Public facing github: https://github.com/Tantiv4/giot
Documentation for google and others is distributed from there

Cloud Side Setup

Setting up the Google IOT infra from the Host computer(MAC or Linux machine recommended).
Basic Overview of Google IOT for beginners.

Install the Google Cloud Command Line Tool

Follow all the commands in the link to install google command line tool

Installing the components

```
gcloud components install beta
```

Authenticate with Google Cloud

```
gcloud auth login
```

Create cloud project - choose your unique project name

```
gcloud projects create YOUR_PROJECT_NAME
```

Add permissions for IoT Core

```
gcloud projects add-iam-policy-binding YOUR_PROJECT_NAME
--member=serviceAccount:cloud-iot@system.gserviceaccount.com
--role=roles/pubsub.publisher
```

Set default values for gcloud

```
gcloud config set project YOUR_PROJECT_NAME
```

Create PubSub topic for device data

```
gcloud beta pubsub topics create <iot-topic>
```

Create PubSub subscription for device data

```
gcloud beta pubsub subscriptions create --topic <iot-topic> <iot-subscription>
```

Create device registry

```
gcloud beta iot registries create <iot-registry> --region us-central1
--event-pubsub-topic=<iot-topic>
```

For basic project creation billing needs to be enabled.

Steps to be run on Host machine

Download the below tool to your computer

- For Linux
- For Windows
- For Mac OS

After Downloading the appropriate Tool, please follow the steps:-

Pre requisites

projectid, registryid and regionid(us-central1 in this case) are the parameters which you have already created and used during the cloud setup. Please use the appropriate values for these three parameters to run in this tool.

deviceid is the new device which you are creating using this tool. You can choose any device name. This device will be created under the same projectid, registryid and regionid

Running the tool:

- In linux system, please run the following command:-

```
chmod 777 giot-linux
```

```
./giot-linux --projectid=<> --registryid=<> --regionid=<> --deviceid=<>
```

- In a Mac, please run the following command:-

```
chmod 777 giot-macos
```

```
./giot-macos --projectid=<> --registryid=<> --regionid=<> --deviceid=<>
```

- In a windows machine, please run the following command:-

```
giot-win.exe --projectid=<> --registryid=<> --regionid=<> --deviceid=<>
```

Checking if device is created:

Check if your registry and device is created in google console

Device Side Setup

Environment Setup

Step 1:

Download and install the docker image for your HOST computer from this link

Step 2:

Download the docker container.

Step 3:

Unzip the container and navigate to the folder *"dockerContainerGiot"* in Command Prompt

Step 4:

Build the docker container by executing

```
docker build -t qc4020 .
```

Create a folder in the host machine to be shared with docker(<host-directory>).

Upon successful completion of the build command the docker container can be started by executing

```
docker run -v <host-directory>:/QCOMM4020/Code/ --interactive --tty qc4020
```

-v option is used to setup shared directory for code.

Host-directory -> absolute path to the directory created in the host machine in the previous step.

Eg. *“docker run -v D:/Code:/QCOMM4020/Code/ --interactive --tty qc4020”* for windows.

“docker run -v /Users/xxx/Code/:/QCOMM4020/Code/ --interactive --tty qc4020” for OSX

This mounts the directory *D:/Code or /Users/xxx/Code/* to your working folder */QCOMM4020/Code/* inside the container.

Step 5:

The successful execution of the above command will give you a working shell inside of docker container irrespective of your Host environment.

Build Instructions

Step 1:

If the above steps to setup the docker build environment are successful you will get the shared folder mounted at */QCOMM4020/Code/*

Register and download the SDK from Qualcomm.

Download the Google IOT patch from here.

Move the SDK and Patch file to the shared folder.

Step 2:

Apply the Google IOT patch to Qualcomm standard SDK.

Move the giot_CDB_v1_2.patch to QCA4020xxxxxxx/ directory

example:

```
mv giot_CDB_v1_3.patch QCA4020.OR.1.1_PostCS1/
```

Go to the QCA4020xxxxxxx directory and apply the patch

Example

```
patch -p1 < giot_CDB_v1_3.patch
```

This will apply the Google IOT Patch.

Step 3:

Navigate to /QCOMM4020/Code/QCA4020xxxxxxx/target/quartz/demo/QCLI_demo/build/gcc/

```
make prepare && make
```

This will compile the code and create output folder with flash binaries.

Flashing Instructions

Note: Flashing can currently be done only via Windows Machine.

