

## ДОДАТОК А

## Лістинг розробленої програми

```
#define BLYNK_TEMPLATE_ID TEMPL4TGIDOOwR
#define BLYNK_TEMPLATE_NAME Project1
#define BLYNK_AUTH_TOKEN Jnra1bhfnSZKN-ORibtio8yMBKyd52S9

#include <AccelStepper.h>
#include <Wire.h>
#include <WiFi.h>
#include <Arduino.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include DHTesp.h

#define BLYNK_PRINT Serial

#define humidity_vpin V1
#define temperature_vpin V2
#define co2_vpin V8
#define rain_vpin V6
#define temperature2_vpin V7

#define manual_mode_button V4
#define perform_manual_action_button V5

char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = Wokwi-GUEST;
char pass[] = ;
```

```
BlynkTimer timer;
```

```
DHTesp dhtSensor;
```

```
DHTesp dhtSensor2;
```

```
#define CO2_PIN 34
```

```
#define RAIN_PIN 35
```

```
const int dirPin = 4;
```

```
const int stepPin = 25;
```

```
const int stepsPerRevolution = 200;
```

```
AccelStepper stepper(AccelStepper::FULL4WIRE, stepPin, dirPin);
```

```
class SensorData {
```

```
public:
```

```
struct TempAndHumidity {
```

```
    float temperature;
```

```
    float humidity;
```

```
    int co2;
```

```
    int rain;
```

```
    float temperature2;
```

```
    float humidity2;
```

```
};
```

```
TempAndHumidity getSensorData() {
```

```
    TempAndHumidity data;
```

```
    data.temperature = dhtSensor.getTemperature();
```

```
    data.humidity = dhtSensor.getHumidity();
```

```
    data.temperature2 = dhtSensor2.getTemperature();
```

```
    data.humidity2 = dhtSensor2.getHumidity();
```

```
int16_t co2Value = analogRead(CO2_PIN);
int16_t rainValue = analogRead(RAIN_PIN);
data.co2 = co2Value * 2;
data.rain = rainValue;
return data;
}
};
enum RoofState {
    CLOSED,
    OPENING,
    OPEN,
    CLOSING,
    MANUAL
};
```

```
String getRoofStateString(RoofState state) {
    switch (state) {
        case CLOSED:
            return Closed;
        case OPENING:
            return Opening;
        case OPEN:
            return Open;
        case CLOSING:
            return Closing;
        case MANUAL:
            return Manual;
        default:
            return Unknown;
    }
}
```

```
}  
}
```

```
RoofState roofState = CLOSED;  
RoofState prevRoofState = CLOSED;
```

```
int manualModeStatus = 0;  
int manualActionStatus = 0;  
bool manualControlEnabled = false;
```

```
float voltage = 1;  
SensorData sensorData;  
SensorData::TempAndHumidity data;
```

```
unsigned long previousMotorMillis = 0;  
const long motorInterval = 1000;  
const int roofDistance = 1600;
```

```
unsigned long previousSensorMillis = 0;  
const long sensorInterval = 2000;
```

```
const int manualStep = 400;
```

```
void controlRoof();  
void openRoof();  
void closeRoof();  
void manualControl();  
float calculateEnergyConsumed();  
BLYNK_WRITE_DEFAULT() {  
    int pin = request.pin;
```

```
int value = param.asInt();

if (pin == manual_mode_button) {
    manualModeStatus = value;
    Blynk.virtualWrite(manual_mode_button, manualModeStatus);
    Serial.println(Manual Mode: + String(manualModeStatus));

    manualControlEnabled = (manualModeStatus == 1);
} else if (pin == perform_manual_action_button && manualControlEnabled) {
    manualActionStatus = value;
    Blynk.virtualWrite(perform_manual_action_button, manualActionStatus);
    Serial.println(Manual Action: + String(manualActionStatus));

    if (manualActionStatus == 1) {
        closeRoof();
    } else if (manualActionStatus == 0) {
        openRoof();
    }

    manualActionStatus = 0;
} else if (pin == perform_manual_action_button && !manualControlEnabled &&
value == 1) {

    if (manualModeStatus == 1) {
        if (roofState == OPEN) {
            closeRoof();
        } else if (roofState == CLOSED) {
            openRoof();
        }
    }
}
```

```
}  
}  
  
void setup() {  
  Serial.begin(115200);  
  Blynk.begin(auth, ssid, pass);  
  
  Blynk.virtualWrite(V0, String(roofState));  
  
  dhtSensor.setup(14, DHTesp::DHT22);  
  dhtSensor2.setup(27, DHTesp::DHT22);  
  
  sensorData = SensorData();  
  
  timer.setInterval(2000L, controlRoof);  
  
  pinMode(stepPin, OUTPUT);  
  pinMode(dirPin, OUTPUT);  
  digitalWrite(dirPin, HIGH);  
  
  stepper.setAcceleration(400);  
  stepper.setMaxSpeed(2000);  
}  
  
void loop() {  
  Blynk.run();  
  
  unsigned long currentMillis = millis();  
  
  if (currentMillis - previousSensorMillis >= sensorInterval) {
```

```
previousSensorMillis = currentMillis;
data = sensorData.getSensorData();
sendSensorData();
controlRoof();
}
if (roofState == OPENING || roofState == CLOSING || roofState == MANUAL) {
    stepper.run();
}
}
void controlRoof() {
    Blynk.run();
    Serial.println(Roof State: + getRoofStateString(roofState));

    if (roofState != prevRoofState) {
        Blynk.virtualWrite(V0, Roof State: + getRoofStateString(roofState));
        prevRoofState = roofState;
    }

    if (manualControlEnabled) {
        manualControl();
        return;
    }

    int rainValue = analogRead(RAIN_PIN);
    data = sensorData.getSensorData();

    Serial.println(Rain Value: + String(rainValue));
    Serial.println(Temperature: + String(data.temperature));
    Serial.println(Humidity: + String(data.humidity));
    Serial.println(CO2: + String(data.co2));
```

```
if (rainValue >= 512) {
  if (roofState != CLOSED) {
    Serial.println(Closing roof because of rain);
    closeRoof();
  }
} else {
  Serial.println(No rain, checking other conditions...);

  if (data.temperature >= 18 && data.temperature <= 24) {
    if (data.humidity >= 30 && data.humidity <= 60) {
      if (data.co2 < 1000) {
        if (roofState == OPEN || roofState == OPENING) {
          Serial.println(Closing roof because of conditions);
          closeRoof();
        }
      } else {
        if (roofState != OPEN) {
          Serial.println(Opening roof because of high CO2);
          openRoof();
        }
      }
    } else {
      if (roofState != OPEN) {
        Serial.println(Opening roof because of low humidity);
        openRoof();
      }
    }
  } else {
    if (rainValue < 512) {
```

```
if (data.temperature > 24) {
  if (data.humidity >= 30 && data.humidity <= 60) {
    if (data.co2 < 1000) {
      if (roofState == OPEN || roofState == OPENING) {
        Serial.println(Closing roof because of high temperature);
        closeRoof();
      }
    } else {
      if (roofState != OPEN) {
        Serial.println(Opening roof because of high temperature and high CO2);
        openRoof();
      }
    }
  } else {
    if (roofState != OPEN) {
      Serial.println(Opening roof because of high temperature and low humidity);
      openRoof();
    }
  }
} else if (data.temperature < 18) {
  if (data.humidity >= 30 && data.humidity <= 60) {
    if (data.co2 < 1000) {
      if (roofState == OPEN || roofState == OPENING) {
        Serial.println(Closing roof because of low temperature);
        closeRoof();
      }
    } else {
      if (roofState != OPEN) {
        Serial.println(Opening roof because of low temperature and high CO2);
        openRoof();
      }
    }
  }
}
```

```

    }
} else {
    if (roofState != OPEN) {
        Serial.println(Opening roof because of low temperature and low humidity);
        openRoof();
    }
} else if (data.temperature > 24) {
    if (data.humidity >= 30 && data.humidity <= 60) {
        if (data.co2 < 1000 && data.temperature2 > 24) {
            if (roofState == OPEN || roofState == OPENING) {
                Serial.println(Closing roof because of high temperature, low humidity, low
CO2, and high secondary temperature);
                closeRoof();
            }
        } else {
            if (roofState != OPEN) {
                Serial.println(Opening roof because of high temperature, low humidity,
low CO2, and high secondary temperature);
                openRoof();
            }
        }
    } else {
        if (roofState != OPEN) {
            Serial.println(Opening roof because of high temperature, low humidity, and
high secondary temperature);
            openRoof();
        }
    }
}

