GUARDIAN III USER GUIDE

Changhong-Microsoft IoT Innovation Center

Revision History

Version	Description	Author	Date
V0.1	Initial version	Junhui Lan	7-21-2021
V1.0	Update	Junhui Lan	10-11-2021
V1.1	Verification and correction	Richard Liang	01-13-2022

Contents

1 Introduction

Guardian III is an IoT gateway product series based on Microsoft Azure Sphere OS. It provides end-to-end IoT security solutions and can be used in a variety of industrial fields.

Guardian III has multiple types of southbound interfaces, such as LoRa, Bluetooth, Ethernet, CAN, RS422, RS485, etc., which can be easily connected to many devices or sensors. At the same time, it also includes three types of northbound interfaces (Wi-Fi, Ethernet, Cellular), you can easily connect to the Internet.

The Guardian III series includes three different product forms, namely GIIICM680L, GIIICM680E, and GIIICM680R. The main difference between the three product forms is the southbound interface.

Product features:

GIIICM680L (GIIICM680L, GIIICM680L-CM2, GIIICM680L-CM6, GIIICM680L-CMA)

- AI-Link WF-M620-RSD2 module
- Northbound link:
  - Cellular(4G/LTE) data communication (GIIICM680L does not include this feature)
  - Dual-band Wi-Fi, 2.4GHz and 5GHz
  - Ethernet port: RJ45 female, 10Mbps
- Southbound link:
  - Bluetooth 5.0 connectivity
  - LoRa data communication
  - USB port (Power + Data)
- GNSS location report
- MCU programming and debugging port
- A/C adapter DC 5V/2A
- 3 LED indicators: 2 programmable Red/Green LEDs, 1 BLE activity LED
- Operating temperature: -40 °C to +85 °C
- Operating humidity: 5% to 95% RH
- Dimensions: 95mm*95mm*25mm
- Enclosure: Mountable, Black color

GIIICM680E (GIIICM680E, GIIICM680E-CE, GIIICM680E-CN)

- AI-Link WF-M620-RSD2 module
- Northbound link:
  - Cellular(4G/LTE) data communication (GIIICM680E does not include this feature)
  - Dual-band Wi-Fi, 2.4GHz and 5GHz
  - Ethernet port: RJ45 female, 10Mbps
- Southbound link:
  - Bluetooth 5.0 connectivity
  - Ethernet port: RJ45 female, up to 1Mbps
  - USB port (Power + Data)
- GNSS location report
- MCU programming and debugging port
- A/C adapter DC 5V/2A

- 3 LED indicators: 2 programmable Red/Green LEDs, 1 BLE activity LED
- Operating temperature: -40 °C to +85 °C
- Operating humidity: 5% to 95% RH
- Dimensions: 95mm*95mm*25mm
- Enclosure: Mountable, Black color

GIIICM680R (GIIICM680R, GIIICM680R-CE, GIIICM680R-CN)

- AI-Link WF-M620-RSD2 module
- Northbound link:
  - Cellular(4G/LTE) data communication (GIIICM680R does not include this feature)
  - Dual-band Wi-Fi, 2.4GHz and 5GHz
  - Ethernet port: RJ45 female, 10Mbps
- Southbound link:
  - Bluetooth 5.0 connectivity
  - RS-422 or RS-485
  - CAN
- Modbus protocol supported
- GNSS location report
- MCU programming and debug port
- A/C adapter DC 5V/2A
- 3 LED indicators: 2 programmable Red/Green LED and 1 BLE activity LED,
- Operating temperature: -40 °C to +85 °C
- Operating humidity: 5% to 95% RH
- Dimensions: 95mm*95mm*25mm
- Enclosure: Mountable, Black color

2 Components

2.1 Hardware List

Item	Number	Description
GIIICM680L/ GIIICM680E/ GIIICM680R	1	Main product
Debug Dongle cable	1	Azure sphere debug cable.
Micro USB power cable	1	Guardian III power for main board.
Micro USB data cable	1	Guardian III for device testing.
5V2A DC Power Adapter	1	Guardian III power for main board

Actual pictures:

Size: 95mm×95mm×25mm

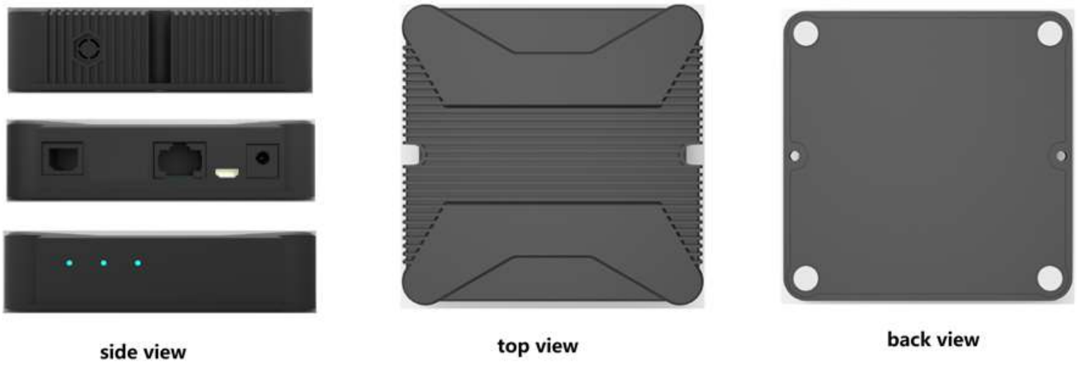


Figure 1:GIIICM680L Box Appearance

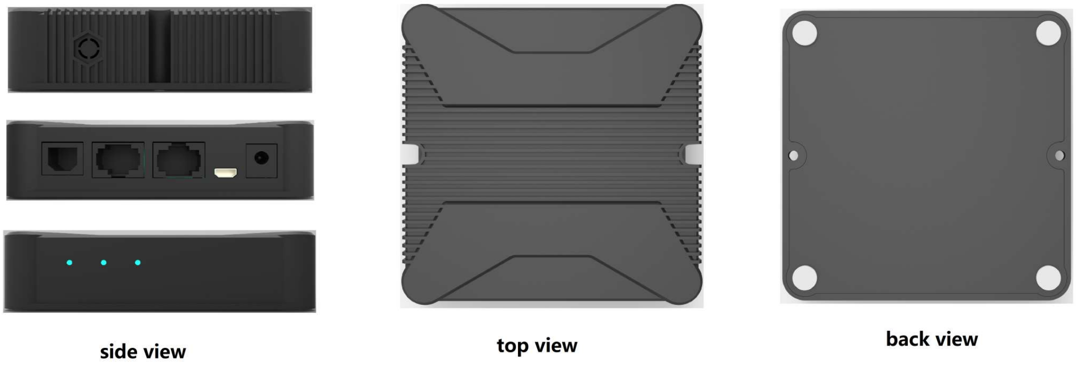


Figure 2: GIIICM680E Box Appearance

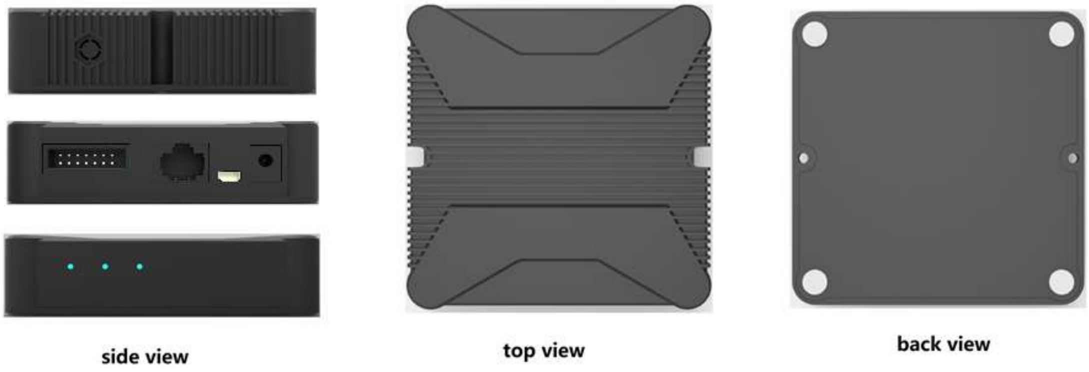


Figure 3: GIIICM680R Box Appearance

2.2 Software List

Product Software:

Item	Version	Description
CRLGeneralTool.apk	1.2.5	Guardian III General Android APP.
Android OS	8.0+	Required to run APP, not provided.

putty	0.74	Download URL: https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html
SouthBound_Ethernet_signed.imagepackage	1.0	Southbound W5500 driver.
FreeRTOS_RTcore_LoRa_signed.imagepackage	1.0	LoRa driver.
lan-enc28j60-isu0-int5_signed.imagepackage	1.0	Northbound network card driver.
GIIICM680L_Demo.zip	1.0	GIIICM680L sample code
GIIICM680E_Demo.zip	1.0	GIIICM680E sample code.
GIIICM680R_Demo.zip	1.0	GIIICM680R sample code.

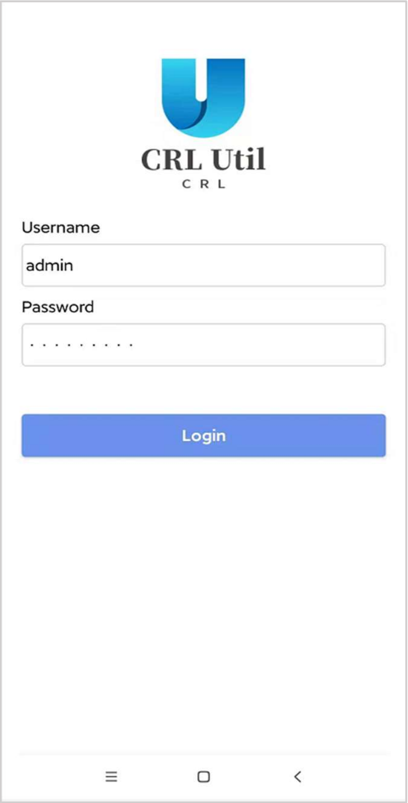
3 Mobile APP

3.1 Install APP

The main human-computer interaction functions of the Guardian III system are realized through the mobile phone APP. These functions include configuring equipment and querying basic equipment information. Currently, the APP only supports Android phones. Download the current version of the APK file from the CRL support website to the computer, then upload the file to the phone directory through the data cable of the Android phone, then use the phone file management to find the APK file, perform the installation, and then run the APP.

3.2 APP Login

Running APP, the login page shows as figure 4: (initial account: admin, password: adminTest). Click "Login" button, and then login the home Interface, as figure 5 show.



Username: admin

Password: · · · · · · · · ·

Figure 4: The Login in Interface

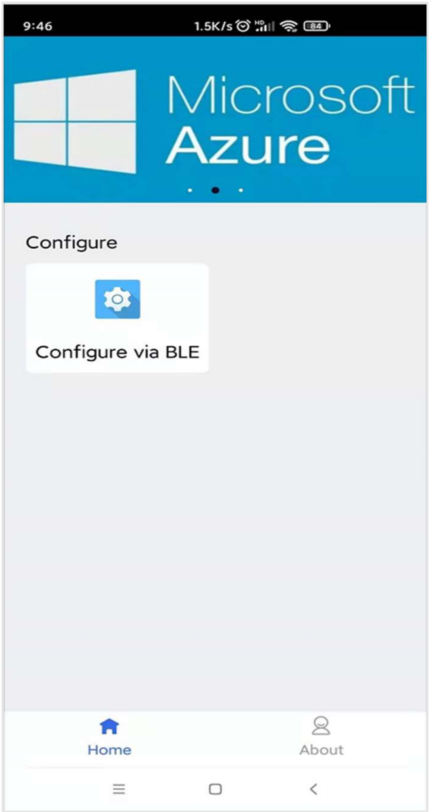
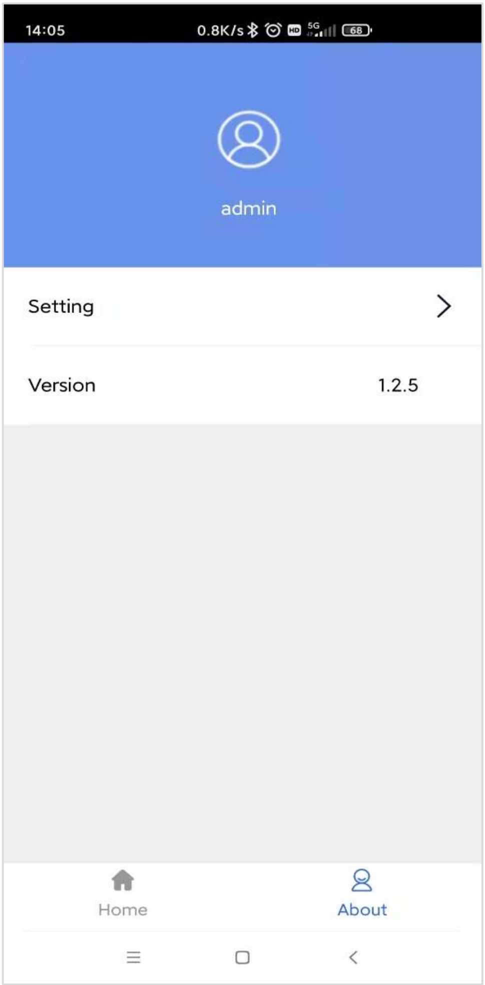


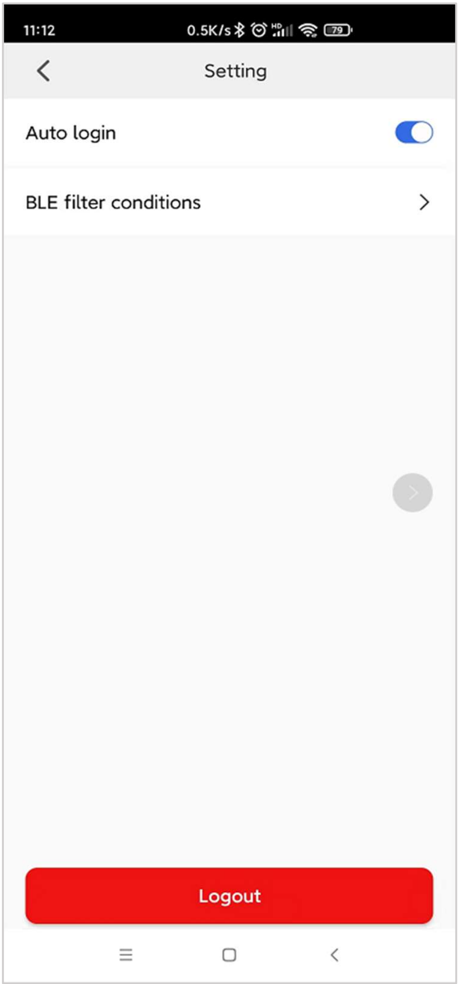
Figure 5:   The Home Interface

The APP provides automatic login and logout function. Running APP and clicking the "About" button, then users can view the user's information, APP information etc. on the "About" interface showing in figure 6. Users can set automatic login in the “Settings” page and click “Logout” button to logout the current account. As figure 7 shows.



