Azure Sphere Guardian-100 Hardware User Guide

v1.0 – October 25, 2019

1 Document Control

Document Version:	v1.0
Document Date:	10/25/2019
Document Author:	Peter Fenn
Document Classification:	Public
Document Distribution:	Public

2 Version History

Version	Date	Comment
1.0	10/25/2019	Initial draft release

Contents

Figures

3 Hardware Checklist

Hardware items recommended for application development for Guardian-100 are the following

#	Item Description
1	Development Computer with Windows-10 Operating System
2	Avnet Azure Sphere Guardian-100 Secure Edge Module (plus provided cables) http://avnet.me/mt3620-guardian

4 Software Checklist

Listed below are the software items mentioned in this document

#	**Item Description**
1	**Visual Studio 2019** (Enterprise, Professional or Community edition) downloadable from: https://visualstudio.microsoft.com/
2	**Microsoft Azure Sphere SDK for Visual Studio Preview** (Windows console application) downloadable from: http://aka.ms/AzureSphereSDK Azure_Sphere_SDK_Preview_for_Visual_Studio.exe
3	iPerf3 Server Application (Windows console application) iperf-3.1.3-win64.zip
4	Module: iPerf3 Test application (production-signed) iperf3_ps.imagepackage
5	Module: guardian_test application (production-signed) guardian_test_ps.imagepackage

5 Introduction

The **Azure Sphere Guardian-100** is a **Secure Edge Module** that connects existing equipment (via Ethernet or USB UART) through a Microsoft Azure Sphere secured wireless connection to the cloud.

Guardian-100 is designed-around Avnet's globally certified Azure Sphere MT3620 Module, which is based on the MT3620 multi-core dual band Wi-Fi SoC. The MT3620 is the first Azure Sphere certified "microcontroller", a completely new class of connected SoC IoT device that features "end-to-end security". User applications can target it's 500 MHz ARM Cortex-A7 core as well as two general purpose 200 MHz ARM Cortex-M4F I/O subsystem cores designed to support real-time requirements. The on-chip peripherals (GPIO, UART, I2C, SPI, I2S, PWM and ADC) can be mapped to any of these three user-accessible cores.

Additional differentiators of the MT3620 device are the built-in Pluton security subsystem (with dedicated Arm Cortex-M4F core) for secure boot and secure system operation, its dual-band 802.11 a/b/g/n Wi-Fi connectivity, as well as integration of on-chip PMU, RTC plus FLASH and SRAM memory. Wi-Fi based OTA firmware and user application updates (using strict certificate-based authentication) are hosted by Microsoft for the lifetime of the MT3620 device

The Arm Cortex-A7 application processor runs Microsoft's Azure Sphere Secure OS. Custom user applications are developed in C using Microsoft Visual Studio IDE, which includes debugging features like single-step execution, breakpoints and watch-points (supported via dedicated Azure Sphere service UART)

Online authentication and firmware updates are supported for the MT3620 device lifetime.

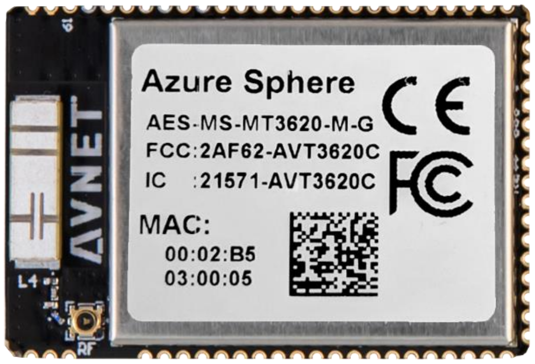


Figure 1 – Avnet Azure Sphere MT3620 Module (Chip Antenna version)

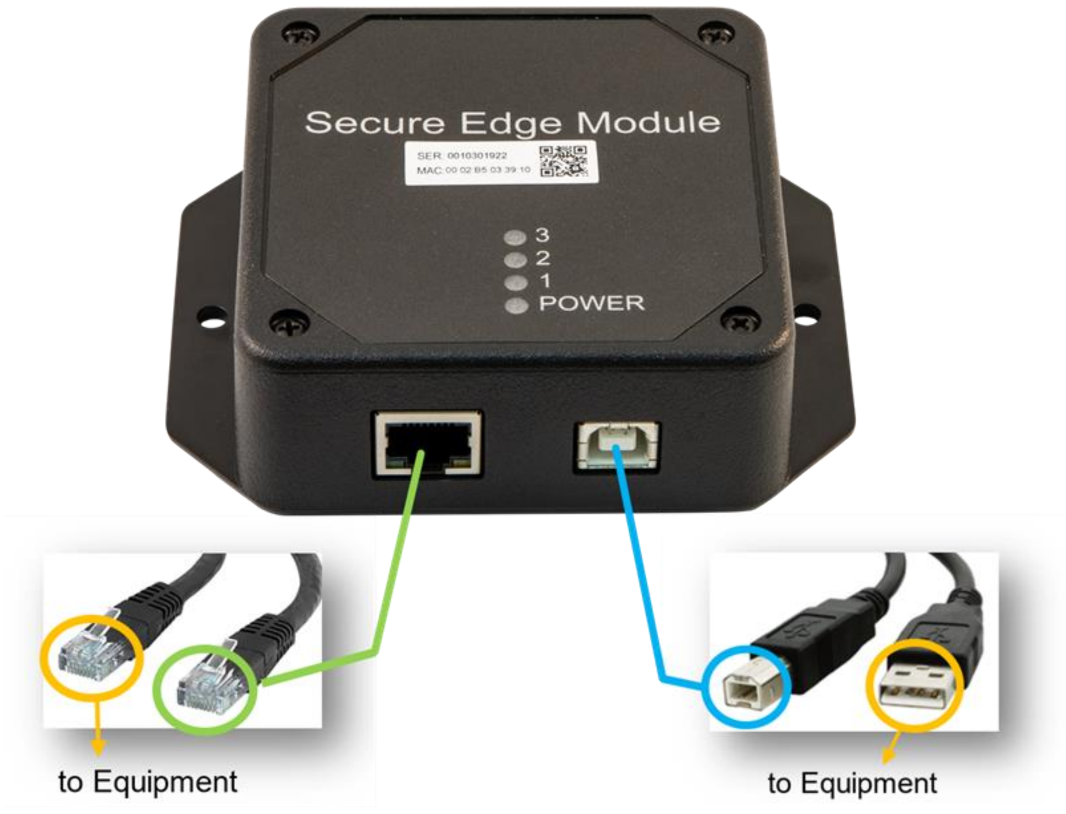


Figure 2 – Avnet Azure Sphere Guardian-100

5.1 Azure Sphere Guardian-100 Info

- Part Number: AES-MS-MT3620-GUARD-100
- Product Page URL: http://avnet.me/mt3620-guardian

5.2 Items Included with Guardian-100

- Azure Sphere Guardian-100 Secure Edge Module
- Azure Sphere Guardian-100 QuickStart Card
- USB 2.0 type-A to type-B cable
- Ethernet Cat-5 cable (RJ45 connectors)
- Access to downloadable reference designs and documentation

5.3 Important Reference Documents

- Azure Sphere Guardian-100 QuickStart Card
- Azure Sphere Guardian-100 Product Brief
- Azure Sphere Guardian-100 Hardware User Guide
- Azure Sphere Guardian-100 Schematic
- Azure Sphere Guardian-100 3D Mechanical Assembly
- Azure Sphere MT3620 Module Product Brief
- Azure Sphere MT3620 Module Datasheet & Integration Guide
- MediatTek MT3620 Product Brief Nov2018
- Microsoft Azure Sphere Installation Instructions
- Microsoft Azure Sphere Detailed Documentation

6 Guardian-100 Architecture and Features

6.1 List of Features

Azure Sphere Guardian-100

- Uses Avnet Azure Sphere Module based on Mediatek MT3620AN SoC that features:
  - 1x 500MHz ARM Cortex A7, 4MB SRAM
  - 2x 200MHz ARM Cortex M4F cores, 64KB SRAM
  - On-chip 16 MB QSPI Flash Memory (2x 8MB dual-channel QSPI Flash memory)
  - Dual-band 2.4/5GHz 802.11 a/b/g/n Wi-Fi
  - Dual-band 2.4/5GHz Chip Antenna (Pulse W3006)

- USB to serial FTDI 4-port Program/Debug interface (internal microUSB connector)
  - MicroUSB interface connector (requires lid removal)
  - Supports development computer Debug, Service, Recovery UARTs and M4 SWD interfaces
  - FT4232HQ 4-port FTDI device, buffers and USB activity LED

- Ethernet 10BaseT Interface, RJ45 connector and Magnetics (uses ISU0)

- USB 2.0 Device Interface and USB power. USB Type-B connector (uses ISU1)

- 6x External LEDs
  - Power, User 1, User-2, User-3
  - Ethernet Connection, Ethernet Activity

