Parkirni senzor temeljen na AVR mikroupravljaču

Šimunović, Marin

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

PARKIRNI SENZOR TEMELJEN NA AVR MIKROUPRAVLJAČU

Diplomski rad

Marin Šimunović

Osijek, 2018.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac D1: Obrazac za imenovanje Povjerenstva za obranu diplomskog rada

Osijek, 11.07.2018.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za obranu diplomskog rada

Ime i prezime studenta:	Marin Šimunović
Studij, smjer:	Diplomski sveučilišni studij Elektrotehnika, smjer Komunikacije i informatika'
Mat. br. studenta, godina upisa:	D1072, 24.09.2017.
OIB studenta:	74016594434
Mentor:	Doc.dr.sc. Davor Vinko
Sumentor:	
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Doc.dr.sc. Vanja Mandrić-Radivojević
Član Povjerenstva:	Izv. prof. dr. sc. Krešimir Grgić
Naslov diplomskog rada:	Parkirni senzor temeljen na AVR mikroupravljaču
Znanstvena grana rada:	**Elektronika (zn. polje elektrotehnika)**
Zadatak diplomskog rada:	Temu rezervirao: Marin Šimunović Zadatak diplomskog rada je razvoj, izrada i evaluacija parkirnog senzora temeljenog na AVR mikroupravljaču. Za više informacija javiti se mentoru: davor.vinko@ferit.hr
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	11.07.2018.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 24.07.2018.

Ime i prezime studenta:	Marin Šimunović
Studij:	Diplomski sveučilišni studij Elektrotehnika, smjer Komunikacije i informatika'
Mat. br. studenta, godina upisa:	D1072, 24.09.2017.
Ephorus podudaranje [%]:	7%

Ovom izjavom izjavljujem da je rad pod nazivom**: Parkirni senzor temeljen na AVR mikroupravljaču**

izrađen pod vodstvom mentora Doc.dr.sc. Davor Vinko

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

Zahvala:

Zahvaljujem se svom mentoru doc.dr.sc. Davoru Vinku na savjetima, pomoći i vodstvu prilikom izrade ovog diplomskog rada.

Veliku zahvalnost dugujem svojoj zaručnici na njenoj podršci tijekom cijelog studiranja, na strpljivosti, savjetima i pomoći kada mi je bilo najteže.

Veliko hvala roditeljima koji su mi omogućili studiranje, davali neizmjernu podršku i vjerovali u mene.

Na kraju hvala svim kolegama i prijateljima bez kojih studiranje ne bi bilo zabavno.

Sadržaj

1. UVOD

U 21. stoljeću svjedoci smo sve većem i bržem razvoju tehnologija, a tako i sve većem porastu stanovništva. S tim, došla je i potreba za vlastitim prijevozom, odnosno automobilom. Kako bi se zadovoljile potrebe za automobilima, a i isto tako i želje stanovništva za sve većim i boljim inovacijama, proizvodi se sve više različitih automobila s različitim karakteristikama i mogućnostima. To dovodi do zasićenosti i prenapučenosti gradova, te stvara probleme pri pronalaska parkirnog mjesta, a isto tako i pri samom parkiranju. Kako bi se olakšalo parkiranje, osmišljeni su sustavi za pomoć pri parkiranju kao jedna od karakteristika današnjih automobila, odnosno parking senzori.

Na početku mogli smo ih pronaći samo na većim i luksuznijim automobilima, a danas se mogu pronaći na većini novijih automobila ili se mogu odabrati kao dodatna oprema pri kupovini automobila.

Cilj ovog rada je napraviti sklop koji će mjeriti udaljenost i ispisivati ju na LCD zaslon uz dodatne različite oblike, koji će predstavljati graf udaljenosti. Sklop je temeljen na Atmega16 mikroupravljaču. Senzor udaljenosti je ultrazvučni, koji radi na principu da pošalje ultrazvučni val i mjeri njegovo vrijeme povratka. Iz tih vrijednosti se pomoću programske podrške i mikroupravljača računa udaljenost od objekta. Osim toga, sklop sadrži i zujalicu, koja svojim zujanjem uz ispis udaljenosti daje veću predodžbu vozaču automobila na kojoj se udaljenosti nalazi. Povezivanjem svih navedenih komponenti i uz programsko rješenje, napravljen je parking senzor.

Rad je podijeljen na hardverski i softverski dio, te su to najvažniji dijelovi. Osim njih, tu je i poglavlje u kojem se opisuje konačan sklop, te na kraju poglavlje sa mjerenjima i analizama istih.

Hardverski dio rada opisuje sve komponente korištene kako bi se dobio sklop koji mjeri udaljenosti i ispisuje istu na zaslon. To su mikroupravljač, LCD zaslon, zujalica i senzor udaljenosti. Opisani su njihovi načini rada, karakteristike i mogućnosti.

U softverskom dijelu opisano je cijelo programsko rješenje odnosno način rada parking senzora. Osim toga, tu se nalaze i postavke mikroupravljača te su opisani svi periferni uređaji koje koristi mikroupravljač za mjerenje udaljenosti.

Na kraju su napravljena mjerenja sa parking senzorom. Pomoću njih je opisano ponašanje senzora u različitim uvjetima, što je kroz rezultate mjerenja i potvrđeno. Prikazane su njegove mogućnosti i preciznost ovisno o različitim temperaturama i različitim vrstama objekata od kojih se val odbija.

1.1. Zadatak diplomskog rada

Zadatak ovog rada je napraviti sustav za pomoć pri parkiranju odnosno parking senzor temeljen na Atmega16 mikroupravljaču. Potrebno je povezati mikroupravljač, ultrazvučni modul, zujalicu i LCD u jedan sklop koji uz programsku podršku mjeri udaljenost od objekta, prikazuje ju na zaslonu i upozorava vozača.

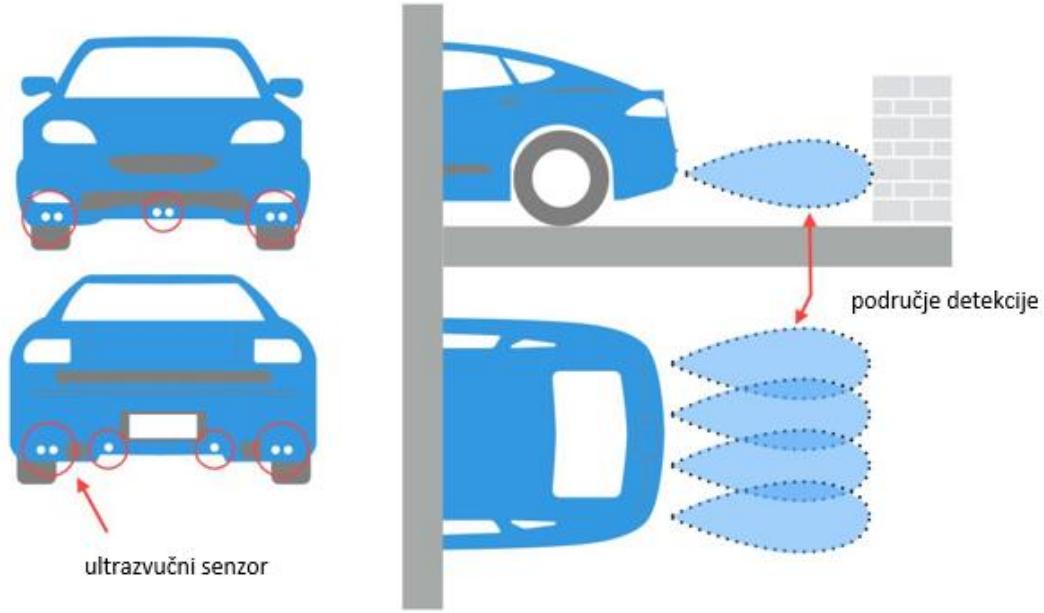
2. HARDVERSKI DIO RADA

2.1. Općenito o parking senzorima

Parking senzori su senzori udaljenosti za cestovna vozila namijenjena upozoravanju vozača na prepreke pri parkiranju. Ova tehnologija omogućava parkiranje vozila sa više samouvjerenosti i općenito čini vožnju u nazad sigurnijom. Oni otkrivaju objekte na putu vozila i emitiraju zvuk ili zvučni signal koji vam omogućuje da znate da se morate zaustaviti i provjeriti iza sebe prije nastavka. Proizvođači ih obično instaliraju na odbojnike vozila. Dakle, ovaj sustav je vrsta sustava pomoći vozaču pri parkiranju. Uz sve veće dimenzije vozila i smanjenje parkirnih mjesta, ovi senzori postaju sve popularniji. Parking senzori su označeni kod različitih proizvođača vozila pod različitim nazivima, kao što su *Park Distance Control, Park Assist, Parktronic* ili *EPS,* ali dolaze samo u dvije varijante: ultrazvučni i elektromagnetski senzori [1], [2].

2.1.1. Ultrazvučni sustavi

Ovi sustavi imaju ultrazvučne detektore udaljenosti koji služe za mjerenje udaljenosti od objekata u blizini, pomoću senzora koji se nalaze u prednjem i / ili stražnjem odbojniku. Senzori emitiraju zvučne impulse, s kontrolnom jedinicom koja mjeri povratni interval svakog reflektiranog signala i izračunava udaljenosti objekta. Sustav upozorava vozača s zvučnim tonovima, frekvencijom koja označava udaljenost objekta, s bržim tonovima koji ukazuju na veću blizinu i kontinuiranim tonovima koji označavaju minimalnu unaprijed definiranu udaljenost. Sustavi također mogu uključivati vizualna pomagala, kao što su LED ili LCD očitanja za označavanje udaljenosti objekta. Vozilo može sadržavati piktogram vozila na zaslonu automobila, s prikazom obližnjih objekata kao blokova u boji. Stražnji senzori se aktiviraju kada je odabran stupanj prijenosa za vožnju unatrag i deaktivirani čim se odabere drugi stupanj prijenosa (vožnja naprijed). Prednji senzori se mogu ručno aktivirati i deaktivirati automatski kada vozilo dosegne unaprijed određenu brzinu, kako bi izbjegli buduća upozorenja o smetnjama. Budući da se ovaj sustav oslanja na korištenje zvučnih valova, postoje slučajevi kada senzor ne može ispravno otkriti objekte iza vas. Sustav ne može otkriti ravne objekte ili objekte nedovoljno velike da bi reflektirali zvuk. Predmeti s ravnim površinama pod okomitim kutom mogu vraćati povratne zvučne valove dalje od senzora, čime se sprječava detekcija objekta. Mekani objekti s jakom apsorpcijom zvuka mogu imati slabiju detekciju, npr. vuna. Također, budući da ultrazvučni sustav radi pomoću četiri do šest pojedinačnih senzora postavljenih na vanjsku stranu odbojnika automobila, postoje slučajevi u kojima će prljavština na samim senzorima pridonijeti nedostatku detekcije [3].



Slika 2.1. *Ultrazvučni sustav parking senzora* [4]

2.1.2. Elektromagnetski sustavi

Kao što je naznačeno imenom, ovi senzori rade pomoću elektromagnetskih valova. Traka koja čini primopredajnik stvara eliptično polje iza automobila, a predmeti koji zadovoljavaju određeni zahtjev za masom i ometaju to polje, utječu na detektor koji prikuplja promjene napona i šalje podatke na računalo u automobilu. Računalo analizira podatke kako bi utvrdio udaljenost od objekta, a zatim upozorava vozača nizom tonova čija glasnoća se povećava ili ubrzava dok se vozilo približava objektu iza njega. Većina, ako ne i svi, od ovih senzora se postavljaju u unutrašnjosti odbojnika vašeg vozila, tako da nema problema s prljavštinom koja ometa sposobnost detekcije sustava [2].

Slika 2.2. *Elektromagnetski sustav parking senzora* [5]

2.1.3. Rad parking senzora

Bilo da senzori rade na ultrazvučnom ili elektromagnetskom principu, kada vozač prebaci u stupanj prijenosa za vožnju unatrag, stražnji parking senzori automatski se aktiviraju i šalju ultrazvučne ili elektromagnetske valove. Kada ti valovi udare od neki objekt ili kada objekt utječe na promjenu elektromagnetskog polja, detektira se prisutnost nekog objekta u blizini vozila. Nakon što se vozilo približi bilo kojem objektu, sustav će upozoriti vozača zvučnim tonom ili vizualnim signalom na nadzornoj ploči. Kako se vozilo približava prema objektu, jačina signala alarma se pojačava, te signalizira vozaču da zaustavi vozilo.

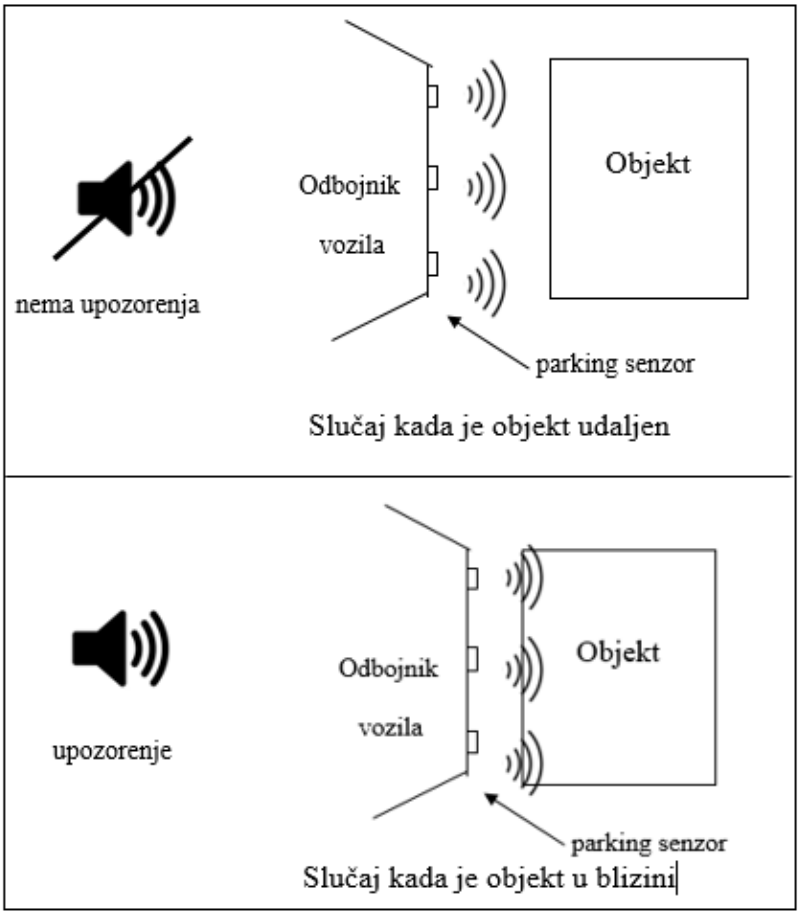
Prednosti parking senzora:

- Smanjenje slijepih točki oko vozila olakšava parkiranje u uskim prostorima
- Sustav smanjuje zamor vozača tijekom parkiranja vozila
- Poboljšana percepcija područja iza vozila smanjuje šanse za oštećenje vozila ili drugih objekata u blizini

Nedostaci parking senzora:

- Ravne predmete ili vertikalne objekte koji su vrlo tanki teško je detektirati ultrazvučnim senzorom
- Ultrazvučni senzor ne prepoznaje ispravno ako se na njegovoj površini nalaze blato, snijeg ili prljavština
- Elektromagnetski senzori su prilično skupi u odnosu na ultrazvučne

- Elektromagnetski senzori detektiraju objekt samo kada se vozilo kreće [1]



Slika 2.3. *Rad parking senzora*

2.2. Senzori udaljenosti u elektronici

Senzor je sofisticirani uređaj koji mjeri fizičku količinu poput brzine ili tlaka i pretvara ga u signal koji se može izmjeriti električno. Senzori se temelje na nekoliko načela rada i vrstama mjerenja. U ovom slučaju gotovo sve vrste senzora emitiraju signale i mjere refleksiju kako bi izvršili mjerenje [6].