Version 1.2.5

Figure 6: The Account Details Interface

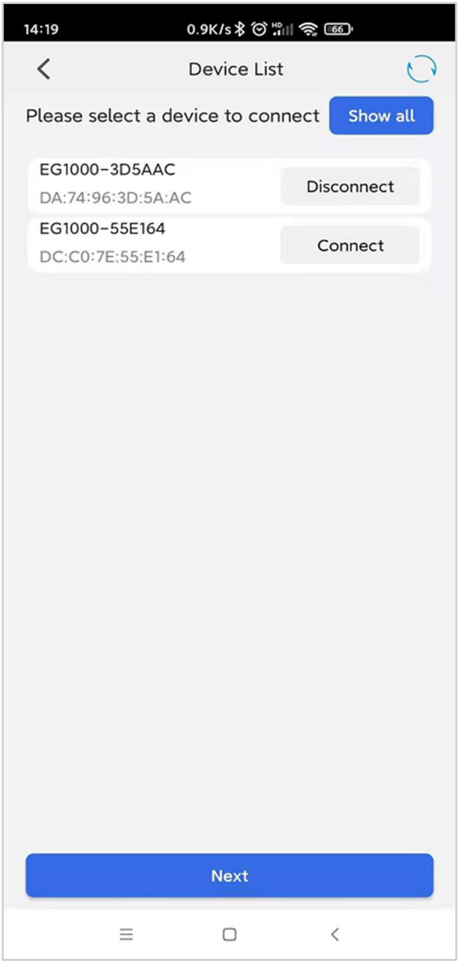


Auto login

Figure 7: The Auto Login Setting Interface

3.3 Query device details

Select "Configure via BLE" on the main interface of the APP, and the APP will actively scan the peripheral devices and display them in a list. as figure 8 show.



14:19 0.9K/s

Device List

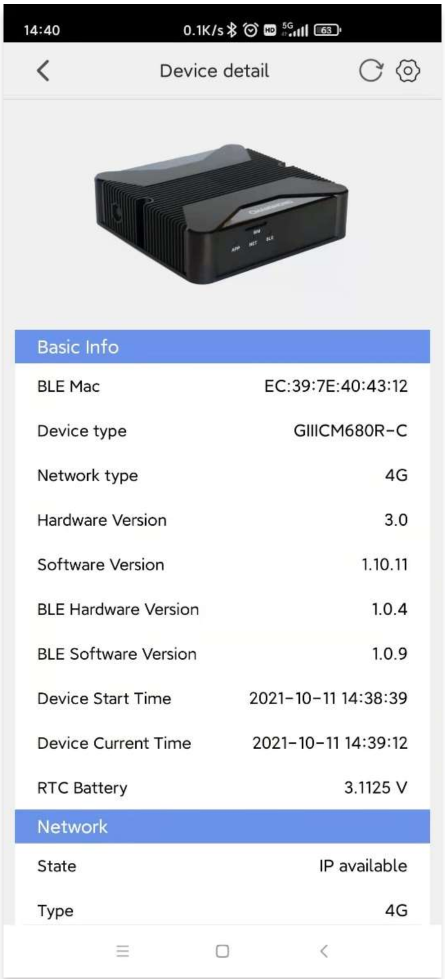
Please select a device to connect

EG1000-3D5AAC
DA:74:96:3D:5A:AC

EG1000-55E164
DC:C0:7E:55:E1:64

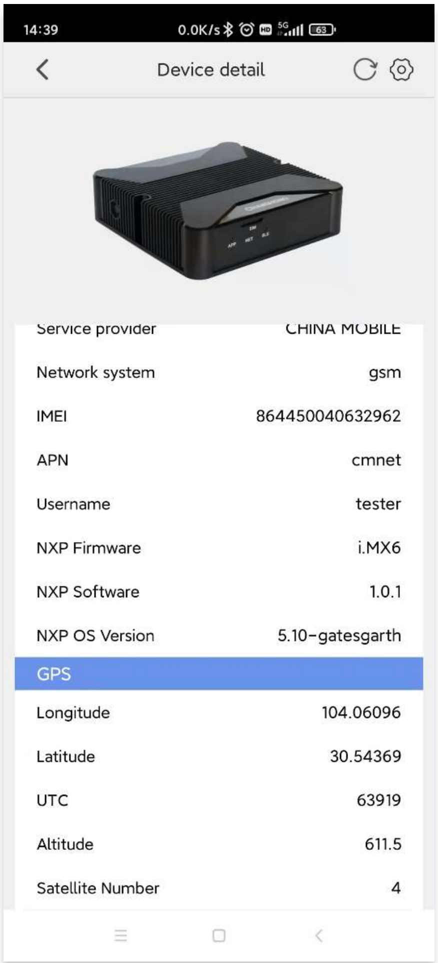
Figure 8: Device list

After selecting the target device and clicking "Connect", the APP will automatically connect with the device. After the connection is successful (the BLE Led of the device changes from flashing to constant light), the device will actively return the basic information of the device, as figure 9 show. If it is Cellular mode, GPS will be included in the return information, as figure 10 show.



Basic Info	
BLE Mac	EC:39:7E:40:43:12
Device type	GIIICM680R-C
Network type	4G
Hardware Version	3.0
Software Version	1.10.11
BLE Hardware Version	1.0.4
BLE Software Version	1.0.9
Device Start Time	2021-10-11 14:38:39
Device Current Time	2021-10-11 14:39:12
RTC Battery	3.1125 V
Network	
State	IP available
Type	4G

Figure 9: Devices Details

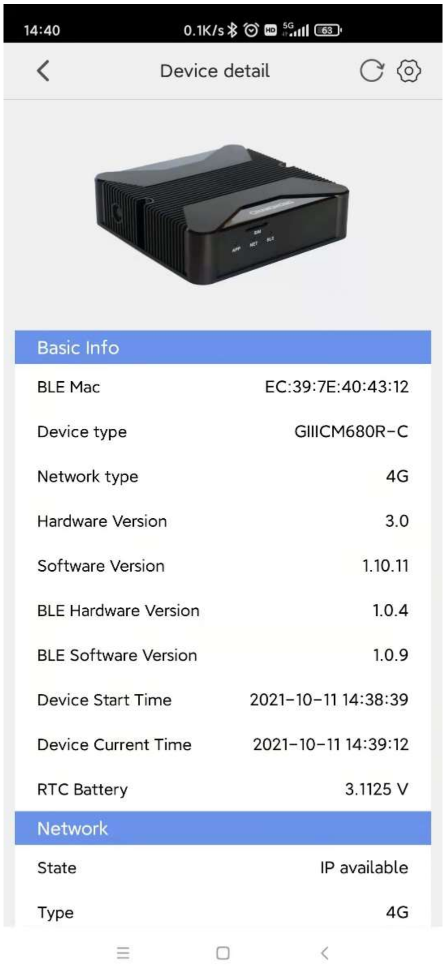


Service provider	CHINA MOBILE
Network system	gsm
IMEI	864450040632962
APN	cmnet
Username	tester
NXP Firmware	i.MX6
NXP Software	1.0.1
NXP OS Version	5.10-gatesgarth
GPS	
Longitude	104.06096
Latitude	30.54369
UTC	63919
Altitude	611.5
Satellite Number	4

Figure 10: GPS Information

3.4 Configure Network

In the device details interface, click the configuration button in the upper left corner, and the APP will jump to the device configuration interface. as figure 11 and 12 show.



14:40 0.1K/s

Device detail

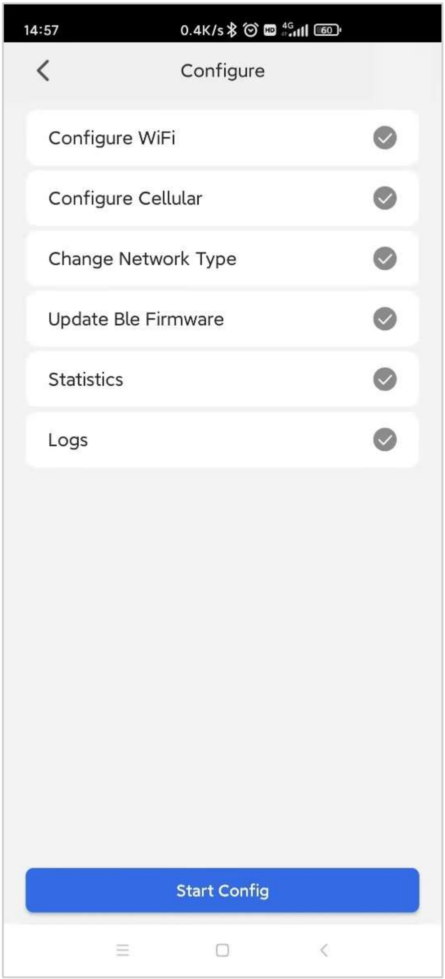
Basic Info

BLE Mac	EC:39:7E:40:43:12
Device type	GIIICM680R-C
Network type	4G
Hardware Version	3.0
Software Version	1.10.11
BLE Hardware Version	1.0.4
BLE Software Version	1.0.9
Device Start Time	2021-10-11 14:38:39
Device Current Time	2021-10-11 14:39:12
RTC Battery	3.1125 V

Network

State	IP available
Type	4G

Figure 11: Enter the configuration interface



14:57 0.4K/s

Configure

- Configure WiFi ✓
- Configure Cellular ✓
- Change Network Type ✓
- Update Ble Firmware ✓
- Statistics ✓
- Logs ✓

Start Config

Figure 12: Configuration interface

3.4.1 Configure Wi-Fi

Select "Configure Wi-Fi" in the configuration interface, and click "Start Config", and then the APP will jump to the Wi-Fi configuration interface. as figure 13 show.

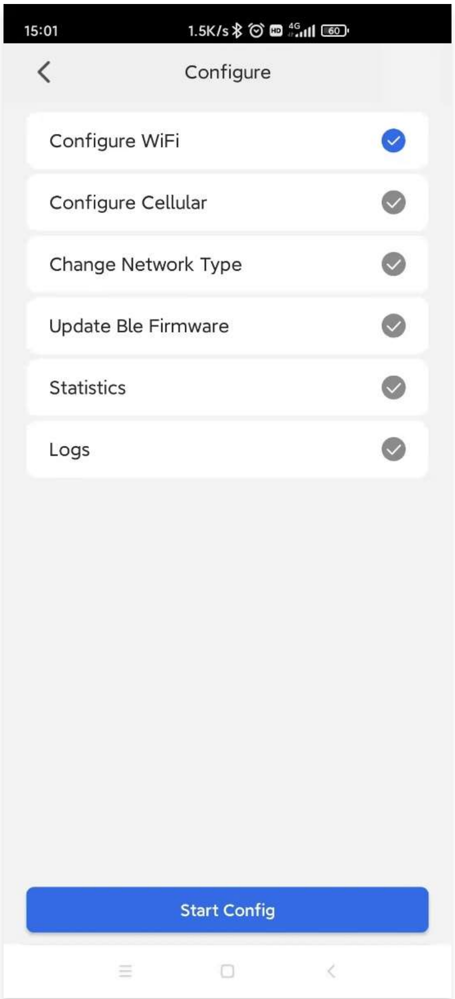
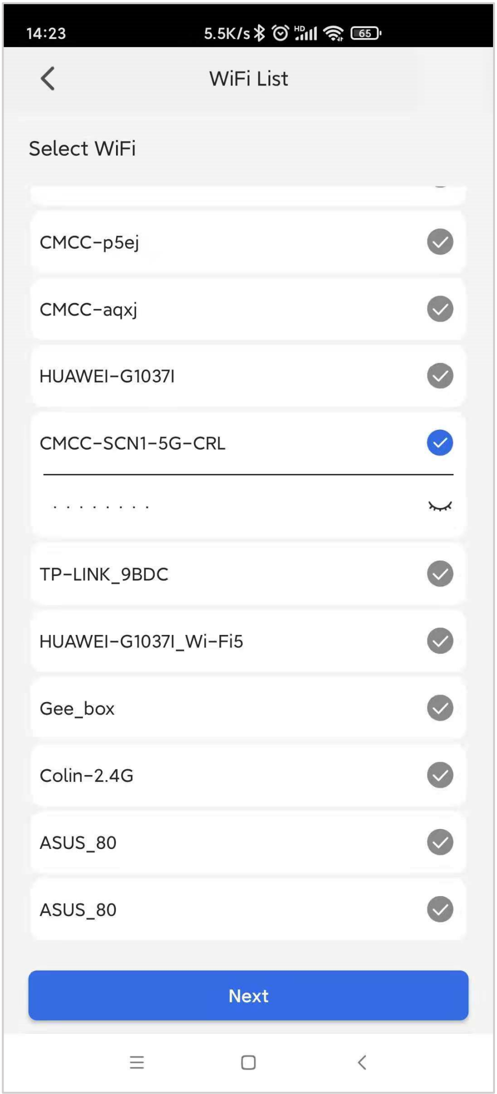


Figure 13: the Configure Interface

Wi-Fi Access Point (AP) hot spot will list after scanning. User can choose Wi-Fi AP, and then input Wi-Fi password. Click “Next” button and the APP will transmit Wi-Fi configuration parameters to devices by BLE showing in figure 14.



Select WiFi

CMCC-p5ej

CMCC-aqxj

HUAWEI-G1037I

CMCC-SCN1-5G-CRL ✓

· · · · · · · ·

TP-LINK_9BDC

HUAWEI-G1037I_Wi-Fi5

Gee_box

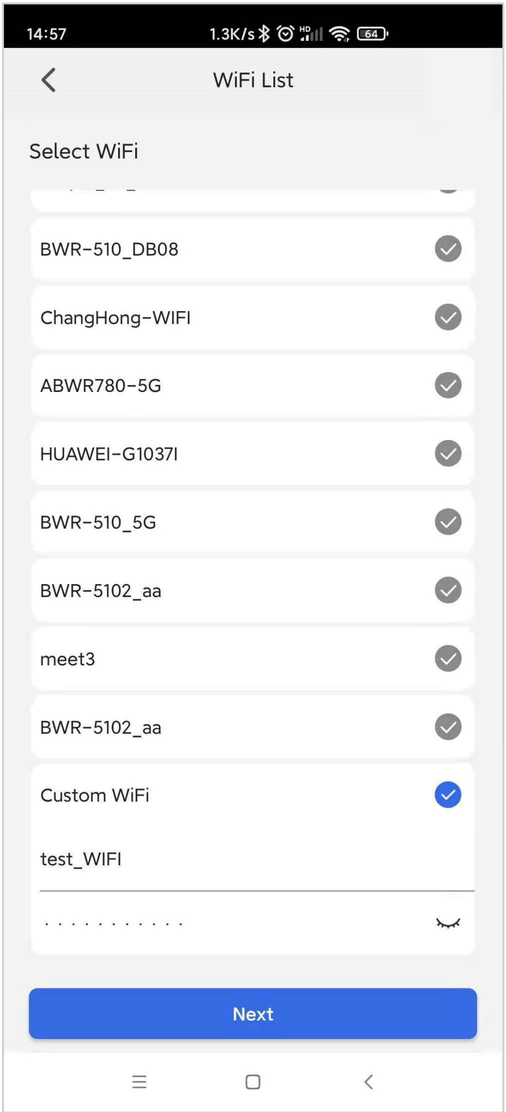
Colin-2.4G

ASUS_80

ASUS_80

Figure 14: the Wi-Fi List Interface

But for the Wi-Fi AP which is not in the Wi-Fi list, users can choose ”Custom Wi-Fi” and input Wi-Fi AP name and password to connect in the bottom of Wi-Fi list. Shows in figure 15.



14:57 1.3K/s

WiFi List

Select WiFi

BWR-510_DB08

ChangHong-WIFI

ABWR780-5G

HUAWEI-G1037I

BWR-510_5G

BWR-5102_aa

meet3

BWR-5102_aa

Custom WiFi

test_WIFI

. . . .

Next

Figure 15: Wi-Fi custom configuration

3.4.2 Configure Cellular

Select "Configure Cellular" in the network configuration interface, and the APP will jump to the Cellular configuration interface. as figure 16 show.

