériích, které pak automaticky spustí určitou akci (resp. reakci) na perifériích jiných.

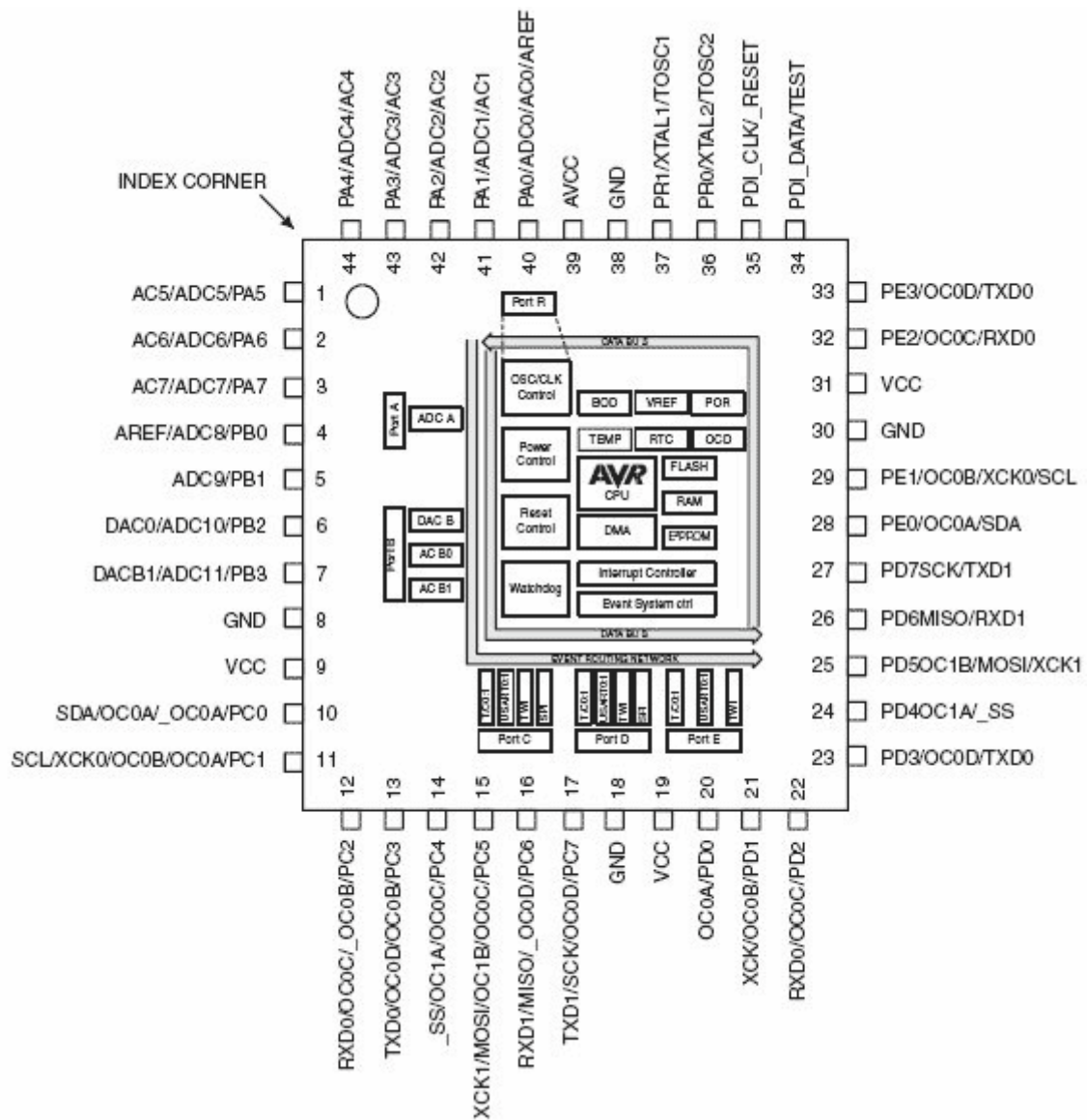
Označování:

| | | |
|--------------------|--|-------------------------------|
| XMega 16 A4 | | |
| jméno | | velikost pouzdra |
| | | velikost interní flash paměti |

Hlavní výhodou daného typu mikrokontroléru je jeho malá spotřeba při vysokém dodaném výpočetním výkonu.

Základní charakteristika [1]:

- pracovní frekvence do 32MHz
- 16 kilo bytů interní FLASH paměti
- 2 kilo byty datové paměti SRAM
- 1 kilo byt paměti EEPROM
- vnitřní oscilátor + možnost připojení externího oscilátoru XTAL
- 4 kanálové DMA
- pětkrát 16 bitový čítač
- pětkrát rozhraní USART
- dvakrát SPI
- dvakrát analogový komparátor
- dvakrát TWI – kompatibilní s I²C a SMBus
- 12 kanálový 12 bitový A/D převodník (2 Msps)
- 2 kanálový 12 bitový D/A převodník (1 Msps)
- RTC32 – 32 bitové hodiny reálného času
- AWeX – pokročilé waveform rozšíření
- Watch – dog Timer
- pracovní napětí 1,6 V až 3,6 V
- podpora vnitřních a externích přerušení

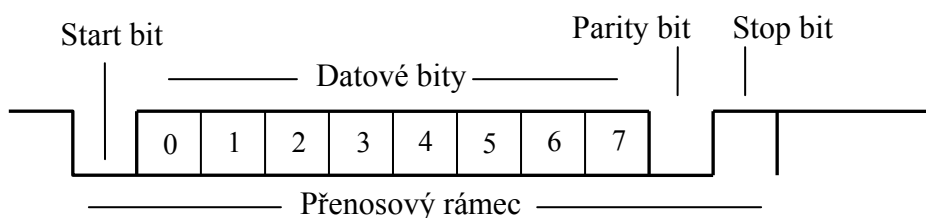


Obr. 2.2: Popis vývodů mikrokontroléru XMEGA16 A4 [1]

2.2 USART

Mikrokontroléry řady AVR obsahují plně duplexní asynchronní přijímač a vysílač pod názvem USART[3]. Vybraný mikrokontrolér obsahuje 5 takových komunikačních jednotek USART, které se budou používat pro bezdrátovou komunikaci a komunikaci s výkonovou ovládací částí (Sabertooth). Základní vlastnosti:

- generátor přenosové rychlosti s velkým počtem nastavitelných přenosových rychlostí
- délka znaku 8 nebo 9 bitů
- filtrace šumu na straně přijímače
- detekce chyby přenosu – ztráta znaku, chyba rámce, falešný start – bit
- přerušení – dokončení vysílání a nebo přijímání, vyprázdnění DATA



Obr. 2.3: Asynchronní sériový přenos

Každý USART mikrokontroléru se řídí sedmi vstupně – výstupními registry[1]:

- DATA – datový registr obsahující přijatou nebo vyslanou hodnotu
- CTRLA – nastavení úrovně přerušení
- CTRLB – řídicí registr, který povoluje příjem a vysílání
- CTRLC – nastavení parity, stop - bitů a velikosti přenášených údajů
- STATUS – stavový registr obsahující stav přenosu
- BAUDCTRLB, BAUDCTRLA – baud kontrol registry – nastavení přenosové rychlosti

Protože jsem se snažil zredukovat datový tok mezi mobilní platformou a řídicím počítačem, zvolil jsem přenosovou rychlost 1200 Baud. Následující výpočet slouží k přesnému určení hodnoty registrů BAUDCTRLB, BAUDCTRLA pro USART, který bude napojen na bezdrátovou komunikační jednotku.

- pro komunikaci se použije 1 start bit, 8 datových a 1 stop bit

$$1200 \text{ Bd} \rightarrow 1200 / (1+8+1) = 120 \rightarrow 120 * 8 = 960 \text{ bit/s}$$

- pro 1200 Bd se přeneše při dané konfiguraci 960 bit/s

Zvolená hodnota $f_{BAUD1} = 1200$ Bd

Asynchronous Normal Speed mode (CLK2X = 0, BSCALE = -2):

$$f_{BAUD} = \frac{f_0}{16 \cdot ((2^{BSCALE} \cdot BSEL) + 1)} \quad (1.1)$$

$$BSEL1 = \frac{1}{2^{BSCALE}} \left(\frac{f_0}{16 \cdot f_{BAUD}} - 1 \right) = \frac{1}{2^{-2}} \left(\frac{16 \cdot 10^6}{16 \cdot 1200} - 1 \right) = 3329$$

f_{BAUD} – přenosová rychlost v Bd
 f_0 – kmitočet oscilátoru mikrokontroléru

Přenosová rychlost mezi mikrokontrolérem a výkonovou částí pro ovládání motorů byla předem definována výrobcem daného modulu Sabertooth na hodnotu:

$f_{BAUD2} = 9600$ Bd

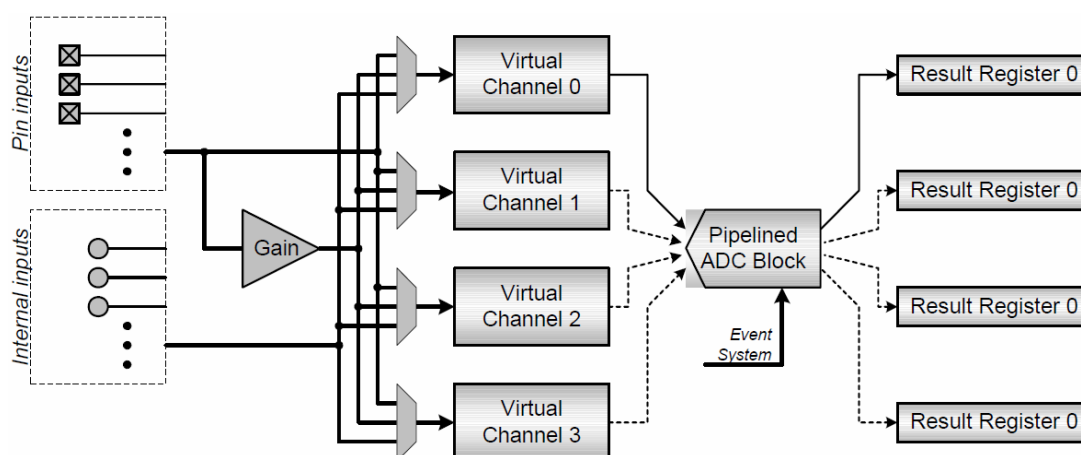
Asynchronous Normal Speed mode (CLK2X = 0, BSCALE = -4):

$$BSEL1 = \frac{1}{2^{BSCALE}} \left(\frac{f_0}{16 \cdot f_{BAUD}} - 1 \right) = \frac{1}{2^{-4}} \left(\frac{16 \cdot 10^6}{16 \cdot 9600} - 1 \right) = 3301$$

f_{BAUD} – přenosová rychlost v Bd
 f_0 – kmitočet oscilátoru mikrokontroléru

2.3 A/D převodník mikrokontroléru [5]

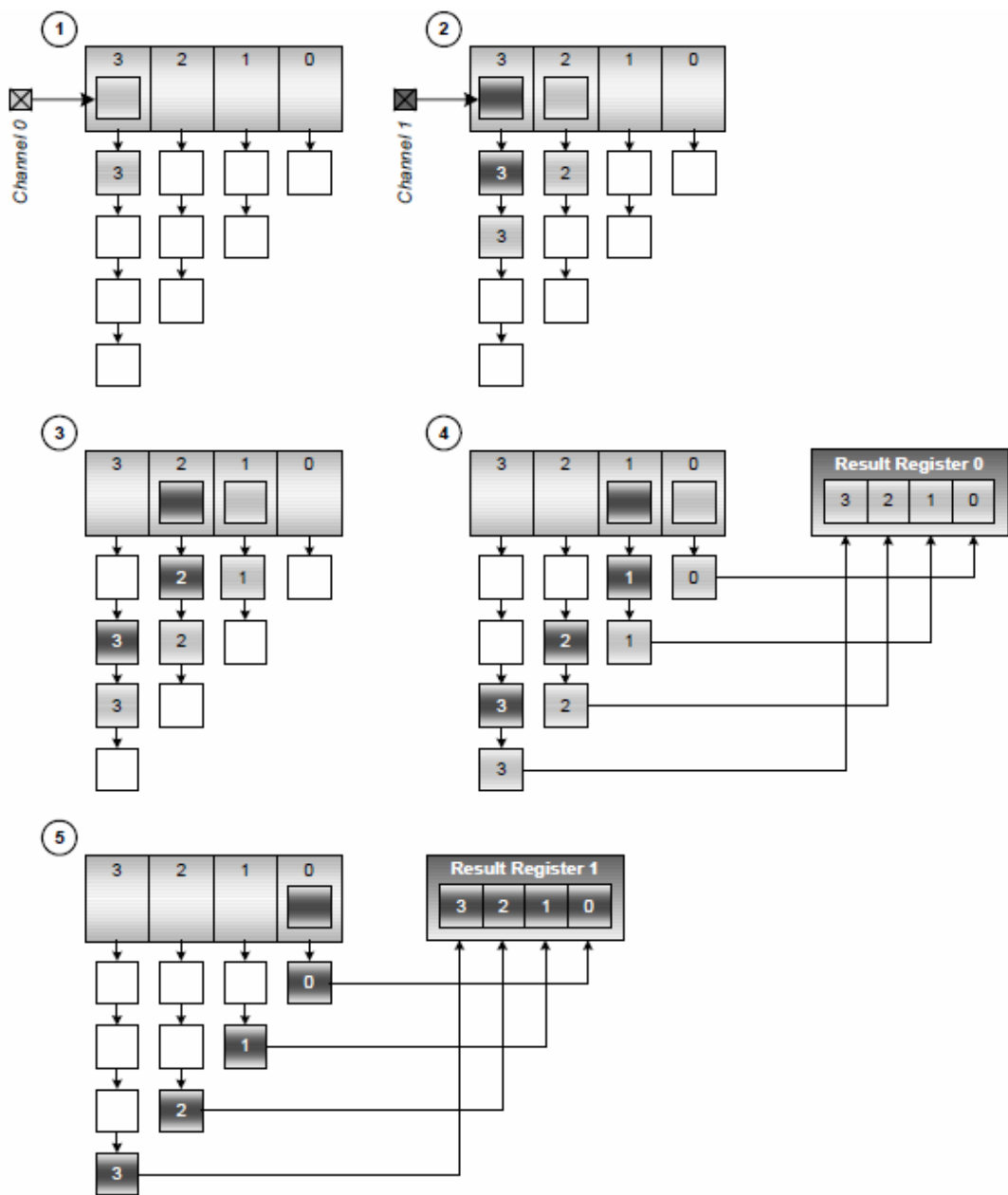
Mikrokontrolér XMega16 A4 obsahuje 12 bitový aproximačný analogovo – digitální převodník[5], který je multiplexovaný na 12 kanálů. Blok A/D převodníku má dvanácti stupňovou pipelined architekturu, která umožňuje vzorkování více signálů paralelně. Čtyři vstupní multiplexory jsou samostatně konfigurovatelné a můžeme si je představit jako virtuální kanály, které se dělí o stejný A/D převodník. Na obrázku 2.4 je znázorněno blokové schéma A/D převodníku použitého v mikrokontroléru XMega.



Obr. 2.4: Blokové schéma A/D převodníku [5]

Výstupy multiplexorů se mohou vzorkovat každé čtyři cykly ADC hodin. Navzorkování jednoho vzorku pro 12 bitový výsledek trvá 7 hodinových cyklů ADC. Navzorkování jednoho bitu pro 8 bitový výsledek trvá 5 hodinových cyklů ADC.

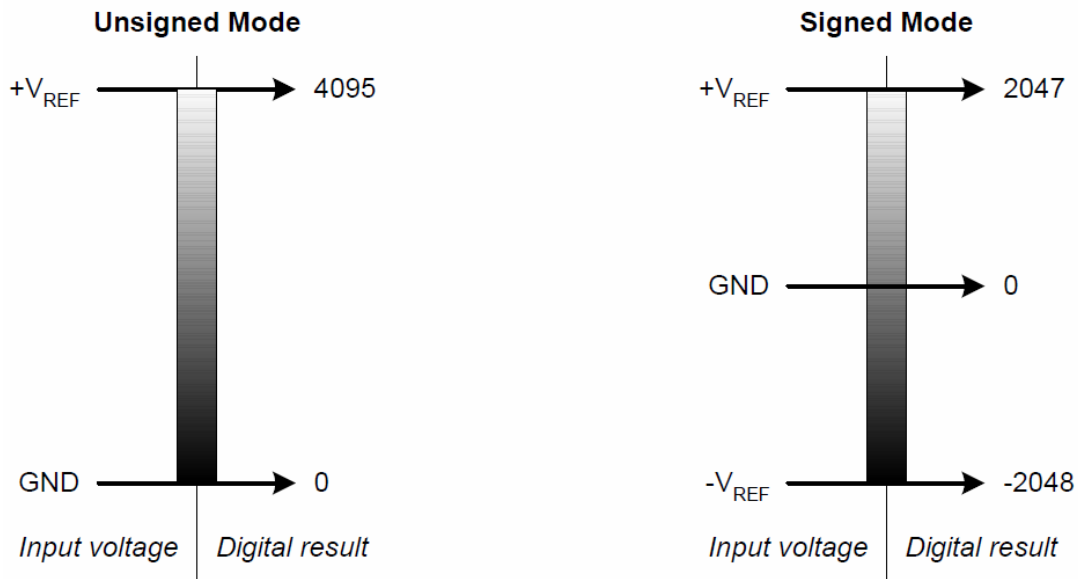
Na obrázku 1.5 je znázorněn čtyř stupňový pipelining [5] (zřetězené zpracování) po dobu převodu dvou vstupních signálů. Diagram vysvětluje, že pokud byl signál navzorkován do pipelinu, tak z první úrovně se uloží MSB signálu 1. V dalším kroku se uloží následující bit z úrovně 2, který reprezentuje bit po MSB signálu 1. V tomto stejném kroku se ale z úrovně 1 uloží i MSB signálu 2.



Obr. 2.5: Pipelining A/D převodníku [5]

Čtyři virtuální kanály mají jeden společný kontrolní registr multiplexoru CHnMUXCTRL, jeden kontrolní registr pro kanál CHnCTRL a jeden pár registrů pro výsledek CHnRESL/CHnRESH.

A/D převodník se může nastavit do dvou módů převodu - do znaménkového nebo bezznaménkového. Obrázek 2.6 popisuje, že pokud se použije bezznaménkový mód, tak bude výsledek navzorkovaný v rozsahu GND až $+V_{ref}$. Ve znaménkovém režimu je však rozsah od $-V_{ref}$ až do $+V_{ref}$.



