

OPTIGA™ TPM Application Note

PSoC 6 Onboarding to AWS IoT Core

Devices

- OPTIGA™ TPM SLB 9670 TPM2.0
- OPTIGA™ TPM SLI 9670 TPM2.0
- OPTIGA™ TPM SLM 9670 TPM2.0

About This Document

Scope and purpose

This document explains how an OPTIGA™ TPM SLx 9670 TPM2.0 can be integrated into a PSoC 6 Wi-Fi BT Prototyping Kit to enable TPM backed onboarding to AWS IoT Core.

Onboarding is a process to obtain SSL/TLS certificates and ultimately gaining access to a server. To do so, a device generates a CSR containing information that a server will use to issue an SSL/TLS certificate. If a device identity is tied to a TPM key, the CSR must contain the TPM public key and must be signed with the corresponding private key. The public key will also be included in the SSL/TLS certificate.

The TPM software library of PSoC 6 is based on the open-source software [4][5]. It is portable across different embedded C/barebone projects.

The OPTIGA™ TPM SLx 9670 TPM2.0 uses a SPI interface to communicate with the PSoC 6 Wi-Fi BT Prototyping Kit. The OPTIGA™ TPM SLx 9670 TPM2.0 product family with SPI interface consists of 3 different products:

- OPTIGA™ TPM SLB 9670 TPM2.0 standard security applications
- OPTIGA™ TPM SLI 9670 TPM2.0 automotive security applications
- OPTIGA™ TPM SLM 9670 TPM2.0 industrial security applications

OPTIGA™ TPM SLx 9670 TPM2.0 products are fully TCG compliant TPM products with CC (EAL4+) and FIPS certification. The OPTIGA™ TPM SLx 9670 TPM2.0 products standard, automotive, and industrial differ with regards to supported temperature range, lifetime, quality grades, test environment, qualification, and reliability to fit the target applications requirements. An overview of all Infineon OPTIGA™ TPM products can be found on Infineon's website [8][9]. More information on TPM specification can be found on Trusted Computing Group (TCG) in reference [10].

Intended audience

This document is intended for customers who want to increase the security level of their platforms using a TPM 2.0 and like to evaluate the implementation of TPM backed onboarding for their target applications.

Table of contents

Table of contents	2
List of figures	3
List of tables	4
Acronyms and Abbreviations	5
1 Prerequisites	6
2 Host Machine Setup	7
3 Project Setup	8
3.1 TPM to PSoC 6 Connection.....	8
3.2 AWS CLI Setup	9
3.3 Serial Over USB.....	9
3.4 PSoC 6 Project Setup.....	9
3.5 PSoC 6 in Operation	12
Appendices	14
References.....	16
Revision history.....	17

List of figures

List of figures

Figure 1	PSoC 6 Wi-Fi BT Prototyping Kit (CY8CPROTO-062-4343W)	6
Figure 2	Iridium9670 TPM2.0.....	6
Figure 3	Iridium9670 TPM2.0 pins.....	8
Figure 4	PSoC 6 Wi-Fi BT prototyping kit GPIO layout, 6.0: MOSI, 6.1: MISO, 6.2: CLK, 6.3: CS	8
Figure 5	MQTT test client	13

List of tables

List of tables

Table 1	GitHub repository	7
Table 2	Project setup	10

Acronyms and Abbreviations

Acronym	Definition
AWS	Amazon Web Services
BSP	Board Support Package
BT	Bluetooth
CA	Certificate Authority
CC	Common Criteria
CLI	Command Line Interpreter
CRT	Certificate
CSR	Certificate Signing Request
GPIO	General-purpose Input/Output
IDE	Integrated Development Environment
IoT	Internet of Things
LTS	Long-term support
MQTT	Message Queuing Telemetry Transport
PSoC	Programmable system-on-chip
RSA	Asymmetric cryptography known as Ron R ivest, Adi S hamir and Leonard A dleman
SPI	Serial Peripheral Interface
SSL	Secure Sockets Layer
TCG	Trusted Computing Group
TCTI	TPM Command Transmission Interface
TIS	TPM Interface Specification
TLS	Transport Layer Security
TPM	Trusted Platform Module
TSS	TPM/TCG Software Stack
USB	Universal Serial Bus
Wi-Fi	Wireless Fidelity

Prerequisites

1 Prerequisites

- A host machine with Ubuntu 18.04.5 LTS installed
- An AWS account (register an account at [2]). There may be a small fee for using AWS IoT services
- PSoC 6 Wi-Fi BT Prototyping Kit (CY8CPROTO-062-4343W) (Figure 1)
- IRIDIUM9670 TPM2.0 (Figure 2)
- Familiar with the procedure of building and programming a PSoC 6 using ModusToolbox [11]
- Familiar with the project mtb-example-anycloud-mqtt-client [3]

