Arduino Based IoT Gardening System

BY

JOEY LOK JO WEI

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) COMPUTER ENGINEERING

Faculty of Information and Communication Technology

(Kampar Campus)

June 2024

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Arduino Based IoT Gardening System

Academic Session: 202406

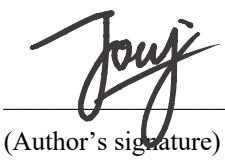
I JOEY LOK JO WEI

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,



(Author's signature)

(Supervisor's signature)

Address:
45, Lebuh Bercham Selatan 1/8,
Taman Kinta Mas,
31400 Ipoh Perak.

DR. MUHAMMAD SYAIFUL AMRI BIN SUHAIMI
(Supervisor's name)

Date: 9 September 2024

Date: 9 September 2024

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 9 September 2024

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ***JOEY LOK JO WEI*** (ID No: **20ACB05187**) has completed this final year project entitled Arduino Based IoT Gardening System" under the supervision of Dr Muhammad Syaiful Amri Bin Suhaimi (Supervisor) from the Department of Computer and Communication Technology, Faculty of Information and Communication Technology.

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

JOEY LOK JO WEI

DECLARATION OF ORIGINALITY

I declare that this report entitled "**Arduino Based IoT Gardening System**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : 

Name : JOEY LOK JO WEI

Date : 9 September 2024

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude and appreciation to the Almighty God, for the abundant blessings bestowed upon me during my studies. Furthermore, I would like to extend my heartfelt gratitude and admiration to my final year project supervisor, Dr. Muhammad Syaiful Amri bin Suhaimi, Assistant Professor at Universiti Tunku Abdul Rahman, for supervising me and giving me the opportunity to handle an Arduino based project. Once again, thanks for your guidance throughout the final year project I and II.

In addition, I would want to express my gratitude to my course mates and friends for their extraordinary patience, unwavering support, and unwavering companionship throughout challenging circumstances. Lastly, I would want to express my gratitude to my parents and family for their support, and consistent encouragement during the duration of my study. Despite encountering numerous inquiries during my study, they are always supportive and patience.

ABSTRACT

This project relates to the trend of technology nowadays, the Internet of Things (IoT). The project title “Arduino Based IoT Gardening System” targets to utilize the IoT technology to improve gardening experience. By using Arduino and IoT technologies, an automated gardening solution is created to enable users to remotely oversee the environmental conditions of their garden. This project will utilize varieties of sensors to collect environmental data such as soil moisture sensor, temperature and humidity sensor, light intensity sensor, and the air quality sensor. The data collected will be processed and send to the users’ devices in graph form.

The Arduino Based IoT Gardening System introduces a few innovative functions such as Automated Irrigation, Collection of Environmental Data and Notifications. The objective of the project is to improve plant care efficiency and to utilize the resources efficiently. This project emphasis on simplicity, allowing amateur or expert to use it easily to improve their gardening experience with technologies.

This project mainly focusses on increasing the sustainability and utilizing the resources efficiently compared to the traditional gardening methods. On the other side, the use of Internet of Things (IoT) helps user to keep track on the environmental data remotely without having to physically observe their garden. This project provides beneficial contribution to the domains of automated gardening and smart gardens.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

°C	Degree Celsius

LIST OF ABBREVIATIONS

IDE	Integrated Development Environment
IOT	Internet of Things
LDR	Light Dependent Resistor
LED	Light Emitting Diode

Chapter 1

Introduction

The background and objective of the project titled "Arduino Based IoT Gardening" will be discussed in this chapter. Besides, any research related to the topic stated above as well as the thesis related to this project will be defined.

1.1 Problem Statement and Motivation

In this modern era, gardening activities can be seen as decreasing across the years, most of the reasons are lack of time, not interesting or bored in the traditional way of gardening. Traditional gardening methods mostly rely on human monitoring and manual care of the plants, which is quite troublesome and might get bored after some time. Furthermore, most of the individuals are separated from their plants because of outstation jobs or often travelling, lack of data of their plant surroundings and lack of care of their plant are not ideal.

The motivation of this project "Arduino Based IoT Gardening System" arises from the needs to overcome few constraints with traditional gardening method such as watering manually and providing enough light for the plant manually. Other than that, breaking the limit of remote users to take care and know their plants surroundings data motivates this project.

1.2 Objectives

In this project, the objectives are providing users with the opportunity to monitor their plant's surrounding remotely and users will not have to manually take care of their plants. With the help of IoT technologies, data of the plant's surroundings can be processed and sent to the user's device, providing useful data for users to keep track on their plant's surrounding. Besides that, automated by using the Arduino to handle the automation of gardening provides users more flexible time to do their jobs, automated system will take place according to the sensor readings, therefore effectively managing the resources such as water and energy.

1.3 Project Scope and Direction

The purpose of the project "Arduino Based IoT Gardening System" will be developing a system that can automatically observe the plant's surrounding information and provide automated plant care for the plants. Besides that, the collected data will be processed into graphical form to ease users to keep track on their plant's situation. To achieve this, both hardware and software are required.

The hardware part includes Arduino UNO, sensors, and irrigation. Besides, the software part includes Arduino IDE as well as Python and few of the libraries to help process the data and send them to the users through Telegram Application.

In short, this project aims to improve individual's interests towards automated gardening, helping amateur gardeners to improve their gardening experience and lastly to reimagine the future of gardening for the good of both human developments and our planet.

1.4 Contributions

This project, "Arduino Based IoT Gardening System" provides three main contributions by providing an advanced gardening system, improving the utilization of resources efficiently, and to enhance the user experience in gardening while promoting an ideal environment condition that can promote plant development. By integrating the hardware components and software technologies, all the automations will take place based on the sensor's readings, ensuring the needs of certain resources before providing them, collected data will be delivered to the users after processing, making sure the users are acknowledged with the plant surroundings.

1.5 Report Organization

The following chapters will explore deeper into the project "Arduino Based IoT Gardening System" with detailed explanations. Chapter 1 explained the general project information and purposes, Chapter 2 will be reviewing related topics. Chapter 3 will be explaining the plans and how the flow should go. In Chapter 4, things such as how it works, how to make it work and the design will be discussed. Reaching Chapter 5, more of the setups and operations will be shown here. In Chapter 6, testing will be carried out and results are to be evaluated here. Lastly is the Chapter 7, conclusion and recommendations, this chapter will conclude the project and recommendations are given for further improvements.

Chapter 2

Literature Review

2.1 The influence of humidity for plant growth forecasting.

The journal article [1], written by S.Y Chia and M. W. Lim, according to their research based on scientific basis and theories, they summarized the relationship between relative humidity (RH) and lettuce growth. The journal suggests that increasing the RH will give positive impact towards the physical growth of lettuce. Physical growth here means the length of the leaf, width of the leaf and the dry weight of the leaf.

Growth Responses	Relative Humidity (% RH)	
	50±2	85±2
Leaf number (quantity)	20	20
Leaf length (cm)	9.20	10.70
Leaf width (cm)	7.45	8.85
Length/width ratio	1.23	1.20
Plant fresh weight (g/plant)	12.06	20.94
Plant dry weight (g/plant)	0.83	1.30
Growth rate (r/day)	1.28	1.31

Figure 1 Results done by the article [1]

According to the data and experiments conducted in the article, increased RH has improved the lettuce leaf length about 10±3%, leaf width approximately 2±1% and the aerial dry weight increased about 28±2%. By concluding the data conducted, they stated that 85±2 is the most optimal relative humidity level to improve lettuce growth.

This article had done a great job in proving the effects of RH towards the growth of lettuce, effectively summarized the relationship between RH and lettuce growth. By the evidence shown in the article, its clear that higher RH levels are beneficial, specifically around 85±2%.

2.2 Similar project done by other researchers.

By reviewing [2], written by V. Kumar. P, K. C. Ramya, Abishek. J.S, Arundhathy. T.S, Bhavvya. B and Gayathri. V, it can be observed that they separate their focuses nicely and clearly, for instance, the automation in agriculture, the sensor technologies & how those sensors work, remote monitoring and control, resource conservation and they even included the future directions in smart agriculture.

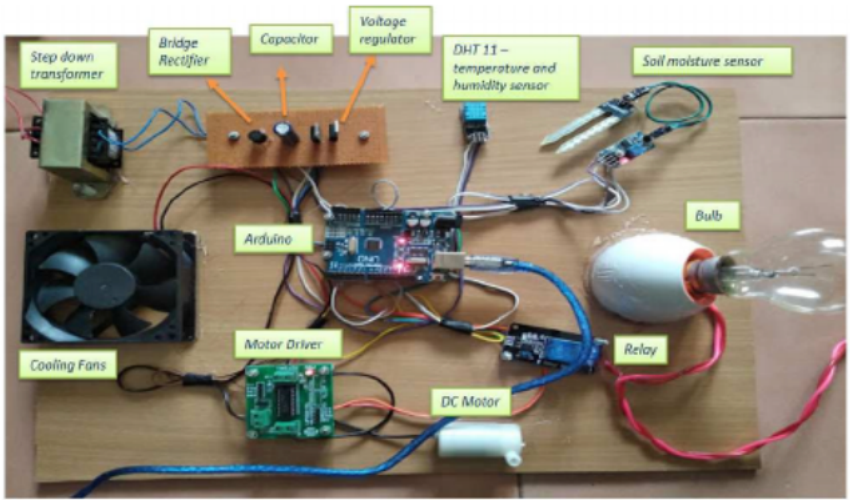


Figure 2 Setup of system in [2]

S.No	Equipment	Specifications
1	Arduino	UNO(5V)
2	Temperature and Humidity Sensor	DHT11 (3.5V-5.5V)
3	Soil Moisture Sensor	SM100 (3.5-5.5V)
4	Motor Driver	LM293D (4.5V-36V)
5	Electric Bulb	10W
6	Relay	-
7	DC Motor	12V - 1A

Figure 3 Specification of equipment shown in [2]

Referring to their proposed system, components and specifications shown above, I think they are doing a simple demonstration only, because they are using the DHT11 instead of DHT22 (a better sensitivity temperature and humidity sensor), and using a simple 10W bulb, which technically does not benefit the plants. But anyway, their explanations are clear and understandable, where users can modify and recreate the project. Besides that, there's a major issue found in this paper, they mentioned the observation of data from an android application, but however, the software and hardware part in implementing them are not mentioned in the paper, even the image of the set up does not contain any modules or component that enables that function.

2.3 IoT- Based Smart Gardening System

Referring to the article [3] titled IoT Based Smart Gardening System, the authors provided a well detailed report about their project in smart gardening, with many visual aids and explanations on the components and their functionality. According to the methodology provided, they are using NodeMCU as the central unit to coordinate the sensors and actuators. By using the NodeMCU, the IoT is made easy to be implemented by using Cloud platforms, in this paper, they are making use of the Blynk IoT platform.

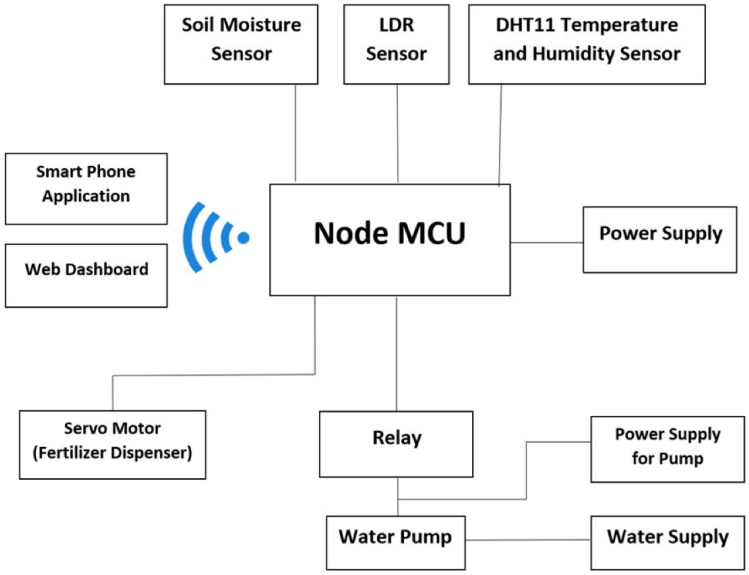


Figure 4 Block Diagram provided in [3]

By referring to the system architecture design shown above, they're implementing three sensors, soil moisture sensor, LDR sensor, and DHT11 sensor. Most research uses DHT11 might be the reason that gardening may not require high sensitivity sensor and it is more cost effective. Other than that, they included a Servo Motor which act as the fertilizer dispenser which is a good point here where some plants require more nutrients to bloom or grow physically.

2.4 Effect of Illumination Towards Plant Growth

This article [4] is being reviewed to understand the actual need of a plant, especially the light intensity required by a plant and type of illumination required. Based on the research of the article stated, LED lights are generally proven beneficial for plant growth, especially in controlled environments like greenhouses or indoor.

After reading through the article, there are number of points pointing towards the advantages of using LED lights for plantations, namely energy efficiency, spectrum customization (combining few colors of LEDs to provide a specific wavelengths of light), durability and minimal heat output. For the specific wavelengths for plant growth, the most effective combination stated in the article is Red from 640-720nm, as Red is important for flowering and fruiting; Blue from 425-490nm, that improves vegetative growth and chlorophyll production; and lastly Green from 490-560nm, allowing it to penetrate deeper into the plant canopy. LED would be a great choice as it has longer lifespan, simple maintenance and it reduce less heat, lowering the chances of plants getting heat stress in a closed area.

According to my understanding, this article tries to express the benefits of LED lights towards plant growth, but however, the research does not stop there, as there’re many types of situations where different plants and environment will require different cares, therefore ongoing research and development are crucial to optimize and fully realize their potential in the agriculture field.

2.5 Importance of Soil Moisture for Plant Growth

To understand the importance of soil moisture for plants, an article [5] is studied. Referring to the article, experiment is conducted using commercial soil hygrometer sensor, which is an advanced soil moisture sensor to detect the soil moisture and DHT11 sensor to measure the temperature and humidity. No control elements for temperature and humidity as it only focusses on the effect of soil moisture.

Week	Average Soil Moisture (%)	Stem Diameter (cm)			
		Stem 1	Stem 2	Stem 3	Stem 4
1	30	0.8	0.9	0.7	1
2	60	1.1	1	0.9	1.2
3	90	1.6	1.4	1.3	1.5

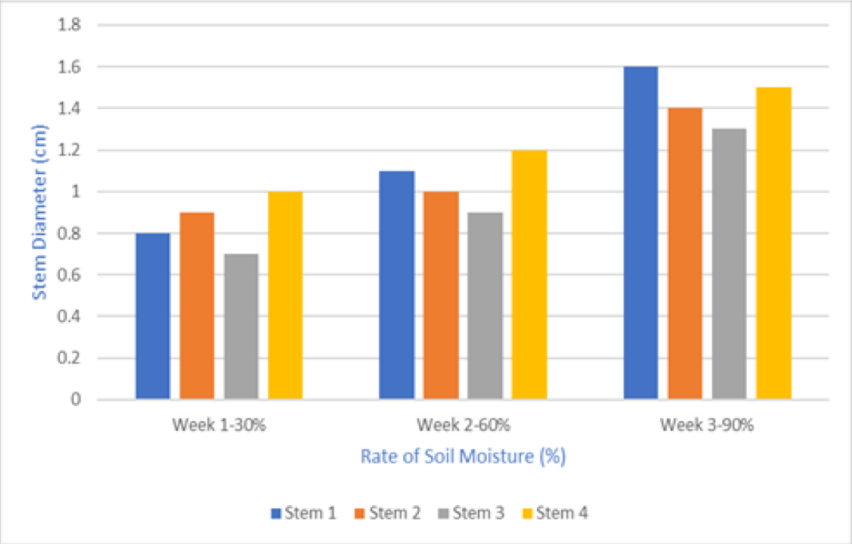


Figure 5 Results of experiment provided in [5]

Next, the soil moisture readings were taken at different stages to show the different volumes of water affects the soil moisture. Then continuing the experiment by measuring the growth of the plant by measuring the stem diameter. Result shown in the figure above shows that higher soil moisture levels improve the diameter of the plant stem, proving that suitable soil moisture is crucial for plant growth.

The article concluded the experiment such that the experiment conducted was successful as the soil moisture, temperature and humidity are monitored and soil moisture is proven to be crucial in plant growth, but excessive watering will lead to water waste and future improvements such as adding a control element like a fan to control the temperature and humidity can be implemented if necessary.

Chapter 3

System Methodology/Approach OR System Model

3.1 System Design Diagram

3.1.1 System Architecture Diagram

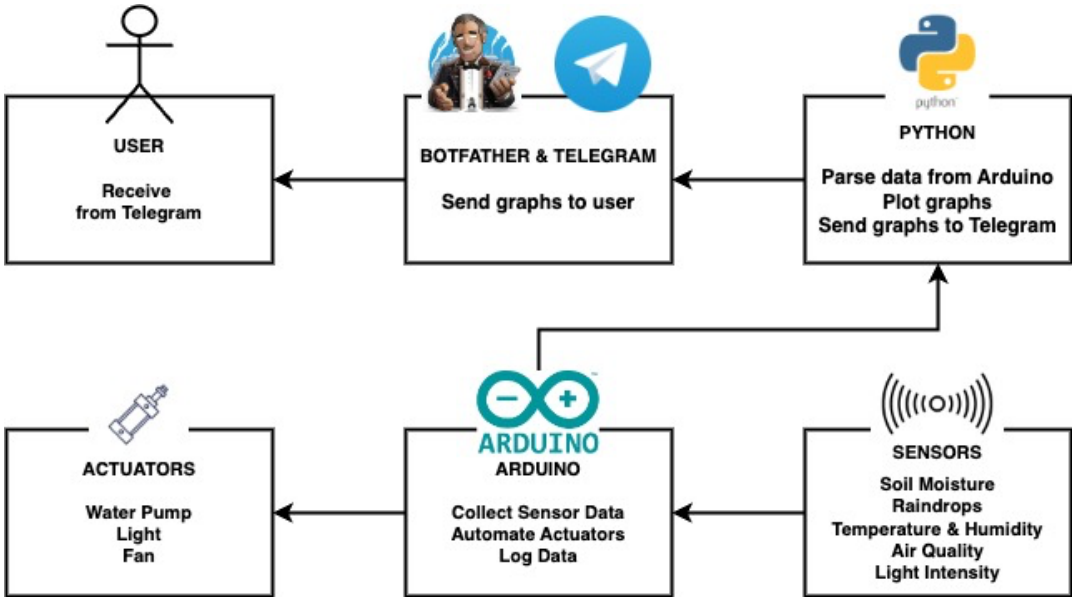


Figure 6 System Architecture Diagram for the System

Above is the system architecture diagram of this project, Arduino will be acting as the main character here as it will collect sensor data from sensors and automate the actuators based on the sensor readings, then readings are then logged to be parsed by Python program and soon be plotted into graphs. Finally, the graphs are to be sent to the users via Telegram platform by using BotFather, which basically allows users to create and modify their own bot and functionality, then the bots will send the graphs to the users for observation.

3.1.2 Use Case Diagram and Description

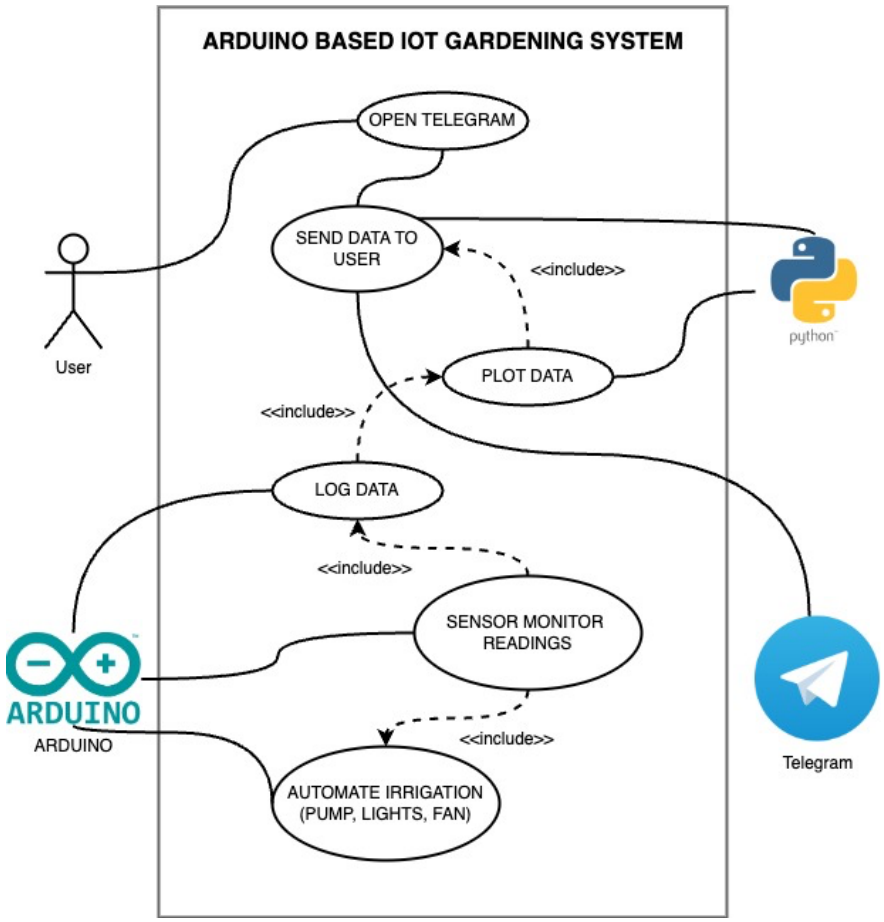


Figure 7 Use case diagram for the system

Based on the diagram shown above, there are four actors: User, Python, Telegram and Arduino. The user will be the end user that monitor the graphs received in telegram. Then, Python will be acting as a script that processes the data parsed from Arduino to generate graphs and sends them to the bot in Telegram. Telegram is a social media application that provides messaging services, in this case it is used to send updates to the user. Lastly the Arduino will be the microcontroller acting as the center of the system, controlling the system traffic.