Step 1:

Download and install Python version 2.7 and set the PATH.

Connect both J6 and J85 of the QC4020 CDB board to HOST Machine using the supplied micro-usb cable.

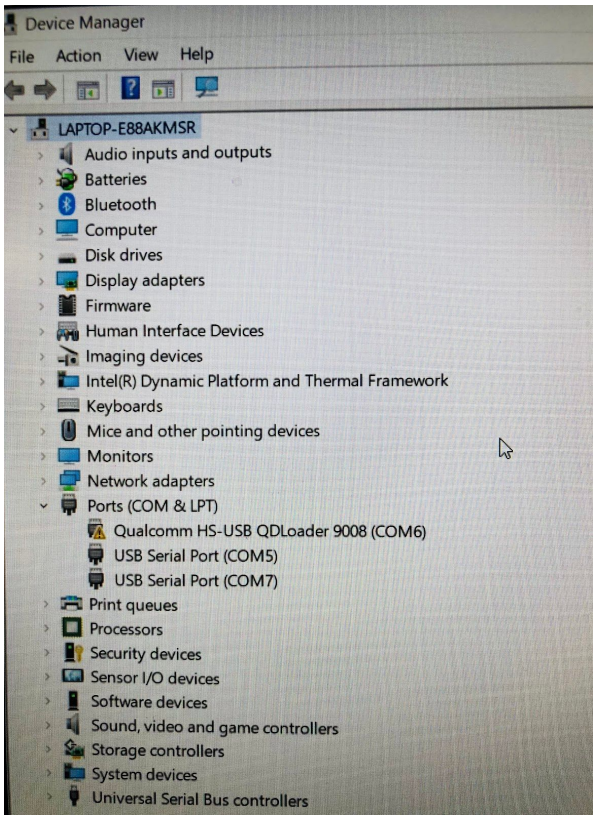
Download and install the driver for Qualcomm UART Interface.

Step 2:

Connect Jumper J32 Pin 1-2 on CDB Board and Power On using switch S7.

Download QDL Driver.

Navigate to Device Manager and Expand the Ports Section.



Right-Click on “Qualcomm HS-USB QDLoader 9008” and select “Update Driver”

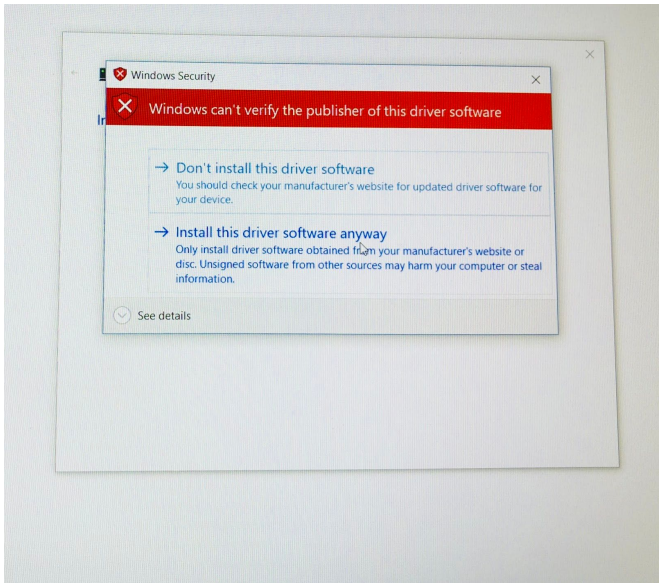
Select 2nd Option “Browse my computer for driver software”

Select “Let me pick up from the list of available drivers on my computer”

Click on “Have Disk” option and select qscer after navigating to Qualcomm_Drivers_QDLoader/Qualcomm Drivers/Qualcomm/Driver/ folder.

Click Next

On the next dialog select “Install this driver software anyway”



This will install the driver for Qualcomm QDLoader.

Note the COM Port number of QDL Loader.

Step 3:

Note: Use Windows Command Prompt only.

Run the following commands:

set FS2IMG=KEEP

python ../../../../../build/tools/flash/qflash.py --comm 8

Once the prompt shows “Flash programming complete!” remove the jumper from j34 and power cycle the CDB board.

