

B-GLASSES

MENTOR:
DALIBOR ĐUMIĆ.

UČENIK:
EMRAH PONJEVIC

SARAJEVO, juni, 2017.god.

1. Uvod

B-GLASSES predstavlja spoj ideje i znanja primjenjenog na elektroničkim komponentama s ciljem da se olakša ljudima koji imaju problem s vidom. To je izum koji zahtijeva poznavanje arduino platforme (u ovom slučaju arduino-nano platforma), ultrasoničnog senzora blizine, rada servo motora, primjena i svojstva zvučnih uređaja i programiranje arduino platforme. Može se slobodno reći da se uz malo elektroničkih komponenti, programerskog znanja i rada može napraviti ozbiljan elektronički uređaj, u ovom slučaju „Pomagalo za slijepe osobe“ .

Osnovni razlog pojave ovakvog uređaja je taj da se zamijeni tzv. „Štap za slijepe osobe“ koji je krajnje primitivan, što je pokazao kroz svoju dugogodišnju primjenu. Pomagalom za slijepe osobe, ti nedostaci bi se potpuno uklonili, korisnik bi bio opušteniji i njegova sigurnost bi bila na većem nivou.

2. B-GLASSES

Uređaj je prvobitno napravljen da uoči prepreke koje se mogu otkriti upotrebom štapa za slijepe osobe, međutim kako se „štap“ pokazao dosta primitivnim i nepraktičnim, morali su se otkloniti nedostaci koji su bili prisutni. Pomoću ultrasoničnog senzora blizine koji radi na principu ultrazvučnih valova koji se odbijaju od prepreke i servo motora čiji je osnovni zadatak pomjeranje senzora tj. uočavanje prepreke u određenom radijusu, što je ustvari bio i osnovni zadatak ovakvog uređaja. Arduino-nano je platforma koja upravlja cjelokupnim procesom tj. ona sadrži naredbe i upute na osnovu kojih radi uređaj. Zvuk je signal koji razumije osoba, a koji dolazi sa slušalice koje su povezane na arduino-nano platformu. Sav napon ovog uređaja obezbjeđuje obična telefonska baterija od 5V, koja je ustvari i dovoljna za rad ovakvog uređaja. Uređaj se sastoji od par elektorničkih komponenti, a one su:

1. arduino-nano platforma
2. servo motor
3. ultrasonični senzor blizine
4. baterija
5. jumperi i
6. slušalice.

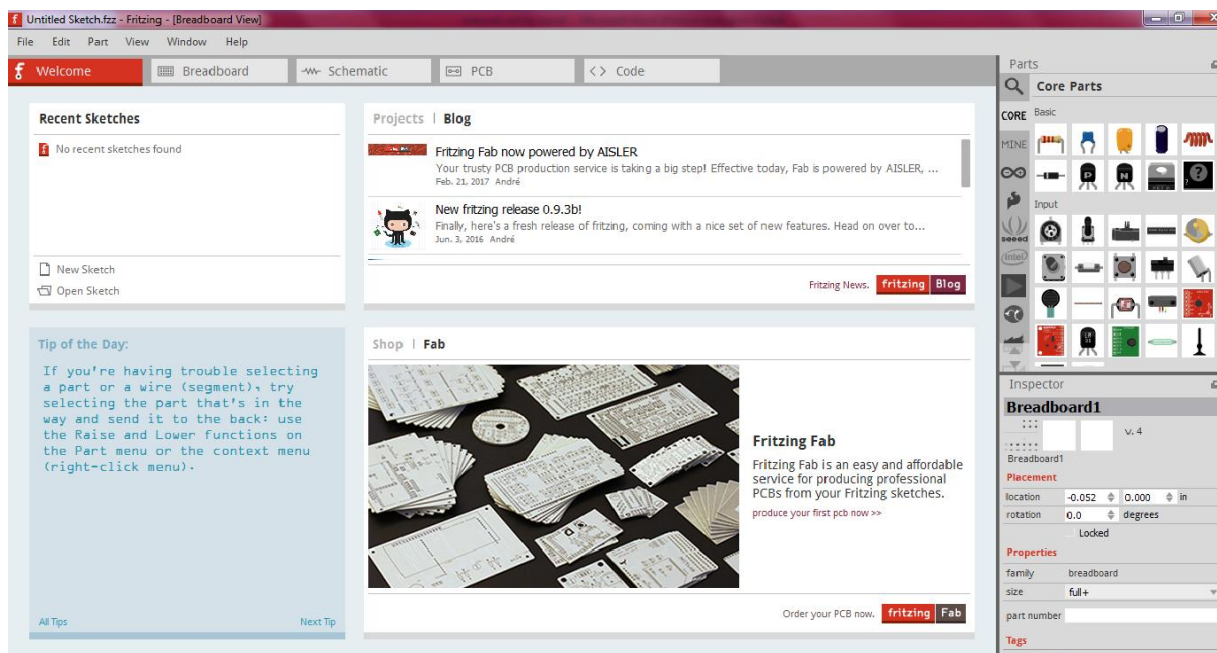
Svi elementi su povezani na digitalne ulaze arduino-nano platforme preko jumper-a (žica), dok su analogni ulazi ostali ne korišteni. Pomagalo za slijepe osobe je prvi ozbiljni uređaj napravljen za osobe koje imaju problema sa vidom.

