Department of Electronic and Telecommunication Engineering

University of Moratuwa

EN2160 - Electronic Design Realization

Mini Weather Station

W. A. S. S. Wanigathunga

200693D

July 14, 2023

Abstract

This report details the design and development of a mini weather station project with a web dashboard and mobile application for data monitoring. The project aims to create a compact and user-friendly weather monitoring system. The mini weather station provides real-time measurements of temperature, humidity, atmospheric pressure, and ambient light intensity. The project also incorporated user feedback through a survey to identify areas for improvement and potential future enhancements. The results highlight the successful implementation of the weather station, its functionality, and the importance of user-driven design considerations.

Through the web dashboard and mobile application, users can conveniently access and analyze the weather data. This project showcases the integration of IoT technologies, user-centric design, and mobile accessibility to create an efficient and versatile weather monitoring solution. Overall, this mini weather station project demonstrates the integration of various sensors and components to create a practical and user-friendly device for monitoring local weather conditions.

Contents

1 Introduction

The primary objective was to create a user-friendly and marketable mini weather station, which would serve as a valuable tool for individuals in various settings. Traditional weather stations can be complex and costly, often equipped with numerous features exceeding the needs of everyday users. The demand for a more accessible and cost-effective solution was recognized, leading to the aim of developing a mini weather station catered to typical home usage. Essential weather measurements like temperature, humidity, atmospheric pressure, and ambient light intensity were incorporated into the weather station, offering a practical and compact solution. Additionally, priority was given to the integration of remote monitoring capabilities through a web dashboard and a mobile application, allowing weather data to be conveniently accessed and monitored from anywhere.

2 Methodology

The methodology employed in this report outlines the systematic approach taken to design, develop, and evaluate the Mini Weather Station project.

2.1 Research and Requirement Analysis

During the research phase, I studied about weather monitoring systems, available sensors, and existing technologies. This involved exploring various weather stations currently available in the market.

It was observed that most weather stations on the market are designed with large displays and include extensive features such as wind speed measurement, rainfall data, and advanced forecasting capabilities. While these weather stations provide comprehensive data, they are often costly and may have features that exceed the typical needs of home users.

2.1.1 User need survey

I have done a small user need survey for the requirement analysis and user expectations for the improvements of the existing weather stations in the market through a google form.
Link: https://docs.google.com/forms/

It was found that users primarily desired a cost-effective weather monitoring solution suitable for typical home usage. And accurate measurements, ease of use, and the ability to monitor data remotely are important factors to be considered.

Considering these findings, the mini weather station project aimed to address these user requirements by providing a compact and affordable solution. Additionally, the project prioritized the development of a user-friendly interface and the ability to monitor data remotely through a web dashboard and a mobile application.

Requirements:

- Cost effective
- Accurate

- Ability to monitor data remotely
- Ease of use
- Useful for typical home usage
- User-friendly interface

2.2 Component Selection and System Design

Variety of sensors are available for measuring temperature, humidity, atmospheric pressure, and ambient light intensity. According to the requirements analyzed, following sensors are selected.

1. Temperature - BMP280
2. Humidity - HTU21D
3. Atmospheric Pressure - BMP280
4. Ambient Light Intensity - LDR sensor

2.2.1 Temperature, Humidity and Pressure Measurements

There are many sensors available for these measurements : BME280, BMP280, HTU21D, DHT11, DHT22 and etc. Following table shows the results obtained by comparing the sensors for the fulfillment of the observed requirements.

Sensor Type	Range	Resolution	Accuracy	Pre-Selection
AM2320	−40 °C to +80 °C	0.1 °C	±0.5 °C	passed
BMP280	−40 °C to +85 °C	0.01 °C	±0.5 °C	passed
BME280	−40 °C to +85 °C	0.01 °C	±0.5 °C	passed
DHT11 [1]	0 °C to +50 °C	1 °C	±2 °C	failed
DHT11 [2]	−20 °C to +60 °C	0.1 °C	±2 °C	failed
DHT22	−40 °C to +80 °C	0.1 °C	±0.5 °C	passed
HTU21D	-40 °C to +125 °C	0.01 °C	±0.3 °C	passed
HDC1080	−40 °C to +125 °C	0.1 °C	±0.2 °C	passed
SHT30	−40 °C to +125 °C	0.01 °C	±0.2 °C	passed
SHT35	−40 °C to +125 °C	0.01 °C	±0.1 °C	passed
Si7021	−40 °C to +125 °C	0.01 °C	±0.4 °C	passed

Figure 1: Temperature Sensors

Sensor Type	Range	Resolution	Accuracy	Pre-Selection
AM2320	0% to 99.9%	0.1%	±3%	passed
BME280	0% to 100%	0.008%	±3%	passed
DHT11 [1]	20% to 90%	1%	±5%	failed
DHT11 [2]	5% to 95%	1%	±5%	failed
DHT22	0% to 100%	0.1%	±2%	passed
HTU21D	0% to 100%	0.04%	±2%	passed
HDC1080	0% to 100%	0.1%	±2%	passed
SHT30	0% to 100%	0.01%	±2%	passed
SHT35	0% to 100%	0.01%	±1.5%	passed
Si7021	0% to 100%	0.025%	±3%	passed

Figure 2: Humidity Sensors

Sensor Type	Range [1]	Resolution [1]	Accuracy [1]	Accuracy [2]	Pre-Selection
BMP280	300 hPa to 1100 hPa	0.0016 hPa	1 hPa	0.01 hPa	passed
BME280	300 hPa to 1100 hPa	0.0016 hPa	1 hPa	0.01 hPa	passed

Figure 3: Pressure Sensors

Considering the above data BME280 was selected as the best matching sensor. But there was a problem with it's price and the market availability. Therefore as a solution for that problem, BMP280 sensor was selected along with the HTU21D sensor for humidity measurements as the BMP280 can only measure temperature and atmospheric pressure.

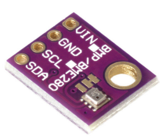


Figure 4: BMP280 Sensor

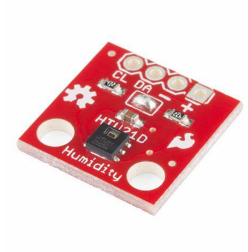


Figure 5: HTU21D Sensor

2.2.2 Light Intensity Measurements

For light intensity measurements, photo-resistors, photo-diodes or photo-transistors can be used. Considering the cost and ease of availability, LDR sensor was selected.

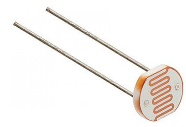


Figure 6: LDR Sensor

2.2.3 Sensor Integration

To ensure seamless integration and efficient communication with the microcontroller (ESP32), the I2C (Inter-Integrated Circuit) protocol is employed. Both the BMP280 and HTU21D sensors support I2C communication, allowing for straightforward connectivity to the ESP32. This ensures compatibility and enables the microcontroller to effectively collect data from these sensors.

To optimize power consumption, a 10kohm series resistor is connected with the LDR sensor, regulating the current flow without sacrificing accuracy. The LDR, along with the resistor, is connected to an ESP32 I/O pin, allowing the microcontroller to measure the voltage across the LDR and determine the light intensity.

