Embedded System Design Techniques™

Connecting Edge Devices to the IoT using Amazon FreeRTOS

Class 3: Setting up and Configuring Amazon FreeRTOS

March 21st, 2018

Jacob Beningo

Presented by:

CEC CONTINUING EDUCATION CENTER

Digi-Key ELECTRONICS

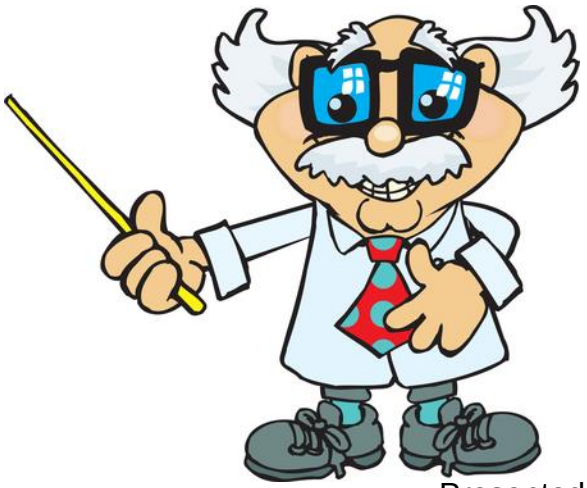
Course Overview

Topics:

- Introduction to Amazon FreeRTOS
- Amazon Web Services Fundamentals
- **Setting up and Configuring Amazon FreeRTOS**
- Amazon FreeRTOS Behind the Scenes
- Creating your own a:FreeRTOS Application

Session Overview

- Setup Procedure
- Toolchain Setup
- Configuring for AWS
- Running a:FreeRTOS



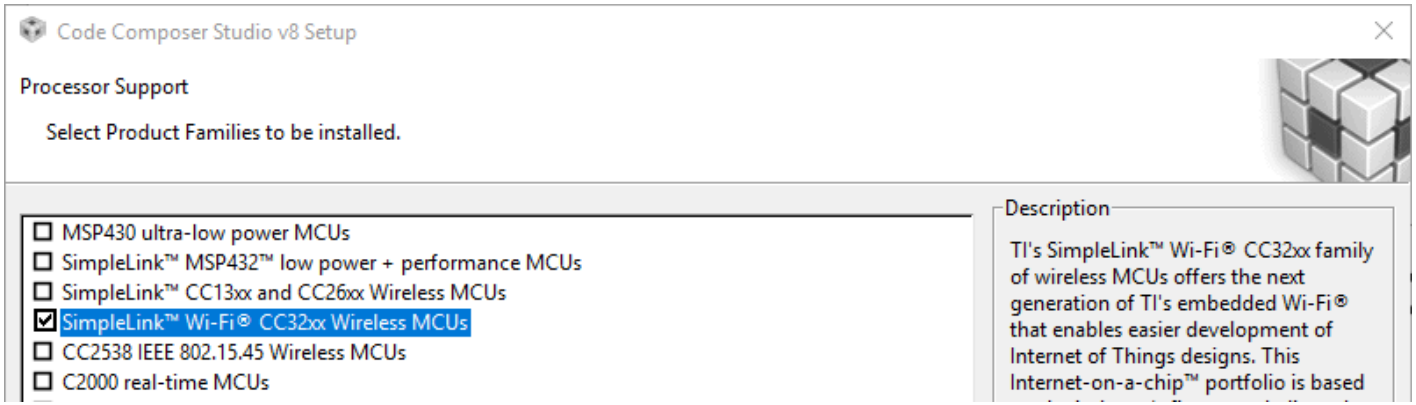
Presented by:

a:FreeRTOS Setup Procedure

1) Install Code Composer
2) Download a:FreeRTOS
   - Configured for the CC3220SF-LAUNCHXL
3) Import a:FreeRTOS
4) Configure for AWS
5) Run the example

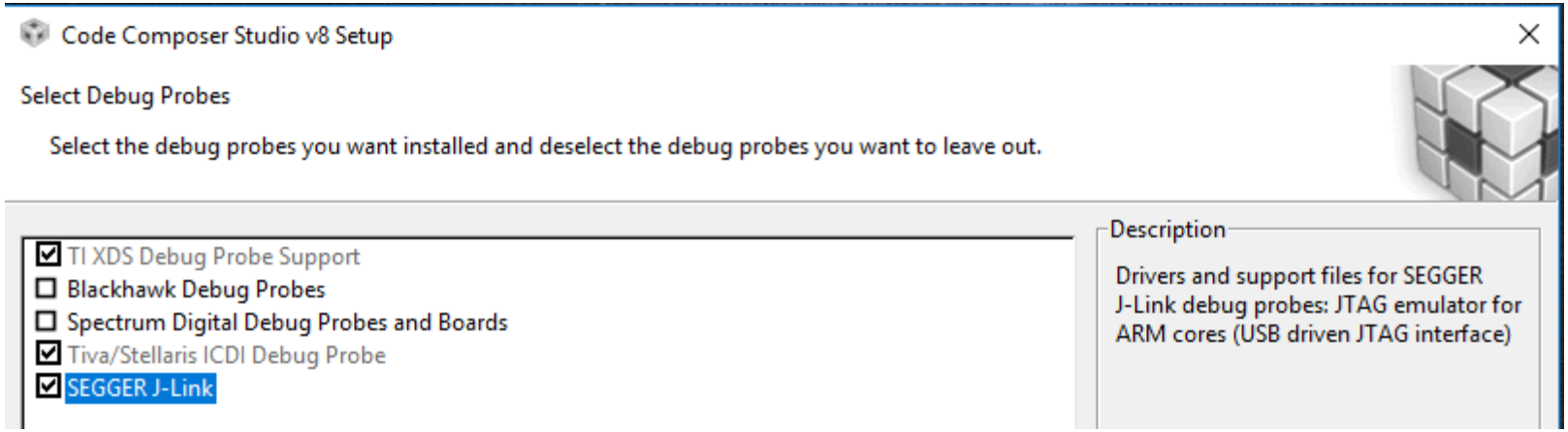
Step 1 - Code Composer Setup

1) Download the latest Code Composer
   - http://processors.wiki.ti.com/index.php/Download_CCS
   - Use the offline installer
2) Select the SimpleLink Wi-Fi CC32xx

