

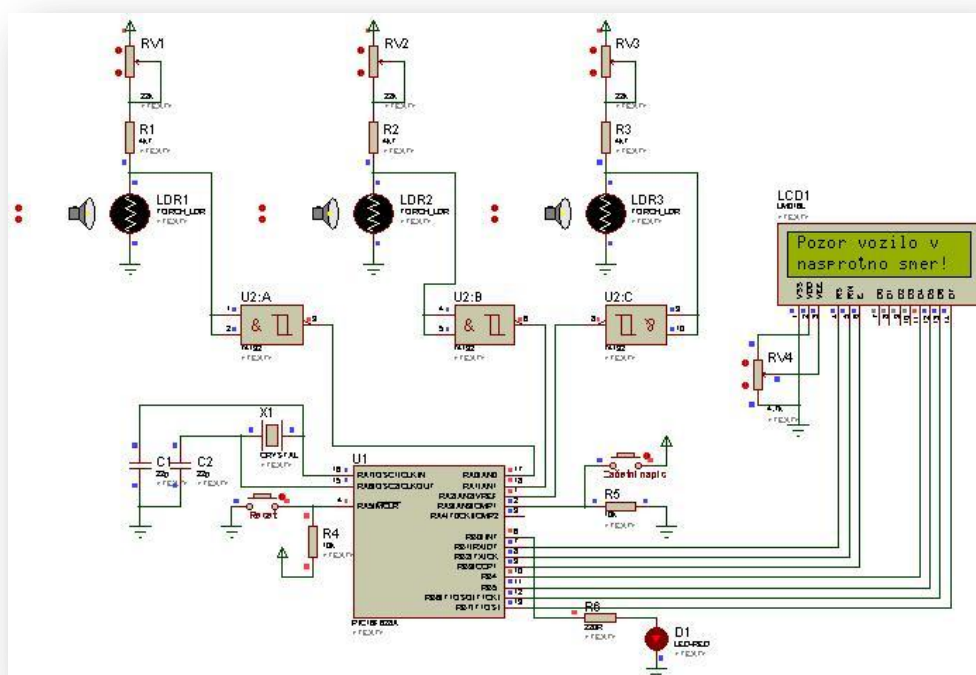
»Mladi za napredek Maribora 2013«

30. srečanje

Pripomoček za varnejšo vožnjo na avtocestah

Raziskovalno področje: Elektrotehnika, elektronika

Inovacijski predlog



0e q ! k V Š O P Á S W Ü P S

T ^ } q ! k Á Š O P Á X Q

¥ [| a Á Ü Ö Ö P R O Z O Š O S V Ü U E Ü C E W P O Š P Q S O Á U Š O Á T O E Ü O U Ü

Maribor, januar 2013

Vsebina

1. POVZETEK	4
2. ZAHVALA.....	4
3. VSEBINSKI DEL	5
3.1. Opredelitev problema	5
3.2. Postopek razvijanja inovacijskega predloga.....	5
3.3. Opis izdelanega modela – inovacijskega predloga.....	7
3.4. Tehnična in tehnološka dokumentacija	9
3.4.1 Elektronski načrt z mikrokontrolerjem	9
3.4.2 LCD prikazovalnik	10
3.4.3 Mikrokontroler PIC16F628a	13
3.4.4 Diagram poteka	15
3.4.5 Program za krmiljenje naprave	16
4. INVESTICIJA PROJEKTA	21
5. DRUŽBENA ODGOVORNOST.....	21
6. SKLEP.....	21
7. VIRI.....	22

Kazalo slik

Slika 1: Prikaz napisa na LCD zaslonu, ko ni nevarnosti zaradi vozila v napačni smeri (vir: avtor naloge).....	8
Slika 2: Napis na LCD zaslonu, ko nastane nevarnost zaradi vozila v napačno smer (vir: avtor naloge).....	8
Slika 3: Elektronski načrt vezja za krmiljenje naprave (vir: avtor naloge).....	10

Slika 4: LCD 2 x 16, naslovi DDRAM - šestnajstiške vrednosti (vir: avtor naloge).....	11
Slika 5: Priključitev LCD zaslona na mikrokontroler (vir: avtor naloge).....	12
Slika 6: Mikrokontroler PIC16F628a [1]	14
Slika 7: Določanje vhodno-izhodnih pinov mikrokontrolerja (vir: avtor naloge)	17
Slika 8: Meritev časa zakasnitve v okolju MPLAB IDE (vir: avtor naloge)	20
Slika 9: Prevajanje programa in zapis v mikrokontroler (vir: avtor naloge).....	20

1. POVZETEK

Inovacijski predlog pripomore k večji varnosti udeležencev v prometu, ki se odvija na avtocestah.

Vsako leto se zgodi veliko prometnih nesreč zaradi vožnje v napačni smeri, kljub temu, da so na priključkih avtoceste postavljene opozorilne table.

Inovacijski predlog rešuje problem na podlagi uporabe senzorjev, ki na priključkih avtoceste ugotovijo smer vožnje. Če je le ta napačna, se na najbližjih opozorilnih tablah pojavi posebno opozorilo, ki udeležence v prometu obvešča o prežeči nevarnosti, ker se jim približuje vozilo iz nasprotne, napačne smeri. Vozniku, ki vozi v napačno smer, pa se vklopi svetlobni opozorilni znak, ki ga obvešča, da vozi v napačno smer.

Za nalogo je uporabljen mikrokontroler, ki neprestano preverja smer vožnje na priključku avtoceste na podlagi senzorjev, ki so montirani pod voziščem. V primeru, da ugotovi napačno smer vožnje, se prikaže ustrezno opozorilo na LCD¹ prikazovalniku in vklopi svetlobni opozorilni znak.

2. ZAHVALA

Za nasvet, pomoč in potrpljenje se zahvaljujem mojemu mentorju. Nekatera dela so mi olajšali tudi sošolci, ki so me spodbujali in mi tudi kaj svetovali, zato se vsakemu posebej zahvaljujem. Zahvaliti se pa moram tudi staršem, ki so me spodbujali in mi pomagali pri izdelavi makete.

¹ LCD (angl.: Liquid Crystal Display), prikazovalnik na tekoče kristale.

3. VSEBINSKI DEL

3.1. Opredelitev problema

Vožnja v nasprotni smeri je izredno nevarna, prepovedana je tudi z zakonom. Ta določa, da je na cesti, ki ima dva ali več označenih prometnih pasov za vožnjo v eno smer, prepovedano voziti v nasprotni smeri.

