

# Avnet BCM4343W IoT Starter Kit Tutorial - Part 1

v2.0 - Nov 29, 2016

# Online Video Material

It is strongly recommended that you view the short technical videos available on the BCM4343W IoT Starter Kit webpage:

<http://cloudconnectkits.org/product/avnet-bcm4343w-iot-starter-kit>

- Part 1: BCM4343W IoT Starter Kit - Board Introduction
- Part 2: BCM4343W IoT Starter Kit - Amazon Web Services
- Part 3: BCM4343W IoT Starter Kit - IBM Bluemix Cloud Services



# Re-installing the Out-of-Box “AWS Shadow” Application



3



29 November 2016



# Basic Installation Items Requiring Attention!

Kits shipped from our initial production have passed factory tests but require re-installation of the out-of-box “Shadow” application...

**Four items must** be attended to on your computer:

- #1** Make sure you have 32bit and 64bit JRE versions installed!  
(prior to installing WICED SDK)
- #2** Install the latest Cypress WICED tool suite (WICED Studio 4.x or later)
- #3** Verify the USB JTAG Driver installation
- #4** Reflash the IoT Starter Kit hardware (using .bat file)

**Pay Attention**  
or pay the price.

# Key Download Links



Download + install **32bit *and* 64bit JRE** (Java Runtime Environment)

<http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>

Download + install **Cypress WICED STUDIO 4.x** SDK tool

<http://www.cypress.com/products/wiced-software>

Download + install **Avnet IoT Starter Kit App Reflash Tool (v2)**

<http://cloudconnectkits.org/sites/default/files/Avnet%20IoT%20Starter%20Kit%20App%20Reflash%20v2.zip>

Download the **BCM4343W IoT Starter Kit Getting Started Guide**

<http://cloudconnectkits.org/product/avnet-bcm4343w-iot-starter-kit>



5



29 November 2016



# JRE32 and JRE64 are both Needed!

#1

## Procedure:

- 1) Check/Fix your JRE (Java Runtime Environment) installation:  
Both the **32 bit JRE** and **64 bit JRE** versions are required!
- 2) 32-bit JRE is needed for Cypress's WICED SDK  
(utilizes a 32-bit version Eclipse based IDE),
- 3) 64-bit JRE is needed for the SDK's installer (JRE is designed to allow  
both 32 and 64 bit variants to be installed on same system)
- 4) Not normally required, but if you have a JRE related issue, check your  
Windows PATH. This should include a path to your Java installation, ie.  
**C:\ProgramData\Oracle\Java\javapath**

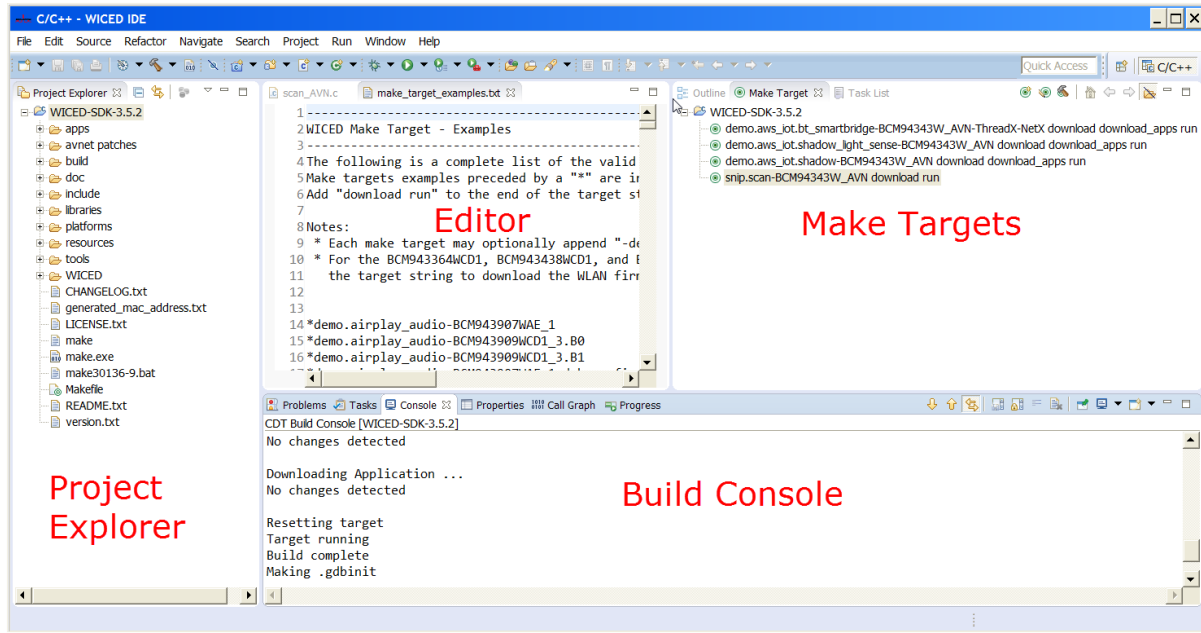
32 bit *and*  
64 bit JRE



# Download + Install Cypress WICED SDK

#2

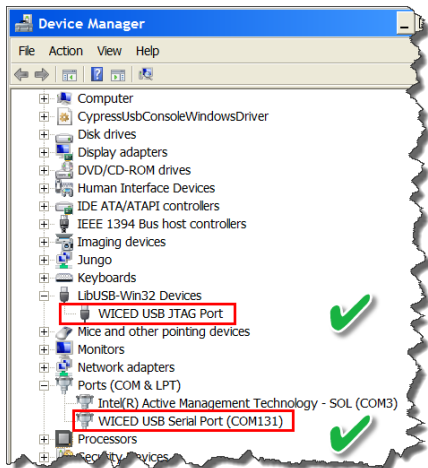
- 5) Download + install **Cypress WICED Studio 4.x** development tools  
<https://community.cypress.com/community/wiced-wifi/wiced-wifi-documentation>
- 6) It is *not* yet necessary to launch the SDK, but the USB drivers included with this installation will be used in the next step...



