Boston University
Electrical & Computer Engineering

EC 463 Senior Design Project

First Prototype Testing Plan

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:

- Raspberry Pi 1 Model B
- ESP32 Microcontroller
- 12V DC Variable Power Supply
- Camera Module (Logi Webcam HD 1080p)
- 12V Peristaltic Water Pump
- Adafruit Soil Moisture Sensors (2)
- SRD (2)
- 9V Battery
- 11W Grow Light
- Flywheel Diode
- Resistor
- 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, light, temperature, humidity) and controls the water pump and grow lights using step recovery diodes.
2. **Raspberry Pi**: 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 real-time sensor data and images captured by the camera.

Pre-testing Setup Procedure:

1. Connect all sensors (soil moisture, light, temperature/humidity) to the ESP32 and verify communication via I2C.
2. Ensure the Raspberry Pi is connected to the camera and can capture images using Python.
3. Verify optimal water levels in water storage.
4. Set up the DC power supply to provide 12V power for the water pump and grow LEDs.
5. Upload the control code to the ESP32 using Arduino IDE.
6. Run a test script on the Raspberry Pi to ensure image capture and basic object detection are functioning correctly.

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 → Raspberry Pi processes ripeness percentage through camera model → Web app displays all sensor data, the image, and ripeness status.

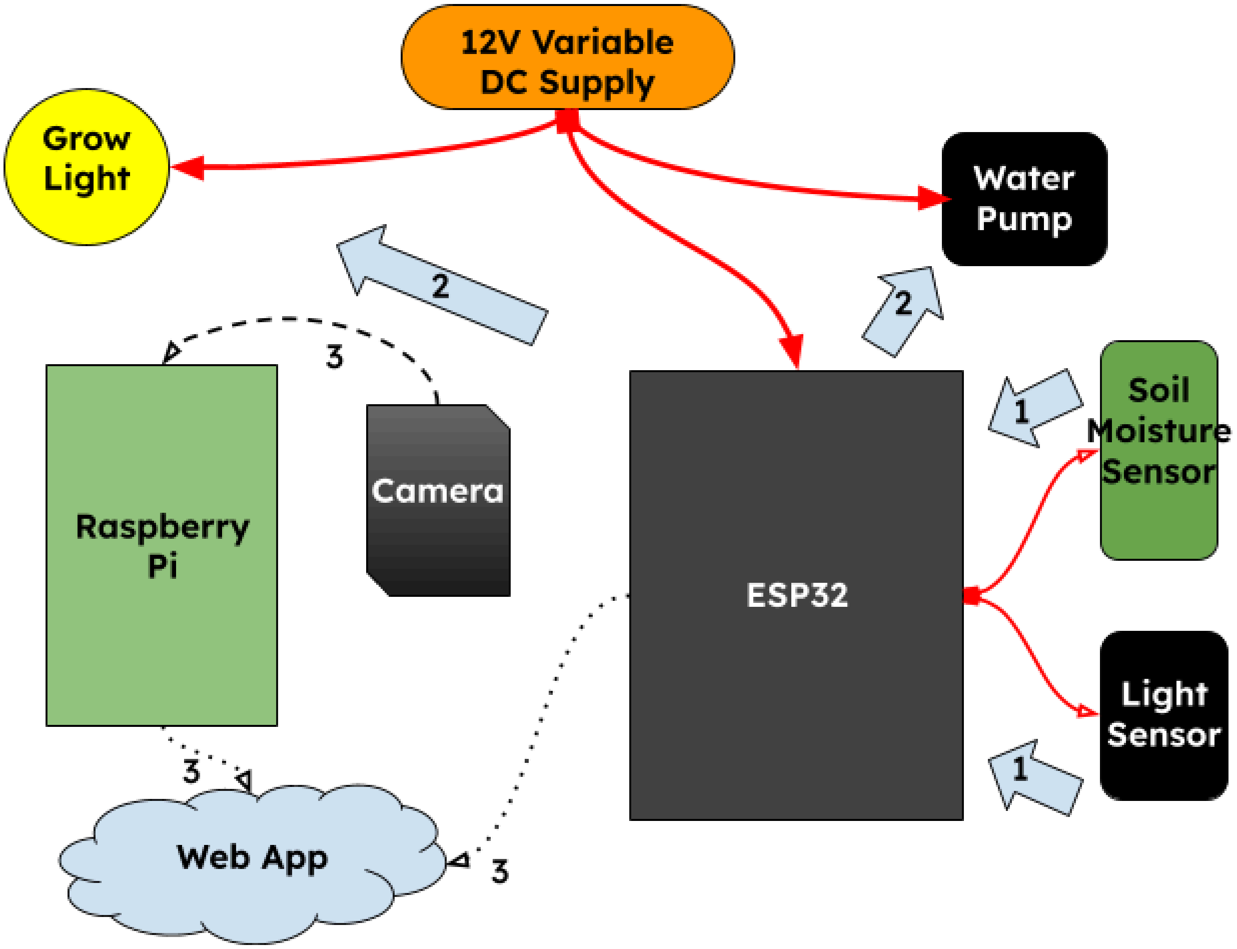

Figure 1: Illustration of Setup and Process Flow

Testing Procedure:

1. **Lighting Test:**
   - Place the planter box under the grow lights.
   - Use the light sensors to monitor ambient light levels measured in lux.
   - Verify that the ESP32 adjusts the grow lights based on the sensor readings using the MOSFET control circuit.
2. **Water Pump Test:**
   - Measure soil moisture levels using capacitive touch sensors. This system provides precise readings on a scale from approximately 200 (very dry) to 2000 (very wet), enabling accurate monitoring of soil hydration.
   - Ensure the water pump operates correctly using the MOSFET control and stops once the desired moisture level is reached.
   - The water pump operates from an ON/OFF command.
3. **Camera and Image Processing Test:**
   - Set up the Raspberry Pi to capture images of the plants.
   - Process the captured images using the Python script for ripeness detection.
   - Display the processed image data and ripeness status on the web app.
4. **Sensor Data Integration Test:**
   - Collect real-time data from all sensors (light [lux], soil moisture [200-2000], temperature [℉], humidity [%]).
   - Verify that the ESP32 sends this data to the web app for visualization.
   - Check the accuracy and stability of the displayed data.

Measurable Criteria:

I. The ESP32 should accurately control the grow lights based on light sensor data, maintaining optimal light levels for plant growth.

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

III. The Raspberry Pi must successfully capture images and provide a ripeness assessment with an accuracy of at least **80%**.

IV. The web app should display real-time sensor data (light, moisture, temperature, humidity) and image outputs 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	
Ripeness Detection	Accurate ripeness assessment (≥ 80%)	
Sensor Data Display	Real-time, accurate data on web app	

Hardware Pinout

Component	ESP32 Pin #	Raspberry Pi Pin #	Usage/Description
Soil Moisture Sensor	GPIO 34	N/A	Moisture reading
—	GPIO 21	N/A	Light level monitoring
—	GPIO 22	N/A	Light level monitoring
Water Pump	GPIO 23	N/A	Water pump control
Grow Lights	GPIO 24	N/A	Grow lights control
Camera	N/A	USB	Image capture