2.3 Conceptual Designs

2.3.1 Conceptual Design -1

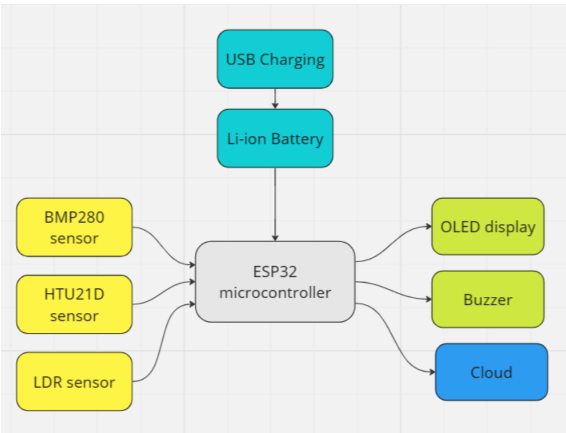


Figure 7: Block Diagram - 1

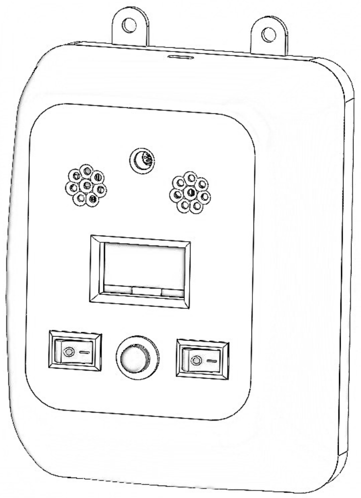


Figure 8: Enclosure - 1

2.3.2 Conceptual Design -2

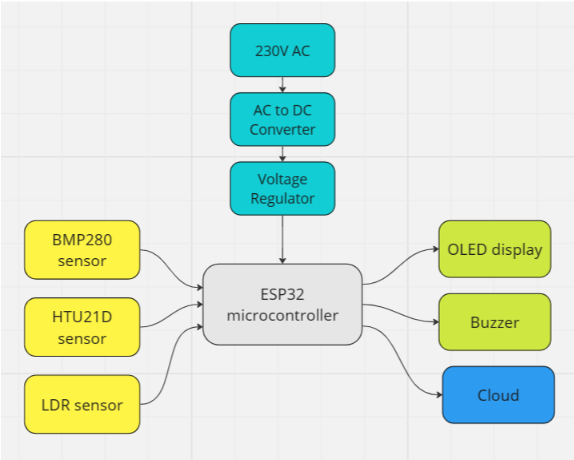


Figure 9: Block Diagram - 2

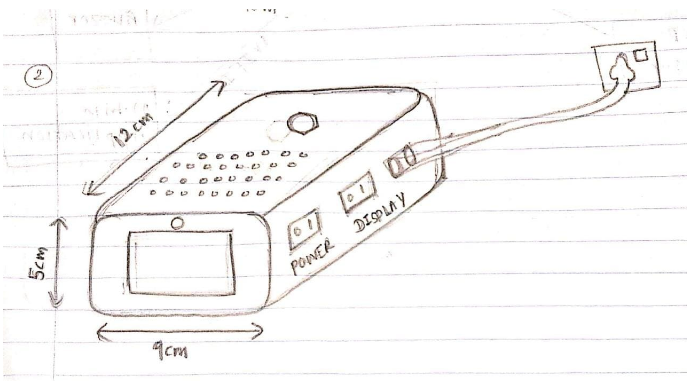


Figure 10: Enclosure - 2

2.3.3 Conceptual Design -3

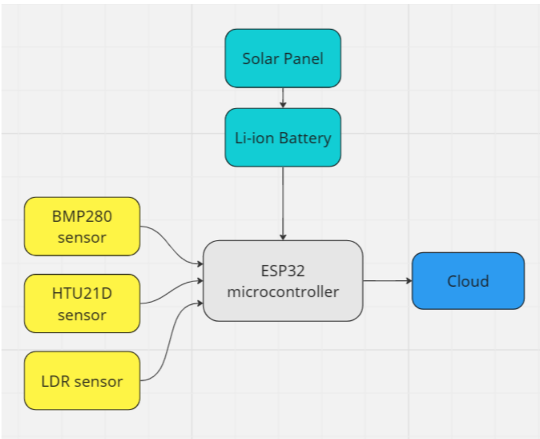


Figure 11: Block Diagram - 3

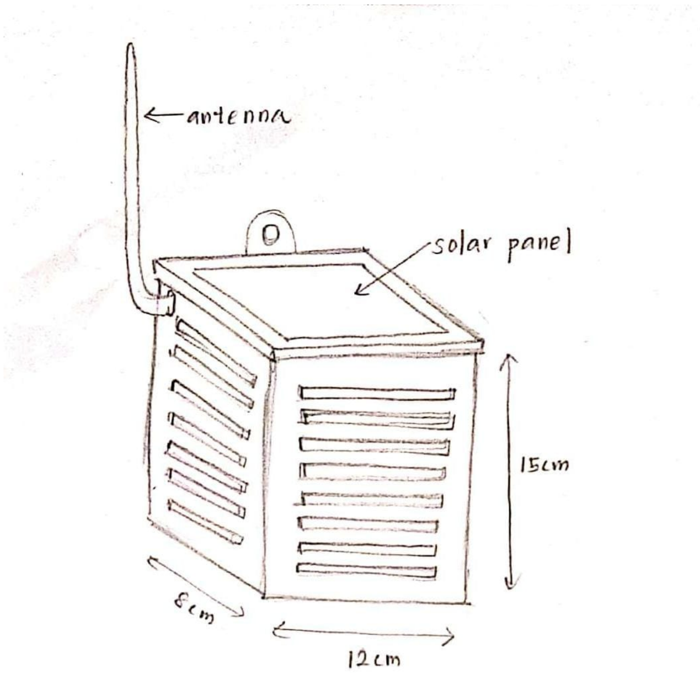


Figure 12: Enclosure - 3

2.3.4 Evaluation Process

	Design - 1	Design - 2	Design - 3
1. Size and dimentions	8	6	7
2. Material and durability	8	9	5
3. Cost effectiveness	8	8	5
4. Heat decipation	7	8	9
5. Aesthetics	8	7	7
6. Customizability	7	5	7
7. Power requirements	7	8	7
8. Manufacturing feasibility	8	8	8
9. Functionality	9	9	8
10. Reliability	9	8	6
Total	79	76	69

Figure 13: Evaluation

After considering many other evaluation criteria, conceptual design - 1 has been selected as the best design.

