Mikroupravljački sustav za emulaciju električnih signala u automobilu

Vuković, Matej

Undergraduate thesis / Završni 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:200:571661

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

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 EMULACIJU ELEKTRIČNIH SIGNALA U AUTOMOBILU

Završni rad

Matej Vuković

Osijek, 2017

SADRŽAJ

1. UVOD 1
1.1. Zadatak završnog rada 1
2. PRIMJENJENE TEHNOLOGIJE I ALATI 2
2.1. Arduino razvojno okruženje 2
2.2. Arduino UNO 3
2.3. Opto-izolatori 4
2.4. Otpornici 5
2.5. Digitalni potenciometar 5
2.6. Microsoft Visual Studio 6
2.6.1. C#, Windows Forms Application 7
3. REALIZACIJA SUSTAVA 8
3.1. Izrada sklopa 8
3.2. Signali u automobilu 10
3.3. Programsko rješenje sustava 11
3.4. Aplikacija 11
3.4.1. Rad aplikacije 13
4. ZAKLJUČAK 15
LITERATURA 16
SAŽETAK 17
ABSTRACT 18
ŽIVOTOPIS 19
PRILOZI 20

1. UVOD

Cilj ovog završnog rada je osmisliti mikroupravljački sustav koji će služiti za emulaciju električnih signala za automobil VW Golf 3. Realizacija ovog sustava može se podijeliti u dva djela. U prvom djelu potrebno je izraditi elektronički sklop koji će biti moguće povezati na kontrolnu ploču automobila. Nakon povezivanja sa kontrolnom pločom potrebno je napisati Arduino kod koji će omogućiti slanje električnih signala na ploču. Drugi dio je izrada stolne (eng. *desktop*) aplikacije u programu C# koja će na jednostavan način kontrolirati rad Arduina odnosno koja će davati naredbe za emuliranje signala.

1.1. Zadatak završnog rada

Zadatak ovog završnog rada je omogućiti emulaciju signala u automobilu. U radu treba izraditi mikroupravljački sklop implementirati ga pomoću Arduina te izraditi C# stolnu aplikaciju.

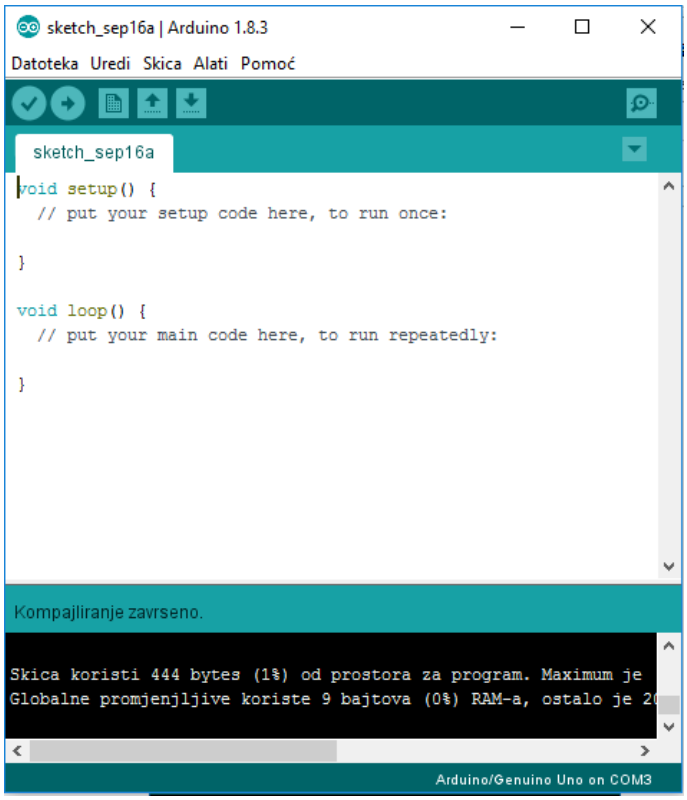
2. PRIMJENJENE TEHNOLOGIJE I ALATI

2.1. Arduino razvojno okruženje

Arduino je platforma otvorenog tipa (eng. *open-source)* za kreiranje elektroničkih prototipova bazirana na sklopovlju i programskom paketu koji je fleksibilan i jednostavan za korištenje [1]. Jednostavnije rečeno Arduino platforma je skup elektroničkih i programskih dijelova koji se mogu jednostavno povezati u složenije cjeline s ciljem izrade zabavnih i poučnih sklopova [2]. Pošto je Arduino platforma otvorenog tipa dozvoljeno je njezino preuređivanje i dijeljenje u svrhu kreiranja novih platformi tako da je proizvedeno još mnogo inačica razvojnih okruženja baziranih na Arduino platformi.

Arduino se sastoji od dva glavna dijela. Prvi dio je *hardware* kojeg predstavlja mikroupravljač te drugi, *software*, kojeg čini *Arduino IDE* (eng. *Integrated Development Environment*) programski paket. Mikroupravljač predstavlja malo računalo smješteno na integriranom sklopu koji sadrži razne dijelove kao što su procesor, radna i programska memorija te ulazno izlazne jedinice. *Arduino IDE* je višeplatformska aplikacija napisana u programskom jeziku Java.

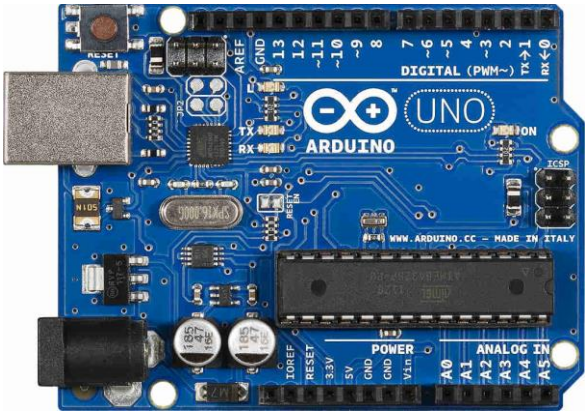
Arduino *IDE* sastoji se od dvije glavne funkcije: *setup()* i *loop().* Prva funkcija *setup()* predstavlja funkciju koja će se samo jednom izvršiti nakon pokretanja programa. Ona služi za definiranje početnih varijabli koje će se koristiti u daljnjem programu. Funkcija *loop()* se poziva nakon *setup()* funkcije ona je glavna i osnovna funkcija svakog programa napisanog za Arduino. U *loop()* funkciji nalaze se sve naredbe koje su dane da Arduino izvrši. Za razliku od *setup*() naredbe *loop()* naredba se izvršava beskonačno mnogo puta, dokle god je Arduino uključen. Treba napomenuti da svaki Arduino program mora imati i *setup*() i *loop*() funkciju neovisno jesu li im zaglavlja prazna ili ne [3].



