arrayent

Get Connected. Get Closer.™

Arrayent DevKit Sample Application Quick Start Guide For TI CC3220

28 March 2017

TABLE OF CONTENTS

Chapter 1. Overview

This document will guide you in getting started with the Arrayent DevKit Sample application (now on referred as sample application) on TI CC3220 Launch Pad hardware platforms:

In this guide, you will learn how to:

1. Load and run the pre-built binary of the sample application onto your development board.
2. Add the sample application source code, Arrayent Connect Agent (ACA) library and support library to your platform SDK.
3. Build and load sample application onto your development board.
4. Control and Monitor your development board over the Internet, by either using Arrayent web apps or Arrayent's "DevKit" android application.

1. Audience and Scope

This document is split into three major sections: setting up the sample application, configuring the Arrayent Connect Cloud (ACC) and monitoring/controlling the development board using Arrayent's web application.

The sections of this document that pertain to building and loading the sample application onto a development board are primarily intended for embedded developers. For these sections we assume that you are familiar with your embedded platform's SDK. You should be able to build and execute the other sample applications in your embedded SDK before attempting to use the Arrayent sample application.

A pre-built binary of the sample application is provided with the DevKit release package for anyone who wants to see the live demo of the development board connecting to the internet. Once the development board has been loaded with the sample application, the workflow for configuring the sample application, cloud and monitoring/controlling the board with a web app is simple and straightforward. This part of the guide is simple to follow for anyone.

2. Prerequisites

In this guide we assume that you signed up for a DevKit via our online form and received an email containing user account credentials, device credentials, web application URLs, and links for software downloads. If you received your DevKit in a different manner you may need to contact your Arrayent support person for some help.

To build and download the sample application to your development board you will need to use command line utilities. We assume familiarity with basic DOS commands like cd, dir, etc.

3. Supported Software and Hardware Platforms

The sample application was tested with the following combination of operating system, platform SDK and development board. The sample application and/or Arrayent library may work with other versions of evaluation board, module, or SDK. But Arrayent cannot guarantee or verify this. Arrayent only supports the combination of hardware and software mentioned below:

TI CC3220 Launchpad:

Operating System

- Windows 8.1

CCS

- Version 7.0

Simplelink CC3220 SDK

- SDK Version: 1.02.02.00
- Download TI CC3220 SDK installer from below link:
  TBD

TI Launchpad

- Development Board: CC3220-LaunchXL Rev-B: Board version is located on the front side of the board as shown in the below figure.

Uniflash

- Version: 4.0

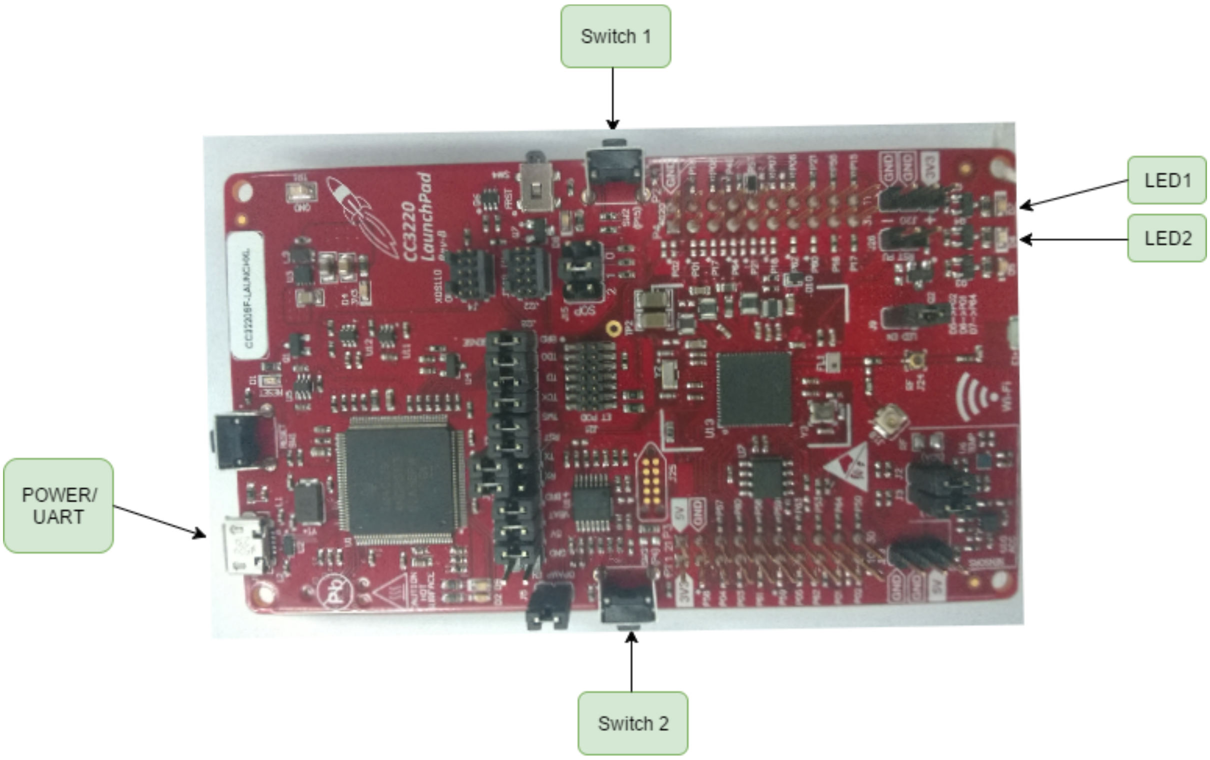


Figure 1: TI Launchpad Front

Chapter 2. Note on DevKit Credentials and Environment

At various times in this guide, we will refer to "device credentials" and "user account credentials".
Arrayent distributes this information in various ways depending upon the context of our engagement with you, so please read the section below that applies to you so that you know where to get this information.

1. Users Who Receive Explicit Credentials from Arrayent

If you have received explicit instructions from Arrayent that lists out your user and device credentials, then you should use those values whenever this guide prompts you for that information. We typically distribute this information in a dedicated document, with a title like "DevKit Account Information", or you may have received it via an automated email after filling out a form.

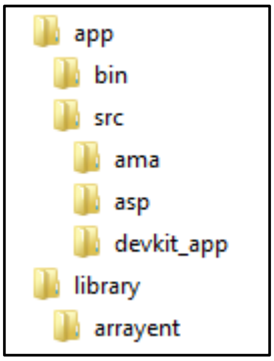
Chapter 3. Sample application Setup

In this section you will learn how to:

- Load the pre-built binary of the sample application
- Setup the SDK, build and load the sample application source code
- Setup WiFi router credentials and ACA configuration

1. Release package Directory Structure

Following is the release package directory structure for all platforms:



```
app
  bin
  src
    ama
    asp
    devkit_app
library
  arrayent
```

- app - sample application and related libraries.
  - app -> bin - Sample application binary
  - app -> src - Sample application and related libraries.
    - app -> src -> devkit_app - Sample application source code.
    - app -> src -> ama - Arrayent multi-attribute helper module
    - app -> src -> asp - Arrayent application support package provides abstraction over different platforms
- library->arrayent - ACA library

2. Steps to load the pre-built devkit application image

TI CC3220 Launchpad:

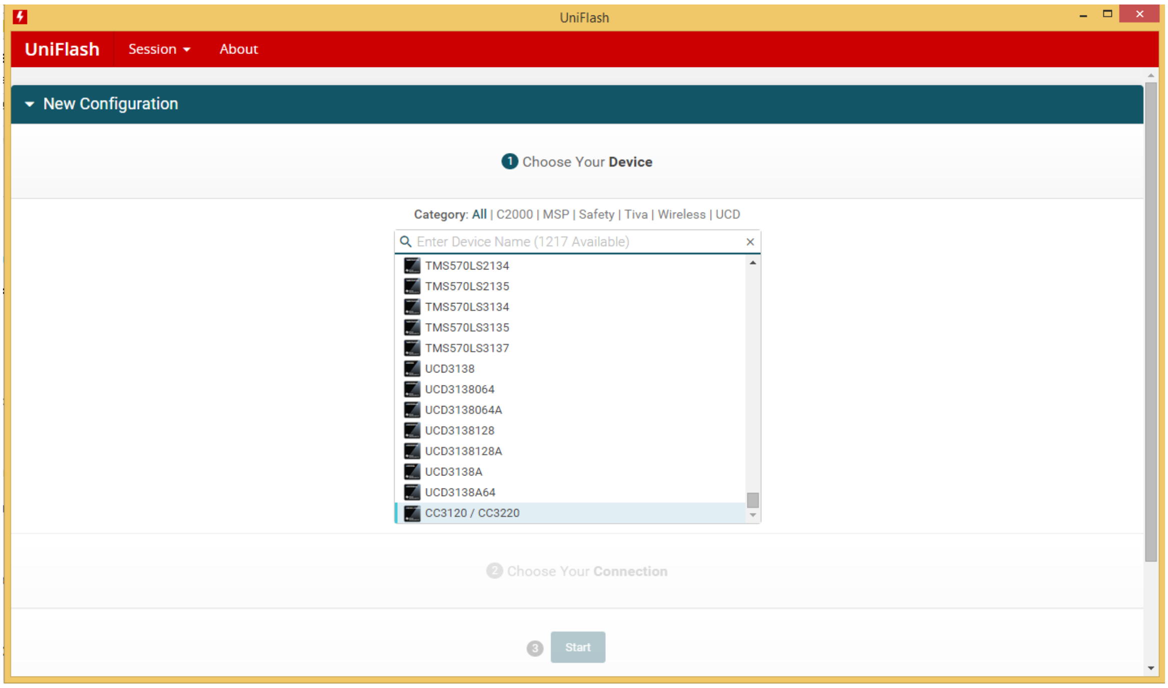
a. Connect the board to the computer using mini USB cable. Wait a minute to let the device drivers install.

b. If you click on the Installing device driver software popup, you can see the status of the device driver installation.

c. Follow the steps below to flash the application and store web pages into flash using uniflash:

   i. Open UniFlash 4.0

   ii. Select “CC3120/CC3220” device from “Enter Device Name” bar.



Choose Your Device

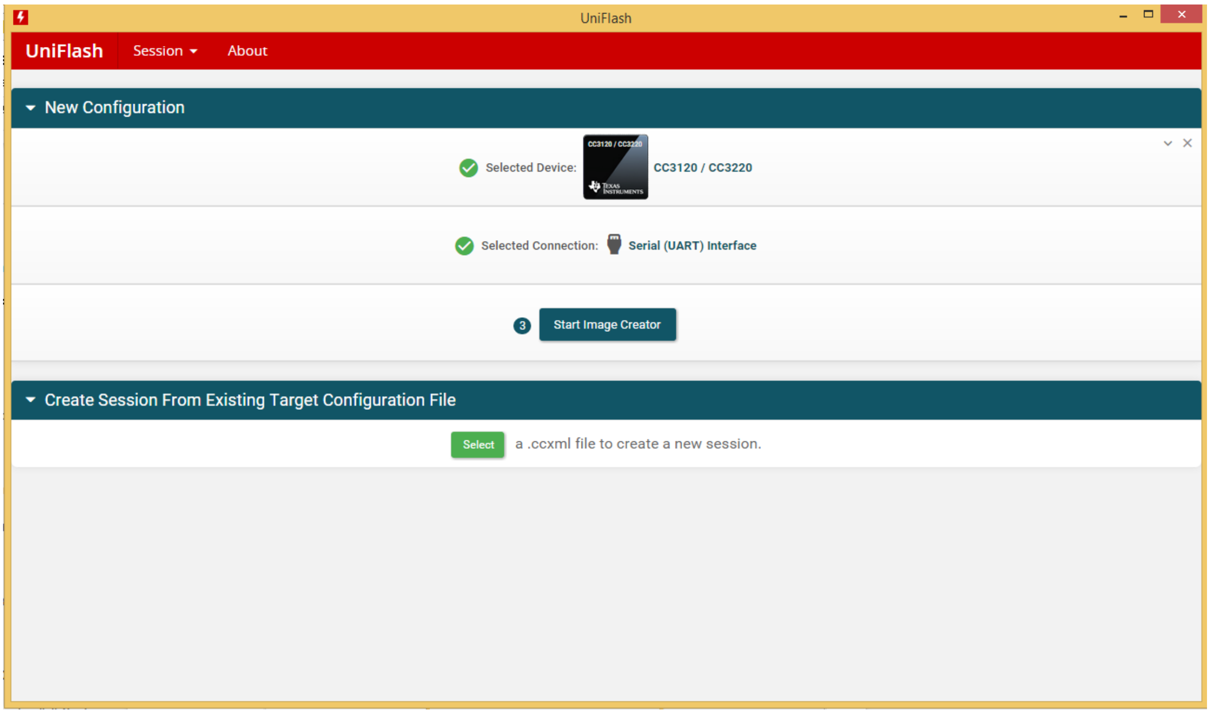
Category: All | C2000 | MSP | Safety | Tiva | Wireless | UCD

Enter Device Name (1217 Available)

- TMS570LS2134
- TMS570LS2135
- TMS570LS3134
- TMS570LS3135
- TMS570LS3137
- UCD3138
- UCD3138064
- UCD3138064A
- UCD3138128
- UCD3138128A
- UCD3138A
- UCD3138A64
- CC3120 / CC3220

2 Choose Your Connection

iii. Verify “Serial (UART) Interface” is selected as a connection.

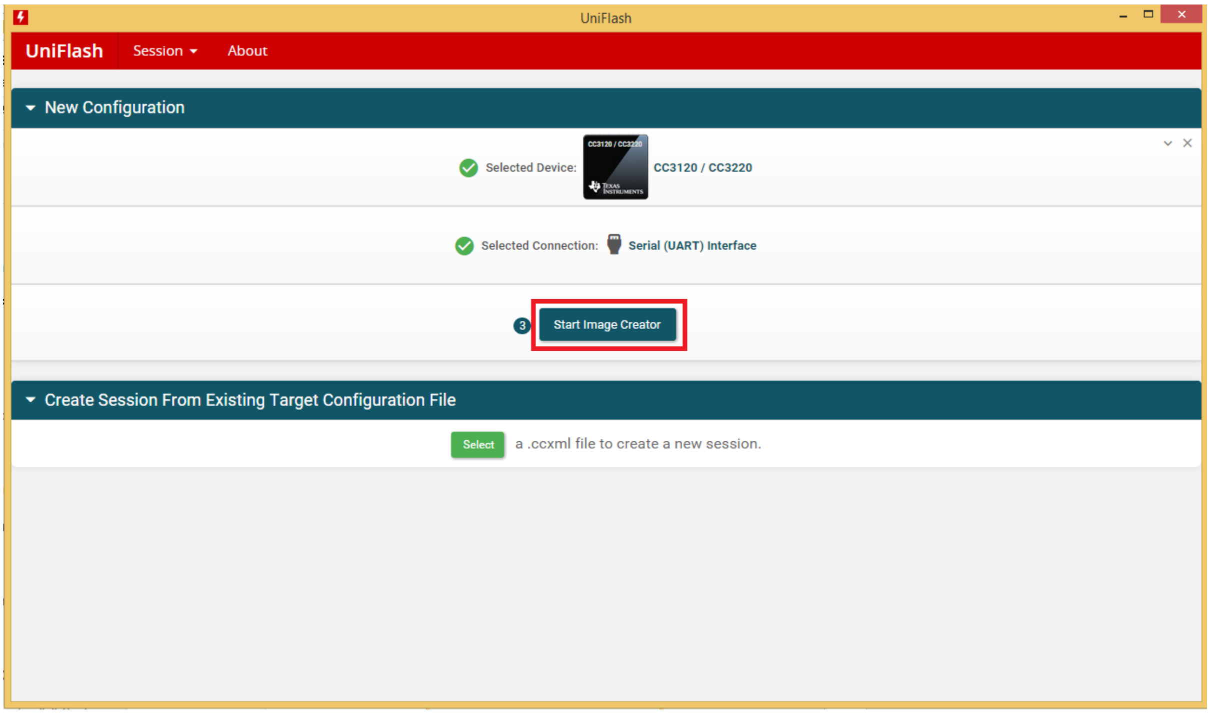


Selected Device: CC3120 / CC3220

Selected Connection: Serial (UART) Interface

Select a .ccxml file to create a new session.

iv. Click on “Start Image Creator” button



UniFlash

Session ▾ About

▾ New Configuration

✓ Selected Device: CC3120 / CC3220

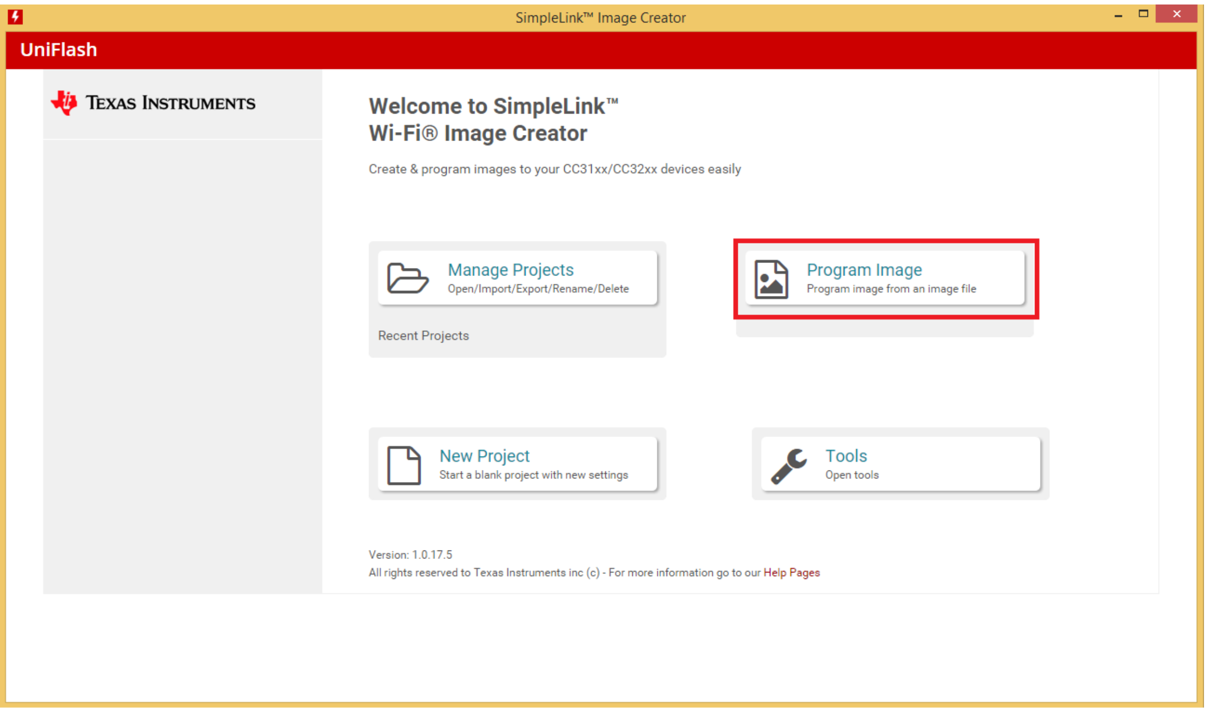
✓ Selected Connection: Serial (UART) Interface

3 Start Image Creator

▾ Create Session From Existing Target Configuration File

Select a .ccxml file to create a new session.

v. Click on “Program Image Button”



SimpleLink™ Image Creator

UniFlash

Texas Instruments

Welcome to SimpleLink™
Wi-Fi® Image Creator

Create & program images to your CC31xx/CC32xx devices easily

Manage Projects
Open/Import/Export/Rename/Delete

Recent Projects

Program Image
Program image from an image file

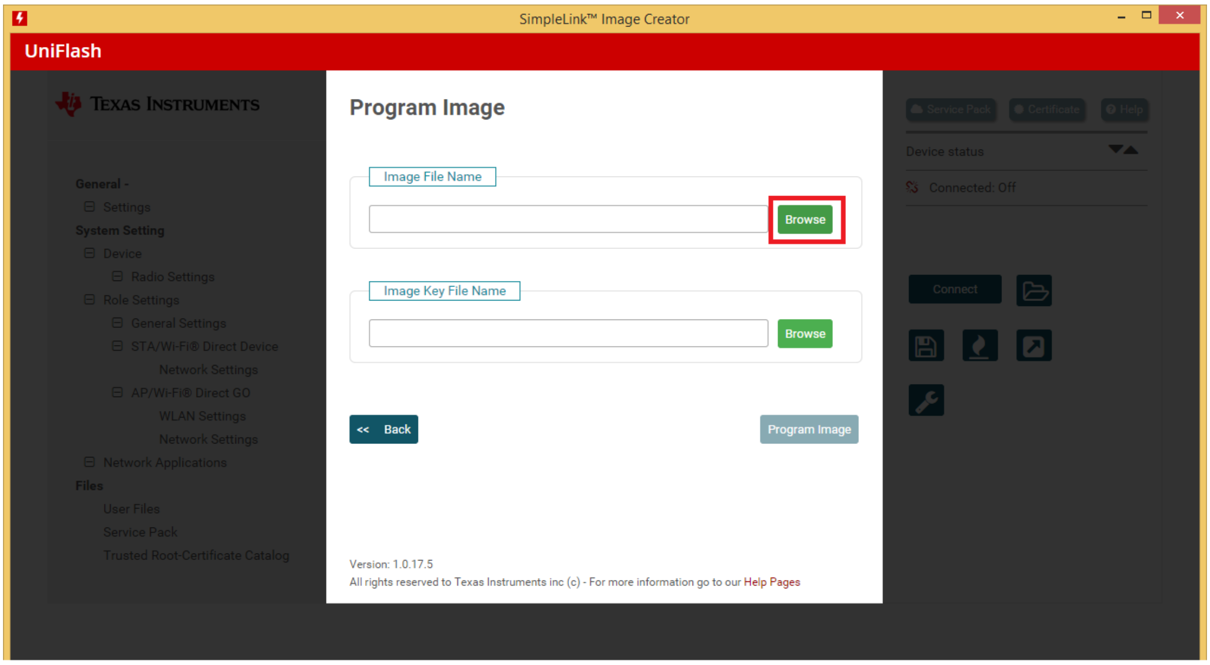
New Project
Start a blank project with new settings

Tools
Open tools

Version: 1.0.17.5

All rights reserved to Texas Instruments inc (c) - For more information go to our Help Pages

vi. Click on Browse button in “Image File Name” Section and select devkit_app_Programming.sli file located at “<*PATH TO DEVKIT APP BUNDLE*>\app\bin”



Version: 1.0.17.5

All rights reserved to Texas Instruments inc (c) - For more information go to our Help Pages

vii. Click on “Program Image” button

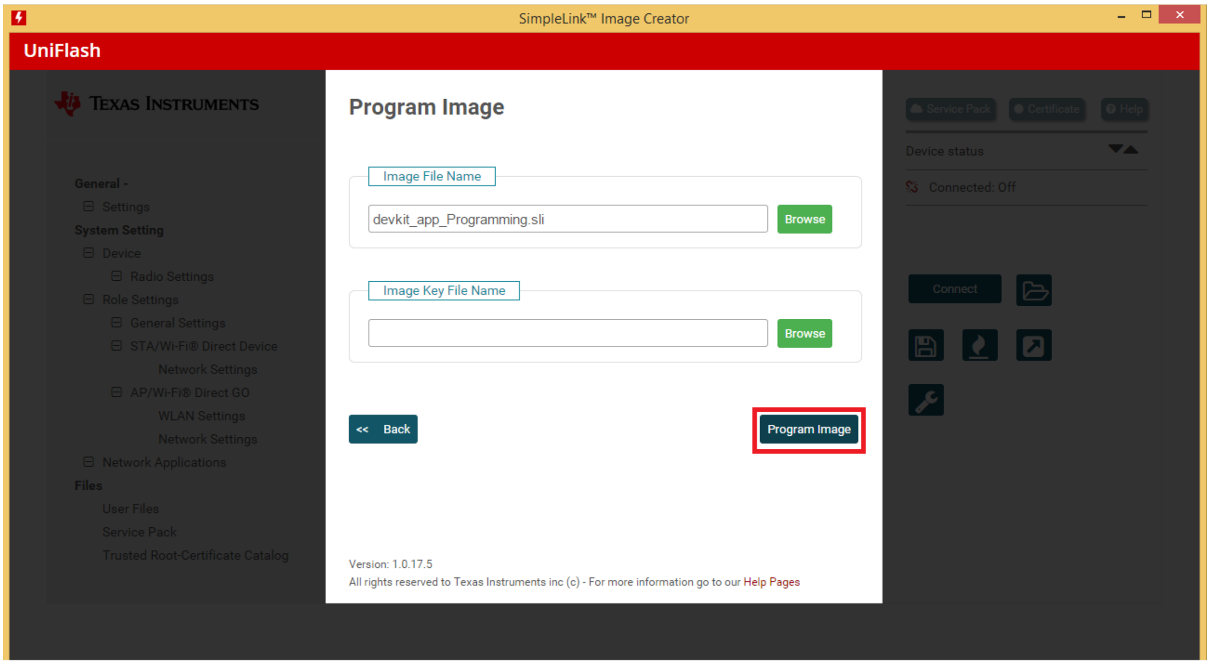


Image File Name: devkit_app_Programming.sli

Version: 1.0.17.5

All rights reserved to Texas Instruments inc (c) - For more information go to our Help Pages

viii. On successful completion, It prints “Programming complete” message

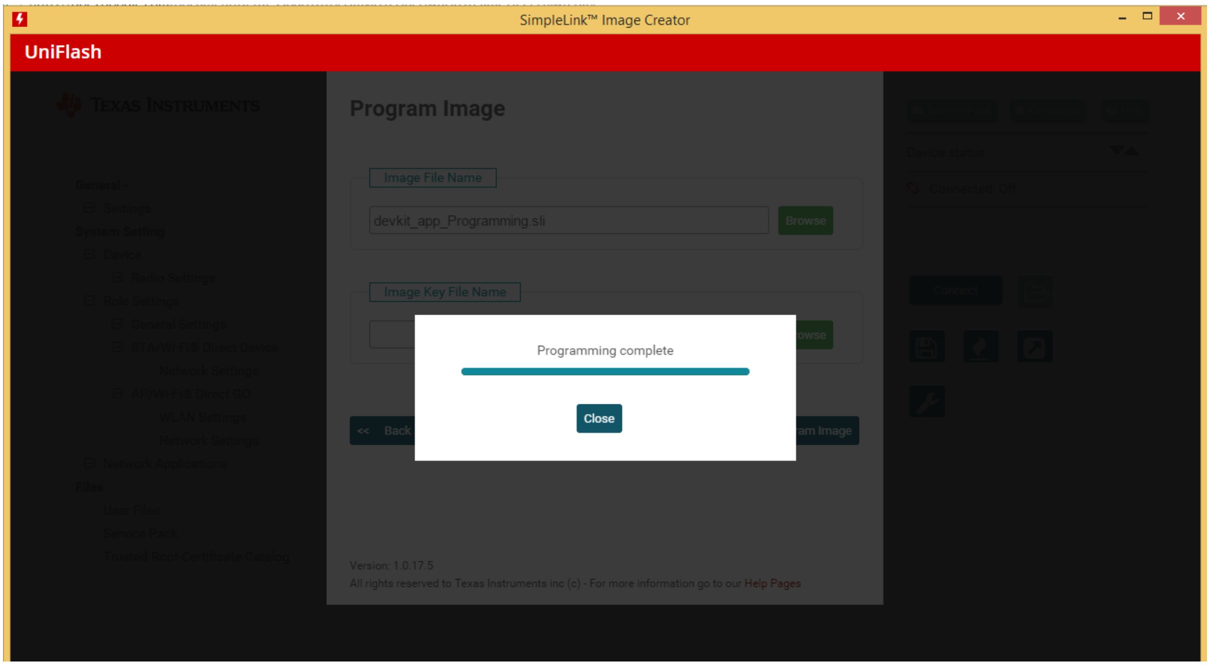


Image File Name: devkit_app_Programming sli

Programming complete

Version: 1.0.17.5

All rights reserved to Texas Instruments inc (c) - For more information go to our Help Pages

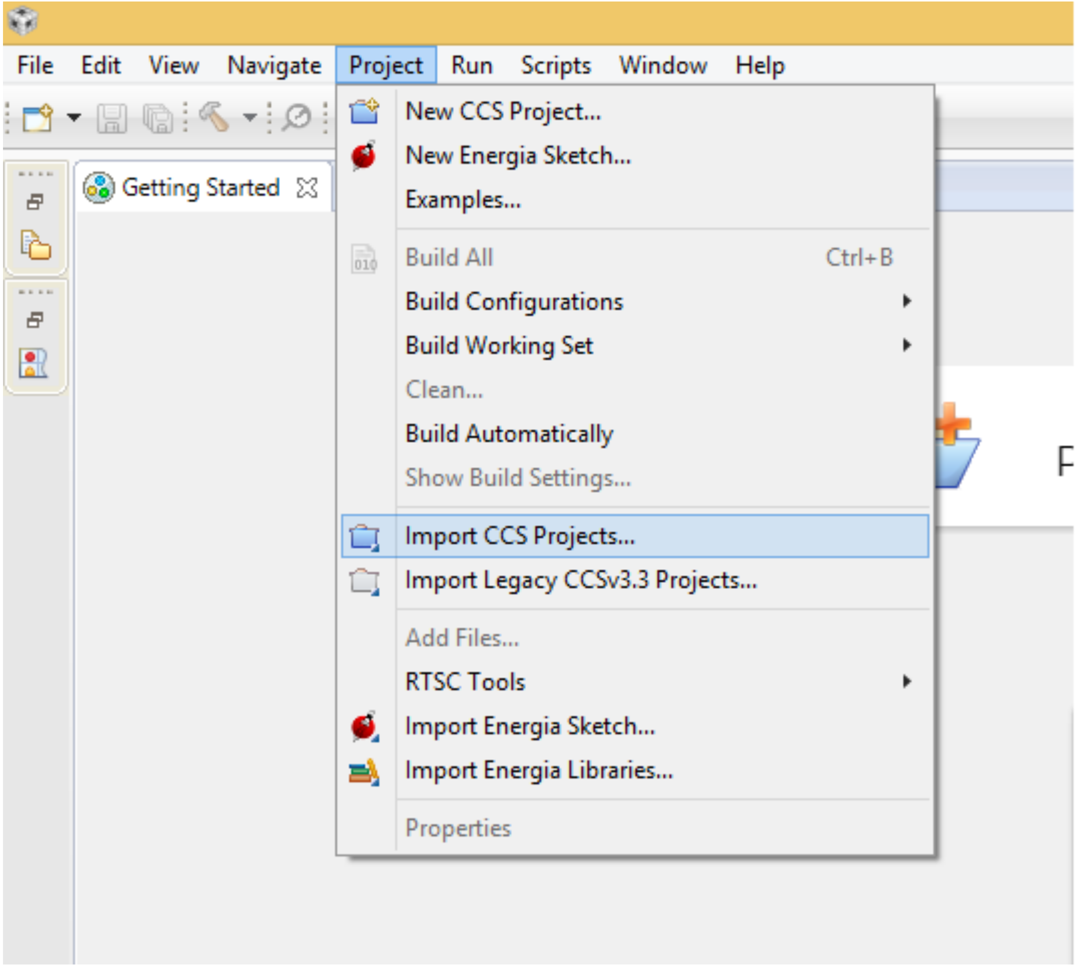
ix. Reset the development board by pressing RESET push button. Demo application will be running once the board is rebooted. You may now try out the internet connectivity by skipping to section 4.