Po podatkih Ministrstva za notranje zadeve (vir: Policija) so najpogostejši razlogi za vožnjo v nasprotno smer po avtocestah:

- namerna vožnja v nasprotno smer (neupoštevanje prometne signalizacije, objestna vožnja, stave),
- vožnja pod vplivom alkohola ali mamil oziroma drugih psihoaktivnih snovi (voznik se ne zaveda svojega dejanja),
- slabše psihofizične sposobnosti za vožnjo (zlasti nekateri starejši občani niso vajeni vožnje po avtocestah in nehote zapeljejo v nasprotno smer po avtocesti).

Leta 2009 je bilo na slovenskih avtocestah 155 prometnih nesreč zaradi vožnje v nasprotni smeri, v katerih je skupno umrlo pet ljudi, v 55 nesrečah pa so se udeleženci poškodovali.

Da bi se pozornost voznikov povečala in da bi se število nesreč zmanjšalo, sem prišel do ideje dodatno opozarjati voznike na pretečo nevarnost z mojim inovacijskim predlogom. Uvedba predloga bi voznike na avtocestah ustrezno opozorila na nasproti vozeče vozilo, udeleženca, ki vozi na priključku avtoceste v nasprotno smer, pa opozorila na napačno smer vožnje.

3.2. Postopek razvijanja inovacijskega predloga

Ko sem prišel do ideje, sem se moral odločiti, kako idejo uresničiti oziroma jo izdelati na modelu. V mislih sem imel predvsem učinkovito opozarjanje voznikov na pretečo nevarnost. Pred vsakim izvozom iz avtoceste je postavljena opozorilna tabla, ki vsebuje določene informacije za voznike. Na teh tablah bi lahko voznike obvestili, da se jim iz nasprotne smeri približuje vozilo. Ti bi na podlagi obvestila zmanjšali hitrost vožnje, vključili vse štiri smerne

kazalce ter vozili še posebej previdno. Vozili bi po desnem smernem vozišču in ne bi prehiteli. S takšnim ravnanjem bi zmanjšali število prometnih nesreč. Posledično bi bilo manj žrtev in poškodovanih, manjša pa bi bila tudi materialna škoda, ki ob takih nezgodah nastane.

Postavilo se mi je vprašanje, kako ugotoviti napačno smer vožnje? Uporabiti bi moral ustrezne senzorje, na podlagi katerih bi lahko ugotovil smer vožnje. Na podlagi dejstva, da so avtomobili in vsa vozna sredstva narejena iz kovin, bi bila najustreznejša uporaba induktivnih senzorjev. V praksi bi takšni senzorji prekrivali celotni vozni pas. Pri izbiri senzorjev sem razmišljal tudi v smeri uporabe fotoupora ali pa IR senzorjev. Vendar bi lahko pri teh senzorjih prišlo do zlorabe in neustreznega delovanja. Zaradi visoke cene induktivnih senzorjev, ki bi bili nedvomno najustreznejši, sem za model izbral fotoupore, saj je tudi z njihovo uporabo moč prikazati delovanje inovacijskega predloga. Uporabil sem tri fotoupore, predvsem zaradi zanesljivosti delovanja.

Nadalje se mi je postavilo vprašanje, kako prikazati ustrezne opozorilne napise za voznike na avtocesti? Izmed možnosti, ki sem jih imel v mislih, sem se odločil obvestilo prikazati na dvovrstičnem LCD prikazovalniku. Kadar ni nevarnosti, bo prikazoval *Izvoz Celje jug 1000 m*, ko pa nastane nevarnost, bo LCD prikazovalnik prikazoval obvestilo *Pozor vozilo v nasprotno smer!*

Pri izdelavi naloge sem seveda naletel na problem krmiljenja naprave. Kako ga izvesti? Ob pomoči mentorja in znanja, ki sem ga pridobil na krožku v šoli, sem se odločil, da bom sistem krmilil z mikrokontrolerjem PIC. Izbral sem Microchipov mikrokontroler PIC16F628a. Pri izdelavi krmilja sem imel kar nekaj težav in problemov. Na podlagi algoritma, ki sem si ga izdelal, sem se lotil programiranja mikrokontrolerja. Glavna naloga mikrokontrolerja je ta, da na določenih vhodnih priključkih neprestano preverja logična stanja, in da na določene izhodne priključke pošilja ustrezna logična stanja, odvisno od izdelanega programa. Na principu tega znanja, sem začel pisati program za krmiljenje naprave. Posebno poglavje je krmiljenje LCD prikazovalnika. Ob pomoči mentorja mi je uspela pravilna rešitev problema.

Naloga mikrokontrolerja je, da preverja vrstni red "pokrivanja" senzorjev in da na podlagi te informacije prikazuje ustrezni napisa na LCD prikazovalniku. Krmilno vezje za

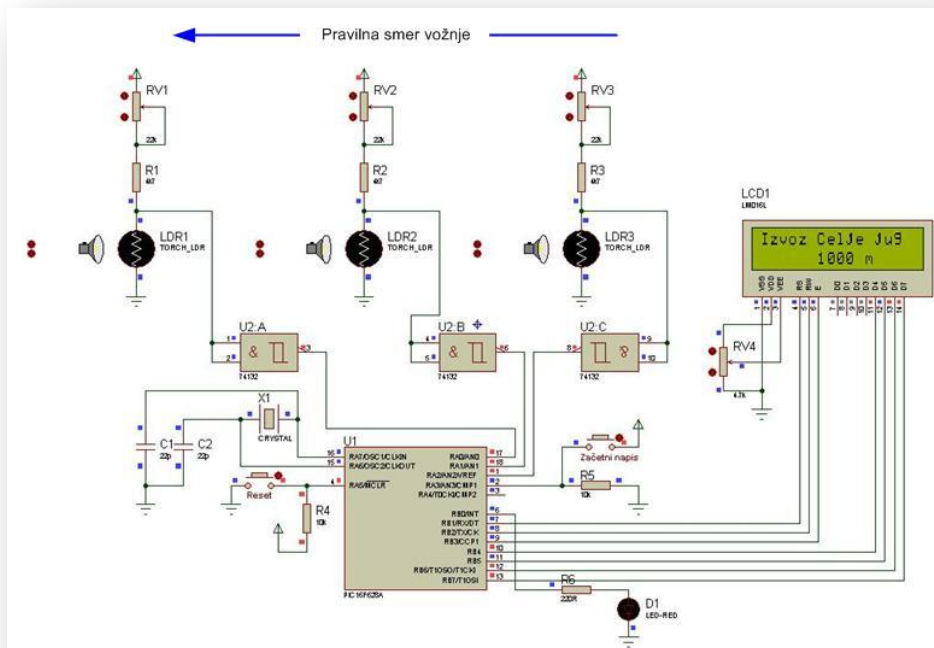
mikrokontroler s pripadajočimi elementi krmilja sem izdelal na tiskanini. Tiskano vezje sem konstruiral s programskim okoljem Eagle in ga izdelal po fotopostopku.

