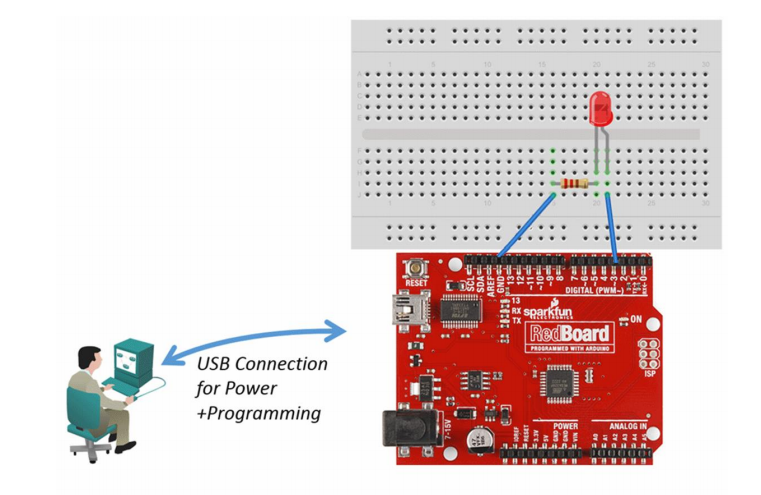
|  |  |
| --- | --- |
| **Politechnika Świętokrzyska w Kielcach Wydział Elektroniki, Automatyki i Informatyki** | |
| **Laboratorium IoT** | |
| Tematy: Blinking an LED using RedBoard and Arduino IDE oraz Buzzer | Grupa: **3ID15A**  Konrad Nowakowski  Michał Ortyl  Mateusz Orłowski |
|  |

Blinking an LED using RedBoard and Arduino IDE

1. Topologia



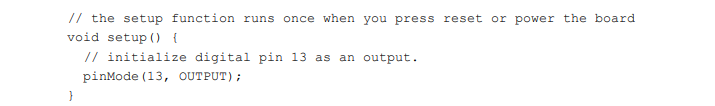
1. Cel zadania

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

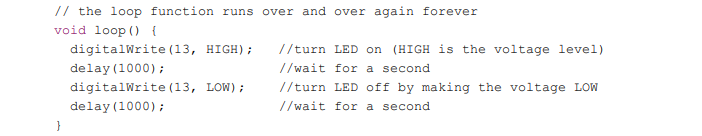
Aby wykonać ćwiczenie potrzebujemy takich urządzeń jak:

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

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

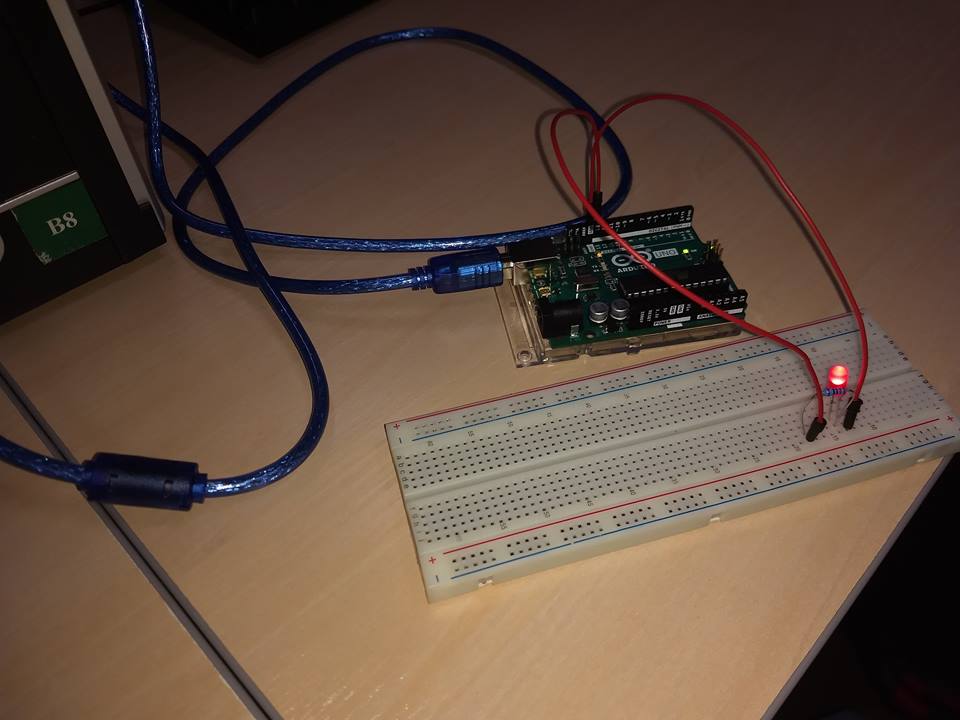


1. Aby dioda zaczęła świecić potrzebujemy poniższej funkcji:



Funkcja **digialWrite** ustawia wartość pinu 13 na wysoki. Pod funkcja 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.