Sl. 2.1. Arduino *software*

2.2. Arduino UNO

Zbog svojih specifikacija i jednostavnosti za realizaciju ovog sustava odabrana je Arduino UNO platforma. Arduino UNO jedna je od prvih, a ujedno i jedna od najjednostavnijih verzija Arduina. Ova Arduino platforma bazirana je na ATmega328P čipu. Sadrži 14 digitalnih *input/output* pinova, šest analognih ulaza, USB priključak, ICSP, 16 MHz kvarcni kristal i *reset* tipku. Tehničke karakteristike prikazane su na slici 2.2.



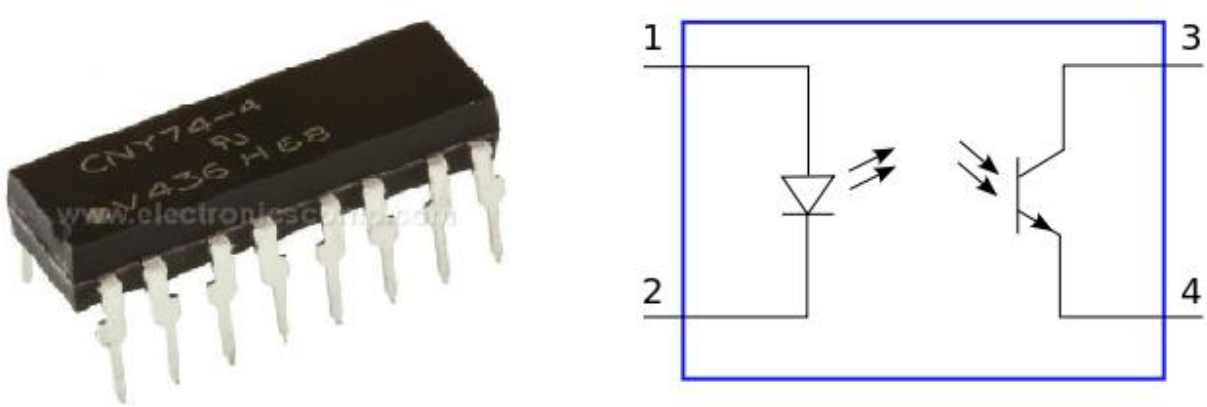
Sl. 2.2. Arduino UNO [4]

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Sl. 2.3. Tehničke karakteristike Arduino UNO platforme [5]

2.3. Opto-izolatori

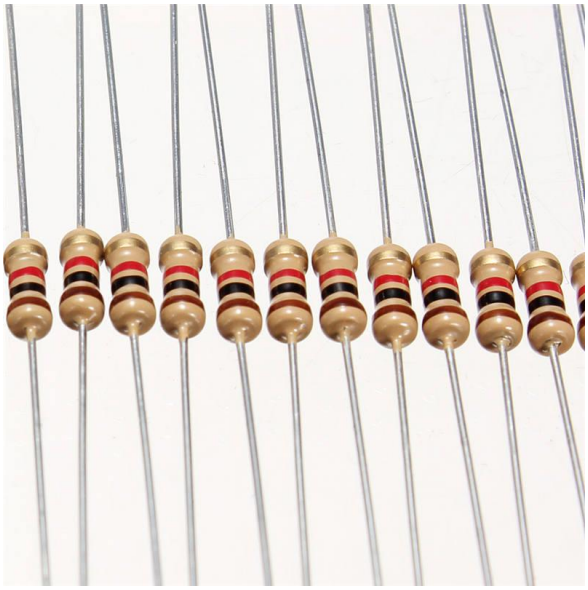
Opto-izolator (eng. *optocoupler*) je elektronička komponenta koja prenosi električne signale između dva izolirana strujna kruga putem svjetlosti. Na jednoj strani opto-izolatora nalazi se LE dioda koja električni ulazni signal pretvara u svjetlost dok se na drugoj strani nalazi fototranzistor koji prima tu svijetlost. Komercijalno dostupni opto-izolatori mogu izdržati napone do 10 kV sa prijenosnim brzinama do 10 kV/μs [6].



Sl. 2.4. Opto-izolator i njegov shematski prikaz [7]

2.4. Otpornici

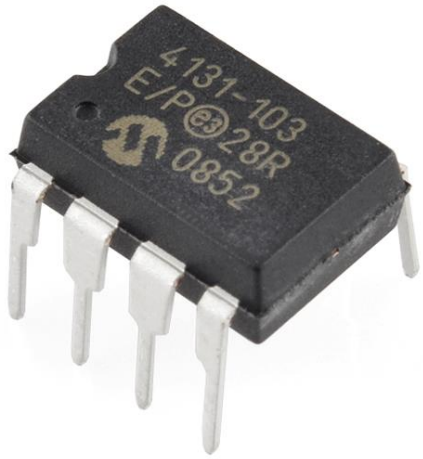
Otpornici su pasivne elektroničke komponente koje pružaju otpor protjecanju električne struje, pri čemu je odnos napona i jakosti struje skladan sa Ohmovim zakonom. Karakteristična veličina otpornika jest električni otpor. Električni otpor jednak je naponu na otporniku podijeljenom sa strujom koja protječe kroz otpornik. Mjerna jedinica za električni otpor je Ohm.[8]



