

- Teraz podłącz płytkę *RobotDyn* WiFi do komputera i wybierz: *Tools* → *Board* → *ESP8266 Boards* → *LOLIN (WEMOS) D1 R2 & mini*.
 - Teraz wybierz *Tools* → *Port*.
 - Następnie należy zainstalować bibliotekę do chmury Adafruit IO. W tym celu wybierz: *Tools* → *Manage Libraries...* i wyszukaj bibliotekę *Adafruit MQTT*.
- (b) Wejdź na stronę: <https://io.adafruit.com> i załóż darmowe konto. Następnie na stronie utwórz *feed* o wybranej nazwie np. *fotorezystor*. W tym celu wybierz zakładkę *Feeds* a następnie *New Feed*.
 - (c) Utwórz pulpit do wizualizacji danych. W tym celu wybierz zakładkę *Dashboards* a potem *New dashboard*. Po jego nazwaniu, wybierz ikonę ustawień w prawym górnym rogu i wybierz opcję *Create New Block*. Pojawi się lista dostępnych elementów, wybierz wykres. Następnie powiąż wybrany wykres z utworzonym *feed*.
 - Powróć do Arduino IDE i spróbuj uzupełnić poniższy kod by wysyłał dane do chmury:

```
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"

// Dane do Wi-Fi
#define WIFI_SSID "Nazwa wifi"
#define WIFI_PASS "Haslo"

// Dane do Adafruit IO
#define AIO_SERVER "io.adafruit.com"
#define AIO_PORT 1883
#define AIO_USERNAME "twoj user name"
#define AIO_KEY "twoj klucz"

// Klient Wi-Fi i MQTT
WiFiClient client;
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_PORT,
    AIO_USERNAME, AIO_KEY);

void connectMQTT()
{
    Serial.print("Laczenie z Adafruit IO...");
    while (mqtt.connect() != 0)
    {
        Serial.println("Nie udalo sie polaczyc.
            Ponowna proba za 5 sek...");
        delay(5000);
    }
    Serial.println("Polaczono z Adafruit IO!");
}
```

```

// Feed do wysyłania rezystancji
Adafruit_MQTT_Publish fotoFeed = Adafruit_MQTT_Publish(&mqtt,
  AIO_USERNAME "/feeds/twoja_nazwa_feed");

void setup()
{
  Serial.begin(115200);
  WiFi.begin(WIFI_SSID, WIFI_PASS);

  Serial.print("Łaczenie z Wi-Fi...");
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nPołączono!");

  // Połączenie z Adafruit IO
  connectMQTT();
}

void loop()
{
  if (!mqtt.connected())
  {
    connectMQTT();
  }
  mqtt.processPackets(10000);

  // Odczyt rezystancji
  int fotorezystor = 1;

  // Wysłanie do Adafruit IO
  if(!fotoFeed.publish(fotorezystor)) Serial.println("
    Błąd wysyłania!");
  else Serial.println("Wysłano poprawnie!");

  delay(5000); // Wysyłaj co 5 sekund
}

```

2 Zamek na breloczek RFID

Stwórz program, który umożliwi otwarcie zamka korzystając z breloczka RFID. W tym celu:

- (a) Podłącz moduł RC-522 do Arduino Mega 2560 zgodnie z tabelą nr 1.

RC-522	Arduino Mega 2560
VCC	3.3 V
RST	22
GND	GND
MISO	50
MOSI	51
SCK	52
SS (NSS)	53

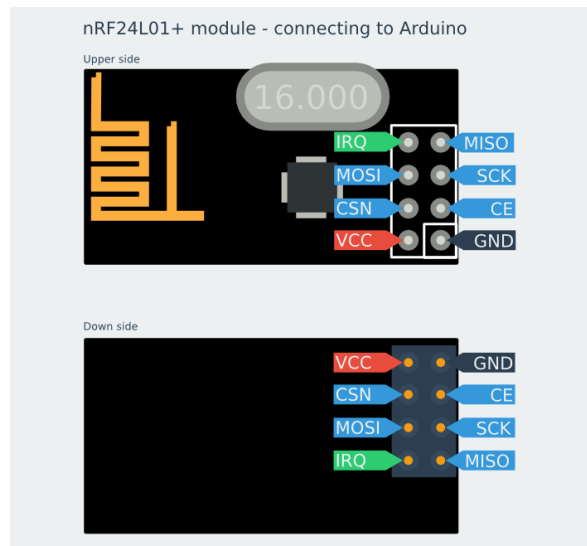
Tabela 1: Podłączenie modułu RC-522 do Arduino Mega 2560.

- (b) Zainstaluj bibliotekę *MFRC522*. Otwórz przykład *readNUID* i przeanalizuj działanie programu.
- (c) Stwórz program, który odczyta ID breloczka RFID.
- (d) Zmodyfikuj program w taki sposób aby po rozpoznaniu breloczka RFID, dioda RGB zapaliła się na zielono a po upływie 0.5s powróciła do koloru czerwonego.
- (e) Zmodyfikuj program w taki sposób aby po rozpoznaniu breloczka RFID, serwo-mechanizm zmieniał położenie z 0° na 90° . A następnie po upływie 0.5s powracał do położenia 0° .

