

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

BEZDRÁTOVĚ ŘÍZENÝ MODEL VOZIDLA

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

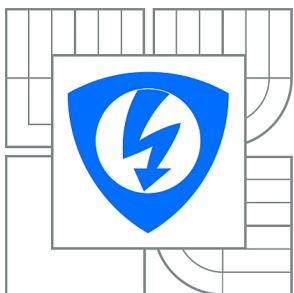
VILÉM ZÁVODNÝ

BRNO 2012



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ**

**ÚSTAV TELEKOMUNIKACÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

## **BEZDRÁTOVĚ ŘÍZENÝ MODEL VOZIDLA**

WIRELESS-CONTROLLED MODEL CAR

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**VILÉM ZÁVODNÝ**

**VEDOUCÍ PRÁCE**

SUPERVISOR

doc. Ing. PAVEL ŠTEFFAN, Ph.D.

BRNO 2012



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav telekomunikací

# Bakalářská práce

bakalářský studijní obor  
Teleinformatika

**Student:** Vilém Závodný

**ID:** 125710

**Ročník:** 3

**Akademický rok:** 2011/2012

**NÁZEV TÉMATU:**

## Bezdrátově řízený model vozidla

### POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je návrh a realizace bezdrátově řízeného modelu vozidla ovládaného pomocí počítače nebo smartphonu. Řídící elektronika modelu musí podporovat standardní funkce jízdy, provádět měření okolních podmínek a zpracovávat měřené údaje v nadřazeném mikrokontroléru. Zpracované hodnoty budou odesílány bezdrátově do počítače. Na modelu realizujte moderní elektronické prvky jako například ABS, ASR apod.

### DOPORUČENÁ LITERATURA:

- [1] Mann Burkhard C pro mikrokontroléry, Praha:2003, BEN, ISBN 80-7300-077-6
- [2] Hrbáček Jiří Komunikace mikrokontroléru s okolím 2., Praha:2000, ISBN 80-86056-73-2
- [3] Axelson Jan, USB complete: everything you need to develop custom USB peripherals, Independent Publishers Group ISBN 978-1-931448-02-4

**Termín zadání:** 6.2.2012

**Termín odevzdání:** 31.5.2012

**Vedoucí práce:** doc. Ing. Pavel Šteffan, Ph.D.

**Konzultanti bakalářské práce:**

**prof. Ing. Kamil Vrba, CSc.**

*Předseda oborové rady*

### UPOZORNĚNÍ:

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

## ABSTRAKT

Práce se zabývá analýzou a praktickou realizací moderních prvků v automobilech, které mají na starosti větší komfort cestujících a vyšší bezpečnost jak cestujících, tak i ostatních účastníků silničního provozu. Celá práce je řešena na dálkově ovládaném modelu vozidla, proto je potřeba se zabývat nejen moderními prvky v automobilech, ale i bezdrátovým přenosem dat, ovládáním motorů, spolehlivostí celého systému a celou řadou dílčích problémů. Všechny tyto problémy, které v průběhu práce nastaly a jejich řešení jsou popsány v následujícím dokumentu.

Práce vznikla na základě seznámení s novými prvky, moduly a součástkami, které je možné nasadit jako senzory, vyhodnocovací prvky a další elektrotechnické a softwarové součásti pro automobilový průmysl v propojení s elektrotechnikou a telekomunikační technikou. Dále je práce zaměřena na efektivní bezdrátové řízení vozidla, měření okolních podmínek a zpracování dat tak, aby bylo možné je odeslat v co nejkratším čase do řídicího počítače nebo na ně zareagovat přímo v mikrokontroléru v případě kritických hodnot.

## KLÍČOVÁ SLOVA

Moderní prvky v automobilech, bezdrátový přenos dat, ZigBee, bluetooth, teplotní senzory, akcelerometr, řízení servomotoru, senzory překážek, ASR.

## ABSTRACT

The work deals with the analysis and practical implementation of advanced features in cars, which are responsible for greater passenger comfort and greater security for both passengers and other road users. All the work is solved on a remote-controlled model car, hence the need to deal not only with modern elements in cars, but also wireless data transmission, motor control, reliability of the system and a number of component problems. All these problems that occurred during work and their solutions are described in the following document.

The work was based on the introduction of new items, modules and components that can be deployed as sensors, evaluation items, and other electrical and software components for the automotive industry in connection with electrical engineering and telecommunications technology. Further work is focused on efficient wireless control of the vehicle, measuring environmental conditions and processing data so that they can be sent as soon as possible to the host PC or respond to them directly in the microcontroller in case of critical values.

## KEYWORDS

Advanced features in cars, wireless data transmission, ZigBee, bluetooth, temperature sensors, accelerometer, control servomotor, sensors of obstruction, ASR.

ZÁVODNÝ, Vilém *Bezdrátově řízený model vozidla*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2012. 65 s. Vedoucí práce byl doc. Ing. Pavel Šteffan, Ph.D.

## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Bezdrátově řízený model vozidla“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

(podpis autora)



Děkuji vedoucímu bakalářské práce Doc. Ing. Pavlovi Šteffanovi, Ph. D. za metodickou pomoc, připomínky, cenné rady při zpracování práce, zapůjčení modelu vozidla a zakoupení potřebných modulů a panu Ing. Ladislavu Macháňovi za příkladnou pomoc v laboratoři.

Dále děkuji firmě SOS electronic za pomoc při výběru součástek a finanční podporu.

# OBSAH

Úvod	12
<b>1 Teoretický úvod</b>	<b>14</b>
1.1 Prvky v moderních automobilech	14
1.1.1 Systém regulace prokluzu kol ASR	14
1.1.2 Detekce překážek	15
1.1.3 Udržování bezpečné vzdálenosti za vozidlem	16
1.1.4 Udržování vozidla v pruhu	16
1.2 Mikrokontroléry Atmel	16
1.2.1 USART	16
1.2.2 I <sup>2</sup> C (TWI)	18
1.2.3 SPI	20
1.2.4 A/D převodník	21
1.3 Bezdrátový přenos dat	22
1.3.1 Bluetooth	22
1.3.2 ZigBee	23
1.4 Senzory	24
1.4.1 Akcelerometr	25
1.4.2 Senzory teploty	27
1.5 Řízení servomotoru	28
<b>2 Praktická realizace</b>	<b>30</b>
2.1 Dvě různá testovaná řešení	30
2.2 Popis jednotlivých bloků schématu	33
2.2.1 Joystick	33
2.2.2 PC	34
2.2.3 Převodník z USB na USART	36
2.2.4 Převodník z USART na SPI	36
2.2.5 MRF24J40MB a BTM-112	37
2.2.6 Mikrokontrolér	38
2.2.7 Snímání polohy akcelerometrem	39
2.2.8 Měření teploty	39
2.2.9 Servomotor řízení	40
2.2.10 Měření vzdálenosti objektů	40
2.2.11 Komparátor	41
2.2.12 Měření a porovnávání otáček kol	41
2.2.13 Speed controller a motor	42

2.2.14	Senzor bílé čáry . . . . .	43
2.3	Celková komunikace . . . . .	44
2.4	Rozmístění a konstrukce jednotlivých prvků ve vozidle . . . . .	46
<b>3</b>	<b>Závěr</b>	<b>50</b>
	<b>Literatura</b>	<b>52</b>
	<b>Seznam příloh</b>	<b>54</b>
<b>A</b>	<b>Přílohy</b>	<b>55</b>
A.1	Obsah přiloženého CD . . . . .	55
A.2	Návod na oživení celé práce . . . . .	55
A.3	USART . . . . .	56
A.4	I <sup>2</sup> C . . . . .	57
A.5	SPI . . . . .	58
A.6	Bluetooth . . . . .	59
A.7	Akcelerometr . . . . .	60
A.8	Schémata . . . . .	61
A.9	Desky plošných spojů . . . . .	63

# SEZNAM OBRÁZKŮ

1	Použitý model vozidla . . . . .	12
1.1	Zobrazení vzdálenosti od překážky parkovacím asistentem . . . . .	15
1.2	Datový rámec USART – asynchronní přenos dat [8] . . . . .	17
1.3	Blokové schéma zapojení několika zařízení na I <sup>2</sup> C [9] . . . . .	18
1.4	Přenos jednoho 8bitového rámce po I <sup>2</sup> C [9] . . . . .	19
1.5	Přenos jednoho 8bitového rámce oběma směry po sběrnici SPI [9] . . . . .	20
1.6	Spojení dvou obvodů pomocí sběrnice SPI [9] . . . . .	21
1.7	Bluetooth modul BTM-112 . . . . .	23
1.8	Připojení modulu k počítači přes USB . . . . .	24
1.9	Posílání AT příkazů do modulu programem „Hercules SETUP utility“ . . . . .	25
1.10	ZigBee modul MRF24J40MB od firmy Microchip . . . . .	26
1.11	Akcelerometr LIS3LV02DQ zapájený na testovací desce . . . . .	26
1.12	Zápis dat do registru akcelerometru přes I <sup>2</sup> C [15] . . . . .	27
1.13	Čtení dat z registru akcelerometru přes I <sup>2</sup> C [15] . . . . .	27
1.14	Teplotní senzor TC74 [11] . . . . .	27
1.15	Impulzy pro řízení servomotoru [17] . . . . .	29
2.1	Fotografie hotové desky sestavené na základě blokového schématu na obrázku 2.7 . . . . .	30
2.2	Fotografie vývojové desky sestavené na základě blokového schématu na obrázku 2.7 . . . . .	31
2.3	Fotografie vývojové desky sestavené na základě blokového schématu na obrázku 2.8 . . . . .	31
2.4	Fotografie hotové desky sestavené na základě blokového schématu na obrázku 2.6 . . . . .	32
2.5	Testované blokové schéma části připojené k PC v případě použití ZigBee . . . . .	32
2.6	Realizované blokové schéma části připojené k PC v případě použití modulu BTM-112 . . . . .	33
2.7	Realizované blokové schéma hardwarové části ve vozidle s použitím BTM-112 . . . . .	34
2.8	Testované blokové schéma části ve vozidle s použitím ZigBee . . . . .	35
2.9	Joystick použitý v mé práci . . . . .	36
2.10	Aplikace naprogramovaná pro testování a ovládání modelu vozidla . . . . .	37
2.11	Schéma převodníku z USB na USART [4] . . . . .	38
2.12	Fotografie IR detektoru vzdálenosti GP2Y0A41 . . . . .	40
2.13	Schéma analogového komparátoru . . . . .	41
2.14	Fotografie použitého magnetického spínače . . . . .	42
2.15	Fotografie použitého speed controlleru ESC300 . . . . .	42

2.16	Senzor QRD1114 pro detekci bílé čáry . . . . .	43
2.17	Schéma zapojení jednoho senzoru . . . . .	44
2.18	Fotografie vytvořeného senzoru na detekci bílé čáry . . . . .	44
2.19	Datový rámec, který přenáší informace mezi PC a vozidlem . . . . .	45
2.20	Umístění a přilepení magnetického spínače k přednímu kolu . . . . .	47
2.21	Umístění a konstrukční uchycení IR detektoru vzdálenosti a IR de- tektorů čáry vzadu . . . . .	47
2.22	Umístění a konstrukční uchycení IR detektoru vzdálenosti vepředu . .	48
2.23	Umístění a uchycení teplotního čidla u akumulátoru . . . . .	48
2.24	Umístění a uchycení teplotního čidla na bluetooth modulu . . . . .	48
2.25	Grafické znázornění hotového vozidla s rozmístěním všech přidaných prvků . . . . .	49
A.1	Schéma zapojení hotového zařízení ve vozidle . . . . .	61
A.2	Schéma zapojení hotového zařízení připojeného k počítači . . . . .	62
A.3	Schéma zapojení hotového komparátoru pro IR detektory překážek a pro zapojení senzorů bílé čáry . . . . .	62
A.4	Deska plošného spoje hotového zařízení ve vozidle - vrstva TOP . . .	63
A.5	Osazovací plán desky plošného spoje hotového zařízení ve vozidle - vrstva TOP . . . . .	63
A.6	Deska plošného spoje hotového zařízení připojeného k počítači - vrstva TOP . . . . .	64
A.7	Deska plošného spoje hotového zařízení připojeného k počítači - vrstva BOTTOM . . . . .	64
A.8	Osazovací plán desky plošného spoje hotového zařízení připojeného k počítači - vrstva TOP . . . . .	64
A.9	Deska plošného spoje komparátoru - vrstva BOTTOM . . . . .	65
A.10	Osazovací plán desky komparátoru - vrstva TOP . . . . .	65

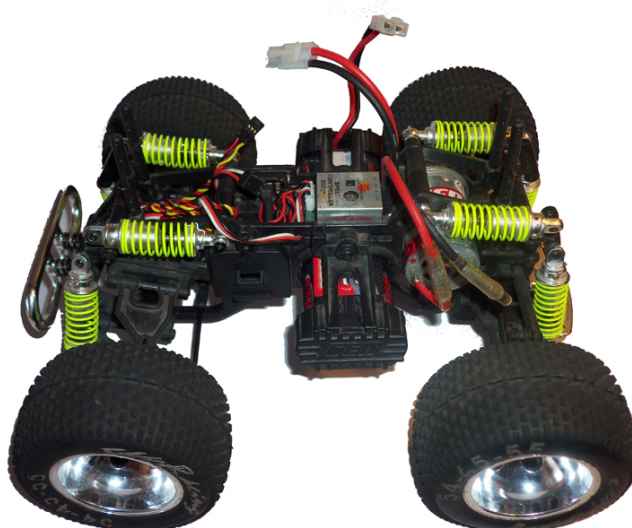
## SEZNAM TABULEK

1.1	Nastavení předděličky u I <sup>2</sup> C . . . . .	20
1.2	Označení teplotních senzorů TC74 a jejich adresy. [11] . . . . .	28
2.1	Přehled dostupných příkazů pro model vozidla . . . . .	46
A.1	Popis jednotlivých bitů registrů <i>UCSRA</i> , <i>UCSRB</i> a <i>UCSRC</i> . [8] [9] [1] . . . . .	56
A.2	Nastavení parity. [8] [9] [1] . . . . .	57
A.3	Nastavení počtu bitů rámce. [8] [9] [1] . . . . .	57
A.4	Popis jednotlivých bitů registrů <i>UCSRA</i> , <i>UCSRB</i> a <i>UCSRC</i> . [8] [9] [1] . . . . .	57
A.5	Příklady nastavení registrů <i>UBRRL</i> a <i>UBRRH</i> při frekvenci krystalu 16 MHz. [8] [9] [1] . . . . .	58
A.6	Nastavení přenosové frekvence u SPI. [8] [9] [1] . . . . .	58
A.7	AT příkazy pro nastavení modulu BTM-112. [12] . . . . .	59
A.8	Popis vybraných registrů akcelerometru. [15] . . . . .	60

# ÚVOD

V dnešní době se setkáváme s moderními automobily, vybavenými speciálními prvky pro větší komfort a bezpečnost, které chrání nejen osoby uvnitř vozidla, ale také chodce, zvířata a ostatní účastníky silničního provozu. Těmito prvky mohou být např. parkovací senzory, brzdňý systém ABS, vyrovnávání v zatáčkách ESP, systém regulace prokluzu kol ASR, sledování jízdy ve správném pruhu, dodržování bezpečné vzdálenosti od ostatních vozidel, sledování chodců na přechodu pro chodce a mnoho dalších.

Tyto moderní prvky jsou tvořeny speciálními senzory a složitou elektronikou, řízenou mikroprocesory. A proto se téma této práce týká realizace těchto prvků. Bylo vyzkoušeno několik možných řešení a různých senzorů, ale k praktické realizaci byly vybrány pouze ty nejvhodnější.



Obr. 1: Použitý model vozidla

Pro demonstraci použití moderních prvků v automobilech je nutné vytvořit model vozidla (obr. 1), který bude řízený bezdrátově počítačem. Vyhodnocování senzorů pak může probíhat v počítači nebo v mikrokontroléru, který bude řídit běh modelu vozidla. Při komunikaci modelu s počítačem musí být kladen důraz na rychlost a spolehlivost přenosu, aby bylo možné vyhodnocovat náhlé změny a rychle na ně reagovat. Proto je nutné zvolit vhodné moduly pro bezdrátový přenos, dostatečnou přenosovou rychlost a řídit tok dat podle zatížení. Dále je potřeba otestovat a zvolit vhodné senzory a čidla okolního prostředí, aby bylo možné včas a správně zareagovat na vzniklou situaci. V tomto případě je nutné brát ohled nejen na rychlost vozidla, ale také na prostředí, ve kterém se vozidlo vyskytuje. Takové prostředí

může být charakterizováno vlhkostí, teplotou, tlakem, světlem a dalšími fyzikálními vlastnostmi. Všechny tyto faktory ovlivňují správný běh nejen elektrotechnických součástí, ale především ideální funkčnost senzorů.

Celá práce vede k pochopení principu mnohých senzorů, bezdrátového přenosu dat a především již zmíněných moderních prvků v automobilech tak, aby je bylo možné prakticky aplikovat v modelu vozidla, otestovat je a popř. i vylepšit.

