



Fakultet elektrotehnike i računarstva, Sveučilišta u Zagrebu
Zavod za elektroničke sustave i obradbu informacija
Sveučilište u Zagrebu

Smart Stick

- Δ Dio projekta „Pametna kuća“
- Δ Arduino, senzori, programiranje
- Δ Detekcija prepreka
- Δ Odabir načina rada
- Δ Detekcija vode i svjetla
- Δ Korisnik na slušalicama prima sve signale iz sustava
- Δ Ringtone za pronalazak palice

16. lipnja



Sažetak

Smart Stick je dio projekta „Pametne kuće“ u sklopu kojeg je svaki student morao realizirati svoj vlastiti projekt. Na taj način smo uspjeli napraviti cijeli moderni sustav tako da smo podijelili kompleksan zadatak na niz podzadataka (lat. *Divide et impera*).

Pomoću „*Smart Stick-a*“, koji je namijenjen slijepim i slabovidnim osobama, korisniku će se uvelike olakšati kretanje kroz prostor, bilo unutarnji (e.g. kuća, stan ili zgrada), bilo vanjski (e.g. šetanje gradom ili parkom).

Iako već sada postoje neke implementacije sličnih sustava pametnih štapova za slijepe osobe u svijetu, nisam primijetio njihovu masovnu uporabu. Također, postojeće implementacije nisu baš jeftine, stoga se javlja potreba za implementacijom jeftinijeg, ali i učinkovitog sustava. Stoga će ovaj projekt pokušati što više pridonijeti popularizaciji i raširenosti tzv. „pametnih štapova“.

Prednosti „*Smart Stick-a*“ pred običnim palicama za slijepe su mnogobrojne. Korisniku se koristeći „*Smart Stick*“ pruža mnogo dodatnih funkcionalnosti, pored osnovne – detekcije prepreka. Palica može detektirati vodu na cesti, mjeriti osvjetljenje u prostoriji ili pomoći u lakšem pronalasku štapa.

Nedostaci „*Smart Stick-a*“ pred ostalim dostupnim pametnim štapovima na tržištu jest njegova konkretna realizacija. Naime, sustav je razvijen na protoboard pločici te da bi se mogao koristiti u realnom svijetu, potrebno je osmisliti kućište u koji će se sklop staviti ili čak razviti svoj *PCB* te time uštedjeti prostor. Također povezivanje preko slušalica jest žično, što postaje pomalo nezgrapno za korisnike koji se tek navikavaju na cijeli sustav. Višestruki ultrazvučni senzori (sonari) za detekciju prepreka iz više smjerova pomogli bi u većoj preciznosti i autonomnosti sustava za razliku od sadašnjeg jednog implementiranog sonara.

Koristi od ovog projekta, imat će sve slijepe i slabovidne osobe, kojima je otežano kretanje kroz prostor.

Sadržaj

1. UVOD	4
2. OPIS SUSTAVA „SMART STICK“	5
3. KOMPONENTE SUSTAVA „SMART STICK“	6
3.1. Arduino uno	6
3.2. Sonar HC-SR04	7
3.3. Slušalice	9
3.4. Zujalica	10
3.5. IR prijamnik	11
3.6. Fotootpornik	12
4. SPAJANJE SUSTAVA	13
5. REALIZACIJA SUSTAVA	14
6. PROGRAMSKI KOD	16
7. ZAKLJUČAK	22
8. LITERATURA	23
9. POJMOVNIK	23
9. POPIS SLIKA	24

Ovaj seminarski rad je izrađen u okviru predmeta „Sustavi za praćenje i vođenje procesa“ na Zavodu za elektroničke sustave i obradbu informacija, Fakulteta elektrotehnike i računarstva, Sveučilišta u Zagrebu.

Sadržaj ovog rada može se slobodno koristiti, umnožavati i distribuirati djelomično ili u cijelosti, uz uvjet da je uvijek naveden izvor dokumenta i autor, te da se time ne ostvaruje materijalna korist, a rezultirajuće djelo daje na korištenje pod istim ili sličnim ovakvim uvjetima.

1. Uvod

Iako tehnologija svakim danom sve više i više napreduje, znanstvenici i istraživači još uvijek nisu uspjeli pronaći način da vrate vid. Stoga je potrebno čim više olakšati njihov život, da mogu ipak malo mirnije i sigurnije provoditi život. To je moguće tako da im se poboljša mogućnost sigurnijeg ali i bezbrižnijeg kretanja, koje čini veliku prepreku u njihovom životu.