3. Setup development environment

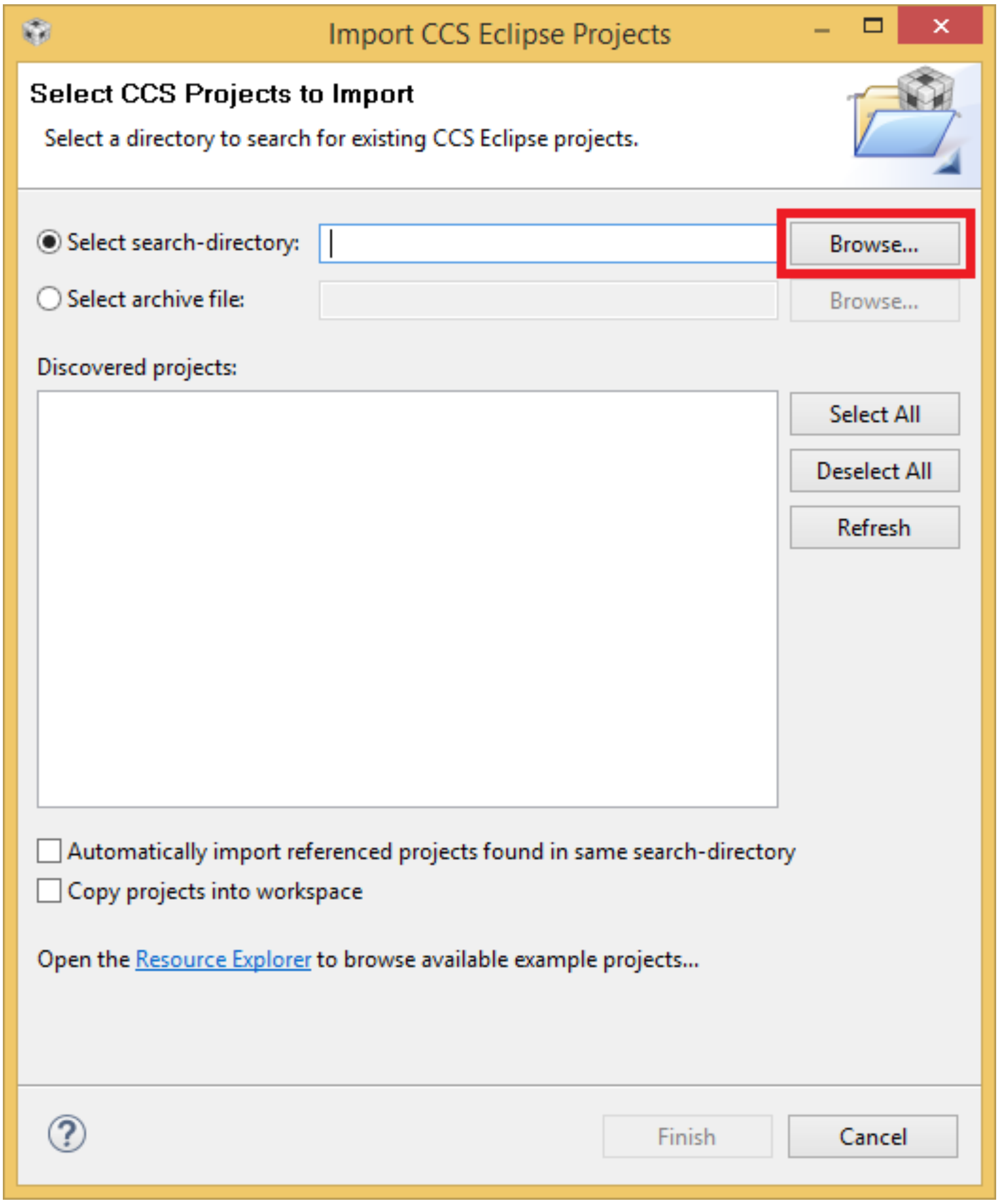
Use this section if you want to try modifying the sample application. You may skip this section and continue on section 4 if you simply want to try out the pre-packaged sample application.

TI CC3220 Launchpad:

a. Follow steps a-b of the previous section to install the drivers for the TI CC3220 development board.
b. Please follow "C:/TI/simplelink_cc32xx_sdk_1_02_02_00/docs/simplelink_mcu_sdk/Quick_Start_Guide.html" document to build sdk and setup CCS for TI CC3220 development.
c. Copy the "<*PATH TO DEVKIT APP BUNDLE*>\library\arrayent" folder to <*PATH TO TI SDK*>\library directory. If you have the debug version of the libaca package, rename the library from libaca_ti_debug.lib to libaca_ti.lib.
d. Open CCS v7.0
e. Select "Project->Import CCS Projects.." in CCS IDE.



f. Click on Browse button and select "C:\TI\simplelink_cc32xx_sdk_1_02_02_00\kernel\tirtos\builds\CC3220SF_LAUNCHXL\release\ccs" folder.



Import CCS Eclipse Projects

Select CCS Projects to Import

Select a directory to search for existing CCS Eclipse projects.

● Select search-directory:

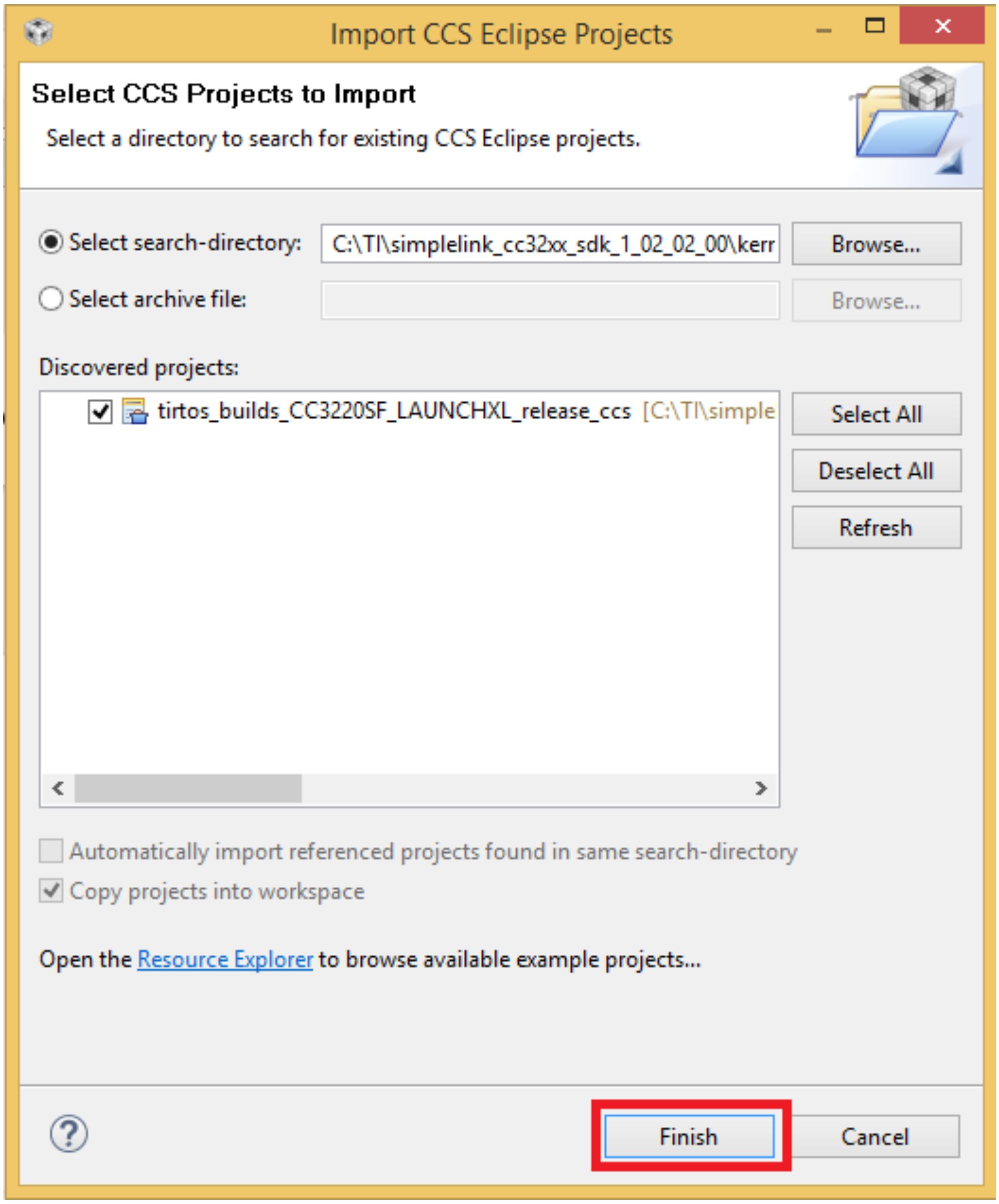
○ Select archive file:

□ Automatically import referenced projects found in same search-directory

□ Copy projects into workspace

Open the Resource Explorer to browse available example projects...

g. Click on Finish button to complete Import process.



Import CCS Eclipse Projects

Select CCS Projects to Import

Select a directory to search for existing CCS Eclipse projects.

● Select search-directory: C:\TI\simplelink_cc32xx_sdk_1_02_02_00\kerr

○ Select archive file:

Discovered projects:

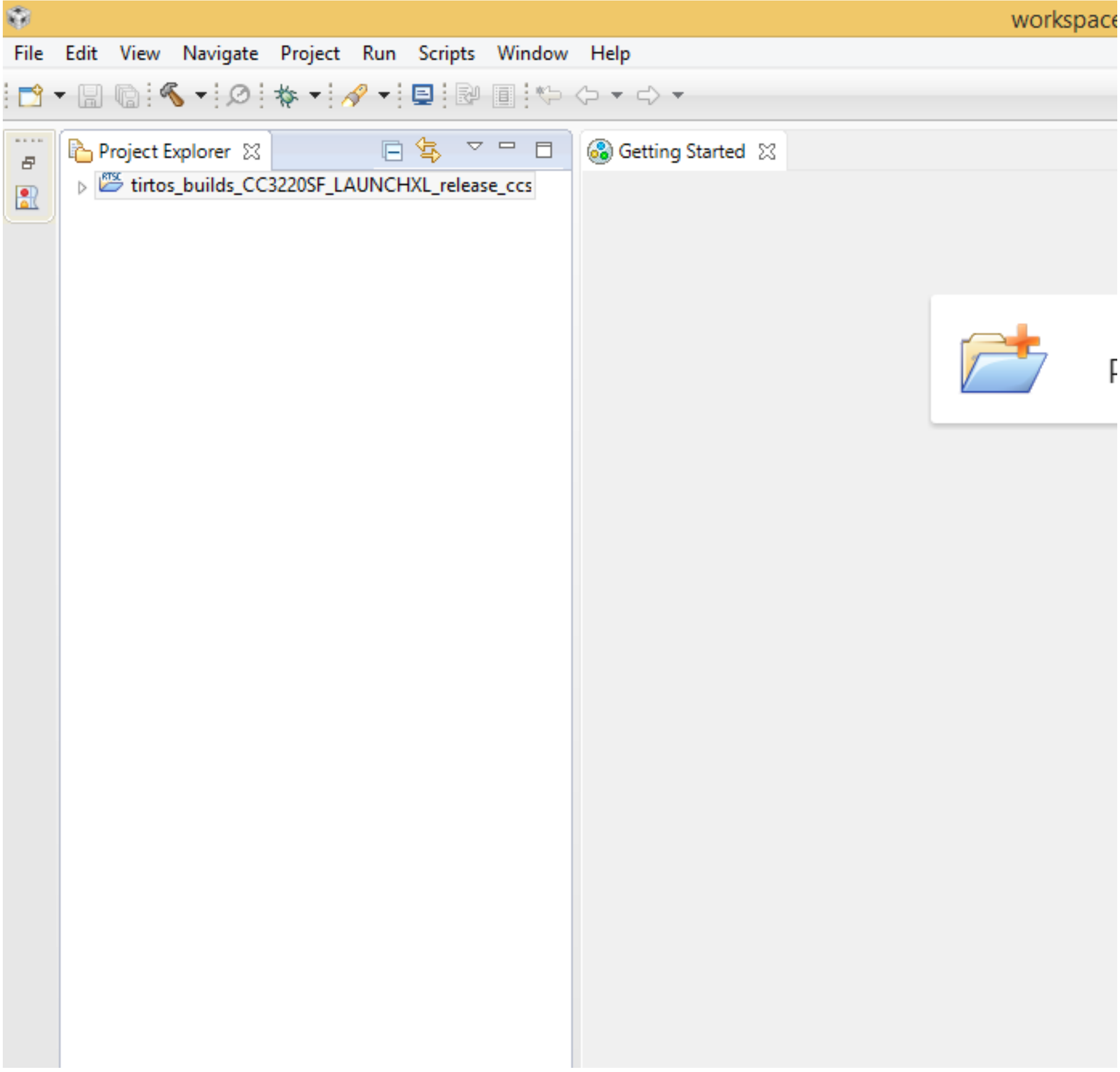
☑ tirtos_builds_CC3220SF_LAUNCHXL_release_ccs [C:\TI\simple

□ Automatically import referenced projects found in same search-directory

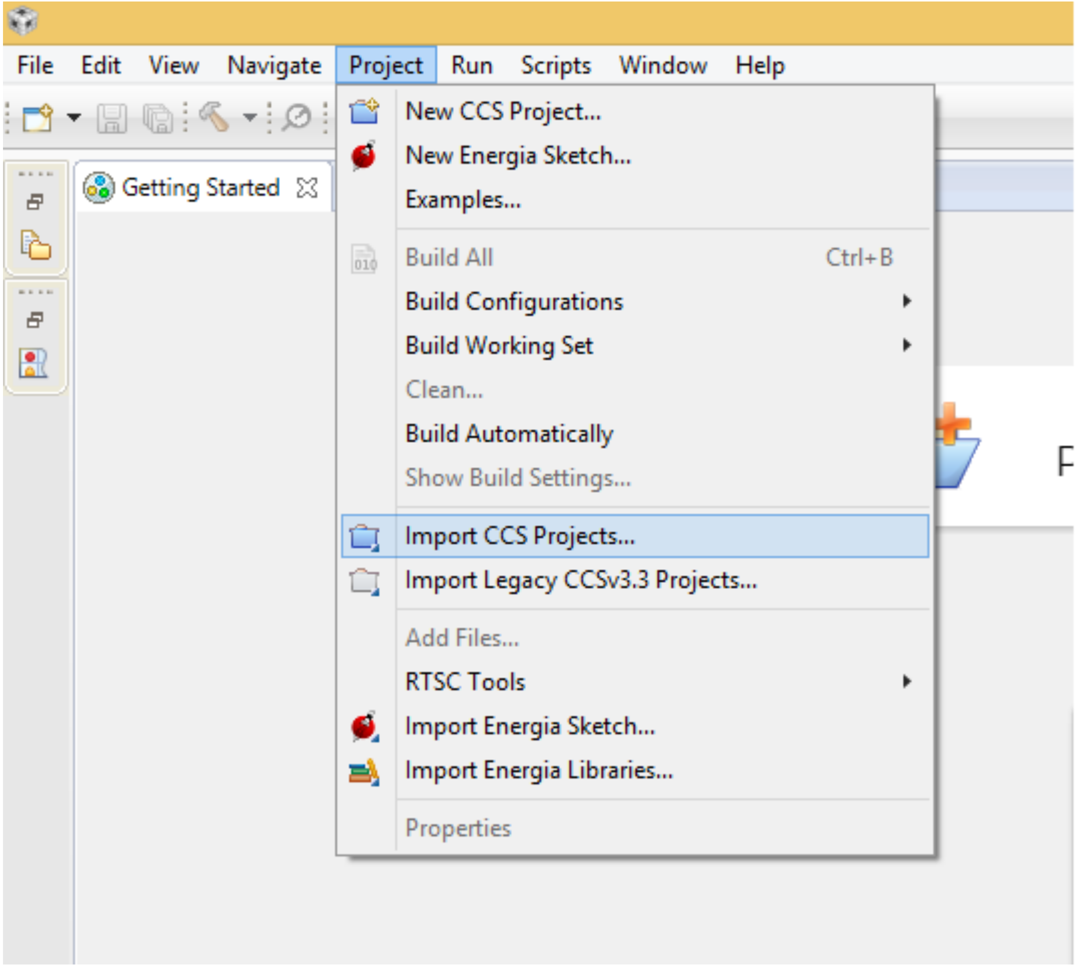
☑ Copy projects into workspace

Open the Resource Explorer to browse available example projects...

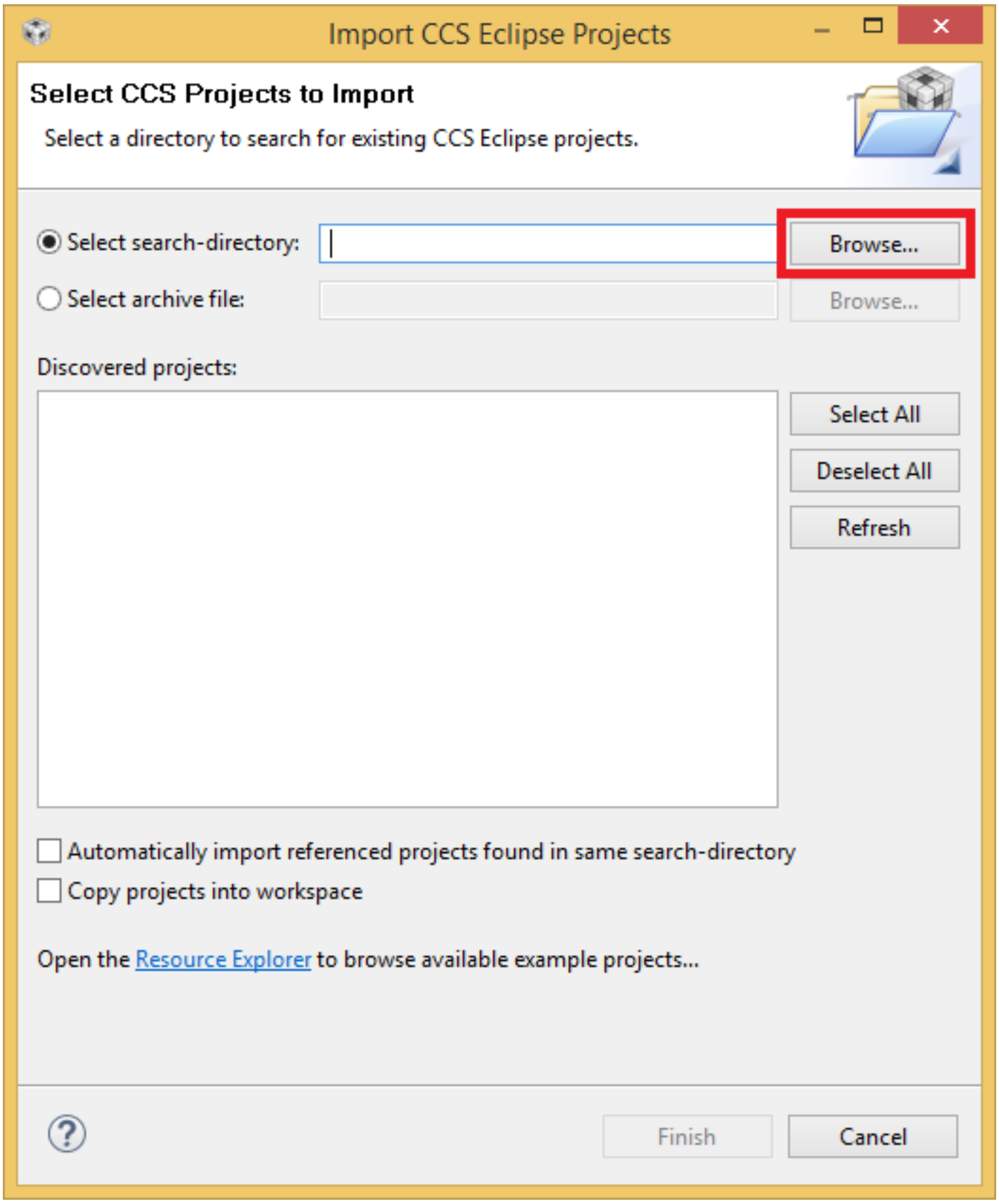
h. On successful tirtos ccs project import, project will appear in project Explorer section


workspace
File Edit View Navigate Project Run Scripts Window Help
Project Explorer
tirtos_builds_CC3220SF_LAUNCHXL_release_ccs
Getting Started

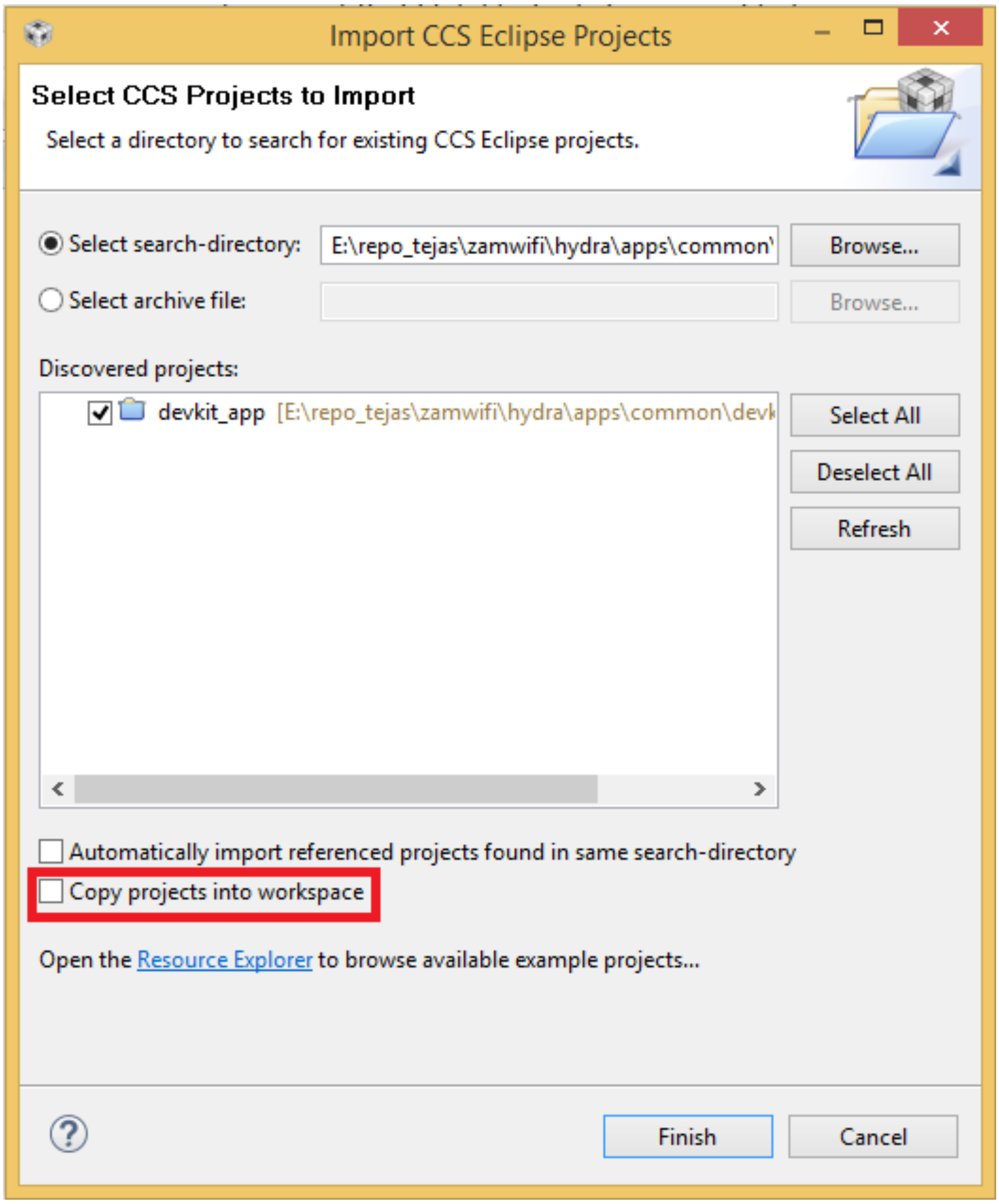
i. Select “Project->Import CCS Projects..” in CCS IDE.



j. Click on Browse button and select "*<PATH TO DEVKIT APP BUNDLE*>\app\src\devkit_app\cc3220\CCS" folder.


Import CCS Eclipse Projects
Select CCS Projects to Import
Select a directory to search for existing CCS Eclipse projects.
● Select search-directory:
Browse...
○ Select archive file:
Browse...
Discovered projects:
Select All
Deselect All
Refresh
□ Automatically import referenced projects found in same search-directory
□ Copy projects into workspace
Open the Resource Explorer to browse available example projects...
Finish
Cancel

k. Please make sure “Copy projects into workspace” is unchecked. Click on Finish button to complete Import process.



Import CCS Eclipse Projects

Select CCS Projects to Import

Select a directory to search for existing CCS Eclipse projects.

● Select search-directory: E:\repo_tejas\zamwifi\hydra\apps\common\

○ Select archive file:

Discovered projects:

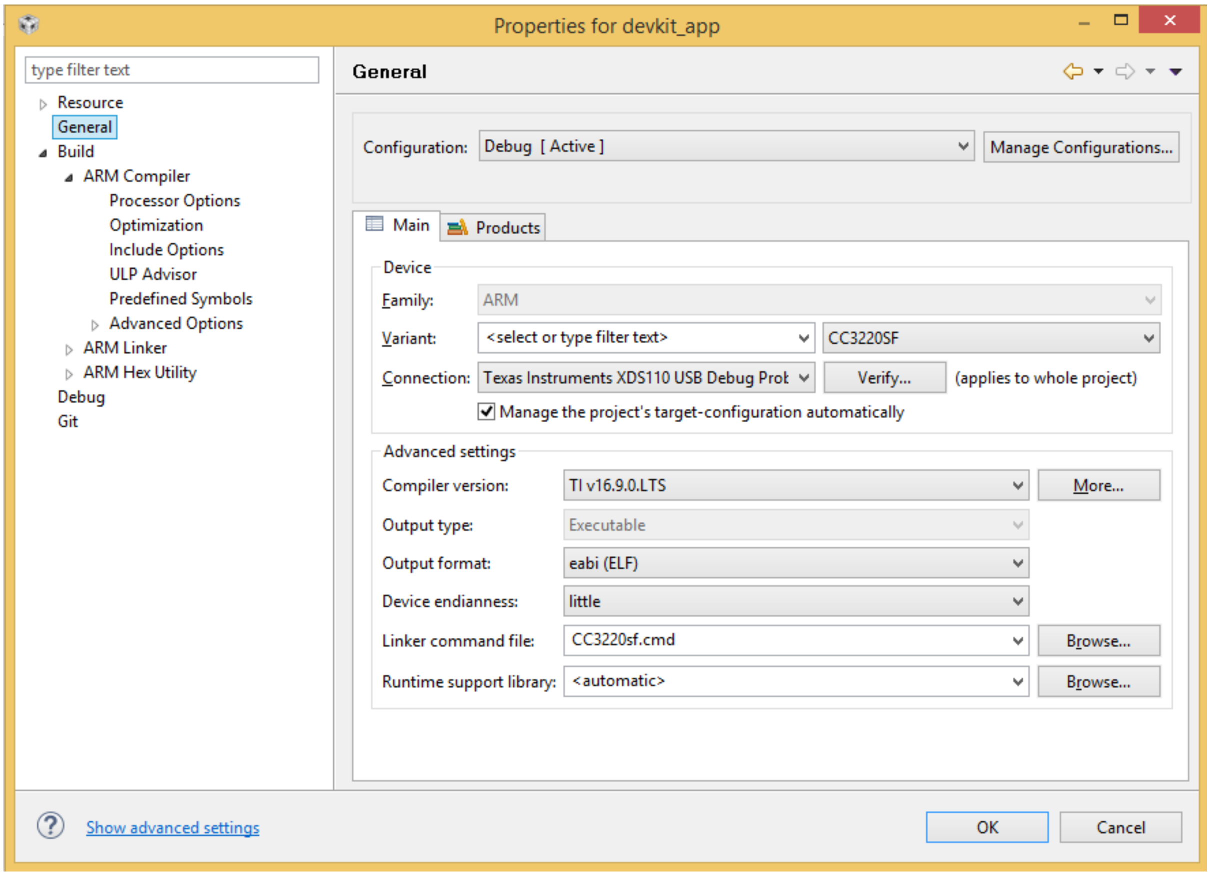
☑ devkit_app [E:\repo_tejas\zamwifi\hydra\apps\common\devk

□ Automatically import referenced projects found in same search-directory

□ Copy projects into workspace

Open the Resource Explorer to browse available example projects...

I. Select “devkit_app” project in Project Explorer window and press Alt+Enter to open properties of devkit_app project.



Configuration: Debug [Active]

Family: ARM

Variant: <select or type filter text> CC3220SF

Connection: Texas Instruments XDS110 USB Debug Prob (applies to whole project)

☑ Manage the project's target-configuration automatically

Compiler version: TI v16.9.0.LTS

Output type: Executable

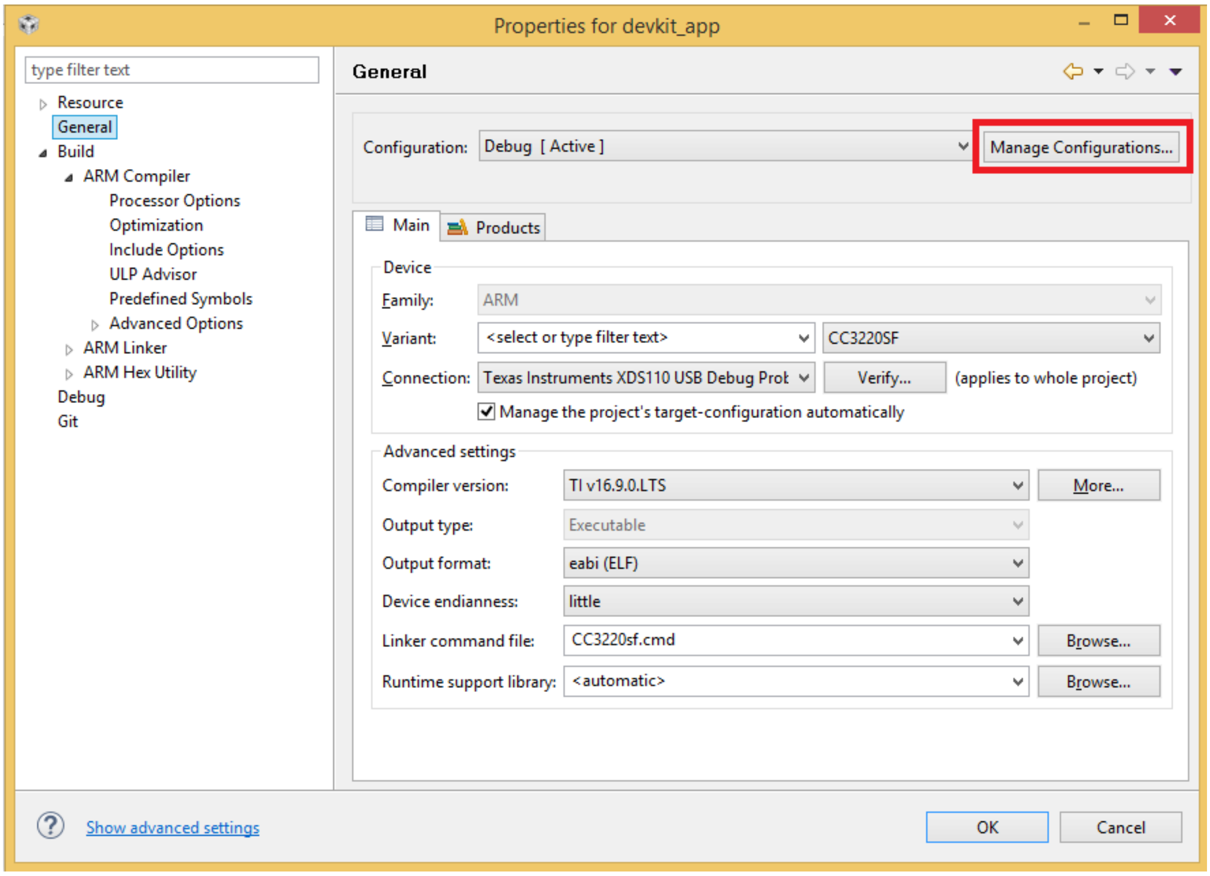
Output format: eabi (ELF)

Device endianness: little

Linker command file: CC3220sf.cmd

Runtime support library: <automatic>

m. Click on “Manage Configuration” button to select project configuration



Configuration: Debug [Active]

Family: ARM

Variant: <select or type filter text> CC3220SF

Connection: Texas Instruments XDS110 USB Debug Prob (applies to whole project)

☑ Manage the project's target-configuration automatically

Compiler version: TI v16.9.0.LTS

Output type: Executable

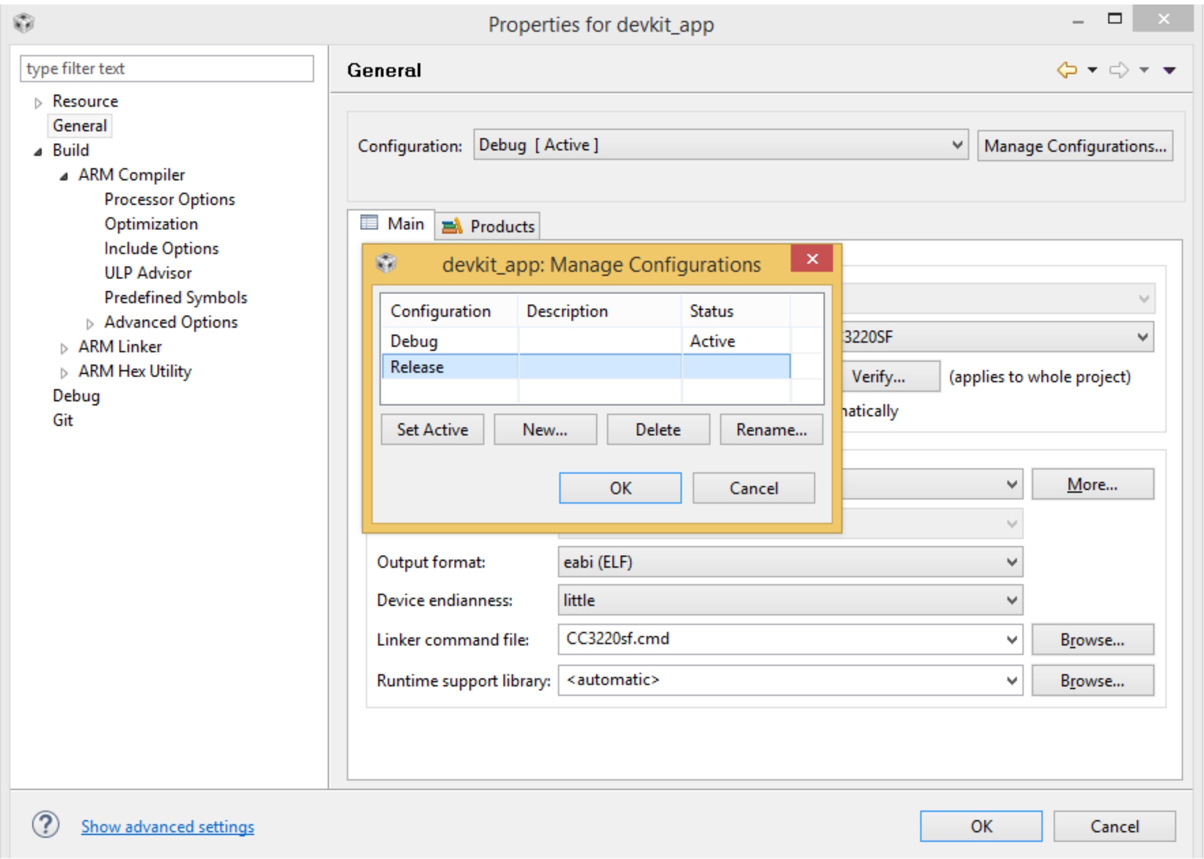
Output format: eabi (ELF)

Device endianness: little

Linker command file: CC3220sf.cmd

Runtime support library: <automatic>

n. Select “Debug” or “Release” configuration and click on “Set Active” button and click on OK button to save configuration.