# 1 TEORETICKÝ ÚVOD

## 1.1 Prvky v moderních automobilech

Jak jsem se již zmínil v úvodu, tak prvky v moderních automobilech zvyšují komfort jízdy a především bezpečnost účastníků provozu. Proto je nutné, aby tyto prvky byly spolehlivé a dokázali situaci rychle vyhodnotit. Spousta těchto senzorů existovalo ještě dříve, než se začaly používat v automobilech, ale jejich cena byla příliš vysoká a nebyly dostatečně spolehlivé. Dnes díky sériové výrobě součástí je možné téměř všechno pořídit za relativně nízkou cenu a především koupit po kusech. Díky tomu se i jednotlivci mohou zabývat testováním nových technologií a tím mají velký význam pro vývoj nových součástí a senzorů.

V následujících kategoriích mé práce se budu zabývat některými moderními prvky v automobilech a jejich praktickou realizací v modelu vozidla. Bohužel ne všechny tyto prvky je možné prakticky realizovat na modelu vozidla, a proto se bude jednat pouze o realizaci některých ze zmíněných moderních prvků.

### 1.1.1 Systém regulace prokluzu kol ASR

Jedním z často používaných moderních prvků v automobilech je systém ASR (z německého Antriebschlupfregelung). Jedná se o systém proti prokluzu hnacích kol vozidla, který spolupracuje s několika dalšími systémy ve vozidle, mezi ně se řadí protiblokovací systém ABS (z anglického Anti-lock Braking System), elektronicky řízený diferenciál EDS (z německého Elektronische Differenzialsperre) a řídicí jednotka motoru. Všechny tyto prvky mají vliv na protáčení kol automobilu.

Prakticky je celá regulace vyřešena následovně. U každého kola automobilu je senzor otáček, který slouží také pro systém ABS. Sensory otáček posílají informace o otáčkách do řídicí jednotky motoru, která vyhodnocuje rozdíly otáčení kol na hnané nápravě a na nápravě nepoháněné. V případě, že kola na hnané nápravě se točí rychleji, pak řídicí jednotka zareaguje a sníží točivý moment motoru nuceným ubráním plynu natolik, aby se otáčky kol vyrovnaly. V případě, že k prokluzu kol dojde v nižší rychlosti např. při rozjezdu vozidla, pak je ještě prokluzující kolo mírně přibrzděno.

Celý protiprokluzový systém působí proto, aby bylo možné plynule se rozjet na kluzkém povrchu a vyrovnává otáčení kol v zatáčkách. Vše je indikováno řídicí kontrolkou ASR, aby i řidič mohl zareagovat na povrch vozovky.

### 1.1.2 Detekce překážek

Jako dalším systémem, který pomáhá řidiči ovládat své auto je detekce překážek kolem vozidla. Zpravidla se využívá tato vymoženost při parkování, kdy jsou vzadu na vozidle, někdy i vepředu, umístěny parkovací senzory, které řidiče upozorňují na překážku. Je několik variant systému. Jednoduché systémy upozorňují na vzdálenost od překážky akustickým signálem. Při zmenšování se vzdálenosti od překážky se snižuje interval akustických tónů, který postupně přechází až v souvislý tón pro minimální vzdálenost. U novějších a vyspělejších automobilů je ještě přidán malý displej, indikující vzdálenost zprava i zleva barevnými led diodami a popř. ještě ukazuje vzdálenost od překážky v metrech. Jeden z takových systémů je zobrazen na obrázku 1.1. Takový asistent se dá do auta bez problémů i přidat, pak je možné displej se zobrazením přilepit na palubní desku nebo v případě již zabudovaného parkovacího asistenta bývá zobrazení vzdálenosti ve zpětném zrcátku.



Obr. 1.1: Zobrazení vzdálenosti od překážky parkovacím asistentem

Snímání vzdálenosti funguje na principu odražení signálu od překážky. V praxi se využívají dvojí senzory pro detekci překážek a těmi jsou ultrazvukové a infračervené. Při použití těchto senzorů se pokaždé signál odráží jinak. V případě ultrazvukových senzorů se jedná o akustickou vlnu, kterou vysílací ultrazvukové čidlo posílá s kmitočtem 40 kHz a počítá dobu mezi přijatou a vyslanou vlnou. U infračervených senzorů je vyslaná vlna světelná v infračerveném spektru. Výhodou je, že ultrazvukové vlny se oproti infračerveným odrážejí od téměř všech předmětů i barev, ale nevýhodou je rychlost vyhodnocení, protože akustická vlna se ve vzduchu šíří rychlostí 340 m/s a oproti tomu světelná vlna v infračerveném spektru se šíří rychlostí  $3 \cdot 10^8$  m/s. Takže je vždy nutné zvážit jestli je nutná rychlá reakce nebo přesnost. Samozřejmě v případě couvání s automobilem se nejedná o vysoké rychlosti jízdy a proto je vhodnější ultrazvukový senzor.

Kromě parkování se může stát, že do vozovky vstoupí neočekávaná překážka v podobě nějakého zvířete nebo člověka. I na tohle dokáží moderní automobily zareagovat nebo upozornit řidiče. Především se to hodí v nepříznivých podmínkách či snížené viditelnosti. Pak ideálním řešením je použití kamery s termovizí, která snímá okolí před vozidlem na dostatečně velkou vzdálenost a zobrazuje vše na LCD displeji ve vozidle na palubní desce. V případě, že se objeví jakýkoliv živý tvor před vozidlem,

je pak zobrazen viditelně na displeji díky termovizi, která reaguje na vyzářené teplo živým organismem.

### 1.1.3 Udržování bezpečné vzdálenosti za vozidlem

Některé automobilky do svých moderních automobilů integrují i systém udržování bezpečné vzdálenosti za vozidlem tzv. adaptivní tempomat. Princip činnosti je skoro stejný s parkovacím asistentem, jen je rozdíl ve vyhodnocování, zpracování a následné reakci. Také se pro rychlejší vyhodnocení používá jiný senzor překážky a tím je mikrovlnný nebo laserový radar. Tento radar zjišťuje rychlost objektu jedoucího před vozidlem a posílá patřičné informace řídicí jednotce, která následně na to upravuje rychlost automobilu tak, aby byla dodržena nastavená vzdálenost. Jestliže vozidlo jedoucí vepředu zastaví, pak tuhle skutečnost řídicí jednotka adaptivního tempomatu vyhodnotí a zastaví automobil.

### 1.1.4 Udržování vozidla v pruhu

Posledním nadstandardním systémem v moderních automobilech, který budu zmiňovat, je udržování vozidla v pruhu nebo se to dá taky pojmout jako sledování bílé čáry. U automobilů s moderní výbavou je řešen celý systém CMOS kamerou umístěnou pod čelním sklem, která snímá vozovku a levou i pravou bílou čáru. Jestliže se pak vozidlo blíží více na některou stranu, mírně pomůže řidiči se vrátit zpět do pruhu.

Celý systém může spolupracovat s adaptivním tempomatem a může tak řidiči velice zpříjemnit a zjednodušit cestování po dálnici. Avšak v případě, že řidič pustí volant, je vyzván zvukovou signalizací a systém udržování vozidla v pruhu se okamžitě vypne. Jakmile řidič zase převezme volant systém se opět zapne.

## 1.2 Mikrokontroléry Atmel

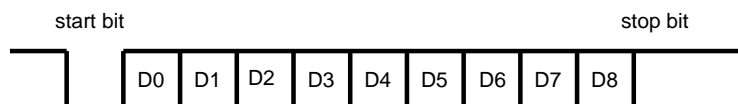
Pro řízení a zpracování dat v realizovaném modelu vozidla jsem použil 8bitový mikrokontrolér Atmel a to konkrétně ATmega32 běžící na frekvenci 16 MHz. Proto se nyní budu zabývat některými jeho možnostmi a hardwarovými prvky, které používám v mé praktické realizaci. Mnohé z těchto informací jsou také popsány v literatuře [1], [8] a [9].

### 1.2.1 USART

USART (Universal Synchronous / Asynchronous Receiver and Transmitter) je synchronní a asynchronní sériové rozhraní pro přenos dat mezi zařízeními v obou smě-

rech (plný duplex). Někdy je také nazývaný pouze jako UART a to v případě, že není možné použít synchronní linku. Používá se pro komunikaci mezi mikrokontroléry, počítači a dalšími zařízeními podporující tento standard.

Pro přenos se používají rámce, které mohou mít 5 až 9 bitů a jsou od sebe odděleny jedním start bitem a jedním nebo dvěma stop bity (obr. 1.2). Každý rámec může obsahovat ještě paritní bit pro kontrolu rámce. Dále je možné nastavit rychlost přenosu dat od 1 200 bps až do 250 kbps.



Obr. 1.2: Datový rámec USART – asynchronní přenos dat [8]

### Některé vlastnosti USART u mikrokontroléru ATmega32

- Plný duplex (může současně přijímat i vysílat).
- Synchronní nebo asynchronní režim.
- Možnost nastavení přenosové rychlosti v rozsahu 1 200 bps až 250 kbps.
- Podpora rámců délky 5 až 9 datových bitů a 1 až 2 stop bity.
- Generátor sudé/liché parity pro vysílač a hardwarové testování parity pro přijímač.
- Detekce ztráty znaku (OverRun).
- Detekce chyby rámce (Frame Error).
- Obsahuje digitální dolnofrekvenční propust pro detekci falešného start bitu a filtraci zákmitů datových bitů.
- Tři nezávislá přerušení (odvysílání znaku, vyprázdnění vysílacího registru a příjem znaku).
- Možnost zdvojnásobit přenosovou rychlost v asynchronním režimu. [1][8][9]

### Nastavení a popis registrů USART u mikrokontroléru ATmega32

Pro používání USART se pro správnou práci používají osmibitové registry. Některé slouží pouze k nastavení, některé k odeslání znaku a některé pro příjem. Tyhle registry jsou popsány zde:

- Aby USART správně fungoval je třeba jej inicializovat a to se řeší pomocí registrů *UCSRA*, *UCSRB* a *UCSRC*, kde každý z bitů těchto registrů má svoji funkci. Jednotlivé bity jsou popsány v tabulce A.1 v příloze.

- Pro příjem a posílání znaku se používá registr  $UDR$ .
- Registry  $UBRRL$  a  $UBRRH$  slouží pro nastavení přenosové rychlosti (hodnotu lze vypočítat podle vzorců (1.1) a (1.2) nebo najít v tabulce A.5 v příloze).

Vzorec na výpočet hodnoty registru  $UBRR$  pro klasický režim

$$BAUD = \frac{f_{osc}}{16(UBRR + 1)} \Rightarrow UBRR = \frac{f_{osc}}{16BAUD} - 1 \quad (1.1)$$

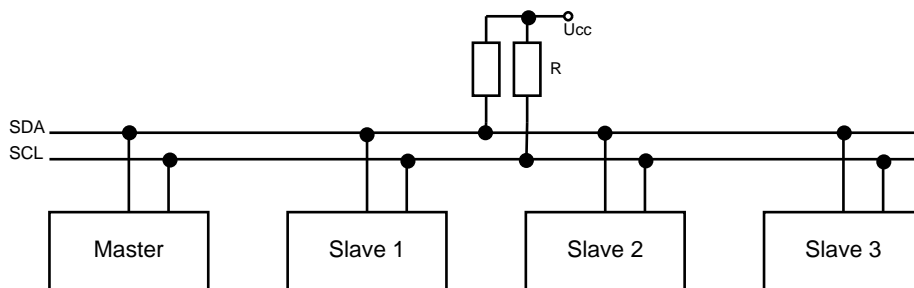
a pro režim dvojnásobné rychlosti

$$BAUD = \frac{f_{osc}}{8(UBRR + 1)} \Rightarrow UBRR = \frac{f_{osc}}{8BAUD} - 1, \quad (1.2)$$

kde  $BAUD$  je přenosová rychlost a  $f_{osc}$  je frekvence krystalu. Vzorec pro synchronní režim zde neuvádím z toho důvodu, že synchronní režim USART ve své práci nevyužívám.

### 1.2.2 I<sup>2</sup>C (TWI)

Jako další uvádím sběrnici I<sup>2</sup>C (Inter Integrated Circuit), někdy nazývanou také jako TWI (Two Wire Interface), která je běžně používaná pro připojení až 128mi zařízení po dvou vodičích (obr. 1.3). Sběrnice je obousměrná, ale v jeden okamžik může vysílat pouze jedno zařízení (poloviční duplex). Oba vodiče musí být připojeny přes rezistor na napájecí napětí (tzv. pull-up rezistory), tím se předejde poškození vysílacích a přijímacích obvodů při kolizi a poškodí se pouze úrovně signálů.

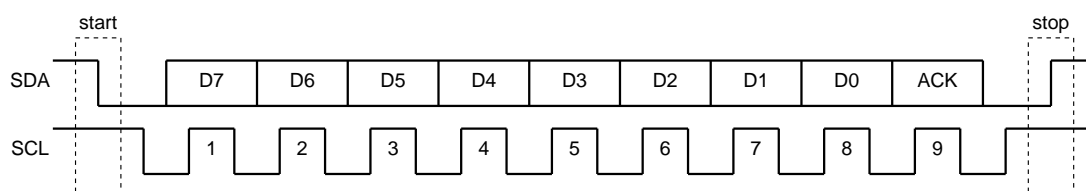


Obr. 1.3: Blokové schéma zapojení několika zařízení na I<sup>2</sup>C [9]

### Přenos dat

Na linku je připojen jeden master (řídící) a několik slave (podřízených) obvodů. Pro posílání datového rámce některému slave zařízení, master vyšle v prvních 7mi

bitech adresu slave zařízení, v osmém bitu log. 1 a čeká na příjem potvrzovacího bitu ACK. Poté master vyšle data po lince SDA. V případě, že je potřeba získat data ze slave zařízení, pak master opět pošle adresu, ale v osmém bitu pošle log. 0, aby slave poznal, že má volno a může poslat data. Synchronizace přenosu dat probíhá impulsy na lince SCL o frekvenci 10 kHz až 400 kHz. Průběh dat na linkách SDA a SCL je zřejmý z obrázku 1.4.



Obr. 1.4: Přenos jednoho 8bitového rámce po I<sup>2</sup>C [9]

### Některé vlastnosti I<sup>2</sup>C u mikrokontroléru ATmega32

- Jednoduché, výkonné a přizpůsobitelné rozhraní.
- Podpora práce v režimu master nebo slave.
- Mikrokontrolér může pracovat jako vysílač i přijímač.
- 7bitový adresovací prostor dovolující rozlišit až 128 adres podřízených obvodů.
- Přenosová rychlost až 400 kHz.
- Možnost probuzení mikrokontroléru z režimu spánku v případě adresace vnějším TWI obvodem.[1][9]

### Nastavení a popis registrů I<sup>2</sup>C u mikrokontroléru ATmega32

- *TWBR*: Registr pro nastavení přenosové rychlosti podle vzorce (1.3).
- *TWCR*: Nastavuje a inicializuje přenos (tabulka A.4 v příloze).
- *TWSR*: Uchovává informace o stavu přenosu a nastavuje předděličku rychlosti (tabulka 1.1).
- *TWDR*: Datový registr.
- *TWAR*: Adresa slave zařízení.

Vzorec pro výpočet hodnoty registru *TWBR*

$$f_{CLK} = \frac{f_{OSC}}{16 + 2(TWBR)4^{TWPS}}, \quad (1.3)$$

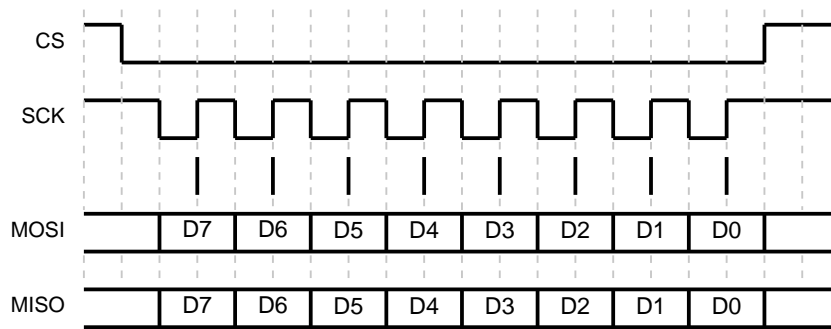
kde  $f_{CLK}$  je frekvence synchronizačních impulsů na lince CLK,  $f_{OSC}$  je frekvence krystalu a  $TWPS$  je hodnota nastavení předděličky.

Tab. 1.1: Nastavení předděličky u I<sup>2</sup>C .

$TWPS1$	$TWPS0$	hodnota předděličky
0	0	1
0	1	4
1	0	16
1	1	64

### 1.2.3 SPI

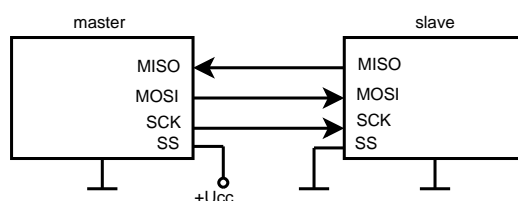
Poslední použitou sběrnicí v mé práci je SPI (Serial Peripheral Interface), která se často používá pro přenos většího objemu dat, protože dosahuje vysokých přenosových rychlostí i u mikrokontrolérů a modulů. Přenos dat u SPI je obousměrný (plný duplex) díky linkám MOSI (master out slave in) a MISO (master in slave out), po kterých se přenáší data synchronizované linkou CLK. Na lince CS (někdy uváděno SS) je pouze informace pro slave obvod, aby bylo rozlišeno, kterému obvodu jsou data určena. Datový rámec je zobrazen na obrázku 1.5, ze kterého je vidět jak jsou data závislá na synchronizačních pulsech. Na obrázku 1.6 je zobrazeno klasické zapojení mezi obvodem master a slave, které je použito v mé práci. Dále existují ještě dva způsoby propojení pro případ více slave obvodů řízených jedním masterem, který pak řídí celou komunikaci buď tím, že přiděluje jednotlivým slave zařízením vysílací čas, nebo si všechny obvody přeposílají data do kruhu.