Sl. 2.5. Otpornici [9]

2.5. Digitalni potenciometar

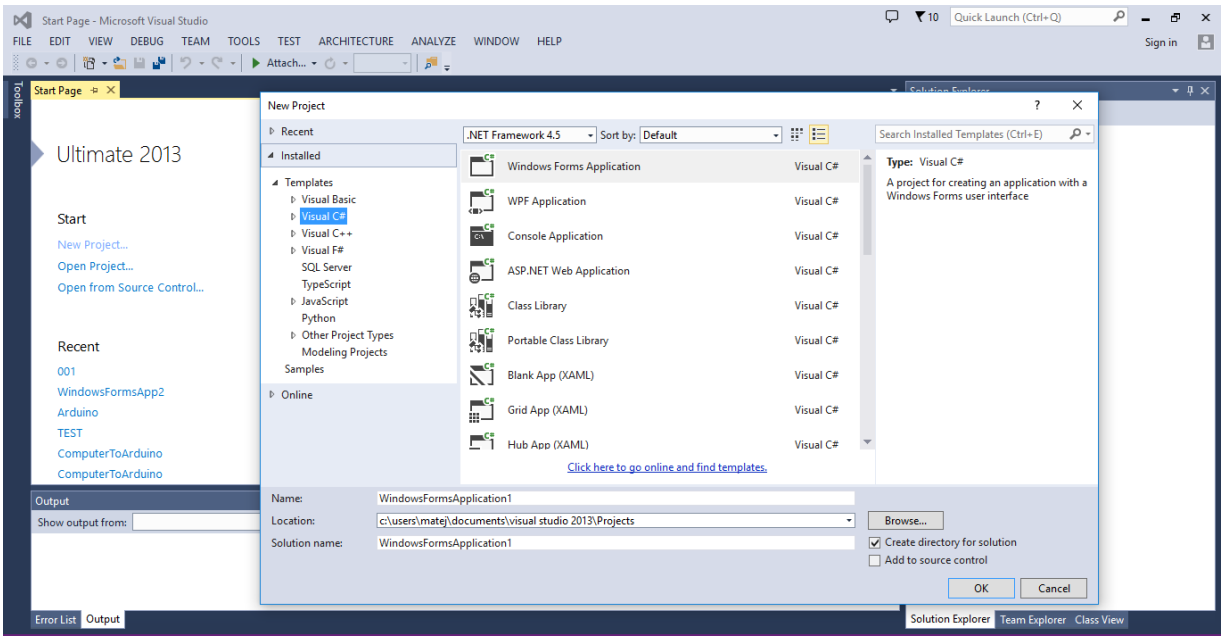
Digitalni potenciometar koji se još naziva otporni digitalno-analogni pretvarač, digitalno je kontrolirana elektronička komponenta koja oponaša analogne funkcije potenciometra. Koristi se za podrezivanje i skaliranje analognih signala putem mikroupravljača.



Sl. 2.6. Digitalni potenciometar [10]

2.6. Microsoft Visual Studio

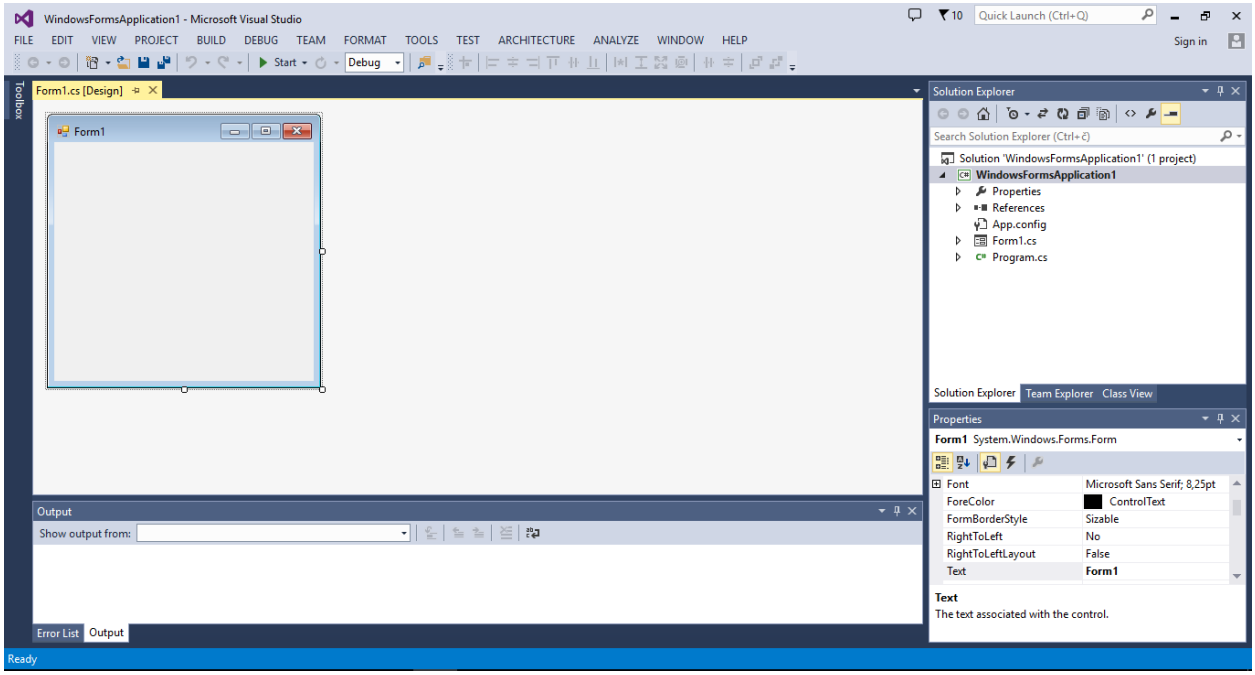
Microsoft Visual Studio je integrirano razvojno okruženje (eng. *Integrated Development Environment*) koje je proizveo Microsoft. Visual Studio koristi se za razvoj programa za Microsoft Windows, mobilnih i web aplikacija, web stranica te sličnih stvari. Visual Studio podržava različite programske jezike. Već ugrađeni jezici unutar Visual-a su C, C++, C++/CLI, VB.NET, C# i F#. Također pruža podršku i za programske jezike kao što su M, Phyton, Ruby, XML/XSLT, HTML/XHTML, Javascript i CSS [11].



