

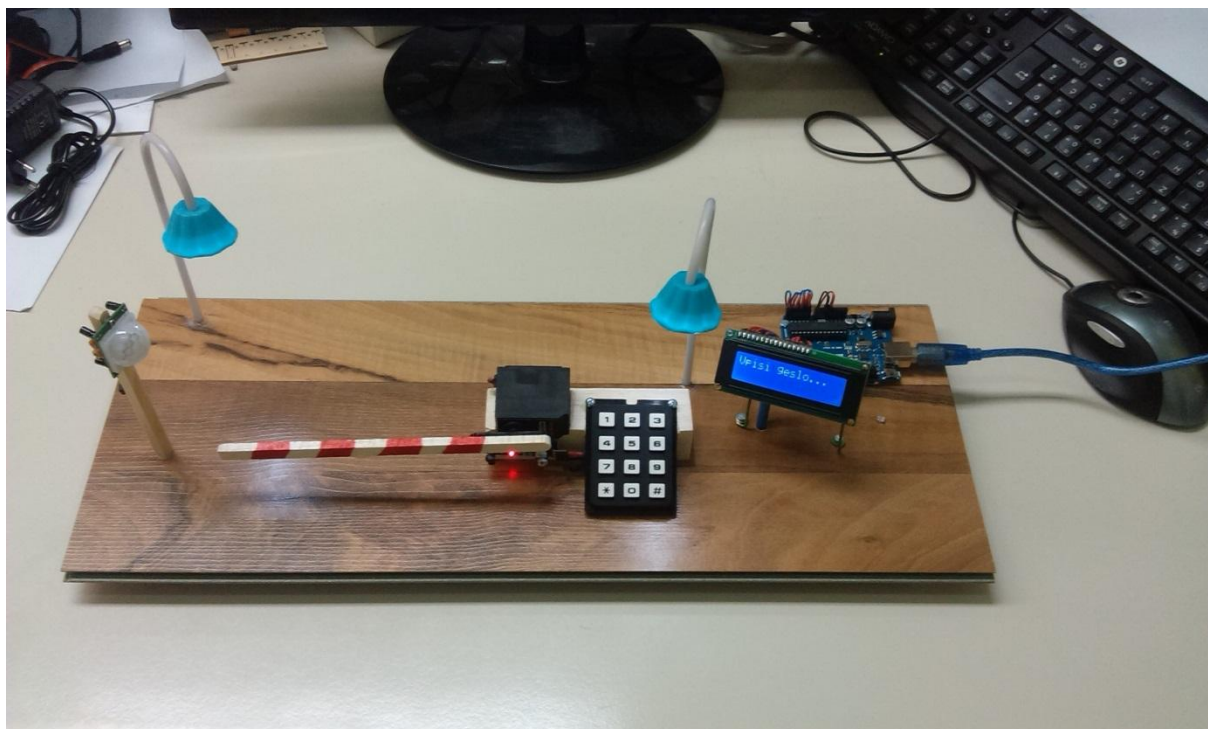
»Mladi za napredek Maribora«

## 32. SREČANJE

### Krmiljenje zapornice z razvojno ploščo Arduino Uno

Raziskovalno področje: Elektrotehnika, elektronika

Raziskovalna naloga



Avtor: TIMOTEJ NIDORFER, SIMON TOPLAK

Mentor: MILAN IVIČ

Šola: SREDNJA ELEKTRO-RAČUNALNIŠKA ŠOLA MARIBOR

Maribor, 2015

## Kazalo

### Vsebina

1. POVZETEK.....	3
2. Zahvala .....	3
3. Uvod .....	4
4. VSEBINSKI DEL.....	5
4.1. Razvojna plošča Arduino Uno .....	5
4.2. Posamezni elementi .....	6
4.2.1 Senzor gibanja .....	6
4.2.2 Senzor ovire.....	6
4.2.3 Senzor svetlobe .....	7
4.2.4 Servo motor.....	7
4.2.5 Matrična tipkovnica.....	8
4.2.6 LCD zaslon.....	9
4.3. Programiranje, opis programske kode .....	10
4.4. Priklučitev elementov na razvojno ploščo Arduino Uno .....	17
5. ZAKLJUČEK.....	17
6. DRUŽBENA ODGOVORNOST.....	18
7. VIRI .....	18

### Kazalo slik

Slika 1: Razvojna plošča Arduino Uno (vir: Arduino).....	5
Slika 2: Senzor gibanja (vir: modmypi) .....	6
Slika 3: Senzor ovire (vir: dx) .....	6
Slika 4: Senzor svetlobe (vir: instructables).....	7
Slika 5: Servo motor RS-2 (vir: Modelcraft).....	7
Slika 6: Zasuk servo motorja v odvisnosti od PWM signala (vir: avtor naloge) .....	8
Slika 7: Matrična tipkovnica (vir: Arduino forum) .....	8
Slika 8: LCD zaslon 16x2 (vir: avtor naloge).....	9
Slika 9: Prikaz priključitve elementov na Arduino Uno. ....	17

## **1. POVZETEK**

Naloga prikazuje krmiljenje zapornice z razvojno ploščo Arduino Uno. Zapornica lahko predstavlja dostop do parkirišča, do delavnice domačega podjetja, lahko pa bi zapornico zamenjali z vhodnimi ali garažnimi vrati. Poganja jo servo motor. Za dvigovanje zapornice je potrebno vpisati izbrano geslo na matrični tipkovnici. V kolikor geslo ni pravilno, dvigovanja ne sprožimo. Pod zapornico je nameščen IR senzor za zaznavanje ovire. Če je prisotna ovira, se zapornica ne spušča. V nočnem času se v primeru gibanja, dostopanja do vhoda oziroma matrične tipkovnice, vklopi razsvetljava. Dogajanje prikazuje dvovrstični LCD zaslon.

