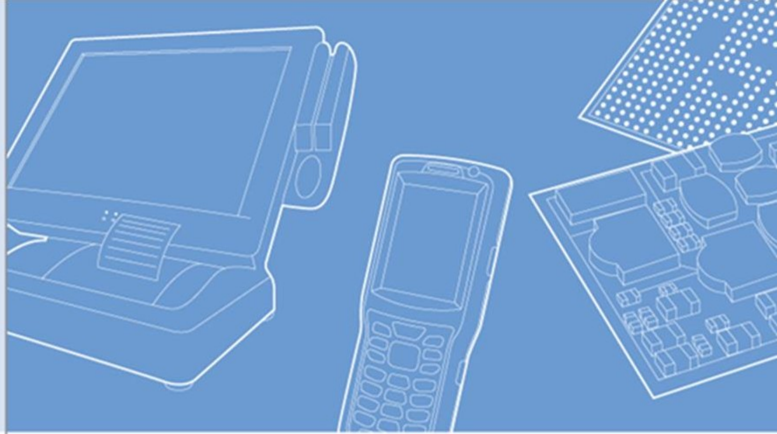


Azure Sphere Module User Guide V1.0

Daniel Tu
FAE
WCS/WP1
Jun. 3rd, 2020

Outline

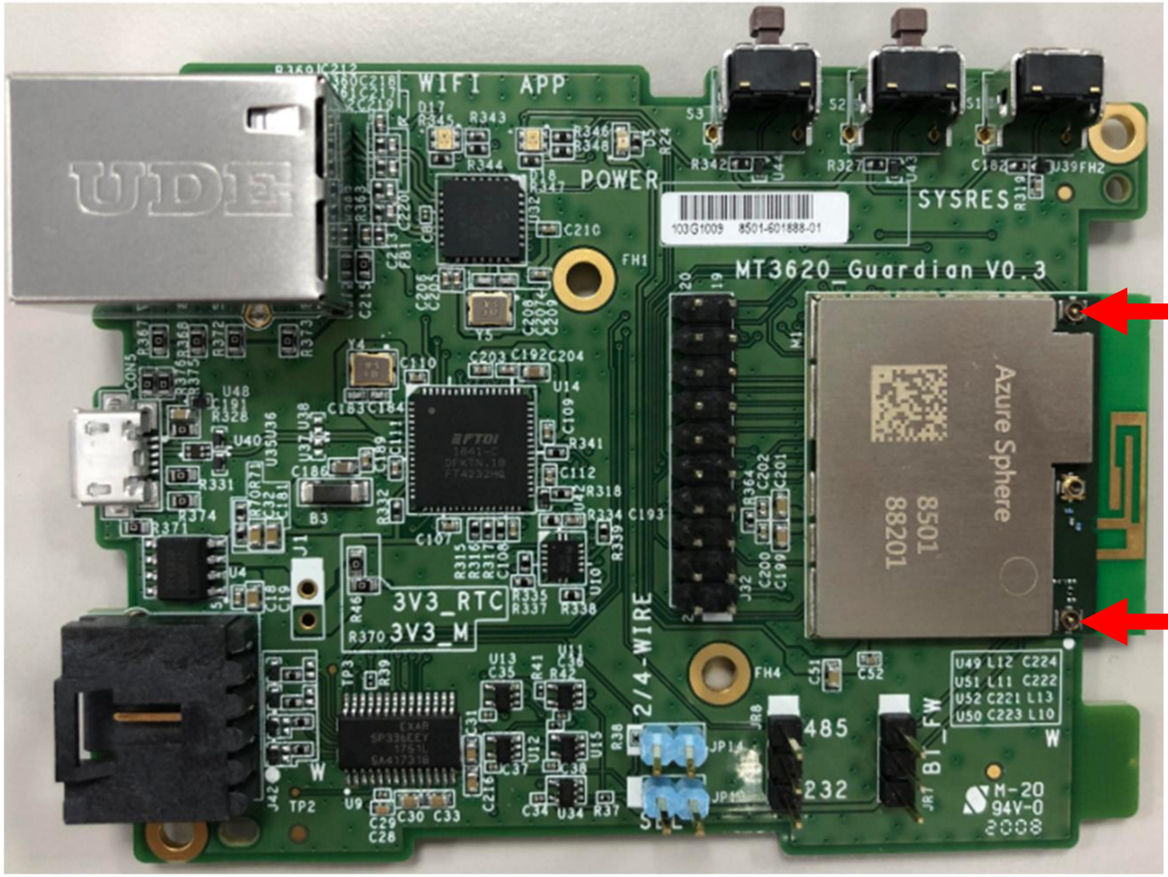
- MT3620 Setup
- Nordic nRF52 BLE Setup
- BLE-based Wi-Fi setup sample
- Private Ethernet sample
- Nordic BLE DFU
- Additional

  - FTDI Programming Configuration

  - FCC/IC Cert Statement

Azure Sphere Carrier Board

- MT3620_Guardian Rev:0.3

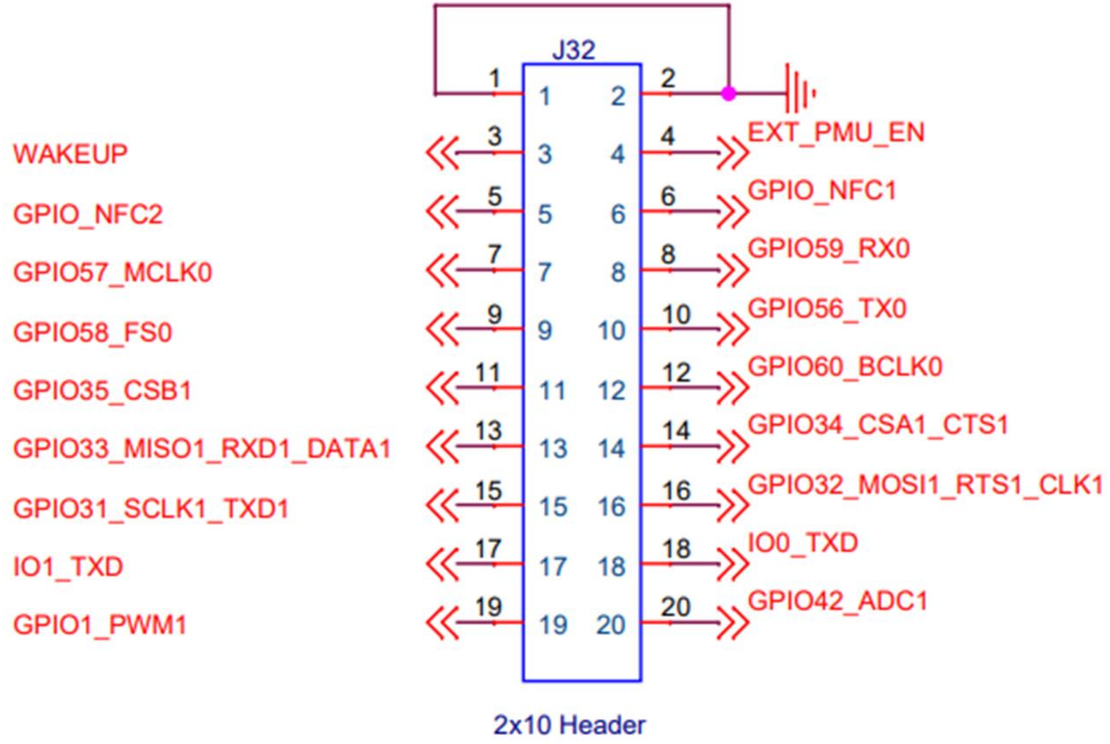


WiFi External Antenna

BLE External Antenna

MT3620 Interface

- MT3620_Guardian Rev:0.2

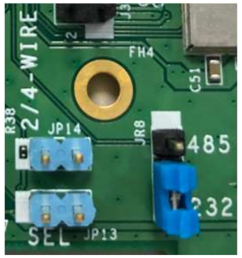

J32
1 2
WAKEUP 3 4 EXT_PMU_EN
GPIO_NFC2 5 6 GPIO_NFC1
GPIO57_MCLK0 7 8 GPIO59_RX0
GPIO58_FS0 9 10 GPIO56_TX0
GPIO35_CSB1 11 12 GPIO60_BCLK0
GPIO33_MISO1_RXD1_DATA1 13 14 GPIO34_CSA1_CTS1
GPIO31_SCLK1_TXD1 15 16 GPIO32_MOSI1_RTS1_CLK1
IO1_TXD 17 18 IO0_TXD
GPIO1_PWM1 19 20 GPIO42_ADC1
2x10 Header

LED	MT3620 GPIO
APP RED	GPIO45
APP GREEN	GPIO46
APP BLUE	GPIO47
WIFI RED	GPIO48
WIFI GREEN	GPIO14
WIFI BLUE	GPIO11

BUTTON	MT3620 GPIO
S2	GPIO12
S3	GPIO13

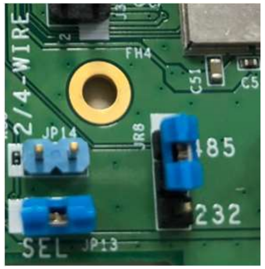
MT3620 Interface

- RS232 : Link Pin2 and Pin3 of JR8



- RS485 Full Duplex : Link Pin1 and Pin2 of JR8. (GPIO2 need pull high)

