```
/* -------------------------------------------------------------------------------------------------------------------------------------------------
*                                                                    SpideySense                 __
*                                                                                               / ‾l
* Ver. 1.9                                                                                    .'   :                  __.....__..._  _____
* Datum: 25. decembar 2016.                                                                  /  /   \           _.-" $$SSSSSSS$$SSSSSSSSSp.
*                                                                                           (`-: .qqp:    .--.'  .p.S$$$$SSSSS$$$$$$$$SSSSp.
* Opis:                                                                                      """yS$$SSSb,.'.g._\.SSSSS^^""       `S""^^$$$SSSb.
*                                                                                              :SS$S$$$$SSSSS^"""-. _.            `.   "^$$$SSb._.._
*       Upotrebom ultrazvučnog senzora za Arduino Mini Pro, ostvarujemo upotrebu               SSS$$S$$SSP^/       `.              \     "^$SSS$$SSb.
* novog čula! Ultrazvučnim senzorom utvrđujemo udaljenost i, proporcionalno toj                :SSSS$SP^" :          \  `-.            `-      "^TSS$$SSb
* vrednosti u cm, aktiviramo mali vibrirajući motor odgovarajućim intezitetom.                  $$$$S'    ;          db                ."        TSSSSSS$,
* Cela sprava montirana je u rukavicu tako da je senzor na strani dlana, a motor                :$$P      ;$b        $ ;    (        /   .-"   .S$$$$$$$;
* na poleđini istog. Variranjem parametara opisanih u komentarima koda, moguće je                 ;-"     :$ ^s.    d' $           .g. .'    .SSdSSSS$P"
* menjati ponašanje uređaja, sve po potrebama i ukusu korisnika.                                 /     .-  T.  `b. 't._$ .- / __.-j$'.'   .sSSSdP^^^'
*                                                                                               /  /      `,T._.dP   "":'  /-"   .'       TSSP'
*                                                                                              :  :        ,\""        ; .'    .'      .-""
*                                                                                             _J  ;        ; `.      /.'    _/    \.-"
* Šema za povezivanje ultrazvučnog senzora:                                                  /  "-:       /"--.b-..-'      .'       ;
*                                                                                           /     /  ""-..'           .--'.-'/  ,  :
*                                                                                          :S.   :     dS$ nik         `-i" ,',_:  _ \
*     Arduino | HC-SR04                                                                    :S$b  '._  dS$;             .'.-"; ; ; j `.l
*    -------------------                                                                    TS$b          "-._          `"  :_/ :_/
*       5V    |   VCC                                                                        `T$b              "-._
*       A2    |   Trig                                                                         :S$p._               "-.
*       A1    |   Echo                                                                          `TSSS$ "-.     )       `.
*       GND   |   GND                                                                              ""^--""^-. :           \
*                                                                                                           ";            \
*                                                                                                           :              `._
*                                                                                                           ; /    \ `._     ""---.
* Autor:                                                                                                   / /     _        `.--.__.'
* Nikola Stojanović                                                                                       : :   / ;  :".  \
*                                                                                                         ; ;  :  :  ;   `.  `.
*                                                                                                        /  ;  :   ; :    `.  `.
* Licenca:                                                                                              /  /:  ;   :  ;      "-'
* Public Domain                                                                                        :_.' ;  ;    ; :
*                                                                                                          /  /     :_l
*                                                                                                          `-'
* -------------------------------------------------------------------------------------------------------------------------------------------------*/
#include <NewPing.h>                                        // Uzimanje u obzir biblioteke NewPing (https://github.com/PaulStoffregen/NewPing).

#define LED 13                                              // Definisanje upotrebe LED na samoj pločici koja je fabrički vezana na pin 13.
#define TRIGGER_PIN  A2                                     // Arduino pin koji je povezan na Trig pin ultrazvučnog senzora.
#define ECHO_PIN     A1                                     // Arduino pin koji je povezan na Echo pin ultrazvučnog senzora.
#define MAX_DISTANCE 330                                    // Maksimalna udaljenost od senzora koju prihvatamo. Senzor može da detektuje
                                                            // udaljenost do 400-500cm.
int dist_old;                                               // Promenljiva koja beleži stari rezultat merenja udaljenosti.
int counter_out;                                            // Promenljiva koja beleži broj puta kada je merenje bilo van zadatog opsega.
int counter_in;                                             // Promenljiva koja beleži broj puta kada je merenje bilo unutar zadatog opsega.
int vibr_old;                                               // Promenljiva koja beleži PWM vrednost za pin koji kontroliše intenzitet rada motora.
int vibr_PWM = 11;                                          // Povezivanje promenljive vibr_PWM sa pinom 11.

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);         // Definisanje vrednosti iz NewPing biblioteke
                                                            // (pin gde je vezan Trig, pin gde je vezan Echo, maksimalna udaljenost u cm).


void setup() {

  Serial.begin(115200);                                     // Pokretanje Serial monitor-a, 115200 baud, u svrhu prikazivanja rezultata.
  pinMode(vibr_PWM, OUTPUT);                                // Definišemo pin 11 (vibr_PWM) kao output.