After that, there are six use cases in the use case diagram shown: sensor monitor readings, automate irrigation, log data, plot data, send data to user and open telegram. First, sensor monitor readings to be read by Arduino, then Arduino activates the actuators which are the fan, light and water pump based on the reading's threshold. Then, the readings are logged and plotted into graphs by Python Scripts, then the graphs are to be sent to the Telegram Bot and displayed to the user when user open the Telegram to check for the graphs.

3.1.3 Activity Diagram

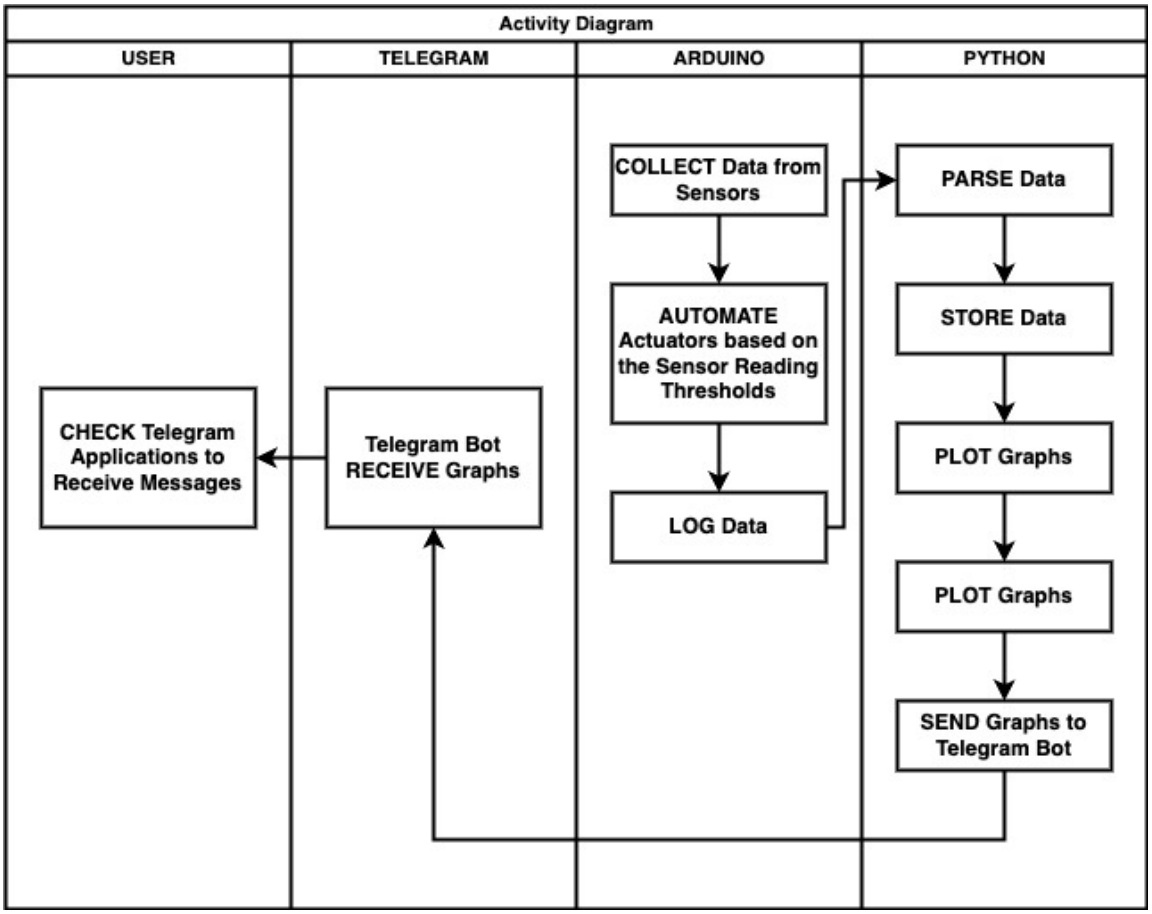


Figure 8 Activity Diagram of the system

Referring to the activity diagram above, Arduino will start initiate all the sensors and actuators, then collect the data from the sensors, comparing the readings to the predefined thresholds to take required actions such as turning on the LED, then the readings will be logged and parsed by the Python Scripts, the script will first store the data into a text file, when data are sufficient, graphs will be plotted and being sent to the Telegram Bot, then the graphs are delivered to the user to observe.

The steps are set to loop, unless the user end the phyton script, the readings are being collected and observed by Arduino every 4 hours. Then the python script will plot the graphs when 6 complete readings are stored, and soon send the graphs to the Telegram Bot to be displayed to the user.

The activity diagram flow emphasizes on continuous monitoring, automated irrigation, data logging and parsing and lastly user notifications.

Chapter 4

System Design

4.1 System Block Diagram

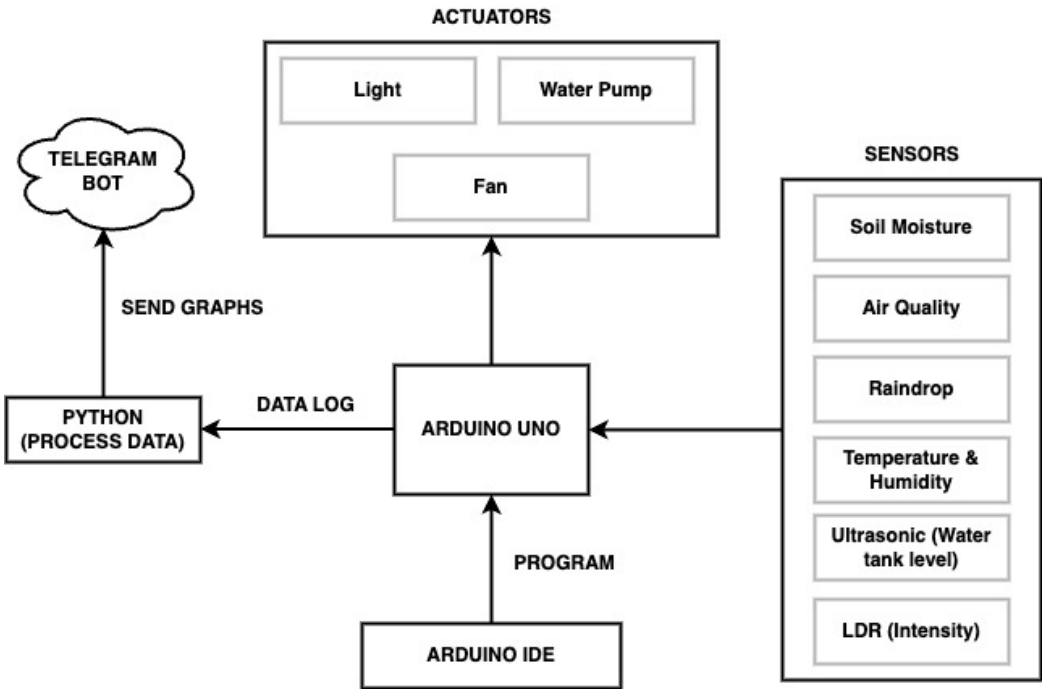


Figure 9 System Block Diagram of the system

This project can be broken into two main parts, the hardware and software part. Hardware part will be consisting of the Sensors unit, Actuators Unit and the Center, Arduino UNO. On the other hand, the software consists of the Arduino IDE, Python and the Telegram Bot, which also can be considered as the cloud service in this case.

Sensors unit has multiple sensors, namely the soil moisture sensor to read soil moisture level, air quality sensor to detect the quality of the surrounding air, raindrop sensor to detect rain, DHT22 sensor to collect data of the temperature and humidity, ultrasonic sensor to detect the water level remaining and lastly the light dependent resistor to read the light intensity. Actuators unit consist of light, the full spectrum LED that can benefit the plant growth, water pump to water the plant and fan to improve the air circulation.

Arduino IDE is used to program the Arduino UNO; Python script will act as the data processor to process the data into graphs and send them to the Telegram.

4.2 System Components Specifications

Table 1 Components Specifications

Components	**Specifications**
Arduino Board	Arduino UNO (5V)
Soil Moisture Sensor	FC-28 (3.3V)
Temperature and Humidity Sensor	DHT22 (3.3V)
Light Intensity Sensor	Light Dependent Resistor (3.3V)
Raindrops Sensor	Raindrop Sensor (3.3V)
Water Level Sensor	Ultrasonic Sensor (5V)
Air Quality Sensor	MQ-135 (3.3V)
Water Pump	DC Submersible Motor (5V1A)
Fan	Portable Fan (5V1A)
Light	Full spectrum LED (5V1A)
PC	MacOS
Power Supply	Socket Adapter (5V for Arduino, 3V for Actuators Relay)

The specifications of the components are shown in the table above, the specific component names are listed in the table too. The actuators such as water pump, fan and light will be connected to the Arduino UNO using a 4 channel Relay Board, this is to allow controls from the Arduino UNO to the individual actuators.

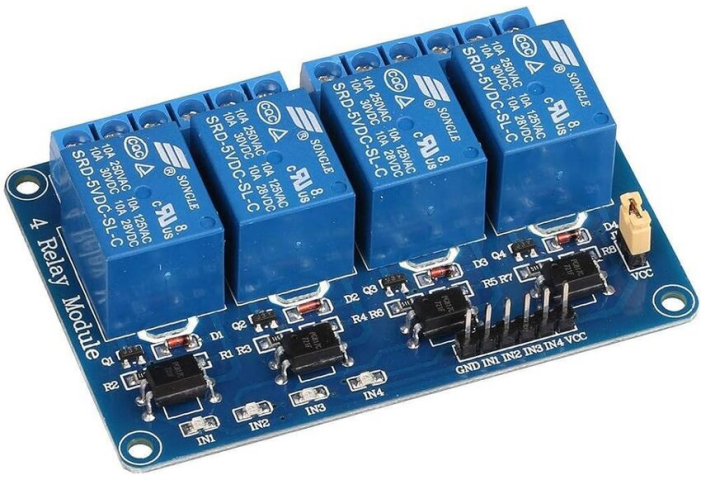


Figure 10 Four channel Relay board

4.3 Circuits and Component Design

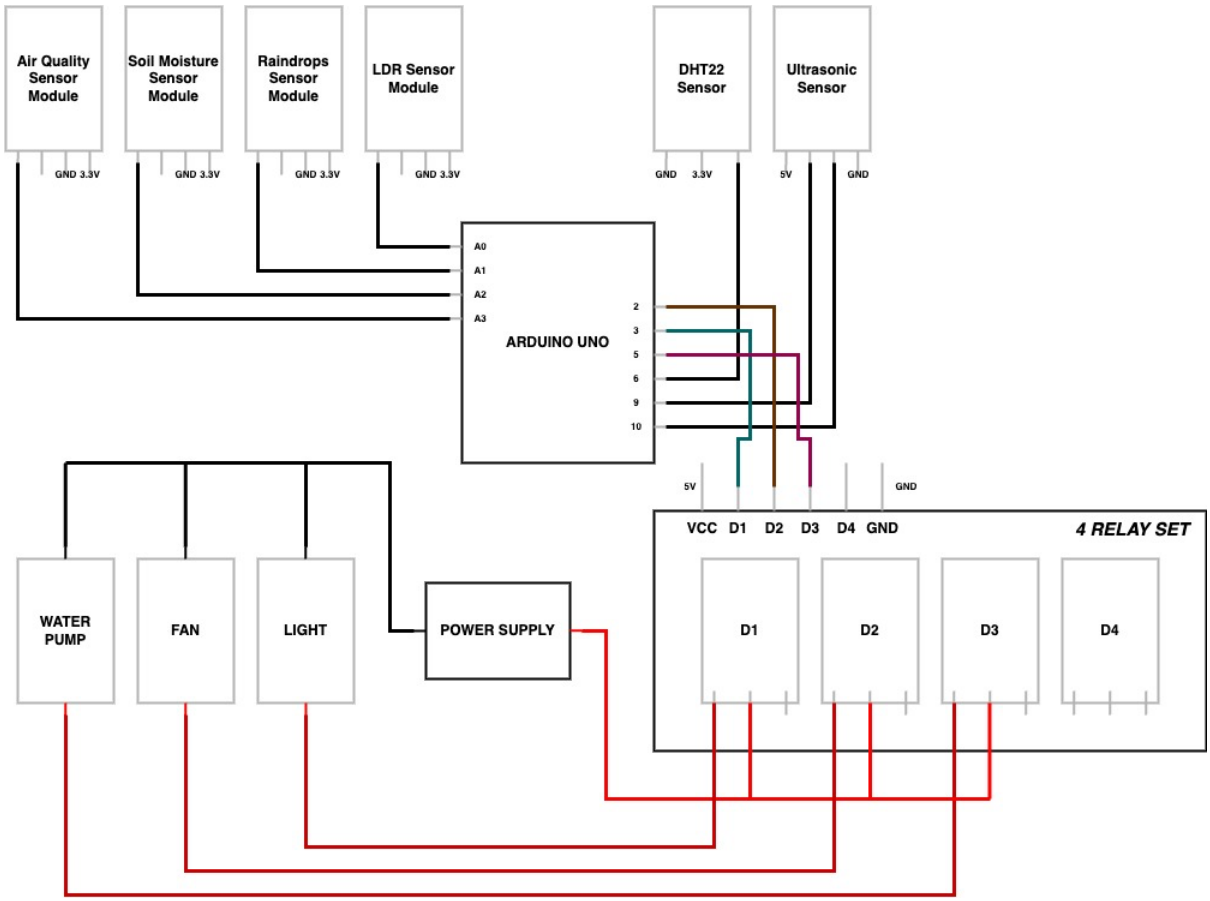


Figure 11 Schematic Diagram of the system built

Figure 11 is the circuits and component design in a schematic way, but the components are being set up with the help of a breadboard. According to the diagram, most of the sensors such as air quality sensor module, soil moisture sensor module, raindrops sensor module and LDR sensor module are connected to the Arduino UNO using the Analog Pins, while the DHT22 sensor and ultrasonic sensor are connected using Digital Pins. Then, three Digital Pins are set to output to send signals to the relay board to activate the actuators when needed. A power supply (5V1A) will be used to provide power to the actuators when needed to be activated.

4.4 System Components Interaction Operation

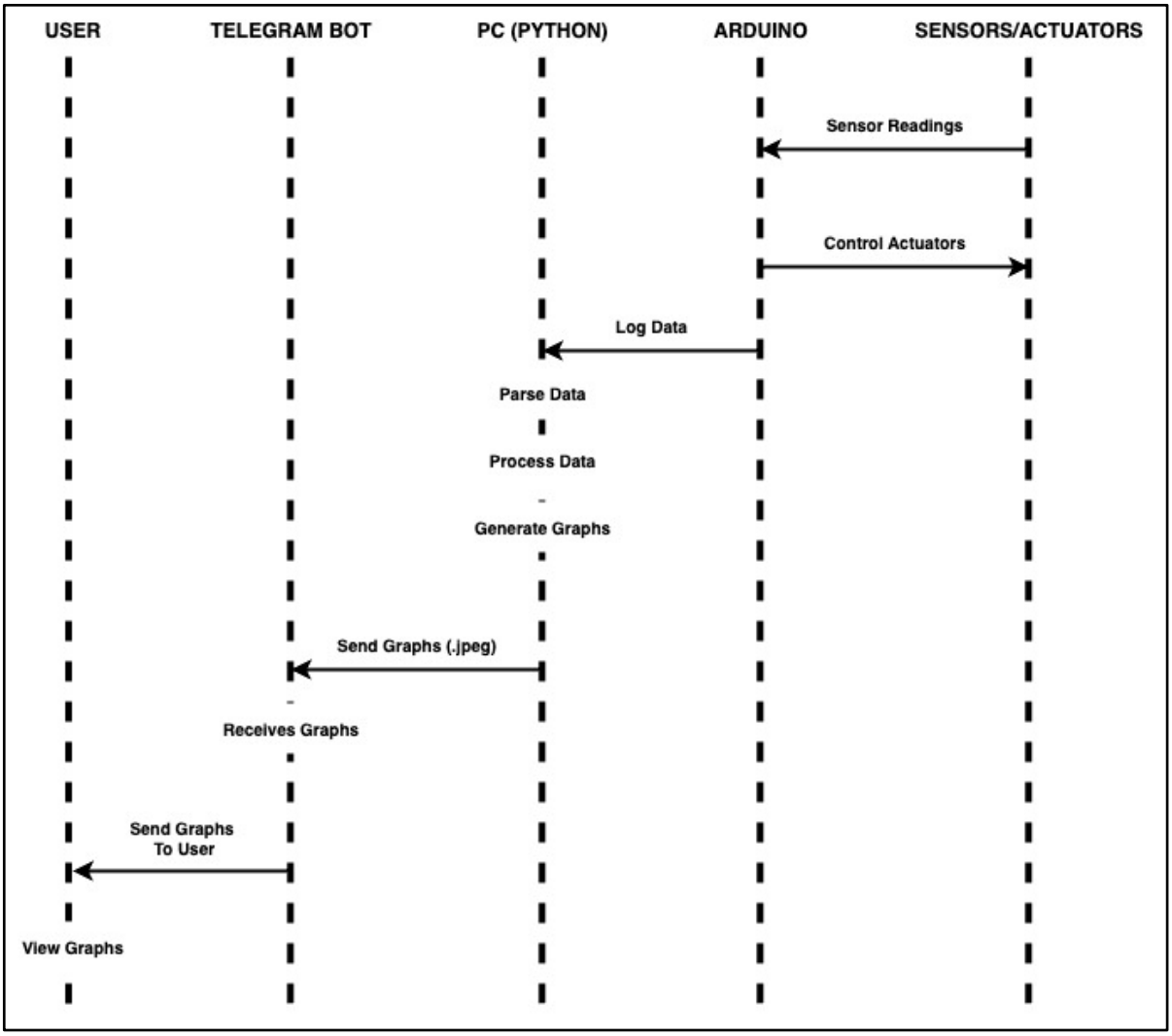


Figure 12 System Components Interaction Operation Diagram of the system

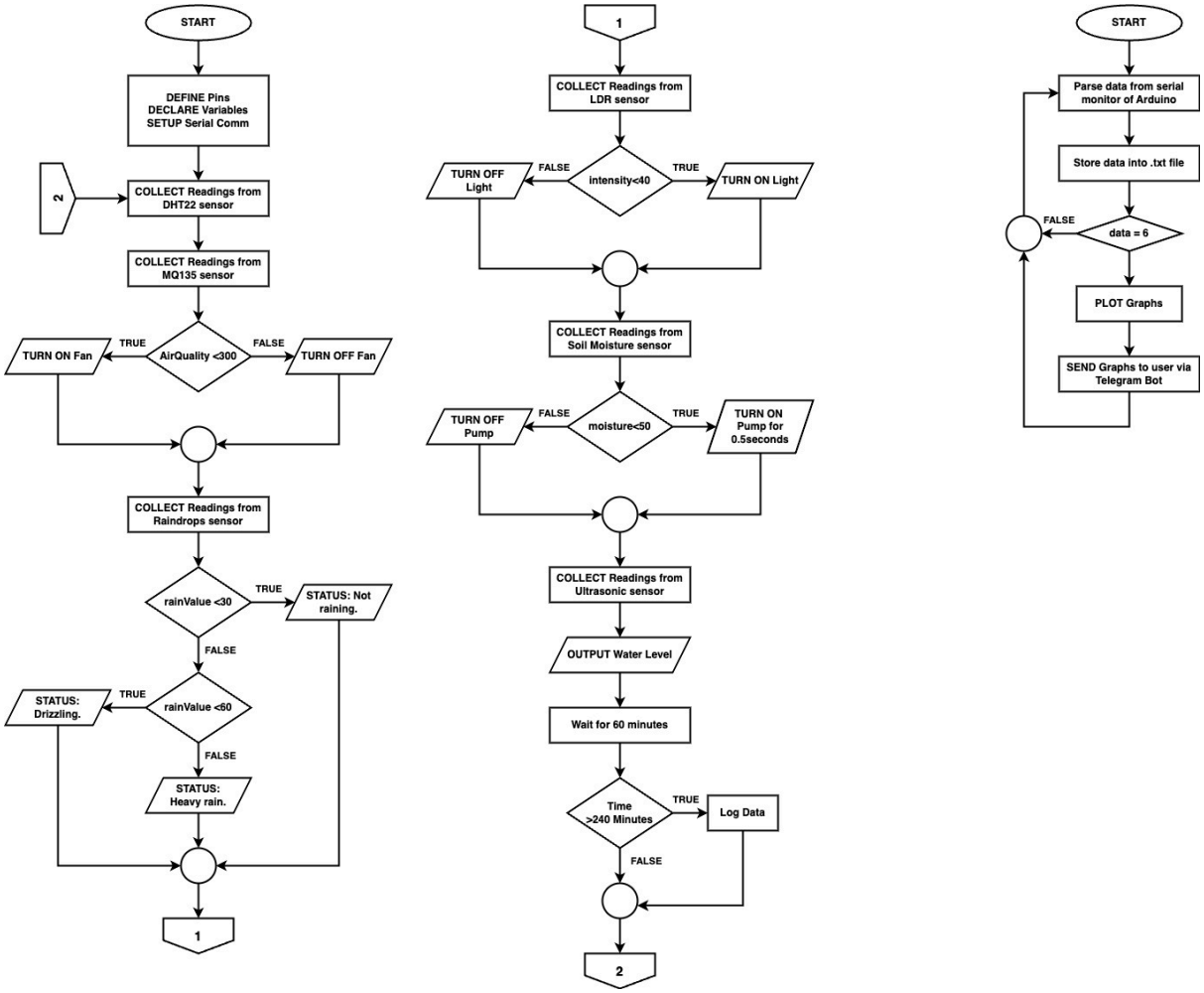


Figure 13 Flowchart of the System

Referring to the system components interaction operation diagram, the operation flows from the right to the left starting from sensors reading data and read by the Arduino UNO, then Arduino will control the actuators based on the predefined threshold of the readings. After that, the readings are being logged and parsed by a PC that has the python script running, the python script will then process the data into graphs illustrations. When the graphs are ready, they will be sent to the telegram bot and soon be sent to the users to be viewed. The explained process from the sensors/actuators to pc(python) will be looped every four hours, where else the process from pc(python) to user will only run when the previous activity run for six times (six times multiply four hours equals to twenty-four, one day).