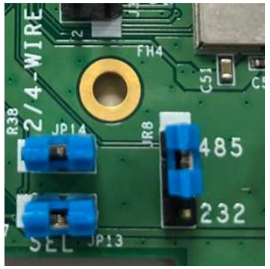
  Link JP13

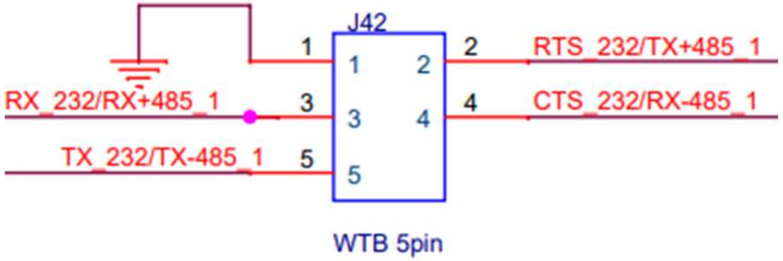


- RS485 Half Duplex : Link Pin1 and Pin2 of JR8. (GPIO2 pull High for TX, pull low for RX)

  Link JP13

  Link JP14





JR8	MT3620 GPIO
PIN 1	GPIO67_MOSI3_RTS3_CLK3
PIN 2	(SP336E RS232 RTS/ RS485 Enable)
PIN 3	GPIO2

MT3620 UART Function

MT3620 UART	Description
ISU0	For Ethernet(ENC28J60) using
ISU1	Header J32 pin 13,14,15,16
ISU2	FTDI(FT4232HQ) Port-A
ISU3	RS232/RS485 (SP336E)
ISU4	Nordic BLE nRF52 UART function

MT3620 Setup Steps

- Follow steps let Visual Studio can download and run debugger to Azure Sphere (MT3620)

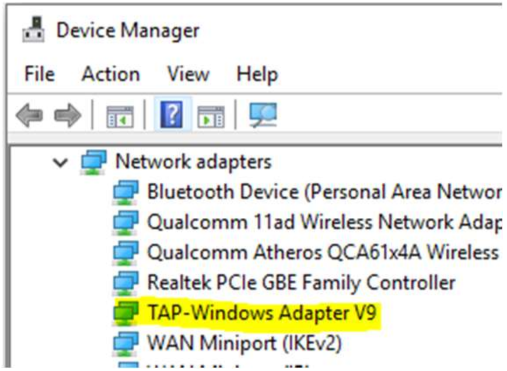
- Install Azure Sphere SDK and Tools
- Verify Azure Sphere SDK Installation
- Verify TAP-Windows Adapter Configuration
- Create tenant and claim your device
- Connect to Wi-Fi
- Prepare MT3620 for development and debugging
- Azure Sphere Application
- Build and Run High-level Application
- Set Up Hardware For Real-time App
- Build a real-time capable application

Install Azure Sphere SDK and Tools

- Install FTDI driver
  - https://www.ftdichip.com/Drivers/VCP.htm
- Install Visual Studio 2019 Enterprise, Professional, or Community version 16.04 or later; or Visual Studio 2017 version 15.9 or later.
  - https://visualstudio.microsoft.com/
- Install the Azure Sphere SDK Preview for Visual Studio
  - https://aka.ms/AzureSphereSDKDownload
  - Azure Sphere SDK only support on Windows 10, version 1607 Update or later
- Read Microsoft Installation Guide for more details.
  - https://docs.microsoft.com/en-us/azure-sphere/install/install-sdk

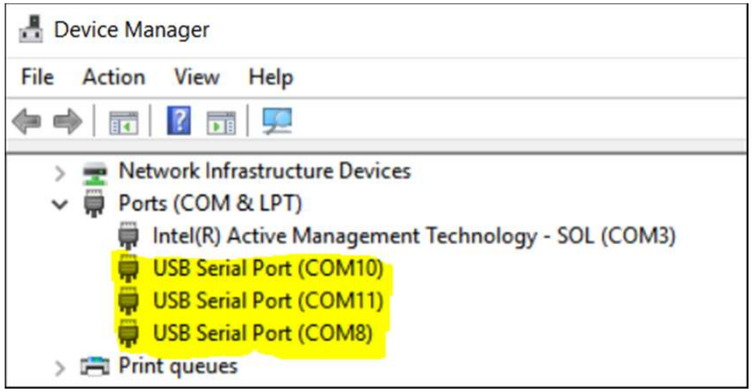
Verify Azure Sphere SDK Installation

- After install SDK, connect the MT3620 guardian board to the PC by USB. Open Device Manager and look for **TAP-Windows Adapter V9**.



Network adapters
Bluetooth Device (Personal Area Networ
Qualcomm 11ad Wireless Network Adap
Qualcomm Atheros QCA61x4A Wireless
Realtek PCIe GBE Family Controller
TAP-Windows Adapter V9
WAN Miniport (IKEv2)

- And three new COM ports. The numbers on your COM ports may be different from those in the figure.

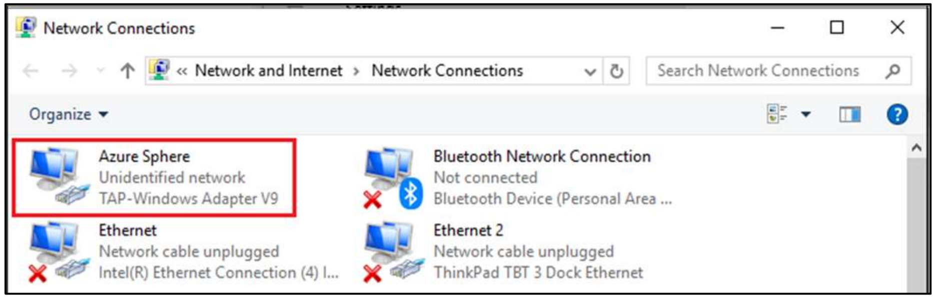


Network Infrastructure Devices
Ports (COM & LPT)
Intel(R) Active Management Technology - SOL (COM3)
USB Serial Port (COM10)
USB Serial Port (COM11)
USB Serial Port (COM8)
Print queues

Reference : https://docs.microsoft.com/en-us/azure-sphere/install/install-sdk

Verify TAP-Windows Adapter Configuration

- Check the Network connection **Azure Sphere (TAP-Windows Adapter V9)** is connected.



Network Connections

« Network and Internet › Network Connections

Search Network Connections

Organize ▾

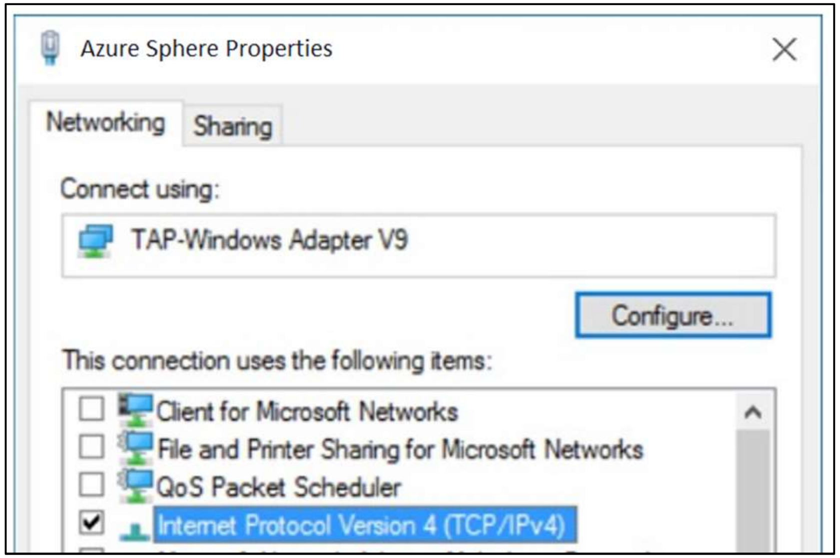
Azure Sphere
Unidentified network
TAP-Windows Adapter V9

Bluetooth Network Connection
Not connected
Bluetooth Device (Personal Area ...

Ethernet
Network cable unplugged
Intel(R) Ethernet Connection (4) I...

Ethernet 2
Network cable unplugged
ThinkPad TBT 3 Dock Ethernet

- Right-click on **Azure Sphere** and check it’s properties are as shown below:



Azure Sphere Properties

Networking | Sharing

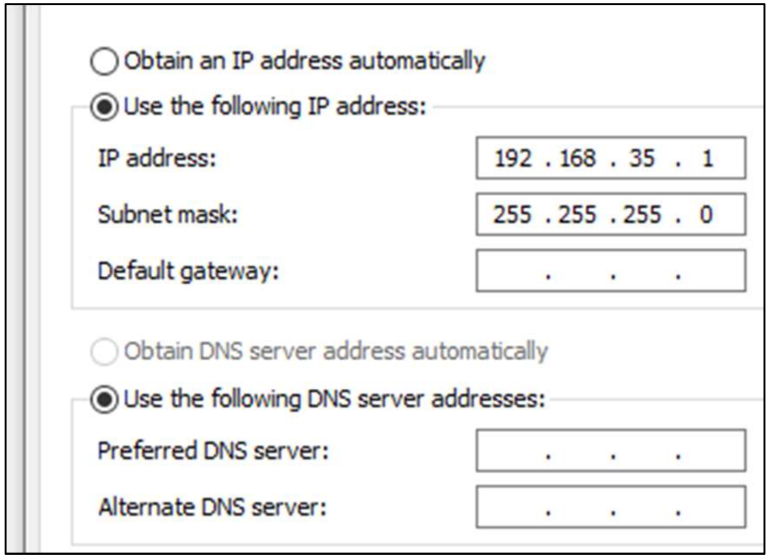
Connect using:

TAP-Windows Adapter V9

Configure...

