Arduino elektronička brava

Horvat, Goran

Undergraduate thesis / Završni rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University North / Sveučilište Sjever**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:122:128617

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-19**

Repository / Repozitorij:

University North Digital Repository

Sveučilište
Sjever

Završni rad br. 370/EL/2016

Arduino elektronička brava

Goran Horvat, 5024/601

Varaždin, ožujak 2016. godine

Sveučilište Sjever

Odjel za elektrotehniku

Završni rad br. 370/EL/2016

Arduino elektronička brava

Student

Goran Horvat, 5024/601

Mentor

Miroslav Horvatić, dipl. ing

Varaždin, ožujak 2016. godine

Sveučilište Sjever
Sveučilišni centar Varaždin
104. brigade 3, HR–42000 Varaždin

UNIVERSITY NORTH

Prijava završnog rada

Definiranje teme završnog rada i povjerenstva

ODJEL: Odjel za elektrotehniku

PRISTUPNIK: Goran Horvat | MATIČNI BROJ: 5024/601

DATUM: 09.02.2016. | KOLEGIJ: Automatsko upravljanje

NASLOV RADA: Arduino elektronička brava

NASLOV RADA NA ENGL. JEZIKU: Arduino Electronic Door Lock

MENTOR: Miroslav Horvatić, dipl. ing. | ZVANJE: predavač

ČLANOVI POVJERENSTVA:

1. mr. sc. Ivan Šumiga, dipl. ing.
2. Dunja Srpak, dipl. ing.
3. Miroslav Horvatić, dipl. ing.
4.
5.

Zadatak završnog rada

BROJ: 370/EL/2016

OPIS:

Realizirati elektroničku bravu korištenjem Arduino Uno razvojnog sustava. Elektronička brava treba otvoriti vrata nakon upisa četveroznamenkaste brojčane šifre te imati mogućnost promjene šifre.

U radu je potrebno:
- Opisati korišteni Arduino razvojni sustav
- Osmisliti sklopovski i programski dio elektroničke brave
- Realizirati sklopovski i programski dio elektroničke brave
- Realizirati povezivanje elektroničke brave i osobnog računala
- Na osobnom računalu realizirati spremanje podataka o korištenju elektroničke brave
- Opisati korištenje elektroničke brave

ZADATAK URUČEN: 15.02.2016. | POTPIS MENTORA

REPUBLIKA HRVATSKA 5 SVEUČILIŠTE SJEVER KOPRIVNICA SVEUČILIŠTE SJEVER

Predgovor

Zahvaljujem mentoru Miroslavu Horvatiću, dipl. ing, na nesebičnoj pomoći i prenijetom znanju te kontinuiranosti praćenja procesa nastanka ovog završnog rada. Ujedno zahvaljujem za sve dobronamjerne savjete kojima je učinio ovaj rad uspješnim.

Zahvaljujem svim profesorima Sveučilišta Sjever u Varaždinu, koji su svojim trudom i entuzijazmom prenijeli svoje znanje, što je veoma pripomoglo izradi ovog završnog rada.

Također se zahvaljujem svojoj obitelji na neizmjernoj potpori i pružanju mogućnosti obrazovanja na Sveučilištu Sjever u Varaždinu. Zahvaljujem i svima ostalima koji su pomogli na bilo koji način oko obrazovanja i uspješne izrade ovog završnog rada.

Sažetak

Ideja završnog rada je razrada i realizacija elektroničke brave pomoću Arduino UNO razvojne platforme te istovremena komunikacija s računalom uz mogućnost spremanja podataka. Niska cijena, mogućnost nadogradnje i zadovoljavajuće specifikacije su razlozi realizacije završnog rada pomoću Arduino UNO razvojne platforme. Zbog široke dostupnosti Arduino razvojne platforme povećava se broj praktičnih primjena u kojima se ona koristi. Realizacija praktičnog rada popraćena je maketom koja je umanjeni prikaz električne brave.

Maketa se sastoji od upravljačke jedinice (Arduino UNO), LCD zaslona (pokazivač sa tekućim kristalima, 16 znakova x 2 reda), tipkovnice (4 x 4 tipke), servo motora, kristalne zvučnice, tipkala, relejnog modula i mehaničke konstrukcije na koju su montirani elektronički i elektromehanički sklopovi brave. Prilikom prvog pokretanja korisnik može birati između dvije opcije. Opcija 1 je opcija koja omogućava unos PIN-a i otključavanje elektroničke brave. Opcija 2 je opcija pomoću koje se može promijeniti PIN koji otključava elektroničku bravu. Otključavanje vrata popraćeno je zvučnim signalom i uključenjem relejnog modula (signalizacija zelene LED diode), nakon čega se pojavljuje podizbornik koji nudi opcije zaključavanja preko tipke „ * “ ili vraćanja u glavni izbornik uz uvjet da su vrata zatvorena. Pozivanje izbornika vrši se pritiskom na tipku „ # “. Prilikom opcije „promjena PIN-a“ korisnik mora unijeti najprije stari PIN te ako je stari PIN ispravan, omogućava se unos novog PIN-a. Arduino istovremeno vrši komunikaciju preko USB komunikacijskog porta s računalom te se tada podaci spremaju i ispisuju na zaslon osobnog računala.

KLJUČNE RIJEČI: Arduino UNO, LCD zaslon, mikrokontroler, relejni modul, servo motor, tipkalo, tipkovnica

Abstract

The idea is elaboration and realization electronic door lock with Arduino UNO development platform and simultaneously communication with computer along posibility to save data. Low price, possibility for upgrading and satisfactorily specifications are main reason of realization with Arduino UNO development platform. Because of wide availability, number of practical appliance Arduino development platform are increasing. Realization of practical work is accompanied with model which is reduced shew electronic door lock.

Model consist of central unit (Arduino UNO), LCD display (liquid crystal display, 16 chars x 2 rows), keyboard (4x4 push buttons), servo motor, buzzer, push button, reley module and mechanical construction on which are mounted electronic and electromechanic door lock set. On the first run user can choose between 2 options. Option number 1 is the option that allows you to enter a PIN and unlocks the electronic lock. Option number 2 is the option that can be used to change the PIN that unlocks the electronic lock. Unlocking the door is accompanied by an audible signal and the inclusion of the relay module (signaling green LED), it will display a sub-menu that offers options lock by pressing "*" or return to the main menu under the condition that the door is closed. Calling menu is done by pressing the „#“. When option „change PIN“ is selected, user first needs to enter old PIN and if is correct it is possible to enter a new PIN. Arduino simultaneously communicate via USB communication port with computer during which data will be stored and displayed on the screen of personal computer.

KEY WORDS: Arduino UNO, LCD display, microcontroller, reley module, servo motor, push button, keyboard

Popis korištenih kratica

COM (communication port) komunikacijski port

DC (direct current) istosmjerna struja

AC (alternating current) izmjenična struja

RFID (radio-frequency identification) identifikacija preko radio frekvencije

NFC (near field communication) komunikacija kratkog polja

HCE (host card emulation)

P2P (peer to peer) korisnik na korisnik

PIN (personal identification number) osobni identifikacijski broj

PWM (pulse-width modulation) pulsno-širinska modulacija

Vcc (positive supply voltage) pozitivni polaritet napona

GND (ground) „masa“

LCD (liquid crystal display) pokazivač sa tekućim kristalima

EEPROM (electricaly erasable programmable read-only memory) električno izbrisiva programibilna ispisna memorija

USB (universal serial bus) univerzalna serijska sabirnica

LED (light emitting diode) svjetleća dioda

Sadržaj

1. Uvod 6
2. Elektronička brava 8
   2.1. RFID tehnologija 9
   2.2. NFC tehnologija 10
3. Realizacija Arduino elektroničke brave 11
   3.1. Arduino UNO 11
   3.2. Arduino IDE programsko sučelje 12
   3.3. Servo motor 13
   3.4. Relejni modul 14
   3.5. LCD zaslon 16
   3.6. Tipkovnica 17
   3.7. Tipkalo 18
   3.8. Kristalna zvučnica 19
   3.9. Microsoft Visual Studio 2015 20
4. Maketa Arduino elektroničke brave 24
5. Montažna shema elektronike 26
6. Princip rada Arduino elektroničke brave 27
7. Blok dijagram Arduino elektroničke brave 31
8. Elektronička shema spajanja na Atmega328P mikrokontroler 32
9. Zaključak 34
10. Literatura 36
11. Prilozi 37

1. Uvod

Arduino UNO je razvojni sustav koji se sastoji od programskog i sklopovskog dijela. Programski dio Arduino UNO razvojnog sustava sastoji se od programa koji se koriste za prevođenje programskog jezika C u asemblerski (strojni) kod. Sklopovski dio Arduino UNO razvojnog sustava sastoji se od razvojne pločice koja sadrži mikrokontroler ATMEL koji razumije samo asemblerski (strojni) kod. Kada se prevedeni programski kod prenese u memoriju mikrokontrolera, izvršava se tako dugo dok ima napajanje. Prilikom nestanka napajanja ili resetiranja, programski kod počinje iznova. Vrijeme potrebno za obradu pojedinih podataka unutar mikrokontrolera kraće je od milisekunde, što omogućava relativno brzu obradu podataka.

Postoje mnoge mogućnosti nadogradnje ovog sklopa kao npr. povezivanje internetom putem ethernet ili wi-fi shielda. Ulazni elementi ovog sklopa su tipkovnica (4x4) kojom se unose potrebni podaci za ulaza/izlazak ili promjenu PIN-a te tipkalo koje služi kao granični prekidač (položaj vrata otvoreno/zatvoreno). Izlazne elemente čine servo motor čija uloga je zaključavanje/otključavanje vrata, zvučnik kao signalizacija otvoreno/zatvoreno ili PIN promijenjen, lcd zaslon koji nas vodi tekstualno kroz mogućnosti elektroničke brave i relej s tranzistorskim upravljanjem koji osigurava sigurno otključavanje ili zaključavanje brave ako u kojem slučaju servo motor ne proradi. Programiranje pojedinih dijelova i povezivanje u cjelinu vrši se pomoću Arduino IDE programskog sučelja. Arduino IDE programsko sučelje koristi programski jezik C.

Najprije je potrebno nabaviti sve potrebne elemente koji će koristiti u realizaciji samog projekta. Nakon čega se pojedinačno spaja element po element uz istovremeno pisanje koda u Arduino IDE programskom sučelju te njegovo testiranje na eksperimentalnoj pločici. Prilikom završetka spajanja svih potrebnih dijelova te njihovog testiranja, potrebno je izraditi završnu verziju programskog koda gdje svi dijelovi rade sinkronizirano u jednoj cjelini. Po završetku spajanja elemenata i pisanja završne verzije programskog koda koji se implementira u Arduino UNO razvojnu platformu, potrebno je napraviti konačnu električnu shemu spajanja elemenata. Nakon što je kod implementiran i potpuno funkcionalan, potrebno je izraditi maketu na temelju istih elemenata koji su se koristili prilikom testiranja na samoj eksperimentalnoj pločici. Prilikom sastavljanja makete moguće je uočiti eventualne nedostatke koje je potrebno izmijeniti u programskom kodu, nakon čega je potrebno napisati program u C# programskom jeziku koji vrši komunikaciju s Arduino UNO te ih prikazuje na zaslon. Budući da računalo ima više različitih portova bilo je potrebno izraditi mogućnost komunikacije s bilo kojim portom. Podaci koji se ispisuju na zaslonu, odnosno podaci koje Arduino UNO šalje istovremeno se spremaju u tekstualnu

datoteku na računalu ovisno o adresi upisanoj u C# programskom kodu. C# programski kod kao i vizualni izgled pisani su u Microsoft Visual Studio Community 2015 programskom sučelju.

Izrađena maketa je umanjeni prikaz onoga što bi u praksi činilo električnu bravu. Mogućnosti nadogradnje prikazanog rješenja elektroničke brave su velike, kao npr. povezivanje internetom te mogućnost dojave samih događaja putem SMS-a ili s druge strane kreiranjem neke mobilne aplikacije kao sučelje za upravljanje i kontrolu nad samim sustavom. Također je moguća nadogradnja alarmnim i vatrodojavnim sustavom.