Za demonstracijo inovacijskega predloga sem izdelal model, ki prikazuje priključek na avtocesto z vgrajenimi senzorji, LCD prikazovalnik za obveščanje voznikov, ki vozijo po avtocesti in svetlobni znak, postavljen na izvozu iz avtoceste, ki opozarja voznika na napačno smer vožnje. Menim, da je tudi s tako izdelanim modelom moč videti delovanje mojega inovacijskega predloga in ugotoviti njegove prednosti.

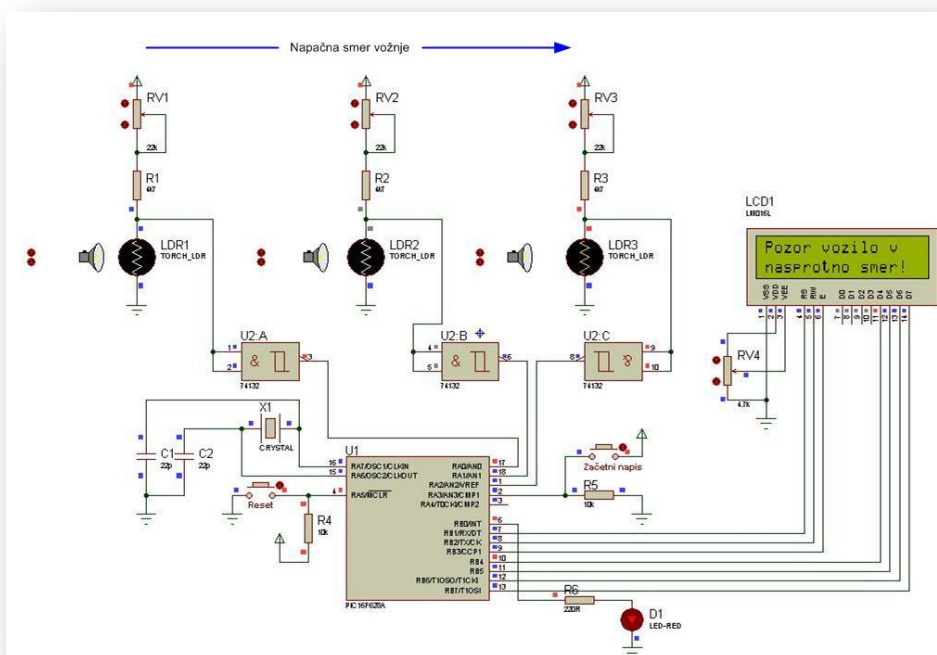
3.3. Opis izdelanega modela – inovacijskega predloga

Model je sestavljen iz avtoceste in priključka za izvoz iz avtoceste. LCD prikazovalnik je postavljen pred izvozom iz avtoceste in je namenjen za obveščanje voznikov, da se približujejo določenemu izvozu iz avtoceste. V prvi vrstici prikazuje *Izvoz Celje jug*, v drugi pa *1000 m*. Na zaslonu bi se lahko prikazovali tudi drugi podatki, na primer *Kolona vozil*, *Močan veter*, *Omejitev hitrosti* in podobno.

Na izvozu iz avtoceste so na vozišču montirani trije fotoupori, ki jih vozila, ki zapuščajo avtocesto, "prekrijejo" po zaporednem vrstnem redu. Vozilo, ki se na izvozu iz avtoceste vključuje na avtocesto, pa "prekrije" fotoupore v nasprotnem vrstnem redu. Na izvozu iz avtoceste je postavljen tudi svetlobni opozorilni znak, obrnjen proti vozniku, ki vozi v napačno smer. Če voznik uporabi izvoz iz avtoceste za vključitev nanjo, ga opozorilni znak obvesti, da je njegova smer vožnje napačna. Ko voznik "prekrije" fotoupore v nasprotnem vrstnem redu, se vklopi svetlobni opozorilni znak, na LCD prikazovalniku pa se prikaže ustrezen napis, v prvi vrstici *Pozor vozilo v*, v drugi pa *nasprotno smer!*



Slika 1: Prikaz napisa na LCD zaslonu, ko ni nevarnosti zaradi vozila v napačni smeri (vir: avtor naloge)



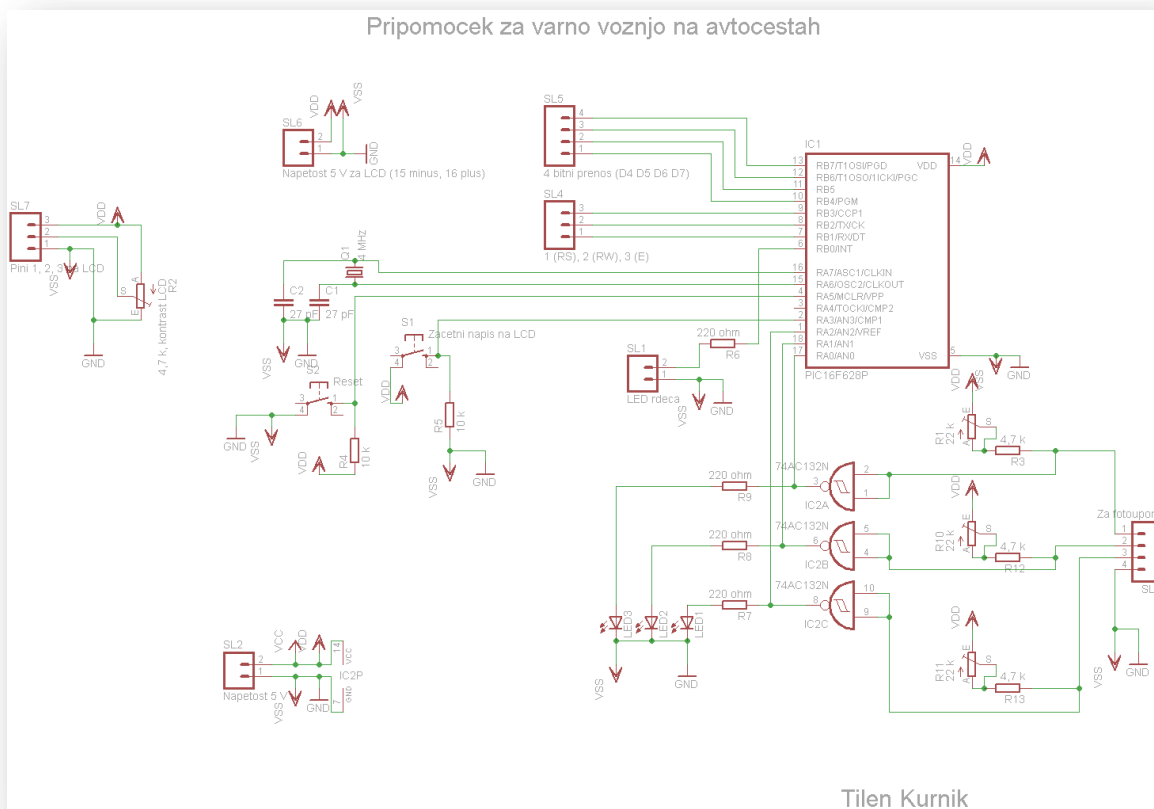
Slika 2: Napis na LCD zaslonu, ko nastane nevarnost zaradi vozila v napačno smer (vir: avtor naloge)