Senzori udaljenosti - koristi se nekoliko senzorskih tehnologija za izradu senzora udaljenosti:

- Ultrazvučni
- Kapacitivni
- Induktivni

- Fotoelektrični
- Elektromagnetski

2.2.1. Ultrazvučni senzor

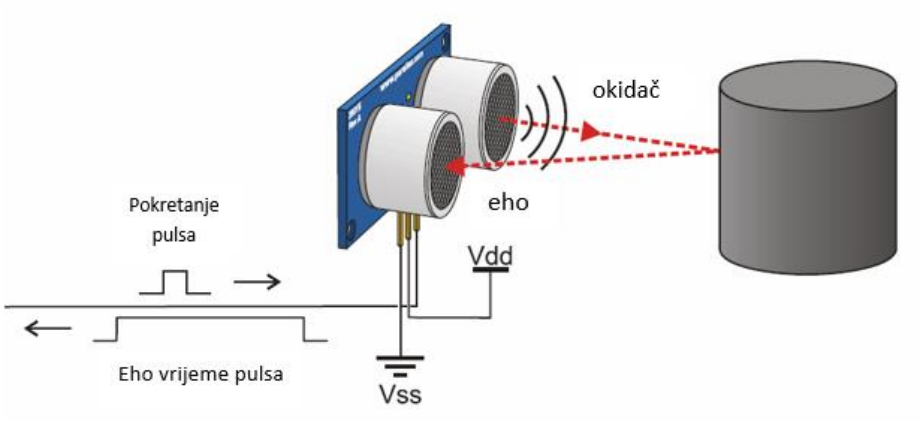
Ultrazvučni senzori udaljenosti se koriste u mnogim automatiziranim proizvodnim procesima. Oni koriste zvučne valove za otkrivanje objekata, tako da boja i transparentnost ne utječu na njih. Ovi senzori su dizajnirani za generiranje visokofrekventnih zvučnih valova i primanje eho odraženog od cilja. Koriste se u širokom rasponu aplikacija kao što su mjerenje udaljenosti, kontrolu razine tekućine i sl.. Vrlo su korisni kada nije važno otkrivanje boja, površinske teksture ili prozirnosti.

Prednosti:

- izlazna vrijednost je linearna s udaljenošću između senzora i cilja
- odziv senzora ne ovisi o bojama, prozirnosti objekata, svojstvima optičke refleksije ili površinskoj teksturi objekta
- ovi senzori su dizajnirani za detekciju bez kontakta
- precizna detekcija čak i od malih objekata
- ultrazvučni senzori mogu raditi u kritičnim uvjetima kao što su prljavština i prašina

Nedostaci:

- ultrazvučni senzori moraju biti usmjereni na površinu visoke gustoće za dobre rezultate. Meka površina poput pjene i platna ima malu gustoću i apsorbira zvučne valove koje emitira senzor
- mogu imati lažne reakcije za neke glasne zvukove
- ultrazvučni senzori imaju manje vrijeme odziva od ostalih vrsta senzora
- ultrazvučni senzor ima minimalnu udaljenost detekcije, koja treba uzeti u obzir prilikom odabira senzora
- neke promjene u okolišu mogu utjecati na odgovor senzora (temperatura, vlažnost, tlak itd.) [6]



Slika 2.4. *Ultrazvučni senzor* [7]

2.2.2. Infracrveni senzor

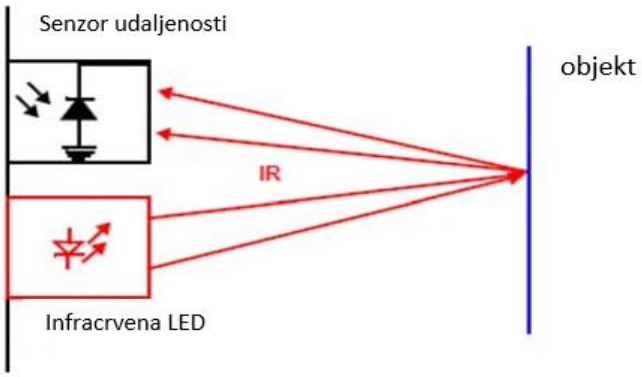
Infracrveni senzor mjeri IR svjetlo koje se prenosi u okruženju kako bi pronašli objekte pomoću IR LED. Ova vrsta senzora je vrlo popularna u navigaciji za izbjegavanje objekata, udaljenosti i sl.. Senzor je vrlo osjetljiv na IR svjetla i sunčevu svjetlost, a to je glavni razlog što se IR senzor koristi s velikom preciznošću u prostorima s niskim svjetlom.

Prednosti:

- infracrveni senzori mogu otkriti infracrvenu svjetlost preko velikog područja
- rade u realnom vremenu
- IR senzor koristi nevidljivu svjetlost za detekciju
- jeftini senzori

Nedostaci:

- senzor je vrlo osjetljiv na IR svjetla i sunčevu svjetlost
- ima manju osjetljivost na tamnije boje poput crne, jer crna boja upija IR zračenje [6]



Slika 2.5. *Infracrveni senzor* [8]

2.2.3. Fotoelektrični senzor

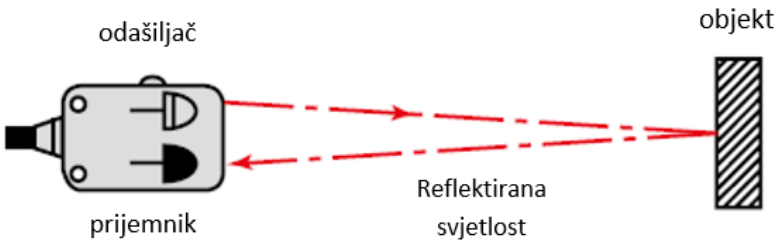
Fotoelektrični senzori su uređaji koji se koriste za otkrivanje udaljenosti, odsutnosti ili prisutnosti objekta pomoću svjetlosnog odašiljača i fotoelektričnog prijemnika. Fotoelektrični senzor sastoji se prvenstveno od predajnika za emitiranje svjetla i prijemnika za primanje svjetla. Kada se emitirano svjetlo prekine ili reflektira od objekt, mijenja se količinu svjetla koja stiže na prijemnik. Prijemnik otkriva ovu promjenu i pretvara ga u električni izlaz. Izvor svjetlosti za većinu fotoelektričnih senzora je infracrvena ili vidljiva svjetlost (obično crvena ili zelena / plava za prepoznavanje boja).

Prednosti:

- Očitavanje za velike udaljenosti
- Brzo vrijeme odgovora
- Visoka rezolucija
- Jednostavno podešavanje
- Identifikacija boja

Nedostaci:

- Tijekom vremena objektiv se kontaminira
- Utjecaj na domet senzora zbog boje i reflektivnosti cilja
- Uređaj zahtijeva odašiljač i prijemnik na odvojenim mjestima, što ga čini složenim [9]



Slika 2.6. *Fotoelektrični senzor* [10]

2.3. Senzor izabran za temu rada

Parking senzor temeljen na AVR mikroupravljaču za krajnji cilj mora biti optimiziran, ispunjavati zadatke potrebne za detekciju udaljenosti i mora biti u skladu sa specifikacijama. Kako smo u temi iznad spomenuli većinu vrsta senzora za detekciju udaljenosti, odabir pravog za ovaj projekt je zapravo proces u kojem je potrebno odabrati senzor s obzirom na potrebe i mogućnosti rada parking senzora.

Kada koristimo riječ objekt, to nas upućuje na objekt poput zida, drveta, ili čak čovjeka koji se može nalaziti iza vozila koje posjeduje senzor udaljenosti. U nekoliko riječi, senzor mora biti odabran u skladu s ciljevima projekta, veličine, oblika i dometa potrebnog da zadovolji zadane ciljeve. Međutim, teško je definirati i odabrati najbolji senzor jer izvedba i preciznost ovise o mnogim čimbenicima [6].

2.3.1. Kriteriji za odabir pravog senzora

- **Vrsta senzora -** prisutnost objekta može se detektirati sa senzorima udaljenosti, a postoji nekoliko vrsta senzorskih tehnologija, uključujući ultrazvučne senzore, kapacitivne, fotoelektrične, induktivne ili elektromagnetske.
- **Preciznost -** preciznost je vrlo važna u otkrivanju i praćenju objekata, a korisno je odabrati senzore s vrijednostima točnosti između željenih mjernih granica
- **Rezolucija** - visoka rezolucija može otkriti najmanju promjenu u položaju objekta
- **Domet** - uključuje odabir senzora na temelju mjernih granica i uspoređivanje sa željenim dometom detekcije parking senzora
- **Upravljačko sučelje** – moraju se poznavati vrste senzora. Velik broj senzora su 4-pinski DC tipa, ali postoji više vrsta, uključujući 3-pinski DC, 2-pinski DC ili 2-pinski AC / DC
- **Stanje u okolini rada** - svaki senzor ima svoje operativne granice, a obično to su temperatura i vlažnost
- **Cijena** - ovisno o ograničenju proračuna projekta, potrebno je odabrati senzor ili senzore koji se mogu koristiti za izradu parking senzora [6]

2.3.2. Ultrazvučni senzor udaljenosti *HC-SR04*

Uzimajući u obzir navedene kriterije za odabir senzora udaljenosti, odabrani senzor za projekt je ultrazvučni senzor *HC-SR04*. Kako je primjena parking senzora i instalacija ponajviše na automobilima, njihov zadatak je detektirati objekt koji je u visini auta, bilo da je to zid pri parkiranju, vozilo koje se nalazi na slijedećem parking mjestu ili čak čovjek koji prolazi iza vozila dok izlazite sa parking mjesta, za takve zadatke je sasvim dovoljan i primjenjiv ultrazvučni tip senzora. Također ovaj senzor je vrlo precizan, može detektirati i male promjene u položaju objekta, a ima domet i do 4 metra, što je dovoljno kao primjena parking senzora. Senzor je DC tipa sa 4 pina, a to su: napajanje, uzemljenje, okidač i eho. Cijena senzora je vrlo mala, te je i samim time senzor vrlo pogodan za ovaj projekt, jer ima vrlo velik odnos uloženog i dobivenog. Osim toga,

HC-SR04 senzor je jednostavno povezati sa *Atmega* mikroupravljačem, a mikroupravljačem kroz kod pisanim u *Atmel* studiju upravljamo radom senzora.

Kao što je napisano iznad, *HC-SR04* ultrazvučni senzor je 4-pinski modul, čiji su nazivi pinova: *Vcc, Trigger, Echo i Ground*. Ovaj senzor je vrlo popularan, korišten je u mnogim primjenama gdje je potrebno mjeriti udaljenost objekata. Modul ima dva oka na prednjoj strani koji tvore ultrazvučni predajnik i prijemnik. Senzor radi po jednostavnoj formuli fizike:

$$udaljenost = brzina * vrijeme \tag{2-1}$$

Ultrazvučni senzor transmitira ultrazvučni val, taj val putuje zrakom i kada naiđe na neki objekt od bilo kakvog materijala, reflektira se nazad. Reflektirani val prati modul prijemnika kako je prikazano na slici ispod [11].

Slika 2.7. *Rad ultrazvučnog senzora HC-SR04* [11]

Kako bi izračunali udaljenost pomoću gore navedene formule, potrebno je znati brzinu i vrijeme. U projektu se koristi modul sa ultrazvučnim valovima, a njihova brzina varira sa različitim uvjetima okoline, ali u normalnim uvjetima (22 °C) brzina ultrazvučnog vala iznosi otprilike 343 m/s. Krugovi ugrađeni na modulu će izračunati vrijeme potrebno da se ultrazvučni val vrati i uključi *echo* pin na visoku razinu za istu količinu vremena koliko je potrebno da se val vrati, te je na kraju poznato i vrijeme potrebno za izračunavanje udaljenosti prema formuli. Uz poznato vrijeme i brzinu, na kraju je potrebno izračunati udaljenost pomoću Atmega16 mikroupravljača.

HC-SR04 senzor udaljenosti se obično koristi s mikroupravljačima i mikroprocesorskim platformama kao što su Arduino, AVR, ARM, PIC, Raspberry Pie itd.. Senzor se napaja pomoću reguliranih +5 V kroz *Vcc* i *Ground* pinove. Struja koju koristi senzor je manja od 15 mA i stoga se može izravno napajati sa +5 V pinovima na ploči. *Trigger* i *echo* pinovi su ulazno/izlazni pinovi i stoga se mogu izravno spojiti na ulazno/izlazne pinove mikroupravljača. Za pokretanje mjerenja, *trigger* pin mora se postaviti u visoko stanje najmanje 10 µs, a zatim se isključuje. To će pokrenuti

osam ciklusa ultrazvučnog vala na 40 kHz od odašiljača i prijemnik će čekati da se val vrati. Kada se val vrati nakon što se reflektirao od nekog objekta, *echo* pin ide u visoko stanje za određenu količinu vremena koja će biti jednaka vremenu koje je potrebno da se val vrati na senzor [11]. Kako bi izračunali udaljenost, potrebno je izmjeriti širinu *echo* pulsa.

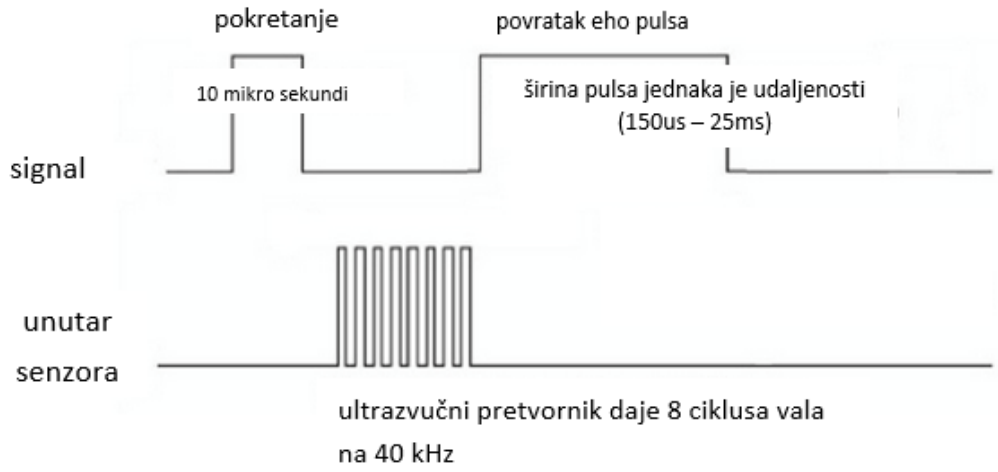
vrijeme = širina *echo* pulsa u µs, osim ako nije drugačije naznačeno.