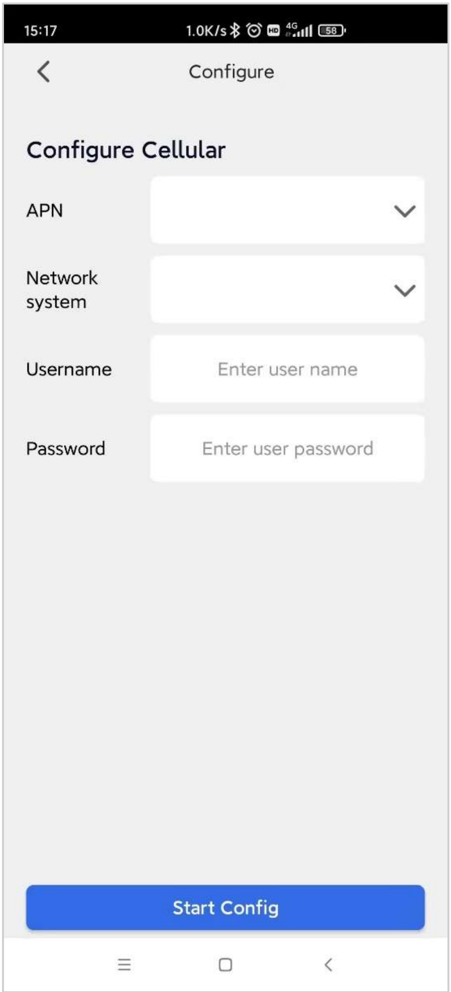
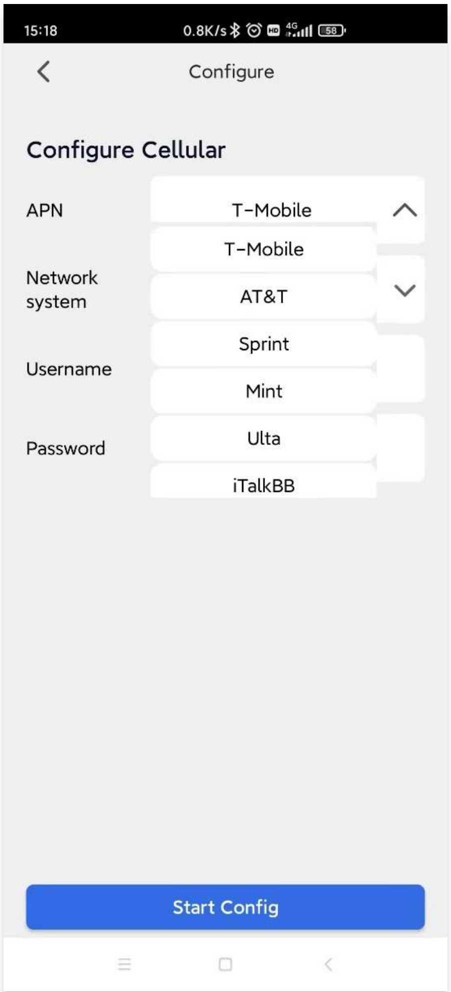


Figure 16: the Cellular List Interface

In the APP built-in part of the operator's APN, you can select your target operator from the drop-down list, and then click "Next", the APP will automatically send the operator's APN information to the device. as figure 17 show.



15:18 0.8K/s

Configure

Configure Cellular

APN T-Mobile

Network system

Username

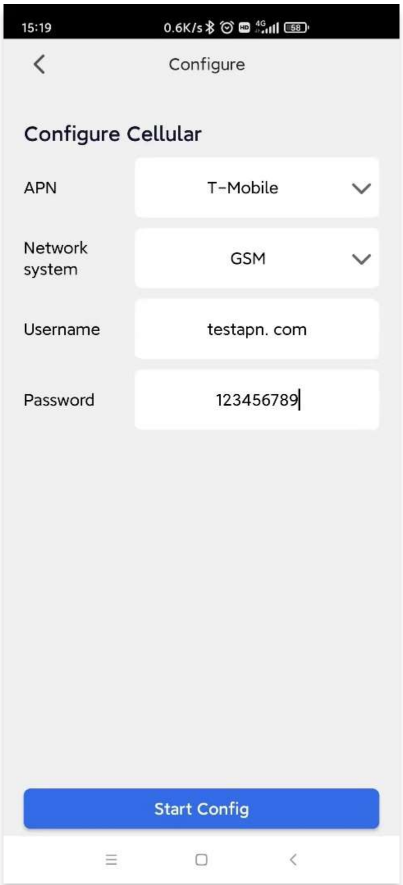
Password

T-Mobile
AT&T
Sprint
Mint
Ulta
iTalkBB

Start Config

Figure 17: Choose operator

If your target operator is not in the list, you can select "Fill the form" and fill in your target APN information. as figure 18 show.



15:19 0.6K/s

Configure

Configure Cellular

APN: T-Mobile

Network system: GSM

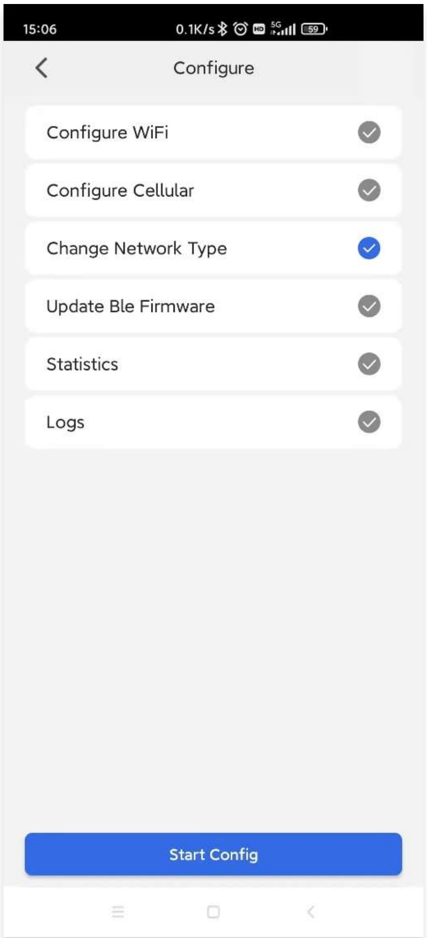
Username: testapn. com

Password: 123456789

Figure 18: Custom APN

3.5 Change Network

In the device configuration interface (figure 19), select "Change Network" to switch the network type showing in figure 20.



15:06 0.1K/s

Configure

Configure WiFi ✓

Configure Cellular ✓

Change Network Type ☑

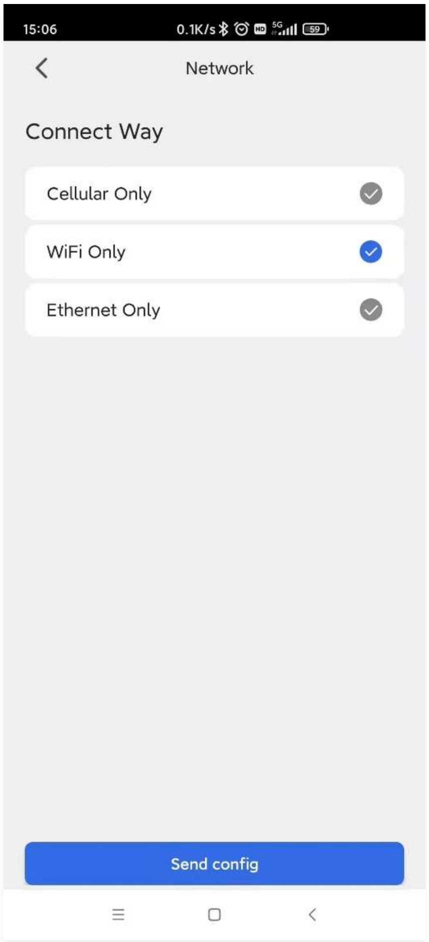
Update Ble Firmware ✓

Statistics ✓

Logs ✓

Start Config

Figure 19 : Change Network interface



15:06 0.1K/s

Network

Connect Way

Cellular Only ✓

WiFi Only ☑

Ethernet Only ✓

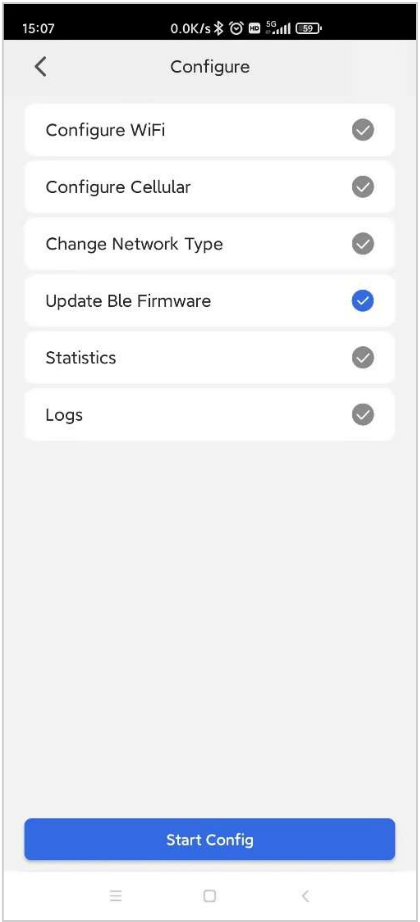
Send config

Figure 20: Change To Wi-Fi

Click "Send Config" and the APP will send a switch network type command to the device. The switching process takes about 1 minute. After the switch is completed, you can view the switched network type on the "Device Details" page.

3.6 Update BLE Firmware

Select "Update BLE Firmware" on the configuration interface and then click “Start config” in the page of BLE Upgrade as figure 21 show.
At the “BLE Upgrade” page, click “Browse” to choose the BLE upgrade file (normally a ZIP file), besides choose the name of BLE in the chip. Click the “Upgrade” button in the bottom. Then the APP will send the upgrade file to the BLE chip and start upgrading. As shows in figure 22.



15:07

Configure

Configure WiFi

Configure Cellular

Change Network Type

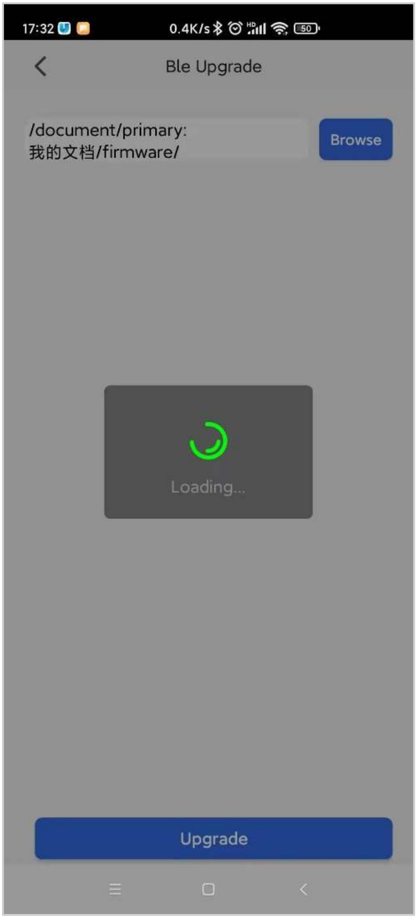
Update Ble Firmware ✓

Statistics

Logs

Start Config

Figure 21: the Configuration Interface



17:32

Ble Upgrade

/document/primary:
我的文档/firmware/

Browse

Loading...

Upgrade

Figure 22: the BLE Upgrade Interface

4 Get started With the Box

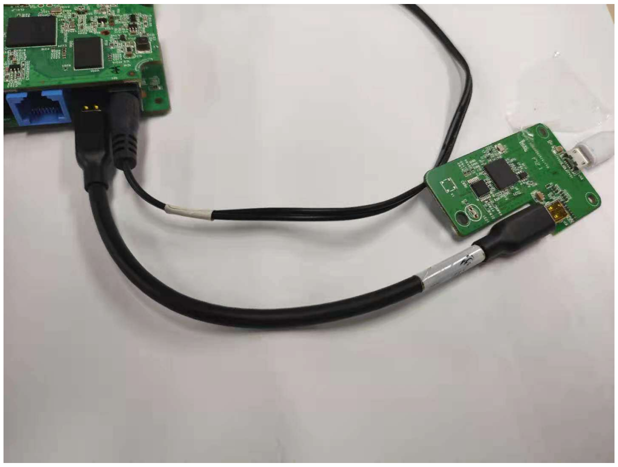
To use Guardian III series products for the first time, you must install Azure Sphere SDK and Dongle debug version driver.

For the establishment of Azure sphere development environment, please refer to:

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

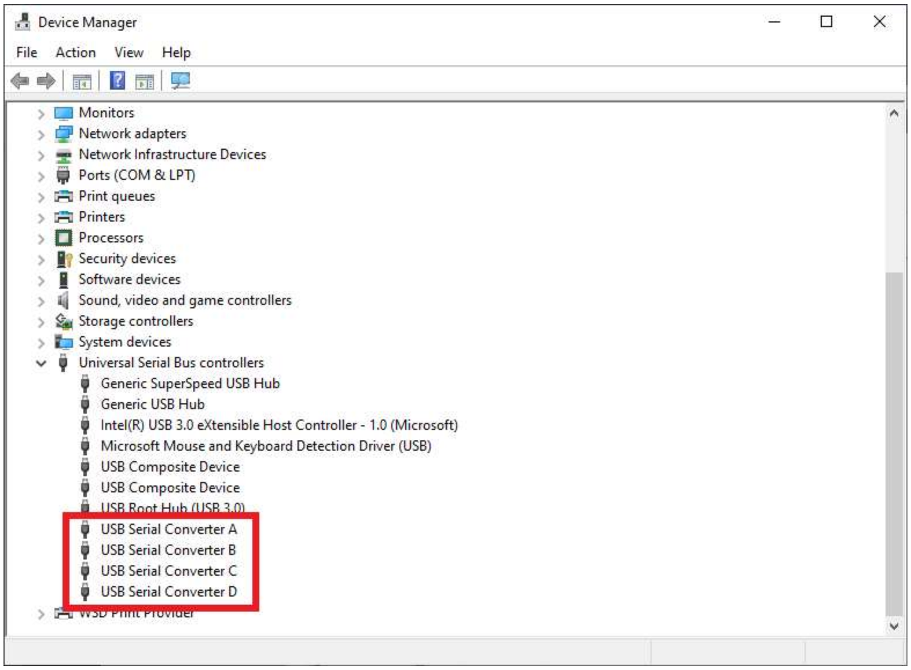
4.1 Connecting the device

Connect Guardian III series products to your PC through USB.



When plugged in, the device exposes four USB Serial Converters, the first time you plug in the device, the drivers should be automatically downloaded and installed. Installation can be slow. If the drivers are not installed automatically, right-click the device name in Device Manager and select Update driver. Alternatively, you can download the drivers from Future Technology Devices International (FTDI). Choose the driver that matches your Windows installation (32-bit or 64-bit).

