

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Matej Peršolja

**Sistem za pomoč vozniku izveden na modelu  
radijsko vodenega avtomobila**

DIPLOMSKO DELO  
NA UNIVERZITETNEM ŠTUDIJU

Mentor: prof. dr. Dušan Kodek

Ljubljana, 2010



Št. naloge: 01697/2010

Datum: 01.09.2010

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **MATEJ PERŠOLJA**

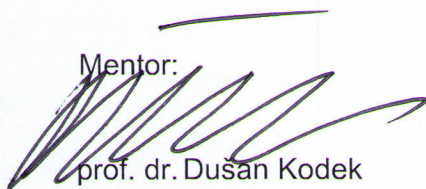
Naslov: **SISTEM ZA POMOČ VOZNIKU IZVEDEN NA MODELU RADIJSKO  
VODENEGA AVTOMOBILA**  
**DRIVER ASSISTANCE SYSTEM IMPLEMENTED ON A RADIO  
CONTROLLED MODEL CAR**

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Namen naprednih sistemov za pomoč vozniku je pomoč pri vožnji s ciljem povečanja varnosti pri vožnji in v prometu sploh. Funkcije kot so avtomatsko uravnavanje razdalje, pomoč pri ohranjanju smeri, avtomatsko parkiranje, razpoznavanje prometnih znakov in podobne, so že danes vgrajene v avtomobile višjega cenovnega razreda. Na osnovi pregleda obstoječih rešitev razvijte in izdelajte strojno in programsko opremo, ki sta potrebni za izvedbo najmanj treh funkcij sistema za pomoč vozniku na modelu radijsko vodenega avtomobila. Za izdelavo uporabite Arduino razvojno platformo in standardne elektronske komponente. Delovanje sistema preizkusite in opišite morebitne pomanjkljivosti.

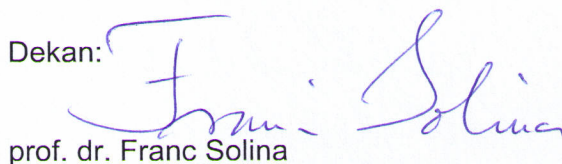
Mentor:



prof. dr. Dušan Kodek



Dekan:



prof. dr. Franc Solina

Rezultati diplomskega dela so intelektualna lastnina Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje Fakultete za računalništvo in informatiko ter mentorja.

*Besedilo je oblikovano z urejevalnikom besedil L<sup>A</sup>T<sub>E</sub>X.*

Namesto te strani **vstavite** original izdane teme diplomskega dela s podpisom mentorja in dekana ter žigom fakultete, ki ga diplomant dvigne v študentskem referatu, preden odda izdelek v vezavo!



# IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani     Matej Peršolja,

z vpisno številko     63030025,

sem avtor diplomskega dela z naslovom:

Sistem za pomoč vozniku izveden na modelu radijsko vodenega avtomobila

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom prof. dr. Dušana Kodeka,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 5. 9. 2010

Podpis avtorja:



# Zahvala

Zahvaljujem se vsem, ki so me podpirali in mi stali ob strani v času študija, mentorju prof. dr. Dušanu Kodeku za nasvete in pomoč pri izdelavi diplomske naloge in podjetju XLAB d. o. o., ki je financiralo nakup vsega potrebnega materiala za realizacijo diplomske naloge.



*Diplomsko delo posvečam kolegom in kolegicam iz smeri računalniški sistemi, s katerimi sem uspešno zaključil študij.*



# Kazalo

<b>Povzetek</b>	<b>1</b>
<b>Abstract</b>	<b>3</b>
<b>1 Opis naprednih asistenčnih sistemov</b>	<b>5</b>
1.1 Pregled asistenčnih sistemov . . . . .	5
1.2 Avtomatsko uravnavanje razdalje . . . . .	6
1.3 Asistenca za ohranjanje smeri . . . . .	7
1.4 Parkirni sistem s parkirno avtomatiko . . . . .	7
<b>2 Opis uporabljenih komponent</b>	<b>10</b>
2.1 Radijsko voden model avtomobila . . . . .	10
2.2 Mikrokrmilnik . . . . .	11
2.3 Arduino kit . . . . .	12
2.4 Infrardeči senzorji razdalje . . . . .	13
2.4.1 Princip delovanja . . . . .	13
2.4.2 Sharp GP2D120 . . . . .	13
2.4.3 Sharp GP2Y0A02 . . . . .	14
2.5 Senzor odbojnega koeficienta . . . . .	14
2.5.1 Princip delovanja . . . . .	15
2.5.2 Parallax QTI Light Sensor . . . . .	15
<b>3 Sestavljanje in pritrditev senzorjev</b>	<b>17</b>
3.1 Testiranje senzorjev . . . . .	17
3.2 Pritrditev senzorjev razdalje . . . . .	19
3.3 Pritrditev razvojne ploščice in mikrokrmilnika . . . . .	20
3.4 Pritrditev senzorja odbojnega koeficienta . . . . .	20
3.5 Pritrditev ostalih komponent . . . . .	21
<b>4 Krmiljenje servomotorjev in interpretacija signalov iz senzorjev</b>	<b>25</b>
4.1 Krmiljenje servomotorjev . . . . .	25
4.1.1 Osnove delovanja servo krmiljenja . . . . .	25

4.1.2	Branje servo signala . . . . .	26
4.1.3	Pisanje servo signala . . . . .	26
4.2	Interpretacija signalov iz senzorjev . . . . .	26
4.2.1	Branje razdalje senzorja Sharp GP2D120 . . . . .	26
4.2.2	Branje razdalje senzorja Sharp GP2Y0A02 . . . . .	27
4.2.3	Branje razdalje senzorja Parallax QTI Light Sensor . . . . .	27
<b>5</b>	<b>Izvedba naprednega asistenčnega sistema</b>	<b>28</b>
5.1	Izvedba funkcije avtomatskega uravnavanja razdalje . . . . .	28
5.2	Izvedba funkcije asistencije voznega pasu . . . . .	32
5.2.1	Podroben opis delovanja . . . . .	32
5.3	Izvedba funkcije avtomatskega parkiranja . . . . .	34
5.3.1	Bočno parkiranje, kot ga učijo v avtošoli . . . . .	35
5.3.2	Iskanje prostega parkirnega mesta . . . . .	36
5.3.3	Bočno parkiranje . . . . .	38
5.4	Izvedba ročnega upravljanja vozila . . . . .	42
5.5	Izbira funkcije asistenčnega sistema . . . . .	44
<b>6</b>	<b>Rezultati</b>	<b>45</b>
6.1	Asistenca uravnavanja razdalje . . . . .	45
6.2	Asistenca za ohranjanje smeri . . . . .	45
6.3	Asistenca parkiranja . . . . .	45
6.4	Avtomatsko parkiranje . . . . .	46
6.5	Zaključek . . . . .	46
	<b>Seznam slik</b>	<b>47</b>
	<b>Literatura</b>	<b>49</b>

# Seznam uporabljenih kratic in simbolov

ACC - Adaptive Cruise Control  
E.S.C. - Electric motor Speed Controller  
PWM - Pulse-Width Modulation  
SRAM - Static Random Access Memory  
EEROM - Electrically Erasable Read-Only Memory  
USB - Universal Serial Bus  
ICSP - In-Circuit Serial Programming  
CCD - Charge-Coupled Device



# Povzetek

Po navedbah National Highway Traffic Safety Administration (NHTSA) [1] 93 odstotkov vseh prometnih nesreč vključuje napake voznika. Večina teh nesreč je povezanih z voznikovo nepozornostjo. Raziskava NHTSA tudi kaže, da bi ena sekunda več opozorila lahko preprečila do 90 odstotkov trčenj v zadnji del vozila.

Napredni asistenčni sistemi so sistemi, ki pomagajo vozniku pri vožnji osebnega avtomobila. Kadar so sistemi zasnovani z varnim vmesnikom človek-stroj, naj bi povečali varnost pri vožnji in splošno varnost v prometu.

Diplomska naloga opisuje izvedbo naprednega asistenčnega sistema na modelu radijsko vodenega avtomobila. V delu so predstavljene tri asistenčne funkcije, ki so tudi izvedene na modelu radijsko vodenega avtomobila. Opisani so vsi uporabljeni senzorji in pritrditev le-teh na model. Nadrobno je opisana programska rešitev za vsako funkcijo sistema, podane pa so tudi omejitve sistemov.

## Ključne besede:

napredni asistenčni sistem, model avtomobila, arduino, servo, varnost v prometu

# Abstract

According to the National Highway Traffic Safety Administration (NHTSA) [1] 93 percent of all car crashes is caused by a driver error. Most of the crashes are caused by the driver inattention. The survey also shows that a notice one second before the crash would prevent up to 90 percent of rear crashes.

Advanced Driver Assistance Systems are systems that help the driver in His driving process. When designed with a safe Human-Machine Interface it should increase car safety and more generally road safety.

The thesis describes the implementation of an advanced driver assistance system, on a radio-controlled car model. In the thesis are described three assistance functions, which were also implemented on the radio controlled car model. All the sensors and the assembly to the car model are described. The software solution for all assistance functions is described and the limitations of the systems are presented.

## Key words:

advanced driver assistance systems, car model, arduino, servo, road safety

# Poglavje 1

## Opis naprednih asistenčnih sistemov

### 1.1 Pregled asistenčnih sistemov

Napredni asistenčni sistemi pomagajo vozniku pri pridobivanju informacij iz okolice, olajšajo njegove odločitve in razbremenjujejo voznika neljubih opravil. Sprva so bili vgrajeni le v avtomobile najvišjega cenovnega razreda, v zadnjih letih pa lahko zasledimo uporabo nekaterih tudi v avtomobilih nižjega cenovnega razreda. V naslednjih poglavjih so podrobno opisani trije napredni asistenčni sistemi, ki so bili tudi izvedeni kot enoten asistenčni sistem na modelu radijsko vodenega avtomobila. Seznam obstoječih naprednih asistenčnih sistemov za pomoč vozniku:

- Navigacijski sistemi
- Avtomatsko uravnavanje razdalje
- Asistenca za ohranjanje smeri
- Asistenca menjave voznega pasu
- Sistem za izogibanje trčenju
- Nočni vid
- Smerno prilagodljiva žarometa
- Parkirni sistem
- Prepoznavna prometnih znakov
- Detekcija vozil v mrtvem kotu
- Prepoznavanje zaspanosti voznika
- Komunikacija med vozili

## 1.2 Avtomatsko uravnavanje razdalje

Sistem za avtomatsko uravnavanje razdalje (Adaptive Cruise Control) združuje funkcije tempomata in radarskega senzorja (Slika 1.1). Ko je sistem vključen, samodejno vzdržuje izbrano oddaljenost od spredaj vozečega vozila in temu ustrezno prilagaja hitrost vožnje. S tem močno razbremeni voznika in poveča udobje.



