Department of Electronic & Telecommunication Engineering

University of Moratuwa

EN2160 - Electronic Design Realization

Digital Temperature and Humidity Meter

(Project Report)

200087A Chandrasiri Y.U.K.K.

Abstract

This report presents the comprehensive analysis of a Digital temperature and humidity measuring device project undertaken for the EN2160 – Electronic Design Realization course. The report encompasses a detailed overview of the device's functionality, specifications, the entire design process, and user instructions. Through this project, a remarkable digital meter has emerged, bringing a new level of convenience and functionality to our daily lives. With its integration of advanced technology, this device surpasses traditional meters by seamlessly displaying time, date, temperature, and humidity information. In addition to its accurate timekeeping capabilities, it serves as a versatile instrument for monitoring environmental conditions. This digital device allows users to effortlessly measure and view temperature and humidity levels in real-time. Whether you're seeking to create the perfect indoor climate or simply stay informed about the surrounding conditions, this digital meter provides invaluable insights at your fingertips. The report offers valuable insights into the design and implementation of the device, shedding light on its contribution towards efficiency and compatibility in life.

Introduction

Temperature & Humidity Monitoring System is designed to provide accurate and real-time measurements, this innovative device offers an all-in-one solution for measuring temperature, humidity, and real-world time and date. With its user-friendly interface and customizable options, it empowers users to effortlessly track and analyze environmental conditions, ensuring optimal comfort and productivity in any setting.

Features:

Our Temperature & Humidity Monitoring System boasts a range of advanced features that set it apart from conventional monitoring devices:

Triple Measurement Display: The device allows users to simultaneously measure and display temperature, humidity, and real-world time and date on a clear and vibrant screen.

Unit of Measurement Customization: Users have the flexibility to switch between Celsius (C) and Fahrenheit (F) units, catering to different regional and individual preferences.

Temperature Focus: The system primarily emphasizes temperature, offering a detailed breakdown of maximum recorded temperature within the past hour, 24 hours, and 30 days. This valuable data enables users to identify trends and patterns in environmental conditions.

Verbal Response Feature: The device goes beyond mere numerical readings. It provides intuitive verbal responses to the measured values, such as "comfortable," "too hot," "too cold," etc. This insightful feedback helps users quickly assess their surroundings and make informed decisions to enhance comfort.

Sleek and Intuitive Design: With a modern and compact design, the Temperature & Humidity Monitoring System seamlessly fits into any environment. Its user-friendly interface ensures easy navigation, making it accessible to users of all ages.

Battery Efficiency: Equipped with an energy-efficient design, the device guarantees prolonged battery life, ensuring reliable performance in extended usage.

Applications:

The Temperature & Humidity Monitoring System finds diverse applications across various sectors:

Home Comfort: Create the perfect indoor environment for your family by monitoring and adjusting temperature and humidity levels to suit personal preferences.

Office Spaces: Improve workplace productivity and well-being by maintaining optimal temperature and humidity conditions.

Industrial Settings: Ensure worker safety and protect equipment by closely monitoring environmental conditions in industrial facilities.

Healthcare Facilities: Support patient comfort and recovery by regulating environmental conditions in hospitals and clinics.

In a world where environmental conditions significantly impact well-being and productivity, our Temperature & Humidity Monitoring System is a game-changer. Combining precision, versatility, and user-friendliness, it empowers users to take control of their surroundings, ensuring utmost comfort and efficiency in any setting.

Product Functionality

The system incorporates two displays to provide real-time output to users. Its primary feature includes setting a safe temperature value, triggering an alarm, and activating an LED when the real-time temperature exceeds the predefined safe threshold. Furthermore, the system allows users to easily deactivate the alarm and LED as per their convenience.

Product Specifications:

DHT11 Sensor: The system is equipped with a reliable and precise DHT11 sensor to measure temperature and humidity levels effectively.

Dual Displays: Two displays are integrated into the system to showcase real-time temperature and humidity readings, ensuring easy accessibility and visibility for users.

Safe Temperature Setting: Users can configure a specific safe temperature threshold according to their requirements. This enables the system to monitor the environment against potential deviations.

Alarm Trigger: When the real-time temperature surpasses the set safe temperature, an audible alarm is automatically triggered to alert users of the unfavorable condition.

LED Indicator: In addition to the alarm, an LED indicator is activated simultaneously with the alarm to provide a visual alert about the temperature breach.

User-Controlled Alarm and LED: Users have the flexibility to turn off the alarm and LED whenever desired, allowing them to manage alerts based on their preferences and situations.

The Environmental Monitoring System operates as follows:

- The DHT11 sensor continuously captures temperature and humidity data from the environment.
- The system processes the sensor data and displays the real-time readings on the two integrated displays for easy monitoring.
- Users can set their preferred safe temperature value using intuitive controls.
- Once the safe temperature threshold is defined, the system actively monitors the real-time temperature.
- If the real-time temperature exceeds the predetermined safe value, the alarm is triggered, emitting an audible alert.
- Simultaneously, the LED indicator lights up, providing a visual indicator of the temperature breach.
- Users can choose to disable the alarm and LED, allowing them to silence the alerts as per their convenience.

