

Meranie vzdialenosti ultrazvukovým senzorom

Pomôcky:

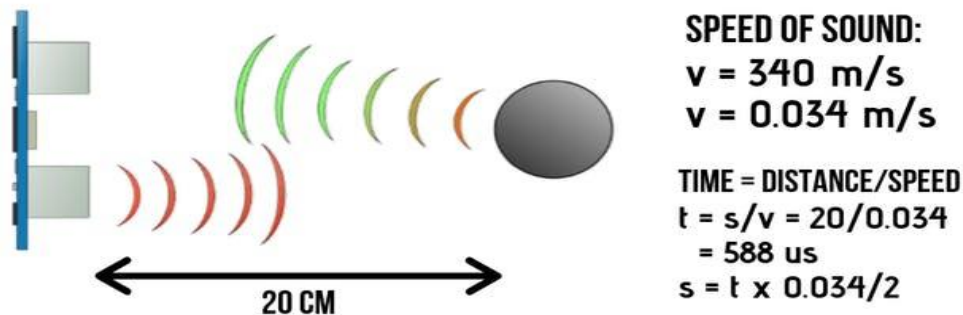
V mojom projekte sme použili dosku Arduino Uno, ultrazvukový senzor HC-SR04, reproduktor, dve LED diódy (zelenú a červenú) s dvoma rezistormi o veľkosti 5k Ω .

Opis:

Ultrazvukový senzor vytvára zvuk s frekvenciou 40 kHz, ktorý sa odrazí od objektu pred ním a vráti sa naspäť do snímača (vid' obrázok č.1). Senzor je pripojený na Arduino dosku pričom malým programovaním získame hodnoty vzdialenosti v centimetroch. Senzor meria vzdialenosť z intervalu <2cm; 450cm>. V mojom projekte bola hranica vzdialenosti 50 cm (int DISTANCE_THRESHOLD), kde po prekročení sa každých 10 cm zvyšovala frekvencia.

Ďalším programovaním určíme akúsi hranicu vzdialenosti. Pokiaľ bude objekt za hranicou bude svietiť zelená LED dióda. Ak bude objekt bližšie ako je daná hranica bude svietiť červená LED dióda. Pri priblížení objektu za hranicu sa spustí reproduktor, ktorý začne pípať s tromi rôznymi frekvenciami (interval medzi pípnutiami – ako senzor pri cúvaní v aute), čím je objekt bližšie tým vyššia frekvencia.

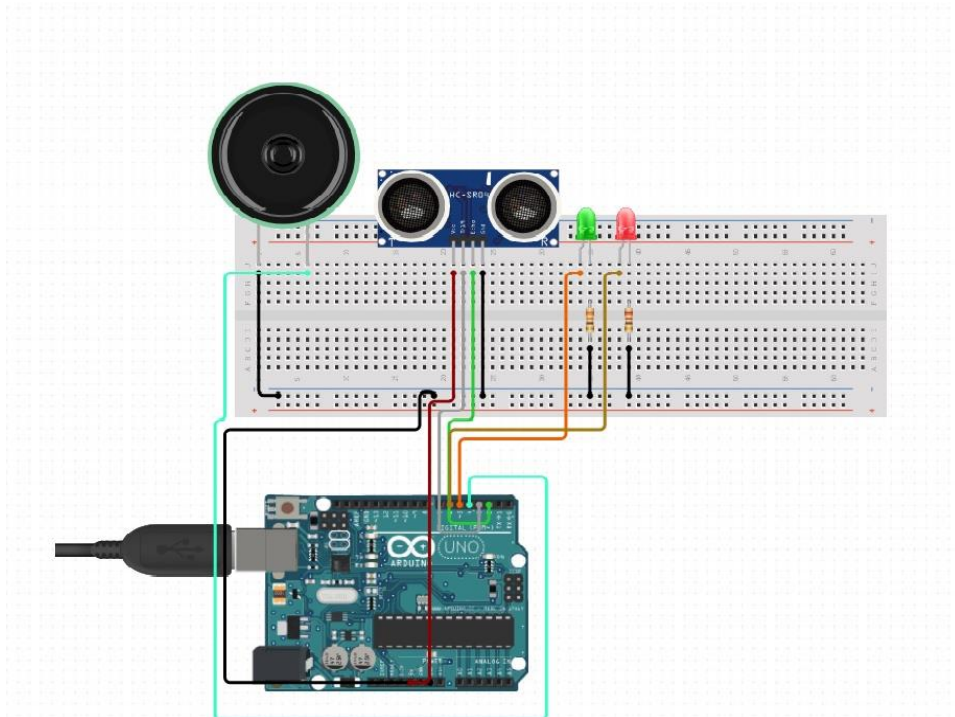
Pomocou knižnice *matplotlib* v Pythone spracujeme dáta v reálnom čase. Namerané hodnoty by sme mohli použiť napríklad na vypočítanie periódy kývadla.



Obrázok 1 - ultrazvukový senzor

Schéma:

Z technických príčin sa nezhoduje kód do Arduina so schémou (č. vstupov).



Obrázok 2 - schéma

```

int trigPin=13;
int echoPin=11;
float pingTime; //time for ping to travel from sensor to target and return
float targetDistance;
float speedOfSound=776.5; //Speed of sound in miles per hour
int LED_PIN1 = 4;
int LED_PIN2 = 3;
int piezoPin = 8;
int DISTANCE_THRESHOLD = 50; // centimeters

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(LED_PIN1, OUTPUT);
  pinMode(LED_PIN2, OUTPUT);
}

void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2000);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(15);
  digitalWrite(trigPin, LOW);
  delayMicroseconds(10);

  pingTime = pulseIn(echoPin, HIGH); //pingTime is presented in microseconds
  pingTime = pingTime/1000000;
  pingTime = pingTime/3600;
  targetDistance = speedOfSound * pingTime; //in miles -> (speed of sound was miles)
  targetDistance = targetDistance/2;
  targetDistance = targetDistance*160934.4; //Convert miles to centimetres

  Serial.println(targetDistance);
  if(targetDistance < DISTANCE_THRESHOLD) {
    digitalWrite(LED_PIN1, HIGH); // turn on LED
    digitalWrite(LED_PIN2, LOW);
    if((targetDistance < DISTANCE_THRESHOLD) && (targetDistance > (DISTANCE_THRESHOLD-10))) {
      tone(piezoPin, 1750, 5);
      delay(250);
      noTone(piezoPin);
    }
    else if ((targetDistance <= (DISTANCE_THRESHOLD-10)) && (targetDistance > (DISTANCE_THRESHOLD-20))) {
      tone(piezoPin, 760, 5);
      delay(150);
      noTone(piezoPin);
    }

    else if ((targetDistance <= (DISTANCE_THRESHOLD-20)) && (targetDistance > (DISTANCE_THRESHOLD-30))) {
      tone(piezoPin, 270, 5);
      delay(100);
      noTone(piezoPin);
    }
    else if (targetDistance <= (DISTANCE_THRESHOLD-30)) {
      tone(piezoPin, 480, 5);
      delay(60);
      noTone(piezoPin);
    }
  }
  else {
    digitalWrite(LED_PIN1, LOW); // turn off LED
    digitalWrite(LED_PIN2, HIGH);
    delay(100); //delay tenth of a second to slow things down a little.
  }
}

```

Obrázok 3 - kód