Mikroupravljački sustav za mjerenje i prikaz parametara vozila

Adrić, Damir

Master's thesis / Diplomski rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:000630

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

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

Sveučilišni studij

MIKROUPRAVLJAČKI SUSTAV ZA MJERENJE I PRIKAZ PARAMETARA VOZILA

Diplomski rad

Damir Adrić

Osijek, 2017.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac D1: Obrazac za imenovanje Povjerenstva za obranu diplomskog rada

Osijek, 21.09.2017.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za obranu diplomskog rada

Ime i prezime studenta:	Damir Adrić
Studij, smjer:	Diplomski sveučilišni studij Računarstvo, smjer Procesno računarstvo
Mat. br. studenta, godina upisa:	D 689 R, 14.10.2014.
OIB studenta:	09476502791
Mentor:	Doc.dr.sc. Ivan Aleksi
Sumentor:	Doc.dr.sc. Zdravko Krpić
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Doc.dr.sc. Tomislav Matić
Član Povjerenstva:	Prof.dr.sc. Željko Hocenski
Naslov diplomskog rada:	Mikroupravljački sustav za mjerenje i prikaz parametara vozila
Znanstvena grana rada:	**Arhitektura računalnih sustava (zn. polje računarstvo)**
Zadatak diplomskog rada:	U ovom radu potrebno je primijeniti mikroupravljački sustav za prikupljanje i prikaz parametara automobila. Parametre je potrebno prikazati na brzinomjeru u digitalnom obliku primjenom LCD pokaznika te svjetlećih dioda.
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: 3 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:	21.09.2017.

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, 12.10.2017.

Ime i prezime studenta:	Damir Adrić
Studij:	Diplomski sveučilišni studij Računarstvo, smjer Procesno računarstvo
Mat. br. studenta, godina upisa:	D 689 R, 14.10.2014.
Ephorus podudaranje [%]:	2

Ovom izjavom izjavljujem da je rad pod nazivom**: Mikroupravljački sustav za mjerenje i prikaz parametara vozila**

izrađen pod vodstvom mentora Doc.dr.sc. Ivan Aleksi

i sumentora Doc.dr.sc. Zdravko Krpić

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:

Sadržaj

1. UVOD 1
1.1. Zadatak diplomskog rada 1
2. PRIMIJENJENE TEHNOLOGIJE I ALATI 2
2.1. Arduino razvojno okruženje 2
2.2. Arduino Mega 2560 razvojni sustav 3
2.3. Otpornik 4
2.4. Kondenzator 5
2.5. LCD pokaznik 5
2.6. LED modul ws2812b 6
2.7. Opto-izolator 7
2.8. Regulator napona 78S05 i 78S10 8
2.9. Dvostruko operacijsko pojačalo LM358 8
2.10. Tranzistor IRF 9540n 9
2.11. Tranzistor BC337-40 10
2.12. RTC DS1307 modul 11
3. REALIZACIJA SUSTAVA 12
3.1. Mjerenje signala 12
3.2. Elektronička shema sustava 17
3.3. Izrada makete sustava 22
3.4. Programsko rješenje sustava 24
4. ZAKLJUČAK 25
LITERATURA 26
SAŽETAK 27
ABSTRACT 28
ŽIVOTOPIS 29
PRILOG A. Kompletna shema sustava 30
PRILOG B. Slika sustava ugrađenog u automobil 31
PRILOG C. Program za Arduino Mega razvojni sustav 32

1. UVOD

Cilj ovog diplomskog rada je napraviti kontrolnu ploču (engl. *dashboard*) koju će biti moguće ugraditi u automobil VW Golf 3 1.9 TD te koristiti u vožnji. Izvorna kontrolna ploča sastoji se od signalnih lampica te analognih uređaja za prikaz brzine, količine goriva u spremniku, temperature motora i sata za prikaz vremena. Ideja je to sve prikazati u digitalnom obliku, pomoću LCD ekrana i LED modula.

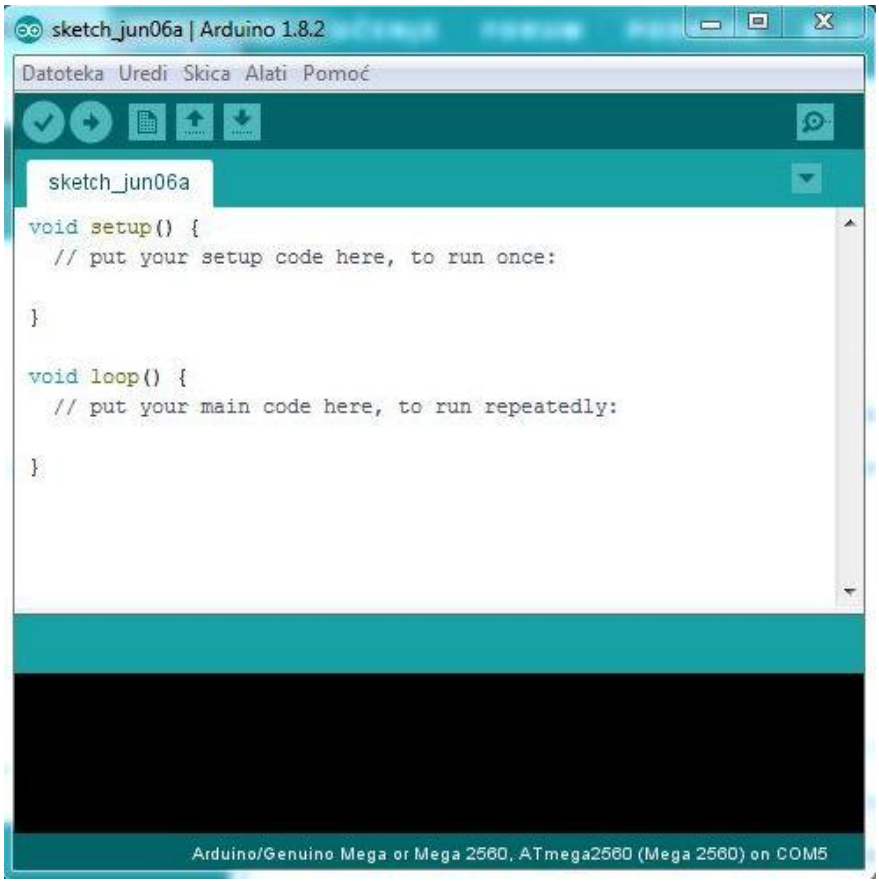
1.1. Zadatak diplomskog rada

U ovom radu potrebno je primijeniti mikroupravljački sustav za mjerenje i prikaz parametara automobila. Parametre je potrebno prikazati na brzinomjeru u digitalnom obliku primjenom LCD pokaznika te svjetlećih dioda.

2. PRIMIJENJENE TEHNOLOGIJE I ALATI

2.1. Arduino razvojno okruženje

Arduino razvojno okruženje sastoji se od nekoliko izbornika, gumba s funkcijama, prostora za pisanje programskog koda te prostora za poruke. Pomoću njega programski se kod prenosi na Arduino uređaj. Program napisan u Arduino razvojnom okruženju naziva se „sketch“, a datoteka programa ima nastavak „.ino“. Prilikom provjere ili spremanja programa, u prostoru za poruke prikazuju se različite informacije i greške [1]. Na slici 2.1 prikazan je izgled Arduino razvojnog okruženja.



Sl 2.1. Arduino razvojno okruženje

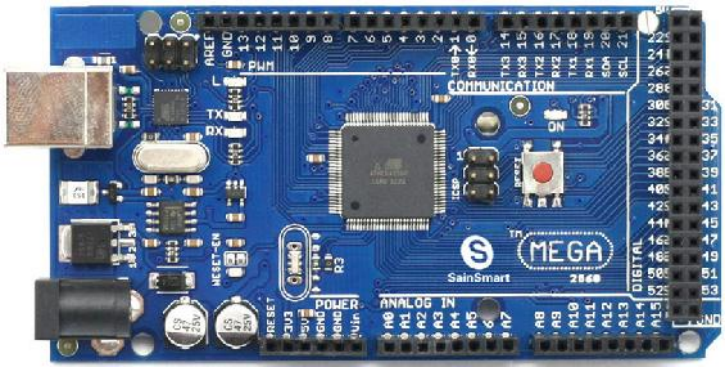
2.2. Arduino Mega 2560 razvojni sustav

Arduino Mega 2560 je mikroupravljačka ploča na kojoj se nalazi Atmega2560. Mega 2560 sadrži 54 ulazno/izlazna digitalna pina te 16 ulaznih analognih pinova. Frekvencija na kojoj radi je 16 MHz. Također sadrži 4 UART-a, USB konektor, konektor napajanja, tipkalo za ponovno pokretanje te ICSP konektor koji predstavlja jedan od načina spajanja pri programiranju Arduina. Napajanje može biti preko USB konektora ili preko konektora za vanjsko napajanje na koji se može spojiti baterija ili AC/DC ispravljač. Napon vanjskog napajanje može biti od 6 V do 20 V, ali zbog mogućeg nastanka problema preporučeni napon je od 7 V do 12 V. Sve specifikacije prikazane su u tablici 2.2.

Mikrokontroler	**Atmega2560**
Operativni napon	5 V
Preporučeni ulazni napon	7-12 V
Granični ulazni napon	6-20 V
Digitalni U/I pinovi	54 (od kojih 15 može davati PWM izlaz)
Analogni ulazni pinovi	16
DC struja po U/I pinu	20 mA
DC struja na 3.3V pinu	50 mA
Flash memorija	256 KB od kojih 8 KB zauzima bootloader
SRAM	8 KB
EEPROM	4 KB
Takt procesora	16 MHz

Tab 2.2. Specifikacije Arduino Mega2560

Arduino Mega 2560 opremljen je osiguračem koji štiti USB utor računala od kratkog spoja ili prevelike struje. Ako se preko USB utora isporuči više od 500mA, osigurač prekida spoj sve dok kratki spoj nije uklonjen. Dizajn Mega 2560 je takav da se na njega može spojiti većina pločica za proširenje, poznatijih kao „štitovi“ koji su dizajnirani za Arduino Uno, Diecimila ili Duemilanove. Na slici 2.2 prikazan je Arduino Mega 2560. Dimenzija Mega 2560 je 101.52mm*53.3mm [2].