```

```

    } else if (data.temperature > 24) {
        if (data.humidity < 30 || data.humidity > 60 || data.co2 > 1000 ||
data.temperature2 < 24) {
            if (roofState != OPEN) {
                Serial.println(Opening roof because of high temperature, out-of-range
humidity, high CO2, or low secondary temperature);
                openRoof();
            }
        }
    } else if (data.temperature < 18) {
        if (data.humidity >= 30 && data.humidity <= 60) {
            if (data.co2 < 1000 && data.temperature2 < 18) {
                if (roofState == OPEN || roofState == OPENING) {
                    Serial.println(Closing roof because of low temperature, normal humidity,
low CO2, and low secondary temperature);
                    closeRoof();
                }
            } else {
                if (roofState != OPEN) {
                    Serial.println(Opening roof because of low temperature, normal humidity,
low CO2, and low secondary temperature);
                    openRoof();
                }
            }
        } else {
            if (roofState != OPEN) {
                Serial.println(Opening roof because of low temperature, out-of-range
humidity, low CO2, or low secondary temperature);
                openRoof();
            }
        }
    }

```

```

    }} else {
    if (roofState != OPEN) {
        Serial.println(Opening roof for general conditions);
        openRoof();
    }}}}}
float calculateEnergyConsumed() {
    float voltage = 12.0; // Припустимо, що напруга дорівнює 12 В
    float motorIntervalInSeconds = motorInterval / 1000.0;
    float motorPowerConsumption = 5.0;
        float energyConsumed = motorPowerConsumption * voltage *
motorIntervalInSeconds;
return energyConsumed;}
void sendSensorData() {
    SensorData::TempAndHumidity data = sensorData.getSensorData();
    Serial.println(—);
    Serial.println(Temp: + String(data.temperature, 2) + C);
    Serial.println(Humidity: + String(data.humidity, 1));
    Serial.println(CO2: + String(data.co2));
    Blynk.virtualWrite(humidity_vpin, data.humidity);
    Blynk.virtualWrite(temperature_vpin, data.temperature);
    Blynk.virtualWrite(co2_vpin, data.co2);
    Blynk.virtualWrite(rain_vpin, data.rain);
    Blynk.virtualWrite(temperature2_vpin, data.temperature2);
}
void openRoof() {
    digitalWrite(dirPin, LOW);
    if (roofState != OPENING) {
        roofState = OPENING;
        stepper.moveTo(roofDistance);

```

```
    }}  
void closeRoof() {  
    digitalWrite(dirPin, HIGH);  
    if (roofState != CLOSING) {  
        roofState = CLOSING;  
        stepper.moveTo(0);}}  
void manualControl() {  
    if (manualModeStatus == 1) {  
        if (manualActionStatus == 1) {  
            if (roofState == CLOSED) {  
                digitalWrite(dirPin, LOW);  
                stepper.moveTo(roofDistance);  
                roofState = OPEN;  
            } else if (roofState == OPEN) {  
                digitalWrite(dirPin, HIGH);  
                stepper.moveTo(0);  
                roofState = CLOSED;}}  
        manualActionStatus = 0;  
        roofState = MANUAL;  
    } else {  
        Serial.println(Manual Mode is disabled. Cannot perform manual actions.);  
    }}  
}}
```

ДОДАТОК Б  
АПРОБАЦІЯ РЕЗУЛЬТАТІВ

# Модернізація модуля керування розсувними конструкціями виробничого приміщення з використанням ІоТ

Бронніков А.І.<sup>1.</sup>, Чернишов Д.І.<sup>2.</sup>

1. Кафедра КІТАМ, Харківський національний університет радіоелектроніки, УКРАЇНА, Харків, пр. Науки. 14., email: artem.bronnikov@nure.ua

2. Кафедра КІТАМ, Харківський національний університет радіоелектроніки, УКРАЇНА, Харків, пр. Науки. 14., email: denys.chernyshov@nure.ua

**Анотація:** В даній роботі розглянуто питання модернізація модуля керування розсувними конструкціями виробничого приміщення з використанням технології ІоТ та Arduino в якості плати керування. Використання ІоТ-технологій для задачі управління розсувними конструкціями вносить важливі питання безпеки та захисту даних, що стають залежними від надійності та функціонування виробничого процесу.

**Ключові слова:** розробка, розсувні системи, технологія Internet of Things(IoT).

## I. ВСТУП

Сучасний світ промисловості характеризується непередбачуваною динамікою та безперервним стремлінням до вдосконалення процесів та підвищення продуктивності. У цьому контексті, однією з найважливіших сфер, яку необхідно переглянути та оптимізувати, є управління розсувними конструкціями виробничих приміщень. Ця сфера стає ключовою для забезпечення ефективності та зручності виробництва [1]. Однак, із швидким розвитком технологій, зокрема, Інтернету речей (ІоТ), відкривається унікальна можливість революціонізувати управління цими конструкціями та забезпечити їхню автоматизацію на основі платформи Arduino.

Основний напрямок роботи полягає в можливостях та перевагах модернізації модуля керування розсувними конструкціями виробничого приміщення на основі платформи Arduino з використанням технологій Інтернету речей, а також впровадження сучасних технологій в процес управління розсувними конструкціями, оцінити їхній вплив на продуктивність та безпеку і дослідити можливості оптимізації робочого середовища [2].

Ця задача стає надзвичайно актуальною в сучасному підприємницькому середовищі, де кожна мить та кожен ресурс мають вагомий цінність. Модернізація модуля керування розсувними конструкціями на базі Arduino та ІоТ може призвести до значних переваг, таких як зменшення витрат, підвищення продуктивності та покращення умов праці. Цей вступ визначає перед нами завдання розглядати перспективний шлях до досягнення оптимізації виробничого середовища, що надає підприємствам конкурентні переваги у сучасному світі виробництва.

## II. ВПРОВАДЖЕННЯ ІОТ-ТЕХНОЛОГІЙ ДЛЯ УПРАВЛІННЯ РОЗСУВНИМИ КОНСТРУКЦІЯМИ

Сучасні розсувні конструкції представляють собою важливий компонент виробничих приміщень та складів, які допомагають підприємствам оптимізувати їхнє просторове розташування та раціонально використовувати обмежену площу. Вони відіграють рішучу роль у забезпеченні продуктивності та безпеки виробничих процесів.

Значущість розсувних конструкцій у виробничому середовищі проявляється у кількох ключових аспектах:

1. функціональність і регулювання простору: розсувні конструкції надають підприємствам можливість ефективно організувати робочі простори, дозволяючи змінювати їх розмір та конфігурацію відповідно до потреб виробництва. Це дозволяє мінімізувати втрати простору та максимізувати його використання;

2. забезпечення безпеки: розсувні конструкції грають ключову роль у забезпеченні безпеки працівників та матеріалів виробничого процесу. Вони можуть служити бар'єром між небезпечними зонами та робочими областями, а також дозволяють швидко втручатися в ситуації, коли це необхідно;

3. збереження енергії: деякі сучасні розсувні конструкції оснащені технологіями енергозбереження, які допомагають підприємствам зменшити витрати на опалення та охолодження приміщень, залишаючи невикористані зони відокремленими від робочих зон;

4. відповідність стандартам: у багатьох галузях існують строгі стандарти щодо безпеки та регулювань. розсувні конструкції допомагають підприємствам відповідати цим вимогам та представити доступ до різних зон.

Загалом, розсувні конструкції стали невід'ємною частиною сучасних виробничих приміщень і відіграють стратегічну роль у забезпеченні їхньої функціональності та продуктивності. Таким чином, вдосконалення управління цими конструкціями за допомогою сучасних

оптимізації виробничих процесів і зменшенню витрат. Сенсори IoT дозволяють стежити за станом обладнання та передбачати можливі відмови чи поломки, попереджаючи невідкладні ремонти та підвищуючи надійність системи. Збір та аналіз даних в реальному часі надає можливість приймати інформовані рішення та розробляти стратегії виробництва [3].