This connection uses the following items:

□ Client for Microsoft Networks
□ File and Printer Sharing for Microsoft Networks
□ QoS Packet Scheduler
☑ Internet Protocol Version 4 (TCP/IPv4)



○ Obtain an IP address automatically
● Use the following IP address:

IP address: 192 . 168 . 35 . 1
Subnet mask: 255 . 255 . 255 . 0
Default gateway: . . .

○ Obtain DNS server address automatically
● Use the following DNS server addresses:

Preferred DNS server: . . .
Alternate DNS server: . . .

Create tenant and claim your device

- Sign in to Azure Sphere, using a Microsoft account:
  - **azsphere login**
  - If you've never logged in to Azure Sphere before, add the --newuser parameter along with your Microsoft account email address:
    - **azsphere login --newuser <email-address>**
- Type the following command to create a tenant.
  - **azsphere tenant create --name <my-tenant>**
  - For Example : azsphere tenant create –name “USITenant”
- Claim your device.
  - **azsphere device claim**
  - You should see output like this:

```
Claiming attached device ID
'ABCDE082513B529C45098884F882B2CA6D832587CAAE1A90B1CEC4A376EA2F22A96C4E7E1FC4D2AFF5633B68
DB68FF4420A5588B420851EE4F3F1A7DC51399ED' into tenant ID 'd343c263-4aa3-4558-adbb-d3fc34631800'.
Successfully claimed device ID Command completed successfully in 00:00:05.5459143.
```

Reference : https://docs.microsoft.com/en-us/azure-sphere/install/claim-device

Connect to WiFi

- Add your Wi-Fi network to the device by using the azsphere device wifi add command as follows:
  - **azsphere device wifi add --ssid <yourSSID> --psk <yourNetworkKey>**
  - MT3620 do not support WEP.
- Check the status of the WiFi connection
  - **azsphere device wifi show-status**
  - The following example shows successful results for a secure WPA2 connection:

```
SSID : NETGEAR21
Configuration state : enabled
Connection state : connected
Security state : psk
Frequency : 2442
Mode : station
Key management : WPA2-PSK
WPA State : COMPLETED
IP Address : 192.168.1.15
MAC Address : 52:cf:ff:3a:76:1b
Command completed successfully in 00:00:01.3976308.
```

Reference : https://docs.microsoft.com/en-us/azure-sphere/install/configure-wifi

Prepare development and debugging

- Make sure that your Azure Sphere device is connected to your PC, and your PC is connected to the internet.
- In an Azure Sphere Developer Command Prompt window, type the following command:
  - **azsphere device enable-development**
  - You should see output similar to the following:

```
Getting device capability configuration for application development.
Downloading device capability configuration for device ID
'ABCDE082513B529C45098884F882B2CA6D832587CAAE1A90B1CEC4A376EA2F22A96C4E7E1FC4D2AFF5633B68DB68FF4420A5588B420851EE4F3F1A7DC51399ED'.
Successfully downloaded device capability configuration.
Successfully wrote device capability configuration file 'C:\Users\user\AppData\Local\Temp\tmpD732.tmp'.
Setting device group ID 'a6df7013-c7c2-4764-8424-00cbacb431e5' for device with ID
'ABCDE082513B529C45098884F882B2CA6D832587CAAE1A90B1CEC4A376EA2F22A96C4E7E1FC4D2AFF5633B68DB68FF4420A5588B420851EE4F3F1A7DC51399ED'.
Successfully disabled over-the-air updates.
Enabling application development capability on attached device.
Applying device capability configuration to device.
Successfully applied device capability configuration to device.
The device is rebooting.
Installing debugging server to device.
Installation started.
Application development capability enabled.
Successfully set up device
'ABCDE082513B529C45098884F882B2CA6D832587CAAE1A90B1CEC4A376EA2F22A96C4E7E1FC4D2AFF5633B68DB68FF4420A5588B420851EE4F3F1A7DC51399ED' for application
development, and disabled over-the-air updates.
Command completed successfully in 00:00:17.1861625.
```

Azure Sphere Application

- Azure Sphere devices can run two types of applications:
  - High-level applications run containerized on the Azure Sphere OS
  - Real-time capable applications (RTApps) run on bare metal or with a real-time operating system (RTOS) on the real-time cores

- A high-level application is required for every Azure Sphere device; RTApps are optional.

Build and Run High-level Application

- Download https://github.com/Azure/azure-sphere-samples
- Start Visual Studio and go to **File>Open>CMake** to open **Samples\GPIO\GPIO_HighLevelApp\CMakeLists.txt**

- Edit the **AzureSphereTargetHardwareDefinitionDirectory** and **AzureSphereTargetHardwareDefinition** fields in the CMakeSettings.json file. For example:
  - **"AzureSphereTargetHardwareDefinitionDirectory": "<path to cloned samples>\Hardware\usi_mt3620_bt_evb"**
  - **"AzureSphereTargetHardwareDefinition": "usi_mt3620_bt_evb.json “**

  - **https://github.com/Azure/azure-sphere-samples/tree/master/Hardware**

Reference : **https://docs.microsoft.com/en-us/azure-sphere/install/qs-blink-application**

Build and Run High-level Application

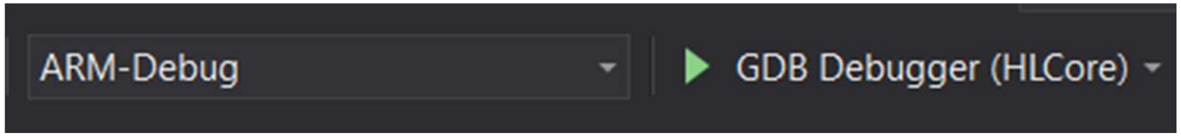
- Add GPIO pin
  - Samples/GPIO/GPIO_HighLevelApp/GPIO_HighLevelApp/app_manifest.json
  - GPIO 45 is LED APP RED

```
"Gpio": [ "$SAMPLE_BUTTON_1", "$MT3620_GPIO45 " ]
```

- Modify LED GPIO
  - Samples/GPIO/GPIO_HighLevelApp/GPIO_HighLevelApp/main.c

```
blinkingLedGpioFd =
GPIO_OpenAsOutput(MT3620_GPIO45, GPIO_OutputMode_PushPull, GPIO_Value_High);
```

- Ensure that your board is connected to your PC by USB. Then select **GDB Debugger (HLCore)** from the menu bar or press **F5**.



- LED APP on the MT3620 guardian board begins blinking red. Push button S2 to cycle the blink interval between three possible values. By default, the Output window shows output from Device Output

Set Up Hardware For Real-time App

- Make sure that your Azure Sphere device is connected to your PC, and your PC is connected to the internet.
- Right-click the Azure Sphere Developer Command Prompt shortcut and select **More>Run as administrator**. When the window opens, issue the following command:
  - **azsphere device enable-development --enablertcoredebugging**
- Use USB to UART Converter for HelloWorld_RTApp_MT3620_BareMetal sample to display output
- Connect GND on the USB to UART Converter to Header J32, pin 1 (GND) on the MT3620 guardian board.
- Connect RX on the USB to UART Converter to Header J32, pin 18 (IO0_TXD) on the MT3620 guardian board.
- On the PC, start the terminal emulator and open a 115200-8-N-1 terminal to the COM port that the adapter uses.

Reference : https://docs.microsoft.com/en-us/azure-sphere/install/qs-real-time-application

Build a real-time capable application-1

- Download https://github.com/Azure/azure-sphere-samples
- Start Visual Studio. On the **File** menu, select **Open>CMake...** and navigate to the folder that contains the sample HelloWorld\HelloWorld_RTApp_MT3620_BareMetal .
- If CMake generation does not start automatically, select the CMakeLists.txt file.
- In the Visual Studio Output window, the CMake output should show the messages **CMake generation started**. **and CMake generation finished.**
- On the CMake menu (if present), click Build All. If the menu is not present, open Solution Explorer, right-click the CMakeLists.txt file, and select Build. The output location of the Azure Sphere application appears in the Output window.

Build a real-time capable application-2

- On the Select Startup Item menu, select **GDB Debugger (RTCore).**
- Press F5 to deploy the application.
- The connected terminal emulator should display output from the HelloWorld_RTApp_MT3620_Baremetal program. The program sends the following words at one-second intervals:
  - **Tick**
  - **Tock**
- Use the debugging options in Visual Studio to set breakpoints, inspect variables, and try other debugging tasks.

