

# Programowanie urządzeń Internetu rzeczy IoT

## Ćwiczenia laboratoryjne 1

Prowadzący: *dr inż. Hubert Zarzycki*

### Zakres ćwiczeń

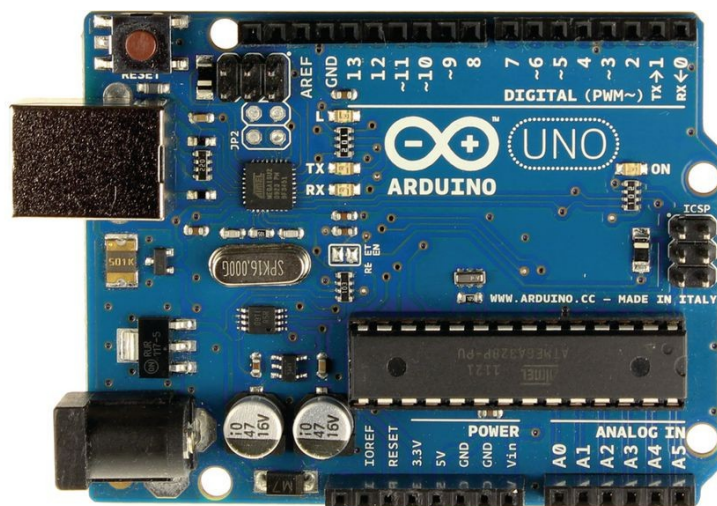
Podstawy mikrokontrolera Arduino Uno. Pierwsze programy i zastosowania. Zapoznanie się z podstawowymi operacjami.

### Podstawowe zagadnienia

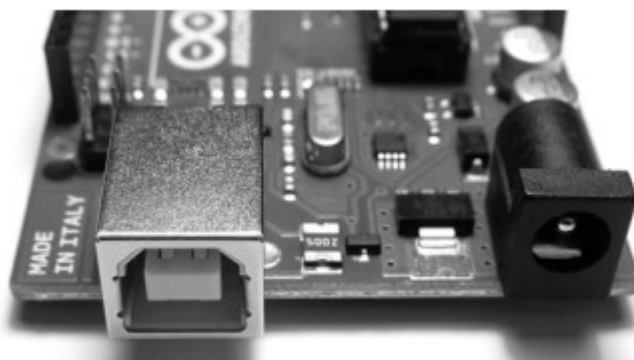
#### Arduino

Arduino jest platformą open source do tworzenia prototypów elektronicznych zbudowaną na bazie elastycznego, łatwego w użyciu sprzętu i oprogramowania. Platformę stworzono z myślą o projektantach, hobby-stach i wszystkich, którzy są zainteresowani budową interaktywnych obiektów lub środowisk.

Tak więc, Arduino jest miniaturowym systemem komputerowym, który można programować za pomocą odpowiednich instrukcji w celu operowania na danych wejściowych i wyjściowych w najróżniejszych formach.



Dzięki możliwości łączenia tego systemu z niemal wszystkimi dostępnymi urządzeniami wejściowymi i wyjściowymi (czujnikami, wskaźnikami, wyświetlaczami, silnikami itp.) konstruktor może zaprogramować wszystkie interakcje niezbędne do stworzenia funkcjonalnego rozwiązania

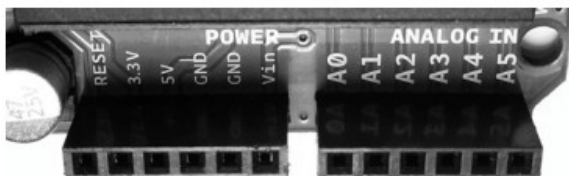


Z lewej strony zamontowano złącze USB. Złącze to spełnia trzy ważne zadania: zapewnia zasilanie płytki, wysyła instrukcje do systemu Arduino lub wysyła dane z i do komputera. Po prawej stronie zainstalowano złącze zasilania. Za pośrednictwem tego złącza można zasilать płytkę Arduino, stosując standardowy zasilacz sieciowy.

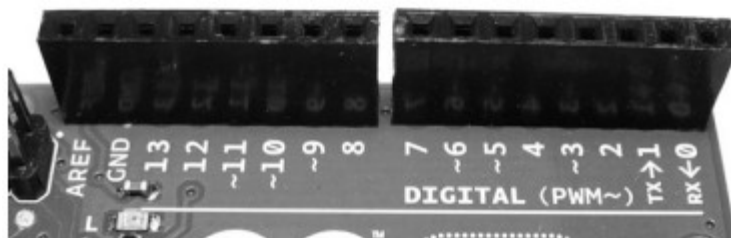


## Programowanie urządzeń Internetu rzeczy IoT

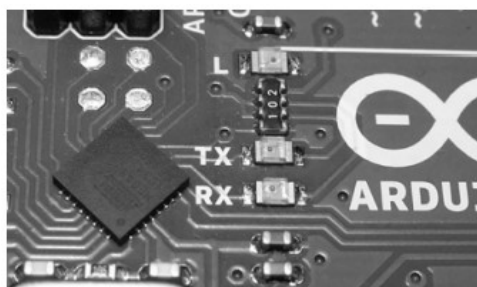
Mikrokontroler to miniaturowy komputer obejmujący procesor (odpowiedzialny za wykonywanie rozkazów), zawierający różne typy pamięci (w których są przechowywane dane i instrukcje zapisane w szkicach) oraz obsługujący wiele kanałów wysyłania i otrzymywania danych



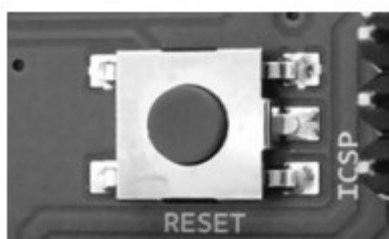
Bezpośrednio pod mikrokontrolerem znajdują się dwa zestawy gniazd. Gniazda zasilania i analogowe umożliwiają połączenia zasilania i umożliwiają podłączenie zewnętrznego przycisku RESET. Drugi wiersz oferuje sześć wejść analogowych przydatnych podczas mierzenia sygnałów elektrycznych ze zmieniającym się napięciem. Piny A4 i A5 mogą dodatkowo być używane także do wysyłania danych do innych urządzeń i odbierania danych od tych urządzeń.



