

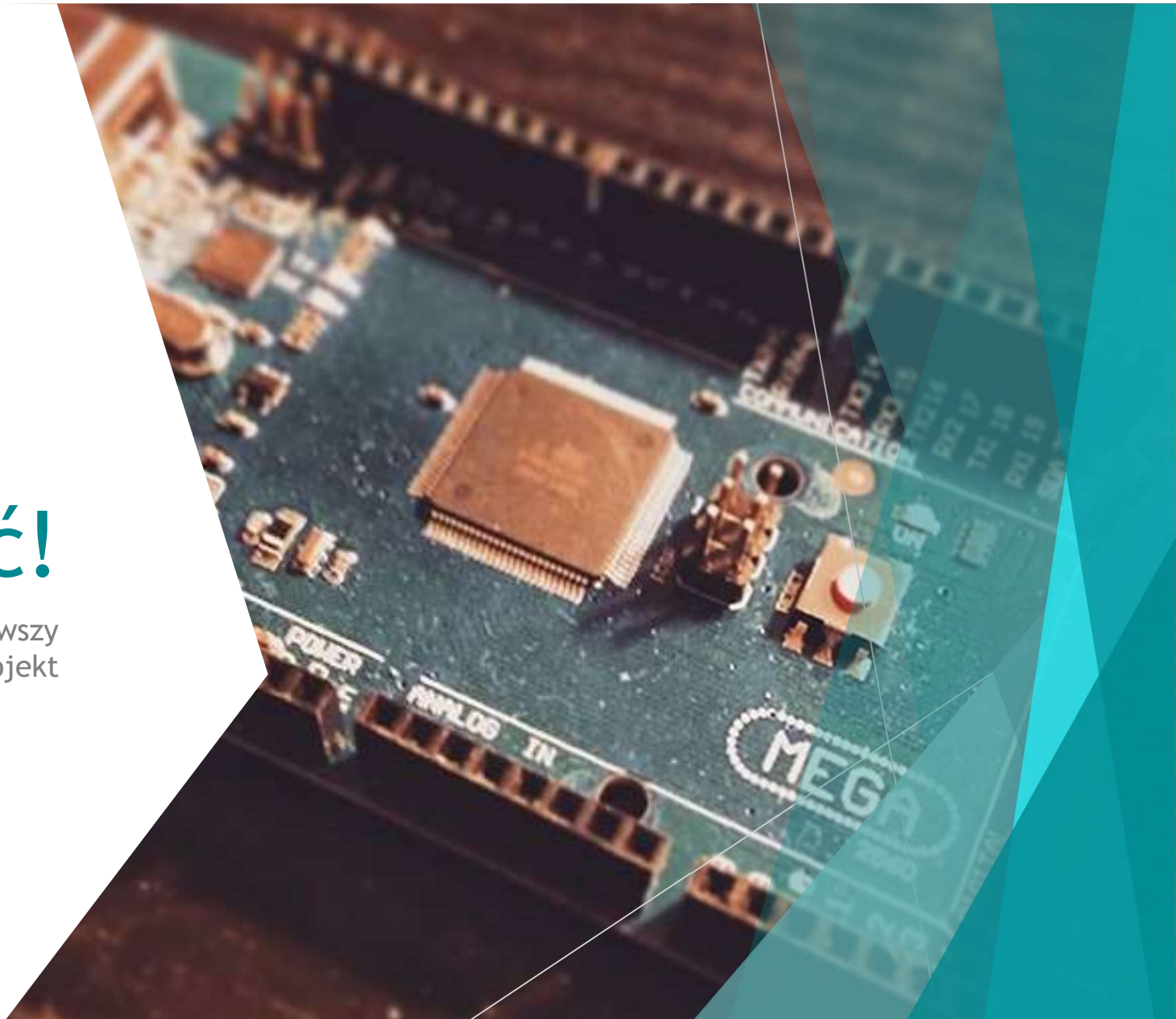
Warsztaty Arduino

Spotkanie #1



Chcę tworzyć!

