**ДОДАТОК 1**

**Код прошивки мікроконтролера блоку метеоспостережень**

// include modules

#include <SPI.h>

#include <DigitalIO.h>

#include <RF24.h>

#include <OneWire.h>

#include <BH1750.h>

#include <DallasTemperature.h>

// define SPI pins for arduino nano

#define CE 9

#define CSN 10

#define SCK 13

#define MOSI 11

#define MISO 12

// define sensors constants

#define LIGHT\_SENSOR\_ID 0

#define TEMPERATURE\_SENSOR\_ID 1

// define sending structures

struct SensorData

{

    int id;

    float value;

};

// init global modules

RF24 radio(CE, CSN);

BH1750 lightSensor;

OneWire temperatureSensorLink(2);

DallasTemperature temperatureSensor(&temperatureSensorLink);

void setup()

{

    // open serial port

    Serial.begin(9600);

    // setup modules

    setupRadio();

    setupSensors();

}

void setupRadio()

{

    // set default props

    const int channel = 12;

    const uint64\_t pipe = 0xFFFFFFFFFFLL;

    // setup radio

    radio.begin();

    radio.setChannel(channel);

    radio.setDataRate(RF24\_250KBPS);

    radio.setPALevel(RF24\_PA\_MIN);

    radio.openWritingPipe(pipe);

}

void setupSensors()

{

    // setup light sensors

    Wire.begin();

    lightSensor.begin(BH1750::ONE\_TIME\_HIGH\_RES\_MODE);

    // setup temperature sensor

    temperatureSensor.begin();

}

void loop()

{

    // read light data

    float lightValue = (float)lightSensor.readLightLevel();

    SensorData light = { LIGHT\_SENSOR\_ID, lightValue };

    Serial.print("Light: ");

    Serial.print(lightValue);

    Serial.println(" lx");

    // read temperature data

    temperatureSensor.requestTemperatures();

    float temperatureValue = (float)temperatureSensor.getTempCByIndex(0);

    SensorData temperature = { TEMPERATURE\_SENSOR\_ID, temperatureValue };

    Serial.print("Temperature: ");

    Serial.print(temperatureValue);

    Serial.println(" C");

    // send data via radio

    SensorData sendPackage[] = { light, temperature };

    radio.write(&sendPackage, sizeof(sendPackage));

    Serial.println("Sending data...\n");

    // delay process

    delay(5000);

}