

»Mladi za napredek Maribora«

35. srečanje

LIFE – SOS zapestnica

Inovacijski predlog

Raziskovalno področje: **Elektrotehnika in elektronika**

Prostor za nalepko

Avtor: JAKA WALDHÜTTER

Mentor: MIRAN WALDHÜTTER

Šola: SREDNJA ELEKTRO-RAČUNALNIŠKA ŠOLA

Maribor, 2018

»Mladi za napredek Maribora«

35. srečanje

LIFE – SOS zapestnica

Inovacijski predlog

Raziskovalno področje: **Elektrotehnika in elektronika**

Prostor za nalepko

Maribor, 2018

1 KAZALO VSEBINE

2	Kazalo vsebine	2
3	Kazalo slik	4
5	Povzetek	6
6	Zahvala	7
7	Vsebinski del	8
7.1	Uvod	8
7.2	Konkurenca	8
7.3	Sestava in elementi zapestnice	10
7.3.1	Arduino Pro Mini	10
7.3.2	Napajanje	11
7.3.3	Senzor kapacitivnosti	12
7.3.4	Povezovanje gumba	13
7.3.5	BK-SIM808 (BK-808)	14
7.3.6	QR koda	15
7.3.7	3D model	16
7.3.8	Senzor dotika	17
7.4	Programiranje posameznih elementov	18
7.4.1	Programiranje v okolju Arduino	18
7.4.2	Senzor kapacitivnosti	18
7.4.3	Senzor dotika – FC-91	19
7.4.4	Merilec srčnega utripa	20

7.4.5	BK-SIM808	21
7.5	Glavni program	25
8	<i>Zaključek</i>	27
9	<i>Družbena odgovornost</i>	28
10	<i>Viri</i>	29
10.1	Spletne strani:	29
10.2	Video viri	30
10.3	Dokumenti na spletu	30
10.4	Viri slik	31

2 KAZALO SLIK

Slika 1: SOS zapestnica	8
Slika 2: SOS obesek	9
Slika 3: MM715	9
Slika 4: Life na začetku	10
Slika 5: Arduino Pro Mini	10
Slika 6: Povezovanje Arduino Pro Mini-ja.....	11
Slika 7: Lipo baterija	11
Slika 8: Polnilno vezje	12
Slika 9: Merilec srčnega utripa	12
Slika 10: Povezava senzorja kapacitivnosti	13
Slika 11: Tipka	13
Slika 12: Povezava tipke	14
Slika 13: BK-SIM808.....	14
Slika 14: GPS antena	14
Slika 15: GSM antena.....	15
Slika 16: QR koda	15
Slika 17: Izpis podatkov na telefonu	16
Slika 18: Ohišje Life.....	16
Slika 19: Pokrov	17
Slika 20: FC-91	17

Slika 21: Programska koda za kapacitivnostni senzor in LED	19
Slika 22: Programska koda za senzor dotika in LED	20
Slika 23: Program merilca srčnega utripa	21
Slika 24: Uvodne nastavitve za BK-SIM808	21
Slika 25: Prekinitveni del program	22
Slika 26: Pridobivanje lokacije in pošiljanje sporočil	23
Slika 27: Branje sporočil	24
Slika 28: Prvi pogojni stavki	25
Slika 29: Pogojni stavek za tipko	25
Slika 30: Preverjanje prejetih sporočil	26
Slika 31: Brisanje sporočil	26

4 POVZETEK

Inovacijski predlog oz. izdelek, ki smo se ga lotili je GPS lokacijska zapestnica, ki deluje preko SMS-jev.

Starejše ljudi pošiljamo v upokojske domove, saj bi nas drugače ovirali. Konstantno moramo spremljati njihovo stanje, saj se, zaradi ostarelih kosti že najmanjša nesreča lahko spremeni v smrtno nevarno, če je ne odkrijemo pravočasno. Zato smo ustvarili zapestnico, katera s pritiskom na gumb, v naprej določenim številkam, pošlje sporočilo z lokacijo ponesrečene osebe.

Zapestnico lahko uporabljajo tudi družine, ki imajo mlajše otroke, ki hodijo peš v šolo. Lahko jo uporabljajo tudi varnostne službe, planinci in avanturisti. Če ima družina člana, ki ima težave s spominom, lahko tudi njegovo varnost zagotovimo z zapestnico.

Zapestnica »Life« pošlje sporočilo z lokacijo lastnika če:

- lastnik pritisne gumb,
- se lastniku ustavi srce,
- na številko zapestnice pošljemo sporočilo.

Zapestnica je sestavljena iz mikrokontrolerja Arduino ProMini. GPS in GSM pa ji omogoča modul BK-SIM808. Zapestnica ima le en gumb, vsebuje pa tudi senzor kapacitivnosti, senzor za dotik in merilec srčnega utripa.

Pripomočki, podobni naši ideji že obstajajo, zato smo se osredotočili na izboljšanje njihovih pomanjkljivosti in dodali še merilec srčnega utripa, ki ga ostali primerljivi pripomočki nimajo. Tako lahko, v primeru zastoja srca, zapestnica pošlje sporočilo, ki opozarja, da je situacija že zelo nevarna in moramo ukrepati drugače npr. poklicati reševalno službo. Ljudje si bodo zapestnico tudi snemali, kar pomeni, da bi merilec, srčnega utripa bral oz. kazal zastoj srca. To smo preprečili z dodajanjem doma izdelanega senzorja kapacitivnosti, ki zagotavlja to, da je zapestnica na roki. Dodali smo tudi senzor dotika FC-91, da minimaliziramo napačna branja.

Da bi preprečili stiskanje SOS gumba po nesreči, smo s pomočjo 3D-printerja ustvarili posebno ohišje, ki to preprečuje.

5 ZAHVALA

Za nasvete, pomoč in potrpljenje se zahvaljujemo svojemu mentorju, ki nam je stal ob strani skozi vse težave in nam svetoval ter pomagal izoblikovati boljšo in učinkovitejšo idejo. Zahvaljujemo se tudi svojim staršem in družinam, saj so nam njihove težave dale to idejo.