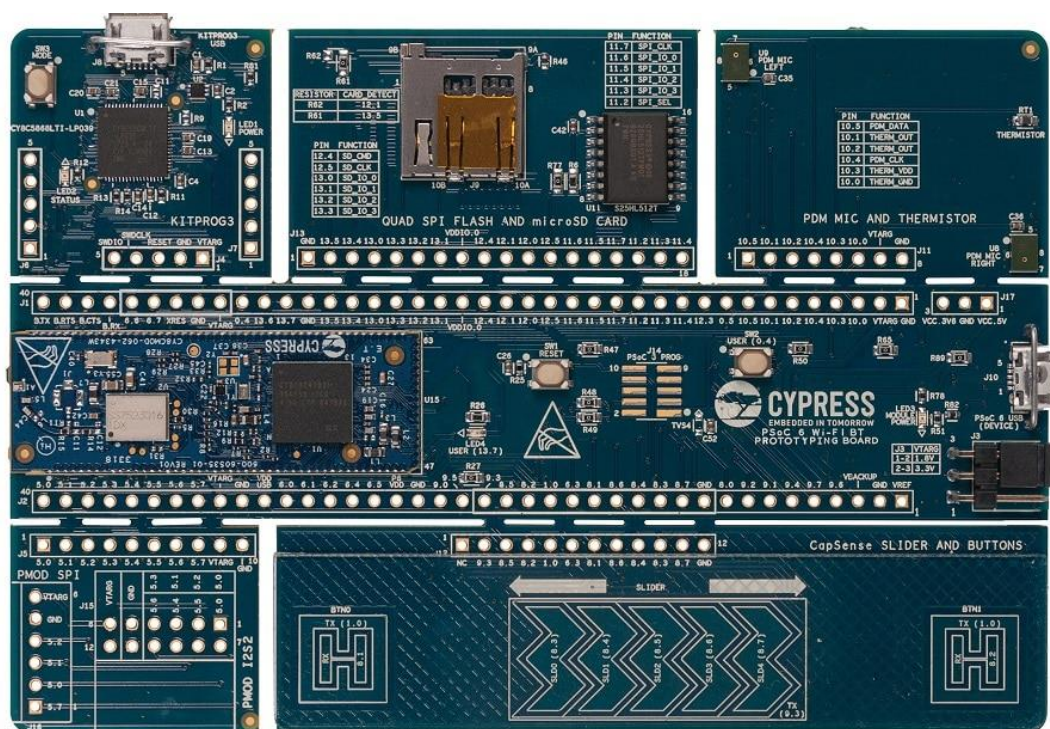


Figure 1 PSoC 6 Wi-Fi BT Prototyping Kit (CY8CPROTO-062-4343W)



Figure 2 Iridium9670 TPM2.0

2 Host Machine Setup

Download ModusToolbox v2.2.0 [1] and complete the installation by following the guide found at ModusToolbox/README_2.2.0.

Install dependencies.

Code Listing 1

```
001 $ sudo apt install make git minicom curl jq
```

First time Git setup, insert your username and email.

Code Listing 2

```
001 $ git config --global user.name "your name"
002 $ git config --global user.email your-email@example.com
```

Download the software package from GitHub (Table 1).

Code Listing 3

```
001 $ cd ~
002 $ git clone https://github.com/Infineon/psoc6-aws-iot-optiga-tpm
```

Table 1 GitHub repository

Folder	Description
COMPONENT_TPM	The TPM software package is based on: <ul style="list-style-type: none"> • TIS from Linux version 5.8.7 [4] • TSS from tpm2-tss version 3.0.0 [5]
awscli	AWS CLI scripts: <ul style="list-style-type: none"> • 1_create_awsiot_thing.sh • 2_convert_to_clanguage.sh • 3_clean_awsiot_thing.sh
mtb_project	A ready to use ModusToolbox project. Uncompress the project before use. <p>Code Listing 4</p> <pre>001 \$ cat mtb_project.tar.gz.* tar -xzf -</pre>
patches	Patches to apply: <ul style="list-style-type: none"> • anycloud_mqtt_client.patch • mbedtls.patch • secure-sockets.patch

3 Project Setup

3.1 TPM to PSoC 6 Connection

Connect TPM board to PSoC 6 Wi-Fi BT prototyping kit. For long connectors, consider adding series termination resistors at TPM board R3 (MOSI), R4 (MISO), and R5 (CLK).