Configuration: Debug [Active]

Configuration	Description	Status
Debug		Active
Release		

3220SF

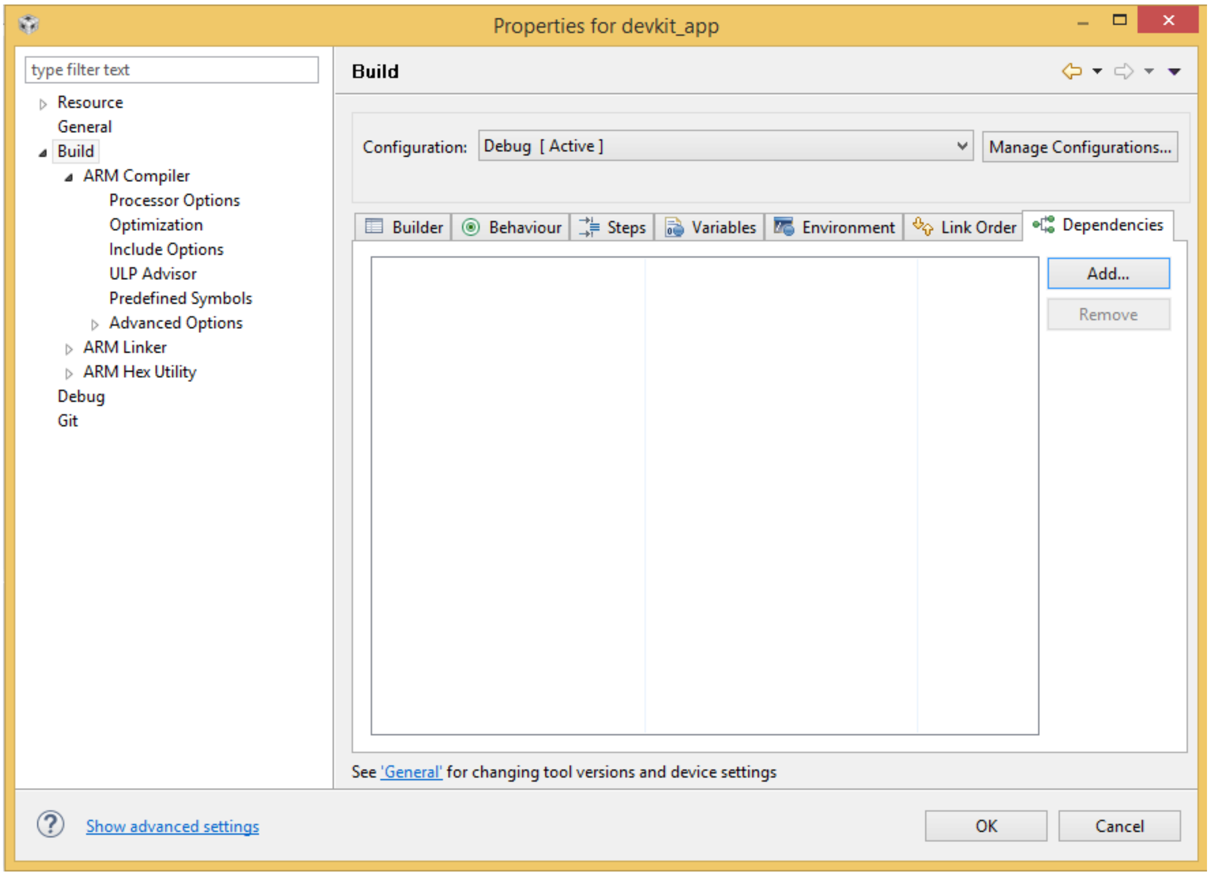
Output format: eabi (ELF)

Device endianness: little

Linker command file: CC3220sf.cmd

Runtime support library: <automatic>

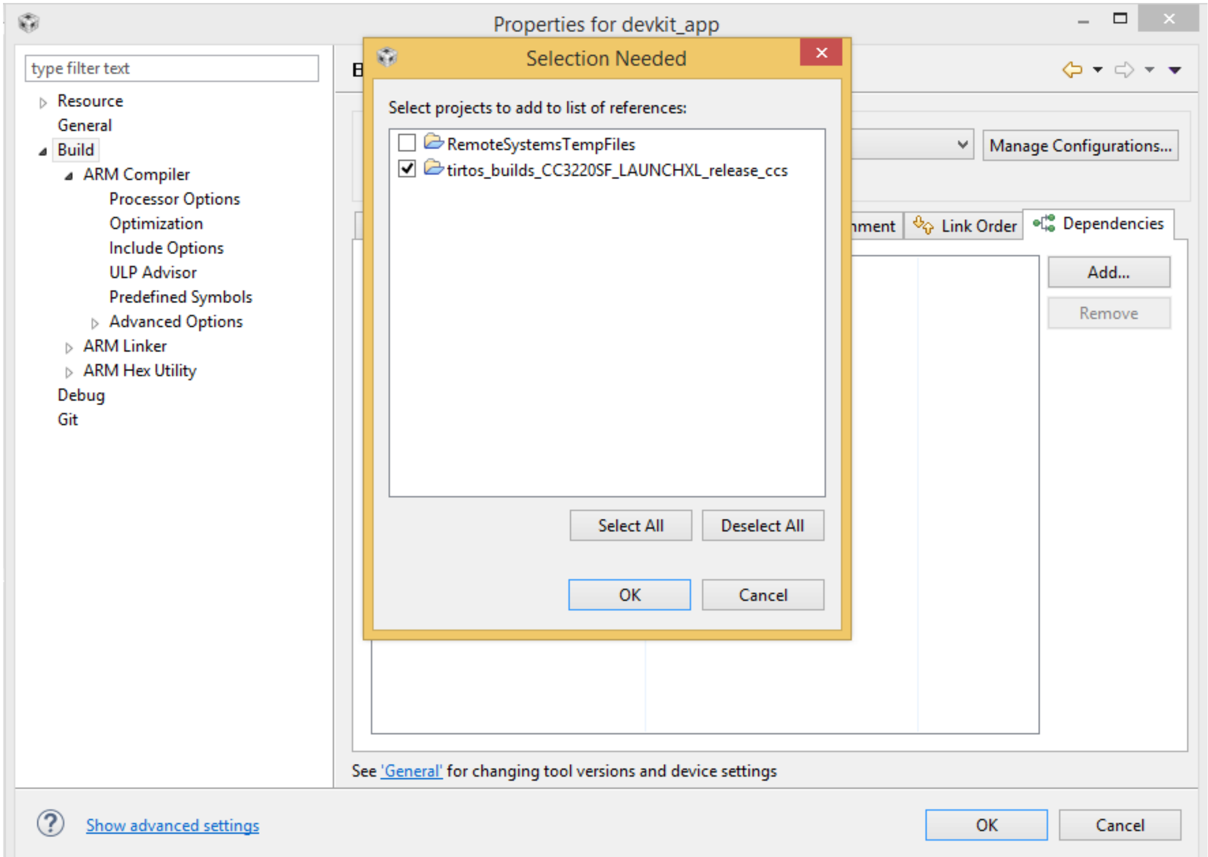
o. Select Build in left side of properties window. Select Dependencies tab and click on Add button.



Configuration: Debug [Active]

See 'General' for changing tool versions and device settings

p. Select “tirtos_builds_CC3220SF_LAUNCHXL_release_ccs” project and click on OK button.



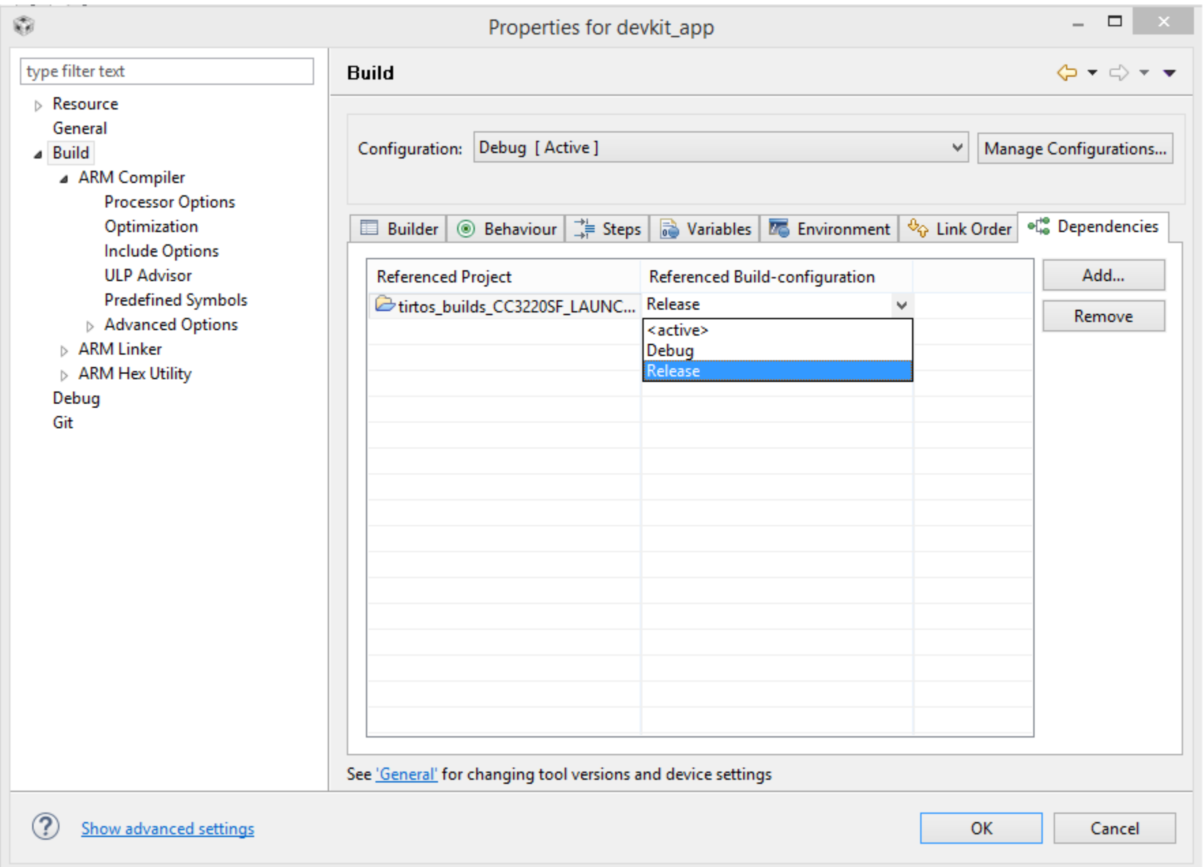
Selection Needed

Select projects to add to list of references:

□ RemoteSystemsTempFiles
☑ tirtos_builds_CC3220SF_LAUNCHXL_release_ccs

See 'General' for changing tool versions and device settings

q. Select Referenced Build-configuration as “Release” as shown in below screenshot:



Build

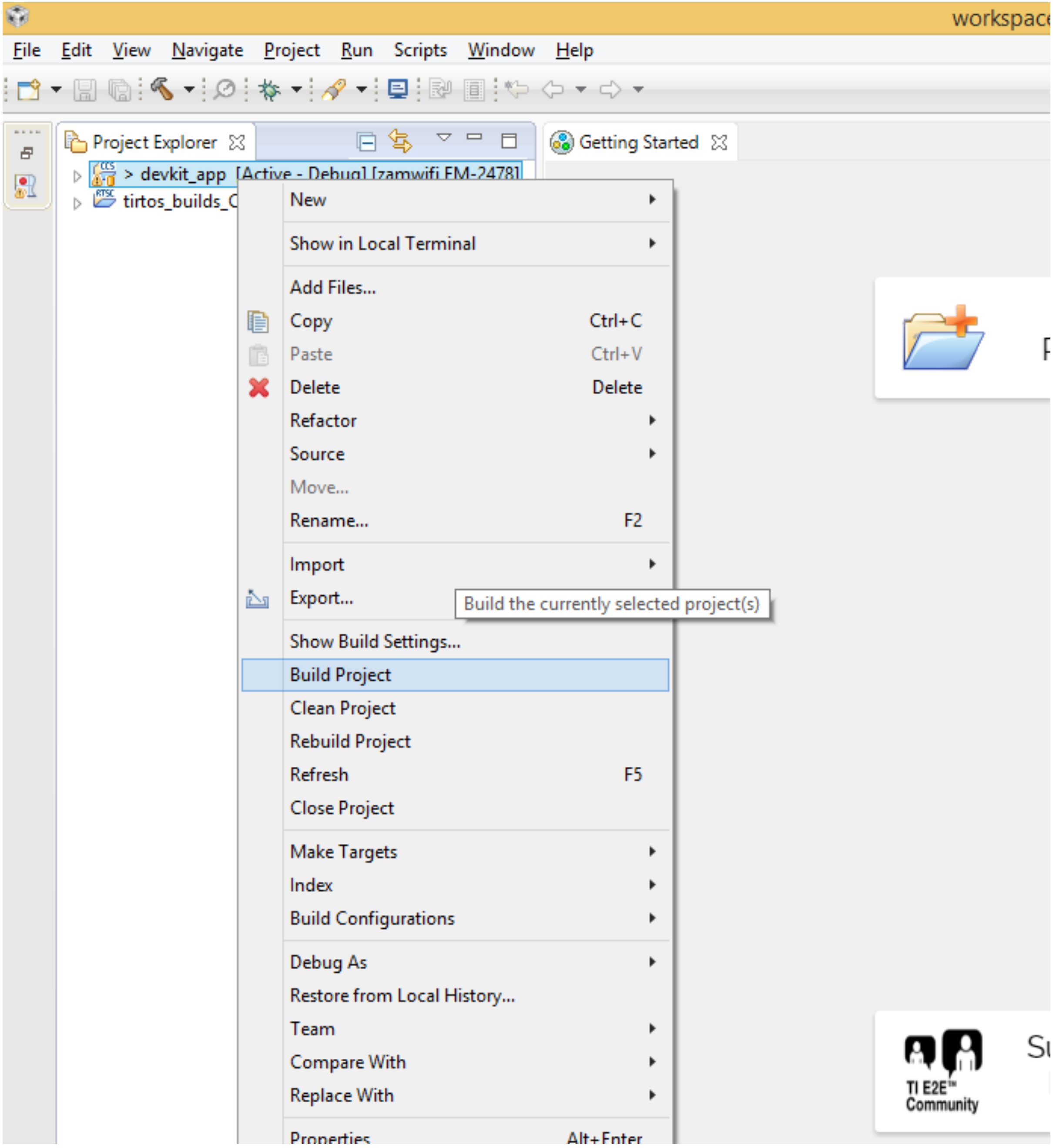
Configuration: Debug [Active]

Referenced Project	Referenced Build-configuration
tirtos_builds_CC3220SF_LAUNC...	Release

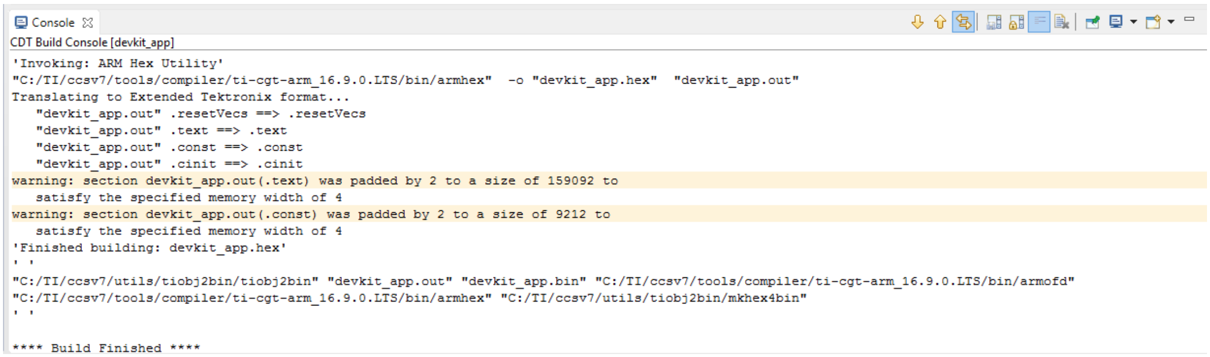
See 'General' for changing tool versions and device settings

r. Click on OK button to save the properties of devkit_app project.

s. Right click on “devkit_app” project in Project Explorer window and select “Build Project” to build Devkit application.



t. On successful devkit application build “Build Finished” message should appear in Console.



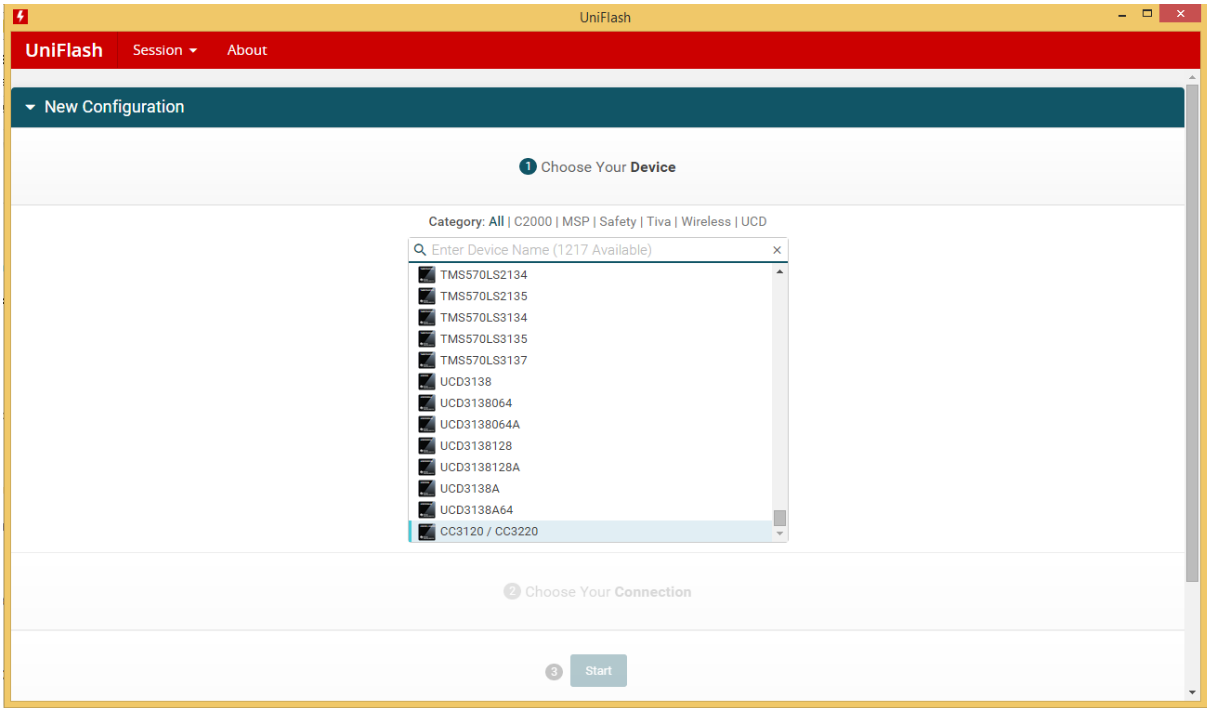
```
'Invoking: ARM Hex Utility'
"C:/TI/ccsv7/tools/compiler/ti-cgt-arm_16.9.0.LTS/bin/armhex"  -o "devkit_app.hex"  "devkit_app.out"
Translating to Extended Tektronix format...
   "devkit_app.out" .resetVecs ==> .resetVecs
   "devkit_app.out" .text ==> .text
   "devkit_app.out" .const ==> .const
   "devkit_app.out" .cinit ==> .cinit
warning: section devkit_app.out(.text) was padded by 2 to a size of 159092 to
   satisfy the specified memory width of 4
warning: section devkit_app.out(.const) was padded by 2 to a size of 9212 to
   satisfy the specified memory width of 4
'Finished building: devkit_app.hex'
' '
"C:/TI/ccsv7/utils/tiobj2bin/tiobj2bin" "devkit_app.out" "devkit_app.bin" "C:/TI/ccsv7/tools/compiler/ti-cgt-arm_16.9.0.LTS/bin/armofd" "C:/TI/ccsv7/tools/compiler/ti-cgt-arm_16.9.0.LTS/bin/armhex" "C:/TI/ccsv7/utils/tiobj2bin/mkhex4bin"
' '

**** Build Finished ****
```

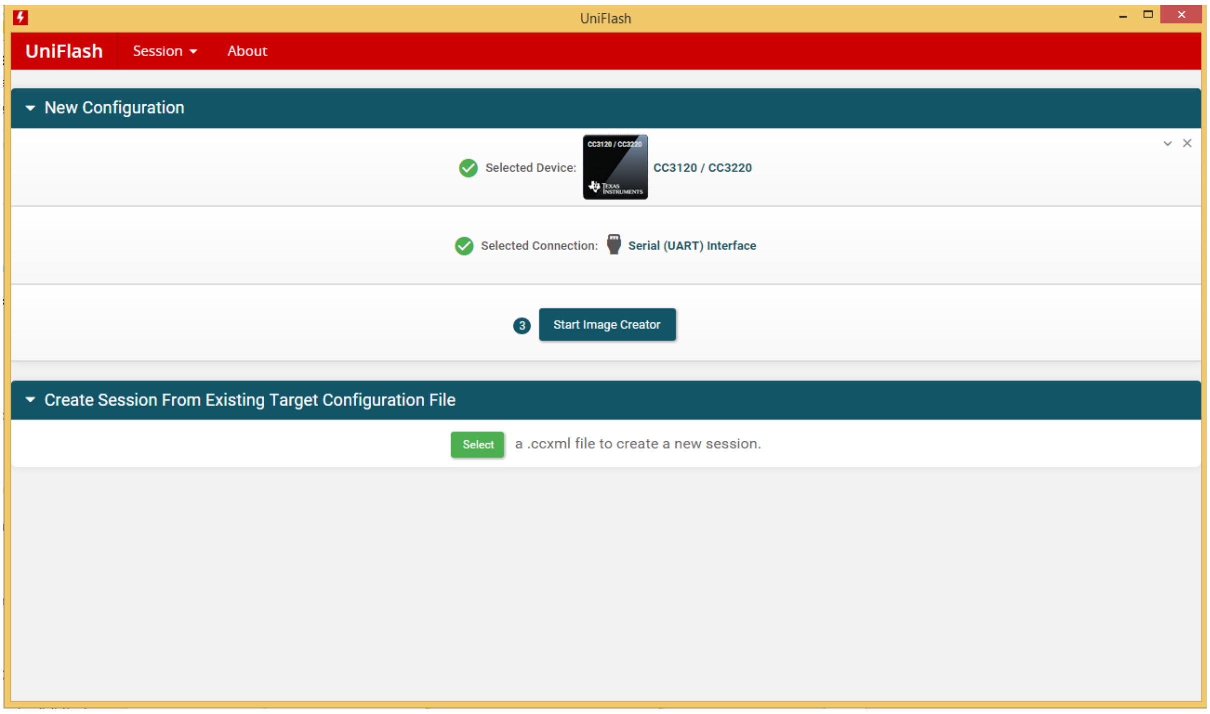
4. Create Image Using UniFlash

a. Open on Uniflash v4.0

b. Select “CC3120/CC3220” device from “Enter Device Name” bar.



c. Verify “Serial (UART) Interface” is selected as a connection.

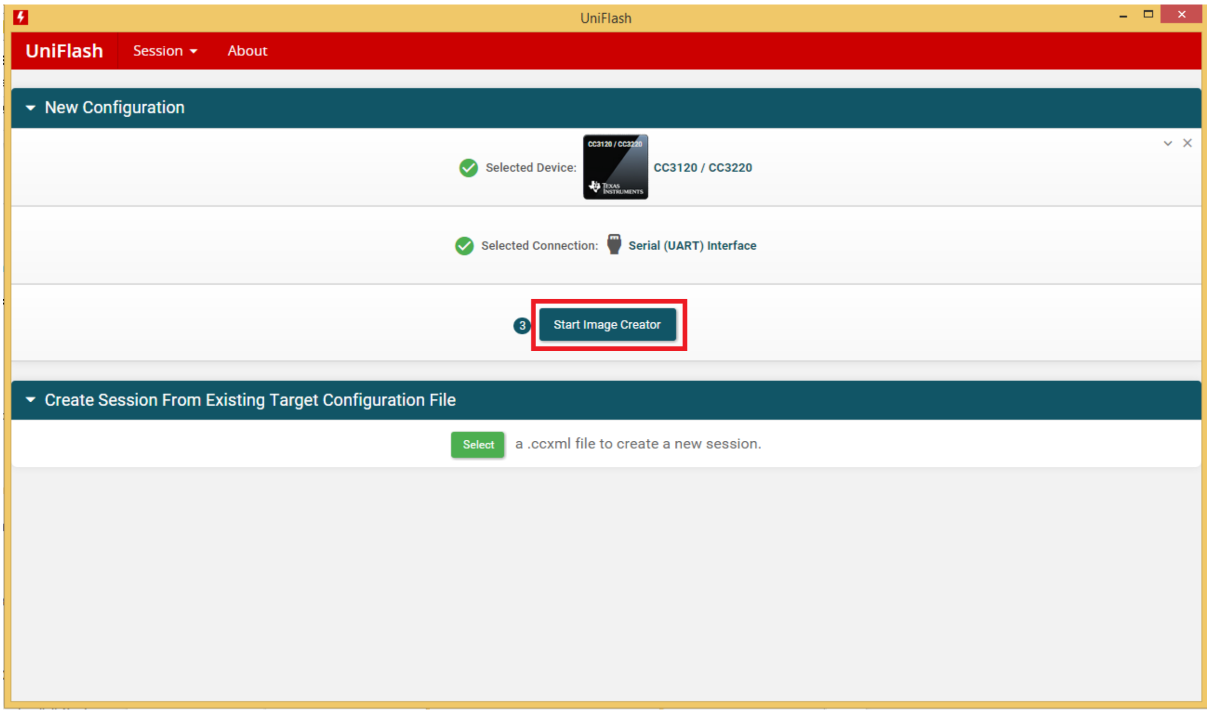


Selected Device: CC3120 / CC3220

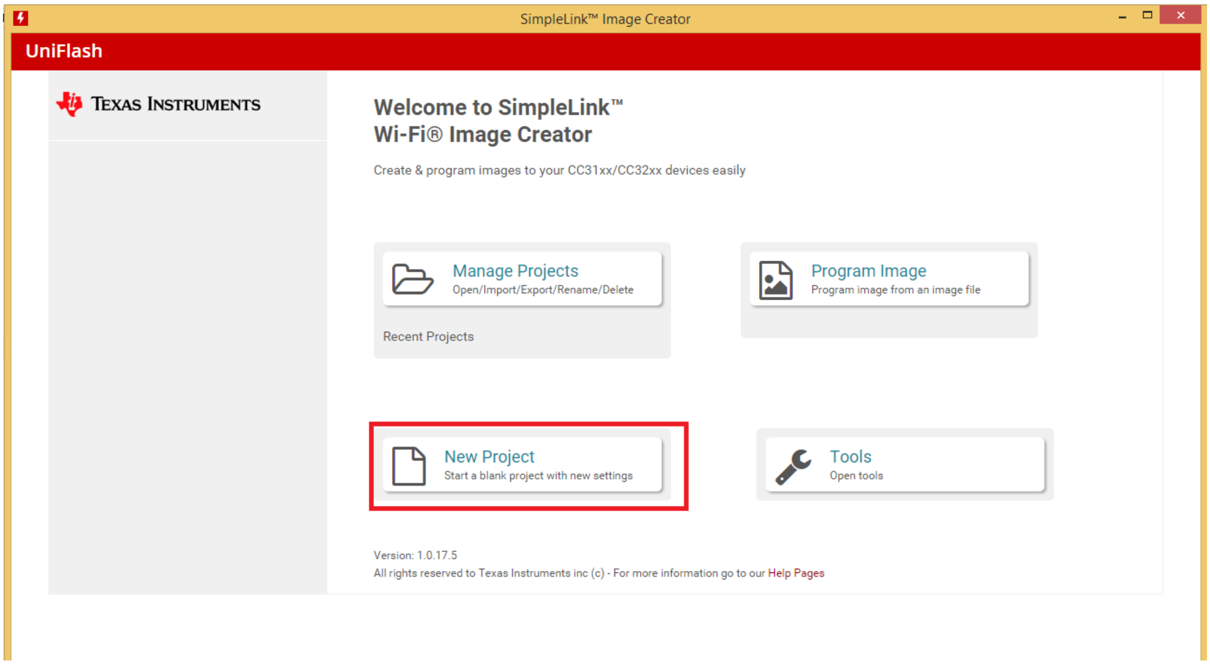
Selected Connection: Serial (UART) Interface

Select a .ccxml file to create a new session.

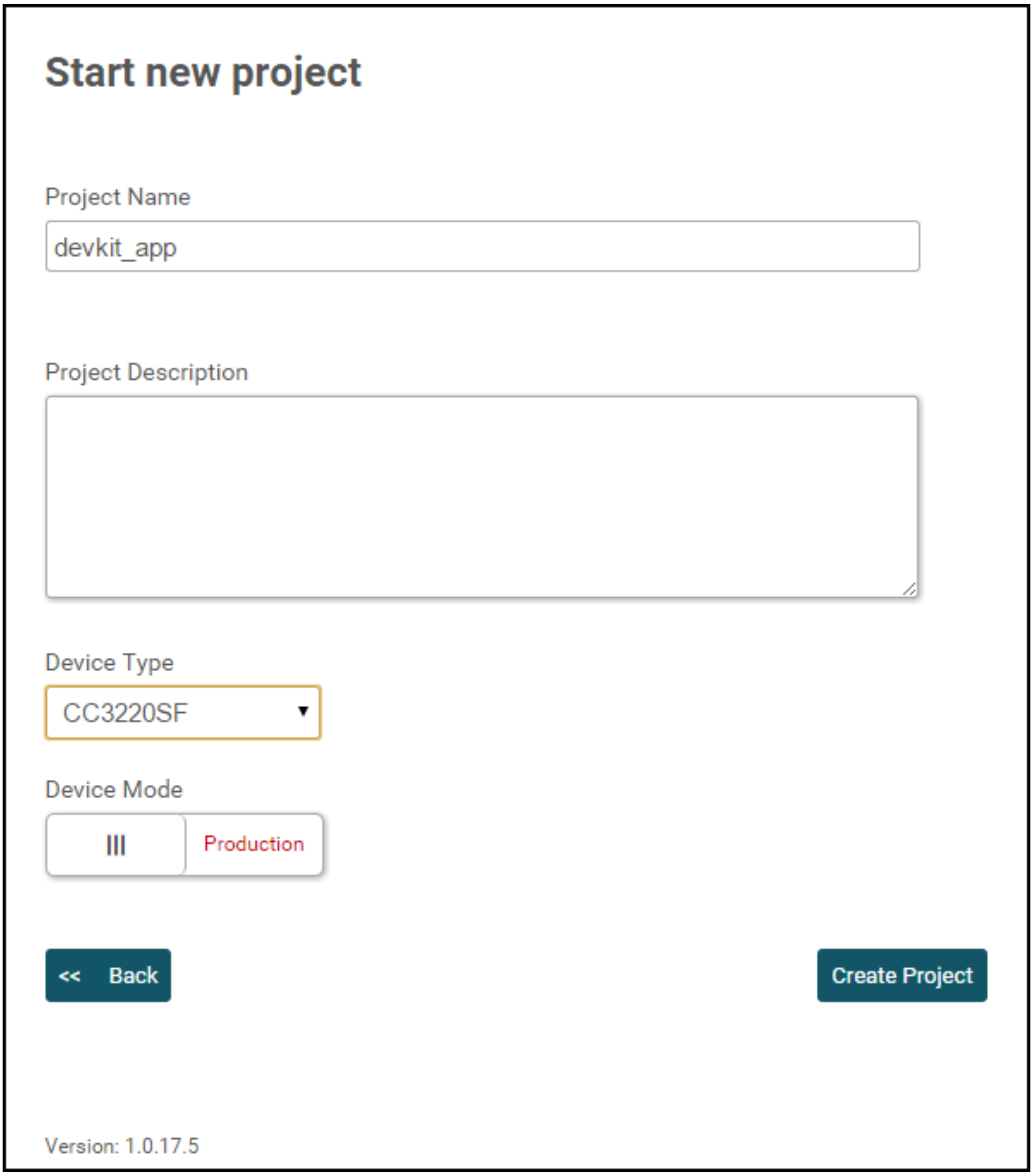
d. Click on “Start Image Creator” button



e. Click on "New Project" button to create new project.


