Smart Pot

Team members;

Arsal Abbasi

Charles Okere

Muhammad Umer Bin Yaqoob

Motivation

A lot of studies have been done on a wide range of subjects. Our topic requires us to conduct studies on a certain issue and to acquire a generative notion about the plants related problems which allow us to get interested in new ideas and approaches

Introduction

- The number of people purchasing indoor plants is increasing daily
- It can be challenging to maintain them
- Germany imported 213.75 million indoor plants in bloom in 2021
- Smart pot to measure temperature and humidity of the pot

Goals

- Assist the people who like to have indoor plants
- Integrating intelligent solutions
- Providing the user acces to monitor plant‘s enviroment

MQTT

- First version developed by Andy Stanford-Clark in 1999
- Message queuing telemetry transport
- Publish/subscribe messsaging protocol
- Broker, Client, Topic, Publish/Subscribe
- Clients connect to the network
- Publish/Subcribe to to the

Concept Description

- Smartpot that can measure temperature and humidity
- The results will be displayed on android phone application and also raspberry pi which is acting as out mqtt broker
- User will be able to change plants‘s location by sending one message from android phone to smartpot
- The basic operating principle of this procedure depends on the data collected by DHT sensor inside the smart pot.


The smart pot can move
between shaded region
and light region according
to the plants requirement

Hardware

- Raspberry Pi – Broker
- Arduino Uno wifi – Client
- Android Phone – Client
- DHT Sensor

