Introduction – The purpose of this device is to record stats about your hikes and record them over time. It used the IMU sensor to count your steps and the ENV sensor to collect environmental data. You can then send this info to thingspeak for data storage.

Implementation – the implementation section of your lab report should include the following:

1. BoM (bill of materials)
    - M5stickC with wrist strap
    - Thingspeak account

2. A video demonstration and a picture of the operating unit

https://www.youtube.com/watch?v=HQDGVpcVix8

3. Source code (the sketch file and any additional code)

```
//WiFi
#include "ThingSpeak.h"
#include "secrets.h"

unsigned long myChannelNumber = SECRET_CH_ID;
const char * myWriteAPIKey = SECRET_WRITE_APIKEY;

#include <WiFi.h>
#include <M5StickC.h>
#include "time.h"

//ENV
#include "DHT12.h"
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP280.h>
#include "bmm150.h"
#include "bmm150_defs.h"
DHT12 dht12;
BMM150 bmm = BMM150();
bmm150_mag_data value_offset;
Adafruit_BMP280 bme;
//connect to wifi with secrets
char ssid[] = SECRET_SSID;   // your network SSID (name)
char pass[] = SECRET_PASS;   // your network password
int keyIndex = 0;            // your network key index number (needed only for WEP)
WiFiClient  client;

//variables for IMU sensor
float accX = 0.0F;
float accY = 0.0F;
float accZ = 0.0F;
```

```
33
34 float maxaccX = 0.0f;
35 float maxaccY = 0.0f;
36 float maxaccZ = 0.0f;
37
38 float accVectorLng = 0.0f;
39 float maxaccVectorLng = 0.0f;
40
41 float steps = 0.0f;
42 float maxTemp = 0.0f;
43
44 //clock
45 // ntp Server that is used for timesync
46 char* ntpServer =  "de.pool.ntp.org";
47
48 // define what timezone you are in
49 int timeZone = 7200;
50
51 // delay workarround
52 int tcount = 0;
53
54 // LCD Status
55 bool LCD = true;
56 bool clockDisplayed = false;
57 RTC_TimeTypeDef RTC_TimeStruct;
58 RTC_DateTypeDef RTC_DateStruct;
59
60
61 // Syncing time from NTP Server
62 void timeSync() {
63     M5.Lcd.setTextSize(1);
64     Serial.println("Syncing Time");
65     Serial.printf("Connecting to %s ", ssid);
66     M5.Lcd.fillScreen(BLACK);
```

```
  M5.Lcd.fillScreen(BLACK);
  M5.Lcd.setCursor(20, 15);
  M5.Lcd.println("connecting WiFi");
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println(" CONNECTED");
  M5.Lcd.fillScreen(BLACK);
  M5.Lcd.setCursor(20, 15);
  M5.Lcd.println("Connected");
  // Set ntp time to local
  configTime(timeZone, 0, ntpServer);

  // Get local time
  struct tm timeInfo;
  if (getLocalTime(&timeInfo)) {
    // Set RTC time
    RTC_TimeTypeDef TimeStruct;
    TimeStruct.Hours   = timeInfo.tm_hour;
    TimeStruct.Minutes = timeInfo.tm_min;
    TimeStruct.Seconds = timeInfo.tm_sec;
    M5.Rtc.SetTime(&TimeStruct);

    RTC_DateTypeDef DateStruct;
    DateStruct.WeekDay = timeInfo.tm_wday;
    DateStruct.Month = timeInfo.tm_mon + 1;
    DateStruct.Date = timeInfo.tm_mday;
    DateStruct.Year = timeInfo.tm_year + 1900;
    M5.Rtc.SetData(&DateStruct);
    Serial.println("Time now matching NTP");
    M5.Lcd.fillScreen(BLACK);
    M5.Lcd.setCursor(20, 15);
```

```
      M5.Lcd.setCursor(20, 10);
      M5.Lcd.println("S Y N C");
      delay(500);
      M5.Lcd.fillScreen(BLACK);
      M5.Lcd.setCursor(40, 0, 2);
      M5.Lcd.println("Simple Clock");
    }
}


#define LED_BUILTIN 10

void setup() {
  // put your setup code here, to run once:
  pinMode(LED_BUILTIN, OUTPUT);
  M5.begin();
  Serial.begin(115200);
  delay(100);

  timeSync();

  WiFi.mode(WIFI_STA);

  Wire.begin(0,26);
  M5.Lcd.setRotation(3);
  M5.Lcd.fillScreen(BLACK);
  M5.Lcd.setCursor(0, 0, 2);
  M5.Lcd.println("Todays Hike");
  pinMode(M5_BUTTON_HOME, INPUT);

  if(bmm.initialize() == BMM150_E_ID_NOT_CONFORM) {
    Serial.println("Chip ID can not read!");
    while(1);
  } else {
```

```
  } else {
    Serial.println("Initialize done!");

  }
  if (!bme.begin(0x76)){
      Serial.println("Could not find a valid BMP280 sensor, check wiring
      while (1);
  }

uint8_t setup_flag = 1;
  WiFi.mode(WIFI_STA);
    ThingSpeak.begin(client);
