**UNIVERSITY OF MORATUWA, SRI LANKA**

**Faculty of Engineering**

**Department of Electronic and Telecommunication Engineering**

**Semester 4 (Intake 2020)**

# EN2160 - Electronic Design Realization

## Report

Marasinghe M.M.H.N.B.

200381U

# Table of Contents

Description of the product ..... 3
Specifications ..... 4
Basic specifications ..... 4
Component specifications ..... **Error! Bookmark not defined.**
Schematic of device ..... 5
PCB Design files ..... 6
Bill of materials and supply chain ..... 10
BOM ..... 10
Supply chain ..... 10
Enclosure design ..... 11
Instructions for assembly ..... 12
PCB fabrication and component soldering process ..... 12
Enclosure manufacturing process ..... 14
Final assembly ..... 14
Tests for functionality ..... 15
Images captured from a basic functionality test (not an accuracy test) ..... 16

## Description of the product

A digital kitchen scale is a small electronic device used for measuring the weight of ingredients in the kitchen. It typically consists of a flat platform on which the ingredients are placed and a digital display screen that shows the weight. Digital kitchen scales are highly accurate and can measure weights in various units, such as grams, ounces, or pounds. They are commonly used in cooking and baking, as precise measurements of ingredients are important for achieving consistent and delicious results. Digital kitchen scales are also compact and easy to store, making them a convenient addition to any kitchen.

Introducing the "Precision Chef" Smart Digital Kitchen Scale – a cutting-edge kitchen companion that takes your cooking experience to a whole new level. Designed to be the ultimate culinary aid, this state-of-the-art scale is packed with impressive functionalities to help you achieve perfection in your recipes.

With its exceptional precision, the "Precision Chef" can handle weights up to 5kg with an astounding accuracy of 1g. Whether you're measuring delicate herbs or precisely portioning ingredients for a gourmet dish, this scale ensures you get it right every time.

Equipped with a rechargeable lithium-ion battery boasting a capacity of 1800mAh, the "Precision Chef" is eco-friendly and long-lasting. You'll never have to worry about changing batteries again, and it's ready to use whenever you need it.

The OLED display is the centerpiece of this intelligent scale, presenting weight readings with utmost clarity. The bright and crisp screen makes it easy to read measurements, even in dimly lit kitchens.

Embracing the modern age of connectivity, the "Precision Chef" can be seamlessly linked to the internet via Wi-Fi. This means you can unlock a whole range of possibilities – from accessing real-time data and updates to interacting with the scale through a dedicated web environment.

The web environment is powered by the robust 'Firebase' real-time database, ensuring smooth and efficient data transfer. Say goodbye to manual logging, as your measurements and cooking history are securely stored and readily available whenever you need them.

In summary, the "Precision Chef" Smart Digital Kitchen Scale revolutionizes your cooking experience. It's more than just a scale – it's your trusted culinary partner, helping you bring precision, convenience, and creativity to every recipe you create. Embrace the future of cooking and let "Precision Chef" elevate your culinary prowess to new heights.

