

»Mladi za napredek Maribora 2016«

33. srečanje

IR krmiljenje naprav

Raziskovalno področje: ELEKTROTEHNIKA, ELEKTRONIKA

Raziskovalna naloga

Avtor: DAVID KONEC, ANDREJ ŠRUMPF

Mentor: MILAN IVIČ

Šola: SREDNJA ELEKTRO-RAČUNALNIŠKA ŠOLA MARIBOR

2016, Maribor

Vsebina

1	ZAHVALA	4
2	POVZETEK	4
3	UVOD.....	4
4	VSEBINSKI DEL	5
4.1	POSAMEZNE KOMPONENTE.....	5
4.1.1	IR Sprejemnik in oddajnik	5
4.1.2	PIR Senzor (senzor gibanja)	8
4.1.3	Senzor plina.....	9
4.1.4	LED razsvetljava.....	10
4.1.5	Senzor ognja.....	11
4.1.6	Servo motor	11
4.1	DELOVANJE PROGRAMA	13
5	ZAKLJUČEK.....	14
6	VIRI.....	15

Kazalo slik:

Slika 1: IR sprejemnik in oddajnik (vir: geeetech.com).	6
Slika 2: Izpis kod na serijskem monitorju.....	7
Slika 3: Kode za posamezne tipke na IR oddajniku.	7
Slika 4: Del programa za sprejem kode.	8
Slika 5: Senzor gibanja (vir: adafruit).....	8
Slika 6: Del programa za senzor gibanja.	9
Slika 7: Del programa za senzor plina.	9
Slika 8: Senzor plina (vir: parallax).....	10
Slika 9: LED dioda, sestava in prikaz delovanja (vir: micro magnet).	10
Slika 10: Senzor ognja (vir: hub360).	11
Slika 11: Servo motor (vir: electrosome).	12
Slika 12: Vključitev knjižnice in nastavitve zasuka servo motorčka.	12

Slika 13: Vrtenje servo motorčka, zanka for.	12
Slika 14: Klicanje funkcij iz glavne zanke.	13

1 ZAHVALA

Iskrena hvala profesorju, ki je spremljal napredek, spodbujal k novim idejam in pomagal, ko se nama je kaj zalomilo.

2 POVZETEK

Namen te raziskovalne naloge je bil predvsem spoznati in uporabiti daljinsko krmiljenje naprav z uporabo razvojne ploščice Arduino. Zato je bilo potrebno spoznati iz raziskati programiranje v C oz. C++ programskem jeziku, na katerem temelji Arduinovo odprto-kodno okolje IDE.

Zadnje čase se v Sloveniji povečuje število vlomov v stanovanjske in druge objekte. Pogosti so tudi požari in zastrupitve z ogljikovim monoksidom. Zato sva se odločila, da bova naredila alarmni sistem, ki bo opozarjal ob takšnih dogodkih.

Vse naprave, ki jih krmiliva v nalogi uporabljajo IR komunikacijo. Z uporabo IR oddajnika in IR sprejemnika lahko krmilimo posamezna naprave od daleč. Takšno krmiljenje naprav je primerno tudi za osebe, ki imajo zaradi bolezni omejeno možnost gibanja. S tem jim olajšamo vsakodnevno življenje.

Za pravilno delovanje posameznih naprav, je bilo potrebno raziskati delovanje teh naprav in izbranih senzorjev, na podlagi katerih te naprave krmilimo.

3 UVOD

Na začetku smo si zadali zahteve, po katerih bo sistem deloval, katere nepravde bomo krmilili in katere senzorje bomo uporabili. Uporabljene elemente bomo montirali v maketo, ki jo bomo izdelali in bo služila predstavitvi delovanja.

