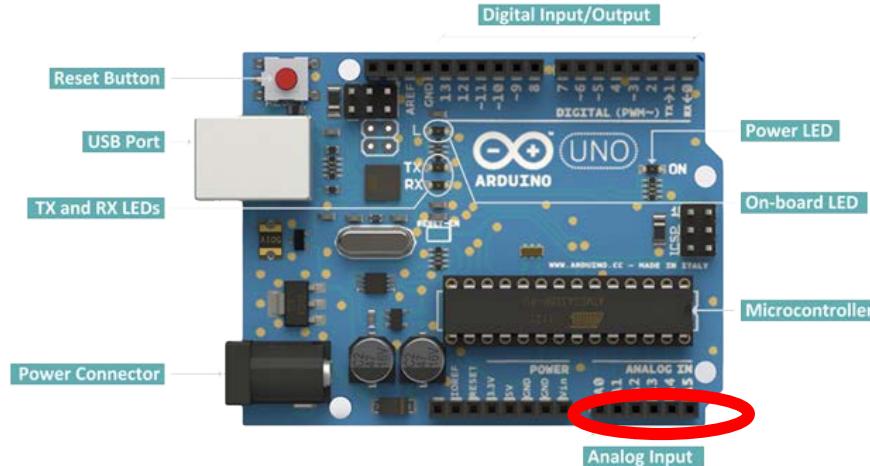


Dizajniranje jednostavnih
kolaborativnih rješenja u
sistemima sa programabilnim
platformama

Analogni ulaz

- Da bi se čitali analogni senzori – čija se vrijednost kontinualno mijenja, trebaju nam analogni pinovi.



- Analogni pinovi nam omogućavaju da odredimo tačnu vrijednost primijenjenog napona uz pomoć `analogRead()` funkcije
- Ova funkcija vraća broj između 0 i 1023, koji predstavlja napone između 0 i 5 V.
- Na primer, ako postoji napon od 2.5 V primijenjen na pinu broj 0, `analogRead(0)` vraća 512.

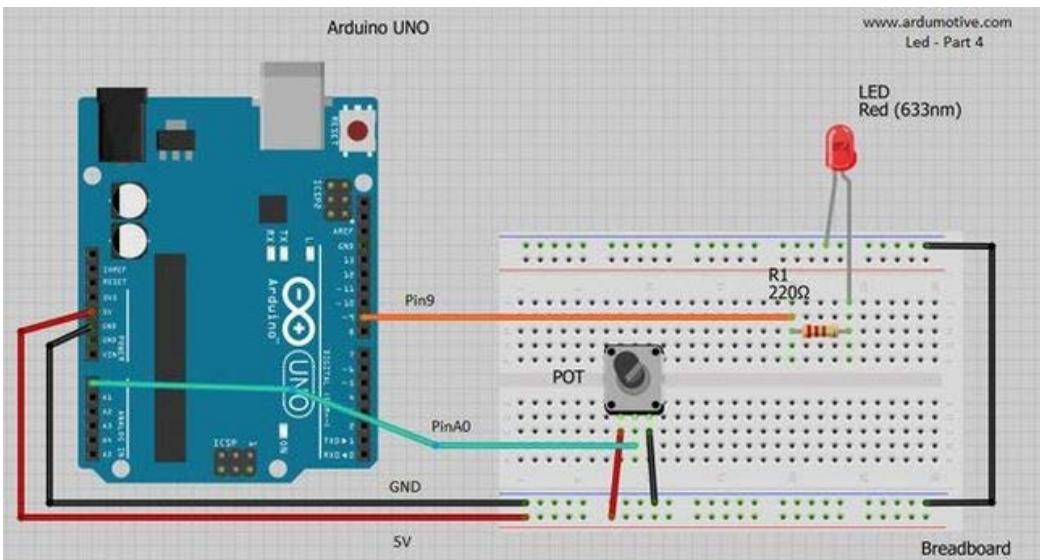
Efekat prigušene svjetlosti

```
// Podesi osvjetljaj diode na vrijednost zadatu
// analognim ulazom
const int LED = 9; // pin za LED
int val = 0; // vrijednost očitana sa senzora

void setup() {
    pinMode(LED, OUTPUT); // LED je OUTPUT
    // Analogni pinovi su sutomatski postavljeni
    // kao INPUT
}
void loop() {
    val = analogRead(0); // čitaj vrijednost sa senzora
    analogWrite(LED, val/4); // postavi LED na vrijednost
    // definisanu senzorom
    // analogWrite očekuje vrijednosti 0-255
    // zato dijelimo sa 4. analogRead daje maksimu
    // 1023
    delay(10); // zaustavi program za zadato kašnjenje
}
```

Potenciometar

- Potenciometar je otpornik promjenljive otpornosti. Otpornost mu se mijenja okretanjem jednostavnog dugmeta i očitava se kao analogna vrijednost.
- Svi potenciometri imaju tri pina:
 - Spoljašnji pinovi se koriste za povezivanje izvora napajanja (Vref i gnd).
 - Srednji pin (izlazni - output) daje promjenljivu vrijednosti otpora definisanu okretanjem dugmeta.



Kada se dugme okreće do kraja na jednu stranu, na izlazni pin potenciometra se šalje 0V, kada se okreće na drugu stranu, šalje se 5V. AnalogRead(0) u tom slučaju očitava 1023.

Naredba `map(value, 0, 1023, 0, 255);` vrši preskaliranje vrijednosti promjenljive value u željeni opseg.

```
// Šalje monitoru vrijednost očitanu sa analognog
// ulaza 0

const int SENSOR = 0;
int val = 0; // vrijednost sa senzora
void setup() {
    Serial.begin(9600); // otvori serijski port za slanje
    // 9600 bits per second
}
void loop() {
    val = analogRead(SENSOR); // čitaj sa senzora
    Serial.println(val); // štampaj vrijednost|
    delay(100); // čekaj 100ms
}
```