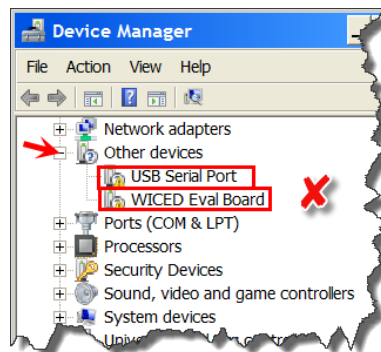
# Verify the USB Drivers Installed Correctly...

#3

- 7) Connect the USB cable from BCM4343W IoT Starter Kit to your computer
- 8) The USB JTAG and USB UART ports should locate and install drivers that were included in the earlier **WICED SDK** installation
- 9) Use Windows Device Manager to check these USB UART and USB JTAG ports  
The USB ports should list as shown on the left below...



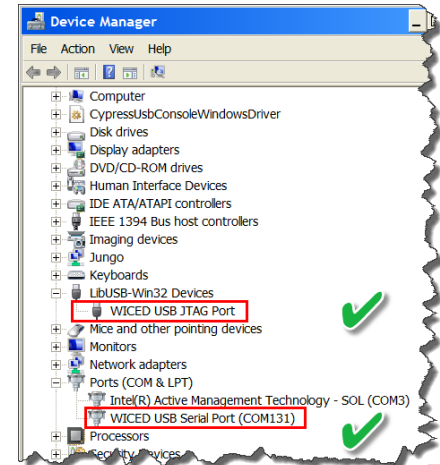
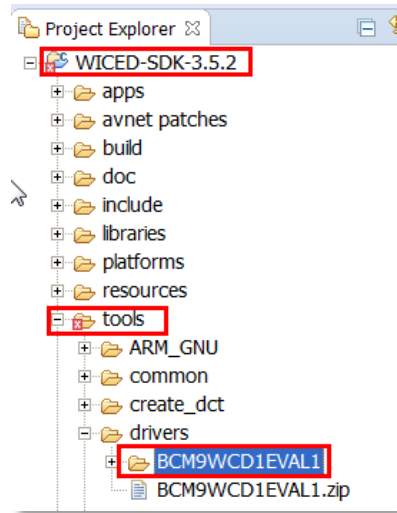
Otherwise these will need to be installed manually...



# Correcting USB Driver Installation (in event of failure)

#3

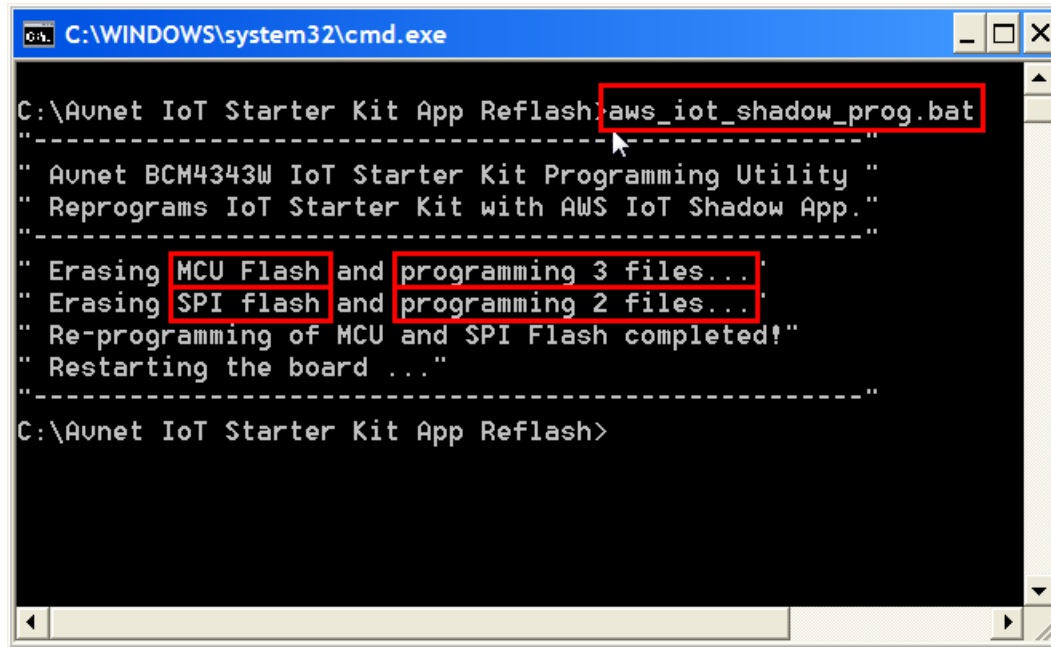
- 10) In the event the USB drivers did not install correctly with WICED SDK, manually install these drivers as follows:
- 11) Using Windows File Explorer go to: `...\tools\drivers\BCM9WCD1EVAL1`
- 12) Double-click `dpinst_x64.exe` to run the **Device Driver Install Wizard**
- 13) Use Windows Device Manager to check these USB ports now list correctly...



# Reflash the IoT Starter Kit hardware (using .bat file)

#4

- 14) Unzip the **Avnet IoT Starter Kit App Reflash v2.zip** to a convenient folder
- 15) Using a command window, **cd** to this folder, then execute **aws\_iot\_shadow\_prog.bat** (this takes about ~1 minute to complete)



```
C:\WINDOWS\system32\cmd.exe
C:\Avnet IoT Starter Kit App Reflash>aws_iot_shadow_prog.bat
-----
" Avnet BCM4343M IoT Starter Kit Programming Utility "
" Reprograms IoT Starter Kit with AWS IoT Shadow App."
-----
" Erasing MCU Flash and programming 3 files..."
" Erasing SPI flash and programming 2 files..."
" Re-programming of MCU and SPI Flash completed!"
" Restarting the board ..."
-----
C:\Avnet IoT Starter Kit App Reflash>
```

# AWS and Kit Configuration for the Out-of-Box “Shadow” Application



11



29 November 2016



# Shadow App. - Items Requiring Attention!

Five additional items require attention to configure and execute the out-of-box “**Shadow**” application...