Za pravilno delovanje krmilja je po vnaprej določenih zahtevah napisan program v Ardoinovem okolju, ki podpira programski jezik C/C++. Na razvojni plošči Arduino Uno so uporabljeni digitalni ter analogni vhodi in izhodi.

Delovanje sistema prikazuje izdelana maketa.

## **2. ZAHVALA**

Za pomoč, nasvete, spodbudo in potrpljenje se iskreno zahvaljujema mentorju, pa tudi sošolcem, ki so nama občasno pomagali in svetovali. Zahvaliti se pa morava tudi staršem, ki so naju spodbujali nama pomagali pri izdelavi makete.

### 3. UVOD

Najprej smo si podali zahteve, kako naj sistem zapornice deluje. Kaj naj bo vključeno v nalogo in s čim naj krmilimo. Odločili smo se, da uporabimo razvojno ploščo Arduino Uno, ki temelji na Atmelovem 8-bitnem mikrokontrolerju ATmega 328P.

Za dvigovanje in spuščanje zapornice smo uporabili servo motor Modelcraft RS-2, saj smo se s tem izognili uporabi končnih stikal. Ker smo se odločili, da lahko dvigovanje zapornice sprožimo le z uporabo gesla, smo vključili matrično tipkovnico 3x4 (tri vrstice in štiri stolpce). V temi ni mogoče videti matrične tipkovnice in je zato oteženo vpisovanje gesla. Zato smo vključili senzor svetlobe, fotoupor in razsvetljavo. Da pa razsvetljave ne bi bila vključena po nepotrebnem, varčevanje z električno energijo, smo dodali senzor gibanja HC-SR501 in tako določili dva pogoja za vklop razsvetljave, prisotnost teme in gibanja v okolici sistema z zapornico.

S pravilno vpisanim geslom sprožimo dvigovanje zapornice. Dvignjena ostane določen čas, nato se začne spuščati v spodnji položaj. Če je pod zapornico prisotna oseba, avtomobil ali kakšna druga ovira, se zapornica sploh ne sme začeti spuščati. Če pa nastopi ovira med spuščanjem zapornice, se mora spuščanje nemudoma ustaviti in postaviti zapornico v gornji položaj. Za zaznavanje ovire smo uporabili infra rdeči senzor KeyesIR, katerega doomet znaša do 40 cm, kar je za prikaz na modelu dovolj.

Dogajanje, navodila in opozorila so prikazana na dvovrstičnem LCD zaslonu.

Pred pisanjem celotnega programa smo preverili delovanje posameznih elementov na tak način, da smo napisali dele programa, za vsak element posebej. Delovanje smo spremljali na serijskem monitorju, uporabili pa smo tudi merilni instrument, s katerim smo merili posamezne signale. Tako je nastalo vezje z vključenimi elementi na eksperimentalni ploščici. Pred pisanjem celotnega programa smo preverili vse zahteve, ki smo si jih zadali, ko smo se odločali, kako naj sistem z zapornico deluje. Moramo povedati, da smo nekatere zahteve spreminjali, saj smo večkrat naleteli na težave. Ker nam je primanjkovalo digitalnih priključkov, smo za LCD zaslon in tudi za matrično tipkovnico uporabili nekatere analogne priključke.

Po dokončni verziji diagrama poteka, smo začeli pisati program v Arduinovem okolju Arduino 1.0.6. Ko smo odpravili vse probleme in nam je sistem, sestavljen na eksperimentalni ploščici deloval, smo se odločili o izdelavi makete. Izbrali smo pozicije posameznih elementov, jih montirali in povezali z razvojno ploščo.

## 4. VSEBINSKI DEL

### 4.1. Razvojna plošča Arduino Uno



Slika 1: Razvojna plošča Arduino Uno (vir: Arduino)

Razvojna plošča Arduino Uno temelji na Atmelovem 8-bitnem mikrokontrolerju ATmega 328P. Njene glavne značilnosti:

Vsebuje 14 digitalnih vhodno / izhodnih priključkov, od katerih jih lahko 6 uporabimo za PWM<sup>1</sup> signale in 6 analognih vhodov. Delovna napetost znaša 5 V, posamezne elemente pa lahko napetostno napajamo kar iz razvojne plošče. Za to imamo na voljo napetost 5 V in 3,3 V. Paziti moramo, da posameznih izhodov ne preobremenimo. Posamezni izhod lahko obremenimo z največ 40 mA pri napetosti 5 V in z največ 50 mA pri napetosti 3,3 V.

Takt diktira 16 MHz oscilator. Vsebuje 32 KB *flash* programskega pomnilnika za vpisovanje programa v mikrokontroler. Ta se ne izbriše, če razvojni plošči izklopimo napajanje. Poleg *flash* vsebuje še 2 KB RAM in 1 KB EEPROM pomnilnika.

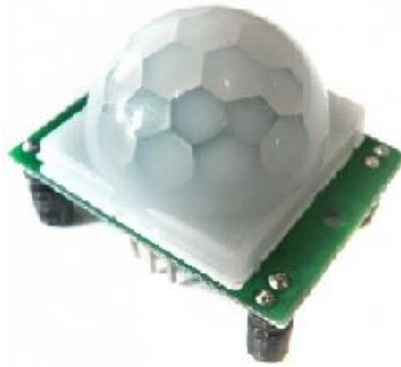
---

<sup>1</sup> PWM (ang. Pulse Width Modulation) pulzno širinska modulacija.

