

Listingi programów

LISTING 1 – „Hello World” – miganie diodą

```
void setup() {  
    pinMode(17, OUTPUT);    // GPIO w trybie wyjścia  
}  
void loop() {  
    digitalWrite(17, HIGH); // LED świeci  
    delay(1000);            // poczekaj 1 s  
    digitalWrite(17, LOW);  // LED nie świeci  
    delay(1000);            // poczekaj 1000 ms = 1 s  
}
```

LISTING 2 – GPIO wejście + Serial

```
void setup() {  
    pinMode(0, INPUT);      // GPIO DP0 w trybie wejścia  
    Serial.begin(9600);     // inicjalizacja Serial  
}  
  
void loop() {  
    bool stan = digitalRead(0); // odczytaj stan  
    Serial.println(stan);       //wyślij stan  
    delay(1000);               // poczekaj 1 s  
}
```

LISTING 3 – ADC + Serial

```
void setup() {  
    Serial.begin(9600);           // inicjalizacja Serial  
}  
  
void loop() {  
    int value = analogRead(A0);   // odczytaj napięcie z A0  
    float miliwolt = value*4.88;  // przelicz na mV  
  
    Serial.print("LSB = ");       // wypisz tekst "LSB = "  
    Serial.println(value);        // wypisz wartość value  
  
    Serial.print("mV = ");  
    Serial.println(miliwolt);  
    delay(1000);                  // poczekaj 1 s  
}
```

LISTING 4 – Czujnik temperatury i ciśnienia BMP180

```
#include <qbcn.h>

BMP180 bmp;

void setup() {
    Serial.begin(9600);

    // Inicjalizacja sensora
    if (bmp.begin()) {
        Serial.println("BMP180 init success");
    }
}

void loop() {
    // Deklaracja zmiennych na temperaturę i ciśnienie
    double T, P;
    // Odczytaj dane z czujnika
    bmp.getData(T,P);
    // Wyślij dane przez Serial port
    Serial.print("Absolute pressure: ");
    Serial.print(P,2);
    Serial.println(" mb.");
    Serial.print("Temperature: ");
    Serial.print(T,2);
    Serial.println(" deg C.");
}
```

LISTING 5 – Hello radio – nadawanie

```
//Include the required libraries

#include <qbcn.h>

//Radio Parameters

constexpr int NodeID = 2;

constexpr int NetworkID = 100;

constexpr int TargetID = 1;


//Radio object

RFM69 radio;


void setup()
{
    //Initialize serial connection for debugging
    Serial.begin(9600);
    Serial.println("REBOOT");
    delay(1000);
    //Initialize radio
    radio.initialize(FREQUENCY,NodeID,NetworkID);
    radio.setHighPower();
    radio.setFrequency(433000000);
    Serial.println("Transmitting at 433 Mhz");
}
```

```
void loop() {  
    radio.send(TargetID, "CanSat 2017 ESERO-PL", 20);  
    Serial.println("Packet sent!");  
    delay(500);  
}
```

LISTING 6 – Hello radio – odbiornik

```
//Include the required libraries  
#include <qbcn.h>  
  
//Radio Parameters  
constexpr int NodeID = 1;  
constexpr int NetworkID = 100;  
  
//Radio object  
RFM69 radio;  
  
void setup()  
{  
    Serial.begin(9600);  
    Serial.println("REBOOT");  
    delay(1000);  
    //Initialize radio  
    radio.initialize(FREQUENCY, NodeID, NetworkID);
```

```
radio.setFrequency(433000000);  
    radio.setHighPower();  
    Serial.println("Listening at 433 Mhz");  
}  
void loop() {  
    if (radio.receiveDone()) {  
        Serial.print("Packet received: ");  
  
        for (byte i = 0; i < radio.DATALEN; i++) {  
            Serial.print((char)radio.DATA[i]);  
        }  
        Serial.println();  
        Serial.print("RX_RSSI: ");  
        Serial.println(radio.RSSI);  
    }  
}
```

LISTING 7 – Wysyłanie danych z czujników

```
//Include the required libraries

#include <qbcn.h>

//Radio Parameters

constexpr int NodeID = 2;

constexpr int NetworkID = 100;

constexpr int TargetID = 1;


//Radio object

char payload[100];

RFM69 radio;

BMP180 bmp;


void setup()

{

    //Initialize serial connection for debugging

    Serial.begin(9600);

    Serial.println("REBOOT");

    if (bmp.begin()) {

        Serial.println("BMP180 init success");

    }

    delay(1000);
```

```
//Initialize radio  
radio.initialize(FREQUENCY,NodeID,NetworkID);  
radio.setHighPower();  
radio.setFrequency(433000000);  
Serial.println("Transmitting at 433 Mhz");  
}  
void loop() {  
    double T, P;  
    bmp.getData(T,P);  
  
    char payload[100];  
    long int P_int = P*100;  
    long int T_int = T*100;  
  
    sprintf(payload, "P: %ld T: %ld", P_int, T_int);  
    radio.send(TargetID, payload, strlen(payload));  
    delay(500);  
}
```