

**#include <Wire.h>**

**#include <Adafruit\_GFX.h>**

**#include <Adafruit\_SSD1306.h>**

**#include <ESP8266WiFi.h>**

**#include <HX711\_ADC.h>**

**#include <EEPROM.h>**

**#define SCREEN\_WIDTH 128 //**  
**OLED display width, in pixels**

**#define SCREEN\_HEIGHT 64 //**  
**OLED display height, in pixels**

**#define OLED\_RESET 0 // Reset**  
**pin # (or -1 if sharing Arduino reset**  
**pin)**

**Adafruit\_SSD1306**

**display(SCREEN\_WIDTH,**

**SCREEN\_HEIGHT, &Wire,  
OLED\_RESET);**

**const int HX711\_dout = D5; //mcu >  
HX711 dout pin**

**const int HX711\_sck = D6; //mcu >  
HX711 sck pin**

**HX711\_ADC LoadCell(HX711\_dout,  
HX711\_sck);**

**const int calVal\_eepromAdress =  
0;**

**long t;**

**String apiKey =**

**"APDPDL4C7U2C60YH"; //**

**Enter your Write API key from  
ThingSpeak**

**const char \*ssid = "Taramma";    //  
replace with your wifi ssid and  
wpa2 key**

**const char \*pass = "12345678";**

**const char\* server =  
"api.thingspeak.com";**

**WiFiClient client;**

**void setup() {**

**if**

**(!display.begin(SSD1306\_SWITCH  
CAPVCC, 0x3C)) { // Address 0x3D  
for 128x64**

**Serial.println(F("SSD1306  
allocation failed"));**

```
    for (;;); // Don't proceed, loop  
forever  
}  
  
Serial.begin(115200);  
delay(10);  
  
Serial.println("Connecting to ");  
Serial.println(ssid);  
WiFi.begin(ssid, pass);  
while (WiFi.status() !=  
WL_CONNECTED)  
{  
    delay(500);  
    Serial.print(".");  
}
```

```
Serial.println("");  
Serial.println("WiFi connected");  
LoadCell.begin();  
float calibrationValue;  
#if defined(ESP8266)||  
defined(ESP32)  
    EEPROM.begin(512);  
#endif  
  
EEPROM.get(calVal_eeepromAdres  
s, calibrationValue);  
  
long stabilizingtime = 2000; //  
preciscion right after power-up  
can be improved by adding a few  
seconds of stabilizing time
```

**boolean \_tare = true; //set this to false if you don't want tare to be performed in the next step**

**LoadCell.start(stabilizingtime, \_tare);**

**if  
(LoadCell.getTareTimeoutFlag()) {**

**Serial.println("Timeout, check MCU>HX711 wiring and pin designations");**

**while (1);**

**}**

**else {**

**LoadCell.setCalFactor(calibration**

**Value); // set calibration value  
(float)**

**Serial.println("Startup is  
complete");**

**}**

**}**

**void loop() {**

**static boolean newDataReady =  
0;**

**const int serialPrintInterval = 0;  
//increase value to slow down  
serial print activity**

**// check for new data/start next  
conversion:**

```
if (LoadCell.update())
newDataReady = true;

// get smoothed value from the
dataset:

if (newDataReady) {

    if (millis() > t +
serialPrintInterval) {

        float i = LoadCell.getData();

        Serial.print("Load_cell output
val: ");

        Serial.println(i);

        float w =4; //

        display.clearDisplay();

        display.setTextColor(WHITE);
```



```
display.setTextSize(1);  
display.setCursor(5, 0);  
display.print("TOTAL WEIGHT  
");  
display.setCursor(10, 15);  
display.setTextSize(1);  
display.print(i);  
display.print("gm");  
display.display();  
int k = i / w;  
display.setTextSize(1);  
display.setCursor(0, 30);  
display.print("PIECES LEFT ");  
display.setCursor(30, 45);
```

```
display.setTextSize(2);  
display.print(k);  
display.print(" Nos");  
display.display();  
if (client.connect(server, 80)) //  
"184.106.153.149" or  
api.thingspeak.com  
{  
    String postStr = apiKey;  
    postStr += "&field1=";  
    postStr += String(i);  
    postStr += "&field2=";  
    postStr += String(k);  
    postStr += "\r\n\r\n";
```

```
client.print("POST /update  
HTTP/1.1\n");
```

```
client.print("Host:  
api.thingspeak.com\n");
```

```
client.print("Connection:  
close\n");
```

```
client.print("X-  
THINGSPEAKAPIKEY: " + apiKey +  
"\n");
```

```
client.print("Content-Type:  
application/x-www-form-  
urlencoded\n");
```

```
client.print("Content-Length:  
");
```

```
client.print(postStr.length());
```

```
    client.print("\n\n");
    client.print(postStr);
    Serial.println(" Sending to
Thingspeak.");
}

client.stop();
Serial.println("Waiting...");
// thingspeak needs minimum
15 sec delay between updates
delay(10);
newDataReady = 0;
t = millis();
}
}
```

}