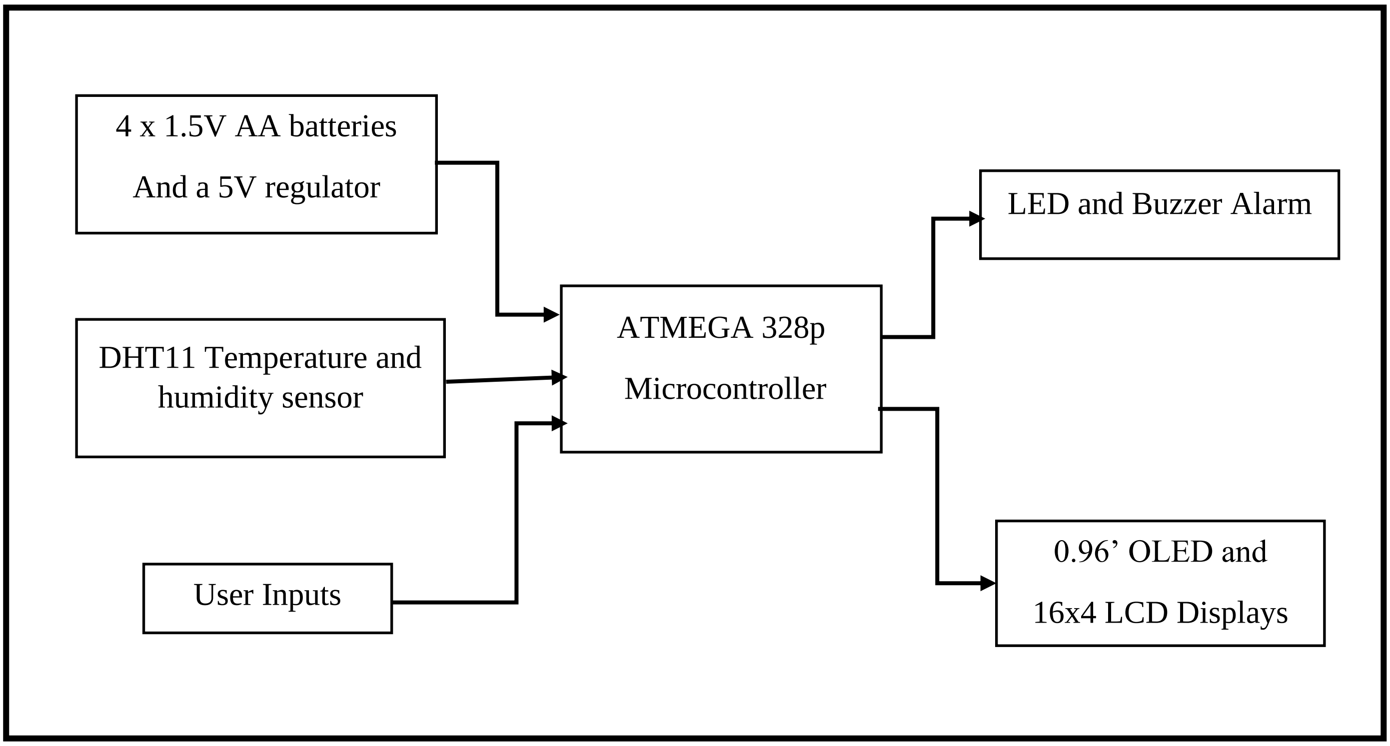


Figure: Block Diagram

Figure: User Interface

Figure: 0.96” OLED

Figure: 16x4 LCD Display

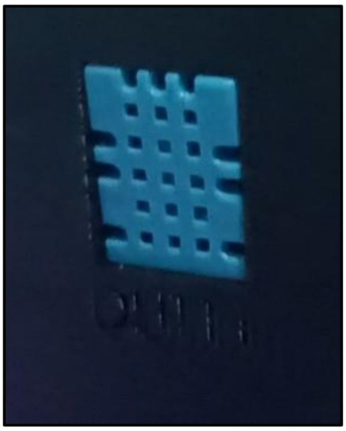


Figure: DHT11

Figure: 4x 1.5V Batteries

Figure: Default Safe Temperature Settings

Figure: LED and Alarm

Product Specifications

- Temperature range: 5 ^{0}C – 85 ^{0}C
- Supply voltage: 5V DC
- Operating voltage: 5V DC
- 0.96" OLED display.
- 16*4 LCD display
- RTC real-time clock

Circuit Design

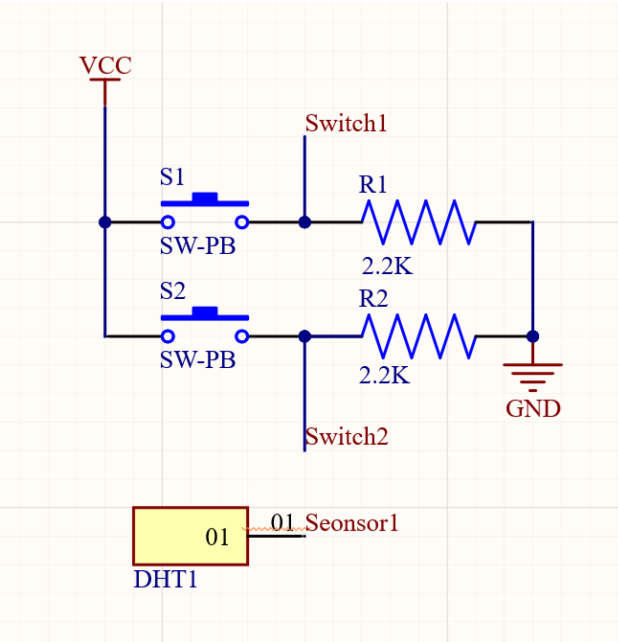


Figure: Inputs

The system employs two push buttons as user inputs to navigate through the menu, while one input is dedicated to interfacing with the DHT11 sensor.

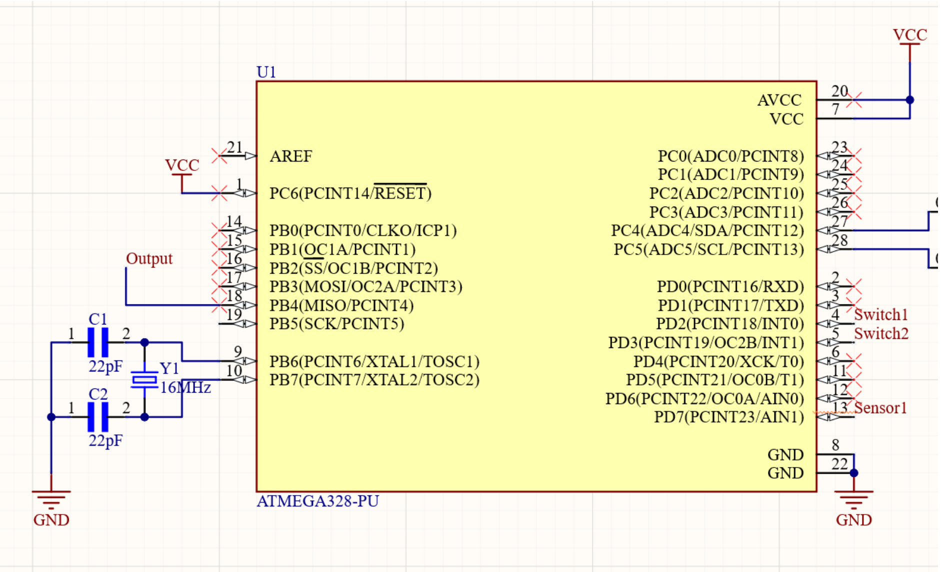


Figure: Microcontroller

The microcontroller of choice is the Atmega 328p, complemented by a 16MHz oscillator for precise timing and clock synchronization.

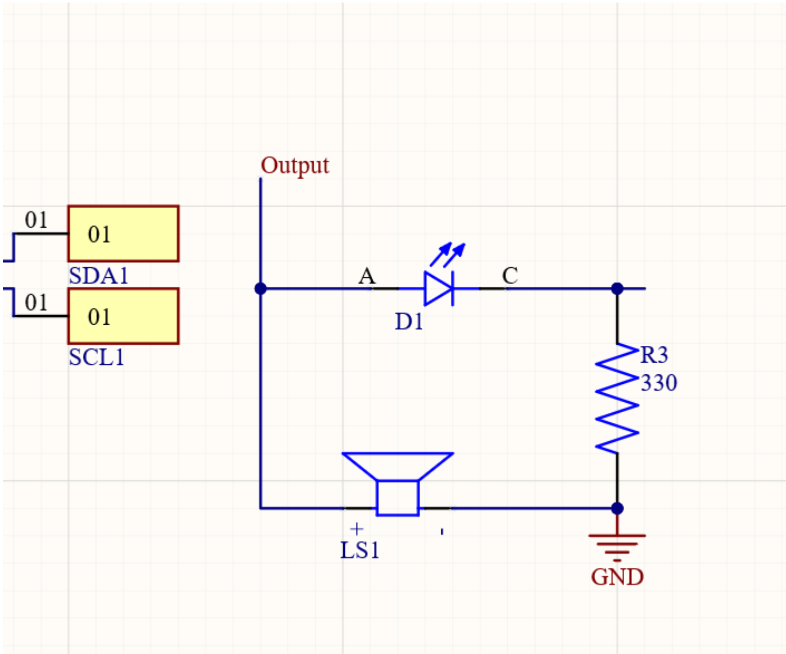


Figure: Output

The SDA and SCL pins are utilized as I2C pins to establish connections with displays, while an LED and a buzzer serve as the primary output components.