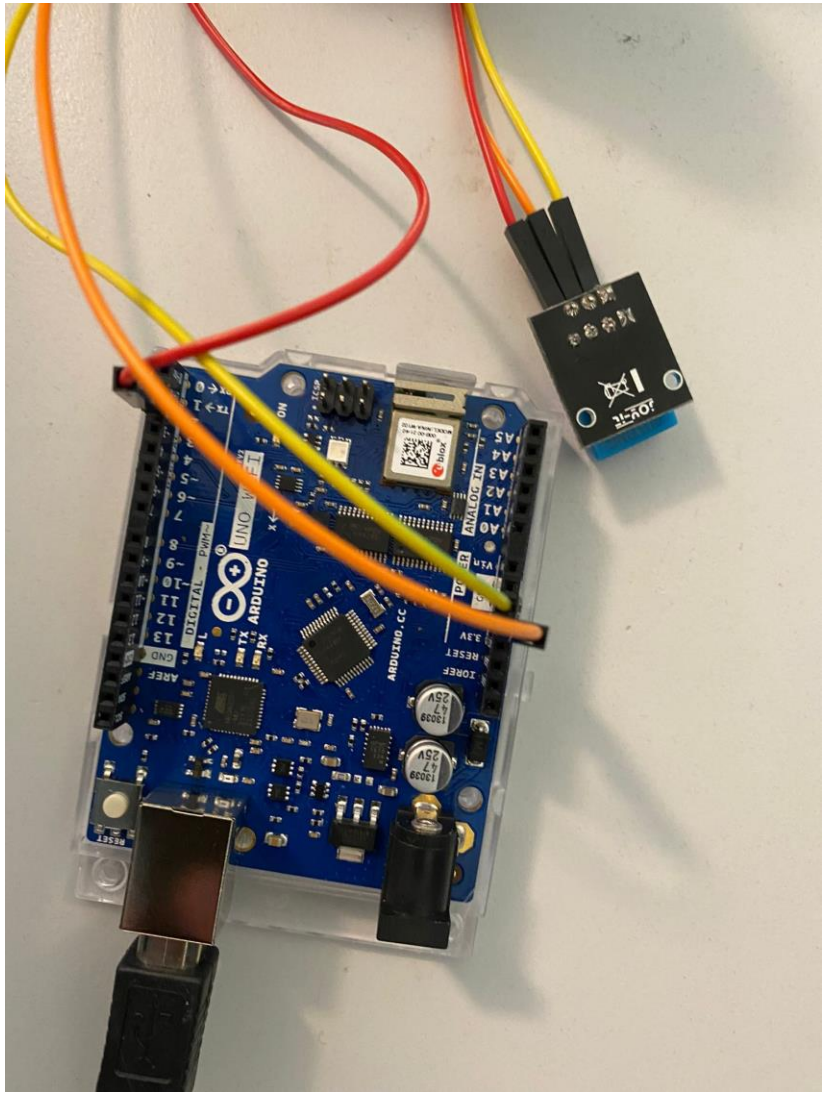

Fig. 1 DHT11 with Arduino uno wifi

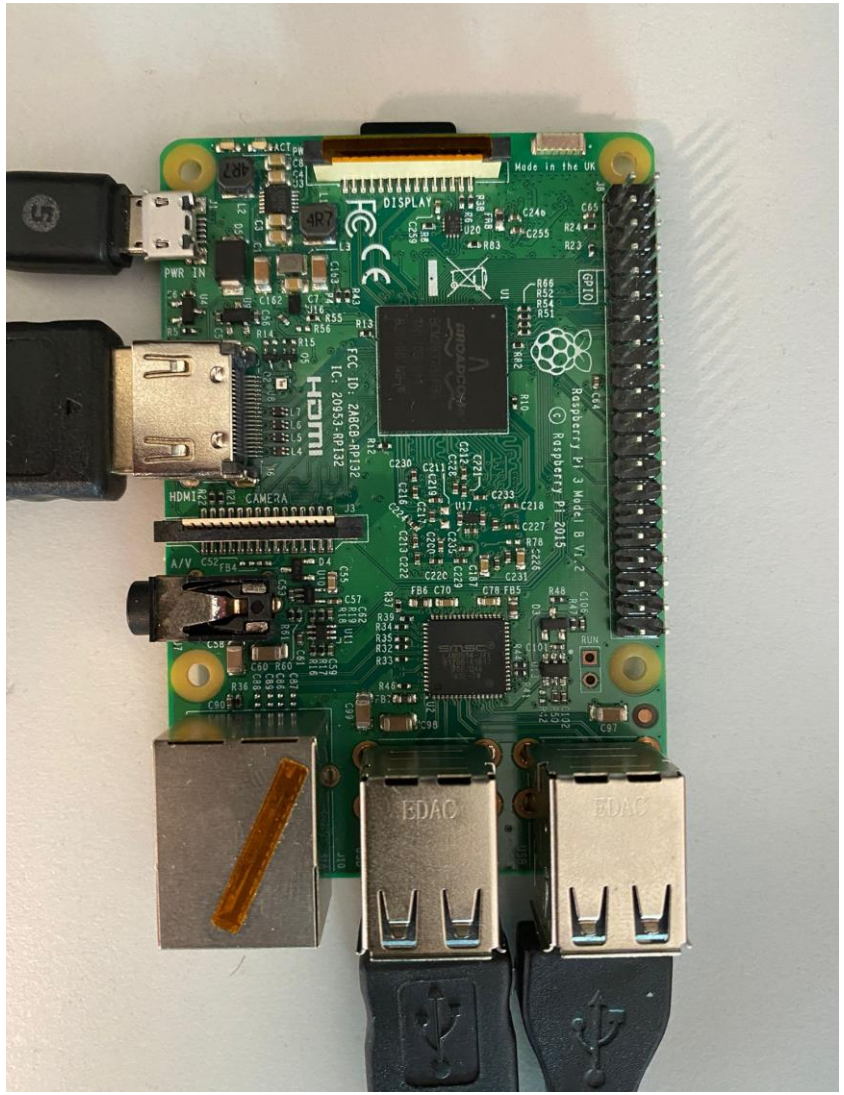

Fig. 2 Raspberry pi

Block Diagram

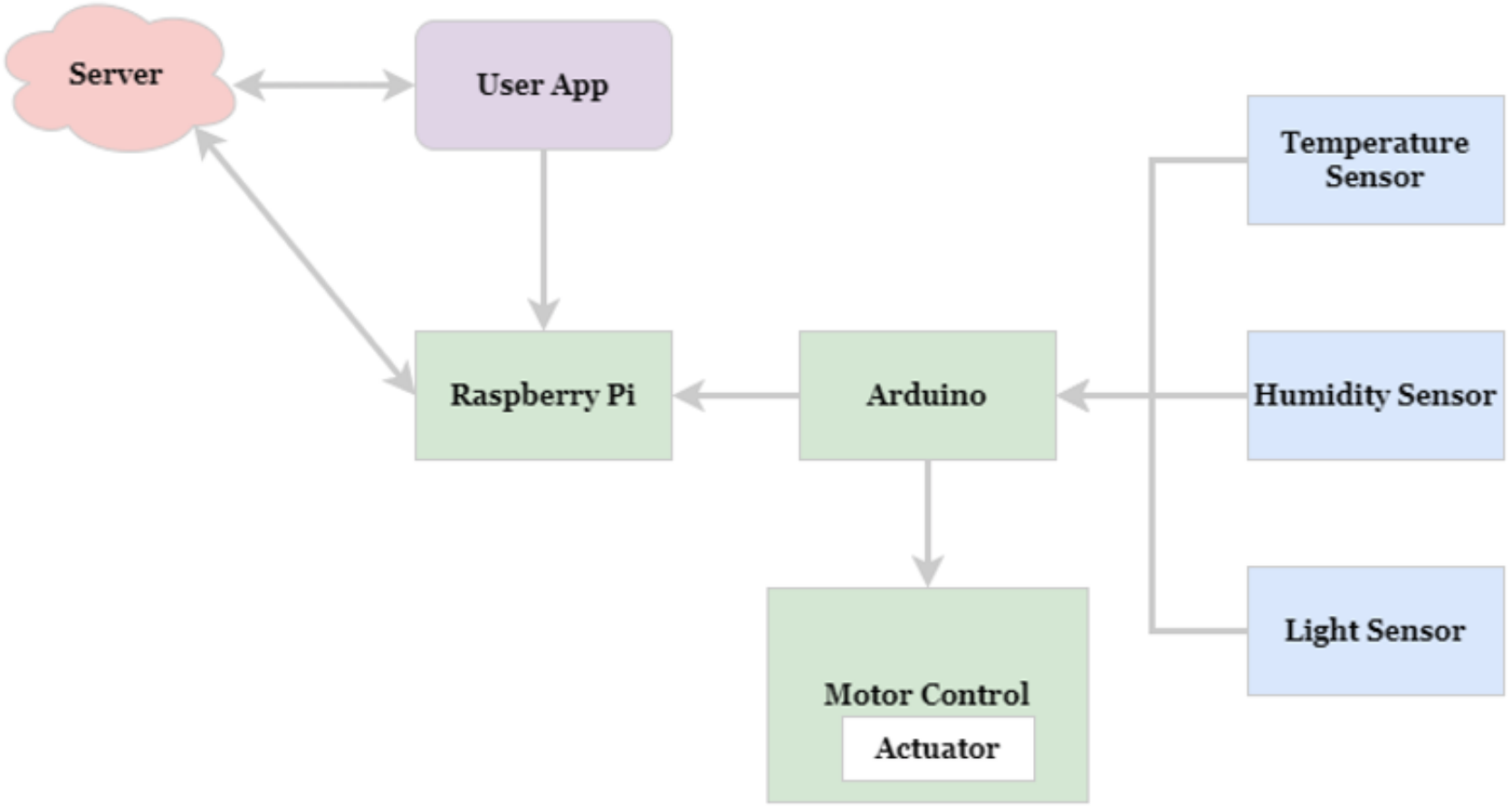

Fig.3 Block Diagram

Class Diagram

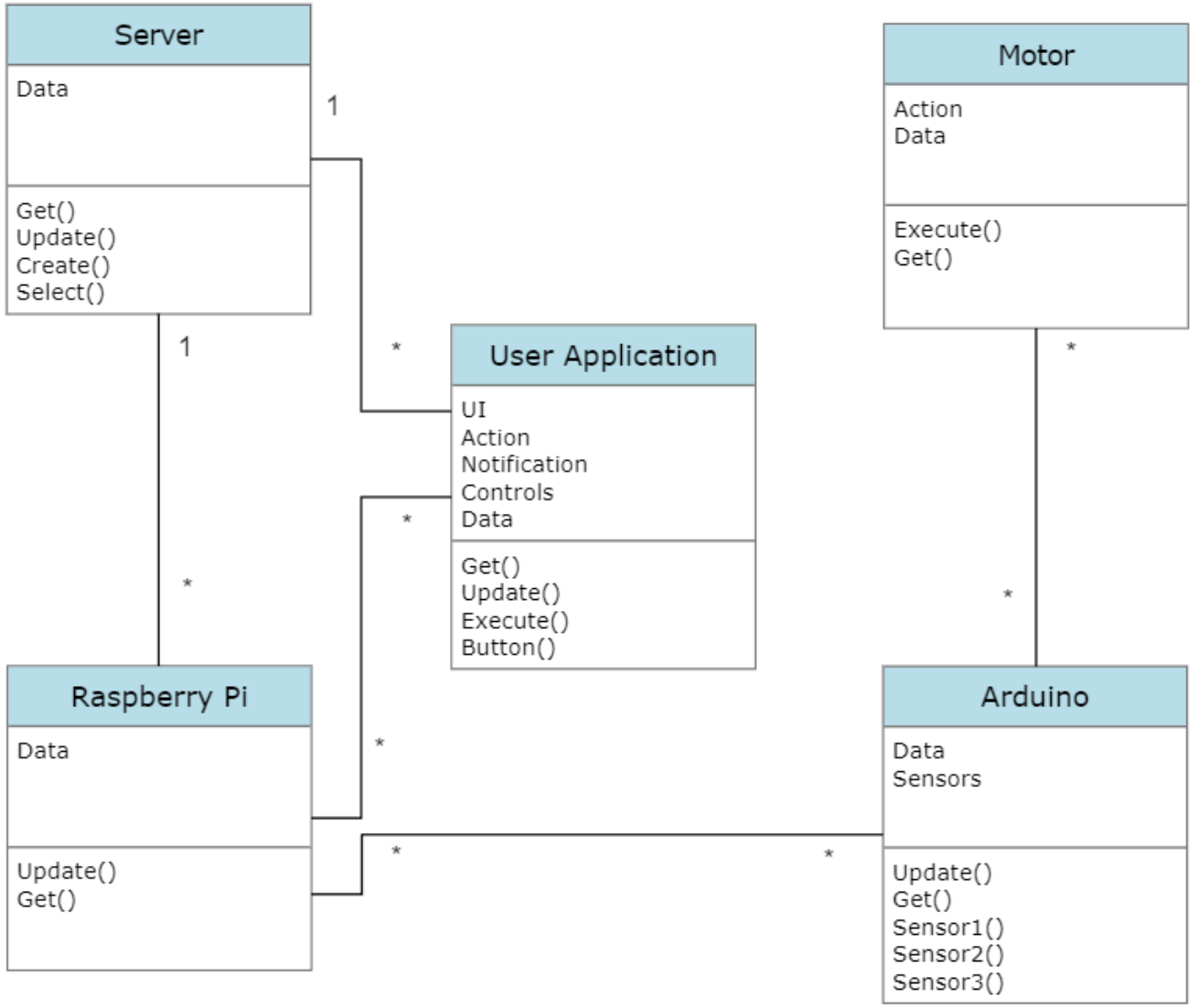

Fig. 5 Class Diagram

Use Case Diagram

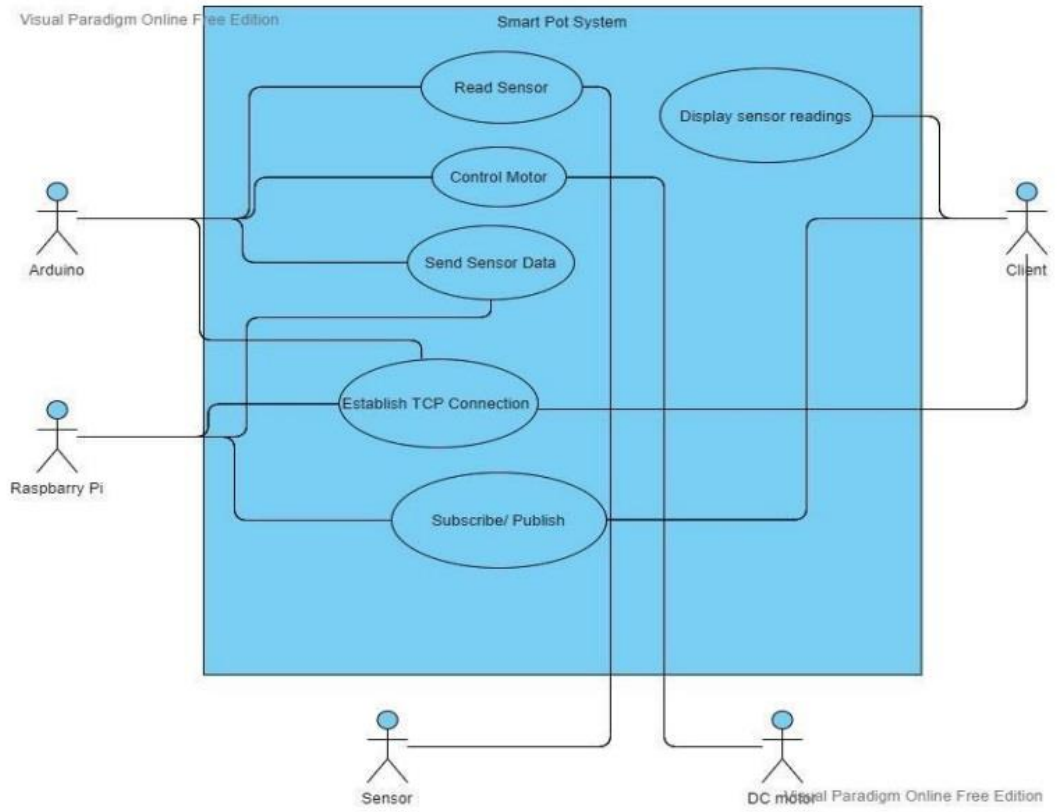

Fig. 4 Use Case Diagram

Methodology

Double Diamond approach;

Discover-Define-Develop and Deliver

Code

Storing Wifi credentials

```
#define SECRET_SSID "umer"
#define SECRET_PASS "Umer1234"
```

Libraries

```
#include <ArduinoMqttClient.h>
#include <WiFiNINA.h>
#include <DHT.h>
#include "arduino_secrets.h"
```

```
// Here the respective input pin is declared
#define DHTPIN 2

// The sensor is initialized
#define DHTTYPE DHT11 // DHT 11
DHT dht(DHTPIN, DHTTYPE);

