

Type1LD AWS IoT Core Quick Start Guide

Document Number: N1-5284
Version: 1.0
Release Date: 2020/6/1

Murata Manufacturing Co., Ltd.

Revision History

Revision Number	Release Date	Comments
Revision 1.0	2020/6/1	Initial

Contents

Contents.....	3
1. About this Document.....	4
1.1. Purpose and Scope	4
1.2. Prerequisites	4
2. Evaluation Board.....	4
3. Setting up AWS IoT Core.....	5
4. Setting up Certificate Files for WICED.....	9
5. Checking Your AWS Endpoint	9
6. Running Applications	10
6.1. Running Publisher Application	10
6.2. Running Subscriber Application	13
6.3. Running AWS Shadow Application	16

1. About this Document

1.1. Purpose and Scope

This document provides instructions to communicate with AWS IoT Core on Murata Type1LD EVB.

1.2. Prerequisites

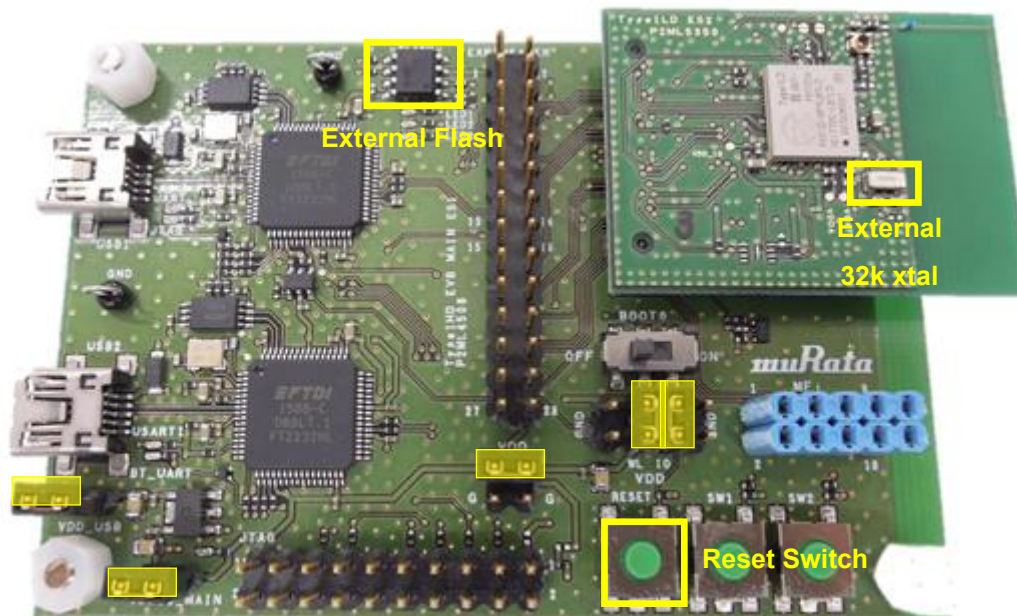
In this guide, it is assumed that you are familiar with WICED SDK. You do not need to be a WICED master, but if you do not know these things, please check Type1LD Evaluation Board Quick Start Guide before moving forward.

- How to create new MakeTarget on WICED SDK
- How to run snip.scan example application on Type1LD EVB
- How to see UART output from snip.scan example application on Tera Term, Putty, etc.

It is also assumed that you have AWS account and you can access [AWS IoT Console](#).

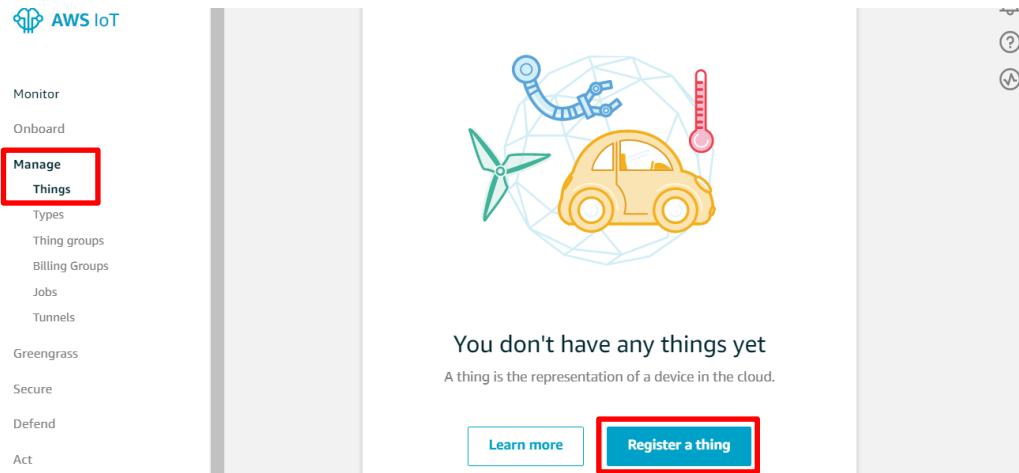
2. Evaluation Board

Verify pin setting for correct operation as below.

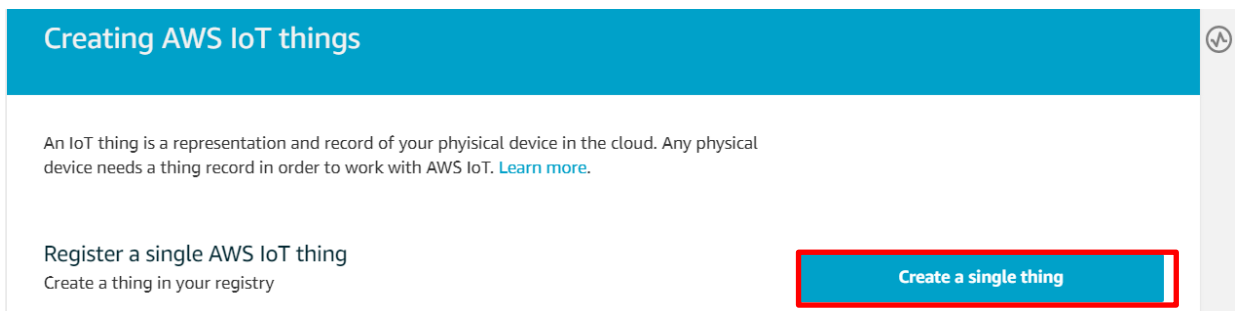


3. Setting up AWS IoT Core

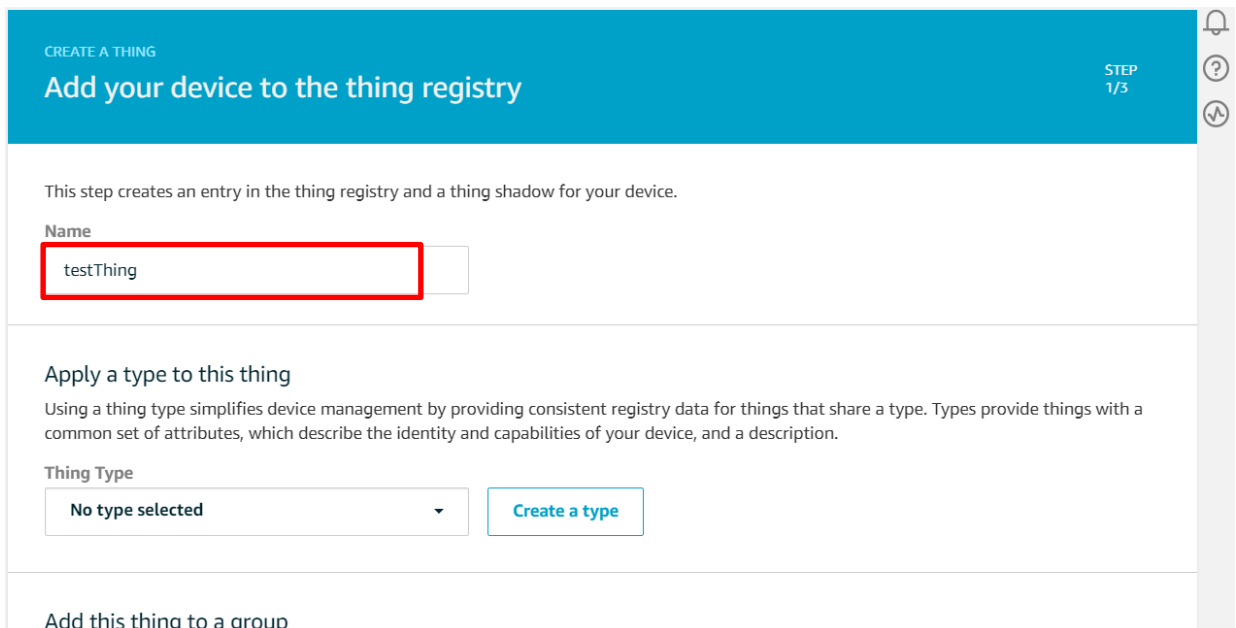
- A. Visit AWS IoT Core Console and select Manage - Things from the left menu, then click Register a thing button.