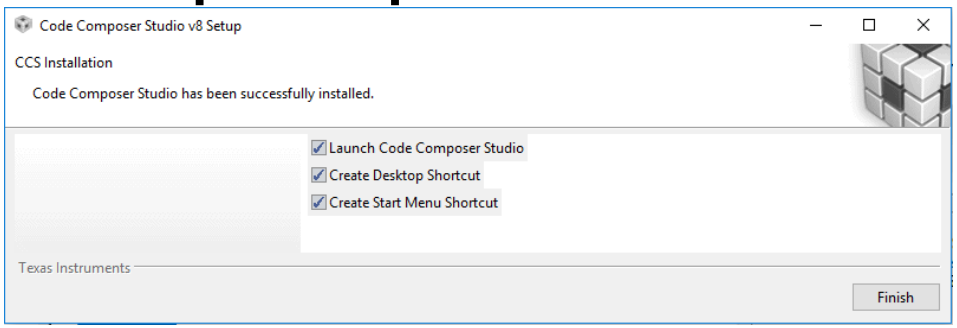


Step 1 - Code Composer Setup

3) Include the J-Link (If you have one)

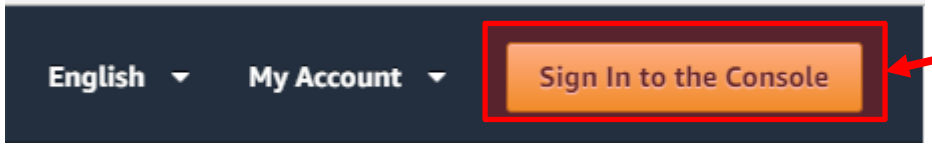


4) Follow the prompts until CC is installed.

