IDIVIDUAL ASSIGNMENT (PRACTICAL)

How IOT contribute to agriculture development

Author: Tran Thanh Minh

Student ID: 103809048

COS30011 – IoT Programming

Fall 2023

Lecturer: Tuan Tran

Swinburne University of Technology

I. Summary

a. Topic background

The emergence of the Internet of Things (IoT) has ushered in a transformative era, reshaping industries across the globe. One sector profoundly impacted is agriculture, where the interconnectivity of devices and data-driven practices are revolutionizing traditional farming methods. In this context, my survey paper delves into the multifaceted influence of IoT innovation on agriculture, exploring its implications from precision farming to sustainable practices.

Traditionally, agriculture has been characterized by tradition and manual labor. However, with the surge in global population and escalating environmental concerns, the necessity for more efficient, sustainable, and data-driven approaches to farming has become increasingly apparent. The integration of IoT into agriculture provides a technological frontier that addresses age-old challenges. This paper investigates how IoT enables farmers to remotely oversee their land and crops, receiving critical updates and insights on their smartphones or personal devices. This level of connectivity affords an unprecedented level of control and responsiveness, empowering farmers, and citizens alike to take timely actions to mitigate potential risks and losses.

In the realm of farming practices, the integration of IoT technologies has ushered in a new era of agricultural management and efficiency. Sustainable farming practices have been significantly bolstered by IoT. The continuous monitoring of environmental conditions facilitated by IoT assists in pest management and disease prevention.

A proposed smart agriculture system for cornfields, utilizing wireless sensor networks and drones, is scrutinized for its benefits and limitations. The need for many sensors and the cost of drones are identified as challenges, despite the potential to improve crop yields.

Lastly, the survey explores a smart farming system based on IoT sensors for data collection and cloud computing for analysis. Machine learning is employed to provide farmers with recommendations on enhancing crop yields and reducing environmental impact. Emphasizing the real-time data collection capabilities of IoT sensors, enabling farmers to make informed decisions for improved crop yields, reduced costs, and environmental sustainability. While acknowledging the transformative power of IoT, the survey paper also highlights existing challenges such as high costs and complexity that need addressing for widespread adoption in agriculture. As IoT continues to evolve, its role in agriculture is poised to grow, contributing to a more efficient, profitable, and sustainable future for farmers and industry.

b. Proposed system

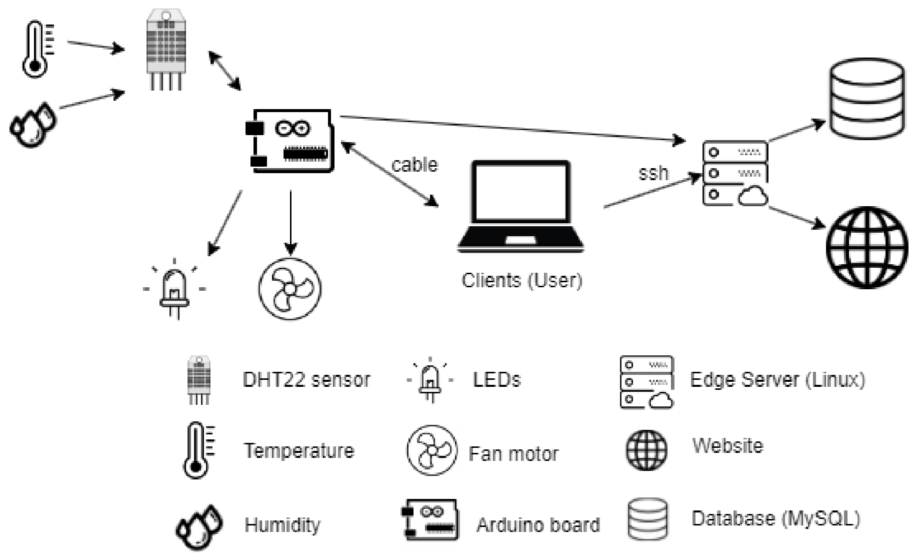

Figure 1: Sketch of proposed system

The proposed system consists of the following components:

- Sensors: The sensors collect data about the environment, such as temperature and humidity.
- Edge server: the edge server collects data from the sensors and processes it in real time. It also can be used to write serial input into the Arduino board.
- Database: The database stores the data collected by the sensors and the edge server.