Ovaj rad, bazirao se na ideji, koji bi uz pomoć odgovarajućih ultrazvučnih senzora detektirao prepreku, te pomoću slušalica slao pravovremenu informaciju korisniku. To je osnovna funkcionalnost koja je implementirana, a mnogobrojne druge, dane su u popisu:

- Odabir između 2 načina rada
- Detekciji vode na cesti
- Detekciji upaljenog svjetla
- Primanje odgovarajućih obavijesti iz sustava na slušalice
- Sviranje melodije na primitak signala iz daljinskog upravljača za lakši pronalazak palice

U nastavku rada, biti će opisan cjelokupni sustav, od sheme spajanja do opisa programske potpore. Detaljnije ulaženje u kod neće biti opisano, iz tog razloga jer je već sam napisan kod dobro komentiran. To omogućava da se nakon nekog vremena korisnik koji je već radio na tom projektu brzo vrati „na stazu“ ili pak da novi korisnik koji želi nadograditi postojeći, brzo shvati što koja linija koda radi.

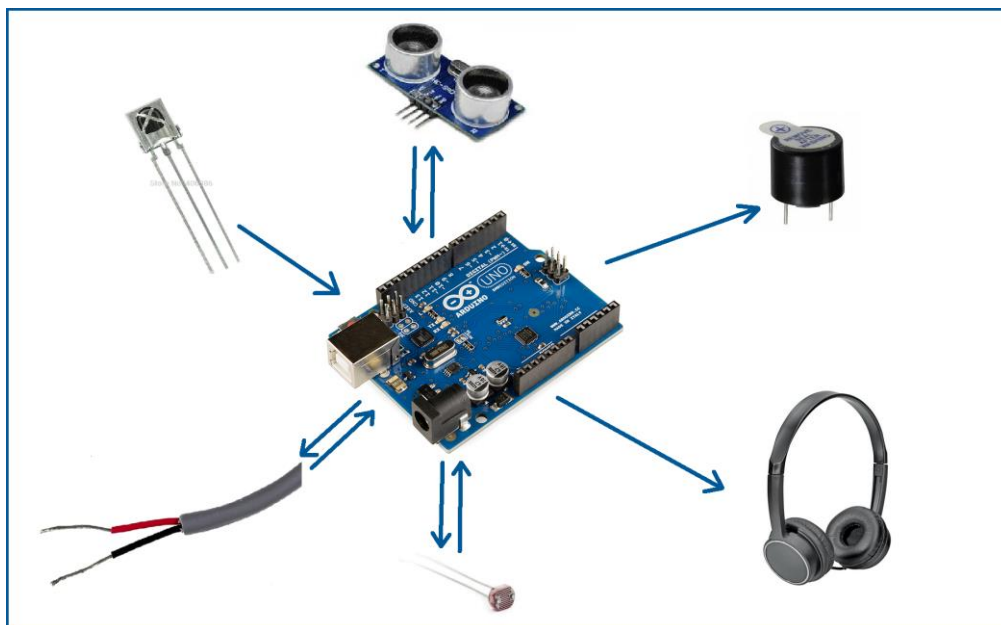
U samom sustavu ima mnogo mjesta za napredak, a najvažnije od njih definirane su u zaključku ovog rada.

2. Opis sustava „Smart stick“

Sustav „Smart stick“ koristiti će mnogobrojne komponente. Glavne komponente sustava su:

- Arduino Uno
- HC-SR04 sonar
- Slušalice
- Zujalica
- IR prijamnik
- Fotootpornik

Generalna shema sustava prikazana je na slici 1.



