PSOC® 64 ARROW IOT SECURITY VIRTUAL WORKSHOP DEVELOPMENT KIT

Pre-Work GUIDE

Abstract

The intended audience for this document is for someone with knowledge in programming and familiar with the PSoC 64 secure microcontroller. This guide will go over how to provision the PSoC 64 Secure AWS IoT Pioneer Kit (CY8CKIT-064S0S2-4343W) with the ModusToolbox IDE and Amazon FreeRTOS.

Any questions please contact us at

psoc64@arrow.com

TABLE OF CONTENTS

1. Introduction 2
Materials 2
PSoC 64 Provisioning Tools 2
Check the appendix 2
2. Prerequisites 3
Install the latest version of ModusToolbox 3
Python Setup in Windows 4
Setup a project folder 5
Freertos download 7
Create an AWS Account 7
AWS First Steps simplified 8
Create IAM User 9
Set Permissions 12
Verify Your IAM User 15
3. Provisioning the PSoC 64 16
4. AWS Credential Setup 19
Create a Policy 19
Register the Device 21
5. Download FreeRTOS 23
6. Provision the board 24
7. Configure the AWS Demos 30
8. Configuring Your AWS IoT Endpoint and WiFi Credentials 31
9. Formatting the AWS IoT Credentials 33
10. Build and Run the FreeRTOS Demo 37
11. Monitoring MQTT Messages on AWS 39
12. Appendix 41
ModusToolbox Installation issues 41
Assembly Instructions for the Kit used in the P64 Security Webinars 41
Tips & Tricks 49
Software Setup 50
Communication with Kit 51
Potential Provisioning Failures 52
Potential Modustoolbox build failure 53

Potential Running Failures 54
13. Revision History 54

1. INTRODUCTION

MATERIALS

- Workshop Kit:
  - PSoC 64 Secure AWS IoT Pioneer Kit (CY8CKIT-064S0S2-4343W)
  - IM69D130 Microphone Shield2Go (S2GOMEMSMICIM69DTOBO1)
  - DPS368XTSA1 Pressure Shield2Go (S2GOPRESSUREDPS368TOBO1)
  - Arrow PSOC6_IOT_Sensor_Shield
  - 1x43 Press Pin Socket
  - (4) 1x10 Press Pin Post
- Software:
  - ModusToolbox IDE (download)
  - AWS Account (online setup)
  - Tera Term or PuTTY (download))

PSOC 64 PROVISIONING TOOLS

In order to provision the PSoC 64 the Secure Boot Software Development Kit (SDK) provided by Infineon must be used. The Secure Boot SDK is a standalone Python CySecureTools package which contains all the necessary scripts to generate keys, policies, and secure bootloader image for the PSoC 64.

For more information with the methods used for transferring the root of trust (RoT) and injecting the keys and policies into the PSoC 64 please refer to the Secure Boot SDK User Guide.

https://www.cypress.com/documentation/software-and-drivers/psoc-64-secure-mcu-secure-boot-sdk-user-guide

CHECK THE APPENDIX

Please check in Appendix for methods to get around common issues and misunderstandings.

If you have additional tips, tricks, clarifications, or suggestions, please e-mail them to psoc64@arrow.com

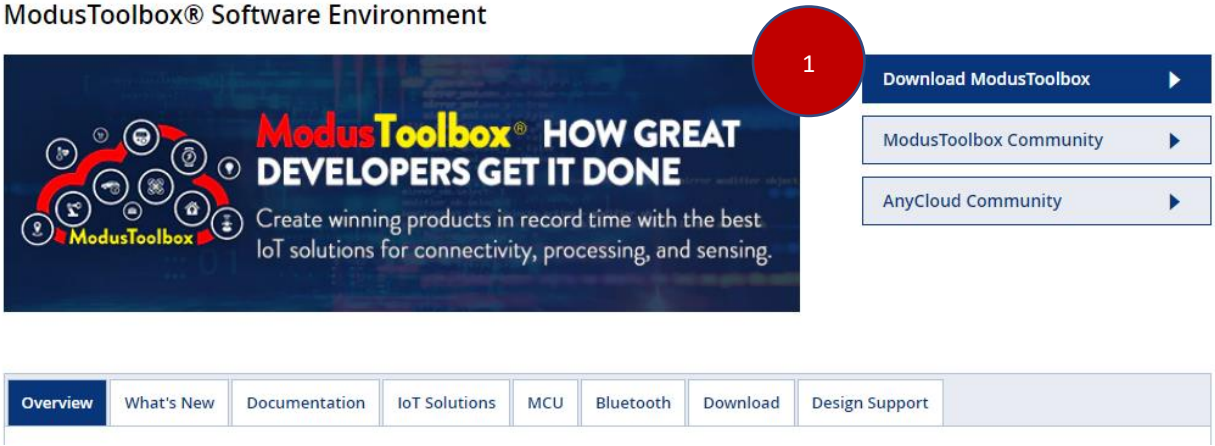
2. PREREQUISITES

This section has instructions for downloading and configuring software tools that aid in the provisioning and development process. It also includes instructions for setting up and configuring Amazon Web Services (AWS)

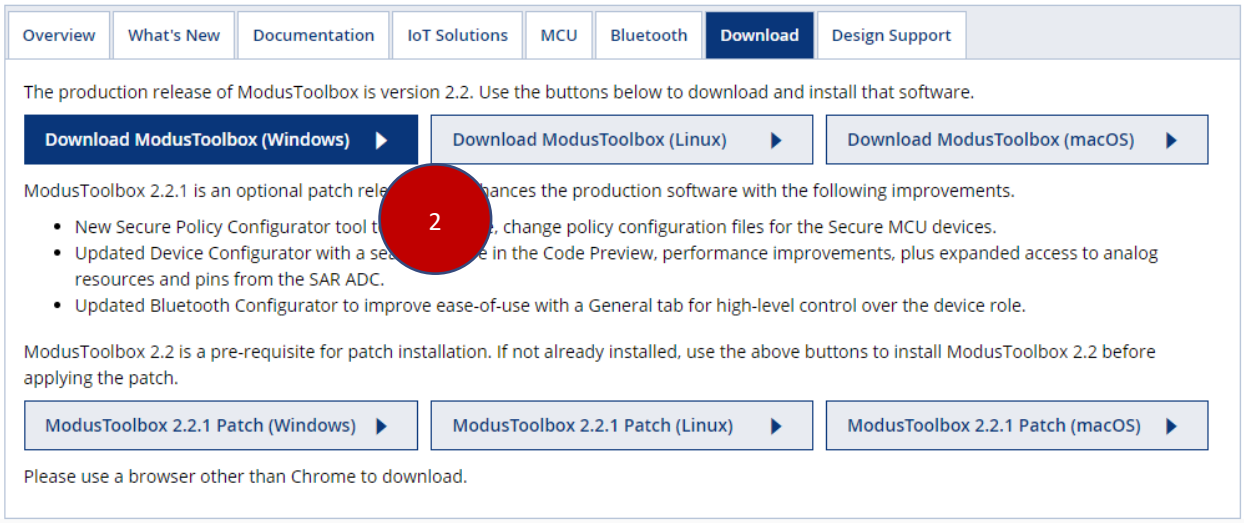
Instructions to assemble the kit are in the Appendix. The kit should be assembled before the workshop on March 18th. Assembly of the kit is NOT required for these pre-work exercises.

INSTALL THE LATEST VERSION OF MODUSTOOLBOX

https://www.cypress.com/products/modustoolbox-software-environment

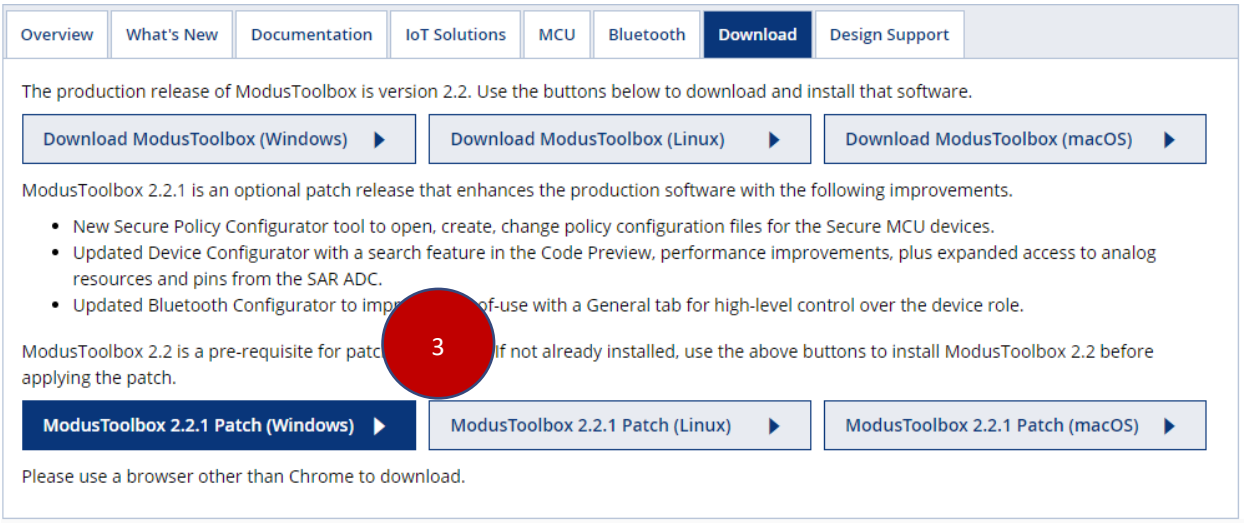


Tip: To use Chrome browser to download, right-click on the download button, copy the link, then past the link into the search bar of a new browser tab.



Tip: If you're not able to login with Admin privileges, refer to 'MODUSTOOLBOX INSTALLATION ISSUES' in the appendix of this document.

Don't forget to download and install the latest update patch as seen below.



REFER TO THE MODUSTOOLBOX INSTALLATION GUIDE FOR DETAILED INSTALLATION INSTRUCTIONS!

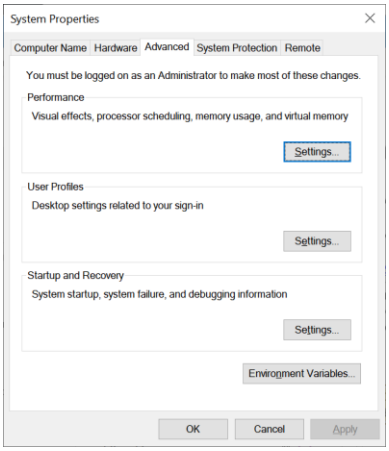
NOTE: IF YOUR MOST CURRENT VERSION OF MODUSTOOLBOX IS BEFORE V2.2, PLEASE UPGRADE TO THE LATEST VERSION OF MODUSTOOLBOX. STARTING WITH MODUSTOOLBOX 2.2, SEVERAL TOOLS THAT SUPPORT SECURITY ARE INCLUDED IN THE MODUSTOOLBOX/TOOLS_2.X DIRECTORY INCLUDING:

- PYTHON 3.7
- CYSECURETOOLS
- LIBUSB

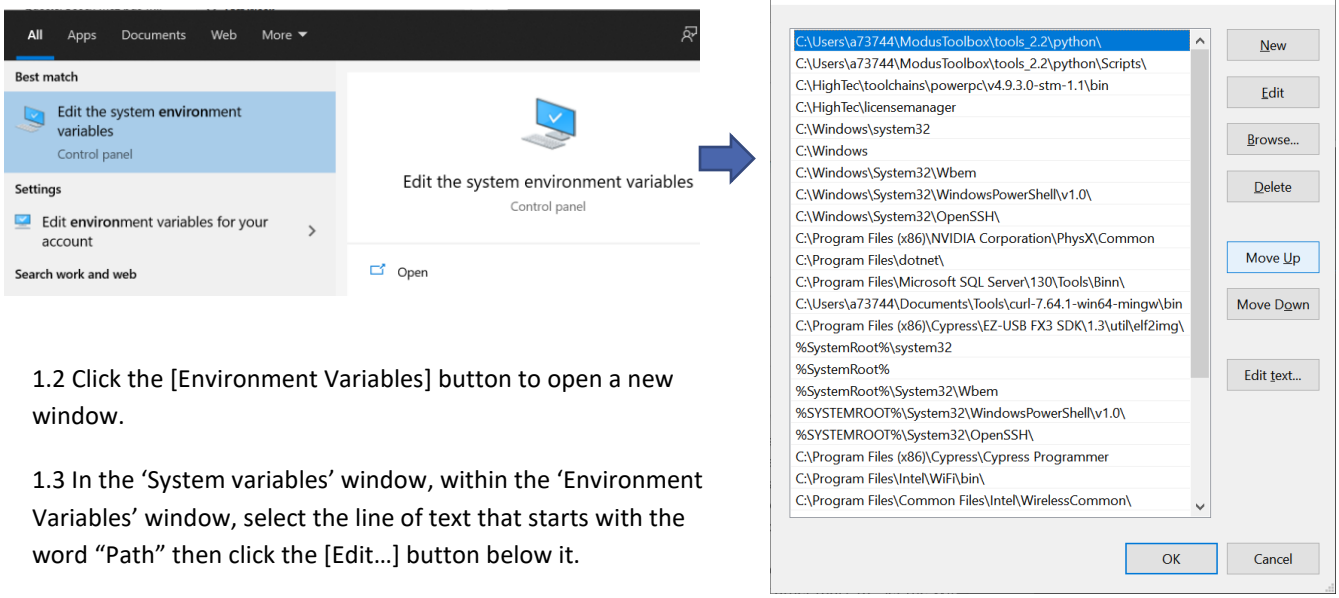
PYTHON SETUP IN WINDOWS

The majority of the exercises in this pre-work run in ModusToolbox or a Modus Shell. However, there is one instance where a batch file is used that calls for a cysecuretools command from a Windows Command Prompt.

Python version 3.4.7 was installed as part of ModusToolbox. Instructions within this pre-work add tools to that installation of Python. To run the same Python from Windows, there are two primary options: a) replicate installation of Python along with all necessary added tools to Windows; b) set an environment variable in Windows to point to Python installed with ModusToolbox. Instructions to set environment variables in Windows 10 are included here:



1.1 From Windows Start Menu, start typing “Environment Variables” until an option to edit system environment variables appears. Open the tool.



1.2 Click the [Environment Variables] button to open a new window.

1.3 In the ‘System variables’ window, within the ‘Environment Variables’ window, select the line of text that starts with the word “Path” then click the [Edit…] button below it.

1.4 In the ‘Edit environment variable’ window that pops up, click the [New] button. Then, in the highlighted line that appears, type the path to Python\Scripts\. Then click the [Move Up] button until the new line is at top of the path.

C:\Users\<user_name>\ModusToolbox\tools_2.2\python\Scripts\

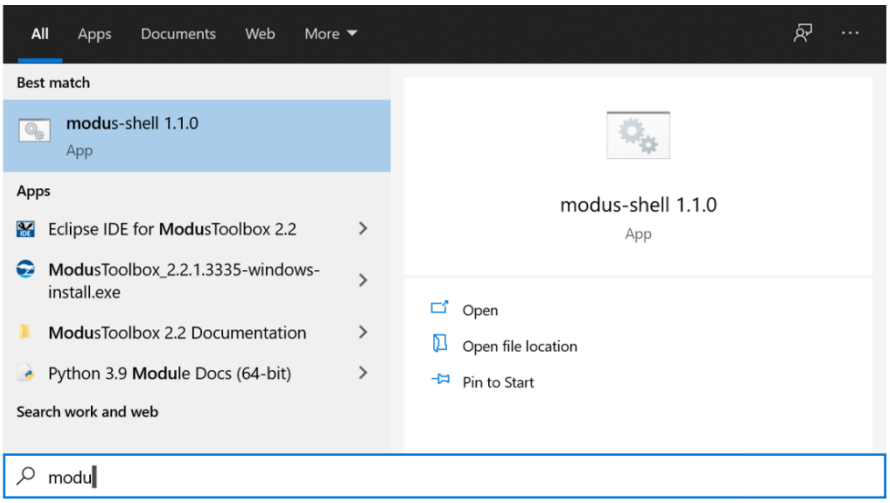
1.5 Click the [New] button again. In the highlighted line that appears, type the path to Python then again click [Move Up] button until the python\ line is now at top.

C:\Users\<user_name>\ModusToolbox\tools_2.2\python\

SETUP A PROJECT FOLDER

Create a folder for a new project near the top level of files on your computer. This new project can be used to hold project materials, a copy of freeRTOS, and be the workspace for ModusToolbox.

1.6 Open a Modus Shell window by typing “Modus Shell” into a Windows Start Menu until a ‘modus-shell…’ app appears, then click on it to open it up.



1.7 When the Modus Shell opens up, type the following commands in sequence:

pwd

cd ..

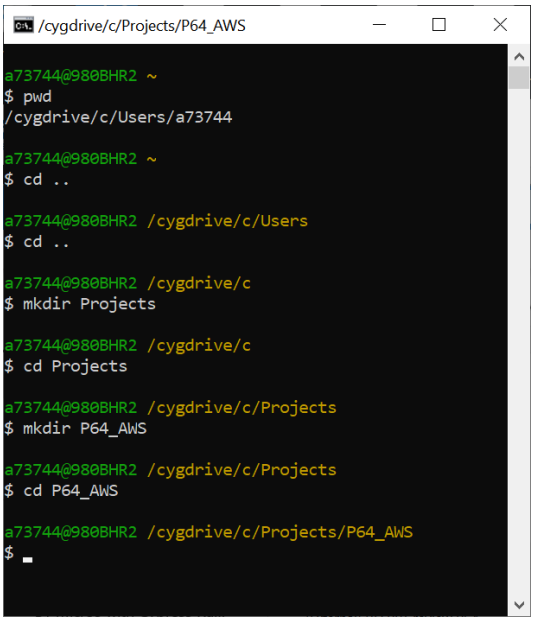
cd ..

mkdir Projects

cd Projects

mkdir P64_AWS

cd P64_AWS



At this point, you've created a project directory which may be called 'P64_AWS' unless you chose a different name.

You then set Modus Shell open within that new project directory.

FREERTOS DOWNLOAD

Installation of freeRTOS may take up to 30 minutes depending on your internet connection. The process to install freeRTOS was moved here to the beginning of this pre-work to allow it time to load while other steps, that don't require freeRTOS, are performed.

1.8 Install version 202007.00 of freeRTOS from an AWS github site by typing or copying the following command into ModusShell within the project directory just created.

```
git clone --branch 202007.00 https://github.com/aws/amazon-freertos --recursive
```

CREATE AN AWS ACCOUNT

This section will walk you through the setup guide to create an AWS account and configure an AWS Identity and Access Management (IAM) user.

NOTE:

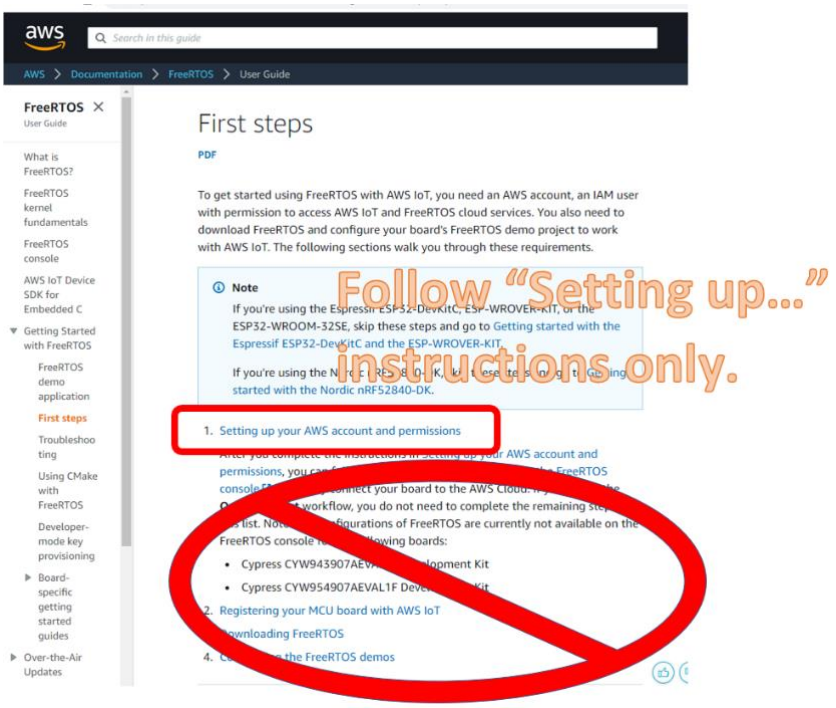
IF YOU DO NOT HAVE AN AWS ACCOUNT, JUMP TO THE "AWS FIRST STEPS SIMPLIFIED" BELOW OR USE THE FIRST SECTION, "SETTING UP YOUR AWS ACCOUNT AND PERMISSIONS" IN "AWS FIRST STEP GUIDE"[1] TO CREATE A FREE[2] AWS ACCOUNT AND AN IAM USER WITH PERMISSIONS.

IF USING THE AWS ON-LINE "FIRST STEPS", STOP AFTER COMPLETING "SETTING UP YOUR AWS ACCOUNT AND PERMISSIONS". THE SECTIONS TO REGISTER YOUR MCU, DOWNLOAD FREERTOS AND CONFIGURE FREERTOS WILL BE DESCRIBED SEPARATELY FOR PSOC 64 SPECIFIC WORKSHOPS. SPECIFIC POLICIES AND VERSIONS OF CODE ARE REQUIRED TO SUPPORT PSOC 64 EXAMPLES THAT WILL BE PRESENTED.