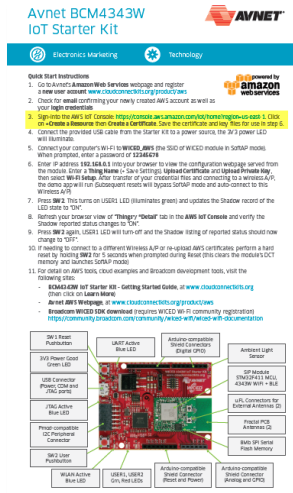
- #5 Use the *Getting Started Guide* (in addition to *QuickStart Instructions*)
- #6 Register an AWS Account at **CloudConnectKits.org**
- #7 Generate in AWS IoT Console: Thing Name, Certificates and Policy
- #8 Configure the Hardware: Thing Name, Certificate + Private Key, WiFi Settings
- #9 Generate in AWS IoT Console: Rules for SNS Push Notifications

**Pay Attention**  
or pay the price.


# Use the Getting Started Guide!

#5

- **QuickStart Instructions** are provided in the kit's box but this lacks detail for a key step when working with the default AWS Shadow application...
- Use the procedure detailed in the **"Getting Started Guide"** to create Thing, Certificate and Policy resources!





## Avnet BCM4343W IoT Starter Kit



Electronics Marketing Technology

### Quick Start Instructions

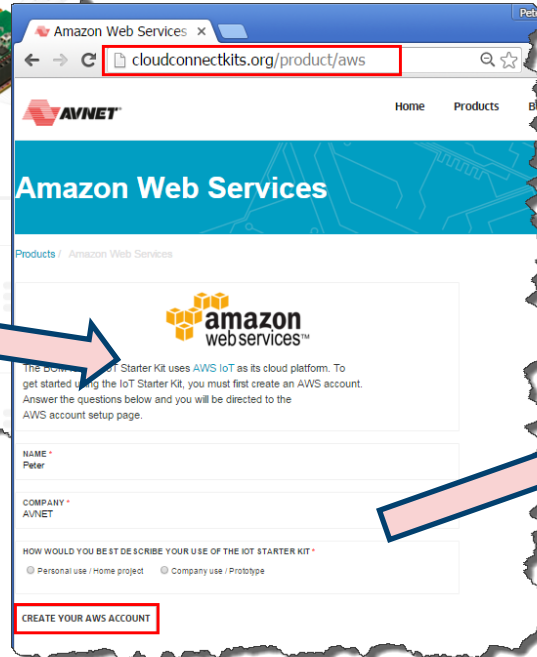
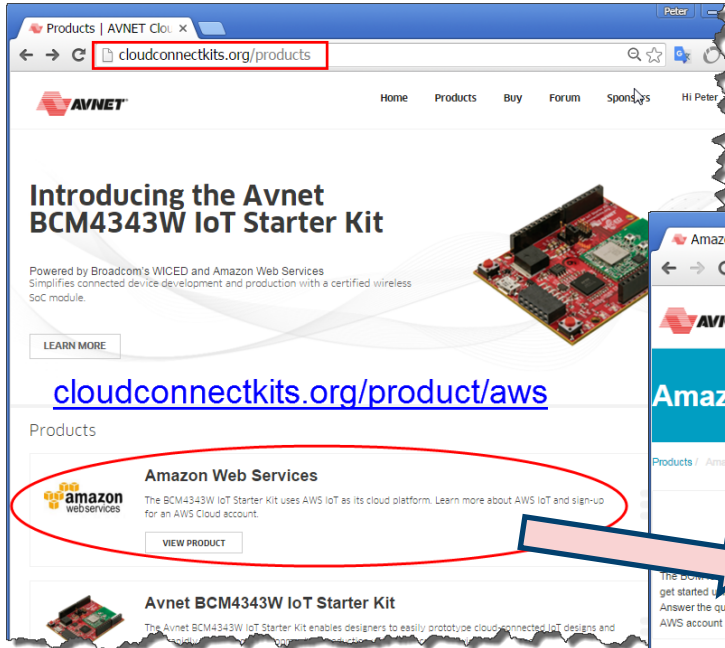
1. Go to Avnet's **Amazon Web Services** webpage and register a **new user account** [www.cloudconnectkits.org/product/aws](http://www.cloudconnectkits.org/product/aws)
2. Check for **email** confirming your newly created AWS account as well as your **login credentials**
3. Sign-into the AWS IoT Console: <https://console.aws.amazon.com/iot/home?region=us-east-1>. Click on **+Create a Resource** then **Create a Certificate**. Save the certificate and key files for use in step 6.
4. Connect the provided USB cable from the Starter Kit to a power source, the 3V3 power LED will illuminate.
5. Connect your computer's Wi-Fi to **WICED\_AWS** (the SSID of WICED module in SoftAP mode). When prompted, enter a password of **12345678**
6. Enter IP address **192.168.0.1** into your browser to view the configuration webpage copied from



# Register an AWS Account at CloudConnectKits.org

#6

<http://cloudconnectkits.org/product/aws>



# Notes Regarding AWS Account



AWS IoT

- Your **Credit Card** is required when registering! (but you are unlikely to incur charges unless the “Free Tier” limits are exceeded...)
- Pay as you go. No minimum fees
- **\$5 per million** messages published to, or delivered in US East (N. Virginia), US West (Oregon), EU (Ireland) **\$8** in Asia Pacific (Tokyo)



## Free Tier

250,000 Messages Per Month  
(Free for first 12 Months)

# Generate Thing Name, Certificates and Policy

#7

- 1) Once you have registered your AWS account, Login to AWS, then click on **AWS IoT**
- 2) Follow the AWS IoT Console procedure in **Getting Started Guide** for generating the **Thing Shadow** plus associated **Certificates** and **Policy**
- 3) Download + save to convenient location the **public key**, **private key** and **certificate** (only **private key** and **certificate** are needed)