SimpleLink™ Image Creator
UniFlash
Texas Instruments
Welcome to SimpleLink™
Wi-Fi® Image Creator
Create & program images to your CC31xx/CC32xx devices easily
Manage Projects
Open/Import/Export/Rename/Delete
Recent Projects
Program Image
Program image from an image file
New Project
Start a blank project with new settings
Tools
Open tools
Version: 1.0.17.5
All rights reserved to Texas Instruments inc (c) - For more information go to our Help Pages

f. Enter project name as "devkit_app" and select device type as "CC3220SF" and device mode as "Production". Click on "Create Project" button to create project



Start new project

Project Name

devkit_app

Project Description

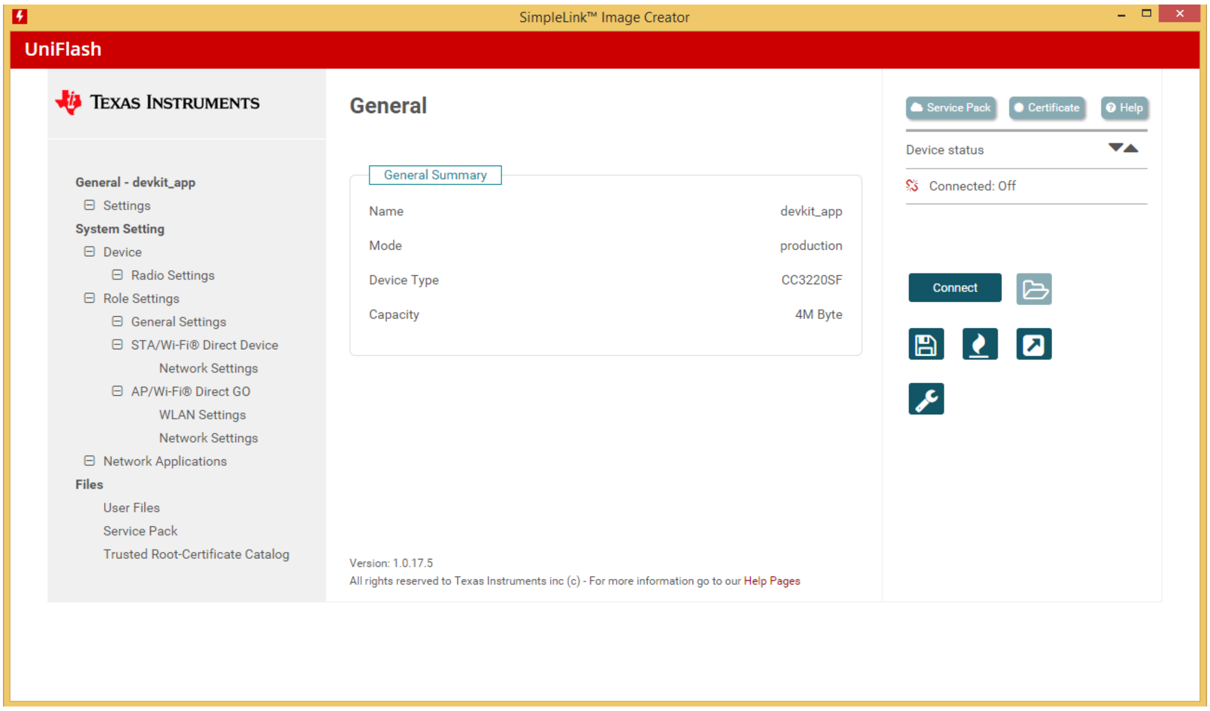
Device Type

CC3220SF

Device Mode

Version: 1.0.17.5

g. Click on connect button to Connect to device.



General Summary

Name	devkit_app
Mode	production
Device Type	CC3220SF
Capacity	4M Byte

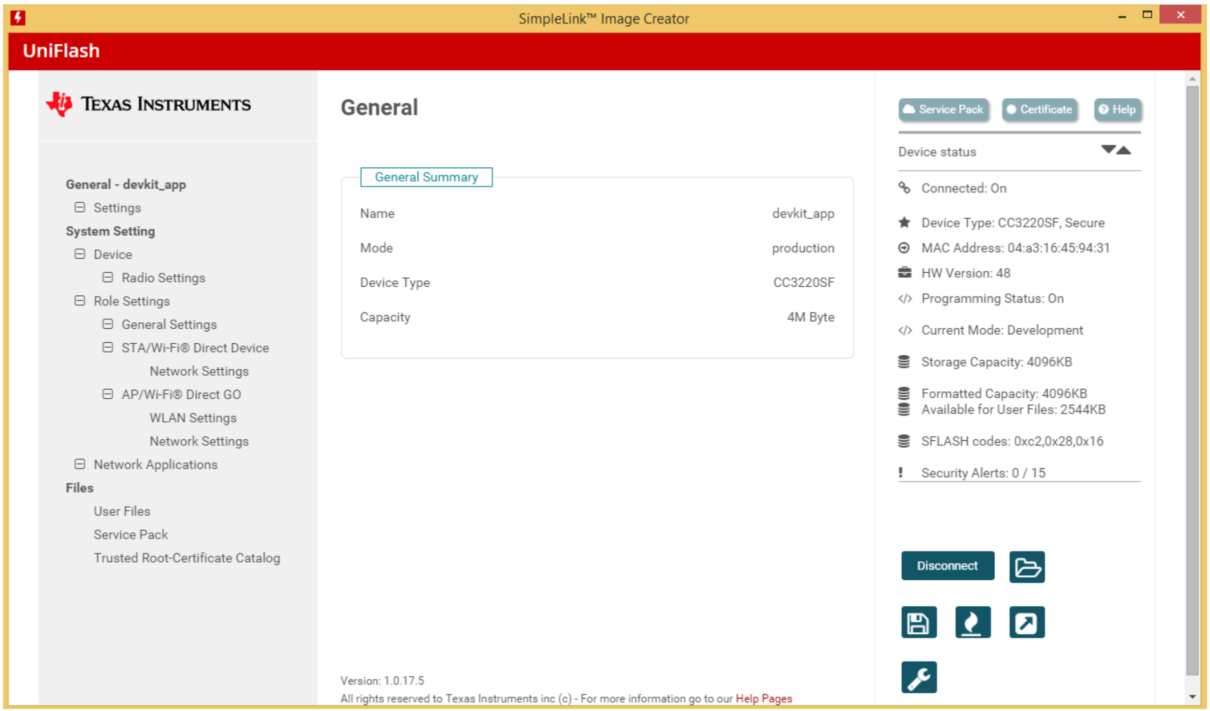
Device status

Connected: Off

Version: 1.0.17.5

All rights reserved to Texas Instruments inc (c) - For more information go to our Help Pages

h. Check if Connection status in On and Device Type as “CC3220SF, Secure”. If device is not secured then skip certificate related steps.



General Summary

Name	devkit_app
Mode	production
Device Type	CC3220SF
Capacity	4M Byte

Device status

Connected: On

Device Type: CC3220SF, Secure

MAC Address: 04:a3:16:45:94:31

HW Version: 48

Programming Status: On

Current Mode: Development

Storage Capacity: 4096KB

Formatted Capacity: 4096KB
Available for User Files: 2544KB

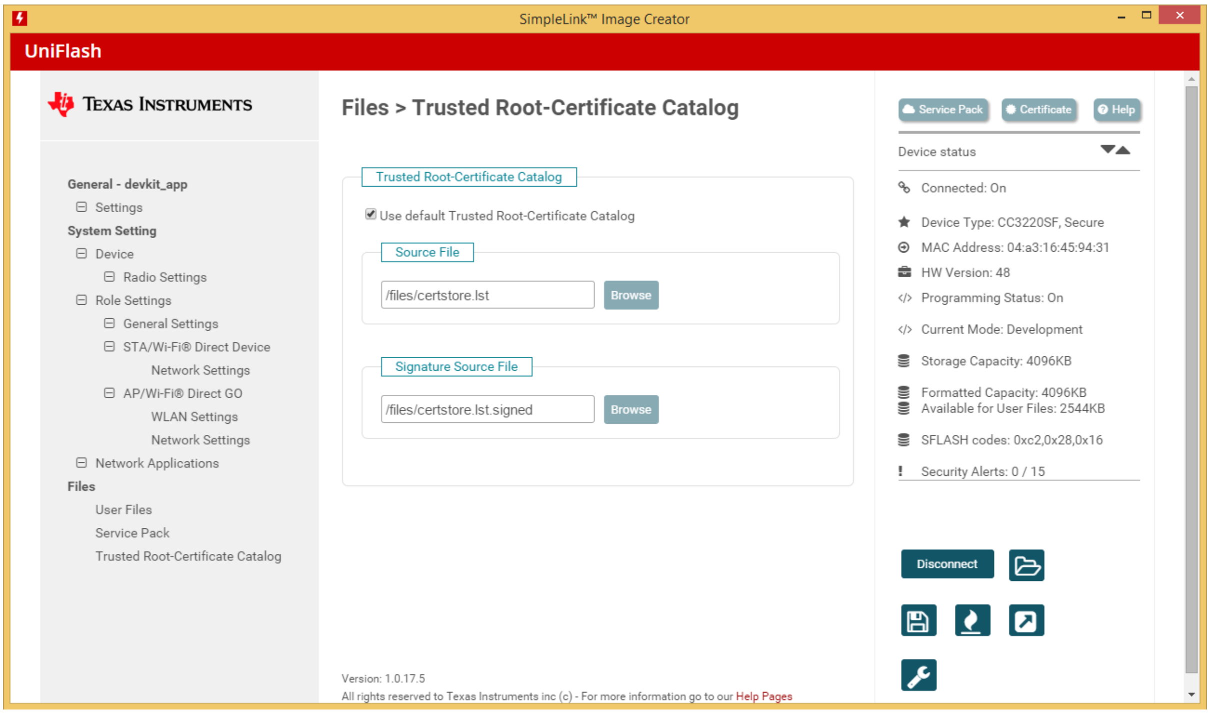
SFLASH codes: 0xc2,0x28,0x16

Security Alerts: 0 / 15

Version: 1.0.17.5

All rights reserved to Texas Instruments inc (c) - For more information go to our Help Pages

i. Select “Trusted Root Certificate Catalog” under “Files” section



Files > Trusted Root-Certificate Catalog

Trusted Root-Certificate Catalog

☑ Use default Trusted Root-Certificate Catalog

Source File: /files/certstore.lst

Signature Source File: /files/certstore.lst.signed

Device status

Connected: On

Device Type: CC3220SF, Secure

MAC Address: 04:a3:16:45:94:31

HW Version: 48

Programming Status: On

Current Mode: Development

Storage Capacity: 4096KB

Formatted Capacity: 4096KB
Available for User Files: 2544KB

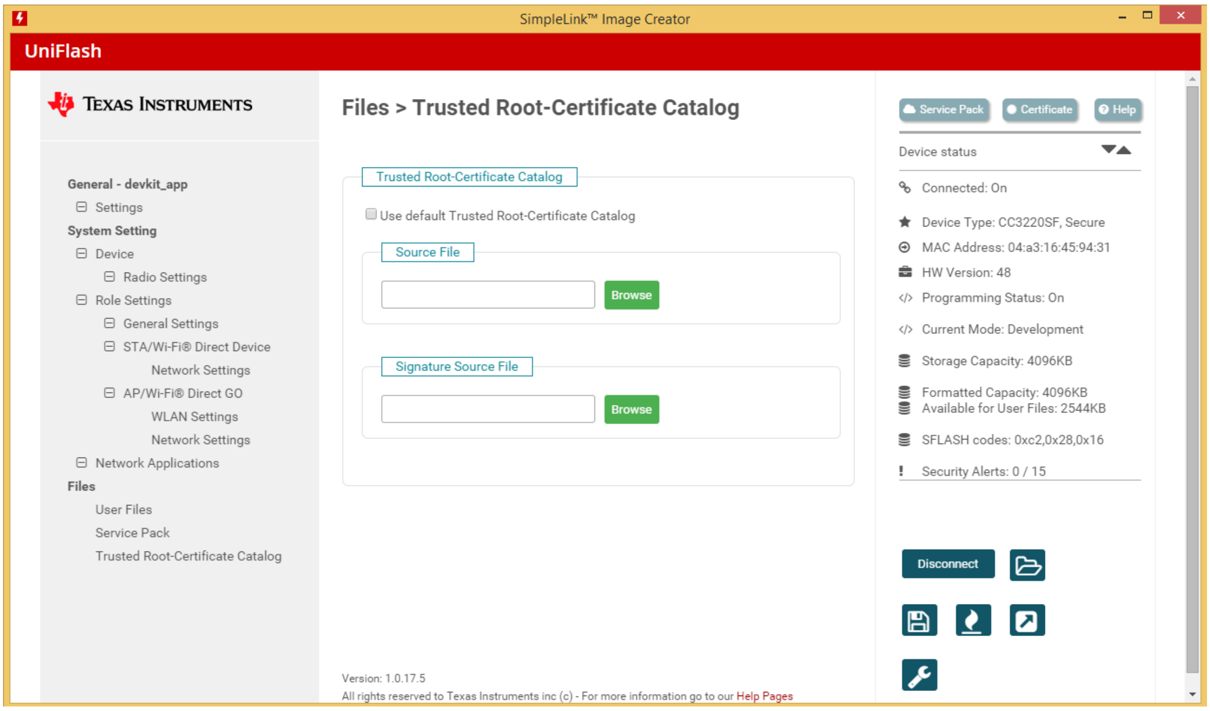
SFLASH codes: 0xc2,0x28,0x16

Security Alerts: 0 / 15

Version: 1.0.17.5

All rights reserved to Texas Instruments inc (c) - For more information go to our Help Pages

j. Uncheck“Use default Trusted Root Certificate Catalog” option



Files > Trusted Root-Certificate Catalog

Trusted Root-Certificate Catalog

☐ Use default Trusted Root-Certificate Catalog

Source File:

Signature Source File:

Device status

Connected: On

Device Type: CC3220SF, Secure

MAC Address: 04:a3:16:45:94:31

HW Version: 48

Programming Status: On

Current Mode: Development

Storage Capacity: 4096KB

Formatted Capacity: 4096KB
Available for User Files: 2544KB

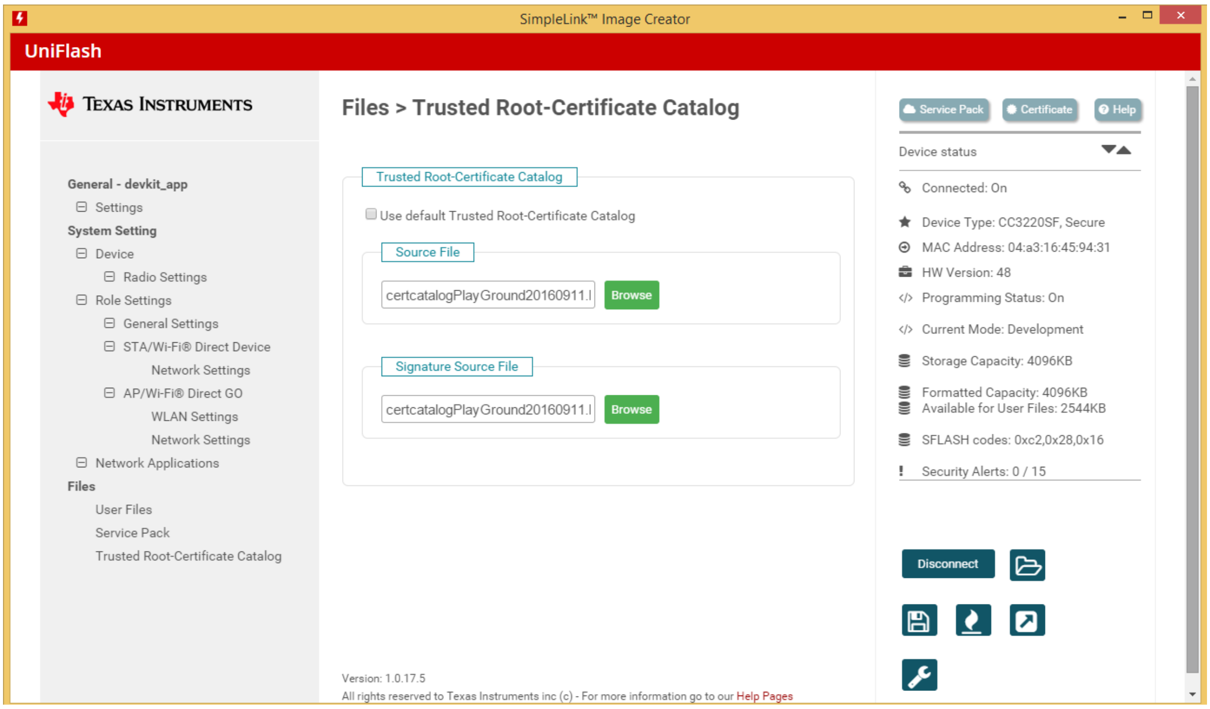
SFLASH codes: 0xc2,0x28,0x16

Security Alerts: 0 / 15

Version: 1.0.17.5

All rights reserved to Texas Instruments inc (c) - For more information go to our Help Pages

k. Browse “Source File” to “C:\TI\simplelink_cc32xx_sdk_1_02_02_00\tools\cc32xx_tools\certificate-playground\certcatalogPlayGround20160911.lst” file and “Signature Source File” to “C:\TI\simplelink_cc32xx_sdk_1_02_02_00\tools\cc32xx_tools\certificate-playground\certcatalogPlayGround20160911.lst.signed.bin” file.



Files > Trusted Root-Certificate Catalog

Trusted Root-Certificate Catalog

□ Use default Trusted Root-Certificate Catalog

Source File: certcatalogPlayGround20160911.l

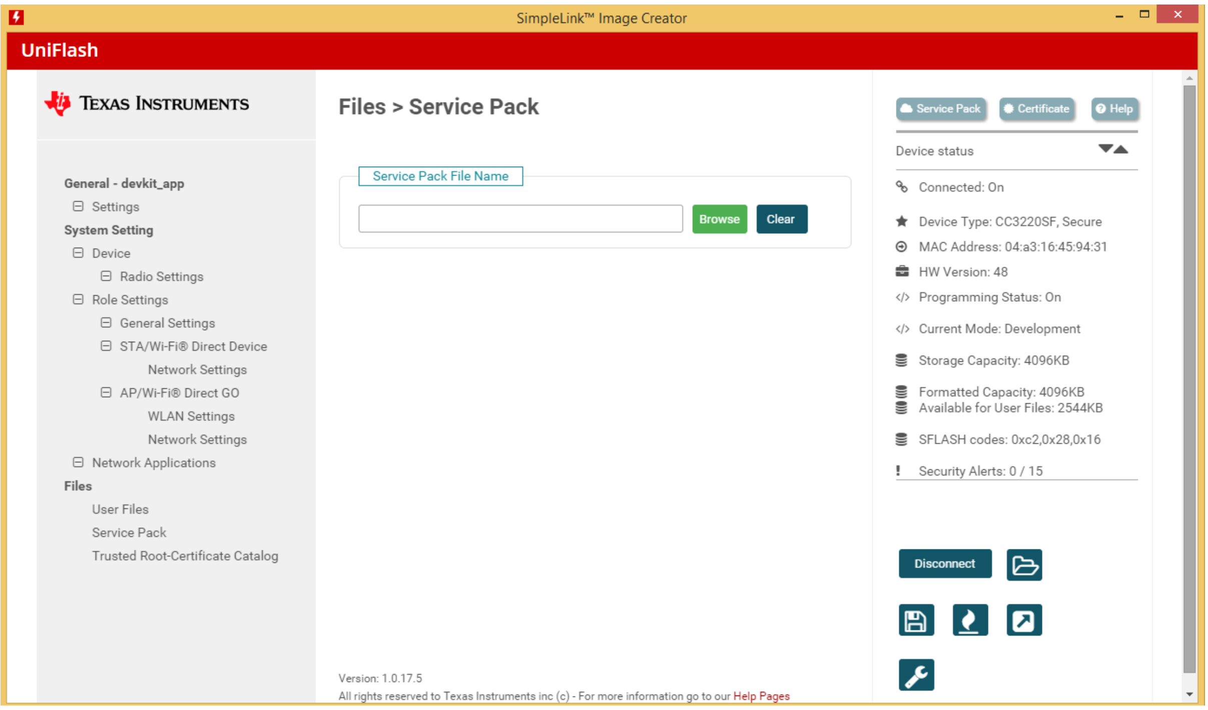
Signature Source File: certcatalogPlayGround20160911.l

Device status

- Connected: On
- Device Type: CC3220SF, Secure
- MAC Address: 04:a3:16:45:94:31
- HW Version: 48
- Programming Status: On
- Current Mode: Development
- Storage Capacity: 4096KB
- Formatted Capacity: 4096KB
  Available for User Files: 2544KB
- SFLASH codes: 0xc2,0x28,0x16
- Security Alerts: 0 / 15

Version: 1.0.17.5
All rights reserved to Texas Instruments inc (c) - For more information go to our Help Pages

l. Select “Service Pack” under “Files” section



Files > Service Pack

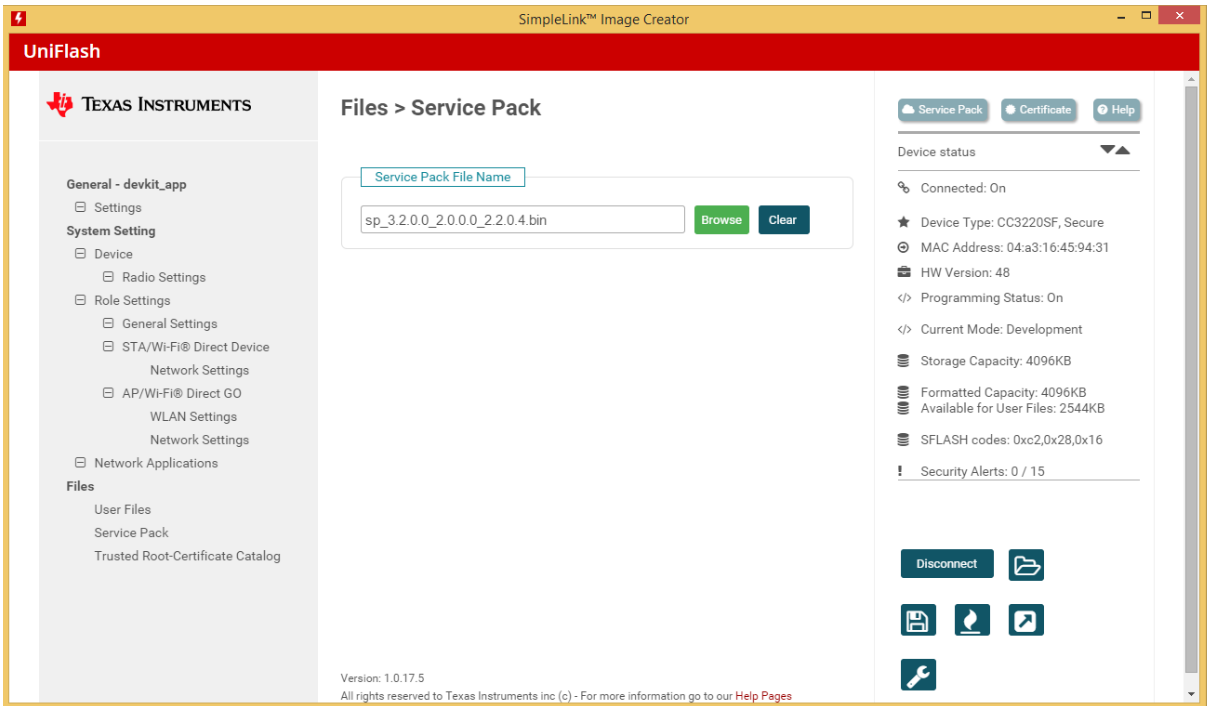
Service Pack File Name:

Device status

- Connected: On
- Device Type: CC3220SF, Secure
- MAC Address: 04:a3:16:45:94:31
- HW Version: 48
- Programming Status: On
- Current Mode: Development
- Storage Capacity: 4096KB
- Formatted Capacity: 4096KB
  Available for User Files: 2544KB
- SFLASH codes: 0xc2,0x28,0x16
- Security Alerts: 0 / 15

Version: 1.0.17.5
All rights reserved to Texas Instruments inc (c) - For more information go to our Help Pages

m. Browse “Service Pack FIle Name” to “C:\TI\simplelink_cc32xx_sdk_1_02_02_00\tools\cc32xx_tools\servicepack-cc3x20\sp_3.2.0.0_2.0.0.0_2.2.0.4.bin”.



Files > Service Pack

Service Pack File Name: sp_3.2.0.0_2.0.0.0_2.2.0.4.bin

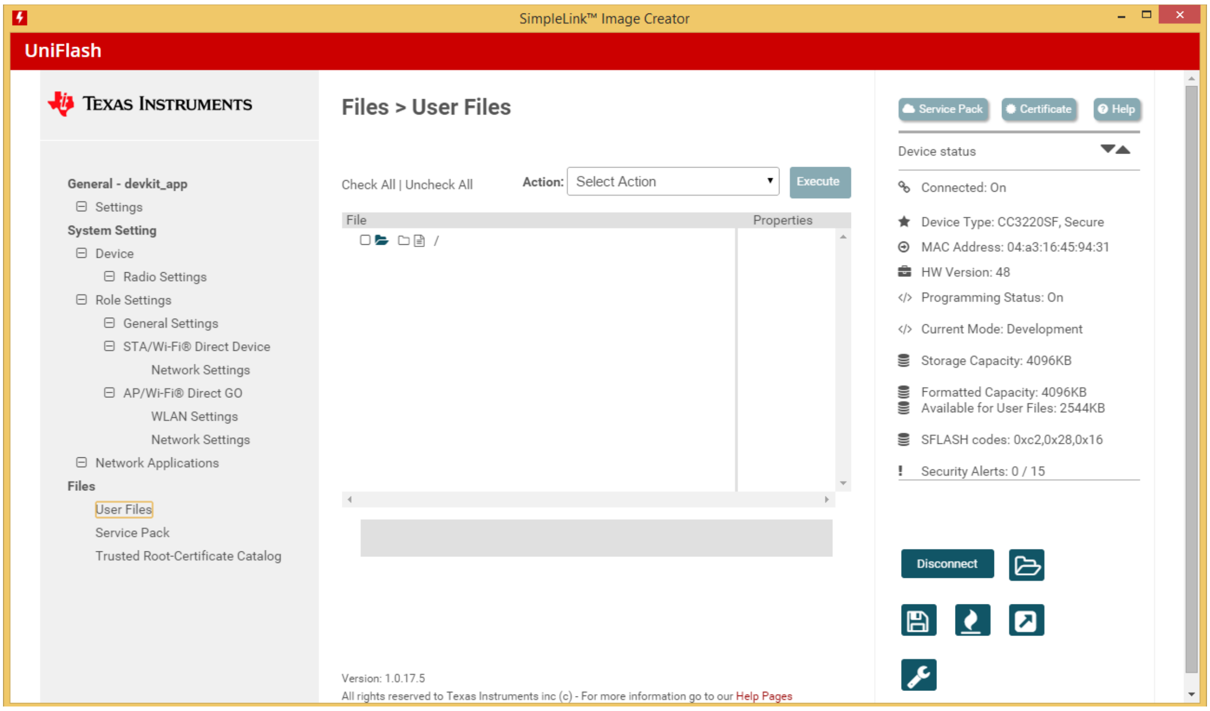
Device status

- Connected: On
- Device Type: CC3220SF, Secure
- MAC Address: 04:a3:16:45:94:31
- HW Version: 48
- Programming Status: On
- Current Mode: Development
- Storage Capacity: 4096KB
- Formatted Capacity: 4096KB
- Available for User Files: 2544KB
- SFLASH codes: 0xc2,0x28,0x16
- Security Alerts: 0 / 15

Version: 1.0.17.5

All rights reserved to Texas Instruments inc (c) - For more information go to our Help Pages

n. Select “User Files” under “Files” section



Files > User Files

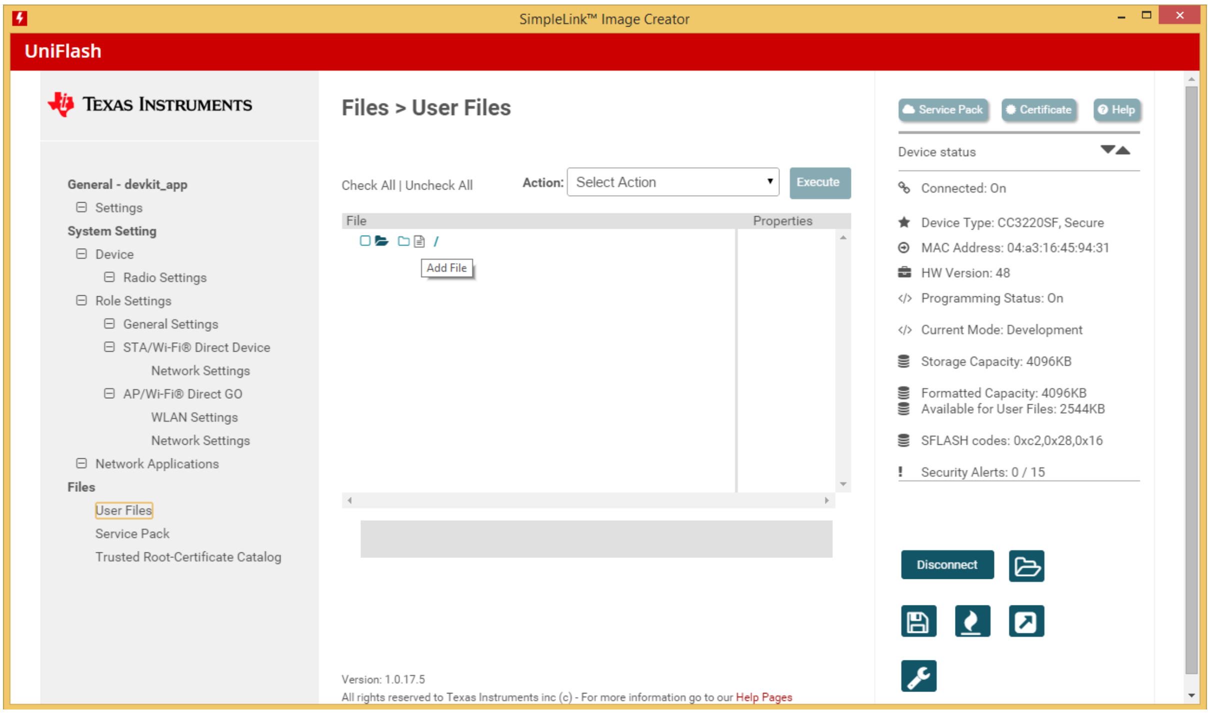
Device status

- Connected: On
- Device Type: CC3220SF, Secure
- MAC Address: 04:a3:16:45:94:31
- HW Version: 48
- Programming Status: On
- Current Mode: Development
- Storage Capacity: 4096KB
- Formatted Capacity: 4096KB
- Available for User Files: 2544KB
- SFLASH codes: 0xc2,0x28,0x16
- Security Alerts: 0 / 15

Version: 1.0.17.5

All rights reserved to Texas Instruments inc (c) - For more information go to our Help Pages

o. Click on add file icon as shown in below screenshot



Connected: On

Device Type: CC3220SF, Secure

MAC Address: 04:a3:16:45:94:31

HW Version: 48

Programming Status: On

Current Mode: Development

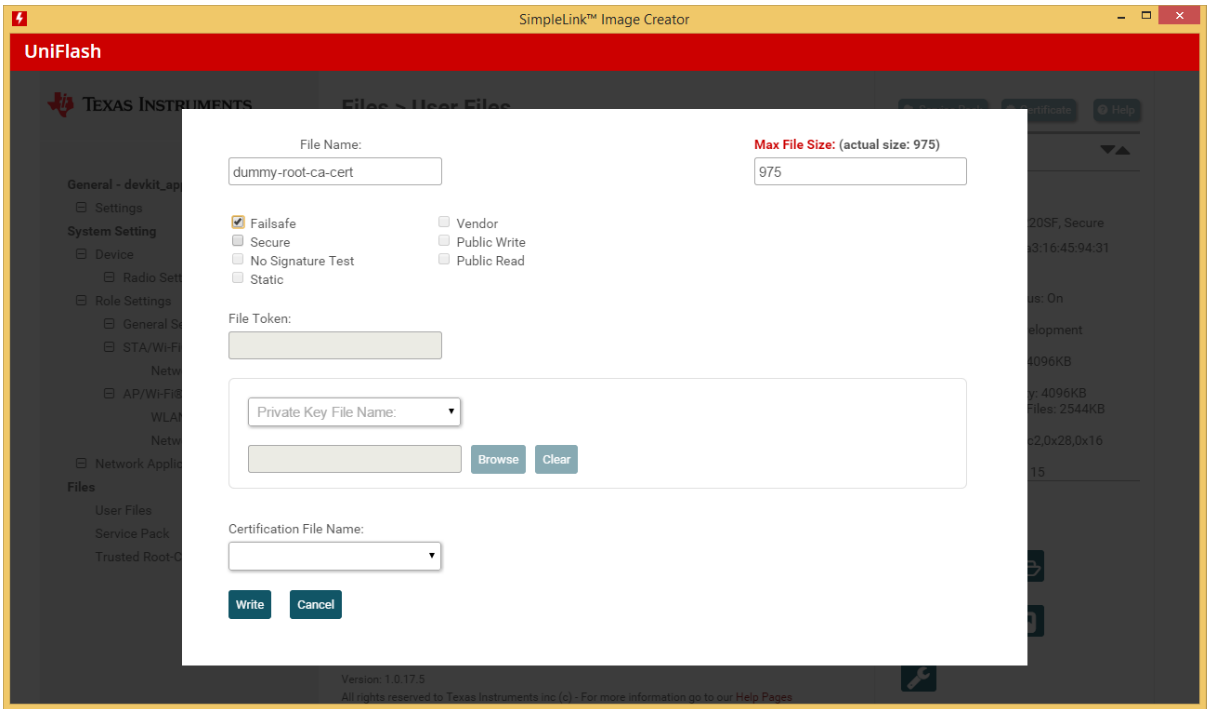
Storage Capacity: 4096KB

Formatted Capacity: 4096KB
Available for User Files: 2544KB

SFLASH codes: 0xc2,0x28,0x16

Security Alerts: 0 / 15

p. Browse to “C:\TI\simplelink_cc32xx_sdk_1_02_02_00\tools\cc32xx_tools\certificate-playground\dummy-root-ca-cert” to add file. After selection file configuration window will appear as shown in below screenshot.