3.4. Tehnična in tehnološka dokumentacija

3.4.1 Elektronski načrt z mikrokontrolerjem

Elektronski načrt vezja, izdelan v programskem okolju Eagle, prikazuje Slika 3. Fotoupori so na vhode mikrokontrolerja priključeni preko NIN logičnih vrat s Schmitt-ovim prožilnikom. Zaradi različne zunanje osvetlitve, je lahko napetost na zatemnjenem ("prekritem") fotouporu različna. Fotouporu se pri zatemnitvi poveča upornost, zato se poveča tudi padeč napetosti na njem. Ta napetost ne bi bila vedno dovolj visoka za doseganje nivoja logične 1, ki ga potrebuje vhod mikrokontrolerja. Zato je vezju dodano integrirano vezje IC 74132, ki vsebuje NIN logična vrata s Schmitt-ovim prožilnikom. Na izhodu teh logičnih vrat dobimo v odvisnosti od velikosti vhodnega signala, vedno napetostni nivo logične 0 ali logične 1, potrebne za pravilno delovanje mikrokontrolerja.

Za določitev ustreznega vhodnega signala logičnih vrat pri različnih zunanjih osvetlitvi, je vsakemu fotouporu v vezju dodan trimer potenciometer. Z njim lahko nastavimo ustrezne napetostne nivoje, potrebne za prikaz delovanja na modelu, pri različni zunanji osvetlitvi. Del cestišča na modelu, kamor so montirani fotoupori, bi moral biti osvetljen, saj mora naprava delovati tudi ponoči.



Slika 3: Elektronski načrt vezja za krmiljenje naprave (vir: avtor naloge)

Za prikaz vrstnega reda "prekrivanja" fotouporov in s tem smer vožnje, so vezju dodane tri LED² diode, zaščitene z ustreznimi upori.

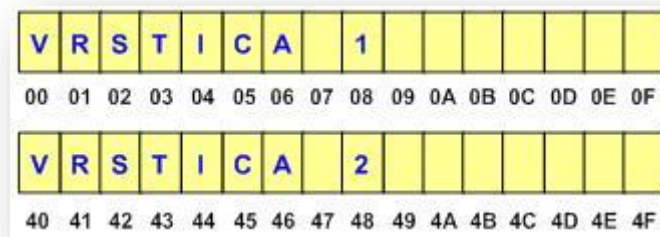
S tipko S1 preklopimo napajanje na LCD prikazovalniku v prvotno stanje, vezju pa je dodana tudi tipka za resetiranje mikrokontrolerja. Takt izvajanja programa mikrokontrolerja določa kristalni oscilator, vrednosti 4 MHz.

3.4.2 LCD prikazovalnik

Mikrokontroler skrbi za prikaz napisa na LCD prikazovalniku, katerega vsebina je odvisna od smeri vožnje. LCD prikazovalniki omogočajo izpisovanje črk, števil in ostalih znakov.

² LED (angl.: Light Emitting Diode), dioda ki oddaja svetlobo.

Uporabljamo jih kot vmesnike med uporabnikom in elektronsko napravo. V nalogi je uporabljen dvovrstični LCD prikazovalnik z lastno osvetlitvijo, ki ima v vsaki vrstici po 16 znakov (DEM 16216 SYH-PY). Vsebuje lasten kontroler (KS0070B-00), ki skrbi za vklapljanje ustreznih pik na zaslonu. Vsak znak je sestavljen iz 5 x 8 pik. Vsak znak na LCD prikazovalniku ima svoj naslov v DDRAM³ pomnilniku, kot prikazuje slika 4. Začetni naslov skrajnega levega znaka v prvi vrstici je 0x00, skrajnega levega znaka v drugi vrstici pa 0x40. Vsaka vrstica je navidezno dolga po 40 znakov, kar znaša skupaj 80 znakov. Od vseh 40 znakov v vsaki vrstici jih trenutno vidimo na zaslonu le 16. Tega lahko poljubno premikamo po celotnem prikazovalniku in tako določimo, kaj bo v določenem trenutku prikazano na zaslonu.



Slika 4: LCD 2 x 16, naslovi DDRAM - šestnajstiške vrednosti (vir: avtor naloge)

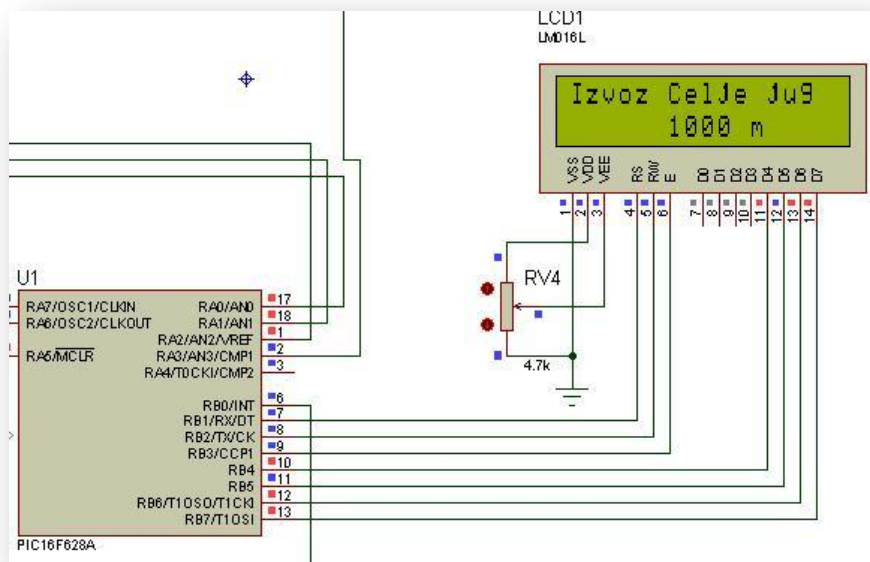
LCD prikazovalniki brez možnosti osvetlitve zaslona imajo 14 priključkov, taki z možnostjo osvetlitve zaslona pa imajo 2 priključka več, torej 16 priključkov. Funkcija posameznega priključka je standardna, vedeti moramo le, kje se nahaja prvi priključek. Če priključki za določen tip LCD-ja niso označeni, poiščemo podatke na spletu. Priključki od 7 do 14 LCD prikazovalnika so namenjeni podatkovnim linijam (od D0 do D7). Preko njih se prenašajo podatki od mikrokontrolerja do LCD-ja, če pišemo v LCD, in od LCD-ja do mikrokontrolerja, če beremo iz LCD-ja. LCD zaslone lahko delujejo v 8 ali v 4-bitnem načinu. V nalogi je izbran 4-bitni način delovanja, saj s tem privarčujemo pri priključkih mikrokontrolerja. Podatki se pri 4-bitnem načinu delovanja prenašajo po 4 bite hkrati, zato moramo poslati obe polovici bajta posebej, pošiljamo pa jih po podatkovnih linijah od D4 do D7. Tretji priključek

