



# PRACOWNIA MIERNICTWA ELEKTRONICZNEGO/ INTERNETU RZECZY

## ĆWICZENIE NR 2

**Temat:**

**Sensory analogowe**

Opracowanie wstępne:

**Sensory cyfrowe i analogowe w systemach Internetu Rzeczy**

**Komponenty:**

Arduino  
Potencjometr THT

Dioda LED  
Rezystor 220  $\Omega$

Czujnik dźwięku

**Cel ćwiczenia:**

Zbudowanie i zaprogramowanie układu, który automatycznie zapala światło (na przykładzie diody LED) po wykryciu ruchu, przy jednoczesnym uwzględnieniu warunków oświetleniowych – układ ma działać jak systemy funkcjonujące np. na klatkach schodowych – wykrycie ruchu zapala światło, ale tylko wtedy, kiedy jest odpowiednio ciemno.

## INSTRUKCJA

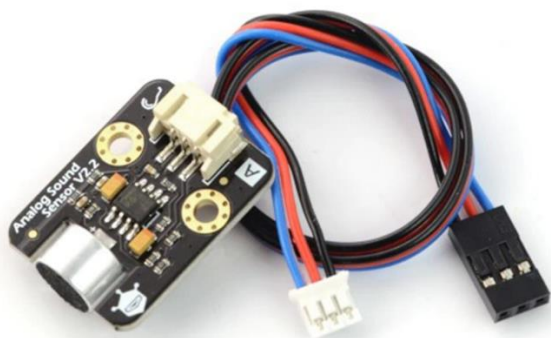
### Krok 0.

Należy podłączyć diodę LED w sposób pokazany w materiale „Podstawy budowy i programowania Arduino oraz budowania obwodów elektronicznych”

### Krok 1.

Należy podłączyć czujnik dźwięku do pinu analogowego (do czujnika dołączony jest przewód, który należy podłączyć w następujący sposób: **niebieski przewód** do pinu analogowego (w tym przykładzie A0); **czzerwony przewód** do źródła napięcia 5V; **czarny przewód** do masy GND. Następnie należy napisać kod, odczytujący i wyświetlający wartość poziomu dźwięku na monitorze portu szeregowego (w Arduino IDE zakładka „Tools” -> „Serial Monitor”, skrót Ctrl+Shift+M).

#### Czujnik dźwięku:



#### Kod:

```
const int sensorPin = A0;
int sound = 0;
void setup() {
  pinMode(sensorPin, INPUT);
  Serial.begin(9600);
}
void loop() {
  sound = analogRead(sensorPin);
  Serial.println(sound);
  delay(1000);
}
```

Używamy tu obiektu „Serial”, który odpowiada za transmisję szeregową. Metoda „Serial.println” pozwala wyświetlać odczytane z danego pinu wartości w monitorze portu szeregowego.

### Krok 2.

Należy napisać kod, który będzie zapalał diodę LED po wykryciu pojedynczego kłaśnięcia i gasił ją przy wykryciu dwóch kłaśnięć w krótkim odstępie czasu.

```
const int ledPin = 6;
const int sensorPin = A0;
int ledVal = 0;
int sensorVal = 0;
int clapCount = 0;
unsigned long clapTime = 0;
unsigned long lastClapTime = 0;
void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(sensorPin, INPUT);
}
```

```

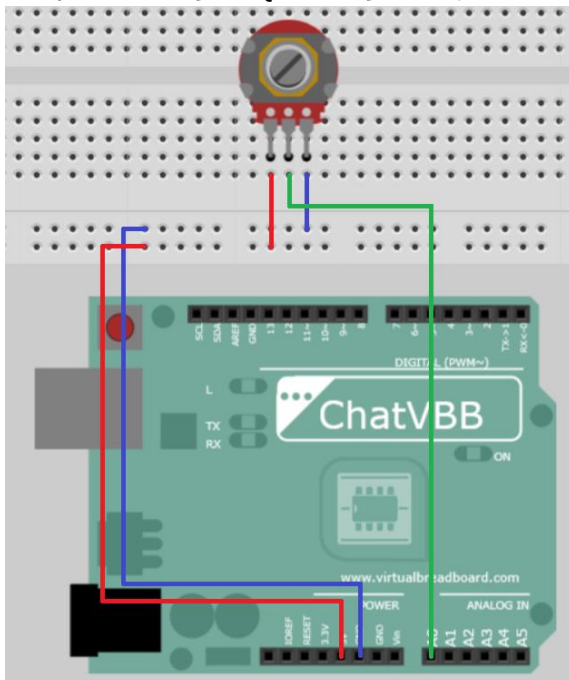
void loop() {
  sensorVal = analogRead(sensorPin);
  if (sensorVal > 100) {
    clapCount++;
    clapTime = millis();
    if (clapCount == 1){
      digitalWrite(ledPin, HIGH);
      lastClapTime = clapTime;
    }
    else{
      if (clapTime - lastClapTime < 2000){
        digitalWrite(ledPin, LOW);
      }
    }
    clapCount = 0;
  }
  delay(100);
}

```

### Krok 3a.

Do (wolnego) pinu analogowego należy podłączyć potencjometr i za pomocą odpowiedniego kodu wyświetlić wskazanie potencjometru w monitorze portu szeregowego.

#### Przykładowe podłączenie potencjometru:



#### Kod:

```

const int potPin = A0;
int potVal = 0;
void setup() {
  pinMode(potPin, INPUT);
  Serial.begin(9600);
}
void loop() {
  potVal = analogRead(potPin);
  Serial.println(potVal);
  delay(1000);
}

```

### Krok 3b.

Obwód 2. należy uzupełnić o obwód 3a. w taki sposób, by z pomocą potencjometru móc regulować poziom dźwięku identyfikowalnego jako „kłaśnięcie”.

Opracowanie końcowe:

Jakie mogą być zastosowania wykorzystanych czujników (i innych sensorów analogowych) w rozwiązaniach Internetu Rzeczy

### **ZADANIE DODATKOWE**

Kod i układ należy zmodyfikować tak, aby dioda miała różne poziomy świecenia (np. 25%, 50%, 75%, 100% mocy) w zależności od poziomu wykrytego dźwięku.

**Podpowiedź:** przydatne będą tu: pin PWM oraz instrukcja `analogWrite` (patrz: materiał „Podstawy budowy i programowania Arduino oraz budowania układów elektronicznych”).

**UWAGA:** Dioda wciąż powinna być podłączona za pośrednictwem rezystora 220  $\Omega$ .