Razvojno ploščo lahko napajamo preko USB vhoda, lahko pa jo priključimo na ustrezní adapter.

## 4.2. Posamezni elementi

### 4.2.1 Senzor gibanja



Slika 2: Senzor gibanja (vir: modmypi)

Senzor gibanja je pasivni in ima napetostno napajanje od 4,5 V do 20 V in porabo do 50  $\mu$ A. Oddaja in infra rdečo sevanje in reagira na spremembo temperature, kar zazna kot gibanje. Na digitalnem izhodu da kot logično 1 napetost 3,3 V, kot logično 0 pa 0 V. Kot zaznavanja znaša 100° na razdalji do 7 m.

### 4.2.2 Senzor ovire



Slika 3: Senzor ovire (vir: dx)

Senzor oddaja IR žarke in sprejema odbite žarke na razdalji od 2 do 40 cm. Napetost napajanja znaša od 3 V do 6 V. Občutljivost lahko nastavljamo s potenciometrom. Deluje na frekvenci 38 KHz, ki jo generira ustrezno vezje s časovnikom NE555. Bolje zazna gladko in belo oviro kot črno in hrapavo.

### 4.2.3 Senzor svetlobe



Slika 4: Senzor svetlobe (vir: instructables)

Uporabljen je fotoupor LDR<sup>2</sup>, katerega upornost znaša pri dnevni svetlobi cca 4 k $\Omega$ , če ga zatemnimo pa mu upornost naraste na cca 80 k $\Omega$ . Povezali smo ga zaporedno z uporom 33 k $\Omega$ , da smo dobili napetostni delilnik. S spremembo svetlobe se v tej vezavi na fotouporu spreminja napetost.

### 4.2.4 Servo motor

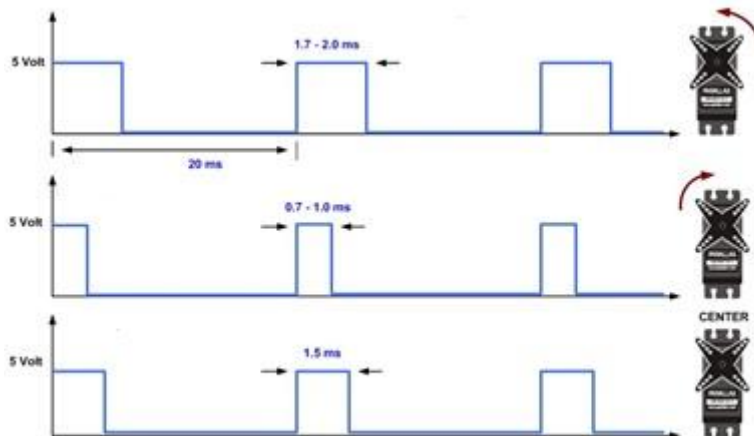


Slika 5: Servo motor RS-2 (vir: Modelcraft)

Srce servo motorja je majhen enosmerni motor, kateri se vrti pri visokih obratih. Ker ima zelo malo navora, ima zraven zobniški prenos, ki visoke obrate z malim navorom pretvori v počasne obrate z velikim navorom. Motor vsebuje še elektronsko vezje na tiskani ploščici, kjer se nahaja senzor. Ta prepozna za koliko se je zavrtela gred motorja. Gred motorja se lahko premika v obsegu 180°, od -90° do +90°. Pozicija gredi motorja je odvisna od dolžine impulza na priključku za signal, frekvenca teh impulzov pa je okoli 50 Hz. Položaj gredi torej določamo s PWM impulzi, ki trajajo od cca. 0,7 ms do 2 ms.

---

<sup>2</sup> LDR (ang. Light Dependent Resistor) svetlobno odvisen upor.



Slika 6: Zasuk servo motorja v odvisnosti od PWM signala (vir: avtor naloge)

Servo motor RS-2 ima navor 33 Ncm pri napetosti 5 V in se zavrtijo za 60° v 190 ms.

#### 4.2.5 Matrična tipkovnica



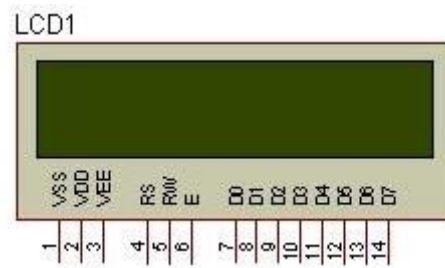
Slika 7: Matrična tipkovnica (vir: Arduino forum)

Matrična tipkovnica ima 12 razporejenih v 3 stolpce in 4 vrstice. Taka tipkovnica potrebuje vsega 7 priključkov, 3 za stolpce in 4 za vrstice. Če pritisnemo npr. tipko 5, staknemo vodnika drugega stolpca in druge vrstice. Kako ugotovimo katera tipka je sklenjena? V določenih časovnih intervalih "pošiljamo" napetost na posamezne stolpce, vendar le na enega hkrati. Vsakič, ko "pošljemo" napetost na posamezni stolpec, pregledamo stanje na posameznih vrsticah. Ker v vsakem trenutku vemo, kateri stolpec je trenutno pod napetostjo, lahko iz



prebranega stanja vrstic ugotovimo katera tipka je pritisnjena. Arduino ima vse to urejeno že v knjižnici, tako da v program zapišemo le matriko in priključke, ki jih bomo uporabili.