To verify installation, open Device Manager. Under Universal Serial Bus controllers, look for four USB Serial Converters. Device Manager with four USB serial converters

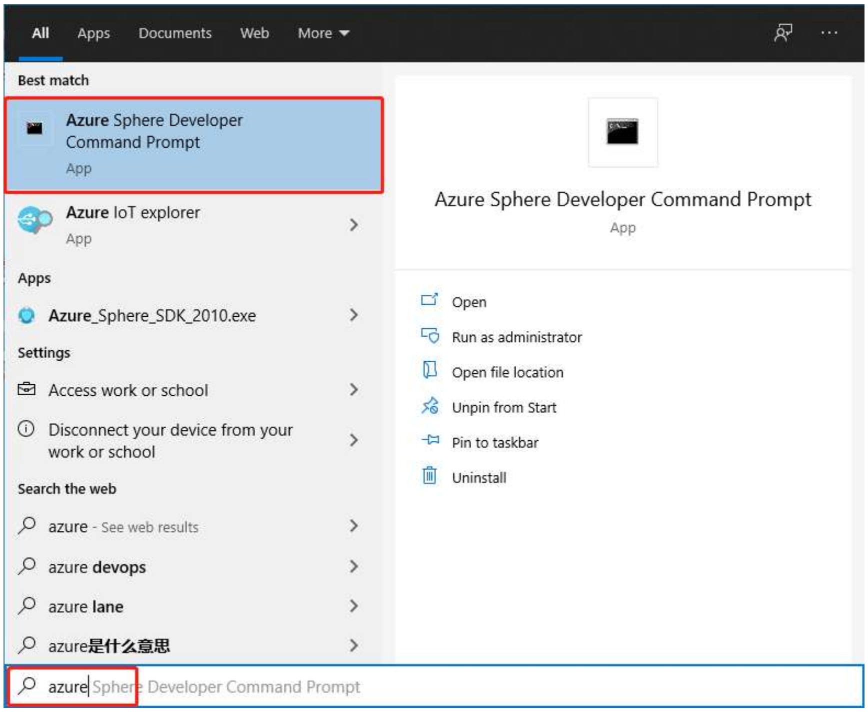


- Monitors
- Network adapters
- Network Infrastructure Devices
- Ports (COM & LPT)
- Print queues
- Printers
- Processors
- Security devices
- Software devices
- Sound, video and game controllers
- Storage controllers
- System devices
- Universal Serial Bus controllers
  - Generic SuperSpeed USB Hub
  - Generic USB Hub
  - Intel(R) USB 3.0 eXtensible Host Controller - 1.0 (Microsoft)
  - Microsoft Mouse and Keyboard Detection Driver (USB)
  - USB Composite Device
  - USB Composite Device
  - USB Root Hub (USB 3.0)
  - USB Serial Converter A
  - USB Serial Converter B
  - USB Serial Converter C
  - USB Serial Converter D
- WSD Print Provider

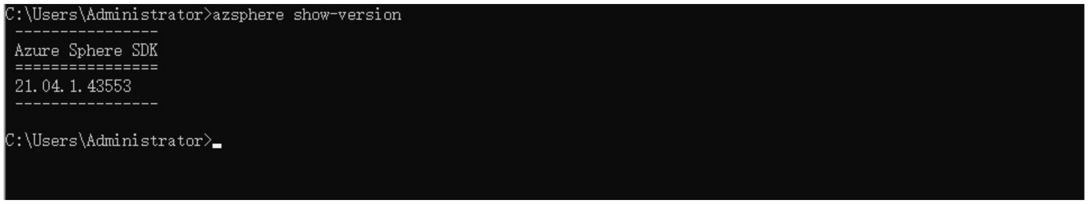
4.2 Install the Azure Sphere SDK

- Download the SDK(https://aka.ms/AzureSphereSDKDownload/). Save the downloaded file on your PC.
- Run the downloaded .exe to install the SDK. Agree to the license terms, and then select Next.
- Select Install to begin installation.
- Accept the elevation prompt if one appears.
- When setup completes, restart your PC if the setup application requests it.

After installing the CLI for the first time, open the CLI through the start menu.



Run ***azsphere show-version*** to check whether it is installed and the correct version is installed.



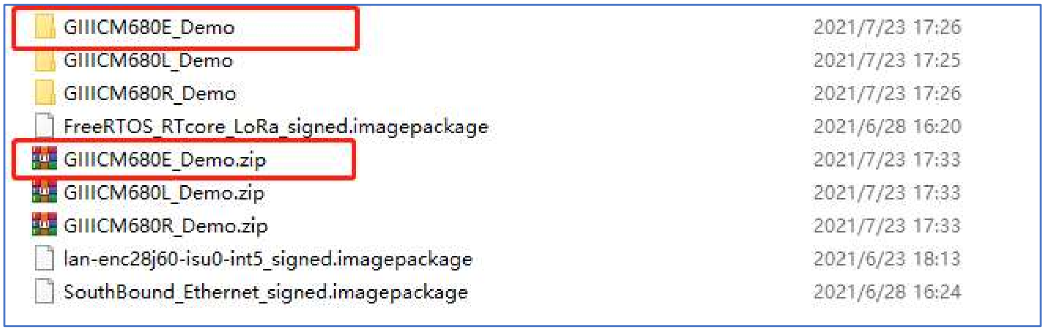
```
C:\Users\Administrator>azsphere show-version
 ----------------
 Azure Sphere SDK
 ================
 21.04.1.43553
 ----------------

C:\Users\Administrator>_
```

4.3 Install Visual Studio and the Azure Sphere extension

- Install Visual Studio, please refer to https://docs.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?preserve-view=true&view=vs-2019.
- To install the Visual Studio extension for Azure Sphere, see https://marketplace.visualstudio.com/items?itemName=AzureSphereTeam.AzureSphereSDKforVisualStudio2019.

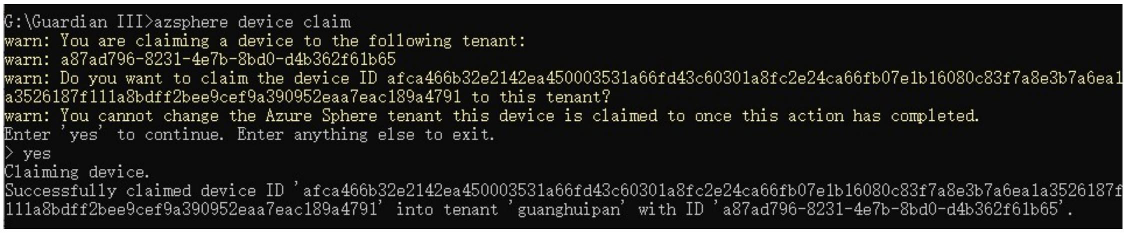
4.4 How to run the sample code

Here is an example of running the GIIICM680E_Demo.zip sample code. Please use the decompression tool to decompress the sample code before running.



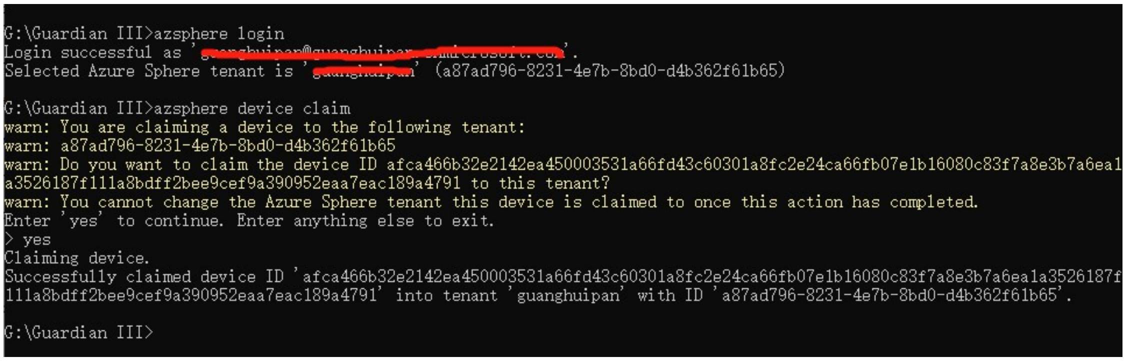
GIIICM680E_Demo	2021/7/23 17:26
GIIICM680L_Demo	2021/7/23 17:25
GIIICM680R_Demo	2021/7/23 17:26
FreeRTOS_RTcore_LoRa_signed.imagepackage	2021/6/28 16:20
GIIICM680E_Demo.zip	2021/7/23 17:33
GIIICM680L_Demo.zip	2021/7/23 17:33
GIIICM680R_Demo.zip	2021/7/23 17:33
lan-enc28j60-isu0-int5_signed.imagepackage	2021/6/23 18:13
SouthBound_Ethernet_signed.imagepackage	2021/6/28 16:24

- Use the "azsphere login" command to log in to your Microsoft account（Each PC only needs to log in once.）



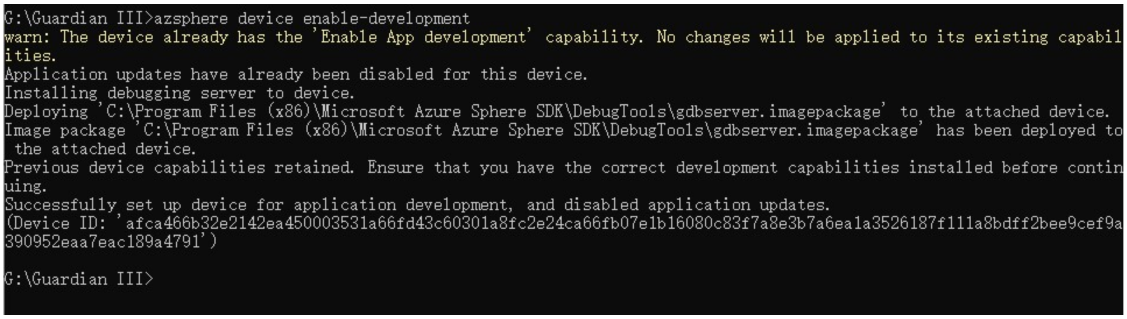
```
G:\Guardian III>azsphere device claim
warn: You are claiming a device to the following tenant:
warn: a87ad796-8231-4e7b-8bd0-d4b362f61b65
warn: Do you want to claim the device ID afca466b32e2142ea450003531a66fd43c60301a8fc2e24ca66fb07e1b16080c83f7a8e3b7a6ea1
a3526187f111a8bdff2bee9cef9a390952eaa7eac189a4791 to this tenant?
warn: You cannot change the Azure Sphere tenant this device is claimed to once this action has completed.
Enter 'yes' to continue. Enter anything else to exit.
> yes
Claiming device.
Successfully claimed device ID 'afca466b32e2142ea450003531a66fd43c60301a8fc2e24ca66fb07e1b16080c83f7a8e3b7a6ea1a3526187f
111a8bdff2bee9cef9a390952eaa7eac189a4791' into tenant 'guanghuipan' with ID 'a87ad796-8231-4e7b-8bd0-d4b362f61b65'.
```

- Declare the device(Each device only needs to be declared once. For more information, please refer to: https://docs.microsoft.com/en-us/azure-sphere/install/claim-device?tabs=cliv2beta.)



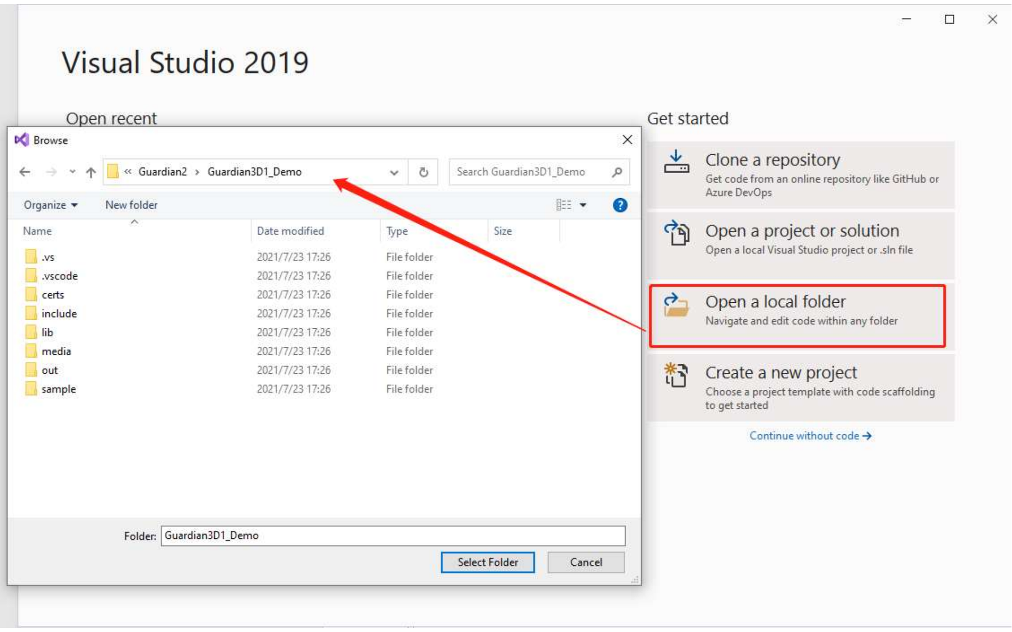
```
G:\Guardian III>azsphere login
Login successful as '[redacted]'.
Selected Azure Sphere tenant is '[redacted]' (a87ad796-8231-4e7b-8bd0-d4b362f61b65)

G:\Guardian III>azsphere device claim
warn: You are claiming a device to the following tenant:
warn: a87ad796-8231-4e7b-8bd0-d4b362f61b65
warn: Do you want to claim the device ID afca466b32e2142ea450003531a66fd43c60301a8fc2e24ca66fb07e1b16080c83f7a8e3b7a6ea1
a3526187f111a8bdff2bee9cef9a390952eaa7eac189a4791 to this tenant?
warn: You cannot change the Azure Sphere tenant this device is claimed to once this action has completed.
Enter 'yes' to continue. Enter anything else to exit.
> yes
Claiming device.
Successfully claimed device ID 'afca466b32e2142ea450003531a66fd43c60301a8fc2e24ca66fb07e1b16080c83f7a8e3b7a6ea1a3526187f
111a8bdff2bee9cef9a390952eaa7eac189a4791' into tenant 'guanghuipan' with ID 'a87ad796-8231-4e7b-8bd0-d4b362f61b65'.

G:\Guardian III>
```

- Enable the developer mode of the device



```
G:\Guardian III>azsphere device enable-development
warn: The device already has the 'Enable App development' capability. No changes will be applied to its existing capabil
ities.
Application updates have already been disabled for this device.
Installing debugging server to device.
Deploying 'C:\Program Files (x86)\Microsoft Azure Sphere SDK\DebugTools\gdbserver.imagepackage' to the attached device.
Image package 'C:\Program Files (x86)\Microsoft Azure Sphere SDK\DebugTools\gdbserver.imagepackage' has been deployed to
 the attached device.
Previous device capabilities retained. Ensure that you have the correct development capabilities installed before contin
uing.
Successfully set up device for application development, and disabled application updates.
(Device ID: 'afca466b32e2142ea450003531a66fd43c60301a8fc2e24ca66fb07e1b16080c83f7a8e3b7a6ea1a3526187f111a8bdff2bee9cef9a
390952eaa7eac189a4791')

G:\Guardian III>
```

- Use Visual Studio to open the sample code.