³ DDRAM (angl.: Display Data Random Access Memory), prikaz podatkov bralno-pisalnega pomnilnika.

LCD-ja služi nastavitvi kontrasta. Kontrast reguliramo z napetostjo na tem priključku. V ta namen uporabimo potenciometer kot delilnik napetosti.

LCD prikazovalnik krmilimo tako, da mu pošljemo ukaze ali podatke. Če postavimo priključek 4 (RS) v stanje logične 0, bo LCD sprejel podatek kot ukaz, če pa ga postavimo na nivo logične 1, bo LCD sprejel navaden podatek. Priključek 6 (E) je namenjen vklopu prikazovalnikove logike. Ko pošljemo neki podatek na LCD, ga moramo nekako obvestiti, da je na podatkovnih linijah nov podatek. To storimo tako, da ta priključek vklopimo (postavimo na 1) in izklopimo (postavimo na 0). Ob tem prehodu bo LCD sprejel nov podatek. Pri 4-bitnem načinu delovanja pošljemo najprej zgornje 4 bite podatka na priključke 11–14 (D4–D7), vmes vklopimo in izklopimo priključek 6 (E) in nato na iste priključke pošljemo še spodnje 4 bite podatka. Zatem zopet vklopimo in izklopimo priključek 6 (E). Vsak vpis ukaza ali podatka vzame LCD-ju nekaj časa, da ga obdela. V tem času, podajajo ga proizvajalci LCD-jev, mu ne smemo pošiljati naslednjega ukaza ali podatka, zato uporabimo metodo zakasnitve, ki bo v našem primeru ca. 5 ms. Časa zakasnitve ni treba točno nastaviti, le manjši od predpisanega ne sme biti.

LCD prikazovalnik moramo pred uporabo inicializirati. To pomeni, da moramo izvesti določeno zaporedje operacij, da ga postavimo v želeni način delovanja.



Slika 5: Priključitev LCD zaslona na mikrokontroler (vir: avtor naloge)

3.4.3 Mikrokontroler PIC16F628a

Osrednji del vezja je mikrokontroler PIC16F628a [vir: Programiranje PIC]:

- Je Microchipov mikrokontroler v DIL ohišju z 18 priključki.
- Od vseh priključkov je 16 vhodno/izhodnih, ki jih lahko programsko krmilimo.
- Napetost napajanja V_{dd} je 5 V.
- Vsebuje 2 k flash programskega pomnilnika, 224 bajtov RAM pomnilnika in 128 bajtov podatkovnega EEPROM pomnilnika.
- Centralno procesna enota je 8 bitna.
- Vsebuje tri timerje in pozna več vrst prekinitev.
- Pozna 35 instrukcij za programiranje v zbirnem jeziku.
- CPU izvaja instrukcije, vgrajen ima 8-bitni delovni register. Vanj se shranjujejo podatki po izvedbi posameznih instrukcij.
- Programski pomnilnik je sestavljen iz 2048 lokacij. V vsako lokacijo lahko vpišemo 14-bitni podatek. Instrukcija zavzame eno lokacijo. Posebno vlogo imata prva in peta lokacija. V prvi se nahaja ukaz, ki se bo izvedel takoj po vklopu mikrokontrolerja. V peti se začne del programa, ki se izvede ob prekinitvi.
- Maksimalna frekvenca zunanjega oscilatorja je 20 MHz.

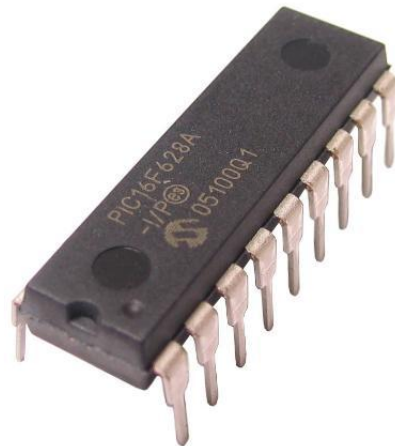
Podatkovni RAM⁴ pomnilnik je razdeljen na štiri banke: banko 0, banko 1, banko2 in banko3. Vsebuje SFR⁵ in GPR⁶ registre. SFR registri zavzemajo prvih 32 lokacij v vsaki banki. Preko njih komuniciramo z ostalimi enotami mikrokontrolerja. GPR registri zavzemajo 224 bajtov RAM pomnilnika, od katerih se zadnjih 16 lokacij preliva v vse štiri banke. V banki 0 in banki 1 zavzemajo GPR registri še po 80 bajtov, v banki 2 pa še 48 bajtov RAM podatkovnega pomnilnika. Če želimo brati ali pisati v register podatkovnega pomnilnika RAM, se moramo nahajati v banki, v kateri je zeleni register. Vsebine GPR registrov, ki so določeni s strani proizvajalca, se izgubijo, ko mikrokontrolerju izklopimo napajanje.

⁴ RAM (angl.: Random Access Memory), bralno-pisalni pomnilnik.

⁵ SFR (angl.: Special Function Registers), posebni funkcijski registri.

