

1 Kod źródłowy

Lisint kodu z etapu 1

```
#include <Wire.h>
#include <SFE_BMP180.h>
#include "DHT.h"
#include <BH1750.h>

#define ALTITUDE 85.0
#define DHTPIN 7
#define DHTTYPE DHT11 // DHT 11

SFE_BMP180 pressure;
DHT dht(DHTPIN, DHTTYPE);
BH1750 lightMeter;

void setup() {
    Serial.begin(9600);
    Serial.println(F("Test polaczenia z DHT rozpoczety"));
    Serial.println(F("Test polaczenia z BH1750 rozpoczety"));
    Serial.println(F("Test polaczenia z BMP180 rozpoczety"));

    Wire.begin();
    lightMeter.begin();
    dht.begin();
    pressure.begin();
}

void loop() {
    char status;
    double T,P,p0;

    status = pressure.startTemperature();
    if (status != 0)
    {
        delay(status);
        status = pressure.getTemperature(T);
        if (status != 0)
        {
            //Wypisz pomiary
            Serial.print("Temperatura: ");
            Serial.print(T,2);
            Serial.print(" deg C || ");

            float h = dht.readHumidity();
            // sprawdzenie czy wczytanie danych przebieglo pomyslnie
            if (isnan(h)) {
                Serial.println(F("Nie udalo sie odczytac danych z DHT11!"));
                return;
            }
            Serial.print(F("Wilgotnosc powietrza: "));
```

```

        Serial.println(h);

    status = pressure.startPressure(3);
    if (status != 0)
    {
        delay(status);
        status = pressure.getPressure(P,T);
        if (status != 0)
        {
            // Wypisz pomiary:
            Serial.print("Cisnienie: ");
            Serial.print(P,2);
            Serial.print(" hPa ");

            p0 = pressure.sealevel(P,ALTITUDE);
            Serial.print(" || Cisnienie ndmp: ");
            Serial.print(p0,2);
            Serial.println(" hPa");

            //Wczytywanie natezenia
            float lux = lightMeter.readLightLevel();
            if (isnan(lux)) {
                Serial.println(F("Nie odczytano danych z BH1750!"));
                return;
            }
            Serial.print("Natezenie swiatla: ");
            Serial.print(lux);
            Serial.println(" lx");
            Serial.println();
            Serial.println();
        }
        else Serial.println("Nie udalo sie odczytac danych z BMP180!\n");
    }
    else Serial.println("Nie udalo sie odczytac danych z BMP180!\n");
}
else Serial.println("Nie udalo sie odczytac danych z BMP180\n");
}
else Serial.println("Nie udalo sie odczytac danych z BMP180!\n");

//opoznienie pomiedzy pomiarami
delay(5000);
}

```

Listing Kodu z dodanym modulem wifi:

```

#include "WiFiEsp.h"
#include <Adafruit_BMP085.h>
#include "DHT.h"
//#include <BH1750.h>

```

```

// Piny 2 i 3 przy poborze danych
#ifndef HAVE_HWSERIAL1
#include "SoftwareSerial.h"
SoftwareSerial Serial1(2, 3); // RX, TX
#endif

Adafruit_BMP085 bmp;
#define ALTITUDE 85.0

//Pin 7 przy poborze danych z DHT11
#define DHTPIN 7
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

//BH1750 lightMeter;

char ssid[] = "StacjaPogodowa"; // nazwa sieci SSID
char pass[] = "12345678"; // haslo
int status = WL_IDLE_STATUS; // status WiFi
int reqCount = 0; // liczba otrzymanych requestow
int ledStatus = LOW;
WiFiEspServer server(80);

// RingBuffer pomaga zwiekszyc szybko w przesyle danych
RingBuffer buf(8);

void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
  Serial.begin(115200); // inicjalizuje polaczenie szeregowe z Arduino
  Serial1.begin(9600); // inicjalizuje polaczenie szeregowe z modulem WiFi

  Wire.begin();
  bmp.begin();
  dht.begin();
  lightMeter.begin();

  WiFi.init(&Serial1); // inicjalizacja modulu ESP

  Serial.println(ssid);

  // access point
  status = WiFi.beginAP(ssid, 10, pass, ENC_TYPE_WPA2_PSK);

  Serial.println("Access point ruszyl");
  printWifiStatus();

  // serwer uruchomiony na porcie 80
  server.begin();
  Serial.println("Serwer ruszy ");