AWS FIRST STEPS SIMPLIFIED

2.1. Using a browser with access to the internet, navigate to AWS First Step Guide @ https://docs.aws.amazon.com/freertos/latest/userguide/freertos-prereqs.html

2.2. Follow the step in "Setting up your AWS account and permissions" or the following instructions here.



The first step under "Setting up…" Redirects to Create and Activate an AWS Account

Creating an AWS account is a step-by-step procedure of providing information including a credit card number to link your access to you financially in case you choose to use more than the free base services or in the unlikely event the account is abused.

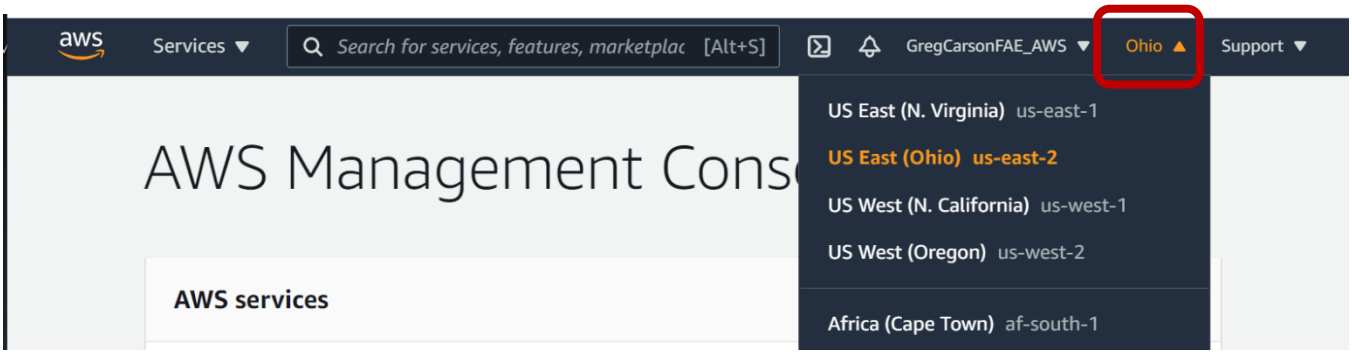
2.3. Record your AWS Account name.

[1] AWS First Step Guide @ https://docs.aws.amazon.com/freertos/latest/userguide/freertos-prereqs.html

[2] AWS accounts require a valid Credit Card number on file for access to advanced features. No charges will be incurred for small activity associated with test and development. Reference Free Tier @ https://aws.amazon.com/free/ or Pricing @ https://aws.amazon.com/pricing/ in AWS documentation for specific information on free tier limits and pricing.

2.4. Select a region. Record the region selected.

The region in the image below is us-east-2 and shows up as Ohio



NOTE: THE SEARCH BOX HAS A SPYGLASS AND THE PHRASE, "SEARCH FOR SERVICES, FEATURES,...."

2.5. Ensure your browser is expanded to reveal the search box.

2.6. Record your AWS Account number – revealed by clicking the down triangle next to your Account name.

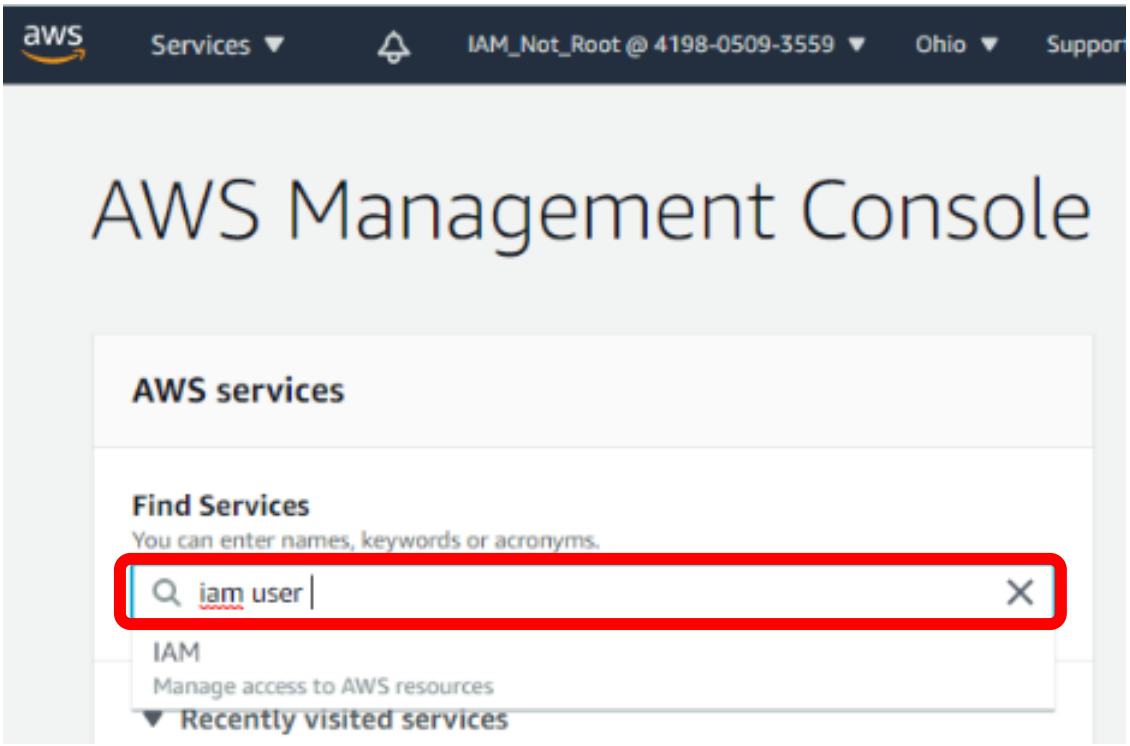
CREATE IAM USER

Once an AWS account is active, Add an IAM User. IAM accounts are described here: IAM User Guide.

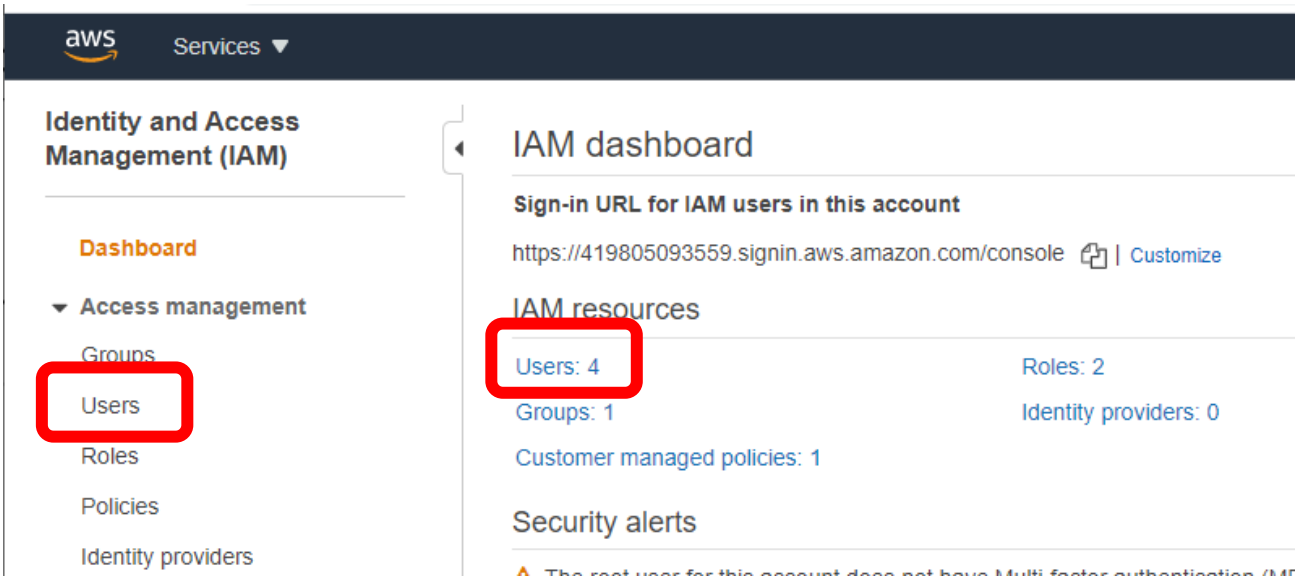
NOTE:
YOUR ROOT AWS ACCOUNT CAN NOT BE RECOVERED IF IT IS COMPROMISED.
USE THE ROOT ACCOUNT ONLY FOR BILLING AND TO CREATE AN ADMINISTRATIVE IAM ACCOUNT.
USE THE ADMIN IAM ACCOUNT TO MAKE AS MANY ADDITIONAL IAM ACCOUNTS AS NEEDED.
IAM ACCOUNTS ARE GIVEN ACCESS TO INFORMATION AND FEATURES OF AWS BY ASSIGNING THEM POLICIES.
ONE POLICY THAT PROVIDES FULL ACCESS IS "ADMINISTRATORACCESS".
ASSIGN THIS POLICY TO YOUR PERSONAL IAM ACCOUNT.

While logged into your AWS account

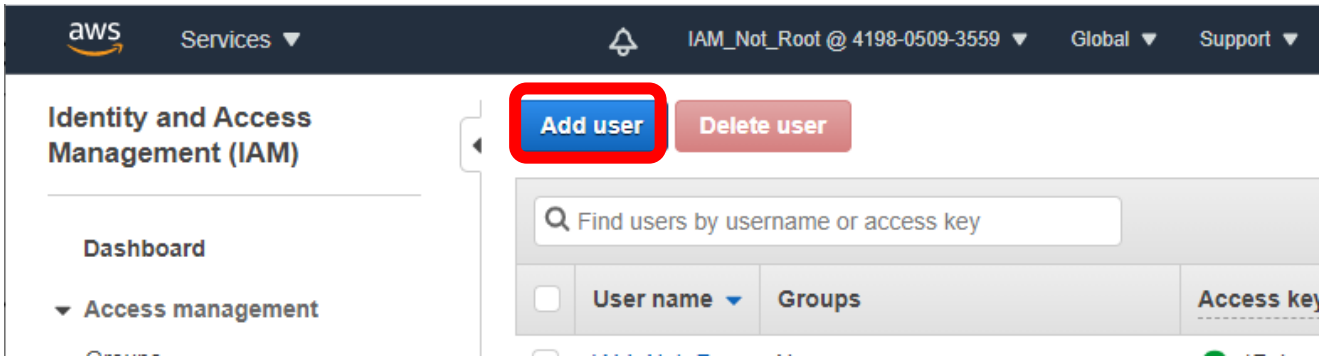
2.7. Search for “iam user” to be redirected to the IAM dashboard.



2.8. Select “Users” either in the left window or within the dashboard.



2.9. Create a new IAM user by selecting "Add User"

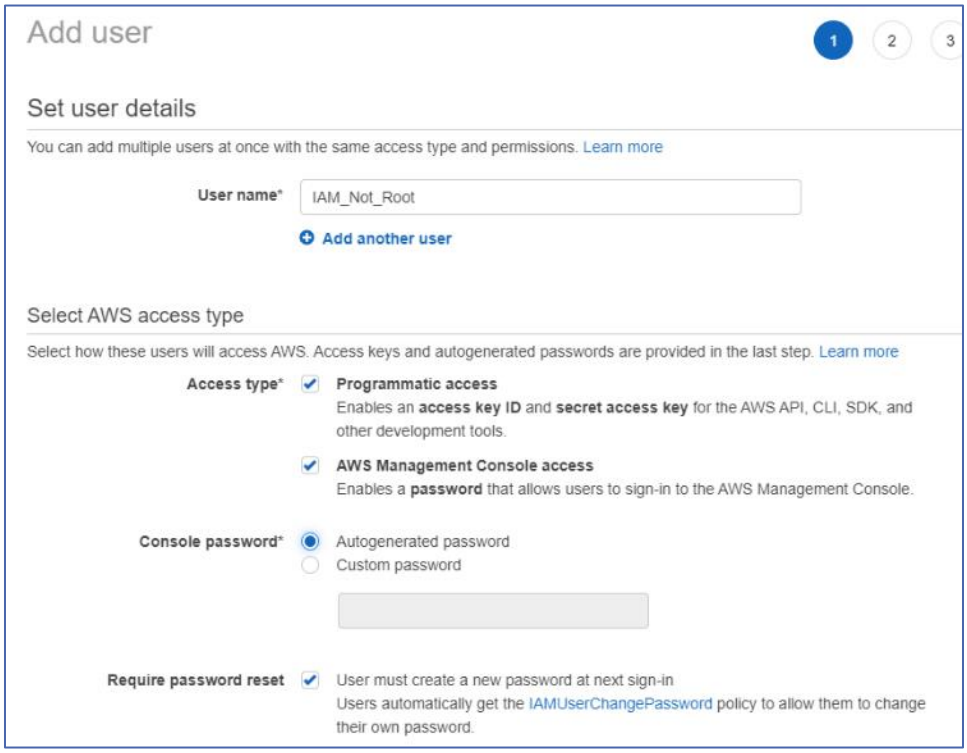


2.11. Create a User Name

It is advised to use a name that signifies the access level it will be given.

2.12. Select Access Type. (Select both for your initial Admin IAM user account.)

- **Programmatic access** = Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK and other development tools.
- **AWS Management Console access** = Enables a **password** that allows users to sign-in to AWS Management Console.



2.13. Click Next Permission

SET PERMISSIONS

2.14. Select “Attach existing policies directly” box

NOTE: YOU CAN USE GROUPS TO CREATE POLICY GROUPS. HOWEVER, IF YOU ARE ONLY CREATING ONE IAM USER, THE “ATTACH EXISTING POLICIES DIRECTLY” OPTION IS QUICKER.

Trivia: As of 11/10/2020 there are 597 policies

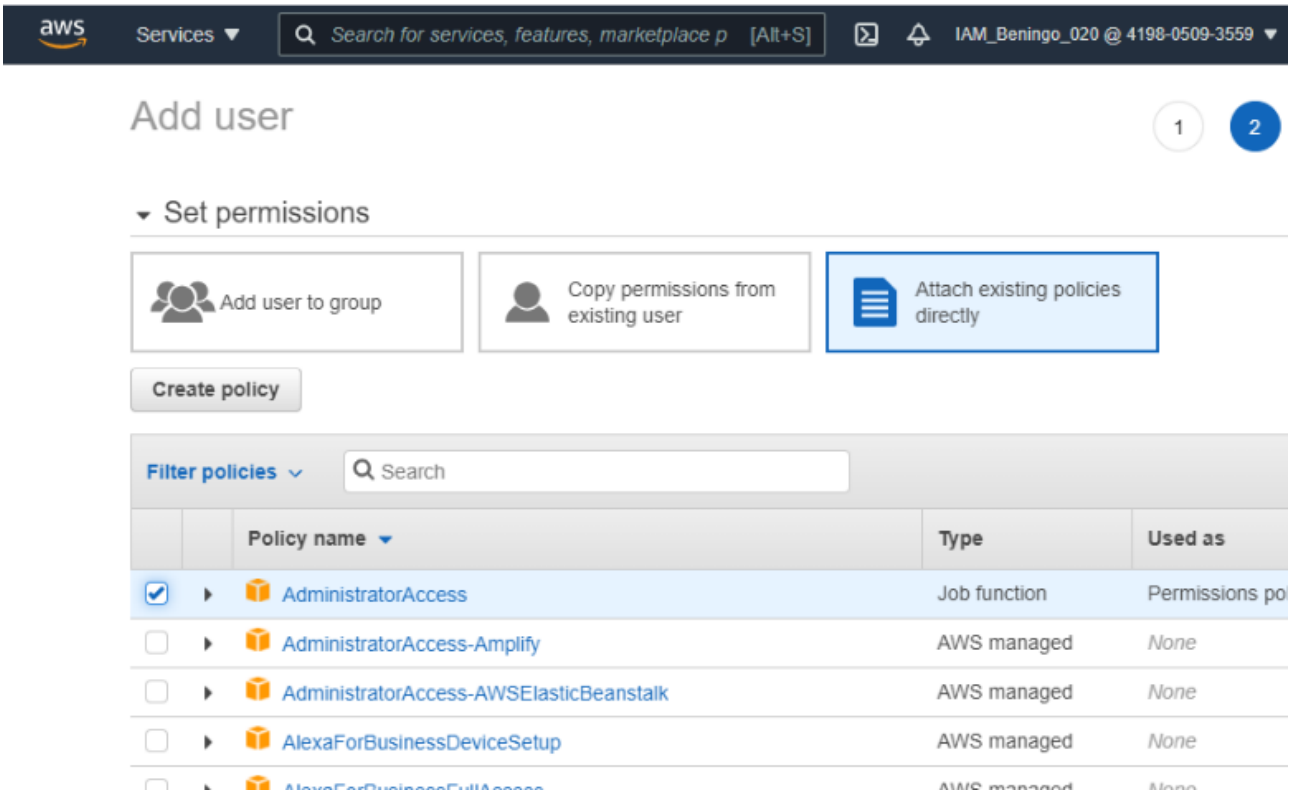
For your personal IAM user assign full access to allow your IAM to do most everything your Root User account could do:

- **AdministratorAccess** provides Full access to all 241 services
- **AmazonFreeRTOSFullAccess** and **AWSIoTFullAccess** are specific policies that will be needed. These two should be subsets of AdministratorAccess.

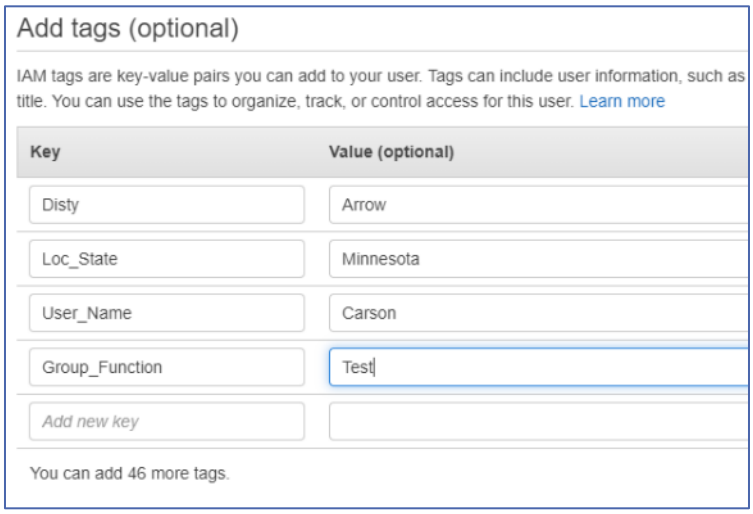
2.15. Click the check box to the left of each desired permission

2.16. Type the names of additional policies in the search box then click the box of additional policies to add.

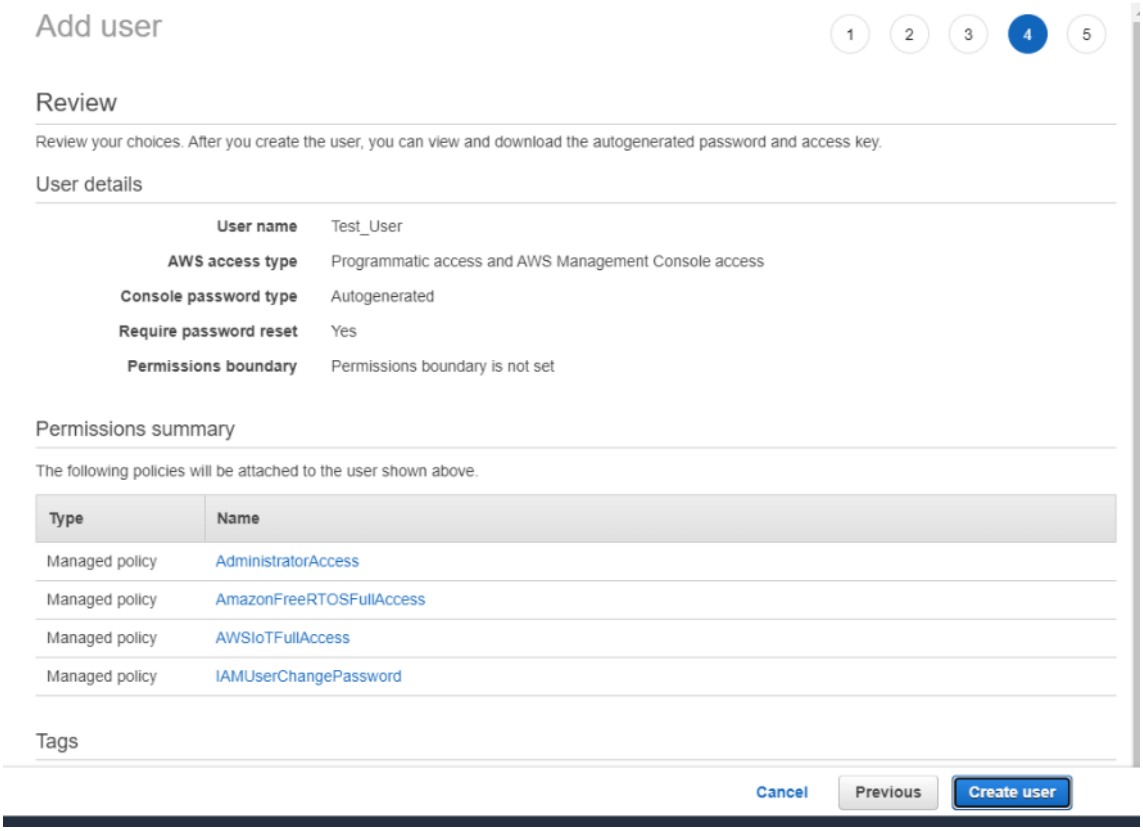
2.17. Click the [Next: Tags] button in the bottom right.