1. Topologia działająca w praktyce



1. Następnie mieliśmy za zadanie uruchomić topologie, ale korzystając z portu 3. Aby wszystko zadziałało 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);*

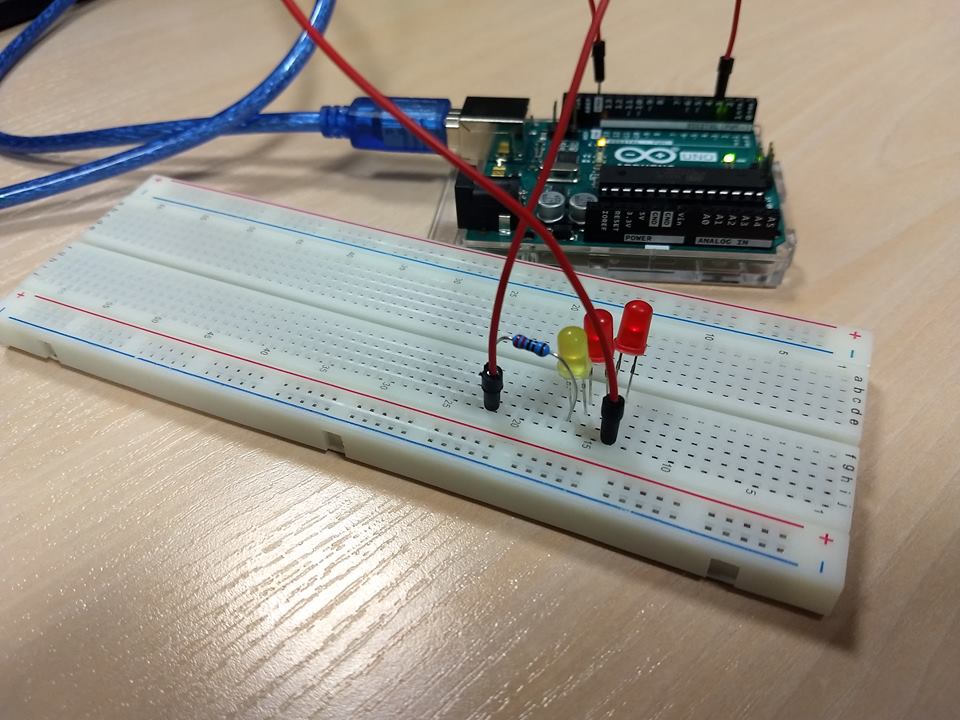
*delay(1000);*

*digitalWrite(3,LOW);*

*delay(1000);*

*}*

1. 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.

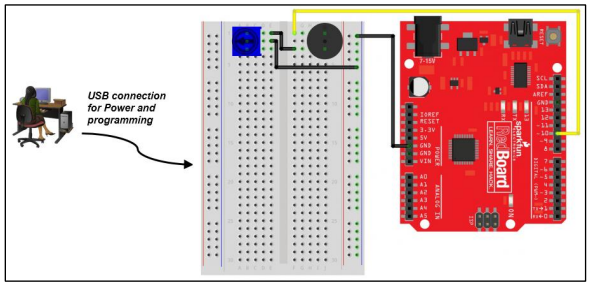


1. Jakie mogą nastąpić problemy, dla których dioda LED nie będzie świecić?

* może być niesprawna
* połączona może być w kierunku zaporowym
* układ został źle połączony