Chapter 5

System Implementation

5.1 Hardware Setup

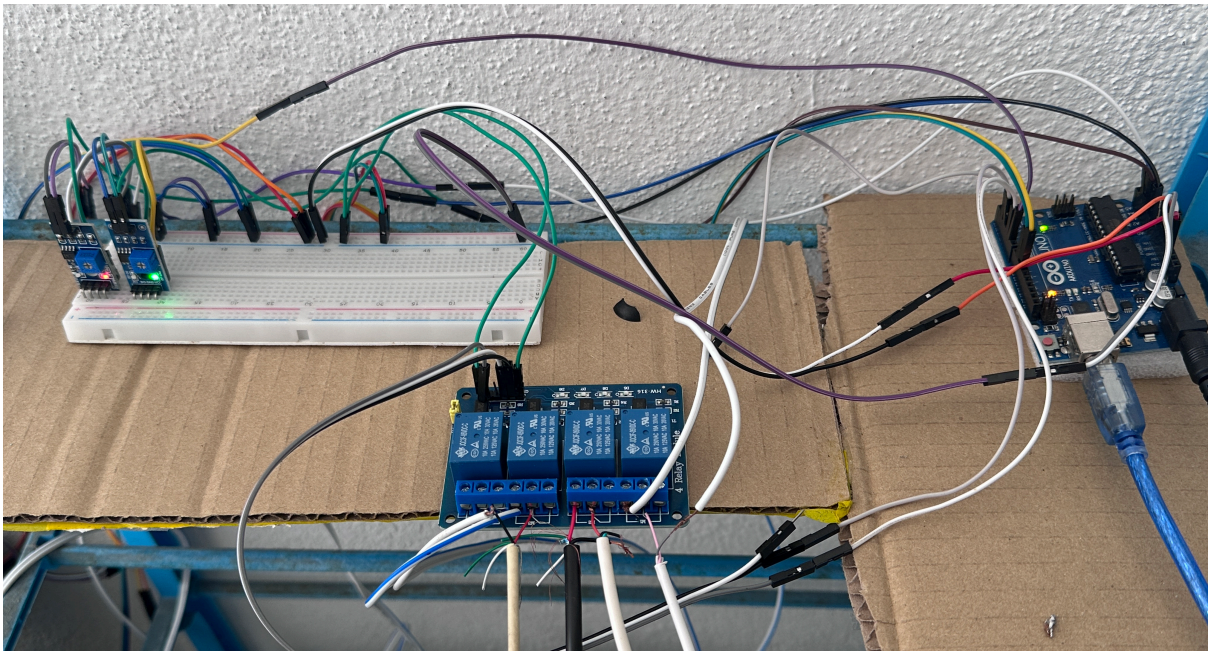


Figure 14 Hardware setup of the Arduino UNO and Relay board

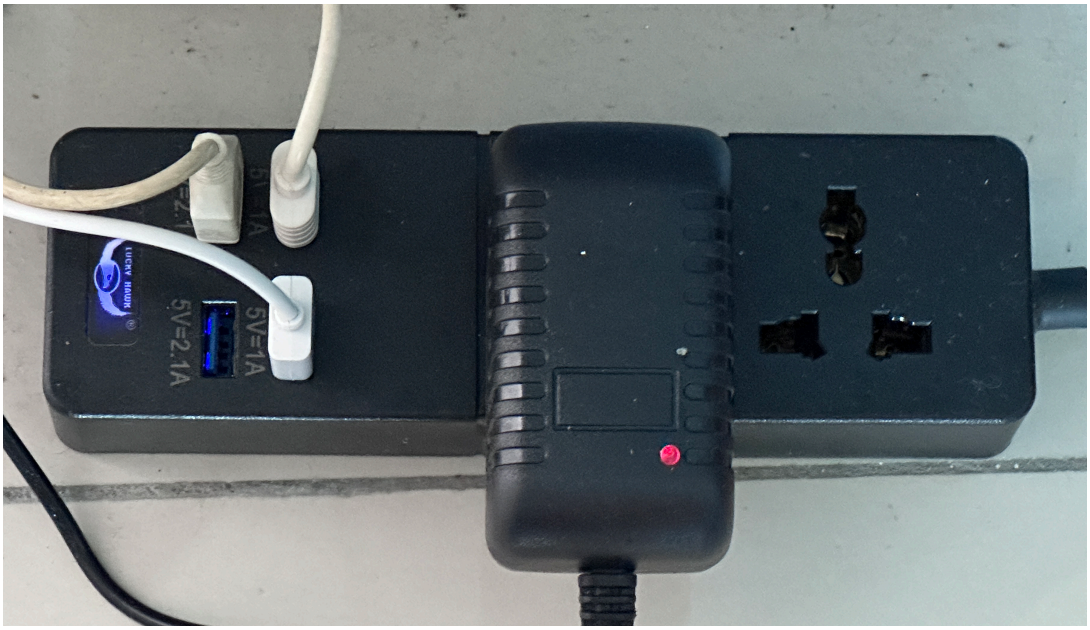


Figure 15 Power supply for Arduino UNO and Actuators

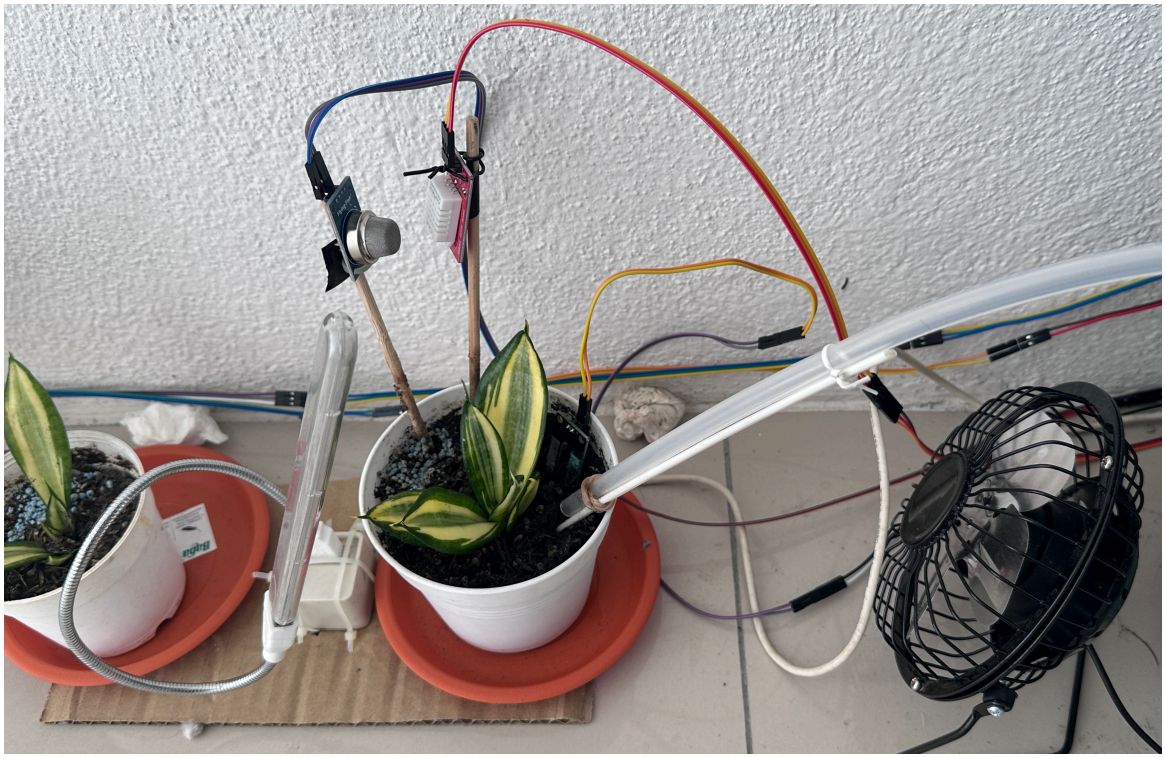


Figure 17 DHT22 Sensor, MQ-135 Sensor, Water pipe, Fan and Full Spectrum LED set up on the plant

Figure 16 Water pump (in the water) and Ultrasonic Sensor to detect water level

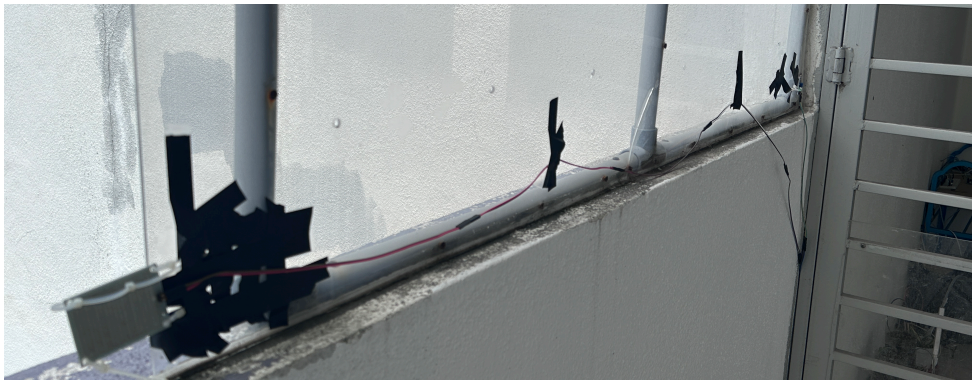


Figure 18 Raindrop sensor (left)

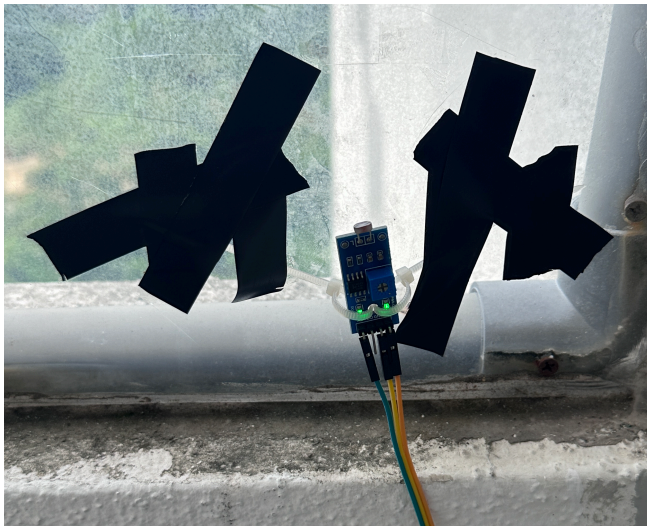


Figure 19 LDR (Light intensity sensor)

The above figures show the hardware setup for the project, Figure 13 shows the connection of the Arduino with the sensors, actuators and relay based on the design shown in Chapter 4.3. For the power supply for the actuators, unwanted USB cables are cut to be used to connect the power supply to the actuators via relay, then a power adapter for Arduino is used to provide power to the board, shown in Figure 13 and 14. In Figure 15, it shows the plant being planted with sensors as well as actuators such as fan, LED and the water pipe that is connected to the water pump in Figure 16. A series of sensors such as DHT22, Soil Moisture Sensor, Air Quality Sensor and Ultrasonic Sensor can be seen in both figures. For the raindrop sensor in Figure 17, it is connected uniquely as it needs to detect raindrops, in this case, it is connected and extended to the side of the corridor using connecting wires. Besides that, the LDR sensor in Figure 18 that collect intensity readings are connected far away from the LED, because it needs to detect the sunlight instead of the corridor lights.

5.2 Software Setup

```
#include <DHT.h>
#define DHTPIN 6
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
#define trigPin 9
#define echoPin 10
#define LDRAnalog A0
#define rainAnalog A1
#define moistAnalog A2
#define airAnalog A3

// Variables
int chk;
float humi;
float temp;
int moist;
int sensor_pin = A0;
int output_value;
long duration, distance;
int airValue;
int rainValue;
int intensityValue;

void setup() {
  Serial.begin(9600);
  Serial.println("Setup started...");
  dht.begin();
  pinMode(5, OUTPUT);    // water pump
  pinMode(3, OUTPUT);    // light
  pinMode(2, OUTPUT);    // fan
  digitalWrite(5, HIGH); // water pump off
  digitalWrite(3, HIGH); // light off
  digitalWrite(2, HIGH); // fan off
  pinMode(9, OUTPUT);    // Sets the trigPin as an OUTPUT
  pinMode(10, INPUT);    // Sets the echoPin as an INPUT
}
```

Figure 20 Arduino IDE Code Snippet 1

```
void loop() {
  // Run the main loop code
  Serial.println("Running main loop code...");
  // DHT22 Codes
  humi = dht.readHumidity();
  temp = dht.readTemperature();
  Serial.print("Humidity: ");
  Serial.print(humi);
  Serial.print("%     ");
  Serial.print("Temp: ");
  Serial.print(temp);
  Serial.println("°C");

  // MQ135 Sensor
  airValue = analogRead(airAnalog);
  Serial.print("The amount of Oxygen (in PPM): ");
  Serial.println(airValue);
  if (airValue < 300) {
    Serial.println("Status: Bad air quality! Auto turning fan on.");
    digitalWrite(2, LOW);
  } else {
    Serial.println("Status: Air quality normal! Fan will be off.");
    digitalWrite(2, HIGH);
  }

  // Raindrops sensor
  rainValue = analogRead(rainAnalog);
  rainValue = map(rainValue, 700, 300, 0, 100);
  if (rainValue < 30) {
    Serial.println("Status: Not raining.");
    rainValue = 0; // Not raining
  } else if (rainValue < 60) {
    Serial.println("Status: Drizzling.");
    rainValue = 1; // Drizzling
  } else {
    Serial.println("Status: Heavy rain.");
    rainValue = 3; // Heavy rain
  }
```

Figure 21 Arduino IDE Code Snippet 2

```
// LDR sensor
intensityValue = analogRead(LDRAnalog);
intensityValue = map(intensityValue, 640, 12, 0, 100);
Serial.print("Intensity: ");
Serial.println(intensityValue);
if (intensityValue < 40) {
  Serial.println("Status: It is DARK! Auto turning lights on.");
  digitalWrite(3, LOW);
} else {
  Serial.println("Status: Sufficient Light! Turning lights off.");
  digitalWrite(3, HIGH);
}
// FC-28 Sensor
moist = analogRead(moistAnalog);
moist = map(moist, 695, 15, 0, 100);
Serial.print("Moisture level: ");
Serial.print(moist);
Serial.println(" %");
if (moist < 50) {
  Serial.println("Status: Dry soil! Auto turning pump on.");
  digitalWrite(5, LOW);
  delay(500);
  digitalWrite(5, HIGH);
} else {
  Serial.println("Status: Sufficient water! Turning pumps off.");
  digitalWrite(5, HIGH);
}
// Ultrasonic Sensor
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = duration * 0.034 / 2;
distance = map(distance, 25, 10, 0, 6000);
Serial.print("Water Level: ");
Serial.print(distance);
Serial.println(" ml");

// Log the data
logData();

// Delay for 4 hours (14400000 milliseconds)
// Delay for 5 seconds during testing (5000 milliseconds)
delay(14400000);
```

Figure 22 Arduino IDE Code Snippet 3

```
void logData() {
  Serial.print(", Humidity: ");
  Serial.print(humi);
  Serial.print(", Temp: ");
  Serial.print(temp);
  Serial.print(", Oxygen: ");
  Serial.print(airValue);
  Serial.print(", Rain: ");
  Serial.print(rainValue);
  Serial.print(", Light intensity: ");
  Serial.print(intensityValue);
  Serial.print(", Moisture: ");
  Serial.print(moist);
  Serial.print(", Water Level: ");
  Serial.println(distance);
}
```

Figure 23 Arduino IDE Code Snippet 4

The figures above show the codes that is programmed into the Arduino UNO board using the Arduino IDE, Figure 20 will be the initialization, defining libraries and variables, and setup of the pins. Then the loop program is defined in Figure 21 and Figure 22, it collects readings from the sensors and map the readings into readable readings. Besides that, there are predefined threshold in multiple readings, for instance, the air quality, soil moisture and light intensity. These thresholds are set to activate the actuators when needed, providing automated irrigation. After that, the logData() function will be called to log the readings to be parsed to further processing, then a delay of 14400000 milliseconds or 4 hours will be called to loop the program again after 4 hours. Figure 23 shows the definition of the logData() function, it basically prints the reading data into the serial monitor which will then be parsed to the python script to be processed.

```
import serial
import time
from datetime import datetime
import matplotlib.pyplot as plt
import telegram
import os
import asyncio
from PIL import Image, ImageDraw, ImageFont
import matplotlib.dates as mdates

# Helper function to get or create an event loop
def get_or_create_event_loop():
    try:
        loop = asyncio.get_event_loop()
        if loop.is_closed():
            loop = asyncio.new_event_loop()
            asyncio.set_event_loop(loop)
    except RuntimeError:
        loop = asyncio.new_event_loop()
        asyncio.set_event_loop(loop)
    return loop

# Asynchronous function to send images to a Telegram bot
async def send_images_to_telegram(collage_image_file):
    if os.path.exists(collage_image_file):
        print(f"Sending {collage_image_file} to Telegram")
        try:
            with open(collage_image_file, 'rb') as photo:
                await bot.send_photo(chat_id=chat_id, photo=photo)
        except telegram.error.TimedOut:
            print(f"Failed to send {collage_image_file} due to timeout")
    else:
        print(f"File {collage_image_file} does not exist")

# Function to check if a serial port is available
def is_port_available(port):
    try:
        ser = serial.Serial(port)
        ser.close()
        return True
    except serial.SerialException:
        return False

# Function to format the x-axis of a plot
def format_x_axis():
    plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%H:%M:%S'))
    plt.gca().xaxis.set_major_locator(mdates.AutoDateLocator())

# Configure the serial port
port = '/dev/cu.usbmodem1301'
if is_port_available(port):
    ser = serial.Serial(port, 9600)
    time.sleep(2)
else:
    print(f"Port {port} is not available. Please check the connection.")
    exit(1)
```

Figure 24 Python Script Code Snippet 1

```
# Open a file to log the data
log_file = open("sensor_data.txt", "w")

# Dictionary to store the data for plotting, grouped by day
data_by_day = {}

# Telegram bot setup
bot_token = '7400639437:AAFCq5-tN2Yw-t-XgEOavs1_0cM5UwWVu9o'
chat_id = '-1002188288594'
bot = telegram.Bot(token=bot_token)

# Counter to track the number of data points plotted
plot_counter = 0

try:
    while True:
        # Read a line from the serial port
        line = ser.readline().decode('utf-8').strip()
        print(f"Received line: {line}")

        # Log the data to the file
        log_file.write(line + "\n")
        log_file.flush()

        # Parse the data
        data = line.split(", ")

        if len(data) == 7:
            # Extract and convert data from the line
            humidity = float(data[1].split(": ")[1])
            temperature = float(data[2].split(": ")[1])
            oxygen = int(data[3].split(": ")[1])
            rain = int(data[4].split(": ")[1])
            light_intensity = int(data[5].split(": ")[1])
            moisture = int(data[6].split(": ")[1])
            water_level = int(data[7].split(": ")[1])

            # Debugging: Print parsed values
            print(f"Parsed Data - Humidity: {humidity}, Temperature: {temperature}")
            date_str = datetime.now().strftime("%Y-%m-%d")

            # Store the data in the dictionary
            if date_str not in data_by_day:
                data_by_day[date_str] = {
                    'humidity': [],
                    'temperature': [],
                    'oxygen': [],
                    'rain': [],
                    'light_intensity': [],
                    'moisture': [],
                    'water_level': []
                }
```