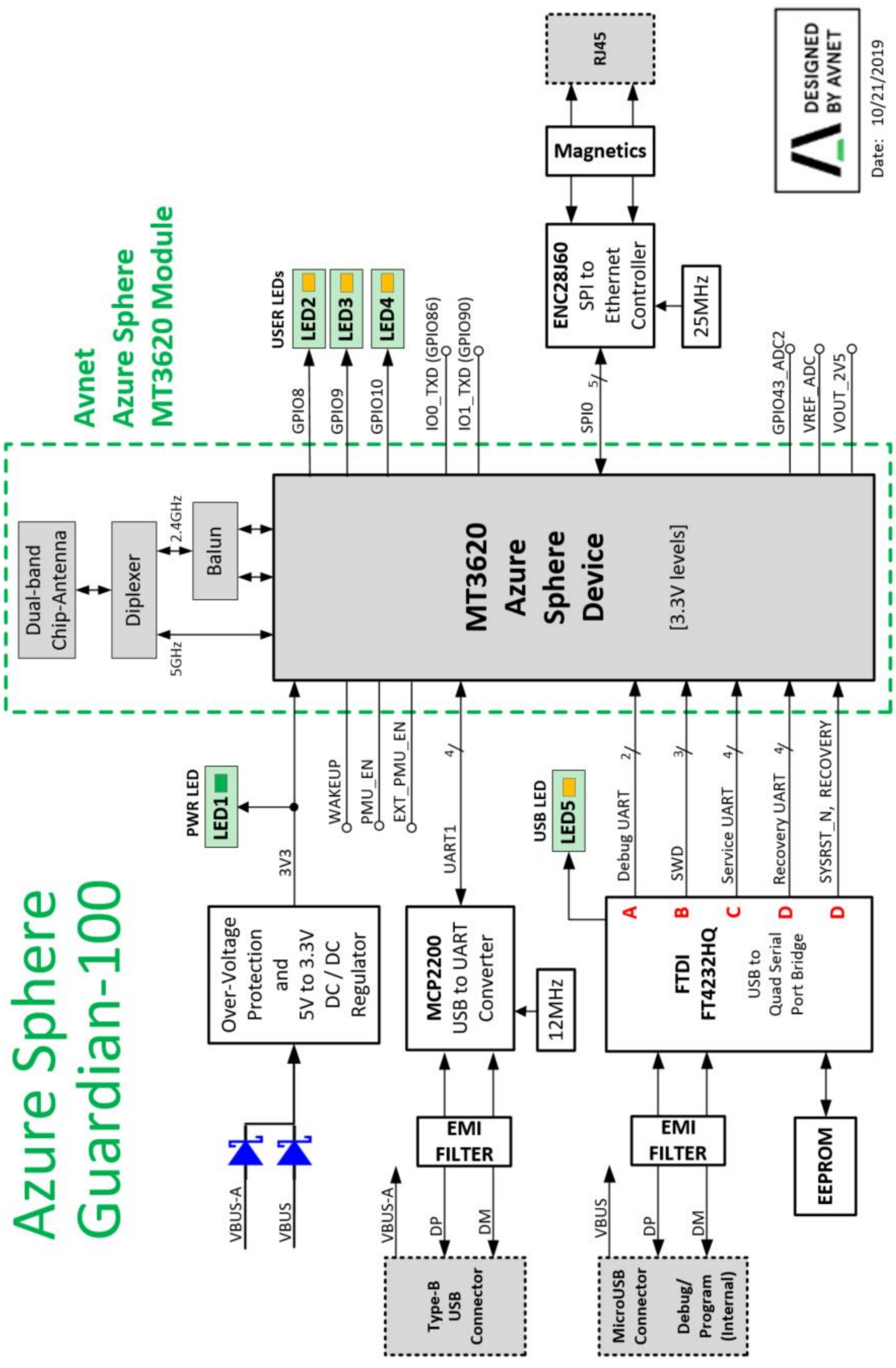
- 5V to 3.3V DC/DC Power Regulation (2A max, with over voltage protection)

- Operating Temperature: -30 ~ 85°C

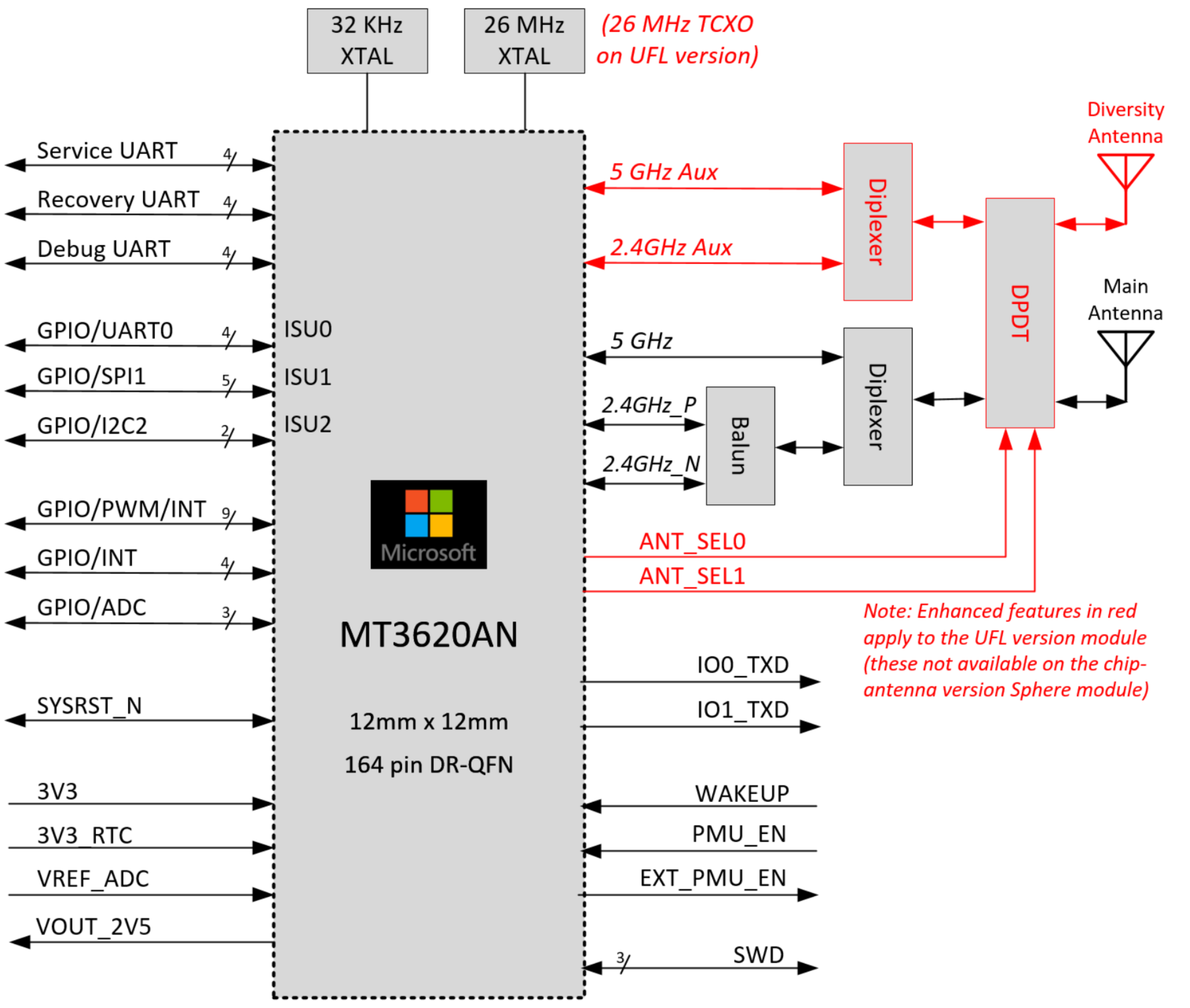
- Dimensions: 108mm x 85mm x 32mm (including mounting flanges)

- Module wireless certifications include FCC, IC, CE, (MIC, Anatel and RCM are pending)

6.2 Block Diagram - Avnet Azure Sphere Guardian-100



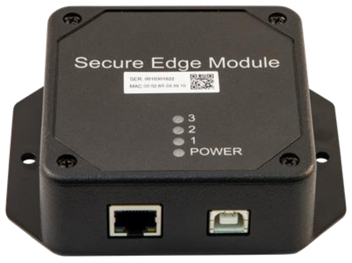
6.3 Block Diagram - Azure Sphere MT3620 Module



Note: Avnet’s Sphere Guardian-100 is fitted with the “chip antenna” version Azure Sphere MT3620 module (ie. sections of diagram colored red are <u>not</u> applicable)

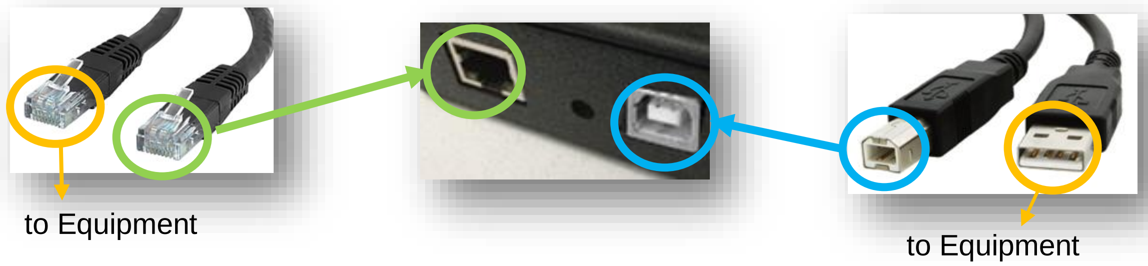
7 Guardian-100 Installation Overview

Ethernet Cable



Guardian Module

USB A-B Cable



7.1 Guardian-100 Connections

An overview of how Guardian interfaces to the equipment that it monitors is as follows:

1. For equipment with an Ethernet interface, connect the provided Ethernet cable from Guardian to the equipment.
2. Connect the USB cable from Guardian (type-B panel connector) to the equipment (this provides power to Guardian, and is also the data interface in cases where the monitored equipment does not have an Ethernet interface).
3. Once connected, the LEDs on the Guardian Module should be as follows:

Status LEDs	Color	Description
Power	Green	Confirmation that 3.3V supply rail voltage is OK
1, 2, 3	Amber	Controlled by the Azure Sphere user application (Factory test application sequences these LEDs)
RJ45 LEDA	Green	Green when connected
RJ45 LEDB	Yellow	Flashes intermittently with Ethernet activity

Note
The factory test application has no utility beyond LED confirmation that Guardian is operational.