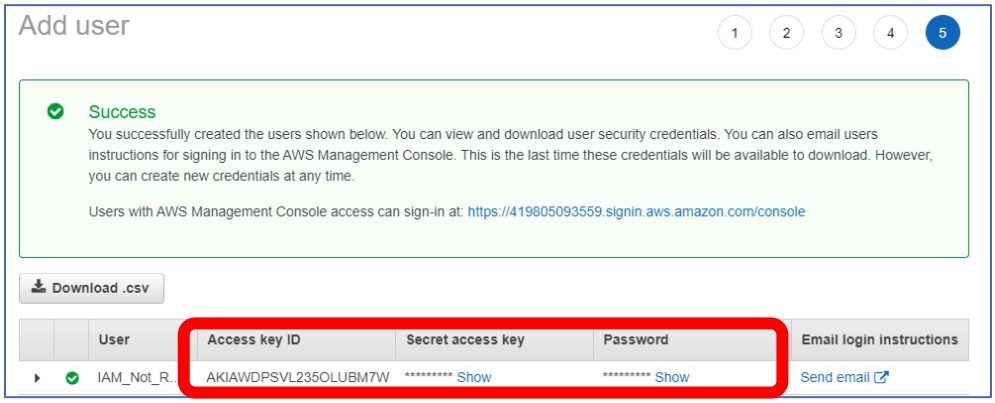
2.18. No tags are needed. To skip this just click **[Next: Review]**

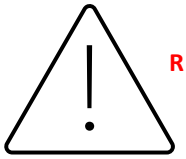


2.19. Verify the information is correct then click **[Create user]**



2.20. Click **[Download.csv]** to record your IAM users' Access key ID, Secret access key and Password.





RECORD THE KEY AND PASSWORD INFORMATION.

THIS IS THE ONLY TIME IT WILL BE REVEALED.

If you lose the Key and Password, you can delete the IAM user account and create a new one.

2.21. Click "Send email"

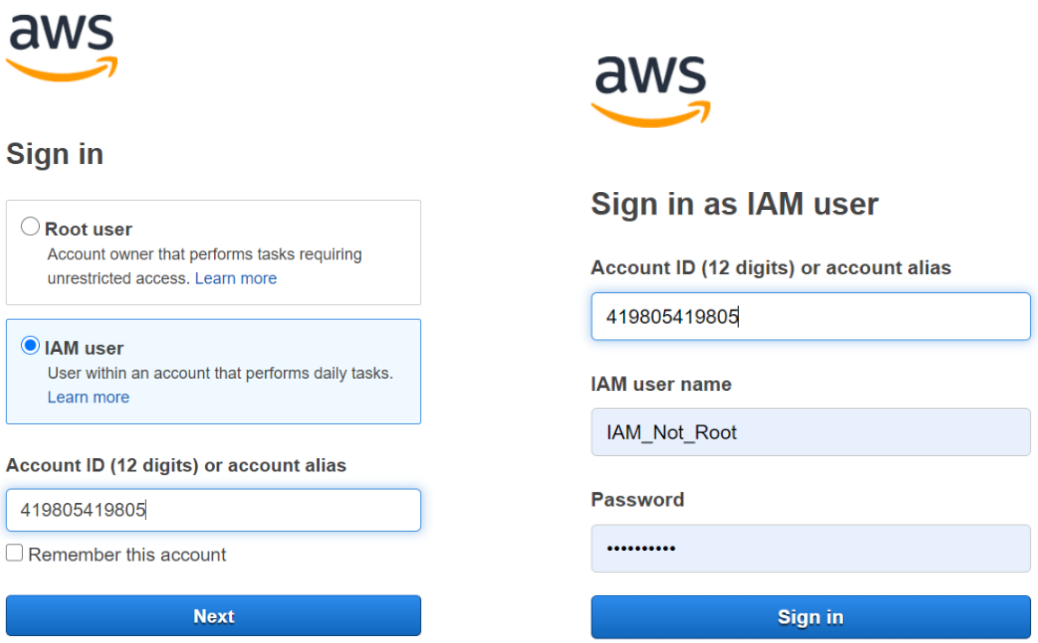
VERIFY YOUR IAM USER

2.22. Click on the email link to login

The password is in your downloaded CSV file

NOTE: IF THE EMAIL DOES NOT COME THROUGH FOR SOME REASON THE LOGIN LINK IS ALSO IN THE CSV FILE YOU JUST DOWNLOADED.

2.23. Sign Out then Sign Back in to the IAM User just created using the sign in link recorded in the last step.



Note: If “Autogenerated Password” was selected when setting up the IAM account, a prompt will request that password is changed upon first login using the new IAM user.

3. PROVISIONING THE PSOC 64

The CY8CKIT-064B0S2-4343W provisioning flow between the various build environments all share the same three steps for device provisioning and are executed in the same order as shown below.

First: Generate Image Keys:

- Generate a new private/public key pair that will be used to sign the firmware

Second: Create a Provisioning Packet:

- Use provided development cy_auth token
- Use the provided OEM RoT key to sign keys and device policies
- Sign provisioning packet with HSM private key

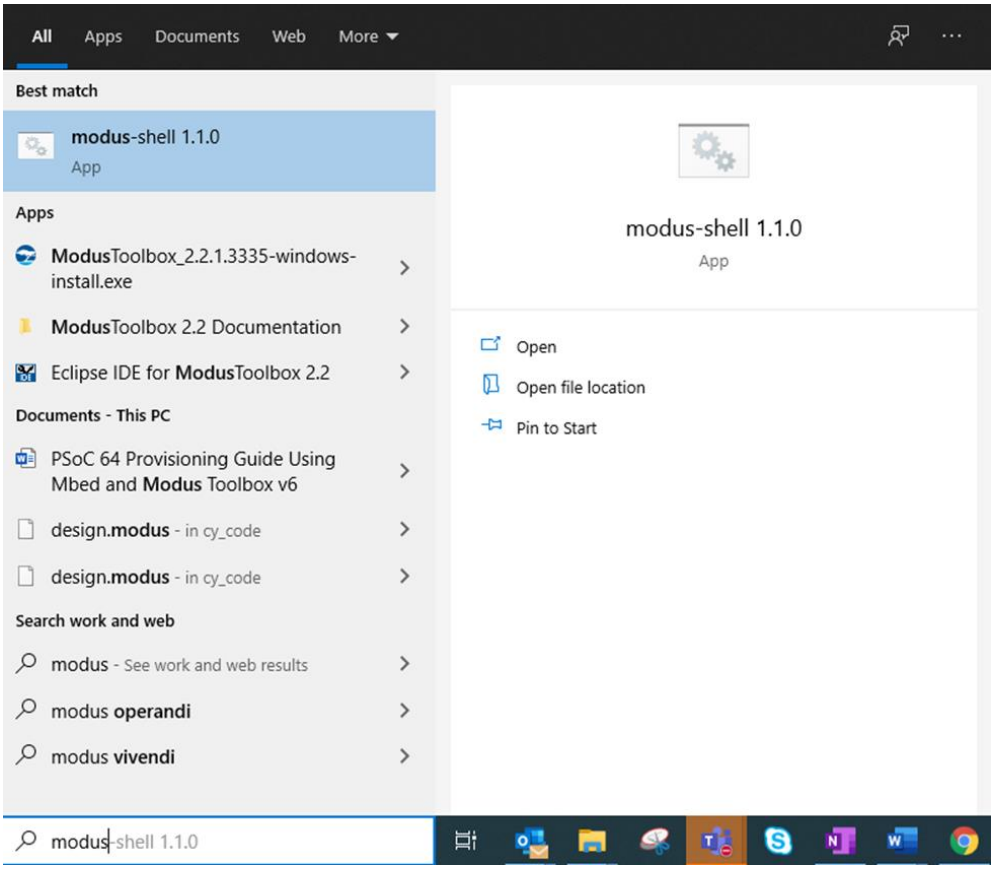
Third: Perform Provisioning:

- Run entrance exam
- Send the signed provisioning package to the PSoC64

Note: For the majority of commands in this pre-work and the subsequent PSoC 64 workshop it is recommended to use Modus Shell and NOT use a Windows Command Prompt. Software included with ModusToolbox has been configured to work with Modus Shell. However, a batch file within the demo program AWS_Demo, calls for python in Windows, outside of the Modus environment. For this case, python and several tools are required to be accessible from Windows Command Prompt.
Steps in a previous 'PYTHON SETUP IN WINDOWS' section setup the proper path variables.

Note: If using a Linux based Operating System, open and use a Terminal in place of Modus Shell or Windows Command Prompt.

3.1. From a Windows operating system, Click on the Windows Start Menu and type "Modus Shell" to reveal the Modus Shell App. Click on the App to open a Linux-Like internal terminal environment.

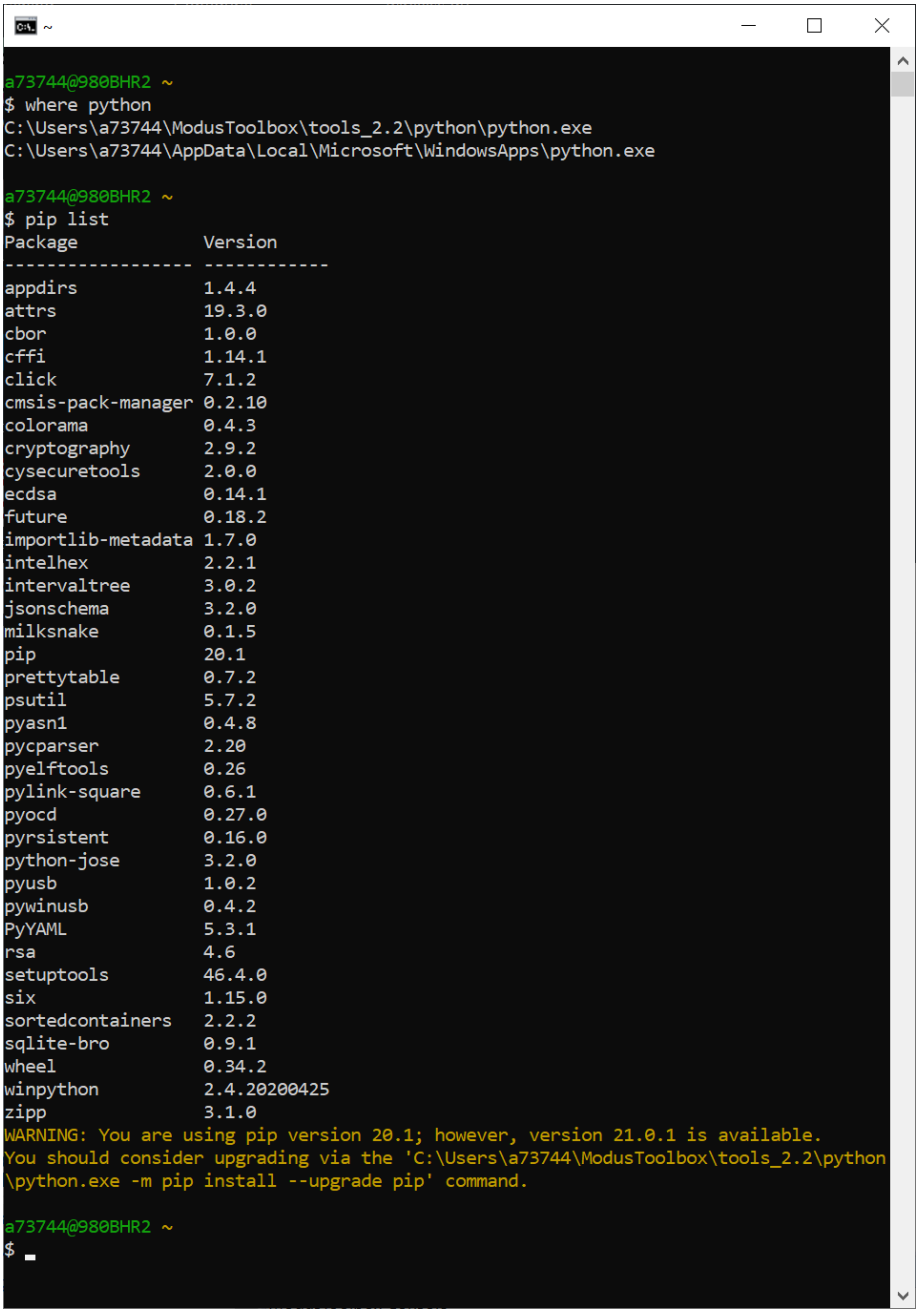


Note: There are two similar commands: “where” and “whereis”. Modus Shell recognizes both. Windows Command Prompt only recognizes “where”. Terminal in Linux only recognizes “whereis”. The “where” command returns a list of the executable programs. The “whereis” command returns executable program locations as well as relevant driver link lists (.dll files).
Note: If using a Linux Operating System, version 2.2 of ModusToolbox for Linux does NOT include Python. Reference the Appendix to learn how to access or install python onto a Linux environment.

3.2. Type the following two commands to reveal the location of Python and the versions of tools pre-installed. For a Windows environment, repeat this for both Modus Shell and Windows Command Prompt.

```
where python
```

```
pip list
```



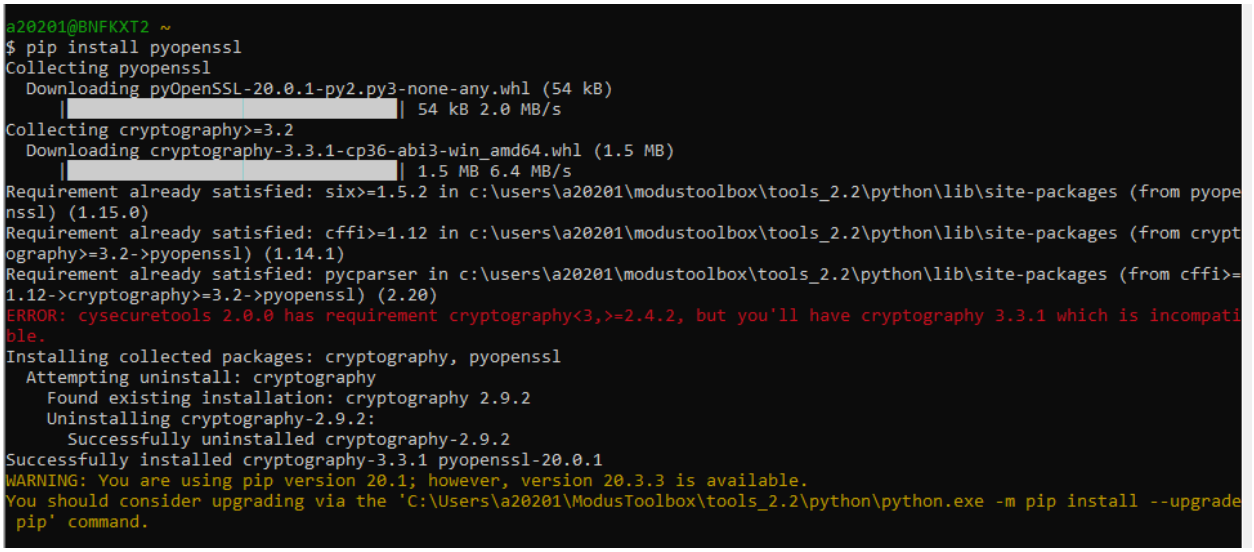
NOTE: THE FIRST LOCATION REVEALED OF PYTHON IS IN MODUSTOOLBOX. PYTHON WITHIN MODUSTOOLBOX WILL BE USED WHEN EXECUTING PYTHON COMMANDS FROM WITHIN MODUS SHELL REGARDLESS IF YOUR MACHINE HAS A SECOND VERSION OF PYTHON INSTALLED.

NOTE: THE INSTALLED VERSION OF PIP (PACKAGE INSTALLER FOR PYTHON) MAY BE AN OLDER VERSION. IF IT IS UPGRADED, ERRORS MAY DISPLAY IN THE SCREEN EVEN THOUGH THE TOOL WILL UPGRADE.

NOTE: "PIP LIST" REVEALS THE VERSION OF PIP ALONG WITH ALL OTHER PYTHON TOOLS.

3.3. Add pyopenssl in any Modus Shell, Command Prompt or Linux Terminal if it doesn't already exist by typing the following in each relative window:

```
pip install pyopenssl
```



3.4. Add cysecuretools in any Modus Shell, Command Prompt or Linux Terminal if it doesn't already exist by typing the following in each relative window:

```
pip install cysecuretools==2.1.0
```

Note: cysecuretools is provided as part of ModusToolbox for Windows. It is not preloaded in Linux.

3.5. For Linux environments only
, add cysecuretools as a program. cysecuretools is the Software Development Kit (SDK) provided by Infineon/Cypress to provision PSoC 64 devices.
Reference the PSoC 64 Secure MCU SDK User Guide @ https://www.cypress.com/documentation/software-and-drivers/psoc-64-secure-mcu-secure-boot-sdk-user-guide

NOTE: FOR MORE INFORMATION ON HOW TO PREPARE YOUR SOFTWARE ENVIRONMENT SEE SECTION 12 "SOFTWARE SETUP" IN THE APPENDIX OF THIS DOCUMENT.

4. AWS CREDENTIAL SETUP

CREATE A POLICY

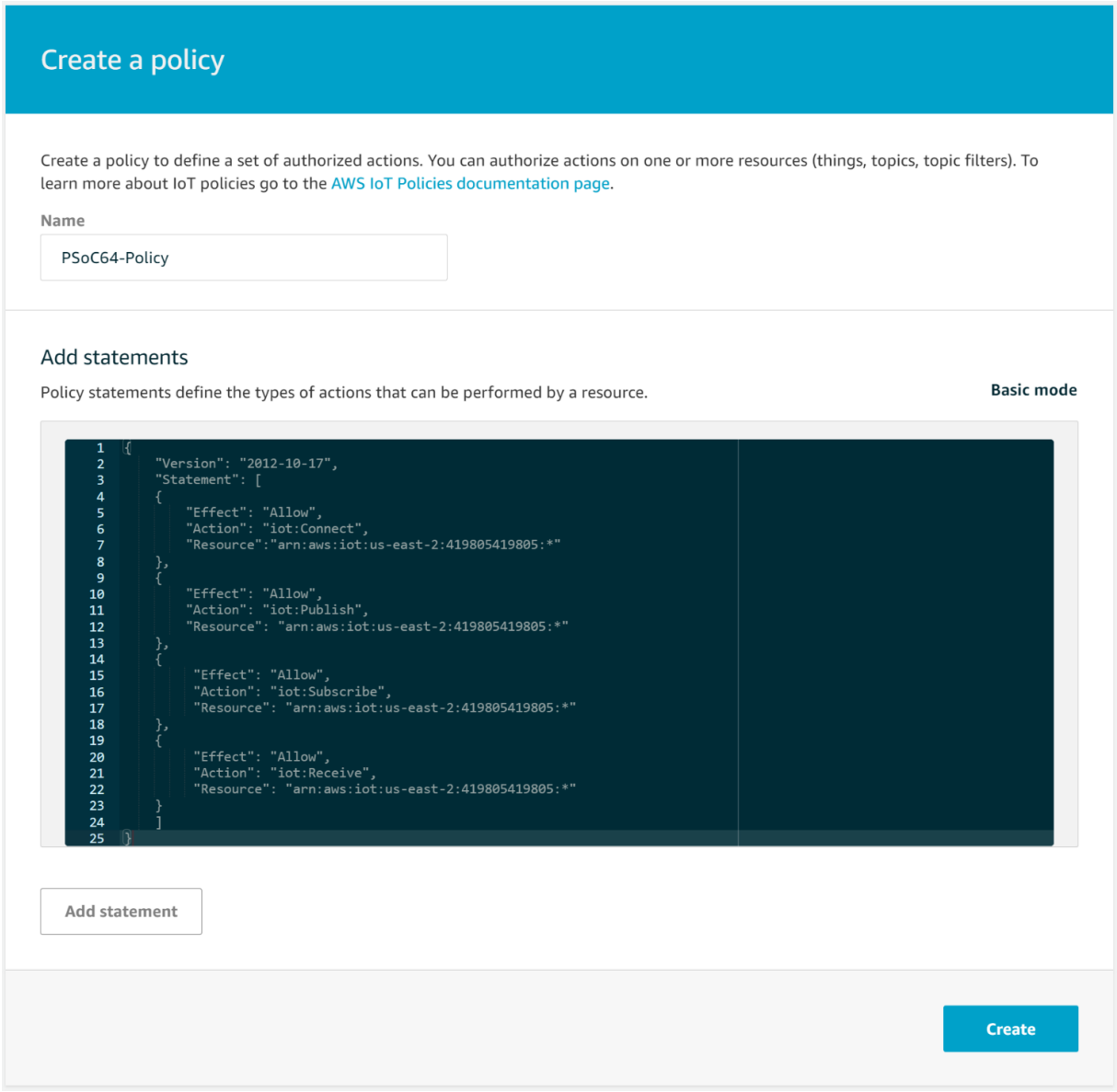
4.1. Log into your AWS account using an IAM User
4.2. Navigate to "AWS IoT" aka "IoT Core" or click this link to console
4.3. Expand the **Secure** tab on the left
4.4. Choose **Policies**
4.5. Choose **Create**
4.6. Enter a name for the policy

4.7. In the Add Statements section, **choose advanced mode**

4.8. Copy the following JSON text into the policy editor window.

```
{
   "Version": "2012-10-17",
   "Statement": [
   {
      "Effect": "Allow",
      "Action": "iot:Connect",
      "Resource":"arn:aws:iot:aws-region:aws-account-id:*"
   },
   {
      "Effect": "Allow",
      "Action": "iot:Publish",
      "Resource": "arn:aws:iot:aws-region:aws-account-id:*"
   },
   {
       "Effect": "Allow",
       "Action": "iot:Subscribe",
       "Resource": "arn:aws:iot:aws-region:aws-account-id:*"
   },
   {
       "Effect": "Allow",
       "Action": "iot:Receive",
       "Resource": "arn:aws:iot:aws-region:aws-account-id:*"
   }
   ]
}
```

- Recall your selected AWS Region and unique AWS account number recorded in Section 2.
- Replace *aws-region* and *aws-account-id* with your own region and account ID.
- Your updated policy statements should look similar to the screenshot below



4.9. Choose **Create**

REGISTER THE DEVICE

4.10. Navigate back to the AWS IoT console

4.11. On the left choose **Manage**

4.12. Choose **Things**

4.13. ***A]*** If there are not any registered things in your account yet choose **Register a Thing**.