Internet of Things  
AWS IoT  
Connect Devices to the Cloud



- Download public key
- Download private key
- Download certificate



# Starting the Shadow Application

#8

- Connect USB cable from BCM4343W IoT Starter Kit to your computer
- Open a serial console, select correct COM port, configured for 115200 baud
- Press kit's Reset button and examine the serial console messages...
- If DCT memory not previously programmed, the Shadow App will start with the BCM4343W in SoftAP mode
- In your computer's network settings, connect the computer to **WICED\_AWS** (password = **12345678**)
- This allows you to open a configuration webpage in browser at **192.168.0.1** to enter AWS IoT configuration details and your WiFi network settings...

```
COM131:115200baud - Tera Term VT
File Edit Setup Control Window Help
Starting WICED v3.5.2
Platform BCM94343W_AVN initialised
Started ThreadX v5.6
Initialising NetX_Duo v5.7_sp2
Creating Packet pools
WWD SDIO interface initialised
WLAN MAC Address : B0:38:29:3D:3A:7D

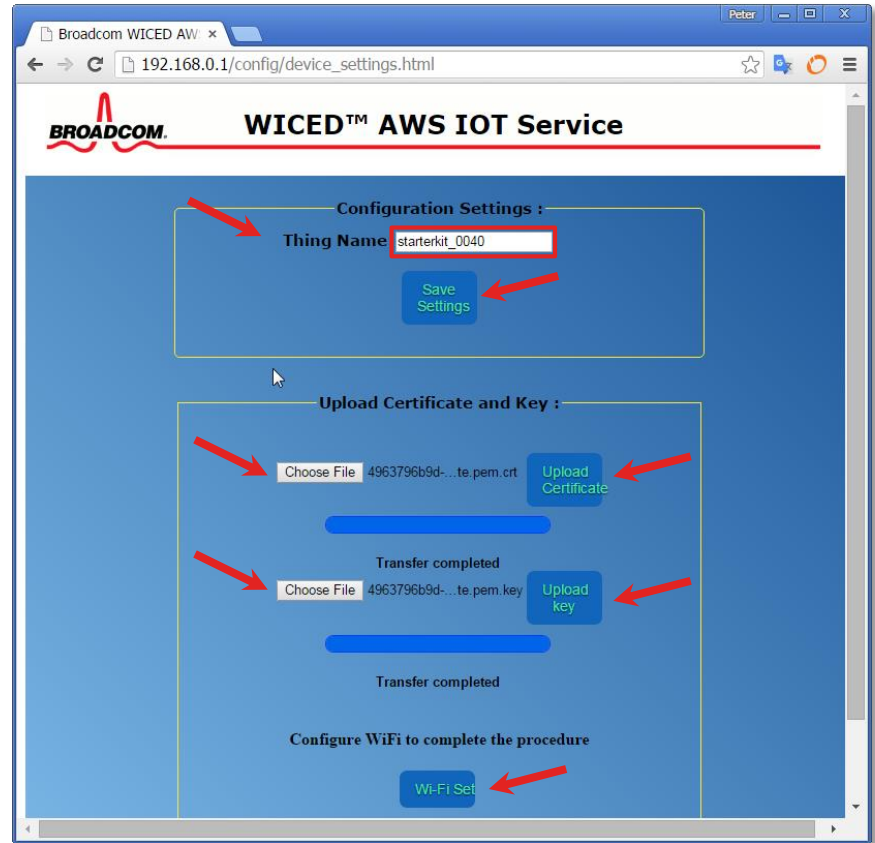
WLAN Firmware : w10: Nov 25 2015 12:57:14 version 7.45.45.1 (r602358) FwID 01-1920c040

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:B238:29FF:FE3D:3A7D
*****
Config SSID : WICED_AWS
Password : 12345678
IP Address : 192.168.0.1
*****
```

# SoftAP Mode Configuration Settings Webpage

#8

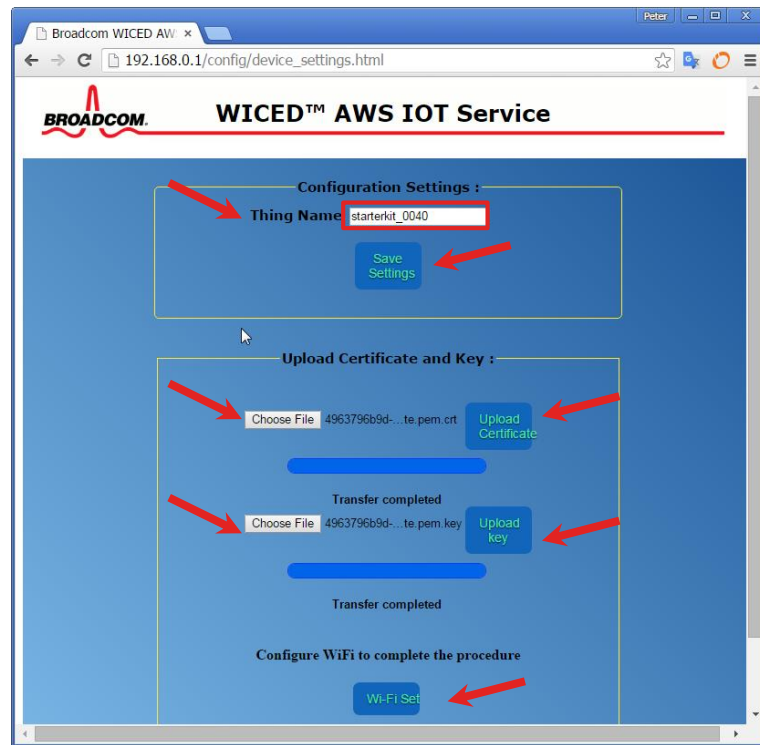
- Computer network = **WICED\_AWS**  
SoftAP password = **12345678**  
Browser address = **192.168.0.1**  
This brings-up the kit's SoftAP Configuration Settings screen
- (You need to have setup an Amazon account and from AWS IoT Console generated the files needed here)
- Note: Each setting must be followed by pressing it's adjacent **Save/Upload/Set** buttons...