Visual Studio 2019

Open recent

Get started

Clone a repository
Get code from an online repository like GitHub or Azure DevOps

Open a project or solution
Open a local Visual Studio project or .sln file

Open a local folder
Navigate and edit code within any folder

Create a new project
Choose a project template with code scaffolding to get started

Continue without code →

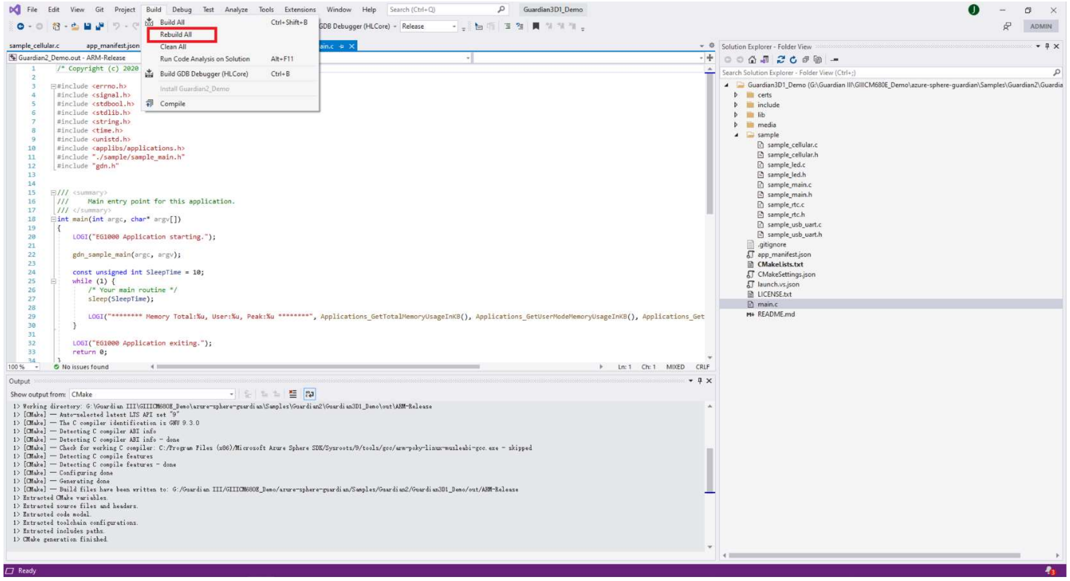
Browse

« Guardian2 › Guardian3D1_Demo

Name	Date modified	Type	Size
.vs	2021/7/23 17:26	File folder	
.vscode	2021/7/23 17:26	File folder	
certs	2021/7/23 17:26	File folder	
include	2021/7/23 17:26	File folder	
lib	2021/7/23 17:26	File folder	
media	2021/7/23 17:26	File folder	
out	2021/7/23 17:26	File folder	
sample	2021/7/23 17:26	File folder	

Folder: Guardian3D1_Demo

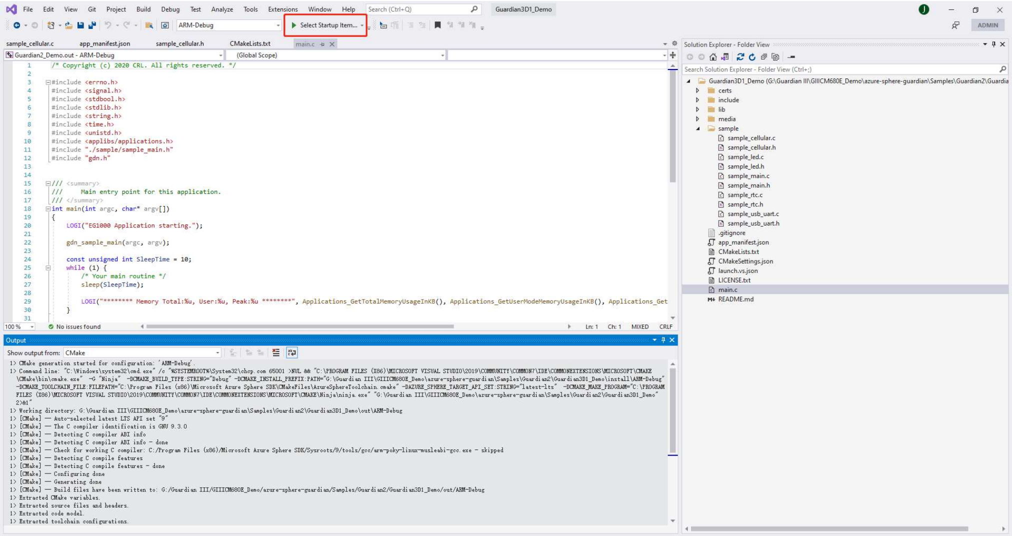
- Compile the sample code.



```
/* Copyright (c) 2020

#include <errno.h>
#include <signal.h>
#include <stdbool.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <unistd.h>
#include <applibs/applications.h>
#include "./sample/sample_main.h"
#include "gdn.h"


/// <summary>
///     Main entry point for this application.
/// </summary>
int main(int argc, char* argv[])
{
    LOGI("EG1000 Application starting.");

    gdn_sample_main(argc, argv);

    const unsigned int SleepTime = 10;
    while (1) {
        /* Your main routine */
        sleep(SleepTime);

        LOGI("******** Memory Total:%u, User:%u, Peak:%u ********", Applications_GetTotalMemoryUsageInKB(), Applications_GetUserModeMemoryUsageInKB(), Applications_Get
    }

    LOGI("EG1000 Application exiting.");
    return 0;
}
```

```
1> Working directory: G:\Guardian III\GIIICM680E_Demo\azure-sphere-guardian\Samples\Guardian2\Guardian3D1_Demo\out\ARM-Release
1> [CMake] -- Auto-selected latest LTS API set "9"
1> [CMake] -- The C compiler identification is GNU 9.3.0
1> [CMake] -- Detecting C compiler ABI info
1> [CMake] -- Detecting C compiler ABI info - done
1> [CMake] -- Check for working C compiler: C:/Program Files (x86)/Microsoft Azure Sphere SDK/Sysroots/9/tools/gcc/arm-poky-linux-musleabi-gcc.exe - skipped
1> [CMake] -- Detecting C compile features
1> [CMake] -- Detecting C compile features - done
1> [CMake] -- Configuring done
1> [CMake] -- Generating done
1> [CMake] -- Build files have been written to: G:/Guardian III/GIIICM680E_Demo/azure-sphere-guardian/Samples/Guardian2/Guardian3D1_Demo/out/ARM-Release
1> Extracted CMake variables.
1> Extracted source files and headers.
1> Extracted code model.
1> Extracted toolchain configurations.
1> Extracted includes paths.
1> CMake generation finished.
```

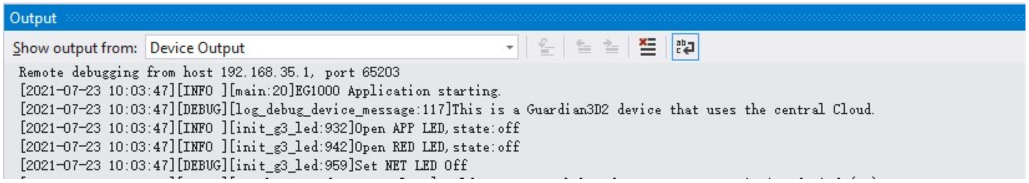
- Run the sample code.



```c
/* Copyright (c) 2020 CRL. All rights reserved. */

#include <errno.h>
#include <signal.h>
#include <stdbool.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <unistd.h>
#include <applibs/applications.h>
#include "./sample/sample_main.h"
#include "gdn.h"


/// <summary>
///     Main entry point for this application.
/// </summary>
int main(int argc, char* argv[])
{
    LOGI("EG1000 Application starting.");

    gdn_sample_main(argc, argv);

    const unsigned int SleepTime = 10;
    while (1) {
        /* Your main routine */
        sleep(SleepTime);

        LOGI("******** Memory Total:%u, User:%u, Peak:%u ********", Applications_GetTotalMemoryUsageInKB(), Applications_GetUserModeMemoryUsageInKB(), Applications_Get
    }

```

```
1> CMake generation started for configuration: 'ARM-Debug'.
1> Command line: "C:\Windows\system32\cmd.exe" /c "%SYSTEMROOT%\System32\chcp.com 65001 >NUL && "C:\PROGRAM FILES (X86)\MICROSOFT VISUAL STUDIO\2019\COMMUNITY\COMMON7\IDE\COMMONEXTENSIONS\MICROSOFT\CMAKE
  \CMake\bin\cmake.exe"  -G "Ninja"  -DCMAKE_BUILD_TYPE:STRING="Debug" -DCMAKE_INSTALL_PREFIX:PATH="G:\Guardian III\GIIICM680E_Demo\azure-sphere-guardian\Samples\Guardian2\Guardian3D1_Demo\install\ARM-Debug"
  -DCMAKE_TOOLCHAIN_FILE:FILEPATH="C:\Program Files (x86)\Microsoft Azure Sphere SDK\CMakeFiles\AzureSphereToolchain.cmake" -DAZURE_SPHERE_TARGET_API_SET:STRING="latest-lts"  -DCMAKE_MAKE_PROGRAM="C:\PROGRAM
  FILES (X86)\MICROSOFT VISUAL STUDIO\2019\COMMUNITY\COMMON7\IDE\COMMONEXTENSIONS\MICROSOFT\CMAKE\Ninja\ninja.exe" "G:\Guardian III\GIIICM680E_Demo\azure-sphere-guardian\Samples\Guardian2\Guardian3D1_Demo"
  2>&1"
1> Working directory: G:\Guardian III\GIIICM680E_Demo\azure-sphere-guardian\Samples\Guardian2\Guardian3D1_Demo\out\ARM-Debug
1> [CMake] — Auto-selected latest LTS API set "9"
1> [CMake] — The C compiler identification is GNU 9.3.0
1> [CMake] — Detecting C compiler ABI info
1> [CMake] — Detecting C compiler ABI info - done
1> [CMake] — Check for working C compiler: C:/Program Files (x86)/Microsoft Azure Sphere SDK/Sysroots/9/tools/gcc/arm-poky-linux-musleabi-gcc.exe - skipped
1> [CMake] — Detecting C compile features
1> [CMake] — Detecting C compile features - done
1> [CMake] — Configuring done
1> [CMake] — Generating done
1> [CMake] — Build files have been written to: G:/Guardian III/GIIICM680E_Demo/azure-sphere-guardian/Samples/Guardian2/Guardian3D1_Demo/out/ARM-Debug
1> Extracted CMake variables.
1> Extracted source files and headers.
1> Extracted code model.
1> Extracted toolchain configurations.
```

When the following log print is entered in the log output window, it indicates that the sample code is running normally.



```
Remote debugging from host 192.168.35.1, port 65203
[2021-07-23 10:03:47][INFO ][main:20]EG1000 Application starting.
[2021-07-23 10:03:47][DEBUG][log_debug_device_message:117]This is a Guardian3D2 device that uses the central Cloud.
[2021-07-23 10:03:47][INFO ][init_g3_led:932]Open APP LED, state:off
[2021-07-23 10:03:47][INFO ][init_g3_led:942]Open RED LED, state:off
[2021-07-23 10:03:47][DEBUG][init_g3_led:959]Set NET LED Off
```

5 Sample and Test

In order to demonstrate the use of Guardian III series products, we provide basic sample codes and test methods for each product.

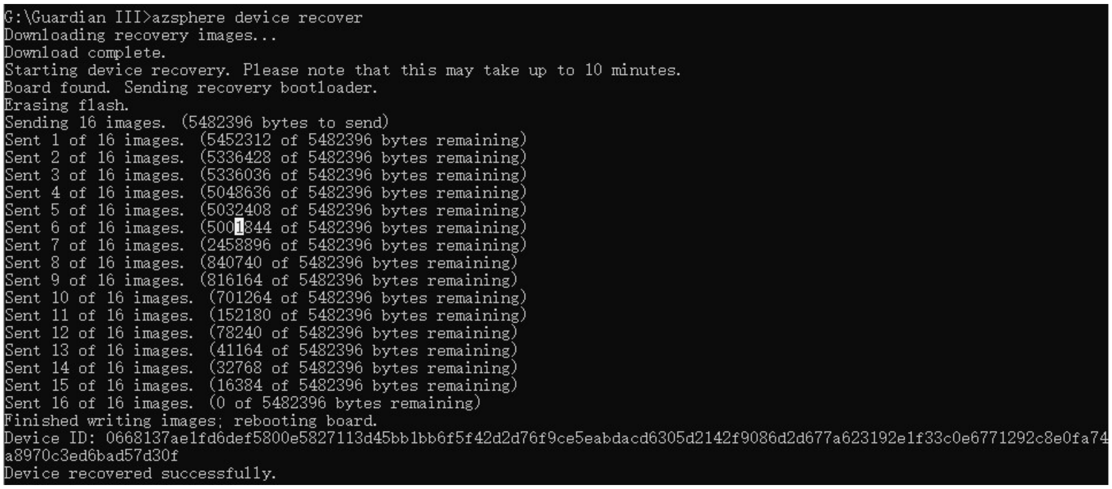
5.1 GIIICM680L

The sample code of GIIICM680L includes LEDs, LoRa, UART demo.
Package list:
FreeRTOS_RTcore_LoRa_signed.imagepackage
lan-enc28j60-isu0-int5_signed.imagepackage
GIIICM680L_Demo.zip

5.1.1 Run the sample code

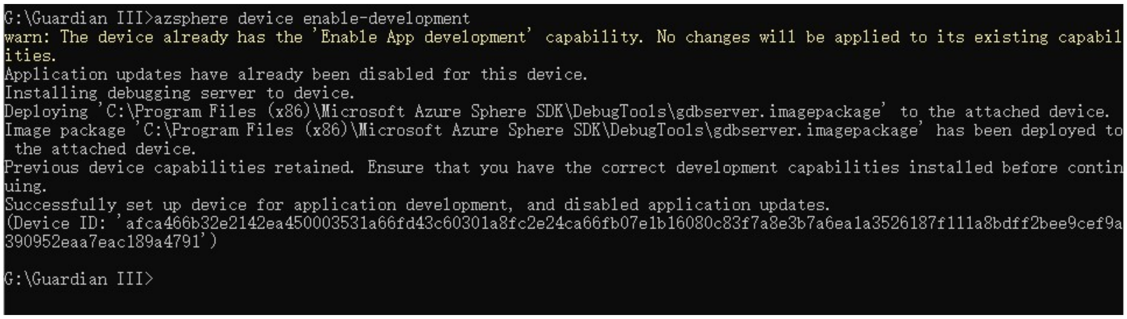
1. Equipment Recover;

```
azsphere device recover
```



```
G:\Guardian III>azsphere device recover
Downloading recovery images...
Download complete.
Starting device recovery. Please note that this may take up to 10 minutes.
Board found. Sending recovery bootloader.
Erasing flash.
Sending 16 images. (5482396 bytes to send)
Sent 1 of 16 images.  (5452312 of 5482396 bytes remaining)
Sent 2 of 16 images.  (5336428 of 5482396 bytes remaining)
Sent 3 of 16 images.  (5336036 of 5482396 bytes remaining)
Sent 4 of 16 images.  (5048636 of 5482396 bytes remaining)
Sent 5 of 16 images.  (5032408 of 5482396 bytes remaining)
Sent 6 of 16 images.  (5001844 of 5482396 bytes remaining)
Sent 7 of 16 images.  (2458896 of 5482396 bytes remaining)
Sent 8 of 16 images.  (840740 of 5482396 bytes remaining)
Sent 9 of 16 images.  (816164 of 5482396 bytes remaining)
Sent 10 of 16 images.  (701264 of 5482396 bytes remaining)
Sent 11 of 16 images.  (152180 of 5482396 bytes remaining)
Sent 12 of 16 images.  (78240 of 5482396 bytes remaining)
Sent 13 of 16 images.  (41164 of 5482396 bytes remaining)
Sent 14 of 16 images.  (32768 of 5482396 bytes remaining)
Sent 15 of 16 images.  (16384 of 5482396 bytes remaining)
Sent 16 of 16 images.  (0 of 5482396 bytes remaining)
Finished writing images; rebooting board.
Device ID: 0668137ae1fd6def5800e5827113d45bb1bb6f5f42d2d76f9ce5eabdacd6305d2142f9086d2d677a623192e1f33c0e6771292c8e0fa74
a8970c3ed6bad57d30f
Device recovered successfully.
```

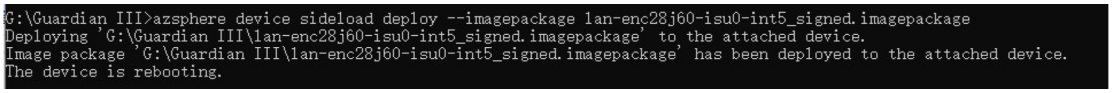
2. Enable the developer mode of the device



```
G:\Guardian III>azsphere device enable-development
warn: The device already has the 'Enable App development' capability. No changes will be applied to its existing capabil
ities.
Application updates have already been disabled for this device.
Installing debugging server to device.
Deploying 'C:\Program Files (x86)\Microsoft Azure Sphere SDK\DebugTools\gdbserver.imagepackage' to the attached device.
Image package 'C:\Program Files (x86)\Microsoft Azure Sphere SDK\DebugTools\gdbserver.imagepackage' has been deployed to
 the attached device.
Previous device capabilities retained. Ensure that you have the correct development capabilities installed before contin
uing.
Successfully set up device for application development, and disabled application updates.
(Device ID: 'afca466b32e2142ea450003531a66fd43c60301a8fc2e24ca66fb07e1b16080c83f7a8e3b7a6ea1a3526187f111a8bdff2bee9cef9a
390952eaa7eac189a4791')

G:\Guardian III>
```

3. Download the internal network card driver package;

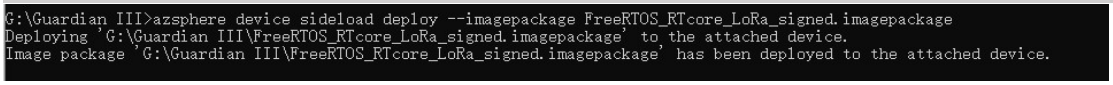
```
azsphere device sideload deploy --imagepackage lan-enc28j60-isu0-int5_signed.imagepackage
```



```
G:\Guardian III>azsphere device sideload deploy --imagepackage lan-enc28j60-isu0-int5_signed.imagepackage
Deploying 'G:\Guardian III\lan-enc28j60-isu0-int5_signed.imagepackage’ to the attached device.
Image package 'G:\Guardian III\lan-enc28j60-isu0-int5_signed.imagepackage’ has been deployed to the attached device.
The device is rebooting.
```

4. Download the LoRa driver package;

```
azsphere device sideload deploy --imagepackage FreeRTOS_RTcore_LoRa_signed.imagepackage
```



```
G:\Guardian III>azsphere device sideload deploy --imagepackage FreeRTOS_RTcore_LoRa_signed.imagepackage
Deploying 'G:\Guardian III\FreeRTOS_RTcore_LoRa_signed.imagepackage’ to the attached device.
Image package 'G:\Guardian III\FreeRTOS_RTcore_LoRa_signed.imagepackage’ has been deployed to the attached device.
```

5. Run the sample code “GIIICM680L_Demo.zip”; (Refer to section 4.4)

5.1.2 APP LEDs

After the software package is installed, restart the device, and APP LED1 and APP LED2 will flash alternately in accordance with "green" -> "red" -> "off". as figure 23 show.