It is essential that a suitable application be programmed into the Azure Sphere device in order for Guardian to perform the relevant cloud-connected equipment management functions

In order to reprogram Guardian, the latest Azure Sphere SDK and Microsoft Visual Studio IDE need to be installed on a development computer. The next section detail these requirements.

8 Software Development Environment Preparation

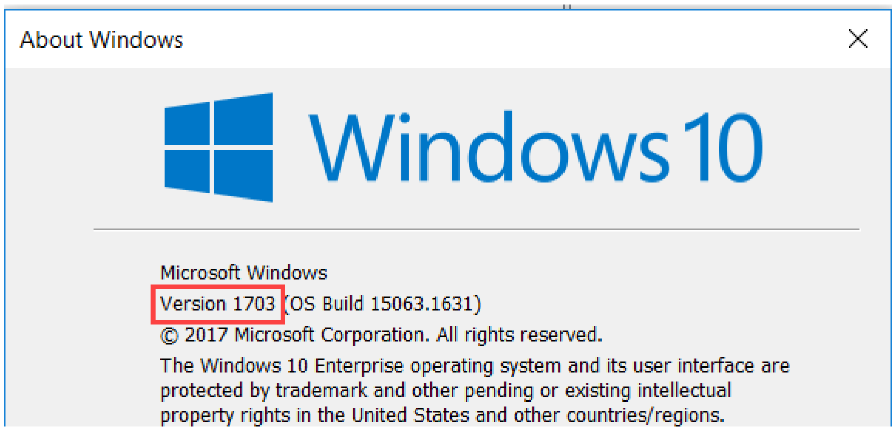
8.1 Microsoft Installation Instructions

Detailed guidance is provided at: https://docs.microsoft.com/en-us/azure-sphere/install/install

8.2 Verify Windows 10 Version

1) Before commencing software installation, verify the version of Windows 10 Operating System meets requirements. In the Windows search box (**Windows key + R**), enter **winver** to check…

2) The version reported must be **1607** or later…
https://en.wikipedia.org/wiki/Windows_10_version_history



Microsoft Windows
Version 1703 (OS Build 15063.1631)
© 2017 Microsoft Corporation. All rights reserved.
The Windows 10 Enterprise operating system and its user interface are protected by trademark and other pending or existing intellectual property rights in the United States and other countries/regions.

8.3 Install Azure Sphere SDK

1) Download and unzip the latest Microsoft Azure Sphere SDK from:
http://aka.ms/AzureSphereSDK

2) Install this SDK on a Windows 10 computer, using the instructions located at:
http://avnet.me/ms_sphere_docs

3) Once installed, launch the application and at the Azsphere command prompt, enter this command to confirm the Sphere SDK version:
```
azsphere show-version
```
The version reported should be **19.09** or later
(Note: Connection to Sphere Guardian-100 hardware is *not* required for this test)

8.4 Debug/Programmer FTDI USB Interface Access

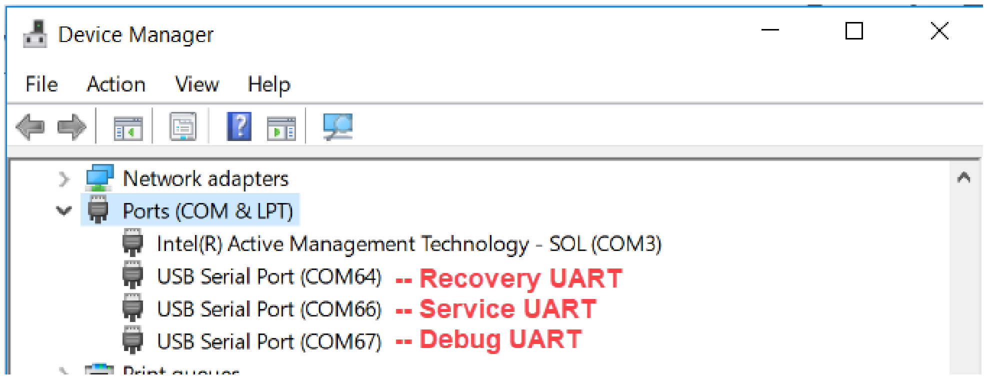
1) Disconnect all cables from Guardian, unscrew the four screws and remove the enclosure lid
2) Plug-in a USB cable (type-A to MicroUSB) from the vertically oriented MicroUSB connector on the Guardian PCB, to a spare USB port on the Windows 10 development computer

8.4.1 Windows FTDI USB Driver Installation

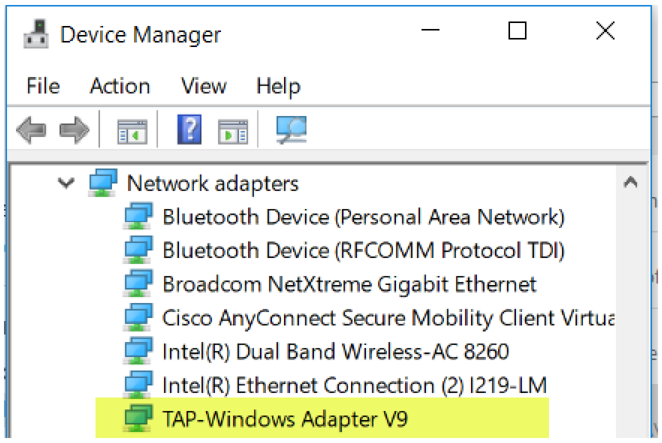
1) On first-time connection of Guardian to the development computer, the USB drivers should automatically download and install (this can be slow). If drivers do not install automatically, right-click on the device name in Windows Device Manager and select **Update driver**.
Alternatively, download the drivers from Future Technology Devices International (FTDI), - choose the driver that matches your Windows 10 installation (32- or 64-bit).
Additional assistance on this aspect is available at:
https://docs.microsoft.com/en-us/azure-sphere/install/install#connect-the-Secure Edge Module

8.4.2 Windows FTDI Interface Verification

2) Open Windows Device Manager and confirm the following are listed:
- three new **COM ports** (under Ports COM & LPT)
- a **TAP-Windows Adaptor V9** (under Network Adaptors)



Network adapters
Ports (COM & LPT)
Intel(R) Active Management Technology - SOL (COM3)
USB Serial Port (COM64) -- Recovery UART
USB Serial Port (COM66) -- Service UART
USB Serial Port (COM67) -- Debug UART

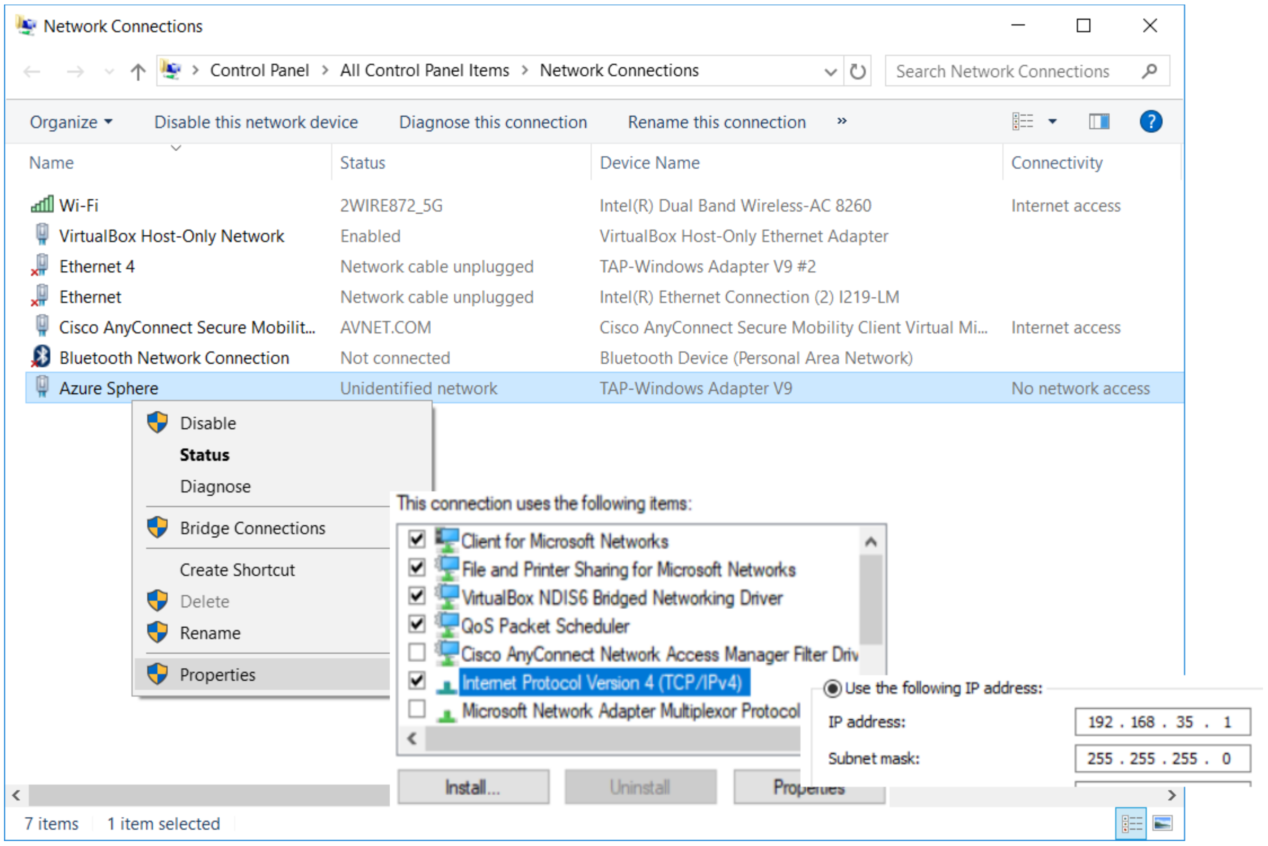


Network adapters
Bluetooth Device (Personal Area Network)
Bluetooth Device (RFCOMM Protocol TDI)
Broadcom NetXtreme Gigabit Ethernet
Cisco AnyConnect Secure Mobility Client Virtua
Intel(R) Dual Band Wireless-AC 8260
Intel(R) Ethernet Connection (2) I219-LM
TAP-Windows Adapter V9

