
TrustFLEX Step by Step Guide

Google Cloud Platform Connect

Table of Contents

1	Introduction	3
1.1	Getting started with Jupyter Notebook Tutorials	3
1.1.1	Starting Jupyter Notebook.....	3
1.2	Jupyter Notebook Basics	3
1.2.1	The Notebook dashboard	3
1.3	Introduction to Jupyter Notebook GUI	4
2	Jupyter Notebook Tutorials	6
3	Generate Manifest files.....	7
3.1	TrustFLEX – Manifest file generation	7
3.2	Trust&GO – Manifest file generation.....	12
4	Use Case Prototyping	14
4.1	Running GCP example on Jupyter Notebook	14
4.2	Running GCP example on Embedded platform	20
4.2.1	Atmel Studio:	20
4.2.2	MPLAB:	23
4.3	CryptoAuth Trust Platform Factory reset.....	26
5	FAQ.....	28

1 Introduction

This document gives a detailed walk through of connecting securely to Google Cloud Platform. If familiar with Jupyter Notebook, can skip this section and move to Section 2.

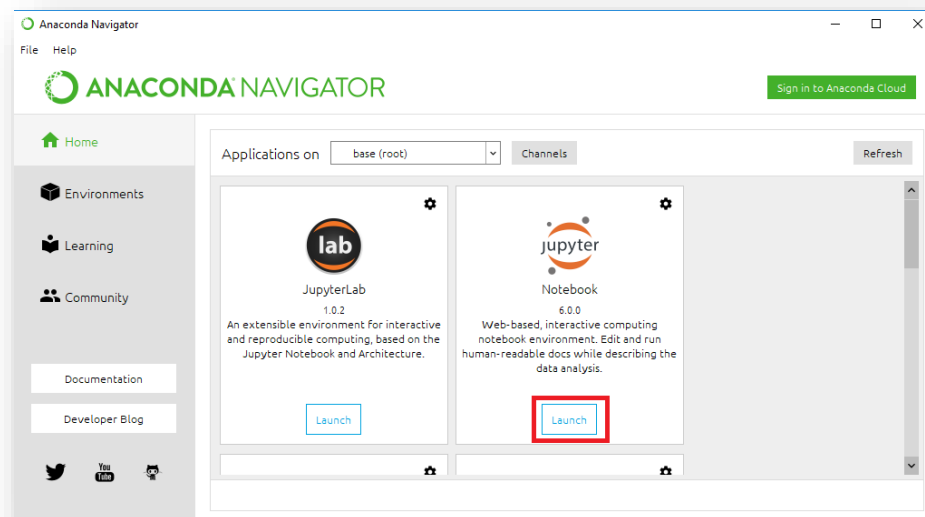
1.1 Getting started with Jupyter Notebook Tutorials

Jupyter Notebook is open source web application which allows you to create documents that contain code that you can execute in place as well as narrative text. It provides GUI elements, ability to execute code in place, ability to add images and gives it the look and feel that normal code files lack.

Jupyter notebooks are mainly used to explain/evaluate code in an interactive way.

1.1.1 Starting Jupyter Notebook

Jupyter notebook can be launched from the Anaconda Navigator main window.



1.2 Jupyter Notebook Basics

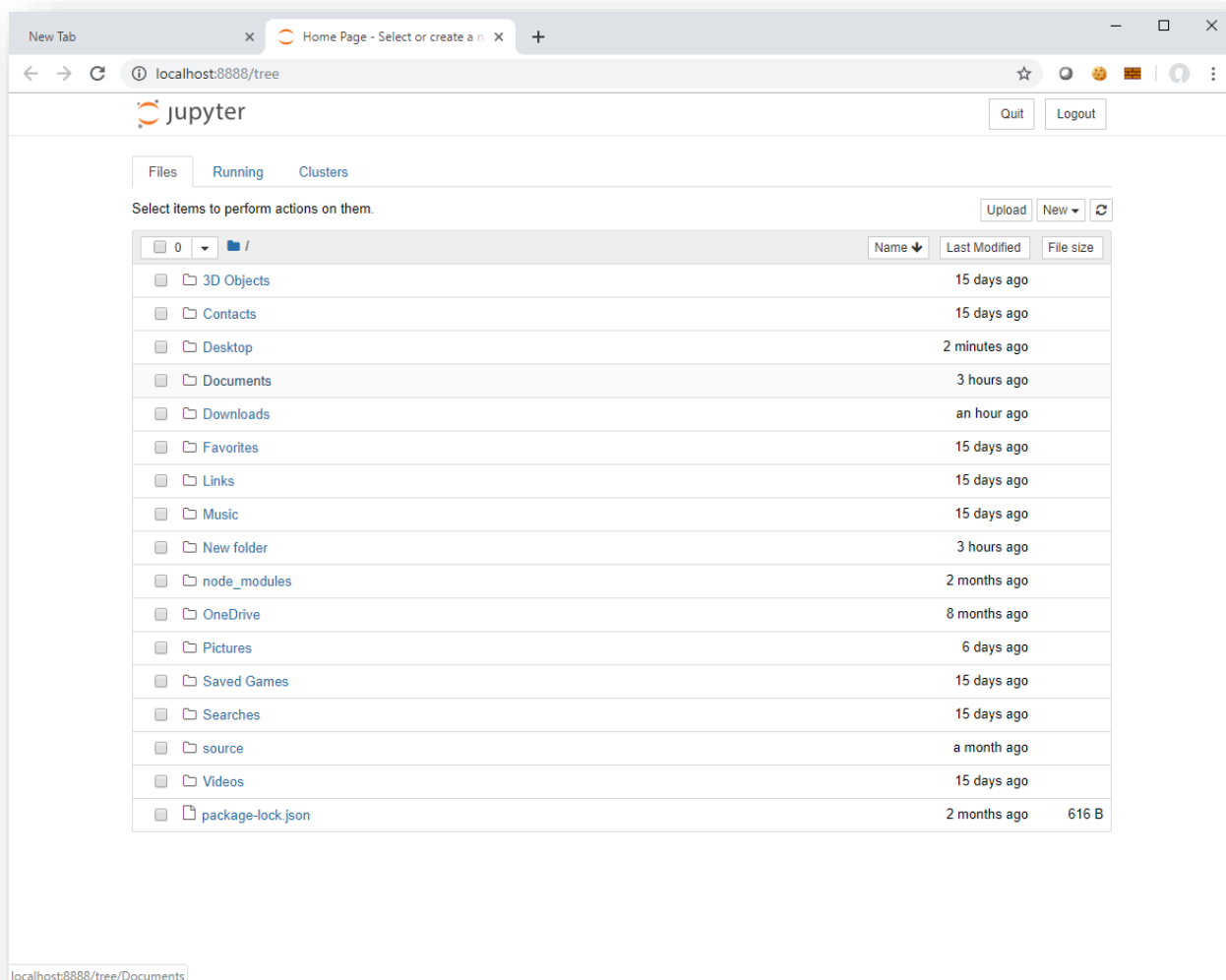
It is recommended to become familiar with Jupyter basic concepts with the online documentation, <https://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/Notebook%20Basics.html>

Some of the content is duplicated here for convenience. The online documentation should always be used as a reference.

1.2.1 The Notebook dashboard

When you first start the notebook server, your browser will open Notebook dashboard. The dashboard serves as a home page for the notebook. Its main purpose is to display the notebooks and files in the current directory.

For example, here is a screenshot of the Jupyter dashboard. The top of the notebook list displays clickable breadcrumbs of the current directory. By clicking on these breadcrumbs or sub-directories in the notebook list, you can navigate your file system.



1.3 Introduction to Jupyter Notebook GUI.

Jupyter Notebooks contain cells where you can either write code or markdown text. Notebooks contain multiple cells, some set as code and others markdown. Code cells contain code that can be executed live, and markdown contains text and images that explains the code.

Below image shows some options in a typical Jupyter Notebook. Individual cells can be executed by pressing on the RUN button as shown in the below image.

All cells in the Notebook can be executed in order by **Kernel->Restart & Run All**.



To run all cells in sequence.



2 Jupyter Notebook Tutorials

The TrustPlatform Design Suite comes with several Notebook Tutorials to easily prototype popular use cases for TrustFLEX and Trust&Go devices. Here is the list of Jupyter Notebook Tutorials.