Senzori za mjerjenje temperature i vlažnosti vazduha – DHT22

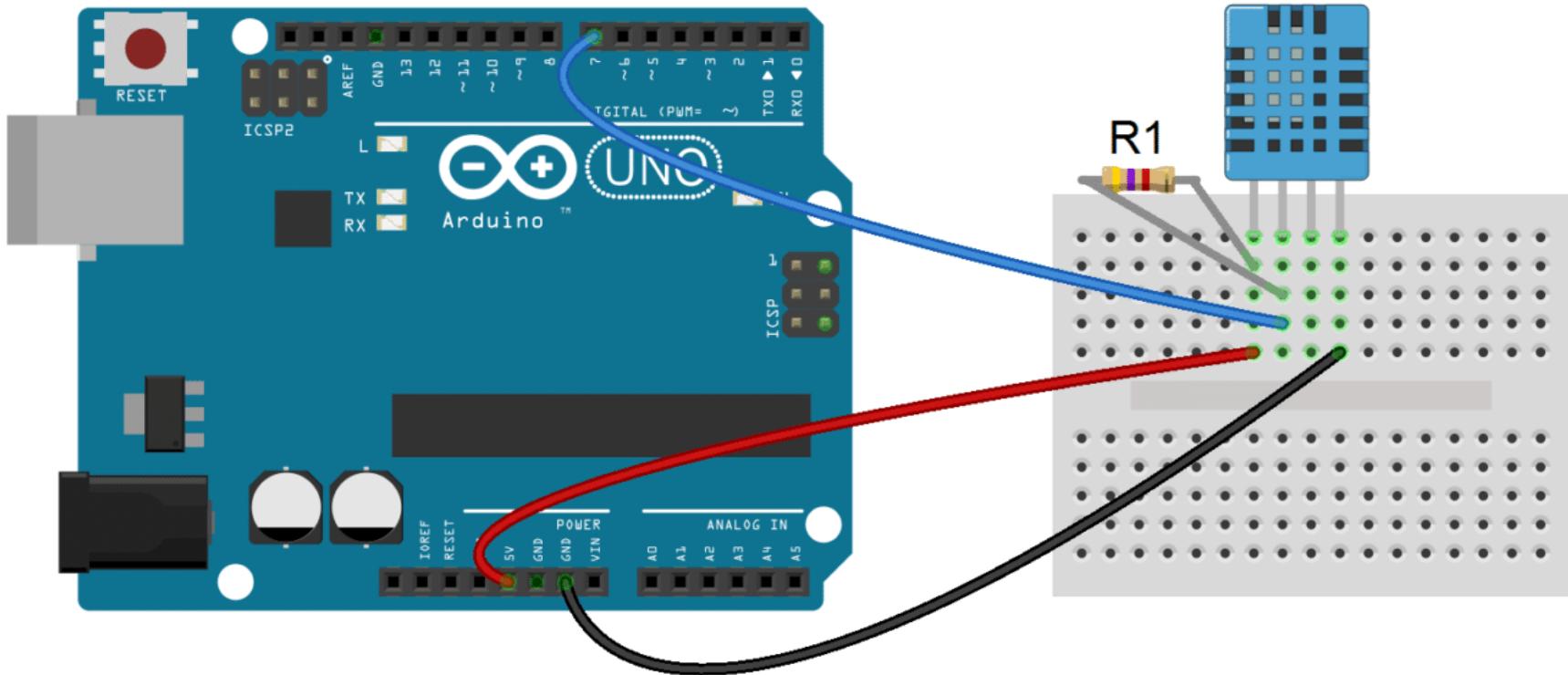
- Digitalni senzor za mjerjenje vlažnosti vazduha i temperature.
- Mjeri relativnu vlažnost. Raspon vlažnosti je 0-100%.
- Temperaturni opseg je -40 - 125 ° C.
- Koristi kapacitivni senzor vlažnosti i termistor za merenje temperature okolnog vazduha. Daje digitalni signal na pin DATA.

DHT22 pins	
1	VCC
2	DATA
3	NC
4	GND



- Prvi pin na lijevoj strani se veže na Vcc, 3-5V napajanje.
- Drugi pin DATA prenosi mjerene podatke
- Četvrти pin GND (krajnji desni se veže na uzemljenje).

Senzori za mjerjenje temperature i vlažnosti vazduha



<http://www.circuitbasics.com/wp-content/uploads/2015/10/Arduino-DHT11-Tutorial-4-Pin-DHT11-Wiring-Diagram.png>

- Otpornik od $10\text{ K}\Omega$ se veže ozmeđu pina sa izlaznim signalom i napajanja.

Senzori za mjerjenje temperature i vlažnosti

- DHTLib biblioteka se mora instalirati da bi se mogao koristiti DHT22 senzor na Arduinu.
- Ova biblioteka ima sve funkcije potrebne za dobijanje očitavanja vlažnosti i temperature sa senzora.
- U Arduino razvojnom okruženju se odabere *Sketch → Include Library → Manage Libraries*. U tom dijelu potrebno je instalirati sljedeće biblioteke:
 - DHT sensor library by Adafruit
 - Adafruit Unified Sensor by Adafruit.

Nakon instalacije se može pokrenuti primjer:

- File → Examples → DHT sensor library

```
// Example testing sketch for various DHT humidity/temperature sensors
// Written by ladyada, public domain

// REQUIRES the following Arduino libraries:
// - DHT Sensor Library: https://github.com/adafruit/DHT-sensor-library
// - Adafruit Unified Sensor Lib: https://github.com/adafruit/Adafruit\_Sensor

#include "DHT.h"

#define DHTPIN 2      // Digital pin connected to the DHT sensor
// Feather HUZZAH ESP8266 note: use pins 3, 4, 5, 12, 13 or 14 --
// Pin 15 can work but DHT must be disconnected during program upload.

// Uncomment whatever type you're using!
//#define DHTTYPE DHT11  // DHT 11
#define DHTTYPE DHT22  // DHT 22 (AM2302), AM2321
//#define DHTTYPE DHT21  // DHT 21 (AM2301)

// Connect pin 1 (on the left) of the sensor to +5V
// NOTE: If using a board with 3.3V logic like an Arduino Due connect pin 1
// to 3.3V instead of 5V!
// Connect pin 2 of the sensor to whatever your DHTPIN is
// Connect pin 4 (on the right) of the sensor to GROUND
// Connect a 10K resistor from pin 2 (data) to pin 1 (power) of the sensor

// Initialize DHT sensor.
// Note that older versions of this library took an optional third parameter to
// tweak the timings for faster processors. This parameter is no longer needed
// as the current DHT reading algorithm adjusts itself to work on faster procs.
DHT dht(DHTPIN, DHTTYPE);
```

```
void setup() {
    Serial.begin(9600);
    Serial.println(F("DHTxx test!"));

    dht.begin();
}

void loop() {
    // Wait a few seconds between measurements.
    delay(2000);

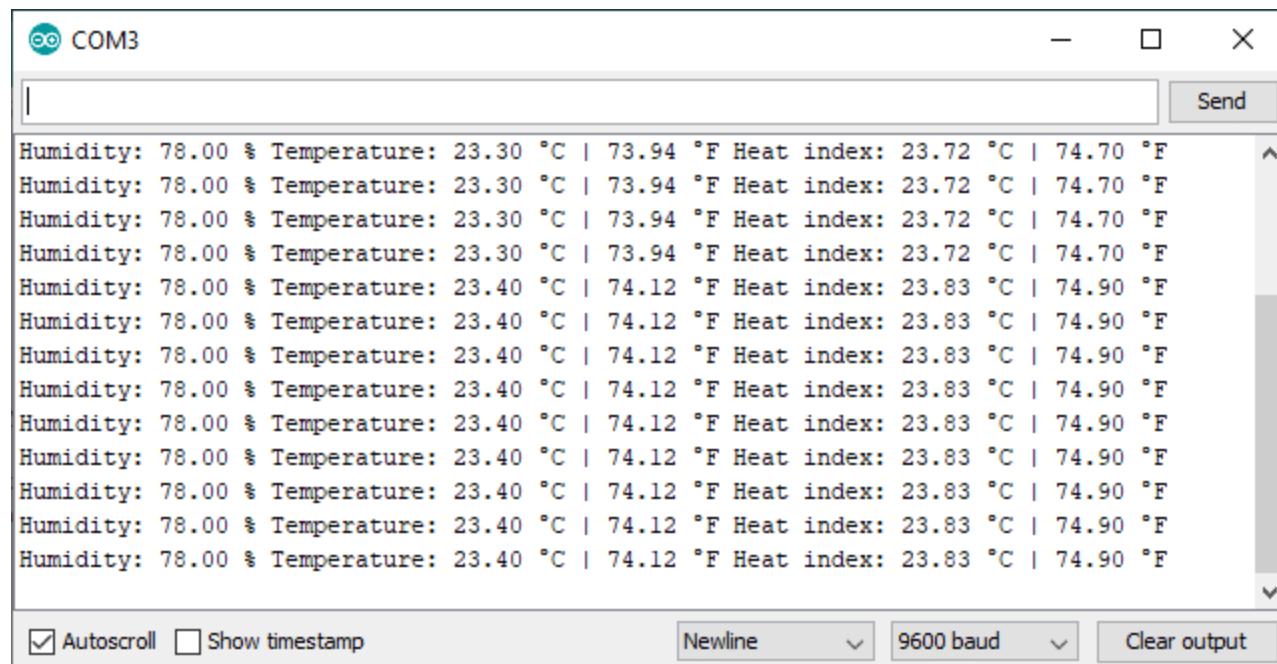
    // Reading temperature or humidity takes about 250 milliseconds!
    // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
    float h = dht.readHumidity();
    // Read temperature as Celsius (the default)
    float t = dht.readTemperature();
    // Read temperature as Fahrenheit (isFahrenheit = true)
    float f = dht.readTemperature(true);

    // Check if any reads failed and exit early (to try again).
    if (isnan(h) || isnan(t) || isnan(f)) {
        Serial.println(F("Failed to read from DHT sensor!"));
        return;
    }

    // Compute heat index in Fahrenheit (the default)
    float hif = dht.computeHeatIndex(f, h);
    // Compute heat index in Celsius (isFahrenheit = false)
    float hic = dht.computeHeatIndex(t, h, false);
```

```
Serial.print("Humidity: ");
Serial.print(h);
Serial.print(" % ");
Serial.print("Temperature: ");
Serial.print(t);
Serial.print(" \xC2\xB0");
Serial.print("C | ");
Serial.print(f);
Serial.print(" \xC2\xB0");
Serial.print("F ");
Serial.print("Heat index: ");
Serial.print(hic);
Serial.print(" \xC2\xB0");
Serial.print("C | ");
Serial.print(hif);
Serial.print(" \xC2\xB0");
Serial.println("F");
```

<https://www.makerguides.com/dht11-dht22-arduino-tutorial/>



Senzor za mjerjenje vlažnosti zemljišta



Mjeri vlažnost zemljišta na osnovu promjene električne provodljivosti zemlje (otpornost zemljišta se povećava sa sušom).

VCC – napajanje;

GND – uzemljenje;

SIG – na analogni pin.