Obr. 1.5: Přenos jednoho 8bitového rámce oběma směry po sběrnicí SPI [9]

## Základní vlastnosti SPI u mikrokontroléru ATmega32

- Plný duplex (schopnost současně přijímat i vysílat).
- Třívodičový synchronní přenos dat.
- Může pracovat jako master (řídící obvod) nebo slave (řízený obvod).
- 7 programovatelných rychlostí (tabulka A.6 v příloze).
- Přenosová rychlost až 8 Mb/s.
- Lze volit pořadí bitů (LSB až MSB nebo MSB až LSB).
- Příznaky konce a kolize přenosu.
- Schopnost procitnutí z režimu idle po příjmu dat v režimu slave. [1][8][9]



Obr. 1.6: Spojení dvou obvodů pomocí sběrnice SPI [9]

### 1.2.4 A/D převodník

Většina fyzikálních veličin v přírodě se vyskytuje jako analogová hodnota, tak i senzory těchto veličin obvykle dávají analogový výstup, nebo jsou tyto senzory značně levnější, než s digitálním výstupem. Nejčastěji se jedná o analogovou hodnotu napětí, která je nějakým způsobem závislá na měřené veličině. Nebo nás také může zajímat přímo měření napětí. Na všechny tyto případy se hodí použít analogově digitální převodník.

Mikrokontrolér Atmel má integrovaný 10bitový A/D převodník s postupnou aproximací. Ten funguje tak, že mikrokontrolér nastavuje postupně bity registru od nejvíce významného až po nejméně významný a porovnává aktuální hodnotu s měřenou.

Použitý typ mikrokontroléru disponuje 8mi kanály pro A/D převodník, tak je možné měřit až 8 nezávislých napětí. Vstupy jsou na portu A.

### Základní parametry A/D převodníku u mikrokontroléru ATmega32

- **10bitové rozlišení.**
- Integrovaná nelinearita 0,5 LSB.
- Doba převodu 65 až 260  $\mu$ s.

- **Rychlost převodu až 15 kSPS** (15 000 vzorků za sekundu).
- **Volitelná zabudovaná reference 2,56 V.**
- Režimy jednoduchý převod, automatické spouštění a volný běh.
- Přerušování po dokončení převodu.
- Možnost spouštět A/D převod pomocí dalších přerušování.
- Potlačovač šumu v režimu idle.[1][8][9]

## 1.3 Bezdrátový přenos dat

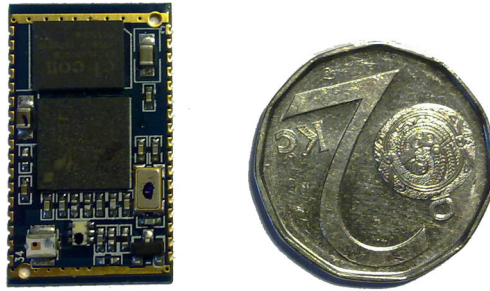
Jak již bylo v zadání i úvodu řečeno, úkolem je také bezdrátově řídit model vozidla. K tomu je vhodné využít některý z běžně prodávaných bezdrátových modulů pro přenos dat. Takových modulů je spousta a jejich výběr závisí na několika kriteriích, kterými může být frekvence přenosu dat, přenosová rychlost, protokol přenosu, maximální dosažitelná vzdálenost, podpora duplexního režimu nebo cena. Proto jsem otestoval několik modulů, u kterých nyní popíšu klady i zápory.

### 1.3.1 Bluetooth

První z testovaných modulů byl bluetooth BTM-112 (obr. 1.7), který pracuje na frekvenci 2,4 GHz. Modul podporuje obousměrný přenos dat (plný duplex) a jeho maximální přenosová rychlost dosahuje rychlosti až 460,8 kbps. Díky tomu je možné tento modul využívat pro rychlé aplikace, kde je potřeba posílat data oběma směry. Obrovskou výhodou je, že nám stačí pouze jeden tento modul při realizaci zařízení, které bude komunikovat s PC nebo mobilním telefonem, protože tyto zařízení již bluetooth mají integrované. Proto je i celá realizace levnější. Bohužel má tento modul i nevýhody. Jednou z nich je maximální dosažitelná vzdálenost přenosu. Díky dost vysoké přenosové rychlosti a přenosové frekvenci není možné dosáhnout převratných vzdáleností. Bez externí antény je tato vzdálenost omezena pouze na několik jednotek metrů. V případě připojení výkonnější externí antény by dosah mohl být až 100 m.

Další výhodou je spousta možných nastavení přenosu a zabezpečení, jako jsou např. rychlost přenosu, název zařízení, přístupový pin kód, parita nebo výběr zda je zařízení master nebo slave a to pomocí tzv. AT příkazů, které jsou popsány v příloze v tabulce A.7. Zadávání AT příkazů je možné pouze po kabelovém připojení přes UART. Pro tento případ, jsem sestrojil převodník mezi rozhraním USB a UART s integrovaným obvodem FT232RL od firmy FTDI, kterým jsem zmíněný modul nastavoval. Celé schéma kabelového připojení je na obrázku 1.8.

Pro zadávání AT příkazů podle tabulky A.7 z přílohy je ideální využít některý z volně dostupných programů pro komunikaci s porty. Jedním z takových programů



Obr. 1.7: Bluetooth modul BTM-112

je „Hercules SETUP utility“ dostupný z [6]. Na obrázku 1.9 je ukázáno jakým způsobem program používat. Je třeba správně zvolit přenosovou rychlost, počet bitů a číslo portu, na kterém je modul připojen. Poté už je možné posílat příkazy a za každým příkazem musí vždy být tzv. ukončovací znak <CR> tak jak je naznačeno v tabulce A.7 v příloze.

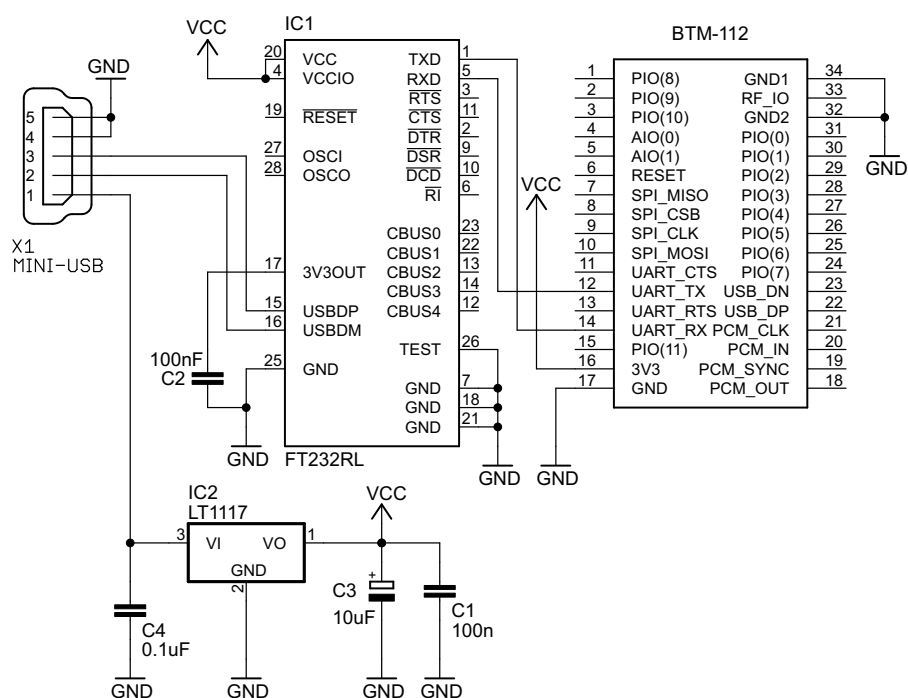
Důležitým poznatkem při mém testování bylo jak resetovat zařízení do továrního nastavení. To je možné udělat pomocí AT příkazů. Ale v případě, že je nastavena přenosová rychlost taková, jakou nejsem schopen realizovat (např. omezení programu nebo převodníku), pak je možné provést hardwarový reset přenosové rychlosti přivedením log. 1 na pin PIO(4) nebo-li pin č. 27 modulu BTM-112.

### Tovární nastavení modulu

- **Pin kód:** 1234
- **Baud rate:** 19 200 bps
- **Datových bitů:** 8
- **Přita:** žádná
- **Stop bity:** 1
- **Řízení toku:** HW nebo žádné [12]

### 1.3.2 ZigBee

Dalším testovaným bezdrátovým přenosem byl protokol ZigBee. Vybral jsem konkrétně moduly od firmy Microchip s označením MRF24J40MB (obr. 1.10). Kromě ZigBee je možné těmito moduly přenášet data i protokolem MiWi, který vyvinula firma Microchip. V případě těchto modulů je potřeba si nejprve říct, na co je chceme použít a jestli jsou pro naši aplikaci vhodné, protože disponují obrovským množstvím nastavení a mají obrovské možnosti. Využití najdeme především v aplikacích, kde je potřeba řídit tok dat, sledovat provoz a používat více klientů. Z toho tedy vyplývá,



Obr. 1.8: Připojení modulu k počítači přes USB

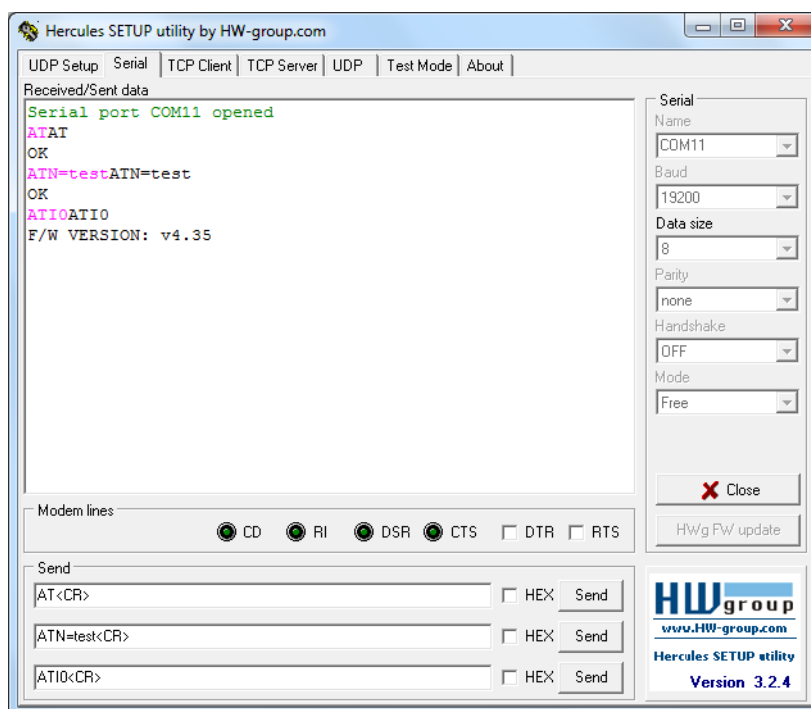
že pro použití P2P<sup>1</sup> připojení nejsou příliš vhodné s použitím protokolu ZigBee. Avšak i přesto byly moduly otestovány pro následující práci.

Pro práci s těmito moduly je potřeba komunikovat s integrovaným čipem, který disponuje několika desítkami registrů pro nastavení a čtení informací. Komunikace s čipem uvnitř modulu se děje pomocí sběrnice SPI, jejíž vlastnosti jsou popsány v kapitole 1.2.3. Pro případ čtení dat z registru je potřeba pouze poslat adresu daného registru. Poté čip pošle přes SPI rozhraní jako odpověď hodnotu uloženou v tomto registru. Pro zápis do registru je potřeba poslat adresu a hned poté data pro zápis.

## 1.4 Senzory

V mojí práci jsem využil několik senzorů např. pro měření teploty nebo snímání polohy vozidla. Nyní popíšu funkci těchto senzorů, praktickou realizaci a jejich uplatnění v praxi.

<sup>1</sup>přímé spojení dvou zařízení



Obr. 1.9: Posílání AT příkazů do modulu programem „Hercules SETUP utility“

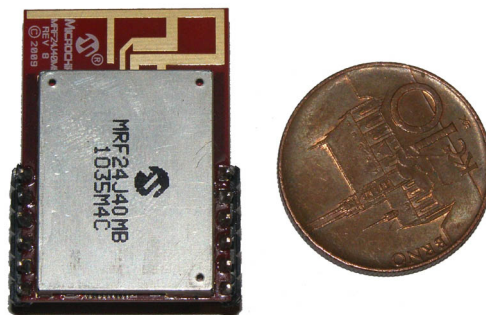
### 1.4.1 Akcelerometr

Jako první se zmíním o akcelerometru, který zjišťuje zrychlení vozidla v osách x, y a z. Pro tuto aplikaci jsem si vybral integrovaný obvod LIS3LV02DQ od firmy STMicroelectronics a to především z toho důvodu, že je možné s ním komunikovat digitálně pomocí rozhraní I<sup>2</sup>C nebo SPI. Další výhodou jsou jeho rozměry, které jsou 10 x 10 mm. Bohužel tohle dělalo v amatérských podmínkách trochu problémy a tak pro testování byla vytvořena deska s vývody, na kterou byl obvod zapájen (obr. 1.11). Ještě je výhodou volba rozlišení, které je možné volit mezi 2g nebo 6g. Informace jsou popsány také v [15].

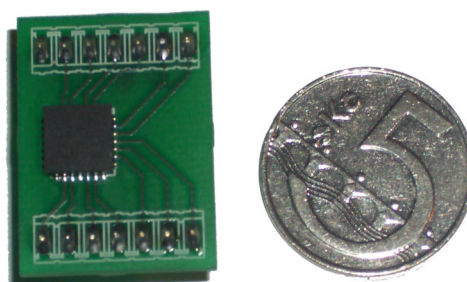
Použitý akcelerometr je tříosý lineární s digitálním výstupem vyrobený technologií MEMS (Micro Electro-Mechanical Systems). Tato technologie umožňuje vyrábět konstrukce mechanických a elektrických zařízení na jednom čipu o velice malých rozměrech. Akcelerometry vyrobeny touto technologií vynikají nízkou spotřebou energie, mechanickou odolností a velmi malými rozměry.

Akcelerometry se používají i běžně v praxi. Příklad využití je u automobilů, kdy senzor zjišťuje přetížení při dopravní nehodě a reaguje na to vystřelením airbagů. Dále je možné použití v herním průmyslu nebo u virtuální reality.

Komunikace s akcelerometrem je velice snadná a to díky registrům, do kterých zapisujeme informaci nebo ji z nich čteme. Popis vybraných registrů je v příloze



Obr. 1.10: ZigBee modul MRF24J40MB od firmy Microchip



Obr. 1.11: Akcelerometr LIS3LV02DQ zapájený na testovací desce

v tabulce A.8. V případě použití sběrnice I<sup>2</sup>C na komunikaci s tímto obvodem se používají rámce na obrázcích 1.12 a 1.13, kde první obrázek označuje zápis do registru a druhý je čtení z registru. Jedná se o klasickou I<sup>2</sup>C komunikaci, kdy mikrokontrolér je master a akcelerometr slave.

### **Zápis dat do registru**

Nejprve se pošle START bit (ST), hned za ním adresa zařízení (SAD) spolu se zapisovacím bitem (W). Adresa je v případě akcelerometru nastavená na (1D)<sub>h</sub>. Dále slave posílá potvrzení (SAK) a master může poslat adresu registru (SUB), se kterým chce komunikovat. Slave zase pošle potvrzení (SAK) a master odešle data pro zapsání do registru (DATA). Slave pošle poslední potvrzení (SAK) a master nastaví STOP bit (SP).

### **Čtení dat z registru**

Jestliže chceme přečíst data z některého z registrů, pak postupujeme obdobně jako u zápisu, jen s menší změnou, která je patrná i z obrázku 1.13. Jako první je tedy

potřeba poslat START bit (ST), poté adresu akcelerometru (SAD) spolu s bitem zápisu (W). Přejde potvrzení (SAK). Nyní se odešle adresa registru (SUB) a přijde další potvrzení (SAK). V tuto chvíli dochází ke změně oproti zápisu, protože se pošle opětovný START bit (SR) a adresa akcelerometru (SAD) spolu se čtecím bitem (R). Přejde opět potvrzení (SAK) a bezprostředně za ním data z žádaného registru (DATA). Master ještě posílá potvrzení (NMAK) a teprve poté STOP bit (SP).