3 Czujnik temperatury

Zadanie składa się z dwóch części. W ramach pierwszej należy zbudować układ, który będzie odbierał zmierzoną temperaturę za pomocą modułu radiowego mod nRF24L01. W tym celu:

- (a) Podłącz moduł radiowy według poniższych wskazówek:
- VCC = 3.3V (**UWAGA NIE wolno podłączyć do 5V**)
 - CSN = 53
 - MOSI = 51
 - GND = GND
 - CE = 48
 - SCK = 52
 - MISO = 50



- (b) Zainstaluj bibliotekę RF24. Następnie przeanalizuj poniższy kod i spróbuj odebrać wartość temperatury od prowadzącego.

```

#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <printf.h>
#include <Wire.h>

RF24 radio(48, 53);
const byte rxAddr[6] = "00009";

float temp;

void setup()
{
    Serial.begin(115200);
    while (!Serial)
    {
        // some boards need to wait to ensure access to
        // serial over USB
    }
    printf_begin();
    radio.begin();
    radio.setPALevel(RF24_PA_MIN);
    radio.openReadingPipe(1, rxAddr);

    radio.startListening();
    radio.printDetails();
}

void loop()

```

```

{
    if (radio.available())
    {
        radio.read(&temp, sizeof(float));
    }
}

```

(c) Wyświetl temperaturę w monitorze portu szeregowego.

W ramach drugiej części układ ma mierzyć temperaturę za pomocą czujnika DS18B20 bądź LM35 (do wyboru) i wysyłać ją do prowadzącego za pomocą modułu radiowego. W tym celu

(d) Podłącz cyfrowy termometr DS18B20 bądź LM35 (patrz laboratorium 3) oraz odczytaj temperaturę na monitorze portu szeregowego.

(e) Napisz program, który będzie wysyłał temperaturę za pomocą modułu radiowego. Adresy dla modułów radiowych: 0000 + numer komputera np. dla komputera z numerem 1: 00001. W tym celu:

- ustaw wzmocnienie: RF24_PA_MAX,
- ustaw aby próbował 15 razy nawiązać połączenie co 15 ms (funkcja setRetries(ile razy,co ile)),
- otwórz łącze do nadawania (funkcja openWritingPipe(adres)),
- zakończ nasłuch i przejdź w tryb transmisji (funkcja stopListening()),
- wyślij dane (funkcja write(adres zmiennej np. &temp, rozmiar zmiennej np. sizeof(float)))

4 Dioda RGB (dodatkowe)

Stwórz program, który umożliwi sterowanie diodą RGB za pomocą smartfona korzystając z modułu Bluetooth. W tym celu:

(a) Zainstaluj bibliotekę RemoteXY

(b) Wejdź na stronę <http://remotexy.com/en/editor/> i skonfiguruj w sekcji "Properties" (po prawej stronie):

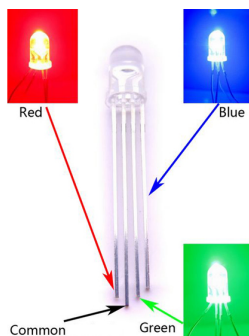
- Rodzaj płytki: Arduino Mega
- Moduł Bluetooth HC-05/06
- Connection interface: Hardware Serial
- Serial Port: TX1 (pin 18) i RX1 (pin 19)

Stwórz GUI użytkownika (3 suwaki dla każdego z kolorów osobno). Następnie pobierz kod ze strony (Get Source Code) i wklej go do Arduino IDE.

(c) Podłącz moduł bluetooth do arduino zgodnie z opisem:

- RX - TX1 (pin 18)
- TX - RX1 (pin 19)
- VCC - 5V
- GND - GND

(d) Podłącz diodę RGB do pinów PWM zgodnie z poniższym rysunkiem. Nasza dioda ma wspólną katodę zatem do nóżki oznaczonej jako "common" należy podłączyć GND. **Pamiętaj o rezystorach!**



(e) Ściągnij na swojego smartfona aplikację **RemoteXY**.

(f) Otrzymane wartości z suwaków dla poszczególnych kolorów prześlij do diody RGB ustawiając odpowiednio wybrane wypełnienia (funkcja `analogWrite(numer pinu, wypełnienie od 0 do 255)`). Jak to zrobić znajdziesz na stronie RemoteXY w sekcji "How it works". Przetestuj aplikację, czy działa poprawnie. Pamiętaj aby sparować telefon z modułem Bluetooth. Hasło do parowania to 1234.

5 Wykaz elementów

Niezbędne elementy do wykonania ćwiczenia

1. Moduł radiowy nRF24L01
2. Czujnik temperatury DS18B20
3. Czujnik temperatury LM35
4. Rezystor $4.7k\Omega$
5. Moduł bluetooth
6. Dioda RGB
7. 3x rezystory 330Ω
8. Diody

9. RFID-RC522

10. Serwomechanizm