Uključi lampicu ako je zemljište suvo

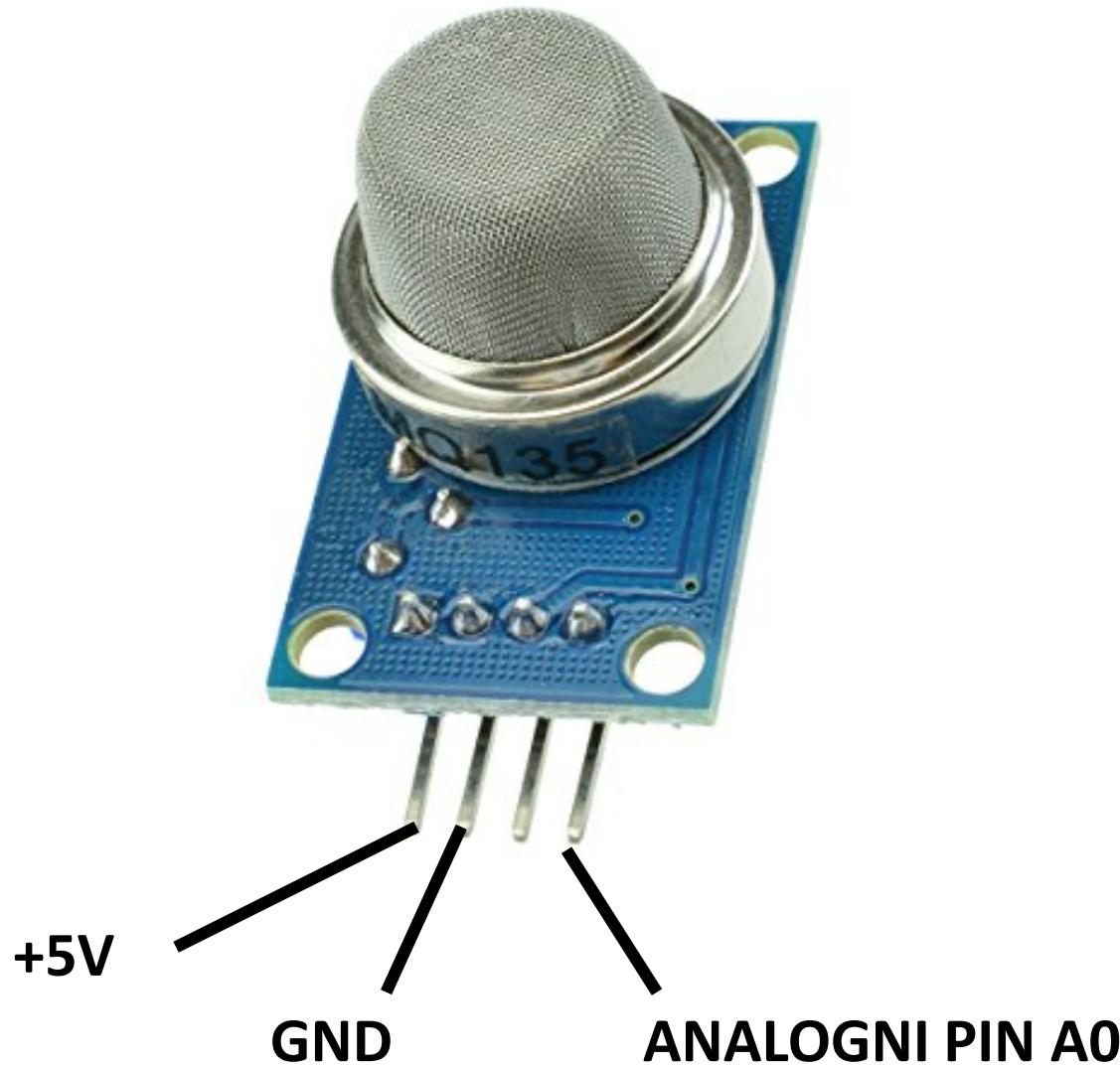
```
int sensorPin = A0; //analogni pin koji koristimo
int sensorValue; // analogna vrijednost koja ce
//se citati sa senzora
int limit = 300; //ako vrijednost ocitana sa senzora bude veca
// od zadatog limita iskljuci integrisanu LED diodu ili diodu
// koju povezemo na ovaj pin
void setup() {
    Serial.begin(9600); //ispisivacemo ocitavanja na serijskom ekranu
    pinMode(13, OUTPUT); // Slacemo visoki napon na ovaj pin
}

void loop() {
    sensorValue = analogRead(sensorPin); //ocitana vrijednost sa senzora
    Serial.println("Analog Value : ");
    Serial.println(sensorValue); //ispisujemo ocitanu vrijednost na
    // serijskom monitoru
    if (sensorValue<limit) { //ako je ocitana vrijednost manja od praga
        digitalWrite(13, HIGH); //lampica treba biti ukljucena
    }
    else { //ako nije
        digitalWrite(13, LOW); //iskljuci lampicu
    }
    delay(1000); // kasnjenje
}
```

Kvalitet vazduha



Povezivanje



Očitavanje

```
void loop()
{
    int sensorValue = analogRead(A0);

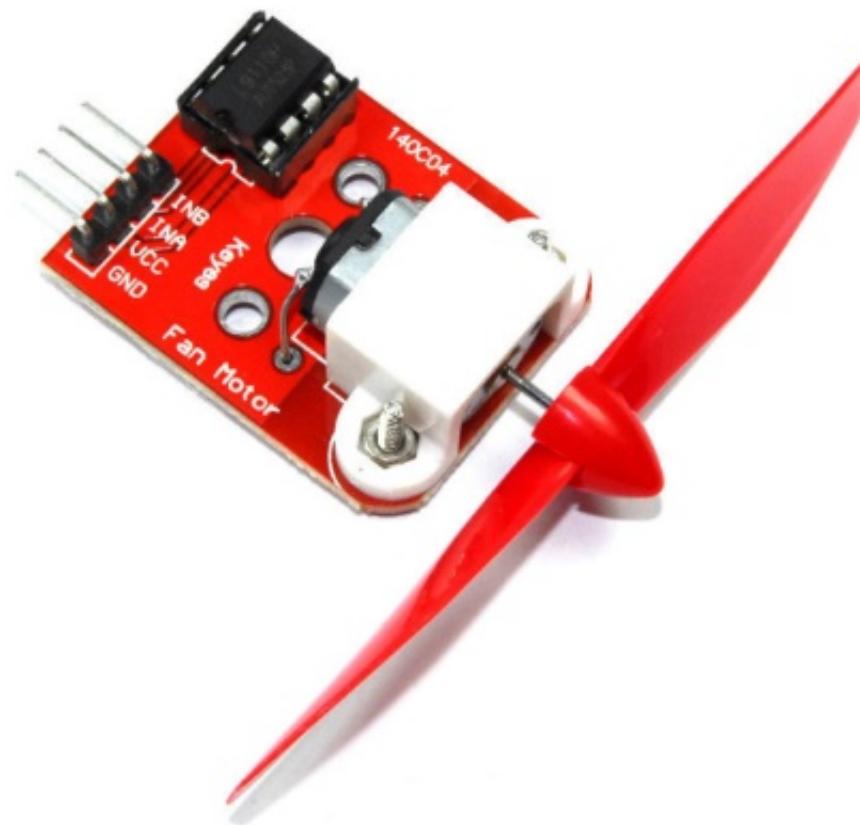
    Serial.print("Air Quality = ");

    Serial.print(sensorValue);

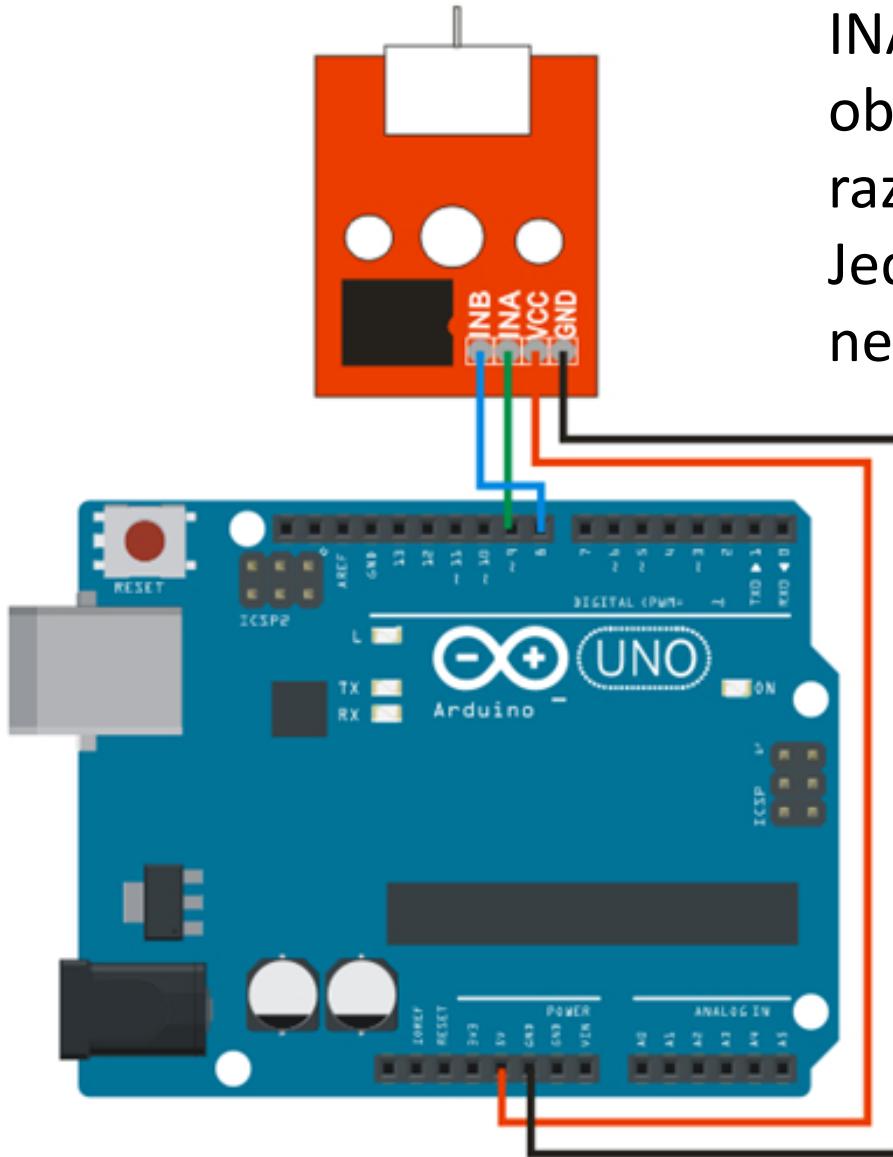
    Serial.println(" *PPM");

    delay(1000);
}
```