W zestawie pinów wejść i wyjść cyfrowych gniazda oznaczone etykietami od 0 do 13 pełnią funkcję pinów wejściawyjścia (I/O). Za pośrednictwem tych pinów można albo wykrywać sygnały elektryczne, albo generować te sygnały na potrzeby innych urządzeń. Piny oznaczone numerami 0 i 1 dodatkowo pełnią funkcję portu szeregowego, który umożliwia wysyłanie danych do innych urządzeń i odbieranie danych od tych urządzeń (podobnie jak w przypadku złącza USB używanego do komunikacji z komputerem). Piny oznaczone znakiem tyldy (~) mogą dodatkowo generować sygnały elektryczne o różnych napięciach, które są przydatne na przykład podczas generowania efektów świetlnych lub sterowania silnikami elektrycznymi.



Na płytce można też znaleźć bardzo przydatne elementy nazywane diodami LED (od ang. light-emitting diodes). Te niewielkie urządzenia świecą, gdy przepływa przez nie prąd. Na płytce Arduino zainstalowano cztery diody LED: jedną z prawej strony (oznaczoną etykietą ON i wskazującą zasilanie całej płytki) oraz trzy w odrębnej grupie



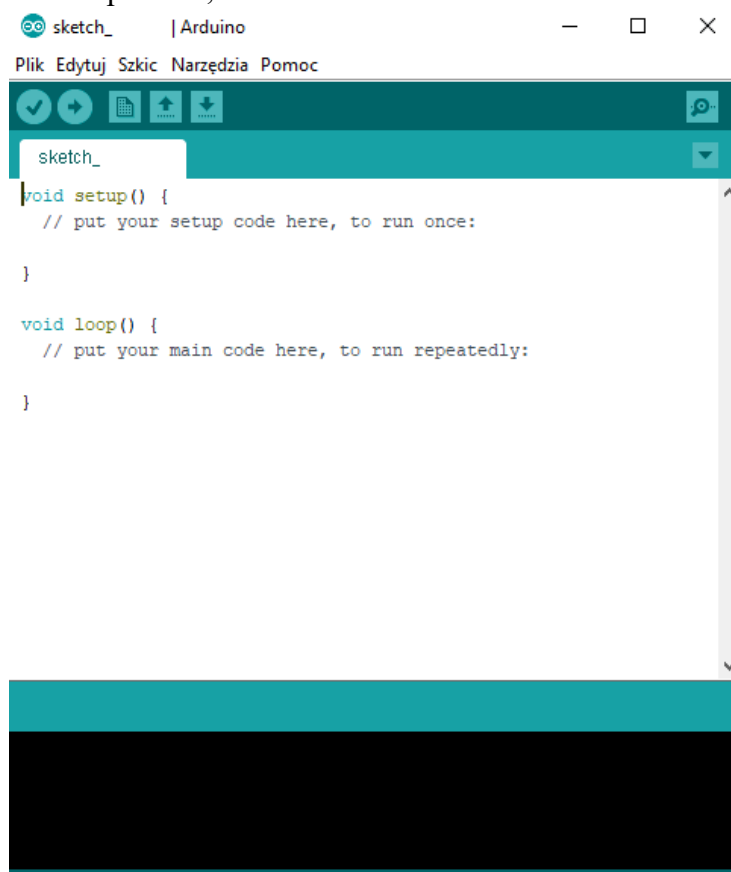
# Programowanie urządzeń Internetu rzeczy IoT

Przycisk RESET na płytce Arduino (patrz rysunek powyżej) służy do ponownego uruchamiania systemu. Szczególnie przydatny w przypadku wystąpienia poważnych błędów.

Jedną z największych zalet systemu Arduino jest łatwość jego roszszerzania, czyli dodawania fizycznych urządzeń wprowadzających nowe funkcje. Dwa wiersze gniazd po obu stronach płytki Arduino umożliwiają podłączanie do systemu tzw. modułów, czyli dodatkowych płytek z odpowiednio rozmierzonymi pinami.

## Środowisko programowania Arduino

Środowisko programowania Arduino IDE przypomina prosty edytor tekstu. Okno środowiska zawiera trzy główne obszary: obszar poleceń, obszar tekstu i obszar komunikatów



Pasek menu zawiera polecenia: Plik, Edytuj, Szkic, Narzędzia, Pomoc

Główną część programu stanowi okno / pole tekstowe, gdzie tworzy się szkic (kod). Poniżej znajduje się niebieski i czarny pasek, gdzie wyświetlane są komunikaty, np. podczas sprawdzania programu oraz kompilowania i wgrywania na płytkę Arduino.

W celu uruchomienia płytki Arduino należy

- podłączyć Arduino UNO do komputera przez załączony w zestawie kabel USB,
- uruchomić program Arduino IDE,
- ustawić w programie odpowiedni model płytki:

Narzędzia > Płytki > wybieramy Arduino Uno

- wybieramy odpowiedni port szeregowy:

Narzędzia > Port > np. COM3 (Arduino/Genuino Uno)

- tworzymy nowy szkic lub otwieramy przykładowy program z wgranych wcześniej bibliotek: Plik > Przykłady > ....

Szkic (sketch) systemu Arduino ma postać zbioru instrukcji przygotowanych z myślą o realizacji określonego zadania.

### Tworzenie szkicu

Warto zacząć szkic od komentarza, by wskazać do czego służy program oraz kto jest jego autorem.

# Programowanie urządzeń Internetu rzeczy IoT

Komentarz liniowy zaczyna się od znaków // i obowiązuje do końca danej linii.

Komentarze blokowe rozpoczyna się od znaków /\* i kończy znakami \*/. Wszystko co znajduje się pomiędzy tymi znakami nie jest kompilowane. Komentarzy blokowych nie wolno zagnieżdżać.

```
/*  
Szkic Arduino Diody LED  
Autor: Jan Kowalski; 2022-03-05  
*/
```

## **Funkcja setup()**

Następnym krokiem w procesie tworzenia każdego szkicu jest dodanie funkcji void setup(). Funkcja zawiera zbiór instrukcji wykonywanych przez system Arduino tylko raz — bezpośrednio po włączeniu lub ponownym uruchomieniu.

```
void setup() {  
  // put your setup code here, to run once:  
}
```

Poniżej znajduje się przykładowy program wymuszenia migania diody LED na płytce Arduino. LED jest połączona z pinem cyfrowym nr 13 platformy Arduino. Pin cyfrowy może służyć albo do wykrywania sygnału elektrycznego, albo do generowania takiego sygnału na podstawie wydawanych poleceń. W tym projekcie będzie generowany sygnał elektryczny włączający diodę LED.

```
void setup() {  
  pinMode(13, OUTPUT);  
}
```