Slika 1. Shema sustava

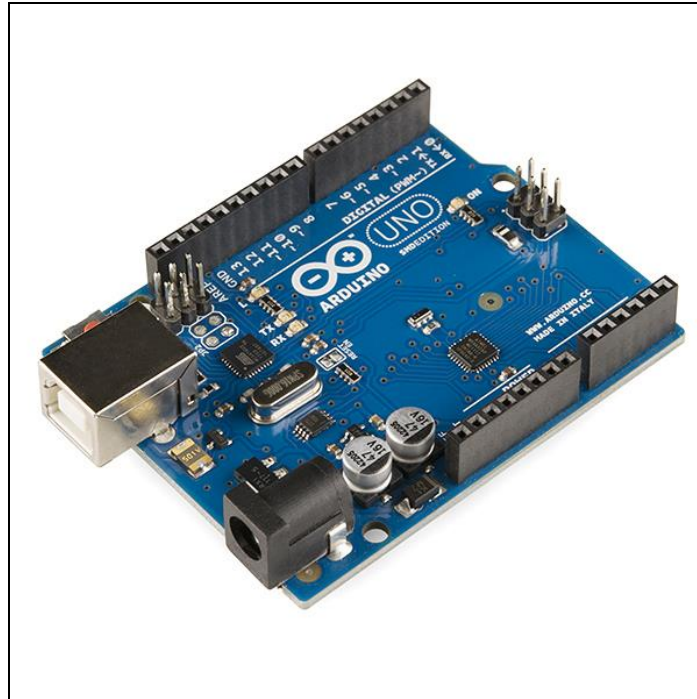
3. Komponente sustava „Smart stick“

Zbog relativne složenosti sustava korišteno je više komponenti. U nastavku su opisane korištene komponente kao i njihova funkcionalnost.

3.1. Arduino uno

Arduino Uno je mikrokontroler temeljen na Atmega328 mikroprocesoru. Ima 14 digitalnih ulazno/izlaznih pinova od kojih se 6 može koristiti kao PWM izlazi, 6 analognih ulaza, 16 MHz keramički rezonator, USB priključak, priključak za napajanje i gumb za reset. Vrlo je jednostavan za korištenje. Na računalo spaja se USB kabelom, a napaja se korištenjem samog USB kabela ili pak AC/DC adapterom ili baterijom. Ima veliku programsku podršku i mogu se naći mnogi primjeri gotovih projekata.

Za programiranje se koristi Arduino software koji je besplatan (Arduino IDE). Odlikuje ga mogućnost spajanja velikog broja modula kao što su Ethernet Modul, Xbee, RFID i slično. U ovom projektu Arduino Uno čini glavnu jezgru, na njega se spajaju senzori, zujalica i slušalice.



Slika 2. Arduino Uno

3.2. Sonar HC-SR04

Ovaj senzor omogućava beskontaktno mjerenje udaljenosti u rasponu od 2 cm do 400 cm uz preciznost mjerenja od oko 3 mm u idealnim uvjetima.

Modul se sastoji od ultrazvučnog odašiljača, ultrazvučnog prijemnika i kontrolne elektronike. Princip rada je sljedeći:

- modul se aktivira slanjem kontrolnog impulsa duljine najmanje $10\mu\text{S}$
- modul zatim automatski šalje osam ultrazvučnih impulsa frekvencije 40kHz
- kada detektira povratne ultrazvučne impulse, generira izlazni signal čija je dužina proporcionalna udaljenosti

Na modulu se nalaze sljedeći pinovi:

- pin 1: **VCC** - napajanje modula, 5V
- pin 2: **Trig** - okidanje/aktiviranje mjerenja
- pin 3: **Echo** - povratni signal, dužina impulsa proporcionalna udaljenosti
- pin 4: **GND**



Slika 3. HC-SR-04 ultrazvučni senzor - sonar

3.3. Slušalice

U ovom sustavu koriste se bilo koje slušalice koje imaju uobičajeni 3.5 mm utor (engl. *Jack*). Za implementaciju slušalica u sustav potreban nam je također i 3.55 mm utor koji će biti spojen jednom žicom na masu, a drugom na jedan od GPIO izlaza Arduina.



Slika 4. Slušalice

3.4. Zujalica

Koristit ćemo vrlo raširenu zujalicu AI-1223-TWT-5V-R (engl. *Buzzer*) koja će biti spojena na jedan od GPIO pinova Arduina.

Zujalica je spojena na napon od 5V, a troši struju od 30 mA. Frekvencijski raspon iznosi od 2000 do 2600 Hz. Za detaljnije specifikacije o zujalici pogledati na [1].

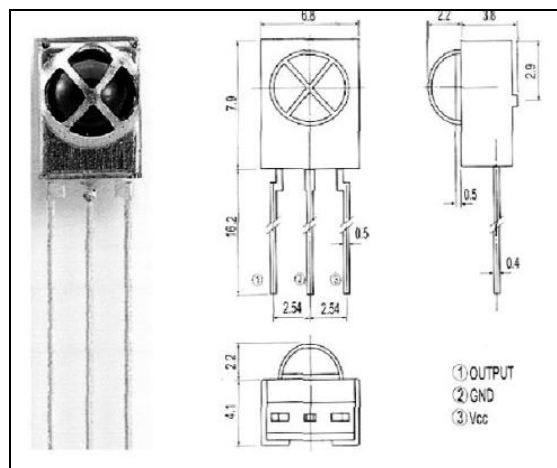


Slika 5. Zujalica

3.5. IR prijamnik

IR prijamnik (engl. *receiver*) služi nam da detektiramo pritisak gumba na daljinskom upravljaču. Koristiti će se standardni IR prijamnik AX-1838HS te daljinski upravljač. Obije komponente prikazane su na slici 6. i 7. redom.