Obr. 2.6: Znaménkový a bezznaménkový převod A/D [5]

Pokud se jedná o 12 bitový A/D převodník, měla by se z důvodu přesného měření dodržet následující základní pravidla:

- rozlišení 12 bitového A/D převodníku mikrokontroléru XMega:

$$\Delta U_x = \frac{V_{ref}}{N} \quad (1.2)$$

$$\Delta U_x = \frac{3,3}{2^{12}} = 0,08mV$$

- přesnost napájecího napětí

$$|\Delta V_{ref}| = \frac{V_{ref}^2}{N \cdot U_x} \quad (1.3)$$

$$|\Delta V_{ref}| = \frac{3,3^2}{2^{12} \cdot 3,3} = \frac{10,89}{13516,8} = 0,08mV$$

$$V_{ref} = U_{vcc} = 3,3V \rightarrow 0,08mv \text{ je } 0,002 \% U_{vcc}$$

- kvantizační šum
 $\pm \frac{1}{2} \text{ LSB} \rightarrow \frac{1}{2} \cdot 0,08 = 0,04mV$
- filtrace napájecího napětí A/D převodníku:

$$f_{cut} = \frac{1}{2\pi\sqrt{LC}} \quad (1.4)$$

Použité součástky pro filtraci napájecího napětí A/D převodníku:

$$L = 10\mu\text{H}$$

$$C_{10} = 20\text{pF}$$

$$C_{11} = 1\text{nF}$$

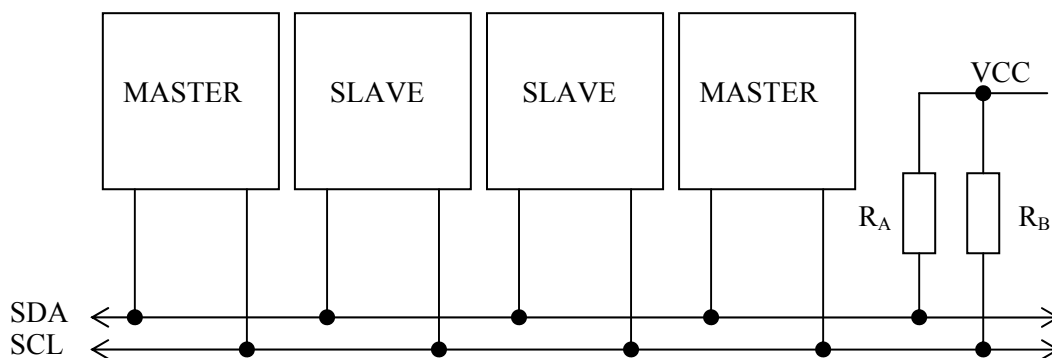
$$C_{12} = 100\text{nF}$$

$$f_{cut} = \frac{1}{2\pi\sqrt{10 \cdot 10^{-6} \cdot 101,02 \cdot 10^{-9}}} = 158349\text{Hz}$$

2.4 TWI mikrokontroléru (I²C) [2]

TWI dvou vodičové sériové rozhraní je obdobou známého I²C od firmy Philips. Jedná se o dvojvodičové datové přepojení mezi jedním nebo několika zařízeními (MASTER) a periferními zařízeními (SLAVE). Zařízení připojená na stejnou sběrnici jsou rozeznávána specifickými adresami. Adresa i data se posílají po stejném vodiči. Existuje celá řada zařízení, která podporuje tento protokol. Jedná se zejména o paměti, A/D, D/A převodníky, budiče displejů, integrované obvody pro televizní a audio techniku a mnoho dalších.

Sběrnice I²C používá sériovou datovou linku SDA a linku hodinového signálu SCL. Data i adresy se přenášejí podobně jako v posuvných registrech, společně s hodinovými impulzy. Následující obrázek vysvětluje princip přepojení sběrnice I²C. Jedná se o multimaster – multislave systém.



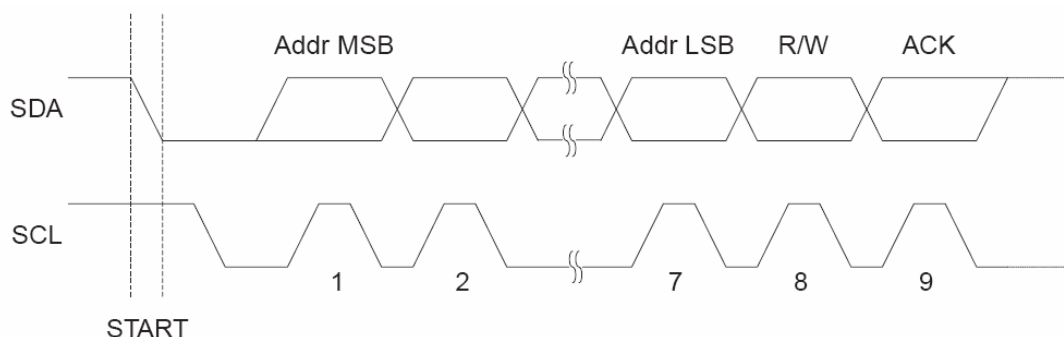
Obr. 2.7: Přepojení sběrnice I²C

Protokol I²C rozeznává celou řadu přesně definovaných situací, které každému účastníkovi umožňují rozeznat začátek a konec přenosu:

- Klidový stav – SDA i SCL jsou prostřednictvím pul – up rezistorů RA a RB na vysoké úrovni a tím nejsou aktivní.
- Podmínka startu – SDA je masterem stažena na nízkou úroveň, zatím co SCL zůstává na vysoké úrovni.
- Podmínka pro stop – SDA přejde z nízké úrovně na vysokou úroveň a SCL zůstává na vysoké úrovni.
- Přenos údajů – příslušný vysílač přivede na datovou linku SDA osm datových bitů, které jsou hodinovými impulzy SCL, vysílaných masterem, posouvané. Přenos začíná bitem s nejvyšší vahou.
- Potvrzení – příslušný přijímač potvrzuje příjem bytu s nízkou úrovní na SDA, dokud master nevyšle devátý hodinový impulz na SCL. Potvrzení současně znamená, že se má přijímat další byt. Požadované ukončení přenosu se musí ohlásit neexistencí potvrzení.

Adresa se přenáší stejně jako údaje. Nejjednodušším přenosovým postupem je:

- Master pošle start podmínku.
- Pošle byt, kde je adresa obsažena v 7. až 1. bitu a 0 bit je nastaven podle směru přenosu – 0 je zápis a 1 čtení.
- Slave zařízení adresu potvrdí.
- Master vyšle údajový byt, který bude opět potvrzen.
- Master může přenos ukončit stop podmínkou nebo může tomuto stejnému slave zařízení poslat další údajový byt.

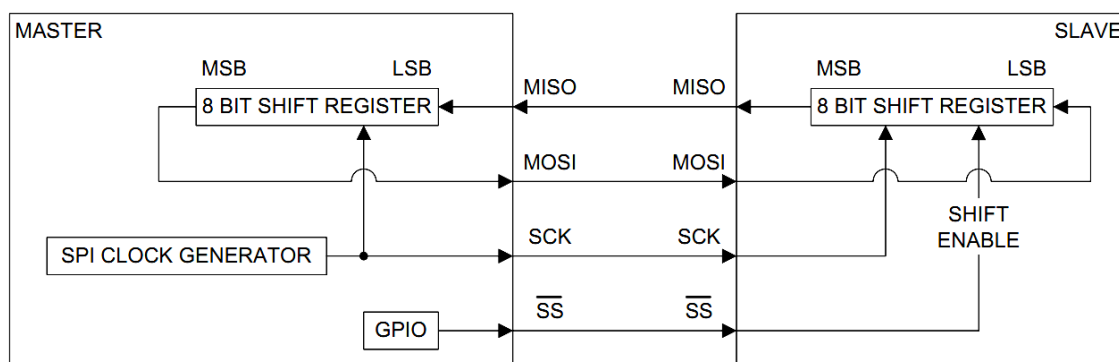


Obr. 2.8: Ukázka přenosu adresy po sběrnici I²C [1]

2.5 SPI mikrokontroléru [4]

Sběrnice SPI [4] představuje jednu z forem sériových externích sběrnic sloužících pro vzájemné propojení dvou či více komunikujících uzlů. Jeden uzel obvykle vystupuje v roli takzvaného řadiče sběrnice (MASTER), ostatní uzly pracují v režimu SLAVE. Uzel, který pracuje jako MASTER, obsahuje generátor hodinového signálu, který je rozveden do všech ostatních uzlů, čímž je umožněn zcela synchronní (navíc obousměrný) přenos dat. Hodinový signál je rozváděn vodičem označovaným symbolem SCK. Kromě vodiče s hodinovým signálem jsou uzly propojeny dvojicí vodičů označovaných většinou symboly MISO (Master In, Slave Out) a MOSI (Master Out, Slave In), pomocí nichž se obousměrně (full duplex) přenáší data. Posledním signálem, který se u této sběrnice používá, je signál SS (Slave Select), jenž slouží – jak již jeho název napovídá – k výběru některého uzlu pracujícího v režimu SLAVE. Všechny čtyři signály, tedy: SCK, MISO, MOSI i SS, pro svou funkci vyžadují pouze jednosměrné porty, což přispívá k jednoduché a především levné implementaci této sběrnice. Přenos dat probíhá v následujícím pořadí [4]:

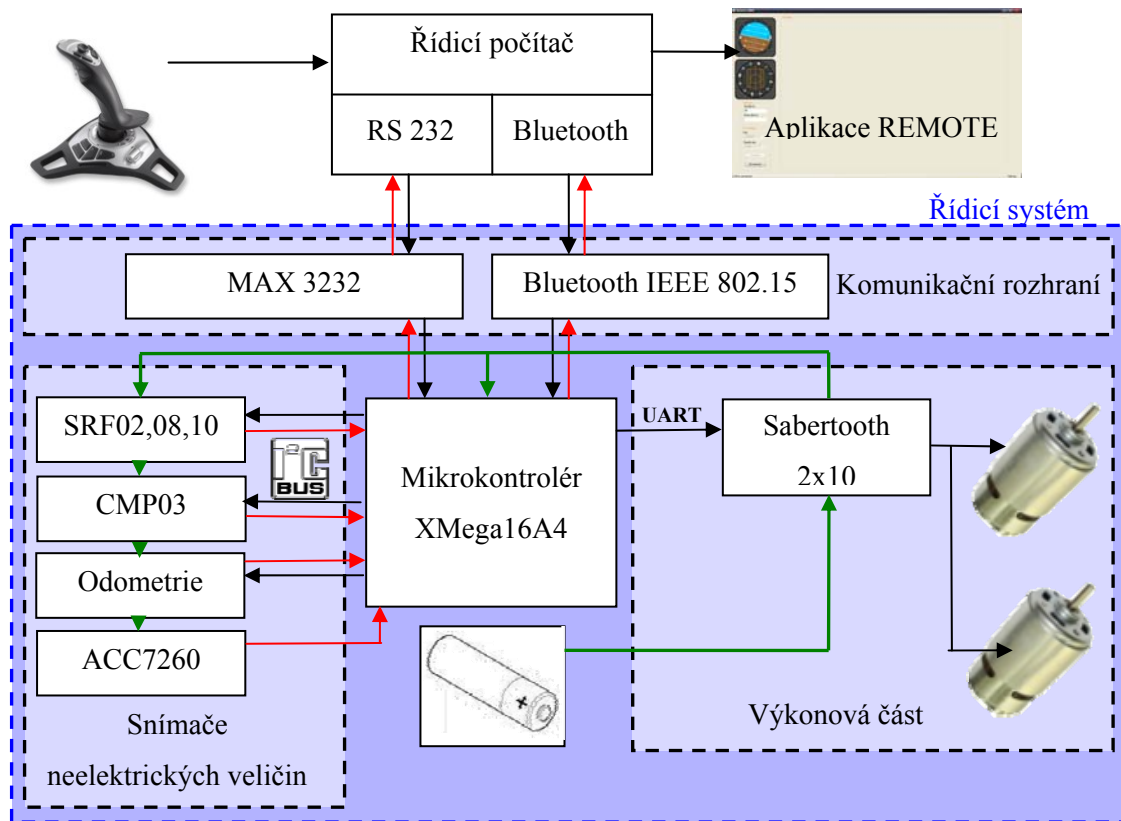
- Pro komunikaci Master nastaví log. 0 na SS zařízení se kterým chce komunikovat.
- Pak začne generovat hodinový signál na SCLK a v té chvíli vyšlou obě zařízení svoje data, přičemž MOSI je vždy Master výstup, Slave vstup a MISO je Master vstup, Slave výstup.
- Jakmile jsou data vyslána, může komunikace dále pokračovat.
- Master dále dodává hodinový signál, hodnota SS se nemění nebo může být ukončena: Master přestane vysílat hodinový signál a nastaví SS do log. 1.
- Délka vyslaných dat je buď 8 bitů nebo 16 bitů.



Obr. 2.9: Blokové schéma SPI komunikace [4]

3 POPIS A REALIZACE

3.1 Blokové schéma řídicího systému



Obr. 3.1: Blokové schéma řídicího systému

Kompletní schéma zapojení je zakresleno v Příloze 1.

3.2 Rozhraní RS 232

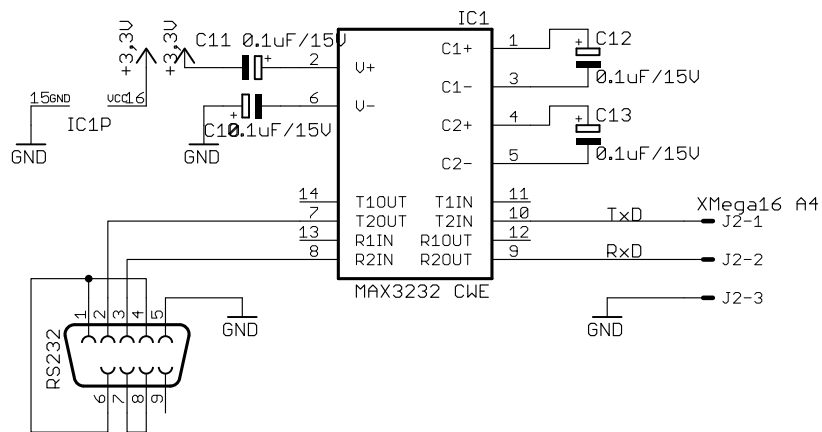
Mikrokontrolér obsahuje více rozhraní pro komunikaci přes USART. Následující tabulka zobrazuje způsob připojení jednotlivých pinů:

| | | | | | |
|-----------------|----------|--------|------------|----------|--------|
| RxD | Příjem | pin 32 | RxD | Příjem | pin 12 |
| TxD | Vysílání | pin 33 | TxD | Vysílání | pin 13 |
| Max 3232/ZigBee | | | Sabertooth | | |

Tab. 3.1: Popis komunikačních pinů mikrokontroléru

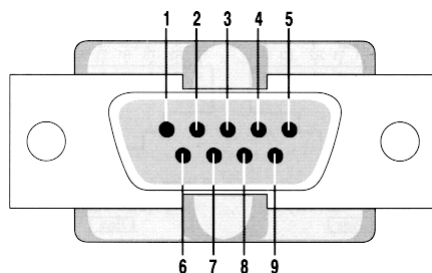
Nejjednodušší způsob vzájemného přepojení dvou sériových portů je přivedení vysílacího výstupu jednoho zařízení na přijímací vstup druhého zařízení a naopak – křížově. Pokud se použijí v zařízení TTL nebo CMOS obvody, je nutné jejich logiku před připojením napětově upravit.

Pro toto upravení se standardně používaly obvody 1488 a 1489, které ale potřebovaly +12V a -12V pro vytvoření výstupních úrovní. Přelom v tomto směru učinila firma MAXIM svým obvodem MAX 3232. [6]Využila totiž svých znalostí ve vývoji spínaných nábojových měničů napětí a vyvinula obvod, který si vystačí s +3,3V. Potřebné napětí si samostatně vyrobí pomocí čtyř externích kondenzátorů.



Obr. 3.2: Zapojení obvodu MAX 3232

Samotný komunikační port v počítači je reprezentován rozhraním RS 232, ke kterému se periferie připojují konektorem CANNON 9.



Obr. 3.3: Popis konektoru RS 232

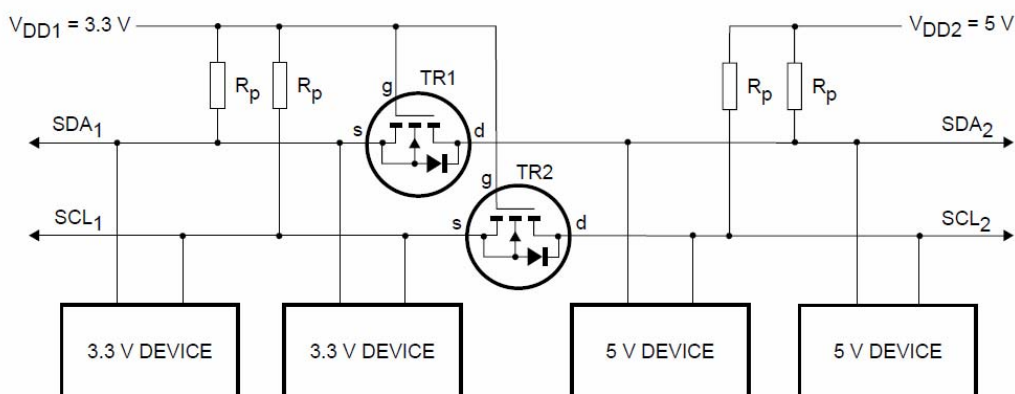
| Pin | Signál | Pin | Signál |
|-----|----------------------------|-----|-------------------------|
| 1 | Datový přenos detekovaný | 6 | Set dat připravený |
| 2 | Příjem dat | 7 | Žádost na odesílání |
| 3 | Odesílání dat | 8 | Připravený na odesílání |
| 4 | Datový terminál připravený | 9 | Indikátor zvonění |
| 5 | Signálová země GND | | |

Tab. 3.2: Popis kontaktů konektoru RS 232

3.3 Převodník úrovní pro sběrnici I²C

Pokud není vybraný mikrokontrolér 5V tolerantní, je zapotřebí upravit úroveň sběrnice I²C z 5V na 3,3V.

Převodník použitý v návrhu řídicí desky je zobrazen na obrázku 3.4. [8] Jádrem zapojení jsou dva MOS-FET tranzistory s kanálem typu N. Gate obou tranzistorů je připojen na část s nižší napětovou úrovní (+3,3V). Drain tranzistorů představuje část sběrnice s 5V logikou a source 3,3V logiku.



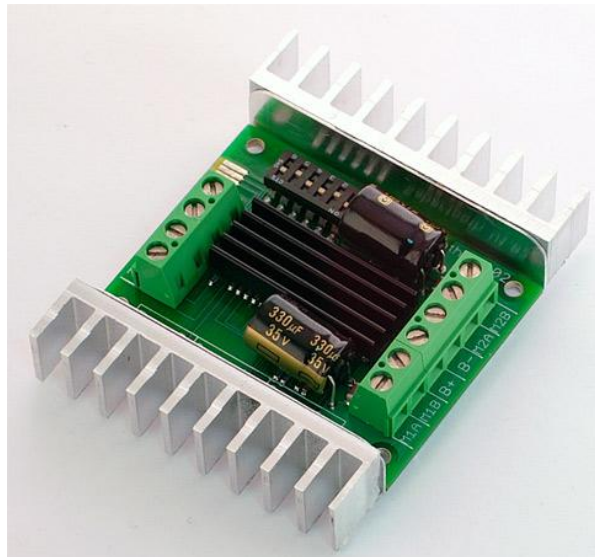
Obr. 3.4: Obousměrný převodník úrovní [8]

3.4 Pohon robota

Celá mobilní platforma je řízena prostřednictvím čtyř stejnosměrných motorů, které jsou přes planetární převodovky napojeny na pásový pohon podvozku. Motory jsou zapojeny jako pár pro jednu stranu. Z toho plyne, že se vlastně jedná o diferenciální podvozek.

3.5 Sabertooth 2x10

Jako výkonová část pro ovládání motorů malé mobilní platformy se používá modul pod názvem Sabertooth 2x10 od firmy DimensionEngineering. [9] Jedná se o dvojitý motor driver, který je napojen na diferenciální pásový podvozek malého mobilního robota.