Alarm se bo vklopil, če bo senzor plina zaznal preveliko koncentracijo plina, če bo senzor ognja zaznal prisotnost ognjenih zubljev in v primeru vloma. V tem primeru bodo nekatere funkcije avtomatsko izklopljene. V primeru požara, avtomatski vklop ventilacije ne sme biti omogočen, saj bi z ventiliranjem prostora razširjanje ognja še pospešili. V primeru povečane koncentracije plina v prostoru, pa se mora ventilacija nemudoma sprožiti, hkrati pa prisotne z zvočnih alarmom obvestiti o nevarnosti. Velikokrat se namreč zgodi, da se ljudje zastrupijo z nevarnimi plini, kot je ogljikov monoksid, med spanjem. Zvočni alarm bi v takšnih primerih marsikomu rešil življenje.

Z IR oddajnikom in sprejemnikom bomo krmilili naslednje naprave:

- Vklop / izklop ventilacije.
- Vklop / izklop notranje razsvetljave.
- Vklop / izklop zunanje razsvetljave.

- Odpiranje / zapiranje garažnih vrat.

Nekatere od teh naprav se bodo vklapljale in izklapljuje avtomatsko, odvisno od izbire uporabnika. Za avtomatsko delovanje skrbijo naslednji sensorji:

- Senzor plina.
- Senzor ognja.
- Senzor svetlobe.
- Senzor gibanja.

Za krmiljenje posameznih naprav v modelu sta uporabljena:

- Ventilator in
- servo motorček

Uporabljen je car MP3 IR daljinec ter AX-1838HS sprejemni modul.

4 VSEBINSKI DEL

Možgani, ki skrbijo za pravilno delovanje vseh naprav ja razvojna plošča Arduino. Obstaja več različnih plošč, v nalogi pa je uporabljena Arduino Uno. Je relativno poceni, če jo nabavimo preko ustreznih spletnih prodajaln. Njegovo programsko okolje je enostavno in razumljivo ter je primerno tako za začetnike kot za tiste uporabnike, ki načrtujejo večje projekte. Arduino Uno je tudi najbolj razširjena plošča.

Arduino Uno vsebuje Atmelov mikrokontroler ATmega328p. Vsebuje 14 vhodno/izhodnih pinov, od katerih jih lahko 6 uporabimo za PWM signal. Poleg teh vsebuje še 6 analognih vhodov. Za izvajanje programa skrbi 16 MHz kristalni oscilator. Z računalnikom jo povežemo preko USB vrat. Vsebuje tipko reset, konektor za priključitev napetostnega napajanja in vhodni napajalni pin za priključitev 9 V baterije.

Paziti moramo na obremenjenost posameznih izhodov. Obremenimo jih lahko do največ 20 mA. Nekateri pini imajo posebne funkcije, kot so RX in TX za serijsko komunikacijo, pina za proženje zunanjih prekinitov mikrokontrolerja, pini za SPI komunikacijo (SS, MOSI, MISO in SCK). Privzeto vsebuje več knjižnic, veliko pa jih najdemo na spletu.

4.1 POSAMEZNE KOMPONENTE

4.1.1 IR Sprejemnik in oddajnik

Infrardeče sevanje označuje elektromagnetno valovanje z valovnimi dolžinami, daljšimi od valovnih dolžin vidne svetlobe. Infra (lat. pod) pomeni, da je frekvenca infrardečega valovanja pod frekvenco rdeče svetlobe. Naš sistem deluje na frekvenci 38 kHz.

Infrardeči (IR) prenos podatkov uporabljamo v komunikaciji na kratke razdalje. Daljinski upravljalca uporablja infrardečo diodo za oddajanje infrardečega valovanja, ki ga leča fokusira v ozek žarek. Žarek je moduliran, tako da lahko prenaša podatke. Te podatke prejema IR sprejemni modul.