6 VSEBINSKI DEL

6.1 Uvod

Vsak od nas ima oz. bo imel starejšo osebo, za katero bo odgovoren. Večina se odloči, da bo svoje starše oz. starejše poslala v upokojenske domove, predvsem zato, ker želijo biti prosti oz. svobodni. Pri starejših in oslabeledih ljudeh se lahko tudi najmanjše nesreče spremenijo v smrtno nevarne, če niso opažene, zato moramo poskrbeti za konstanten nadzor, da jim lahko zagotovimo varnost. Družine, ki imajo člane s težavami s spominom morajo biti nenehno ob njihovi strani ali jim zagotoviti varuško, ki bo nanje pazila večino svojega časa. Družine, ki so v finančni stiski si ne morejo privoščiti javnega prevoza, svojemu otroku pa vseeno želijo zagotoviti varnost.

Z našim inovacijskim predlogom smo želeli rešiti vse te probleme.

Odločili smo se, da bomo izdelali zapestnico, ki bo omogočala GPS lociranje s pomočjo sporočil oz. GSM komuniciranja.

6.2 Konkurenca

Po raziskavi smo na trgu opazili podobne izdelke, primere katerih lahko vidimo na slikah 1, 2 in 3.



Slika 1: SOS zapestnica



Slika 2: SOS obesek



Slika 3: MM715

Po raziskavi smo opazili primanjkljaje pri vseh izdelkih :

- Izdelki za svoje delovanje potrebujejo povezavo s telefonom,
- Možnost klika po nesreči je zelo velika,
- Uporaba je zahtevna oz. zakomplicirana,
- Je v obliki obeska ali verižice, zato ni vedno dostopen.

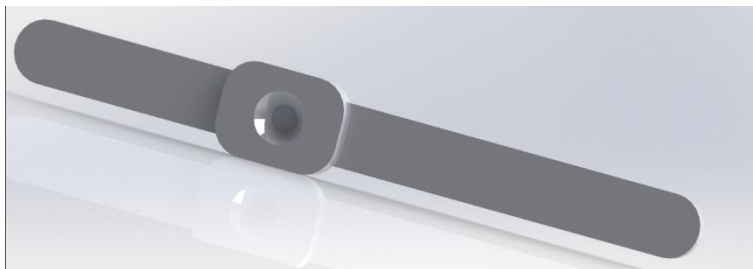
Po teh ugotovitvah smo ugotovili, da je bila odločitev, da bo izdelek v obliki zapestnice dobra, saj je tako najbolj dostopna v kritičnih situacijah. Odločili smo se, da bomo izdelali zapestnico, ki bo preprosta za uporabo in bo delovala brez povezave s telefonom. Odločili smo se tudi, da bomo v zapestnico vključili merilec srčnega utripa.

Med zbiranjem idej smo povprašali tudi nekaj ljudi o njihovih idejah ter željah, zanimalo pa nas je tudi ali bi jih tak izdelek zanimal.

Po pogovorih smo ugotovili, da je zelo zaželeno obojestransko komuniciranje. Odločili smo se da bomo poskusili omogočiti tudi prejemanje sporočil, to pa bi omogočilo uporabo zapestnice brez pasu za varovanje hišnih ljubljencev.

6.3 Sestava in elementi zapestnice

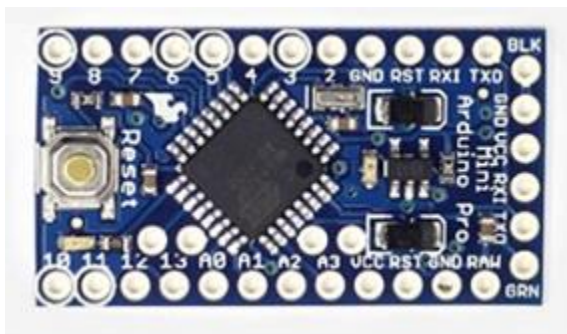
Na začetku smo želeli rešiti problem nenamernega stiskanja SOS gumba. Odločili smo se da bomo v ohišje naredili luknjo oz. udrtino vanjo pa dali gumb, kar bi preprečilo nezaželen stisk gumba ob udarcu na ravno podlago. Idejo smo poskusili zmodelirati s pomočjo Programa SolidWorks. Prvotno zamisel zapestnice lahko vidite na sliki 4.



Slika 4: Life na začetku

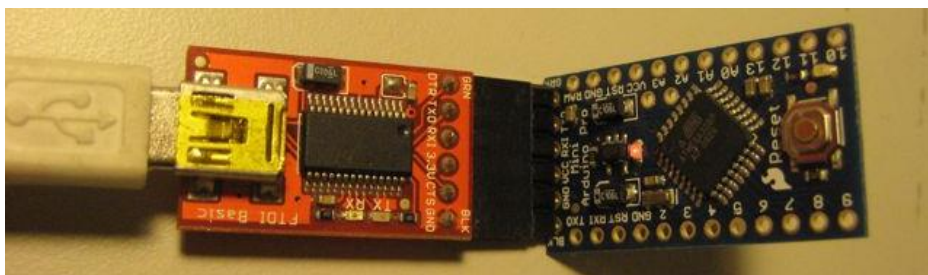
6.3.1 Arduino Pro Mini

Mikrokrmilniki Arduino so univerzalni krmilniki, ki delujejo na odprtokodni kulturi, zato so idealni za inovacijske projekte. V osrčju naše zapestnice je Arduino Pro Mini, saj je najmanjši in zato najbolj primeren za našo zapestnico. Vidimo ga lahko na sliki 5.



Slika 5: Arduino Pro Mini

Arduino Pro Mini je zasnovan na čipu ATmega328. Uporabili smo 5V verzijo, saj za pravilno delovanje modula BK-SIM808 potrebujemo takšno napetost. Arduino Pro Mini ima 14 digitalnih in 6 analognih vhodov. Za programiranje tega mikrokontrolerja potrebujemo poseben adapter, s pomočjo katerega se lahko povežemo z računalnikom (glej sliko 6).



