



Univerzitet Singidunum
Tehnički Fakultet

Arduino prototip „*Lightsaber*“, mača

- Projektni rad –

Predmet: Programabilni hardver i ugrađeni sistemi

Profesor:

Petar Spalević

Predmetni asistent:

Petar Jakić

Student:

Stefan Dimitrijević 2018200497

Sadržaj

1. Uvod.....	1
2. Šema i specifikacija projekta.....	1
3. Princip rada.....	3
4. Izrada koda koji pokreće svu logiku.....	3
5. Linkovi.....	7

1. Uvod

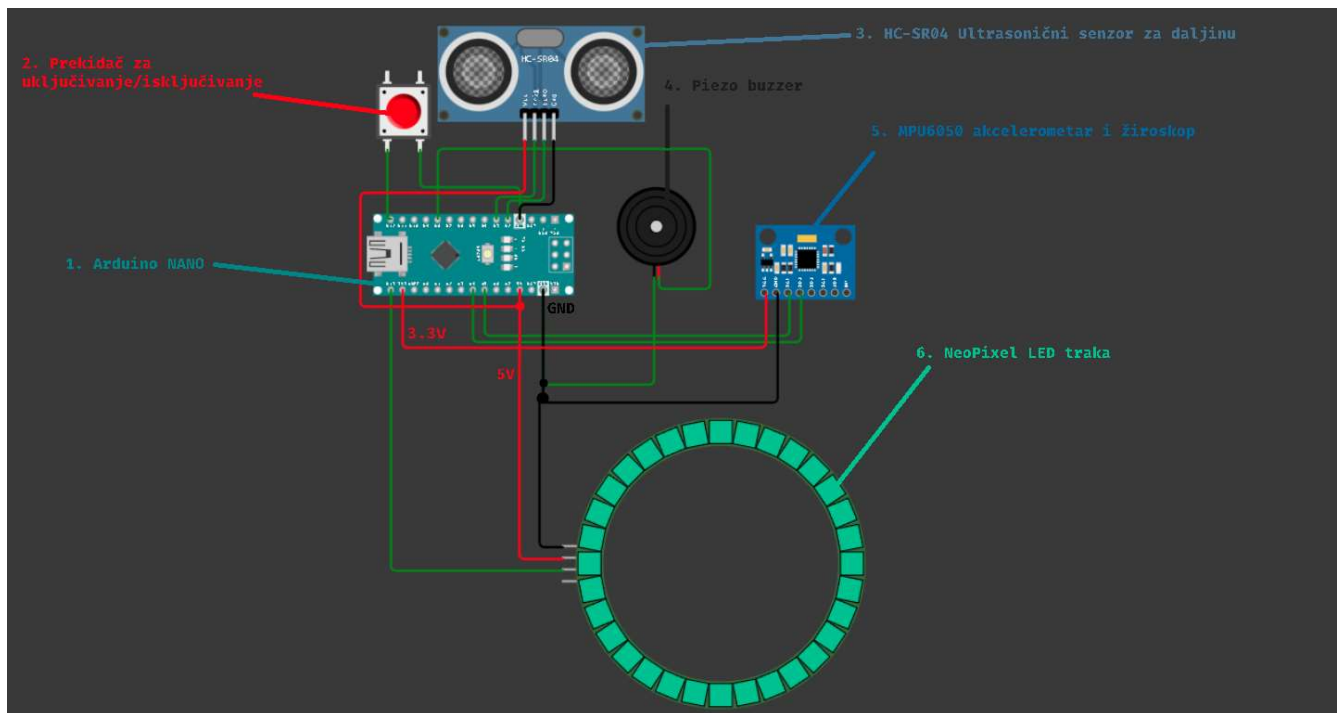
Projekat koji je opisan u ovom dokumentu osmišljen je da što je bolje simulira poznati mač iz serijala *Star Wars* – tzv. „lightsaber,,.

Suština projekta je sledeća: potrebno je bilo osmisлити sklop koji bi, koristeći *Arduino* familiju ploča(projekat je realizovan *Arduino NANO* pločom), u simulatoru oponašao(koliko je moguće) mač iz filma. Potrebno je imati jedno dugme(prekidač) kojim bi se mač uključivao i isključivao(*toggle* princip). Takođe, potrebno bi bilo simulirati kretanje mača u prostoru za potrebe oponašanja različitih zvukova – za šta su iskorišćeni *akcelerator* i *žiroskop*. Prilikom „dodira,, mača sa bilo kakvim objektom, trebalo bi to registrovati u vidu promene boje mača(iskorišćena je bela boja), kao i reprodukcije drugačijeg, višefrekventnog zvuka.