#### 4.2.6 LCD zaslon



Slika 8: LCD zaslon 16x2 (vir: avtor naloge)

LCD<sup>3</sup> zasloni omogočajo izpisovanje črk, števil in ostalih znakov. Uporabljamo jih kot vmesnike med uporabnikom in elektronsko napravo. Uporabili bomo dvovrstične LCD prikazovalnike z lastno osvetlitvijo, ki imajo v vsaki vrstici po 16 znakov. Vsak znak je sestavljen iz 5 x 8 pik. LCD zasloni brez možnosti osvetlitve zaslona imajo 14 priključkov, taki z možnostjo osvetlitve zaslona pa imajo 2 priključka več, torej 16 priključkov.

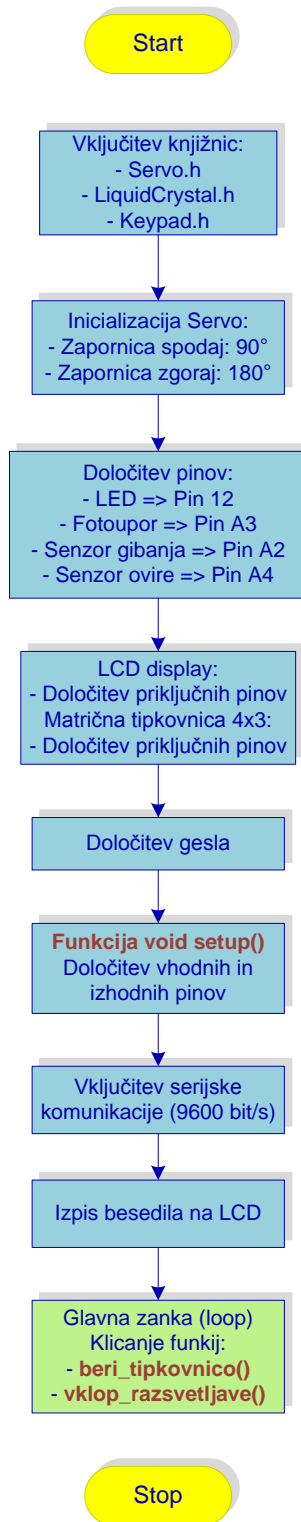
Priključki od 7 do 14 LCD zaslona so namenjeni podatkovnim linijam (od D0 do D7). Preko njih se prenašajo podatki od Arduino do LCD-ja, če pišemo v LCD in od LCD-ja do Arduino, če beremo iz LCD-ja. LCD prikazovalniki lahko delujejo v 8-bitnem ali 4-bitnem načinu. Mi smo izbrali 4-bitni način, saj s tem privarčujemo pri priključkih razvojne plošče Arduino. Ker iz LCD-ja ne bomo ničesar brali, smo priključek 5 (R/W) povezali z maso. Podatki se pri 4-bitnem načinu delovanja prenašajo po 4 bite hkrati, zato moramo poslati obe polovici bajta posebej, pošiljamo pa jih po podatkovnih linijah od D4 do D7. Tretji priključek LCD-ja služi nastavitvi kontrasta. Kontrast reguliramo z napetostjo na tem priključku.

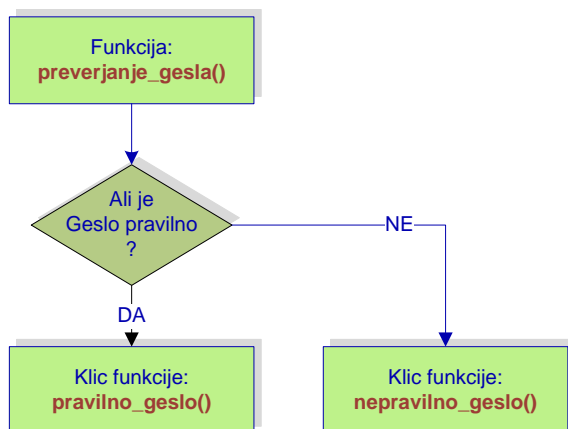
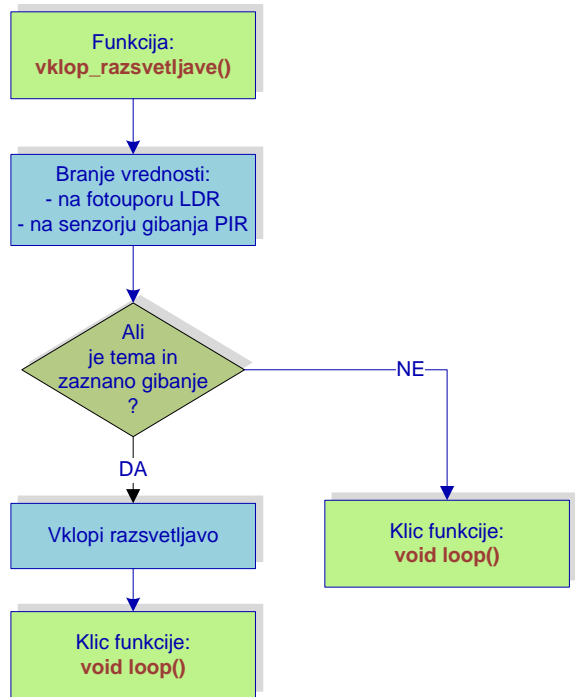
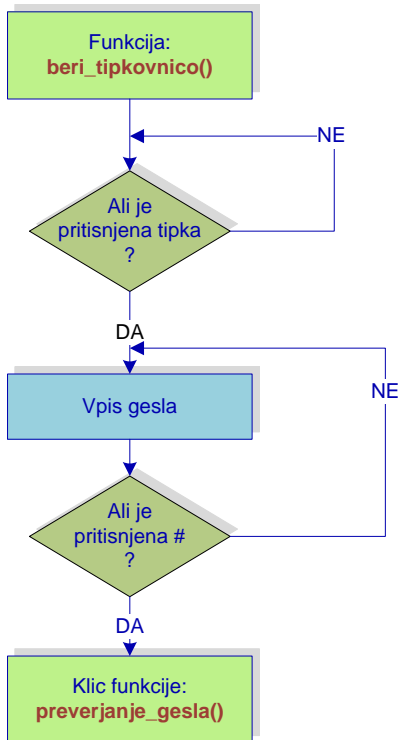
LCD krmilimo tako, da mu pošiljamo ukaze ali podatke. Če postavimo priključek 4 (RS) v stanje logične 0, bo LCD sprejel podatek kot ukaz, če pa ga postavimo na nivo logične 1, bo LCD sprejel navaden podatek. Priključek 6 (E) je namenjen vklopu prikazovalnikove logike. Ko pošiljamo neki podatek na LCD, ga moramo nekako obvestiti, da je na podatkovnih linijah nov podatek. To storimo tako, da ta priključek vklopimo (postavimo na 1) in izklopimo (postavimo na 0). Ob tem prehodu bo LCD sprejel nov podatek.

