Plant Monitoring Systems- an IoT-based Project

Lecturer in-charge: Prof. Ali Hayek
Contact email: Ali.Hayek@hshl.de

Nnaemeka Valentine Eze **Onyesi John Abiagam**

1 Abstract

The Internet of Things (IoT) refers to a network of physical objects embedded with sensors, software, and other technologies to connect and exchange data with other devices and systems over the internet.

2 Introduction

Internet of Things (IoT) refers to a network of physical objects embedded with sensors, software, and other technologies to connect and exchange data with other devices and systems over the internet.

The goal is to build a smart farm that is capable of controlling and monitoring its surroundings to maximize plant development. The farm should be able to monitor the PH level, maintain appropriate soil humidity levels, and modify its moisture content in accordance with the current conditions.

In other to accomplish this therefore, the Smart farm will have the appropriate sensors and actuators installed, and it will be linked to a central server via the MQTT protocol to provide real-time control and monitoring.

2.1 Motivation

2.2 System Overview

The goal of this Internet of Things system is to use sensor data to automate watering and monitor soil conditions. The pH and moisture content of the soil can be monitored by the system using an Arduino-based microcontroller coupled with a humidity and pH sensor. The water pump is managed by a servo motor that communicates to a Raspberry Pi by a broker network.

Upload the images showing how different sample rates affect the wave

3 Concept Draft

The Sensor node: The sensor network is made up of;

DFRobot Gravity PH sensor: It is an analog sensor with Analog to Digital Converter(ADC) connected to a probe. The probe is inserted into a solution, in other to read the PH value. The signal pin is connected to A0 pin of the Arduino wifi-rev2, as a client. It needs between 3.3 to 5.5 volt input, with dedicated library.

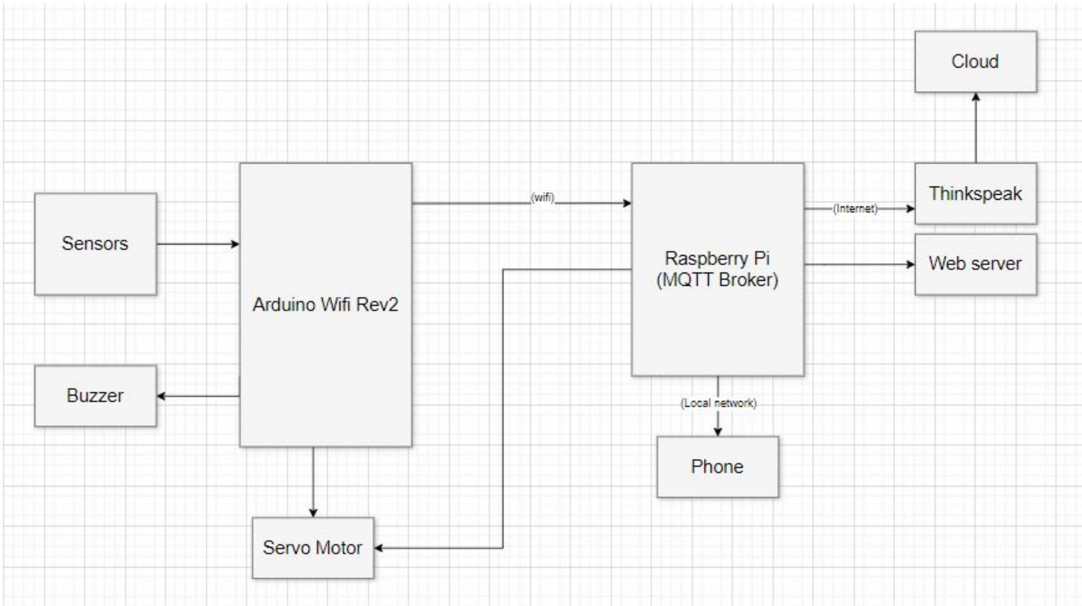


Figure 1: Concept Design

Moisture sensor: The simple resistive moisture sensor when immeresed in the soil, measures the percentage moisture level. It accomplishes this by analyzing the electrical resistance between two probes inserted into the soil. The resistance changes as a function of soil moisture content.

Dry soil exhibits high resistance between probes due to poor electrical conductivity, while wet soil exhibits low resistance due to water's ability to facilitate current flow.