Buzzer

1. Topologia



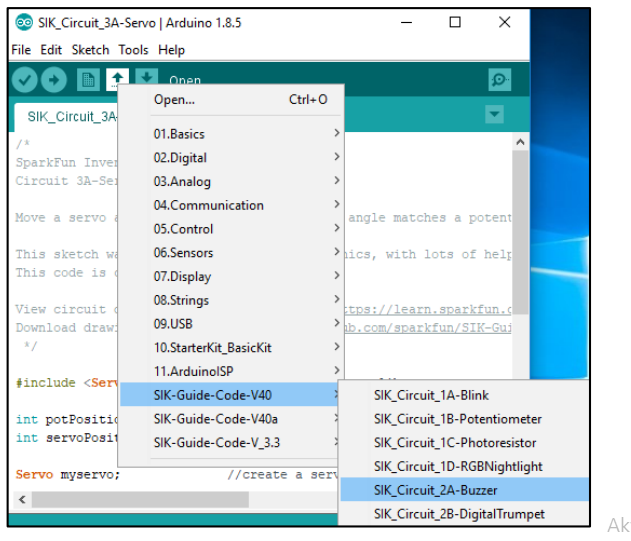
1. Cel zadania

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

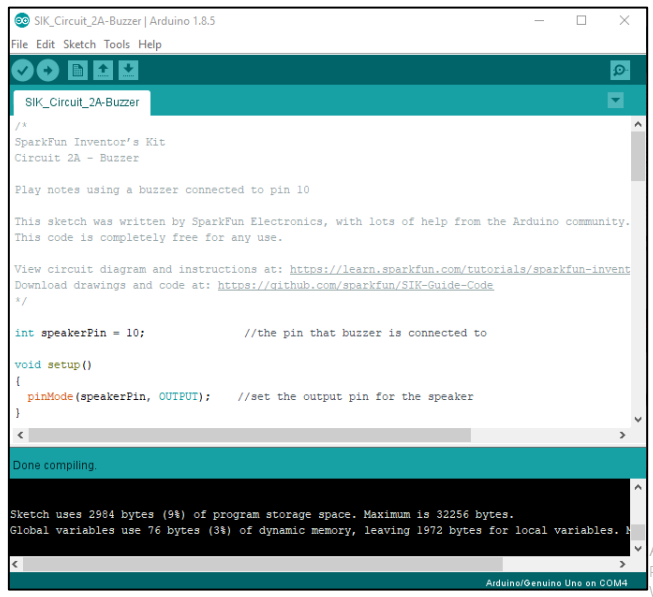
Aby wykonać ćwiczenie potrzebujemy takich narzędzi jak:

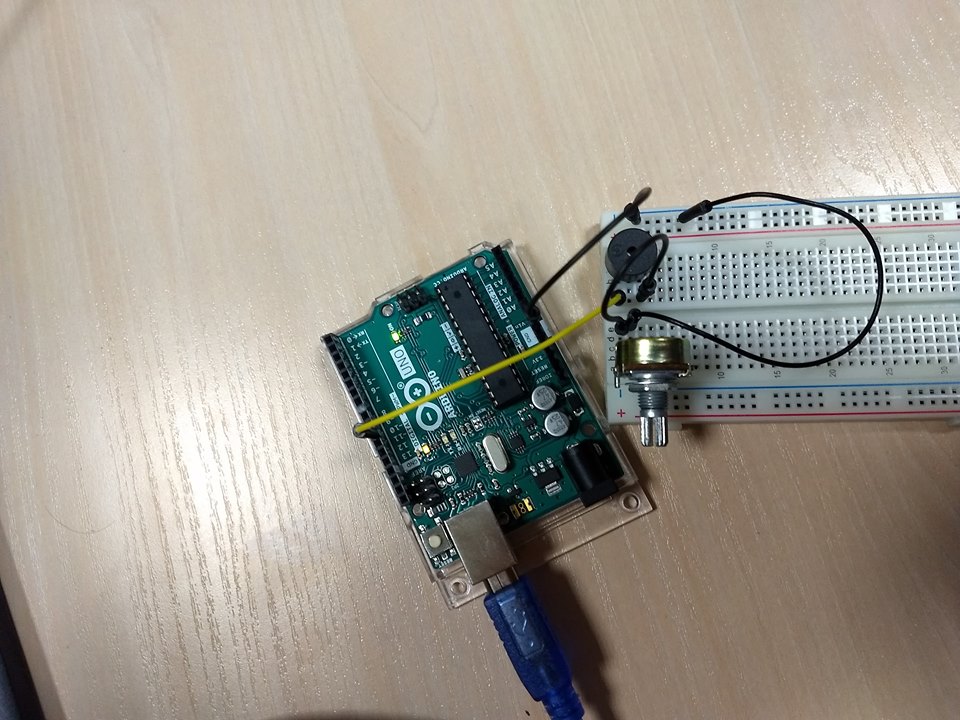
* Zestaw SparkFun,
* Komputer PC z Arduino
* Plik przykładowych kodów przewodnika SIK,
* przewody, buzzer oraz potencjometr.

1. Wykonanie
2. podłączenie urządzeń i przewodów według topologii
3. wybór odpowiedniego kodu(arduino) dla Buzzera

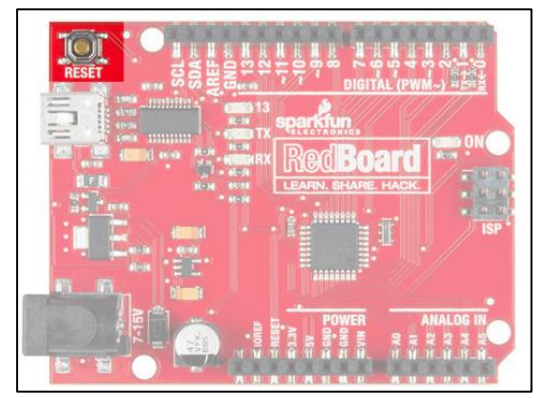


Upewniamy się czy nasz układ został dobrze podłączony, oraz czy jest on podłączony do komputera kablem USB. Następnie przesyłamy oprogramowanie od ReadBorder’a.





Po przesłaniu 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**.



1. Przejrzyj kod. Którą funkcje i jej parametr należy skonfigurować, aby zmienić tempo utworu?

Aby dokonać zmiany tempa potrzebujemy w funkcji **void play( char note, int beats)** zmienić tablice częstotliwości **int frequencies[]** która dopasowuje do każdej litery jej częstotliwość.

Np. czwarta nuta to „f” z ustawiona częstotliwością 175, zmiana częstotliwości kolejnych liter zmieni nam tempo odtwarzania utworu.