



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV MECHANIKY TĚLES, MECHATRONIKY
A BIOMECHANIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND
BIOMECHANICS

ŘÍDICÍ JEDNOTKA PRO AUTOMOBILOVÉ PARKOVACÍ SENZORY

CONTROL UNIT FOR AUTOMOTIVE PARKING SENSORS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JIŘÍ OTÁHAL

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JOSEF VEJLUPEK

BRNO 2014

Abstrakt

Tato práce se zabývá programováním řídicí jednotky pro automobilové parkovací senzory v prostředí Matlab Simulink, návrhem schématu jednotky v programu Eagle na základně předešle verze, její výrobou a osazením.

Abstract

This thesis focuses on programing a control unit for automotive ultrasonic sensors in Matlab Simulink environment. It designs the schematic of the control unit based on the previous version using Eagle, and realizes this design.

Klíčová slova

Ultrazvukové senzory, řídicí jednotka, detekce překážky, piezoelektrický jev, parkovací asistent

Keywords

Ultrasonic sensors, control unit, obstacle detection, piezoelectric effect, parking assistance system

Bibliografická citace

OTÁHAL, J. *Řídicí jednotka pro automobilové parkovací senzory*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2014. 35 s. Vedoucí bakalářské práce Ing. Josef Vejlupek.

Čestné prohlášení

Prohlašuji, že jsem bakalářskou práci na téma „Řídící jednotka pro automobilové parkovací senzory“ vypracoval a napsal samostatně s použitím odborné literatury uvedené v seznamu.

6. Červen 2011

.....
Otáhal Jiří

Poděkování

Rád bych poděkoval panu Ing. Josefu Vejlupkovi, který mi i přes jeho časovou vytíženost poskytl cenné připomínky a rady při vypracování méj bakalářské práce.

Obsah

1. Úvod	7
2. Rešerše	8
2.1. Využití ultrazvukových senzorů – parkovací asistent Valeo	8
2.2. Popis funkce ultrazvukových senzorů	8
2.2. Výhody a nevýhody ultrazvukových senzorů	9
2.3. Námi použité senzory.....	11
2.4. Jiné typy senzorů a jejich srovnání	12
3. Návrh elektroniky	14
3.1. Požadavky při návrhu elektroniky na míru autíčku	14
3.2. Popis navržené elektroniky	15
3.2.1. Připojení senzorů	15
3.2.2. Použitý microcontroller	16
3.2.3. Napájení desky ze zdroje	16
3.2.4. CAN Transceiver	17
3.3. Návrh desky plošných spojů	18
3.4. Testování funkčnosti hotové řídicí jednotky	19
4. Komunikace senzorů s řídicí jednotkou a Car4	20
4.1. Použitý software.....	20
4.1.1. Ukázka bloků v simulinku:	20
4.1.2. Ukázka programu v MPLAB X IDE:	21
4.2. Komunikace senzorů s řídicí jednotkou	22
4.1. Komunikace řídicí jednotky s Car4.....	23
4.2. Zpracování dat ze senzoru a jejich vyhodnocení	23
4.3. Testování přesnosti senzorů a rychlosti komunikace.....	25

5. Model překážky – zabránění kolize	27
5.1. Rozmístění senzorů na Car4.....	27
5.2. Vytvoření modelu překážky	28
5.3. Vyhodnocení polohy překážky	30
5.4. Implementace algoritmu.....	31
5.5. Testování námi vytvořeného algoritmu	31
5.6. Možná rozšíření modelu.....	32
6. Závěr	34
Seznam použitých zdrojů	35

1. Úvod

Ultrazvukové senzory mají v dnešní době široké využití a to hlavně v automobilovém průmyslu. Společně s řídicí jednotkou zvyšují při malých cenových nákladech bezpečnost a komfort automobilu. Bezpečnost je zde zvýšena v podobě parkovacího asistenta, který může včas varovat řidiče před následující kolizí. Zvýšený komfort pak může tvořit např. automatický systém parkování, kdy řidič může bezpečně z automobilu vystoupit a přes mobil sledovat a kontrolovat stav parkování. Ultrazvukové senzory navíc nejsou příliš drahé a jejich montáž je jednoduchá. Všechny tyto vlastnosti jsou důvodem, proč jsou v automobilovém průmyslu tyto senzory tak hojně využívány.

Má práce se zabývá realizací řídicí jednotky parkovacího asistenta určeného pro mobilní robot Car4. V první kapitole je nejprve popsána funkce ultrazvukových senzorů, jejich výhody a nevýhody a srovnání s jiným typem senzorů. Další kapitoly se již zabývají návrhem a následnou realizací řídicí jednotky, její naprogramování a následné testování.

2. Rešerše

2.1. Využití ultrazvukových senzorů – parkovací asistent Valeo

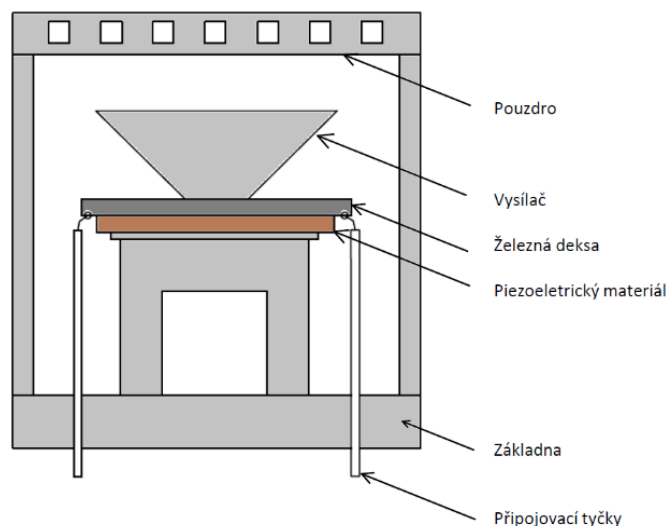
Na obr. 1 je zobrazen koncept automatického parkovacího systému „PARK4U“ od firmy Valeo. Tento systém je schopen najít parkovací místo při rychlosti auta až 30km/h. Po nalezení místa může řidič vystoupit a nechat auto automaticky zaparkovat, přičemž může použít aplikaci ve smartphonu pro výběr paralelního nebo podélného parkování a zároveň může vozidlo kdykoliv zabrzdít. Podobný koncept by se mohl nacházet i na autě Car4, kde ultrazvukové senzory by sledovaly okolní prostředí a vytvářely by jeho přibližný model. Po nalezení vhodného místa pro parkování, by se spustil algoritmus pro bezpečné zaparkování Car4 a zároveň by senzory neustále sledovaly vzdálenost auta od překážek, aby v případě hrozící kolize mohla řídicí jednotka zavčas auto zastavit. Celý tento systém by se navíc dal ovládat pomocí dálkového ovladače.



Obr. 1. Parkovací asistent Valeo

2.2. Popis funkce ultrazvukových senzorů

Ultrazvukové senzory pracují na principu generování zvukových vln o vysoké frekvenci (nad 20 000Hz), které se po odražení od překážky vrátí zpět do senzoru. Pro generování zvukových vln při měření vzdálenosti ve vzduchu se většinou používá piezoelektrického jevu. Tento efekt je způsoben piezoelektrickým krystalem, který má schopnost se při přivedení napětí deformovat a naopak při jeho deformování dojde ke vzniku napětí na jeho povrchu. Pokud tedy na krystal obsažený v senzoru přivedeme napětí, dojde k jeho smrštění a vytvoření vibrací a tím i zvukových vln. Po odražení od překážky se vlny vrátí zpět do snímače, a jak již bylo uvedeno, při aplikování síly krystal generuje odpovídající napětí, což zachytí elektronika v senzoru. Pro šíření ultrazvukových vln navíc potřebujeme i vhodné prostředí. Senzory lze použít ve vzduchu nebo kapalině, ale nelze je použít např. ve vakuu. Ukázka vnitřku senzoru je na obr. 2.



Obr. 2. Náčrtek vnitřku senzoru

2.2. Výhody a nevýhody ultrazvukových senzorů

Jelikož jsou ultrazvukové senzory používány v relativně náročných podmínkách (např. továrny, nárazníky automobilů, nádrže atd.), musí být také dostatečně odolné. Navíc by komunikace mezi senzory a jejich řídicí jednotkou měla být co nejjednodušší a zároveň co nejspolehlivější. Z těchto důvodů mají senzory tyto vlastnosti:

- Vodě odolné /vodotěsné – Samotný senzor, krystal a elektronika jsou uzavřeny v plastovém pouzdře. Konektor pro spojení senzorů s řídicí jednotkou obsahuje zámek a je ohraničen těsněním.
- Tepelně odolné – Senzory jsou většinou navrhovány a testovány tak, aby rozmezí jejich bezchybného fungování bylo mezi -40°C až $+90^{\circ}\text{C}$.
- Nárazu odolné – Senzory neobsahují žádné mechanické části, jejich kryt tvoří nejčastěji plast a na namáhané části je upevněn gumový kroužek.
- Přesné – Pokud se jedná o senzor používaný například v průmyslu, kde je poloha senzoru fixována a měří pouze jeden typ překážky, může být jeho přesnost v rozmezí $\pm 1\text{mm}$. Senzory v automobilech mají přesnost horší (několik centimetrů) a to z důvodu, že tvar i typ odráženého materiálu je různý a auto, tedy i senzor, se během detekce pohybují. V automobilech však přesnost v rozmezí několika centimetrů naprosto postačuje.
- Spolehlivé – Senzory pracují stejně, i když se jedná o tekutý, sypký nebo pevný měřený povrch. Senzor je navíc schopný bezchybně pracovat za deště, sněžení nebo v zaprášeném prostoru.