3) The steps on this page are **only required** if the FTDI **SERVICE interface fails** during first-time connection to the Azure Sphere Guardian-100

4) Open Windows network adapter settings
ie. Windows search box (**Windows key + R**) then enter **ncpa.cpl**

5) Right-click on **Azure Sphere** TAP-Windows Adapter V9.
Check it's properties are as shown below:



Name	Status	Device Name	Connectivity
Wi-Fi	2WIRE872_5G	Intel(R) Dual Band Wireless-AC 8260	Internet access
VirtualBox Host-Only Network	Enabled	VirtualBox Host-Only Ethernet Adapter	
Ethernet 4	Network cable unplugged	TAP-Windows Adapter V9 #2	
Ethernet	Network cable unplugged	Intel(R) Ethernet Connection (2) I219-LM	
Cisco AnyConnect Secure Mobilit...	AVNET.COM	Cisco AnyConnect Secure Mobility Client Virtual Mi...	Internet access
Bluetooth Network Connection	Not connected	Bluetooth Device (Personal Area Network)	
Azure Sphere	Unidentified network	TAP-Windows Adapter V9	No network access

This connection uses the following items:

- [x] Client for Microsoft Networks
- [x] File and Printer Sharing for Microsoft Networks
- [x] VirtualBox NDIS6 Bridged Networking Driver
- [x] QoS Packet Scheduler
- [ ] Cisco AnyConnect Network Access Manager Filter Driv
- [x] Internet Protocol Version 4 (TCP/IPv4)
- [ ] Microsoft Network Adapter Multiplexor Protocol

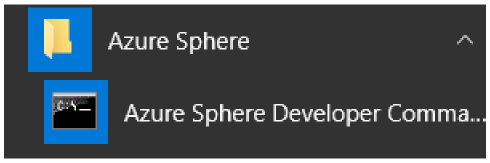
● Use the following IP address:

IP address: 192 . 168 . 35 . 1

Subnet mask: 255 . 255 . 255 . 0

8.5 SERVICE interface

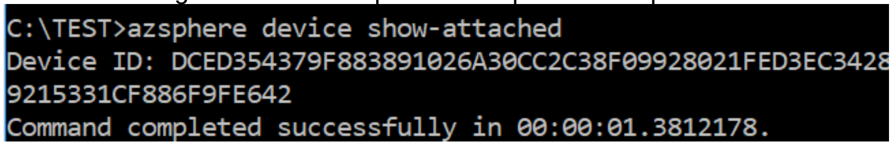
1) Open the Azure Sphere Developer Command-Line tool… (Sphere CLI)



2) Plug-in the USB cable from Guardian-100 to the PC, then enter the following *Sphere CLI* command:

```
azsphere device show-attached
```

3) The Secure Edge Module will report it's unique Azure Sphere Device ID…



```
C:\TEST>azsphere device show-attached
Device ID: DCED354379F883891026A30CC2C38F09928021FED3EC3428
9215331CF886F9FE642
Command completed successfully in 00:00:01.3812178.
```

8.6 DEBUG Interface

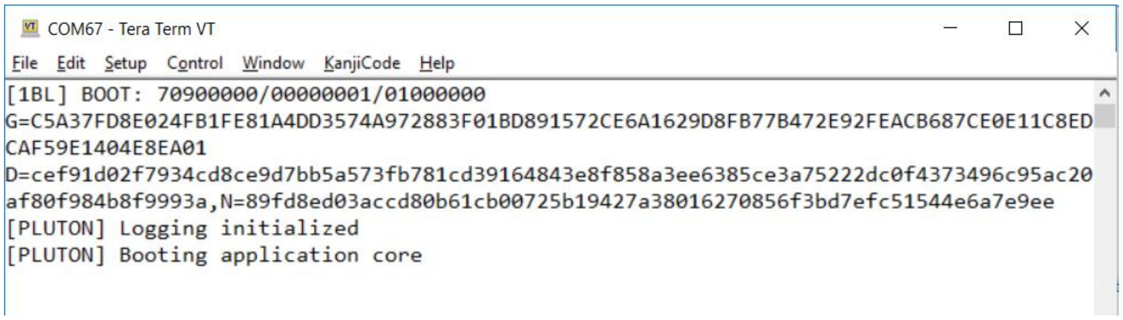
1) The DEBUG UART is typically the <u>highest numbered COM port</u> (of the three new COM ports) reported by Windows Device Manager, for the Guardian-100's FTDI USB interface

2) To view the output of this serial port, open Tera Term (or other serial console application) and configure it for the noted COM number, with UART set for 115200 8N1 communication rate

3) Connect the Tera Term terminal then enter the following *Sphere CLI* command:

```
azsphere device restart
```

4) Startup debug text similar to the following should appear on the terminal screen



```
[1BL] BOOT: 70900000/00000001/01000000
G=C5A37FD8E024FB1FE81A4DD3574A972883F01BD891572CE6A1629D8FB77B472E92FEACB687CE0E11C8ED
CAF59E1404E8EA01
D=cef91d02f7934cd8ce9d7bb5a573fb781cd39164843e8f858a3ee6385ce3a75222dc0f4373496c95ac20
af80f984b8f9993a,N=89fd8ed03accd80b61cb00725b19427a38016270856f3bd7efc51544e6a7e9ee
[PLUTON] Logging initialized
[PLUTON] Booting application core
```

5) <u>Note!</u> Terminal connection to the Debug Interface must be closed before attempting to use the RECOVERY interface!

8.7 RECOVERY Interface

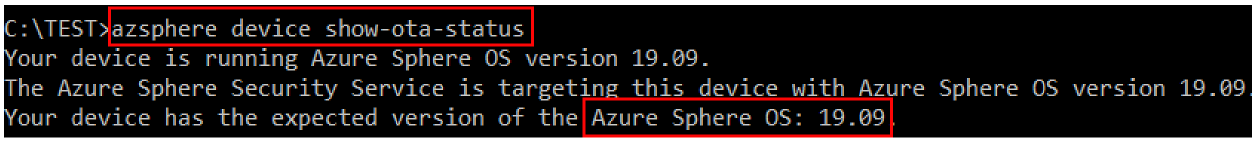
This interface is for reloading/updating the Azure Sphere OS via a wired UART interface (typically for factory reprogramming of the MT3620 device) and will not be required by most developers.

Once an Azure Sphere Guardian-100 is connected to the internet, Sphere OS updates are initiated automatically (or on demand) via the Over-The-Air (OTA) Wi-Fi interface

The Azure Sphere OS programmed into Guardian-100 by the manufacturer is out of date with the later reference designs provided. It is necessary to update the OS to version **19.09** (or later)

Check the current OS version by entering the following command at the SDK prompt:

```
azsphere device show-ota-status
```



```
C:\TEST>azsphere device show-ota-status
Your device is running Azure Sphere OS version 19.09.
The Azure Sphere Security Service is targeting this device with Azure Sphere OS version 19.09.
Your device has the expected version of the Azure Sphere OS: 19.09.
```

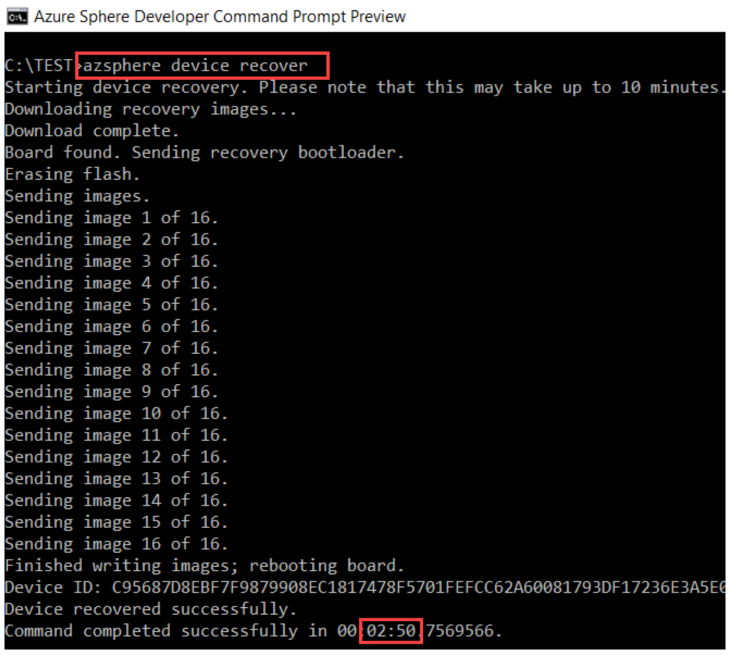
Two methods are available to update the OS on Guardian-100:
- via the development computer (must have internet connection and Azure Sphere SDK installed)
- via direct OTA update of the MT3620 device (requires configured device Wi-Fi settings)

The first method is recommended. Use the following command to download and program to the device to the latest available Azure Sphere OS version:

```
azsphere device recover
```

Note that all contents of flash memory get erased during RECOVERY (ie. Sphere OS, application software, Wi-Fi credentials and other configuration data)