2.4 Schematic Design and PCB Layout

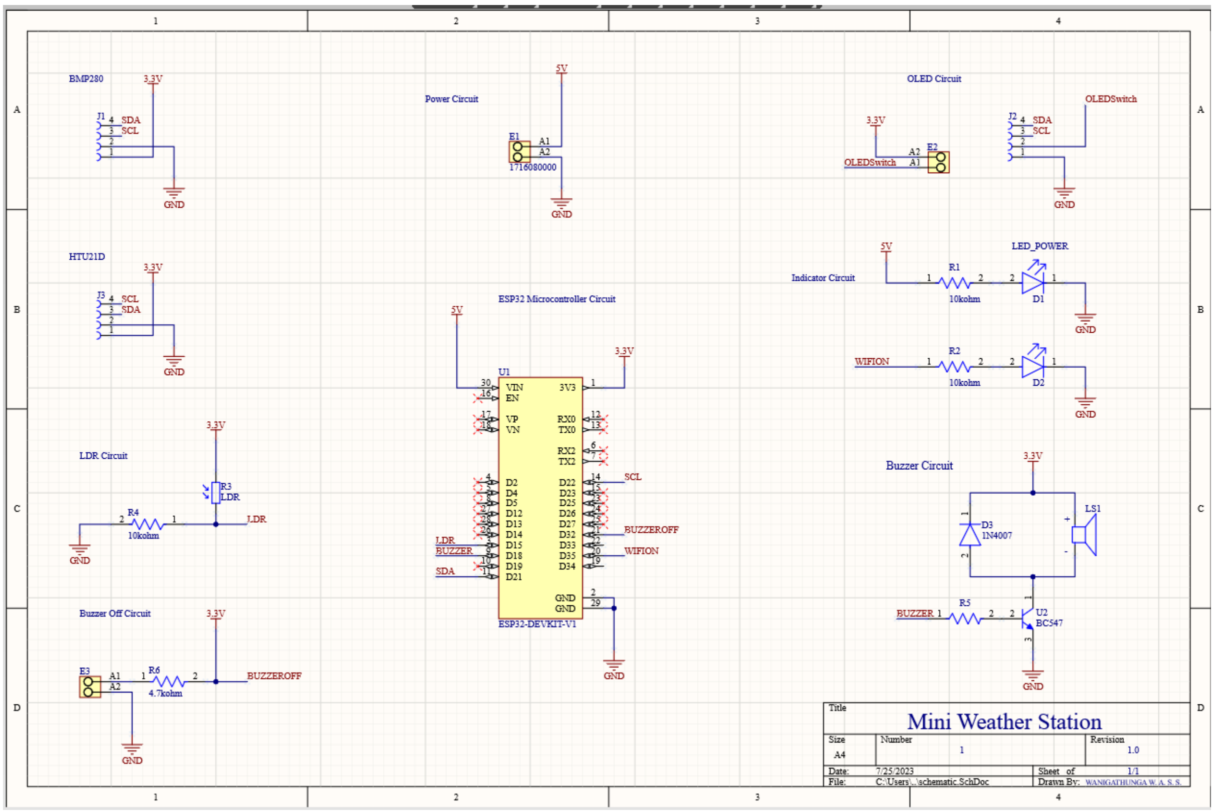


Figure 14: Schematic Design

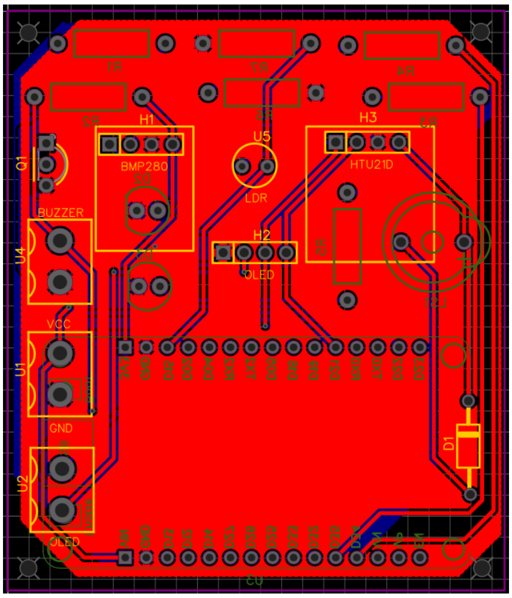


Figure 15: PCB Layout - Top Layer

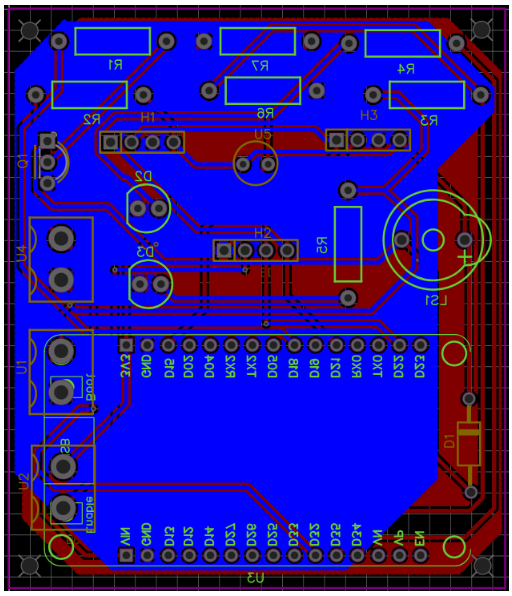


Figure 16: PCB Layout - Bottom Layer

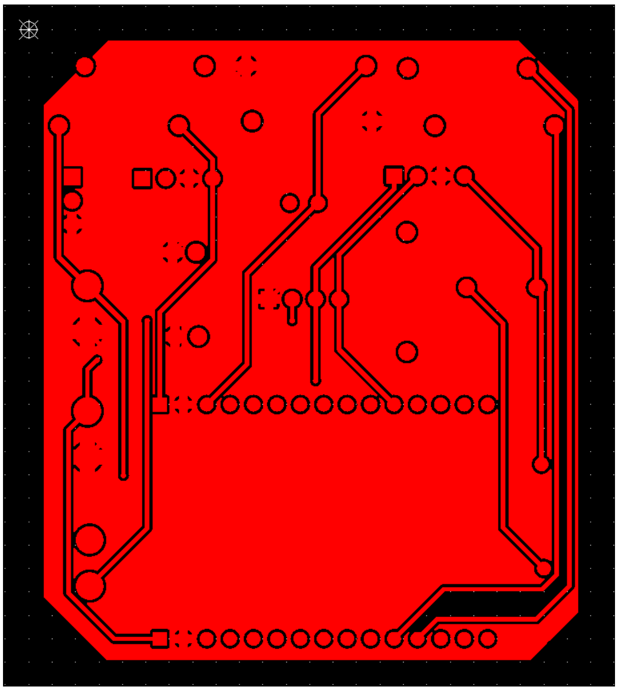


Figure 17: Gerber Top Layer

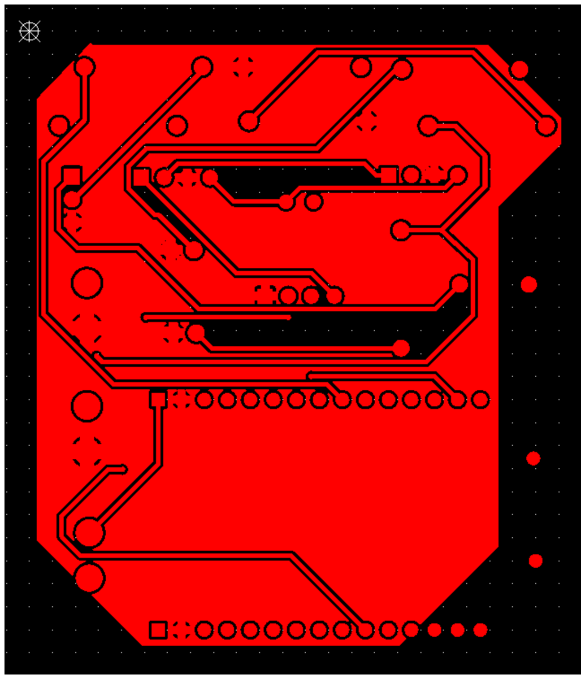


Figure 18: Gerber Bottom Layer

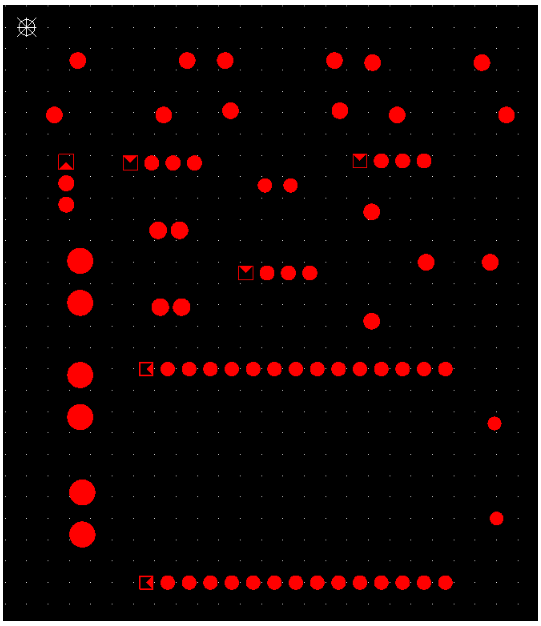


Figure 19: Gerber Top Solder Mask Layer

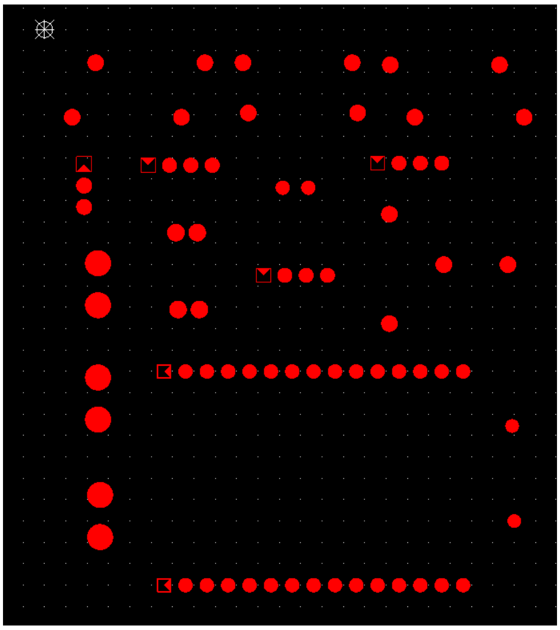


Figure 20: Gerber Bottom Solder Mask Layer

2.5 Enclosure Design

2.5.1 Hand Sketches

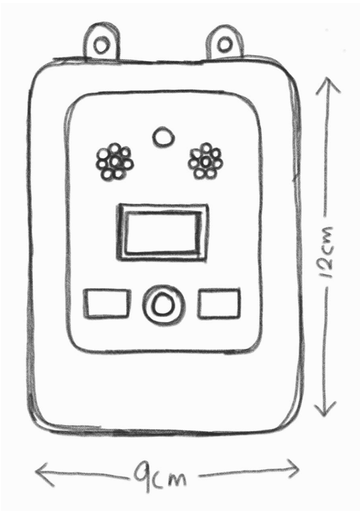


Figure 21: Sketch - 1

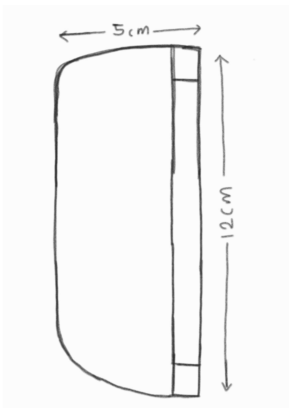


Figure 22: Sketch - 2

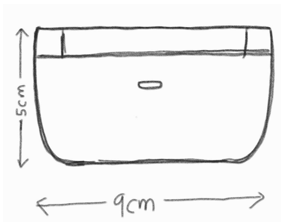


Figure 23: Sketch - 3

2.5.2 SOLIDWORKS Design

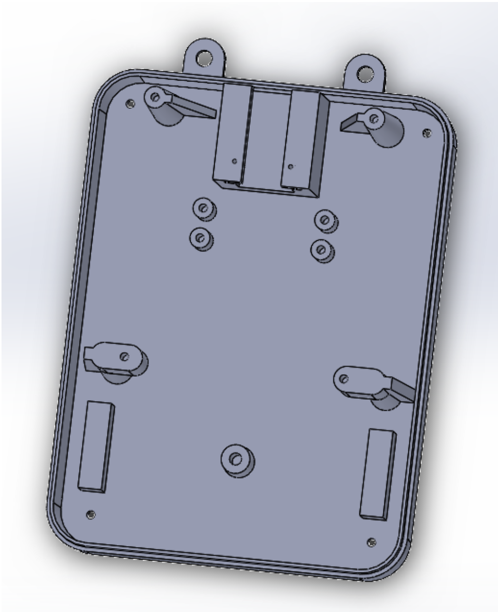


Figure 24: Bottom Part

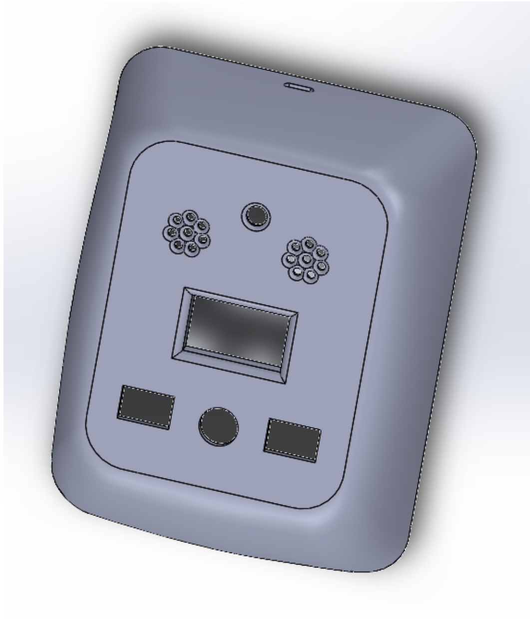


Figure 25: Top Part

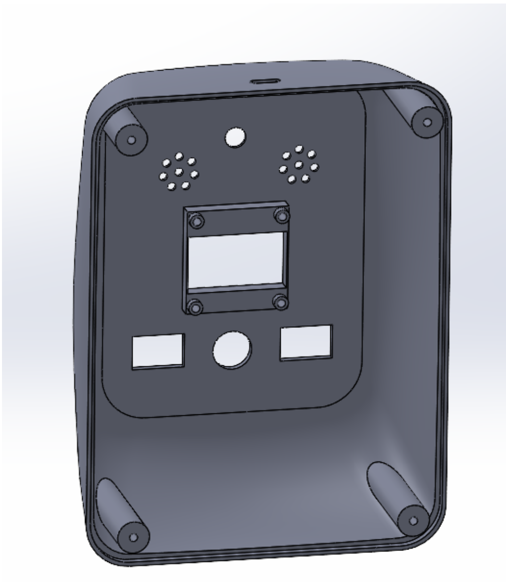


Figure 26: Inside Top Part

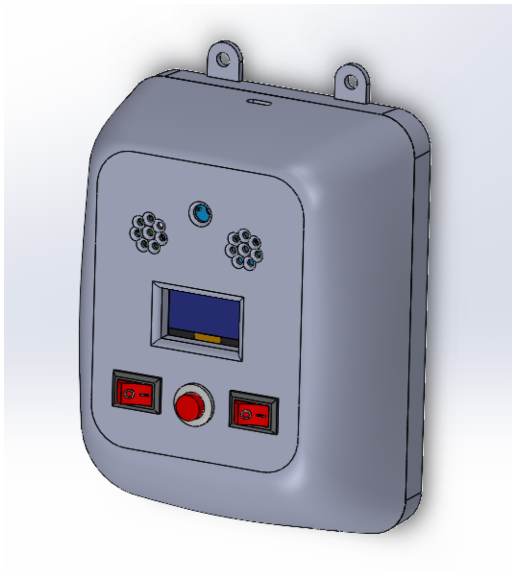


Figure 27: Assembly

2.6 Software Implementation and User Interface Design

In the software development phase, the firmware for the mini weather station was meticulously coded using the Arduino IDE, along with the ESP32 libraries. The firmware was designed to interface seamlessly with the sensors, collect data, and control the OLED display and buzzer functionalities.