3. Izrada B-GLASSES ili pomagala za slijepe osobe

Izrada pomagala za slijepe osobe zavisi od mnogo aspekata kao što je poznavanje hardvera i softvera i vještina koje osoba posjeduje. U ovom slučaju hardver su arduino-nano platforma, ultrasonični senzor blizine, baterija, jumperi, baterija i slušalice, dok softver predstavlja program koji će povezati sve ove komponente pri tome da one obavljaju svoju funkciju za koju su namjenjene. Vještine koje osoba mora poznavati pri izradi ovog projekta su najmanje bitne, a one su kreativnost i mirnoća, što ustvari ne igra toliku ulogu kao dva prethodna aspekta.

3.1. Spajanje komponenti i kreiranje sheme

Da bi pravilno hardverski povezali komponente i time one pravilno funkcionirale potrebno je prije toga, povezati ih u nekog programu čime bi se uklonile greške ili eventualna oštećenja komponenti za izradu ovog projekta. Neke greške koje se mogu javiti usred ne pažnje pri spajanju ovih komponenti su: pregaranje arduino-nano platforme koja ne podržava napon veći od 5V ili pregaranje slušalica koje se ne mogu spojiti na arduino bez nekog otpornika srednje otpornosti jer on ustvari smanjuje napon koji se dovodi u slušalice i time sprječava njihovo pregaranje. Za softversko spajanje komponenti i provjeru njihovog rada postoji više programa kao što su: ZenitPCB, FreePCB, TinyCad, Osmond PCB, BSch3V, Kicad, gEDA, Fritzing ili DesignSpark PCB, ali u ovom slučaju koristit ću program Fritzing zbog poznavanja rada i efikasnosti koju nudi. Program možete besplatno skinuti na linku: <http://fritzing.org/download/>.