# Two Options for Configuring Your Hardware

#8

- **SoftAP mode Config Settings** webpage (available on startup) allows you to enter:
  - The **Thing Name**
  - **Certificate** and **Private Key**
  - **WiFi Network Settings**
- Alternatively, “**bake-in**” these settings! Convenient for development, this is done by **re-naming** and **copying** the downloaded certificate files into:
  - `...\resources\apps\aws_iot`and making edits to two source files:
  - `aws_config.h`
  - `wifi_config_dct.h`
- For more info, refer to *Getting Started Guide*



(AWS IoT Config Settings Webpage)

# Testing the “AWS Shadow” App



20



29 November 2016



# Shadow: Simple Pub/Sub + Thing Shadow Demo



## Avnet BCM4343W IoT Starter Kit:

- User Pushbutton state is “Published” to the Thing Shadow
- User LED controlled by “Subscription” to the Thing Shadow
- The User LED toggles On/Off in response to remote user changes to the Thing Shadow (persistent record)

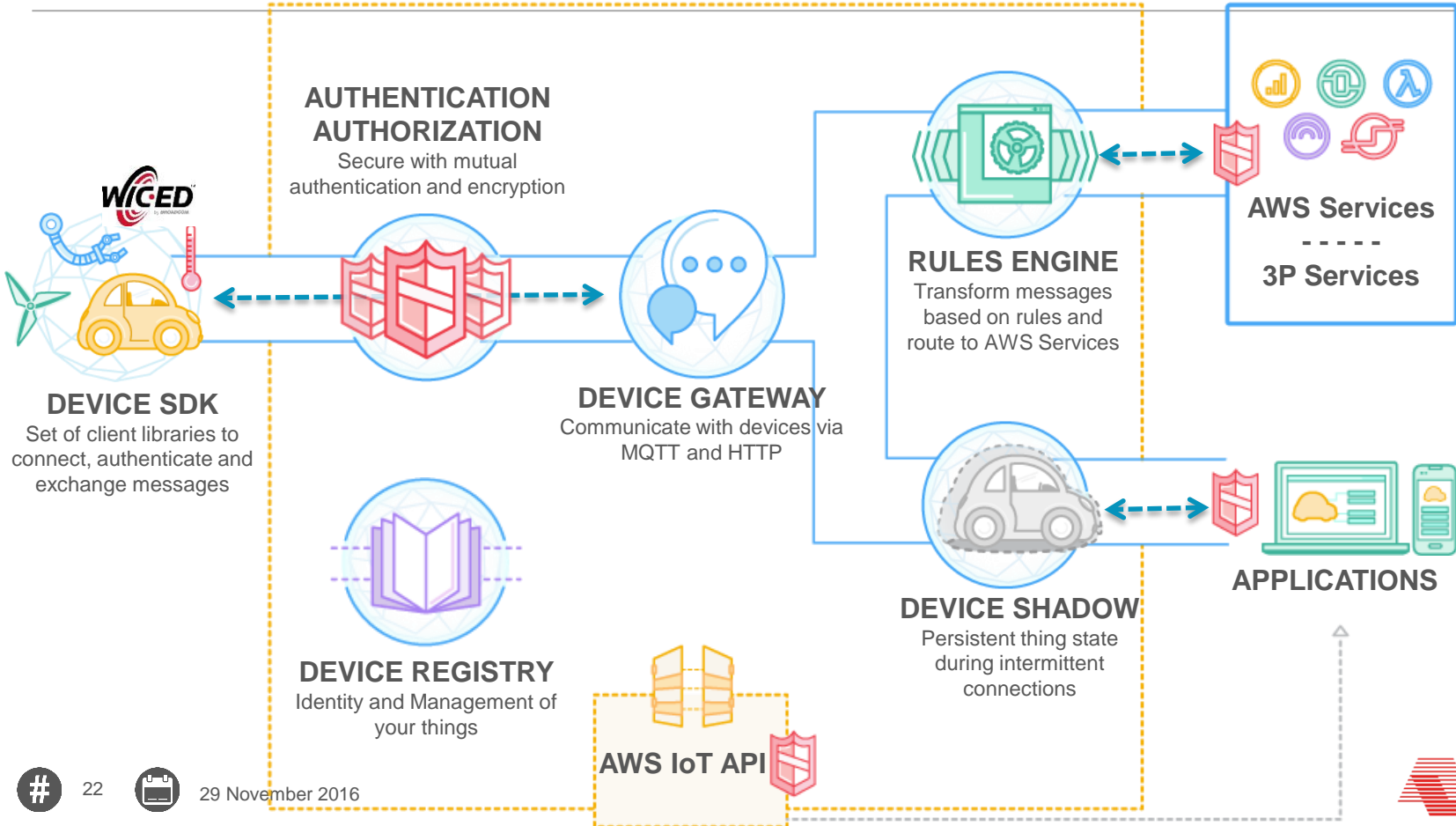
## AWS IoT:

- Stores BCM4343W IoT Kit’s User Pushbutton state in a Thing Shadow
- Allows Cloud-based updates to the User LED on IoT Kit
- Rules Engine: Set-up to send SMS text or email notification from AWS SNS service when User Pushbutton is pressed, or when the Shadow copy of the LED state is updated

## Mobile Device:

- Receives SMS alert from AWS SNS service
- Receives email alert from AWS SNS service

# AWS IoT



# Using the AWS IoT Console



- Avnet **Getting Started Guide** details steps to connect **IoT Starter Kit** to **AWS IoT Console**
- The **AWS IoT Console** facilitates creating and managing **Things, Certificates, Policies** and **Rules** via an easy to use Graphical User Interface
- A Command Line Interface (**CLI**) is also provided (This for experienced users. Less useful now since the AWS IoT Console has become available...)

<https://aws.amazon.com/iot>

The screenshot displays the AWS IoT console interface. The main area shows a 'Resources' section with a 'Close create panel' button and four icons for 'Create a thing', 'Create a rule', 'Create a certificate', and 'Create a policy'. The 'Create a thing' form is active, showing a 'Name' field with the value 'sk\_0001' and an 'Add attribute' button. The 'Attributes' section below explains that attributes are key-value pairs used to describe the device. On the right, a 'Detail' panel for the device 'sk\_0001' is open, displaying its REST API endpoint, MQTT topic, last update time (1 hour ago), and shadow status ('In sync'). A JSON snippet is visible in the detail panel, showing the shadow state with 'desired' status 'OFF' and 'reported' status 'ON' for a 'bulb' attribute.





