



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	3
Document structure.....	3
Materials.....	3
References	4
PSoC 64 Provisioning Tools	4
Check the appendix	4
2. Prerequisites	5
Overview.....	5
Knowledge and Artifacts (if you have all these, skip this chapter).....	5
Install the latest version of ModusToolbox.....	5
Python Setup in Windows.....	6
Setup a project folder	7
Freertos download.....	8
Create an AWS Account	9
AWS First Steps simplified.....	9
Create IAM User.....	11
Set Permissions.....	13
Verify Your IAM User	16
3. Provisioning the PSoC 64	17
Overview.....	17
Knowledge and Artifacts (if you have all these, skip this chapter).....	17
Provisioning Flow.....	17
4. AWS Credential Setup.....	21
Overview.....	21
Knowledge and Artifacts (if you have all these, skip this chapter).....	21
Create a Policy	21
Register the Device	23
5. Download FreeRTOS (The contents of this section have been merged into section 2)	25
6. Provision the board.....	26
Overview.....	26
Knowledge and Artifacts (if you have all these, skip this chapter).....	26
Setup the Provisioning Keys.....	26
Prep the Kit for Provisioning.....	29

Provision or Re-Provision the Kit.....	31
7. Configure the AWS Demos.....	33
Overview.....	33
Knowledge and Artifacts (if you have all these, skip this chapter).....	33
Setup the Provisioning Keys.....	33
8. Configuring Your AWS IoT Endpoint and WiFi Credentials.....	34
Overview.....	34
Knowledge and Artifacts (if you have all these, skip this chapter).....	34
Determine the AWS Endpoint Identifier for your THing	34
Open the project defines for Endpoint ID and WiFi Login Credentials	34
9. Formatting the AWS IoT Credentials.....	37
Overview.....	37
Knowledge and Artifacts (if you have all these, skip this chapter).....	37
Update Defines in project for the AWS linkage.....	37
10. Build and Run the FreeRTOS Demo.....	40
Overview.....	40
Knowledge and Artifacts (if you have all these, skip this chapter).....	40
Build the Project	40
View execution steps of the Project	40
11. Monitoring MQTT Messages on AWS.....	42
Overview.....	42
Knowledge and Artifacts (if you have all these, skip this chapter).....	42
Subscribe to incoming messages	42
12. Appendix	45
ModusToolbox Installation issues.....	45
Assembly Instructions for the Kit used in the P64 Security Webinars	45
Tips & Tricks.....	53
Software Setup.....	54
Communication with Kit	55
Potential Provisioning Failures.....	56
Modus Shell Output from Provisioning REv*A kit.....	58
Potential Modustoolbox build failure	61
Potential Running Failures	62
13. Revision History	62

1. INTRODUCTION

This section has no steps to perform. This section provides an overview of this Pre-Work Guide to help users navigate quickly with minimal issues.

DOCUMENT STRUCTURE

This introductory document is broken into chapters. Each Chapter starts with an overview of what the chapter covers, followed by a list of what knowledge and artifacts should be in your possession after completing the chapter. If you already have the knowledge and artifacts, skip the chapter.

Individual instructions are prefaced with a chapter number followed by a step number. Executing each instructions in sequence without reading the material should get you through the manual successfully.

MATERIALS

- Workshop Kit:
 - [PSoC 64 Secure AWS IoT Pioneer Kit \(CY8CKIT-064S0S2-4343W\)](#) check version per Note below
 - IM69D130 Microphone Shield2Go (S2GOMEMSMICIM69DTOBO1)
 - DPS368XTSA1 Pressure Shield2Go (S2GOPPRESSUREDPS368TOBO1)
 - 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))

Note: There are two functionally different versions of the CY8CKIT-064S0S2-4343W.

There are NO physical differences with the “New” Rev*A kits from the “Old” Rev** kits.

The new version of this kit is pre-configured to comply with updates to Infineon Software Tools.

Kit	Revision on Kit Box	Revision on Kit Circuit Board	Compatible SDK
CY8CKIT-064S0S2-4343W	Rev **	Up through Rev 08	cysecuretools version 2.x
CY8CKIT-064S0S2-4343W	Rev *A	Starting at Rev 09	cysecuretools version 3.x

Note: If attempting to run a Rev ** kit with cysecuretools version 3.x, the following message will appear:

Early Production Units detected, please get earlier version of tools by running 'pip install --upgrade --force-reinstall cysecuretools==2.1.0'

REFERENCES

Various support materials have been created to help the developer understand the capabilities of the PSoC 64 secure processor, the tools available to customize and program those capabilities and conceptual applications where the capabilities can offer security protection.

Technical Product Descriptions and Application Notes:

Documents tab @ <https://www.cypress.com/products/psoc-6-microcontrollers-32-bit-arm-cortex-m4m0>

Data Sheet = Provides specifications for physical capabilities of each component.

Technical Reference Manuals = Describes features of the component family

SDK User Guide

Training Materials

(Basics of Security) Public Key cryptography @ https://en.wikipedia.org/wiki/Public-key_cryptography

Webinars @ www.arrow.com/psoc64

Video Tutorials @ <https://www.cypress.com/training/free-online-video-training-and-tutorials-cypress>

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)

OVERVIEW

- Install ModusToolbox
- Setup Python for Windows
- Setup Project folder
- Download FreeRTOS
- Create AWS account with IAM user

KNOWLEDGE AND ARTIFACTS (IF YOU HAVE ALL THESE, SKIP THIS CHAPTER)

ModusToolbox version 2.3 or higher

In Windows Start Menu, start typing “Eclipse IDE for ModusToolbox” until an option shows with version.

Python version 3.7

Type “where python” and “python -V” in each Modus Shell and CMD prompt to verify same version ~ 3.7 in each

Project folder with amazon-freertos

Path of 20 characters or less with NO spaces to your project folder.

Using Modus Shell, navigate to your project folder and type “pwd” to see the full path

Using Command Prompt, navigate to your project folder and see the full path.

FreeRTOS Version 202007.00 for Rev** kits / Version 202107.00 for Rev*A kits

Within the freeRTOS directory is a file, “CHANGELOG.md” that contains Version info

IAM User account on AWS

You're able to log into AWS using an IAM User account.

INSTALL THE LATEST VERSION OF MODUSTOOLBOX

Navigate to the ModusToolbox WebPage to download the latest version @

<https://www.cypress.com/products/modustoolbox>

Tip: To use Chrome browser to download, it may be necessary to 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.

Tip: Download and install any update patches that are included on the ModusToolbox WebPage.

REFER TO THE MODUSTOOLBOX INSTALLATION GUIDE FOR DETAILED INSTALLATION INSTRUCTIONS!

NOTE: MULTIPLE VERSIONS OF MODUSTOOLBOX CAN COEXIST ON A COMPUTER. TO SUPPORT A REV** KIT, DOWNLOAD VERSION 2.2 OF MODUSTOOLBOX TO OBTAIN VERSION 2.X OF CYSECURETOOLS. DOWNLOAD THE CURRENTVERSION OF MODUSTOOLBOX. SEVERAL TOOLS THAT SUPPORT SECURITY ARE INCLUDED IN EACH MODUSTOOLBOX/TOOLS_2.X DIRECTORY INCLUDING:

- PYTHON
- CYSECURETOOLS
- LIBUSB

PYTHON SETUP IN WINDOWS

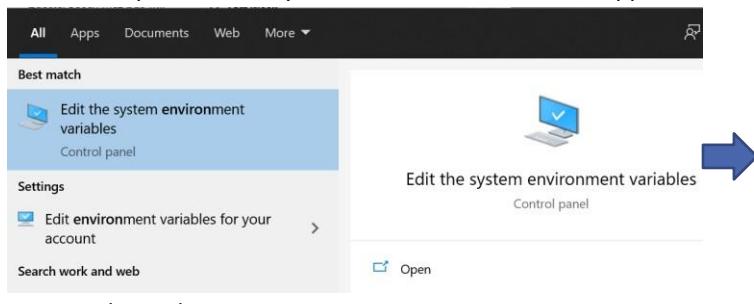
The majority of the exercises in this pre-work run in ModusToolbox or a Modus Shell. However, there may be an instance where a batch file is used that calls for a cysecuretools command from a Windows Command Prompt.

To run the same Python in Modus and Windows, there are two primary options:

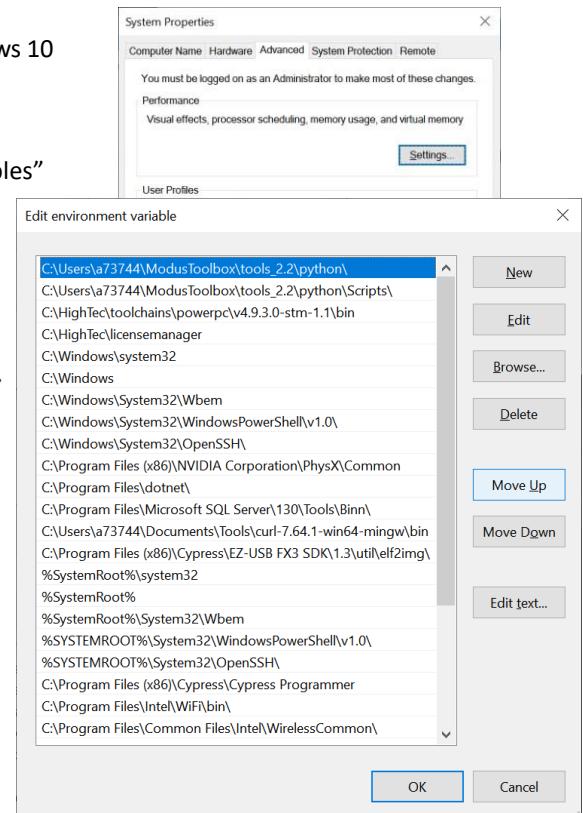
a) replicate installation of the same identical version of Python along with all necessary added tools to Windows;

b) set an environment variable in Windows to point to Python in 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\ in the latest version x.x of ModusToolbox\tools installed. Then click the [Move Up] button until the new line is at top of the path.

C:\Users\<user_name>\ModusToolbox\tools_<x.x>\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.

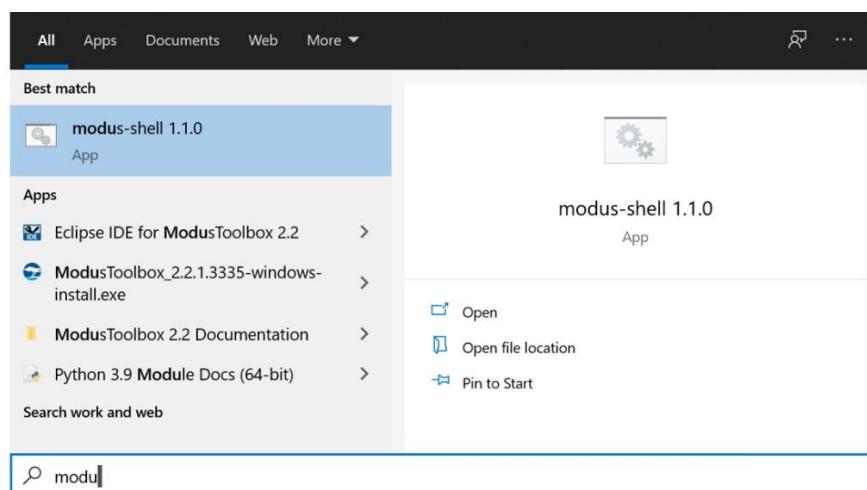
C:\Users\<user_name>\ModusToolbox\tools_<x.x>\python\

SETUP A PROJECT FOLDER

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

Tip: The documents folder in Windows should NOT be used in the path to a project folder. The pathname to the documents folder in Windows often contains extra path information that is not visible in Windows File explorer and NOT compatible with software compilers. Specifically, be aware of embedded OneDrive path names and avoid the use of spaces (replace them with an underscore “_”).

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 to be open within that new project directory.

The screenshot shows a terminal window with a black background and white text. It displays the following command sequence:

```
a73744@980BHR2 ~
$ pwd
/cygdrive/c/Users/a73744

a73744@980BHR2 ~
$ cd ..

a73744@980BHR2 /cygdrive/c/Users
$ cd ..

a73744@980BHR2 /cygdrive/c
$ mkdir Projects

a73744@980BHR2 /cygdrive/c
$ cd Projects

a73744@980BHR2 /cygdrive/c/Projects
$ mkdir P64_AWS

a73744@980BHR2 /cygdrive/c/Projects
$ cd P64_AWS

a73744@980BHR2 /cygdrive/c/Projects/P64_AWS
$ -
```

FREERTOS DOWNLOAD

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

Install a specific version of FreeRTOS to match your PSoC 64 AWS kit.

Kit	Rev on Box	Rev on Board	Compatible freeRTOS
CY8CKIT-064S0S2-4343W	Rev**	Rev 08 or lower	202007.00
CY8CKIT-064S0S2-4343W	Rev*A	Rev 09 or higher	202107.00

1.8

Install the compatible version of freeRTOS from an AWS github site by typing or copying the proper command below that matches your kit into ModusShell within the project directory just created.

For Rev** of the PSoc 64 AWS kit, download version 2020_07_00

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

For Rev*A of the PSoC 64 AWS kit, download version 2021_07_00

```
git clone --branch 202107.00 https://github.com/aws/amazon-freertos --
recursive
```

CREATE AN AWS ACCOUNT

If you already have an AWS Account with an IAM user setup, skip to section 3.

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 FREE² 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>

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.

2.2.

Follow the step in “Setting up your AWS account and permissions” or the following instructions

The screenshot shows the AWS FreeRTOS User Guide with the 'First steps' section selected. A large red circle highlights the first step, '1. Setting up your AWS account and permissions'. An orange annotation box with the text 'Follow “Setting up...” instructions only.' is overlaid on the page.

First steps

To get started using FreeRTOS with AWS IoT, you need an AWS account, an IAM user with permission to access AWS IoT and FreeRTOS cloud services. You also need to download FreeRTOS and configure your board's FreeRTOS demo project to work with AWS IoT. The following sections walk you through these requirements.

Note

If you're using the Express ESP32-DevKitC, ESP-WROVER-KIT, or the ESP32-WROOM-32SE, skip these steps and go to Getting started with the Express ESP32-DevKitC and the ESP-WROVER-KIT.

If you're using the Nordic nRF52840-DK, refer to Setting up your MCU board to get started with the Nordic nRF52840-DK.

1. Setting up your AWS account and permissions
2. Registering your MCU board with AWS IoT
3. Downloading FreeRTOS
4. Configuring the FreeRTOS demos

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.

2.4.

Select a region. Record the region selected.

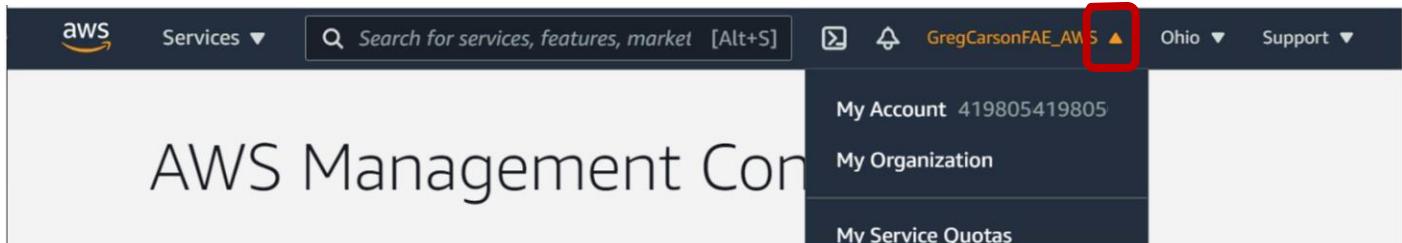
The region in the image below is us-east-2 and shows up as Ohio

The screenshot shows the AWS Management Console with the region dropdown menu open. The 'Ohio' region is highlighted with a red box. The dropdown menu lists the following regions:

- US East (N. Virginia) us-east-1
- US East (Ohio) us-east-2**
- US West (N. California) us-west-1
- US West (Oregon) us-west-2
- Africa (Cape Town) af-south-1

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 triangle next to your Account name.



CREATE IAM USER

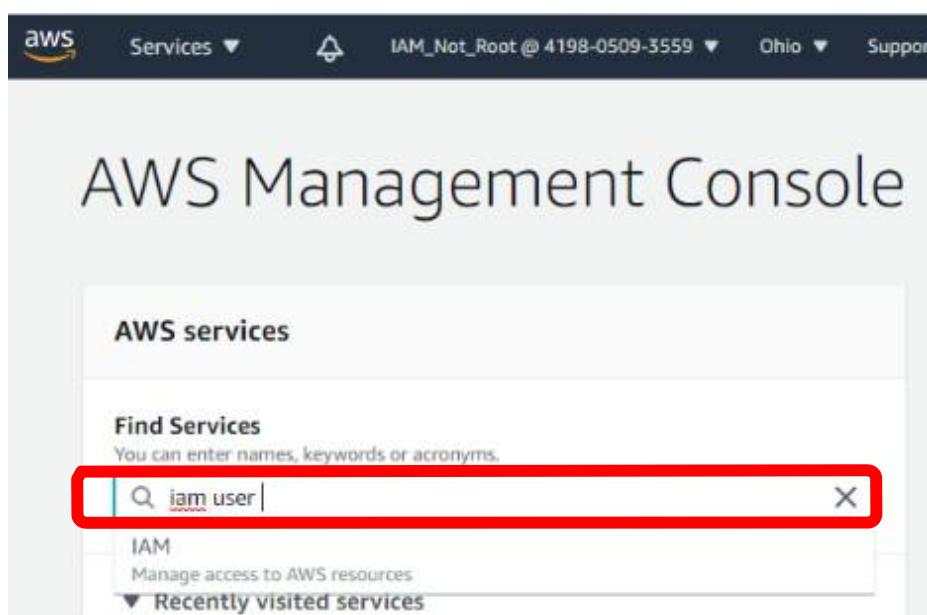
Once an AWS account is active, Add an IAM User. IAM accounts are described here: [IAM User Guide](#).