B] If an IoT thing has been registered previously, choose **Create**

4.14. On the **Creating AWS IoT Things** page, choose **Create a single thing**

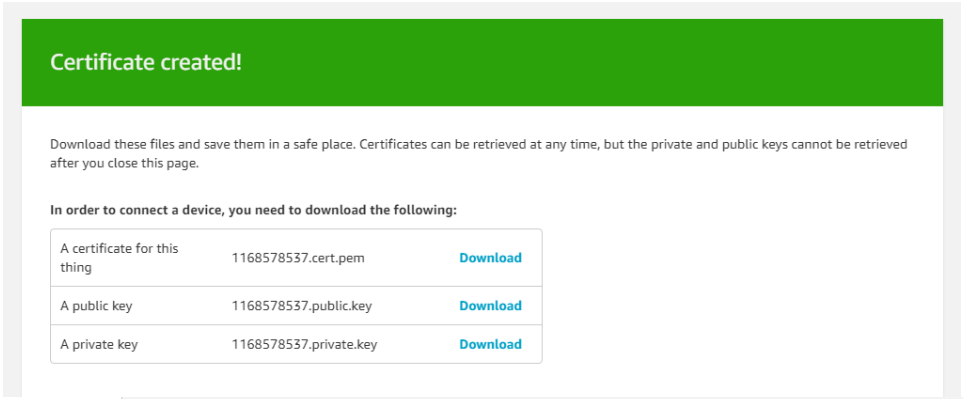
4.15. Enter a name for the PSoC 64 thing.

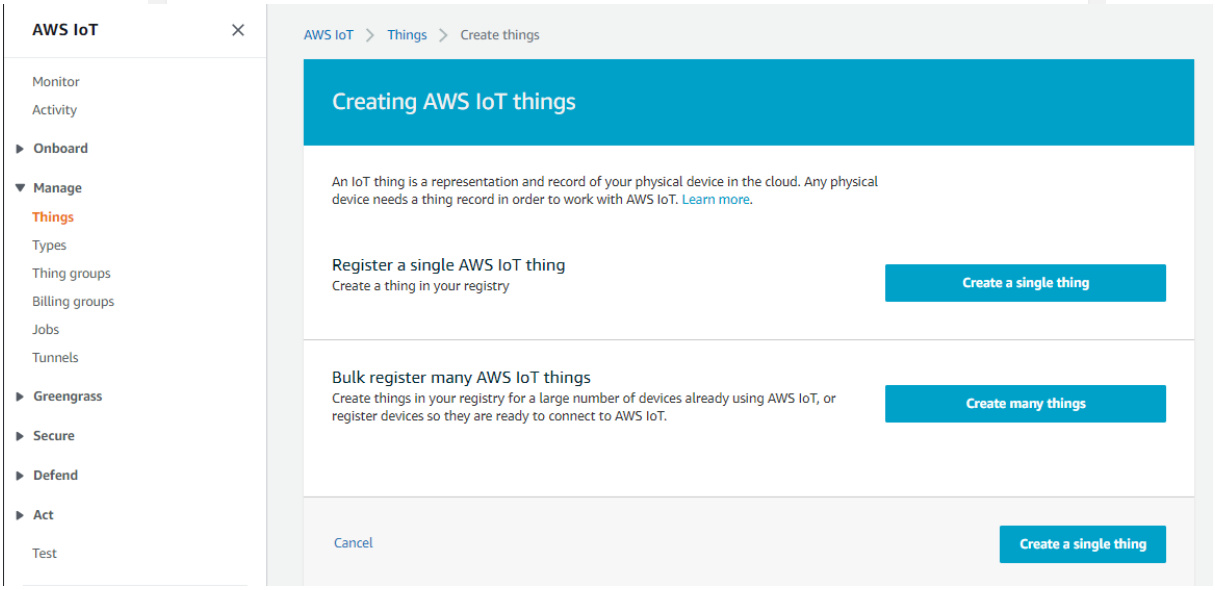
4.16. Leave the rest of the entries blank and choose **Next**

4.17. On the **Add a certificate for your thing** page, locate the **One-click certificate creation** option and choose **Create certificate**

4.18. On the following page download the certificate, public and private key files locally on your PC and choose **Activate** to activate the certificate that will be used by the PSoC 64 thing

4.19. There is no need to download a root CA for this setup and can be skipped





NOTE: THIS ABOVE FLOW IS DESIGNED FOR A DEVELOPMENT ENVIRONMENT WHERE YOU ARE WORKING WITH A SMALL NUMBER OF UNITS. FOR PRODUCTION QUANTITIES, THERE IS A MULTI-ACCOUNT-REGISTRATION (MAR) FLOW THAT HIDES THE PRIVATE KEY AND NEVER EXPOSES KEYS IN FW.

4.20. Lastly choose **Attach a policy**

4.21. Select the policy that was previously created

4.22. Choose **Register Thing**

5. DOWNLOAD FREERTOS

ENSURE FREERTOS RELEASE 202007.00 IS CHOSEN FOR THIS SETUP. LATER VERSIONS ARE NOT CURRENTLY SUPPORTED

NOTE: THE DOWNLOADING OF FREERTOS WAS MOVED TO AN EARLIER SECTION OF THESE EXERCISE INSTRUCTIONS TO ALLOW TIME FOR THE DOWNLOAD WHILE UNRELATED STEPS COULD BE PERFORMED.

1. **SKIP THESE THREE STEPS IN THIS SECTION IF FREERTOS VERSION 202007.00 HAS ALREADY BEEN DOWNLOADED.** **If not already done so… Download FreeRTOS** from the GitHub repository using Git. (Downloading the zip will NOT work)
   - **5.2.** From within Modus Shell, navigate to the directory where you'd like to install freeRTOS
   - **5.3.** If freeRTOS version 202007.00 has not already been downloaded…Type to following command to download the full correct version of freeRTOS:

```
git clone --branch 202007.00 https://github.com/aws/amazon-freertos --recursive
```

6. PROVISION THE BOARD

This provisioning process utilizes keys, policies, certificates and a python script developed for the Infineon/Cypress PSoC 6. Some of that information is included with freeRTOS. Additional information is customized for each user and added to the freeRTOS files using the steps below. All this information is contained within the directory:

```
<freertos>\vendors\cypress\MTB\psoc6\psoc64tfm\security
```

6.1. Within Modus Shell, navigate to the location where FreeRTOS was downloaded. Type dir to ensure you see "amazon-freertos" in the list of files and directories.

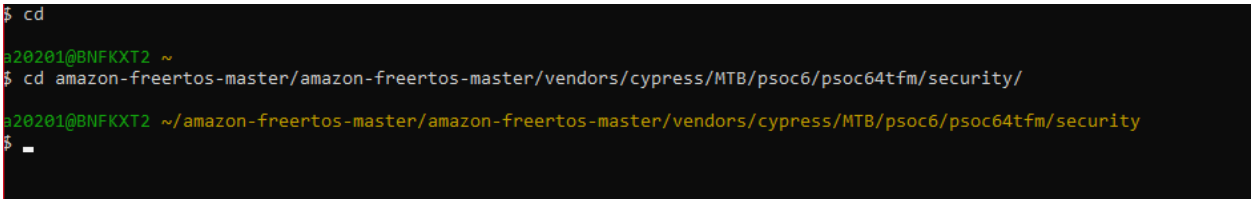
Tip: To know what directory you are in within Modus Shell, type "pwd".

Tip: To move up a directory, type "cd ..". Note: The top level C:/ directory will appear as /cygdrive/c/

Tip: When navigating directories within Modus Shell, the Forward Slash (/) must be used

6.2. Navigate to the correct security folder within freeRTOS directory or copy the following command into Modus Shell

```
cd vendors/cypress/MTB/psoc6/psoc64tfm/security
```



6.3. Run the following command to setup the FreeRTOS workspace with the Secure Boot SDK for the CY8CKIT-064S0S2-4343W kit

```
cysecuretools --target CY8CKIT-064S0S2-4343W init
```

6.4. You may be asked to overwrite files, type "y" in the command window and hit Enter to initialize the files for the board

- In this step CySecureTools provides default policies to choose from and sets up the folder with all the required security assets for the CY8CKIT-064S0S2-4343W
- A similar printout on the command prompt window should be seen in the below screenshot

```
$ cysecuretools --target CY8CKIT-064S0S2-4343W init
2021-01-22 20:43:54,615 : C : INFO  : Copy 'C:\Users\a20201\amazon-freertos-master\amazon-freertos-master\vendors\cyp
ress\MTB\psoc6\psoc64tfm\security\packets\cy_auth_2m_s0_sample.jwt'
2021-01-22 20:43:54,623 : C : INFO  : Copy 'C:\Users\a20201\amazon-freertos-master\amazon-freertos-master\vendors\cyp
ress\MTB\psoc6\psoc64tfm\security\packets\control_dap_cert.json'
2021-01-22 20:43:54,631 : C : INFO  : Copy 'C:\Users\a20201\amazon-freertos-master\amazon-freertos-master\vendors\cyp
ress\MTB\psoc6\psoc64tfm\security\keys\hsm_state.json'
2021-01-22 20:43:54,637 : C : INFO  : Copy 'C:\Users\a20201\amazon-freertos-master\amazon-freertos-master\vendors\cyp
ress\MTB\psoc6\psoc64tfm\security\keys\oem_state.json'
2021-01-22 20:43:54,639 : C : INFO  : Copy 'C:\Users\a20201\amazon-freertos-master\amazon-freertos-master\vendors\cyp
ress\MTB\psoc6\psoc64tfm\security\policy\policy_multi_CM0_CM4.json'
2021-01-22 20:43:54,646 : C : INFO  : Copy 'C:\Users\a20201\amazon-freertos-master\amazon-freertos-master\vendors\cyp
ress\MTB\psoc6\psoc64tfm\security\policy\policy_multi_CM0_CM4_smif.json'
2021-01-22 20:43:54,654 : C : INFO  : Copy 'C:\Users\a20201\amazon-freertos-master\amazon-freertos-master\vendors\cyp
ress\MTB\psoc6\psoc64tfm\security\policy\policy_single_CM0_CM4.json'
2021-01-22 20:43:54,661 : C : INFO  : Copy 'C:\Users\a20201\amazon-freertos-master\amazon-freertos-master\vendors\cyp
ress\MTB\psoc6\psoc64tfm\security\policy\policy_single_CM0_CM4_smif.json'
2021-01-22 20:43:54,676 : C : INFO  : Copy 'C:\Users\a20201\amazon-freertos-master\amazon-freertos-master\vendors\cyp
ress\MTB\psoc6\psoc64tfm\security\prebuilt\CyBootloader_Release\CypressBootloader_CM0p.hex'
2021-01-22 20:43:54,683 : C : INFO  : Copy 'C:\Users\a20201\amazon-freertos-master\amazon-freertos-master\vendors\cyp
ress\MTB\psoc6\psoc64tfm\security\prebuilt\CyBootloader_Release\CypressBootloader_CM0p.jwt'
2021-01-22 20:43:54,692 : C : INFO  : Copy 'C:\Users\a20201\amazon-freertos-master\amazon-freertos-master\vendors\cyp
ress\MTB\psoc6\psoc64tfm\security\prebuilt\CyBootloader_WithLogs\CypressBootloader_CM0p.hex'
2021-01-22 20:43:54,699 : C : INFO  : Copy 'C:\Users\a20201\amazon-freertos-master\amazon-freertos-master\vendors\cyp
ress\MTB\psoc6\psoc64tfm\security\prebuilt\CyBootloader_WithLogs\CypressBootloader_CM0p.jwt'

a20201@BNFKXT2 ~/amazon-freertos-master/amazon-freertos-master/vendors/cypress/MTB/psoc6/psoc64tfm/security
$
```

NOTE: FOR THIS ENVIRONMENT SETUP, THE "POLICY_MULTI_CM0_CM4_TFM.JSON" POLICY FILE PROVIDED WITH THE FREERTOS REPOSITORY WILL BE USED. THE FILE CAN BE LOCATED AT THE BELOW LOCATION:

<freertos>/vendors/cypress/MTB/psoc6/psoc64tfm/security/policy/policy_multi_CM0_CM4_tfm.json

A high-level overview of the policy is shown in the following table

Feature	Policy Setup
CM0+ Debug Port	Open
CM4 Debug Port	Open
SysAP Debug Port	Open
CM0+ (Trusted Firmware) Flash Size	320KB
CM4 (Application Firmware) Flash Size	1152KB
External Memory Enabled for Update?	Yes

The FreeRTOS package has default keys available to use for provisioning but it is recommended to create a new key pair to sign the firmware by running the following command

6.5. You may be asked to overwrite files again. Overwrite those files.

6.6. Ensure you are in <freertos>\vendors\cypress\MTB\psoc6\psoc64tfm\security

```
cysecuretools --policy ./policy/policy_multi_CM0_CM4_tfm.json --target
CY8CKIT-064S0S2-4343W create-keys
```

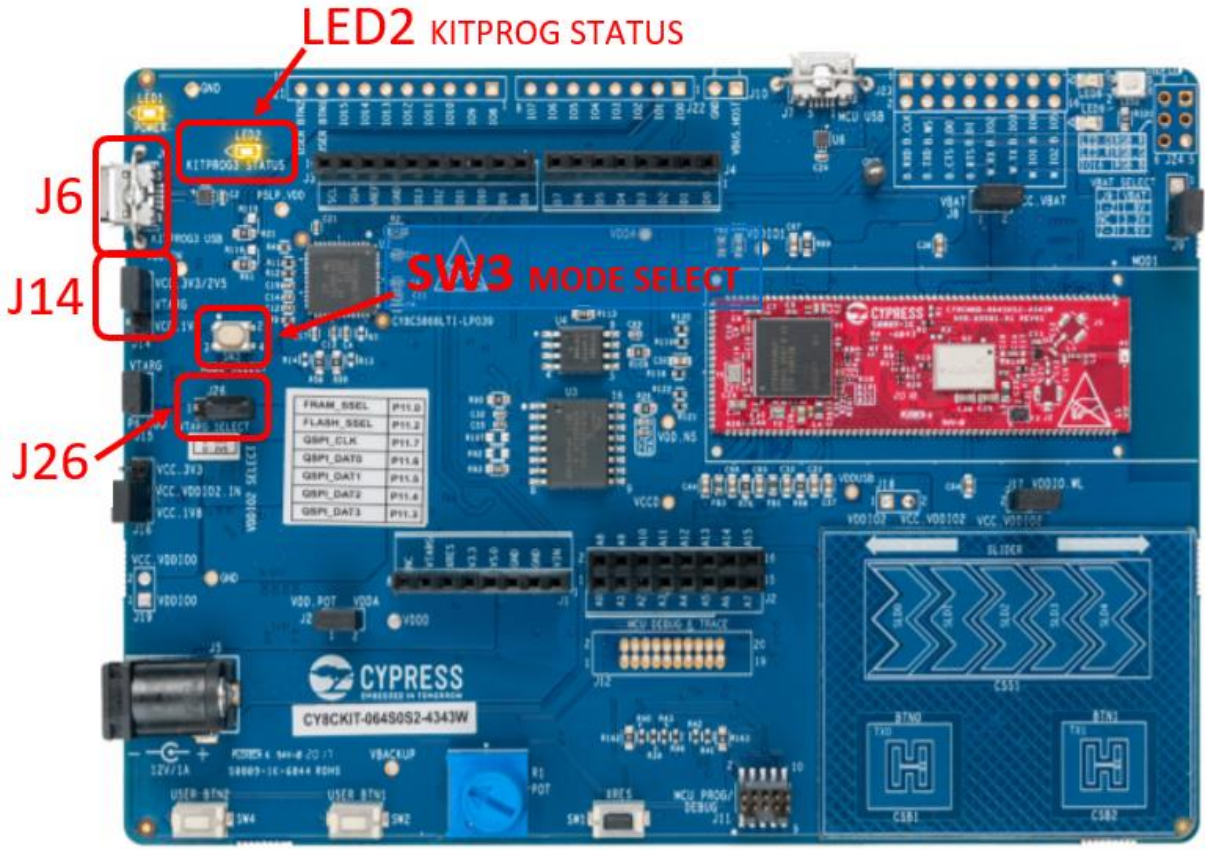
```
a20201@BNFKXT2 ~/amazon-freertos-master/amazon-freertos-master/vendors/cypress/MTB/psoc6/psoc64tfm/security
$ cysecuretools --policy ./policy/policy_multi_CM0_CM4_tfm.json --target CY8CKIT-064S0S2-4343W create-keys
2021-01-22 20:46:30,483 : C : INFO  : Created key in C:\Users\a20201\amazon-freertos-master\amazon-freertos-master\ve
ndors\cypress\MTB\psoc6\psoc64tfm\security\keys\TFM_S_KEY.json
2021-01-22 20:46:30,485 : C : INFO  : Created key in C:\Users\a20201\amazon-freertos-master\amazon-freertos-master\ve
ndors\cypress\MTB\psoc6\psoc64tfm\security\keys\TFM_NS_KEY.json

a20201@BNFKXT2 ~/amazon-freertos-master/amazon-freertos-master/vendors/cypress/MTB/psoc6/psoc64tfm/security
$
```

Note: This is the first point in this pre-work where the kit will be used. The sensors and interface board are NOT required at this time. If the interface board and sensors have been assembled per instructions in the Appendix, it is recommended to remove the interface board and sensors from the base PSoC 64 kit in order to access the mode switch SW3 and to more easily see the “Kitprog status” LED.

Note: If you prefer to leave the interface board and sensor on, the FW-Loader software tool can be used to determine and change Kitprog status. Reference “COMMUNICATION WITH KIT” section in the Appendix.

6.7. Remove the jumper from [J26] to change the VTARG voltage to 2.5V and ensure the jumper on [J14] is placed between pins 2 and 3



6.8. Connect your kit to the PC using the provided USB cable through the KitProg3 USB connector [J6]

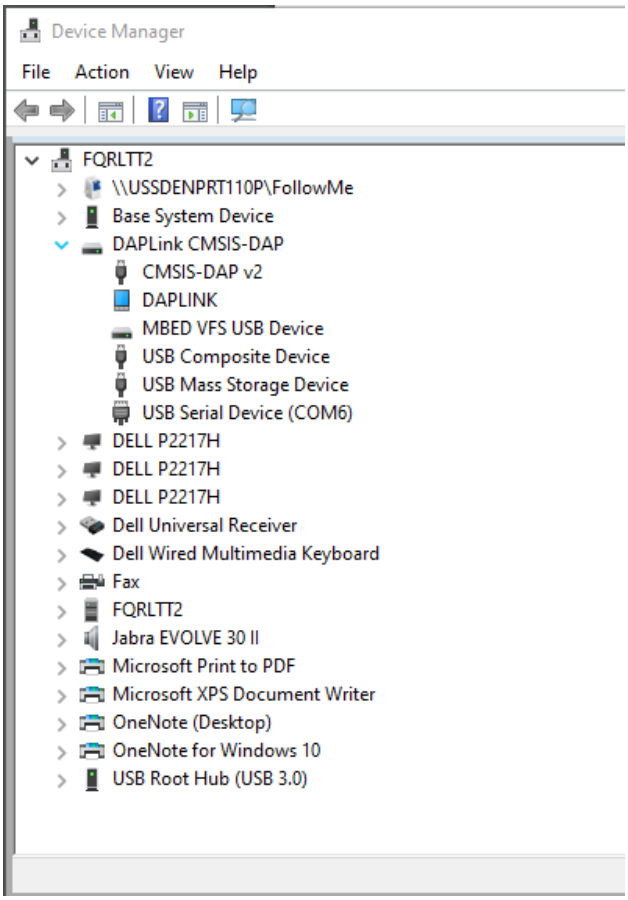
6.9. Locate [SW3] near the micro USB port and ensure the kit is in DAPLink mode

- DAPLink mode is indicated by the status LED [LED2] ramping on/off fast (~2Hz)
- If the kit is not in DAPLink mode (if [LED2] is not ramping on/off), press [SW3] once and wait for [LED2] to change states. If needed, repeat this process until [LED2] is ramping on/off thus indicating the proper board state.

6.10. There have been some rare issues with the KitProg3 modes and Windows 10 not able to register the USB device correctly when its plugged in. If this occurs, please see Chapter 7 of the KitProg3 User Guide for information on fixing this

6.11. When KitProg3 is set to DAPLink mode the kit should register as a Disk Drive in Windows

This can be confirmed by opening Device Manger and under the **View** tab select **Devices by Container** where there should be disk drive named "DAPLink CMSIS-DAP"



6.12. Ensure you are still in the following directory:

<freertos>\vendors\cypress\MTB\psoc6\psoc64tfm\security

then run the below command to provision a new kit

```
cysecuretools --policy ./policy/policy_multi_CM0_CM4_tfm.json --target
CY8CKIT-064S0S2-4343W provision-device
```

- If you have previously provisioned the board, it can be reprovisioned with the below command

```
cysecuretools --policy ./policy/policy_multi_CM0_CM4_tfm.json --target
CY8CKIT-064S0S2-4343W re-provision-device
```

NOTE: IF THE PROVISIONING PROCESS FAILS, PLEASE REFERENCE THE APPENDIX SECTION 12 “POTENTIAL MODUSTOOLBOX BUILD FAILURE” FOR POSSIBLE SOLUTIONS.