Za izradu projekta korišćen je **wokwi.com** online besplatni simulator koji sadrži sve neophodne komponente za realizaciju.

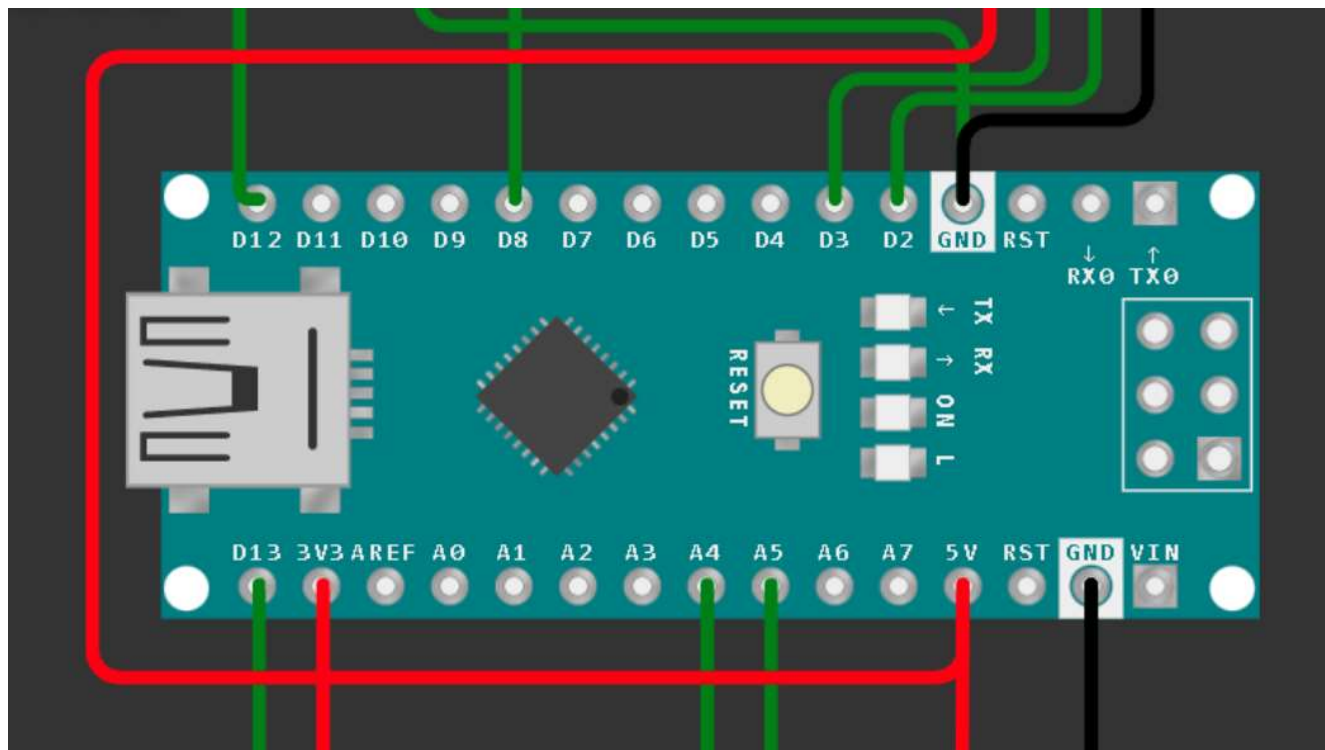
2. Šema i specifikacija projekta

Na sledećim slikama biće prikazana i opisana šema kao i specifikacija projekta koju šema oslikava.



Slika 1. Šema kola

Na sledećoj slici nalaziće se uvećana *Arduino NANO* ploča radi boljeg prikaza pinova.



Slika 2. Prikaz pinova na samoj *Arduino NANO* ploči

Počecemo od prvog elementa, koji u suštini predstavlja naš mikrokontroler – *Arduino* ploča. Ova ploča se koristi da uveže sve elemente putem svojih magistrala kao i da ih napaja, u realnosti bi se koristilo eksterno napajanje(baterija), ali simulator putem pinova **3V3** i **5V** dovodi napone od 3,3 V i 5 V respektivno na aktivne komponente.