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.

The screenshot shows the AWS Identity and Access Management (IAM) dashboard. On the left, there is a navigation pane with the following structure:

- Identity and Access Management (IAM)**
- Dashboard**
- Access management** (with sub-options: Groups, **Users**, Roles, Policies, Identity providers)

The main dashboard area displays the following statistics:

- IAM resources**:
 - Users: 4 (highlighted with a red box)
 - Groups: 1
 - Roles: 2
 - Identity providers: 0
- Customer managed policies: 1**
- Security alerts**: A note indicating "The root user for this account does not have Multi Factor Authentication (MFA)."

2.9. Create a new IAM user by selecting “Add User”

The screenshot shows the AWS IAM Users page. The left navigation pane includes:

- Identity and Access Management (IAM)**
- Dashboard**
- Access management** (with sub-options: Groups, **Add user**, Delete user)

The main content area features a search bar labeled "Find users by username or access key" and a table header with columns: User name ▾ Groups Access key ▾. A prominent blue button labeled "Add user" is highlighted with a red box.

2.10. Create a User Name

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

2.11. 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.

Add user

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name*

[+ Add another user](#)

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type* **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 the AWS Management Console.

Console password* Autogenerated password
 Custom password

Require password reset User must create a new password at next sign-in
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

2.12. Click Next Permission

SET PERMISSIONS

2.13. 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.

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

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

- 2.16.** Click the [Next: Tags] button in the bottom right.

Add user

Set permissions

Add user to group Copy permissions from existing user **Attach existing policies directly**

Create policy

Policy name		Type	Used as
<input checked="" type="checkbox"/>	AdministratorAccess	Job function	Permissions boundary
<input type="checkbox"/>	AdministratorAccess-Amplify	AWS managed	None
<input type="checkbox"/>	AdministratorAccess-AWSElasticBeanstalk	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessDeviceSetup	AWS managed	None
<input type="checkbox"/>	AmazonPollyFullAccess	AWS managed	None

Add tags (optional)

IAM tags are key-value pairs you can add to your user. Tags can include user information, such as title. You can use the tags to organize, track, or control access for this user. [Learn more](#)

Key	Value (optional)
Disty	Arrow
Loc_State	Minnesota
User_Name	Carson
Group_Function	Test
Add new key	

You can add 46 more tags.

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

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

Add user

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	Test_User
AWS access type	Programmatic access and AWS Management Console access
Console password type	Autogenerated
Require password reset	Yes
Permissions boundary	Permissions boundary is not set

Permissions summary

The following policies will be attached to the user shown above.

Type	Name
Managed policy	AdministratorAccess
Managed policy	AmazonFreeRTOSFullAccess
Managed policy	AWSIoTFullAccess
Managed policy	IAMUserChangePassword

Tags

Cancel Previous Create user

Add user

Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://419805093559.signin.aws.amazon.com/console>

[Download .csv](#)

User	Access key ID	Secret access key	Password	Email login instructions
IAM_Not_R...	AKIAWDPSVL235OLUBM7W	***** Show	***** Show	Send email

2.19. 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.

2.20. 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**Sign in**

Root user
Account owner that performs tasks requiring unrestricted access. [Learn more](#)

IAM user
User within an account that performs daily tasks.
[Learn more](#)

Account ID (12 digits) or account alias

419805419805

Remember this account

Next

**Sign in as IAM user**

Account ID (12 digits) or account alias

419805419805

IAM user name

IAM_Not_Root

Password

.....

Sign in

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

Load Infineon/Cypress Security SDK

OVERVIEW

Setup cysecuretools on your PC to match your kit

KNOWLEDGE AND ARTIFACTS (IF YOU HAVE ALL THESE, SKIP THIS CHAPTER)

cysecuretools version 2.x if you have Kit Rev**

cysecuretools version 3.x if you have Kit Rev*A

From Modus Shell or Command Prompt, type “cysecuretools version”

PROVISIONING FLOW

The CY8CKIT-064B0S2-4343W and CY8KIT-064S0S2-4343W 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:

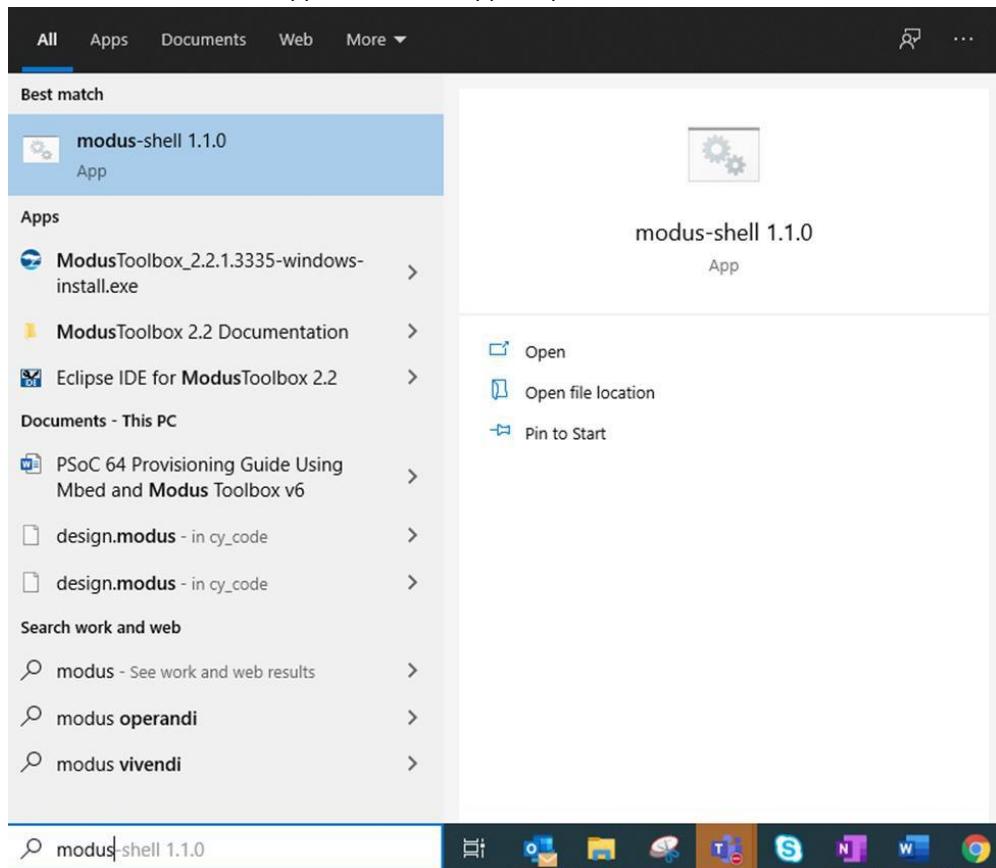
- Run entrance exam
- Send the signed provisioning package to the PSoC64

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, 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
```

Package	Version
appdirs	1.4.4
attrs	19.3.0
cbor	1.0.0
cffi	1.14.1
click	7.1.2
cmsis-pack-manager	0.2.10
colorama	0.4.3
cryptography	2.9.2
cysecuretools	2.0.0
ecdsa	0.14.1
future	0.18.2
importlib-metadata	1.7.0
intelhex	2.2.1
intervaltree	3.0.2
jsonschema	3.2.0
milksnake	0.1.5
pip	20.1
prettytable	0.7.2
psutil	5.7.2
pyasn1	0.4.8
pycparser	2.20
pyelftools	0.26
pylink-square	0.6.1
pyocd	0.27.0
pyrsistent	0.16.0
python-jose	3.2.0
pyusb	1.0.2
pywinusb	0.4.2
PyYAML	5.3.1
rsa	4.6
setuptools	46.4.0
six	1.15.0
sortedcontainers	2.2.2
sqlite-bro	0.9.1
wheel	0.34.2
winpython	2.4.20200425
zipp	3.1.0

WARNING: You are using pip version 20.1; however, version 21.0.1 is available.
You should consider upgrading via the 'C:\Users\...\ModusToolbox\tools_2.2\python\python.exe -m pip install --upgrade pip' command.

NOTE: THE ABOVE IMAGE IS OUTDATED. MANY VERSIONS HAVE BEEN UPDATED AND ADDITIONAL PACKAGES MAY EXIST.

NOTE: THE “WHERE PYTHON” COMMAND REVEALS THE ORDERED SEQUENCE OF LOCATIONS WHERE AN EXECUTABLE VERSION OF PYTHON, IF FOUND, WILL BE EXECUTED. THE FIRST LOCATION REVEALED OF PYTHON IN THE IMAGE ABOVE IS IN MODUSTOOLBOX TOOLS DIRECTORY. PYTHON WITHIN MODUSTOOLBOX WILL BE USED WHEN EXECUTING PYTHON COMMANDS FROM WITHIN MODUS SHELL REGARDLESS IF THE MACHINE HAS A SECOND VERSION OF PYTHON INSTALLED.

NOTE: IT IS TYPICALLY OK BUT NOT NECESSARY TO UPGRADE PIP (PACKAGE INSTALLER FOR PYTHON) IF REQUESTED. THE INSTALLED VERSION OF PIP 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
```

```
a20201@BNFKXT2 ~
$ pip install pyopenssl
Collecting pyopenssl
  Downloading pyOpenSSL-20.0.1-py2.py3-none-any.whl (54 kB)
    |████████| 54 kB 2.0 MB/s
Collecting cryptography>=3.2
  Downloading cryptography-3.3.1-cp36-abi3-win_amd64.whl (1.5 MB)
    |████████| 1.5 MB 6.4 MB/s
Requirement already satisfied: six>=1.5.2 in c:\users\ a20201\modustoolbox\tools_2.2\python\lib\site-packages (from pyopenssl) (1.15.0)
Requirement already satisfied: cffi>=1.12 in c:\users\ a20201\modustoolbox\tools_2.2\python\lib\site-packages (from cryptography>=3.2->pyopenssl) (1.14.1)
Requirement already satisfied: pycparser in c:\users\ a20201\modustoolbox\tools_2.2\python\lib\site-packages (from cffi>=1.12->cryptography>=3.2->pyopenssl) (2.20)
ERROR: cysecurertools 2.0.0 has requirement cryptography<3,>=2.4.2, but you'll have cryptography 3.3.1 which is incompatible.
Installing collected packages: cryptography, pyopenssl
  Attempting uninstall: cryptography
    Found existing installation: cryptography 2.9.2
    Uninstalling cryptography-2.9.2:
      Successfully uninstalled cryptography-2.9.2
Successfully installed cryptography-3.3.1 pyopenssl-20.0.1
WARNING: You are using pip version 20.1; however, version 20.3.3 is available.
You should consider upgrading via the 'C:\Users\ a20201\ModusToolbox\tools_2.2\python\python.exe -m pip install --upgrade pip' command.
```

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

FOR REV** OF PSOC 64 AWS KIT, USE CYSECURETOOLS 2.X

```
pip install cysecurertools==2.1.0
```

or

```
pip install --upgrade --force-reinstall cysecurertools==2.1.0
```

FOR REV*A OF PSOC 64 AWS KIT, USE CYSECURETOOLS 3.X

```
pip install cysecurertools
```

or

```
pip install --upgrade --force-reinstall cysecurertools==3.1.0
```

Note: cysecurertools is an SDK (Software Development Kit) from Infineon/Cypress provided as part of the Python tools set in ModusToolbox for Windows. Python and associated tools may not be preloaded in a Linux installation of ModusToolbox..

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 Thing for your IAM User with Security Keys and Policies

OVERVIEW

On AWS, create:

- Policy
- Thing
- Certificate

KNOWLEDGE AND ARTIFACTS (IF YOU HAVE ALL THESE, SKIP THIS CHAPTER)

An active Thing with a Policy and Certificate attached in your IAM User account

Certificate, Public and Private Key files for your AWS Thing on your computer

CREATE A POLICY

To Create a Thing on AWS, start by creating 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

Create a policy

Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters). To learn more about IoT policies go to the [AWS IoT Policies documentation page](#).

Name
PSoC64-Policy

Add statements

Policy statements define the types of actions that can be performed by a resource.

Basic mode

```

1  [
2    "Version": "2012-10-17",
3    "Statement": [
4      {
5        "Effect": "Allow",
6        "Action": "iot:Connect",
7        "Resource": "arn:aws:iot:us-east-2:419805419805:/*"
8      },
9      {
10        "Effect": "Allow",
11        "Action": "iot:Publish",
12        "Resource": "arn:aws:iot:us-east-2:419805419805:/*"
13      },
14      {
15        "Effect": "Allow",
16        "Action": "iot:Subscribe",
17        "Resource": "arn:aws:iot:us-east-2:419805419805:/*"
18      },
19      {
20        "Effect": "Allow",
21        "Action": "iot:Receive",
22        "Resource": "arn:aws:iot:us-east-2:419805419805:/*"
23      }
24    ]
25  ]

```

Add statement

Create

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 **Create Things**.
 - B]** If an IoT thing has been registered previously, click on the Name of the Thing.

4.14. On the Creating AWS IoT Things page, choose Create things

The screenshot shows the AWS IoT Things list page. At the top, there are navigation links: AWS IoT > Manage > Things. Below this, a heading says "Things (2) Info". A sub-instruction states: "An IoT thing is a representation and record of your physical device in the cloud. A physical device needs a thing record in order to work with AWS IoT." There are several buttons at the top: "C" (Create), "Advanced search", "Run aggregations", "Edit", "Delete", and a prominent orange "Create things" button. Below these buttons is a search bar with the placeholder "Filter things by: name, type, group, billing, or searchable attribute." To the right of the search bar are navigation arrows and a refresh icon. The main list area has columns for "Name" and "Thing type". Two items are listed:

Name	Thing type
Thing_8_31_21_01	-
P64_thing1	-

The screenshot shows the "Create things" page. At the top, there are navigation links: AWS IoT > Manage > Things > Create things. Below this, a heading says "Create things Info". A sub-instruction states: "A thing resource is a digital representation of a physical device or logical entity in AWS IoT. Your device or entity needs a thing resource in the registry to use AWS IoT features such as Device Shadows, events, jobs, and device management features." The main section is titled "Number of things to create". It contains two options:

- Create single thing**: Create a thing resource to register a device. Provision the certificate and policy necessary to allow the device to connect to AWS IoT.
- Create many things**: Create a task that creates multiple thing resources to register devices and provision the resources those devices require to connect to AWS IoT.

At the bottom of the page are "Cancel" and "Next" buttons.

4.15. Enter a name for the PSoC_64_thing, NOTE: a thing name can continue only alphanumeric, hyphens, colons, and underscores.

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

4.17. On the **Attach policies to certificate – optional** page, select the policy that you created and press the **Create thing** tab.

4.18. On the following page download the certificate, public and private key files locally on your PC, keys are automatically activated and the certificate that will be used by the PSoC_64_thing, you will be forced to download the Public key file and Private key file in order to close this page.

Note: 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.

6. PROVISION THE BOARD

Provisioning a kit loads board level security keys and permanently configures the kit for specific modes of operation.

One policy setting we have selected in this Provisioning process is the ability to re-provision. In a production environment a product is typically provisioned with a policy that does NOT allow re-provisioning.

Infineon has a python script that can be run on an un-provisioned kit to ensure it hasn't been tampered before provisioning. That test expectedly fails on provisioned kit. Therefore, there's a separate process for provisioning a kit the first time versus re-provisioning a kit.

Note: Multiple projects can utilize a single provisioned kit. Individual project folders can co-exist by copying the Security Directory of a project aligned with the provisioned kit to each project folder. The instructions here are for a first time use case assuming one project for one provisioned kit, as such, the kit is re-provisioned and security folder updated for each new project. Once re-provisioned with a new set of keys or policy, all previous Security Folder keys and Certificates are no longer valid.

Note: Amazon-FreeRTOS contains a demo project and security directory that will be used in this demonstration. Both the project and security folder are modified in this demonstration. To repeat this demo with an unmodified project, a copy of amazon-freertos can be copied into each new project folder.

Alternatively each project can link to a single copy of amazon-freertos. If a single copy of amazon-freertos is being used and this demo is performed on multiple kits, any creation of new keys for kit provisioning created in or copied to the security directory of amazon-freertos for the AWS_Demo, will disallow the ability to build projects for any previously provisioned kits.

I.E. If the project folder is shared, the security key folder for that project, is also shared.

Note: AWS Security Keys and Certificates will be used in the Project to allow communication between your project and AWS. AWS keys are NOT related to the Security Keys and Certificates for kit Provisioning, which set policies for your physical kit and ensure only projects built with the board security can be loaded onto the provisioned board.

OVERVIEW

Copy or Create Security Keys for Provisioning the PSoC 64

Provision or Re-Provision the PSoC 64

KNOWLEDGE AND ARTIFACTS (IF YOU HAVE ALL THESE, SKIP THIS CHAPTER)

A security Folder in your project that contains a unique set of security keys that were used to Provision your Kit to its current provisioned state.