Slika 1.1: Radarski senzor

Osrednja komponenta sistema za avtomatsko uravnavanje razdalje je radarski senzor z dosegom 200 m v območju, ki s sredinsko vzdolžno osjo vozila tvori kot 12 stopinj (na levi in desni strani). Radarski senzor in krmilnik tvorita eno enoto. Prednost radarske tehnologije je, da zanesljivo deluje ne glede na vremenske razmere in razmere na cestišču.

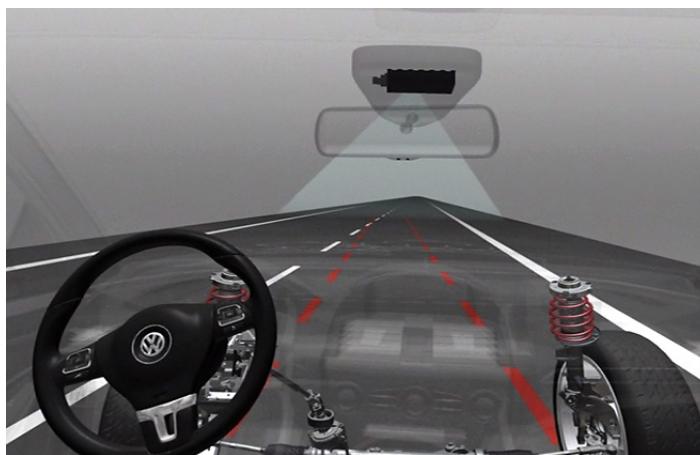


Slika 1.2: Oddaljenost spredaj vozečega vozila

Krmilnik iz signalov radarskega senzorja izračuna oddaljenost in relativno hitrost glede na spredaj vozeče vozilo (Slika 1.2). Na cestah z več voznimi pasovi dodatno izračuna še zamik med obema voziloma. Če je v območju dosega senzorja hkrati več vozil, krmilnik na osnovi dobljenih podatkov določi tisto vozilo, na osnovi katerega sistem prilagaja hitrost, in tako vzdržuje izbrano razdaljo. Radarski senzor ne zaznava mirujočih ovir, kot so npr. stoječe kolone in zaščitne ograje.

### 1.3 Asistenca za ohranjanje smeri

Na daljših vožnjah z monotonimi voznimi situacijami voznikova koncentracija pogosto popusti. Zato ni presenetljivo, da se številne nezgode s poškodbami zgodijo zato, ker eden od voznikov nenadoma zapelje s svojega voznega pasu. Asistenca za ohranjanje smeri (Lane Assist) lahko pripomore k preprečevanju tovrstnih nezgod: s posegom v krmiljenje namreč po potrebi korigira zasuk volana in vozniku pomaga ohraniti želeno smer vožnje.



Slika 1.3: Kamera v predelu notranjega vzvratnega ogledala

Če je sistem aktiviran, se samodejno vključi pri povečanju hitrosti nad 65 km/h. Preko kamere v predelu notranjega vzvratnega ogledala (Slika 1.3) spremlja označbe na cestišču in analizira položaj vozila. V primeru odstopanj od predvidene smeri vožnje ustrezno korigira zasuk volana. Če maksimalni krmilni moment ne zadošča za ponovno zagotovitev ustrezne smeri ali če se hitrost vozila zmanjša pod 60 km/h, asistenca za ohranjanje smeri voznika na nevarnost opozori z zvočnim signalom. V tej situaciji mora voznik smer vožnje ustrezno korigirati sam.

Ko je Lane Assist aktiviran, sveti rumen simbol, ki ponazarja cestišče (Slika 1.4). Takoj ko kamera locira ustrezne označbe na cestišču, se rumeni simbol spremeni v zelenega. Sedaj je sistem v celoti aktiven. Če se vozilo oddalji od idealne linije, ga Lane Assist s počasnim in kontinuiranim sukanjem volana ponovno usmeri v pravo smer. Če voznik volan povsem spusti, ga sistem z glasovnimi signali in opozorilnim besedilom v kombiniranem instrumentu pozove, naj znova prevzame nadzor nad krmilom.

### 1.4 Parkirni sistem s parkirno avtomatiko

Parkirni sistem s parkirno avtomatiko (Park Assist) pri vzvratnem bočnem parkiranju samodejno vrtil volan in vozilo po idealni liniji vodi do izbranega parkirnega mesta. Voznik



Slika 1.4: Aktivna asistenca za ohranjanje smeri

mora le ustrezno pritiskati na pedal za plin in na zavorni pedal. Tudi ustreznost parkirnih mest samodejno določa sistem, in sicer na osnovi meritev senzorjev (Slika 1.5). Voznik ima med parkiranjem ves čas popoln nadzor nad vozilom. Če sam začne vrteti volan ali če s pritiskom na zavorni pedal zaustavi vozilo, se krmilna avtomatika izključi. Vse, kar mora voznik storiti, je, da z ustrezno tipko vključi sistem in da s prilagojeno hitrostjo (največ 30 km/h) vozi vzporedno z robom cestišča. Parkirni sistem s parkirno avtomatiko (Park Assist) s pomočjo senzorjev preverja velikost parkirnih mest na levi in na desni strani (npr. v enosmernih ulicah). Pred začetkom parkiranja voznika s prikazi v kombiniranem instrumentu usmeri na ustrezni izhodiščni položaj. Ob pomiku prestavne ročice v vzvratno prestavo se vključi krmilna avtomatika. Voznik lahko med parkiranjem mirno spusti volan, še vedno pa mora budno spremljati dogajanje ob vozilu. On je namreč tisti, ki nosi odgovornost za morebitne neljube dogodke. Asistenčni sistemi so povzeti iz modelov Volkswagnovih vozil [6]. Drugi proizvajalci vozil tudi delajo na razvoju asistenčnih sistemov, ki delujejo na podobnih principih.



Slika 1.5: Domet senzorjev

## Poglavje 2

# Opis uporabljenih komponent

### 2.1 Radijsko voden model avtomobila

Za model je bil uporabiljen radijsko vodeni avtomobil podjetja LRP z oznako S10 Blast BX (Slika 2.1). Model poganja visoko zmogljivostni brezkrtačni elektromotor. Pogon se preko sredinske osi poganja na vsa 4 kolesa. Obračanje sprednjih koles je krmiljeno preko servomotorja. Uravnavanje moči motorja poteka preko E.S.C. naprave (Electric Motor Speed Controller), ki glede na vhodni servo signal uravnava hitrost in smer vrtenja elektromotorja.



Slika 2.1: S10 Blast BX

Upravljanje modela poteka preko visoko frekvenčne daljinske dvokanalne krmilne palice (Slika 2.2). Oddajnik oddaja signal na frekvenci 2.4 GHz. Sprejemnik, ki sprejema ta signal, vodi servomotor za krmiljenje smeri in E.S.C. za krmiljenje hitrosti vozila (Slika 2.3).



Slika 2.2: Krmilna palica



Slika 2.3: Sprejemnik

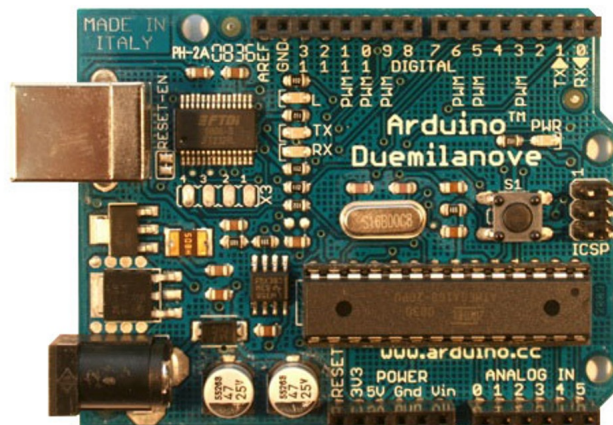
## 2.2 Mikrokrmilnik

Za upravljanje avtomobila in simulacijo izvedbe asistenčnega sistema na modelu avtomobila je bil uporabiljen Arduino Duemilanove mikrokrmilnik (Slika 2.4).

Arduino je univerzalni mikrokrmilnik zasnovan na ATmel tehnologiji in je idealen za elektroniko in robotiko. Isto platformo uporablja tudi znani robotski sesalnik iRobot - Roomba, Scooba in ostali iz družine. Arduino je odprtokodna platforma zasnovana na enostavni razvojni ploščici z vhodno/izhodnimi konektorji in uporabniku prijaznem razvojnem okolju (brezplačna programska oprema). Podatki povzeti po [7].

Arduino 2009 je osnovni model mikrokrmilnika.

- Mikroprocesor: ATMega328
- Frekvenca procesorja: 16 MHz
- Napetost: 5 V



Slika 2.4: Arduino Duemilanove

- Vhodna napetost (priporočeno): 7-12 V
- Vhodna napetost (omejena): 6-20 V
- Digitalni vhodno/izhodni pini: 14 (od tega 6 PWM izhodi)
- Analogni vhodni pini: 6
- Pomnilnik: Flash (32 kB), SRAM (2 kB) in EEROM 1 kB
- Enosmerni tok na vhodno/izhodni pinih: 40 mA
- Enosmerni tok za pine 3,3 V: 50 mA
- Priključki: USB, ICSP priključek
- Tipka: za resetiranje

## 2.3 Arduino kit

Za realizacijo diplomske naloge je bil poleg Arduino mikrokrmilnika uporabljen še Arduino kit komplet (Slika 2.5). Ta vsebuje led diode, upore, razvojno ploščico in ostale komponente, ki so potrebne za razvoj.



Slika 2.5: Arduino kit komplet

## 2.4 Infrardeči senzorji razdalje

Za izmero razdalje do predmetov okoli avtomobila so bili uporabljeni IR senzorji. Za meritev razdalje na krajših razdaljah je bil uporabljen Sharp GP2D120 IR Sensor, za daljše razdalje pa zmogljivejši model GP2Y0A21.