}

void loop() {

 int dist_measured;                                         // Promenljiva koja beleži sirovi rezultat merenja udaljenosti.
 int dist_new;                                              // Promenljiva koja beleži novu vrednost udaljenosti (nakon usrednjavanja).
 int vibr_raw;                                              // Promenljiva koja beleži sirovu vrednost za vibraciju (nakon mapiranja).
 int vibr_new;                                              // Promenljiva koja beleži novu vrednost za vibraciju (nakon usrednjavanja).
 int min_dist;                                              // Promenljiva koja beleži minimalnu prihvaćenu udaljenost.
 int volt_sensor = analogRead(A7);                          // Promenljiva u koju upisujemo očitavanje na analognom pinu A7 sa baterije.
 float volt_value = volt_sensor * (5.00 / 1023);            // Promenljiva u koju upisujemo vrednost volt_sensor konvertovanu u volte.

 delay(13);                                                 // Pauziraj 30ms između pingova (oko 30 pingova u sekundi). 29ms bi trebalo
                                                            // da bude najmanja vrednost ovde.

if (volt_value > 2.90) {                                    // Ako je napon očitan na bateriji veći od 2.90V, nastavljamo sa ostatkom programa.
                                                            // Poenta ovde je da zaštitimo bateriju od preteranog pražnjenja.

 dist_measured = sonar.ping_cm();                           // Slanje pinga, merenje vremena neophodnog da se vrati (Echo), računanje
                                                            // udaljenosti u cm i smeštanje te vrednosti u  promenljivu dist_measured.
                                                            // U slučaju da je vrednost veća od one definisane promenjivom MAX_DISTANCE,
```

```
                                                    // funkcija vraća vrednost 0.
                                                    // U nastavku koda vršimo uslovljavanje parametara i obradu izmerenih podataka.

if (dist_measured != 0) {                           // Ukoliko vrednost dist_measured nije jednaka 0,
  counter_in = counter_in + 1;                      // povećavamo vrednost counter_in za jedan.

     if (counter_in > 1) {                          // Ako se ovo desilo barem 2 puta,
        counter_out = 0;                            // resetujemo vrednost brojača counter_out na 0,
        dist_new = (dist_measured+dist_old)/2;      // vršimo usrednjavanje vrednosti za udaljenost i zapisujemo je u dist_new,
        dist_old = dist_new;                        // i konačno, vraćamo tu vrednost u dist_old.
    }
 }

 else {                                             // U suprotnom, ako dist_measured jeste jednaka 0,
    counter_out = counter_out + 1;                  // povećavamo vrednost brojača counter_out za 1.
     if(counter_out > 30) {                         // Potom, ako se to desilo 31 ili više puta uzastopno,
      counter_in = 0;                               // resetujemo vrednost brojača counter_in na vrednost 0,
      dist_old = (dist_old + MAX_DISTANCE)/2;       // postavljamo udaljenost na maksimalnu vrednost na aritmetičku sredinu
     }                                              // MAX_DISTANCE i prethodne vrednosti dist_old, u cilju glatkijeg prelaza.
     else {                                         // Ako se to desilo 30 ili manje puta,
      dist_old = dist_old;                          // Ostavljamo promenljivu dist_old na prethodnoj vrednosti.
   }

  }


 min_dist = 25;                                     // Ovde je definisana vrednost za najmanju udaljenost koju želimo da uzmemo u obzir.

  if(dist_old < min_dist) {                         // Ako je vrednost dist_old manja od dist_min,
  analogWrite(vibr_PWM, 0);                         // izlaz na pinu vibr_PWM postaje 0.
  }

 else {                                             // U suprotnom,
  vibr_raw = map(dist_old, 25.00, 310.00, 250.00, 10.00);  // Mapiramo vrednosti iz skupa dist_old [25, MAX_DISTANCE] u novi skup vibr_raw [250, 0].
  vibr_new = (vibr_raw+vibr_old)/2;                 // Usrednjavamo vrednost za signal za motor (vibr_raw)
  vibr_old = vibr_new;                              // i beležimo je u vibr_new.
  analogWrite(vibr_PWM, vibr_old);                  // Šaljemo signal preko pina vibr_PWM (pin 11), vrednosti vibr_old.
  }

  if (dist_old < min_dist || counter_out > 10) {    // U slučaju da je dist_old manja od definisane min udaljenosti ili ako je
     analogWrite(vibr_PWM, 0);                      // broj merenja udaljenosti van opsega (counter_out) veći od 10, LED na pinu 13.
     digitalWrite(LED, LOW);                        // palimo i gasimo u pulsirajućem obrascu.
     delay(300);
     digitalWrite(LED, HIGH);
     delayMicroseconds(1000);
     digitalWrite(LED, LOW);
     delay(150);
     digitalWrite(LED, HIGH);
     delayMicroseconds(1000);
   }
  else {                                            // U suprotnom, prilikom bilo kog uspešnog merenja u definisanom opsegu,
     digitalWrite(LED, HIGH);                       // palimo LED na pinu 13.
   }


                                                    // Ispisujemo sve željene rezultate preko Serial monitora.
 Serial.print("Ping: ");
 Serial.print(dist_old);
 Serial.print(" cm");
 Serial.print("  _____  Vibrator Voltage: ");
 if(dist_old < min_dist) {
   Serial.print("0.00");
  }
 else {
  Serial.print(vibr_old/51.00);
  }
 Serial.print(" V");
 Serial.print("  _____  Battery voltage: ");
 Serial.print(volt_value);
 Serial.println(" V");
 }

else {                                              // Ako je napon na bateriji manji od ili jednak 2.90V, gasimo vibrator i LED na pinu 13.
     analogWrite(vibr_PWM, 0);
     digitalWrite(LED, LOW);
     Serial.print("Battery voltage: ");
     Serial.print(volt_value);
     Serial.println(" V");
 }
}
```