Smart Irrigation and Plant Health Detection using Raspberry PICO

How to Use Oled Display,Capacitive Moisture Sensor,DHT11 Temperature,Humidity Sensor and Relays with WIZnet Pico

In this project, we'll create a smart irrigation system that can detect the moisture level in the water and irrigate the plant accordingly. The system also contains a failsafe mechanism that allows the moisture sensor to recalibrate so that the plant is not overwatered. It also shows the current humidity level and temperature in degrees, as well as the plant's health condition, which is presented on the OLED screen. This project employs YOLOV5, a powerful object detection model capable of monitoring plant health in real time and transmitting the information to a MQTT server. The MQTT server sends messages to the Raspberry PICO, which it displays on the OLED panel.

Things used in this project

1 Hardware

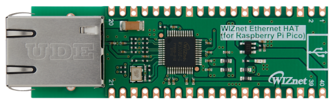
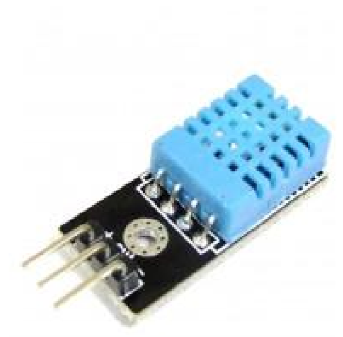
- Wiznet Pico board
- Jumper wires
- OLED Display 0.96inch
- Capacitive Moisture Sensor
- DHT11 Temperature & Humidity Sensor
- 5v Pump
- 5v single channel relay
- Light Pipe
- Bread Board
- DC Power Jack
- 5V Power Supply
- Web Cam

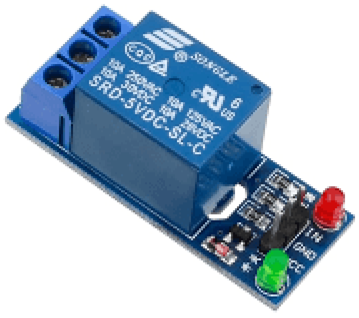
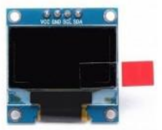
2 Software

- Arduino IDE
- YOLOV5 Object Detection

Story

India is one of the largest freshwater users in the world, and our country uses a larger amount of fresh water than other countries. There is a large amount of water used in agriculture rather than in the domestic and industrial sector. 65% of total water is contributed as groundwater. Today water has become one of the most important sources on the earth and most of it is used in the agriculture field. As the a capacitive soil-moisture sensor and temperature/humidity sensor are placed in the root zone of the plants, the system can distribute this information to the microcontroller.The raspberry pico is the heart of the system.The system can use the sensors such as soil moisture sensor and atmosphere humidity and temperature.The raspberry pi model is programmed such that if either soil moisture dips below a predefined threshold level, the irrigation system is activated, i.e. the relay connected to the raspberry pico will turn ON or OFF the motor.In order to prevent over irrigation a timeout is added.For small use cases such as a houseplant, the motor will turn on only for 5 seconds waits for recalibration of the moisture sensor if water is adequate and is above threshold it will stop irrigating else it will pump again for 5 seconds.Moreover the onboard OLED display and Serial Monitor will display the current temperature,moisture,humidity level and healthiness of the plant.Healthiness of the plant is determined by using a webcam connected to a laptop and by making use of YOLOV5 which will send the plants current status to an MQTT server. This project presents an efficient, fairly cheap and easy automated irrigation system. This system once installed has less maintenance cost and is easy to use.

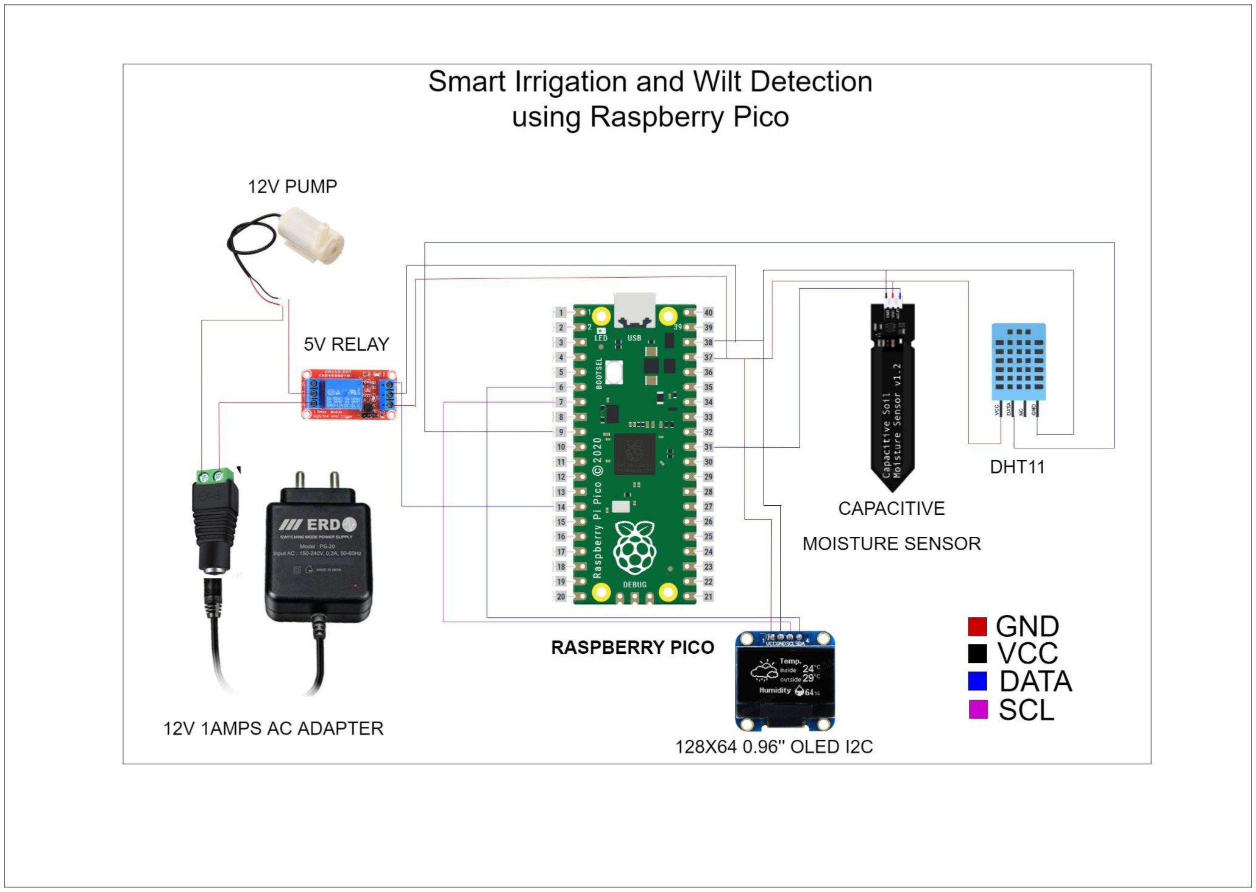
Component Used	Image	Description
Wiznet PICO ethernet hat		Controls and coordinates the pump system, collects data from the sensors, receives messages from the MQTT Server and displays it on the OLED Display.
Capacitive Moisture Sensor		Reads Moisture value from the soil.
DHT11		Reads Temperature and Humidity from the root of soil.

5V Single Channel Relay		Activates pump based on the humidity reading
0.96 inch OLED Display		Reads and displays data from the sensors. Also displays the plant health information from the MQTT Server
Breadboard and Jumper wire		Helps with interconnecting circuit elements with the Raspberry PICO