Hence, the Arduino reads the voltage signal that the sensor produces after measuring resistance. The moisture content of the soil can be estimated by the microcontroller by interpreting this signal as shown in the code below.

Actuator (servo): The servo motor tied to the water pump is operated by the Raspberry Pi. When necessary, the servo motor receives orders from the sensor to pump water, ensuring that the soil moisture content is optimum.

A servo is an electric motor that can accurately regulate torque, speed, and angular or linear position . The control signal is commonly a pulse width modulation (PWM) signal, which indicates the intended position of the servo motor's output shaft. The DC or AC motor spins in the appropriate direction based on the feedback system's reactive signal.

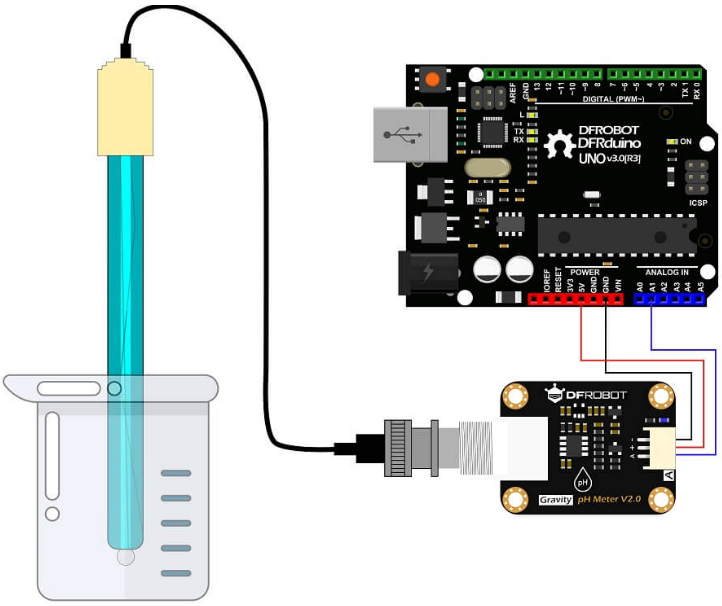
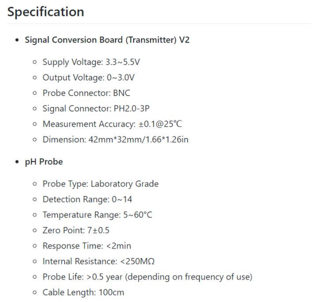


Figure 2: PH Sensor



Specification

- **Signal Conversion Board (Transmitter) V2**
  - Supply Voltage: 3.3~5.5V
  - Output Voltage: 0~3.0V
  - Probe Connector: BNC
  - Signal Connector: PH2.0-3P
  - Measurement Accuracy: ±0.1@25℃
  - Dimension: 42mm*32mm/1.66*1.26in
- **pH Probe**
  - Probe Type: Laboratory Grade
  - Detection Range: 0~14
  - Temperature Range: 5~60°C
  - Zero Point: 7±0.5
  - Response Time: <2min
  - Internal Resistance: <250MΩ
  - Probe Life: >0.5 year (depending on frequency of use)
  - Cable Length: 100cm

Figure 3: PH Sensor Datasheet

4 Project/Team management

The initial project was organised using a top-down approach, in other to conceptualise the main idea, during the initial lab series. We further iteratively accelerated the project employing a scrum model. Each week, our sprint is aimed at breanstorming on present task and getting it done as soon. The overall team progress is reassessed after each sprint, and documented by the project lead. The overall team lead is structured, but the lead on each weekly meeting is varied according to the sprint task. This is because, each member takes on a mini-lead role in certain task such that he research deeply in while we all meet and brainstorm on them before the next sprint. In this case, he leads the session, while others colaborate.