Sl 2.2. Arduino Mega 2560 [3]

2.3. Otpornik

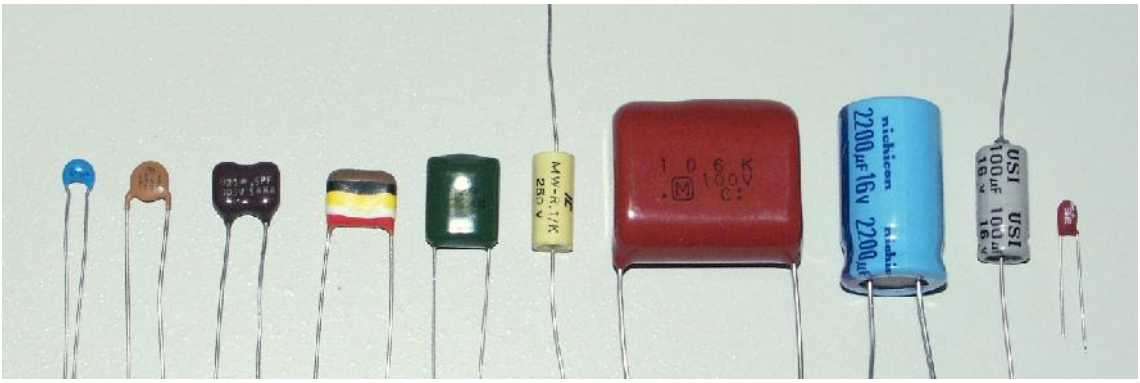
Otpornik je jedna od najprisutnijih komponenata u elektronici. To je pasivna elektronička komponenta koja se odupire prolasku električne struje. Otpornici se mogu pronaći u gotovo svim elektroničkim sklopovima. Mjerna jedinica za otpor je Ohm (Ω). Jedan Om je onaj otpor na kojem pri protjecanju struje od 1 A nastaje pad napona od 1 V. Otpornici se koriste za ograničavanje struje, kao djelilo napona, za stvaranje poznatog naponsko-strujnog odnosa, itd. Postoje dvije vrste otpornika, a to su fiksni otpornici i promjenjivi otpornici. Kod fiksnih otpornika otpor je točno definiran i ne može se mijenjati te je ova vrsta otpornika je najčešće korištena. Izgled fiksnog otpornika prikazan je na slici 2.3. Promjenjivi otpornici su oni otpornici kod kojih se otpor može podešavati na neku željenu vrijednost. Najviše korišteni promjenjivi otpornici su potenciometri. Postoje i otpornici koji mijenjaju otpor na osnovu neke fizikalne veličine, npr. foto-otpornici kod kojih se otpor mijenja promjenom intenziteta ulazne svjetlosti, termo-otpornici kod kojih se otpor mijenja promjenom temperature, itd. [4].



Sl 2.3. Fiksni otpornik [5]

2.4. Kondenzator

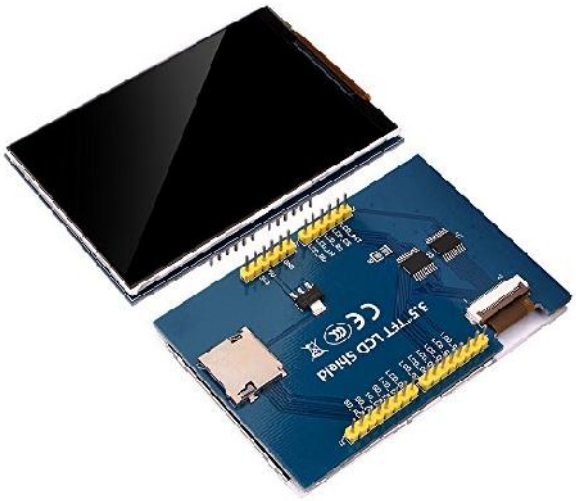
Kondenzator je pasivna elektronička komponenta u kojoj se pohranjuje električna energija. Električni kapacitet veličina je koja opisuje svaki kondenzator, a mjerna jedinica električnog kapaciteta je Farad (F). Kondenzatori se proizvode u različitim oblicima i veličinama, ovisno o tome gdje se primjenjuju. Poput otpornika, mogu se pronaći u gotovo svim elektroničkim sklopovima. Također kao i kod otpornika, postoje kondenzatori fiksne vrijednosti i kondenzatori kojima se može mijenjati vrijednost. Najčešći tipovi fiksnih kondenzatora jesu: elektrolitski, papirni, keramički i folijski. S obzirom na kapacitet, najrasprostranjeniji su oni manjega kapaciteta, reda veličine 1pF do nekoliko mF, jer je kapacitet od 1F vrlo velik. Različite vrste kondenzatora prikazane su na slici 2.4.



Sl 2.4. Različite vrste kondenzatora [6]

2.5. LCD pokaznik

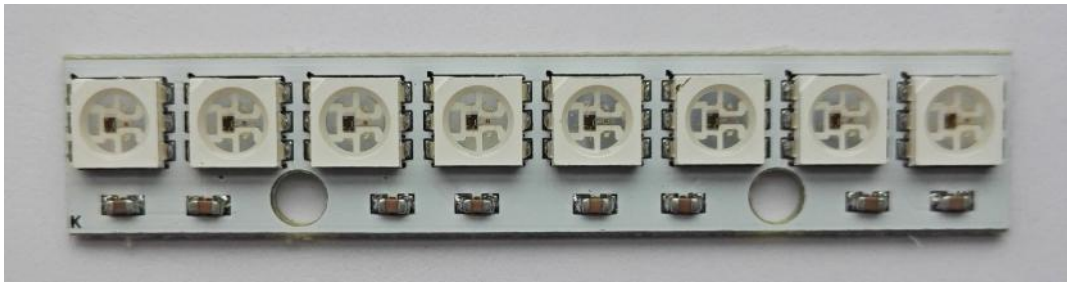
LCD (engl. *liquid-crystal display*) pokaznik je uređaj koji koristi tekuće kristale za prikaz slike. Tekući kristali ne emitiraju svjetlost, da bi prikaz bio vidljiv, koristi se pozadinsko osvjetljenje. LCD pokaznici imaju vrlo široku primjenu u računalnim monitorima, televizorima, različitim panelima, satovima, mobitelima, digitalnim fotoaparatima, itd. Ne troše puno električne energije i ne zauzimaju mnogo prostora. Svaki piksel na LCD pokazniku sastoji se od molekula između dvije prozirne elektrode i dva filtera polarizatora. Kada je riječ o LCD pokazniku u boji, svaki piksel podijeljen je na tri podpiksela, tj. crveni, zeleni i plavi, gdje se sa svakim upravlja posebno i tako se tvori jedna boja tog piksela [7]. Korišteni LCD pokaznik može se vidjeti na slici 2.5.



Sl 2.5. LCD pokaznik namijenjen za Arduino [8]

2.6. LED modul ws2812b

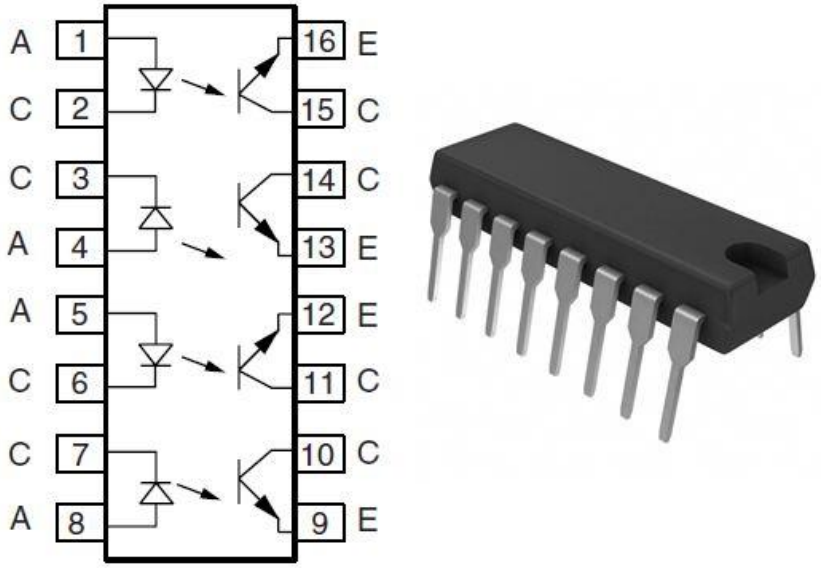
LED modul ws2812b sastoji se od 6 LED dioda koje je moguće programirati te mogu svijetliti neovisno jedna o drugoj. Diode su oblika 5050, što znači da je dimenzija svake 5.0mm*5.0mm. Svaka dioda u sebi ima ugrađen integrirani krug koji služi za njeno kontroliranje. Dioda može svijetliti u 3 boje i svaka boja može svijetliti u 256 različitih intenziteta, što dovodi do prikaza 16777216 različitih boja. LED diode su adresabilne pa se upravljanje vrši pomoću jednog pina. LED moduli ws2812b mogu se spajati jedan na drugi i tako stvoriti proizvoljan niz. Napon potreban za rad je od 3.5 V do 5.3 V. Za komunikaciju između mikroupravljača i ws2812b modula, koristi se *non-return-to-zero (NZR)* kod, gdje je stanje „1“ obično prikazano pozitivnim naponom, a stanje „0“ negativnim naponom. Diode su dugotrajne, visokog intenziteta osvjetljenja te niske potrošnje energije [9]. Korišteni LED modul ws2812b prikazan je na slici 2.6.



Sl 2.6. LED modul ws2812b

2.7. Opto-izolator

Opto-izolator je elektronička komponenta koja pomoću svjetlosti prenosi električni signal između dva izolirana strujna kruga. Strujni krugovi su galvanski odvojeni pa kvar ili kratki spoj na jednom strujnom krugu, ne može uzrokovati problem na drugom strujnom krugu. Za prijenos električnog signala najčešće se koristi LED dioda s jedne strane te foto-tranzistor s druge strane, a između njih je izolacijska barijera. LED dioda električni ulazni signal pretvara u svjetlost, a foto-tranzistor detektira svjetlost iz koje direktno proizvodi električnu energiju ili ju propušta iz nekog vanjskog izvora. Osim što LED diode pretvaraju električnu energiju u svjetlost, mogu raditi i obrnuto tako da postoje opto-izolatori koji rade u oba smjera. Glavna funkcija opto-izolatora je da blokira visoke napone i naponske prijelaze koji se mogu pojaviti u jednom dijelu sustava, tako da ne oštete drugi dio sustava [10]. Na slici 2.7. prikazan je opto-izolator.