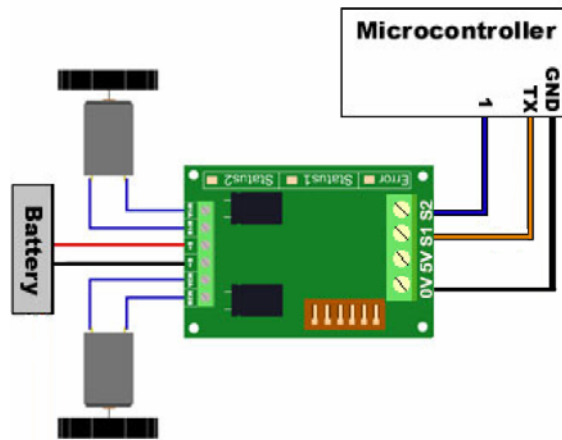


Obr. 3.5: Sabertooth 2x10 [9]

Základní parametry modulu [9]:

- Napájecí napětí 6-24V, maximálně 30V
- Výstupní proud 10A na kanál, peak 15A
- Možnost napájení s baterií NiMH, NiCd nebo LiPol
- Rozměr 59x75x17mm
- Tepelná a protiproudová ochrana
- PWM frekvence 32kHz
- 4 módy řízení – analogový vstup, R/C vstup, jednoduchý seriál, packetovaný seriál
- Konfigurace prostřednictvím DIP spínačů

Na řízení se bude používat jednoduchý seriál mód. Po sériovém kanálu USART se nastavuje rychlost a směr motorů. Modul se připojuje dle následujícího obrázku:

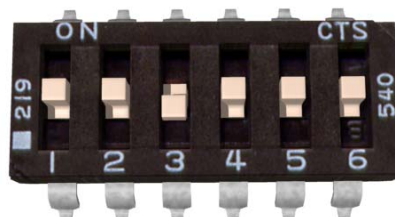


Obr. 3.6: Zapojení Sabertooth 2x10 v jednoduchém sériovém módu [9]

Pro konfiguraci modulu Sabertooth slouží DIP přepínač zobrazen na obrázku 3.7 a popis funkčnosti jednotlivých elementů přepínače je popsán v následující tabulce:

| Číslo | Popis |
|-------|---------------------------------------|
| 1 | Udává formát vstupních řídicích údajů |
| 2 | Udává formát vstupních řídicích údajů |
| 3 | Lithium mód |
| 4 | Mixed mód pro diferenciální podvozek |
| 5 | Exponenciální odezva |
| 6 | Citlivost |

Tab. 3.3: Popis DIP přepínače modulu Sabertooth



Obr. 3.7: Konfigurační DIP přepínač modulu Sabertooth [9]

3.6 Snímače neelektrických veličin

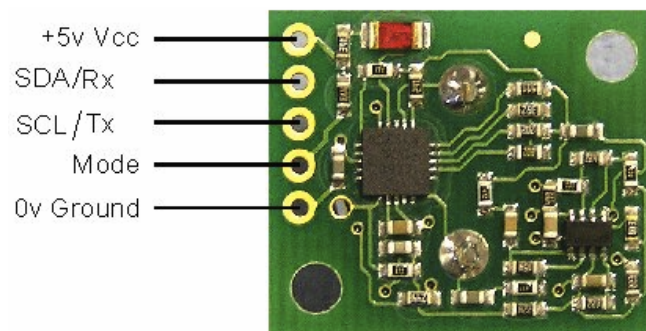
Malý mobilní robot bude dále doplněn o čtyři druhy snímačů. O ultrazvukové snímače vzdálenosti - dvakrát SRF08, SRF10 a SRF02, které budou mapovat okolí robota. Dalším snímačem bude trojosý akcelerometr ACC7260, který bude detekovat naklonění podvozku. Na detekci natočení podvozku, vzhledem k zemi, bude sloužit modul elektronického kompasu CMPS03. Posledním navrženým snímačem bude modul odometrie s obvodem ADNS-5030.

3.6.1 Ultrazvukové snímače vzdálenosti typu SRFxx

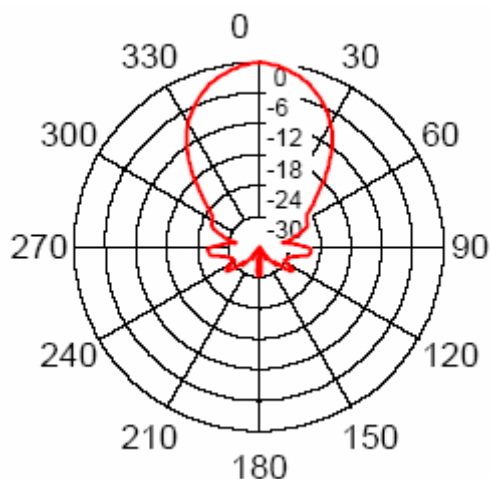
Základním principem měření vzdálenosti ultrazvukovým snímačem je měření času od vyslání ultrazvukového signálu do příjmu odraženého signálu - TOF. Následně se dá z této doby určit vzdálenost předmětu, která odrazila ultrazvukový signál.

Charakteristika SRF02 [10]:

- I²C a UART rozhraní
- Vyzařovací úhel senzoru 55°
- Napájení 5V
- Proudový odběr 4 mA
- Měřicí rozsah 15cm – 6m
- Pracovní frekvence 40kHz
- Jenom jeden ultrazvukový měnič
- Možnost vyslání pulsu bez měření a měření bez vyslání pulsu
- Rozměry 24 x 20 x 17 mm



Obr. 3.8: Zapojení vývodů dálkoměru SRF02 [10]



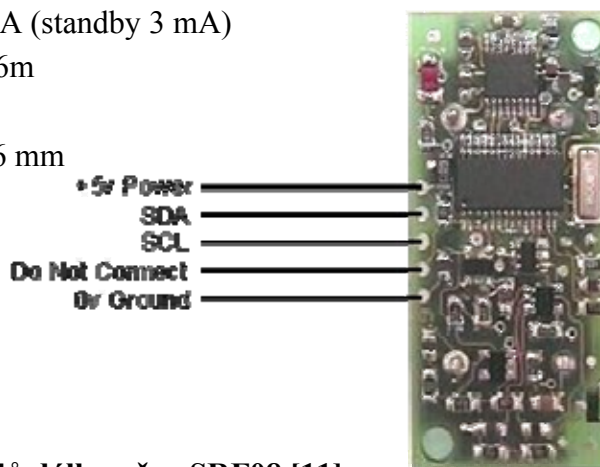
Obr. 3.9: Vyzařovací charakteristika SRF02 [10]

| Číslo registru | Čtení | Zápis |
|----------------|-----------------------------|-------------------|
| 0 | Softwarová kontrola | Příkazový registr |
| 1 | Nevyužitý | - |
| 2 | Vzdálenost horní byte | - |
| 3 | Vzdálenost dolní byte | - |
| 4 | Min. vzdálenost, horní byte | - |
| 5 | Min. vzdálenost, dolní byte | - |

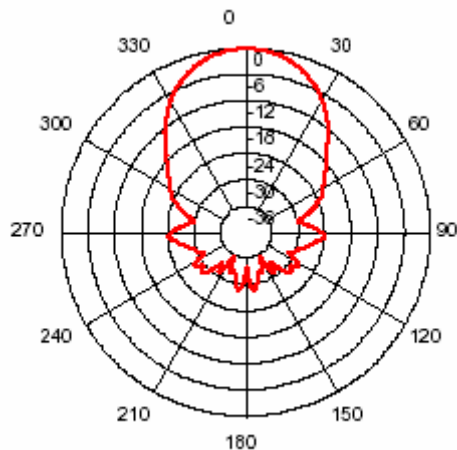
Tab. 3.4: Popis registrů dálkoměru SRF02

Charakteristika SRF08 [11]:

- I²C rozhraní
- Vyzařující úhel senzoru 55°
- Napájení 5V
- Proudový odběr 15 mA (standby 3 mA)
- Měřicí rozsah 3cm – 6m
- Světelný senzor
- Rozměry 43 x 20 x 16 mm



Obr. 3.10: Zapojení vývodů dálkoměru SRF08 [11]



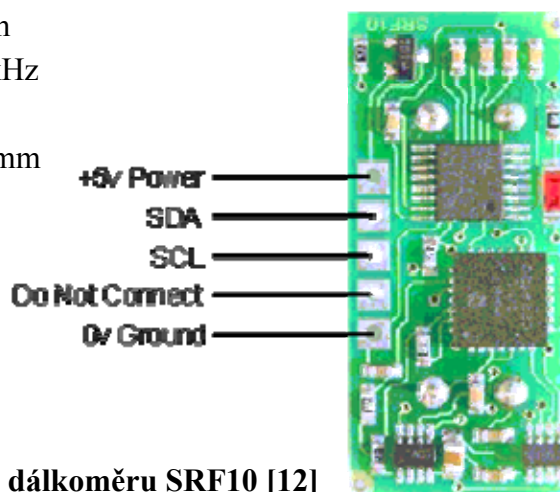
Obr. 3.11: Vyzařovací charakteristika SRF08 [11]

| Číslo registru | Čtení | Zápis |
|----------------|-----------------------|-------------------|
| 0 | Softwarová kontrola | Příkazový registr |
| 1 | Světelný senzor | Max. zesílení |
| 2 | 1. odraz, horní byte | Dosahový registr |
| 3 | 1. odraz, dolní byte | - |
| ... | ... | ... |
| 34 | 17. odraz, horní byte | - |
| 35 | 18. odraz, dolní byte | - |

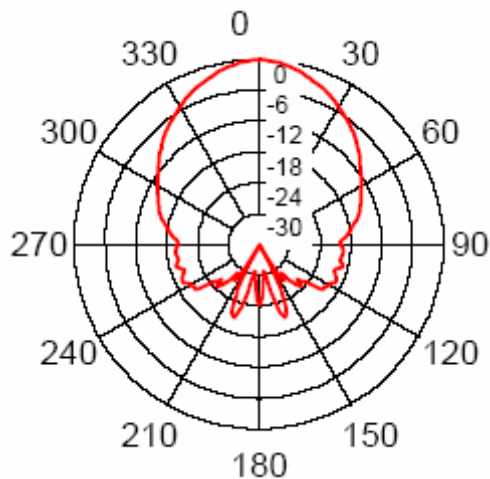
Tab. 3.5: Popis registrů dálkoměru SRF08

Charakteristika SRF10 [12]:

- I²C rozhraní
- Vyzařující úhel senzoru 72°
- Napájení 5V
- Proudový odběr 15 mA (standby 3 mA)
- Měřicí rozsah 3cm – 6m
- Pracovní frekvence 40kHz
- Malé rozměry
- Rozměry 32 x 15 x 10 mm



Obr. 3.12: Zapojení vývodů dálkoměru SRF10 [12]



Obr. 3.13: Vyzařovací charakteristika SRF10 [12]

| Číslo registru | Čtení | Zápis |
|----------------|-------------------------|-------------------|
| 0 | Softwarová kontrola | Příkazový registr |
| 1 | Nevyužitý | Max. zesílení |
| 2 | Vzdálenost - horní byte | Dosahový registr |
| 3 | Vzdálenost - dolní byte | - |

Tab. 3.6: Popis registrů dálkoměru SRF10

Komunikace se snímači probíhá pomocí I²C. Výrobce doporučuje na I²C sběrnici připojit pul – up rezistory s hodnotou 1,8 kΩ. Standardně nastavená I²C adresa od výrobců je 0xE0. Uživatel si ji však může změnit na 16 různých adres, z toho plyne, že na jednu sběrnici I²C je možné připojit až 16 SRFxx snímačů.

Po připojení napájecího napětí krátce zablikne LED dioda a následně, pokud je I²C adresa změněna, LED dioda prostřednictvím blikání zobrazí binární kód nastavené adresy.

Následující tabulka ukazuje nastavení adres pro jednotlivé použité snímače:

| Typ snímače (Typ_por.č) | Adresa | Připojení (Rozhraní_por.č) |
|----------------------------|--------|-------------------------------|
| SRF08_1 | 0xE0 | TWI_1 |
| SRF02_1 | 0xF0 | TWI_1 |
| SRF10_1 | 0xE0 | TWI_2 |
| SRF08_2 | 0xF0 | TWI_2 |

Tab. 3.7: Nastavení adres snímaču SRFxx

Jedno měření vzdálenosti trvá 65 ms. Měření začíná po zápisu příkazu měření do příslušného příkazového registru podle následující tabulky, která platí pro všechny tři použité typy snímačů:

| Příkaz | Popis |
|--------|---------------------------|
| 0x50 | Výsledek měření v inch |
| 0x51 | Výsledek měření v cm |
| 0x52 | Výsledek měření v μ s |

Tab. 3.8: Popis příkazů dálkometrů

Po provedení měření vzdálenosti se automaticky vždy uloží i hodnota světelného senzoru u SRF08, která je v rozsahu 0 – 255. Úplné osvětlení vrací hodnotu 0.

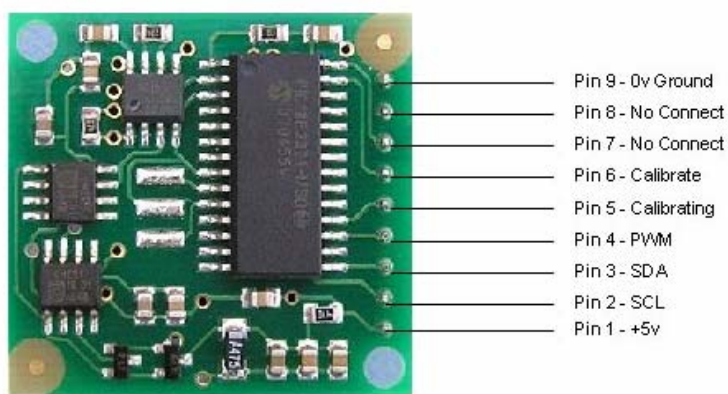
3.6.2 Modul kompasu CMPS03

Tento modul kompasu byl speciálně vyvinut pro robotické aplikace, především pro robotickou navigaci. Hlavním cílem bylo nasnímat polohu robota vzhledem k zemi. Nejjednodušším způsobem je snímání zemského gravitačního pole. [12] Jako snímač magnetického pole se používají snímače firmy Philips KMZ51. Jejich magnetická citlivost je tak velká, že jsou schopny zaznamenat i změny magnetického pole země.

Charakteristika CMPS03 [12]:

- Pracovní frekvence SCL až do 1MHz
- PWM od 1ms do 37ms v 0.1ms krocích
- Napájení 5V
- Proudový odběr 25 mA
- Citlivost 0,1°
- Přesnost 3-4°, podle kalibrace
- Rozměry 32 x 35 mm

Výstup ze snímače je číselná reprezentace úhlu natočení. Pro propojení s mikrokontrolérem se používá rozhraní I²C. Daný modul CMPS03 má i PWM výstup, jehož střída je závislá na úhlu natočení.



Obr. 3.14: Modul elektronického kompasu CMPS03 [12]

Základní interval vzorkování kompasu CPMS03 je 100ms. Nový firmware Rev15 však umožňuje také 33ms skenování (30 aktualizací za 1s) a 300ms skenování, které reprezentuje 3,3 aktualizací za sekundu. 33ms skenování se doporučuje do aplikací, které požadují vysokou rychlost aktualizace údajů kompasu. Nevýhodou rychlého vzorkování je, že může nastat nestabilita načteného výsledku. Proto v aplikacích, které nepotřebují rychlé vzorkování, je výhodnější používat 300ms režim pro přesné čtení hodnoty úhlu z kompasu. Výrobce přednastavená hodnota vzorkování je 100ms.

Následující tabulka popisuje funkci jednotlivých registrů [12]:

| Číslo registru | Popis |
|----------------|---|
| 0 | Softwarová kontrola |
| 1 | Úhel kompasu v bytu, 0-255 celé otočení |
| 2, 3 | Úhel kompasu jako slovo, 0-3599 pro 0-359.9° |
| 4, 5 | Vnitřní diferenční test snímače 1 – hodnota v ose x |
| 6, 7 | Vnitřní diferenční test snímače 2 – hodnota v ose y |
| 8, 9 | Kalibrační hodnoty snímače 1 |
| 10, 11 | Kalibrační hodnoty snímače 2 |
| 12 | Odblokovací kód 1 pro změnu I ² C adresy |
| 13 | Odblokovací kód 2 |
| 14 | Odblokovací kód 3 |
| 15 | Příkazový registr |

Tab. 3.9: Popis registrů kompasu CMPS03

Registr číslo 15 je příkazový registr. Na změnu adresy I²C sběrnice slouží příkazy 0xC0-CE a 0xF2 pro nastavení kalibrační hodnoty od výrobce. Tento příkaz je totožný s 0xFF, avšak 0xF2 potřebuje odblokovací kód (unlock code).

3.6.3 Trojosý akcelerometr ACC7260

Daný akcelerometr [13] je od firmy SnailInstrument a jeho jádrem je obvod MMA7260. Jedná se o nízkopříkonový MEMS kapacitní akcelerometr of firmy FreeScale. Jeho funkčnost je založena na změně kapacity plošných kondenzátorů změnou polohy seizmické hmotnosti. Základní charakteristika modulu trojosého akcelerometru ACC7260 [13]:

- Měření vzdálenosti ve 3 osách
- Měřený rozsah $\pm 1,5G \dots \pm 6G$
- Napájení 3,6 V až 16 V
- Citlivost 800mV/g .. 200mV/g
- Nelinearita +/- 1% z rozsahu
- Křížová citlivost: 5%
- Spotřeba: 500 μ A, sleep módu 3 μ A
- Rozměry modulu 23x28mm, DIL8 0.6"

Akcelerometry jsou senzory pro měření statického nebo dynamického zrychlení. Jsou vhodné nejen pro měření odstředivých a setrvačných sil, ale i pro určování pozice tělesa, jeho naklonění nebo vibrací.

Použitý senzor MMA7260 je připojen na malém modulu spolu s filtračními RC články a stabilizátorem napětí. Modul takto umožňuje jednodušší použití této SMD součástky. K upevnění modulu slouží kromě patice DIL8 0.6" také tři otvory o průměru 2,2 mm.



Obr. 3.15: Trojosý akcelerometr ACC7260

Modul umožňuje vybrat ze 4 měřících rozsahů podle zapojení vývodů G1 a G2. Pro náš návrh jsem použil rozsah $\pm 1,5G$, avšak vývody G1 a G2 jsem připojil na externí konektor řídicí desky. Takto se naskytla další možnost změny měřícího rozsahu akcelerometru přepojovacími jumpery.

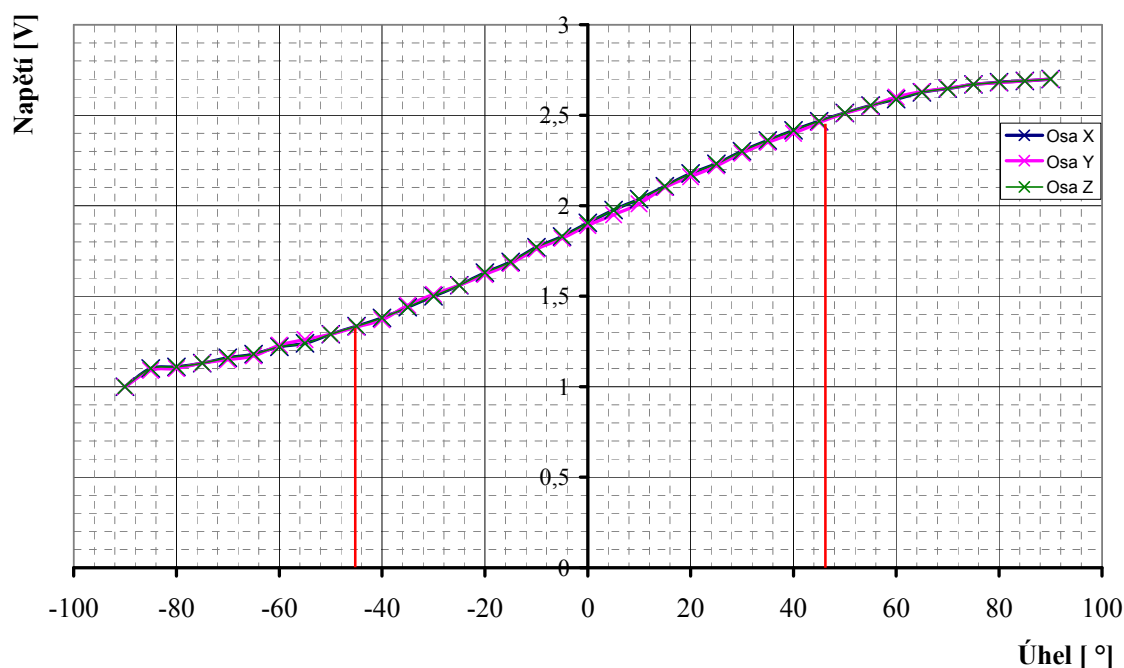
| G1 | G2 | Rozsah |
|------|------|------------|
| HIGH | LOW | $\pm 1,5G$ |
| HIGH | LOW | $\pm 2,0G$ |
| LOW | HIGH | $\pm 4,0G$ |
| HIGH | HIGH | $\pm 6,0G$ |

Tab. 3.10: Konfigurace měřicího rozsahu modulu ACC7260

Výstupním signálem je napětí, jehož hodnota je úměrná zrychlení v příslušné ose X, Y a Z. Toto napětí jsem přivedl do A/D převodníku mikrokontroléru. Při nulovém zrychlení v příslušné ose je výstup polovinou napájecího napětí senzoru (1.65 V). Kladný a záporný směr pro každou osu je uveden na plošném spoji modulu.