Prilikom obrazovanja stečena su mnoga znanja koja su uistinu pripomogla izradi ovo završnog rada, isto tako prilikom izrade završnog rada stečena su nova znanja koja će sa sigurnošću koristiti u daljnjem školovanju ili s druge strane pronalasku posla. Problematika koja se javljala pomogla je usavršavanju snalažljivosti i nadogradnji znanja.

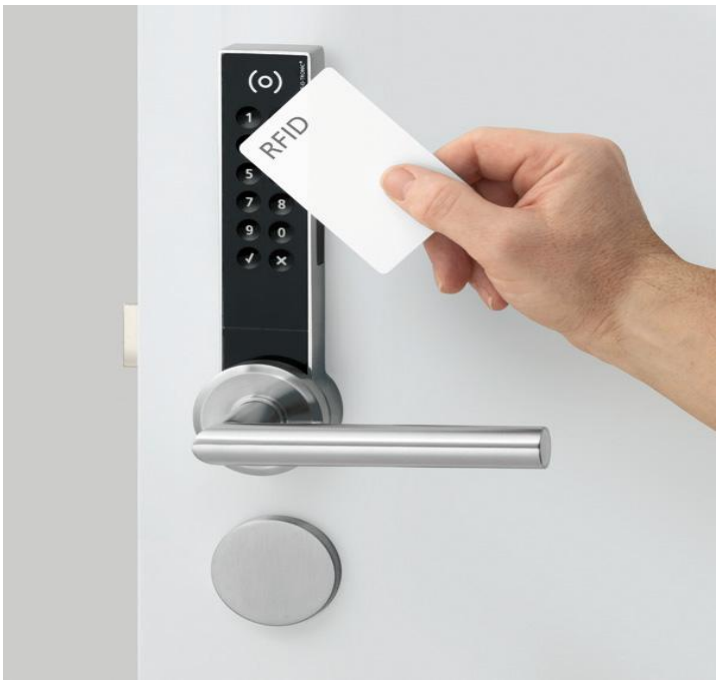
2. Elektronička brava

Elektronička brava je sustav zaštite objekata, sefova, privatnih kuća, stanova itd. Danas to nije samo onaj klasičan način pomoću mehaničkih uređaja već se princip rada zasniva na temelju programskog koda koji je integriran u mikrokontroler. Podaci kao što je PIN ili ostali oblici identifikacije koriste tvornički zadanu vrijednost te prilikom prvog korištenja elektroničke brave korisnik unosi vlastitu vrijednost. Vrijednost koju korisnik unese se pohranjuje u EEPROM memoriju da bi podaci prilikom nestanka napajanja ili resetiranja elektroničke brave ostali očuvani. Otključavanje se izvrši nakon što je identifikacija potvrđena tako da izvršni element pokrene mehanički sklop za otključavanje. Zaključavanje se izvrši ukoliko su vrata zatvorena. Zbog mogućnosti izvršenja pogrešne identifikacije ugrađene su dodatne sigurnosne mjere kao što je trajno zaključivanje. Poznati načini identifikacije današnjice su RFID i NFC tehnologija[3], numerički znakovi[4], sigurnosni tokeni[5], biometrika[6] itd.

Slika 1. Sustav elektroničke brave [1]

2.1. RFID tehnologija

RFID tehnologija (Slika 2) je među novijim tehnologijama koja omogućava jednostavnu identifikaciju putem radio frekvencije. Komunikacija se vrši između host računala te drugih prijenosnih uređaja (kartica koja sadržava programirani čip). RFID tag/labela sadrži čip koji se programira, bateriju te read/write (čitaj/piši) sustav. Tagovi se izrađuju prema potrebama primjene, a koriste se i kod elektroničkih brava.



Slika 2. RFID tehnologija identifikacije [2]

RFID Frequency Band	Scan Distance
120-150 kHz (Low Frequency, LF)	Up to 10 cm
13.56 MHz (High Frequency, HF)	Up to 1 m
433 MHz (Ultra High Frequency, UHF)	1-100 m
865-868 MHz & 902-928 MHz (Ultralight High Frequency, UHF)	1-2 m
2450-5800 MHz (Microwave)	1-2 m
3.1-10 GHz (Microwave)	Up to 200 m

Slika 3. Specifikacije RFID tehnologije [3]

2.2. NFC tehnologija

NFC tehnologija je dodatak temeljen na RFID standardima. Jedan je od principa identifikacije putem radio frekvencije. Kratkog je dometa, a koristi elektromagnetsku indukciju između predajnika i prijamnika kako bi došlo do razmjene podataka (identifikacije). NFC tehnologija dijeli neke fizičke osobine s RFID tehnologijom, a jedna od osobina je komunikacija bez direktnog pristupa. Postoje tri ključne razlike [3]:

- NFC je za složenije interakcije (HCE i P2P dijeljenje) zbog mogućnosti dvosmjerne komunikacije
- NFC ima ograničenu daljinu komunikacije, 5cm ili manje
- samo jedan NFC tag je moguće skenirati istovremeno

	HF RFID	NFC
Operating Frequency	13.56 MHz	13.56 MHz
Communication	One way	Two way
Standards	ISO 14443, 15693, 18000	ISO 14443
Scan Distance	Up to 1 m	Up to 10 cm
Scan Tags Simultaneously	Yes	No

Slika 4. Razlike između RFID i NFC tehnologije [3]

NFC tehnologija je postojana isključivo zbog sigurnosnih plaćanja putem mobilnih uređaja ili PayPass kartica („prisloni i plati“ kartica), a također ima veliku primjenu kod sigurnosnih sustava poput identifikacije kod elektroničke brave [3].

3. Realizacija Arduino elektroničke brave

Realizacija ovog završnog rada bio je skup zadataka koji su se spojili u cjelinu. Potrebna je realizacija sklopovske sheme, zatim programskog koda i po završetku povezivanje s računalom preko USB komunikacijskog porta te ispis svih podataka na zaslon osobnog računala uz istovremenu pohranu. Arduino elektronička brava mora imati mogućnosti unosa PIN-a i prikaza poruka te promjene PIN-a.

3.1. Arduino UNO

Arduino UNO (Slika 5) (prilog 1) je razvojna platforma bazirana na Atmelovom mikrokontroleru. Atmelov mikrokontroler Atmega328P je upravljačka jedinica ovog završnog rada, a u njemu se nalazi programski kod kojeg neprestano izvršava.

Mikrokontroler	ATmega328P
Radni napon	5V
Ulazni napon (preporučeno)	7-12V
Ulazni napon (granica)	6-20V
Digitalni I/U pinovi	14 (od kojih su 6 PWM izlazi)
PWM digitalni I/U pinovi	6
Analogni ulazni pinovi	6
DC struja po I/U pinu	20 mA
DC struja za 3.3V pin	50 mA
Flash memorija	32 KB (ATmega328P) od koje 0.5 KB koristi bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Brzina rada	16 MHz

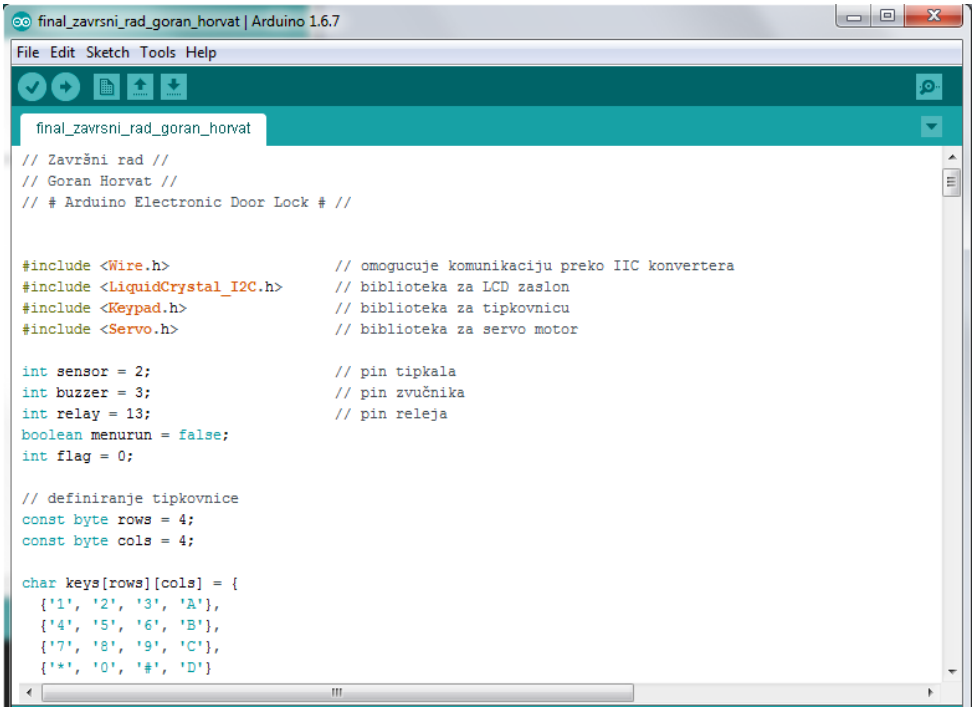
Tablela 1. Specifikacije Arduino UNO razvojne platforme [7]

Mikrokontroler radi na naponskoj razini od 5V pa treba paziti prilikom spajanja vanjskih elemenata da naponska razina ne prelazi radnu naponsku razinu, kako bi se izbjegla oštećenja. Programiranje mikrokontrolera vrši se u Arduino IDE programskom sučelju, pa se upravljački program u mikrokontroler učitava putem USB komunikacijskog porta [7].

Slika 5. Arduino UNO razvojna platforma [8]

3.2. Arduino IDE programsko sučelje

Nakon što je realiziran elektronički sklop, realizacija sheme elemenata te spajanje istih, potrebno je realizirati programski kod. Programski kod je pisan u Arduino IDE programskom sučelju koje koristi programski jezik C. Najprije se trebalo upoznati s pojedinim elementima elektroničkog sklopa te na temelju toga napisati funkcije pojedinih elemenata koje se kasnije spajaju u cjelinu.



Slika 6. Arduino IDE programsko sučelje

Prilikom pisanja programskog koda potrebno je najprije uključiti knjižnice koje se koriste za pojedini element. Unutar knjižnice se nalaze prethodno definiran način rada pojedinog elementa te se koriste radi lakšeg pisanja koda. Svaka knjižnica sadržava ključne riječi koje se koriste prilikom definiranja kao npr. `Servo myservo;`. `Servo` je ključna riječ iz knjižnice `<Servo.h>` koja poziva funkciju za definiciju, a `myservo` je naziv servo motora koji je definiran.