Sledeći element (broj 2) je crveni taster kojim ćemo uključivati, odnosno isključivati naš sklop; povezan je na ploču na sledeći način: kontakt **1.1** na prekidaču vezali smo za digitalni ulaz **D12** na ploči, dok je kontakt **2.1** vezan za uzemljenje, odnosno *Ground*(**GND**).

Element broj 3. je ultrasonični senzor koji detektuje koliko je neki objekat blizu. Ima 4 pina, **Vcc** koji je naponski pin i vezuje se na napon od 5 V, standardno mora imati i pin za uzemljenje(crna žica) i ide na pin **GND**. Preostala dva pina **TRIG** i **ECHO** vezuju se na digitalne pinove **D3** i **D2** respektivno, gde će pin **D3** biti pin za *Output*(sa aspekta mikrokontrolera), dok će pin **D2** služiti kao ulazni(*Input*) pin i koji će *feedovati* mikrokontroler svojim očitavanjima.

Sledeći je **Piezo Buzzer** element, jednostavni zvučnik koji će reprodukovati zvuke određene *frekvencije*. Ima svega 2 kontakta: crni koji se vezulje za nulu i crveni koji se vezuje na digitalni pin mikrokontrolera **D4**. Putem ovog pina će mikrokontroler slati zvuk za reprodukciju.

Pretposlednji element, tačnije element broj 5 je čip koji na sebi ima *akcelerometar* i *žiroskop* koje ćemo kontrolisati da detektujemo *da li* se „mač,, pomera(simulira se pokret), u kom *smeru* i kojim *intenzitetom*, pa će se ova dva parametra iskoristiti da promene zvuk koji se reprodukuje. Ova komponenta ima 4 pina; standardno 2 za samo napajanje(koristi napajanje od 3,3 V) i još 2 *analogna* kontakta za *serijsku(I2C)* komunikaciju sa mikrokontrolerom- **SCL** (vezana za analogni pin **A5**) je linija na koju se dovodi *clock* za sinhronizaciju primanja, odnosno slanja podataka, kao i **SDA** (vezana za analogni pin **A4**) linija kojom se šalju, odnosno primaju podaci.

Poslednji preostali element je **LED** traka koja ima 4 kontakata – od kojih se koriste samo 3, 2 su za napajanje(koristi napon od 5 V) i **DIN** kontakt putem kojeg **MCU** šalje traci podatke, vezana na **D13**.

3. Princip rada

Princip rada je sledeći: Pritiskom na crveni taster i puštanjem aktiviraju se sve navedene komponente. *Buzzer* zuji frekvencijom od 50 Hz i promenom parametara na **MPU6050** kolu menja se ova učestanost, te samim tim zvuk je drugačiji i simulira se kao da neko „pomera,, mač u prostoru. Istovremeno, aktivna je i *LED* traka koja svetli cijan bojom, a zahvaljujući i **ultrasoničnom senzoru**, ukoliko se detektuje „kolizija,,(tako što se pomera slajder na senzoru) sa nekim objektom(ukoliko je taj objekat udaljen manje od 5 cm), *LED* traka svetli belom bojom, a frekvencija zvuka koju bazer emituje je znatno viša da simulira da se sablja „sudarila,, sa nečim u prostoru.

Ponovnim pritiskom crvenog tastera isključuju se sve komponente.

