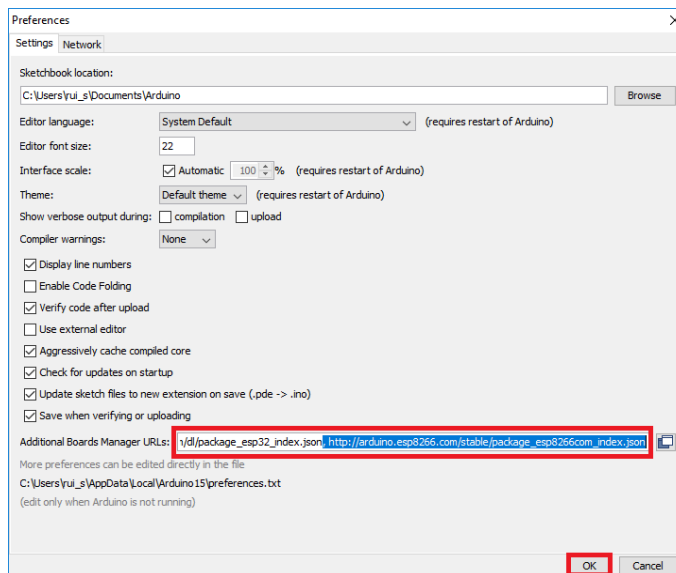


Zad.1. Konfiguracja środowiska. Dodanie płytki *ESP8266 NodeMCU v2* do środowiska *Arduino IDE*.

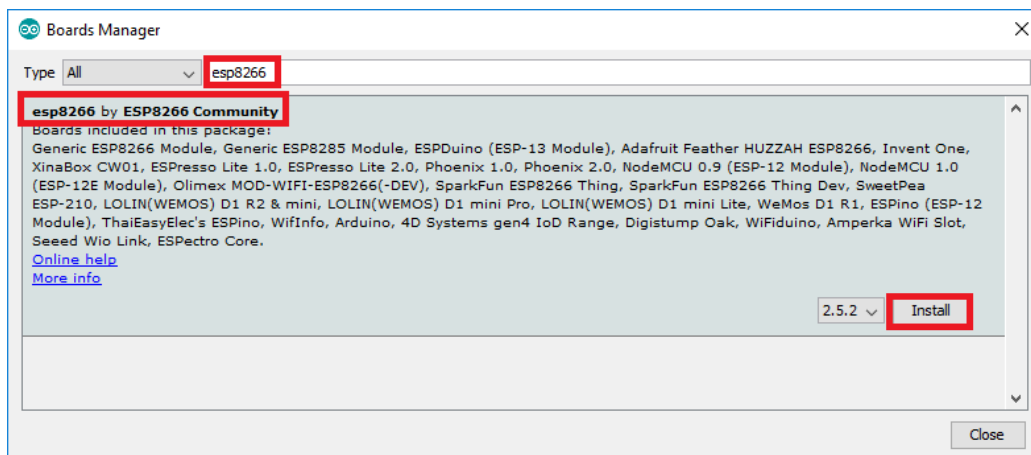
Po uruchomieniu środowiska Arduino IDE należy, z menu *File* wybrać *Preferences*. W sekcji *Additional Boards Manager URLs* dodajemy link:

```
arduino.esp8266.com/stable/package_esp8266com_index.json
```

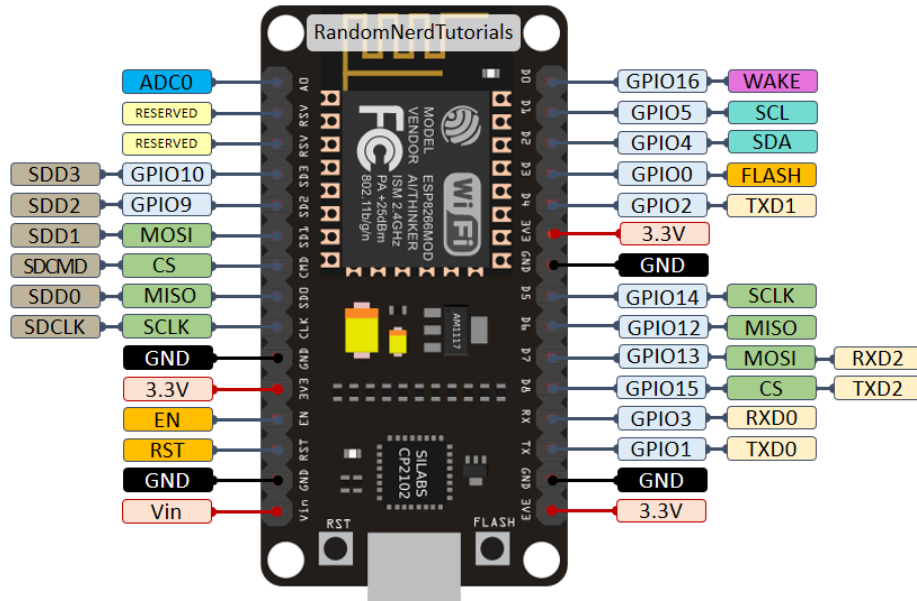


Wskazany link można też skopiować ze strony github.com/esp8266/Arduino. Po wprowadzeniu należy nacisnąć przycisk **OK**.