- B. Click Create a single thing button.



- C. Type "testThing" in the Name field, then scroll down and click Next button.



D. Click Create certificate button

E. Download all of files and save them in a safe place. **Make sure that you download private key and public key** as they cannot be retrieved after you close this page.

F. Regarding root CA for AWS IoT, download RSA 2048 bit key: Amazon Root CA 1.

- G. After making sure you have downloaded private key and public key (and certificate and Amazon root CA), scroll down and click Activate button, then click Done button.

In order to connect a device, you need to download the following:

A certificate for this thing	832e0d81aa.cert.pem	Download
A public key	832e0d81aa.public.key	Download
A private key	832e0d81aa.private.key	Download

You also need to download a root CA for AWS IoT:
A root CA for AWS IoT [Download](#)


[Activate](#)

[Cancel](#) [Done](#) [Attach a policy](#)

- H. Select Secure - Policies from left menu and click Create policy button.

AWS IoT

- Monitor
- Onboard
- Manage
- Greengrass
- Secure**
 - Certificates
 - Policies**
 - CAs
 - Role Aliases
 - Authorizers
- Defend
- Act
- Test



You don't have any policies yet

AWS IoT policies give things permission to access AWS IoT resources (like other things, MQTT topics, or thing shadows).

[Learn more](#) [Create a policy](#)

- I. Type "iot:*" in the Action field, "*" in the Resource ARN field, check Allow and click Create button.

Add statements

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

Action

iot:*

Resource ARN

*

Effect

☒ Allow ☐ Deny

Remove

Add statement

Create

- J. Click Secure - Certificates from left menu then click three dots menu on your certificate, and then click Attach policy.

Monitor

Onboard

Manage

Greengrass

Secure

Certificates

Policies

CAs

Role Aliases

Authorizers

Defend

Act

Search certificates

832e0d81a...
ACTIVE

...

Activate

Deactivate

Revoke

Accept transfer

Reject transfer

Revoke transfer

Start transfer

Attach policy

Attach thing

Download

Delete

Card

- K. Check testPolicy you just created and click Attach button.

Attach policies to certificate(s)

Policies will be attached to the following certificate(s):
832e0d81aa1fae58fa27844df45d3ec5b7ce2ddb7d098f6ddb482d8f97af6cc

Choose one or more policies

Search policies

☒ testPolicy View

1 policy selected

Cancel

Attach

4. Setting up Certificate Files for WICED

You have module certificate and Amazon certificate in previous section “3. Setting up AWS IoT Core”. Rename these files to fit with WICED configuration and place them in the appropriate folder as below.

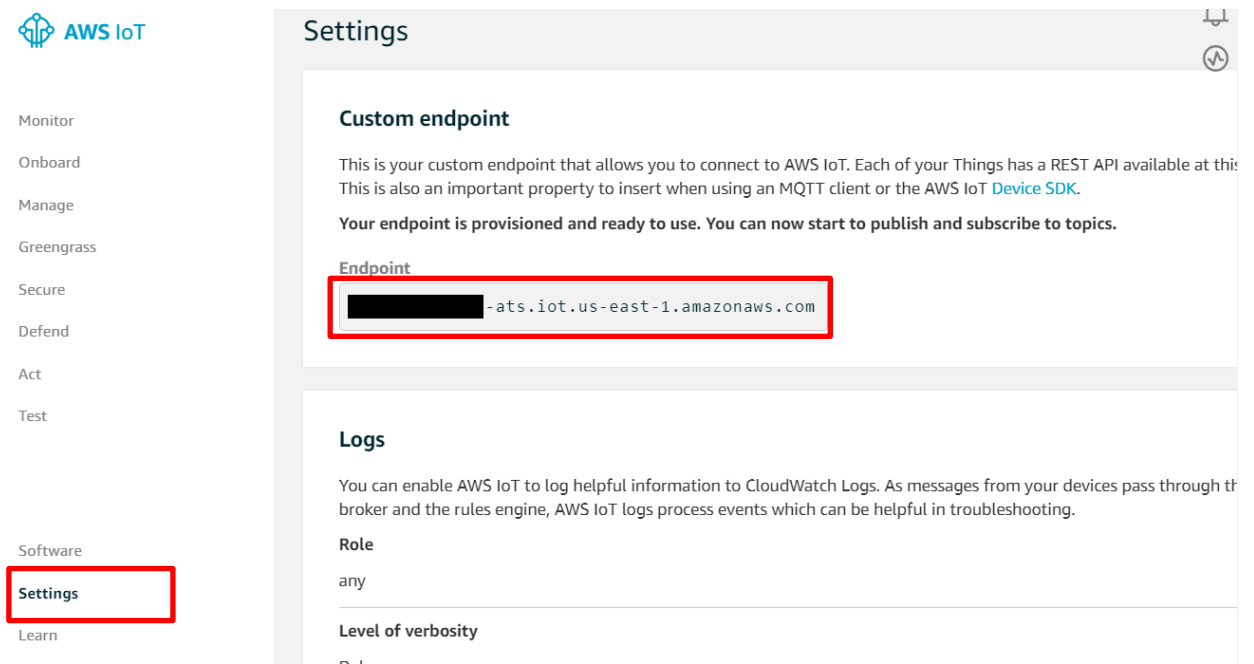
*You must have unique characters for xxxxxxxxxx.

File	Rename to	Location
AmazonRootCA1.pem	rootca.cer	<ul style="list-style-type: none"> 43xxx_Wi-Fi\resources\apps\aws\iot
xxxxxxxxxx-private.pem.key	privkey.cer	<ul style="list-style-type: none"> 43xxx_Wi-Fi\resources\apps\aws\iot\publisher 43xxx_Wi-Fi\resources\apps\aws\iot\subscriber
xxxxxxxxxx-certificate.pem.crt	client.cer	<ul style="list-style-type: none"> 43xxx_Wi-Fi\resources\apps\aws\iot\publisher 43xxx_Wi-Fi\resources\apps\aws\iot\subscriber

Note: WICED folder is at “C:\Users\<user name>\Documents\WICED-Studio-<VERSION>” with default installation.

5. Checking Your AWS Endpoint

Visit AWS IoT Core Console and select Settings from the left menu, then copy the Endpoint. You need this in later sections.



The screenshot shows the AWS IoT Core console interface. On the left, there is a navigation menu with options: Monitor, Onboard, Manage, Greengrass, Secure, Defend, Act, Test, Software, **Settings** (highlighted with a red box), and Learn. The main content area is titled 'Settings' and contains two sections: 'Custom endpoint' and 'Logs'. The 'Custom endpoint' section explains that this is the custom endpoint for connecting to AWS IoT and provides a text box for the endpoint URL. The URL shown is [redacted]-ats.iot.us-east-1.amazonaws.com, which is highlighted with a red box. The 'Logs' section explains how to enable logging to CloudWatch Logs.

6. Running Applications

6.1. Running Publisher Application

This is Wireless Wall Switch example which publishes messages as “WICED_BULB” topic when you push SW1.

- A) Open WICED IDE
- B) Open 43xxx_Wi-Fi\apps\demo\aws\iot\pub_sub\publisher\publisher.c and update the endpoint with yours. If you do not know your endpoint, please check “5. Checking Your AWS Endpoint”.

```

101 static wiced_aws_qos_level_t qos = WICED_AWS_QOS_ATMOST_ONCE;
102
103 static wiced_aws_thing_security_info_t my_publisher_security_creds =
104 {
105     .private_key = NULL,
106     .key_length = 0,
107     .certificate = NULL,
108     .certificate_length = 0,
109 };
110
111 static wiced_aws_endpoint_info_t my_publisher_aws_iot_endpoint = {
112     .transport = WICED_AWS_TRANSPORT_MQTT_NATIVE,
113     .uri = "a38td4ke8seeky-ats.iot.us-east-1.amazonaws.com",
114     .peer_common_name = NULL,
115     .ip_addr = {0},
116     .port = WICED_AWS_IOT_DEFAULT_MQTT_PORT,
117     .root_ca_certificate = NULL,
118     .root_ca_length = 0,
119 };
120