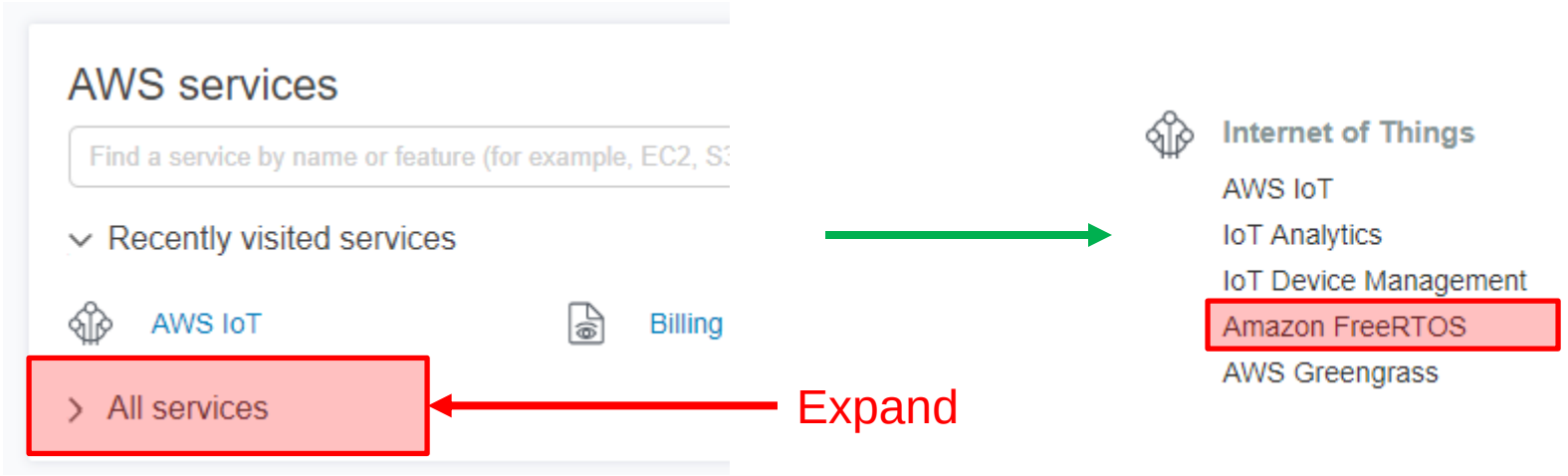


Step 2 – a:FreeRTOS Download

https://aws.amazon.com

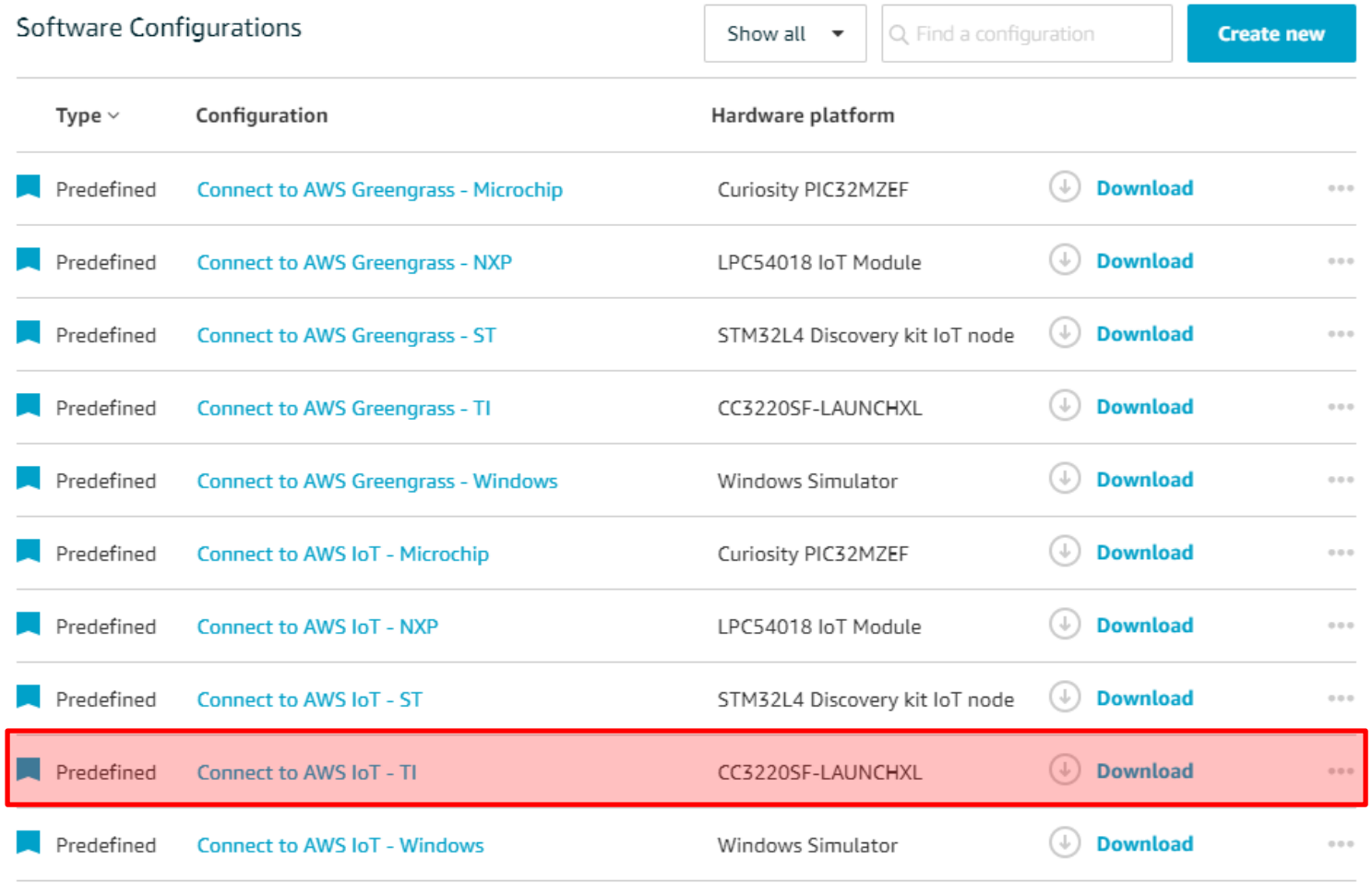


Sign-in / Create Account



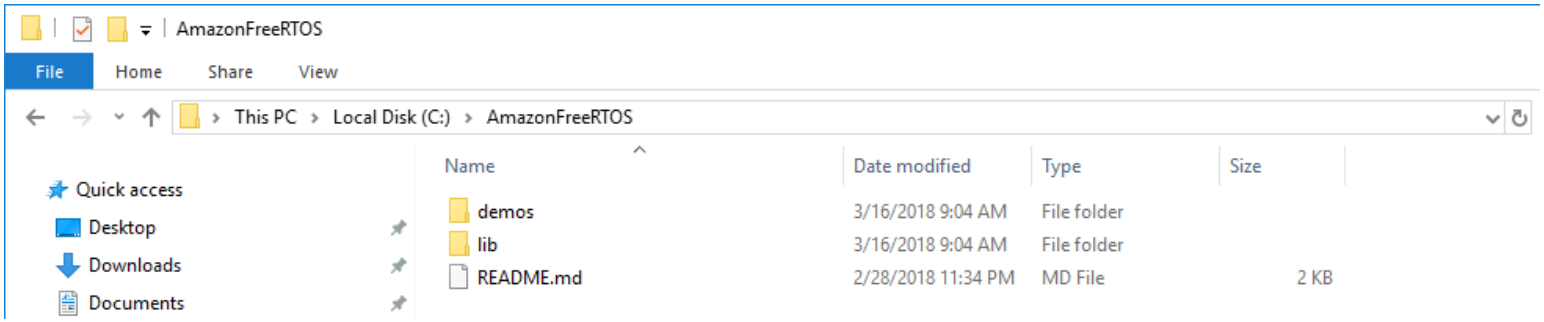
Expand

Step 2 – a:FreeRTOS Download



Step 2 – a:FreeRTOS Download

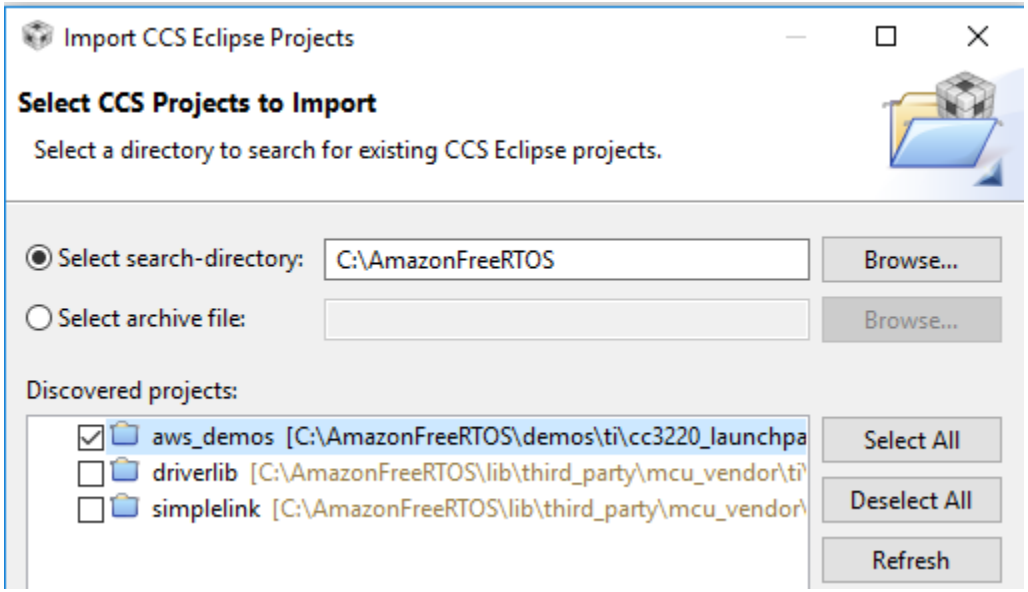
Unzip the sample project and place it in your root C:\