Liczba 13 na powyższym listingu reprezentuje numer portu cyfrowego, którego dotyczy ta instrukcja. W tym przypadku program włącza tryb danych wyjściowych (OUTPUT) dla pinu nr 13, zatem wskazany pin będzie służył do generowania sygnałów elektrycznych. Gdyby pin miał służyć do wykrywania przychodzących sygnałów elektrycznych, należałoby zastosować tryb INPUT

## **Funkcja loop()**

System Arduino wykonuje wielokrotnie instrukcje zapisane w funkcji loop(). Wykonanie szkicu następuje aż do momentu odłączenia zasilania lub naciśnięcia przycisku RESET.

```
void loop() {  
  // put your main code here, to run repeatedly:  
}
```

W celu spowodowania migania diody wprowadzamy do pętli loop() następujące instrukcje.

```
void loop() {  
  digitalWrite(13, HIGH); // włączenie pinu cyfrowego nr 13  
  delay(1000); // wstrzymanie działanie przez 1 sekundę  
  digitalWrite(13, LOW); // wyłączenie pinu cyfrowego nr 13  
  delay(1000); // wstrzymanie działanie przez 1 sekundę  
}
```

Funkcja digitalWrite() steruje napięciem pinu cyfrowego (w tym przypadku pinu nr 13 połączanego z diodą LED). Przekazanie za pośrednictwem drugiego parametru tej funkcji wartości HIGH powoduje ustawienie tzw. stanu wysokiego, w którym przez pin przepływa prąd, powodując świecenie diody LED. Wywołanie tej samej funkcji z parametrem LOW spowoduje odcięcie zasilania i wyłączenie diody LED. Szkic wstrzymuje działania na 1 sekundę (1000 milisekund) kiedy uruchomione jest polecenie delay(1000).

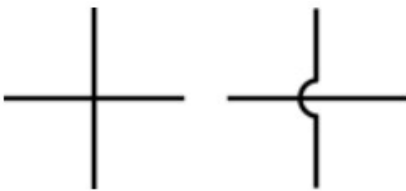


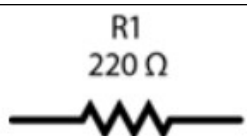
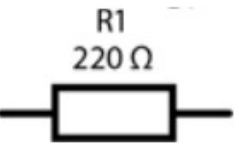
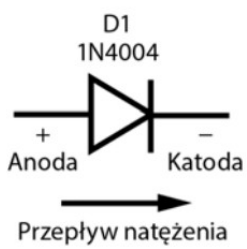
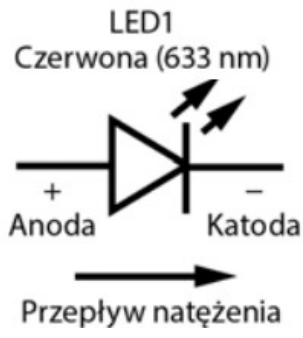
Po utworzeniu skryptu należy zachować szkic. Arduino IDE nie zapisuje szkiców automatycznie.

W celu sprawdzenia poprawności kodu należy wybrać opcję Weryfikuj z menu i sprawdzić poprawność wykonania.

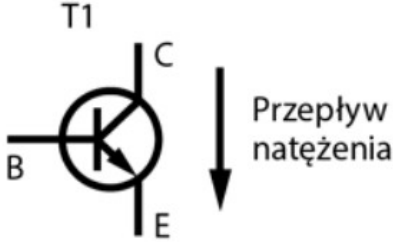
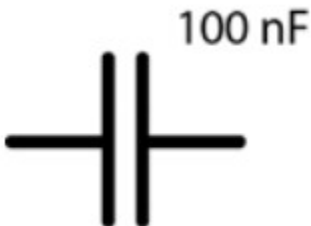
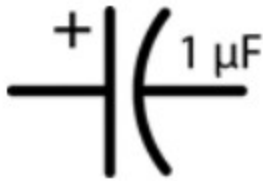
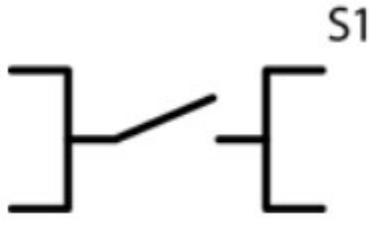
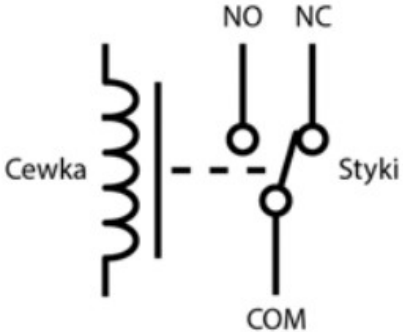
# Programowanie urządzeń Internetu rzeczy IoT

## Symbole komponentów

Poniższa tabela ukazuje symbole reprezentujące różne komponenty oraz sposoby graficznego reprezentowania połączeń

Symbol	Opis
	Niepołączone przecinające się przewody
	Połączone przewody
	Uziemienie
 	Rezystor. Oprócz samego symbolu rezystora dobrą praktyką jest umieszczanie na schemacie wartości rezystancji i numeru komponentu (w tym przypadku 220 Ω i R1)
	Dioda prostownicza. Diody prostownicze są komponentami spolaryzowanymi, a prąd przepływa od anody do katody.
	Symbol diody LED. Symbole diod LED dodatkowo zawierają dwie równoległe strzałki sygnalizujące emitowane światło.