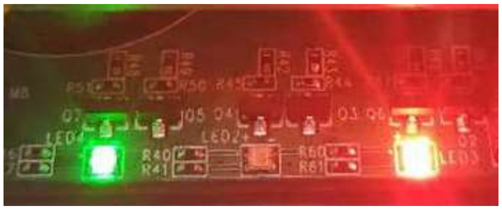


Figure 23: LED status

5.1.3 Test Lora

In order to complete the Lora test, we need to introduce the third LoRa test module, which mainly completes the conversion of LoRa data and USB serial port data in order to complete the data transmission and reception test. The test module needs to configure the following parameters:

frequency:	910.3M
bandwidth:	125K
spreading factor:	7
coding rate:	4/5
CRC enabled:	true
preamble length:	8bytes
power:	20dbm
invert IQ:	true

After using the serial port tool putty to connect the test module, the string sent by putty is sent to the device, and the device will return the data as it is after receiving the data, so you can see in putty that you are sending data.

5.1.4 Test UART

Use Type-B to connect the PC and the device. After the connection is complete, open the serial port through the serial port tool putty.

Type-B cable:

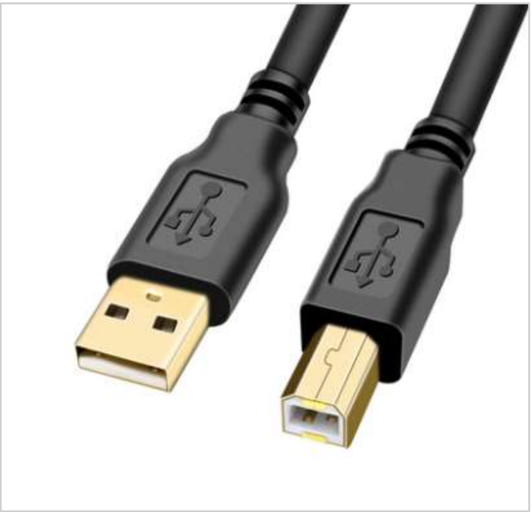
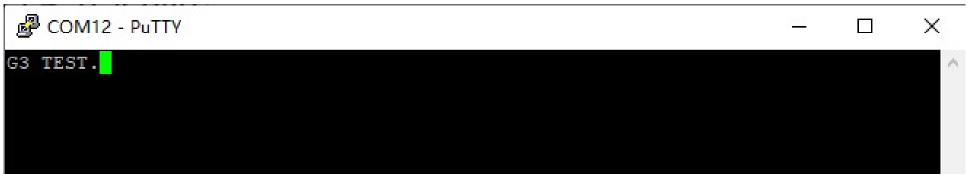


Figure 24: Type-B cable

Serial port connection parameters:

Speed(baud):	115200bps
Data bits:	8
Stop bits:	1
Parity:	None
Flow control:	None

After connecting the device with the serial port tool putty, the string sent by putty is sent to the device, and the device will return the data as it is after receiving the data, so you can see in putty that you are sending data.



```
G3 TEST.
```

Figure 25: putty output information

5.2 GIIICM680E

The sample code of GIIICM680E includes LEDs, W5500, UART demo.

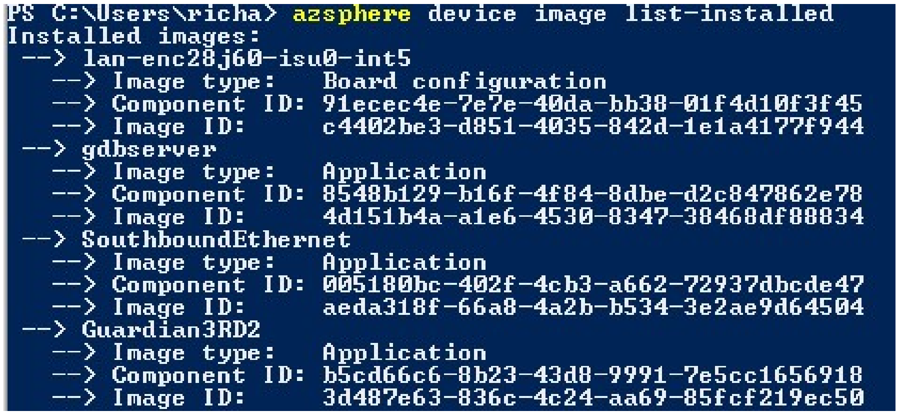
Package list:

SouthBound_Ethernet_signed.imagepackage
lan-enc28j60-isu0-int5_signed.imagepackage
GIIICM680E_Demo.zip

5.2.1 Run the sample code

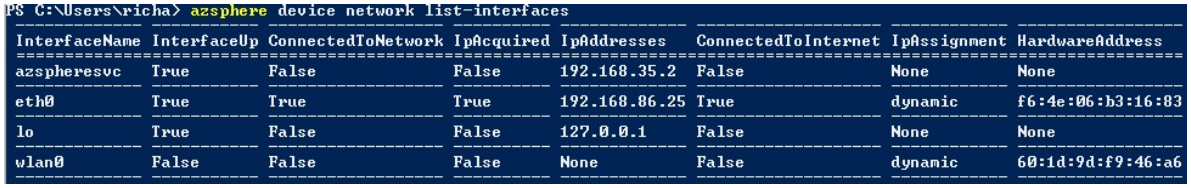
If you get a box with sample code loaded, do the following to test it:

1. Connect the debug dongle and power on the box, from the PowerShell of windows, check images installed, it should be like this:



```
PS C:\Users\richa> azsphere device image list-installed
Installed images:
 --> lan-enc28j60-isu0-int5
   --> Image type:   Board configuration
   --> Component ID: 91ecec4e-7e7e-40da-bb38-01f4d10f3f45
   --> Image ID:     c4402be3-d851-4035-842d-1e1a4177f944
 --> gdbserver
   --> Image type:   Application
   --> Component ID: 8548b129-b16f-4f84-8dbe-d2c847862e78
   --> Image ID:     4d151b4a-a1e6-4530-8347-38468df88834
 --> SouthboundEthernet
   --> Image type:   Application
   --> Component ID: 005180bc-402f-4cb3-a662-72937dbcde47
   --> Image ID:     aeda318f-66a8-4a2b-b534-3e2ae9d64504
 --> Guardian3RD2
   --> Image type:   Application
   --> Component ID: b5cd66c6-8b23-43d8-9991-7e5cc1656918
   --> Image ID:     3d487e63-836c-4c24-aa69-85fcf219ec50
```

2. Check interface connection:



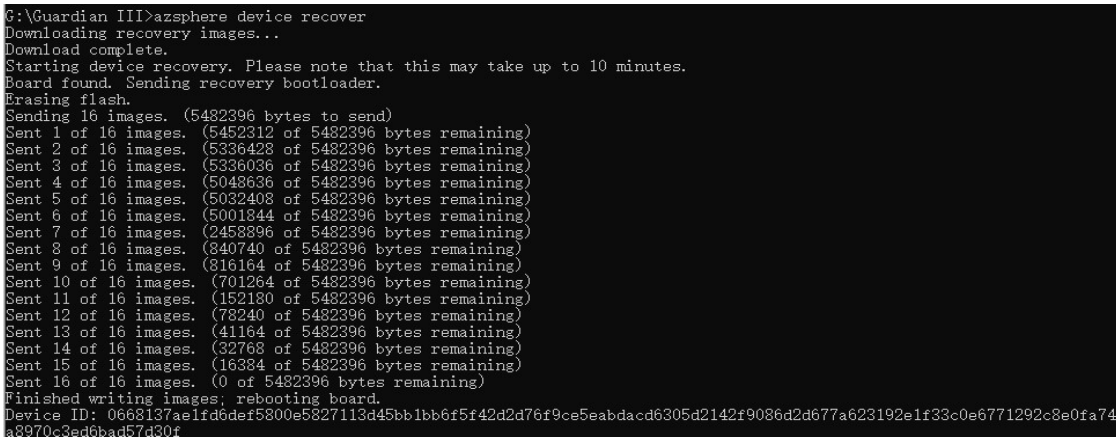
```
PS C:\Users\richa> azsphere device network list-interfaces
 ------------- ----------- ------------------ ---------- ------------- ------------------- ------------ -----------------
 InterfaceName InterfaceUp ConnectedToNetwork IpAcquired IpAddresses   ConnectedToInternet IpAssignment HardwareAddress
 ======================================================================================================================
 azspheresvc   True        False              False      192.168.35.2  False               None         None
 ------------- ----------- ------------------ ---------- ------------- ------------------- ------------ -----------------
 eth0          True        True               True       192.168.86.25 True                dynamic      f6:4e:06:b3:16:83
 ------------- ----------- ------------------ ---------- ------------- ------------------- ------------ -----------------
 lo            True        False              False      127.0.0.1     False               None         None
 ------------- ----------- ------------------ ---------- ------------- ------------------- ------------ -----------------
 wlan0         False       False              False      None          False               dynamic      60:1d:9d:f9:46:a6
 ------------- ----------- ------------------ ---------- ------------- ------------------- ------------ -----------------
```

3. Use the mobile App to watch southbound connection status. Refer 5.2.3 for the test setup.

If you get a brand-new box without code loaded, do the following to install test samples:

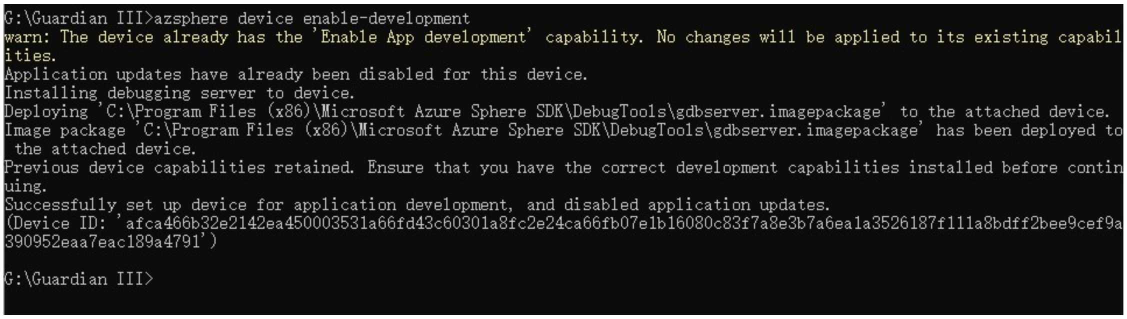
1. Equipment Recover.

```
azsphere device recover
```



```
G:\Guardian III>azsphere device recover
Downloading recovery images...
Download complete.
Starting device recovery. Please note that this may take up to 10 minutes.
Board found. Sending recovery bootloader.
Erasing flash.
Sending 16 images. (5482396 bytes to send)
Sent 1 of 16 images. (5452312 of 5482396 bytes remaining)
Sent 2 of 16 images. (5336428 of 5482396 bytes remaining)
Sent 3 of 16 images. (5336036 of 5482396 bytes remaining)
Sent 4 of 16 images. (5048636 of 5482396 bytes remaining)
Sent 5 of 16 images. (5032408 of 5482396 bytes remaining)
Sent 6 of 16 images. (5001844 of 5482396 bytes remaining)
Sent 7 of 16 images. (2458896 of 5482396 bytes remaining)
Sent 8 of 16 images. (840740 of 5482396 bytes remaining)
Sent 9 of 16 images. (816164 of 5482396 bytes remaining)
Sent 10 of 16 images. (701264 of 5482396 bytes remaining)
Sent 11 of 16 images. (152180 of 5482396 bytes remaining)
Sent 12 of 16 images. (78240 of 5482396 bytes remaining)
Sent 13 of 16 images. (41164 of 5482396 bytes remaining)
Sent 14 of 16 images. (32768 of 5482396 bytes remaining)
Sent 15 of 16 images. (16384 of 5482396 bytes remaining)
Sent 16 of 16 images. (0 of 5482396 bytes remaining)
Finished writing images; rebooting board.
Device ID: 0668137ae1fd6def5800e5827113d45bb1bb6f5f42d2d76f9ce5eabdacd6305d2142f9086d2d677a623192e1f33c0e6771292c8e0fa74
a8970c3ed6bad57d30f
```

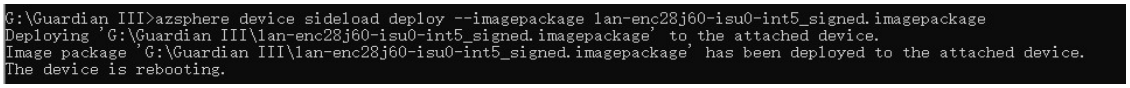
2. Enable the developer mode of the device



```
G:\Guardian III>azsphere device enable-development
warn: The device already has the 'Enable App development' capability. No changes will be applied to its existing capabil
ities.
Application updates have already been disabled for this device.
Installing debugging server to device.
Deploying 'C:\Program Files (x86)\Microsoft Azure Sphere SDK\DebugTools\gdbserver.imagepackage' to the attached device.
Image package 'C:\Program Files (x86)\Microsoft Azure Sphere SDK\DebugTools\gdbserver.imagepackage' has been deployed to
 the attached device.
Previous device capabilities retained. Ensure that you have the correct development capabilities installed before contin
uing.
Successfully set up device for application development, and disabled application updates.
(Device ID: 'afca466b32e2142ea450003531a66fd43c60301a8fc2e24ca66fb07e1b16080c83f7a8e3b7a6ea1a3526187f111a8bdff2bee9cef9a
390952eaa7eac189a4791')

G:\Guardian III>
```

3. Download the internal network card driver package;

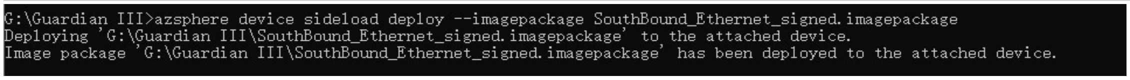
```
azsphere device sideload deploy --imagepackage lan-enc28j60-isu0-int5_signed.imagepackage
```



```
G:\Guardian III>azsphere device sideload deploy --imagepackage lan-enc28j60-isu0-int5_signed.imagepackage
Deploying 'G:\Guardian III\lan-enc28j60-isu0-int5_signed.imagepackage' to the attached device.
Image package 'G:\Guardian III\lan-enc28j60-isu0-int5_signed.imagepackage' has been deployed to the attached device.
The device is rebooting.
```

4. Download the W5500 driver package;

```
azsphere device sideload deploy --imagepackage SouthBound_Ethernet_signed.imagepackage
```



```
G:\Guardian III>azsphere device sideload deploy --imagepackage SouthBound_Ethernet_signed.imagepackage
Deploying 'G:\Guardian III\SouthBound_Ethernet_signed.imagepackage' to the attached device.
Image package 'G:\Guardian III\SouthBound_Ethernet_signed.imagepackage' has been deployed to the attached device.
```

5. Run the sample code “GIIICM680E_Demo.zip”;　(Refer to section 4.4)

5.2.2 APP LEDs

After the software package is installed, restart the device, and APP LED1 and APP LED2 will flash alternately in accordance with "green" -> "red" -> "off". as figure 26 show.

