Course: IOT Smart Devices

Bean Sprout Cultivating Assistant

Instructor: Dr. Vo Bich Hien

Member
Le Mai Bao Linh - 10423072
Phan Hoang Khanh Van - 10423181

Table of Contents

I. Introduction

In this era, the Internet of Things (IoT) has become an integral part of modern life, seamlessly connecting everyday objects to the internet and enabling communication and data exchange between people, processes, and devices. By embedding sensors, actuators, and connectivity into physical objects, IoT systems collect, analyze, and share data, creating intelligent networks that can make real-time decisions and automate tasks. This technology can be applied in numerous scenarios, including smart cities for traffic and energy management, smart homes for enhanced comfort and security, healthcare for patient monitoring and diagnostics, retail for inventory tracking, agriculture for precision farming, and supply chain management for efficient logistics and tracking. By leveraging IoT, these systems can enhance reliability, reduce manual labor, optimize resource use, and enable data-driven decision-making, significantly improving productivity and quality of life.

In this project, IoT technology is utilized to assist people in growing bean sprouts at home with the Bean Sprouts Cultivating Assistant, a system designed to monitor and manage the developmental environment of bean sprouts automatically. This assistant uses sensors to track critical environmental parameters such as temperature, humidity, and water levels to ensure optimal growth conditions for the bean sprouts throughout their development stages. It can automatically water the sprouts when needed, reducing the manual effort required from the user and ensuring that the sprouts receive consistent care. By automating the monitoring and watering processes, the Bean Sprouts Cultivating Assistant not only helps users cultivate fresh, healthy bean sprouts conveniently but also encourages sustainable home food production using IoT technology.

II. Objectives

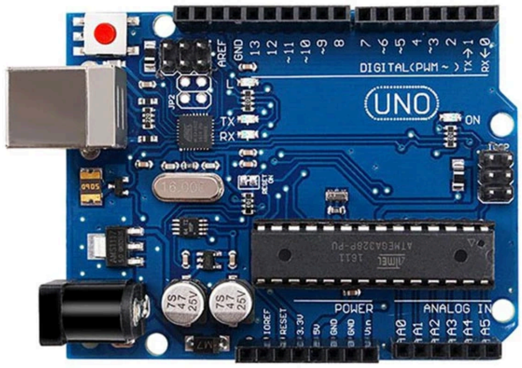
The Bean Sprouts Cultivating Assistant is designed to collect real-time data through a variety of sensors that monitor critical environmental factors necessary for the healthy growth of bean sprouts, such as soil moisture, temperature, and humidity. This data is continuously collected and transmitted to a cloud service, allowing users to remotely view and track the growth environment of their bean sprouts through an intuitive interface on their smartphones or computers. By utilizing the cloud, the system provides users with insights into historical data trends, helping them better understand the conditions that support optimal growth and enabling data-driven adjustments to the cultivation process if needed.

In addition to monitoring, the system features an automatic watering mechanism that activates when the soil moisture level drops below an acceptable threshold, ensuring that the bean sprouts receive the necessary amount of water at the right time without requiring constant manual checking from the user. By combining real-time data monitoring with automated watering, the Bean Sprouts Cultivating Assistant significantly reduces the user's workload while ensuring that the bean sprouts are cultivated under the best possible conditions, promoting a convenient and sustainable approach to home-based food production.

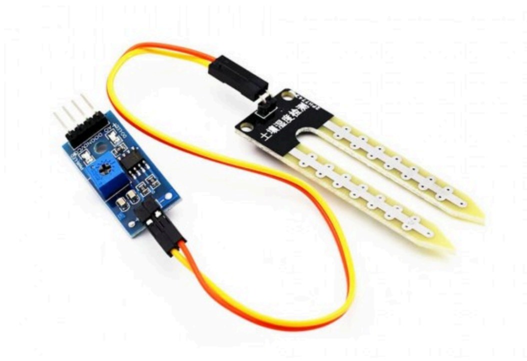
III. Background Material

1. Hardware

   Arduino UNO R3 is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button.



Soil moisture sensor is an instrument used to measure the moisture content in soil. It senses changes in moisture within the soil and converts them into readable electrical signals.



Relay Module (1 Channel) is an electronic component that acts as a switch, enabling a low-power signal to control a high-power circuit.