Figure 25 Python Script Code Snippet 2

```
        # Append the data to the corresponding lists
          data_by_day[date_str]['humidity'].append(humidity)
          data_by_day[date_str]['temperature'].append(temperature)
          data_by_day[date_str]['oxygen'].append(oxygen)
          data_by_day[date_str]['rain'].append(rain)
          data_by_day[date_str]['light_intensity'].append(light_intensity)
          data_by_day[date_str]['moisture'].append(moisture)
          data_by_day[date_str]['water_level'].append(water_level)

          # Plot the data and save as separate JPEGs
          image_files = []

          plt.figure()
          plt.plot(data_by_day[date_str]['humidity'], label='Humidity')
          format_x_axis()
          plt.xlabel('Time')
          plt.ylabel('Humidity')
          plt.title('Humidity over Time')
          plt.legend()
          humidity_image = f"{datetime.now().strftime('%Y-%m-%d_%H-%M-%S')}_humidity.jpg"
          plt.savefig(humidity_image)
          plt.close()
          image_files.append(humidity_image)

          plt.figure()
          plt.plot(data_by_day[date_str]['temperature'], label='Temperature')
          format_x_axis()
          plt.xlabel('Time')
          plt.ylabel('Temperature')
          plt.title('Temperature over Time')
          plt.legend()
          temperature_image = f"{datetime.now().strftime('%Y-%m-%d_%H-%M-%S')}_temperature.jpg"
          plt.savefig(temperature_image)
          plt.close()
          image_files.append(temperature_image)

          plt.figure()
          plt.plot(data_by_day[date_str]['oxygen'], label='Oxygen')
          format_x_axis()
          plt.xlabel('Time')
          plt.ylabel('Oxygen')
          plt.title('Oxygen over Time')
          plt.legend()
          oxygen_image = f"{datetime.now().strftime('%Y-%m-%d_%H-%M-%S')}_oxygen.jpg"
          plt.savefig(oxygen_image)
          plt.close()
          image_files.append(oxygen_image)
```

Figure 26 Python Script Code Snippet 3

```
        plt.figure()
        plt.plot(data_by_day[date_str]['rain'], label='Rain')
        format_x_axis()
        plt.xlabel('Time')
        plt.ylabel('Rain')
        plt.title('Rain over Time')
        rain_legend_labels = ['0: No Rain', '1: Drizzling', '2: Heavy Rain']
        rain_handles = [plt.Line2D([0], [0], color='blue', lw=2, label=label) for label in rain_legend_labels]
        plt.legend(handles=rain_handles)
        rain_image = f"{datetime.now().strftime('%Y-%m-%d_%H-%M-%S')}_rain.jpg"
        plt.savefig(rain_image)
        plt.close()
        image_files.append(rain_image)

        plt.figure()
        plt.plot(data_by_day[date_str]['light_intensity'], label='Light Intensity')
        format_x_axis()
        plt.xlabel('Time')
        plt.ylabel('Light Intensity')
        plt.title('Light Intensity over Time')
        plt.legend()
        light_intensity_image = f"{datetime.now().strftime('%Y-%m-%d_%H-%M-
        %S')}_light_intensity.jpg"
        plt.savefig(light_intensity_image)
        plt.close()
        image_files.append(light_intensity_image)

        plt.figure()
        plt.plot([date_str]['moisture'], label='Moisture')
        format_x_axis()
        plt.xlabel('Time')
        plt.ylabel('Moisture')
        plt.title('Moisture over Time')
        plt.legend()
        moisture_image = f"{datetime.now().strftime('%Y-%m-%d_%H-%M-%S')}_moisture.jpg"
        plt.savefig(moisture_image)
        plt.close()
        image_files.append(moisture_image)

        plt.figure()
        plt.plot(data_by_day[date_str]['water_level'], label='Water Level')
        format_x_axis()
        plt.xlabel('Time')
        plt.ylabel('Water Level')
        plt.title('Water Level over Time')
        plt.legend()
        water_level_image = f"{datetime.now().strftime('%Y-%m-%d_%H-%M-%S')}_water_level.jpg"
        plt.savefig(water_level_image)
        plt.close()
        image_files.append(water_level_image)
```

Figure 27 Python Script Code Snippet 5

```
# Create a collage of the images with 3 images per row
images = [Image.open(img) for img in image_files]
widths, heights = zip(*(i.size for i in images))

# Assuming all images have the same width and height
image_width = widths[0]
image_height = heights[0]

# Calculate the number of rows needed
num_images = len(images)
num_columns = 3
num_rows = (num_images + num_columns - 1) // num_columns

# Calculate the total width and height of the collage
total_width = num_columns * image_width
title_height = 50
total_height = num_rows * image_height

collage_image = Image.new('RGB', (total_width, total_height))

# Draw the title
draw = ImageDraw.Draw(collage_image)
title_text = f"Data Readings of {datetime.now().strftime('%Y-%m-%d_%H-%M-%S')}"
font_size = 40
font = ImageFont.load_default(),
text_bbox = draw.textbbox((0, 0), title_text, font=font)
text_width = text_bbox[2] - text_bbox[0]
text_height = text_bbox[3] - text_bbox[1]
title_position = ((total_width - text_width) // 2, 10)
draw.text(title_position, title_text, fill="white", font=font)

# Place each image in the correct position within the collage
for index, img in enumerate(images):
    row = (index // num_columns)
    col = index % num_columns
    x_offset = col * image_width
    y_offset = row * image_height + title_height
    collage_image.paste(img, (x_offset, y_offset))

collage_image_file = f"{datetime.now().strftime('%Y-%m-%d_%H-%M-%S')}_collage.jpg"
collage_image.save(collage_image_file)

# Increment the plot counter
plot_counter += 1
```

Figure 28 Python Script Code Snippet 6

```
            # Send images to Telegram bot after six data points have been plotted
            if plot_counter >= 6:
                loop = get_or_create_event_loop()
                loop.run_until_complete(send_images_to_telegram(collage_image_file))
                plot_counter = 0  # Reset the counter

        time.sleep(1)

except Exception as e:
    print(f"An error occurred: {e}")
finally:
    # Close the serial connection and log file
    ser.close()
    log_file.close()
```

Figure 29 Python Script Code Snippet 7

Figure 24 to Figure 29 shows the Python Script codes, in this code, the data are parsed from the serial port, which is the Arduino UNO connected to the PC, after that the data will be generated into different graphs. Then the graphs are sent to the Telegram bot and then to the user using the Telegram platform. The import of libraries is mainly used to handle serial communication, plotting of graphs, image creating and editing and most importantly the asynchronous operations. Because it handles non-blocking operations, particular for sending the images to Telegram.

Based on the codes, there are few functions implemented, get_orcreate_event_loop() function make sure that there is an active asyncio event loop and create a new one if necessary. The send_images_to_telegram(collage_image_file) function sends the collaged image to Telegram bot if the image file exists. Next is the is_port_available(port) is to check if the serial port is connected and ready to run. The format_x_axis() function is to format the x-axis of the graphs to be the real time displaying in HH:MM format.

In the main loop codes, data are continuously read from the serial port, data collected will be logged into a text file and a dictionary will be initialized to store the data grouped by day. Then the graphs are plotted according to the data stored. Graphs generated will then be collaged into an image to be sent to the Telegram that has been set up using the token and chatID inserted. The purpose of sending a collaged image is to avoid sending the graphs separately creating problems such as spam or bot being banned.

5.3 Setting and Configuration

To set up and configure the whole system, first, the Arduino UNO board will be programmed with the codes provided in Chapter 5.2 by connecting a USB cable from the Arduino UNO to the PC. Then, the Arduino UNO will be turned on using a power adapter to let it run for one day first. The purpose of doing so is to let all the sensors to warm up or to get used to the surroundings first, especially the MQ135 Air Quality Sensor. After that, a Telegram Bot is required to allow graphs to be sent for observing, therefore BotFather bot will be used to create a personal bot that can be personalize, shown in the figures below. The bot token given will be inserted into the Python Script shown in Chapter 5.2. Then lastly, the Python Script will be run with the USB cable connected with the Arduino UNO board to the PC. Then, the running Python Script should parse the serial outputs from the Arduino and process the data and send the graphs to the Telegram Bot.

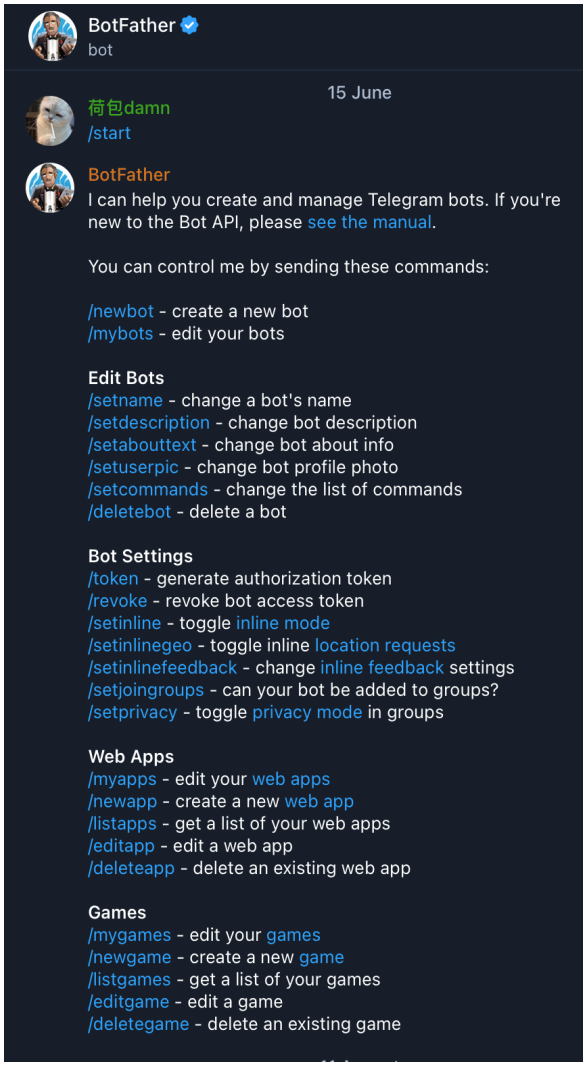


Figure 30 Telegram Bot setup 1

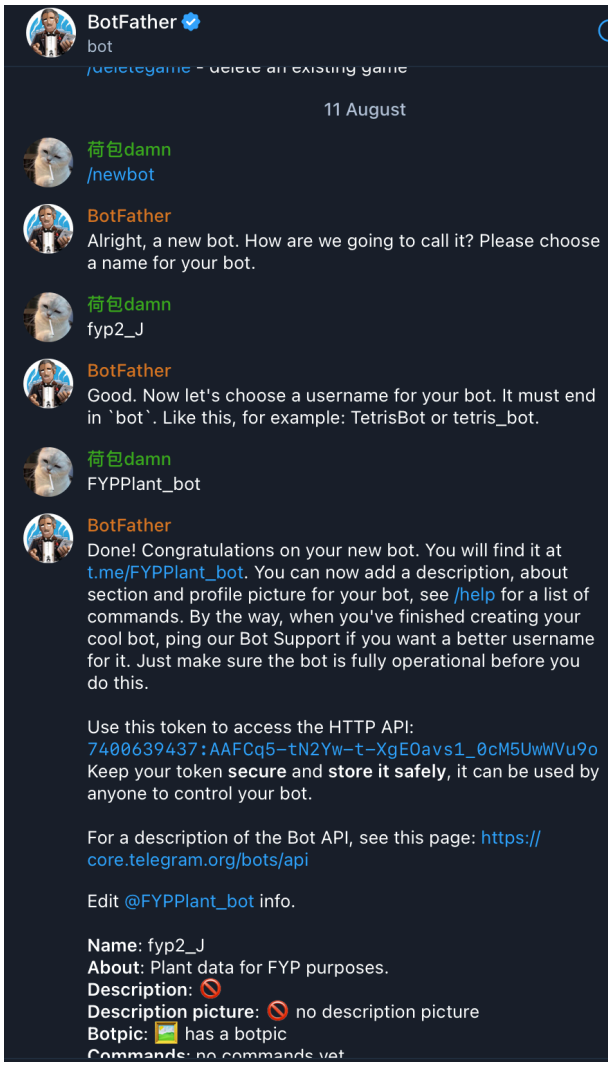


Figure 31 Telegram Bot setup 2

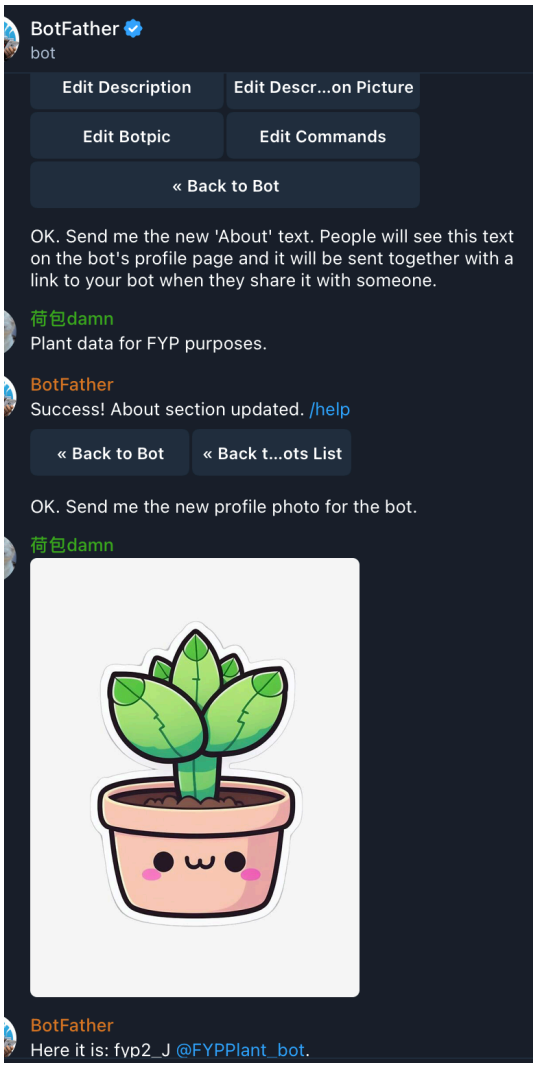
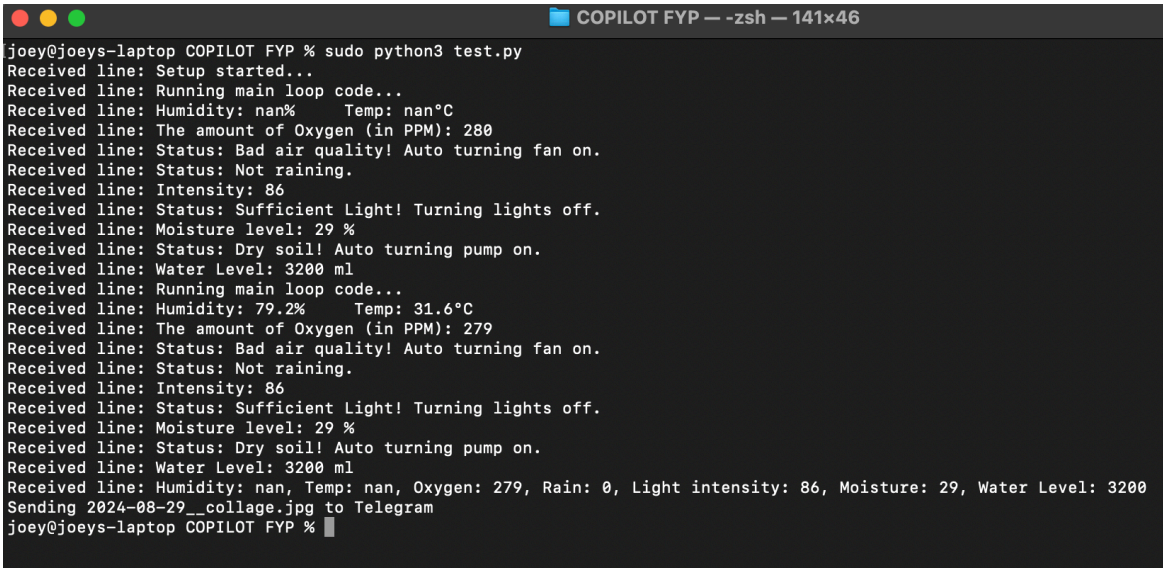


Figure 32 Telegram Bot setup 3



5.4 System Operation

Figure 33 Python Script output (screenshot)

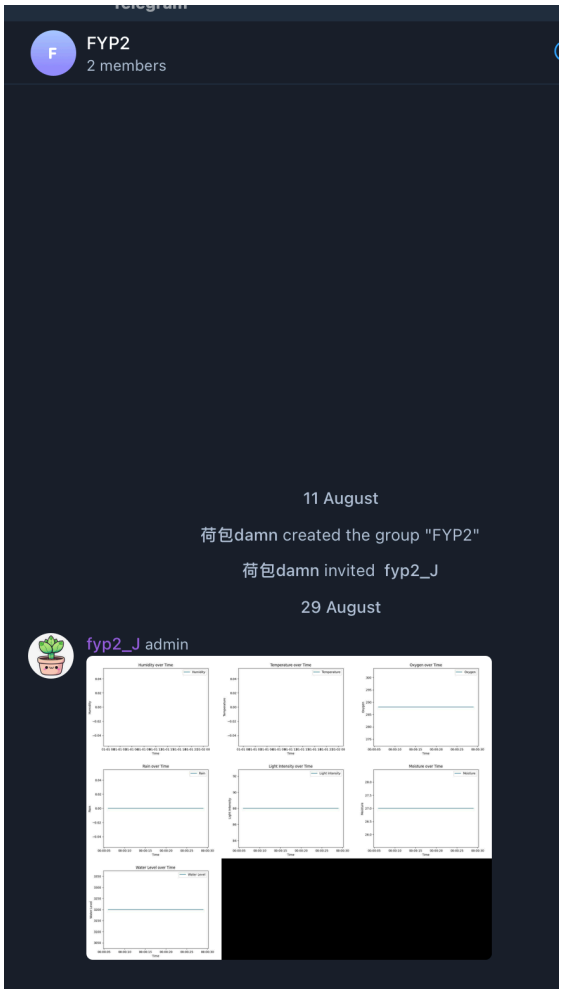


Figure 34 Telegram Bot sending the collaged image of graphs

When the system is completely configured and set, the PC connected to the Arduino UNO need to run the Python Script using terminal in MacOS or command prompt in Windows. The screenshot above shows the Python Script being run and the data being parsed, and graphs are sent to the Telegram Bot. The screenshots shown in the figures above are for to testing purposes only, the duration of getting readings is changed to 5 seconds and plotting graphs with only one value. The screenshot of Telegram chat shows the collaged graphs are successfully sent by the bot.

5.5 Implementation Issues and Challenges

Initially the system will use a SD Card Module to log and store the data, but the SD Card Module did not work during the integration part, therefore, the method of using Python Script is introduced and data are stored in the PC storage. The data are to be deleted automatically when the collaged image is sent successfully to the user, ensuring the storage efficient usage.

Besides that, the initial plan of this project utilizes the Wi-Fi module to send data to Telegram, but the component, which is ESP01, it has still freshly been introduced, implementing it makes things more complicated, therefore this idea was terminated. Before the implementation of the power supply and USB cables for the actuators, the prototype utilizes the batteries to power up the water pump, fan and LED, but after operating for more than 48 hours, the battery dried up, so direct power from the sockets using trimmed USB cables is currently implemented.