```

- C) Open 43xxx_Wi-Fi\apps\demo\aws\iot\pub_sub\publisher\wifi_config_dct.h and update CLIENT_AP_SSID, CLIENT_AP_PASSPHRASE and CLIENT_AP_SECURITY to match with your access point you will use.

```

47
48 /* This is the soft AP used for device configuration */
49 #define CONFIG_AP_SSID "WICED_AWS"
50 #define CONFIG_AP_CHANNEL 1
51 #define CONFIG_AP_SECURITY WICED_SECURITY_WPA2_AES_PSK
52 #define CONFIG_AP_PASSPHRASE "12345678"
53
54 /* This is the soft AP available for normal operation (if used)*/
55 #define SOFT_AP_SSID "WICED Device"
56 #define SOFT_AP_CHANNEL 1
57 #define SOFT_AP_SECURITY WICED_SECURITY_WPA2_AES_PSK
58 #define SOFT_AP_PASSPHRASE "WICED_PASSPHRASE"
59
60 /* This is the default AP the device will connect to (as a client)*/
61 #define CLIENT_AP_SSID "AWS_IOT_PUB_AP" /* Change this to your AP */
62 #define CLIENT_AP_PASSPHRASE "YOUR_AP_PASSPHRASE"
63 #define CLIENT_AP_BSS_TYPE WICED_BSS_TYPE_INFRASTRUCTURE
64 #define CLIENT_AP_SECURITY WICED_SECURITY_WPA2_MIXED_PSK
65 #define CLIENT_AP_CHANNEL 1
66 #define CLIENT_AP_BAND WICED_802_11_BAND_2_4GHZ

```

- D) Open 43xxx_Wi-Fi\apps\demo\aws\iot\pub_sub\publisher\publisher.mk and add MurataType1LD as a VALID_PLATFORMS.

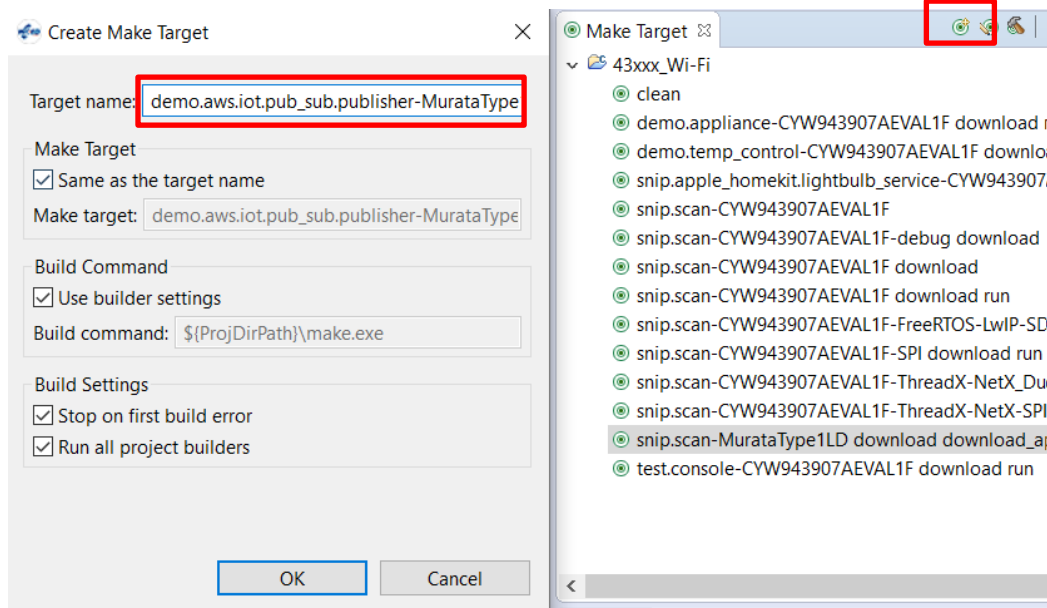
```

37
38 $(NAME)_COMPONENTS := protocols/AWS
39
40 WIFI_CONFIG_DCT_H := wifi_config_dct.h
41
42 $(NAME)_RESOURCES := apps/aws/iot/rootca.cer \
43     apps/aws/iot/publisher/client.cer \
44     apps/aws/iot/publisher/privkey.cer
45
46 # To support Low memory platforms, disabling components which
47 GLOBAL_DEFINES += WICED_CONFIG_DISABLE_SSL_SERVER \
48     WICED_CONFIG_DISABLE_DTLS \
49     WICED_CONFIG_DISABLE_ENTERPRISE_SECURITY \
50     WICED_CONFIG_DISABLE_DES \
51     WICED_CONFIG_DISABLE_ADVANCED_SECURITY_CURV
52
53 VALID_OSNS_COMBOS := ThreadX-NetX_Duo
54 VALID_PLATFORMS := CY8CKIT_062 \
55     CYW943907AEVAL1F \
56     CYW943455EVB_02 \
57     MurataType1LD
58

```

- E) Click New Make Target button and create new Make Target as below

demo.aws.iot.pub_sub.publisher-MurataType1LD download download_apps run



- F) Double click “clean” from the Make Target to make sure you will have the latest files included.
G) Double click the Make Target you just created.
H) While you are waiting the build to complete, open your appropriate COM port with terminal tool such as TeraTerm and set baud rate as 115200bps
I) Wait for a while to complete the build, then you will see Connection Successful message as below.

```
Starting WICED Wiced_006.004.000.0061
Platform MurataType1LD initialised
Started ThreadX v5.8
WICED_core Initialized

Initialising NetX_Duo v5.10_sp3
Creating Packet pools
WLAN MAC Address : 00:9D:6B:59:D7:AC

WLAN Firmware : w10: May 2 2019 02:39:20 version 7.45.98.83 (r714225 CY) FUID 01-476cc09d
WLAN CLM : API: 12.2 Data: 9.10.39 Compiler: 1.29.4 ClmImport: 1.36.3 Creation: 2019-05-02 02:29:53

Obtaining IPv4 address via DHCP
L1420 : dhcp_client_init() : DHCP CLIENT hostname = [WICED IP]
IPv4 network ready IP: 10.0.0.250
Setting IPv6 link-local address
IPv6 network ready IP: FE80:0000:0000:0000:029D:6BFF:FE59:D7AC
[Application/AWS] Opening connection...
[Application/AWS] Try Connecting...
[AWS] AWS endpoint: [redacted]-ats.iot.us-east-1.amazonaws.com is at 52.3[redacted]
[AWS/MQTT] Event received 1
[Application/AWS] Connection Acknowledgment Received
[Application/AWS] Connection Successful...
[Application/AWS] Press the button to Publish Message...
[Application/AWS] Press the button to Publish Message...
[Application/AWS] Press the button to Publish Message...
[Application/AWS] Press the button to Publish Message...
```

- J) Visit AWS IoT Core Console and select Test from the left menu, then type “WICED_BULB” in the Subscription topic field, then click Subscribe to topic button.

Subscriptions

Subscribe to a topic
Publish to a topic

Subscribe
Devices publish MQTT messages on topics. You can use this client to subscribe to a topic and receive these messages.

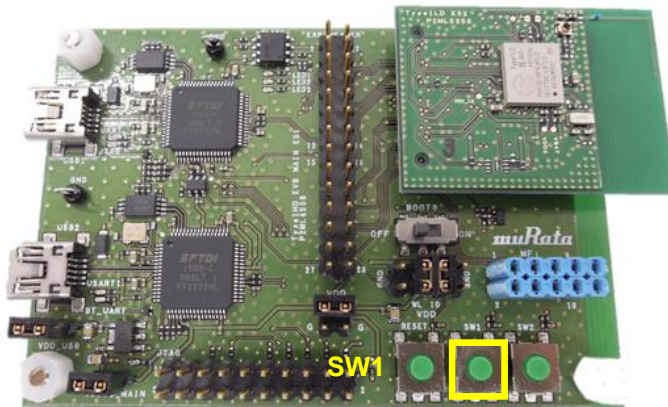
Subscription topic
WICED_BULB

Subscribe to topic

Max message capture ?
100

Quality of Service ?
☒ 0 - This client will not acknowledge to the Device Gateway that messages are received
☐ 1 - This client will acknowledge to the Device Gateway that messages are received

- K) Push SW1 on 1LD EVB.



- L) You will see “LIGHT ON”/”LIGHT OFF” messages on your browser.

Publish
Specify a topic and a message to publish with a QoS of 0.