Master	ST	SAD + W		SUB		DATA		SP
Slave			SAK		SAK		SAK	

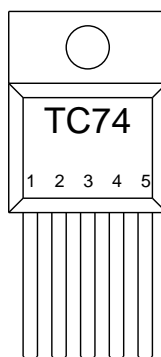
Obr. 1.12: Zápis dat do registru akcelerometru přes I<sup>2</sup>C [15]

Master	ST	SAD + W		SUB		SR	SAD + R			NMAK	SP
Slave			SAK		SAK			SAK	DATA		

Obr. 1.13: Čtení dat z registru akcelerometru přes I<sup>2</sup>C [15]

## 1.4.2 Senzory teploty

V mém modelu je vhodné i měřit teplotu motoru, akumulátoru nebo teplotu okolního vzduchu. Pro jednoduchost jsem využil teplotní senzory firmy Microchip s označením TC74, které disponují digitální sběrnici I<sup>2</sup>C, tak není náročné je integrovat do mé koncepce. Nevýhodou tohoto senzoru je přesnost a jeho nízké rozlišení, které dosahuje přesnosti 1°. Ale v případě mé aplikace je to naprosto dostačující.



Obr. 1.14: Teplotní senzor TC74 [11]

Zmiňované senzory se běžně používají např. ve výpočetní technice na měření teploty v počítači. Konkrétně jsou integrované v pevných discích, na základních deskách a grafických kartách. I v tomto případě je jejich přesnost dostačující. Jediným rozdílem oproti mému použití je rozměr součástky, protože já využil velké pouzdro TO-220 (obr. 1.14), ale obvykle se používá SOT-23. Na funkčnost to samozřejmě nemá žádný vliv.

Teplotní senzor obsahuje dva registry. První je registr pro nastavení (*CONFIG*) a druhý obsahuje naměřenou teplotu (*TEMP*). Přístup k oběma registrům probíhá pomocí sběrnice I<sup>2</sup>C. Adresa teplotního senzoru je dána výrobcem podle označení součástky. Je možné sehnat až 8 teplotních senzorů s různou adresou (tabulka 1.2).

Tab. 1.2: Označení teplotních senzorů TC74 a jejich adresy. [11]

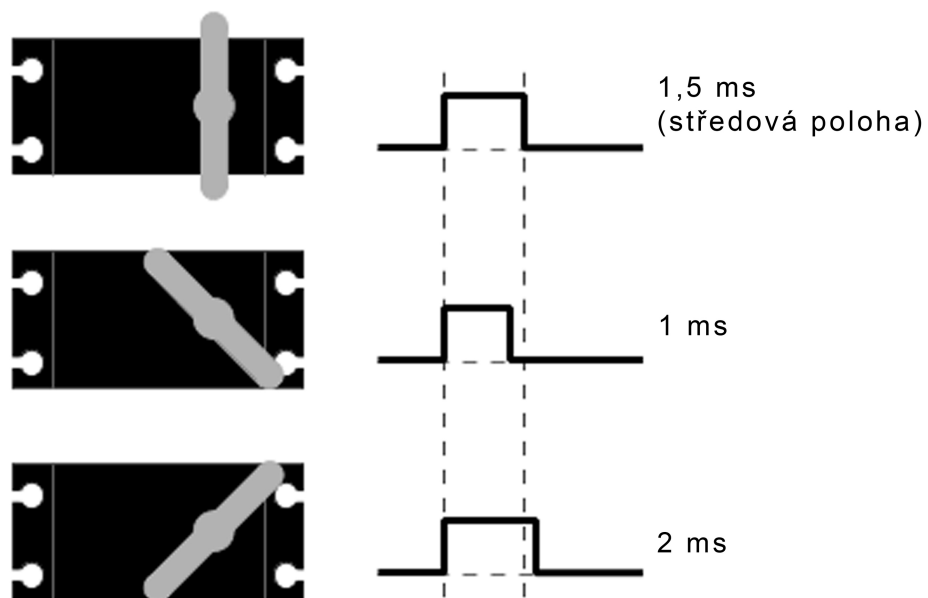
označení	adresa (hex)
TC74A0	92
TC74A1	93
TC74A2	94
TC74A3	96
TC74A4	98
TC74A5	9A
TC74A6	9C
TC74A7	9E

## 1.5 Řízení servomotoru

Servomotory jsou klasické elektromotory, které mají převody a obsahují řídicí elektroniku se senzorem, pro pohodlné a rychlé nastavování polohy v rozsahu 180°. Tyto motory jsou vhodné pro modely a podobné aplikace zvláště díky jejich nižší spotřebě oproti krokovým motorům, rychlé reakci na změnu a relativně přesným nastavením úhlu. Díky převodům uvnitř motoru, jsou také silnější než klasické elektromotory.

V této práci je využitý servomotor především na zatáčení kol vozidla, kde je možné rozeznávat až 100 jednotlivých poloh otočení. Dále je uvažováno použití servomotoru na otáčení infračerveného senzoru vzdálenosti a detekci objektů a to proto, aby senzor měl širší rozsah záběru. Pro obě aplikace je servomotor nejvhodnější řešení.

Řízení takového servomotoru se realizuje posíláním impulsů přímo do jeho řídicí elektroniky, která vyhodnotí natočení podle šířky impulsu. Impulsy se posílají o frekvenci 30 až 90 Hz. Mikrokontrolér tedy generuje impulsy PWM (pulsně šířková modulace). Šířka impulsu může být 0,5 až 2,5 ms. Přesné informace o délce pulsu jsou na obrázku 1.15.



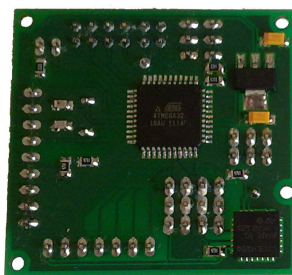
Obr. 1.15: Impulsy pro řízení servomotoru [17]

## 2 PRAKTICKÁ REALIZACE

Prakticky bylo realizováno a vyzkoušeno několik řešení a ne všechny vedly k úspěšnému cíli. Proto se zde už budu zabývat pouze takovým řešením, které je vhodné pro následující projekt a vede ke kvalitnímu výsledku. Taková řešení jsou celkem dvě a liší se pouze v použitém protokolu pro bezdrátový přenos dat a tudíž i modulem realizujícím komunikaci mezi oběma stranami. Kromě elektrotechnické části však bylo nutné napsat i programy pro řízení a zpracování všech výsledků. Jednalo se o programování softwaru pro PC, který je napsán v jazyce Visual Basic 6.0 a firmwaru pro mikrokontrolér, psaného v jazyce C pro mikrokontroléry AVR a laděného v AVR Studiu 4 od firmy Atmel.

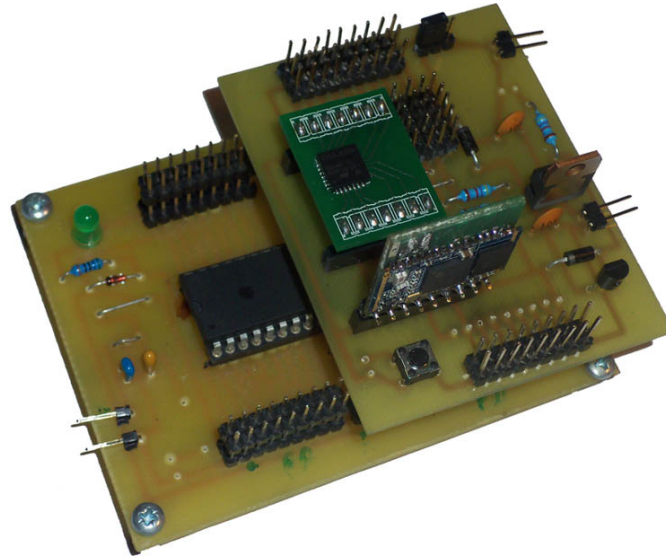
### 2.1 Dvě různá testovaná řešení

Jedním z prvních úspěšných pokusů bylo využití bluetooth modulu BTM-112 pro realizaci bezdrátového přenosu dat. Výhodou bylo, že na straně počítače nebyla nutná žádná hardwarová část, protože bluetooth je integrovaný přímo v notebooku. Tím pádem se celé řešení zjednodušilo a bylo potřeba realizovat pouze hardware ve vozidle. Blokové schéma je na obrázku 2.7, ze kterého vyplývá, že komunikace mezi BTM-112 a mikrokontrolérem Atmel probíhá pomocí vestavěné sběrnice USART. Rychlost přenosu dat byla nastavena na 115 200 bps z důvodu rychlých reakcí a odezvy. Fotografie vývojové desky je na obrázku 2.2 a fotografie hotové desky je na obrázku 2.1.



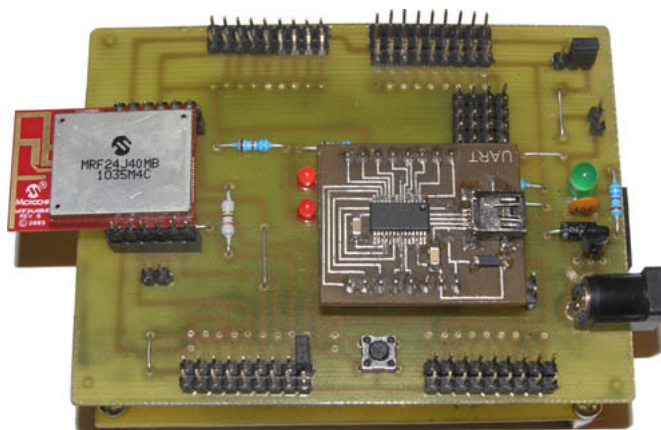
Obr. 2.1: Fotografie hotové desky sestavené na základě blokového schématu na obrázku 2.7

Problém nastal s dosažitelnou vzdáleností, která je pouze několik metrů bez použití antény a přibližně 10 m s malou wi-fi anténou. Kromě toho, je obrovskou nevýhodou, že se modul po připojení tváří jako další sériový port v počítači. Pro ovládací software to je každopádně výhodné, ale jakmile se modul dostane mimo



Obr. 2.2: Fotografie vývojové desky sestavené na základě blokového schématu na obrázku 2.7

dosah, pak spadne celý ovládací program, protože zmizí sériový port. Určitě by bylo možné chybu ošetřit a odpojit se od portu, ale bylo vhodnější použít vysílací a přijímací modul, který by byl fyzicky připojený k počítači kabelem a pouze komunikoval vzduchem s dalším modulem. V tom případě, i když by bylo vozidlo mimo dosah, program stále může komunikovat s modulem připojeným kabelem. Ještě se vyskytla jedna dosti významná komplikace a to spotřeba bluetooth modulu, která se pohybuje v klidovém stavu okolo (60 mA) a to v případě napájení akumulátoru je příliš mnoho.



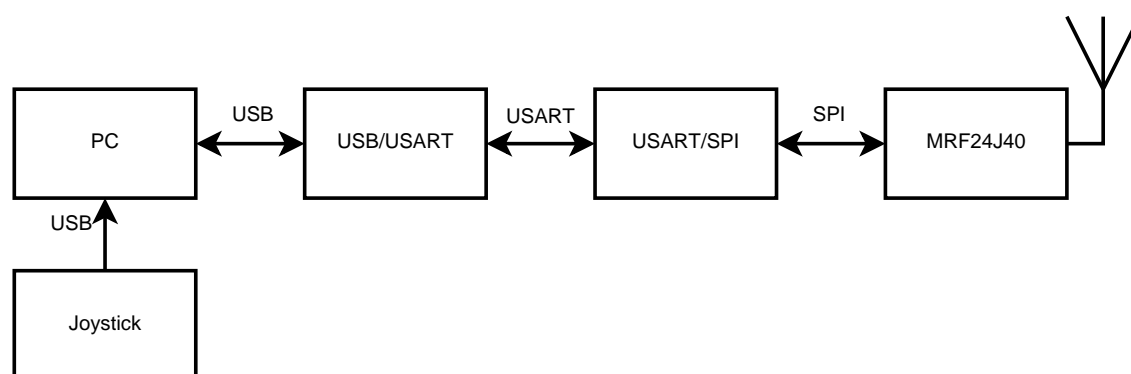
Obr. 2.3: Fotografie vývojové desky sestavené na základě blokového schématu na obrázku 2.8

Proto jako vhodnější jsem zvolil použití dvojice modulů od firmy Microchip s označením MRF24J40MB, které používají, mimo jiné, protokol ZigBee. Detailně tyto moduly jsou popsány v kapitole 1.3.2. Moduly byly vybrány pro jejich rozsáhlé možnosti použití a především nižší spotřebě, která je v klidovém stavu přibližně 25 mA. Rozhodující byl i dosah s integrovanou anténou, který je uváděn v literatuře [10] více než 1 km. Pro testování byly vytvořeny dva stejné moduly vyobrazeny na fotografii 2.3. Realizováno tohle bohužel nakonec nebylo, protože nastala při testování chyba, kdy bylo možné realizovat komunikaci pouze na jednotky metrů. Proto se od tohoto řešení upustilo a realizován je přenos dat přes bluetooth, jak bylo původně plánováno.

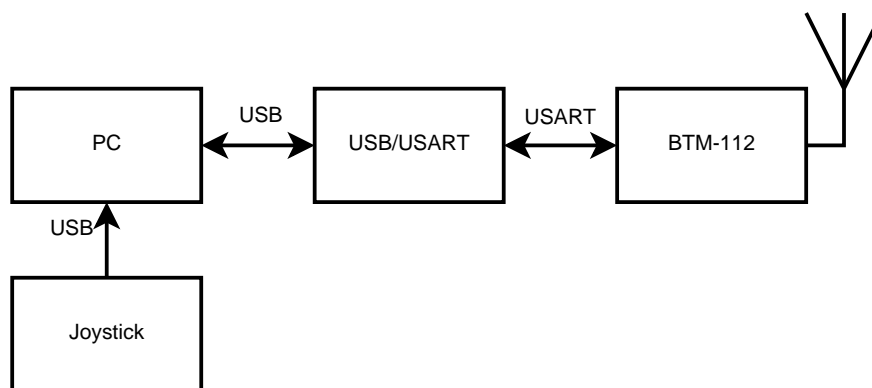
Oproti původnímu prvnímu testovanému řešení jsem ale využil dva bluetooth moduly. Jeden ve vozidle, jak jsem testoval v první variantě a druhý je připojený k počítači přes převodník USB/USART, aby se eliminovaly problémy prvního řešení. Oba moduly jsou navíc vybaveny externí anténou a tím se zvýšil dosah až na 100 m ve volném prostředí. Fotografie hotového modulu připojeného k počítači je na obrázku 2.4.



Obr. 2.4: Fotografie hotové desky sestavené na základě blokového schématu na obrázku 2.6



Obr. 2.5: Testované blokové schéma části připojené k PC v případě použití ZigBee



Obr. 2.6: Realizované blokové schéma části připojené k PC v případě použití modulu BTM-112

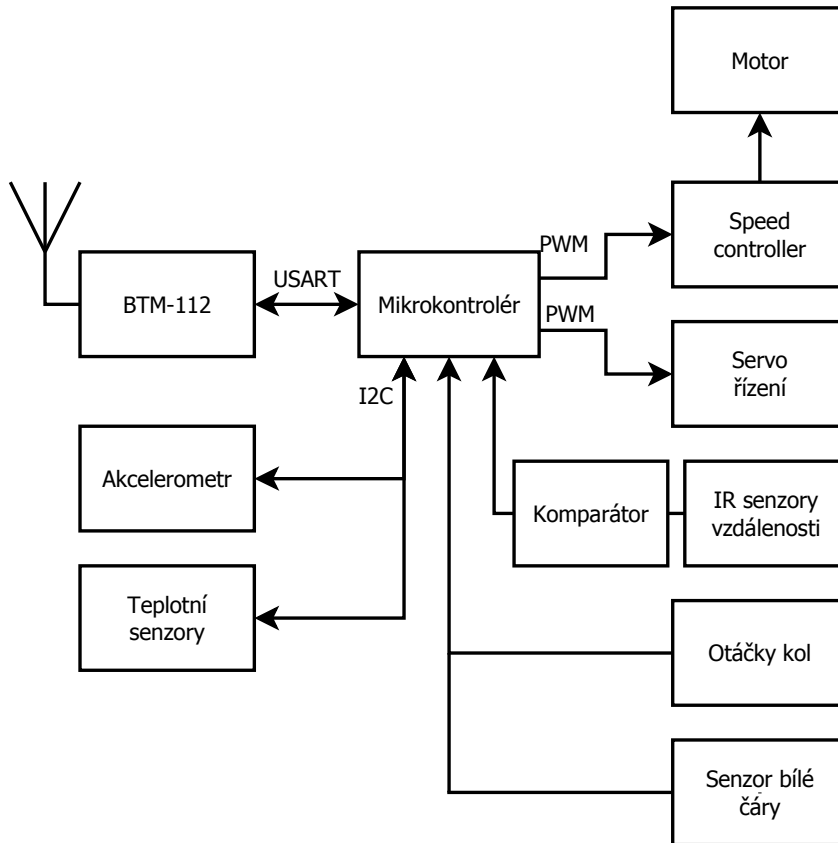
Praktické použití v celém systému je naznačeno na blokových schématech, které jsou rozděleny na dvě části. První komunikační část kabelově spojená s PC (obr. 2.5 a 2.6), a druhá část, která je zde uvedena dvakrát a liší se pouze v použitém přijímacím modulu, kompletně obsažená v ovládaném vozidle (obr. 2.7 a obr. 2.8). Jednotlivé bloky těchto schémat jsou popsány v kapitole 2.2, kde je i vysvětlena praktická realizace a komunikace mezi bloky.

## 2.2 Popis jednotlivých bloků schématu

V této kapitole dopodrobna rozeberu realizované i testované zapojení, které je znázorněno na blokových schématech na obrázcích 2.5, 2.6, 2.7 a 2.8.