3.3. Servo motor

Servo motor TG9e radi na 5V DC, ima moment od 0.15Nm, malih je dimenzija (23x12.2x29mm) te je korišten kao glavni element za otključavanje/zaključavanje sustava elektroničke brave. Korištenje pripremljene knjižnice `#include <Servo.h>` u Arduino IDE programskom sučelju omogućava laku primjenu. Definiranje servo motora vrši se naredbom `Servo myservo;`. Nakon što se servo motor definira potrebno mu je dodijeliti njegov upravljački pin. Pin mora sadržavati PWM regulaciju zbog slanja više uzastopnih impulsa prilikom promjene položaja, a definiran je u `Servo.attach(12);`. Broj 12 označava pin na koji je spojena upravljačka žica servo motora. Promjena položaja obavlja se korištenjem for petlje:

```
for(int y = 0; y <= 90; y++){    //definicija for petlje
        myservo.write(y);  // položaj ovisan o y(broj zakreta)
        delay(3);          // pauza između svakog koraka
  }
```

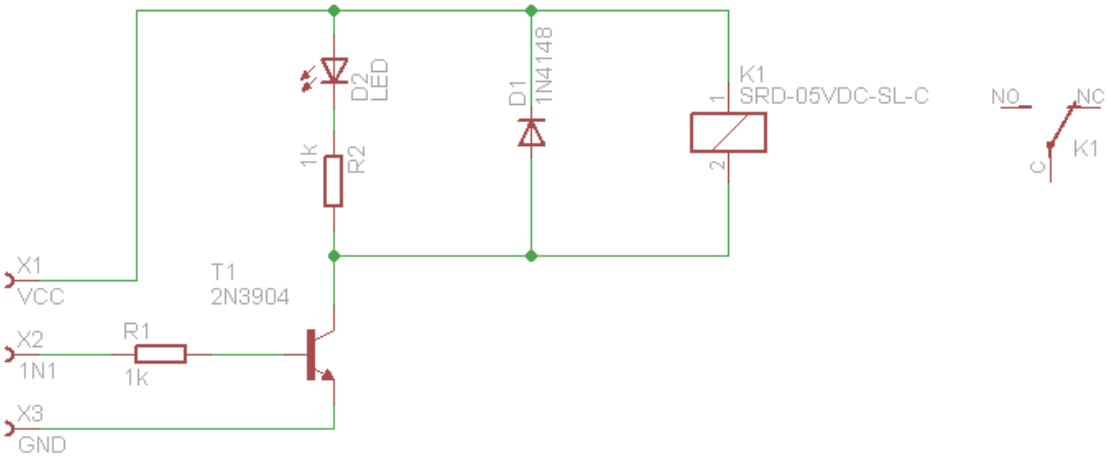
, zbog toga jer servo motoru treba određeno vrijeme da izvrši pojedini zakret. Prema specifikacijama ovog servo motora to vrijeme iznosi 2 milisekunde, ali zbog sigurnog zakreta postavljeno je na 3 milisekunde. Da bi se izbjegli čudni zvukovi i vibracije motora, odnosno dobilo konstantno napajanje, u paralelu između VCC i GND dodan je kondenzator od 470 µF (prilog 2).

Slika 7. Servo motor TG9e [9]

3.4. Relejni modul

Da bi se osiguralo sigurnosno otključavanje/zaključavanje u slučaju da servo motor ne proradi ili zbog nekih drugih razloga, u sustav je dodan relejni modul koji ima namjenu privući odnosno otpustiti kotvu. Relejni modul se sastoji od tranzistorske sklopke (NPN) i releja te ostalih potrebnih pasivnih elemenata. Primarni (upravljački) strujni krug radi na 5V DC, a sekundarni (izvršni) strujni krug može podnijeti i do 220V AC. Upravljanje se vrši putem jednog od digitalnih pinova. Definicija u programskom kodu je jednostavna te nije potrebna nikakva knjižnica. Najprije se definira pin na koji će se spojiti upravljačka žica relejnog modula `int relay = 13;`, a zatim se definira da je izlazni pin naredbom `pinMode(relay, OUTPUT);`.

Slika 8. Relejni modul



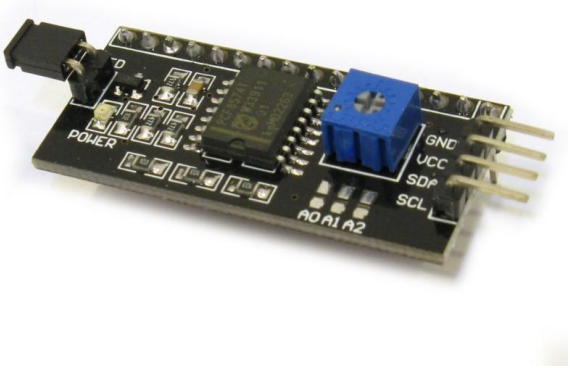
Slika 9. Shema relejnog modula

3.5. LCD zaslon

Radi lakšeg snalaženja te ispisivanja trenutnih stanja sustava elektroničke brave koristi se LCD zaslon 16x2 (Slika 10), što znači 16 znakova u svakom redu sa maksimalno 2 reda. Radni napon mu je 5V DC te sadrži 2 registra: Command i Data. Command registar pohranjuje isključivo naredbe koje su mu date kao npr. brisanje trenutnog sadržaja na zaslonu, postavljanje kursora, kontrola zaslona itd. Dok s druge strane imamo Data registar koji pohranjuje podatke koji se moraju prikazati na LCD zaslonu. Podatak je u ASCII kodu što znači da se podaci šalju u 8 bitnoj kombinaciji te je potrebno 8 digitalnih pinova. Budući da imamo ograničen broj pinova radi lakšeg korištenja, LCD zaslon je povezan na I2C pretvarač (Slika 11). I2C pretvarač se koristi zbog jednostavnosti primjene unutar programskog koda i broja pinova potrebnih za komunikaciju. U programskom kodu koji upravlja mikrokontrolerom potrebno je uključiti knjižnicu za rad sa I2C pretvaračem, što je realizirano korištenjem naredbe `#include <LiquidCrystal_I2C.h>`, a definicija trenutnog LCD zaslona naredbom `LiquidCrystal_I2C lcd(0x27, 16, 2);`. Ostale naredbe definirane su unutar funkcija.

Slika 10. LCD 16x2 zaslon [10]

I2C pretvarač sadrži 4 ulazna pina od kojih su dva napajanje (VCC i GND), a preostala dva podatkovna (SDA i SCL) te 16 izlaznih pinova od kojih se 8 koriste za komunikaciju s LCD zaslonom. Ovisno o brzini rada procesora SCL mijenja svoja logička stanja (0,1). SDA prelazi iz logičke 0 u logičku 1 kada SCL poprimi vrijednost logičke 1, a prijenos podatkovnog bita s Arduinom započinje kada su SDA i SCL u stanju logičke 1. Za svaki prelazak SDA iz logičke 0 u logičku 1, a da je pritom SCL u stanju logičke 1 šalje se po 1 podatkovni bit.



Slika 11. IIC konverter [11]

3.6. Tipkovnica

Unos PIN-a ili pak navođenje kroz izbornike elektroničke brave vrši se putem tipkovnice 4x4 (Slika 12). Tipkovnica sadrži 16 tipkala te se trenutna stanja pojedine tipke prenose putem 8-žične komunikacije kod koje prva 4 bita služe za redove, a preostala 4 bita za stupce. Pozivanje knjižnice za rad sa tipkovnicom vrši se naredbom `#include <Keypad.h>`, a samo definiranje:

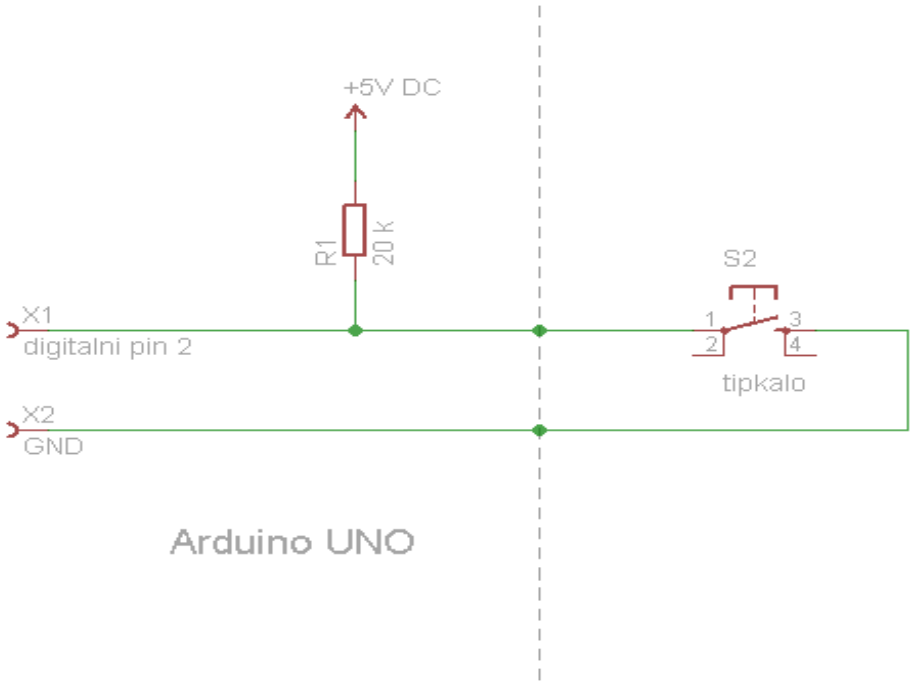
```
const byte rows = 4;              // broj redaka tipkovnice
const byte cols = 4;              // broj stupca tipkovnice

char keys[rows][cols] = {     // definiranje matrice tipki
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};

byte rowPins[rows] = {11, 10, 9, 8};  // pinovi redova
byte colPins[cols] = {7, 6, 5, 4};    // pinovi stupaca
```

Kreiranje nove tipkovnice vrši se naredbom `Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, rows, cols);`.

Slika 12. Tipkovnica 4x4 [12]

3.7. Tipkalo

Tipkalo služi kao granični prekidač u ovom završnom radu. Korištenjem tipkala realizirana je detekcija položaja vrata (otvoreno/zatvoreno) (Slika 13). Definiranje pina izvršeno je naredbama `int sensor = 2;`, `pinMode(sensor, INPUT_PULLUP);`. Čitanje stanja vrši se naredbom `digitalRead(sensor);`. Pošto je ulaz definiran putem „PULLUP-a“ (pin je preko otpornika spojen na +5V DC) onda vrijedi obrnuta logika. Kada na ulaz dođe logička jedinica, mikrokontroler je čita kao logičku nulu.



Slika 13. Električna shema - spoj tipkala

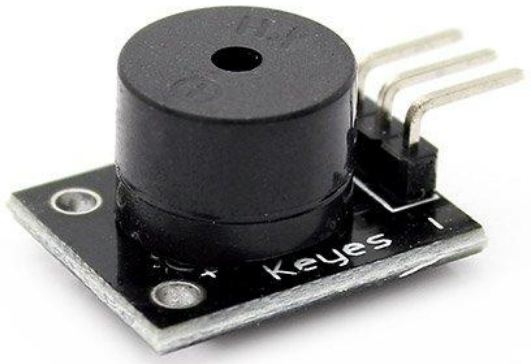
Svaki od 14 digitalnih pinova na Arduino UNO razvojni platformi može se spajati na „PULLUP“ otpornik, a iznos otpornika je od 20 - 50 kOhm-a. Po zadanim postavkama taj otpornik nije spojen, a definicija spoja vrši se unutar programskog koda sintaksom `pinMode(sensor, INPUT_PULLUP);`.

3.8. Kristalna zvučnica

Svaka izvršena radnja signalizirana je zvučnim signalom radi lakšeg prepoznavanja promjene stanja. Kristalna zvučnica Keyes radi na 5V DC frekvencije zvuka 2-5 kHz. Definiranje se vrši naredbom `int buzzer = 3;`, a naredba za postavljanje izlaznog pina `pinMode(buzzer, OUTPUT);`. Duljina zvučnog signala definirana je unutar funkcije korištenjem for petlje

```
void ring_bell() {                    // funkcija zvuka
   for (int x = 0; x <= 150; x++) {   // definicija for petlje
        digitalWrite(buzzer, HIGH);   // stanje zvučnika - ON
        delay(1);                     // frekvencija zvuka
        digitalWrite(buzzer, LOW);    // stanje zvučnika – OFF
        delay(1);
   }
}
```

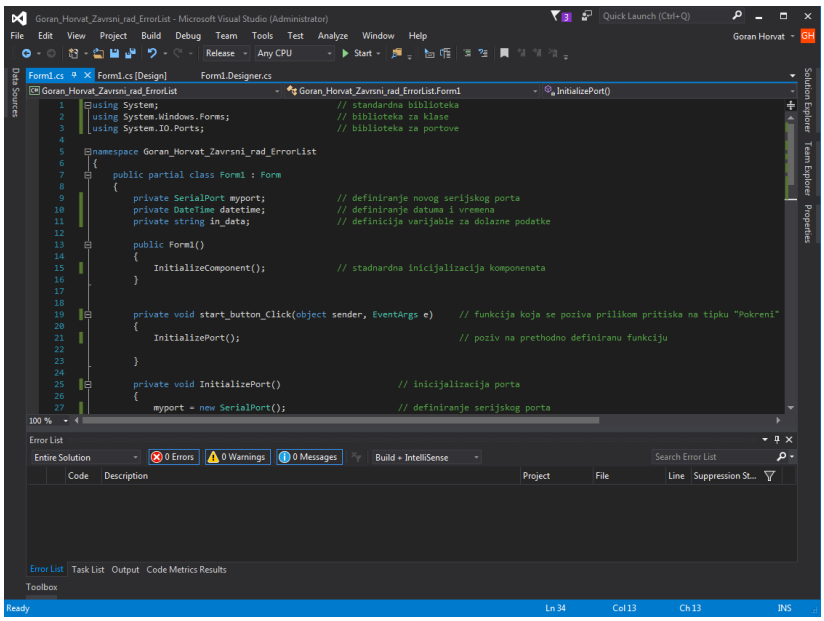
Funkcija naizmjenično postavlja vrijednosti izlaznog pina kristalne zvučnice iz logičke 1 u logičku 0 i obrnuto. Vremenska pauza između promjene stanja definira frekvenciju zvuka, dok broj unutar definicije petlje definira dužinu zvuka odnosno broj ponavljanja petlje.