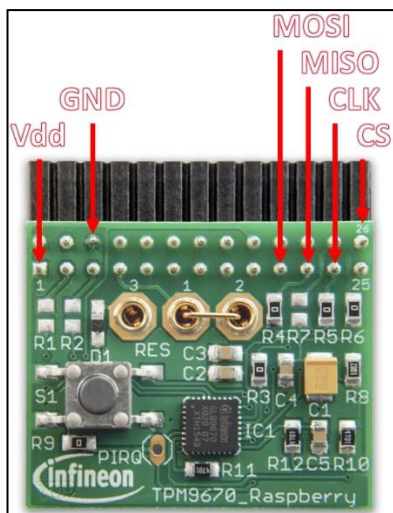


Figure 3 Iridium9670 TPM2.0 pins

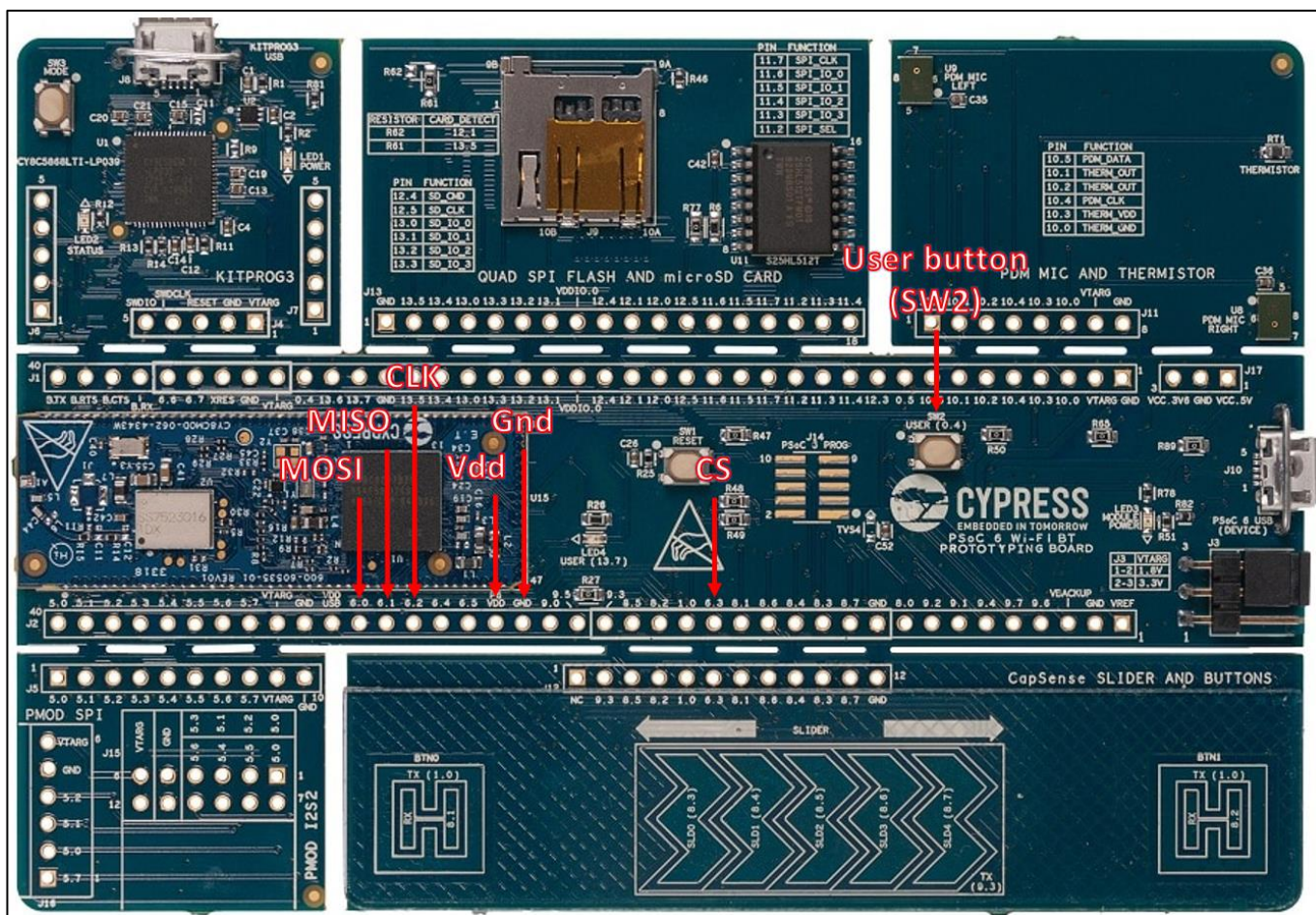


Figure 4 PSoC 6 Wi-Fi BT prototyping kit GPIO layout, 6.0: MOSI, 6.1: MISO, 6.2: CLK, 6.3: CS

Project Setup**3.2 AWS CLI Setup**

Install AWS CLI. The official guide can be found at [6].

Code Listing 5

```
001      $ cd ~
002      $ mkdir awscli
003      $ cd awscli
004      $ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64-
        2.0.30.zip" -o "awscliv2.zip"
005      $ unzip awscliv2.zip
006      $ sudo ./aws/install
```

Set AWS account credential and region (for more information visits [12][13]).