The clients can interact with the system through web browsers. This proposed system offers several benefits including Real-time monitoring, distributed architecture, and scalability. The system can monitor the real-time and detect changes immediately allow the system to respond to changes quickly and alert through actuators. The components of the system are in different places and make the system more scalable and reliable and can be expanded by adding more sensors and actuators. This system can help to monitor and control devices in an agriculture setting such as irrigation systems and greenhouses. This system also involves the use of storage MySQL and can store both humidity, temperature, and current time stamp for managing.

For the sensor for collecting humidity and temperature, DHT22 is suitable for this use case, but humidity will be mainly used for controlling the system. Humidity is about 60% - 80% is ideal for plant (PLNTS.com, 2021) but in this assignment I will change the humidity for trigger conditions so that changes in humidity can be easier to managed.

II. Conceptual Design

a. Block diagrams

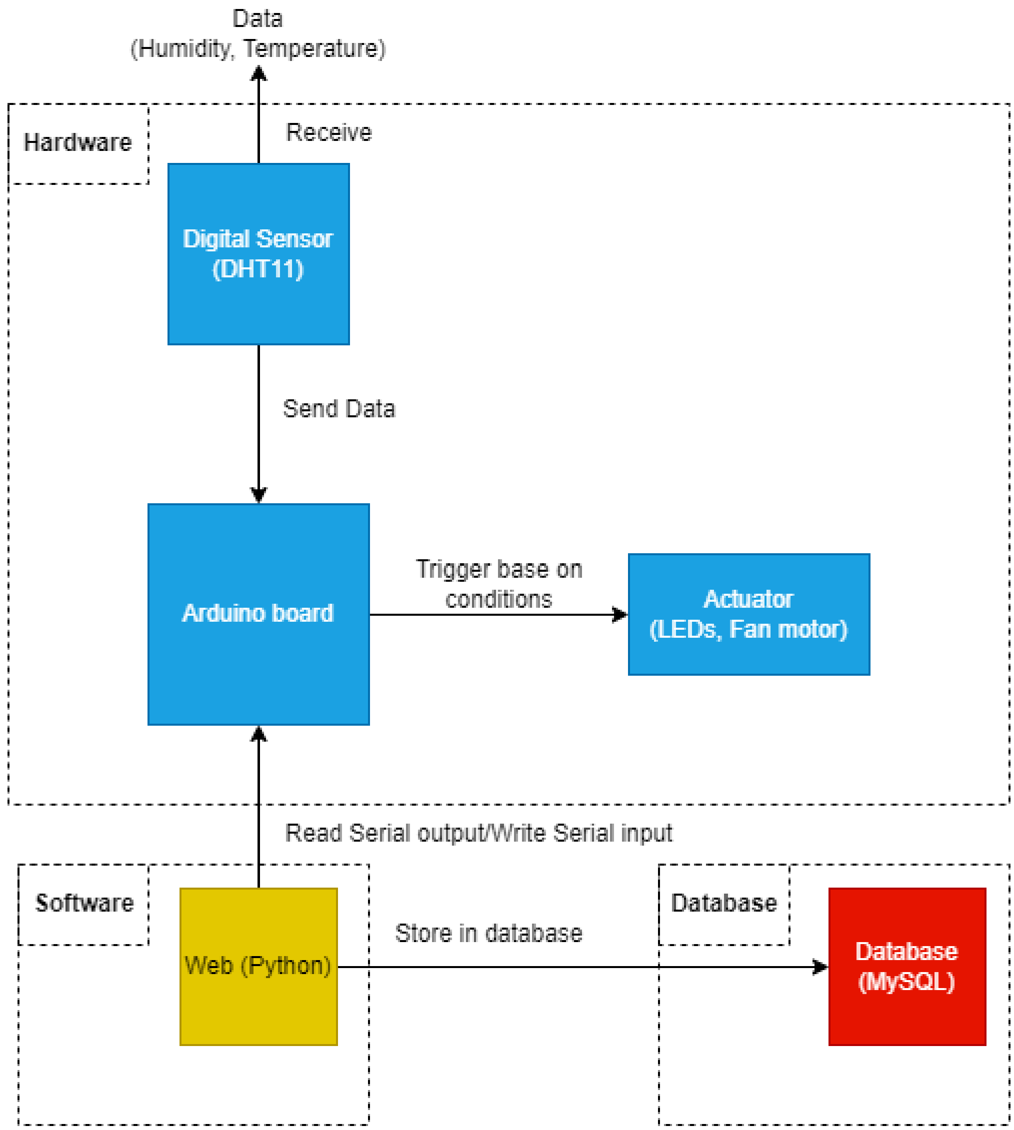

Figure 2: Block diagram for proposed system

First, the digital sensor (DHT11) will receive the data (Humidity and Temperature) from the environment then it will send the data to the Arduino board. Based on the data and conditions, the Arduino board will trigger the actuator (LEDs) for notifying. The web application can show the front end and read the serial output from the Arduino board and display the data. Depending on the user, the web application can perform the action to store the data to the database (MySQL).

