

USI Azure Sphere Guardian Board Example User Guide v1.0

Terry Wang
SW Engineer
WCS/WP1
Dec. 03, 2019

Outline

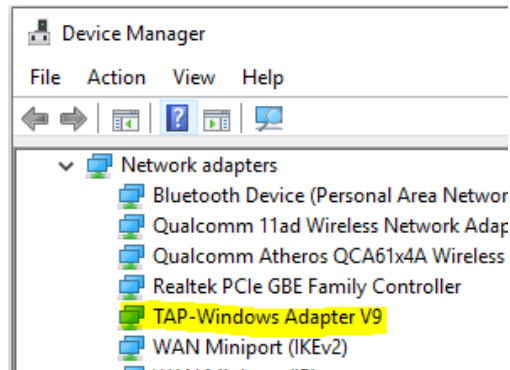
- Azure Sphere Setup
- Create an Azure IoT Central application
- Setup Azure Sphere Guardian Example

Install Azure Sphere SDK and Tools

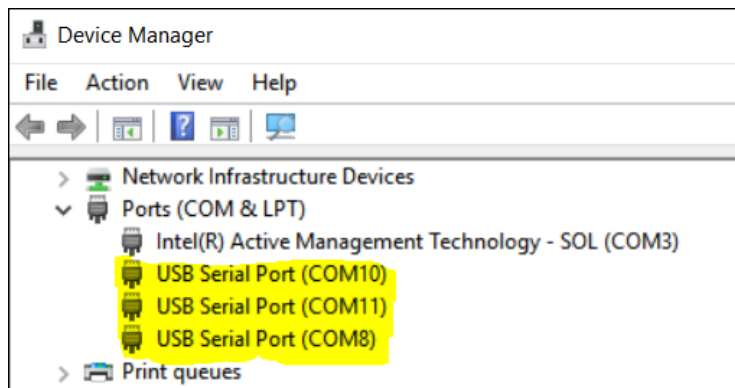
- Install FTDI driver
 - <https://www.ftdichip.com/Drivers/VCP.htm>
- Install Visual Studio 2019 Enterprise, Professional, or Community version 16.04 or later; or Visual Studio 2017 version 15.9 or later.
 - <https://visualstudio.microsoft.com/>
- Install the Azure Sphere SDK Preview for Visual Studio
 - <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>

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**.



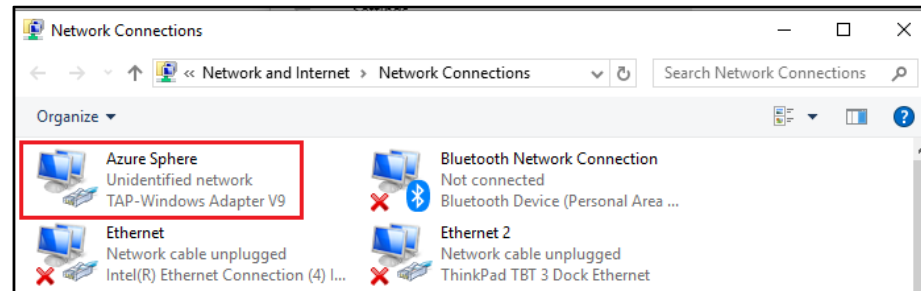
- And three new COM ports. The numbers on your COM ports may be different from those in the figure.



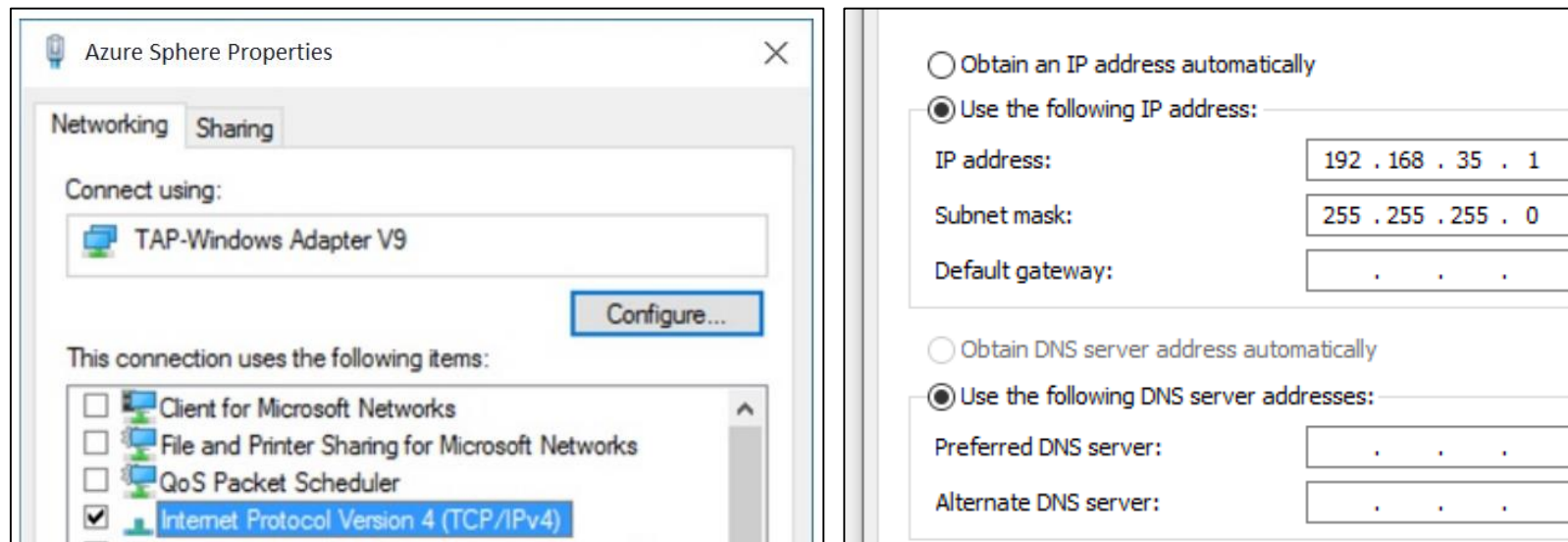
Reference : <https://docs.microsoft.com/en-us/azure-sphere/install/install#connect-the-azure-sphere-device>

Verify TAP-Windows Adapter Configuration

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



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



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

Set up an account for Azure Sphere

- Follow this web site to set up an account for Azure Sphere
- <https://docs.microsoft.com/en-us/azure-sphere/install/azure-directory-account>
- You must create a user on **Azure Active Directory (AAD)** for Azure Sphere access control.

