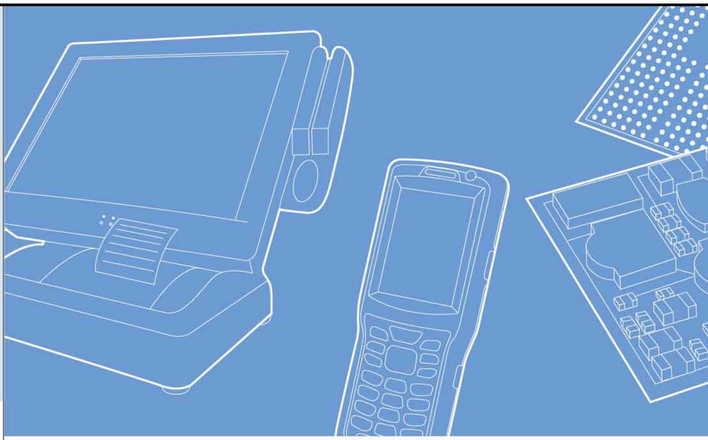
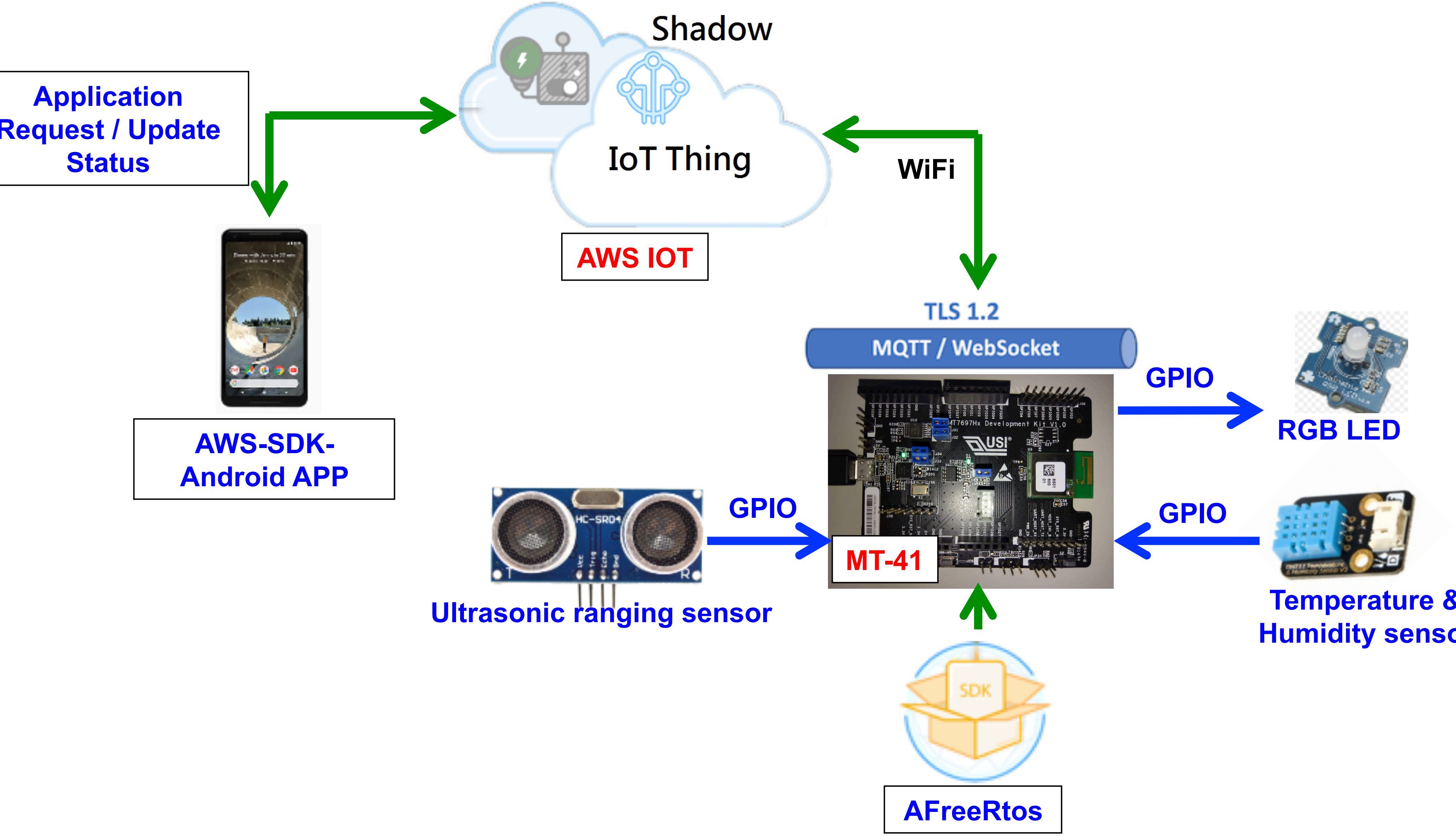


