

Politechnika Świętokrzyska w Kielcach

Wydział Elektroniki, Automatyki i Informatyki

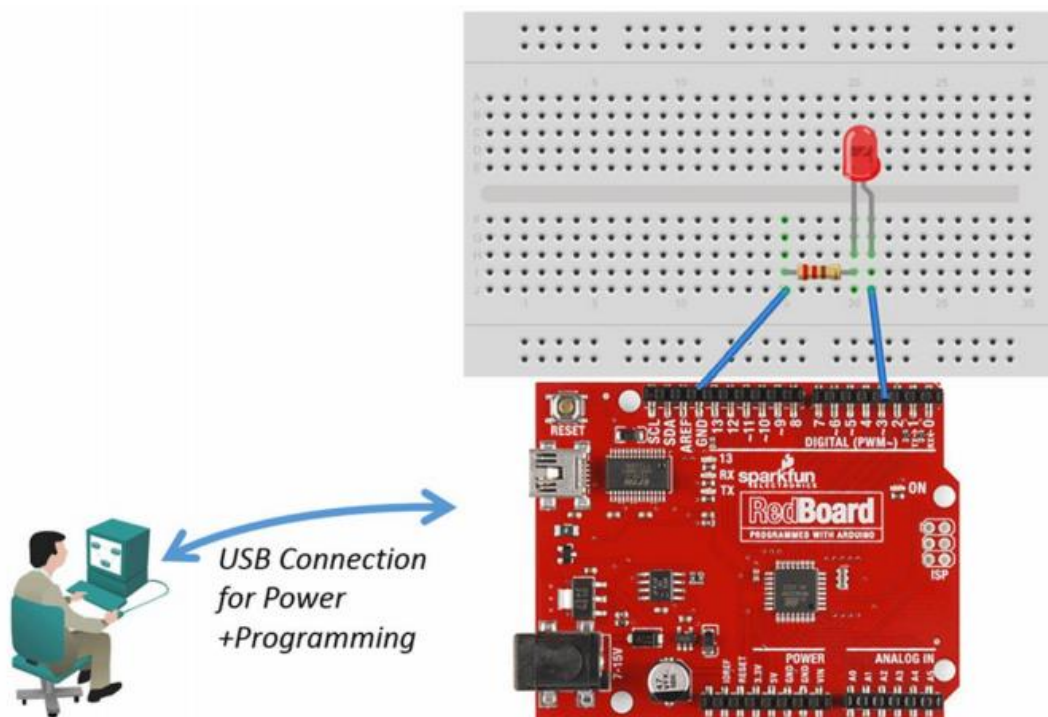
Laboratorium IoT

Tematy: Blinking an LED using RedBoard and Arduino IDE oraz Buzzer

Grupa: **3ID15A**
Konrad Nowakowski
Michał Ortyl
Mateusz Orłowski

Blinking an LED using RedBoard and Arduino IDE

1. Topologia



2. Cel zadania

W tym ćwiczeniu nauczymy się obsługi Arduino i Arduino IDE. Naszym celem będzie sprawienie, aby dioda LED zaczęła świecić.

Aby wykonać ćwiczenie potrzebujemy takich urządzeń jak:

- PC z dostępem do Internetu,
- Zestaw SparkFun,
- Rezystor, dioda LED, przewody oraz rezystor.

3. Wykonanie

a) Aby ustawić pin 13 jako wyjściowy należy użyć funkcji **pinMode(13, OUTPUT)**.