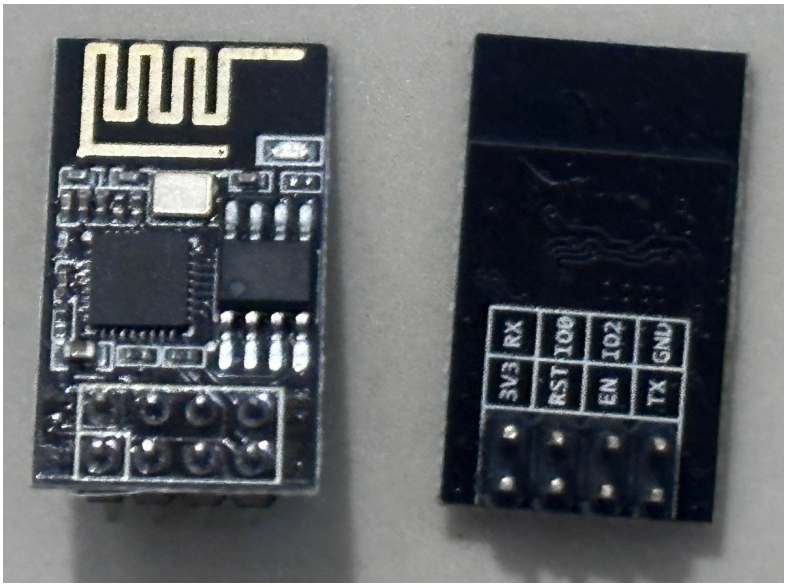


Figure 35 ESP01 Module

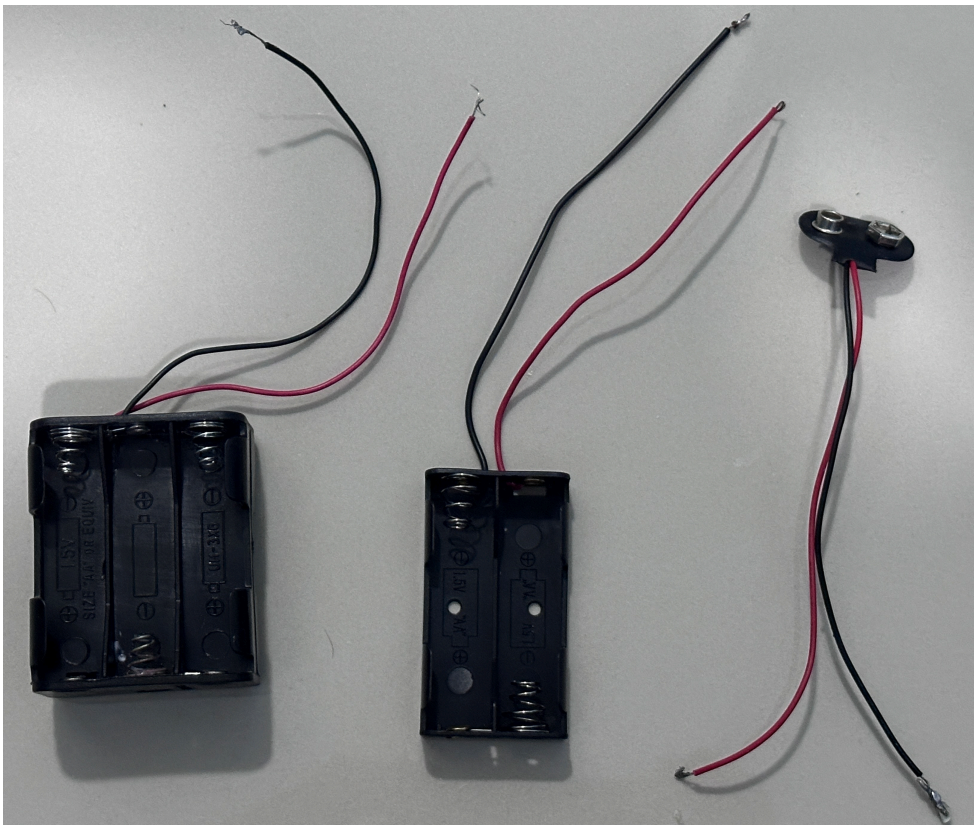


Figure 36 Battery Holders

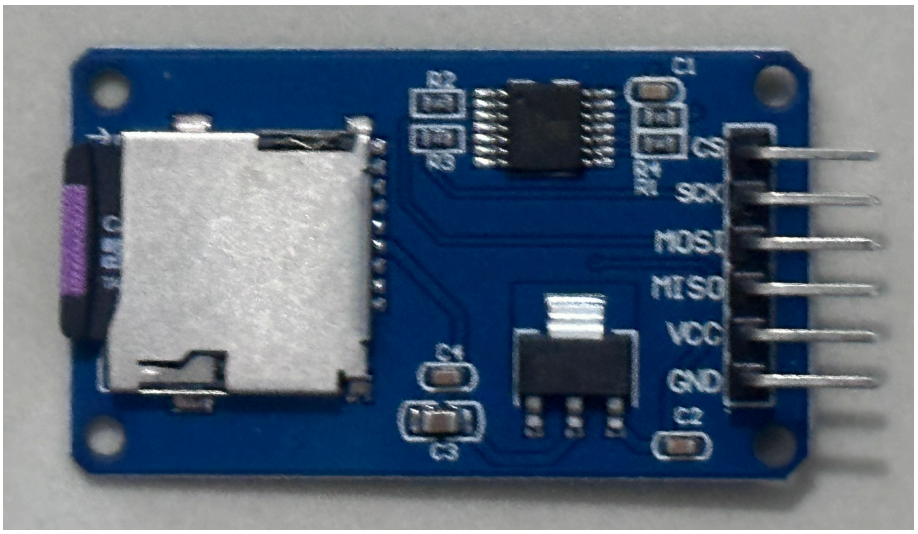


Figure 37 SD Card Module

5.6 Concluding Remark

To conclude this chapter, the system is being set up in an indoor environment which a series of sensors and actuators connected to the Arduino UNO. The plant being used here is the young Sansevieria Plant, a famous plant being placed indoor as a decoration plant. The Arduino UNO must be programmed first before running the Python Code to capture its serial output. Received collaged image of multiple graphs showing readings indicates the system has been set up correctly. Challenges faced during the implementation is the SD card module issues, complication of ESP01 module as well as the batteries drying off issue. All these challenges are solved using alternative solutions.

Chapter 6

System Evaluation and Discussion

6.1 System Testing and Performance Metrics

For the system testing, the system will be tested based on the objectives and what the system are expected to do. The functionality of the system mainly focusses on automated gardening, such as automatic watering, turning on the LED and the fan. Then, the data read by the sensors are to be logged and generated into graphs to let users observe.

Besides that, the performance of the system under different environment will be observed to, the system will be tested to run for one week time. Data and performance collected during this one week will be evaluated.

6.2 Testing Setup and Result

A table of test cases and result are shown below, the test cases are created to ensure that the actuators will be activated when threshold of certain readings are reached, making the system automation functionality reliable. After one day, the data read from the sensors should be collected and generated into different collaged images of graphs. The graphs of the seven days will be shown in this chapter as well. But, due to the lack of internet connection in testing, the result of sending of graphs to Telegram bot will not be shown and evaluated here.

Lastly, to ensure the system is effective in improving plant physical growth and health, the condition of the plant will be observed too, but in this project, the plant being used will not show big difference physically, so the number of new leaves will be used to evaluate the effectiveness.

Table 2 Sensor readings test

Sensors	Readings
Temperature and Humidity (DHT22)	28.2 °C
Air Quality (MQ-135)	494 (no unit)
Soil Moisture (FC-28)	68%
Water Level (Ultrasonic)	3200ml
Light Intensity (LDR Sensor)	62%
Rain (Raindrop sensor module)	0 (0=no rain, 1= drizzling, 2= heavy rain)

Table 3 Actuator status test

Threshold	Actuators	Status
Air quality < 300	Fan	ON
Intensity < 40%	Full Spectrum LEDs	ON
Soil moisture < 50%	Water pump	ON

Table 4 Plant growth

	With Automated System	Without Automated System
New leaves	3	1
Leaves color	Fresh green	Yellowish green

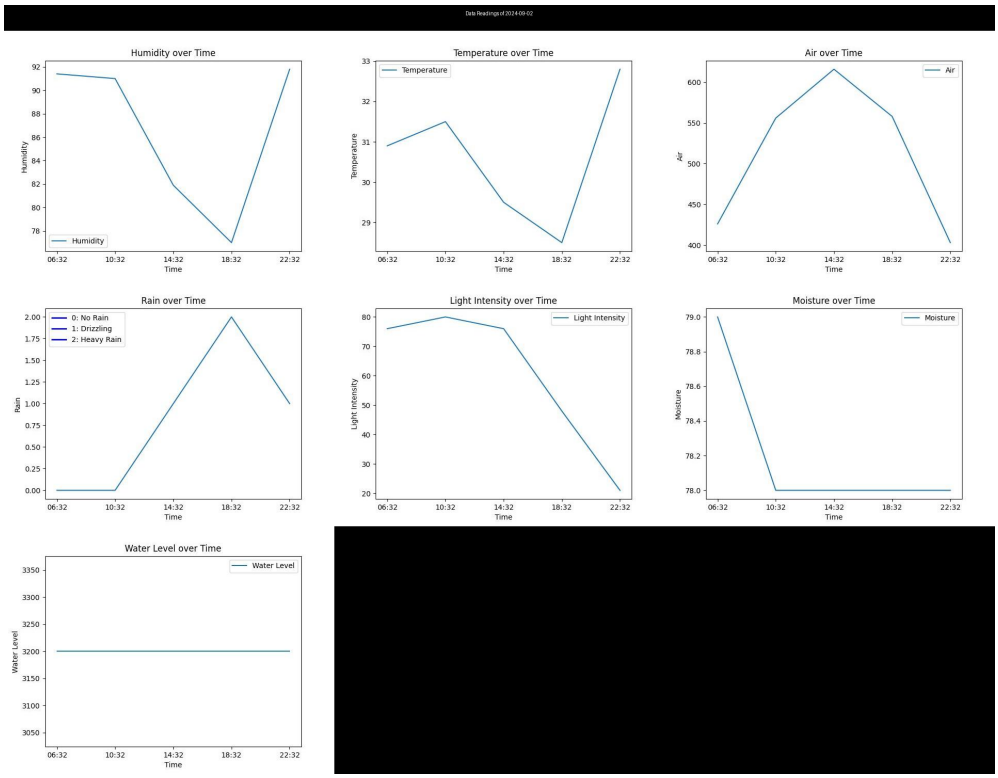


Figure 38 Graph generated on Day 1

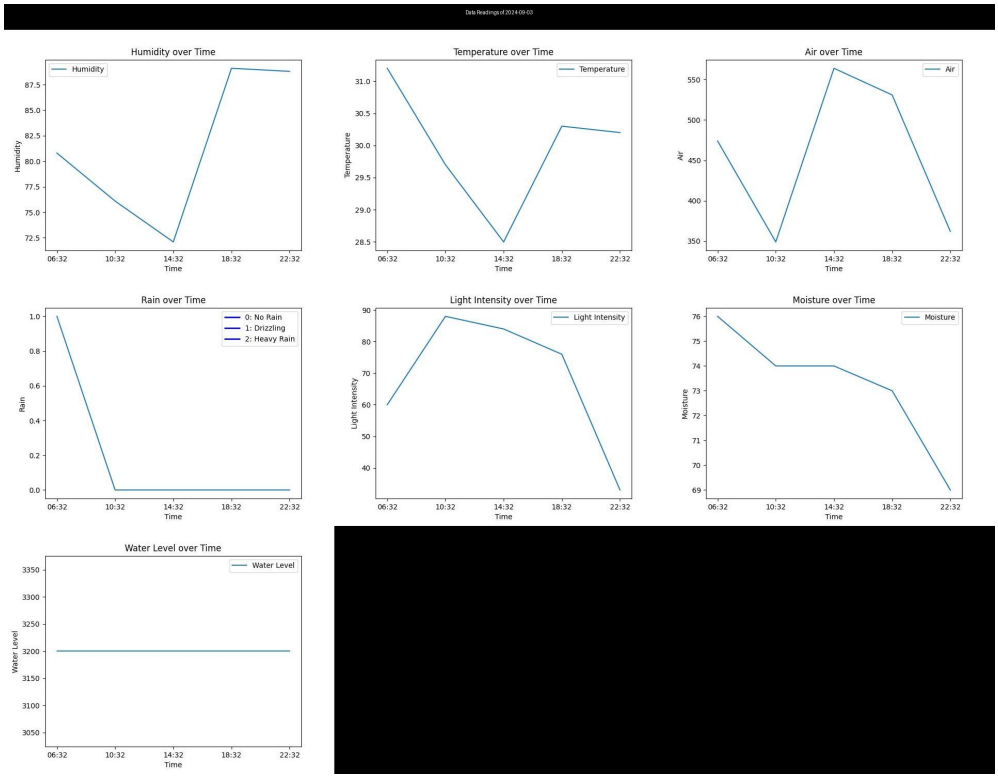


Figure 39 Graph generated on Day 2

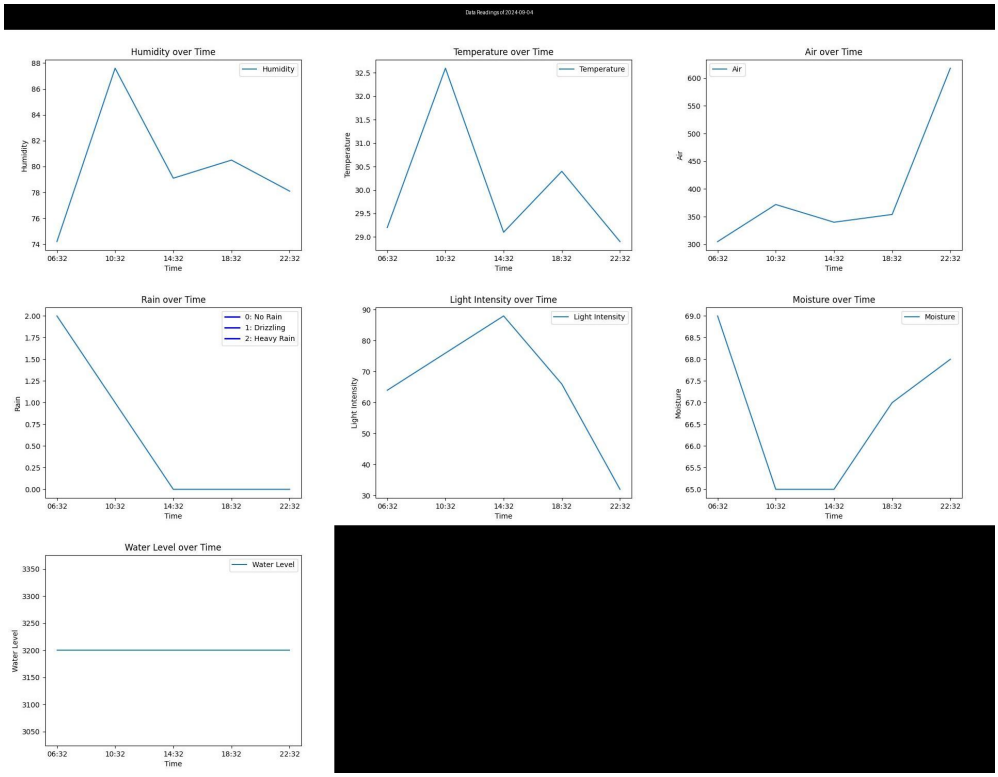


Figure 40 Graph generated on Day 3

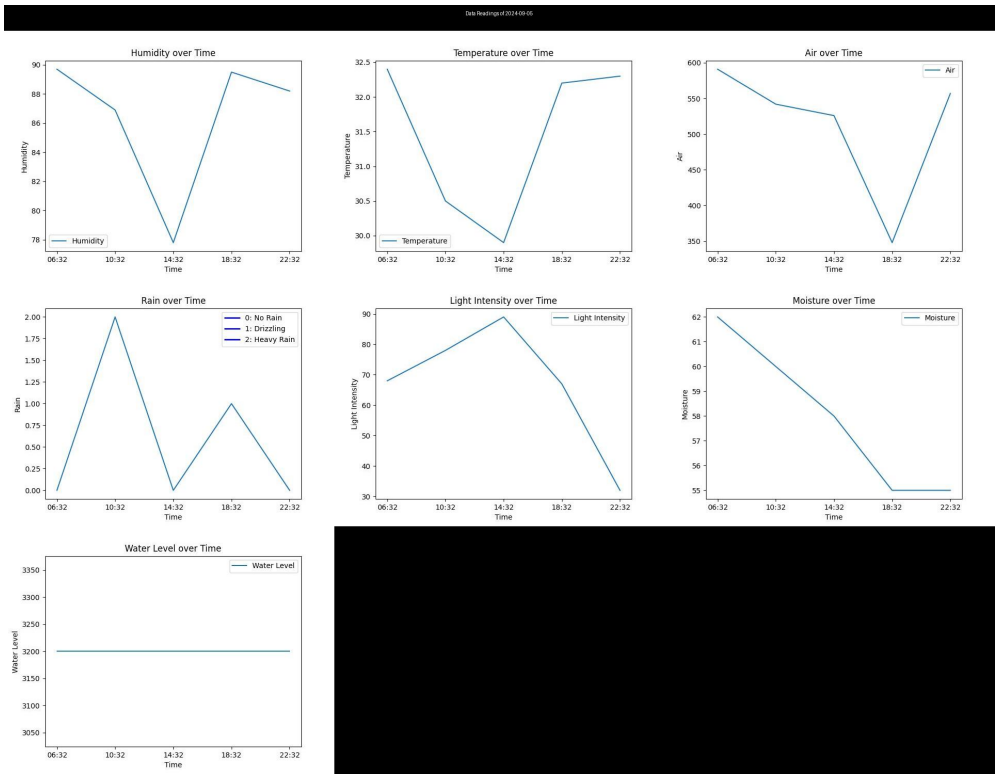


Figure 41 Graph generated on Day 4

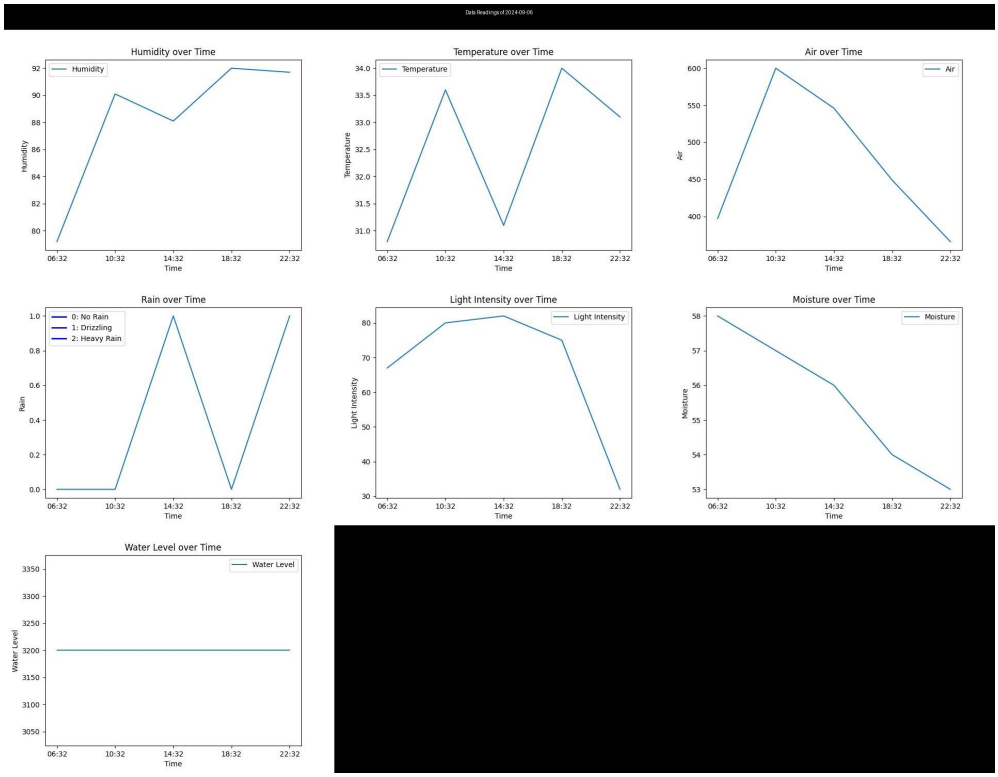


Figure 42 Graph generated on Day 5

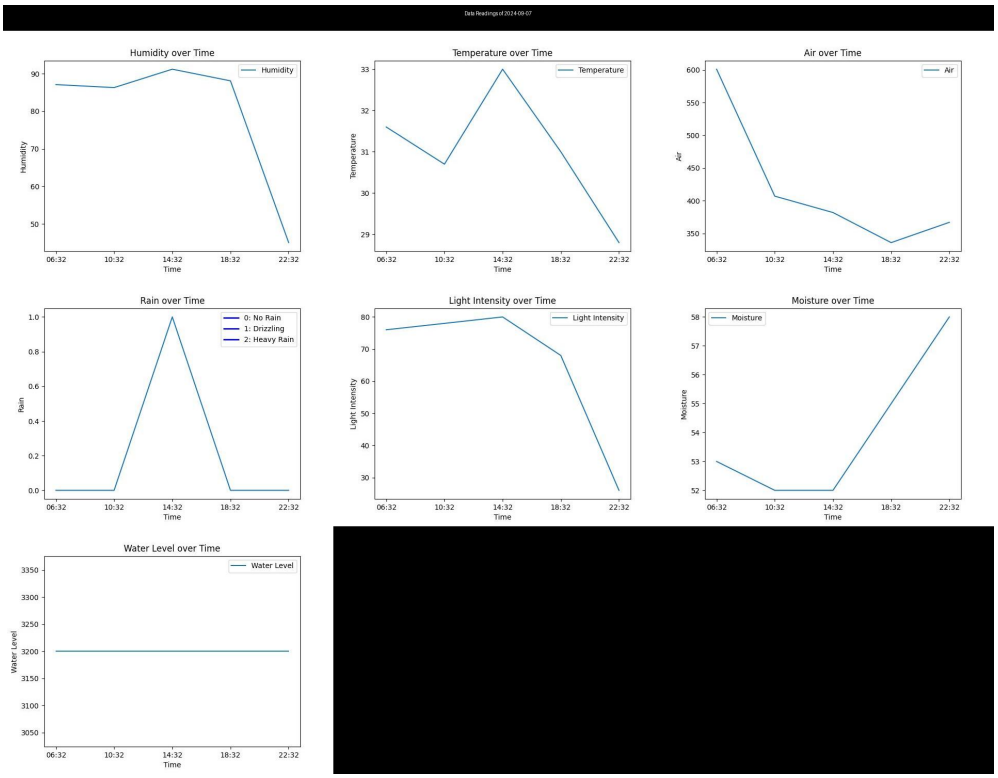


Figure 43 Graph generated on Day 6

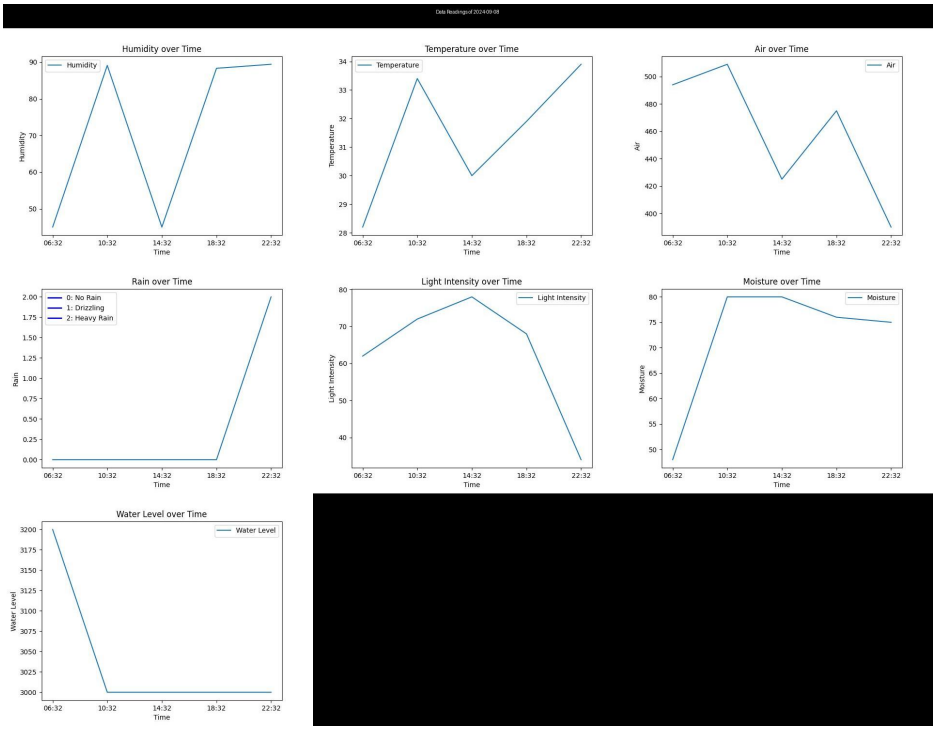


Figure 44 Graph generated on Day 7

6.3 Project Challenges

After the seven days experiment and testing process, there are few issues that caused appeared to cause accuracy issues of the system. First, it's the FC-28 soil moisture sensor issue, it was exposed to air and humid for a long period of time causing it to rust and may not be able to read data accurately. Next, the fan was initially using simple motor and mini fan blades to run, after some time, the motor burnt caused by overloading and overheating. Therefore, a new portable mini fan is implemented. Besides that, the lack of Wi-Fi connection caused this whole project to be conducted using personal tethering hotspot and the experiment part of this project can only be conducted until the generating of graphs, not until the sending of graphs.