Create tenant and claim your device

- This document use Azure Sphere SDK 19.05
- Sign in to Azure Sphere, using your work or school account:
 - **azsphere login**
- Type the following command to create a tenant.
 - **azsphere tenant create --name <my-tenant>**
 - Example : **azsphere tenant create --name "USITenant"**
- Claim your device.
 - **azsphere device claim**
 - You should see output like this:
Claiming attached device ID
'ABCDE082513B529C45098884F882B2CA6D832587CAAE1A90B1CEC4A376EA2F22A96C4
E7E1FC4D2AFF5633B68DB68FF4420A5588B420851EE4F3F1A7DC51399ED' 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> --key <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 prep-debug**
- 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.

Create Azure IoT Central

Create an Azure IoT Central application

- Go to [Azure IoT Central](#) in your browser and sign in with your Azure credentials.
- On the Create Application page, choose the Trial payment plan and the Custom Application template. Accept the default values for the Application Name and URL, enter contact information, and then click **Create** at the bottom of the page.
- On the Home Page, select Create Device Templates.



Create Device Templates

Device templates are blue-prints that describe your devices.




Create an Azure IoT Central application

- On the **Device Templates > New Template > Custom Template** page, enter AzureSphere as the device name and click **Create**..

Device Templates > New Template > Custom Template

You don't need a real device to get started, a simulated device is created with this template.

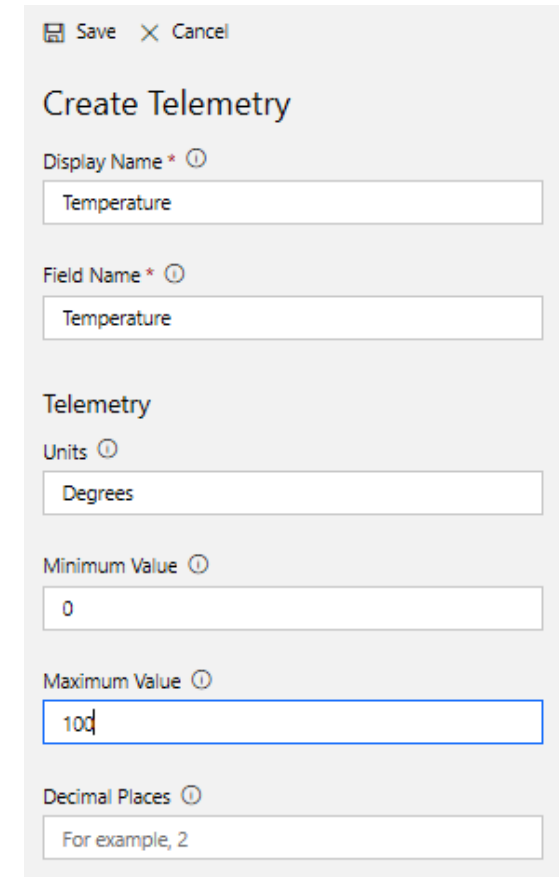


Enter the name for your device template *

Create

Create an Azure IoT Central application

- On the **Device Template** page, click **New Measurement** and select **Telemetry**.
- Set the Display Name and Field Name to **Temperature**. The Field Name must exactly match the name in the sample code, so this value is case-sensitive. Set Units to **Degrees**, and set the Minimum Value and Maximum Value to **0** and **100**, respectively. Then click **Save**.



Save Cancel

Create Telemetry

Display Name * ⓘ
Temperature

Field Name * ⓘ
Temperature

Telemetry

Units ⓘ
Degrees

Minimum Value ⓘ
0

Maximum Value ⓘ
100

Decimal Places ⓘ
For example, 2

Create an Azure IoT Central application



- On the **Device Template** page, click **Property** and select **Device Property**.
- Set the Display Name to **Device To Cloud** and Field Name to **sendToCloud**. The Field Name must exactly match the name in the sample code, so this value is case-sensitive. Select Data Type to **text**. Then click **Save**.

Save Cancel

Configure Device Property

Display Name * ⓘ

Device To Cloud

Field Name * ⓘ

sendToCloud

Data Type * ⓘ

text

Description ⓘ

Example: Set display text on the screen.

* Required

Create an Azure IoT Central application



- On the **Device Template** page, click **Settings** and select **Text**.
- Set the Display Name to **Cloud To Device** and Field Name to **sendToDevice**. The Field Name must exactly match the name in the sample code, so this value is case-sensitive. Set Initial Value to **null**. Then click **Save**.

Save Cancel

Configure Text

Display Name * ⓘ

Cloud To Device

Field Name * ⓘ

sendToDevice

Trim Leading Spaces ⓘ

☐ Off

Trim Trailing Spaces ⓘ

☐ Off

Case Sensitivity In Comparison ⓘ

☐ Off

Case Sensitivity In Data Entry ⓘ

mixed

Minimum Length ⓘ

0

Maximum Length ⓘ



100

Initial Value * ⓘ

null

Create an Azure IoT Central application

- On the **Device Template** page, click **Settings** and select **Toggle**.
- Set the Display Name to **WiFi LED BLUE** and Field Name to **StatusLED**. The Field Name must exactly match the name in the sample code, so this value is case-sensitive. Set ON Display Text to **ON**. Set OFF Display text to **OFF**. Set Initial Value to **OFF** Then click **Save**.