To ensure user-friendly data presentation, the data processing algorithms were implemented, including calibration and temperature compensation. The data collected from sensors such as light intensity and humidity were converted to user-friendly units, specifically percentages, for easy understanding.

WiFi functionality was integrated into the firmware to enable data transmission to the Blynk IoT platform. The mini weather station sends real-time weather data to the Blynk platform, where users can access the information with interactive graphs, gauges, and other visualization tools. This allows users to monitor weather conditions remotely through either the mobile application or the web dashboard provided by Blynk.

Figure 28: Mobile Application Interface

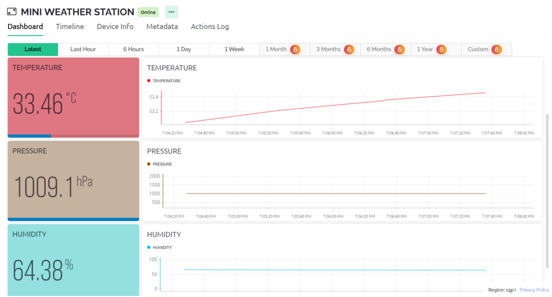


Figure 29: Web Dashboard Interface

The user interface was carefully designed to facilitate a seamless user experience. When the mini weather station is turned on, it acts as an Access Point (AP) named "AutoConnect." Users can connect to this AP using their devices, access the configuration page at "192.168.4.1," and input their WiFi credentials. Once saved, the OLED display will activate and show real-time weather data. Simultaneously, the weather data is sent to the Blynk platform, ensuring continuous data monitoring and logging.