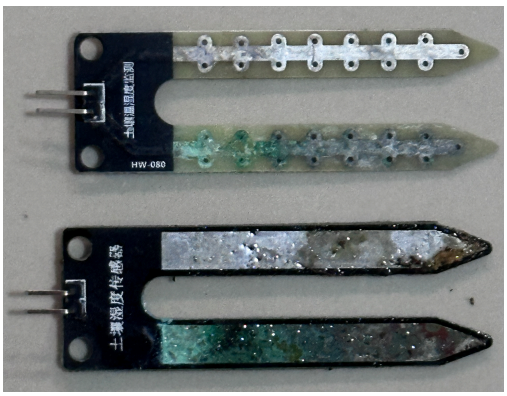


Figure 45 Rusted soil moisture sensors

6.4 Objectives Evaluation

Based on the results, the automated gardening objective is achieved, the sensors keep tracks of the readings accurately and actuators will automatically be activated when the threshold is reached. Besides, the functionality of the Python Script is tested for a few times shown in Chapter 5.3, and it works perfectly, it is not evaluated because there's no stable connection provided for a straight seven-day testing period. Besides that, the plant physical growth has improved in the seven-day period, new leaves can be seen grow up from the middle of the plant compared to the plant being left without any special care, this might be caused by the sufficient light, water and good air circulation provided to the plant all day long.

6.5 Concluding Remark

Based on the system evaluation done, the system developed in this project works fine and the objectives are achieved. The challenges faced are resolved by using another set of soil moisture sensor and a new portable fan. The automation of actuators and sensor readings are tested to be working in good condition, graphs are plotted correctly and can be sent to users when internet connection are stable and available. Lastly, the plant physical growth can be observed to be improved compared to the one without care.

Chapter 7

Conclusion and Recommendation

7.1 Conclusion

In conclusion, this report has explained the overall operation of the system, in the form of diagrams, illustrations, physical hardware setup as well as the software setup, explanations are included to make things clearer and understandable, so that readers can easily remake the system and even make their own changes to improve the system in the future.

The system containing a series of sensors and actuators such as temperature sensor, humidity sensor, water level sensor, light intensity sensor, soil moisture sensor, air quality sensor, rain sensor, fan, full spectrum LEDs, and water pump being illustrated in this report were successfully built. A simple set up and implementation were done to ensure the system was ready to be tested for a week.

After that, the system underwent a testing duration of seven day, few issues occurred, and all the issues are solved with alternative solutions. All the test results are provided in Chapter 6, showing that the system can be used effectively, therefore, the objectives of the project can be said achieved. Finally, the results shows that plant with automated system tends to have better physical growth.

7.2 Recommendation

There are quite a few improvements that can be made to improve this existing system built. For example, two more pumps can be implemented to provide nutrient fluid and pest repellant spray. Both pumps can be set to activate after a certain of time as there are no sensors yet to take readings of soil nutrients and to detect pests. Besides that, to improve the user interaction with the system, NodeMCU can be implemented instead of Arduino UNO as it can provide connectivity, so that users are able to control the actuators and get the reading data anytime they wish. Lastly, the sensor components can be still upgraded to better qualities ones, such as the SM150T soil moisture sensor shown below to avoid the rusting issue of soil moisture sensor.

Figure 46 SM150T Soil Moisture Sensor

REFERENCES

[1]S. Y. Chia and M. W. Lim, "A critical review on the influence of humidity for plant growth forecasting," IOP Conference Series: Materials Science and Engineering, vol. 1257, no. 1, p. 012001, Oct. 2022, doi: https://doi.org/10.1088/1757-899x/1257/1/012001.

[2] V. Kumar. P, K. C. Ramya, Abishek. J.S, Arundhathy. T.S, Bhavvya. B, and Gayathri. V, "Smart Garden Monitoring and Control System with Sensor Technology," 2021 3rd International Conference on Signal Processing and Communication (ICPSC), May 2021, doi: https://doi.org/10.1109/icspc51351.2021.9451788.

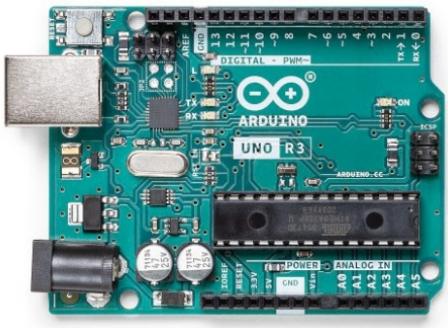
[3] G. R. Choudhari, P. A. Dagale, I. S. Dashetwar, R. R. Desai, and A. A. Marathe, "IoT-based Smart Gardening System," Journal of physics. Conference series, vol. 2601, no. 1, pp. 012006–012006, Sep. 2023, doi: https://doi.org/10.1088/1742-6596/2601/1/012006.

[4] M. M. Rahman, D. L. Field, S. M. Ahmed, M. T. Hasan, M. K. Basher, and K. Alameh, "LED Illumination for High-Quality High-Yield Crop Growth in Protected Cropping Environments," Plants, vol. 10, no. 11, p. 2470, Nov. 2021, doi: https://doi.org/10.3390/plants10112470.

[5] Nina Korlina Madzhi and A. Nor, "Control of Plant Growth by Monitoring Soil Moisture, Temperature and Humidity in Dry Climate," vol. 1192, no. 1, pp. 012027–012027, Nov. 2021, doi: https://doi.org/10.1088/1757-899x/1192/1/012027.

Appendix
Arduino UNO Datasheet

Product Reference Manual
SKU: A000066



Description

The Arduino® UNO R3 is the perfect board to get familiar with electronics and coding. This versatile development board is equipped with the well-known ATmega328P and the ATMega 16U2 Processor.

This board will give you a great first experience within the world of Arduino.

Target areas:

Maker, introduction, industries

Features

- **ATMega328P** Processor
  - **Memory**
    - AVR CPU at up to 16 MHz
    - 32 kB Flash
    - 2 kB SRAM
    - 1 kB EEPROM
  - **Security**
    - Power On Reset (POR)
    - Brown Out Detection (BOD)
  - **Peripherals**
    - 2x 8-bit Timer/Counter with a dedicated period register and compare channels
    - 1x 16-bit Timer/Counter with a dedicated period register, input capture and compare channels
    - 1x USART with fractional baud rate generator and start-of-frame detection
    - 1x controller/peripheral Serial Peripheral Interface (SPI)
    - 1x Dual mode controller/peripheral I2C
    - 1x Analog Comparator (AC) with a scalable reference input
    - Watchdog Timer with separate on-chip oscillator
    - Six PWM channels
    - Interrupt and wake-up on pin change
  - **ATMega16U2 Processor**
    - 8-bit AVR® RISC-based microcontroller
  - **Memory**
    - 16 kB ISP Flash
    - 512B EEPROM
    - 512B SRAM
    - debugWIRE interface for on-chip debugging and programming
  - **Power**
    - 2.7-5.5 volts

CONTENTS

1 The Board

1.1 Application Examples

The UNO board is the flagship product of Arduino. Regardless if you are new to the world of electronics or will use the UNO R3 as a tool for education purposes or industry-related tasks, the UNO R3 is likely to meet your needs.

First entry to electronics: If this is your first project within coding and electronics, get started with our most used and documented board; UNO. It is equipped with the well-known ATmega328P processor, 14 digital input/output pins, 6 analog inputs, USB connections, ICSP header and reset button. This board includes everything you will need for a great first experience with Arduino.

Industry-standard development board: Using the UNO R3 board in industries, there are a range of companies using the UNO R3 board as the brain for their PLC's.

Education purposes: Although the UNO R3 board has been with us for about ten years, it is still widely used for various education purposes and scientific projects. The board's high standard and top quality performance makes it a great resource to capture real time from sensors and to trigger complex laboratory equipment to mention a few examples.

1.2 Related Products

- Arduino Starter Kit
- Arduino UNO R4 Minima
- Arduino UNO R4 WiFi
- Tinkerkit Braccio Robot

2 Ratings

2.1 Recommended Operating Conditions

Symbol	Description	Min	Max
	Conservative thermal limits for the whole board:	-40 °C (-40 °F)	85 °C (185 °F)

NOTE: In extreme temperatures, EEPROM, voltage regulator, and the crystal oscillator, might not work as expected.

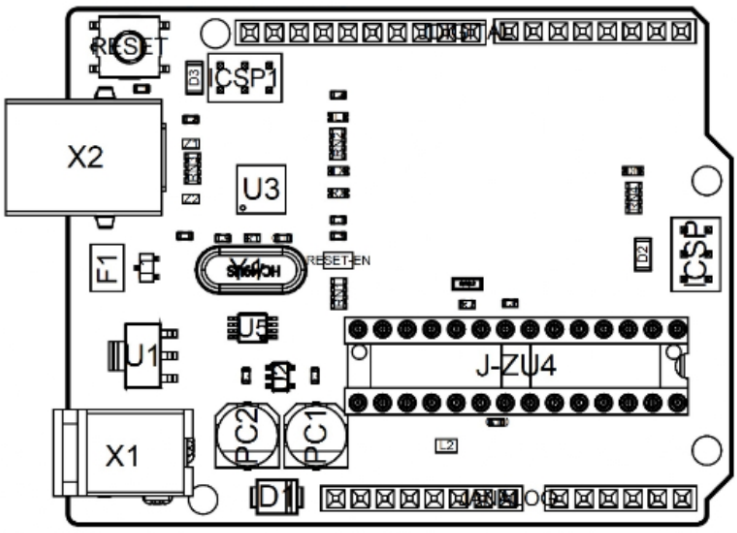
2.2 Power Consumption

Symbol	Description	Min	Typ	Max	Unit
VINMax	Maximum input voltage from VIN pad	6	-	20	V
VUSBMax	Maximum input voltage from USB connector		-	5.5	V
PMax	Maximum Power Consumption	-	-	xx	mA

3 Functional Overview

3.1 Board Topology

Top view



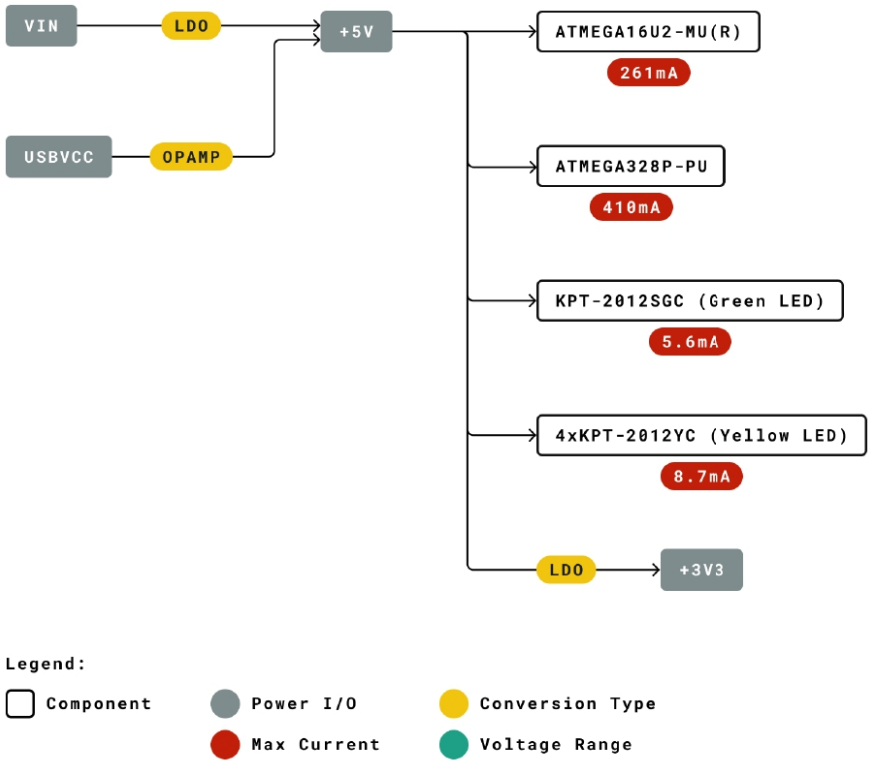
Board topology

Ref.	Description	Ref.	Description
X1	Power jack 2.1x5.5mm	U1	SPX1117M3-L-5 Regulator
X2	USB B Connector	U3	ATMEGA16U2 Module
PC1	EEE-1EA470WP 25V SMD Capacitor	U5	LMV358LIST-A.9 IC
PC2	EEE-1EA470WP 25V SMD Capacitor	F1	Chip Capacitor, High Density
D1	CGRA4007-G Rectifier	ICSP	Pin header connector (through hole 6)
J-ZU4	ATMEGA328P Module	ICSP1	Pin header connector (through hole 6)
Y1	ECS-160-20-4X-DU Oscillator		

3.2 Processor

The Main Processor is a ATmega328P running at up to 20 MHz. Most of its pins are connected to the external headers, however some are reserved for internal communication with the USB Bridge coprocessor.

3.3 Power Tree



Power tree

4 Board Operation

4.1 Getting Started - IDE

If you want to program your UNO R3 while offline you need to install the Arduino Desktop IDE [1] To connect the UNO R3 to your computer, you'll need a USB-B cable. This also provides power to the board, as indicated by the LED.

4.2 Getting Started - Arduino Cloud Editor

All Arduino boards, including this one, work out-of-the-box on the Arduino Cloud Editor [2], by just installing a simple plugin.

The Arduino Cloud Editor is hosted online, therefore it will always be up-to-date with the latest features and support for all boards. Follow **[3]** to start coding on the browser and upload your sketches onto your board.

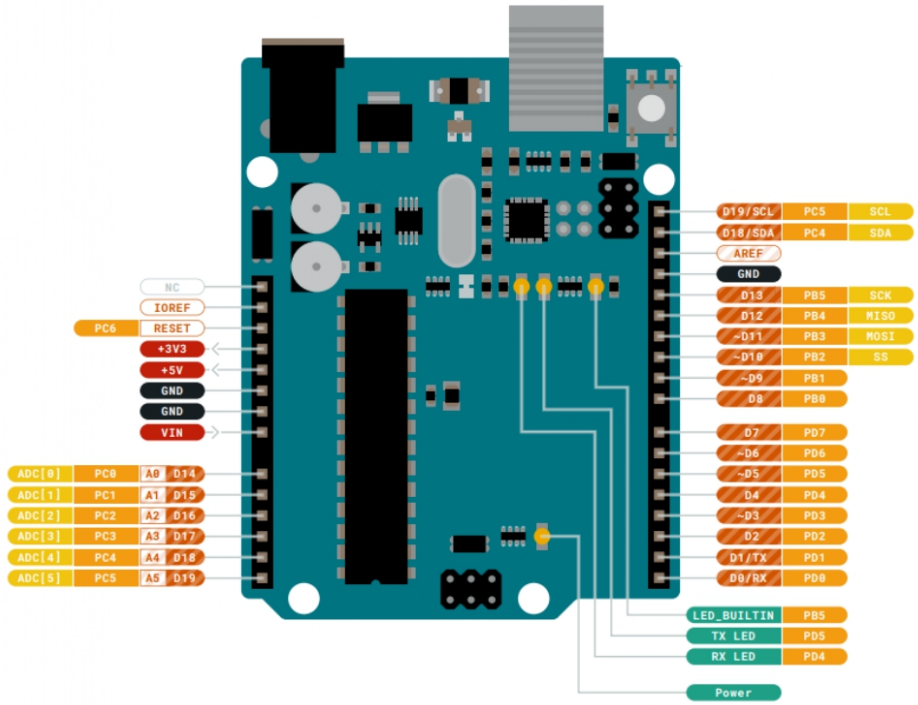
4.3 Sample Sketches

Sample sketches for the UNO R3 can be found either in the "Examples" menu in the Arduino IDE or in the "Documentation" section of the Arduino website [4].

4.4 Online Resources

Now that you have gone through the basics of what you can do with the board you can explore the endless possibilities it provides by checking exciting projects on Arduino Project Hub [5], the Arduino Library Reference [6] and the online Arduino store [7] where you will be able to complement your board with sensors, actuators and more.

5 Connector Pinouts



Pinout

5.1 JANALOG

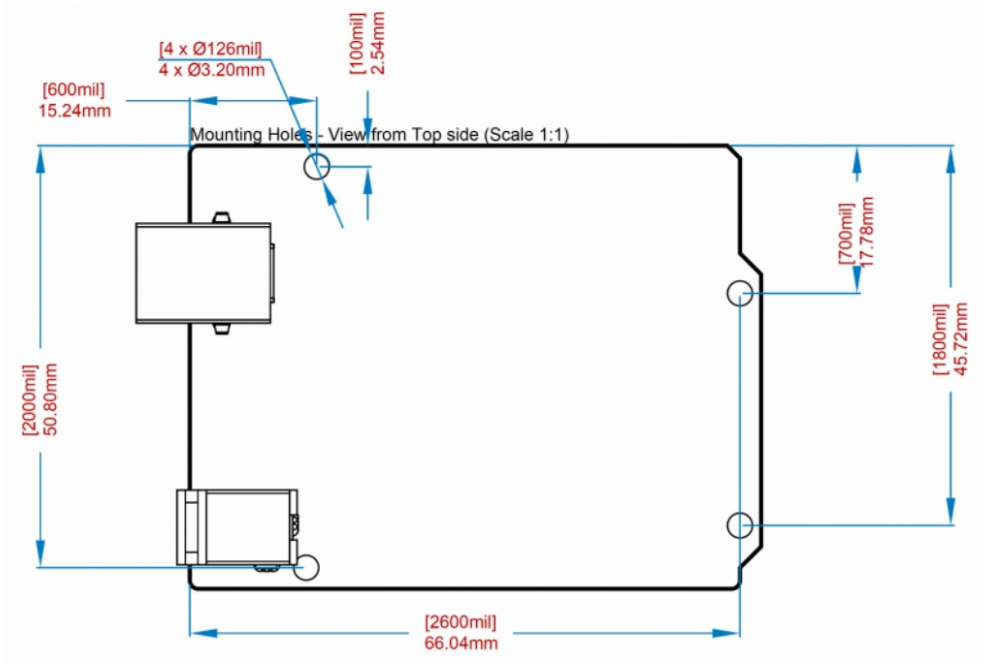
Pin	Function	Type	Description
1	NC	NC	Not connected
2	IOREF	IOREF	Reference for digital logic V - connected to 5V
3	Reset	Reset	Reset
4	+3V3	Power	+3V3 Power Rail
5	+5V	Power	+5V Power Rail
6	GND	Power	Ground
7	GND	Power	Ground
8	VIN	Power	Voltage Input
9	A0	Analog/GPIO	Analog input 0 /GPIO
10	A1	Analog/GPIO	Analog input 1 /GPIO
11	A2	Analog/GPIO	Analog input 2 /GPIO
12	A3	Analog/GPIO	Analog input 3 /GPIO
13	A4/SDA	Analog input/I2C	Analog input 4/I2C Data line
14	A5/SCL	Analog input/I2C	Analog input 5/I2C Clock line

5.2 JDIGITAL

Pin	Function	Type	Description
1	D0	Digital/GPIO	Digital pin 0/GPIO
2	D1	Digital/GPIO	Digital pin 1/GPIO
3	D2	Digital/GPIO	Digital pin 2/GPIO
4	D3	Digital/GPIO	Digital pin 3/GPIO
5	D4	Digital/GPIO	Digital pin 4/GPIO
6	D5	Digital/GPIO	Digital pin 5/GPIO
7	D6	Digital/GPIO	Digital pin 6/GPIO
8	D7	Digital/GPIO	Digital pin 7/GPIO
9	D8	Digital/GPIO	Digital pin 8/GPIO
10	D9	Digital/GPIO	Digital pin 9/GPIO
11	SS	Digital	SPI Chip Select
12	MOSI	Digital	SPI1 Main Out Secondary In
13	MISO	Digital	SPI Main In Secondary Out
14	SCK	Digital	SPI serial clock output
15	GND	Power	Ground
16	AREF	Digital	Analog reference voltage
17	A4/SD4	Digital	Analog input 4/I2C Data line (duplicated)
18	A5/SD5	Digital	Analog input 5/I2C Clock line (duplicated)

5.3 Mechanical Information

5.4 Board Outline & Mounting Holes



Board outline

6 Certifications

6.1 Declaration of Conformity CE DoC (EU)

We declare under our sole responsibility that the products above are in conformity with the essential requirements of the following EU Directives and therefore qualify for free movement within markets comprising the European Union (EU) and European Economic Area (EEA).