Wprowadzenie do środowiska i pierwszy projekt



Pobranie i instalacja środowiska

- <https://www.arduino.cc/en/Main/Software>

Download the Arduino IDE



The screenshot shows the Arduino IDE download page. On the left is the Arduino logo (an infinity symbol with a minus and plus sign). To its right, the text reads: **ARDUINO 1.8.7**, followed by a description of the IDE and its compatibility with Windows, Mac OS X, and Linux. Below this is a link to the 'Getting Started' page. On the right side of the page, there is a red dashed box containing links for 'Windows Installer, for Windows XP and up', 'Windows ZIP file for non admin install', 'Windows app Requires Win 8.1 or 10' (with a 'Get' button), 'Mac OS X 10.8 Mountain Lion or newer', 'Linux 32 bits', 'Linux 64 bits', 'Linux ARM', 'Release Notes', 'Source Code', and 'Checksums (sha512)'. A red arrow points from the URL in the list above to the 'Windows ZIP file' link in the red dashed box.

ARDUINO 1.8.7

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions.

Windows Installer, for Windows XP and up
Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10
[Get](#)

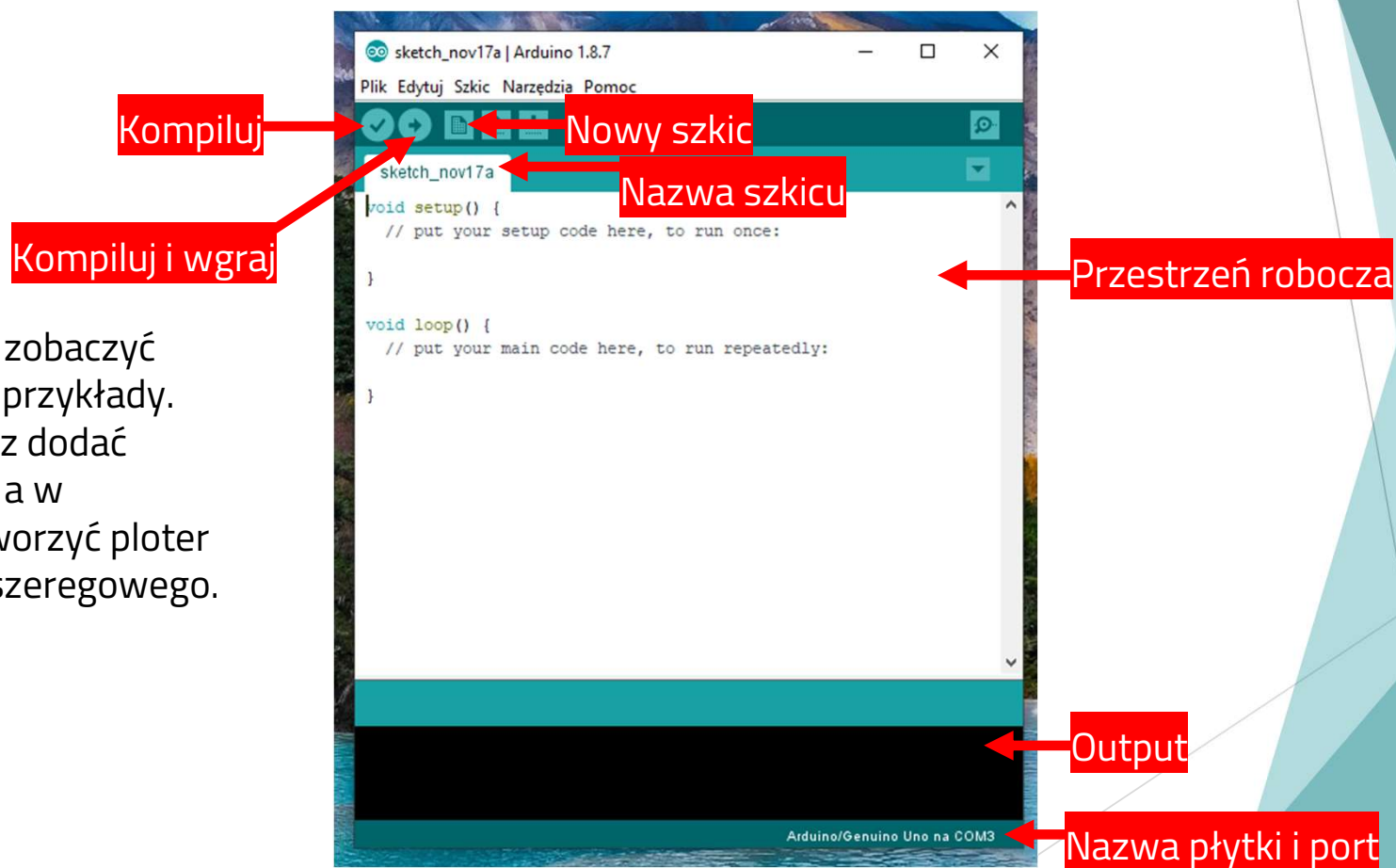
Mac OS X 10.8 Mountain Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM

[Release Notes](#)
[Source Code](#)
[Checksums \(sha512\)](#)



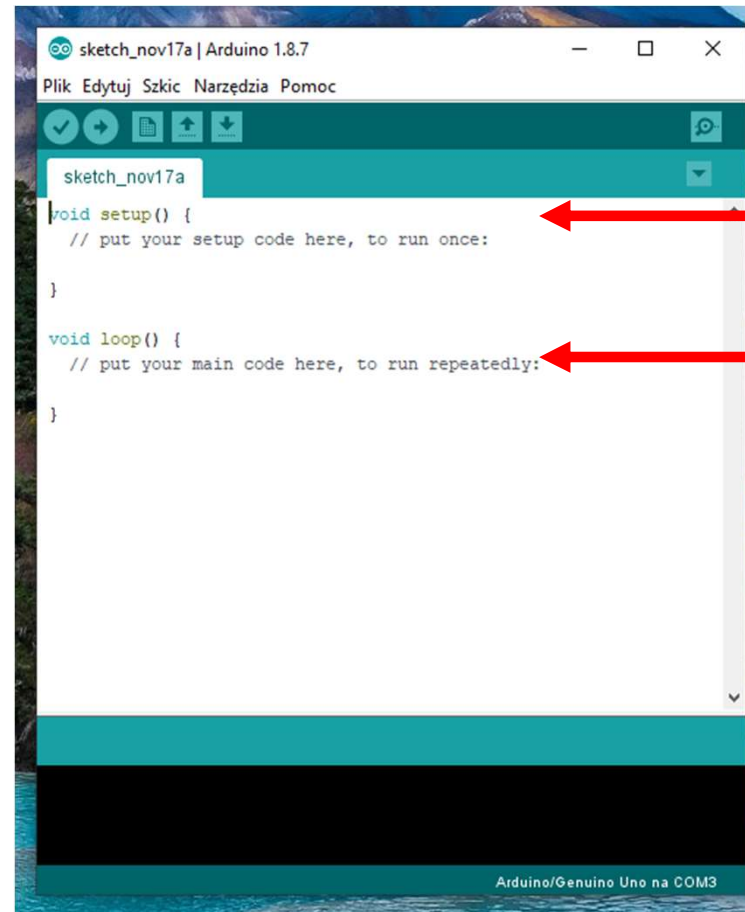
Poruszanie się po Arduino IDE



W **Pliku** możesz zobaczyć ostatnie szkice i przykłady. W **Szkicu** możesz dodać swoje biblioteki, a w **Narzędziach** otworzyć ploter i monitor portu szeregowego.