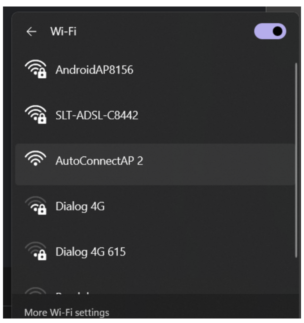


Figure 30: ESP32 Access Point

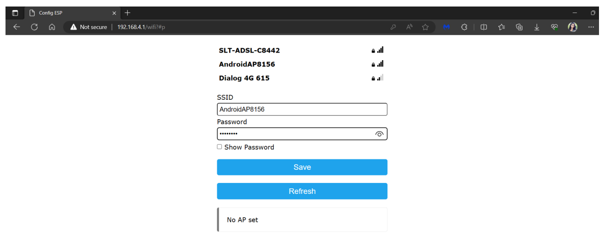


Figure 31: Setting WiFi Credentials

The mini weather station can be charged conveniently using a micro USB cable and a 5V adapter. Users can power off the OLED display when desired, but the device remains operational, continuing to send data to the cloud as long as it is powered on.

Overall, the mini weather station's user-friendly software and interface provide a seamless data monitoring experience, allowing users to access weather data through the mobile application or web dashboard, both of which can be accessed using a provided code at the time of purchase.

2.7 Bill of Materials and Supply Chain

Component	Supplier	Unit price($)	Quantity	Total($)
ESP32 DevKit V1	lcsc.com	10.6719	1	10.6719
BMP280 module	lcsc.com	4.99	1	4.99
HTU21D module	lcsc.com	3.18	1	3.18
LDR sensor	lcsc.com	0.0380	1	0.0380
0.96 inch 128X64 OLED Display	lcsc.com	2.72	1	2.72
$10k\Omega$ resistors	lcsc.com	0.0088	3	0.0264
$4.7k\Omega$ resistors	lcsc.com	0.0129	1	0.0129
220Ω resistor	lcsc.com	0.0138	1	0.0138
Buzzer	lcsc.com	0.2724	1	0.2724
IN40007 Diode	lcsc.com	0.0168	1	0.0168
BC547 Transistor	lcsc.com	0.0920	1	0.0920
5V DC Boost Converter	Tronic.lk	0.85	1	0.85
Power Bank Module	Tronic.lk	0.36	1	0.36
ON/OFF Switches	Tronic.lk	0.045	2	0.09
Push Button	Tronic.lk	0.51	1	0.51
1800mAh Li-ion Battery	Tronic.lk	1.81	1	1.81
Battery holder case for Li-ion battery	Tronic.lk	0.21	1	0.21
Female Headers	Tronic.lk	0.18	-	0.18
JST Connectors	Tronic.lk	0.12	1	0.12
Terminal Blocks	Tronic.lk	0.06	3	0.18
Total				26.3442

Table 1: Bill of Materials and Supply Chain

3 Production Procedure

3.1 Product Description

The Mini Weather Station is a user-friendly device designed to accurately measure and display real-time weather data. The weather station is equipped with high-quality sensors, including the BMP280 for temperature and atmospheric pressure, the HTU21D for humidity, and an LDR sensor for ambient light intensity. The data collected from these sensors is processed and displayed on an OLED screen in easy-to-understand units, such as Celsius for temperature and percentage for humidity and light intensity. The device is powered by a rechargeable 3.7V 1800mAh li-ion battery, and it can be charged using a 5V adapter via a micro USB cable. The weather station also features WiFi functionality, enabling seamless data transmission to the Blynk IoT platform, allowing users to monitor weather conditions remotely.

3.1.1 Key Features of the Product

1. **Real-Time Weather Data:** The mini weather station provides accurate and real-time weather data, including temperature, humidity, atmospheric pressure, and

ambient light intensity. Users can access up-to-date information to make informed decisions for various applications.

2. **User-Friendly Interface:** The OLED display presents weather data in a clear and intuitive format, making it easy for users to read and understand the current environmental conditions at a glance.

3. **Wireless Connectivity:** Integrated WiFi functionality enables seamless data transmission to the Blynk IoT platform. Users can remotely monitor weather conditions via a web dashboard or mobile application, enhancing convenience and accessibility.

4. **Rechargeable Li-ion Battery:** The weather station is powered by a rechargeable 3.7V 1800mAh li-ion battery, ensuring portability and continuous operation without the need for frequent battery replacements.

5. **Optimized Power Consumption:** The ESP32 microcontroller's low-power consumption modes are intelligently utilized to further conserve energy. During periods of inactivity, the ESP32 enters Deep Sleep mode, significantly reducing power draw. Data transmission to the cloud is scheduled at 30-minute intervals, ensuring data is sent efficiently without compromising power efficiency. Moreover, the selected BMP280 and HTU21D sensors boast low-power consumption capabilities, contributing to the overall energy-efficient design of the mini weather station. Additionally the integration of a 10kohm series resistor with the LDR sensor regulates power consumption, effectively extending the battery life of the mini weather station.

6. **Cost-Effective Enclosure Design:** The weather station's enclosure is meticulously designed using CAD software, ensuring durability and protection of internal components. It is optimized for injection molding, facilitating mass production.

7. **Versatile Weather Monitoring:** With sensors for temperature, humidity, atmospheric pressure, and ambient light intensity, the weather station caters to a broad range of applications, from home use to educational, professional settings and for home gardens. The mini weather station's versatility makes it a valuable tool for a wide range of users, empowering them with real-time weather information for their specific needs.

8. **AutoConnect WiFi Setup:** The weather station acts as an Access Point (AP), allowing users to easily configure WiFi settings via a web-based configuration page. This simplifies the initial setup process for seamless connectivity.

9. **Weather Data Logging:** The mini weather station provides continuous data logging to the Blynk IoT platform, allowing users to analyze weather trends and patterns over time, making it a valuable tool for research and analysis.

10. **Extreme Weather Condition Alert with Buzzer:** The mini weather station features a built-in buzzer that acts as an essential safety feature during extreme weather conditions. When the weather station detects severe weather events, such as storms, hurricanes, or rapidly changing weather patterns, the buzzer is programmed to emit a distinct and attention-grabbing alarm.

3.1.2 Product Specifications

1. Measurement Ranges(Operating Ranges) :
    (a) Temperature : -40°C to 85°C
    (b) Atmospheric Pressure : 300hPa to 1100 hPa
    (c) Humidity : 0% to 100% RH
    (d) Ambient Light Intensity : 0 lux to 1000 lux
2. Accuracy of Measurements :
    (a) Temperature : $\pm 0.5^\circ$C
    (b) Atmospheric Pressure : 1 hPa
    (c) Humidity : $\pm 2\%$
    (d) Ambient Light Intensity : $\pm 20\%$
3. Display :
    (a) Resolution : 128 x 64 pixels
    (b) Screen Size : 0.96 inches
    (c) Power consumption : ¡40mW
4. Power Source :
    (a) Battery : Rechargeable Li-ion (3.7V, 1800mAh)
    (b) Charging : Micro USB with 5V adapter
5. Battery Life :
    (a) With display is turned on continuously : Approximately 22 hours
    (b) With turned off display : Approximately 33 hours
6. Dimensions : 120 mm x 90 mm x 50 mm (L x W x H)
7. Weight : Approximately 250 grams (with battery)

3.2 PCB Fabrication

The design is created using Altium Designer, and manual prototyping and testing are performed before proceeding with manufacturing. Then the double-sided printed circuit board for the Mini Weather Station is outsourced to a Chinese PCB manufacturing company, JLCPCB (https://jlcpcb.com/).