ROHS 2 Directive 2011/65/EU	
Conforms to:	EN50581:2012
Directive 2014/35/EU. (LVD)	
Conforms to:	EN 60950-1:2006/A11:2009/A1:2010/A12:2011/AC:2011
Directive 2004/40/EC & 2008/46/EC & 2013/35/EU, EMF	
Conforms to:	EN 62311:2008

6.2 Declaration of Conformity to EU RoHS & REACH 211 01/19/2021

Arduino boards are in compliance with RoHS 2 Directive 2011/65/EU of the European Parliament and RoHS 3 Directive 2015/863/EU of the Council of 4 June 2015 on the restriction of the use of certain hazardous substances in electrical and electronic equipment.

Substance	Maximum limit (ppm)
Lead (Pb)	1000
Cadmium (Cd)	100
Mercury (Hg)	1000
Hexavalent Chromium (Cr6+)	1000
Poly Brominated Biphenyls (PBB)	1000
Poly Brominated Diphenyl ethers (PBDE)	1000
Bis(2-Ethylhexyl} phthalate (DEHP)	1000
Benzyl butyl phthalate (BBP)	1000
Dibutyl phthalate (DBP)	1000
Diisobutyl phthalate (DIBP)	1000

Exemptions: No exemptions are claimed.

Arduino Boards are fully compliant with the related requirements of European Union Regulation (EC) 1907 /2006 concerning the Registration, Evaluation, Authorization and Restriction of Chemicals (REACH). We declare none of the SVHCs (https://echa.europa.eu/web/guest/candidate-list-table), the Candidate List of Substances of Very High Concern for authorization currently released by ECHA, is present in all products (and also package) in quantities totaling in a concentration equal or above 0.1%. To the best of our knowledge, we also declare that our products do not contain any of the substances listed on the "Authorization List" (Annex XIV of the REACH regulations) and Substances of Very High Concern (SVHC) in any significant amounts as specified by the Annex XVII of Candidate list published by ECHA (European Chemical Agency) 1907 /2006/EC.

6.3 Conflict Minerals Declaration

As a global supplier of electronic and electrical components, Arduino is aware of our obligations with regards to laws and regulations regarding Conflict Minerals, specifically the Dodd-Frank Wall Street Reform and Consumer Protection Act, Section 1502. Arduino does not directly source or process conflict minerals such as Tin, Tantalum, Tungsten, or Gold. Conflict minerals are contained in our products in the form of solder, or as a component in metal alloys. As part of our reasonable due diligence Arduino has contacted component suppliers within our supply chain to verify their continued compliance with the regulations. Based on the information received thus far we declare that our products contain Conflict Minerals sourced from conflict-free areas.

7 FCC Caution

Any Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions:

(1) This device may not cause harmful interference

(2) this device must accept any interference received, including interference that may cause undesired operation.

FCC RF Radiation Exposure Statement:

1. This Transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.
2. This equipment complies with RF radiation exposure limits set forth for an uncontrolled environment.
3. This equipment should be installed and operated with minimum distance 20cm between the radiator & your body.

English: User manuals for license-exempt radio apparatus shall contain the following or equivalent notice in a conspicuous location in the user manual or alternatively on the device or both. This device complies with Industry Canada license-exempt RSS standard(s). Operation is subject to the following two conditions:

(1) this device may not cause interference

(2) this device must accept any interference, including interference that may cause undesired operation of the device.

French: Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes :

(1) l' appareil nedoit pas produire de brouillage

(2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

IC SAR Warning:

English This equipment should be installed and operated with minimum distance 20 cm between the radiator and your body.

French: Lors de l' installation et de l' exploitation de ce dispositif, la distance entre le radiateur et le corps est d 'au moins 20 cm.

Important: The operating temperature of the EUT can't exceed 85°C and shouldn't be lower than -40°C.

Hereby, Arduino S.r.l. declares that this product is in compliance with essential requirements and other relevant provisions of Directive 2014/53/EU. This product is allowed to be used in all EU member states.

8 Company Information

Company name	Arduino S.r.l
Company Address	Via Andrea Appiani 25 20900 MONZA Italy

9 Reference Documentation

Reference	Link
Arduino IDE (Desktop)	https://www.arduino.cc/en/Main/Software
Arduino Cloud Editor	https://create.arduino.cc/editor
Arduino Cloud Editor - Getting Started	https://docs.arduino.cc/arduino-cloud/guides/editor/
Arduino Website	https://www.arduino.cc/
Arduino Project Hub	https://create.arduino.cc/projecthub?by=part&part_id=11332&sort=trending
Library Reference	https://www.arduino.cc/reference/en/
Arduino Store	https://store.arduino.cc/

10 Revision History

Date	Revision	Changes
25/04/2024	3	Updated link to new Cloud Editor
26/07/2023	2	General Update
06/2021	1	Datasheet release

中文（ZH）

描述

Arduino UNO R3 是熟悉电子技术和编码的完美开发板。这款多功能开发板配备了著名的 ATmega328P 和 ATMega 16U2 处理器。该开发板将为您带来 Arduino 世界绝佳的初次体验。

目标领域：

创客、介绍、工业领域

特点

- **ATMega328P** 处理器
  - **内存**
    - AVR CPU 频率高达 16 MHz
    - 32KB 闪存
    - 2KB SRAM
    - 1KB EEPROM
  - **安全性**
    - 上电复位 (POR)
    - 欠压检测 (BOD)
  - **外设**
    - 2x 8 位定时器/计数器，带专用周期寄存器和比较通道
    - 1x 16 位定时器/计数器，带专用周期寄存器、输入捕获和比较通道
    - 1x USART，带分数波特率发生器和起始帧信号检测功能
    - 1x 控制器/外设串行外设接口 (SPI)
    - 1x 双模控制器/外设 I2C
    - 1 个模拟比较器 (AC)，带可扩展参考输入
    - 看门狗定时器，带独立的片上振荡器
    - 6 通道 PWM
    - 引脚变化时的中断和唤醒
  - **ATMega16U2 处理器**
    - 基于 AVR® RISC 的 8 位微控制器
  - **内存**
    - 16 KB ISP 闪存
    - 512B EEPROM
    - 512B SRAM

- 用于片上调试和编程的 debugWIRE 接口

- **电源**
  - 2.7-5.5 伏特

目录

11 电路板简介

11.1 应用示例

UNO 电路板是 Arduino 的旗舰产品。无论您是初次接触电路板产品，还是将 UNO 用作教育或工业相关任务的工具，UNO 都能满足您的需求。

初次接触电子技术: 如果这是您第一次参与编码和电子技术项目，那么就从我们最常用、记录最多的电路板 Arduino UNO 开始吧。它配备了著名的 ATmega328P 处理器、14 个数字输入/输出引脚、6 个模拟输入、USB 连接、ICSP 接头和复位按钮。该电路板包含了您获得良好的 Arduino 初次体验所需的一切。

** 行业标准开发板:** 在工业领域使用 Arduino UNO R3 开发板，有许多公司使用 UNO 开发板作为其 PLC 的大脑。

教育用途: 尽管我们推出 UNO R3 电路板已有大约十年之久，但它仍被广泛用于各种教育用途和科学项目。该电路板的高标准和一流性能使其成为从传感器采集实时数据和触发复杂实验室设备等各种应用场合的绝佳资源。

11.2 相关产品

- Starter Kit
- Arduino UNO R4 Minima
- Arduino UNO R4 WiFi
- Tinkerkit Braccio Robot

12 额定值

12.1 建议运行条件

符号	描述	最小值	最大值
	整个电路板的保守温度极限值：	-40 °C (-40°F)	85 °C (185°F)

注意： 在极端温度下，EEPROM、电压调节器和晶体振荡器可能无法正常工作。

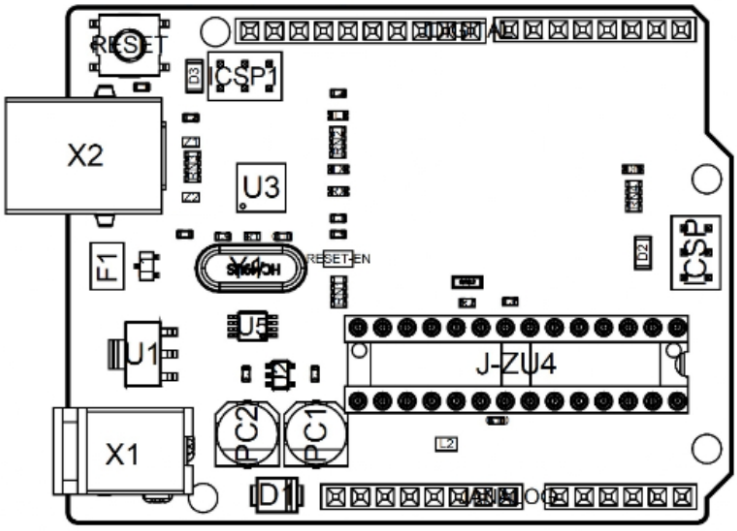
12.2 功耗

符号	描述	最小值	典型值	最大值	单位
VINMax	来自 VIN 焊盘的最大输入电压	6	-	20	V
VUSBMax	来自 USB 连接器的最大输入电压		-	5.5	V
PMax	最大功耗	-	-	xx	mA

13 功能概述

13.1 电路板拓扑结构

俯视图



电路板拓扑结构

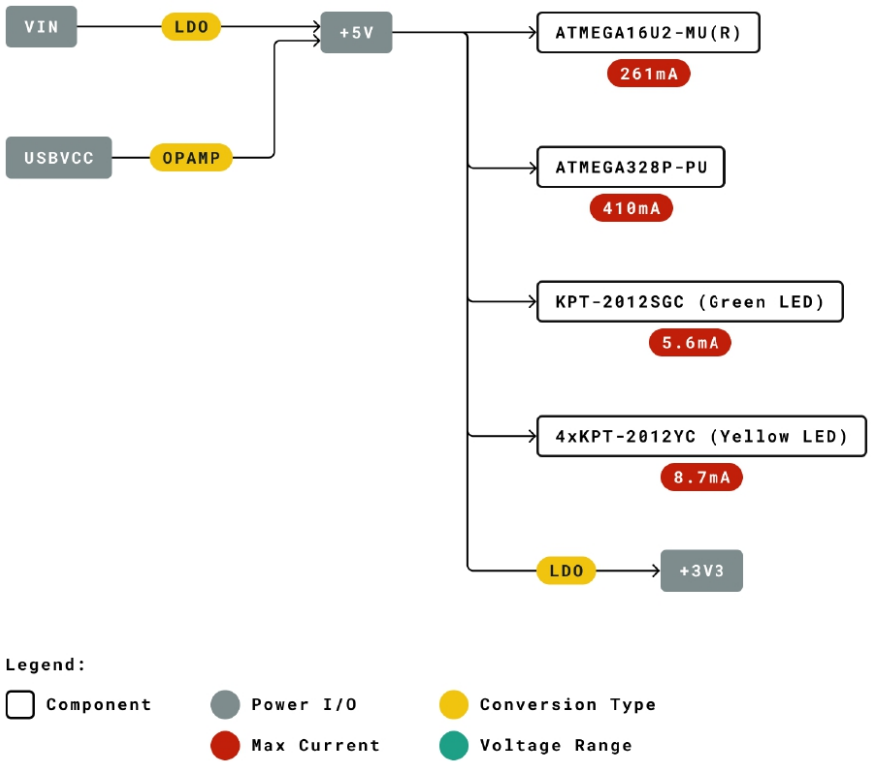
编号	描述	编号	描述
X1	电源插孔 2.1x5.5 毫米	U1	SPX1117M3-L-5 调节器

编号	描述	编号	描述
X2	USB B 连接器	U3	ATMEGA16U2 模块
PC1	EEE-1EA470WP 25V SMD 电容器	U5	LMV358LIST-A.9 IC
PC2	EEE-1EA470WP 25V SMD 电容器	F1	片式电容器，高密度
D1	CGRA4007-G 整流器	ICSP	引脚接头连接器（通过 6 号孔）
J-ZU4	ATMEGA328P 模块	ICSP1	引脚接头连接器（通过 6 号孔）
Y1	ECS-160-20-4X-DU 振荡器		

13.2 处理器

主处理器是 ATmega328P，运行频率高达 20 MHz。它的大部分引脚都与外部接头相连，但也有一些引脚用于与 USB 桥协处理器进行内部通信。

13.3 电源树



电源树

14 电路板操作

14.1 入门指南 - IDE

如需在离线状态下对 Arduino UNO R3 进行编程，则需要安装 Arduino Desktop IDE [1] 若要将 Arduino UNO 连接到计算机，需要使用 USB-B 电缆。如 LED 指示灯所示，该电缆还可以为电路板供电。

14.2 入门指南 - Arduino Cloud Editor

包括本电路板在内的所有 Arduino 电路板，都可以在 Arduino Cloud Editor [2] 上开箱即用，只需安装一个简单的插件即可。

Arduino Cloud Editor 是在线托管的，因此它将始终提供最新功能并支持所有电路板。接下来**[3]**开始在浏览器上编码并将程序上传到您的电路板上。

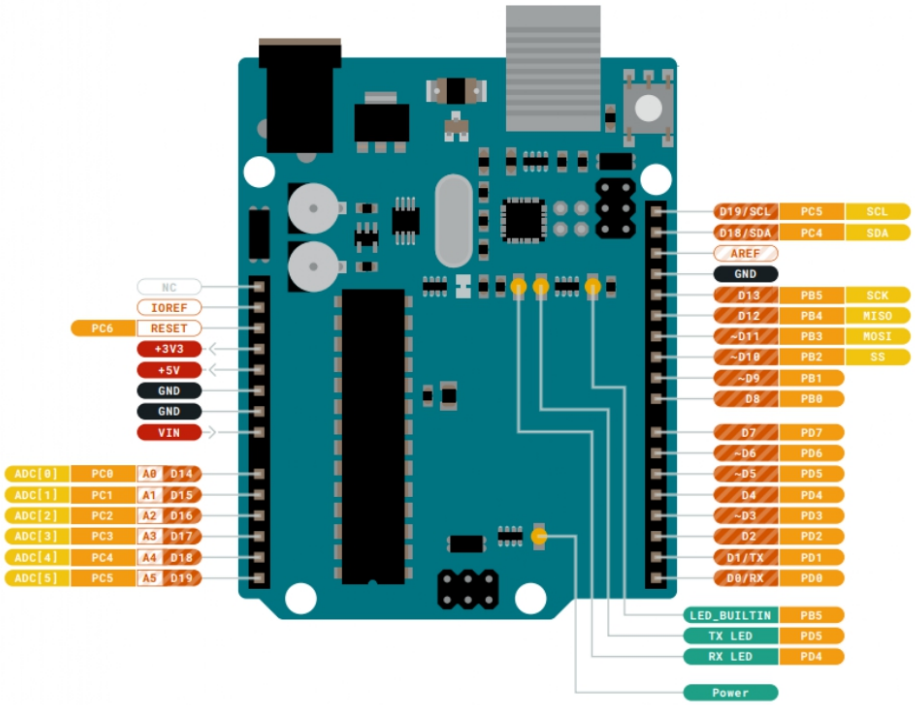
14.3 示例程序

Arduino UNO R3 的示例程序可以在 Arduino IDE 的"示例"菜单或 Arduino 网站 [4] 的"文档"部分找到

14.4 在线资源

现在，您已经了解该电路板的基本功能，就可以通过查看 Arduino Project Hub **[5]**、Arduino Library Reference **[6]** 以及在线 Arduino 商店 **[7]**上的精彩项目来探索它所提供的无限可能性；在这些项目中，您可以为电路板配备传感器、执行器等。

15 连接器引脚布局



布局

15.1 JANALOG

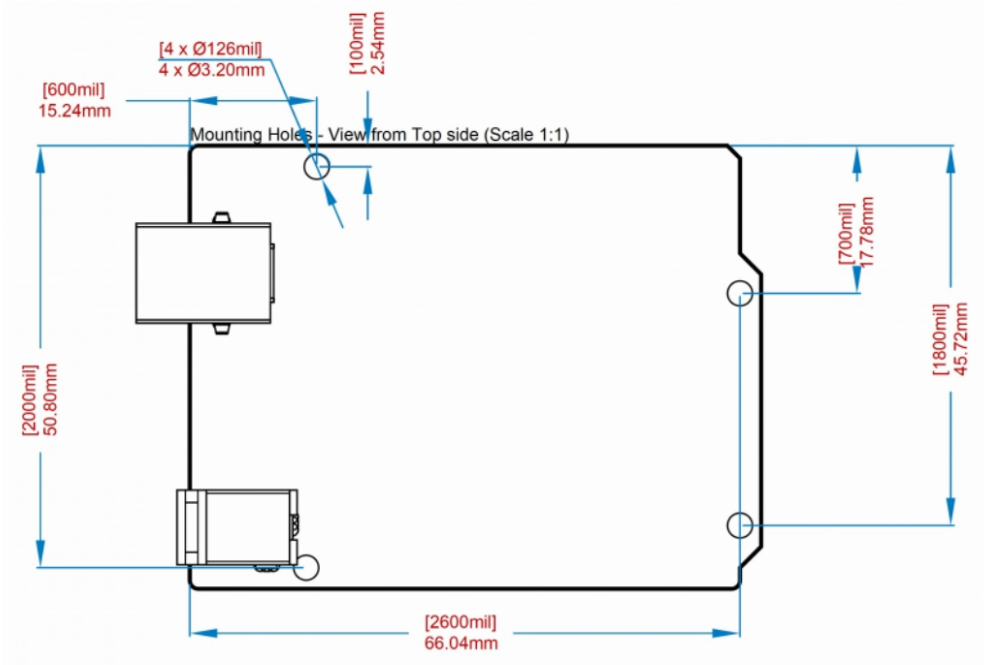
引脚	功能	类型	描述
1	NC	NC	未连接
2	IOREF	IOREF	数字逻辑参考电压 V - 连接至 5V
3	复位	复位	复位
4	+3V3	电源	+3V3 电源轨
5	+5V	电源	+5V 电源轨
6	GND	电源	接地
7	GND	电源	接地
8	VIN	电源	电压输入
9	A0	模拟/GPIO	模拟输入0 / GPIO
10	A1	模拟/GPIO	模拟输入1 / GPIO
11	A2	模拟/GPIO	模拟输入2 / GPIO
12	A3	模拟/GPIO	模拟输入3 / GPIO
13	A4/SDA	模拟输入/I2C	模拟输入 4/I2C 数据线
14	A5/SCL	模拟输入/I2C	模拟输入 5/I2C 时钟线

15.2 JDIGITAL

引脚	功能	类型	描述
1	D0	数字引脚/GPIO	数字引脚 0/GPIO
2	D1	数字引脚/GPIO	数字引脚 1/GPIO
3	D2	数字引脚/GPIO	数字引脚 2/GPIO
4	D3	数字引脚/GPIO	数字引脚 3/GPIO
5	D4	数字引脚/GPIO	数字引脚 4/GPIO
6	D5	数字引脚/GPIO	数字引脚 5/GPIO
7	D6	数字引脚/GPIO	数字引脚 6/GPIO
8	D7	数字引脚/GPIO	数字引脚 7/GPIO
9	D8	数字引脚/GPIO	数字引脚 8/GPIO
10	D9	数字引脚/GPIO	数字引脚 9/GPIO
11	SS	数字	SPI 芯片选择
12	MOSI	数字	SPI1 主输出副输入
13	MISO	数字	SPI 主输入副输出
14	SCK	数字	SPI 串行时钟输出
15	GND	电源	接地
16	AREF	数字	模拟参考电压
17	A4/SD4	数字	模拟输入 4/I2C 数据线（重复）
18	A5/SD5	数字	模拟输入 5/I2C 时钟线（重复）

15.3 机械层信息

15.4 电路板外形图和安装孔



电路板外形图

16 认证

16.1 符合性声明 CE DoC（欧盟）

我们在此郑重声明，上述产品符合以下欧盟指令的基本要求，因此有资格在包括欧盟（EU）和欧洲经济区（EEA）在内的市场内自由流通。

RoHS 2 指令 2011/65/EU	
符合：	EN50581:2012
指令 2014/35/EU。(LVD)	
符合：	EN 60950-1:2006/A11:2009/A1:2010/A12:2011/AC:2011
指令 2004/40/EC & 2008/46/EC & 2013/35/EU, EMF	
符合：	EN 62311:2008

16.2 声明符合欧盟 RoHS 和 REACH 211 01/19/2021

Arduino 电路板符合欧洲议会关于限制在电子电气设备中使用某些有害物质的 RoHS 2 指令 2011/65/EU 和欧盟理事会于 2015 年 6 月 4 日颁布的关于限制在电子电气设备中使用某些有害物质的 RoHS 3 指令 2015/863/EU。

物质	最大限值 (ppm)
铅 (Pb)	1000
镉 (Cd)	100
汞 (Hg)	1000
六价铬（Cr6+）	1000
多溴联苯（PBB）	1000
多溴联苯醚（PBDE）	1000
邻苯二甲酸二(2-乙基己)酯 (DEHP)	1000
邻苯二甲酸丁苄酯 (BBP)	1000
邻苯二甲酸二丁酯（DBP）	1000
邻苯二甲酸二异丁酯（DIBP）	1000

豁免：未申请任何豁免。

Arduino 电路板完全符合欧盟法规 (EC) 1907/2006 中关于化学品注册、评估、许可和限制 (REACH) 的相关要求。我们声明，所有产品（包括包装）中的 SVHC (https://echa.europa.eu/web/guest/candidate-list-table)，（欧洲化学品管理局目前发布的《高度关注物质候选授权清单》）含量总浓度均未超过 0.1%。据我们所知，我们还声明，我们的产品不含 ECHA（欧洲化学品管理局）1907/2006/EC 公布的候选清单附件 XVII 中规定的"授权清单"（REACH 法规附件 XIV）和高度关注物质 (SVHC) 所列的任何物质。

16.3 冲突矿产声明

作为电子和电气元件的全球供应商，Arduino 意识到我们有义务遵守有关冲突矿产的法律法规，特别是《多德-弗兰克华尔街改革与消费者保护法案》第 1502 条。Arduino 不直接采购或加工锡、钽、钨或金等冲突矿物。冲突矿物以焊料的形式或作为金属合金的组成部分存在于我们的产品中。作为我们合理尽职调查的一部分，Arduino 已联系供应链中的元件供应商，以核实他们是否始终遵守法规的相关规定。根据迄今收到的信息，我们声明我们的产品中含有来自非冲突地区的冲突矿物。

17 FCC 警告

任何未经合规性负责方明确批准的更改或修改都可能导致用户无权操作设备。

本设备符合 FCC 规则第 15 部分的规定。操作须满足以下两个条件：

(1) 此设备不会造成有害干扰

(2) 此设备必须接受接收到的任何干扰，包括可能导致不良操作的干扰。

FCC 射频辐射暴露声明:

1. 此发射器不得与任何其他天线或发射器放置在同一位置或同时运行。
2. 此设备符合为非受控环境规定的射频辐射暴露限值。
3. 安装和操作本设备时，辐射源与您的身体之间至少应保持 20 厘米的距离。

English: User manuals for license-exempt radio apparatus shall contain the following or equivalent notice in a conspicuous location in the user manual or alternatively on the device or both. This device complies with Industry Canada license-exempt RSS standard(s). Operation is subject to the following two conditions:

(1) this device may not cause interference

(2) this device must accept any interference, including interference that may cause undesired operation of the device.

