Trust&GO 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 Trust&GO – Manifest file generation 7

4 Use Case Prototyping 12

4.1 Running GCP example on Jupyter Notebook 12

4.2 Running GCP example on Embedded platform 17

4.2.1 Atmel Studio: 17

4.2.2 MPLAB: 20

4.3 CryptoAuth Trust Platform Factory reset 23

5 FAQ 25

1 Introduction

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

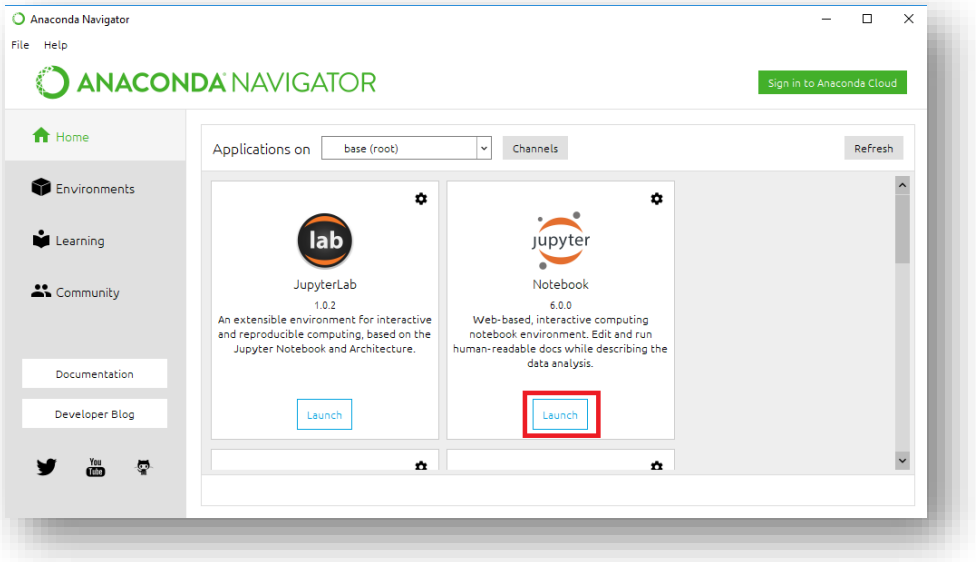
1.1 Getting started with Jupyter Notebook Tutorials

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

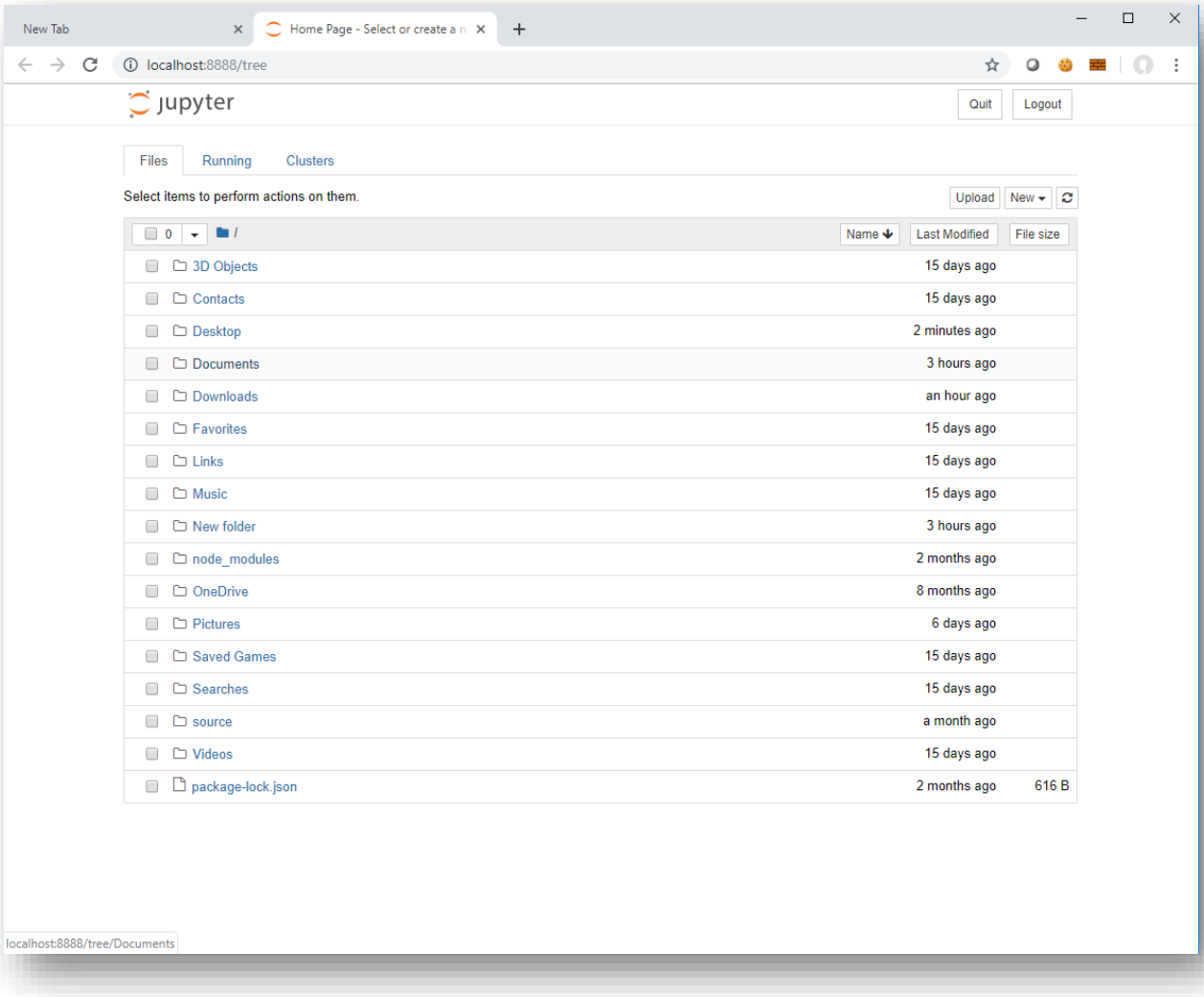
It is recommended to become familiar with Jupyter basic concepts with the online documentation, https://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/Notebook%20Basics.html

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.