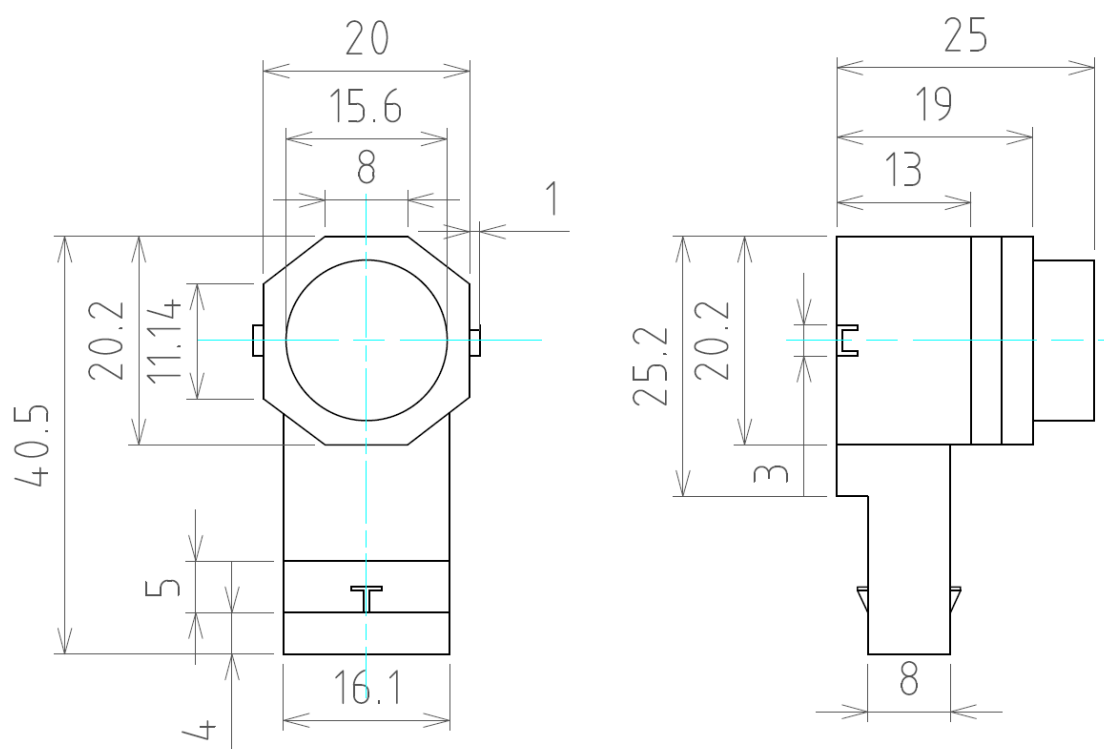
- Bezúdržbové – Jelikož nedochází k doteku senzoru s detekovaným objektem, není zde žádné tření a opotřebování. Senzory mají navíc malou spotřebu, přibližně 20mA.

Samozřejmě každé odvětví, kde se ultrazvukové senzory používají, mají jiné nároky na jejich vlastnosti. Jelikož senzory pracují se zvukovými vlnami, je jejich limitace hlavně v tomto ohledu. Zvukové vlny se mohou libovolně odrážet od překážek, některé materiály mají schopnost zvukové vlny pohlcovat a navíc tvar překážek může způsobit naprosté ztracení vlny. Podmínky způsobující omezení funkce senzorů jsou:

- Teplota vzduchu – Teplota ovlivňuje rychlost zvuku, tudíž senzor bude podávat při změně teploty různé výsledky. Tento fakt lze odstranit přídavným teploměrem, který bude měřit teplotu a posílat ji do řídicí jednotky, která vypočítá skutečnou rychlost zvuku.
- Tvar překážky – Problémem pro detekci jsou například zaoblené objekty nebo šikmé překážky. Zvuková vlna se může odrazit špatným směrem a senzor ji pak nemůže zachytit. Dalším problémem je, že zvukové vlny se mohou odrazit několikrát od objektů a potom se vrátit zpět do senzoru, který však špatně vyhodnotí vzdálenost předmětu.
- Materiál povrchu – Problém při detekci může nastat, pokud je detekovaným objektem například kapalina, na které je pěna. Zvuková vlna se není schopna odrazit od pěny a senzor ztrácí svoji funkci.
- Vzdálenost překážky – Senzor není schopný detekovat velmi blízké objekty (většinou pod 20cm). Další problém může nastat při detekování velmi vzdálených objektů (u běžných senzorů do vzdálenosti 3m), kdy senzor již není schopen detekovat odraženou zvukovou vlnu.
- Poloha překážky – Vysílaná vlna má kuželovitý tvar, což znamená, že senzor má problémy s detekcí překážek pod úhlem větším než 70° na jednu stranu. Tento fakt většinou není příliš důležitý, protože například na automobilech je většinou více než 4 senzory na každé straně, což spolehlivě tento efekt odstraní.

2.3.Námi použité senzory

Pro naši práci jsme použili senzory 4H0 919 275 od firmy Valeo Germany. Tyto senzory se hojně používají v automobilech např. ve firmě Škoda (modely Škoda Octavia, Fabia, Yeti, Roomster, Superb), Audi (modely Audi A3, A4, RS4, A5, A6, RS6, A8, TT) a Volkswagen (modely VW Golf 6, Touran, Touran aj.). Rozměry senzoru můžeme vidět na obr. 3, který byl vytvořen pomocí výukové verze programu Autocad. Některé rozměry nejsou okótovány (drážky pro upnutí senzoru do konektoru), jelikož jejich rozměr není pro tuto ukázkou příliš podstatný.



Obr. 3. Rozměry ultrazvukového senzoru „4h0 919 275“



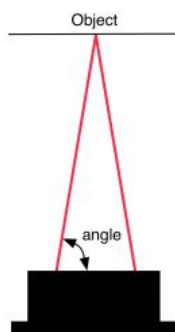
Obr: 4. Konektor a ultrazvukový senzor „4h0 919 275“

2.4. Jiné typy senzorů a jejich srovnání

Ultrazvukové senzory mají mnoho limitací a to např. malou měřicí vzdálenost (i u speciálních senzorů není možné měřit objekty vzdálené 8m a více) a naopak není možné měřit vzdálenost menší než cca. 20 cm. Další problém je, že ultrazvukový senzor vysílá velmi široký paprsek, tudíž jeho přesnost by v některých aplikacích nemusela stačit (měření průhybu, vzdálenosti elektronických součástek apod.) V automobilovém průmyslu se převážně používají ultrazvukové senzory, protože se zde nepracuje s velkými vzdálenostmi a velkou přesností, ale například v průmyslu nebo stavebnictví je potřebné použít jiný typ senzorů.

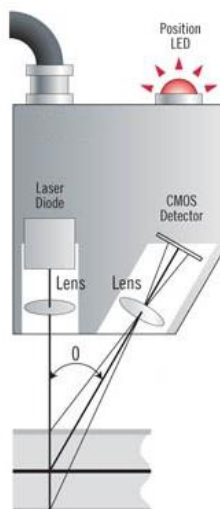
Jiné senzory použitelné pro měření vzdálenosti jsou:

- IR senzory – Pro jejich činnost je potřebná IR dioda, která nejprve vyšle na měřenou překážku infračervený signál, který je po odražení snímán CMOS senzorem. Ten zaznamená úhel dopadu, z čehož se vypočítá vzdálenost překážky. Princip je zobrazen na obrázku 5. Oproti ultrazvukovým senzorům má paprsek menší rozptyl a signál je mnohem rychlejší, ale jeho dosah je menší (např. senzor Sharp GP2Y0A21YK má maximální dosah 80cm).



Obr: 5. Infračervený senzor

- Laserové senzory – Stejně jako ultrazvukové senzory se jedná o bezkontaktní způsob měření vzdálenosti. Senzory však mají mnohem větší dosah měření, proto je jejich využití hlavně ve stavebnictví a průmyslu. Paprsek navíc nemá kuželovitý tvar, ale je sdružený do úzkého svazku, měření je tedy přesnější. V průmyslu se využívají např. pro měření výšky, šířky a délky součástí, kontrole desek plošných spojů, měření profilů, výchylky nebo průhybu aj. Na automobilech se senzory nepoužívají, protože by detekovaly pouze velmi malou oblast, jelikož je jejich paprsek sdružený do jednoho svazku. Navíc by jejich cena byla vyšší. Na obr. 6 je zobrazen popis funkce laserového senzoru používaného v průmyslu. Laserová dioda vyšle světelný paprsek, který se odrazí od povrchu a projde čočkou do detektoru.



Obr: 6. Princip fungování laserových senzorů

3. Návrh elektroniky

3.1. Požadavky při návrhu elektroniky na míru autíčku

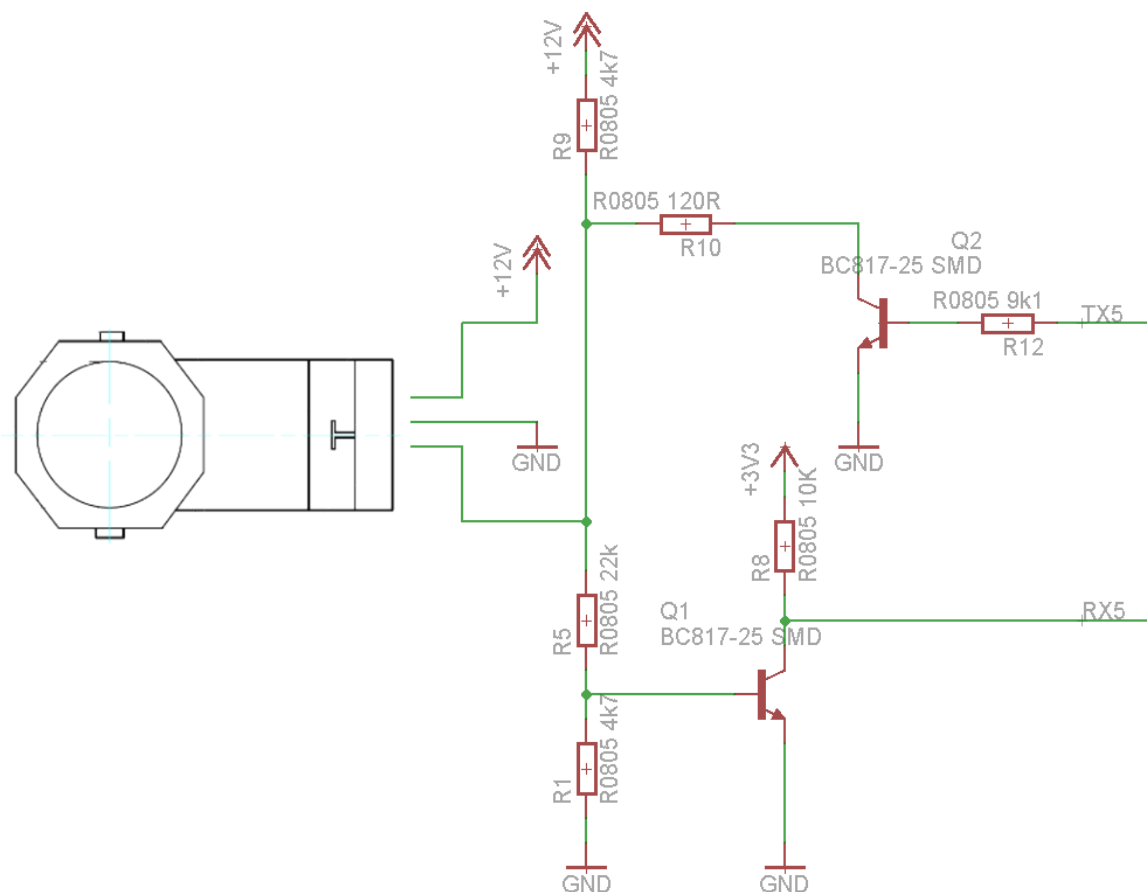
Při návrhu elektroniky jsme vycházeli ze staré verze řídicí jednotky Škoda parking-hil 12CH. Požadavky na úpravu desky byly:

- Změna microcontrolleru – Jako microcontroller pro naši řídicí jednotku byl použit typ DSPIC33FJ128MC708/PT od firmy Microchip.
- Optimalizace desky – Pro potřeby Car4 musely být upraveny rozměry původní desky na co nejmenší. Po úpravách jsou tedy konečné rozměry 70x60mm, což je i přes omezené místo na car4 dostačující.
- Přidání budiče pro sběrnici CAN – Zajišťuje sériovou komunikaci CAN mezi řídicí jednotkou a PC na CAR4.
- Rozdělení napájení – Jelikož car4 má vlastní napájecí zdroj pro jeho elektroniku, byly na řídicí jednotce přidány Wago svorky, barevně od sebe rozlišené, pro všechny potřebné napětí. Senzory pro svůj chod potřebují napětí 12V, přídavný obvod a microcontroller 3,3V a napájení servomotorů 6V.
- Přidání zenerových diod – Jako ochrana proti případným špičkám zdroje autíčka byly na desku pro všechny napětí přidány zenerovy diody, které tyto výkyvy odstraní. Byla zde také použita ochrana proti přepólování pomocí mosfet tranzistoru.
- Přidání konektorů pro servomotory – Jelikož navržená řídicí jednotka má výstupy pro 8 senzorů, z čehož pouze 6 senzorů bylo využito na detekci překážky a zabránění kolize, budou dva senzory připevněny na otočném servomotoru. Tímto rozložením můžeme tyto 2 senzory libovolně otáčet v rozmezí 360°, což může být přínosem např. při algoritmu hledání parkovacího místa nebo pro skenování překážek v okolí autíčka.
- Přidání signalizačních LED – Na několik nevyužitých pinů microcontrolleru byly připojeny LED diody, které taky slouží jako signalizace např. při běhu programu, testování jednotky aj.

3.2. Popis navržené elektroniky

3.2.1. Připojení senzorů

Jelikož má senzor pro signál (vstup a výstup) určený pouze jeden pin a na microcontroller je potřebné zapojit vstup Rx a výstup Tx zvlášť. Kde výstup Tx je potřeba zesílit a naopak výstup je potřeba upravit na napětí microcontrolleru.



Obr. 7. Připojení senzoru k microcontrolleru

3.2.2. Použitý microcontroller

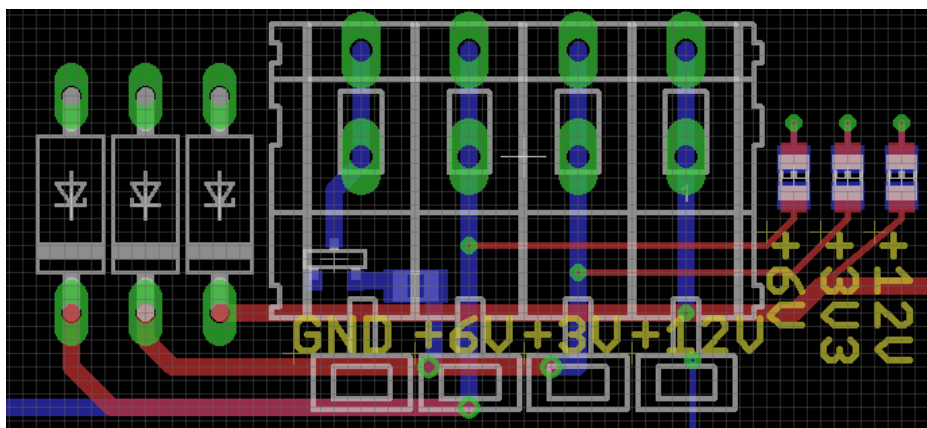
Jako microcontroller pro naši desku byl použit 16-bitový Microchip DSPIC33FJ128MC708/PT ze série dsPIC33F. Při jeho výběru jsme dbali na to, aby měl dostatek pinů pro jednotlivé senzory, protože každý pin senzoru potřebuje dva piny na microcontrolleru. Dále jsme potřebovali několik PWM pinů pro řízení servomotorů. Starší verze tohoto microcontrolleru byla navíc použita na původní verzi řídicí jednotky Škoda parking-hil 12CH, kde spolehlivě pracovala.



Obr. 8. Microcontroller DSPIC33FJ128MC708/PT

3.2.3. Napájení desky ze zdroje

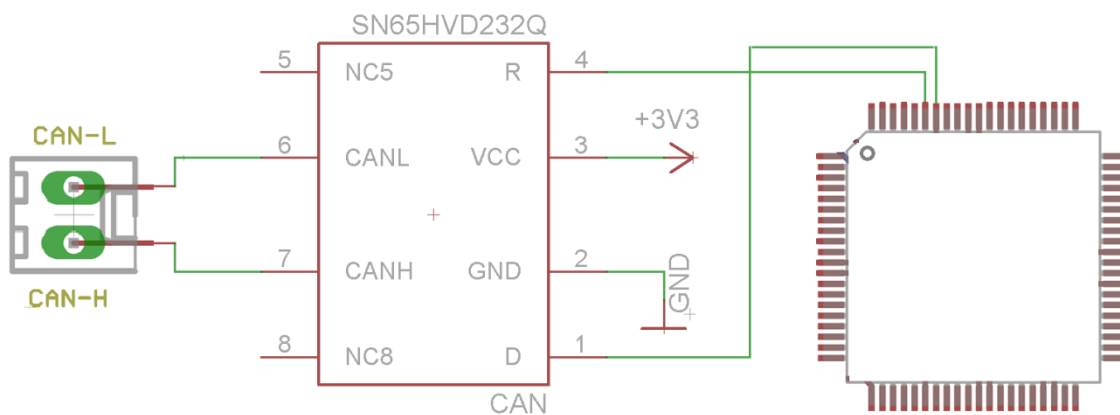
Jelikož autíčko CAR4 má vlastní napájecí zdroj, který všechna naše potřebná napětí zajišťuje, byly pro jeho propojení s naší řídicí jednotku přidělaný pouze Wago svorky. Proti případným špičkám zdroje jsme jako ochranu přidělali pro všechna napětí zenerovy diody. Pro napětí 12V byla použita dioda 1N5350B, pro 3,3V dioda 1N5335B a pro 6V dioda 1N5341B. U wago svorky GND lze vidět ochranu proti přepólování, kterou tvoří tranzistor MOSFET TSM2302CX RF s odporem R0805 10k. Ochrana je napojena pouze na 3.3V, jelikož na tomto napětí je napojen samotný procesor a při nechtěném připojení většího napětí by se mohl poškodit. Pro každé napětí je navíc přidána smd dioda pro signalizaci chodu.



Obr. 9. Napájení desky

3.2.4. CAN Transceiver

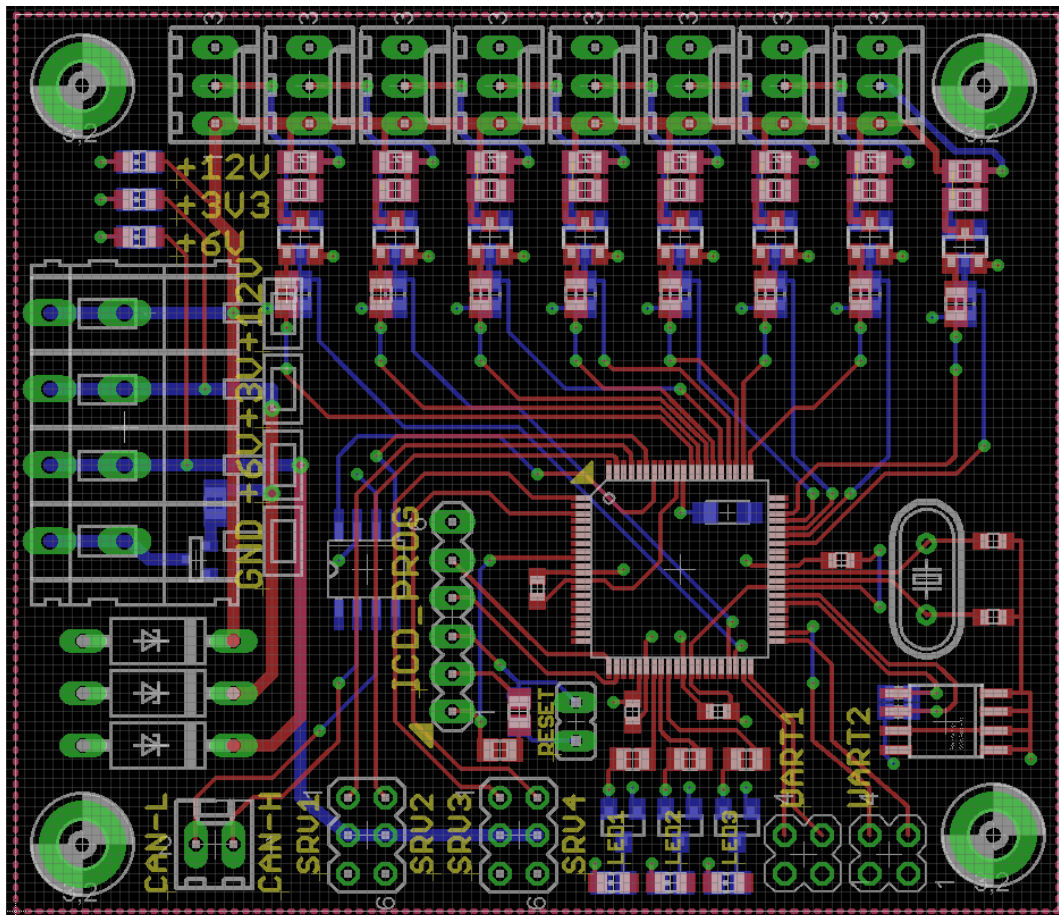
Pro komunikaci s PC umístěného na CAR4 jsme zvolili sběrnici CAN. Pro zajištění této komunikace s námi zvoleným microcontrollerem byl použit CAN transceiver SN65HVD232Q. Jeho zapojení je uvedeno na obr. 10