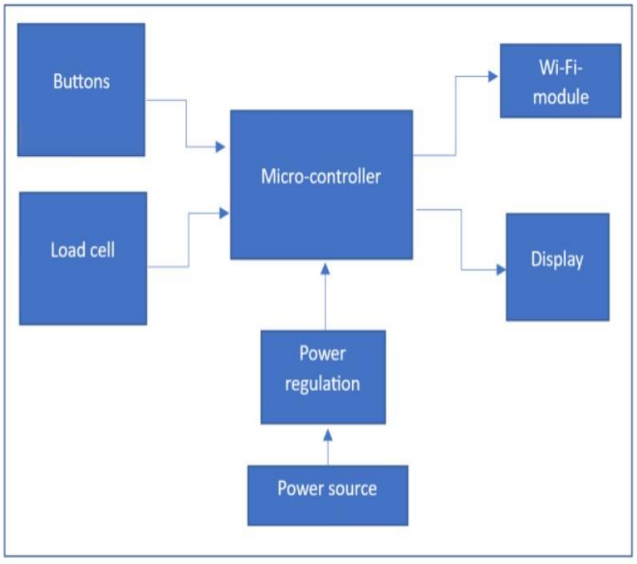


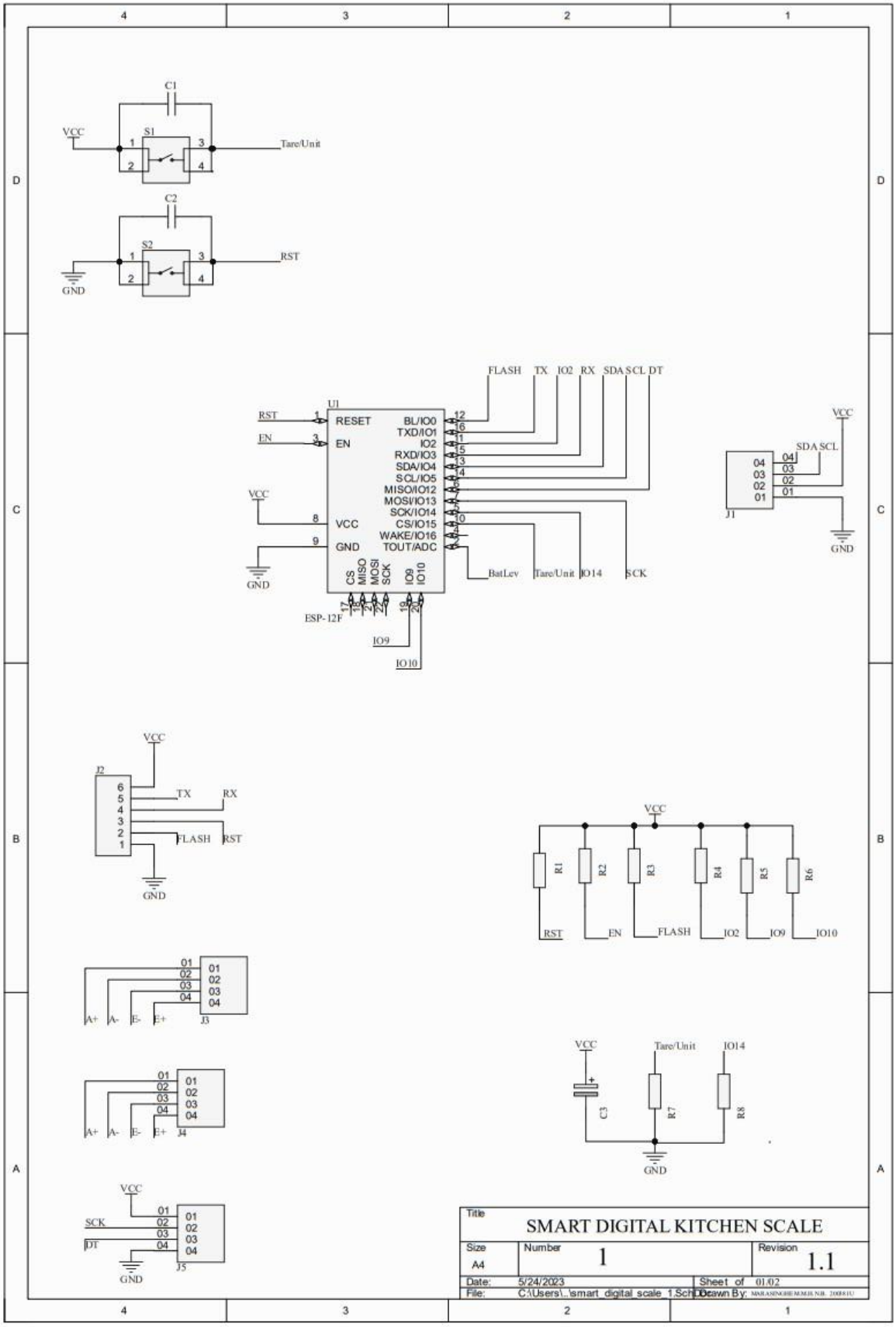

Figure 1 – initial block diagram

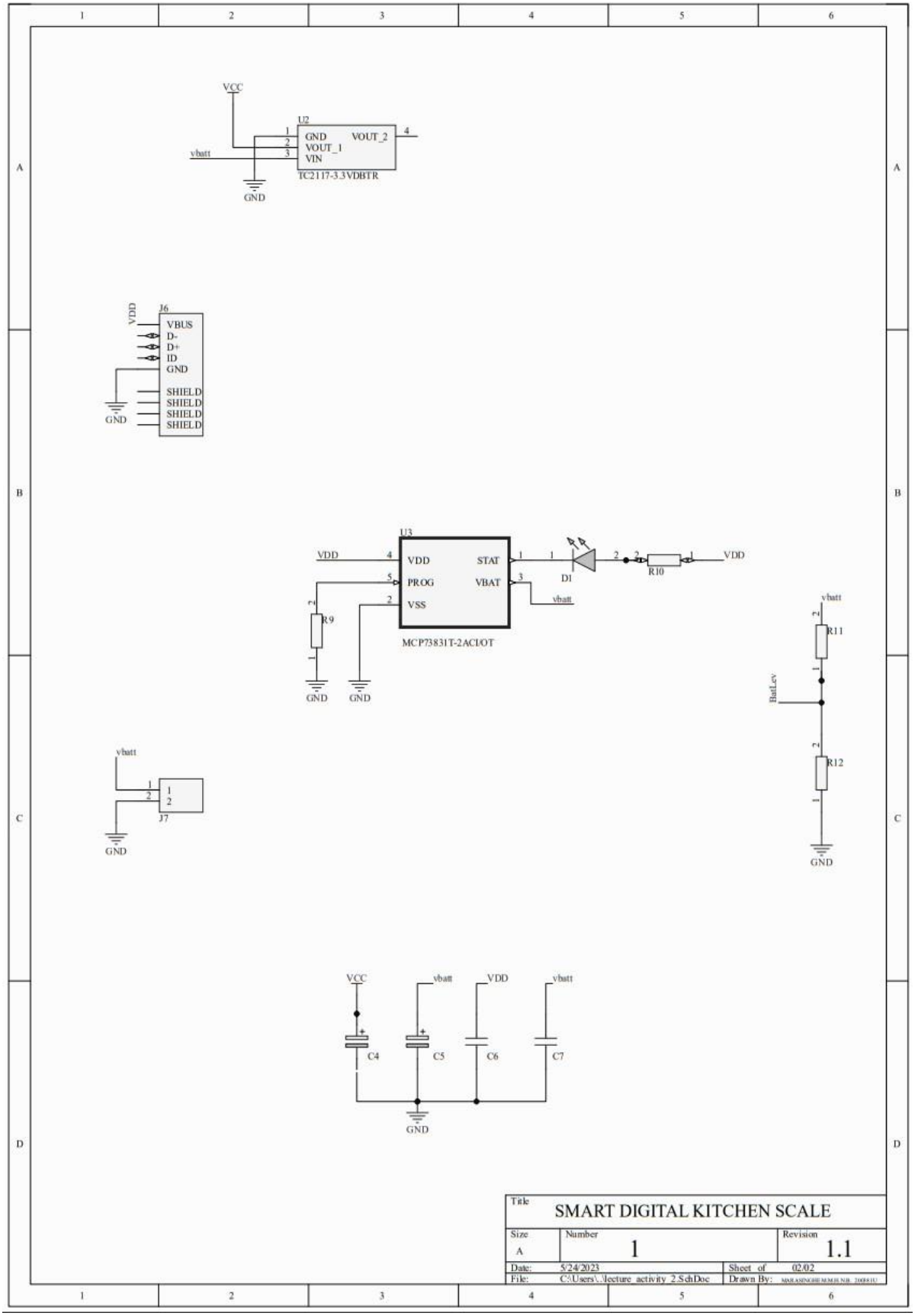
# Specifications

## Basic specifications

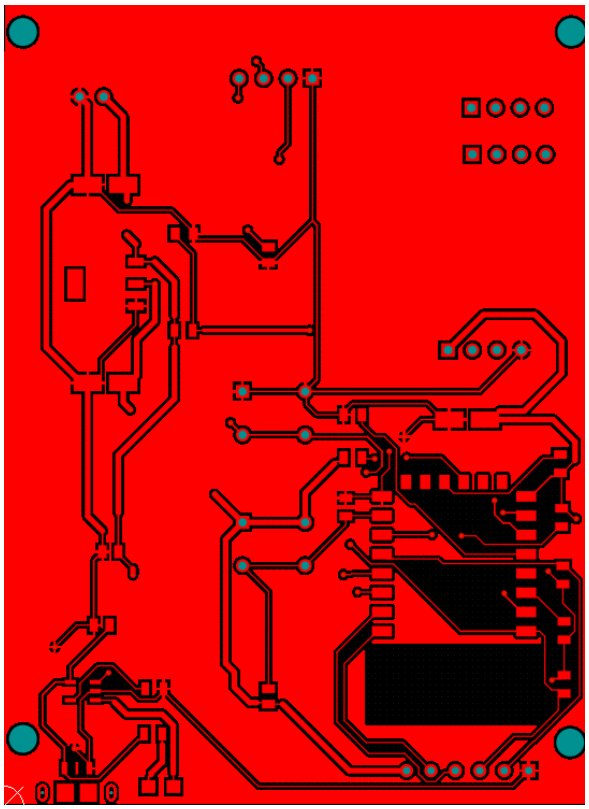
| | |
|---|---|
| Capacity | up to 5 kilograms |
| Maximum error | 1 g (specified by sensor manufacturer) |
| Accuracy | Percentage error <1% for measurements higher than 100g |
| Display (resolution) | 0.91-inch 128 x 24 OLED display with SSD1306 driver |
| Units of measurement | grams and ounces |
| Tare function | The ability to reset the scale to zero, which is useful for weighing |
| Calibration function | The ability to calibrate the scale to ensure accurate |
| Power source | Li-polymer battery 3.7V (with recharging functionality) |
| Enclosure material | PLA |