12V Water Pump

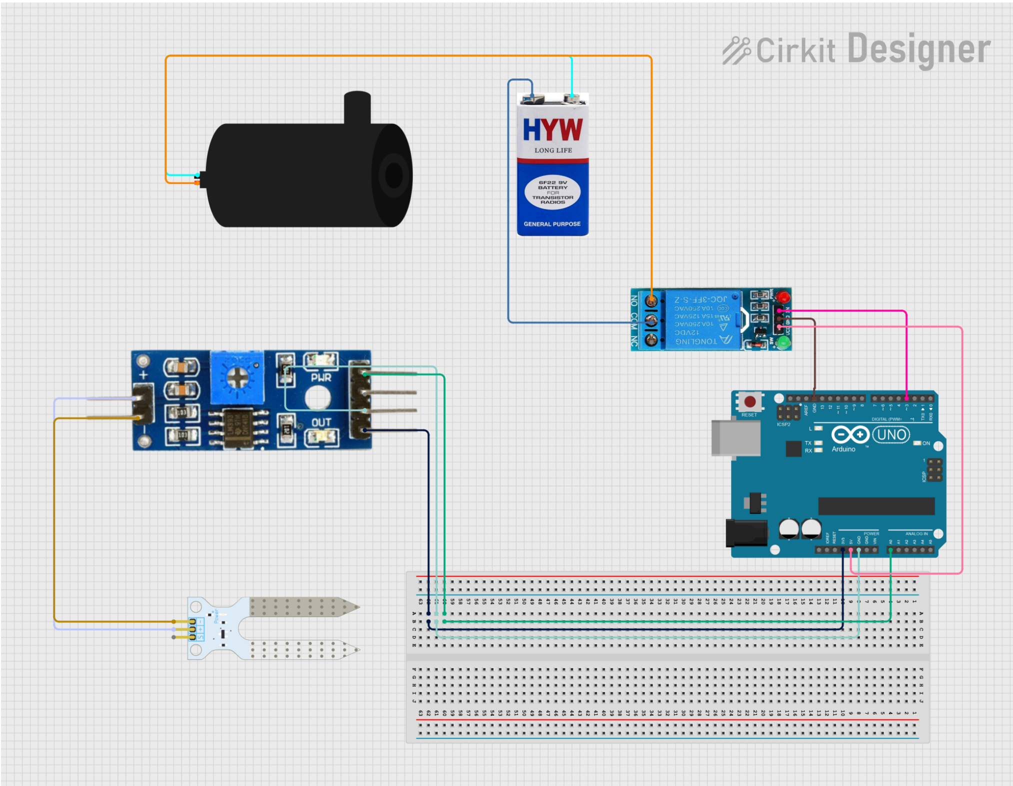
An external Power Source provides water pump power.

2. Programming Language
   Python is used to read data from the Arduino, train the model, and send the processed data to the cloud for monitoring and analysis. It handles data collection, machine learning, and cloud communication, ensuring the system can adjust conditions based on the collected data.

   C is used on the Arduino to set up and control the Bean Sprouts Cultivating Assistant. It manages sensors, actuators, and local decision-making to maintain optimal growing conditions for the bean sprouts.

IV. Implementation

1. Circuit
   An Arduino Uno board is connected to the computer via USB for programming and data communication. A capacitive soil moisture sensor is connected to the Arduino (VCC to 5V, GND to GND, and Analog Output to pin A0) to monitor soil humidity. A relay module is connected to a digital pin (D3) on the Arduino, allowing control of an external water pump. The relay controls the pump through a separate power supply using its COM and NO terminals. All components share a common ground to ensure proper electrical operation.



2. Code

The software implementation of this project involves two primary components: the Arduino program for sensing and actuation, and a Python application for data transmission, machine learning inference, and cloud communication.

   a) *Arduino Program*

   The Arduino board is programmed using the Arduino IDE. It performs the following tasks:
   - Reads soil moisture values from the analog sensor every 5 seconds.
   - Calculates the percentage of soil moisture using a mapping function.
   - Controls a relay module that activates a water pump when moisture drops below a set threshold (60%).
   - Sends the moisture readings via Serial (USB connection) to a computer for further processing.