Kolejnym krokiem jest dodanie płytki NodeMCU. Z menu *Tools* (Narzędzia) wybieramy *Boards* (Płytki), a następnie *Board Manager...* (Menadżer płytek...). W polu wyszukiwania wpisujemy **esp8266**. Instalujemy najnowszą wersję pakietu - przycisk **Install**.



Ostatnim krokiem jest wybranie odpowiedniej płytki. Z menu *Tools (Narzędzia)* wybieramy *Boards (Płytki)*, a następnie **NodeMCU v1.0** (ESP-12E Module). Na poniższym rysunku przedstawiono opis wyprowadzeń wraz z funkcjami dodatkowymi wybranego modułu NodeMCUv2.



Moduł podłączamy do portu USB komputera. Z menu *Tools (Narzędzia)* należy wybrać, za pomocą opcji *Port*, odpowiedni port komunikacyjny (COM). Za pomocą przycisku *Upload* zaprogramować płytkę.



Przycisk Upload

Komunikat *Done uploading* oznacza, że programowanie płytki odbyło się poprawnie i można przystąpić do realizacji kolejnych zadań.

Zad.2. Przetestować działanie poniższego programu do połączenia z siecią bezprzewodową.

```
#include<ESP8266WiFi.h>
#define BAUDRATE 115200

const char* ssid = "Laboratorium-IoT";
const char* pass = "";

void setup() {
    Serial.begin(BAUDRATE);
    WiFi.begin(ssid,pass);
    Serial.print("Connecting .");
    while(WiFi.status() != WL_CONNECTED) {
        Serial.print(".");
        delay(250);
    }
    Serial.print("\nIP: ");
    Serial.println(WiFi.localIP());
}

void loop() {
}
```

Zad.3. Przetestować działanie poniższego programu.

```
#include<ESP8266WiFi.h>
#define BAUDRATE 115200

const char* ssid = "Laboratorium-IoT";
const char* pass = "";

#define CONN_DELAY 500
long p_millis = 0;

void setup() {
    Serial.begin(BAUDRATE);
    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid,pass);
    Serial.print("Connecting to "); Serial.print(ssid);
    Serial.print("with password "); Serial.println(pass);
```

```
Serial.print("Start connection .");
while((WiFi.status() != WL_CONNECTED) &&
      (millis() - p_millis > CONN_DELAY)) {
  Serial.print(".");
  switch(WiFi.status()) {
    case WL_IDLE_STATUS:
      Serial.println("Changing between statuses");
      break;

    case WL_NO_SSID_AVAIL:
      Serial.println("Configured SSID cannot be reached");
      break;

    case WL_CONNECT_FAILED:
      Serial.println("Connection failed");
      break;

    case WL_CONNECT_WRONG_PASSWORD:
      Serial.println("Password is incorrect");
      break;

    case WL_DISCONNECTED:
      Serial.println("Module is not configured in STA mode");
      break;

    default:
      Serial.println("Unknown WiFi status message");
      break;
  }
  p_millis = millis();
}

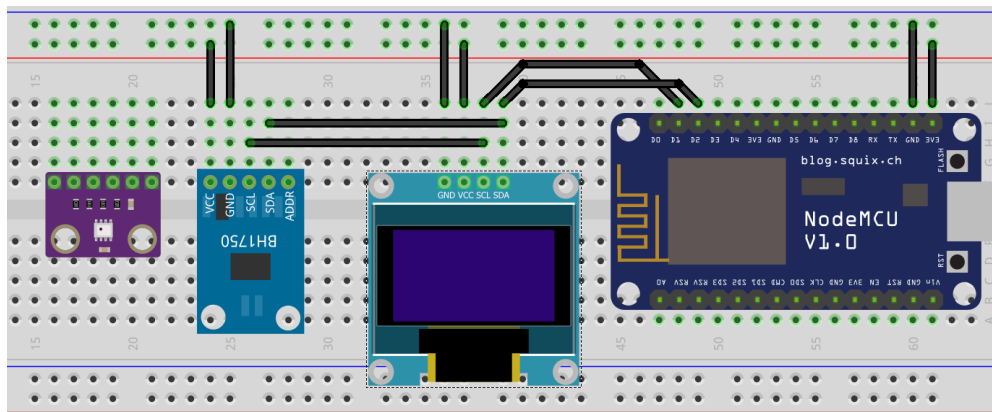
Serial.print("ESP8266 board IP ");
Serial.print(WiFi.localIP());
WiFi.setAutoReconnect(true);    // automatically reconnect to
WiFi.persistent(true);        // the previously connected AP

Serial.setDebugOutput(true);
WiFi.printDiag(Serial);

Serial.printf("Connection status: %d\n", WiFi.status());
Serial.print("RRSI: ");
Serial.println(WiFi.RSSI());
```

```
}  
  
void loop() {  
}
```