Nordic nRF52 BLE Setup

Nordic nRF52 Setup Steps

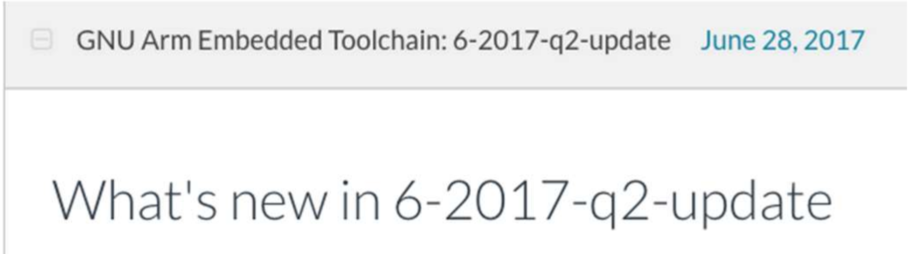
- Follow steps download and run debugger to Nordic nRF52
- OS: Linux /Mac OS

- Install GCC
- Install nRF5 Command Line Tools
- Download nRF52 SDK
- SEGGER J-Link connect to MT3620 EVB by SWD interface
- Build and download sample
- Start Debugger

Install GCC

- Download and install GCC
  - https://gcc.gnu.org/
  - In Ubuntu OS, you can type below command:
  - **$ sudo apt-get install gcc**

- Download gcc-arm-none-eabi-6-2017-q2-update
  - https://developer.arm.com/open-source/gnu-toolchain/gnu-rm/downloads



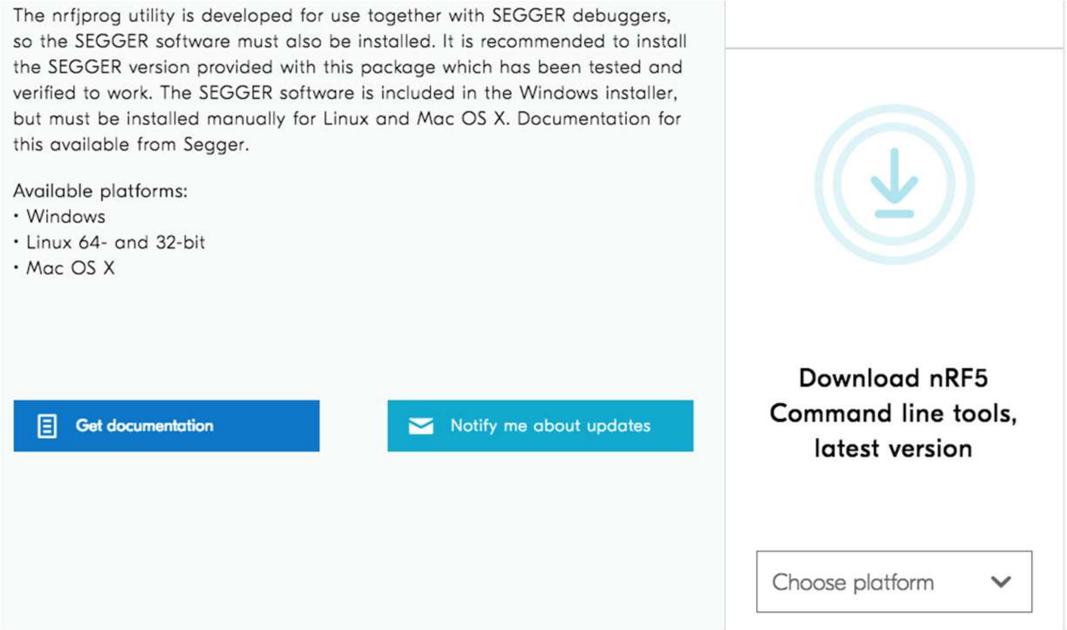
GNU Arm Embedded Toolchain: 6-2017-q2-update June 28, 2017

What's new in 6-2017-q2-update

  - Extract gcc-arm-none-eabi-6-2017-q2-update.tar.bz2 to **/usr/local/**
    - **Path : /usr/local/gcc-arm-none-eabi-6-2017-q2-update/**

Install nRF5 Command Line Tools

- Download and install nRF5 Command Line Tools
  - https://www.nordicsemi.com/Software-and-Tools/Development-Tools/nRF5-Command-Line-Tools



The nrfjprog utility is developed for use together with SEGGER debuggers, so the SEGGER software must also be installed. It is recommended to install the SEGGER version provided with this package which has been tested and verified to work. The SEGGER software is included in the Windows installer, but must be installed manually for Linux and Mac OS X. Documentation for this available from Segger.

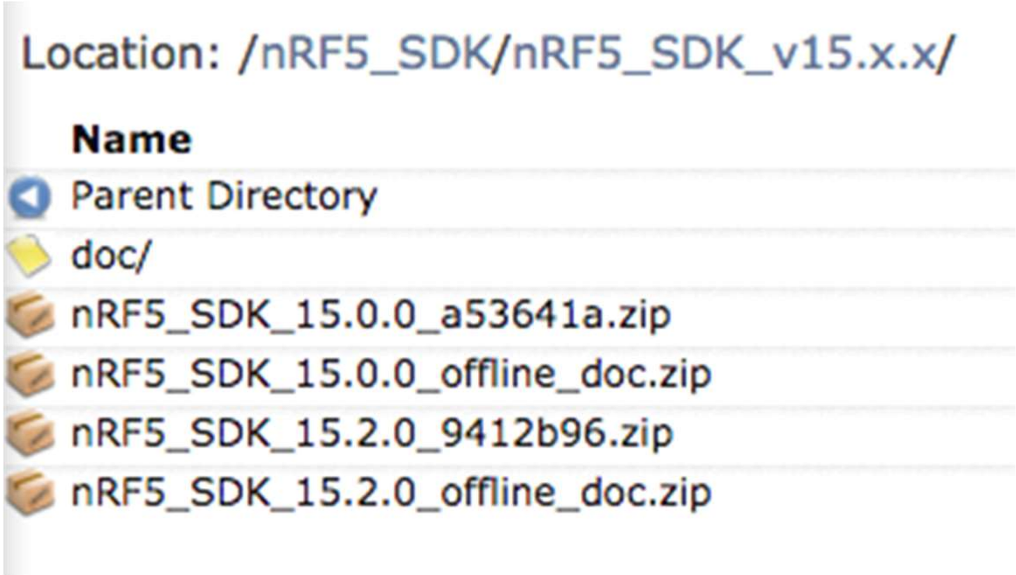
Available platforms:
- Windows
- Linux 64- and 32-bit
- Mac OS X

- Untar nRF5 Command Line Tools and set to $PATH
  - Add nRF5 Command Line Tools path to .bash_profile.
  - **$ echo export PATH=“< nRF5 Command Line Tools path >:$PATH" >> ~/.bash_profile**

Download nRF52 SDK

- Azure Sphere Combo Module BLE Chip: Nordic 52832
  - https://infocenter.nordicsemi.com/index.jsp

- Download nRF52 SDK (Current only support SDK 15.2.0)
  - https://developer.nordicsemi.com/nRF5_SDK/



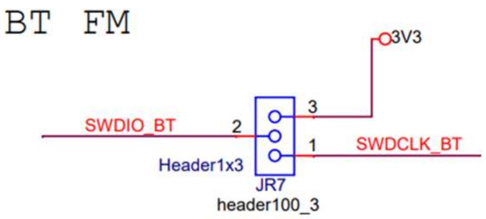
Location: /nRF5_SDK/nRF5_SDK_v15.x.x/

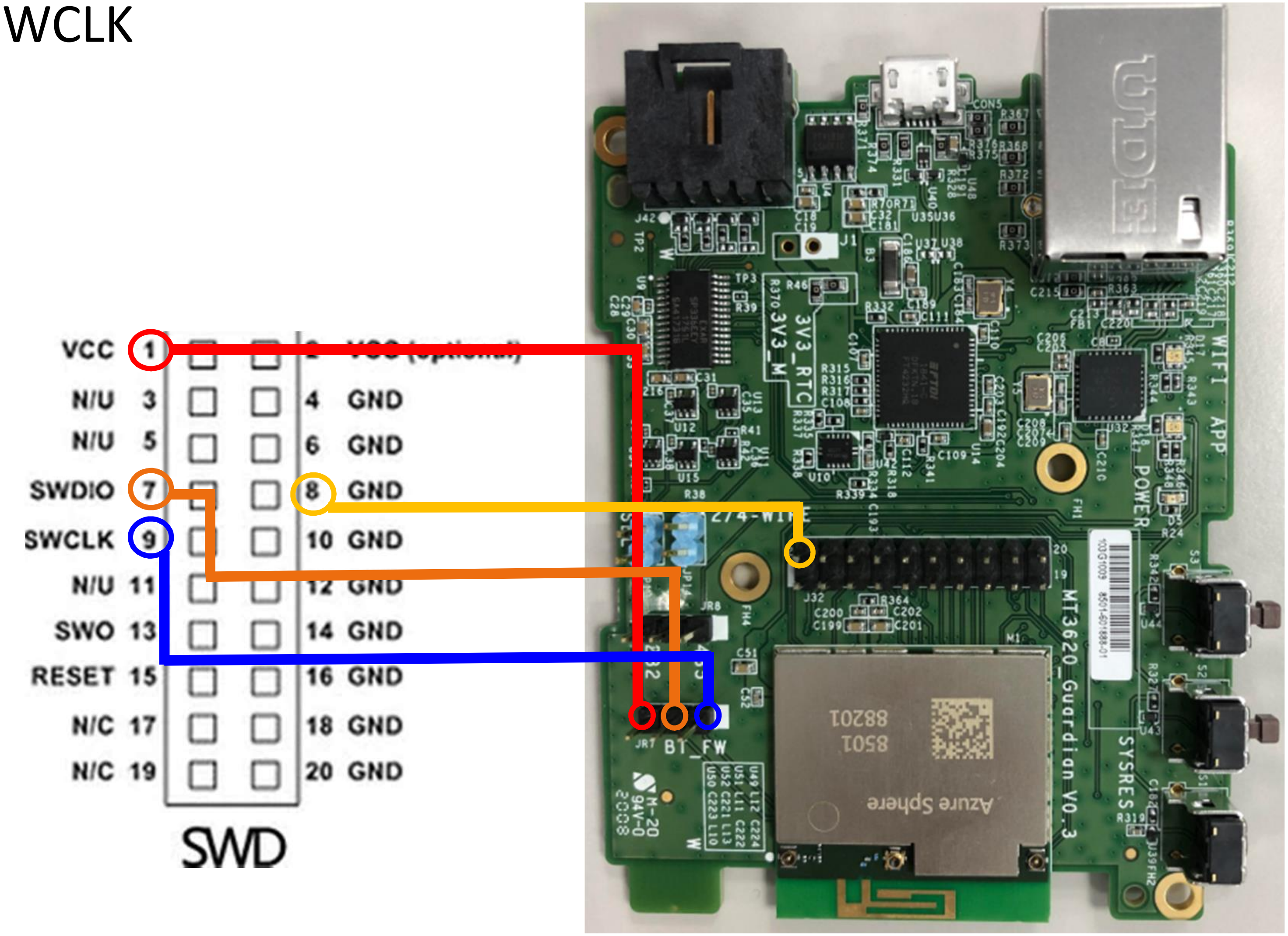
Name
Parent Directory
doc/
nRF5_SDK_15.0.0_a53641a.zip
nRF5_SDK_15.0.0_offline_doc.zip
nRF5_SDK_15.2.0_9412b96.zip
nRF5_SDK_15.2.0_offline_doc.zip

