

Додаток А
Програмний модуль зчитування та передачі даних

Програмний модуль для зчитування з BH1750 та HC-SR501 і формування JSON повідомлення для публікації через MQTT:

```
#include <Wire.h>
#include <BH1750.h>
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>

// Налаштування Wi-Fi
const char* ssid = "YOUR_WIFI_SSID";
const char* password = "YOUR_WIFI_PASSWORD";

// Налаштування MQTT
const char* mqtt_server = "YOUR_MQTT_BROKER_IP";
const int mqtt_port = 1883;
const char* mqtt_topic = "streetlighting/sensor_data";

// Ініціалізація об'єктів
BH1750 lightMeter;
WiFiClient espClient;
PubSubClient client(espClient);

// HC-SR501 – GPIO PIN (наприклад, D5)
const int motionPin = 5;
int motionState = LOW;

void setup_wifi() {
    delay(10);
    Serial.println();
    Serial.print("Connecting to ");
```

```
Serial.println(ssid);

WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("\nWiFi connected");
Serial.print("IP address: ");
Serial.println(WiFi.localIP());
}

void reconnect() {
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    if (client.connect("ESP32Client")) {
      Serial.println("connected");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}

void setup() {
  Serial.begin(115200);

  // Ініціалізація I2C та BH1750
```

```
Wire.begin();
lightMeter.begin();
Serial.println(F("BH1750 test begin"));

// Ініціалізація HC-SR501
pinMode(motionPin, INPUT);

// Ініціалізація Wi-Fi та MQTT
setup_wifi();
client.setServer(mqtt_server, mqtt_port);
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();

  // Зчитування освітленості (люкси)
  float lux = lightMeter.readLightLevel();
  Serial.print("Light: ");
  Serial.print(lux);
  Serial.println(" lx");

  // Зчитування стану HC-SR501
  motionState = digitalRead(motionPin);
  Serial.print("Motion detected: ");
  Serial.println(motionState);

  // Створення JSON повідомлення
```

```
StaticJsonDocument<200> doc;
doc["timestamp"] = "2025-06-06T16:30:15Z"; // Приклад – у реальному кодi
можна згенерувати час
doc["sensor_id"] = "node_01";
doc["light"] = lux;
doc["motion"] = motionState;

char buffer[256];
size_t n = serializeJson(doc, buffer);

// Публікація в MQTT
client.publish(mqtt_topic, buffer, n);
Serial.println("Published to MQTT");

// Затримка між циклами (наприклад, 5 сек)
delay(5000);
}
```

Додаток Б
Відправка даних через MQTT

Публікації даних з вузла IoT у брокер MQTT:

```
#include <Wire.h>
#include <BH1750.h>
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>

// Налаштування Wi-Fi
const char* ssid = "YOUR_WIFI_SSID";
const char* password = "YOUR_WIFI_PASSWORD";

// Налаштування MQTT
const char* mqtt_server = "YOUR_MQTT_BROKER_IP";
const int mqtt_port = 1883;
const char* mqtt_topic = "streetlighting/sensor_data";

// Ініціалізація об'єктів
BH1750 lightMeter;
WiFiClient espClient;
PubSubClient client(espClient);

// HC-SR501 – GPIO PIN (наприклад, D5)
const int motionPin = 5;
int motionState = LOW;

// Підключення до Wi-Fi
void setup_wifi() {
  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nWiFi connected");
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
}

// Підключення до MQTT
void reconnect() {
```

```

while (!client.connected()) {
  Serial.print("Attempting MQTT connection...");
  if (client.connect("ESP32Client")) {
    Serial.println("connected");
  } else {
    Serial.print("failed, rc=");
    Serial.print(client.state());
    Serial.println(" try again in 5 seconds");
    delay(5000);
  }
}
}
}