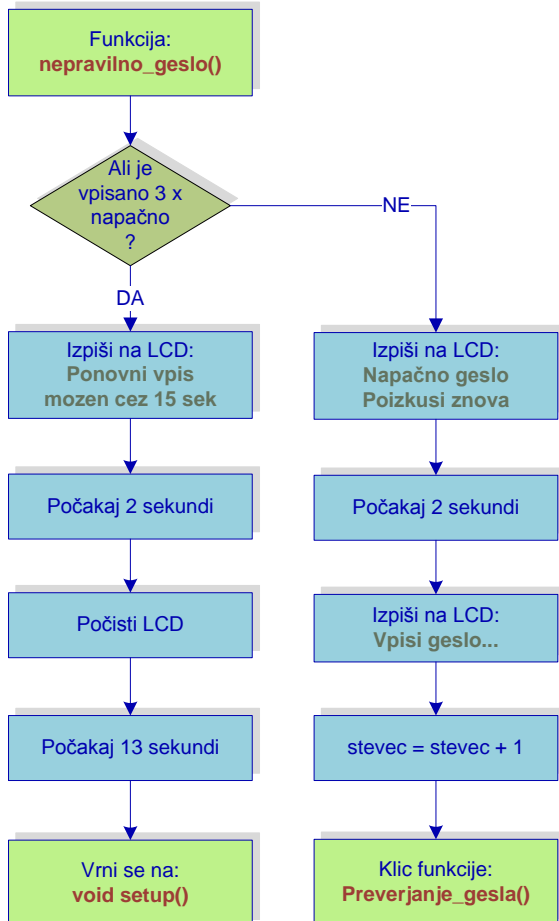
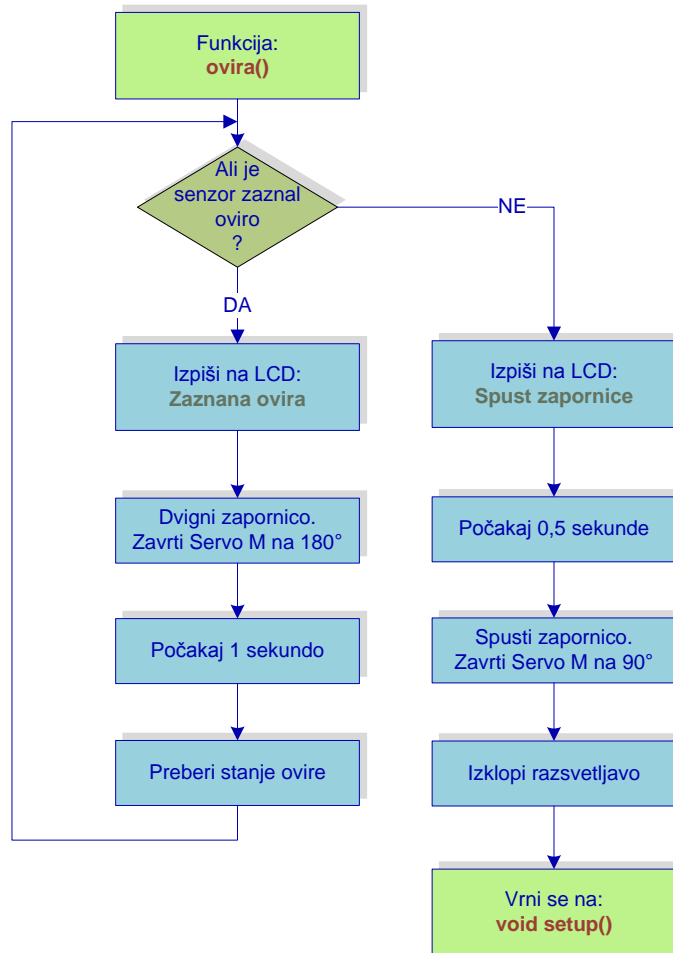
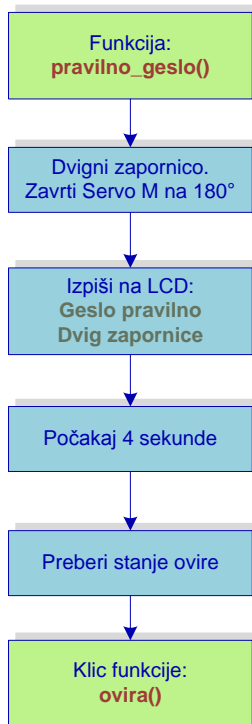
<sup>3</sup> LCD (ang. Liquid Cristal Display) prikazovalniki na tekoče kristale

### 4.3. Programiranje, opis programske kode

Najprej smo izdelali diagram poteka, ki ni preveč podroben, saj bi se izgubil pregled delovanja posameznih delov programa. Iz glavne zanke *void loop()* kličemo dve funkciji iz katerih nato po potrebi kličemo določene rutine oziroma podprograme.