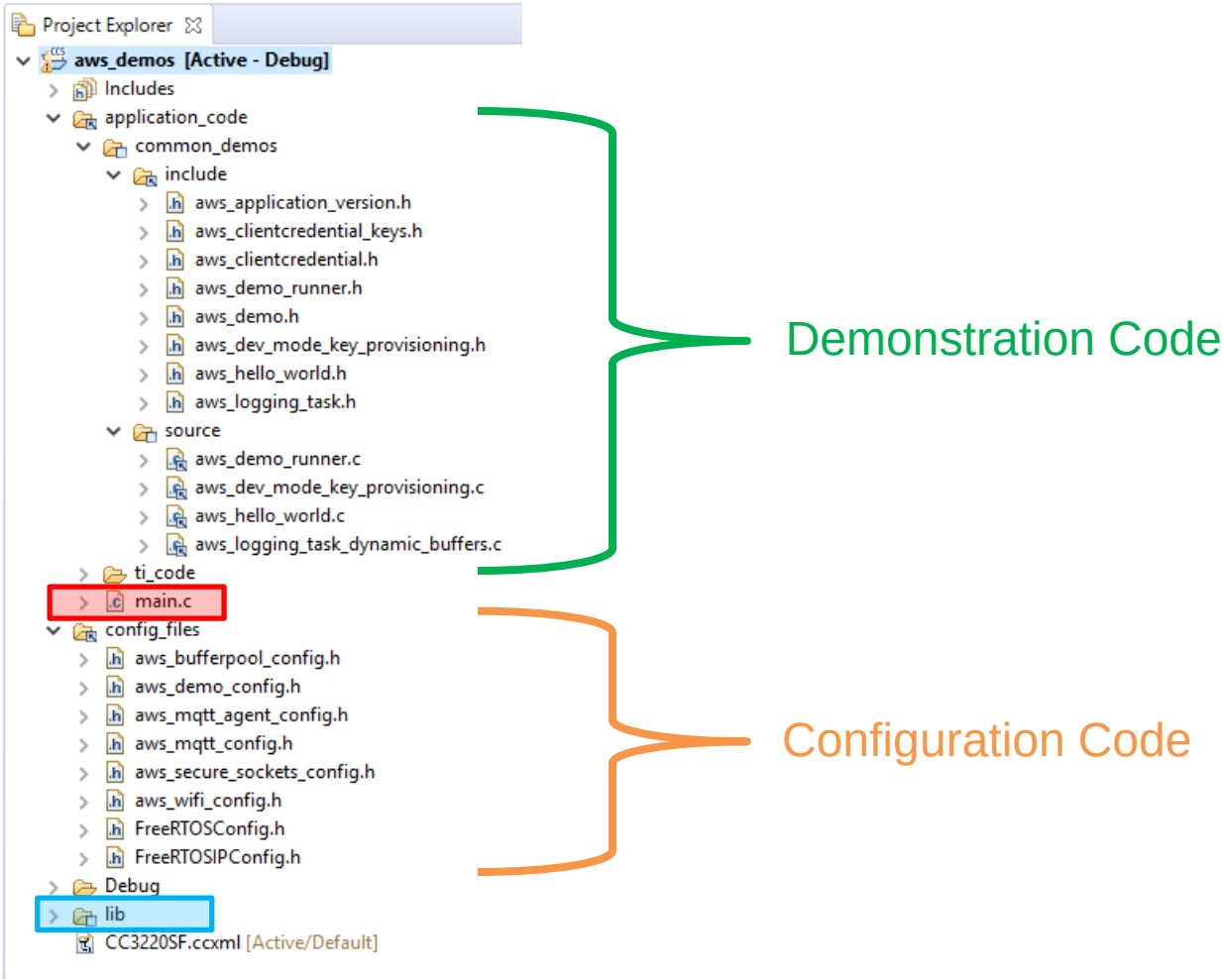
Note: Windows is limited to 260 characters for a path and the longest path in a:FreeRTOS is 122 characters!

Step 3 – Import a:FreeRTOS

1) Open Code Composer
2) Project -> Import CSS Project
3) Select the a:FreeRTOS Directory