Prije rukovanja s daljinskim upravljačem potrebno je proučiti odgovarajuće kodove signala, kako bismo mogli jasno identificirati koji gumb je korisnik pritisnuo.



Slika 6. IR prijamnik

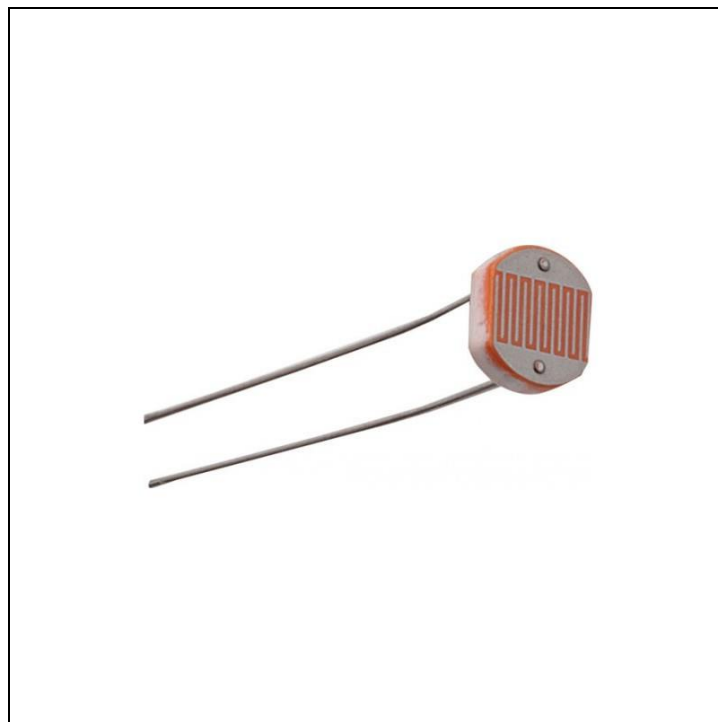


Slika 7. Daljinski upravljač

3.6. Fotootpornik

Korišten je *CdS* fotootpornik koji košta manje od 1\$. Promjenom intenziteta svjetla mijenja se i njegov otpor, čime se mijenja napon. Tako možemo efektivno mijeriti razinu osvjetljenja.

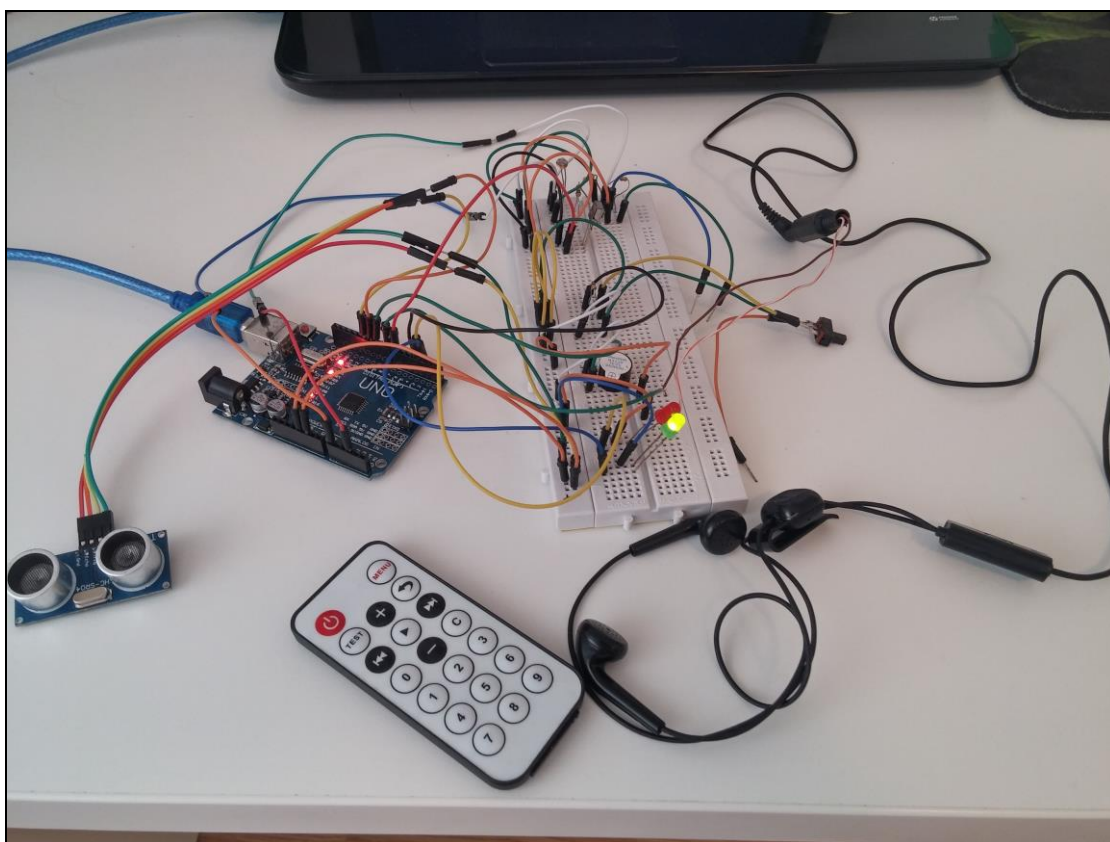
CdS fotootpornik koristi napon od 5V, a prije uključjenja mora biti spojena s otpornikom od 10 K Ω (engl. *Pull-down resistor*). Također navedeni fotootpornik mora biti spojen na analogni pin GPIO od Arduina.