Nnaemeka Valentine Eze - Arduino sensor nodes and Cloud services

Onyesi John Abiagam - Raspberry Pi(MQTT) Dashboard and Database

5 Technologies

There are several technologies central to the overall project. These may include;

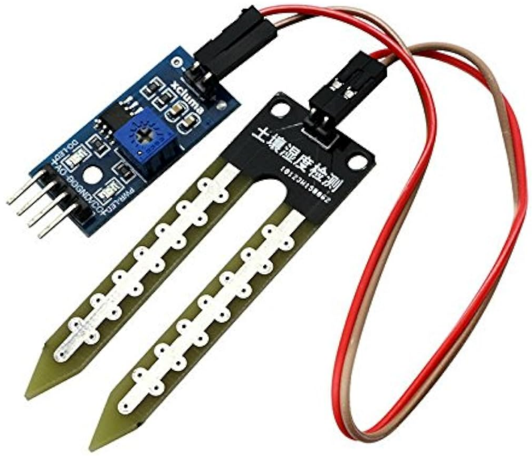


Figure 4: Moisture Sensor

Figure 5: Servo Motor

Wireless Sensor network A wireless sensor network is made up of a number of separate, autonomous sensors, or "nodes," that link to one another wirelessly.

Communication via wireless networks is a fundamental component of a WSN. In this setup, sensor data is sent electronically from the Arduino to the Raspberry Pi via a Wi-Fi module (the Arduino Wifi-Rev2). This wireless data transmission is made possible via the MQTT protocol, which enables real-time communication between the Raspberry Pi broker and the sensor nodes, which are pH and humidity sensors connected to the Arduino wifi.

PH Sensor Node: Reports the soil's pH level.

Humidity Sensor Node: Tracks the percentage of moisture in the soil.

Arduino Arduino is an open-source electronics platform built on user-friendly hardware and software. It is intended to increase everyone's accessibility to electronics, regardless of skill level. The platform includes integrated development environments (IDEs) and microcontroller boards, which are used to write and upload code to the boards[1]. The codes for the sensor nodes were written in C on Arduino 2 IDE environment which is compatible with Arduino wifi.

Python Python scripts were written on the Raspberry pi in other to establish effective communication in the system. **MQTT** MQTT(Message Queuing Telemetry Transport—) is a lightweight messaging protocol suitable for networks with unstable or low bandwidth and high latency. Its effectiveness and simplicity of use make it a popular choice for Internet of Things applications. Its key function in the project;

As a broker; it stands as central server that distributes messages. For instance, publishers provide messages to it, and it forwards them to subscribers and vice versa. PH ans moisture level are both sent as published messages(PHTopic and MoistureTopic) to subscribers. The detailed flow is shown on the codes and github. The connection(publishers/subscribers) is made possible with TCP/IP. We connected the broker via its ip address (172.20.10.10) unto the mqtt server, through port 1883.

Raspberry Pi

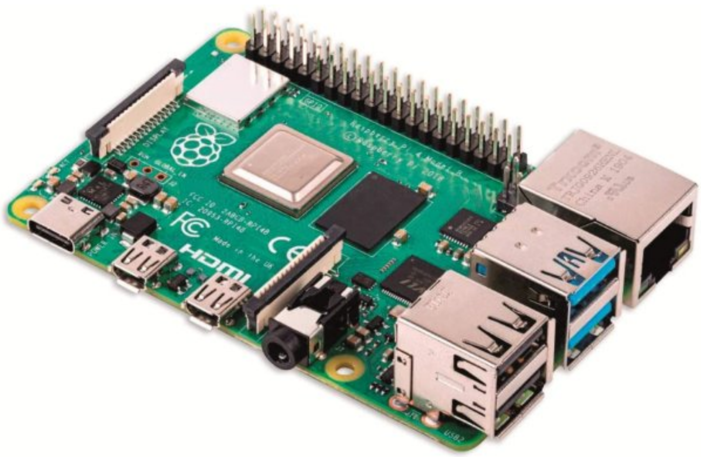


Figure 6: Raspberry pi
[4]

6 Implementation

6.1 Connect Sensors to Arduino:

- Soil Moisture Sensor: Connect the sensor's output pin to one of the analog input pins on the Arduino (e.g., A0).
- pH Sensor: Connect the pH sensor's output pin to another analog input pin on the Arduino (e.g., A1).
- Ensure the sensors are powered correctly according to their specifications.

6.2 Connect Servo Motor:

- Connect the control pin of the servo motor to a digital pin on the Arduino (e.g., D9).
- Connect the power and ground pins of the servo to the Arduino's power and ground. .