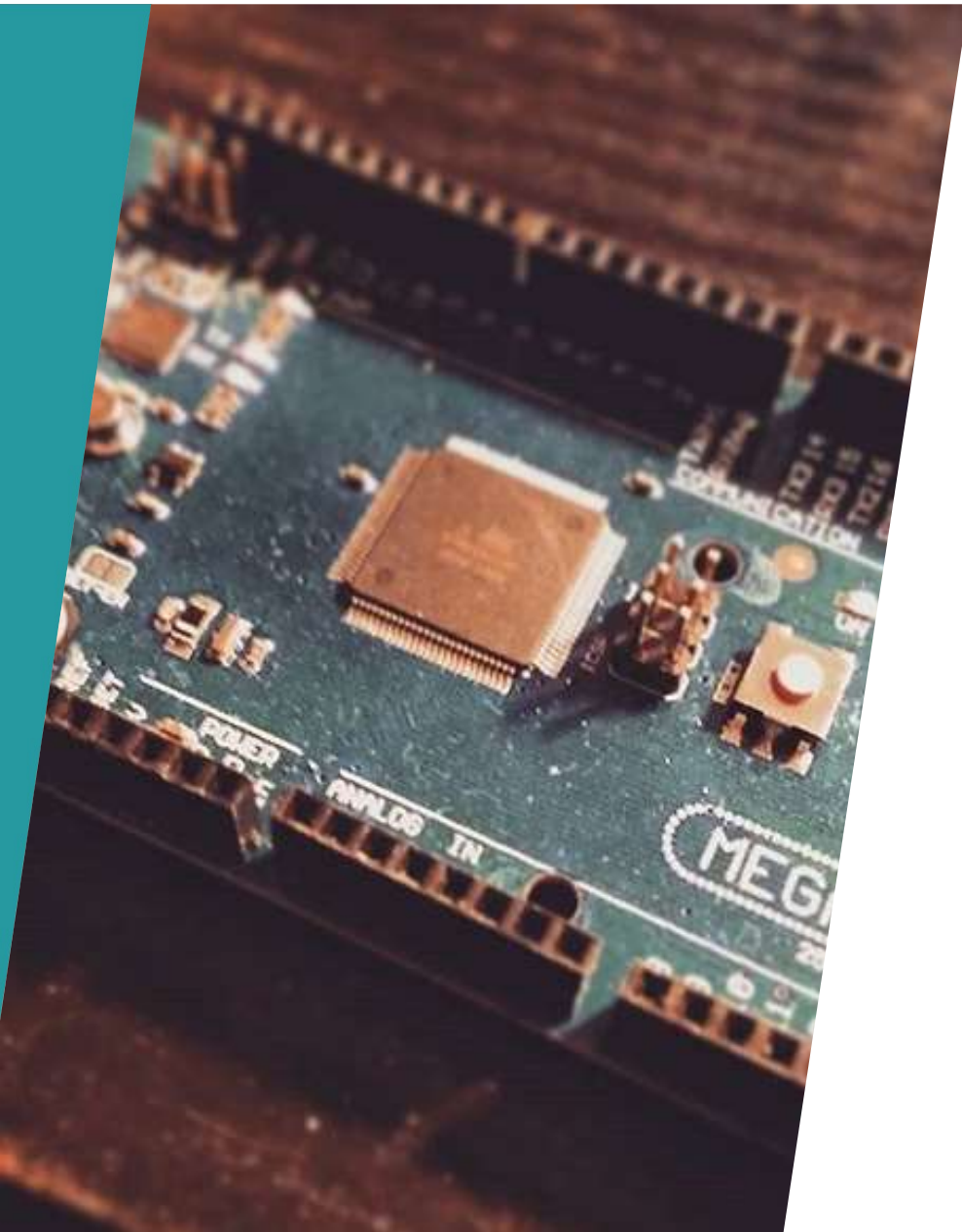
Programowanie Arduino

Arduino programujemy przy
użyciu języka **C++** i bibliotek
dostarczonych przez
producenta lub możemy je
dodać sami.