    M5.IMU.Init();
}
uint8_t setup_flag = 1;


void loop() {
    // put your main code here, to run repeatedly:
M5.update();
    float tmp = dht12.readTemperature();
    float Ftmp = (tmp * 1.8)+32;
    maxTemp = max(maxTemp, Ftmp);

  float hum = dht12.readHumidity();
  M5.Lcd.setCursor(0, 20, 2);
  //M5.Lcd.printf("Temp: %2.1f Humi: %2.0f%%", Ftmp, hum);

  float pressure = bme.readPressure();

  M5.Lcd.setCursor(0, 40);
  //M5.Lcd.printf("pressure: %2.1f", pressure);
  delay(100);
```

```
//IMU code
   M5.IMU.getAccelData(&accX,&accY,&accZ);

  accX = abs(accX);
  accY = abs(accY);
  accZ = abs(accZ);

  maxaccX = max (maxaccX, accX);
  maxaccY = max (maxaccY, accY);
  maxaccZ = max (maxaccZ, accZ);

  accVectorLng = sqrt(sq(accX) + sq(accY));
  maxaccVectorLng = max(maxaccVectorLng, accVectorLng);

      M5.Lcd.setCursor(0, 60, 2);
    M5.Lcd.printf("Steps Taken : %.2f ", steps);
  delay(500);
  if(maxaccVectorLng >= 1){
    steps = steps + 1;
  }
 delay(100);
    maxaccX = 0.0f;
    maxaccY = 0.0f;
    maxaccZ = 0.0f;
    maxaccVectorLng = 0.0f;


  //send data to thingspeak on button press
  if (M5.BtnA.isPressed()){
  // Connect or reconnect to WiFi
  if (WiFi.status() != WL_CONNECTED) {
    //Serial.print("Attempting to connect to SSID: ");
    Serial.println(SECRET_SSID);
```

```
    Serial.println(SECRET_SSID);
    while (WiFi.status() != WL_CONNECTED) {
      WiFi.begin(ssid, pass); // Connect to WPA/WPA2 network. Change thi
     // Serial.print(".");
      delay(5000);
    }
  //  Serial.println("\nConnected.");
  }

  // Measure Signal Strength (RSSI) of Wi-Fi connection
  long rssi = WiFi.RSSI();

  // Write values to Field of a ThingSpeak Channel
  int httpCode = ThingSpeak.writeField(myChannelNumber, 1, steps, myWrit
  delay(15000);
  int httpCode2 = ThingSpeak.writeField(myChannelNumber, 2, maxTemp, myW


  if (httpCode == 200) {
    Serial.println("Channel write successful.");
  }
  else {
    Serial.println("Problem writing to channel. HTTP error code " + Stri
  }

  // Wait 20 seconds to update the channel again
  delay(20000);
}
if (M5.BtnB.isPressed()){
  if (clockDisplayed == false){
      M5.Lcd.fillScreen(BLACK);
      M5.Lcd.setCursor(0, 0, 2);
      M5.Lcd.println("Todays Hike");
      M5.Lcd.setCursor(0, 20, 2);
```

```
      M5.Lcd.setCursor(0, 0, 2);
      M5.Lcd.println("Todays Hike");
      M5.Lcd.setCursor(0, 20, 2);
     M5.Lcd.printf("Temp: %2.1f Humi: %2.0f%%", Ftmp, hum);
       M5.Lcd.setCursor(0, 40);
     M5.Lcd.printf("pressure: %2.1f", pressure);
        M5.Lcd.setCursor(0, 60, 2);
     M5.Lcd.printf("Steps Taken : %.2f ", steps);
  clockDisplayed = true;
  }
  else if(clockDisplayed == true){
     // Printing time to LCD
  M5.Lcd.setRotation(3);
  M5.Lcd.fillScreen(BLACK);
  M5.Lcd.setTextSize(1);
  M5.Lcd.setTextColor(WHITE,BLACK);
  M5.Lcd.setCursor(40, 0, 2);
  M5.Lcd.println("Simple Clock");
    M5.Lcd.setTextSize(2);
    M5.Rtc.GetTime(&RTC_TimeStruct);
    M5.Rtc.GetData(&RTC_DateStruct);
    M5.Lcd.setCursor(27, 15);
    M5.Lcd.printf("%02d:%02d:%02d\n", RTC_TimeStruct.Hours, RTC_Time
    M5.Lcd.setCursor(27, 50);
    M5.Lcd.setTextSize(1);
    M5.Lcd.printf("Date: %04d-%02d-%02d\n", RTC_DateStruct.Year, RTC_
    clockDisplayed = false;
  }
}
if (M5.BtnB.pressedFor(5000)){
steps = 0;
maxTemp = 0;
 }
}
```

4. An explanation of the structure of your program, and an explanation of how your program works

I combined the IMU sensor code, the ENV sensor code, and the sample watch code to make this device work. I start by setting these up and syncing the clock using wifi. The set up takes care of the initial display. In the loop I update the data for the hike being shown to the user and built in some button functions. Button A connects to thingspeak and transfers data. Button B changes the screen display from environmental to a watch, or If you hold the button resets your steps and max temp to 0 for the next hike.

5. Any calculations that you completed during the lab

I only had to calculate the Celsius to Fahrenheit