Step 3 – Import a:FreeRTOS



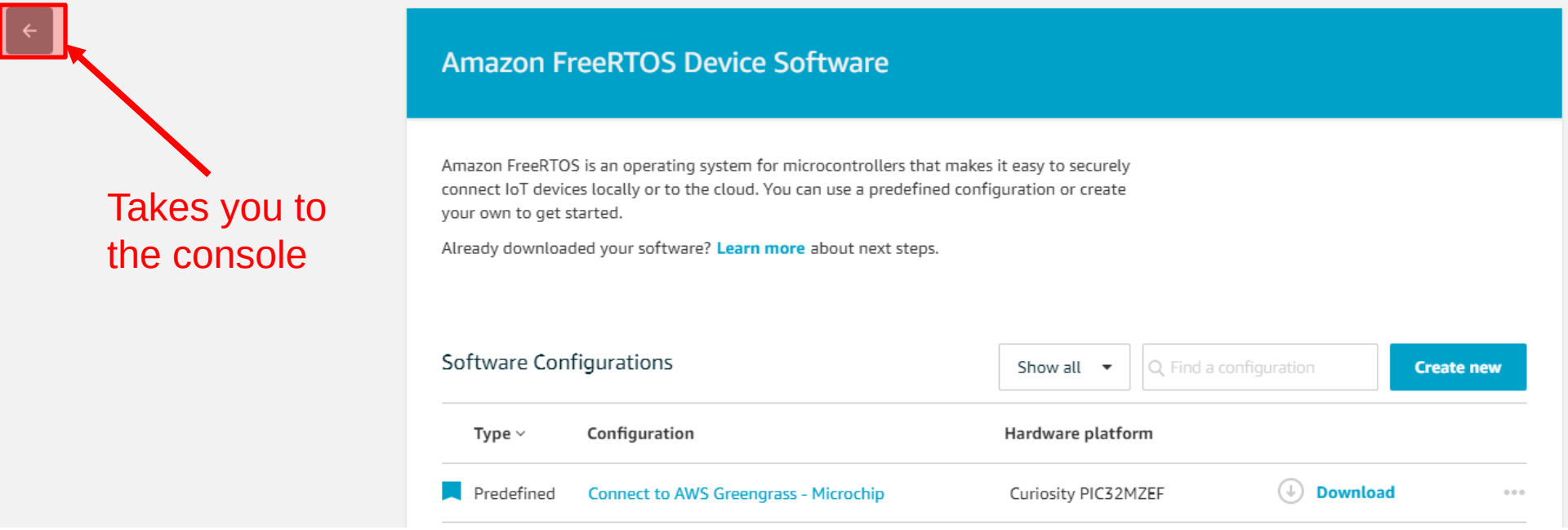
Step 4 – Configure for AWS

aws_clientcredential.h

```
23  * http://www.FreeRTOS.org
24  */
25
26
27 #ifndef __AWS_CLIENTCREDENTIAL__H__
28 #define __AWS_CLIENTCREDENTIAL__H__
29
30 /*
31  * Include for device certificate and private key
32  */
33 #include "aws_clientcredential_keys.h"
34
35 /*
36  * MQTT Broker endpoint.
37  */
38 static const char clientcredentialMQTT_BROKER_ENDPOINT[] = "Paste AWS IoT Broker endpoint here.";
39
40
41 /* Use of a "define" and not a "static const" here to be able to
42 * use pre-compile concatenation on the string. */
43 #define clientcredentialIOT_THING_NAME "Paste AWS IoT Thing name here."
44
```

Step 4 – Configure for AWS

Login to the AWS IoT Console



Step 4 – Configure for AWS

Step 4 – Configure for AWS

aws_clientcredential.h

```
/*
 * WIFI network to join.
 */
#define clientcredentialWIFI_SSID          "Paste WiFi SSID here."

/*
 * Password needed to join WiFi network.
 */
#define clientcredentialWIFI_PASSWORD      "Paste WiFi password here."

/**
 * @brief Security type
 * WPA2 Security, @see WIFISecurity_t
 * other values are - eWiFiSecurityOpen, eWiFiSecurityWEP, eWiFiSecurityWPA
 */
#define clientcredentialWIFI_SECURITY      eWiFiSecurityWPA2
```

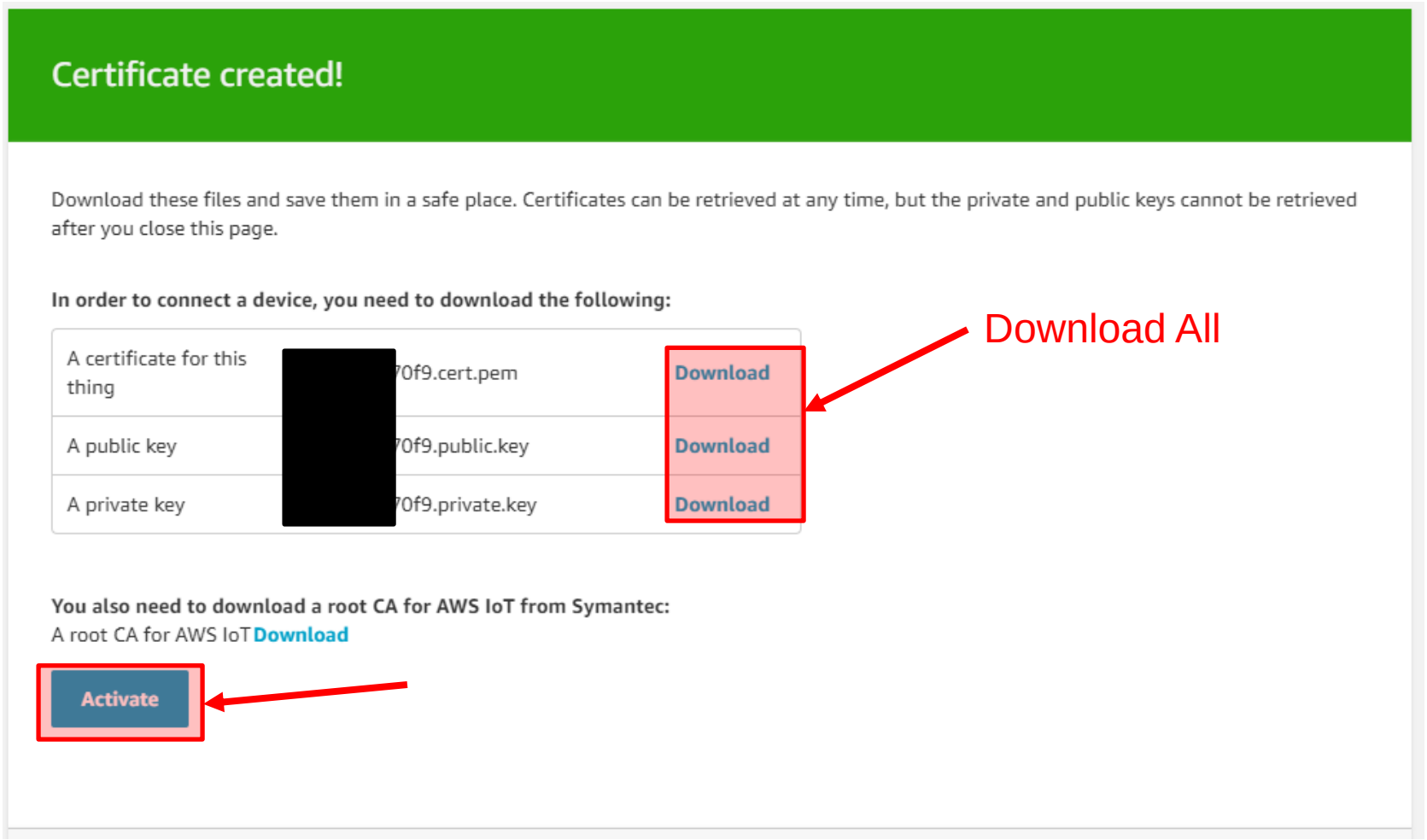
Update for your settings!