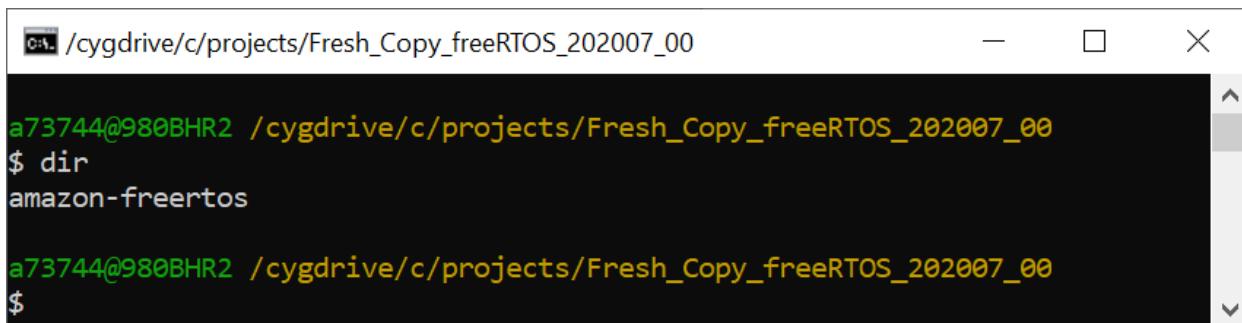
A kit that is provisioned with the unique set of security keys in the project folder being used

SETUP THE PROVISIONING KEYS

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 of the FreeRTOS that will be used for this AWS_Demo project. Type dir to ensure you see “amazon-freertos” in the list of files and directories.



```
a73744@980BHR2 /cygdrive/c/projects/Fresh_Copy_freeRTOS_202007_00
$ dir
amazon-freertos

a73744@980BHR2 /cygdrive/c/projects/Fresh_Copy_freeRTOS_202007_00
$
```

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

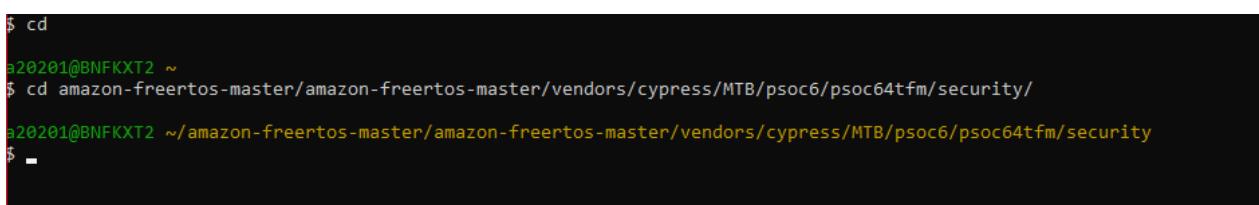
Tip: To move up a directory, type “cd ..”. To move down into a directory, type “cd <directory_name>”.

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. I.E. Copy the following command into a Modus Shell that is set to your project folder.

cd amazon-freertos/vendors/cypress/MTB/psoc6/psoc64tfm/security



```
$ cd
a20201@BNFKXT2 ~
$ cd amazon-freertos-master/amazon-freertos-master/vendors/cypress/MTB/psoc6/psoc64tfm/security/
a20201@BNFKXT2 ~/amazon-freertos-master/amazon-freertos-master/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. If prompted, 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 as in the below screenshot

```
$ cysecuretools --target CY8CKIT-064S0S2-4343W init
2021-01-22 20:43:54,615 : C : INFO : Copy 'C:\Users\aa20201\amazon-freertos-master\amazon-freertos-master\vendors\cyress\MTB\psoc6\psoc64tfm\security\packets\cy_auth_2m_s0_sample.jwt'
2021-01-22 20:43:54,623 : C : INFO : Copy 'C:\Users\aa20201\amazon-freertos-master\amazon-freertos-master\vendors\cyress\MTB\psoc6\psoc64tfm\security\packets\control_dap_cert.json'
2021-01-22 20:43:54,631 : C : INFO : Copy 'C:\Users\aa20201\amazon-freertos-master\amazon-freertos-master\vendors\cyress\MTB\psoc6\psoc64tfm\security\keys\hsm_state.json'
2021-01-22 20:43:54,637 : C : INFO : Copy 'C:\Users\aa20201\amazon-freertos-master\amazon-freertos-master\vendors\cyress\MTB\psoc6\psoc64tfm\security\keys\oem_state.json'
2021-01-22 20:43:54,639 : C : INFO : Copy 'C:\Users\aa20201\amazon-freertos-master\amazon-freertos-master\vendors\cyress\MTB\psoc6\psoc64tfm\security\policy\policy_multi_CM0_CM4.json'
2021-01-22 20:43:54,646 : C : INFO : Copy 'C:\Users\aa20201\amazon-freertos-master\amazon-freertos-master\vendors\cyress\MTB\psoc6\psoc64tfm\security\policy\policy_multi_CM0_CM4_smif.json'
2021-01-22 20:43:54,654 : C : INFO : Copy 'C:\Users\aa20201\amazon-freertos-master\amazon-freertos-master\vendors\cyress\MTB\psoc6\psoc64tfm\security\policy\policy_single_CM0_CM4.json'
2021-01-22 20:43:54,661 : C : INFO : Copy 'C:\Users\aa20201\amazon-freertos-master\amazon-freertos-master\vendors\cyress\MTB\psoc6\psoc64tfm\security\policy\policy_single_CM0_CM4_smif.json'
2021-01-22 20:43:54,676 : C : INFO : Copy 'C:\Users\aa20201\amazon-freertos-master\amazon-freertos-master\vendors\cyress\MTB\psoc6\psoc64tfm\security\prebuilt\CyBootloader_Release\CypressBootloader_CM0p.hex'
2021-01-22 20:43:54,683 : C : INFO : Copy 'C:\Users\aa20201\amazon-freertos-master\amazon-freertos-master\vendors\cyress\MTB\psoc6\psoc64tfm\security\prebuilt\CyBootloader_Release\CypressBootloader_CM0p.jwt'
2021-01-22 20:43:54,692 : C : INFO : Copy 'C:\Users\aa20201\amazon-freertos-master\amazon-freertos-master\vendors\cyress\MTB\psoc6\psoc64tfm\security\prebuilt\CyBootloader_WithLogs\CypressBootloader_CM0p.hex'
2021-01-22 20:43:54,699 : C : INFO : Copy 'C:\Users\aa20201\amazon-freertos-master\amazon-freertos-master\vendors\cyress\MTB\psoc6\psoc64tfm\security\prebuilt\CyBootloader_WithLogs\CypressBootloader_CM0p.jwt'

a20201@BNFKX2 ~ /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. Ensure you are in <freertos>\vendors\cypress\MTB\psoc6\psoc64tfm\security

6.6. After running the following command, you may be asked to overwrite files. Overwrite those files.

```
cysecurertools --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
$ cysecurertools --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\vendors\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\vendors\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: Using the 3.x version of cysecurertools for the Rev*A kit, the above command creates a warning:

WARN : There is gap between regions 269926400:270254080 and 270303232:270336000 (49152 bytes)

```
a73744@980BHR2 /cygdrive/c/projects/p64_1007/amazon-freertos/vendors/cypress/MTB/psoc6/psoc64tfm/security
$ cysecurertools --policy ./policy/policy_multi_CM0_CM4_tfm.json --target CY8CKIT-064S0S2-4343W create-keys
2021-10-07 16:52:24,292 : C : WARN : There is gap between regions 269926400:270254080 and 270303232:270336000 (49152 bytes)
2021-10-07 16:52:24,292 : C : WARN : Policy validation finished with warnings
2021-10-07 16:52:24,392 : C : INFO : Created key in C:\projects\p64_1007\amazon-freertos\vendors\cypress\MTB\psoc6\psoc64tfm
2021-10-07 16:52:24,392 : C : INFO : Created key in C:\projects\p64_1007\amazon-freertos\vendors\cypress\MTB\psoc6\psoc64tfm
```

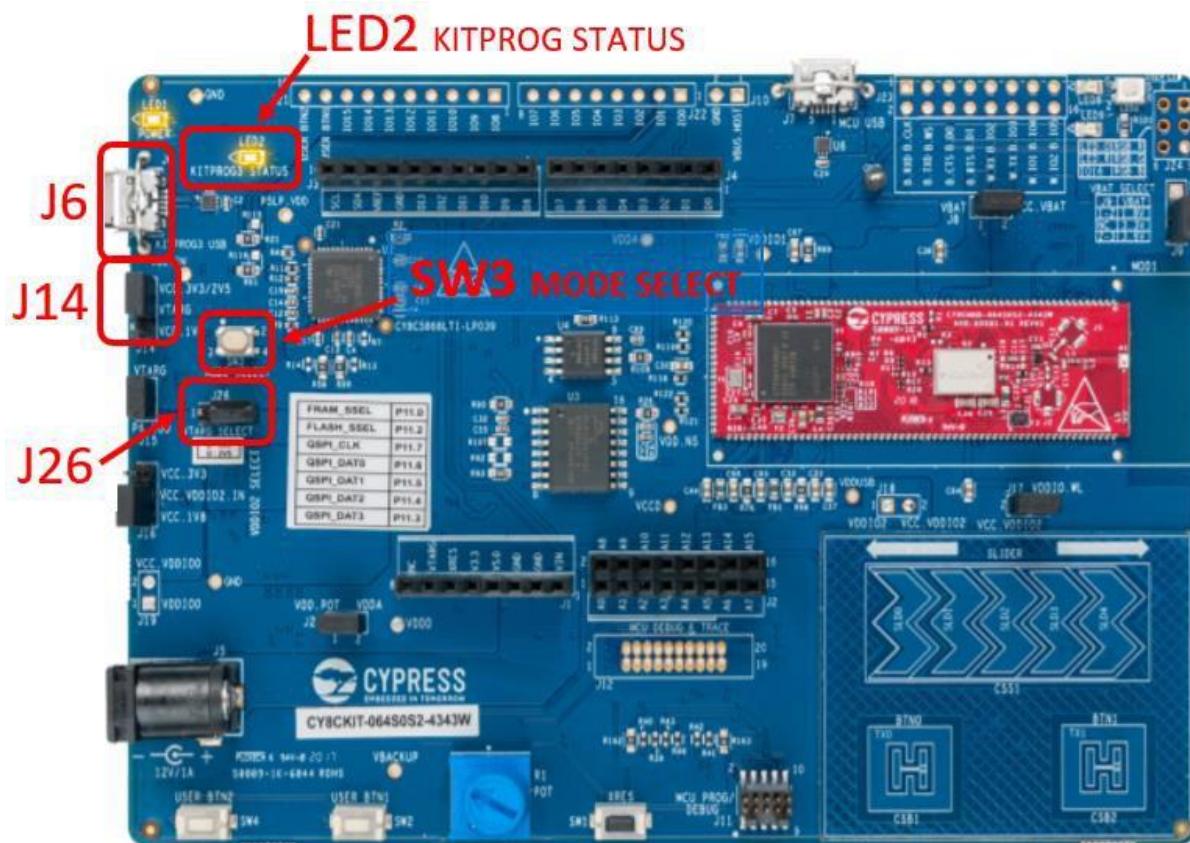
PREP THE KIT FOR PROVISIONING

Connect the kit to your computer in the proper configuration for provisioning.

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, before power is applied, 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



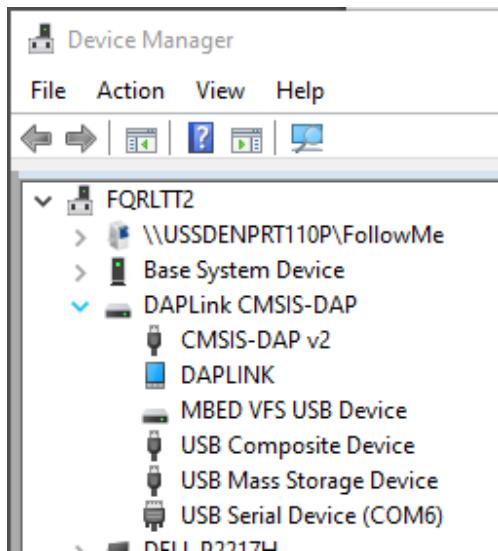
- 6.8.** Connect your kit to the computer using the provided USB cable through the KitProg3 USB connector [J6]

- 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.

Note: 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

Confirm KitProg3 is set to DAPLink mode by verifying the kit is registered as a Disk Drive in Windows

- 6.10.** In Windows, open Device Manager, then under the **View** tab select **Devices by Container**. Identify a drive named “DAPLink CMSIS-DAP”



PROVISION OR RE-PROVISION THE KIT

- 6.11.** In Modus Shell, ensure you are still in the following directory:

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

then run one of the below commands:

- To provision a new kit

```
cysecurertools --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

```
cysecurertools --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\cys06xx\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...

$20201@BNFKXT2 ~/amazon-freertos-master/amazon-freertos-master/vendors/cypress/MTB/psoc6/psoc6tfm/security
$
```

The above image was from provisioning a Rev** kit. An output from provisioning a Rev*A kit is in the appendix.

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

6.13. 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

Import AWSDemo Project into ModusToolbox.

OVERVIEW

Setup AWSDemo

KNOWLEDGE AND ARTIFACTS (IF YOU HAVE ALL THESE, SKIP THIS CHAPTER)

AWSDemo setup as a project in ModusToolbox

SETUP THE PROVISIONING KEYS

Import AWSDemo Project into ModusToolbox.

7.1. Open the ModusToolbox Eclipse IDE and choose or create a new workspace using the full path.

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

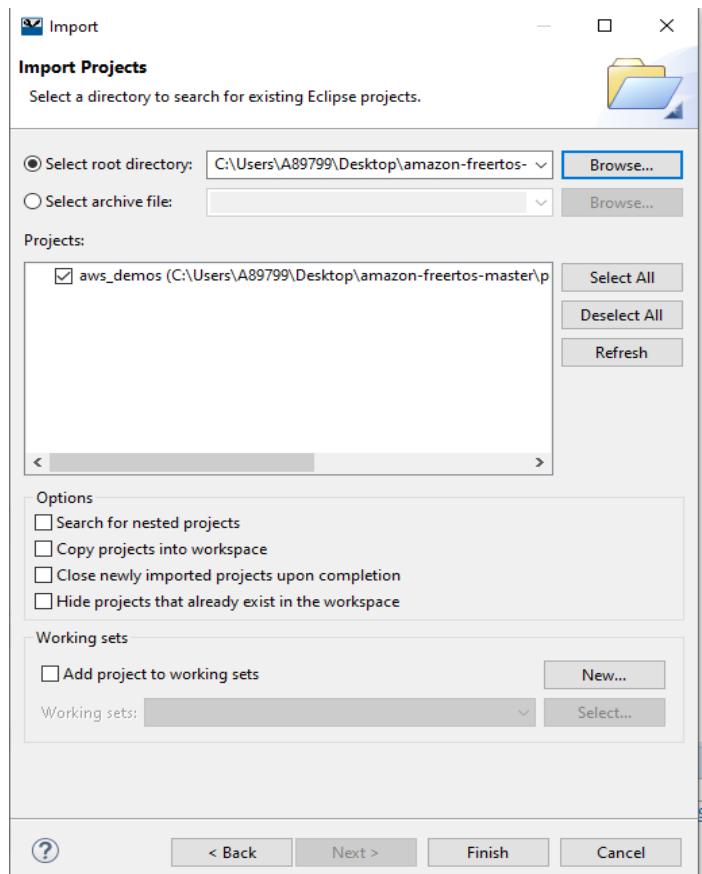
7.3. Expand the **General** tab and choose **Existing Projects into Workspace** then click **Next**

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

Provide specific information for your ModusToolbox project to access your AWS Thing through a WiFi connection.