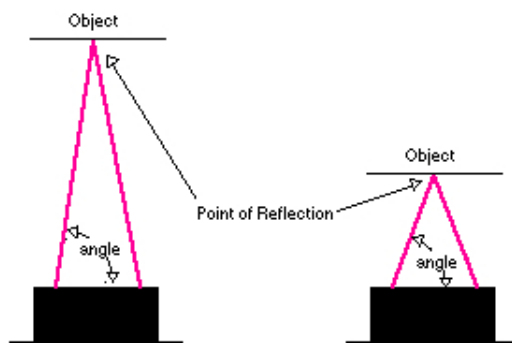
### 2.4.1 Princip delovanja

Z uvedbo serije GP2DXX Sharpovih detektorjev razdalj je bil razvit nov pristop detekcije predmetov. Ta poleg detekcije navzočnosti omogoča tudi detekcijo razdalje (modeli GP2D12, GP2D120 in GP2DY0A). Novi detektorji omogočajo tudi mnogo večjo odpornost proti motnjam zaradi ambientne svetlobe. Novi detektorji razdalje uporabljajo za določitev prisotnosti in razdalje predmeta triangulacijo in majhno linearno polje CCD (Charge-Coupled Device). Osnovna ideja je naslednja: oddajnik odda žarek IR svetlobe. Žarek potuje po vidnem spektru in ali zadene oviro ali pa potuje naprej. Če ovire ne zadene, ne pride do odboja in detektor ne zazna predmeta. Če pa zadene oviro, se žarek odbije in tako se naredi trikotnik med oddajnikom, oviro in sprejemnikom. Koti v tem trikotniku se spreminjajo glede na oddaljenost predmeta (Slika 2.6). Sprejemnik ima natančno lečo, ki glede na kot pošilja različne dele svetlobe na polje CCD. Tako lahko polje CCD zazna, pod kakšnim kotom se je vrnila IR svetloba in tako izračuna oddaljenost predmeta.

### 2.4.2 Sharp GP2D120

Sharp GP2D120 lahko natančno in zanesljivo bere razdalje od 4 do 30 cm (Slika 2.7). Glede na oddaljenost vrača analogni signal, iz katerega se da izračunati oddaljenost

(Slika 2.8).



Slika 2.6: Princip delovanja IR senzorja razdalje



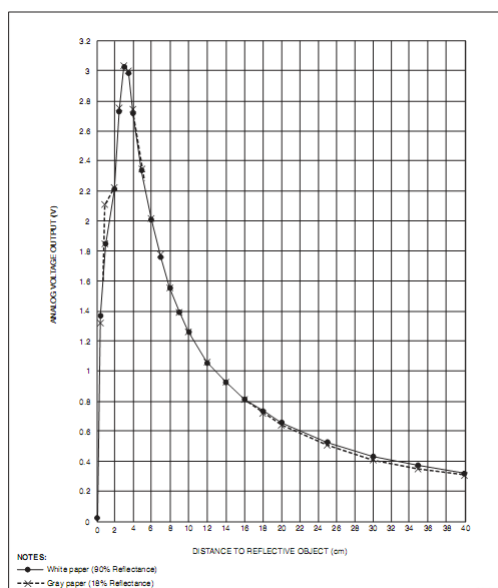
Slika 2.7: Sharp GP2D120

### 2.4.3 Sharp GP2Y0A02

Sharp GP2Y0A02 je najzmogljivejši senzor iz družine Sharpovih IR senzorjev, saj lahko natančno in zanesljivo bere razdalje od 20 do 150 cm (Slika 2.9). Glede na spremembo razdalje se spreminja izhodna napetost (Slika 2.10).

## 2.5 Senzor odbojnega koeficienta

Senzor odbojnega koeficienta je bil uporabljen za zaznavanje bele črte na cestišču. Uporabljen je Parallaxov QTI Light Sensor (Slika 2.11).



Slika 2.8: Sprememba napetosti glede na oddaljenost



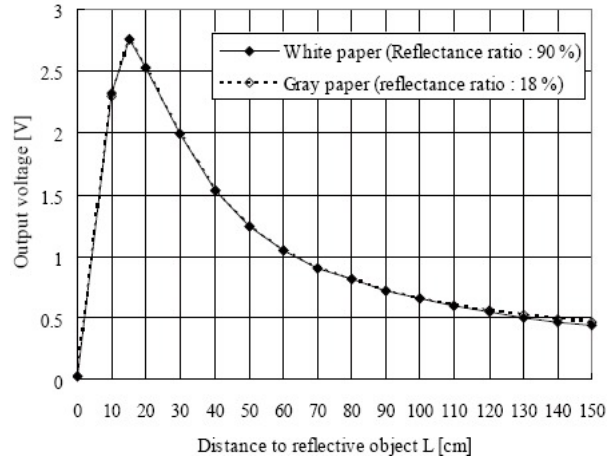
Slika 2.9: Sharp GP2Y0A02

### 2.5.1 Princip delovanja

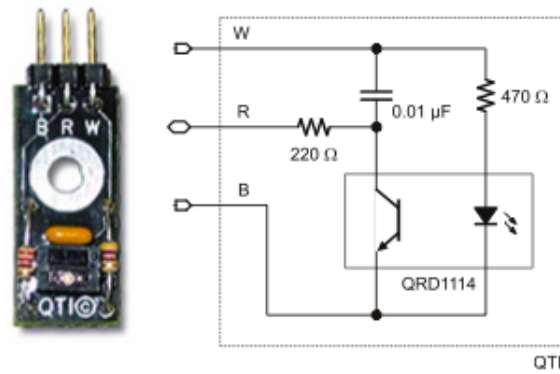
Senzor odbojnega koeficienta deluje na principu odboja svetlobe od površine, ki se nahaja pod senzorjem. Na primer: ko je senzor nad temno površino, je odbojni koeficient zelo majhen, in ko je senzor nad svetlo površino, je odbojni koeficient zelo velik.

### 2.5.2 Parallax QTI Light Sensor

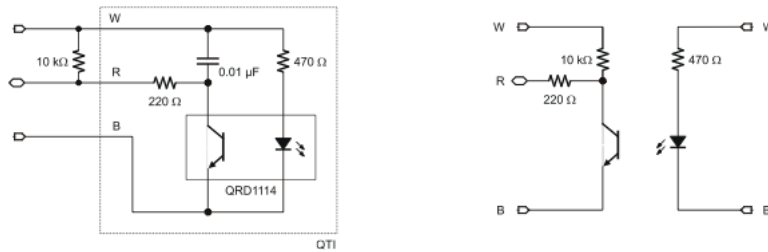
Za zaznavanje le črne ali bele se da senzor spremeniti v digitalni senzor tako, da mu dodamo  $10\text{ k}\Omega$  upor med vhodoma W in R. Senzor se nato obnaša podobno, kot je prikazano na sliki 2.12.



Slika 2.10: Krivulja izhodne napetosti glede na razdaljo



Slika 2.11: Parallax QTI Light Sensor



Slika 2.12: Senzor odbojnega koeficienta z dodanim uporom

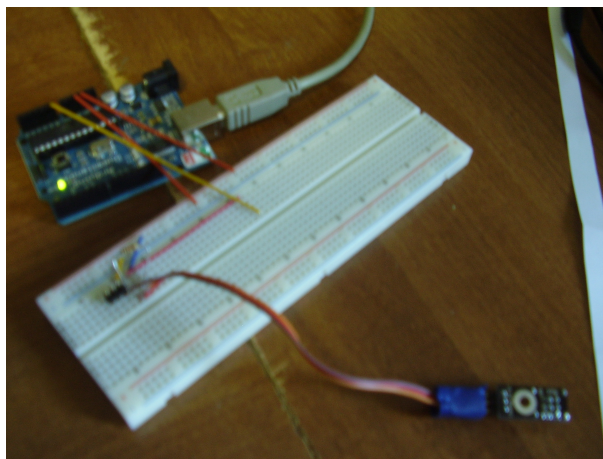
## Poglavje 3

# Sestavljanje in pritrditev senzorjev

Radijsko voden avtomobil ima močan elektromotor, zato je bilo nekaj časa posvečeno temu, da med delovanjem ne bi prišlo do neželenih prekinitev komunikacije med senzorji in mikrokontrolnikom. Pred pritrditvijo na radijsko voden avtomobil je bilo delovanje posameznih senzorjev najprej preverjeno.

### 3.1 Testiranje senzorjev

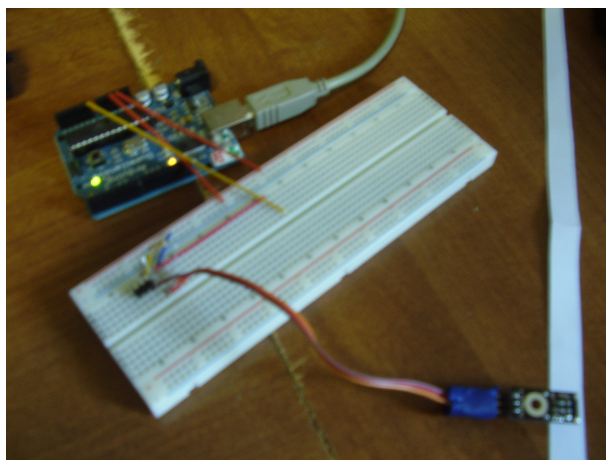
Začel sem s senzorjem Parallax QTI Light Sensor.



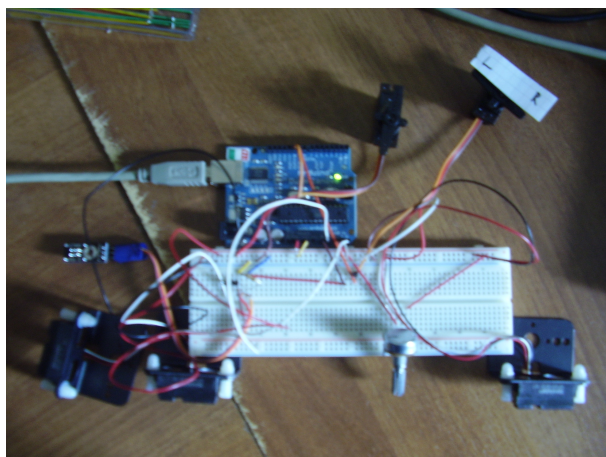
Slika 3.1: Bela podlaga ni zaznana

Na osnovi testiranja je bila določena mejna vrednost kdaj naj bo vrednost senzorja interpretirana kot bela podlaga (Slika 3.2) in kdaj ne (Slika 3.1). Sledilo je testiranje ostalih senzorjev (Slika 3.3).