MT-41 Demo Guide

Kevin Chang
RD/SW
WP1/WCS/ICS UG

MT-41 Demo Architecture



Application Request / Update Status

Shadow

IoT Thing

AWS IOT

WiFi

AWS-SDK-Android APP

TLS 1.2

MQTT / WebSocket

GPIO

RGB LED

GPIO

MT-41

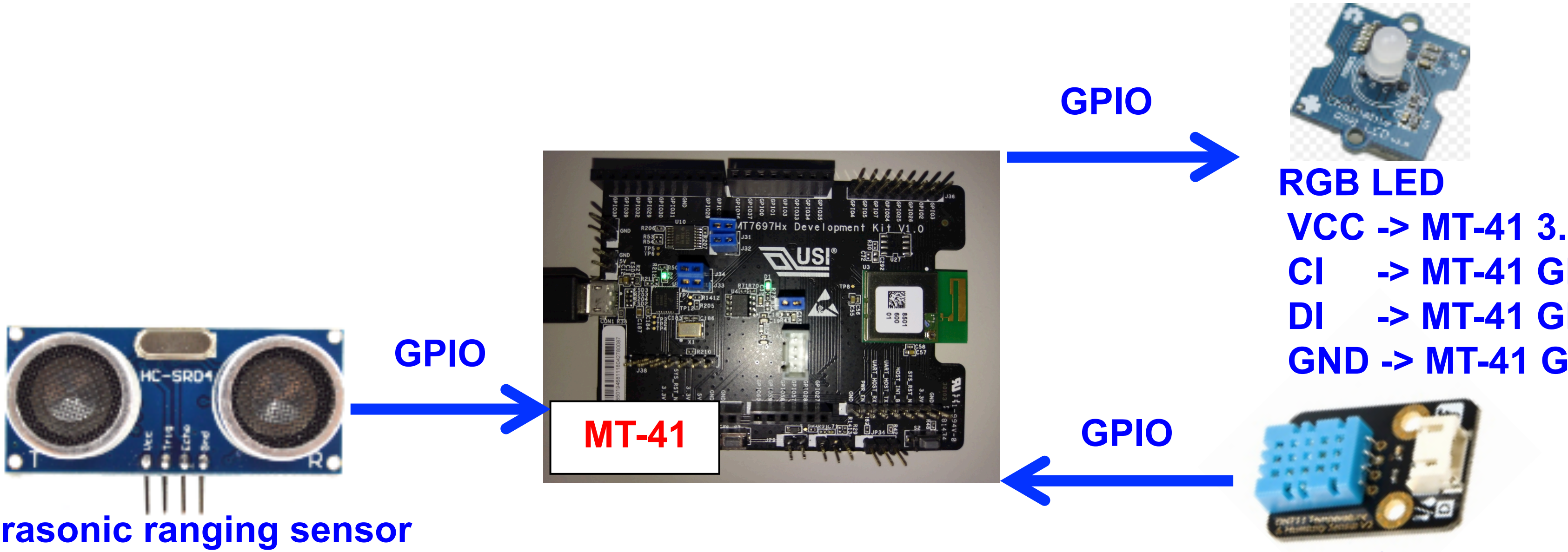
GPIO

Ultrasonic ranging sensor

Temperature &
Humidity senso

AFreeRtos

MT-41 Demo Pin Connection



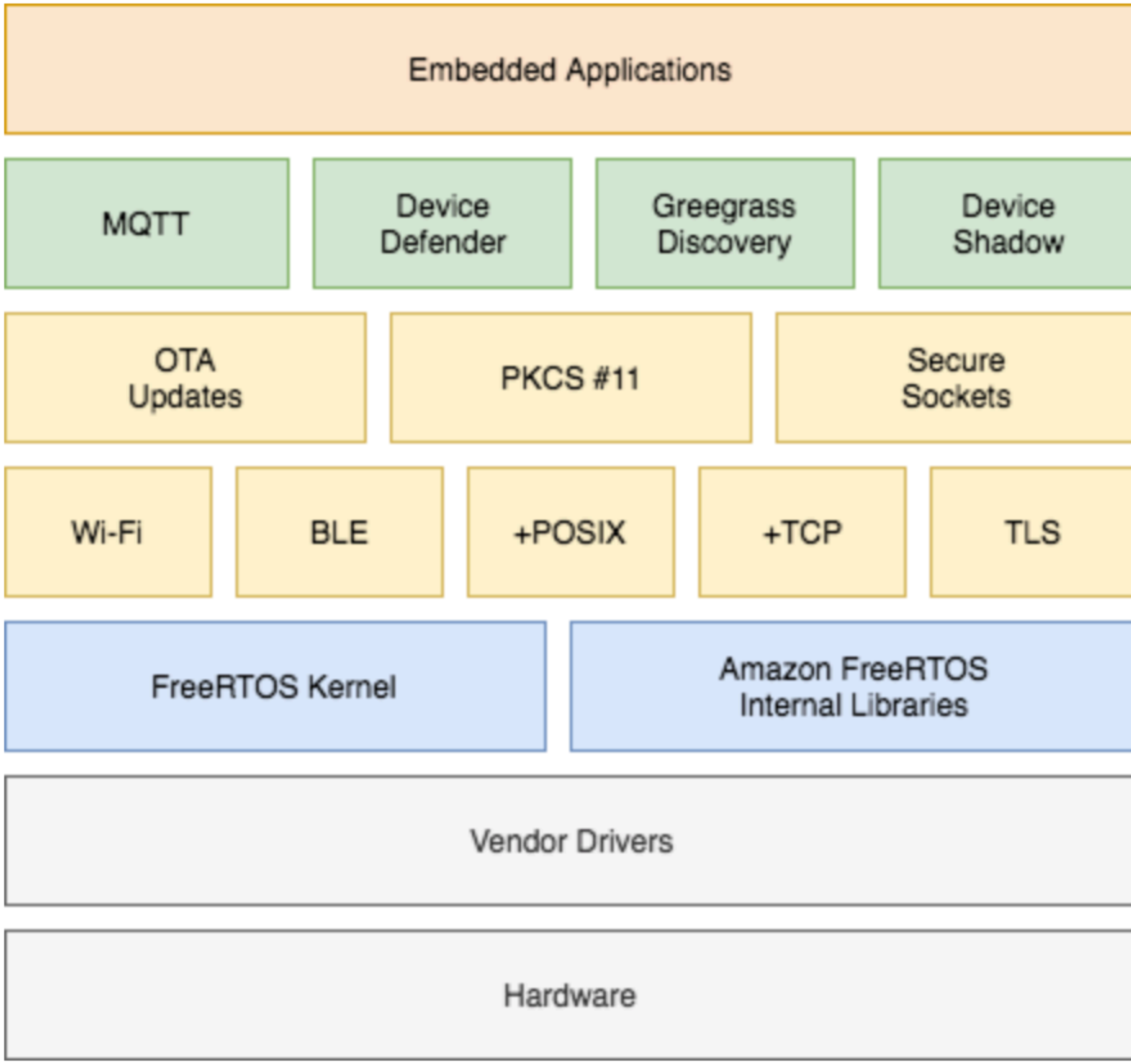
RGB LED
VCC -> MT-41 3.3V
CI -> MT-41 GPIO25
DI -> MT-41 GPIO26
GND -> MT-41 GND