# What Can You Do With AWS Shadow?



**Name** starterkit\_0040

**REST API endpoint** https://A1DTCSS7KAHTDF.iot.us-east-1.amazonaws.com/things/starterkit\_0040/shadow

**MQTT Topic** '\$aws/things/starterkit\_0040/shadow/update'

**Last update** 2 Minutes Ago  


**Attributes** There are no attributes

**Linked certificates** [Show all](#)

**State Detail** In sync

**State Version** 13

**State**

```
1 - {
2   "desired": {
3     "status": "ON"
4   },
5   "reported": {
6     "status": "ON" 
7   }
8 }
```

**Ex.1:** User P/B is pressed and it's State is Published to AWS IoT


- Thing Shadow reported status is monitored in AWS IoT Console


**Ex.2:** State of User1 LED is manually updated in the Thing Shadow, - desired State then Published to AWS IoT

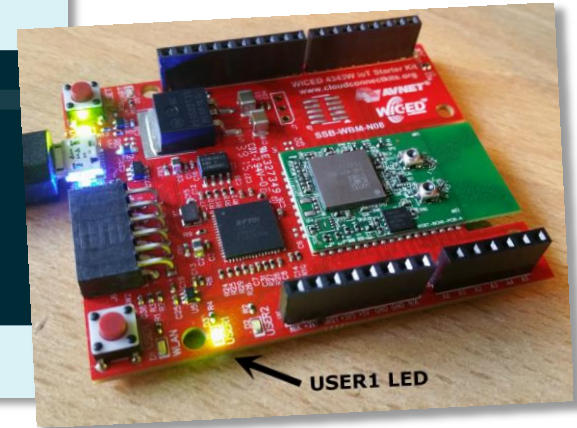
- IoT Starter Kit Subscription to this topic results in **User1 LED** getting updated

[Learn More](#) [Detail](#) [Update State](#) [Edit](#)

**State Detail**

```
1 - {
2   "desired": {
3     "status": "ON" 
4   },
5   "reported": {
6     "status": "OFF"
7   }
8 }
```

[Cancel](#) [Update State](#) 



# Using Rules with AWS Shadow



#9

**Ex.3:** Create a **rule** to trigger AWS SNS push notification to **email**



**Ex.4:** Create a **rule** to trigger AWS SNS push notification to **SMS text message**

Create a thing   **Create a rule**   Create a certificate   Create a policy

## Create a rule

Create a rule to evaluate inbound messages published into AWS IoT. Your rule can trigger an action, such as sending a message to the topic of another device, or to a cloud endpoint such as a DynamoDB instance.

Name your rule and add an optional description.

**Name**

**Description**

FILE MESSAGE McAfee E-mail Scan

Fri 10/30/2015 11:01 AM

AWS-IOT <no-reply@sns.amazonaws.com>

**AWS Notification Message**

To Fenn, Peter

```
{ "state": { "desired": { "status": "ON" }, "reported": { "status": "ON" } } }
```

30304 Mobile 13:45

1:44 PM

AWS-IOT> {"version": 237, "timestamp": 1446237887, "state": {"status": "OFF"}, "metadata": {"status": {"timestamp": 1446237887}}}

+ |Type a message...



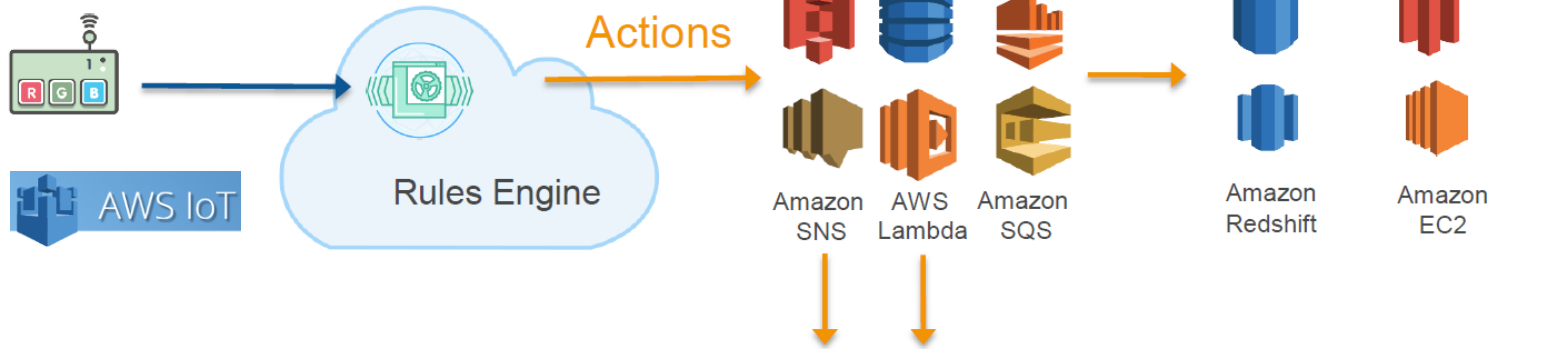
# Using Rules with AWS Shadow

#9

Rules Engine connects AWS IoT to External Endpoints and AWS Services.

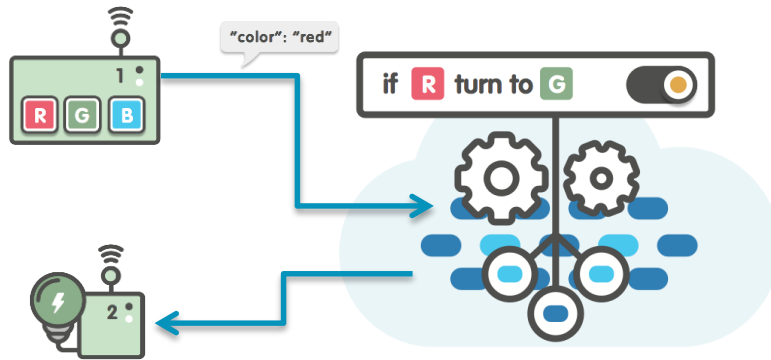
## 1. AWS Services (Direct Integration)

## 2. Rest of AWS (via Amazon Kinesis, AWS Lambda, Amazon S3, and more)



## 3. External Endpoints (via Lambda and SNS)

# AWS IoT Rules Engine Basics



```
SELECT *  
FROM 'things/thing-2/color'  
WHERE color = 'red'
```

## Simple & Familiar Syntax

- SQL Statement to define **topic filter**
- Optional **WHERE** clause
- Advanced JSON support

## Functions improve signal : noise

- String manipulation (regex support)
- Mathematical operations
- Context based helper functions
- Crypto support
- UUID, Timestamp, rand, etc.