Testiranje ultrazvočnih senzorjev je bilo opravljeno tako, da so bile na listu papirja označene razdalje in sem bel list oddaljeval in približeval senzorju razdalje. V testnem



Slika 3.2: Bela podlaga je zaznana



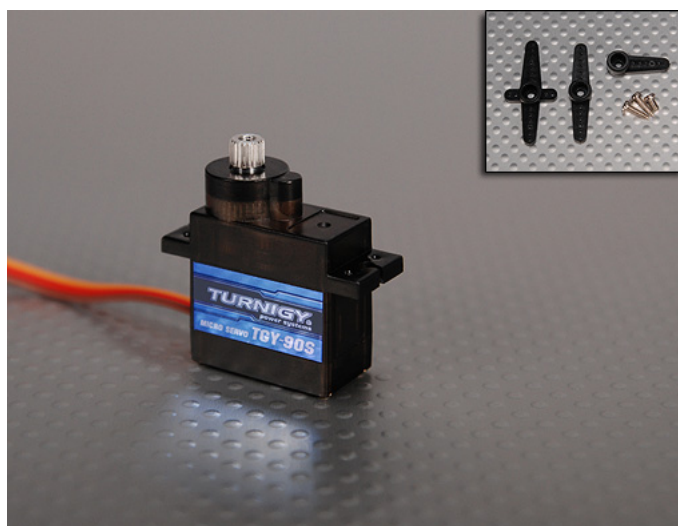
Slika 3.3: Testiranje senzorjev

programu je mikrokontroler preko serijske komunikacije pošiljal izračunane vrednosti razdalje. Za začetno testiranje upravljanja servomotorja z Arduino mikrokontrolerom je bil uporabljen potenciometer. Testirano je bilo upravljanje mikro servomotorja TGY-90S (Slika 3.4). Za testiranje je bil uporabljen program iz spletne strani Arduino, ki obrača servomotor glede na obračanje potenciometra:

```
void loop()
{ // reads the value of the potentiometer (value between 0 and 1023)
  val = analogRead(potpin);
  // scale it to use it with the servo (value between 0 and 180)
  val = map(val, 0, 1023, 0, 179);
  // sets the servo position according to the scaled value
```

```
myservo.write(val);  
// waits for the servo to get there  
delay(25);  
}
```

Da ne bi prišlo do slabih stikov, so bile vse žice, ki so bile povezane v razvojno ploščico, prisvajkane na bakrene nožice (Slika 3.5). Za spajkanje je bil uporabljen spajkalnik Parkside PLS 30, ki ima moč 30 W. Spajkalnik ima tudi večnamensko stojalo, ki je bilo pri spajkanju zelo uporabno.



Slika 3.4: Mikro servomotor TGY90S

## 3.2 Pritrditev senzorjev razdalje

Za pritrditev stranskih senzorjev razdalje so bili uporabljeni nosilci, ki so bili namenjeni pritrditvi zadnjega spojlerja. Na nosilce spojlerja je bilo pritrjeno posebno stojalo senzorja, namenjeno Sharpovim senzorjem razdalje (Slika 3.6).

Za bočne senzorje razdalje je ta umestitev idealna, ker segajo malo več kot 4 cm v notranjost avtomobila. To je pomembno, ker rezultati bližje od 4 cm niso zanesljivi zaradi samega delovanja senzorja (Slika 3.7 in Slika 3.8). Zaradi same oblike radijsko vodenega avtomobila ni bilo mogoče najti tako dobre rešitve tudi za sprednji in zadnji senzor razdalje. Ta dva sta privita na nosilce prednjih in zadnjih amortizerjev (Slika 3.9 in Slika 3.10). Težava pri taki pritrditvi je, da lahko pride do odčitavanja razdalj, ki se bodo napačno preračunale. To omejitev je bilo potrebno upoštevati pri nadaljnjem delu.



Slika 3.5: Spajkanje žic na bakrene nožice



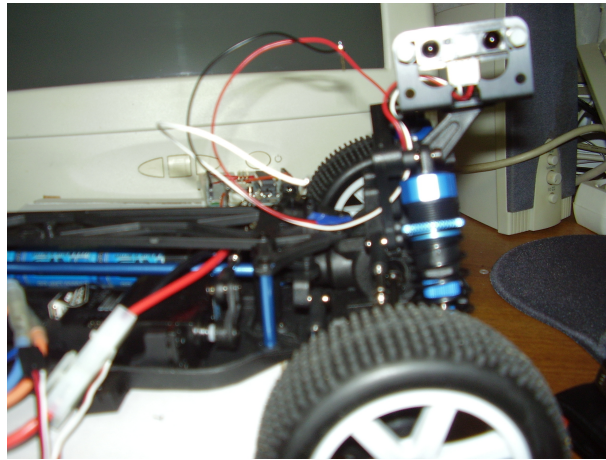
Slika 3.6: Stojalo senzorja razdalje

### 3.3 Pritrditev razvojne ploščice in mikrokrmilnika

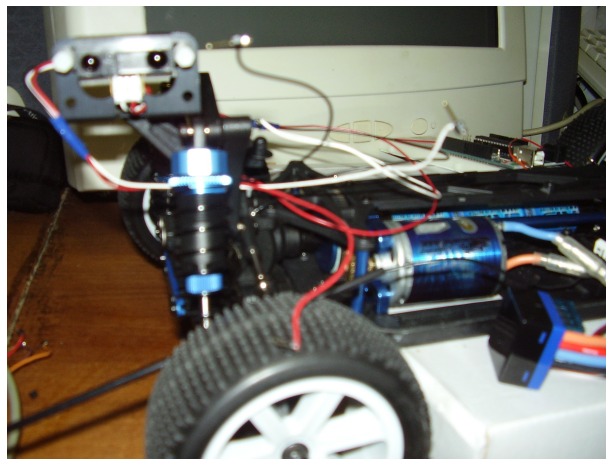
Razvojna ploščica in mikrokrmilnik sta bila nameščena na avtomobil tako, da je bil mikrokrmilnik postavljen na sredino razvojne ploščice. Ploščica je bila z izolirnim trakom prilepljena na nosilec, ki povezuje sprednji in zadnji del vozila (Slika 3.11). Na tem nosilcu je bil tudi nastavek za anteno, ki je bil odžagan, da se je lahko ploščica lepo pritrdila.

### 3.4 Pritrditev senzorja odbojnega koeficienta

Senzor odbojnega koeficienta deluje najbolje, če je približno pet milimetrov nad površino, na kateri merimo svetlobno odbojnost. Senzor je bil pritrjen preko nosilca za senzor razdalje na spodnji del modela avtomobila (Slika 3.13). Ker je senzor majhen in pritrjen na plastični vijak, je bil za zaščito le-tega pritrjen še en nosilec za senzorje, ki se po potrebi spusti in dvigne (Slika 3.12). Ta drugi nosilec nudi zaščito pri morebitnem trku.



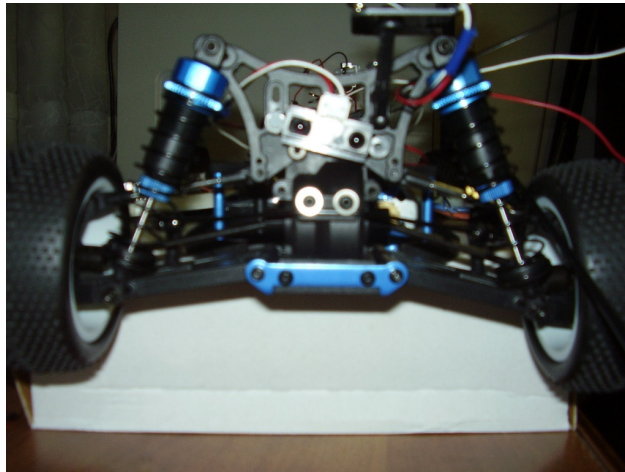
Slika 3.7: Pritrditev sprednjega bočnega sensorja razdalje



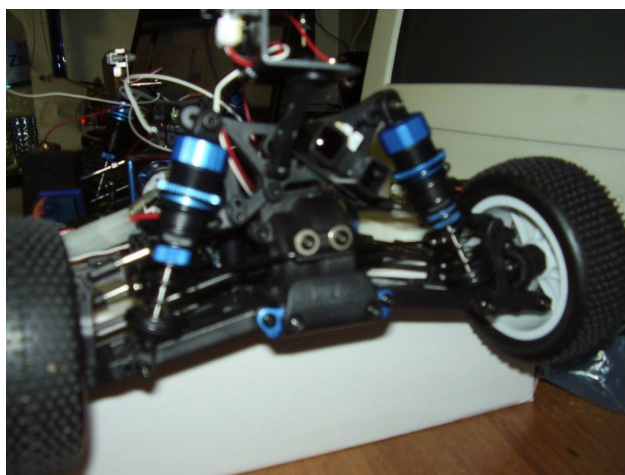
Slika 3.8: Pritrditev zadnjega bočnega sensorja razdalje

### 3.5 Pritrditev ostalih komponent

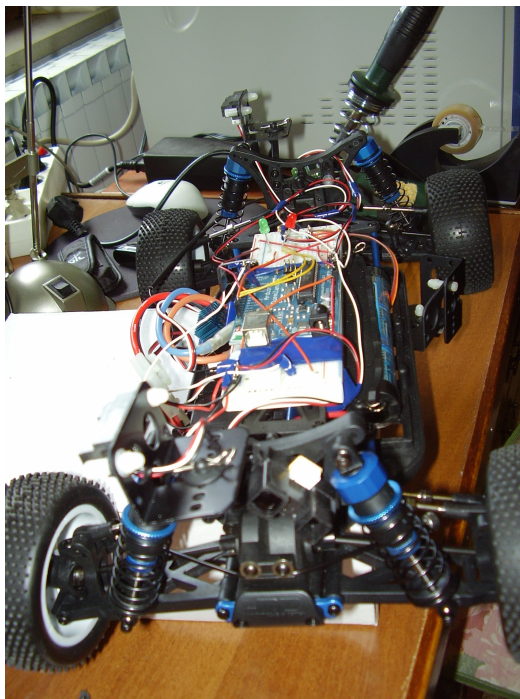
Za izbiro trenutno aktivne funkcije asistenčnega sistema je uporabljen linearni potenciometer (Slika 3.14). Glede na pozicijo potenciometra se izbere ustrezno funkcijo asistenčnega sistema. Za tako rešitev sem se odločili, ker je bil na vezju na razpolago samo še en analogni vhod.