Obr: 10. Zapojení CAN Transceiveru

3.3. Návrh desky plošných spojů

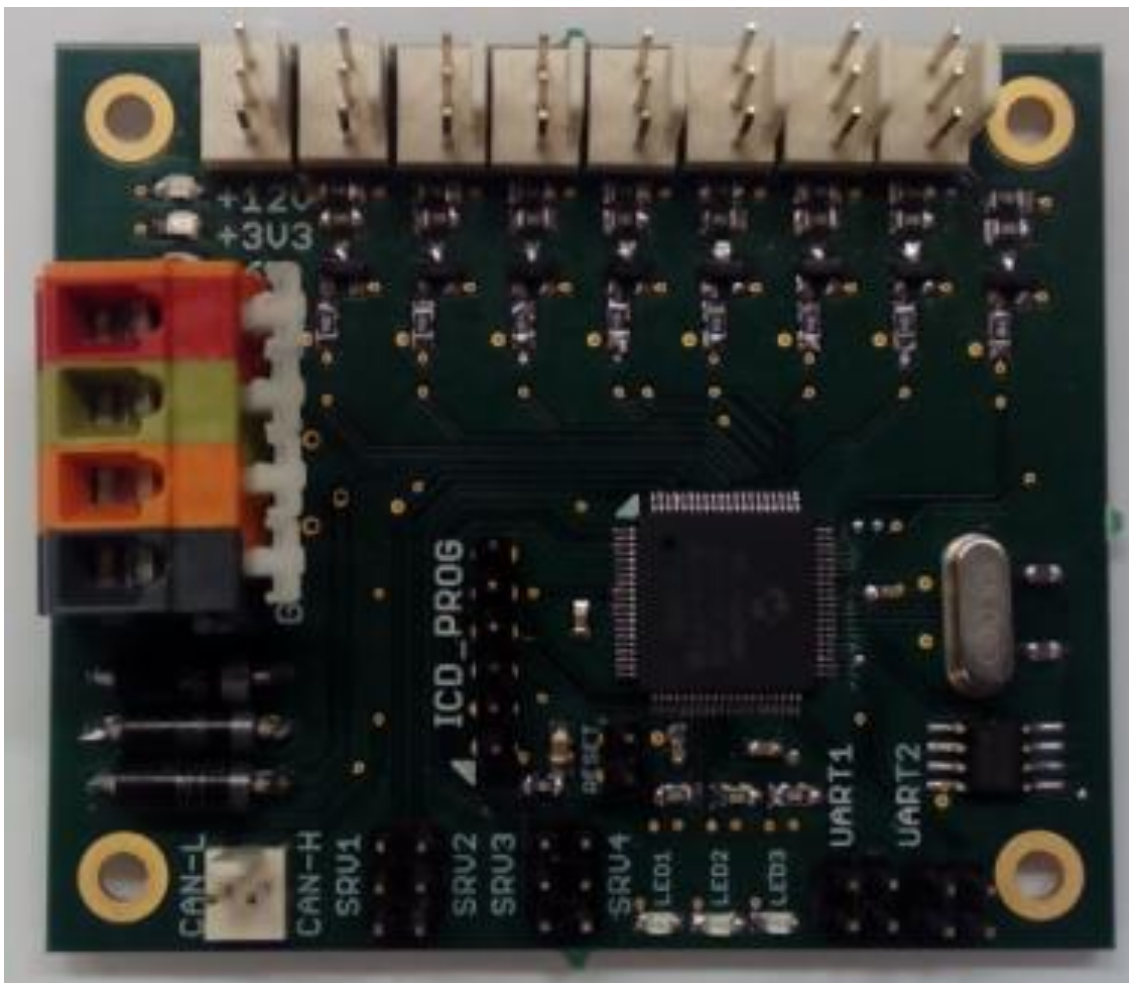
Pro návrh desky byl použit program Eagle 6.5.0. Seznam všech použitých součástek je uveden v příloze.



Obr: 11. Navržená deska v programu Eagle

Jednotlivé rozmístění konektorů:

- Horní strana desky:
 - a. Konektory sloužící pro připojení 8 senzorů
- Spodní strana desky:
 - a. Konektor CAN-L/H – Slouží pro spojení CAN desky s Car4 resp.
 - b. Konektor SRV1...4 – Konektory sloužící pro připojení servomotorů
 - c. Konektor UART1...2 – Slouží k propojení počítače s řídicí jednotkou pomocí rozhraní UART.



Obr: 12. Ukázka vyrobené desky

3.4. Testování funkčnosti hotové řídicí jednotky

Při testování funkčnosti jsme postupovali podle následujících kroků:

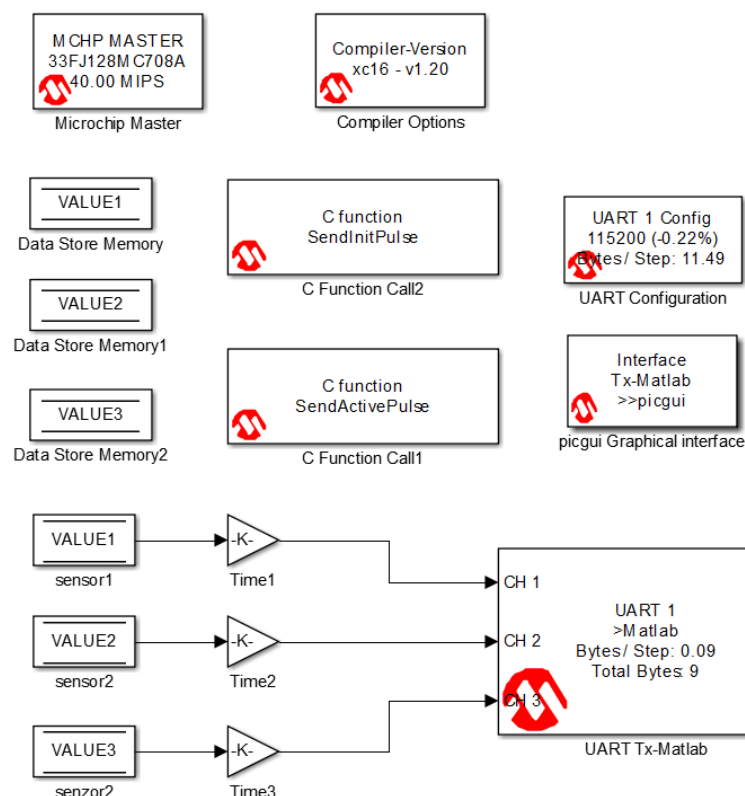
- Nejprve jsme pomocí mikroskopu zkontrolovali všechny pájené spoje a to, jestli se mezi sebou součástky nedotýkají (např. nožičky mikrokontroléru) nebo jestli jsou správně usazeny na plošky desky.
- Dále byly pomocí multimetru „prosvíceny“ všechny cesty a kontrolovala se jejich průchodnost.
- Po přivedení napětí na desku, se naskytlo několik problémů. Většinou se jednalo o studené spoje, špatně zapájené součástky nebo přehozené hodnoty odporů.
- Po opravení všech výše uvedených problémů jsme již do desky nahráli jednoduchý testovací program na rozsvícení signalizačních led a následně i program pro ultrazvukové senzory.

4. Komunikace senzorů s řídicí jednotkou a Car4

4.1. Použitý software

Při vytváření programu pro senzory jsme použili program Matlab Simulink a MPLAB X IDE. V simulinku byl vytvořen nejprve základ pro celou komunikaci, tedy nadefinování proměnných, verzi čipu, UARTU apod. Nemohly jsme však zde použít bloky digital output a digital input pro posílání a následnou detekci signálu ze senzoru. Důvodem je to, že toto řešení by bylo pro naše účely příliš pomalé, jelikož se zde pracuje s časy v desítkách μs . Museli jsme proto výsledný kód ze simulinku ještě upravit v programu MPLAB X IDE, kde bylo pak posílání a detekování pulsů přesně nadefinováno.

4.1.1. Ukázka bloků v simulinku:

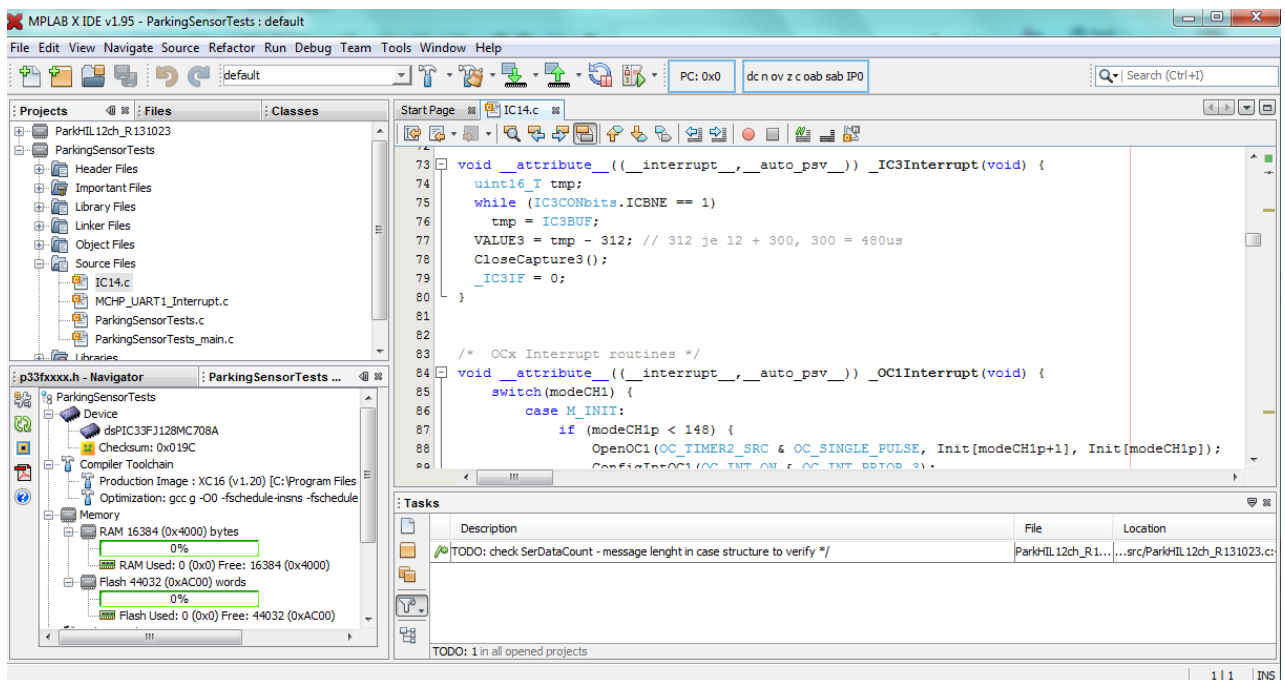


Obr: 13. Bloky v Simulinku

- Microchip Master – Nadeřinování řipu, s kterým budeme pracovat
- Compiler Options – Vřbřer verze řekladaře
- Data Store Memory – Vytvoření proměnných pro data ze senzorů
- C Function Call – Vlastní funkce napsaná v jazyku C.
- UART Configuration – Nakonfigurování rozhraní UART
- Picgui Graphical interface – Slouží později pro vykreslení dat z UARTU.
- Senzor – Zapsaná data ze senzorů
- Time – Převod tiků microcontrolleru na čas
- Uart Tx-Matlab – Poslání dat pomocí UARTU na desce

4.1.2. Ukázka programu v MPLAB X IDE:

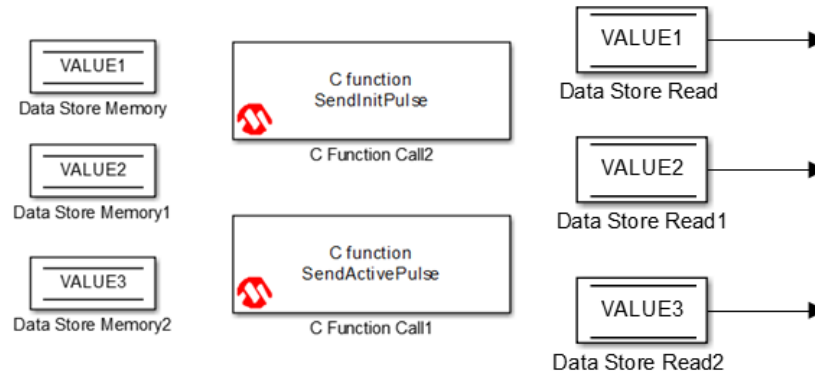
Pro vytváření funkce jsme používali verzi MPLAB X IDE v.1.95. Ukázka programu a kódu je na obr. 14.