Slika 6: Povezovanje Arduino Pro Mini-ja

6.3.2 Napajanje

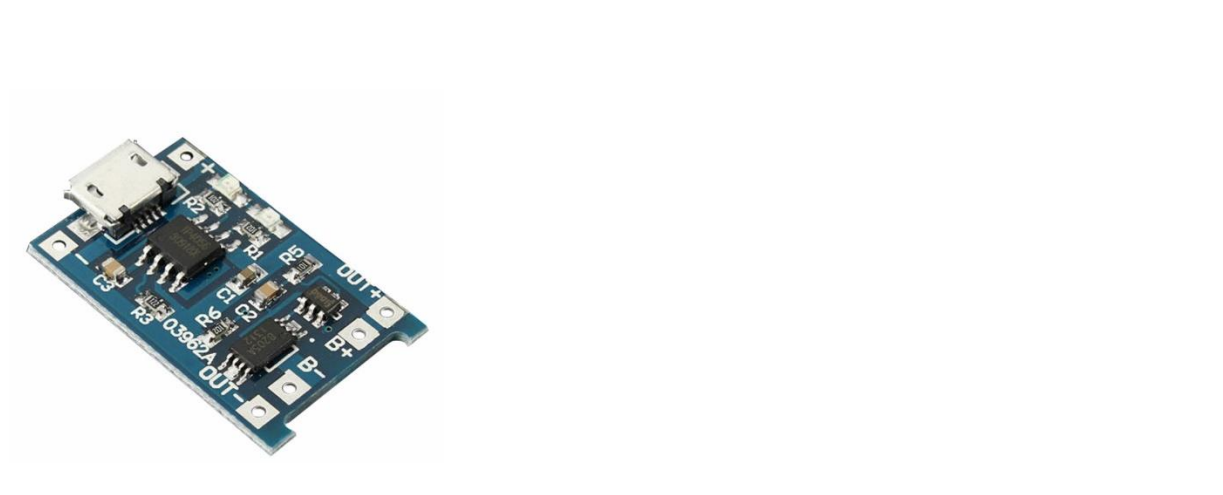
Za napajanje bomo uporabili LiPo (Litij-ionske) baterije. To so polnilne baterije, ki jih najpogosteje najdemo v telefonih. Mikrokrmilnik Arduino Pro Mini in modul bk-SIM808, za svoje delovanje potrebujeta 5 Voltov napajanja. Arduino Pro Mini ima vgrajen regulator, zaradi katerega lahko sprejme napajanje od 5 – 12 V.

Zaradi prostorske stiske smo se odločili da bomo uporabili 3,7 V Lipo baterije, ki jih bomo zaporedno povezali, da bomo dobili 7,4 V. Baterije, ki jih bomo uporabili lahko vidimo na sliki 7.



Slika 7: Lipo baterija

Ker lipo baterije zahtevajo, posebno polnjenje, za katero bi baterijo morali odstraniti iz zapestnice, zato smo nabavili posebno polnilno vezje, ki je na sliki 8. Le to nam omogoča polnjenje preko micro USB kabla, baterija pa se lahko polni medtem ko je priključena na Arduino Pro Mini. Na priključke b+ in b- priključimo baterijo, OUT+ in OUT- pa povežemo za GND pin VCC priključkom na Arduino Pro Mini-ju.



Slika 8: Polnilno vezje

Na sliki 9 lahko vidimo merilnik srčnega utripa, ki smo ga uporabili v zapestnici. Povezan mora biti na napetost in analogni vhod. Deluje na principu odboja svetlobe. Sestavljen je iz svetlobne diode (LED) in sprejemnika svetlobe. Svetlobna dioda sveti v tkivo, kjer se nekaj svetlobe absorbira nekaj, pa se je odbije nazaj. Odboj svetlobe je odvisen od volumna krvi v tkivu in je v razmerju z srčnim utripom.



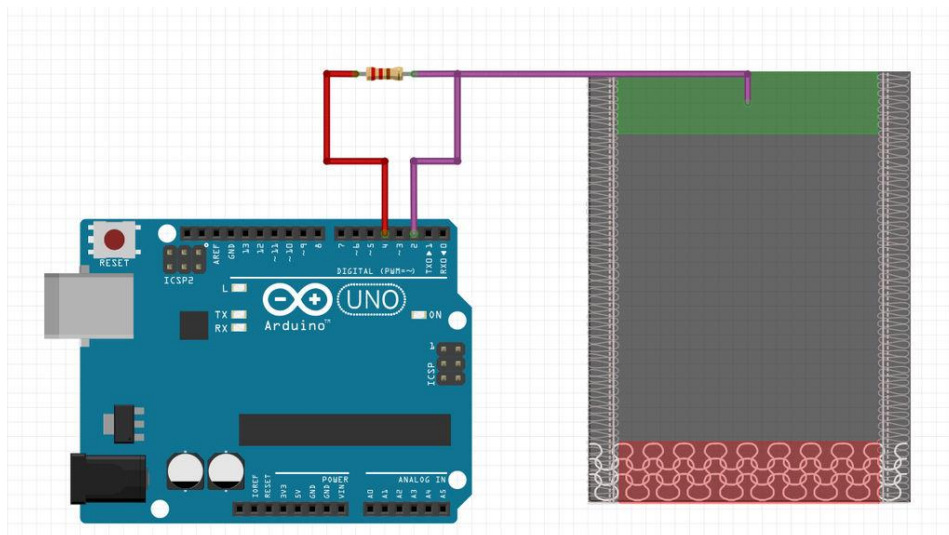
Slika 9: Merilec srčnega utripa

6.3.3 Senzor kapacitivnosti

Da bi preprečili oz. zmanjšali napačna (prazna) branja merilca srčnega utripa, smo v projekt dodali tudi senzor kapacitivnosti, ki smo ga izdelali kar sami. Njegova sestava je zelo preprosta – sestavljen je iz dveh žic, upora in nekega kosa kovine – aluminijaste folije, žice itd.

Žici sta povezani na dva digitalna vhoda na Arduino Pro Mini-ju, med njima pa je upor. To lahko vidimo na sliki 10 – vijolična žica je povezana na plast kovine.

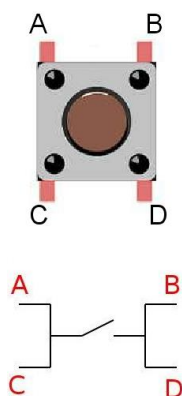
Senzor kapacitivnosti je v bistvu prevodna plast kovine, ki v bližini katerekoli druge prevodne plasti – npr. človeške kože začne prevajati. Deluje podobno kot kondenzator.