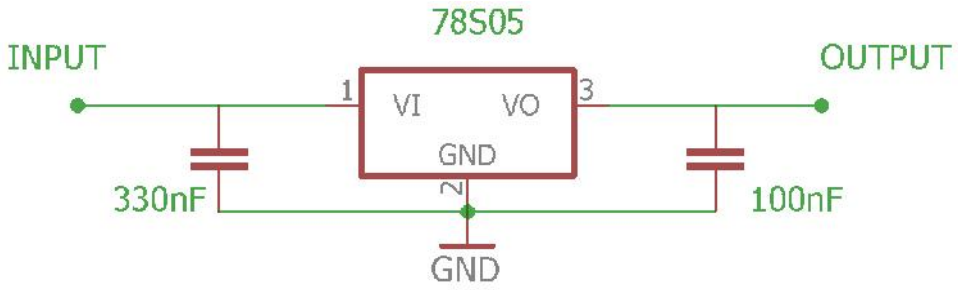


Sl 2.7. Opto-izolator[11]

2.8. Regulator napona 78S05 i 78S10

Automobil kao izvor električne energije koristi akumulator odnosno alternator koji daju 12 V – 14 V. LED modul ws2812b ne može raditi na naponu tih vrijednosti. Zbog toga je potrebno spustiti napon na 5 V. U tu svrhu korišten je regulator napona 78S05. Ulazni napon može biti od 7 V do maksimalno 255 V, a izlazni napon je fiksno 5 V. Struja koju može isporučiti je do 2 A.

Preporučeni napon za napajanje Arduina je 7 V – 12 V te se i u tom slučaju napon treba spustiti na preporučenu vrijednost za što je korišten regulator napona 78S10. Ulazni napon može biti od 12.5 V do maksimalno 25 V, a izlazni napon je fiksno 10 V. Struja koju može isporučiti je do 2 A. Oba regulatora se u strujni krug spajaju prema slici 2.8.



Sl 2.8. Shema spajanja naponskog regulatora

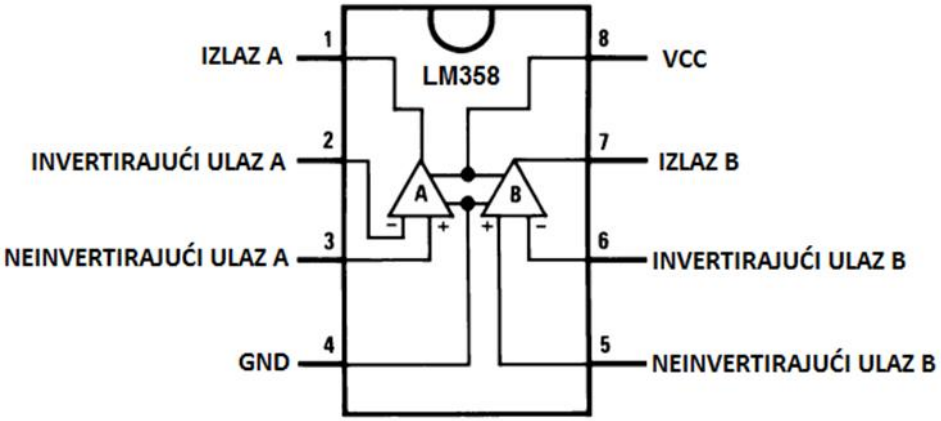
Kondenzatori prikazani na shemi služe za eliminiranje oscilacija odnosno šuma napona izvora.

2.9. Dvostruko operacijsko pojačalo LM358

Operacijsko pojačalo je elektronička komponenta namijenjena za pojačanje napona. Glavna svrha ove komponente je pojačavanje ulaznog napona. Također se može koristiti kao komparator, naponsko sljedilo, derivator, naponski i strujni regulator, itd. Idealno operacijsko pojačalo smatra se ono koje ima beskonačno velik ulazni otpor, dok je izlazni otpor zanemariv te je naponsko pojačanje beskonačno veliko. Prema slici 2.9. vidljivo je da LM358 sadrži dva neovisna operacijska pojačala, svako pojačalo ima dva pina za ulaz, jedan za izlaz te dva zajednička pina za napajanje [12].

Specifikacije:

• Veliko pojačanje DC napona: 100 dB
• Široki pojas frekvencija (bez pojačanja): 1 MHz
• Široki raspon napajanja:
    - unipolarno: od 3 V do 32 V
    - bipolarno: ± 1.5 V do ± 16 V
• Vrlo niska struja odvoda (500 μA)
    - neovisna o naponu napajanja
• Niska DC komponenta ulaznog napona: 2 mV
• Ulazni raspon napona uključuje GND
• Diferencijalni napon jednak je naponu napajanja

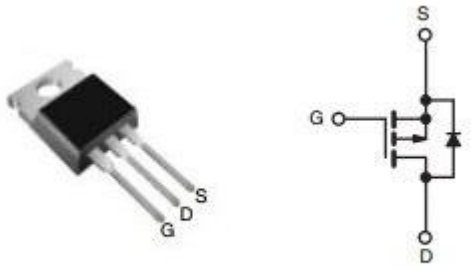


Sl 2.9. Shema LM358 [13]

2.10. Tranzistor IRF 9540n

IRF 9540n je unipolarni P-kanalni MOSFET tranzistor. FET tranzistori dijele se na više vrsta, a jedna od njih je MOSFET što je skraćeno od *Metal Oxide Semiconductor Field-Effect Transistor*. Postoje dva tipa unipolarnih tranzistora: P-kanalni i N-kanalni. Nazivaju se unipolarni zbog toga što vodljivost ovisi samo o većinskim nositeljima električnog naboja, a to su šupljine kod P-kanalnih ili elektroni kod N-kanalnih tranzistora. Tranzistor ima 3 elektrode odnosno nožice. Kod unipolarnog to su: S – uvod (engl. *Source*), D – odvod (engl. *Drain*) i G – upravljačka elektroda (engl. *Gate*). Raspored nožica i izgled spomenutog tranzistora prikazan je na slici 2.10. FET tranzistori imaju velik ulazni otpor pa troše vrlo malo struje te je opterećenje strujnog kruga minimalno. Rad FET tranzistora ovisi o naponu na upravljačkoj elektrodi (G). FET tranzistori se više koriste od BJT

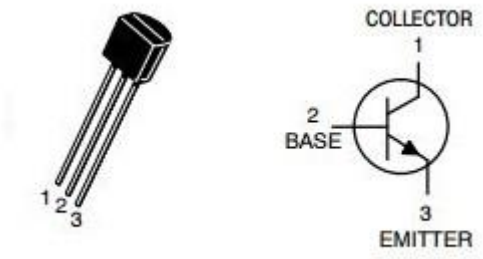
tranzistora, ali su skuplji za proizvesti. Za otkriće tranzistora zaslužan je William Shockley i to 1947. godine te je od tada počeo razvoj bipolarnih tranzistora, a prvi unipolarni tranzistori proizvedeni su 1960. godine. Od tada pa sve do danas u gotovo svim elektroničkim uređajima koriste se tranzistori. Najviše se koriste kao pojačala i sklopke [14].



Sl 2.10. Tranzistor IRF9540[14]

2.11. Tranzistor BC337-40

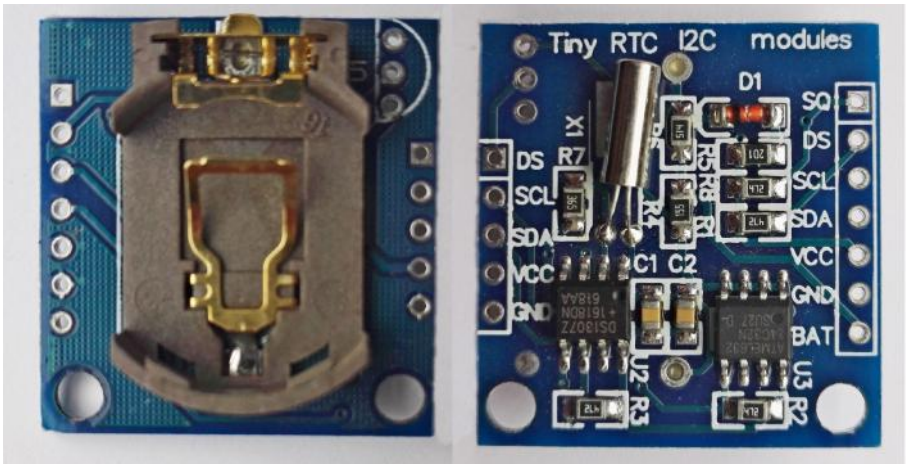
BC337-40 je bipolarni NPN BJT tranzistor. BJT je skraćeno od *Bipolar Junction Transistor*. Bipolarni tranzistori također se kao i unipolarni, dijele na dvije vrste: NPN i PNP. Kod bipolarnih tranzistora elektrode su: E – emiter (engl. *Emitter*) , B – baza (engl. *Base*) i C – kolektor (engl. *Collector*). Raspored nožica i izgled spomenutog tranzistora prikazan je na slici 2.11. Bipolarni su zbog toga što vodljivost ovisi o većinskim i o manjinskim nositeljima električnog naboja. BJT tranzistori imaju malen ulazni otpor pa „vuku“ više struje iz strujnog kruga zbog čega može doći do opterećenja strujnog kruga. Rad BJT ovisi o struji na bazi, za razliku od FET tranzistora koji su naponski upravljani [15].



Sl 2.11. Tranzistor BC3370-40 [15]

2.12. RTC DS1307 modul

RTC (engl. *Real-Time Clock*) je uređaj za praćenje vremena. S Arduinom komunicira serijski preko I2C dvosmjerne sabirnice. Pruža informacije o trenutnom vremenu, tj. koji je sat, minuta, sekunda, dan, mjesec i godina. Troši vrlo malo električne energije, a kada se vrijeme jednom postavi pamti ga uz pomoć baterije ugrađene u RTC modul. Prikaz vremena može biti u obliku 24h ili 12h. Modul ima ugrađen senzor za detekciju gubitka napajanja te se u tom slučaju automatski prebacuje na baterijsko napajanje. Za potrebe pamćenja vremena u stanju mirovanja, modul troši manje od 500 nA iz baterije. Preporučena temperatura okoline u kojoj se koristi navedeni modul je od -40°C do +85°C [16]. RTC DS1307 modul prikazan je na slici 2.12.