 Save
  Cancel

Configure Toggle

Display Name * ⓘ

Field Name * ⓘ

ON Display Text ⓘ

OFF Display text ⓘ

Initial Value * ⓘ

☒ On
 ☐ Off

Description ⓘ

Get IoT Central Connection Info

- On the **Administration** page, click **Device connection**.
 - Check **ID Scope**
- Click **Shared access signature (SAS) => View Keys**
 - Check **Primary Key**
- In an Azure Sphere Developer Command Prompt, type the following command to get Device ID.
 - Note: The Azure IoT Central requires that the device ID be in lowercase characters. You can use the ToLower function in PowerShell to convert, if necessary. In an Azure Sphere Developer Command Prompt, enter the following command, which gets the ID of the attached device and converts it to lowercase:
 - **powershell -Command ((azsphere device show-attached)[0] -split ' ')[1].ToLower()**

Azure Sphere Guardian Example

Setup Azure Sphere Guardian Example

- Download Example:
 - <https://github.com/USIWP1Module/USI-Guardian-Board>
 - Support on Azure Sphere SDK 19.05
- Run IoT Central Tools to get config data
 - Path : AzureIoTCentralTools\ShowIoTCentralConfig.exe
 - Entry account type, IoT Central Scope ID, Device ID,

```
D:\AzureIoTCentralTools>ShowIoTCentralConfig.exe
Tool to show Azure IoT Central configuration for Azure Sphere applications

Are you using a Work/School account to sign into your IoT Central Application (Y/N) ?n
Enter the Scope Id from the IoT Central App 'Administration | Device Connection' >One000ABCD1
Enter the Primary Saas Key from the IoT Central App 'Administration | Device Connection' >abcdefghijklmnpqrstuvwxyz12345
abcdefghijklmnpqrstuvwxyz12345
Enter any real Device Id from the IoT Central App >1234567890qwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghdsafdsafdsafdsadg
fdsafdsafdsafdsafsa

Find and modify the following lines in your app_manifest.json:
"CmdArgs": [ "One000ABCD1" ],
"AllowedConnections": [ "global.azure-devices-provisioning.net", "iotc-abcde1234-1234-5678-abcd-abc12345678.azure-device
s.net" ],
"DeviceAuthentication": "--- YOUR AZURE SPHERE TENANT ID--- ",

Obtain your Azure Sphere Tenant Id by opening an Azure Sphere Developer Command Prompt and typing the following command:
'Azsphere tenant show-selected'
```

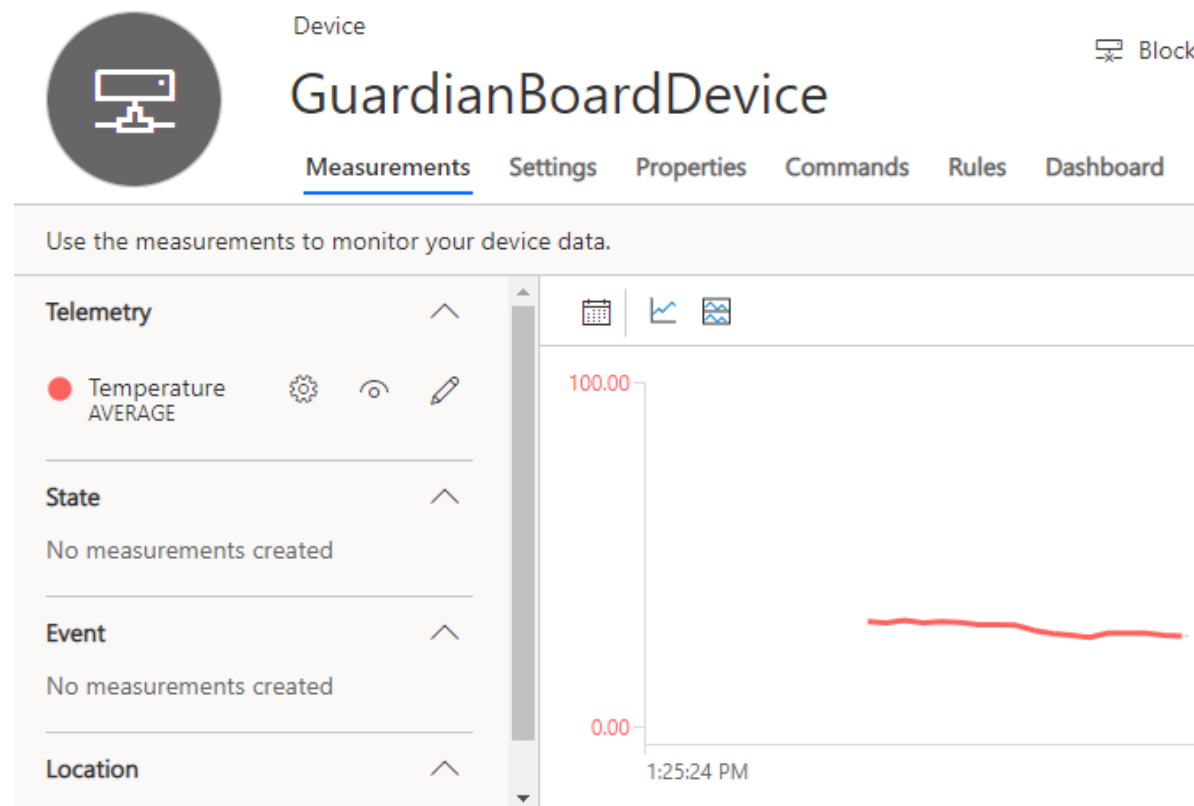
- Copy the information from the output into the app_manifest.json file in Visual Studio (MT3620App).

Build Azure Sphere Guardian Example

- In an Azure Sphere Developer Command Prompt, use `azsphere image-package pack-board-config` to create a board configuration image package for the Ethernet:
 - `azsphere image package-board-config --preset lan-enc28j60-isu0-int5 --output enc28j60-isu0-int5.imagepackage`
- Following command sideloads the Ethernet board configuration image package:
 - `azsphere device sideload deploy --imagepackage enc28j60-isu0-int5.imagepackage`
- Ensure that your device is connected to the internet.
- Go to the Build menu, and select Build All. Alternatively, open Solution Explorer and select Build. This will build the application and create an imagepackage file. The output location of the Azure Sphere application appears in the Output window.

