# Chapter 6: Cloud Connectivity using MQTT + Amazon Web Services

## Objective

At this end of Chapter 6 you will understand:

* How Message Queue Telemetry Transport (MQTT) works:
  + How MQTT fits into the TCP/IP Networking Stack
  + What a Topic is
  + What a Message Broker is
  + What a Publisher is
  + What a Subscriber is
  + Understand Quality of Service (QOS)
  + MQTT Security
  + What are the Ports used by MQTT
  + What is Keep Alive
* How to use the JavaScript Object Notation (JSON) language.
* How the Amazon AWS MQTT Cloud works
  + Know how to provision “things” in the Amazon Cloud by creating things, policies and certificates
  + Understand what a “shadow device” is
  + Know how to use an AWS MQTT Client to subscribe and publish to topics
  + Understand the scope of systems that can be implemented in the cloud (SNS, Database etc.)
  + Understand the role of public and private key cryptography in the AWS cloud
* Understand in **DETAIL** how to write WICED firmware to interact with the AWS IOT Cloud

## Time: 4 Hours

## Fundamentals

### Networking Stack the Application Layer

### (Application Protocol) MQQT

#### Topic

#### Publisher

#### Subscriber

### (Data Format) JSON

### (Cloud) AWS (MQTT)

#### <http://www.slideshare.net/PeterREgli/mq-telemetry-transport>

#### Introduction + Setup

#### Shadow

Legal shadow topic format of topic

<http://docs.aws.amazon.com/iot/latest/developerguide/thing-shadow-mqtt.html>

(format of json that you need to “publish” <http://docs.aws.amazon.com/iot/latest/developerguide/thing-shadow-document-syntax.html>

## Exercise(s)

* Provision a new thing in the AWS IOT Cloud, establish its policy and credentials
* Build the publisher app (modify dct and MQTT functions)
* Subscribe to the topic using the AWS MQTT client (and test the publishing)
* Explain in detail the firmware flow for publisher app
  + How do the MQTT library functions (e.g. wiced\_mqtt\_publish ) get into your project?
  + Why did the firmware author create a function called “wait\_for\_result”?
  + What mechanism does the “wait\_for\_result” function use to cause the “wait”?
  + What are the 7 WICED MQTT events? What file are they defined in?
  + Do you have to name the client certificate client.cer? How would you change the name?
  + What is the naming convention used to differentiate WICED MQTT library functions versus wrappers around those functions in the publisher app?
  + What is the sequence of function calls to open a connection?
  + What prevents a hung connection from deadlocking the publisher app?
  + What function is called when the button is pressed?
  + How does the button callback unlock the main thread?
  + What is the name of the flag that prevents the firmware from sending multiple button presses before the publish is finished?
  + Are all message sent to the AWS IOT MQTT Message broker required to be in JSON format?
  + Add debugging prints to the function mqtt\_connetion\_event\_cb
  + To open a connection, the flow looks like this:
    - mqtt\_conn\_open calls wiced\_mqtt\_connect with a call back of mqtt\_connection\_event\_cb
    - mqtt\_conn\_open calls wait\_for\_result which gets the msg\_semaphore (which halts everything)
    - When Amazon responds, the mqtt\_connecton\_event\_cb occurs and wakes up the thread
    - The mqtt\_connection\_event\_cb updates the “expected\_event” global variable and set the msg\_semaphore then returns
    - Back in wait\_for\_event If the message WICED\_MQTT\_EVENT\_TYPE\_CONNECT\_REQ\_STATUS is received then it then it returns WICED\_SUCCESS else it returns WICED\_FAIL
    - What is the equivalent flow to publish?
* Implement the subscriber app (don’t forget to update the DCT parameters and the MQTT parameters). Test using the MQTT client app
* Implement the subscriber and publisher in two different kits and test
* Change the subscriber APP to print on the screen the messages that are received
* Modify the publisher app to publish updates to the shadow
* Implement the shadow application
* “Bake-In” the AWS security credential into the shadow app
* “Bake-In” the WiFi credentials into the shadow app

## Related Example “Apps”

|  |  |
| --- | --- |
| **App Name** | **Function** |
| secure\_mqtt |  |
|  |  |

## References

[www.cloudconnectkits.org](http://www.cloudconnectkits.org)

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

quick start

<http://cloudconnectkits.org/system/files/kit-WICED4343W-IoTStarterKit-QSC-v1b.pdf>

getting started guide

<http://cloudconnectkits.org/system/files/GSG-BCM4343W%20IoT%20Starter%20Kit%20-%20Getting%20Started%20%28v1.1%29.pdf>

user guide part 1

<http://cloudconnectkits.org/system/files/Tutorial%20Part1%20-%20Tool%20Install%2C%20USB%20drivers%20and%20AWS%20Shadow%20%28v1.1%29.pdf>

user guide part 2

<http://cloudconnectkits.org/system/files/Tutorial%20Part2-App%20Development%20using%20WICED%20SDK-v1.1.pdf>

hardware user guide

<http://cloudconnectkits.org/system/files/UG%20-%20BCM4343W%20IoT%20StarterKit%20-%20Hardware%20User%20Guide%20rev.1.0.pdf>

schematics

<http://cloudconnectkits.org/system/files/BCM4343W_STARTER_REV_B_Schematics.PDF>

Reference Designs (these are jacked)