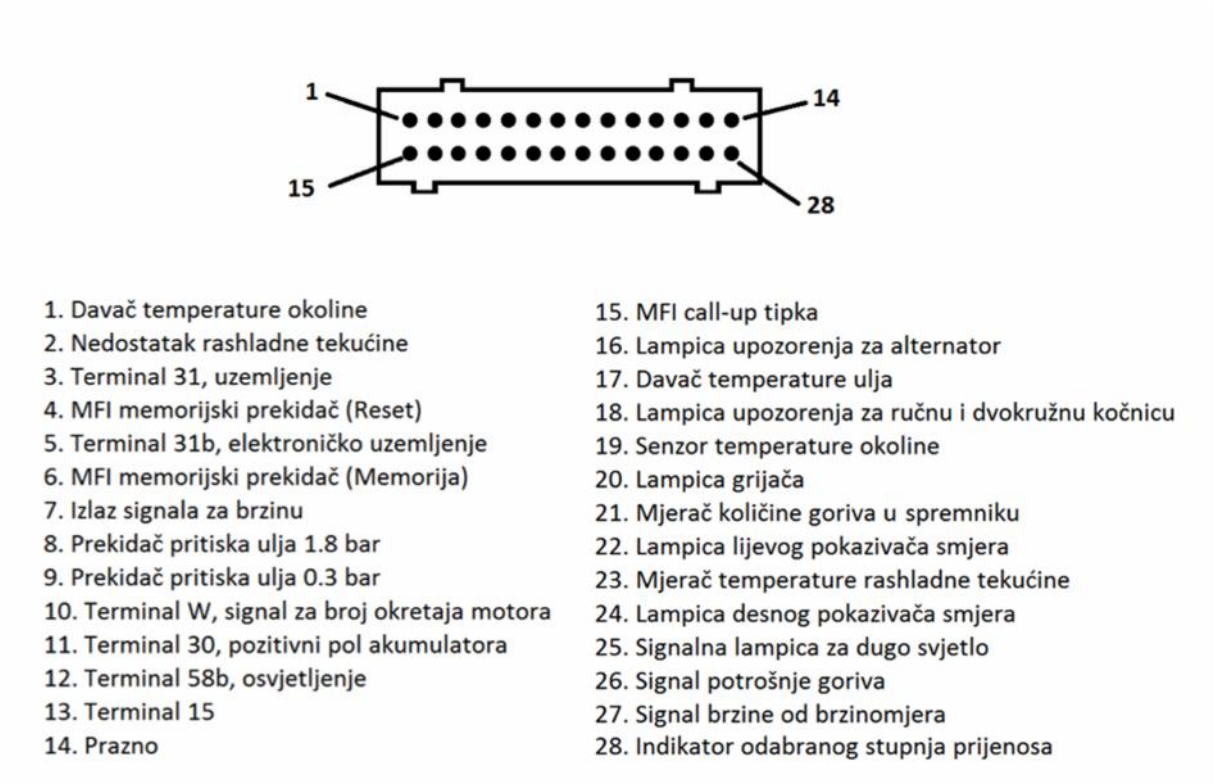
Sl 2.12. RTC DS1307

3. REALIZACIJA SUSTAVA

Kao što je već objašnjeno u uvodu, potrebno je izraditi novu automobilsku kontrolnu ploču, ljepšeg i suvremenijeg dizajna, koristeći novije tehnologije. Kontrolna ploča treba biti kompatibilna za ugradnju u VW Golf 3 1.9 TD. Prvo je potrebno izmjeriti signale sa konektora u automobilu te im odrediti vrstu i raspon. Nakon toga treba odabrati koji signali, odnosno pinovi, će biti korišteni za prikaz informacija na kontrolnoj ploči te na koji način će Arduino dohvaćati signale. U skladu s tim treba napraviti elektroničku shemu sustava na osnovu koje se izrađuje maketa sustava te se naposljetku izvodi programsko rješenje sustava.

3.1. Mjerenje signala

Da bi se izradila elektronička shema sustava prvo je trebalo doći do informacije o signalima na pinovima automobilskog konektora. Konektor koji se spaja na kontrolnu ploču je 28-pinski (sastoji se od dva reda po 14 pinova).



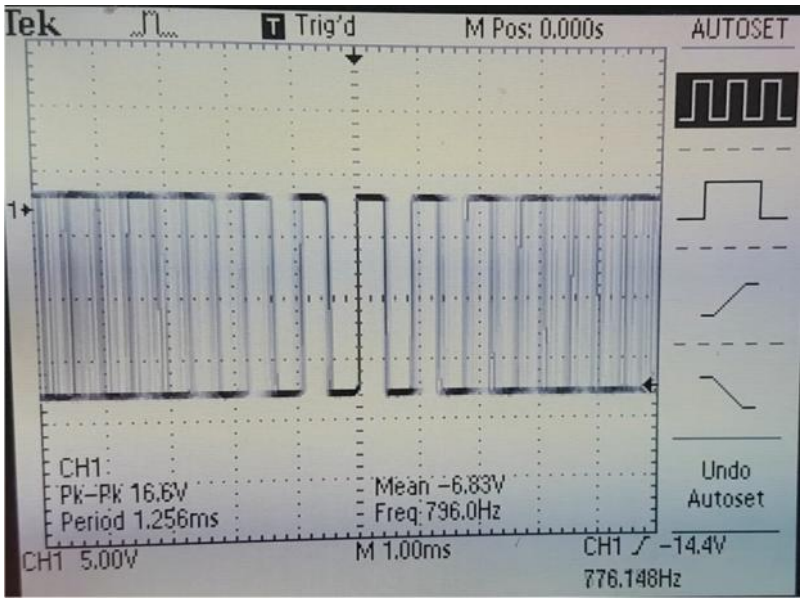
Sl 3.1. 28-pinski konektor na elektroničkoj instalaciji automobila

Na slici 3.1. prikazan je općeniti raspored pinova za VW Golf 3, ali broj korištenih pinova ovisi o opremi i motoru ugrađenima u automobil. Npr. automobili sa benzinskim motorom nemaju grijače pa tako nemaju ni pin koji služi za lampicu grijača. Grijači pomažu u pokretanju hladnog dizelskog motora, zagrijavaju komprimirani zrak u cilindrima automobila i tako pomažu pri samozapaljenju ubrizganog dizela. Ne postoji puno informacija o točnim vrstama signala koje pojedini pinovi isporučuju pa je bilo potrebno te signale nekako izmjeriti. U tu svrhu korišten je osciloskop i multimetar. Vrijednost pinova mjerena je tako da se ispitni vodovi multimetra ili ispitne sonde osciloskopa spoje na pin čija se vrijednost želi izmjeriti i na pin 5 koji je elektroničko uzemljenje ili na pin 11 koji je „+“ pol akumulatora. Prema pinu 5 se mjere signali sa pinova: 2, 10, 16, 21, 22, 23, 24, 25 dok se prema pinu 11 mjere: 9, 18, 20 i 27. Pinovi su redom mjereni i zapisivana su očitanja s pinova. Dobiveni oblici signala su u različitim oblicima, a to su napon, električni otpor i frekvencija. Oblici signala očitani na pinovima prikazani su u tablici 3.1.

Redni broj i naziv pina	**Informacija o pinu i oblik signala**
1. Davač temperature okoline	Prazan pin
2. Nedostatak rashladne tekućine	Beskonačan otpor prilikom nedostatka tekućine
3. Terminal 31, uzemljenje	Negativni pol akumulatora
4. MFI mem. prekidač	Prazan pin
5. Terminal 31b, ele. uzemljenje	Negativni pol akumulatora
6. MFI mem. prekidač	Prazan pin
7. Izlaz signala za brzinu	Prazan pin
8. Prekidač pritiska ulja 1.8 bar	Kada je tlak ispod 1.8 bar na pinu je GND
9. Prekidač pritiska ulja 0.3 bar	Kada je tlak ispod 0.3 bar na pinu je GND
10. Terminal W, broj okretaja motora	Frekvencija u ovisnosti o broju okretaja motora
11. Terminal 30, pozitivni pol akumulatora	Pozitivni pol akumulatora
12. Terminal 58b, osvjetljenje	12-14 V kada su upaljena svjetla na automobilu
13. Terminal 15	Prilikom davanja kontakta i dok je motor automobila pokrenut, na pinu je ~14 V
14. Prazno	Prazan pin
15. MFI *call-up* tipka	Prazan pin
16. Lampica upozorenja za alternator	Isporučuje ~14 V kada alternator puni akumulator
17. Davač temperature ulja	Prazan pin
18. Lampica dvokružne i ručne kočnice	Kada je dignuta ručna kočnica, na pinu je GND
19. Senzor temperature okoline	Prazan pin
20. Lampica grijača	Kada su grijači uključeni, na pinu je GND
21. Mjerač količine goriva u spremniku	Na pinu je električna otpornost vrijednosti od 0Ω za pun spremnik do 160 Ω za prazan
22. Lampica lijevog pokazivača smjera	Kada je pokazivač smjera uključen, na pinu je 12-14 V
23. Mjerač temp. rashladne tekućine	Na pinu je električna otpornost u rasponima na slici 3.4
24. Lampica desnog pokazivača smjera	Kada je pokazivač smjera uključen, na pinu je 12-14 V
25. Lampica dugog svjetla	Kada je dugo svjetlo uključeno, na pinu je 12-14 V
26. Signal potrošnje goriva	Prazan pin
27. Signal brzine od brzinomjera	Frekvencija u ovisnosti o brzini kretanja
28. Indikator odabranog stupnja prijenosa	Prazan pin

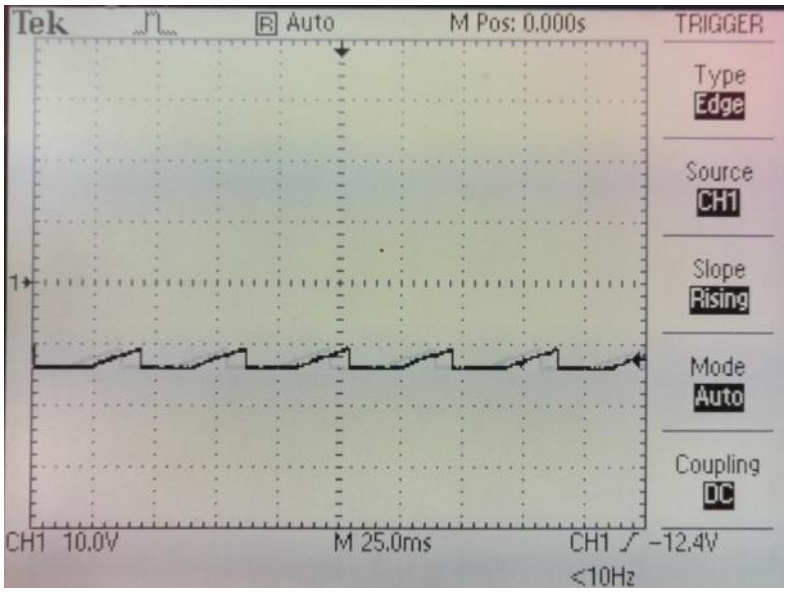
Tab 3.1. Očitane vrijednosti pinova

Iz tablice 3.1. vidljivo je da postoje različite vrste očitanih signala, a na slici 3.2 prikazano je kako izgleda signal broja okretaja motora. Signal je u obliku frekvencije te je paralelnim očitavanjem i uspoređivanjem vrijednosti s automobilskog panela i osciloskopa utvrđeno da frekvencija od 1 Hz odgovara 3 o/min. S obzirom da originalna kontrolna ploča na mjestu za prikaz broja okretaja ima sat za prikaz vremena, do prikaza broja okretaja se dolazi ulaskom u servisni izbornik koji se prikazuje na ekranu ispod brzinomjera.