b. UML diagram

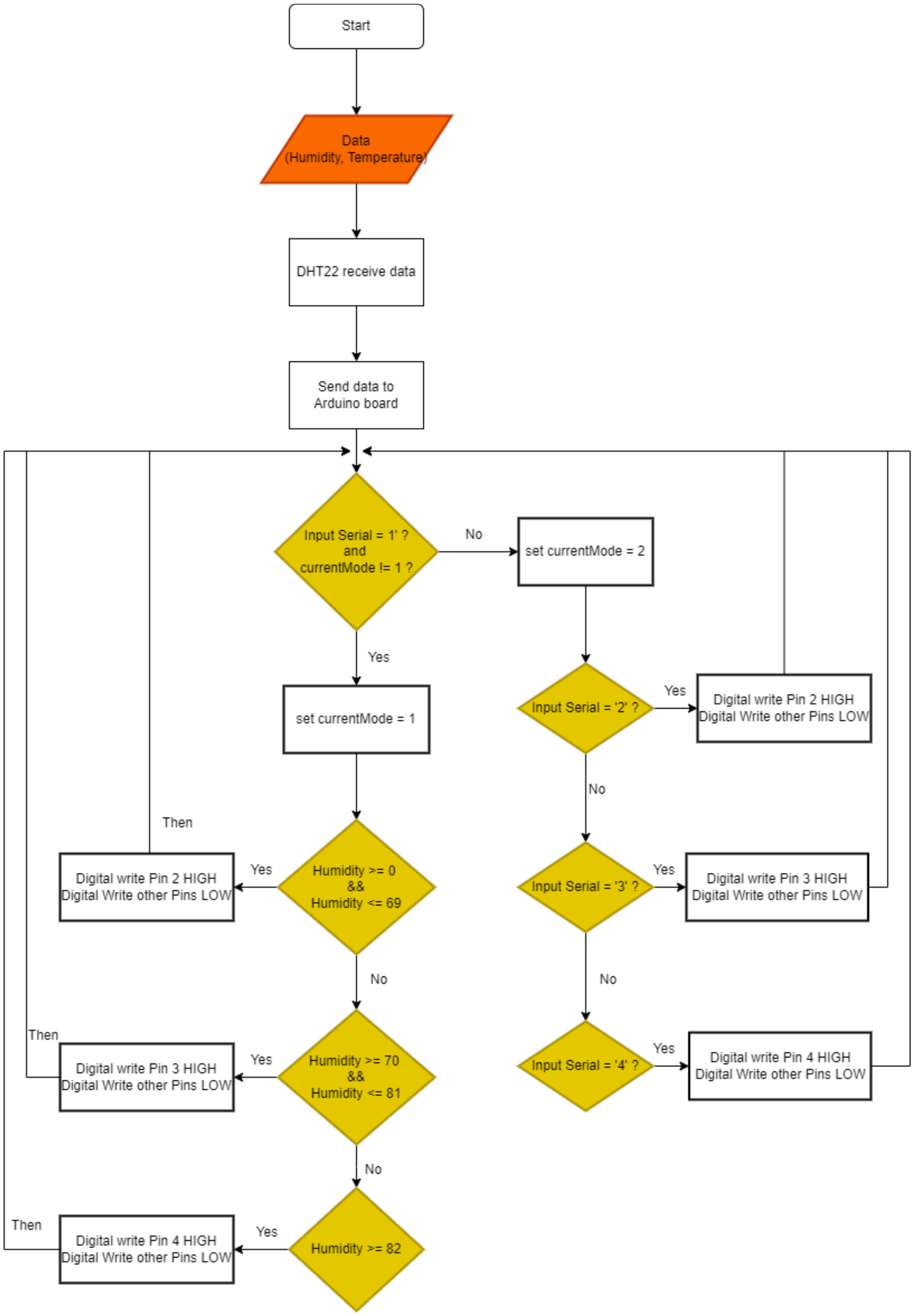

Figure 3: Flow chart for how the Arduino work

For the above chart, the Arduino code runs in the order and keeps looping. The data are humidity and temperature will be received from the DHT22. It can be divided into 2 modes: for the first mode when the user input ‘1’ to the Arduino, the code will automatically run and detect the humidity it receives from the DHT22 sensor. If the user enters ‘2’ or ‘3’ or ‘4’ to the Arduino, the current mode will switch to 2 and perform the action of setting the corresponding pin to HIGH and other pins to LOW state.