Sl. 2.7. Microsoft Visual Studio

2.6.1. C#, Windows Forms Application

C# je objektno orijentiran programski jezik razvijen od strane Microsoft-a. Proizveden je 2000. godine s ciljem da bude moderan, siguran i jednostavan objektno orijentiran programski jezik sa visokim performansama baziran na .NET platformi. C# je vrlo sličan C++ i Java jeziku na temelju kojih je i nastao (sintaksa i semantika je velikim dijelom preuzeta iz Jave, koja je poput C# potpuno objektno orijentirani jezik). Za razliku od Jave C# nije neovisan o platformi, odnosno operativnom sustavu, već je razvijen za izradu stolnih te internet aplikacija u .Microsoft .NET okruženju [12]. Za izradu grafičke aplikacije u C#-u koristi se predložak *Windows Forms Aplication*.



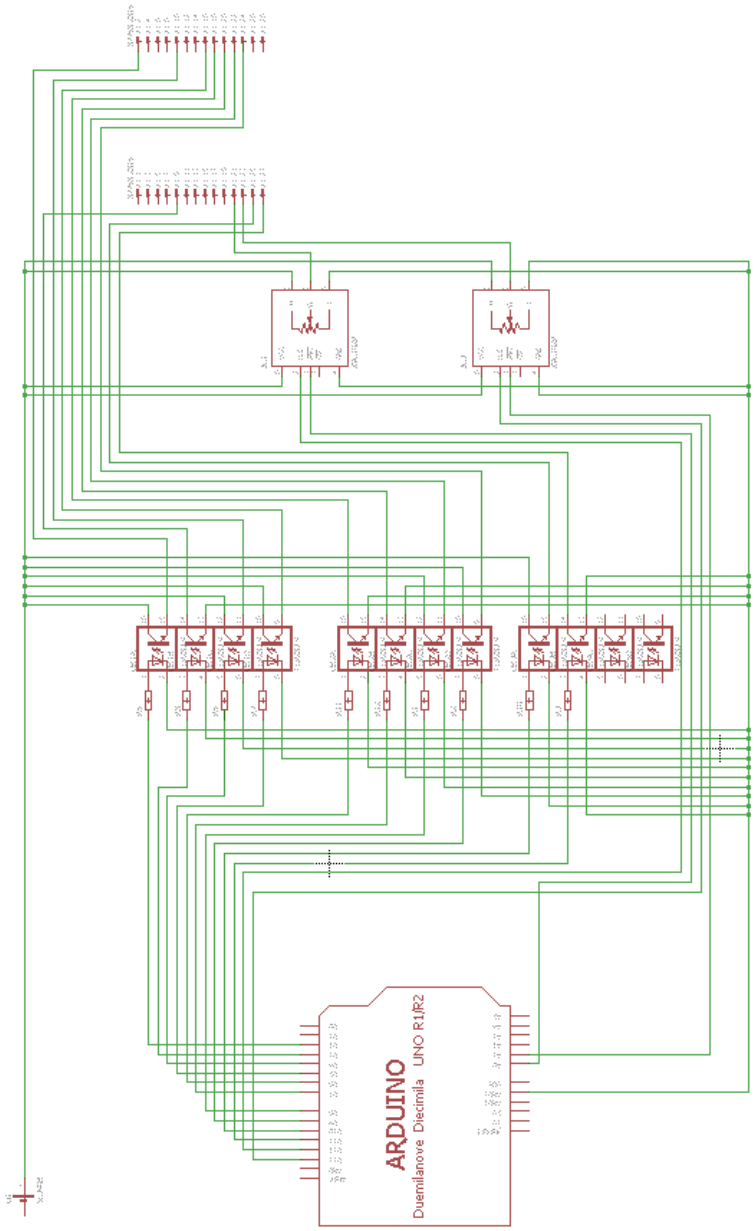
Sl. 2.8. Izgled praznog Windows Forms Aplication predloška u C#-u

3. REALIZACIJA SUSTAVA

Realizacija sustava sastoji se iz dva dijela, započinje izradom mikroupravljačkog sklopa nakon čega slijedi programiranje njegovog mikroupravljača te završava izradom C# aplikacije.