Wykonuje się raz

Wykonuje się cyklicznie



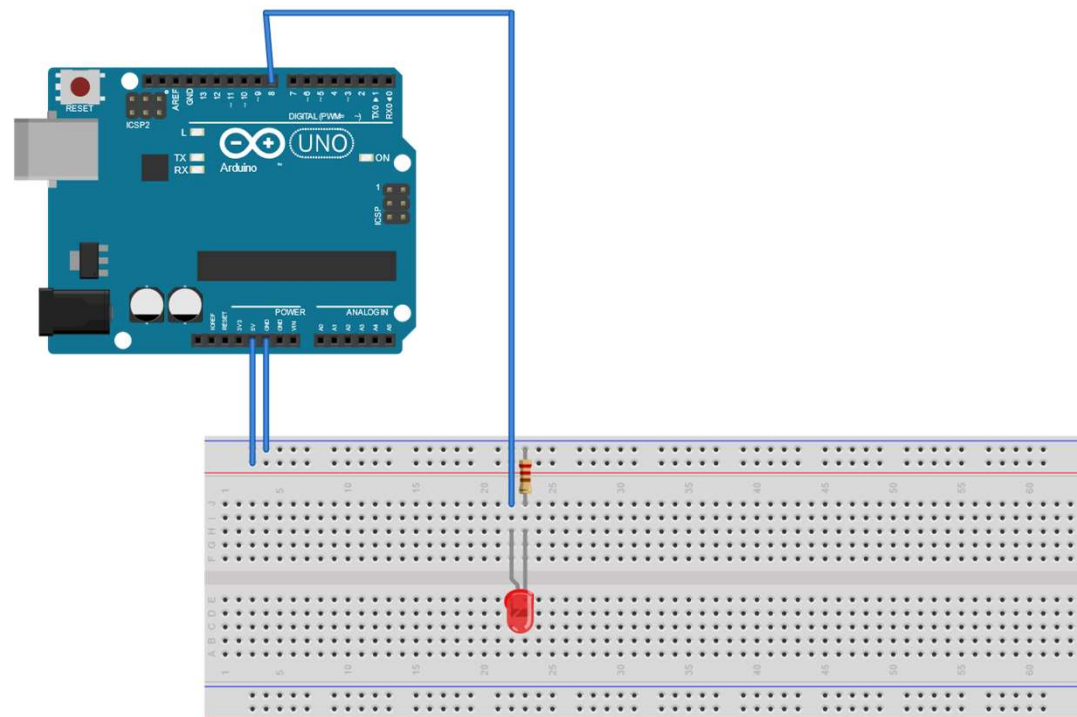
Budujemy!

Projekty na dziś

Dioda i Rezystor

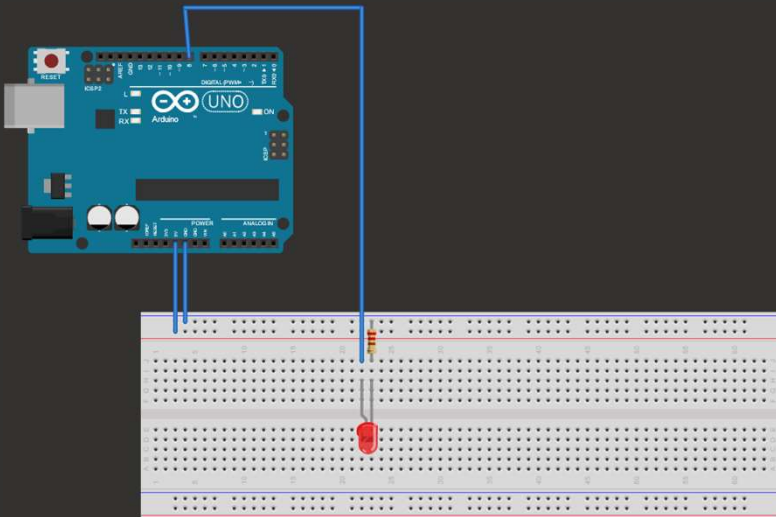
Projekt #1, Schemat

- ▶ Potrzebne części:
 - Arduino UNO R3
 - Diody LED
 - Rezystory 220 Ohm
 - Przewody