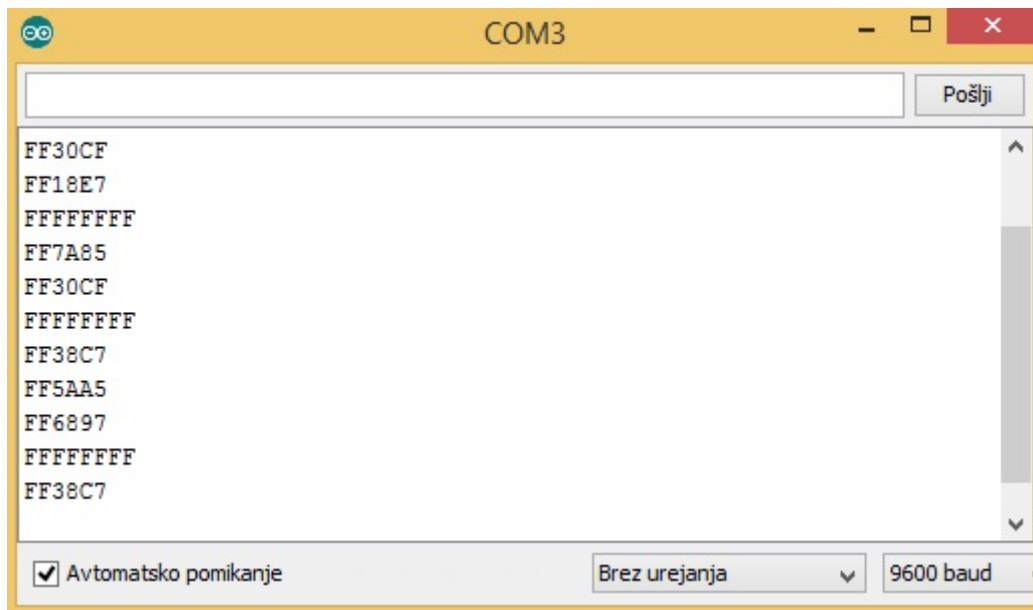
Ljudje infrardeče svetlobe ne vidimo, saj je izven vidnega spektra, lahko jo pa zazna video kamera ali digitalni fotoaparati na prenosnem telefonu. Če daljinski upravljalnik usmerimo v takšno napravo in pritisnemo na katerikoli gumb daljinskega upravljalnika, vidimo utripanje IR LED diode.



Slika 1: IR sprejemnik in oddajnik (vir: geetech.com).

S tem, ko IR LED dioda za kratek čas utripa, pošlje kodirani signal na priključek (pin) razvojne plošče Arduino Uno, ki ta signal demodulira. Za IR sprejemni modul smo uporabili [AX-1838HS](#), IR daljinski upravljalnik pa lahko uporabimo katerikoli, uporabimo pač tistega, ki ga imamo pri roki in deluje pri frekvenci 38 kHz.

Potrebovali smo knjižnico *IRremote.h*, ki smo jo našli na spletu. Ko vstavimo knjižnico v ustrezno mapo, zaženemo okolje Arduino in odpremo primer iz te knjižnice: *Datoteka > Primeri > IRremote > IRrecvDemo*. Program, ki ga zapišemo v Arduino Uno avtomatsko dekodira vrsto daljinskega upravljalnika, ki ga uporabljamo in ugotovi, na katero tipko smo pritisnili. Odpremo serijski monitor in pritisnemo na različne tipke IR daljinskega upravljalnika. Za vsako tipko, ki jo pritisnemo, se nam izpiše ustrezna koda v šestnajstiški (hex) obliki (`Serial.println(results.value, HEX);`).



Slika 2: Izpis kod na serijskem monitorju.

Na ta način smo prebrali kode, ki so se izpisale na serijskem monitorju. Ko smo pritisnili na določeno tipko IR daljinskega upravljalnika, smo dobili naslednje kode:

Tipka na IR daljinskem upravljalniku	Koda
0	FF6897
1	FF30CF
2	FF18E7
3	FF7A85
4	FF10EF
5	FF38C7
6	FF5AA5
7	FF42BD
8	FF4AB5
9	FF52AD

Slika 3: Kode za posamezne tipke na IR oddajniku.