Ultrasonic ranging sensor
VCC -> MT-41 5V
Trig -> MT-41 GPIO5
Echo -> MT-41 GPIO4
GND -> MT-41 GND

Temperature & Humidity sensor
VCC -> MT-41 3.3V
DATA -> MT-41 GPIO24
GND -> MT-41 GND

Amazon FreeRTOS Architecture

- Amazon FreeRTOS includes libraries that enable to:
  - Securely connect devices to the AWS IoT cloud using MQTT and device shadows
  - Discover and connect to AWS IoT Greengrass cores
  - Manage Wi-Fi connections
  - Listen for and process Amazon FreeRTOS Over-the-Air Updates



Embedded Applications

MQTT | Device Defender | Greegrass Discovery | Device Shadow

OTA Updates | PKCS #11 | Secure Sockets

Wi-Fi | BLE | +POSIX | +TCP | TLS

FreeRTOS Kernel | Amazon FreeRTOS Internal Libraries

Vendor Drivers

Hardware

Getting Start with Amazon FreeRTOS

- Setting Up Your AWS Account and Permissions
  - To attach the AmazonFreeRTOSFullAccess policy to your IAM user
  - To attach the AWSIoTFullAccess policy to your IAM user
  - https://docs.aws.amazon.com/en_us/freertos/latest/userguide/freertos-account-and-permissions.html

- Registering Your MCU Board with AWS IoT
  - To register your board with AWS IoT, you need the following:
    - An AWS IoT policy
    - An AWS IoT thing
    - A private key and X.509 certificate
  - https://docs.aws.amazon.com/en_us/freertos/latest/userguide/get-started-freertos-thing.html

Getting Start with Amazon FreeRTOS

- Downloading Amazon FreeRTOS
  - from the Amazon FreeRTOS console
    - Go to the Amazon FreeRTOS console
    - Under Predefined configurations, find Connect to AWS IoT- Platform, and then choose Download
  - or from GitHub

    https://github.com/aws/amazon-freertos

Getting Start with Amazon FreeRTOS

- Configuring the Amazon FreeRTOS Demos
  - To configure your AWS IoT endpoint :
  - 1. Browse to the AWS IoT console. In the navigation pane, choose Settings.
    - Your AWS IoT endpoint is displayed in Endpoint. It should look like <1234567890123>-ats.iot.<us-east-1>.amazonaws.com. Make a note of this endpoint
  - 2. In the navigation pane, choose Manage, and then choose Things.
    - Your device should have an AWS IoT thing name. Make a note of this name
  - 3. Open <BASE_FOLDER>\demos\common\include\aws_clientcredential.h. Specify values for the following constants:
    - static const char clientcredentialMQTT_BROKER_ENDPOINT[] = "Your AWS IoT endpoint";
    - #define clientcredentialIOT_THING_NAME "The AWS IoT thing name of your board"

Getting Start with Amazon FreeRTOS

Configuring the Amazon FreeRTOS Demos

- To configure your Wi-Fi :
- 1.Open <BASE_FOLDER>\demos\common\include\aws_clientcredential.h. Specify values for the following #define constants:
  - #define clientcredentialWIFI_SSID ”Your Wi-Fi network SSID"
  - #define clientcredentialWIFI_PASSWORD ”Your Wi-Fi network Password"
  - #define clientcredentialWIFI_SECURITY Your Wi-Fi network Security Type

- Valid security types are:
  - eWiFiSecurityOpen (Open, no security)
  - eWiFiSecurityWEP (WEP security)
  - eWiFiSecurityWPA (WPA security)
  - eWiFiSecurityWPA2 (WPA2 security)