Jupyter Notebook Tutorials	Relative Path	Applicable devices
Manifest Generation	TNGTLS_Manifest_Generation\notebooks\TNGTLS Manifest File Generation.ipynb	Trust&GO
Resource Generation	TFLXTLS_resource_generation\Crypto Resource Generator.ipynb	TrustFLEX
Accessory Authentication	TFLXTLS_Use_Cases\notebooks\accessory-authentication\Accessory Authentication.ipynb	TrustFLEX
AWS Custom PKI	TFLXTLS_Use_Cases\notebooks\aws-iot\aws-iot with ECC608A-TLFXTLS.ipynb	TrustFLEX
Firmware Validation	TFLXTLS_Use_Cases\notebooks\firmware-validation\Firmware Validation with ECC608A-TFLXTLS Tutorial.ipynb	TrustFLEX
IP Protection	TFLXTLS_Use_Cases\notebooks\ipprotection\IP Protection with ECC608A-TFLXTLS Tutorial.ipynb	TrustFLEX
Secure Public Key Rotation	TFLXTLS_Use_Cases\notebooks\public-key-rotation\Public Key Rotation with ECC608A-TFLXTLS Tutorial.ipynb	TrustFLEX
GCP Connect	TFLXTLS_Use_Cases\notebooks\gcp-iot\gcp-iot with ECC608A-TLFXTLS.ipynb	TrustFLEX

3 Generate Manifest files

In the real scenarios, the Manifest files for Trust&GO and TrustFLEX should be downloaded from microchipDirect. Once devices have shipped, you will be able to download the Manifest file from your Microchip Purchasing & Client Services Account. The file can then be uploaded into your cloud service account.

Kits, demonstration boards do not ship with a Manifest file.

The following sections provide steps to generate manifest files for Trust&GO and TrustFLEX devices during prototyping the Usecases.

Note: Before executing the cells on Crypto Trust Platform, its required to have factory default program running on SAMD21 of Trust Platform. Refer to [4.3 CryptoAuth TrustPlatform Factory reset](#) section for reloading default program.

3.1 TrustFLEX – Manifest file generation

TFLXTLS device is one of the three devices available in the Trust Platform USB Dongle Board.

TrustFLEX devices come pre-programmed with certificates in slots 10, 11 and 12, also slots 0-4 have pre-generated private keys, other than the mentioned slots all the other slots have no meaningful data in them.

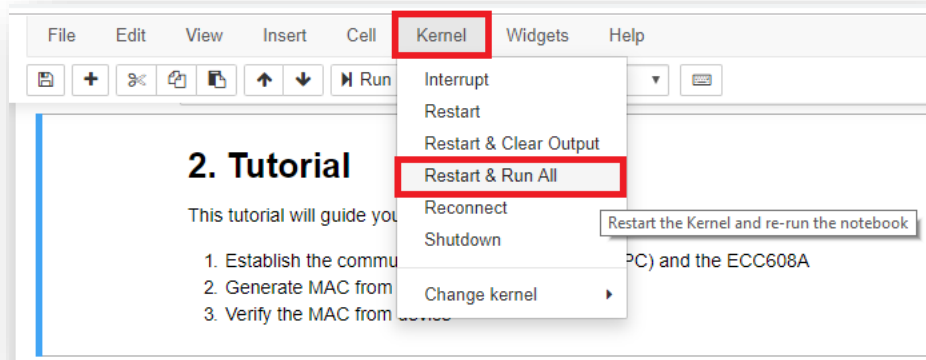
The Resource Generator Notebook will create development keys and certificates for all slots that can be further customized. Keys and Certificate chains are stored in the PC filesystem. These keys should never be used for production purposes as their generation is not handled in a secure environment. These development keys will be later used by the other notebooks to implement the various pre-defined use cases.

By default, Jupyter starts in Users directory (\$HOME for MacOS or Linux systems). For the remainder of this document, it will be assumed that the trust_platform folder is contained in Users directory. If this is not the case, please move trust_platform folder to your Users directory

Within the Jupyter Dashboard, navigate trust_platform\DesignTools\TFLXTLS_Resource_Generation folder to open Crypto Resource Generator.ipynb notebook



Run all cells of the Crypto Resource Generator Notebook: Kernel->Restart & Run All



It will execute and prompt you to choose between MCHP certificate and a custom certificate chain, enter '1' (default) and press Enter key.

The Notebook will generate several keys and certificates. Make sure you have an error free output before continuing to the next steps of the training.

The output log should resemble this:

Choose certificate type

1 - MCHP standard certificate

2 - Custom certificate

Enter certificate type

1

MCHP certificate related files are pre-generated

MCHP certificates found in the device

Backing up certificates from device

Backing up certificates from device - Success

TFLXTLS Root Certificate:

Crypto Authentication Root CA 002

-----BEGIN CERTIFICATE-----

MIIB8TCCA ZegAwIBAgIQd9NtIW7IrmIF5Y46y5hagTAKBggqhkJOPQQDAjBPMSEw
HwYDVQQKDBhNaWNyb2NoaXAgVGVjaG5vbG9neSBjb2MxKjAoBgNVBAMMIUNyeXB0

byBBdXRozW50aWNhdGlvbiBSb290IENBIDAwMjAgFw0xODExMDgxOTEyMTIaGA8y
MDU4MTEwODE5MTIxOVowTzEhMB8GA1UECgwYTWljcm9jaGlwIFRIY2hub2xvZ3kg
SW5jMSowKAYDVQDDCFDcnlwdG8gQXV0aGVudGljYXRpb24gUm9vdCBDQSAwMDIw
WTATBgcqhkJOPQIBBggqhkjOPQMBBwNCAAS9VOZt44dUABrU64VgNUKOGnnit9V
eNhc4tVN1bgwKWv/3W5vclb72Z7xoRaxHTOtSRA6oYWHODz65DfhnWNOo1MwUTAd
BgNVHQ4EFgQUeu19bca3eJ2yOAGI6EqMsKQOKowwHwYDVR0jBBgwFoAUeu19bca3
eJ2yOAGI6EqMsKQOKowwHwYDVR0TAQH/BAUwAwEB/zAKBggqhkjOPQQDAgNIADBF
AiEAodxjRZDsgZ7h3luBEmVRrdTCxPjllSgu4EvnaOx8AnMCID5rp06eTArWjCSw
+y7nk9LmvpRlyhXQ6lvIf1V5mVyt
-----END CERTIFICATE-----

TFLXTLS Root Public Key:

-----BEGIN PUBLIC KEY-----

MFkwEwYHKOZIZj0CAQYIKoZIZj0DAQcDQgAEvVTmbeOHVIAA1OuFYDVCqBp54rf
VXjYXOLVTdW4MClr/91ub3JW+9me8aEWsR0zrUkQOqGFhznC+uQ34Z1jTg==

-----END PUBLIC KEY-----

Validate Root Certificate:

OK

TFLXTLS Signer Certificate:

Crypto Authentication Signer F630

-----BEGIN CERTIFICATE-----

MIIcBTCCAaaggAwIBAgIQRcJ1QIEEuMwF/Gi5b8JPqjAKBggqhkjOPQQDAjBPMSEw
HwYDVQKDBhNaWNyb2NoaXAgaGVudGljYXRpb24gU2lnbmVvIEY2MzAw
byBBdXRozW50aWNhdGlvbiBSb290IENBIDAwMjAgFw0xODEyMTQxOTAwMDBaGA8y
MDQ5MTIxNDE5MDAwMFowTzEhMB8GA1UECgwYTWljcm9jaGlwIFRIY2hub2xvZ3kg
SW5jMSowKAYDVQDDCFDcnlwdG8gQXV0aGVudGljYXRpb24gU2lnbmVvIEY2MzAw
WTATBgcqhkJOPQIBBggqhkjOPQMBBwNCAAR8oS1bpUPChuYczcWZTuxiSzPpkzes
7M4I2mewk6bwN/1p1NR0P/JFIKrOfcGuSKh0HUhd5c0PKdpGzA9jz2+Fo2YwZDAO
BgNVHQ8BAf8EBAMCAYYwEgYDVR0TAQH/BAgwBgEB/wIBADAdBgNVHQ4EFgQUUuHOL
JSono8MbKMMSEhWEP9UsE0QwHwYDVR0jBBgwFoAUeu19bca3eJ2yOAGI6EqMsKQO
KowwCgYIKoZIZj0EAWIDSQAwwRgIhAL9CT/kQFEpVZnSdjZKvE59yrkKh9GwIYerh
EmFrKTIMAIEA4BdsNsIGQwSFAecmt1vRqWWPP7DxVssCevcwOgTbE7Y=
-----END CERTIFICATE-----

TFLXTLS Signer Public Key:

-----BEGIN PUBLIC KEY-----

MFkwEwYHKOZIZj0CAQYIKoZIZj0DAQcDQgAEfKEtW6VDwobmHM3FmU7sYksZ8ZM3
rOzOCNpnsJOm8Df9adTUd/yRSCqzhXBrkiodB1IXeXNDynaRswPY89vhQ==

-----END PUBLIC KEY-----

Validate Signer Certificate:

OK

TNG Device Certificate:

0123B667EC7AC4DA01 ATECC

-----BEGIN CERTIFICATE-----

MIIcBTCCAaaggAwIBAgIQV94yUCHK+/wNLP3DIVdzBDAKBggqhkjOPQQDAjBPMSEw
HwYDVQKDBhNaWNyb2NoaXAgaGVudGljYXRpb24gU2lnbmVvIEY2MzAw
byBBdXRozW50aWNhdGlvbiBTaWduZXIgaGVudGljYXRpb24gU2lnbmVvIEY2MzAw
MDQ3MDczMTAwMDAwMFowRjEhMB8GA1UECgwYTWljcm9jaGlwIFRIY2hub2xvZ3kg
SW5jMSEwHwYDVQDDBgwMTIzQjY2N0VDN0FDNERBMDEgQVRFRQ0MwWTATBgcqhkJOPQIBBggqhkjOPQMBBwNCAASsxZ5jWD6Jf0GJiAnti875Gr6thjscngIJjb2JwIXW

```
Z8CM9FUKcV7cQudAjWfeRn2vmXfZUicUXiLX6ZpeTpDgo2AwXjAMBgNVHRMBAf8E
AjAAMA4GA1UdDwEB/wQEAwIDiDAdBgNVHQ4EFgQUjtfazq7y8JxjZOL8XH00O1Aj
XwIwHwYDVR0jBBgwFoAUUhOLJSono8MbkMMSEhwEP9UsE0QwCgYIKoZIzj0EAwID
SAAwRQIgEXG+riBBipRRJPhdd8AKlj4vFs0q8dswVv3sTk8LIOUCIQcVQYjNb2cR
hRCvAh50ZawONNAxGfdgJMYvyZHK8E2/xg==
-----END CERTIFICATE-----
```

```
TFLXTLS Device Public Key:
-----BEGIN PUBLIC KEY-----
MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAErMWeY1g+iX9BiYgJ7YvO+Rq+rYY7
HJ4CCY29icCF1mfAjPRVCnFe3ELnQI1n3kZ9r5I32VInFF4i1+maXk6Q4A==
-----END PUBLIC KEY-----
```

```
Validate Device Certificate:
OK
```

```
reading slot 0 public key
reading slot 1 public key
reading slot 2 public key
reading slot 3 public key
reading slot 4 public key
```

```
Generated the manifest file 0123b667ec7ac4da01_manifest.json
```

```
-----
```

```
Certificate related steps completed successfully
```

```
-----
```

The Notebook has also generated a manifest file (xxxxxxx_manifest.json) to be uploaded into the public cloud of your choice (Google GCP, AWS IoT and soon to be supported Microsoft Azure).

3.2 Trust&GO – Manifest file generation

Trust&GO device is one of the three devices available in the Trust Platform USB Dongle Board.

Trust&GO devices come with pre-programmed certificates in slots 10, 11 and 12, also slots 0-4 have pre-generated private keys, other than the previously mentioned slots all the other slots are locked.

The secure element manifest format is designed to convey the unique information about a device including its unique ID (e.g. serial number), public keys, and certificates. The manifest file generated can be used to register the device to cloud providers.

By default, Jupyter starts in Users directory (\$HOME for MacOS or Linux systems). For the remainder of this document, it will be assumed that the trust_platform folder is contained in Users directory. If this is not the case, please move trust_platform folder to your Users directory

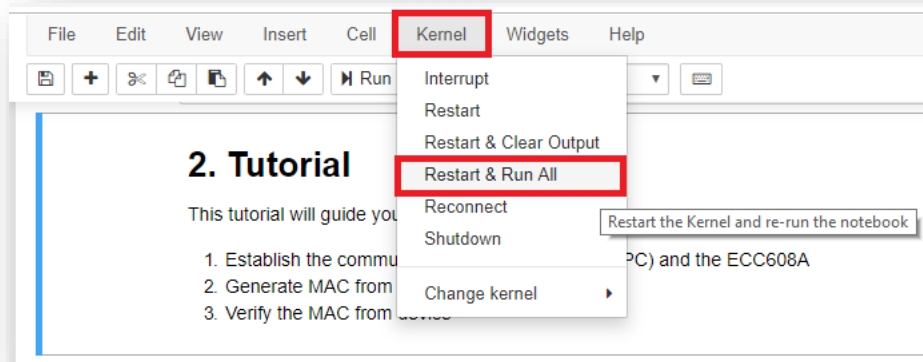
Within the Jupyter Dashboard, navigate trust_platform\DesignTools\TNGTLS_Manifest_Generation\notebooks folder to open TNGTLS Manifest File Generation.ipynb notebook



The screenshot shows the Jupyter Dashboard interface. At the top, there's a header with the Jupyter logo and 'Quit' and 'Logout' buttons. Below the header, there are tabs for 'Files', 'Running', and 'Clusters'. The 'Files' tab is active, showing a file browser. The breadcrumb path is '/ TNGTLS_Manifest_Generation / notebooks'. Below the path, there's a table of files and folders. The file 'TNGTLS Manifest File Generation.ipynb' is highlighted with a red box. Other files listed are 'common.py', 'manifest_generation_helper.py', and 'manifest_signing_helper.py'.

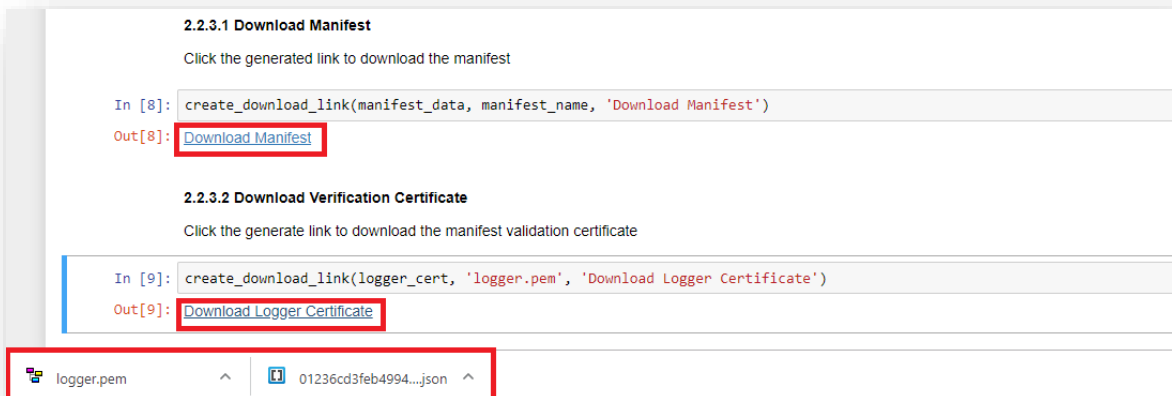
Name	Last Modified	File size
..	seconds ago	
TNGTLS Manifest File Generation.ipynb	4 hours ago	7.4 kB
common.py	a day ago	4.13 kB
manifest_generation_helper.py	a day ago	11.9 kB
manifest_signing_helper.py	a day ago	5.03 kB

Run all cells of the TNGTLS_Manifest_Generation Notebook: Kernel->Restart & Run All



The Notebook will be used to generate a manifest file which can be uploaded into the public cloud provider of your choice (Google GCP, AWS IoT and soon to be supported Microsoft Azure). TNGTLS Manifest Generation notebook needs to be run for all Trust&Go example Notebooks that require a Manifest file.

If all the steps are run without errors, you will see two download links as shown below.



Click on "Download Manifest" and "Download Logger Certificate" to download the manifest and logger file.

4 Use Case Prototyping

This hands-on lab is intended to demonstrate the usage of TrustFLEX/Trust&GO to secure a Google Cloud Platform connection.