Pro měření náklonu robota je důležité znát statickou charakteristiku akcelerometru na nakloněné rovině ve všech osách. V případě, že charakteristika bude nelineární, je potřeba ji linearizovat nebo se omezit na měření náklonu jen v lineární části.

Měření výstupní charakteristiky bylo provedeno pro každou osu v rozsahu ± 90 . Jelikož charakteristika není v celé své šířce lineární. Nejmenší chybovost při měření náklonu je v lineární části charakteristiky, což odpovídá -45° do $+45^\circ$. Rozdíly mezi jednotlivými osami jsou způsobeny rozdílnými výstupními napětími při působení $0g$.

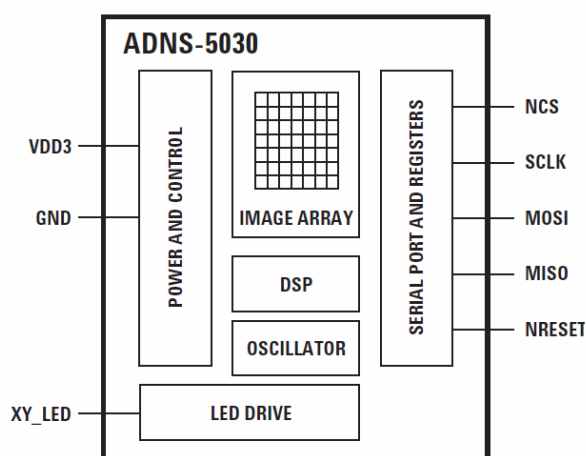


Obr. 3.16: Výstupní charakteristika akcelerometru ACC7260

3.6.4 Modul pro odometrii podvozku

Pro sledování ujeté dráhy robota – odometrii je použita optická metoda prostřednictvím integrovaného obvodu ADNS-5030.

Je to snímač pro optické myši od firmy Avago Technologies. Integrovaný obvod ADNS-5030 [15] obsahuje systém pro snímání obrazu (IAS), realizovaný CCD čipem, digitální signální procesor a sériové SPI rozhraní. CCD čip snímá obraz podložky a na základě změny snímaných obrazů (framů) signální procesor vyhodnocuje směr a velikost výchylky. Pro eliminaci dalších vnějších komponent obsahuje integrovaný obvod také i řadič pro osvětlovací LED diodu.



Obr. 3.17: Blokové schéma ADNS-5030 [15]

Základní charakteristika ADNS-5030 [15]:

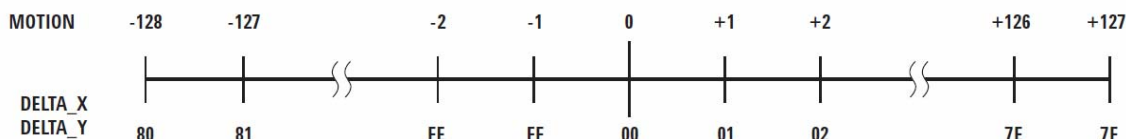
- Napájecí napětí 3,3 V
- 4 vodičové SPI + reset
- Maximální frekvence SCLK je 1MHz
- Přesný pohyb 14 IPS (inch per second), zrychlení 2g
- Rozlišení 1000 CPI (count per inch)

Pro komunikaci s ADNS-5030 se využívá sériového SPI rozhraní. Obvod je aktivován a připraven na komunikaci přivedením nízké úrovně GND na čip select NCS. Po dobu operace zápisu vzorkuje ADNS-5030 data na MOSI na každé náběžné hraně SCK a při čtení na každé doběžné hraně SCK.

| Adresa | Popis | Forma přístupu |
|-------------|--|------------------|
| 0x00 | Identifikační číslo produktu | R (default 0x11) |
| 0x01 | Identifikační číslo revize | R (default 0x00) |
| 0x02 | Pohyb | R |
| 0x03 | Delta X | R |
| 0x04 | Delta Y | R |
| 0x05 | SQUAL kvalita povrchu | R |
| 0x06 | Horních 8 bitů času elektronické uzávěrky | R |
| 0x07 | Dolních 8 bitů času elektronické uzávěrky | R |
| 0x08 | Maximální hodnota pixelu v aktuálním framu | R |
| 0x09 | Suma pixelů | R |
| 0x0A | Minimální hodnota pixelu v aktuálním framu | R |
| 0x0B | Pixel grabber | R/W |
| 0x0C | Rezervované | – |
| 0x0D | Nastavení rozlišení a způsobu vypnutí | R/W |
| 0x0e – 0x39 | Rezervované | – |
| 0x3A | Čip reset | W |
| 0x3b – 0x3E | Rezervované | – |
| 0x3F | Inverzní hodnota registru 0x01 | R |
| 0x40 | Nastavení proudu osvětlovací LED | W |
| 0x41 – 0x44 | Rezervované | – |
| 0x45 | Nastavení úsporného režimu | R/W |
| 0x46 – 0x62 | Rezervované | – |
| 0x63 | Burst mód | R |

Tab. 3.11: Seznam registrů obvodu ADNS-5030

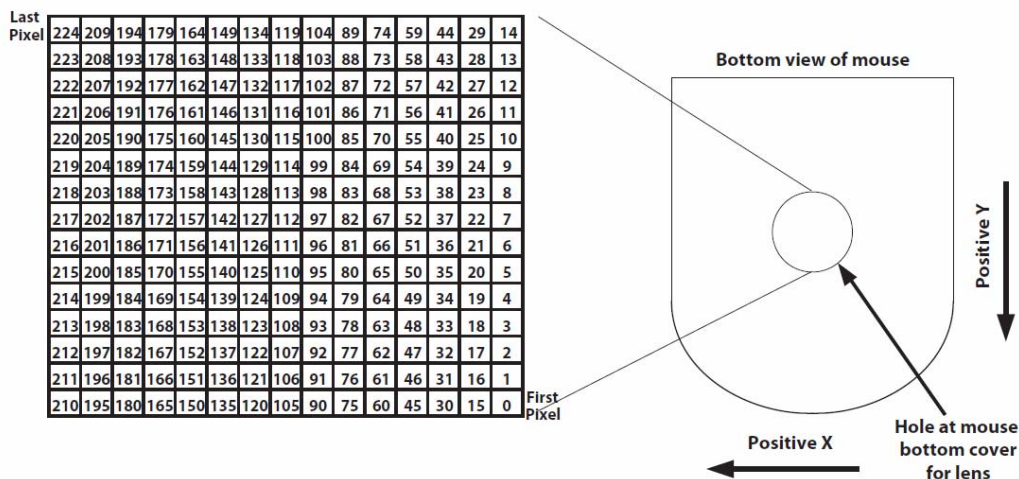
Pro měření odometrie se mohou využívat registry 0x03 a 0x04, které přímo obsahují číselnou hodnotu zaznamenaného rozdílu ujeté dráhy ve dvou směrech x a y.



Obr. 3.18: Číselná reprezentace pohybu v ose x a y [15]

Obvod byl navržen do optických myši. Vzdálenost od povrchu, ve které snímač dokáže zaostřit obraz povrchu a z něj vykonávat další výpočty, je pro optické myši malá. Pro robotické účely odometrie se tato vzdálenost musela zvětšit vhodnou optikou nasazenou před integrovaný obvod.

Ostrost snímaného povrchu se dá sledovat vyčtením obrazu o rozlišení 18x18 pixelů. Cyklickým čtením registru 0x0B se vyčtou jednotlivé pixely podle obrázku 3.19.



Obr. 3.19: Reprezentace pixelů snímaného obrazu [15]

Na základě uvedeného integrovaného obvodu jsem zkonstruoval modul pro odometrii, který je zobrazen v příloze 12. Příloha 10 dále obsahuje schématické zapojení modulu. Modul se kromě obvodu ADNS-5030 skládá z filtračních kondenzátorů, přípojného místa osvětlovací LED diody a konektorů pro napájení, sériového rozhraní SPI a NCS s resetem.

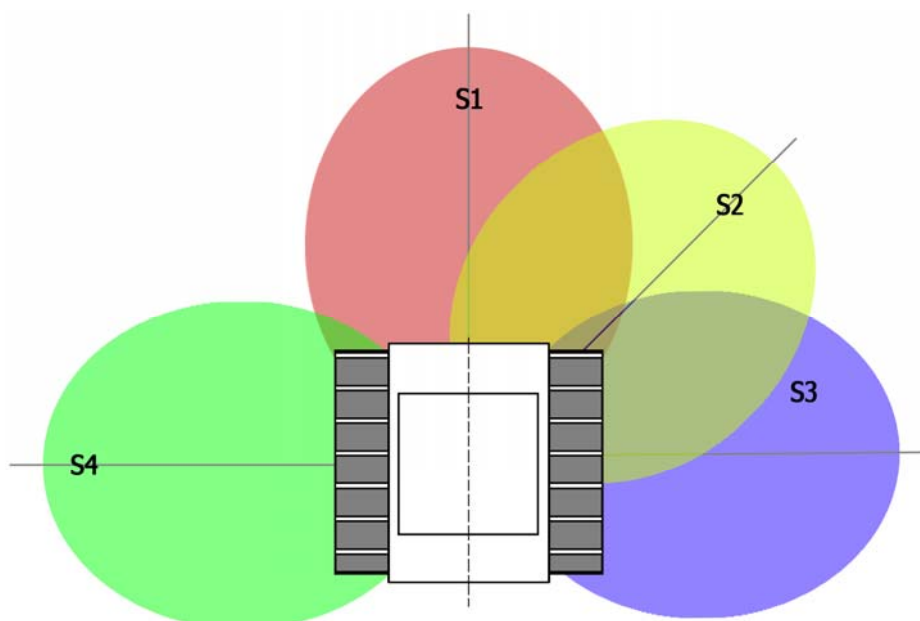
Pro ladění funkčnosti snímače jsem používal jako optiku čočku z laserové mechaniky CD-romu. Toto řešení však nebylo ideální. Následkem byla velká chybovost odometrie, která byla způsobena neostrým snímáním obrazu povrchu.

3.7 Rozložení snímačů

Mobilní podvozek není vybaven žádnou kamerou ani jiným zařízením na snímání obrazu okolí. Proto jsem se rozhodl zvolit navigaci podvozku na základě informací získaných ze snímačů neelektrických veličin jako jsou ultrazvukové dálkoměry, akcelerometr pro naklonění podvozku a elektronický kompas na snímání pootočení podvozku. Dalším rozšířením pro odometrii má být snímání ujeté dráhy podvozku optickou metodou. Ultrazvukové dálkoměry jsou rozloženy s orientací na pravou stranu, čímž se eliminuje počet snímačů levé strany.

Detailní rozložení ultrazvukových snímačů je zobrazeno na obr. 3.20. Mezi snímačem S1 a S2 je 45° odchylka a mezi S1 a S3 90° . Levá strana je hlídána jedním snímačem S4, který je odchýlen o 90° vzhledem k S1. Akcelerometr s kompasem jsou umístěni uprostřed podvozku, aby se mohlo vykonávat co nejpřesnější měření stavu podvozku.

Hodnoty vzdálenosti získané z následující konfigurace slouží k přibližné rekonstrukci prostředí v uživatelském rozhraní REMOTE.



Obr. 3.20: Rozložení snímačů na mobilní platformě – pohled shora

Tabulka 3.7 obsahuje popis jednotlivých použitých snímačů – všeobecnou označovanou symboliku snímačů použitých v práci.

| Typ | Označení | Symbolika |
|--------|----------|-----------|
| SRF08 | S4 | SRF08_1 |
| SRF10 | S1 | SRF10_1 |
| SRF02 | S2 | SRF02_1 |
| SRF08 | S3 | SRF08_2 |
| CMPS03 | S0 | CMPS03_1 |

Tab. 3.12: Označení jednotlivých snímačů

3.8 Výpočet naklonění

Pro výpočet úhlu náklonu je nutné znát jednotlivé složky gravitačního zrychlení v jednotlivých měřených osách X , Y a Z , jak je uvedeno na obrázku 3.21.

Jelikož výstupní charakteristika není v celé své šířce lineární, tak se nejmenší chybovosti při měření náklonu dopustíme v lineární části charakteristiky, která se nachází v intervalu od -45° do $+45^\circ$. Pro výpočet úhlu náklonu je nutné znát jednotlivé složky zrychlení v daných osách X , Y a Z .

Pro zrychlení v měřené ose lze odvodit následující vzorec:

$$A = \frac{V_{A/D} - V_{offset}}{\frac{\Delta V}{\Delta g}} [ms^{-1}] \quad (2.1)$$

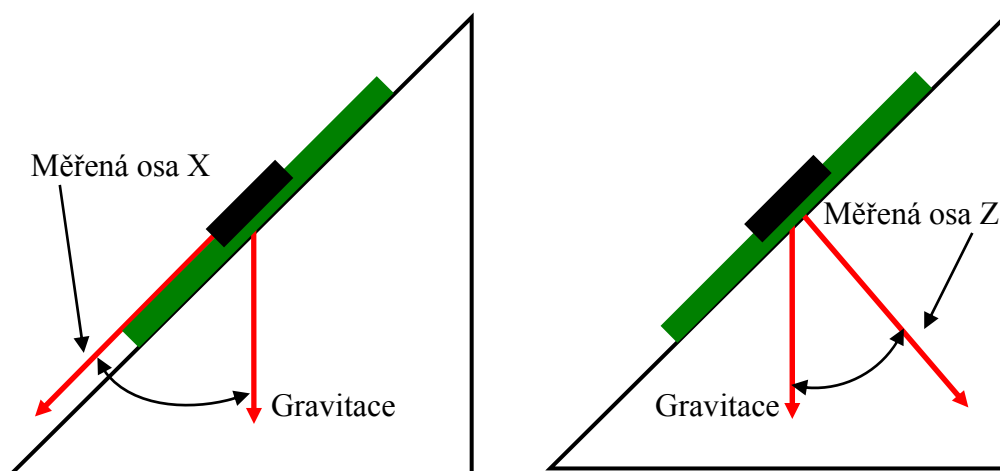
A je zrychlení v měřené ose,

$V_{A/D}$ je výstupní napětí akcelerometru připojené na A/D převodník,

V_{OFFSET} je výstupní napětí při $0g$,

$\frac{\Delta V}{\Delta g}$ je citlivost snímače

Při měření náklonu robota tříosým akcelerometrem lze použít všechny tři osy akcelerometru pro získání všech složek zrychlení. Úhel ρ vymezuje úhel mezi osou X a zemským povrchem, úhel ϕ vymezuje úhel mezi osou Y a zemským povrchem a úhel ϕ je úhel mezi osou Z a gravitační silou.



Obr. 3.21: Složky gravitačního zrychlení působící na osu X a Z akcelerometru.

Pro jednotlivé úhly náklonu v příslušné ose robota platí následující rovnice:

$$\rho = \arctg \left(\frac{A_x}{\sqrt{A_y^2 + A_z^2}} \right) [^\circ] \quad (2.2)$$

$$\varphi = \arctg \left(\frac{A_y}{\sqrt{A_x^2 + A_z^2}} \right) [^\circ] \quad (2.3)$$

$$\phi = \arctg \left(\frac{\sqrt{A_x^2 + A_y^2}}{A_z} \right) [^\circ] \quad (2.4)$$

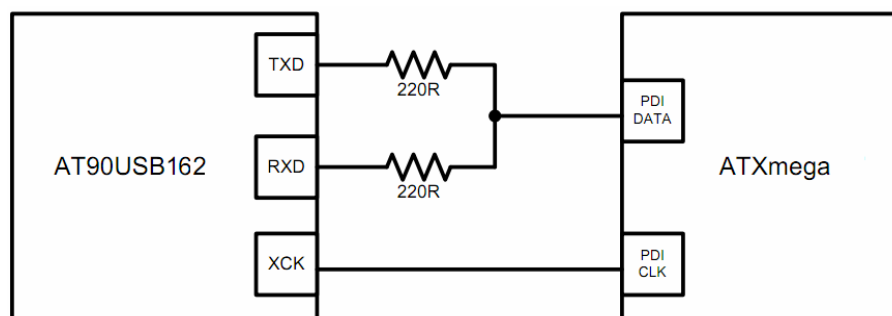
Výsledná suma všech zrychlení se musí rovnat statickému zrychlení $1g$:

$$1g = \sqrt{A_x^2 + A_y^2 + A_z^2} \quad (2.5)$$

Pro práci robota bude stačit počítat pouze odklon od zemského povrchu, což odpovídá úhlům ρ a φ . Na základě těchto úhlů bude možné sledovat naklonění podvozku mobilní platformy prostřednictvím uživatelského rozhraní, případně zamezit převrácení robota.

3.9 Programátor USBTiny MkII

Jedná se o AVR programátor zkonstruovaný na základě LUFA projektu, který emuluje původní AVRISP – MKII. Podporuje programování 8 bitových AVR mikrokontrolérů přes ISP, řadu mikrokontrolérů Xmega přes PDI rozhraní a TPI rozhraní pro řadu Tiny.



Obr. 3.22: Hardwarové spojení programátoru [18]

Programovací a debugovací PDI [18] je proprietární rozhraní firmy ATMEL® pro externí programování a on-čip ladění zařízení. Podporuje programování NVM, Flash, EEPROM paměti, pojistek, lock bitů. Je to dvou pinové rozhraní, které využívá reset pin (35) jako hodinový vstup (PDI_CLK) a pin 34 jako datový vstup/výstup (PDI_DATA). PDI fyzická vrstva používá stejného protokolu jako UART, jeho blokové zapojení je znázorněno na obr. 3.22.

Základní parametry programátoru [17]:

- ISP, PDI a TPI protokol
- 4MHz ISP záchranný hodinový výstup pro chybně nakonfigurované pojistky
- Přepínatelné cílové napájecí napětí (5V USB – cílové VCC)
- Cíl musí být 5V tolerantní pro použití 5V USB napětí, v naší aplikaci je možno použít jedině cílového VCC
- LED indikátor stavu (ACTIVE) a připojení k USB (PWR)
- Funkcionalita pod AVRStudio pro Windows a i pro AVRdude pro Linux
- Standardní AVR 6 pinové zapojení konektorů podle tabulky 3.7
- Možnost aktualizace firmwaru
- Open source

Základem zapojení programátoru je mikrokontrolér AT90USB162, který přímo obsahuje rozhraní pro operace s USB. Datový signál na USB sběrnici je ošetřen dvěma zenerovými diodami 3V3, aby se předešlo napěťovému přetížení na data+ a data- signálech USB sběrnice. Mikrokontrolér je napájen napětím z USB portu. Datové signály pro jednotlivé protokoly jsou dále přivedeny na integrovaný obvod GTL2010 (Gunning Transceiver Logic). Je to 10 bitový obousměrný převodník úrovní. Díky němu je možno měnit formu napájení programovacího konektoru. U programování obvodů s 5V tolerantní logikou lze použít napájení přímo z USB portu. Přepínačem lze také změnit zdroj napájení přímo z desky, kterou programujeme. Což je vhodné na jiné napěťové úrovni než 5V, v našem případě 3,3V.