IR sprejemnik smo priključili na napetost 5 V, izhodni pin IR sprejemnika pa smo povezali z ustreznim pinom ploščice Arduino. Zaradi boljšega sledenja in preverjanja sprejemnih kod, smo vključili komunikacijo s serijskim vmesnikom. Tako smo na serijskem monitorju imeli pregled prispelih kod, ki so bile odvisne od pritiskov na posamezne tipke IR oddajnika.

S strukturo switch-case smo v programu določili nalogo, ki se mora izvršiti ob pritisku na določeno tipko. V spodnjem primeru vidimo, da se mora vklopiti razsvetljava (LED dioda), če na IR oddajniku pritisnemo na tipko 1.

```
if (irrecv.decode(&results))
{
  switch(results.value)
  {
    case 0xFF30CF:      //Pritisnjena je tipka 1 na daljinskem upravljalniku.
      digitalWrite(2, HIGH); //Vklopi LED diodo, priključeno na pin 2.
      break;
  }
}
```

Slika 4: Del programa za sprejem kode.

Za vklop oz. izklop posameznih naprav v nalogi smo si določili tipke na IR oddajniku, to pa smo upoštevali pri pisanju programa.

4.1.2 PIR Senzor (senzor gibanja)

Uporabljeni senzor gibanja (PIR - pasivni infra rdeči senzor) ima odzivnost 0,3 do 18 sekund, odvisno od nastavitve. Za delovanje uporabljajo infrardečo energijo, signal ki sproži izhod sensorja ob premikajoči osebi. Domet sensorja je okoli 7 m, zaznavni kot pa znaša 120°.



Slika 5: Senzor gibanja (vir: adafruit).

Senzor gibanja ima digitalni izhod, zato smo izhodni pin sensorja povezali z digitalnim vhodom plošče Arduino. V programu smo s stavkom if - else preverjali stanje izhoda IR sensorja in na ta način dobili podatek o tem, ali je gibanje v dometu sensorja zaznano. V kolikor je gibanje zaznano, se vklopi ustrezna LED dioda, v nasprotnem primeru pa se izklopi.


```

void Senzor_Gibanja_Zaznavanje()
{
  if(digitalRead(Senzor_Gibanja) == HIGH) //Ali je zaznano gibanje?
  {
    digitalWrite(LED_Pin, HIGH);          //Gibanje je zaznano, vklopi LED diodo
  }
  else
  {
    digitalWrite(LED_Pin, LOW);          //Ni gibanja, izklopi LED diodo
  }
}

```

Slika 6: Del programa za senzor gibanja.

4.1.3 Senzor plina

V nasprotju s senzorjem gibanja ima senzor plina analogni izhod. Zato smo izhod senzorja povezali z analognim vhodom plošče Arduino. Analogni vhodi plošče Arduino vsebujejo 10-bitne analogno-digitalne pretvornike. To pomeni, da analogno napetost na vhodu, ki se lahko spreminja od 0 V do 5 V pretvori v digitalni podatek. Ker je pretvornik 10-bitni, je lahko njegova največja vrednost 1023.

Delovanje senzorja plina smo preizkušali s plinom iz vžigalnika. Na serijskem monitorju smo spremljali analogni vrednost in na ta način določili mejo, ko se mora zaradi prisotnosti plina vklopiti alarm.

```

void Senzor_plina_zaznavanje()
{
  if(analogRead(Senzor_plina) > 460) //Ali je zaznana prisotnost plina?
  {
    digitalWrite(Piskač, HIGH);      //Plin je prisoten, vklopi piskač.
  }
  else
  {
    digitalWrite(Piskač, LOW);      //Plin ni prisoten, izklopi piskač.
  }
}

```

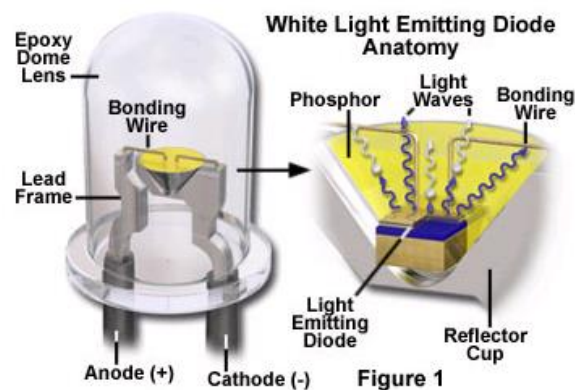
Slika 7: Del programa za senzor plina.