   b) *Python Script*

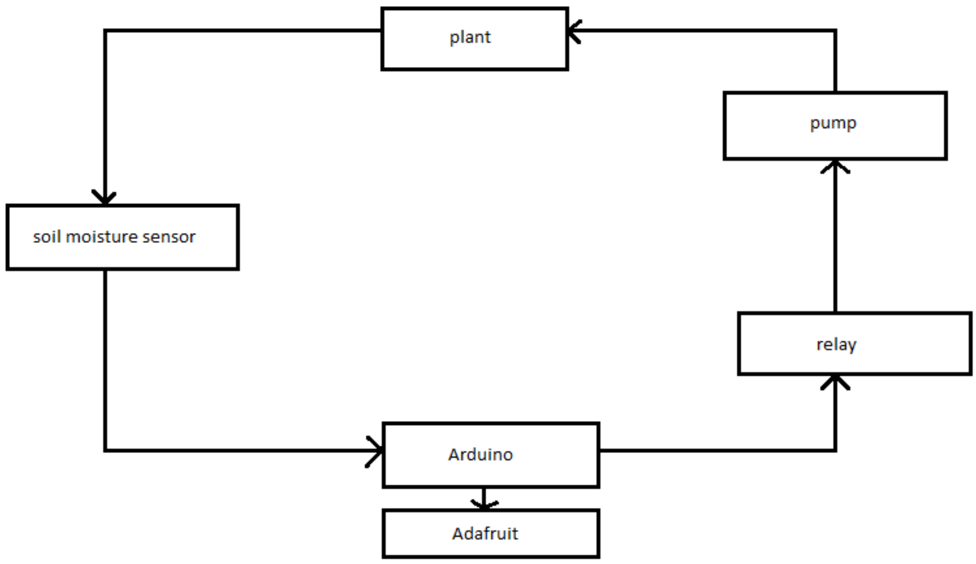
   The Python script *iot.py*, developed using Visual Studio Code, runs on a connected computer and handles:
   - Receiving serial data from the Arduino (moisture readings).
   - Logging the data and sending it to Adafruit IO feeds (soil-moisture and relay).
   - Loading a trained machine learning model *moisture_predictor.pkl,* using joblib.
   - Predicting future soil moisture values based on the latest three readings.
   - Uploading the predicted values to the predicted-moisture feed for visualization.

   The script is configured to transmit real soil moisture data to Adafruit IO every 20 minutes. If at least three valid readings are available, the system performs a prediction within 2 seconds after the last real value is sent.

   c) *Model Training*

   A separate Python script *train_model.py* is used offline to train a linear regression model using historical data from *moisture_data.csv*. The script:
   - Creates lag features from the time-series data.
   - Trains a model to predict the next moisture reading using the previous three values.
   - Saves the model for later use in the main script.

V. Prototype



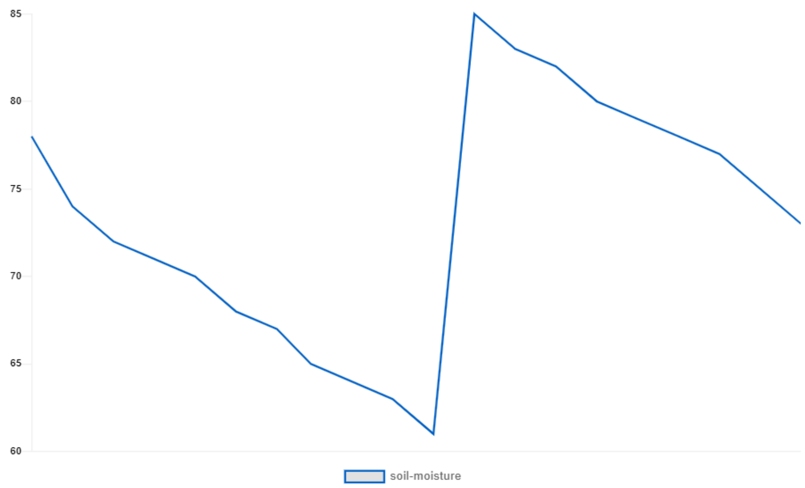
The Arduino continuously reads moisture values from the sensor and sends the data via Serial to a computer running a Python script. The Python script receives the moisture data, sends it to Adafruit IO for real-time visualization, and controls the pump by sending commands back to the Arduino through the relay.

A separate Python script *train_model.py* is used to train a machine learning model using historical moisture data. This model is saved as *moisture_predictor.pkl* and loaded during runtime to predict future soil moisture levels. The machine learning model uses the last 3 recorded moisture values to predict the next value using linear regression. This prediction helps the system anticipate upcoming dryness before it occurs, improving responsiveness. Predicted values are published to Adafruit IO alongside real-time data for monitoring trends over time.