File Name: dummy-root-ca-cert

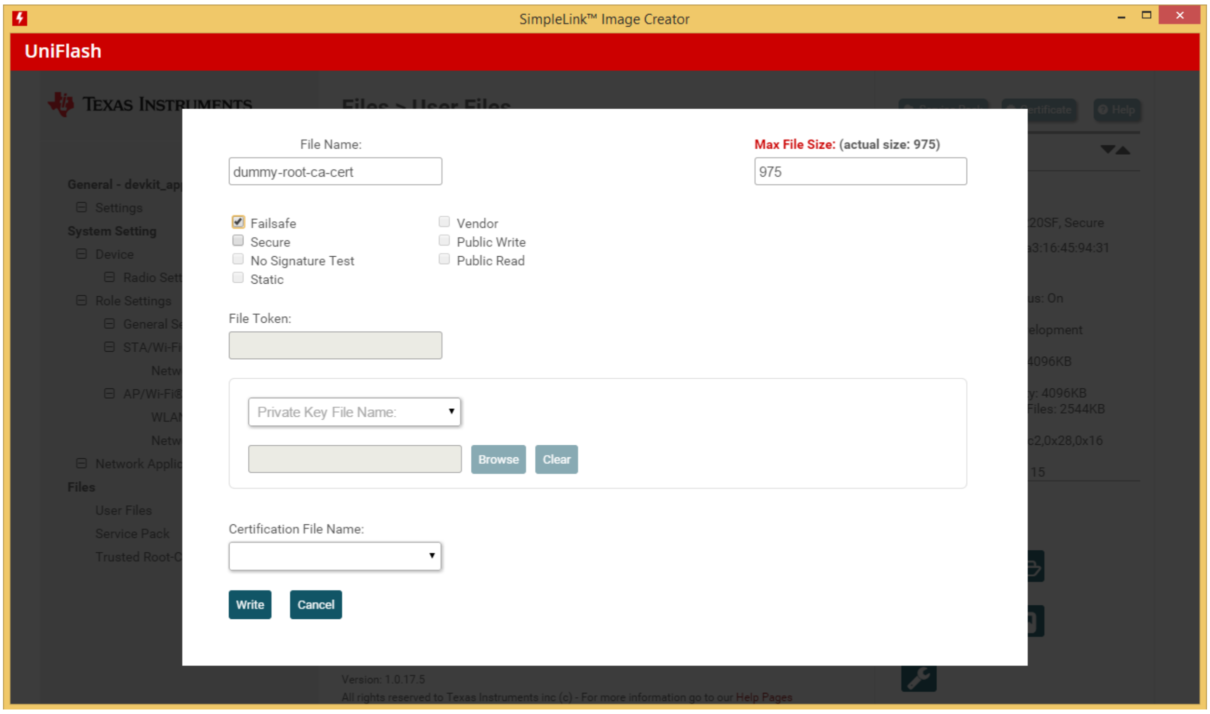
Max File Size: (actual size: 975) 975

☑ Failsafe
☐ Secure
☐ No Signature Test
☐ Static
☐ Vendor
☐ Public Write
☐ Public Read

File Token:

Certification File Name:

q. Select “Failsafe” option and click on Write button to add File to project.



File Name: dummy-root-ca-cert

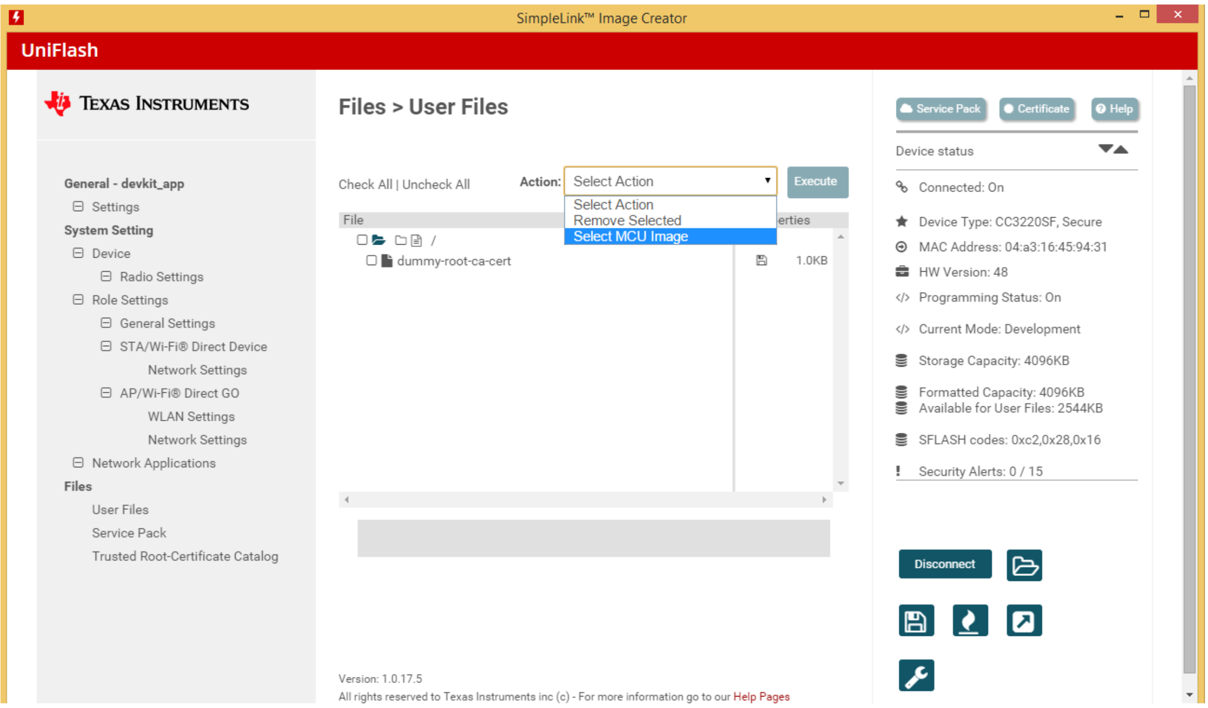
Max File Size: (actual size: 975) 975

☑ Failsafe
☐ Secure
☐ No Signature Test
☐ Static
☐ Vendor
☐ Public Write
☐ Public Read

File Token:

Certification File Name:

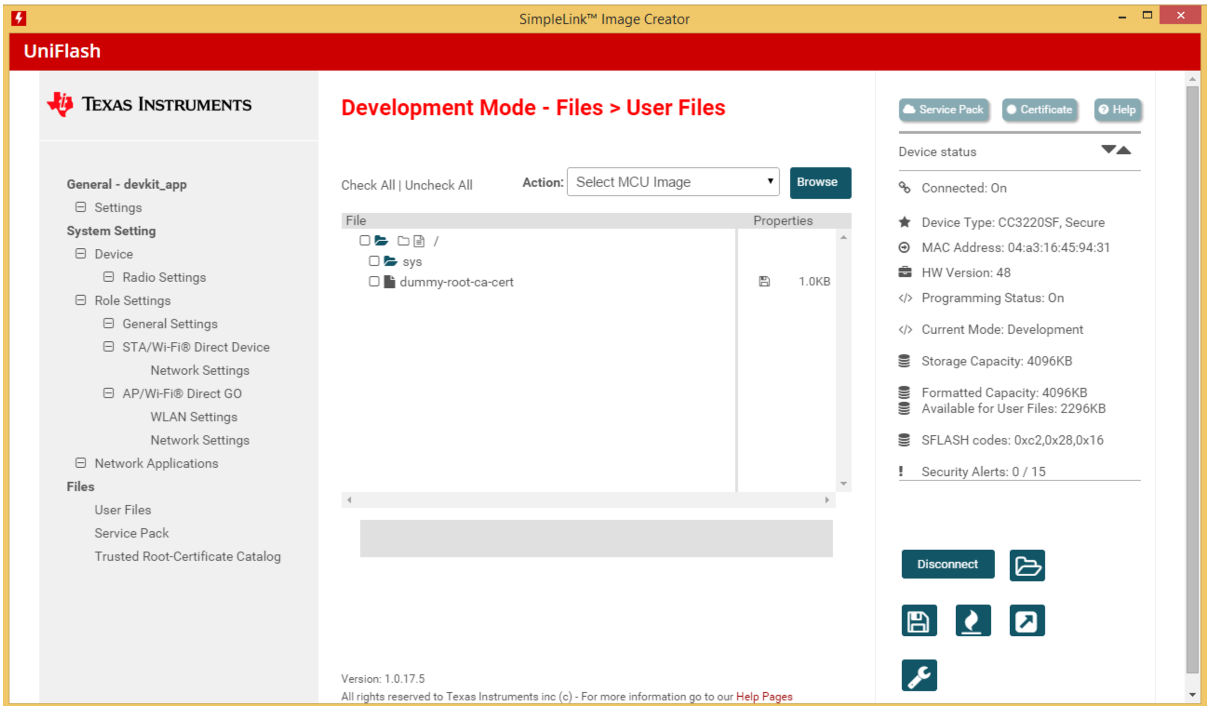
r. Select “ Select MCU Image” option in “Action” dropdown menu.



dummy-root-ca-cert 1.0KB

Connected: On
Device Type: CC3220SF, Secure
MAC Address: 04:a3:16:45:94:31
HW Version: 48
Programming Status: On
Current Mode: Development
Storage Capacity: 4096KB
Formatted Capacity: 4096KB
Available for User Files: 2544KB
SFLASH codes: 0xc2,0x28,0x16
Security Alerts: 0 / 15

s. Click on “Browse” button and navigate to devkit_app binary located at (Debug- “*<PATH TO DEVKIT APP BUNDLE>*\app\src\devkit_app\cc3220\CCS\Debug\devkit_app.bin” and Release- “*<PATH TO DEVKIT APP BUNDLE>*\app\src\devkit_app\cc3220\CCS\Release\devkit_app.bin”



Development Mode - Files > User Files

General - devkit_app
- Settings

System Setting
- Device
  - Radio Settings
- Role Settings
  - General Settings
  - STA/Wi-Fi® Direct Device
    - Network Settings
  - AP/Wi-Fi® Direct GO
    - WLAN Settings
    - Network Settings
- Network Applications

Files
- User Files
- Service Pack
- Trusted Root-Certificate Catalog

Check All | Uncheck All   Action: Select MCU Image

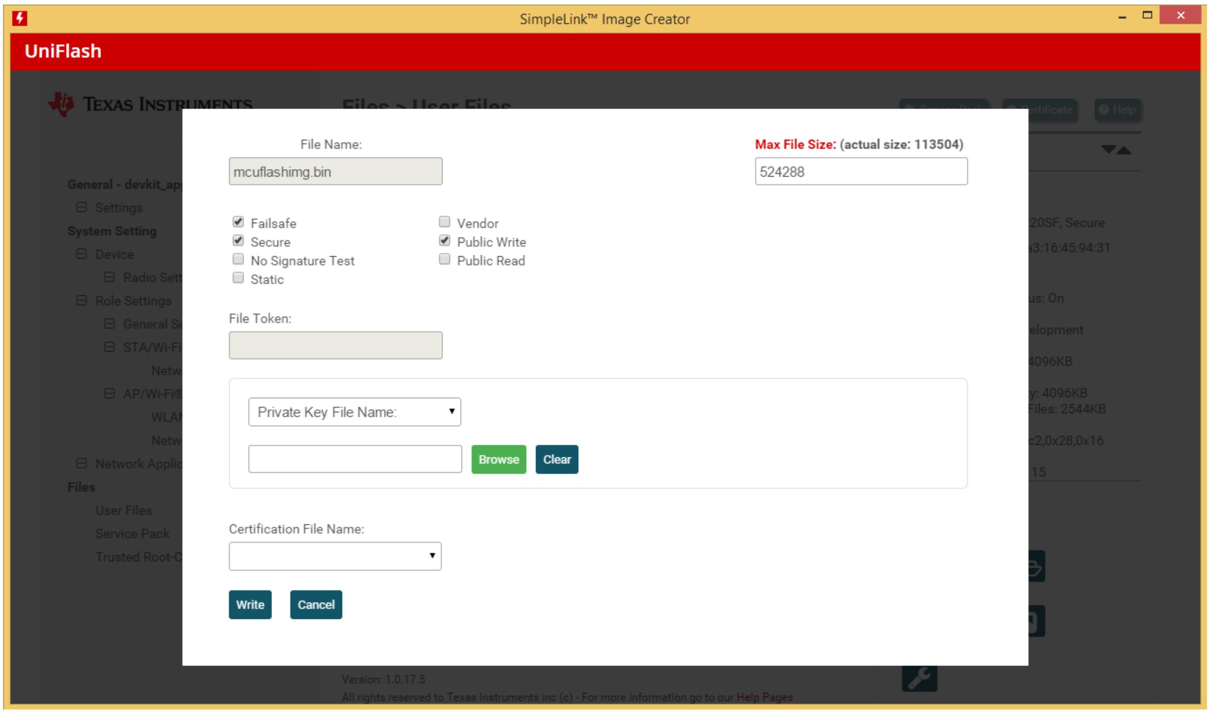
File	Properties
□ /	
□ sys	
□ dummy-root-ca-cert	1.0KB

Device status

- Connected: On
- Device Type: CC3220SF, Secure
- MAC Address: 04:a3:16:45:94:31
- HW Version: 48
- Programming Status: On
- Current Mode: Development
- Storage Capacity: 4096KB
- Formatted Capacity: 4096KB
  Available for User Files: 2296KB
- SFLASH codes: 0xc2,0x28,0x16
- Security Alerts: 0 / 15

Version: 1.0.17.5
All rights reserved to Texas Instruments inc (c) - For more information go to our Help Pages

t. Select “Failsafe”, “Secure” and “Public Write” options



File Name: mcuflashimg.bin

Max File Size: (actual size: 113504) 524288

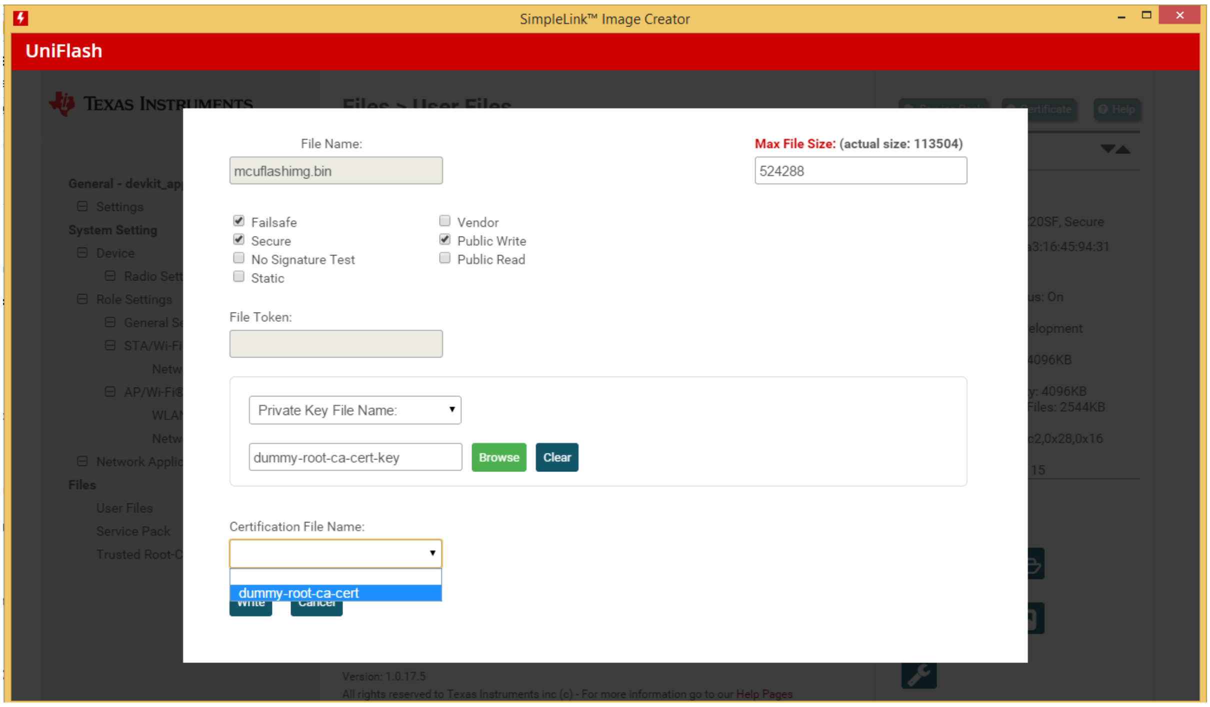
☑ Failsafe   ☐ Vendor
☑ Secure   ☑ Public Write
☐ No Signature Test   ☐ Public Read
☐ Static

File Token:

Private Key File Name:

Certification File Name:

u. Select “Private Key File Name” from the drop down menu and browse to “C:\TI\simplelink_cc32xx_sdk_1_02_02_00\tools\cc32xx_tools\certificate-playground\dummy-root-ca-cert-key”.



File Name: mcuflashimg.bin

Max File Size: (actual size: 113504) 524288

☑ Failsafe
☑ Secure
□ No Signature Test
□ Static
□ Vendor
☑ Public Write
□ Public Read

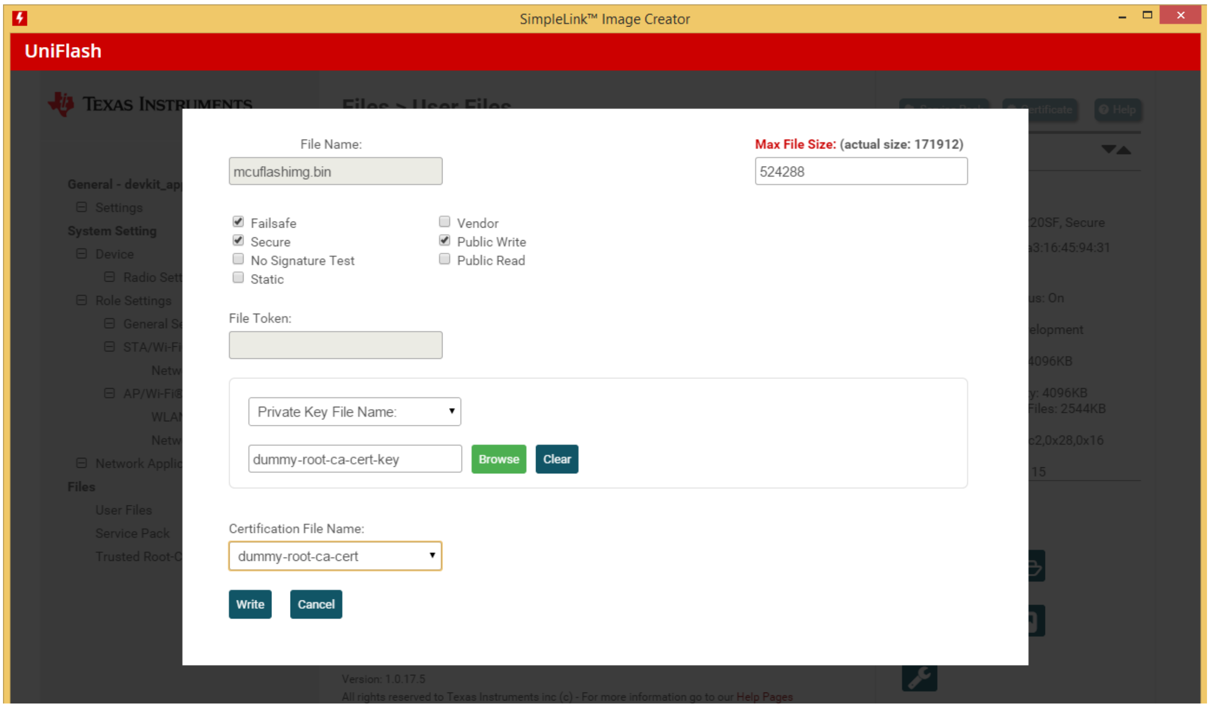
File Token:

Private Key File Name: dummy-root-ca-cert-key [Browse] [Clear]

Certification File Name:
dummy-root-ca-cert

[Write] [Cancel]

v. Select “dummy-root-ca-cert” from “Certificate File name” dropdown menu.



File Name: mcuflashimg.bin

Max File Size: (actual size: 171912) 524288

☑ Failsafe
☑ Secure
□ No Signature Test
□ Static
□ Vendor
☑ Public Write
□ Public Read

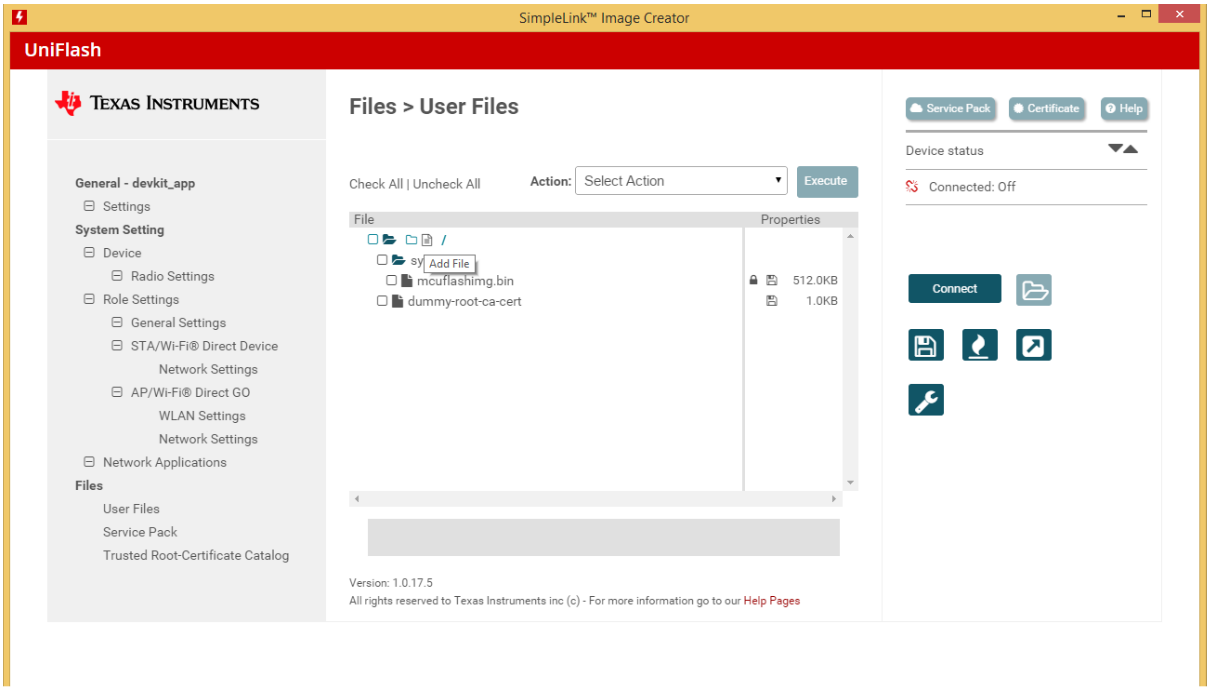
File Token:

Private Key File Name: dummy-root-ca-cert-key [Browse] [Clear]

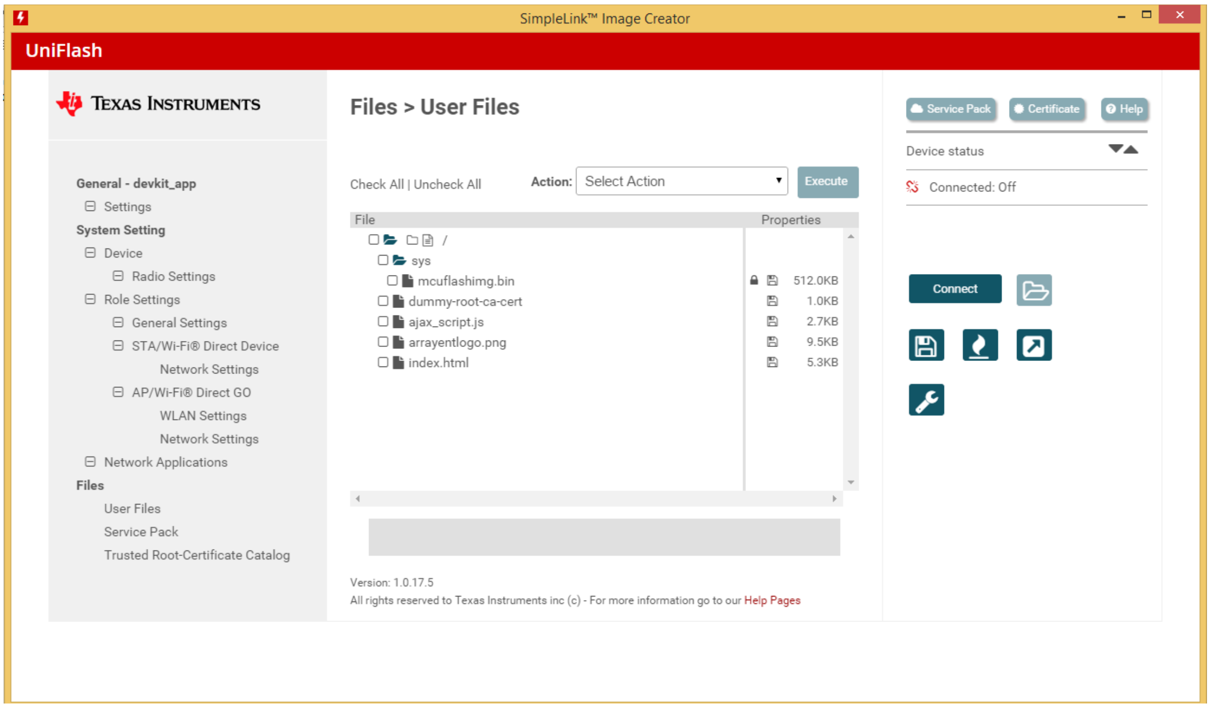
Certification File Name: dummy-root-ca-cert

[Write] [Cancel]

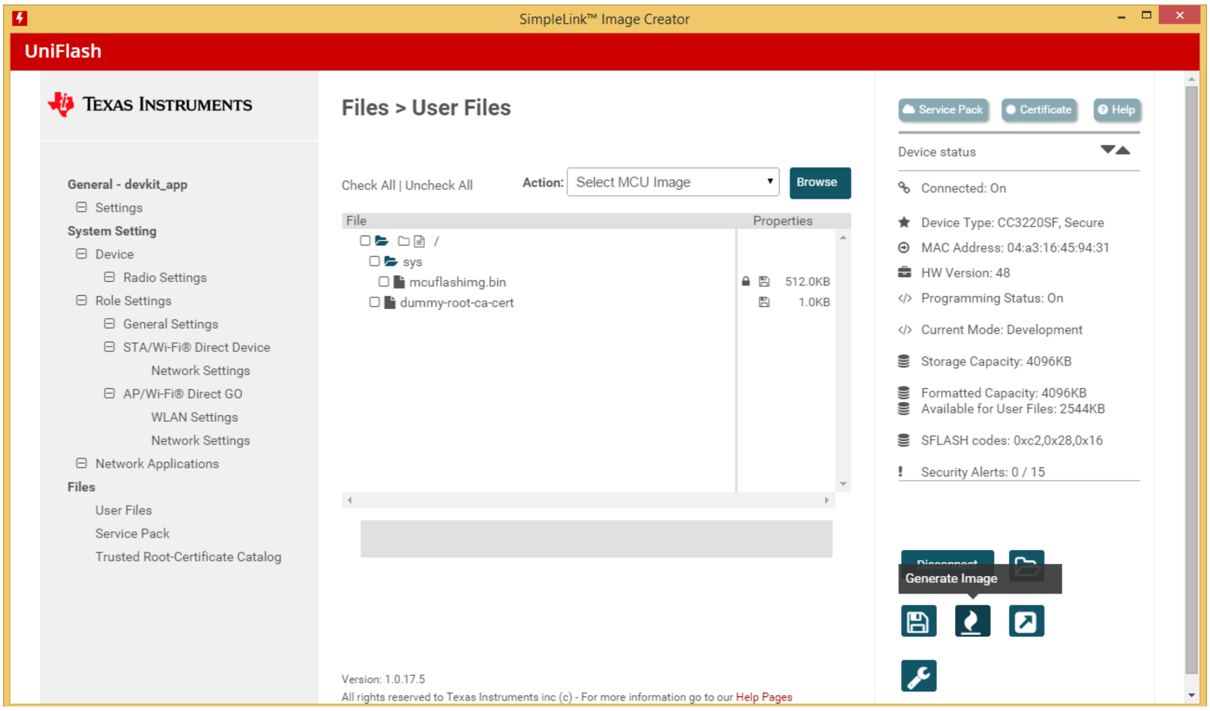
w. Click on add file icon as shown in below screenshot



x. Add “ajax_script.js”, “arrayentlogo.png” and “index.html” (located at <*DEVKIT RELEASE BUNDLE*>/app/devkit_app/resources/arrayent) file using add button.



y. Click on “Generate Image” icon as shown in below screenshot



File		
□ □ /		
□ sys		
□ mcuflashimg.bin		512.0KB
□ dummy-root-ca-cert		1.0KB

Connected: On

Device Type: CC3220SF, Secure

MAC Address: 04:a3:16:45:94:31

HW Version: 48

Programming Status: On

Current Mode: Development

Storage Capacity: 4096KB

Formatted Capacity: 4096KB
Available for User Files: 2544KB

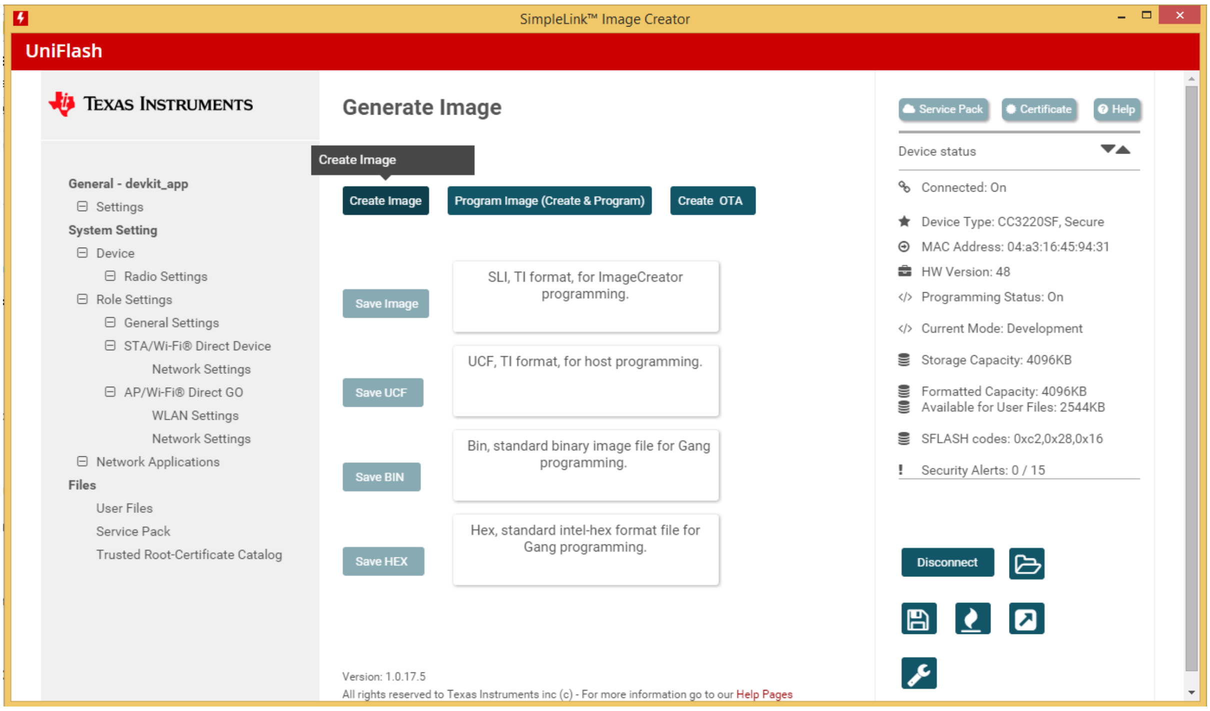
SFLASH codes: 0xc2,0x28,0x16

Security Alerts: 0 / 15

Version: 1.0.17.5

All rights reserved to Texas Instruments inc (c) - For more information go to our Help Pages

z. Click on “Create Image” button to generate Image.