## Programowanie urządzeń Internetu rzeczy IoT

	<p>Tranzystor. Pionowa linia na górze tego symbolu (oznaczona literą C) reprezentuje kolektor, pozioma linia z lewej strony (oznaczona literą B) reprezentuje bazę, a pionowa linia na dole (oznaczona literą E) reprezentuje emiter. Strzałka w środkowej części symbolu (skierowana w prawą stronę w dół) wskazuje, że jest to tranzystor typu NPN, ponieważ tylko w tranzystorach NPN prąd może przepływać od kolektora do emitera. (W tranzystorach PNP prąd może przepływać od emitera do kolektora).</p>
	<p>Symbol niespolaryzowanego kondensatora (pojemność kondensatora należy zapisać w prawym górnym rogu symbolu). Kondensatory ceramiczne, podobne do tego pokazanego na rysunku, są bardzo małe i mogą utrzymywać niewielkie ładunki elektryczne. Kondensatory tego typu nie są spolaryzowane, zatem prąd może przez nie przechodzić w obu kierunk</p>
	<p>Symbol kondensatora spolaryzowanego. Symbol kondensatora elektrolitycznego zawiera symbol plusa (+) wskazujący polaryzację kondensatora. Kondensatory elektrolityczne często są używane do utrzymywania większych ładunków elektrycznych do stabilizacji napięcia dostarczanego przez zasilacz. Kondensatory tego typu, niczym niewielkie, tymczasowe baterie, mogą z powodzeniem służyć do stabilizacji napięcia w bezpośrednim sąsiedztwie obwodów lub części narażonych na chwilowe przerw w dostarczaniu prądu przez zasilacz.</p>
	<p>Symbol przycisku. Po naciśnięciu przycisku obie strony są łączone, zatem przycisk przepuszcza napięcie i natężeni</p>
	<p>PrzełącznikSymbole przełączników mają różne formy i mogą obejmować odmienne zestawy styków. Pewne elementy są jednak wspólne dla wszystkich symboli przełączników. Pierwszym takim elementem jest cewka (reprezentowana przez pionową krzywą po lewej stronie). Drugim elementem wspólnym są styki przełącznika. Styk COM (wspólny, od ang. common) zwykle jest używany jako wejście, natomiast styki oznaczone etykietami NO (standardowo otwarty, od ang. normally open) i NC (standardowo zamknięty, od ang. normally closed) najczęściej są używane jako wyjścia</p>

# Programowanie urządzeń Internetu rzeczy IoT

## Zadania

1. Dodaj do programu przykładowe komentarze blokowe. Sprawdź czy linie z komentarzem dadzą się debugować.

/\*

Szkic Arduino Diody LED

Autor: Jan Kowalski; 2022-03-05

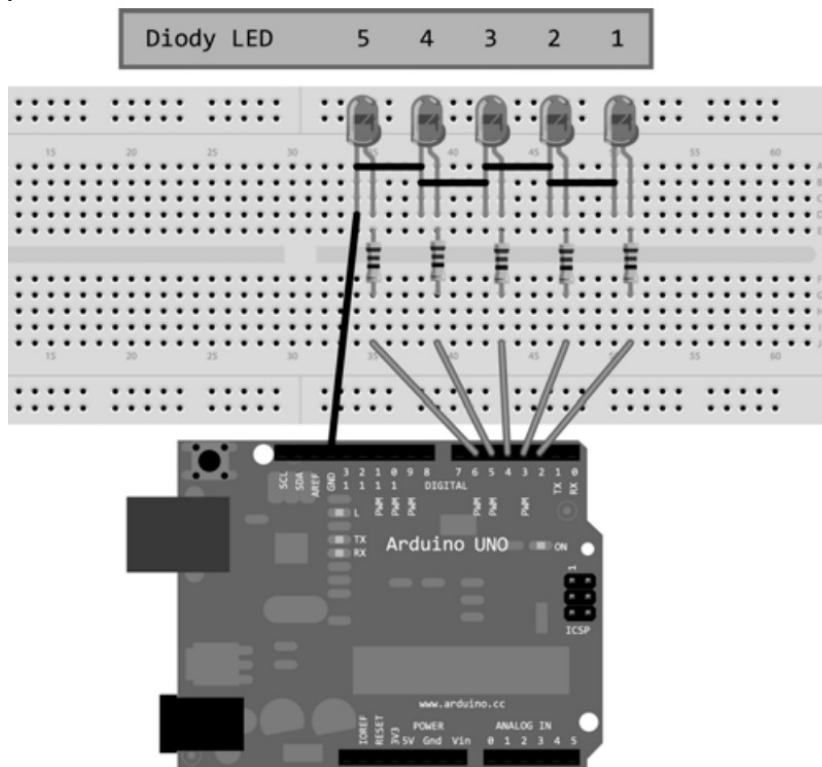
\*/

2. Proszę utworzyć rozwiązanie i uruchomić kod na podstawie poniższego schematu i opisu.

Oto algorytm przygotowany dla tego projektu:

- Włącz diodę LED nr 1.
- Odczekaj pół sekundy.
- Wyłącz diodę LED nr 1.
- Włącz diodę LED nr 2.
- Odczekaj pół sekundy.
- Wyłącz diodę LED nr 2.
- Kontynuuj do momentu włączenia diody LED nr 5, a następnie odwróć ten proces, tak aby były włączane diody LED od 5. do 1.
- Powtarzaj ten proces w nieskończoność.

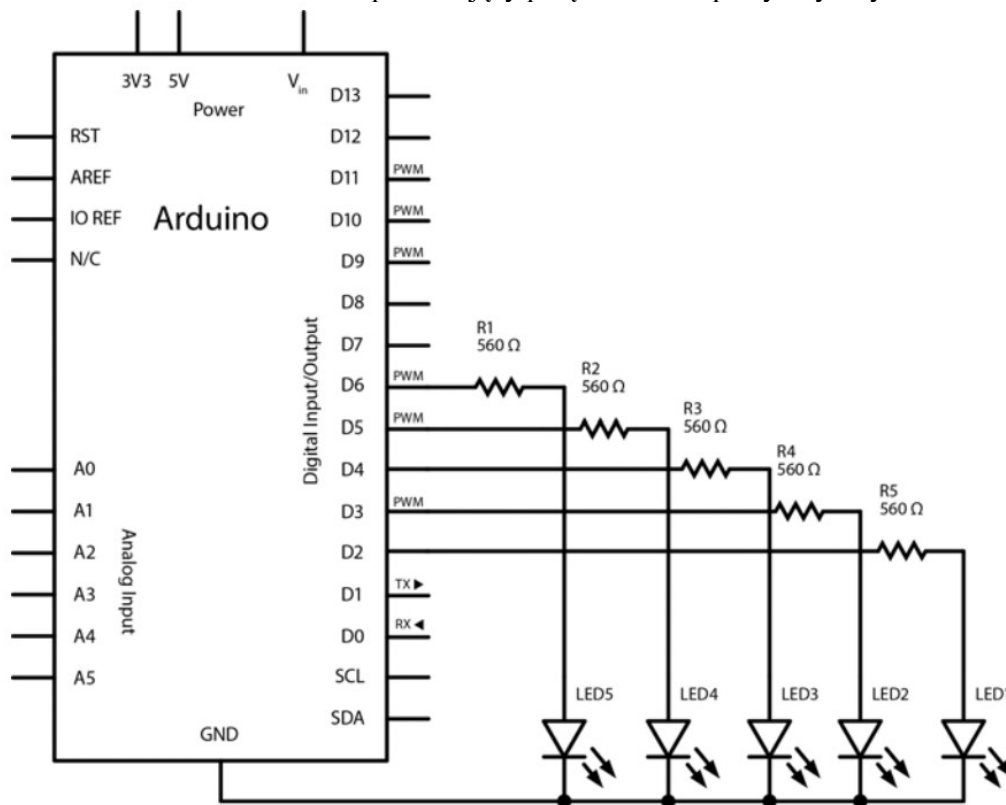
Poniżej widok połączeń:





## Programowanie urządzeń Internetu rzeczy IoT

Poniżej schemat obwodu dla zadania 2 odpowiadający połączeniom w powyższym rysunku.



Przykładowy kod:

```
void setup() {  
    pinMode(2, OUTPUT); // pin sterujący pierwszą diodą LED jest ustawiany  
    // jako wyjściowy  
    pinMode(3, OUTPUT); // to samo należy zrobić dla diod LED od 2 do 5  
    pinMode(4, OUTPUT);  
    pinMode(5, OUTPUT);  
    pinMode(6, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(2, HIGH); // włącza 1 diodę LED  
    delay(500); // oczekiwanie 0.5 sekundy  
    digitalWrite(2, LOW); // wyłącza 2 diodę LED  
    digitalWrite(3, HIGH); // w ten sam sposób włączane i wyłączone są diody LED od 2  
    // do 5  
    delay(500);  
    digitalWrite(3, LOW);  
    digitalWrite(4, HIGH);  
    delay(500);  
    digitalWrite(4, LOW);  
    digitalWrite(5, HIGH);  
    delay(500);  
}
```



## Programowanie urządzeń Internetu rzeczy IoT

```
digitalWrite(5, LOW);  
digitalWrite(6, HIGH);  
delay(500);  
digitalWrite(6, LOW);  
digitalWrite(5, HIGH);  
delay(500);  
digitalWrite(5, LOW);  
digitalWrite(4, HIGH);  
delay(500);  
digitalWrite(4, LOW);  
digitalWrite(3, HIGH);  
delay(500);  
digitalWrite(3, LOW);  
}
```

3. Przygotuj zadanie nr 2 przy wykorzystaniu pętli for.
4. Utwórz projekt z zastosowaniem przycisku, którego włączenie spowoduje zapalenie diody LED na jedną sekundę.

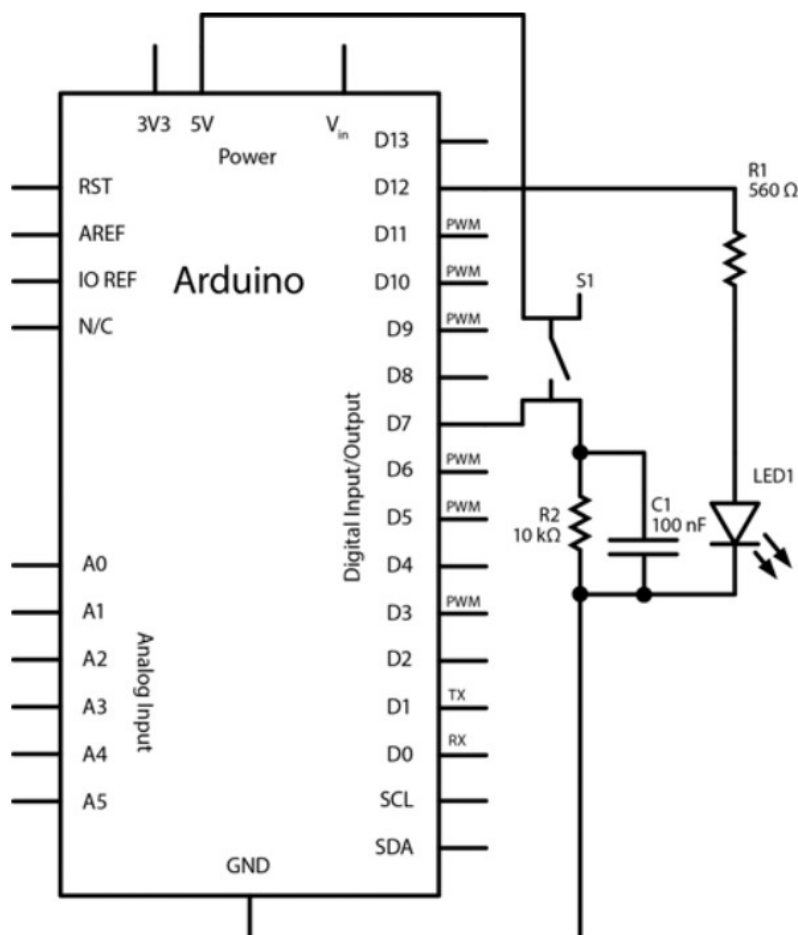
Algorytm Oto algorytm dla tego projektu:

- Sprawdź, czy przycisk został wciśnięty.
- Jeśli przycisk jest wciśnięty, włącz diodę LED na 1 sekundę, a następnie wyłącz tę diodę.
- Jeśli przycisk nie został wciśnięty, nie wykonuj żadnych działań.
- Powtarzaj ten proces w nieskończoność.

Lista urządzeń potrzebnych do realizacji tego projektu:

- jeden przycisk;
- jedna dioda LED;
- jeden rezystor  $560\ \Omega$ ;
- jeden rezystor  $10\ k\Omega$ ;
- jeden kondensator  $100\ nF$ ;
- różne przewody;
- płytki uniwersalna;
- płytki Arduino i przewód USB.