## Schematic of device

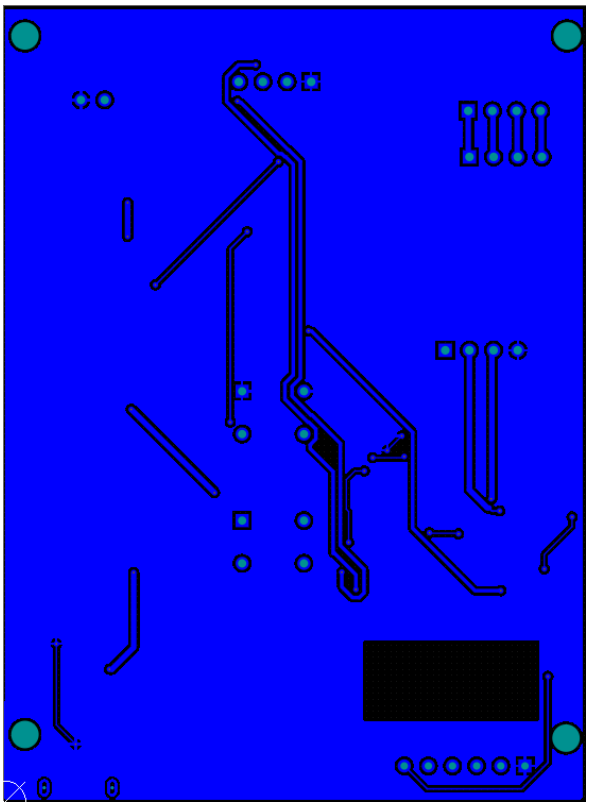




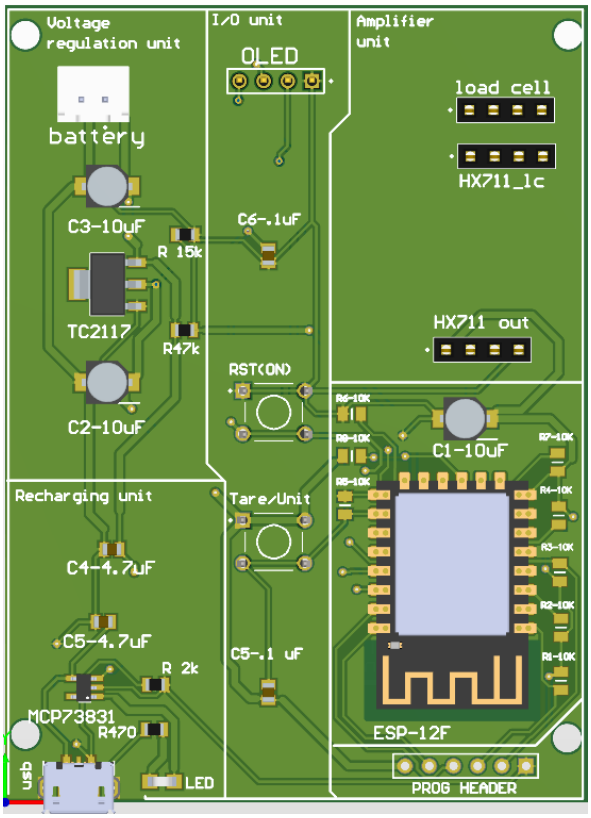

## PCB Design files



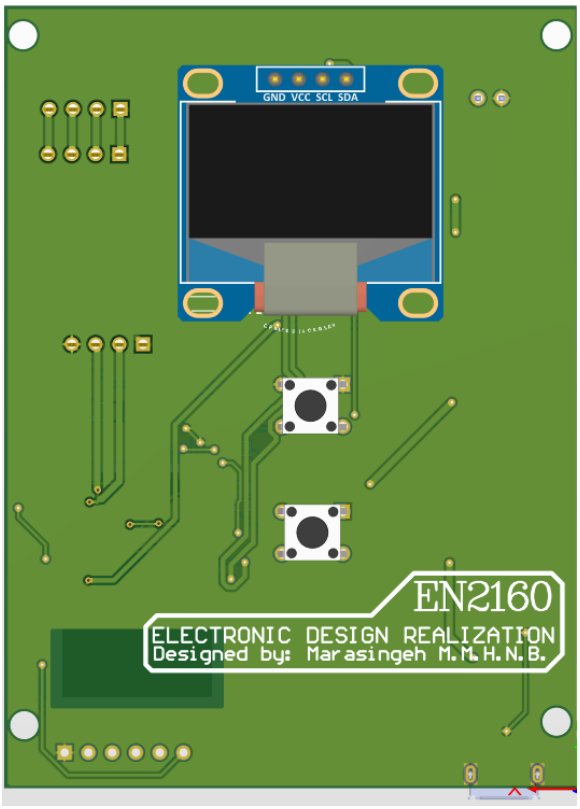
Top layer



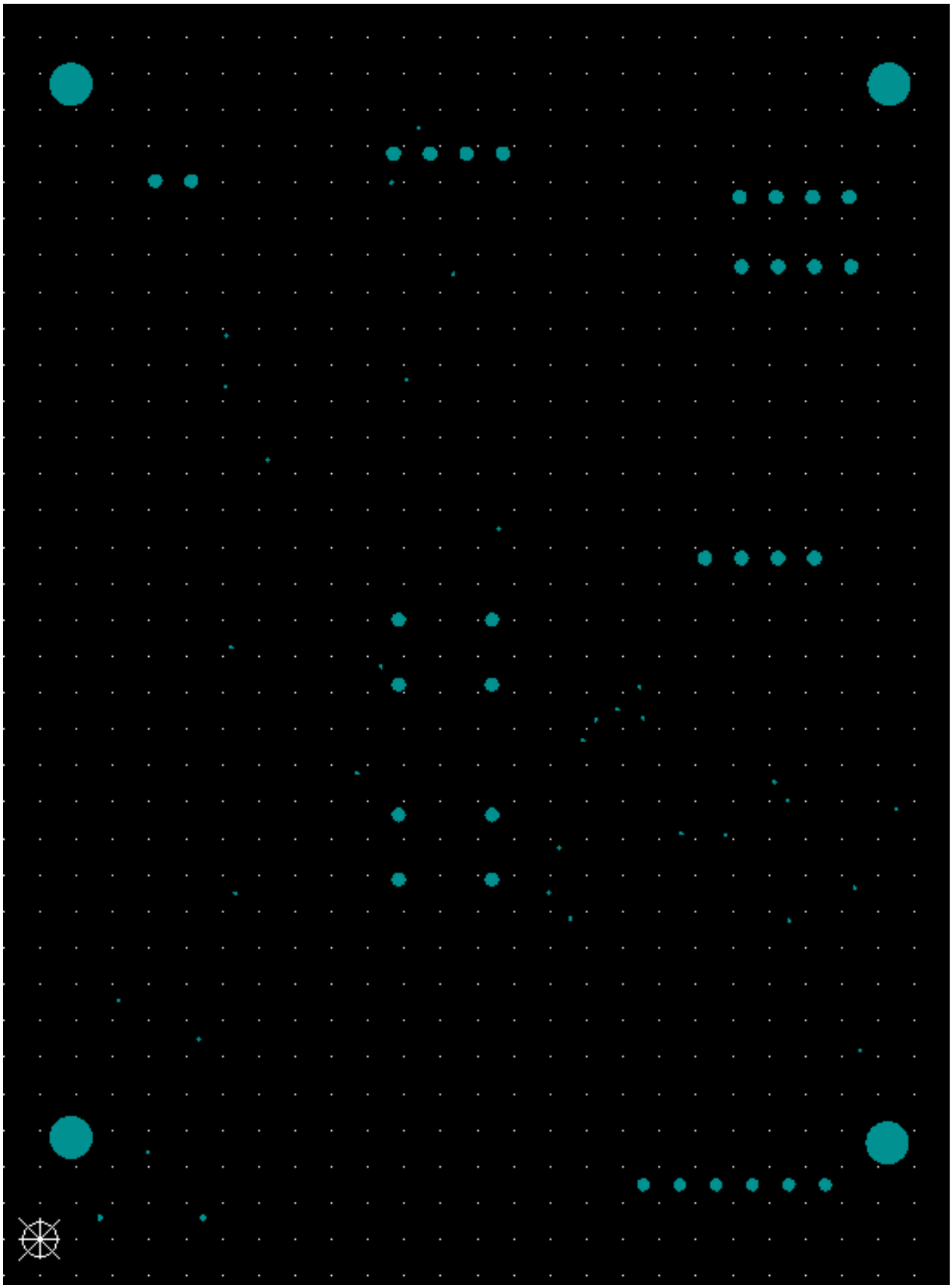
Bottom layer



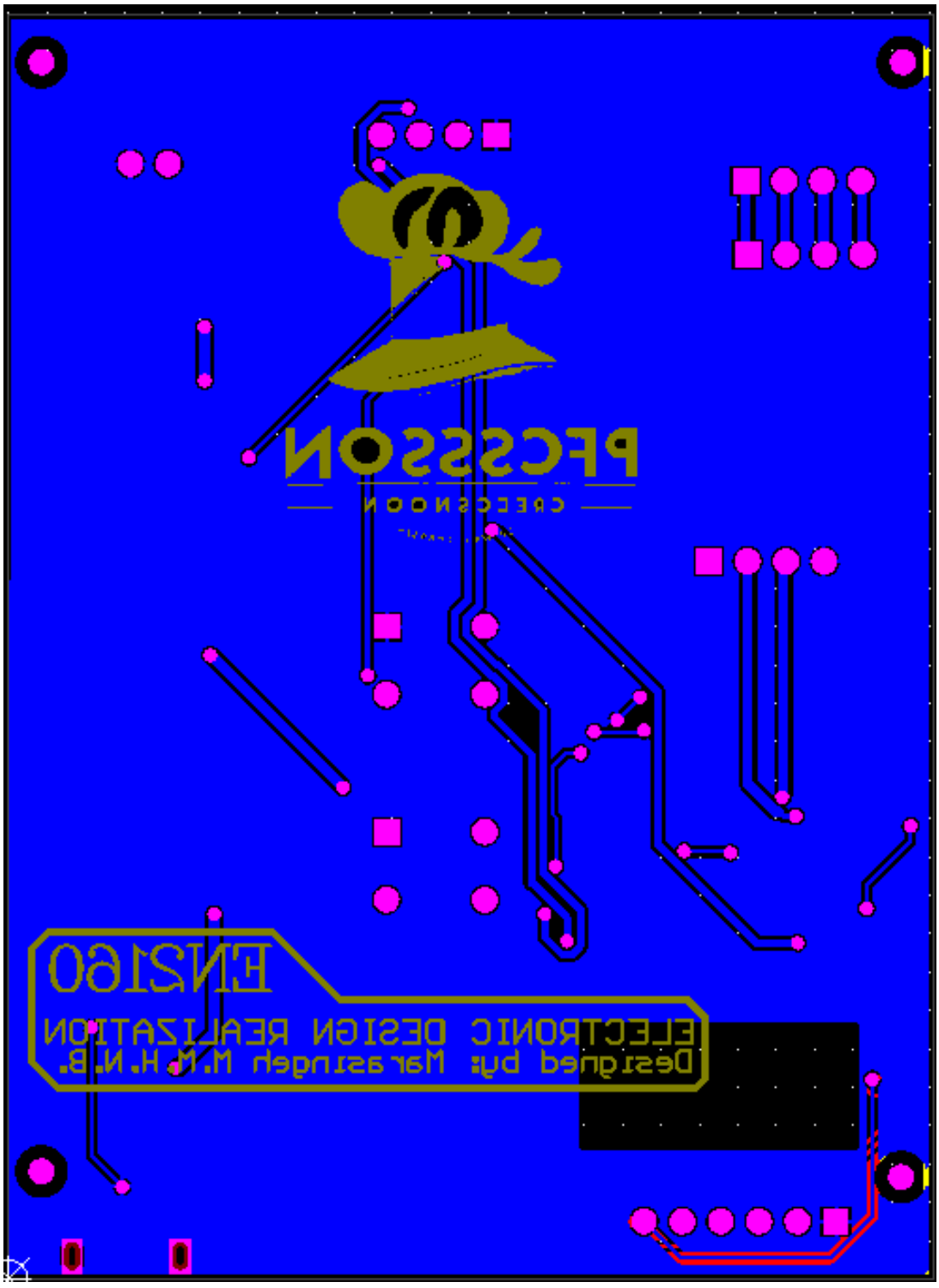

3d view(top)




3d view (bottom)



NC drill drawings


PCBSSON
EN2160
ELECTRONIC DESIGN REALIZATION
Designed by: Marasingeh M.M.H.N.B.

Gerber file

# Bill of materials and supply chain

## BOM

| NO. | Name | Qtn. | Price |
|---|---|---|---|
| 1 | 10 K resistor | 10 | $0.10 |
| 2 | 10uF polarized caps | 6 | $0.18 |
| 3 | .1uF ceramic caps | 4 | $0.12 |
| 4 | 4.7uF ceramic caps | 4 | $0.15 |
| 5 | 15k  resistor | 2 | $0.10 |
| 6 | 47k resistor | 2 | $0.10 |
| 7 | TC2117 | 2 | $1.05 |
| 8 | MCP73831 | 2 | $0.81 |
| 9 | 2k resistor | 2 | $0.10 |
| 10 | 470 resistor | 2 | $0.10 |
| 11 | LED | 2 | $0.13 |
| 12 | .OLED display module (SSD1306) | 1 | $3.01 |
| 13 | Load cell and amplifier module | 1 | $2.57 |
| 14 | push buttons | 2 | $0.40 |
| 15 | pin headers | 1 | $0.10 |
| 16 | PCB printing - JLC PCB (with tax&shipping) | 5 | $5.33 |