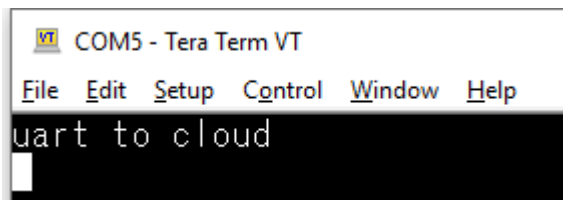
Run Azure Sphere Guardian Example

- In Azure IoT Central, go to **Device Templates** and select your template. Note that the device is sending simulated temperatures at regular intervals. You might need to refresh the window to see the data:

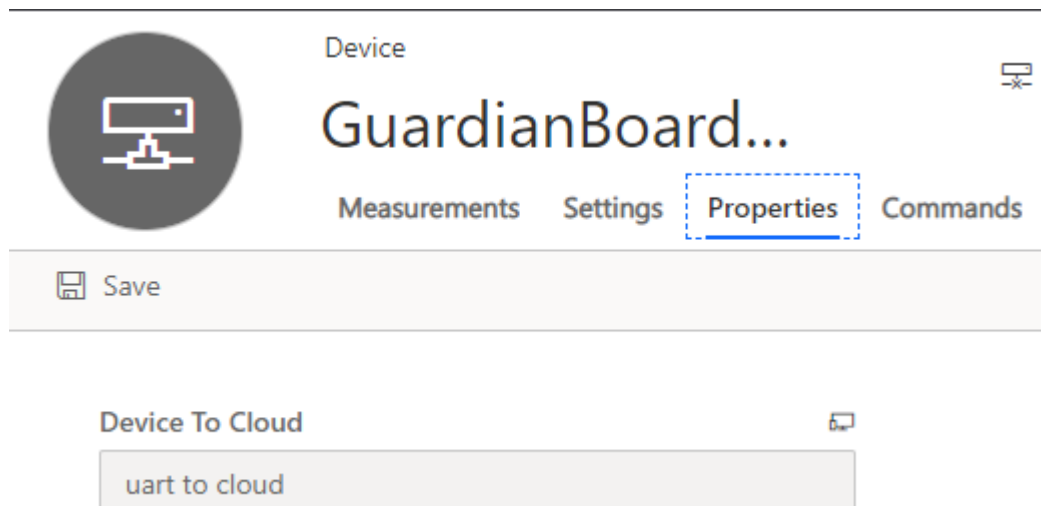


UART With IoT Central

- Send UART data from device to cloud
 - Use terminal emulator such as TeraTerm or putty to open minimum number of Azure Sphere com port (Azure Sphere have 3 com ports).
 - Type data on terminal

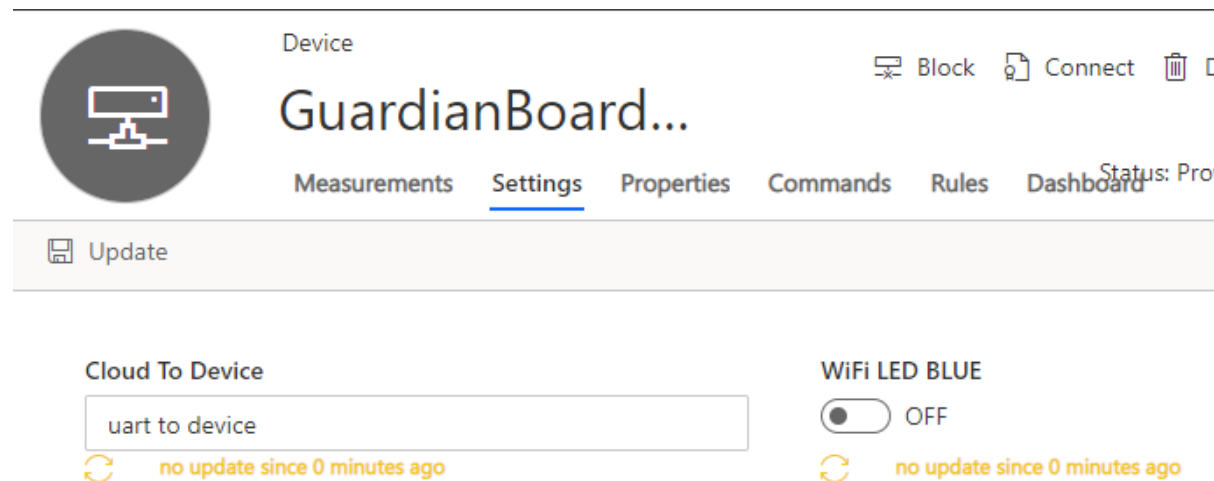


- In Azure IoT Central, go to Properties. You might need to refresh the window to see the data:

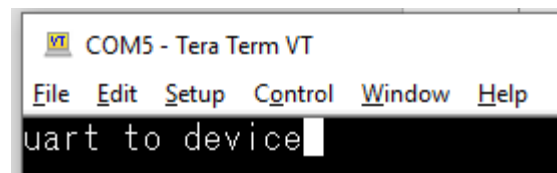


UART With IoT Central

- Receive UART data from cloud to device
 - In Azure IoT Central, go to Settings. Type data in **Cloud To Device** textbox and click **Update**.

