ECE40862: Software for Embedded Systems

Fall 2020

Lab 6– Uploading Data to Google Sheets using IFTTT

Due by 11:59pm, Tuesday, December 1, 2020.

1. Overview

This is an extension of Lab 5. Here, you will be working with the IFTTT service (If-This-Then-That). You will find more information here https://ifttt.com/.

You should use the setup created in Lab 5.

1.1.Overall Application Workflow

In addition to the tasks in Lab 5, you need to perform the following tasks for this extension:

- At every minute after switch 2 has been pressed, you program should upload all the sensor readings along with a SessionID to **Google Sheet** using the **IFTTT** service, i.e. it should upload *SessionID, Velocity X, Velocity Y, Velocity Z, Pitch, Roll, Theta, Temperature*. You can use your own Google Account or a dummy account for this. (Use a random 6-digit number for *SessionID*. Increment this by 1 every time you upload. You do not have to generate a random number. You can use a hardcoded 6-digit value for the initial *SessionID*)

2. Exercises

The first step is to setup an account in IFTTT. You can easily do this by signing up at www.ifttt.com. After creating an account, go to 'Create' and click on 'Add'. Select 'Webhooks' from the list of services.

NOTE: You must find out how you can use Webhooks to receive data from ESP32. You can use the following tutorial as a guidance. Make sure to keep things simple!

3. Submission

Make sure you follow these instructions precisely. Points will be deducted for any deviations. You need to turn in your code on Brightspace. Please create a zipfile named ***username*_lab6**,

where username is your CAREER account login ID. *This zipfile should contain only the following files, i.e., no executables, no temporary files, etc.*

1. ***spinner.py:*** Your program for Lab 6 (Edit the code from Lab5. You can make improvements to the previous code if you want)
2. ***myDecisions.txt***: A brief description of what you did and why in this text file

Zip the files and name it as *username*_lab6.zip and **upload the .zip file to Brightspace**.

NOTE: Follow the lab document strictly when using different peripherals/modules/packages. Points will be deducted if you fail to follow the lab instructions. If anything is NOT mentioned explicitly, you can use package/module to write your program.

REFERENCES

[1] Getting started with MicroPython on the ESP32 https://docs.micropython.org/en/latest/esp32/tutorial/intro.html

[2] ESP32 WROOM-32 Datasheet https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf

[3] ESP32 Technical Reference Manual https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf

[4] Adafruit HUZZAH32 – ESP32 Feather Online Manual https://learn.adafruit.com/adafruit-huzzah32-esp32-feather

[5] Adafruit ESP32 Feather Schematics https://cdn-learn.adafruit.com/assets/assets/000/041/630/original/feather_schem.png?1494449413

[6] MicroPython GitHub https://github.com/micropython/micropython

[7] ESP32 specific functionalities in MicroPython http://docs.micropython.org/en/latest/library/esp32.html

[8] Learn how to talk to I^2C devices with MicroPython: https://learn.adafruit.com/micropython-hardware-i2c-devices/i2c-master

[9] ADXL343 triple-axis Accelerometer: https://www.adafruit.com/product/4097

[10] ADXL343 datasheet: https://www.analog.com/media/en/technical-documentation/data-sheets/ADXL343.pdf

[11] ADT7410 precision Temperature Sensor: https://www.adafruit.com/product/4089

[12] ADT7410 datasheet: https://www.analog.com/media/en/technical-documentation/data-sheets/adt7410.pdf

[13] IFTTT: https://ifttt.com/