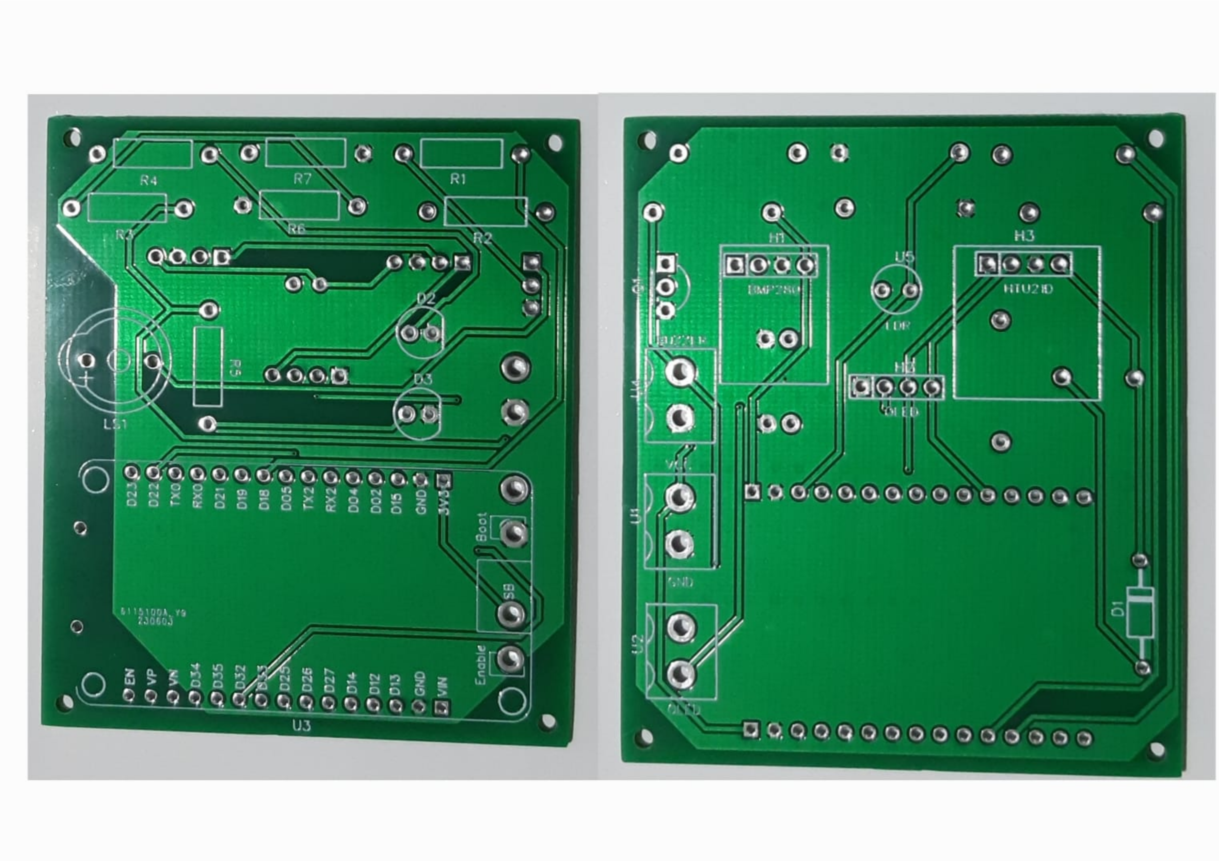


Figure 32: Unsoldered PCB

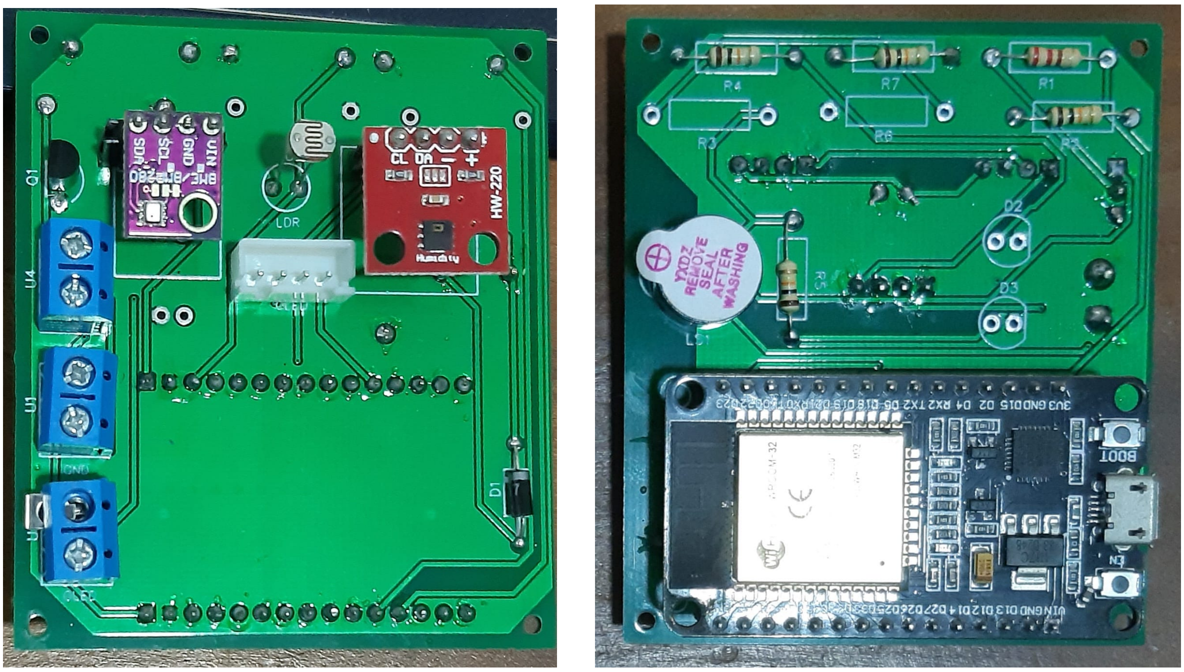


Figure 33: Soldered PCB

3.3 Enclosure Manufacturing

The entire design of the mini weather station is meticulously modeled using the CAD software Solidworks. Comprehensive analyses, including draft analysis, undercut analysis, and parting line analysis, are conducted to ensure that the design is entirely moldable using injection molding for mass production. However, for the initial few units, 3-D printing is utilized as a cost-effective alternative. Injection molding proves to be more affordable when producing a larger quantity of units.

To house the internal components, the enclosure is manufactured using plastic as the raw material. Plastic is the optimal choice among available alternatives due to its versatility, durability, and cost-effectiveness. This selection ensures that the mini weather station's housing is well-suited to protect the delicate electronics while maintaining an attractive and functional design.

Figure 34: 3D Printed Enclosure

3.4 Instructions for Assembly

The assembly process of the final product involves several steps that ensure its quality and functionality.

1. Obtain the components listed on the Bill of Materials from the trusted suppliers and verified.

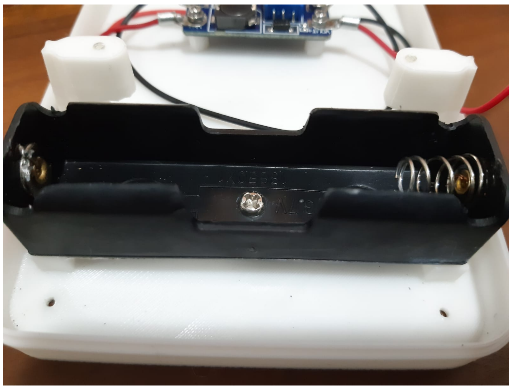
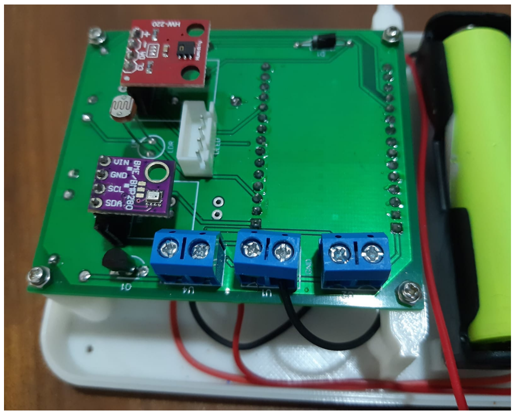
2. Select and prepare the items that have passed the testing and quality control procedures for assembly.

