



**Arrayent DevKit  
Sample Application  
Quick Start Guide  
For TI CC3200**

30 November 2016

# TABLE OF CONTENTS

## **Chapter 1. Overview**

[Audience and Scope](#)

[Prerequisites](#)

[Supported Software and Hardware Platforms](#)

[TI CC3200 Launchpad:](#)

## **Chapter 2. Note on DevKit Credentials and Environment**

[Users Who Receive Explicit Credentials from Arrayent](#)

## **Chapter 3. Sample application Setup**

[Release package Directory Structure](#)

[Steps to load the pre-built devkit application binary](#)

[TI CC3200 Launchpad:](#)

[Setup development environment](#)

[TI CC3200 Launchpad:](#)

[Setup terminal program](#)

[Understanding Application State](#)

[Configure WiFi router credentials](#)

[Using Web Browser](#)

[Using DevKit Smartphone app](#)

[Using WPS](#)

[Using console commands](#)

[Setup ACA configuration and connect to cloud](#)

## **Chapter 4: Configuring the Arrayent Cloud**

[Logging In to the Configurator](#)

[Creating a Device Data Model](#)

[Creating the “Devkit\\_app” Device Type](#)

[Creating Device Attributes](#)

[Creating a Customer Account for Monitor and Control](#)

[Using Configurator:](#)

[Using Devkit Android Application:](#)

## **Chapter 5: Monitoring and Controlling the DevKit board**

[Using Arrayent Utility Web application](#)

[Logging In to the Utility Application](#)

[Adding the device to your account](#)

[Controlling the LED](#)

[Monitoring Push Button Events](#)

[Using Devkit Android Application](#)

[Logging In to the Devkit Android Application](#)

[Adding the device to your account](#)

[Controlling the LED](#)

[Monitoring Push Button Events](#)

[Using Console commands to Update Attribute value](#)

[Monitoring and setting Temperature:](#)

[Setting SW1 Count:](#)

[Sending Property message:](#)

## **Chapter 6: Next Steps**

## **Chapter 7: Appendix**

[Sample application Data Model](#)

## 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 CC3200 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 CC3200 Launchpad:

### Operating System

- Window 8.1

### IAR Embedded Workbench ARM

- Version 7.30.4

### TI SDK

- SDK Version: 1.2.0
- Download TI CC3200 SDK installer from below link: [http://software-dl.ti.com/download/ecs/@BMU7KPXVHI7E9D@FQ85KRTAUAE\\_VTWN/CC3200SDK-1.2.0-windows-installer.exe](http://software-dl.ti.com/download/ecs/@BMU7KPXVHI7E9D@FQ85KRTAUAE_VTWN/CC3200SDK-1.2.0-windows-installer.exe)

### TI Launchpad

- Development Board: CC3200-LaunchXL Rev 4.1: Board version is located on the front side of the board as shown in the below figure.

### Uniflash

- Version: 3.2.0

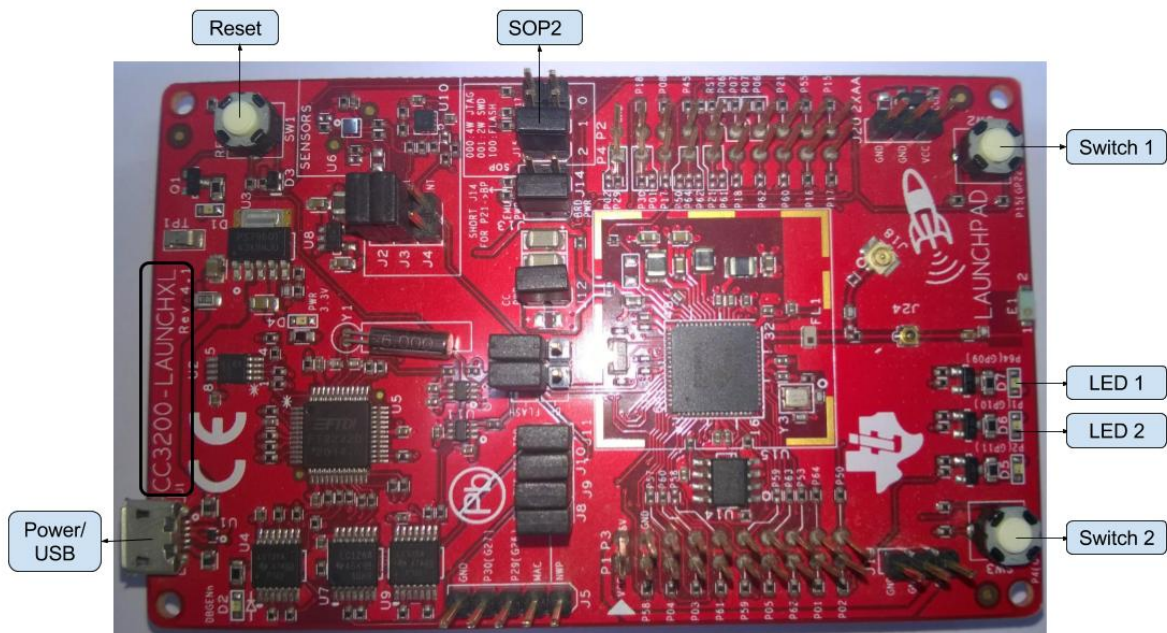


Figure 1: TI Launchpad Front

## **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.

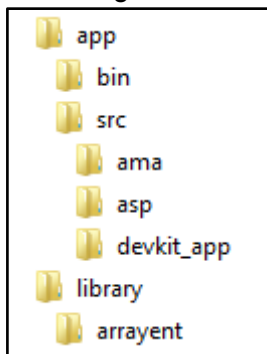
## Chapter 3. Sample application Setup

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:

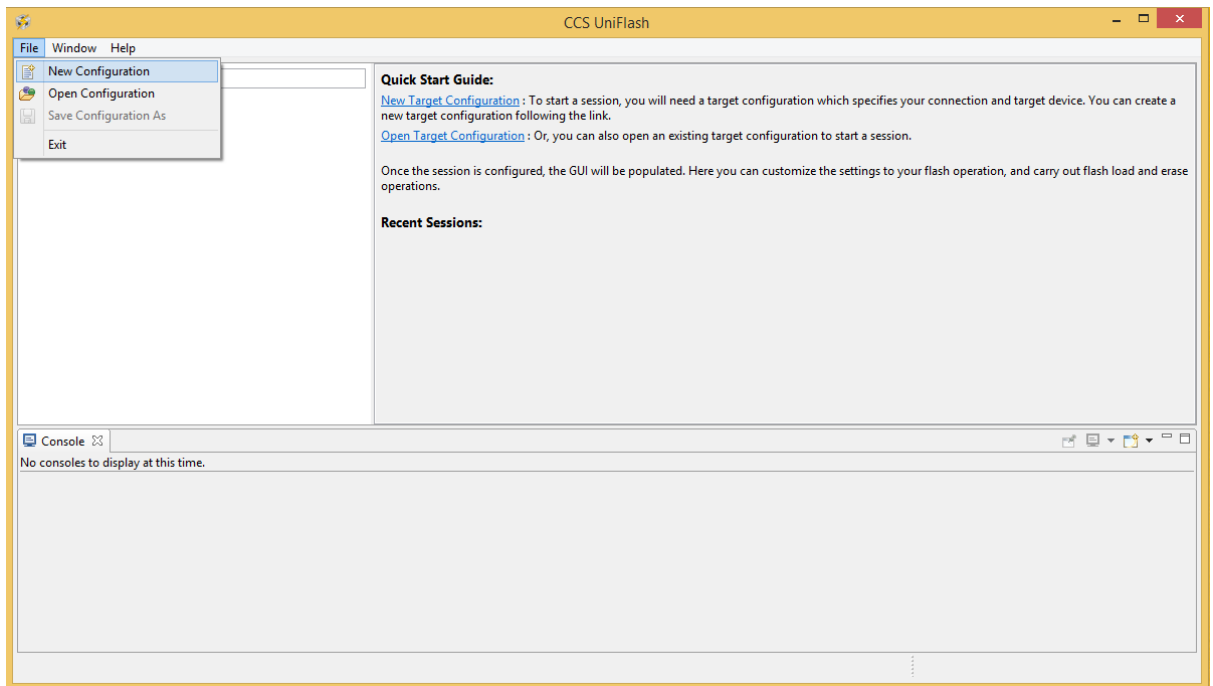


- app - sample application and related libraries.
  - app -> bin - Sample application binary
  - app -> src - Sample application and related libraries.
    - app -> src -> devkit\_app - Sample application source code.
    - app -> src -> ama - Arrayent multi-attribute helper module
    - app -> src ->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

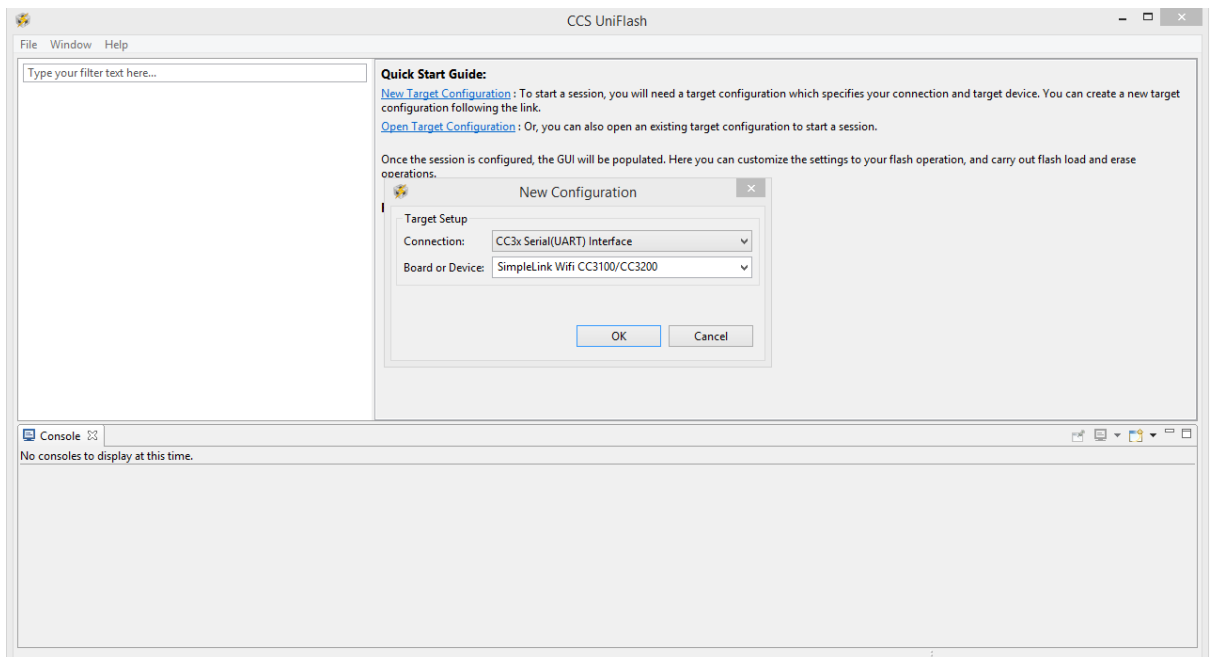
### 2. Steps to load the pre-built devkit application binary

TI CC3200 Launchpad:

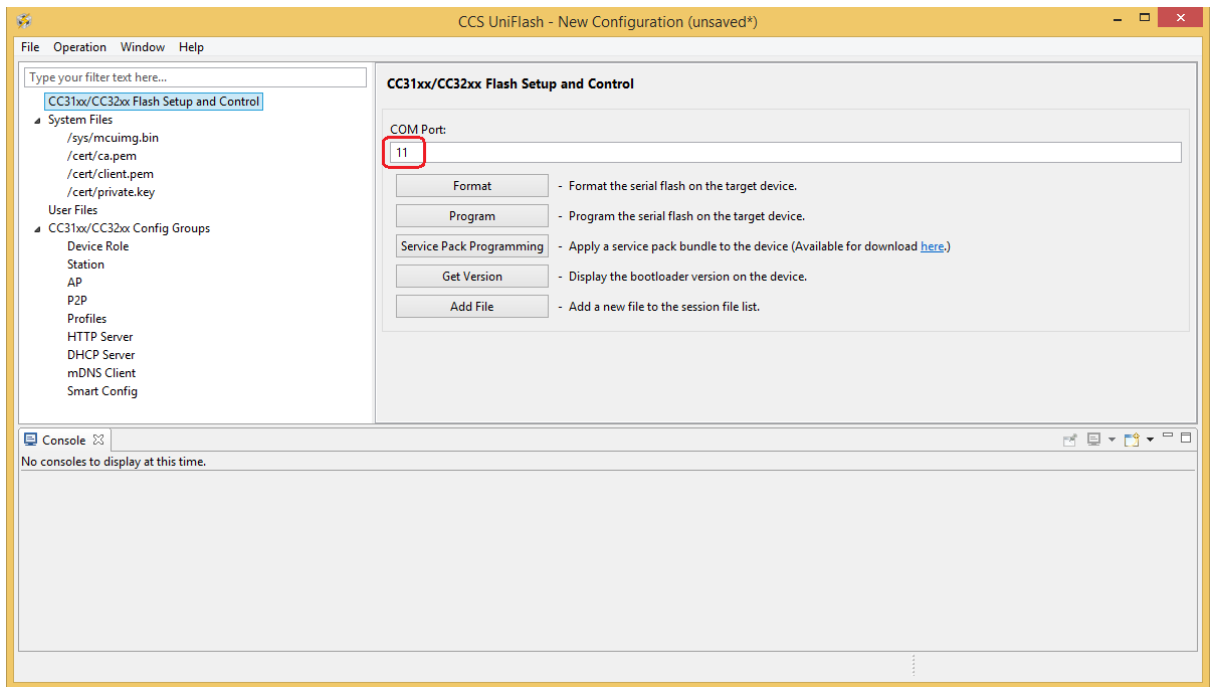
- a. Connect the board to the computer using mini USB cable. Wait a minute to let the device drivers install.
- b. If you click on the Installing device driver software popup, you can see the status of the device driver installation.
- c. Make sure SOP2 jumper is attached. Refer Figure 1 for SOP2 jumper location on TI CC3200 Launchpad.
- d. Follow the steps below to flash the application and store web pages into flash using uniflash:
  - i. Open Uniflash and select File->New Configuration



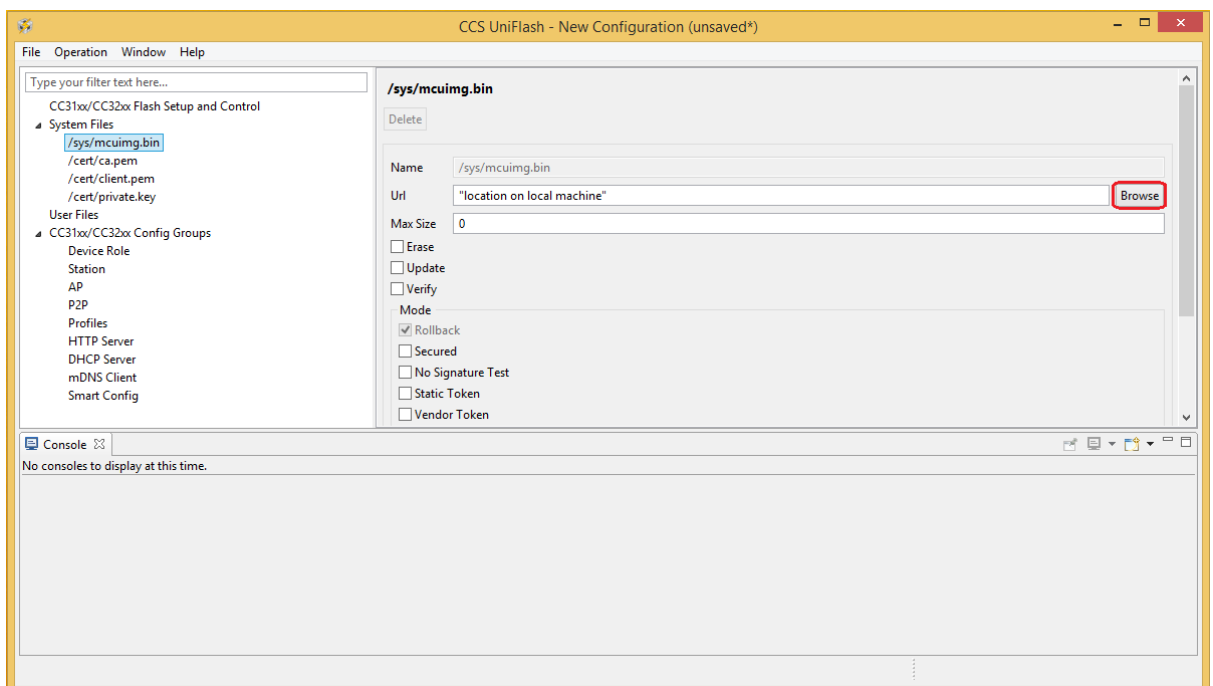
- ii. Select “CC3x Serial(UART) Interface” in connection and “SimpleLink Wifi CC3100/CC3200” in Board or Device



- iii. Enter COM port at which you board is connected(It is listed in Device Manager as “CC3200 Dual Port”)

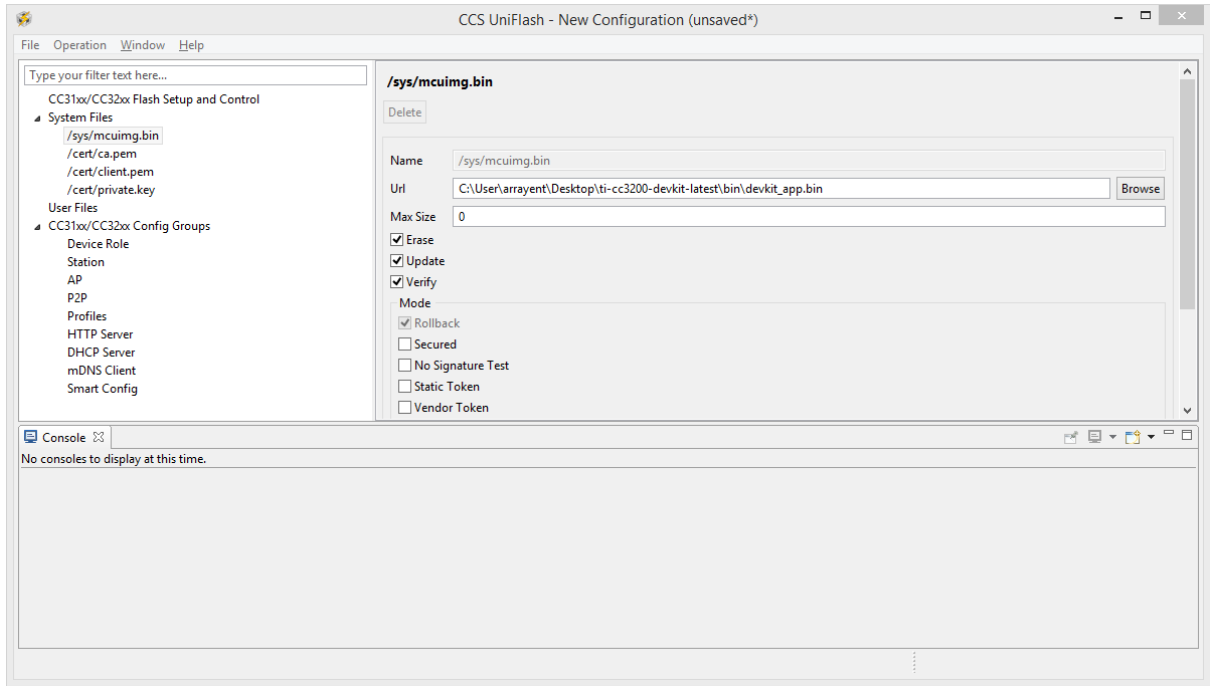


- iv. Select System Files->/sys/mcuimg.bin click on Browse Button highlighted in below screenshot

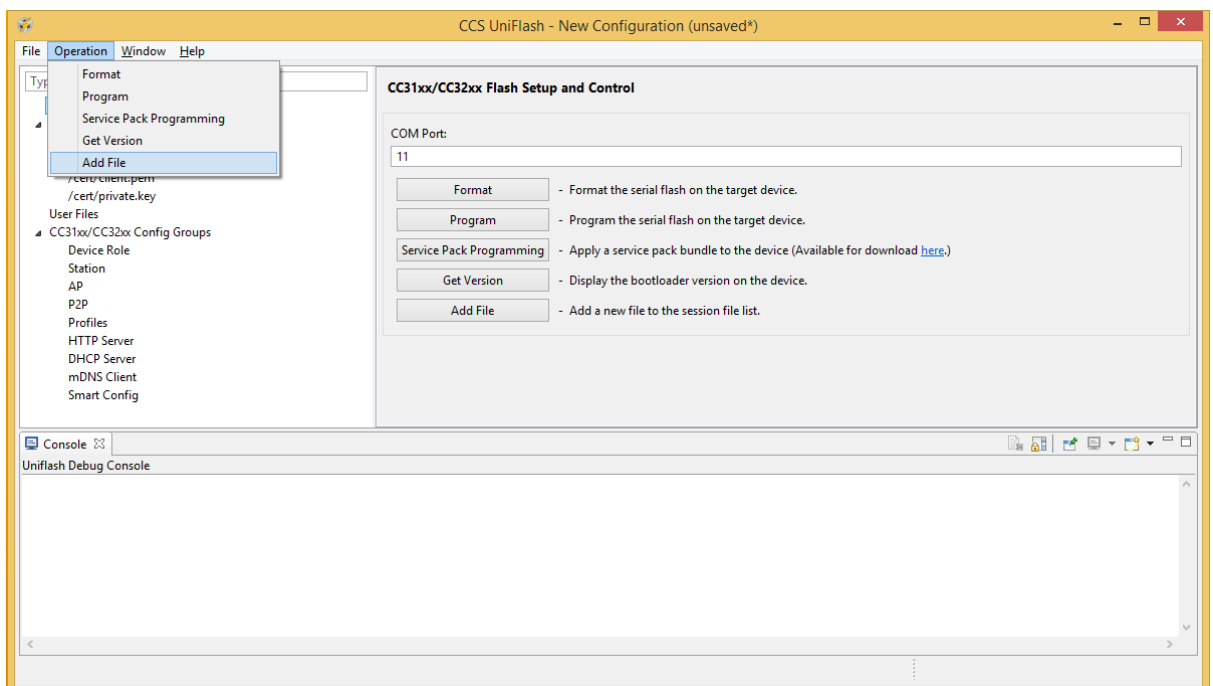


- v. Select Devkit application binary located in devkit\_app\_bundle under “bin” directory. Enable Erase, Update and Verify.