Step 4 – Configure for AWS

aws_clientcredential_keys.h

```
/*
 * PEM-encoded client certificate
 *
 * Must include the PEM header and footer:
 * "-----BEGIN CERTIFICATE-----"
 * "...base64 data..."
 * "-----END CERTIFICATE-----";
 */
static const char clientcredentialCLIENT_CERTIFICATE_PEM[] = "Paste client certificate here.";



/*
 * PEM-encoded client private key.
 *
 * Must include the PEM header and footer:
 * "-----BEGIN RSA PRIVATE KEY-----"
 * "...base64 data..."
 * "-----END RSA PRIVATE KEY-----";
 */
static const char clientcredentialCLIENT_PRIVATE_KEY_PEM[] = "Paste client private key here.";


```

Step 4 – Configure for AWS



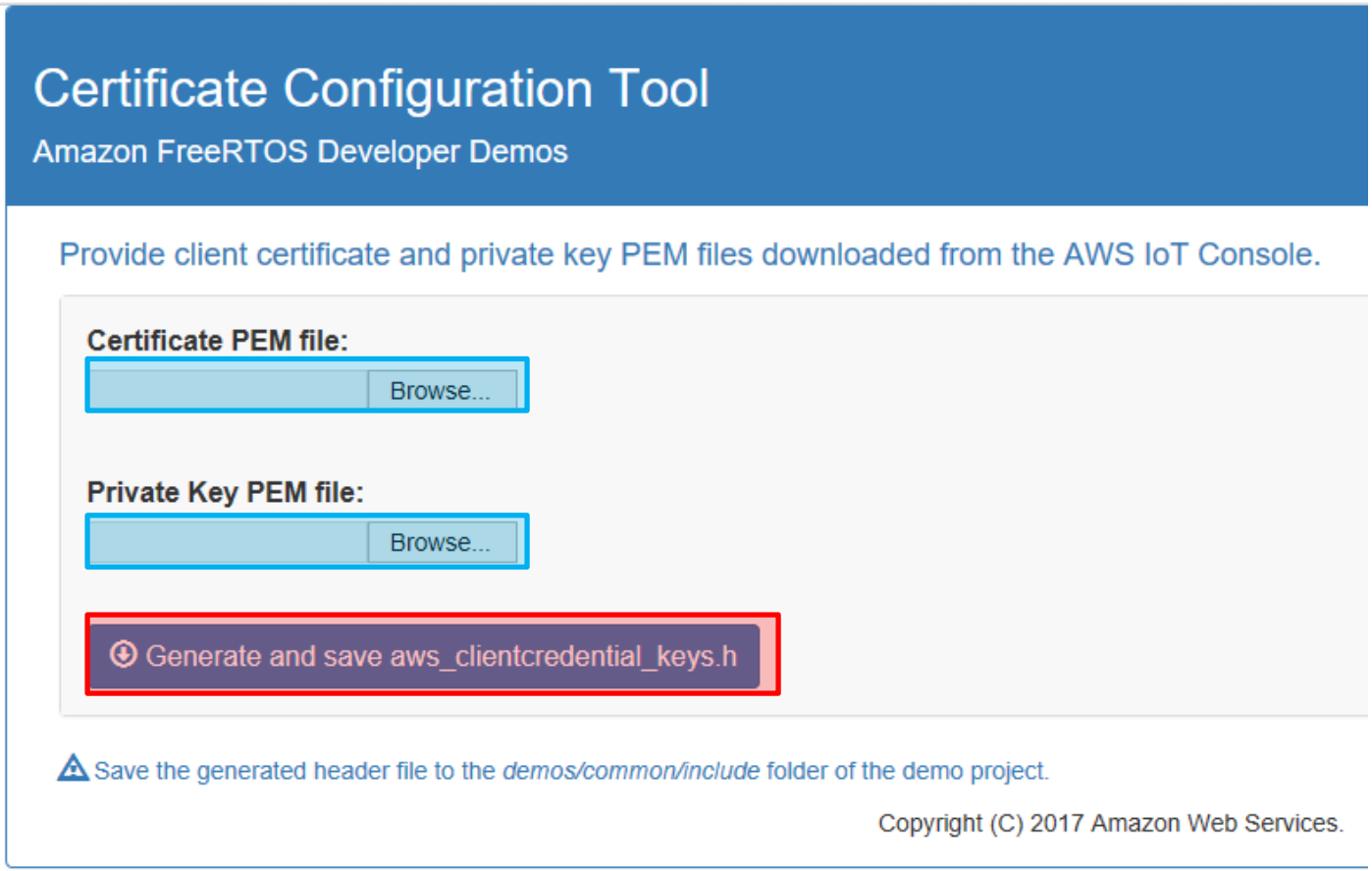
Step 4 – Configure for AWS

Navigate to:

aFreeRTOS\demos\common\devmode_key_provisioning\CertificateConfigurationTool\

Open CertificateConfigurator.html in a browser window.