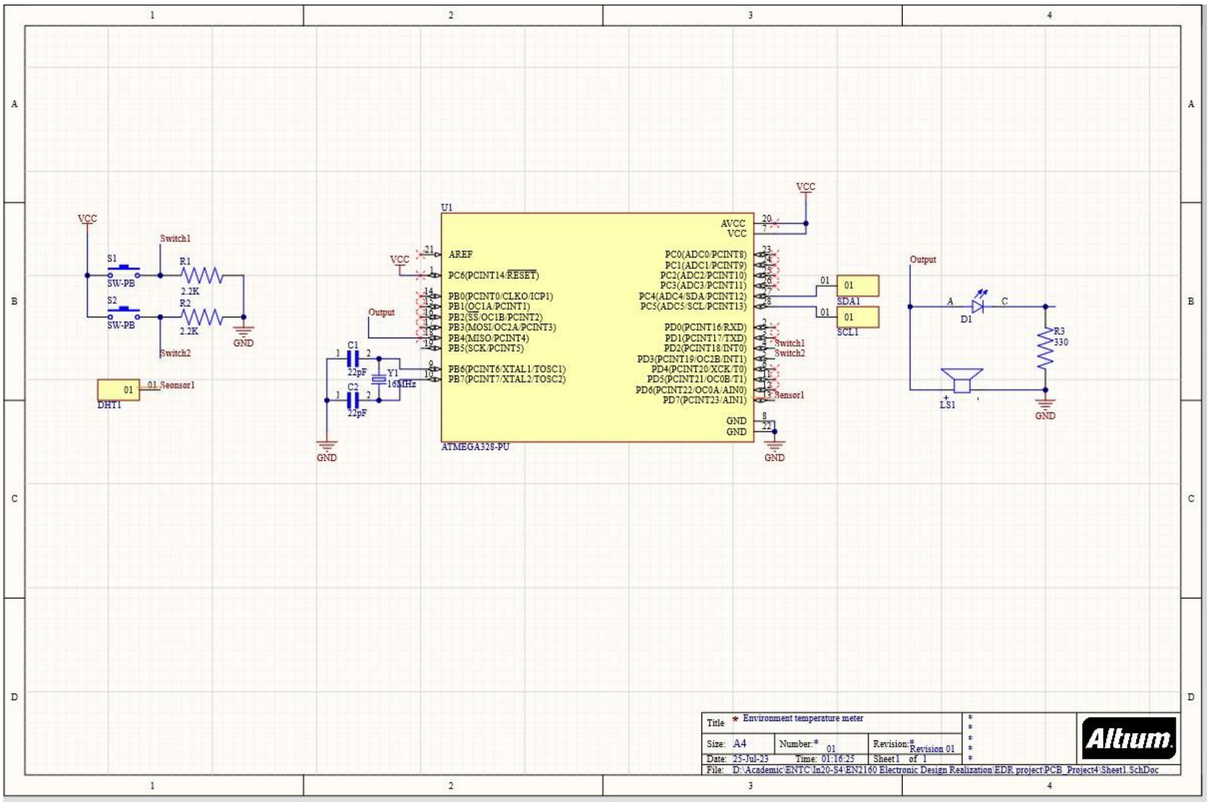


Figure: Altium Schematic

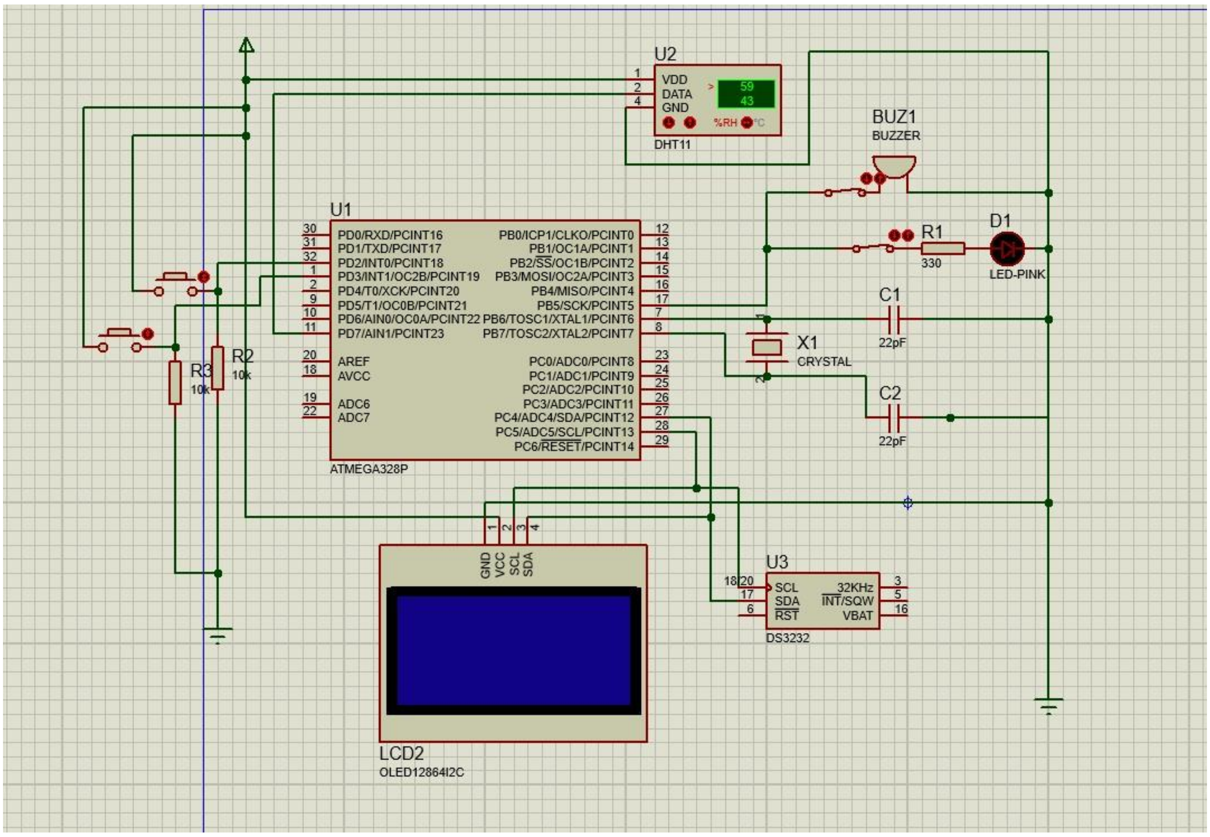


Figure: Proteus Simulation

PCB Design

The PCB and Schematic design for the device were accomplished using the Altium Designer software, following the specific requirements of the manufacturer, JLCPCB. The circuit layout was designed as a double-layer PCB, which is a common and cost-effective choice for many electronic devices. To ensure proper grounding and minimize interference, both PCB layers were grounded by creating polygon pours.