### 2.2.1 Joystick

K ovládání vozidla jsem se rozhodl využít klasický herní joystick s analogovými páčkami (obr. 2.9), který se připojuje k počítači přes USB port. Výhodou použití joysticku je velké množství nastavitelných mezikroků páčky a díky tomu je možné pohodlně ovládat zatáčení kol vozidla v širokém rozsahu a také snadno volit rychlost pohybu vozidla. Oproti tomu na klávesnici, v případě použití šípek, které mají hodnotu pouze sepnuto a rozepnuto, by vozidlo jelo maximální rychlostí nebo stálo respektive zatočilo úplně vlevo, či vpravo. Ale joystick má 65 536 mezistavů na každé ose a vždy v půlce tohoto rozsahu je nulová poloha. Proto k zatáčení stačí pouze přepočítat tyto mezistavy na 100 kroků servomotoru a pro změnu rychlosti stejným způsobem přepočítat na 100 stavů pro speed controller. Přepočet mezi hodnotami joysticku a hodnotami pro servo respektive speed controller řeší program v počítači podle vzorce (2.1).



Obr. 2.7: Realizované blokové schéma hardwarové části ve vozidle s použitím BTM-112

Joystick disponuje dvěma analogovými páčkami, takže kromě samotného řízení je možné druhou použít např. na ovládání serva, které bude otáčet kamerou, IR senzorem nebo jakoukoliv jinou součástí vozidla.

Dále je možné využít na joysticku klasické tlačítka např. pro spínání světel nebo na dotazování se mikrokontroléru ve vozidle na určité stavy.

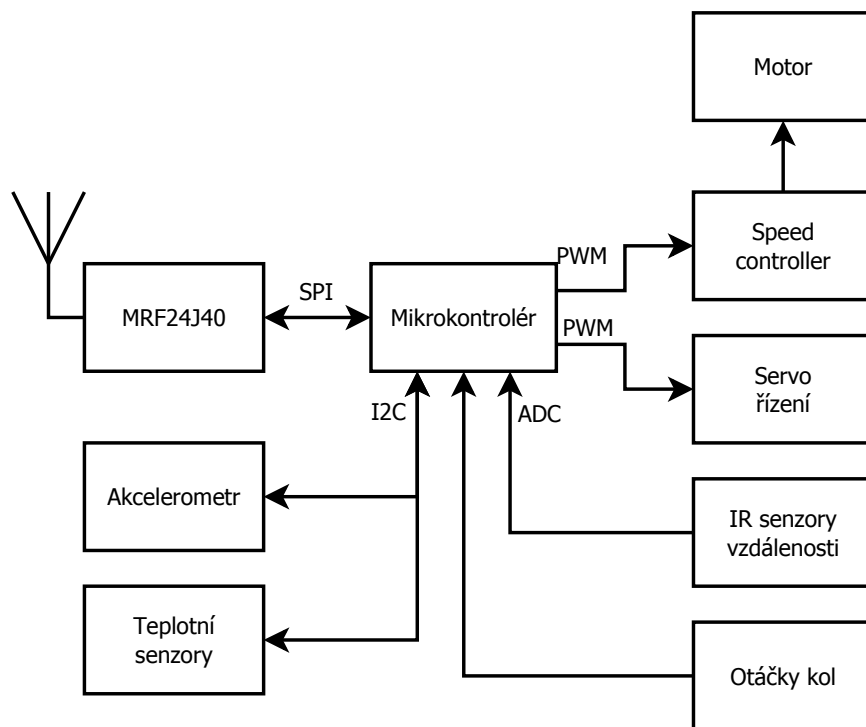
Vzorec pro přepočítání hodnot joysticku na hodnoty servomotoru respektive hodnoty speed controlleru

$$Y = \left| 100 + \frac{X}{655,35} \right|, \quad (2.1)$$

kde  $X$  je souřadnice joysticku.

### 2.2.2 PC

Pro sjednocení veškerého hardwaru jsem využil notebook s operačním systémem Microsoft Windows 7, který disponuje minimálně jedním USB portem a rozhra-



Obr. 2.8: Testované blokové schéma části ve vozidle s použitím ZigBee

ním bluetooth. Na tomto notebooku je spuštěný software, který jsem naprogramoval pro ovládání vozidla. Tento software zpracovává hodnoty joysticku, přepočítává je a podle toho odesílá datové rámce na virtuální COM port vytvořený spojením bluetooth.

Ovládací software je naprogramován v jazyce Visual Basic 6.0 a to především proto, že má snadný přístup k portům a je jednoduchý k programování následující aplikace. Dalším důvodem bylo i to, že s tímto jazykem mám pro práci s hardwarem největší zkušenosti.

Hlavní okno aplikace (obr. 2.10) samo o sobě neobsahuje skoro žádné prvky, jen několik formulářových voleb pro základní nastavení celé aplikace a pole pro výpis informací přijatých z vozidla. Na místo ovládacích prvků je použitý joystick.

Po spuštění programu je nutné nejprve nastavit vhodný port a uskutečnit spojení mezi bluetooth ve vozidle a počítačem. V tu chvíli se také zapne časovač, který se každých 20 ms dotazuje knihovny ve windows na informace o poloze páček joysticku. Knihovna vrátí polohu a pak následuje přepočet na hodnoty pro ovládání serva podle vzorce 2.1. Poté následuje podmínka, zda-li se hodnota změnila oproti minulé, aby se neodesílalo zbytečně více stejných rámců stále za sebou. Rámec se tedy odešle pouze při změně. Pak už se jen zapíše nová hodnota do proměnné s předcházející hodnotou pro další porovnání. V průběhu ještě navíc dochází k dělení rychlosti, tak aby bylo



Obr. 2.9: Joystick použitý v mé práci

možné řídit model vozidla i v malých prostorech. Pak tedy maximum na joysticku odpovídá podělené rychlosti, která se dá rovněž nastavit pomocí formulářové volby.

Dále program přijímá informace od vozidla o teplotě, překážkách kolem vozidla a bílé čáře pod vozidlem. Tyto informace jsou graficky zobrazovány v okně programu.

### 2.2.3 Převodník z USB na USART

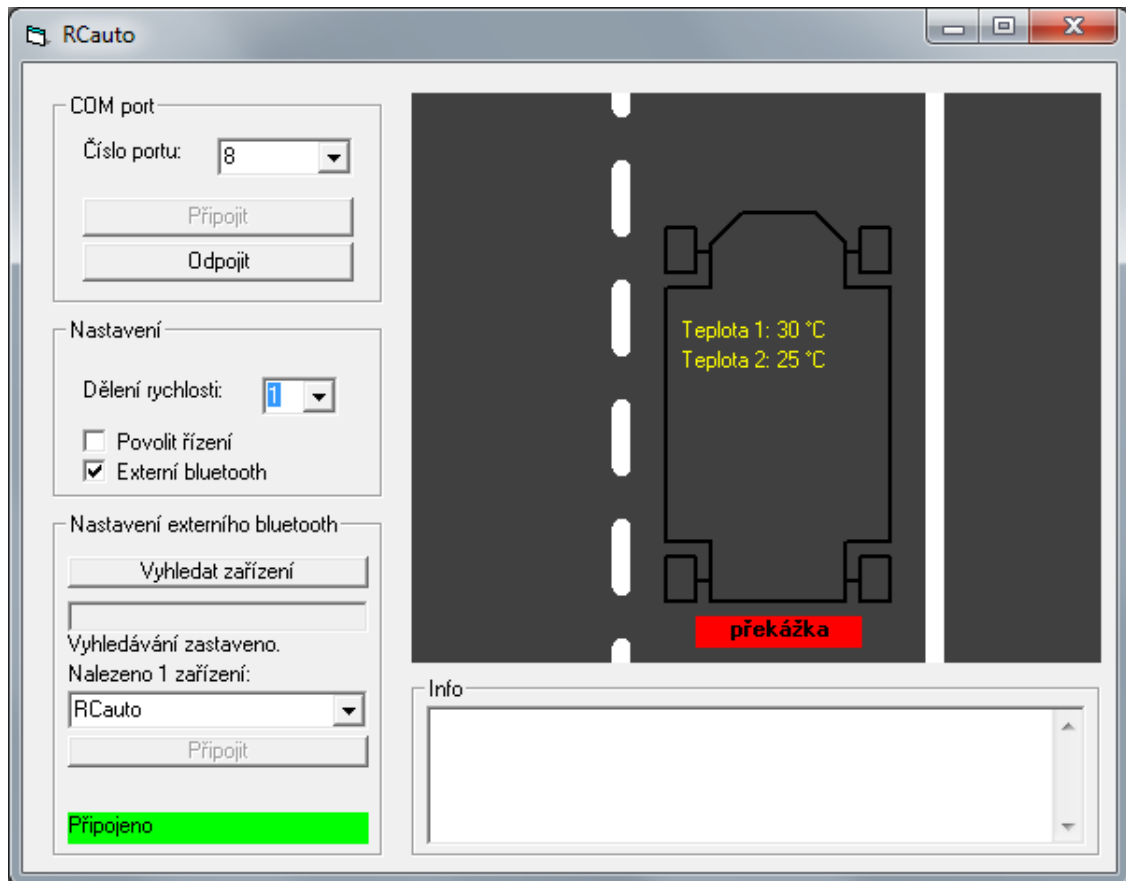
Převodník z USB na USART jsem již využil v kapitole 1.3.1, kdy bylo potřeba se spojit s bluetooth modulem kabelově. Nyní tento převodník slouží pro komunikaci mezi počítačem a mikrokontrolérem Atmel, který řeší převod mezi USART a SPI.

Převodník je realizován integrovaným obvodem FT232RL, který vyrábí firma FTDI chip. Schéma převodníku je velice jednoduché, stačí pouze připojit k integrovanému obvodu USB konektor, napájecí napětí (max. 6 V), několik blokovacích kondenzátorů a volitelně dvě LED diody pro indikaci přenosu dat v obou směrech. Vše je zobrazeno na obrázku 2.11.

### 2.2.4 Převodník z USART na SPI

Dalším blokem je mikrokontrolér Atmel, který realizuje převod mezi rozhraními USART a SPI (popis funkčnosti těchto rozhraní je v kapitolách 1.2.1 a 1.2.3). Konkrétně je mikrokontrolér naprogramován tak, aby přijal data přes rozhraní USART z počítače a podle nich vyhodnotil, co pošle modulu MRF24J40MB přes SPI. V případě příjmu dat z bezdrátového modulu, se celý proces děje naopak. Nejprve se přijme informace přes SPI poslaná z vozidla, a poté mikrokontrolér danou informaci přešle přes USART a USB do počítače.

Jedná se tedy pouze o aktivaci hardwarové podpory SPI a USART v mikrokontroléru tak, aby bylo možné komunikovat s okolními bloky. Dále jsou ještě v mik-



Obr. 2.10: Aplikace naprogramovaná pro testování a ovládání modelu vozidla

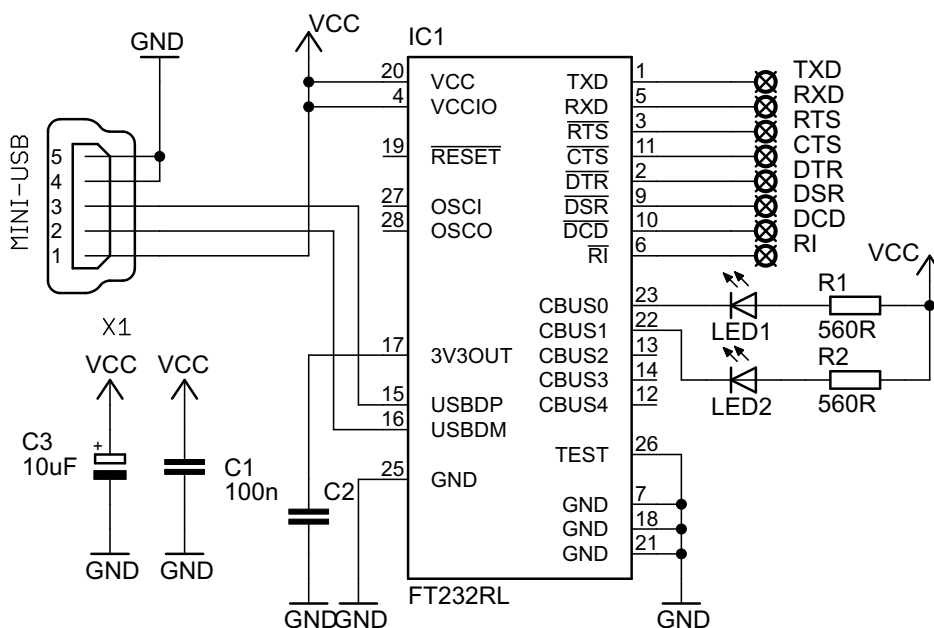
rokontroléru implementovány komunikační funkce přímo pro inicializaci a nastavení ZigBee modulu.

### 2.2.5 MRF24J40MB a BTM-112

Tyto dva bloky schématu jsou již detailně popsány v úvodu. Zde už se jedná pouze o připojení nastavených modulů ke správným sběrnicím a to v případě BTM-112 je USART a modul MRF24J40MB je připojen na sběrnici SPI.

**Sběrnice USART je nastavena následovně:**

- **Baud rate:** 115 200 bps
- **Datových bitů:** 8
- **Přita:** sudá
- **Stop bity:** 1
- **Řízení toku:** HW nebo žádné



Obr. 2.11: Schéma převodníku z USB na USART [4]

## 2.2.6 Mikrokontrolér

Srdcem celého schématu i vozidla je mikrokontrolér Atmel ATmega32 s externím krystalovým oscilátorem nastaveným na 16 MHz. Jeho výběr byl ovlivněn především velikostí paměti pro programovou část, ale i z důvodu velkého množství integrovaných hardwarových periférií, které je možné snadno používat téměř nezávisle na programu a tím také snížit velikost zdrojového kódu.

Program pro mikrokontrolér je naprogramován v jazyce C pro mikrokontroléry Atmel AVR a byl laděn ve vývojovém prostředí AVR Studio 4 od firmy Atmel, které je volně ke stažení. Programovací jazyk je velice podobný klasickému C, s tím rozdílem, že pro hardwarové periferie se zapisuje přímo do registrů a jsou k dispozici některé funkce navíc.

Na začátku programu se inicializují všechny sběrnice, hardwarové periferie a povolí se přerušení, které je hlavním stavebním prvkem programu. Jedná se především o přerušení ze sběrnice USART, které se provede pouze tehdy, když jsou na sběrnici data k příjmu. V přerušení se provádí v cyklu načítání jednotlivých znaků postupně do proměnné, která je jako pole znaků o délce 100 (dále nazývané jako řetězec). Poté je v nekonečném cyklu zjišťována přítomnost znaku hash (#) v načteném řetězci funkcí `strstr(text, "#")`, která vrací hodnotu NULL v případě, že se znak

nevyskytuje v řetězci. V opačném případě vrací hodnotu ukazatele na první výskyt tohoto znaku a tím je podmínka splněna. Poté se provádí rozdělení řetězce podle znaku dolar (\$) pomocí funkce `znaky[i]=strtok(text,delims)`, kde `text` je načtený řetězec a `delims` je řetězec obsahující znaky dolaru. Jednotlivé části řetězce jsou postupně vráceny v cyklu do pole řetězců `znaky` a nakonec se ještě provede vyprázdňení řetězce `text`, aby byl připraven pro další načtení znaků z USART.

Další částí nekonečného cyklu jsou podmínky porovnávající první řetězec v poli `znaky` funkcí `strcmp(znaky[1], "S")`, která porovnává dva řetězce a vrací hodnotu 0, když jsou řetězce shodné. Tyto podmínky zpracovávají znak v první části datového rámce podle obrázku 2.19, podle kterého se následně rozhoduje, jaký příkaz se vykoná. Jakmile je některá z těchto podmínek splněna, pak je uvnitř podmínky buď zápis do proměnné pro PWM nebo volání jiných funkcí ke splnění příkazu. Jako např. funkce pro čtení hodnot A/D převodníku, komunikaci se sběrnici I<sup>2</sup>C nebo zápis hodnot do registru příslušného portu k sepnutí výstupu.

### 2.2.7 Snímání polohy akcelerometrem

K mikrokontroléru je přes sběrnici I<sup>2</sup>C připojený akcelerometr, který snímá zrychlení vozidla. Teoretické informace o tomto modulu jsou detailně popsány v kapitole 1.4.1. Jak je napsané v teorii, akcelerometr obsahuje ke každé souřadnici dva registry a může nám informaci o zrychlení posílat v 12ti nebo 16ti bitovém výsledku. Na to, aby jsme dostali z výsledného zrychlení rychlost vozidla případně ujetou dráhu, je potřeba tento výsledek zintegrovat.

Mikrokontrolér bohužel integrály nezná, proto je potřeba načítat zrychlení z akcelerometru ve stejných intervalech a počítat integrál např. obdélníkovou metodou, kterou jsem zvolil já. Samozřejmě tím se snižuje přesnost výsledku. Proto je potřeba mít vzorkovací interval co nejkratší a to vyřeší přerušení čítače/časovače v mikrokontroléru, které se spouští vždy ve stejný čas a načte další hodnotu zrychlení. Tím je mikrokontrolér schopný načtené hodnoty zpracovat a vypočítat z nich okamžitou rychlost, průměrnou rychlost a ujetou dráhu vozidla. Tyto hodnoty pak uchovává v dočasné paměti a software v PC si je může vyžádat příkazem z tabulky 2.1.