Slika 14. Kristalna zvučnica Keyes [13]

3.9. Microsoft Visual Studio 2015

Kreiranje aplikacije koja omogućava ispis podataka na zaslon te istovremenu pohranu u tekst (.txt) datoteku izvršeno je u Microsoft Visual Studio 2015 [14] (Slika 15). Najprije je kreiran dizajn, postavljene su tipke i određeni tekst blokovi (Slika 16) koji se kasnije programiraju programskim jezikom C# te spajaju u cjelinu.

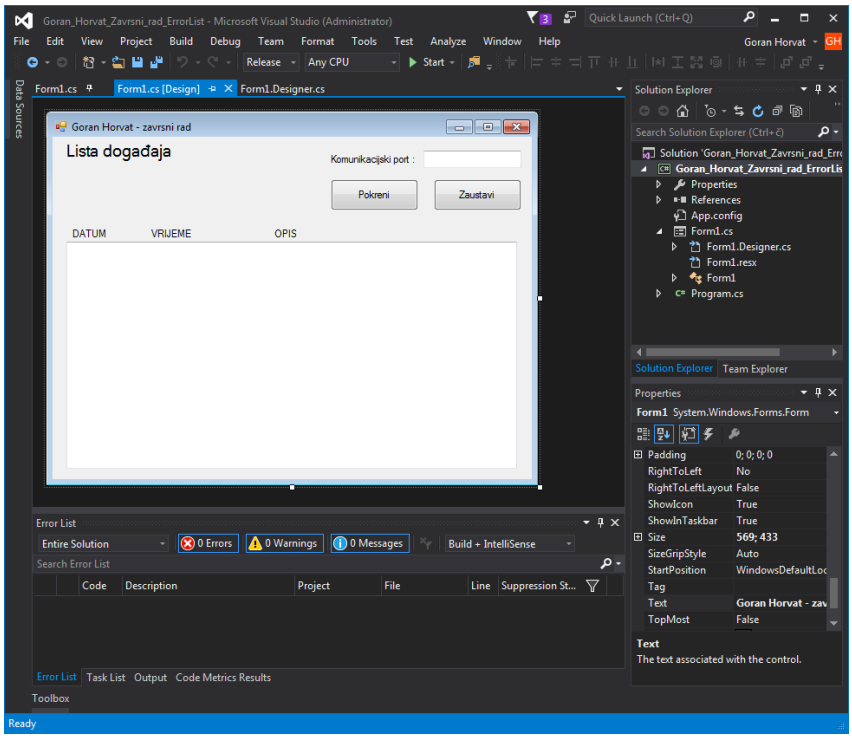


Slika 15. Visual Studio 2015 - kod

Najbitnije je dobro inicijalizirati port koji dohvaća podatke koje Arduino šalje, a inicijalizacija se vrši na ovaj način:

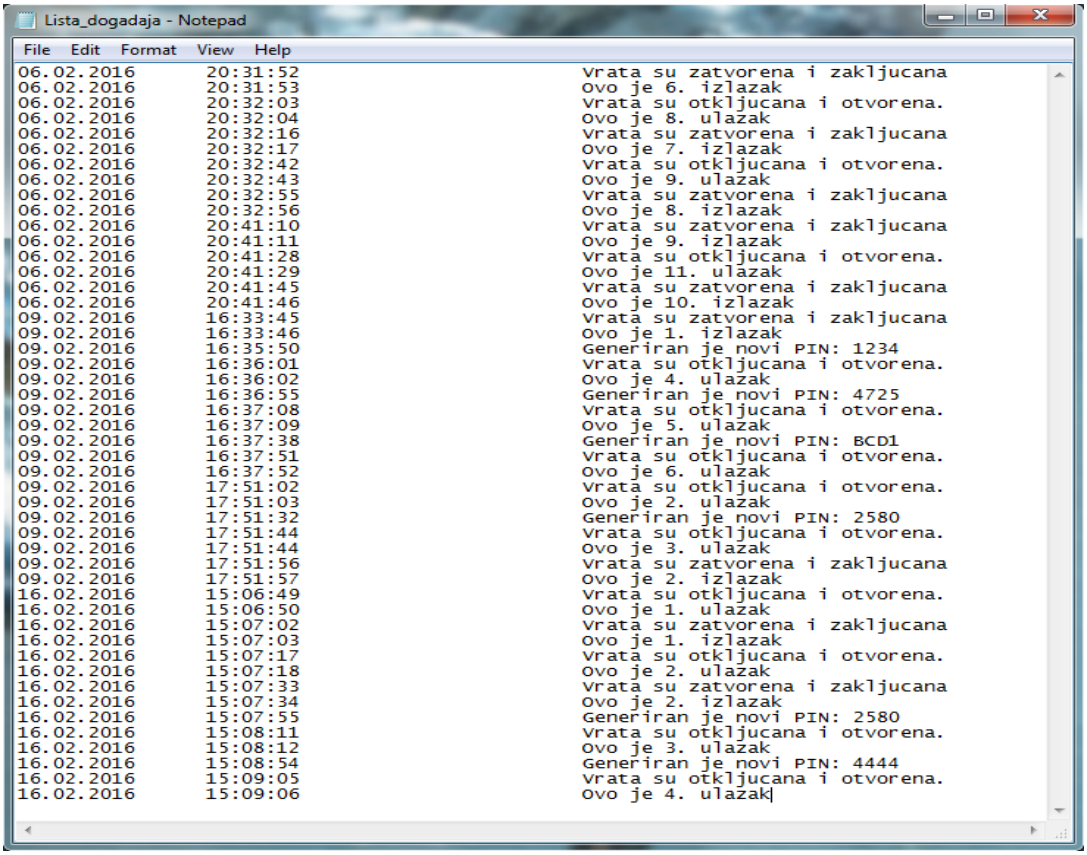
```
private void InitializePort(){      // inicijalizacija porta
    myport = new SerialPort();  // definiranje serijskog
    porta
    myport.BaudRate = 9600;     // definiranje brzine
    prijenosa podataka
    myport.PortName = port_tab.Text;  // definiranje
    prozora za komunikacijski port
    myport.Parity = Parity.None;   // definiranje   pariti
    bita, u ovom slučaju nije potrebno, ignorira se
    myport.DataBits = 8;                                //
    veličina dolaznog podatka definira se na 8 bitova
    myport.StopBits = StopBits.One;       // definiranje broja
    bitova za prekid podatkovne veze
    myport.DataReceived += myport_DataReceived;  // definiranje
    dohvata podataka

    try{
        myport.Open();          // otvaranje porta
        data_tab.Text = "";     // varijabla za tekst koji će se
        pojaviti (""- zadano)
    }
    catch (Exception ex){  // u slučaju greške ispisuje se
    error (npr. ako ne može dohvatiti podatke)
            MessageBox.Show(ex.Message, "Error");
    }
}
```

Definiran je serijski port pod nazivom “myport” kojem je dodijeljena brzina prijenosa podataka serijske komunikacije. Za odabir porta kreirana je labela gdje korisnik unosi naziv porta. Uneseni naziv porta osigurava komunikaciju s Arduino UNO razvojnom platformom. USB komunikacijski port ne koristi pariti bit pa je taj bit definiran kao nepostojeći. Veličina dolaznog podatka je 8 bitna, dok se jedan od bitova koristi za raskid komunikacije. Ukoliko prilikom inicijalizacije porta dođe do pogreške pojavljuje se prozor s uzrokom iste.



Slika 16. Visual Studio 2015 – dizajn

Na početku kreiranja vizualnog izgleda aplikacije postojao je samo prozor s zadanim izgledom. Razvijanje dizajna odvijalo se dodavanjem teksta, tekstualnih prozora i tipki. Funkcija tekstualnih prozora definirana je unutar programskog koda ovisno o tome dali služi za upis porta ili ispis teksta. Definiran je događaj na temelju kojeg se vrši ispisivanje podataka iz tekstualnog prozora u tekstualnu datoteku uz uvjet da se postojani tekst u datoteci ne briše. Datum i vrijeme su definirani radi lakšeg praćenja događaja koji su se dogodili u prošlosti. Tipka „Pokreni“ ili „Enter“ definira početak inicijalizacije porta koja je ranije objašnjena. Tipka „Zaustavi“ prekida serijsku komunikaciju.

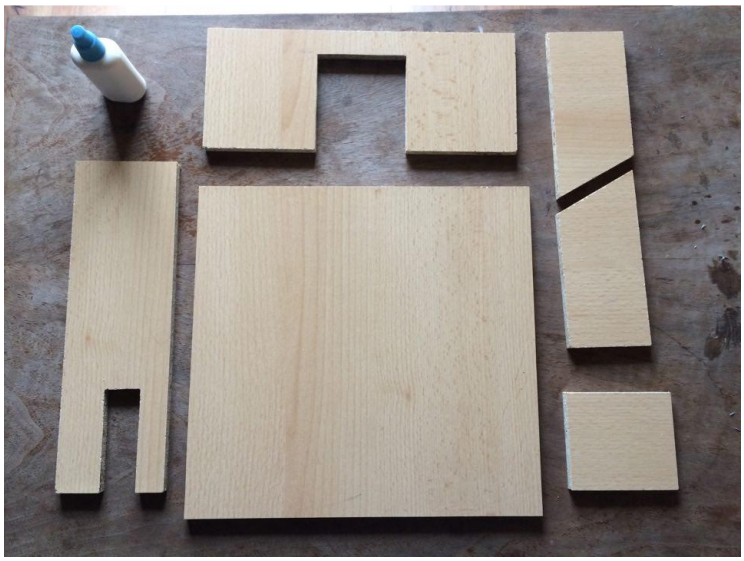


Slika 17. Pohranjeni podaci u tekst datoteci

Ispis podataka u tekst datoteku definiran je unutar programskog koda. Naknadna promjena adrese ispisa podataka nije moguća. Radi lakšeg praćenja prošlosti korištenja sustava Arduino elektroničke brave tekst datoteka sadržava datum, vrijeme i opis događaja. Ispravnost datuma i vremena ovisi o postavkama istih na osobnom računalu. Opis događaja sadrži informacije o stanju elektroničke brave koje Arduino UNO šalje na osobno računalo putem USB komunikacijskog porta.

4. Maketa Arduino elektroničke brave

Realizirana je maketa Arduino elektroničke brave. Dijelovi mehaničke konstrukcije makete prikazani su na slici 18. Maketa je kreirana od prije spomenutih elemenata te konstrukcije. Materijal konstrukcije je drvo pod nazivom „furnirana iverica“. Najprije je bilo potrebno napraviti nacrt po kojem će se maketa raditi. Nakon realizacije nacrta, iz drveta su izrezani dijelovi koji su se kasnije sastavili (Slika 19).