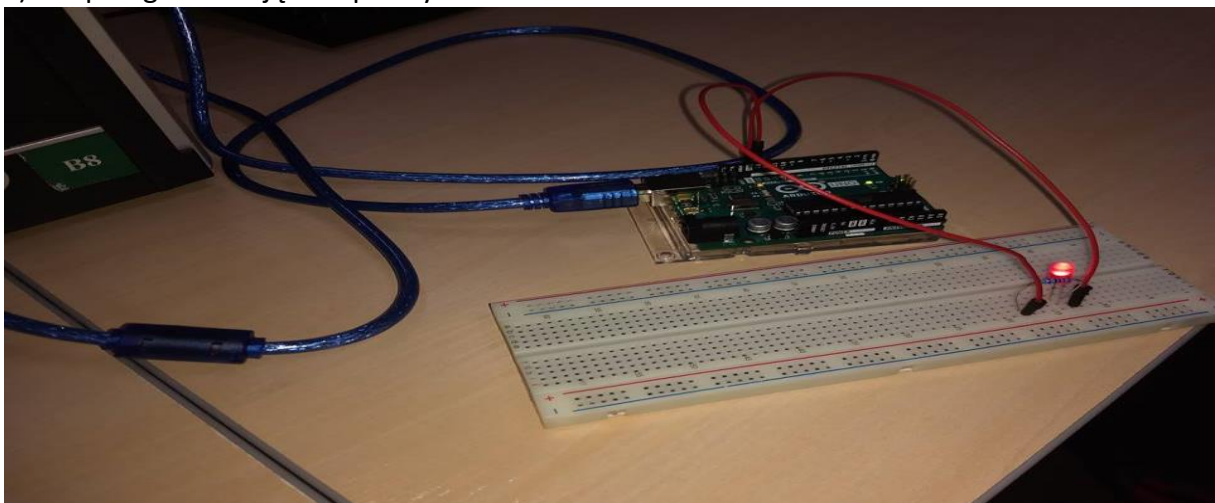
```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}
```

b) Aby dioda zaczęła świecić potrzebujemy poniższej funkcji:

```
// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); //turn LED on (HIGH is the voltage level)
  delay(1000);            //wait for a second
  digitalWrite(13, LOW);  //turn LED off by making the voltage LOW
  delay(1000);            //wait for a second
}
```

Funkcja **digitalWrite** ustawia wartość pinu 13 na wysoki. Pod funkcją **digitalWrite** zostaje ustawione opóźnienie(1s), następnie funkcja **digitalWrite** modyfikuje wartość portu 13 z wysokiego na niski i znów następuje opóźnienie. Za zakończeniu tych funkcji pętla zostanie ponownie uruchomiona i będzie działać do momentu wyłączenia ReadBoard.

c) Topologia działająca w praktyce

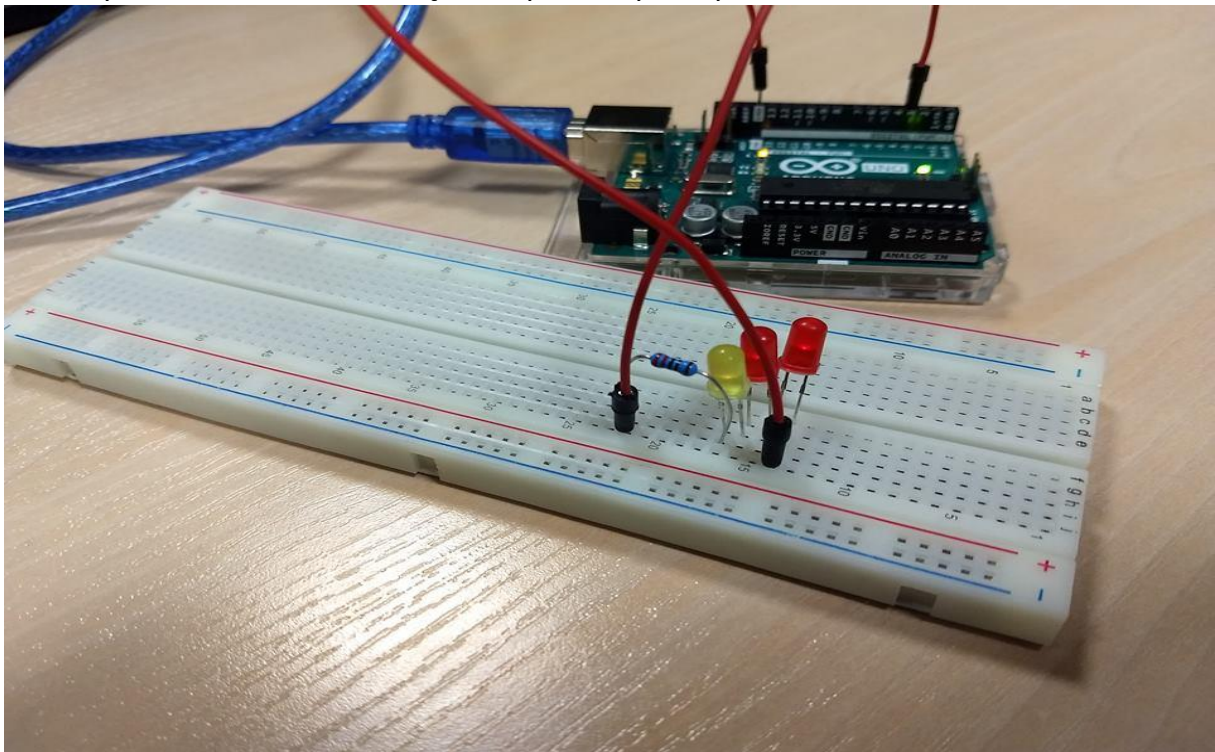


- d) Następnie mieliśmy za zadanie uruchomić topologie, ale korzystając z portu 3. Aby wszystko zadziało i dioda LED zaczęła świecić trzeba zmienić nasz kod oraz przepiąć przewody na pin o numerze 3.