Slika 8. *CdS* fotootpornik

4. Spajanje sustava

Način spajanja komponenti u sustav prikazan je na sljedećoj slici. Iako postoji mnogo konektora u sustavu, shema spajanja jest relativno jednostavna.



Slika 9. Spajanje komponenti u sustav

5. Realizacija sustava

U ovom projektu korisnik može birati između dva načina rada koristeći prekidač.

1. Prekidač gore – **INDOOR MODE** – *max.dist=100*
2. Prekidač dolje – **OUTDOOR MODE** - *max.dist=200*

Promjena na prekidaču provjerava se kod svakog prolaza glavne *loop()* petlja na Arduinu. Kad se način rada promijeni, korisnika se alarmira da je promijenio način rada na slušalice. 5 zvučnih signala kad mijenja mod iz INDOOR u OUTDOOR, a 10 suprotno. Također definirali smo maksimalne udaljenosti za pojedini mod rada.

Koristeći gotovu biblioteku <NewPing.h> koja radi s mnogim sensorima, a među kojima je i naš sonar HC-SR04, možemo efektivno dobivati podatke koristeći gotov API. Na početku našeg koda (*setup*) inicijaliziramo instancu (objekt) sonar te kasnije u glavnoj petlji (*loop*) koristimo gotove funkcije za dobivanje vrijednosti udaljenosti.

Prednosti korištenja API-a jest u manjem broju linija koda, manjoj mogućnosti pogreška, razumljivijem kodu te uštedi vremena prilikom razvoja projekta.

Alarmiranje korisnika se vrši tako da se s obzirom na trenutno stanje (INDOOR / OUTDOOR) provjeri udaljenost od prepreke. Ako je ta prepreka daleko, nema zvukova, ako je relativno blizu (< distance/3) korisniku se šalju zvučni signali sporije, a ako je dosta blizu tada se vrlo brzo šalju zvučni signali.

Kada korisnik stisne gumb na daljinskom upravljaču, generira se signal kojeg Arduino dekodira te poduzima odgovarajuće akcije. Koristeći biblioteku <IRremote.h> možemo koristiti gotov API te pozivati funkciju za obradu signala gdje ćemo pozvati odgovarajuću funkciju koristeći

statusni kod kao argument *switch case* upravljačke naredbe. Također na početku koda moramo definirati instancu *results* naše IR poruke.

Sviranje muzike omogućeno je korištenjem biblioteka `<melody.h>` i `<pitches.h>`. Prva datoteka jest zapravo globalno polje u kojem se nalaze tonovi za sviranje naše pjesme, a druga datoteka definira frekvenciju za odgovarajuću notu koji naš programski kod i koristi. U ovom sustavu implementirano je sviranje pjesme Super Mario, ali postoji širok spektar odabira pjesme na internet stranicama¹.