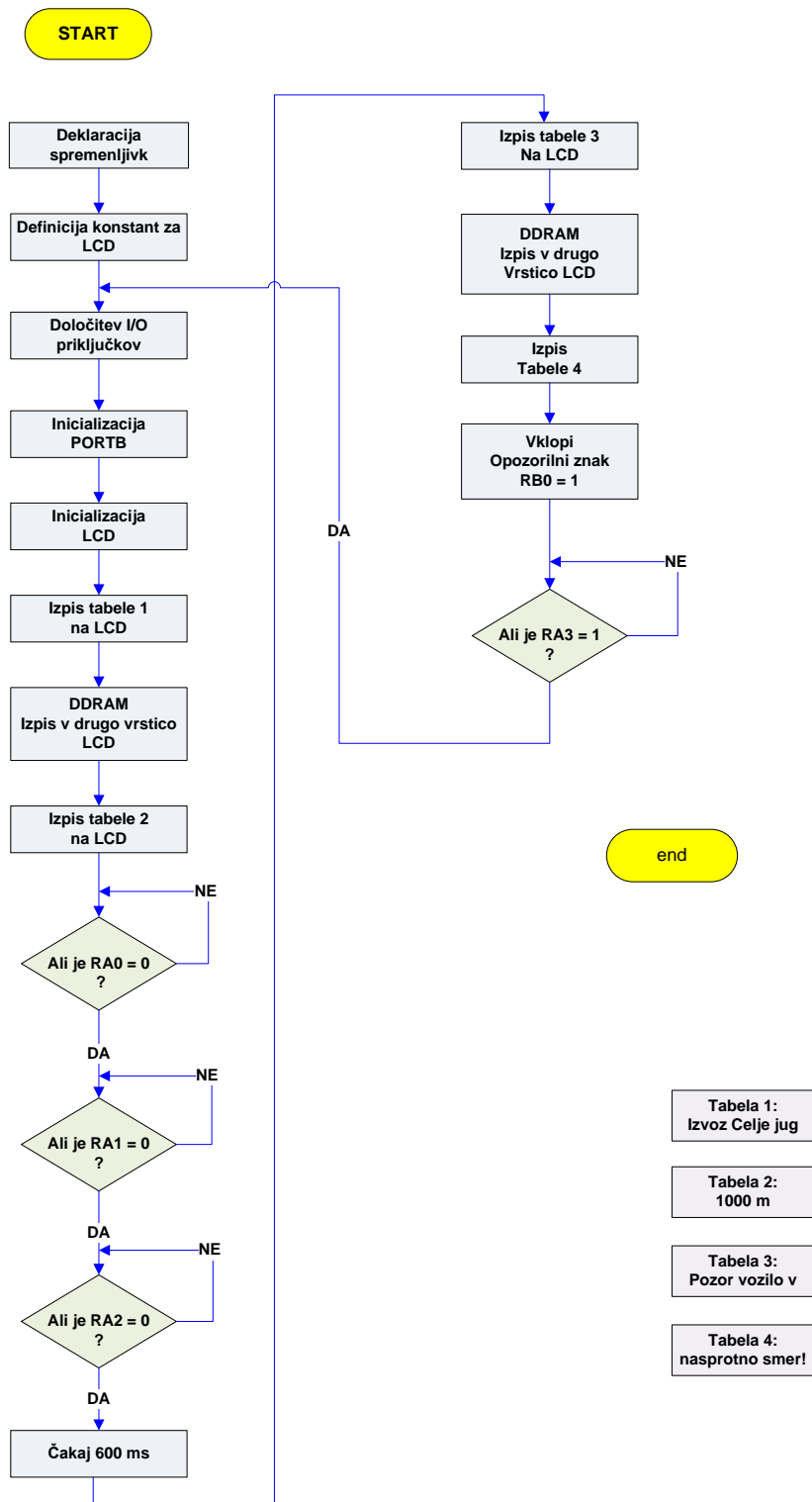
⁶ GPR (angl.: General Purpose Registers), registri namenjeni splošni uporabi.

Za izvajanje vseh instrukcij v mikrokontrolerju PIC skrbi 8-bitna aritmetična logična enota ALU (Arithmetic logic unit). Lahko seštevata in odšteva 8-bitna števila ter z njimi izvaja nekatere logične operacije (AND, OR, XOR, NOT ...). Po izvršitvi vsake instrukcije se v registru STATUS postavijo trije biti na ustrezne vrednosti. Iz njihovih vrednosti lahko ugotovimo, kakšen je dobljeni rezultat. Ta je lahko pozitiven, negativen, enak 0 ali napačen. Te tri bite imenujemo zastavice.



Slika 6: Mikrokontroler PIC16F628a [1]

3.4.4 Diagram poteka



3.4.5 Program za krmiljenje naprave

Program je napisan v zbirnem jeziku v okolju MPLAB IDE v8.80. V RAM podatkovnem pomnilniku so od naslova 0x20 naprej definirane konstante in spremenljivke, potrebne za posamezna zakasnitve, za LCD zaslon in za napise, ki so zapisani v tabelah.

S konfiguracijskimi biti je izbran zunanji oscilator, stabilizacija, izključitev časovnika stražnega mehanizma in izključitev zaščite pred kopiranjem. Konfiguracijske bite nastavimo z določitvijo vrednosti posameznih bitov v registru CONFIG. Do tega registra, njegov specialni naslov v podatkovnem pomnilniku je 2007h, lahko dostopamo le med programiranjem. Konfiguracijske bite določimo na začetku programa, takoj za direktivo `#include`, ta vključi v program datoteko `p16f628a.inc`, ki vsebuje imena registrov v mikrokontrolerju:

```
list      p=16f628a          ; Tip mikrokontrolerja
#include  <p16f628a.inc>     ; Vključi v program datoteko p16f628a.inc,
                           ; ki vsebuje imena registrov v mikrokontrolerju.

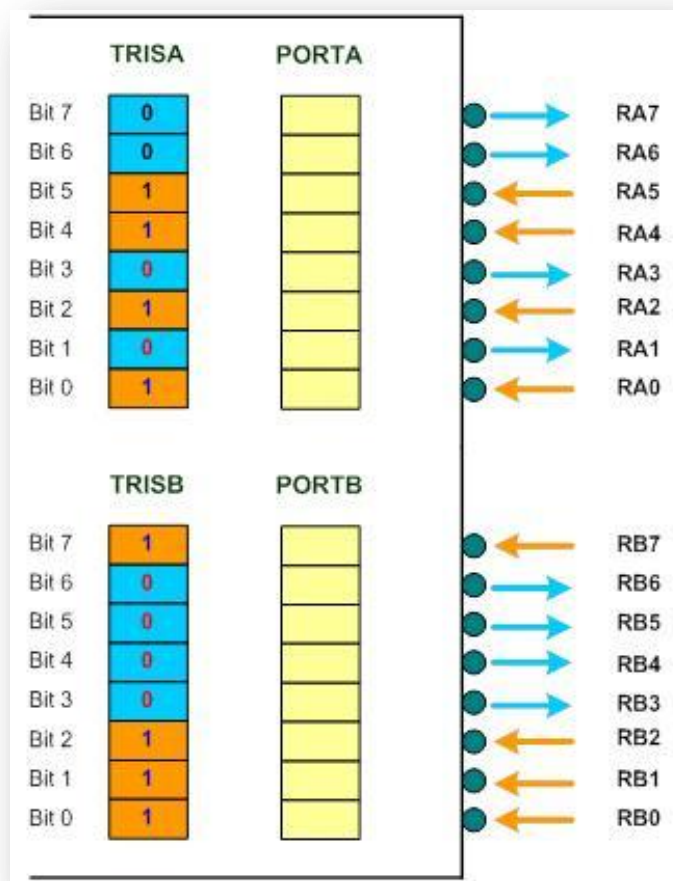
__CONFIG _CP_OFF & _WDT_OFF & _PWRTE_ON & _MCLRE_ON & _LVP_OFF & _XT_OSC
```

