ECE40862: Software for Embedded Systems

Fall 2019

Lab 6 Extension – Uploading Data to Google Sheets using IFTTT

To be done by SAME GROUP from LAB 6

Due by 11:59pm, Wednesday, October 30, 2019.

1. Overview

This is an extension of Lab 6. This extension needs to be done by the same group and submitted along with Lab 6. Here, you will be working with the IFTTT service (If-This-Then-That). You will find more information here https://ifttt.com/ as well as the lecture slides.

Your team should use the setups created in Lab 6.

1.1.Overall Application Workflow

The flow of tasks executed by spinner #1 and spinner #2 are already described in Lab 6 doc.

In addition to them, you need to perform the following tasks for this extension:

- When Spinner 1 receives SessionID from Spinner2, in addition to measuring sensor data, encrypting and generating hmac, (refer to Lab 6 doc, Section 1.1, Para 2.a, 2.b, 2.c), it should also upload all the sensor readings along with each SessionID to **Google Sheet** using the **IFTTT** service, i.e. it should upload *Spinner #1 Node ID, SessionID, Accel X, Accel Y, Accel Z, Temperature*. You can use the Team Member #1 Google Account for this.
- Spinner 2 should decrypt all information as mentioned in Lab 6. It should also upload the same data (replace Spinner #1 Node ID by Spinner #2 Node ID) to a different **Google Sheet** using **IFTTT** service. You can use the Team Member #2 Google Account for this.
- A simple comparison (manual comparison is enough) is done between the two sheets to check whether both instances of the data are the same for each SessionID.

2. Programming Exercises

This section describes the additional software configuration which you should perform for the Lab 6 extension.

2.1. Setting up IFTTT

This step is to be followed for each spinner individually as both should upload data to two different google sheets.

- The first step is to setup an account in IFTTT. You can easily do this by signing up at www.ifttt.com
- The next step is to create an 'Applet'. Go to 'My Applets' and click on 'Create' from the Menu.
- **Creating 'this' event**: Click on 'this' (the one in blue). Then select the 'Webhooks' icon and choose the trigger 'Receive a web request'. Name your event and click the 'Create trigger' button.
- **Creating 'that' event**: Now click on 'that'. Search for a service named 'Google sheets' and select it. Select 'Connect'. Choose action as 'Add row to spreadsheet' as you are going to store all the data from the spinners in Google Sheet. Complete all the necessary fields. Finally, select 'Create action'. Click on 'Finish'. Now, you have created applet.
- To send the data to IFTTT, you need to put the data in a JSON object. Make sure you use proper field names to distinguish the separate values.

NOTE: You must find out how you can use Webhooks to receive data from ESP32. You must decide how frequent the data must be sent to Google sheets (IFTTT and Google sheets might have some limitations). Make sure to keep things simple!

2.2. Complete System

Your final demo should include Lab 6 as well as the extension.

3. Submission (Only one Submission for Lab 6 + Extension)

Make sure you follow these instructions precisely. Points will be deducted for any deviations. You need to turn in your code on Blackboard. Please create a directory named ***username*_lab6**, where username is your CAREER account login ID. *This directory should contain only the following files, i.e., no executables, no temporary files, etc.*

1. ***spinner1.py***: your program for Spinner#1. This should include lab 6 code + extension.
2. ***spinner2.py***: your program for Spinner#2. This should include lab 6 code + extension.
3. ***crypt.py***: completed encryption file.

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

4. Grading

20 teams (40 students) will be randomly selected for lab demo, they will be informed on Thursday, 31st October, 10 AM *via your Purdue email id*. If you are selected for the demonstration, you will need to login to any of the computers in the **EE217 lab**, download your submitted source codes from Blackboard, copy them to your ESP32 board *using rshell* and show the TA that your code works correctly. You will also be asked to show the TA your

code listing and answer conceptual questions about your code. Marks are only awarded if you satisfy all requirements.

In addition, automatic evaluation scripts will be executed on your uploaded source codes on Blackboard and you will be graded according to the evaluation results.

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^{2}C 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] Cloud MQTT: https://www.cloudmqtt.com/

[14] AES Wiki: https://en.wikipedia.org/wiki/Advanced_Encryption_Standard

[15] More on AES encryption: https://www.comparitech.com/blog/information-security/what-is-aes-encryption/

[16] HMAC Wiki: https://en.wikipedia.org/wiki/HMAC

[17] More on HMAC: https://www.jscape.com/blog/what-is-hmac-and-how-does-it-secure-file-transfers

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