Korisnik pritiskom na gumb generira signal te njegov štap počinje svirati te na taj način korisnik može vrlo lako pronaći štap, u slučaju da ga je negdje zametnuo.

U slučaju da korisnik, koji je pod pretpostavkom slijepa ili slabovidna osoba, želi provjeriti je li svijetlo u kući upaljeno ili ne može jednostavno pritisnuti gumb na daljinskom te će zujalica odgovarajući broj puta odaslati zvučni signal. Broj zvučnih signala biti će jednak znamenki stotica trenutne vrijednosti osvjetljenja. Npr. neka je trenutna osvjetljenost 855² te korisnik pritisne gumb za mjerenje svjetlosti, zujalica će odaslati zvučni signal 8 puta.

Mjerenje vlage implementirano je jednostavnim definiranjem dviju žica od kojih se jedna spaja na napon (engl. *VCC*), a druga se spaja na analogni ulaz Arduina Una. Ako slučajno štap padne u vodu kad korisnik prelazi preko ceste, tada će se promijeniti napon između te dvije žice. Promjenom napona te definiranjem praga (engl. *threshold*) možemo alarmirati korisnika slanjem odgovarajućih zvučnih signala na slušalice da se ispred njega nalazi lokva.

¹ Zbog ograničene memorije Arduino Uno platforme, potrebno je obratiti pozornost na korištenu melodiju.

² Pretvorba iz analognih vrijednosti u LUX u ovom projektu nije implementirana.

6. Programski kod

Programski kod dan je na sljedećim stranicama. U detalje programskog koda neće se ulaziti jer su principi objašnjeni u prethodnom poglavlju, a svi potrebni komentari, nalaze se u samom kodu.