### 2.2.8 Měření teploty

Kromě akcelerometru jsou na sběrnici I<sup>2</sup>C připojeny ještě teplotní senzory TC74, které jsou rovněž popsány již v kapitole 1.4.2, která se zabývá teoretickými údaji těchto senzorů. Sensory jsou určeny pro měření teploty motoru, akumulátoru a okolního prostředí. Výhodná je konstrukce těchto senzorů, protože s jejich pouzdem se snadno pracuje a mají kovovou plochu na přichycení přímo k měřené součásti. Další

výhodou je digitální výstup. Protože veškeré měření teploty a její zpracování se děje přímo v senzoru a přes sběrnici I<sup>2</sup>C už se čte pouze hodnota registru, ve kterém je uložena informace o teplotě. Mikrokontrolér tedy nemusí vůbec řešit žádný převod, protože dostane přímo konkrétní hodnotu teploty a tu posílá počítači na vyžádání příkazem z tabulky 2.1.

## 2.2.9 Servomotor řízení

Nyní se dostáváme k bloku, který je představován přímo konkrétním servomotorem s označením HS-645MG a ve vozidle je použitý k zatáčení přední nápravy. Jakým způsobem se servo ovládá je detailně popsáno v kapitole 1.5. V podstatě se jedná o jednoduchou PWM, která je realizována v mikrokontroléru čítačem/časovačem a ovlivňována přijatými datovými rámci z počítače. Tyto rámce odpovídají struktuře na obrázku 2.19 a tabulce 2.1.

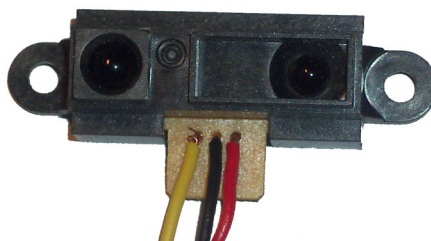
## 2.2.10 Měření vzdálenosti objektů

Blok pro měření vzdálenosti od objektů je realizován modulem zakoupeným ve firmě Snail Instruments (obr. 2.12). Tento modul s označením GP2Y0A41 měří vzdálenost objektů na principu infračervených vln, které modul vyšle, ony se odrazí a vrací zpět do přijímače. Modul mezitím počítá čas mezi odrazem a příjmem. Z toho vypočítá vzdálenost, která je reprezentována napětím na výstupu a to je úměrné vzdálenosti podle vzorce (2.2).

Vzorec pro výpočet vzdálenosti objektu podle výstupního napětí modulu [14]

$$l = \frac{2914}{ADV + 5} - 1, \quad (2.2)$$

kde  $l$  je vzdálenost a  $ADV$  je naměřená hodnota 10bitovým A/D převodníkem.



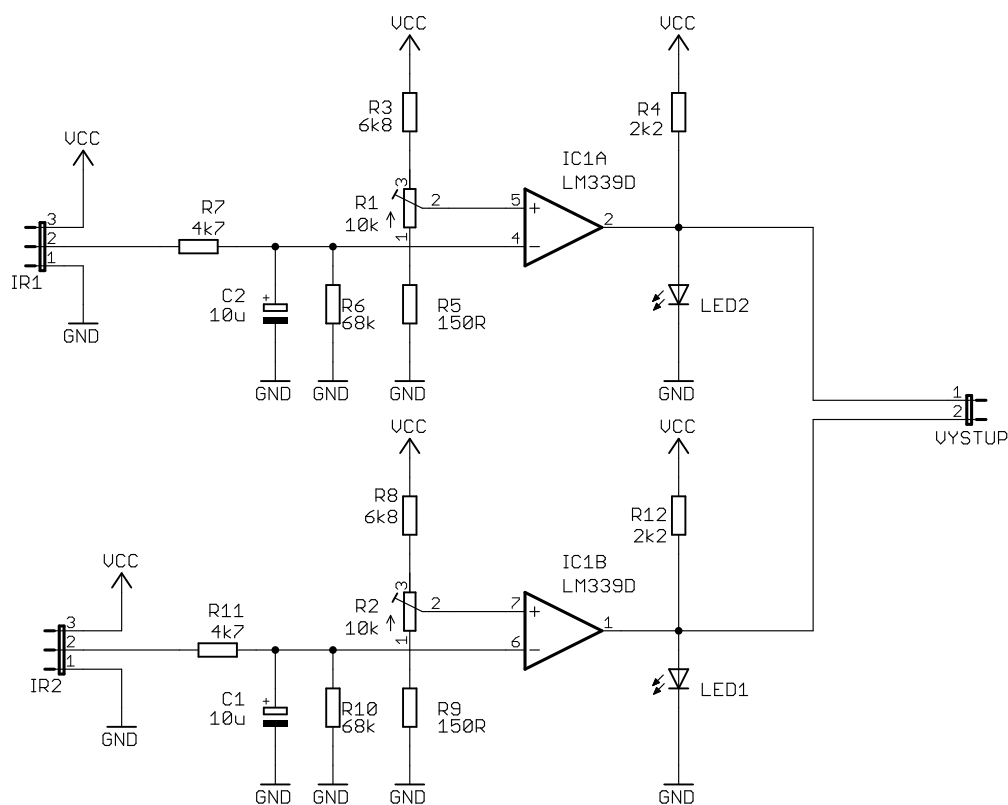
Obr. 2.12: Fotografie IR detektoru vzdálenosti GP2Y0A41

## 2.2.11 Komparátor

Blok analogového komparátoru jsem vložil do mého výsledného řešení až na poslední chvíli, protože jsem chtěl využít dva IR detektory překážek a měl jsem na desce vyvedený pouze jeden vstup A/D převodníku.

Použil jsem velice jednoduché a ověřené zapojení komparátoru s operačním zesilovačem LM338 (schéma na obrázku 2.13), který je výhodný především proto, že nepotřebuje symetrické napájení.

Vstup je označen na schématu jako IR1 (resp. IR2) a připojuje se k němu analogový výstup IR detektoru. Hodnota na vstupu je porovnávána komparátorem s hodnotou nastavenou pomocí děliče, který je možné regulovat trimrem R1 (resp. R2). Tím se tedy reguluje reakce na vzdálenost od překážky. Na výstupu komparátoru je logická hodnota, kterou zpracovává mikrokontrolér.



Obr. 2.13: Schéma analogového komparátoru

## 2.2.12 Měření a porovnávání otáček kol

Pro realizaci některých prvků ve vozidle je potřeba snímat počet otáček každého kola zvlášť. Proto bylo nutné zvážit, jaké senzory použít tak, aby jejich montáž do

modelu nebyla náročná a přitom byly dostatečně rychlé a přesné. Existuje několik variant, např. optická závora, mechanický spínač, magnetický spínač a mnohé další. Z nich jsem si zvolil magnetický jazýčkový spínač (obr. 2.14), protože stačí připevnit permanentní magnet na kolo a magnetický spínač dává impulsy podle otočení kola. V případě využití více permanentních magnetů na kole se dá dosáhnout rychlejších reakcí, ale to pouze za předpokladu, že magnetický spínač bude dostatečně rychlý. Kladem takového řešení je i jeho cena, protože permanentní magnety nejsou nijak drahé a magnetický spínač také ne. Místo magnetického spínače je možné využít i Hallovu sondu, kterou je možné snímat vyšší otáčky.



Obr. 2.14: Fotografie použitého magnetického spínače

### 2.2.13 Speed controller a motor

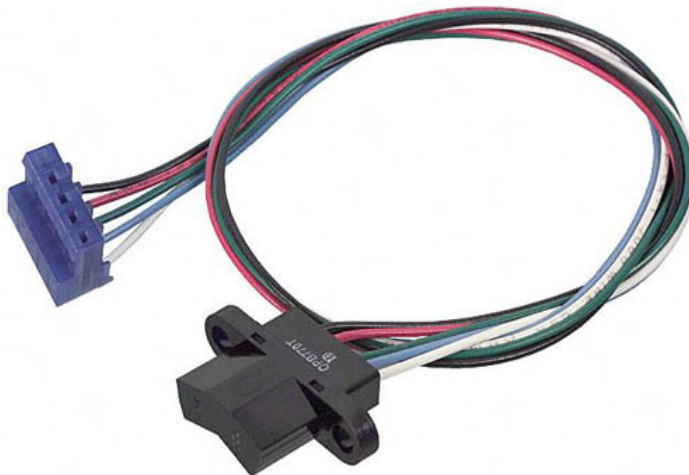
Nedílnou součástí vozidla je pohon. V mém případě se jedná o klasický elektromotor na napětí 7,2 V, který je spínán budičem speed controller ESC300 (obr. 2.15). Tento budič se vyrábí přímo pro modely vozidel a byl zvolen především proto, že jeho ovládání je snadné a díky němu je možné měnit rychlost otáček motoru. Řízení tohoto budiče se děje pomocí PWM stejně tak, jako u servomotoru. Šířkou impulsu se nastavuje rychlost otáček.



Obr. 2.15: Fotografie použitého speed controlleru ESC300

### 2.2.14 Senzor bílé čáry

Moderní automobily dokáží udržovat vozidlo v pruhu podle bílých čar na vozovce a případně řidiče upozornit při nechtěném vybočení z pruhu. Jak již jsem se zmínil v úvodu, tak se k tomu využívá nejčastěji kamera, která snímá prostor před vozidlem a patřičný software vyhodnocuje jestli je vozidlo v pruhu.

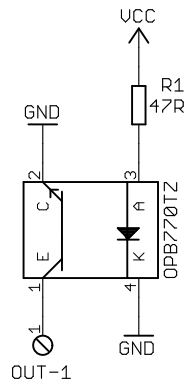


Obr. 2.16: Senzor QRD1114 pro detekci bílé čáry

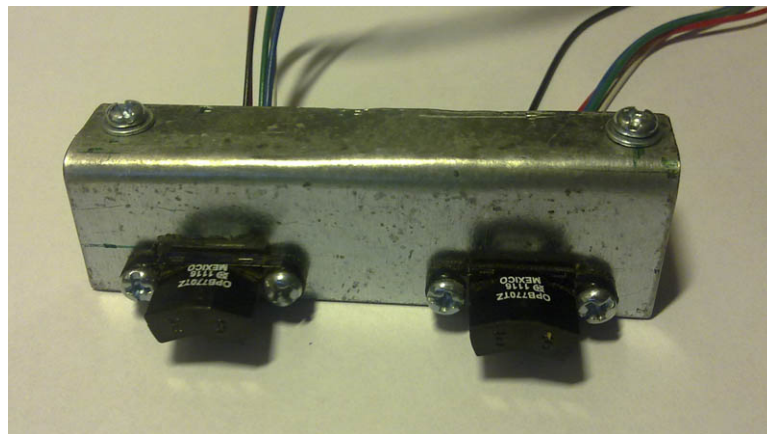
Já jsem pro tuto nadstandardní funkci mého vozidla využil dva senzory s označením OPB770TZ (obr. 2.16). Jeden senzor se skládá ze dvou prvků. Jedním je IR dioda a druhým je fototranzistor. Schéma zapojení jednoho senzoru je na obrázku 2.17. Podle schématu je zřejmé, že IR dioda je stále pod napětím a stále osvětluje plochu před sebou. Jestliže se světelný paprsek odrazí od plochy, pak se jedná o bílou barvu, protože od černé se infračervené světlo neodráží, ale je pohlceno. V případě bílé barvy je tedy na výstupu log. 0, protože je fototranzistor v sepnutém stavu. U černé barvy je tomu opačně. Fototranzistor je rozepnutý a na výstupu je log. 1.

Senzory jsem využil dva proto, aby bylo možné detekovat i směr vybočení vozidla. Na obrázku 2.18 je výsledný modul, který jsem umístil pod vozidlo. Je nutné, aby vzdálenost mezi senzorem a vozovkou byla méně než 1 cm. To je nevýhodou celého řešení detekce bílé čáry, protože na silnicích jsou různé výmoly a tak by detekce nebyla přesná. Proto se tento způsob v praxi nepoužívá.

V hlavním okně aplikace se vybočení znázorňuje zčervenáním patřičné čáry vedle obrázku vozidla.



Obr. 2.17: Schéma zapojení jednoho senzoru



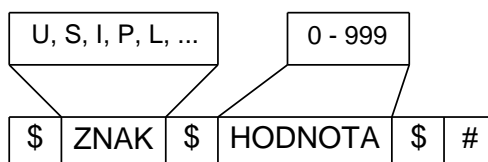
Obr. 2.18: Fotografie vytvořeného senzoru na detekci bílé čáry

## 2.3 Celková komunikace

Nyní, když jsem vysvětlil všechny bloky mého řešení, se dostávám k podstatné části a tou je přenos dat, jejich vyhodnocení, zpracování a popř. poslání správné odpovědi. Proto tedy začnu ze strany PC. Joystick připojený do USB počítače s ovládacím programem posílá informace o poloze analogových páček softwaru v PC. Poté se ovládací program dotazuje v pravidelných dosti krátkých intervalech knihovny implementované ve Windows, která slouží na komunikaci se vstupně/výstupními zařízeními, na hodnoty v podobě dvou celých čísel v rozsahu 0 – 65 536. Tyto čísla představují polohu páčky v osách x a y, kdy neutrální tedy středová poloha nastává tehdy, když program dostane přesně poloviční číslo 32 768. Menší čísla jsou pak záporné hodnoty na osách a větší představují kladnou část osy. Dále se program může ptát na hodnoty tlačítek na joysticku a tím zjišťovat zda-li je tlačítko v danou chvíli stisknuté či nikoliv. Podle toho program dále vyhodnocuje jestli se změnila poloha páčky oproti

minulému cyklu a jestliže ano, pak pošle na virtuální sériový port příslušný rámeček dat.

Příklad použitého rámce je zobrazen na obrázku 2.19. Jedná se o znakově orientované rámce, kdy každý rámeček začíná znakem dolar (\$), po něm se posílá příkaz pro vykonání a jeho hodnota. Za příkazem i hodnotou je znovu obsažený znak (\$), aby bylo možné rozdělit tento rámeček a vytáhnout z něj příslušná data. Konec rámce je vždy označen znakem hash (#).



Obr. 2.19: Datový rámeček, který přenáší informace mezi PC a vozidlem

Knihovna pro práci s porty rozloží každý znak na jednotlivé bity podle nastaveného USART přenosu. V mém případě se jedná o 8 bitů, jeden stop bit a jeden paritní bit, který kontroluje lichou paritu. Pak se celá zpráva přenesou rychlostí 115 200 bit/s do integrovaného bluetooth modulu v notebooku, který ji vzduchem přenesou do modulu BTM-112 a ten ji předá mikrokontroléru ve vozidle také přes USART. Mikrokontrolér musí mít stejné nastavení USART jako BTM-112 a jako počítač, který vysílá data.

Příjem dat v mikrokontroléru probíhá jako přerušení na vstupu USART a přijímá se v cyklu znak po znaku. Jakmile se v přijímaném řetězci objeví hash (#), pak je řetězec rozdělen na dvě části podle znaku dolar (\$). Tyto dvě části jdou k porovnání. První část je porovnávána jako první a určuje příkaz, který se má vykonat. Všechny příkazy, které je možné použít jsou popsány v tabulce 2.1. Jakmile se přijatý příkaz shoduje s některým v tabulce, pak se zpracovává druhá část řetězce, která určuje hodnotu.

Jestliže se jedná o příkaz U nebo S, pak se přečtená hodnota ukládá do proměnné, která nastavuje šířku impulsů u PWM a podle ní se otáčí servomotor řízení nebo nastavuje speed controller pro ovládání rychlosti jízdy. Jakmile je přijatý příkaz I nebo P, jedná se pak o čtení hodnoty A/D převodníku a její přepočtení na požadovaný tvar. Tím je myšleno, že se přepočítá hodnota z registru na napětí podle zvoleného referenčního napětí podle vzorce (2.3). Dalším případem je příkaz L, který pouze nastaví na příslušném portu log. 1, pro sepnutí relé, tranzistoru nebo rozsvícení přímo LED diody. Příkaz T, aktivuje I<sup>2</sup>C linku, pošle adresu příslušnému teplotnímu čidlu a vyžádá od něj hodnotu z registru *TEMP*, který obsahuje aktuální teplotu.

Pro přepočítání hodnoty registru A/D převodníku na napětí se používá následující vzorec

$$U = \frac{CISLO \cdot 2,56}{1024}, \quad (2.3)$$

kde *CISLO* je hodnota uložená v registru a *U* je měřené napětí v rozsahu 0 až 2,56 V. Vzorec je pro referenční napětí 2,56 V.

Tab. 2.1: Přehled dostupných příkazů pro model vozidla

příkaz	hodnoty	popis
U	100 – 200	Příkaz nastavuje jízdu vozidla.
S	100 – 200	Příkazem se zatáčí.
T	1 – 3	Příkaz se dotazuje teplotního senzoru přes I <sup>2</sup> C na aktuální teplotu.
V	0 – 999	Dotaz na počet impulsů z předního kola. Podle toho se pak počítá ujetá vzdálenost.
P	0 – 1	Aktivace/deaktivace detektorů překážek.
I	D1, U1, 0	Vozidlo posílá informaci o překážkách.
C	R1, L1, LR0	Vozidlo posílá informaci o bílé čáře.
A	DL, DR, 0	Vozidlo posílá informaci o ASR.