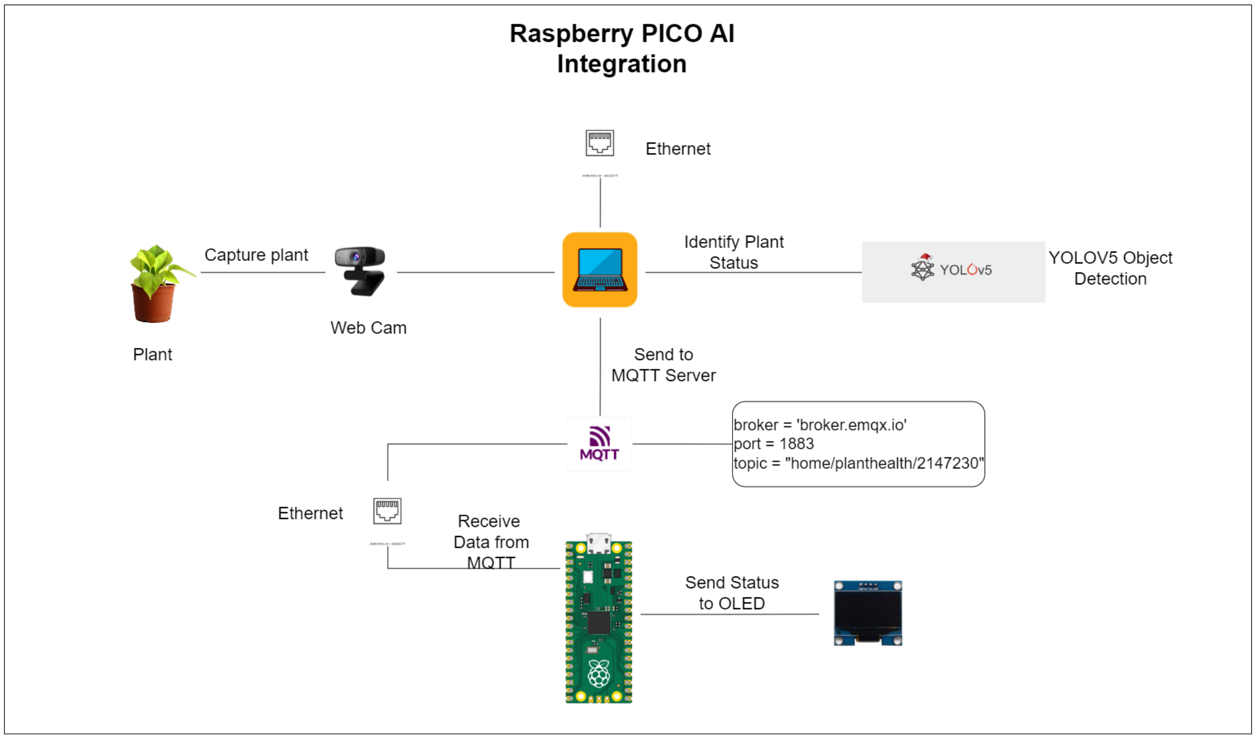
USB Webcam		Captures plant images to the YOLOV5 Object detection model for analysis
Light pipe		To collect water from reservoir to the soil
5V Pump		Supplies water from reservoir to soil

DC 5v Jack		Helps providing power supply to the 5V pump
5V DC Adapter		Converts AC Current to DC current at 5V 1 amps to power the pump

Circuit Diagram



AI Integration



Explanation

VCC , GND are connected to a common point in the bread board so that other sensors can share this point. The DHT sensor is connected to GPIO PIN 9 of the pico board which will provide temperature and humidity readings. The capacitive moisture sensor is connected to Analogue PIN A0 which will provide current soil moisture readings. The SDA and SCL pins of the OLED display are connected to the GPIO 5 and 4. Readings from DHT11 and moisture sensor are displayed on the OLED display. A relay is connected to the GPIO PIN 14 for turning on and off the pump. The +ve of the pump is connected to the relay and -ve is directly connected to the pump from the DC jack. +ve of the DC jack is connected to the common point of the relay. The system uses a 5v motor powered by a 5v DC adapter. If the moisture percent drops below 30% the relay will activate the pump mechanism for 5 seconds. It turns off the pump for 10 sec inorder for the moisture sensor to recalibrate. The microcontroller will check for the moisture level and activate the pump if the moisture level is still below 30. The relay will stop activating if the water level is above 30 inorder to prevent over watering. The OLED display will also display the health status of the plant.

How to integrate AI ?

In this project we have used the YOLOV5 object detection model on a custom trained dataset which could detect accurately when a plant is dry or not. Training is done by collecting dry pictures and healthy pictures of the same plant type. We then annotate these pictures differentiating between dry and healthy. Then feed it to the training algorithm. Ensure that the present working directory is the YOLOV5 directory and all dependencies are installed. The coco128.yaml file has to be updated with the dataset directory and also the classes used [Healthy,dry]. Train using the command:

```
!python train.py --img 416 --batch 16 --epochs 100 --data coco128.yaml --weights yolov5s.pt --cache
```

A pertained weight has been assigned to the training [yolov5s.pt] and is running at 100 epochs. Once the training is done check in the "runs" folder to check the accuracy of predictions in the .jpg files. The real time detection can be used with the command :

```
python detect.py --weights runs\plant\best.pt --img 416 --conf 0.25 --source 1
```

While running this command the present working directory should be the root of the yolov5 folder. The weight file ending in ".pt" file has to be given in the - -weight argument. The source can be set to 1 or 0 depending on the webcam used internal or external. On running this command a frame opens up with bounding box detection. In the detect.py we have made some edits so that the frames will only only be sent at a slower rate so that it matches with the speed of mqtt receiver in the microcontroller. The status of the plant will be displayed in the console. In this project we are sending this plant status to the mqtt server . These status messages include "plant is healthy","plant is dried", in case no plants are detected "no plant detected". On receiving the status from the mqtt server in raspberry pico ,we check for keywords such as "healthy" , "dried" and "no". A global variable is assigned for each status and this status will be displayed in the OLED display as well as the Serial Monitor.

Code

```
// Author Sain Saji
// Reference for OLED
// https://github.com/sainsaji/SSD1306-Base-Code-for-Raspberry-Pico-Ardunio-Library/edit/main/README.md


//Libraries Included
#include <SPI.h>
#include <Wire.h>
#include <Ethernet.h>
#include <PubSubClient.h>
#include <Adafruit_GFX.h>     //External Libraries for driving the OLED display
#include <Adafruit_SSD1306.h> //External Libraries for driving the OLED display
#include <DHT.h>

//MQTT Global Variables
byte mac[]    = {  0xDE, 0xED, 0xBA, 0xFE, 0xFE, 0xED };
IPAddress ip(172, 16, 0, 100);
IPAddress myDns(192, 168, 0, 1);
char server [] ="broker.emqx.io";
EthernetClient ethClient;
PubSubClient client(ethClient);

//Definitions
#define SCREEN_WIDTH 128        // OLED display width, in pixels
#define SCREEN_HEIGHT 64        // OLED display height, in pixels
#define OLED_RESET     -1       // Make sure this is set to -1 for Pico
#define SCREEN_ADDRESS 0x3C     // Use 3C with Pico for 128x64 OLED
#define DHTPIN 9                // Digital pin connected to the DHT sensor
#define DHTTYPE DHT11           // Defining DHT Version ,use DHT22 for newer models

//Initialization
```

```
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
DHT dht(DHTPIN, DHTTYPE);

//Global Variables
int current_status=3;       //Initial plant status [0:Healthy,1: Unhealthy, 3: No plant detected]
int rightX=26;              //Display right X value
int leftY=5;                //Display left  Y value
int soilMoistureValue = 0;  //Initial moisture value
int soilmoisturepercent=0;  //Moisture value in percentage

//Global Constants
const int AirValue = 620;     // Airvalue after calibrating
const int WaterValue = 310;   // Water Value after calibrating

//Display Frame bitmap
const unsigned char baseframe [] PROGMEM = {
  0x00, 0x1e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x3f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x73, 0x80, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0xe1, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0xe1, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0xe1, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0xe1, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0xe1, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0xe1, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0xed, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0xed, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0xed, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0xed, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0xed, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0xed, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0xed, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0xed, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x01, 0xcc, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x03, 0x9e, 0x70, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x03, 0xbf, 0x70, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x03, 0x3f, 0x30, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x03, 0x3f, 0x30, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x03, 0xbf, 0x70, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x03, 0x9e, 0x70, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x01, 0xc0, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0xf3, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x7f, 0x80, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x3f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x0c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x08, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x1e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x1c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x3e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x1e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x7f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x3f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0xff, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x7f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0xff, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0xff, 0x80, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01, 0xff, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0xff, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x03, 0xff, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x00,
0x01, 0xff, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x03, 0xdf, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x00,
0x03, 0xff, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x07, 0x0e, 0x70, 0x00, 0x00, 0x00, 0x00, 0x00,
0x03, 0xff, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x07, 0x24, 0x78, 0x00, 0x00, 0x00, 0x00, 0x00,
0x07, 0xff, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x0f, 0x0c, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00,
0x07, 0xff, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x01, 0x0f, 0x99, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x00,
0x0f, 0xff, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x01, 0x1f, 0xf2, 0x7c, 0x00, 0x00, 0x00, 0x00, 0x00,
0x1f, 0xff, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x01, 0x1f, 0xe4, 0x3e, 0x00, 0x00, 0x00, 0x00, 0x00,
0x1f, 0xff, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x01, 0x3f, 0xc9, 0x3e, 0x00, 0x00, 0x00, 0x00, 0x00,
0x1f, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x01, 0x3f, 0x9c, 0x7e, 0x00, 0x00, 0x00, 0x00, 0x00,
0x3f, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x01, 0x3f, 0xff, 0xff, 0x00, 0x00, 0x00, 0x00, 0x00,
0x3f, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x01, 0x3f, 0xff, 0xff, 0x00, 0x00, 0x00, 0x00, 0x00,
0x3f, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x01, 0x38, 0x61, 0x87, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x3f, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x01, 0x38, 0x00, 0x07, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x3f, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x01, 0x38, 0x00, 0x07, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x3f, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x01, 0x38, 0x00, 0x0e, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x1f, 0xff, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x01, 0x3c, 0x00, 0x0e, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x1f, 0xff, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x01, 0x1c, 0x00, 0x1e, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x0f, 0xff, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x01, 0x1e, 0x00, 0x3c, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x07, 0xff, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x01, 0x0f, 0x00, 0x78, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x03, 0xff, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x07, 0xe1, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x01, 0xff, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x03, 0xff, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x7f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0xff, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x3e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
};

//MQTT callback funtion
void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  char messageBuffer[30];
  memcpy(messageBuffer, payload, length);
  messageBuffer[length] = '\0';
  Serial.println(messageBuffer);
  for (int i=0;i<length;i++) {
    Serial.print((char)payload[i]);
  }
  //For convering payload to string
  char message[30];
  for(int i=0;i<length;i++){
    message[i]=payload[i];
    }
  String msg = String(message);

  //Checking if index contain healthy
  int he = msg.indexOf("healthy");
  //Checking if index contain "no" as in "no plants detected"
  int no = msg.indexOf("no");

  //if index contain "healthy" set current_status as 0 else 1
  if(he>0)
  current_status=0;
  else
  current_status=1;
  //if index contain "no" set current_status as 3
  if(no>=0)
  current_status=3;
  Serial.println();
}

//MQTT Reconnect callback
void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Attempt to connect
    if (client.connect("arduinoClient")) {
      Serial.println("connected");
      client.subscribe("home/planthealth/2147230");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
```