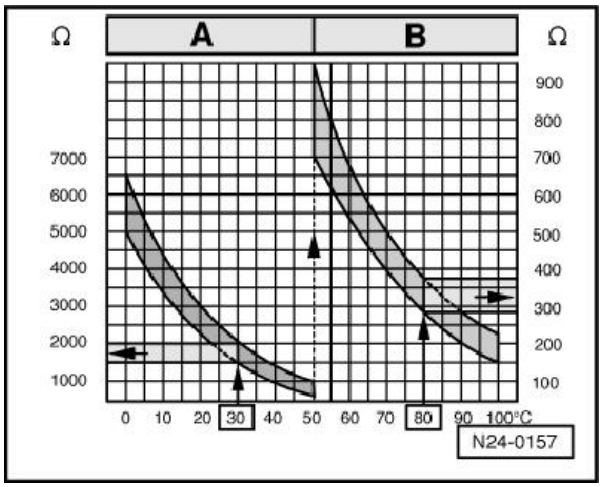
Sl 3.2. Očitavanje broja okretaja motora

Na slici 3.3. prikazano je očitavanje signala brzine automobila koji je također frekvencija, ali se razlikuje od frekvencije broja okretaja. Amplituda frekvencije brzine automobila kreće se od ~ -10 V do ~ -14 V. Mjerenjem je utvrđeno da frekvencija 1 Hz odgovara brzini od 1 km/h.



Sl 3.3. Očitavanje brzine automobila

Promjena otpora temperaturnog senzora u ovisnosti o promjeni temperature prikazana je na slici 3.4.



Sl 3.4. Vrijednosti otpora senzora za pojedinu temperaturu[17]

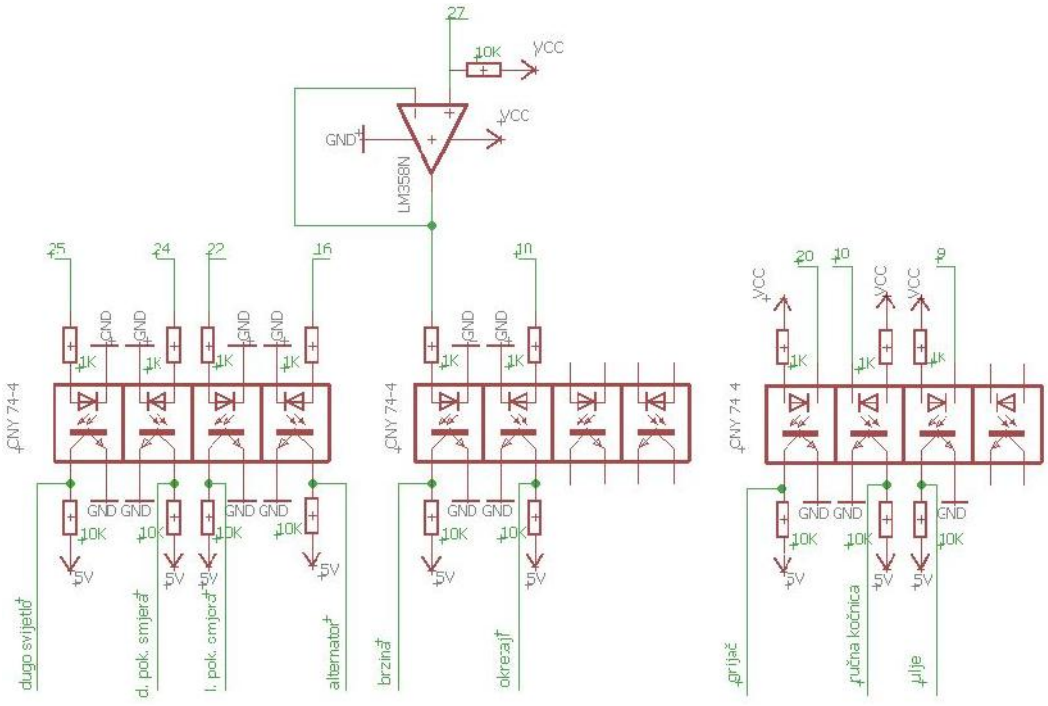
Na pinovima 4, 6, 14, 15, 17, 19, 26 i 29 nije bilo nikakvih očitanja, odnosno pinovi su bili prazni. Razlog nepostojanja većine od ovih pinova je to što korišteni automobil nema kontrolnu jedinicu motora ili ECU (engl. *Engine control unit*). ECU upravlja velikim brojem aktuatora na motoru s unutarnjim izgaranjem, u svrhu osiguravanja optimalnih voznih svojstava. ECU očitava vrijednosti mnogo senzora i s obzirom na očitane vrijednosti upravlja aktuatorima. On općenito kontrolira svu elektroniku u automobilu, od

detekcije neispravne žarulje u automobilu pa sve do određivanja smjese zraka i goriva. Prije nego što je ECU izumljen, aktuatorima se upravljalo mehanički. Svi automobili proizvedeni u današnje vrijeme posjeduju ECU, što prije 25 godina nije bio slučaj [18].

Za izradu makete sustava ne koriste se svi raspoloživi pinovi jer su nepotrebni, npr. pin 12 koji na novoj kontrolnoj ploči nema svrhu jer se ona sastoji od LED modula i LCD pokaznika koji svijetle i nije im potrebno dodatno osvjetljavanje. Dakle, pri izradi nove kontrolne ploče koristi se 15 pinova, a to su redom: 2, 5, 9, 10, 11, 13, 16, 18, 20, 21, 22, 23, 24, 25 i 27. Ostali pinovi se ne spajaju na Arduino.

3.2. Elektronička shema sustava

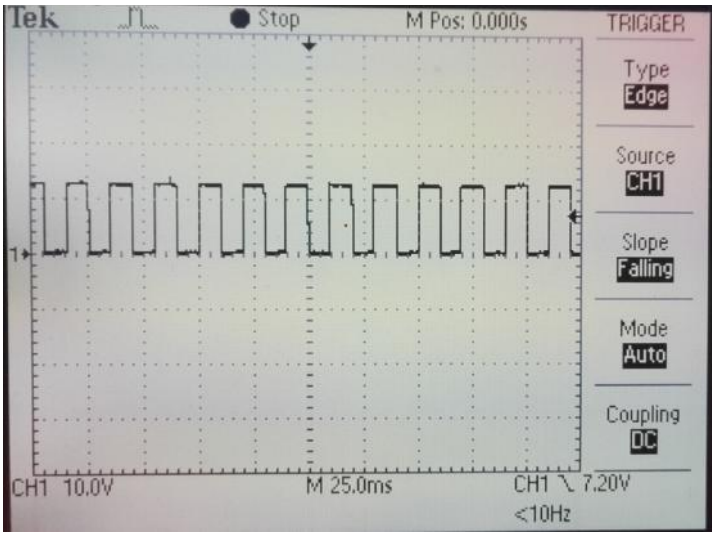
Pošto su neki pinovi davali vrijednosti napona od 12 V – 14 V, a na digitalne ulaze Arduina se smije dovesti najviše 5 V, bilo je potrebno spustiti napon signala. Spuštanje naponskih signala izvršeno je uz pomoć opto-izolatora. Opto-izolatori su spojeni prema slici 3.5.



Sl 3.5. Shema spajanja opto-izolatora u sustavu

Na stranu opto-izolatora gdje su smještene diode dovodi se signal sa konektora automobila. Napon je potrebno smanjiti da ne bi došlo do neispravnosti dioda, a u tu svrhu su

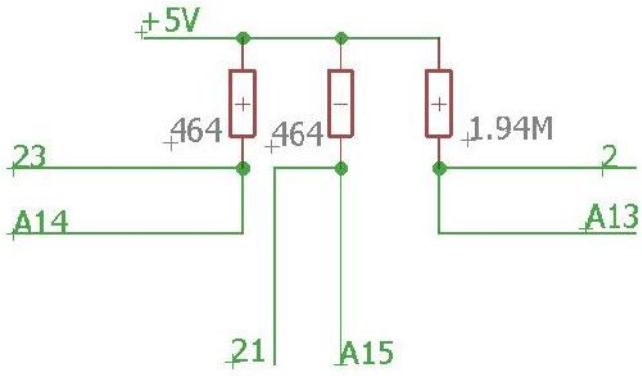
postavljeni otpornici vrijednosti 1K. Kod očitavanja digitalnih ulaza javio se problem očitavanja pina 27, odnosno signala za brzinu. Kao što je vidljivo na slici 3.3. napon se nikada ne spušta na 0 V, tako da je nemoguće očitati frekvenciju koristeći samo opto-izolator. Zbog toga je korišteno operacijsko pojačalo LM358 na kojega se kao ulaz dovodi pin 27, a na izlazu se dobije jednaka frekvencija, ali oblika koji odgovara za očitavanje preko opto-izolatora. Oblik frekvencije nakon prolaska kroz LM358 prikazan je na slici 3.6.



Sl 3.6. Signal brzine nakon prolaska kroz LM358

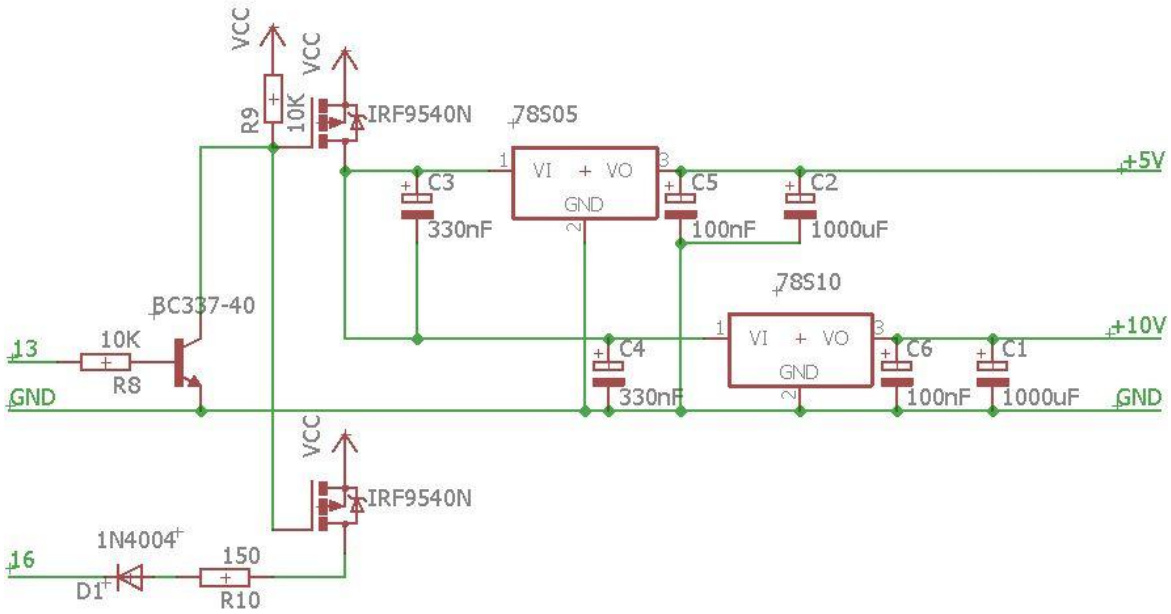
S druge strane opto-izolatora nalaze se nožice foto-tranzistora koji u ovisnosti o promjeni na diodi propušta ili ne propušta signal prema Arduinu. Tu su još spojeni i *pull-up* otpornici jer ulazni pinovi Arduina ne smiju biti „u zraku“.