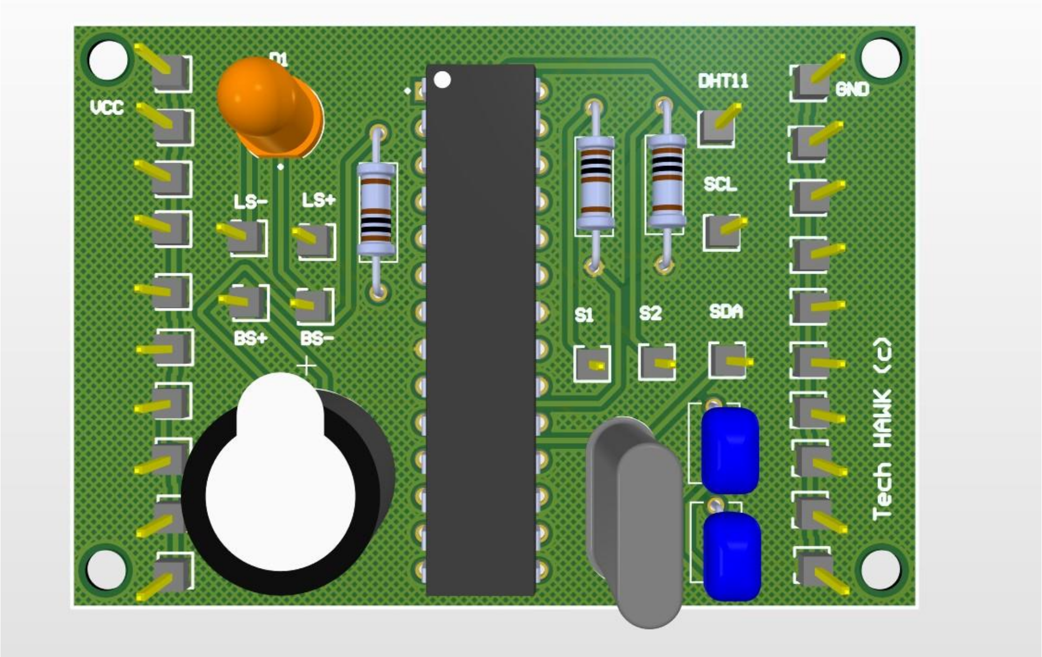


Figure: 3D view of the PCB

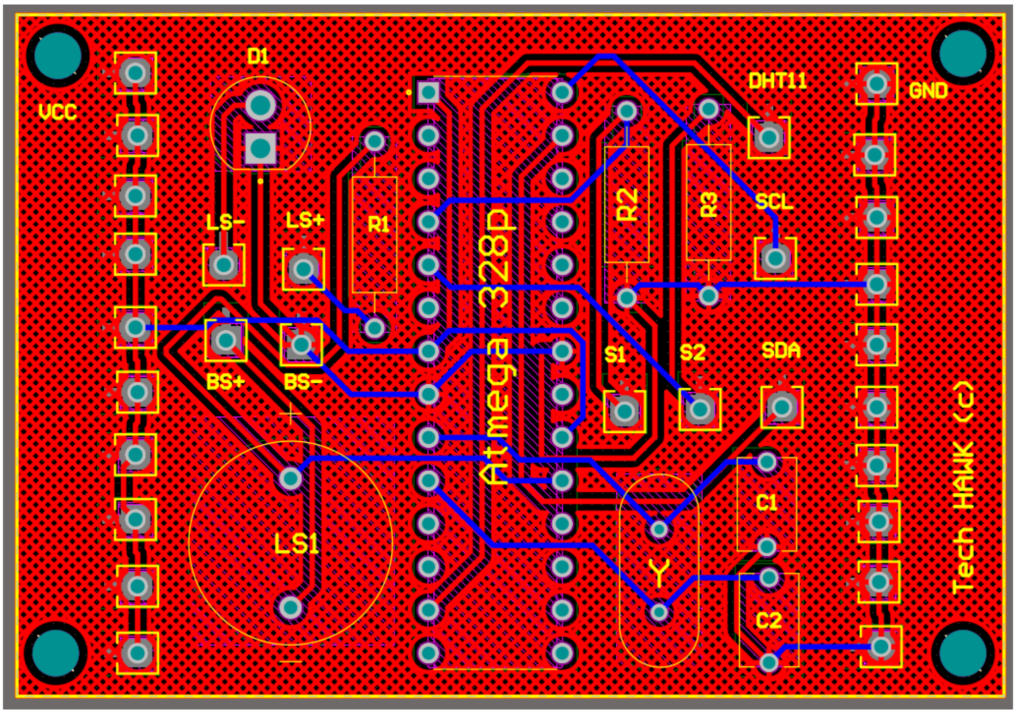


Figure: PCB

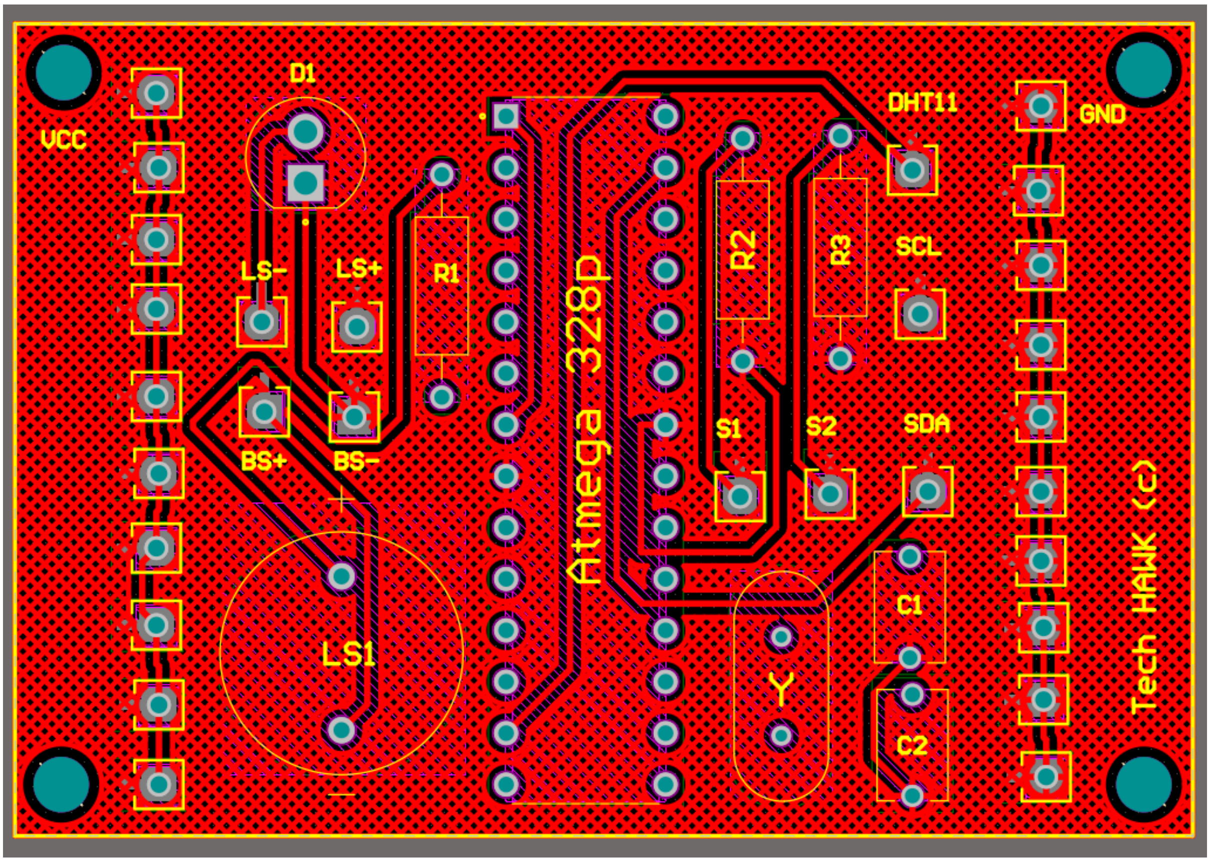


Figure: Top Layer

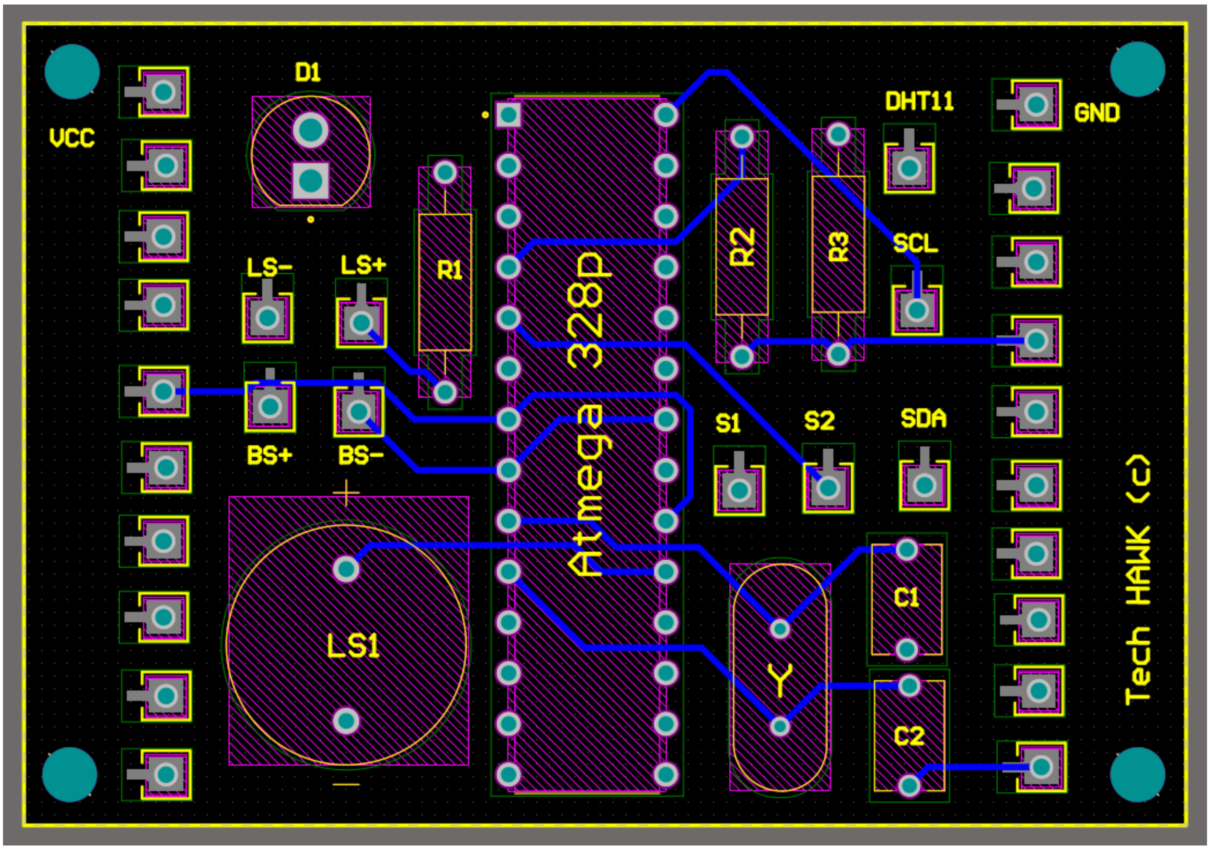


Figure: Bottom Layer

Source Documents

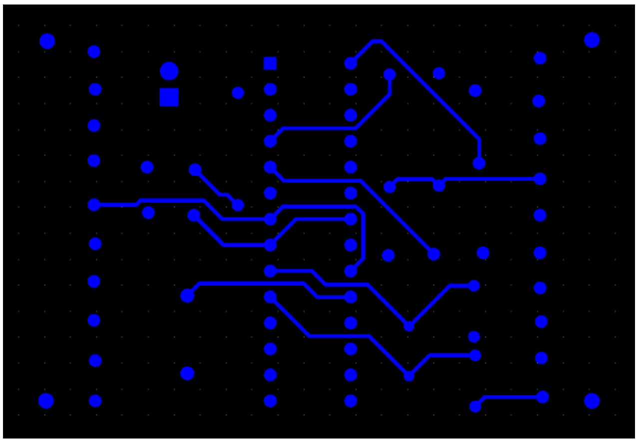