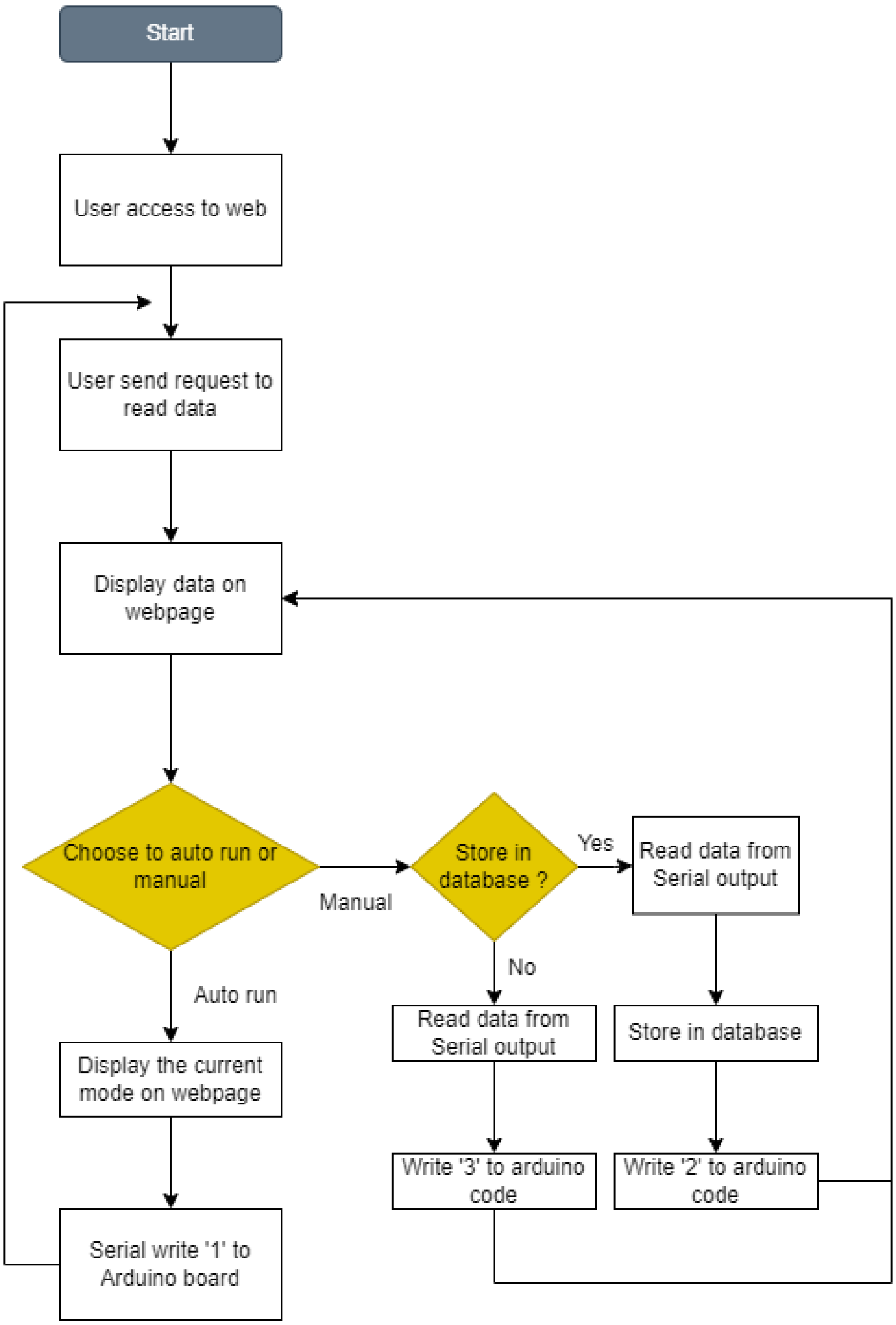

Figure 4: Flow chart for how the python program works and send request.

For the above image, the description of the system is visualized. First, when the user accesses the web, the webpage will display and then when the user requests the data, the data will be fetched by Python and display back in the web. The user can choose whenever the Arduino runs automatically or manually for managing the state. If the user chooses to run automatically, the back end of the website will write the Serial input to the Arduino board and the program will auto run (refer to the Figure 3). However, if the user chooses to run manually, the user can choose whenever to store the data in the database or not. In both cases, the first thing the backend of the web does is to read the data (humidity and temperature) from the Serial output. If the user chooses to store, the backend of the web will store the data (humidity, temperature) in the database include the time and date of the current request. the backend of the web will send Serial input ‘2’ to the Arduino code and then the Arduino board will trigger the suitable pin to alert (refer to Figure 4).

III. Implementation

The Arduino servers as a node, collecting and processing data from sensors and sending it to the edge server. The edge server processes this data further and communicates with a high-level system and displays it on web page. The edge server powered by Python and Flask communicates with Arduino through serial communication, fetching sensor data and sending control signals.

a. Sensors

DHT is a sensor for measuring the temperature and humidity of the surrounding environment. In this project, DHT is used to monitor the environmental conditions. This is crucial for applications such as climate control or data logging where knowing temperature and humidity is essential. The DHT22 is integrated with Arduino, which reads the sensor values and prints out Serial output in the customized format. It can take actions based on the data retrieved.