Загалом, використання IoT для управління розсувними конструкціями виробничих приміщень є ключовим чинником у сучасному виробництві, що дозволяє підприємствам підвищити продуктивність, зменшити витрати та покращити якість продукції. Розвиток цих технологій обіцяє ще більше можливостей і переваг для майбутнього управління розсувними конструкціями.

### III. БЕЗПЕКА ТА ЗАХИСТ ДАНИХ У КОНТЕКСТІ ІОТ-ВИРІШЕНЬ ДЛЯ ВИРОБНИЧИХ ПРИМІЩЕНЬ

Використання IoT-вирішень для управління розсувними конструкціями виробничих приміщень вносить важливі питання безпеки та захисту даних, що стають залежними від надійності та нерухомості функціонування виробничого процесу. Такі як:

1. кібератаки та несанкціонований доступ: завдяки підключенню до мережі Інтернет речей, системи керування розсувними конструкціями можуть стати об'єктом кібератак. Це включає в себе спроби несанкціонованого доступу до системи, ведення вірусів або зловживання правами доступу [4];

2. перехоплення та злам зв'язку: зловмисники можуть намагатися перехопити комунікацію між IoT-пристроями та серверами, що може призвести до витоку конфіденційних даних або навіть контролю над системою. Нападники можуть використовувати IoT-пристрої для відправлення спаму, фішингових повідомлень або розсилки вірусів;

3. неактуальне програмне забезпечення: багато IoT-пристроїв мають обмежені ресурси та можуть бути вразливими до атак, якщо не отримують регулярних оновлень і вдосконалень вони можуть бути використанні зловмисниками.

Щоб зменшити ризики кібератак та несанкціонованих доступів можна використовувати:

а. шифрування: всі дані, які передаються між IoT-пристроями та серверами, мають бути зашифрованными для захисту від перехоплення та читання третіми особами;

б. аутентифікація та авторизація: для доступу до системи потрібно встановити суворі процедури аутентифікації та авторизації, щоб перешкоджати несанкціонованому входу.

Для запобігання перехоплення та зламу зв'язку можна використовувати моніторинг, резервне копіювання, ізоляція мережі та створення білих списків.

Моніторинг та виявлення загроз. Важливо мати системи моніторингу, які виявляють незвичайну активність та можливі загрози та сповіщають

адміністраторів.

Резервне копіювання. Регулярне створення резервних копій даних допомагає відновити інформацію у випадку її втрати або пошкодження.

Ізоляція мереж. Централізована мережа IoT повинна бути відокремлена від основної корпоративної мережі для запобігання несанкціонованому доступу до інших систем.

Створення білих списків. Дозволити доступ лише відомим та довіреним пристроям та користувачам, створюючи білі списки для затвердження підключення.

Важливо постійно оновлювати програмне забезпечення та встановлювати нові патчі для виправлення виявлених вразливостей, це допомагає утримувати систему в безпеці [5].

В сучасному світі, де IoT знаходить широке застосування в виробництві, безпека та захист даних стають критично важливими аспектами. Застосування вищезазначених стратегій та заходів допомагає зменшити ризик і забезпечує надійну захищеність виробничих IoT-систем.

### IV. МАТЕМАТИЧНА МОДЕЛЬ

Основна ідея полягає у використанні математичних виразів для відображення впливу впровадження IoT-технологій на показники ефективності та зручності управління розсувними конструкціями виробничого приміщення [6].

Ефективність виробництва ( $E_f$ ) може бути виражена як функція від таких параметрів:

1. кількість розсувних конструкцій ( $N$ );
2. час управління однією розсувною конструкцією без використання IoT ( $T_1$ );
3. час управління однією розсувною конструкцією з використанням IoT ( $T_2$ ).

$$E_f = \left(\frac{T_1}{T_2}\right) * N \quad (1)$$

Дана формула відображає, як швидкість управління розсувними конструкціями змінюється при впровадженні IoT. Чим більше значення  $E_f$ , тим ефективніше використовується час.

Зручність управління ( $C$ ) може бути оцінена як сума показників зручності для кожної розсувної конструкції:

- зручність без використання IoT ( $C_1$ );
- зручність з використанням IoT ( $C_2$ ).

$$C_N = \sum C_1 + \sum C_2 \quad (2)$$

де  $\sum$  вказує на суму значень для всіх розсувних конструкцій.

Ця модель дозволяє порівняти ефективність та зручність управління розсувними конструкціями до і після впровадження IoT.

Для розширення математичної моделі, яка відображає вплив впровадження IoT-технологій на управління

розсувними конструкціями виробничого приміщення, можна додати додаткові параметри та фактори, які впливають на ефективність та зручність управління:

1. швидкість IoT-з'єднання (S): представимо швидкість IoT-з'єднання як параметр, що впливає на час передачі даних та відповідь системи. Нехай S вимірюється в бітах на секунду (біт/с);

2. вартість впровадження IoT (C<sub>i</sub>): введемо вартість встановлення IoT-системи як вартість в доларах (USD);

3. надійність IoT-системи (R): представимо надійність як ймовірність безвідмовної роботи IoT-системи, де R є значенням від 0 до 1, де 1 відповідає повній надійності, а 0 - повній ненадійності;

4. віддаленість розсувних конструкцій (D): введемо відстань між розсувними конструкціями як величину, вимірювану в метрах (м);

5. потужність IoT-пристроїв (P): виразимо потужність пристроїв IoT у ваттах (Вт);

6. часовий горизонт аналізу (T): визначимо часовий горизонт аналізу у днях (дні).

Тепер можна оновимо математичну модель:

Ефективність виробництва (E<sub>f</sub>): Ми можемо оновити формулу для ефективності, враховуючи швидкість IoT-з'єднання:

$$E_f = \left( \frac{T_1}{T_2 + D/S} \right) * N \quad (3)$$

де D/S визначає час затримки через IoT-з'єднання на відстані D.

Зручність управління (C): Ми можемо оновити формулу для зручності, додавши параметр вартості та враховуючи надійність:

$$C_N = \sum C_1 + \sum C_2 - C_i * (1 - R) \quad (4)$$

де C<sub>i</sub> \* (1 - R) відображає витрати на впровадження IoT, зменшені на величину, яка залежить від надійності системи.

Таким чином, вдосконалена математична модель враховує додаткові параметри, що впливають на ефективність та зручність управління розсувними конструкціями виробничого приміщення після впровадження IoT-технологій.

## V. ВИСНОВКИ

Впровадження IoT-технологій розширює горизонти можливостей і допомагає перетворити традиційні виробничі процеси на більш продуктивні, безпечні та ефективні.

Сенсори IoT та системи моніторингу дозволяють збирати важливі дані в реальному часі, що надає можливість для прийняття інформованих рішень та запобігання можливим поломкам та відмовам обладнання. Віддалене керування та моніторинг роблять виробничий процес більш гнучким та адаптивним.

Підвищення продуктивності, зменшення витрат, покращення якості продукції та підвищення безпеки

працівників – це лише частина переваг, які вносить IoT в управління розсувними конструкціями. Розширення можливостей IoT може відкрити нові шляхи до інновацій та конкурентної переваги на ринку.

Однак разом зі всіма перевагами, важливо також звертати належну увагу до захисту даних та безпеки системи в контексті IoT. Застосування відповідних заходів безпеки є невід'ємною частиною успішної модернізації.

Загалом, модернізація модуля керування розсувними конструкціями за допомогою IoT є важливим кроком у розвитку виробництва та створенні більш продуктивного та конкурентоспроможного підприємства. Такі інновації відкривають нові можливості для підприємств та допомагають їм залишатися на передовому ринку.

Математична модель, яку ми розробили, дозволяє систематично оцінити вплив впровадження IoT на виробничий процес. Для цього ми використовуємо параметри, такі як кількість розсувних конструкцій, час управління до та після впровадження IoT, швидкість IoT-з'єднання, вартість впровадження, надійність системи, відстань між конструкціями, потужність IoT-пристроїв і часовий горизонт аналізу.