Za mjerenje vrijednosti otpora s Arduinom potrebno je izraditi naponsko djelilo. Da bi se odredila vrijednost nepoznatog otpornika, u strujni krug je potrebno spojiti otpornik poznatog otpora. Za otpornike poznate vrijednosti korišteni su otpornici kojima je izmjerena otpornost od 464 Ω. Shema spajanja naponskog djelila u sustavu prikazana je na slici 3.7.



Sl 3.7. Shema spajanja naponskog djelila

Kao što je već spomenuto, napon koji daju akumulator odnosno alternator previsok je za napajanje sustava te ga je potrebno spustiti na odgovarajuće razine. Za napajanje LED modula napon je spušten na 5 V uz pomoć naponskog regulatora 78S05, a za napajanje Arduina napon je spušten na 10 V uz pomoć naponskog regulatora 78S10. Uključivanje sustava prilikom pokretanja motora automobila te njegovo isključivanje prilikom zaustavljanja rada motora kontrolira se uz pomoć pina 13. Prilikom davanja kontakta i dok je motor automobila pokrenut, pin 13 isporučuje ~14 V. Kao sklopke za uključivanje i isključivanje napajanja u sustavu koriste se tranzistori IRF 9540n i BC337-40. Također je potrebno trošilo između pinova 16 i 11 tako da bi alternator počeo puniti akumulator. U originalnoj kontrolnoj tabli trošilo je lampica koja svijetli kada se akumulator ne puni, jer tada je pin 16 GND a pin 11 je +12 V. Prilikom pokretanja motora automobila ta lampica povlači struju i alternator započinje s punjenjem akumulatora i lampica se gasi jer tada je na pinovima 16 i 11 napon od +14 V. U svrhu trošila korišten je otpornik 150 Ω, 2 W. Otpornik je spojen preko tranzistora, jednako kao i napajanje sustava tako da prilikom oduzimanja kontakta ne troši energiju akumulatora. Na slici 3.8. prikazana je shema spajanja napajanja i trošila.



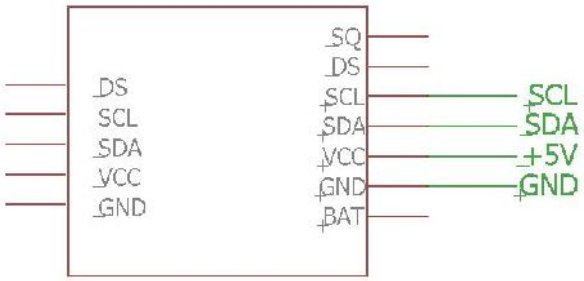
Sl 3.8. Shema napajanja sustava i trošila

U izradi sustava koristi se 5 LED modula ws2812b. Svaki modul ima 4 ulazna i 4 izlazna pina te se serijski povezuju jedan na drugi. Ulazna strana je ona gdje je DIN pin, a izlazna gdje je DOUT pin. Na Arduino se spaja samo DIN pin prvog modula, ali se između modula i Arduina treba spojiti otpornik vrijednosti od 100 Ω do 500 Ω radi zaštite ulaznog pina. Ostali moduli se serijski spajaju tako da DOUT pin prethodnog modula ide na DIN pin sljedećeg modula. Na slici 3.9. je prikazana shema spajanja LED modula.



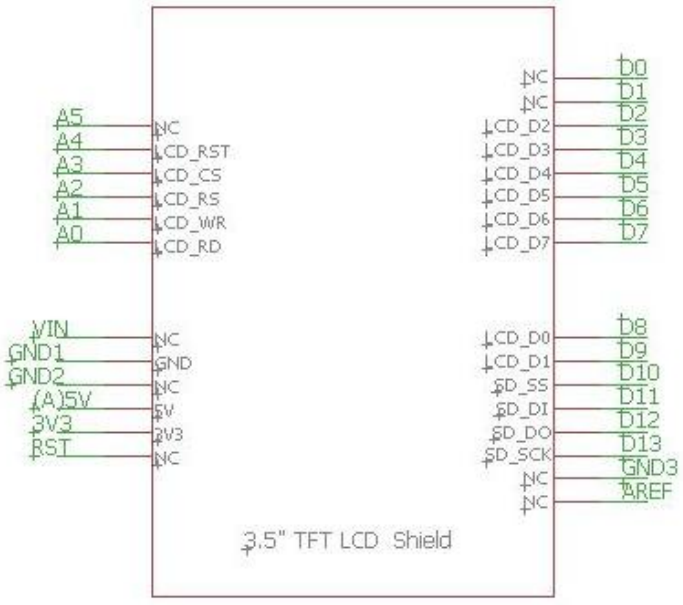
Sl 3.9. Shema spajanja LED modula

Za informaciju o trenutnom vremenu koristi se RTC DS1307. Komunikacija između RTC i Arduina odvija se preko I2C sabirnice i za komunikaciju su potrebne samo dvije žice. Na slici 3.10. prikazana je shema spajanja RTC modula na Arduino.



Sl 3.10. Shema spajanja RTC modula

Za prikaz vremena, brzine i kontrolnih lampica koristi se LCD pokaznik. LCD pokaznik integriran je na Arduino štit (engl. *Shield*) te za spajanje nisu potrebne žice. Konektori sa LCD štita spajaju se direktno na konektore Arduina te je time ujedno onemogućeno krivo spajanje. Shema spajanja LCD pokaznika na Arduino prikazana je na slici 3.11.



Sl 3.11. Shema spajanja LCD pokaznika

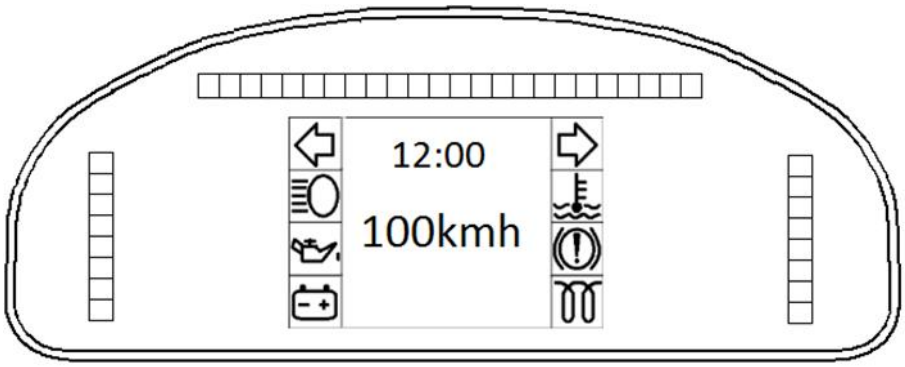
3.3. Izrada makete sustava

Za izradu makete sustava odnosno nove kontrolne ploče, trebalo je osmisliti na koji način će pojedine vrijednosti biti prikazane te kako će to sve izgledati. Izrada makete sastoji se od izrade kućišta i izrade elektroničkog dijela. Za bazu pri izradi kućišta neophodno je bilo koristiti originalnu kontrolnu ploču iz koje je izbačena originalna elektronika i prerađena je za postavljanje nove elektronike. Slika 3.12. prikazuje originalnu kontrolnu ploču prije i nakon izbacivanja nepotrebnih dijelova.



Sl 3.12. Originalna kontrolna ploča prije i poslije izbacivanja elektronike

Za prikaz temperature motora, količine goriva u spremniku i broj okretaja motora koristi se pet WS2812 LED modula spojenih u seriju. Po jedan modul koristi se za prikaz temperature motora i količine goriva u spremniku, a 3 modula za prikaz broja okretaja motora. Kod prikaza broja okretaja jedna LED dioda predstavlja 200 o/min, a kod prikaza temperature raspon prikaza je od 50°C do 120°C gdje jedna LED dioda predstavlja 10°C. Kod prikaza količine goriva u spremniku jedna LED dioda predstavlja 1/8 spremnika. LCD pokaznik koristi se za prikaz trenutne brzine, vremena i osam signalnih žaruljica. Na slici 3.13 prikazana je skica po kojoj je izrađen panel na koji se postavljaju WS2812 LED moduli i LCD pokaznik.



Sl 3.13. Skica panela

Za izradu panela korištena je pjenasta PVC ploča, poznata i po nazivu Forex ploča. Ploča se vrlo lagano reže skalpelom, tako da su utori vrlo jednostavno izrađeni. Ploča je bijele boje pa je zbog estetike na ploču zalijepljena crna karbonska folija. Na gotovi panel pričvršćeni su WS2812 LED moduli i LCD pokaznik te su ugrađeni u pripremljeno kućište kontrolne ploče. Elektroničke komponente zalemljene su na pertinaks ploču i iskorišten je konektor sa originalne kontrolne ploče tako da bi se nova kontrolna ploča mogla povezivati s automobilom na isti način kao i originalna. Na slici 3.14. prikazana je izrađena kontrolna ploča.



Sl 3.14. Digitalna kontrolna ploča

3.4. Programsko rješenje sustava

Zadatak Arduino programa je da pročita ulazne vrijednosti na pinovima Arduina, obradi ih tj. pretvori u oblik pogodan za prikaz te prikaže na LCD pokazniku i LED modulima. U programskom kodu su najprije definirane biblioteke te korištene varijable i konstante. U *setup* dijelu programskog koda nalaze se različite funkcije, a to su funkcije: za čitanje vrijednosti ulaznih pinova, za crtanje ikona za signalne lampice, za osvjetljavanje LED modula na osnovu broja okretaja, količine gorive u spremniku i temperature motora, za čitanje trenutnog vremena iz RTC, za prikaz trenutnog vremena i prikaz trenutne brzine kretanja automobila. U *loop* dijelu programskog koda pozivaju se funkcije za čitanje vrijednosti i prikaz vrijednosti, a one automatski pozivaju i sve ostale korištene funkcije.