Figure: PCB.GBL

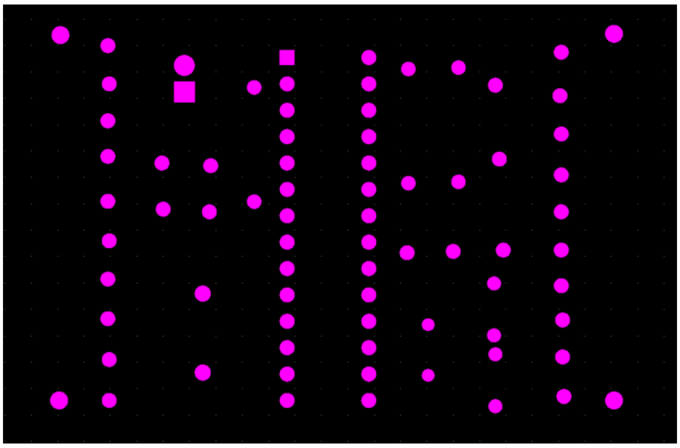


Figure: PCB.GBS

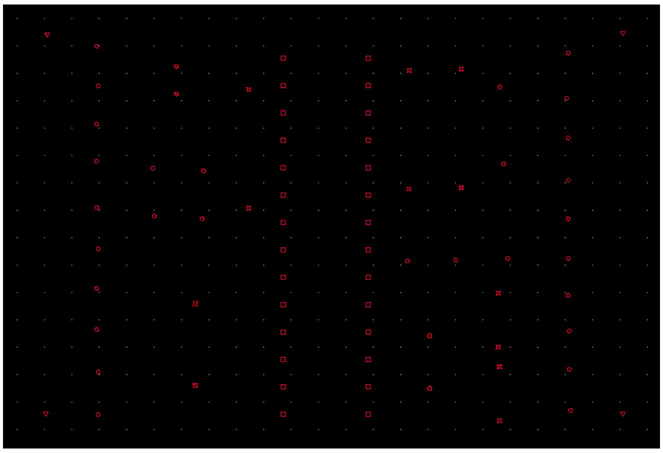


Figure: PCB.GD1

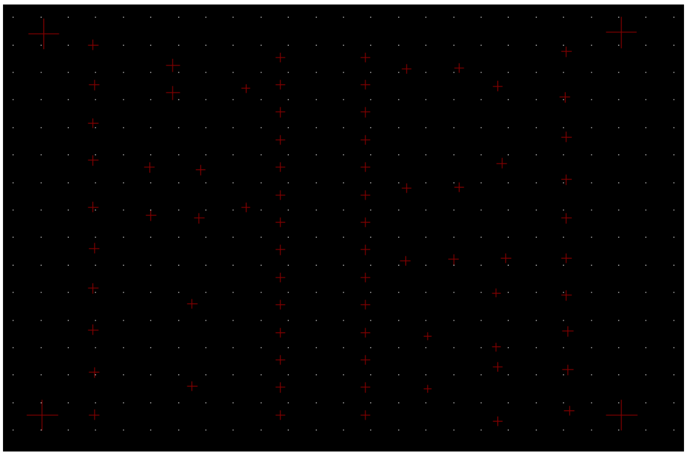


Figure: PCB.GG1

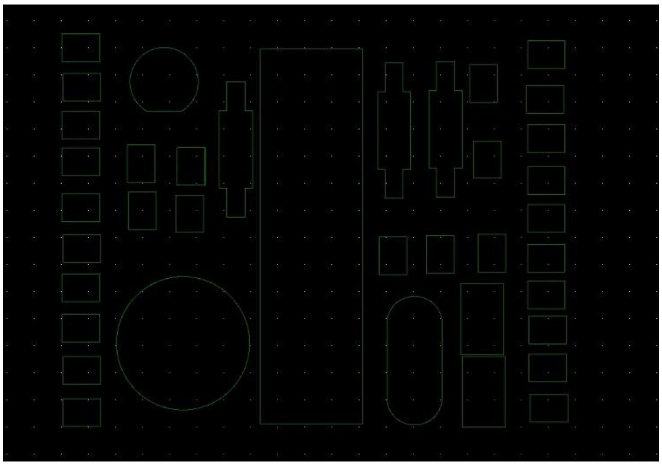


Figure: PCB.GM15

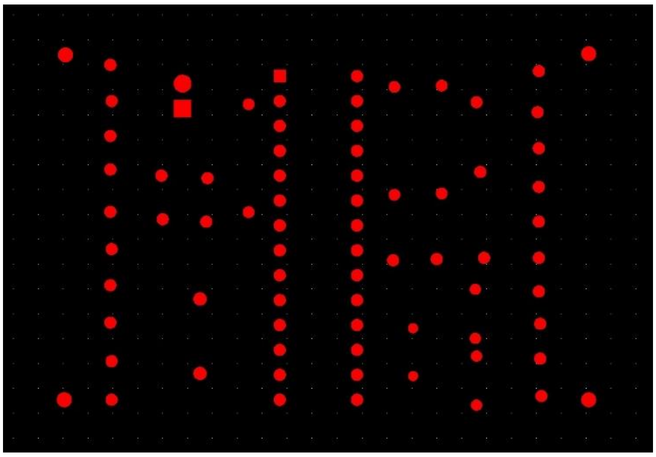


Figure: PCB.GPB