Mikrokontrolerju določimo nekatere pine kot vhodne in nekatere kot izhodne zaradi vezja, s katerim deluje. Glavna naloga mikrokontrolerja je namreč ta, da na posamezne pine pošilja napetost +5 V (1) ali napetost 0 V (0) in preverja, ali je na določenih pinih napetost +5 V (1) ali je ni (0). V odvisnosti od vezja, s katerim bo deloval, določimo ustrezne pine kot vhodne oziroma kot izhodne. Za krmiljenje vezja po elektronskem načrtu so pini RA0 do RA3 določeni kot vhodi, vsi ostali pa kot izhodi. Na vhodne pine so priključeni fotoupori in tipka za ponastavitev napisa na LCD zaslonu, na izhodne pa LCD zaslon in svetleči opozorilni znak.

```
bsf      STATUS,RP0        ;Banka 1
movlw   b'00001111'
movwf   TRISA              ;Pini RA0, RA1, RA2 in RA4 so vhodni, drugi so izhodni.
clrf    TRISB              ;Vsi pini na PORTB so izhodni.
bcf     STATUS,RP0        ;Banka 0
movlw   .7
movwf   CMCON              ;Pini na PORTA so digitalni (prvi trije biti registra CMCON so na 1)
```

Smer vsakega pina (ali bo deloval kot vhodni ali kot izhodni pin) določimo z vrednostjo bitov v registru TRISA za PORTA in v registru TRISB za PORTB. Če ima določen pin v registru

TRIS vrednost 0, bo deloval kot izhodni pin, če pa ima vrednost 1, bo deloval kot vhodni pin. Pine RA0, RA1, RA2 in RA3 lahko uporabimo kot vhode analognih komparatorjev, pina RA3 in RA4 sta v tem primeru izhoda komparatorjev. Lahko pa pine RA0–RA3 uporabimo kot digitalne vhode oziroma izhode. Izbiro nastavimo z določitvijo vrednosti bitov registra CMCON (Comparator control register).



Slika 7: Določanje vhodno-izhodnih pinov mikrokontrolerja (vir: avtor naloge)

Vrstni red pokrivanja fotouporov in s tem smer vožnje ugotavljamo s testiranjem pinov, kamor so fotoupori preko logičnih vrat povezani:

```

;***** Program za ugotavljanje stanja tipk *****
;***** Preberemo zaporedno stanje tipk: RA0, RA1 in RA2 *****
Beri
    goto    Beri
Preveri_RA0
    btfsc  PORTA,0
    goto   Preveri_RA0
Preveri_RA1
    btfsc  PORTA,1
    goto   Preveri_RA1
Preveri_RA2
    btfsc  PORTA,2
    goto   Preveri_RA2

```

V kolikor se ta del programa izvede, to pomeni, da je smer vožnje napačna. Na LCD zaslonu se prikaže ustrezen napis, vklopi pa se tudi opozorilni svetlobni znak.

```

;----- Naši nizi, Napisani so v tabelah -----
Tabela1 addwf PCL,f
        DT      "Izvoz Celje jug"      ;Tabela1 z nizom
Tabela2 addwf PCL,f
        DT      "1000 m"              ;Tabela1 z nizom
Tabela3 addwf PCL,f
        DT      "Pozor vozilo v"      ;Tabela1 z nizom
Tabela4 addwf PCL,f
        DT      "nasprotno smer!"     ;Tabela1 z nizom

```

Programski pomnilnik mikrokontrolerja pogosto uporabljamo za shranjevanje podatkov, napisanih v obliki tabele. Ta ima obliko podprograma. Preden tabelo pokličemo z instrukcijo call, moramo v delovni register vpisati zaporedno številko (odmik) podatka, ki ga želimo dobiti iz tabele. Prvi element tabele ima odmik 0. Ko se tabela izvrši, dobimo v delovnem registru vrednost izbranega podatka v tabeli. Tak način pridobivanja podatkov iz tabele nam omogoči direktiva DT (**d**efine **t**able – definiraj tabelo). Ta direktiva nadomesti vsak 8-bitni podatek za njo od leve proti desni. Podatek je lahko številka, znak ali beseda. Podatke ločimo z vejico. Paziti moramo, da pri klicu ne prekoračimo konca tabele. Tako lahko v tabelo zapišemo največ 256 8-bitnih podatkov. Podprogram za izpis niza iz tabele 1, ki vsebuje besede "Izvoz Celje jug" je sledeč:

```

;***** Podprogram za izpis niza iz tabele

LCD_Niz1
    clrf    Naslov           ;Naslov = 0
    bsf     RS              ;Podatkovni način
    movlw   .15
    movwf   Stevec2         ;Niz ima 15 znakov (Izvoz Celje jug)

Delaj1
    movf    Naslov,w
    movwf   Naslov
    call    Tabela1
    call    LCD_pisi        ;Podatek -> Lcd
    incf    Naslov,f        ;Naslednji podatek
    decfsz  Stevec2,f      ;Ali smo končali?
    goto    Delaj1         ;Ne še
    bcf     RS
    return                   ;Da, izhod iz podprograma

```

Ker bo izpis niza iz tabele 2, ki vsebuje niz "1000 m" izpisan v drugi vrstici LCD zaslona od sedmega znaka naprej, moramo to določiti v programu:

```

movlw    0x46
call     DDRAM_n         ;Pišemo v drugo vrstico LCD-ja (od DDRAM naslova 0x46 naprej)
bsf      RS              ;Podatkovni način, pisanje znakov
call     LCD_Niz2        ;V drugo vrstico izpišemo vsebino tabele 2
clrf     Stevec2

