

Politechnika Świętokrzyska

Wydział Elektrotechniki, Automatyki i Informatyki

IoT - rozproszone sieci sensoryczne

Temat:

**Blinking an LED using
RedBoard and Arduino IDE &
Alternate Lab - Buzzer**

Wykonali:

Gaweł Cecot
Karolina Gałka
Dominik Łączkiewicz
Mateusz Sałata
Jakub Serwicki

Grupa: **3ID15A**

Nr laboratorium:

4

Data wykonania:

17.12.2018r.

Wstęp

Arduino IDE jest wieloplatformową aplikacją napisaną w języku Java, wydzieloną z IDE przygotowanego dla języka Processing i projektu Wiring. Środowisko jest zaprojektowane w taki sposób, aby było przyjazne dla hobbystów i osób niezajmujących się tworzeniem oprogramowania. IDE zawiera edytor kodu z takimi funkcjami, jak podświetlanie składni czy automatyczne wcięcia w kodzie, oraz pozwala na kompilację i wysłanie programu do płyty Arduino. Zazwyczaj nie ma potrzeby dodatkowej edycji plików Makefile lub uruchamiania programów z linii poleceń.

Płyta Arduino składa się z 8-bitowego mikrokontrolera Atmel AVR z uzupełniającymi elementami w celu ułatwienia programowania oraz włączenia innych układów. Ważnym aspektem jest standardowy sposób, w jaki wyprowadzone są złącza, pozwalające na podłączenie płyty z mikrokontrolerem do różnych wymiennych modułów dodatkowych (nazywanych *shieldami*). Oficjalne płyty Arduino wykorzystują układy z serii megaAVR, a konkretnie ATmega8, ATmega168, ATmega328, oraz ATmega1280 i ATmega 2560. Kilka innych układów jest wykorzystywanych w klonach Arduino. Większość płyt zawiera 5V regulator napięcia, 16 MHz rezonator kwarcowy (w niektórych odmianach rezonator ceramiczny), chociaż niektóre projekty, takie jak LilyPad działają na częstotliwości 8 MHz i rezygnują z regulatora napięcia na płycie ze względu na szczególne ograniczenia wielkości układu. Mikrokontroler Arduino jest wstępnie zaprogramowany z wykorzystaniem programu rozruchowego, co upraszcza przesyłanie programu do pamięci flash układu – w porównaniu do innych urządzeń, gdzie najczęściej potrzebny jest zewnętrzny programator mikrokontrolera.

Przebieg ćwiczenia

1. Konfiguracja Arduino

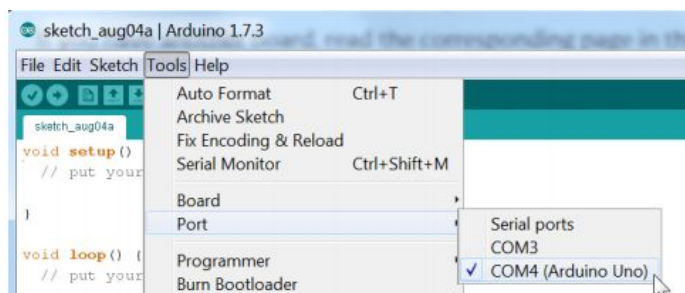
Do przeprowadzenia ćwiczenia potrzebowaliśmy następujących komponentów:

- oprogramowania Arduino IDE
- platformy SparkFun RedBoard (w skrócie jest to drukowana płytka pcb, która została zmodyfikowana w celu łatwego tworzenia prototypów obwodów)

Instalacja Arduino jest stosunkowo prosta, wystarczyło pobrać oprogramowanie ze strony <https://www.arduino.cc/en/Main/Software>, zainstalować je, a następnie podłączyć platformę SparkFun RedBoard do komputera za pomocą kabla mini usb B-A. Po podłączeniu platformy, w menedżerze urządzeń powinien pojawić się wpis o zainstalowanym sterowniku (Arduino UNO [Nr