WICED_BULB

Publish to topic

```
1 {
2   "message": "Hello from AWS IoT console"
3 }
```

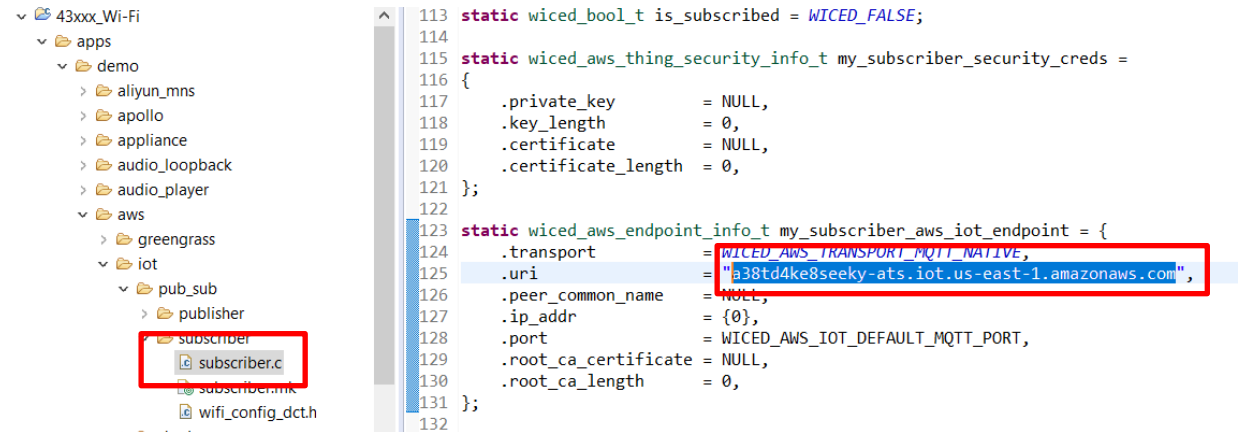
WICED_BULB	May 14, 2020 6:47:45 PM -0700	Export	Hide
LIGHT OFF			
WICED_BULB	May 14, 2020 6:47:42 PM -0700	Export	Hide
LIGHT ON			

6.2. Running Subscriber Application

This is Wireless Light Bulb example which subscribes “WICED_BULB” topic and turn on/off LED1.

A) Open WICED IDE

Open 43xxx_Wi-Fi\apps\demo\aws\iot\pub_sub\subscriber\subscriber.c and update the endpoint with yours. If you do not know your endpoint, please check “5. Checking Your AWS Endpoint”.

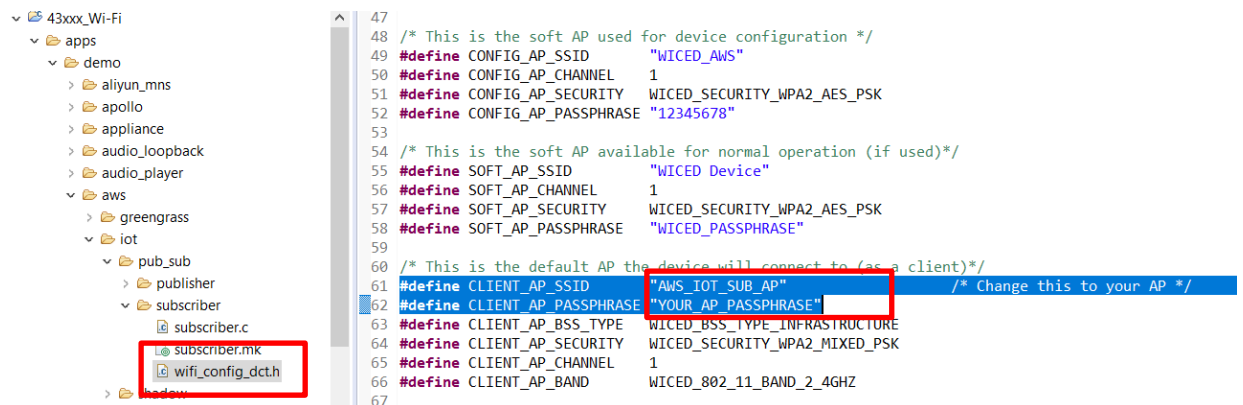


```

113 static wiced_bool_t is_subscribed = WICED_FALSE;
114
115 static wiced_aws_thing_security_info_t my_subscriber_security_creds =
116 {
117     .private_key      = NULL,
118     .key_length       = 0,
119     .certificate       = NULL,
120     .certificate_length = 0,
121 };
122
123 static wiced_aws_endpoint_info_t my_subscriber_aws_iot_endpoint = {
124     .transport         = WICED_AWS_TRANSPORT_MQTT_NATIVE,
125     .uri               = "a38td4ke8seeky-ats.iot.us-east-1.amazonaws.com",
126     .peer_common_name  = NULL,
127     .ip_addr           = {0},
128     .port              = WICED_AWS_IOT_DEFAULT_MQTT_PORT,
129     .root_ca_certificate = NULL,
130     .root_ca_length    = 0,
131 };
132

```

B) Open 43xxx_Wi-Fi\apps\demo\aws\iot\pub_sub\subscriber\wifi_config_dct.h and update CLIENT_AP_SSID, CLIENT_AP_PASSPHRASE and CLIENT_AP_SECURITY to match with your access point you will use.

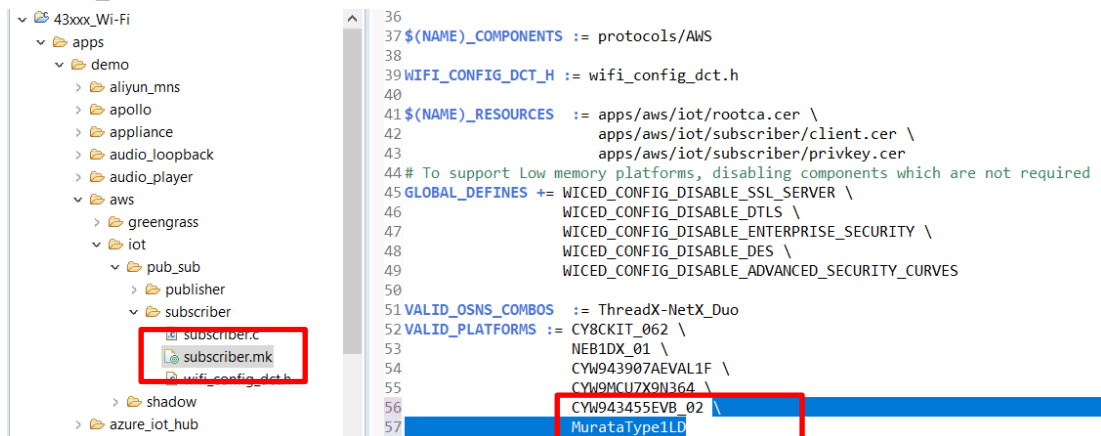


```

47
48 /* This is the soft AP used for device configuration */
49 #define CONFIG_AP_SSID "WICED_AWS"
50 #define CONFIG_AP_CHANNEL 1
51 #define CONFIG_AP_SECURITY WICED_SECURITY_WPA2_AES_PSK
52 #define CONFIG_AP_PASSPHRASE "12345678"
53
54 /* This is the soft AP available for normal operation (if used)*/
55 #define SOFT_AP_SSID "WICED Device"
56 #define SOFT_AP_CHANNEL 1
57 #define SOFT_AP_SECURITY WICED_SECURITY_WPA2_AES_PSK
58 #define SOFT_AP_PASSPHRASE "WICED_PASSPHRASE"
59
60 /* This is the default AP the device will connect to (as a client)*/
61 #define CLIENT_AP_SSID "AWS_IOT_SUB_AP" /* Change this to your AP */
62 #define CLIENT_AP_PASSPHRASE "YOUR_AP_PASSPHRASE"
63 #define CLIENT_AP_BSS_TYPE WICED_BSS_TYPE_INFRASTRUCTURE
64 #define CLIENT_AP_SECURITY WICED_SECURITY_WPA2_MIXED_PSK
65 #define CLIENT_AP_CHANNEL 1
66 #define CLIENT_AP_BAND WICED_802_11_BAND_2_4GHZ
67

```

C) Open 43xxx_Wi-Fi\apps\demo\aws\iot\pub_sub\subscriber\subscriber.mk and add MurataType1LD as a VALID_PLATFORMS.