- $$udaljenost\ [m] = \frac{vrijeme\ [s]\ *\ 343\ [m/s]}{2} \tag{2-2}$$

ili sa skraćenom formulom

- $$udaljenost\ [cm] = \frac{vrijeme}{58} \tag{2-3}$$

Vremenski dijagram *trigger* i *echo* signala prikazan je na slici ispod.



Slika 2.8. *Vremenski dijagram HC-SR04* [12]

U tablici ispod prikazane su karakteristike i parametri *HC-SR04* senzora.

Radni napon	DC 5 V
Radna struja	15 mA
Radna frekvencija	40 kHz
Maksimalni domet	4 m
Minimalni domet	2 cm
Kut mjerenja	30 stupnjeva
trigger ulazni signal	10 µs puls
echo izlazni signal	širina pulsa jednaka vremenu reflektiranog signala od objekta do senzora
Dimenzije	45*20*15 mm

Tablica 2.1. *Karakteristike i parametri HC-SR04* [13]

Slika 2.9. *Izgled senzora HC-SR04* [11]

2.4. AVR mikroupravljač

Mikroupravljač je čip koji sadrži najmanje jedan procesor, stalnu memoriju, privremenu memoriju, brojač i ulazno/izlaznu kontrolnu jedinicu. Mikroupravljač se može opisati kao računalo na čipu. Razlika između mikroupravljača i računala je da je računalo općenito računalo, a mikrokontroler je računalo usmjereno na jedan ili samo nekoliko zadataka. Mikroupravljač osim navedenih dijelova obično uključuje i mogućnosti serijskih komunikacija, prekidne rutine i analogne ulazno/izlazne mogućnosti [14].

AVR je obitelj mikroupravljača koje je razvila tvrtka *Atmel* 1996. godine. Arhitekturu AVR-a razvili su *Alf-Egil Bogen* i *Vegard Wollan*. AVR dobiva ime od svojih programera poznato kao ***A**dvanced **V**irtual **R**ISC*, ili skraćeno AVR. AVR je također bio jedan od prvih mikroupravljača koji su koristili *flash* memoriju na čipu za pohranu programa, za razliku od jednokratno programibilnog ROM-a, EPROM-a ili EEPROM-a koje su tada koristili drugi mikroupravljači.

AVR mikroupravljači dostupni su u tri kategorije:

- **TinyAVR** - manje memorije, male veličine, pogodno samo za jednostavnije aplikacije
- **MegaAVR** - to su najpopularniji mikroupravljači koji imaju veću količinu memorije (do 256 KB), veći broj ugrađenih perifernih uređaja i pogodni su za umjerene do složenih primjena.
- **XmegaAVR** - komercijalno se koristi za složene aplikacije, koje zahtijevaju veliku programsku memoriju i veliku brzinu [15].

2.4.1. **AVR** Atmega16 **mikroupravljač**

Za projekt parking senzora izabran je AVR Atmega16 mikroupravljač. Atmega16 je 8-bitni mikroupravljač visokih performansi iz Atmelove Mega AVR obitelji. To je 40-pinski

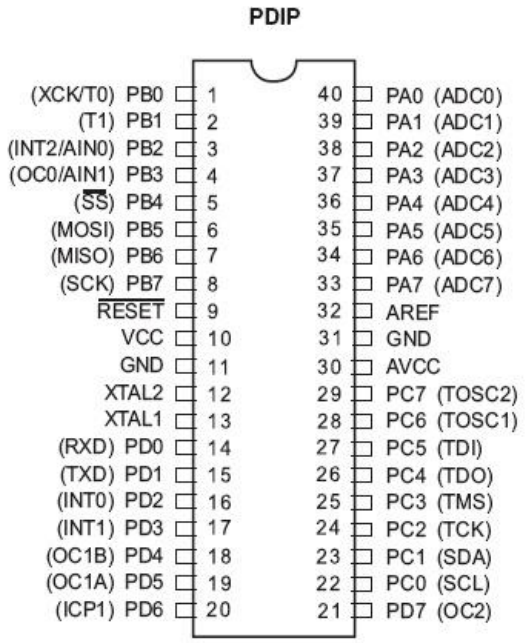
mikroupravljač temeljen na poboljšanoj RISC arhitekturi (engl. *Reduced Instruction Set Computing*). Ima programibilnu *flash* memoriju od 16 KB, statički RAM od 1 KB i EEPROM od 512 bajta. Radi na frekvencijama od 1 MHz do maksimalnih 16 MHz [16].

Slika 2.10. *Izgled Atmega16 mikroupravljača* [17]

Atmega16 sadrži 32 ulazno/izlazne linije koje su podijeljene na četiri 8-bitna porta označena kao PA, PB, PC i PD. Atmega16 ima različite ugrađene periferne uređaje poput USART-a, ADC-a, analognog komparatora, SPI-a, JTAG-a itd.. Svaki ulazno/izlazni pin ima alternativni zadatak koji se odnosi na ugrađene periferne uređaje [16].

2.4.1.1. **Opis priključaka** Atmega16 **mikroupravljača**

- **Port A**
  Koristi se za analogne ulaze A/D pretvarača. Ako A/D pretvarač nije omogućen, također služi u 8-bitnom dvosmjernom priključku za ulaz i izlaz.
- **Port B**
  Priključak B se koristi kao ulazno / izlazni 8-bitni dvosmjerni priključak koji ima unutarnje otpore.
- **Port C**
  Posebna značajka priključka C je JTAG sučelje. Ako je omogućeno JTAG sučelje, otpori na pinovima PC5, PC4, PC3 i PC2 će se aktivirati čak i ako se pojavi *reset*. Osim toga, priključak C se može koristiti kao ulazno/izlazni 8-bitni dvosmjerni priključak.
- **Port D**
  Port D služi funkcijama posebnih značajki kao što su prekidne rutine, brojača i UART-a [18].



Slika 2.11. *Atmega16 priključci* [18]

2.5. 16x2 LCD

LCD (engl. *Liquid Crystal Display*) označava zaslon od tekućeg kristala. To je elektronički modul prikaza. 16x2 LCD zaslon je osnovni modul i vrlo se često koristi u različitim uređajima i krugovima. Ovi moduli se preferiraju ispred sedam segmentnih zaslona i ostalih LED segmenata. Neki od razloga su: LCD su ekonomični, lako ih je programirati, nemaju ograničenja prikazivanja posebnih ili čak prilagođenih znakova (za razliku od sedam segmentnog prikaza), animacija itd.. LCD 16x2 znači da modul može prikazati 16 znakova po retku, a postoje dva retka. Na ovom LCD zaslonu svaki je znak prikazan u matrici 5x7 elemenata. Ovaj LCD ima dva registra, komandni i podatkovni registar. Komandni registar pohranjuje komandne instrukcije dane na LCD. Naredba je instrukcija koja se daje LCD-u za obavljanje zadanog zadatka kao što je inicijalizacija, brisanje zaslona, podešavanje položaja pokazivača, kontrolu zaslona itd.. Podatkovni registar pohranjuje podatke koji se prikazuju na LCD zaslonu. Podaci su ASCII vrijednost znaka koji se prikazuje na LCD zaslonu [19].

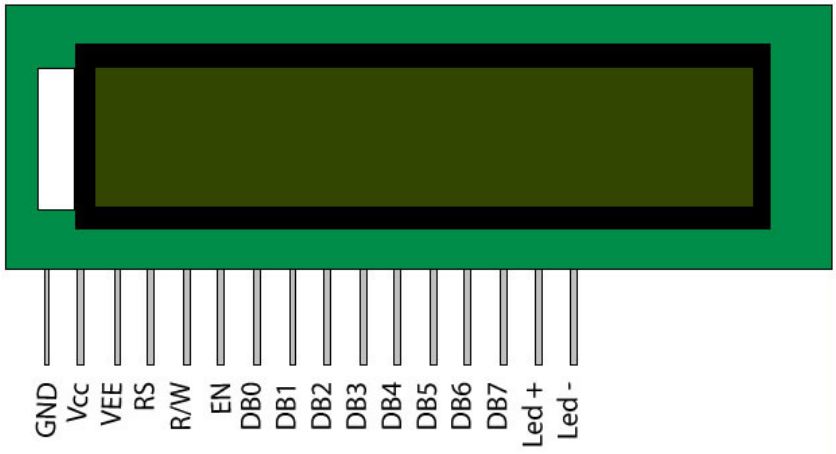
2.5.1. Načini rada LCD-a

LCD može raditi na dva različita načina: 4-bitni način rada i 8-bitni način rada. U 4-bitnom načinu rada podatke šaljemo u grupama od 4 bita, prvo gornja 4 bita (D4-D7), zatim donja 4 bita (D0-D3), jer je na LCD spojeno samo 4 podatkovne linije. To nam omogućuje slanje 8 bita podataka. U 8-bitnom načinu rada, možemo poslati 8-bitne podatke izravno u jednom potezu, budući da

koristimo sve 8 podatkovnih linija. Iz ovoga možemo zaključiti da je 8-bitn način rada brži i besprijekorniji od 4-bitnog moda, glavni nedostatak je što treba biti povezano 8 podatkovnih linija sa mikroupravljačem.

S obzirom da u ovom radu nije od velikog značaja broj podatkovnih linija povezanih sa mikroupravljačem, a velika brzina i dobar odziv su bitni, koristimo 8-bitni način rada LCD-a [20].

2.5.2. Opis priključaka LCD-a



Slika 2.12. *Izgled i položaj priključaka 16x2 LCD-a* [19]

Broj priključka	Funkcija	Naziv
1	Uzemljenje (0 V)	GND
2	Napajanje (5 V)	Vcc
3	Podešavanje kontrasta	V_{EE}
4	Odabire naredbeni registar kada je 0 i podatkovni registar kada je 1	*Registar Select*
5	0 za pisanje u registar, 1 za čitati iz registra	*Read/Write*
6	Šalje podatke na podatkovne priključke	*Enable*
7	8-bitni podatkovni priključak	DB0
8	8-bitni podatkovni priključak	DB1
9	8-bitni podatkovni priključak	DB2
10	8-bitni podatkovni priključak	DB3
11	8-bitni podatkovni priključak	DB4
12	8-bitni podatkovni priključak	DB5
13	8-bitni podatkovni priključak	DB6
14	8-bitni podatkovni priključak	DB7
15	Pozadinsko svjetlo (Vcc)	Led+
16	Pozadinsko svjetlo (GND)	Led

Tablica 2.2. *Opis priključaka LCD-a* [19]

2.6. Zujalica (engl. *buzzer*)

Zujalica (engl. *buzzer, beeper*) je zvučno-signalizacijski uređaj, koji može biti mehanički, elektromehanički ili piezoelektrični. Primjer mehaničkog su zujalice koje su se koristile u starim budilicama, a elektromehaničke su povezane s zvonom ulaznih vrata [21].

Piezo efekt je sposobnost materijala da generira električni naboj na dano mehaničko opterećenje. Ime potječe od grčke riječi *piezo* što znači pritisnuti. Kada se piezoelektrični materijal izloži mehaničkom opterećenju (pritisak) događa se premještanje pozitivnog i negativnog naboja, što rezultira vanjskim električnim poljem. Kada govorimo o zujalici, u njemu se nalazi mala zavojnica i magnet. Kada struja poteče kroz zavojnicu, ona se magnetizira i povlači prema magnetu. Iz prethodno objašnjenog *piezo* efekta, jasno je da će to izazvati mali "klik". Kako se to ponavlja nekoliko tisuća puta u sekundi, "klik" postaje zvuk [21].

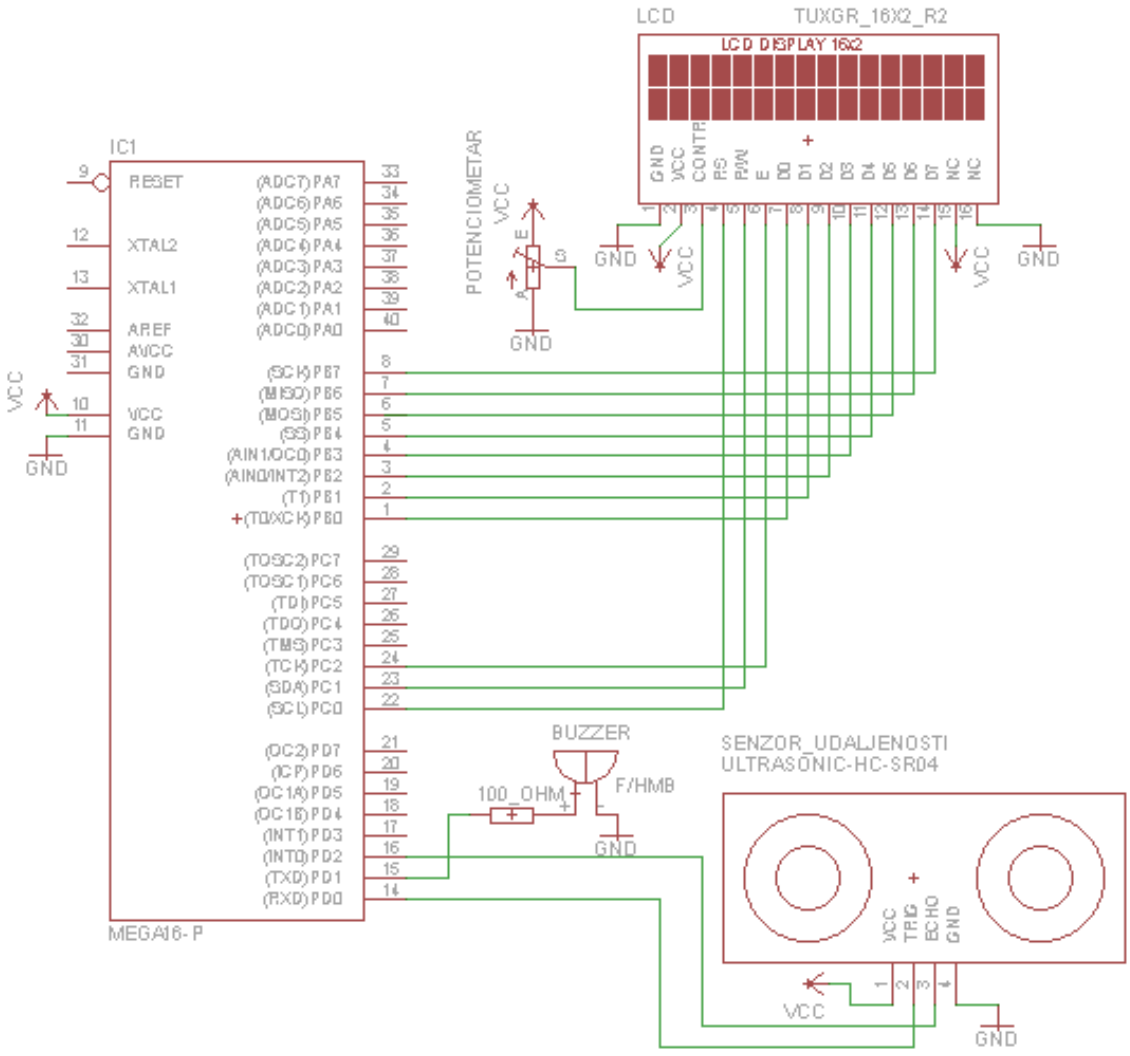
Slika 2.13. *Piezo zujalica* [21]

2.7. Dijelovi parking senzora i njegova shema

Kao što smo prethodno naveli i opisali, glavni dijelovi parking senzora su:

- AVR Atmega16 mikroupravljač
- *HC-SR04* senzor udaljenosti
- 16x2 LCD
- zujalica (engl. *buzzer*)

Na slici 2.14. prikazana je shema parking senzora. U njoj se vide svi spomenuti dijelovi, način njihova spajanja, pinovi mikroupravljača koji su korišteni i ostale elektroničke komponente. Prema toj shemi rađeno je spajanje komponenti na razvojnoj pločici.