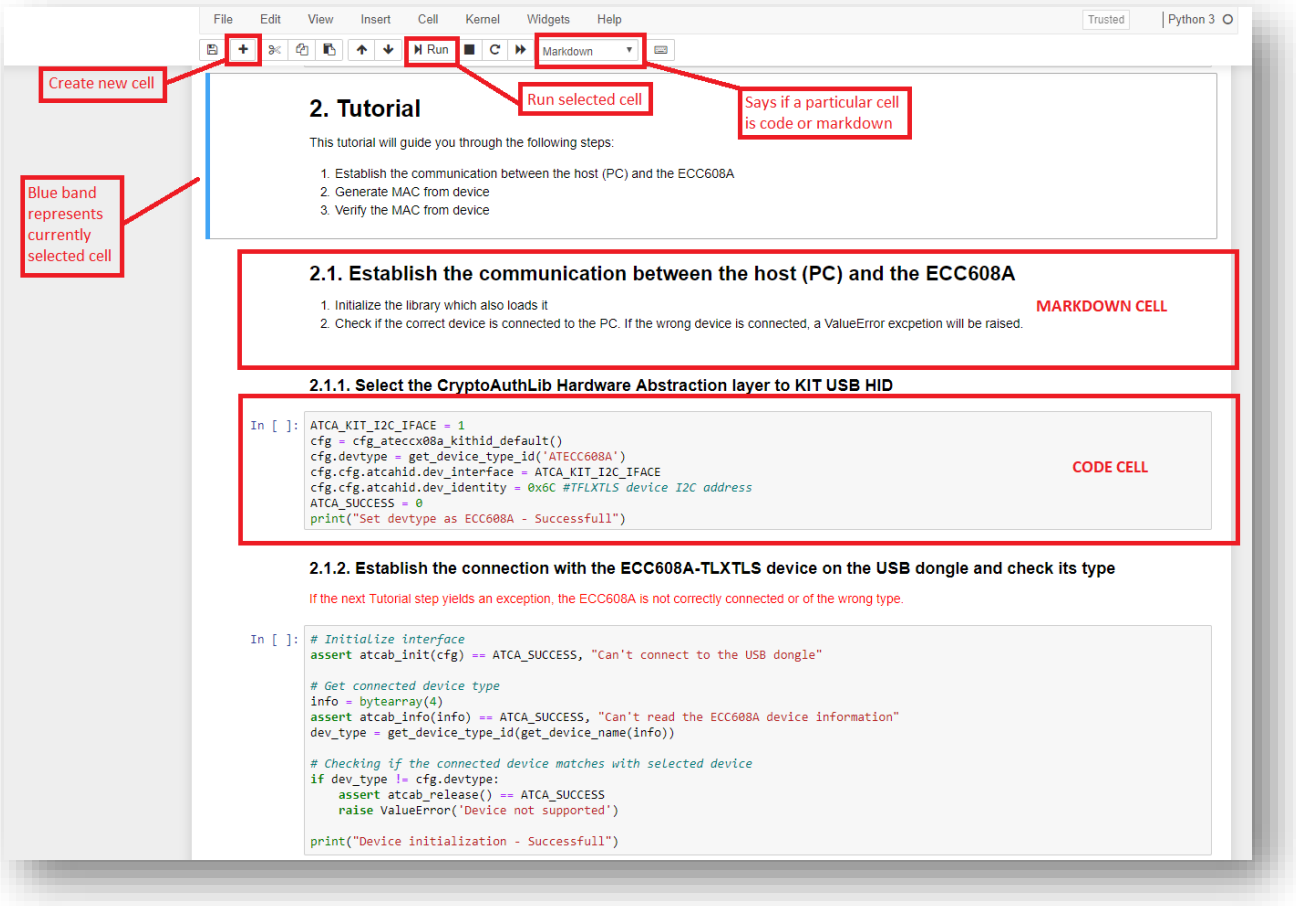


1.3 Introduction to Jupyter Notebook GUI.

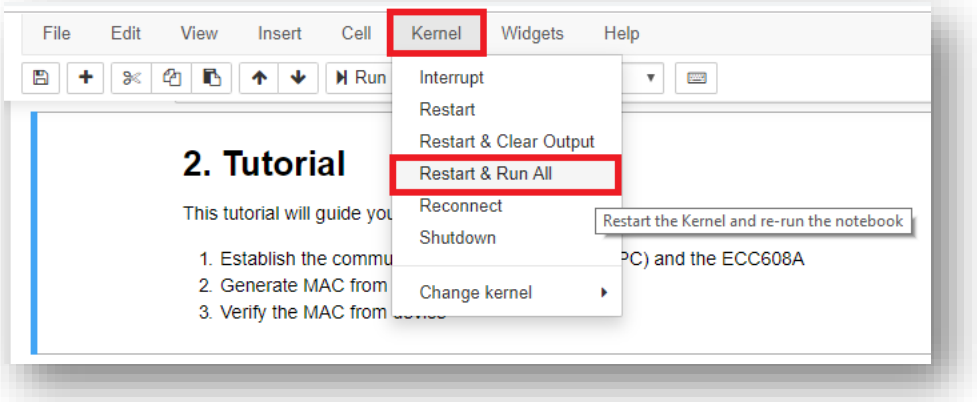
Jupyter Notebooks contain cells where you can either write code or markdown text. Notebooks contain multiple cells, some set as code and others markdown. Code cells contain code that can be executed live, and markdown contains text and images that explains the code.

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 Trust&Go devices. Here is the list of Jupyter Notebook Tutorials.

Jupyter Notebook Tutorials	**Relative Path**	**Applicable Devices**
Manifest Generation	TrustnGO\00_resource_generation\TNGTLS_manifest_file_generation.ipynb	Trust&GO
AWS Connect	TrustnGO\01_aws_connect\notebook\ TNGTLS_aws_connect.ipynb	Trust&GO
GCP Connect	TrustnGO\02_gcp_connect\notebook\TNGTLS_GCP_connect.ipynb	Trust&GO

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 Puchasing & 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 Trust&GO – Manifest file generation

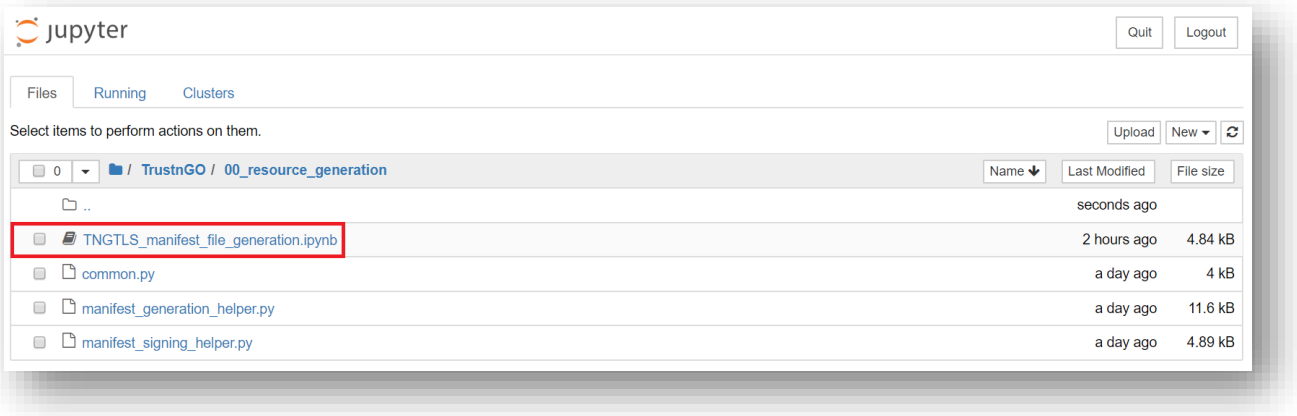
Trust&GO device is one of the three devices available in the Trust Platform USB Dongle Board.

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.

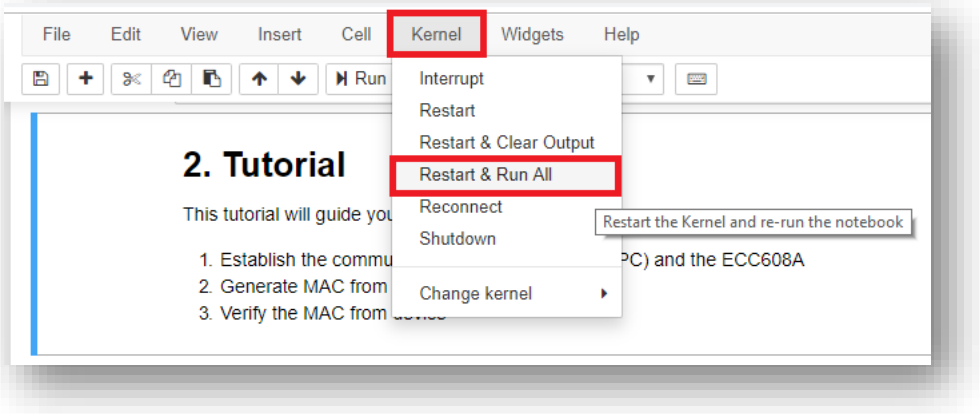
The secure element manifest format is designed to convey the unique information about a device including its unique ID (e.g. serial number), public keys, and certificates. The manifest file generated can be used to register the device to cloud providers.