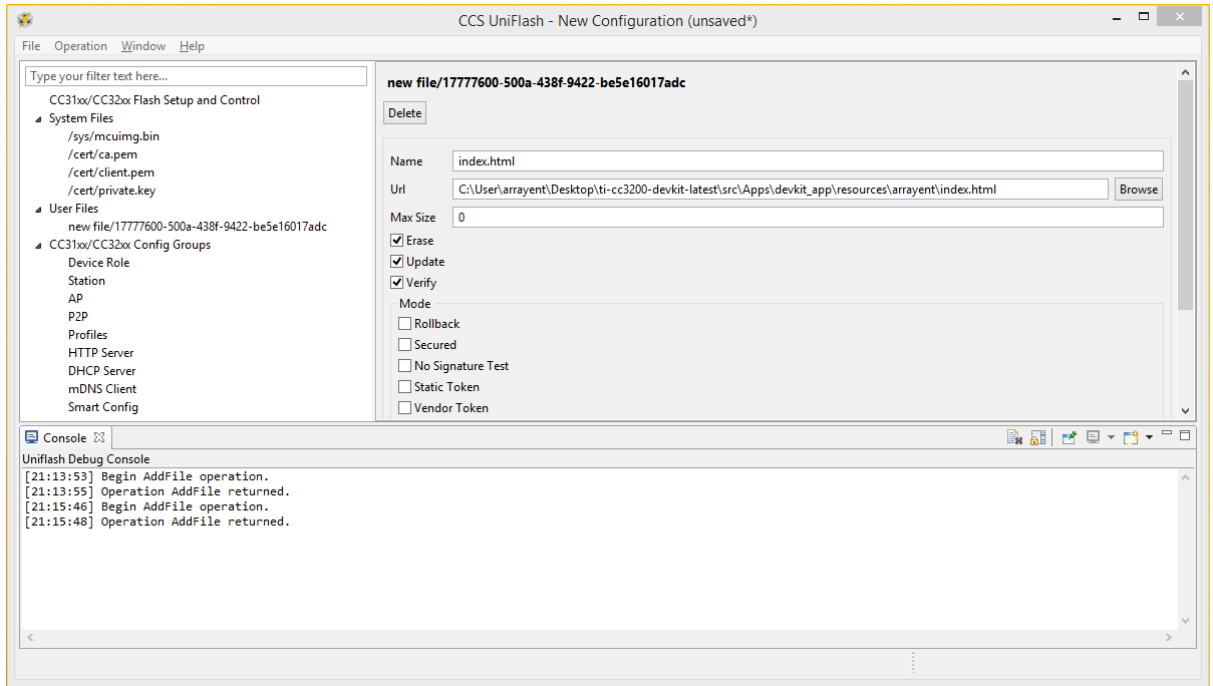




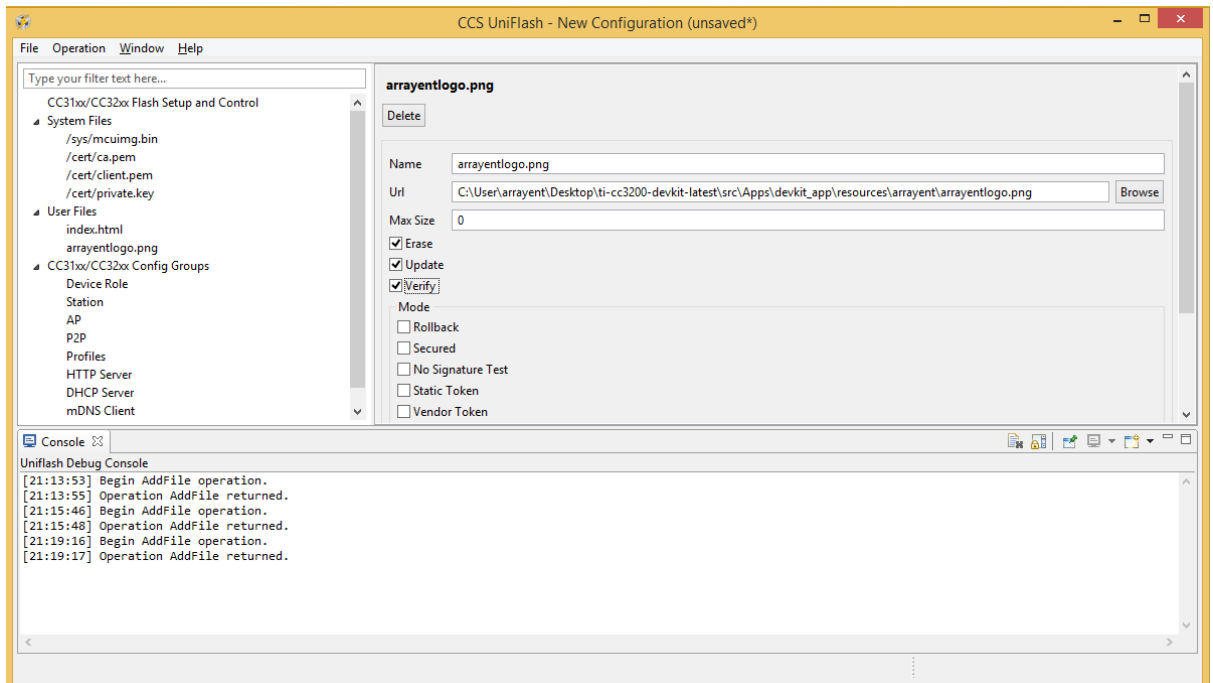
- vi. To store web pages into flash, Select Operation->Add File as shown in below screenshot:



- vii. Select the just added file under User Files, and change Name to "index.html" and Browse Url to "<PATH TO DEVKIT APP BUNDLE>\src\Apps\devkit\_app\resources\arrayent\index.html". Enable Erase, Update and Verify

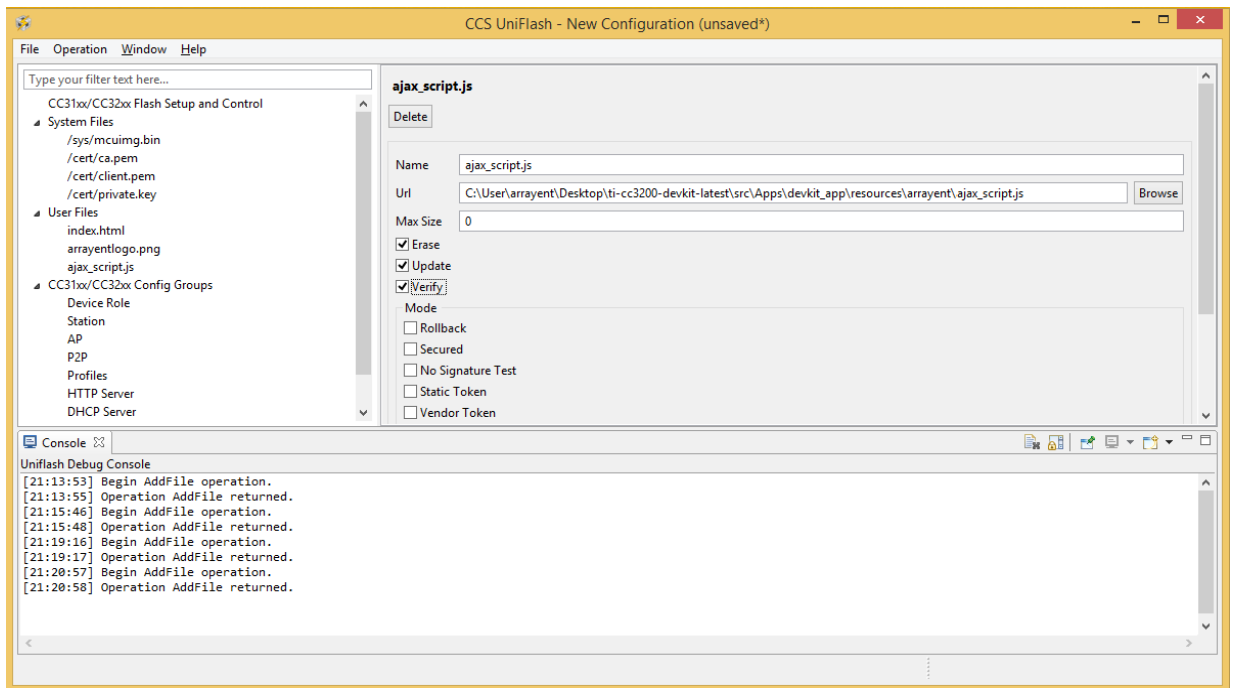


- viii. Follow step vi to add new file. Select the just added file under User Files and change Name to “arrayentlogo.png” and Browse Url to “<PATH TO DEVKIT APP BUNDLE>\app\src\devkit\_app\resources\arrayent\arrayentlogo.png”. Enable Erase, Update and Verify.

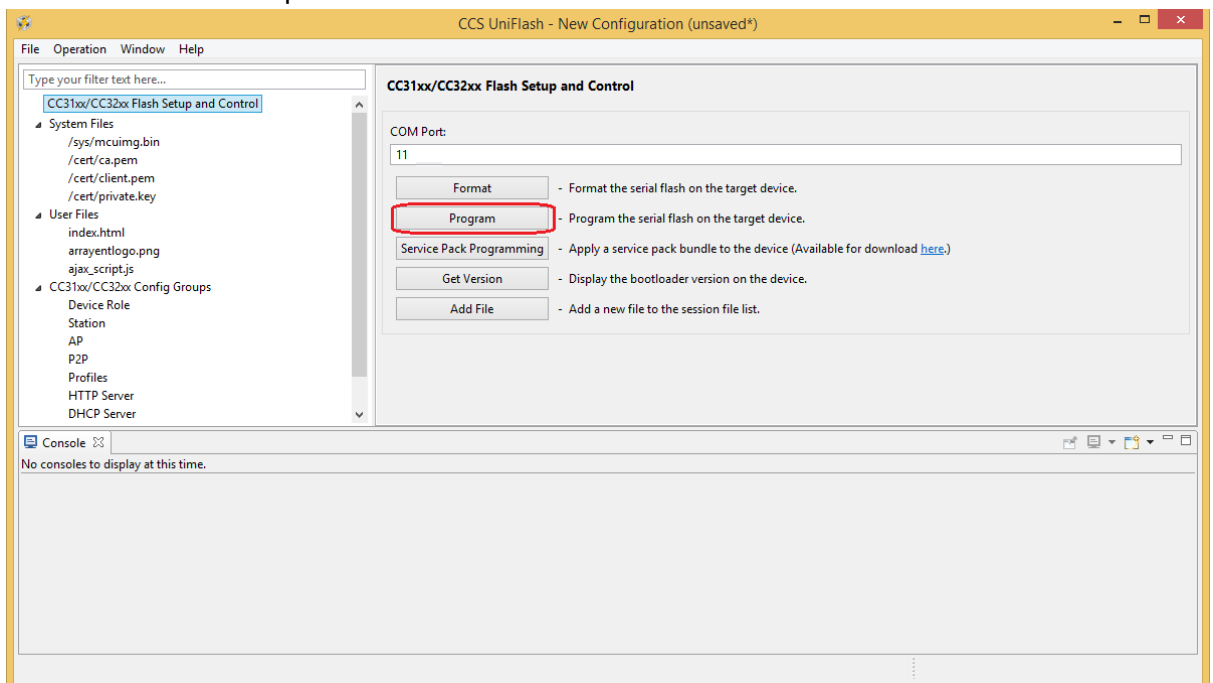


- ix. Follow step vi to add new file. Select the just added file under User Files and change Name to “ajax\_script.js” and Browse Url to “<PATH TO DEVKIT APP