Slika 2.14. *Shema parking senzora*

3. SOFTVERSKI DIO RADA

3.1. Atmel Studio 7

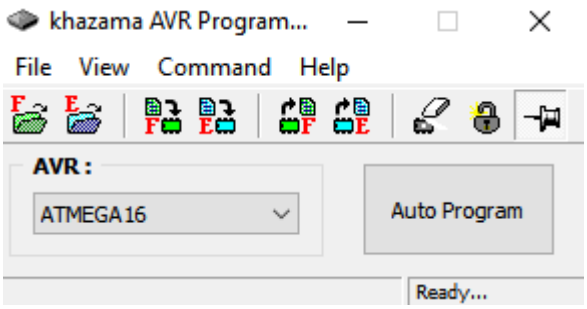
Kod za rad parking senzora temeljenog na Atmega16 mikroupravljaču pisan je u *Atmel Studio*-u verzije 7. *Studio 7* je integrirana razvojna platforma za razvoj i ispravljanje pogrešaka svih AVR mikroupravljačkih aplikacija. *Atmel Studio 7* daje besprijekorno i jednostavno okruženje za pisanje, izgradnju i ispravljanje aplikacija napisanih u C / C + +.

Glavne značajke:

- Podrška za 500 + AVR i SAM uređaja
- Velika biblioteka izvornog koda, uključujući upravljačke programe, komunikacijske stogove, 1.600+ primjera projekata s izvornim kodom, grafičke usluge i sl.
- Pisanje koda u C / C + + i ispravljanje s integriranim kompajlerom [22]

3.2. *Khazama* programator i postavke rada mikroupravljača

Khazama AVR programator se koristi za upisivanje softvera u mikroupravljač pomoću *USBAsp* programatora. To je mali, brzo odzivni program, koji se lako koristi. U njemu se mogu podesiti sve potrebne postavke ovisno o projektu.



Slika 3.1. *Khazama AVR programator*

Nakon što se spoji mikoupravljač sa programatorom, potrebno je odabrati tip AVR mikroupravljača (u ovom slučaju Atmega16), nakon toga se provjerava ispravnost mikroupravljača (engl. *Command/Read Chip Signature*). Ako je sve uredu, učitava se *.hex* datoteka koja sadrži kod i na kraju je potrebno kliknuti na *AutoProgram*. Na taj način je učitan kod u mikroupravljač.

3.2.1. **Postavke** Atmega16 **mikroupravljača za rad parking senzora**

Fuse bitovi pomažu u postavljanju postavki za AVR mikroupravljače. Sve postavke su dokumentirane i lako se mogu odabrati iz komandi ili postaviti izravno iz potvrdnih okvira u *Khazama* programatoru. Nove generacije mikroupravljača sposobne su raditi i sa unutarnjim i vanjskim opcijama rada. Na pitanja kako omogućiti da radi na određenim opcijama, te kako onemogućiti/omogućiti određenu periferiju, odgovor su *fuse* bitovi. Mikroupravljač ima nekoliko parametara koji se koriste za konfiguriranje prije nego što se može koristiti za vanjsko okruženje, odnosno periferne uređaje. Ti se parametri postavljaju pomoću *fuse* bitova. Drugim riječima, *fuse* bitovi određuju ponašanje mikroupravljača. Nakon što su *fuse* bitovi postavljeni za određenu konfiguraciju, mikroupravljač se može ponovno koristiti, bez ponovnog postavljanja svaki put kada se koristi, sve dok ga više ne želimo koristiti pod istom konfiguracijom [23].

U slučaju ovog rada i parking senzora, *fuse* bitovi su postavljeni jednom i ne mijenjaju se. Postavljeni su na zadane parametre Atmega16 mikroupravljača. Jedina izmjena s obzirom na zadane postavke je to što je onemogućen JTAG. Kada su postavljeni *fuse* bitovi po zadanim parametrima, dio priključaka C na mikroupravljaču nije radio. To je stvaralo probleme u radu parking senzora, jer na te priključke su bili spojeni registri LCD-a te nije bilo nikakvog ispisa na zaslonu. Nakon dugog istraživanja, greška je pronađena. Problem je bio omogućeni JTAG interfejs koji kada je upaljen zauzima dio priključaka C i ništa više preko tih priključaka ne može raditi. Frekvencija rada mikroupravljača je 1 MHz. Tu frekvenciju održava interni oscilator Atmega16 mikroupravljača.

Postavke *fuse* bitova su prikazane u tablici ispod.

High fuse	*Low fuse*
JTAG: onemogućen	*Brown-out* razina: 2.7V
OCD: onemogućen	*Brown-out* detekcija: onemogućena
SPI programiranje: omogućeno	Frekvencija izvora: 1 MHz
CKOPT: CKOPT=0	
EEPROM: onemogućeno	
BOOT veličina: 1024 riječi	

Tablica 3.1. *Postavke fuse bitova Atmega16 mikroupravljača*

3.2.2. Opis i postavke perifernih uređaja mikroupravljača potrebnih za rad parking senzora

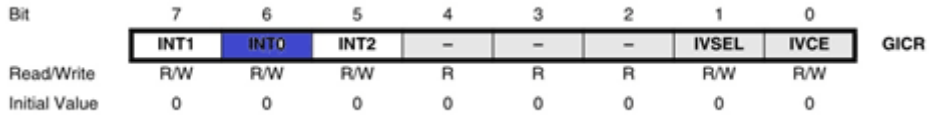
Za rad parking senzora korištena su dva periferna uređaja, a to su prekidna rutina (engl. *Interrupt service routine*) i brojač (engl. *Timer/Counter*). U nastavku je opisano njihovo djelovanje i postavke potrebne za rad parking senzora.

3.2.3. Interrupt service routine (ISR) – prekidna rutina

Prekidne rutine su u osnovi događaji koji zahtijevaju trenutnu pozornost mikroupravljača. Kada se dogodi prekidna rutina, mikroupravljač zaustavlja trenutni zadatak i prisustvuje prekidu izvršavanjem prekidne rutine, te na kraju prekidne rutine mikroupravljač se vraća na zadatak čije je izvođenje zaustavio i nastavlja izvođenje operacije. AVR mikroupravljači imaju dvije kategorije prekidnih rutina, a to su interne i eksterne prekidne rutine [24]. Ovdje govorimo o eksternim prekidnim rutinama. Atmega16 sadrži tri eksterna prekidna (engl. *interrupt*) priključka, a u radu se koristi jedan, *INT0*. Kako bi se mogle koristiti prekidne rutine, potrebno je postaviti određene postavke, a postavke za rad parking senzora su objašnjene u daljnjem tekstu.

Postavke registara za omogućavanje prekidnih rutina:

- GICR (engl. *General Interrupt Control Registar*) – za rad parking senzora je korištena prekidna rutina *INT0,* označena plavom bojom na slici ispod. Ovaj bit mora biti postavljen u jedinicu, kako bi se omogućila eksterna prekidna rutina. Postavljanje registra u kodu izgleda ovako: `GICR |=(1<<INT0);`
  Osim omogućavanja specifične prekidne rutine, globalne prekidne rutine moraju biti omogućene da mikroupravljač reagira na prekidne rutine. To je u kodu omogućeno na sljedeći način: `sei();` [25]



Slika 3.2. *Kontrolni registar prekidnih rutina* [25]

- MCUCR (engl. *MCU Control Registar*) - pomoću ovog registra podešena je osjetilna kontrola prekidnih rutina, odnosno određeno je na koje promjene i kada će se pokrenuti prekidna rutina. Moguće postavke ovog registra su prikazane u tablici ispod [25].

ISC01	*ISC00*	opis
0	0	niska razina INT0 generira prekidnu rutinu
0	1	bilo kakva promjena na INT0 generira prekidnu rutinu
1	0	padajući brid INT0 generira prekidnu rutinu
1	1	rastući brid INT0 generira prekidnu rutinu

Tablica 3.2. *Moguće postavke MCUCR registra* [25]

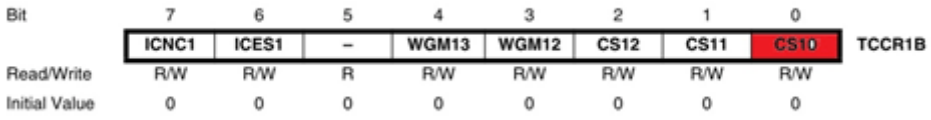
O *echo* pinu senzora udaljenosti ovisi pokretanje prekidnih rutina. U radu je potrebno da se prekidna rutina pokrene i kada *echo* prelazi iz niskog u visoko stanje i obrnuto. Zbog toga MCUCR registar je postavljen na sljedeći način: *MCUCR |=(1<<ISC00);* . To znači da će se prekidna rutina biti pokrenuta na svaku promjenu *echo* pina. Prekidne rutine su potrebne kako bi pokrenule brojač i zaustavile ga. Dakle, *ISC00* je postavljen u jedinicu, te se prekidna rutina pokreće kada logika na *INT0* prelazi iz niske u visoku razinu i prekidna rutina kada logika prelazi iz visoke u nisku razinu. Kao što je spomenuto, promjene na *INT0* ovise o promjenama na *echo* priključku senzora udaljenosti.

3.2.4. *Timer/Counter1* – mjerač vremena ili brojač

Mjerač vremena je registar, ali ne običan. Vrijednost tog registra se povećava ili smanjuje automatski. Kod Atmega16 mikroupravljača postoje dvije vrste mjerača vremena: 8-bitni i 16-bitni. U 8-bitnom širina registra je 8 bita, dok je u 16-bitnom širina registra 16 bita. To znači da 8-bitni mjerač može brojati 2 ^ 8 = 256 koraka, od 0 do 255, a 16-bitni mjerač može brojati 2 ^ 16 = 65536 koraka, od 0 do 65535. Zbog ove značajke, mjerači vremena su također poznati kao brojači. Kada brojač dosegne svoj maksimum, vraća se na početnu vrijednost. Možemo reći da dolazi do prelijevanja (engl. *overflow*). Brojač je pot puno neovisan o procesoru, te se pokreće paralelno s procesorom, što ga čini prilično točnim [26].

Postavke registara brojača potrebnih za rad parking senzora:

- TCCR1B (engl. *Timer/Counter1 Control Register B*) – za rad parking senzora nam je potreban samo CS10 bit, označen crvenom bojom na slici ispod.



Slika 3.3. *Sadržaj TCCR1B registra* [25]

To je bit za odabir takta brojača koji jednostavno omogućuje i onemogućuje brojač. Napisano u kodu to izgleda ovako: *TCCR1B |=(1<<CS10);*

Iako radi zajedno s drugim bitovima *CS10* i *CS12*, njih ovdje ne koristimo zbog toga što nema skaliranja (engl. *prescaling*) frekvencije dovedene na brojač, odnosno na brojač se dovodi frekvencija ista koja se koristi za rad mikroupravljača.

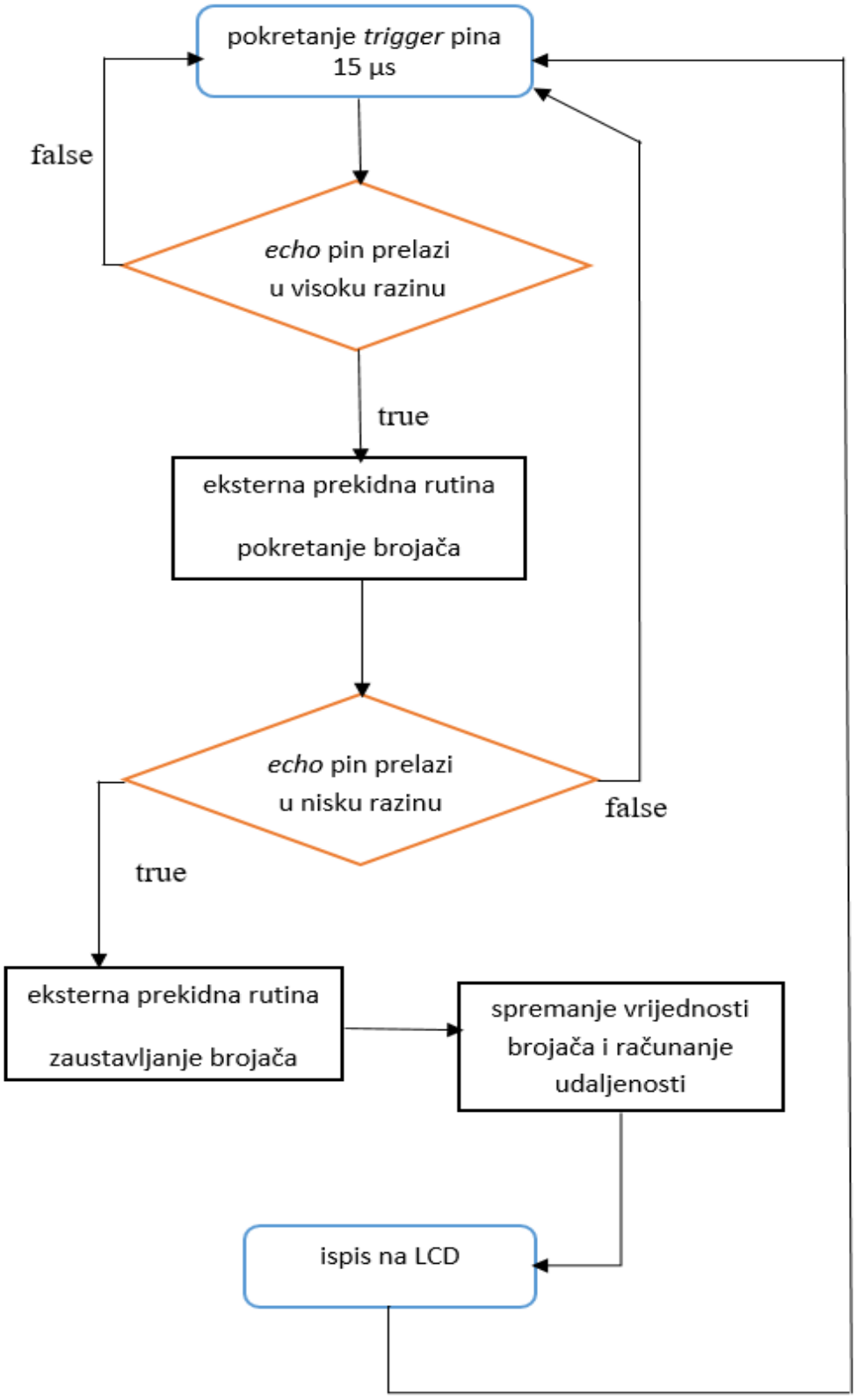
- TCNT1 (engl. *Timer/Counter1*) – ovaj registar ima 16-bitnu širinu zbog toga što je *Timer1* 16-bitni registar. On omogućuje izravan pristup, i za čitanje i za pisanje, 16-bitnom brojaču. Vrijednost brojača je nakon brojanja pohranjena u ovom registru.