## 2.4 Rozmístění a konstrukce jednotlivých prvků ve vozidle

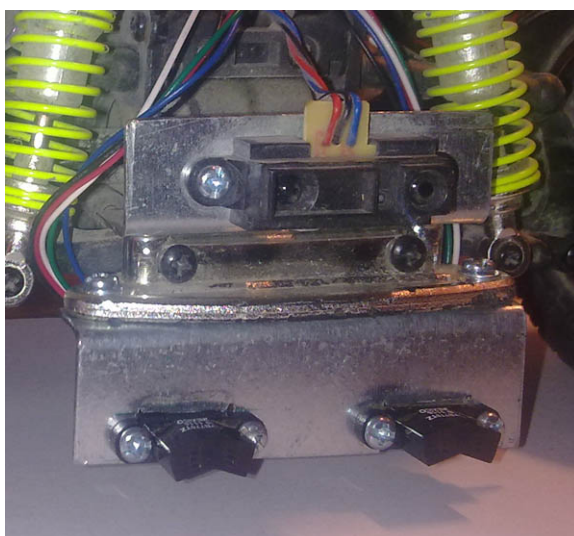
V průběhu tvorby mé práce jsem byl několikrát nucen vhodně volit rozmístění všech prvků ve vozidle tak, aby nepřekážely a plnily správně svou funkci. Výsledné rozmístění je naznačeno na obrázku 2.25, kde každý prvek či senzor má svoje číslo. Pod číslem 1 je IR detektor vzdálenosti, který jsem umístil vepředu vozidla, aby detekoval překážky před vozidlem. Analogicky je umístěný také IR detektor vzdálenosti vzadu za vozidlem pro detekci objektů za vozidlem při couvání, který je na obrázku pod číslem 2. Čísla 3 a 4 jsou IR detektory bílé čáry umístěné vzadu na vozidlem a snímají cestu. Tyto senzory musejí být umístěny 1 cm od vozovky, aby správně dokázali detekovat bílou čáru. Dalšími senzory jsou teplotní čidla. Jedno je umístěno na akumulátoru (číslo 5) a druhé na bluetooth modulu (číslo 6), pro vnější měření teploty. Oba teplotní senzory jsou vyfotografovány na obrázku 2.23 a 2.24. Nakonec jsem umístil ještě magnetické spínače ke třem kolům (obr. 2.20). Jsou pod čísly 7, 8 a 9. Na předních kolech stačí pouze jeden snímač, protože ten se používá

pouze jako reference oproti snímačům na zadních kolech. Tyto snímače slouží pro vyhodnocení prokluzu kol a snímač číslo 7 také pro měření ujeté vzdálenosti vozidla.



Obr. 2.20: Umístění a přilepení magnetického spínače k přednímu kolu

Kromě rozmístění zmíněných prvků bylo také nutné je nějak uchytit na vozidlo. U teplotních sensorů to nebyl žádný problém, použil jsem elektrikářskou izolační pásku, kterou jsem senzory přilepil na zmíněná místa. Konstruktivně náročnější bylo umístění IR detektorů vzdálenosti a IR detektorů bílé čáry (obr. 2.21 a 2.22). Na tyto detektory jsem udělal z plechu speciální držáky. Na závěr jsem řešil senzory pro ASR, které jsou umístěny u kol (obr. 2.20). Tyto senzory jsem přilepil tavnou pistolí k vozidlu.



Obr. 2.21: Umístění a konstrukční uchycení IR detektoru vzdálenosti a IR detektorů čáry vzadu



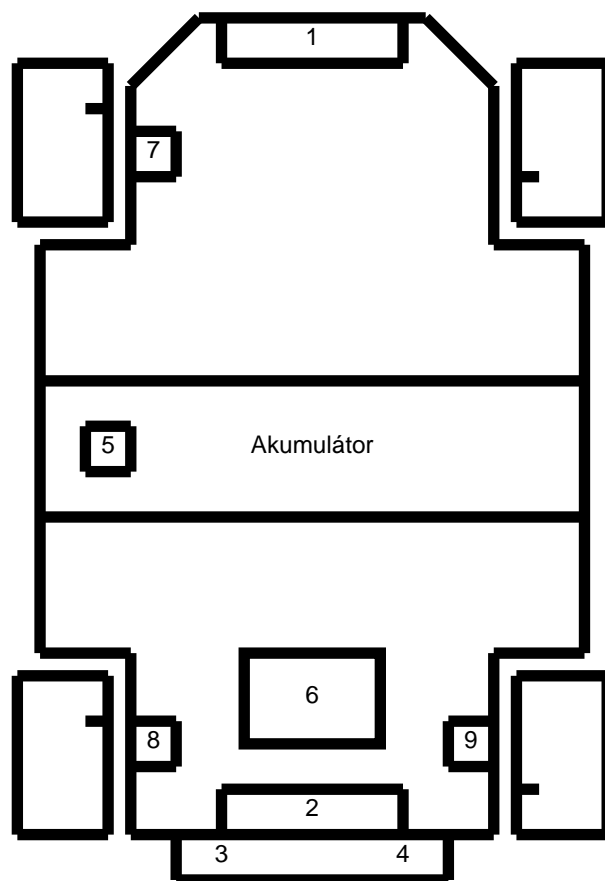
Obr. 2.22: Umístění a konstrukční uchycení IR detektoru vzdálenosti vepředu



Obr. 2.23: Umístění a uchycení teplotního čidla u akumulátoru



Obr. 2.24: Umístění a uchycení teplotního čidla na bluetooth modulu



Obr. 2.25: Grafické znázornění hotového vozidla s rozmístěním všech přidanych prvků

### 3 ZÁVĚR

Výsledkem mé bakalářské práce je plně pojízdný model vozidla. Navrhl jsem a realizoval vlastní elektroniku pro model vozidla tak, aby byl schopen nejen jezdit s regulací rychlosti a zatáčet v různých úhlech, ale také zpracovávat informace od různých senzorů a čidel, vyhodnocovat tyto informace a na jejich základě se rozhodovat. Kromě již zmíněných funkcí jsem vytvořil ovládání tohoto vozidla pomocí joysticku připojenému k počítači. Z toho plyne, že řízení vozidla je realizováno počítačem.

Ovládání vozidla jsem realizoval bezdrátovým přenosem dat. Proto jsem se nejprve zabýval výběrem vhodného modulu pro přenos dat rádiovým prostředím. Měl jsem na výběr buď modul bluetooth BTM-112 nebo ZigBee MRF24J40MB. Oba moduly jsem otestoval a jako vhodnější jsem zvolil bluetooth. I když se na první pohled zdá, že ZigBee modul díky své spotřebě a udávanému dosahu by byl lepší pro moji práci, tak jsem nakonec zvolil bluetooth modul. Především díky jeho jednoduchosti oproti ZigBee. A navíc se mi i s bluetooth modulem podařilo realizovat přenos dat až na 100 m díky externím anténám na obou stranách (vozidlo – počítač).

Kromě elektroniky bylo potřeba také napsat program pro počítač a program pro mikrokontrolér. Program pro počítač jsem napsal ve Visual Basicu 6 a program pro mikrokontrolér Atmel jsem psal v jazyce C pro mikrokontroléry Atmel AVR a kompiloval v AVR studiu 4. Oba programovací jazyky mají své výhody i nevýhody. Jako velkou nevýhodu Visual Basicu 6 bych viděl fakt, že není multiplatformní, ale je závislý na operačním systému od Microsoftu. Výhodou zmíněného programovacího jazyka je jeho jednoduchost a především snadné používání portů a komunikace s hardwarem. U programovacího jazyka použitého pro mikrokontrolér je velkou výhodou jeho jednoduchost a také spousta studijních materiálů pro programování. Oproti dalším jazykům C pro mikrokontroléry Atmel je také výhodou, že prostředí pro kompilaci a ladění je zcela zdarma přímo od firmy Atmel. Žádnou velkou nevýhodu jsem u tohoto programovacího jazyka nenašel, snad jen to, že existují i jednodušší jazyky, ale prostředí pro ně je nutné zakoupit.

Cílem mé práce bylo mimo jiné také realizovat některé moderní prvky používané v automobilech na mém modelu, otestovat je a případně zhodnotit jejich využití v praxi. Na mém modelu jsem realizoval hned několik různých moderních prvků, kterými jsou ASR, detekce překážek před a za vozidlem a detekce bílé čáry. Všechny tyto prvky jsou plně funkční. Informace o zmíněných moderních prvcích se posílají z vozidla do počítače a zobrazují se v ovládacím programu. Realizované prvky jsem navrhl jinak, než jsou použity u skutečných automobilů, kromě ASR, které se v mé práci příliš neliší od reálného použití. Zbývající dva moderní prvky jsem realizoval pomocí IR detektorů. U obou prvků jsem našel spíše nevýhody, které by mohly vést k problémům při nasazení u skutečných automobilů. Jedná se především o spoleh-

livost těchto detektorů. Kdy detektory překážek nemusejí detekovat vždy překážku, protože od černých objektů se IR paprsek neodrazí a tím tyto objekty nejsou detekovány. Tohoto jevu se využívá naopak u detekce bílé čáry, ale zde je nevýhodou malá maximální vzdálenost mezi senzorem a vozovkou.

Na závěr jsem ještě implementoval do mého vozidla teplotní senzory pro měření teploty akumulátoru a měření teploty okolí. Měření teploty je nutné především proto, aby se některá součástka nepřehřála a nezničila se případně některá mechanická část v automobilu. V mé práci počítám s tím, že je možné připojit i další senzory teploty např. k motoru nebo dalším částem, u kterých se předpokládá přehřátí.

Když ještě srovnám můj model vozidla s klasickými bezdrátově řízenými modely, tak hlavní výhodou mého modelu je větší dosah. Dále řízení joystickem, u kterého si člověk připadá jako, když hraje počítačovou hru a senzory, které brání při nižších rychlostech srážce s předmětem. Jednou z nevýhod může pro někoho také být nutnost použití počítače pro řízení modelu. Takže při jízdě po venku je nutné sebou vždy mít notebook a nestačí pouze ovladač jako je tomu u komerčních modelů.

## LITERATURA

- [1] Atmel Corporation *ATmega32* [online datasheet]. 2011, poslední aktualizace 26. 2. 2011 [cit. 18. 10. 2011]. Dostupné z: <[http://www.atmel.com/dyn/resources/prod\\_documents/doc2503.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc2503.pdf)>.
- [2] AXELSON, J. *USB Complete: Everything You Need to Develop Custom USB Peripherals*. Third edition. Madison: Independent Publishers Group, 2005. 594 s. ISBN 978-1-931448-02-4.
- [3] BURKHARD, M. *C pro mikrokontroléry*. Odpovědná redaktorka Iveta Kubíková. Praha: BEN - technická literatura, 2003. 280 s. ISBN 80-7300-077-6.
- [4] FTDI chip *Future Technology Devices International Ltd. FT232R USB UART IC* [online datasheet]. 2011, poslední aktualizace 20. 4. 2011 [cit. 18. 11. 2011]. Dostupné z: <[http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS\\_FT232R.pdf](http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf)>.
- [5] HRBÁČEK, J. *Komunikace mikrokontroléru s okolím 2*. Odpovědná redaktorka Iveta Kubíková. Praha: BEN - technická literatura, 2000. 152 s. ISBN 80-86056-73-2.
- [6] *HW group* [online]. 2011, poslední aktualizace 28. 10. 2011 [cit. 28. 10. 2011]. Hercules SETUP utility. Dostupné z: <[http://www.hw-group.com/products/hercules/index\\_en.html](http://www.hw-group.com/products/hercules/index_en.html)>.
- [7] *Květákov.NET* [online]. 2008, poslední aktualizace 11. 1. 2008 [cit. 23. 10. 2011]. Sběrnice I2C. Dostupné z: <<http://www.kvetakov.net/clanky/avr/79-sbornice-i2c.html>>.
- [8] MATOUŠEK, D. *Práce s mikrokontroléry ATMEL AVR AT90S*. Odpovědná redaktorka Iveta Kubíková. 2. vydání. Praha: BEN - technická literatura, 2006. 376 s. ISBN 80-7300-209-4.
- [9] MATOUŠEK, D. *Práce s mikrokontroléry ATMEL ATmega16*. Odpovědná redaktorka Iveta Kubíková. 1. vydání. Praha: BEN - technická literatura, 2006. 320 s. ISBN 80-7300-174-8.
- [10] Microchip *MRF24J40M* [online datasheet]. [cit. 10. 12. 2011]. Dostupné z: <<http://ww1.microchip.com/downloads/en/DeviceDoc/70599B.pdf>>.

- [11] Microchip *Tiny Serial Digital Thermal Sensor* [online datasheet]. [cit. 26. 11. 2011]. Dostupné z: <<http://ww1.microchip.com/downloads/en/DeviceDoc/21462c.pdf>>.
- [12] Rayson *Bluetooth Module BTM-112* [online datasheet]. 2011 [cit. 3. 11. 2011]. Dostupné z: <[http://www.soselectronic.cz/a\\_info/resource/d/BTM-112.pdf](http://www.soselectronic.cz/a_info/resource/d/BTM-112.pdf)>.
- [13] SAJDL, J. *autolexicon.net* [online]. [cit. 1. 11. 2011]. Dostupné z: <<http://cs.autolexicon.net>>.
- [14] Snail Instruments *GP2Y0A41* [online]. [cit. 1. 11. 2011]. Dostupné z: <[http://shop.snailinstruments.com/index.php?main\\_page=product\\_info&products\\_id=664](http://shop.snailinstruments.com/index.php?main_page=product_info&products_id=664)>.
- [15] STMicroelectronics *LIS3LV02DQ* [online datasheet]. [cit. 26. 11. 2011]. Dostupné z: <[http://www.st.com/internet/com/TECHNICAL\\_RESOURCES/TECHNICAL\\_LITERATURE/DATASHEET/CD00047926.pdf](http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00047926.pdf)>.
- [16] VÁŇA, V. *Mikrokontroléry ATMEL AVR - programování v jazyce C*. Odpovědná redaktorka Iveta Kubicová. 2. dotisk 1. vydání. Praha: BEN - technická literatura, 2009. 216 s. ISBN 80-7300-102-0.
- [17] *vlastikd* [online]. [cit. 27. 10. 2011]. Jak funguje modelářské servo. Dostupné z: <<http://vlastikd.webz.cz/bastl/serva.htm>>.

# SEZNAM PŘÍLOH

<b>A Přílohy</b>	<b>55</b>
A.1 Obsah přiloženého CD . . . . .	55
A.2 Návod na oživení celé práce . . . . .	55
A.3 USART . . . . .	56
A.4 I <sup>2</sup> C . . . . .	57
A.5 SPI . . . . .	58
A.6 Bluetooth . . . . .	59
A.7 Akcelerometr . . . . .	60
A.8 Schémata . . . . .	61
A.9 Desky plošných spojů . . . . .	63

## A PŘÍLOHY

Následující informace doplňují vždy teorii mé práce.

### A.1 Obsah přiloženého CD

Na přiloženém CD jsou uloženy kromě elektronické verze mé práce také všechny zdrojové soubory a návrhy plošných spojů vyrobených desek.

Popis jednotlivých příloh na CD:

- **desky** - v této složce jsou uloženy návrhy vytvořených desek plošných spojů v programu Eagle
- **firmware** - složka obsahuje zdrojové soubory programu pro mikrokontrolér Atmel napsané v jazyce C pro mikrokontroléry AVR
- **ovladače** - složka s ovladači pro převodník USB/UART
- **práce** - ve složce je uložena elektronická verze mé práce ve formátu PDF
- **software** - do této složky jsem uložil zdrojové soubory pro aplikační software, který je napsaný ve Visual Basicu 6

### A.2 Návod na oživení celé práce

Aby bylo možné celou moji práci rozjet, je potřeba provést několik kroků. Nejprve připojit joystick a bluetooth modul k počítači přes USB a nainstalovat patřičné ovladače. V případě Windows 7 budou ovladače nainstalovány automaticky. V nižších verzích systému Windows je potřeba ovladače nainstalovat z přiloženého CD nebo stáhnout volně z internetu od firmy FTDI. Poté je ještě potřeba zkopírovat dvě knihovny ze složky **software/knihovny/**, která je uložena na CD do složky **C:/Windows/systém32/** v počítači. Nyní už je možné spustit aplikační software **RCauto.exe**.

Jestliže se všechny předchozí kroky povedly bez chyby, pak už stačí pouze zapnout vozidlo a připojit se k němu přes bluetooth pomocí spuštěné aplikace na počítači.