OVERVIEW

Enable a WiFi connection: Customize the project with login information (SSID and PWD) for a WiFi Access Point

Tell the project what to communicate with: Provide the address of your AWS Thing to the project

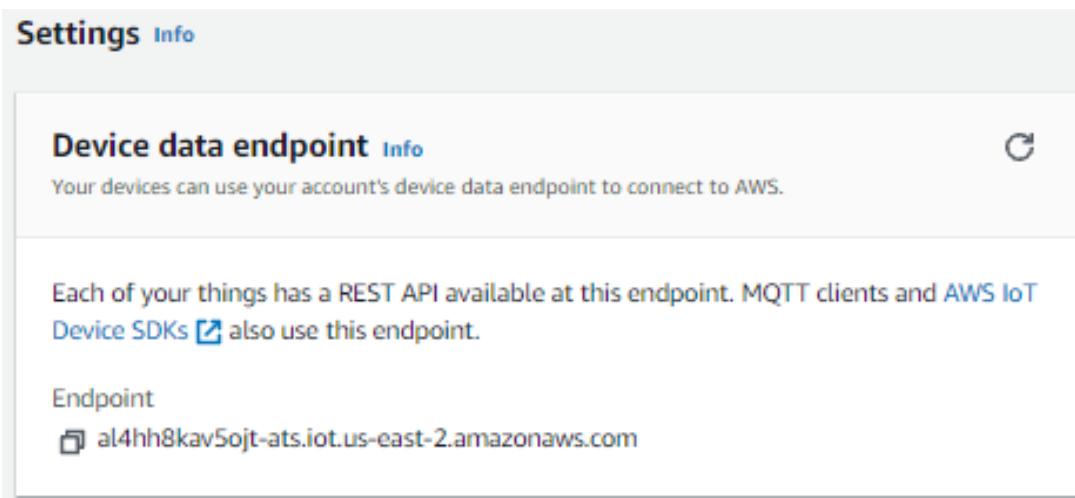
KNOWLEDGE AND ARTIFACTS (IF YOU HAVE ALL THESE, SKIP THIS CHAPTER)

A header file, aws_clientcredential.h, in your ModusToolbox AWS_Demo project that contains your WiFi SSID and Password as well as address to your AWS Thing

DETERMINE THE AWS ENDPOINT IDENTIFIER FOR YOUR THING

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:

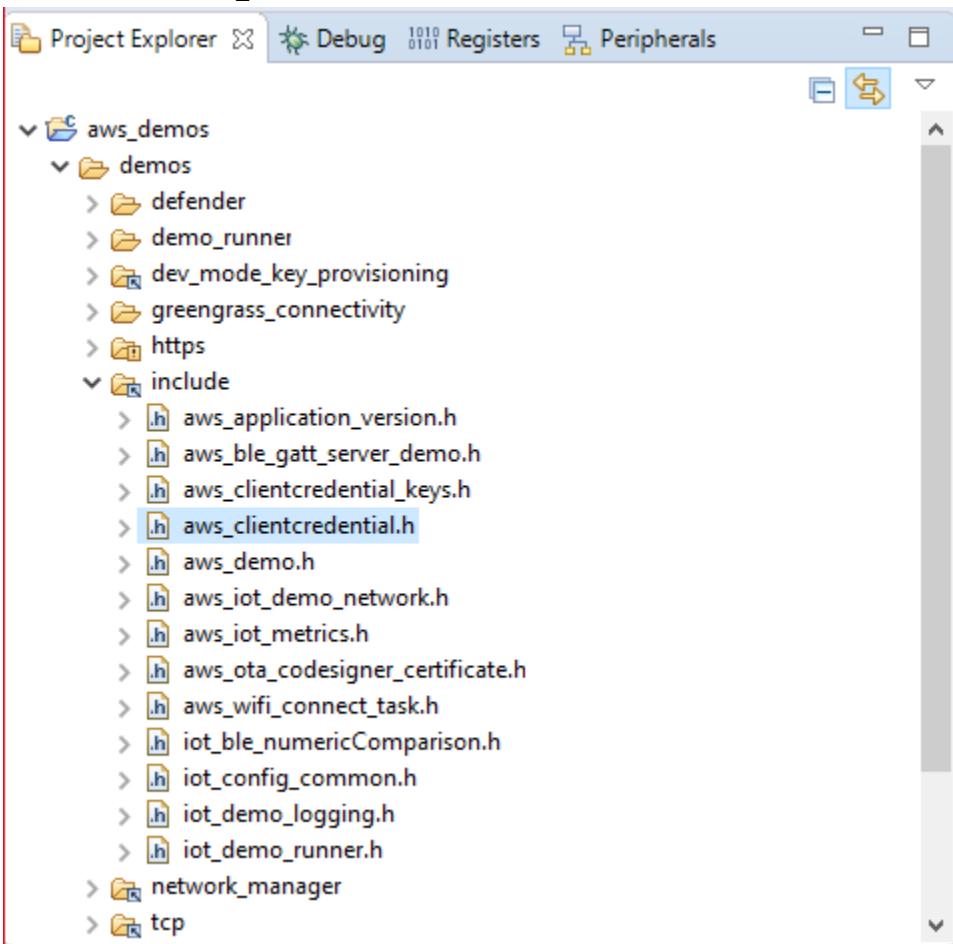


8.3. Expand **Manage** in the navigation pane and choose **Things**

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

OPEN THE PROJECT DEFINES FOR ENDPOINT ID AND WIFI LOGIN CREDENTIALS

- 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 "YourAWSIoTthingname"
```

```
#define clientcredentialWIFI_SSID "TheSSIDforyourWi-Finetwork"
```

```
#define clientcredentialWIFI_PASSWORD "ThepasswordforyourWi-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

```

#ifndef __AWS_CLIENTCREDENTIAL_H__
#define __AWS_CLIENTCREDENTIAL_H__

/* @brief MQTT Broker endpoint.
 */
/* @todo Set this to the fully-qualified DNS name of your MQTT broker.
 */
#define clientcredentialMQTT_BROKER_ENDPOINT      "a3m3zunaszckgk-ats.iot.us-east-2.amazonaws.com"

/* @brief Host name.
 */
/* @todo Set this to the unique name of your IoT Thing.
 */
#define clientcredentialIOT_THING_NAME           "PSoC_JQ"

/* @brief Port number the MQTT broker is using.
 */
#define clientcredentialMQTT_BROKER_PORT         8883

/* @brief Port number the Green Grass Discovery use for JSON retrieval from cloud is using.
 */
#define clientcredentialGREENGRASS_DISCOVERY_PORT 8443

/* @brief Wi-Fi network to join.
 */
/* @todo If you are using Wi-Fi, set this to your network name.
 */
#define clientcredentialWIFI_SSID                 "Jacob's iPhone"

/* @brief Password needed to join Wi-Fi network.
 */
/* @todo If you are using WPA, set this to your network password.
 */
#define clientcredentialWIFI_PASSWORD            "12345678"

/* @brief Wi-Fi network security type.
 */
/* @see WIFISecurity_t.
 */
/* @note Possible values are eWiFiSecurityOpen, eWiFiSecurityWEP, eWiFiSecurityWPA,
 * eWiFiSecurityWPA2 (depending on the support of your device Wi-Fi radio).
 */
#define clientcredentialWIFI_SECURITY            eWiFiSecurityWPA2

```

9. FORMATTING THE AWS IOT CREDENTIALS

Load the Private Key and Certificate associated with your AWS Thing into your ModusToolbox Project.

The AWS_Demo project needs the AWS IoT certificate and private key that is linked with your registered thing and its permissions policies to successfully communicate with your Thing in AWS IoT.

OVERVIEW

Load your IAM Thing's AWS Private Key and Certificate into the AWS Demo project

KNOWLEDGE AND ARTIFACTS (IF YOU HAVE ALL THESE, SKIP THIS CHAPTER)

A header file, aws_clientcredential_keys.h, in your ModusToolbox AWS_Demo project that contains your AWS IAM Thing's Private Key and Credentials

UPDATE DEFINES IN PROJECT FOR THE AWS LINKAGE

AWS_Demo is a FreeRTOS based C language project. Certificate and private key must be specially formatted then 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
```

The screenshot shows the Eclipse IDE interface for the ModusToolbox. On the left, the Project Explorer displays the 'aws_demos' project structure. The 'aws_clientcredential_keys.h' file is currently selected and open in the main editor area. This file contains the following code:

```

#ifndef AWS_CLIENT_CREDENTIAL_KEY_H
#define AWS_CLIENT_CREDENTIAL_KEY_H

#include <stdint.h>

/* PEM-encoded client certificate.
 *
 * Must include the PEM header and footer:
 * -----BEGIN CERTIFICATE-----\n"
 * ...base64 data...\n"
 * -----END CERTIFICATE-----\n
 */

#define keyCLIENT_CERTIFICATE_PEM \
"-----BEGIN CERTIFICATE-----\n" \
"MIIDWjCCAKKgAwIBAgIVAOoL0OTxRK64SILvkMjU2UXYEtqQMA0GCSqGSIb3DQEByw\n" \
"CwUAME0xSzBjBgNVBASMKqFtyXpvbiBXWlgiZvdm1jZXNlgTz1BbWF6b24uV29t\n" \
"IEluYy4gTD1TzWf0dGx1IFNUPVdhc2hpbd0b24gQz1VUaEfv0yMDA3MjcxOTA0\n" \
"MzFaFw00OTEyMzEyMzU5NTlaMB4xHDAaBgNVBAMME0FXUyBjb1QgQ2VydGlmawNh\n" \
"dGUwggiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQc+pm7vcr/XNKpkHNHKt\n" \
"EHRaBpDG0UDVymZ+bRdkKQYC4dAMNS5fa8cdDBXVH7IijfsYJjhxaTYnSPrMTx/Q\n" \
"x66Ulwtfx4EsvNQln0ACpUmQig1+vAJQ4gxMZETLaw6k3xPMgu+DQqtRnUOHMLCv\n" \
"vkstiv/pz85s6VTBweaIoJeb4xEGUhnVACKL&/AGC3AWPgH5mDw6QJPw8z5qQW\n" \
"sKxsPXmbTjeYRm48y+P1PxAz7WUJApCq2SdbtQQqZKErFFikZwVzyTQ6jDkJ+f\n" \
"OyBNV1JYhG1SKwehCSk8GEexcn013ZPyr8M2/+xEpa9VvtX1BiXBBdGDYgVuptFPU\n" \
"MiCAgMBAAgjYDBeMB8GA1UdIwQYMbaAFDZfYf5qHv280mba81Gf1VoKs6m0MB0G\n" \
"A1UdDgQWBRIWOUqiCP/244owu9bCZhGnQz6BTAMBgNVHRMBAf8EAjAAMA4GA1Ud\n" \
"DwEB/wQEAvIHgDANBgkqhkiGw0BAQsFAAOCAQEAc5FEp1cPw5xhC6fZf7fH1YNW\n" \
"nF2TvcvTiybSQZP08QvOMB0uabDnL2GA0+Yy73eRCj5tBYxfXHVJU7D7o7z/2r1H\n" \
"m4z3T/PFS1Q4HQe5kCQ6n3n1H/luVNlkGF6H+N2MURBFV3eGrE0w1bPICbcMr\n" \
"aHA+7xvi8mt7+VQ1c3a+JieNoPhv7PrduBsnnQK50z+KepFEEJdEsTgIRhiImcQ\n" \
"Sya+FPwbv1VS9cRJ888ZhvHnruatBYYwlgQhFsyoYhCrGZGby2kRS1q092wV5A7D\n" \
"nx5T2hqv4N55qXD7UHcg0U9NEBjY9ET/TGoA0cavEiAAL5M9RvDsJcqBjhgb=\n" \
"-----END CERTIFICATE-----\n

/* PEM-encoded client private key.
 *
 * Must include the PEM header and footer:
 * -----BEGIN RSA PRIVATE KEY-----\n"
 * ...base64 data...\n"
 * -----END RSA PRIVATE KEY-----\n
 */

#define keyCLIENT_PRIVATE_KEY_PEM \
"-----BEGIN RSA PRIVATE KEY-----\n" \
"MIIEogIBAAKCAQEAvqZu73K/1zSqZDRYrRB0WgaQxtFA1cpfm0Q5CkGAuHQDFku\n" \
"X2vHHQwV1R+yIo37GCY4cIk2J0j6zE8f0MeuljC LX+MRLlzuJZ9AAqVJkIoNfrwC\n" \
"UOIMTGREy2sOpNBtzILvg0KrUz1DhzCwr75LLYiac/ErOlbeQchmiKCXm+MRBlj5\n" \
"1QAIi/HvwBgtwfj4B+Zg80kCT1M+aqEFrcsbD15gbY3mEZuPMvj5T1wM+1lCQKQ\n" \
"qtknw7UEKmaiKxRYp6cFc8k1kOow5CfnzsgTVZSWIRpUisHoQkpPBhMXJ9Jd2T8\n" \
"q/DNv/sRD2vWFbV9QYlwQRg2IFbqbRT1DMA1wIDAQABoIBAHtNhR4UZwK2qyUX\n" \
"zrsjkzzOVkBgAcDd0LPpq8cEyUB/72CtJRLNz/93W391em5iyWz6+zJOS5no1vg\n" \
"KMGNAspICL4oW+sq6KaonLgOKchtHaotzQ9gdKKjwtx1DWfiWc4W9RZR/pyEzd\n" \
"-----END RSA PRIVATE KEY-----\n"

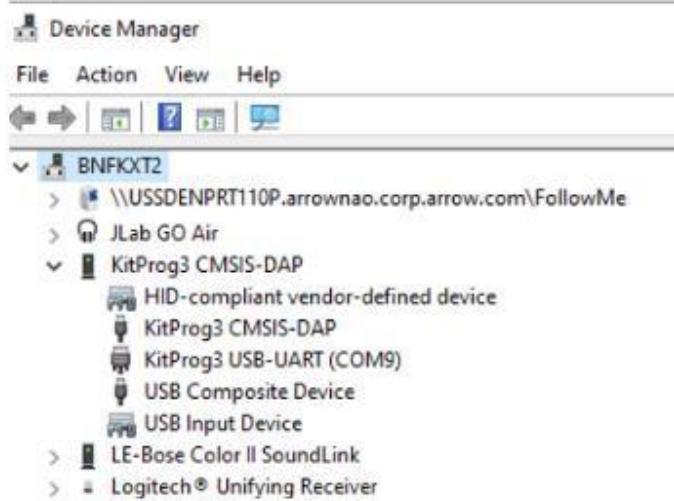
```

The bottom of the screen shows the Eclipse interface with tabs for Console, Problems, Memory, and News.

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

9.9. Identify the KitProg3 USB-UART port number

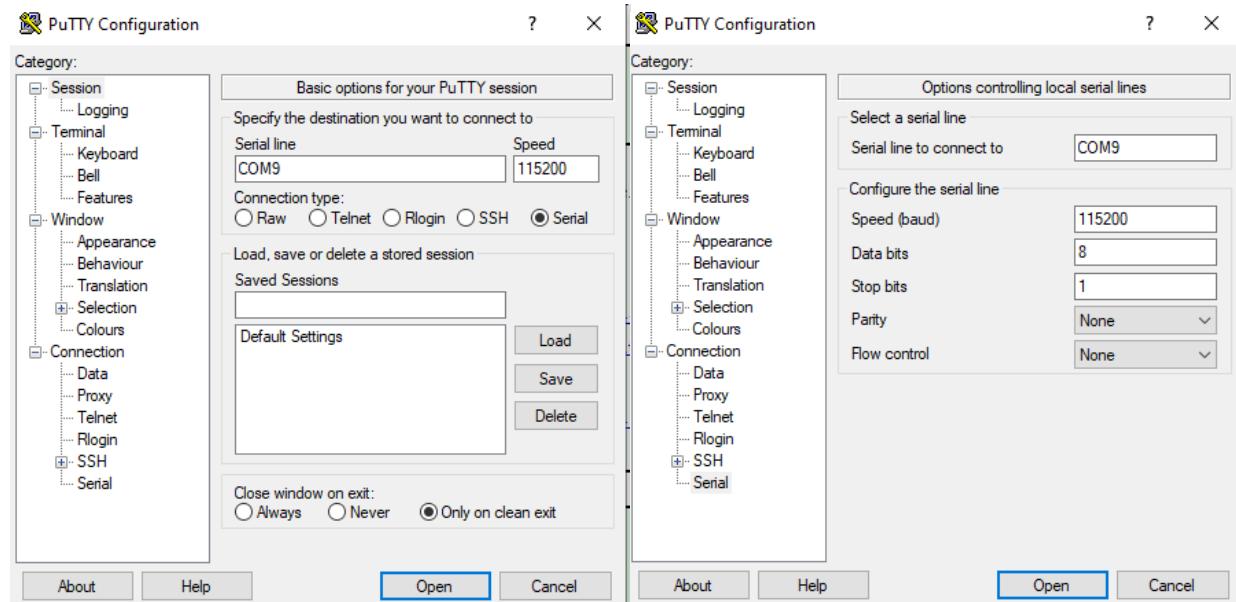
NOTE: IN WINDOWS, OPEN DEVICE MANAGER AND EXPAND PORTS (COM & LPT) -- OR -- VIEW DEVICES BY CONTAINER THEN EXPAND KITPROG3 CMSIS-DAP TO IDENTIFY THE PORT NUMBER



9.10. 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

Build and load the customized AWS Demo project onto your kit

OVERVIEW

Build the AWS Demo project by compiling it using the compiler associated with your instance of ModusToolbox

The Build process compiles the project and embeds a security certificate based on the keys and policies in the Security folder, associated with a kit that was provisioned with those same keys and policies.

Program the secure AWS Demo project onto your provisioned kit

KNOWLEDGE AND ARTIFACTS (IF YOU HAVE ALL THESE, SKIP THIS CHAPTER)

The AWS Demo program loaded onto your kit runs through one cycle of instructions that it mirrors on a terminal window.

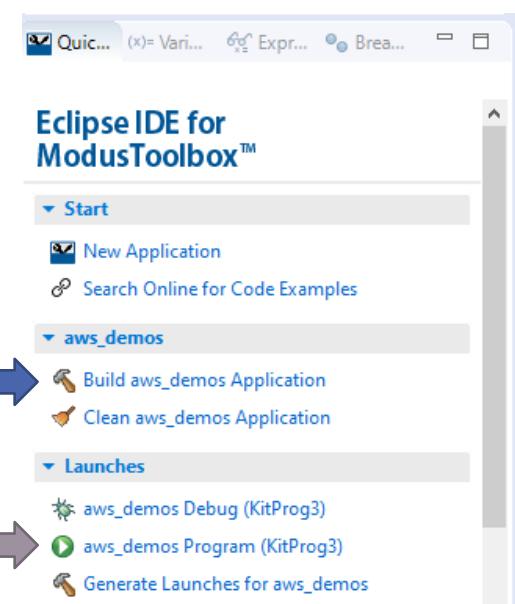
The AWS Demo program runs from the start after the black XRES reset button on the kit is pressed then released.

BUILD THE PROJECT

- 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)**.

Program (KitProg3) will program the CY8CKIT_064SOS2_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

VIEW EXECUTION STEPS OF THE PROJECT

- 10.3.** View the status of the running application in the serial terminal and notice the following initialization steps on the target device:
- The device connects to the configured Wi-Fi access point (AP) and acquires an IP address
 - Once an internet connection is made the device is provisioned to the AWS cloud with the configured endpoint address, certificate, and private key
 - 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

```

subregion 4 enabled      subregion 5 enabled      subregion 6 enabled

f800, size = 0x800 bytes, all subregions enabled      Starting Cortex-M4 at 0x10050400
Non-secure code running on non-secure core.