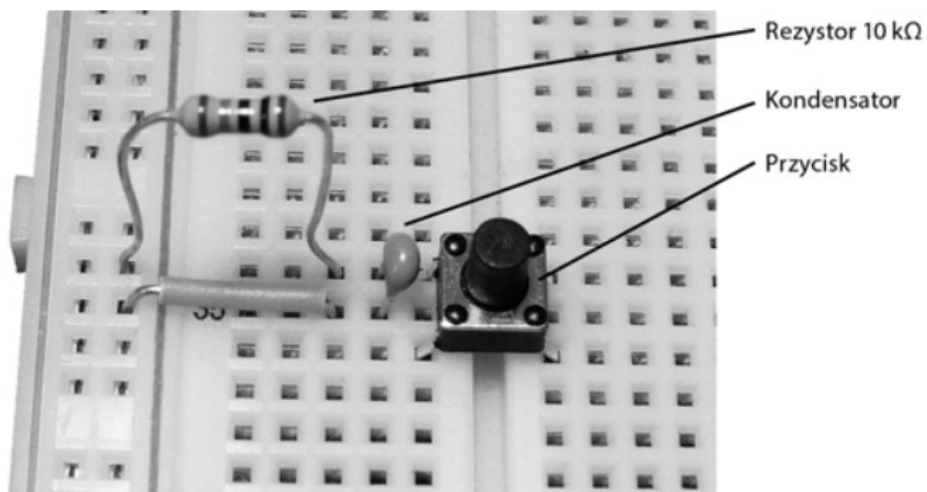
Schemat obwodu:



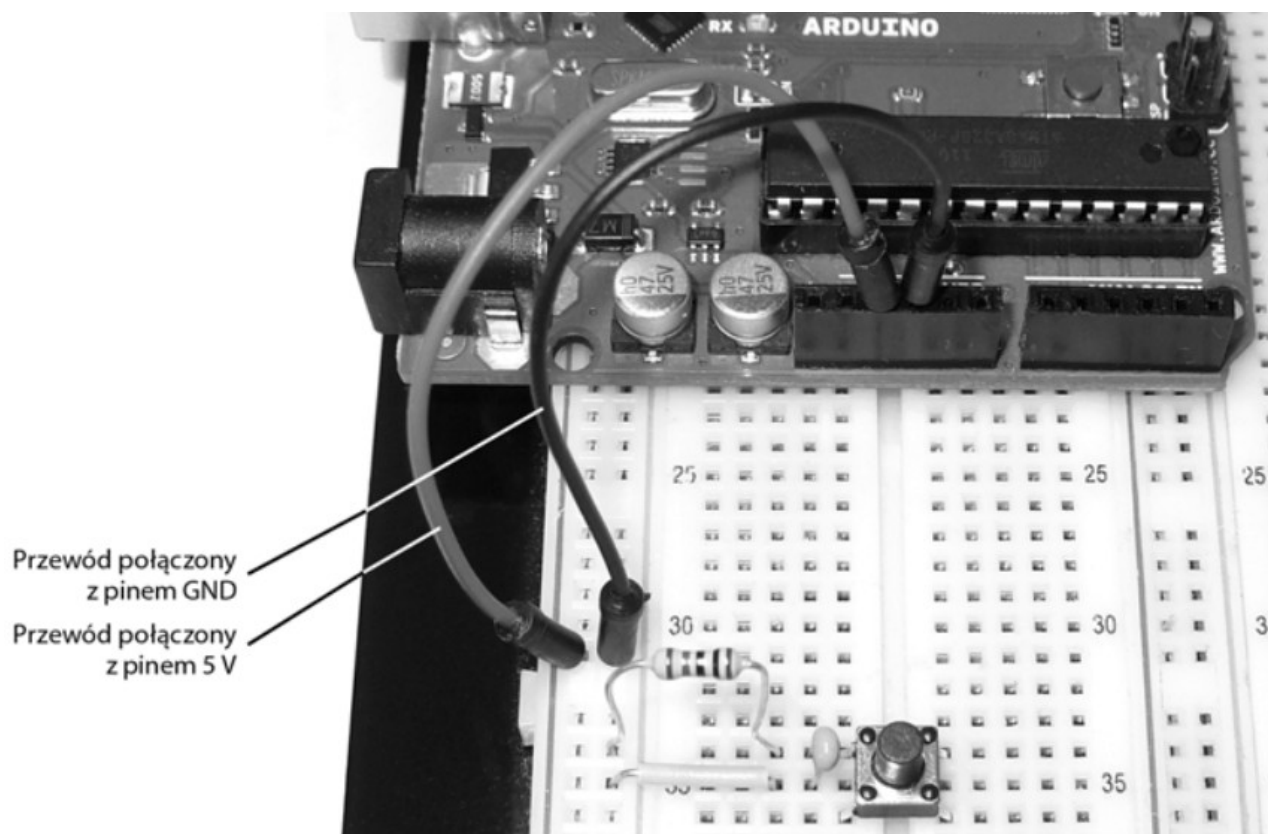
## Programowanie urządzeń Internetu rzeczy IoT

5.

Poniżej sposób połączenia rezystora 10k k $\Omega$ , kondensatora i przycisku.

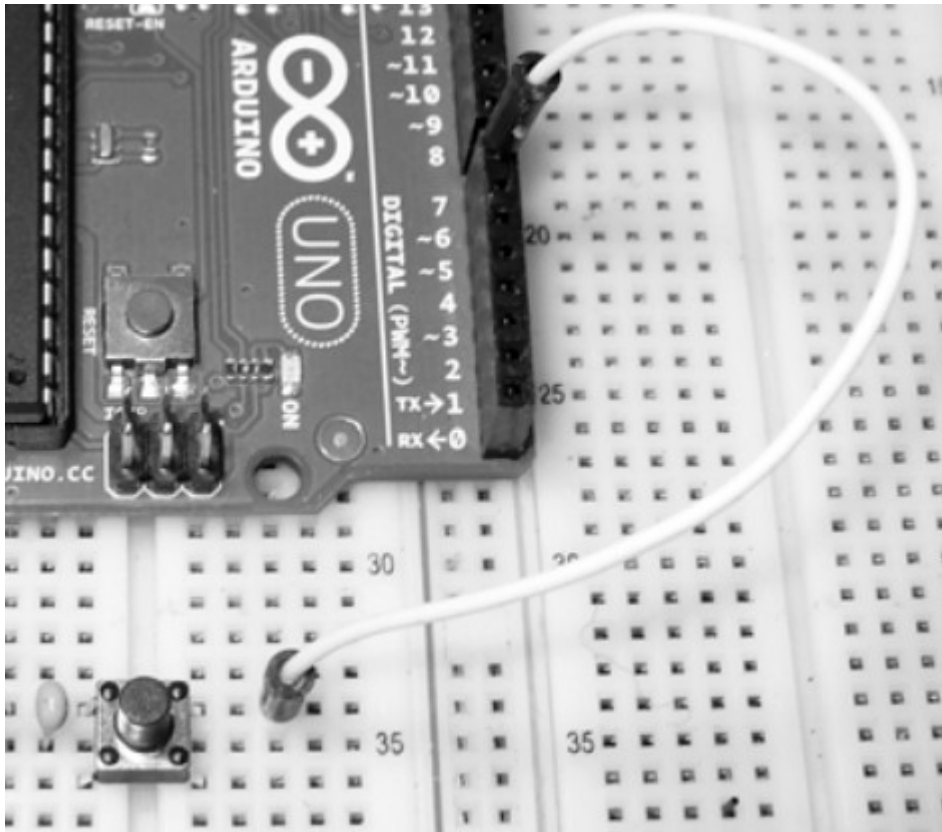


Przewody połączone z pinami 5 V (czerwony) i GND (czarny)