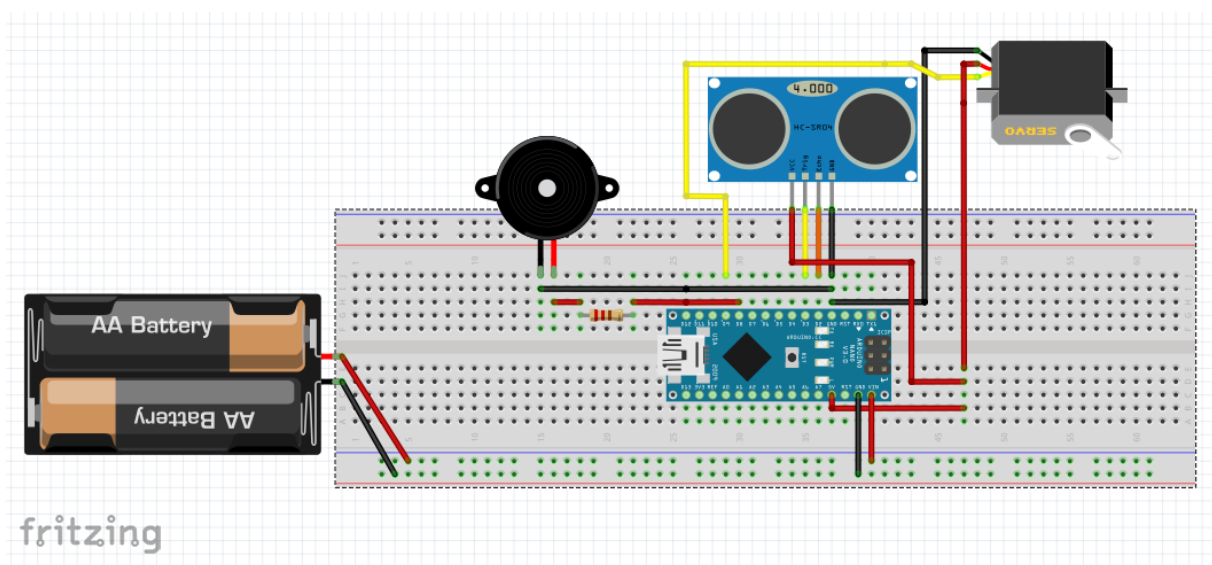
Slika 3.1.1 Fritzing [Breadboard View]

U nastavku pokrećemo opciju File i opciju New gdje počinjemo sa radom i kreiranjem shemi za projekat „Pomagalo za slijepe osobe“. Sve komponente možemo pronaći u desnom uglu pod nazivom „Core Part“ ili u slučaju da se ne nalazi neka naša komponenta možemo

izvršiti download sa Fritzing-ove oficialne stranice i jednostavno je kopirati i zalijepiti u folder gdje se nalaze i ostale komponente. Komponente koje su potrebne za spajanje su:

1. Arduino-nano razvojna platforma
2. Ultrasonični senzor blizine HC-SR04
3. Servo motor
4. Buzzer
5. Otpornik (100 Ω ili 220 Ω)
6. Baterija
7. Matador i
8. Jumper žice.

Nakon što smo dodali sve komponente i spojili ih na pinove koje smo zadali u programskom kodu, naša shema bi trebala izgledati kao na slici 3.1.2.



Slika 3.1.2. Fritzing shema spajanja komponenti

NAPOMENA: Kod spajanja komponenti u matadori treba naglasiti da je umjesto slušalica korišten buzzer, dok je kod eksterne baterije korištena obična baterija. Razlog tome je taj što Fritzing program ne sadrži slusalice i eksterne baterija kao svoje komponente, ali je princip spajanja i rada jednih i drugih isti pa je suvišno praviti neke nepostojeće razlike.