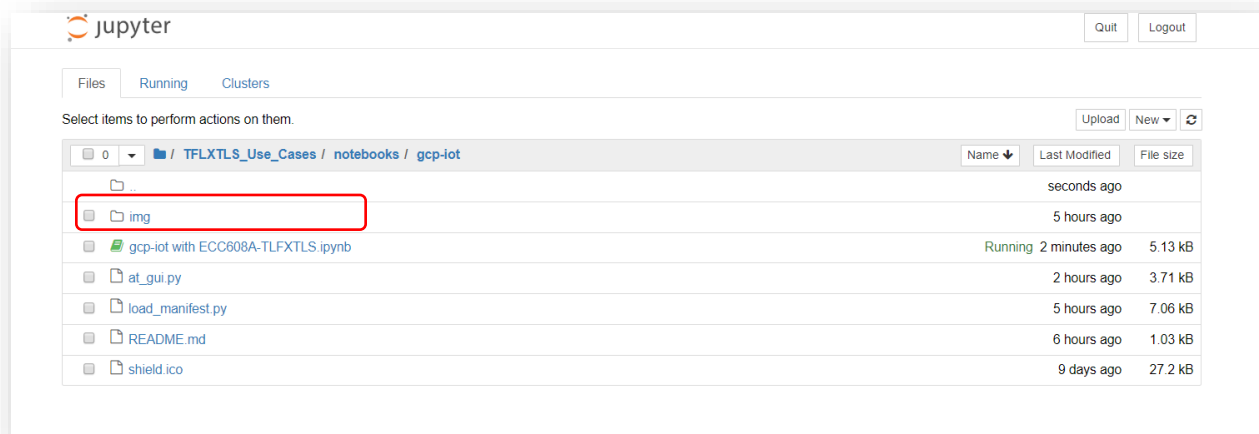
The reference implementation is provided with Embedded projects and Notebooks. The generation of manifest can be achieved through the execution of Jupyter Notebook Tutorials.

Note: It is required to have Google account test account setup prior to running this. Instruction to setup the account is provided in **Docs\TrustFLEX GCP Account setup instructions.pdf**.

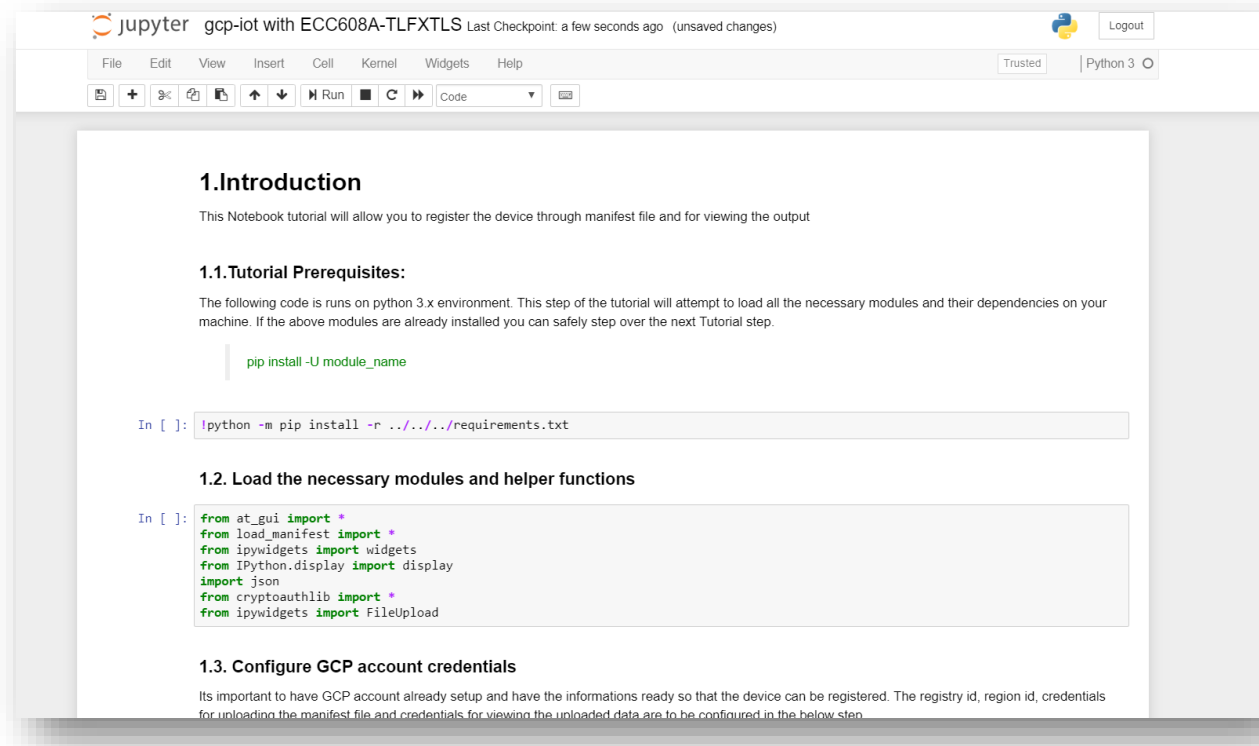
4.1 Running GCP example on Jupyter Notebook

By running this step, one should be able to register the secure element to Google account by uploading device manifest file generated in the previous section. To run this Notebook, its required to have device manifest file (generated in previous section), google account credentials for manifest and data view (saved as part of GCP account setup).

1. From the Jupyter Home page, navigate to **TFLXTLS_Use_Cases\notebooks\gcp-iot\gcp-iot with ECC608A-TLFXTLS.ipynb** notebook file and open it.

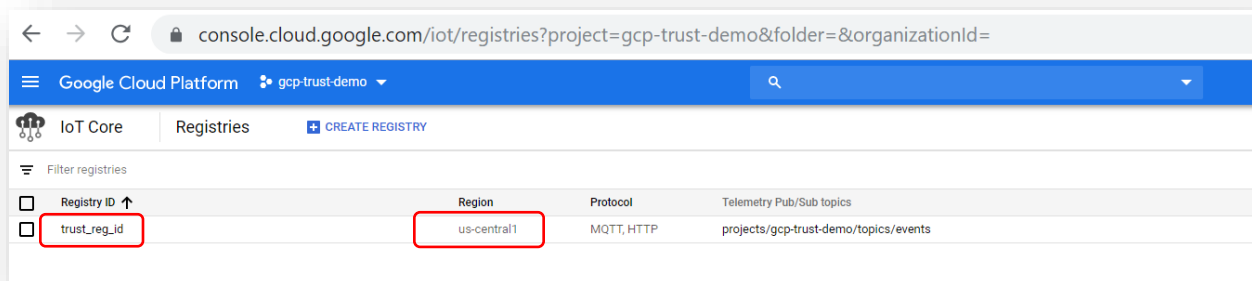


Opening the Jupyter notebook example should load the following on the browser.



2. Before running the Jupyter we need to configure the GCP credentials setup in google cloud.

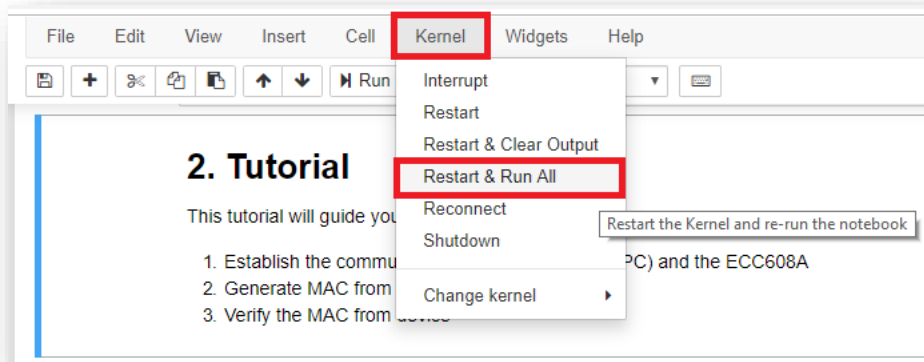
The registry id and region id need to be setup in jupyter notebook. This information can be fetched from GCP account. Search IoT Core in the google cloud console and it will result as in below figure



