Planta-Arca

IoT Based Smart Plant Monitoring System

Nazlı Güler
Department of Computer Science
Binghamton University
nguler1@binghamton.edu

Mustafa İkbal Mehmedoğlu
Department of Computer Science
Binghamton University
mmehmed1@binghamton.edu

ABSTRACT

Plants play a vital role in maintaining the ecological cycle. To maintain a plant's proper growth and health, some external factors must be tracked constantly. In this project, Planta-Arca is able to monitor light, humidity, temperature and soil moisture levels of plants with the provided sensor and the measured data can be viewed from the mobile app by users.

KEYWORDS
Internet of Things, smart plant monitoring, Arduino, mobile app, ESP32D

1. Introduction

Our main goal is to provide the ideal conditions to plants. To implement Planta-Arca we designed a smart plant monitoring module which contains various sensors to monitor light, humidity, temperature and soil moisture levels of plants growing in that module.

This system also encompasses a mobile application through which users can remotely monitor the measured sensor values and on/off lights, water pump and fan by pressing the buttons.

2.Design

2.1 Hardware Components

In Figure 1, Planta-Arca module is shown.

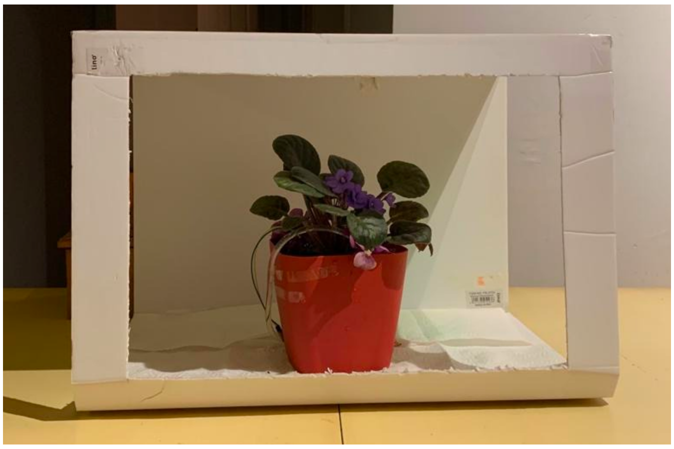

Figure 1: Planta-Arca Module

In Figure 2, all the hardware components used to monitor and control the plant environment is shown.

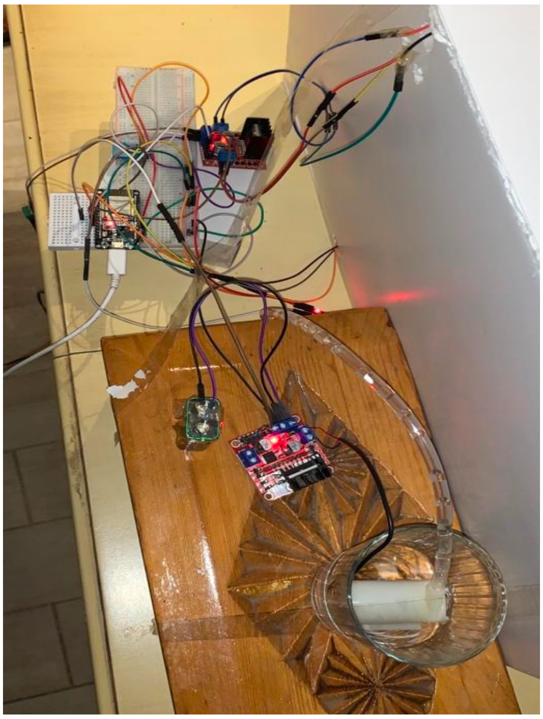

Figure 2: Microcontrollers and sensors used

2.1.1 ESP Wroom 32D

It is a powerful microcontroller which is also Wi-Fi and Bluetooth capable. We started this project with the idea that we can use an Arduino UNO with the ESP 8266 Wi-Fi module, but even though we made sure that our wiring was correct, the module would not work, therefore we went with the ESP32D option, which is again, much powerful and easier to use compared to Arduino UNO.

2.1.2 DHT11 Digital Humidity and Temperature Sensor

DHT11 is one of the most common digital temperature and humidity sensor, used in many projects. We read the corresponding values with this sensor and send them to our gauges in the mobile app.

2.1.3 YL-69 Soil Moisture Sensor

Again, one of the most common and preferable sensor of its kind, YL-69 is capable of sending HIGH and LOW values according to the situation. We use it by sticking both ends of the sensor to the soil and send values to a led in our mobile app.