|  |
| --- |
| [AV01: Connect to IBM Watson IoT using Avnet BCM4343W IoT Starter Kit](http://cloudconnectkits.org/system/files/redirect-ibm-referencedesigns_1.html)  [Download](http://cloudconnectkits.org/system/files/redirect-ibm-referencedesigns_1.html)  [AV02: Sending Alerts from BCM4343W IoT Starter Kit using IBM Bluemix](http://cloudconnectkits.org/system/files/AV02%20-%20Sending%20Alerts%20for%20BCM4343W%20IoT%20Starter%20Kit%20via%20Bluemix%20IoT%20Real-Time%20Insights%20%28v1.0%29.pdf)  [Download](http://cloudconnectkits.org/system/files/AV02%20-%20Sending%20Alerts%20for%20BCM4343W%20IoT%20Starter%20Kit%20via%20Bluemix%20IoT%20Real-Time%20Insights%20%28v1.0%29.pdf)  [AV03: WICED Sense BLE data publish to IBM Bluemix via BCM94343W\_AVN](http://cloudconnectkits.org/system/files/redirect-ibm-referencedesigns_2.html)  [Download](http://cloudconnectkits.org/system/files/redirect-ibm-referencedesigns_2.html)  [AV04: Alexa Voice-Controlled Smart Home demo using BCM4343W IoT Starter Kit and IBM Bluemix](http://cloudconnectkits.org/system/files/AV04%20-%20Alexa%20Voice-Controlled%20Smart%20Home%20Demo%20using%20BCM4343W%20IoT%20Starter%20Kit%20and%20IBM%20Bluemix%20%28v1.0%29.zip)  [Download](http://cloudconnectkits.org/system/files/AV04%20-%20Alexa%20Voice-Controlled%20Smart%20Home%20Demo%20using%20BCM4343W%20IoT%20Starter%20Kit%20and%20IBM%20Bluemix%20%28v1.0%29.zip)  [AV05 - Sending Alerts from BCM4343W IoT Starter Kit Alexa Voice-Controlled Smart Home Demo (v1.1)](http://cloudconnectkits.org/system/files/AV05%20-%20Sending%20Alerts%20from%20BCM4343W%20IoT%20Starter%20Kit%20Alexa%20Voice-Controlled%20Smart%20Home%20Demo%20%28v1.1%29.pdf)  [Download](http://cloudconnectkits.org/system/files/AV05%20-%20Sending%20Alerts%20from%20BCM4343W%20IoT%20Starter%20Kit%20Alexa%20Voice-Controlled%20Smart%20Home%20Demo%20%28v1.1%29.pdf)  [AV06: WICED Sense2 BLE Tag to IBM Bluemix demo](http://cloudconnectkits.org/system/files/redirect-ibm-referencedesigns_0.html)  [Download](http://cloudconnectkits.org/system/files/redirect-ibm-referencedesigns_0.html)  [AV07 - Creating Dashboards Using Bluemix Services (v1.1)](http://cloudconnectkits.org/system/files/AV07%20-%20Creating%20Dashboards%20Using%20Bluemix%20Services%20%28v1.1%29.pdf)  [Download](http://cloudconnectkits.org/system/files/AV07%20-%20Creating%20Dashboards%20Using%20Bluemix%20Services%20%28v1.1%29.pdf)  [AV08: Sensor-to-Cloud Using TI SensorTag and Watson IoT Quickstart](http://cloudconnectkits.org/system/files/AV08%20-%20SensorTag%20Cloud%20Files%20%28v1.0%29.zip)  [Download](http://cloudconnectkits.org/system/files/AV08%20-%20SensorTag%20Cloud%20Files%20%28v1.0%29.zip)  [AV09: Zentri to IBM Bluemix Notifications and Visualizations](http://cloudconnectkits.org/system/files/AV09%20-%20Zentri%20to%20IBM%20Bluemix%20Notifications%20and%20Visualizations%20%28v1.0%29.pdf)  [Download](http://cloudconnectkits.org/system/files/AV09%20-%20Zentri%20to%20IBM%20Bluemix%20Notifications%20and%20Visualizations%20%28v1.0%29.pdf)  [AV10: NXP 3D Shield LED Matrix demo](http://cloudconnectkits.org/system/files/redirect-ibm-referencedesigns.html)  [Download](http://cloudconnectkits.org/system/files/redirect-ibm-referencedesigns.html)  [AV11 - Webpage Controlled LED Matrix Display (ZentriOS, Wi-Fi v1.0)](http://cloudconnectkits.org/system/files/AV11%20-%20Webpage%20Controlled%20LED%20Matrix%20Display%20%28ZentriOS%20Wi-Fi%20v1.0%29.pdf)  [Download](http://cloudconnectkits.org/system/files/AV11%20-%20Webpage%20Controlled%20LED%20Matrix%20Display%20%28ZentriOS%20Wi-Fi%20v1.0%29.pdf)  [View Reference Designs for the BCM4343W IoT Starter Kit on IBM DeveloperWorks Recipe site](http://cloudconnectkits.org/system/files/redirect-ibm-referencedesigns_3.html)  [Download](http://cloudconnectkits.org/system/files/redirect-ibm-referencedesigns_3.html)  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  https://github.com/CloudConnectKits  AWS IoT and Other Discussion Forums  https://forums.aws.amazon.com/forum.jspa?forumID=210 https://forums.aws.amazon.com |

## Known Errata + Enhancements + Comments