Kod po zmianie będzie wyglądał w taki sposób:

```
Void setup() {  
    pinMode(3, OUTPUT);  
}  
Void loop(){  
    digitalWrite(3,HIGH);  
    delay(1000);  
    digitalWrite(3,LOW);  
    delay(1000);  
}
```

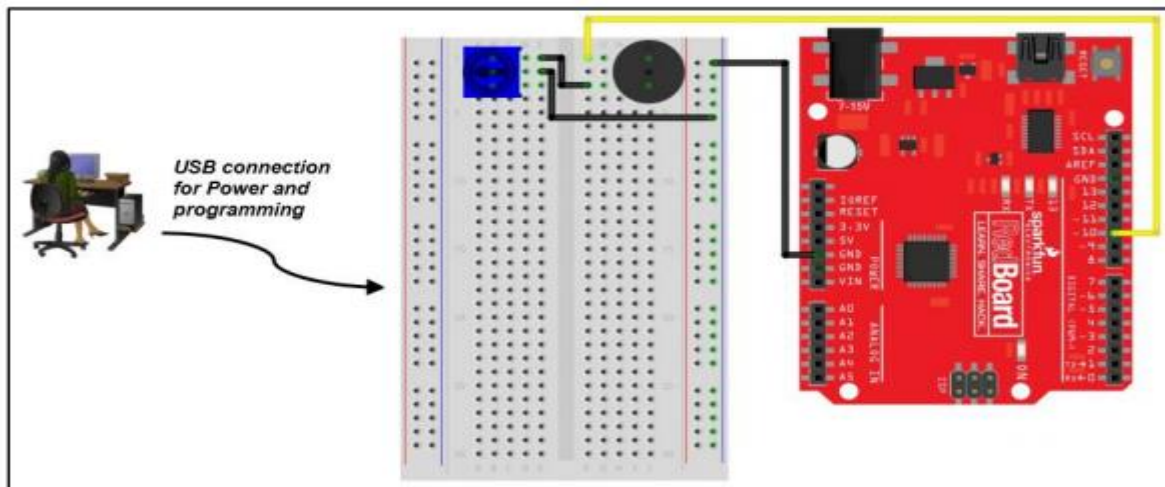
- e) Wykonaliśmy także naszą topologię, ale tym razem z diodami połączonymi w sposób równoległy. Diody w tym połączenia świeciły ale dużo słabiej, ponieważ napięcie w danym momencie musiało się rozłożyć na trzy diody.



- f) Jakie mogą nastąpić problemy, dla których dioda LED nie będzie świecić?
- może być niesprawna
 - połączona może być w kierunku zaporowym
 - układ został źle połączony