Obr. 14. Ukázka programu MPLAB

4.2. Komunikace senzorů s řídicí jednotkou

Jak již bylo řečeno, při komunikaci senzorů s řídicí jednotkou jsme nemohli použít bloky v simulinku a museli jsme posílání a detekci pulsů naprogramovat v programu MPLAB. V simulinku jsme si nejprve vytvořili všechny základní bloky a dále bloky:



Obr: 15. Používané bloky

- Data Store Memory - Jedná se o paměťové místo, do kterého můžou všechny ostatní bloky, obsažené ve stejném systému a mající stejné jméno ukládané hodnoty, zapisovat nebo číst data. Hodnoty jsou ukládány v datovém formátu uint16. Tento blok vytvoří z pohledu programu globální proměnou, do které dále ukládáme dobu odezvy ultrazvukového snímače
- Data Store Read- Načítá data z Data Store Memory. Název dat VALUE1, VALUE2, VALUE3, musí být v obou případech stejný. Tento blok má již v simulinku svůj vlastní výstup a můžeme jej připojit na ostatní bloky a tak číst data.
- C function call1,2 – Jedná se o uživatelem definovatelné bloky, které jsou napsány v jazyce C.

V programu MPLAB X jsme potom postupovali:

1. Napsání funkce pro Output Compare OC:

- Vygenerování reset pulsu – SendResetPulse
- Vygenerování inicializační sekvence – SendInitPulse
- Vygenerování Aktivního pulzu – SendActivePulse
- Vygenerování pasivního pulzu – SendPassivePulse

2. Napsání funkce pro obsluhu snímače – příjem odezvy

- Poslech pulsu - IC1Interrupt

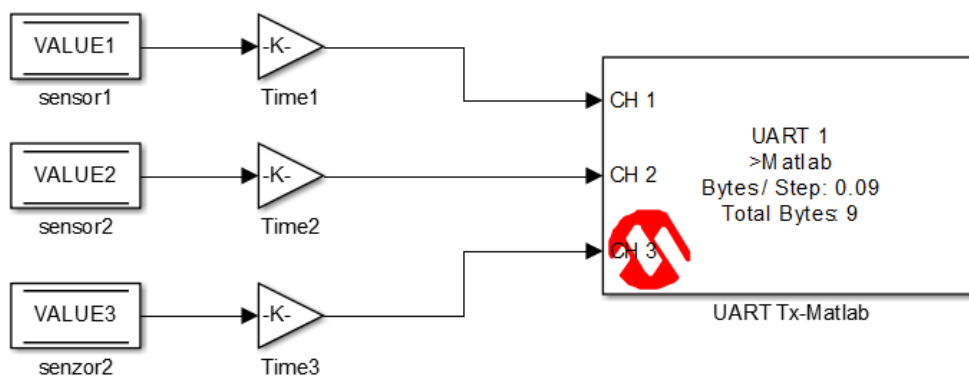
Generování aktivních pulsů popsaných v bodě 1 provádějí všechny senzory zároveň ve stejný okamžik vyšlou senzory zvukovou vlnu. Toto řešení jsme zvolili, protože zde pracujeme pouze se 3 senzory a k vyhodnocení překážek nepoužíváme složitý algoritmus, který by změnu modu aktivní/pasivní potřeboval.

4.1. Komunikace řídicí jednotky s Car4

Jelikož jedním z požadavků při návrhu desky bylo přidání CAN transceiveru, bude komunikace mezi Car 4 realizována právě pomocí CAN sběrnice. Autíčko Car4 nebylo v současné době v provozuschopném stavu a proto komunikace nemohla být otestována.

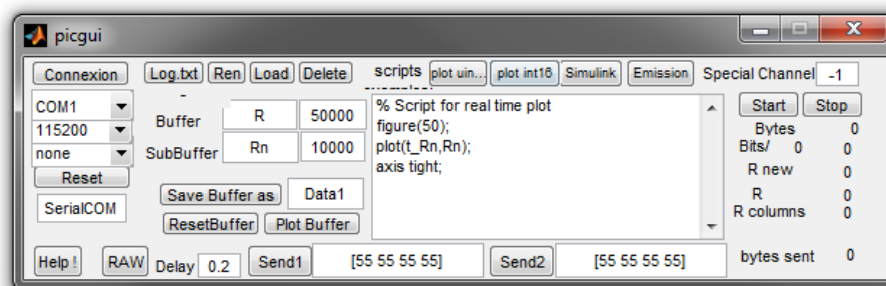
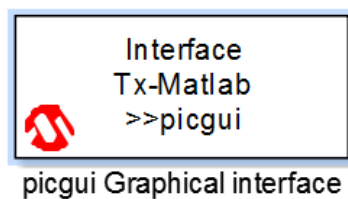
4.2. Zpracování dat ze senzoru a jejich vyhodnocení

Jelikož data ze senzorů představují „tiky“ microcontolleru, musí se nejprve převést na čas. Časovač timer2 byl nastaven s předděličkou 64 a základní takt hodin byl 40MHz, čímž se po vydělení těchto dvou hodnot (frekvence dělená předděličkou) dostáváme k frekvenci časovače 625KHz. Perioda, tedy čas, je pak obrácená hodnota frekvence, což po výpočtu dalo, že jeden tik Timeru2 se rovná 1,6 us. Bloky, starající se o zpracování dat, jsou ukázány na obr. 16.



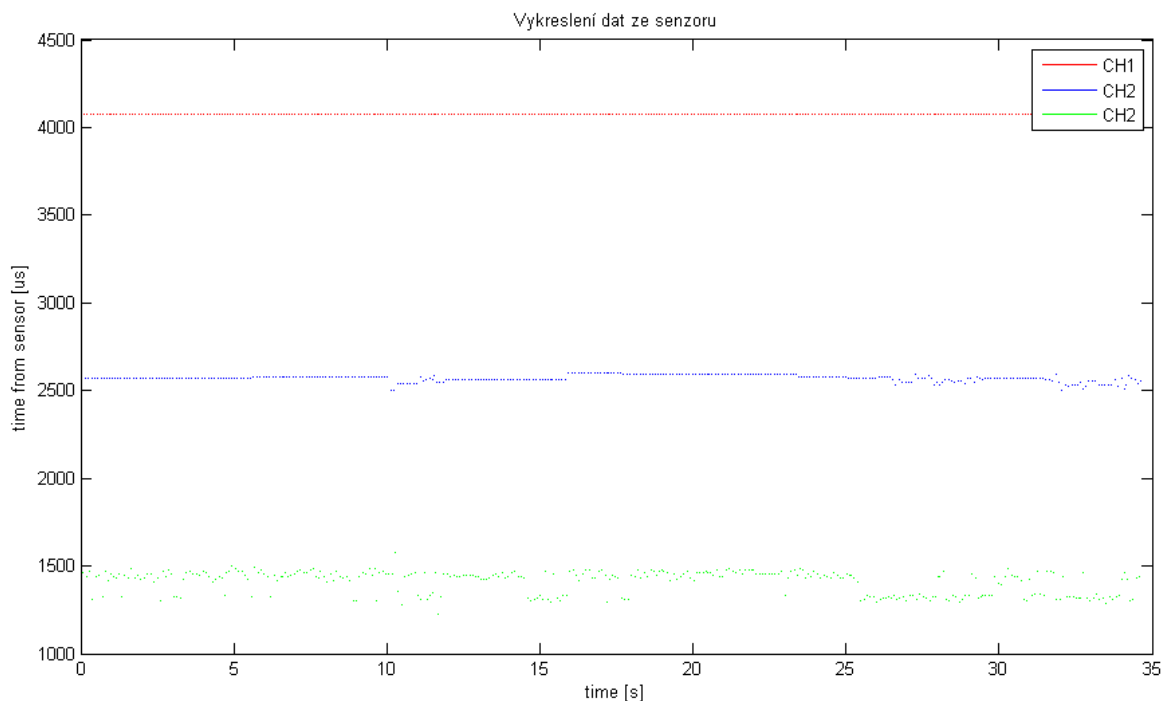
Obr. 16. Odesílání naměřených hodnot do PC přes linku UART

Bloky value1, value2, value3 čtou data ze senzorů resp. data uložená v globálních proměnných se stejným jménem. Pro převod tiků na čas se starají bloky gain, označeny Time1, Time2 a Time3. Poslední blok pak všechny data posílá přes UART na desce do UART převodníku, který je pomocí USB spojený s počítačem. Tyto data můžeme v simulinku číst pomocí bloku picgui Graphical (obr.17) interface.



Obr. 17. Picgui Graphical interface

Po jeho otevření se objeví grafické okno, kde si navolíme port, na kterém je UART převodník připojen. Můžeme také vybrat různé vykreslení dat, stejně jako je tomu v matlabu, a po stisknutí start se pak všechny data vykreslí do grafu. Při stisknutí tlačítka „log.txt“ nebo „Save Buffer as“ pak můžeme hodnoty ukládat do textového dokumentu a do proměnných v matlabu.