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\ TrustnGO\00_resource_generation** folder to open **TNGTLS_manifest_file_generation.ipynb**



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



If all the steps ran without errors, you will see result as shown below.

```
Loading logger key

Generating self-signed logging certificate
   Saving to log_signer.crt
-----------------------------------------
TNG Root Certificate:
Certificate:
   Data:
      Version: 3 (0x2)
      Serial Number:
         77:d3:6d:95:6e:c8:ae:62:05:e5:8e:3a:cb:98:5a:81
      Signature Algorithm: ecdsa-with-SHA256
      Issuer: O = Microchip Technology Inc, CN = Crypto Authentication Root CA 002
      Validity
         Not Before: Nov  8 19:12:19 2018 GMT
         Not After : Nov  8 19:12:19 2058 GMT
      Subject: O = Microchip Technology Inc, CN = Crypto Authentication Root CA 002
      Subject Public Key Info:
         Public Key Algorithm: id-ecPublicKey
            Public-Key: (256 bit)
            pub:
               04:bd:54:e6:6d:e3:87:54:84:00:6b:53:ae:15:80:
               d5:0a:a0:69:e7:8a:df:55:78:d8:5c:e2:d5:4d:d5:
               b8:30:29:6b:ff:dd:6e:6f:72:56:fb:d9:9e:f1:a1:
               16:b1:1d:33:ad:49:10:3a:a1:85:87:39:dc:fa:e4:
               37:e1:9d:63:4e
            ASN1 OID: prime256v1
            NIST CURVE: P-256
      X509v3 extensions:
         X509v3 Subject Key Identifier:
            7A:ED:7D:6D:C6:B7:78:9D:B2:38:01:A5:E8:4A:8C:B0:A4:0E:2A:8C
         X509v3 Authority Key Identifier:
            keyid:7A:ED:7D:6D:C6:B7:78:9D:B2:38:01:A5:E8:4A:8C:B0:A4:0E:2A:8C
```

```
            X509v3 Basic Constraints: critical
                CA:TRUE
    Signature Algorithm: ecdsa-with-SHA256
         30:45:02:21:00:a1:dc:63:45:90:ec:81:9e:e1:de:5b:81:12:
         65:51:ad:d4:c2:c4:f8:e5:95:28:2e:e0:4b:e7:68:ec:7c:02:
         73:02:20:3e:6b:a7:4e:9e:4c:0a:d6:8c:24:b0:fb:2e:e7:93:
         d2:e6:be:94:65:ca:15:d0:ea:5b:c8:7f:55:79:99:5c:ad
-----BEGIN CERTIFICATE-----
MIIB8TCCAZegAwIBAgIQd9NtlW7IrmIF5Y46y5hagTAKBggqhkjOPQQDAjBPMSEw
HwYDVQQKDBhNaWNyb2NoaXAgVGVjaG5vbG9neSBJbmMxKjAoBgNVBAMMIUNyeXB0
byBBdXRoZW50aWNhdGlvbiBSb290IENBIDAwMjAgFw0xODExMDgxOTEyMTlaGA8y
MDU4MTEwODE5MTIxOVowTzEhMB8GA1UECgwYTWljcm9jaGlwIFRlY2hub2xvZ3kg
SW5jMSowKAYDVQQDDCFDcnlwdG8gQXV0aGVudGljYXRpb24gUm9vdCBDQSAwMDIw
WTATBgcqhkjOPQIBBggqhkjOPQMBBwNCAAS9VOZt44dUhABrU64VgNUKoGnnit9V
eNhc4tVN1bgwKWv/3W5vclb72Z7xoRaxHTOtSRA6oYWHOdz65DfhnWNOo1MwUTAd
BgNVHQ4EFgQUeu19bca3eJ2yOAGl6EqMsKQOKowwHwYDVR0jBBgwFoAUeu19bca3
eJ2yOAGl6EqMsKQOKowwDwYDVR0TAQH/BAUwAwEB/zAKBggqhkjOPQQDAgNIADBF
AiEAodxjRZDsgZ7h3luBEmVRrdTCxPjllSgu4EvnaOx8AnMCID5rp06eTArWjCSw
+y7nk9LmvpRlyhXQ6lvIf1V5mVyt
-----END CERTIFICATE-----

Validate Root Certificate:
OK

TNG Signer Certificate:
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            7c:31:16:e0:34:06:5b:25:e0:53:bf:b0:9e:26:1a:66
        Signature Algorithm: ecdsa-with-SHA256
        Issuer: O = Microchip Technology Inc, CN = Crypto Authentication Root CA 002
        Validity
            Not Before: Dec 14 19:00:00 2018 GMT
            Not After : Dec 14 19:00:00 2049 GMT
        Subject: O = Microchip Technology Inc, CN = Crypto Authentication Signer F640
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
                pub:
                    04:0e:7f:32:95:f2:eb:46:40:f4:aa:39:4c:e5:e5:
                    20:d7:d9:b5:0b:b9:54:f5:c3:f8:bb:6f:3f:2c:18:
                    5a:4d:b9:90:91:bc:af:49:ea:06:fc:4b:cc:5c:2a:
                    27:c8:d0:02:2b:1a:9b:f0:1b:f4:46:dd:9a:c5:62:
                    30:4c:7a:38:76
                ASN1 OID: prime256v1
                NIST CURVE: P-256
        X509v3 extensions:
            X509v3 Key Usage: critical
                Digital Signature, Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE, pathlen:0
```

```
            X509v3 Subject Key Identifier:
                EB:62:03:2B:8C:81:58:16:42:9A:1C:AB:F1:BE:8C:22:1E:B7:A6:C9
            X509v3 Authority Key Identifier:
                keyid:7A:ED:7D:6D:C6:B7:78:9D:B2:38:01:A5:E8:4A:8C:B0:A4:0E:2A:8C

    Signature Algorithm: ecdsa-with-SHA256
         30:46:02:21:00:e0:78:ed:06:27:15:f0:31:32:f0:ef:31:97:
         22:a5:49:e2:e1:86:28:73:b5:f2:b7:5f:28:3e:e3:7d:13:6e:
         bc:02:21:00:df:68:76:be:05:22:ae:7f:e9:15:84:82:fa:08:
         45:8d:da:9c:f3:08:17:2f:d6:60:74:fc:11:90:7a:30:47:7b
-----BEGIN CERTIFICATE-----
MIICBTCCAaqgAwIBAgIQfDEW4DQGWyXgU7+wniYaZjAKBggqhkjOPQQDAjBPMSEw
HwYDVQQKDBhNaWNyb2NoaXAgVGVjaG5vbG9neSBJbmMxKjAoBgNVBAMMIUNyeXB0
byBBdXRoZW50aWNhdGlvbiBSb290IENBIDAwMjAgFw0xODEyMTQxOTAwMDBaGA8y
MDQ5MTIxNDE5MDAwMFowTzEhMB8GA1UECgwYTWljcm9jaGlwIFRlY2hub2xvZ3kg
SW5jMSowKAYDVQQDDCFDcnlwdG8gQXV0aGVudGljYXRpb24gU2lnbmVyIEY2NDAw
WTATBgcqhkjOPQIBBggqhkjOPQMBBwNCAAQOfzKV8utGQPSqOUzl5SDX2bULuVT1
w/i7bz8sGFpNuZCRvK9J6gb8S8xcKifI0AIrGpvwG/RG3ZrFYjBMejh2o2YwZDAO
BgNVHQ8BAf8EBAMCAYYwEgYDVR0TAQH/BAgwBgEB/wIBADAdBgNVHQ4EFgQU62ID
K4yBWBZCmhyr8b6MIh63pskwHwYDVR0jBBgwFoAUeu19bca3eJ2yOAGl6EqMsKQO
KowwCgYIKoZIzj0EAwIDSQAwRgIhAOB47QYnFfAxMvDvMZcipUni4YYoc7Xyt18o
PuN9E268AiEA32h2vgUirn/pFYSC+ghFjdqc8wgXL9ZgdPwRkHowR3s=
-----END CERTIFICATE-----

Validate Signer Certificate:
OK

TNG Device Certificate:
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            71:4f:f4:a3:24:1b:e8:18:dc:6d:8b:b0:c5:53:aa:6f
        Signature Algorithm: ecdsa-with-SHA256
        Issuer: O = Microchip Technology Inc, CN = Crypto Authentication Signer F640
        Validity
            Not Before: Aug  1 00:00:00 2019 GMT
            Not After : Aug  1 00:00:00 2047 GMT
        Subject: O = Microchip Technology Inc, CN = 0123CEDC3CD337AA01 ATECC
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
                pub:
                    04:97:24:09:48:8c:20:89:19:6f:24:47:b0:8a:4d:
                    90:cc:74:d6:cc:14:37:60:b0:aa:b5:76:4d:cd:bf:
                    03:f6:06:e9:46:80:18:e5:2c:9f:75:4a:3b:c8:3f:
                    17:2d:b5:24:d7:5b:c4:ba:f7:34:92:f3:a3:45:58:
                    c2:eb:ae:69:34
                ASN1 OID: prime256v1
                NIST CURVE: P-256
        X509v3 extensions:
            X509v3 Basic Constraints: critical
                CA:FALSE
```

```
        X509v3 Key Usage: critical
            Digital Signature, Key Agreement
        X509v3 Subject Key Identifier:
            09:61:2F:3F:E6:E6:AE:20:35:71:7D:CB:C7:A7:49:AC:63:E5:FC:66
        X509v3 Authority Key Identifier:
            keyid:EB:62:03:2B:8C:81:58:16:42:9A:1C:AB:F1:BE:8C:22:1E:B7:A6:C9

    Signature Algorithm: ecdsa-with-SHA256
        30:46:02:21:00:a1:52:e2:bb:be:01:1d:91:8e:12:0c:9b:a6:
        67:67:6a:55:78:e9:66:6a:72:84:ba:17:2f:ce:9b:72:f7:6c:
        3b:02:21:00:c7:29:b1:2c:5d:43:cb:7c:f2:67:7d:12:c8:59:
        82:eb:4f:78:0e:49:8a:16:11:62:55:2a:15:45:f8:2a:f4:a5
-----BEGIN CERTIFICATE-----
MIIB9jCCAZugAwIBAgIQcU/0oyQb6BjcbYuwxVOqbzAKBggqhkjOPQQDAjBPMSEw
HwYDVQQKDBhNaWNyb2NoaXAgVGVjaG5vbG9neSBJbmMxKjAoBgNVBAMMIUNyeXB0
byBBdXRoZW50aWNhdGlvbiBTaWduZXIgRjY0MDAgFw0xOTA4MDEwMDAwMDBaGA8y
MDQ3MDgwMTAwMDAwMFowRjEhMB8GA1UECgwYTWljcm9jaGlwIFRlY2hub2xvZ3kg
SW5jMSEwHwYDVQQDDBgwMTIzQ0VEQzNDRDMzN0FBMDEgQVRFQ0MwWTATBgcqhkjO
PQIBBggqhkjOPQMBBwNCAASXJAlIjCCJGW8kR7CKTZDMdNbMFDdgsKq1dk3NvwP2
BulGgBjlLJ91SjvIPxcttSTXW8S69zSS86NFWMLrrmk0o2AwXjAMBgNVHRMBAf8E
AjAAMA4GA1UdDwEB/wQEAwIDiDAdBgNVHQ4EFgQUCWEvP+bmriA1cX3Lx6dJrGPl
/GYwHwYDVR0jBBgwFoAU62IDK4yBWBZCmhyr8b6MIh63pskwCgYIKoZIzj0EAwID
SQAwRgIhAKFS4ru+AR2RjhIMm6ZnZ2pVeOlmanKEuhcvzpty92w7AiEAxymxLF1D
y3zyZ30SyFmC6094DkmKFhFiVSoVRfgq9KU=
-----END CERTIFICATE-----

Validate Device Certificate:
OK

------------------------------------------------------
Generated the manifest file 0123cedc3cd337aa01_manifest.json
```