Once the command has finished executing, a similar successful printout should appear in the command line window

```
2021-01-22 21:02:17,265 : C : INFO  : ****************************************
2021-01-22 21:02:17,266 : C : INFO  :        ENTRANCE EXAM PASSED
2021-01-22 21:02:17,266 : C : INFO  : ****************************************
2021-01-22 21:02:17,493 : P : INFO  : DP IDR = 0x6ba02477 (v2 rev6)
2021-01-22 21:02:17,711 : C : INFO  : JWT packet size: 2448
2021-01-22 21:02:19,627 : C : INFO  : Read FlashBoot firmware status:
2021-01-22 21:02:19,630 : C : INFO  : FlashBoot firmware status = 0xf7000107
2021-01-22 21:02:19,630 : C : INFO  : Received FB_FW_STATUS = 0xf0000000
2021-01-22 21:02:19,633 : C : INFO  : Expected FB_FW_STATUS = 0xf0000000
2021-01-22 21:02:19,634 : C : INFO  : BOOT slot will remain the same and can affect rollback counter
2021-01-22 21:02:19,635 : C : INFO  : Erase main smif slots:
2021-01-22 21:02:19,639 : C : INFO  : Use cm4 AP
2021-01-22 21:02:19,644 : P : INFO  : Clearing TEST_MODE bit...
2021-01-22 21:02:19,645 : C : INFO  : erasing address 0x18000000, size 0x120000 ...
2021-01-22 21:02:19,646 : P : INFO  : Acquiring target...
2021-01-22 21:02:19,697 : P : INFO  : DP IDR = 0x6ba02477 (v2 rev6)
2021-01-22 21:02:19,702 : P : INFO  : DP IDR = 0x6ba02477 (v2 rev6)
2021-01-22 21:02:19,707 : P : INFO  : DP IDR = 0x6ba02477 (v2 rev6)
2021-01-22 21:02:19,716 : P : INFO  : DP IDR = 0x6ba02477 (v2 rev6)
2021-01-22 21:02:19,959 : P : INFO  : Erasing sector 0x18000000 (262144 bytes)
2021-01-22 21:02:20,420 : P : INFO  : Erasing sector 0x18040000 (262144 bytes)
2021-01-22 21:02:20,886 : P : INFO  : Erasing sector 0x18080000 (262144 bytes)
2021-01-22 21:02:21,344 : P : INFO  : Erasing sector 0x180c0000 (262144 bytes)
2021-01-22 21:02:21,803 : P : INFO  : Erasing sector 0x18100000 (262144 bytes)
2021-01-22 21:02:22,289 : C : INFO  : Erasing complete

2021-01-22 21:02:22,289 : C : INFO  : Use system AP
2021-01-22 21:02:22,296 : P : INFO  : Clearing TEST_MODE bit...
2021-01-22 21:02:25,302 : C : INFO  : Use cm4 AP
2021-01-22 21:02:25,310 : P : INFO  : Clearing TEST_MODE bit...
2021-01-22 21:02:25,311 : C : INFO  : Programming bootloader 'C:\Users\a20201\ModusToolbox\tools_2.2\python\lib\site-
packages\cysecuretools\targets\cys06xxa\prebuilt\CyBootloader_WithLogs\CypressBootloader_CM0p.hex':
2021-01-22 21:02:25,379 : P : INFO  : Acquiring target...
2021-01-22 21:02:25,430 : P : INFO  : DP IDR = 0x6ba02477 (v2 rev6)
2021-01-22 21:02:25,435 : P : INFO  : DP IDR = 0x6ba02477 (v2 rev6)
2021-01-22 21:02:25,440 : P : INFO  : DP IDR = 0x6ba02477 (v2 rev6)
2021-01-22 21:02:25,451 : P : INFO  : DP IDR = 0x6ba02477 (v2 rev6)
[====================] 100%
2021-01-22 21:02:29,266 : P : INFO  : Acquiring target...
2021-01-22 21:02:29,318 : P : INFO  : DP IDR = 0x6ba02477 (v2 rev6)
2021-01-22 21:02:29,323 : P : INFO  : DP IDR = 0x6ba02477 (v2 rev6)
2021-01-22 21:02:29,328 : P : INFO  : DP IDR = 0x6ba02477 (v2 rev6)
2021-01-22 21:02:29,335 : P : INFO  : DP IDR = 0x6ba02477 (v2 rev6)
2021-01-22 21:02:29,339 : P : INFO  : Erased 60928 bytes (119 sectors), programmed 60928 bytes (119 pages), skipped 0
 bytes (0 pages) at 15.01 kB/s
2021-01-22 21:02:29,340 : C : INFO  : Programming bootloader complete
2021-01-22 21:02:29,340 : C : INFO  : Use system AP
2021-01-22 21:02:29,346 : P : INFO  : Clearing TEST_MODE bit...
2021-01-22 21:02:29,569 : P : INFO  : DP IDR = 0x6ba02477 (v2 rev6)
2021-01-22 21:02:33,356 : C : INFO  : Run provisioning syscall:
2021-01-22 21:02:33,356 : C : INFO  : JWT packet size: 6460
2021-01-22 21:02:36,721 : P : INFO  : DP IDR = 0x6ba02477 (v2 rev6)
2021-01-22 21:02:39,737 : C : INFO  : FlashBoot firmware status = 0xa1000101
2021-01-22 21:02:39,737 : C : INFO  : ****************************************
2021-01-22 21:02:39,738 : C : INFO  :        PROVISIONING PASSED
2021-01-22 21:02:39,740 : C : INFO  : ****************************************

2021-01-22 21:02:39,746 : P : INFO  : Clearing TEST_MODE bit...

a20201@BNFKXT2 ~/amazon-freertos-master/amazon-freertos-master/vendors/cypress/MTB/psoc6/psoc64tfm/security
$
```

6.13. Disconnect the kit from the PC and put the jumper back on [J26] to set the kit at a 3V3 operating voltage

6.14. Power on the kit and press [SW3] one time to change KitProg3 into CMSIS-DAP Bulk mode. The Status LED [LED2] should be on steady and not blinking. If needed, repeat this process until [LED2] is on steady, indicating the board is in the proper state.

The kit is officially provisioned and now ready to accept signed firmware!

7. CONFIGURE THE AWS DEMOS

7.1. **Open the ModusToolbox Eclipse IDE** and choose or create a new workspace

7.2. From the **File** menu choose **Import**

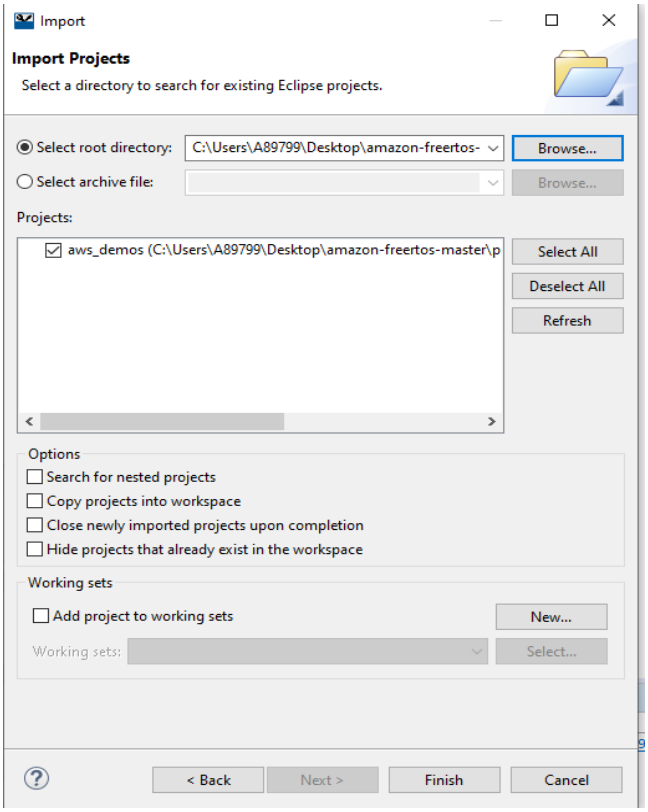
7.3. Expand the **General** tab and choose **Existing Projects into Workspace** then click **Next**

7.4. In the root directory browse to where you downloaded the FreeRTOS package and navigate to the aws_demos folder then click **Select Folder**.

<freertos>/projects/cypress/CY8CKIT-064S0S2-4343W/mtb/aws_demos

7.5. The **Projects** section should be filled with a project named “aws_demos”

7.6. Select the project and click **finish** to import it into your workspace

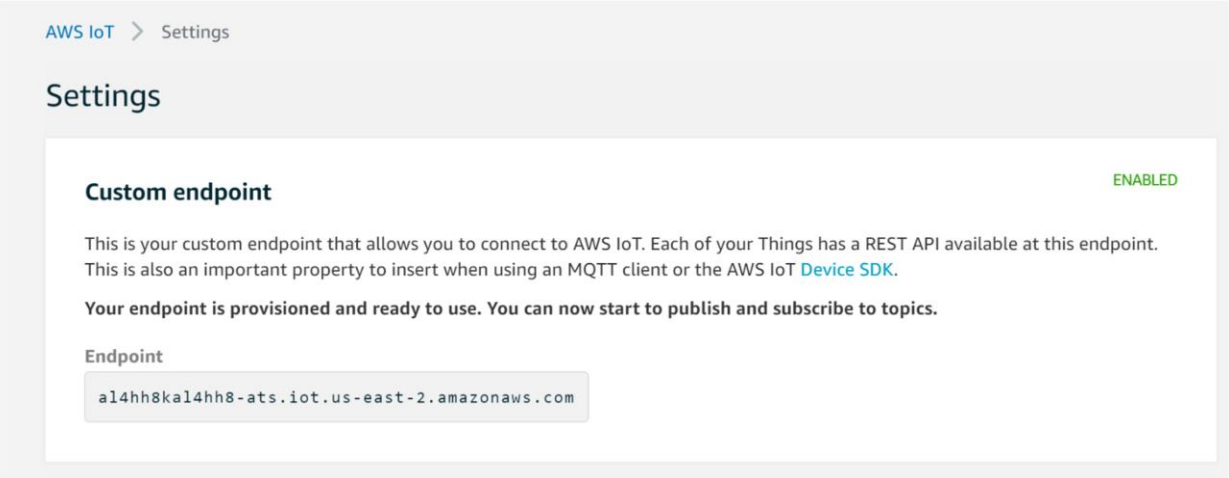


In order to connect to AWS you have to edit some configuration files with the proper credentials that will link to the thing that’s registered in AWS

8. CONFIGURING YOUR AWS IOT ENDPOINT AND WIFI CREDENTIALS

8.1. In the **AWS Management Console** navigate to the **IoT Core** also referred to as **AWS IoT**

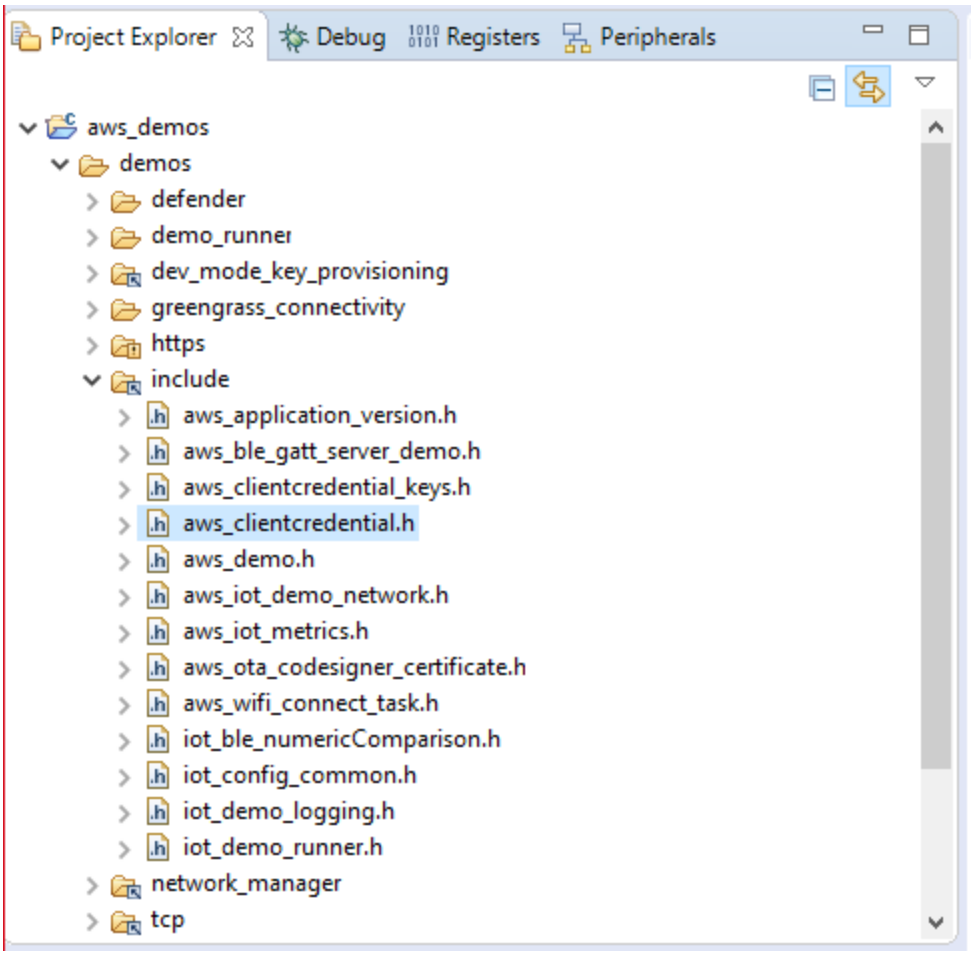
8.2. Choose **Settings** towards the bottom left of the navigation pane and make a note of the endpoint address, it should look like:



8.3. Expand **Manage** in the navigation pane and choose **Things**

8.4. Make a note of the AWS IoT thing name that you assigned to the PSoC 64 device

8.5. In ModusToolbox under Project Explorer expand the aws_demos root directory and open **/demos/include/aws_clientcredential.h**



8.6. Locate the below #define directives and set the proper values for your endpoint address, AWS IoT Thing name, and WiFi credentials

#define clientcredentialMQTT_BROKER_ENDPOINT " *Your AWS IoT endpoint*"

#define clientcredentialIOT_THING_NAME " *Your AWS IoT thing name*"

#define clientcredentialWIFI_SSID " *The SSID for your Wi-Fi network*"

#define clientcredentialWIFI_PASSWORD " *The password for your Wi-Fi*"

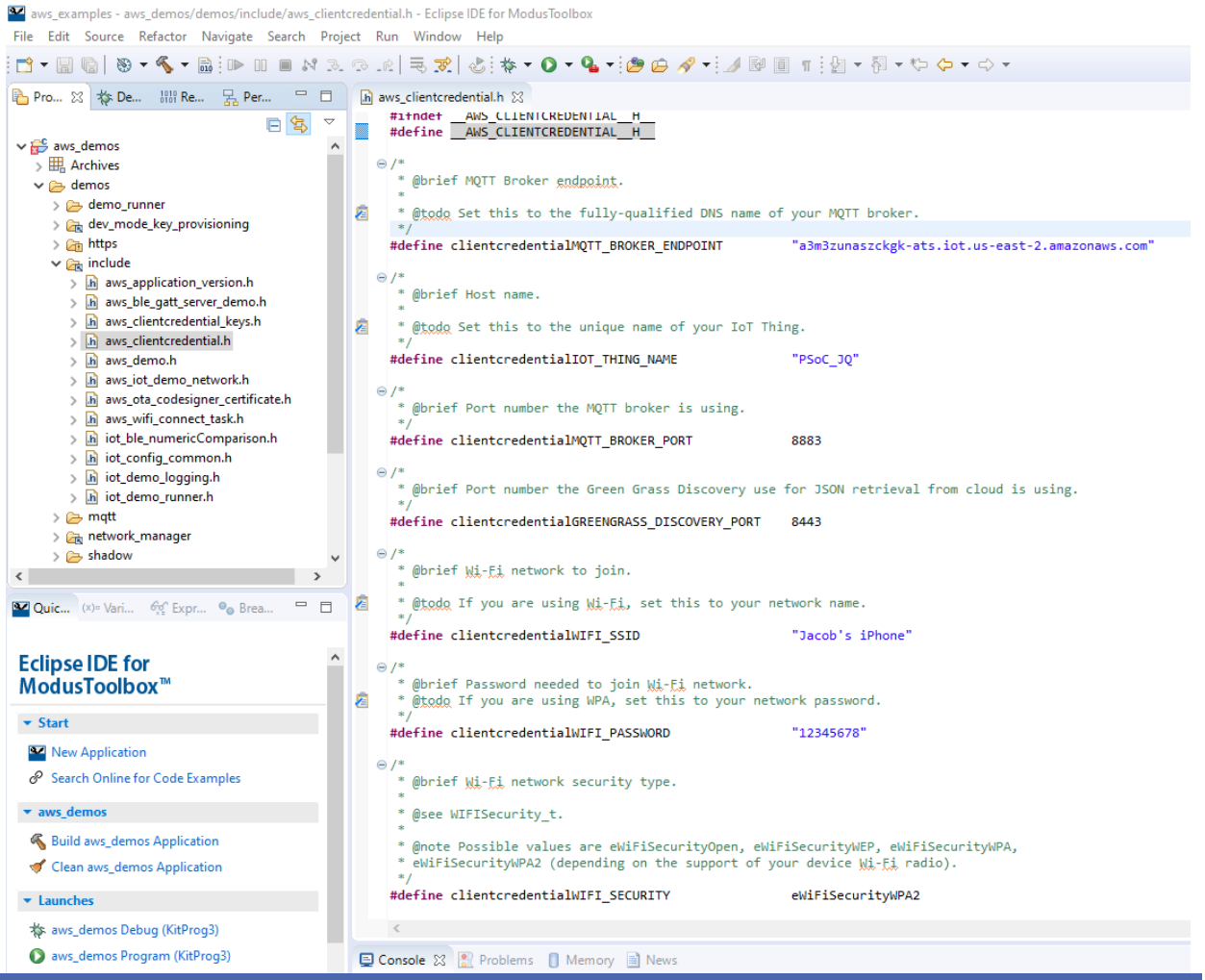
#define clientcredentialWIFI_SECURITY *The security type of your Wi-Fi network*

Valid security types are:

- eWiFiSecurityOpen (Open, no security)
- eWiFiSecurityWEP (WEP security)
- eWiFiSecurityWPA (WPA security)
- eWiFiSecurityWPA2 (WPA2 security)
    - WPA2 is most common

8.7. The header file should look similar to the below screenshot for reference

8.8. In the upper left corner of ModusToolbox, save the edits made to the project



9. FORMATTING THE AWS IOT CREDENTIALS

FreeRTOS needs the AWS IoT certificate and private key that is linked with your registered thing and its permissions policies to successfully communicate with AWS IoT.

FreeRTOS is a C language project, and the certificate and private key must be specially formatted to be added to the project.