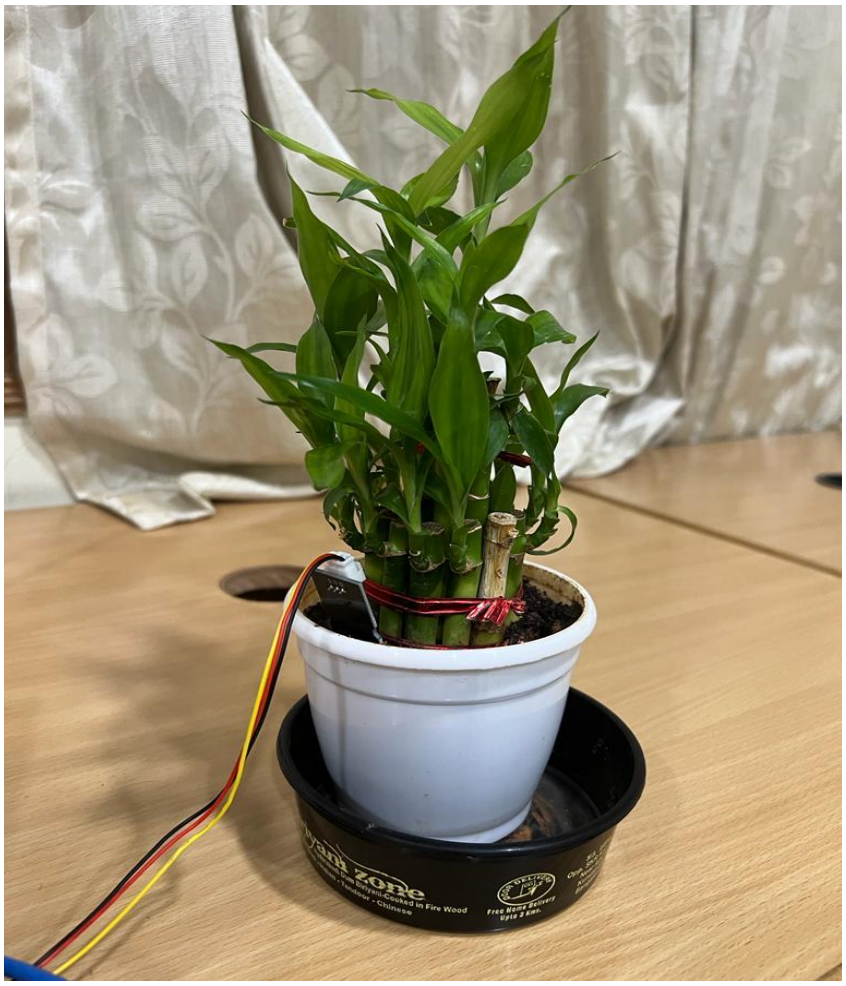
```
}

//MQTT Ethernet initialization
void ethstart()
{
  //ethernet codes
  Ethernet.init(17);  // WIZnet W5100S-EVB-Pico
  Serial.begin(115200);
  while (!Serial) {
    ; // wait for the serial port to connect. Needed for native USB port only
  }

    Serial.println("Initialize Ethernet with DHCP:");
  if (Ethernet.begin(mac) == 0) {
    Serial.println("Failed to configure Ethernet using DHCP");
    // Check for Ethernet hardware present
    if (Ethernet.hardwareStatus() == EthernetNoHardware) {
      Serial.println("Ethernet shield was not found.  Sorry, can't run without hardware. :(");
      while (true) {
        delay(1); // do nothing, no point running without Ethernet hardware
      }
    }
    if (Ethernet.linkStatus() == LinkOFF) {
      Serial.println("Ethernet cable is not connected.");
    }
    // try to congifure using IP address instead of DHCP:
    Ethernet.begin(mac, ip, myDns);
  } else {
    Serial.print("  DHCP assigned IP ");
    Serial.println(Ethernet.localIP());
  }
  // give the Ethernet shield a second to initialize:
  delay(1000);
  Serial.print("connecting to ");
  Serial.print(server);
  Serial.println("...");
  client.setClient(ethClient);
  client.setServer(server, 1883);
  client.setCallback(callback);
  // Allow the hardware to sort itself out
  delay(1500);

}

//To print temperature to OLED
void printTemp(int temp)
{
  display.setCursor(rightX,leftY);
  display.setTextSize(2);
  display.setTextColor(WHITE);
  display.println(temp);
  display.setCursor(rightX+24,leftY);
  display.println("C");
}

//To print Humidity to OLED
void printHumidity(int humid)
{
  display.setCursor(rightX,leftY+33);
  display.setTextSize(2);
  display.setTextColor(WHITE);
  display.println(humid);
  display.setCursor(rightX+24,leftY+33);
  display.println("%");
}

//To print moisture to OLED
void printMoist(int moist)
{
  display.setCursor(rightX+65,leftY+33);
  display.setTextSize(2);
  display.setTextColor(WHITE);
  display.println(moist);
  display.setCursor(rightX+24+65,leftY+33);
  display.println("%");
}