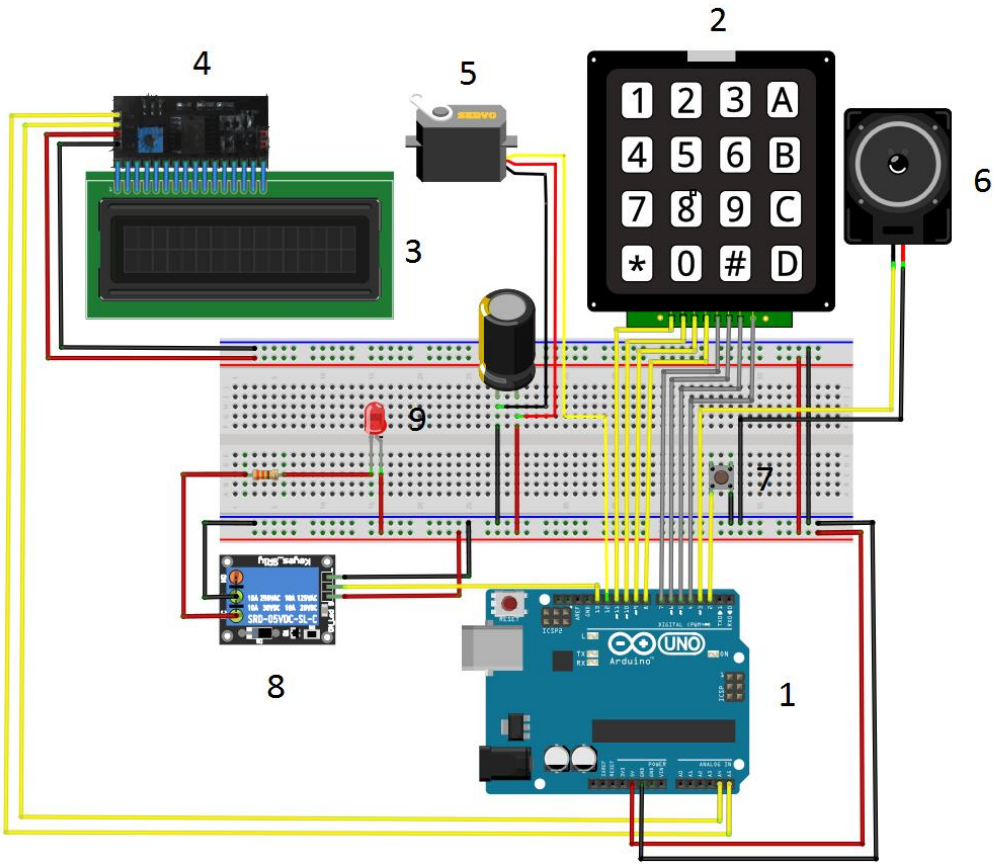
Slika 19. Izrezani dijelovi konstrukcije makete

Kada se konstrukcija sastavila, slijedilo je implementiranje elektronike. Nakon uspješne implementacije elektronike, testiralo se ponašanje sustava.

Slika 19. Slika gotove makete Arduino elektroničke brave

5. Montažna shema elektronike

Montažna shema izrađena je u programskom sučelju Fritzing [15]. Fritzing sadrži sve potrebne elemente iz područja elektronike te elemente vezane uz Arduino razvojnu platformu.



Slika 20. Montažna shema Arduino elektroničke brave

U mikrokontroleru Arduino UNO razvojne platforme (1) nalazi se prethodno učitan programski kod koji obavlja zadane funkcije. Ispis podataka vrši se korištenjem 2-žične komunikacije s I2C pretvaračem (4) koji je spojen na analogne pinove Arduino UNO razvojne platforme. LCD zaslon (3) je spojen na izlaze I2C pretvarača s kojim se vrši komunikacija. Navođenje kroz izbornike obavlja se putem tipkovnice (2) koja je spojena sa 8 žica na digitalne pinove Arduino UNO razvojnu platformu. Za otključavanje odnosno zaključavanje koriste se servo motor (5) i relejni modul (8), a spojeni su na digitalne pinove Arduino UNO razvojne platforme. Tipkalo (7) spojeno na digitalni pin Arduino UNO razvojne platforme detektira položaja vrata. LED dioda (9) i kristalna zvučnica (6) signaliziraju promjenu.

6. Princip rada Arduino elektroničke brave

Prilikom prvog pokretanja vrši se inicijalizacija, postavljanje početnih stanja vrijednosti određene varijable. LCD zaslon ispisuje poruku, a nakon toga prikazuje glavni izbornik.

Slika 21. Autorska poruka

Slika 22. Glavni izbornik

Glavni izbornik se sastoji od dva podizbornika. Prvi korak omogućuje korisniku ulazak, dok drugi korak korisniku nudi promjenu PIN-a. Prilikom odabira pod „1“ korisniku se prikazuje poruka na LCD zaslonu kao na sljedećoj slici.

Slika 23. Poruka za unos PIN-a

Korisnik unosi četveroznamenkasti PIN. Ukoliko PIN nije točan slijedi poruka na LCD zaslonu te upućuje korisnika na ponovni unos PIN-a.

Slika 24. Poruka o pogrešnom PIN-u

Slika 25. Poruka za unos PIN-a

Ukoliko je PIN točan, prikazuje se slijedeći slijed poruka na LCD zaslonu. Prilikom prikazivanja poruka izvršni elementi (servo motor i relejni modul) promjene svoj položaj, a ta je promjena potvrđena zvučnim i svjetlosnim signalom.

Slika 26. Poruka o otključavanju vrata

Slika 27. Poruka „Vrata su otključana!“

Nakon što se vrata otključaju i otvore, korisniku se nude dvije opcije. Prva opcija služi za zaključavanje vrata, a druga opcija je za ulazak u glavni izbornik.

Slika 28. Podizbornik

Prilikom pritiska na znak „ * “ uz ispunjen uvjet, vrata se zaključavaju. Prikazuje se poruka na LCD zaslonu da se vrata zaključavaju te u međuvremenu izvršni elementi promjene svoj položaj što je potvrđeno zvučnim i svjetlosnim signalom.

Slika 29. Poruka o zaključavanju vrata

Slika 30. Poruka „Vrata su zaključana!“

Prilikom pritiska na znak „ # “ uz ispunjen uvjet, vrata se zaključavaju kao u slučaju ranije, bez ispisa poruka na zaslon LCD-a. Izvršni elementi promjene položaj te se prikazuje sljedeća poruka na zaslonu LCD-a.

Slika 31. Glavni izbornik

U oba slučaja je uvjet da su vrata zatvorena (tipkalo pritisnuto). Ako vrata nisu zatvorena pojavljuje se poruka na LCD zaslonu te kada se vrata zatvore (uvjet je ispunjen), automatski se zaključavaju.

Slika 32. Poruka o stanju vrata

Poruka na slici 32 ukazuje da vrata nisu zatvorena. Ukoliko se vrata zatvore, tipkalo šalje potvrdni signal mikrokontroleru te se nastavlja proces zaključavanja elektroničke brave.

Drugi korak glavnog izbornika omogućuje korisniku promjenu PIN-a. Prilikom promjene PIN-a korisnik mora znati i unijeti trenutni (stari) PIN.

Slika 33. Poruka za unos starog PIN-a

Ukoliko korisnik unese pogrešan stari PIN, pojavljuje se poruka na LCD zaslonu te se traži ponovni upis starog PIN-a.

Slika 34. Poruka o pogrešnom PIN-u

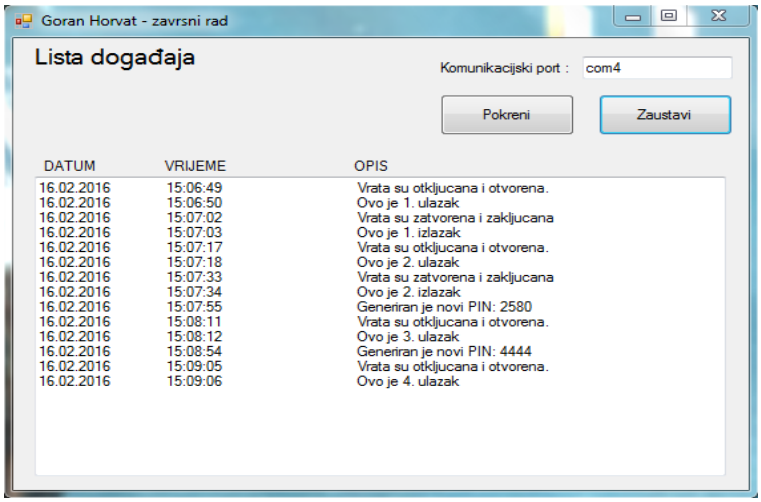
Slik 35. Poruka za unos starog PIN-a

Ukoliko korisnik unese točan stari PIN, nudi mu se unos novog PIN-a. Kada je unos novog PIN-a gotov pojavljuje se poruka i zvučni signal.

Slika 36. Poruka da je PIN promijenjen

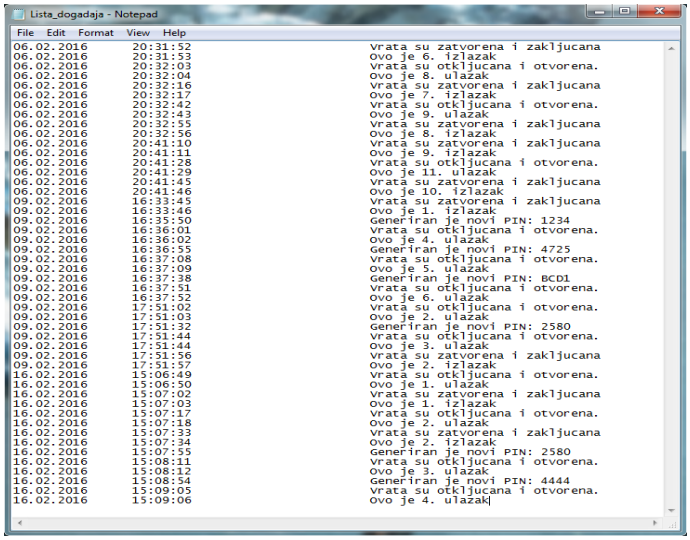
Nakon uspješne promjene PIN-a, korisniku se omogućuje otključavanje putem novog PIN-a. Stari PIN više ne vrijedi te nije moguće otključavanje ako se unese.

Arduino istovremeno vrši komunikaciju s računalom putem USB komunikacijskog porta. Potrebno je unijeti u labelu „Komunikacijski port:“ ime porta s kojim se izvršava komunikacija (npr. COM4). Nakon što se unese naziv porta, pritiskom na tipku „Pokreni“ ili „Enter“ pokreće se komunikacija između Arduina i računala. Poruka koja se ispisuje za svaku promjenu sadrži datum, vrijeme i opis događaja. Prekid komunikacije vrši se pritiskom na tipku „Zaustavi“ ili izlazom iz aplikacije (slika 37).