## ПЕРЕЛІК ПОСИЛАНЬ

[1] Інтелектуальні системи управління: Експертні системи – основи проектування та застосування в системах автоматизації [Електронний ресурс] : навч. посіб. для студ. спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» / КПІ ім. Ігоря Сікорського; уклад.: Л. Д. Ярошук. – Електронні текстові дані. – Київ : КПІ ім. Ігоря Сікорського, 2019. – 136с.

[2] Економіка галузевих ринків: Навчально-методичний комплекс для студентів спеціальності «051 Економіка», освітніх програм «Економіка та економічна політика», «Економіка та право», «Економіка підприємства» / Упорядники Ігнатюк А.І., Колоша В.В., Коваленко О.Я.– К., 2018. – 44 с.

[3] Технології інтернету речей. Навчальний посібник [Електронний ресурс]: навч. посіб. для студ. спеціальності 126 «Інформаційні системи та технології», спеціалізація «Інформаційне забезпечення робототехнічних систем» / Б. Ю. Жураковський, І.О. Зенів; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 12,5 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2021. – 271 с.

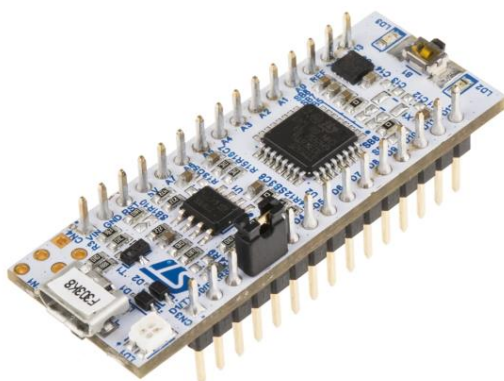
[4] Secure Channel Communication between IOT Devices and Computers / H. Kata et al. International Research Journal of Engineering and Technology (IRJET). 2023.

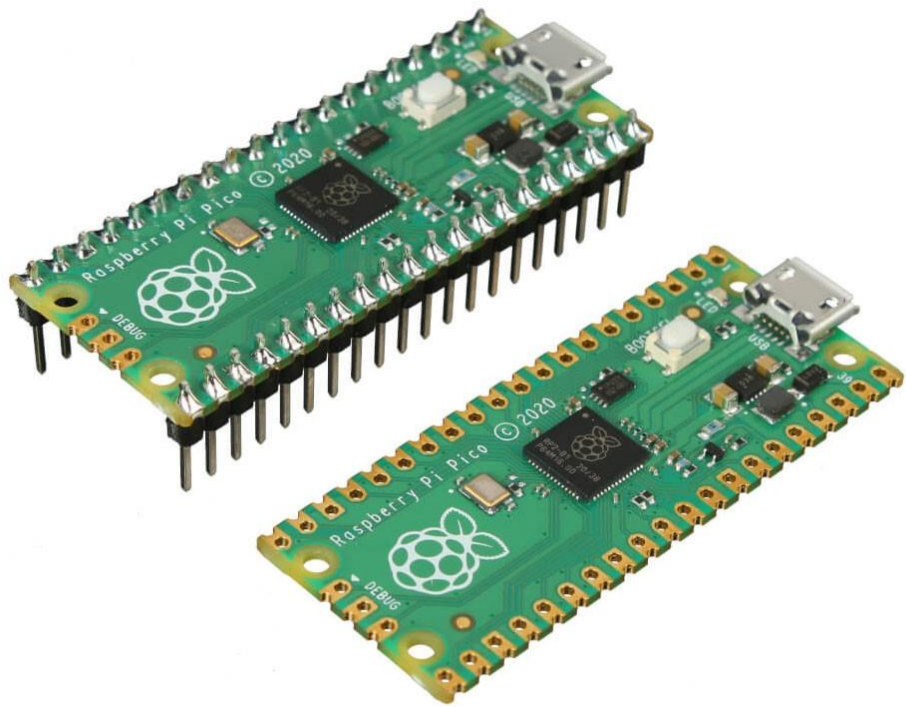
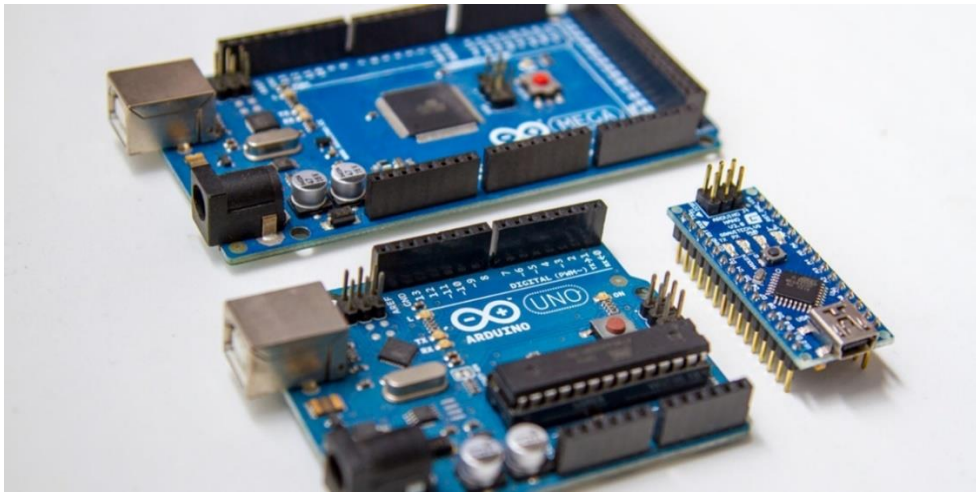
[5] Smith, L. (2021). Cyber Curiosity: A Beginner's Guide to Cybersecurity – How to Protect Yourself in the Modern World. New Degree Press

[6] Математичні моделі та новітні технології управління Конспект лекцій дисципліни «Математичне моделювання» для студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології»/ Упоряд. В.В. Безкоровайний. – Харків: ХНУРЕ, 2018. –

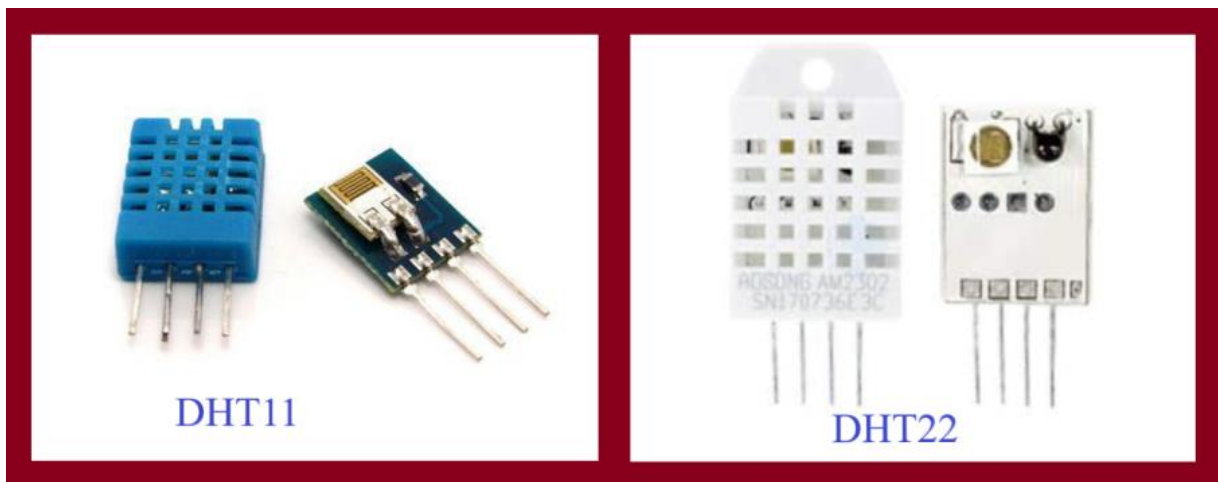
ДОДАТОК В  
ДЕМОНСТРАЦІЙНИЙ МАТЕРІАЛ

# МІКРОКОНТРОЛЕРИ ТА МІКРОПРОЦЕСОРИ STM32, ESP32, ARDUINO, RASPBERRY PI





ДАВАЧІ ОПАДІВ MLX90393, МН-RD, ДАВЧІ ВОЛОГОСТІ DHT11, DHT22 (AM2302), ДАВАЧІ ВИМІРЮВАННЯ РІВНІВ СО2: МН-Z19, MG-811 ТА SENSEAIR S8