Najprej smo uvozili knjižnice, potrebne za krmiljenje servo motorja, LCD prikazovalnika in matrične tipkovnice. Knjižnico za matrično tipkovnico smo morali poiskati na spletu, saj je v programskem okolju ni med privzetimi knjižnicami.

```
#include <Servo.h>           //Vključitev knjižnice za servo motor
#include <LiquidCrystal.h>   //Vključitev knjižnice za LCD
#include "Keypad.h"         //Vključitev knjižnice za matrično tipkovnico
```

Nato smo inicializirali položaj servo motorja v položaju, ko je zapornica dvignjena, deklarirali spremenljivko za štetje napačnih vpisov gesla ter določili priključke, na katere bomo priključili razsvetljava, senzor svetlobe, senzor ovire in senzor gibanja. Za posamezne senzore smo deklarirali spremenljivke, v katere se bodo zapisovale prebrane vrednosti. Določili smo tudi priključke na razvojni plošči Arduino, kamor bomo priključili LCD zaslon:

```
/*
Pini za LCD zaslon:
RS: Pin A1
EN: Pin A0
D4: Pin 2
D5: Pin 3
D6: Pin 4
D7: Pin 5
*/

LiquidCrystal lcd(A1, A0, 2, 3, 4, 5); //Priključki LCD na pine Arduino
```

Definirali smo tudi tip matrične tipkovnice ter priključitev vrstic in stolpcev:

```
//Definiranje tipa matrične tipkovnice:
const byte VRSTICE = 4; //štiri vrstice
const byte STOLPCI = 3; //trije stolpci
char tipke[VRSTICE][STOLPCI] =
{
  {'1', '2', '3'},
  {'4', '5', '6'},
  {'7', '8', '9'},
  {'*', '0', '#'}
};

//Matrična tipkovnica:
byte vrstice_Pini[VRSTICE] = {A5, 6, 7, 8}; //Priključitev vrstic matrične tipkovnice na pine Arduino
byte stolpci_Pini[STOLPCI] = {9, 10, 11}; //Priključitev stolpcev matrične tipkovnice na pine Arduino
Keypad keypad = Keypad( makeKeymap(tipke), vrstice_Pini, stolpci_Pini, VRSTICE, STOLPCI );
```

Določili smo geslo, ki ga je potrebno vtiskati na matrični tipkovnici za dvigovanje zapornice. V rutini *void setup()* smo določili vhodne in izhodne priključke razvojne plošče Arduino Uno, navodilo za uporabnike, ki se izpiše na LCD zaslon ter določili komunikacijo s serijskim monitorjem. Ta rutina se izvede samo enkrat, ob vklopu ali resetiranju. Glavna zanka *void loop()* pa se izvaja neprestano. V njej kličemo dve funkciji, funkcijo za branje tipkovnice in funkcijo za vklop razsvetljave:

```
void loop()
{
  beri_tipkovnico();      //Klicanje funkcije z imenom beri_tipkovnico
  vklop_razsvetljave(); //Klicanje funkcije z imenom vklop_razsvetljave
}
```

Pri branju tipkovnice najprej preverimo, ali je bila pritisnjena katera od tipk na matrični tipkovnici. Če je bila, se vtiskane vrednosti shranjujejo v spremenljivko *vpis\_gesla* tako dolgo, dokler ni pritisnjena tipka # ali tipka \*. Nato kličemo podprogram *preverjanje\_gesla()*, v katerem preverimo ali je vpisano geslo pravilno. Če je vtiskano geslo pravilno, se na LCD zaslon izpiše "Geslo pravilno" in zapornica se začne dvigovati.

Ker bi servo motor prehitro zavrtel zapornico v zgornji položaj, smo uporabili zanko for, ki se ponovi 70-krat. Ob vsakem obhodu zanke, se servo motor zavrti za 1° in počaka 15 ms.

```
for(int i = 70; i > 0; i--) //Zanka for se izvede 70 krat od 0 do 70 po koraku 1.
{
  servo.write(i); //Ob vsakem obhodu zanke for se zavrti servo motor za 1°.
  delay(15);      //Za počasnejše dvigovanje zapornice.
}
```

Zapornica ostane dvignjena 4 sekunde. Po preteku tega časa preberemo vrednost na analognem vhodu, kamor je priključen senzor ovire ter pokličemo podprogram *Ovira()*:

```
delay(4000);
Analogna_vrednost_senzorja = analogRead(Senzor_ovire); //Preverjanje stanja senzorja ovire.
Ovira(); //Klicanje rutine z imenom ovira.
```