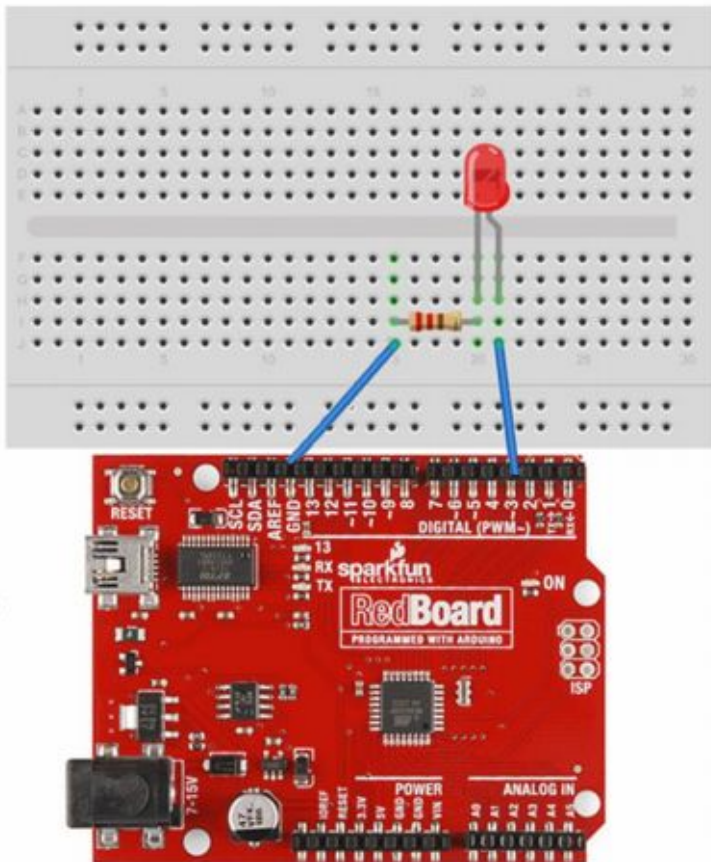
portu])

Pod koniec konfiguracji, należało w Arduino IDE ustawić odpowiedni port COM (co pozwala oprogramowaniu ustalić drogę przepływu instrukcji do platformy)



2. Lab - Blinking an LED using RedBoard and Arduino IDE

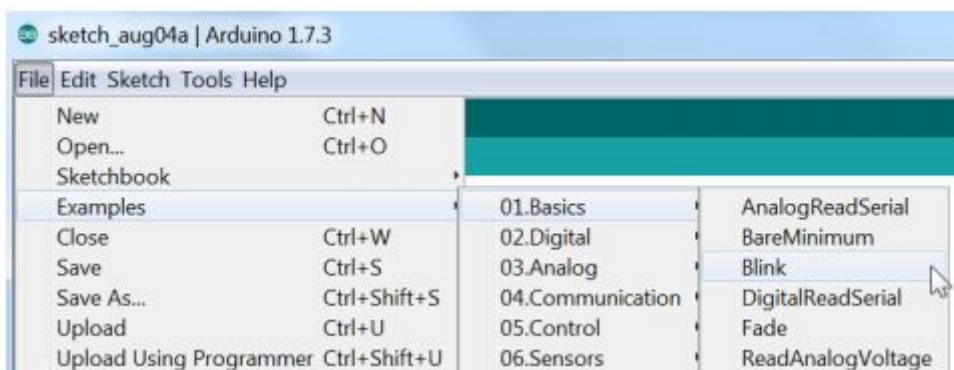
2.1. Topologia



Notka: Na zajęciach zastosowaliśmy rezystory 220 Ohm (Ω) (ze względu na brak rezystorów 330 Ohm (Ω) w sali laboratoryjnej)

Uwaga: Należało pamiętać, o każdorazowym otwieraniu obwodu w przypadku chęci zmiany/dodawania komponentów na płytce.

Po podłączeniu komponentów zgodnie z powyższym schematem, mogliśmy przejść do konfiguracji kodu w Arduino IDE.



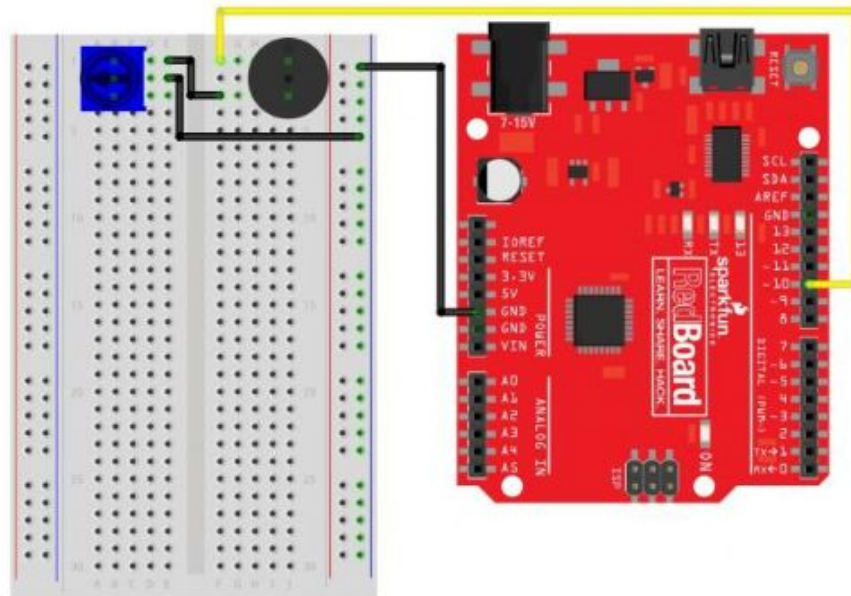
W Arduino IDE wybieramy odpowiedni kod wzorcowy (w tym przypadku "Blink")

Głównie działanie polegało na przesłaniu wysokiego napięcia na pin nr 13, odczekaniu sekundy, następnie przesłaniu niskiego napięcia na ten sam pin i ponowne odczekanie sekundy.