Code Listing 6

```
001      $ aws configure
002      AWS Access Key ID [None]: ...
003      AWS Secret Access Key [None]: ...
004      Default region name [None]: ...
005      Default output format [None]: json
```

Check if the configuration is done correctly.

Code Listing 7

```
001      $ cat ~/.aws/credentials
002      [default]
003      aws_access_key_id = ...
004      aws_secret_access_key = ...
005      $ cat ~/.aws/config
006      [default]
007      region = ...
008      output = json
```

3.3 Serial Over USB

Connect a PSoC 6 to a host machine and check if a serial port is connected (e.g. /dev/ttyACM0).

Code Listing 8

```
001      $ dmesg | grep tty
```

Launch a terminal software (e.g. minicom) to establish serial communication with a PSoC 6.

Code Listing 9

```
001      $ sudo minicom -b 115200 --device /dev/ttyACM0
```

3.4 PSoC 6 Project Setup

Complete the steps in Table 2 to create a ModusToolbox project. There are 2 options:

Project Setup

Option 1. **[Preferred]** Skip step 1-6 by using the source from ~/psoc6-aws-iot-optiga-tpm/mtb_project. Follow the instruction in Table 1 to uncompress the project. Move both folders (anyccloud_mqtt_client and mtb_shared) to ~/mtw. Complete step 2 before proceeding to step 7-8.

Option 2. Complete all the steps in Table 2.

Note: Option 2 may fail due to the design of PSoC 6 project creator. The project creator does not provide an option to pull specific version of software dependencies; instead, the latest version will always be pulled. This will make patches unusable or cause unpredictable behaviour in the future. This option is documented to provide a general guideline on integration of TPM for developer interest.

Table 2 Project setup

Step	Description
1.	<p>Create a version specific project using project-creator-cli.</p> <p>Code Listing 10</p> <pre> 001 \$ mkdir ~/mtw 002 \$ ~/ModusToolbox/tools_2.2/project-creator/project-creator-cli --board-uri https://github.com/cypresssemiconductorco/TARGET_CY8CPROTO- 062-4343W --board-id CY8CPROTO-062-4343W --board-commit release-v2.0.0 --app-uri https://github.com/cypresssemiconductorco/mtb-example- anycloud-mqtt-client --app-id mtb-example-anycloud-mqtt- client --app-commit release-v2.0.0 --user-app-name anycloud_mqtt_client --target-dir ~/mtw </pre>
2.	<p>Launch the ModusToolbox Eclipse IDE and set the workspace to ~/mtw.</p> <p>Code Listing 11</p> <pre> 001 \$ ~/ModusToolbox/ide_2.2/eclipse/ModusToolbox </pre> <p>Import the project by navigating to File > Import > ModusToolbox > ModusToolbox Application Import > set project location to ~/mtw/anyccloud_mqtt_client.</p> <p>Two projects (mtb_shared and anycloud_mqtt_client) should appear under the Project Explorer tab.</p>
3.	<p>Copy the folder COMPONENT_TPM (check Table 1) to the root directory of the project anycloud_mqtt_client.</p>
4.	<p>Apply the patch (mbedtls.patch) to the library mbedtls (mbedtls-2.16.7) to extend the support for MBEDTLS_PK_RSA_ALT in CSR and CRT generation.</p> <p>Code Listing 12</p> <pre> 002 \$ cd ~/mtw/mtb_shared/mbedtls/mbedtls-2.16.7 003 \$ git am --signoff < ~/psoc6-aws-iot-optiga- tpm/patches/mbedtls.patch </pre>
5.	<p>Apply the patch (secure-sockets.patch) to the library secure-sockets (release-v2.1.0) to enable TLS operation using TPM-based RSA key.</p> <p>Code Listing 13</p> <pre> 001 \$ cd ~/mtw/mtb_shared/secure-sockets/latest-v2.X </pre>