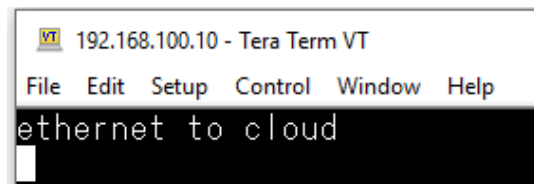


- Terminal will show received data

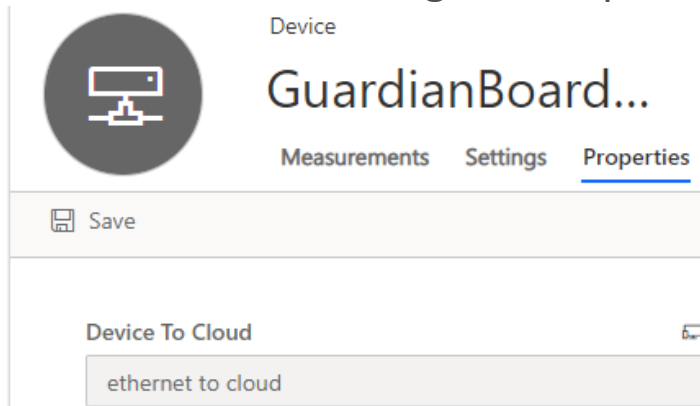


Ethernet With IoT Central

- Send private network data by Ethernet from device to cloud
 - Plugin an Ethernet cable from the Ethernet adapter on the device to the Ethernet connection on your computer. Open a command prompt on your computer and type ipconfig. You should see that the DHCP server has issued an IP address in the 192.168.100.xxx range to your PC for its network connection.
 - Use terminal emulator such as TeraTerm to a raw TCP connection to the Azure Sphere example's TCP server at 192.168.100.10 port 11000.
 - Type data on terminal

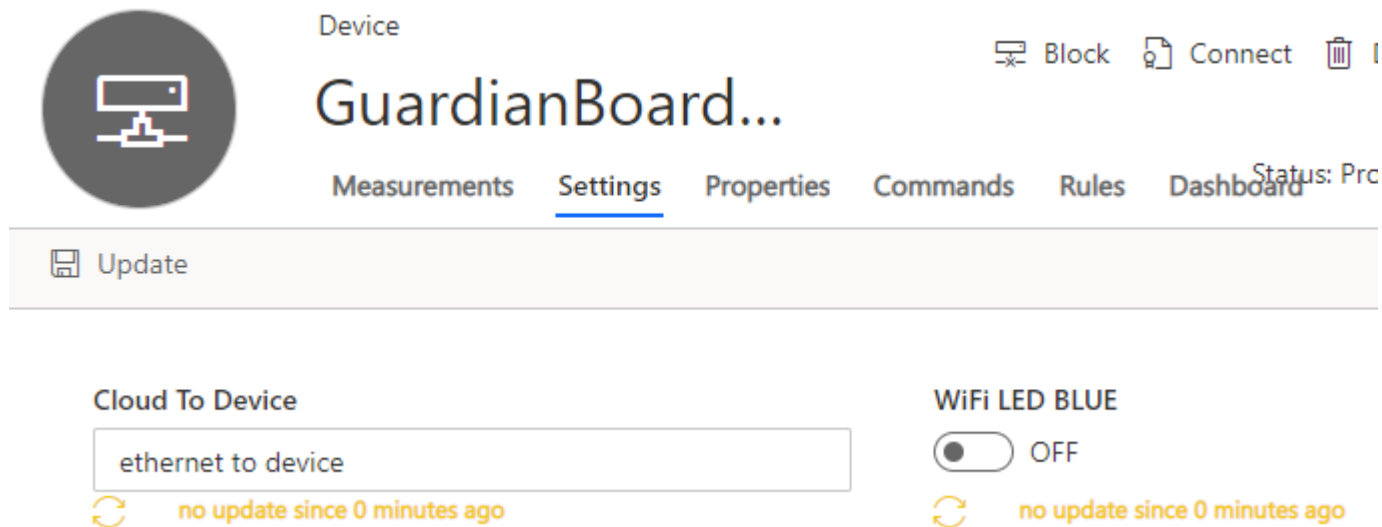


- In Azure IoT Central, go to Properties. You might need to refresh the window to see the data:

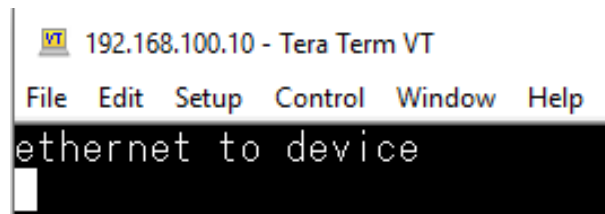


Ethernet With IoT Central

- Receive private network data by Ethernet from cloud to device
 - In Azure IoT Central, go to Settings. Type data in **Cloud To Device** textbox and click **Update**.

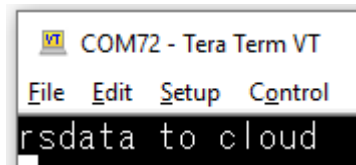


- Terminal will show received data

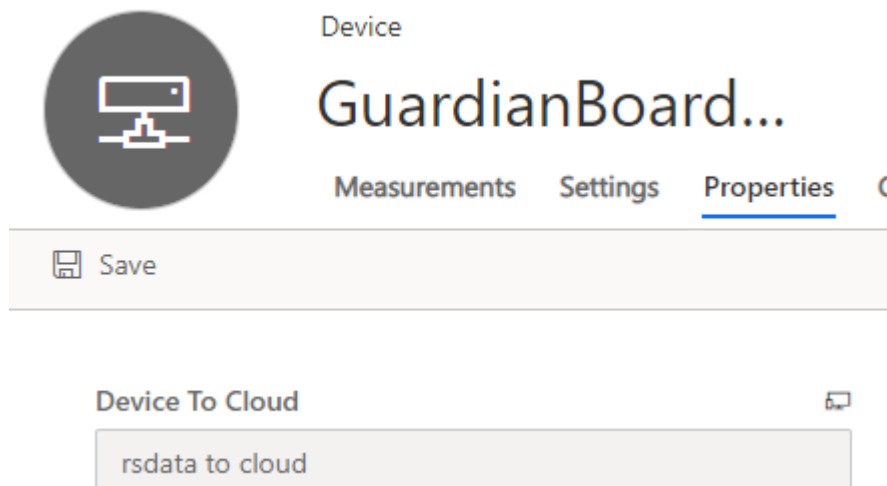


RS485 With IoT Central

- Send RS485 data from device to cloud
 - Plugin RS485 cable to Azure Sphere Guardian Board RS485 adapter and PC.
 - Use terminal emulator such as TeraTerm or putty to open RS485 com port.
 - Type data on terminal