3.3. Opis dijelova koda i rad parking senzora

3.3.1. Princip rada parking senzora

1. Pokretanje senzora pokretanjem *trigger* pina najmanje 15 μs.
2. Kada *echo* pin pređe u visoku razinu dobivamo eksternu prekidnu rutinu i pokreće se brojač unutar prekidne rutine, koji se izvršava odmah iza okidanja prekidne rutine.
3. Kada *echo* pin pređe iz visoke u nisku razinu, ponovo se generira prekidna rutina, ali ovaj put se brojač zaustavlja.
4. Dok je puls na *echo* pinu prešao iz visoke u nisku razinu, brojač je pokrenut i zaustavljen. Taj broj koji daje brojač ažurira se u memoriju za dobivanje udaljenosti, jer tako je dobivena širina *echo* pulsa u brojčanom obliku.
5. Nakon toga rade se izračuni po izrazu (2-3) za izračun udaljenosti u cm.
6. Poslije izračuna vrši se prikaz na 16x2 LCD-u.

3.3.2. Princip rada kroz dijagram toka



Slika 3.4. *Dijagram toka*

3.3.3. Opis rada parking senzora kroz kod

U daljnjem tekstu opisani su pojedini dijelovi koda, koji omogućavaju rad parking senzora, mjerenje udaljenosti, paljenje i gašenje zujalice, te prikaz na 16x2 LCD-u.

Na slici 3.5. prikazana su sva zaglavlja, funkcije i varijable koje je potrebno definirati i/ili deklarirati.

```
1   #include <avr/io.h>
2   #include <avr/interrupt.h>
3   #define F_CPU 1000000
4   #define DL _delay_ms(10);
5   #include <util/delay.h>
6   #include <stdlib.h>
7
8   void lcd_init();
9   void lcd_cmd(char cmd);
10  void lcd_data(char data);
11  void lcd_string(char *str);
12  void lcd_Set_Cursor(char a, char b);
13  void LCD_Custom_Char (unsigned char loc, unsigned char *str);
14  void beep(int nbeep);
15  void beep2(int nbeep);
16  void beep3(int nbeep);
17
18  static volatile int pulse = 0;
19  static volatile int i = 0;
```

Slika 3.5. *Definiranje zaglavlja, deklariranje funkcija i postavljanje varijabli*

Na početku svakog koda nužno je definirati zaglavlja koja će biti potrebna u kodu. U ovom slučaju to su zaglavlja koja omogućavaju kontrolu toka podataka preko priključaka mikroupravljača, zatim zaglavlje koje omogućuje korištenje prekidnih rutina, te zaglavlje za korištenje funkcija kašnjenja ili odgode. Potrebno je još definirati frekvenciju rada mikroupravljača.

U nastavku, deklarirane su funkcije potrebne za rad parking senzora. To su funkcije inicijalizacije LCD-a, izvršavanja naredbi, te prikaza podataka i teksta na LCD. Tu je i funkcija postavljanja kursora na željenu poziciju i funkcija za definiranje prilagođenih oblika karaktera koji će se prikazivati na zaslonu LCD-a, zatim funkcije koje kontroliraju rad zujalice. Spomenute funkcije su na kraju koda i definirane.

Na kraju, definirane su dvije varijable kao *static volatile,* gdje *static* znači da je varijabla vidljiva u cijelom programu, a *volatile* označava da je to informacija za kompajler koja kaže da varijabla može biti promijenjena izvan normalnog puta izvršenja, kao što je prekidna rutina, što je ovdje i slučaj. U varijablu *pulse* će biti spremana vrijednost brojača, iz koje se računa udaljenost, a

varijabla *i* će biti korištena za označavanje prelaska *echo* pina iz niske u visoku vrijednosti i obrnuto, odnosno za pokretanje i zaustavljanje brojača u prekidnoj rutini.

Nakon što su postavljena sva zaglavlja, deklarirane funkcije i definirane varijable, slijedi glavna funkcija programa, ***int main();*** funkcija. Kako bi mogli biti prikazani različiti oblici karaktera na zaslonu LCD-a, potrebno je definirati oblik karaktera i spremiti ga u CGRAM (engl. *Character generator RAM*). Poznato je da je svaki karakter kombinacija 5*8 elemenata. Prema tome odabrani su elementi s obzirom kakav je oblik potreban. U ovoj slučaju, potrebni su karakteri koji će na zaslonu LCD-a prikazivati u obliku rastućeg/padajućeg grafa udaljenost od određenog objekta. Izgled grafova je prikazan u daljnjem tekstu, a na slici 3.6. vidljiv je način definiranja različitih karaktera, spremljenih u niz u heksadecimalnom obliku.

```
int main(void)
{
    unsigned char p0[8]= {0x00,0x00,0x00,0x00,0x00,0x01,0x03,0x07};
    unsigned char p1[8]= {0x00,0x00,0x00,0x00,0x00,0x10,0x18,0x1C};
    unsigned char p2[8]= {0x07,0x03,0x01,0x00,0x00,0x00,0x00,0x00};
    unsigned char p3[8]= {0x1C,0x18,0x10,0x00,0x00,0x00,0x00,0x00};
    unsigned char p4[8]= {0x00,0x00,0x00,0x10,0x18,0x0C,0x04,0x06};
    unsigned char p5[8]= {0x06,0x04,0x0C,0x18,0x10,0x00,0x00,0x00};
    unsigned char p6[8]= {0x10,0x08,0x0C,0x06,0x02,0x03,0x03,0x03};
    unsigned char p7[8]= {0x03,0x03,0x03,0x02,0x06,0x0C,0x08,0x10};
    unsigned char p8[8]= {0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00};

     LCD_Custom_Char(0, p0);
     LCD_Custom_Char(1, p1);
     LCD_Custom_Char(2, p2);
     LCD_Custom_Char(3, p3);
     LCD_Custom_Char(4, p4);
     LCD_Custom_Char(5, p5);
     LCD_Custom_Char(6, p6);
     LCD_Custom_Char(7, p7);
     LCD_Custom_Char(8, p8);
```

Slika 3.6. *Definiranje oblika karaktera i njihovo postavljanje u CGRAM*

Kada su oblici karaktera spremni, potrebno ih je postaviti u CGRAM. To je memorija u koju se može pisati i iz koje se može čitati, te kako joj ime kaže, služi za generiranje i spremanje različitih karaktera. Spremanje u CGRAM napravljeno je pomoću ***LCD_Custom_Char*** funkcije. Kada su oblici jednom učitani, mogu se prikazivati na zaslonu jednostavnim pozivanjem lokacije na koju je karakter spremljen pomoću funkcije za ispis.

Osim različitih oblika karaktera, potrebno je definirati pinove mikroupravljača koji će biti korišteni za rad parking senzora. Na slici 3.7. prikazani su registri pomoću kojih su definirani pinovi, zatim

varijable u koje se sprema izračunata udaljenost parking senzora i funkcija za inicijalizaciju LCD-a. Tu su i registri kojima se definiraju prekidne rutine, koje su prethodno opisane.

```
46      DDRA = 0b00000001;
47      DDRB = 0xFF;
48      DDRD = 0b11111011;
49      PORTD =(0<<PD1);
50      lcd_init();
51
52      _delay_ms(50);
53
54      GICR|=(1<<INT0);
55      MCUCR|=(1<<ISC00);
56      sei();
57
58      double COUNTA = 0;
59      char SHOWA [16];
```

Slika 3.7. *Definirani pinovi, varijable i prekidne rutine*

Pomoću registra DDRA (engl. *Dana Direction Registar A*) definiran je PA0 (engl. *PinA0*) kao izlazni pin. Taj pin je korišten za upravljanje pozadinskim svjetlom LCD-a. Kada udaljenost senzora od objekta premaši određenu udaljenost, pozadinsko svjetlo se gasi, jer nam na velikim udaljenosti nisu potrebne mogućnosti parking senzora, nego samo kad se senzor približava objektima. DDRB (engl. *Dana Direction Registar B*) registar korišten je za postavljanje cijelog porta B kao izlaznog, za spajanje i omogućavanje rada LCD-a. Vrijednost PD1 (engl. *PinD1*) postavljena je u 0. Taj pin je korišten za upravljanje radom zujalice, čije paljenje i gašenje ovisi o udaljenosti parking senzora od objekta. Nakon što su postavljeni pinovi, inicijaliziran je LCD pomoću funkcije ***lcd_init().*** U njoj se nalazi više funkcija izvršavanja naredbi, kojima inicijaliziramo LCD i pripremamo ga za ispis podataka na njemu. To su funkcije brisanja zaslona, funkcija kojom kažemo da koristimo 8-bitni način rada LCD-a i na kraju postavljanje kursora na početnu poziciju. Nakon definiranja pinova, postavljeno je kašnjenje od 50 ms, kako bi isti bili pravilno definirani. Definirane su i dvije varijable, *double* i *char* tipa. U *double* varijablu sprema se izračunata udaljenost parking senzora. *Double* koristimo zbog dvostruke preciznosti i mogućnosti prikaza decimalne vrijednosti. *Char* varijabla se koristi za prikaz udaljenosti na LCD-u.

Nakon što su definirane i deklarirane sve potrebne funkcije, varijable, zaglavlja i registri, na red dolazi beskonačna *while* petlja, ***while(1);***, u kojoj se odrađuju sve aktivnosti nužne za mjerenje

udaljenosti parking senzora od objekta. Na slici 3.8. prikazan je početak programa, odnosno okidač koji pokreće senzor udaljenosti.

```
62      while(1)
63      {
64          PORTD|=(1<<PIND0);
65          _delay_us(15);
66          PORTD &=~(1<<PIND0);
67
```

Slika 3.8. *Okidač za pokretanje senzora udaljenosti*

PIND0 postavljen je u visoku razinu na 15 µs, a zatim se isključuje. Time je napravljen utjecaj na *trigger* pin senzora udaljenosti, što generira 8 ciklusa vala od 40 kHz koje šalje odašiljač. Nakon što je odašiljač poslao valove, prijemnik čeka njegovo vraćanje nakon reflektiranja od objekta. Kada se val vrati, *echo* pin ide u visoku razinu, što će rezultirati aktiviranjem prekidne rutine i pokretanjem brojača, što je prikazano na slici 3.9..

```
196 ISR(INT0_vect)
197 {
198     if (i==1)
199     {
200         TCCR1B=0;
201         pulse=TCNT1;
202         TCNT1=0;
203         i=0;
204     }
205         if (i==0)
206         {
207             TCCR1B|=(1<<CS10);
208             i=1;
209         }
210 }
```

Slika 3.9. *Prekidna rutina, pokretanje i zaustavljanje brojača*

Kada je program ušao u prekidnu rutinu, prvo se obavlja uvjet kada je varijabla *i=0*, jer je tako postavljeno pri definiranju varijable na početku programa. Time je pokrenut brojač i postavljena je varijabla ***i*** u jedinicu. Kada *echo* pin pređe iz visoke u nisku razinu, ponovno se pokreće prekidna rutina, brojač se zaustavlja i postavlja u nulu, a vrijednost brojača je pohranjena u TCNT1 registar. Ta vrijednost je spremljena u varijablu *pulse*, a registar i varijablu *i* postavljamo u nulu. Tako je brojač spreman za ponovno pokretanje kada se pokrene okidač.

Nakon što je vrijednost brojača spremljena u *pulse* varijablu, računa se udaljenost od objekta prema formuli za udaljenost, izraz (2-3), što je vidljivo na slici 3.10..

```
68          COUNTA = (double)pulse/58;
69          dtostrf(COUNTA, 2, 2, SHOWA);
```

Slika 3.10. *Računanje udaljenosti*

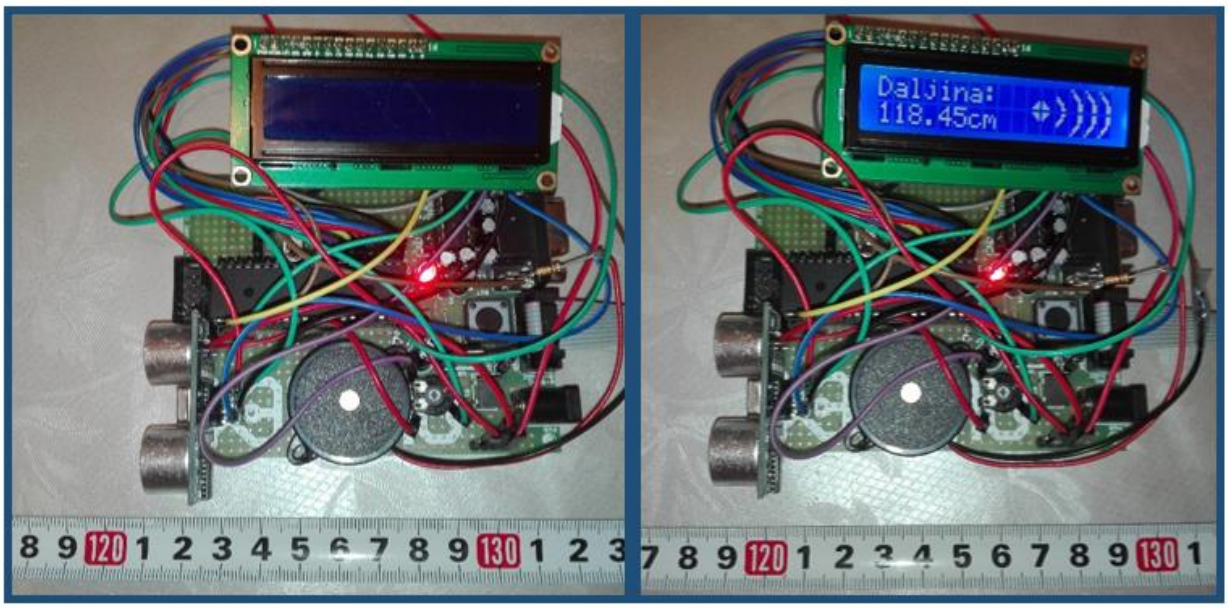
Za prikazivanje udaljenosti na zaslonu LCD-a, pomoću funkcije *dtostrf* potrebno je prebaciti vrijednost udaljenosti *double* tipa u *char* tip varijable, jer funkcija za ispis teksta na LCD je *char* varijabla.

Kada je izračunata vrijednost udaljenosti, potrebno ju je ispisati na zaslon LCD-a. Osim udaljenosti u centimetrima, na zaslon se ispisuju u razni oblici karaktera koje su prethodno kreirani. Ti karakteri prikazuju graf koji uz udaljenost dodatno daje predodžbu korisniku da se nalazi jako blizu objekta ili je još na udaljenosti koja nije zabrinjavajuća. Na slici 3.11. vidljiv je način prikaza grafa na zaslonu. U ovisnosti o udaljenosti, na zaslon se ispisuje veći ili manji graf ili je zaslon potpuno prazan. Taj dio je urađen sa *if, else* i *else if* uvjetima. Prvi uvjet određuje da li je senzor unutar intervala mjerenja. Ako je senzor udaljen više od 120 cm od objekta, zaslon je prazan, ne prikazuje se ništa, a pozadinsko svjetlo je ugašeno. To znači da iznad 120 cm nije potreban parking senzor, jer vozač automobila još nije u osjetljivoj zoni.

```
72          if (COUNTA >= 120){
73
74              lcd_Set_Cursor(1, 0);
75              lcd_string("                ");
76              lcd_Set_Cursor(2, 0);
77              lcd_string("                ");
78              PORTA = (0<<PA0);
79          }
80          else{
81              lcd_Set_Cursor(1,0);
82              lcd_string("Daljina:");
83              lcd_Set_Cursor(2, 0);
84              lcd_string(SHOWA);
85              lcd_string("cm");
86              PORTA = (1<<PA0);
87          }
```

Slika 3.11. *Ispis udaljenosti i različitih oblika karaktera na zaslon LCD-a*

Sa prvim uvjetom eliminirana je potreba rada parking senzora. Ako se senzor nalazi unutar 120 cm od objekta, pozadinsko svjetlo se pali, ispisuje se udaljenost na zaslon.