Zad.4. Połączyć układ jak na rysunku. Przetestować działanie biblioteki do obsługi wyświetlacza OLED - U8g2lib.



```
#include<ESP8266WiFi.h>  
#include<Wire.h>  
#include<U8g2lib.h>  
#define BAUDRATE 115200  
  
const char* ssid = "Laboratorium-IoT";  
const char* pass = "";  
  
U8G2_SSD1306_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0,  
                                           U8X8_PIN_NONE);  
  
#define ROW_1 8  
#define ROW_2 18  
#define COL_START 0  
  
void setup() {  
  Serial.begin(BAUDRATE);  
  WiFi.begin(ssid,pass);  
  Serial.print("Connecting to "); Serial.print(ssid);  
  Serial.print("with password "); Serial.println(pass);  
  Serial.print("Start connection .");  
}
```

```
while(WiFi.status()!=WL_CONNECTED) {
    delay(500);
    Serial.print("."); }
Serial.print("ESP8266          board          IP          ");
Serial.print(WiFi.localIP());

Wire.begin();
u8g2.begin();
u8g2.clearBuffer();
u8g2.setFont(u8g2_font_ncenB08_tr); // choose a suitable font
u8g2.setCursor(COL_START, ROW_1);
u8g2.print("IP "); u8g2.print(WiFi.localIP());
u8g2.sendBuffer();
}

void loop() {
    u8g2.setCursor(COL_START, ROW_2);
    String message = "Test OLED";
    u8g2.print(message);
    u8g2.sendBuffer();}
```

Zaprezentować na wyświetlaczu OLED wartość natężenia oświetlenia, mierzoną za pomocą modułu BH1750. Do realizacji zadania można wykorzystać bibliotekę BH1750.

Zad.5. Zmiana MAC adres.

```
#include<ESP8266WiFi.h>
#define BAUDRATE 115200

uint8_t newMACAddress[] = {0x11, 0x22, 0x33, 0x44, 0x55, 0x66};

void setup() {
    Serial.begin(BAUDRATE);
    Serial.print("Current ESP Board MAC Address: ");
    Serial.println(WiFi.macAddress());

    wifi_set_macaddr(STATION_IF, &newMACAddress[0]);
    Serial.print("New ESP8266 Board MAC Address: ");
    Serial.println(WiFi.macAddress());
}
```

```
void loop() {  
}
```

Zad.6. Statyczny adres IP.

```
#include<ESP8266WiFi.h>  
#define BAUDRATE 115200  
  
IPAddress local_IP(192, 168, 1, 184);  
IPAddress gateway(192, 168, 1, 1);  
IPAddress subnet(255, 255, 0, 0);  
IPAddress primaryDNS(8, 8, 8, 8); //optional  
IPAddress secondaryDNS(8, 8, 4, 4); //optional  
  
const char* ssid = "Laboratorium-IoT";  
const char* pass = "";  
  
void setup() {  
    if (!WiFi.config(local_IP, gateway, subnet, primaryDNS,  
secondaryDNS)) {  
        Serial.println("STA Failed to configure");  
    }  
  
    Serial.print("Connecting to ");  
    Serial.println(ssid);  
    WiFi.begin(ssid, pass);  
    while (WiFi.status() != WL_CONNECTED) {  
        delay(250);  
        Serial.print(".");  
    }  
    Serial.print("\nIP address: ");  
    Serial.println(WiFi.localIP());  
}  
  
void loop() {  
}
```

Dla zainteresowanych:

1. Strona Arduino core for ESP8266 WiFi chip:
www.esp8266.com
2. ESP8266 Community Forum:
github.com/esp8266/Arduino
3. Serwis Last Minute Engineers:
lastminuteengineers.com/getting-started-with-esp8266
4. Serwis Random Nerd Tutorials:
randomnerdtutorials.com/projects-esp8266/