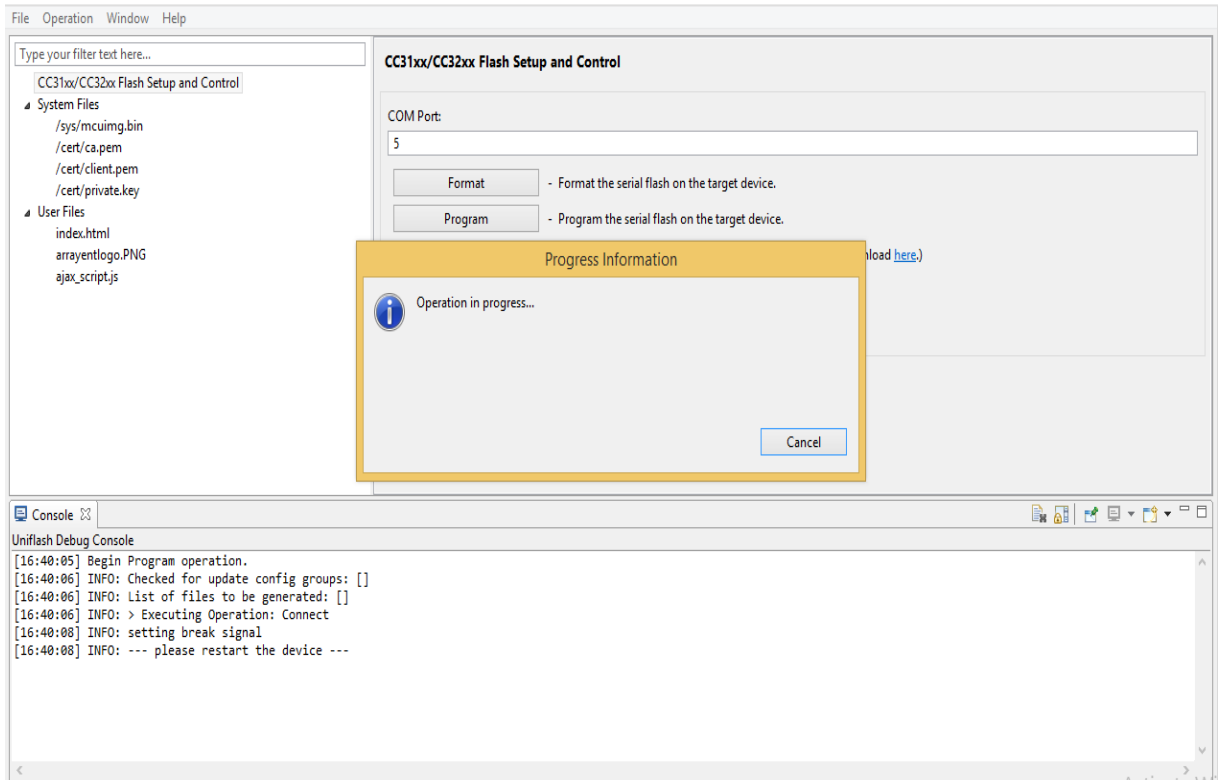
**BUNDLE>\app\src\devkit\_app\resources\arrayent\ajax\_script.js".**  
**Enable Erase, Update and Verify.**



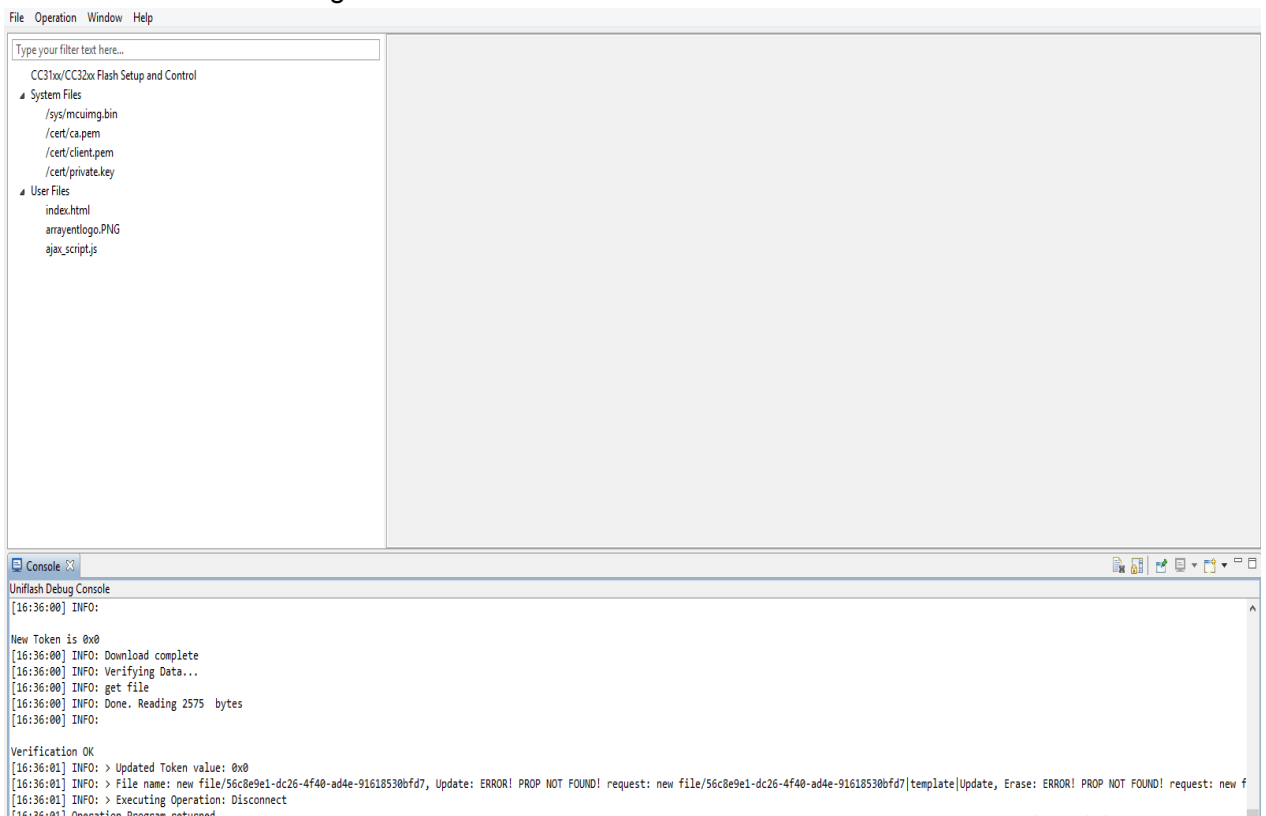
- x. Click on Program tab to download devkit application on TI CC3200 development board



- xi. If uniflash console displays to reset device as shown in following screenshot, Reset device using SW1 switch on TI development board.



- xii. On successful application download you will get output similar to the following screenshot:



- xiii. 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.
- xiv. Remove SOP2 jumper to boot devkit application.

### 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 CC3200 Launchpad:

- a. Follow steps a-b of the previous section to install the drivers for the TI CC3200 development board.
- b. Add Environment variable "TI\_CC3200SDK" with value *<PATH TO TI SDK>*. Default path for TI SDK is "C:\ti\CC3200SDK\_1.2.0\cc3200-sdk".
- c. Copy the "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.a to libaca\_ti.a.
- d. Open a command line interface and change directories to the larbuild directory.

```
>cd C:\Program Files (x86)\IAR Systems\Embedded Workbench 7.0\common\bin
```

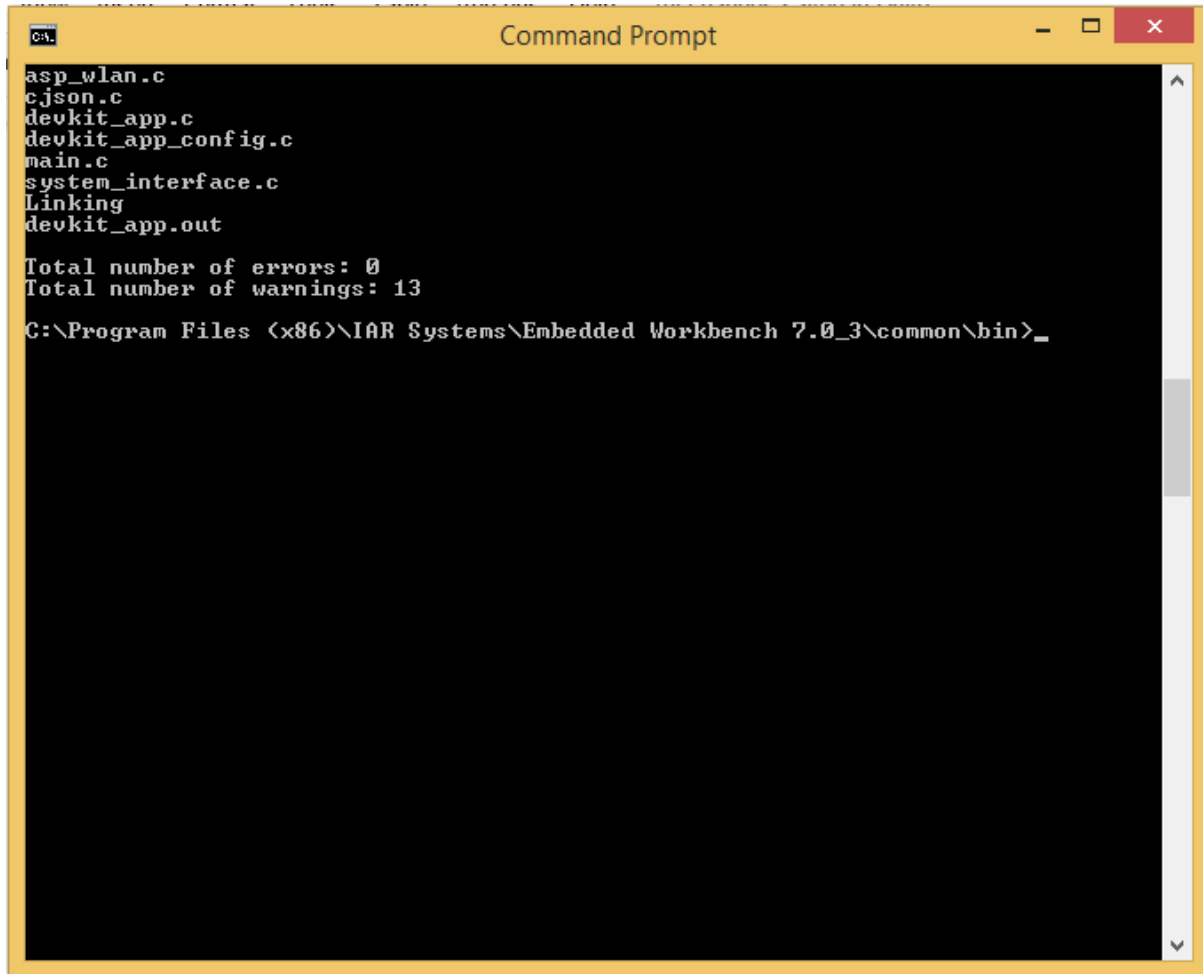
- e. Now execute the following command to build the sample application in Release mode (Without debug prints).

```
>IarBuild.exe <PATH TO DEVKIT APP BUNDLE>\app\devkit_app\ti\devkit_app.ewp -build Release
```