Propeller Fan Control



INA, INB brzina
obrtanja u dva
različita smjera.
Jedan mora biti
neaktivovan - LOW



Neka se propeler okreće zadatom brzinom



The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** sketch_feb26a | Arduino 1.8.12
- Menu Bar:** File Edit Sketch Tools Help
- Toolbar:** Includes icons for Save, Run, Stop, Upload, and Download.
- Sketch Name:** sketch_feb26a
- Code Area:** Contains the following C++ code for an Arduino sketch:

```
int INA = 9; //moze se kontrolisati sa analogwrite
int INB = 8; //bice nam neaktivan - okrecemo propeler u jednom smjeru
int velocity; //zadata brzina

void setup() {
    pinMode(INA, OUTPUT);
    pinMode(INB, OUTPUT);
}

void loop() {
    velocity = 100;
    analogWrite(INA, velocity);
    digitalWrite(INB, LOW);
}
```

Modifikacija

Isporigramirati propeler tako da:

- Povećava brzinu od MIN (0) do MAX (255)
- Smanjuje brzinu od MAX (255) do MIN (0)

U jednoj loop() funkciji.



[File](#) [Edit](#) [Sketch](#) [Tools](#) [Help](#)

sketch_feb26a

```
int INA = 9; //moze se kontrolisati sa analogwrite
int INB = 8; //bice nam neaktivan - okrecemo propeler u jednom smjeru
int velocity; //zadata brzina

void setup() {
    pinMode(INA, OUTPUT);
    pinMode(INB, OUTPUT);
}

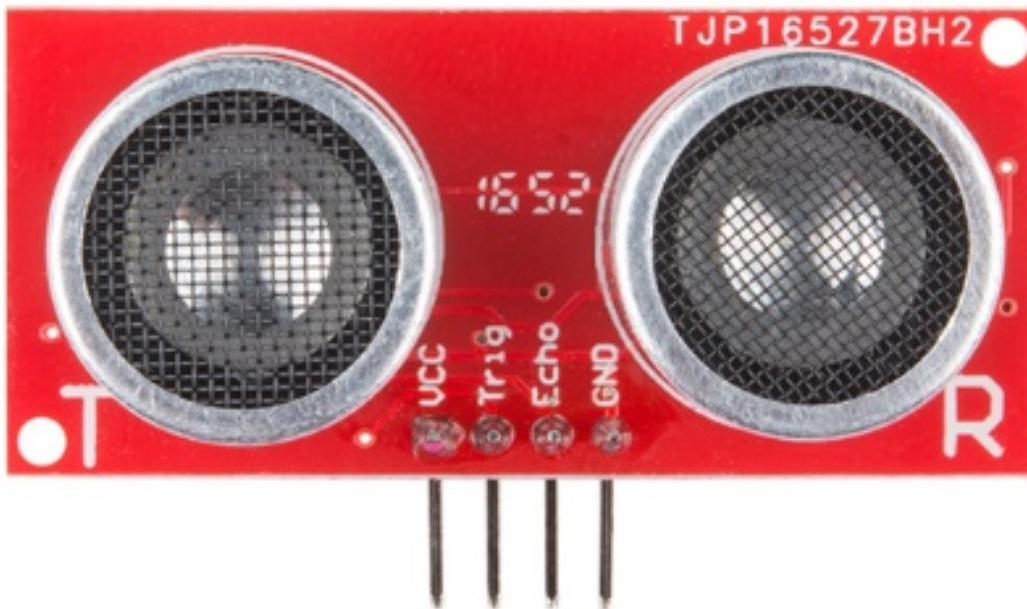
void loop() {
    for(velocity = 0; velocity < 256; velocity++) {
        analogWrite(INA, velocity);
        delay(3000);
    }
    for(velocity = 255; velocity >= 0; velocity--) {
        analogWrite(INA, velocity);
        delay(3000);
    }
    digitalWrite(INB, LOW);
}
```

Done compiling.