3. Solder the components following the provided circuit diagram and ensuring proper alignment.

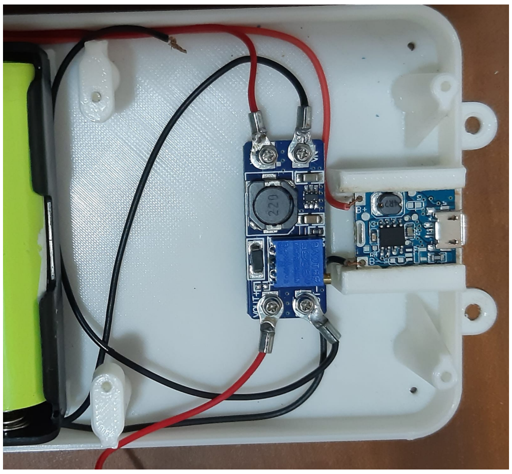
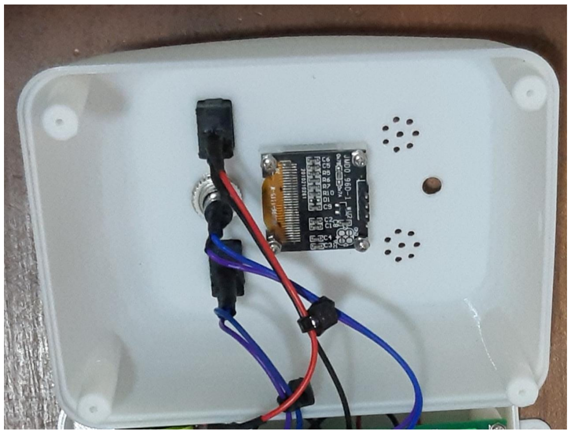
4. Carefully insert BMP280, HTU21D sensor modules and ESP32 Development board to the designated places on PCB.

5. Connect the Li-ion battery with the correct alignment: the positive (+) and negative (-) terminals with the corresponding markings on the mini weather station's power

input.

6. Place power bank module and the boost converter on the appropriate places of the bottom part of the PCB.
7. Place the PCB on the mounting holes of the bottom part of the PCB.
8. Attach switches, push button and OLED display to the top part of the enclosure.

9. Double-check all connections and ensure there are no loose wires or solder joints.

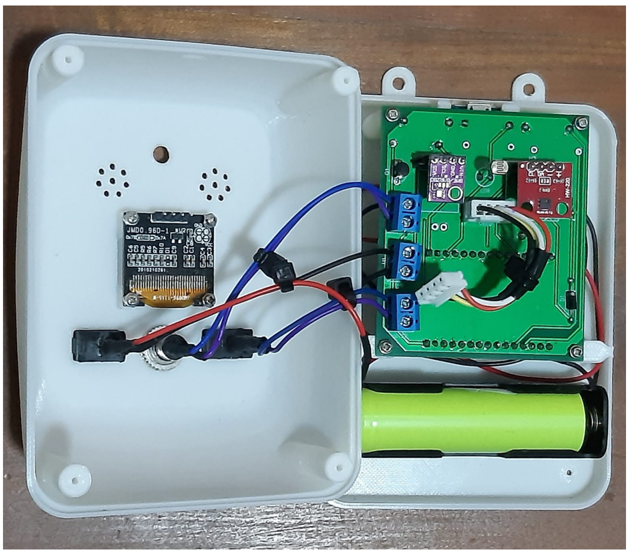


Figure 35: Assembled enclosure and PCB

3.5 Product Validation Process

The validation process for the mini weather station involves rigorous testing and verification to confirm that the weather station meets the specified performance criteria and complies with the intended requirements.

3.5.1 Functionality Testing

- **Sensor Readings Verification :** Each sensor (BMP280, HTU21D, LDR) is individually connected to the esp32 development board and tested to validate its ability to measure temperature, humidity, atmospheric pressure, and ambient light intensity accurately. The collected data is cross-checked with known reference values to verify precision and consistency.

- **Buzzer Activation Testing :**

    1. **Temperature :**
        - **High-Temperature Test :** Place the weather station in an environment with artificially increased temperature (e.g., using a heat source or heat gun) to simulate high-temperature conditions.
        - **Low-Temperature Test :** Place the weather station in an environment with artificially decreased temperature (e.g., using a freezer or refrigeration unit) to simulate low-temperature conditions.

       Monitor the behavior of the mini weather station during this test to ensure that the buzzer activates and emits the alarm sound. Verify that the station's sensors continue to provide accurate readings on the OLED display.

    2. **Humidity :** To test the buzzer activation in response to humidity changes, place the weather station in a sealed container with added moisture, and verify if the buzzer activates during the buildup of humidity.

    3. **Pressure :** To test the buzzer activation in response to atmospheric pressure changes, vary the altitude or use a pressure chamber and observe if the buzzer triggers the alarm sound accordingly.

- **WiFi Connectivity and Data Transmission Testing:** The weather station's connectivity is tested by periodically disconnecting and reconnecting it to the network. The test verifies that data updates are consistently and accurately displayed on the Blynk platform, confirming the successful integration of WiFi functionality for cloud synchronization and data transmission.

3.5.2 Power Consumption Analysis :

Using a multimeter, the power consumption is measured during various operational phases, including sensor activation, display usage, buzzer activation, WiFi transmission, and data processing. The data collected allows for the calculation of the overall power consumption during regular operation. By comparing the power consumption with the battery capacity, the estimated battery life of the weather station can be determined.

This process enables the identification of areas for power optimization, such as adjusting sleep modes and sensor polling frequency, to enhance the device's power management and ensure prolonged operation with optimized battery usage.

3.5.3 Long-Term Stability Testing :

The mini weather station can be undergone prolonged testing to assess its stability and reliability over extended periods. It helps identify any potential issues or variations that may occur over time.

3.5.4 Environmental Durability Testing :

The weather station can be subjected to environmental tests, including temperature variations, humidity exposure, and mechanical stress, to ensure its durability and resilience under varying conditions.

4 References

- BMP280 Datasheet -
  https://www.bosch-sensortec.com/products/environmental-sensors/pressure-sensors/bmp280/
- HTU21D Datasheet -
  https://cdn-shop.adafruit.com/datasheets/1899_HTU21D.pdf
- ESP32 DevKit V1 Datasheet -
  https://testzdoc.zerynth.com/reference/boards/doit_esp32/docs/
- ESP32 Datasheet -
  https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
- Sensor Comparison -
  https://www.mdpi.com/2079-9292/11/15/2448

