Boston University
Electrical & Computer Engineering

EC 463 Senior Design Project

First Prototype Report

by

Team 26

Dilhara DeSilva dilharad@bu.edu
Arthur Hua ahua102@bu.edu
Alex Muntean munteana@bu.edu
Jared Solis jared11@bu.edu
Sourav Shib shib0826@bu.edu

Required Materials:

Hardware:

- ESP32 Microcontroller
- 12V DC Variable Power Supply
- Camera Module (Logi Webcam HD 1080p)
- 12V Peristaltic Water Pump
- Adafruit Soil Moisture Sensors (2)
- Temperature Sensor
- Humidity Sensor
- MOX Sensor (air quality)
- SRD (2)
- 9V Battery
- 11W Grow Light
- Inverter

Software:

- Python 3 Scripts
    - Capturing photos
    - Object recognition
- Arduino IDE
- React Native Expo
- Arduino IDE for ESP32
- Python scripts for image recognition (OpenCV, TensorFlow)
- Web server (Node.js) for data visualization
- Sensor handling libraries (Adafruit libraries for sensors)

Set Up:

The system consists of the following components:

1. **ESP32**: Manages sensor readings (soil moisture, temperature, humidity, MOX) and controls the water pump and grow lights using step recovery diodes.
2. **Laptop**: Handles image capture using the webcam and processes the images using a trained model for ripeness detection.
3. **Power Supply**: Uses a DC variable power supply for consistent voltage output due to unreliable battery performance in previous tests.
4. **Web App**: Displays scheduling information and all-time statistics (water, light).

Workflow Testing:

1. Light scheduled to turn on → ESP32 triggers grow light activation.
2. Soil moisture sensor detects dry soil → ESP32 triggers water pump activation.
    a. Water pump scheduled to turn on → ESP32 triggers water pump activation.
3. Camera captures plant image → The laptop prompts the model → Model provides reasonable plant assessment → Web app displays scheduling information and all-time statistics.