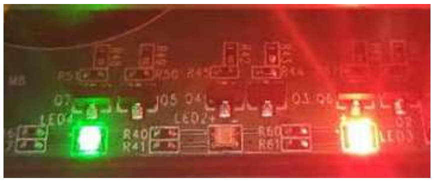
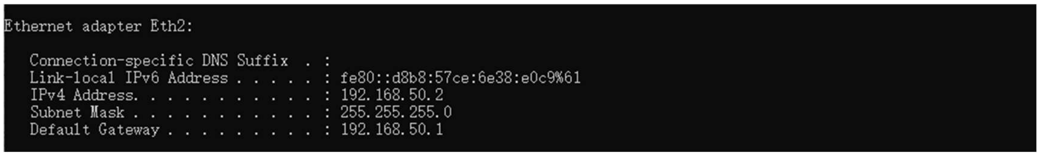


Figure 26: LED status

5.2.3 Test southbound Ethernet (W5500)

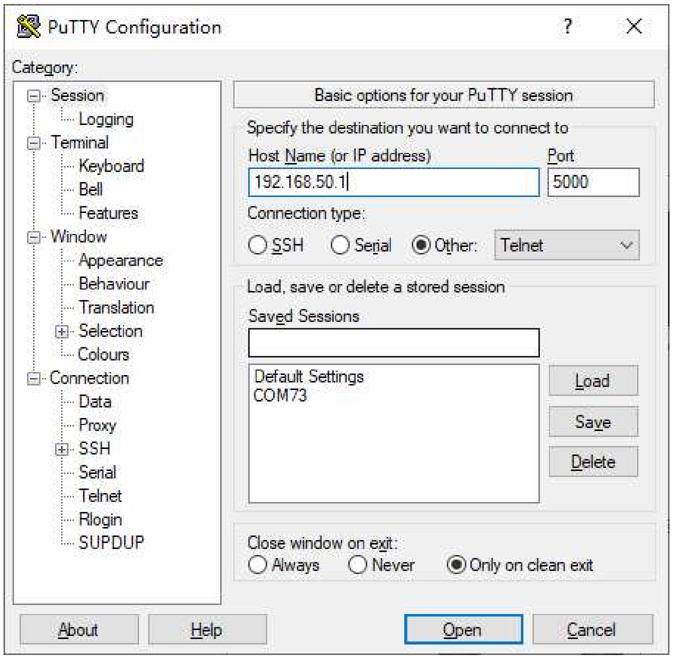
Use the RJ-45 network cable to connect the device and the PC, and set the PC to DHCP client mode. After the connection is completed, W5500 will assign a 192.168.50.X network segment ip to the PC, and the default gateway address is 192.168.50.1.. as figure 27 show.



```
Ethernet adapter Eth2:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::d8b8:57ce:6e38:e0c9%61
   IPv4 Address. . . . . . . . . . . : 192.168.50.2
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.50.1
```

Figure 27: Get Client IP

The W5500 internal test module starts TCP port 5000 at 192.168.50.1 by default, and uses putty to connect.



Specify the destination you want to connect to

Host Name (or IP address): 192.168.50.1

Port: 5000

Connection type:

○ SSH ○ Serial ● Other: Telnet

Load, save or delete a stored session

Saved Sessions

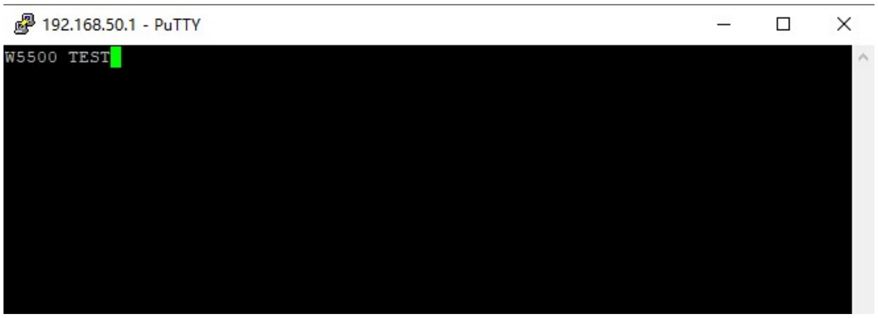
Default Settings
COM73

Close window on exit:

○ Always ○ Never ● Only on clean exit

Figure 28: putty configuration parameters

Any data received by the W5500 internal test module will be returned as it is.



```
W5500 TEST
```

Figure 29: putty output information

5.2.4 Test UART

Use Type-B to connect the PC and the device. After the connection is complete, open the serial port through the serial port tool putty.

Type-B cable:

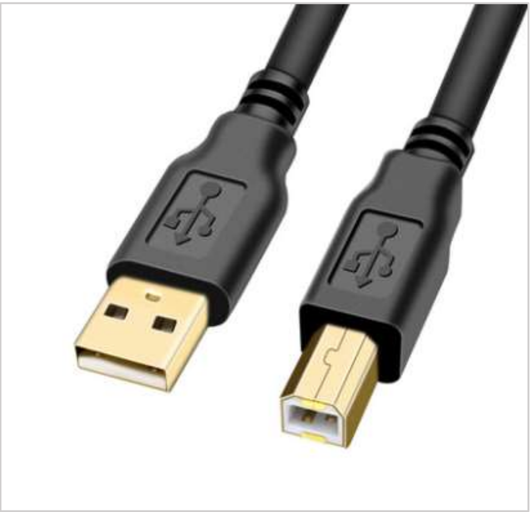
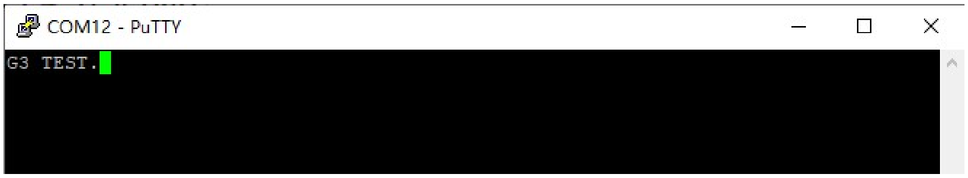


Figure 30: Type-B cable

Serial port connection parameters:

Speed(baud):	115200
Data bits:	8
Stop bits:	1
Parity:	None
Flow control:	None

After connecting the device with the serial port tool putty, the string sent by putty is sent to the device, and the device will return the data as it is after receiving the data, so you can see in putty that you are sending data.



```
G3 TEST.
```

Figure 31: putty output information

5.3 GIIICM680R

The sample code of GIIICM680R includes LEDs, CAN, RS422, RS485 demonstration.

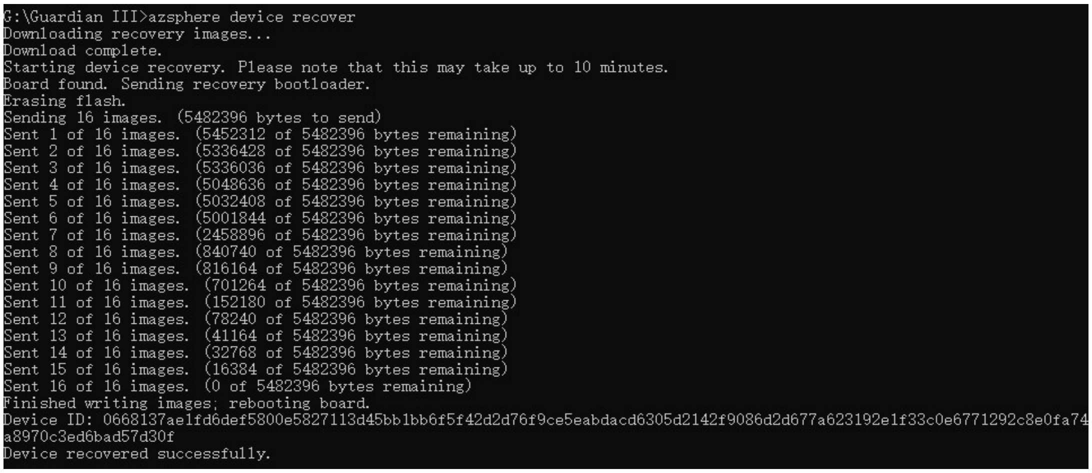
Package list:

lan-enc28j60-isu0-int5_signed.imagepackage
GIIICM680R_Demo.zip

5.3.1 Run the sample code

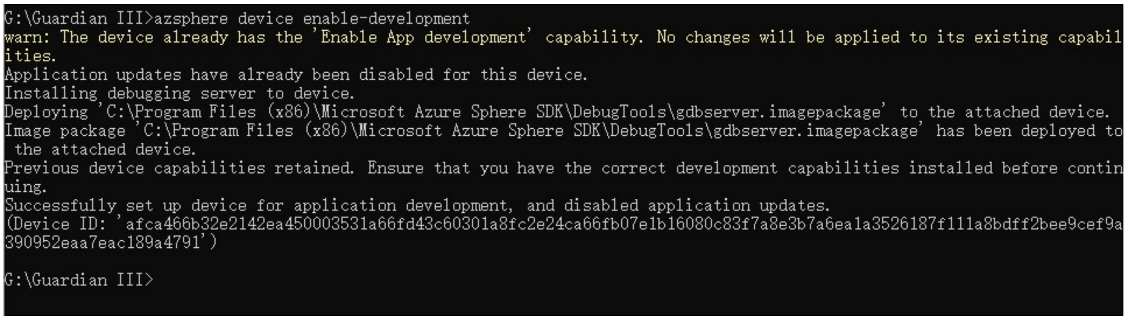
1. Equipment Recover;

```
azsphere device recover
```



```
G:\Guardian III>azsphere device recover
Downloading recovery images...
Download complete.
Starting device recovery. Please note that this may take up to 10 minutes.
Board found. Sending recovery bootloader.
Erasing flash.
Sending 16 images. (5482396 bytes to send)
Sent 1 of 16 images. (5452312 of 5482396 bytes remaining)
Sent 2 of 16 images. (5336428 of 5482396 bytes remaining)
Sent 3 of 16 images. (5336036 of 5482396 bytes remaining)
Sent 4 of 16 images. (5048636 of 5482396 bytes remaining)
Sent 5 of 16 images. (5032408 of 5482396 bytes remaining)
Sent 6 of 16 images. (5001844 of 5482396 bytes remaining)
Sent 7 of 16 images. (2458896 of 5482396 bytes remaining)
Sent 8 of 16 images. (840740 of 5482396 bytes remaining)
Sent 9 of 16 images. (816164 of 5482396 bytes remaining)
Sent 10 of 16 images. (701264 of 5482396 bytes remaining)
Sent 11 of 16 images. (152180 of 5482396 bytes remaining)
Sent 12 of 16 images. (78240 of 5482396 bytes remaining)
Sent 13 of 16 images. (41164 of 5482396 bytes remaining)
Sent 14 of 16 images. (32768 of 5482396 bytes remaining)
Sent 15 of 16 images. (16384 of 5482396 bytes remaining)
Sent 16 of 16 images. (0 of 5482396 bytes remaining)
Finished writing images; rebooting board.
Device ID: 0668137ae1fd6def5800e5827113d45bb1bb6f5f42d2d76f9ce5eabdacd6305d2142f9086d2d677a623192e1f33c0e6771292c8e0fa74
a8970c3ed6bad57d30f
Device recovered successfully.
```

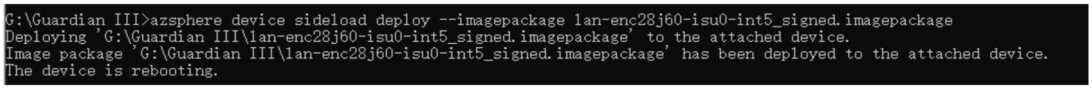
2. Enable the developer mode of the device



```
G:\Guardian III>azsphere device enable-development
warn: The device already has the 'Enable App development' capability. No changes will be applied to its existing capabil
ities.
Application updates have already been disabled for this device.
Installing debugging server to device.
Deploying 'C:\Program Files (x86)\Microsoft Azure Sphere SDK\DebugTools\gdbserver.imagepackage' to the attached device.
Image package 'C:\Program Files (x86)\Microsoft Azure Sphere SDK\DebugTools\gdbserver.imagepackage' has been deployed to
 the attached device.
Previous device capabilities retained. Ensure that you have the correct development capabilities installed before contin
uing.
Successfully set up device for application development, and disabled application updates.
(Device ID: 'afca466b32e2142ea450003531a66fd43c60301a8fc2e24ca66fb07e1b16080c83f7a8e3b7a6ea1a3526187f111a8bdff2bee9cef9a
390952eaa7eac189a4791')

G:\Guardian III>
```

3. Download the internal network card driver package;

```
azsphere device sideload deploy --imagepackage lan-enc28j60-isu0-int5_signed.imagepackage
```



```
G:\Guardian III>azsphere device sideload deploy --imagepackage lan-enc28j60-isu0-int5_signed.imagepackage
Deploying 'G:\Guardian III\lan-enc28j60-isu0-int5_signed.imagepackage' to the attached device.
Image package 'G:\Guardian III\lan-enc28j60-isu0-int5_signed.imagepackage' has been deployed to the attached device.
The device is rebooting.
```

Run the sample code "GIIICM680R_Demo.zip"; (Refer to section 4.4)

5.3.2 APP LEDs

After the software package is installed, restart the device, and APP LED1 and APP LED2 will flash alternately in accordance with "green" -> "red" -> "off". as figure 32 show.

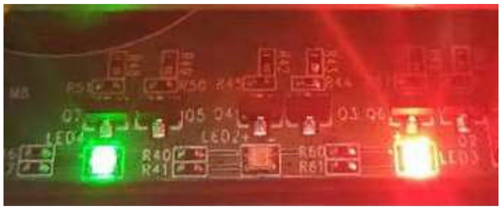


Figure 32: LED status

5.3.3 Test CAN

Use a third-party CAN test tool to connect the device. The CAN test module used in this example is a CAN module produced by China Chuangxin Technology.

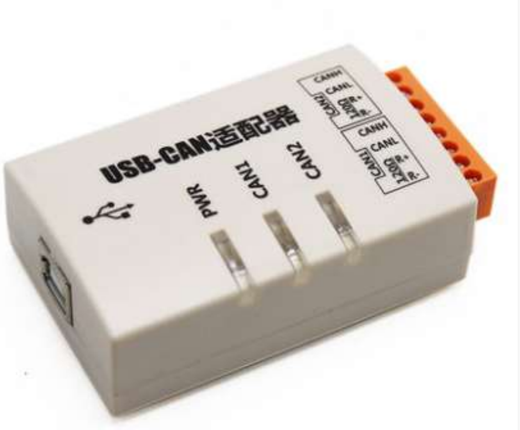
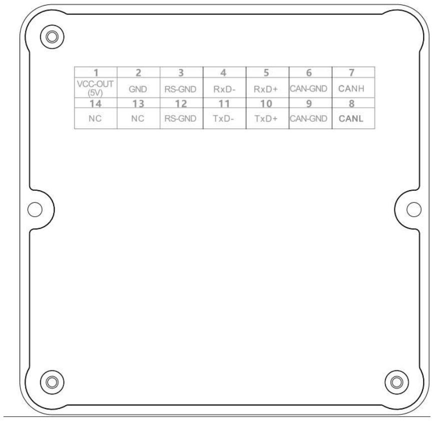


Figure 33: CAN test module Appearance



1	2	3	4	5	6	7
VCC-OUT (5V)	GND	RS-GND	RxD-	RxD+	CAN-GND	CANH
14	13	12	11	10	9	8
NC	NC	RS-GND	TxD-	TxD+	CAN-GND	CANL

Figure 34: Device GPIOs

Use CAN to connect the device, connect the wire sequence GPIO7---CHNH, GPIO8---CHNL.