9.1. In File Explorer open the below HTML file in a browser
<freertos>/tools/certificate_configuration/CertificateConfigurator.html

9.2. Under the **Certificate PEM file** and **Private Key PEM file** entries, locate the device certificate and private key files that were downloaded from AWS

9.3. Populate the entries with their respective file path locations

Certificate PEM file: <User>/Downloads/*ID-certificate.pem.crt*

Private Key PEM file: <User>/Downloads/*ID-private.pem.key*

9.4. Choose **Generate and save aws_clientcredential_keys.h** and save the file in <freertos>/demos/include
NOTE: IN WINDOWS YOU WILL GET A WARNING TO OVERWRITE THE EXISTING FILE IN THE FOLDER

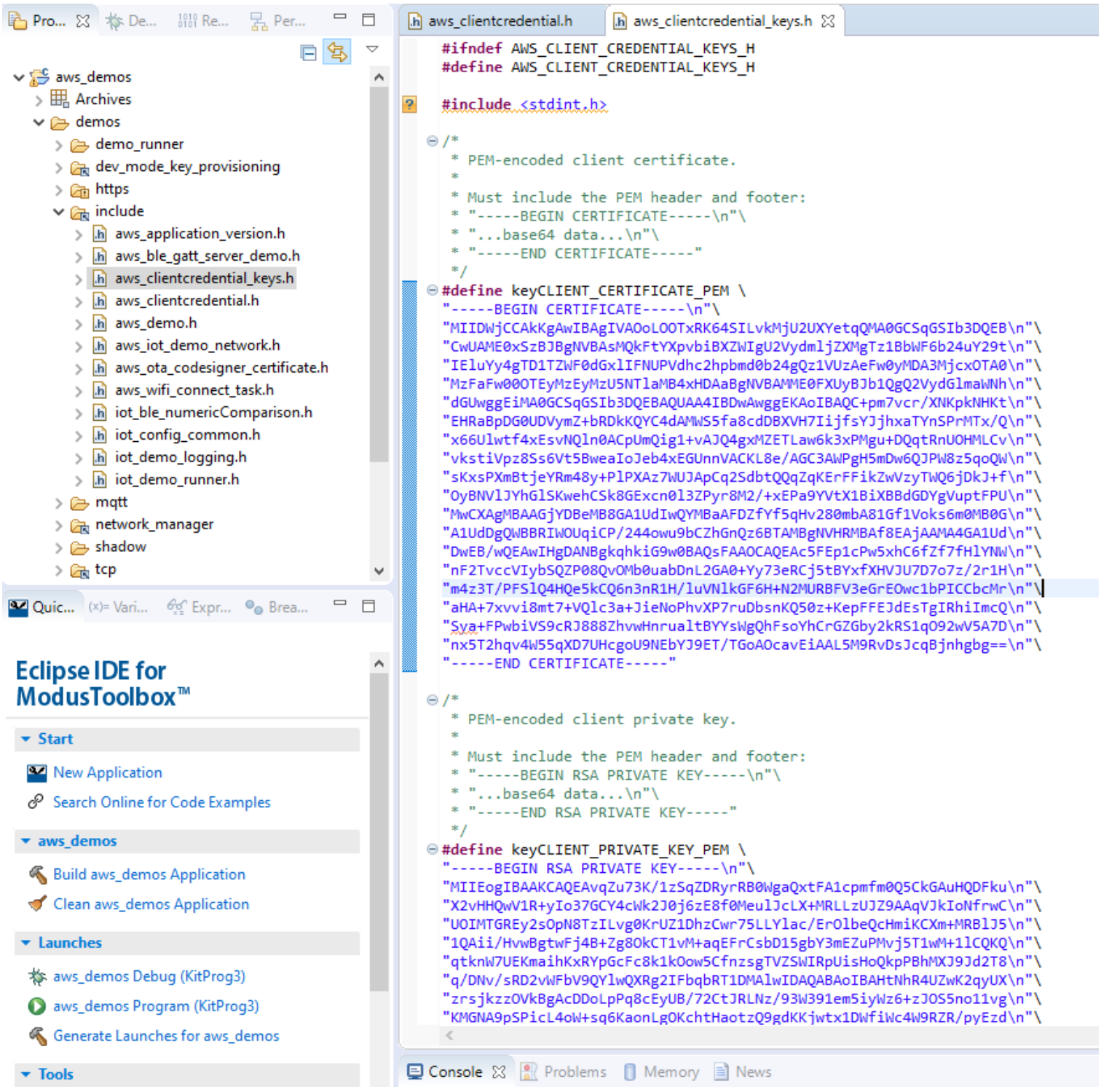
9.5. Once the file is saved, the aws_demos project in ModusToolbox should automatically update the **/demos/include** directory with the new aws_clientcredential_keys.h header file

9.6. In ModusToolbox, open the aws_clientcredential_keys.h header file and confirm that the file has been updated

9.7. The file should be populated with your certificate and private key beneath the following two preprocessor directives

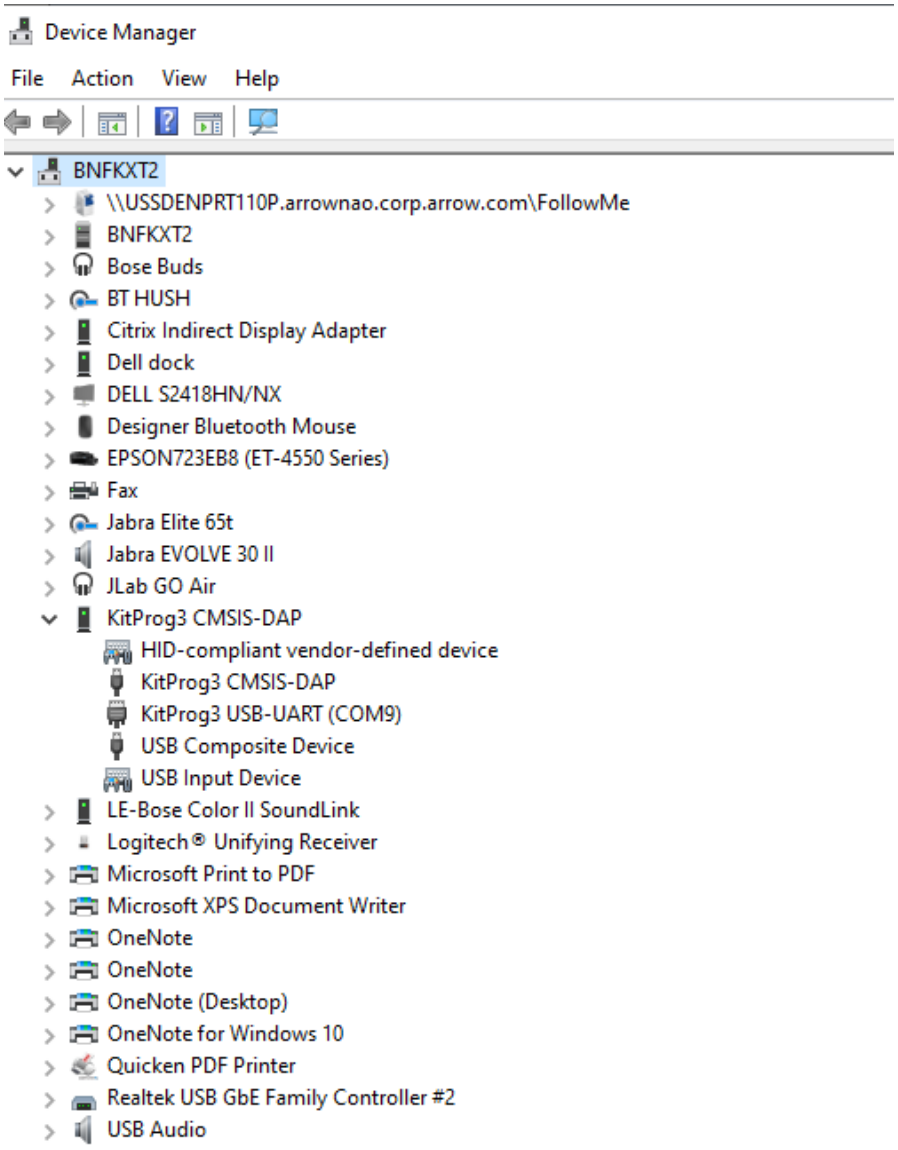
```
#define keyCLIENT_CERTIFICATE_PEM

#define keyCLIENT_PRIVATE_KEY_PEM
```


Pro...
De...
Re...
Per...
aws_demos
Archives
demos
demo_runner
dev_mode_key_provisioning
https
include
aws_application_version.h
aws_ble_gatt_server_demo.h
aws_clientcredential_keys.h
aws_clientcredential.h
aws_demo.h
aws_iot_demo_network.h
aws_ota_codesigner_certificate.h
aws_wifi_connect_task.h
iot_ble_numericComparison.h
iot_config_common.h
iot_demo_logging.h
iot_demo_runner.h
mqtt
network_manager
shadow
tcp
Quic...
(x)= Vari...
Expr...
Brea...
Eclipse IDE for ModusToolbox™
Start
New Application
Search Online for Code Examples
aws_demos
Build aws_demos Application
Clean aws_demos Application
Launches
aws_demos Debug (KitProg3)
aws_demos Program (KitProg3)
Generate Launches for aws_demos
Tools
aws_clientcredential.h
aws_clientcredential_keys.h
#ifndef AWS_CLIENT_CREDENTIAL_KEYS_H
#define AWS_CLIENT_CREDENTIAL_KEYS_H
#include <stdint.h>
/*
* PEM-encoded client certificate.
*
* Must include the PEM header and footer:
* "-----BEGIN CERTIFICATE-----\n"\
* "...base64 data...\n"\
* "-----END CERTIFICATE-----"
*/
#define keyCLIENT_CERTIFICATE_PEM \
"-----BEGIN CERTIFICATE-----\n"\
"MIIDWjCCAkKgAwIBAgIVAOoLOOTxRK64SILvkMjU2UXYetqQMA0GCSqGSIb3DQEB\n"\
"CwUAME0xSzBJBgNVBAsMQkFtYXpvbiBXZWIgU2VydmljZXMgTz1BbWF6b24uY29t\n"\
"IEluYy4gTD1TZWF0dGxlIFNUPVdhc2hpbmd0b24gQz1VUzAeFw0yMDA3MjcxOTA0\n"\
"MzFaFw00OTEyMzEyMzU5NTlaMB4xHDAaBgNVBAMME0FXUyBJb1QgQ2VydGlmaWNh\n"\
"dGUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC+pm7vcr/XNKpkNHKt\n"\
"EHRaBpDG0UDVymZ+bRDkKQYC4dAMWS5fa8cdDBXVH7IijfsYJjhxaTYnSPrMTx/Q\n"\
"x66Ulwtf4xEsvNQln0ACpUmQig1+vAJQ4gxMZETLaw6k3xPMgu+DQqtRnUOHMLCv\n"\
"vkstiVpz8Ss6Vt5BweaIoJeb4xEGUnnVACKL8e/AGC3AWPgH5mDw6QJPW8z5qoQW\n"\
"sKxsPXmBtjeYRm48y+PlPXAz7WUJApCq2SdbtQQqZqKErFfikZwVzyTWQ6jDkJ+f\n"\
"OyBNVlJYhGlSKwehCSk8GExcn0l3ZPyr8M2/+xEPa9YVtX1BiXBBdGDYgVuptFPU\n"\
"MwCXAgMBAAGjYDBeMB8GA1UdIwQYMBaAFDZfYf5qHv280mbA81Gf1Voks6m0MB0G\n"\
"A1UdDgQWBBRIWOUqiCP/244owu9bCZhGnQz6BTAMBgNVHRMBAf8EAjAAMA4GA1Ud\n"\
"DwEB/wQEAwIHgDANBgkqhkiG9w0BAQsFAAOCAQEAc5FEp1cPw5xhC6fZf7fHlYNW\n"\
"nF2TvccVIybSQZP08QvOMb0uabDnL2GA0+Yy73eRCj5tBYxfXHVJU7D7o7z/2r1H\n"\
"m4z3T/PFSlQ4HQe5kCQ6n3nR1H/luVNlkGF6H+N2MURBFV3eGrEOwc1bPICCbcMr\n"\
"aHA+7xvvi8mt7+VQlc3a+JieNoPhvXP7ruDbsnKQ50z+KepFFEJdEsTgIRhiImcQ\n"\
"Sya+FPwbiVS9cRJ888ZhvwHnrualtBYYsWgQhFsoYhCrGZGby2kRS1qO92wV5A7D\n"\
"nx5T2hqv4W55qXD7UHcgoU9NEbYJ9ET/TGoAOcavEiAAL5M9RvDsJcqBjnhgbg==\n"\
"-----END CERTIFICATE-----"
/*
* PEM-encoded client private key.
*
* Must include the PEM header and footer:
* "-----BEGIN RSA PRIVATE KEY-----\n"\
* "...base64 data...\n"\
* "-----END RSA PRIVATE KEY-----"
*/
#define keyCLIENT_PRIVATE_KEY_PEM \
"-----BEGIN RSA PRIVATE KEY-----\n"\
"MIIEogIBAAKCAQEAvqZu73K/1zSqZDRyrRB0WgaQxtFA1cpmfm0Q5CkGAuHQDFku\n"\
"X2vHHQwV1R+yIo37GCY4cWk2J0j6zE8f0MeulJcLX+MRLLzUJZ9AAqVJkIoNfrwC\n"\
"UOIMTGREy2sOpN8TzILvg0KrUZ1DhzCwr75LLYlac/ErOlbeQcHmiKCXm+MRBlJ5\n"\
"1QAii/HvwBgtwFj4B+Zg8OkCT1vM+aqEFrCsbD15gbY3mEZuPMvj5T1wM+1lCQKQ\n"\
"qtknW7UEKmaihKxRYpGcFc8k1kOow5CfnzsgTVZSWIRpUisHoQkpPBhMXJ9Jd2T8\n"\
"q/DNv/sRD2vWFbV9QYlwQXRg2IFbqbRT1DMAlwIDAQABAoIBAHtNhR4UZwK2qyUX\n"\
"zrsjkzzOVkBgAcDDoLpPq8cEyUB/72CtJRLNz/93W391em5iyWz6+zJOS5no11vg\n"\
"KMGNA9pSPicL4oW+sq6KaonLgOKchtHaotzQ9gdKKjwtx1DWfiWc4W9RZR/pyEzd\n"\
Console
Problems
Memory
News

9.8. Ensure the kit is connected to your PC and KitProg3 is set to CMSIS-DAP Bulk mode ([LED2] always on) and identify the port number

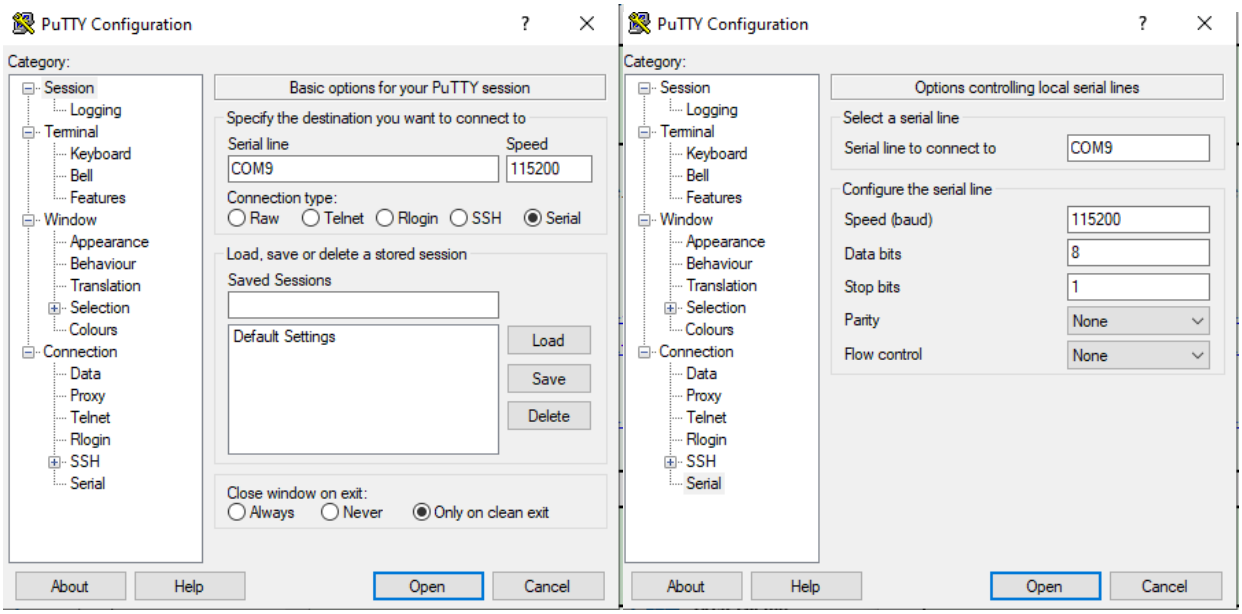
NOTE: IN WINDOWS, OPEN DEVICE MANAGER AND EXPAND PORTS (COM & LPT) TO IDENTIFY THE PORT NUMBER



9.9. Start a serial terminal program (Putty or teraterm for example). Open a connection to your board with the below settings:

- Baud Rate: 115200
- Data: 8 bit
- Parity: None
- Stop bits: 1
- Flow control: None

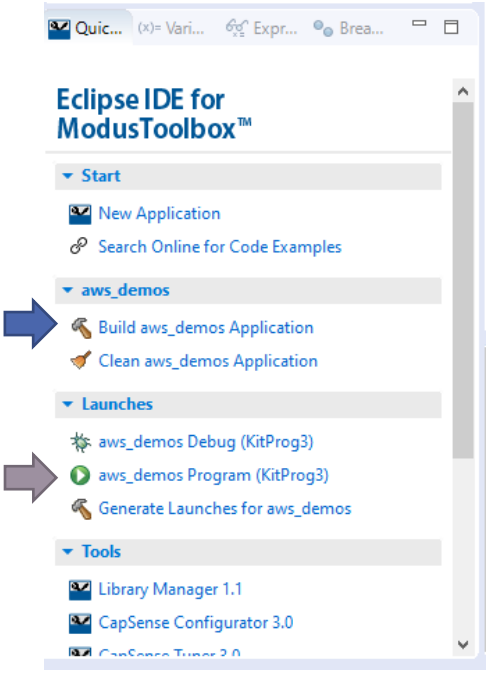
Note: A terminal window with a blank black background will appear on your Windows based computer screen.



10. BUILD AND RUN THE FREERTOS DEMO

10.1. From the **Quick Panel** tab in MTB, select **Build aws_demos Application**

10.2. Once the build has finished successfully without any errors, select **aws_demos Program (KitProg3).** This will program the CY8CKIT_064S0S2_4343W target board and the demo application will start running once programming has finished



NOTE: IF THE BUILD FAILS WITH ERRORS, PLEASE REFER TO SECTION 12 “POTENTIAL MODUSTOOLBOX BUILD FAILURE” FOR POTENTIAL SOLUTIONS

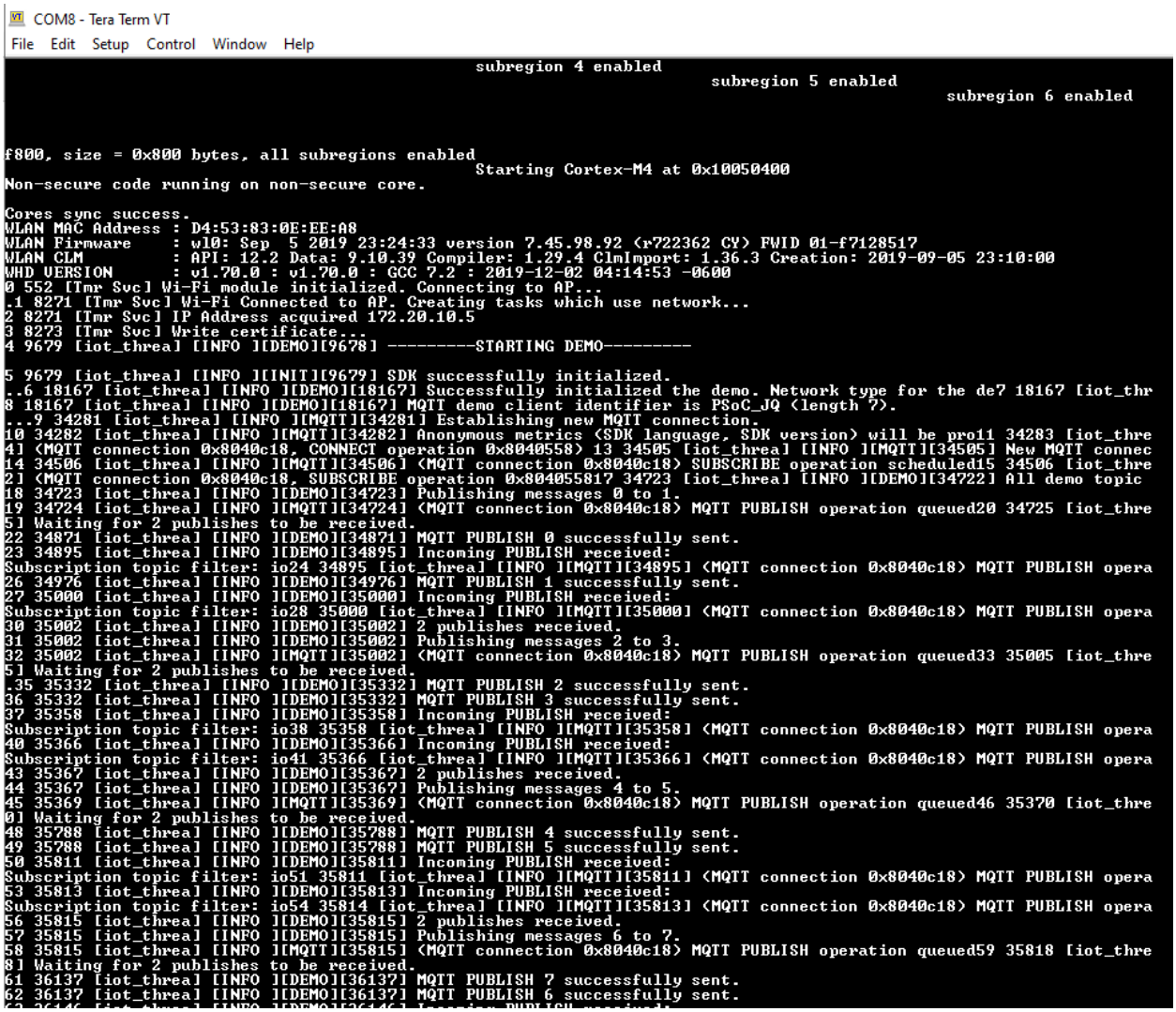
10.3. View the status of the running application in the serial terminal and notice the following initialization steps on the target device:

a) The device connects to the configured Wi-Fi access point (AP) and acquires an IP address

b) Once an internet connection is made the device is provisioned to the AWS cloud with the configured endpoint address, certificate, and private key

c) Then the FreeRTOS MQTT library is initialized and the device establishes MQTT connection with the AWS IoT MQTT broker to periodically publish and receive messages on a specified topic

The following figure shows a section of the terminal output



- If the process fails, note the numbers on the left of the terminal to determine where the first issue shows up. Note the numbers are not all aligned to the left edge.
  - If it stops at 8 or fails in 9, try connecting your PSoC 64 kit to a different network
  - If it stops at 13, check the policy attached to your Thing in AWS under the IoT Console
- The MQTT demo publishes messages on four different topics (iotdemo/topic/n where n=4) and subscribes to all of the topics to receive the same message back. When the board receives a message from AWS, it will publish an acknowledgment message on the topic iotdemo/acknowledgements.

- The CY8CKIT_064S0S2_4343W will continue to loop through the publish, receive, and acknowledge cycle for all 4 topics and then end the demo
- Note: If you choose to re-run the demo, Press and hold SW1 "XRES" on your kit for a couple seconds then release. We have found some WiFi Access Points may reject a reconnect if the request to connect happens before the Access Point realizes the previous connect was terminated.
- Note. The demo can also be re-run by reloading the program from ModusToolbox per step 10.2 above.