Slika 3.12. *Opseg rada parking senzora*

Na slici 3.12. prikazan je opseg rada senzora, odnosno granična vrijednost udaljenosti nakon koje se senzor gasi. Lijevo na slici senzor se nalazi nešto iznad 120 cm od objekta, točnije 121 cm. Prema upaljenoj LED-ici na razvojnoj pločici jasno je da senzor radi, odnosno u pozadini on još mjeri udaljenost, međutim nije ju potrebno prikazivati korisniku odnosno vozaču. Kada senzor uđe u područje mjerenja, desno na slici, pali se pozadinsko svjetlo i sklop prikazuje izmjerenu udaljenost na zaslonu LCD-a.

Kada je određeno da li je senzor unutar područja rada i mjerenja, te ako jest, uz udaljenost, na zaslon se prikazuje i graf, čija veličina i oblik ovise o udaljenosti. Što je udaljenost manja, manji je i uži graf, a što je veća udaljenost, veći je i širi graf.

Na slici 3.13. prikazan je *if* uvjet, kojim je postavljeno da ako je udaljenost manja ili jednaka 20 cm, na zaslon će biti ispisan najmanji dio grafa, jer je vozač na jako maloj i kritičnoj udaljenosti od objekta. Osim ispisa grafa, reproduciran je i zvuk zujalice, s velikom brzinom paljenja i gašenja, kako bi alarmirao vozača da je na vrlo maloj udaljenosti od objekta.

```
87          if (COUNTA <= 20)
88          {
89              lcd_Set_Cursor(1,10);
90              lcd_data(0);
91              lcd_data(1);
92              lcd_string(" ");
93              lcd_string(" ");
94              lcd_string(" ");
95              lcd_string(" ");
96
97              lcd_Set_Cursor(2,10);
98              lcd_data(2);
99              lcd_data(3);
100             lcd_string(" ");
101             lcd_string(" ");
102             lcd_string(" ");
103             lcd_string(" ");
104             beep(3);
105
106         }
107
```

Slika 3.13. *Ispis grafa na zaslonu LCD-a*

Graf je ispisan tako da je kursor postavljen u prvi red, zatim su ispisani karakteri kakve želimo i dalje se nalaze praznine. Na isti način je ispisan i drugi red. Za ispis grafa na zaslon korišteno je po šest karaktera u svakom redu. Ovisno o udaljenosti, biti će ispisan određeni broj karaktera koji formira graf koji će pokazati da li smo blizu objekta ili smo još na sigurnoj udaljenosti. Zbog toga u ovom slučaju su u svakom redu ispisana po dva karaktera i četiri praznine. Na slici 3.14 prikazano je kako to izgleda u praksi.

Slika 3.14. *Ispunjavanje prvoj uvjeta udaljenosti*

Praznine su pisane u kodu zbog toga što osim ovog uvjeta postoji još *else if* uvjeta koji su postavljeni na veće udaljenosti i ispisuju veći graf. Kada je senzor na većoj udaljenosti ispisuje se veći graf, a kada se krene približavati objektu, program ispunjava *else if* uvjete sa manjom vrijednosti udaljenosti, te je potreban i manji graf. Tada praznine u uvjetu brišu karaktere grafa koji ne označavaju trenutnu udaljenost, nego neku veću. Tako ovisno o udaljenosti, graf se povećava ispisivanjem više specifičnih karaktera i smanjuje dodavanjem praznina na mjesta koja

želimo izbrisati. Osim spomenutog uvjeta koji predstavlja udaljenost do 20 cm, postoje i drugi uvjeti sa većim udaljenostima, od 40, 60, 80 i 120 cm. Na svakoj spomenutoj udaljenosti ispisuje se veći graf nego na 20 cm i zujalica ima veće kašnjenje između svakog paljenja i gašenja, što znači da se sporije pali i gasi i tako korisniku daje predodžbu da je na većoj i sigurnijoj udaljenosti. Na slici 3.15. prikazan je uvjet koji ispisuje graf i aktivira rad zujalice za udaljenost do 40 cm.

```
107             else if (COUNTA <= 40)
108             {
109                 lcd_Set_Cursor(1, 10);
110                 lcd_data(0);
111                 lcd_data(1);
112                 lcd_data(4);
113                 lcd_string(" ");
114                 lcd_string(" ");
115                 lcd_string(" ");
116 
117                 lcd_Set_Cursor(2, 10);
118                 lcd_data(2);
119                 lcd_data(3);
120                 lcd_data(5);
121                 lcd_string(" ");
122                 lcd_string(" ");
123                 lcd_string(" ");
124                 beep2(3);
125             }
```

Slika 3.15. *Ispis grafa uz uvjet do 40 cm udaljenosti*

Za usporednu sa slikom 3.13., razlika je što je ovom uvjetu ispisan karakter više, što formira veći graf. Za sve sljedeće uvjete ispisuje se veći graf i zujalica se pali i gasi sve sporije. Uvjetu su pisani na isti način te ih nećemo sve posebno objašnjavati. Osim toga, na kraju programa opisane su i definirane sve funkcije koje se koriste u programu, funkcije rada sa LCD-om te funkcije za paljenje i gašenja zujalice, a biti će vidljive u prilogu rada.

4. PARKING SENZOR

Rad je podijeljen na hardverski i softverski dio. Hardverski dio baziran je na komponente potrebne da bi se napravio parking senzor, a u softverskom dijelu najvažniju ulogu ima kod i postavke rada mikroupravljača. Kada su odabrane sve komponente za parking senzor, od senzora udaljenosti, LCD-a, mikroupravljača i zujalice, formirana je shema sklopa. Nakon toga bilo je potrebno napisati kod koji će omogućiti sklopu da mjeri udaljenost, odnosno obavlja zadatke parking senzora. Sklop je spojen prema shemi na slici 2.13., a mikroupravljač je podešen za potrebne uvjete rada i u njega je ubačen funkcionalan kod koji omogućuje mjerenje udaljenosti i ispis raznih karaktera na zaslon LCD-a. Izgled i rad parking senzora, te njegove glavne funkcije prikazane su u daljnjem tekstu.

Kao što je već kod objašnjavanja koda rečeno, parking senzor aktivira se na udaljenosti od 120 cm i manje. Iako senzor ima veću moć mjerenja od 120 cm, za rad parking senzora to nije potrebno. Mjerenja senzora su podijeljena u dijelove. To znači da između 0 i 20 cm ispisuje se najmanji graf i zujalica ima najveću brzinu paljena i gašenja, i tako svakih 20 cm povećava se graf na zaslonu i smanjuje brzina paljenja i gašenja zujalice, sve do 80 cm. Tada je razlika nije 20 cm nego 40 cm, te je najveći graf između 80 i 120 cm, odnosno tada je korisnik još uvijek na sigurnoj udaljenosti prilikom parkiranja. Na slikama ispod prikazani su načini izgleda zaslona LCD-a, promjene udaljenosti i veličine grafa.

Slika 4.1. *Izgled zaslona na udaljenosti od 16 cm*

Na slici 4.1. vidljiv je izgled zaslona na udaljenosti od 16 cm. Udaljenost od 16 cm ulazi u prvi uvjet udaljenosti (između 0 i 20 cm), te je graf u tom slučaju najmanji, kako i treba biti.

Slika 4.2. *Izgled zaslona na udaljenosti od 21 cm*

Slika 4.2. prikazuje izgled zaslona na udaljenosti od 21 cm. U usporedbi sa slikom 4.1., vidljivo je da se sa povećanjem udaljenosti iznad 20 cm, odnosno ulaskom u drugi uvjet udaljenosti, povećala i veličina grafa udaljenosti.

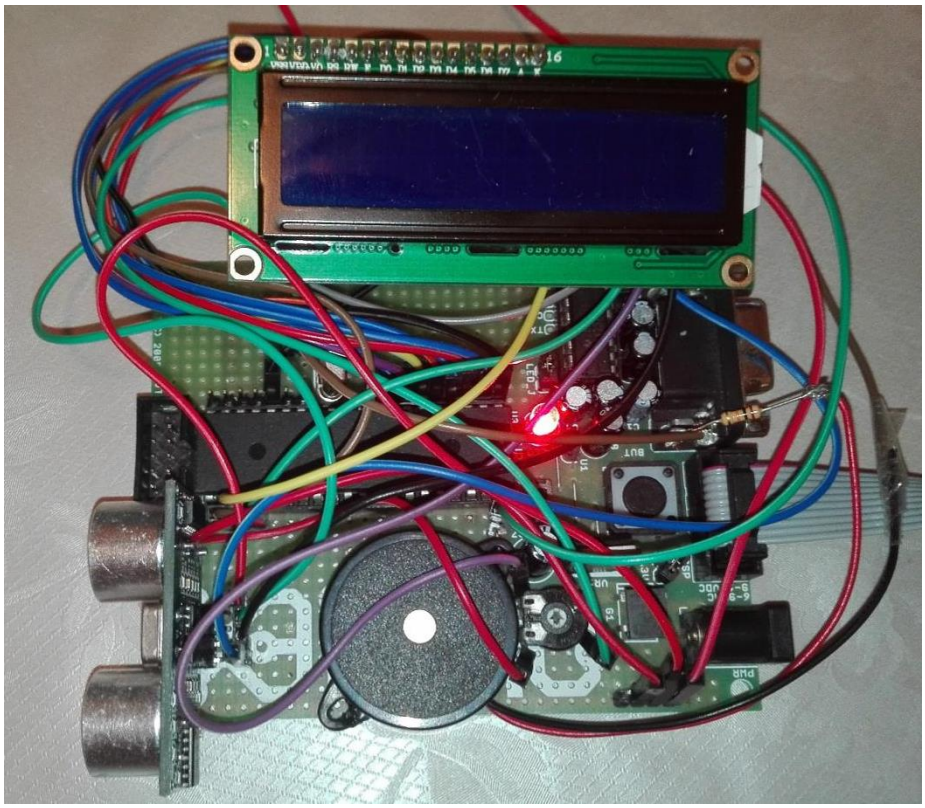
Slika 4.3. *Izgled zaslona na udaljenosti od 43 cm*

Slika 4.3. prikazuje izgled zaslona na udaljenosdi od 43 cm. Kao i u prethodnom slučaju, vidimo da povećanjem udaljenosti, povećava se i veličina grafa udaljenosti.

Možemo zaključiti da se stalnim povećanjeam udaljenosti, povećava i veličina grafa. Tako povećavanjem udaljenosti, dolazimo do krajnjih granica mjerenja i rada parking senzora.

Slika 4.4. *Izgled zaslona na udaljenosti od 88 cm*

Slika 4.4. prikazuje zaslon kada se ispunjava zadnji uvjet udaljenosti, između 80 i 120 cm. U opsegu tih udaljenosti graf je najveći, kao što je vidljivo na slici. Nakon 120 cm zaslon je ugašen i ne prikazuje se udaljenost, dok god senzor ponovno ne uđe u opseg mjerenja.



Slika 4.5. *Izgled prototipa parking senzora*

Na slici 4.5. prikazan je prototip parking senzora, odnosno radna verzija. Kao što je navedeno u radu, sklop sadrži sve prethodno navedene komponente, što je vidljivo i na slici. Tu je u prvom planu razvojna pločica na koju su spojene sve komponente. Na njoj se nalazi Atmega16 mikroupravljač, zatim senzor udaljenosti *HC-SR04*, zujalica, te na kraju i LCD na kojem se prikazuje mjerena vrijednost. Osim glavnih komponenti, korištene su razne žice za spajanje, potenciometar za podešavanje kontrasta LCD-a, te otpornik od 100 Ω kako bi ublažio glasan zvuk zujalice.

5. MJERENJA S PARKING SENZOROM

Nakon što je ostvaren funkcionalan rad parking senzora, napravljena su mjerenja točnosti i preciznosti istog. Budući da točnost mjerenja parking senzora direktno vezana za brzinu zvuka, a brzina zvuka ovisi o temperaturi i gustoći zraka u kojem se vrši propagacija vala, napravljeno je više mjerenja u različitim uvjetima kako bi pokazali mogućnosti, vrline i mane parking senzora. Prema tome, napravljena su mjerenja na različitim temperaturama, na temperaturi od 22 °C, prema kojoj je kalibriran senzor, zatim na temperaturi od 40 °C koja će biti prikazana u tablici. Objekt od kojeg se odbija val, odnosno od kojeg mjerimo udaljenost je čvrsta drvena ploča, koja omogućava reflektiranje vala u cijelosti nazad. Kako bi bili zorno prikazani rezultati mjerenja, pomoću grafova prikazani su odnosi između stvarne udaljenosti i udaljenosti dobivene pomoću parking senzora. Osim mjerenja na različitim temperaturama, napravljena su i mjerenja s obzirom na različite vrste objekata od kojih se val odbija prilikom mjerenja. S obzirom na različit sadržaj i gustoću objekata, o tome ovisi i koliko će se količine vala vratiti do senzora udaljenosti, a koliko će biti apsorbirano u objektu. Potrebno je da je što manja količina vala apsorbirana kako bi mjerenje bilo preciznije. S toga, osim mjerenja s čvrstim i gustim objektima kao što je drvo ili zid, napravljena su i mjerenja sa spužvom, kartonom, staklom i tkaninom. Sva mjerenja, grafovi i analize su prikazani u daljnjem tekstu i tablicama.

stvarna udaljenost [cm]	izmjerena udaljenost na 22 °C [cm]	izmjerena udaljenost na 40 °C [cm]
5,00	5,03	4,98
10,00	10,17	9,81
15,00	14,86	14,25
20,00	20,1	19,12
25,00	24,78	23,95
30,00	29,72	28,97
35,00	34,86	33,95
40,00	39,9	38,98
45,00	45,05	43,88
50,00	50,03	48,9
55,00	55,05	53,83
60,00	59,98	58,55
70,00	70,12	68,74
80,00	80,19	78,6
90,00	90,66	88,22