Project Setup

	<pre>002 \$ git am --signoff < ~/psoc6-aws-iot-optiga-tpm/patches/secure-sockets.patch</pre>
6.	<p>Apply the patch (anyccloud_mqtt_client.patch) to the project anycloud_mqtt_client (release-v2.0.0) for the following reasons.</p> <ul style="list-style-type: none"> • Include COMPONENT_TPM into the build (Makefile) • Insert TPM operation in MQTT task (mqtt_task.c) • Configure MQTT connection (mqtt_client_config.h) • Configure Mbed TLS to accept TPM key (mbedtlsls_user_config.h) <p>Code Listing 14</p> <pre>001 \$ cd ~/mtw/anyccloud_mqtt_client 002 \$ git am --signoff < ~/psoc6-aws-iot-optiga-tpm/patches/anyccloud_mqtt_client.patch</pre>
7.	<p>Setup Wi-Fi connectivity (~/mtw/anyccloud_mqtt_client/configs/wifi_config.h).</p> <ul style="list-style-type: none"> • Set WIFI_SSID and WIFI_PASSWORD
8.	<p>Some MQTT parameters (~/mtw/anyccloud_mqtt_client/configs/mqtt_client_config.h) require further customization for AWS onboarding to work.</p> <ol style="list-style-type: none"> 1. Set ROOT_CA_CERTIFICATE to Amazon Root CA [7] 2. Set MQTT_BROKER_ADDRESS according to an individual AWS account (e.g., "e23ch0zch30v6h-ats.iot.ap-southeast-1.amazonaws.com"). The endpoint can be obtained by using AWS CLI. <p>Code Listing 15</p> <pre>001 \$ aws iot describe-endpoint --endpoint-type iot:Data-ATS</pre> <ol style="list-style-type: none"> 3. Remember to complete all afore configurations before proceeding. <ol style="list-style-type: none"> i. Build the project and program a PSoC 6. To learn about ModusToolbox, please visit [11]. ii. Observe the serial terminal for a CSR (Appendix A). The error message "MQTT connection failed" is expected since device is not onboarded yet. iii. Create a file named "tpm.pem.csr". iv. Copy the CSR from the serial terminal to "tpm.pem.csr". A valid CSR must start with "-----BEGIN CERTIFICATE REQUEST-----" and end with "-----END CERTIFICATE REQUEST-----". v. Move "tpm.pem.csr" to ~/psoc6-aws-iot-optiga-tpm/awscli/csr/. vi. Run the script ~/psoc6-aws-iot-optiga-tpm/awscli/1_create_awsiot_thing.sh to perform onboarding and obtain a client certificate (tpm.pem.crt). vii. Run the script ~/psoc6-aws-iot-optiga-tpm/awscli/2_convert_to_clanguage.sh to convert the "tpm.pem.crt" to a C formatted client certificate printed on the terminal. viii. Open ~/mtw/anyccloud_mqtt_client/configs/mqtt_client_config.h, find CLIENT_CERTIFICATE then change it to the C formatted client certificate. ix. Rebuild the project and program a PSoC 6. It is now ready to connect to AWS IoT Core. <p><i>Note: Run the script ~/psoc6-aws-iot-optiga-tpm/awscli/3_clean_awsiot_thing.sh to deregister a device from AWS IoT Core.</i></p>

3.5 PSoC 6 in Operation

After completing section 3.4, power-on the PSoC 6 and the following message should appear on the serial terminal.

Code Listing 16

```

001  =====
002  CE229889 - AnyCloud Example: MQTT Client
003  =====
004
005  tpm_task : started
006  tpmt_api : found key handle 8100beef
007  tpmt_api : found key handle 8100cafe
008  tpmt_api : TPM is already provisioned, no work to be done
009  tpmt_api : TPM open encrypted session
010  tpmt_api : TPM close encrypted session
011  tpmt_api : TPM read public key of handle: 0x8100cafe
012  mbedtls_tpmt_api : mbedtls_tpmt_genCsr invoked, TPM generates CSR
    (Certificate Sign Request)
013  mbedtls_tpmt_api : mbedtls_tpmt_platfrom_entropy_poll invoked
014  tpmt_api : TPM open encrypted session
015  tpmt_api : TPM close encrypted session
016  tpmt_api : TPM get random
017  tpmt_api : TPM open encrypted session
018  tpmt_api : TPM close encrypted session
019  tpmt_api : TPM read public key of handle: 0x8100cafe
020  mbedtls_tpmt_api : mbedtls_tpmt_rsa_sign_func invoked
021  tpmt_api : TPM open encrypted session
022  tpmt_api : TPM close encrypted session
023  tpmt_api : TPM signing using RSA key handle 0x8100cafe
024  mbedtls_tpmt_api : generated CSR encoded in PEM format
025  CSR PEM:
026  -----BEGIN CERTIFICATE REQUEST-----
027  MIIC4DCCAcgCAQAwgZoxCzAJBgNVBAYTA1NHMQswCQYDVQQIDAjTRzEZMBcGA1UE
028  BwwQOCBLYWxsYW5nIFNlY3RvcjEeMBwGA1UECgwVSW5maW5lb24gVGJjaG5vbG9n
029  aWVzMSEwHwYDVQQQLDBhDb25uZWNOZWQgU2VjdXJlIFN5c3RlbXMxIDAeBgNVBAMM
030  F0FXUYBjB1QgVFBNIENlcnRpZmljYXRlMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8A
031  MIIBCgKCAQEAl6gxgHw3bwKBZ5D3Wk9N5M65qu9uGT/uhEooeTGD6YtNi56mPM6C
032  A36tXnyMK2N7N07G4e2JhhLnz/+5HvdlOMT7rjv+7HA1eEm2n5RkBLTg+u8fArk/
033  +rjxxCh1jk3MZpMoC1lBve09KpBIwKx7Q6jRbZn5JFE8jEMpfm208VbmA1Y0OVj0
034  +Ow40SvC00hiNw52gmmQM9fXYcWng6MiEk8IYauBDqKrUEOvXQdTfJzls5qe0S
035  9xKT5unnBXzt3sQtFvuziz1lwo+LGgvJk38UOGyrBJebsXJozAa/MJ8+hHX2FI7
036  0bbFvnFZiAi/z7FKcA10KrL6ukaSlzHSDwIDAQABoAAwDQYJKoZIhvcNAQELBQAD
037  ggEBACJrqH+VSiMH8ulrCOvt+Jk5HIWYUggj4uHwbrOTfCuNKKsVoURipsHVw4Nv
038  t5J3/K3hf7sUX+ZM22MLWGJSZQM+6KqXscBfG2jwRreth9iXu6mp/ly9pIvN4Hct
039  f1jOFKY4j2AOc6F1YikeNLRGi/lMgzB9SLTuWl8rxW1lN4krz7+/eIcch6qNAYdn
040  DUdq/oN75e03RUgrY6p66DDE8OHbUEExHKdAJ0NK3kvwe6shPQCJFMeWjt9GftXil
041  slMezCsG5cf8Q4Wm3o/epxo7lMXJZW3GuL6nbsjQoQYZeAtWkMDJtz76rRC6jViG
042  ECGpjAjrwpTJJiVJ5MeeRk8QTzU=
043  -----END CERTIFICATE REQUEST-----
044
045  WLAN MAC Address : 00:9D:6B:9C:E8:DD
046  WLAN Firmware : wl0: Dec 13 2020 18:58:44 version 7.45.98.110
    (r727676 CY) FWID 01-aefd28c3