4. Izrada koda koji pokreće svu logiku

Pre svega, potrebno je proći kroz deklaracije neophodnih promenljivih, konstanti i funkcija koje se koriste

```
1  #include <Adafruit_MPU6050.h>
2  #include <Adafruit_Sensor.h>
3  #include <Adafruit_NeoPixel.h>
4  #ifdef __AVR__
5  | #include <avr/power.h> // Required for 16 MHz Adafruit Trinket
6  #endif
7  #include <Wire.h>
8
9  //MPU6050 akcelerometar i žiroskopski senzor
10 Adafruit_MPU6050 mpu;
11
12 #define PIXEL_PIN 13 // Digitalni pin na koji se vezuje LED traka(Data IN)
13
14 #define PIXEL_COUNT 32 // Broj dioda na traci(definisan broj u diagram.json)
15
16 // Deklaracija Neostrip objekta
17 Adafruit_NeoPixel strip(PIXEL_COUNT, PIXEL_PIN, NEO_GRB + NEO_KHZ800);
18 // Argument 1 = Broj "pixela" (individualnih dioda) na traci
19 // Argument 2 = Arduino pin #
20 // Argument 3 = flegovi koji opisuju tip LED dioda:
21 // NEO_KHZ800 800 KHz bitstream (most NeoPixel products w/WS2812 LEDs)
22 // NEO_KHZ400 400 KHz (classic 'v1' (not v2) FLORA pixels, WS2811 drivers)
23 // NEO_GRB Pixels are wired for GRB bitstream (most NeoPixel products)
24 // NEO_RGB Pixels are wired for RGB bitstream (v1 FLORA pixels, not v2)
25 // NEO_RGBW Pixels are wired for RGBW bitstream (NeoPixel RGBW products)
26
27 //pinovi za tastere - povezani su na pinovima 12 i 4
28 #define BUTTON_PIN_12 12
29
30 //definiseemo na kom pinu su ECHO i TRIG
31 #define ECHO_PIN 2 //digitalna dvojka
32 #define TRIG_PIN 3 //digitalna trojka
33
```

Slika 3. Promenljive

Biblioteke koje su neophodne za rad su *AdaFruit* biblioteke za rad sa **MPU6050** jedinicom, senzorima, **NeoPixel LED** trakom I sa samim povezivanjem.

Deklarišemo promenljivu *mpu* pomoću koje ćemo pristupiti *event*-ima koje generiše **MPU6050** uređaj da bismo odredili merenja ovog uređaja, takođe deklariramo **LED** traku pod nazivom „*strip*“, koja uzima 3 parametra: broj pojedinačnih dioda na traci, pin na kojem je traka povezana na kućištu ploče i **režim rada** (režimi su dati kao komentari ispod promenljive).

Koristeći pretprocesorske direktive deklariramo konstante koje ćemo (ili već na ovoj slici koristimo) za inicijalizaciju komponenti. Primera radi **PIXEL_COUNT**, **PIXEL_PIN** su konstante koje smo deklarirali pretprocesorskim direktivama da bi ih koristili u kodu da inicijalizujemo **LED** traku. Takođe imamo konstante koje govore da je dugme na pinu 12, **ECHO** kontakt na pinu **D2** i **TRIG** kontakt na pinu **D3**.

```
59 //dogadjaji koje detektuje senzor
60 sensors_event_t event;
61 boolean is_turned_on = false;
62 boolean was_high = false;
63
64 void colorWipe(uint32_t color, int wait); // funkcija koja boji sve diode specificiranom bojom
65 int mySpecialNumber( sensors_event_t a,sensors_event_t g,sensors_event_t temp); // "magicni" broj
66 float readDistanceCM(); //cita distancu objekta od ultrasoničnog senzora distance u santimetrima
```

Slika 4. Promenljive i funkcije za rad

Na 60, 61 i 62. liniji koda vidimo još 3 promenljive, prva predstavlja prethodno pomenute *evente* koje generiše **MPU6050** i koje ćemo koristiti da dohvatimo izmerene vrednosti, sledeće dve promenljive su logičkog tipa i koriste se za uključenje ostalih komponenti putem crvenog tastera.

ColorWipe funkcija uzima boju u rasponu od [0 – 16,777,215] i boji celu **LED** traku tom bojom, što takođe možemo specificirati korišćenjem funkcije **strip.Color(red, green, blue)**, gde sva 3 parametra uzimaju vrednosti od [0 – 255], odnosno sveukupno 256 vrednosti, pa kada bi smo pomnožili 256*256*256 dobili bismo tačno 16,777,215 unikatnih boja.

Funkcija *mySpecialNumber* generiše broj na osnovu parametara koje kolo **MPU6050** vrati. I konačno funkcija *readDistanceCM* vraća distancu u santimetrima od samog ultrasoničnog distancometra.