5 Appendix

Mini Weather Station Code

```
#include <stdio.h>

#include <Wire.h>
#include <SPI.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP280.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <BlynkSimpleEsp32.h>
#include "Adafruit_HTU21DF.h"
#include <WiFiManager.h>

#define BLYNK_TEMPLATE_ID "TMPL6JvKokjjm"
#define BLYNK_TEMPLATE_NAME "MINI WEATHER STATION"
#define BLYNK_AUTH_TOKEN "iLkh0wUUdjs4Z4Hd0Q8avTICnW6YLA9s"

#define BLYNK_PRINT Serial

#define SEALEVELPRESSURE_HPA (1013.25)
#define SCREEN_WIDTH 128  // OLED display width, in pixels
#define SCREEN_HEIGHT 64  // OLED display height, in pixels
#define OLED_RESET    -1  // Reset pin # (or -1 if sharing reset pin)
#define LIGHT_SENSOR_PIN 15 //LDR sensor pin
#define BUZZER 18
#define LED_WIFI 34
#define BUTTON 14

//for buzzer alarm
bool break_happened = false;

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
Adafruit_BMP280 bmp;
Adafruit_HTU21DF htu = Adafruit_HTU21DF();

double calcLightIntensity(int AnVal, int Resistor){
  float lux = (AnVal / 4095.0) * 100.0;
  //double Vout= AnVal*0.0048828125;
  //int lux=((2500/Vout-500)/Resistor);
  return lux;
}

char auth[] = BLYNK_AUTH_TOKEN;
char ssid[40] ="";  // type your wifi name
char pass[40] ="";  // type your wifi password
BlynkTimer timer;
```

```
void sendSensor()
{
    digitalWrite(LED_WIFI,HIGH);
    float temp = bmp.readTemperature();
    float pressure = bmp.readPressure()/100;
    float rel_hum = htu.readHumidity();
    //float alt = bmp.readAltitude(1013.25);
    int analogVal = analogRead(LIGHT_SENSOR_PIN); //read values from D15
    int Resistance= 10; //I am using a 10k ohms resistor
    int luxval=calcLightIntensity(analogVal, Resistance );


  // You can send any value at any time.
  // Please don't send more that 10 values per second.
    if (Blynk.connected()){
      Blynk.virtualWrite(V0, temp);
      Blynk.virtualWrite(V1, pressure);
      Blynk.virtualWrite(V2, rel_hum);
      Blynk.virtualWrite(V3, luxval);
    }


}

void ring_buzzer(){

  bool button_pressed=digitalRead(BUTTON);
  //Serial.println(digitalRead(BUTTON),break_happened);
  //Ring the buzzer
  while (break_happened == false && button_pressed == HIGH) {
    bool button_pressed=digitalRead(BUTTON);
    if (button_pressed == LOW) {
      //Serial.println(digitalRead(BUTTON));
      delay(200);
      break_happened = true;
      digitalWrite(BUZZER,LOW);
      break;
    }
    digitalWrite(BUZZER,HIGH);
    delay(500);
  }
}

void setup() {
  Serial.begin(9600);
```

```
WiFiManager wm;
wm.resetSettings();
bool res = wm.autoConnect("AutoConnectAP","password");
if(!res) {
  Serial.println("Failed to connect");
  // ESP.restart();
}
else {
  //if you get here you have connected to the WiFi
  Serial.println("connected...yeey :)");
  String ssidString = wm.getWiFiSSID();
  String passString = wm.getWiFiPass();
  ssidString.toCharArray(ssid, sizeof(ssid));
  passString.toCharArray(pass, sizeof(pass));
}
pinMode(BUZZER, OUTPUT);
pinMode(LIGHT_SENSOR_PIN, INPUT);
pinMode(LED_WIFI,OUTPUT);
pinMode(BUTTON,INPUT);

Blynk.begin(auth, ssid, pass);
/***
while (Blynk.connect() == false) {
  if(((millis()/1000) - timeout) > 10){   // issue msg if not connected to Blynk in
    break;
  }
}
***/

Wire.begin();
// initialize the SSD1306 OLED display with I2C address = 0x3D
display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
// clear the display buffer.
display.clearDisplay();
display.setTextSize(1);   // text size = 1
display.setTextColor(WHITE, BLACK);  // set text color to white and black backgroun
display.setTextWrap(false);           // disable text wrap
//display.setCursor(5, 0);              // move cursor to position (0, 4) pixel
//display.print("Mini Weather Station");
/***
display.setCursor(123, 4);            // move cursor to position (123, 4) pixel
display.println("S");  display.setCursor(123, display.getCursorY());
display.println("T");  display.setCursor(123, display.getCursorY());
display.println("A");  display.setCursor(123, display.getCursorY());
display.println("T");  display.setCursor(123, display.getCursorY());
display.println("I");  display.setCursor(123, display.getCursorY());
display.println("O");  display.setCursor(123, display.getCursorY());
display.println("N");
```

```
  display.display();        // update the display
  ***/

  unsigned status;
  status = bmp.begin(0x76);
  if (!status) {
     // connection error or device address wrong!
    Serial.println(F("Could not find a valid BMP280 sensor, check wiring!"));
    display.setCursor(34, 23);
    display.print("Connection");
    display.setCursor(49, 33);
    display.print("Error");
    display.display();        // update the display
    while (1) // stay here
      delay(1000);
  }
  if (!htu.begin()) {
    Serial.println("Couldn't find sensor!");
    while (1);
  }

  display.display();
  // update the display

  /* Default settings from datasheet. */
  bmp.setSampling(Adafruit_BMP280::MODE_NORMAL,     /* Operating Mode. */
                  Adafruit_BMP280::SAMPLING_X2,     /* Temp. oversampling */
                  Adafruit_BMP280::SAMPLING_X16,    /* Pressure oversampling */
                  Adafruit_BMP280::FILTER_X16,      /* Filtering. */
                  Adafruit_BMP280::STANDBY_MS_500); /* Standby time. */

  timer.setInterval(100L, sendSensor);

}

void loop() {
  timer.setInterval(100L, sendSensor);

  float temp_bmp280 = bmp.readTemperature();    // get temperature in degree Celsius
  float pres = bmp.readPressure()/100;       // get pressure in Pa
  //float alti = bmp.readAltitude(SEALEVELPRESSURE_HPA);       // get altitude in met

  float temp_htu21d = htu.readTemperature();
  float rel_hum = htu.readHumidity();

  int analogVal = analogRead(LIGHT_SENSOR_PIN); //read values from D15
  int Resistance= 10; //I am using a 10k ohms resistor
  int luxval=calcLightIntensity(analogVal, Resistance );
```

```
 delay(1000);  // wait a second
// print data on the LCD

  // print temperature
 //display.clearDisplay();
 Serial.print("Temperature1 = ");
 Serial.print(temp_bmp280);
 Serial.println("*C");
 Serial.print("Temperature2 = ");
 Serial.print(temp_htu21d);
 Serial.println("*C");

 display.setCursor(0, 16);
 display.print("Temperature:");
 display.setCursor(75, 16);
 if (temp_bmp280 < 0)
   display.printf("-%02u.%02u C", (int)abs(temp_bmp280) % 100, (int)(abs(temp_bmp280
 else
   display.printf(" %02u.%02u C", (int)temp_bmp280 % 100, (int)(temp_bmp280 * 100) %
// print degree symbols ( ° )
 display.drawRect(111, 16, 3, 3, WHITE);
 //delay(2000);


//print pressure
 //display.clearDisplay();
 Serial.print("Pressure = ");
 Serial.print(pres);
 Serial.println("hPa");
 display.setCursor(0, 28);
 display.print("Pressure:");
 display.setCursor(60, 28);
 display.printf("%02u hPa", (int)(pres));
 //delay(2000);


 //print humidity
 //display.clearDisplay();
 Serial.print("Humidity = ");
 Serial.print(rel_hum);
 Serial.println("%");
 display.setCursor(0, 40);
 display.print("Humidity:");
 display.setCursor(60, 40);
 display.printf("%02u %s", (int)(rel_hum), "%");
 //delay(2000);
```

```
   //print Altitude
  //Serial.print("Approx. Altitude = ");
  //Serial.print(alti);
  //Serial.println("m");
  //display.setCursor(38, 56);
  //display.printf("%02u m", (int)(alti));

  //print lux
  //display.clearDisplay();
  Serial.print("Light Intensity:");
  Serial.print(luxval);
  Serial.println(" lux");
  display.setCursor(0, 52);
  display.print("Light Intensity:");
  display.setCursor(100, 52);
  display.printf("%02u %s", (int)(luxval), "%");
  //delay(2000);

  //extreme conditions
  if (temp_bmp280>35 || temp_bmp280<20 || rel_hum > 80 || rel_hum < 30 || pres > 1100
    Serial.println(digitalRead(BUTTON));
    ring_buzzer();
  }
  else{
    bool break_happened = false;
  }


 // update the display
  display.display();
  delay(60000);  // wait a second
  Serial.println();

  Blynk.run();
  timer.run();
}
```