- f. Now execute the following command to build the sample application in Debug mode (With debug prints).

```
>IarBuild.exe <PATH TO DEVKIT APP BUNDLE>\app\devkit_app\ti\devkit_app.ewp -build Debug
```

- g. Upon a successful build and download you will see output similar to the following screenshot:



```
C:\Program Files (x86)\IAR Systems\Embedded Workbench 7.0_3\common\bin>
asp_wlan.c
cjson.c
devkit_app.c
devkit_app_config.c
main.c
system_interface.c
Linking
devkit_app.out

Total number of errors: 0
Total number of warnings: 13

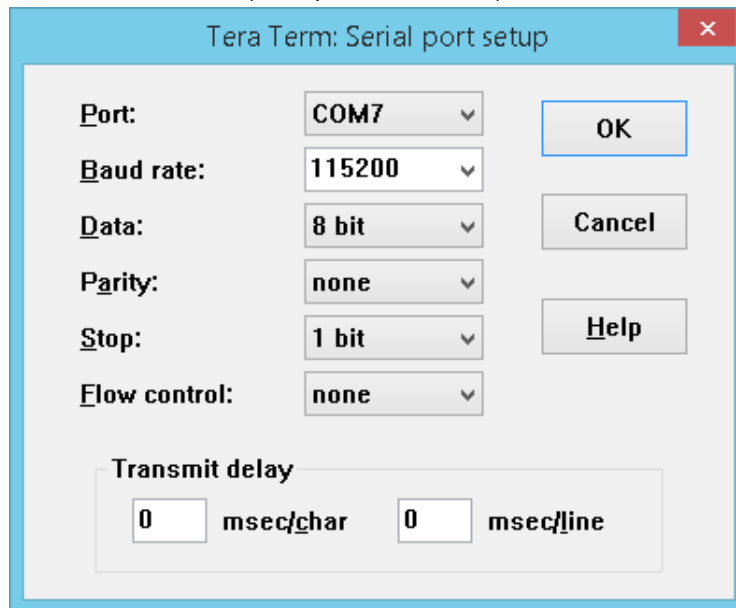
C:\Program Files (x86)\IAR Systems\Embedded Workbench 7.0_3\common\bin>
```

- h. devkit\_app.bin file will be generated under Release directory.
- i. Follow the steps of “Steps to load the pre-built sample application binary for TI CC3200 Launchpad” to download sample application binary on development board

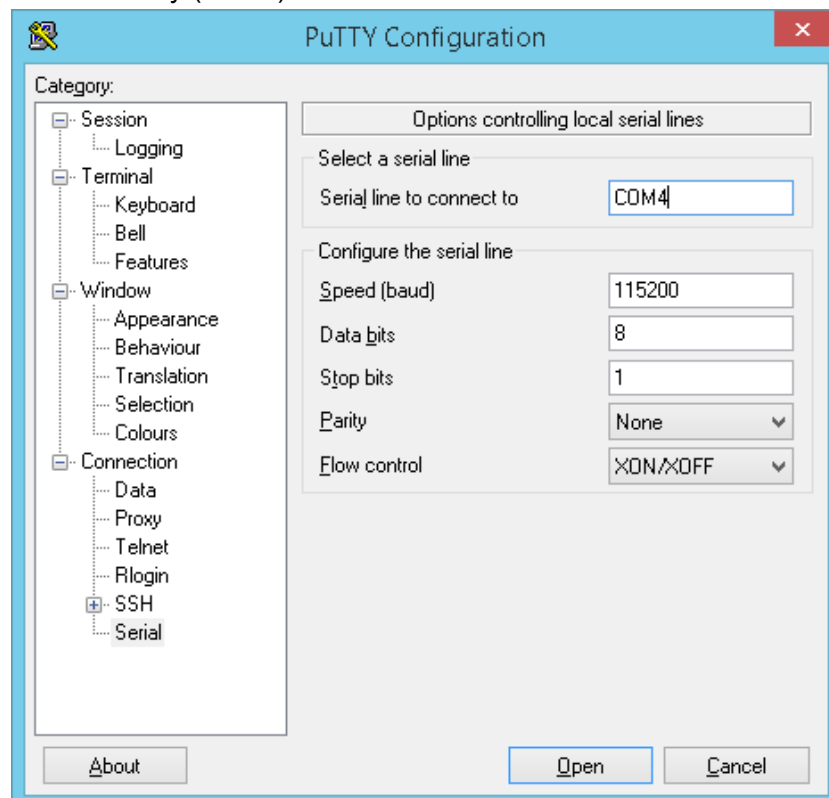
#### 4. Setup terminal program

We have taken TI CC3200 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):



- ii. For Putty (Serial):

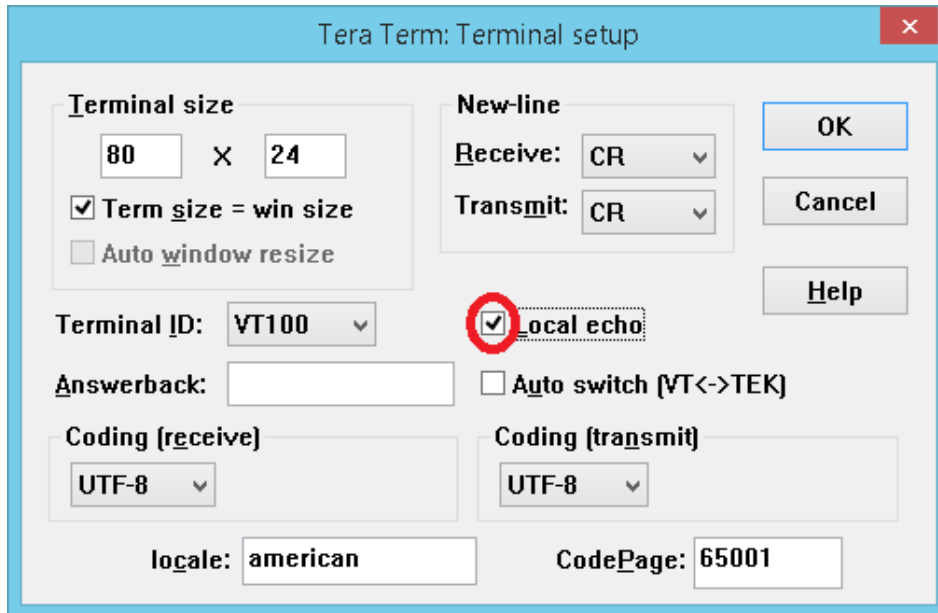


- iii. Configure and open minicom using following command:

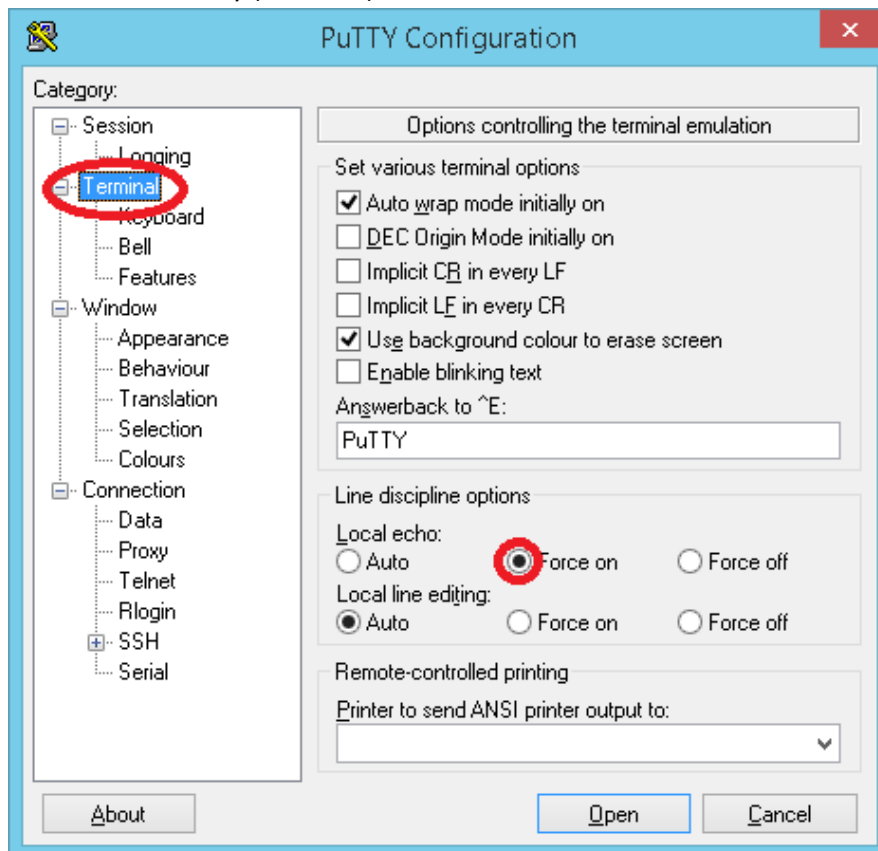
```
$ sudo minicom -D /dev/ttyUSB1
```

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

- i. For TeraTerm (Setup -> Terminal):



- ii. For Putty (Terminal):