Schéma zapojení programátoru je uvedeno v příloze 6.

| | ISP | PDI | TPI |
|---|------|------|------|
| 1 | MISO | DATA | DATA |
| 2 | VCC | VCC | VCC |
| 3 | SCK | NC | CLK |
| 4 | MOSI | NC | NC |
| 5 | RES | CLK | RES |
| 6 | GND | GND | GND |

Tab. 3.13: Zapojení konektorů programátoru

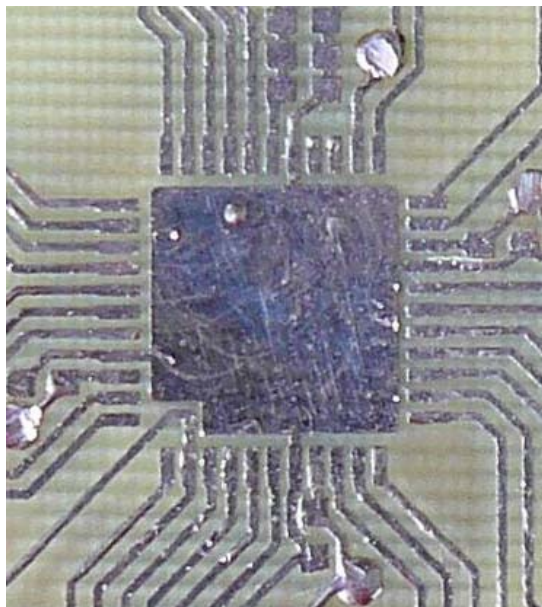
3.10 Návrh DPS řídicí jednotky

Deska plošného spoje řídicí jednotky malého mobilního robota byla navržena v CAD systému Eagle 5.4 Výsledné schéma zapojení je zobrazeno v příloze 1 a deska plošného spoje dále v příloze 2 až 5.

Po dobu návrhu řídicí desky jsem dbal na správné rozmístění blokovacích a filtračních kondenzátorů. DPS je dvouvrstvá a horní vrstva, která je vylitá mědí, má sloužit jako stínění řídicí desky. Na ploše pod mikrokontrolérem, pomocí měděných vrstev DPS, jsou vytvořeny elektrody dalšího ochranného kondenzátoru.

První okruh napájecího obvodu je realizován pomocí LE33, což je 3,3V stabilizátor napětí, ošetřený filtračními kondenzátory podle katalogových listů výrobce. Druhým 5V okruhem je přímo 5V výstup desky Sabertooth. Chod motorů může vyvolat v přírodních vodičích rušivé naindukované napětí. Proto jsem použil transil jako

ochranu proti přepětí a eliminaci rušivých napěťových špiček. Napájecí část je jištěna tavnou pojistkou 160mA a je navržena jenom na napájení mikrokontroléru a snímačů neelektrických veličin.



Obr. 3.23: Detail elektródy vytvořeného ochranného kondenzátoru na DPS

Všechny porty mikrokontroléru jsou vyvedeny na konektory pod názvem PORTA, PORTB, PORTC, PORTD a PORTE. Konektory také obsahují i dva 3,3V napájecí piny. Deska je dále osazena 4 konektory pro připojení zařízení na I²C sběrnici, 2 konektory pro UART, jeden pro RS232 a nakonec konektory pro vložení modulů akcelerometru ACC7260 a kompasu CMPS03.

3.11 Napájení obvodů

Napájení celého řídicího systému a jeho komponentů je realizováno prostřednictvím dvou sériově zapojených lithium polymerových akumulátorů typu KOKAM 2000mAh – 15C jištěných přes pojistku. Výhodou daných akumulátorových článků je velký dodaný výkon při malé hmotnosti. Použitý akumulátor se používá v R/C modelech. Při práci si musí dbát na to, aby se jednotlivé akumulátorové články nevybily pod doporučovanou úroveň 3V.

Základní parametry jednoho akumulátoru:

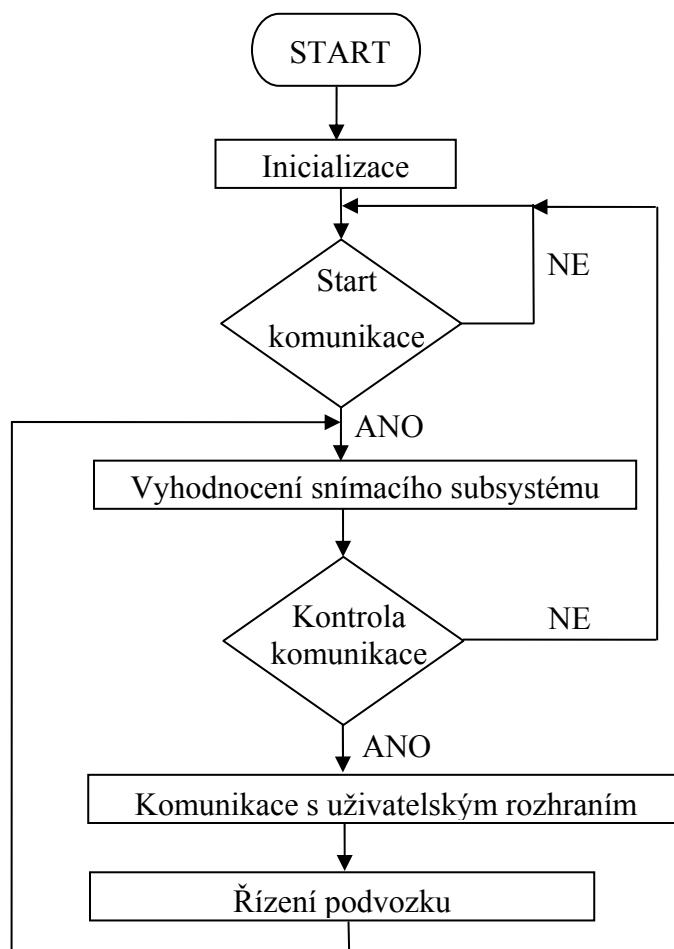
- Napětí 11,1V
- 2000mAh
- Hmotnost 68g
- Rozměry: 74 x 43 x 11 mm

Z daného sériového spojení dvou akumulátorů $U_{vcc} = 11,1 + 11,1 = 22,2V$ je napájena deska Sabertooth. Deska má stabilizovaný 5V výstup, ze kterého se dále napájí řídicí deska s mikrokontrolérem a snímači neelektrických veličin.

3.12 Popis zdrojového kódu mikrokontroléru

Zdrojový kód pro řídicí mikrokontroler XMega16 byl napsán a laděn v prostředí AVRStudio 4.18 v jazyce C.

Před hlavní main smyčkou programu jsou definovány jednotlivé globální proměnné jako adresy jednotlivých zařízení a makra pro výpočet přenosových rychlostí.



Obr. 3.24: Vývojový diagram programu mikrokontroléru

Po naběhnutí main funkce se nastaví hodnota interního oscilátoru na 16MHz, zinicilizují se UARTy, TWI, SPI, časovač a AD převodník s funkcemi:

UART_Init(int bsel, uint8_t bscale) - inicializuje UART na portu C a E na základě vstupních proměnných bsel a bscale, které udávají přenosovou rychlost.

TWI_Init(TWI_t *module, uint8_t baudRateRegisterSetting) – inicializuje TWI na portu modulu s přenosovou rychlostí udávanou v baudRateRegisterSetting. Výpočet hodnoty probíhá makrem v deklarační části.

SPI_Init(SPI_t *module) – inicializuje SPI na portu modulu v Master modu s frekvencí SCL 250kHz, polarita v neaktivitě je v logické 1, vzorkování dat je na sestupné hraně SCL.

Timer_Init(TC0_t *module, uint8_t prescaller, uint16_t period) – inicializuje časovač Timer udaného proměnnou module. Dalšími vstupními parametry funkce jsou

prescaller, který udává hodnotu předděličky a period udává počet ticků do vyvolání přerušení.

AD_init(void) – inicializuje AD převodník portu A na 12 bitový s výsledkem zarovnaným vlevo. Jako referenční hodnotu převodníku nastaví na interních 2,0625V.

Dále se nastaví nulová rychlost motorů podvozku, provede se reset modulu odometrie a povolí se přerušení od jednotlivých komponent.

Po inicializaci se v nekonečné smyčce sleduje hodnota globálních proměnných, které se aktualizují v příslušných přerušeních:

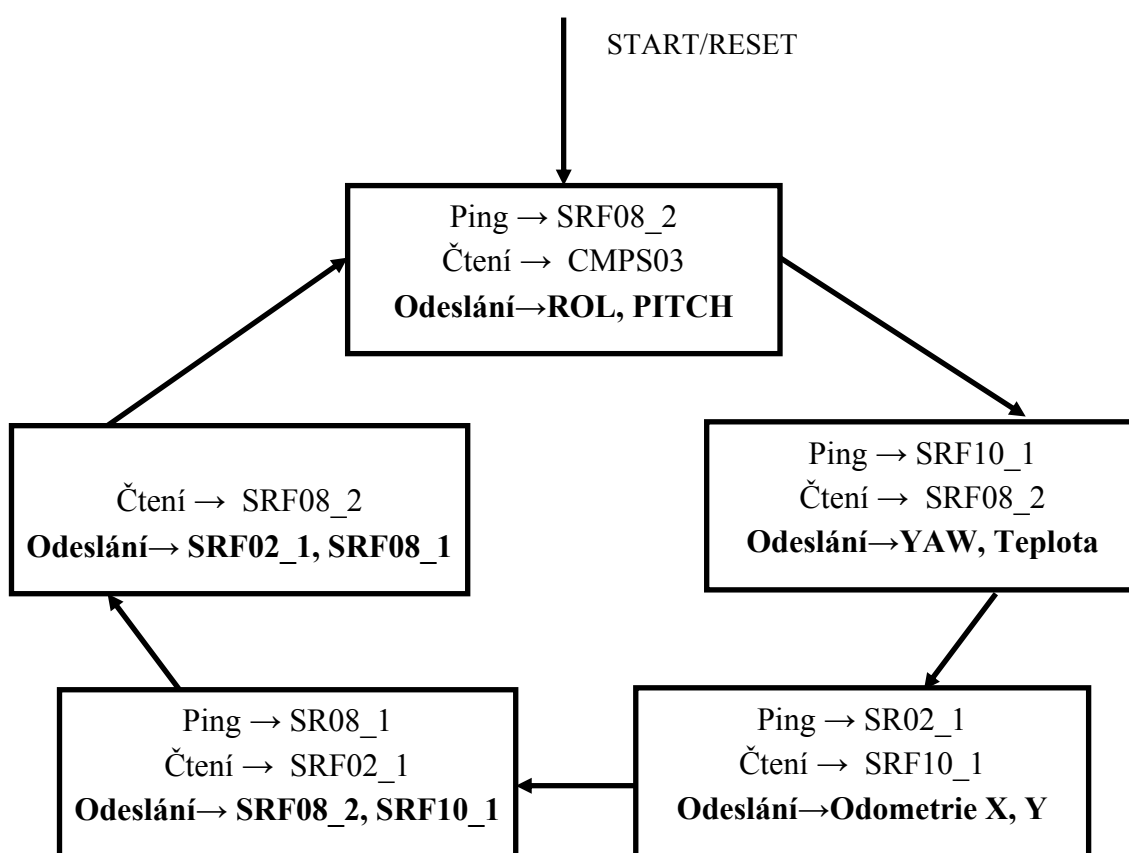
ISR(USARTC0_RXC_vect) – přerušení od USARTu připojeného na bezdrátovou komunikační jednotku. Provede čtení z datového registru USARTu, překontroluje platnost dat na základě porovnání paritního bitu generovaného funkcí **uint8_t ParityBit(uint8_t Data1)** a paritního bitu obsaženého v hlavičce zprávy. Na základě těchto dat se aktualizují globální proměnné pro jednotlivé rychlosti motorů a sleduje se případná ztráta komunikace.

ISR(ADCA_CH0_vect) – přerušení od nultého kanálu A/D převodníku ADCA. Cyklicky sleduje hodnoty napětí v jednotlivých osách x, y a z připojeného akcelerometru. Hodnoty postupně ukládá do polí AD_Array_X, AD_Array_Y a AD_Array_Z. V případě naplnění jednoho z polí o velikosti AD_LENGTH, se nastaví příslušná proměnná SortX, SortY nebo SortZ, což má za následek vyvolání třídění daného pole v hlavní main smyčce funkcí **void BubbleSort(uint16_t list[], uint8_t n)**. Po setřídění se vybere prostřední prvek pole reprezentovaný hodnotou AD_HALF a uloží se do příslušné globální proměnné Result_X, Result_Y nebo Result_Z. Tímto se vykonává mediánová filtrace dat. Výpočet hodnot naklonění jednotlivých os se převádí na základě už filtrovaných hodnot.

ISR(TCC0_OVF_vect) – přerušení od časovače TC0. Časovač je povolen přijetím datového paketu s hlavičkovou hodnotou 0xBB a ukončen při selhání komunikace. Funkce přerušení je rozdělena do pěti etap. Tyto etapy se vykonávají jednotlivě při každém přerušení, tzn. po 200 ms. Toto rozdělení slouží především pro časový odstup mezi snímáním vzdálenosti jednotlivými ultrazvukovými dálkoměry. Předjede se tak zachycení falešně vyslaných signálů, které mohou být vyslány jiným snímačem, než tím, kterým by byl zachycen. Grafické znázornění skenování snímacího subsystému je znázorněno na obrázku 3.25. Popis jednotlivých etap:

- Vyvolá se měření na snímači SRF08_2 (označování podle tabulky 2.7) a čtení hodnoty elektronického kompasu. Po vykonání měření se vyšle první datový paket s údaji ROL a PITCH.

- Vyvolá se měření na snímači SRF10_1 a čtení ze SRF08_2. Následně se vyšle druhý datový paket nesoucí informace YAW a Teplota.
- Vyvolá se měření na snímači SRF02_1 a čtení ze SRF10_1. Dále se vykoná čtení hodnot x a y odometrie, které jsou přímo vyslány v třetím datovém paketu.
- Vyvolá se měření na snímači SRF08_1 a čtení SRF02_1. Jako čtvrtý paket je vyslán údaj vzdálenosti získaný ze snímačů SRF08_2 a SRF10_1.
- V poslední etapě se vykoná čtení hodnoty ze snímače SRF08_1 a v posledním paketu jsou vyslány údaje vzdáleností ze snímačů SRF02_1 a SRF08_1.



Obr. 3.25: Bloková schéma skenování snímacího subsystému

Popis dalších využitých funkcí:

UART_ReceiveByte(USART_t *module) – kde modul je ukazatelem na modul UART-u na konkrétním portu. Návratová hodnota funkce je typu uint8_t a reprezentuje vyčtenou hodnotu z přijímacího registru daného UART-u.

UART_TransmitByte(USART_t *module, uint8_t data) – kde modul je ukazatelem na modul UART-u na konkrétním portu a proměnná data reprezentují hodnotu vyslanou na daný UART.

TWI_MasterWrite(TWI_t *module, uint8_t address, uint8_t *writeData, uint8_t bytesToWrite) – funkce vykonává zápis v master modu na TWI modul udaný ukazatelem module. Na adresu slave zařízení hodnoty address vyšle tok údajů o velikosti bytesToWrite z umístění writeData.

TWI_MasterRead(TWI_t *module, uint8_t address, uint8_t reg, uint8_t *readData, uint8_t bytesToRead) – funkce vykonává čtení v master modu na TWI modulu udaného ukazatelem module. Na adrese slave zařízení hodnoty address přečte z registru reg bytesToRead počet bytů, které uloží do readData.

ADNS_Write(SPI_t *module, uint8_t addr, uint8_t data) – funkce vykonává zápis na SPI modulu udaného ukazatelem module. Na adresu addr vybraného zařízení zapíše hodnotu data.

ADNS_Read(SPI_t *module, uint8_t addr) – funkce vykonává čtení z SPI modulu udaného ukazatelem module z adresy addr a vyčtenou hodnotu vrátí jako návratovou hodnotu typu uint8_t.

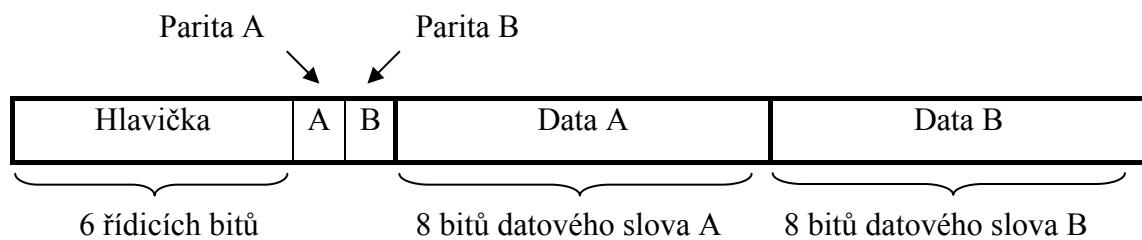
Zavedením sledování vzdálenosti sonarem umístěným v přední části mobilní platformy se dosáhla určitá forma autonomnosti robota. Při překročení 10 cm limitu přiblížení podvozku k překážce se podvozek zastaví, i kdyby mněl operátor tendence narazit do překážky.

Tato funkce se cyklicky vykonává po dobu čtení hodnot ultrazvukového snímače SRF10_1 v přerušení od časovače. Porovnává se naměřená hodnota. V případě přiblížení se lineárně snižuje rychlost podvozku, kterou udává operátor, až na nulovou rychlost.

Řídicí jednotka je ovládána prostřednictvím bezdrátové komunikační jednotky. Proto se cyklicky sleduje také komunikace mezi ovládacím programem REMOTE. Při sledování se využívá faktu, že aplikace REMOTE vysílá v pravidelných intervalech řídicí paket pro ovládání rychlosti motorů. V případě, že ve třech cyklech není přijat žádný paket (600 ms), můžeme vydedukovat, že se spojení přerušilo. Toto má za následek zastavení podvozku a konec zasílání údajů ze senzorického subsystému.

3.13 Komunikační protokol

Mezi řídicím počítačem a mikrokontrolérem je datová komunikace prostřednictvím bezdrátového spojení podle paketu na obr 3.24 .