Save Image — SLI, TI format, for ImageCreator programming.

Save UCF — UCF, TI format, for host programming.

Save BIN — Bin, standard binary image file for Gang programming.

Save HEX — Hex, standard intel-hex format file for Gang programming.

Connected: On

Device Type: CC3220SF, Secure

MAC Address: 04:a3:16:45:94:31

HW Version: 48

Programming Status: On

Current Mode: Development

Storage Capacity: 4096KB

Formatted Capacity: 4096KB
Available for User Files: 2544KB

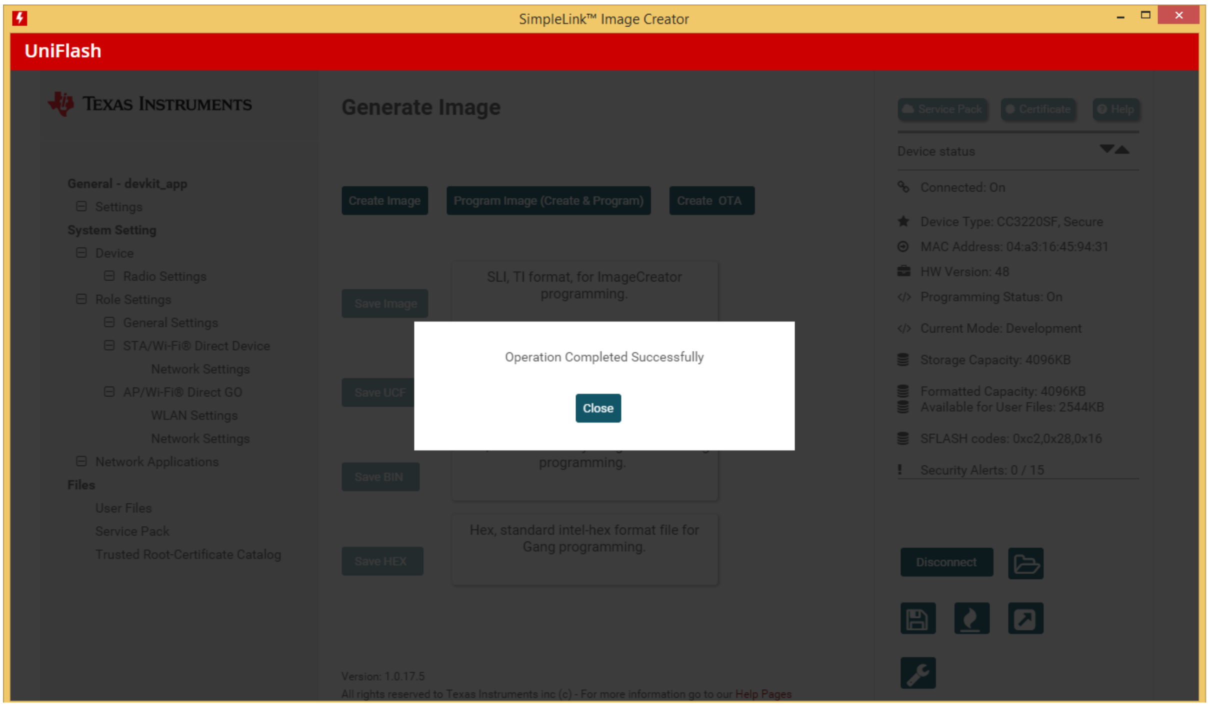
SFLASH codes: 0xc2,0x28,0x16

Security Alerts: 0 / 15

Version: 1.0.17.5

All rights reserved to Texas Instruments inc (c) - For more information go to our Help Pages

aa. On successful Image generation “Operation Successfully Completed “ Message will pop up. Click on close button to close the popup.



Operation Completed Successfully

SLI, TI format, for ImageCreator programming.

Hex, standard intel-hex format file for Gang programming.

Device status

Connected: On

Device Type: CC3220SF, Secure

MAC Address: 04:a3:16:45:94:31

HW Version: 48

Programming Status: On

Current Mode: Development

Storage Capacity: 4096KB

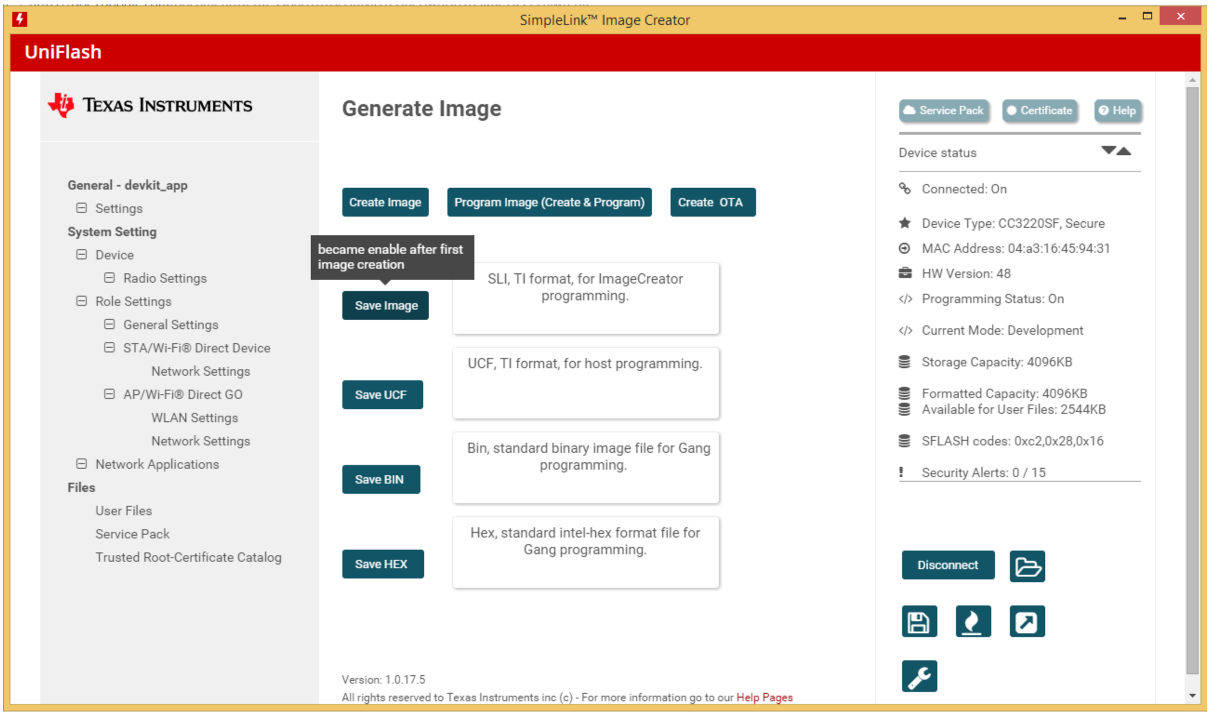
Formatted Capacity: 4096KB
Available for User Files: 2544KB

SFLASH codes: 0xc2,0x28,0x16

Security Alerts: 0 / 15

Version: 1.0.17.5
All rights reserved to Texas Instruments inc (c) - For more information go to our Help Pages

bb. Click on “Save Image” to store “.sli” file to local storage.



became enable after first image creation

SLI, TI format, for ImageCreator programming.

UCF, TI format, for host programming.

Bin, standard binary image file for Gang programming.

Hex, standard intel-hex format file for Gang programming.

Device status

Connected: On

Device Type: CC3220SF, Secure

MAC Address: 04:a3:16:45:94:31

HW Version: 48

Programming Status: On

Current Mode: Development

Storage Capacity: 4096KB

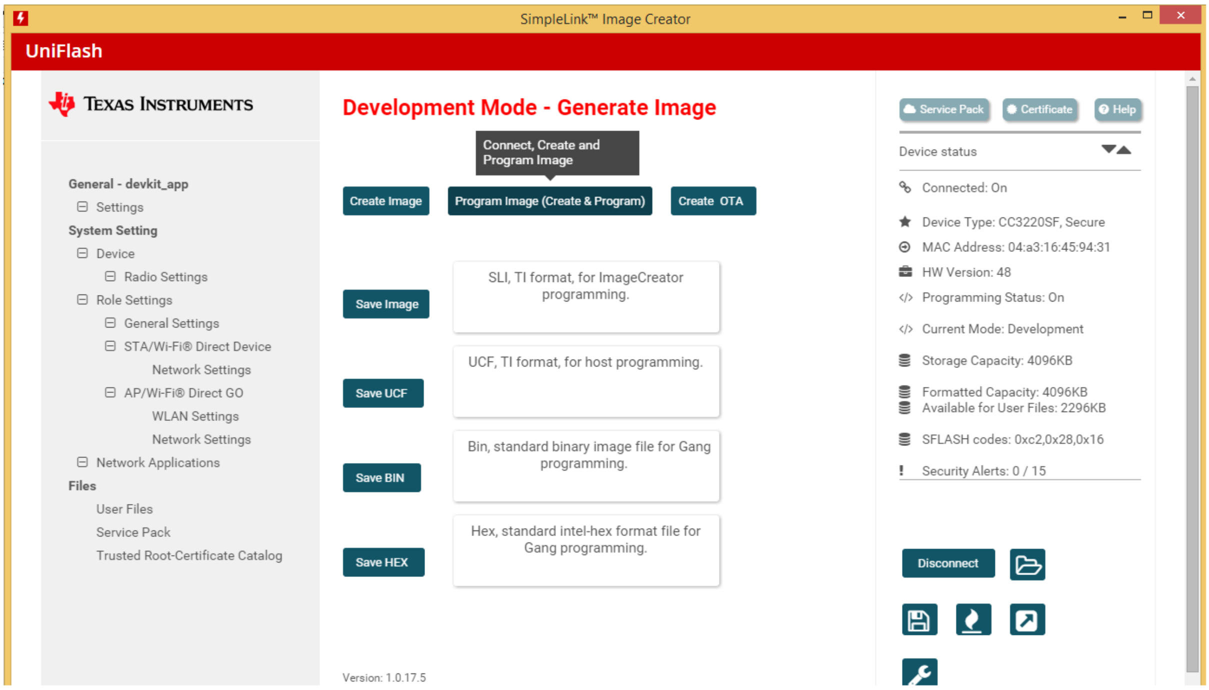
Formatted Capacity: 4096KB
Available for User Files: 2544KB

SFLASH codes: 0xc2,0x28,0x16

Security Alerts: 0 / 15

Version: 1.0.17.5
All rights reserved to Texas Instruments inc (c) - For more information go to our Help Pages

cc. Click in “Program Image(Create & Program)” button to Create and Program Image to device.



SLI, TI format, for ImageCreator programming.

UCF, TI format, for host programming.

Bin, standard binary image file for Gang programming.

Hex, standard intel-hex format file for Gang programming.

Connected: On

Device Type: CC3220SF, Secure

MAC Address: 04:a3:16:45:94:31

HW Version: 48

Programming Status: On

Current Mode: Development

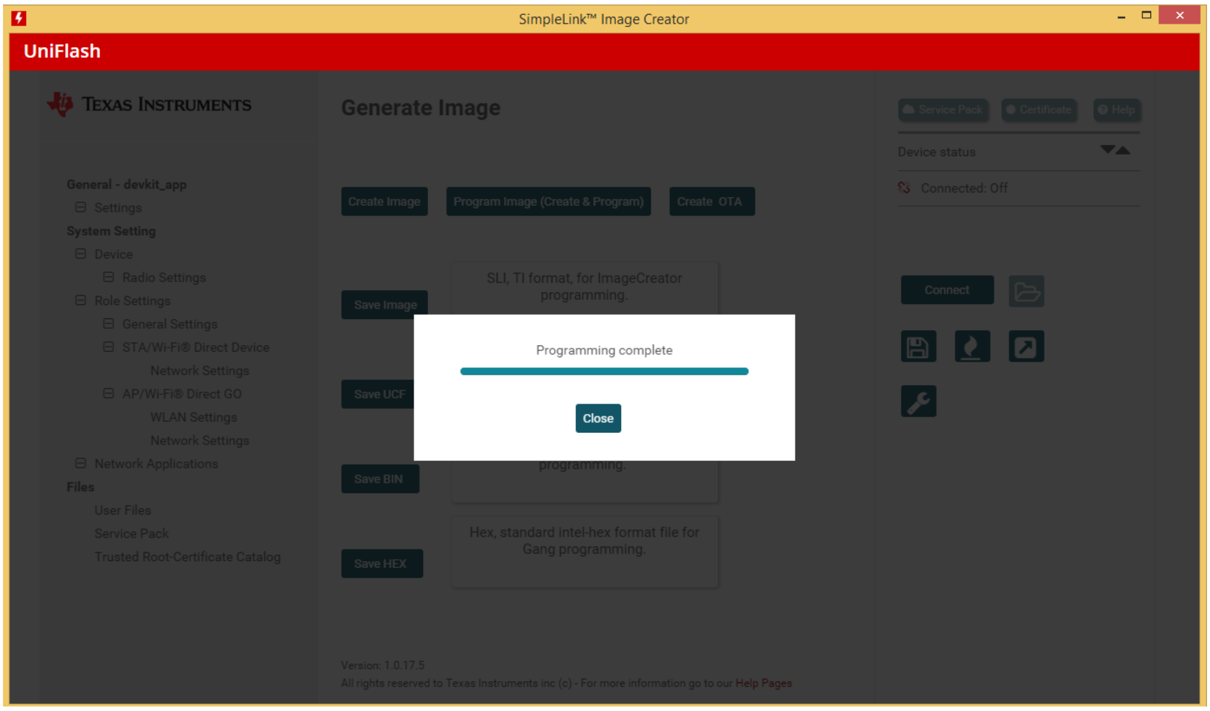
Storage Capacity: 4096KB

Formatted Capacity: 4096KB
Available for User Files: 2296KB

SFLASH codes: 0xc2,0x28,0x16

Security Alerts: 0 / 15

dd. On successful flash image to device, “Programming Complete” message will appear



Connected: Off

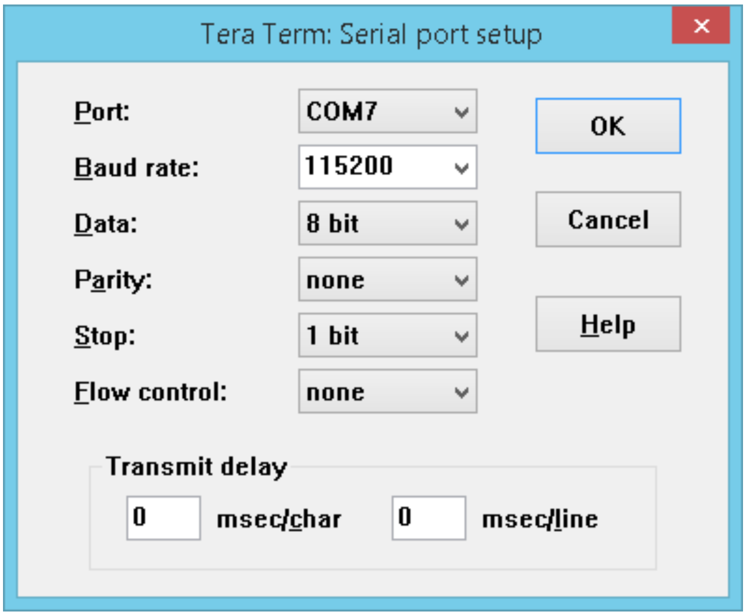
Programming complete

5. Setup terminal program

We have taken TI CC3220 as a reference board for configuring terminal and arrayent credentials. Same will be applicable for others.

a. Open a serial terminal program (example: TeraTerm or putty or minicom) and configure the serial port as shown in figures below:

i. For Teraterm (Setup->Serial Port):



Tera Term: Serial port setup

Port: COM7

Baud rate: 115200

Data: 8 bit

Parity: none

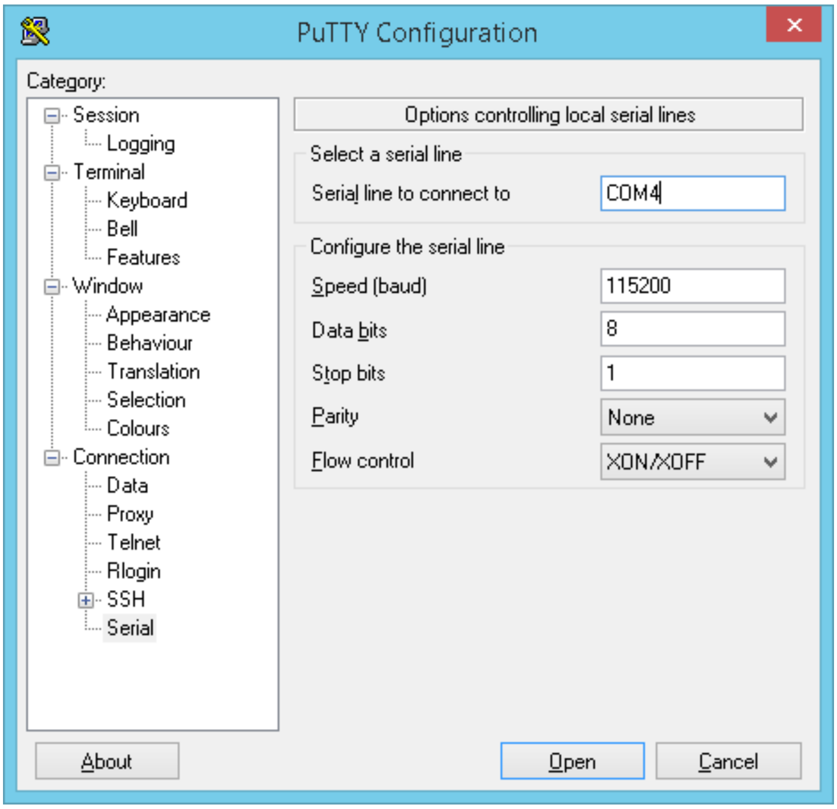
Stop: 1 bit

Flow control: none

Transmit delay

0 msec/char 0 msec/line

ii. For Putty (Serial):



PuTTY Configuration

Options controlling local serial lines

Select a serial line

Serial line to connect to: COM4

Configure the serial line

Speed (baud): 115200

Data bits: 8

Stop bits: 1

Parity: None

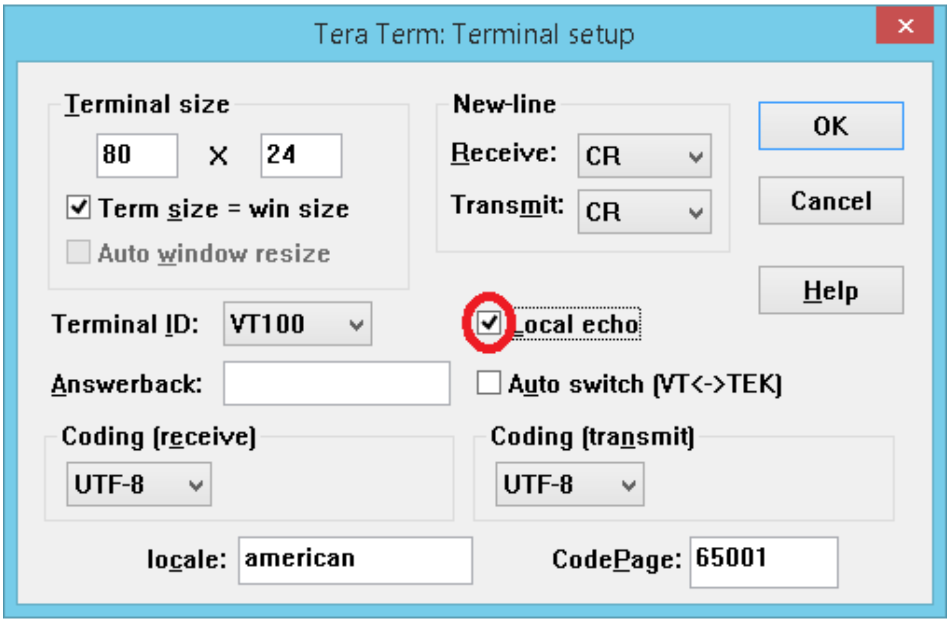
Flow control: XON/XOFF

iii. Configure and open minicom using following command:

```
$ sudo minicom -D /dev/ttyUSB1
```

b. Enable the “Local echo” on the terminal settings (as shown below:

i. For Teraterm (Setup -> Terminal):



Tera Term: Terminal setup

Terminal size: 80 × 24

☑ Term size = win size

□ Auto window resize

New-line

Receive: CR

Transmit: CR

Terminal ID: VT100

Answerback:

☑ Local echo

□ Auto switch (VT<->TEK)

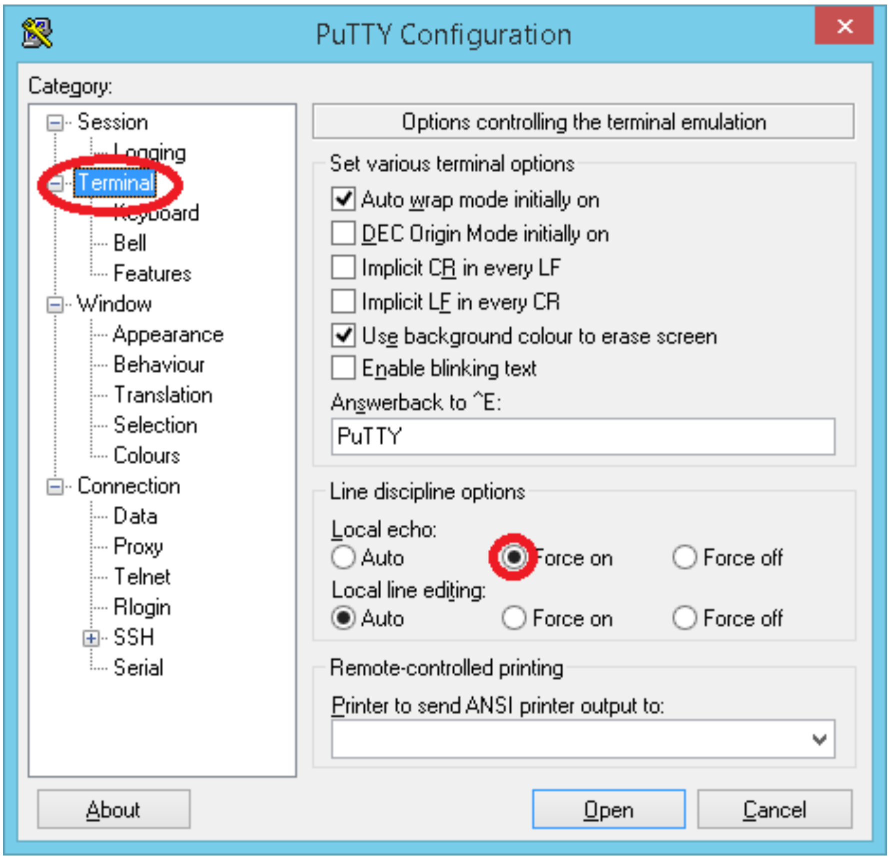
Coding (receive): UTF-8

Coding (transmit): UTF-8

locale: american

CodePage: 65001

ii. For Putty (Terminal):



PuTTY Configuration

Options controlling the terminal emulation

Set various terminal options

☑ Auto wrap mode initially on

□ DEC Origin Mode initially on

□ Implicit CR in every LF

□ Implicit LF in every CR

☑ Use background colour to erase screen

□ Enable blinking text

Answerback to ^E: PuTTY

Line discipline options

Local echo: ○ Auto ● Force on ○ Force off

Local line editing: ● Auto ○ Force on ○ Force off

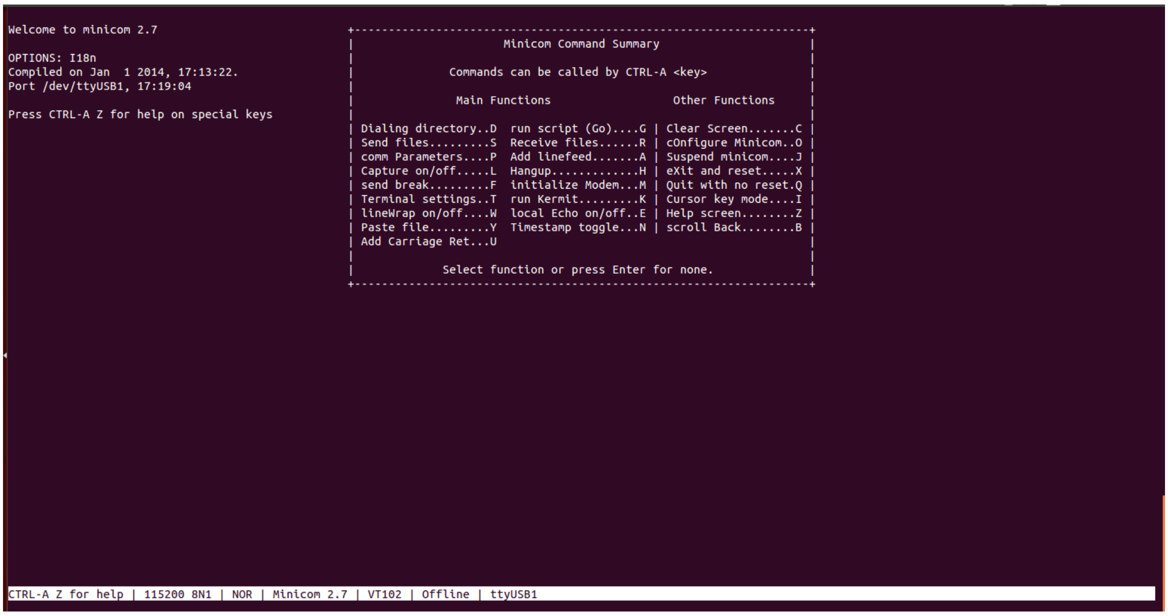
Remote-controlled printing

Printer to send ANSI printer output to:

Note: For Putty the Local line editing option value must be "Auto".

iii. For Minicom:

- For Teraterm Press following key Ctrl+A and Z it will give output similar to following screenshot:



```
Welcome to minicom 2.7

OPTIONS: I18n
Compiled on Jan  1 2014, 17:13:22.
Port /dev/ttyUSB1, 17:19:04

Press CTRL-A Z for help on special keys
```

```
+-------------------------------------------------------------------+
|                      Minicom Command Summary                      |
|                                                                   |
|              Commands can be called by CTRL-A <key>               |
|                                                                   |
|               Main Functions                  Other Functions    |
|                                                                   |
| Dialing directory..D  run script (Go)....G | Clear Screen.......C |
| Send files.........S  Receive files......R | cOnfigure Minicom..O |
| comm Parameters....P  Add linefeed.......A | Suspend minicom....J |
| Capture on/off.....L  Hangup.............H | eXit and reset.....X |
| send break.........F  initialize Modem...M | Quit with no reset.Q |
| Terminal settings..T  run Kermit.........K | Cursor key mode....I |
| lineWrap on/off....W  local Echo on/off..E | Help screen........Z |
| Paste file.........Y  Timestamp toggle...N | scroll Back........B |
| Add Carriage Ret...U                                              |
|                                                                   |
|             Select function or press Enter for none.              |
+-------------------------------------------------------------------+
```

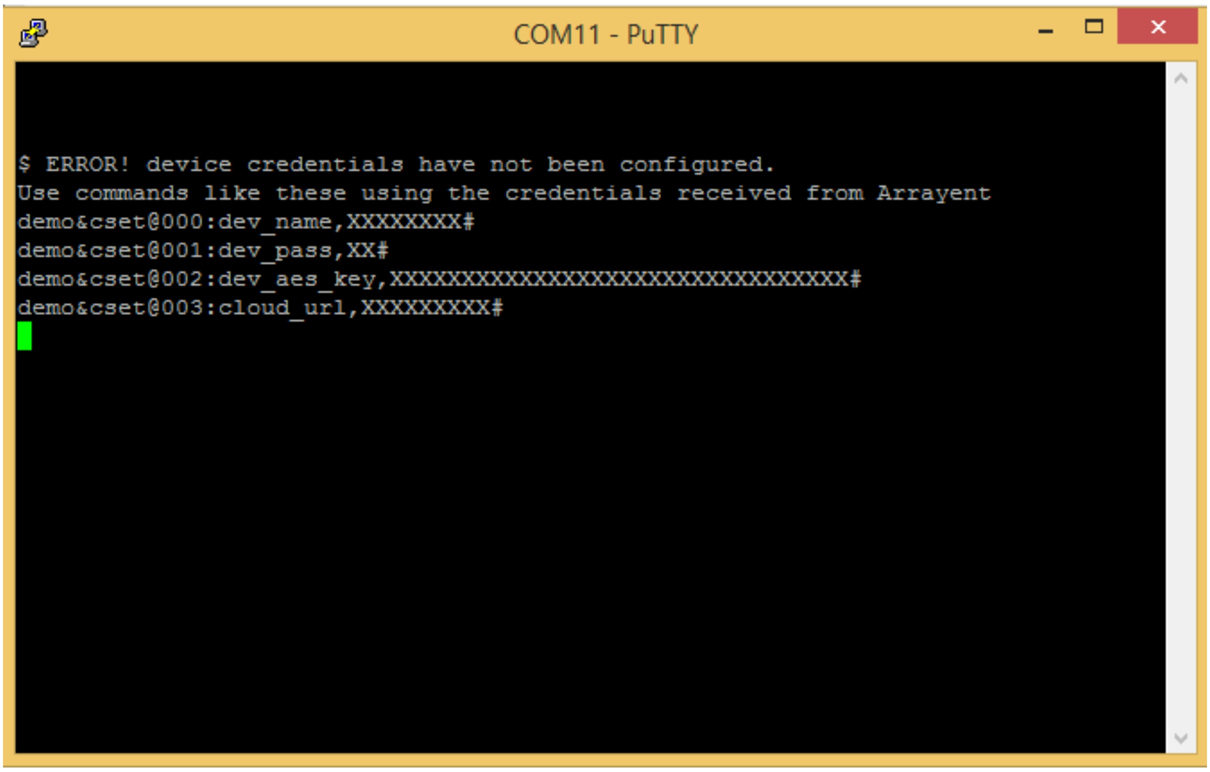
```
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7 | VT102 | Offline | ttyUSB1
```

- Select local Echo on by pressing E

c. Connect the board to the computer using mini USB cable.

d. Open the terminal and Press the RESET button.

e. If the Serial Port settings are correct you will see the output similar to the below screenshot:



```
$ ERROR! device credentials have not been configured.
Use commands like these using the credentials received from Arrayent
demo&cset@000:dev_name,XXXXXXXX#
demo&cset@001:dev_pass,XX#
demo&cset@002:dev_aes_key,XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX#
demo&cset@003:cloud_url,XXXXXXXXX#
```

6. Understanding Application State

Following table mentions the LEDs indication for each application state:

Application state	LED1	LED2
Running	ON	ON
Soft AP	ON	OFF
Init	ON	BLINK
Connect to Cloud	BLINK	ON
Connect to AP	OFF	ON
WPS	BLINK	OFF
ERROR condition	BLINK	BLINK
Update	OFF	BLINK
(Power down)	OFF	OFF

You will need to refer this table frequently in order to find what is the application state indicated by the LEDs' state.