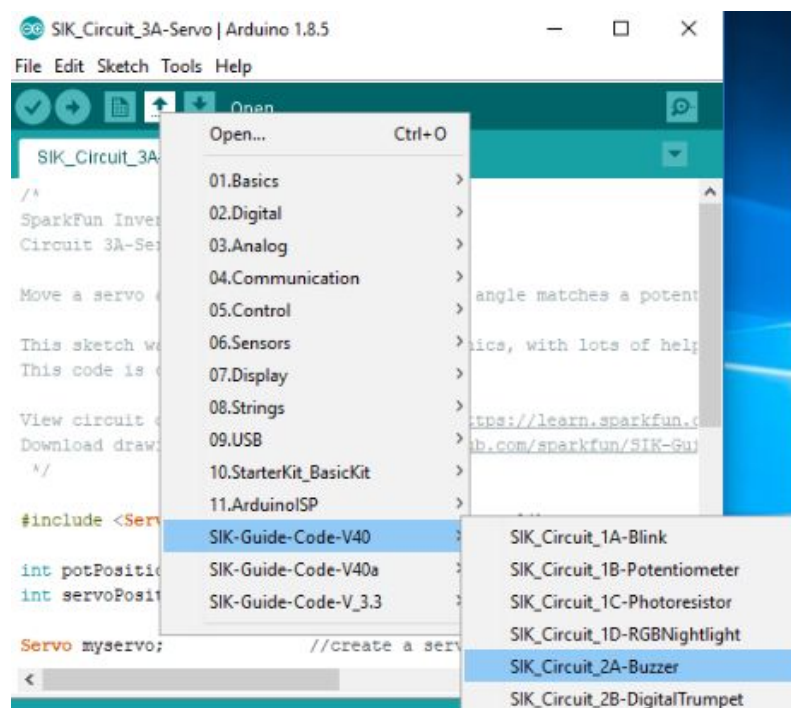
```
// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);  //turn LED on (HIGH is the voltage level)
  delay(1000);             //wait for a second
  digitalWrite(13, LOW);   //turn LED off by making the voltage LOW
  delay(1000);             //wait for a second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

3. Alternate Lab - Buzzer

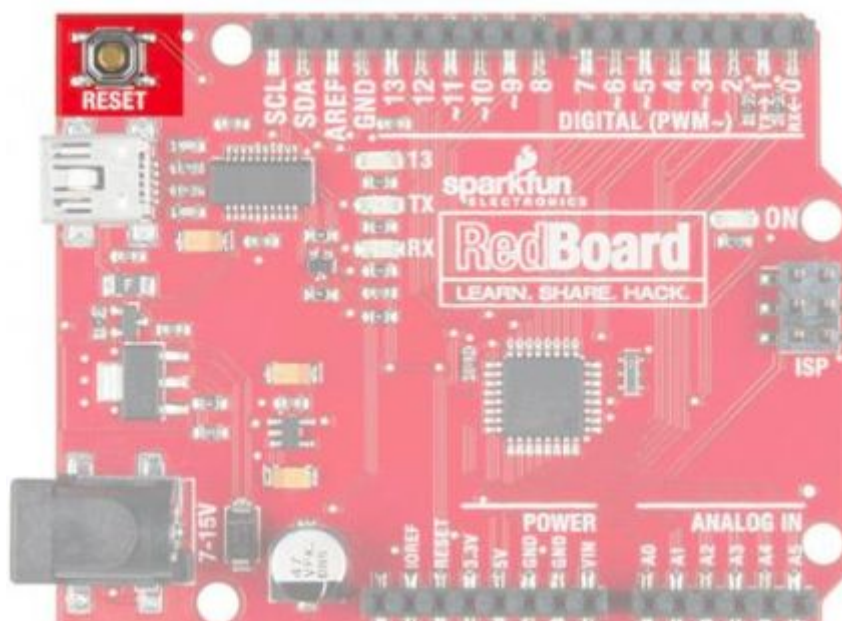
3.1. Topologia



Po podłączeniu komponentów zgodnie z powyższym schematem należało wejść na stronę <https://www.sparkfun.com/sikcode> i pobrać **SIK-Guide-Code-V40**. Aby SIK-Guide-Code-V40 był dostępny w Arduino IDE musimy umieścić to co pobraliśmy standardowo w C:\Program Files\Arduino\examples lub C:\Program Files (x86)\Arduino\examples. Gdy wykonaliśmy wszystko poprawnie to w Arduino IDE po kliknięciu na strzałkę do góry w rozwijanym menu powinniśmy zobaczyć SIK-Guide-Code-V40 a gdy rozwiniemy dalej tą opcję to powinniśmy wybrać SIK_Circuit_2A-Buzzer.



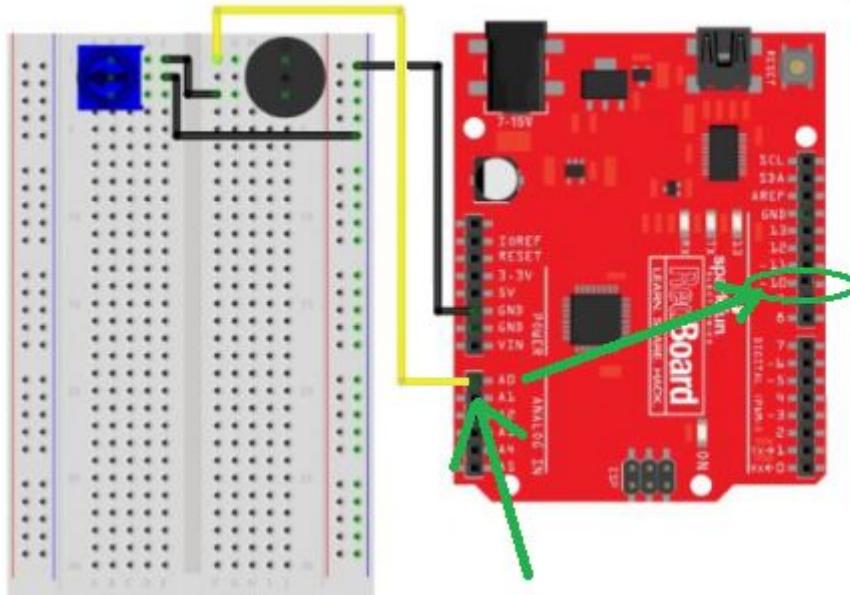
Po załadowaniu tego domyślnego kodu oraz uruchomieniu usłyszeliśmy brzęczenie buzzera którego głośność mogliśmy zmienić za pomocą potencjometru. "Utwór" został odtworzony tylko raz, aby ponownie go odsłuchać należy przycisnąć przycisk reset na płytce Arduino



Wnioski

- Zauważyliśmy że trzeba zwrócić uwagę na to jak podłączona jest dioda - ważne jest aby nie była w kierunku zaporowym.(jeśli przyłożone napięcie wsteczne przekroczy pewną graniczną wartość, wówczas z powodu przebiecia lawinowego następuje szybki spadek rezystancji i w efekcie uszkodzenie elementu) Wyjątkiem jest dioda Zenera gdzie istnieje możliwość przewodzenia prądu w kierunku zaporowym.