## A.3 USART

Tab. A.1: Popis jednotlivých bitů registrů *UCSRA*, *UCSRB* a *UCSRC*. [8] [9] [1]

<b>BIT</b>	<i>UCSRA</i>	
7	<i>RXC</i>	příjem kompletní
6	<i>TXC</i>	vysílání kompletní
5	<i>UDRE</i>	prázdný buffer
4	<i>FE</i>	chyba rámce
3	<i>DOR</i>	přetečení bufferu
2	<i>PE</i>	chyba parity
1	<i>U2X</i>	dvojnásobná přenosová rychlost
0	<i>MPCM</i>	multiprocessorový komunikační režim
<b>BIT</b>	<i>UCSRB</i>	
7	<i>RXCIE</i>	povolení přerušení při příjmu
6	<i>TXCIE</i>	povolení přerušení při odvysílání
5	<i>UDRIE</i>	povolení přerušení při prázdném bufferu
4	<i>RXEN</i>	povolení příjmu
3	<i>TXEN</i>	povolení vysílání
2	<i>UCSZ2</i>	určuje počet bitů (tabulka A.3)
1	<i>RXB8</i>	9. přijatý bit
0	<i>TXB8</i>	9. odvysílaný bit
<b>BIT</b>	<i>UCSRC</i>	
7	<i>URSEL</i>	určuje přístup k registrům <i>UBRRH</i> a <i>UCSRC</i>
6	<i>UMSEL</i>	výběr mezi synchronním a asynchronním režimem
5	<i>UPM1</i>	parita (tabulka A.2)
4	<i>UPM0</i>	parita (tabulka A.2)
3	<i>USBS</i>	počet stop bitů
2	<i>UCSZ1</i>	určuje počet bitů (tabulka A.3)
1	<i>UCSZ0</i>	určuje počet bitů (tabulka A.3)
0	<i>UCPOL</i>	polarita dat synchronního režimu

Tab. A.2: Nastavení parity. [8] [9] [1]

<i>UPM1</i>	<i>UPM0</i>	<b>Režim</b>
0	0	deaktivováno
0	1	vyhrazeno
1	0	sudá parita
1	1	lichá parita

Tab. A.3: Nastavení počtu bitů rámce. [8] [9] [1]

<i>UCSZ2</i>	<i>UCSZ1</i>	<i>UCSZ0</i>	<b>Počet bitů</b>
0	0	0	5
0	0	1	6
0	1	0	7
0	1	1	8
1	0	0	vyhrazeno
1	0	1	vyhrazeno
1	1	0	vyhrazeno
1	1	1	9

## A.4 I<sup>2</sup>C

Tab. A.4: Popis jednotlivých bitů registrů *UCSRA*, *UCSRB* a *UCSRC*. [8] [9] [1]

<b>BIT</b>	<i>TWCR</i>	
7	<i>TWINT</i>	příznak přerušení
6	<i>TWEA</i>	povoluje potvrzování ACK
5	<i>TWSTA</i>	generuje START stav
4	<i>TWSTO</i>	generuje STOP stav
3	<i>TWWC</i>	detekuje kolizi zápisu
2	<i>TWEN</i>	povuleje I <sup>2</sup> C
1		nepoužito
0	<i>TWIE</i>	povoluje přerušení sběrnice

Tab. A.5: Příklady nastavení registrů  $UBRRL$  a  $UBRRH$  při frekvenci krystalu 16 MHz. [8] [9] [1]

<b>Baud Rate (bps)</b>	$U2X = 0$ $UBRR$	$U2X = 1$ $UBRR$
2400	416	832
4800	207	416
9600	103	207
14 400	68	138
19 200	51	103
28 800	34	68
38 400	25	51
57 600	16	34
76 800	12	25
115 200	8	16
230 400	3	8
250 000	3	7

## A.5 SPI

Tab. A.6: Nastavení přenosové frekvence u SPI. [8] [9] [1]

$SPI2X$	$SPR1$	$SPR0$	<b>Frekvence na SCK</b>
0	0	0	$f_{osc}/4$
0	0	1	$f_{osc}/16$
0	1	0	$f_{osc}/64$
0	1	1	$f_{osc}/128$
1	0	0	$f_{osc}/2$
1	0	1	$f_{osc}/8$
1	1	0	$f_{osc}/32$
1	1	1	$f_{osc}/64$

## A.6 Bluetooth

Následující tabulka popisuje příkazy pro nastavení bluetooth modulu BTM-112.

Tab. A.7: AT příkazy pro nastavení modulu BTM-112. [12]

<b>Příkaz</b>	<b>Možnosti</b>	<b>Funkce</b>
AT<CR>		ověřuje funkci AT módu (vrací OK)
ATN=nazev<CR>		změní jméno zařízení (nazev je jméno)
ATP=1234<CR>	ATP0 - vypne pin kód	změní pin kód zařízení (v tomto případě na 1234)
ATI <sub>x</sub> <CR>	za x je možné zvolit: 0 - verze firmware 1 - info o aktuálním nastavení 2 - RSSI	vypíše info
ATL <sub>x</sub> <CR>	za x je možné zvolit: * - 1 200 bps # - 2 400 bps 0 - 4 800 bps 1 - 9 600 bps 2 - 19 200 bps 3 - 38 400 bps 4 - 56 000 bps 5 - 115 200 bps 6 - 230 400 bps 7 - 460 800 bps	změna přenosové rychlosti (Baud rate)
ATH<CR>		odpojí se (jen v případě, že je připojeno zařízení přes bluetooth)
ATC <sub>x</sub> <CR>	za x je možné zvolit: 0 - vypnuto 1 - zapnuto (default)	nastavení řízení toku (RTS/CTS)

## A.7 Akcelerometr

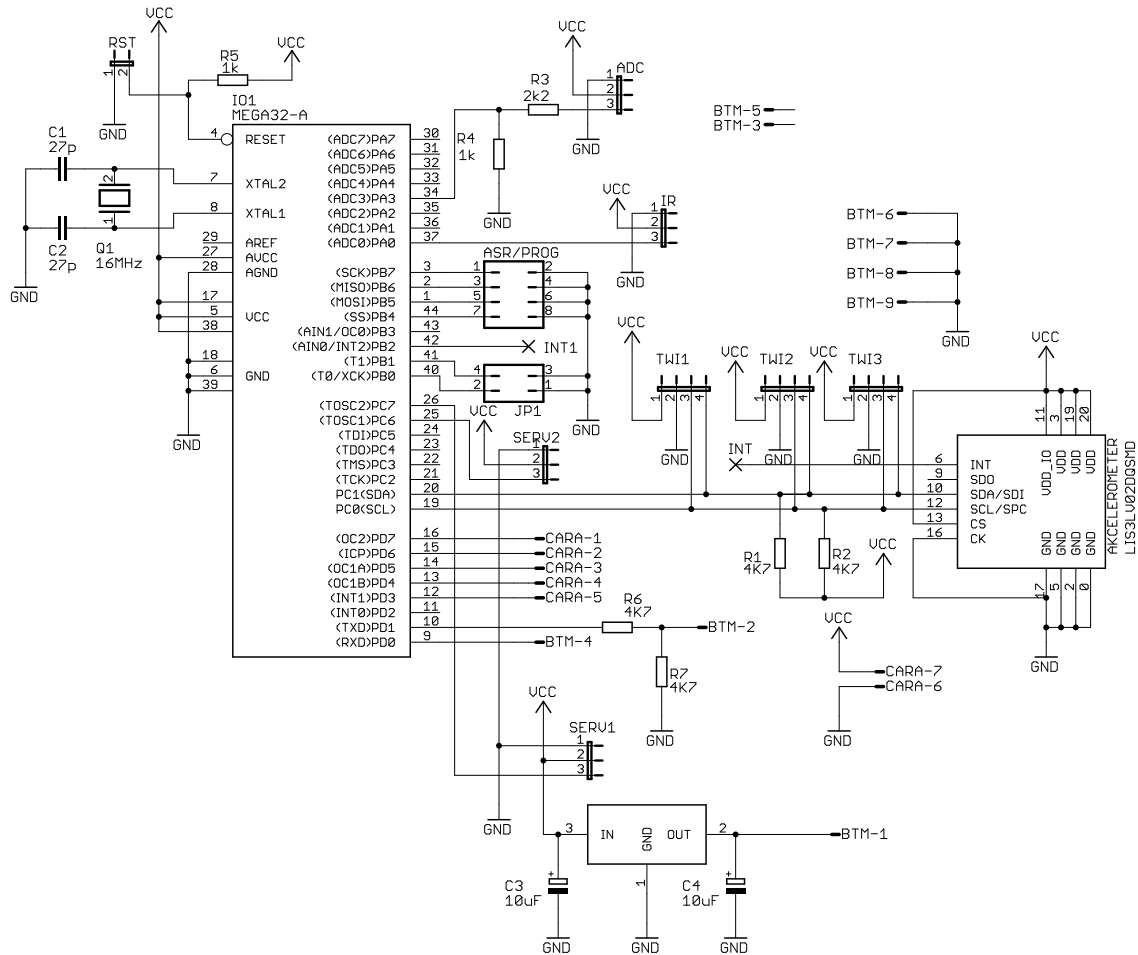
V následující tabulce jsou popsány nejdůležitější registry akcelerometru LIS3LV02DQ.

Tab. A.8: Popis vybraných registrů akcelerometru. [15]

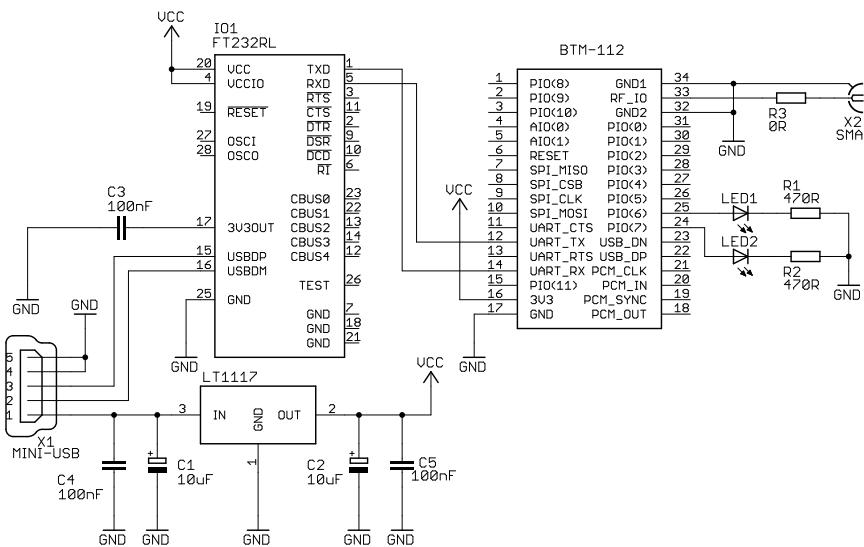
<b>Registr</b>	<b>Adresa (hex)</b>	<b>Popis</b>
WHO_AM_I	0F	Adresa zařízení (pouze ke čtení).
CTRL_REG1	20	Nastavení, inicializace a povolení některých nadstandardních možností.
CTRL_REG2	21	
CTRL_REG3	22	
OUTX_L	28	Zrychlení v ose x.
OUTX_H	29	
OUTY_L	2A	Zrychlení v ose y.
OUTY_H	2B	
OUTZ_L	2C	Zrychlení v ose z.
OUTZ_H	2D	

## A.8 Schémata

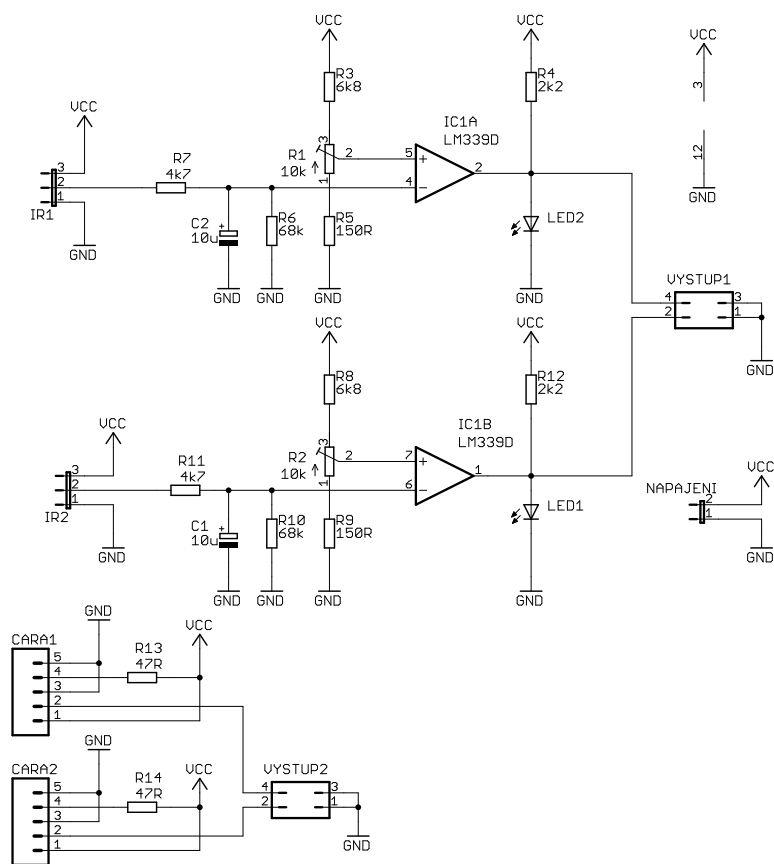
Zde uvádím schémata hotových zařízení. Desky plošných spojů k těmto schématům jsou v následující kapitole.



Obr. A.1: Schéma zapojení hotového zařízení ve vozidle



Obr. A.2: Schéma zapojení hotového zařízení připojeného k počítači

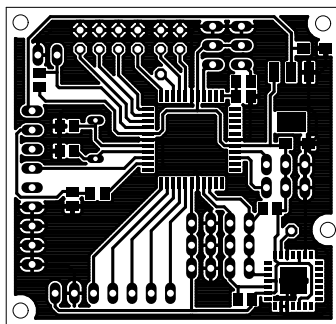


Obr. A.3: Schéma zapojení hotového komparátoru pro IR detektory překážek a pro zapojení senzorů bílé čáry

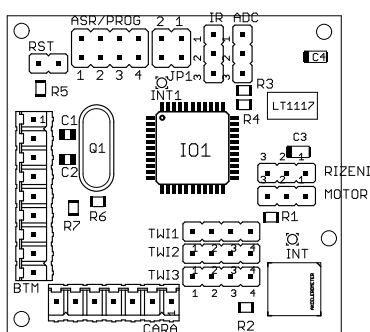
## A.9 Desky plošných spojů

Na následujících několika stranách se nacházejí desky plošných spojů hotových zařízení a modulů i s osazovacími plány.

Na obrázcích A.4 a A.5 je deska plošného spoje s osazovacím plánem pro modul ve vozidle.



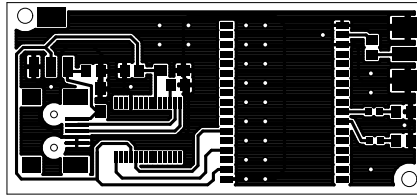
Obr. A.4: Deska plošného spoje hotového zařízení ve vozidle - vrstva TOP



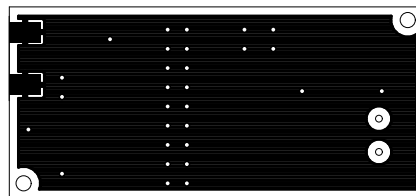
Obr. A.5: Osazovací plán desky plošného spoje hotového zařízení ve vozidle - vrstva TOP

Pro modul připojený k počítači je deska plošného spoje na obrázku A.6 a osazovací plány pro tuto desku jsou obrázcích A.7 a A.8.

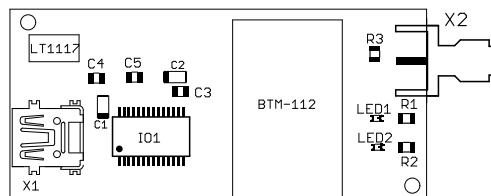
Na závěr ještě modul komparátoru. Deska plošného spoje je na obrázku A.9 a osazovací plán k této desce je na obrázku A.10.



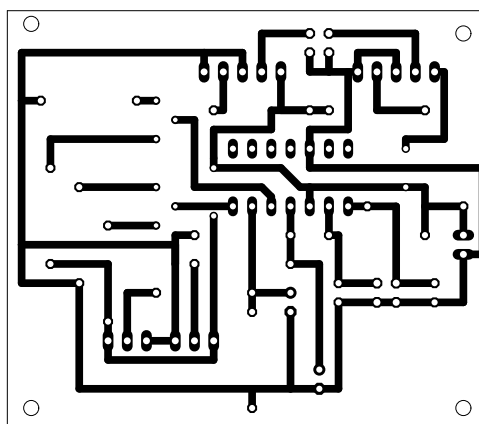
Obr. A.6: Deska plošného spoje hotového zařízení připojeného k počítači - vrstva TOP



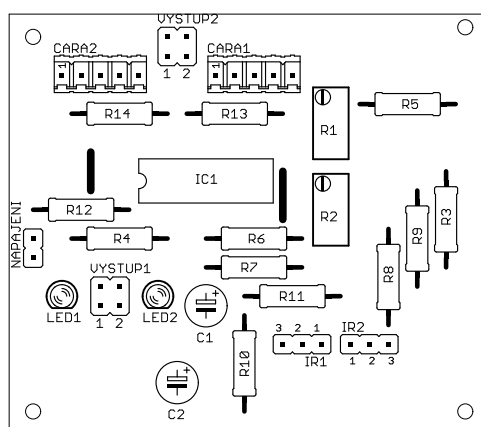
Obr. A.7: Deska plošného spoje hotového zařízení připojeného k počítači - vrstva BOTTOM



Obr. A.8: Osazovací plán desky plošného spoje hotového zařízení připojeného k počítači - vrstva TOP



Obr. A.9: Deska plošného spoje komparátoru - vrstva BOTTOM



Obr. A.10: Osazovací plán desky komparátoru - vrstva TOP