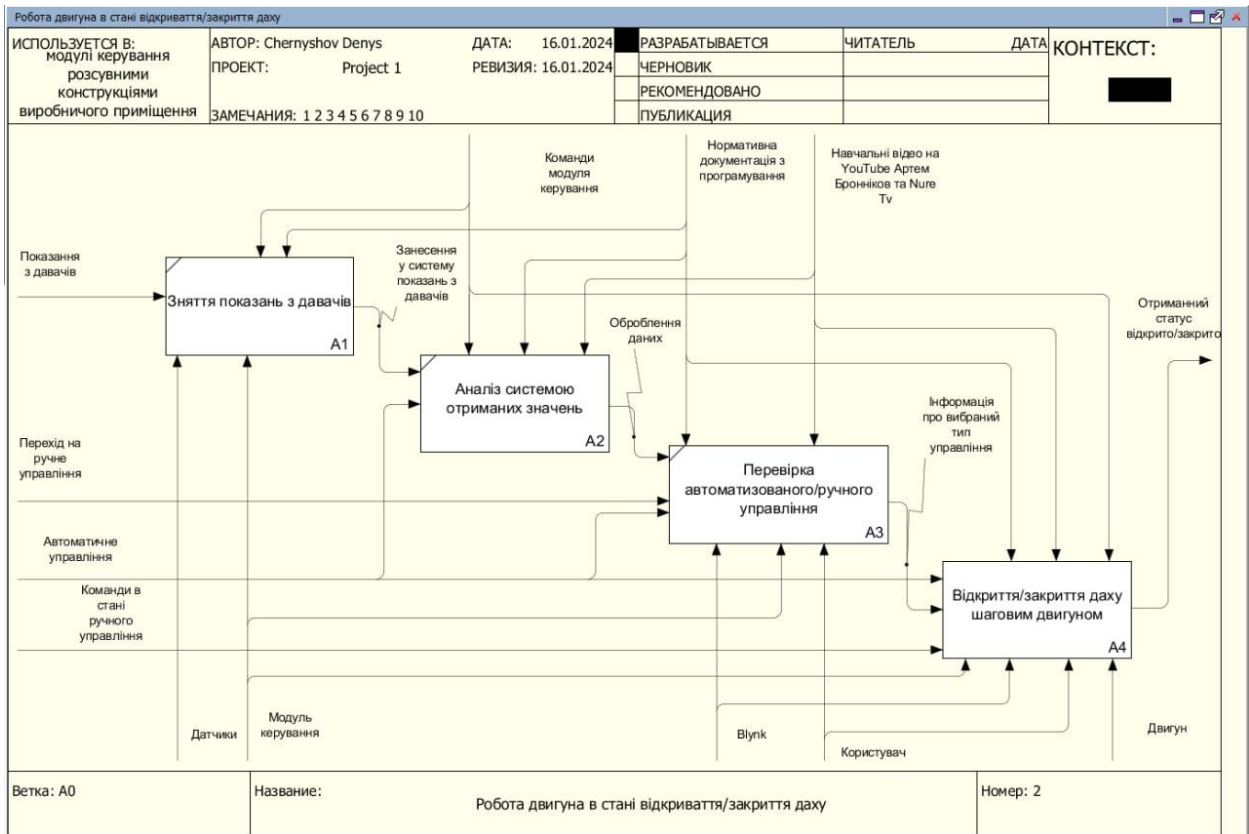
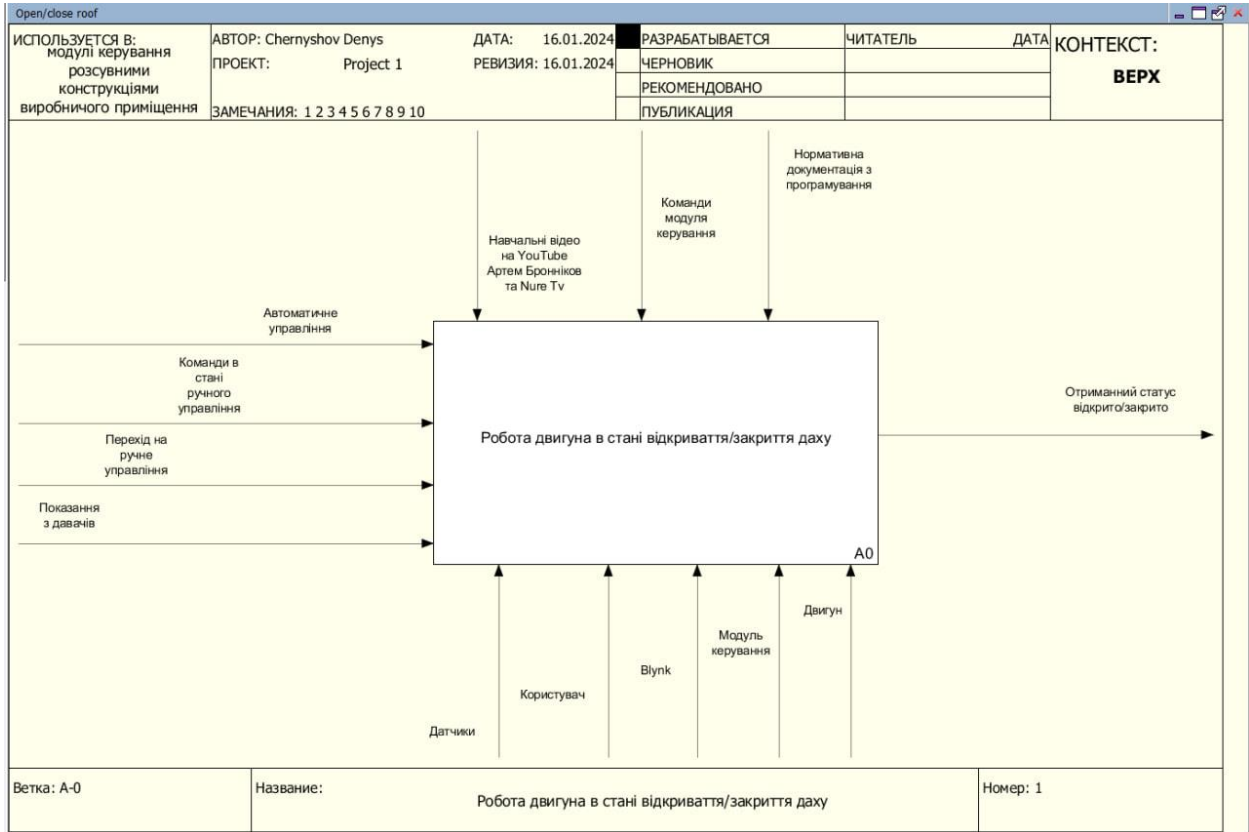
## КРОКОВІ ДВИГУНИ NEMA 17, NEMA 23 ТА NEMA 34