Slika 10: Povezava senzorja kapacitivnosti

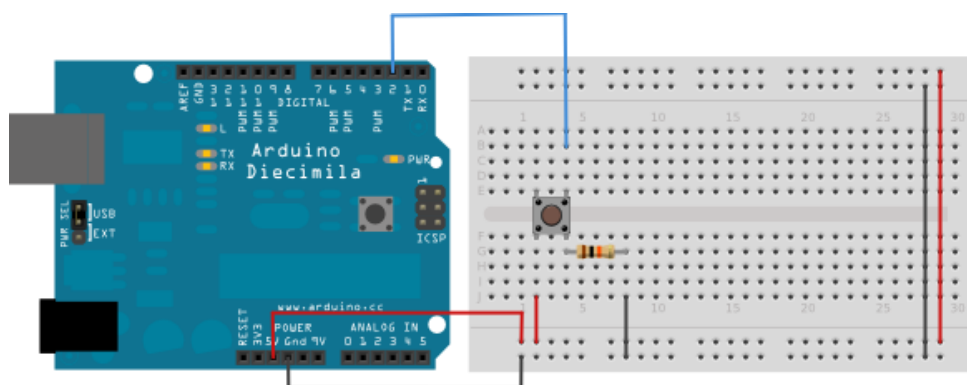
6.3.4 Povezovanje gumba

Zapestnica, bo imela samo en gumb (SOS gumb), ki bo ob pritisku poslal sporočilo. Zato ker bo gumb morda večkrat uporabljen, ne moremo uporabiti stikala, temveč moramo uporabiti tipko - ta ustvari stik le ko je pritisnjena. Sestavo tipke lahko vidimo na sliki 11.



Slika 11: Tipka

Tipko moramo na mikrokmlnik povezati , tako kot vidimo na sliki 12. Uporabiti moramo upor, mi smo uporabili upor z vrednostjo 10k Ω . Če tega upora ne bi bilo, bi vrednost na priključku bila ne definirana in vrednost bi se spreminjala.



Slika 12: Povezava tipke

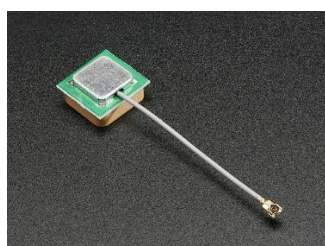
6.3.5 BK-SIM808 (BK-808)

Za GPS in GSM komunikacijo smo se odločili za vezje BK-SIM808. Vidimo ga lahko na sliki 13.

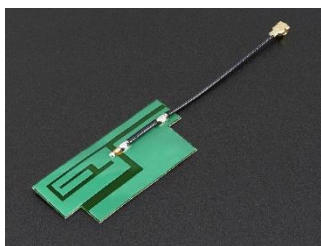


Slika 13: BK-SIM808

Vezje je zgrajeno okoli modula SIM808 podjetja SIM.Com. Ta nam omogoča GPS, GSM in Bluetooth komunikacijo. Mi bomo v projektu uporabili le GPS in GSM komunikacijo, za vsako od teh pa potrebujemo tudi ustrezne antene (glej sliki 14 in 15). Uporabili bomo pasivne antene, zato, da bomo porabili čim manj energije.



Slika 14: GPS antena



Slika 15: GSM antena

BK-SIM808 ima šest priključkov, komunikacija pa poteka le preko RXD in TXD serijskih priključkov.

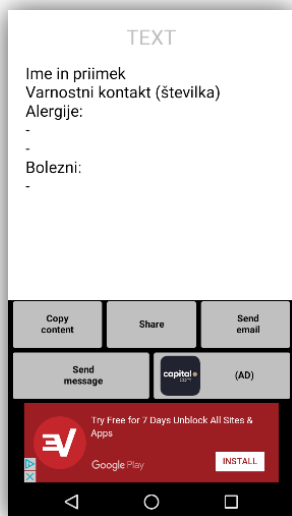
Za svoje delovanje BK-SIM808 potrebuje mikro SIM kartico, preko katere se lahko poveže v omrežje in začne s komunikacijo.

6.3.6 QR koda

Med raziskovanjem smo dobili še idejo, da bi lahko ljudje s pomočjo zapestnice dobili podatke o bolniku. Razmišljali smo o NFC tehnologiji, nato pa smo ugotovili, da kljub uporabnosti tehnologije, ta še ni dovolj razširjena, da bi bila zares uporabna. Odločili smo se, da bomo na zapestnico prilepili oz. dodali QR kodo, ki jo lahko ustvarimo brezplačno preko spleta. Vanjo bi vstavili, podatke o alergijah in posebnih boleznih ali potrebah bolnika, prav tako pa tudi ime in varnostni kontakt. Primer takšne QR kode lahko vidimo na sliki 16. Na sliki 17 pa lahko vidimo izpis podatkov na telefonu.