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.

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.

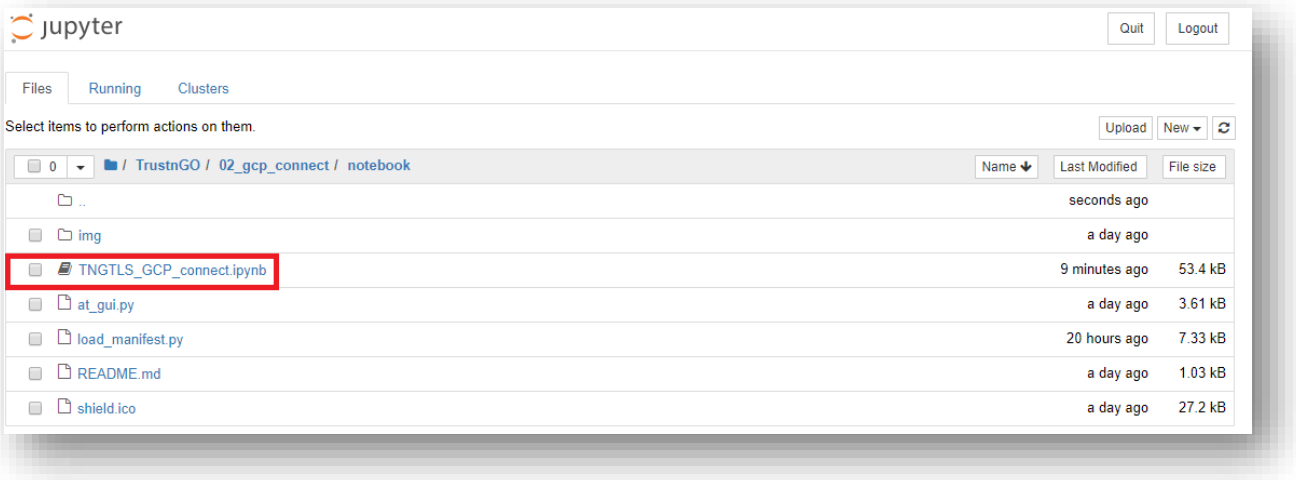
The reference implementation is provided with Embedded projects and Notebooks. The generation of manifest can be achieved through the execution of Jupyter Notebook Tutorials.

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_guide_GCP_demo_account_setup.pdf**.

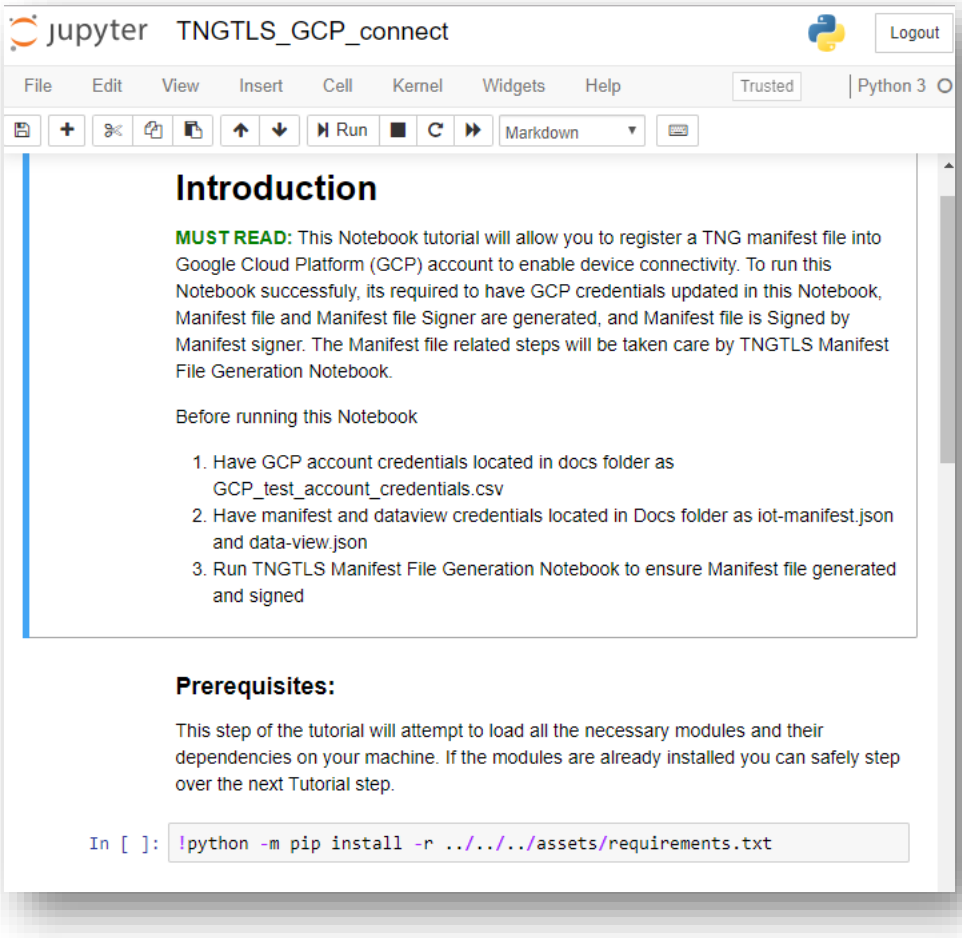
4.1 Running GCP example on Jupyter Notebook

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).