7. Configure WiFi router credentials

Note: Please type console commands manually on console to configure WiFI router credentials and application credentials.

a. **Using Web Browser**

i. Connect the development board to the computer using mini USB cable.

ii. Open the terminal program.

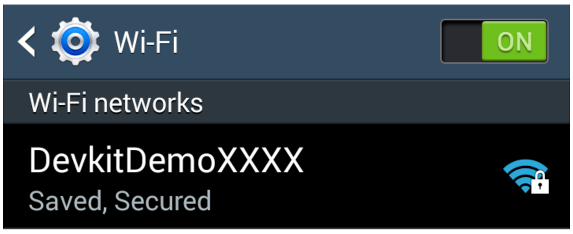
iii. Check that the application is in "Soft AP" state from the LED state.

iv. Application goes to "Soft AP" state if WiFi router credentials are not previously configured.

v. If application is not in "Soft AP" state press SW2 to go to "Soft AP" state.

vi. Check that the application is in "Soft AP" state from the LED state.

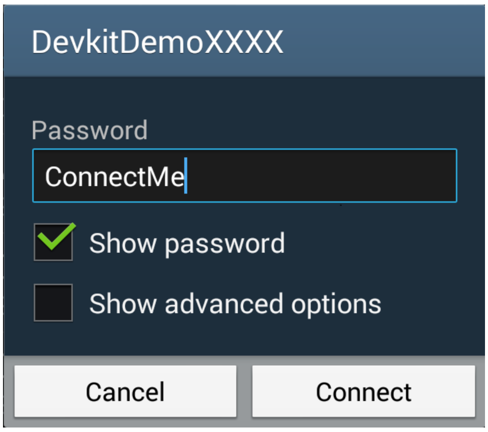
vii. Scan for Wi-Fi devices on your device(e.g. Mobile, laptop).



Wi-Fi networks

DevkitDemoXXXX
Saved, Secured

viii. Connect to “DevkitDemoXXXX” Wi-Fi AP using “ConnectMe” as a passphrase.(If device is already configured with device code then last 4 character will be replace by last 4 digit of device code).



DevkitDemoXXXX

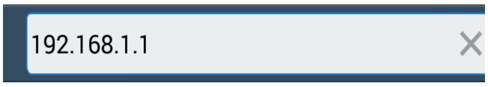
Password

ConnectMe

☑ Show password

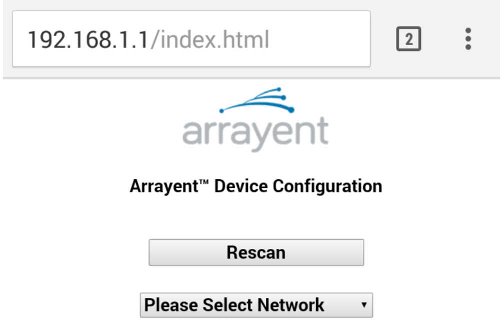
□ Show advanced options

ix. Open web browser and type “192.168.1.1” in URL bar.



192.168.1.1

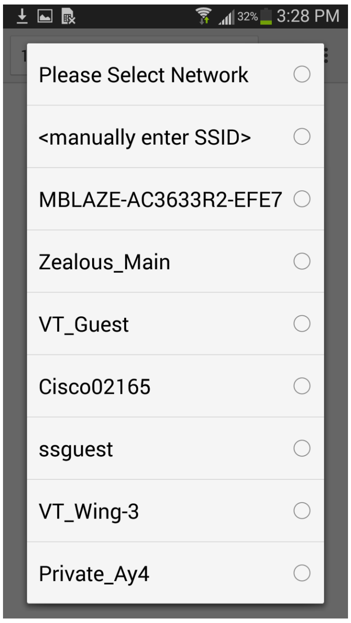
x. You will see the webpage similar to following screenshot:



192.168.1.1/index.html

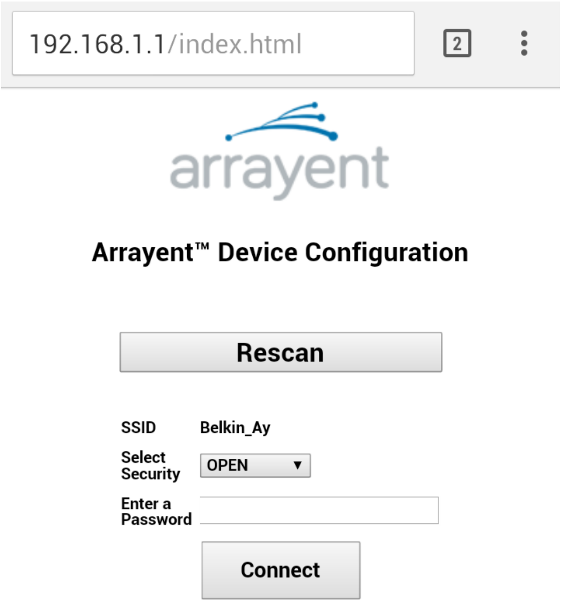
Arrayent™ Device Configuration

xi. Select the Wi-Fi AP using “Please Select Network” scroll down list. If somehow you are not able to see your Wi-Fi AP in the scroll down list then Select “<manually enter SSID>” Option and enter Wi-Fi credential manually.



- Please Select Network ○
- <manually enter SSID> ○
- MBLAZE-AC3633R2-EFE7 ○
- Zealous_Main ○
- VT_Guest ○
- Cisco02165 ○
- ssguest ○
- VT_Wing-3 ○
- Private_Ay4 ○

xii. Once SSID is selected or manually entered you will see the output similar to following screenshot:



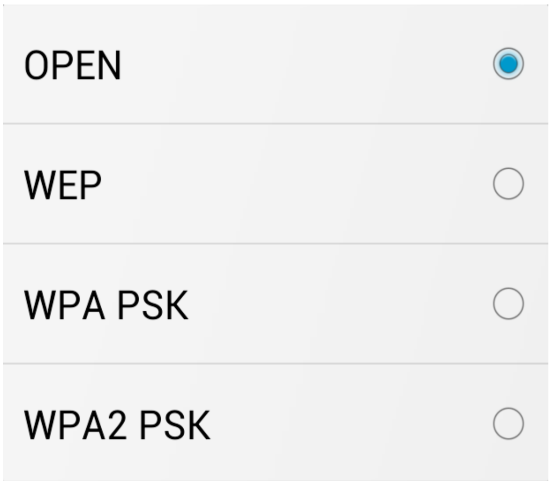
Arrayent™ Device Configuration

SSID Belkin_Ay

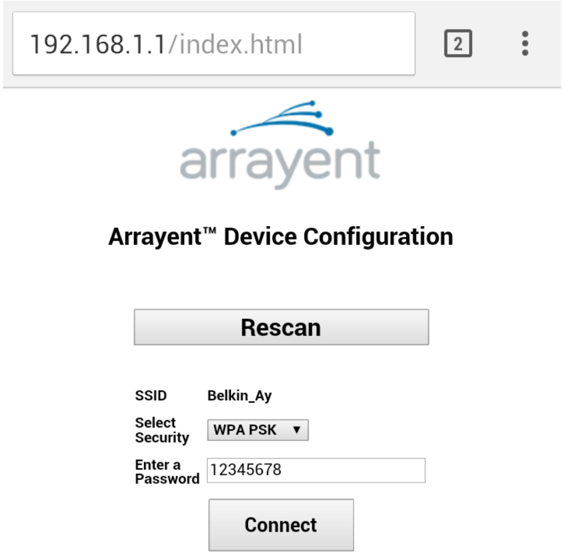
Select Security OPEN ▼

Enter a Password

i. Select Wi-Fi AP security using “Select Security” scroll down list



ii. Enter passphrase in “Enter a Password” Textbox.



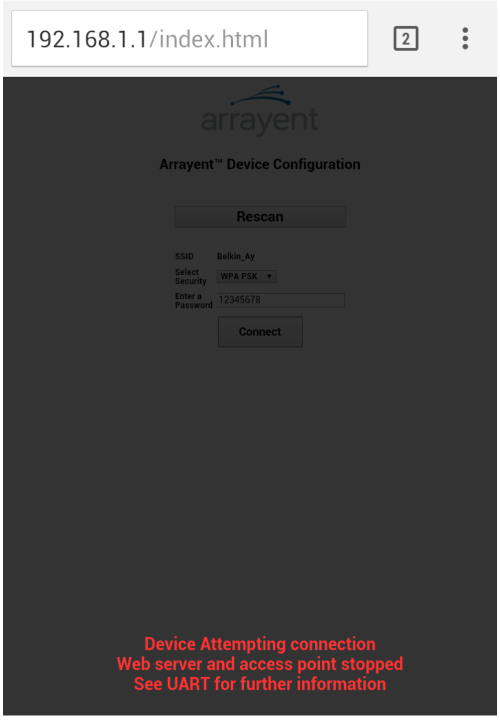
Arrayent™ Device Configuration

SSID Belkin_Ay

Select Security WPA PSK ▼

Enter a Password 12345678

iii. Click on Connect button.



192.168.1.1/index.html

Arrayent™ Device Configuration

SSID Belkin_Ay

Select Security: WPA PSK

Enter a Password: 12345678

Device Attempting connection
Web server and access point stopped
See UART for further information

iv. On successful connection with your WiFi router you will see the application switching its state to "Connect to Cloud" from LED state.

b. **Using WPS**

i. Connect the development board to the computer using mini USB cable.

ii. Open the terminal program.

iii. Check that the application is in "Soft AP" state from the LED state.

Application goes to "Soft AP" state if WiFi router credentials are not previously configured.

iv. If application is not in "Soft AP" state press SW2 to go to "Soft AP" state.

v. Check that the application is in "Soft AP" state from the LED state.

vi. Push the SW1 push button.

vii. Check that the application is in "WPS" state from the LED state.

viii. Press WPS button on your WiFi router.

ix. On successful connection with your WiFi router you will see the application switching its state to "Connect to Cloud" from LED state.

c. **Using console commands**

Note:- Please configure “wpassphrase” after configuring “ssid” and “wsecurity”.

i. Follow steps i - ii from section a. Using DevKit Smartphone app.

ii. Enter the following commands on the terminal console to configure your WiFi router credentials:

```
demo&cset@1:ssid,<your_ssid>#
demo&cset@2:wsecurity,<security_level>#
demo&cset@3:wpassphrase,<your_passphrase>#
```

where, security_level can have one of the following values:

- 0 - OPEN

- 1 - WEP
- 2 - WPA_PSK
- 3 - WPA2_PSK

iii. Reset the board by pressing RESET push button.

iv. On successful connection with your WiFi router you will see the application switching its state to "Connect to Cloud" from LED state

8. Setup ACA configuration and connect to cloud

a. Please make sure that you have already configured the WiFi router credential using one of the methods described in section 6. Configure WiFi router credentials.

b. Check that the sample application is in "Connect to Cloud" state from the LED state.

c. Connect the board to the computer using mini USB cable.

d. Open the terminal program.

e. Enter the following commands on the terminal console to configure ACA device credentials provided to you in your DevKit credentials email:

```
demo&cset@001:dev_name,<device-name>#
demo&cset@002:dev_pass,<device-password>#
demo&cset@003:dev_aes_key,<device-aes-key>#
demo&cset@004:prod_id,<product-id>#
demo&cset@005:prod_aes_key,<product-aes-key>#
```

f. On successful ACA configuration the application will try to connect to the Arrayent cloud.

g. To configure arrayent cloud other than default value, Enter following command on the terminal console:

```
demo&cset@004:cloud_url,<load-balancer>#
```

h. On successful cloud connection the sample application will switch to "Running" state and send application attributes' state to the cloud. You can check application state from LED state.

You have now successfully completed the sample application setup required to connect to the cloud.

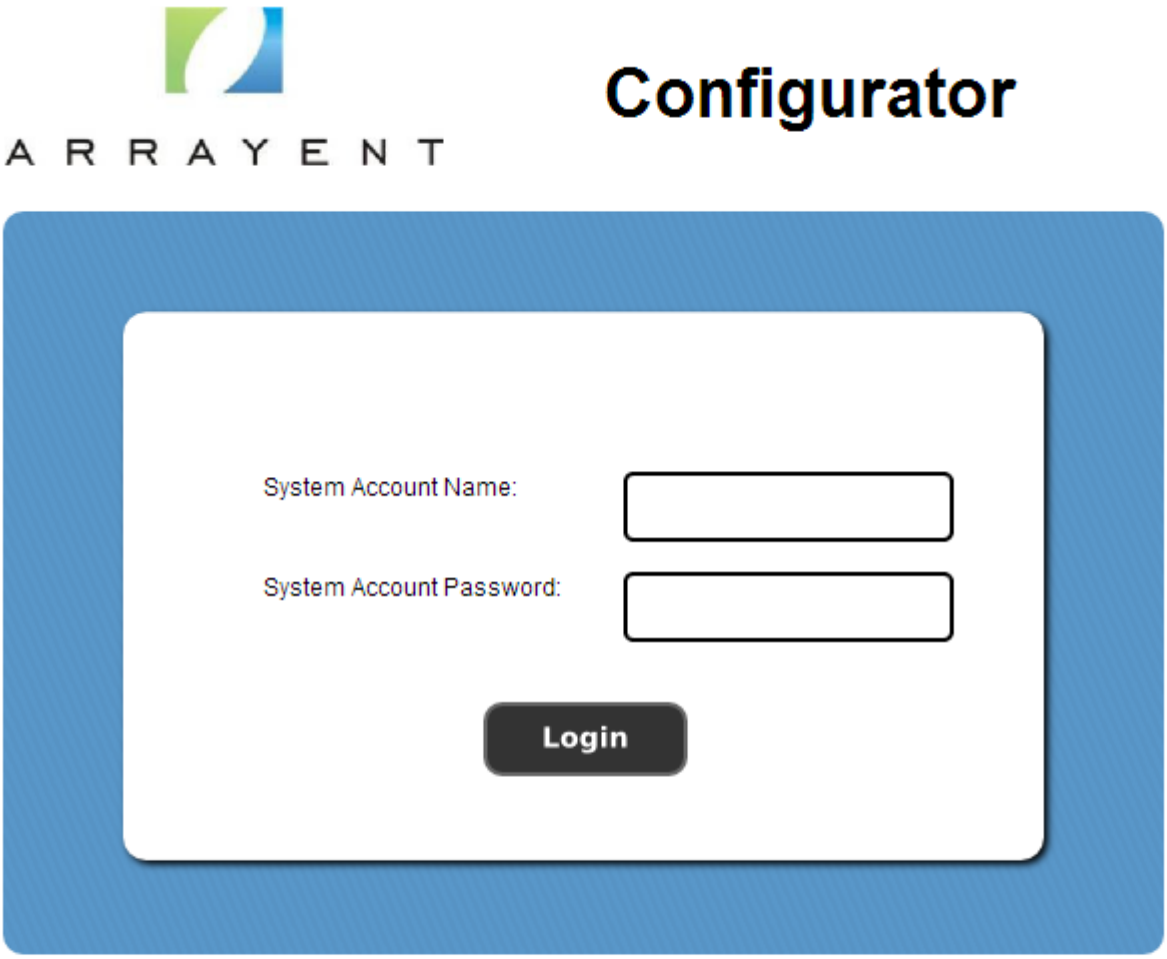
Chapter 4: Configuring the Arrayent Cloud

In this section you use the Arrayent Configurator web application to create a device data model in the Arrayent Cloud. Then you create a Customer Account in the Arrayent Cloud, which you will use later on to monitor and control your device.

1. Logging In to the Configurator

Follow the instructions below to learn how to log in to the Arrayent Configurator web application.

a. Open a web browser and go to https://devkit-api.arrayent.com:8081/Configurator unless Arrayent has instructed you otherwise.



b. Enter your System Account credentials into the login form. These credentials are listed in the email you received from Arrayent.

c. Click Login.

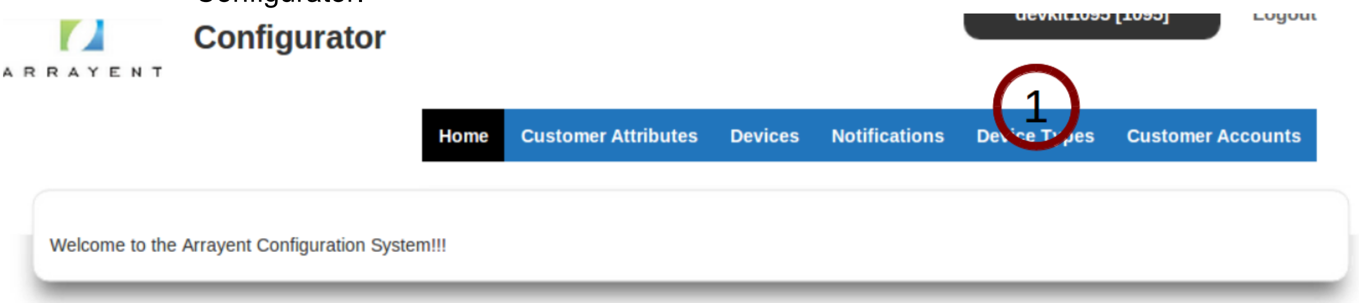
2. Creating a Device Data Model

In this section you learn how to create a device data model using the Arrayent Configurator web application. In Arrayent terminology a device data model is referred to as a “Device Type”.

Creating the "Devkit_app" Device Type

The name of the Device Type we are creating will be called "Devkit_app".

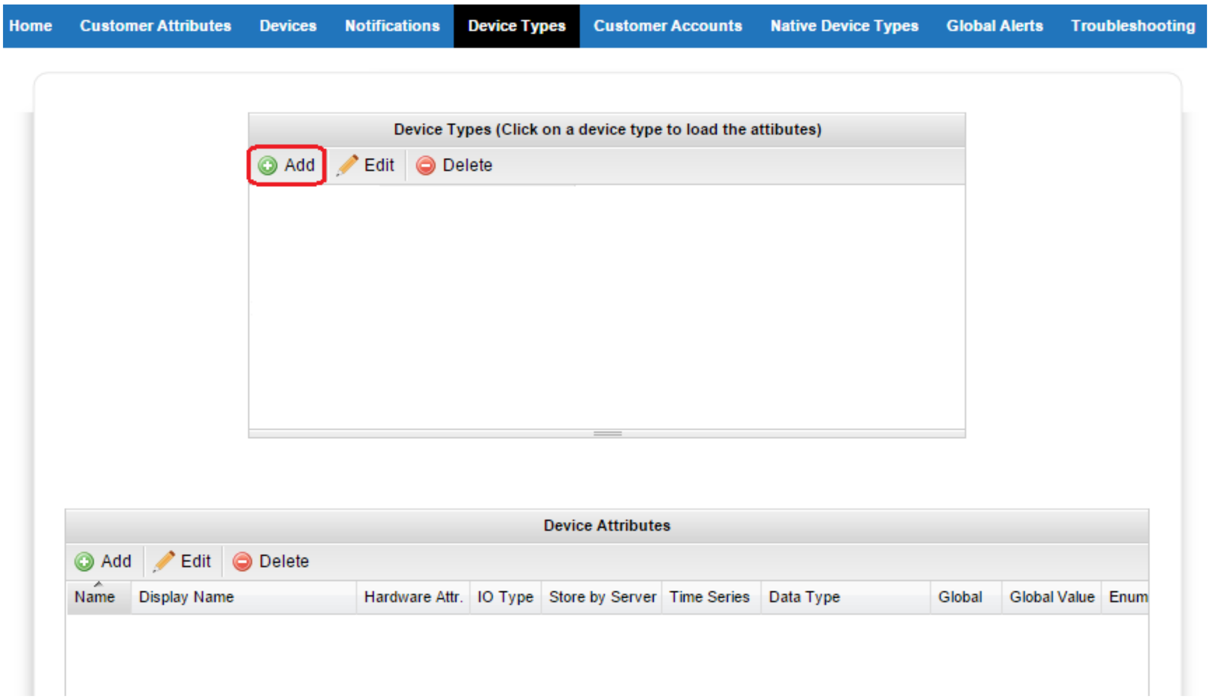
a. Click the **Device Types** tab to go to the Device Types page of the Configurator.



Welcome to the Arrayent Configuration System!!!

This page enables you to create, modify, and delete device data models.

b. Click **Add**.



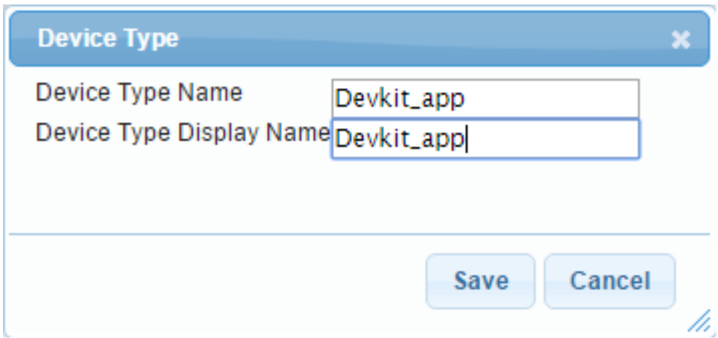
Device Types (Click on a device type to load the attibutes)

Device Attributes

Name	Display Name	Hardware Attr.	IO Type	Store by Server	Time Series	Data Type	Global	Global Value	Enum

c. For Device Type Name enter Devkit_app.

d. For Device Type Display Name enter Devkit_app.



Device Type Name Devkit_app

Device Type Display Name Devkit_app

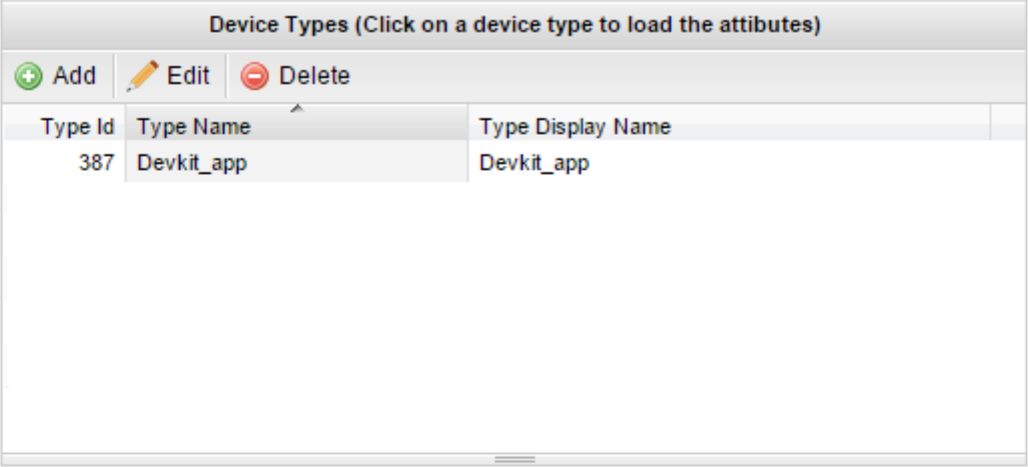
e. Click **Save**.

Creating Device Attributes

In this section you use the Configurator to define Device Attributes, called SW1_count, Led1_cmd and Led1_State.

a. Look at the Device Types page of the Configurator. You should see two tables. The top table is titled Device Types. The bottom is titled Device Attributes.

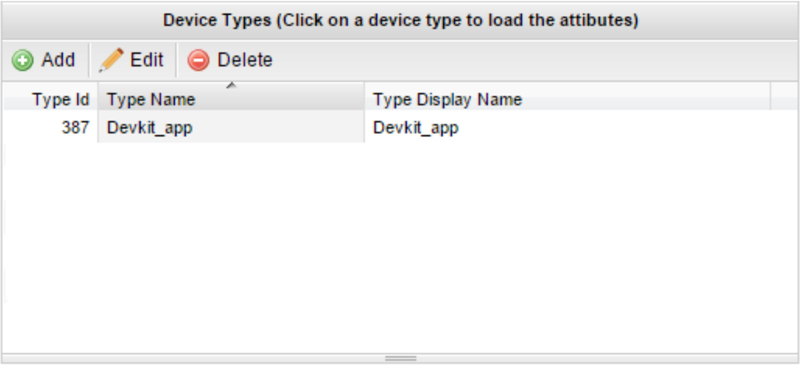
b. Click on the row containing our new Device Type, Devkit_app. This should highlight the row in blue



Device Types (Click on a device type to load the attibutes)

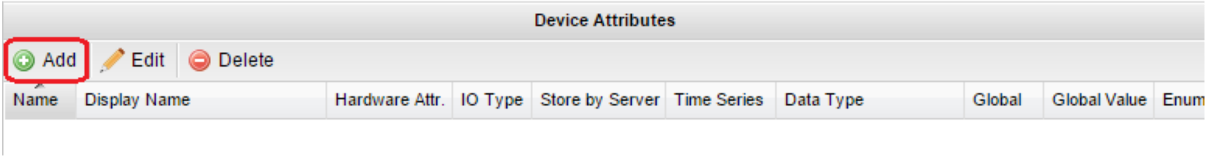
Type Id	Type Name	Type Display Name
387	Devkit_app	Devkit_app

c. In the bottom table titled **Device Attributes** click the **Add** button.



Device Types (Click on a device type to load the attibutes)

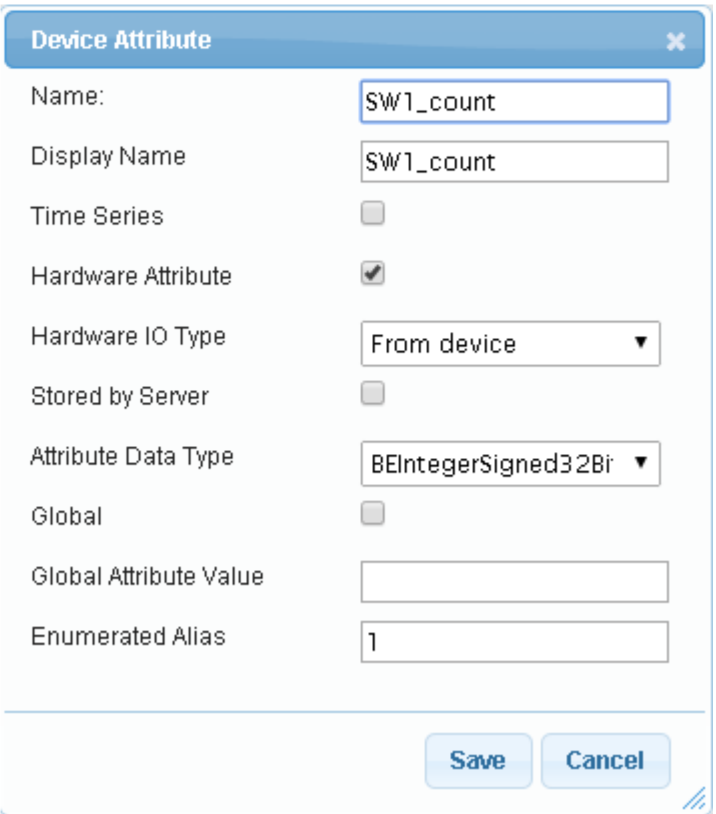
Type Id	Type Name	Type Display Name
387	Devkit_app	Devkit_app



Device Attributes

Name	Display Name	Hardware Attr.	IO Type	Store by Server	Time Series	Data Type	Global	Global Value	Enum

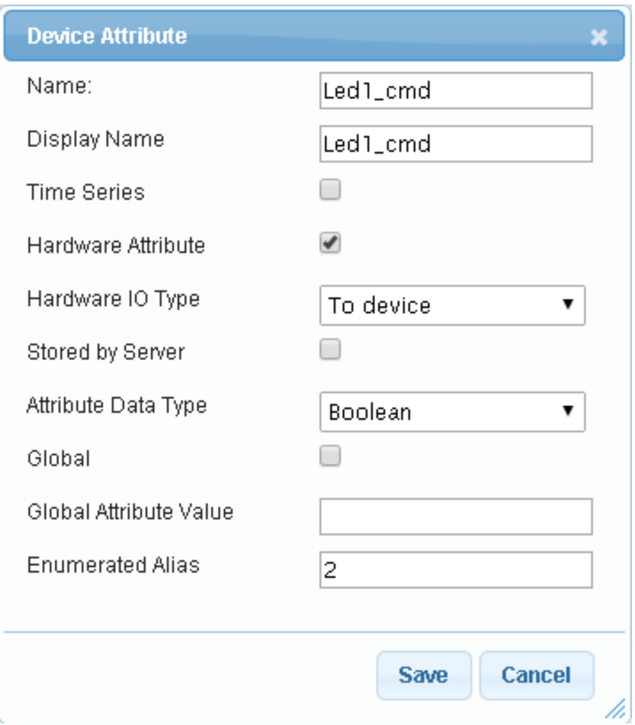
d. Fill out the form so that your new Device Attribute matches the image below. Click **Save** when finished.



Device Attribute

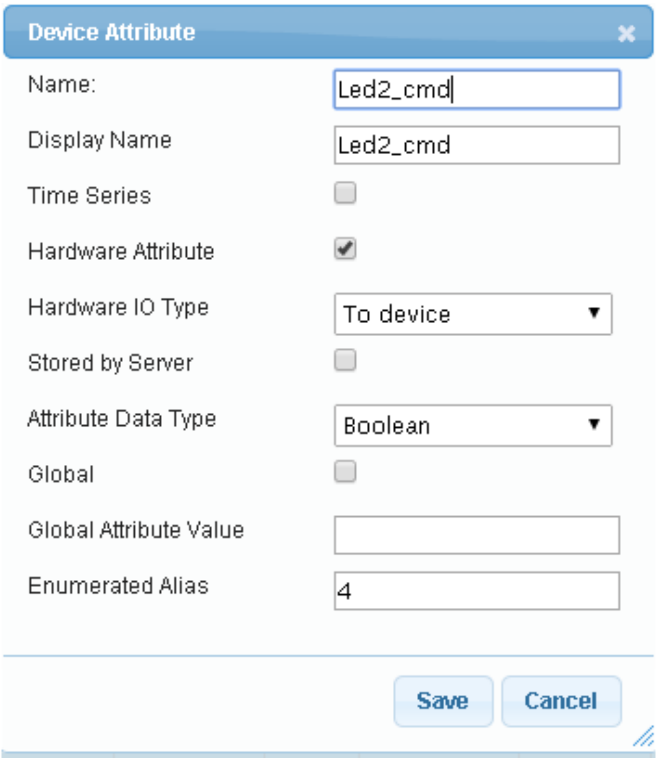
Name:	SW1_count
Display Name	SW1_count
Time Series	□
Hardware Attribute	☑
Hardware IO Type	From device
Stored by Server	□
Attribute Data Type	BEIntegerSigned32Bi
Global	□
Global Attribute Value	
Enumerated Alias	1

e. Now repeat the process to create another Device Attribute that matches the images below:



Device Attribute

Name:	Led1_cmd
Display Name	Led1_cmd
Time Series	□
Hardware Attribute	☑
Hardware IO Type	To device
Stored by Server	□
Attribute Data Type	Boolean
Global	□
Global Attribute Value	
Enumerated Alias	2



Name:	Led2_cmd
Display Name	Led2_cmd
Time Series	□
Hardware Attribute	☑
Hardware IO Type	To device
Stored by Server	□
Attribute Data Type	Boolean
Global	□
Global Attribute Value	
Enumerated Alias	4

See Device Types Guide for a description of how these fields affect a Device Attribute definition.