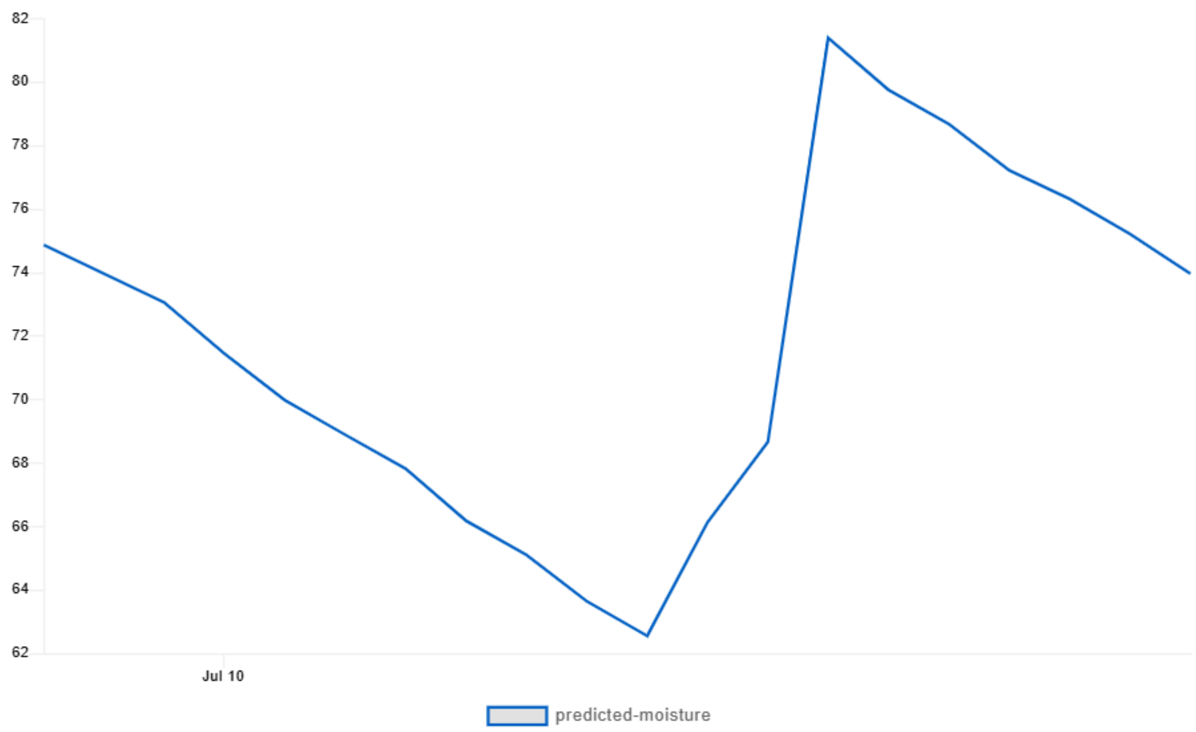
By combining live data with short-term forecasts, the system supports smarter, data-driven irrigation control.

VI. Testing Results

The system successfully reads moisture levels and automatically triggers the water pump when the value drops below a set threshold. Data is accurately transmitted and visualized on Adafruit IO dashboards.



Soil moisture



Predicted moisture

The prediction model is functioning, though further tuning is needed for higher accuracy.

Bean sprouts

VII. Conclusion

The Bean Sprout Cultivating Assistant has demonstrated the potential of IoT technology in automating agricultural monitoring and control. The prototype system successfully reads soil moisture levels, controls a water pump through a relay, sends data to Adafruit IO for visualization, and uses a machine learning model to forecast future moisture levels.

Currently, the system connects to a computer via a USB cable for both power and data transmission. This method is suitable for experimental environments but limits flexibility in real-world deployment. A future enhancement would be to add wireless communication capabilities (such as Wi-Fi or Bluetooth) using modules like ESP32. This would allow the device to function independently without being tethered to a computer, making it more practical for household or commercial use.

The capacitive soil moisture sensor used in this project is inexpensive and sufficient for prototyping; however, it tends to degrade over time, leading to reduced accuracy. For long-term use, we recommend upgrading to more durable, industrial-grade sensors to ensure consistent and reliable readings.

Additionally, the machine learning model currently applies a simple linear regression algorithm based on the last three moisture values. While this provides a reasonable prediction, the model can be improved using more sophisticated techniques such as decision trees, random forests, or LSTM neural networks. These methods can better handle fluctuations in environmental conditions and lead to more accurate forecasting.

With these enhancements, the system can evolve from a working prototype to a fully autonomous and intelligent assistant for small-scale urban farming or indoor gardening.

Reference

Caring for your vegetable garden in Maryland. (n.d.). University of Maryland Extension. https://extension.umd.edu/resource/caring-your-vegetable-garden-maryland/

Brooks, D. R. (2016). *Arduino-based dataloggers: Hardware and software* (Version 1.3). Institute for Earth Science Research and Education. © 2014–2016.

Wilber, J. (n.d.). *Linear regression*. MLU-Explain. https://mlu-explain.github.io/linear-regression/#:~:text=Linear%20regression%20is%20a%20supervised,unmodeled%20parts%20of%20our%20data.