Running GIOT on CDB 4020

Connect the EVK to the Host computer via serial cable

Open terminal 115200-8-N-1

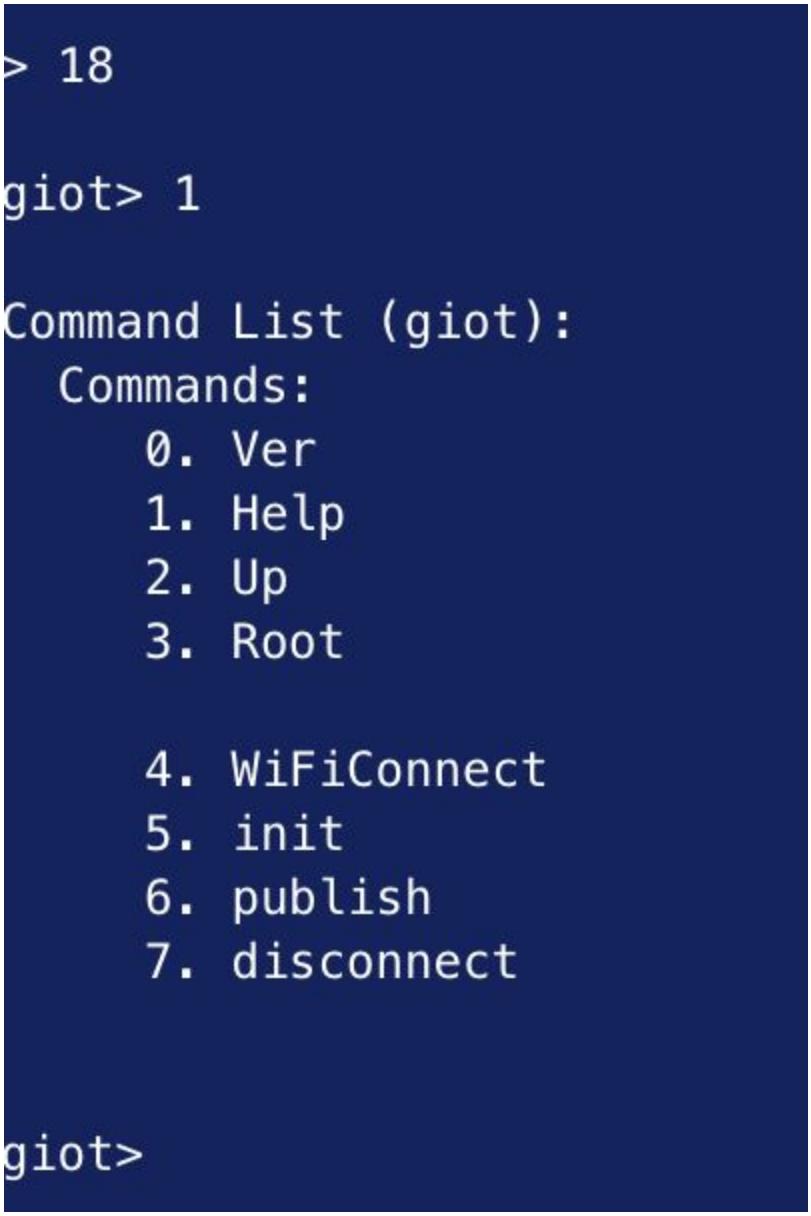
At the terminal prompt use command ‘1’ to see the available CLI options.

```
> 1

Command List:
  Commands:
     0. Ver
     1. Help
     2. Exit

  Subgroups:
     3. BLE
     4. HMI
     5. WLAN
     6. Net
     7. Coex
     8. FwUp
     9. LP
    10. Fs
    11. Ecosystem
    12. SecureFs
    13. Crypto
    14. ZigBee
    15. Thread
    16. Platform
    17. JSON
    18. giot
```

- Select option 18 to go to giot sub menu. Select 1 to see the options



Connecting to Network

- Use option 4 to connect to network
- Usage:

`4 <SSID> <Passphrase>`

- Passphrase can be left blank in case of a open network
- The device will connect to the network and get IP