Slika 8: Senzor plina (vir: parallax).

4.1.4 LED razsvetljava

Svetleča LED dioda je polprevodniški elektronski element. Njene električne karakteristike so podobne navadni polprevodniški diodi s to razliko, da kadar prevaja tok, sveti. Razlikujejo se po barvi, velikosti, obliki in električnih karakteristikah. Svetloba ki jo oddajajo ima valovno dolžino v ozkem pasu. Svetleče LED diode sevajo svetlobo na principu t.i. sevanja trdnih teles. Kadar se elektron spusti na nižje energijsko stanje, odda odvečno energijo v obliki elektromagnetnega sevanja. Če je odvečne energije zadosti, jo zaznamo kot svetlobo. Izsevana svetloba je odvisna od uporabljenih materialov.



Slika 9: LED dioda, sestava in prikaz delovanja (vir: micro magnet).

Pri priključevanju LED diod na digitalne izhode plošče Arduino smo morali paziti, da je nismo preobremenili. Zato smo vsako LED diodo zaščitili z uporom vrednosti 470 Ω . Paziti moramo tudi na obremenjenost digitalnih izhodov plošče Arduino. Zato smo zunanjo razsvetljava na modelu, ki vsebuje več LED diod, priključili v kolektorski krog NPN tranzistorja BC337, ki ga krmili digitalni izhod plošče Arduino.

4.1.5 Senzor ognja

Senzor ognja se uporablja za zaznavanje ognja ali drugih virov svetlobe z valovno dolžino 760 nm do 1100 nm. Njegova občutljivost je najboljša v kotu sprejemanja svetlobe do 60° in v razdalji enega metra od vira svetlobe. Nekateri moduli imajo vgrajeno identifikacijsko LED diodo, ki ob primeru zaznavanja ognja začne svetiti. Uporabljen senzor ognja ima 4 priključke, dva za napetostno napajanje (5 V), ima pa še analogni in digitalni izhod. Mi smo v nalogi uporabili analogni izhod, zato smo ga priključili na analogni vhod plošče Arduino. Paziti smo morali, da nam pri nalogi na zmanjka digitalnih pinov plošče Arduino.

Občutljivost senzorja lahko do neke mere spreminjamo s trimer potenciometrom, ki ga modul vsebuje.



Slika 10: Senzor ognja (vir: hub360).

4.1.6 Servo motor

Servo motorji so samostojne električne naprave, ki premikajo in obračajo dele strojev s precizno natančnostjo. Najdemo jih skoraj povsod, od igrač do avtomobilov. Poleg vezic za napajanje je srce servo motorja 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. Dolžina impulzov PWM signala od 0,7 ms do 1 ms zavrtijo gred servo motorja v smeri urinega kazalca, dolžine od 1,7 ms do 2 ms pa gred zavrtijo proti smeri urinega kazalca. Pri standardnem servo motorju dolžina impulzov PWM signala 1,5 ms zavrtijo gred v položaj sredine.



Slika 11: Servo motor (vir: electrosome).

Servo motorček smo uporabili za odpiranje in zapiranje garažnih vrat. Za uporabo servo motorčka smo morali uvoziti knjižnico Servo.h. Plošča Arduino Uno vsebuje Atmelov mikrokontroler ATmega328p. Ta mikrokontroler vsebuje tri časovnike (timer-je) Timer0, Timer1 in Timer2. Timer0 in Timer2 sta 8-bitna, Timer1 pa je 16-bitni. Osnovna razlika med 8-bitnim in 16-bitnim je resolucija. 8-bitni ima resolucijo 255, 16-bitni pa 65535. V Arduino Uno je Timer0 namenjen zakasnitvam delay(), millis(), micros(), Timer1 uporablja knjižnico za servo motorčke Servo(), Timer2 pa za generiranje tonov Tone().

```
#include <Servo.h>           //Vključitev knjižnice za servo motor.

Servo servo;
int Garazna_vrata = 0;      //Garažna vrata zaprta (0°).
```

Slika 12: Vključitev knjižnice in nastavitve zasuka servo motorčka.