```

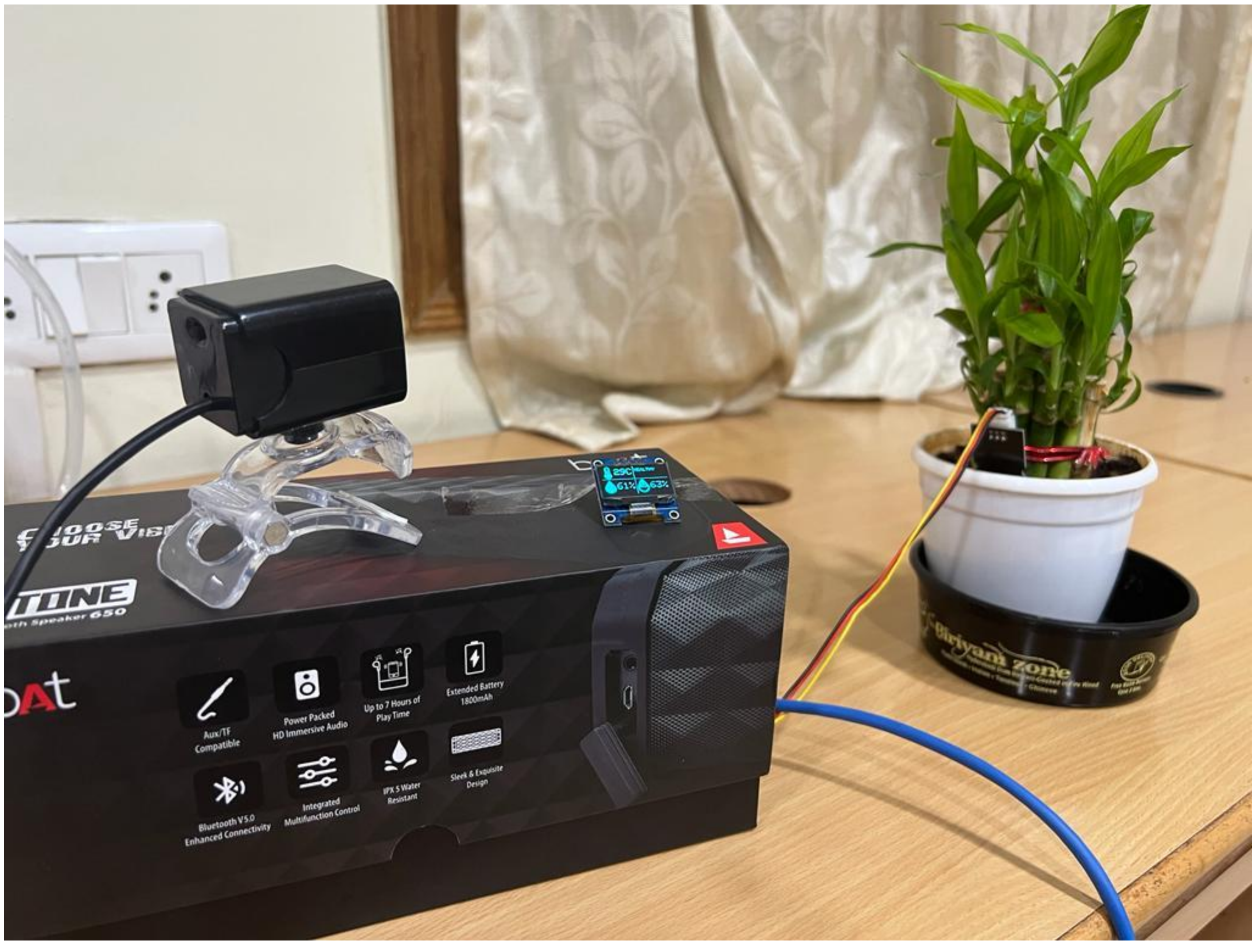
```
////To print plant status to OLED
void printStatus(String status)
{
  display.setCursor(rightX+40,leftY);
  display.setTextSize(1.5);
  display.setTextColor(WHITE);
  display.println(status);
}


//setup
void setup()
{
  Serial.begin(115200);
  dht.begin();
  pinMode(14,OUTPUT);
  pinMode(LED_BUILTIN, INPUT);
  display.begin(SSD1306_SWITCHCAPVCC,0x3C);
  display.fillScreen(0); //0 for filling with black dots. 1 for white
  printStatus("Recalibrating");
  display.display();
  ethstart();
  pinMode(LED_BUILTIN, OUTPUT);
}


void loop()
  {
    //Reconnect to MQTT server if disconnected
    if (!client.connected()) {
    reconnect();
    }
    //Subscribe to topic
   client.subscribe("home/planthealth/2147230");

  //Read temperature and humidity
  int temp=dht.readTemperature();
  int humidity = dht.readHumidity();

  //Read mositure Value
  soilMoistureValue = analogRead(A0);  //put Sensor insert into soil
  soilmoisturepercent = map(soilMoistureValue, AirValue, WaterValue, 0, 100);

   // Check if any reads failed and exit early (to try again).
  if (isnan(humidity) || isnan(temp)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }
  display.clearDisplay();
  printTemp(temp);
  Serial.println("Temperature");
  Serial.println(temp);
  printHumidity(humidity);
  Serial.println("Humidity");
  Serial.println(humidity);
  printMoist(soilmoisturepercent);

  //Startup pump if moisture percentage drops below 30%
  if(soilmoisturepercent<=30)
  {
    digitalWrite(14,HIGH);
    Serial.println("Pump Status: ");
    Serial.print("ON");
    delay(2000);
    digitalWrite(14,LOW);
    Serial.println("Pump Status: ");
    Serial.println("OFF");
    Serial.println("Recalibrating Moisture....... ");
    delay(10000);
  }
  else
  {
    digitalWrite(14,LOW);
  }
  Serial.println("Moisture");
  Serial.println(soilmoisturepercent);

  //Draw base frame on OLED
  display.drawBitmap(0, 0, baseframe, 128, 64, WHITE);

  //Display plant status based on current_status variable
  if(current_status==0)
  {
    printStatus("HEALTHY");
   }
   else if(current_status==1)
   {
    printStatus("DRIED");
    }
  else
  printStatus("NO PLANT");
  display.display();
  delay(100);

  //To recieve MQTT messages
  client.loop();
}