This takes around ~3 minutes to complete



Azure Sphere Developer Command Prompt Preview

```
C:\TEST>azsphere device recover
Starting device recovery. Please note that this may take up to 10 minutes.
Downloading recovery images...
Download complete.
Board found. Sending recovery bootloader.
Erasing flash.
Sending images.
Sending image 1 of 16.
Sending image 2 of 16.
Sending image 3 of 16.
Sending image 4 of 16.
Sending image 5 of 16.
Sending image 6 of 16.
Sending image 7 of 16.
Sending image 8 of 16.
Sending image 9 of 16.
Sending image 10 of 16.
Sending image 11 of 16.
Sending image 12 of 16.
Sending image 13 of 16.
Sending image 14 of 16.
Sending image 15 of 16.
Sending image 16 of 16.
Finished writing images; rebooting board.
Device ID: C95687D8EBF7F9879908EC1817478F5701FEFCC62A60081793DF17236E3A5E
Device recovered successfully.
Command completed successfully in 00:02:50.7569566.
```

9 Configure Device Wi-Fi Network Settings

9.1 Scan for Wi-Fi Access Points

A quick-check of Wi-Fi reception can be done by entering the following *Sphere CLI* command:

```
azsphere device wifi scan
```

After 10 seconds or so, a scan report displays detected SSIDs, signal-levels, etc in the format shown below:

```
Scan results:

SSID                  : 2WIRE872_5G
Security state        : psk
BSSID                 : 2c:56:dc:d6:fc:84
Signal level          : -46
Frequency             : 5180

SSID                  : 2WIRE872
Security state        : psk
BSSID                 : 2c:56:dc:d6:fc:80
Signal level          : -27
Frequency             : 2457
...
```

9.2 Configuring the Wi-Fi Network Settings

Use the following command to configure the Wi-Fi settings
(replace **??????** with the applicable credentials)

```
azsphere device wifi add --ssid ?????? --psk ???????
```

or abbreviated to:

```
azsphere device wifi add -s ?????? -p ???????
```

Verify the present Wi-Fi connectivity status by entering:

```
azsphere device wifi show-status
```

Other useful Wi-Fi commands are:

```
azsphere device wifi list

azsphere device wifi enable

azsphere device wifi disable

azsphere device wifi forget -i0
```

Note: Appendix-B in this document includes instructions for running a pre-compiled copy of the **iPerf3** test application, to check Wi-Fi bit-rate performance with the currently selected Wi-Fi Access Point

10 Hardware Functional Description

10.1 Avnet Azure Sphere MT3620 module

The module pins-out a subset of the MT3620 SoC device functionality, via 66 castellated "stamp-hole" pads along three edges of it's compact 33mm x 22mm form-factor.

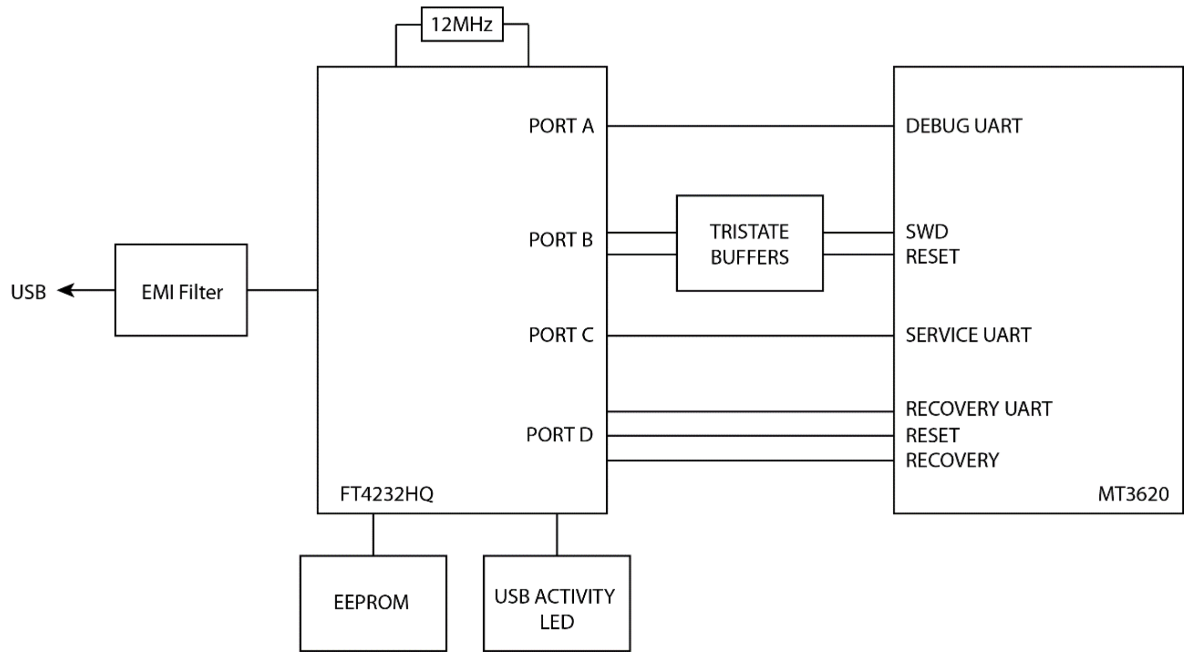
Refer to the following documents for detailed information on Avnet's certified Azure Sphere MT3620 module as well as the MT3620 Azure Sphere SoC device that this is based on

- Azure Sphere MT3620 Module Product Brief
- Azure Sphere MT3620 Module Datasheet & Integration Guide
- Media Tek MT3620 Product Brief Nov2018

10.2 USB FTDI Programmer Interface (FT4232HQ)

Guardian-100 includes on-board the Microsoft-specified FTDI 4-port USB to Serial bridge implementation of the RECOVERY, SERVICE, DEBUG and SWD interfaces. These interfaces are primarily used for software application development and/or programming of Guardian-100. This requires a USB cable to be connected from the internal microUSB connector, to a Windows-10 (or Linux) computer on which Azure Sphere SDK and the relevant USB drivers have been installed

A simplified block diagram of this 4-port USB to Serial bridge circuit is shown below:



See section of this document on Windows FTDI USB Driver Installation and Verification, for in-depth detail on driver installation and the use of these four interfaces

10.3 USB-UART Application Interface (MCP2200)

The onboard MicroChip MCP2200 USB-UART device provides a USB to Serial CDC type interface. (This device includes 64 byte transmit and 64 byte receive buffers, as well as 256 byte User EEPROM).

ISU1 is used for **UART1** interface signals to the MCP2200 device

Signal Name UART1 (ISU1)	**MT3620 GPIO #**	**Comments**
APP_TXD	GPIO31	ISU1
APP_RTS	GPIO32	ISU1
APP_RXD	GPIO33	ISU1
APP_CTS	GPIO34	ISU1
RST	3V3	

10.4 Ethernet Interface (ENC28J60)

The onboard MicroChip ENC28J60 ethernet controller device provides a 10 Mbps interface (compatible with 10/100/1000 Base-T networks) using TCP or UDP network protocols:

a) Private network, with network services (not connected to the internet), or
b) Public network, communicating with Azure IoT or other internet-based services.

Use of Ethernet requires a "*board configuration image*" in addition to the user application image. This contains info required by the Azure Sphere Security Monitor to add Ethernet support to Azure Sphere OS. **ISU1** is used for **SPI1** interface signals to the ENC286J60 device, with interrupts on **GPIO5**.

Signal Name SPI0 (ISU0)	**MT3620 GPIO #**	**Comments**
SCK_ENC	GPIO26	ISU0
MOSI_ENC	GPIO27	ISU0
MISO_ENC	GPIO28	ISU0
CS_ENC	GPIO29	ISU0
INT_ENC	GPIO5	
RST_N_ENC	GPIO6	Not connected
SYSRST_N	SYSRST_N	

More detail on this topic is available here:
https://docs.microsoft.com/en-us/azure-sphere/network/connect-ethernet

Microsoft provides sample application code for the Ethernet interface for a number of use cases, eg.

- The Private Network Services sample demonstrates how to connect Azure Sphere to a private network and use several network services.
- The AzureIoT sample demonstrates how to use the Azure IoT SDK C APIs in an Azure Sphere application to communicate with Azure IoT Central or Azure IoT Hub.
- The HTTPS samples demonstrate how to use the cURL APIs with Azure Sphere over a secure HTTPS connection.

Ethernet and Wi-Fi network interfaces can run simultaneously, connected to public (internet-connected) or private networks. At least one interface however, must be connected to a public network.

In Private Network applications, the high-level application configures Azure Sphere OS managed DHCP and SNTP servers, implementing a basic TCP server

10.5 Dual Band Wi-Fi Interface (MT3620)

The MT3620 device on the Azure Sphere module integrates a Wi-Fi 802.11 abgn radio with on-board dual-band chip antenna. This is used to connect Guardian to a wireless access point for Internet access. This separate Azure Sphere module PCB assembly has global regulatory certifications and is intended as a building block component for OEM boards.

10.6 Status / Indicator LEDs

Four LEDs are visible through the top-side enclosure lid of Guardian-100

The LED2, LED3 and LED4 functions are user defined in the application software and their intensity can be varied via MT3620 PWM settings

Status LEDs	Color	Ref. Des.	MT3620 GPIO	MT3620 Function
POWER	Green	LED1	-	-
1 (R)	Amber	LED2	GPIO8	GPIO / PWM
2 (G)	Amber	LED3	GPIO9	GPIO / PWM
3 (B)	Amber	LED4	GPIO10	GPIO / PWM

Two LEDs in the lower corners of the RJ45 connector, provide Ethernet status information
One LED is internal (only seen when enclosure lid is removed and programming USB cable is fitted)

Status LEDs	Color	Ref. Des.	LED is controlled by
FTDI USB Activity	Amber	LED5	FT4232 device
RJ45 LEDA	Green	n/a	ENC28J60 device
RJ45 LEDB	Yellow	n/a	ENC28J60 device

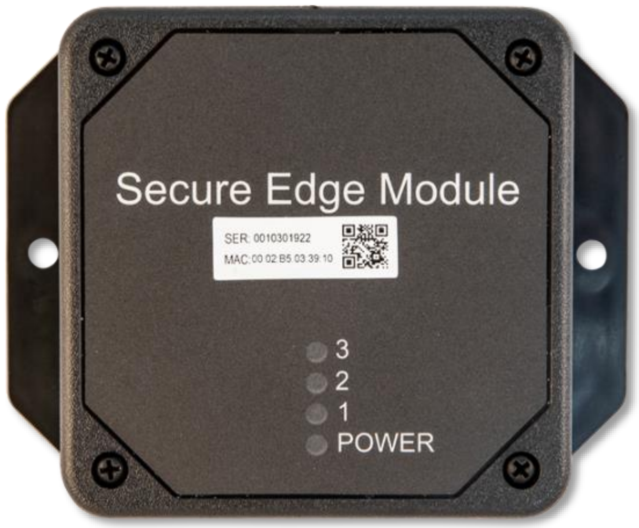


Figure 3 – Location of the Status LEDs

10.7 Hardware Expansion

Hardware expansion options are limited to devices that can be attached via the Wi-Fi, Ethernet and USB-Serial interfaces.