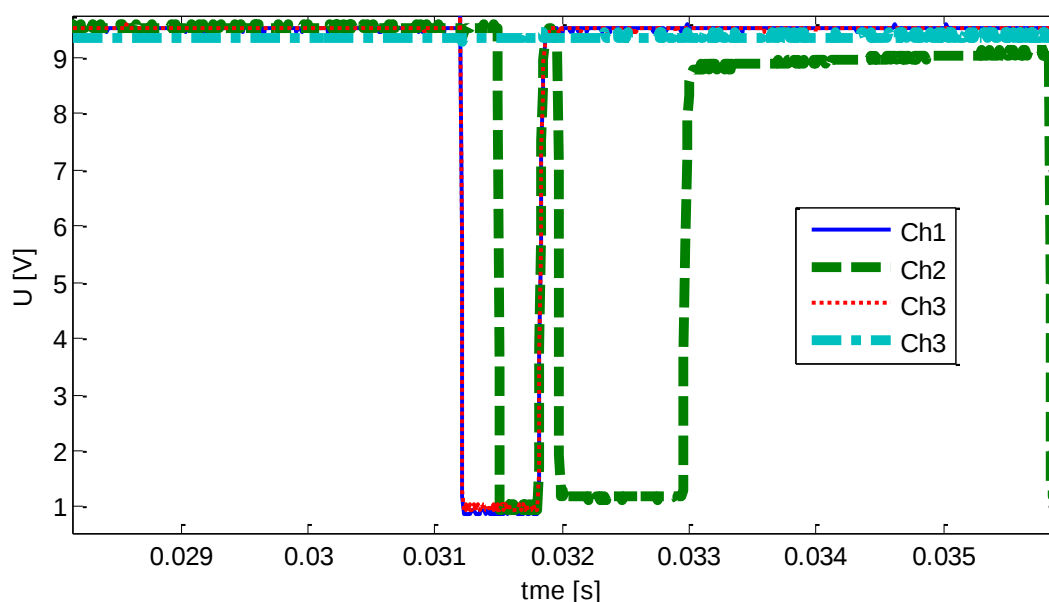


Obr. 18. Graf dat z UARTU

Na obr. 18 můžeme vidět data z UARTU, vykreslená pomocí matlabu. Hodnoty ze senzorů jsou pouze přepočítané na μs , tedy pouze vynásobené hodnotou 1,6. Senzory při měření byly položeny na stole, snímaly pouze strop.

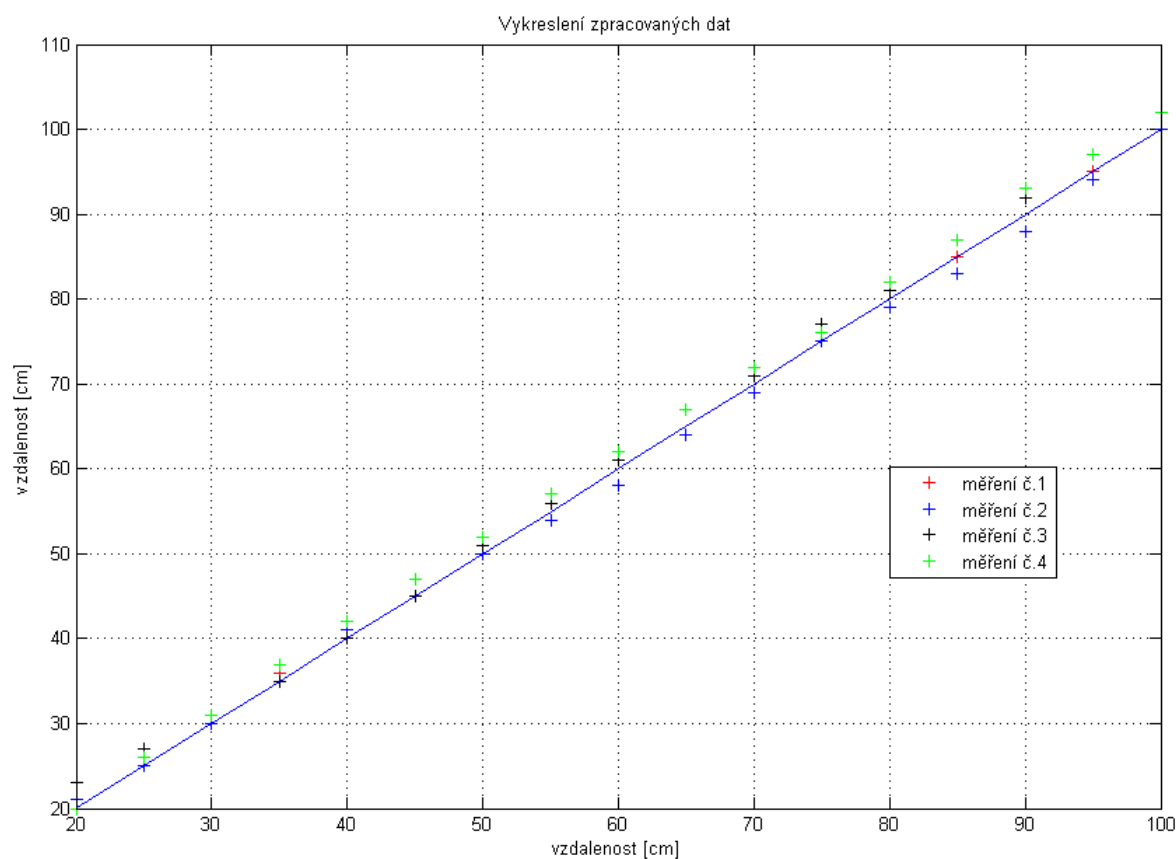
4.3. Testování přesnosti senzorů a rychlosti komunikace

Při měření jsme začínali až na hodnotě 20cm, protože senzor není schopen měřit velmi malé vzdálenosti, pro náš senzor konkrétně 17cm. To vychází z problému, který můžeme vidět na obr. 19 (data z osciloskopu vykreslená pomocí matlabu -> signál je zde invertovaný), že Ch2 je v aktivním módu, vysílá tedy zvukové vlny. Nejprve vyšle zvukovou vlnu a pak chvíli musí počkat na změnu módu, což na grafu symbolizuje překmit. Tato prodleva, v našem případě cca. 0,0005s, vynásobená s rychlostí zvuku za normálních podmínek, pak dala právě dráhu 17cm, tedy „hluchou“ vzdálenost senzoru.



Obr: 19. Graf vykreslený pomocí matlabu

Nejprve jsme si metrem odměřili různé vzdálenosti překážky (po 5 cm) od senzorů. Přepočítané vzdálenosti ze senzorů pak byly pomocí picgui interface uloženy do proměnných v matlabu a porovnány s reálnými hodnotami, viz. obr. 20.



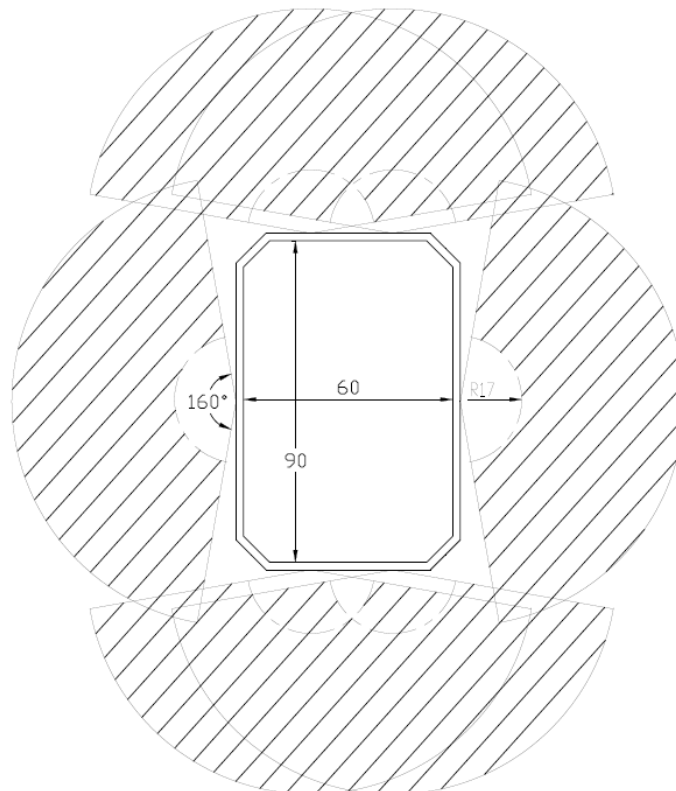
Obr. 20. Graf naměřených hodnot

Osy x a y představují vzdálenost od 20 po 100 cm. Senzory měří relativně přesně a jejich odchylka je maximálně 3cm. Překážku pro naše měření představovala stěna vyrobená z kartonu, tudíž nebyla ideálně rovná, což mohlo způsobit další nepřesnosti. Při zaměření senzorů na cihlovou rovnou stěnu byla odchylka při vzdálenosti 182cm pouze jeden centimetr.

5. Model překážky – zabránění kolize

5.1. Rozmístění senzorů na Car4

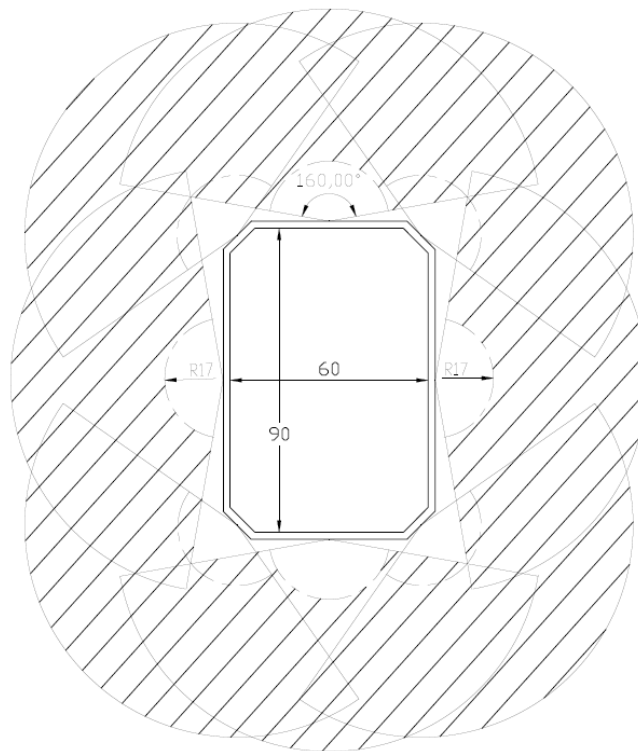
Původní návrh rozmístění senzorů byl, že dva senzory budou připevněny na přední nárazník a dva na zadní nárazník autíčka. Pro potřeby hledání parkovacího místa pak měl být na každou stranu umístěn ještě jeden senzor. Později se měli dva další senzory připevnit na věž ze servomotorů. Problém tohoto rozdělení je, že zvukové vlny by nepokryly celý prostor kolem Car4 a vznikali by zde hluchá místa, kde by senzory nebyly schopny měřit. Celý původní návrh je zobrazen na obr. 21.



Obr: 21. Původní návrh rozmístění senzorů

Šrafy zde představují místo, které je senzor schopen měřit (senzory mají rozpětí měření 160°), velká kružnice je pouze orientační a značí určitou pracovní oblast, malá kružnice je pak vzdálenostní limitace snímání senzorů.