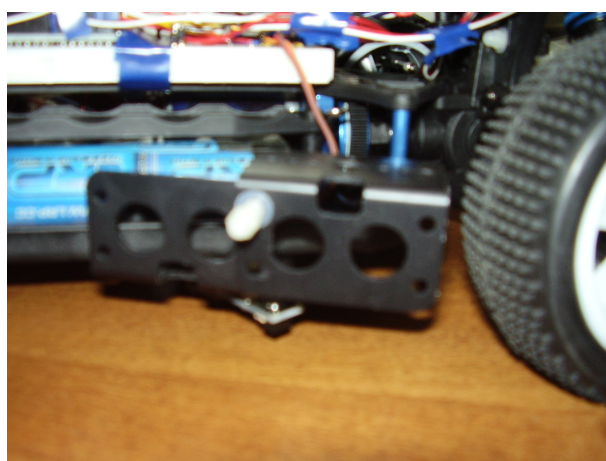
Slika 3.9: Pritrditev zadnjega senzorja razdalje



Slika 3.10: Pritrditev prednjega senzorja razdalje



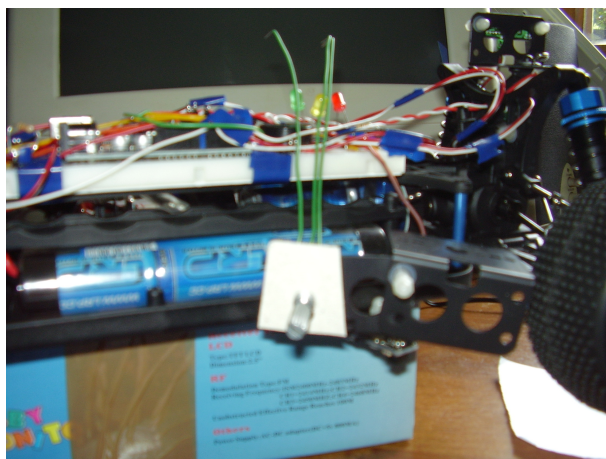
Slika 3.11: Pritrditev ploščice in mikrokrmilnika



Slika 3.12: Pritrditev senzorja odbojnega koeficienta



Slika 3.13: Senzor odbojnega koeficienta - pogled izpod vozila



Slika 3.14: Namestitev linearnega potenciometra

## Poglavje 4

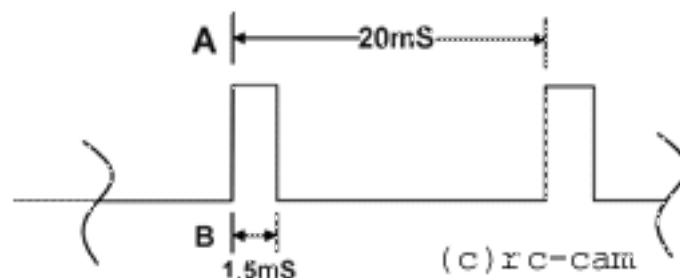
# Krmiljenje servomotorjev in interpretacija signalov iz senzorjev

### 4.1 Krmiljenje servomotorjev

Mikrokrmilnik bere ukaze iz radijskega sprejemnika in pošilja ukaze na servomotorje s pomočjo servo signala.

#### 4.1.1 Osnove delovanja servo krmiljenja

Servo signal je preprost digitalni impulz, ki je večino časa na logični ničli (0 V). Približno vsakih 20 ms gre v logično enko (3-6 VDC), potem pa hitro nazaj v ničlo (Slika 4.1). To ozko okno logične enke vzbudi pozornost servomotorja.



Slika 4.1: Servo signal

Sodobni servo mehanizmi imajo definirano sredinsko pozicijo pri 1,5 ms logične enke. Skrajno leva pozicija je definirana pri dolžini logične enke 2 ms, skrajno desna stran pa pri dolžini 1 ms. Vmesne pozicije so določene z dolžino pulza med 1ms in 2 ms.

### 4.1.2 Branje servo signala

Za branje s servo signala z Arduino mikrokrmilnikom je uporabljen ukaz:

```
pulseIn(pin, value, timeout).
```

Ukaz čaka, da preide signal v logično 0 ali 1 in začne meriti čas, dokler ne preide spet v logično 1 ali 0. Z odčitavanjem te vrednosti lahko določimo želeno smer in moč motorja. Ko delamo s servomotorji, je najlažje, da računamo s koti, zato vrednosti pretvorimo v kote s funkcijo `map`

```
map(value, fromLow, fromHigh, toLow, toHigh).
```

### 4.1.3 Pisanje servo signala

Za premikanje servomotorja in upravljanje E.S.C. (Electric motor Speed Controller) je uporabljena Arduino servo knjižnica [2]. Primer uporabe knjižnice:

```
#include <Servo.h>
```

```
Servo myservo;
```

```
void setup()
```

```
{
```

```
  myservo.attach(9);
```

```
  myservo.write(90); // set servo to mid-point
```

```
}
```

Za razliko od branja, kjer beremo dolžino pozitivnega stanja, tukaj vpisujemo kot. 90 stopinj je sredinska pozicija, 0 in 180 pa sta skrajni poziciji.

## 4.2 Interpretacija signalov iz senzorjev

Za branje signala iz senzorjev so pri vseh senzorjih uporabljena analogna vhodna vrata. Za določitev dolžine iz signala pri senzorjih razdalje so uporabljene že obstoječe funkcije. Za senzor odbojnega koeficienta pa je meja določena s kalibriranjem.

### 4.2.1 Branje razdalje senzorja Sharp GP2D120

Za odčitavanje razdalje pri senzorju GP2D120 je uporabljena metoda `read gp2d12 range` [3]. Metoda glede na podana vhodna vrata, iz katerih odčita vhodno napetost, vrne razdaljo do predmeta.

```
float read_gp2d12_range(byte pin)
{
    int tmp;

    tmp = analogRead(pin);
    if (tmp < 3)
        return -1; // invalid value

    return (6787.0 / ((float)tmp - 3.0)) - 4.0;
}
```

### 4.2.2 Branje razdalje senzorja Sharp GP2Y0A02

Podobno kot za GP2D120 se tudi za GP2Y0A02 odčitava vhodna napetost, iz katere se izračuna razdalja. Izračun razdalje iz napetosti je povzet iz [4].

```
float read_GP2Y0A_range(byte pin)
{
    int sensorValue = readSensor(pin);
    if (sensorValue < 3)
        return -1; // invalid value

    return (16667 / (sensorValue + 15)) - 10;
}
```

### 4.2.3 Branje razdalje senzorja Parallax QTI Light Sensor

Pri senzorju odbojnega koeficienta je pomembno le, ali je pod njim bela črta ali ne. Zato funkcija, ki interpretira podatke tega senzorja, vrača le podatek, ali je nad črto ali ni, in ne tudi vrednosti odbojnosti površine.

```
boolean read_LightSensor(int inPin)
{
    val = analogRead(inPin);
    delay(100);
    if (val < 100) { //reads white
        return true;
    }
    else if (val > 100) { //reads other
        return false;
    }
}
```

## Poglavje 5

# Izvedba naprednega asistenčnega sistema

### 5.1 Izvedba funkcije avtomatskega uravnavanja razdalje

Med delovanjem sistem bere podatke iz senzorja Sharp GP2D120, ki je nameščen na sprednjem delu vozila, in stalno meri razdaljo do spredaj vozečega vozila. Če vozilo stoji, se mu približa do določene varnostne razdalje. Da je funkcija sistema aktivna, mora upravitelj vozila na krmilni palici imeti vklopljeno dodajanje moči; če tega ni, se sistem samodejno izklopi. Glede na oddaljenost do spredaj vozečega vozila mikrokrmilnik računa moč, s katero se mora vrteti elektromotor. Bolj kot je vozilo oddaljeno, hitreje se lahko premika. Ko je sistem vklopljen, se bo vozilo ustavilo tudi ob nepredvideni oviri na cesti. Za zaviranje sta bili napisani dve funkciji:

```
/*
  breakNow
  when the distance is decreasing the car can must slow down
*/
void breakNow()
{
  setThrottle(0);
  digitalWrite(breakPin, HIGH);
}
```

Funkcija `breakNow()` postavi moč motorja na 0. To zaviranje je umirjeno, saj model avtomobila ustavlja le sila trenja. Tudi če se funkcijo neprestano kliče, ni težav, ker bo avtomobil obstal.

```

/*
  breakNow
  when the distance is decreasing the car can must slow down
*/
void breakNow()
{
  setThrottle(0);
  digitalWrite(breakPin, HIGH);
}

```

Funkcija `breakBackward()` postavi smer motorja na vzvratno smer s skoraj maksimalno močjo. To funkcijo se uporabi, ko senzor razdalje zazna hitro spreminjanje razdalje do spredaj vozečega vozila. Ko se kliče to funkcijo, začne vozilo sunkovito zavirati. Problem pri tem je, da bo pri predolgem klicu E.S.C. naprava interpretirala, kot da hoče upravljalec vozila z veliko hitrostjo peljati vzvratno. Zato se ta funkcija kliče samo ob zaznavi velike spremembe med prejšnjo odmerjeno in na novo odmerjeno razdaljo.

```

/*
  breakBackward
  break by setting full reverse
*/
void breakBackward()
{
  setThrottle(-80);
  digitalWrite(breakPin, HIGH);
  digitalWrite(ledPin, HIGH);
}

```

Če je funkcija avtomatskega uravnavanja razdalje vklopljena, se neprestano kliče funkcija `cruiseAsist()`. Ob vsakem ciklu se bere razdalja prednjega sensorja razdalje in na koncu beleži kot prejšnja razdalja. Glede na razliko vrednosti trenutne in prejšnje razdalje se kliče primerno funkcijo. Če je bila prejšnja razdalja manjša od sedanje ali je vrednost odčitane razdalje večja od 150 centimetrov, se kliče funkcijo `setMotorSpeed()`. Če je razlika od prejšnje razdalje do sedanje večja za več kot 6 centimetrov, se nevarno hitro približujemo spredaj vožečemu vozilu in se kliče funkcijo `breakBackward()`. V tretjem primeru, če ne velja nobena od prejšnjih trditev, mirno zaviramo z funkcijo `breakNow()`.

```

void cruiseAsist()
{
  readFrontDistance();
  readUserThrottle();
}