Obr. 3.26: Popis přenosového paketu

Je to troj bytový datový paket, který má přenosovou kapacitu 16 bitů. První byte reprezentuje hlavičku zprávy, prvních 6 bitů je řídicích (Tab. 3.14) a následující 2 bity slouží jednotlivě jako CRC kontrolní součet. Paritní bit je redundantní bit přidáný k datovému slovu a obsahuje paritní informaci o počtu jedničkových bitů ve slově. 7 bit hlavičky paketu se nastavuje podle parity datového slova A a 8 bit podle parity datového slova B. Používá se sudá parita, což znamená, že počet jedniček spolu i s paritním bitem je sudý. Paritní bit lze získat prostou funkcí XOR mezi všemi bity datového slova. Za pomoci paritního bitu je možno detekovat jednu chybu nebo jeden výmaz přenesené informace.

| Bitová kombinace hlavičky | Data A | Data B |
|---------------------------|-------------------------------------|-----------------------|
| 0010 01xx | Pitch | Roll |
| 0010 10xx | Raw | Temp |
| 0010 11xx | Vzdálenost ze SRF08_2 | Vzdálenost ze SRF10_1 |
| 0011 00xx | Vzdálenost ze SRF02_1 | Vzdálenost ze SRF08_1 |
| 0011 01xx | Hodnota odometrie X | Hodnota odometrie Y |
| 1000 11xx | Rychlost motoru 1 | Rychlost motoru 2 |
| 1100 0011 | Identifikace zařízení (ASCII kód ?) | |
| 1011 1011 | Synchronizace a začátek vysílání | |

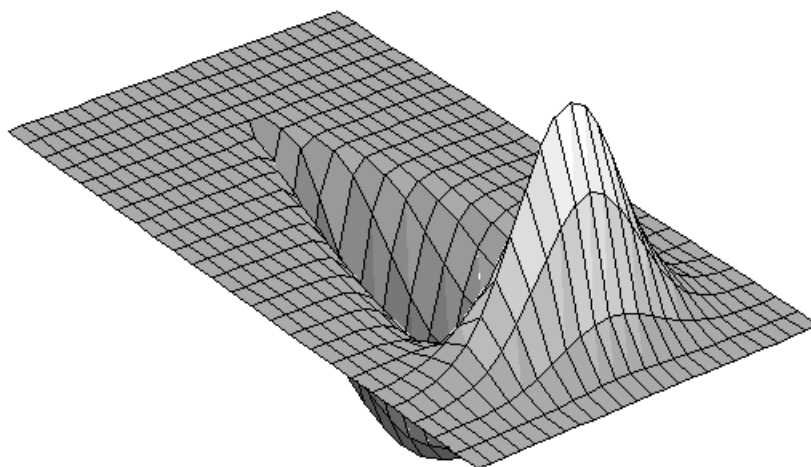
Tab. 3.14: Popis řídicích bitů hlavičky přenosového paketu

3.14 Mřížka obsazenosti

Pokud zapneme robota v neznámém prostředí, robot získává informace o daném neznámém prostředí jedině za pomoci svých snímačů. Měření prováděné těmito snímači je zatíženo řadou chyb jako jsou např. nepřesnost měření vzdálenosti nebo zejména zrcadlový efekt (specular reflection). Proto se při zpracování naměřených dat využívá zejména teorie pravděpodobnosti a statické zpracování dat.

Pro určení pozic překážek a objektu byl použit postup s využitím rozdělení spojitého prostředí do diskrétní oblasti do takzvaných buněk (cells). Každé buňce je pak přiřazena hodnota pravděpodobnosti jakou může být obsazena překážkou, případně jiným objektem. Tyto buňky dohromady tvoří mřížku nazvanou mřížka obsazenosti (Occupancy grid, Certainty grid, Inference grid, Evidence grid).

Malá mobilní platforma je vybavena 4 ultrazvukovými snímači vzdálenosti – sonary. Sonar po vyslání zvukové vlny čeká na odraz od okolních objektů. Pokud se mu vrátí odraz od nějaké překážky, můžeme předpokládat, že někde v prostoru, kde se odrazila zvuková vlna, je nějaký objekt. Není možné přesně určit vzdálenost a směr překážky, protože ultrazvukový sonar vysílá signál v jednom hlavním laloku s několika postranními laloky. Je třeba počítat pouze s pravděpodobností.



Obr. 3.27: Model vyzařování ultrazvukového sonaru

Stav buňky je reprezentován pravděpodobností jejího obsazení. Buňka může nabývat dvou základních stavů: obsazená – occupied ($S_{Ci} = OCC$) nebo prázdná – empty ($S_{Ci} = EMP$). Tato pravděpodobnost je ale určitým způsobem založena na měřených datech, což znamená podmíněnou pravděpodobnost. Na její výpočet je nejvhodnější Bayesův vzorec:

$$p(S_{Ci} = OCC | r = r_{xy}) = \frac{p(r = r_{xy} | S_{Ci} = OCC) \cdot p(S_{Ci} = OCC)}{p(r = r_{xy})} \quad (2.6)$$

kde r_{xy} značí jednu změřenou hodnotu vrácenou snímačem a xy je index buňky mřížky.

V Bayesově vzorci figuruje pravděpodobnost obsazení buňky $p(S_{CI} = OCC)$. Tato hodnota má hlavní význam pro mřížku obsazenosti, umožňuje její rekurzivní aktualizaci podle následujícího vzorce:

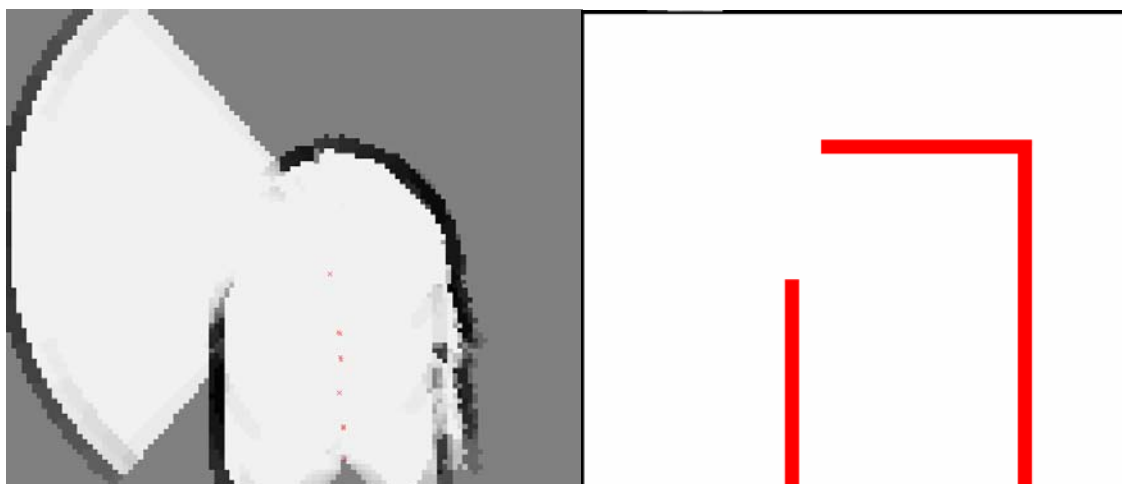
$$p(S_{CI_{t+1}} = OCC | r_t = r_{xy}) = \frac{p(r = r_{xy} | S_{CI_t} = OCC) \cdot p(S_{CI_t} = OCC)}{p(r = r_{xy} | S_{CI_t} = OCC) \cdot p(S_{CI_t} = OCC) + p(r = r_{xy} | S_{CI_t} = EMP) \cdot p(S_{CI_t} = EMP)} \quad (2.7)$$

Hodnota pravděpodobnosti $p(S_{CI_{t+1}} = OCC | r_t = r_{xy})$ v čase měření $t + 1$ se získá na základě předchozí pravděpodobnosti obsazení $p(S_{CI_t} = OCC)$ v čase t . Před prvním měřením se hodnota pravděpodobnosti obsazení nastaví na 0,5, protože robot o okolním světě nic neví. Proto je stejná pravděpodobnost na obsazenou i prázdnou buňku.

Daná teorie a vzorce platí jenom při:

- Mapovaný prostor se v čase jinak nemění, je statický. Robot nemůže vykonávat žádné akce, které by měnily daný prostor.
- Pozice robota musí být známa přesně.

Na základě práce Bc. Iva Hermana bylo zaimplementováno do uživatelského rozhraní programu REMOTE i vytváření mapy prostředí metodou mřížky obsazenosti. Následující obrázek 3.26 byl namapován a vytvořen na základě údajů z 3 sonarů malé mobilní platformy. A to z údajů sonarů S1, S3 a S4.



Obr. 3.28: Vlevo – mapa vytvořená uživatelským rozhraním mobilní platformy, vpravo – náčrves mapovaného prostoru

3.15 Popis řídicího programu REMOTE

Řídicí program REMOTE je uživatelským rozhraním pro ovládání malé mobilní platformy. Je napsán a odladěn v programovacím jazyku C# v prostředí Microsoft Visual Studio 2008. Jedná se o windows form aplikaci.

Uživatelským vstupem do programu jsou data získaná z joysticku. Na zpracování dat z joysticku slouží Windows API DirectInput. Číselná hodnota osy x a y joysticku je upravena podle následujícího vzorce, abychom obdrželi přímo hodnoty rychlostí pro jednotlivé motory mobilní platformy:

$$R_1 = \frac{X + Y}{516} + 64 \quad (2.8)$$

$$R_2 = \frac{X - Y}{516} + 64 \quad (2.9)$$

X, Y – výchylka joysticku ve směru x a y, rozsah hodnot 0 – 65536

R_1 – rychlost motoru 1, rozsah rychlostí: 255 – 128

R_2 – rychlost motoru 2, rozsah rychlostí: 127 – 1

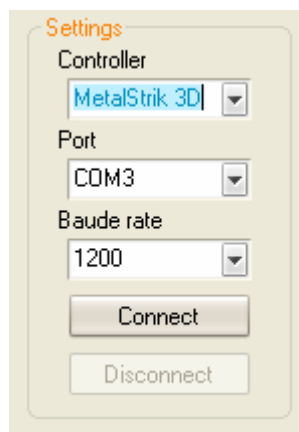
Pro příjemnější řízení z pohledu operátora se hodnoty x, y mohou umocnit. Kvadrát hodnot má za následek plynulejší rozjezd podvozku, ale menší citlivost ve velkých rychlostech. Uživatelské rozhraní se skládá ze šesti částí:

- nastavení komunikačního portu
- umělý horizont
- kompas
- ukazatel přenosové rychlosti
- navigační mapa
- stavový řádek

Rozhraní je znázorněno v příloze č. 16.

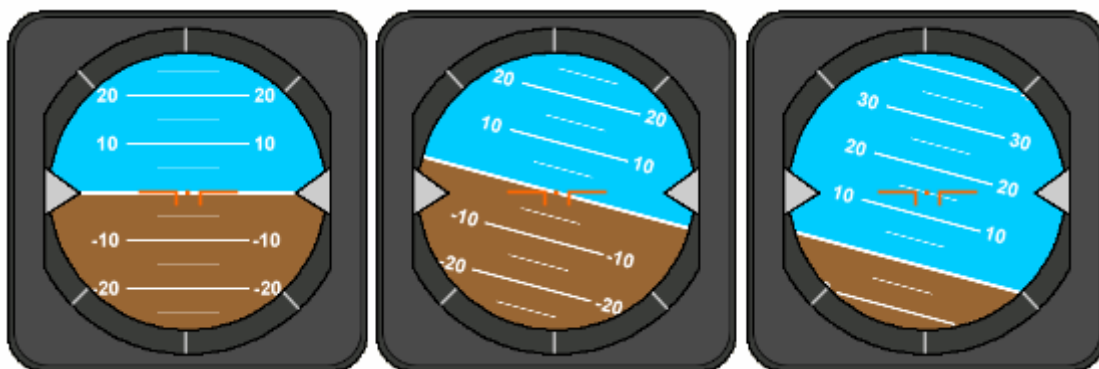
Pro komunikaci s mobilní platformou a počítačem slouží bezdrátové spojení prostřednictvím Bluetooth sériové linky. Program umožňuje vybrat číslo COM portu a definovat přenosovou rychlost. Funkce **GetPortNames()** vrací seznam COM portů přístupných operačnímu systému. Nevýhodou je, že funkce nefunguje spolehlivě s virtuálními Bluetooth COM porty. Někdy seznam přístupných portů neobsahoval Bluetooth port. Dále program v části nastavení komunikačního portu nabízí výběr ze seznamu ovládacích prvku – gamecontrollerů. Předdefinovaná hodnota přenosové rychlosti aplikace je na 1200Bd (rychlost udaná mobilní platformou). Ovládacími

prvky, které jsou zobrazeny na obrázku 3.27, je však možné měnit jak přenosovou rychlost, tak i číslo komunikačního COM portu.



Obr. 3.29: Ovládací prvky pro základní nastavení aplikace REMOTE

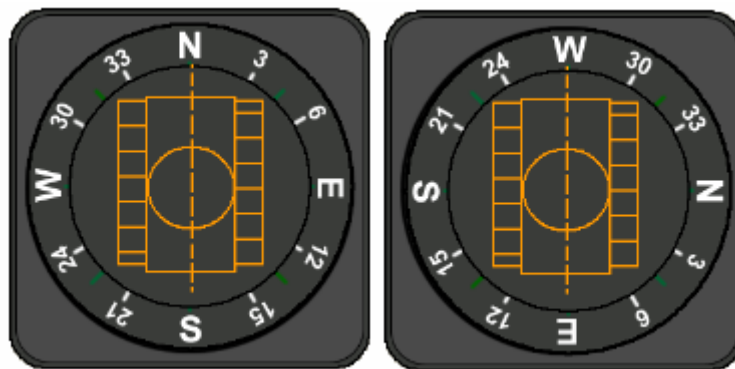
Umělý horizont je grafická animace přístroje zobrazujícího horizont prostředí z pohledu mobilní platformy. Hnědě vyplněná část reprezentuje pevninu horizontu, jehož poloha se mění v závislosti na naklonění podvozku pitch (předo/zadní sklon) a roll(levo/pravý náklon). Na čelním panelu přístroje je vodorovně cejchovaná hlavní stupnice po 10° .



Obr. 3.30: Ukázka umělého horizontu, vlevo – ROLL = 0, PITCH = 0, uprostřed – ROLL = 15°, PITCH = 0, vpravo – ROLL = 15°, PITCH = 15°

Jak umělý horizont tak i kompas byly sestrojeny na základě ukazatelů z kokpitu letadla, aby údaje zobrazené na nich byly pro operátora jednoznačně průkazné.

Grafická animace kompasu je založena na rotaci ocejchovaného prstence kolem statického obrazu robota. Na kompasu je vlastně možné sledovat hodnotu yaw (pootočení/směr).



Obr. 3.31: Ukázka umělého horizontu, vlevo – podvozek směřující na sever, vpravo – podvozek směřující na západ

Ukazatel přenosové rychlosti je číselná reprezentace aktuální hodnoty datového toku směrem k mobilní platformě – UP a od mobilní platformy DOWN v bitech/sekundu.

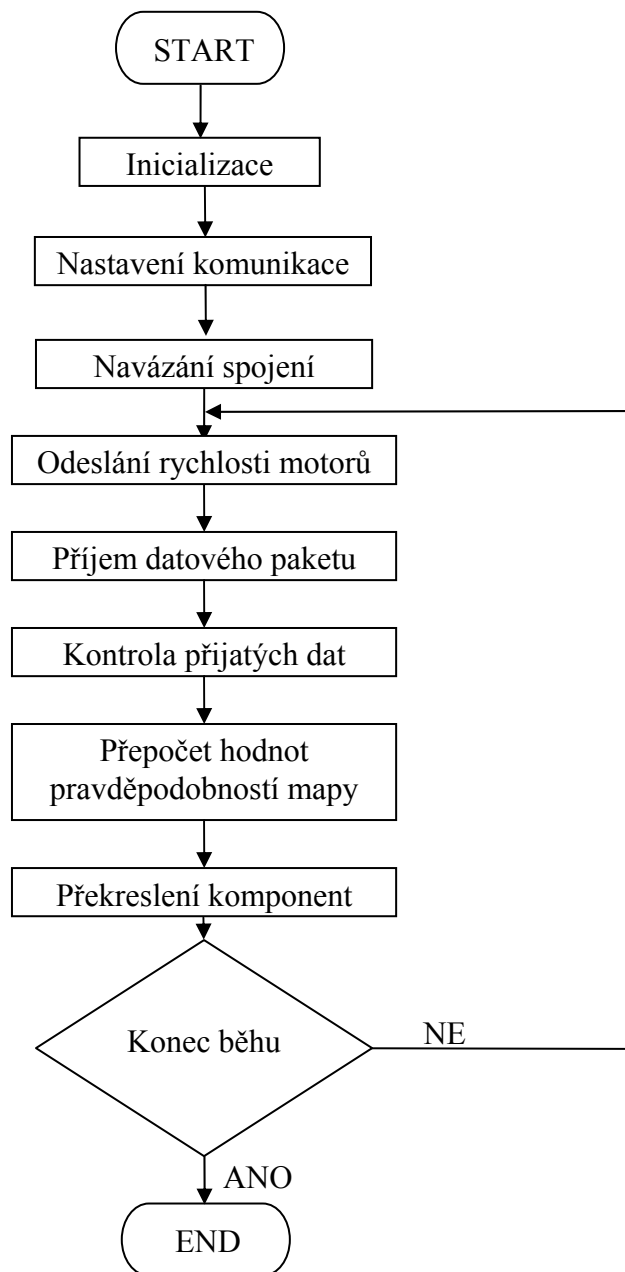
Stavový řádek obsahuje vždy aktuální informaci běhu programu, např. o úspěšném připojení COM portu, úspěšném uložení mapy, případně varovné signály.

Poslední část programu je navigační mapa, která je založena na implementaci třídy 2DSvet [20]. Jedná se o vykreslování vypočtených hodnot pravděpodobností, které se vypočítaly pro sonarové modely typu Murphy. Odometrie mobilní platformy pracovala s velkou chybovostí. Proto byla použita možnost přepínání zadávání trasy robota do mapy manuálně (klikáním myši) a automaticky podle přijatých údajů odometrie.

Aby se předešlo blikání mezi jednotlivými překreslovacími přechody, tak vykreslování komponent je vykonávanou s dvojitým zásobníkem (double buffering).

Pro vyvolání přerušení od příjmu údajů ze sériové linky COM se instancí `new SerialDataReceivedEventHandler (DataRecevedHandler)` vytvoří handler `DataRecevedHandler`. V těle této funkce se vykonává kontrola přijatých údajů porovnáním paritních bitů hlavičky paketu a paritních bitů vypočtených funkcí `ParityBit (int Data1)`.

Funkce `ParityBit (int Data1)` pracuje na stejném algoritmu jako funkce vykonávající výpočet v mikrokontroléru. Jednotlivé bity binární reprezentace vstupní hodnoty `Data1` se postupně logicky posouvají vpravo. Na toto číslo je implementována bitová maska, která vynuluje všechny bity proměnné, až na nultý bit. Sčítáním těchto maskovaných nultých bitů je výsledkem buď liché nebo sudé číslo, což rozhoduje o přidělení paritního bitu.



Obr. 3.32: Základní vývojový diagram programu REMOTE

4 ZÁVĚR

Po dobu realizace této diplomové práce jsem si osvojil důležité poznatky o mikrokontroléru XMega16 A4, a to hlavně při ladění řídicího programu.

Takto jsem si ověřil i funkčnost zkonstruovaného programátoru USBTiny MkII, přes který se nahrával laděný kód programu do mikrokontroléru.

Dalším dosaženým cílem bylo navázání komunikace se snímači neelektrických veličin a zpracování jejich údajů. Navrhnul jsem rozmístění minimálního počtu snímačů za účelem minimálního datového toku, aby se podle informací z nich dala mobilní platforma navigovat. Po dobu testování jsem také zjistil, že na vytvoření navigační mapy postačí pouze 3 ultrazvukové dálkoměry, čímž by se navržený komunikační protokol mohl ještě více zredukovat.

Navrhnul jsem a odladil program REMOTE, který má sloužit jako uživatelské rozhraní operátora. Komunikace mezi mobilní platformou a uživatelským rozhraním REMOTE probíhá podle navrženého datového paketu a bylo použito Bluetooth spojení. Práce je koncipovaná tak, aby Bluetooth komunikační jednotka mohla být snadno nahrazena komunikační jednotkou s lepšími parametry – ZigBee.

Nedostatek robota spočívá ve velké nepřesnosti použitého modulu odometrie. Tato nepřesnost by se dala odstranit použitím vhodné optiky (optika webkamery) a dosažením ostřejšího obrazu pro odometrii.