Ob vklopu naprave, ko se upoštevajo začetne nastavitve, se servo motorček postavi v pozicijo 0°. Ko pa s pritiskom na ustrezno tipko IR oddajnika sprožimo odpiranje garažnih vrat, se servo motorček zavrti v pozicijo 90°. Pri tem smo upočasnili vrtenje servo motorčka z uporabo zanke for. Ob vsakem obhodu zanke for, se servo motorček zavrti za 1°. Ker je v zanki upoštevana zakasnitev 15 ms, se servo motorček iz položaja 0° do položaja 90° zavrti v času 1,35 s.

```
//Zanka for se izvede 90 krat od 0 do 90 po koraku 1:
for(int i = 0; i < 90; i++)
{
  servo.write(i);           //Ob vsakem obhodu zanke for se servo zavrti za 1°.
  delay(15);                //Za počasnejše dvigivanje zapornice.
}
```

Slika 13: Vrtenje servo motorčka, zanka for.

Princip upočasnjenega vrtenja servo motorčka smo uporabili tudi pri zapiranju garažnih vrat.

4.1 DELOVANJE PROGRAMA

Na začetku, preden smo začeli s pisanjem programa smo si zadali vse naloge oziroma zahteve, po katerih mora naprava delovati. Izbrali smo si elemente, ki jih bomo uporabili v nalogi, potrebne senzorje, izvršilne člene, IR oddajnik ter IR sprejemnik.

Na začetku programa smo vključili vse potrebne knjižnice, objekta za servo motor in sprejem IR kod ter deklarirali potrebne spremenljivke, ki smo jim dodelile vrednosti pinov, kamor jih bomo priključili.

V funkciji void setup(), ki se izvede le 1 krat na začetku, smo določili vhodne in izhodne pine plošče Arduino. Na vhodnih pinih Arduino (mikrokontroler ATmega328p) neprestano preverja stanje. Na digitalnih pinih preverja ali je na vhodnem pinu napetost 0 V (logična 0) oziroma napetost 5 V (logična 1). Na analognih vhodih pa preverja kakšna je napetost na teh pinih. Na izhodne pine pa Arduino pošilja napetost 0 V ali pa napetost 5 V. V tej funkciji smo določili še kot, na katerega se zavrti servo motorček ob vklopu naprave ter omogočili sprejemanje IR kod.

Vsak sestavni del oziroma sklop smo preizkusili posebej, tako da smo vezje sestavili na preizkusno ploščico, povezali s ploščo Arduino, napisali program za posamezni sklop in ugotavljali delovanje. Pri preizkušanju delovanja in kontroli smo si veliko pomagali z merilnim instrumentom, v veliko pomoč pa nam je bil serijski monitor, saj smo posamezne dele programa pisali tako, da so se nam vrednosti posameznih spremenljivk, sprejete kode IR oddajnika in pogoji sproti izpisovali.

Za posamezne sklope tako smo imeli programe že napisane in preizkušene. Pojavilo se nam je vprašanje, kako vse te dele programov spraviti v glavno zanko void loop(). Zaradi preglednosti programa in posameznih opravil smo se odločili, da iz glavne zanke kličemo posamezne funkcije za posamezna opravila.

```
void loop
{
  Zaznavanje_svetlosti();
  Ugotavljanje_pritisnjene_tipke_IR();
  Prisotnost_plina();
  Prisotnost_ognja();
  Zaznavanje_gibanja();
}
```

Slika 14: Klicanje funkcij iz glavne zanke.

