GRADUATING GARDENERS

Graduating Gardeners

Team E3: Hiroko Abe, Sarah Jang, Kanon Kihara

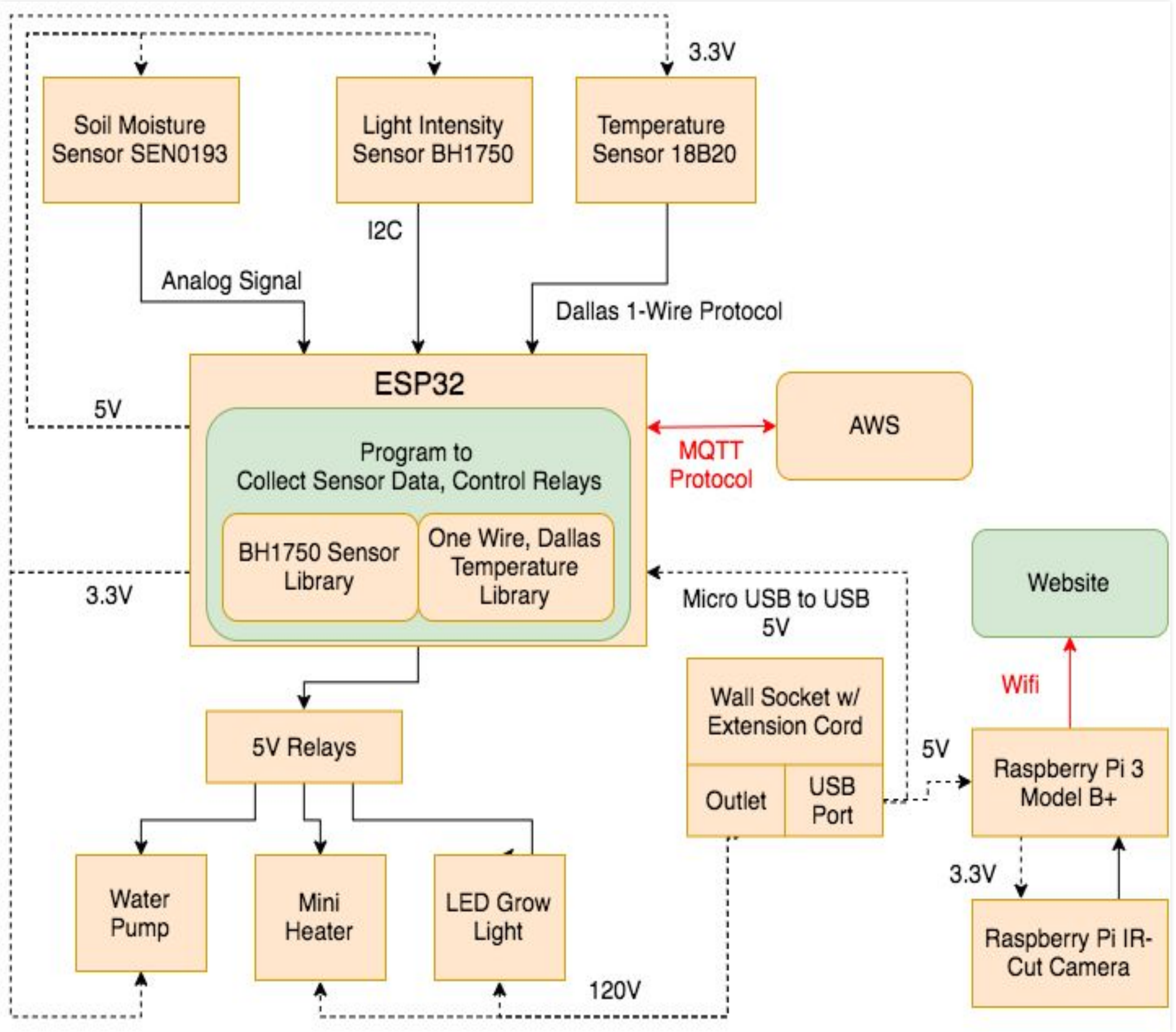
Electrical & Computer ENGINEERING

Product Pitch

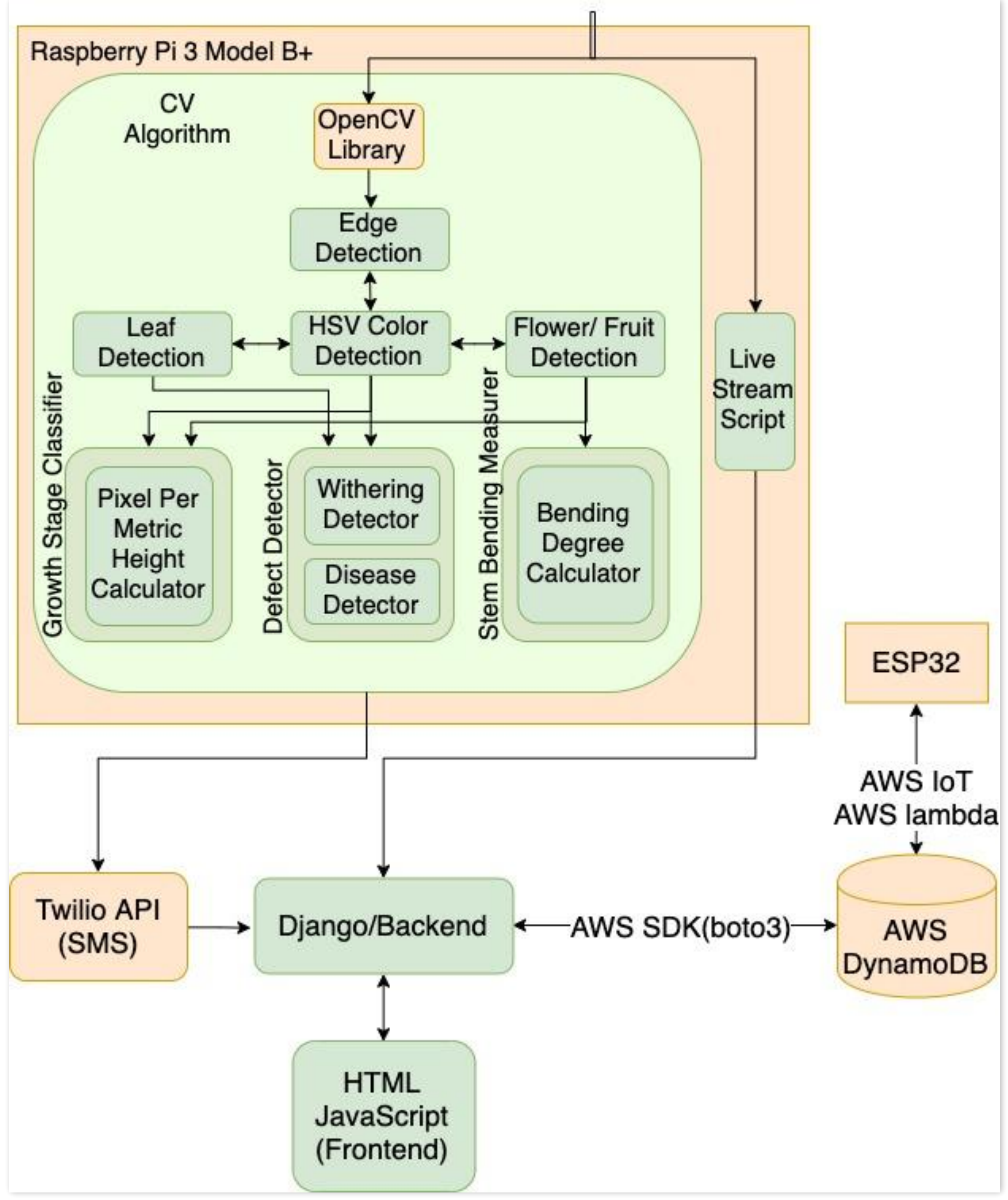
Ever since the beginning of the COVID-19 pandemic, gardening has been gaining popularity. Following this trend, we decided to base our capstone project on building an automated greenhouse. Our greenhouse can **automatically maintain** specific temperature, lighting, and watering conditions, and **detects growth status and defects of plants**.
The greenhouse is connected to an **interactive web application** where users can **monitor their plants live** and remotely control environmental variables within the greenhouse. The average time to send/receive data between web application and our greenhouse is **1.8s**. Users can also receive SMS alerts if any changes occur to the plant, such as hitting a certain growth stage or withering with an **error rate of 10%**.