Internally Guardian-100 provides a limited set of GPIO on PCB test-points that can facilitate further functionality, but the enclosure does not provide connector access to these signals:

Test-Point PCB Label	MT3620 GPIO #	MT3620 Alternative Function
TP13	GPIO43	ADC2 input
IO0	GPIO86	IO0_TXD output
IO1	GPIO90	IO1_TXD output

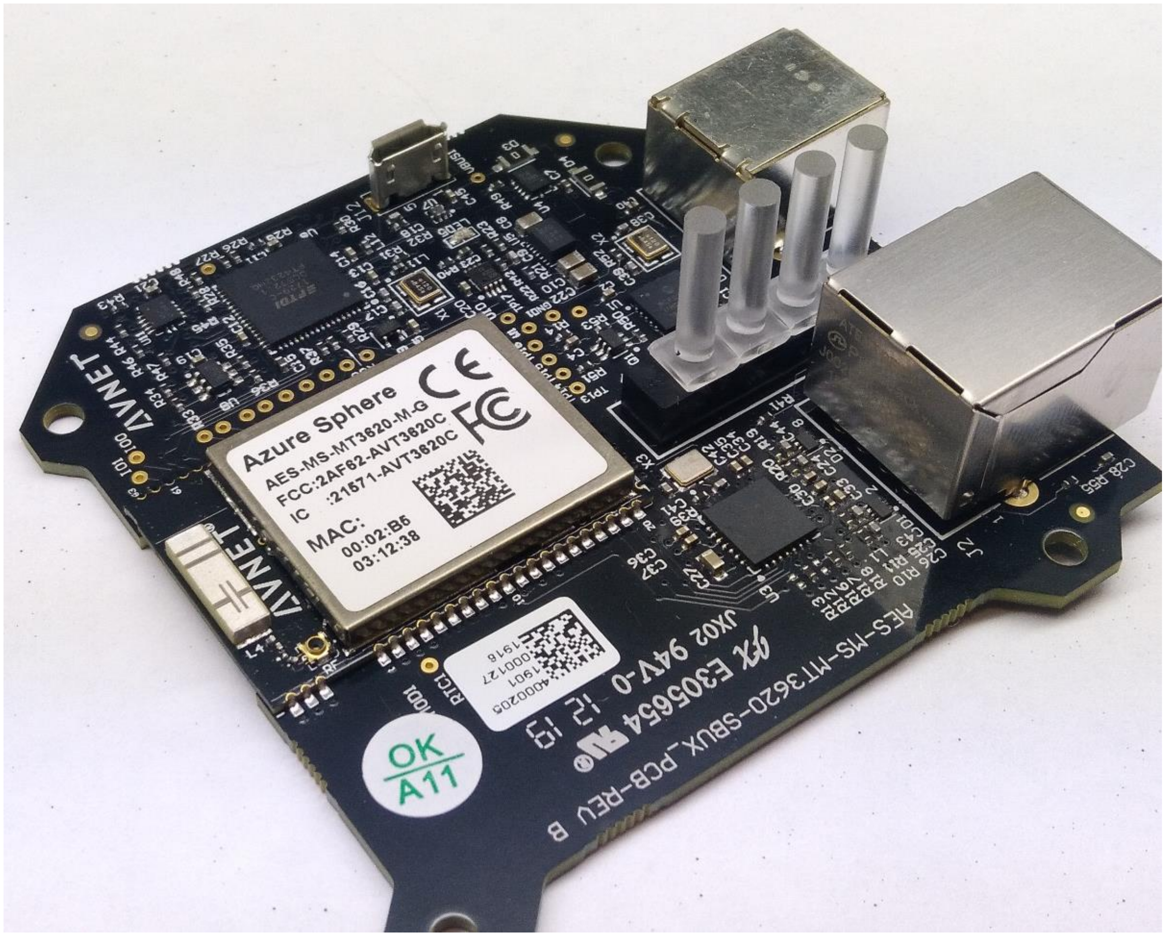


Figure 4 – Guardian-100 PCB Assembly

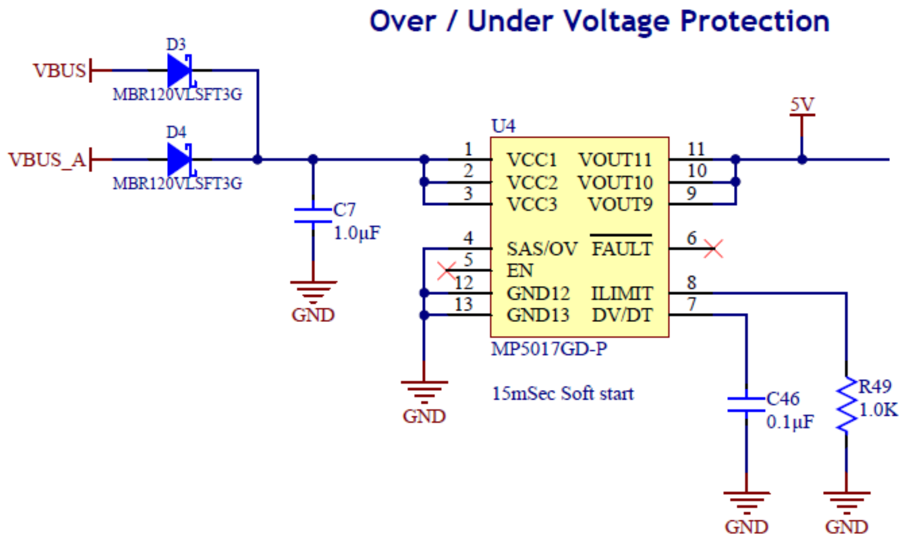
10.8 Power Interfaces, Regulation and Protection

The +5V supply to Guardian is from VBUS of one of the two USB cables, ie.

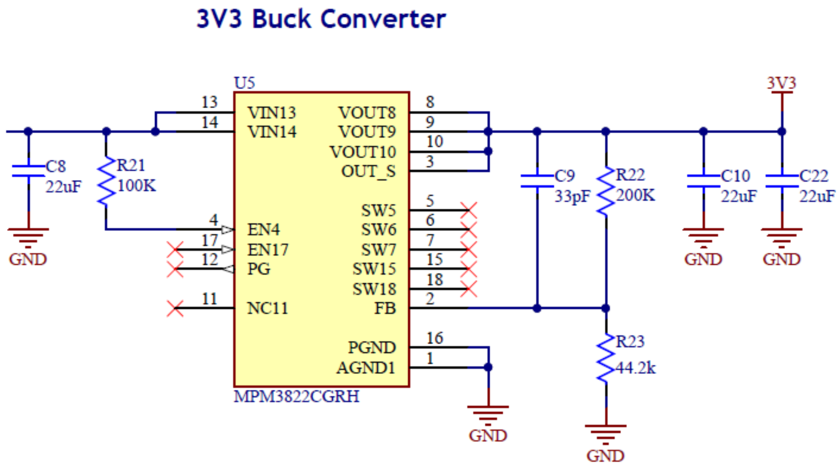
a) **USB FTDI Programmer Interface**
(used for RECOVERY, SERVICE, DEBUG and SWD interfaces)

b) **USB-UART Application Interface**
(used for UART-based communication between Guardian and the equipment)

An active “Over/Under Voltage Protection” circuit provides input voltage protection to the DC/DC regulator (MP5018GD input voltage max = 6.0V)



A 5V to 3.3V buck convertor (rated for 2A max) regulates the 3.3V VCC rail voltage



11 Contact Info and Technical Support

Documentation and reference designs are available for download from the product page:
http://avnet.me/mt3620-guardian

Relevant instructional blogs are also available under the Azure Sphere Starter Kit community page:
http://avnet.me/mt3620-kit

For further info on Avnet-designed Guardian-100s, contact your local Avnet representative at:

Region	Organization	Email	Address & Phone
North America	Avnet Americas	eval.kits@avnet.com	AVNET - Americas 2211 South 47th Street Phoenix, AZ 85034, USA Phone: +1-800-585-1602
Europe	Avnet Silica	Microsoft@silica.com	Avnet Silica Gruber Str. 60c 85586 Poing, Germany Phone: +49-8121-77702

12 Disclaimer

The Azure Sphere Guardian-100 Secure Edge Module is intended as a secure communication accessory to an end-product, without additional steps being performed to ensure regional certification compliance.

Avnet assumes no liability for modifications that a user chooses to make to this Guardian-100.

13 Safety Warnings

Safety Warnings

1) This product can be powered from one of two power sources:

    a) +5V via the provided microUSB cable, connected to the development computer
    b) +5V via the provided type-B USB cable, connected to end equipment that is rated to deliver at least 1.0 A. The end-equipment power source shall comply with relevant regulations and standards applicable in the country of intended use.

2) Only compatible equipment shall be connected to Guardian-100. Connection of incompatible equipment may affect compliance or result in damage to the unit and void the warranty.