2.1.4 LM393 Photosensitive Light Dependent Control Sensor

This is a basic control sensor that is able to determine if the area around the sensor is lit. It is not as sensitive as it can be expected from a sensor, it is not certain to read a LOW value when it is not pitch dark or a HIGH value when the environment is slightly lit.

2.1.5 Custom Build Leds

We soldered jumper cables and 3 led lights with 12V inputs to create this motor.

2.1.6 Marxlow Fan

To cool the area around the plant, we can send a signal to the fan, it is used in the same motor driver with the leds since they are both 12V.

2.1.7 Brushless Motor Submersible Water Pump

When the output of soil moisture is LOW or when a user input is entered, this motor pumps water to the plant’s soil. This motor requires and accepts 6V input therefore has its own motor driver, separate from the other two motors.

2.1.8 L298N Motor Driver

We have two of these motors that control our motors. Since we cannot supply 12 and 6 Volt motors with the Arduino itself, we needed to get these motors wired to the motors with an external power source, an adapter that outputs 12V power.

2.2 Software Components

The system basically evolved around the Arduino (ESP32D) and Blnyk software. We coded the Arduino code from scratch using the examples in the Blynk website. Before changing to the ESP32D board, we used C++ and python in addition to the android studio to edit a template mobile app, but that work ended when we started using ESP32D.

2.2.1 Arduino IDE

Within Arduino IDE, there are a lot of examples, and when you start using external libraries, some of them add their own sample codes in addition to those. Since we were both inexperienced with the Arduino, we mostly used those examples, then derived our own code.

2.2.2 Blynk

For non-commercial use, Blynk allows users to use an authorization code to match the mobile app with the Wi-Fi capable boards. We have implemented our Arduino code concentric with the Blynk codes. To learn about Blynk's capabilities, we went through their examples and a tutorial on how to use it. We also set our layout on the Planta-Arca module at the Blynk app by examining the example layouts.

3. Implementation

3.1 Data Flow

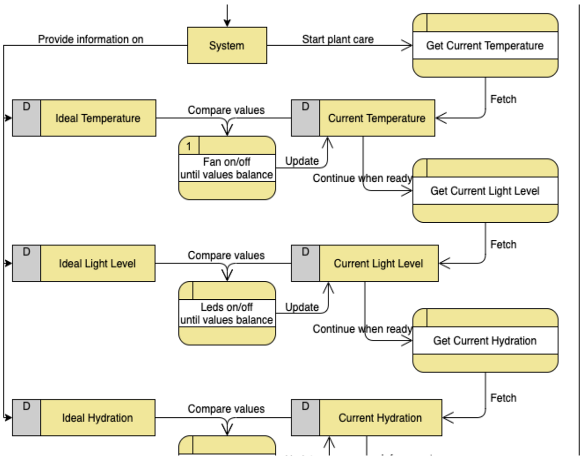

Figure 3: Data Flow Diagram

3.2 Mobile App

The Planta-Arca app shows the real-time information gathered from the module. It displays the humidity and temperature values and users can remotely on/off lights, water pump and fan by pressing the buttons. This allows users to monitor their plant status and health.

3.3 Module Control

The module around the microcontroller is controlled by the ESP32D itself and to read the values from the board, Blynk app can be used. The app lets us read or check all the values from the sensors and to control motors. Originally, we thought to have this system automated, and that is totally achievable by adjusting the board to the automatic Arduino sketch as well.

4. Conclusion

We successfully achieved the functionality as designed:

1. Read the environmental data of;

- Humidity
- Temperature
- Moisture
- Light

around the plant using the sensors.

2. Control the corresponding motors to those values through the mobile app.

3. Created a terminal to also control or check any given input, such as typing;

- "H" or "Humidity",
- "T" or "Temperature",
- "Open *motor name*",
- "Close *motor name*",
- Or "open-close motors",

to the terminal achieve the wanted action.

We started off this project with the idea that by modularly controlling one plant, then extending it to many plants in a certain environment such as a greenhouse, we could achieve a smart, monitored greenhouse. We now know that it is possible and actually really simple using these components and the software used in this project.

5. References

[1] Sadasivam, Sreeram & Vadhri, Viswanath & Ramesh, Supradha. (2015). Smart Plant Monitoring System. 10.13140/RG.2.1.2915.6563/1.

[2] Elangovan, Ramkumar & Santhannakrishnan, Nagarani & Rozario, Roger & Banu, Arjuman. (2018). Tomen:A Plant monitoring and smart gardening system using IoT. International Journal of Pure and Applied Mathematics. Volume 119.

[3] Kohli, Ankur & Kohli, Rohit & Singh, Bhupendra. (2019). Smart Plant Monitoring System using IoT Technology. 10.4018/978-1-5225-9574-8.ch016.