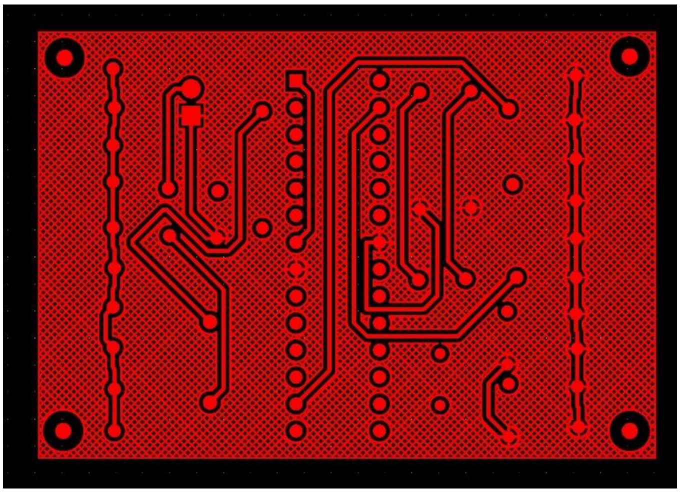


Figure: PCB.GTL

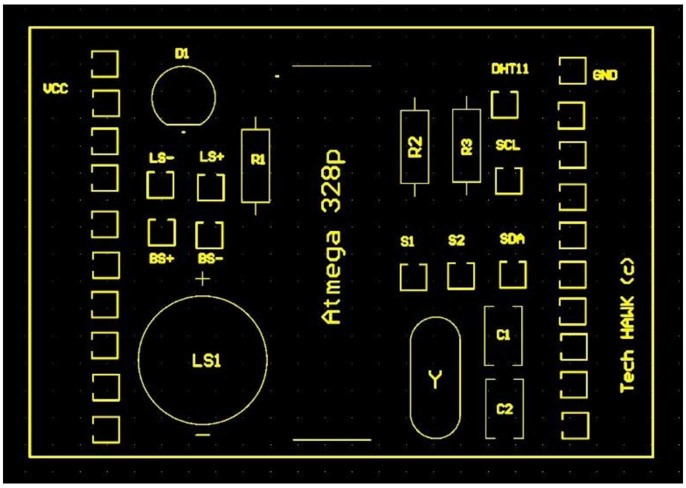


Figure: PCB.GTO

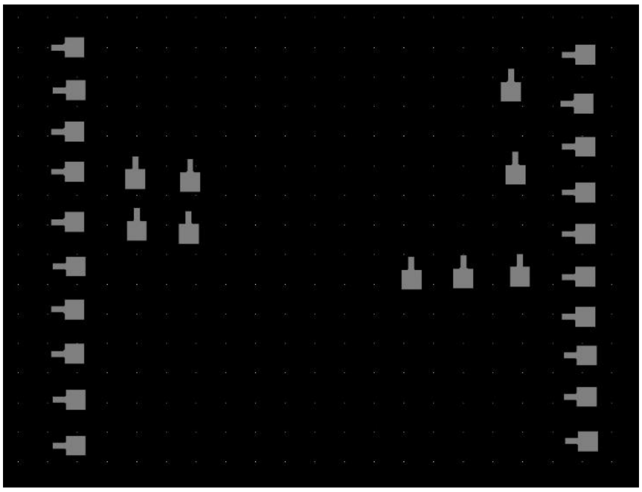


Figure: PCB.GTP

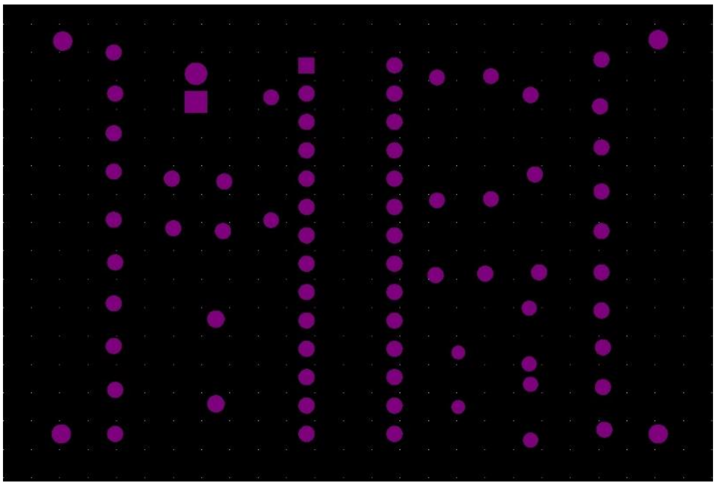


Figure: PCB.GTS

Enclosure Design

The Enclosure of the device was designed using the SOLIDWORKS software.