Analogni vhod vsebuje 10-bitno analogni pretvornik, zato je pri napetosti 5 V njegova vrednost 1023. Kolikšna je ta vrednost, če ni ovire in kolikšna je v primeru ovire, smo programsko prebrali iz serijskega monitorja. Na podlagi dobljenih rezultatov smo določili mejo 250. Če je vrednost manjša, je zanesljivo prisotna ovira.

```
//Branje vrednosti na analognem vhodu A4 (IR senzor):
Analogna_vrednost_senzorja = analogRead(Senzor_ovire);
Serial.println(Analogna_vrednost_senzorja);           //Izpis btine vrednosti na serijskem monitorju.
if (Analogna_vrednost_senzorja > 250)                //Blok se izvede, če ni ovire.
```

V kolikor senzor ni zaznal ovire, se zapornica spušča po korakih za 1°. Po vsakem koraku preverimo prisotnost ovire. Če je le ta nastala med spuščanjem zapornice, se spuščanje ustavi in zapornica se nemudoma dvigne v zgornji položaj kjer ostane tako dolgo, dokler ni ovira odstranjena.

```
if (Analogna_vrednost_senzorja > 250)                //Blok se izvede, če ni ovire.
{
  lcd.clear();
  lcd.setCursor(2,0);
  lcd.print("Spuscanje");
  lcd.setCursor(2,1);
  lcd.print("zapornice");
  for(int j = 0; j<70; j++)    //Zanka for se izvaja po korakih 1 od 70 do 0 (če ni ovire).
  {
    Analogna_vrednost_senzorja = analogRead(Senzor_ovire);
    if (Analogna_vrednost_senzorja > 250)    //Ali je nastala ovira med spuščanjem zapornice?
    {
      servo.write(j);                //Ob vsakem obhodu zanke for se zavrti servo motor za 1°.
      delay(15);                    //Za počasnejše spuščanje zapornice.
    }
    else
    {
      //Je ovira.
      j=70;    //Iniciliziramo j, da servo začne spuščati zapornico iz zgornjega položaja.
      Ovira();    //Idi na rutino (podprogram) Ovira().
    }
  }
  digitalWrite(LED, LOW);    //Izklopi razsvetljavo.
  lcd.clear();
}
```

V primeru prisotnosti ovire se na LCD zaslonu prikazuje napis "POZOR OVIRA !!". V času gibanja zapornice sveti opozorilna LED dioda.

Uporabnik lahko trikrat vpiše napačno geslo. Če je napačen tudi četrti vpis, je ponoven vpis možen po določenem času, v našem primeru po 15-tih sekundah. Hkrati se na LCD zaslon izpisujejo ustrezna obvestila.

```
void nepravilno_geslo() //Se izvede, če geslo ni pravilno.
{
  if (stevec < 3)    //Ali je že 3-krat vpisano napačno geslo?
  {
    lcd.setCursor(0,0);
```

```

lcd.print("Napacno geslo!");
delay(1500);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Poizkusi znova");
delay(1000);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Vpisi geslo...");
}
else //Se izvede, če je bilo geslo vpisano 3-krat napačno in hočemo vpisati geslo četrtič.
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Ponovni vpis");
  lcd.setCursor(0,1);
  lcd.print("mozen cez 15 sek");
  delay(2000);
  lcd.clear();
  delay(13000);
  setup(); //Izvajanje programa na začetek, klicanje setup.
}
stevec = stevec + 1;
}

```

Razsvetljava se vklopi, še je tema in je zaznano gibanje v okolici zapornice. Podobno kot za senzor ovire, smo tudi senzor gibanje priključili na analogni vhod razvojne plošče. Vključena razsvetljava omogoča uporabniku gibanje in vpisovanje gesla preglednost. Razsvetljava se izklopi, ko se zapornica spusti v spodnji položaj. Pri postavitvi razsvetljave na modelu smo pazili pri izbiri senčil zaradi manjšega svetlobnega onesnaževanja.