Cores sync success.
WLAN MAC Address : D4:53:83:0E:EE:A8
WLAN Firmware   : w10: Sep 5 2019 23:24:33 version 7.45.98.92 (r722362 CY) FWID 01-f7128517
WLAN CLM        : API: 12.2 Data: 9.10.39 Compiler: 1.29.4 ClmImport: 1.36.3 Creation: 2019-09-05 23:10:00
WHD VERSION    : v1.70.0 : v1.70.0 : GCC 7.2 : 2019-12-02 04:14:53 -0600
0 552 [Tmr_Svc] Wi-Fi module initialized. Connecting to AP...
1 8271 [Tmr_Svc] Wi-Fi Connected to AP. Creating tasks which use network...
2 8274 [Tmr_Svc] IP Address acquired 172.20.10.5
3 8273 [Tmr_Svc] Write certificate...
4 9679 [iot_threal] [INFO] [DEMO][9679] -----STARTING DEMO-----
5 9679 [iot_threal] [INFO] [INIT][9679] SDK successfully initialized.
.. 6 18167 [iot_threal] [INFO] [DEMO][18167] Successfully initialized the demo. Network type for the dev 18167 [iot_thre
8 18167 [iot_threal] [INFO] [DEMO][18167] MQTT demo client identifier is PSoC_JQ (length ?).
.. 9 34281 [iot_threal] [INFO] [MQTT][34281] Establishing new MQTT connection.
10 34282 [iot_threal] [INFO] [MQTT][34282] Anonymous metrics (SDK language, SDK version) will be prod1 34283 [iot_thre
41 34283 [iot_threal] [INFO] [MQTT][34283] CONNECT operation 0x8040c18> 13 34505 [iot_threal] [INFO] [MQTT][34505] New MQTT connec
14 34506 [iot_threal] [INFO] [MQTT][34506] <MQTT connection 0x8040c18> SUBSCRIBE operation scheduled15 34506 [iot_thre
21 34507 [iot_threal] [INFO] [MQTT][34507] <MQTT connection 0x8040c18> SUBSCRIBE operation scheduled16 34507 [iot_thre
18 34723 [iot_threal] [INFO] [DEMO][34723] Publishing messages 0 to 1.
19 34724 [iot_threal] [INFO] [MQTT][34724] <MQTT connection 0x8040c18> MQTT PUBLISH operation queued20 34725 [iot_thre
51 Waiting for 2 publishes to be received.
22 34871 [iot_threal] [INFO] [DEMO][34871] MQTT PUBLISH 0 successfully sent.
23 34895 [iot_threal] [INFO] [DEMO][34895] Incoming PUBLISH received:
Subscription topic filter: io24 34895 [iot_threal] [INFO] [MQTT][34895] <MQTT connection 0x8040c18> MQTT PUBLISH opera
26 34926 [iot_threal] [INFO] [DEMO][34926] MQTT PUBLISH 1 successfully sent.
27 35000 [iot_threal] [INFO] [DEMO][35000] Incoming PUBLISH received:
Subscription topic filter: io28 35000 [iot_threal] [INFO] [MQTT][35000] <MQTT connection 0x8040c18> MQTT PUBLISH opera
30 35002 [iot_threal] [INFO] [DEMO][35002] 2 publishes received.
31 35002 [iot_threal] [INFO] [DEMO][35002] Publishing messages 2 to 3.
32 35002 [iot_threal] [INFO] [MQTT][35002] <MQTT connection 0x8040c18> MQTT PUBLISH operation queued33 35005 [iot_thre
51 Waiting for 2 publishes to be received.
.. 35 35332 [iot_threal] [INFO] [DEMO][35332] MQTT PUBLISH 2 successfully sent.
36 35332 [iot_threal] [INFO] [DEMO][35332] MQTT PUBLISH 3 successfully sent.
37 35358 [iot_threal] [INFO] [DEMO][35358] Incoming PUBLISH received:
Subscription topic filter: io38 35358 [iot_threal] [INFO] [MQTT][35358] <MQTT connection 0x8040c18> MQTT PUBLISH opera
49 35366 [iot_threal] [INFO] [DEMO][35366] Incoming PUBLISH received:
Subscription topic filter: io41 35366 [iot_threal] [INFO] [MQTT][35366] <MQTT connection 0x8040c18> MQTT PUBLISH opera
43 35367 [iot_threal] [INFO] [DEMO][35367] 2 publishes received.
44 35367 [iot_threal] [INFO] [DEMO][35367] Publishing messages 4 to 5.
45 35369 [iot_threal] [INFO] [MQTT][35369] <MQTT connection 0x8040c18> MQTT PUBLISH operation queued46 35370 [iot_thre
01 Waiting for 2 publishes to be received.
48 35788 [iot_threal] [INFO] [DEMO][35788] MQTT PUBLISH 4 successfully sent.
49 35788 [iot_threal] [INFO] [DEMO][35788] MQTT PUBLISH 5 successfully sent.
50 35811 [iot_threal] [INFO] [DEMO][35811] Incoming PUBLISH received:
Subscription topic filter: io51 35811 [iot_threal] [INFO] [MQTT][35811] <MQTT connection 0x8040c18> MQTT PUBLISH opera
53 35813 [iot_threal] [INFO] [DEMO][35813] Incoming PUBLISH received:
Subscription topic filter: io54 35814 [iot_threal] [INFO] [MQTT][35813] <MQTT connection 0x8040c18> MQTT PUBLISH opera
56 35815 [iot_threal] [INFO] [DEMO][35815] 2 publishes received.
57 35815 [iot_threal] [INFO] [DEMO][35815] Publishing messages 6 to 7.
58 35815 [iot_threal] [INFO] [MQTT][35815] <MQTT connection 0x8040c18> MQTT PUBLISH operation queued59 35818 [iot_thre
81 Waiting for 2 publishes to be received.
61 36137 [iot_threal] [INFO] [DEMO][36137] MQTT PUBLISH 7 successfully sent.
62 36137 [iot_threal] [INFO] [DEMO][36137] MQTT PUBLISH 6 successfully sent.
.. 63 36146 [iot_threal] [INFO] [DEMO][36146] Incoming PUBLISH received.

```

- 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

Setup a window within your AWS IAM account to display the messages being sent from your kit.

The AWS_Demo project sends data encapsulated in MQTT messages to the AWS cloud destined for the address of the Thing created in your AWS IAM account. From AWS IAM account, you can accept to see (“subscribe”) and display those messages that are being sent from your kit (“publish”).

OVERVIEW

Subscribe to view information being sent to AWS from your kit

KNOWLEDGE AND ARTIFACTS (IF YOU HAVE ALL THESE, SKIP THIS CHAPTER)

The viewing pane on AWS updates with X messages published by the AWS Demo program running on your kit.

SUBSCRIBE TO INCOMING MESSAGES

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

Note: It may take several minutes for the demo to get to the point of publishing data.

The following output shows one instance that paused at step 10 for a couple minutes. After each step number is a timestamp. Step 10 in the output below stalls for some 120000 ticks (that appears to be 2 minutes), then, after a couple quick fails in steps 11-14, it re-tries, keeps going and successfully publishes information successfully.

```
1 698 [Tmr Svc] Wi-Fi module initialized. Connecting to AP Maddie...
.2 8800 [IOT-Wifi-] Notify application that IP is changed!
3 8811 [Tmr Svc] WiFi connected to AP Maddie.
```

```
4 8811 [Tmr Svc] IP Address acquired 192.168.0.34.  
.5 10172 [Tmr Svc] Write certificate...  
6 11466 [iot_threa] [INFO ][DEMO][lu] -----STARTING DEMO-----  
7 11468 [iot_threa] [INFO ][INIT][lu] SDK successfully initialized.  
.8 20200 [IOT-Wifi-] Notify application that IP is changed!  
9 20215 [iot_threa] [INFO ][DEMO][lu] Successfully initialized the demo. Network type for the  
demo: 1  
10 20215 [iot_threa] [INFO] Creating a TLS connection to al4hh8kav5ojs-ats.iot.us-east-  
2.amazonaws.com:8883.  
.....11 145677 [iot_threa] ERROR: TLS handshake failed trying to connect. SSL  
- The peer notified us that the connection is going to be closed : <No-Low-Level-Code>  
12 145677 [iot_threa] TLS_Connect fail (0x7880, al4hh8kav5ojs-ats.iot.us-east-2.amazonaws.com)  
13 145677 [iot_threa] [ERROR] Failed to establish new connection. secureSocketStatus=-1.  
14 145678 [iot_threa] [WARN] Connection to the broker failed. Attempting connection retry after  
backoff delay.  
15 145865 [iot_threa] [INFO] Retry attempt 2 out of maximum retry attempts 5.  
16 145865 [iot_threa] [INFO] Creating a TLS connection to al4hh8kav5ojs-ats.iot.us-east-  
2.amazonaws.com:8883.  
....17 173176 [iot_threa] [INFO] Creating an MQTT connection to al4hh8kav5ojs-ats.iot.us-east-  
2.amazonaws.com.  
18 173283 [iot_threa] [INFO] Packet received. ReceivedBytes=2.  
19 173283 [iot_threa] [INFO] CONNACK session present bit not set.  
20 173283 [iot_threa] [INFO] Connection accepted.  
21 173283 [iot_threa] [INFO] Received MQTT CONNACK successfully from broker.  
22 173283 [iot_threa] [INFO] MQTT connection established with the broker.  
23 173283 [iot_threa] [INFO] An MQTT connection is established with al4hh8kav5ojs-ats.iot.us-  
east-2.amazonaws.com.  
24 173283 [iot_threa] [INFO] Attempt to subscribe to the MQTT topic P64_AWS/example/topic.  
25 173286 [iot_threa] [INFO] SUBSCRIBE sent for topic P64_AWS/example/topic to broker.  
26 173440 [iot_threa] [INFO] Packet received. ReceivedBytes=3.  
27 173440 [iot_threa] [INFO] Subscribed to the topic P64_AWS/example/topic with maximum QoS 1.  
28 174440 [iot_threa] [INFO] Publish to the MQTT topic P64_AWS/example/topic.  
...
```

The screenshot shows the MQTT client interface with the following details:

- Subscriptions:** iotdemo/#
- Publish:** iotdemo/#
- Message Content:**