System Architecture

Hardware

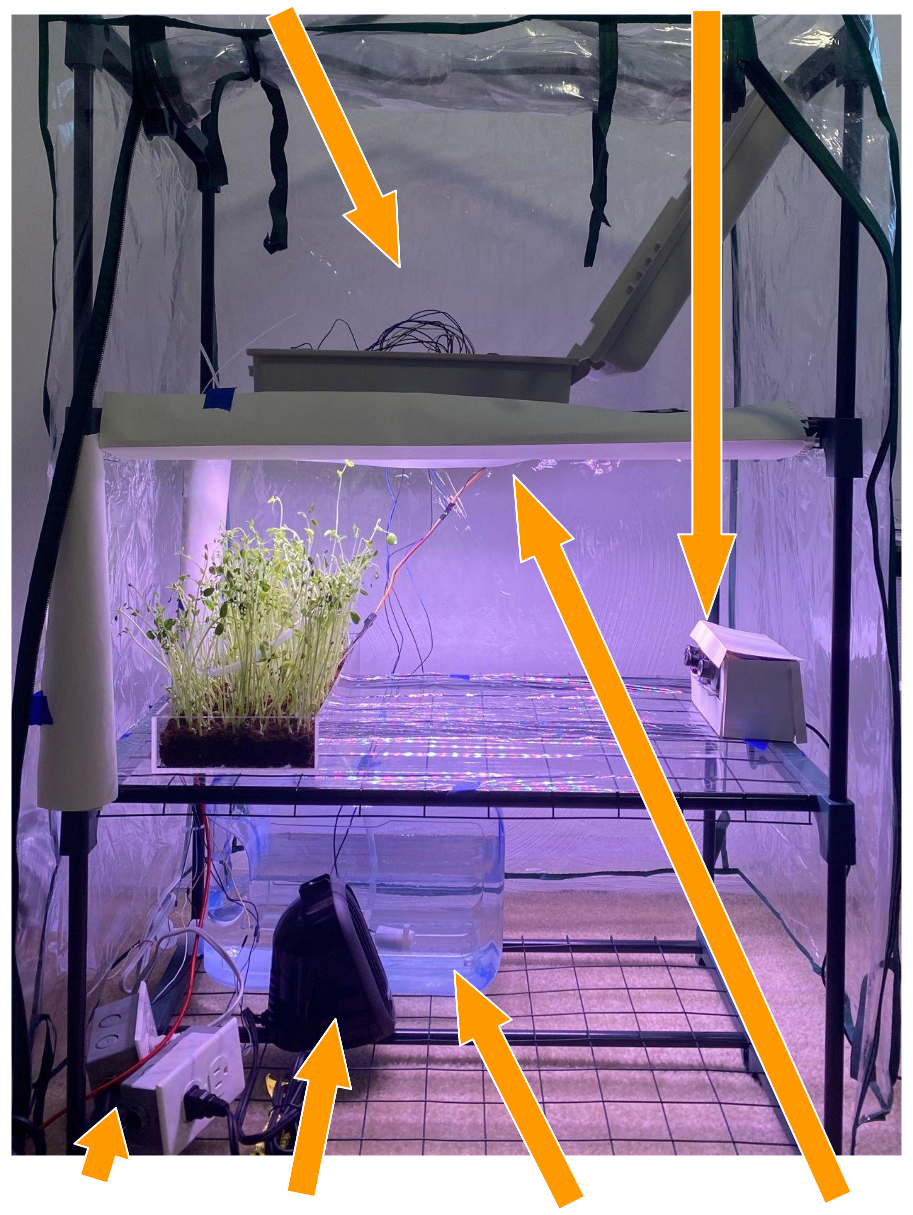


Software

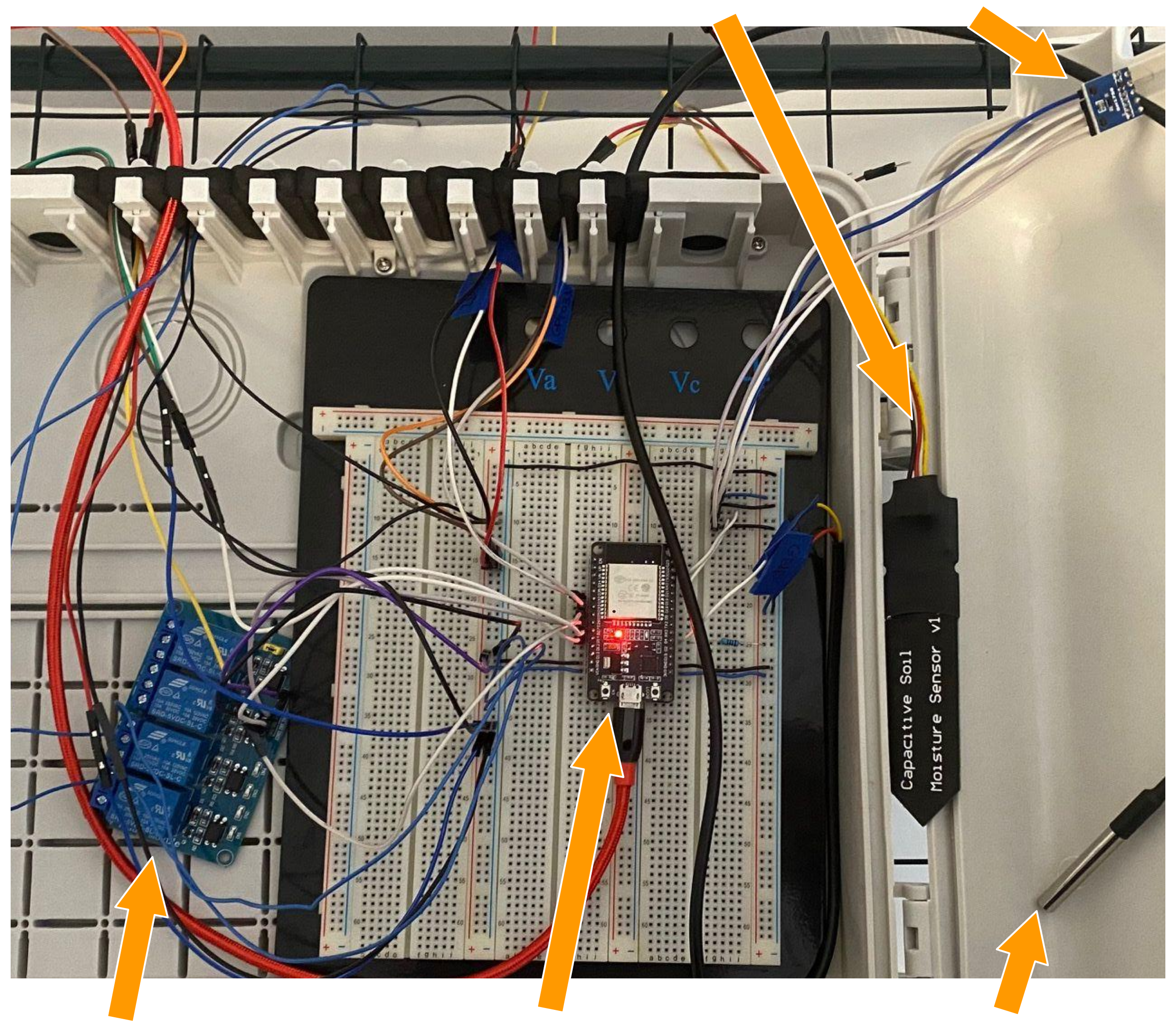


System Description

Greenhouse Components Overview



Electrical Components (microcontroller, sensors, relays)



System Evaluation

Trade offs

We decided to use both ESP32 and RPi so that the sensor component can work on ESP32 while the camera component can work on RPi.
Because of the greenhouse size and our camera's range, we had to set our plant capacity to 1, which limited plant diversity.

Hardware Testing Results

Functionality	Testing Results
Heating	Reach 84°F from 74°F in 10 minutes. Heater turns off after reaching 84°F.
Watering	Water pump turns on when target is set higher, stays off when target is lower
Data transmission (Web to Esp32)	Average 1.8s

CV Testing Results

Functionality	Testing Results
Growth Stage	Error Rate: 10%
Withering Detect.	False pos. = 8%, False neg. = 5.5%
Disease Detect.	False pos. = 8.5%, False neg. = 6%
Vine Bending	Error Rate: 9%
Video Stream	Average latency = 8.769 sec.