Pre nego što se pokrene glavna *loop* funkcija u *Arduinu*, koja se odvija iznova i iznova(u beskonačnost), prvo se poziva funkcija *setup* koja, kao što joj ime kaže, za cilj ima da pruži početnu konfiguraciju komponenti.

```

35 void setup() {
36     // put your setup code here, to run once:
37     //koristimo serijski port za komunikaciju sa baud rate-om 115200 (jer akcelerometar+ziroskop rade u ovom rezimu)
38     Serial.begin(115200);
39
40     while (!mpu.begin()) {
41         Serial.println("MPU6050 not connected!");
42         delay(1000);
43     }
44     Serial.println("MPU6050 ready!");
45
46     //digitalni pin 4 i 12 su tasteri
47     pinMode(BUTTON_PIN_12, INPUT_PULLUP);
48
49     //Digitalni pin 3 je OUTPUT
50     pinMode(TRIG_PIN, OUTPUT);
51     //Digitalni pin 2 je INPUT
52     pinMode(ECHO_PIN, INPUT);
53
54     strip.begin(); // Inicijalizacija NeoStrip LED trake
55     strip.show(); // ovim zapravo renderujemo osvetljenje(posto ga nismo podesili bice black u pocetku)
56
57 }

```

Slika 5. *Setup* funkcija

Pošto **MPU6050** radi u serijskom **I2C** modu, potrebno je inicijalizovati **Serijski** port i odabrati mu *Baud rate* – broj bitova u sekundi prenosa/prijema (to je kod nas 115200 *BR*).

Zatim se inicijalizuje pin digitalni 12 da koristi *Pullup* otpornik, pomoću kojeg ćemo registrovati taster. Takođe inicijalizuju se pinovi **TRIG** kao *Output* na digitalnom pinu **D3** i **ECHO** pin kao *Input* na digitalnom pinu **D2**.

Inicijalizuje se i **LED** traka sa *begin* funkcijom i nakon nje *show* funkcijom koja renderuje samo diode „crnom,, bojom(**OFF**).

I konačno dolazimo do glavne funkcije – *loop* koja se izvršava u beskonačnost:

```
68 void loop() {
69     // put your main code here, to run repeatedly:
70
71     //citamo DIGITALNU vrednost sa dugmeta koje je vezano na PIN 12
72     int value = digitalRead((BUTTON_PIN_12));
73
74     //citamo distancu sa proximity senzora
75     float distance = readDistanceCM();
76     //ukoliko je nesto na manje od 5 cm od senzora, detektujemo ga kao true
77     bool isNearby = distance < 5;
78
79     //Dohvata acceleration(a), gyroscope(g)
80     sensors_event_t a, g, temp;
81     mpu.getEvent(&a, &g, &temp);
82
83     if(value == HIGH)
84         was_high = true;
85
86     if(value == LOW && was_high){
87         was_high = false;
88         is_turned_on = !is_turned_on;
89     }
90
91     //opseg boja [0-16,777,215] brojcano
92     if(is_turned_on){
93         colorWipe(
94             isNearby ?
95             strip.Color(255,255,255):2277000);
96
97         //frekventni opseg [50, 125]
98         tone( 8,
99             isNearby ?
100             250:mySpecialNumber(a,g, temp),
101             250);
102     }else{
103         //ako je uredjaj ugasen boja je "crna", a ton se ne cuje
104         colorWipe( strip.Color(0,0,0));
105         tone( 8, 0, 250);
106     }
107
108
109     delay(10);
110
111 }
```

Slika 6. *Loop* funkcija

Prvo se iščitava vrednost sa digitalnog pina na kojem je povezan taster, zatim se iščitava distanca objekta od uređaja, pa se setuje promenljiva *isNearby* na *true* ako je objekat manje od 5 cm udaljen od senzora. Nakon toga, koristeći *event* senzora **MPU6050** iščitavaju se vrednosti za ubrzanje, rotaciju i temperaturu (koja se ne koristi).

Ukoliko je iščitana vrednost **HIGH** to znači da je znak da se pokrene ceo ovaj sklop, u suprotnom se isključuje.

Ukoliko je uključen, svira se određeni ton na osnovu ubrzanja i rotacije sa prethodno pomenutog senzora, i diode se boje u *cijan* boju. Ukoliko je pak sklop i **uključen**, i objekat je u blizini, tada se svira drugačiji zvuk a diode su bele boje.

U suprotnom ako nije uključen sklop diode se boje „crnom,, bojom, a zvuk koji se svira ima učestanost 0 – odnosno nema zvuka niti uključenih dioda.

5. Linkovi

[1] Link ka projektu: [sketch.ino – Wokwi Arduino Simulator -
https://wokwi.com/arduino/projects/316612696608866880](https://wokwi.com/arduino/projects/316612696608866880)