Global variables use 11 bytes (0%) of dynamic memory, leaving 2037 byt

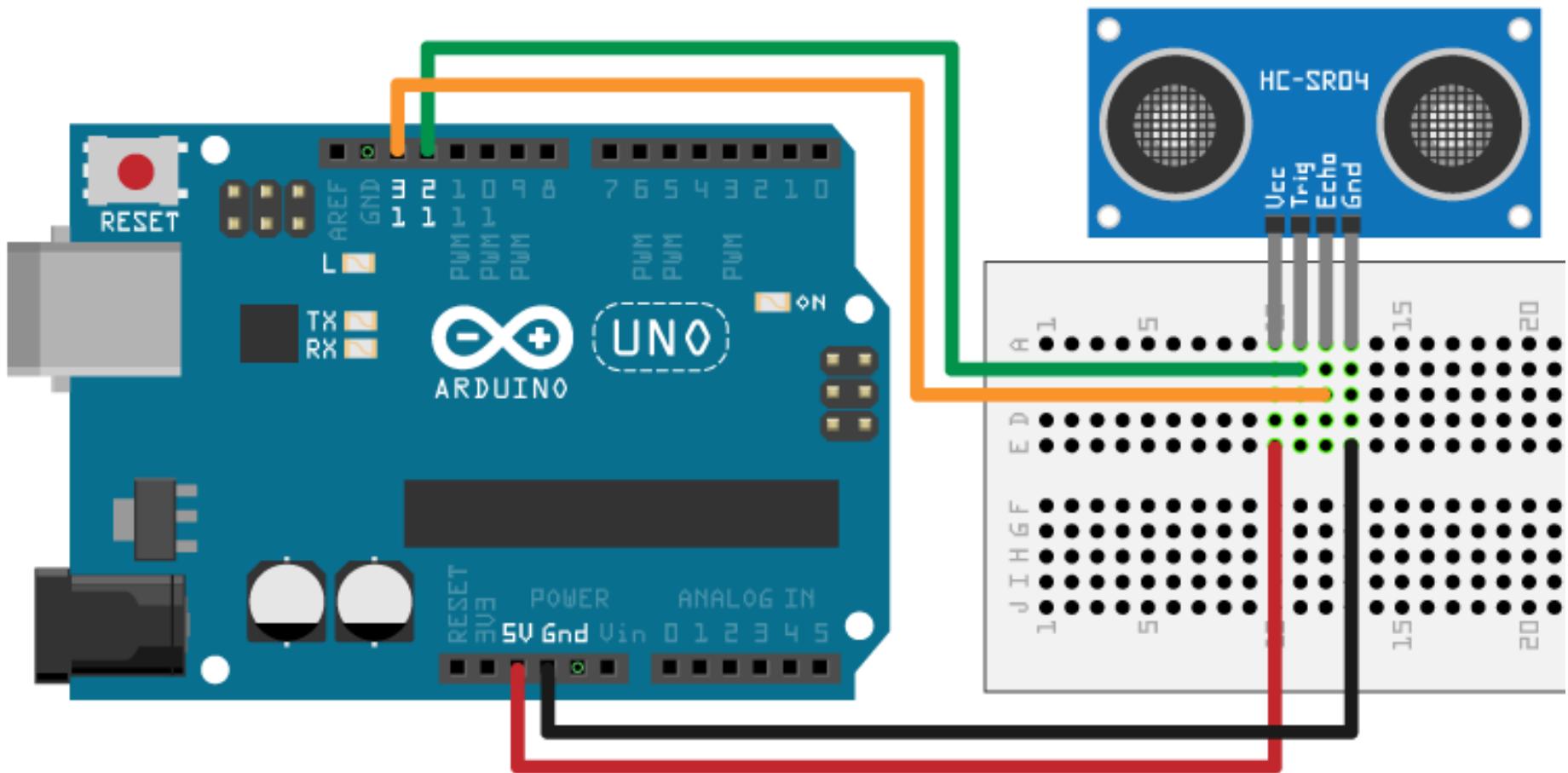
Ultrasonic Sensor

Mjerenje rastojanja

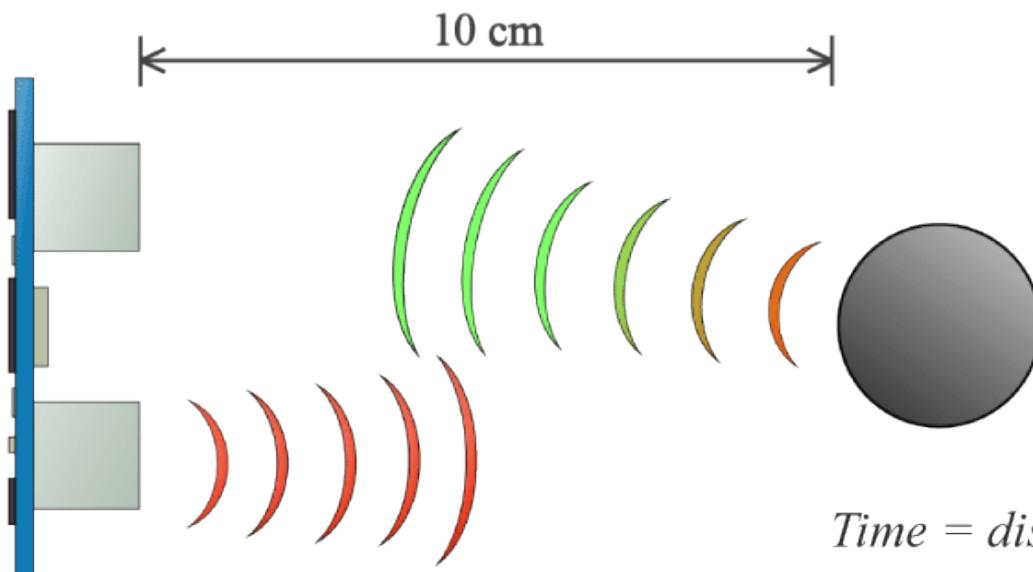


Funkcioniše tako što šalje zvučne talase iz predajnika (kojim se upravlja preko Trig pina), koji se potom odbijaju od objekat i vraćaju ka prijemniku (Echo). Udaljenost objekta od kojeg je odbijen zvučni talas se računa na osnovu vremena potrebnog da se zvučni talasi vrati do senzora.