| 17 | Enclosure - 3d printed local | 1 | $17.14 |
| 18 | Misellenius | | $5.00 |
| | Total Cost | | $36.39 |

## Supply chain

**Components from mouser:**

1.Thick Film Resistors - SMD CRGCQ 0805 10K 5% SMD Resistor RoHS Compliant By Exemption

2.Aluminum Electrolytic Capacitors - SMD WCAP-ASLI 10uF 16V 20% SMD/SMT RoHS Compliant

3.Multilayer Ceramic Capacitors MLCC - SMD/SMT 50V 0.1uF X7R 0805 10% RoHS Compliant

4.Multilayer Ceramic Capacitors MLCC - SMD/SMT WCAP-CSGP 4.7uF 0805 20% 16V MLCC RoHS Compliant

5.Thick Film Resistors - SMD 0805 15Kohms 5% AEC-Q200 RoHS Compliant By Exemption

6.Thick Film Resistors - SMD 47K 5% RoHS Compliant By Exemption

7.LDO Voltage Regulators 800mA Fixed Output RoHS Compliant

8.Battery Management Charge mgnt contr. RoHS Compliant

9.Thick Film Resistors - SMD 2K 5% RoHS Compliant By Exemption

10.Thick Film Resistors - SMD 1/8Watt 470ohms 1% Commercial Use RoHS Compliant By Exemption

11.Standard LEDs - SMD Amber LED RoHS Compliant

12.micro usb jack

**Components from Chinese suppliers:**

1.OLED display module (SSD1306)