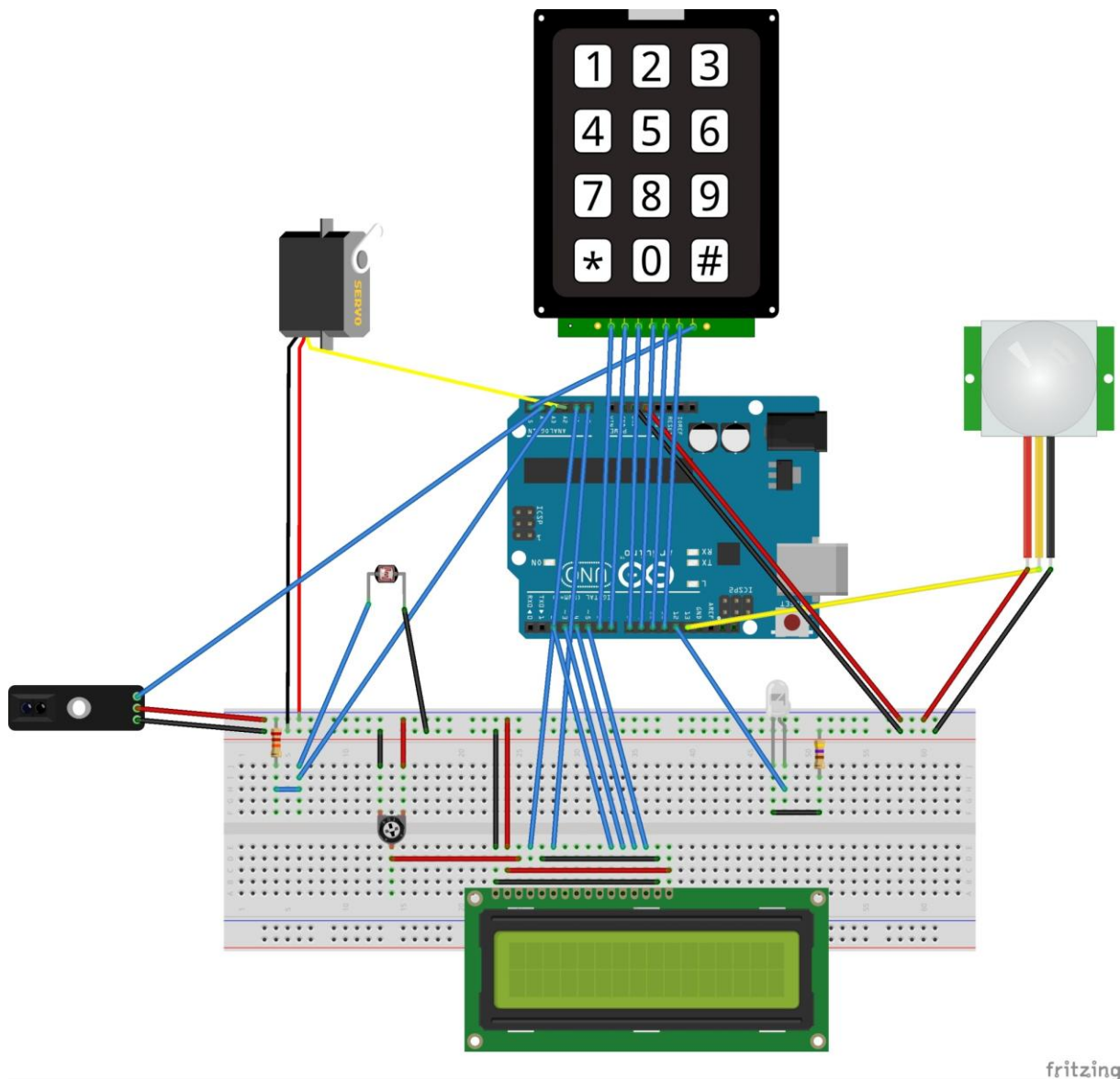
```

void vklop_razsvetljave()
{
  Analogna_vrednost_LDR = analogRead(LDR); //Branje vrednosti LDR.
  Analogna_vrednost_gibanje = analogRead(Senzor_gibanja); //Branje vrednosti senzorja gibanja.
  Serial.print(Analogna_vrednost_gibanje); //Izpiši podatek na serijskem monitorju.
  Serial.print(" ");
  Serial.println(Analogna_vrednost_LDR); //Izpiši podatek na serijskem monitorju.
  if((Analogna_vrednost_LDR > 250) && (Analogna_vrednost_gibanje > 500))
  {
    digitalWrite(LED, HIGH); //Vkljopi razsvetljava.
  }
}

```



#### 4.4. Priključitev elementov na razvojno ploščo Arduino Uno



fritzing

Slika 9: Prikaz priključitve elementov na Arduino Uno.

## 5. ZAKLJUČEK

Pri realizaciji raziskovalne naloge se je po eni strani pokazala enostavnost pri programiranju in uporabi Arduinove razvojne plošče, po drugi strani pa je prišlo prav znanje iz predmeta, kjer se učimo programski jezik C++.

Vsak element, ki smo ga vključili v nalogo zahteva znanje in razumevanje njegovega delovanja, saj smo le tako lahko določili njegovo vlogo in uspešno zaključili nalogo. Pri izvedbi nam je še največ preglavic delala vključitev pravilnega delovanja ovire, saj je lahko ovira nastala tudi tedaj, ko se je zapornica že spuščala.

Naloga ima vse lastnosti praktične izvedbe na katerem od že obstoječih objektov. Lahko bi tudi krmilili garažna vrata ali kakšna od drsnih vhodnih vrat. Za večje porabnike moči bi morali izdelati ustrezne vmesnike, da bi jih lahko krmilili brez težav.

Za vključitev dodatnih elementov, bi morali uporabiti večjo ploščo, na primer Arduino mega, saj smo na obstoječi porabili vse obstoječe analogne in digitalne vhode oz. izhode.

## 6. DRUŽBENA ODGOVORNOST

Naloga prikazuje na neki način varovanje objektov, saj je dostop možen le z uporabo gesla. Na ta način varujemo imetje pred nepridipravi. Vključeno je tudi varčevanje z energijo, saj se brez potrebe ne porablja, kot je na primer vključitev razsvetljave v okolici zapornice. Poskrbljeno pa je tudi za varnost, saj se zapornica v primeru ovire ne spušča in ne more poškodovati naključnih mimoidočih, če bi zašli v območje pod zapornico.

Znanje ki smo ga pridobili je lepa izkušnja za bodoče delo, ki bo v vsakem primeru nadgradnja obstoječega.

## 7. VIRI

- Arduino, <http://arduino.cc/en/Main/arduinoBoardUno> (08. Jan. 2015)
- modmypi, <https://www.modmypi.com/pir-motion-sensor> (08. Jan. 2015)
- dx, <http://www.dx.com/p/ir-infrared-sensor-switch-module-21517#.VM2wDy7BZnQ> (08. Jan. 2015)
- instructables, <http://www.instructables.com/file/FD829SGGYQES5W9> (08. Jan. 2015)
- Modelcraft, <http://www.servodatabase.com/servo/modelcraft/rs-2> (08. Jan. 2015)
- Arduino forum, <http://forum.arduino.cc/index.php?topic=27377.0> (08. Jan. 2015)