```

1  [
2   "message": "Hello from AWS IoT console"
3 ]

```
- Published Messages:**
 - iotdemo/acknowledgements - August 04, 2020, 12:39:28 (UTC-0600)
 - Client has received PUBLISH 1 from server.
 - iotdemo/acknowledgements - August 04, 2020, 12:39:28 (UTC-0600)
 - Client has received PUBLISH 0 from server.
 - iotdemo/topic/2 - August 04, 2020, 12:39:28 (UTC-0600)
 - Hello world 1!
 - iotdemo/topic/1 - August 04, 2020, 12:39:28 (UTC-0600)
 - Hello world 0!

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_064S052_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.

Note: The section “FreeRTOS demos” is on the 241st of 442 pages (marked page 233) in version 202012.00 of the FreeRTOS User Guide.

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

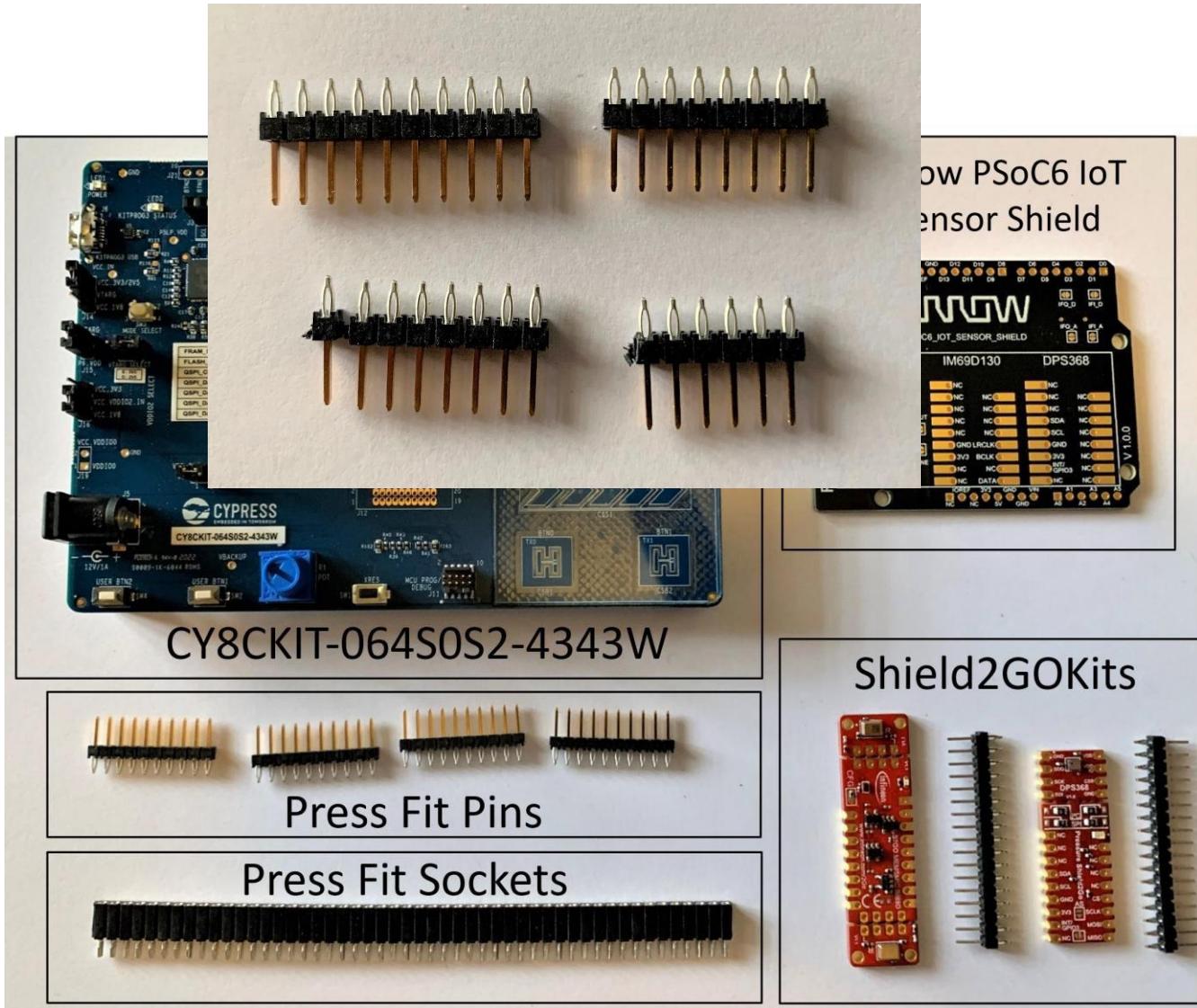
Alternatively run as the user who installed ModusToolbox.

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:

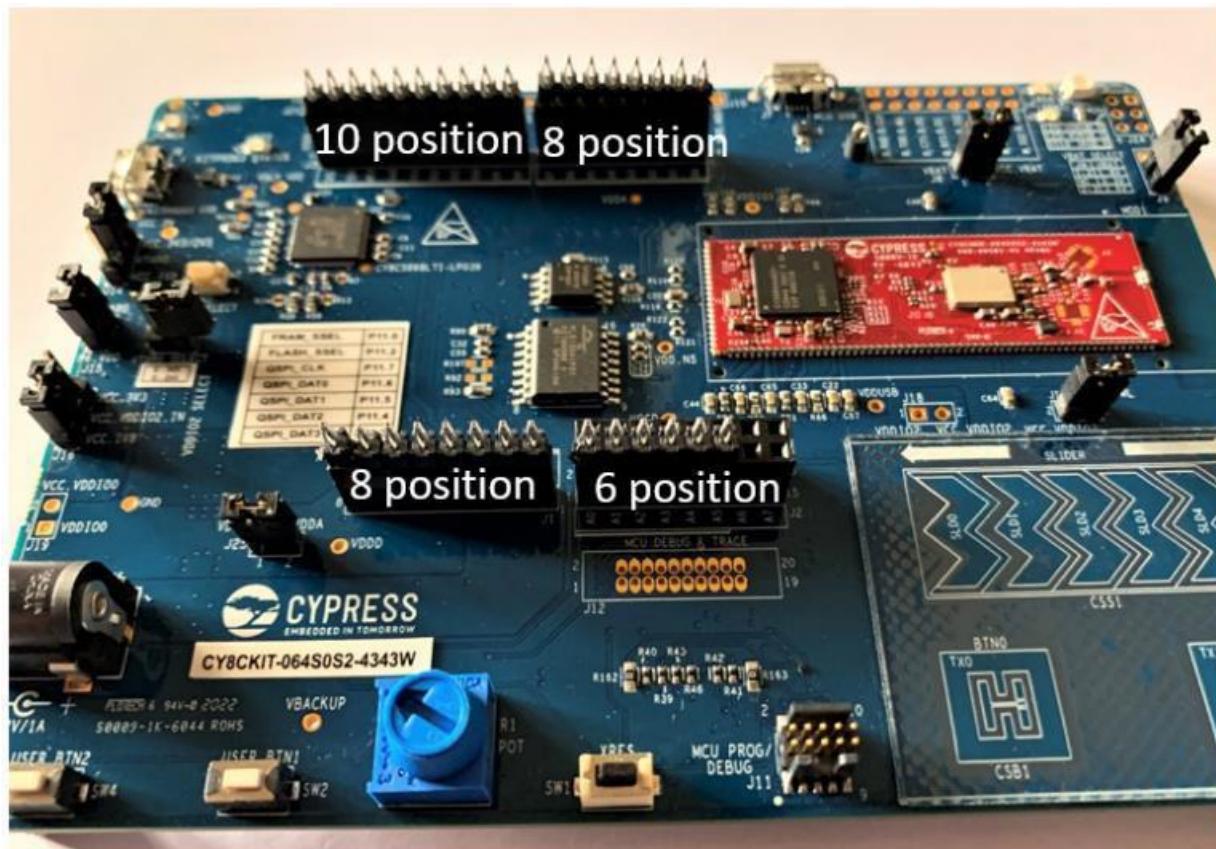
- 1 – CY8CKIT-064S0S2-4343W: PSoC6 Standard Secure – AWS WiFi Pioneer Kit
 1 – Arrow PSoC6 IoT Sensor Shield
 1 – DPS386 Shield2Go Kit for reading barometric pressure



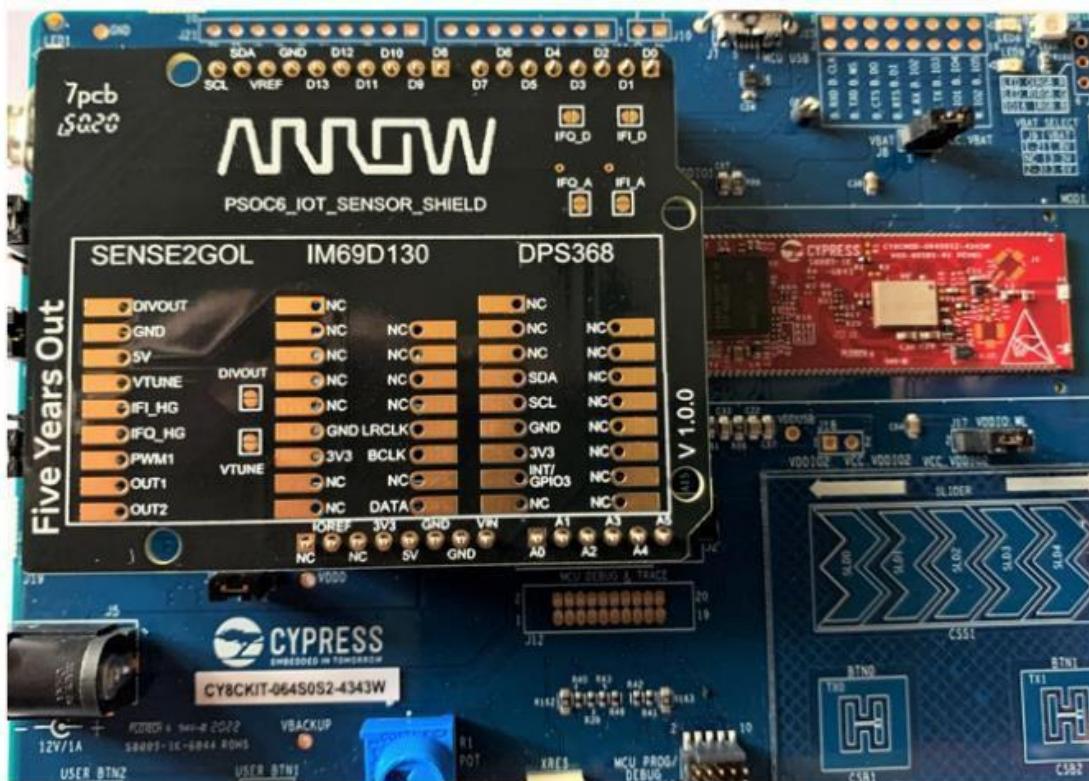
- 1 – IM69D130 Shield2Go kit (MEMS Microphone)
 4 – 10 position press fit pins
 1 – Strip of press fit sockets

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 off 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

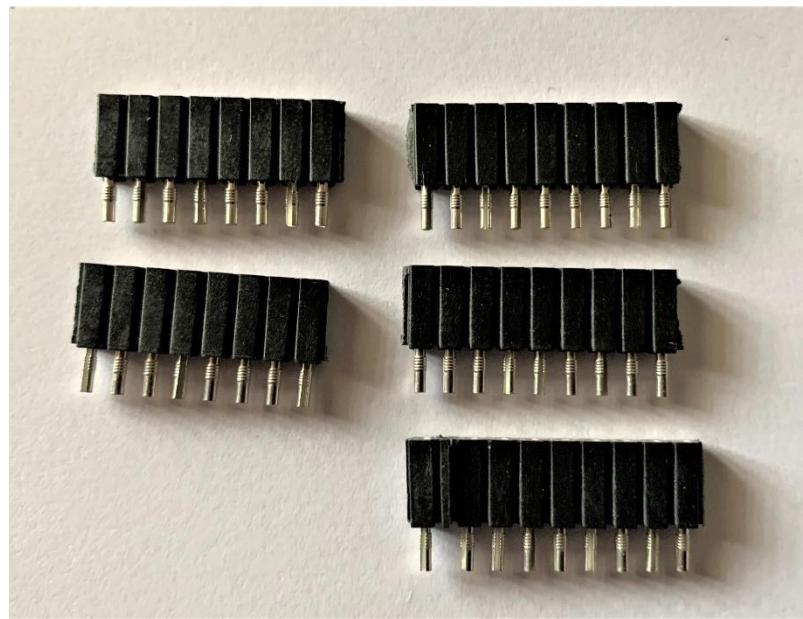
2. Put the press fit pins on the Pioneer Kit



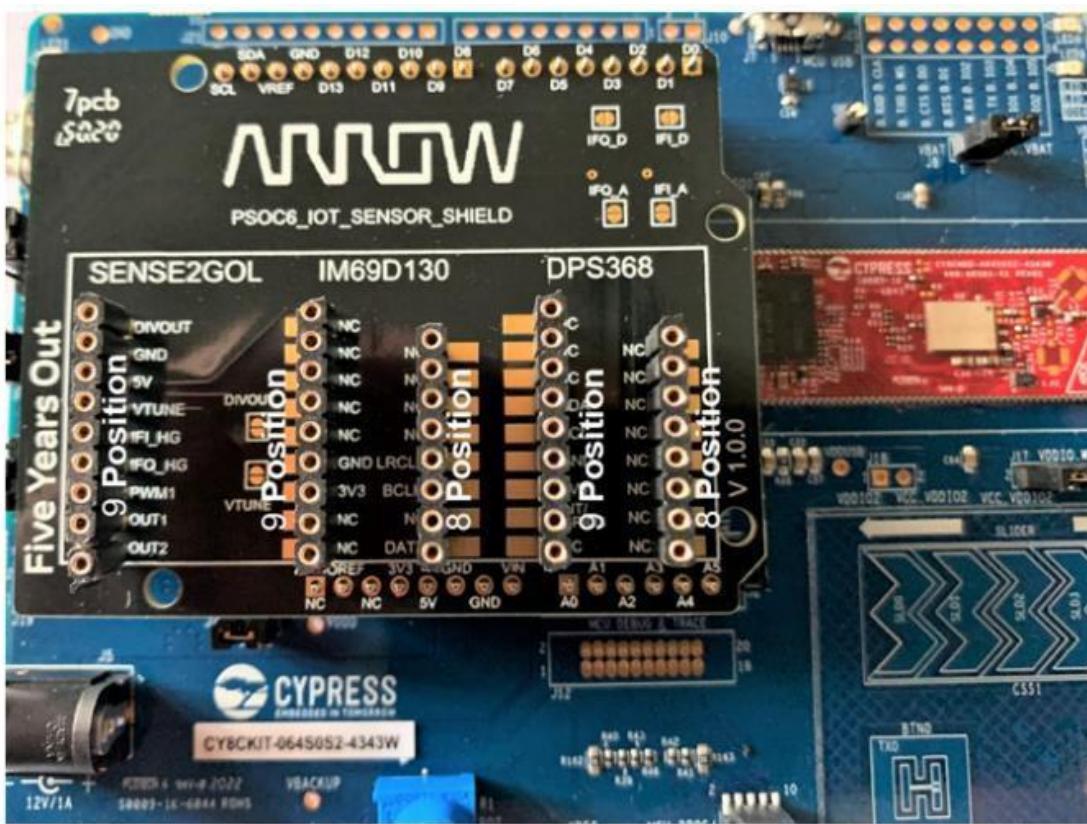
3. Mount the Arrow PSoC6 IoT Sensor Shield and press it down on the press fit pins



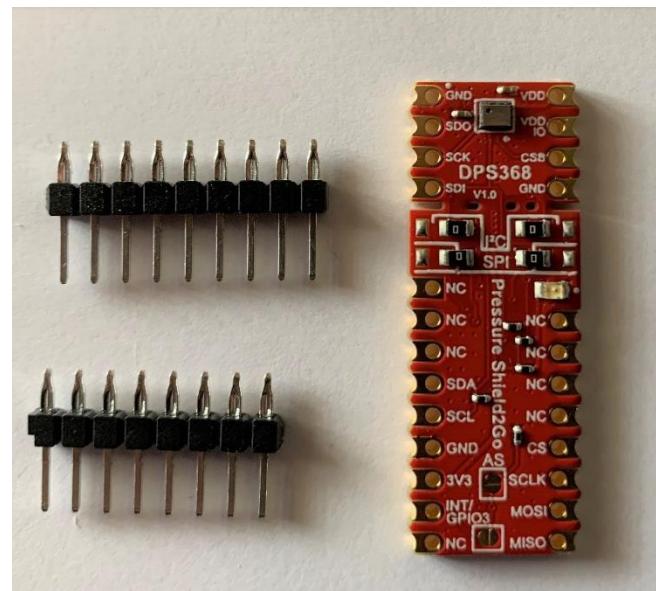
4. Break the press fit socket strip into three 9 position strips and two 8 position strips.



5. Place sockets on the Arrow PSoC6 IoT Sensor Shield



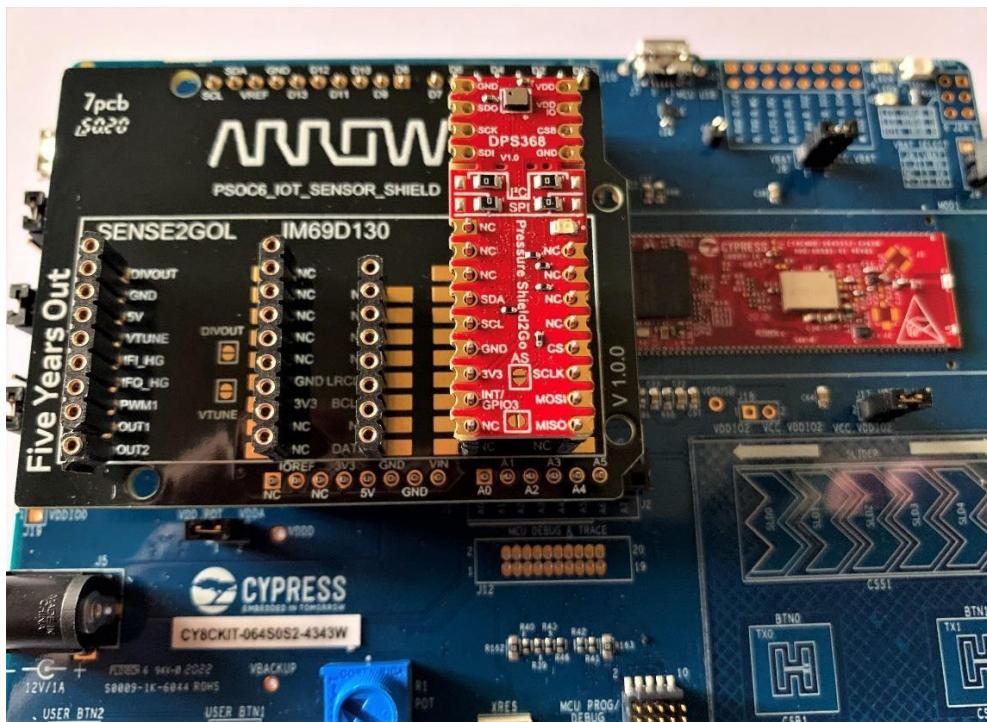
6. Break the press fit pins for the DPS368 Shield2Go Kit into one 8 position row and one 9 position row



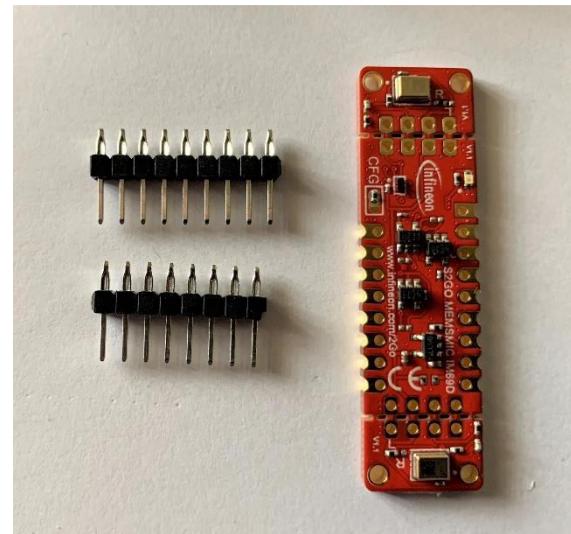
7. Place the pin press fit up in the DPS368 socket on the Arrow PSoC6 IoT Sensor Shield



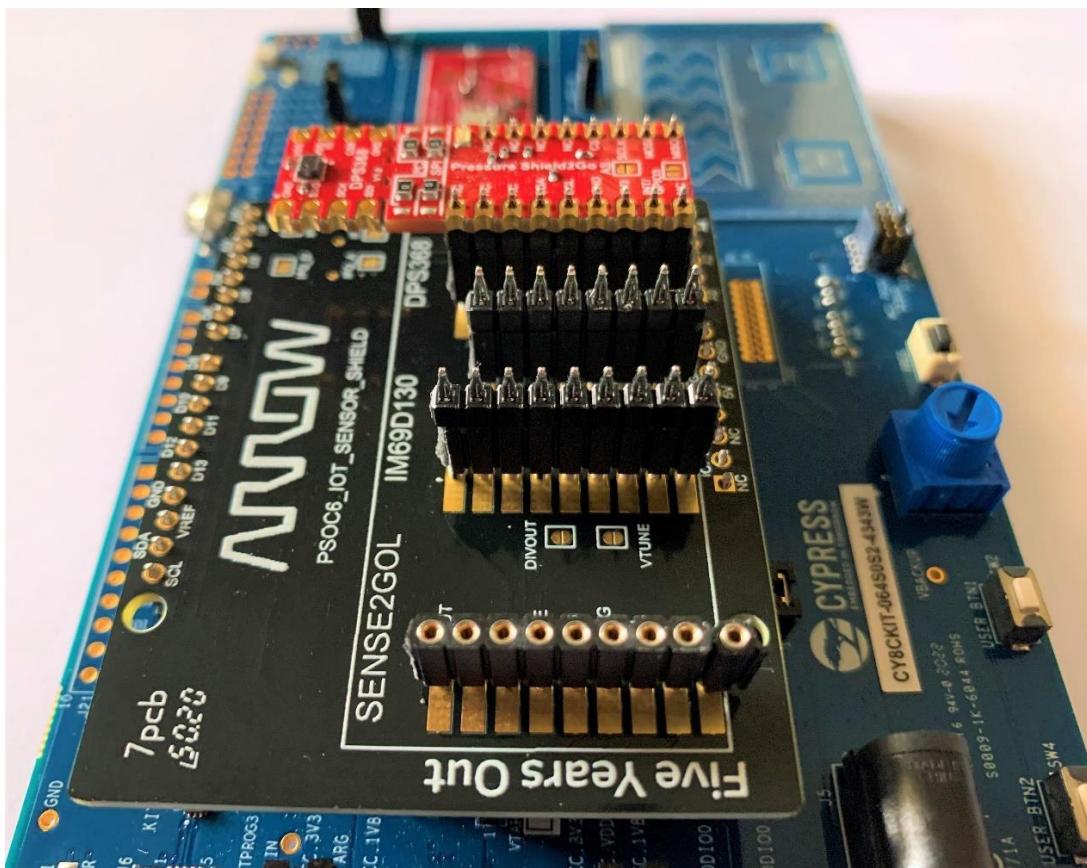
8. Place the DPS368 Shield2Go on the press fit pins components side up.



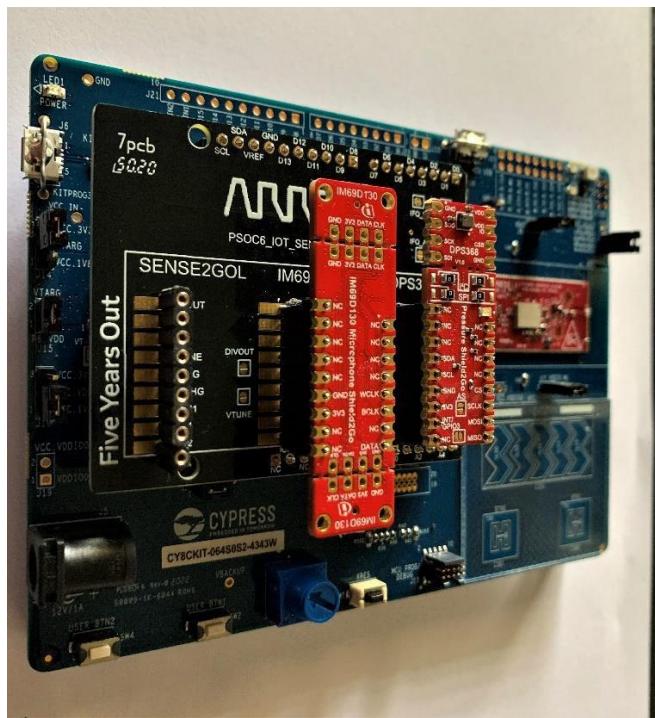
9. Break the press fit pins for the IM69D130 Shield2Go Kit into one 8 position row and one 9 position row



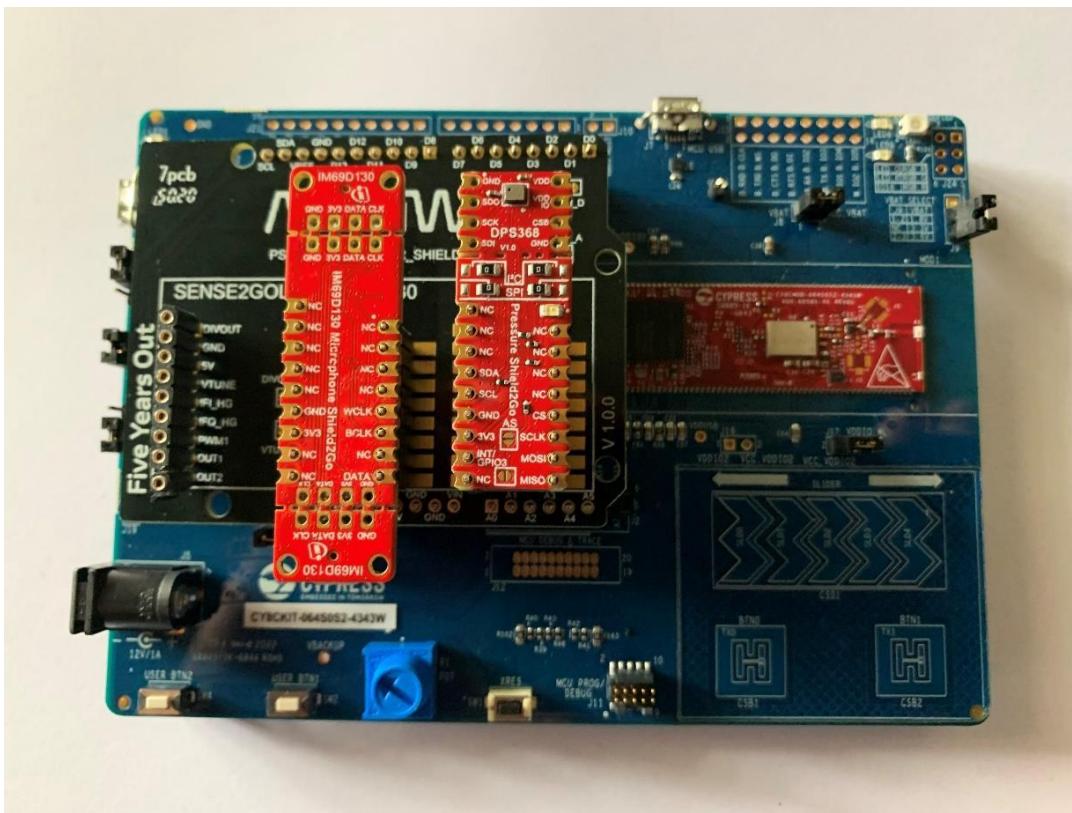
10. Place the pin press fit up in the IM69D130 socket on the Arrow PSoC6 IoT Sensor Shield



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

Or try

python reprov_helper.py -p policy/policy_single_CM0_CM4.json

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