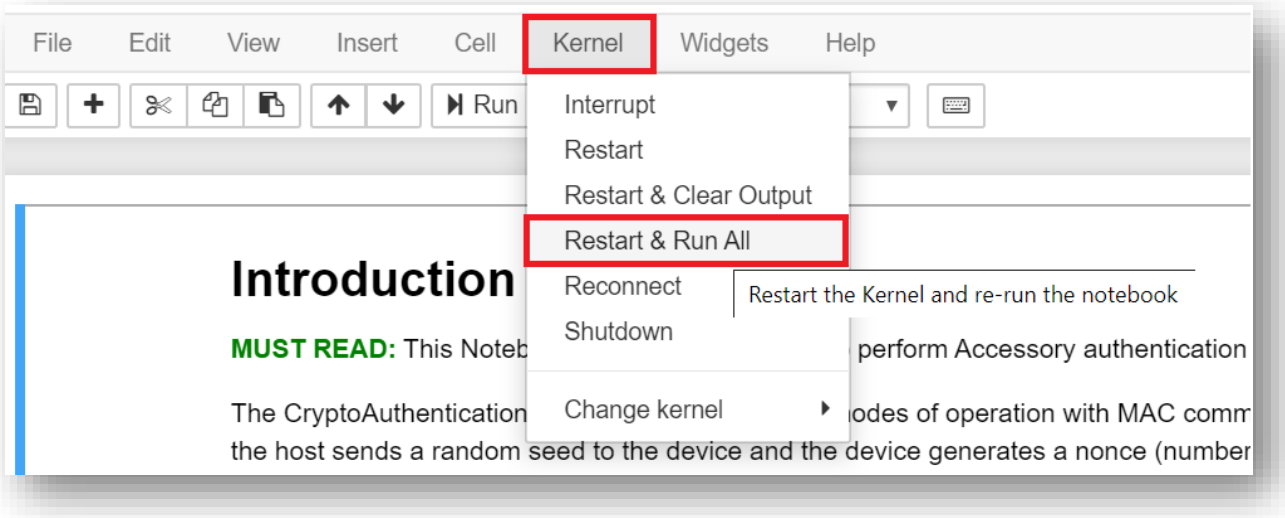
1. From the Jupyter Home page, navigate to **TrustnGO\02_gcp_connect\notebook\ TNGTLS_GCP_connect.ipynb** notebook file and open it.



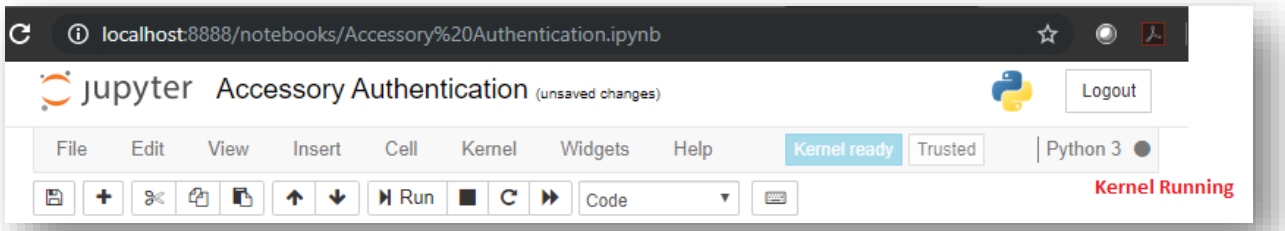
Opening the Jupyter notebook example should load the following on the browser.

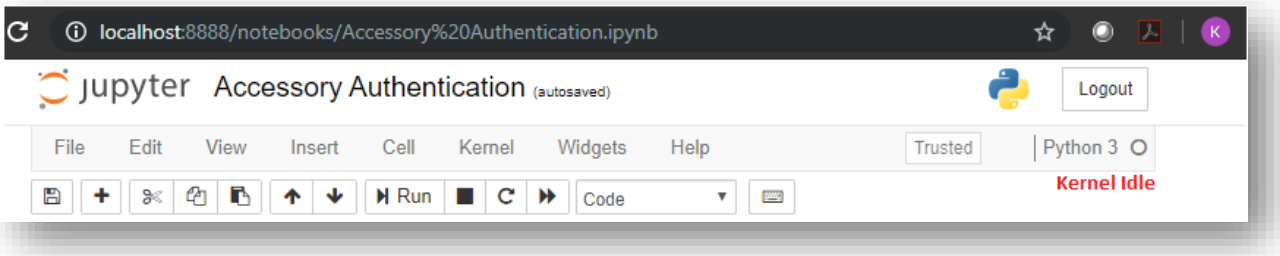

jupyter TNGTLS_GCP_connect
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Markdown
Introduction
MUST READ: This Notebook tutorial will allow you to register a TNG manifest file into Google Cloud Platform (GCP) account to enable device connectivity. To run this Notebook successfuly, its required to have GCP credentials updated in this Notebook, Manifest file and Manifest file Signer are generated, and Manifest file is Signed by Manifest signer. The Manifest file related steps will be taken care by TNGTLS Manifest File Generation Notebook.
Before running this Notebook
1. Have GCP account credentials located in docs folder as GCP_test_account_credentials.csv
2. Have manifest and dataview credentials located in Docs folder as iot-manifest.json and data-view.json
3. Run TNGTLS Manifest File Generation Notebook to ensure Manifest file generated and signed
Prerequisites:
This step of the tutorial will attempt to load all the necessary modules and their dependencies on your machine. If the modules are already installed you can safely step over the next Tutorial step.
In [ ]:
!python -m pip install -r ../../../assets/requirements.txt

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



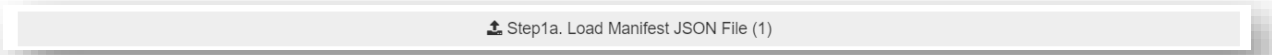
It may take a while to complete, wait for the kernel to complete all processing i.e. from Kernel Running to Kernel Idle state (Check circle above **RED** text)





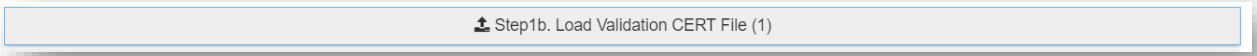
3. Navigate through different cells output for the description of the step and result from the execution.

4. There are 3 major steps:
   Load Manifest File:
   Under the section **Upload Manifest File**, click the button '**Load Manifest JSON File**' and select the
   manifest file generated from the TrustnGO Resource generation notebook.



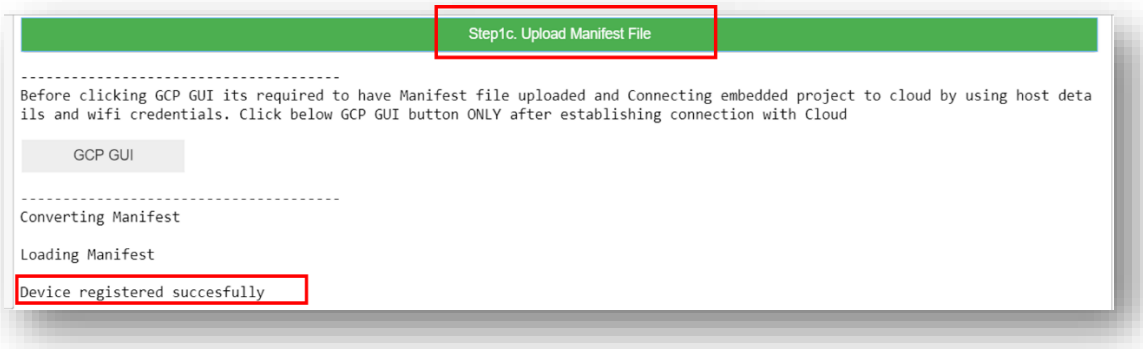
Load validation certificate:
click the button '**Load Validation CERT File**' and
select the validation certificate which signed the manifest file and it should be present in the following folder with name log_signer.crt

For Trust&GO - TrustnGO\00_resource_generation\



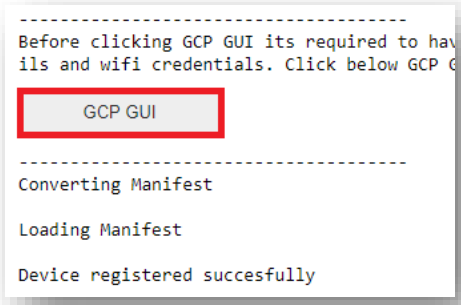
Register device manifest file:
Code block of this step generates "**Upload manifest File**" button. Clicking the button, it registers the device manifest file to the GCP account. Once the manifest file is registered, the gcp cloud authorizes the Trust Platform device and it will be able to
communicate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

Upon successful execution, the log should look like this.