- Próba podłączenia 3 diod szeregowo zakończyła się niepowodzeniem (diody nie świeciły) **Wyjaśnienie:** Rezystancja zastępcza układu szeregowego jest sumą rezystancji n-elementów. ($R_z = R_1 + R_2 + \dots + R_n$). Odnosząc się do naszych testów oznacza to, że z powodu zbyt wysokiej rezystancji prąd nie płynął przez diody.
- Próba podłączenia 3 diod równolegle zakończyła się powodzeniem, wszystkie diody świeciły. **Wyjaśnienie:** Rezystancja zastępcza układu równoległego jest ilorazem rezystancji i n-elementów ($R_z = R/n$) Oznacza to ,że rezystancja układu nie zmieniła się przez co prąd mógł płynąć swobodnie przez diody .
- Gdy wykonaliśmy ćwiczenie w którym korzystaliśmy z potencjometru oraz buzzera zauważyliśmy że wraz ze wzrostem wartości na potencjometrze, buzzer głośniej brzęczał.

- Podczas wykonywania [Alternate Lab - Buzzer](#) początkowo układ połączyliśmy następująco:



W wyniku takiej topologii Buzzer nie chciał działać. **Wyjaśnienie:** Jeżeli do Arduino są podłączone jakieś analogowe sensory, to zawsze powinniśmy pamiętać aby po jednej stronie połączony był sensor analogowy (w tym przypadku pin 10), natomiast napięcie po przeciwnej stronie. W innym przypadku opór i indukcyjność uziemienia spowoduje, że źródło napięcia zacznie wywierać zakłócenia na wejścia analogowe, czego następstwem jest zakłócenie lub brak działania sensorów analogowych.