# Topic Filter using Rules (SQL Syntax)



SQL (**S**tructured **Q**uery **L**anguage) is a popular, special-purpose programming language for managing data in relational databases and data streams

SELECT DATA FROM TOPIC WHERE FILTER

```
SELECT isbn, title, price,  
FROM book  
WHERE price < 20.00  
ORDER BY title;
```

*No filter of parameter or topic*

```
SELECT *  
FROM '#'
```

*Filter on parameters, topics and state*

```
SELECT state.reported.bulb  
FROM '$aws/things/#/shadow/update'  
WHERE state.reported.bulb = 'ON'  
OR state.reported.bulb = 'OFF'
```



# Using the MQTT.fx Tool

*(Optional PC-based MQTT Client for testing)*



29



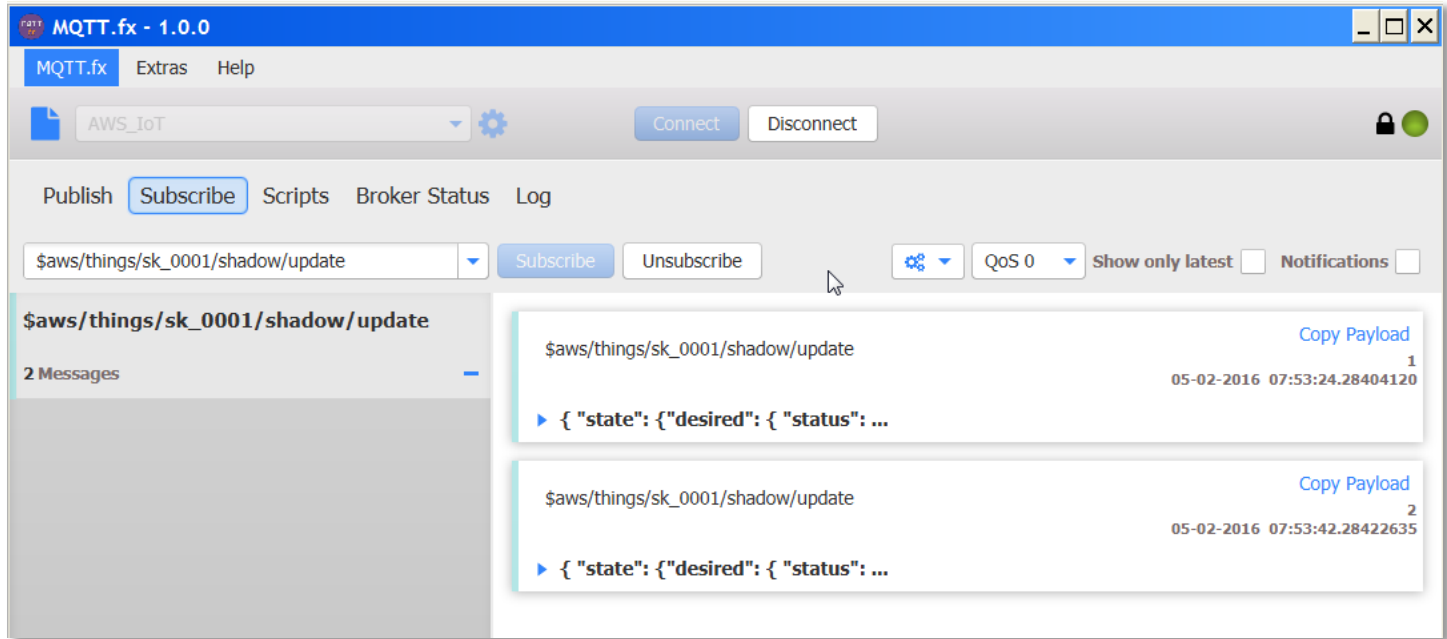
29 November 2016



# Using MQTT.fx to Debug AWS IoT



- MQTT.fx is an MQTT client written in JavaFX, based on Eclipse Paho



# Using MQTT.fx to Debug AWS IoT



**Edit Connection Profiles**

AWS\_IoT  
M2M Eclipse

### Connection Profile

Profile Name:

Broker Address:   
Broker Port:   
Client ID:

**General** | User Credentials | SSL/TLS | Proxy | Last Will and Testament

Connection Timeout:   
Keep Alive Interval:   
Clean Session:   
MQTT Version:  Use Default

# Using MQTT.fx to Debug AWS IoT



**Edit Connection Profiles**

AWS\_IoT  
M2M Eclipse

### Connection Profile

Profile Name:

Broker Address:

Broker Port:

Client ID:

General | User Credentials | **SSL/TLS** | Proxy | Last Will and Testament

Enable SSL/TLS

Protocol:

CA signed server certificate  
 CA certificate file  
 CA certificate keystore  
 Self signed certificates

CA File:  ...

Client Certificate File:  ...

Client Key File:  ...

Client Key Password:

PEM Formatted

Self signed certificates in keystores

# Using MQTT.fx to Debug AWS IoT



## Notes:

- 1) Download MQTT.fx from: <http://mqttx.fx4ee.org>
- 2) After installing, configure MQTT.fx by clicking the gear icon at top of the page
- 3) Broker Address: A CLI command is used to retrieve your AWS account-specific AWS IoT endpoint: **aws iot describe-endpoint**
- 4) Broker Port: Egress to port **8883** must be allowed on your network!
- 5) Choose the SSL/TLS tab to view the SSL/TLS settings page.
  - Check the **Select SSL/TLS** box
  - Select **Self-Signed Certificates**
  - Enter the paths to your private key, certificate and root CA certificate
  - Check the **Select PEM formatted** box
- 6) AWS documentation on MQTT.fx is available at:  
<http://docs.aws.amazon.com/iot/latest/developerguide/verify-pub-sub.html>



# Subscribing to an MQTT Topic



- Click on the Subscribe tab to view the Subscribe page...
- Enter the topic in the text box, eg. **\$aws/things/sk\_0001/shadow/update**
- From the drop-down list, select **QoS 0**, then click on **Subscribe**
- Click on a message to view it's full JSON payload
- If subscribing to multiple topics, their message boxes can be color-coded

The screenshot shows the MQTT.fx 1.0.0 web interface. The browser title is "MQTT.fx - 1.0.0". The interface includes a menu bar with "MQTT.fx", "Extras", and "Help". Below the menu is a toolbar with a file icon, a dropdown menu showing "AWS\_IoT", a settings gear icon, and "Connect" and "Disconnect" buttons. A status bar shows a lock icon and a green circle. The main content area has a "Publish" section with a "Subscribe" button (highlighted), "Scripts", "Broker Status", and "Log" links. Below this is a text input field containing "\$aws/things/sk\_0001/shadow/update", a "Subscribe" button, an "Unsubscribe" button, a settings gear icon, a "QoS 0" dropdown menu, and checkboxes for "Show only latest" and "Notifications". The message list on the left shows "2 Messages" for the topic "\$aws/things/sk\_0001/shadow/update". Two messages are displayed in the main area, each with a "Copy Payload" link and a truncated JSON payload starting with {"state": {"desired": {"status": ...}}.