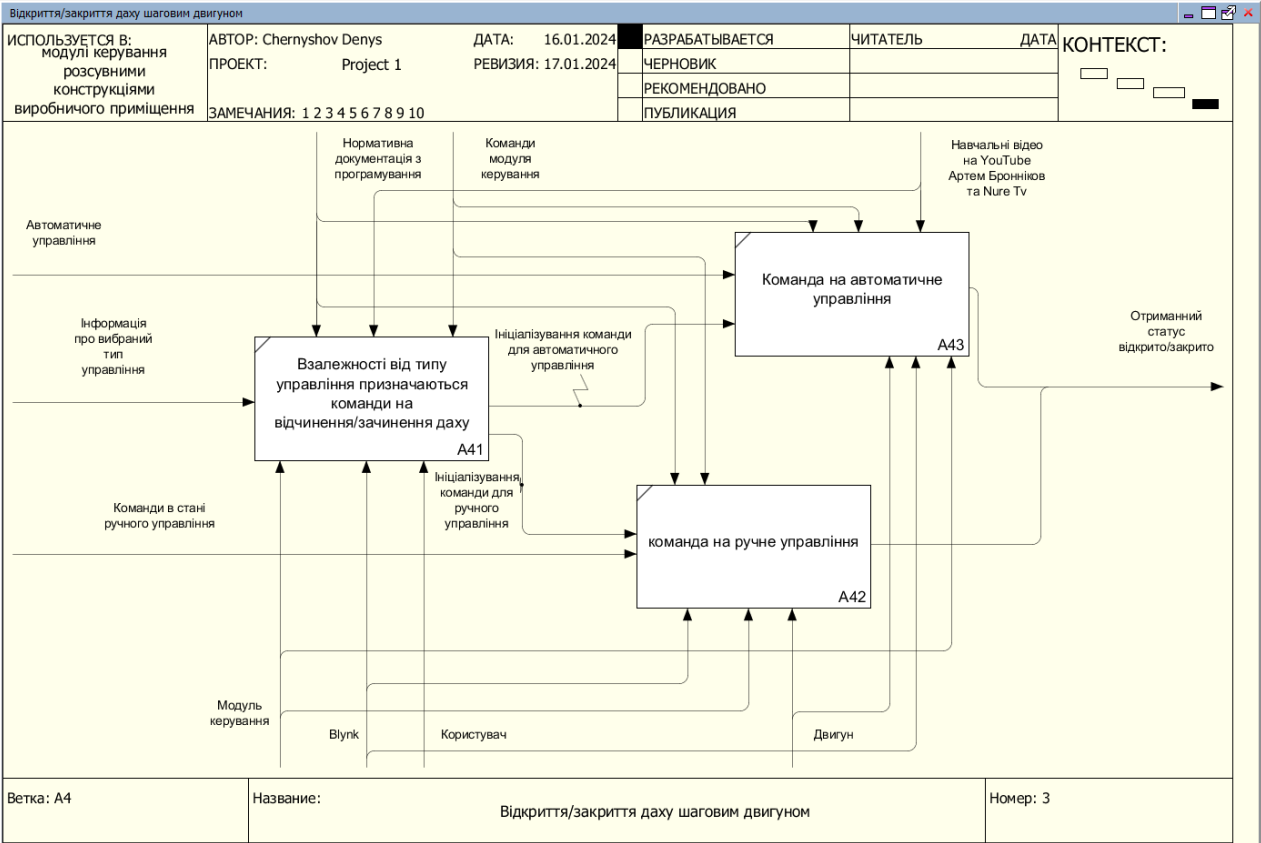


# ДІАГРАМА ПРЕЦЕДЕНТІВ

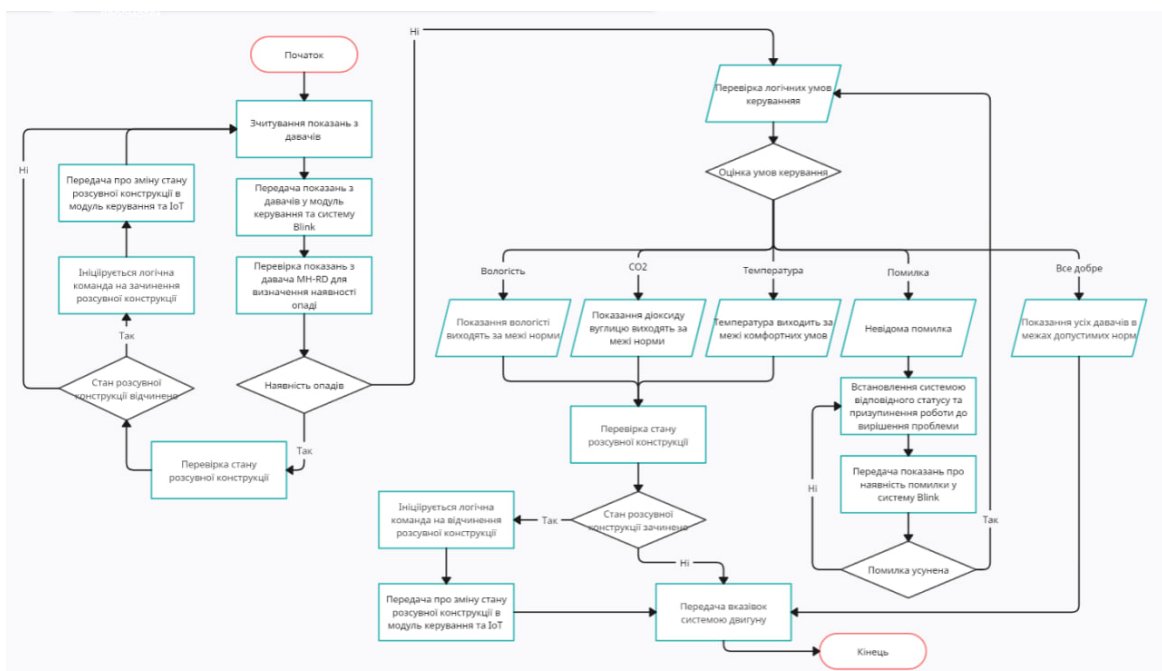
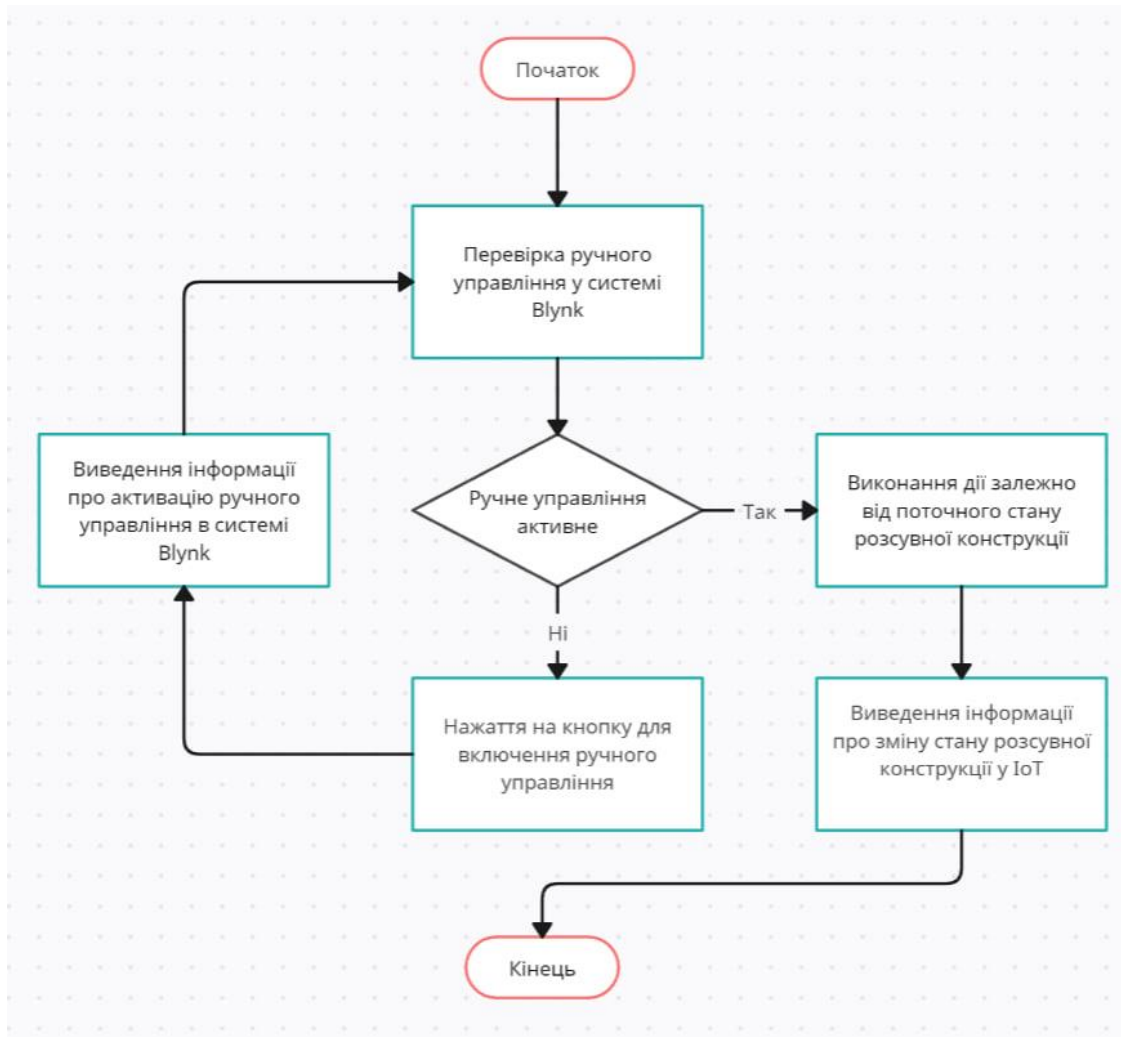