```

```

//if the user breaks or the distance value is invalid skip the sistem
if(userThrottle > 90 || checkValue2(frontCm) == -1)
{
    powerservo.write(userThrottle);
}
else
{
    if(frontCm >= prevDist || frontCm > 150)
    {
        digitalWrite(breakPin, LOW);
        setMotorSpeed(frontCm);
    }
    else if((prevDist-frontCm) > 6 && (prevDist-frontCm) < 20)
    {
        breakBackward();
    }
    else
    {
        breakNow();
        digitalWrite(ledPin, LOW);
    }
}

if(checkValue2(frontCm) != -1)
{
    prevDist = frontCm;
}
}

```

Za uravnavanje hitrosti motorja je bila napisana funkcija `setMotorSpeed()`, ki uravnava hitrost glede na podano dolžino v centimetrih do spredaj vozečega vozila. Če je razdalja med 20 in 30 centimetri, tudi če spredaj vozeče vozilo pospešuje, motorju ne dodajamo moči. Vzrok za to je varnostna razdalja in tudi to, da se dolžine, krajše od 20 centimetrov, ne odčitavajo pravilno. Ko je spredaj vozeče vozilo na razdalji med 30 in 40 centimetri, začnemo motorju počasi dodajati moč. Na razdalji od 40 do 200 centimetrov se dodaja moč glede na oddaljenost. Večja kot je oddaljenost, več moči lahko dodamo. Če je razdalja večja od 200 centimetrov, sistem ne deluje več in se upošteva voznikova odločitev. Da sistem sam nastavlja moč motorja, mora voznik držati tipko na krmilni palici v položaju za dodajanje moči. Ko upravitelj vozila tipko spusti ali premakne v položaj za zaviranje, sistem avtomatsko upošteva upravljalčeve nastavitve moči. Tak način se uporablja tudi v osebnih avtomobilih. Vedno ko uporabnik poseže v upravljanje, se nadzor preda njemu.

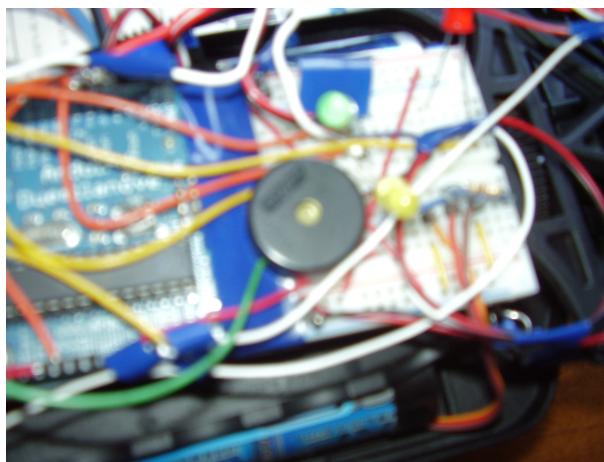
```
/*
  setMotorSpeed
  when the distance is increasing the car can go faster
*/
void setMotorSpeed(float cm)
{
  float motorSpeed = 0.0;

  if(cm > 20 && cm <= 30)
  {
    breakNow();
    motorSpeed=0.0;
  }
  else if(cm > 30 && cm <= 40)
  {
// the motor starts moving only at 5 and more
    motorSpeed = 0.6 * (cm - 30);
  }
  else if(cm > 40 && cm <= 200)
  {
    motorSpeed = 0.6; // the motor starts moving only at 5 and more
    motorSpeed = motorSpeed * (cm-30);
  }

  readUserThrottle();
  //the user must press the throttle to make the sistem work
  //if the sistem throttle is lower than the user use the sistem throttle
  if(userThrottle < 88 && (cm < 200) && autoThrottle > userThrottle)
  {
    setThrottle(motorSpeed);
  }
  else
  {
    powerservo.write(userThrottle);
  }
}
```

## 5.2 Izvedba funkcije asistencije voznega pasu

Funkcija asistencije deluje na osnovi zaznavanja bele površine na senzorju odbojnega koeficienta. Če zazna belo površino, se kličejo funkcije, ki popravijo smer gibanja vozila. Algoritem je zasnovan tako, da se različno korigira smer vožnje - odvisno je namreč od kota, pod katerim so bila obrnjena kolesa, ko je vozilo zapeljalo na belo površino. Večji kot je kot, močnejše se korigira smer. Na primer: če je vozilo zapeljalo na belo površino in so bila kolesa pod kotom 5 stopinj v levo, se bo smer korigirala na 5 stopinj desno, dokler vozilo ne pride spet na nebelo podlago. Če bi recimo bil kot 15 stopinj, bi se smer korigirala na 15 stopinj v nasprotno smer. Ker je ta sistem mišljen za uporabo na avtocestah, je običajno kot odmika majhen in se vozilo z majhno relativno spremembo smeri korigira na vozni pas. Sistem se kot v osebnih avtomobilih vklopi šele ob določeni hitrosti. Ko je funkcija sistema aktivna, je prižgana rumena led dioda na vezju (Slika 5.1). Če pride do prečkanja bele površine, se prižge tudi rdeča led dioda in preko piskača sproži zvočni signal.



Slika 5.1: Led diode in piskač

### 5.2.1 Podroben opis delovanja

Ko je sistem aktiven, gori rumena led dioda. Uravnavanje moči poteka vedno preko ukazov upravljalca vozila. Če je hitrost dovolj velika, je aktivna tudi asistenca smeri.

```
if(laneAsistActive)
{
    digitalWrite(LineLed, HIGH);
    manualThrottle();
    if(userThrottle < 88)
```

```
    {
        laneAssist();
    } else
    {
        manualDirection();
    }
}
```

Funkcija za asistenco smeri laneAssist():

```
void laneAssist()
{
    // check for the white line
    boolean isWhite = read_LightSensor(inPin);

    if (isWhite) //reads white
    {
        digitalWrite(ledPin, HIGH);
        frontout = true;
    }
    else //reads other
    {
        digitalWrite(ledPin, LOW);
        frontout = false;
    }

    //read the user input and turn the car
    readUserDirection();

    if(frontout == false && iscorr == false) //the car is on track
    {
        // sets the servo position according to the scaled value
        directionservo.write(userDirection);
        // waits for the servo to get there
        delay(15);
    } //the car is going out of the track, correct the direction
    else if(frontout == true && iscorr == false)
    {
        usrdir = directionservo.read();
        moddir = usrdir - 90;
        moddir = moddir+moddir;
    }
}
```

```

    int newdir = usrdir - moddir;
    directionservo.write(newdir);
    iscorr=true;
} //the car is still outside but the direction is correct
else if(frontout == true && iscorr == true)
{
    buzzNow();
} //the car is back on track, need to change the dirrection
else if(frontout == false && iscorr == true)
{
    directionservo.write(usrdir);
    iscorr=false;
}
}

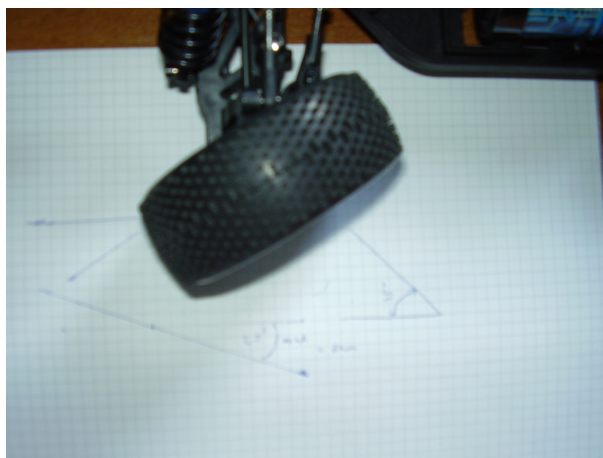
```

Na začetku funkcije se prebere vrednost senzorja odbojnega koeficienta. Če se model avtomobila nahaja nad belo površino, se vkopi rdeča led dioda, ki je na izhodu spremenljivke ledPin. Vrednost spremenljivke frontOut se postavi na logično enico. Tedaj se model avtomobila nahaja nad belo površino. Spremenljivka isCorr določa, ali je sistem že prevzel nadzor smeri in trenutno popravlja smer vožnje. Privzeta vrednost te spremenljivke je logična ničla. Prebere se smer, ki jo določa upravljalet vozila. Če je vozilo na nebeli podlagi in ni v stanju popravljanja smeri vozila, se ta smer uporabi in kolesa se postavijo v smer, ki jo je določil upravljalet vozila. Če je senzor odbojnega koeficienta zaznal, da je model avtomobila nad belo podlago in sistem še ni prevzel nadzora, se izračuna nova smer in sistem prevzame vodenje smeri. Če je senzor odbojnega koeficienta zaznal, da je model avtomobila nad belo podlago in je sistem že prevzel nadzor, se vklopi zvočno opozorilo. Če je senzor odbojnega koeficienta zaznal, da model avtomobila ni več nad belo podlago in sistem še vedno vodi vozilo, se spremenljivka isCorr postavi na logično ničlo. Tako upravljalet vozila dobi spet nadzor nad vozilom, ko le-to ni več nad belo podlago.

### 5.3 Izvedba funkcije avtomatskega parkiranja

Izvedba te funkcije asistenčnega sistema je bila najzahtevnejša. Sprva sem predpostavil, da bom uporabil kar algoritem, ki ga tudi sam uporabljam, ko bočno parkiramo, a na žalost sem takoj na začetku naletel na problem. Premikanje prednjih koles pri modelu avtomobila je zelo omejeno. Pri vseh osebnih avtomobilih se da sprednja kolesa zavrteti za 45 stopinj v levo oz. desno stran. Model avtomobila, ki je namenjen predvsem dirkanju po progi, ima kot pravi športni avtomobili možen le manjši zamik koles v levo oz. desno (Slika 5.2).