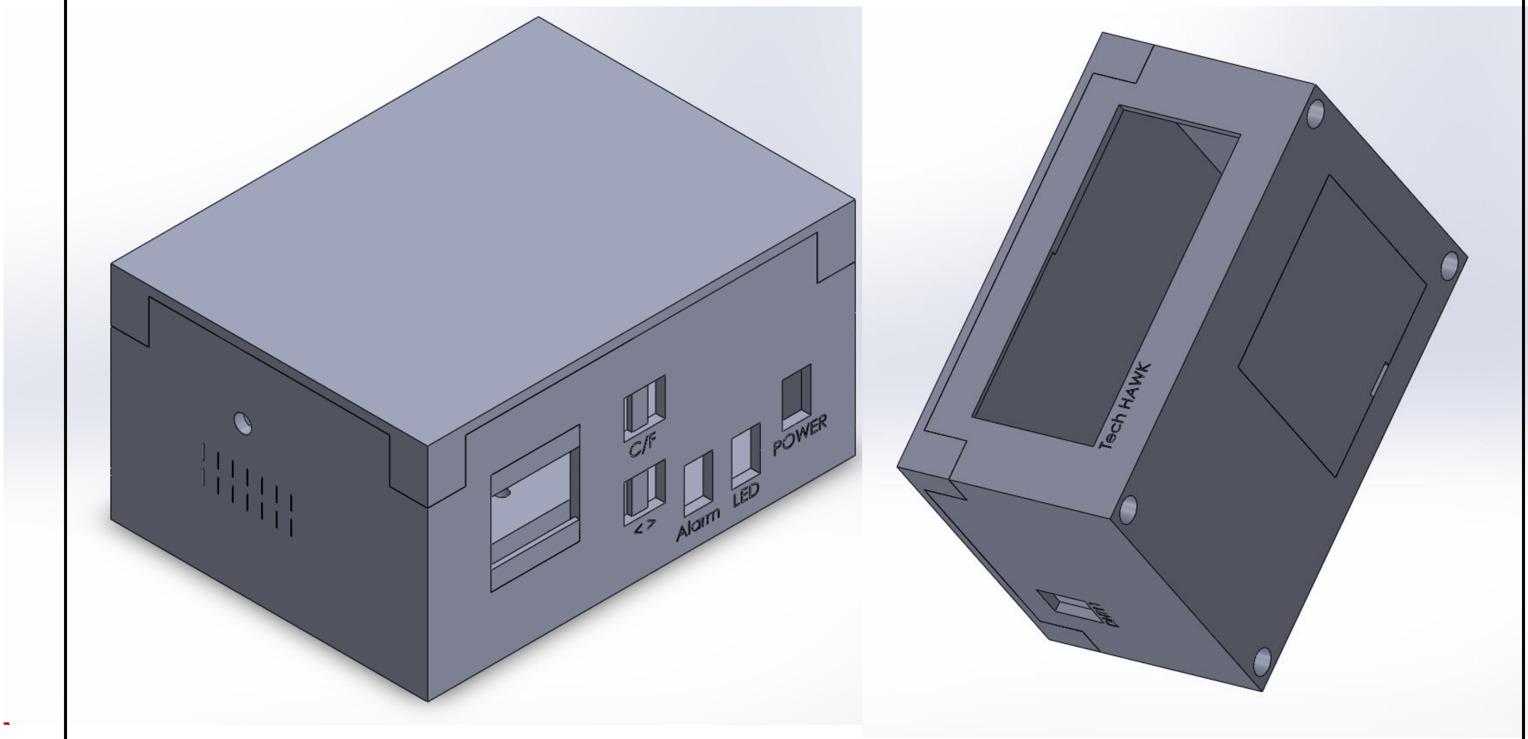
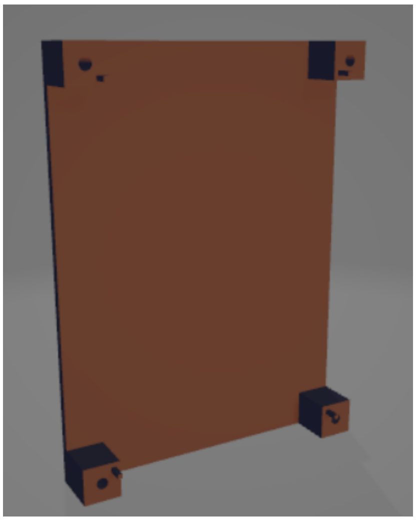
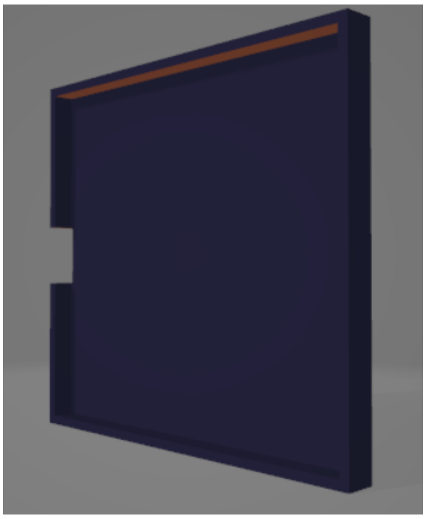


Figure: Enclosure for 3D printing

.STL files





Figures: base.stl lid.stl batteryCover.stl

Bill Of Material

Component	Quantity	Supplier	Price (LKR)
Atmega 328p	1	Mouser Electronics	4000
DHT11	1	Mouser Electronics	600
5V Buck converter	1	Mouser Electronics	790
RTC	1	Mouser Electronics	820
1.5V Sony Battery	4	Mouser Electronics	1600
4 AA Battery casing	1	Mouser Electronics	180
22pF Capacitor	2	Mouser Electronics	20
1K Resistor	2	Mouser Electronics	20
330 Resistor	2	Mouser Electronics	20
OLED Display	1	Mouser Electronics	1980
16x4 LCD Display	1	Mouser Electronics	1450
I2c converter	1	Mouser Electronics	650
On/Off switches	3	Mouser Electronics	120
Push Buttons	2	Mouser Electronics	80
Red LED	1	Mouser Electronics	70
5V Buzzer	1	Mouser Electronics	300

Manufacturing Details

PCB Manufacturing

PCB Manufacturer: Jia Li Chuang (Hong Kong) Co., Limited (JLC PCB)

PCB Specifications

- Base Material: FR-4
- Layer: 2
- Dimensions: 58.19mm * 44.33mm
- PCB Color: Green
- Silkscreen Color: White
- Surface Finish: HASL(with lead)
- Copper weight: 1 oz

For PCB Manufacturing, the total cost: 3000LKR (with TAX and shipping cost)

Enclosure Manufacturing

Prototype has been 3D printed locally.