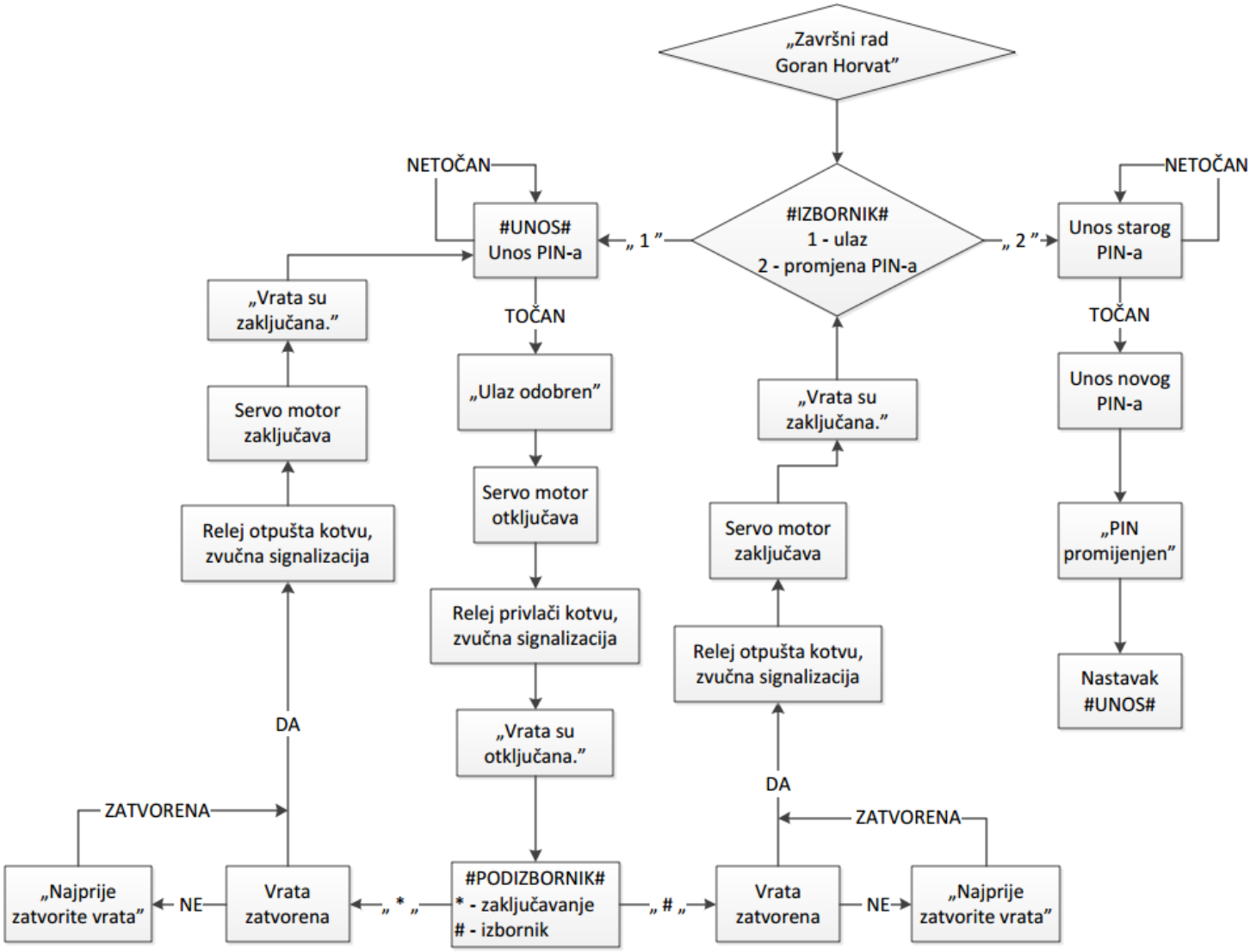
Slika 37. Lista događaja - aplikacija

Podaci koji se prikazuju na zaslonu aplikacije imaju istovremenu pohranu u tekst datoteku. Adresa datoteke je prethodno definirana unutar programskog koda te nije moguća izmjena adrese pohrane podataka. Tekst datoteka sadržava iste podatke koji se ispisuju na zaslon aplikacije uz povijest ispisanih događaja.



Slika 38. Lista događaja - tekst datoteka

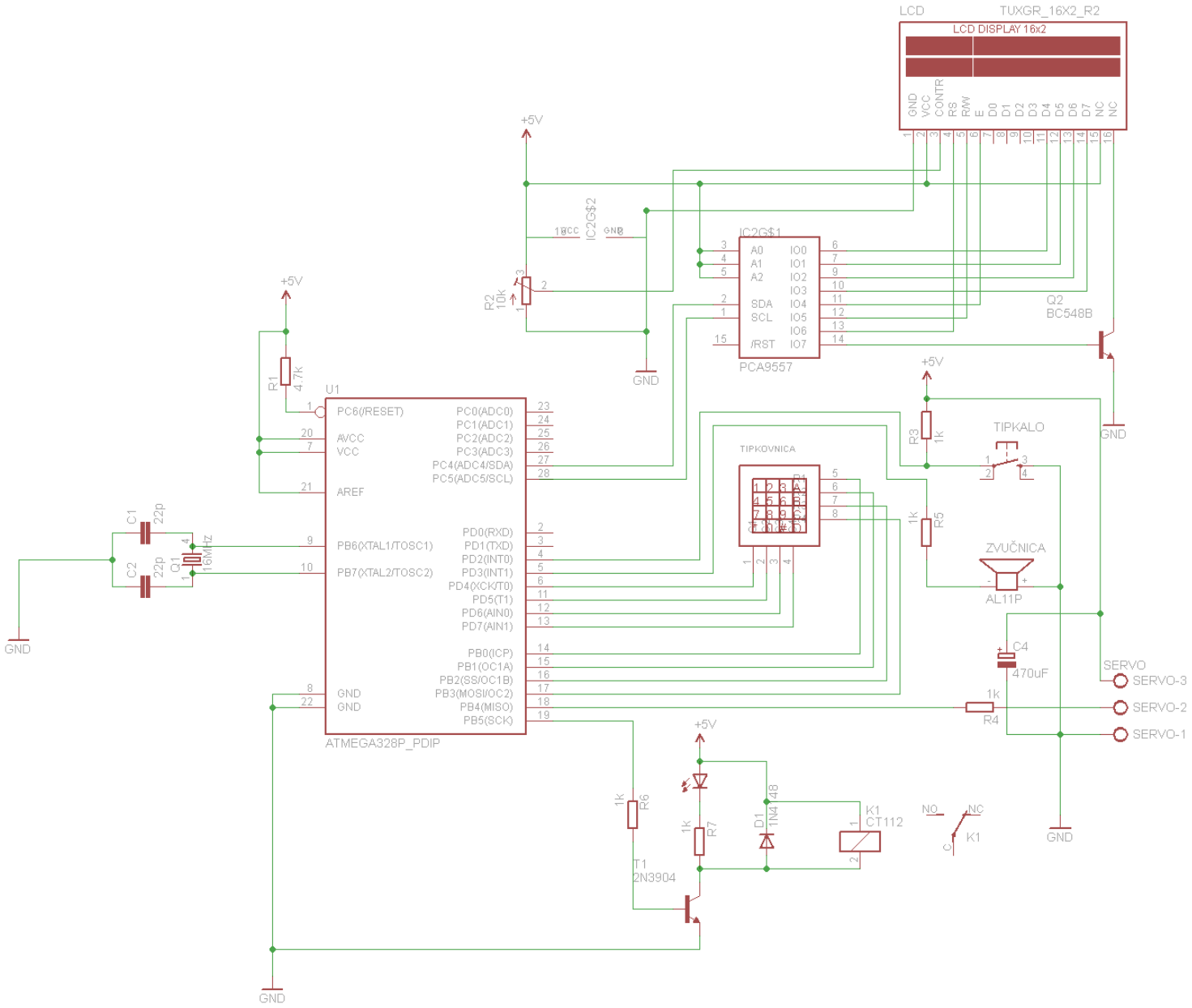
7. Blok dijagram Arduino elektroničke brave



Slika 39. Dijagram toka Arduino elektroničke brave

Kod prvog pokretanja vrši se inicijalizacija početnih stanja te uslijedi prikaz izbornika koji korisniku nudi dvije opcije. Odabir prve opcije, opcije za otključavanje brave od korisnika traži unos PIN-a. Ako je PIN točan izvršni elementi otključaju bravu što omogućava ulaz. Nakon ulaska pojavljuje se podizbornik koji također nudi dvije opcije. Opcija „*“ služi za zaključavanje brave, a opcija „#“ služi za ulazak u glavni izbornik. U oba slučaja je uvjet da su vrata zatvorena, a ako nisu neće doći do sljedećeg koraka tako dugo dok se ne zatvore. Druga opcija glavnog izbornika je opcija promjene PIN-a, a njenim odabirom se od korisnika najprije traži unos starog PIN-a te ako je točan onda unos novog PIN-a. Ukoliko je izvršena i potvrđena promjena PIN-a omogućuje se unos PIN-a i otključavanje brave.

8. Elektronička shema spajanja na Atmega328P mikrokontroler



Slika 40. Shema povezivanja ATmega328P mikrokontrolera s elektroničkim elementima

Ukoliko bi bilo potrebno realizirati maketu kao gotov proizvod, shema povezivanja mikrokontrolera ATmega328P s elektroničkim elementima izgledala bi kao na slici 40. Komunikacija između mikrokontrolera i I2C pretvarača koji je ranije objašnjen vrši se slanjem 8 bitnog signala kojeg I2C pretvarač obrađuje i šalje na zaslon LCD-a. Kontrolirani rad servo motora vrši se PWM regulacijom, tj. Arduino šalje uzastopnu promjenu digitalnog signala (0/1)

na upravljački pin servo motora. Brzina slanja impulsa određena je pauzama unutar for petlje koja ja ranije objašnjena. Sigurnosno otključavanje/zaključavanje vrši se slanjem digitalnog signala na upravljački pin relejnog modula. Tipkovnica komunicira s Arduinom slanjem digitalnog podatka duljine 8 bita, 4 bita za redove i 4 bita za stupce. Ovisno o kombinaciji logičkih 0 i 1 Arduino raspoznaje koja tipka je pritisnuta. Tipkalo komunicira s Arduinom preko digitalnih impulsa koji ovise o stanju preklopke. Ugradnjom „pullup“ otpornika realizirana je obrnuta logika. Ukoliko bi se realiziralo bez „pullup-a“ (klasični spoj elementa) mikrokontroler nebi reagirao na stanje logičke nule. Radom zvučnice upravlja se na istom principu kao radom servo motora.

9. Zaključak

Sustav elektroničke brave je sve prisutniji iz razloga osiguranja imovine. Elektronička brava korisniku nudi razne mogućnosti koje su ranije bile opisane, a jedna od mogućnosti je naknadna nadogradnja.

Arduino UNO razvojna platforma je cijenom prihvatljiva i pruža mnoge mogućnosti. U slučaju da nam specifikacije Arduina UNO nisu odgovarajuće, postoje i novije verzije razvojne platforme Arduina koje rade sa većim brojem pinova. Programiranje rada mikrokontrolera se vrši u besplatnom programskom sučelju Arduino IDE koje nam omogućuje implementaciju raznih knjižnica. Knjižnice se mogu pronaći bilo gdje na internetu, a također postoje i razni forumi gdje ljudi dijele svoje knjižnice i radove uz koje je moguće puno naučiti. Ključne riječi knjižnice su jednostavne, a programski kod je baziran na programskom jeziku C. Vrijedno je napomenuti da Arduino razvojna platforma i sklopovi koji se realiziraju za istu u Hrvatskoj imaju sve veću primjenu.

Nakon nabave svih potrebnih elektroničkih elemenata za realizaciju Arduino elektroničke brave te realiziranja elektroničke sheme bilo je potrebno spajanje istih na testnoj pločici. Kada je sve bilo spojeno i provjereno slijedila je realizacija programskog koda. Na temelju korištenih elektroničkih elemenata u programski kod su se uključile knjižnice. Uslijedilo je definiranje varijabli i njihovih početnih stanja te elemenata pomoću ključnih riječi pojedine knjižnice. Korištenjem definiranih varijabli i inicijalnih riječi pojedine knjižnice pisane su funkcije za svaki element, a ukoliko je bilo moguće istovremeno se izvršilo testiranje na testnoj pločici. Kada su svi programski kodovi pojedinih funkcija bili napisani uslijedilo je testiranje i korekcija programskog koda, da bi sustav radio što efikasnije. Programsko sučelje Arduino IDE sadrži opciju „Serial Monitor“ koja se koristila za testiranje serijske komunikacije putem USB komunikacijskog porta s osobnim računalom. Nakon što je uspješno uspostavljena komunikacija slijedilo je stvaranje aplikacije u Microsoft Visual Studio 2015 kao što je opisano u radu.

Princip rada sustava elektroničke brave prikazan je maketom. Naravno, maketa sadržava osnovne funkcije elektroničke brave, koje je kasnije radi potrebe moguće nadograditi. Jedno od bitnih nadogradnji je povezivanje internetom, kreiranjem neke aplikacije preko koje je moguće praćenje korištenja samog sustava. Prilikom prvog pokretanja sustava vrši se inicijalizacija te nakon toga dolazi glavni izbornik koji je prikazan na zaslonu LCD-a. Preko tipkovnice biraju se opcije koje nam elektronička brava nudi. Prilikom odabira navedene opcije, unosom traženog, mijenja se položaj izvršnih elemenata te je svaka promjena obilježena zvučnim ili svjetlosnim

signalom. Prilikom rada Arduino vrši komunikaciju s USB portom računala te se podaci ispisuju i spremaju u tekst datoteku.