3.2. Programiranje Arduino-nano razvojne platforme

Programiranje Arduino-nano razvojne platforme se ne razlikuje od drugih platformi kao što je Uno i sl. Princip programiranja je isti jer je u pitanju samo veličina platforme. Kod samog programiranja potrebno je napisati ispravno kod, odabrati port na koji ćemo poslati kod i eventualno uključiti neke biblioteke koje je potrebno uključiti. Kod na temu „Pomagalo za slijepe osobe“ za moj maturski sa komentarima je sljedeći:

```
#define NOTE_B4 494
#define NOTE_C5 523
#define NOTE_CS5 554
#define NOTE_D5 587
#define NOTE_DS5 622
#define NOTE_E5 659
#define NOTE_F5 698
#define NOTE_FS5 740
#define NOTE_G5 784
#define NOTE_GS5 831
#define NOTE_A5 880
#define NOTE_AS5 932
#define NOTE_B5 988

#include <Servo.h> // uključi biblioteku za servo motor

Servo myServo; // kreiraj servo varijablu za kontrolu servo motora
int echopin = 2; // spoji echopin na digitalni pin 2
int trigpin = 3; // spoji trigpin na pin digitalni 3
int buzzerPin = 8; // spoji buzzerPin na digitalni pin 8
int cm; // varijable kreirana za dužinu koja će biti u centimetrima (cm)
int i = 0; // varijabla za pohranu pozicije servo motora

void setup(){
  pinMode(3, OUTPUT); // definiši pin 3 za izlaz
  pinMode(2, INPUT); // definiši pin 2 za ulaz
  pinMode(8, OUTPUT); // definiši pin 8 za izlaz
  myServo.attach(9); // spoji varijablu servo motora na pin 9
  myServo.write(0); // postavi poziciju servo motora na 0 stepeni
  delay(1000); // kašnjenje od 1 sekunde
}
void loop(){
  for (i=30; i<=90; i +=1){ // idi od pozicije 30 stepeni do pozicije 90 stepeni
    myServo.write(i); // reci servo motoru da ide na poziciju smještenu u varijabli 'i'
    cm = calculateDistance(); // poziva funkciju koja računa udaljenost mjerenu pomoću
    ultrasoničnog senzora za svaki stepen za koji se pomjeri servo motor
    delay(45); // čekaj 45ms dok servo motor ne dođe na poziciju 'i'
    if(cm>150){
      noTone(8); // ne sviraj ništa ako je udaljenost veća od 150 cm
    }
    else if((cm<=150)&&(cm>140)){
      tone( NOTE_B4, 10); // pusti notu B4 ako je udaljenost manja ili jednaka od 150 cm i
      veća od 140 cm
    }
  }
}
```

```

    }
    else if((cm<=140)&&(cm>130)){
        tone(NOTE_C5, 10); // generiši ton C5 ako je udaljenost manja ili jednaka 140 cm i veća
od 130 cm
    }
    else if((cm<=130)&&(cm>120)){
        tone(NOTE_CS5, 10); // pusti notu CS5 ako je udaljenost manja ili jednaka 130 cm i
veća od 120 cm
    }
    else if((cm<=120)&&(cm>110)){
        tone(NOTE_D5, 10); // pusti notu D5 ako je udaljenost manja ili jednaka 120 cm i veća
od 110 cm
    }
    else if((cm<=110)&&(cm>100)){
        tone(NOTE_DS5, 10); // pusti notu DS5 ako je udaljenost manja ili jednaka 110 cm i
veća od 100 cm
    }
    else if((cm<=100)&&(cm>90)){
        tone(NOTE_E5, 10); //pusti notu E5 ako je udaljenost manja ili jednaka 100 cm i veća
od 90 cm
    }
    else if((cm<=90)&&(cm>80)){
        tone(NOTE_F5 , 10); // pusti notu F5 ako je udaljenost manja ili jednaka 90 cm i veća
od 80 cm
    }
    else if((cm<=80)&&(cm>70)){
        tone(NOTE_FS5, 10); // pusti notu FS5 ako je udaljenost manja ili jednaka 80 cm i
veća od 70 cm
    }
    else if((cm<=70)&&(cm>60)){
        tone( NOTE_G5, 10); // pusti notu G5 ako je udaljenost manja ili jednaka 70 cm i veća
od 60 cm
    }
    else if((cm<=60)&&(cm>50)){
        tone(NOTE_GS5, 10); //pusti notu GS5 ako je udaljenost manja ili jednaka 60 cm i
veća od 50 cm
    }
    else if((cm<=50)&&(cm>40)){
        tone( NOTE_A5, 10); // pusti notu A5 ako je udaljenost manja ili jednaka 50 cm i veća
od 40 cm
    }
    else if((cm<=40)&&(cm>30)){
        tone(NOTE_AS5, 10); // pusti notu AS5 ako je udaljenost manja ili jednaka 40 cm i
veća od 30 cm
    }
    else if(cm<=30){
        tone(NOTE_B5, 10); // pusti notu B5 ako je udaljenost manja ili jednaka 30 cm
    }
}
for (i=90; i>30; i -=1){ // idi od pozicije 90 stepeni do pozicije 30 stepeni