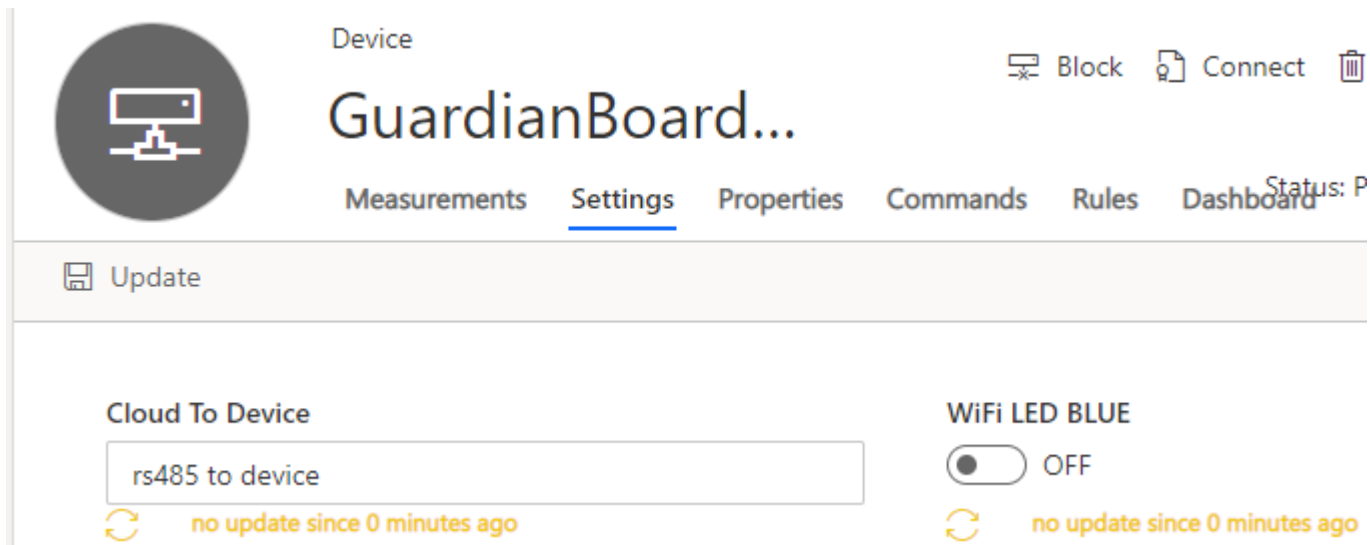


- In Azure IoT Central, go to Properties. You might need to refresh the window to see the data:

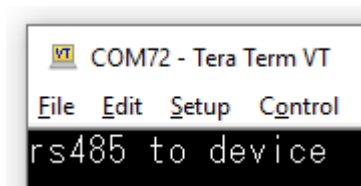


RS485 With IoT Central

- Receive RS485 data from cloud to device
 - In Azure IoT Central, go to Settings. Type data in **Cloud To Device** textbox and click **Update**.

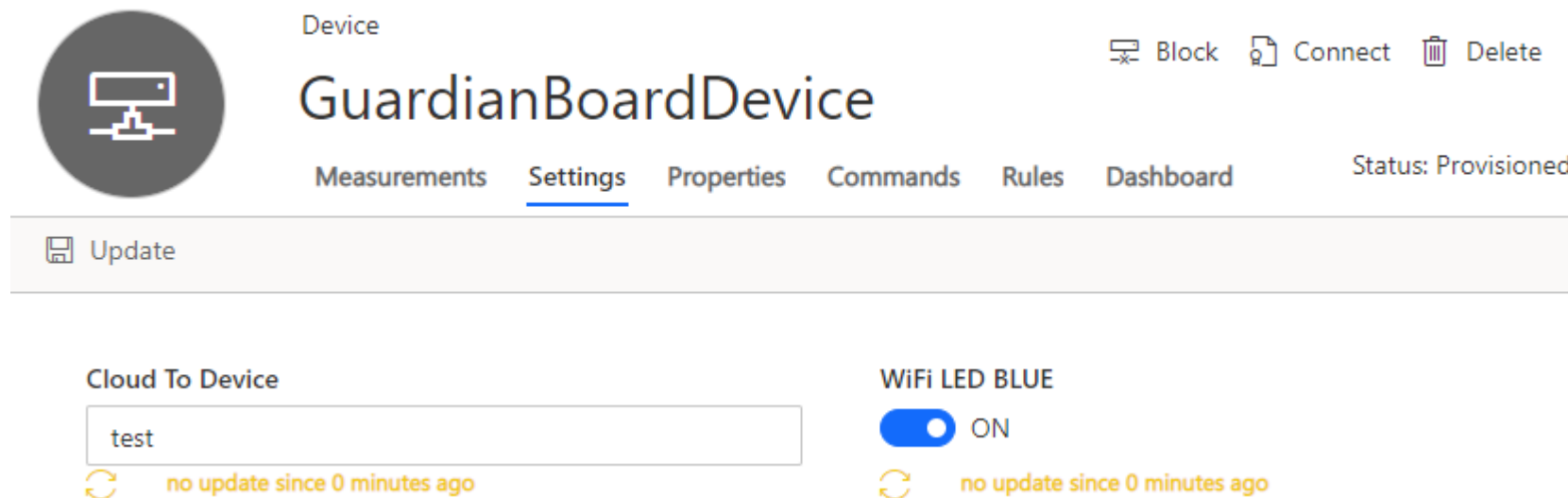


- Terminal will show received data



LED With IoT Central

- Control LED Status from IoT Central
 - In Azure IoT Central, go to Settings. Switch WiFi LED BLUE to **ON** and click **Update**.