Ko smo izdelali celoten program, smo ga preizkusili na podlagi sestavljenih sklopov vseh uporabljenih elementov na preizkusni ploščici. Seveda so se nam pojavljali problemi, ko posamezni sklopi niso delovali po predvidevanjih. Včasih smo celo mislili, da je v IR oddajniku slaba baterija, preverjali delovanje IR oddajne diode s pametnim telefonom. Ko smo pravilno

vklučili izpise kod v odvisnosti od pritisnjene tipke na IR oddajniku na serijskem monitorju, smo napake hitreje odkrivali.

Ugotovili smo tudi, da smo pozabili vključiti varovanje, vklop alarma, če pride do vloma. Za opravljanje te naloge smo naknadno vključili reed rele, ki bo nameščen ob vhodnih vratih. Rele bo preklopil, ko se bodo vhodna vrata odprla in posledično vklopil alarm.

Ko se program izvaja, je izvajanje večkrat zasedeno s posameznim opravilom. Če smo v programu uporabili zakasnitev `delay()`, in zakasnitev traja daljši čas, potem program ne reagira takoj na drugo opravilo, ki bi jo moral izvesti, na primer vklop alarma. Enako se dogaja, če je izvajanje programa za daljši čas ujeto v zanko `for`. V kolikor bi hoteli takojšnji odziv na drugo pomembno opravilo, bi morali uporabiti prekinitve v mikrokontrolerju. Arduino Uno lahko proži prekinitve na časovniku (timer), pozna pa tudi zunanje prekinitve, ki jih lahko prožimo s spremembo signala na pinu 2 ali na pinu 3. Ko pride do prekinitve (če jih mi dovolimo), mikrokontroler preneha z izvajanjem programa v glavni zanki `loop`, skoči v prekinitveno rutino imenovano `ISR` (Interrupt service routine). Ko se ta zaključi, mikrokontroler nadaljuje z izvajanjem glavnega programa tam, kjer ga je pred prekinitvijo končal. Uporabljanje prekinitvev je za naju nekaj povsem novega, zato sva se odločila, da prekinitve v tej nalogi ne bova uporabila. Je pa izziv za delo v bodoče.

5 ZAKLJUČEK

Z raziskovalno nalogo sva zadovoljna, veliko novega sva se naučila in izvedela. Na začetku sploh nisva vedela kako bi se stvari lotila. Ko pa sva s pomočjo mentorja osvojila posamezna znanja, je bilo že nekoliko lažje. Včasih smo za posamezni sklop potrebovali veliko časa, saj nama mentor ni takoj izdal rešitve, temveč nama je na razumljiv način nakazal pot do rešitve problema.

Spoznala sva tudi, da je veliko znanja na spletu, le poiskati ga je potrebno in pridobljene informacije uporabiti pri reševanju problemov. Za posamezne senzorje, ki sva jih uporabila pri nalogi, sva morala na spletu poiskati tehnične podatke. In jih upoštevati pri vključevanju v nalogo.

Z znanjem, ki sva ga pridobila se bova lažje spopadala z novimi izzivi, ki jih imava že kar nekaj v planu. Najprej pa pridejo na vrsto prekinitve v mikrokontrolerju.

Pri delu sva bila odgovorna za vsako napako, zamudo in površnost. Misliva da je to tudi edina prava pot v najini bodoči poklicni karieri. Z najino raziskovalno nalogo sva hotela pomagati tudi ljudem z gibalnimi težavami, saj jim daljinsko upravljanje določenih naprav lajša življenje.