100,00	100,33	98,17
110,00	110,43	108,17
120,00	119,95	118,24

Tablica 5.1. *Rezultati mjerenja na različitim temperaturama*

U tablici 5.1. prikazani su rezultati mjerenja provedenih na različitim temperaturama. Udaljenosti su u razmacima od 5 i 10 cm. Uzeto je 18 uzoraka mjerenja. Iz tablice možemo vidjeti da rezultati mjerenja na temperaturi od 22 ºC su skoro identični stvarnim udaljenostima, dok rezultati na temperaturi od 40 ºC imaju odstupanja, ali ne prevelika. Odstupanja se događaju zbog različite brzine zvuka na različitim temperaturama, a rad senzora se zasniva na ultrazvučnim valovima. Brzina zvuka pri temperaturi od 22 ºC iznosi približno 343 m/s, a pri 40 ºC iznosi približno 355 m/s. Računanje udaljenosti u kodu je kalibrirano prema brzini zvuka u normalnim uvjetima rada, odnosno na 343 m/s. Zbog toga, kada se parking senzor nađe u uvjetima povišene temperature i veće brzine zvuka, dolazi do otklona od stvarnih vrijednosti. To se može objasniti na sljedeći način: senzor se okida i šalje zvučni val, val se reflektira i vraća na senzor. Tada se aktivira brojač koji broji isto vremena koliko je bilo potrebno da se reflektirani val vrati na senzor. S obzirom da je brzina zvuka veća, što znači da će se val brže putovati do objekta i nazad, te će samim time i brojač kraće brojati. Kada tu vrijednost brojača na većoj temperaturi, uvrstimo u formulu kalibriranu prema manjoj brzini zvuka, dolazi do različitih rezultata u odnosu na stvarne vrijednosti. Vrijednosti su manje u odnosu na stvarne. Matematički to možemo prikazati na sljedeći način pomoću formule (2-2):

$$udaljenost\ [\mathrm{m}] = \frac{vrijeme\ [\mathrm{s}]\ *\ 343\ [\mathrm{m/s}]}{2} \qquad [5\text{-}1]$$

Za udaljenost od 40 cm pri brzini zvuka od približno 343 m/s:

$$0.4\ [\mathrm{m}] = \frac{vrijeme\ [\mathrm{s}] * 343\ [\mathrm{m/s}]}{2} \qquad [5\text{-}2]$$

$$vrijeme\ [\mathrm{s}] = 0,0023324 \qquad [5\text{-}3]$$

Za udaljenost od 40 cm pri brzini zvuka od približno 355 m/s:

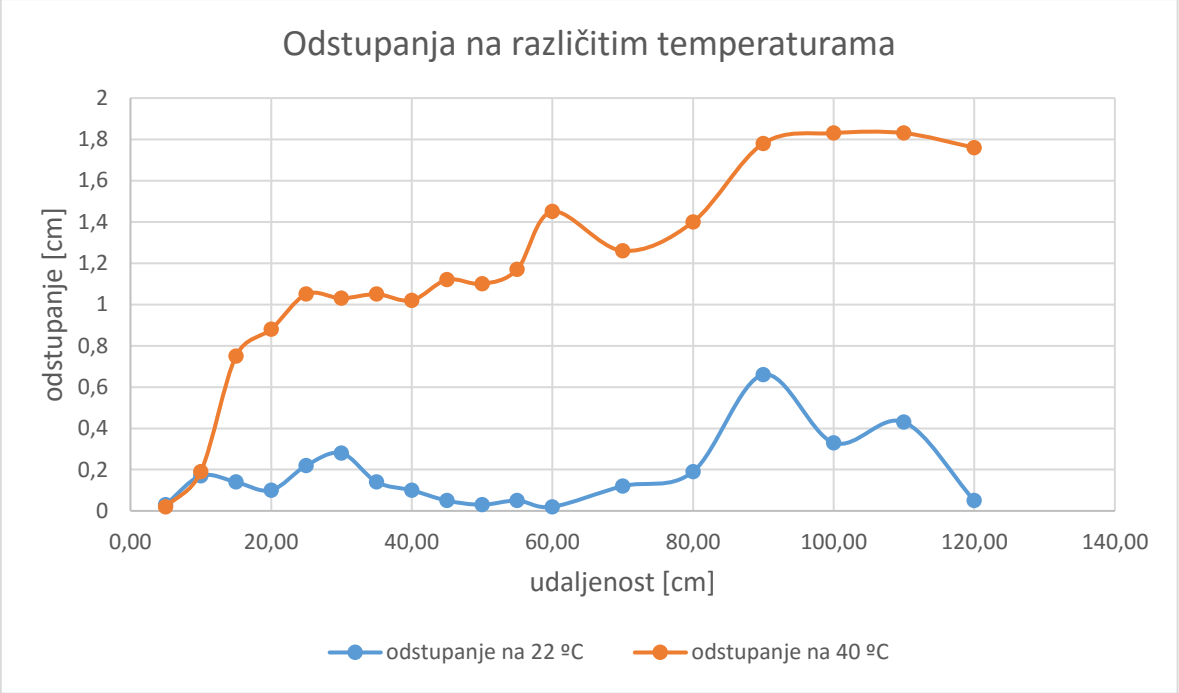
$$vrijeme\ [\mathrm{s}] = 0,0022535 \qquad [5\text{-}4]$$

Kada dobivenu vrijednost brojača na 355 m/s, uvrstimo u formulu kalibriranu na brzinu od 343 m/s dolazi do odstupanja.

$$udaljenost\ [\mathrm{m}] = \frac{0,0022535\ [\mathrm{s}] * 343\ [\mathrm{m/s}]}{2} = 0,3865 \qquad [5\text{-}5]$$

Dobiveni rezultat je 38,65 cm. Pogledamo li u tablicu jasno je vidljivo da vrijednosti nemaju preveliku razliku za udaljenost od 40 cm, odnosno da su približno iste mjerena vrijednost i vrijednost dobivena matematičkim računanjem. S ovime je potvđen razlog postojanja razlike u mjerenjima na različitim temperaturama.

Pomoću grafa prikazana su odstupanja rezultata od stvarnih vrijednosti. Stvarne i mjerene vrijednosti su oduzete, a dobiveni rezultat je prikazan u grafu ispod.



Graf 5.1 *Odstupanja od zadanih vrijednosti na različitim temperaturama*

U grafu 5.1. prikazane su odstupanja mjerenja na različitim temperaturama od stvarnih vrijednosti udaljenosti. Iz grafa je vidljivo da za temperaturu od 22 ºC vrijednosti nemaju velika odstupanja, te su mjerenja blizu stvarnih vrijednosti. Najveće odstupnje je na udaljenosti od 90 cm, a najmanje na 60 cm. Srednje odstupanje iznosi 0,17 cm.

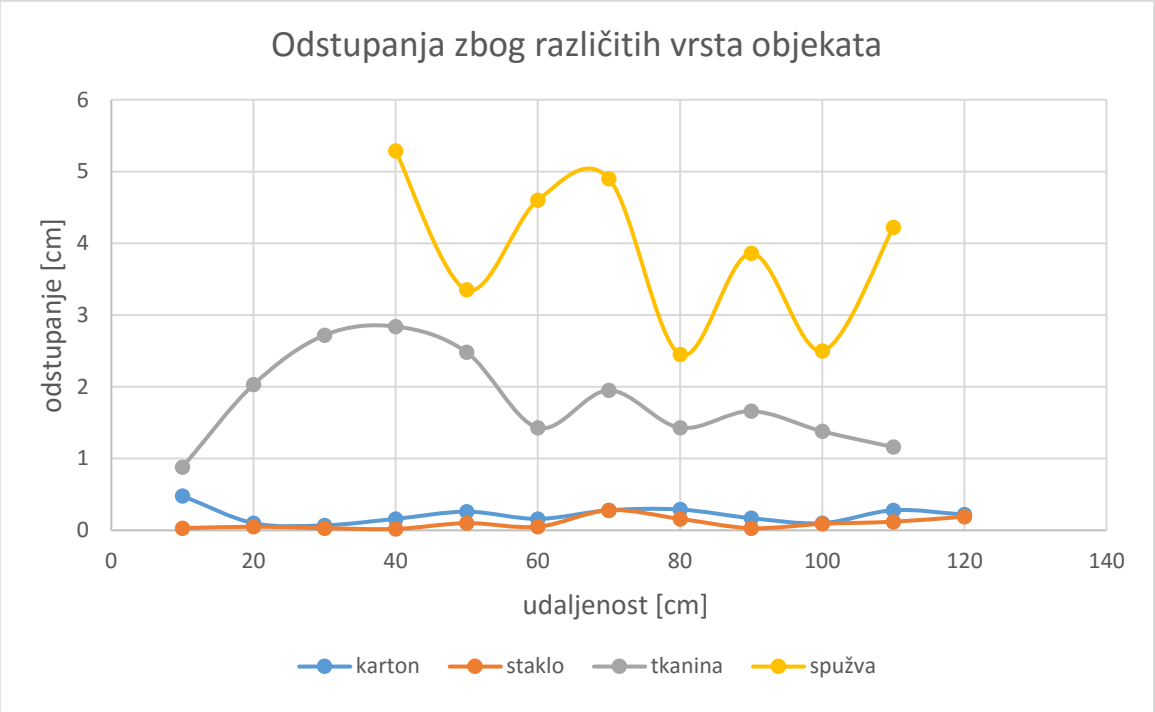
Na temperaturu od 40 ºC postoje veća i značajnija odstupanja, za razliku od odstupanja na 22 ºC. Iz grafa je vidljivo da sa povećanjem udaljenosti, povećava se lagano i odstupanje. Najmanje odstupanje je na udaljenosti od 5 cm i 10 cm, a najveće je između 90 cm i 120 cm. Srednje odstupanje za temperaturu 40 ºC iznosi 1,15 cm. Vidljiva je razlika srednjih vrijednosti odstupanja na različitim temperaturama, što je uzrok različitih brzina zvuka na različitim temperaturama.

udaljenost [cm]	karton [cm]	staklo [cm]	tkanina [cm]	spužva [cm]
10	10.48	9.97	10.88	
20	20.1	19.95	22.03	
30	30.07	29.97	32.72	
40	39.84	39.98	42.84	45.29
50	49.74	50.1	52.48	53.35
60	59.84	60.05	61.43	64.6
70	69.72	69.72	71.95	74.9
80	79.71	80.16	81.43	82.45
90	89.83	90.03	91.66	93.86
100	99.9	100.09	101.38	102.5
110	109.72	110.12	111.16	114.22
120	119.78	119.81		

Tablica 5.2. *Rezultati mjerenja sa različitim objektima*

Tablica 5.2. prikazuje rezulatate mjerenja sa različitim vrstama objekata na umjerenoj temperaturi od 22 °C i pri istim uvjetima. Mjerenja su rađena sa kartonom, staklom, tkaninom i spužvom, te u razmacima od 10 cm. Uzeto je 12 uzoraka mjerenja. Prema rezultatima iz tablice zaključeno je da najbolje rezultate daje staklo kao objekt odbijanja ultrazvučnog vala. Njegove mjerene vrijednosti imaju minimalna i zanemariva odstupanja od stvarnih vrijednosti. Karton ima približno slične vrijednosti kao i staklo, iako lošije, zanemarivog su odstupanja uzimajući u obzir da kada parkiramo, ne približavamo se na 1 cm od objekta, nego se držimo na sigurnijoj udaljenosti. Preostala dva materijala, tkanina i spužva, imaju osjetno veća odstupanja koja mogu imati velik utjecaj ako se približavamo istima prilikom parkiranja. Mjerene vrijednosti koje su dobivene koristeći tkaninu kao objekt odbijanja vala se kreću između 0,8 cm i 2,84 cm. Iz toga se može zaključiti da vrijednosti varijaju i da su ipak dosta velika odstupanja uspoređujući sa staklom i kartonom. Vrijednost na 120 cm se nije mogla očitati, jer se val izgubio prilikom reflektiranja i vraćanja na senzor. Na kraju, zadnji objekt koji je korišten za reflektiranje, spužva, ima najveća odstupanja te se za nekoliko udaljenosti nije mogla niti očitati vrijednost. Iz tablice je vidljivo da za prva tri uzorka mjerenja nema očitanja, spužva je apsorbirala većinu poslanog vala. Gledajući dalje tablicu, odstupanja za preostale udaljenosti su između 2,45 cm i 5,29 cm, s tim da za zadnju udaljenost nema očitanja. To su najveća odstupanja od svih vrsta objekata korištenih pri mjerenjima, te kao takva su prevelika i nisu prihvatljiva u svrsi parking senzora. Ovim mjerenjima su prikazane mane ultrazvučnog parking senzora s obzirom na vrstu materijala od kojih se val reflektira. Međutim, kada vozač parkira automobil, objekti na koje mora paziti najviše su betonski zid ili drugi automobil pokraj kojeg parkira, a mjerenja napravljena s čvrstim objektima su dala

najbolje i najpreciznije rezultate. S obzirom na to, možemo reći da je ovaj parking senzor prikladan, precizan i iskoristiv u stvarnom okruženju.



Graf 5.2. *Odstupanja od zadanih vrijednosti zbog različitih vrsta objekata reflektiranja*

Kao što je objašnjeno prije, graf 5.2. prikazuje odstupanja od stvarnih vrijednosti za sve objekte na kojima su napravljena mjerenja. Vidljivo je da staklo i karton daju najbolje rezulate, približno jednakih odstupanja. Tkanina kao objekt reflektiranja vala je na granici s obzirom na parking senzor. Ima prilično velika odstupanja, međutim senzor tkaninu detektira, što znači da će upozoriti vozača da se približava objektu, ali sa manjom preciznosti. Na kraju, spužva kao objekt reflektiranja vala ima daje najlošije rezultate i ima najveća odstupanja, s tim da da na nekim udaljenosti ni ne dolazi do očitanja udaljenosti. To se događa jer je spužva šupljikav objekt koji većinu vala apsorbira. Prema tome, ultrazvučni parking senzor ne zadovoljava uvjete s obzirom na potrebnu preciznost i očitanja kada je u pitanju spužvasti objekt.

6. ZAKLJUČAK

Cilj rada je bio napraviti sklop koji mjeri udaljenost i ispisuje ju na zaslon LCD-a. Da bi to bilo ostvareno, rad je podijeljen na dva glavna dijela, hardverski i softverski.

U hardverskom dijelu opisane su sve komponente potrebne da bi sklop obavljao svoju zadaću. Glavna komponenta je ultrazvučni senzor udaljenosti, *HC-SR04*. Odabran je zbog svoje velike raspostranjenosti i uporabe u automobilima današnjice, ali i preciznosti. Osim toga ima velik omjer uloženog i dobivenog, jer cijena mu je niska, a obavlja posao vrlo dobro. Da bi senzor pravilno mjerio udaljenost potreban je mikroupravljač koji će upravljati s njim. Mikroupravljač je Atmega16, proizvođača Atmel. Ostale komponente su 16x2 LCD i zujalica.

Softverski dio odnosi se najviše na programsku podršku odnosno kod, kojim se preko mikroupravljača upravlja radom cijelog sklopa. Osim koda, bitne su i postavke mikroupravljača. U kodu su napisane razne funkcije, a najvažnije dijelovi su pokretanje *triggera* ultrazvučnog senzora, zatim prekidne rutine koje se aktiviraju kada *echo* pin pređe u stanje jedinice. Time se aktivira brojač, koji mjeri vrijeme potrebno da se val reflektira od objekta i vrati na senzor. Kada *echo* pin pređe u stanje nule, brojač se deaktivira, a njegova vrijednost se sprema u registar. Kod sadrži i funkcije koje omogućuju prikaz udaljenosti na zaslon. Osim ispisa vrijednosti udaljenosti u cm, ispisuju se i graf koji zajedno sa prikazom udaljenosti pomaže i olakšava vozaču prilikom parkiranja automobila.