char ssid[] = SECRET_SSID;        // your network SSID (name)
char pass[] = SECRET_PASS;    // your network password (use for WPA, or use as key for WEP)
char temp = 0;;

WiFiClient wifiClient;
MqttClient mqttClient(wifiClient);

const char broker[] = "192.168.137.55";
int        port     = 1883;
const char topic[]  = "temperature";
const char topic2[]  = "humidity";
const char topic3[]  = "dcmotor";

//set interval for sending messages (milliseconds)
const long interval = 8000;
unsigned long previousMillis = 0;

int count = 0;
float temperature = 25;
float humidity = 25;
```

```
void onMqttMessage(int messageSize);
void setup() {
  //Initialize serial and wait for port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }

  // attempt to connect to Wifi network:
  Serial.print("Attempting to connect to WPA SSID: ");
  Serial.println(ssid);
  while (WiFi.begin(ssid, pass) != WL_CONNECTED) {
    // failed, retry
    Serial.print(".");
    delay(5000);
  }

  Serial.println("You're connected to the network");
  Serial.println();

  Serial.print("Attempting to connect to the MQTT broker: ");
  Serial.println(broker);

  if (!mqttClient.connect(broker, port)) {
    Serial.print("MQTT connection failed! Error code = ");
    Serial.println(mqttClient.connectError());

    while (1);
  }

  Serial.println("You're connected to the MQTT broker!");
  Serial.println();
    dht.begin();


}
void loop() {
  // call poll() regularly to allow the library to send MQTT keep alive which
  // avoids being disconnected by the broker
  mqttClient.poll();

  unsigned long currentMillis = millis();

 if (currentMillis - previousMillis >= interval) {
    // save the last time a message was sent