Výsledkem této diplomové práce je malá mobilní platforma, která je schopna na základě svého snímacího subsystému zprostředkovat data z prostředí uvnitř budov pro navigaci operátorem.

Práce byla rovněž úspěšně obhájena na konferenci a soutěži Student EEICT 2011. Výsledkem bylo zveřejnění článku ve sborníku prací:

MASTER STUDENTS, et al. *Student EEICT proceedings of the 17th conference*. Brno : NOVAPRESS s.r.o., 2011. Control System for a Small Mobile Robot, s. 352.

Literatura

- [1] *ATMEL 8-bit AVR XMEGA A Microcontroller* [datasheet]. San Jose : Atmel Corporation, 2009. Dostupné z WWW:
<http://www.atmel.com/dyn/resources/prod_documents/doc8077.pdf>
- [2] HANKOVEC, David. *Vývoj a výroba elektroniky na zakázku* [online]. 2002 – 2011 [cit. 2011-05-15]. I2C - o co jde a jak pracuje. Dostupné z WWW: <<http://dhservis.cz/iic.htm>>
- [3] HANKOVEC, David. *Vývoj a výroba elektroniky na zakázku* [online]. 2002 – 2011 [cit. 2011-05-15]. Sériový kanál - o co jde a jak pracuje. Dostupné z WWW: <<http://dhservis.cz/serial.htm>>
- [4] *AVR1309: Using the XMEGA SPI* [datasheet]. San Jose : Atmel Corporation, 2008. Dostupné z WWW:
<http://www.atmel.com/dyn/resources/prod_documents/doc8057.pdf>
- [5] *AVR1300: Using the Atmel AVR XMEGA ADC* [datasheet]. San Jose : Atmel Corporation, 2010. Dostupné z WWW:
<http://www.atmel.com/dyn/resources/prod_documents/doc8032.pdf>
- [6] *The I²C-Bus Specification V2.1* [datasheet]. California : PHILIPS, 2000. Dostupné z WWW:
<http://www.nxp.com/acrobat_download2/literature/9398/39340011.pdf>
- [7] MATOUŠEK, David. *Práce s mikrokontroléry ATMEL AVR*. Praha : Technická literatura BEN, 2006. 363 s
- [8] *MAXIM 3.0V to 5.0V, Low-Power, up to 1Mbps, True RS-232 Transceivers* [datasheet]. Sunnyvale: Maxim Integrated Products, 2007. Dostupné z WWW:
<<http://datasheets.maxim-ic.com/en/ds/MAX3222-MAX3241.pdf>>
- [9] *Sabertooth 2x10 User's Guide* [datasheet]. Akron : DimensionEngineering , 2007. Dostupné z WWW:
<<http://www.dimensionengineering.com/datasheets/Sabertooth2x10.pdf>>
- [10] *Robot electronics* [online]. 2002 [cit. 2011-05-15]. SRF02 Ultra Sonic Range Finder. Dostupné z WWW:
< <http://www.robot-electronics.co.uk/htm/srf02tech.htm>>
- [11] *Robot electronics* [online]. 2002 [cit. 2011-05-15]. SRF08 Ultra Sonic Range Finder. Dostupné z WWW:
<<http://www.robot-electronics.co.uk/htm/srf08tech.shtml>>
- [12] *Robot electronics* [online]. 2002 [cit. 2011-05-15]. SRF10 Ultra Sonic Range Finder. Dostupné z WWW:
< <http://www.robot-electronics.co.uk/htm/srf10tech.htm>>

- [12] *Robot electronics* [online]. 2007 [cit. 2011-05-15]. CMPS03 Compass Module. Dostupné z WWW: <<http://www.robot-electronics.co.uk/htm/cmeps3tech.htm>>
- [14] *Modul tříosého akcelerometru ACC7260* [datasheet]. Beroun: Snail Instruments, 2007. Dostupné z WWW: <<http://www.snailinstruments.com/docs/acc7260.pdf>>
- [15] *ADNS-5030 Low Power Optical Mouse Sensor* [datasheet]. United States: Avago Technologies 2009. Dostupné z WWW: <<http://www.avagotech.com/docs/AV02-0113EN>>
- [16] H.R. Everett, *Sensors for Mobile Robots - Theory and Application*, A K Peters, 1995
- [17] *LUFA* [online]. 2010 [cit. 2011-05-15]. Four Walled Cubicle. Dostupné z WWW: <<http://www.fourwalledcubicle.com/LUFA.php>>
- [18] *AVR1612: PDI programming driver* [datasheet]. San Jose : Atmel Corporation. 2010 Dostupné z WWW: <http://www.atmel.com/dyn/resources/prod_documents/doc8282.pdf>
- [19] *Vše o elektronice a programování* [online]. 1997 - 2009 [cit. 2011-05-15]. Dostupné z WWW: <<http://hw.cz/>>
- [20] HERMAN, I. Mřížky obsazenosti pracovního prostoru robota. Brno, Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 82 s.

Seznamy

Seznam zkratk

| | |
|------------------|--|
| A/D | Analogově – digitální převodník |
| CAD | Computer Aided Design, počítačem podporované projektování |
| CCD | Charge – Coupled Device, součástka používaná na snímání obrazu |
| CMOS | Complementary Metal–Oxide–Semiconductor, doplňující se kov -oxid- polovodič |
| COM | Standard definující asynchronní sériovou komunikaci pro přenos dat |
| D/A | Digitálně – analogový převodník |
| DC | Direct current, označení jednosměrných motorů |
| IAS | Image Acquisition System, systém pro snímání obrazu |
| I ² C | Inter – Integrated Circuit |
| LED | Light Emitting Diode, elektroluminiscenční dioda |
| NVM | Non – Volatile memory, Energeticky nezávislá paměť |
| PWM | Pulse – Width modulation |
| RISC | Reduced Instruction Set Computer, počítač s redukovanou instrukční sadou |
| SMD | Surface Mount Devices |
| SMBus | System Management Bus |
| SCK | Serial clock, sériové hodiny na synchronizaci přenosu dat |
| SPI | Serial Programming Interface, rozhraní na sériové programování |
| TOF | Time of Flight, doba letu |
| TTL | Transistor-Transistor-Logic, tranzistorově-tranzistorová logika |
| TWI | Two-wire serial interface – dvojvodičové sériové rozhraní |
| USART | Addressable universal Synchronous Asynchronous Receiver Transmitter |

Seznam obrázků

| | |
|--|----|
| Obr.2.1: Blokové schéma mikrokontroléru XMega [1]..... | 10 |
| Obr. 2.2: Popis vývodů mikrokontroléru XMega16 A4 [1] | 12 |
| Obr. 2.3: Asynchronní sériový přenos | 13 |
| Obr. 2.4: Blokové schéma A/D převodníku [5]..... | 15 |
| Obr. 2.5: Pipelining A/D převodníku [5]..... | 16 |
| Obr. 2.6: Znaménkový a bezznaménkový převod A/D [5]..... | 17 |
| Obr. 2.7: Přepojení sběrnice I ² C | 18 |
| Obr. 2.8: Ukázka přenosu adresy po sběrnici I ² C [1] | 19 |
| Obr. 2.9: Blokové schéma SPI komunikace [4]..... | 20 |
| Obr. 3.1: Blokové schéma řídicího systému | 21 |
| Obr. 3.2: Zapojení obvodu MAX 3232..... | 22 |
| Obr. 3.3: Popis konektoru RS 232 | 22 |
| Obr. 3.4: Obousměrný převodník úrovní [8] | 23 |
| Obr. 3.5: Sabertooth 2x10 [9] | 24 |
| Obr. 3.6: Zapojení Sabertooth 2x10 v jednoduchém sériovém módu [9]..... | 25 |
| Obr. 3.7: Konfigurační DIP přepínač modulu Sabertooth [9] | 25 |
| Obr. 3.8: Zapojení vývodů dálkoměru SRF02 [10]..... | 26 |
| Obr. 3.9: Vyzařovací charakteristika SRF02 [10] | 27 |
| Obr. 3.11: Vyzařovací charakteristika SRF08 [11] | 28 |
| Obr. 3.13: Vyzařovací charakteristika SRF10 [12] | 29 |
| Obr. 3.14: Modul elektronického kompasu CMPS03 [12]..... | 31 |
| Obr. 3.15: Trojosý akcelerometr ACC7260..... | 32 |
| Obr. 3.16: Výstupní charakteristika akcelerometru ACC7260..... | 33 |
| Obr. 3.17: Blokové schéma ADNS-5030 [15]..... | 34 |
| Obr. 3.18: Číselná reprezentace pohybu v ose x a y [15] | 35 |
| Obr. 3.19: Reprezentace pixelů snímaného obrazu [15]..... | 36 |
| Obr. 3.20: Rozložení snímačů na mobilní platformě – pohled shora | 37 |
| Obr. 3.21: Složky gravitačního zrychlení působící na osu X a Z akcelerometru. | 39 |
| Obr. 3.22: Hardwarové spojení programátoru [18] | 40 |
| Obr. 3.23: Detail elektródy vytvořeného ochranního kondenzátoru na DPS | 42 |
| Obr. 3.24: Vývojový diagram programu mikrokontroléru | 44 |

| | |
|---|----|
| Obr. 3.25: Bloková schéma skenování snímacího subsystému | 46 |
| Obr. 3.26: Popis přenosového paketu | 48 |
| Obr. 3.27: Model vyzařování ultrazvukového sonaru | 49 |
| Obr. 3.28: Vlevo – mapa vytvořená uživatelským rozhraním mobilní platformy, vpravo – náčrt mapovaného prostoru | 50 |
| Obr. 3.29: Ovládací prvky pro základní nastavení aplikace REMOTE..... | 52 |
| Obr. 3.30: Ukázka umělého horizontu, vlevo – $ROLL = 0$, $PITCH = 0$, uprostřed – $ROLL = 15^\circ$, $PITCH = 0$, vpravo – $ROLL = 15^\circ$, $PITCH = 15^\circ$ | 52 |
| Obr. 3.31: Ukázka umělého horizontu, vlevo – podvozek směřující na sever, vpravo – podvozek směřující na západ..... | 53 |
| Obr. 3.32: Základní vývojový diagram programu REMOTE..... | 54 |

Seznam tabulek

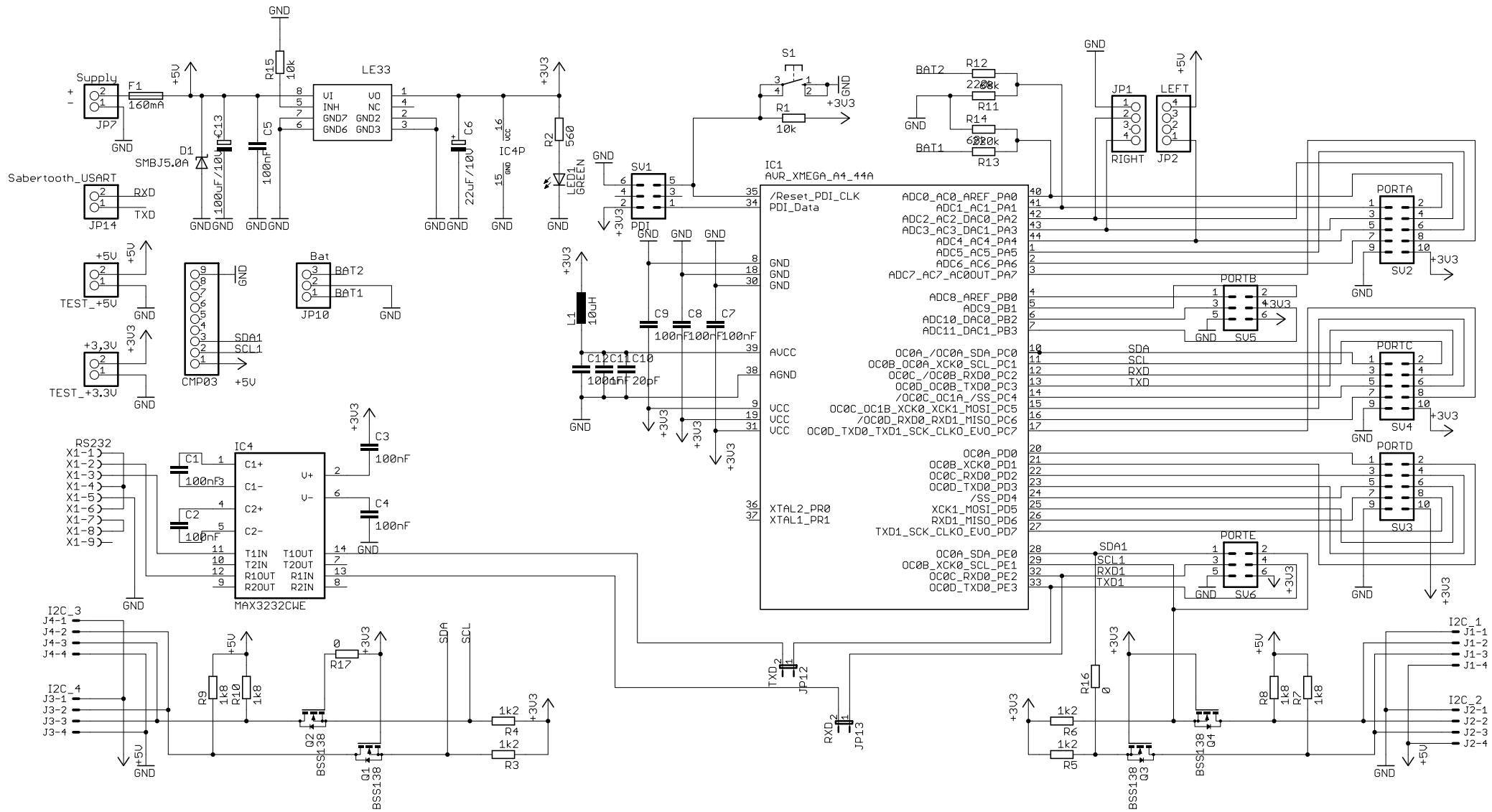
| | |
|---|----|
| Tab. 3.1: Popis komunikačních pinů mikrokontroléru | 21 |
| Tab. 3.2: Popis kontaktů konektoru RS 232 | 23 |
| Tab. 3.3: Popis DIP přepínače modulu Sabertooth..... | 25 |
| Tab. 3.4: Popis registrů dálkoměru SRF02..... | 27 |
| Tab. 3.5: Popis registrů dálkoměru SRF08..... | 28 |
| Tab. 3.6: Popis registrů dálkoměru SRF10..... | 29 |
| Tab. 3.7: Nastavení adres snímaču SRFxx | 29 |
| Tab. 3.8: Popis příkazů dálkometrů..... | 30 |
| Tab. 3.9: Popis registrů kompasu CMPS03 | 31 |
| Tab. 3.10: Konfigurace měřicího rozsahu modulu ACC7260 | 33 |
| Tab. 3.11: Seznam registrů obvodu ADNS-5030 | 35 |
| Tab. 3.12: Označení jednotlivých snímačů..... | 38 |
| Tab. 3.13: Zapojení konektorů programátoru..... | 41 |
| Tab. 3.14: Popis řídicích bitů hlavičky přenosového paketu..... | 48 |

Seznam příloh

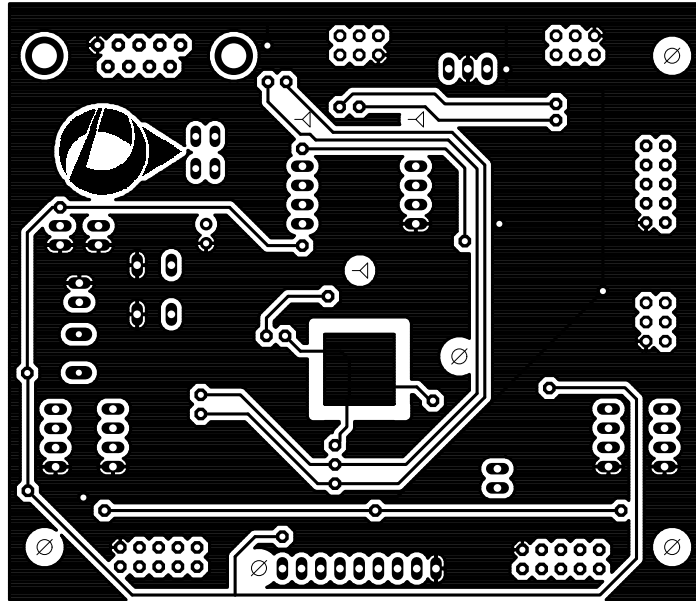
- Příloha 1: Schéma zapojení řídicí jednotky malého mobilního robota
- Příloha 2: Plošný spoj řídicí jednotky TOP
- Příloha 3: Plošný spoj řídicí jednotky BOTTOM
- Příloha 4: Osazovací plán plošného spoje řídicí jednotky TOP
- Příloha 5: Osazovací plán plošného spoje řídicí jednotky BOTTOM
- Příloha 6: Schéma programátoru USBTiny MkII pro procesory Atmel AVR
- Příloha 7: Plošný spoj programátoru USBTiny MkII
- Příloha 8: Osazovací plán plošného spoje programátoru USBTiny MkII BOTTOM
- Příloha 9: Osazovací plán plošného spoje programátoru USBTiny MkII TOP
- Příloha 10: Schéma zapojení modulu odometrie
- Příloha 11: Vpravo -plošný spoj modulu odometrie – BOTTOM, vlevo - Osazovací plán plošného spoje modulu odometrie
- Příloha 12: Osazený plošný spoj modulu odometrie
- Příloha 13: Realizace DPS řídicí jednotky pro malého mobilního robota
- Příloha 14: Realizace programátoru USBTiny MkII
- Příloha 15: Uživatelské rozhraní programu REMOTE
- Příloha 16: Malý mobilní robot osazený řídicí jednotkou a snímacím subsystémem

Obsah příloženého CD

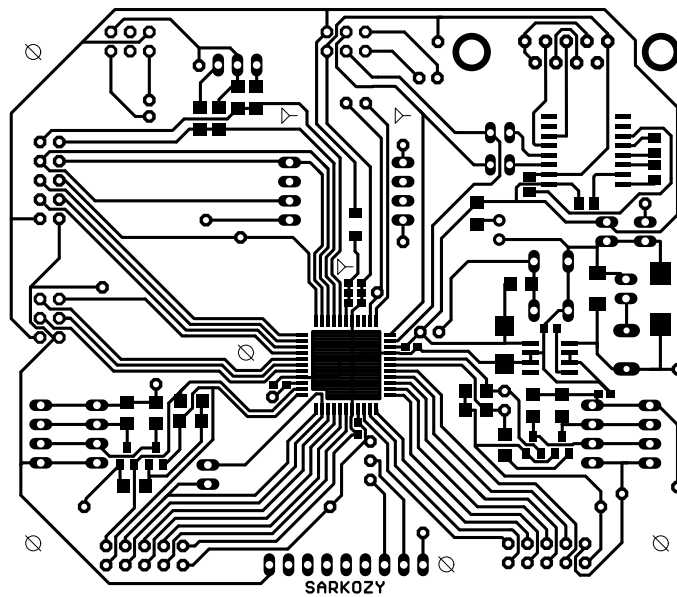
| | |
|-------------------------------|--|
| \\PRACE_PDF | elektronická forma diplomové práce v PDF |
| \\PRACE_DOC | elektronická forma diplomové práce v DOC |
| \\EAGLE_5.4 | schéma zapojení a návrhy DPS |
| \\AVR_STUDIO_4.18 | zdrojový kód mikrokontroléru XMega16 A4 |
| \\MS_VISUAL_STUDIO_2008 | zdrojový kód aplikace REMOTE |
| \\USB_TINY_MKII | schéma a zdrojové kódy pro programátor |
| \\REMOTE | EXE soubor ovládací aplikace |
| \\DATASHEETS | katalogové listy |