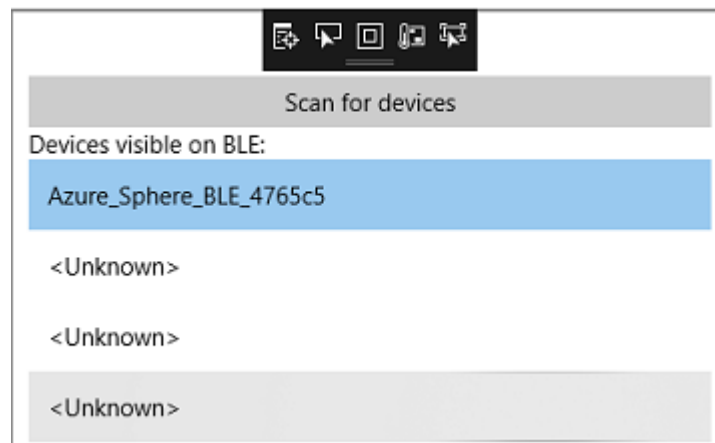
- The Azure Sphere Guardian Board WiFi LED BLUE will turn on.

Windows 10 BLE Companion App

- Run the Windows 10 BLE companion app on your PC
- Start a separate instance of Visual Studio.
- Open WindowsApp/WifiSetupAndDeviceControlViaBle.sln.
- Build and debug the application (F5). If this is your first time developing Universal Windows Platform (UWP) applications on this computer, you may be required to download the Universal Windows Platform Development workload.

Configure The Wi-Fi Settings By BLE

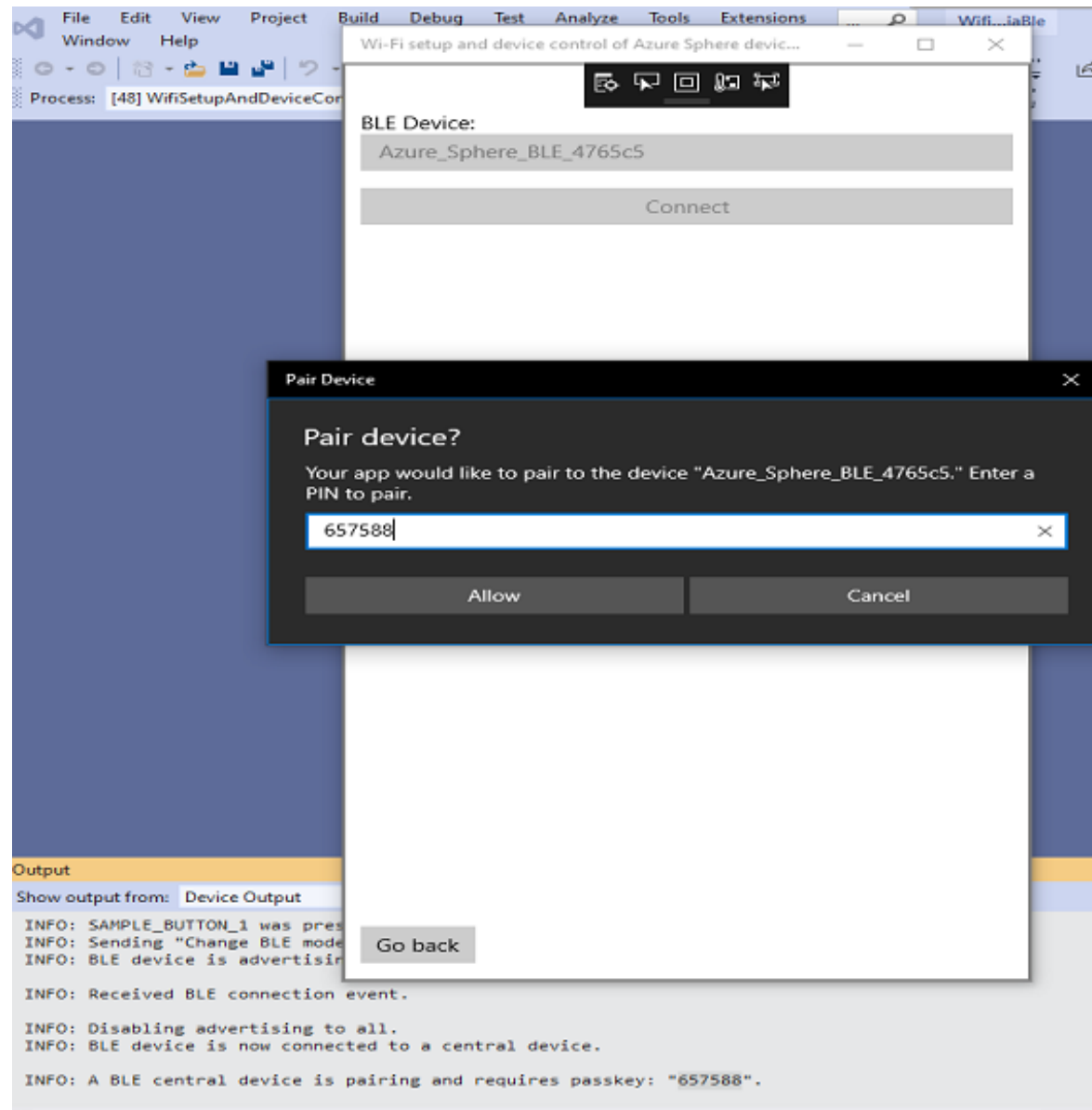
- Press button S2 on the MT3620 board. The Azure Sphere app requests that the BLE app advertise its availability to connect to a new BLE device for the next 60 seconds. LED APP on the MT3620 should light up red.
- In the Windows 10 companion app, click the **Scan for devices** button at the top to scan for BLE devices.
- Select your device from the list. It has the name you noted earlier in the Run the Azure Sphere app step.



Configure The Wi-Fi Settings By BLE

- Click **Connect**. LED APP on the MT3620 lights up green to indicate a BLE device (the Windows 10 computer) is connected to the BLE App, and a passkey appears in the Output window in the Visual Studio instance for the Azure Sphere app. The BLE App doesn't yet trust this BLE device.
- When prompted, type the passkey you noted earlier. When the connection is confirmed, the nRF52 app will remember the Windows 10 computer as a known and trusted BLE device and allow it to read and write values over BLE. The Windows 10 companion app will now be able to read and display the current Wi-Fi status on the MT3620.