3. Creating a Customer Account for Monitor and Control

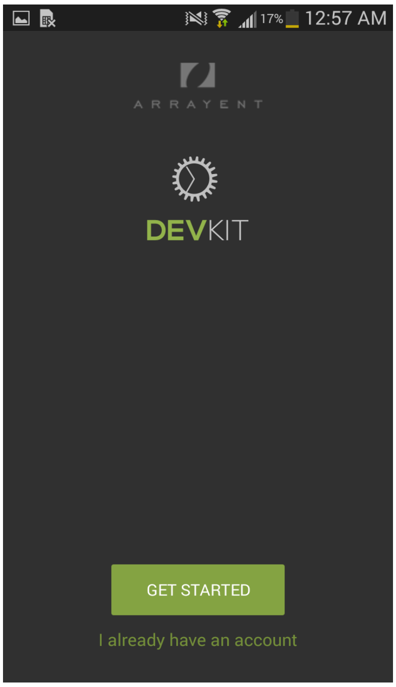
Follow the instructions in the next section to learn how to create a new Customer Account in the Arrayent Cloud. Later on you will log in to the Arrayent Cloud as this Customer Account and then monitor and control your evaluation board.

a. Using Configurator:

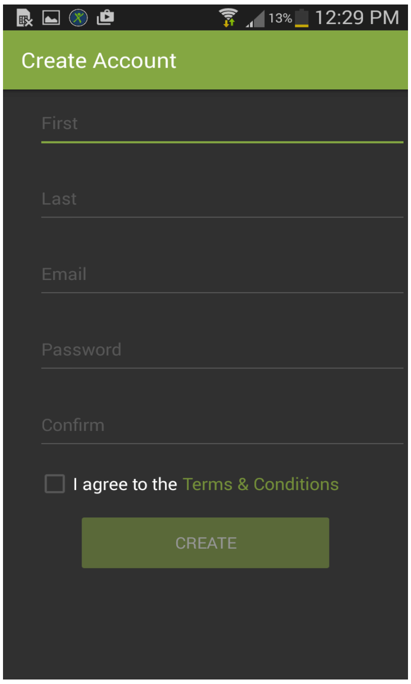
i. Enter the username of the new account in the text field next to **Customer Account Name**.
ii. Select the checkbox. This will enable you to enter text in the textbox next to **Customer Account Password**.
iii. Enter the password of the new account.
iv. Leave **Master Account Name** empty.
v. Enter values for the other user attributes as needed. All attributes other than username and password are optional.
vi. Click **Save**.

b. Using Devkit Android Application:

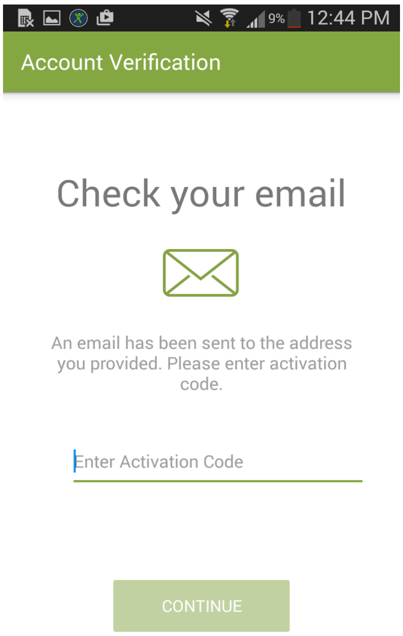
i. “Devkit” android application Home page will look similar to the below screenshot:



ii. Click on **GET STARTED** button. you will see the output similar to the below screenshot:



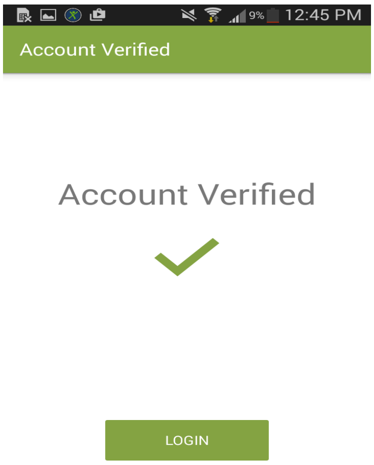
iii. Fill account details and tick on **I agree to the Terms & Conditions** box to create account on Arrayent cloud. Click on **CREATE** button. After that you will see the output similar to the below screenshot:



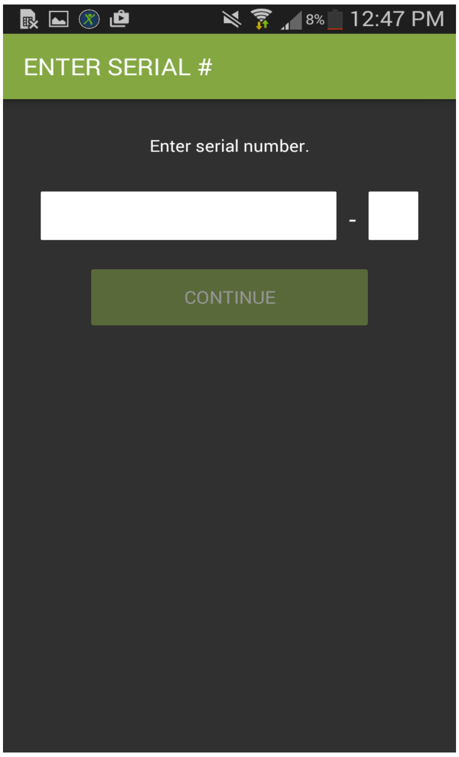
Check your email

An email has been sent to the address you provided. Please enter activation code.

iv. Enter verification code send to your email ID entered during create account details. Click on **CONTINUE** button. On successful account creation you will see the output similar to the below screenshot:

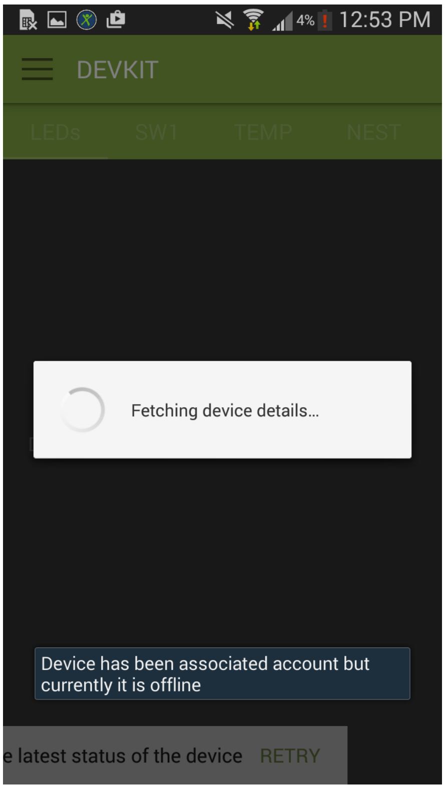


Account Verified



Enter serial number.

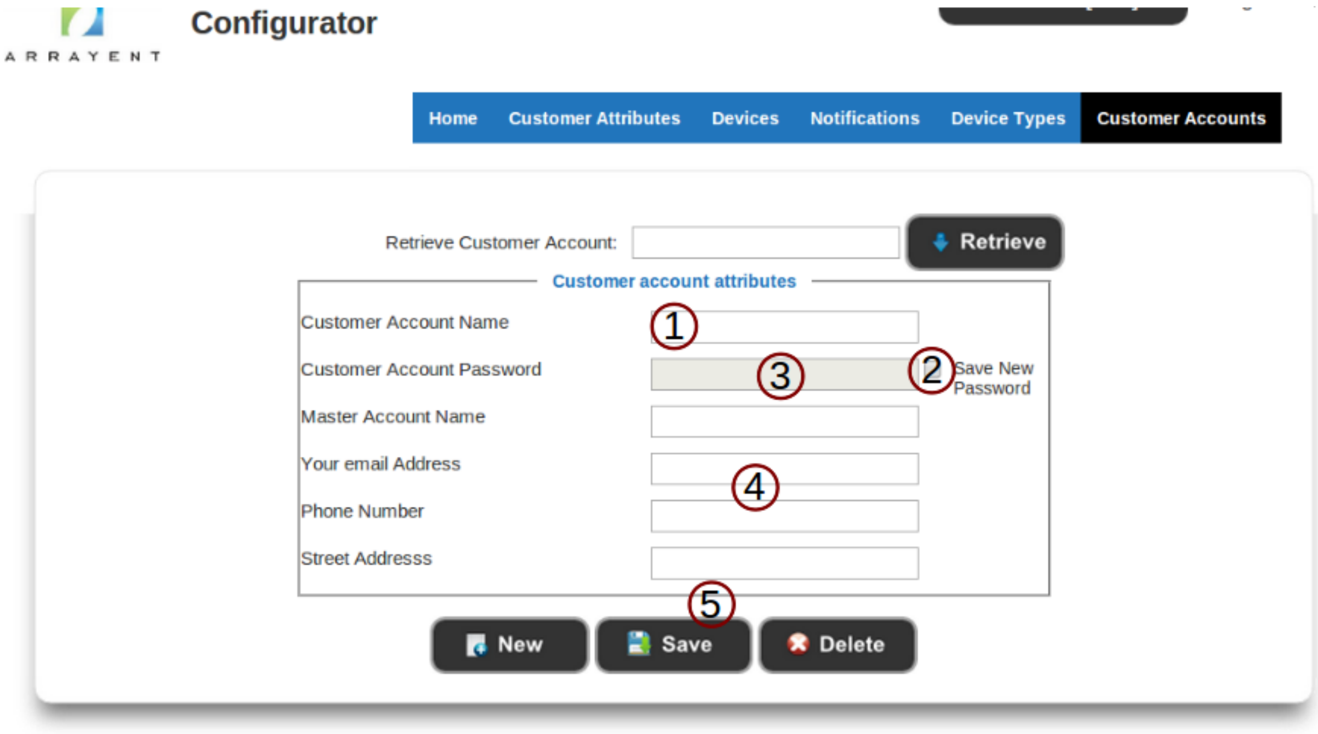
v. Enter Device ID in first Box and Device Password in another box. click on “CONTINUE” to Add device to your account. On successful Association of device with account you will see the output similar to the below screenshot:



Fetching device details...

Device has been associated account but currently it is offline

e latest status of the device



When you log in to the Utility app as the new Customer Account, you will need to provide the App ID (A.K.A. System Account ID) of the Customer Account. Customer Accounts inherit the App ID of the System Account that created the account. So if a System Account named “admin” with App ID “10” creates a Customer Account named “user”, the App ID of “user” will also be “10”.

Chapter 5: Monitoring and Controlling the DevKit board

Using Arrayent Utility Web application

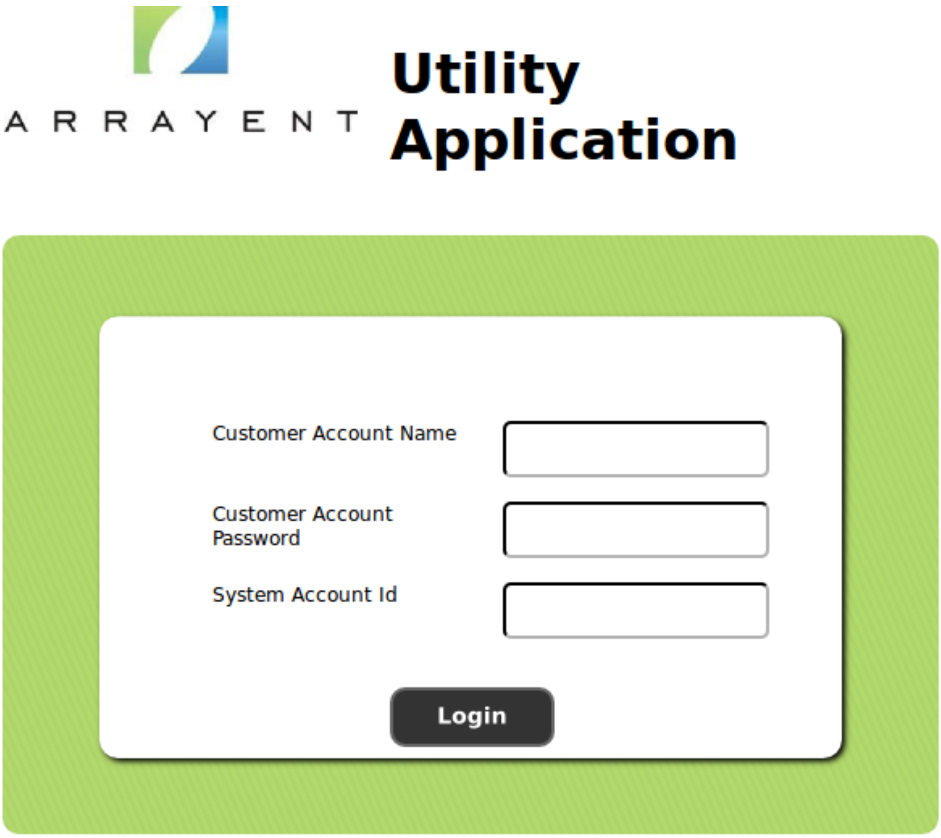
In this section you use the Arrayent Utility web application to monitor and control your evaluation board.

a. Logging In to the Utility Application

Follow the instructions below to learn how to log in to the Arrayent Utility web application.

i. Open a web browser and go to the URL below, unless instructed otherwise by Arrayent.
https://devkit-api.arrayent.com:8081/Utility

ii. Log in with the Customer Account credentials that you created earlier with the Configurator.



Customer Account Name

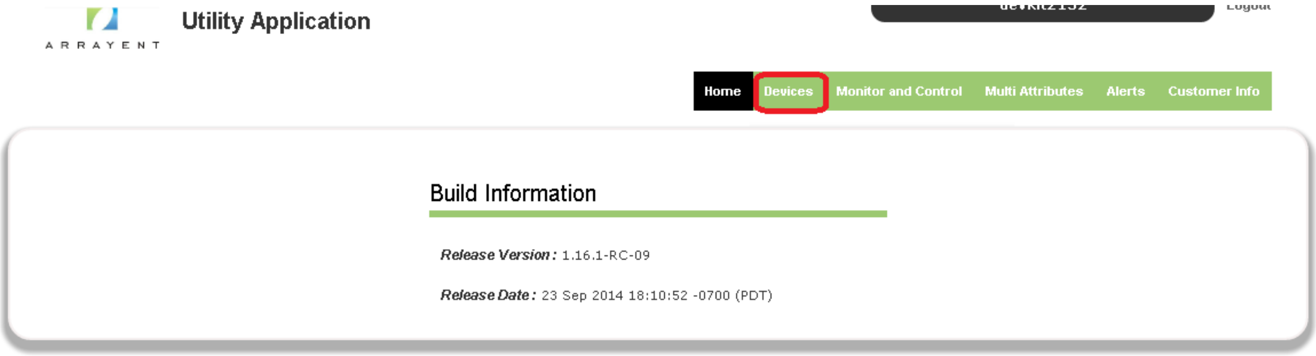
Customer Account Password

System Account Id

b. **Adding the device to your account**

In the **Arrayent** Cloud, a device (in this case, your evaluation board) must be owned by a Customer Account before it can be monitored and controlled.

i. Click the **Devices** tab.

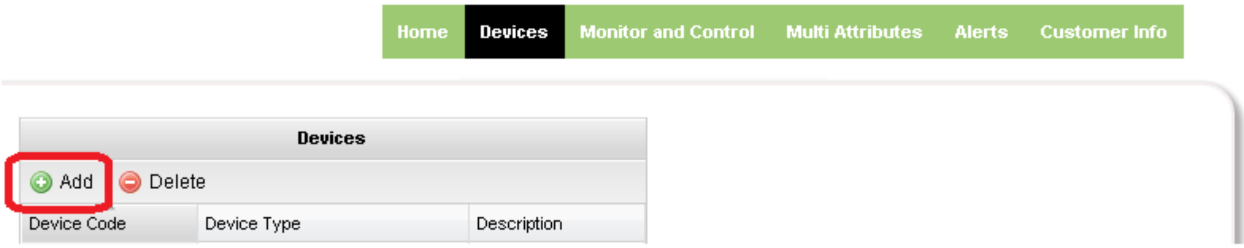


Build Information

Release Version : 1.16.1-RC-09

Release Date : 23 Sep 2014 18:10:52 -0700 (PDT)

ii. Click **Add**.



Device Code	Device Type	Description

iii. For **Device Name** enter the value of device_name in your email from Arrayent.

iv. For **Device Password** enter the value of device_password in your email from Arrayent.
v. For **Device Type** select Devkit_app.
vi. Click **Save**.

The device is now claimed by this Customer Account. No other Customer Account can access this device.

c. Controlling the LED

i. Follow the instructions below to learn how to use the Utility application to remotely control the LED on your evaluation board.
ii. Note the **Device Name** drop down menu. If you had multiple devices you would select your device from this menu.
iii. Click the **Monitor and Control** tab.
iv. Find the row containing the Led1_cmd attribute.
v. Click the ON radio button in the Led1_cmd row.
vi. Look at your evaluation board. The LED1 should be on.
vii. Click the OFF radio button.
viii. Look at your board again. The LED should be off now.
ix. The Led1_State attribute slider bar in the utility app will show the current state of the LED1.

d. Monitoring Push Button Events

In this section you use the Utility application to monitor push button events on your evaluation board.

i. You should still be on the Monitor and Control page of the Utility application. If not, go back to that page now.
ii. Find the row containing the SW1_count attribute.
iii. Press and release the push button on your evaluation board labeled SW1.
iv. You should see the SW1_count increment each time you press SW1 push button.

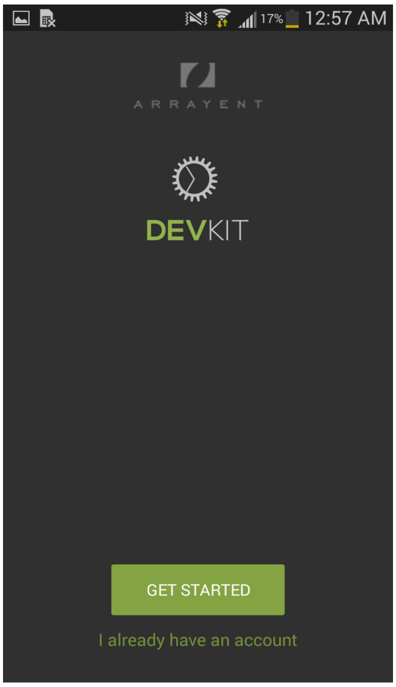
Using Devkit Android Application

In this section you use the Devkit Android application to monitor and control your evaluation board.

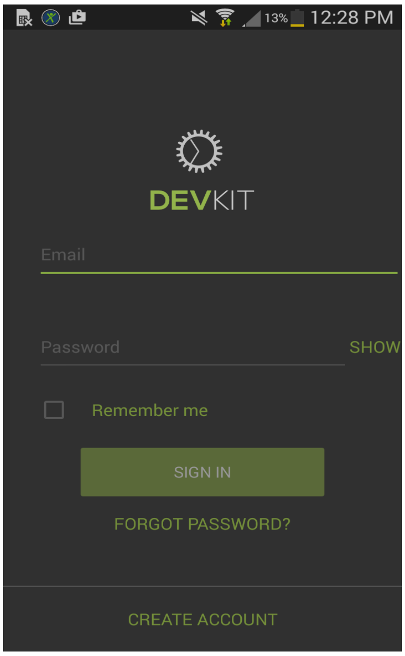
a. Logging In to the Devkit Android Application

Follow the instructions below to learn how to log in to the Devkit application.

i. Open a Devkit application and Click on “I already have account” button.



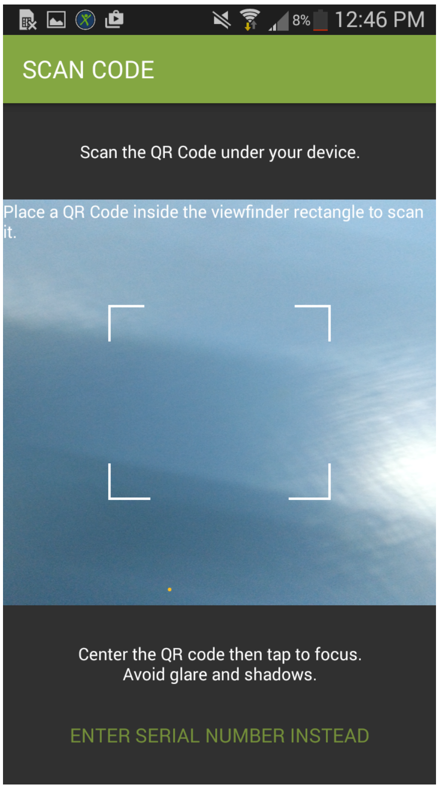
ii. Log in with the Customer Account credentials that you created earlier with the Devkit android application.Click on **SIGN IN** button.



b. **Adding the device to your account**

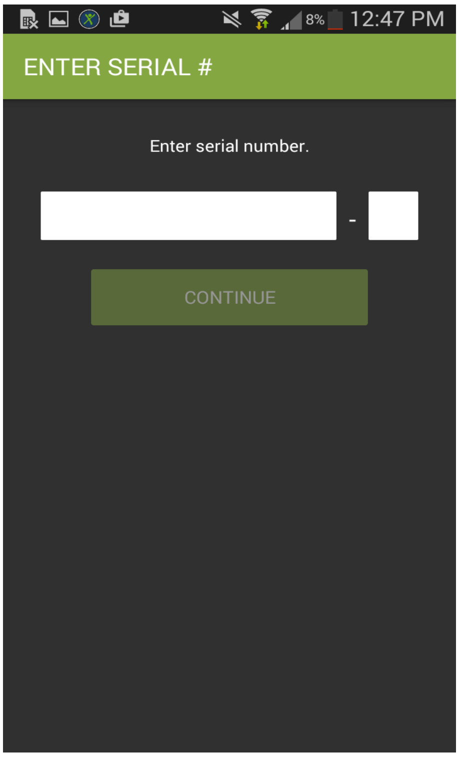
In the **Arrayent** Cloud, a device (in this case, your evaluation board) must be owned by a Customer Account before it can be monitored and controlled.

i. If you don’t have QR code then click on **ENTER SERIAL NUMBER INSTEAD**.

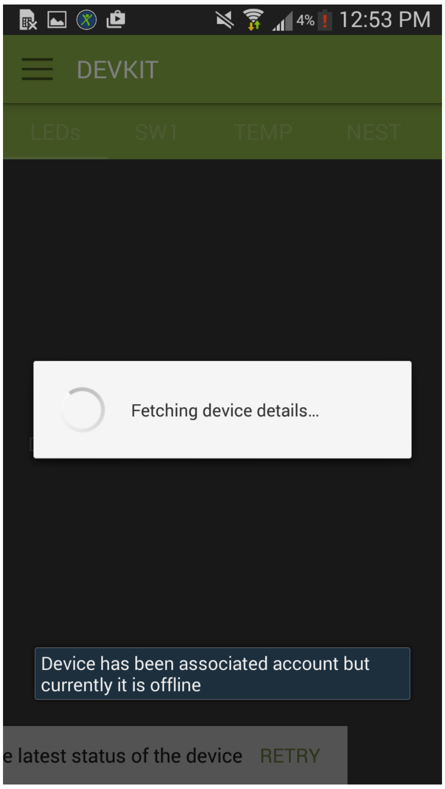


12:46 PM
SCAN CODE
Scan the QR Code under your device.
Place a QR Code inside the viewfinder rectangle to scan it.
Center the QR code then tap to focus.
Avoid glare and shadows.
ENTER SERIAL NUMBER INSTEAD

ii. Enter Device ID in first Box and Device Password in another box. click on “CONTINUE” to Add device to your account.



iii. On successful Association of device with account you will see the output similar to the below screenshot:



Fetching device details...

Device has been associated account but currently it is offline

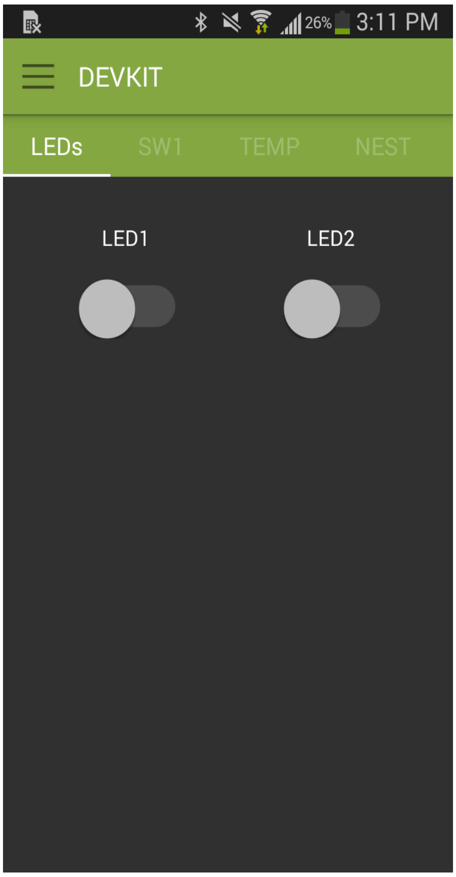
e latest status of the device RETRY

The device is now claimed by this Customer Account. No other Customer Account can access this device.

c. Controlling the LED

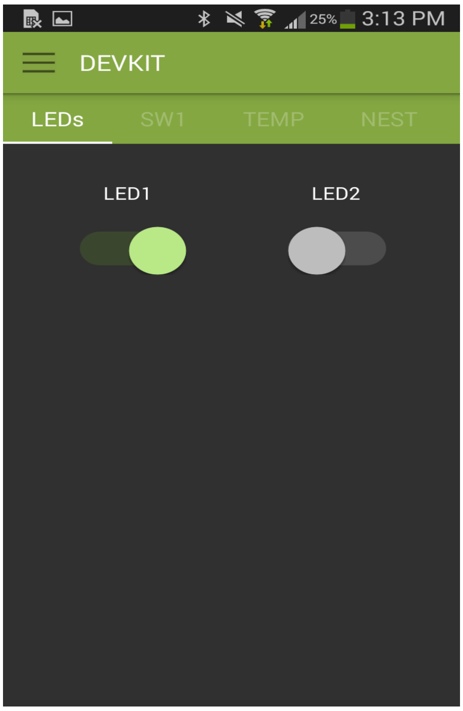
i. Follow the instructions below to learn how to use the Devkit android application to remotely control the LED on your evaluation board.

ii. Once device is online you will see screen similar to following:



iii. Click the LED1 switch.

iv. Look at your evaluation board. The LED1 should be on. Devkit application screen will be similar to following screenshot:



v. Click the LED1 switch again.

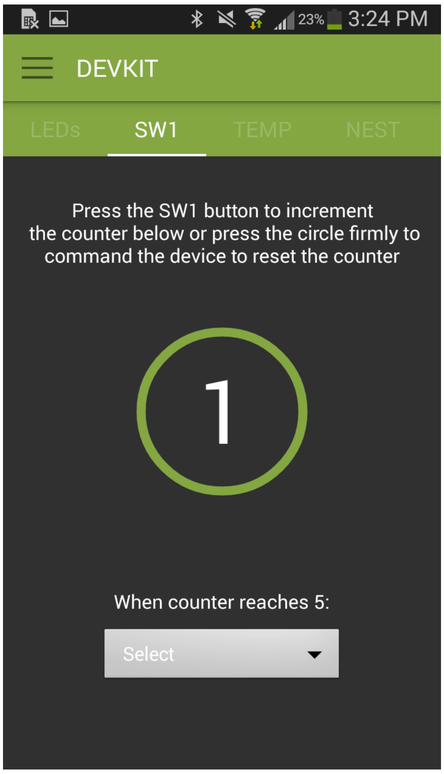
vi. Look at your board again. The LED should be off now.

vii. The Led1_State attribute slider bar in the utility app will show the current state of the LED1.

d. Monitoring Push Button Events

In this section you use the Devkit android application to monitor push button events on your evaluation board.

i. You should still be on LEDs tab of the Devkit android application.
ii. Click on SW1 tab to monitor switch press(SW1).
iii. Press and release the push button on your evaluation board labeled SW1.
iv. You should see the count increment each time you press SW1 push button.



Press the SW1 button to increment the counter below or press the circle firmly to command the device to reset the counter

When counter reaches 5:

Using Console commands to Update Attribute value

a. Monitoring and setting Temperature:

i. ACA must be connected with Arrayent Cloud and sample application should be in Running state (verify from LED state).
ii. Find the row containing Temperature attribute. This attribute gets updated by the application. Please follow Chapter 7 for further details of this attribute.
iii. You can set temperature from console command. Below is example to set Temperature attribute value to 100. Verify the Temperature attribute is updated with value 100.

```
demo&update@02:settemp,100#
```

b. Setting SW1 Count:

i. ACA must be connected with Arrayent Cloud and sample application should be in Running state (verify from LED state).

ii. Find the row containing SW1_count attribute.
You can update the SW1 count from the console commands. Below is the example command to increase push button count. The value of SW1_count attribute will be incremented by 1.

```
demo&update@02:pushButton,1#
```

c. Sending Property message:

i. ACA must be connected with Arrayent Cloud and sample application should be in Running state (verify from LED state).

ii. You can send single property message to Arrayent cloud using console command. Below is the sample command to set value of sys-ping attribute. You can verify that the value of sys-ping is updated to devkit_sample_app.

```
demo&sendkvp@02:sys-ping,devkit_sample_app#
```

Chapter 6: Next Steps

Congratulations! You just learned how to use the Arrayent Connect Platform to monitor and control a connected device.

- Read the Architecture Overview to understand how the Arrayent Connect Platform enables connectivity in your devices.
- Check out the source code of the Arrayent sample application to learn how to use the ACA in your own application. Relative to the file that you downloaded from Arrayent, the sample application directory is located at <Release Package>/app/src/devkit_app. Check out the Makefile (arrayent_demo.mk) and the application source code to learn how to include the ACA library into your own project.
- Read the ACA API Reference for complete details on the ACA's features.
- Read the ACA Developer's Guide for higher-level guides on implementing a connected device application that uses the ACA.

Chapter 7: Appendix

1. Sample application Data Model

We have only added a part of the device attributes supported by the sample application on the "Devkit_app" data model. The sample application supports attributes listed in the following table:

Attribute name	Description	Type	Direction	Default Value Upon reset
ConnectionType	Reports if the application is connected to the cloud using TCP or UDP	String (TCP, UDP)	To Cloud	UDP
SW1_count	Reports the number of times that SW1 has been pressed since it was reset	Integer32	To Cloud	0
SW1_count_reset_cmd	If this is TRUE the embedded app resets the SW1_count to 0 and then modifies this attribute to FALSE	BOOLEAN	From Cloud	
Led1_cmd	Command to set Led1 to ON or OFF	Boolean (ON, OFF)	From Cloud	
Led2_cmd	Command to set Led1 to ON or OFF	Boolean (ON, OFF)	From Cloud	
Led1_state	Reports the state of LED1	Boolean (ON, OFF)	To Cloud	OFF
Led2_state	Reports the state of LED3	Boolean (ON, OFF)	To Cloud	OFF
Temperature	Reports the temperature read from the temperature sensor in 1/100s of a degree (e.g. 7520 means 75.20 degrees)	Integer16	To Cloud	0xFFFF
TemperatureIncrement	Set by cloud to determine upon what change in temperature should the application report a change of temperature to the cloud Default is 100 (i.e. every full degree)	Integer	From Cloud	

ReadTimeAndDate_cmd	Command to application to get the time and date	Boolean	From Cloud	
TimeAndDate	Last time and date read from cloud	String	To Cloud	1-Jan-1970
UpdateNow_cmd	Flag indicating to application that it must perform a firmware update	BOOLEAN	From Cloud	
UpdateURL	URL of where to find updated software	String (URL format)	From Cloud	
RefreshAllAttributes_cmd	Command from cloud to force application to send all the “To Cloud” attributes in the data model using the API ArrayentSetMultiAttribute()	Boolean	From Cloud	

SetRecvMultiAttribute_cmd	Cloud can set this to make application change from/to SingleAttribute mode to/from MultiAttribute mode	Boolean
		FALSE=Single
		TRUE=Multi
			From Cloud	

MultiAttribute_state	Reports if the application is using single or multi attribute to receive properties from the cloud	Boolean
		FALSE=Single
		TRUE=Multi
			To Cloud	0

Application_state	Description of the state in which the application is currently running. Mostly used to indicate it is going to enter the Soft-AP state or it is Running or it is Upgrading	String	To Cloud	Init
RSSI	Reports the RSSI value to the cloud every 15 minutes	Integer	To Cloud	
app-version	Description of the internal version of the embedded application	String	To Cloud	0.0.0.0
app-timezone ENUM: 206	Contains a value in minutes of the time difference between the device’s local timezone and UTC	Integer	From Cloud	0
app-dstflag ENUM: 207	The daylight savings time flag is a boolean that daylight savings is applicable in the device’s local zone	BOOLEAN	From Cloud	0

sys-version	ACA internal version	String	To Cloud	
sys-ping	ACA internal	String	From Cloud	
sys-pong	ACA internal – reflects what is sent to sys-ping	String	To Cloud	