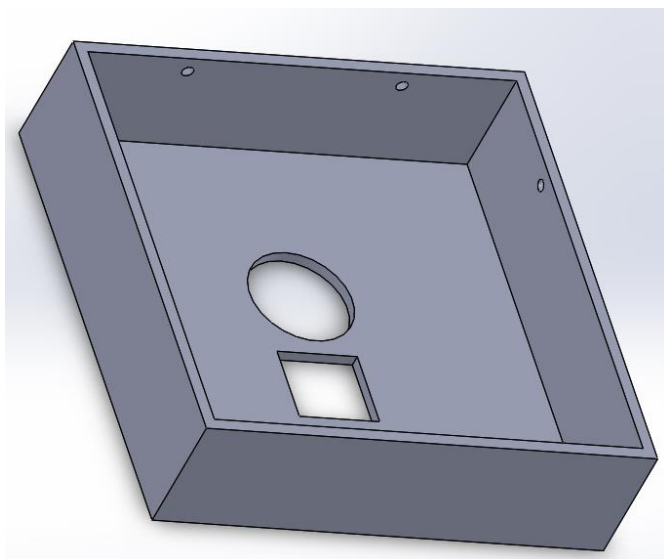
Slika 16: QR koda



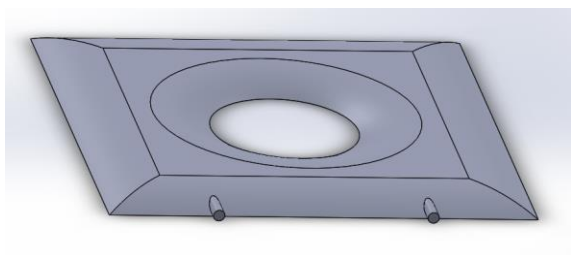
Slika 17: Izpis podatkov na telefonu

6.3.7 3D model

Odločili smo se, da bomo 3D natisnili ohišje, pas pa si bomo sposodili pri navadnem uri. V programu SolidWorks smo ustvarili model ohišja. Na sliki 18 lahko vidimo del ohišja v katerem se nahaja vsa elektronika na sliki 19 pa lahko vidimo pokrov. Pokrov na mestu ostane s pomočjo neodimskih magnetov ali vzmeti.



Slika 18: Ohišje Life



Slika 19: Pokrov

6.3.8 Senzor dotika

Pri preizkušanju senzorja kapacitivnosti so se v meritvah pojavljale napačne vrednosti, zato smo se odločili, da bomo v zapestnico vstavili še senzor dotika FC-91. Tega lahko vidimo na sliki 20.



Slika 20: FC-91

Ker se senzor odziva na dotik katerekoli podlage, smo spodnji del ohišja zaoblili za 0,4 cm. Zato lahko zapestnico odložimo na neko podlago in senzor kljub temu ne bo zaznal dotika. V dno smo dodali tudi dodatno luknjo. Senzor za merjenje srčnega utripa bo deloval, le če se zapestnica dotika človeške roke – če bo senzor kapacitivnosti zaznal njeno prisotnost, senzor dotika pa njen dotik.

6.4 Programiranje posameznih elementov

Arduino je zasnovan na odprto kodni kulturi, to nam je omogočilo, da smo programe za nekatere elemente sneli iz spleta, vendar smo hitro po tem ugotovili, da lahko te programe uporabimo zgolj za učenje.

6.4.1 Programiranje v okolju Arduino

Programiranje poteka v Arduino IDE – programu namenjeno programiranju mikrokrmilnikov Arduino. Arduino uporablja prirejen jezik, podoben programerskemu jeziku C++. Sestavljen je iz Void setup in Void loopa. Pred Void setup-om Določimo spremenljivke in programu povemo katere knjižnice in priključke bomo uporabljali. Programska koda napisana v Void Setup-u se izvede le enkrat in je namenjena določanju vrednosti, kot so npr. definiranje izhodov ali vhodov. Void Loop se pa izvaja do neskončnosti oz. dokler ima mikrokrmilnik napajanje.

6.4.2 Senzor kapacitivnosti

Arduino ima že ustvarjen program, ki omogoča branje senzorja kapacitivnosti. Tukaj uporabljamo priključka 2 in 4. V programu na sliki s pomočjo kapacitivnega senzorja vklapljammo svetilno diodo (LED).

```
#include <CapacitiveSensor.h>

CapacitiveSensor cs_4_2 = CapacitiveSensor(4,2); // 10 megohm resistor between pins 4 & 2, pin 2 is sensor pin, add wire, foil
int led = 7;
void setup(){

  cs_4_2.set_CS_Autocal_Millis(0xFFFFFFFF); // turn off autocalibrate on channel 1 - just as an example Serial.begin(9600);
  pinMode(led, OUTPUT);
}

void loop(){
  Serial.begin(9600);
  long start = millis();

  long totall = cs_4_2.capacitiveSensor(30);

  Serial.print(millis() - start); // check on performance in milliseconds

  Serial.print("\t"); // tab character for debug window spacing

  Serial.println(totall); // print sensor output 1

  delay(10); // arbitrary delay to limit data to serial port

  if(totall>5){
    digitalWrite(led, HIGH);}
  else
    digitalWrite(led, LOW);
  //delay(10);

}
```

Slika 21: Programska koda za kapacitivnostni senzor in LED

Vrednosti senzorja se izpišejo na serijskem monitorju, zato lahko spremljamo potek preko računalnika (glej sliko 22). Z »if« stavkom preverjamo, če senzor preseže določeno vrednost. Takrat je roka prisotna. Vrednost smo dobili z preizkušanjem, zato obstaja možnost, da se bo v končnem programu ta vrednost spremenila.

6.4.3 Senzor dotika – FC-91

Senzor dotika ima le en priključek, ki nam pove, če se kaj dotika senzorja. Če senzor zazna dotik, bo to pokazal z vrednostjo 1. Program na sliki 22 prikazuje program, ki ob dotiku senzorja svetlobno diodo vklopi in izklopi trikrat – s tem smo preverili, če se ob izpolnitvi pogoja izvede celoten program v if (pogojnem) stavku.

```
int senzor = 0;
int led = 7;
void setup() {
  pinMode(7, INPUT);
  pinMode(6, OUTPUT);
}
void loop() {

  if (digitalRead(senzor) == HIGH) {
    digitalWrite(led, HIGH);
    delay (500);
    digitalWrite (led, LOW);
    delay(500);
    digitalWrite(led, HIGH);
    delay(500);
    digitalWrite(led, LOW);
    delay (500);
    digitalWrite (led, HIGH);

  }

}
```

Slika 22: Programska koda za senzor dotika in LED

Z ukazom »int« programu dodamo spremenljivko, določimo pa ji lahko tudi priključek na katerem se nahaja. Z ukazom »pinMode« programu povemo ali je spremenljivka izhodna ali vhodna - ali jo bomo pisali ali brali. V tem primeru je senzor vhodna spremenljivka saj spremljamo njene spremembe, svetlobna dioda pa je izhodna spremenljivka, saj jo bomo sami spreminjali. Z ukazom »digitalWrite« lahko spreminjamo vrednosti na izhodnih spremenljivkah, z ukazom »digitalRead« pa beremo vrednost vhodnih spremenljivk.