6.3 Raspberry Pi Setup:

- Make sure your Raspberry Pi is running the latest version of Raspbian OS.
- Connect the Raspberry Pi to the internet (via Ethernet or Wi-Fi).
- Open a terminal on the Raspberry Pi

6.4 Install Mosquitto MQTT Broker:

- Install Mosquitto: In the terminal, install the Mosquitto MQTT broker by running: sudo apt-get install mosquitto mosquitto-clients

6.5 Set Up MQTT Client on Raspberry Pi:

- Create a Python Script: Create a Python script on the Raspberry Pi to subscribe to the sensor data and control the servo motor.

.

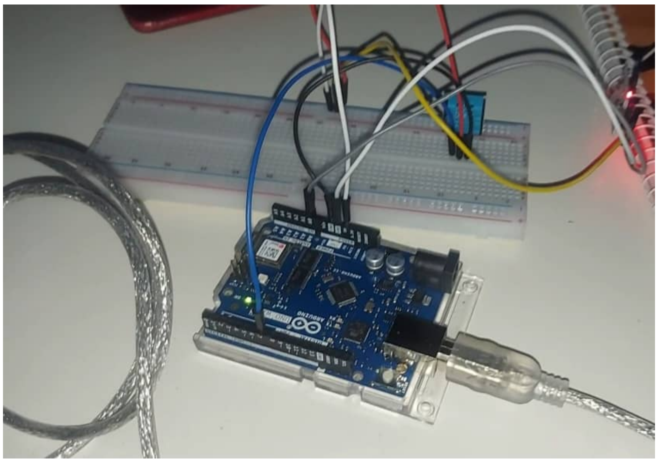


Figure 7: Arduino connections

Client and access points
Database
Broker: The broker netWork was realised in other to establish a wireless communication between the clients and server. It therefore transmits data between them as shown below. The result is equipped with an external screen designed for data visualisation dashboard.

Cloud Services While we could read the corresponding data outome on the dashboard, we could equally visualise it remotely via cloud service. There are several cloud services, but we chose Blynk over Thinkspeak, that was initially considered. THis is due to its user-friendly mobile envirenment, cross-platform support, as well as relevant community support[2]. It offers connection using MQTT and POST protocols. We need the credentials of the Thinkspeak channel to connect. The credentials include API Write and Read keys, MQTT username and password, as well as channel ID.

7 Use Case

7.1 Getting Started

- Put the pH and soil moisture sensors in the soil around the plant as shown in fig 8.
- Switch on the Raspberry Pi and both of the Arduino WiFi boards.
- To begin gathering data, run the Python script on the Raspberry Pi.

Figure 8: Setup
[3]

7.2 Interacting with the Dashboards

7.2.1 Local MQTT Panel Dashboard

- Connect your device (computer/tablet) to the same local network as the Raspberry Pi
- Open a web browser and navigate to the MQTT panel URL (e.g., http://raspberrypi-ip:1880/ui)
- You'll see real-time data from all sensors displayed on the local dashboard as shown in fig 9

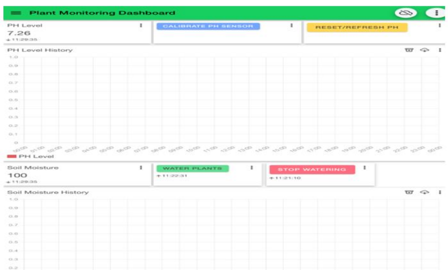


Figure 9: MQTT Dashboard

7.2.2 Blynk Cloud Dashboard

- Open the Blynk app on your smartphone (available anywhere there's an internet connection).
- To access the Plant Monitoring System, enter the Auth Token associated with your project.
- View the pH sensor level and soil moisture widgets as shown in fig 10.