```

cd /cygdrive/c/Projects/jq0128/amazon-freertos-202007/vendors/cypress/MTB/psoc6/psoc64tfm/security
$ cd Projects

a73744@980BHR2 /cygdrive/c/Projects/jq0128
$ cd jq0128

a73744@980BHR2 /cygdrive/c/Projects/jq0128/amazon-freertos-202007
$ cd amazon-freertos-202007

a73744@980BHR2 /cygdrive/c/Projects/jq0128/amazon-freertos-202007/vendors/cypress/MTB/psoc6/psoc64tfm/security
$ cysecuretools --policy ./policy/policy_multi_CM0_CM4_tfm.json --target CY8CKIT-064S0S2-4343W re-provision-device
2021-01-29 03:35:05,244 : C : INFO : #####
2021-01-29 03:35:05,244 : C : INFO : Provisioning packet is created
2021-01-29 03:35:05,244 : C : INFO : #####
2021-01-29 03:35:06,694 : C : INFO : Target: cy8ckit-064s0s2-4343w
2021-01-29 03:35:06,959 : P : INFO : Target type is cy8c64_sysap
2021-01-29 03:35:06,991 : P : INFO : DP IDR = 0x6ba02477 (v2 rev6)
2021-01-29 03:35:06,991 : P : INFO : AHB-AP#0 IDR = 0x84770001 (AHB-AP var0 rev8)
2021-01-29 03:35:07,006 : P : INFO : AHB-AP#0 Class 0x1 ROM table #0 @ 0xf1000000 (designer=034 part=102)
2021-01-29 03:35:07,006 : C : INFO : Use system AP
2021-01-29 03:35:07,006 : C : INFO : Probe ID: 19111301a419071100a4190700000000000000002e127069
2021-01-29 03:35:07,414 : P : INFO : DP IDR = 0x6ba02477 (v2 rev6)
2021-01-29 03:35:07,414 : C : INFO : Checking cm0 AP permissions
2021-01-29 03:35:08,286 : C : ERROR : SWD/JTAG Transfer Fault @ 0x101fb777-0x101fb777. Check the log for details
Error: Failed processing!

```

If you have an error for a missing file “entrance_exam.jwt”

```

a3196@9386R73 /cygdrive/c/Projects/P64_AWS/amazon-freertos/vendors/cypress/MTB/psoc6/psoc64tfm/security
$ cysecuretools --policy ./policy/policy_multi_CM0_CM4_tfm.json --target CY8CKIT-064S0S2-4343W re-provision-device
2021-06-09 17:28:28,992 : C : INFO : #####
2021-06-09 17:28:28,992 : C : INFO : Provisioning packet is created
2021-06-09 17:28:28,992 : C : INFO : #####
2021-06-09 17:28:29,007 : C : INFO : Target: cy8ckit-064s0s2-4343w
2021-06-09 17:28:29,043 : P : INFO : Target type is cy8c64_sysap
2021-06-09 17:28:29,072 : P : INFO : DP IDR = 0x6ba02477 (v2 rev6)
2021-06-09 17:28:29,081 : P : INFO : AHB-AP#0 IDR = 0x84770001 (AHB-AP var0 rev8)
2021-06-09 17:28:29,082 : P : INFO : AHB-AP#0 Class 0x1 ROM table #0 @ 0xf1000000 (designer=034 part=102)
2021-06-09 17:28:29,082 : C : INFO : Use system AP
2021-06-09 17:28:29,082 : C : ERROR : [Errno 2] No such file or directory: 'C:\\Projects\\P64_AWS\\amazon-freertos\\\\\\cypress\\MTB\\psoc6\\psoc64tfm\\security\\packets\\entrance_exam.jwt'. Check the log for details
Error: Failed processing!

```

Run the following command in the Modus shell:

`cysecuretools --target cyb06xx5 init` to get entrance exam file.

You will now need to start back at section 6.3 of the workshop and continue through the rest of the workshop.

“ERROR : No ACK received. Check the log for details”

Should you encounter the “No ACK” error, simply re-try provisioning. It often works the second time.

MODUS SHELL OUTPUT FROM PROVISIONING REV*A KIT

Following is result of the following command:

```
cysecuretools --policy ./policy/policy_multi_CM0_CM4_tfm.json --target CY8CKIT-064SOS2-4343W re-provision-device
```

```
a73744@980BHR2 /cygdrive/c/projects/p64_1007/amazon-freertos/vendors/cypress/MTB/psoc6/psoc64tfm/security
$ cysecuretools --policy ./policy/policy_multi_CM0_CM4_tfm.json --target CY8CKIT-064SOS2-4343W re-provision-device
2021-10-07 17:29:29,034 : C : WARN  : There is gap between regions 269926400:270254080 and 270303232:270336000 (49152 bytes)
2021-10-07 17:29:29,034 : C : WARN  : Policy validation finished with warnings
2021-10-07 17:29:29,050 : C : INFO  : #####
2021-10-07 17:29:29,050 : C : INFO  : Provisioning packet is created
2021-10-07 17:29:29,050 : C : INFO  : #####
2021-10-07 17:29:29,066 : C : WARN  : There is gap between regions 269926400:270254080 and 270303232:270336000 (49152 bytes)
2021-10-07 17:29:29,081 : C : WARN  : Policy validation finished with warnings
2021-10-07 17:29:29,081 : C : INFO  : Target: cy8ckit-064s0s2-4343w
2021-10-07 17:29:29,783 : P : INFO  : Target type is cy8c64_ssysp
2021-10-07 17:29:29,815 : P : INFO  : DP IDR = 0x6ba02477 (v2 rev6)
2021-10-07 17:29:29,815 : P : INFO  : AHB-AP#0 IDR = 0x84770001 (AHB-AP var0 rev8)
2021-10-07 17:29:29,836 : P : INFO  : AHB-AP#0 Class 0x1 ROM table #0 @ 0xf1000000 (designer=034 part=102)
2021-10-07 17:29:29,836 : C : INFO  : Use system AP
2021-10-07 17:29:29,836 : C : INFO  : Probe ID: 19111130123030e200323030e000000000000000002e127069
2021-10-07 17:29:29,852 : C : INFO  : Secure Flash Boot version: 4.0.2.1842
2021-10-07 17:29:30,638 : C : INFO  : Device CyBootloader version: 2.0.0.4117
2021-10-07 17:29:30,654 : C : INFO  : Package CyBootloader version: 2.0.0.4117
2021-10-07 17:29:31,055 : C : INFO  : Chip protection state: Secure
2021-10-07 17:29:31,655 : P : INFO  : DP IDR = 0x6ba02477 (v2 rev6)
2021-10-07 17:29:31,655 : C : INFO  : Checking cm0 AP permissions
2021-10-07 17:29:35,535 : C : INFO  : cm0 AP open
2021-10-07 17:29:35,535 : P : ERROR  : link exception during target disconnect:
Traceback (most recent call last):
  File "c:\users\173744\modustoolbox\tools_2.3\python-3.7.155\lib\site-packages\pyocd\probe\cmsis_dap_probe.py", line 256, in
read_dp_result_callback
    return result()
  File "c:\users\173744\modustoolbox\tools_2.3\python-3.7.155\lib\site-packages\pyocd\probe\pydapaccess\dap_access_cmsis_dap.py",
line 830, in read_reg_cb
    res = transfer.get_result()
  File "c:\users\173744\modustoolbox\tools_2.3\python-3.7.155\lib\site-packages\pyocd\probe\pydapaccess\dap_access_cmsis_dap.py",
line 143, in get_result
    self.daplink.flush()
  File "c:\users\173744\modustoolbox\tools_2.3\python-3.7.155\lib\site-packages\pyocd\utility\concurrency.py", line 28, in
_locking
    return func(self, *args, **kwargs)
  File "c:\users\173744\modustoolbox\tools_2.3\python-3.7.155\lib\site-packages\pyocd\probe\pydapaccess\dap_access_cmsis_dap.py",
line 667, in flush
    self._read_packet()
  File "c:\users\173744\modustoolbox\tools_2.3\python-3.7.155\lib\site-packages\pyocd\utility\concurrency.py", line 28, in
_locking
    return func(self, *args, **kwargs)
  File "c:\users\173744\modustoolbox\tools_2.3\python-3.7.155\lib\site-packages\pyocd\probe\pydapaccess\dap_access_cmsis_dap.py",
line 915, in _read_packet
    decoded_data = cmd.decode_data(raw_data)
  File "c:\users\173744\modustoolbox\tools_2.3\python-3.7.155\lib\site-packages\pyocd\probe\pydapaccess\dap_access_cmsis_dap.py",
line 451, in decode_data
    data = self._decode_transfer_data(data)
  File "c:\users\173744\modustoolbox\tools_2.3\python-3.7.155\lib\site-packages\pyocd\probe\pydapaccess\dap_access_cmsis_dap.py",
line 358, in _decode_transfer_data
    self._check_response(data[2])
  File "c:\users\173744\modustoolbox\tools_2.3\python-3.7.155\lib\site-packages\pyocd\probe\pydapaccess\dap_access_cmsis_dap.py",
line 341, in _check_response
    raise DAPAccessIntf.TransferError("No ACK received")
pyocd.probe.pydapaccess.dap_access_api.DAPAccessIntf.TransferError: No ACK received

The above exception was the direct cause of the following exception:

Traceback (most recent call last):
  File "c:\users\173744\modustoolbox\tools_2.3\python-3.7.155\lib\site-packages\pyocd\board\board.py", line 98, in uninit
    self.target.disconnect(resume)
  File "c:\users\173744\modustoolbox\tools_2.3\python-3.7.155\lib\site-packages\pyocd\core\coresight_target.py", line 306, in
disconnect
    self.dp.power_down_debug()
  File "c:\users\173744\modustoolbox\tools_2.3\python-3.7.155\lib\site-packages\pyocd\coresight\dap.py", line 399, in
power_down_debug
    r = self.read_reg(DP_CTRL_STAT)
```

```

File "c:\users\al73744\modustoolbox\tools_2.3\python-3.7.155\lib\site-packages\pyocd\coresight\dap.py", line 345, in read_reg
    return self.read_dp(addr, now)
File "c:\users\al73744\modustoolbox\tools_2.3\python-3.7.155\lib\site-packages\pyocd\coresight\dap.py", line 519, in read_dp
    return read_dp_cb()
File "c:\users\al73744\modustoolbox\tools_2.3\python-3.7.155\lib\site-packages\pyocd\coresight\dap.py", line 507, in read_dp_cb
    result = result_cb()
File "c:\users\al73744\modustoolbox\tools_2.3\python-3.7.155\lib\site-packages\pyocd\probe\CMSIS_DAP_Probe.py", line 258, in
read_dp_result_callback
    six.raise_from(self._convert_exception(error), error)
File "<string>", line 3, in raise_from
pyocd.core.exceptions.TransferError: No ACK received
2021-10-07 17:29:35,594 : C : INFO : Target: cy8ckit-064s0s2-4343w
2021-10-07 17:29:35,773 : P : INFO : Target type is cy8c64xa_cm0_full_flash
2021-10-07 17:29:35,804 : P : INFO : DP IDR = 0x6ba02477 (v2 rev6)
2021-10-07 17:29:35,804 : P : INFO : AHB-AP#0 IDR = 0x84770001 (AHB-AP var0 rev8)
2021-10-07 17:29:35,804 : P : INFO : AHB-AP#1 IDR = 0x84770001 (AHB-AP var0 rev8)
2021-10-07 17:29:35,820 : P : INFO : AHB-AP#0 Class 0x1 ROM table #0 @ 0xf1000000 (designer=034 part=102)
2021-10-07 17:29:35,820 : P : INFO : AHB-AP#1 Class 0x1 ROM table #0 @ 0xf0000000 (designer=034 part=102)
2021-10-07 17:29:35,835 : P : INFO : [0]<e00ff000:ROM class=1 designer=43b part=4c0>
2021-10-07 17:29:35,835 : P : INFO : AHB-AP#1 Class 0x1 ROM table #1 @ 0xe00ff000 (designer=43b part=4c0)
2021-10-07 17:29:35,835 : P : INFO : [0]<e000e000:SCS-M0+ class=14 designer=43b part=008>
2021-10-07 17:29:35,848 : P : INFO : [1]<e0001000:DWT-M0+ class=14 designer=43b part=00a>
2021-10-07 17:29:35,848 : P : INFO : [2]<e0002000:BPU class=14 designer=43b part=00b>
2021-10-07 17:29:35,848 : P : INFO : [1]<f0002000:CTI class=9 designer=43b part=9a6 devtype=14 archid=1a14 devid=1040800:0:0>
2021-10-07 17:29:35,863 : P : INFO : [2]<f0003000:MTB-M0+ class=9 designer=43b part=932 devtype=31 archid=0a31 devid=0:0:0>
2021-10-07 17:29:35,863 : P : INFO : CPU core #1 is Cortex-M0+ r0p1
2021-10-07 17:29:35,863 : P : INFO : 2 hardware breakpoints
2021-10-07 17:29:35,879 : P : INFO : 4 hardware breakpoints, 0 literal comparators
2021-10-07 17:29:35,879 : C : INFO : Use system AP
2021-10-07 17:29:35,895 : P : INFO : Clearing TEST_MODE bit...
2021-10-07 17:29:35,895 : C : INFO : Probe ID: 1911130123030e200323030e00000000000000002e127069
2021-10-07 17:29:36,480 : P : INFO : DP IDR = 0x6ba02477 (v2 rev6)
2021-10-07 17:29:36,480 : C : INFO : Erase SWAP status partition memory region:
2021-10-07 17:29:36,480 : C : INFO : Use cm0 AP
2021-10-07 17:29:36,496 : P : INFO : Clearing TEST_MODE bit...
2021-10-07 17:29:36,496 : C : INFO : erasing address 0x101c8000, size 0x8000 ...
2021-10-07 17:29:36,496 : P : INFO : Acquiring target...
2021-10-07 17:29:36,982 : P : INFO : DP IDR = 0x6ba02477 (v2 rev6)
2021-10-07 17:29:36,982 : P : INFO : DP IDR = 0x6ba02477 (v2 rev6)
2021-10-07 17:29:36,997 : P : INFO : DP IDR = 0x6ba02477 (v2 rev6)
2021-10-07 17:29:36,997 : P : INFO : DP IDR = 0x6ba02477 (v2 rev6)
2021-10-07 17:29:37,051 : P : INFO : Erasing sector 0x101c8000 (512 bytes)
2021-10-07 17:29:37,066 : P : INFO : Erasing sector 0x101c8200 (512 bytes)
2021-10-07 17:29:37,097 : P : INFO : Erasing sector 0x101c8400 (512 bytes)
2021-10-07 17:29:37,113 : P : INFO : Erasing sector 0x101c8600 (512 bytes)
2021-10-07 17:29:37,129 : P : INFO : Erasing sector 0x101c8800 (512 bytes)
2021-10-07 17:29:37,151 : P : INFO : Erasing sector 0x101c8a00 (512 bytes)
2021-10-07 17:29:37,166 : P : INFO : Erasing sector 0x101c8c00 (512 bytes)
2021-10-07 17:29:37,182 : P : INFO : Erasing sector 0x101c8e00 (512 bytes)
2021-10-07 17:29:37,198 : P : INFO : Erasing sector 0x101c9000 (512 bytes)
2021-10-07 17:29:37,229 : P : INFO : Erasing sector 0x101c9200 (512 bytes)
2021-10-07 17:29:37,251 : P : INFO : Erasing sector 0x101c9400 (512 bytes)
2021-10-07 17:29:37,267 : P : INFO : Erasing sector 0x101c9600 (512 bytes)
2021-10-07 17:29:37,282 : P : INFO : Erasing sector 0x101c9800 (512 bytes)
2021-10-07 17:29:37,298 : P : INFO : Erasing sector 0x101c9a00 (512 bytes)
2021-10-07 17:29:37,329 : P : INFO : Erasing sector 0x101c9c00 (512 bytes)
2021-10-07 17:29:37,351 : P : INFO : Erasing sector 0x101c9e00 (512 bytes)
2021-10-07 17:29:37,367 : P : INFO : Erasing sector 0x101ca000 (512 bytes)
2021-10-07 17:29:37,383 : P : INFO : Erasing sector 0x101ca200 (512 bytes)
2021-10-07 17:29:37,398 : P : INFO : Erasing sector 0x101ca400 (512 bytes)
2021-10-07 17:29:37,414 : P : INFO : Erasing sector 0x101ca600 (512 bytes)
2021-10-07 17:29:37,433 : P : INFO : Erasing sector 0x101ca800 (512 bytes)
2021-10-07 17:29:37,451 : P : INFO : Erasing sector 0x101caa00 (512 bytes)
2021-10-07 17:29:37,467 : P : INFO : Erasing sector 0x101cac00 (512 bytes)
2021-10-07 17:29:37,482 : P : INFO : Erasing sector 0x101cae00 (512 bytes)
2021-10-07 17:29:37,498 : P : INFO : Erasing sector 0x101cb000 (512 bytes)
2021-10-07 17:29:37,514 : P : INFO : Erasing sector 0x101cb200 (512 bytes)
2021-10-07 17:29:37,529 : P : INFO : Erasing sector 0x101cb400 (512 bytes)
2021-10-07 17:29:37,551 : P : INFO : Erasing sector 0x101cb600 (512 bytes)
2021-10-07 17:29:37,567 : P : INFO : Erasing sector 0x101cb800 (512 bytes)
2021-10-07 17:29:37,583 : P : INFO : Erasing sector 0x101cba00 (512 bytes)
2021-10-07 17:29:37,598 : P : INFO : Erasing sector 0x101cbc00 (512 bytes)
2021-10-07 17:29:37,614 : P : INFO : Erasing sector 0x101cbe00 (512 bytes)
2021-10-07 17:29:37,630 : P : INFO : Erasing sector 0x101cc000 (512 bytes)
2021-10-07 17:29:37,652 : P : INFO : Erasing sector 0x101cc200 (512 bytes)
2021-10-07 17:29:37,667 : P : INFO : Erasing sector 0x101cc400 (512 bytes)
2021-10-07 17:29:37,683 : P : INFO : Erasing sector 0x101cc600 (512 bytes)
2021-10-07 17:29:37,699 : P : INFO : Erasing sector 0x101cc800 (512 bytes)
2021-10-07 17:29:37,714 : P : INFO : Erasing sector 0x101cca00 (512 bytes)
2021-10-07 17:29:37,730 : P : INFO : Erasing sector 0x101ccc00 (512 bytes)
2021-10-07 17:29:37,752 : P : INFO : Erasing sector 0x101cce00 (512 bytes)
2021-10-07 17:29:37,768 : P : INFO : Erasing sector 0x101cd000 (512 bytes)
2021-10-07 17:29:37,783 : P : INFO : Erasing sector 0x101cd200 (512 bytes)

```