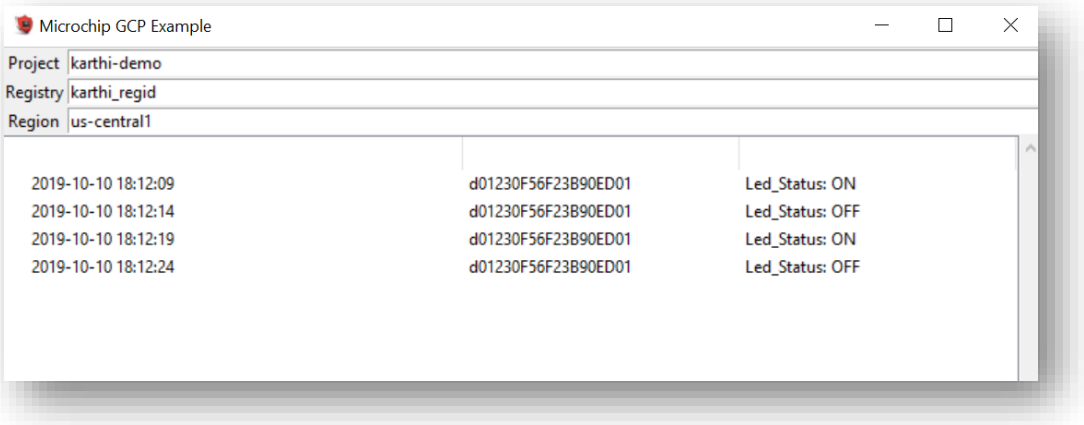


WARNING: It is required to execute C project successfully before executing the next step in the Jupyter notebook. To execute C project, refer "Running GCP IoT example on Embedded platform" next section.

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



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

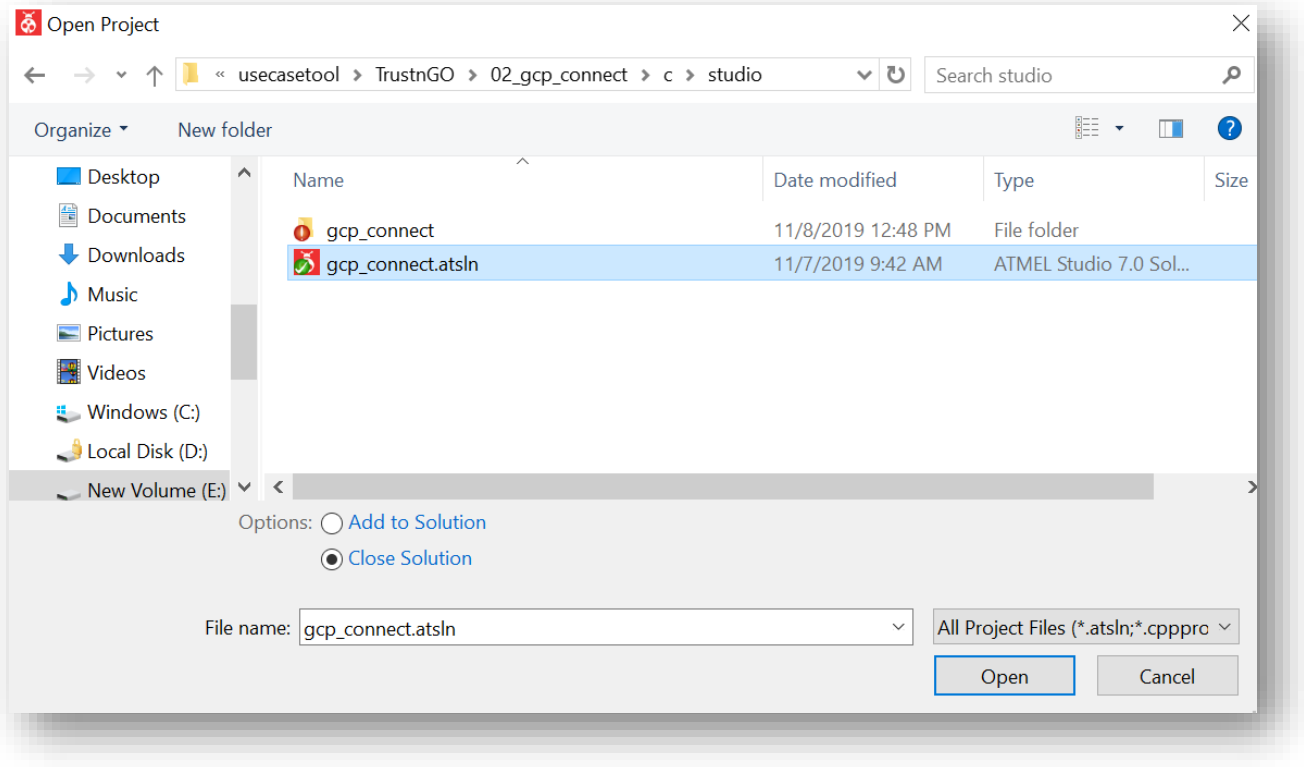
4.2 Running GCP example on Embedded platform

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.

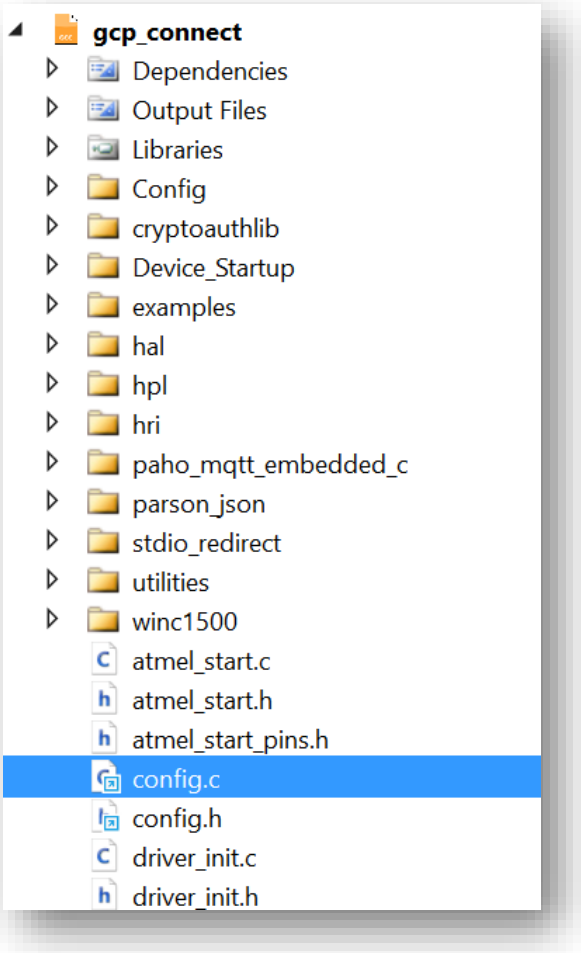
This project establishes a TLS connection and subscribe to MQTT. It is required to use the GCP IoT Jupyter notebook to register the device through manifest file. Prior to executing the application, it is required to update Wifi credentials, GCP account details. Following steps provides the instructions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