```

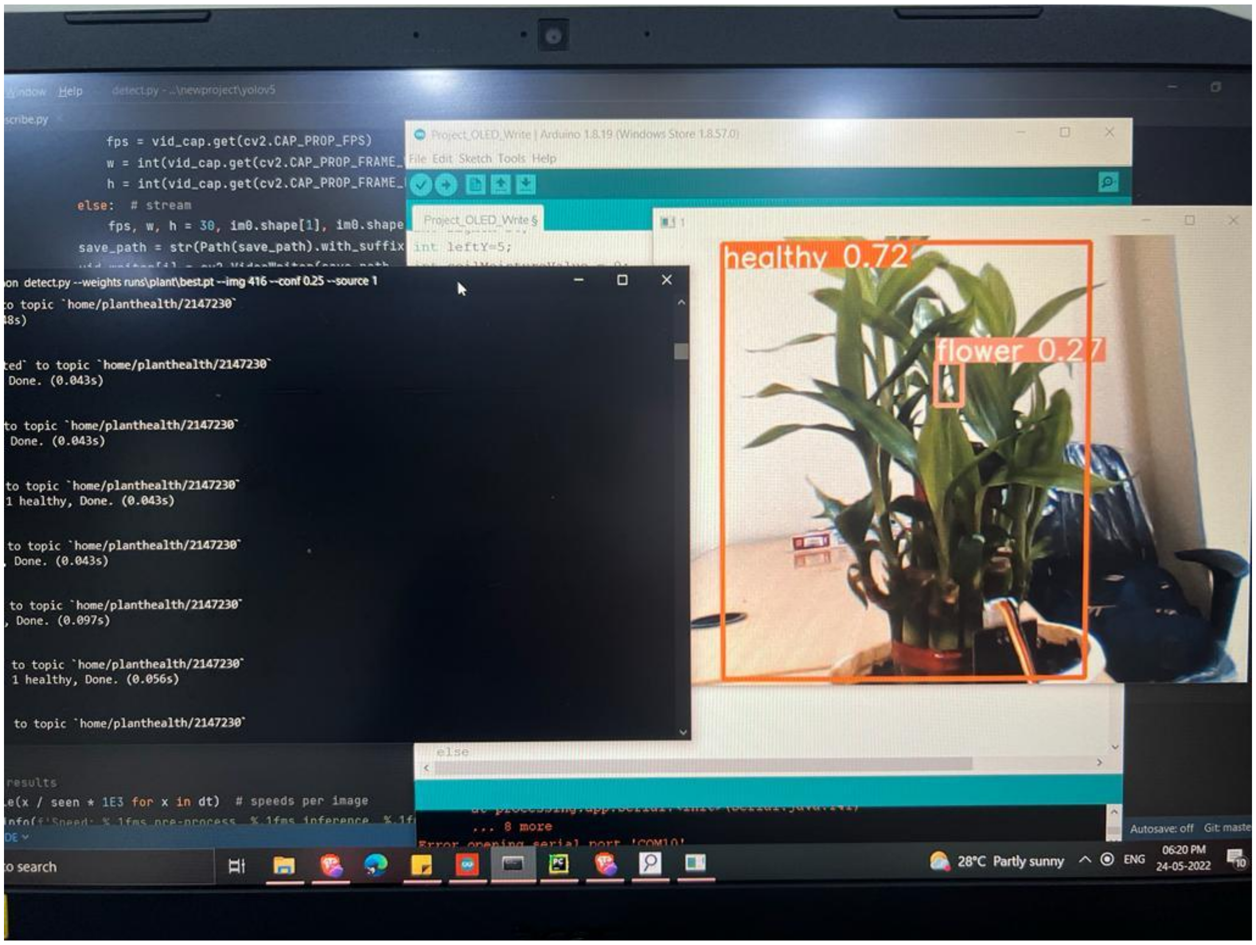
Code Explanation

The code starts with including all the libraries required for the components and the mqtt server (line no 7-13).All global variables and definitions for the MQTT server are initialized(Line no 16-21).Width and Height information for the OLED display and the DHT11 pin number is initialized(Line no 24-29). The baseframe for the OLED display bitmap is initialized(Line no 47-112). On powering the system, the microcontroller initializes all the global variables and readings from the sensors (line no 36-40). The system has to be connected to ethernet so that it will be able to receive messages from the MQTT server. Global constants for the moisture sensor is provided (Line no 43,44) .The void loop function initializes pins and the OLED display and attempt to connect to the MQTT server , when this script is run a recalibration message is displayed on the OLED screen (Line no 255-256). The ethernet initialization is done by the function ethstart(); (Line no 264) and is defined in Line 169-206. It attempts to connect to the ethernet and displays a warning message to the serial port in case of any issues.
Once connected to the ethernet it also provides a delay for the system to sort itself out (Line no 206). Once the void setup function is executed looping begins in the void loop() function (Line no 269-339). If the microcontroller is disconnected from the ethernet it attempts to reconnect (Line no 272-274). Subscription to the topic is defined in line 276. The temperature,humidity and moisture data is read (Line no 279-284). If the reading fails it displays a warning message to the serial monitor (Line no 287-289). The temperature,humidity and moisture data is displayed to the OLED display using function printTemp, printHumidity and printMoist. These functions are defined in Line no 211,222,233 which sets upto the cursor in the specific row and column and prints the data to the display. Based on the moisture level the pump will be activated ,if the moisture drops below 30% the pump activates for 5 seconds by turning on the relay and stops this allowing the moisture sensor to calibrate.If the moisture percentage is above 30 percent the relay stays off(Line no 324-335).
The global variable current_status is set by the callback function to MQTT server (Line no 115). The message from the MQTT server is in the form of a character array. We use the indexOf function to check whether a string exists in the array(Line no 134,136). If found “healthy” the current variable is set to 0, else 1. If “no” is found then it is set to 3. This information is displayed to OLED using the printStatus() function (Line no 244).

Images






Window Help detect.py - ...\newproject\yolov5
scribe.py
Project_OLED_Write | Arduino 1.8.19 (Windows Store 1.8.57.0)
File Edit Sketch Tools Help
Project_OLED_Write §
healthy 0.72
flower 0.27
28°C Partly sunny
ENG
06:20 PM
24-05-2022
Autosave: off
Git: maste

```
            fps = vid_cap.get(cv2.CAP_PROP_FPS)
            w = int(vid_cap.get(cv2.CAP_PROP_FRAME_
            h = int(vid_cap.get(cv2.CAP_PROP_FRAME_
        else:  # stream
            fps, w, h = 30, im0.shape[1], im0.shape
        save_path = str(Path(save_path).with_suffix
```

```
int leftY=5;
```

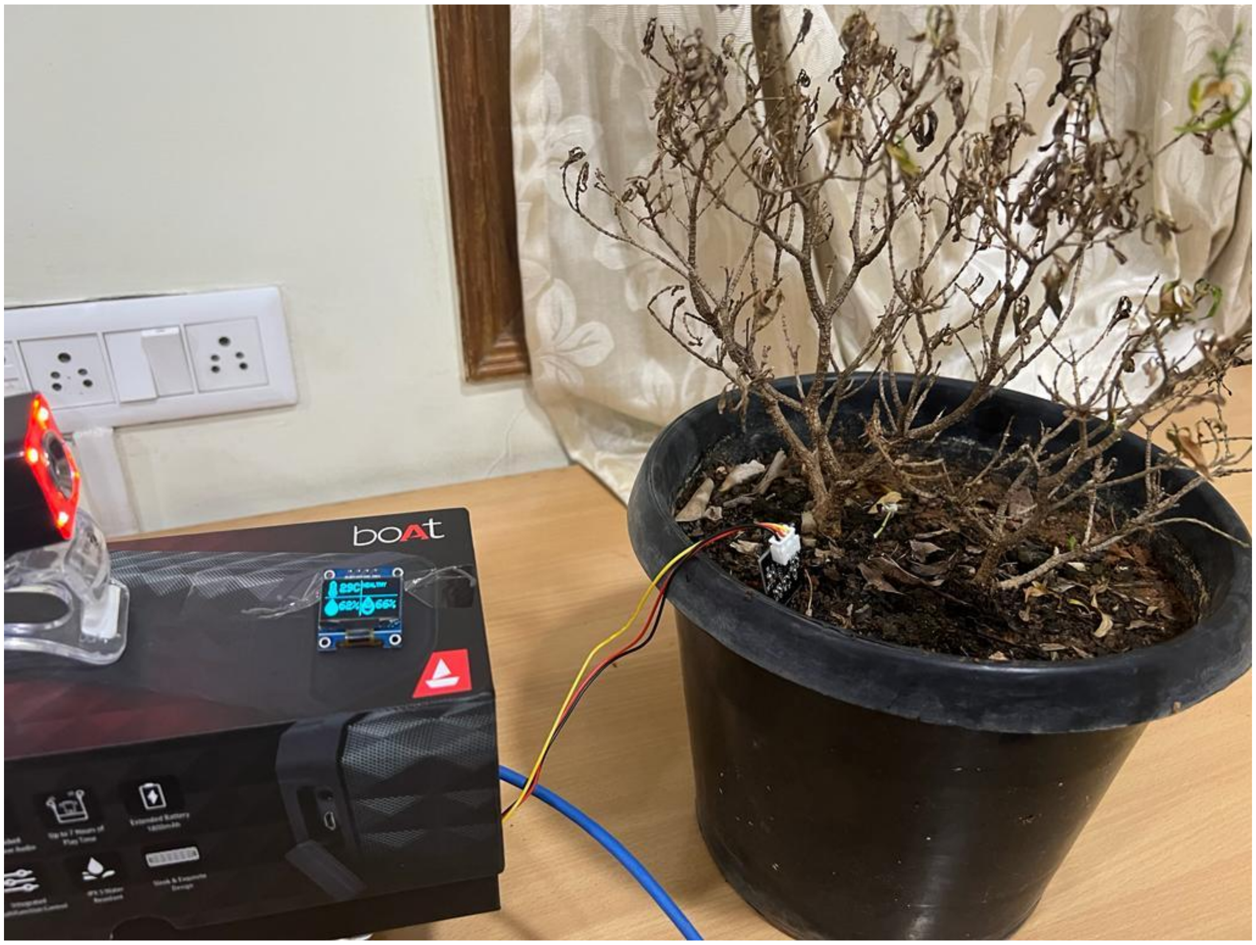
```
on detect.py --weights runs\plant\best.pt --img 416 --conf 0.25 --source 1
to topic `home/planthealth/2147230`
48s)

ted` to topic `home/planthealth/2147230`
 Done. (0.043s)

to topic `home/planthealth/2147230`
 Done. (0.043s)

to topic `home/planthealth/2147230`
1 healthy, Done. (0.043s)

to topic `home/planthealth/2147230`
, Done. (0.043s)

to topic `home/planthealth/2147230`
, Done. (0.097s)

to topic `home/planthealth/2147230`
1 healthy, Done. (0.056s)

to topic `home/planthealth/2147230`
```