- Unzip nRF52 SDK 15.2.0

J-Link connect to Guardian Board

- SEGGER J-Link connect to MT3620 guardian board by SWD interface
  - VCC,GND,SWDIO,SWCLK





Build and download sample

- Entry SDK sample path (Here use saadc sample on nRF52832)
  - nRF52832 : pca10040
  - **$ cd examples/peripheral/saadc/pca10040/blank/armgcc**
- Build and Download sample
  - **$ make flash**
  - You should see output similar to the following:

```
DONE nrf52832_xxaa
Flashing: _build/nrf52832_xxaa.hex
nrfjprog -f nrf52 --program _build/nrf52832_xxaa.hex --sectorerase
Parsing hex file.
Erasing page at address 0x0.
Erasing page at address 0x1000.
Erasing page at address 0x2000.
Erasing page at address 0x3000.
Applying system reset.
Checking that the area to write is not protected.
Programming device.
nrfjprog -f nrf52 --reset
Applying system reset.
Run.
```

Start Debugger - 1

- 1. Download and install J-Link Software and Documentation pack
  - https://www.segger.com/downloads/jlink/#J-LinkSoftwareAndDocumentationPack

- 2. Modify **saadc/pca10040/blank/config/sdk_config.h**
  - Modify to **#define NRF_LOG_BACKEND_RTT_ENABLED 1**

- 3. Build and Download sample
  - **$ make flash**

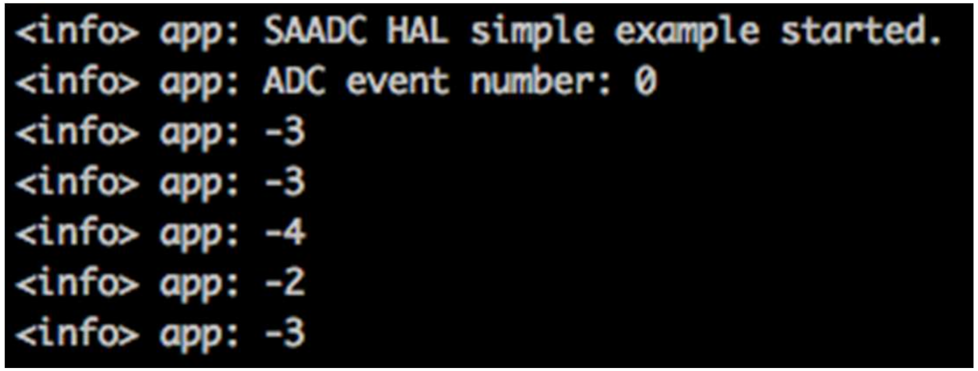
Start Debugger - 2

- Create a new terminal and type below command:
  - **$ JLinkExe -device nRF52 -speed 4000 -if SWD -autoconnect 1**
  - You should see output similar to the following:

```
SEGGER J-Link Commander V6.40 (Compiled Oct 26 2018 15:07:23)
DLL version V6.40, compiled Oct 26 2018 15:07:12
Connecting to J-Link via USB...O.K.
Firmware: J-Link OB-SAM3U128-V2-NordicSemi compiled Jul 12 2018 11:44:41
Hardware version: V1.00
Device "NRF52" selected.
Connecting to target via SWD
Found SW-DP with ID 0x2BA01477
Found SW-DP with ID 0x2BA01477
Scanning AP map to find all available APs
AP[2]: Stopped AP scan as end of AP map has been reached
AP[0]: AHB-AP (IDR: 0x24770011)
AP[1]: JTAG-AP (IDR: 0x02880000)
Iterating through AP map to find AHB-AP to use
AP[0]: Core found
AP[0]: AHB-AP ROM base: 0xE00FF000
CPUID register: 0x410FC241. Implementer code: 0x41 (ARM)
Found Cortex-M4 r0p1, Little endian.
FPUnit: 6 code (BP) slots and 2 literal slots
CoreSight components:
…
ROMTbl[0][5]: E0041000, CID: B105900D, PID: 000BB925 ETM
Cortex-M4 identified.
J-Link>
```

Start Debugger - 3

- Create a new terminal and type below command:
  - **$ JLinkRTTClient**
  - The debug message will show on terminal



```
<info> app: SAADC HAL simple example started.
<info> app: ADC event number: 0
<info> app: -3
<info> app: -3
<info> app: -4
<info> app: -2
<info> app: -3
```

Nordic nRF52 GPIO PIN Define

- Nordic nRF52 GPIO pin define with MT3620

nRF52 GPIO	MT3620 GPIO
P0_02/AIN0	BT_PA_EN(GPIO_21)
P0_03/AIN1	BT_LNA_EN(GPIO_20)
P0_04/AIN2	GPIO_04_BT (J34 EVB pin out)
P0_05/AIN3	GPIO_05_BT (J34 EVB pin out)
P0_07	BT_FW_EN(DFU)(GPIO_23)
P0_09	NFC1 (J28 EVB pin out)
P0_10	NFC2 (J28 EVB pin out)
P0_21/RST	BT_nRST(GPIO_22)
P0_25(UART_RTS)	GPIO74_MT_CTS4 (ISU4:[UART_Id]8)
P0_26(UART_TX)	GPIO73_MT_RX4 (ISU4:[UART_Id]8)
P0_27(UART_CTS)	GPIO72_MT_RTS4 (ISU4:[UART_Id]8)
P0_28/AIN4(UART_RX)	GPIO71_MT_TX4 (ISU4:[UART_Id]8)

- You need add **#define CONFIG_NFCT_PINS_AS_GPIOS** to config nRF52 GPIO 9 and 10 in sample code.
- GPIO_04_BT and GPIO_05_BT only support MT3620_EVB_V2.0 and later.

Config to Internal RC Oscillator

- Azure Sphere Combo Module BLE chip use internal RC oscillator. Please set this config for BLE app.
  - In Nrf52App/pca10040/s132/config/sdk_config.h

```
#define NRFX_CLOCK_CONFIG_LF_SRC 0
#define NRF_SDH_CLOCK_LF_SRC 0
#define NRF_SDH_CLOCK_LF_RC_CTIV 16
#define NRF_SDH_CLOCK_LF_RC_TEMP_CTIV 4
#define NRF_SDH_CLOCK_LF_ACCURACY 1
```

BLE-based Wi-Fi setup sample

BLE-based Wi-Fi setup sample

- This sample demo how to setup MT3620 WIFI by Nordic nRF52 BLE .For more detail, please refer:
  - https://github.com/Azure/azure-sphere-samples/tree/master/Samples/WifiSetupAndDeviceControlViaBle

- Follow below step to configure for MT3620 Guardian Board

Nordic nRF52 App

- Modify Nordic SDK path
  - In Nrf52App/usi_evb/s132/armgcc/Makefile
  - SDK_ROOT := <CHANGE_THIS_TO_YOUR_NORDIC_SDK_PATH>

- Modify to internal RC oscillator
  - In Nrf52App/usi_evb/s132/config/sdk_config.h

```
#define NRFX_CLOCK_CONFIG_LF_SRC 0
#define NRF_SDH_CLOCK_LF_SRC 0
#define NRF_SDH_CLOCK_LF_RC_CTIV 16
#define NRF_SDH_CLOCK_LF_RC_TEMP_CTIV 4
#define NRF_SDH_CLOCK_LF_ACCURACY 1
```

Reference : https://github.com/Azure/azure-sphere-samples/tree/master/Samples/WifiSetupAndDeviceControlViaBle