# КОНТЕКСТНА ДІАГРАМА IDF0

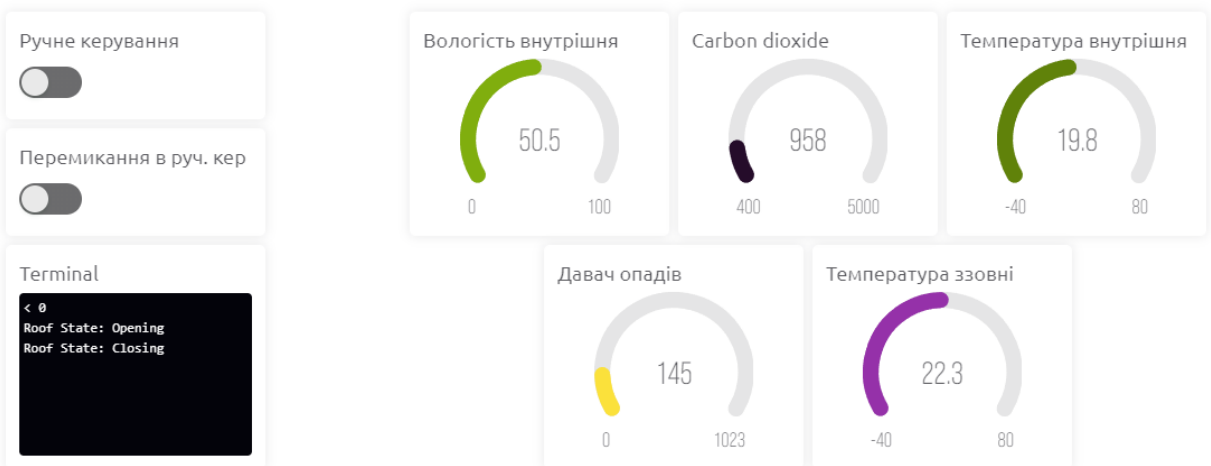
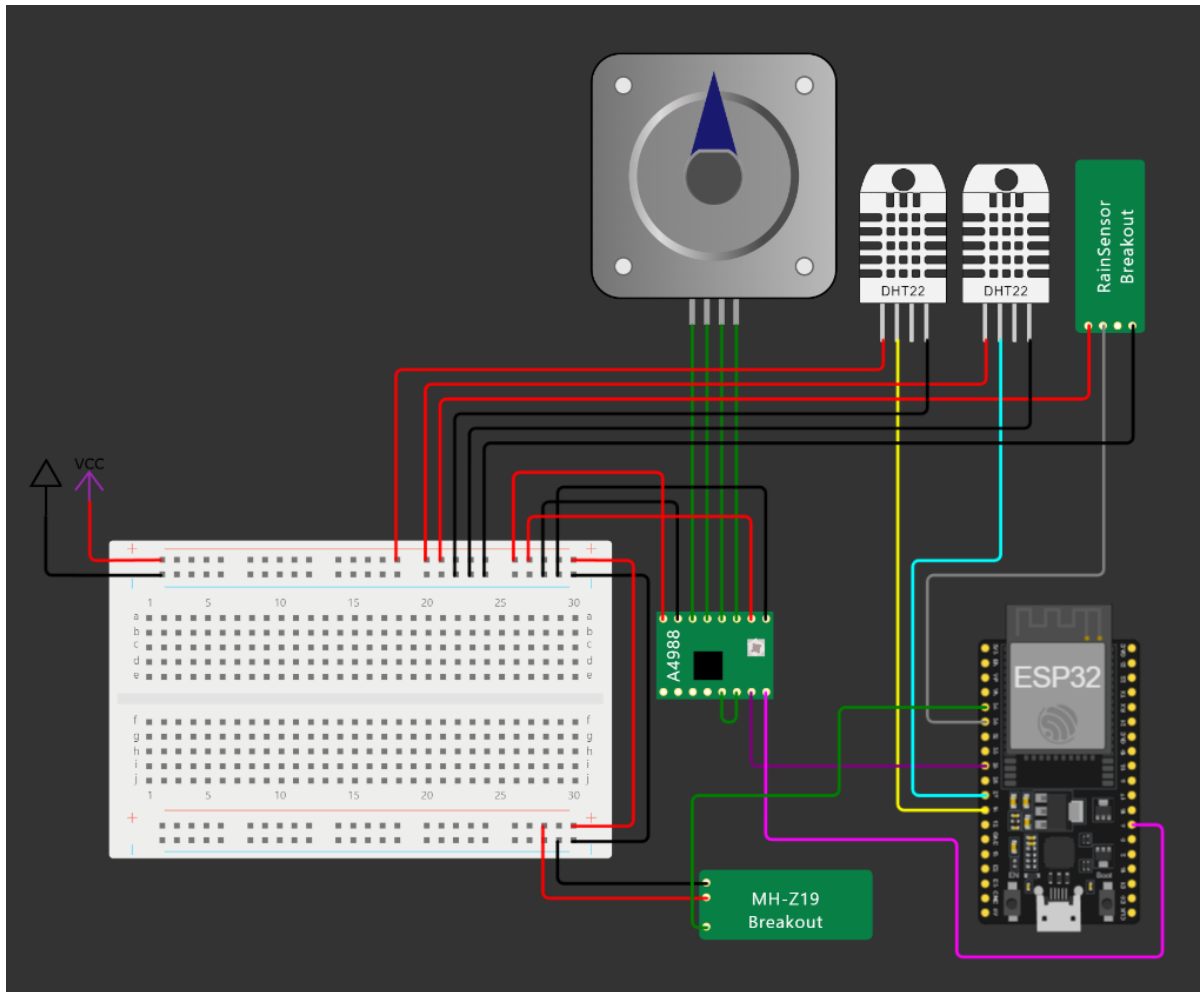