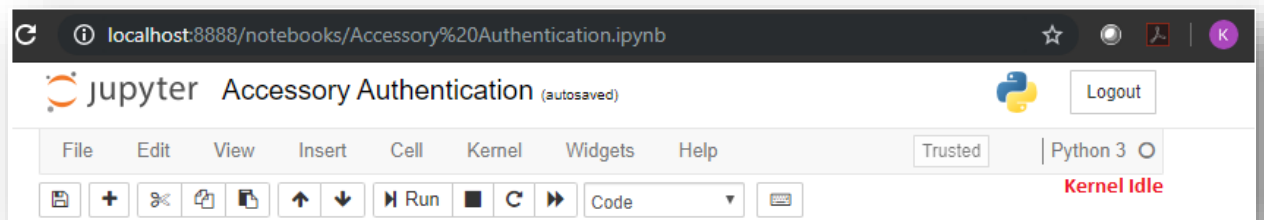
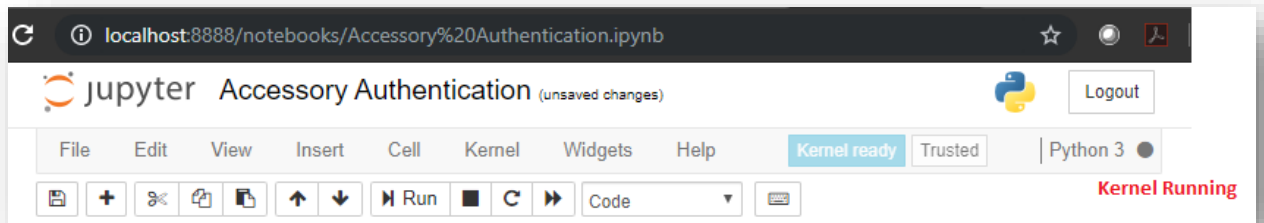
Enter your information's (highlighted above) in the below text. In GCP account setup we would have downloaded two json files, enter the manifest and dataview credentials file names.

```
registry_id = "xxxxxxx"  
region_id = "xxxxxxx"  
manifest_credential_file = '.././../Docs/xxxxxxx.json'  
dataview_credential_file = '.././../Docs/xxxxxxx.json'
```

3. Run All Cells by using Kernel -> Restart & Run All



It may take a while to complete, wait for the kernel to complete all processing i.e. from Kernel Running to Kernel Idle state (Check circle above **RED** text)

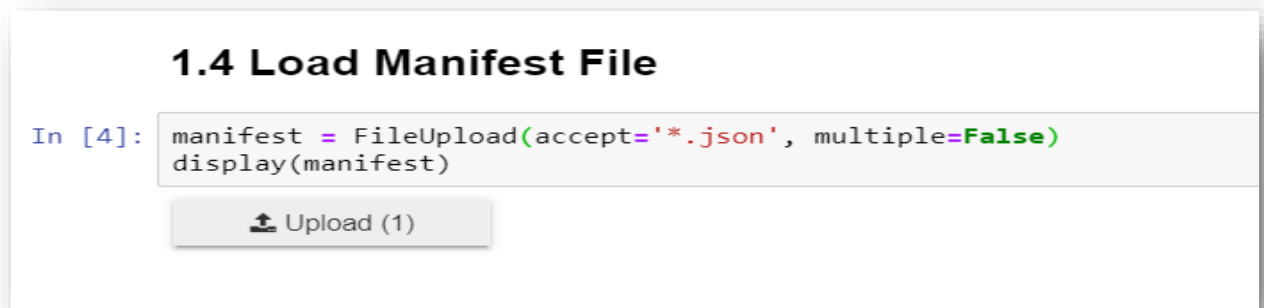


4. Navigate through different cells output for the description of the step and result from the execution.

5. There are 3 major steps:

Load Manifest File:

Under the section **1.4 Load Manifest File**, click the button '**Upload**' and select the manifest file generated in the previous sections.



Load validation certificate:


Under the section **1.5 Load Validation Certificate**, click the button '**Upload**' and select the validation certificate which signed the manifest file and it should be present in the following folder with name log_signer.crt

For TrustFLEX – TFLXTLS_resource_generation\

For Trust&GO - TNGTLS_Manifest_Generation\notebooks\

1.5 Load Validation Certificate

```
In [5]: validator = FileUpload(accept='*.pem', multiple=False)
        display(validator)
```

 Upload (1)

Register device manifest file:

Code block of this step generates “**Upload manifest to**” button. Clicking the button, it registers the device manifest file to the GCP account. Once the manifest file is registered, the gcp cloud authorizes the Trust Platform device and it will be able to communicate to them.

Upon successful execution, the log should look like this.

Upload manifest to ...

GCP GUI

Converting Manifest

Loading Manifest

Device registered succesfully

WARNING: It is required to execute C project successfully before executing the next step in the Jupyter notebook. To execute C project, refer “[Running GCP IoT example on Embedded platform](#)” next section.

GCP GUI:

Code block of this step generates “**GCP GUI**” button. Clicking the button, it will create a very basic graphical interface that will display the trust platform board LED status.

Below screenshot display the graphical interface

Microchip GCP Example		
Project	karthi-demo	
Registry	karthi_regid	
Region	us-central1	
2019-10-10 18:12:09	d01230F56F23B90ED01	Led_Status: ON
2019-10-10 18:12:14	d01230F56F23B90ED01	Led_Status: OFF
2019-10-10 18:12:19	d01230F56F23B90ED01	Led_Status: ON
2019-10-10 18:12:24	d01230F56F23B90ED01	Led_Status: OFF

This GUI displays the packets exchanged between CryptoAuth Trust Platform and GCP.

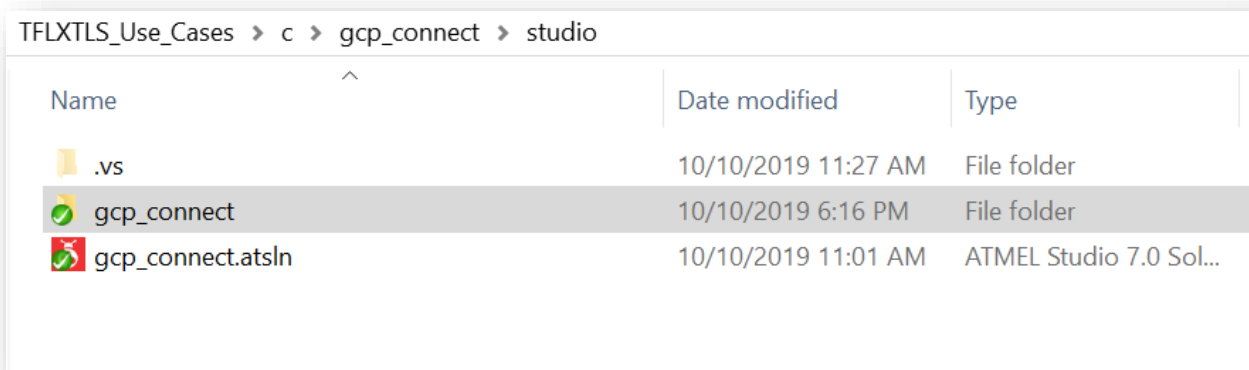
4.2 Running GCP example on Embedded platform

Once the resources are generated and manifest file uploaded to GCP account, both Atmel Studio and MPLAB projects provided can be used to run the use case on CryptoAuth Trust Platform.

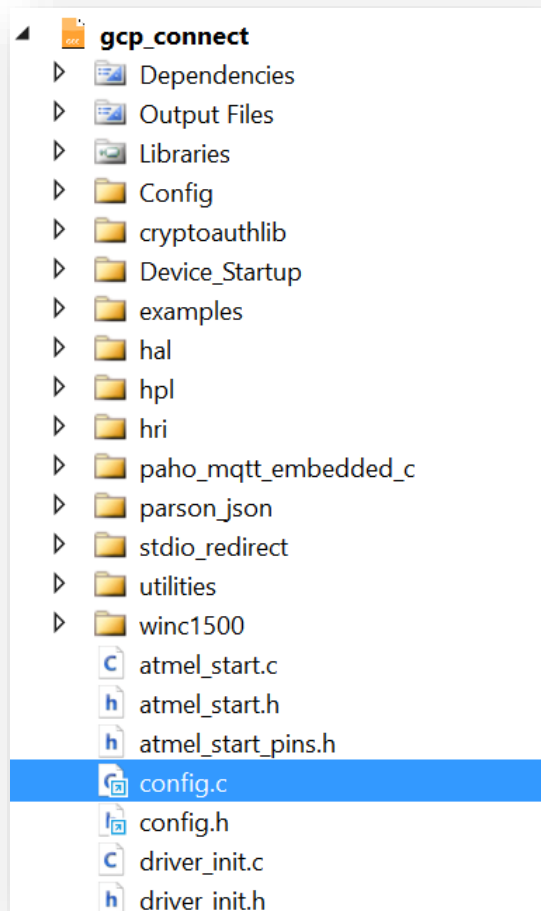
This project establishes a TLS connection and subscribe to MQTT. It is required to use the GCP IoT Jupyter notebook to register the device through manifest file. Prior to executing the application, it is required to update Wifi credentials, GCP account details. Following steps provides the instructions for the same,