**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
- Connect the board to the computer using mini USB cable.
  - Open the terminal and Press the RESET button.
  - If the Serial Port settings are correct you will see the output similar to the below screenshot:

```
COM11 - PuTTY

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

## 5. 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.

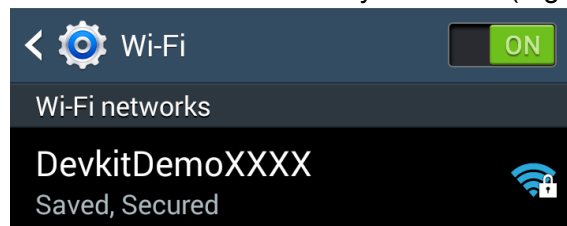
## 6. Configure WiFi router credentials

**Note:** Devkit application on Atmel Development Board does not support WiFi router credentials configuration through Soft AP and WPS. Please use console commands to configure WiFi router credentials on Atmel Development Board.

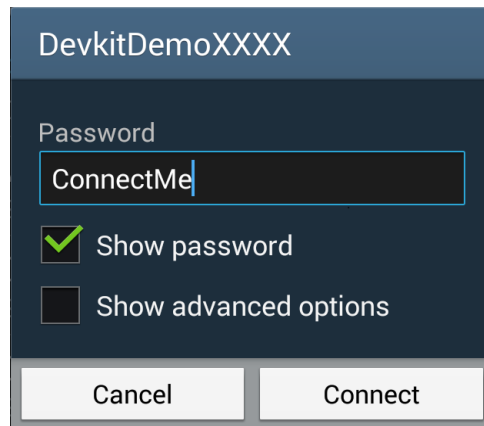
### a. Using Web Browser

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

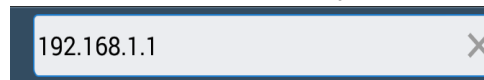
- iv. If application is not in "Soft AP" state press SW2 to go to "Soft AP" state.
- v. Check that the application is in "Soft AP" state from the LED state.
- vi. Scan for Wi-Fi devices on your device(e.g. Mobile, laptop).



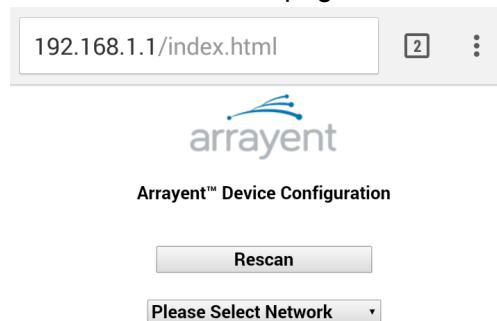
- vii. Connect to "DevkitDemoXXXX" Wi-Fi AP using "ConnectMe" as a passphrase.(If device is already configured with device code then last 4 character will be replace by last 4 digit of device code).



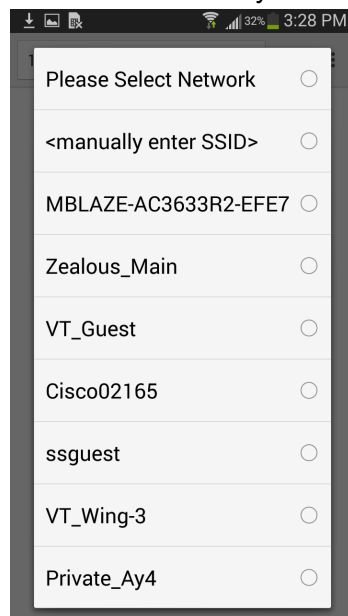
- viii. Open web browser and type “192.168.1.1” in URL bar.




- ix. You will see the webpage similar to following screenshot:



- x. 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.



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



192.168.1.1/index.html

arrayent

**Arrayent™ Device Configuration**

Rescan

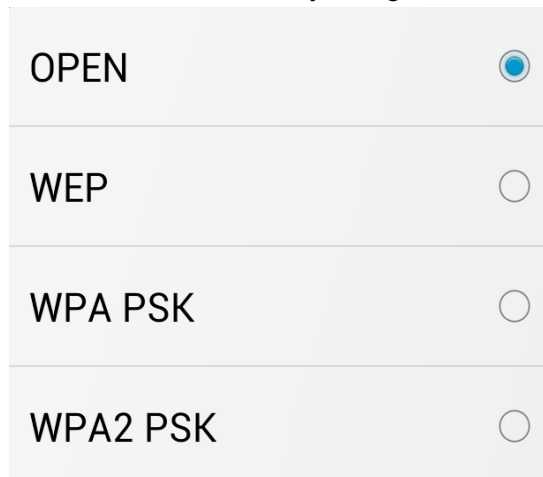
SSID Belkin\_Ay

Select Security OPEN

Enter a Password

Connect

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



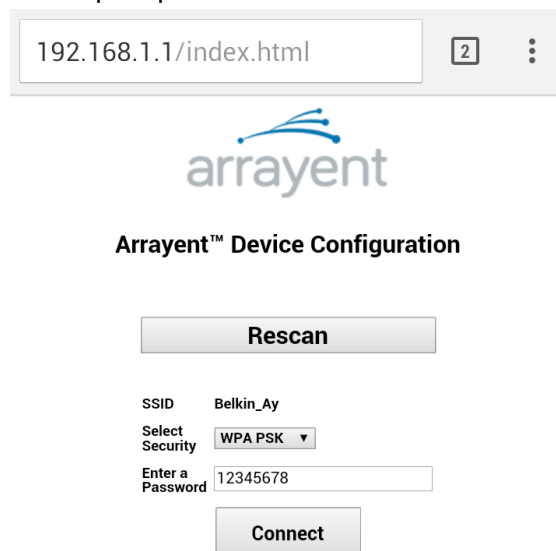
OPEN

WEP

WPA PSK

WPA2 PSK

- xiii. Enter passphrase in “Enter a Password” Textbox.



192.168.1.1/index.html

arrayent

**Arrayent™ Device Configuration**

Rescan

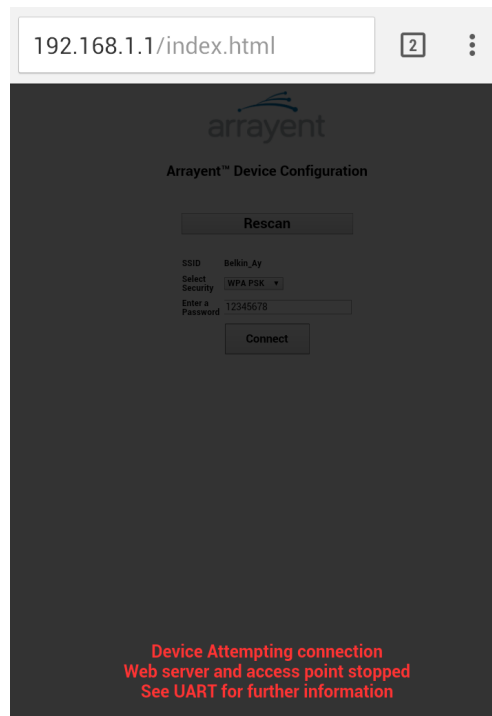
SSID Belkin\_Ay

Select Security WPA PSK

Enter a Password 12345678

Connect

- xiv. Click on Connect button.



- xv. 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. Follow steps i - v from section a. Using DevKit Smartphone app.
- ii. Push the SW1 push button.
- iii. Check that the application is in "WPS" state from the LED state.
- iv. Press WPS button on your WiFi router.
- v. 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**

- 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:wpassphrase,<your_passphrase>#
demo&cset@3:wsecurity,<security_level>#
```

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

## 7. 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.

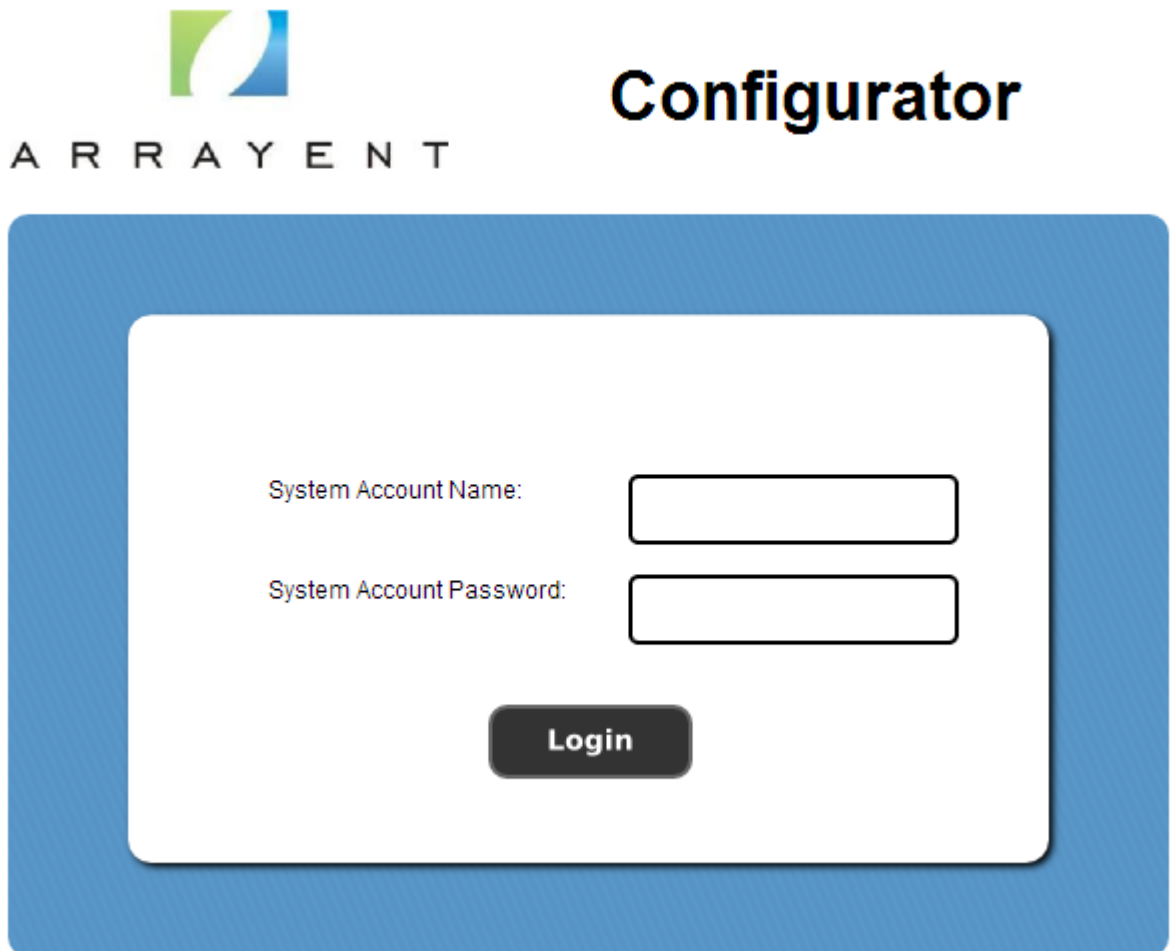
## Chapter 4: Configuring the Arrayent Cloud

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.