Pri merjenju je bilo ugotovljeno, da se da pri radijsko vodenem avtomobilu premakniti sprednja kolesa le za 33 stopinj v levo in 20 stopinj v desno. Algoritem, ki je bil napisan,

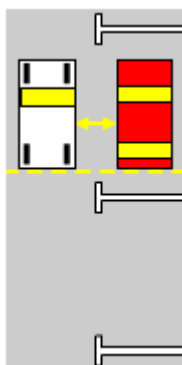


Slika 5.2: Zamik prednjih koles

še preden sem dobil model avtomobila, zato ni deloval. Zaradi te hibe potrebuje model več prostora za bočno parkiranje in optimalno bočno parkiranje žal ni izvedljivo. Celoten sistem parkiranja je razdeljen na dva dela: iskanje prostega mesta in bočno parkiranje. Izvedeni sta asistenca parkiranja in tudi avtomatsko parkiranje. V prvem poglavju je opis asistenc parkiranja. Avtomatsko parkiranje poleg smeri koles upravlja še gibanje radijsko vodenega avtomobila naprej in vzvratno.

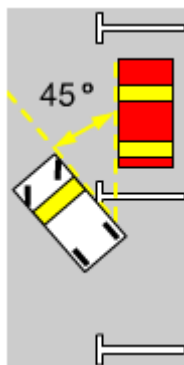
### 5.3.1 Bočno parkiranje, kot ga učijo v avtošoli

Opis je povzet iz [5]. Vozilo se ustavi vzporedno z vozilom, za katerim nameravamo parkirati okoli 60 cm od njega. Zadnji odbijači obeh vozil naj bodo v isti črti, če je naše vozilo po velikosti enako ali pa krajše kot vozilo, za katerim bomo parkirali. Če pa je naše vozilo daljše, potem naj bosta prednja odbijača v isti črti (Slika 5.3).



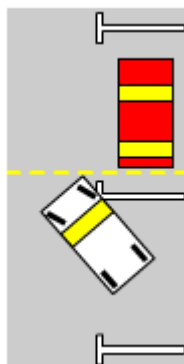
Slika 5.3: Prvi korak

Speljemo po polževo vzvratno in sukamo volan v desno, da pripeljemo vozilo v položaj kota 45 stopinj (Slika 5.4).



Slika 5.4: Drugi korak

Z vzvratno vožnjo nadaljujemo tako, da se desni konec odbijača na našem vozilu čim bolj približa zadnjemu delu stoječega avtomobila poleg nas (Slika 5.5).

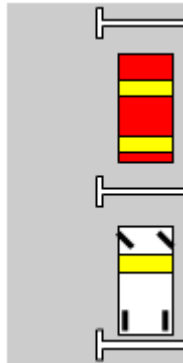


Slika 5.5: Tretji korak

V tej legi začnemo sukati volan v levo in se pomikati nazaj toliko časa, da se vozilo ustavi vzporedno z robom vozišča. Prednji kolesi naj ostaneta v poševni legi, kar nam omogoča, da pri speljevanju s parkirnega prostora izpeljemo po istih kolesnicah (Slika 5.6).

### 5.3.2 Iskanje prostega parkirnega mesta

Za iskanje prostega parkirnega mesta sta uporabljena bočna senzorja razdalje. Ko oba senzorja zaznata dovolj veliko razdaljo, se spremenljivka prostega parkirnega mesta postavi na pozitivno enico. Radijsko voden avtomobil se premika naprej, dokler s sprednjim bočnim senzorjem ne zazna konca prostega parkirnega prostora. Ko to zazna, začne zavirati tako, da ko zadnji bočni senzor zazna konec parkirnega mesta, je avto pravilno poravnan z



Slika 5.6: Četrty korak

vzporedno parkiranim vozilom (Slika 5.7). Pri avtomatskem parkiranju se model avtomobila sam počasi premika naprej, pri asistenci parkiranja pa upravljaec vozila določa hitrost. Algoritem za avtomatsko iskanje parkirnega mesta:

```

/*
 findEmptySpotA
 Function finds a empty parking spot en stops the
 car at the correct position to parallel park it
 */

void findEmptySpotA()
{
  boolean found = false;
  //adjust the direction to be exactly straight
  turnWheel(4);
  setThrottle(forwardSpeed); //move slowly forward

  // Serial.println("started moving");
  digitalWrite(greenLed, LOW);
  while(!found )
  {
    refreshBackLeftSensor();
    refreshFrontLeftSensor()
    if(backOk && frontOk)
    {
      found = true;
      digitalWrite(greenLed, HIGH);
    }
  }
}

```

```

// Serial.println("found spot");
while(frontOk)
{
    refreshFrontLeftSensor();
    //wait for the car to reach the end of the spot
}
int breakv = 80;
breakValue(breakv);
//the front of the car already reached the desired position
//move slowly so the car can stop at the righth position

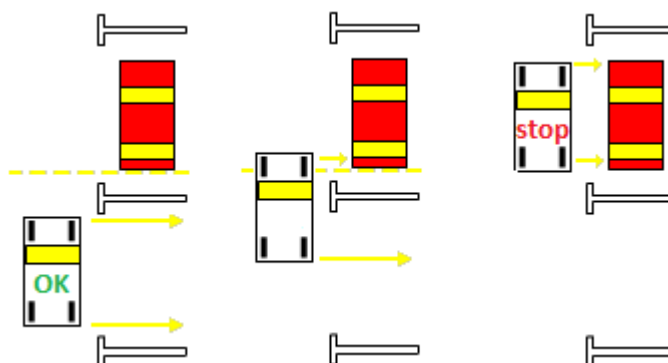
int i = 0;
while(backOk && i < 4)
{
    refreshBackLeftSensor();
    //to prevent the car to go backward

    if(breakv > 0)
    {
        breakv = breakv - 2;
        breakValue(breakv);
    }
    else
    {
        breakNow();
    }
    i++;
    //wait for the car to reach the end of the spot
}
// Serial.println("reached end of car");
delay(1000);
digitalWrite(greenLed, LOW);
}

```

### 5.3.3 Bočno parkiranje

Poleg bočnih senzorjev razdalje je za bočno parkiranje uporabljen tudi senzor razdalje, ki je pritrjen na zadnjem delu vozila. Sprva sem predvideval, da bom lahko z uporabo le zadnjega senzorja razdalje uspešno bočno parkiral, a se je izkazalo, da ta naloga še zdaleč ni tako enostavna. Dodaten problem je povzročalo to, da sprednjih koles ni mogoče



Slika 5.7: Iskanje prostega parkirnega mesta

obračati pod poljubnim kotom. Pri testiranju in razmišljanju o možni rešitvi sem prišel do spoznanja, da je bistvenega pomena točka, v kateri radijsko voden avtomobil obrne kolesa iz skrajno desne smeri v skrajno levo. Z uporabo samo zadnjega senzorja je bilo nemogoče točno določiti točko tega premika. Prelomna je bila ugotovitev, da lahko s pomočjo zadnjega in zadnjega bočnega senzorja določimo, kdaj je radijsko voden avtomobil pod kotom 45 stopinj glede na referenčno površino. Ker sta senzorja zelo blizu, lahko z zamikom pri odčitavanju vrednosti predpostavimo, da sta v isti točki. Tako se dobi referenčna točka, ko bosta oba senzorja razdalje odčitavala enako razdaljo in bo radijsko voden avtomobil točno pod kotom 45 stopinj relativno glede na referenčno površino. Kadar je radijsko voden avtomobil pod kotom 45 stopinj, se ustavi in kolesa se zasukajo v skrajno levo stran. Ustavi se, ko je vzporedno poravnana z referenčno površino. Torej takrat, ko je vrednost prednjega bočnega senzorja enaka zadnjemu bočnemu senzorju. Takrat se model avtomobila ustavi in parkiranje se zaključi (Slika 5.8).

Algoritem za avtomatsko bočno parkiranje:

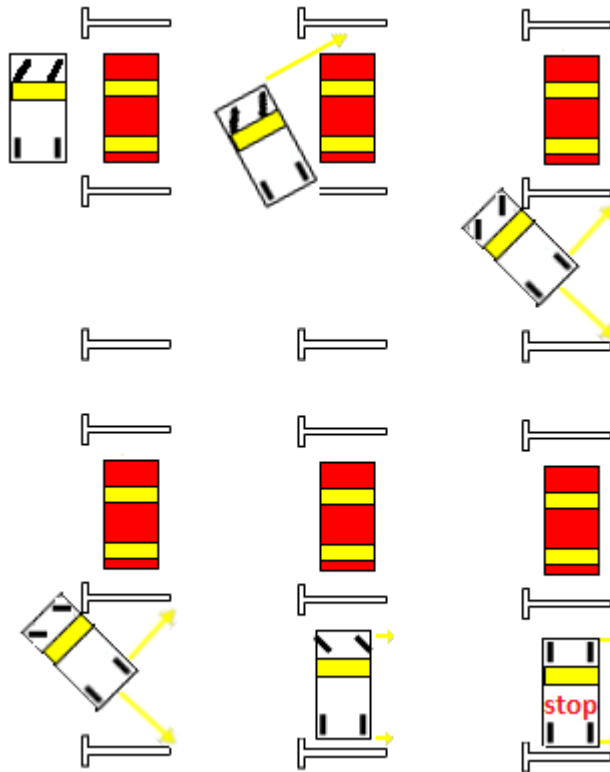
```

/*
 parkIntoA
 automatically parks the car into the parking place
*/
void parkIntoA()
{
 //move slowly backward
 setThrottle(backwardSpeed);

 //if there is a very large place in the back the sensor can sense
 //too far as an error

 //find the position when the car is in the 45 degrees position
 //the car actually cant move the wheels in that position,

```



Slika 5.8: Bočno parkiranje

```
//but the wheels
//are moved as far as possible in the desired direction
int diference45 = 5;

//wait the car moves away from the side parked car
while(!frontOk)
{
  refreshFrontLeftSensor();
}

//wait until the car is in a 45 degree position
while(diference45 > 15)
{
  backLeftCm = checkValue(read_gp2d12_range(backLeftSenPin));
  delay(50);
  backCm = checkValue(read_gp2d12_range(backSenPin));
  delay(50);
}
```

```
    difference45 = abs((backCm - 1) - (backLeftCm+5));
  }
  //car reached middle way into
  //turn the wheels 45 in the left direction
  turnWheel(-45);
  digitalWrite(LineLed, HIGH);
  digitalWrite(greenLed, HIGH);

  //wait for the car to stop in the middle position
  setThrottle(0);
  breakNow();
  delay(1000);
  digitalWrite(greenLed, LOW);
  // Serial.println("reached middle of parking spot");

  //move slowly backward
  setThrottle(backwardSpeed );

  // wait until the car is parallel to
  // the referenced surface
  int difrenceSide = 4;
  while(difrenceSide > 3)
  {

    refreshBackLeftSensor();
    refreshFrontLeftSensor();
    // manualThrottle();
    if((backLeftCm != -1 )&&(frontLeftCm != -1))
    {
      difrenceSide = abs(backLeftCm - frontLeftCm);
    }
  }
  turnWheel(0);
  // Serial.println("reached end of parking spot");

  setThrottle(0);
  digitalWrite(greenLed, HIGH);
}
```