b. Actuators

1. LED is used in this project for status display. It can help to signal the specific conditions when the data received meets one of the conditions such as turning on it when a certain temperature is reached. It is integrated with the Arduino and controlled by the program logic to provide visual feedback.
2. Fan motor serves as drying the humidity based on the humidity readings from the DHT22. If the humidity goes over the at some point, the fan motor activates to dry the environment. Integrating with the Arduino enables automated humidity control and is crucial for real-life applications like cooling systems.

c. Software/Libraries:

Python is one of the programming languages used in this project along with its libraries like MySQLdb, serial, Flask, render_template is used for employing data processing, communication and front-end. Python interacts with the Arduino via the serial port, retrieving sensor and storing it in a MySQL database. Flask and render_template will facilitate the creation of a web interface for users to monitor and control the system by sending requests to the programmed endpoints.

MySQL will serve as the database to store sensor data. This relational database system allows efficient management and retrieval of information which will support for storing and managing all the record including time and date information. This can help for analyzing the temperature and humidity patterns.

IV. Resources

Circuit Basics (2023). *How to Use a DHT11 Humidity Sensor on the Arduino - Ultimate Guide to the Arduino #38*. *YouTube*. Available at: https://www.youtube.com/watch?v=dJJAQxyryoQ [Accessed 25 Oct. 2023].

PLNTS.com. (2021). *Temperature and humdity*. [online] Available at: https://plnts.com/en/care/doctor/temperature-and-humidity [Accessed 28 Oct. 2023].

V. Appendix