4.2.1 Atmel Studio:

1. Open **gcp_connect.atsln** project by navigating **TrustnGO\02_gcp_connect\c\studio \gcp_connect.atsln**



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

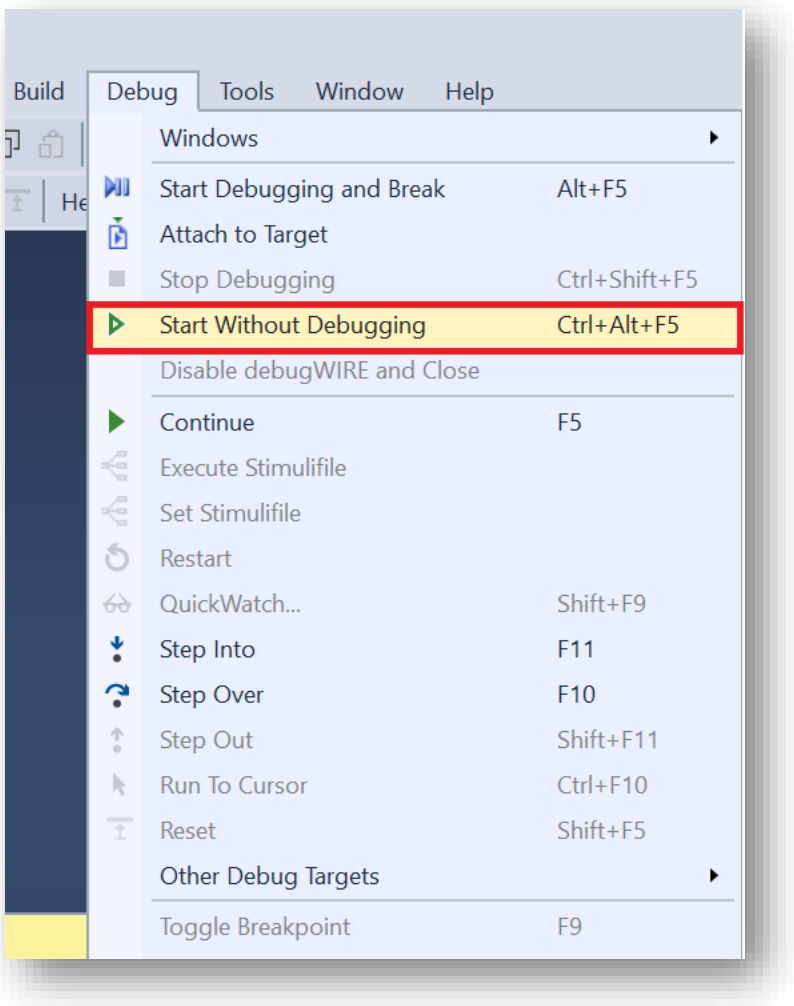


update the following constants before building the project:
The project id, region id and registry id should be same as in the gcp account setup.

- `config_demo_ssid`
- `config_demo_pass`
- `config_gcp_project_id`
- `config_gcp_region_id`
- `config_gcp_registry_id`

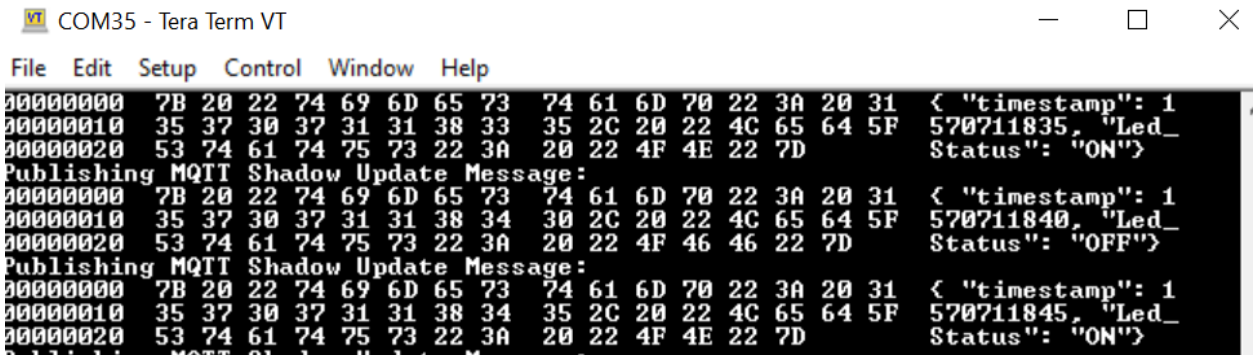
```
/* Example Configuration Data  Global Variables */
const char config_demo_ssid[] = "xxxxxxxxx";
const char config_demo_pass[] = "xxxxxxxxx";

const char config_gcp_project_id[] = "xxxxxxxxx";
const char config_gcp_region_id[] = "xxxxxxxxx";
const char config_gcp_registry_id[] = "xxxxxxxxx";
```

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.

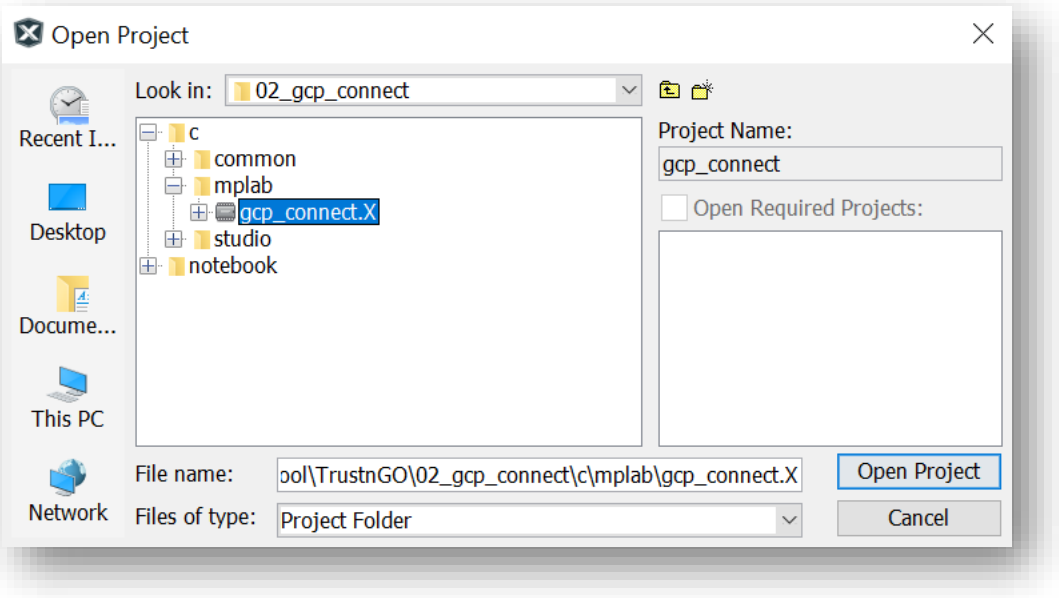
Once the programming is done, reset the hardware (press the reset button) and view the Console messages by using applications like 'Tera Term'. Open the application with the COM related to CryptoAuth Trust Platform with 115200-8-N-1 settings.



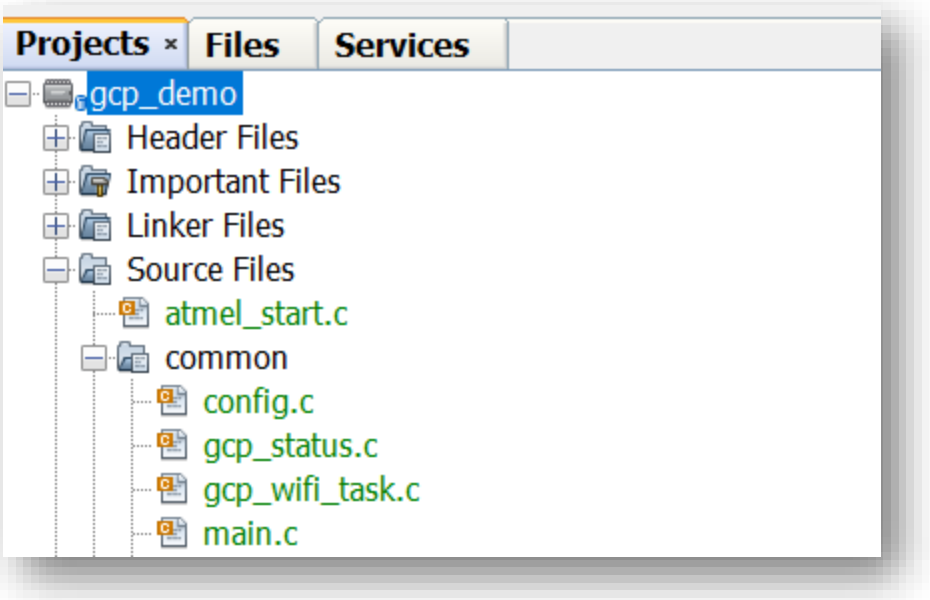
Once successfully programmed the CryptoAuth Trust Platform, navigate to previous section 1.7 to run the last step (GCP GUI) in the Jupyter Notebook.