Pozdější, tedy finální návrh, má na předním nárazníku 3 senzory a na zadním nárazníku 3 senzory. Jelikož téma hledání parkovacího místa je potřebné pro účely dalších bakalářských prací, byly senzory na stranách zachovány. Na obr. 22 lze vidět finální náčrtek pokrytí zvukových vln ze senzorů. Pokrytí oblasti autíčka je větší než při použití konfigurace se dvěma senzory

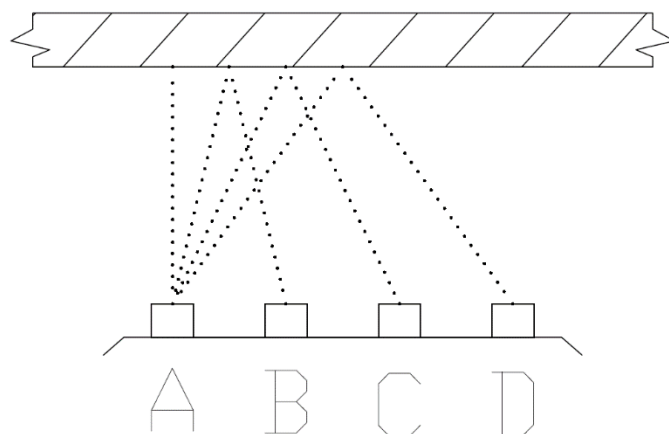


Obr: 22. Finální návrh rozmístění senzorů

5.2. Vytvoření modelu překážky

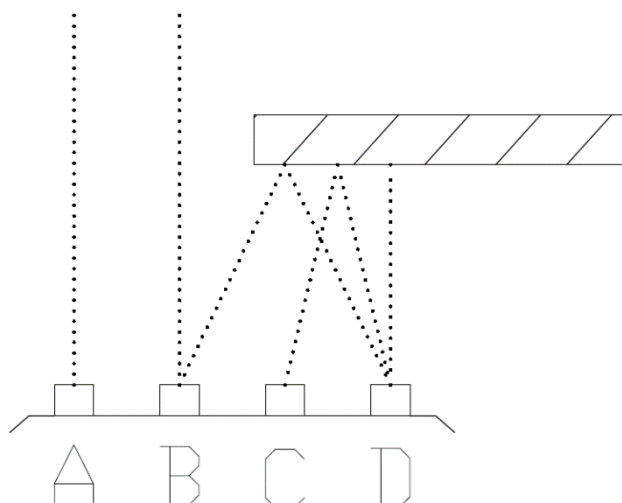
Při vytváření modelu překážky jsme uvažovali stav, kdy je překážka rovnoběžná s nárazníkem autíčka. Vstupem do modelu byly všechny naše známé hodnoty, tedy rychlost zvuku při teplotě 21°C, poloha a velikost překážky, počet senzorů a jejich rozmístění na nárazníku. Výstupem pak mají být všechny možné časy doby letu zvukových vln ze senzorů, které mohou nastat při dané velikosti překážky:

- Nekonečně dlouhá překážka – Všechny senzory jsou zapojeny do měření, tedy výstupem budou všechny možné kombinace časů a to ze senzoru A do senzoru A,B,C a D, dále ze senzoru B do senzoru A,B,C a D atd. celkově 16 možných. Schéma je ukázáno na obr. 23 (na obrázku uvažujeme pouze dráhu zvuků ze senzoru A).



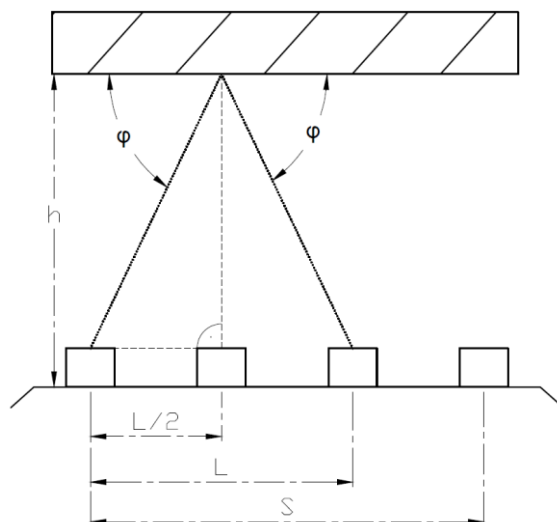
Obr: 23. Model pro nekonečně dlouhou překážku

- Překážka o určité velikosti – Jak můžeme vidět na obr. 24, při tomto stavu nemusí být všechny senzory zapojeny do měření (např. senzor A) resp. doba letu zvukové vlny z toho senzoru bude stanovena na nekonečno.



Obr: 24. Model pro specifickou překážku

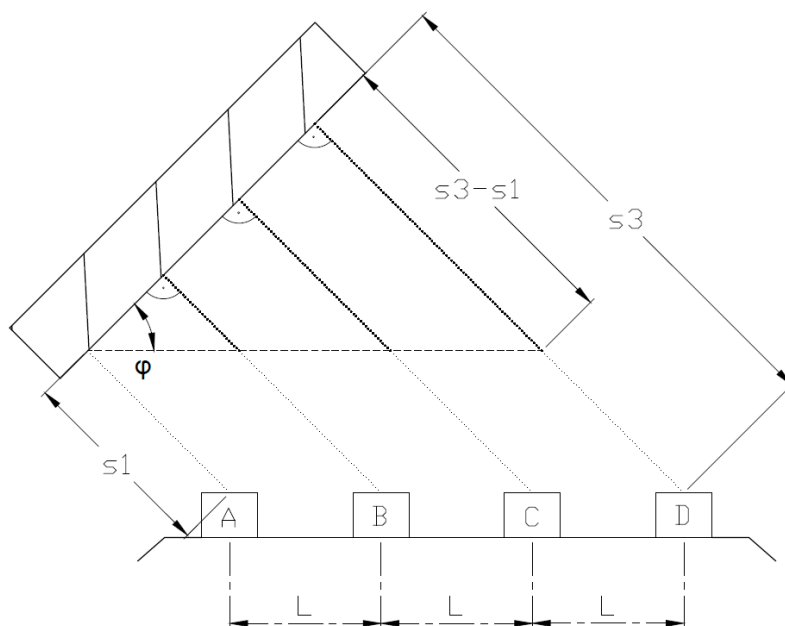
Pro výpočet času jsme předpokládaly, že zvuková vlna se odrazí od překážky pod stejným úhlem jako při dopadu. Místo odrazu resp. dopadu je pak přesně v půlce vzdálenosti mezi senzory. Tímto můžeme pomocí pythagorovy věty vypočítat uraženou vzdálenost a následně i dobu letu vln. Na obr. 25 lze vidět grafické znázornění a následně výpočet času.



Obr: 25. Grafické znázornění výpočtu doby letu

5.3. Vyhodnocení polohy překážky

Při použití 3 senzorů nejsme ještě schopni určit např. tvar libovolného objektu nebo detekovat více objektů záraz. Pokud však uvažujeme, že překážkou je rovný předmět (stěna, deska atd.), pak lze určit jeho vzdálenost od autíčka a tvar resp. úhel, který svírá s nárazníkem autíčka. Pro toto určení použijeme dráhy A-A, B-B, C-C (obr. 5.7) a porovnáme je. Pokud budou dráhy podobné, pak je překážka pravděpodobně rovnoběžná k nárazníku. Pokud jsou rostoucí, tedy A-A je nejmenší a C-C největší, lze předpokládat, že překážka je šikmo a svírá s kladnou osou x kladný úhel. V opačném případě, kdy je C-C nejmenší a A-A největší, svírá záporný úhel. Princip je zobrazen na obr. 26.

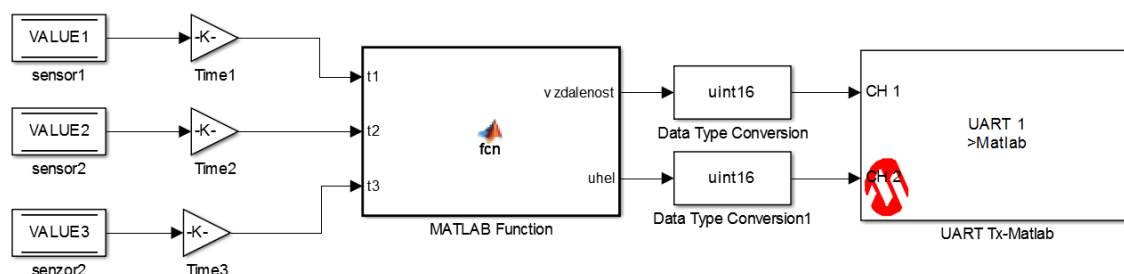


Obr: 26. Grafické znázornění výpočtu úhlu překážky

Nejdříve vezmeme nejmenší trojúhelník, jehož přeponu tvoří vodorovná čára ve výšce vzdálenosti A-A, jednu odvěsnu tvoří sama překážka a druhou odvěsnu vzdálenost $|B-B|-|A-A|$. Jelikož známe přeponu i odvěsnu, můžeme vypočítat úhel, který svírá překážka s námi vytvořenou vodorovnou osou. Stejný princip aplikujeme i na další vytvořené trojúhelníky. Získané úhly nakonec spolu porovnáme, a pokud se odchylka úhlů bude nalézat v určité toleranci, pak tento úhel přibližně odpovídá sklonu překážky. Pokud však budou data od sebe rozdílná, pak nastal stav, kdy snímač měří např. více překážek, malé předměty, zakřivené nebo zakulacené tvary apod. Vzdálenost překážky od senzoru pak uvažujeme vždy tu nejmenší.

5.4. Implementace algoritmu

Při implementaci algoritmu jsme použili v simulinku blok „MATLAB function“. Po otevření tohoto bloku můžeme naši funkci se vstupy a výstupy napsat přímo do matlab editoru. Simulink si po uložení funkce sám nastaví počet vstupů a výstupů.