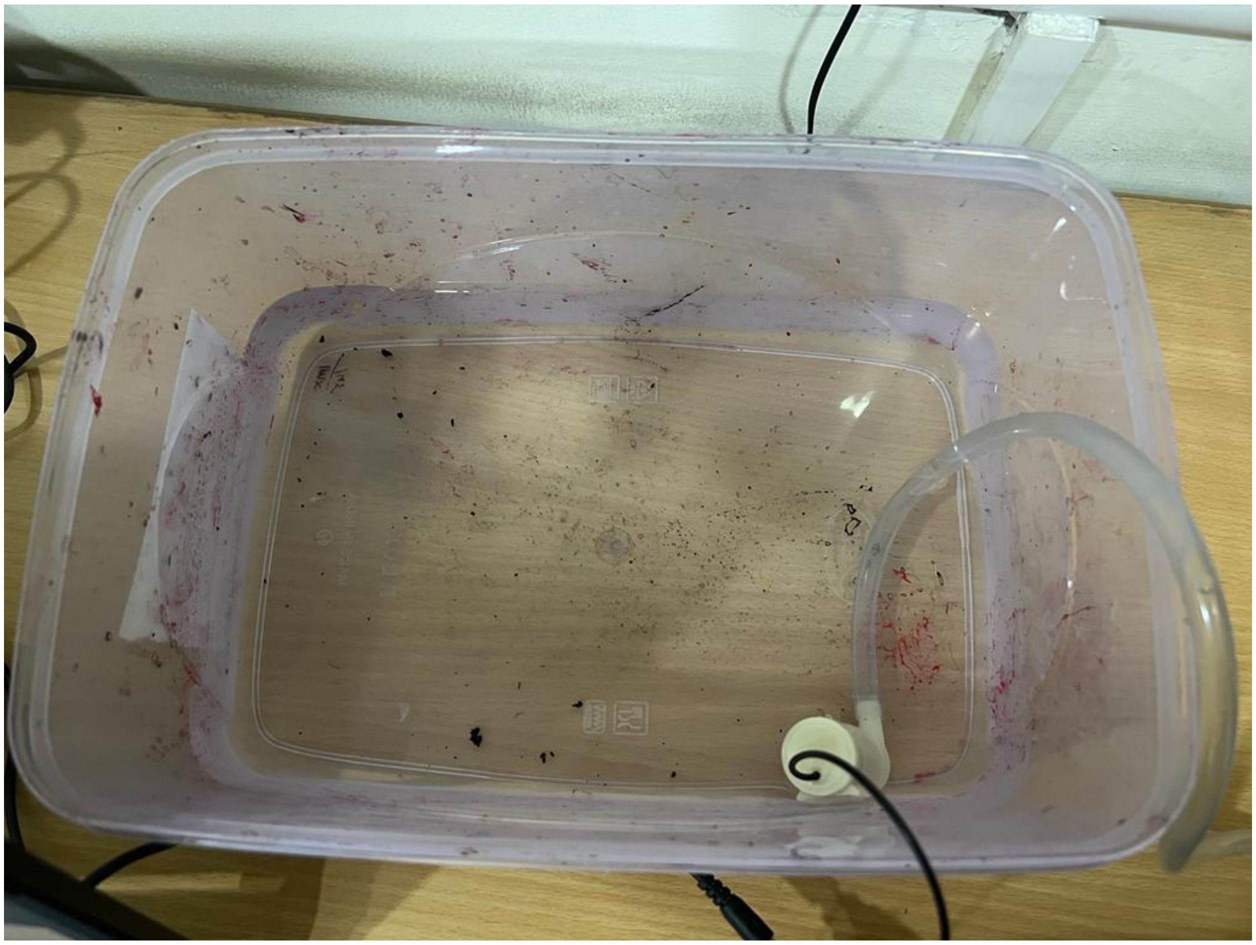
```
results
e(x / seen * 1E3 for x in dt)  # speeds per image
```

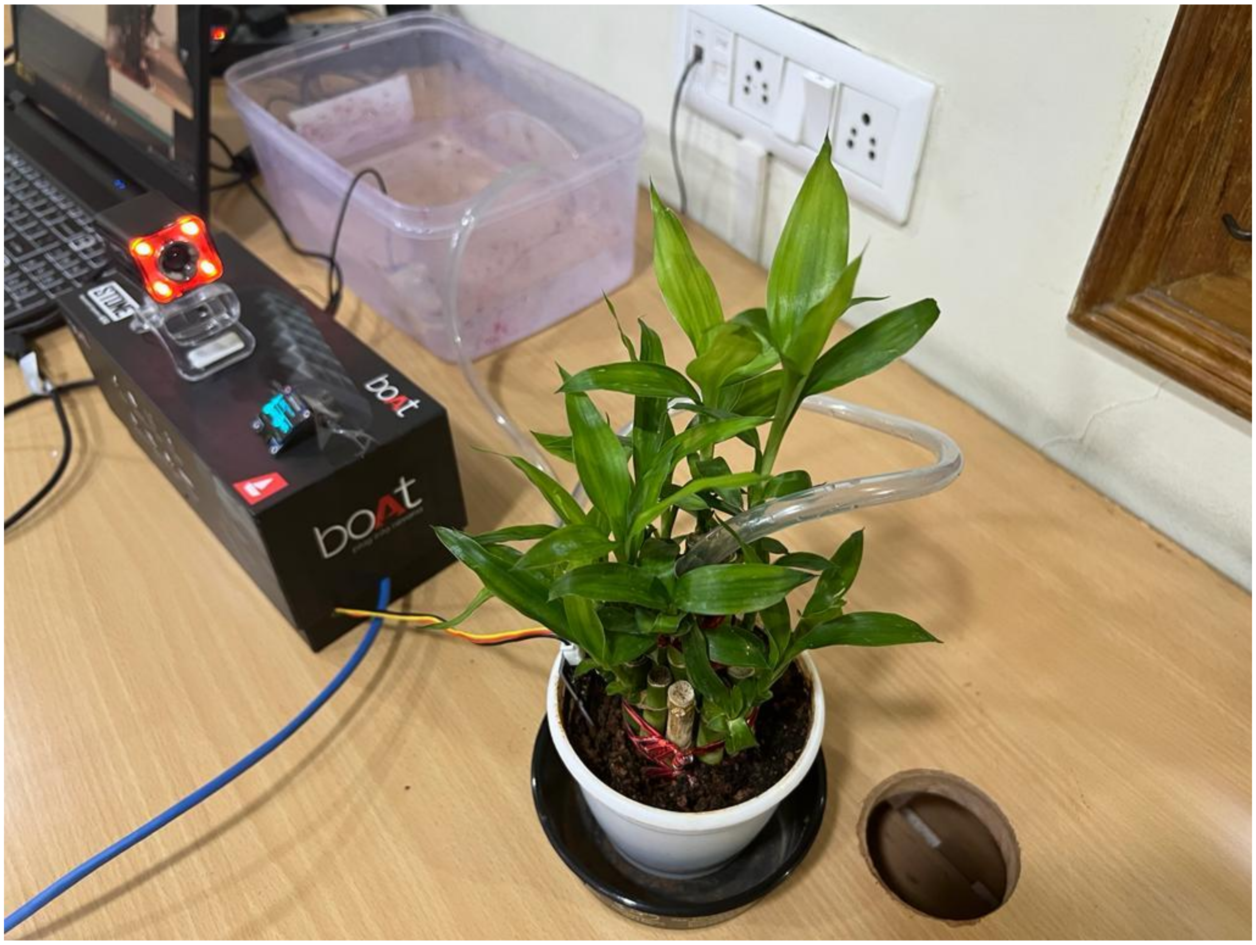
```
else
```

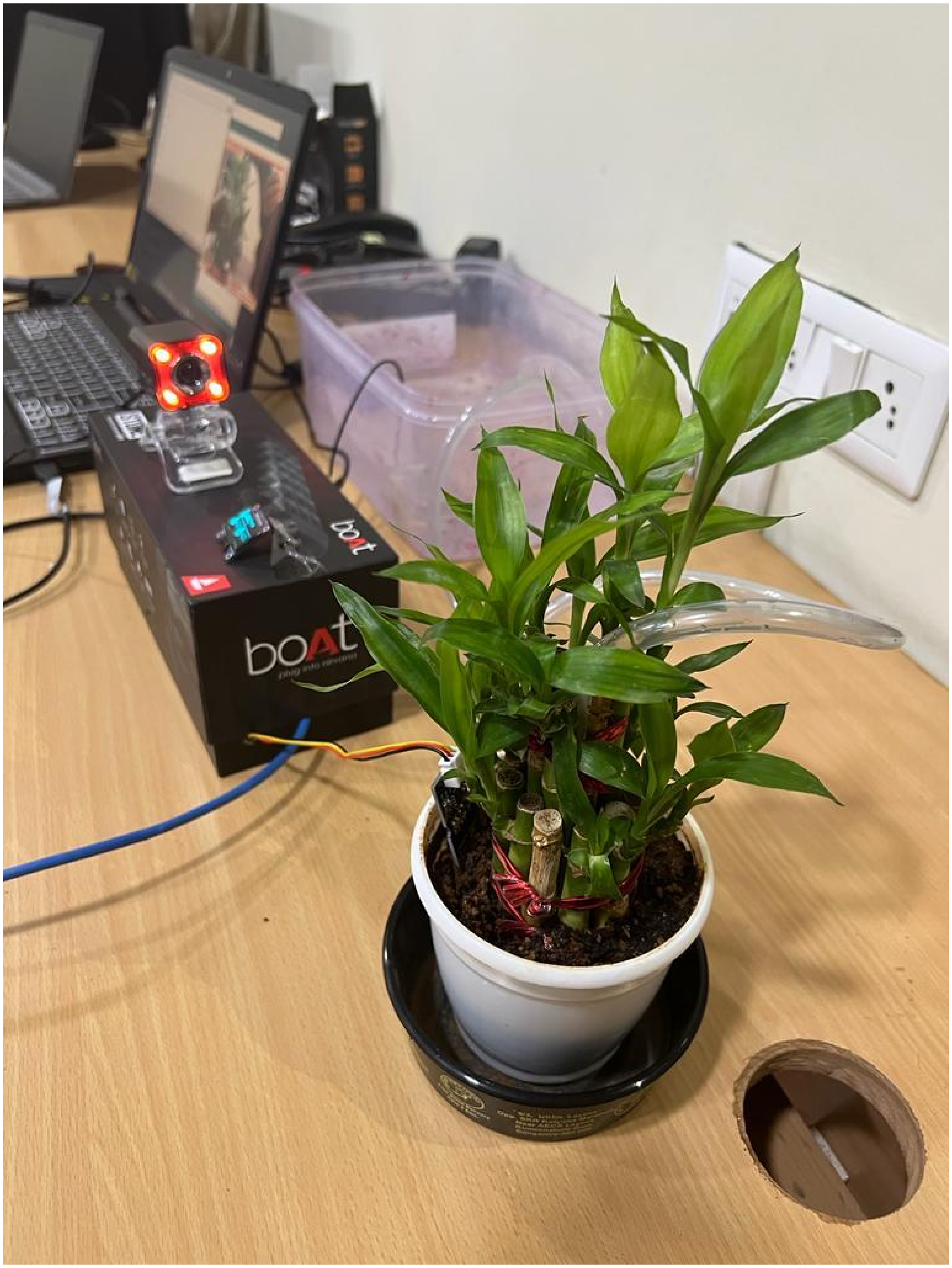
```
... 8 more
Error opening serial port 'COM10'
```