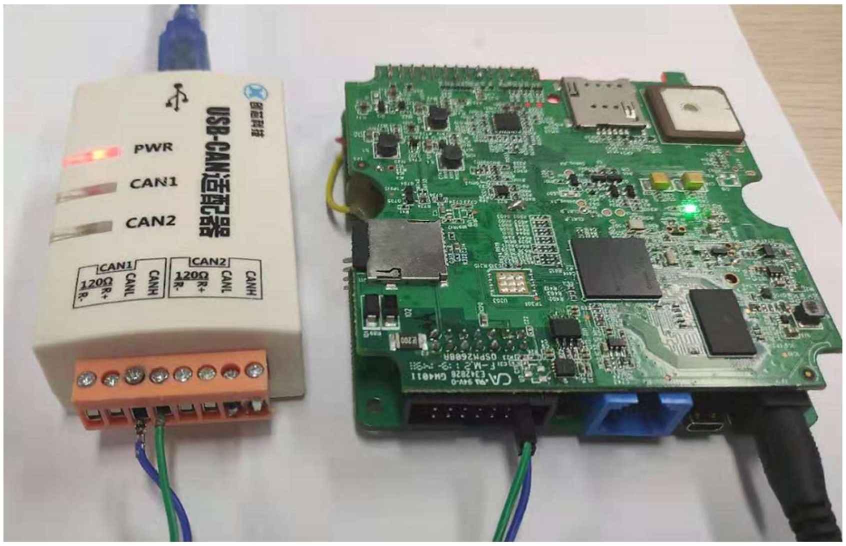
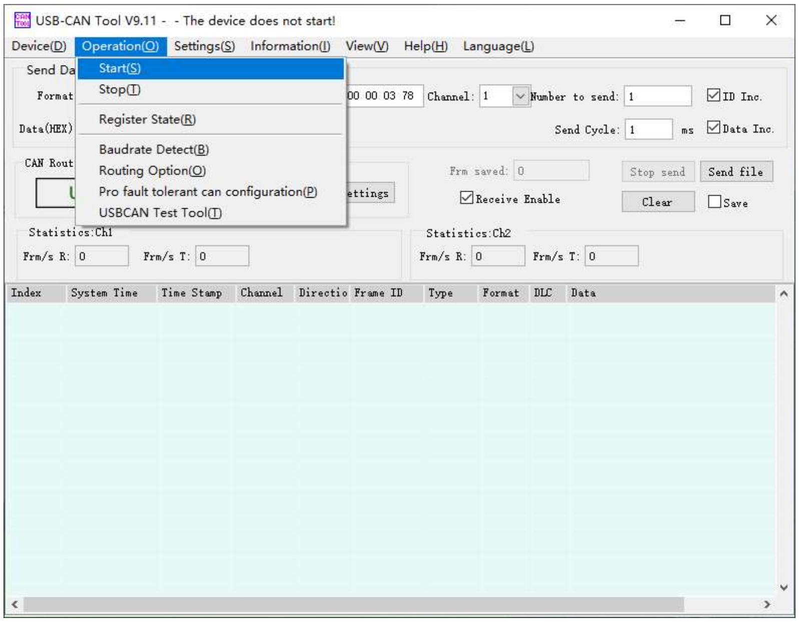


Figure 35: CAN connection diagram

After the connection is complete, use the test tool provided with the CAN test module to turn on the device.



USB-CAN Tool V9.11 - - The device does not start!

Send Da | Format | Data(HEX) | 00 00 03 78 | Channel: 1 | Number to send: 1 | ☑ID Inc. | Send Cycle: 1 ms | ☑Data Inc.

CAN Rout

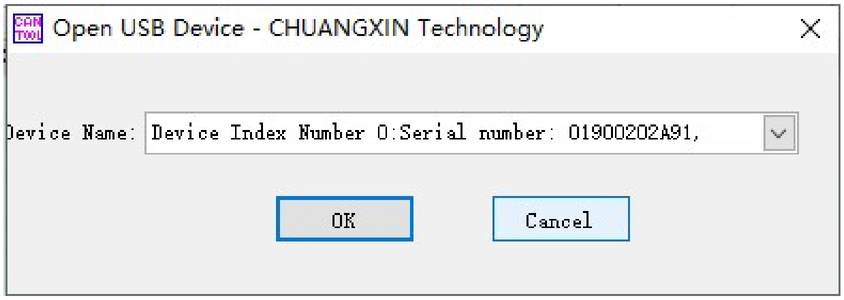
Frm saved: 0 | ☑Receive Enable | □Save

Statistics:Ch1 | Frm/s R: 0 | Frm/s T: 0

Statistics:Ch2 | Frm/s R: 0 | Frm/s T: 0

Index	System Time	Time Stamp	Channel	Directio	Frame ID	Type	Format	DLC	Data

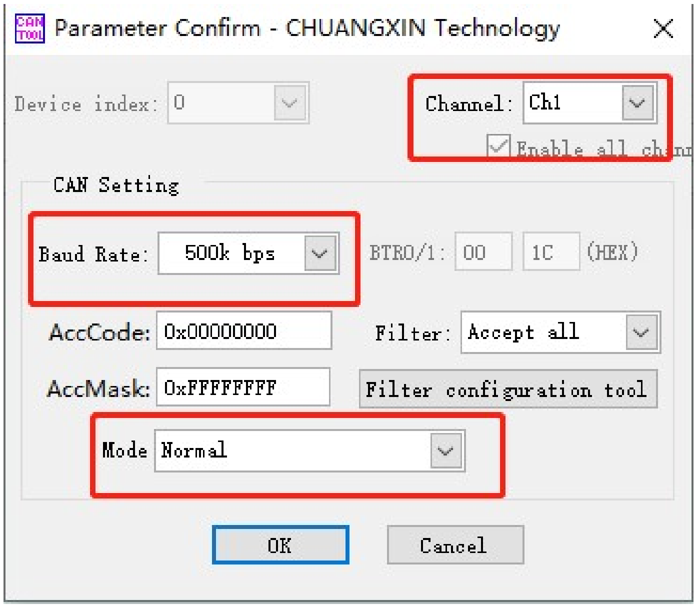
Figure 36: Open device



Open USB Device - CHUANGXIN Technology

Device Name: Device Index Number 0:Serial number: 01900202A91,

Figure 37: Device connection parameters


Parameter Confirm - CHUANGXIN Technology

Device index: 0

Channel: Ch1

☑ Enable all chann

CAN Setting

Baud Rate: 500k bps

BTR0/1: 00 1C (HEX)

AccCode: 0x00000000

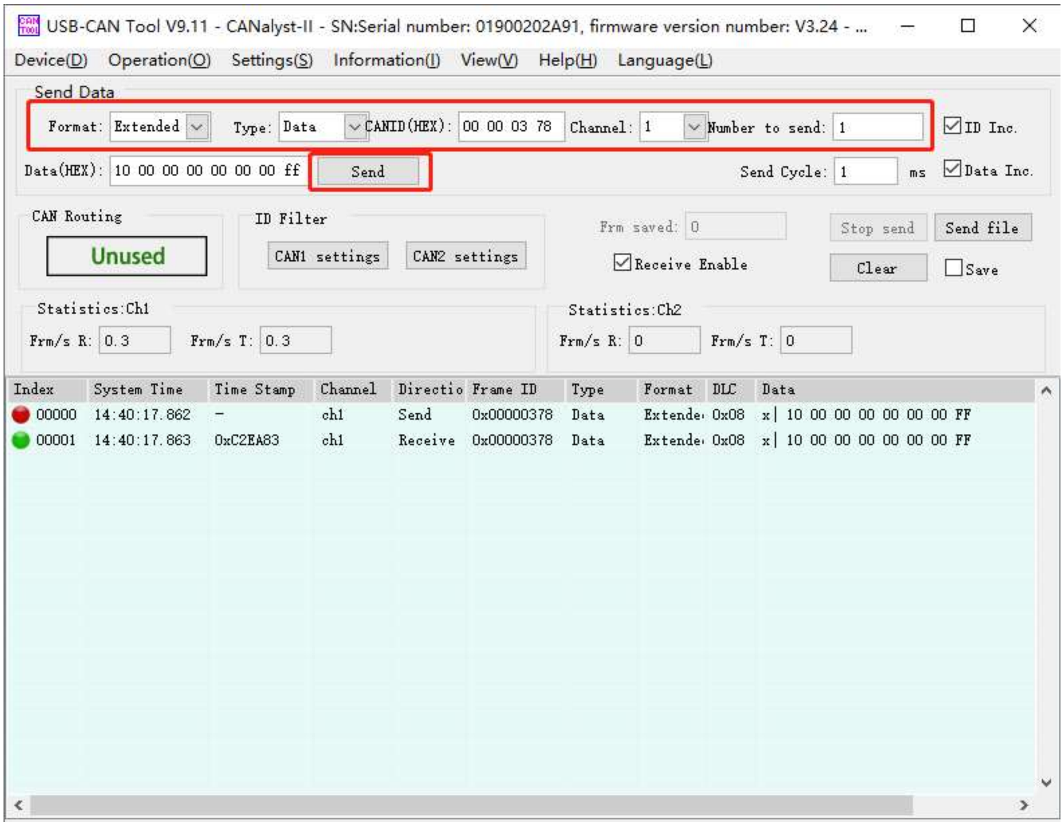
Filter: Accept all

AccMask: 0xFFFFFFFF

Mode Normal

Figure 38: Can interface parameters

After the test module is successfully connected to the device, send the test data to the device through the tool, and the device will return the test data as it is, as shown in the figure below. as figure 39 show.


USB-CAN Tool V9.11 - CANalyst-II - SN:Serial number: 01900202A91, firmware version number: V3.24 - ...

Device(D) Operation(O) Settings(S) Information(I) View(V) Help(H) Language(L)

Send Data

Format: Extended Type: Data CANID(HEX): 00 00 03 78 Channel: 1 Number to send: 1 ☑ ID Inc.

Data(HEX): 10 00 00 00 00 00 00 ff Send Cycle: 1 ms ☑ Data Inc.

CAN Routing: Unused

Frm saved: 0

☑ Receive Enable

□ Save

Statistics:Ch1 Frm/s R: 0.3 Frm/s T: 0.3

Statistics:Ch2 Frm/s R: 0 Frm/s T: 0

Send
ID Filter
CAN1 settings
CAN2 settings
Stop send
Send file
Clear

Index	System Time	Time Stamp	Channel	Directio	Frame ID	Type	Format	DLC	Data
● 00000	14:40:17.862	–	ch1	Send	0x00000378	Data	Extende	0x08	x| 10 00 00 00 00 00 00 FF
● 00001	14:40:17.863	0xC2EA83	ch1	Receive	0x00000378	Data	Extende	0x08	x| 10 00 00 00 00 00 00 FF

Figure 39: Data sending and receiving

5.3.4 Test RS422

The RS422 interface test needs to use a standard DB9 serial debugging line. In the example, the USB2.0 RS422/RS485 universal serial debugging line produced by China Z-TEK is used.

Figure 40: RS422/RS485 cable

Device and DB9 connection line sequence:

GIIICM680L		**DB9**
GPIO4(RXD-)	---	T/R-
GPIO5(RXD+)	---	T/R+
GPIO10(TXD+)	---	RXD+
GPIO11(TXD-)	---	RXD-

Connection example diagram：

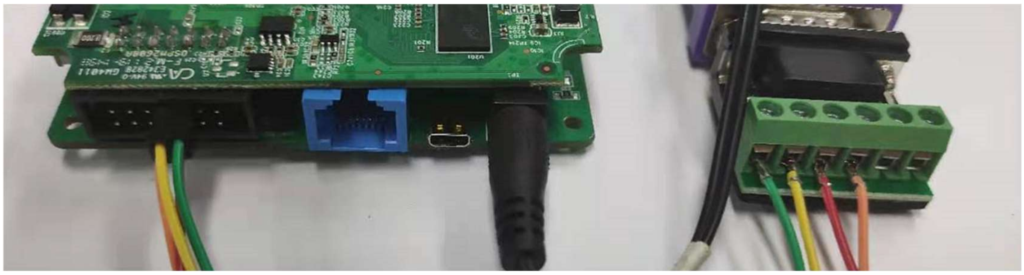
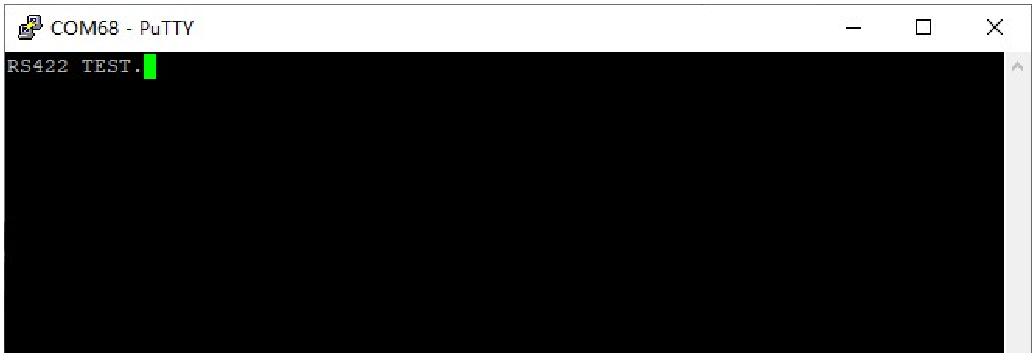


Figure 41: RS422 connection diagram

Connect the DB9 interface through putty, the baud rate is 115200bps, send data to the device, the device will return the data as it is. as figure 42 show.



```
RS422 TEST.
```

Figure 42: putty output information

5.3.5 Test RS485

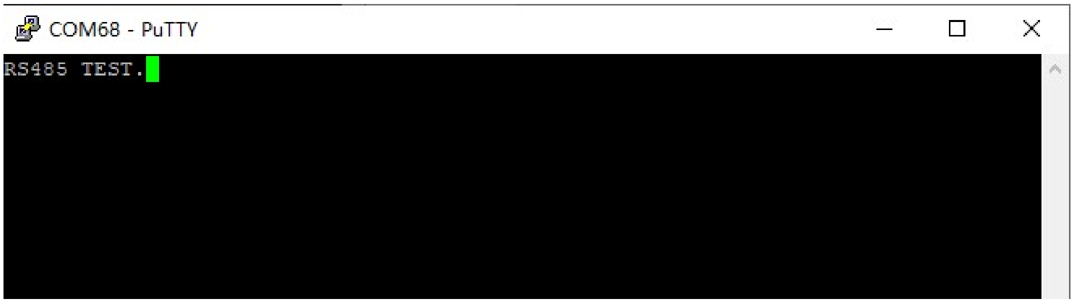
The RS422 interface test needs to use a standard DB9 serial debugging line. In the example, the USB2.0 RS422/RS485 universal serial debugging line produced by China Z-TEK is used.

Figure 43: RS422/RS485 cable

Device and DB9 connection line sequence:

GIIICM680L		**DB9**
GPIO4	---	T/R-
GPIO5	---	T/R+
GPIO10	---	T/R+
GPIO11	---	T/R-

Connect the DB9 interface through putty, the baud rate is 115200bps, send data to the device, the device will return the data as it is. as figure 44 show.



```
RS485 TEST.
```

Figure 44: putty output information

6 Support

For all general, partnership, career, and/or press inquiries, please contact us through email or phone number:

Email: crl@changhong.us
Phone: 1-408-970-0349
Address: 2580 North First Street, Suite 100, San Jose, CA 95131