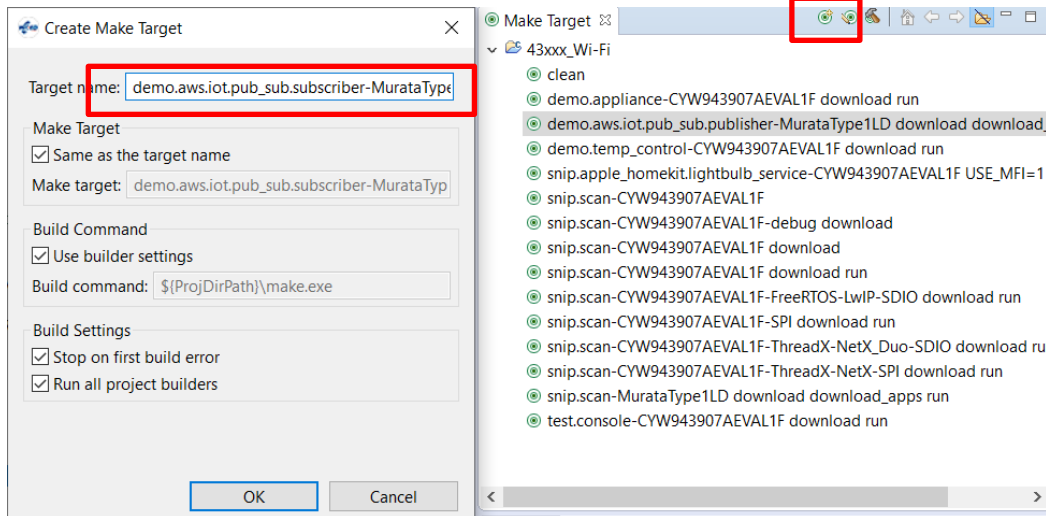
```

36
37 $(NAME)_COMPONENTS := protocols/AWS
38
39 WIFI_CONFIG_DCT_H := wifi_config_dct.h
40
41 $(NAME)_RESOURCES := apps/aws/iot/rootca.cer \
42     apps/aws/iot/subscriber/client.cer \
43     apps/aws/iot/subscriber/privkey.cer
44 # To support Low memory platforms, disabling components which are not required
45 GLOBAL_DEFINES += WICED_CONFIG_DISABLE_SSL_SERVER \
46     WICED_CONFIG_DISABLE_DTLS \
47     WICED_CONFIG_DISABLE_ENTERPRISE_SECURITY \
48     WICED_CONFIG_DISABLE_DES \
49     WICED_CONFIG_DISABLE_ADVANCED_SECURITY_CURVES
50
51 VALID_OSNS_COMBOS := ThreadX-NetX_Duo
52 VALID_PLATFORMS := CY8CKIT_062 \
53     NEB1DX_01 \
54     CYN943907AEVAL1F \
55     CYN943907AEVAL1F \
56     CYN943455EVB_02 \
57     MurataType1LD

```

- D) Click New Make Target button and create new Make Target as below

demo.aws.iot.pub_sub.subscriber-MurataType1LD download download_apps run



- E) Double click “clean” from the Make Target to make sure you will have the latest files included.
- F) Double click the Make Target you just created.
- G) While you are waiting the build to complete, open your appropriate COM port with terminal tool such as TeraTerm and set baud rate as 115200bps
- H) Wait for a while to complete the build, then you will see Connection Successful message as below.

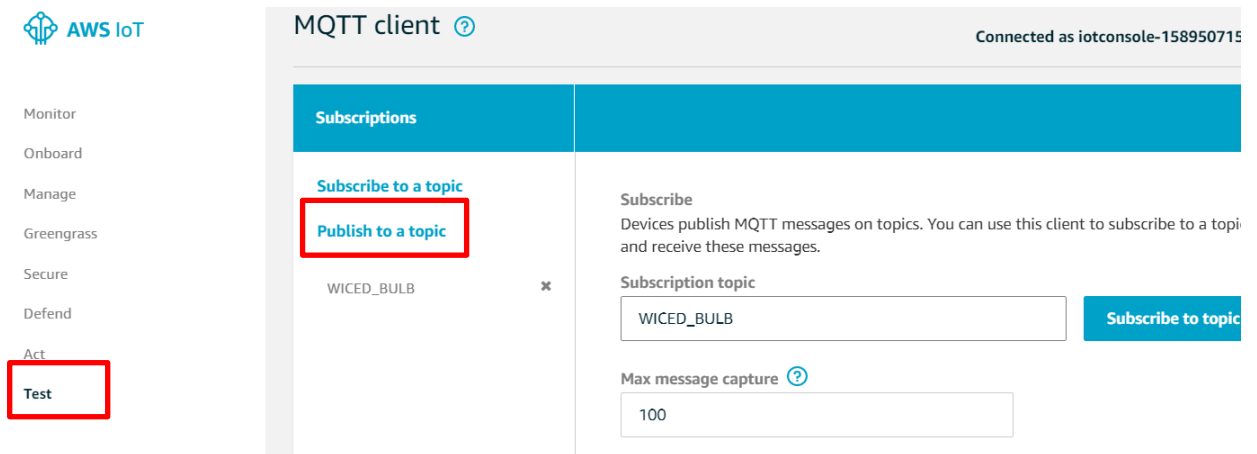
```
Starting WICED Wiced_006.004.000.0061
Platform MurataType1LD initialised
Started ThreadX v5.8
WICED_core Initialized

Initialising NetX_Duo v5.10_sp3
Creating Packet pools
WLAN MAC Address : 00:9D:6B:59:D7:AC

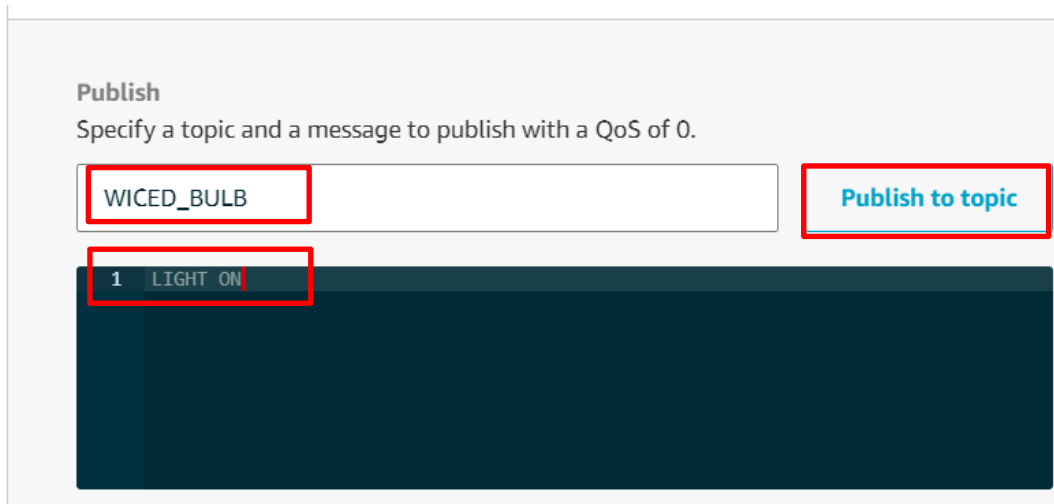
WLAN Firmware : w10: May 2 2019 02:39:20 version 7.45.98.83 <r714225 CY> FWID 01-476cc09d
WLAN CLM : API: 12.2 Data: 9.10.39 Compiler: 1.29.4 ClmImport: 1.36.3 Creation: 2019-05-02 02:29:53

Obtaining IPv4 address via DHCP
L1420 : dhcp_client_init() : DHCP CLIENT hostname = [WICED IP]
IPv4 network ready IP: 10.0.0.250
Setting IPv6 link-local address
IPv6 network ready IP: FE80:0000:0000:0000:029D:6BFF:FE59:D7AC
[Application/AWS] Opening connection!
[Application/AWS] Try Connecting...
[AWS] AWS endpoint: [redacted]-ats.iot.us-east-1.amazonaws.com is at 34.19[redacted]
[AWS/MQTT] Event received 1
[Application/AWS] Connection Acknowledgment Received
[Application/AWS] Connection Successful...
[Application/AWS] Subscribing Topic WICED_BULB 1 ...
[AWS/MQTT] Event received 4
[Application/AWS] Subscribe Acknowledgment Received
```

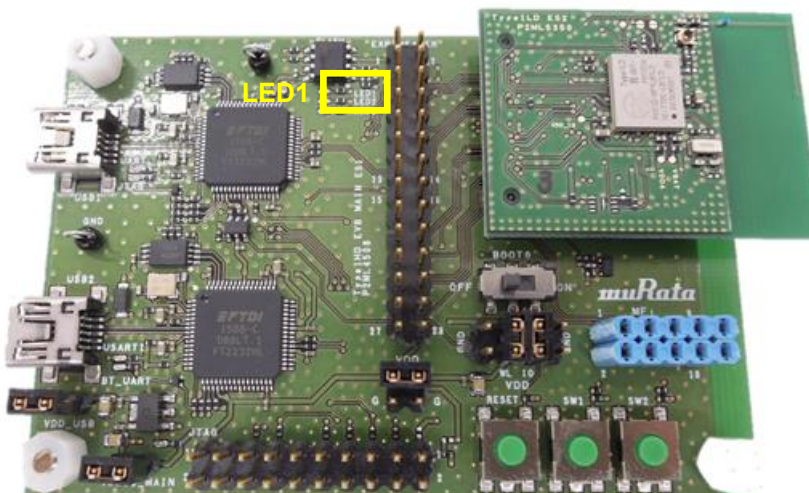

- I) Visit AWS IoT Core Console and select Test from the left menu, then click Publish to a topic.