## 5.4 Izvedba ročnega upravljanja vozila

Poleg uporabe asistenčnega sistema je radijsko voden avtomobil možno upravljati tudi brez asistence. Za razliko od upravljanja takega avtomobila brez mikrokrmilnika je za ročno upravljanje končna hitrost vozila precej omejena.

```

if(manualActive)
{
    digitalWrite(greenLed, HIGH);
    manualThrottle();
    manualDirection();
}

```

Ko je izbran program ročnega upravljanja vozila, je zelena led dioda stalno prižgana in se kličejo funkcije `manualThrottle` in `manualDirection`.

```

void manualDirection()
{
    readUserDirection();
    directionservo.write(userDirection);
}

```

```

void manualThrottle()
{
    readUserThrottle();
    powerservo.write(userThrottle);
}

```

Ti dve funkciji bereta, kaj sporoča krmilna palica in glede na to upravljajo servo izhodne signale. Obe imata tudi tako imenovani "failsafe"; če se avto predaleč oddalji od krmilne palice ali krmilni palici zmanjka baterij, se postavijo vrednosti v nevtralno stanje. Torej kolesa se poravnajo naravnost in elektromotor stoji na mestu.

```

/*
readUserDirection
Reads the user direction from the channel 1 on the receiver
*/
void readUserDirection()
{
    Channel1Value = pulseIn (2, HIGH, 20000); //read RC channel 1
    if (Channel1Value == 0) {
        Channel1Value = lastgood1;
    }
}

```

```
}
else
{
    lastgood1 = Channel1Value;
}
int maxDir = 1911;
int minDir = 1122;
//    Serial.println (Channel1Value);

//fail safe
if(Channel1Value < 1912 && Channel1Value > 1121)
{
    userDirection = map(Channel1Value, minDir, maxDir , 0, 179);
}
else
{
    userDirection = 90;
}
}
/* readUserThrottle
Reads the user direction from the channel 2 on the receiver
*/
void readUserThrottle()
{
    Channel2Value = pulseIn (4, HIGH, 20000); //read RC channel 1
    if (Channel2Value == 0) {
        Channel2Value = lastgood2;
    }
    else {
        lastgood2 = Channel2Value;
    }
    int maxDir = 2129;
    int minDir = 880;

    if(Channel2Value < 2130 && Channel2Value > 879)
    {
        //downgrade throttle
        userThrottle = map(Channel2Value, minDir, maxDir , 75, 104);
        // userThrottle = map(Channel2Value, minDir, maxDir , 0, 179);
    }
    else
```

```
{  
    userThrottle = 90;  
}  
}
```

## 5.5 Izbira funkcije asistenčnega sistema

Zaradi boljše odzivnosti in omejenosti mikroprocesorja je v nekem trenutku lahko aktivna le ena funkcija asistenčnega sistema. Izbira funkcije sistema poteka preko potenciometra, ki je podrobneje opisan v poglavju 3.5. Če želi upravljalec vozila preklopiti na drugo funkcijo asistenčnega sistema, mora zavrteti potenciometer tako, da kaže na želeno številko funkcije asistenčnega sistema in na mikrokrmilniku stisne gumb za "reset".

1. Asistenca za ohranjanje smeri
2. Avtomatsko parkiranje
3. Asistenca parkiranja
4. Avtomatsko uravnavanja razdalje
5. Ročno upavljanje vozila

# Poglavje 6

## Rezultati

Mikrokrmilnik uspešno komunicira z vsemi priklopljenimi senzorji. Do težav pride le, če zaporedoma beremo vse senzorje enega za drugim. Takrat se že pozna slabša odzivnost pri izvajanju asistenčnega sistema, ker porabi tako branje veliko procesorskega časa. Pritrditev samega vezja in bakrenih nožic v razvojno ploščico je dobro izpeljana in do slabih stikov med vezjem in senzorji ne prihaja, tudi pri manjših trkih ne.

### 6.1 Asistenca uravnavanja razdalje

Asistenca deluje solidno, težave se pojavijo le, če pride predmet pred vozilo bližje kot 20 cm. Kot je omenjeno v poglavju 3.2, sprednji senzor razdalje napačno interpretira razdalje krajše od 20 cm.

### 6.2 Asistenca za ohranjanje smeri

Asistenca deluje dobro do mejnih pogojev. Če zapeljemo čez črto pod prevelikim kotom, asistenca ne bo uspela pravočasno popraviti smeri vozila. Težava je tudi z odzivnostjo sistema. Zato je bila pri končnem testiranju uporabljena precej debelejša bela cestna oznaka, kot bi bila, če bi bila v merilu radijsko vodenega avtomobila.

### 6.3 Asistenca parkiranja

Funkcija deluje dobro, če upravljalec vozila poskrbi, da pelje model avtomobila dovolj počasi. V nasprotnem primeru zavorna razdalja vpliva na pravilno postavitve vozila pred bočnim parkiranjem. Model avtomobila je sposoben močnih pospeškov, zato so zavorne poti lahko kar dolge.

## 6.4 Avtomatsko parkiranje

Funkcija deluje le ob idealnih pogojih. Dodatne težave povzročajo neenakomerno polna akumulatorska baterija. Radijsko voden avtomobil je težko peljati počasi, ker ni namenjen temu. Zato je do dana na tem sistemu tudi ročno kalibracija, ki se izvede pri izbiri programa s potenciometrom.

## 6.5 Zaključek

Z rezultati sem zadovoljen, najbolj z izvedbo funkcije avtomatskega parkiranja. Čeprav deluje le v idealih primerih, je korak naprej od asistence; lahko bi rekli, da je to avtonomna vožnja, saj upravljalcu vozila ni potrebno narediti ničesar. Posnetek tega sistema si lahko ogledate na [8]. Veliko težav je povzročal sam radijsko voden avtomobil, saj je njegova celotna zasnova prilagojena visokim hitrostim. Veliko lažje bi bilo, če bi uporabil enega od robotskih kit kompletov, ki bi ga krmilil z mikrokrmilnikom. Slabost tega bi bila, da bi bila izvedba in simulacija sistemov veliko manj podobna realnosti. Ena od možnosti razširitve diplomske naloge bi bila dejanska izvedba enega od sistemov na osebni avtomobil. Seveda brez poseganja v samo vožnjo, ampak samo kot opozorilni sistem.

# Slike

1.1	Radarski senzor . . . . .	6
1.2	Oddaljenost spredaj vozečega vozila . . . . .	6
1.3	Kamera v predelu notranjega vzvratnega ogledala . . . . .	7
1.4	Aktivna asistenca za ohranjanje smeri . . . . .	8
1.5	Domet senzorjev . . . . .	9
2.1	S10 Blast BX . . . . .	10
2.2	Krmilna palica . . . . .	11
2.3	Sprejemnik . . . . .	11
2.4	Arduino Duemilanove . . . . .	12
2.5	Arduino kit komplet . . . . .	13
2.6	Princip delovanja IR senzorja razdalje . . . . .	14
2.7	Sharp GP2D120 . . . . .	14
2.8	Sprememba napetosti glede na oddaljenost . . . . .	15
2.9	Sharp GP2Y0A02 . . . . .	15
2.10	Krivulja izhodne napetosti glede na razdaljo . . . . .	16
2.11	Parallax QTI Light Sensor . . . . .	16
2.12	Senzor odbojnega koeficienta z dodanim uporom . . . . .	16
3.1	Bela podlaga ni zaznana . . . . .	17
3.2	Bela podlaga je zaznana . . . . .	18
3.3	Testiranje senzorjev . . . . .	18
3.4	Mikro servomotor TGY90S . . . . .	19
3.5	Spajkanje žic na bakrene nožice . . . . .	20
3.6	Stojalo senzorja razdalje . . . . .	20
3.7	Pritrditev sprednjega bočnega senzorja razdalje . . . . .	21
3.8	Pritrditev zadnjega bočnega senzorja razdalje . . . . .	21
3.9	Pritrditev zadnjega senzorja razdalje . . . . .	22
3.10	Pritrditev prednjega senzorja razdalje . . . . .	22
3.11	Pritrditev ploščice in mikrokrmilnika . . . . .	23
3.12	Pritrditev senzorja odbojnega koeficienta . . . . .	23
3.13	Senzor odbojnega koeficienta - pogled izpod vozila . . . . .	24

3.14	Namestitev linearnega potenciometra . . . . .	24
4.1	Servo signal . . . . .	25
5.1	Led diode in piskač . . . . .	32
5.2	Zamik prednjih koles . . . . .	35
5.3	Prvi korak . . . . .	35
5.4	Drugi korak . . . . .	36
5.5	Tretji korak . . . . .	36
5.6	Četrty korak . . . . .	37
5.7	Iskanje prostega parkirnega mesta . . . . .	39
5.8	Bočno parkiranje . . . . .	40

# Literatura

- [1] <http://www.nhtsa.gov/>
- [2] <http://www.arduino.cc/en/Reference/Servo>
- [3] <http://www.arduino.cc/playground/Main/ReadGp2d12Range>
- [4] <http://www.semi-blog.com/wp-content/uploads/2010/03/DistanceSensor.c>
- [5] <http://www.tecajcpp.com/cpp/bocno.php>
- [6] [http://www.volkswagen.si/ve\\_o\\_volkswagnu/inovacije\\_tehnika/asisten\\_ni\\_sistemi/](http://www.volkswagen.si/ve_o_volkswagnu/inovacije_tehnika/asisten_ni_sistemi/)
- [7] <http://www.smakrobot.si/tag/mikrokrmilnik/>
- [8] <http://www.youtube.com/watch?v=whHbdcnFbQY>
- [9] R. Massimo Banzi, *Getting Started with Arduino* , O'Reilly Media / Make, Oct 2008.
- [10] Jonathan Oser, Hugh Blemings, *Practical Arduino: Cool Projects for Open Source Hardware* , Apress, Dec 2009