    previousMillis = currentMillis;
 // Two seconds pause between measurements
 delay(2000);

 // Humidity is measured
 humidity = dht.readHumidity();
 // temperature is measured
 temperature = dht.readTemperature();

 // Checking if the measurements have passed without errors
 // if an error is detected, a error message is displayed here
 if (isnan(humidity) || isnan(temperature)) {
   Serial.println("Error reading the sensor");
   return;
}
    Serial.print("Sending message to topic: ");
    Serial.println(topic);
    Serial.println(temperature);

    Serial.print("Sending message to topic: ");
    Serial.println(topic2);
    Serial.println(humidity);


    // send message, the Print interface can be used to set the message contents
    mqttClient.beginMessage(topic);
    mqttClient.print(temperature);
    mqttClient.endMessage();

    mqttClient.beginMessage(topic2);
    mqttClient.print(humidity);
    mqttClient.endMessage();

      mqttClient.subscribe(topic3);
 // set the message receive callback
  mqttClient.onMessage(onMqttMessage);

 }
```

```
  // call poll() regularly to allow the library to send MQTT keep alive which
  // avoids being disconnected by the broker
  mqttClient.poll();
}



void onMqttMessage(int messageSize) {
  // we received a message, print out the topic and contents
 Serial.println("Received a message with topic '");
  Serial.print(mqttClient.messageTopic());
Serial.println();
Serial.print("', length ");
  Serial.print(messageSize);
  Serial.println(" bytes:");

  while (mqttClient.available()) {
     temp = (char)mqttClient.read();
      Serial.print((char)mqttClient.read());
  }

  Serial.println();
  Serial.println();

}
```