- J) Type "WICED_BULB" in the topic field and "LIGHT ON" in the message field, then click Publish to topic button.



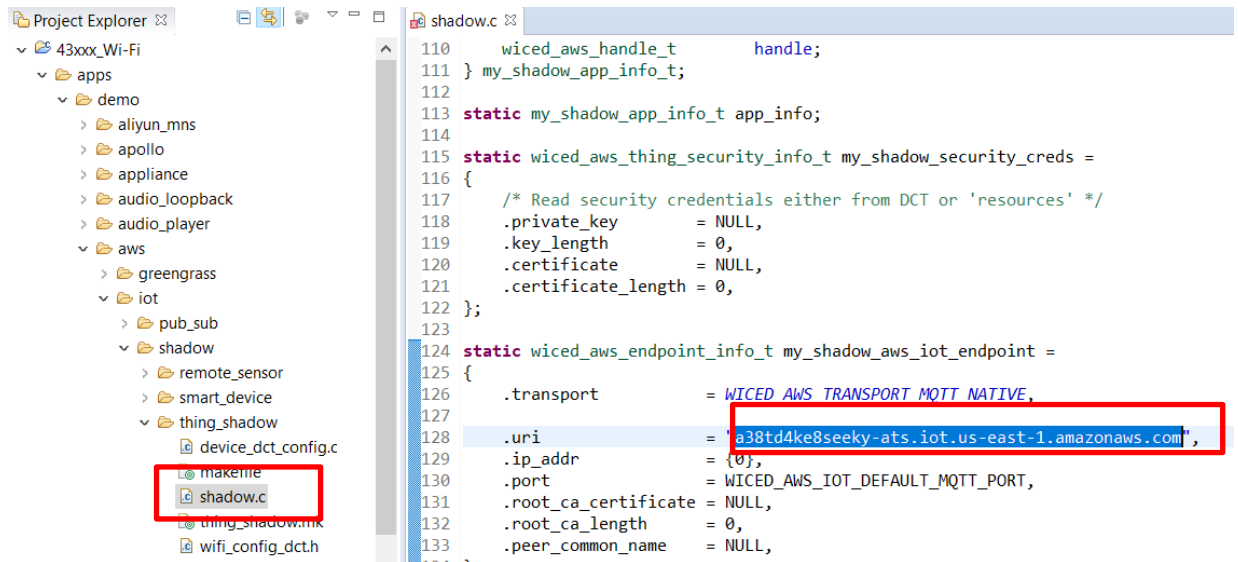
- K) You will see "light on" on your Tera Term and LED1 on EVB turned on. You can send "LIGHT OFF" message to turn of the LED.



6.3. Running AWS Shadow Application

This is Wireless Light Bulb example which has Shadow on AWS IoT and turn on/off LED1 with WiFi initialization steps as SoftAP.

- A) Open WICED IDE
- B) Open 43xxx_Wi-Fi\apps\demo\aws\iot\shadow\thing_shadow\shadow.c and update the endpoint with yours. If you do not know your endpoint, please check “5. Checking Your AWS Endpoint”.

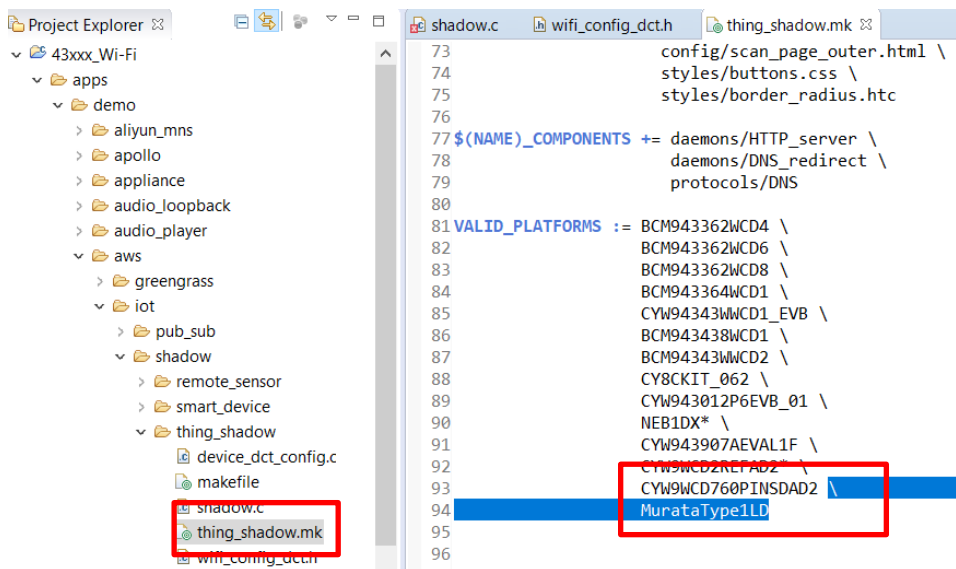


```

110 wiced_aws_handle_t handle;
111 } my_shadow_app_info_t;
112
113 static my_shadow_app_info_t app_info;
114
115 static wiced_aws_thing_security_info_t my_shadow_security_creds =
116 {
117     /* Read security credentials either from DCT or 'resources' */
118     .private_key = NULL,
119     .key_length = 0,
120     .certificate = NULL,
121     .certificate_length = 0,
122 };
123
124 static wiced_aws_endpoint_info_t my_shadow_aws_iot_endpoint =
125 {
126     .transport = WICED_AWS_TRANSPORT_MQTT_NATIVE,
127     .uri = "a38td4ke8seeky-ats.iot.us-east-1.amazonaws.com",
128     .ip_addr = {0},
129     .port = WICED_AWS_IOT_DEFAULT_MQTT_PORT,
130     .root_ca_certificate = NULL,
131     .root_ca_length = 0,
132     .peer_common_name = NULL,
133 }

```

- C) Open 43xxx_Wi-Fi\apps\demo\aws\iot\shadow\thing_shadow\thing_shadow.mk and add MurataType1LD as a VALID_PLATFORMS.

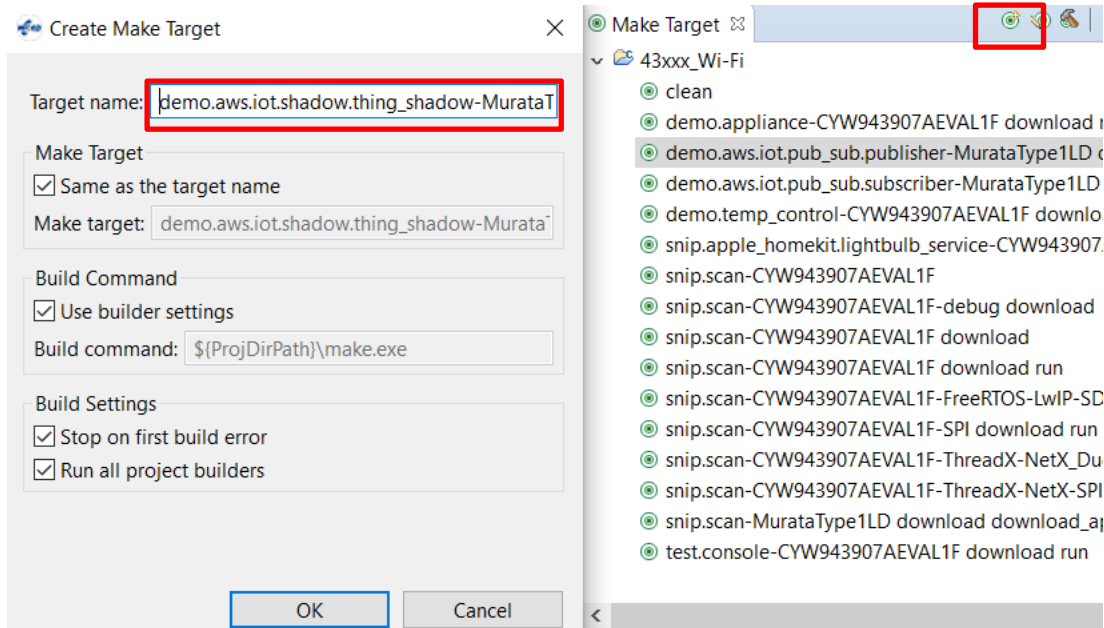


```

73 config/scan_page_outer.html \
74 styles/buttons.css \
75 styles/border_radius.htc
76
77 $(NAME)_COMPONENTS += daemons/HTTP_server \
78 daemons/DNS_redirect \
79 protocols/DNS
80
81 VALID_PLATFORMS := BCM943362WCD4 \
82 BCM943362WCD6 \
83 BCM943362WCD8 \
84 BCM943364WCD1 \
85 CYW94343WCD1_EVB \
86 BCM943438WCD1 \
87 BCM94343WCD2 \
88 CY8CKIT_062 \
89 CYW943012P6EVB_01 \
90 NEB1DX* \
91 CYW943907AEVAL1F \
92 CYW943907AEVAL2* \
93 CYW943907AEVAL2 \
94 MurataType1LD
95
96

```


- D) Click New Make Target button and create new Make Target as below
demo.aws.iot.shadow.thing_shadow-MurataType1LD download download_apps run



- M) Double click the Make Target you just created.
N) While you are waiting the build to complete, open your appropriate COM port with terminal tool such as TeraTerm and set baud rate as 115200bps
E) Wait for a while to complete the build, then you will see below.