Build and download BLE Sample

- J-Link connect to MT3620 Guardian Board by SWD interface
- Entry BLE Sample folder
  - **$ cd azure-sphere-samples/Samples/WifiSetupAndDeviceControlViaBle/Nrf52App/usi_evb/s132/armgcc/**
- Download Nordic nRF5_SDK_15.2.0 s132 softdevice
  - **$ make flash_softdevice**
- Build and Download sample
  - **$ make flash**

Build and Run sample

- Run the Azure Sphere app
  - Start Visual Studio and go to File>Open>CMake to open Samples\WifiSetupAndDeviceControlViaBle\AzureSphere_HighLevelApp\CMakeLists.txt
  - Edit the AzureSphereTargetHardwareDefinitionDirectory and AzureSphereTargetHardwareDefinition fields for USI MT3620 EVB(usi_mt3620_bt_evb) in the CMakeSettings.json
  - Modify GPIO Pin for Guardian Board LED
    - In AzureSphere_HighLevelApp/WifiSetupAndDeviceControlViaBle/app_manifest.json

```
"Gpio": [
    "$MT3620_GPIO45",
    "$MT3620_GPIO46 ",
    "$MT3620_GPIO47 ",
    "$MT3620_GPIO48 ",
    "$SAMPLE_BUTTON_1",
    "$SAMPLE_BUTTON_2",
    "$SAMPLE_NRF52_RESET",
    "$SAMPLE_NRF52_DFU"
  ],
```

    - In AzureSphere_HighLevelApp/WifiSetupAndDeviceControlViaBle/main.c
      - Replace SAMPLE_RGBLED_RED to MT3620_GPIO45
      - Replace SAMPLE_RGBLED_GREEN to MT3620_GPIO46
      - Replace SAMPLE_RGBLED_BLUE to MT3620_GPIO47
      - Replace SAMPLE_DEVICE_STATUS_LED to MT3620_GPIO48
  - Then select GDB Debugger (HLCore) from the menu bar or press F5.
  - Press button S2 on the MT3620 board. The Azure Sphere app will start the nRF52, wait for notification that the nR52 app is active, and then request that it begin advertising.

Run Windows sample

- Run the Windows 10 companion app on your PC
  - Start a separate instance of Visual Studio.
  - Open WifiSetupAndDeviceControlViaBle/WindowsApp/WifiSetupAndDeviceControlViaBle.sln
  - Build and debug the application (F5).
  - For more details please refer to https://github.com/Azure/azure-sphere-samples/tree/master/Samples/WifiSetupAndDeviceControlViaBle

Nordic BLE DFU

External MCU update

- This External MCU update solution demo how you might deploy an update to an external MCU(Nordic BLE Chip) device using Azure Sphere. For more detail, please refer:
    - https://github.com/Azure/azure-sphere-samples/tree/master/Samples/ExternalMcuUpdate

- Follow below step to configure for MT3620 Guardian Board

Nordic nRF52 Bootloader

- J-Link connect to MT3620 EVB by SWD interface
- Entry BLE Sample folder
  - **$ cd azure-sphere-samples/Samples/ExternalMcuUpdate/Binaries/**

- Follow below step to download Nordic nRF5_SDK_15.2.0 bootloader
  - **$ nrfjprog -f nrf52 --eraseall**
  - **$ nrfjprog -f nrf52 --program usi_evb_Softdevice_Bootloader.hex --sectorerase**
  - **$ nrfjprog -f nrf52 --reset**

Nordic nRF52 Blinky App

- Follow below to build Blinky app FW
  - https://github.com/Azure/azure-sphere-samples/tree/master/Samples/ExternalMcuUpdate#build-and-deploy-your-own-app-firmware-for-the-nrf52

- Blinky app must config to internal RC oscillator
  - nRF5_SDK_15.2.0_9412b96/examples/ble_peripheral/ble_app_blinky/pca10040/s132/config/sdk_config.h

```
#define NRFX_CLOCK_CONFIG_LF_SRC 0
#define NRF_SDH_CLOCK_LF_SRC 0
#define NRF_SDH_CLOCK_LF_RC_CTIV 16
#define NRF_SDH_CLOCK_LF_RC_TEMP_CTIV 4
#define NRF_SDH_CLOCK_LF_ACCURACY 1
```

Build and Run High-level Application

- Start Visual Studio and go to **File>Open>CMake** to open **Samples\ExternalMcuUpdate\AzureSphere_HighLevelApp\CMakeLists.txt**
- Edit the **AzureSphereTargetHardwareDefinitionDirectory** and **AzureSphereTargetHardwareDefinition** fields in the CMakeSettings.json file. For example:
  - **"AzureSphereTargetHardwareDefinitionDirectory": "<path to cloned samples>\Hardware\usi_mt3620_bt_evb"**
  - **"AzureSphereTargetHardwareDefinition": "usi_mt3620_bt_evb.json “**
  - **https://github.com/Azure/azure-sphere-samples/tree/master/Hardware**
- Then select GDB Debugger (HLCore) from the menu bar or press F5.

Private Ethernet Sample

Build and install sample

- Package and deploy the board configuration image for the Microchip ENC286J60 Ethernet chip, command :
  - **azsphere image package-board-config --preset lan-enc28j60-isu0-int5 --output enc28j60-isu0-int5.imagepackage**
  - **azsphere device sideload deploy --imagepackage enc28j60-isu0-int5.imagepackage**
- **Note: This board configuration uses ISU0 (I2C/SPI/UART port 0) on the MT3620, which is also used by other samples. For now, it's not possible to adapt this Private Ethernet sample to use another ISU port.**
- Run the Azure Sphere app
  - Clone the Azure Sphere samples repo and find the PrivateNetworkServices sample.
  - Start Visual Studio and go to File>Open>CMake to open Samples\PrivateNetworkServices\CMakeLists.txt
  - Edit the AzureSphereTargetHardwareDefinitionDirectory and AzureSphereTargetHardwareDefinition fields for USI MT3620 EVB(usi_mt3620_bt_evb) in the CMakeSettings.json
  - Then select GDB Debugger (HLCore) from the menu bar or press F5.

Reference : https://docs.microsoft.com/en-us/azure-sphere/network/connect-ethernet
https://github.com/Azure/azure-sphere-samples/tree/master/Samples/PrivateNetworkServices

Reference

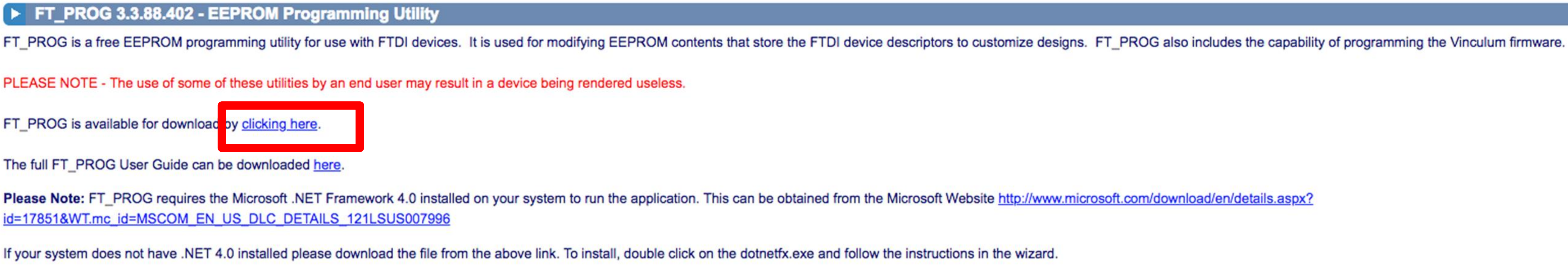
- FTDI FT_PROG
  - https://www.ftdichip.com/Support/Utilities.htm#FT_PROG
- FTDI Driver
  - https://www.ftdichip.com/Drivers/VCP.htm
- Azure Sphere Hardware Designs
  - https://github.com/Azure/azure-sphere-hardware-designs/tree/master/P-MT3620RDB-1-0
- Azure Sphere
  - https://docs.microsoft.com/en-us/azure-sphere/
- Visual Studio
  - https://visualstudio.microsoft.com/
- Azure Sphere Samples
  - https://github.com/Azure/azure-sphere-samples
- Nordic Development Tools
  - https://www.nordicsemi.com/Software-and-Tools/Development-Tools/nRF5-Command-Line-Tools
- Nordic SDK
  - https://developer.nordicsemi.com/nRF5_SDK/
- Segger J-Link tools
  - https://www.segger.com/downloads/jlink/
- Nordic Infocenter
  - https://infocenter.nordicsemi.com

Additional

FTDI FT_PROG programming tool

- The FTDI interface chip provides a set of pins that must be connected to a small EEPROM that is used to store manufacturer's details and a serial number. After board assembly, this information is programmed into the EEPROM over USB using a software tool provided by FTDI,

- https://www.ftdichip.com/Support/Utilities.htm#FT_PROG
- Download and install FT_PROG in the default location to your Computer.
- FT_PROG default location (C:\Program Files(x86)\FTDI\FT_Prog\)



FT_PROG 3.3.88.402 - EEPROM Programming Utility

FT_PROG is a free EEPROM programming utility for use with FTDI devices. It is used for modifying EEPROM contents that store the FTDI device descriptors to customize designs. FT_PROG also includes the capability of programming the Vinculum firmware.