Configure The Wi-Fi Settings By BLE

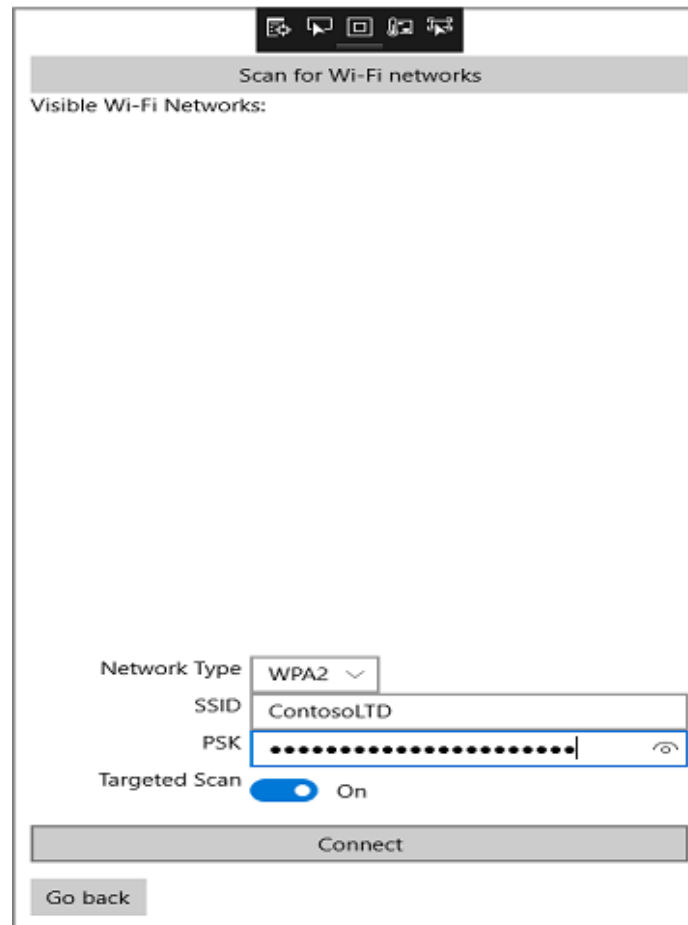


Configure The Wi-Fi Settings By BLE

- If there is no active Wi-Fi network, click **Add new network**.... If an active network is present, press and hold button B on the MT3620 dev board for at least three seconds to delete it.
- Click **Scan for Wi-Fi networks**. It may take a few seconds to display a complete list of networks that the Azure Sphere device can see. Only open and WPA2 networks are supported.

Configure The Wi-Fi Settings By BLE

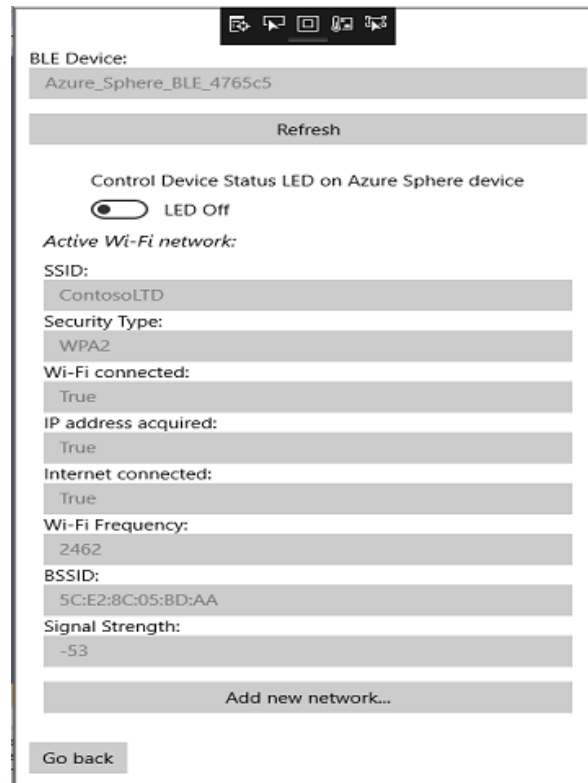
- If you are connecting to an open network, simply click **Connect**. If the network is secured, a prompt appears for a network password. Enter the password and then click **Connect**.



The screenshot displays a mobile application interface for configuring Wi-Fi settings. At the top, there is a navigation bar with several icons. Below it, a header reads "Scan for Wi-Fi networks". Underneath, the text "Visible Wi-Fi Networks:" is followed by a large empty space, presumably for a list of detected networks. At the bottom of the screen, there are configuration options: "Network Type" is set to "WPA2" with a dropdown arrow; "SSID" is "ContosoLTD"; "PSK" is a field filled with dots, with a toggle icon to its right; and "Targeted Scan" is a toggle switch currently turned "On". A large "Connect" button is positioned below these settings, and a "Go back" button is at the very bottom left.

Configure The Wi-Fi Settings By BLE

- Observe that the current Wi-Fi status is displayed again and refreshed every 5 seconds. You should see the Azure Sphere device connect to the new Wi-Fi network successfully. If you are running the Azure Sphere and Windows apps in debug mode in Visual Studio, the Output window should show the protocol communications they are sending and receiving.



BLE Device:
Azure_Sphere_BLE_4765c5

Refresh

Control Device Status LED on Azure Sphere device
☐ LED Off

Active Wi-Fi network:

SSID:
ContosoLTD

Security Type:
WPA2

Wi-Fi connected:
True

IP address acquired:
True

Internet connected:
True

Wi-Fi Frequency:
2462

BSSID:
5C:E2:8C:05:BD:AA

Signal Strength:
-53

Add new network...

Go back

Reference

- Azure Sphere Documentation
 - <https://docs.microsoft.com/en-us/azure-sphere/>
- Azure Sphere GitHub
 - <https://github.com/Azure/azure-sphere-samples>

Thank You
www.usiglobal.com