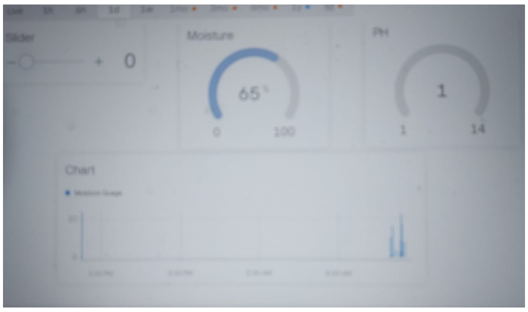


Figure 10: Blynk Dashboard

7.3 Example Scenarios

7.3.1 Check Plants Every Day (Local Network)

- Launch your local device's MQTT panel dashboard.
- Verify the reading for soil moisture.
  - If greater than70 percent, make sure adequate drainage.
  - If greater than70 percent, make sure adequate drainage.
- Examine the pH Level
  - Aim for 6.0–7.0 for most plants.
  - If outside of this range, modify the water or soil as necessary.

7.3.2 Remote Monitoring (Using Blynk)

- Open the Blynk app when away from home
- Check the Soil Moisture reading:
  - If below 30 percent, water the plant
  - If above 70 percent,no action is needed
- For important notifications, use Blynk's notification feature.

7.3.3 Responding to Alerts

- Configure alerts for crucial situations in the MQTT panel and Blynk.
- As soon as one gets a warning:
  - Check the MQTT panel for real-time information if someone is around the house
  - If absent, monitor the plant using the Blynk app.
  - Take the necessary steps or get in touch with someone who can

7.3.4 Weekly Plant Health Assessment

- To understand the plant situation, use the historical data graphs on the MQTT panel.
- We check for any inconsistencies with the data that Blynk has stored in the cloud.
- Based on these observations, we try to modify our parameters accordingly.

7.4 Troubleshooting

- If MQTT panel isn't updating:
  - Verify that the mqtt panel is linked to the local network.
  - Verify Raspberry Pi is powered on and connected
  - Make sure the MQTT broker is operational by executing the sudo systemctl status mosquitto command from the terminal.
  - If required, restart Mqtt (sudo systemctl restart).
- if the Blynk dashboard fail to update:
  - if the Blynk dashboard fail to update:
  - Check to ensure that the Raspberry Pi is connected to the internet.
  - Ensure that the Python script connecting Blynk to MQTT is active.
- In the case where neither dashboard is updated:
  - Verify that the LED indicators are on and the Arduino boards are powered on.
  - Ensure that the Arduino boards are switched on and that the LED indications are lit up.

7.5 Maintenance

- Every month, use calibration solutions to calibrate the pH sensor.
- Every two weeks, clean the soil moisture sensor probes.
- If necessary, check and replace the batteries in the Arduino boards.
- If the Raspberry Pi's data storage is becoming full, prune it.
- Frequently review and update the widgets and project settings for Blynk.

The system was tested in a modular form before being deployed collectively. The PH was calibrated effectively for the 4 solution and 7 solutions, and reported accurate readings. This could be a very important system in fish farming. Some fishes in their early stages are very sensitive to water concentration. Acidic water usually results in high mortality at this stage, unlike in later stage. Getting an accurate PH becomes very important after cleaning up the pond. It can equally be used in real time to decide when pond needs to be sanitised.

Finally, we equally tested components of varied soluble contents like water and soil types.

Observation:

The readings is more stable when reading soluble constituents like water. However, we obtained an accurate arable soil PH readings of slightly alkalinity between 7.2 - 8.5, but are less unstable than that of water and solution 4 and 7.

7.6 Limitations

While the project has demonstrated its effectiveness in monitoring and regulating plant environment, we observed that the use of the PH sensor is most adapted for soluble solutions.

To be adapted for soil PH measurement therefore, it would be most recommended to measure a soluble soil sample as possible, and calibrated accordingly. These findings creates a room for further research into a more test cases and use cases of this system.

Test results, scenarios and explanations. How the system is supposed to work

8 Sources/References

References

[1] Arduino education.

[2] Blynk: a low-code IoT software platform for businesses and developers.

[3] DFRobot open-source hardware electronics and kits.

[4] Raspberry Pi Ltd. Raspberry pi.

We, Eze Nnaemeka Valentine, Onyesi John Abiagam, herewith declare that we have composed the present paper and work by myself and without the use of any other than the cited sources and aids. Sentences or parts of sentences quoted literally are marked as such; other references with regard to the statement and scope are indicated by full details of the publications concerned. The paper and work in the same or similar form have not been submitted to any examination body and have not been published. This paper was not yet, even in part, used in another examination or as a course performance. We agree that my work may be checked by a plagiarism checker.