4.2.1 Atmel Studio:

1. Open **gcp_connect.atsln** project by navigating
TFLXTLS_Use_Cases\c\gcp_connect\studio\gcp_connect.atsln



2. In the project navigate to **gcp_connect -> config.c** file



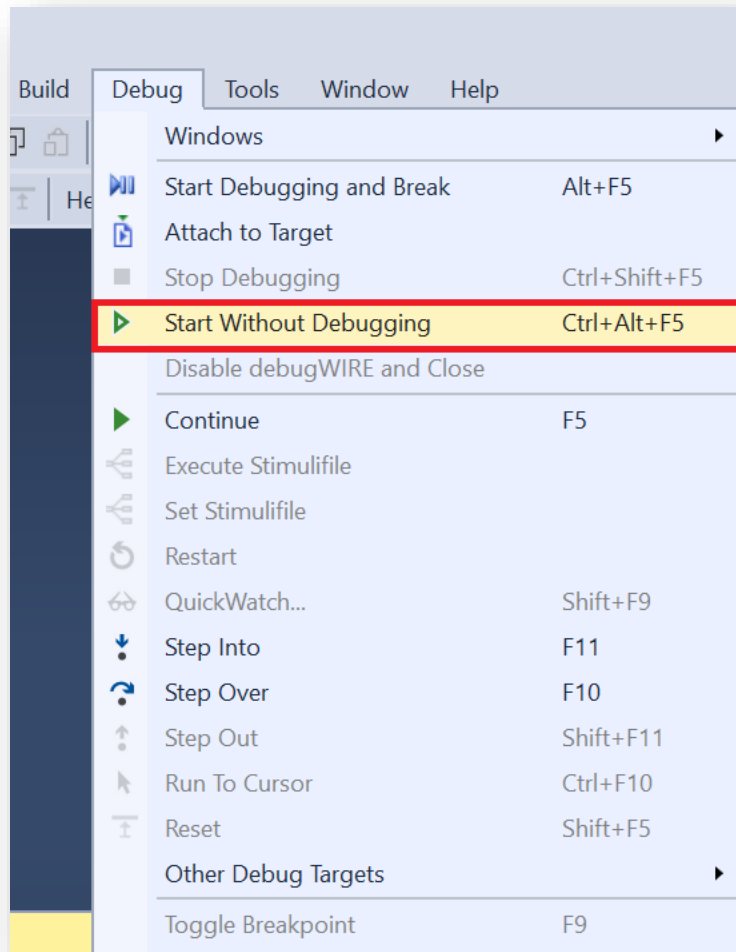
update the following constants before building the project:

- config_demo_ssid
- config_demo_pass
- config_gcp_project_id
- config_gcp_region_id
- config_gcp_registry_id

```
/* Example Configuration Data Global Variables */
const char config_demo_ssid[] = "karthi";
const char config_demo_pass[] = "karthikeyan";

const char config_gcp_project_id[] = "karthi-demo";
const char config_gcp_region_id[] = "us-central1";
const char config_gcp_registry_id[] = "karthi_regid";
```

3. Program the CryptoAuth Trust Platform by navigating to **Debug -> Start Without Debugging**



This step may take some time, wait for Atmel Studio to compile and program the device.

Once the programming is done, reset the hardware (press the reset button) and view the Console messages by using applications like 'Tera Term'. Open the application with the COM related to CryptoAuth Trust Platform with 115200-8-N-1 settings.

```

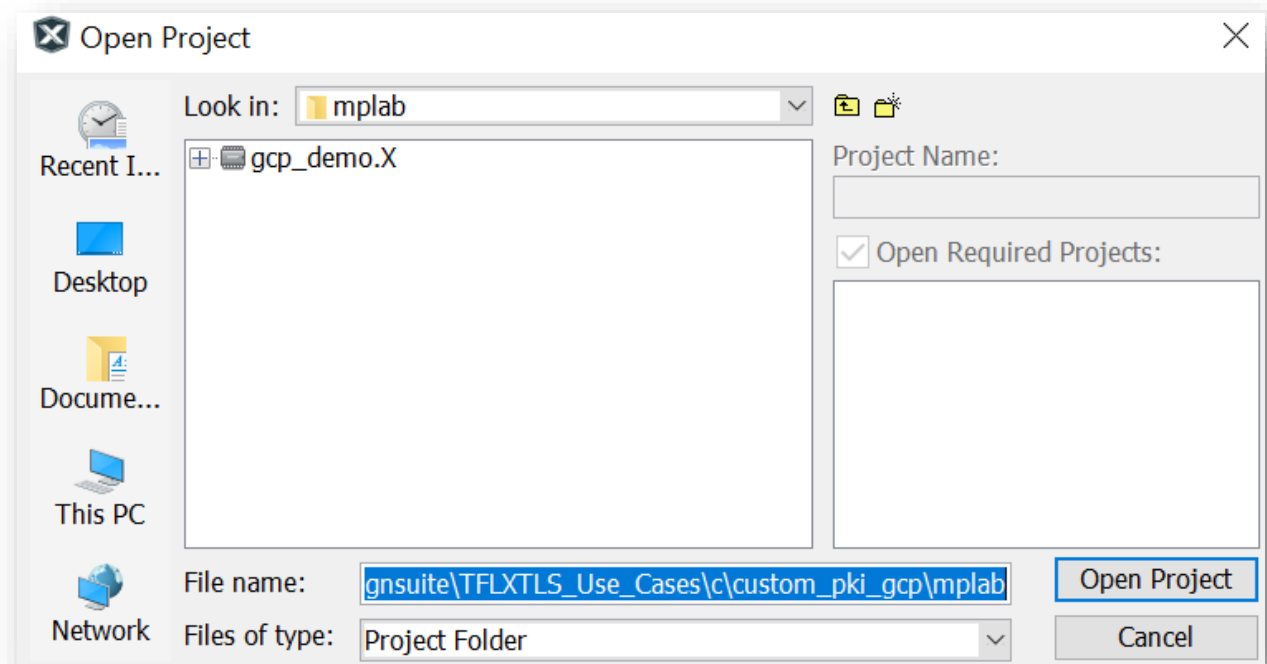
COM35 - Tera Term VT
File Edit Setup Control Window Help
00000000 7B 20 22 74 69 6D 65 73 74 61 6D 70 22 3A 20 31 < "timestamp": 1
00000010 35 37 30 37 31 31 38 33 35 2C 20 22 4C 65 64 5F 570711835, "Led_
00000020 53 74 61 74 75 73 22 3A 20 22 4F 4E 22 7D Status": "ON">
Publishing MQTT Shadow Update Message:
00000000 7B 20 22 74 69 6D 65 73 74 61 6D 70 22 3A 20 31 < "timestamp": 1
00000010 35 37 30 37 31 31 38 34 30 2C 20 22 4C 65 64 5F 570711840, "Led_
00000020 53 74 61 74 75 73 22 3A 20 22 4F 46 46 22 7D Status": "OFF">
Publishing MQTT Shadow Update Message:
00000000 7B 20 22 74 69 6D 65 73 74 61 6D 70 22 3A 20 31 < "timestamp": 1
00000010 35 37 30 37 31 31 38 34 35 2C 20 22 4C 65 64 5F 570711845, "Led_
00000020 53 74 61 74 75 73 22 3A 20 22 4F 4E 22 7D Status": "ON">

```

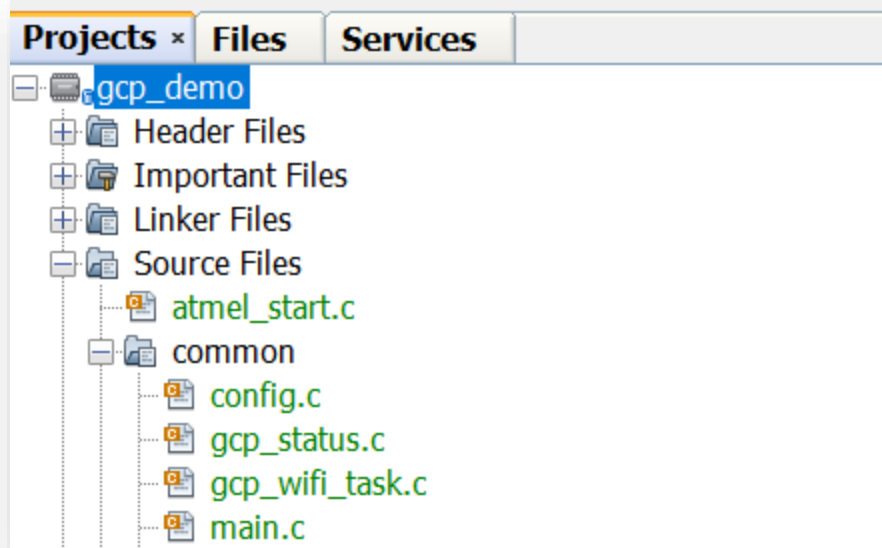
Once successfully programmed the CryptoAuth Trust Platform, navigate to previous section 1.7 to run the [last step \(GCP GUI\)](#) in the Jupyter Notebook.