4.2.2 MPLAB:

1. Open **gcp_demo.X** project by navigating to MPLAB -> File -> Open Project -> **TrustnGO\02_gcp_connect\c\mplab\gcp_demo.X**



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

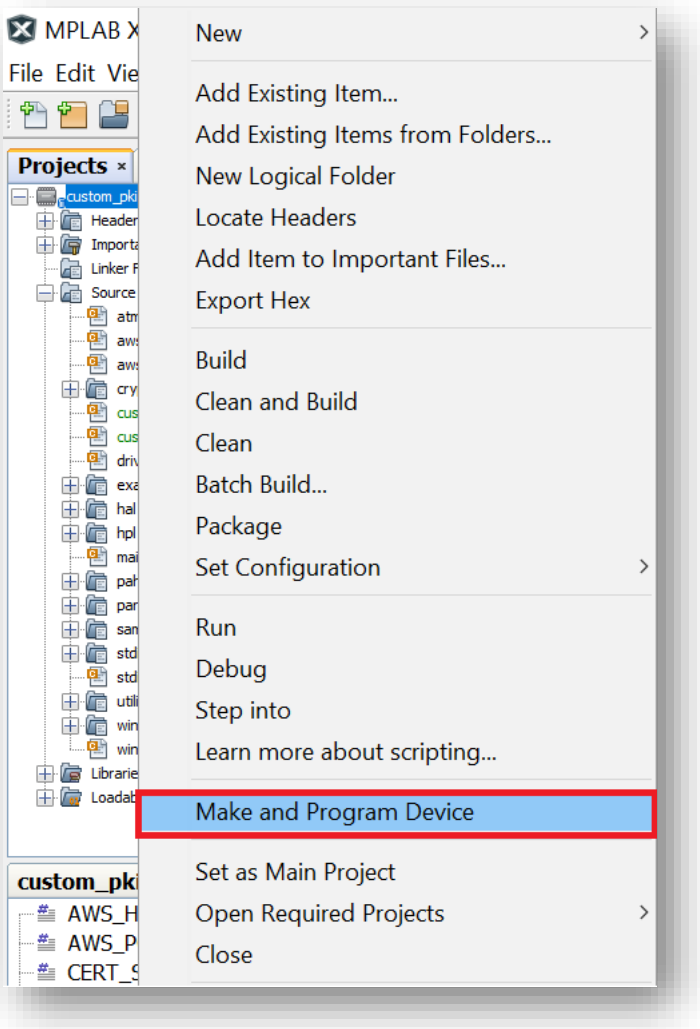


update the following constants before building the project:
The project id, region id and registry id should be same as in the gcp account setup.

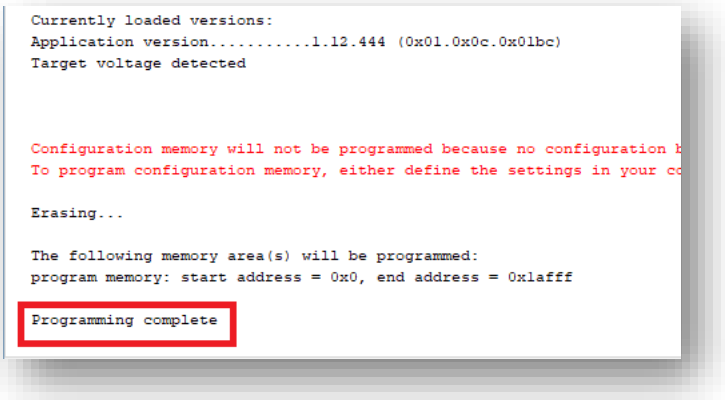
- config_demo_ssid
- config_demo_pass
- config_gcp_project_id
- config_gcp_region_id
- config_gcp_registry_id

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

const char config_gcp_project_id[] = "xxxxxxxxx";
const char config_gcp_region_id[] = "xxxxxxxxx";
const char config_gcp_registry_id[] = "xxxxxxxxx";
```

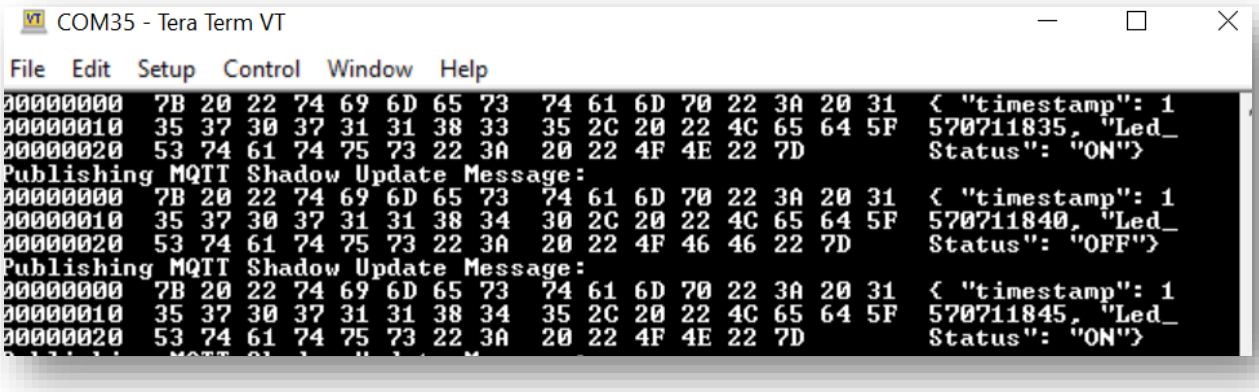
3. Program the CryptoAuth Trust platform by navigating to **gcp_connect -> Make and Program Device**



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.



Once successfully programmed the CryptoAuth Trust Platform, navigate to previous section 4.1 to run the last step (GCP GUI) in the Jupyter Notebook.

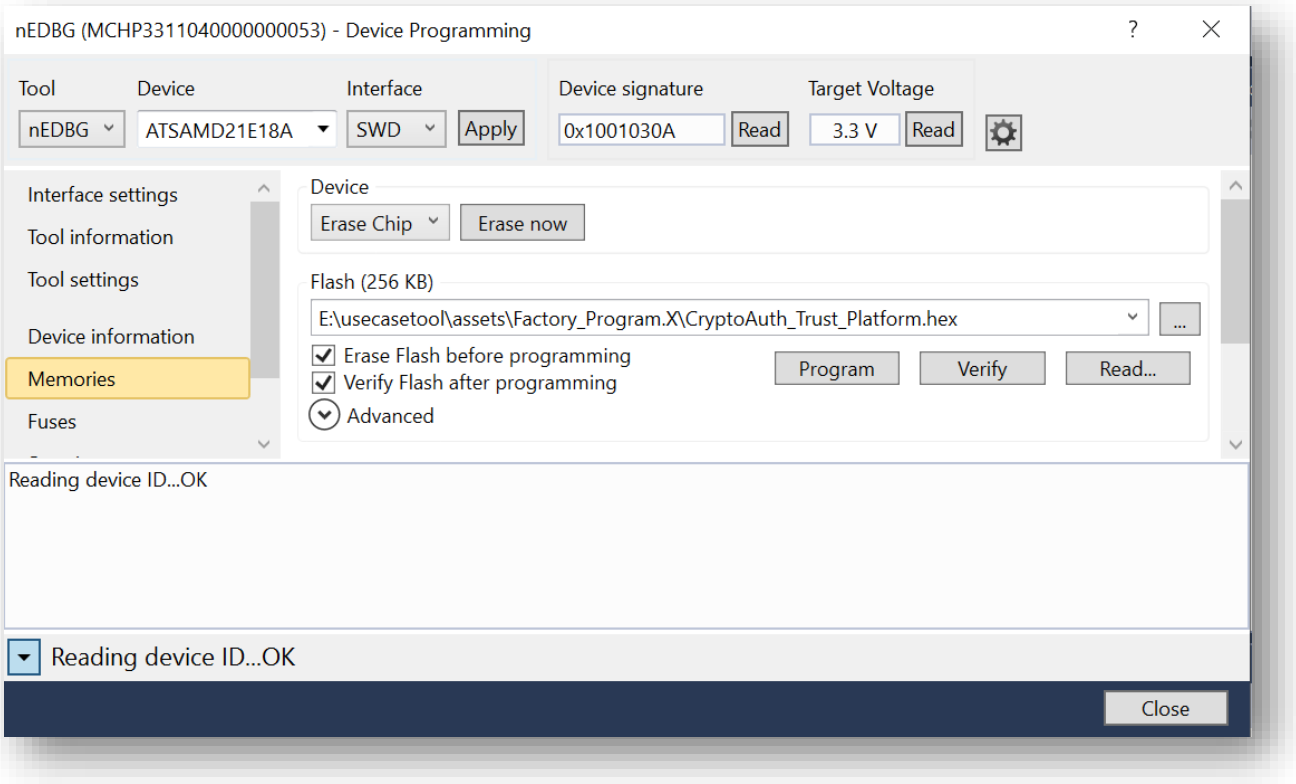
4.3 CryptoAuth Trust Platform Factory reset

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 **assets\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 **assets\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

Trademarks

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

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

ASIA/PACIFIC

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

ASIA/PACIFIC

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

EUROPE

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