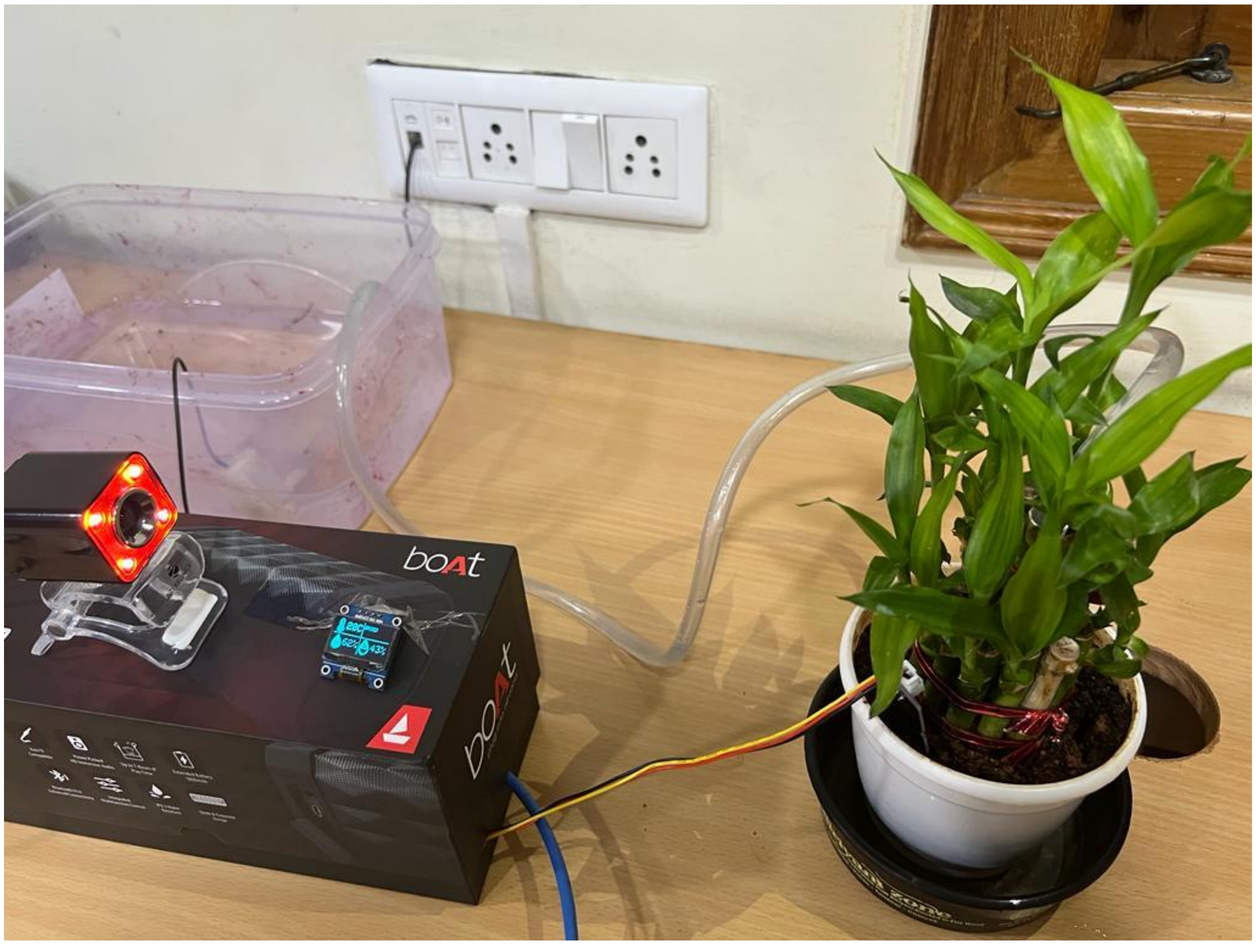
boAt
61%
64%

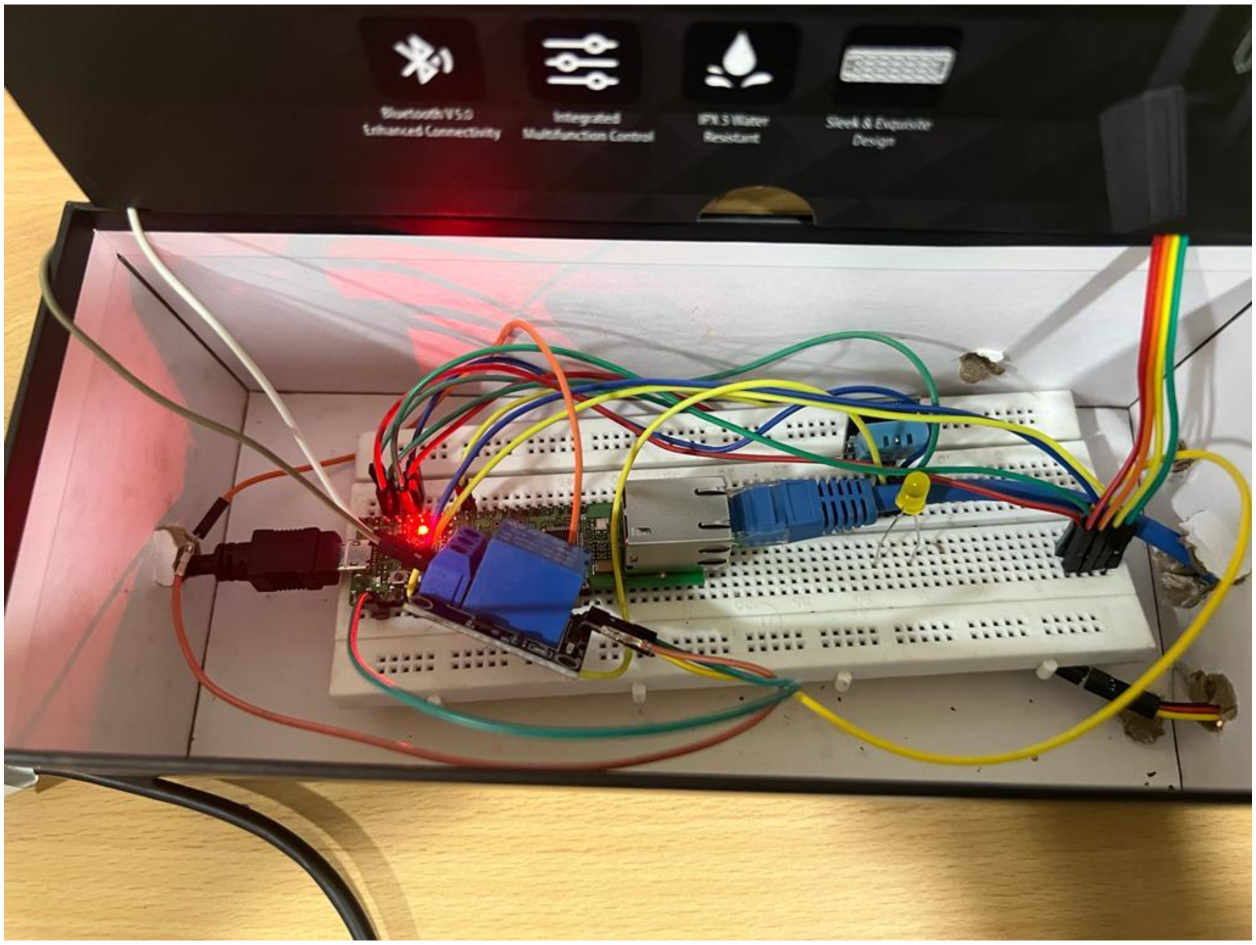

boAt
29C
HEALTHY
62%
66%

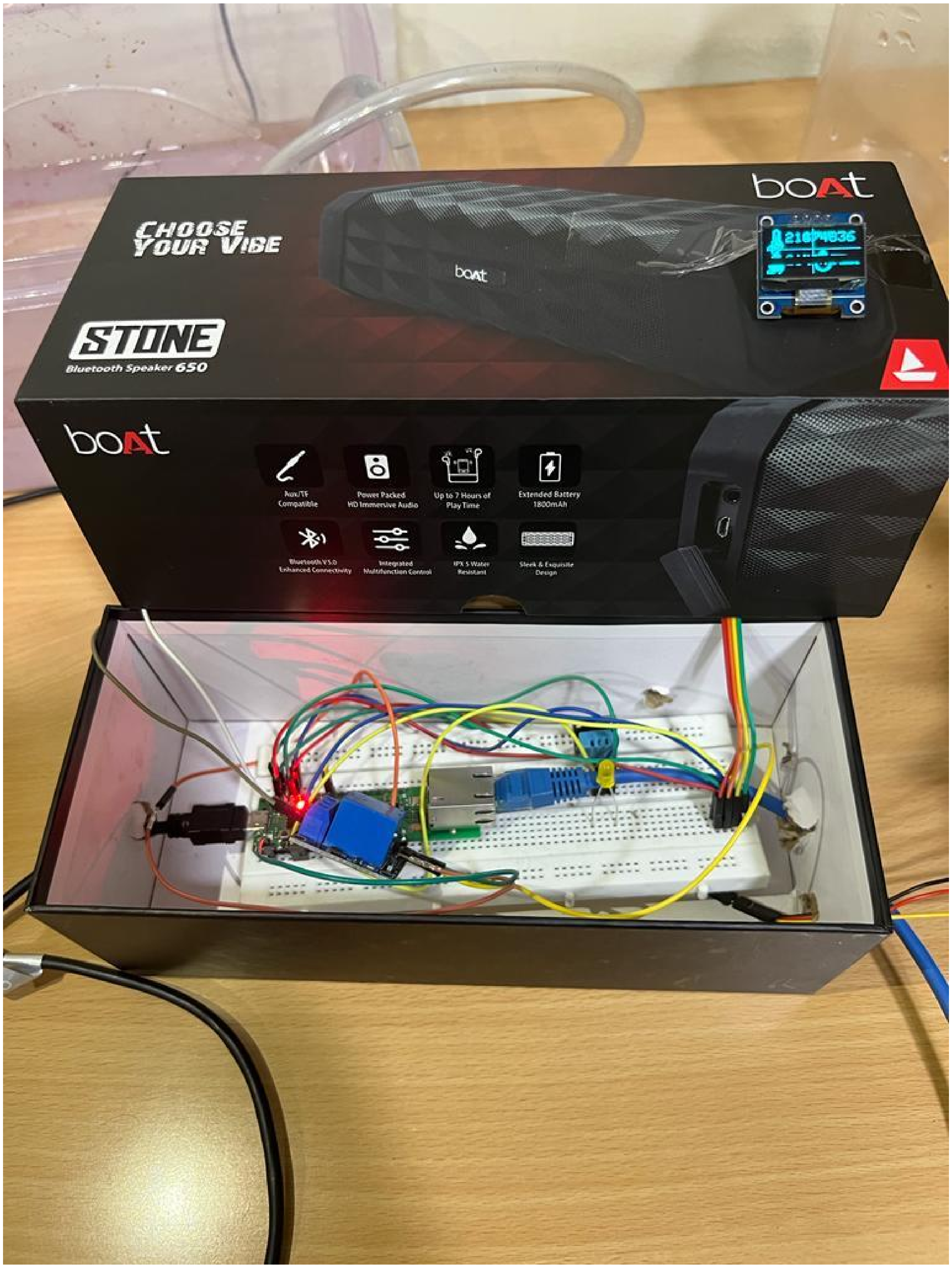



boAt
boAt

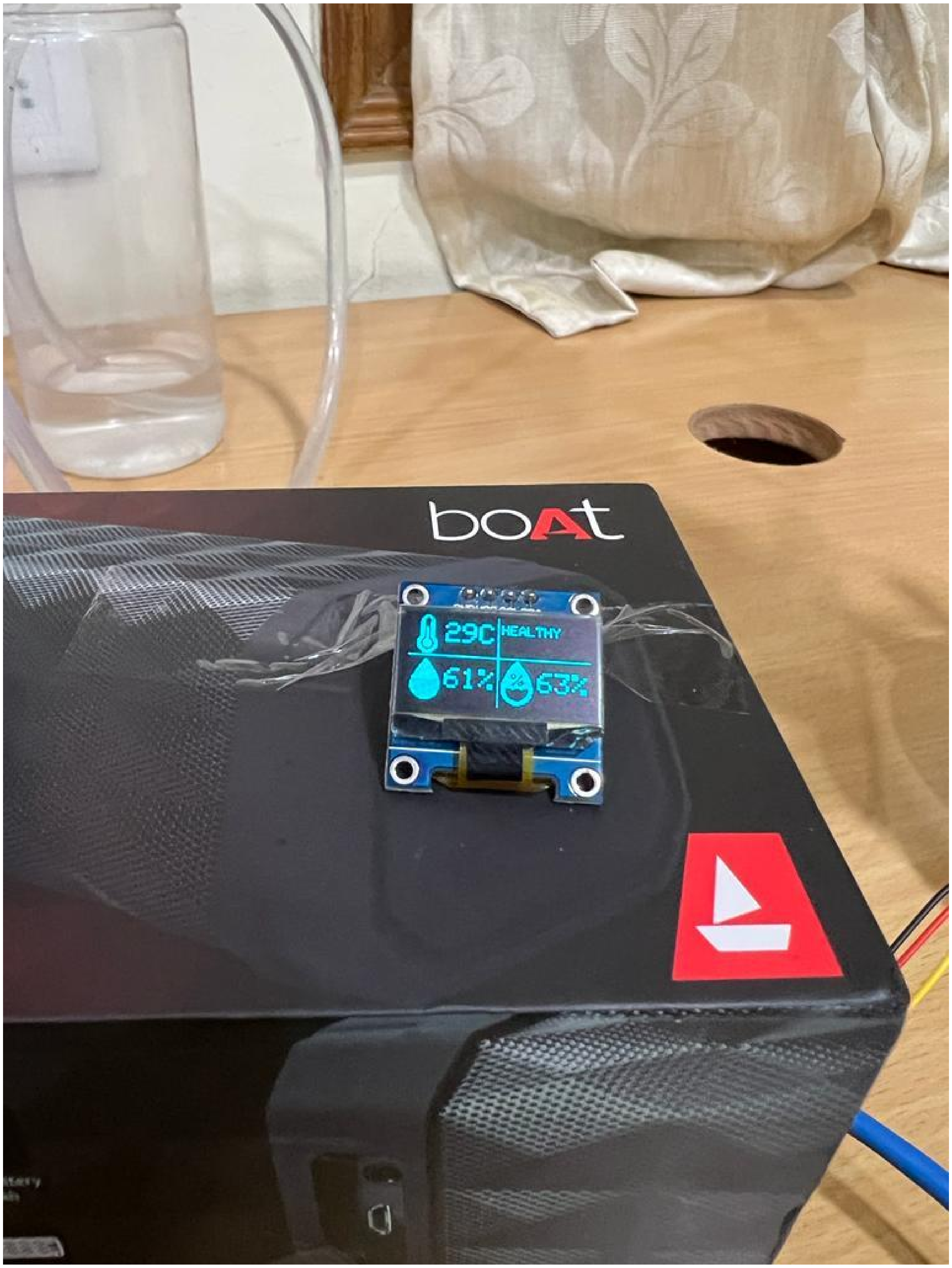

boAt
plug into nirvana
boAt


boAt
68%
43%


Bluetooth V5.0 Enhanced Connectivity
Integrated Multifunction Control
IPX 5 Water Resistant
Sleek & Exquisite Design



boAt


29C
HEALTHY
61%
63%
boAt

Video Link

References

- Pawar, S. B., Rajput, P., & Shaikh, A. (2018). Smart irrigation system using IOT and raspberry pi. *International Research Journal of Engineering and Technology*, *5*(8), 1163-1166.
- https://how2electronics.com/interface-capacitive-soil-moisture-sensor-arduino/
- https://create.arduino.cc/projecthub/pibots555/how-to-connect-dht11-sensor-with-arduino-uno-f4d239
- https://lastminuteengineers.com/oled-display-arduino-tutorial/