3) This product must be operated in a well-ventilated environment.
   If an enclosure is used, this must provide adequate ventilation.

4) Ambient operating temperature when using Guardian-100 Starter Kit shall not exceed the range of: -30C to +85C

Appendix-A: Azure Sphere Module Pinout Detail

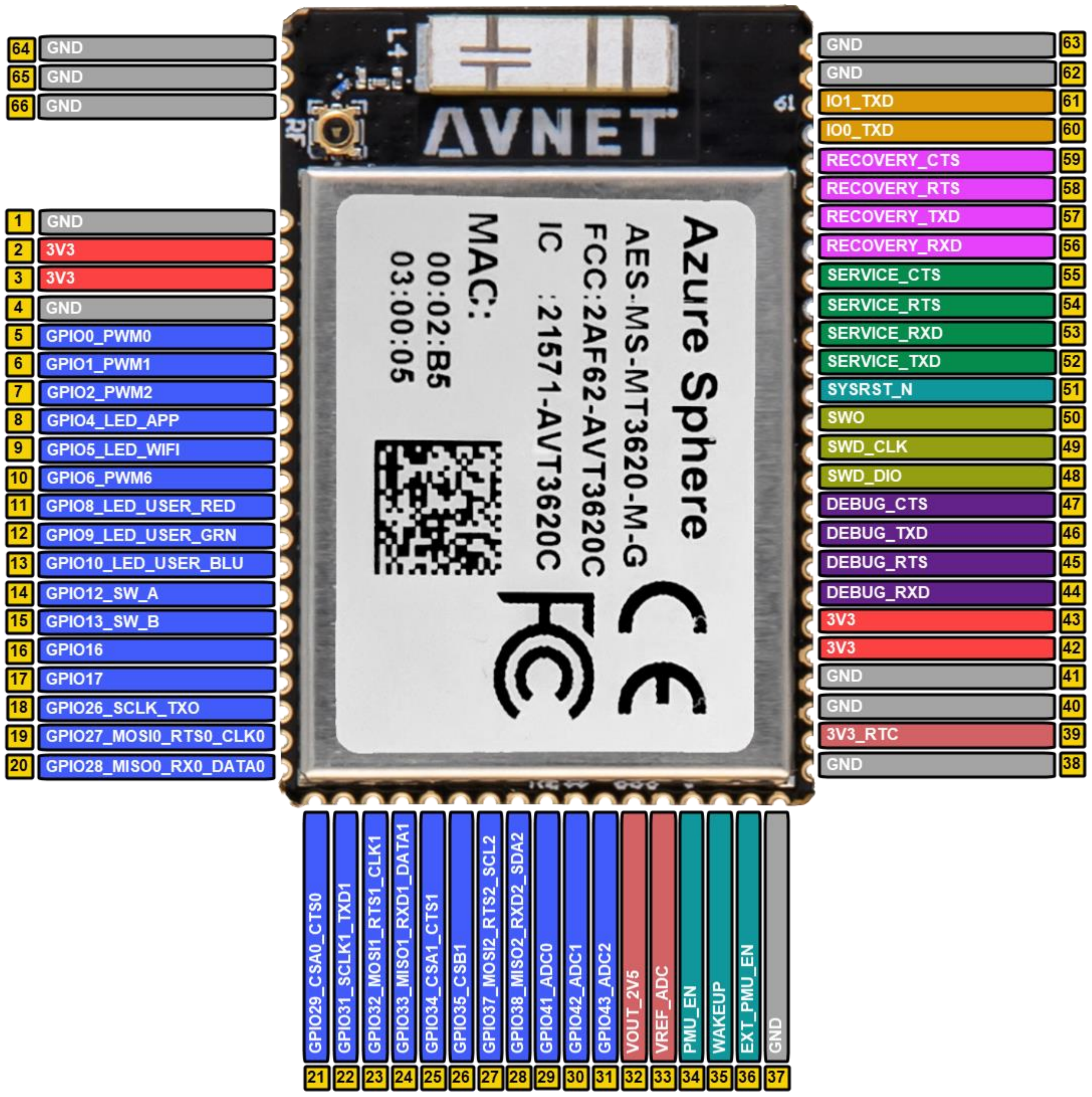


Figure 5 – Azure Sphere Module Pinout

Azure Sphere Module Pinout Detail

Module Pad	MT3620 Pad	MT3620 Net Name	I/O	Pin Function	Pre-Assigned Guardian-100 Function=blue
1		GND	GND		
2	2,3	3V3	Power		
3	2,3	3V3	Power		
4			GND		
5	13	GPIO0_PWM0	I/O	GPIO / INT in / PWM out	
6	14	GPIO1_PWM1	I/O	GPIO / INT in / PWM out	
7	15	GPIO2_PWM2	I/O	GPIO / INT in / PWM out	
8	17	GPIO4_PWM4	I/O	GPIO / INT in / PWM out	
9	18	GPIO5_PWM5	I/O	GPIO / INT in / PWM out	**INT_ENC**
10	19	GPIO6_PWM6	I/O	GPIO / INT in / PWM out	**RST_N_ENC**
11	21	GPIO8_PWM8	I/O	GPIO / INT in / PWM out	**GPIO8_LED_USER_R** (1)
12	22	GPIO9_PWM9	I/O	GPIO / INT in / PWM out	**GPIO9_LED_USER_G** (2)
13	25	GPIO10_PWM10	I/O	GPIO / INT in / PWM out	**GPIO10_LED_USER_B** (3)
14	27	GPIO12	I/O	GPIO / INT in	
15	28	GPIO13	I/O	GPIO / INT in	
16	31	GPIO16	I/O	GPIO / INT in	
17	32	GPIO17	I/O	GPIO / INT in	
18	39	GPIO26_SCLK0_TXD0	I/O	GPIO / ISU0	**SCK_ENC** (SPI0)
19	40	GPIO27_MOSI0_RTS0_SCL0	I/O	GPIO / ISU0	**MOSI_ENC** (SPI0)
20	42	GPIO28_MISO0_RXD0_SDA0	I/O	GPIO / ISU0	**MISO_ENC** (SPI0)
21	43	GPIO29_CSA0_CTS0	I/O	GPIO / ISU0	**CS_ENC** (SPI0)
22	46	GPIO31_SCLK1_TXD1	I/O	GPIO / ISU1	**APP_TXD** (UART1)
23	47	GPIO32_MOSI1_RTS1_SCL1	I/O	GPIO / ISU1	**APP_RTS** (UART1)
24	48	GPIO33_MISO1_RXD1_SDA1	I/O	GPIO / ISU1	**APP_RXD** (UART1)
25	49	GPIO34_CSA1_CTS1	I/O	GPIO / ISU1	**APP_CTS** (UART1)
26	50	GPIO35_CSB1	I/O	GPIO / ISU1	
27	52	GPIO37_MOSI2_RTS2_SCL2	I/O	GPIO / ISU2	
28	53	GPIO38_MISO2_RXD2_SDA2	I/O	GPIO / ISU2	
29	58	GPIO41_ADC0	I/O	GPIO / ADC in	
30	59	GPIO42_ADC1	I/O	GPIO / ADC in	
31	60	GPIO43_ADC2	I/O	GPIO / ADC in	**TP13**
32	66	VOUT_2V5	AO		
33	67	VREF_ADC	AI		min 1.8V, max 2.5V
34	81	PMU_EN	I		pull-up on module
35	70	WAKEUP	I	Ext. Wakeup Input	pull-up on module
36	69	EXT_PMU_EN	O	Ext. 3V3 regulator enable	
37		GND	GND		

Module Pinout Detail (continued)

Module Pad	MT3620 Pad	MT3620 Net Name	I/O	Pin Function	Pre-Assigned Guardian-100 Function=BLUE
38		GND	GND		
39	71	3V3_RTC	Power		min 2.50 V, max 3.63V
40		GND	GND		
41		GND	GND		
42	88,89	3V3	Power		
43	88,89	3V3	Power		
44	94	DEBUG_RXD	I	Debug UART	**DEBUG_RXD**
45	96	DEBUG_RTS	O	Debug UART (pulled-down / FTDI controlled strapping state on Guardian-100)	**DEBUG_RTS**
46	95	DEBUG_TXD	O	Debug UART (pulled-down on module)	**DEBUG_TXD**
47	97	DEBUG_CTS	I	Debug UART	**DEBUG_CTS**
48	98	SWD_DIO	I/O	CM4F SWD	**SWD_DIO**
49	99	SWD_CLK	I	CM4F SWD	**SWD_CLK**
50	100	SWO	O	CM4F SWD	**SWO**
51	125	SYSRST_N	I		**SYSRST_N**
52	127	SERVICE_TXD	O	Service UART	**SERVICE_TXD**
53	129	SERVICE_RXD	I	Service UART	**SERVICE_RXD**
54	128	SERVICE_RTS	O	Service UART	**SERVICE_RTS**
55	130	SERVICE_CTS	I	Service UART	**SERVICE_CTS**
56	134	RECOVERY_RXD	I	Recovery UART	**RECOVERY_RXD**
57	135	RECOVERY_TXD	O	Recovery UART (PU on module)	**RECOVERY_TXD**
58	136	RECOVERY_RTS	O	Recovery UART (pulled-down on module)	**RECOVERY_RTS**
59	137	RECOVERY_CTS	I	Recovery UART	**RECOVERY_CTS**
60	139	IO0_GPIO86/IO0_TXD	O	IO0_GPIO / IO0_TXD (pulled-down on module)	**IO0_TXD**
61	143	IO1_GPIO90/IO1_TXD	O	IO1_GPIO / IO1_TXD (pulled-down on module)	**IO1_TXD**
62 - 66		GND	GND	GND pour	
67		PADGND	GND	Thermal pad for MT3620	