Dioda i Rezystor

Projekt #1, Kod



```
#define LED_PIN 8
void setup() {
    pinMode(LED_PIN, OUTPUT);
}

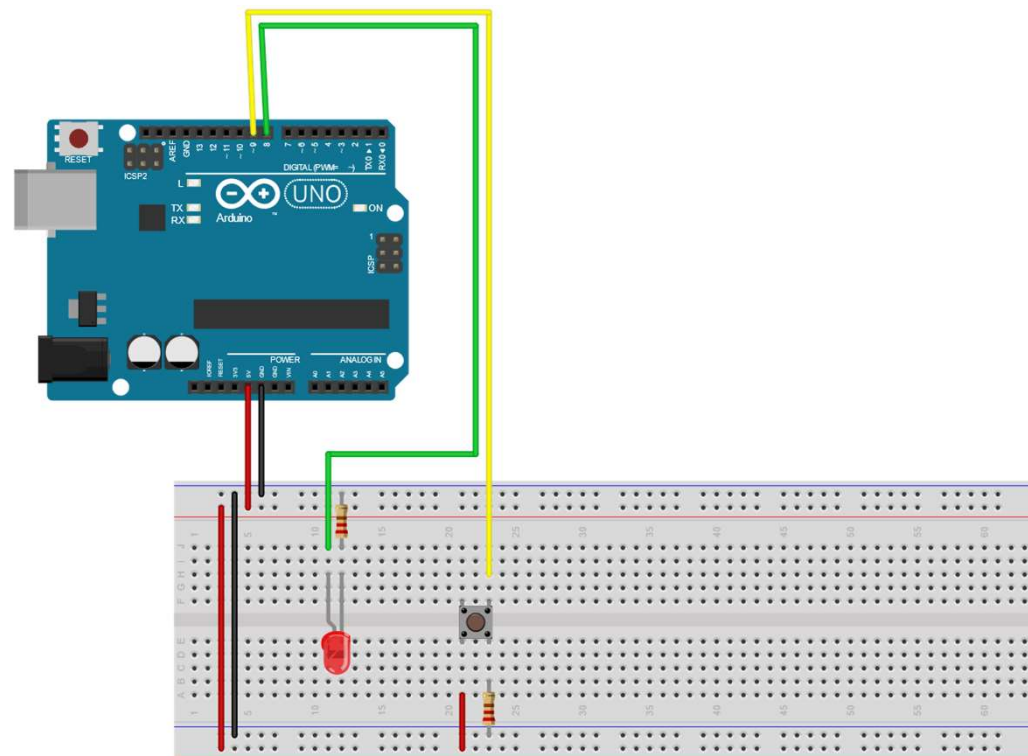
void loop() {
    digitalWrite(LED_PIN, HIGH);
    delay(1000);
    digitalWrite(LED_PIN, LOW);
    delay(1000);
}
```


Dioda i przycisk

Projekt #2, Schemat

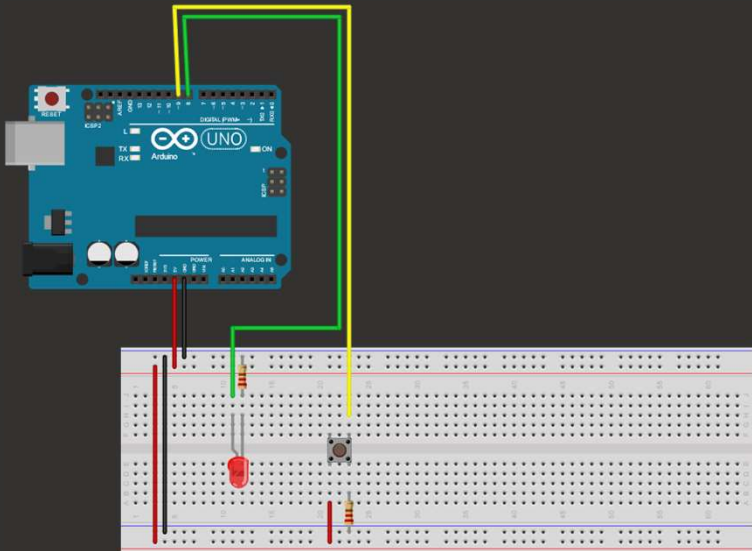
► Potrzebne części:

- Arduino UNO R3
- Diody LED
- Rezystory 220 Ohm
- Switch
- Przewody



Dioda i przycisk

Projekt #2, Kod



```
#define LED_PIN 8
#define BUTTON_PIN 9
```

```
void setup() {
    pinMode(LED_PIN, OUTPUT);
    pinMode(BUTTON_PIN, INPUT);
}
```