```

Code Listing 16

```

047 WLAN CLM          : API: 12.2 Data: 9.10.39 Compiler: 1.29.4
    ClmImport: 1.36.3 Creation: 2020-12-13 18:48:35
048 WHD VERSION       : v1.93.0 : v1.93.0 : GCC 9.3 : 2020-12-21
    13:24:03 +0530
049
050 Wi-Fi Connection Manager initialized.
051 Connecting to Wi-Fi AP 'my-ap'
052 Successfully connected to Wi-Fi network 'my-ap'.
053 IPv4 Address Assigned: 192.168.0.130
054
055 IoT SDK initialized successfully.
056 Network stack initialized successfully.
057 MQTT library initialization successful.
058
059 MQTT client 'psoc6-mqtt-client10905' connecting to MQTT broker
    'e23ch0zch30v6h-ats.iot.ap-southeast-1.amazonaws.com'...
060 tpmt_api : TPM open encrypted session
061 tpmt_api : TPM close encrypted session
062 tpmt_api : TPM read public key of handle: 0x8100cafe
063 mbedtls_tpmt_api : mbedtls_tpmt_rsa_sign_func invoked
064 tpmt_api : TPM open encrypted session
065 tpmt_api : TPM close encrypted session
066 tpmt_api : TPM signing using RSA key handle 0x8100cafe
067 MQTT connection successful.
068
069 MQTT client subscribed to the topic 'ledstatus' successfully.
070
071 Press the user button (SW2) to publish "TURN ON"/"TURN OFF" on
    the topic 'ledstatus'...

```

Once the MQTT connection is established, press on the user button SW2 (Figure 4) to publish a message. On the AWS IoT Core console, click on the “Test” option on the left menu to enter the MQTT test client (Figure 5), subscribe to the topic “ledstatus” to view all incoming messages or to publish messages to the PSoC 6.