PLEASE NOTE - The use of some of these utilities by an end user may result in a device being rendered useless.

FT_PROG is available for download by clicking here.

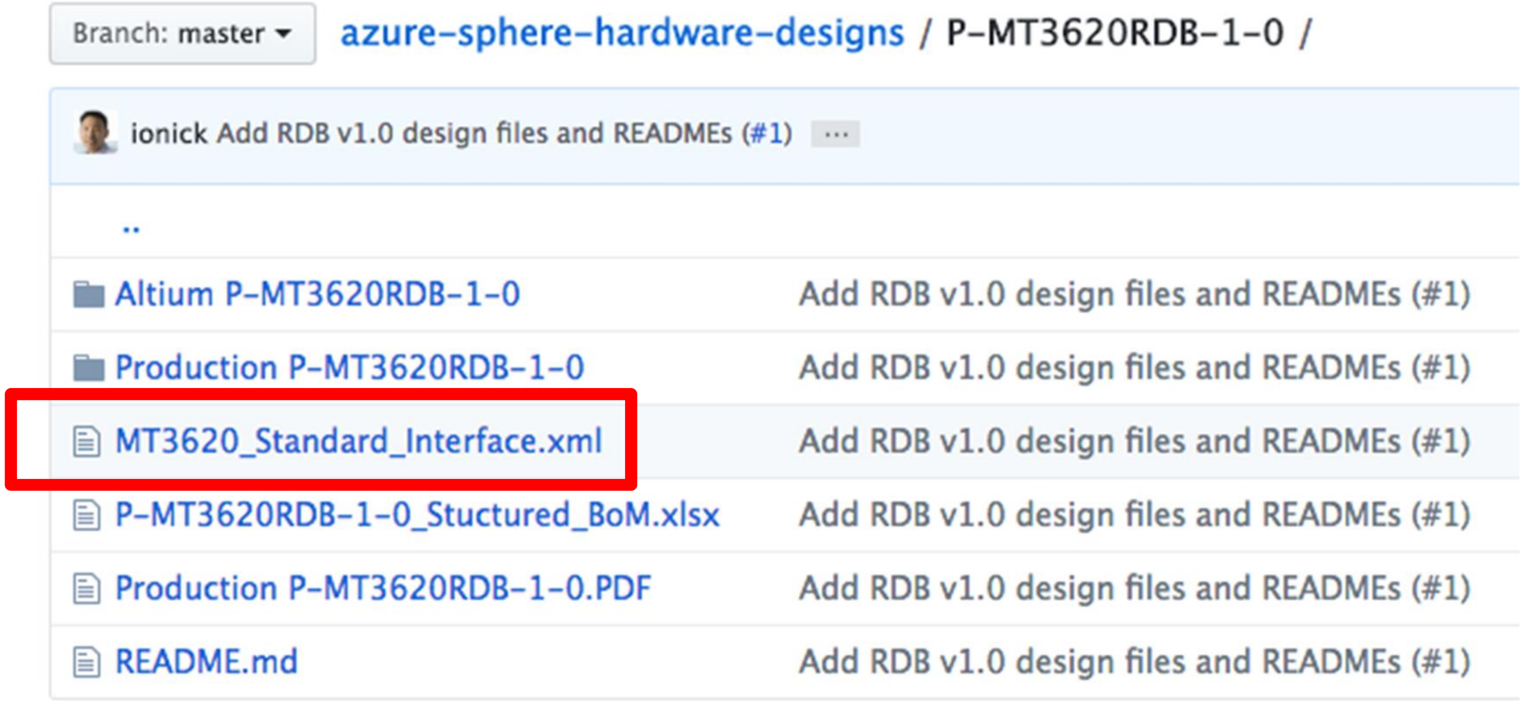
The full FT_PROG User Guide can be downloaded here.

Please Note: FT_PROG requires the Microsoft .NET Framework 4.0 installed on your system to run the application. This can be obtained from the Microsoft Website http://www.microsoft.com/download/en/details.aspx?id=17851&WT.mc_id=MSCOM_EN_US_DLC_DETAILS_121LSUS007996

If your system does not have .NET 4.0 installed please download the file from the above link. To install, double click on the dotnetfx.exe and follow the instructions in the wizard.

Download configuration file

- https://github.com/Azure/azure-sphere-hardware-designs/tree/master/P-MT3620RDB-1-0
- Download MT3620 standard configuration file
- MT3620_Standard_Interface.xml => Right Click => Save File



Branch: master ▾ azure-sphere-hardware-designs / P-MT3620RDB-1-0 /

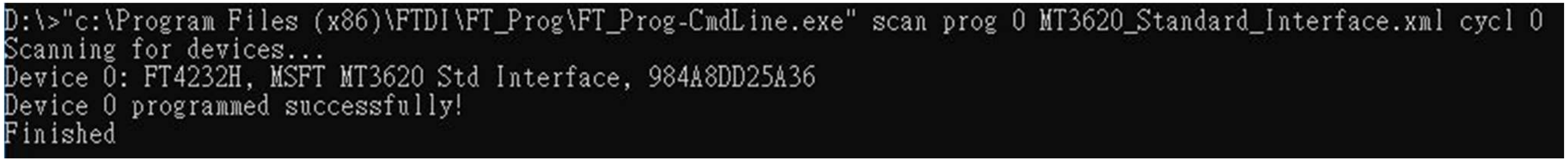
ionick Add RDB v1.0 design files and READMEs (#1) …

..

Altium P-MT3620RDB-1-0	Add RDB v1.0 design files and READMEs (#1)
Production P-MT3620RDB-1-0	Add RDB v1.0 design files and READMEs (#1)
MT3620_Standard_Interface.xml	Add RDB v1.0 design files and READMEs (#1)
P-MT3620RDB-1-0_Stuctured_BoM.xlsx	Add RDB v1.0 design files and READMEs (#1)
Production P-MT3620RDB-1-0.PDF	Add RDB v1.0 design files and READMEs (#1)
README.md	Add RDB v1.0 design files and READMEs (#1)

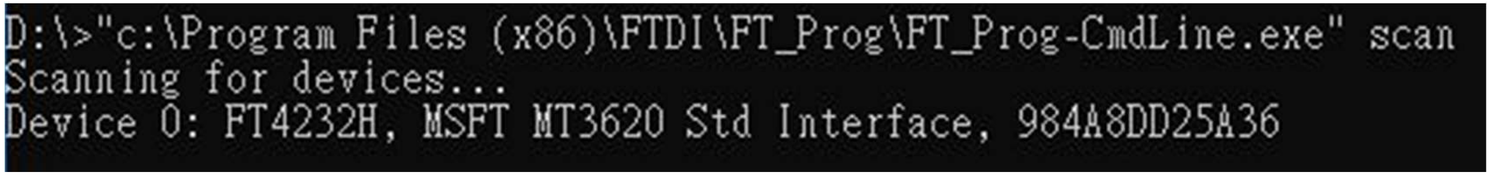
FTDI EEPROM programming

- Ensure that the MT3620 dev kit is the only development board attached to the host PC.
- Open a command prompt (for example, cmd.exe) and change to the folder where you saved the MT3620_Standard_Interface.xml
- Type the following command:



```
D:\>"c:\Program Files (x86)\FTDI\FT_Prog\FT_Prog-CmdLine.exe" scan prog 0 MT3620_Standard_Interface.xml cycl 0
Scanning for devices...
Device 0: FT4232H, MSFT MT3620 Std Interface, 984A8DD25A36
Device 0 programmed successfully!
Finished
```

- To verify that programming was successful, enter:



```
D:\>"c:\Program Files (x86)\FTDI\FT_Prog\FT_Prog-CmdLine.exe" scan
Scanning for devices...
Device 0: FT4232H, MSFT MT3620 Std Interface, 984A8DD25A36
```

FCC Certification Statement-1

Federal Communication Commission Interference Statement
This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.
This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one of the following measures:

- **Reorient or relocate the receiving antenna.**
- **Increase the separation between the equipment and receiver.**
- **Connect the equipment into an outlet on a circuit different from that**
- **to which the receiver is connected.**
- **Consult the dealer or an experienced radio/TV technician for help.**

FCC Certification Statement-2

FCC Caution:
Any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate this equipment.
This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.
This device meets all the other requirements specified in Part 15E, Section 15.407 of the FCC Rules.

Radiation Exposure Statement:
This equipment complies with FCC radiation exposure limits set forth for an uncontrolled environment.
This equipment should be installed and operated with minimum distance 20cm between the radiator & your body.

This module is intended for OEM integrators only. Per FCC KDB 996369 D03 OEM Manual v01 guidance, the following conditions must be strictly followed when using this certified module:

KDB 996369 D03 OEM Manual v01 rule sections:
2.2 List of applicable FCC rules
This module has been tested for compliance to FCC Part 15.
2.3 Summarize the specific operational use conditions
The module is tested for standalone mobile RF exposure use condition. Any other usage conditions such as co-location with other transmitter(s) or being used in a portable condition will need a separate reassessment through a class II permissive change application or new certification.
2.4 Limited module procedures
Not applicable.
2.5 Trace antenna designs
Not applicable.