6 VIRI

<https://www.quora.com/How-do-an-IR-transmitter-and-receiver-work> (09. Jan. 2015)

<https://learn.sparkfun.com/tutorials/ir-communication> (09. Jan. 2015)

<http://www.futureelectronics.com/en/optoelectronics/infrared-receivers.aspx> (09. Jan. 2015)

http://missionscience.nasa.gov/ems/07_infraredwaves.html (09. Jan. 2015)

https://sl.wikipedia.org/wiki/Infrarde%C4%8De_valovanje (09. Jan. 2015)

<https://www.arduino.cc/en/Main/ArduinoBoardUno> (09. Jan. 2015)

<https://www.arduino.cc/en/Guide/Introduction> (09. Jan. 2015)

<https://www.arduino.cc/en/Tutorial/PWM> (11. Jan. 2015)

https://en.wikipedia.org/wiki/Pulse-width_modulation (11. Jan. 2015)

<http://playground.arduino.cc/Main/MQGasSensors> (11. Jan. 2015)

<http://www.fosilum.si/si/zakaj-led-svetila/led-dioda/> (11. Jan. 2015)

<http://whatis.techtarget.com/definition/light-emitting-diode-LED> (11. Jan. 2015)

https://sl.wikipedia.org/wiki/Svetle%C4%8Da_dioda (11. Jan. 2015)

https://en.wikipedia.org/wiki/Light-emitting_diode#Working_principle (11. Jan. 2015)

<http://www.jameco.com/jameco/workshop/howitworks/how-servo-motors-work.html>
(11. Jan. 2015)

<https://en.wikipedia.org/wiki/Servomotor> (16. Jan. 2015)

<http://www.ro.feri.uni-mb.si/predmeti/servosi/gradivo/SERVOMOTOR.PDF> (16. Jan. 2015)

<http://blog.electrodragon.com/analog-and-digital-sense-of-sensors-lm393-the-voltage-comparator/> (16. Jan. 2015)

<http://hub360.com.ng/shop-2/flame-sensor-2/> (16. Jan. 2015)

http://www.fut-electronics.com/wp-content/plugins/fe_downloads/Uploads/Flame-sensor-arduino.pdf (16. Jan. 2015)

https://tkkrlab.nl/wiki/Arduino_KY-026_Flame_sensor_module (16. Jan. 2015)

https://en.wikipedia.org/wiki/Flame_detector (16. Jan. 2015)

[http://www.seeedstudio.com/wiki/Grove - Flame Sensor](http://www.seeedstudio.com/wiki/Grove_-_Flame_Sensor) (16. Jan. 2015)

<http://www.fbselektronik.com/delovanje-reed-senzorjev.html> (19. Jan. 2015)

https://en.wikipedia.org/wiki/Reed_switch (19. Jan. 2015)

<http://www.fbselektronik.com/media/uploads/files/REED%20KATALOG-9-2014.pdf> (19. Jan. 2015)

https://en.wikipedia.org/wiki/Passive_infrared_sensor (21. Jan. 2015)

<http://www.internetnatrgovina.com/senzorji-gibanja-n-55.html?newsPath=12> (21. Jan. 2015)

Sparkfun:

<https://cdn.sparkfun.com/assets/4/1/6/1/c/5159e980ce395f8840000000.jpg> (11. Jan. 2015)

<https://cdn.sparkfun.com/assets/parts/1/7/8/4/08642-02-L.jpg> (19. Jan. 2015)

Wikipedia:

[https://upload.wikimedia.org/wikipedia/commons/3/38/Arduino Uno - R3.jpg](https://upload.wikimedia.org/wikipedia/commons/3/38/Arduino_Uno_-_R3.jpg) (11. Jan. 2015)

https://upload.wikimedia.org/wikipedia/commons/9/9e/Verschiedene_LEDs.jpg (11. Jan. 2015)

Adafruit:

https://learn.adafruit.com/system/assets/assets/000/002/305/medium800/learn_arduino_motor_servo.jpg?1396781543 (16. Jan. 2015)

<https://blog.adafruit.com/wp-content/uploads/2014/01/pirsensor.jpg> (21. Jan. 2015)

hub360:

<http://hub360.com.ng/wp-content/uploads/2015/03/Flame-Sensor-Module1.jpg> (16. Jan. 2015)

Parallax:

https://www.parallax.com/sites/default/files/styles/full-size-product/public/27983a.png?itok=l_OnVRKm (11. Jan. 2015)