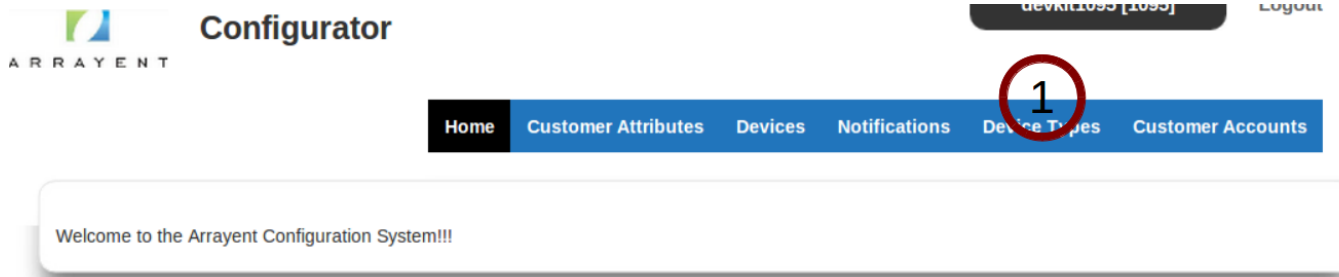
## 2. Creating a Device Data Model

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

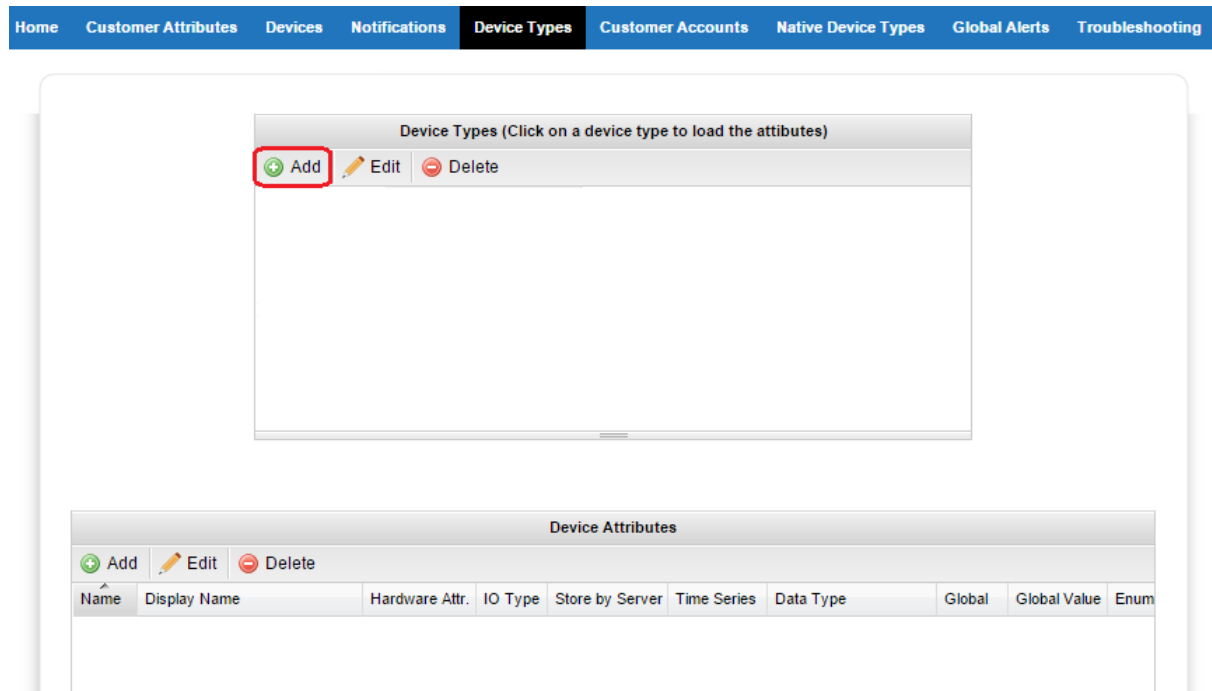
The name of the Device Type we are creating will be called “Devkit\_app”.

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



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

b. Click **Add**.



c. For Device Type Name enter Devkit\_app.

d. For Device Type Display Name enter 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 attributes)              |            |                   |
|---|------------|-------------------|
| <div> <span>+ Add</span> <span>✎ Edit</span> <span>✖ Delete</span> </div> |            |                   |
| 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 attributes)              |            |                   |
|---|------------|-------------------|
| <div> <span>+ Add</span> <span>✎ Edit</span> <span>✖ Delete</span> </div> |            |                   |
| Type Id   | Type Name  | Type Display Name |
| 387   | Devkit_app | Devkit_app        |

| Device Attributes   |              |                |         |                 |             |           |        |              |      |
|---|--------------|----------------|---------|-----------------|-------------|-----------|--------|--------------|------|
| <div> <span>+ Add</span> <span>✎ Edit</span> <span>✖ Delete</span> </div> |              |                |         |                 |             |           |        |              |      |
| 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.

The 'Device Attribute' dialog box is shown with the following fields and values:

| Field                  | Value                               |
|------------------------|-------------------------------------|
| Name:                  | SW1_count                           |
| Display Name           | SW1_count                           |
| Time Series            | <input type="checkbox"/>            |
| Hardware Attribute     | <input checked="" type="checkbox"/> |
| Hardware IO Type       | From device                         |
| Stored by Server       | <input type="checkbox"/>            |
| Attribute Data Type    | BEIntegerSigned32Br                 |
| Global                 | <input type="checkbox"/>            |
| Global Attribute Value |                                     |
| Enumerated Alias       | 1                                   |

Buttons: Save, Cancel

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

The 'Device Attribute' dialog box is shown with the following fields and values:

| Field                  | Value                               |
|------------------------|-------------------------------------|
| Name:                  | Led1_cmd                            |
| Display Name           | Led1_cmd                            |
| Time Series            | <input type="checkbox"/>            |
| Hardware Attribute     | <input checked="" type="checkbox"/> |
| Hardware IO Type       | To device                           |
| Stored by Server       | <input type="checkbox"/>            |
| Attribute Data Type    | Boolean                             |
| Global                 | <input type="checkbox"/>            |
| Global Attribute Value |                                     |
| Enumerated Alias       | 2                                   |

Buttons: Save, Cancel

|                        |                                     |
|------------------------|-------------------------------------|
| Name:                  | Led2_cmd                            |
| Display Name           | Led2_cmd                            |
| Time Series            | <input type="checkbox"/>            |
| Hardware Attribute     | <input checked="" type="checkbox"/> |
| Hardware IO Type       | To device                           |
| Stored by Server       | <input type="checkbox"/>            |
| Attribute Data Type    | Boolean                             |
| Global                 | <input type="checkbox"/>            |
| Global Attribute Value |                                     |
| Enumerated Alias       | 4                                   |

Save Cancel

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

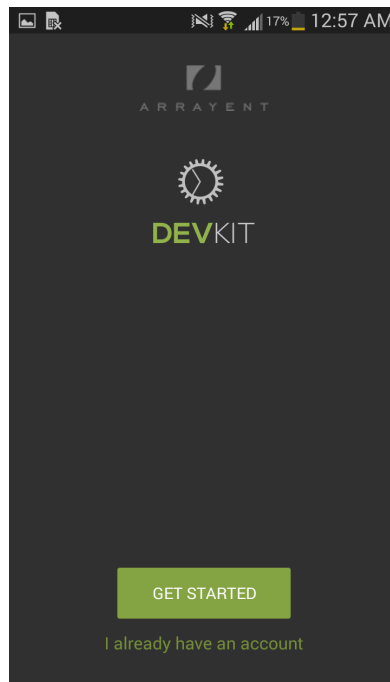
Follow the instructions in the next section to learn how to create a new Customer Account in the Arrayent Cloud. Later on you will log in to the Arrayent Cloud as this Customer Account and then monitor and control your evaluation board.

#### a. Using Configurator:

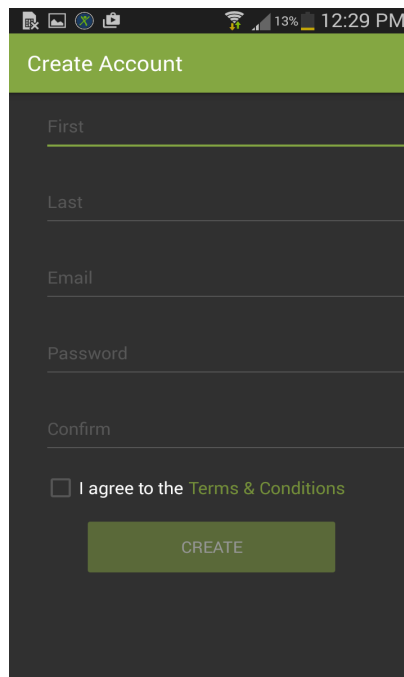
- i. Enter the username of the new account in the text field next to **Customer Account Name**.
- ii. Select the checkbox. This will enable you to enter text in the textbox next to **Customer Account Password**.
- iii. Enter the password of the new account.
- iv. Leave **Master Account Name** empty.
- 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.

Enter Activation Code

CONTINUE

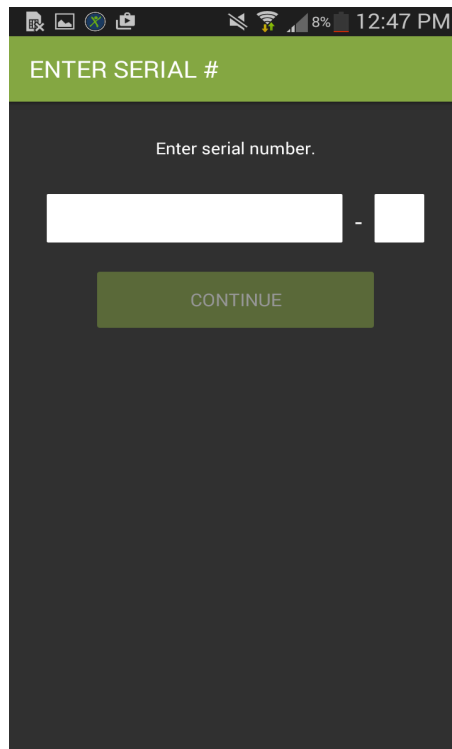
- iv. Enter verification code send to your email ID entered during create account details. Click on **CONTINUE** button. On successful account creation you will see the output similar to the below screenshot:



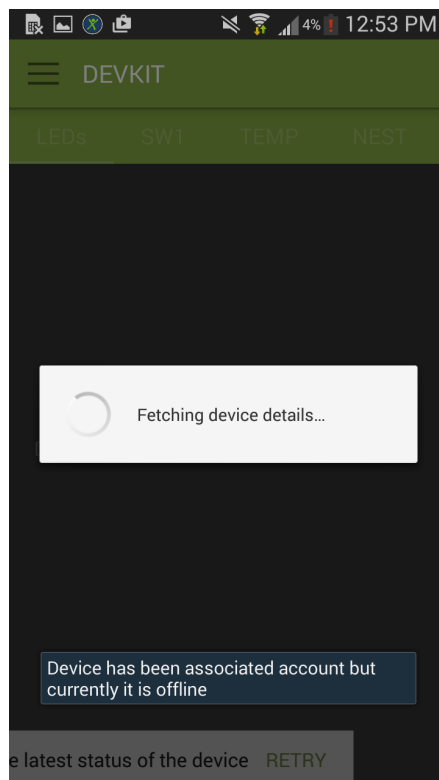
## Account Verified



LOGIN



- 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:





## Configurator

[Home](#)
[Customer Attributes](#)
[Devices](#)
[Notifications](#)
[Device Types](#)
[Customer Accounts](#)

Retrieve Customer Account:  Retrieve

**Customer account attributes**

Customer Account Name 1  
 Customer Account Password 3 2 Save New Password  
 Master Account Name  
 Your email Address 4  
 Phone Number  
 Street Address

5
New
Save
Delete

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.

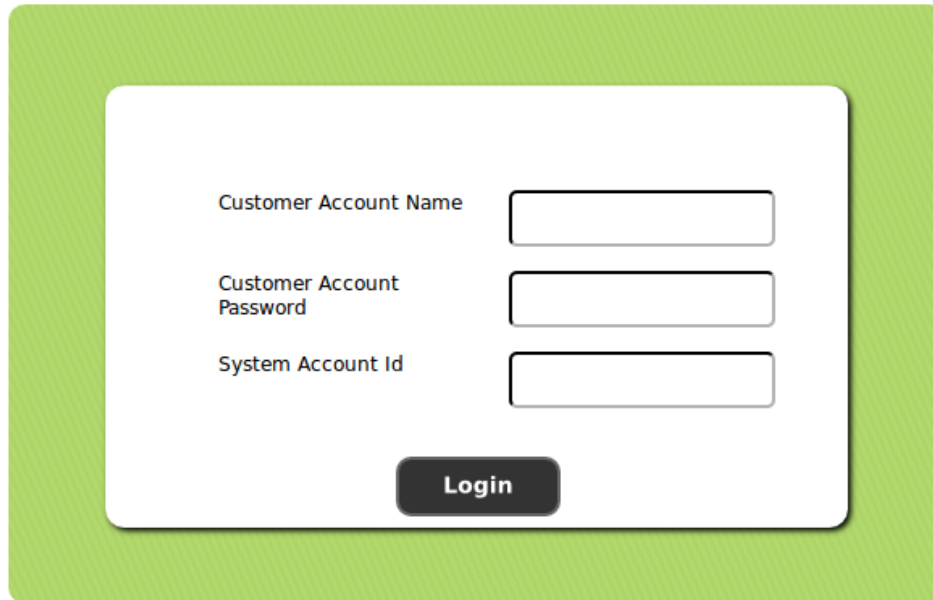
#### a. Logging In to the Utility Application

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.