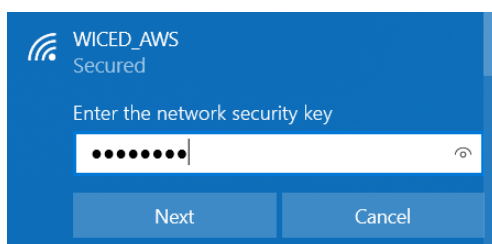
```
Starting WICED Wiced_006.004.000.0061
Platform MurataType1LD initialised
Started ThreadX v5.8
WICED_core Initialized

Initialising NetX_Duo v5.10_sp3
Creating Packet pools
WLAN MAC Address : 00:9D:6B:59:D7:AC

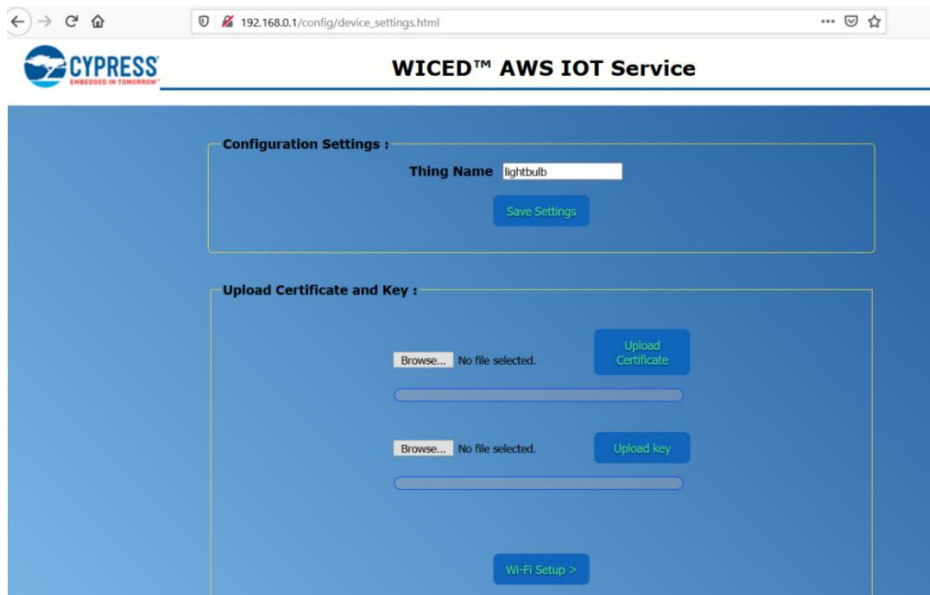
WLAN Firmware : w10: May 2 2019 02:39:20 version 7.45.98.83 (r714225 CY) FWID 01-476cc09d
WLAN CLM : API: 12.2 Data: 9.10.39 Compiler: 1.29.4 ClmImport: 1.36.3 Creation: 2019-05-02 02:29:53

Please wait, connecting to network...
(To return to SSID console screen, hold USER switch for 5 seconds during RESET to clear DCT configuration)
IPv4 network ready IP: 192.168.0.1
Setting IPv6 link-local address
IPv6 network ready IP: FE80:0000:0000:0000:009D:6BFF:FE59:D7AC
*****
Config SSID : WICED_AWS
Password : 12345678
IP Address : 192.168.0.1
*****
```

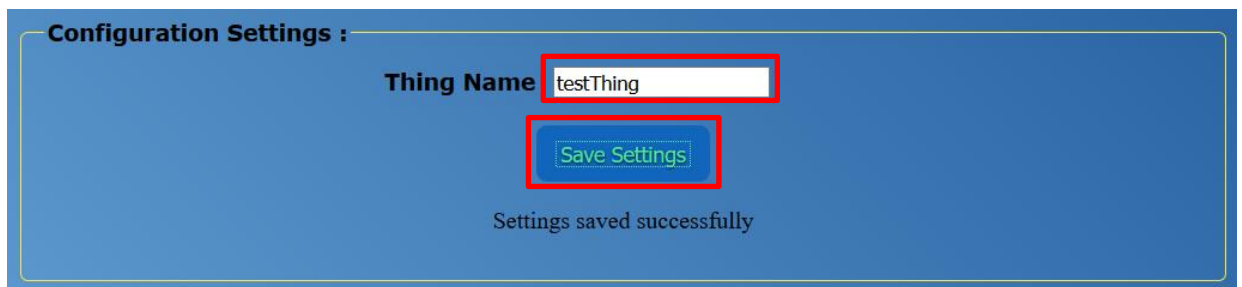
- F) 1LD is now configured in soft-AP mode. Connect to the SSID WICED_AWS, from your laptop using the Password in the terminal.



- G) Open the web browser and access the URL <http://192.168.0.1>



- H) Type “testThing” in the Thing Name field and click Save Settings button. Thing Name must match with the name you created on AWS IoT Core Console.



- I) Click the first Choose File button and select 43xxx_Wi-Fi\resources\apps\aws\iot\subscriber\client.cer, then click Upload Certificate button.

Then click the second Choose File button and select 43xxx_Wi-Fi\resources\apps\aws\iot\subscriber\privkey.cer, then click Upload key button.



Note) In some environment, Chrome doesn't work fine. Please try Firefox instead if you see this error message in later step: [Shadow] Failed to Initialize Wiced AWS library

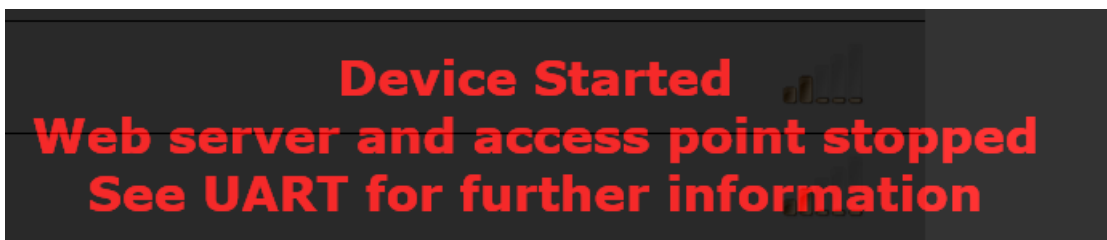
- J) Click Wi-Fi Setup > button, then select SSID of your access point and type the password. Click Connect button.



✓ Scan Complete



- K) 1LD will restart.