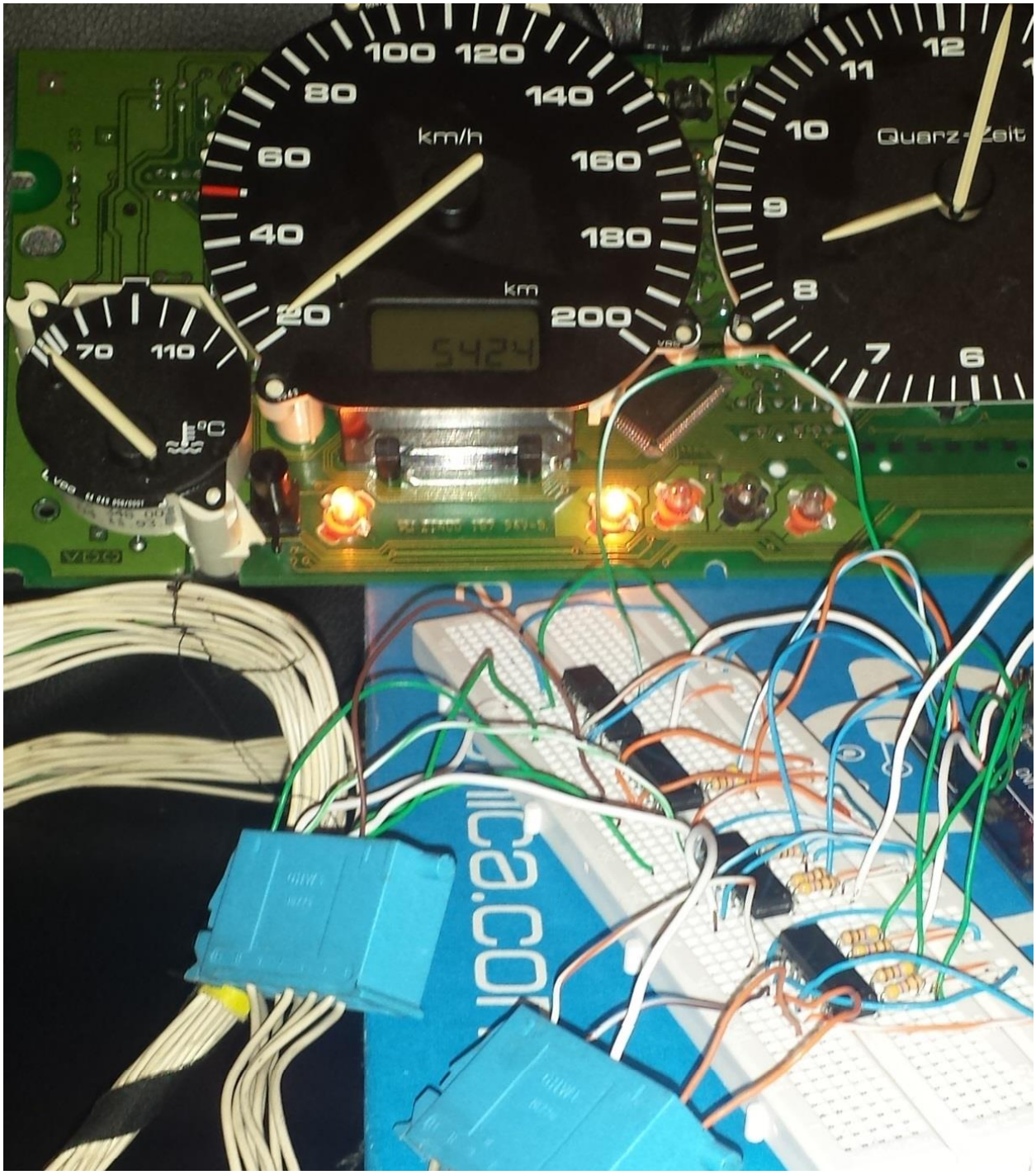
3.1. Izrada sklopa

Prije same izrade bilo je potrebno je napraviti shemu po kojoj će sklop biti sastavljen. Za izradu sheme korišten je programski paket Cadsoft Eagle Layout Editor. Na slici 3.1. prikazana je shema sustava kreirana u programu Eagle .



Sl. 3.1. Shema sklopa

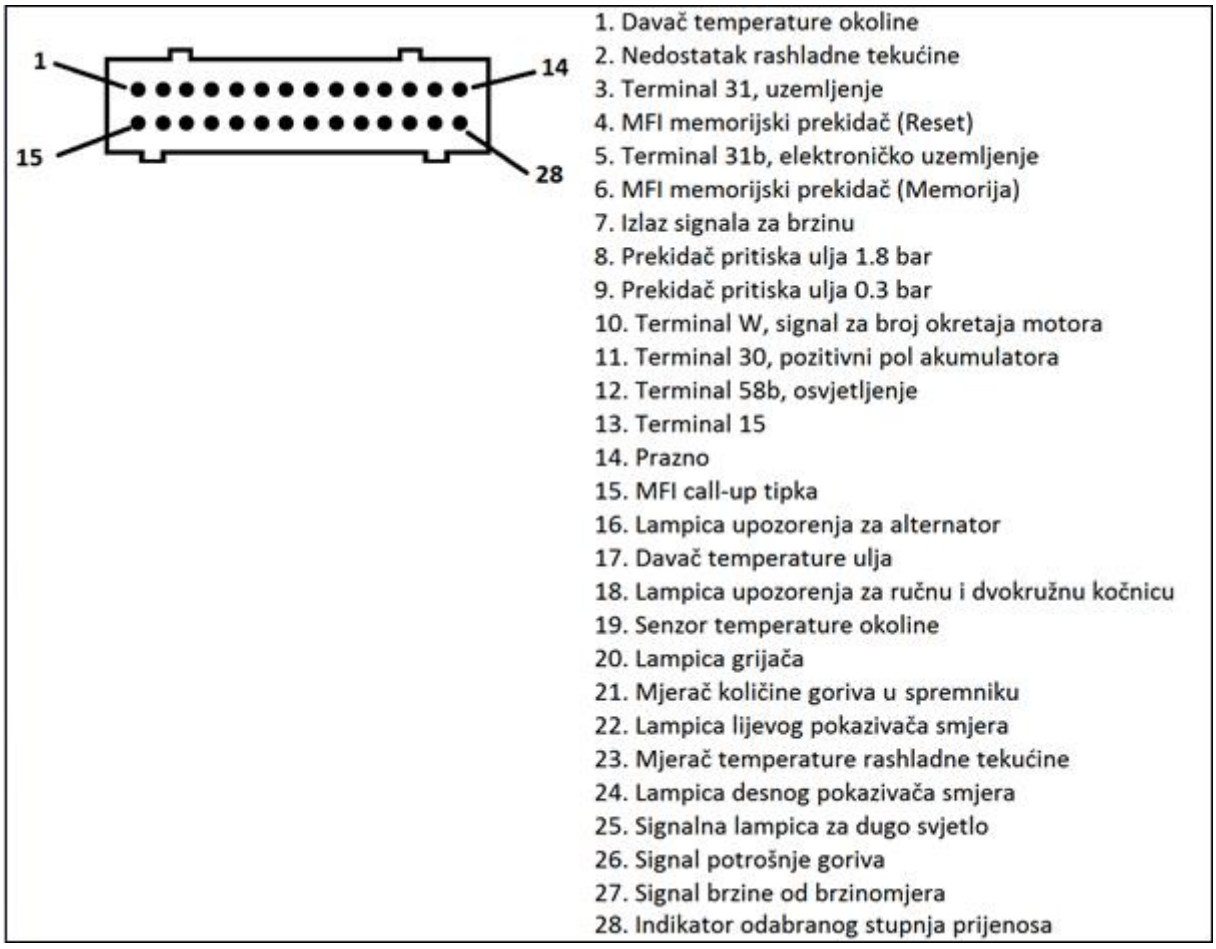
Za izradu sklopa korišten je Arduino UNO, otpornici od 180 ohma, 3 opto-izolatora i 2 digitalna otpornika. Uz Arduino bilo je potrebno odabrati dodatni izvor napajanja pošto Arduino može dati samo napon od 5 V (volta), a za kontrolnu ploču potrebno je 12 V koliko proizvodi automobil. Kao dodatni, vanjski izvor napajanja odabran je AC/DC adapter, model RED-1220 napona 12V i jakosti struje 2 A (ampera). Na slici 3.2. prikazan je sklop nakon spajanja komponenti.