6.4.4 Merilec srčnega utripa

Merilec moramo povezati na analogni vhod – ta lahko bere stanje spremenljivk, ki se spreminjajo po stopnjah, medtem ko lahko digitalni vhodi zaznajo le spremembo iz vrednosti 0 na vrednost 1. Merilec srčnega utripa deluje tako, da meri moč odbite svetlobe nazaj v senzor, zato pri njem potrebujemo analogni vhod.

Na sliki 23 je prikazan program s pomočjo katerega senzor določi vrednost utripov na sekundo. To vrednost nam izpiše na serijskem monitorju. Naša zapestnica se bo zagnala, ko bo utrip padel pod 40 utripov na sekundo.

```

#define USE_ARDUINO_INTERRUPTS true
#include <PulseSensorPlayground.h>

const int OUTPUT_TYPE = SERIAL_PLOTTER;
const int PIN_INPUT = A0;
const int PIN_BLINK = 13;    // Pin 13 is the on-board LED
const int PIN_FADE = 5;
const int THRESHOLD = 550;   // Adjust this number to avoid noise when idle
PulseSensorPlayground pulseSensor;

void setup() {
    Serial.begin(115200);

    pulseSensor.analogInput(PIN_INPUT);
    pulseSensor.blinkOnPulse(PIN_BLINK);
    pulseSensor.fadeOnPulse(PIN_FADE);

    pulseSensor.setSerial(Serial);
    pulseSensor.setOutputType(OUTPUT_TYPE);
    pulseSensor.setThreshold(THRESHOLD);
}

void loop() {
    delay(20);
    pulseSensor.outputSample();

    if (pulseSensor.sawStartOfBeat()) {
        pulseSensor.outputBeat();
    }
}

```

Slika 23: Program merilca srčnega utripa

6.4.5 BK-SIM808

Modul, ki ga uporabljamo, se upravlja le z AT ukazi. Programiramo ga lahko samo preko serijskih priključkov na mikrokrmilniku. S pomočjo AT ukazov lahko dobimo podatke o lokaciji, pošljemo sporočilo, preberemo dobljeno sporočilo ali pokličemo določeno številko.

```

#include <SoftwareSerial.h>
SoftwareSerial mySerial(2, 3); // RX, TX

int8_t answer;
int onModulePin= 13;
int counter;
long previous;
int x=0;
char Basic_str[500];

void setup()
{
    Serial.begin(9600);
    mySerial.begin(9600);
}

```

Slika 24: Uvodne nastavitve za BK-SIM808

```
void loop()
{
    if(x==0)
    {

        Serial.println("Starting....");
        mySerial.println("AT+CGNSPWR=1");
        delay(100);
        mySerial.println("AT+CGNSTST=1");
        mySerial.println();

        delay(5000);

        if (mySerial.available())
            Serial.write(mySerial.read());
        if (Serial.available())
            mySerial.write(Serial.read());

        while( mySerial.available() > 0) mySerial.read();
        delay(100);
        counter = 0;
        answer = 0;
        memset(Basic_str, '\0', 100);
        previous = millis();
        do{

            if(mySerial.available() != 0){
                Basic_str[counter] = mySerial.read();
                counter++;

                if (strstr(Basic_str, "OK") != NULL)
                {
                    answer = 1;
                }
            }
        }
```

Slika 25: Prekinitveni del program

Na začetku dodamo prekinitveni program, ki prepreči izvajanje programa, če se nekaj ne odziva pravilno. Ta del lahko vidimo na sliki 25.

```

}while((answer == 0) && ((millis() - previous) < 2000));

Basic_str[counter-3] = '\0';

Serial.println("*****");
Serial.print("Basic string: ");
Serial.println(Basic_str);

Serial.println("Starting....");
mySerial.println("AT+CGNPFWR=0");
Serial.println("AT+CGNPFWR=0");
delay(100);
mySerial.println("AT+CGNSTST=0");
Serial.println("AT+CGNSTST=0");
mySerial.println();

delay(2000);
Serial.println("sms Starting....");

mySerial.print("AT+CMGF=1\r");
delay(100);
mySerial.println("AT+CMGS=\"38630285360\"");
delay(100);
mySerial.println(Basic_str);
delay(100);
mySerial.write(0x1A);
delay(100);
mySerial.println();

delay(5000);
Serial.println("send sms ok");

x=10;
}
}

```

Slika 26: Pridobivanje lokacije in pošiljanje sporočil

Na sliki 26 je program, ki pridobi GPS lokacijo, nato pa jo pošlje preko sms sporočil. Z ukazom »mySerial.println« lahko z programsko kodo vstavljamo AT ukaze. Ta program se izvede vsakič, ko je eden izmed treh pogojev izpolnjen¹.

Kodo za preverjanje in sprejemanje sporočil pa je bilo težje poiskati. Našli smo program, ki ni namenjen temu vezju, ampak večjemu vezju, v osrčju katerega se nahaja isti modul. Ker modula še nimamo, smo kodo vstavili v program, vendar ne vemo ali bo delovala ali ne. Pogojni stavek, ki preverja ali je na modul prejel kakšno sporočilo je na sliki 27. Program na sliki sporočilo tudi izpiše. Mi bomo uporabljali samo stavek s katerim modul preverja prejetje sporočil.

¹ Glej poglavje 5.5


```
if(gsm.readSMS(smsbuffer, 160, n, 20))  
{  
    Serial.println(n);  
    Serial.println(smsbuffer);  
}
```

Slika 27: Branje sporočil

6.5 Glavni program

Program je sestavljen iz treh glavnih pogojnih stavkov, ki ob izpolnitvi pogoja izvedejo dobivanje lokacije in pošiljanje sporočila².

Prvi teh stavkov je na sliki 28. Tukaj imamo štiri pogojne stavke. Najprej mora biti vključeno stikalo za merjenje utripa. Nato mora senzor dotika zaznati dotik. Šele ko bo senzor kapacitivnosti zaznal kožo bo začel merilec srčnega utripa meriti utrip. Z ukazom »if(pulseSensor.sawStartOfBeat« preverjamo ali je senzor zaznal kakšen utrip.

```
x=0;
if (digitalRead(gumb)==HIGH) {

    if(digitalRead(dotik) == HIGH){
        long start = millis();
        long totall = cs_4_2.capacitiveSensor(30);

        delay(10);

        if(totall>5){
            if (pulseSensor.sawStartOfBeat())
                delay(300);
        }
        else{
            if (pulseSensor.sawStartOfBeat()){
                delay(300);
            }
        }
        else{
```

Slika 28: Prvi pogojni stavki

Drugi pogojni stavek je le preverjanje tipke – vidimo ga lahko na sliki 29.

```
if (digitalRead(tipka) == HIGH){
```

Slika 29: Pogojni stavek za tipko

Tretji pogojni stavek preverja če je modul prejel kakšno sporočilo. Stavek, ki preverja prejeta sporočila je viden na sliki 30.