```

```

void setup() {
  Serial.begin(115200);

  // Ініціалізація I2C та BH1750
  Wire.begin();
  lightMeter.begin();
  Serial.println(F("BH1750 test begin"));

  // Ініціалізація HC-SR501
  pinMode(motionPin, INPUT);

  // Підключення Wi-Fi та MQTT
  setup_wifi();
  client.setServer(mqtt_server, mqtt_port);
}

```

```

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();

  // Зчитування освітленості (люкси)
  float lux = lightMeter.readLightLevel();
  Serial.print("Light: ");
  Serial.print(lux);
  Serial.println(" lx");

  // Зчитування стану HC-SR501
  motionState = digitalRead(motionPin);
  Serial.print("Motion detected: ");
  Serial.println(motionState);
}

```

```
// Формування JSON-повідомлення
JsonObject doc;

// Тут – у реальному кодi бажано використовувати час із NTP, зараз – статичний
приклад:
doc["timestamp"] = "2025-06-06T16:35:10Z";
doc["sensor_id"] = "node_01";
doc["light"] = lux;
doc["motion"] = motionState;

// Сериалізація у буфер
char buffer[256];
size_t n = serializeJson(doc, buffer);

// Публікація у брокер MQTT
if (client.publish(mqtt_topic, buffer, n)) {
  Serial.println("MQTT Publish successful:");
  Serial.println(buffer);
} else {
  Serial.println("MQTT Publish failed");
}

// Затримка між публікаціями (наприклад, 5 сек)
delay(5000);
}
```

Додаток В
Програмний модулі системи

Програмний модуль для для ESP32

```

#include <WiFi.h>
#include <PubSubClient.h>
#include <Wire.h>
#include <BH1750.h>

#define PIR_PIN 14 // Сенсор руху
#define RELAY_PIN 27 // Реле ліхтаря

const char* ssid = "YOUR_WIFI_SSID";
const char* password = "YOUR_WIFI_PASSWORD";
const char* mqtt_server = "BROKER_IP";

WiFiClient espClient;
PubSubClient client(espClient);
BH1750 lightMeter;

unsigned long lastMsg = 0;
float lux = 0;
bool motionDetected = false;

void setup_wifi() {
  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
}

void callback(char* topic, byte* payload, unsigned int length) {
  // Якщо хочеш – тут можна реагувати на команди з брокера (не обов'язково)
}

void reconnect() {
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");

```

```

if (client.connect("StreetLightController")) {
  Serial.println("connected");
  client.subscribe("streetlight/control"); // Приклад топика
} else {
  Serial.print("failed, rc=");
  Serial.print(client.state());
  Serial.println(" try again in 5 seconds");
  delay(5000);
}
}
}

void setup() {
  Serial.begin(115200);
  Wire.begin();
  lightMeter.begin();

  pinMode(PIR_PIN, INPUT);
  pinMode(RELAY_PIN, OUTPUT);
  digitalWrite(RELAY_PIN, LOW); // Початково вимкнено

  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();

  unsigned long now = millis();
  if (now - lastMsg > 2000) { // Оновлення кожні 2 секунди
    lastMsg = now;

    lux = lightMeter.readLightLevel();
    motionDetected = digitalRead(PIR_PIN);

    Serial.print("Lux: ");
    Serial.print(lux);
    Serial.print(" | Motion: ");
    Serial.println(motionDetected);

    // Простий алгоритм керування

```

```
if (lux < 50) { // Поріг освітленості
  if (motionDetected) {
    digitalWrite(RELAY_PIN, HIGH); // Вмикаємо світло
  } else {
    digitalWrite(RELAY_PIN, LOW); // Вимикаємо світло
  }
} else {
  digitalWrite(RELAY_PIN, LOW); // Вдень – вимикаємо світло
}

// Публікуємо стан у MQTT
String payload = String("{\"lux\":") + lux + ", \"motion\": " + motionDetected + "}";
client.publish("streetlight/status", payload.c_str());
}
}
```

Додаток Г
Демонстраційний матеріал