Sl. 3.2. Izgled sklopa spojenog na kontrolnu ploču

3.2. Signali u automobilu

Veza između kontrolne ploče i mikroupravljačkog sklopa to jest veza koja će omogućiti komunikaciju između Arduina i kontrolne ploče je 28-pinski priključak tako zvani *jack* priključak. *Jack* priključak kao što je spomenuto sadrži 28 pinova odnosno 2 reda po 14 pinova. Svaki pin je zadužen za obavljanje neke funkcije na kontrolnoj ploči kao što su primjerice paljenje lijevog i desnog pokazivač smjera, mjerenje količine goriva u spremnik, paljenje lampice upozorenja za ručnu kočnicu i slično. Na slici 3.2. prikazan je općeniti raspored svih pinova unutar VW Golfa 3 te njihove funkcije.



Sl. 3.3. 28-pinski priključak kontrolne ploče u automobilu (Jack)

U ovom radu neće biti korišteni svi pinovi jer broj pinova ovisi o motoru i opremi automobila. Pinovi koji će biti korišteni za ovaj sustav su pin broj 2, 9, 10, 16, 18, 20, 21, 22, 23, 24, 25 i 27 odnosno pin za nedostatak rashladne tekućine, pritisak ulja, broj okretaja motora, lampicu alternatora, ručnu kočnicu, lampicu grijača, količinu goriva u spremniku, lijevi pokazivač smjera,

temperaturu rashladne tekućine, desni pokazivač smjera, dugo svijetlo te brzinu. Dakle ukupno 12 signala je potrebno emulirati.

3.3. Programsko rješenje sustava

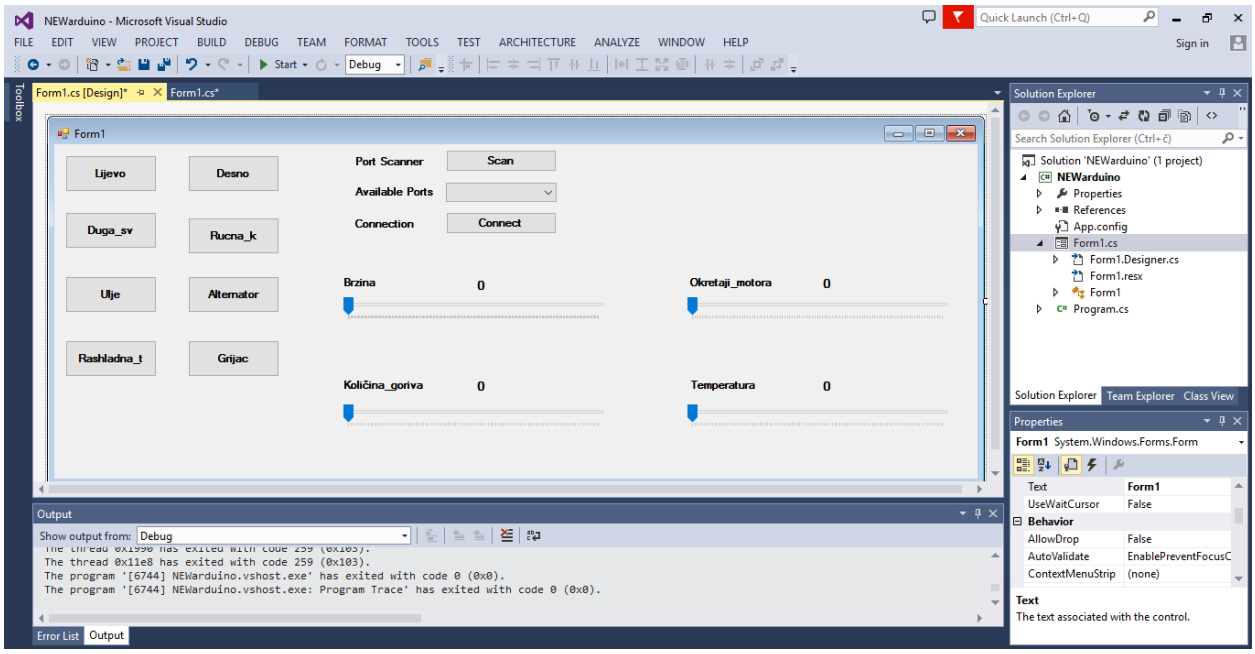
Za programiranje mikroupravljača korišten je Arduino *Software* (*IDE*). Arduino *software* čine dvije glavne funkcije: *setup*() i *loop*(). *Setup*() se izvodi se na početku kada se Arduino upali dok *loop*() funkcija predstavlja beskonačnu petlju koja je ujedno glavni program Arduina.

Program započinje deklariranjem varijabli odnosno pridjeljuju im se pinovi na Arduinu putem kojih će se slati signali. U *setup*() funkciji definirana je brzina kojom će Arduino raditi i određeni su OUTPUT pinovi. U glavnoj funkciji kod je implementiran tako da čita sve podatke koji se šalju na *Serial port*. Pročitani podatak ispituje odnosno uspoređuje ih sa zadanim vrijednostima. Ako neka vrijednost odgovara podatku Arduino šalje signal na odgovarajući pin koji preko 28-pinskog priključka dolazi do kontrolne ploče automobila i obavlja svoju funkciju.