² Glej poglavje 6.4.5

```
if(gsm.readSMS(smsbuffer, 160, n, 20)){
```

Slika 30: Preverjanje prejetih sporočil

Če bi pustili program takšen, kot je, bi ob prvem prejetem sporočilu pri vsakem preverjanju sporočila zaznal prejeto sporočilo, zato moramo sporočila iz modula zbrisati na koncu tretjega pogojnega stavka. Ta del programa je viden na sliki 31.

```
    mySerial.println( "AT+CMGF=1" );  
    //delay(1000);  
    mySerial.print("AT+CMGDA=\"");  
    mySerial.println("DEL ALL\"");  
    ,
```

Slika

31:

Brisanje

sporočil

7 ZAKLJUČEK

V tem inovacijskem predlogu sem se spoznal z veliko novimi senzorji, naučil pa sem se tudi zelo veliko o serijski komunikaciji s pomočjo AT ukazov. Predlog in izpeljava je bila zahtevna, saj sem se kot elektrotehnik prvič zares spoznal s težjimi in zahtevnejšimi oblikami programiranja, vendar sem z idejo in inovacijami na že obstoječih rešitvah zelo zadovoljen.

Rešitev je preprosta vendar efektivna. Opazil sem pa slabost pri merilcu srčnega utripa saj so meritve pogosto nenatančne, pri konstantnem merjenju pa se baterija porabi veliko hitreje. Zato sem se odločil, da bom v zaščitnico vstavili gumb, ki bo možnost branja srčnega utripa izključil popolnoma. Izkopi se tudi merjenje senzorja kapacitivnosti in senzorja dotika.

Izdelek zaradi težav z dobavitelji, ni izdelan zato moram – le tega še »pripeljati k življenju«. Verjamemo, da nam bomo izdelek oz. inovacijski predlog uspešno izpeljali do konca, s projektom pa se želim podati tudi v podjetniške vode.

8 DRUŽBENA ODGOVORNOST

Zapestnica lahko reši veliko ljudi. Zaradi zapestnice bodo lahko starejši ljudje živeli doma, kar pa bo preprečilo trganje čustvenih vezi in jim bo dalo občutek samostojnosti. Ugotovili smo namreč, da se marsikatera starejša oseba ob odhodu v dom ostarelih počuti, kakor da ni zmožna opravljati osnovnih opravil ali skrbeti zase – v marsikaterem primeru temu ni tako.

Med izdelovanjem naloge sem ugotovil tudi, da lahko zapestnico uporabi veliko večji spekter ljudi, kot sem uvodno mislil. Zraven starejših ljudi jo lahko uporabljajo:

- Otroci,
- Ljudje, ki imajo težave s spominom,
- Planinci,
- Smučarji,
- Avanturisti,
- Hišni ljubljenci (odstranimo pas),
- Varnostne službe itd.

Zapestnica ponuja zelo majhno vrednost elektromagnetnega onesnaževanja, če pa ne uporabljamo merilca srčnega utripa pa lahko baterija zdrži zelo dolgo.

Reši lahko odnose med družinami, otrokom lahko omogoči varno svobodo, v zasebnost ljudi pa v nasprotju s pametnimi urami ne vdira.

Tehnologijo, ki sem jo raziskoval v tem inovacijskem predlogu lahko uporabimo še na drugih področjih življenja. Zamislil sem si tudi lokator, ki pošlje lokacijo le ob prejetem sporočilu in deluje tudi kot prenosna baterija. Dodamo ga lahko v kovčke, to nam zagotavlja njihovo varnost, istočasno pa nam ponuja energijo na letališču. Naredil, bi lahko lokator za avto, ki se priključi v 12 V napajanje na avtu. To ga polni in napaja, zato ne potrebujemo dodatnega napajanja, deluje pa tudi kot pretvornik v USB polnilec s katerim lahko polnimo telefone ali druge elektronske naprave. Ustvarili bi lahko tudi lokator za hišne ljubljence, ki bi se pripel na
ovratnico.

9 VIRI

9.1 Spletne strani:

<https://forum.arduino.cc/index.php?topic=404814.0> 12.02.2018

<http://goqr.me/> 12.02.2018

https://wiki.eprolabs.com/index.php?title=Arduino_Pro_Mini 12.02.2018

<https://www.arduino.cc/en/Guide/ArduinoProMini> 12.02.2018

http://wiki.seeed.cc/Mini_GSM_GPRS_GPS_Breakout_SIM808/ 12.02.2018

<http://forum.arduino.cc/index.php?topic=319193.0> 12.02.2018

<https://forum.mysensors.org/topic/341/arduino-mini-pro-5v-how-to-power/2> 12.02.2018

https://en.wikipedia.org/wiki/Lithium_polymer_battery 12.02.2018

https://sl.wikipedia.org/wiki/Litij-ionska_baterija 12.02.2018

<https://www.arduino.cc/en/Tutorial/Button> 12.02.2018

<https://playground.arduino.cc/Main/CapacitiveSensor?from=Main.CapSense> 12.02.2018

<http://www.instructables.com/id/Capacitive-Sensing-for-Dummies/> 12.02.2018

<http://playground.arduino.cc/Main/CapacitiveSensor?from=Main.CapSense> 12.02.2018

<https://pulsesensor.com/pages/pulse-sensor-amped-arduino-v1dot1> 12.02.2018

<https://pulsesensor.com/pages/code-and-guide> 12.02.2018

<http://www.raviyp.com/embedded/140-learn-how-a-heart-beat-sensor-works> 12.02.2018