Datoteke Melody.h i Pitches.h ne navode se ovdje već se nalaze u .zip datoteci zajedno sa glavnim .ino kodom.

```
/* include library for HC-SR04 sonar */
#include <NewPing.h>

/* include library for IR signals */
#include "IRremote.h"

/* define notes */
#include "pitches.h"

/* Mario main theme melody - duration of notes */
#include "melody.h"

/* define pins */
#define TRIGGER_PIN 12 // Trigger PIN
#define ECHO_PIN 11 // Echo PIM
#define MAX_DISTANCE 300 // define max distance for Sonar
#define LED1 2 // green LED
#define LED2 3 // red LED
#define BUZZER 10 // buzzer PIN
#define HEADPHONES 8 // headphone PIN
#define INDOOR_PIN 4 // state PIN
#define NUMBER_STARTUP 3 // counter for loops in script
#define NUMBER_CHANGESTATE_INDOOR 5 // number of beeps for
changing state I.
#define NUMBER_CHANGESTATE_OUTDOOR 10 // number of beeps for
changing state II.
#define IRreceiver 7 // IR receiver PIN
#define ANALOG_PHOTORES 0 // photoresistor analog PIN
#define VAL_PROBE 3 // water level indicator PIN
```



```
//=====GLOBAL VARIABLES=====
/* set indoor and outdoor disance mode */
unsigned char mode_distance[] = {40, 70};
unsigned char state;          // state (0-indoor, 1-outdoor)
int pass=0;                  // varibale for loop count - when having
warning - put bigger delay
int photocellReading;        // light value
int moisture;                // moisture value
//=====GLOBAL VARIABLES =====

//=====DECLARE OBJECTS=====
IRrecv irrecv(IRreceiver); // create instance of 'IRrecv'

decode_results results;     // create instance of 'decode_results'

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // new instance
of 'sonar'
//=====DECLARE OBJECTS=====

//=====DECLARE FUNCTIONS=====

/* on state change play "var" number of beeps */
void play_change(int var){
  int i;
  digitalWrite(HEADPHONES, LOW);
  for(i=0;i<var;i++){
    buzz(HEADPHONES, 260, 80);
    delay(250);
  }
}

/* play N number of beeps, where N is light sensor value in hundereds
*/
void play_value(int var){
  int value=var/100;
  if(value==0){
    buzz(HEADPHONES, 260, 1000);
    delay(250);
  }else{
    for(int i=0;i<value;i++){
      buzz(HEADPHONES, 260, 80);
      delay(250);
    }
  }
}
}
```

```
/* read in what state is user */
int read_user_state(){
  if( digitalRead(INDOOR_PIN) == 1 )
    return 0;    //INDOOR STATE
  else
    return 1;    //OUTDOOR STATE
}

/* take action based by IR code received and take regular actions */
void IRsignal(){
  Serial.println("Play me some song please !");
  switch(results.value)    //check the IR signal received
  {
    case 0xFFA25D:{        // TURN ON button = find my stick

      Serial.println(" 'Mario Theme'");
      int size = sizeof(melody) / sizeof(int);
      for (int thisNote = 0; thisNote < size; thisNote++) {

        // to calculate the note duration, take one second
        // divided by the note type.
        // e.g. quarter note = 1000 / 4, eighth note = 1000/8, etc.
        int noteDuration = 1000/tempo[thisNote];

        buzz(BUZZER, melody[thisNote],noteDuration);

        // to distinguish the notes, set a minimum time between
        them.
        // the note's duration + 30% seems to work well:
        int pauseBetweenNotes = noteDuration * 1.30;
        delay(pauseBetweenNotes);

        // stop the tone playing:
        buzz(BUZZER, 0, noteDuration);
      }
      break;

    }

    case 0xFFE21D:{        // MENU button = what is light result
      photocellReading = analogRead(ANALOG_PHOTORES);
      Serial.print("Analog reading = ");
      Serial.println(photocellReading);
      play_value(photocellReading);    // call function for playing
      Light reading
      break;
    }
  }
}
```

```
    default:      // if user pushed some other button
        Serial.println(" other button");
    }
}

/* play the tone on targetPin with arguments as frequency and period
*/
void buzz(int targetPin, long frequency, long length) {

    long delayValue = 1000000/frequency/2;      // calculate the
    delay value between transitions
    Serial.println(delayValue);
    //// there are two phases to each cycle
    long numCycles = frequency * length/ 1000;  // calculate the
    number of cycles for proper timing
    //// multiply frequency, which is really cycles per second, by the
    number of seconds to
    //// get the total number of cycles to produce
    for (long i=0; i < numCycles; i++){        // for the calculated
    length of time...
        digitalWrite(targetPin,HIGH);        // write the buzzer pin
high to push out the diaphragm
        delayMicroseconds(delayValue);      // wait for the calculated
delay value
        digitalWrite(targetPin,LOW);        // write the buzzer pin low
to pull back the diaphragm
        delayMicroseconds(delayValue);      // wait again or the
calculated delay value
    }
}

/* main SETUP loop */
void setup() {
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
    pinMode(INDOOR_PIN, INPUT);
    pinMode(BUZZER, OUTPUT);
    pinMode(HEADPHONES, OUTPUT);
    state=read_user_state(); // read user state on startup
    play_change(NUMBER_STARTUP); // play sound on startup
    irrecv.enableIRIn(); // start the IR receiver
    Serial.begin(9600);
}
```

```
/* main loop */
void loop() {

    moisture = analogRead(VAL_PROBE);    // read moisture

    if(moisture<1000){                  //if moisture level is less than some
limit (1000), turn on the buzzer for 1 second
        digitalWrite(BUZZER, HIGH);
        delay(1000);
        digitalWrite(BUZZER, LOW);
        delay(1000);
    }

    if (irrecv.decode(&results))        // check if you received an IR
signal
    {
        Serial.println(results.value, HEX);    // print result code of
IF
        IRsignal();                          // process IRsignal
        irrecv.resume();                      // receive next value
        delay(200);
    }

    int udaljenost = sonar.ping_cm();      // get sonar output in cm

    if( state != read_user_state() ){     // check if the state has
changed
        Serial.println("User has changed his state!");
        state=read_user_state();
        if(state==0)
            play_change(NUMBER_CHANGESTATE_INDOOR);    // inform user about
current state, state = 0 -> 5 times on/off buzzer
        else
            play_change(NUMBER_CHANGESTATE_OUTDOOR); // inform user about
current state, state = 1 -> 10 times on/off buzzer
    }

    /* check if sonar is close to an object - warning */
    if ( udaljenost < mode_distance[state] && udaljenost!=0) {
        Serial.print("Warning, you're about to hit something !");
        /* set LED output */
        digitalWrite(LED1,LOW);
        digitalWrite(LED2,HIGH);

        /* if user is very close to object (d/3), than beep faster */
        if( udaljenost <= mode_distance[state] / 3){
```

```
        Serial.println("Big warning, you're surely gonna hit something!");
        buzz(HEADPHONES, 260, 40);
        /* else beep slower */
    }else if(pass%2){
        buzz(HEADPHONES, 260, 80);
        pass++;
    }else{
        digitalWrite(HEADPHONES, LOW);
        pass++;
    }

    /* if sonar is not close to an object, set LED output */
} else {
    digitalWrite(LED1,HIGH);
    digitalWrite(LED2,LOW);
    digitalWrite(HEADPHONES, LOW);
}

/* Print state, distance and moisture*/
Serial.print("Current state: ");
Serial.print(state);
Serial.print("    Current distance: ");
Serial.print(udaljenost);
Serial.print("cm");
Serial.print("    Current moisture: ");
Serial.println(moisture);

delay(200);    //refresh every 0.2 seconds = 200 ms
}
```