4. ZAKLJUČAK

U ovome diplomskom radu izrađena je digitalna kontrolna ploča za prikaz parametara vozila VW Golf 3 1.9 TD. Prvo su izvršena mjerenja signala u automobilu, a nakon toga je određeno na koji način će određeni signali biti pročitani, obrađeni i prikazani u digitalnom obliku. Sljedeći korak je bila izrada elektroničke sheme sustava te su kupljene potrebne komponente koje su zalemljene na pertinaks pločici. S originalne kontrolne ploče je iskorišten konektor tako da bi se digitalna kontrolna ploča mogla jednostavno povezivati na automobil. Iz originalne kontrolne ploče izbačena je elektronika te je tako stvoren prostor za ugrađivanje nove elektronike i panela sa LED modulima i LCD pokaznikom. Digitalna kontrolna ploča je ugrađena u vozilo i uspješno izvršava svoju zadaću. Stara kontrolna ploča nije imala klasičan mjerač broja okretaja, nego se na tom mjestu nalazi sat za prikaz vremena. Tako da je najveća prednost digitalne kontrolne ploče upravo u brojaču okretaja motora. Za prikaz broja okretaja motora namijenjeno je 24 LED diode gdje jedna dioda predstavlja 200 o/min.

LITERATURA

[1] Ardunio razvjno okruženje, https://www.arduino.cc/en/guide/environment , kolovoz 2017.

[2] Arduino Mega, https://store.arduino.cc/arduino-mega-2560-rev3 , kolovoz 2017.

[3] Arduino Mega, https://www.sainsmart.com/media/catalog/product/i/m/img_0393__.jpg, kolovoz 2017.

[4] Što je to otpornik?, http://www.resistorguide.com/what-is-a-resistor/, kolovoz 2017.

[5] Fiksni otpornik, http://media.rs-online.com/t_large/R0132494-01.jpg, kolovoz 2017.

[6] Kondenzatori, http://ecee.colorado.edu/~mathys/ecen1400/labs/capxb.jpg, kolovoz 2017.

[7]LCD pokaznik, https://en.wikipedia.org/wiki/Liquid-crystal_display

[8] LCD pokaznik, https://images-na.ssl-images-amazon.com/images/I/51suk9bX8hL.jpg, kolovoz 2017.

[9] LED WS2812b, https://cdn.sparkfun.com/datasheets/BreakoutBoards/WS2812B.pdf, kolovoz 2017.

[10] Opto-izolator, https://en.wikipedia.org/wiki/Opto-isolator, kolovoz 2017.

[11] Opto-izolator, https://www.allaboutcircuits.com/uploads/articles/Figure10_CNY74-4H-HD_IC_pinout.png, kolovoz 2017.

[12] LM358, http://www.ti.com/lit/ds/symlink/lm158-n.pdf, kolovoz 2017.

[13]Shema LM358, http://www.learningaboutelectronics.com/images/LM358-dual-op-amp-pinout.png, kolovoz 2017.

[14] IRF 9540n, http://www.vishay.com/docs/91078/91078.pdf, kolovoz 2017.

[15] BC337-40, https://www.onsemi.com/pub/Collateral/BC337-D.PDF, kolovoz 2017.

[16] RTC DS1307, https://datasheets.maximintegrated.com/en/ds/DS1307.pdf, kolovoz 2017.

[17] Graf otpora i temperature, http://www.audifans.net/modules/Forums/files/thumbs/t_coolant_164.png, kolovoz 2017

[18] ECU, https://en.wikipedia.org/wiki/Engine_control_unit, kolovoz 2017.

SAŽETAK

Naslov: Mikroupravljački sustav za mjerenje i prikaz parametara vozila

U ovom diplomskom radu izrađena je digitalna kontrolna ploča za automobil. Izrada je temeljena na mikroupravljačkom sustavu Arduino Mega. Preko digitalnih i analognih ulaza Arduino prikuplja podatke, obrađuje ih i prikazuje na digitalnoj kontrolnoj ploči koja se sastoji od LCD pokaznika i LED modula. Iz originalne kontrolne ploče izbačena je postojeća elektronika te je kontrolna ploča iskorištena za ugradnje nove elektronike izrađene u ovom diplomskom radu. Digitalna kontrolna ploča je ugrađena u automobil i prikazuje parametre automobila.

Ključne riječi: Arduino, Kontrolna ploča, Automobil, LCD pokaznik, WS2812b

ABSTRACT

Title: Microcontroller system for measuring and displaying parameters of vehicle

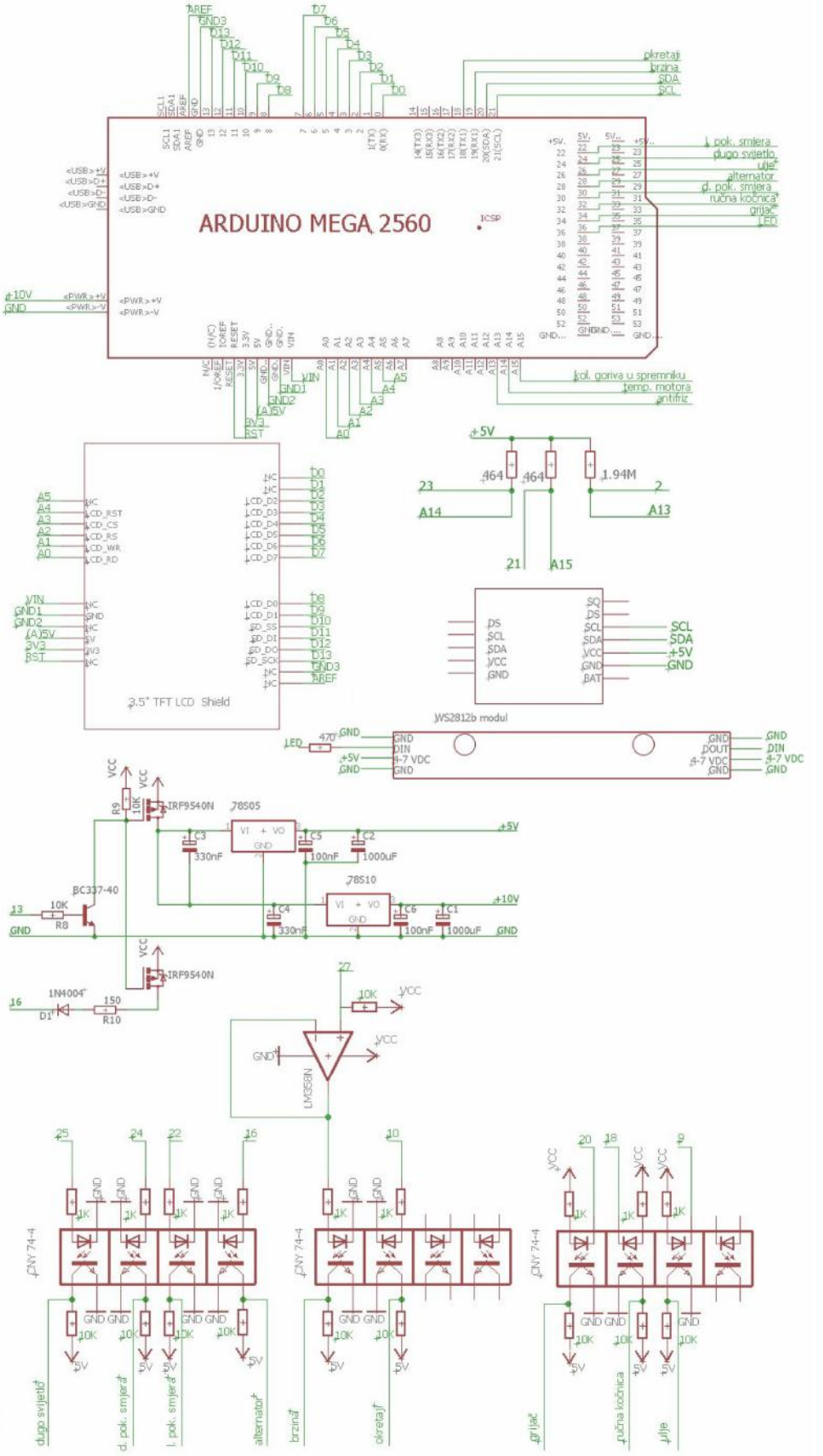
In this master's thesis was made digital control panel for the car. The design is based on Arduino Mega microcontroller system. Through digital and analogue inputs, Arduino collects data, processes and displays them on a digital control panel that consists of LCD display and LED modules. Existing electronics were removed from the original control panel and the control panel was used to install the new electronics made in this master's thesis. The digital control panel is built into the car and displays the car parameters.

Keywords: Arduino, Control panel, Car, LCD display, WS2812b

ŽIVOTOPIS

Damir Adrić rođen je u Đakovu dana 12.04.1992. Od rođenja živi u selu Novi Perkovci koje se nalazi u okolici Đakova. Osnovnu školu od prvog do četvrtog razreda pohađa u OŠ „Matija Gubec“ u Novim Perkovcima. Nakon završetka četvrtog razreda, osnovnu školu nastavlja pohađati u OŠ „Matija Gubec“ Piškorevci te ju završava odličnim uspjehom. Obrazovanje nastavlja upisom smjera računalni tehničar za strojarstvo u SSŠ „Braća Radić“ Đakovo. Srednju školu završava odličnim uspjehom te se 2011. godine upisuje na Elektrotehnički fakultet Osijek na preddiplomski studij računarstva. Preddiplomski studij završava 2014. godine, s temom završnog rada „Diofantske jednadžbe u programskom jeziku C“. Iste godine nastavlja obrazovanje upisom diplomskog studija na Elektrotehničkom fakultetu u Osijek, smjer procesno računarstvo.

