Automated Smart Garden System

Students: Ayaulym Akimbetova (S5572470)
Arailym Talgatkyzy (S5636248)
Zhangirkhan Tastemir (S5650251)

Introduction

Our IoT Garden Monitoring System aims to optimize the maintenance and health of a garden by continuously tracking key environmental parameters such as light levels, temperature, soil moisture, and motor status for irrigation or other actuations. The system architecture integrates multiple technologies, ensuring efficient data collection, transmission, storage, and retrieval

Next ⟶

The goals of project:

- Real-time Data Collection: Continuously gather data from various sensors placed in the garden.
- Efficient Data Transmission: Use MQTT protocol to transmit the sensor data to a central server reliably.
- Data Storage and Management: Store the collected data in a MongoDB database for easy access and analysis.
- User Accessibility: Provide a user-friendly interface for users to view and analyze the collected data.

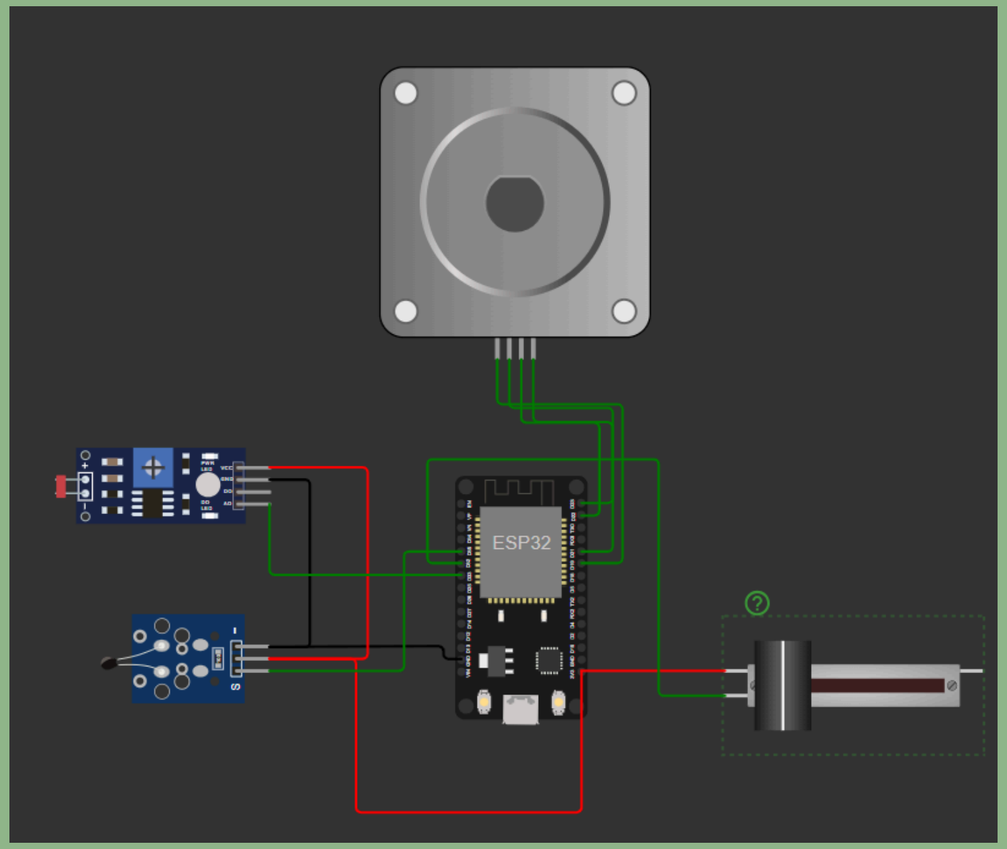
Next ⟶

Used libraries and frameworks

- ***Arduino*:** For programming and interfacing with the ESP32.
- ***Paho MQTT*:** For implementing the MQTT protocol in Python.
- ***Express: Node.js*** web application framework used for the backend server.
- ***Mongoose:*** MongoDB object modeling tool for Node.js.
- ***Ubidots:*** IoT platform for data visualization and analysis.
- ***MongoDB Atlas:*** Cloud database service used to store sensor data.
- ***MQTT:*** Lightweight messaging protocol used for communication between the sensors and the server.

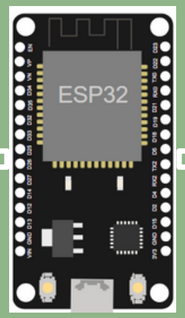
Hardware and sensors

01. ESP32 simulator
02. Photoresistor sensor
03. Temperature sensor
04. Soil moisture sensor
05. Motor for water

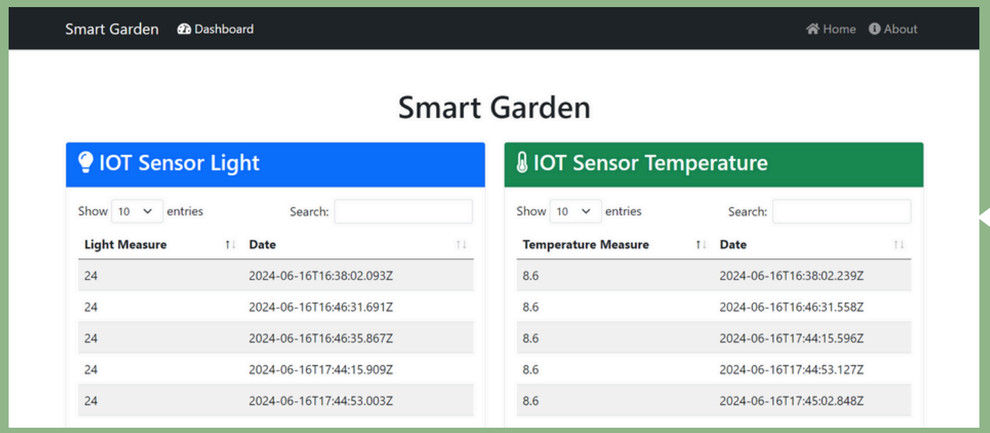


Next ⟶

Soil moisture sensor
Temperature sensor
Light sensor
Motor



MQTT Broker



Express JS

Next ⟶

MQTT Protocol: Used for efficient and lightweight data transmission between devices.

```
const char* mqtt_server = "industrial.api.ubidots.com";
const int stepsPerRevolution = 200;
String prev_temp = "";
String prev_ldr = "";
String prev_soil = "";
#define mqtt_port 1883
#define MQTT_USER "BBUS-vLMeqqA1T5zMXudRX0pJGri7nxc7rx"
#define MQTT_PASSWORD ""
```

```
MQTT_HOST = 'industrial.api.ubidots.com'
USERNAME = 'BBUS-ms7rdltV7E5peUZnLuwu7a1BDyoTIK'
PASSWORD = ''
if MQTT_HOST == None:
    raise ValueError('No MQTT Broker provided. Will exit.')
    exit(-1)
```

Next ⟶

- **For data processing and storage we used MongoDB.**

The MQTT subscriber processes incoming data and prepares it for storage.

```
const IotSensorSchema = new Schema({
    _id: ObjectId,
    truck: String,
    bktSize: Number,
    m: Array,
    max_ts: Date
});
```

```
mongo_client = pymongo.MongoClient(MONGO_URI, tlsCAFile=ca)
db = mongo_client.test
status_collection = db.iot_sensors
```

Next ⟶

Thank You!