Realizacijom sustava Arduino elektroničke brave prikazano je da brava sadrži samo osnovne funkcije. Nadogradnja potrebnim funkcijama kao što je trajno spremanje PIN-a ili povezivanje s drugim sustavima je moguća. Jedno od ideja mogla bi biti npr. kreiranje mobilne aplikacije koja vrši komunikaciju s Arduino UNO i bilježi trenutna stanja elektroničke brave uz mogućnost istovremenog čitanja istih na mobilnom uređaju.

10. Literatura

[1] http://www.dhresource.com/0x0s/f2-albu-g2-M00-16-33-rBVaG1UnXPSAahPYAAmZA1r_J3s656.jpg/assa-abloy-digi-weatherproof-digital-electronic.jpg (dostupno 12.2.2016.)

[2] http://cms.sag.dvs.net/upload/DS_Varianten_Gastzimmer.jpg (dostupno 12.2.2016.)

[3] https://rapidnfc.com/blog/72/the_difference_between_nfc_and_rfid_explained (dostupno 12.2.2016.)

[4] https://en.wikipedia.org/wiki/Personal_identification_number (dostupno 25.2.2016.)

[5] https://en.wikipedia.org/wiki/Security_token (dostupno 25.2.2016.)

[6] http://www.technovelgy.com/ct/Technology-Article.asp?ArtNum=12 (dostupno 24.2.2016.)

[7] https://www.arduino.cc/en/Main/ArduinoBoardUno (dostupno 12.2.2016.)

[8] https://www.arduino.cc/en/uploads/Main/ArduinoUno_R3_Front_450px.jpg (dostupno 13.2.2016.)

[9] http://www.hobbyking.com/hobbyking/store/catalog/9549.jpg (dostupno 11.2.2016.)

[10] https://e-radionica.com/media/catalog/product/cache/5/image/9df78eab33525d08d6e5fb8d27136e95/d/s/dsc_4752.jpg (dostupno 14.2.2016.)

[11] http://modtronix.com/images/detailed/1/mod-lcdi2c-bb1.jpg (dostupno 14.2.2016.)

[12] http://www.allaboutsensors.com/image/cache/data/4x4%20matrix-500x500.JPG (dostupno 15.2.2016.)

[13] https://www.visualstudio.com/en-us/products/visual-studio-community-vs.aspx (dostupno 24.2.2016.)

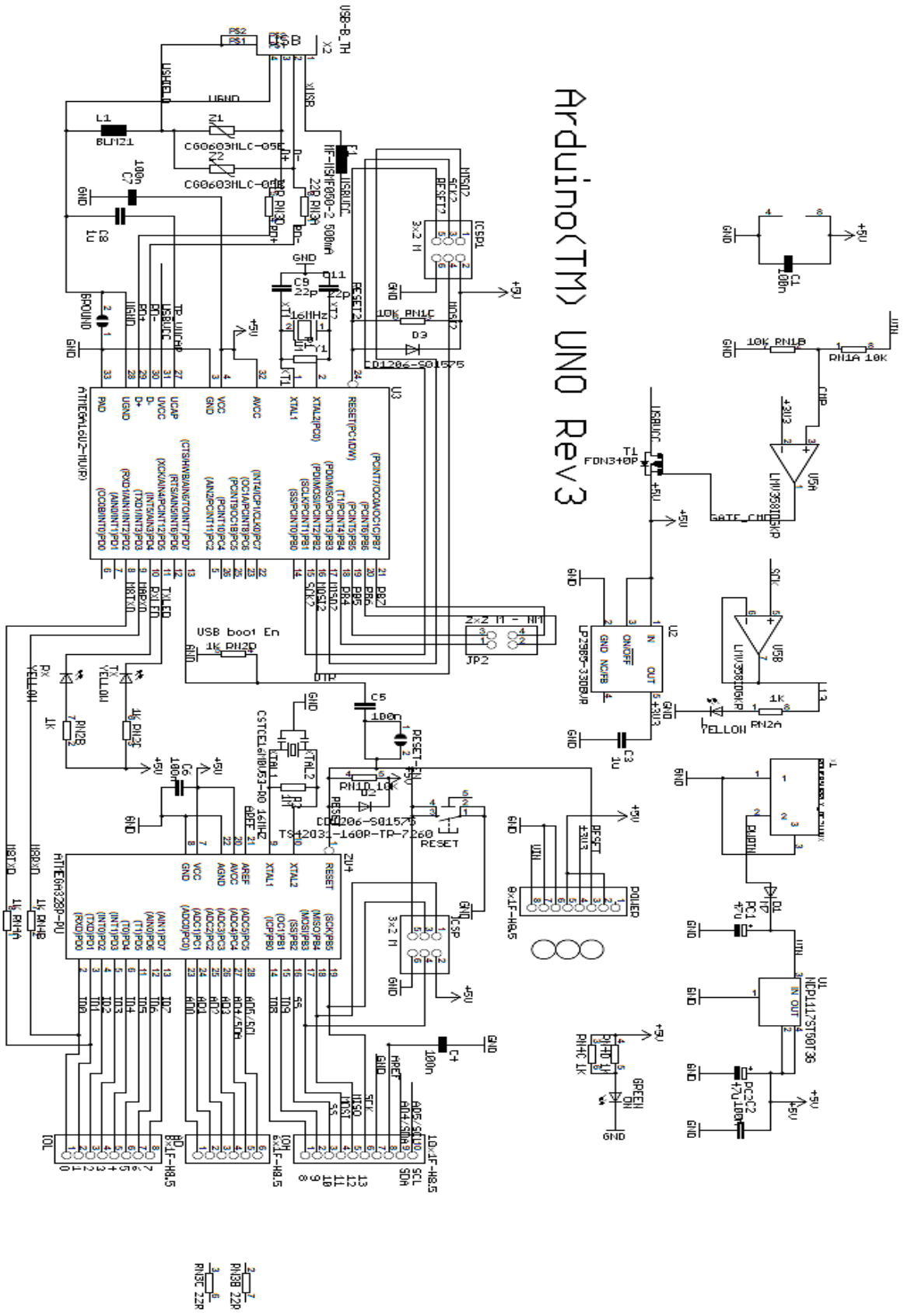
[14] https://img.fasttechcdn.com/121/1219302/1219302-1.jpg (dostupno 15.2.2016.)

[15] http://fritzing.org/home/ (dostupno 13.2.2016)

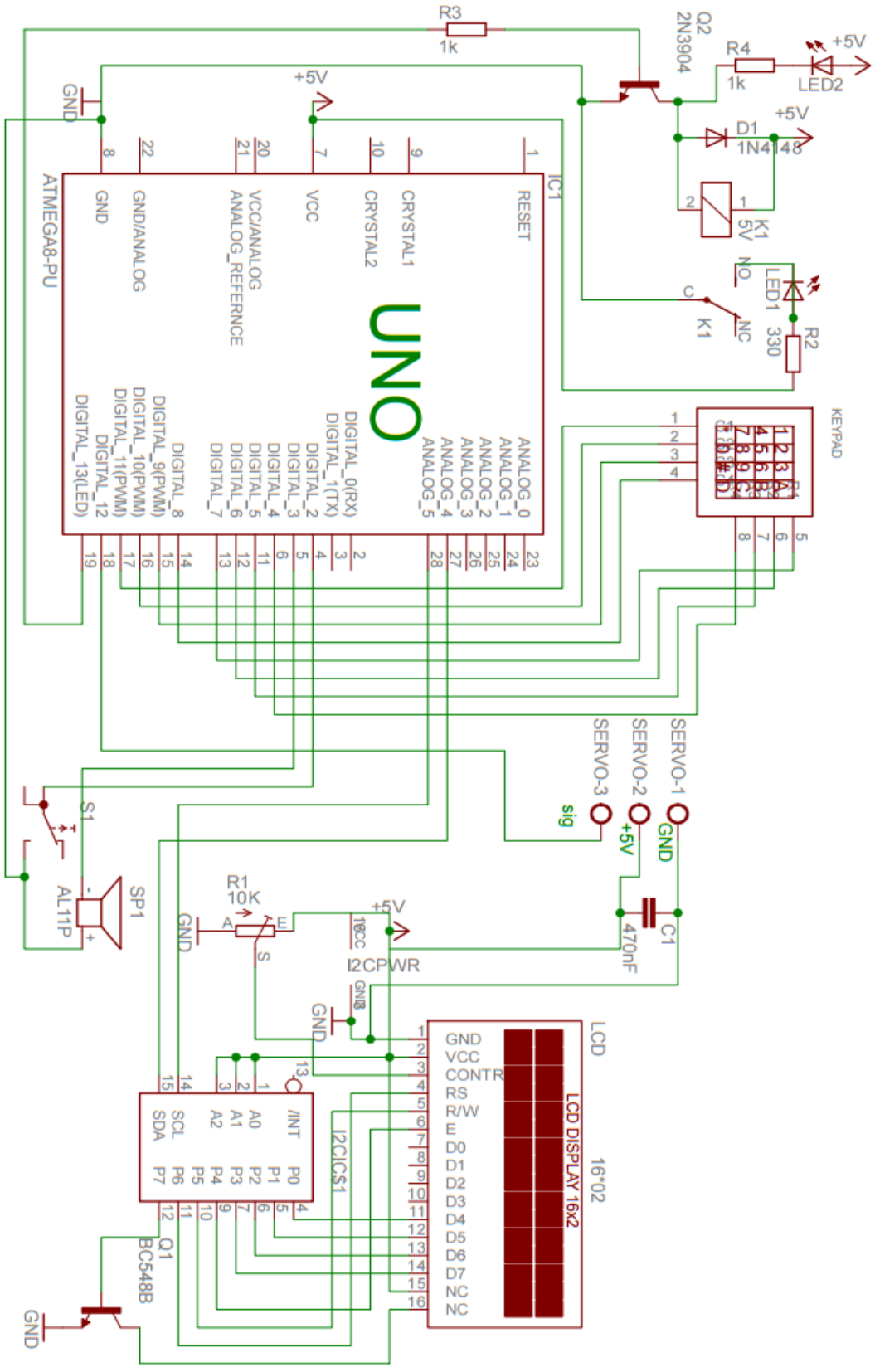
[16]http://www.electroschematics.com/wp-content/uploads/2014/11/arduino-uno-r3-schematic.png (dostupno 13.2.2016.)