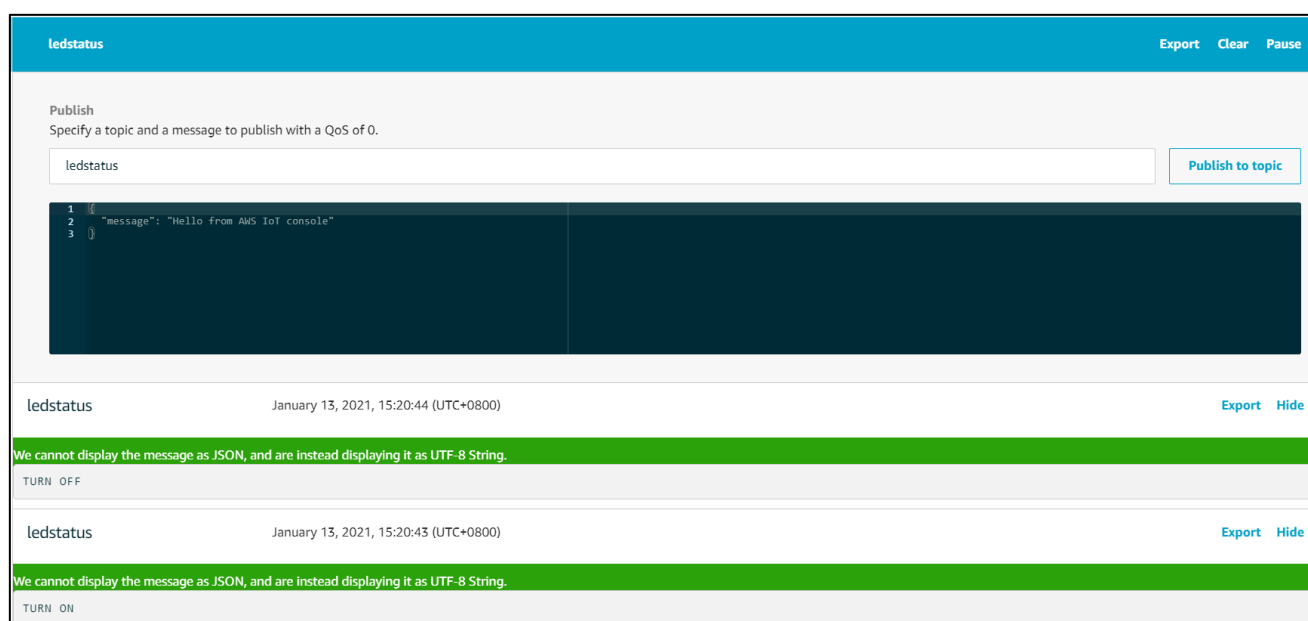


Figure 5 MQTT test client

Appendices

Appendix A: TPM Provisioning

TPM provisioning process will be invoked if required keys (key handles 0x100beef and 0x8100cafe) are not found in a TPM. Connection to the AWS MQTT is expected to fail because PSoC 6 is not onboarded yet.

Code Listing 17

```

001  =====
002  CE229889 - AnyCloud Example: MQTT Client
003  =====
004
005  tpm_task : started
006  tpmt_api : TPM force clear
007  tpmt_api : TPM take ownership
008  tpmt_api : TPM Primary RSA key created as transient handle
009  tpmt_api : TPM transient (0x80000000) moved to persistent
010          (0x8100beef)
011  tpmt_api : TPM clear transient handle: 0x80000000
012  tpmt_api : TPM leaf key created as transient handle
013          (0x8100cafe)
014  tpmt_api : TPM clear transient handle: 0x80000001
015  tpmt_api : TPM provisioning completed
016  tpmt_api : TPM open encrypted session
017  tpmt_api : TPM close encrypted session
018  tpmt_api : TPM read public key of handle: 0x8100cafe
019  mbedtls_tpmt_api : mbedtls_tpmt_genCsr invoked, TPM generates CSR
020          (Certificate Sign Request)
021  mbedtls_tpmt_api : mbedtls_tpmt_platfrom_entropy_poll invoked
022  tpmt_api : TPM open encrypted session
023  tpmt_api : TPM close encrypted session
024  tpmt_api : TPM get random
025  tpmt_api : TPM open encrypted session
026  tpmt_api : TPM close encrypted session
027  tpmt_api : TPM read public key of handle: 0x8100cafe
028  mbedtls_tpmt_api : mbedtls_tpmt_rsa_sign_func invoked
029  tpmt_api : TPM open encrypted session
030  tpmt_api : TPM close encrypted session
031  tpmt_api : TPM signing using RSA key handle 0x8100cafe
032  mbedtls_tpmt_api : generated CSR encoded in PEM format
033  CSR PEM:
034  -----BEGIN CERTIFICATE REQUEST-----
035  MIIC4DCCAQgCAQAwZoxCzAJBgNVBAYTA1NHMQswCQYDVQQLIDAJTRzEZMBcGA1UE
036  BwwQOCBLYWxsYW5nIFNlY3RvcjEeMBwGA1UECgwVSW5maW5lb24gVG9jaG5vbG9n
037  aWVzMSEwHwYDVQQLDBhDb25uZW50ZWQgU2VjdXJlIFN5c3RlbXMxIDAeBgNVBAMM
038  F0FXUyBJb1QgVFBNIENlcnRpZmljYXRlMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8A
039  MIIBCgKCAQEAl6gxgHw3bwKBZ5D3Wk9N5M65qu9uGT/uhEooeTGD6YtNi56mPM6C
040  A36tXnyMK2N7N07G4e2JhhLnz/+5HvdlOMT7rjv+7HA1eEm2n5RkBLTg+u8fArk/
041  +rjjxCh1jk3MZpMoC1lBve09KpBIwx7Q6jRbZn5JFE8jEMpfm208VbmAlY0OVj0
042  +Ow40SvC00hiNw52gmmQM9fXYcWng6MiEk8IYauBDqKrUEOvXQdTfpfJzls5qe0S
043  9xKT5unnBXzt3sQtFvuziz1lwo+LGgvJk38UOGyrBJebsXJozAa/MJ8+hHX2FI7
044  0bbFvnFZiAi/z7FKCa10KrL6ukaSlzHSDwIDAQABoAAwDQYJKoZIhvcNAQELBQAD