PRILOG A. Kompletna shema sustava



PRILOG B. Slika sustava ugrađenog u automobil



PRILOG C. Program za Arduino Mega razvojni sustav

```
#include <TimerOne.h>
#include <TimerThree.h>
#include <Adafruit_NeoPixel.h>
#include <Adafruit_GFX.h>
#include <UTFTGLUE.h>
#include "icon.h"
#include <MD_DS1307.h>
#include <Wire.h>

#define LEDS_PIN     36
#define N_PIXELS     40
#define BRIGHTNESS   44
#define COLOR_DEPTH  24

#if COLOR_DEPTH == 24
#include "ColorMaps24.h"
#else
#undef COLOR_DEPTH
#define COLOR_DEPTH 128
#include "ColorMaps.h"
#endif

#define BLACK   0x0000
#define BLUE    0x001F
#define RED     0xF800
#define GREEN   0x07E0
#define CYAN    0x07FF
#define MAGENTA 0xF81F
#define YELLOW  0xFFE0
#define WHITE   0xFFFF
#define ORANGE  0xFD20

Adafruit_NeoPixel pixels = Adafruit_NeoPixel(N_PIXELS, LEDS_PIN, NEO_GRB +
NEO_KHZ800);
UTFTGLUE D(0x9481,A2,A1,A3,A4,A0);

#define ANALOG_VALUE_CHANGE 20
#define N_ANALOG_PINS 3
#define N_DIGITAL_PINS 7
#define N_INTERRUPT_PINS 2
int analogPin[N_ANALOG_PINS]={A14,A15,A13};
byte digitalPin[N_DIGITAL_PINS]={22,24,26,28,30,32,34};
byte interruptPin[N_INTERRUPT_PINS]={19,18};
int analogPinValue[N_ANALOG_PINS];
byte digitalPinValue[N_DIGITAL_PINS]={true,true,true,true,true,true,true};
bool analogPinDisplay[N_ANALOG_PINS]={false,false, false};
bool
digitalPinDisplay[N_DIGITAL_PINS]={false,false,false,false,false,false,false}
;
bool interruptPinDisplay[N_INTERRUPT_PINS]={false,false};
bool displayTime = false;
String timeString = "";
volatile unsigned long int c[3]={0,0};
volatile double f[2]={0x0000,0x0000};
volatile double T[2]={0.500, 0.100};
float R1 = 464;
float R2 = 1940000;
float tempResLevel[8] = {900,700,530,400,275,200,150,70};
int t=119;
```

```
void setup() {
  //setTime(12,34);
  pinMode(A13, INPUT);
  pinMode(A14, INPUT);
  pinMode(A15, INPUT);
  analogReference(DEFAULT);
  Serial.begin(9600);
  pinMode(A0, OUTPUT);
  digitalWrite(A0, HIGH);
  D.InitLCD();
  D.clrScr();
  D.setFont(BigFont);
  D.setTextSize(7);
  for(int i=0; i<8; i++){DrawIcon(i, digitalPinValue[i]);}


  pixels.begin();
  pixels.setBrightness(BRIGHTNESS);

  for(int i=0;i<N_DIGITAL_PINS;i++){
    pinMode(digitalPin[i],INPUT);
  }

  Timer1.initialize(1000000.0f*T[0]);
  Timer1.attachInterrupt(FreqRead1);
  Timer3.initialize(1000000.0f*T[1]);
  Timer3.attachInterrupt(FreqRead2);
  attachInterrupt(digitalPinToInterrupt(interruptPin[0]), brzina, RISING);
  attachInterrupt(digitalPinToInterrupt(interruptPin[1]), obrtaj, RISING);
}


void DrawIcon(uint8_t i, bool state) {
  unsigned int x0, y0, x1, y1;
  x0 = (i<4) ? 4 : 404;
  y0 = (i<4) ? i*80+4 : (i-4)*80+4;
  x1 = (i<4) ? 0 : 400;
  y1 = (i<4) ? i*80 : (i-4)*80;
  switch(i){
    case 1:
      if(state) {
        D.setColor(BLACK);
        D.fillRect(x1,y1,x1+80,y1+80);
        D.setColor(GREEN);
      } else {
        D.setColor(BLUE);
        D.fillRect(x1,y1,x1+80,y1+80);
        D.setColor(WHITE);
      }
      break;
    case 0:
    case 4:
      if(state) {
        D.setColor(BLACK);
        D.fillRect(x1,y1,x1+80,y1+80);
        D.setColor(GREEN);
      } else {
        D.setColor(GREEN);
        D.fillRect(x1,y1,x1+80,y1+80);
        D.setColor(BLACK);
```

```
      }
      break;
    case 3:
      if(state) {
        D.setColor(RED);
        D.fillRect(x1,y1,x1+80,y1+80);
        D.setColor(BLACK);
      } else {
        D.setColor(BLACK);
        D.fillRect(x1,y1,x1+80,y1+80);
        D.setColor(GREEN);
      }
      break;
    case 5:
    case 2:
    case 6:
      if(state) {
        D.setColor(BLACK);
        D.fillRect(x1,y1,x1+80,y1+80);
        D.setColor(GREEN);
      } else {
        D.setColor(RED);
        D.fillRect(x1,y1,x1+80,y1+80);
        D.setColor(BLACK);
      }
      break;
    case 7:
      if(state) {
        D.setColor(BLACK);
        D.fillRect(x1,y1,x1+80,y1+80);
        D.setColor(GREEN);
      } else {
        D.setColor(ORANGE);
        D.fillRect(x1,y1,x1+80,y1+80);
        D.setColor(BLACK);
      }
      break;
    default: break;
  }
  for(unsigned int j=0; j<72; j++) {
    for(unsigned int k=0; k<9; k++) {
      uint8_t b = icon[i][k+j*9];
      for(uint8_t m=0; m<8; m++) {
        if(b&(1<<(7-m))) {
          unsigned int x = j;
          unsigned int y = k*8+m;
          D.drawPixel(x+x0,y+y0);
        }
      }
    }
  }
  D.setColor(GREEN);
}

void setLEDtemp(byte x) {
  int i,j;
  for(i=0 ; i<x ; i++) {
    j = map(i,0,8,0,COLOR_DEPTH-1);
    pixels.setPixelColor(i, pixels.Color(cm[j][0],cm[j][1],cm[j][2]));
  }
  for(   ; i<8 ; i++) {
```

```
    pixels.setPixelColor(i,0,0,0,0);
  }
  pixels.show();
}

void setLEDrpm(int x) {
  int i,j;
  for(i=0 ; i<x ; i++) {
    j = map(i,0,23,0,COLOR_DEPTH-1);
    pixels.setPixelColor(i+8, pixels.Color(cm[j][0],cm[j][1],cm[j][2]));
  }
  while(i<24) { pixels.setPixelColor(i+ 8,0,0,0,0); i++; }
  pixels.show();
}

void setLEDfuel(byte x) {
  int i,j;
  for(i=0 ; i<x ; i++) {
    j = map(i,0,8,0,COLOR_DEPTH-1);
    pixels.setPixelColor(39-i, pixels.Color(cm[j][2],cm[j][1],cm[j][0]));
  }
  while(i<8) { pixels.setPixelColor(39-i,0,0,0,0); i++; }
  pixels.show();
}

String ConvertToString3(int x) {
  char s[] = {' ',' ','0',0};
  int i = 2;
  while(x>0 && i>=0) {
    s[i--] = (x%10) + 48;
    x = x / (int) 10;
  }
  return String(s);
}
String ConvertToString4(int x) {
  char s[] = {' ',' ',' ','0',0};
  int i = 3;
  while(x>0 && i>=0) {
    s[i--] = (x%10) + 48;
    x = x / (int) 10;
  }
  return String(s);
}

void brzina(){c[0]++;}
void obrtaj(){c[1]++;}

void FreqRead1() {
    f[0]=double(c[0])/T[0];
    c[0]=0x00000000;
    interruptPinDisplay[0]=true;
    t++;
}
void FreqRead2() {
  f[1]=double(c[1])/T[1];
  c[1]=0x00000000;
  interruptPinDisplay[1]=true;
}

byte tempResookUpTable(float R2){
  if (R2 > tempResLevel[0]                          ){ return 0; } // <50
```

```
  if (R2 > tempResLevel[1] && R2 <= tempResLevel[0] ){ return 1; } // 50
  if (R2 > tempResLevel[2] && R2 <= tempResLevel[1] ){ return 2; } // 60
  if (R2 > tempResLevel[3] && R2 <= tempResLevel[2] ){ return 3; } // 70
  if (R2 > tempResLevel[4] && R2 <= tempResLevel[3] ){ return 4; } // 80
  if (R2 > tempResLevel[5] && R2 <= tempResLevel[4] ){ return 5; } // 90
  if (R2 > tempResLevel[6] && R2 <= tempResLevel[5] ){ return 6; } // 100
  if (R2 > tempResLevel[7] && R2 <= tempResLevel[6] ){ return 7; } // 110
  if (                        R2 <= tempResLevel[7] ){ return 8; } // >110
}

inline void readPinValues(){
  for(int i=0;i<N_DIGITAL_PINS; i++){
    byte value = digitalRead(digitalPin[i]);
    if(digitalPinValue[i]!=value){
      digitalPinDisplay[i]=true;
      digitalPinValue[i] = value;
    }
  }
  for(int i=0;i<N_ANALOG_PINS;i++){
    int value = analogRead(analogPin[i]);
    if(abs(analogPinValue[i]-value)>ANALOG_VALUE_CHANGE){
      analogPinValue[i]=value;
      analogPinDisplay[i]=true;
    }
  }

}

inline void displayValues(){
  for(int i=0;i<N_DIGITAL_PINS;i++){
    if(digitalPinDisplay[i]){
      DrawIcon(i<5 ? i : i+1,digitalPinValue[i]);
      digitalPinDisplay[i]=false;
    }
  }
  if(analogPinDisplay[0]){
    float Vout1 = analogPinValue[0];
    float R2=R1/((1023/Vout1)-1);
    byte temp = tempResookUpTable(R2);
    setLEDtemp(temp);
    analogPinDisplay[0]=false;
  }
  if(analogPinDisplay[1]){
    float Vout2 = analogPinValue[1];
    float R3=R1/((1023/Vout2)-1);
    R3 = R3 > 160 ? 0 : 160-R3;
    byte fuel=map(R3,     0   ,   160   ,0, 8);
    setLEDfuel(fuel);
    analogPinDisplay[1]=false;
  }
  if(analogPinDisplay[2]){
    float Vout3 = analogPinValue[2];
    float R4=R2/((1023/Vout3)-1);
    if(R4>12000000){DrawIcon(5,false);}
    else {DrawIcon(5,true);}
    analogPinDisplay[2]=false;
  }
  if(interruptPinDisplay[0]){
    int s = f[0];
    D.print(ConvertToString4(s),80,150);
    interruptPinDisplay[0]=false;
```

```
    char k[3]= "kmh";
    D.print(k,250,150);
  }
  if(interruptPinDisplay[1]){
    int s=map(f[1],      0   ,   1600   ,0,24);
    setLEDrpm(s);
    interruptPinDisplay[1]=false;
  }
  if(t==120){
    D.print(getTime(),140,30);
    t=0;
    }

}
void setTime(uint8_t h, uint8_t m){
  RTC.control(DS1307_ON,      true);
  RTC.control(DS1307_SQW_LOW, true);
  RTC.h = h;
  RTC.m = m;
  RTC.writeTime();
}

String getTime()
{
  RTC.readTime();
  return String(ConvertToString2(RTC.h)) + ":" +
String(ConvertToString2(RTC.m));
}

String ConvertToString2(int x){
  static char c[3] = { "00" };
  c[0] = ((x / 10) % 10) + '0';
  c[1] = (x % 10) + '0';
  return(c);
}

void loop()
{
  readPinValues();
  displayValues();

      }
```