Povezivanje



Kako dobiti rastojanje iz vremena potrebnog zvučnom signalu da stigne do objekta i da se vrati nazad?



speed of sound:

$$v = 340 \text{ m/s}$$

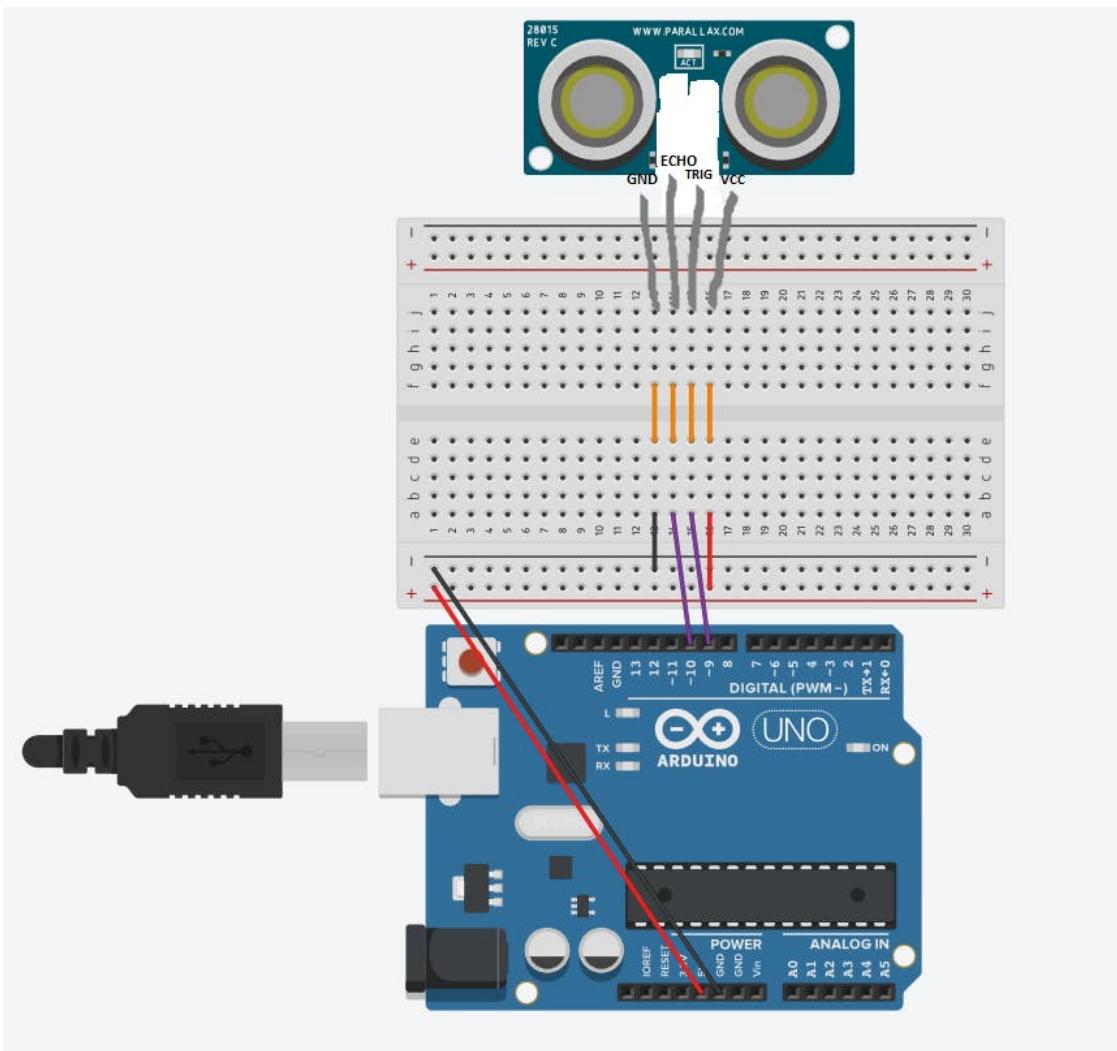
$$v = 0,034 \text{ cm} / \mu\text{s}$$

Time = distance / speed:

$$t = s / v = 10 / 0,034 = 294 \mu\text{s}$$

Distance:

$$s = t \cdot 0,034 / 2$$



<https://create.arduino.cc/projecthub/Isaac100/getting-started-with-the-hc-sr04-ultrasonic-sensor-036380>

File Edit Sketch Tools Help



sketch_feb27a §

```
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
Serial.begin(9600); // zelimo ispis mjeranja

}

void loop() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    duration = pulseIn(echoPin, HIGH);
    distance = (duration*.0343)/2;
    Serial.print("Distance: ");
    Serial.println(distance);
    delay(100);
}
```