Appendix-B: Running Pre-Compiled Example Applications

Pre-compiled, production-signed applications can be side-loaded to the MT3620 from **Azure Sphere SDK** without need to rebuild an executable file from Microsoft Visual Studio.

iPerf3 Test Application to Check Wi-Fi Performance

Download the following file and copy it to **C:\TEST**
`iperf3_ps.imagepackage`

iPerf3 Server:

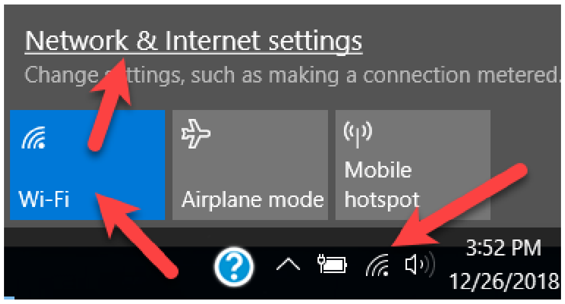
On the development computer, the following setup is required before commencing testing:

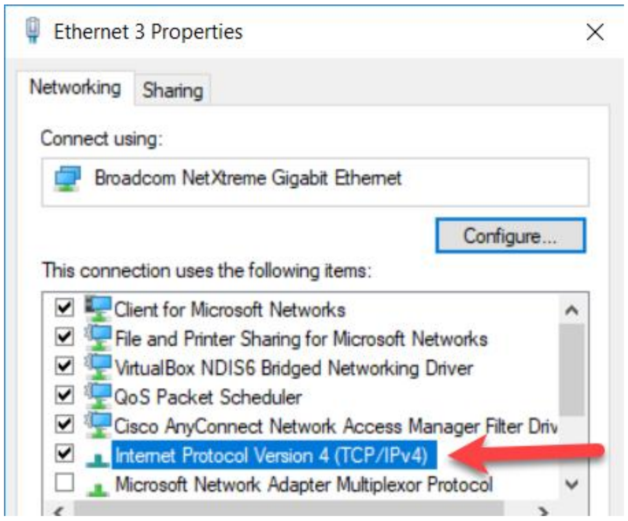
a) Download and unzip the iPerf 3.1.3 Windows application from https://iperf.fr/iperf-download.php

b) It is recommended to turn-off the computer's Wi-Fi adaptor and instead use a wired LAN connection from computer to the network router

c) Connect the ethernet port of the test computer via CAT-5 cable connection to the Network Router whose Wi-Fi access points Guardian will connect to

d) Configure the computer's **ethernet adaptor** to have a static IP address of **192.168.1.35** (plus the other highlighted settings shown below…)


Network & Internet settings
Change settings, such as making a connection metered.
Wi-Fi
Airplane mode
Mobile hotspot
3:52 PM
12/26/2018

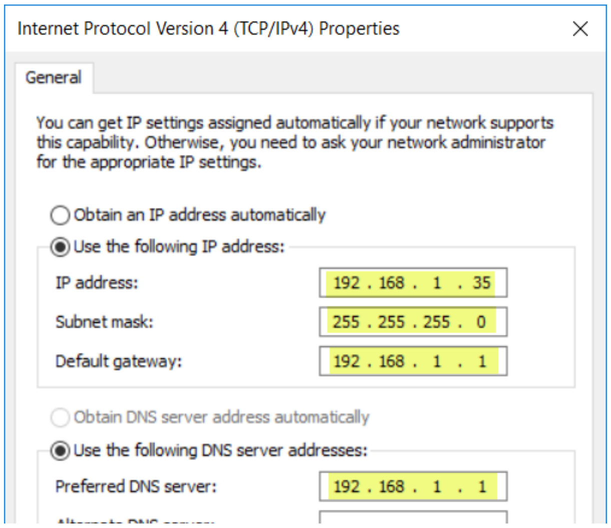

Ethernet 3 Properties

Networking | Sharing

Connect using:
Broadcom NetXtreme Gigabit Ethernet

Configure...

This connection uses the following items:
- [✓] Client for Microsoft Networks
- [✓] File and Printer Sharing for Microsoft Networks
- [✓] VirtualBox NDIS6 Bridged Networking Driver
- [✓] QoS Packet Scheduler
- [✓] Cisco AnyConnect Network Access Manager Filter Driv
- [✓] Internet Protocol Version 4 (TCP/IPv4)
- [ ] Microsoft Network Adapter Multiplexor Protocol


Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

Obtain an IP address automatically
[✓ Use the following IP address:]

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

Obtain DNS server address automatically
[✓ Use the following DNS server addresses:]

Preferred DNS server:	192 . 168 . 1 . 1

Launch the **iperf3 server** using the command: **`iperf3 –s -1`**

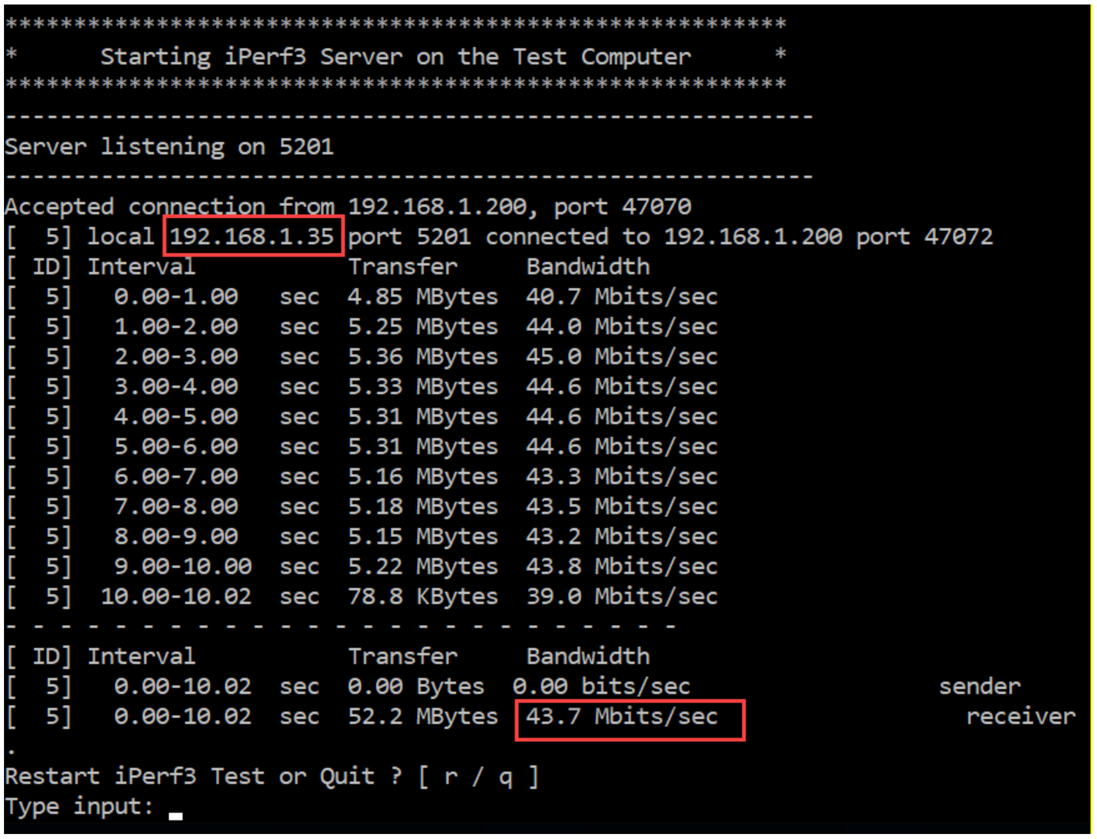
<u>iPerf3 Client</u>:

Use the following command to configure the Wi-Fi settings:
(replace **??????** with the applicable credentials)

```
azsphere device wifi add -ssid ?????? -psk ???????
```

Note: Guardian **<u>must</u>** connect to the <u>same</u> subnet as the computer that is running the iPerf3 server!

After Wi-Fi connection is established, reported iPerf bitrates should start appearing in the console window



```
*********************************************************
*      Starting iPerf3 Server on the Test Computer      *
*********************************************************
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
Accepted connection from 192.168.1.200, port 47070
[  5] local 192.168.1.35 port 5201 connected to 192.168.1.200 port 47072
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-1.00   sec  4.85 MBytes  40.7 Mbits/sec
[  5]   1.00-2.00   sec  5.25 MBytes  44.0 Mbits/sec
[  5]   2.00-3.00   sec  5.36 MBytes  45.0 Mbits/sec
[  5]   3.00-4.00   sec  5.33 MBytes  44.6 Mbits/sec
[  5]   4.00-5.00   sec  5.31 MBytes  44.6 Mbits/sec
[  5]   5.00-6.00   sec  5.31 MBytes  44.6 Mbits/sec
[  5]   6.00-7.00   sec  5.16 MBytes  43.3 Mbits/sec
[  5]   7.00-8.00   sec  5.18 MBytes  43.5 Mbits/sec
[  5]   8.00-9.00   sec  5.15 MBytes  43.2 Mbits/sec
[  5]   9.00-10.00  sec  5.22 MBytes  43.8 Mbits/sec
[  5]  10.00-10.02  sec  78.8 KBytes  39.0 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-10.02  sec  0.00 Bytes  0.00 bits/sec                  sender
[  5]   0.00-10.02  sec  52.2 MBytes  43.7 Mbits/sec                  receiver
.
Restart iPerf3 Test or Quit ? [ r / q ]
Type input: _
```

<u>Notes</u>:
The iPerf3 Client and iPerf3 Server **<u>must</u>** both be on the same network sub-net
ie, connected via same network router

If the reported bandwith (bitrate) is zero, use:
CTL+C to stop the iPerf3 application, then
`iperf3 -s` to restart the iPerf3 Server application (on the development computer)