- L) Once 1LD is restarted, you shall see the below

```
Starting WICED Wiced_006.004.000.0061
Platform MurataType1LD initialised
Started ThreadX v5.8
WICED_core Initialized

Initialising NetX_Duo v5.10_sp3
Creating Packet pools
WLAN MAC Address : 00:9D:6B:59:D7:AC

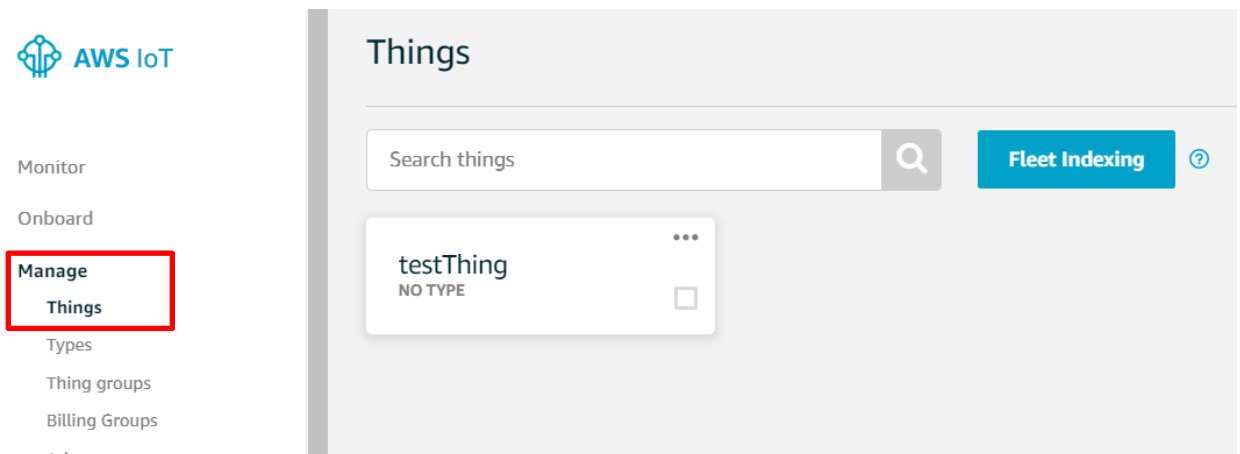
WLAN Firmware : wl0: May 2 2019 02:39:20 version 7.45.98.83 (r714225 CY) FWID 01-476cc09d
WLAN CLM : API: 12.2 Data: 9.10.39 Compiler: 1.29.4 ClmImport: 1.36.3 Creation: 2019-05-02 02:29:53

Please wait, connecting to network...
(To return to SSID console screen, hold USER switch for 5 seconds during RESET to clear DCT configuration)
Joining : 
Successful
Obtaining IPv4 address via DHCP
L1420 : dhcp_client_init() : DHCP CLIENT hostname = [WICED IP]

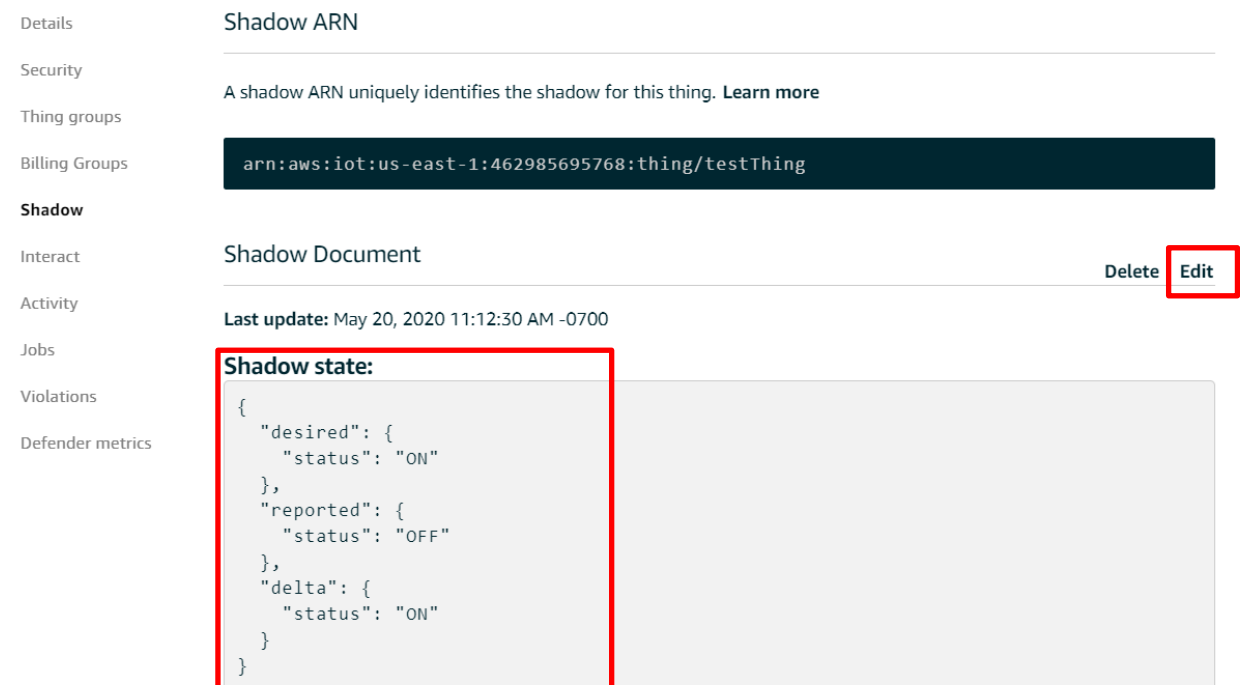
IPv4 network ready IP: 10.0.0.251
Setting IPv6 link-local address
IPv6 network ready IP: FE80:0000:0000:0000:029D:6BFF:FE59:D7AC
[Shadow] Reading Device's certificate and private key from DCT...
[Shadow] Thing Name: testThing
[Shadow] Shadow State Topic: $aws/things/testThing/shadow/update
[Shadow] Shadow Delta Topic: $aws/things/testThing/shadow/update/delta
[AWS] AWS endpoint: -ats.iot.us-east-1.amazonaws.com is at 35.
[AWS/MQTT] Event received 1
[AWS/MQTT] Event received 3
[AWS/MQTT] Event received 3
[AWS/MQTT] Event received 4
```

Note) In some environment, Chrome fails uploading certificate and private key in step I). If you see this error message, please try Firefox instead: [Shadow] Failed to Initialize Wiced AWS library

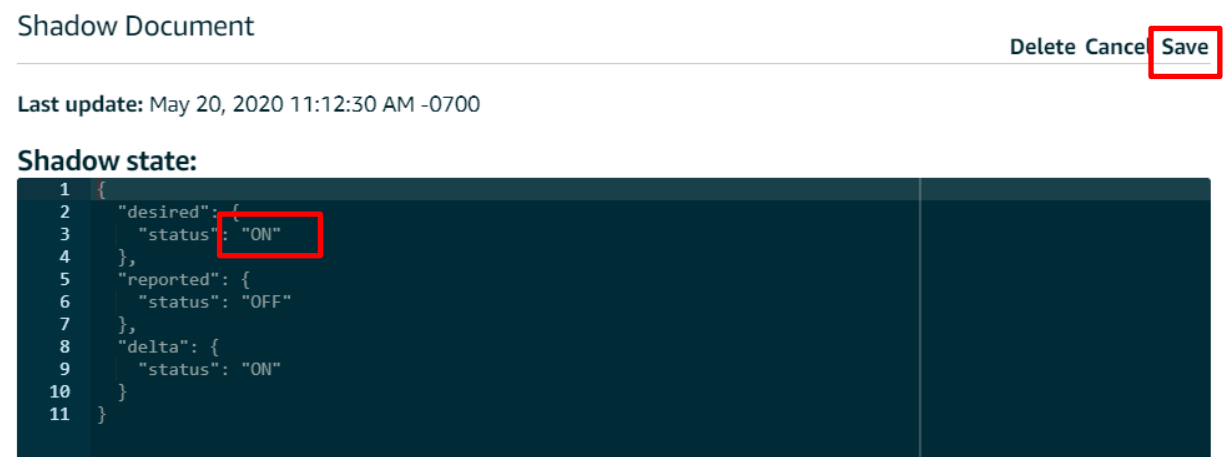
M) Visit AWS IoT Core Console and select Manage - Things from the left menu, then click testThing.



N) You will see below if everything is going well. Click Edit to update 1LD status.

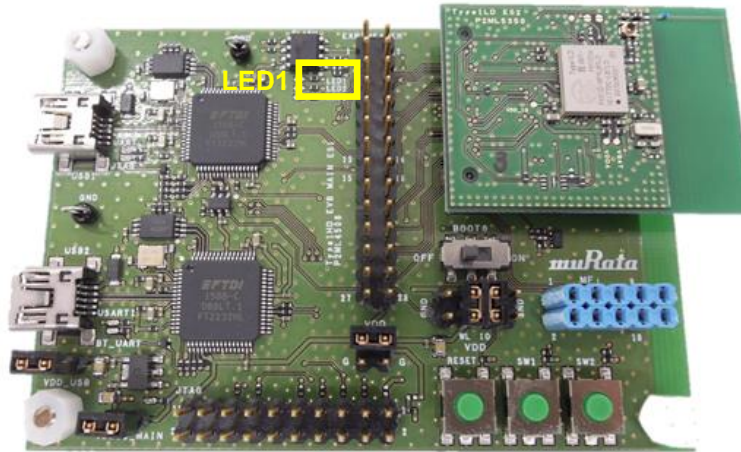


O) Confirm the desired status is ON, then click Save.



- P) You will see Received Payload message on your Tera Term and LED1 on EVB turned on. You can send OFF as desired status to turn off the LED.

```
[Shadow] Received Payload [ Topic: $aws/things/testThing/shadow/update/delta ] :  
====  
<"version":5,"timestamp":1589998544,"state":{"status":"ON"},"metadata":{"status":{"timestamp":1589998544}}>  
====  
[Shadow] LED State OFF[current] ---> ON[requested]  
[AWS/MQTT] Event received 3
```



(END)