```

2021-10-07 17:29:37,799 : P : INFO : Erasing sector 0x101cd400 (512 bytes)
2021-10-07 17:29:37,814 : P : INFO : Erasing sector 0x101cd600 (512 bytes)
2021-10-07 17:29:37,830 : P : INFO : Erasing sector 0x101cd800 (512 bytes)
2021-10-07 17:29:37,852 : P : INFO : Erasing sector 0x101cda00 (512 bytes)
2021-10-07 17:29:37,868 : P : INFO : Erasing sector 0x101cdc00 (512 bytes)
2021-10-07 17:29:37,883 : P : INFO : Erasing sector 0x101cde00 (512 bytes)
2021-10-07 17:29:37,899 : P : INFO : Erasing sector 0x101ce000 (512 bytes)
2021-10-07 17:29:37,915 : P : INFO : Erasing sector 0x101ce200 (512 bytes)
2021-10-07 17:29:37,930 : P : INFO : Erasing sector 0x101ce400 (512 bytes)
2021-10-07 17:29:37,952 : P : INFO : Erasing sector 0x101ce600 (512 bytes)
2021-10-07 17:29:37,968 : P : INFO : Erasing sector 0x101ce800 (512 bytes)
2021-10-07 17:29:37,984 : P : INFO : Erasing sector 0x101cea00 (512 bytes)
2021-10-07 17:29:37,999 : P : INFO : Erasing sector 0x101cec00 (512 bytes)
2021-10-07 17:29:38,015 : P : INFO : Erasing sector 0x101cee00 (512 bytes)
2021-10-07 17:29:38,031 : P : INFO : Erasing sector 0x101cf000 (512 bytes)
2021-10-07 17:29:38,053 : P : INFO : Erasing sector 0x101cf200 (512 bytes)
2021-10-07 17:29:38,068 : P : INFO : Erasing sector 0x101cf400 (512 bytes)
2021-10-07 17:29:38,084 : P : INFO : Erasing sector 0x101cf600 (512 bytes)
2021-10-07 17:29:38,100 : P : INFO : Erasing sector 0x101cf800 (512 bytes)
2021-10-07 17:29:38,115 : P : INFO : Erasing sector 0x101cf900 (512 bytes)
2021-10-07 17:29:38,131 : P : INFO : Erasing sector 0x101cfcc00 (512 bytes)
2021-10-07 17:29:38,153 : P : INFO : Erasing sector 0x101cfe00 (512 bytes)
2021-10-07 17:29:38,169 : C : INFO : Erasing complete
2021-10-07 17:29:38,169 : C : INFO : Use system AP
2021-10-07 17:29:38,184 : P : INFO : Clearing TEST_MODE bit...
2021-10-07 17:29:38,184 : C : INFO : Erase SCRATCH memory region:
2021-10-07 17:29:38,200 : C : INFO : Use cm0 AP
2021-10-07 17:29:38,216 : P : INFO : Clearing TEST_MODE bit...
2021-10-07 17:29:38,216 : C : INFO : erasing address 0x18280000, size 0x80000 ...
2021-10-07 17:29:38,216 : P : INFO : Acquiring target...
2021-10-07 17:29:38,717 : P : INFO : DP IDR = 0x6ba02477 (v2 rev6)
2021-10-07 17:29:38,717 : P : INFO : DP IDR = 0x6ba02477 (v2 rev6)
2021-10-07 17:29:38,732 : P : INFO : DP IDR = 0x6ba02477 (v2 rev6)
2021-10-07 17:29:39,988 : P : INFO : DP IDR = 0x6ba02477 (v2 rev6)
2021-10-07 17:29:40,258 : P : INFO : Erasing sector 0x18280000 (262144 bytes)
2021-10-07 17:29:40,674 : P : INFO : Erasing sector 0x182c0000 (262144 bytes)
2021-10-07 17:29:41,122 : C : INFO : Erasing complete
2021-10-07 17:29:41,122 : C : INFO : Use system AP
2021-10-07 17:29:41,138 : P : INFO : Clearing TEST_MODE bit...
2021-10-07 17:29:41,138 : C : INFO : Read FlashBoot firmware status:
2021-10-07 17:29:41,138 : C : INFO : FlashBoot firmware status = 0xa1000101
2021-10-07 17:29:41,138 : C : INFO : Received FB_FW_STATUS = 0xa0000000
2021-10-07 17:29:41,138 : C : INFO : Expected FB_FW_STATUS = 0xa0000000
2021-10-07 17:29:41,138 : C : INFO : BOOT slot will remain the same and can affect rollback counter
2021-10-07 17:29:41,138 : C : INFO : Erase main smif slots:
2021-10-07 17:29:41,138 : C : INFO : Use cm0 AP
2021-10-07 17:29:41,153 : P : INFO : Clearing TEST_MODE bit...
2021-10-07 17:29:41,153 : C : INFO : erasing address 0x18024400, size 0x11c000 ...
2021-10-07 17:29:41,160 : P : INFO : Acquiring target...
2021-10-07 17:29:41,654 : P : INFO : DP IDR = 0x6ba02477 (v2 rev6)
2021-10-07 17:29:41,661 : P : INFO : DP IDR = 0x6ba02477 (v2 rev6)
2021-10-07 17:29:41,661 : P : INFO : DP IDR = 0x6ba02477 (v2 rev6)
2021-10-07 17:29:41,661 : P : INFO : DP IDR = 0x6ba02477 (v2 rev6)
2021-10-07 17:29:41,908 : P : WARN : sector address 0x18024400 is unaligned
2021-10-07 17:29:41,908 : P : INFO : Erasing sector 0x18000000 (262144 bytes)
2021-10-07 17:29:42,362 : P : INFO : Erasing sector 0x18040000 (262144 bytes)
2021-10-07 17:29:42,825 : P : INFO : Erasing sector 0x18080000 (262144 bytes)
2021-10-07 17:29:43,242 : P : INFO : Erasing sector 0x180c0000 (262144 bytes)
2021-10-07 17:29:43,712 : P : INFO : Erasing sector 0x18100000 (262144 bytes)
2021-10-07 17:29:44,159 : P : INFO : Erasing sector 0x18140000 (262144 bytes)
2021-10-07 17:29:44,566 : C : INFO : Erasing complete
2021-10-07 17:29:44,566 : C : INFO : Use system AP
2021-10-07 17:29:44,582 : P : INFO : Clearing TEST_MODE bit...
2021-10-07 17:29:47,588 : C : INFO : Use cm0 AP
2021-10-07 17:29:47,603 : P : INFO : Clearing TEST_MODE bit...
2021-10-07 17:29:47,603 : C : INFO : Programming bootloader 'C:\projects\p64_1007\amazon-freeertos\vendors\cypress\MTB\psoc6\psoc64tfm\security\prebuilt\CyBootloader_WithLogs_swap\CypressBootloader_CM0p.hex':
2021-10-07 17:29:47,719 : P : INFO : Acquiring target...
2021-10-07 17:29:48,204 : P : INFO : DP IDR = 0x6ba02477 (v2 rev6)
2021-10-07 17:29:48,220 : P : INFO : DP IDR = 0x6ba02477 (v2 rev6)
2021-10-07 17:29:48,245 : P : INFO : DP IDR = 0x6ba02477 (v2 rev6)
2021-10-07 17:29:48,250 : P : INFO : DP IDR = 0x6ba02477 (v2 rev6)
2021-10-07 17:29:49,492 : P : INFO : DP IDR = 0x6ba02477 (v2 rev6)
[=====] 100%
2021-10-07 17:29:50,525 : P : INFO : Acquiring target...
2021-10-07 17:29:51,010 : P : INFO : DP IDR = 0x6ba02477 (v2 rev6)
2021-10-07 17:29:51,026 : P : INFO : DP IDR = 0x6ba02477 (v2 rev6)
2021-10-07 17:29:51,040 : P : INFO : DP IDR = 0x6ba02477 (v2 rev6)
2021-10-07 17:29:52,297 : P : INFO : DP IDR = 0x6ba02477 (v2 rev6)
2021-10-07 17:29:52,297 : P : INFO : Erased 0 bytes (0 sectors), programmed 0 bytes (0 pages), skipped 53760 bytes (105 pages) at 11.43 kB/s
2021-10-07 17:29:52,313 : C : INFO : Programming bootloader complete
2021-10-07 17:29:52,313 : C : INFO : Use system AP

```

```

2021-10-07 17:29:52,313 : P : INFO  : Clearing TEST_MODE bit...
2021-10-07 17:29:52,313 : P : INFO  : Clearing TEST_MODE bit...
2021-10-07 17:29:52,313 : C : INFO  : Target: cy8ckit-064s0s2-4343w
2021-10-07 17:29:52,495 : P : INFO  : Target type is cy8c64_sysap
2021-10-07 17:29:52,542 : P : INFO  : DP IDR = 0x6ba02477 (v2 rev6)
2021-10-07 17:29:52,542 : P : INFO  : AHB-AP#0 IDR = 0x84770001 (AHB-AP var0 rev8)
2021-10-07 17:29:52,564 : P : INFO  : AHB-AP#0 Class 0x1 ROM table #0 @ 0xf1000000 (designer=034 part=102)
2021-10-07 17:29:52,566 : C : INFO  : Use system AP
2021-10-07 17:29:52,566 : C : INFO  : Probe ID: 1911130123030e200323030e00000000000000002e127069
2021-10-07 17:29:53,161 : P : INFO  : DP IDR = 0x6ba02477 (v2 rev6)
2021-10-07 17:29:56,166 : C : INFO  : Run provisioning syscall:
2021-10-07 17:29:56,166 : C : INFO  : JWT packet size = 7088
2021-10-07 17:30:18,411 : C : INFO  : Device response =
'ewoJImFsZyI6CSJFUz1N1IKfQ.ewoJInR5cGuigOkgiREVWx1JTUCIsCgkizGV2X21kIjoJInNpbGljb25faWQ9RTRBMC4xMiwgZmFtaWx5X21kPTEwMiIsCgkizG11X
21kIjoJewoJCSJs3QiOgk5MDA2MDM5LaoJCSJ3YWz1ciI6CTExLAoJCSJ4cG9z1joJNjUsCgkJInlw3MiOgk2MSwKCQkizGF5IjoJMTYsCgkJIm1vbRoIjoJNCwKCQ
kieVVhciI6CTEyMa0Jf5wKCSJkZXZfcHViX2tlesI6CXsKCQkiz3R5IjoJlkVDIiwKCQkidoJNljoJInNpZyIsCgkJImtpZC16CSIyiwi
KCQkiaeCI6CSJGMk90YXJSeDvyWnpfYWNBy3dWUzBhSnAzdEFkaFjkRGxNelpTY3N2eEqwiwKCQkiaS16CSJYVlhraTI5MTBmdFN1Y3dfSS0xbEdrM19JR1pKTUoYLVd6
Z1Q30Eq3eEzN1goJsfwKCSJvZw1fcHVix2tlesI6CXsKCQkiz3J2IjoJ1AtMjU2IiwKCQkia3R5IjoJlkVDIiwKCQkidoJNljoJInNpZyIsCgkJImtpZC16CSIiiwKC
QkieCI6CSJ2ZmI3X2pld1R4cEZWSU5jWGRyWlFKqFyQzVpZ3JOMEJMYzc4M0zpZ3JNIIiwKCQkiaeS16CSI5ckJCvUtyenBqMUE1SzdmFB0RWFKZHnmzbzKal93c0Y3TF
RaTGMtc1BN1goJsfwKCSJwcm9X21kIjoJIm15X3RoaW5nIiwlKCSJjb21wbGV0ZSI6CRydWUKfQ.ZliLXX1_CIHyYY5xoYzNd-
EURqieEw7yaxBMZtsNHeKziBHJu59W_eXnG-LHqKQzN5YvLm51GSGByV78KUYi5A'
2021-10-07 17:30:18,411 : C : INFO  : Saved device response to 'C:\projects\p64_1007\amazon-
freertos\vendors\cypress\MTB\psoc6\psoc64tfm\security\packets\device_response.jwt'
2021-10-07 17:30:18,997 : P : INFO  : DP IDR = 0x6ba02477 (v2 rev6)
2021-10-07 17:30:22,015 : C : INFO  : FlashBoot firmware status = 0xa1000101
2021-10-07 17:30:22,015 : C : INFO  : ****RE-PROVISIONING PASSED*****
2021-10-07 17:30:22,015 : C : INFO  : ****

```

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
2.01	Draft	G Carson	Attempting Rev*A of P64 AWS kit (Rev9 of board) and current Amazon-FreeRTOS 20210700.
2.02	Draft	G.Carson for R.Smith	Reviewed updated Rusty Smith had made to a parallel copy and incorporated them here, merging the two documents. Rusty's document was labeled 3.01 on Teams. It was marked as 1.7 internally.

2.03	Draft	G.Carson	Running through full procedure from start to finish.
2.04	Draft	G.Carson	Sent a .pdf copy out of 2.03 so captured that by updating to 2.04
2.05	Draft	G.Carson	Updated Introductory information for each chapter to explain what was being performed in the chapter and how to know if it was already done.
2.06	Draft	G.Carson	Turning off visibility of all changes and stepping through the Kit Provisioning and ModusToolbox sections (Chapters 6 through 11)
2.07	Draft	G.Carson	One final pass through to clean up formatting before peer review.
2.08	Draft	R.Smith	Peer Review comments
3.00	10/24/2021	G.Carson	All changes accepted & posted to GitHub https://github.com/ArrowElectronicsESC/P64_Webinar/tree/master/Pre_Work_Examples
3.01	12/17/2021	R.Smith	Update AWS section to current AWS version