2.Load cell and amplifier module

**Local components:**

1.push buttons

2.pin headers

**PCB printing:**

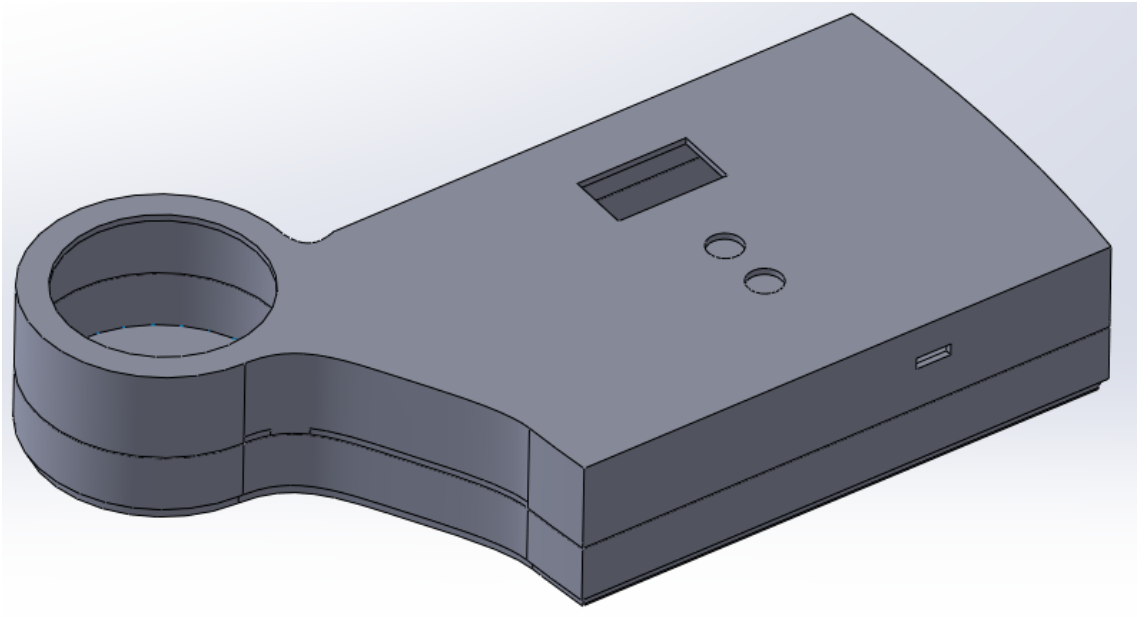
*JLC PCB

**Enclosure:**

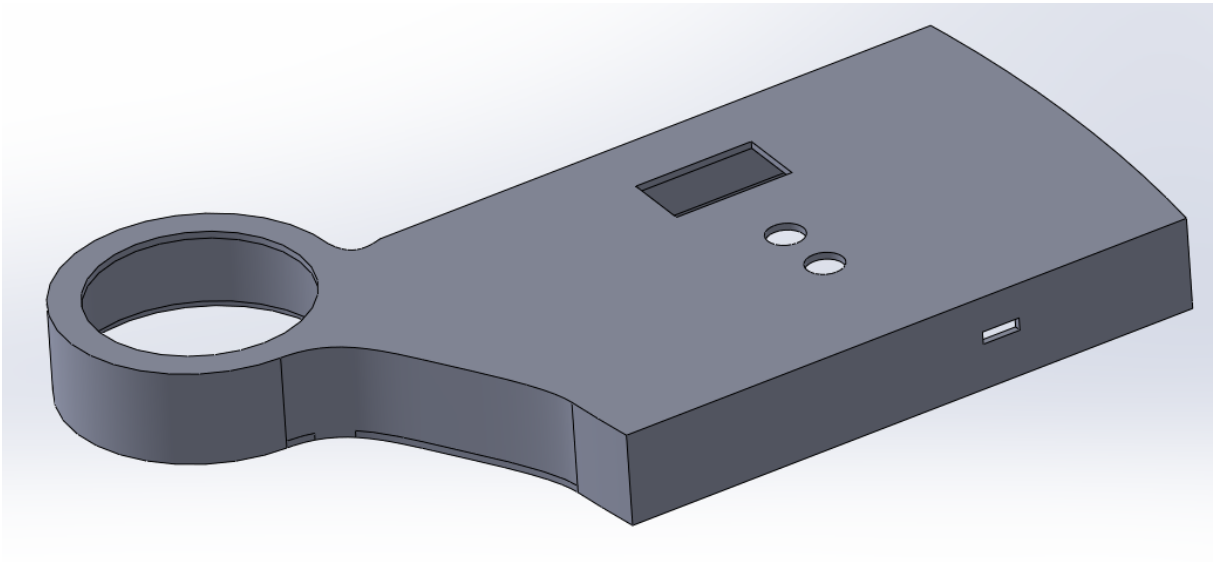
*3D printed - locally

## Enclosure design

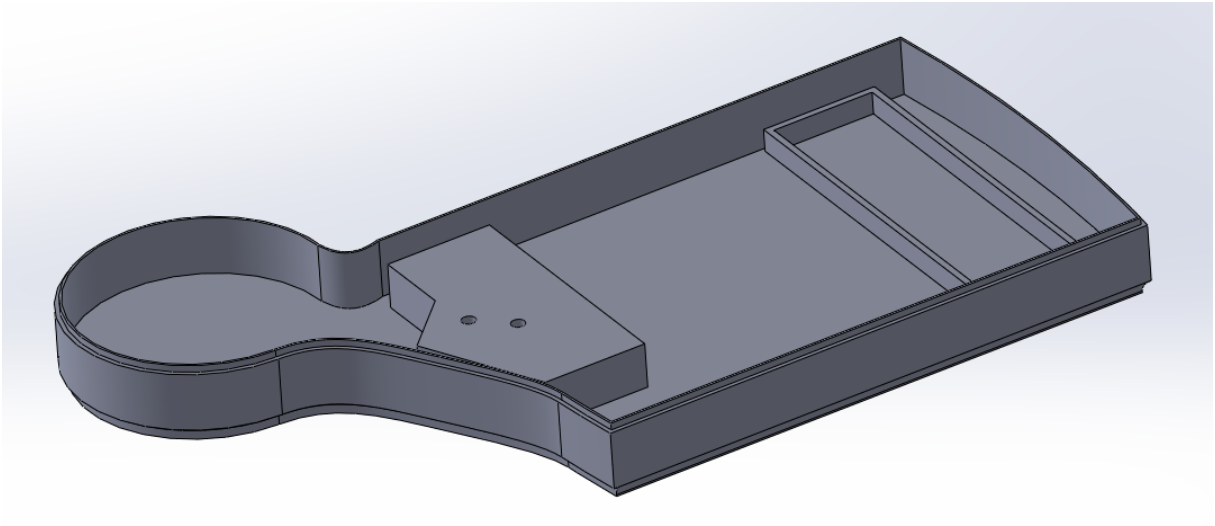
When designing the enclosure for the smart digital kitchen scale(Precision Chef) using SolidWorks software, the central focus was on optimizing user-friendliness and crafting a delightful user experience. The design process revolved around creating an ergonomic and aesthetically pleasing form factor that seamlessly integrates into the user's daily activities. The enclosure design strikes a perfect balance between compactness and functionality. The placement of components and the overall layout were thoughtfully considered to ensure intuitive user interaction. The design also emphasized the optimal positioning of the display, guaranteeing clear visibility of measurement data.



Enclosure – isometric view



Enclosure – top part

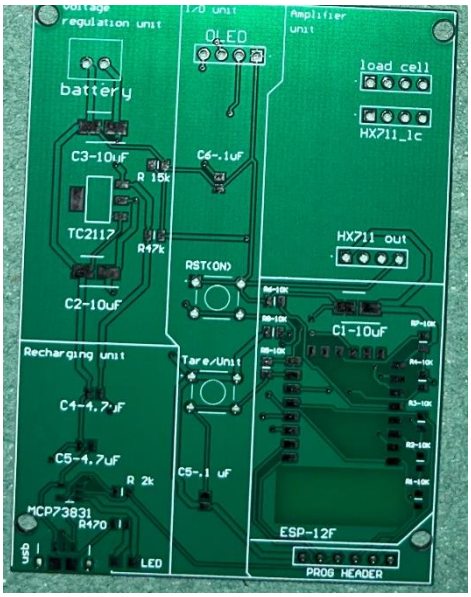


Enclosure – bottom part

## Instructions for assembly

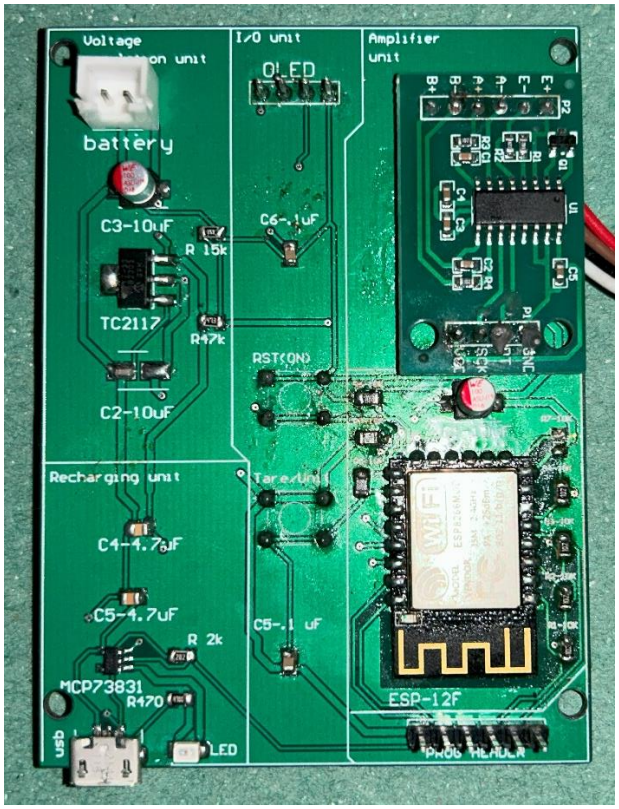
The following sections describe the PCB fabrication process, Soldering process, Enclosure manufacturing process.