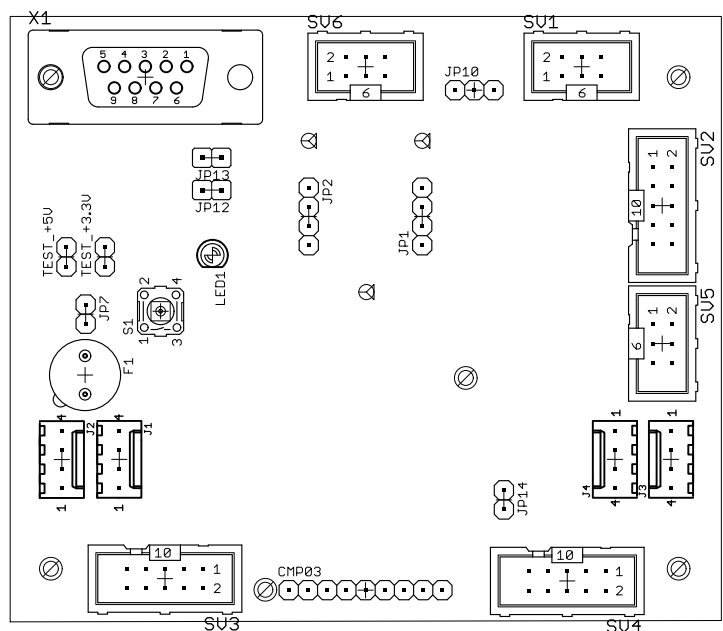
Příloha 1: Schéma zapojení řídicí jednotky malého mobilního robota



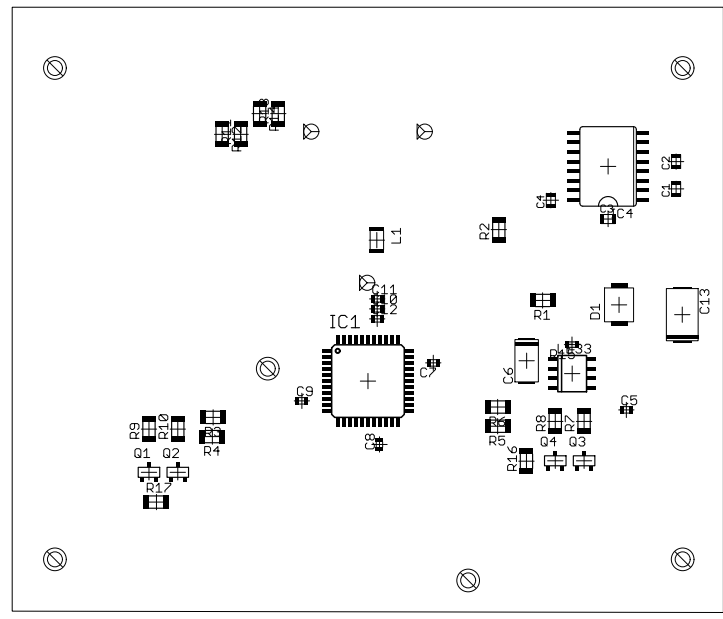
Příloha 2: Plošný spoj řídicí jednotky TOP



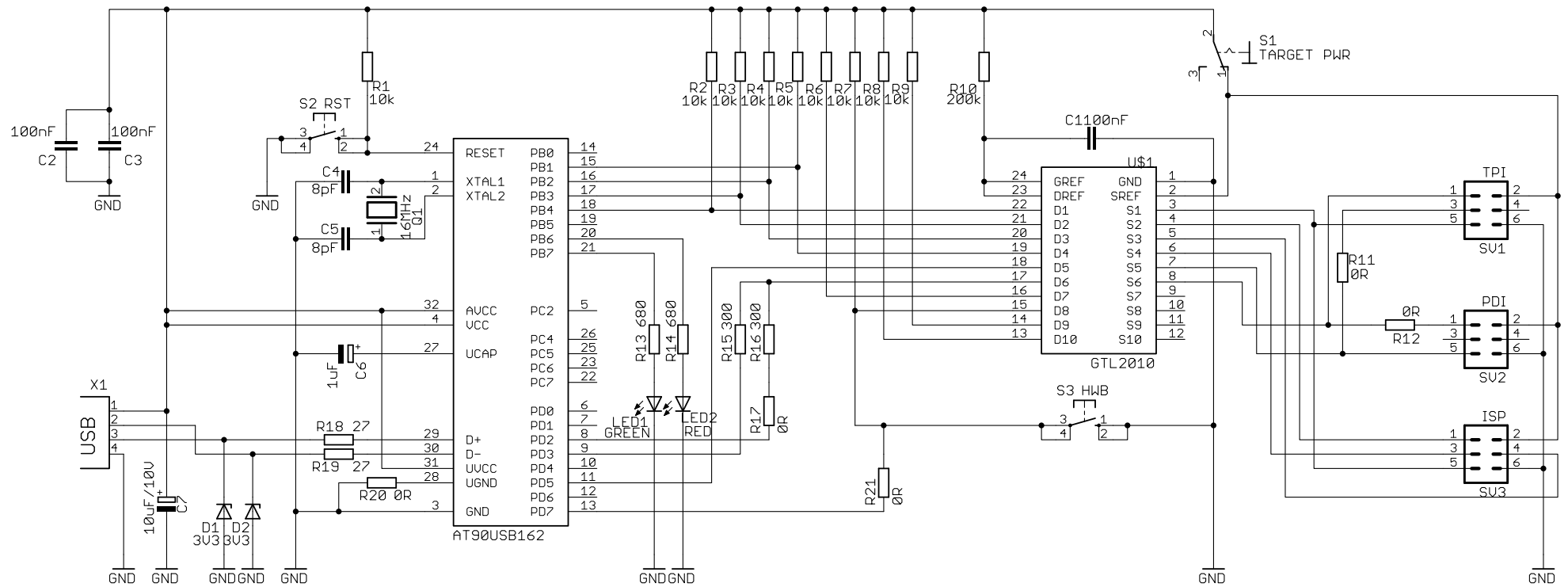
Příloha 3: Plošný spoj řídicí jednotky BOTTOM



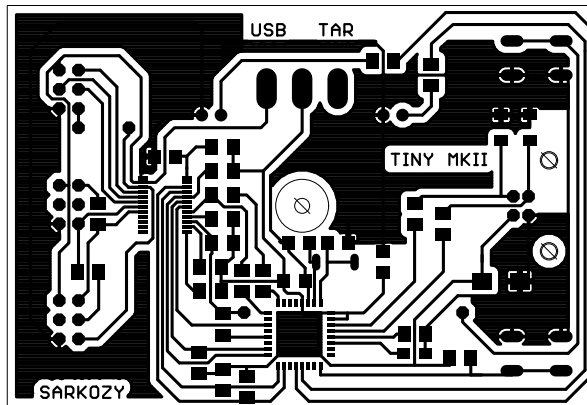
Příloha 4: Osazovací plán plošného spoje řídicí jednotky TOP



Příloha 5: Osazovací plán plošného spoje řídicí jednotky BOTTOM

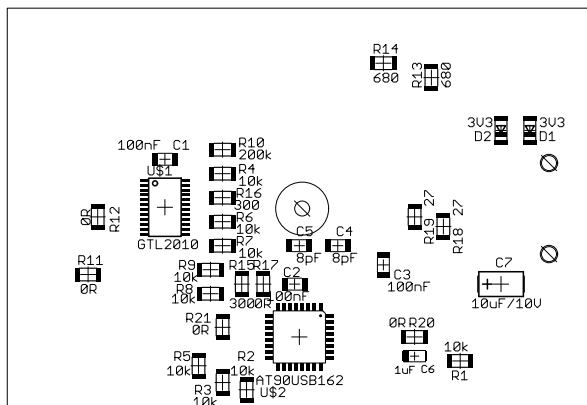


Příloha 6: Schéma programátoru USBTiny MkII pro procesory Atmel AVR

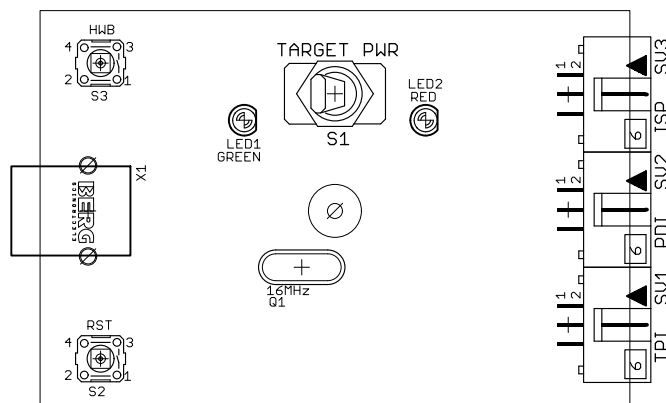


Příloha 7: Plošný spoj programátoru USBTiny MkII

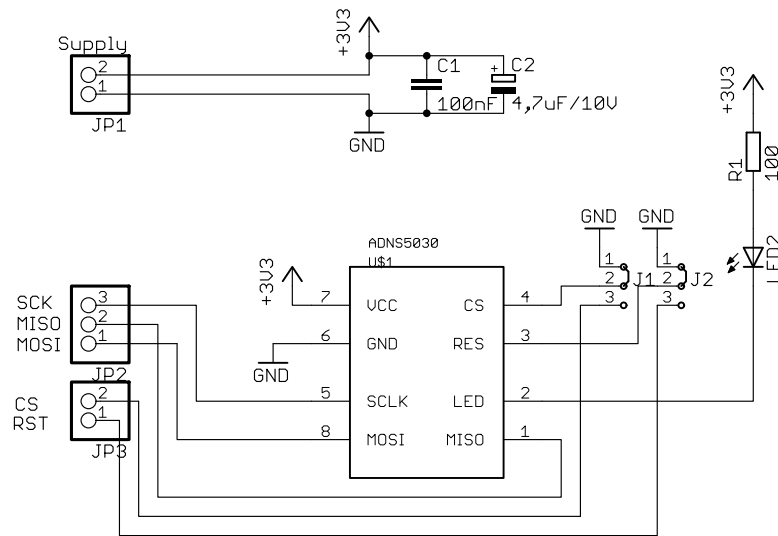
BOTTOM



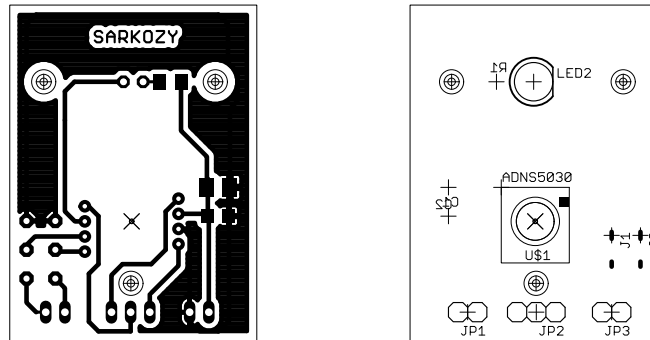
Příloha 8: Osazovací plán plošného spoje programátoru USBTiny MkII BOTTOM



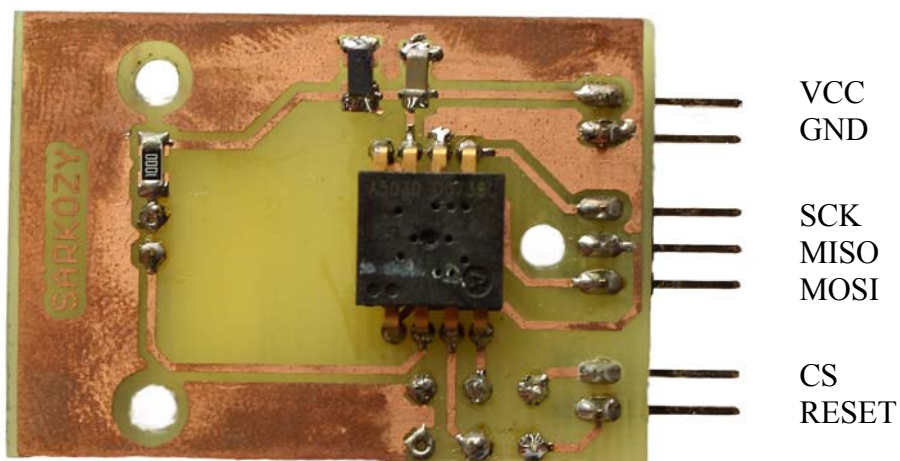
Příloha 9: Osazovací plán plošného spoje programátoru USBTiny MkII TOP



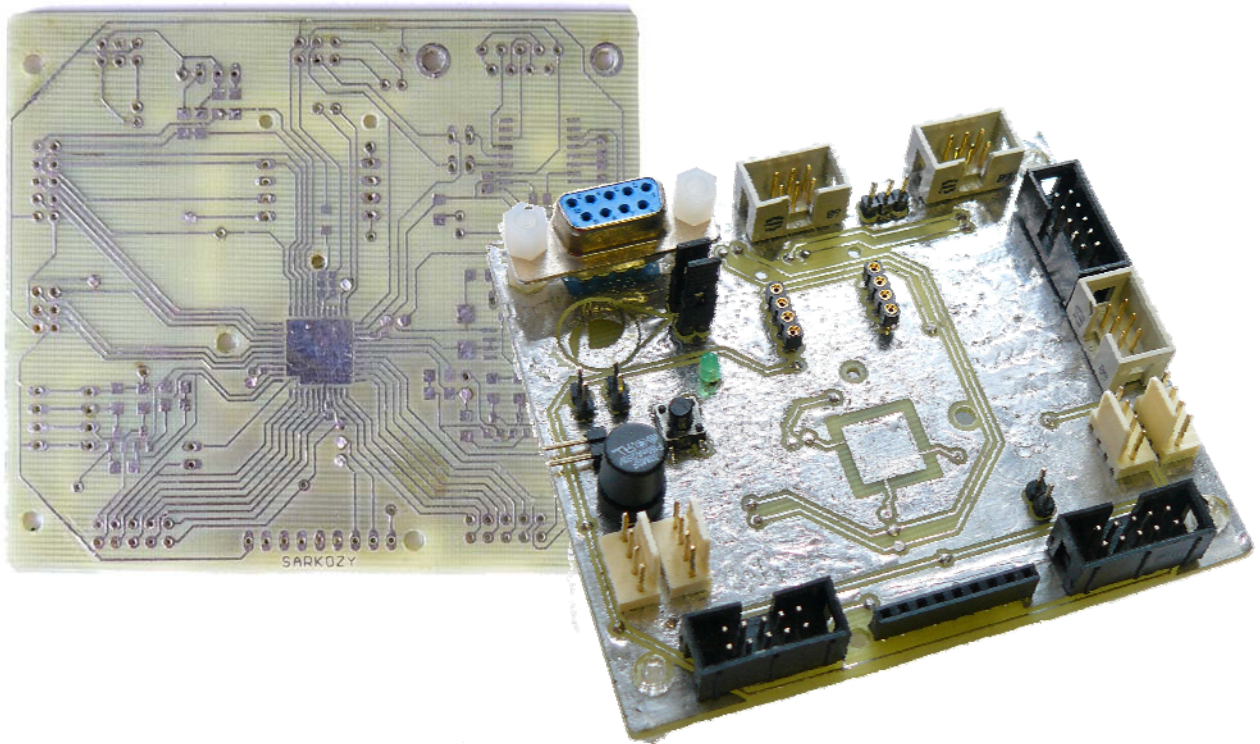
Příloha 10: Schéma zapojení modulu odometrie



Příloha 11: Vpravo -plošný spoj modulu odometrie – BOTTOM, vlevo - Osazovací plán plošného spoje modulu odometrie



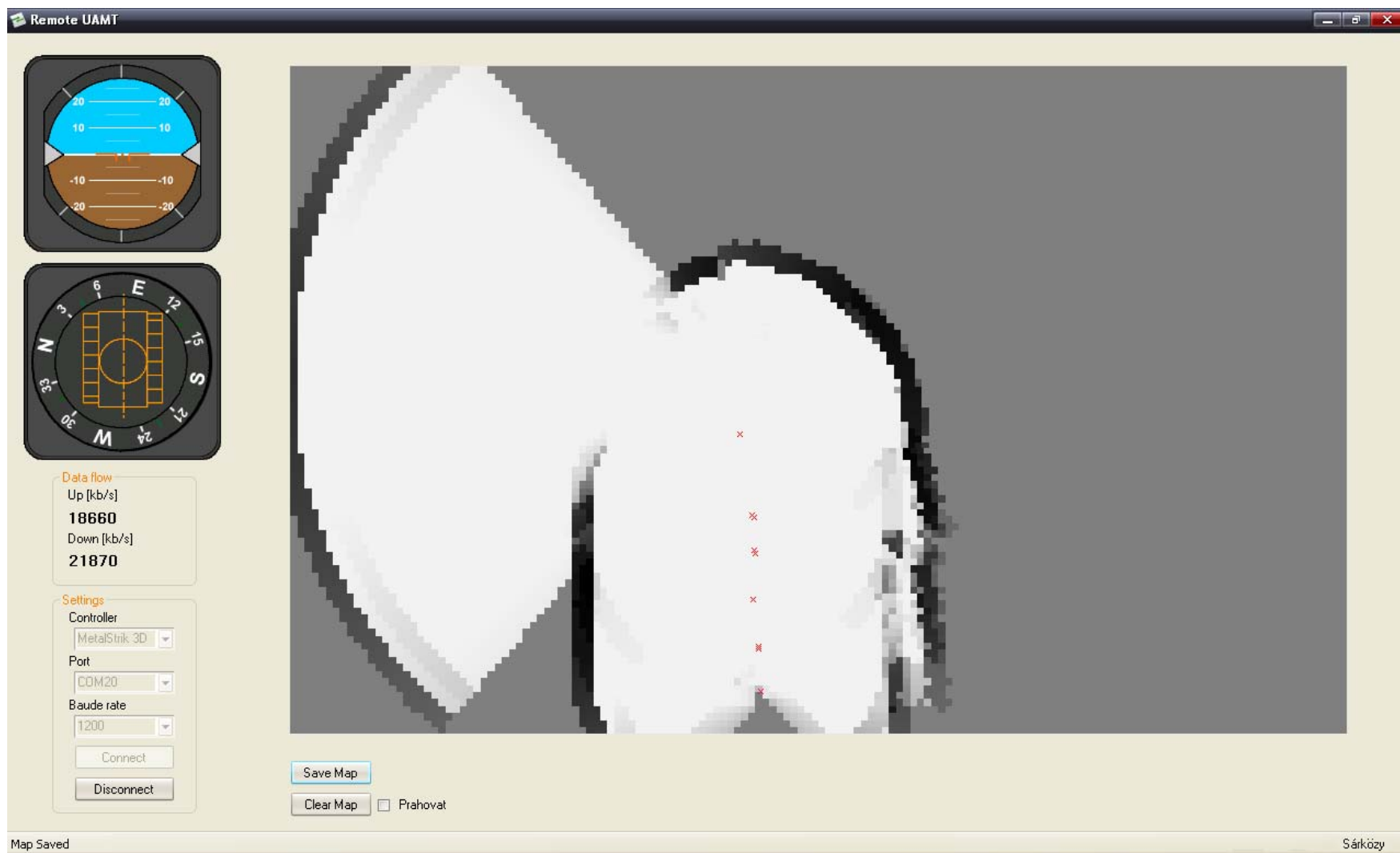
Příloha 12: Osazený plošný spoj modulu odometrie



Příloha 13: Realizace DPS řídicí jednotky pro malého mobilního robota



Příloha 14: Realizace programátoru USBTiny MkII



Příloha 15: Uživatelské rozhraní programu REMOTE



Příloha 16: Malý mobilní robot osazený řídicí jednotkou a snímacím subsystémem