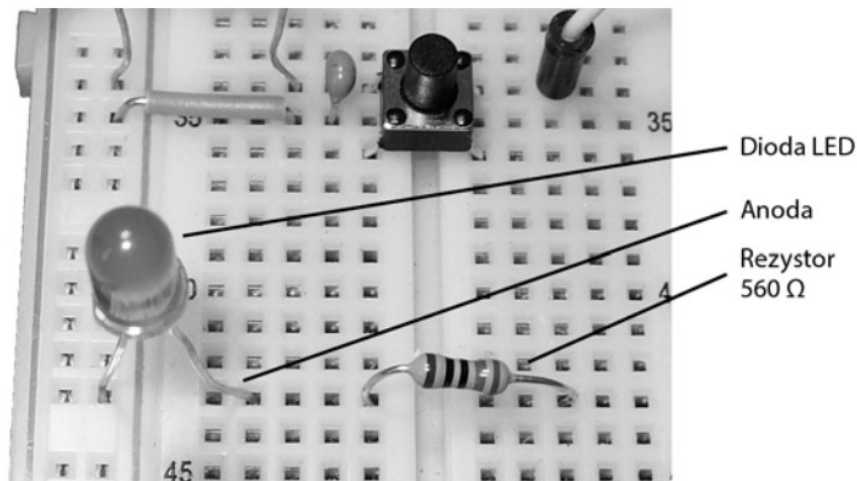


# Programowanie urządzeń Internetu rzeczy IoT

Połączenie przycisku z cyfrowym pinem wejściowym



Poniżej sposób instalacji diody LED i rezystora 560  $\Omega$



Przykładowy kod:

```
#define LED 12
#define BUTTON 7
void setup()
{
  pinMode(LED, OUTPUT); // pin wyjściowy dla diody LED
  pinMode(BUTTON, INPUT); // pin wejściowy dla przycisku
}
```

## Programowanie urządzeń Internetu rzeczy IoT

```
void loop()
{
  if ( digitalRead(BUTTON) == HIGH )
  {
    digitalWrite(LED, HIGH); // włącza diodę LED
    delay(1000); // czeka jedną sekundę
    digitalWrite(LED, LOW); // wyłącza diodę LED
  }
}
```

6. Zmodyfikuj powyższy projekt, aby lepiej dopasować długość czasu świecenia diody i reakcji na użycie przycisku.
7. Przygotuj model systemu do sterowania ruchem samochodowym. Należy zaplanować projekt ruchu wahadłowy na wąskim przejeździe (np. na moście).

Zadaniem jest instalacja sygnalizacji świetlnej (złożonej z trzech świateł różnych kolorów) na obu końcach przejazdu (mostu). Światła spowodują, że ruch w danej chwili będzie się odbywał tylko w jednym kierunku. W momencie wykrycia przez czujniki systemu samochodu oczekującego na wjazd po jednej stronie mostu (stojącego na czerwonym świetle) światło zmienia się, tak aby był możliwy przejazd w odpowiednim kierunku

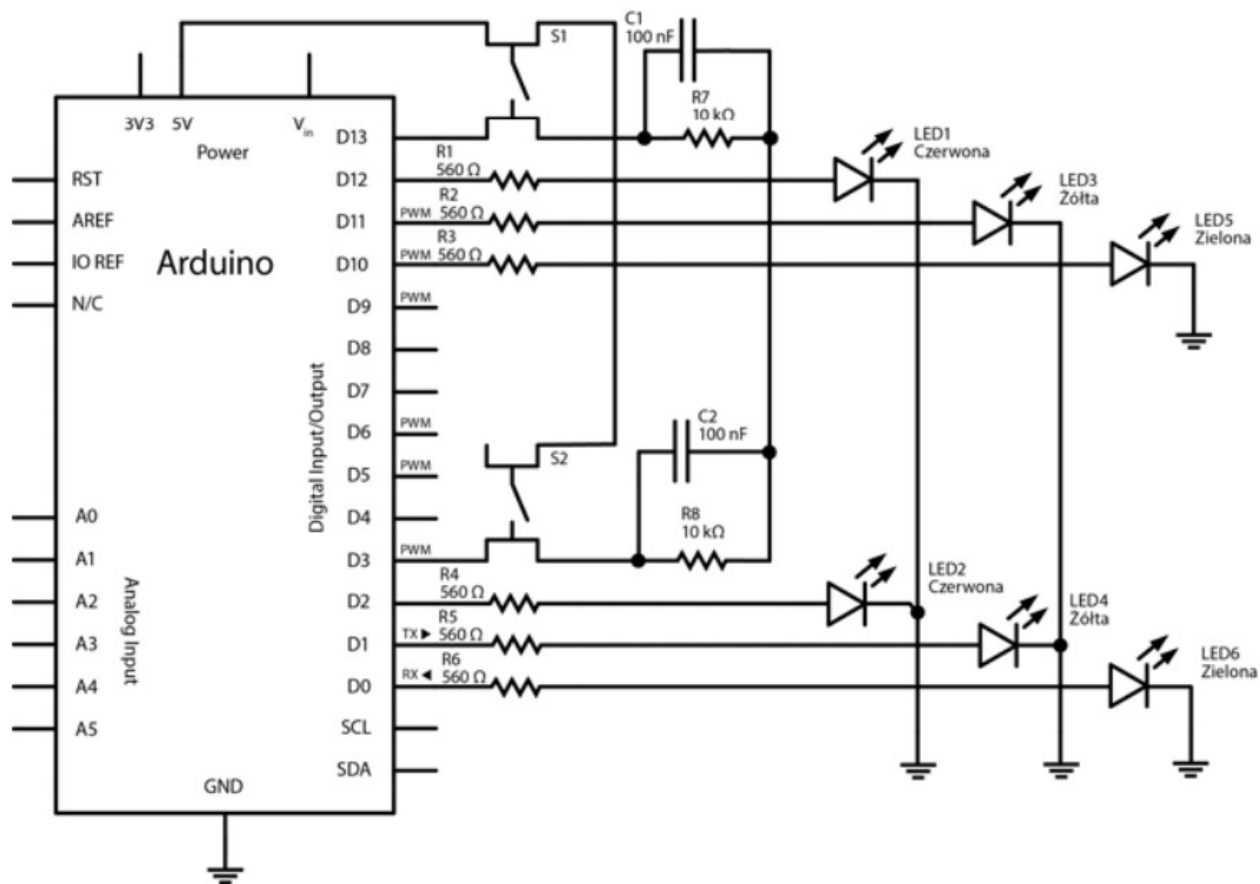
Algorytm Do symulowania czujników pojazdów po obu stronach mostu zostaną użyte dwa przyciski. Każdy sygnalizator świetlny będzie złożony z trzech diod LED: czerwonej, żółtej i zielonej. Początkowo system będzie dopuszczał ruch ze wschodu na zachód, zatem sygnalizator skierowany na wschód będzie pokazywał zielone światło, a sygnalizator skierowany na zachód będzie pokazywał czerwone światło. Jeśli samochód zbliża się do mostu (do symulowania tego zdarzenia zostanie użyty przycisk) od strony czerwonego światła, system zmieni światło po drugiej stronie mostu z zielonego na czerwone i odczeka pewien czas, aby umożliwić opuszczenie mostu ewentualnym pojazdom, które już się na nim znajdują. Po tym czasie system włączy żółte światło (od strony oczekujących pojazdów), aby dać kierowcom sygnał do przygotowania się przed ostatecznym włączeniem zielonego światła. Światło pozostanie zielone do czasu pojawienia się pojazdów po drugiej stronie. Opisany proces będzie powtarzany bez końca