### PCB fabrication and component soldering process




Before soldering(top)

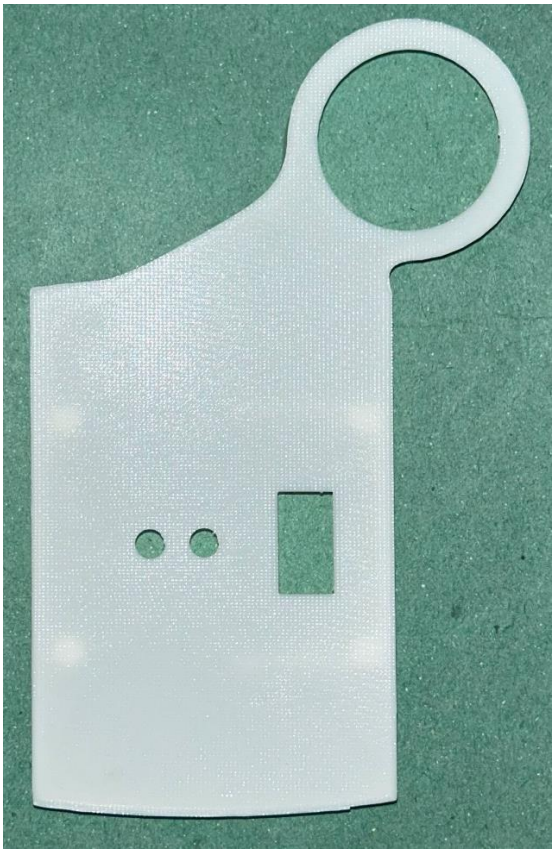

Before soldering(bottom)




After soldering (top)


After soldering (bottom)

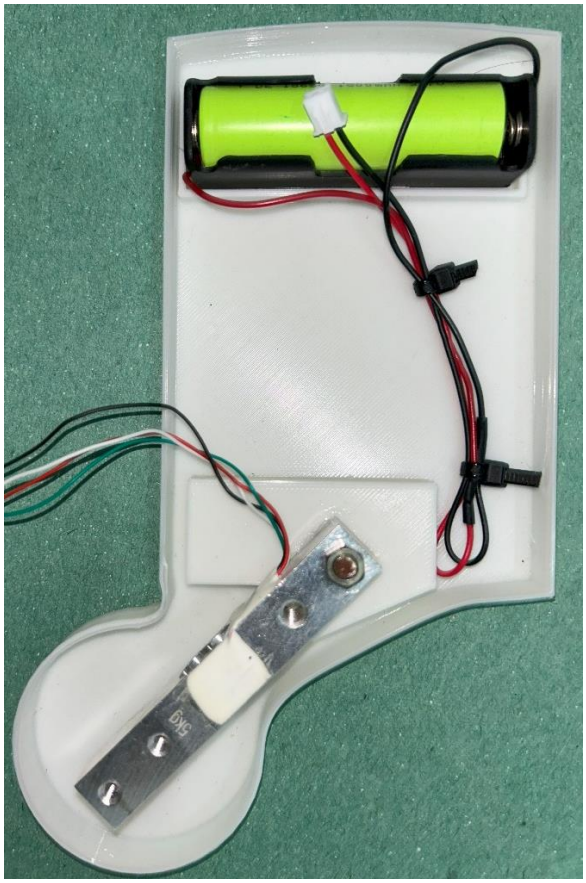
## Enclosure manufacturing process



3d printed enclosure (top part)

3d printed enclosure (bottom part)

## Final assembly



Assembled bottom part

Assembled top part

Tray(with screw for fixing tray)

Fully assembled product

## Tests for functionality

To test the functionality of a smart digital kitchen scale (Precision Chef), you can perform the following steps:

- Inspect the Scale: First, visually inspect the scale for any physical damage or defects. Ensure that the display screen is clear and there are no visible issues with the buttons or sensors.

- Power On: Turn on the kitchen scale using the power button or any specified method.

- Calibration: Some digital scales might require calibration before use. Check the user manual for instructions on how to calibrate the scale properly. Calibration is essential to ensure accurate measurements.

- Units of Measurement: Check if the scale allows you to switch between different units of measurement (grams, ounces, pounds, kilograms, etc.). Test each unit to ensure it displays the correct measurements.

- Tare Function: The tare function allows you to zero out the weight of the container or plate so that you can measure only the contents. Test the tare function by placing an empty container on the scale, pressing the tare button, and verifying if the scale returns to zero.

- Weighing Items: Place known weights or standard items with pre-determined weights (e.g., 100g, 250g, 500g) on the scale to check its accuracy. Ensure that the scale displays the correct weight for each item.

- Capacity: Test the scale's maximum weight capacity by gradually adding items until you reach its limit. Ensure it does not malfunction or display inaccurate readings when near its maximum capacity.

- Auto Shut-off: Many digital scales have an auto shut-off feature to conserve battery power. Verify that the scale turns off automatically after a period of inactivity(2 min) and turns on when you interact with it.

- Connectivity (if applicable): If the scale has smart features, such as Bluetooth or Wi-Fi connectivity, test the connection with a compatible device (smartphone, tablet, etc.). Check if it syncs data and functions as expected with the associated app or software.

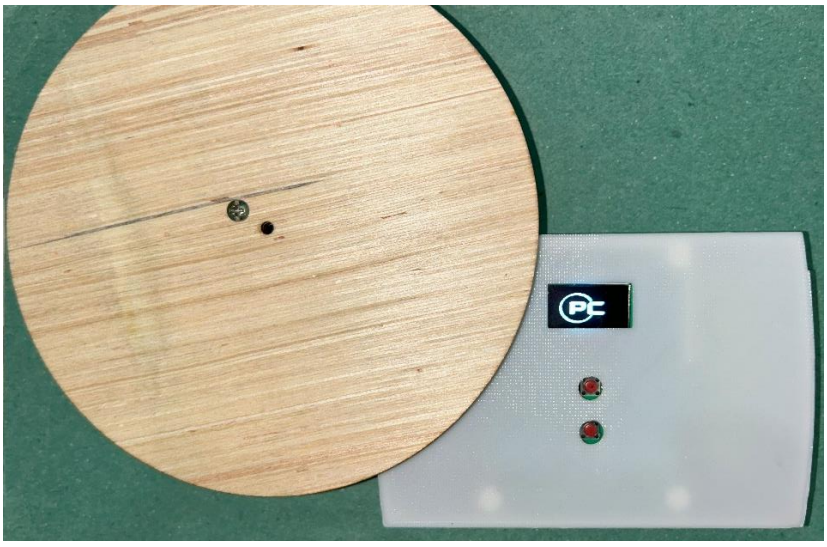
- Battery Life: Test the battery life by leaving the scale on for an extended period or following the manufacturer's instructions for battery testing.

- Accuracy and Precision: Compare the measurements from the smart kitchen scale with another accurate scale or reference weight to ensure its accuracy and precision.