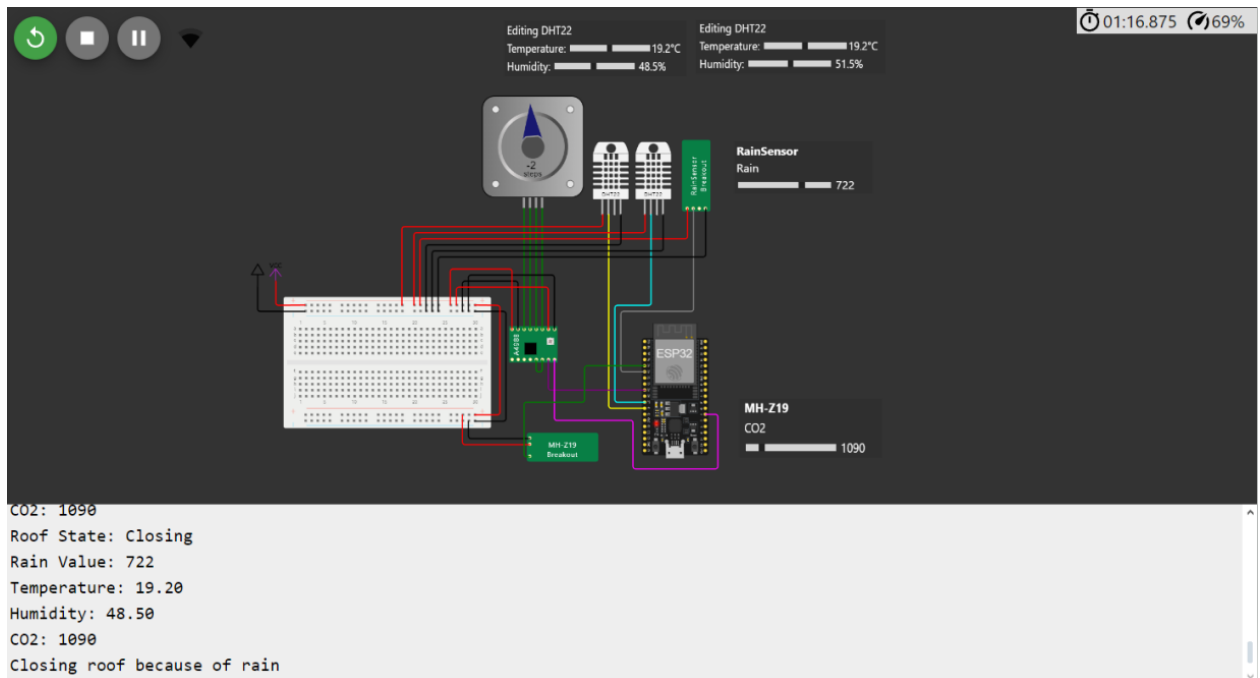


# БЛОК-СХЕМА АЛГОРИТМУ РОБОТИ МОДУЛЯ КЕРУВАННЯ



# МОДЕЛЬ МОДУЛЯ КЕРУВАННЯ ТА ВІДОБРАЖЕННЯ У BLYNK





Project1 Online ...  
 Denys My organization - 2756AQ  
 Add Tag

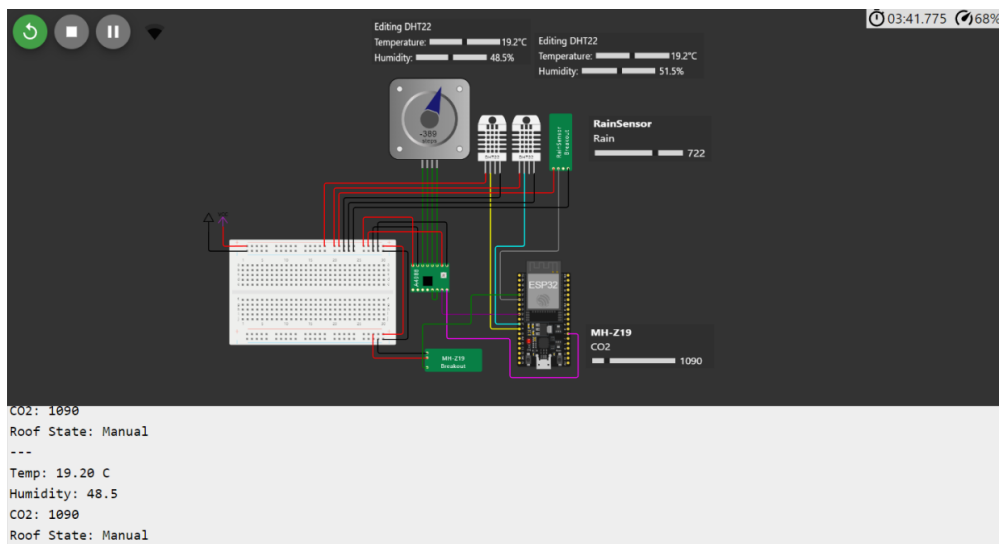
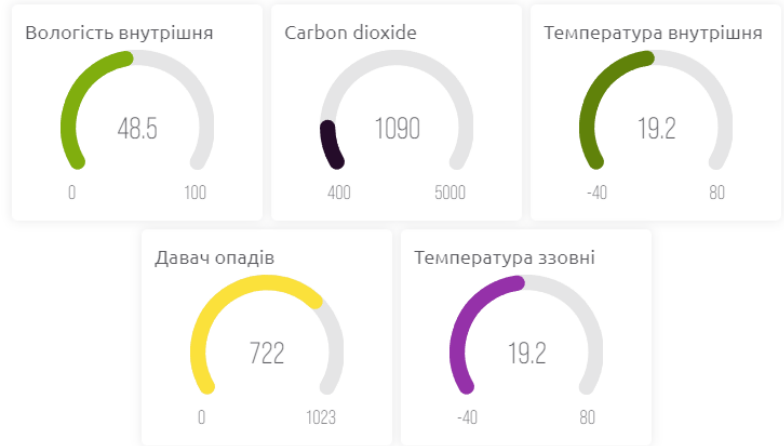
Dashboard Timeline Device Info Metadata Actions Log Datastreams

Latest Last Hour 6 Hours 1 Day 1 Week 1 Month 3 Months 6 Months 1 Year

Ручне керування

Перемикання в руч. кер

Terminal  
 < Roof State: Opening  
 Roof State: Closing



Project1 Online ...  
 Denys My organization - 2756AQ  
 Add Tag

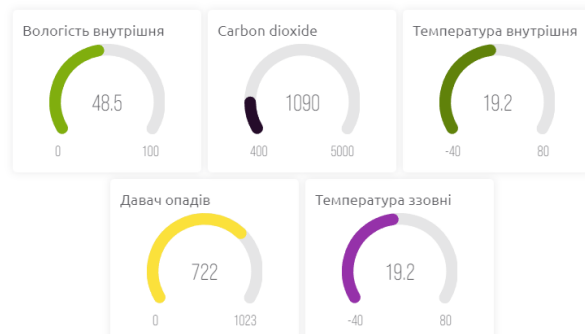
Dashboard Timeline Device Info Metadata Actions Log Datastreams

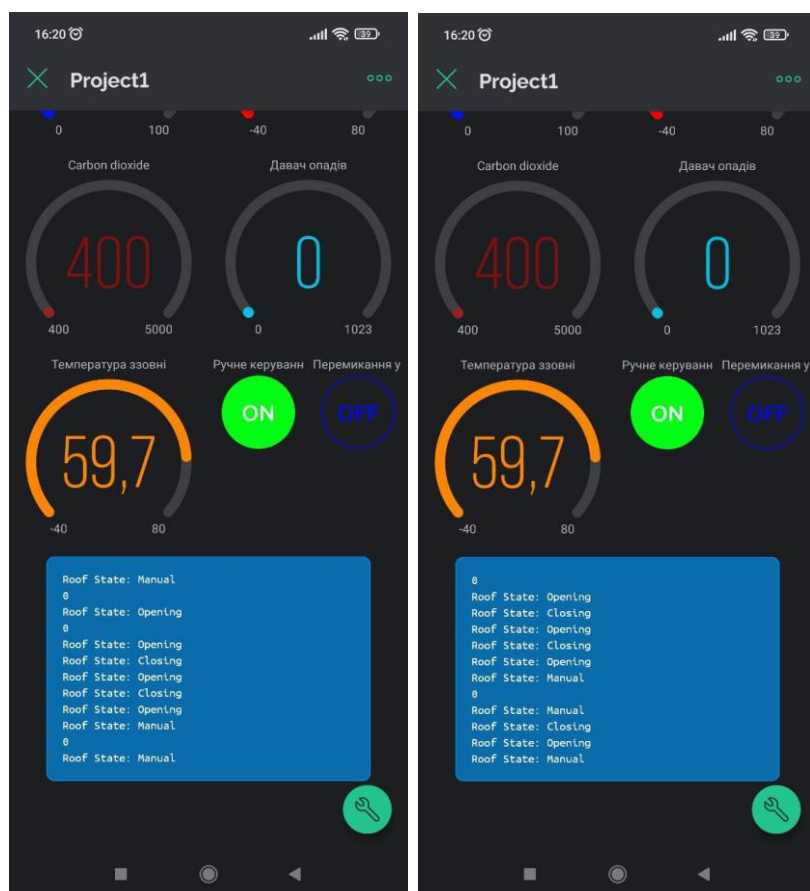
Latest Last Hour 6 Hours 1 Day 1 Week 1 Month 3 Months 6 Months 1 Year

Ручне керування

Перемикання в руч. кер

Terminal  
 < Roof State: Opening  
 Roof State: Closing  
 Roof State: Opening  
 Roof State: Manual





```

CO2: 1052
Roof State: Manual
---
Temp: 19.80 C
Humidity: 48.5
CO2: 896
Roof State: Manual

```