<https://www.elprocus.com/heartbeat-sensor-working-application/> 12.02.2018

<http://forum.arduino.cc/index.php?topic=422433.0> 12.02.2018

<https://www.dronethusiast.com/ultimate-drone-battery-care/> 12.02.2018

<http://forum.arduino.cc/index.php?topic=86611.0> 12.02.2018

https://www.itead.cc/wiki/SIM808_GSM/GPRS/GPS_Module 12.02.2018

https://www.elecrow.com/wiki/index.php?title=SIM808_GPRS/GSM%2BGPS_Shield_v1.0
12.02.2018

9.2 Video viri

<https://www.youtube.com/watch?v=-okAX7ZoGDk&index=1&list=PLfGV0MulN3W9kwvE1Glawn9T4i7-i7YO1> 12.02.2018

<https://www.youtube.com/watch?v=QS6ntRbCOTM> 12.02.2018

<https://www.youtube.com/watch?v=dvjGIo4gQc> 12.02.2018

https://www.youtube.com/watch?v=BHQPqQ_5ulc 12.02.2018

9.3 Dokumenti na spletu

https://cdn-shop.adafruit.com/product-files/2637/SIM800+Series+Embedded+AT+Sleep+Application+Note_V1.01.pdf 12.02.2018

<https://www.eecs.yorku.ca/~jr/res/m/MD/PulseSensor.pdf> 12.02.2018

https://cdn-shop.adafruit.com/product-files/2637/SIM800+Series_AT+Command+Manual_V1.09.pdf 12.02.2018

<https://media.digikey.com/pdf/Data%20Sheets/Pulse%20PDFs/PulseSensorAmpedGettingStartedGuide.pdf> 12.02.2018

https://cdn-shop.adafruit.com/datasheets/SIM808_Hardware+Design_V1.00.pdf 12.02.2018

<https://cdn.instructables.com/ORIG/FAO/80RU/IXLALERK/FAO80RUIXLALERK.pdf>
12.02.2018

9.4 Viri slik

Slika 1:

<https://image.made-in-china.com/43f34j00cKStqpBnMezO/Mini-GPS-Tracker-Watch-for-Kids-and-Elderly-GSM-Monitor-Sos-Phone-Call.jpg>

Slika 2:

https://www.google.si/search?safe=off&rlz=1C1GGRV_enSI752SI753&biw=1600&bih=745&tbm=isch&sa=1&ei=8CN_WrHGGoLaU5LxvOAH&q=sos+gsm&oq=sos+gsm&gs_l=psy-ab.3..0i30k1.41911.42236.0.42539.3.3.0.0.0.84.84.1.1.0....0...1c.1.64.psy-ab..2.1.83....0.Hhpoy2L7glg#imgdii=1K4-l7qbnlfP9M:&imgsrc=bhE3npfWBmLJOM:

Slika 3:

https://www.google.si/search?safe=off&rlz=1C1GGRV_enSI752SI753&biw=1600&bih=745&tbm=isch&sa=1&ei=8CN_WrHGGoLaU5LxvOAH&q=sos+gsm&oq=sos+gsm&gs_l=psy-ab.3..0i30k1.41911.42236.0.42539.3.3.0.0.0.84.84.1.1.0....0...1c.1.64.psy-ab..2.1.83....0.Hhpoy2L7glg#imgdii=PeMN8jQV4bPktM:&imgsrc=nCfKx7TQkJFzoM:

Slika 4:

Avtor naloge

Slika 5:

https://www.google.si/search?safe=off&rlz=1C1GGRV_enSI752SI753&biw=1600&bih=745&tbm=isch&sa=1&ei=2kJ_WuKgEcKYsAffh4SQBA&q=arduino+promini&oq=arduino+promini&gs_l=psy-ab.3..0i19k1j0i10i19k1j0i19k1.632182.636350.0.636828.15.13.0.0.0.300.1165.0j1j3j1.5.0...0...1c.1.64.psy-ab..10.5.1165...0j0i30k1.0.3xZeD_1ml8U#imgsrc=mmstmmOXIy32SM:

Slika 6:

https://www.google.si/search?q=programming+pro+mini+arduino&safe=off&rlz=1C1GGRV_enSI752SI753&source=lnms&tbn=isch&sa=X&ved=0ahUKEwj045z9hpzZAhXC8qQKHZe2DvwQ_AUICigB&biw=1600&bih=794#imgsrc=ZKCjYDo1_mgCnM:

Slika 7:

https://images-na.ssl-images-amazon.com/images/I/51pr%2BPSaANL_SL1100_.jpg

Slika 8:

https://images-na.ssl-images-amazon.com/images/I/61goWYOLeML_SL1001_.jpg

Slika 9:

https://images-na.ssl-images-amazon.com/images/I/51oH7J0rJsL_SL1024_.jpg

Slika 10:

<https://cdn.instructables.com/FIX/IGXH/I11YTWD0/FIXIGXH11YTWD0.LARGE.jpg>

Slika 11:

<https://www.glgprograms.it/immagini/elet/dispenseArduino/BottoneAnalisi.png>

Slika 12:

<https://www.arduino.cc/en/uploads/Tutorial/button.png>

Slika 13:

https://images-na.ssl-images-amazon.com/images/I/614%2Brmf38hL_SL1001_.jpg

Slika 14: https://images-na.ssl-images-amazon.com/images/I/81DbLzpiB5L_SL1200_.jpg

Slika 15: https://images-na.ssl-images-amazon.com/images/I/71FU1KkRGyL_SL1200_.jpg

Slika 16:

[https://api.qrserver.com/v1/create-qr-code/?data=Ime+in+priimek%0AVarnostni+kontakt+\(%C5%A1tevilka\)%0AAlergije%3A%0A-%0A-%0ABolezni%3A%0A-&size=220x220&margin=0](https://api.qrserver.com/v1/create-qr-code/?data=Ime+in+priimek%0AVarnostni+kontakt+(%C5%A1tevilka)%0AAlergije%3A%0A-%0A-%0ABolezni%3A%0A-&size=220x220&margin=0)

Slika 20: https://http2.mlstatic.com/sensor-touch-toque-capacitivo-fc-91-ttp223-arduino-pic-D_NQ_NP_120825-MLB25497429377_042017-F.jpg

Slika 17 – 19 in 21 - 31: Avtor naloge