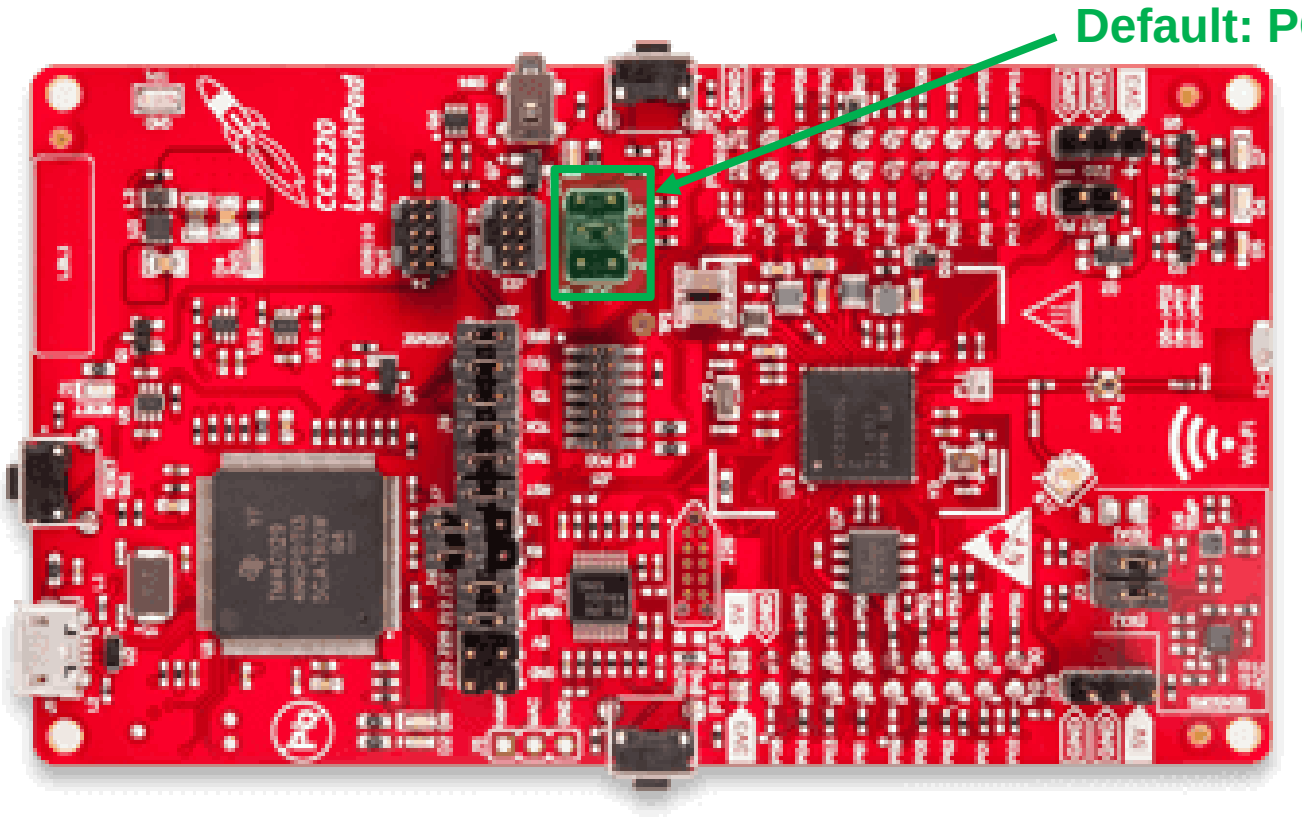
Step 4 – Configure for AWS



Replace existing aws_clientcredential_key.h with generated file!