GIOT initialization

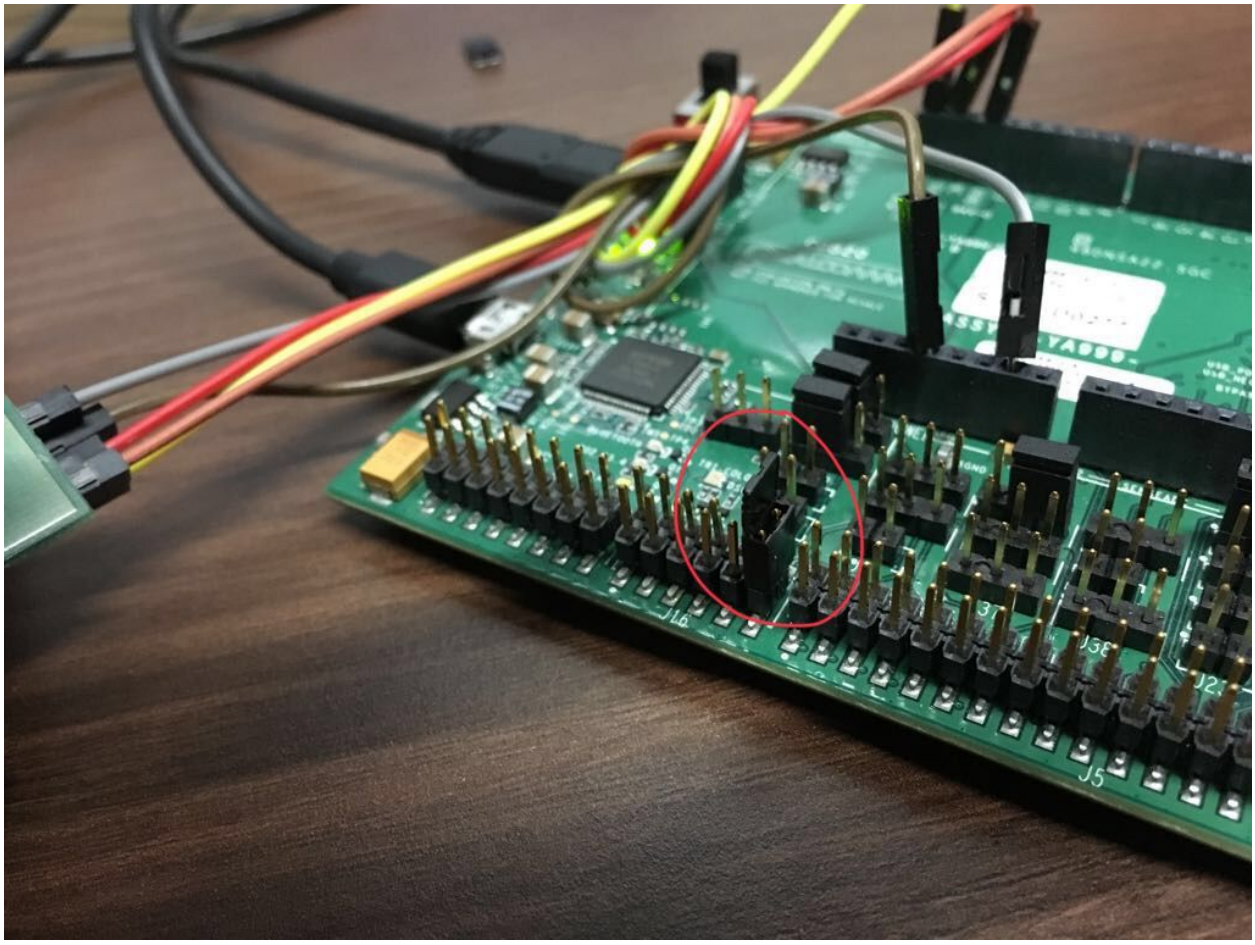
- Inside the GIOT menu
- init
    - init -p <project_id> -r <registry_id> -d <device_ID> - R <region ID>
    - Make sure the project id, registry id, device id and region id are the same as already used for device creation
- Upon successful Connection ,a message “Welcome to 4020” will be sent as a device state.
- In case of error(SSL error), disconnect from GIOT and connect again
- To disconnect use option 7 under GIOT commands
- To check the state go to Google IOT console.
    - Select the appropriate registry
    - Select the device from the devices list
    - Go to “Configuration and State History”

Sending Message from Cloud to Device

- To send a message to device use UPDATE CONFIG option in Google IOT Console. Any text message can be sent.
- The message can be seen in the serial console of the device

Configuring LED from IOT console

- LED connections for QC4020 CDB
    - Use the below jumper setting in J16



- To turn on the LED, send the message “Turn ON LED” from the UPDATE CONFIG in Google IOT console
- To turn off the LED, send the message “Turn OFF LED”

Sending Message from Device to Cloud

- To send message from Device to Cloud:
    - publish -m < message>-t <events/state>
    - Message:- Is the message that will be sent
    - -t is optional. Default is state. ‘-t’ will send the telemetry data
    - Example to publish state:

      publish -m “Hello world”

Section on writing the user code missing<>