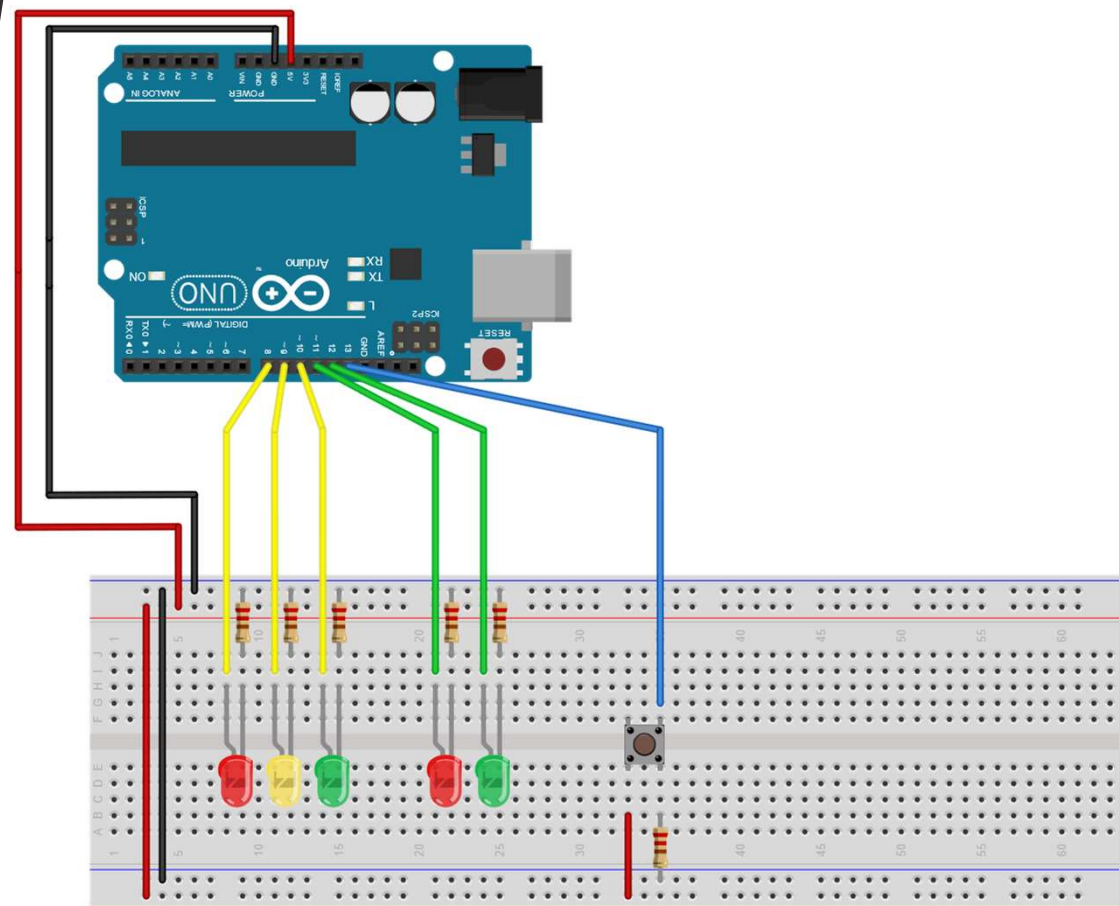
```
void loop() {
    if(digitalRead(BUTTON_PIN) == LOW){
        digitalWrite(LED_PIN, LOW);
    }
    else{
        digitalWrite(LED_PIN, HIGH);}
}
```