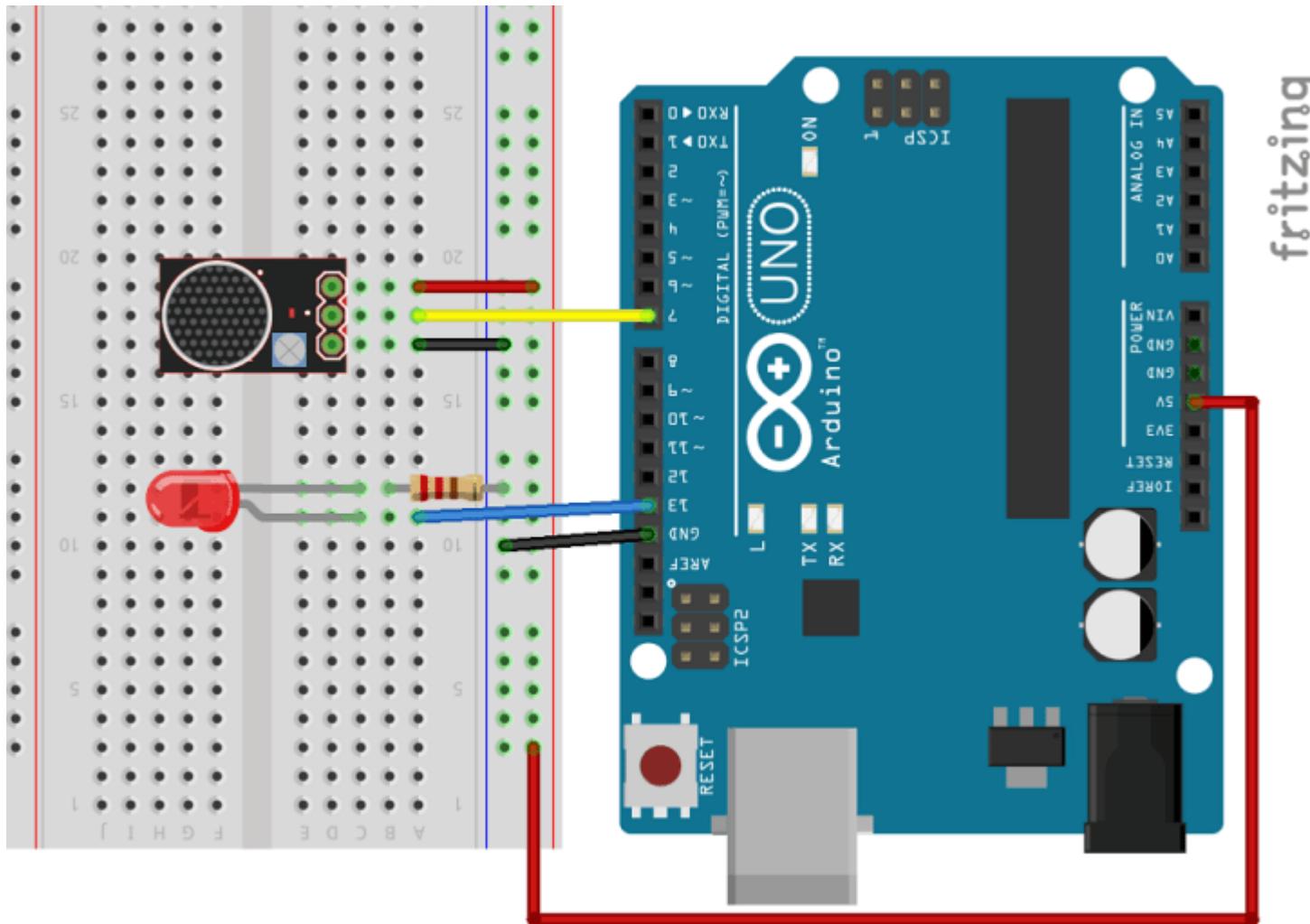
U trenutku kada se pošalje signal sa *trigPin-a*, *echoPin* počinje da osluškuje i postavlja se na *HIGH*. Ostaje *HIGH* sve dok ne primi signal poslat sa trgera i odbijen od objekta čiju udaljenost mjerimo.

Mikrofon

A0 – analogni pin
D0 – digitalni pin



Neka dioda zasvijetli ako se aktivira digitalni pin – zvuk preko praga



<https://randomnerdtutorials.com/guide-for-microphone-sound-sensor-with-arduino/>

File Edit Sketch Tools Help

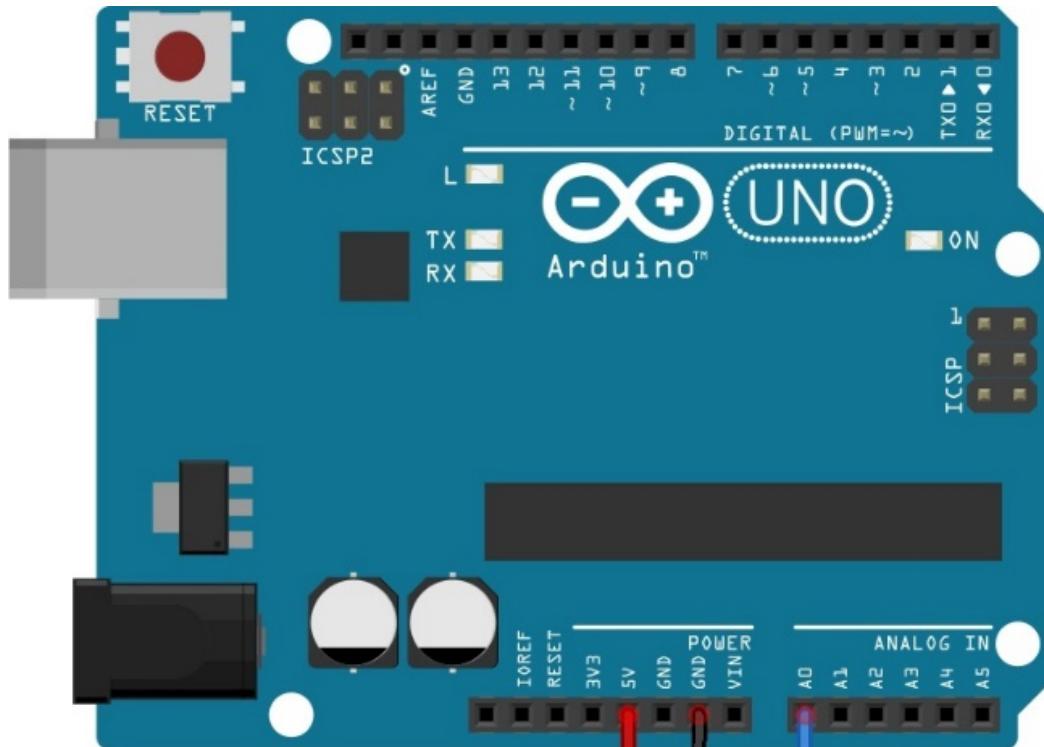


sketch_feb27a

```
int ledPin=13;//za upravljanje diodom
int sensorPin=7;// za osluskivanje zvuka
boolean val =0;

void setup() {
    pinMode(ledPin, OUTPUT);
    pinMode(sensorPin, INPUT);
    Serial.begin (9600);
}

void loop () {
    val =digitalRead(sensorPin);//citaj sa senzora vrijednost
    Serial.println (val);
    // ako je vrijednost preko praga definisanog digitalnim pinom
    if (val==HIGH)
        digitalWrite(ledPin, HIGH);
    else
        digitalWrite(ledPin, LOW);
}
```



Može se čitatavati
analognim pinom