11. Prilozi

PRILOG 1 : ELEKTRONIČKA SHEMA ARDUINO UNO (REV 3) [16]



PRILOG 2 : ELEKTRONIČKA SHEMA ARDUINO ELEKTRONIČKE BRAVE



PRILOG 3 : PROGRAMSKI KOD ARDUINO

```
// Završni rad //
// Goran Horvat //
// # Arduino Electronic Door Lock # //

#include <Wire.h>                    // knjižnica za komunikaciju I2C
pretvarača s LCD zaslonom
#include <LiquidCrystal_I2C.h>       // knjižnica za LCD zaslon
#include <Keypad.h>                  // knjižnica za tipkovnicu
#include <Servo.h>                   // knjižnica za servo motor

int sensor = 2;                      // pin tipkala
int buzzer = 3;                      // pin zvučnice
int relay = 13;                      // pin relejnog modula
boolean menurun = false;
int flag = 0;
int cup = 0;
int cdown = 0;

// definiranje tipkovnice
const byte rows = 4;
const byte cols = 4;

char keys[rows][cols] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};

byte rowPins[rows] = {11, 10, 9, 8};
byte colPins[cols] = {7, 6, 5, 4};

// kreiranje funkcije servo motora
Servo myservo;

// definiranje pocetnog PIN-a
```

```
char password[5] = "1234";
char tempP[5];
int tempPI = 0;
int tempPI2 = 0;

// kreiranje tipkovnice
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, rows,
cols);

// definicija lcd zaslona, 0x27 oznacava adresu
LiquidCrystal_I2C lcd(0x27, 16, 2);

// void setup je petlja koja se obavlja samo prilikom prvog ukljucenja
(sluzi za inicijalizaciju pocetnih stanja)
void setup() {

  Serial.begin(9600);
  delay(200);

  lcd.begin();
  lcd.backlight();

  myservo.attach(12);
  myservo.write(0);

  pinMode(buzzer, OUTPUT);
  pinMode(sensor, INPUT_PULLUP);
  pinMode(relay, OUTPUT);

  digitalWrite(relay, LOW);
  checkSensor();
  displayCodeEntryScreen();
}

// void loop je petlja koja se neprestano obavlja
void loop() {

  int l;
```

```
char key = keypad.getKey();

checkSensor();
delay(100);

while (menurun == false) {
  delay(500);
  menu();
}

if (int(key) != 0) {
  lcd.setCursor(10, 1);
  lcd.print("    ");
  lcd.setCursor(10, 1);

  for (l = 0; l <= tempPI; ++l)
  {
    lcd.print("*");
  }

  tempP[tempPI] = key;
  tempPI++;
  if (tempPI > 3) {
    tempPI = 0;
    if (checkP()) {
      unlockDoor();
      delay(2000);
      menu1();
    }
    else {
      invalidCode();
    }
  }
}

if (key) {
  switch (key) {
    case '*':
```

```
      checkSensor();
      while (flag == 0) {
        clearScreen();
        lcd.setCursor(4, 0);
        lcd.print("Najprije");
        lcd.setCursor(0, 1);
        lcd.print("zatvorite vrata!");
        delay(1000);
        clearScreen();
        checkSensor();
        delay(100);
      }
      tempPI = 0;
      lockDoor();
      flag = 1;
      break;

    case '#':
      while (flag == 0) {
        clearScreen();
        lcd.setCursor(4, 0);
        lcd.print("Najprije");
        lcd.setCursor(0, 1);
        lcd.print("zatvorite vrata!");
        delay(1000);
        clearScreen();
        checkSensor();
        delay(100);
      }
      lockDoor1();
      delay(500);
      lcd.setCursor(10, 1);
      lcd.print("    ");
      lcd.setCursor(10, 1);
      menurun = false;
      break;
  }
}
```

```
}

// funkcija promjene PIN-a - provjerava dali je unesen stari PIN
void ChangePassword() {
  clearScreen();
  delay(500);
  lcd.setCursor(3, 0);
  lcd.print("Stari PIN:");
  lcd.setCursor(10, 1);
  int l = 0;
  bool done = false;
  tempPI = 0;
  while (!done)
  {
    char key = keypad.getKey();
    if (int(key) != 0)
    {
      lcd.setCursor(10, 1);
      lcd.print("    ");
      lcd.setCursor(10, 1);
      for (l = 0; l <= tempPI; ++l)
      {
        lcd.print("*");
      }

      tempP[tempPI] = key;
      tempPI++;
      if (tempPI > 3) {
        tempPI = 0;
        if (checkP()) {
          done = true;
          SetNewPassword();
        }
        else {
          tempPI = 0;
          invalidCode();
          ChangePassword();
        }
```

```
      }
    }
  }
  clearScreen();
  delay(500);
  lcd.setCursor(0, 0);
  lcd.print("PIN promijenjen!");
  ring_bell();
  delay(75);
  ring_bell();
  delay(750);
  loginPage();
}

// funkcija unosa novog PIN-a te prebrisavanje starog
void SetNewPassword() {
  clearScreen();
  delay(500);
  lcd.setCursor(3, 0);
  lcd.print("Novi PIN:");
  lcd.setCursor(10, 1);
  int l = 0;
  bool done = false;
  tempPI2 = 0;
  while (!done)
  {
    char key = keypad.getKey();
    if (int(key) != 0) {
      lcd.setCursor(10, 1);
      lcd.print("    ");
      lcd.setCursor(10, 1);

      for (l = 0; l <= tempPI2; ++l)
      {
        lcd.print("*");
      }

      tempP[tempPI2] = key;
```

```
      tempPI2++;
      if (tempPI2 > 3) {
        tempPI2 = 0;
        for (int i = 0; i < 4; i++)
        {
          password[i] = tempP[i];
        }
        done = true;
      }
    }
  }
  Serial.print("Generiran je novi PIN: ");
  Serial.println(password);
}

// funkcija provjere stanja senzora
boolean checkSensor() {
  if (digitalRead(sensor) == HIGH) {
    flag = 0;
  }
  else flag = 1;
}

// funkcija provjere ispravnosti PIN-a
boolean checkP() {
  if (strncmp(password, tempP, 4) == 0) {
    return true;
  }
  return false;
}

// funkcija zvučne signalizacije
void ring_bell() {
  for (int x = 0; x <= 150; x++) {
    digitalWrite(buzzer, HIGH);
    delay(1);
    digitalWrite(buzzer, LOW);
    delay(1);
```

```
  }
}

// funkcija pogresnog PIN-a
void invalidCode() {
  clearScreen();
  lcd.setCursor(1, 0);
  lcd.print("Pogresan PIN!");
  delay(1500);
  loginPage();
}

// funkcija otkljucavanja
void unlockDoor() {
  clearScreen();
  lcd.setCursor(3, 0);
  lcd.print("Vrata se");
  lcd.setCursor(0, 1);
  lcd.print("otkljucavaju.");
  delay(500);
  lcd.setCursor(13, 1);
  lcd.print(".");
  delay(500);
  lcd.setCursor(14, 1);
  lcd.print(".");
  myservo.write(90);
  delay(500);
  ring_bell();
  digitalWrite(relay, HIGH);
  clearScreen();
  lcd.setCursor(4, 0);
  lcd.print("Vrata su");
  lcd.setCursor(2, 1);
  lcd.print("otkljucana!");
  Serial.println("Vrata su otkljucana i otvorena.");
  cup++;
  delay(1000);
  Serial.print("Ovo je ");
```

```
  Serial.print(cup);
  Serial.println(". ulazak");
  delay(1000);
}

// funkcija zakljucavanja
void lockDoor() {
  clearScreen();
  lcd.setCursor(3, 0);
  lcd.print("Vrata se");
  lcd.setCursor(1, 1);
  lcd.print("zakljucavaju.");
  delay(500);
  lcd.setCursor(14, 1);
  lcd.print(".");
  delay(100);
  lcd.setCursor(15, 1);
  lcd.print(".");
  delay(100);
  clearScreen();
  digitalWrite(relay, LOW);
  delay(100);
  myservo.write(0);
  delay(100);
  ring_bell();
  lcd.setCursor(4, 0);
  lcd.print("Vrata su");
  lcd.setCursor(2, 1);
  lcd.print("zakljucana!");
  Serial.println("Vrata su zatvorena i zakljucana");
  cdown++;
  delay(1000);
  flag = 0;
  Serial.print("Ovo je ");
  Serial.print(cdown);
  Serial.println(". izlazak");
  loginPage();
}
```

```
// 2. funkcija zakljucavanja u slucaju da vrata nisu zatvorena
void lockDoor1() {
  delay(100);
  clearScreen();
  digitalWrite(relay, LOW);
  delay(100);
  myservo.write(0);
  delay(100);
  ring_bell();
  delay(100);
  flag = 0;
}

// funkcija za prikaz autora
void displayCodeEntryScreen() {
  clearScreen();
  lcd.setCursor(2, 0);
  lcd.print("Zavrsni rad");
  lcd.setCursor(2, 1);
  lcd.print("Goran Horvat");
  delay(2300);
}

// funkcija glavnog izbornika
void menu() {
  clearScreen();
  lcd.setCursor(0, 0);
  lcd.print("1-ulaz");
  lcd.setCursor(0, 1);
  lcd.print("2-promjena PIN-a");
  lcd.setCursor(10, 0);
  char key = keypad.getKey();
  if (key == '1') {
    clearScreen();
    loginPage();
    menurun = true;
    tempPI = 0;
```

```
  }
  if (key == '2') {
    clearScreen();
    ChangePassword();
    menurun = true;
    tempPI = 0;
  }
}

// funkcija pod meni-a
void menu1() {
  clearScreen();
  lcd.setCursor(0, 0);
  lcd.print("*-zakljucavanje");
  lcd.setCursor(0, 1);
  lcd.print("#-izbornik");
  lcd.setCursor(10, 1);
  delay(500);
}

// funkcija unosa PIN-a
void loginPage() {
  clearScreen();
  lcd.setCursor(2, 0);
  lcd.print("Unesite PIN:");
  lcd.setCursor(10, 1);
}

// funkcija za brisanje sadržaja na LCD zaslonu
void clearScreen() {
  lcd.setCursor(0, 0);
  lcd.print("                    ");
  lcd.setCursor(0, 1);
  lcd.print("                    ");
  lcd.setCursor(10, 1);
}
```

PRILOG 4 : PROGRAMSKI KOD C#

```
using System;
using System.Windows.Forms;
using System.IO.Ports;

namespace Goran_Horvat_Zavrsni_rad_ErrorList
{
    public partial class Form1 : Form
    {
        private SerialPort myport;
        private DateTime datetime;
        private string in_data;

        public Form1()
        {
            InitializeComponent();
        }
        private void start_button_Click(object sender, EventArgs e)
        {
            InitializePort();
        }

        private void InitializePort()
        {
            myport = new SerialPort();
            myport.BaudRate = 9600;
            myport.PortName = port_tab.Text;
            myport.Parity = Parity.None;
            myport.DataBits = 8;
            myport.StopBits = StopBits.One;
            myport.DataReceived += myport_DataReceived;

            try
            {
                myport.Open();
                data_tab.Text = "";
            }

            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Error");
            }
        }


      void myport_DataReceived(object sender,
      SerialDataReceivedEventArgs e)
        {

            in_data = myport.ReadLine();
            this.Invoke(new EventHandler(displaydata_event));

        }
```

```
    private void displaydata_event(object sender, EventArgs e)
    {
        datetime = DateTime.Now;
        string time = datetime.ToString("dd/MM/yyyy\tHH:mm:ss");
        data_tab.AppendText(time + "\t\t\t" + in_data + "\n\n");
        try
        {
            string pathfile = @"C:\Users\Gogo\Desktop\DATA\";
            string filename = "Lista_dogadaja.txt";
            System.IO.File.AppendAllText(pathfile + filename, time
       + "\t\t\t" + in_data + "\n\n");
        }
        catch (Exception ex2)
        {
            MessageBox.Show(ex2.Message, "Error");
        }
    }


    private void stop_button_Click(object sender, EventArgs e)
    {
        try
        {
            myport.Close();
        }
        catch (Exception ex1) {
            MessageBox.Show(ex1.Message, "Error");
        }

    }

    private void port_tab_KeyDown(object sender, KeyEventArgs e)
    {
        if(e.KeyCode == Keys.Enter)
        {
            InitializePort();

        }
    }

    private void Form1_Load(object sender, EventArgs e)
    {

    }
}
```

UNIVERSITY NORTH

Sveučilište Sjever

VŽ KC

MMI

SVEUČILIŠTE SJEVER

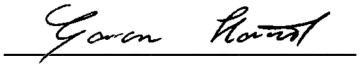
IZJAVA O AUTORSTVU

I
SUGLASNOST ZA JAVNU OBJAVU

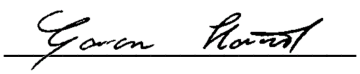
Završni/diplomski rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, Goran Horvat pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor završnog rada pod naslovom Arduino elektronička brava te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Goran Horvat



Sukladno Zakonu o znanstvenoj djelatnost i visokom obrazovanju završne/diplomske radove sveučilišta su dužna trajno objaviti na javnoj internetskoj bazi sveučilišne knjižnice u sastavu sveučilišta te kopirati u javnu internetsku bazu završnih/diplomskih radova Nacionalne i sveučilišne knjižnice. Završni radovi istovrsnih umjetničkih studija koji se realiziraju kroz umjetnička ostvarenja objavljuju se na odgovarajući način.

Ja, Goran Horvat neopozivo izjavljujem da sam suglasan s javnom objavom završnog rada pod naslovom Arduino elektronička brava čiji sam autor.

Goran Horvat