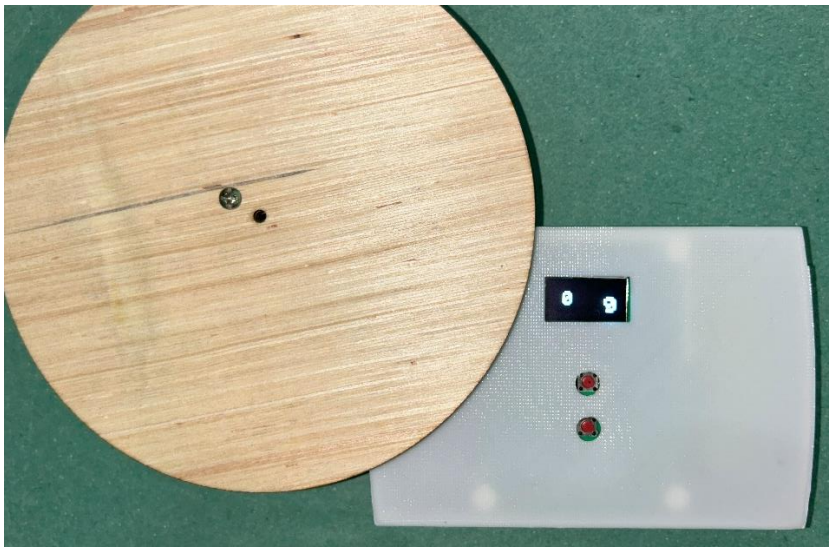
By conducting these tests, you can verify the functionality, accuracy, and reliability of your smart digital kitchen scale. If you encounter any issues or discrepancies, refer to the user manual or contact the manufacturer for assistance.

## Images captured from a basic functionality test (not an accuracy test)

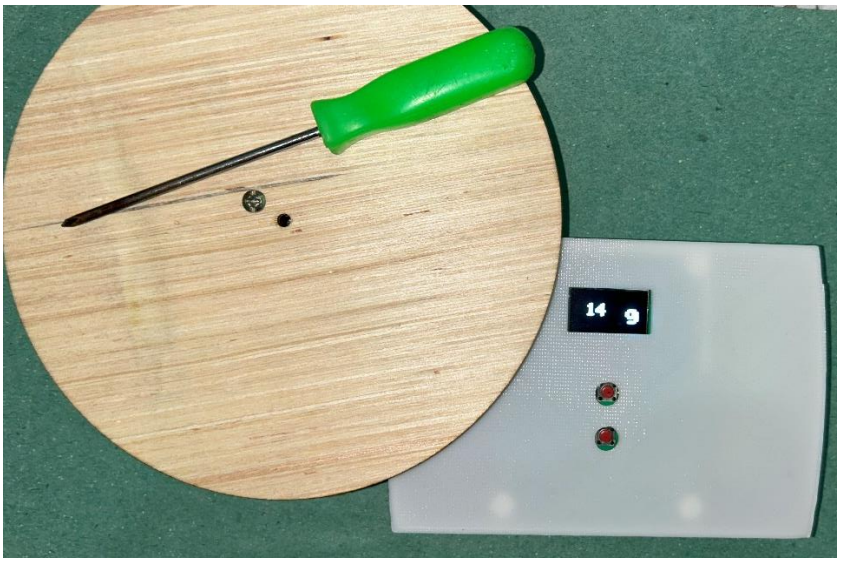
1. Powering up (try to connect to Wi-Fi, if no connection found proceed to offline operation.)



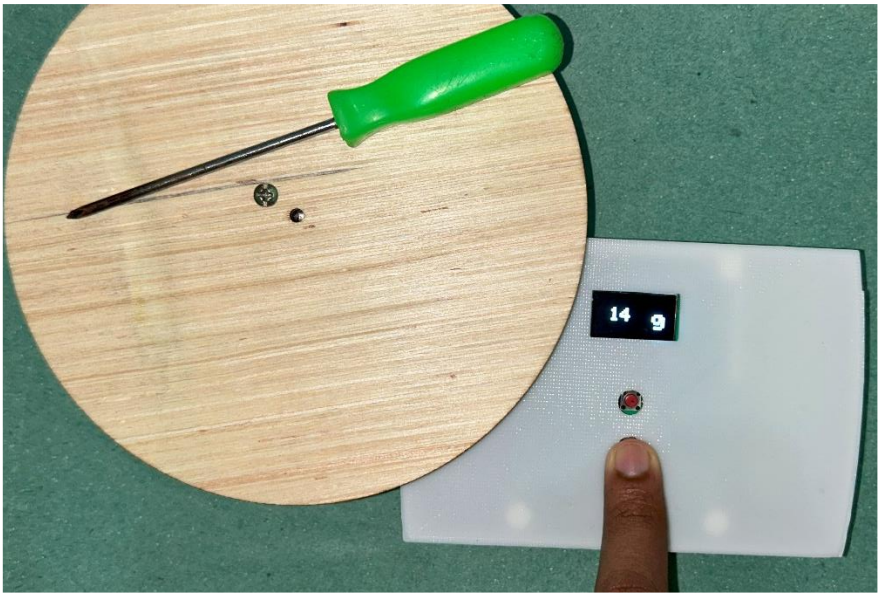
2. Ready to measure.