# Using MQTT.fx to Debug AWS IoT



```
2016-02-05 08:23:44,688 INFO --- MqttFX ClientModel : MqttClient with ID MQTT_FX_Client assigned.
2016-02-05 08:23:56,731 INFO --- SubscribeController : onSubscribe
2016-02-05 08:23:57,171 INFO --- MqttFX ClientModel : getReceivedMessages().size() = 0
(getRefreshBufferCounter() + refreshBufferSize) = 1
2016-02-05 08:23:57,172 INFO --- MqttFX ClientModel : attempt to add PublishTopic
2016-02-05 08:23:57,174 INFO --- MqttFX ClientModel : sucessfully subscribed to topic
$aws/things/sk_0001/shadow/update (QoS 0)
2016-02-05 08:24:01,387 INFO --- MqttFX ClientModel : messageArrived for: $aws/things/sk_0001/shadow/update
2016-02-05 08:24:01,393 INFO --- MqttFX ClientModel : messageArrived: 12
2016-02-05 08:24:02,607 INFO --- MqttFX ClientModel : messageArrived for: $aws/things/sk_0001/shadow/update
2016-02-05 08:24:02,612 INFO --- MqttFX ClientModel : messageArrived: 13
```



# Reference Docs and Additional Info



36



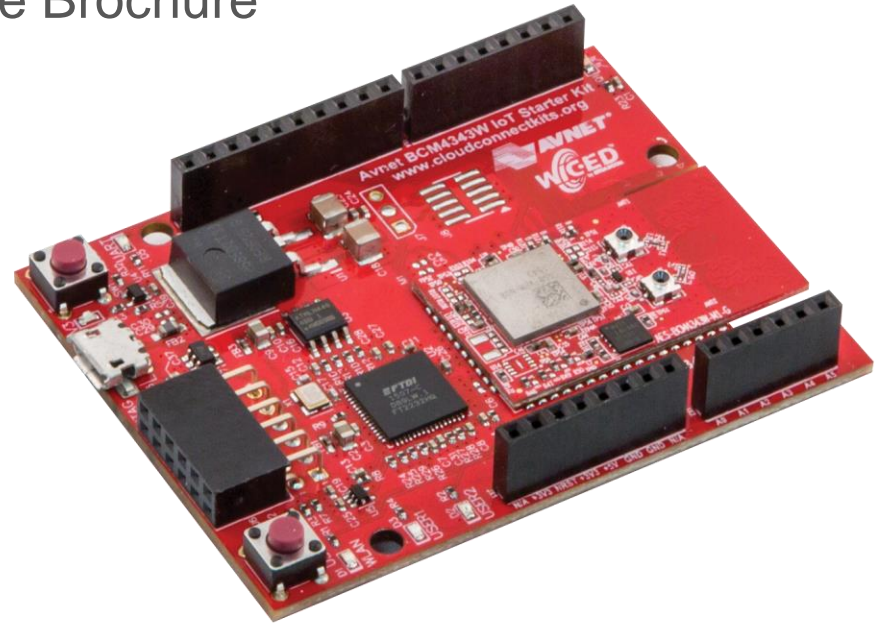
29 November 2016



# Reference Docs and Additional Info

Avnet documents located at: [www.cloudconnectkits.org](http://www.cloudconnectkits.org)

- Avnet BCM4343W IoT Starter Kit Brochure
- Avnet BCM4343W SoC Module Brochure
- Quick Start Card
- Getting Started Guide
- Hardware User Guide
- Schematics
- Bill Of Materials
- PCB Layout (Gerber files)
- Mechanical Drawing



# Where to go for Support?



## Avnet CloudConnectKits User Forum

<http://cloudconnectkits.org/forum>



## Avnet Documentation and GitHub Websites

<http://cloudconnectkits.org/product/avnet-bcm4343w-iot-starter-kit>

<https://github.com/CloudConnectKits>



## Cypress WICED WiFi Forum

<https://community.cypress.com/community/wiced-wifi/wiced-wifi-forums>



## AWS Getting Started Website

<http://aws.amazon.com/iot/getting-started>



## AWS IoT and Other Discussion Forums

<https://forums.aws.amazon.com/forum.jspa?forumID=210>

<https://forums.aws.amazon.com>



---

# Other Important AWS Links

---

## AWS IoT Documentation

<https://aws.amazon.com/documentation/iot>

## AWS IoT Developer Guide

<https://docs.aws.amazon.com/iot/latest/developerguide>

## AWS CLI Tool Installation

<http://docs.aws.amazon.com/cli/latest/userguide/installing.html>



# Q & A

- **Price and Availability ?**
  - IoT Starter Kits are in stock and orderable now (\$49.95)
  - BCM4343W SoC Modules shipping since 2Q16  
(module pricing based on quantity, contact your local Avnet resource for more info)
- **How to order the BCM4343W IoT Starter Kit and BCM4343W SoC Modules?**
  - Yes, part#: **AES-BCM4343W-M1-G**
- **FCC, CE, etc Certifications ?**
  - BCM4343W SoC Module FCC and CE certification is currently in progress
- **Additional Reference Designs and Software ?**
  - Dozens of examples are available in WICED SDK,
  - BLE + WiFi + AWS and BLE + WiFi + Bluemix combo examples available (BT Smartbridge, BLE2Bluemix)
- **Support for Apple HomeKit ?**
  - Module has an unpopulated site for MFI Auth.device, however this will not be formally supported
- **Hands-on Customer trainings ?**
  - Online webinars as well as live Avnet workshops are available...