Buzzer

1. Topologia



2. Cel zadania

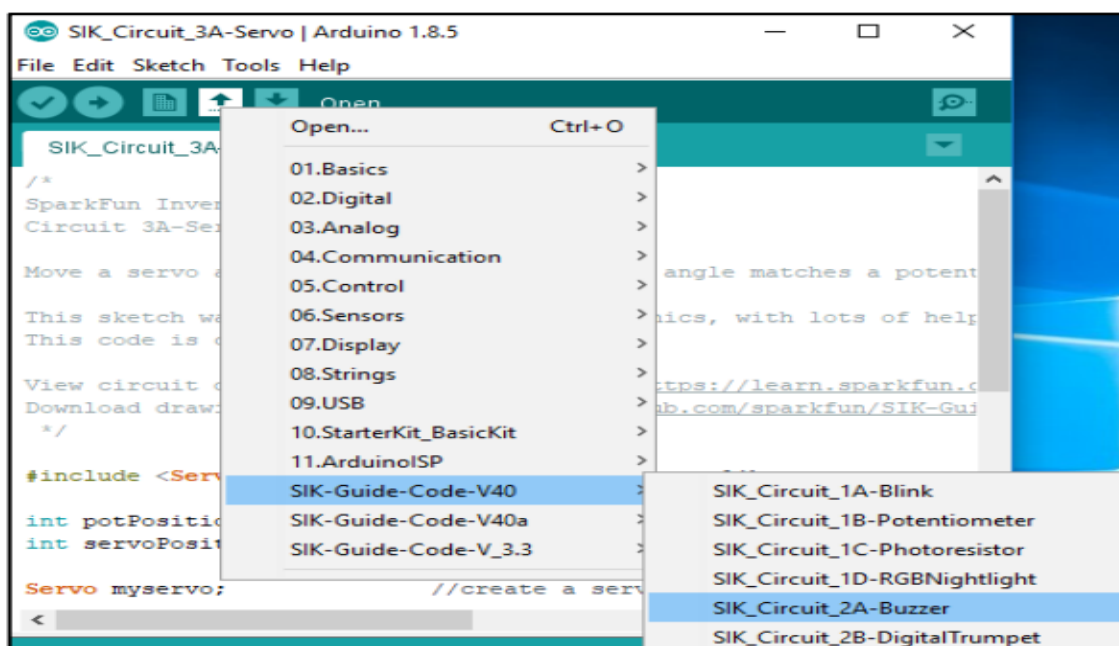
Połączenie oraz konfiguracja urządzeń według topologii, w taki sposób aby z brzęczyka(buzzer) wydobywała się melodia.

Aby wykonać ćwiczenie potrzebujemy takich narzędzi jak:

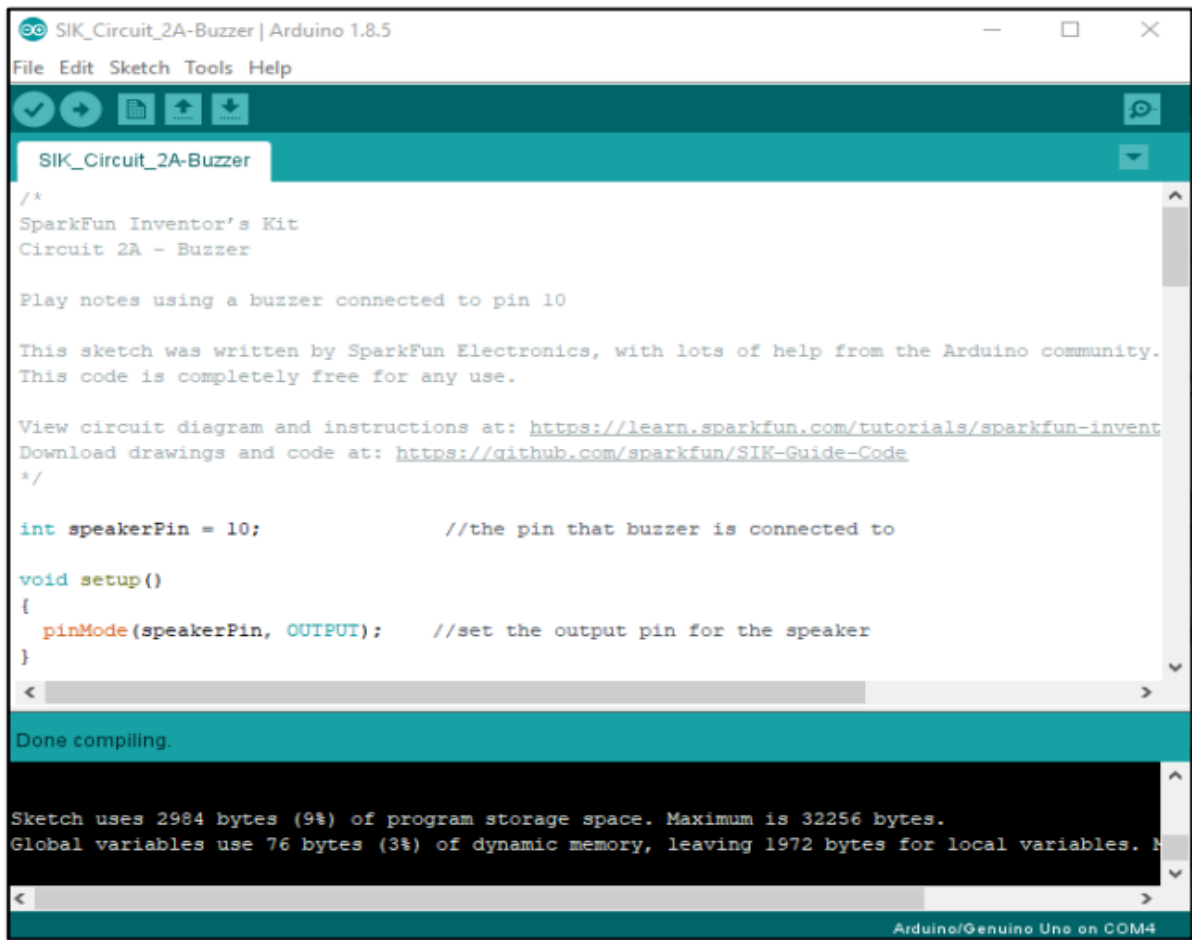
- Zestaw SparkFun,
- Komputer PC z Arduino
- Plik przykładowych kodów przewodnika SIK,
- przewody, buzzer oraz potencjometr.