```

Za potrebne časovne zakasnitve v programu so izdelani podprogrami. Za zakasnitev 119 μ s, ki je potrebna takrat, ko LCD-ju pošljemo novi podatek, poskrbi podprogram:

```

;***** Naredi zakasnitev 119 mikrosekund

Cakaj119u
    movlw   .38             ;Število ponovitev zanke -> W
    movwf   Temp2          ;Vrednost delovnega registra v Temp2
Se    decfsz Temp2,f      ;Zmanjšaj Temp2 za 1, preskoči naslednjo instrukcijo, če je Temp2=0
    goto    Se             ;Temp2 še ni enak nič
    return                   ;Vrnitev iz podprograma

```

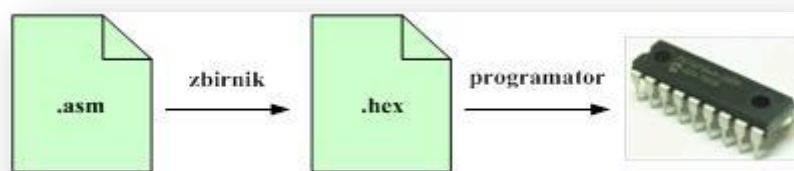
Podprogrami se vedno nahajajo na koncu glavnega programa. Če bi podprogram zapisali med glavnim programom, bi ga mikrokontroler izvedel tudi takrat, kadar ga ne bi poklicali, saj bi bil zanj le del glavnega programa. Da je program čitljivejši in preglednejši, podprograme ustrezno označimo s komentarji. Instrukcija **decfsz** zmanjšuje vrednost registra, zapisanega v prvem parametru, za 1 in nato preveri, ali je vrednost registra enaka 0. Če je enaka 0, preskoči naslednjo instrukcijo, v nasprotnem primeru nadaljuje izvajanje programa. Drugi parameter je

f, kar pomeni, da se bo rezultat operacije shranil nazaj v register, napisan v prvem parametru. V podprogramu Cakaj119u določimo vrednost spremenljivki Temp2. Ker smo omejeni z registri enega bajta, lahko z eno zanko napravimo največ 255 ponovitev. Zmanjšujemo Temp2 za 1. Ko je enak 0, to se zgodi po 38-tih ponovitvah zanke, se izvajanje programa z instrukcijo **return** vrne iz podprograma natanko tja, kjer je v glavnem programu skočil v podprogram. Ena ponovitev porabi 3 urine cikle, instrukcija **decfsz** enega, če ni izpolnjen pogoj, instrukcija **goto** pa dva.

	Stopwatch	Total Simulated
Instruction Cycles	119	20655
Time (uSecs)	119.000000	20655.000000
Processor Frequency (MHz)	4.000000	

Slika 8: Meritev časa zakasnitve v okolju MPLAB IDE (vir: avtor naloge)

Ko je program napisan, ga prevedemo. V mapi, kjer je projekt shranjen, se ustvari datoteka s končnico .hex. S programatorjem jo prenesemo v programski pomnilnik mikrokontrolerja. Pri tem smo uporabili programator PICKit 2, ki ga dobimo v prodajalnah za dokaj nizko ceno. Vsebuje tudi logični analizator, na računalnik pa ga lahko priključimo preko USB-vhoda. Programator ne potrebuje zunanjega napajanja.



Slika 9: Prevajanje programa in zapis v mikrokontroler (vir: avtor naloge)

4. INVESTICIJA PROJEKTA

Investicija projekta bi obsegala postavitev induktivnih zank pod vozišča na izhodih iz avtoceste, postavitev dodatnih opozorilnih znakov na teh priključkih ter vzpostavitev ustrezne komunikacije sistema.

Stroškov projekta nisem uspel izračunati oziroma določiti na podlagi podobnih projektov, sem pa mnenja, da bi uvedba takega sistema prihranila marsikatero življenje, kar je neprecenljivo.

5. DRUŽBENA ODGOVORNOST

Naloga prispeva k družbeno odgovornemu ravnanju do drugih, saj je osnovni namen varovanje človekovega življenja. Razlogi za vožnjo v napačno smer so lahko različni, večkrat so vozniki, ki peljejo v napačno smer starejši ljudje za volanom. Ti se preprosto ne znajdejo, so zmedeni, imajo težave z ožjim vidnim poljem in kratkotrajnim spominom. Zakaj ne bi takšnim pomagali in jih z dodatnim svetlobnim opozorilom seznanili, da vozijo v napačno smer, hkrati pa voznike na avtocesti opozarjali na nevarnost in na skrajno previdno vožnjo. Služba, ki se ukvarja z opozarjanjem in obveščanjem v prometu lahko vpliva na varnost in s tem odgovorno vpliva na varnost v družbi.

6. SKLEP

Z uspešno zaključeno izdelavo moje inovacijske naloge sem zelo zadovoljen in ponosen. Če pogledam na začetek raziskovanja, ko sem imel samo zamisel, sem videl pred sabo veliko dela. Že na začetku sem veliko premišljeval kako naj se naloge sploh lotim. Veliko časa sem zapravil z raziskovanjem delovanja in programiranja mikrokontrolerja.

S pomočjo mentorja in radovednosti in zagnanosti, sem svojo inovacijo vse bolj izpopolnjeval. Sedaj vidim, da končana izdelana maketa ni takšna, kot sem si jo na začetku zamislil. Maketo sem izboljšal, ter dopolnil z boljšimi idejami.

Zelo bi bil zadovoljen, če bi se vsaj kanček naloge približal realizaciji projekta v realnosti. Menim, da ta inovacijski predlog ne predstavlja cenovno velikih stroškov, bi pa ogromno prispeval k varnosti v prometu. Opozarjanje bi pripomoglo k zmanjšanju prometnih nesreč, ki se zgodijo na avtocestah.

Nalogo nameravam predstaviti tudi pristojnim službam, ki so odgovorni za cestni promet. Prisluhnil bom njihovim pripombam in predlogom. Nato pa bom svojo nalogo še bolj izpopolnil in izboljšal.

7. VIRI

1. Policija [spletni vir]. Dostopno na URL:

<http://www.policija.si/index.php/component/content/article/156-prometna-varnost/7360-vonja-v-nasprotni-smeri> (21. 01. 2013)

2. Programiranje PIC [spletni vir]. Dostopno na URL:

<http://sites.google.com/site/programiranjepic/home> (26. 01. 2013)

3. [1] Dostopno na URL: <http://mac.br.com/sl85/produto/Microcontrolador-PIC16F628A.html> (26. 01. 2013)