```
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include "HX711.h"
#define LOADCELL_DOUT_PIN  3
#define LOADCELL_SCK_PIN  2

int BUZZ_LED = 4;
int RELAY = 5;
int SPDT = 6;
float R = 8.314;
float Gm = 0.6485;
float At = 19.635; // diameter of 5mm;
int temperature;
unsigned long StartTime = 0;
unsigned long CurrentTime = 0;

HX711 scale;
float calibration_factor = -35900;

const char* ssid = "TEAM_C";
const char* password = "MARSSS";
const char* host = "script.google.com";
const int httpsPort = 443;

WiFiClientSecure client;
String GAS_ID = "AKfycbyLbKLg5KFFwbiNnDfT27LRD88IcaQ_TDN8kTiHwRzDJJPy3SCd";

void setup() {
  Serial.begin(9600);
  pinMode(BUZZ_LED, OUTPUT);
  pinMode(RELAY, OUTPUT);
  pinMode(SPDT, INPUT);
  scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
  scale.set_scale();
  Serial.begin(115200);
  Serial.println();
  Serial.print("connecting to ");
  Serial.println(ssid);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void loop() {
  {  int SwitchPos = digitalRead(SPDT);
  if (SwitchPos == 1) {
    int x;
    for (x = 0; x < 10; x++){
      digitalWrite(BUZZ_LED, HIGH);
      delay(500);
      digitalWrite(BUZZ_LED, LOW);
      delay(500);
    }
```

```
    digitalWrite(RELAY, HIGH);
    delay(1000);
  }
  else {
    digitalWrite(RELAY, LOW);
  }
  scale.tare();
  }

  StartTime = millis();
  while(scale.get_units() > 0){
  int adc_val = analogRead(A1);
  temperature = (((adc_val * 4.88) - 0.0027) / 10) - 25;
  Serial.print("Temperature = ");
  Serial.print(temperature);

  int weight = scale.get_units();
  float thrust = weight*9.81;
  scale.set_scale(calibration_factor);
  Serial.print(" | Thrust: ");
  Serial.print(thrust, 1);
  Serial.print(" Newtons");
  Serial.println();

  if (Serial.available()){
    char temp = Serial.read();
    if (temp == '+')
      calibration_factor += 10;
    else if (temp == '-')
      calibration_factor -= 10;
  }
}
  CurrentTime = millis();
  int Bt = (CurrentTime - StartTime);
  //delay(1000);
  float Mf=abs(scale.get_value());
  float mdot=abs(Mf/Bt);
  int Tc = (temperature / 0.909);
  float Pc = ((mdot / At) * ( (sqrt(R * Tc)) / Gm ));

  if(Pc > 0){
  Serial.print("Chamber Pressure: ");
  Serial.print(Pc);
  Serial.print(" Pascals");

  Serial.println();
  }

sendData(temperature, thrust);
}

void sendData(int temp, int thr){
  client.setInsecure();
  Serial.print("connecting to ");
  Serial.println(host);
  if (!client.connect(host, httpsPort)) {
    Serial.println("connection failed");
    return;
  }
```

```
  String string_temperature =  String(temp, DEC);
  String string_thrust =  String(thr, DEC);
  String url = "/macros/s/" + GAS_ID + "/exec?temperature=" + string_temperature + "&thrust=" +
string_thrust;
  Serial.print("requesting URL: ");
  Serial.println(url);

  client.print(String("GET ") + url + " HTTP/1.1\r\n" +
         "Host: " + host + "\r\n" +
         "User-Agent: BuildFailureDetectorESP8266\r\n" +
         "Connection: close\r\n\r\n");
  Serial.println("request sent");
  while (client.connected()) {
  String line = client.readStringUntil('\n');
  if (line == "\r") {
    Serial.println("headers received");
    break;
  }
  }
}
```