4.2.2 MPLAB:

1. Open **gcp_demo.X** project by navigating to MPLAB -> File -> Open Project -> **TFLXTLS_Use_Cases\c\gcp_connect\mplab\gcp_demo.X**



2. Open **config.c** file by navigating to **gcp_demo-> Source Files -> common-> config.c**

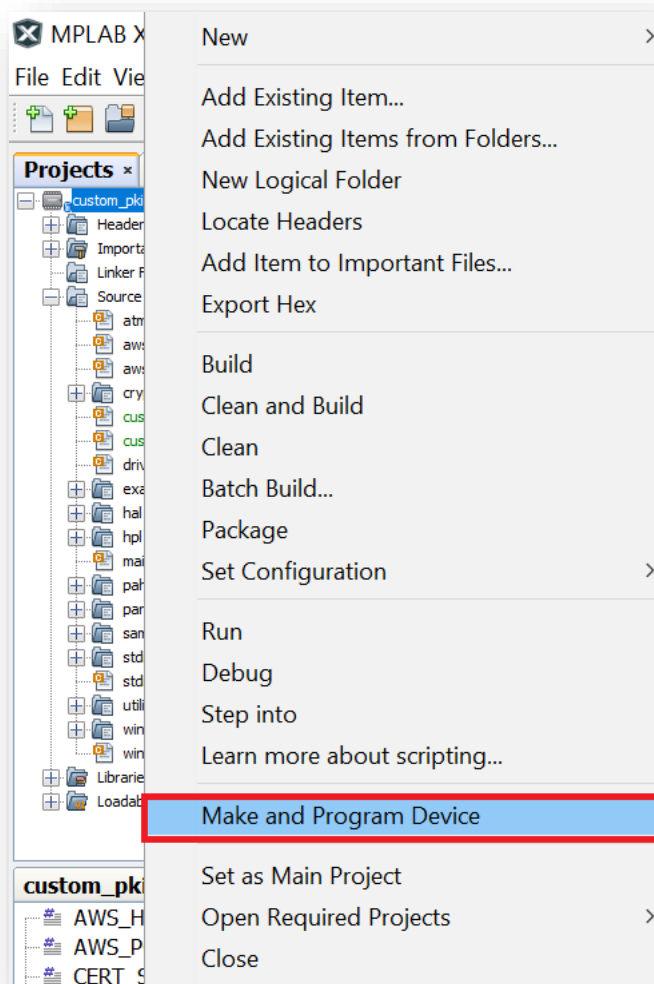


update the following constants before building the project:

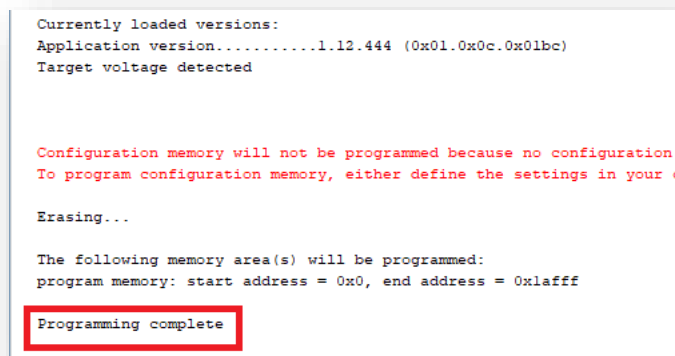
- config_demo_ssid
- config_demo_pass
- config_gcp_project_id
- config_gcp_region_id
- config_gcp_registry_id

```
/* Example Configuration Data Global Variables */  
const char config_demo_ssid[] = "karthi";  
const char config_demo_pass[] = "karthikeyan";  
  
const char config_gcp_project_id[] = "karthi-demo";  
const char config_gcp_region_id[] = "us-central1";  
const char config_gcp_registry_id[] = "karthi_regid";
```

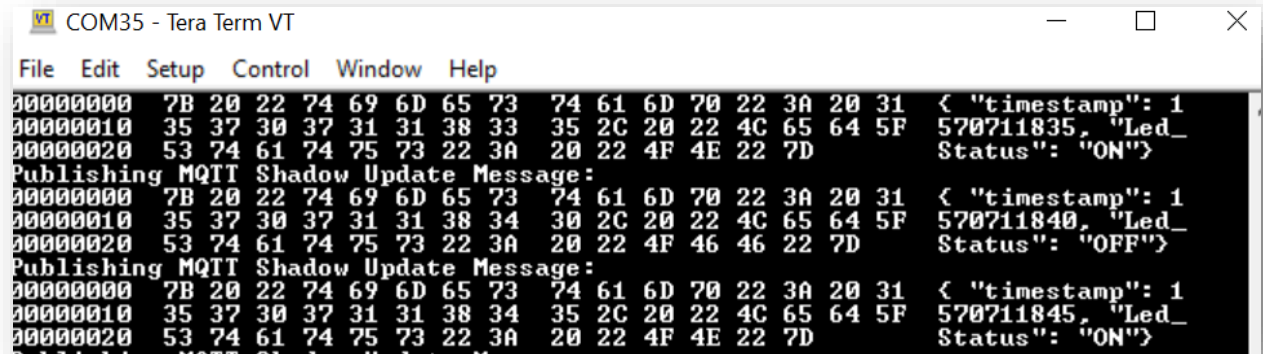
3. Program the CryptoAuth Trust platform by navigating to **gcp_connect** -> **Make and Program Device**



This step may take some time, wait for MPLAB to program the device. Once it is done programming you will see "**Programming complete**" message in Output Window.



Once the programming is done, reset the hardware (press the reset button) and view the Console messages by using applications like 'Tera Term'. Open the application with the COM related to CryptoAuth Trust Platform with 115200-8-N-1 settings.



The screenshot shows a Tera Term window titled 'COM35 - Tera Term VT'. The window displays a series of MQTT Shadow Update messages. Each message is preceded by a hex dump of the raw data. The messages are as follows:

```

00000000 7B 20 22 74 69 6D 65 73 74 61 6D 70 22 3A 20 31 < "timestamp": 1
00000010 35 37 30 37 31 31 38 33 35 2C 20 22 4C 65 64 5F 570711835, "Led_
00000020 53 74 61 74 75 73 22 3A 20 22 4F 4E 22 7D Status": "ON">
Publishing MQTT Shadow Update Message:
00000000 7B 20 22 74 69 6D 65 73 74 61 6D 70 22 3A 20 31 < "timestamp": 1
00000010 35 37 30 37 31 31 38 34 30 2C 20 22 4C 65 64 5F 570711840, "Led_
00000020 53 74 61 74 75 73 22 3A 20 22 4F 46 22 7D Status": "OFF">
Publishing MQTT Shadow Update Message:
00000000 7B 20 22 74 69 6D 65 73 74 61 6D 70 22 3A 20 31 < "timestamp": 1
00000010 35 37 30 37 31 31 38 34 35 2C 20 22 4C 65 64 5F 570711845, "Led_
00000020 53 74 61 74 75 73 22 3A 20 22 4F 4E 22 7D Status": "ON">

```

Once successfully programmed the CryptoAuth Trust Platform, navigate to previous section 1.7 to run the [last step \(GCP GUI\)](#) in the Jupyter Notebook.