Nakon osposobljavanja sklopa, izvršena su razna testiranja istog. Napravljena su mjerenja na raznim temperaturama, te mjerenja sa različitim objektima od kojih se val reflektira. Mjerenje udaljenosti senzora je u kodu kalibrirano i postavljeno u odnosu na umjerenu temperaturu i brzinu zvuka na toj temperaturi (22 °C). Prema tome, mjerenja na temperaturi od 22 °C su jako precizna i imaju mala odstupanja. Međutim, mjerenja na temperaturi od 40 °C imaju odstupanja, ali ne prevelika. To se događa iz razloga povećanja brzine zvuka zbog povećanja temperature. Mjerenjima na raznim objektima je utvrđeno da čvršći i gušći objekti bolje reflektiraju val nazad na senzor. Zbog toga, mjerenja na staklu i kartonu imaju mala odstupanja, dok mjerenja na tkanini i spužvi daju loše vrijednosti, te mogu imati negativan utjecaj prilikom parkiranja automobila.

Zaključak je da je parking senzor prikladan za uporabu u stvarnom okruženju iz razloga što prilikom parkiranja automobila, u najviše slučajeva potrebno je paziti na druge automobile ili betonski zid, a to su sve čvrsti objekti. Sustav je ispunio očekivanja, precizan je i vrlo lako se koristi.

LITERATURA

[1] „CBT“, How parking sensors work?, https://carbiketech.com/parking-sensors/, (11.05.2018.)

[2] „do it yourself“, How do reverse parking sensors work?, https://www.doityourself.com/stry/how-do-reverse-parking-sensors-work, (11.05.2018.)

[3] „Wikipedia“, Parking sensor, https://en.wikipedia.org/wiki/Parking_sensor, (11.05.2018.)

[4] URL slike: http://inrix.com/blog/2017/12/ultrasonic-sensor-parking-availability-technology/, (11.05.2018.)

[5] URL slike: https://www.parkingdynamics.co.uk/Universal-Sensors/Parking-Dynamics-PD1/Parking-Dynamics-PD1-Parking-Sensor, (11.05.2018.)

[6] „intorobotics“, Types of sensors for target detection and tracking, https://www.intorobotics.com/types-sensors-target-detection-tracking/, (12.05.2018.)

[7] URL slike: http://www.themakersworkbench.com/tutorial/triggering-servo-using-hc-sr04-distance-sensor-and-arduino, (12.05.2018.)

[8] URL slike: https://www.researchgate.net/figure/IR-proximity-sensor-working-principle_fig2_301787143, (12.05.2018.)

[9] „OMRON“, Photoelectric sensors, https://www.ia.omron.com/support/guide/43/introduction.html, (12.05.2018.)

[10] URL slike: https://www.keyence.com/ss/products/sensor/sensorbasics/photoelectric/info/

[11] „Components101“, HC-SR04 Ultrasonic Sensor, https://components101.com/ultrasonic-sensor-working-pinout-datasheet, (15.05.2018.)

[12] „Acmesystems“, HC-SR04 ultrasonic sensor, https://www.acmesystems.it/HC-SR04, (15.05.2018.)

[13] „Components101“, Ultrasonic Ranging Module HC - SR04, https://components101.com/sites/default/files/component_datasheet/HCSR04%20Datasheet.pdf, (15.05.2018.)

[14] „AVR tutorials“, Microcontrollers Basics, http://www.avr-tutorials.com/general/microcontrollers-basics, (19.05.2018.)

[15] „EngineersGarage“, AVR Microcontroller, https://www.engineersgarage.com/articles/avr-microcontroller, (19.05.2018.)

[16] „electronicsforu“, Atmega16 pin diagram & description, https://electronicsforu.com/resources/learn-electronics/Atmega16-pin-diagram-description, (19.05.2018.)

[17] URL slike: https://components101.com/microcontrollers/Atmega16-pinout-features-datasheet

[18] „Microcontrollers Lab“, Atmega16 Microcontroller, http://microcontrollerslab.com/Atmega16-microcontroller/, (19.05.2018.)

[19] „EngineersGarage“, LCD, https://www.engineersgarage.com/electronic-components/16x2-lcd-module-datasheet, (20.05.2018.)

[20] „CircuitDigest“, 16x2 LCD Display Module, https://circuitdigest.com/article/16x2-lcd-display-module-pinout-datasheet, (20.05.2018.)

[21] „e-radionica“, Buzzer, https://e-radionica.com/hr/blog/2015/08/19/buzzer/, (20.05.2018.)

[22] „Microchip“, Atmel Studio 7, https://www.microchip.com/mplab/avr-support/atmel-studio-7, (25.05.2018.)

[23] „EngineersGarage“, AVR Atmega16/32 Fuse Bits, https://www.engineersgarage.com/tutorials/avr-Atmega16-fuse-bits?page=1, (25.05.2018.)

[24] „AVR Tutorials“, AVR Microcontrollers Interrupt, http://www.avr-tutorials.com/interrupts/about-avr-8-bit-microcontrollers-interrupts, (26.05.2018.)

[25] „Atmega16 datasheet“, http://ww1.microchip.com/downloads/en/DeviceDoc/doc2466.pdf, (26.05.2018.)

[26] „maxEmbedded“, Introduction to AVR Timers, https://maxembedded.wordpress.com/2011/06/22/introduction-to-avr-timers/, (26.05.2018.)

SAŽETAK

Zbog sve većeg broja automobila i nedostatka parkirnih mjesta, posebice u gradovima, razvila se potreba za parking senzorima kako bi se olakšalo parkiranje u uskim i malim prostorima. U tom smjeru napravljen je i ovaj rad, odnosno sklop koji mjeri udaljenosti i ispisuje ju na zaslon, parking senzor. Glavna komponenta sklopa je ultrazvučni senzor udaljenosti. On ima široku primjenu, kako u elektronici kao komponenta, tako i u većini današnjih automobila kao parking senzor. Sklop se sastoji od sljedećih komponenti: Atmega16 mikroupravljač, senzor udaljenosti *HC-SR04*, zujalica i 16x2 LCD. Kako bi sklop služio kao parking senzor napisan je funkcionalan kod koji to omogućuje. Sustav je na kraju i testiran. Testiranjem je utvrđeno da je sklop precizan i točan, posebice na umjerenim temperaturama (22 ºC). Mala odstupanja su na temperaturama iznad 40 ºC. Također mjerenja ovise o svojstvu materijala od kojeg se val odbija. Što je gušći i tvrđi materijal, mjerenje je točnije. Kada parkiramo automobil, objekti koji nam smetaju su većinom zid ili drugo vozilo. S obzirom na to, možemo reći da je sklop prikladan za uporabu u stvarnom okruženju.

Ključne riječi: Atmega16, parking senzor, ultrazvučni senzor, LCD, udaljenost

SUMMARY

Due to the increasing number of cars and the lack of parking spaces, especially in cities, the need for parking sensors has evolved to facilitate parking in narrow and small spaces. In this direction, this work was also made, device which measures the distance and writes it on the screen. The main component of the device is the ultrasonic distance sensor. It has a wide application, both in electronics as a component and in most modern cars as a parking sensor. The device consists of the following components: Atmega16 microcontroller, *HC-SR04* distance sensor, buzzer and 16x2 LCD. In order for the device to function as a parking sensor, a functional code is written that allows it. The system was finally tested. Testing found that the device is accurate, especially at moderate temperatures (22 ºC). Small deviations are at temperatures above 40 ºC. Measurements also depend on the properties of the material from which the wave is reflected. The more denser and harder the material, the measurement is more accurate. When we park the car, objects that bother us are mostly wall or another car. Considering this, we can say that the device is suitable for use in the real environment.

Key words: Atmega16, parking sensor, ultrasonic sensor, LCD, distance

ŽIVOTOPIS

Marin Šimunović rođen je 11.10.1994. godine u Tešnju, Bosna i Hercegovina. Osnovnu školu pohađao je u Usori, BiH. Tijekom osnovne škole dobio je razna priznanja za odličan uspjeh i uzorno vladanje, te je na kraju školovanja proglašen učenikom generacije. Nakon osnovne škole upisao je Gimnaziju također u Usori. Gimnazijsko školovanje završava 2013. godine sa odličnim uspjehom te iste godine upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. U roku završava preddiplomski studij i postaje inženjer elektrotehnike. Aktivno se služi računalom i internetom, dobro poznaje C programski jezik, kao i rad u mnogim programima korištenim na fakultetu (Matlab, HFSS, Cisco Packet Tracer). Aktivno se služi engleskim jezikom.

PRILOZI

Programska podrška (kod) za rad parking senzora:

```
#include <avr/io.h>
#include <avr/interrupt.h>
#define F_CPU 1000000
#define DL _delay_ms(10);
#include <util/delay.h>
#include <stdlib.h>

void lcd_init();
void lcd_cmd(char cmd);
void lcd_data(char data);
void lcd_string(char *str);
void lcd_Set_Cursor(char a, char b);
void LCD_Custom_Char (unsigned char loc, unsigned char *str);
void beep(int nbeep);
void beep2(int nbeep);
void beep3(int nbeep);

static volatile int pulse = 0;
static volatile int i = 0;

int main(void)
{
     lcd_init();
unsigned char p0[8]={0x00,0x00,0x00,0x00,0x00,0x01,0x03,0x07};
unsigned char p1[8]= {0x00,0x00,0x00,0x00,0x00,0x10,0x18,0x1C};
unsigned char p2[8]= {0x07,0x03,0x01,0x00,0x00,0x00,0x00,0x00};
unsigned char p3[8]= {0x1C,0x18,0x10,0x00,0x00,0x00,0x00,0x00};
unsigned char p4[8]= {0x00,0x00,0x00,0x10,0x18,0x0C,0x04,0x06};
unsigned char p5[8]= {0x06,0x04,0x0C,0x18,0x10,0x00,0x00,0x00};
unsigned char p6[8]= {0x10,0x08,0x0C,0x06,0x02,0x03,0x03,0x03};
unsigned char p7[8]= {0x03,0x03,0x03,0x02,0x06,0x0C,0x08,0x10};
unsigned char p8[8]= {0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00};
```

```
 LCD_Custom_Char(0, p0);
 LCD_Custom_Char(1, p1);
 LCD_Custom_Char(2, p2);
 LCD_Custom_Char(3, p3);
 LCD_Custom_Char(4, p4);
 LCD_Custom_Char(5, p5);
 LCD_Custom_Char(6, p6);
 LCD_Custom_Char(7, p7);
 LCD_Custom_Char(8, p8);

DDRA = 0b00000001;
DDRB = 0xFF;
DDRD = 0b11111011;
PORTD =(0<<PD1);

_delay_ms(50);

GICR|=(1<<INT0);
MCUCR|=(1<<ISC00);
sei();

double COUNTA = 0;
char SHOWA [16];

while(1)
{
     PORTD|=(1<<PIND0);
     _delay_us(15);
     PORTD &=~(1<<PIND0);

     COUNTA = (double)pulse/58;
     dtostrf(COUNTA, 2, 2, SHOWA);


     if (COUNTA >= 120){
```

```
        lcd_Set_Cursor(1, 0);
        lcd_string("                ");
        lcd_Set_Cursor(2, 0);
        lcd_string("                ");
        PORTA = (0<<PA0);
    }
    else{
        lcd_Set_Cursor(1,0);
        lcd_string("Daljina:");
        lcd_Set_Cursor(2, 0);
        lcd_string(SHOWA);
        lcd_string("cm");
        PORTA = (1<<PA0);
    }
        if (COUNTA <= 20)
        {
            lcd_Set_Cursor(1,10);
            lcd_data(0);
            lcd_data(1);
            lcd_string(" ");
            lcd_string(" ");
            lcd_string(" ");
            lcd_string(" ");

            lcd_Set_Cursor(2,10);
            lcd_data(2);
            lcd_data(3);
            lcd_string(" ");
            lcd_string(" ");
            lcd_string(" ");
            lcd_string(" ");
            beep(3);
        }


        else if (COUNTA <= 40)
```

```
{
     lcd_Set_Cursor(1, 10);
     lcd_data(0);
     lcd_data(1);
     lcd_data(4);
     lcd_string(" ");
     lcd_string(" ");
     lcd_string(" ");

     lcd_Set_Cursor(2, 10);
     lcd_data(2);
     lcd_data(3);
     lcd_data(5);
     lcd_string(" ");
     lcd_string(" ");
     lcd_string(" ");
     beep2(3);
}

else if (COUNTA <= 60)
{
     lcd_Set_Cursor(1, 10);
     lcd_data(0);
     lcd_data(1);
     lcd_data(4);
     lcd_data(6);
     lcd_string(" ");
     lcd_string(" ");

     lcd_Set_Cursor(2, 10);
     lcd_data(2);
     lcd_data(3);
     lcd_data(5);
     lcd_data(7);
     lcd_string(" ");
     lcd_string(" ");
```

```
     beep3(3);
}

else if (COUNTA <= 80)
{
     lcd_Set_Cursor(1, 10);
     lcd_data(0);
     lcd_data(1);
     lcd_data(4);
     lcd_data(6);
     lcd_data(6);
     lcd_string(" ");

     lcd_Set_Cursor(2, 10);
     lcd_data(2);
     lcd_data(3);
     lcd_data(5);
     lcd_data(7);
     lcd_data(7);
     lcd_string(" ");
}

else if (COUNTA < 120)
{
     lcd_Set_Cursor(1, 10);
     lcd_data(0);
     lcd_data(1);
     lcd_data(4);
     lcd_data(6);
     lcd_data(6);
     lcd_data(6);

     lcd_Set_Cursor(2, 10);
     lcd_data(2);
     lcd_data(3);
     lcd_data(5);
```

```
                    lcd_data(7);
                    lcd_data(7);
                    lcd_data(7);
               }
          }
}

ISR(INT0_vect)
{
     if (i==1)
     {
          TCCR1B=0;
          pulse=TCNT1;
          TCNT1=0;
          i=0;
     }
          if (i==0)
          {
               TCCR1B|=(1<<CS10);
               i=1;
          }
}

void lcd_init(){

     DDRB=0xff;
     DDRC|=(1<<0)|(1<<1)|(1<<2);
     lcd_cmd(0x38);
     lcd_cmd(0x01);
     lcd_cmd(0x0e);
     lcd_cmd(0x80);
}

void lcd_cmd(char cmd){
     PORTB=cmd;
     PORTC|=(1<<2);
```

```
     PORTC&=~((1<<0)|(1<<1));
     DL;
     PORTC&=~(1<<2);
     DL;
}

void lcd_data(char data){
     PORTB=data;
     PORTC|=(1<<0)|(1<<2);
     PORTC&=~(1<<1);
     DL;
     PORTC&=~(1<<2);
     DL;

}

void lcd_string(char *str){

     while(*str !='\0'){
          lcd_data(*str++);
     }
}
void lcd_Set_Cursor(char a, char b)
{
     if(a == 1)
     lcd_cmd(0x80 + b);
     else if(a == 2)
     lcd_cmd(0xC0 + b);
}

void LCD_Custom_Char (unsigned char loc, unsigned char *str)
{
     unsigned char i;
     if(loc<8)
     {
          lcd_cmd (0x40 + (loc*8));
```

```
          for(i=0;i<8;i++)
          lcd_data(str[i]);
     }
}

void beep(int nbeep)
{
     while(nbeep--){
          PORTD = (1<<PD1);
          _delay_ms(50);
          PORTD  = (0<<PD1);
          _delay_ms(20);
     }
}

void beep2(int nbeep)
{
     while(nbeep--){
          PORTD = (1<<PD1);
          _delay_ms(100);
          PORTD  = (0<<PD1);
          _delay_ms(80);
     }
}

void beep3(int nbeep)
{
     while(nbeep--){
          PORTD = (1<<PD1);
          _delay_ms(150);
          PORTD  = (0<<PD1);
          _delay_ms(120);
     }
}
```