3. Wykonanie

- a) podłączenie urządzeń i przewodów według topologii
- b) wybór odpowiedniego kodu(arduino) dla Buzzera



Upewniamy się czy nasz układ został dobrze podłączony, oraz czy jest on podłączony do komputera kablem USB. Następnie przesyłamy oprogramowanie od ReadBorder'a.



```
SIK_Circuit_2A-Buzzer | Arduino 1.8.5
File Edit Sketch Tools Help

SIK_Circuit_2A-Buzzer

/*
SparkFun Inventor's Kit
Circuit 2A - Buzzer

Play notes using a buzzer connected to pin 10

This sketch was written by SparkFun Electronics, with lots of help from the Arduino community.
This code is completely free for any use.

View circuit diagram and instructions at: https://learn.sparkfun.com/tutorials/sparkfun-inventor-kit-circuit-2a-buzzer
Download drawings and code at: https://github.com/sparkfun/SIK-Guide-Code
*/

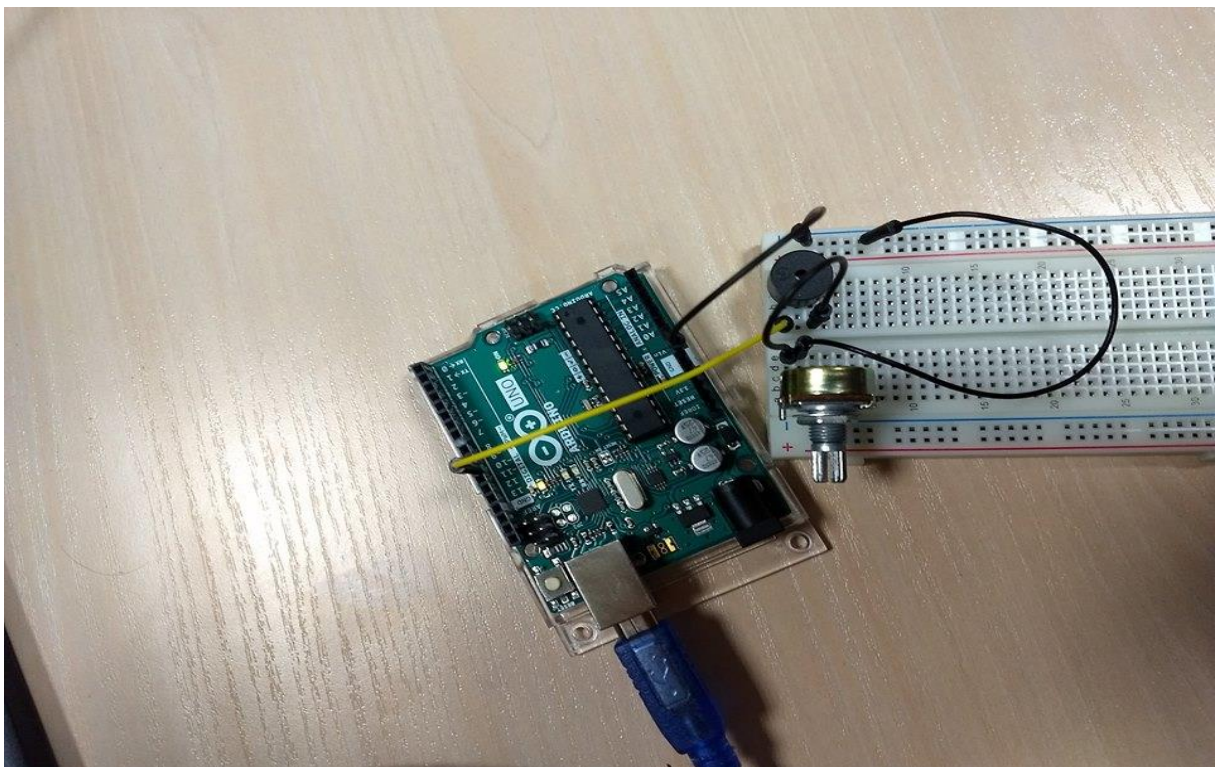
int speakerPin = 10;           //the pin that buzzer is connected to

void setup()
{