4.3 CryptoAuth Trust Platform Factory reset

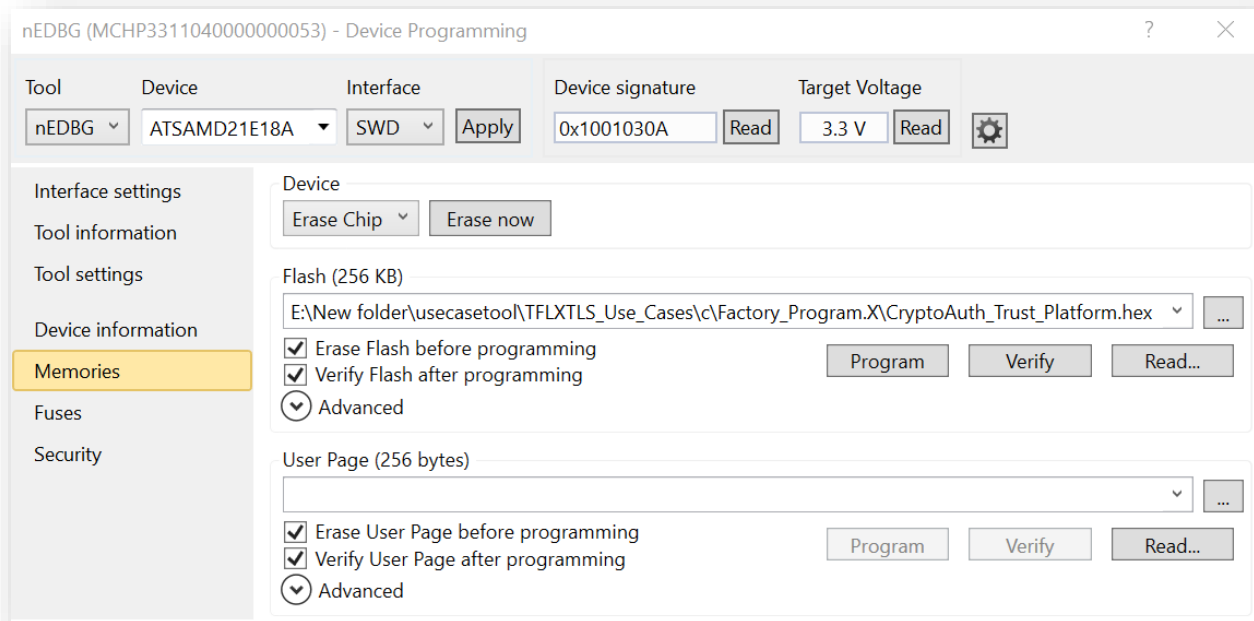
If any embedded project is loaded to CryptoAuth Trust Platform, the default program that enables interaction with CryptoAuth Trust Platform tools will be erased.

Before using the CryptoAuth Trust Platform with any other notebook or tools on PC, its required to reprogram the default firmware. Default hex file is available at

TFLXTLS_Use_Cases\c\Factory_Program.X\CryptoAuth_Trust_Platform.hex

To reprogram using Atmel Studio:

1. Navigate to AtmelStudio -> Tools -> Device Programming
2. Select Tool as nEDBG and Apply
3. Go to Memories and navigate to above path under Flash dropdown
4. Check both Erase Flash and Verify Flash
5. Click on Program



To reprogram using MPLAB:

1. Open **TFLXTLS_Use_Cases\c\Factory_Program.X** project in MPLAB IDE
2. Program the Crypto Trust platform by navigating to
CryptoAuth_Trust_Platform_Factory_Program -> Make and Program Device

Now, CryptoAuth Trust Platform contains factory application that enables interactions with Notebooks and/or PC tools.

5 FAQ

1. What are the reasons for “**AssertionError: Can't connect to the USB dongle**” error?

There are many possibilities like,

1. Crypto Trust Platform is having different application than factory reset firmware. Refer to “CryptoAuth TrustPlatform Factory reset” section any usecase TrustFLEX Guide for reloading it
2. Check the switch positions on Crypto Trust Platform and/or ATECC608A Trust board
 - a. Correct Trust device should be connected and only one device of that type is allowed on the I2C bus. Multiple devices with same address results in error
3. Check USB connections to Crypto Trust Platform

2. How to reload factory default application to Crypto Trust Platform?

Refer to “CryptoAuth TrustPlatform Factory reset” section any usecase TrustFLEX Guide for reloading it.

3. Why does my C projects generates No such file or directory with ../../../../TFLXTLS_resource_generation/?

C project generates this error when the resources are not generated prior to using embedded projects. Running the resource generation notebook ensures these files and secrets are generated.

4. Before running any use case notebook and/or C project, why is it mandate to execute resource generation?

When resource generation notebook is executed, it generates and programs the required resources like secrets, keys and certificates. These are only prototyping keys and cannot be used for production. These keys will be used part of Usecase notebooks and C projects

5. How to know the resources being used in a use case?

Refer to individual Usecase description html for details on transaction diagrams, resources being used and other details. The resources required for given use case is mentioned in INFER CRYPTOGRAPHIC ASSETS section.

The Microchip Web Site

Microchip provides online support via our web site at <http://www.microchip.com/>. This web site is used as

a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Customer Change Notification Service

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at <http://www.microchip.com/>. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support.

Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: <http://www.microchip.com/support>

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the

operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.

- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be

a

violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL,

STATUTORY

OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE.

Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, AVR, AVR logo, AVR Freaks, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, Helder, JukeBlox, KeeLoq,

Kleer, LANCheck, LINK MD, maXStylus, maXTouch, MediaLB, megaAVR, MOST, MOST logo, MPLAB,

OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, Prochip Designer, QTouch, SAM-BA, SpyNIC, SST,

SST Logo, SuperFlash, tinyAVR, UNI/O, and XMEGA are registered trademarks of Microchip Technology

Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, EtherSynch, Hyper Speed Control, HyperLight

Load, IntelliMOS, mTouch, Precision Edge, and Quiet-Wire are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming,

ICSP, INICnet, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, memBrain, Mindi, MiWi,

motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient

Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE,

Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2018, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN:

Quality Management System Certified by DNV

ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California

and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.



Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: http://www.microchip.com/support Web Address: www.microchip.com Atlanta Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 Austin, TX Tel: 512-257-3370 Boston Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 Chicago Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 Dallas Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 Detroit Novi, MI Tel: 248-848-4000 Houston, TX Tel: 281-894-5983 Indianapolis Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 Los Angeles Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 Raleigh, NC Tel: 919-844-7510 New York, NY Tel: 631-435-6000 San Jose, CA Tel: 408-735-9110 Tel: 408-436-4270 Canada - Toronto Tel: 905-695-1980 Fax: 905-695-2078	Australia - Sydney Tel: 61-2-9868-6733 China - Beijing Tel: 86-10-8569-7000 China - Chengdu Tel: 86-28-8665-5511 China - Chongqing Tel: 86-23-8980-9588 China - Dongguan Tel: 86-769-8702-9880 China - Guangzhou Tel: 86-20-8755-8029 China - Hangzhou Tel: 86-571-8792-8115 China - Hong Kong SAR Tel: 852-2943-5100 China - Nanjing Tel: 86-25-8473-2460 China - Qingdao Tel: 86-532-8502-7355 China - Shanghai Tel: 86-21-3326-8000 China - Shenyang Tel: 86-24-2334-2829 China - Shenzhen Tel: 86-755-8864-2200 China - Suzhou Tel: 86-186-6233-1526 China - Wuhan Tel: 86-27-5980-5300 China - Xian Tel: 86-29-8833-7252 China - Xiamen Tel: 86-592-2388138 China - Zhuhai Tel: 86-756-3210040	India - Bangalore Tel: 91-80-3090-4444 India - New Delhi Tel: 91-11-4160-8631 India - Pune Tel: 91-20-4121-0141 Japan - Osaka Tel: 81-6-6152-7160 Japan - Tokyo Tel: 81-3-6880-3770 Korea - Daegu Tel: 82-53-744-4301 Korea - Seoul Tel: 82-2-554-7200 Malaysia - Kuala Lumpur Tel: 60-3-7651-7906 Malaysia - Penang Tel: 60-4-227-8870 Philippines - Manila Tel: 63-2-634-9065 Singapore Tel: 65-6334-8870 Taiwan - Hsin Chu Tel: 886-3-577-8366 Taiwan - Kaohsiung Tel: 886-7-213-7830 Taiwan - Taipei Tel: 886-2-2508-8600 Thailand - Bangkok Tel: 66-2-694-1351 Vietnam - Ho Chi Minh Tel: 84-28-5448-2100	Austria - Wels Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 Denmark - Copenhagen Tel: 45-4450-2828 Fax: 45-4485-2829 Finland - Espoo Tel: 358-9-4520-820 France - Paris Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 France - Saint Cloud Tel: 33-1-30-60-70-00 Germany - Garching Tel: 49-8931-9700 Germany - Haan Tel: 49-2129-3766400 Germany - Heilbronn Tel: 49-7131-67-3636 Germany - Karlsruhe Tel: 49-721-625370 Germany - Munich Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 Germany - Rosenheim Tel: 49-8031-354-560 Israel - Ra'anana Tel: 972-9-744-7705 Italy - Milan Tel: 39-0331-742611 Fax: 39-0331-466781 Italy - Padova Tel: 39-049-7625286 Netherlands - Drunen Tel: 31-416-690399 Fax: 31-416-690340 Norway - Trondheim Tel: 47-7289-7561 Poland - Warsaw Tel: 48-22-3325737 Romania - Bucharest Tel: 40-21-407-87-50 Spain - Madrid Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 Sweden - Gothenberg Tel: 46-31-704-60-40 Sweden - Stockholm Tel: 46-8-5090-4654 UK - Wokingham Tel: 44-118-921-5800 Fax: 44-118-921-5820