Getting Start with Amazon FreeRTOS

Configuring the Amazon FreeRTOS Demos

- To format your AWS IoT credentials :
- You need the private key and certificate that downloaded from the AWS IoT console when registered the device
- 1. In a browser window, open <BASE_FOLDER>\tools\certificate_configuration\CertificateConfigurator.html
- 2. Under Certificate PEM file, choose the <ID>-certificate.pem.crt that downloaded from the AWS IoT console
- 3. Under Private Key PEM file, choose the <ID>-private.pem.key that downloadec from the AWS IoT console
- 4. Choose Generate and save aws_clientcredential_keys.h, and then save the file in <BASE_FOLDER>\demos\common\include. This overwrites the existing file in the directory

Getting Start with the MT-41 Dev. Kit

- Setting Up the Environment
  - Download and Install Keil MDK
    - Go to the Keil MDK Getting Started page, and choose Download MDK-Core

      http://www2.keil.com/mdk5/install/
  - Set Up a Serial Connection
    - Install the Arm Mbed Windows serial port driver

      https://os.mbed.com/docs/mbed-os/v5.12/tutorials/windows-serial-driver.html
    - After install the driver, a COM port appears in the Windows Device Manager
    - For debugging, you can open a session to the port with a terminal utility tool such as HyperTerminal or TeraTerm

Getting Start with the MT-41 Dev. Kit

- Setting Up Your AWS IoT Topic
  - aws_hello_world.c at the <BASE_FOLDER>/demos/common/mqtt/
  - Specify values for the following constants:
    - #define LEDTOPIC_NAME ((const uint8_t *) "Your LED TOPIC")
    - #define SensorTOPIC_NAME ((const uint8_t *) "Your Sensor TOPIC")

Getting Start with the MT-41 Dev. Kit

- Add Sensor Control in the Demo Project
  - Chainable P9813 LED:
  - p9813_led.c at the <BASE_FOLDER>/demos/mediatek/mt7697hx-dev-kit/common/application_code/mediatek_code/source/
  - p9813_led.h at the <BASE_FOLDER>/demos/mediatek/mt7697hx-dev-kit/common/application_code/mediatek_code/include/
  - Specify values for the following constants:

#define P9813_CLK	HAL_GPIO_25
#define P9813_DATA	HAL_GPIO_26
#define P9813_CLK_FUN_IDX	HAL_GPIO_25_GPIO25
#define P9813_DATA_FUN_IDX	HAL_GPIO_26_GPIO26

Getting Start with the MT-41 Dev. Kit

- Add Sensor Control in the Demo Project
  - DHT11 Temperature & Humidity Sensor:
  - dht.c at the <BASE_FOLDER>/demos/mediatek/mt7697hx-dev-kit/common/application_code/mediatek_code/source/
  - dht.h at the <BASE_FOLDER>/demos/mediatek/mt7697hx-dev-kit/common/application_code/mediatek_code/include/
  - Specify values for the following constants:
    - #define DHT_DATA HAL_GPIO_24
    - #define DHT_DATA_FUN_IDX HAL_GPIO_24_GPIO24