**Sprzęt** Oto lista urządzeń potrzebnych do realizacji tego projektu:

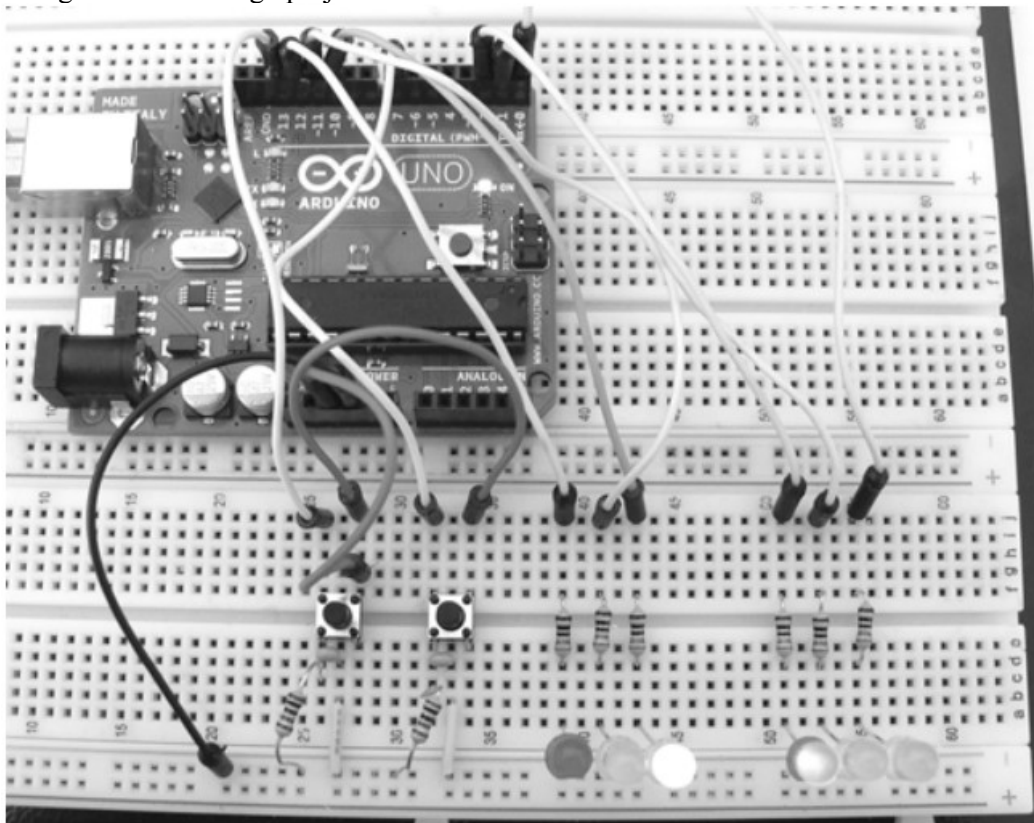
- dwie czerwone diody LED (LED1 i LED2);
- dwie żółte diody LED (LED3 i LED4);
- dwie zielone diody LED (LED5 i LED6);
- sześć rezystorów  $560\ \Omega$  (od R1 do R6);
- dwa rezystory  $10\ k\Omega$  (R7 i R8);
- dwa kondensatory  $100\ nF$  (C1 i C2);
- dwa przyciski (S1 i S2);
- płytki uniwersalna średniej wielkości;
- płytki Arduino i przewód USB;
- różne przewody.

Poniżej schemat dla projektu:

## Programowanie urządzeń Internetu rzeczy IoT



Widok gotowego zrealizowanego projektu:



## Programowanie urządzeń Internetu rzeczy IoT

8. Pokaż prowadzącemu zajęcia programy i wyniki ich działania lub prześlij wyniki (pliki projektu) do platformy elearningowej.

Część zagadnień na podst.:

*Monk S., Arduino dla początkujących – podstawy i szkice.*

*Boxall J. Arduino Workshop.*

*Karvinen K. Czujniki dla początkujących Poznaj otaczający Cię świat za pomocą elektroniki, Arduino i Raspberry.*

*Boxall J. Arduino 65 Praktycznych Projektów*

*Schwartz M. Arduino Home Automation Projects*

*[www.wikipedia.org](http://www.wikipedia.org),*