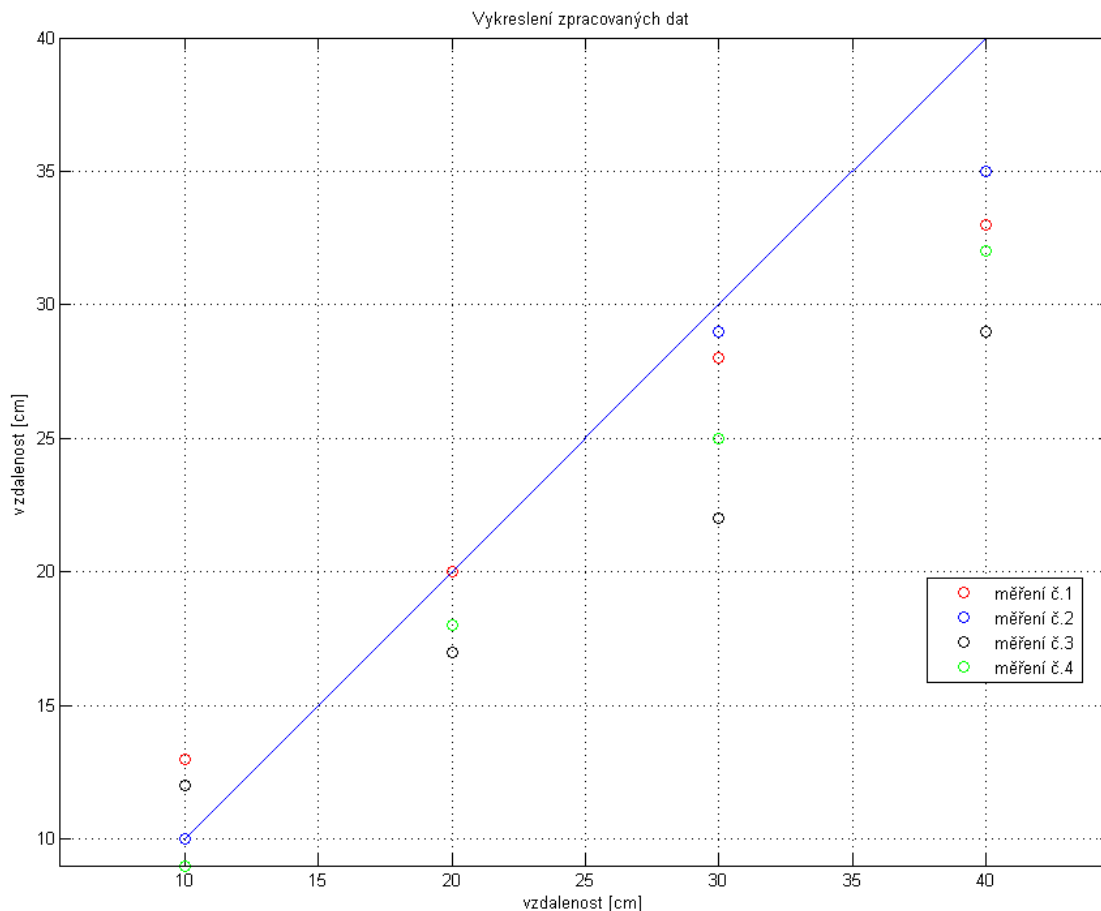
Obr: 27. Ukázka bloků ze simulinku

Na obrázku 27 můžeme vidět, že vstupy do soustavy jsou již přepočítané časy z timeru a výstupem vzdálenost a úhel natočení. Jelikož funkce používá datový typ double, je potřeba jej převést na typ uint16, abychom mohli tyto data poslat přes UART do převodníku a následně do počítače.

5.5. Testování námi vytvořeného algoritmu

Při testování algoritmu jsme postupovali stejně jako v předchozí kapitole 4. Komunikace s řídicí jednotkou. Použili jsme tedy znovu UART-TX pro posílání dat po UARTU do převodníku spojeného s počítačem a pomocí picgui Graphical interface byla pak data z měření převedena do proměnných v matlabu a následně vykreslena do grafu.

Měřili jsme s dostatečnou dlouhou a rovnou překážkou, abychom zamezili ztracení zvukových vln. Měření úhlu překážky je však i přes to relativně nepřesné. Pro malé úhly je odchylka $\pm 5^\circ$, avšak při zvyšujícím úhlu se odchylka zvětšuje až na $\pm 15^\circ$. Tento fakt je způsoben tím, že při větších úhlech se odražené zvukové vlny od překážky nevrátí zpět do senzoru, nebo se vrátí ve špatném čase, jelikož vlna se může odrazit od jiného náhodného bodu, než předpokládáme v našem algoritmu.



Obr. 28. Porovnání úhlů

5.6. Možná rozšíření modelu

Jelikož v našem modelu uvažujeme, že překážka je rovnoběžná s nárazníkem, lze tento model rozšířit i pro jiné tvary překážky, např. pro šikmou stěnu, větší množství překážek nebo pro zakulacenou překážku.

Pravděpodobný tvar šikmé překážky můžeme také zjistit z vyhodnocených dat ze dvou senzorů. Při použití jednoho senzoru lze zjistit vzdálenost překážky od senzoru, ale nelze zjistit její polohu resp. překážka se nachází někde na kružnici, jejíž poloměr je roven právě vzdálenosti předmětu od senzoru. Využijeme toho, že první senzor A bude nejprve v aktivním módu a druhý B v pasivním. Aktivní senzor tedy vyšle zvukovou vlnu a dále se oba dva přepnou do pasivního módu, kdy „poslouchají“ zvukové vlny. Vzniklé dráhy jsou z A-A a A-B. Jak již bylo řečeno z dráhy A-A zjistíme pravděpodobnou polohu překážky někde na kružnici. Z dráhy A-B zjistíme polohu překážky někde na elipse. Dále přepneme senzor B do aktivního režimu a zopakujeme postup. Vznikne nám tedy kružnice z bodu B-B elipsa B-A. Cílem je pak zjistit společnou tečnu všech těchto tří útvarů, která bude představovat tvar překážky.

Při detekci více objektů je postup problematický a je nutno použít více senzorů. Proto jsou na automobilech většinou na každé straně 4 senzory ve větší vzdálenosti

od sebe, aby byly schopny zaznamenat všechny objekty. Dalším řešením by mohlo být jiné uspořádání senzorů a to např. do půlkružnice nebo do tvaru U.

Problémem u zakulacených tvarů je, že pokud je překážka konkávní zvuk se v ní může libovolně odrážet a do senzoru se nám pak může vracet nesprávná zvuková vlna. Při konvexní překážce zase senzory detekují pouze vrchol překážky (nejkratší vzdálenost od senzoru), tudíž překážka může být vyhodnocena jako „tyčka. Problém by se dal vyřešit pauzou mezi měřeními, tedy senzor by vyslal zvukovou vlnu a počkal by, než všechny nežádoucí vlny odezní, a znovu by pak začal měřit.

6. Závěr

Pro zajištění komunikace senzorů s autíčkem byla nejprve navržena a následně vytvořena řídicí jednotka. Při jejím návrhu jsme vycházeli ze staré jednotky Škoda parkhil, kterou bylo potřeba upravit pro potřeby autíčka Car4. Po vytvoření této jednotky jsme prostřednictvím programu matlab simulink a MPLAB naprogramovali komunikaci mezi senzory, která byla dále testována. Pro určení úhlu a vzdálenosti byl ještě vytvořen program (Tvar_prekazky.m), který pomocí algoritmu a dat ze senzorů tyto hodnoty vypočítá. Na základě měření pak byly vytvořeny grafy (obr. 4.8 a obr 5.8) znázorňující přesnost senzorů a funkčnost námi vytvořeného algoritmu. Z naměřených hodnot vyplývá, že ultrazvukové senzory jsou relativně přesné a jejich odchylka od reálných hodnot je v řádech centimetrů. Jejich limitace je však při měření velmi malých vzdáleností (viz. kap. 4.5) a naopak s nimi nelze měřit velké vzdálenosti. Z obr. 5.2 lze pak vidět, že při současném rozložení senzorů na autíčku, můžeme pokrýt celý prostor kolem autíčka a tím pak v čas určit vzdálenost překážky a zabránit kolizi.

Po domluvě s vedoucím práce bylo rozhodnuto, že pro naši práci nebudou využity IR senzory (BP Šimurda 2010), jelikož jejich parametry by nebyly pro naše potřeby dostačující. Důvodem je, jak bylo napsáno v rešerši, že pro automobilový průmysl mají větší využití právě ultrazvukové senzory, jelikož jejich paprsek je více rozprostřen do prostoru, tudíž pokryje větší oblast. Použití IR senzorů by tedy mohlo způsobit komplikace a to např. že senzory by nedetekovaly překážky nižší než 5cm, překážky nad zemí apod. Z tohoto důvodu proto nebyly IR senzory využity.

Seznam použitých zdrojů

- [1] Datasheet: SN65HVD232Q . Texas Instruments , 2002. URL: <<http://www.ti.com/lit/ml/sgzv001/sgzv001.pdf>>
- [2] Datasheet: dsPIC33FJ128MC708 . Microchip Technology , 2009.
URL < <http://ww1.microchip.com/downloads/en/DeviceDoc/70287C.pdf>>
- [3] Installing MPLAB X IDE. Microchip Technology Inc. 2009.
URL <<http://microchip.wikidot.com/mplabx:installation>>
- [4] Ultrasonic sensor. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2014-06-04]. Dostupné z: http://en.wikipedia.org/wiki/Ultrasonic_sensor
- [5] ph-elec. James Kemp. 2010.
URL < <http://ph-elec.com/archives/hacking-gm-ultrasonic-park-assist-sensors/>>

Seznam obrázků:

Obr: 1.	Parkovací asistent Valeo	8
Obr: 2.	Nákres vnitřku senzoru	9
Obr: 3.	Rozměry ultrazvukového senzoru „4h0 919 275“	11
Obr: 4.	Konektor a ultrazvukový senzor „4h0 919 275“	12
Obr: 5.	Infračervený senzor	13
Obr: 6.	Princip fungování laserových senzorů	13
Obr: 7.	Připojení senzoru k microcontrolleru	15
Obr: 8.	Microcontroller DSPIC33FJ128MC708/PT	16
Obr: 9.	Napájení desky	16
Obr: 10.	Zapojení CAN Transceiveru	17
Obr: 11.	Navržená deska v programu Eagle	18
Obr: 12.	Ukázka vyrobené desky	19
Obr: 13.	Bloky v Simulinku	20
Obr: 14.	Ukázka programu MPLAB	21
Obr: 15.	Používané bloky	22
Obr: 16.	Odesílání naměřených hodnot do PC přes linku UART	23
Obr: 17.	Picgui Graphical interface	24
Obr: 18.	Graf dat z UARTU	24
Obr: 19.	Graf vykreslený pomocí matlabu	25
Obr: 20.	Graf naměřených hodnot	26
Obr: 21.	Původní návrh rozmístění senzorů	27
Obr: 22.	Finální návrh rozmístění senzorů	28
Obr: 23.	Model pro nekonečně dlouhou překážku	29
Obr: 24.	Model pro specifickou překážku	29
Obr: 25.	Grafické znázornění výpočtu doby letu	30
Obr: 26.	Grafické znázornění výpočtu úhlu překážky	30
Obr: 27.	Ukázka bloků ze simulinku	31
Obr: 28.	Porovnání úhlů	32

Příloha:

Seznam použitých součástek:

Název	Počet
CKS0603 100n/16V X7R	4
CKS0603 68p/50V NPO	2
SN65HVD232Q	1
CTS 10u/16V AVX-TAJ A 10%	1
CKS0805 100N X7R	1
1N5341B	1
1N5335B	1
1N5350B	1
24C02 SMD	1
PSH02-02PG	1
PSH02-03PG	8
PFH02-02P	1
PFH02-03P	8
PFF02-01FG	26
LED 0805 RED	3
LED 0805 GREEN	3
Q 8MHZ	1
BC817-25 SMD	19
R0805 120R	8
R0805 150R	4
R0805 330R	1
R0805 560R	4
R0603 2K2 1%	2
R0805 4k7	17
R0805 9k1	8
R0805 10K	9
R0805 22k	8
MA02-2	2
TSM2302CX RF	1
WAGO255-742	1
WAGO255-747	1
WAGO255-746	1
WAGO255-740	1
WAGO255-600	1