Getting Start with the MT-41 Dev. Kit

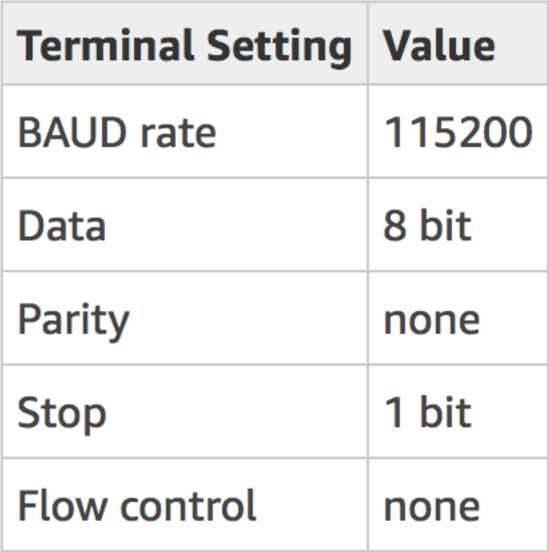
- Add Sensor Control in the Demo Project
  - HC-SR04 Ultrasonic Sensor:
  - hc_ranging.c at the <BASE_FOLDER>/demos/mediatek/mt7697hx-dev-kit/common/application_code/mediatek_code/source/
  - hc_ranging.h at the <BASE_FOLDER>/demos/mediatek/mt7697hx-dev-kit/common/application_code/mediatek_code/include/
  - Specify values for the following constants:
    - #define HCECHO HAL_GPIO_4
    - #define HCTRIG HAL_GPIO_5
    - #define HCECHO_FUN_IDX HAL_GPIO_4_GPIO4
    - #define HCTRIG_FUN_IDX HAL_GPIO_5_GPIO5

Getting Start with the MT-41 Dev. Kit

- Build the Demo Project with Keil MDK
  - Open the <BASE_FOLDER>/demos/mediatek/mt7697hx-dev-kit/uvision/aws_demo.uvprojx project file
  - After the project is built, the demo executable file at <BASE_FOLDER>/demos/mediatek/mt7697hx-dev-kit/uvision/out/Objects/aws_demo.axf
- Download the Demo Project with Keil MDK
  - Set the MT-41 Development Kit to PROGRAM mode
    - press and hold the PROG button. With the PROG button still pressed, press and release the RESET button, and then release the PROG button
  - From the menu, choose Flash, and then choose Configure Flash Tools
  - In Options for Target 'aws_demo', choose the Debug tab. Select Use, set the debugger to CMSIS-DAP Debugger, and then choose OK
  - From the menu, choose Flash, and then choose Download

Monitor with Terminal Emulator

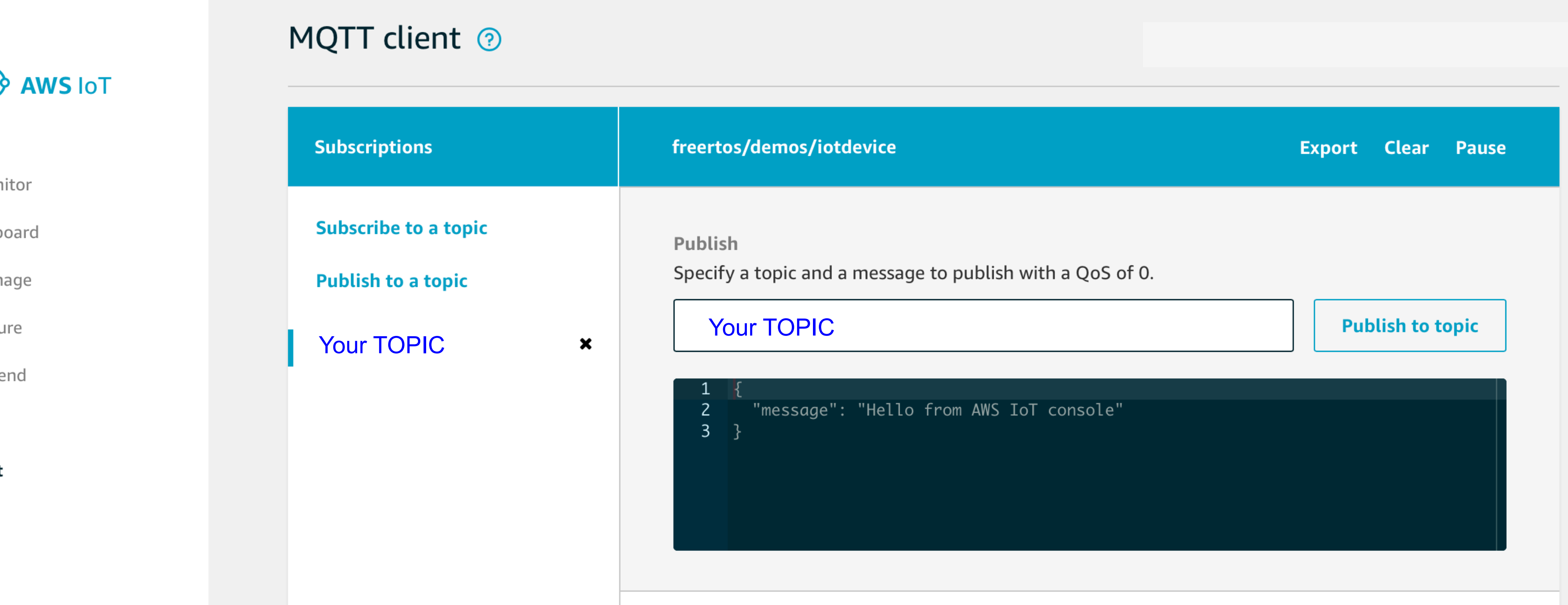
- A terminal emulator can help you diagnose problems or verify that your device code is running properly.
- Connect your board to the computer before establish a serial connection to your board with a terminal emulator.
- Use the following settings to configure your terminal emulator:



Terminal Setting	Value
BAUD rate	115200
Data	8 bit
Parity	none
Stop	1 bit
Flow control	none

Monitor with MQTT Client

- Use the MQTT client to monitor the messages that device sends to the AWS Cloud
- To subscribe to the MQTT topic with the AWS IoT MQTT client
  - Sign in to the AWS IoT console, choose Test to open the MQTT client
  - In Subscription topic, enter “Your TOPIC”, and then choose subscribe to topic



Publish

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

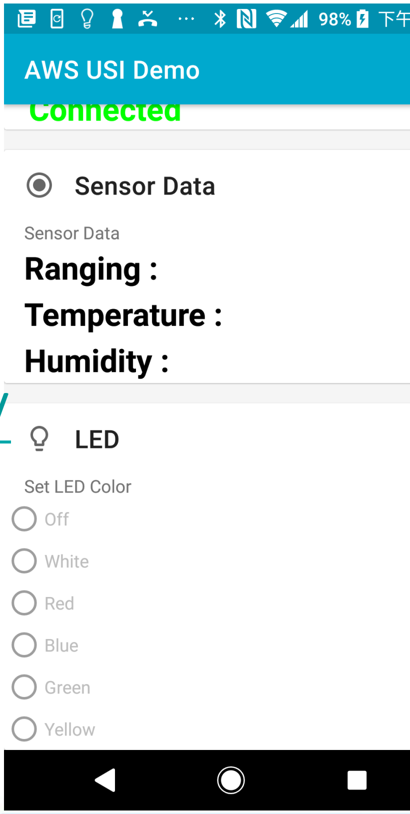
Your TOPIC

```
{
  "message": "Hello from AWS IoT console"
}
```

Getting Start with AWS Android SDK

Setting Up Your AWS IoT Endpoint

- WiFiDemoActivity.java at androidawsusidemo/app/src/main/java/android/usidemo/
- Specify values for the following constants:
  - private static final String CUSTOMER_SPECIFIC_ENDPOINT = "Your AWS IoT endpoint";
  - private static final String AWS_IOT_POLICY_NAME = "Your AWS IoT policy”;
  - private static final Regions MY_REGION = Regions.Your AWS IoT Region ;
  - private static final String PUBLISH_TOPIC =”Your LED TOPIC";
  - private static final String SUBSCRIBE_TOPIC =”Your Sensor TOPIC";
- Refer to aws-sdk-android-samples https://github.com/awslabs/aws-sdk-android-samples/tree/master/AndroidPubSub



Connected

◉ Sensor Data

Sensor Data

Ranging :

Temperature :

Humidity :

LED

Set LED Color

○ Off
○ White
○ Red
○ Blue
○ Green
○ Yellow

Thank You

www.usish.com

Realizing IDEAS Together