3.4. Aplikacija

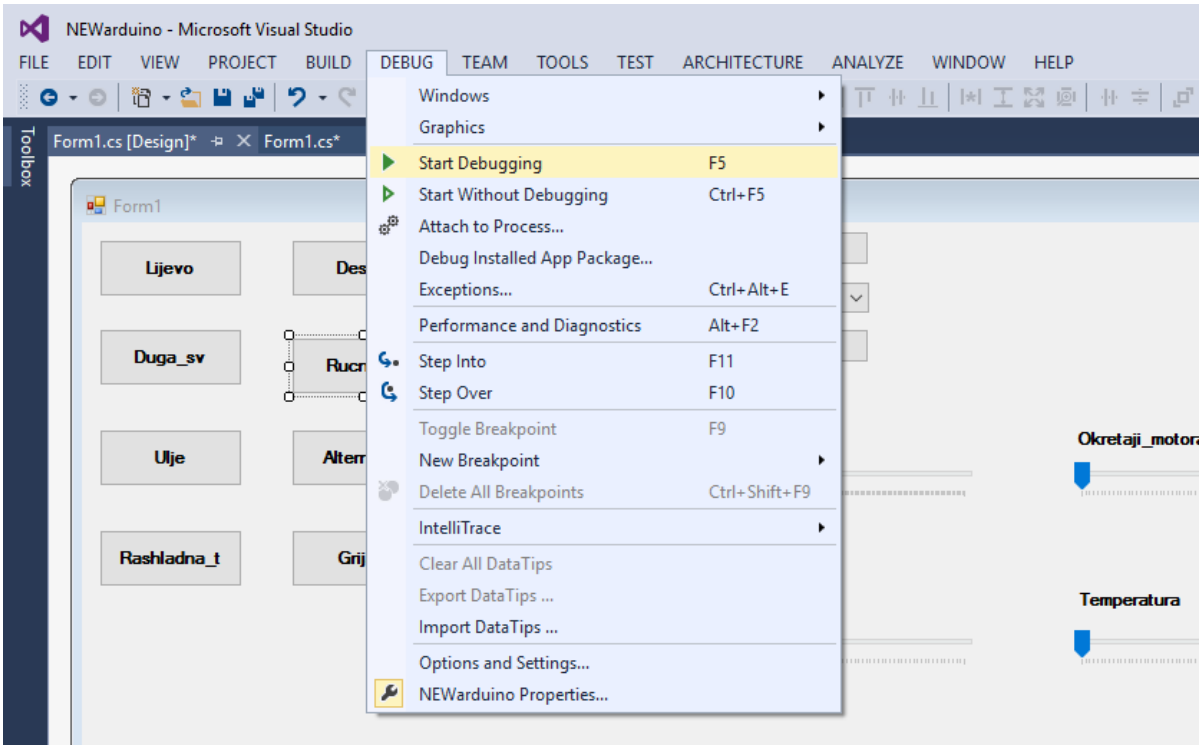
Nakon napravljenog sklopa i njegove implementacije slijedi izrada aplikacije. Zbog svojih mogućnosti za izradu ove aplikacije odabran je programski jezik C# iz paketa Microsoft Visual Studio.

Sama aplikacija će biti rađena u predlošku Windows Forms Application. Prije svega bilo je potrebno bilo je osmisliti dizajn aplikacije. Kao što je ranije spomenuto ovaj sustav treba emulirati signale za 12 pinova stoga je za slanje signala u aplikaciji odabrano 8 tipki i 4 klizaća. Signali koji će ići preko tipki su nedostatak rashladne tekućine, pritisak ulja, lampica alternatora, ručna kočnica, lampicu grijača, lijevi i desni pokazivač smjera te dugo svijetlo. Preostala 4 klizaća će predstavljati brzinu, broj okretaja motora, temperaturu i količinu goriva u spremniku.



Sl. 3.4. Dizajn aplikacije

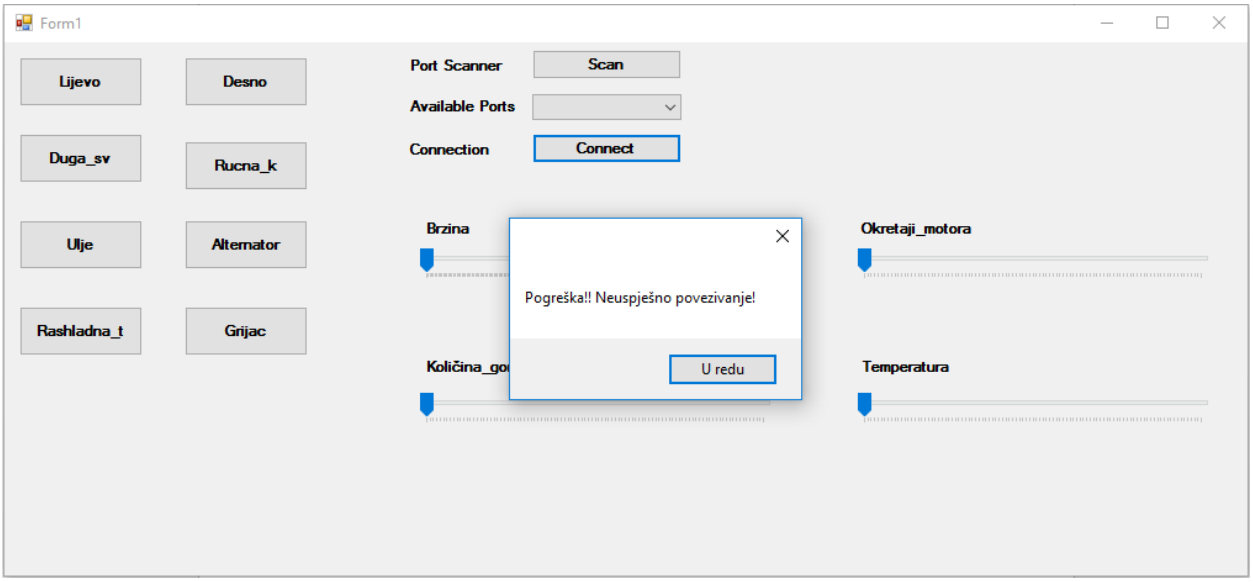
Osim tipki i klizača za slanje signala, u aplikaciji se nalaze još dvije tipke i jedan *combobox* koji zajedno čine program za pretraživanje dostupnih portova i povezivanje putem putem *Serial Porta*. Spomenuti program je vrlo važan jer se na taj način se vrši komunikacija između Arduina i C#. Aplikacija se pokreće odabirom opcije *Start Debugging*.