7. Zaključak

Navedeni projekt „Smart stick“ sigurno ima potencijala za konkretnu uporabu u stvarnom životu slijepa osobe. Pomoću „Smart sticka“ omogućuje se lakše snalaženje u prostoru slijepim i slabovidnim osobama. Pomoću ultrazvučnog senzora koji bi trebao biti postavljen na palici detektira se najbliža prepreka.

Ovaj sustav je vrlo jednostavan, te bi se za realnu primjenu trebalo provoditi daljnja testiranja. Kako su korišteni HC senzori vrlo jeftini, tako su i donekle neprecizni. Naime za maksimalnu sigurnost slijepih osoba potrebni su precizni i pouzdani senzori, stoga bi se moglo umjesto postojećih koristiti neke preciznije senzore.

U radu je riješena većina zadataka dana u početnoj projektnoj dokumentaciji. Mjesta za daljnji napredak jesu dodavanje još senzora u sustav za detektiranje prepreka iz više smjerova te sama fizička realizacija palice.

Složeniji budući rad na projektu mogao bi se vidjeti u dodavanju GPS modula te integracija s mobilnom mrežom. Na taj bi način obitelj mogla uvijek znati gdje se korisnik kreće te bi sustav mogao alarmirati obitelj ili hitnu putem sms poruke u slučaju da je korisnik pao. To se može detektirati ne-micanjem štapa, ne-prisutnosti ruke na palici kao i neuobičajeni položaj palice. Uz navedene komponente potrebno bi još bilo staviti IMU³ u sustav.

Nadam se da će ovaj rad potaknuti i druge da preuzmu ovaj projekt te ga nadgrade, jer jedna od prednosti dobro dokumentiranog projekta jest u tome da netko drugi može brzo preuzeti postojeći rad i nadograditi ga, za što mu treba nemjerljivo manje vremena nego pišući kod od nule.

³ https://en.wikipedia.org/wiki/Inertial_measurement_unit

8. Literatura

- [1] Arduino Tutorial. URL: <http://www.arduino.cc/> (2015-10-06)
- [2] Arduino Reference. URL: <http://arduino.cc/en/Reference/HomePage> (2015-10-06)
- [3] Arduino Duemilanove. URL: <http://www.arduino.cc/en/Main/ArduinoBoardDuemilanove> (2015-10-06)
- [4] Eckel T., NewPing Library for Arduino: <http://playground.arduino.cc/Code/NewPing> (2015-10-06)
- [5] Ultrasonic Ranging Module HC - SR04: <http://elecfreaks.com/store/download/HC-SR04.pdf>
- [6] ButtonState Change: <http://www.arduino.cc/en/Tutorial/ButtonStateChange> (2015-10-06)

9. Pojmovnik

Pojam	Kratko objašnjenje	Više informacija potražite na
White paper	Kratak dokument koji daje uvid u neko područje, tehniku, politiku, proizvod, metodu, standard i sl.	en.wikipedia.org/wiki/White_paper
Arduino	Arduino razvojna platforma	http://www.arduino.cc/
GPIO	Engl. General Purpose Input Output. Izvodi mikrokontrolera koji mogu biti ulazni i izlazni.	http://en.wikipedia.org/wiki/Generalpurpose_input/output
HC-SR04	Senzor udaljenosti	http://www.micropik.com/PDF/HCSR04.pdf
Zujalica	Zvučna indikacija.	http://en.wikipedia.org/wiki/Buzzer

9. Popis slika

Slika 1. Shema sustava	5
Slika 2. Arduino Uno	7
Slika 3. HC-SR-04 ultrazvučni senzor - sonar	8
Slika 4. Slušalice	9
Slika 5. Zujalica.....	10
Slika 6. IR prijamnik	11
Slika 7. Daljinski upravljač.....	11
Slika 8. CdS fotootpornik.....	12
Slika 9. Način spajanja komponenti u sustav.....	13