French: Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes :

(1) l' appareil nedoit pas produire de brouillage

(2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

IC SAR警告:

English This equipment should be installed and operated with a minimum distance of 20 cm between the radiator and your body.

French: Lors de l' installation et de l' exploitation de ce dispositif, la distance entre le radiateur et le corps est d 'au moins 20 cm.

重要提示： EUT 的工作温度不能超过 85°C，也不能低于 -40°C。

Arduino S.r.l. 特此声明，本产品符合 2014/53/EU 指令的基本要求和其他相关规定。本产品允许在所有欧盟成员国使用。

18 公司信息

公司名称	Arduino S.r.l
公司地址	Via Andrea Appiani 25 20900 MONZA Italy

19 参考资料

参考资料	链接
Arduino IDE (Desktop)	https://www.arduino.cc/en/Main/Software
Arduino IDE (Cloud)	https://create.arduino.cc/editor
Cloud IDE 入门指南	https://create.arduino.cc/projecthub/Arduino_Genuino/getting-started-with-arduino-web-editor-4b3e4a
Arduino 网站	https://www.arduino.cc/
Arduino Project Hub	https://create.arduino.cc/projecthub?by=part&part_id=11332&sort=trending
库参考	https://www.arduino.cc/reference/en/
在线商店	https://store.arduino.cc/

20 修订记录

日期	版次	变更
2023/07/26	2	一般更新
2021/06	1	数据表发布

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3, Trimester 3	**Study week no.: 1**
Student Name & ID: Joey Lok Jo Wei	
Supervisor: Dr. Muhammad Syaiful Amri bin Suhaimi	
Project Title: Arduino Based IoT Gardening System	

1. WORK DONE

Look for place to set up the system.
Gather materials to build the system.

2. WORK TO BE DONE

Source for the materials and components needed.

3. PROBLEMS ENCOUNTERED

Components sold in physical stores are pricey.

4. SELF EVALUATION OF THE PROGRESS

Spent too much time on sourcing for materials.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3, Trimester 3	**Study week no.: 2**
Student Name & ID: Joey Lok Jo Wei	
Supervisor: Dr. Muhammad Syaiful Amri bin Suhaimi	
Project Title: Arduino Based IoT Gardening System	

1. WORK DONE

Build the system outside the unit (partially indoor).

2. WORK TO BE DONE

Borrow wires from the lab as the sensors are far from the breadboard and Arduino.

3. PROBLEMS ENCOUNTERED

Some sensors malfunctioned and rusted.

4. SELF EVALUATION OF THE PROGRESS

Components need to be kept nicely.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3, Trimester 3	**Study week no.: 3**
Student Name & ID: Joey Lok Jo Wei	
Supervisor: Dr. Muhammad Syaiful Amri bin Suhaimi	
Project Title: Arduino Based IoT Gardening System	

1. WORK DONE

Starts to program the new sensors and actuators using Arduino IDE.

2. WORK TO BE DONE

Try to log the sensor readings into the SD card using the SD card module.

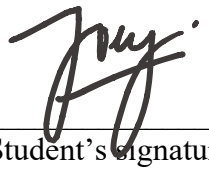
3. PROBLEMS ENCOUNTERED

SD card module runs perfectly in simple test code, but not working when integrated into the system.

4. SELF EVALUATION OF THE PROGRESS

Unable to find out the issue causing the SD card module to not work when integrated.

Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3, Trimester 3	**Study week no.: 4**
Student Name & ID: Joey Lok Jo Wei	
Supervisor: Dr. Muhammad Syaiful Amri bin Suhaimi	
Project Title: Arduino Based IoT Gardening System	

1. WORK DONE

Soldered the wires of batteries holders with the actuator's wires into the relay board.

2. WORK TO BE DONE

Continue to fix the SD card issue.
Test the actuators connections.

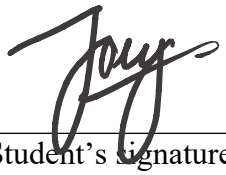
3. PROBLEMS ENCOUNTERED

The full spectrum LED light looks dim, 9V batteries may not be enough for it to work.
SD card issue still not solved.

4. SELF EVALUATION OF THE PROGRESS

Should find other solutions instead of spending too much time fixing the same issue.

Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3, Trimester 3	**Study week no.: 5**
Student Name & ID: Joey Lok Jo Wei	
Supervisor: Dr. Muhammad Syaiful Amri bin Suhaimi	
Project Title: Arduino Based IoT Gardening System	

1. WORK DONE

Gave up on the SD card module method, sourced for solution on the Internet.

2. WORK TO BE DONE

Find a new substitute to replace the SD card module function.

3. PROBLEMS ENCOUNTERED

Solutions found are over budgeted, need to ask for more recommendations from some forums.

4. SELF EVALUATION OF THE PROGRESS

Need to have plan B in every method.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3, Trimester 3	**Study week no.: 6**
Student Name & ID: Joey Lok Jo Wei	
Supervisor: Dr. Muhammad Syaiful Amri bin Suhaimi	
Project Title: Arduino Based IoT Gardening System	

1. WORK DONE

Found a solution to substitute the SD card module, which is using Python Script to parse and process the data into graphs and send them to telegram directly.

2. WORK TO BE DONE

Code the python script.

3. PROBLEMS ENCOUNTERED

Low level of Python knowledge, need time to code.

4. SELF EVALUATION OF THE PROGRESS

Need to spend more time in learning other programming languages during free time.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3, Trimester 3	**Study week no.: 7**
Student Name & ID: Joey Lok Jo Wei	
Supervisor: Dr. Muhammad Syaiful Amri bin Suhaimi	
Project Title: Arduino Based IoT Gardening System	

1. WORK DONE

The system is set up.
Tested all the sensors and actuators.

2. WORK TO BE DONE

Continue to code in Python.

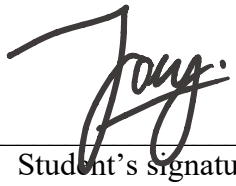
3. PROBLEMS ENCOUNTERED

No problem faced.

4. SELF EVALUATION OF THE PROGRESS

The soil moisture sensor shouldn't be left in the moist soil, it rusted.

Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3, Trimester 3	**Study week no.: 8**
Student Name & ID: Joey Lok Jo Wei	
Supervisor: Dr. Muhammad Syaiful Amri bin Suhaimi	
Project Title: Arduino Based IoT Gardening System	

1. WORK DONE

Completed the python code.

2. WORK TO BE DONE

Test run the python code and create a Telegram Bot to send graphs to user.

3. PROBLEMS ENCOUNTERED

Have issue looking for the Chat ID of the telegram bot.

4. SELF EVALUATION OF THE PROGRESS

Need to schedule my time nicely during midterm weeks.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3, Trimester 3	**Study week no.: 9**
Student Name & ID: Joey Lok Jo Wei	
Supervisor: Dr. Muhammad Syaiful Amri bin Suhaimi	
Project Title: Arduino Based IoT Gardening System	

1. WORK DONE

Telegram Bot set up completed. Token and chat ID obtained.

2. WORK TO BE DONE

Test run the python code.

3. PROBLEMS ENCOUNTERED

No problem faced.

4. SELF EVALUATION OF THE PROGRESS

Should test the Python Script earlier because there will be bugs to be fix.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3, Trimester 3	**Study week no.: 10**
Student Name & ID: Joey Lok Jo Wei	
Supervisor: Dr. Muhammad Syaiful Amri bin Suhaimi	
Project Title: Arduino Based IoT Gardening System	

1. WORK DONE

Tested to run the Python Script.

2. WORK TO BE DONE

Ready for the experiment and test run.

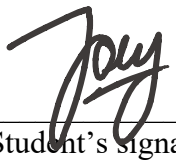
3. PROBLEMS ENCOUNTERED

Python Script appears to have many bugs, such as graphs with no data, error parsing serial output from the Arduino and issue when sending images to telegram bot.

4. SELF EVALUATION OF THE PROGRESS

Need more time to fix the issues one by one.

Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3, Trimester 3	**Study week no.: 11**
Student Name & ID: Joey Lok Jo Wei	
Supervisor: Dr. Muhammad Syaiful Amri bin Suhaimi	
Project Title: Arduino Based IoT Gardening System	

1. WORK DONE

Fixed some issue with the graph plotting and parsing of data from serial monitor.

2. WORK TO BE DONE

Make sure its error free to test run for one week.

3. PROBLEMS ENCOUNTERED

Some components rusted and malfunctioned.

4. SELF EVALUATION OF THE PROGRESS

Should seek advice and help from some seniors.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3, Trimester 3	**Study week no.: 12**
Student Name & ID: Joey Lok Jo Wei	
Supervisor: Dr. Muhammad Syaiful Amri bin Suhaimi	
Project Title: Arduino Based IoT Gardening System	

1. WORK DONE

Python script errors and bugs fixed, tested to run for a week.

2. WORK TO BE DONE

Evaluate the readings and output, ensuring the consistency of the output.
Complete the report and presentation preparations.

3. PROBLEMS ENCOUNTERED

Need to put the laptop near the Arduino so that Python script can be run.

4. SELF EVALUATION OF THE PROGRESS

Should have get a longer USB cable to connect the laptop with Arduino UNO.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3, Trimester 3	**Study week no.: 13**
Student Name & ID: Joey Lok Jo Wei	
Supervisor: Dr. Muhammad Syaiful Amri bin Suhaimi	
Project Title: Arduino Based IoT Gardening System	

1. WORK DONE

Completed the report.

2. WORK TO BE DONE

Submit report and prepare for presentation.

3. PROBLEMS ENCOUNTERED

No problem faced.

4. SELF EVALUATION OF THE PROGRESS

N/A

Supervisor's signature

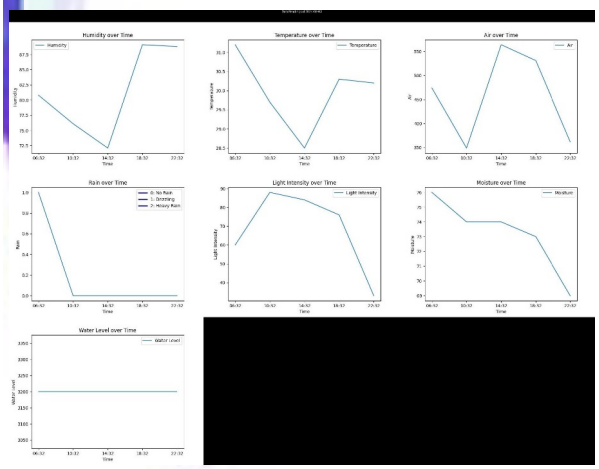
Student's signature

ARDUINO based IoT Gardening System

UTAR
UNIVERSITI TUNKU ABDUL RAHMAN
Wholly owned by UTAR Education Foundation
(Co. No. 578227-M)
DU012(A)

INTRODUCTION

This project will develop an automated gardening system and send the surrounding data to the user via Telegram.



METHODOLOGY

The system consist of three main components, software, hardware and cloud.

Hardware includes Arduino UNO, sensors and actuators.

Software includes Python Script and Arduino IDE.

Telegram as the cloud platform to deliver graphs to the users.

RESULTS OF EXPERIMENT

	With Automated System	Without Automated System
New leaves	3	1
Leaves color	Fresh green	Yellowish green

WHY CHOOSE THIS SYSTEM?

A series of sensors to read surrounding data into graphical form.

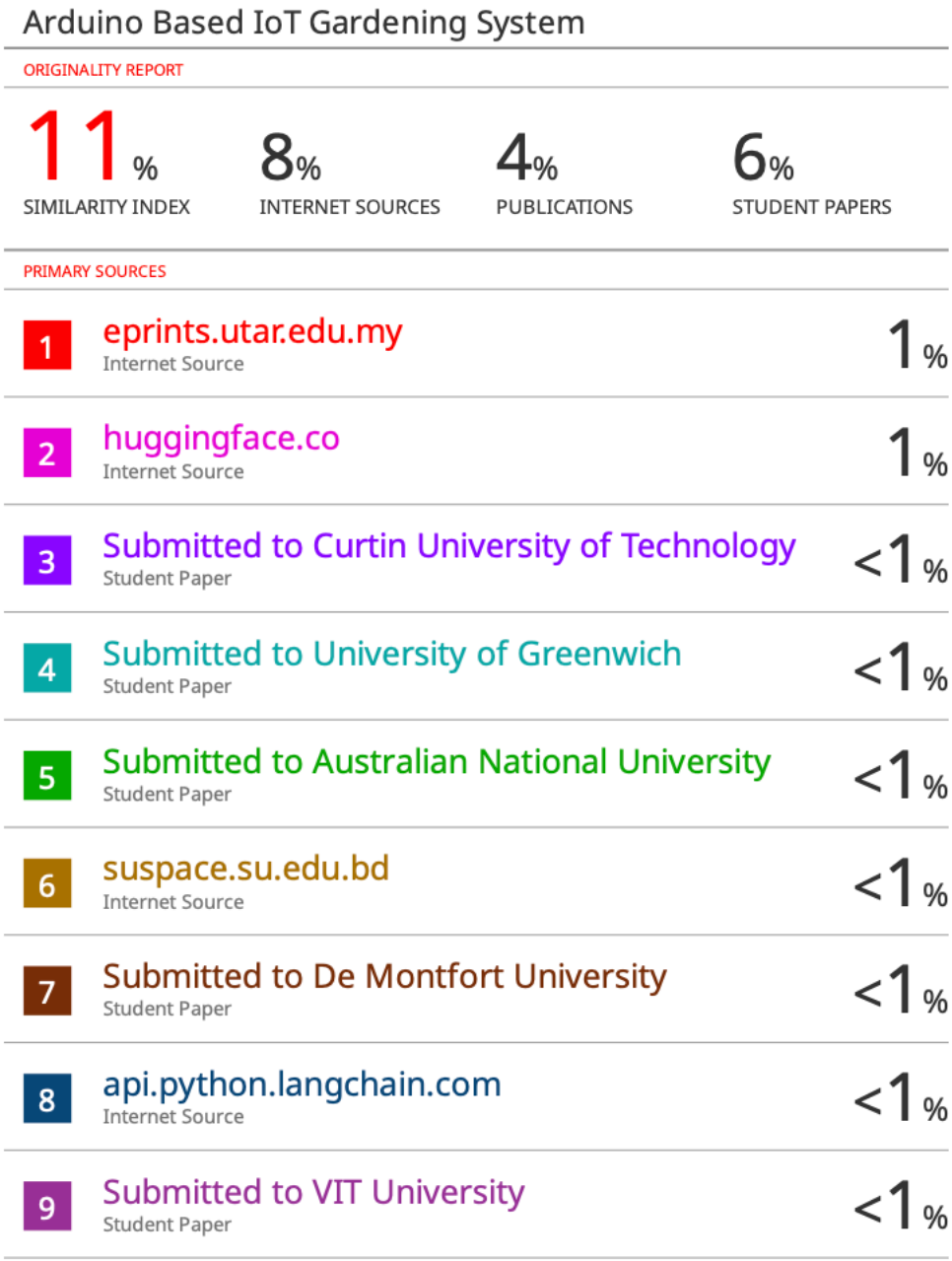
Automated gardening: auto watering, turning on LED and fan.

Efficient use of resources: Actuators only runs when threshold is reached.

Tested to prove improvement in plant growth after implementation!

Proposed by: Joey Lok Jo Wei
Supervised by: Dr. Muhammad Syaiful Amri bin Suhaimi

PLAGIARISM CHECK RESULT



Arduino Based IoT Gardening System

ORIGINALITY REPORT

11%	8%	4%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	eprints.utar.edu.my Internet Source	1%
2	huggingface.co Internet Source	1%
3	Submitted to Curtin University of Technology Student Paper	<1%
4	Submitted to University of Greenwich Student Paper	<1%
5	Submitted to Australian National University Student Paper	<1%
6	suspace.su.edu.bd Internet Source	<1%
7	Submitted to De Montfort University Student Paper	<1%
8	api.python.langchain.com Internet Source	<1%
9	Submitted to VIT University Student Paper	<1%

10	Durgesh Kumar Mishra, Nilanjan Dey, Bharat Singh Deora, Amit Joshi. "ICT for Competitive Strategies", CRC Press, 2020 Publication	<1 %
11	Submitted to Napier University Student Paper	<1 %
12	Submitted to Universiti Tunku Abdul Rahman Student Paper	<1 %
13	Submitted to University of Edinburgh Student Paper	<1 %
14	Submitted to Al Ain University of Science and Technology Student Paper	<1 %
15	gist.github.com Internet Source	<1 %
16	Gauri R Choudhari, Pratik A Dagale, Isha S Dashetwar, Rutuja R Desai, Abha A Marathe. "IoT-based Smart Gardening System", Journal of Physics: Conference Series, 2023 Publication	<1 %
17	Soraya Norma Mustika, Muladi, Anik Nur Handayani, Muhammad Afnan Habibi, Zein Farhan Makarim, Eko Noerhayati. "Smart Garden Monstera Adansonii Based On IoT Using DHT11", 2022 3rd International	<1 %

	Conference on Electrical Engineering and Informatics (ICon EEI), 2022 Publication	
18	Submitted to University of Northampton Student Paper	<1 %
19	Submitted to Kean University Student Paper	<1 %
20	Submitted to Escuela Politecnica Nacional Student Paper	<1 %
21	Submitted to Florida Atlantic University Student Paper	<1 %
22	S Y Chia, M W Lim. "A critical review on the influence of humidity for plant growth forecasting", IOP Conference Series: Materials Science and Engineering, 2022 Publication	<1 %
23	aurora-asi-lib.readthedocs.io Internet Source	<1 %
24	pypi.org Internet Source	<1 %
25	Submitted to International School of the Stockholm Region Student Paper	<1 %
26	Submitted to Montana State University, Bozeman Student Paper	<1 %

27	Yogesh. "Programming and Interfacing with Arduino", CRC Press, 2021 Publication	<1 %
28	Submitted to South Bank University Student Paper	<1 %
29	create.arduino.cc Internet Source	<1 %
30	community.element14.com Internet Source	<1 %
31	forum.arduino.cc Internet Source	<1 %
32	programtalk.com Internet Source	<1 %
33	dokumen.pub Internet Source	<1 %
34	www.bio-conferences.org Internet Source	<1 %
35	www.slideshare.net Internet Source	<1 %
36	Sudha Senthilkumar, Meghena Danasekar, Brindha K.. "chapter 15 IoT-Based Smart Gardening System Using the Cloud", IGI Global, 2022 Publication	<1 %

37	Pawan Singh Mehra, Dhirendra Kumar Shukla. "Artificial Intelligence, Blockchain, Computing and Security - Volume 2", CRC Press, 2023 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	JOEY LOK JO WEI
ID Number(s)	20ACB05187
Programme / Course	BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) COMPUTER ENGINEERING
Title of Final Year Project	ARDUINO BASED IOT GARDENING SYSTEM

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 11% Similarity **by source** Internet Sources: 8% Publications: 4% Student Papers: 6%	
Number of individual sources listed of more than 3% similarity: **N/A**	
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor

Name: Dr. Muhammad Syaiful Amri bin Suhaimi

Date: 9 September 2024

Signature of Co-Supervisor

Name:

Date:

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY
(KAMPAR CAMPUS)

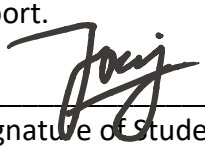
CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	20ACB05187
Student Name	JOEY LOK JO WEI
Supervisor Name	Dr. Muhammad Syaiful Amri bin Suhaimi

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.



(Signature of Student)
Date: 9 September 2024