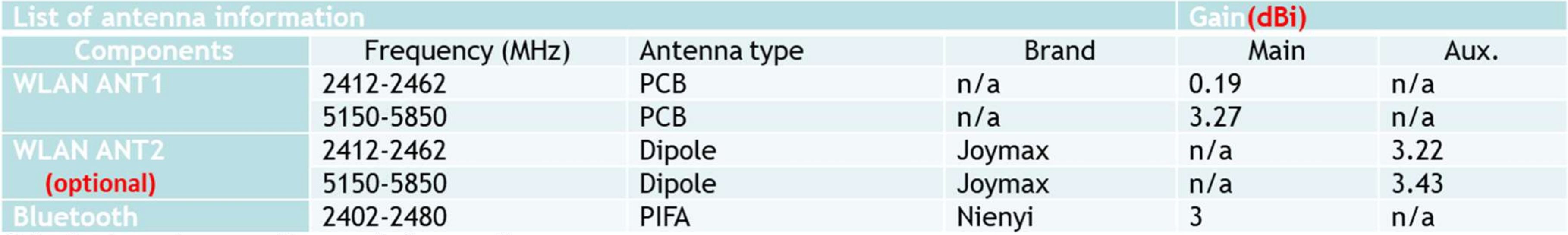
FCC Certification Statement-3

2.6 RF exposure considerations

This equipment complies with FCC mobile radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with a minimum distance of 20cm between the radiator & your body. If the module is installed in a portable host, a separate SAR evaluation is required to confirm compliance with relevant FCC portable RF exposure rules.

2.7 Antennas

The following antennas have been certified for use with this module; antennas of the same type with equal or lower gain may also be used with this module. The antenna must be installed such that 20cm can be maintained between the antenna and users.



List of antenna information				Gain(dBi)	
Components	Frequency (MHz)	Antenna type	Brand	Main	Aux.
WLAN ANT1	2412-2462	PCB	n/a	0.19	n/a
	5150-5850	PCB	n/a	3.27	n/a
WLAN ANT2 (optional)	2412-2462	Dipole	Joymax	n/a	3.22
	5150-5850	Dipole	Joymax	n/a	3.43
Bluetooth	2402-2480	PIFA	Nienyi	3	n/a

2.8 Label and compliance information

The final end product must be labeled in a visible area with the following: “Contains FCC ID: COF-AS01”. The grantee's FCC ID can be used only when all FCC compliance requirements are met.

2.9 Information on test modes and additional testing requirements

This transmitter is tested in a standalone mobile RF exposure condition and any co-located or simultaneous transmission with other transmitter(s) or portable use will require a separate class II permissive change re-evaluation or new certification.

FCC Certification Statement-4

2.10 Additional testing, Part 15 Subpart B disclaimer

This transmitter module is tested as a subsystem and its certification does not cover the FCC Part 15 Subpart B (unintentional radiator) rule requirement applicable to the final host. The final host will still need to be reassessed for compliance to this portion of rule requirements if applicable.

As long as all conditions above are met, further transmitter test will not be required. However, the OEM integrator is still responsible for testing their end-product for any additional compliance requirements required with this module installed.

IMPORTANT NOTE: In the event that these conditions can not be met (for example certain laptop configurations or co-location with another transmitter), then the FCC authorization is no longer considered valid and the FCC ID can not be used on the final product. In these circumstances, the OEM integrator will be responsible for re-evaluating the end product (including the transmitter) and obtaining a separate FCC authorization.

IC Certification Statement-1

Industry Canada statement

This device complies with ISED’s licence-exempt RSSs. Operation is subject to the following two conditions:
this device may not cause interference, and
this device must accept any interference, including interference that may cause undesired operation of the device.
Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes:
l'appareil ne doit pas produire de brouillage, et
l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

CAN ICES-3(B)/ NMB-3(B)

This device and its antenna(s) must not be co-located or operating in conjunction with any other antenna or transmitter, except tested built-in radios.
Cet appareil et son antenne ne doivent pas être situés ou fonctionner en conjonction avec une autre antenne ou un autre émetteur, exception faites des radios intégrées qui ont été testées.

The County Code Selection feature is disabled for products marketed in the US/ Canada.
La fonction de sélection de l'indicatif du pays est désactivée pour les produits commercialisés aux États-Unis et au Canada.

IC Certification Statement-2

This radio transmitter (IC: 10293A-AS01 / Model: AS-01) has been approved by ISED to operate with the antenna type listed below with maximum permissible gain indicated. Antenna types not included in this list, having a gain greater than the maximum gain indicated for that type, are strictly prohibited for use with this device.

Le présent émetteur radio (IC: 10293A-AS01 / Model: AS-01) a été approuvé par ISED pour fonctionner avec les types d'antenne énumérés ci-dessous et ayant un gain admissible maximal. Les types d'antenne non inclus dans cette liste, et dont le gain est supérieur au gain maximal indiqué, sont strictement interdits pour l'exploitation de l'émetteur.

Approved antenna(s) list

List of antenna information				Gain(dBi)	
Components	Frequency (MHz)	Antenna type	Brand	Main	Aux.
WLAN ANT1	2412-2462	PCB	n/a	0.19	n/a
	5150-5850	PCB	n/a	3.27	n/a
WLAN ANT2 (optional)	2412-2462	Dipole	Joymax	n/a	3.22
	5150-5850	Dipole	Joymax	n/a	3.43
Bluetooth	2402-2480	PIFA	Nienyi	3	n/a

Radiation Exposure Statement:

This equipment complies with IC radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with minimum distance 20cm between the radiator & your body.

Déclaration d'exposition aux radiations:

Cet équipement est conforme aux limites d'exposition aux rayonnements IC établies pour un environnement non contrôlé. Cet équipement doit être installé et utilisé avec un minimum de 20cm de distance entre la source de rayonnement et votre corps.

IC Certification Statement-3

Caution:
The local network device user guide should include specific instructions on the above restrictions, including:
the device for operation in the band 5150–5250MHz is only for indoor use to reduce the potential for harmful interference to co-channel mobile satellite systems; for devices with detachable antenna(s), the maximum antenna gain permitted for devices in the bands 5250-5350MHz and 5470-5725MHz shall be such that the equipment still complies with the e.i.r.p. limit;for devices with detachable antenna(s), the maximum antenna gain permitted for devices in the band 5725-5850MHz shall be such that the equipment still complies with the e.i.r.p. limits as appropriate;

Avertissement:
Le guide d'utilisation des dispositifs pour réseaux locaux doit inclure des instructions précises sur les restrictions susmentionnées, notamment :les dispositifs fonctionnant dans la bande de 5150 à 5250MHz sont réservés uniquement pour une utilisatin à l'intérieur afin de réduire les risques de brouillage préjudiciable aux systèmes de satellites mobiles utilisant les mêmes canaux;pour les dispositifs munis d'antennes amovibles, le gain maximal d'antenne permis pour les dispositifs utilisant les bandes de 5250 à 5350MHz et de 5470 à 5725MHz doit être conforme à la limite de la p.i.r.e; pour les dispositifs munis d'antennes amovibles, le gain maximal d'antenne permis (pour les dispositifs utilisant la bande de 5725 à 5850MHz) doit être conforme à la limite de la p.i.r.e. spécifiée, selon le cas;

IC Certification Statement-4

This device is intended only for OEM integrators under the following conditions: (For module device use)
1) The antenna must be installed such that 20 cm is maintained between the antenna and users, and
2) The transmitter module may not be co-located with any other transmitter or antenna.
As long as 2 conditions above are met, further transmitter test will not be required. However, the OEM integrator is still responsible for testing their end-product for any additional compliance requirements required with this module installed.

Cet appareil est conçu uniquement pour les intégrateurs OEM dans les conditions suivantes: (Pour utilisation de dispositif module)
1) L'antenne doit être installée de telle sorte qu'une distance de 20 cm est respectée entre l'antenne et les utilisateurs, et
2) Le module émetteur peut ne pas être coïmplanté avec un autre émetteur ou antenne.
Tant que les 2 conditions ci-dessus sont remplies, des essais supplémentaires sur l'émetteur ne seront pas nécessaires. Toutefois, l'intégrateur OEM est toujours responsable des essais sur son produit final pour toutes exigences de conformité supplémentaires requis pour ce module installé.
IMPORTANT NOTE:
In the event that these conditions can not be met (for example certain laptop configurations or co-location with another transmitter), then the Canada authorization is no longer considered valid and the IC ID can not be used on the final product. In these circumstances, the OEM integrator will be responsible for re-evaluating the end product (including the transmitter) and obtaining a separate Canada authorization.
NOTE IMPORTANTE:
Dans le cas où ces conditions ne peuvent être satisfaites (par exemple pour certaines configurations d'ordinateur portable ou de certaines co-localisation avec un autre émetteur), l'autorisation du Canada n'est plus considéré comme valide et l'ID IC ne peut pas être utilisé sur le produit final. Dans ces circonstances, l'intégrateur OEM sera chargé de réévaluer le produit final (y compris l'émetteur) et l'obtention d'une autorisation distincte au Canada.

IC Certification Statement-5

Label and compliance information
The final product must be labeled in a visible area with the following: "Contains IC: 10293A-AS01". The grantee's IC ID can be used only when all IC compliance requirements are met.

Étiquette et informations de conformité
Le produit final doit être étiqueté dans une zone visible avec les éléments suivants: "Contient IC: 10293A-AS01". L'identifiant IC du bénéficiaire ne peut être utilisé que lorsque toutes les exigences de conformité IC sont remplies.

Thank You

www.usiglobal.com