Światła drogowe

Projekt #3, Schemat

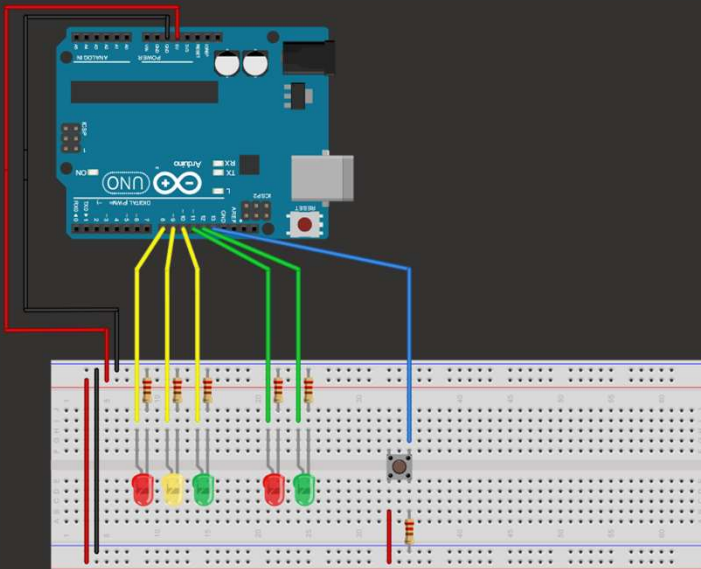
► Potrzebne części:

- Arduino UNO R3
- Diody LED
- Rezystory 220 Ohm
- Switch
- Przewody



Światła drogowe

Projekt #3, Kod - Część pierwsza



github.com/filesmuggler/octopus

```
#define RED_CAR 8
#define YELLOW_CAR 9
#define GREEN_CAR 10
#define RED_PED 11
#define GREEN_PED 12
#define BUTTON_PED 13

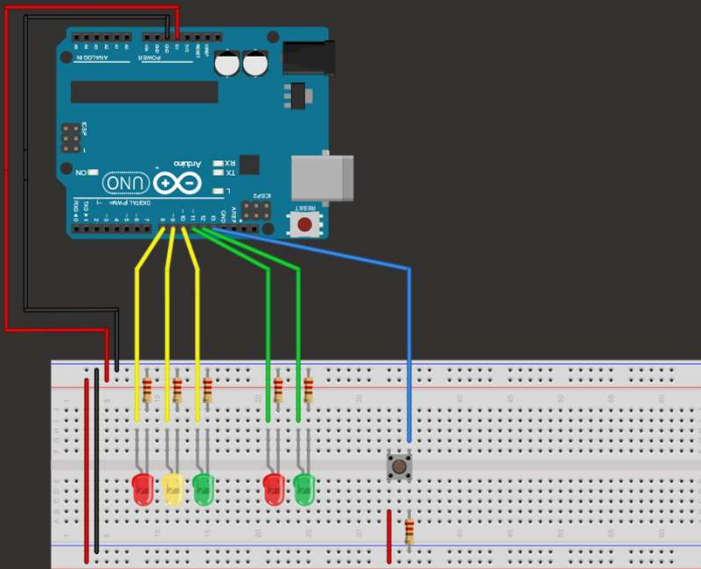
void trafficLights(){
    delay(2000);
    digitalWrite(GREEN_CAR, LOW);
    delay(700);
    digitalWrite(YELLOW_CAR, HIGH);
    delay(700);
    digitalWrite(YELLOW_CAR, LOW);
    digitalWrite(RED_CAR, HIGH);
    digitalWrite(RED_PED, LOW);
    digitalWrite(GREEN_PED, HIGH);

    delay(5000);

    digitalWrite(RED_PED, HIGH);
    digitalWrite(GREEN_PED, LOW);
    digitalWrite(YELLOW_CAR, HIGH);
    delay(500);
    digitalWrite(RED_CAR, LOW);
    digitalWrite(YELLOW_CAR, LOW);
    digitalWrite(GREEN_CAR, HIGH);
}
```

Światła drogowe

Projekt #3, Kod - Część druga



```
void setup(){
  pinMode(RED_CAR, OUTPUT);
  pinMode(YELLOW_CAR, OUTPUT);
  pinMode(GREEN_CAR, OUTPUT);
  pinMode(RED_PED, OUTPUT);
  pinMode(GREEN_PED, OUTPUT);
  pinMode(BUTTON_PED, INPUT);
  digitalWrite(GREEN_CAR, HIGH);
  digitalWrite(RED_PED, HIGH);
}

void loop(){
  if(digitalRead(BUTTON_PED)==HIGH)
  {
    trafficLights();
  }
}
```

Warsztaty Arduino



github.com/filesmuggler/octopus



@kncybair



@CybAIR