Enclosure Manufacturing total cost: 7200LKR.

Code

Main Code <code.ino>

```
#include <Wire.h>

#include <Adafruit_GFX.h>

#include <Adafruit_SSD1306.h>

#include <RTClib.h>

#include <dht.h>


char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday"};

int led=13;

int buz=12;

char tempunit='C';

int safetemp=40;

float hourtemp=0;

float todaytemp=0;

int page=0;

float z;

float x;

int th; int tm;int dy;int dm;int dd;

DateTime now;


RTC_DS3231 rtc;

dht tnh;


#define OLED_RESET 4

Adafruit_SSD1306 display(OLED_RESET);


void setup() {
```

```
  pinMode(led, OUTPUT);pinMode(buz, OUTPUT);
  pinMode(2,INPUT); pinMode(3,INPUT);

  attachInterrupt(0,button1,RISING);
  attachInterrupt(1,button2,RISING);

  Serial.begin(9600);
  Wire.begin();
  rtc.begin();
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.display();
  //delay(1000);
  display.clearDisplay();
  rtc.adjust(DateTime(2023, 4, 19, 12, 45, 30));
}

void loop() {
  now = rtc.now();
  int tval=tnh.read11(7);
  display.clearDisplay();
  display.setTextColor(WHITE);
  x=tnh.temperature;

  if (int(x)> todaytemp) todaytemp=x;
  if (now.hour()==0 && now.minute()==0) todaytemp=0;

  if (int(x)> hourtemp) hourtemp=x;
  if (now.minute()==0) hourtemp=0;

  if (page==0) page0();
```

```
  else if (page==1) page1();


  else if (page==2) page2();


  else if (page==3) page3();
  display.display();
  delay(1000);
}

void button1() {
  if (page>=3) page=0;
  else page++;
}

void button2() {
  if (page==0){
  if (tempunit=='C') tempunit='F';
  else tempunit = 'C';
  }
  else if (page==1){
    if (tempunit=='C') tempunit='F';
    else tempunit = 'C';
  }
  else if (page==2){
    safetemp+=1;
    if (safetemp>60) safetemp=10;
//    th = now.hour() + 1;
//    if (th > 23) {
//        th = 0;
```

```
//    }
//    rtc.adjust(DateTime(now.year(), now.month(), now.day(), th, now.minute(), 15));
  }
}

float ctof(float x){
  return (x*9/5+32);
}

void page0(){
  //Time
    display.setTextSize(2);
    display.setCursor(100,0);
    if(now.hour() < 10) display.print("0");
    display.print(now.hour(), DEC);
    display.setCursor(100,16);
    if(now.minute() < 10) display.print("0");
    display.print(now.minute(), DEC);
  //  if(now.second() < 10) display.print("0");
  //  display.println(now.second(), DEC);

    //Temp
    display.setTextSize(2);
    display.setCursor(0,0);
    if (tempunit=='F'){
      display.print(int(ctof(x)));
      display.print(" F");  }
    else {
      display.print(int(x));
      display.print(" C");
```

```
  }
  if (x>safetemp){
    digitalWrite(13,HIGH);
  }else{
    digitalWrite(13,LOW);
  }

  //Humidity
  z=tnh.humidity;
  display.setTextSize(1);
  display.setCursor(0,16);
  display.print("H ");
  display.print(int(z));
  display.print(" %");



  //Verbal Output
  display.setTextSize(1);
  display.setCursor(16,24);
  if (x>30) display.print(" Hot Day");
  else if (x<20) display.print(" Cool Day");
  else if (x>27 && z>60) display.print(" Swaety Day");
  else display.print(" Comfortable");

//  display.setTextSize(1);
//  display.print(now.day(), DEC);display.print("/");
//  display.print(now.month(), DEC);display.print("/");
//  display.println(now.year(), DEC);
}
```

```
void page1(){
  display.setTextSize(1);
  display.setCursor(0,0);
  if (tempunit=='F'){
    display.print("Now: ");
    display.print(ctof(x));
    display.println(" F");
    display.print("This Hour: ");
    display.print(int(ctof(hourtemp)));
    display.println(" F");
    display.print("Today: ");
    display.print(int(ctof(todaytemp)));
    display.println(" F");  }
  else {
    display.print("Now: ");
    display.print(x);
    display.println(" C");
    display.print("This Hour: ");
    display.print(int(hourtemp));
    display.println(" C");
    display.print("Today: ");
    display.print(int(todaytemp));
    display.println(" C");
    }
  float z=tnh.humidity;
  display.setTextSize(1);
  display.setCursor(0,24);
  display.print("Humidity: ");
```

```
  display.print(z);
  display.print(" %");
  }

void page2(){
  display.setTextSize(1);
    display.setCursor(0,0);
    if (tempunit=='F'){
      display.print(int(ctof(x)));
      display.print(" F");  }
    else {
      display.print(int(x));
      display.print(" C");
    }
    display.setTextSize(1);
    display.setCursor(0,8);
    display.print("Safe Temperature");
    display.setTextSize(2);
    display.setCursor(0,16);
    if (tempunit=='F'){
      display.print(int(ctof(safetemp)));
      display.print(" F");  }
    else {
      display.print(safetemp);
      display.print(" C");
    }
}

void page3(){
  display.setTextSize(2);
```

```
  display.setCursor(0,0);
  if(now.hour() < 10) display.print("0");
  display.print(now.hour(), DEC); display.print(":");
  if(now.minute() < 10) display.print("0");
  display.print(now.minute(), DEC);display.print(":");
  if(now.second() < 10) display.print("0");
  display.println(now.second(), DEC);

  display.setTextSize(1);
  display.setCursor(0,24);
  display.print(now.day(), DEC);display.print("/");
  display.print(now.month(), DEC);display.print("/");
  display.print(now.year(), DEC);display.print(" ");
  display.print(daysOfTheWeek[now.dayOfTheWeek()]);
  }
```

Code for scanning i2c address <i2c_scan.ino>

```
// --------------------------------------
// i2c_scanner
//
// Modified from https://playground.arduino.cc/Main/I2cScanner/
// --------------------------------------

#include <Wire.h>

// Set I2C bus to use: Wire, Wire1, etc.
#define WIRE Wire

void setup() {
  WIRE.begin();
```

```
  Serial.begin(9600);
  while (!Serial)
     delay(10);
  Serial.println("\nI2C Scanner");
}


void loop() {
  byte error, address;
  int nDevices;

  Serial.println("Scanning...");

  nDevices = 0;
  for(address = 1; address < 127; address++ )
  {
    // The i2c_scanner uses the return value of
    // the Write.endTransmisstion to see if
    // a device did acknowledge to the address.
    WIRE.beginTransmission(address);
    error = WIRE.endTransmission();

    if (error == 0)
    {
      Serial.print("I2C device found at address 0x");
      if (address<16)
        Serial.print("0");
      Serial.print(address,HEX);
      Serial.println("  !");
```

```
      nDevices++;
    }
    else if (error==4)
    {
      Serial.print("Unknown error at address 0x");
      if (address<16)
        Serial.print("0");
      Serial.println(address,HEX);
    }
  }
  if (nDevices == 0)
    Serial.println("No I2C devices found\n");
  else
    Serial.println("done\n");

  delay(5000);           // wait 5 seconds for next scan
}
```