Step 5 – Run the Example

Make sure the SOP is set to position 0



Step 5 – Run the Example

Compile the project

Console

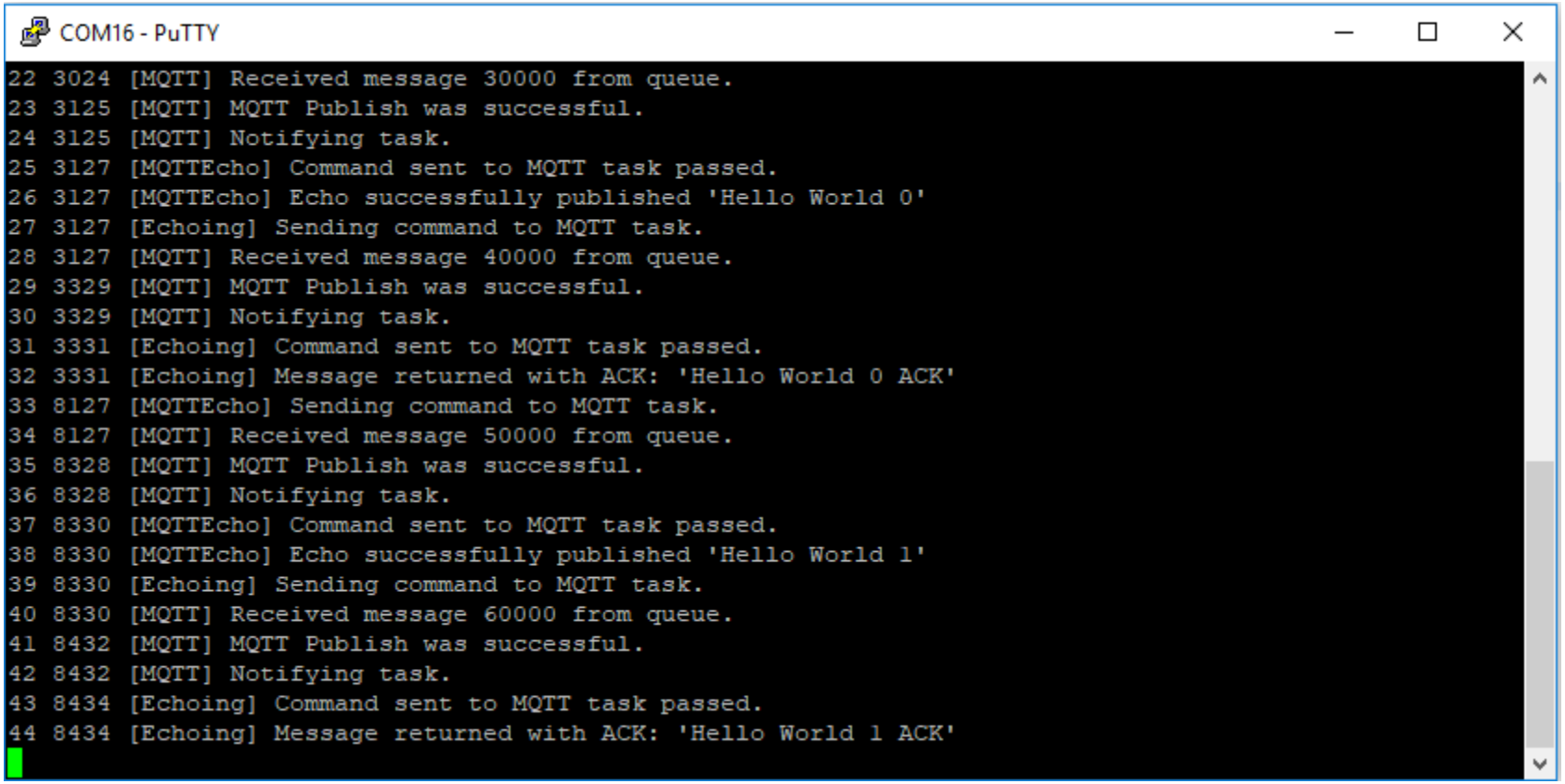
CDT Build Console [aws_demos]

```
"./lib/third_party/mcu_vendor/ti/SimpleLink_CC32xx/v1_40_01_00/kernel/freertos/posix/sleep_freertos.obj"
"./lib/third_party/mcu_vendor/ti/SimpleLink_CC32xx/v1_40_01_00/kernel/freertos/posix/timer_freertos.obj"
"./lib/third_party/mcu_vendor/ti/SimpleLink_CC32xx/v1_40_01_00/kernel/freertos/startup/startup_cc32xx_ccs.obj"
"C:/AmazonFreeRTOS/demos/ti/cc3220_launchpad/common/application_code/ti_code/CC3220SF_LAUNCHXL_FREERTOS.cmd"  -llibc.a -
l"C:/AmazonFreeRTOS/lib/third_party/mcu_vendor/ti/SimpleLink_CC32xx/v1_40_01_00/source/ti/devices/cc32xx/driverlib/ccs/Release/driverlib.a"
l"C:/AmazonFreeRTOS/lib/third_party/mcu_vendor/ti/SimpleLink_CC32xx/v1_40_01_00/source/ti/drivers/lib/drivers_cc32xx.aem4" -
l"C:/AmazonFreeRTOS/lib/third_party/mcu_vendor/ti/SimpleLink_CC32xx/v1_40_01_00/source/ti/drivers/net/wifi/ccs/rtos/simplelink.a"
<Linking>
Finished building target: "aws_demos.out"

C:/ti/ccsv8/utils/tiobj2bin/tiobj2bin aws_demos.out aws_demos.bin C:/ti/ccsv8/tools/compiler/ti-cgt-arm_18.1.1.LTS/bin/armofd
C:/ti/ccsv8/tools/compiler/ti-cgt-arm_18.1.1.LTS/bin/armhex C:/ti/ccsv8/utils/tiobj2bin/mkhex4bin


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

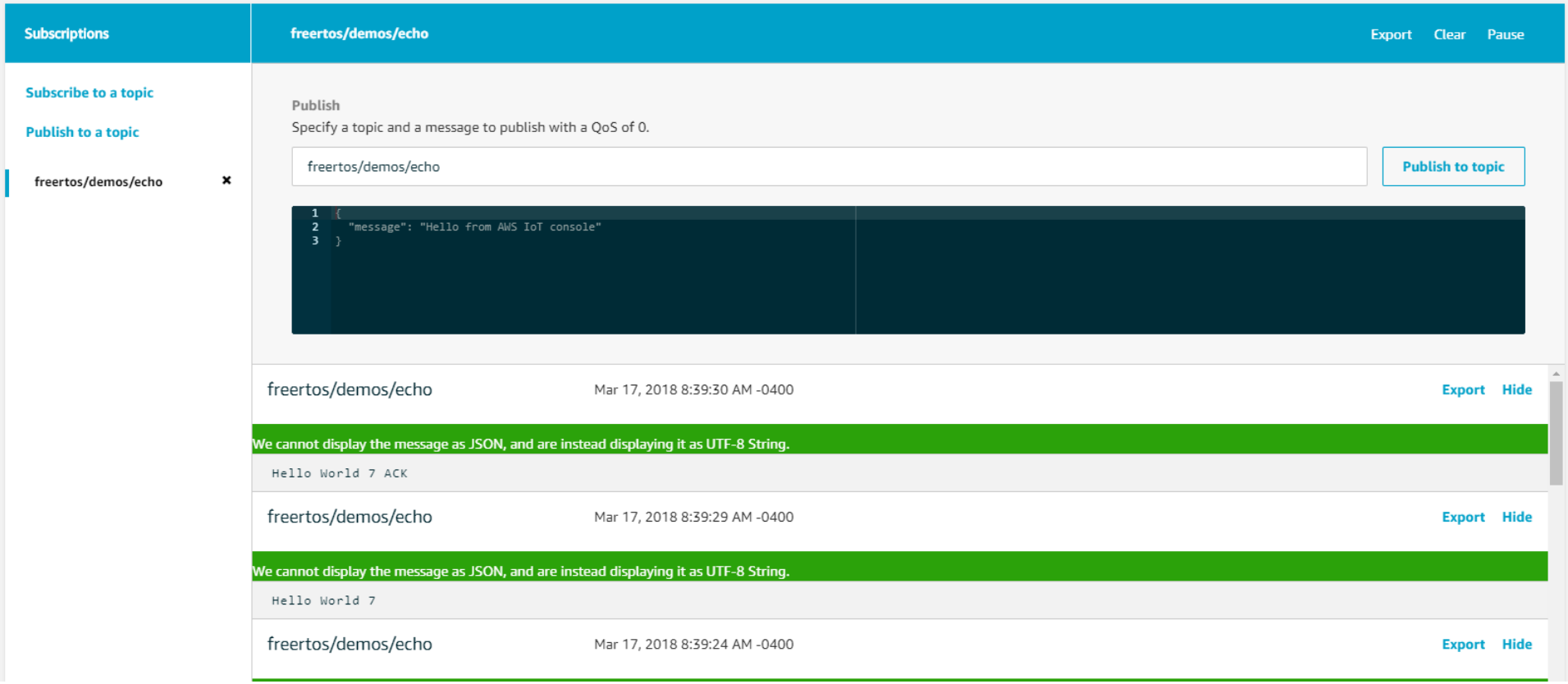
Step 5 – Run the Example

Subscribe to freertos/demos/echo

Step 5 – Run the Example



Step 5 – Run the Example



Additional Resources

- Download Course Material for
  - C/C++ Doxygen Templates
  - Example source code
  - Blog
  - YouTube Videos
- Embedded Bytes Newsletter
  - http://bit.ly/1BAHYXm

From www.beningo.com under

- Blog > CEC – Connecting Edge Devices to the IoT using Amazon FreeRTOS