  pinMode(speakerPin, OUTPUT); //set the output pin for the speaker
}

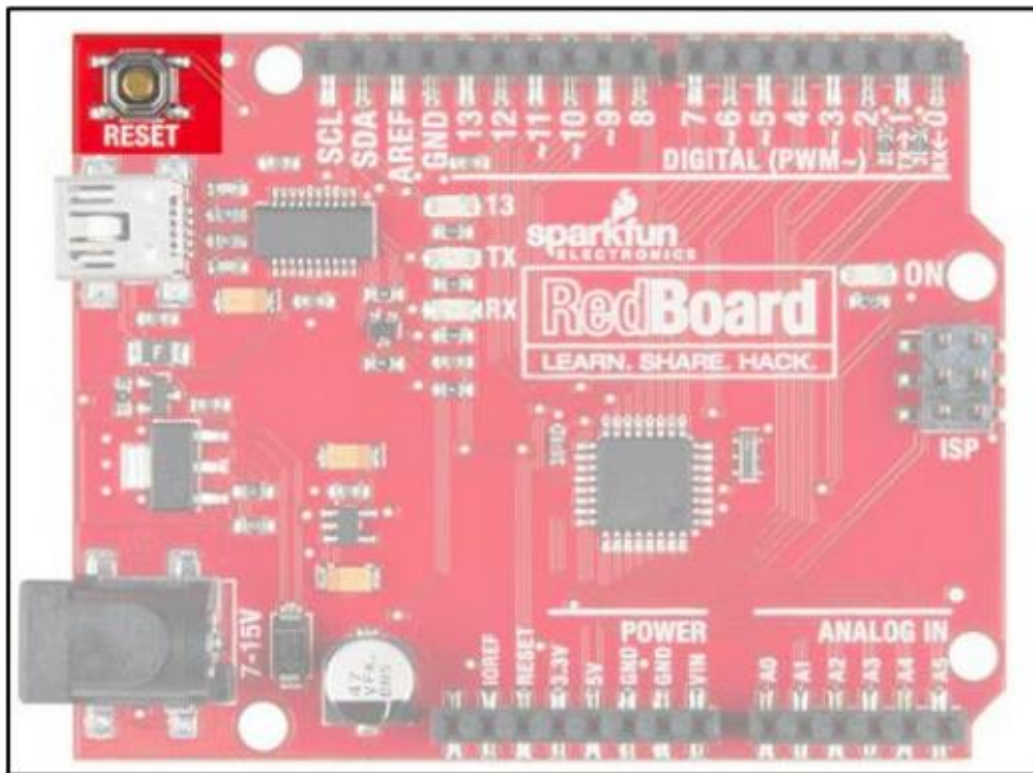
Done compiling.

Sketch uses 2984 bytes (9%) of program storage space. Maximum is 32256 bytes.
Global variables use 76 bytes (3%) of dynamic memory, leaving 1972 bytes for local variables. N

Arduino/Genuino Uno on COM4
```



Po przestaniu konfiguracji brzęczyk rozpocznie odtwarzanie utworu. Można obracać kółkiem na potencjometrze w celu regulacji tonu. Piosenka zostanie zagrana tylko raz, gdy będziemy chcieli usłyszeć ją ponownie należy nacisnąć guzik **RESET**.



- c) Przejrzyj kod. Którą funkcję i jej parametr należy skonfigurować, aby zmienić tempo utworu?

Aby dokonać zmiany tempa potrzebujemy w funkcji **void play(char note, int beats)** zmienić tablicę częstotliwości **int frequencies[]** która dopasowuje do każdej litery jej częstotliwość. Np. czwarta nuta to „f” z ustawioną częstotliwością 175, zmiana częstotliwości kolejnych liter zmieni nam tempo odtwarzania utworu.