11. MONITORING MQTT MESSAGES ON AWS

You can use the MQTT client in the AWS IoT console to view the messages that the device is publishing to the AWS cloud. To subscribe to the MQTT topic, follow these steps:

11.1. Sign in to the AWS IoT console

11.2. In the navigation pane, choose **Test** to open the MQTT client

11.3. In the **Subscription topic** textbox enter iotdemo/#

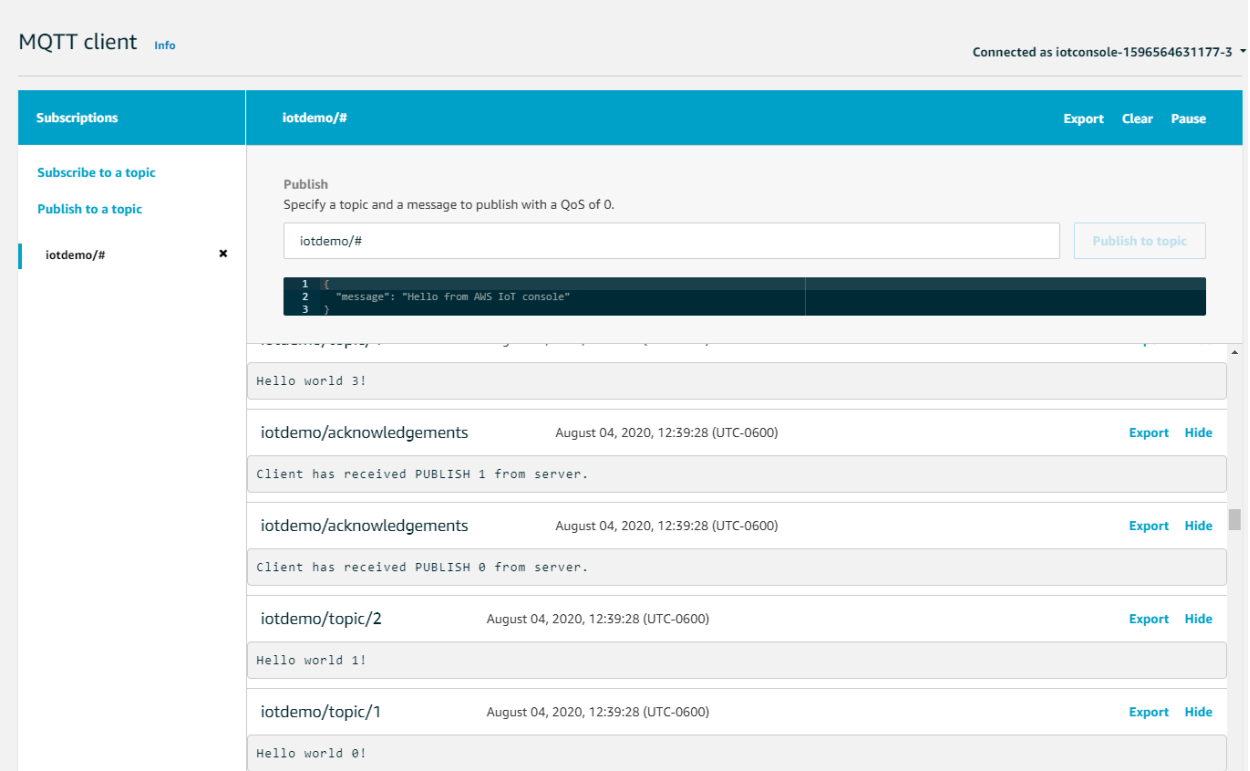
11.4. Under **Quality of Service**, choose **1**

11.5. Under **MQTT payload display** choose **Display payloads as strings (more accurate)**

11.6. Choose **Subscribe to topic** and then reset the kit with [SW1] to restart the demo

11.7. The hash (#) symbol at the end of a topic acts as a wildcard. This demo for example will have the MQTT client in AWS receive any messages published to any topic that begin with iotdemo/#

- Specifically the kit will publish the messages "*Hello World (n) !*" on 4 separate topics named iotdemo/topic/1-4
- When the kit receives the corresponding messages from the AWS server it will publish another set of messages on the iotdemo/acknowledgements topic
- The published messages can be seen in the MQTT client window



The Hello World MQTT demo is enabled by default, but the following demo applications have been tested and verified to work with the current release. These demos can be found under the *<freertos>/demos* directory

- Bluetooth Low Energy demo
- Over-the-Air Updates demo
- Secure Sockets Echo Client demo
- AWS IoT Device Shadow demo

To enable a demo application open: *<freertos>/vendors/cypress/boards/board/CY8CKIT_064S0S2_4343W/aws_demos/config_files/aws_demo_config.h* and define the demo that you want to run. Please refer to page 240 in the FreeRTOS User Guide for more information on how to run these demos.

12. APPENDIX

Please check in this Appendix for methods to get around common issues and misunderstandings.

If you have additional tips, tricks, clarifications, or suggestions, please e-mail them to psoc64@arrow.com

MODUSTOOLBOX INSTALLATION ISSUES

If ModusToolbox is installed on a computer for a single user, other users of that computer may not have direct access to start menu shortcuts.

Shortcuts and links can be copied from the directories associated with the user who did the installation. Shortcuts may be found at the following link:

C:\Users\<user_name>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\ModusToolbox 2.2

Alternatively run as the user who installed ModusToolbox.

Note If you install ModusToolbox in a non-default location, you will need to set the CY_TOOLS_PATHS environment variable for your system to point to the /ModusToolbox/tools_2.2 folder, or set that variable in each Makefile. You must use forward slashes in the variable's path, even in Windows. Refer to the "Product Versioning" section in the ModusToolbox User Guide.

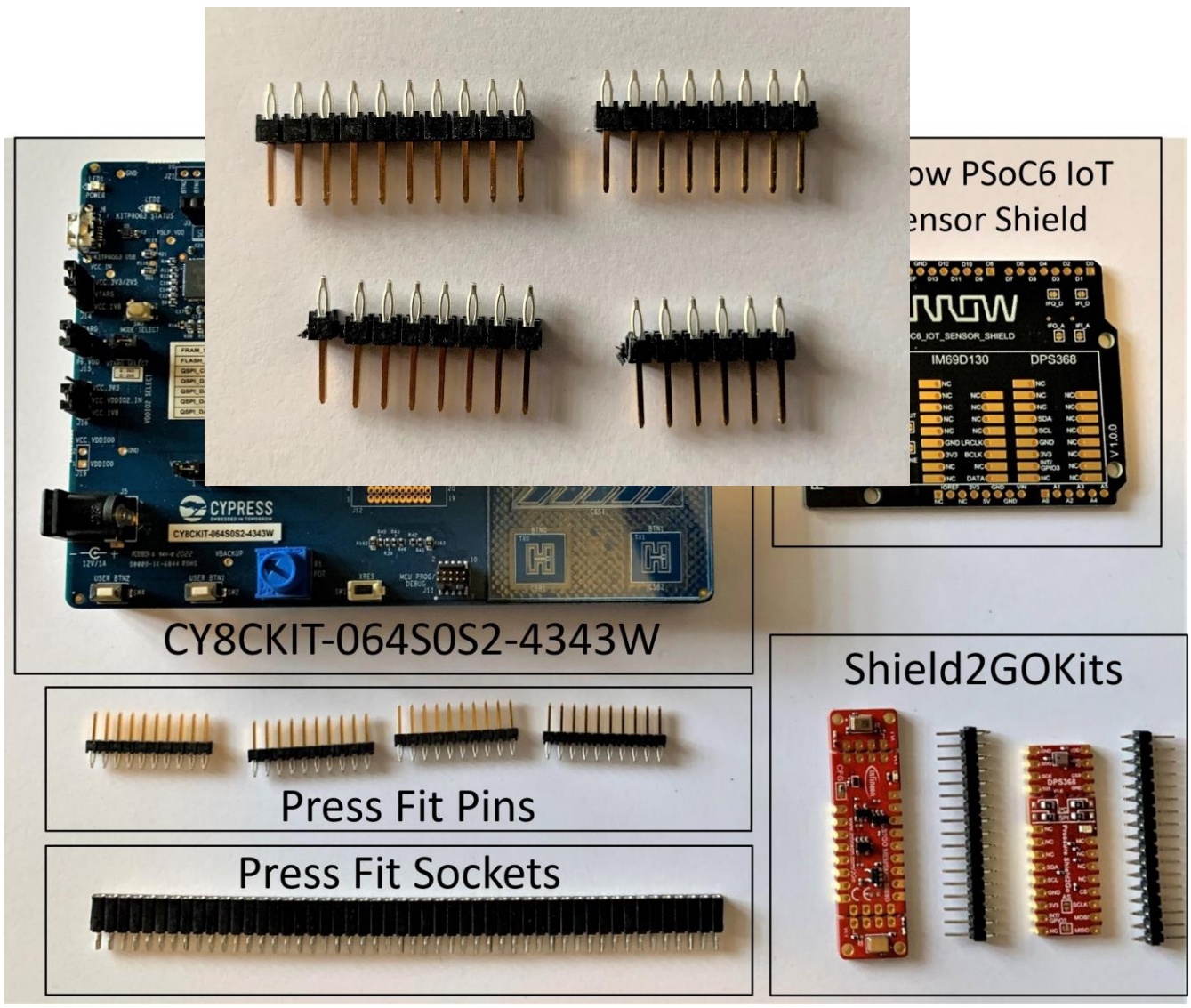
ASSEMBLY INSTRUCTIONS FOR THE KIT USED IN THE P64 SECURITY WEBINARS

The kit contains the follow items:

1 – CY8CKIT-064S0S2-4343W: PSoC64 Standard Secure – AWS WiFi Pioneer Kit

1 – Arrow PSoC6 IoT Sensor Shield

1 – DPS386 Shield2Go Kit for reading barometric pressure



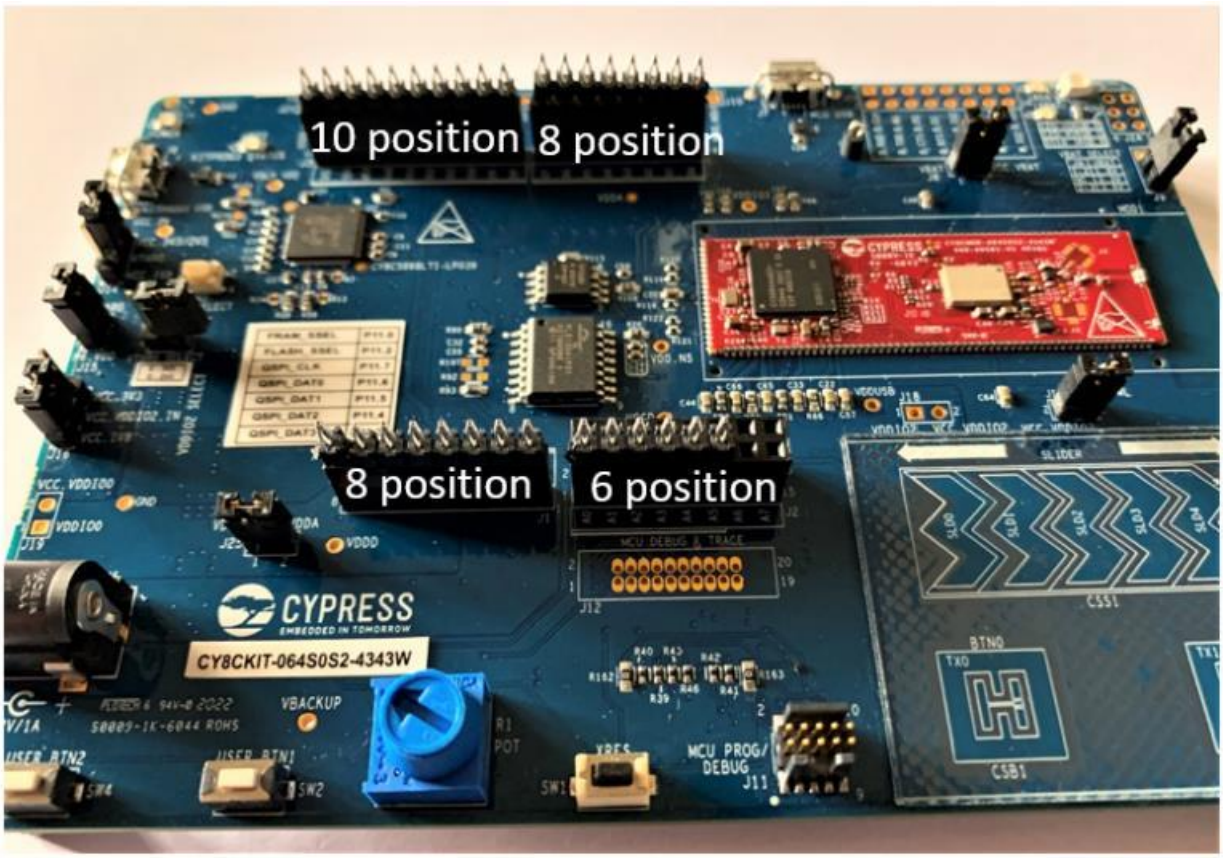
1 – IM69D130 Shield2Go kit (MEMS Microphone)

4 – 10 position press fit pins

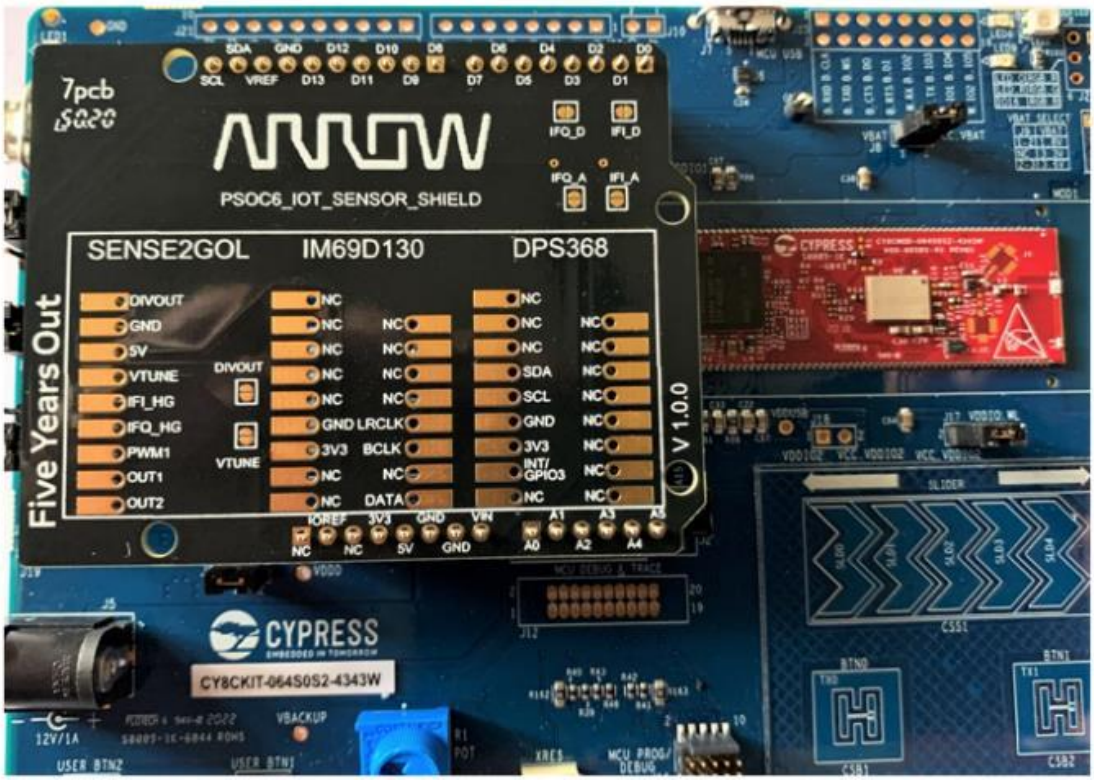
1 – Strip of press fit sockets

1. One 10 position, two 8 position and one 6 position press fit pins are needed. Leave 1 of the 10 position press fits intact. Break of 2 pins of off 2 of the other 10 pin press fit connectors. Break off 4 pins on the last 10 pin press fit connector. See below

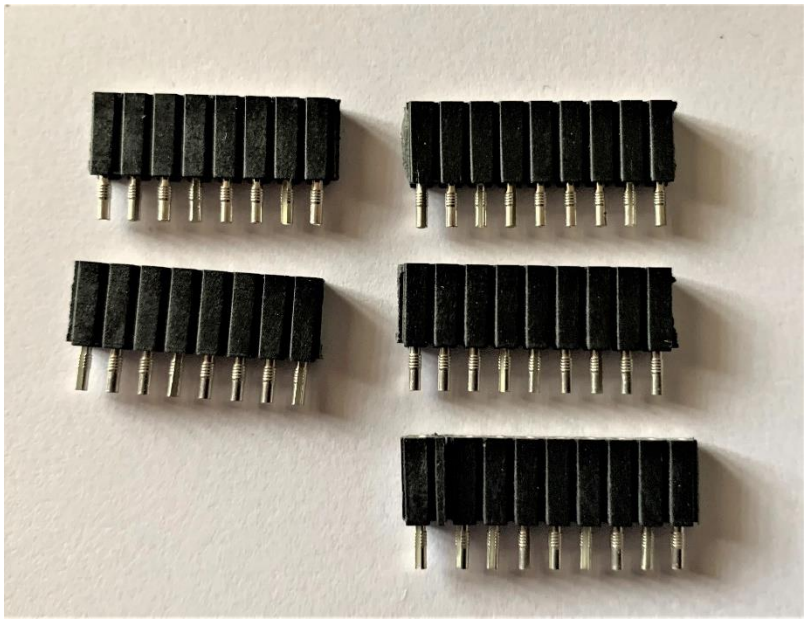
2. Put the press fit pins on the Pioneer Kit



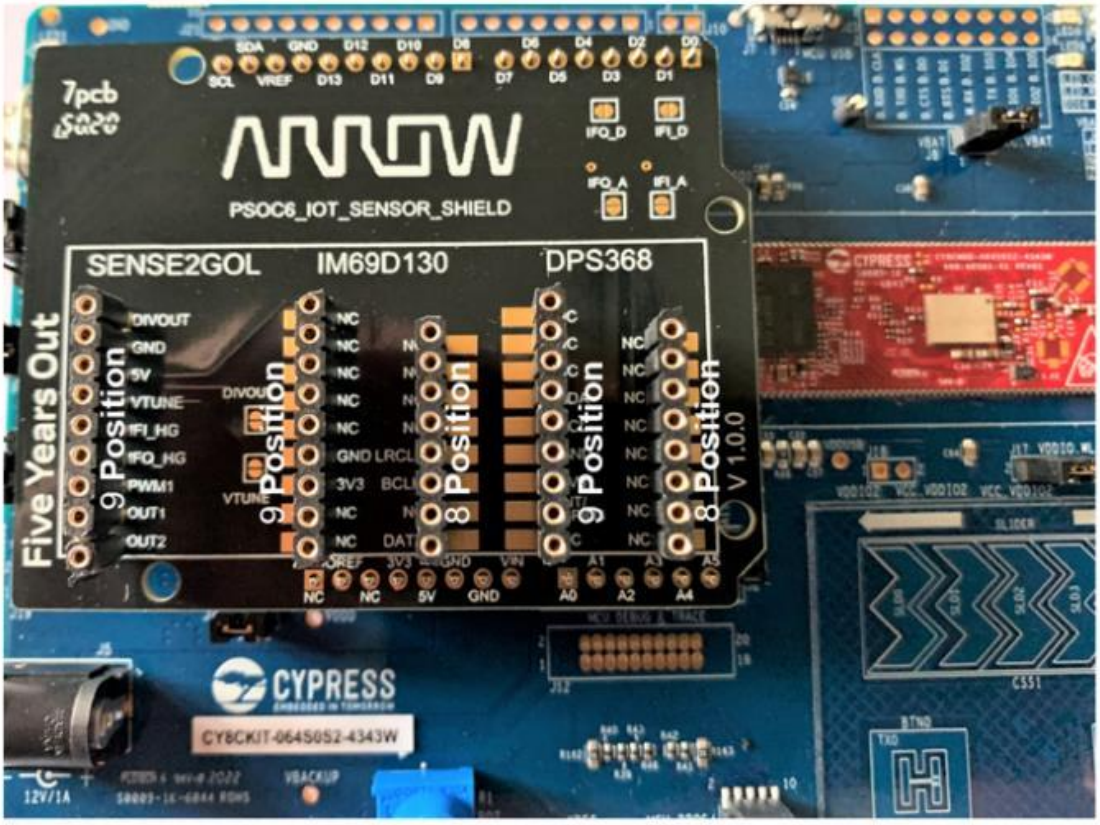
3. Mount the Arrow PSoC6 IoT Sensor Shield and press it down on the press fit pins



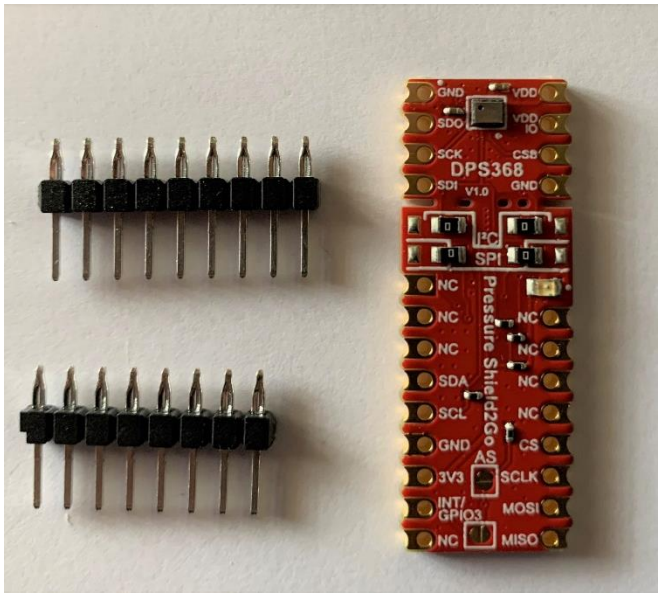
4. Break the press fit socket strip into three 9 position strips and two 8 position strips.