Implementation

https://youtu.be/VUJcxS1a9kw

Results

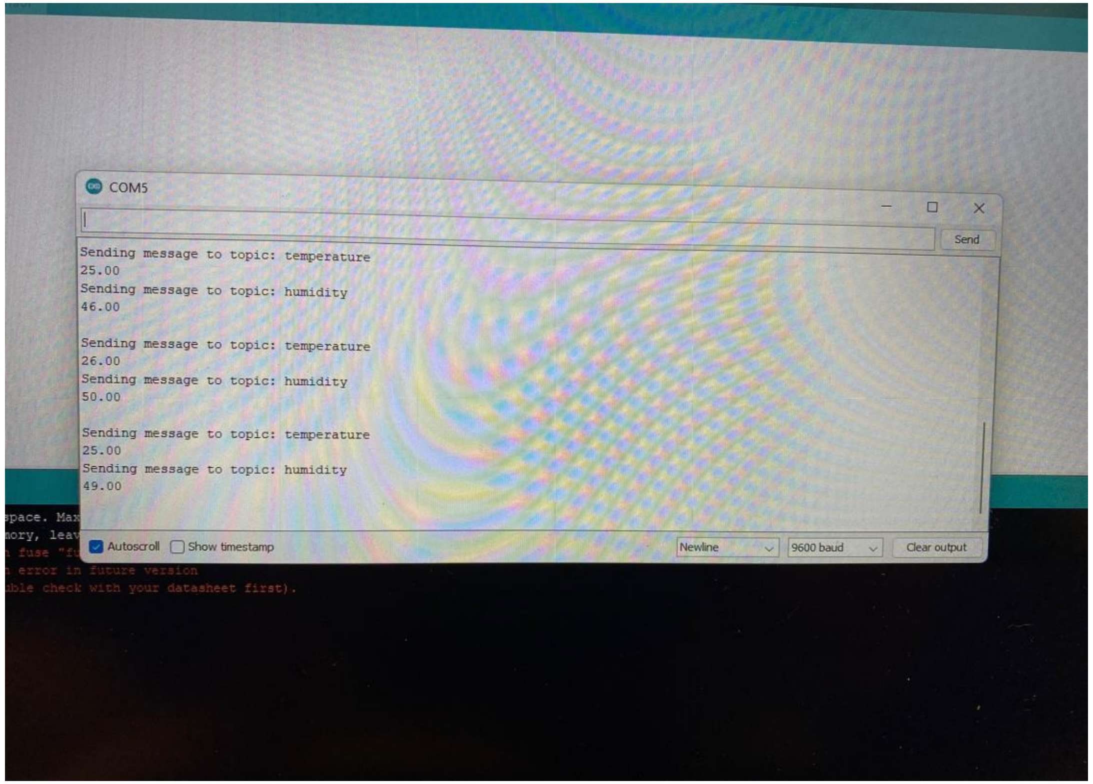

Fig. 6 results

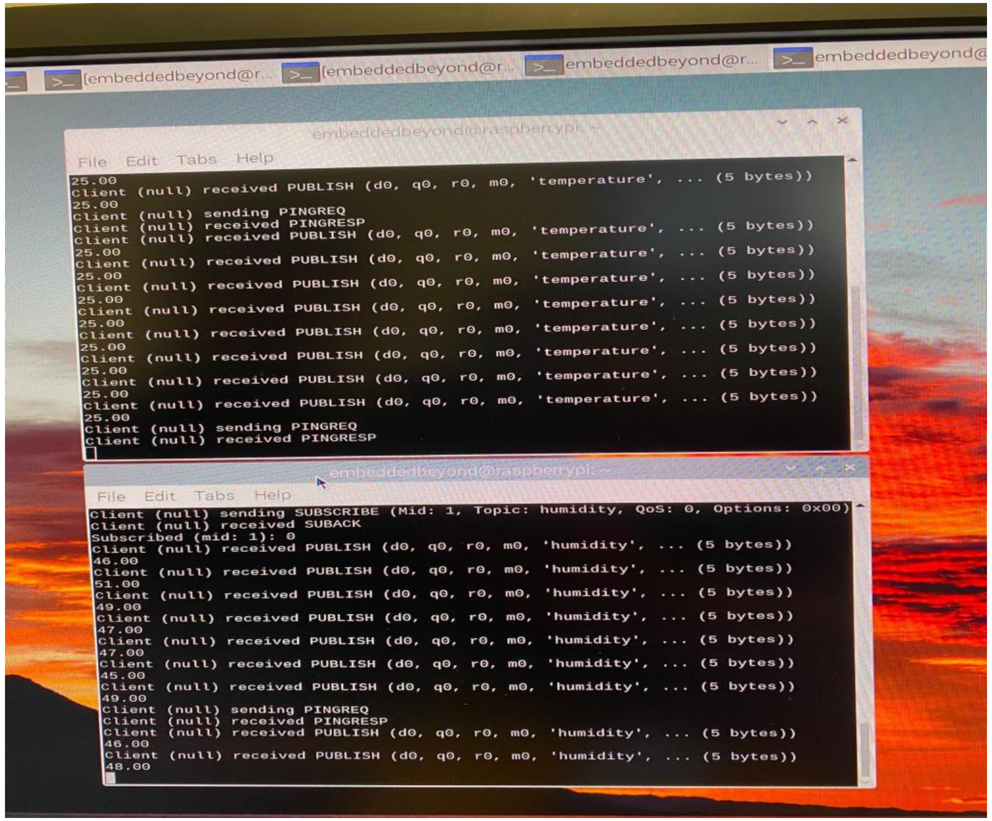

Fig. 7 results raspberry pi subscriber

Results

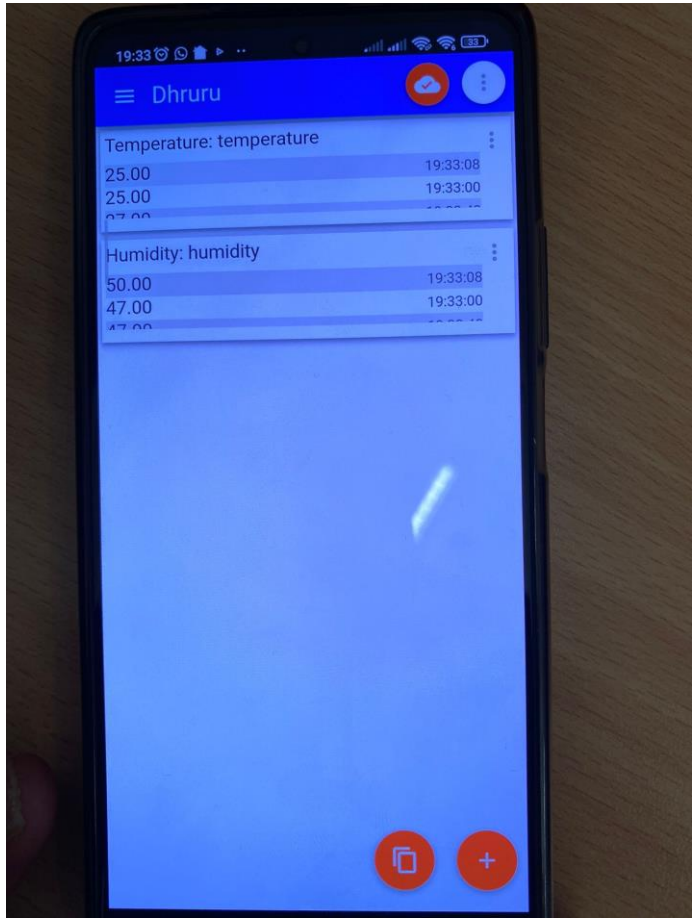

Fig. 8 results android app

Conclusion

- We encountered several problems
- Smart pot will assist our users
- Achieved accurate results

Thankyou for listening!