```



```

myServo.write(i); // reci servo motoru da ide na poziciju smješenu u varijabli 'i'
cm = calculateDistance(); // poziva funkciju koja računa udaljenost mjerenu pomoću
ultrasoničnog senzora za svaki stepen za koji se pomjeri servo motor
delay(45); // čekaj 45ms dok servo motor ne dođe na poziciju 'i'
  if(cm>150){
    noTone(8); // ne sviraj ništa ako je udaljenost veća od 150 cm
  }
  else if((cm<=150)&&(cm>140)){
    tone( NOTE_B4, 10); // pusti notu B4 ako je udaljenost manja ili jednaka od 150 cm i
veća od 140 cm
  }
  else if((cm<=140)&&(cm>130)){
    tone(NOTE_C5, 10); // pusti notu C5 ako je udaljenost manja ili jednaka 140 cm i veća
od 130 cm
  }
  else if((cm<=130)&&(cm>120)){
    tone(NOTE_CS5, 10); // pusti notu CS5 ako je udaljenost manja ili jednaka 130 cm i
veća od 120 cm
  }
  else if((cm<=120)&&(cm>110)){
    tone(NOTE_D5, 10); // pusti notu D5 ako je udaljenost manja ili jednaka 120 cm i veća
od 110 cm
  }
  else if((cm<=110)&&(cm>100)){
    tone(NOTE_DS5, 10); // pusti notu DS5 ako je udaljenost manja ili jednaka 110 cm i
veća od 100 cm
  }
  else if((cm<=100)&&(cm>90)){
    tone(NOTE_E5, 10); //pusti notu E5 ako je udaljenost manja ili jednaka 100 cm i veća
od 90 cm
  }
  else if((cm<=90)&&(cm>80)){
    tone(NOTE_F5 , 10); // pusti notu F5 ako je udaljenost manja ili jednaka 90 cm i veća
od 80 cm
  }
  else if((cm<=80)&&(cm>70)){
    tone(NOTE_FS5, 10); // pusti notu FS5 ako je udaljenost manja ili jednaka 80 cm i
veća od 70 cm
  }
  else if((cm<=70)&&(cm>60)){
    tone( NOTE_G5, 10); // pusti notu G5 ako je udaljenost manja ili jednaka 70 cm i veća
od 60 cm
  }
  else if((cm<=60)&&(cm>50)){
    tone(NOTE_GS5, 10); //pusti notu GS5 ako je udaljenost manja ili jednaka 60 cm i
veća od 50 cm
  }
  else if((cm<=50)&&(cm>40)){
    tone( NOTE_A5, 10); // pusti notu A5 ako je udaljenost manja ili jednaka 50 cm i veća
od 40 cm

```

```

    }
    else if((cm<=40)&&(cm>30)){
        tone(NOTE_AS5, 10); // pusti notu AS5 ako je udaljenost manja ili jednaka 40 cm i
veća od 30 cm
    }
    else if(cm<=30){
        tone(NOTE_B5, 10); // pusti notu B5 ako je udaljenost manja ili jednaka 30 cm
    }
    }
}
// funkcija za izračunavanje udaljenosti mjerene ultrazvučnim senzorom blizine
int calculateDistance(){
    digitalWrite(3, LOW);
    delayMicroseconds(2);
// postavi trigPin u stanje nisko na 3 ms
    digitalWrite(3, HIGH);
    delayMicroseconds(10);
// postavi trigPin u stanje visoko na 10 ms
    digitalWrite(3, LOW); //postavi trigPin u stanje nisko
    duration = pulseIn(2, HIGH); // čita signale dobivene sa echoPin-a i vraća dužinu trajanja
zvučnog talasa u milisekundama
    cm= duration*0.034/2; // formula za izračunavanje udaljenosti
    return cm; // vrati cm u void loop funkciju
}

```