# Utility Application



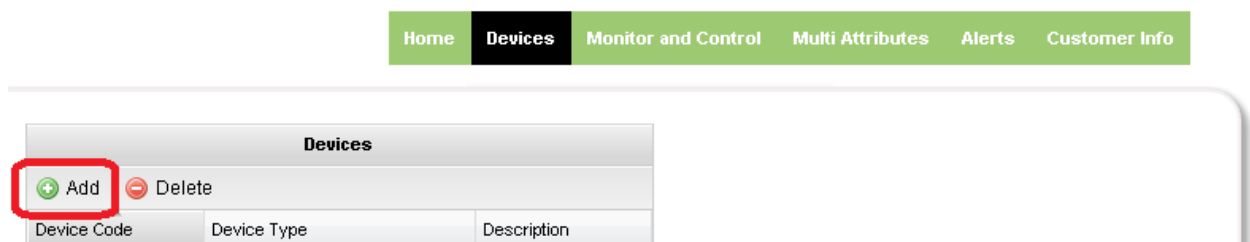
## b. Adding the device to your account

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.



- ii. Click **Add**.



- 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.

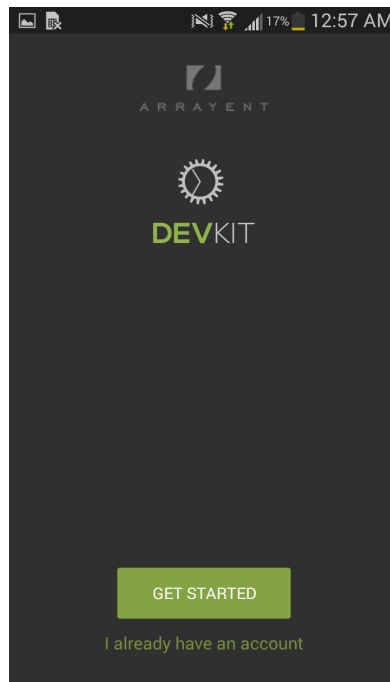
## Using Devkit Android Application

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

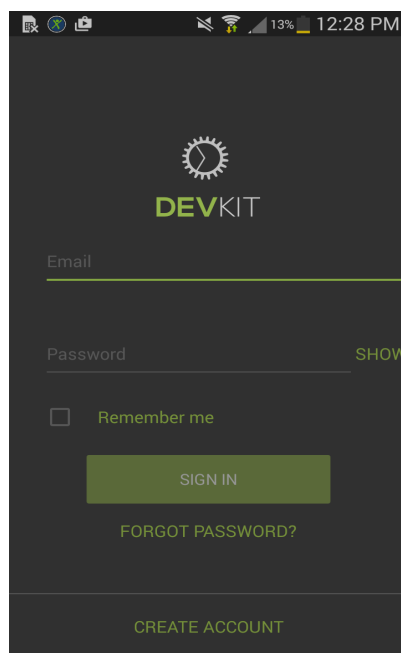
#### a. Logging In to the Devkit Android Application

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.



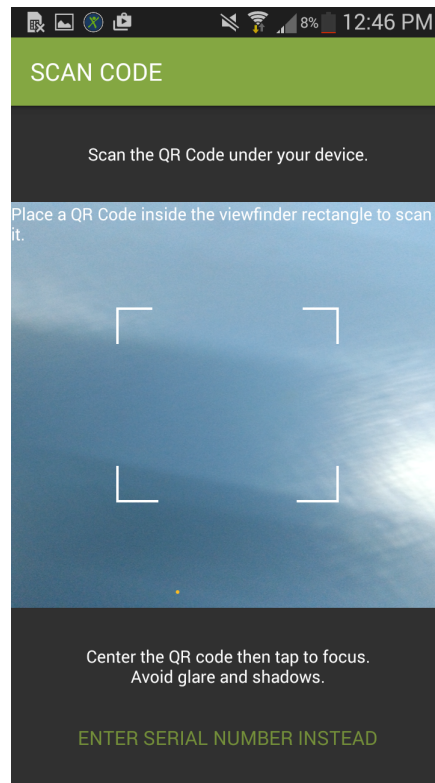
- ii. Log in with the Customer Account credentials that you created earlier with the Devkit android application. Click on **SIGN IN** button.



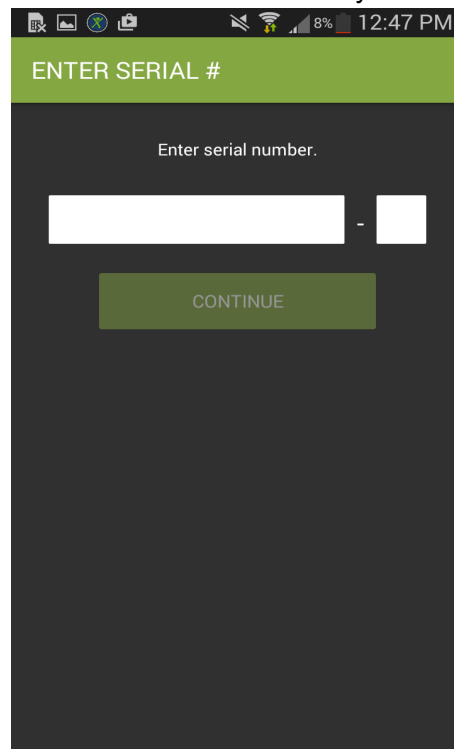
**b. Adding the device to your account**

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.

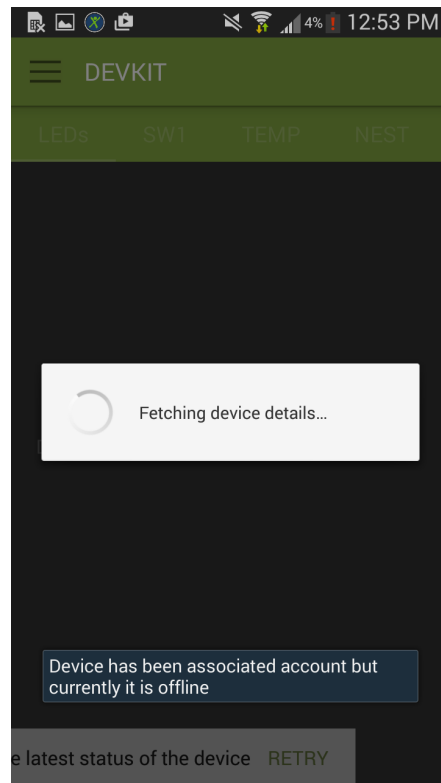
- i. If you don't have QR code then click on **ENTER SERIAL NUMBER INSTEAD**.



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



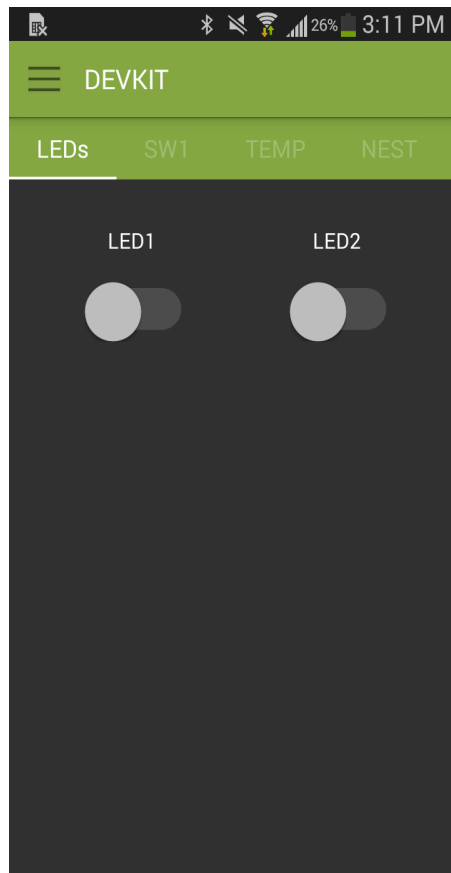
- iii. On successful Association of device with account you will see the output similar to the below screenshot:



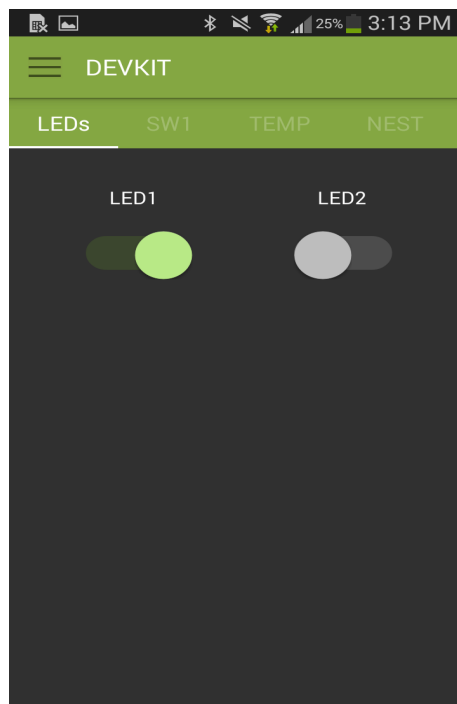
The device is now claimed by this Customer Account. No other Customer Account can access this device.

**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:

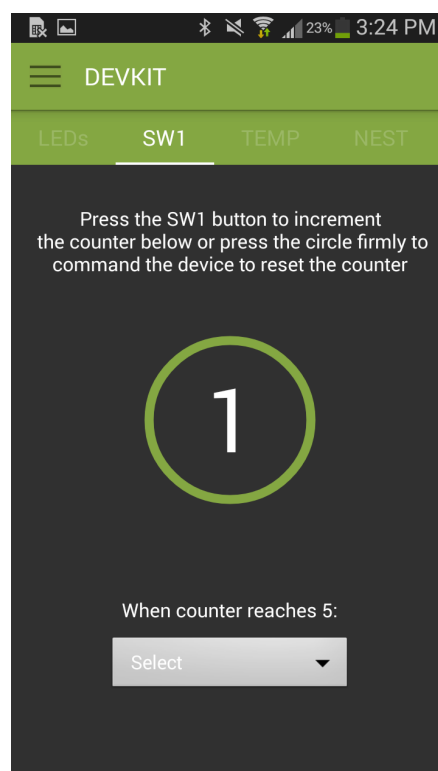


- v. Click the LED1 switch again.
- vi. Look at your board again. The LED should be off now.
- vii. The Led1\_State attribute slider bar in the utility app will show the current state of the LED1.

**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.

**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<br>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<br>FALSE=Single<br>TRUE=Multi | From Cloud |            |
| MultiAttribute_state      | Reports if the application is using single or multi attribute to receive properties from the cloud   | Boolean<br>FALSE=Single<br>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<br>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<br>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   |  |