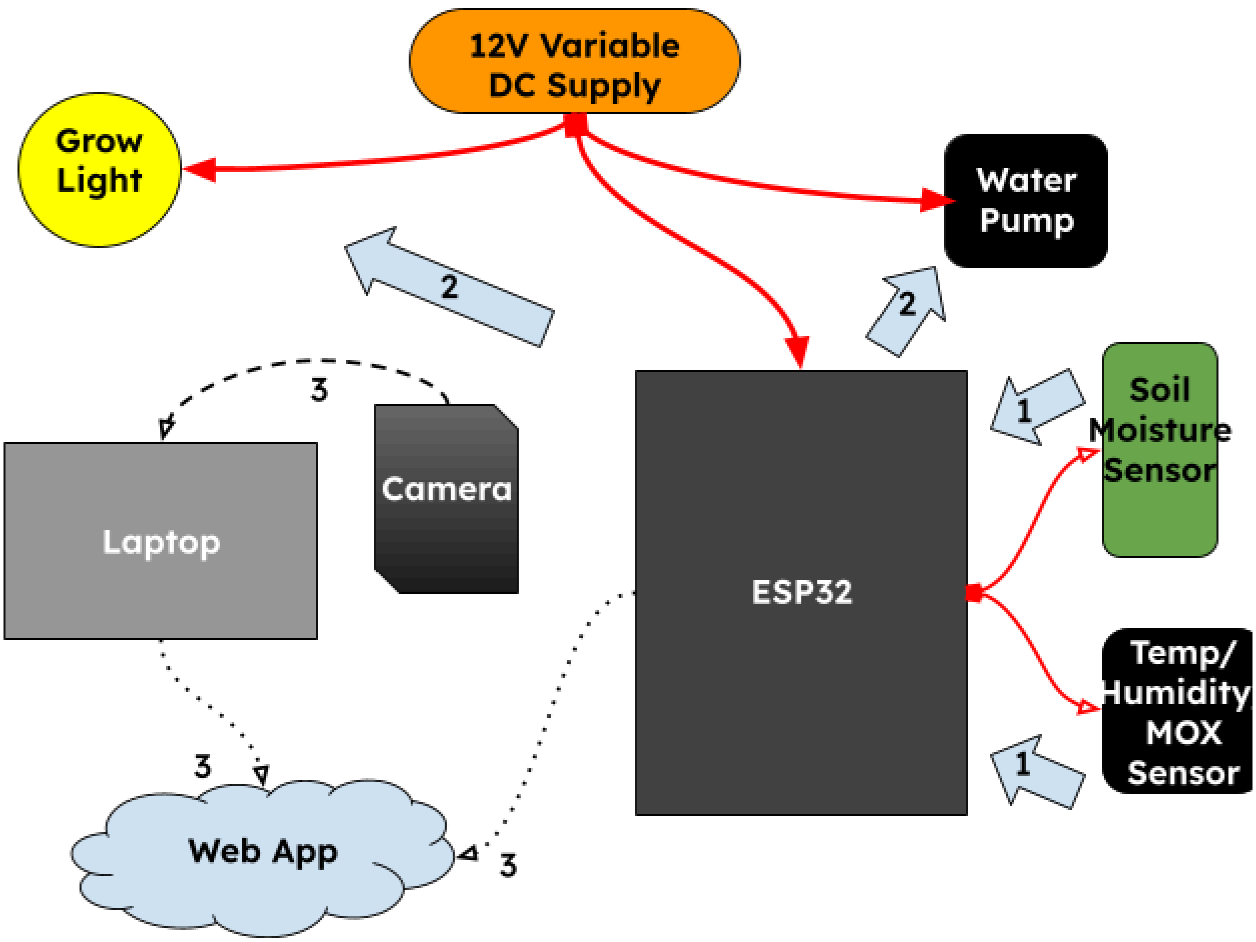

Figure 1: Illustration of Setup and Process Flow

Measurable Criteria:

I. The ESP32 should activate the grow lights when given the ON command, keeping light levels on and constant for plant growth.

II. The water pump must activate when given the command to turn on from the ESP32.

III. The camera and code script must successfully capture images and provide a reasonable plant assessment based on the image.

IV. The web app should display all-time information like total hours of light and total water deployed through scheduling without significant delay.

Score Sheet:

Test Case	**Expected Outcome**	**Pass/Fail**
Light Control	On/off function	✓
Water Pump Activation	On/off function	✓
Image Capture	Successful image capture	✓
Plant Assessment	Plant type, plant health, ripeness	✓
Sensor Data Display	Real-time, accurate data	✓

Detailed Measurements:

Comprehensive testing was conducted to measure the performance of individual components and the overall system. The following measurements were recorded:

1. **Soil Moisture Sensors**:
    - **Sensor 1**: Initial reading of **400** () before watering; post-watering, the reading increased to **1015**, indicating optimal hydration.
    - The sensor consistently provided accurate moisture levels, enabling reliable feedback for the water pump operation.
2. **Temperature and Humidity Sensors**:
    - Temperature readings averaged **25°F**, consistent with the controlled indoor environment.
    - Humidity levels ranged between **20% and 40%**, within the optimal range for the chosen plant type.
3. **Air Quality Sensor (MOX)**:
    - The MOX sensor measured CO_2 levels in ppm. During testing, CO_2 readings ranged from **350 ppm (ambient air quality)** to **450 ppm**, depending on environmental changes. These readings provide valuable data for understanding the growing environment.
4. **Grow Lights**:
    - The grow lights were tested for simple on/off functionality. The lights activated reliably via the ESP32 command and remained operational for the desired duration. Total hours of light provided were tracked and displayed on the web app.
5. **Water Pump**:
    - The water pump activated when manually triggered by the ESP32. It successfully delivered water until the soil moisture sensors indicated optimal levels. Total water dispensed was tracked and recorded on the web app.
6. **Camera and Image Processing**:
    - Images captured by the camera were processed by the llama3.2vision model. Model responses included accurate identification of plant type, ripeness status, and health status, achieving an response time at approximately 4 minutes of processing per response.

Conclusions Based on Test Data:

The prototype testing revealed significant insights into the functionality and integration of the Simple Sprouts system. Key conclusions are as follows:

1. **Grow Light and Water Pump Operation**:
   The ESP32 successfully controlled the on/off operation of both the grow lights and the water pump. While the grow lights and water pump performed reliably during manual testing, the lack of scheduling functionality highlighted an area for improvement. Scheduling automation is essential for reducing user intervention and improving the prototype's long-term reliability.
2. **Sensor Data Integration**:
   The system seamlessly collected real-time data from all sensors, including soil moisture (2 sensors), temperature, humidity, and air quality (MOX sensor measuring CO_2 in ppm). These readings were accurately displayed on the web app, along with cumulative statistics such as total hours of grow light operation and total liters of water dispensed. The ability to monitor these metrics remotely demonstrates a critical step towards an autonomous vertical farming system.
3. **Camera and Image Processing**:
   The camera model, integrated with chatbot functionality, provided accurate assessments of plant type, ripeness status, and health status. This feature adds significant value to the system by offering actionable insights to the user in an interactive format. While the reliance on a teammate's laptop for processing limits scalability, this temporary solution validated the potential of the image processing component.
4. **Overall System Reliability**:
   The system proved reliable in its current state, with all sensors and actuators functioning as intended. However, addressing the lack of scheduling for the water pump and grow lights, as well as transitioning image processing to a dedicated onboard solution, are critical steps for the next iteration. These improvements will enhance autonomy and scalability for future deployments.