```

```

}

float lux;
float h;
char status;
double T,P,p0;

void loop()
{
    WiFiEspClient client = server.available();
    startMeasure();
    if (client) {
        Serial.println("Nowy klient");
        buf.init();
        while (client.connected()) {
            if (client.available())
                char c = client.read();
                buf.push(c);

            // zakończenie HTTP requestu to dwa znaki nowej linii
            if (buf.endsWith("\r\n\r\n")) {
                sendHttpResponse(client);
                break;
            }
            // Sprawdzenie czy zadaniem od klienta nie było włączenie lampki
            if (buf.endsWith("GET /H")) {
                Serial.println("LED włączony");
                ledStatus = HIGH;
                digitalWrite(LED_BUILTIN, HIGH);
            }
            else if (buf.endsWith("GET /L")) {
                Serial.println("LED wyłączony");
                ledStatus = LOW;
                digitalWrite(LED_BUILTIN, LOW);
            }
        }
    }

    // czas dla przeglądarki na odbiór danych
    delay(10);

    // close the connection
    client.stop();
    Serial.println("Klient rozłączony");
}

void sendHttpResponse(WiFiEspClient client)
{
    client.println("HTTP/1.1 200 OK");
}

```

```

client.println("Content-type:text/html");
client.println();

// Zawarto stronki z danymi
client.println("<!DOCTYPE HTML>");
client.println("<html>");
client.println("<body style=\"background-color:powderblue;\">");
client.println("<h1 style=\"font-size:5vw;\">Witaj w moim domu:</h1>");
client.println("<h2 style=\"font-size:3vw;\">Temperatura: ");
client.print(T);
client.print(" &degC<br>Wilgotnosc: ");
client.print(h);
client.print(" %<br>Cisnienie: ");
client.print(P);
client.print(" hPa<br>Natezenie swiatla: ");
client.print(lux);
client.println(" lx<br><br>");

client.print("LED: ");
client.print(ledStatus);
client.println("<br>Wejdz <a href=\"/H\">tu</a> by wlaczyc (1)<br>");
client.println("Wejdz <a href=\"/L\">tu</a> by wylaczyc (0)</h2><br>");
client.println("</body>");
client.print("</html>\r\n");

// Odpowiedz protokolu HTTP to znak nowej linii
client.println();
}

void printWifiStatus()
{
    // Wypisanie adresu IP
    IPAddress ip = WiFi.localIP();
    Serial.print("IP Address: ");
    Serial.println(ip);

    // Informacje na temat gdzie przejsc, by zobaczyc strone
    Serial.println();
    Serial.print("By zobaczy strone, polacz sie z siecia");
    Serial.print(ssid);
    Serial.print(" i otworz przegladarke na http://");
    Serial.println(ip);
    Serial.println();
}

void startMeasure()
{
    lux = lightMeter.readLightLevel();
    if (isnan(lux)) {
        Serial.println(F("Nie udalo sie odczytac danych z czujnika BH1750"));
        lux = -1;
    }
}

```

```

    }

    T = bmp.readTemperature();
    if (isnan(T)) {
        Serial.println(F("Nie odczytano danych z BMP180 - temperatura!"));
        T = -1;
    }

    P = bmp.readPressure();
    if (isnan(P)) {
        Serial.println(F("Nie odczytano danych z BMP180 - cisnienie!"));
        P = -1;
    } P=P/100;

    h = dht.readHumidity();
    if (isnan(h)) {
        Serial.println(F("Nie odczytano danych z DHT11 - wilgotnosc!"));
        h = -1;
    }
}

```