```

Code Listing 17

```
043 ggEBACJrqH+VSiMH8ulrCOVt+Jk5HIWyUggj4uHwbrOTfCuNKKsVoURipsHVw4Nv
044 t5J3/K3hf7sUX+ZM22MLWGJSZQM+6KqXscBfG2jwRreth9iXu6mp/1Y9pIvN4HCt
045 f1jOFKY4j2AOc6F1YikeNLRGi/1MgzB9SLTuWl8rxW1lN4krz7+/eIcch6qNAYdn
046 DUdq/oN75e03RUgrY6p66DDE8OHbUExHKdAJ0NK3kvwe6shPQCJFMeWjt9GftXil
047 slMezCsG5cf8Q4Wm3o/epxo7lMXJZW3GuL6nbsjQoQYzEAtWkMDJtz76rRC6jViG
048 ECGpjAjrwpTJJiVJ5MeeRk8QTzU=
049 -----END CERTIFICATE REQUEST-----
050
051 WLAN MAC Address : 00:9D:6B:9C:E8:DD
052 WLAN Firmware : wl0: Dec 13 2020 18:58:44 version 7.45.98.110
(r727676 CY) FWID 01-aefd28c3
053 WLAN CLM : API: 12.2 Data: 9.10.39 Compiler: 1.29.4
ClmImport: 1.36.3 Creation: 2020-12-13 18:48:35
054 WHD VERSION : v1.93.0 : v1.93.0 : GCC 9.3 : 2020-12-21
13:24:03 +0530
055
056 Wi-Fi Connection Manager initialized.
057 Connecting to Wi-Fi AP 'my-ap'
058 Successfully connected to Wi-Fi network 'my-ap'.
059 IPv4 Address Assigned: 192.168.0.130
060
061 IoT SDK initialized successfully.
062 Network stack initialized successfully.
063 MQTT library initialization successful.
064
065 MQTT client 'psoc6-mqtt-client18041' connecting to MQTT broker
'e23ch0zch30v6h-ats.iot.ap-southeast-1.amazonaws.com'...
066 tpmt_api : TPM open encrypted session
067 tpmt_api : TPM close encrypted session
068 tpmt_api : TPM read public key of handle: 0x8100cafe
069 mbedtlsl_tpmt_api : mbedtlsl_tpmt_rsa_sign_func invoked
070 tpmt_api : TPM open encrypted session
071 tpmt_api : TPM close encrypted session
072 tpmt_api : TPM signing using RSA key handle 0x8100cafe
073 [ERROR][NET][lu] cy_socket_connect failed
074
075 [ERROR][MQTT][lu] Failed to create network connection: 1
076 [ERROR][MQTT][lu] Failed to establish new MQTT connection, error
NETWORK ERROR.
077 MQTT connection failed with error 'NETWORK ERROR'!
078
079 Disconnected from the Wi-Fi AP!
```

References

References

- [1] <https://www.cypress.com/documentation/software-and-drivers/modustoolbox-software-archives>
- [2] <https://aws.amazon.com/free>
- [3] <https://github.com/cypresssemiconductorco/mtb-example-anycloud-mqtt-client>
- [4] <https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/tag/?h=v5.8.7>
- [5] <https://github.com/tpm2-software/tpm2-tss/tree/3.0.0>
- [6] <https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2-linux.html>
- [7] <https://www.amazontrust.com/repository/AmazonRootCA1.pem>
- [8] <https://www.infineon.com/cms/en/product/evaluation-boards/iridium9670-tpm2.0-linux/>
- [9] <http://www.infineon.com/tpm>
- [10] <https://trustedcomputinggroup.org/resource/tpm-main-specification/>
- [11] <https://www.cypress.com/documentation/application-notes/an228571-getting-started-psoc-6-mcu-modustoolbox>
- [12] <https://docs.aws.amazon.com/general/latest/gr/aws-sec-cred-types.html>
- [13] https://docs.aws.amazon.com/general/latest/gr/connect_region.html

Revision history

Reference	Description
Revision 1.0, 2021-02-02	
all	Initial version

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2021-02-02

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2021 Infineon Technologies AG.

All Rights Reserved.

Do you have a question about this document?

Email:

csscustomerservice@infineon.com

IMPORTANT NOTICE

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (www.infineon.com).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.