Sl. 3.5. Pokretanje aplikacije

3.4.1. Rad aplikacije

Nakon što je pokrenuta aplikacija prvo što je potrebno učiniti je povezati aplikaciju sa Arduinom tj. omogućiti komunikaciju putem *Serial porta*. To će se učiniti klikom na tipku *Scan* koji skenira to jest pretražuje *Serial port* i daje listu dostupnih portova. Nakon toga, odabire se povoljan port te klikom na tipku *connect* povezuje sa odabranim, ovo je nužan korak jer bez njega aplikacija ne može dalje funkcionirati. Ukoliko je pak došlo do nekakve greške i nije se uspjelo povezati sa portom, aplikacija će izbaciti upozorenje da se pojavila pogreška i potrebno je pokušati ponovno.



Sl. 3.6. Aplikacija javlja grešku

Kada se C# uspješno poveže sa Arduinom može početi operativni rad aplikacije. Aplikacija radi tako da klikom na neku od tipki ili pomicanjem nekog klizača određeni podatak se automatski šalje na *Serial port*, Arduino čita podatak sa *Serial porta* i ispituje je li poslan podatak odgovara zadanoj vrijednosti. Ako vrijednost odgovara Arduino šalje signal na određeni pin *jack* priključka preko kojeg signal putuje do kontrolne ploče i naposljetku izvršava svoju funkciju.

Izvršen je primjer u pokrenutoj aplikaciji gdje se sa klizaćem za signal brzine odabrala vrijednost 97, odabrana vrijednost poslana je na *Serial port* gdje ju Arduino čita, prihvaća i prosljeđuje signal na kontrolnu ploču koja kazaljku od brzine pomiče na 97 km/h.



Slika 3.7. Prikaz rada aplikacije

4. ZAKLJUČAK

U ovom završnom radu napravljen je mikroupravljački sustav za emulaciju električnih signala u automobilu. Sustav se sastoji od dvije veće cjeline: mikroupravljački sklop i njegova Arduino implementacija te C# stolna aplikacija. Sustav radi tako da se putem C# aplikacije šalje podatak koji Arduino čita i ako je podatak odgovarajući Arduino dalje šalje električni signal preko mikroupravljačkog sklopa na kontrolnu ploču automobila. Emulacija signala putem ovog sustava je relativno spora, veća brzina mogla bi se postići korištenjem Arduino platforme sa bržim čipom kao što je Arduino DUE.

LITERATURA

[1] Arduino, https://www.arduino.cc, rujan 2017.

[2] Knjiga: Paolo Zenzerović, ARDUINO kroz jednostavne primjere, Zagreb, 2014.

[3] Internet: Arduino IDE, https://arduinisti.wordpress.com, rujan 2017.

[4] Arduino UNO, https://robotechshop.com/shop/arduino/arduino-board/arduino-uno-r3-china/?v=fd4c638da5f8 , rujan 2017.

[5] Arduino Uno tehničke karakteristike, https://www.arduino.cc/en/Main/arduinoBoardUno/, rujan 2017.

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

[7] Slika opto-izolator, https://electronics.stackexchange.com/questions/171339/555-as-a-digital-coupler, rujan 2017.

[8] Otpornik, https://en.wikipedia.org/wiki/Resistor, rujan 2017.

[9] Slika otpornika, https://www.dhgate.com/product/wholesale-best-promotion-50-pcs-carbon-film/394936529.html, rujan 2017.

[10] Digitalni otpornik, https://www.sparkfun.com/products/10613, rujan 2017.

[11] Microsoft Visual Studio, https://en.wikipedia.org/wiki/Microsoft_Visual_Studio , rujan 2017.

[12] C#, http://www.etfos.unios.hr/~lukic/oop/Auditorne_vje%C5%BEbe_5.pdf, rujan 2017.

SAŽETAK

Naslov: Mikroupravljački sustav za emulaciju električnih signala u automobilu

U ovom završnom radu napravljen je mikroupravljački sustav za emulaciju signala. Rad je podijeljen u 2 djela: mikroupravljački sklop i njegova Arduino implementacija te izrada C# stolne aplikacije. U radu su opisani korišteni alati i tehnologije te je objašnjena realizacija sustava koja je bila uspješna.

Ključne riječi: električni signal, emulacija signala, Arduino, C#, mikroupravljač

ABSTRACT

Title: Microcontroller system for emulation of electric signals in a car

In this final thesis microcontroller system for emulation of electric signals was developed. The work is divided into 2 main parts: microcontroller and its Arduino implementation and creating a C# desktop application. The paper describes the used tools and technology and it is explained the realization of the system that was successful.

Keywords: electric signal, signal emulation, Arduino, C#, microcontroller

ŽIVOTOPIS

Matej Vuković rođen je 4. studenoga 1994. godine u Đakovu. Od rođenja živi u mjestu Budrovci. Nakon završene osnovne škole u Budrovcima upisuje Gimnaziju A. G. Matoša U Đakovu, prirodoslovno – matematički smjer. Državnu maturu polaže 2013. godine. Iste godine upisuje preddiplomski sveučilišni studij računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku.

PRILOZI

DVD

- Rad u .docx i .pdf formatu
- Arduino programski kod
- C# programski kod