5. Place sockets on the Arrow PSoC6 IoT Sensor Shield

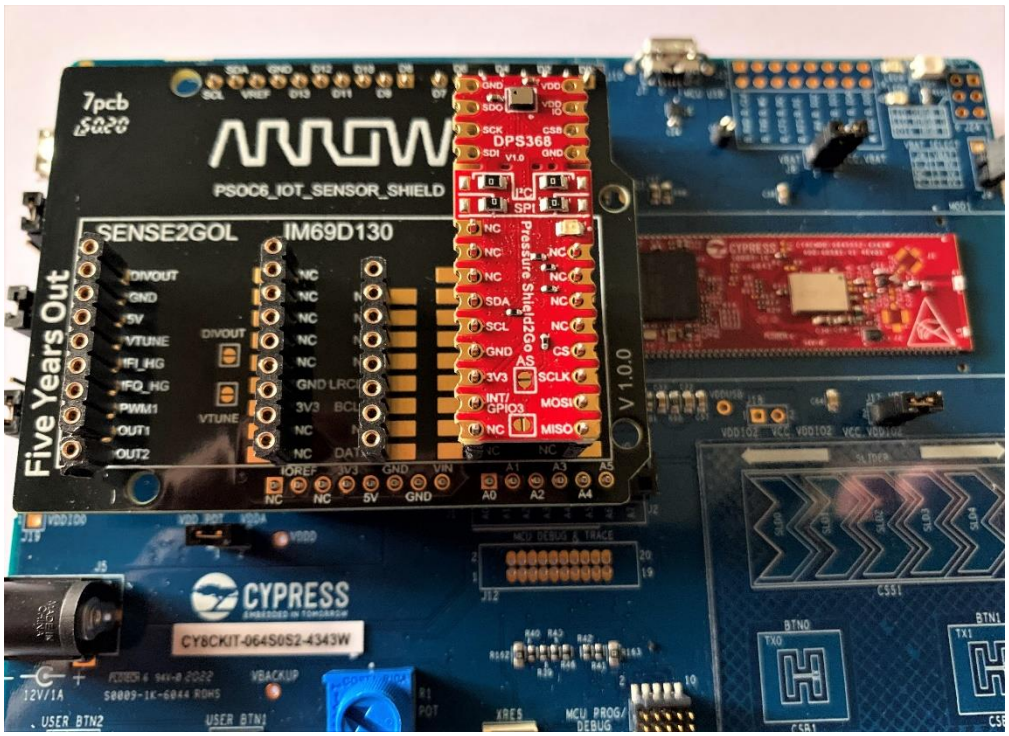


6. Break the press fit pins for the DPS368 Shield2Go Kit into one 8 position row and one 9 position row

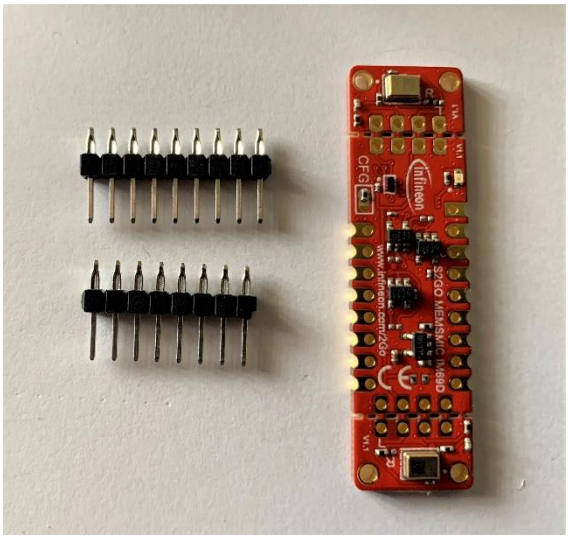


7. Place the pin press fit up in the DPS368 socket on the Arrow PSoC6 IoT Sensor Shield

8. Place the DPS368 Shield2Go on the press fit pins components side up.

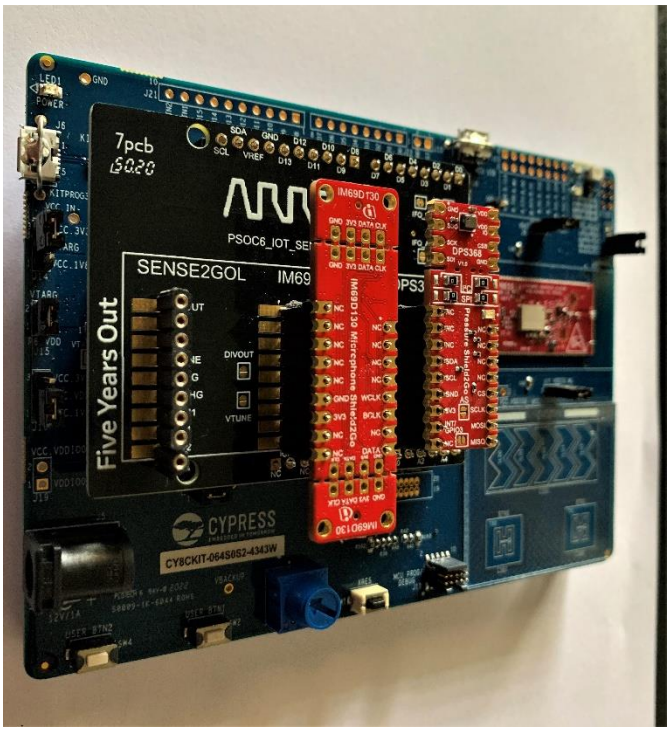


9. Break the press fit pins for the IM69D130 Shield2Go Kit into one 8 position row and one 9 position row

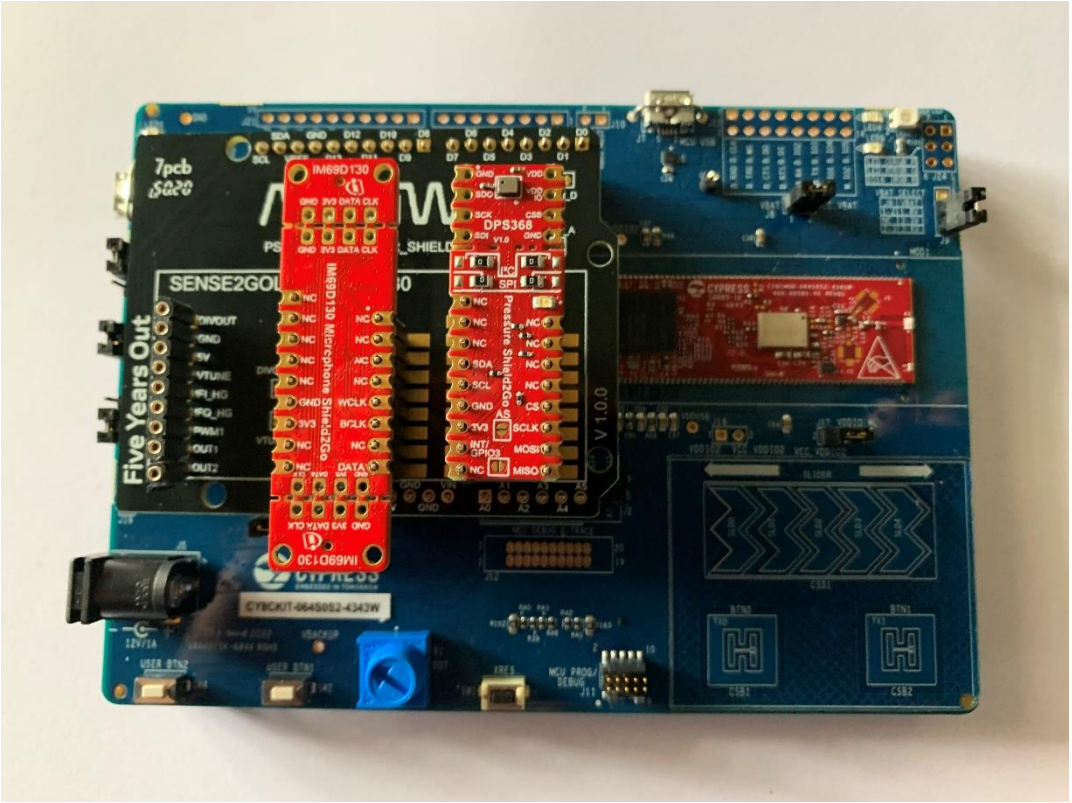


10. Place the pin press fit up in the IM69D130 socket on the Arrow PSoC6 IoT Sensor Shield

11. Place the IM69D130 Shield2Go board (component side down) on the press fit pins



12. This is the finished assembly.



TIPS & TRICKS

Modus Shell useful commands and knowledge.

Note: If using a Linux Operating System, use Terminal instead of Modus Shell.

Note: Modus Shell accepts Linux type commands. For comparison to Windows Command Prompt, use the "whereis" command in front of a couple commands to see how environmental paths may be set to point at different installations of similar tools. Start with "whereis python"

- *where <command>* ➔ *show where the program to run a <command> is located. If multiple programs are listed, they will be attempted in order.*
- *whereis <command>* ➔ *Show where the program and related .dll files to run a <command> are located. If multiple programs are listed, they will be attempted in order.*
- *pwd* ➔ *To know what directory you're in within Modus Shell*
- *cd..* ➔ *To move up a directory,*
- *pip list* ➔ *Useful to see all the tools installed with python along with their versions*

Note: The top level C:/ directory will appear as /cygdrive/c/

Note: When navigating directories within Modus Shell, the Forward Slash (/) must be used

SOFTWARE SETUP

In order to successfully provision the PSoC 64, please ensure that all the below requirements are met prior to installing the additional tools for the chosen software environment. The provisioning process and environment setup was done using a 64-bit Windows 10 machine with Python v3.7. MacOS and Linux host machines are supported as well.

NOTE: PYTHON IS PRE-INSTALLED WITH MODUSTOOLBOX 2.2. ALL COMMANDS RUN FROM AN INTERNAL TERMINAL WINDOW SHOULD USE MODUS SHELL, WHICH HAS LINKS TO THE CORRECT TOOLS. WINDOWS COMMAND PROMPT DOES NOT HAVE THE CORRECT LINKS. THERE IS, HOWEVER, ONE CASE WITHIN THE AWS_DEMO PROJECT WHERE PYTHON AND CYSECURETOOLS ARE SEARCHED FOR IN THE WINDOWS ENVIRONMENT. TO ENSURE THE AWS_DEMO'S MAKE FILE FINDS PYTHON AND CYSECURETOOLS, EITHER SET THE WINDOWS ENVIRONMENT VARIABLES TO POINT TO PYTHON IN MODUSTOOLBOX/TOOLS_2.X OR INSTALL PYTHON WITH CYSECURETOOLS VERSION 2.X IN WINDOWS.

- Install Python3.x on the Windows based Host PC (Note: Python version 3.7 has been verified)
  - https://www.python.org/downloads/
  - Change the install directory to *C:/Python37* during installation and ensure python.exe file location is added to the system path: *C:/Python37/python.exe*
    Note: The length of the default Python37 install directory can cause path file length issues. The default location is *C:/User/<username>/AppData/ Local/ Programs/Python/Python37/python.exe*
- Install Python 3.x on Linux based computer
  - Note: Type "python" in a Terminal window on a Linux based computer. Python may be installed as "python3". Calls to python may need to be changed to python3. Alternatively, there are ways to create an alias for python to call python3. That is currently for advanced users to ensure the alias doesn't prevent other programs from finding a different version of python if required.
  - If python3 is not set as default, run the following commands. The number at the end of each command denotes a priority:

    update-alternatives --install /usr/bin/python python /usr/bin/python2.7 1

    update-alternatives --install /usr/bin/python python /usr/bin/python3.7 2

  - If a version of python 3.x is not installed, find correct version @ www.python.org under Downloads
  - Install pip from a Linux Terminal window by navigating to the directory where Python3 resides then using the command "sudo apt install python3-pip"

- If previous versions of python are installed on the Host PC ensure Python37 has a higher priority in system Path.
- ModusToolbox version 2.2 for Windows includes Python v3.7
- ModusToolbox version 2.2 for Windows includes cysecuretools
  - For these instructions and the initial PSoC 64 kits, cysecuretools 2.x must be used.
  - Check the version of cysecuretools by running either "pip list" or "pip show cysecuretools"
  - If a higher version is installed, uninstall then re-install a correct version.
    - To install a specific version, un-install then re-install the desired version

- pip uninstall <tool>
- pip install '<tool>==x.x' ← x.x = the first two fields of the version

COMMUNICATION WITH KIT

Note: The USB to serial communication function on Infineon/Cypress PSoC kits is referred to as KitProg. Reference KitProg User Guide @ https://www.cypress.com/documentation/development-kitsboards/kitprog-user-guide

Open FW-Loader

FW-Loader is installed with ModusToolbox 2.2. Find FW-loader in the tools directory or in the Windows start menu. FW-Loader is a program that runs in Modus Shell opened in the directory where FW-Loader resides.
With the kit plugged into your computer, type:

./fw-loader –device-list

Note: There are four modes for the kit: kp3-hid; kp3-bulk; kp3-bootloader; and kp3-daplink
The different modes represent different protocols of communication.
Following are the commands to select each mode.

./fw-loader --mode kp3-hid
./fw-loader --mode kp3-bulk
./fw-loader --mode kp3-bootloader
./fw-loader --mode kp3-daplink

Windows Drivers

A common error when trying to communicate with the kit is an incompatible Windows Device Driver. Open Device Manager in a Windows Operating system and see how the kit shows up in each KitProg mode.
If you see a yellow caution triangle in the Device Manager for any individual mode, you'll need to uninstall the driver AND remove all software before unplugging and re-plugging the kit back in.

Further explanation:

Each kit has a USB to Serial interface that is implemented with a PSoC 5 chip near the USB port marked Kitprog3 on the kit. The PSoC 5 is running code called "Kitprog version 3" or "Kitprog3". There are four (4) modes that the Kitprog can run in to communication over the USB port. Some modes can be selected by pressing the "Mode Select" button near the PSoC 5 chip. Modes are indicated by the Kitprog Status LED2.

POTENTIAL PROVISIONING FAILURES

"Waiting for a debug probe"

If the process appears to stop with the line, "Waiting for a debug probe to be connected…", do the following: Open a new fw-loader window by clicking on with windows start and typing fw-loader. From within the fw-loader window, type "./fw-loader –device-list" and note the state of the kit. Type "./fw-loader –mode kp3-hid" to change the mode of the kit and see if the process in the modus shell starts up.

"SWD/JTAG Transfer Fault"

If the process fails after checking for cm0 AP permissions, there is a solution although the specific issue has not been identified. Run the "reprov_helper.py" program found in the freeRTOS security folder. It will fail, but not before it clears out some memory locations that may be causing the "SWD/JTAG Transfer Fault"

Open a Modus Shell

Navigate to the security folder @ <freeRTOS>/vendors/cypress/MTB/psoc6/psoc64tfm/security

Identify the full path to fw-loader included with ModusToolbox. Make sure to use forward slashes.

C:/Users/<user_name>/ModusToolbox/tools_2.2/fw-loader

Verify the path, insert your user_name then run reprov_helper by typing:

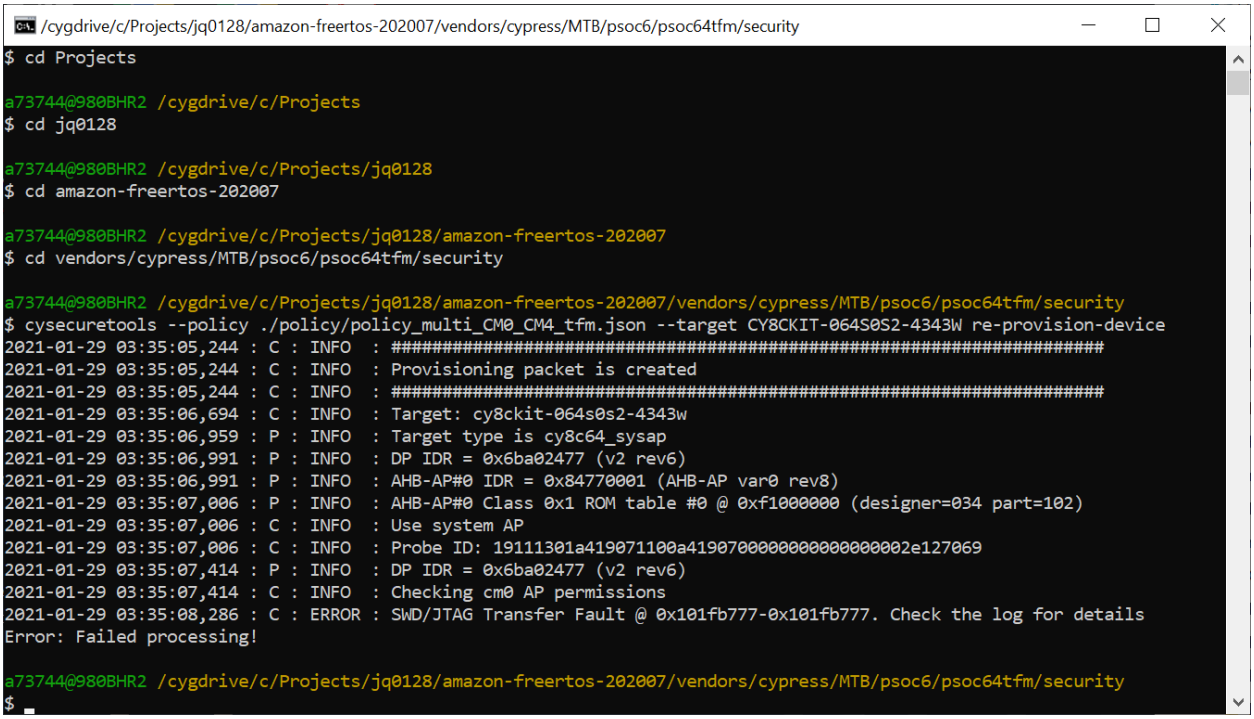
python reprov_helper.py -f C:/Users/<user_name>/ModusToolbox/tools_2.2/fw-loader -y

- Enter **Y** for the first two prompts
- Enter a short sequence of numbers (12345678) for a unique serial number
- Enter **Y** for the last prompt

The reprov helper script will run through many commands and eventually fails due to a file not found while looking for a rootCA.key.

After running reprov_helper, re-run the previous provisioning command in the "Provision the Board" section of this document.

Following is an image of the "SWD/JTAG Transfer Fault" error



POTENTIAL MODUSTOOLBOX BUILD FAILURE

“cysecuretools not found”

A failure to build with a complaint: “cysecuretools not found” is likely due to a make file in the AWS_Demo pointing to python and cysecuretools outside of ModusToolbox.

Test this by typing the following commands in both a Modus Shell and a Windows Command Prompt:

where python

where cysecuretools

The returned path will disclose where each internal terminal is finding the referenced tools.

The AWS_Demo project that comes with freeRTOS relies on a .mk file that runs a postbuild python script. Unfortunately, this older .mk file calls scripts from a bash interface – not the modus-shell interface.

What this means is that the postbuild script will not find the cysecuretools or python packaged with modus-shell for new versions of ModusToolbox. To resolve this, either install python 3.7.9 and cysecuretools==2.1.0 in windows environment using Command Prompt or set the Windows Environment Variable to point to Python within ModusToolbox/Tools_x.x.

No rule to make target

Seeing the following error in the ModusToolbox console after attempting a build of AWS_Demo might imply an incorrect or incomplete version of freeRTOS was downloaded. To resolve this issue, download freeRTOS version 202007 again using the "git" command and making sure to enable recursive mode so it pulls everything.

```
make: *** No rule to make target
'../../../../../freertos_kernel/portable/MemMang/heap_4.c', needed by
'C:/Projects/JQ0128/amazon-freertos-
202007/build/cy/aws_demos/CY8CKIT_064S0S2_4343W/Debug/user/freertos_kernel/portable/M
emMang/heap_4.o'.  Stop.
make: *** Waiting for unfinished jobs....
```

POTENTIAL RUNNING FAILURES

Watch the Tera-Term, Putty or other terminal output after plugging in or resetting the kit

- Failure after step 7 may be due to a parameter in iot_pkcs11_config.h set to 1

- Failure after step 8 may be due to un-initialized network. See WiFi Network initialization

- Failure at step 13 may be due to incorrect Policy in AWS for the Thing

13. REVISION HISTORY

Revision #	Date	Editor	Note
1.0	2/4/2021	G Carson & V Pea	First full released version
1.1	2/5/2021	G Carson	All notes incorporated and changes accepted
1.2	2/8/2021	V Pea	Updated materials list and numbered sections
1.3	2/21/2021	M Roberts, R Meyer, & G Carson	Added Kit Assembly instructions and incorporated recommendations to procedure steps.
1.4	3/8/2021	V Pea	Updated formatting
1.5	3/9/2021	G Carson, A Giday	Changed image of AWS Things, added image of PSoC 64 kit and Notes how to change modes,
1.6x	Draft	G Carson	Moved Notes before steps, Added Python for Windows setup, moved freeRTOS download to front, other minor fixes