Lab2 - MediaTek Cloud Sandbox (MCS)

NCTU Introduction to IoT

TA 劉自強, Khoi

October 30, 2019
Outline

• Lab Objective
• Lab Description
Lab Objective

• In this lab, you will learn how to connect Raspberry Pi to MCS

• Learn how to send the HTTP requests using RESTful API
Lab Description – data service platform

- MediaTek Cloud Sandbox (MCS) is a cloud based data service platform for Internet of Things devices.

- There are some alternative platforms: ThingSpeak, AWS IoT, IBM Bluemix, Google Firebase
Lab Description – create an account

Click the follow link to create an account and sign in:

https://mcs.mediatek.com/oauth/en/login
Lab Description – create a prototype

• Development -> Prototype
• Create a new prototype
• Follow the setting on next page
Lab Description – create a prototype

• You can type anything you want at here.
• Since we will connect Raspberry Pi, so here I will choose Raspberry Pi development board.
• Finally, click the save button!
Lab Description – create a data channel

• On the prototype page, add a new data channel
Lab Description – create a data channel

- Choose “Controller” for switches, and choose “Display” for values published by sensor.
Lab Description – create a data channel

- Please remember data channel id, since we will use it in Restful API
Lab Description – create a data channel
Lab Description – create a test device

- Don’t click “Create as public device” if you don’t want to let everyone watch your data.
Lab Description – create a test device

- Development -> Test devices
- Click the Detail and play with Restful API!
Requests library in Python

- A standard for making HTTP request.
- Allow you to use APIs (sending HTTP requests and receiving responses) in Python
Requests - installation

- pip install request
- Use “import request” in your python code.
Requests - using

- request.get(url)
- Request.get will return the server’s response.
- We also can use **params** and headers into the get requests:
  - response = request.get(url, params={'key': 'data'}, headers={'key': 'data'})
Requests - using

- HTTP response status code:
  - Informational responses (100–199)
  - Successful responses (200–299)
  - Redirects (300–399)
  - Client errors (400–499)
  - Server errors (500–599)

- Use `response.json()` to convert the content of response to json format.
- Use `response.json()["key"]` to get data of key.
Reference


https://realpython.com/python-requests/