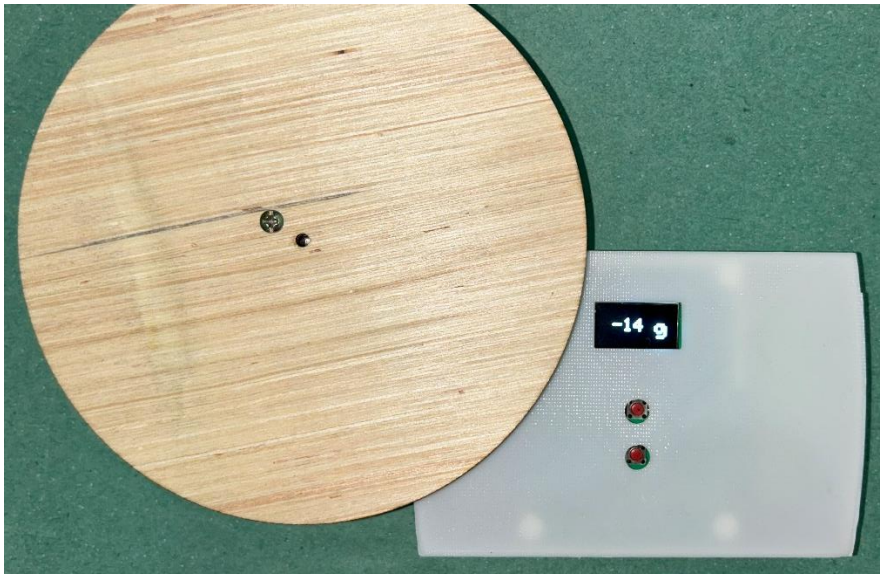

3. Measuring



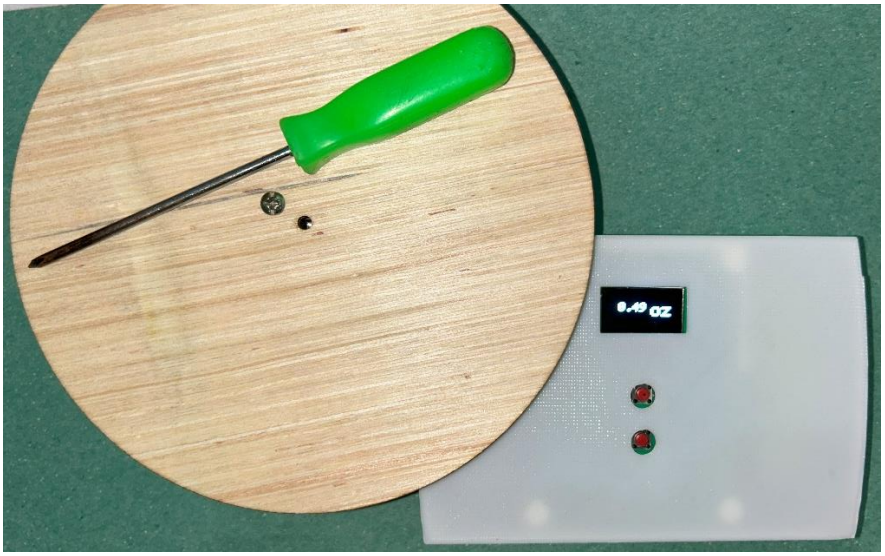

4. Short press (unit/tare button)



5. Remove weight and check whether negative value of previous measurement appears.




6. Long press unit/tare button
7. Check whether ounce value of the initial reading appears.



# Appendix

## Appendix I – code

```
#include <Arduino.h>
#include "HX711.h"
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <logo.h>
#include <wifi_connected.h>
#include <wifi_disconnected.h>
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////
#if defined(ESP32)
  #include <WiFi.h>
#elif defined(ESP8266)
  #include <ESP8266WiFi.h>
#endif
#include <Firebase_ESP_Client.h>

//Provide the token generation process info.
#include "addons/TokenHelper.h"
//Provide the RTDB payload printing info and other helper functions.
#include "addons/RTDBHelper.h"

// Insert your network credentials
#define WIFI_SSID "test"
#define WIFI_PASSWORD "12345678"

// Insert Firebase project API Key
#define API_KEY "AIzaSyBzWWLQnqViUEYKUp5fx_aXHsadgWly_YQ"

// Insert RTDB URLefine the RTDB URL */
```

```
#define DATABASE_URL "https://digitalkitchenscale-3af0b-default-rtdb.asia-southeast1.firebasedatabase.app/"

//Define Firebase Data object
FirebaseData fbdo;

FirebaseAuth auth;
FirebaseConfig config;

unsigned long sendDataPrevMillis = 0;
int count = 0;
bool signupOK = false;
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////

// HX711 circuit wiring
const int LOADCELL_DOUT_PIN = 12;
const int LOADCELL_SCK_PIN = 13;

HX711 scale;
int reading;
int lastReading;
//REPLACE WITH YOUR CALIBRATION FACTOR
#define CALIBRATION_FACTOR -478.507

//OLED Display
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels



// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
#define OLED_RESET     -1 // Reset pin # (or -1 if sharing Arduino reset pin)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

//Button
```

```
#define BUTTON_PIN 15
//Pushbutton button(BUTTON_PIN);
int currentState = 0;
int buttonPressed = 0;
long pressedTime;
long releasedTime;



//Button short press and long press
const int SHORT_PRESS_TIME = 2000; // 2000 milliseconds

//unit changing
int unit = 0; // 0 for grams and 1 for ounces

//sleep mode
int change = 1;
int timmer = 0;
long beginTime;
long currentTime;
const int TIMEOUT = 60000;



void displayWeight(int weight){
  display.clearDisplay();
  display.setTextColor(WHITE);
  // Display static text
  //display.setTextSize(1);
  //display.setCursor(0, 10);
  //display.println("unit:");
  //display.println(unit);
  //display.display();
  display.setCursor(24, 20);
  if(!unit) { display.setTextSize(3); display.print(weight);}
  else {display.setTextSize(2); display.print(weight*0.035274);}
  display.display();
  //display.print(" ");
```

```
  display.setTextSize(4);
  display.setCursor(96, 20);
  if(!unit) display.print("g");
  else{ display.setCursor(80, 20); display.setTextSize(3); display.print("oz");}
  display.display();
}

void setup() {
  Serial.begin(115200);
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("SSD1306 allocation failed"));
    for(;;);
  }
  //delay(2000);
  display.clearDisplay();
  display.drawBitmap(0, 0, logoPC, 128, 64, 1);
  display.display();

////////////////////////////////////////////////////////////////////////////////////////////////////
//////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////
//////////////////////
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  long wifitime = millis();
  long current = 0;
  while (WiFi.status() != WL_CONNECTED){
    current = millis();
    if(current-wifitime > 60000) break;
    Serial.print(".");
    delay(300);
  }
  if (WiFi.status() != WL_CONNECTED) {display.drawBitmap(0, 0, wifi_connected, 128, 64, 1);}
  else {display.drawBitmap(0, 0, wifi_disconnected, 128, 64, 1);}
  Serial.println();
  Serial.print("Connected with IP: ");
```

```
  Serial.println(WiFi.localIP());
  Serial.println();

  /* Assign the api key (required) */
  config.api_key = API_KEY;

  /* Assign the RTDB URL (required) */
  config.database_url = DATABASE_URL;

  /* Sign up */
  if (Firebase.signUp(&config, &auth, "", "")){
    Serial.println("ok");
    signupOK = true;
  }
  else{
    Serial.printf("%s\n", config.signer.signupError.message.c_str());
  }

  /* Assign the callback function for the long running token generation task */
  config.token_status_callback = tokenStatusCallback; //see addons/TokenHelper.h

  Firebase.begin(&config, &auth);
  Firebase.reconnectWiFi(true);

////////////////////////////////////////////////////////////////////////////////////////////////////
//////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////
//////////////////

  display.setTextColor(WHITE);

  Serial.println("Initializing the scale");
  scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);

  scale.set_scale(-518);   // this value is obtained by calibrating the scale with known weights
  scale.tare();            // reset the scale to 0
```

```
 pinMode(BUTTON_PIN,INPUT);

}

void loop() {

 //Serial.println(button.getSingleDebouncedPress());
 buttonPressed = digitalRead(BUTTON_PIN);
 if (buttonPressed && !currentState){
   pressedTime = millis();
   currentState = HIGH;
 }
 else if(!buttonPressed && currentState){
  releasedTime = millis();
  currentState = LOW;
  // check whether the press is short or long and tare or change unit accordingly
  long pressDuration = releasedTime - pressedTime;
   //Serial.print("Button Pressed time: ");
    //Serial.print(pressDuration);
  if( pressDuration < SHORT_PRESS_TIME ){
    //Serial.print("tare...");
    //Serial.read();
    scale.tare();
  }
  else{
    if(unit) unit = 0;
    else unit = 1;
    displayWeight(reading);
  }

 }


 if (scale.wait_ready_timeout(200)) {
  reading = round(scale.get_units());
  //Serial.print("Weight: ");
```

```
  //Serial.println(reading);
  if (reading != lastReading){
    displayWeight(reading);
    change=1;
  }
  else change=0;

  lastReading = reading;

  if(change) {timmer=0;}
  else if(!change && !timmer){ timmer=1; beginTime = millis();}
  else if(!change && timmer) {
     currentTime = millis();
     if(currentTime-beginTime>TIMEOUT){
      display.clearDisplay();
      display.display();
      ESP.deepSleep(0);

      }
   }



 }
 else {
  Serial.println("HX711 not found.");
 }

////////////////////////////////////////////////////////////////////////////////////////////////////
//////////////

////////////////////////////////////////////////////////////////////////////////////////////////////
/////////////
  if (Firebase.ready() && signupOK && (millis() - sendDataPrevMillis > 500 || sendDataPrevMillis ==
0)){
   sendDataPrevMillis = millis();
   // Write an Int number on the database path test/int
   if (Firebase.RTDB.setInt(&fbdo, "test/count", count)){
```

```
    Serial.println("PASSED");
    Serial.println("PATH: " + fbdo.dataPath());
    Serial.println("TYPE: " + fbdo.dataType());
   }
   else {
    Serial.println("FAILED");
    Serial.println("REASON: " + fbdo.errorReason());
   }
   count++;

   // Write weight
   if (Firebase.RTDB.setFloat(&fbdo, "test/weight", reading)){
    Serial.println("PASSED");
    Serial.println("PATH: " + fbdo.dataPath());
    Serial.println("TYPE: " + fbdo.dataType());
   }
   else {
    Serial.println("FAILED");
    Serial.println("REASON: " + fbdo.errorReason());
   }

   // write unit 1- ounce 0-grams
   if (Firebase.RTDB.setFloat(&fbdo, "test/unit", unit)){
    Serial.println("PASSED");
    Serial.println("PATH: " + fbdo.dataPath());
    Serial.println("TYPE: " + fbdo.dataType());
   }
   else {
    Serial.println("FAILED");
    Serial.println("REASON: " + fbdo.errorReason());
   }
  }

////////////////////////////////////////////////////////////////////////////////////////////////////
///////////////
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//////////////
}
```