

Language: c

Connect Himax-AIoT-NB-G2 device to your Azure IoT services

Table of Contents

| | |
|----------------------------------------------|----|
| I. Introduction | 2 |
| II. Prerequisites | 3 |
| III. Prepare the Device | 3 |
| IV. Connect to Azure IoT Central..... | 5 |
| V. Integration with Azure IoT Explorer | 7 |
| VI. Additional Links | 13 |

I. Introduction

About this document

This document describes how to connect Himax-AIoT-NB-G2 to Azure IoT Hub using the Azure SDK for Embedded C with certified device application and device models.

IoT Plug and Play certified device simplifies the process of building devices without custom device code. Using Solution builders can integrated quickly using the certified IoT Plug and Play enabled device based on Azure IoT Central as well as third-party solutions.

This getting started guide provides step by step instruction on getting the device provisioned to Azure IoT Hub using Device Provisioning Service (DPS) and using Azure IoT Explorer to interact with device's capabilities.

Himax-AIoT-NB-G2 development kit is equipped with industrial the best ultra-low power and performance tiny AIoT processor “WE-I Plus” and NB-IoT connectivity for battery powered extremely edge AIoT devices applications. WE-I Plus leverages open tiny AI frame work Tensor Flow Lite for Microcontroller(TFLu) to easily port customer AI models into WE-I Plus and Azure RTOS to quickly pass Azure PnP certification by customers. The target applications of Himax-AIoT-NB-G2 include smart buildings, manufacturing, retail, agriculture....etc.

Applications

- Smart buildings
- Manufacturing
- Retail
- Agriculture

II. Prerequisites

You should have the following items ready before beginning the process:

For Azure IoT Central

- [Azure Account](#)
- [Azure IoT Central application](#)

For Azure IoT Hub

- [Azure IoT Hub Instance](#)
- [Azure IoT Hub Device Provisioning Service](#)
- [Azure IoT Public Model Repository](#)

III. Prepare the Device

■ Hardware Environmental Setup

1. Prepare Himax-AIoT-NB-G2 and connect to PC using micro USB.
2. Prepare a NB-IOT SIM Card for connect cloud.

■ Software Environmental Setup

Himax-AIoT-NB-G2 SDK Download :

Download the source code from this [GitHub](#).

Pre-request Software :

1. Install GNU Development Toolkit

- ❖ [ARC GNU Tool Chain](#) section for more detail, current released GNU version is [GNU Toolchain for ARC Processors, 2020.09](#). After download and extract toolkit to local space, please remember to add it to environment PATH.

For example :

```
export PATH=[location of your ARC_GNU_ROOT]/bin:$PATH
```

- ❖ Build source code, generate 「ELF」 and 「MAP」 files.

2. Curl command

- ❖ Installing curl for Ubuntu Linux.

```
sudo apt update
```

```
sudo apt upgrade
```

```
sudo apt install curl
```

3. Install FT4222H Interface A/B Driver

- ❖ Windows USB driver for downloading **flash image** (and other

operation) via debug board (FT4222 inside).

4. Install Terminal - Tera Term

- ❖ Display log message.

5. Install [Azure IoT Hub Explorer](#)

- ❖ Use the tool to interact with and test your IoT Plug and Play devices.

6. HMX-AIOT-NB-G2_GUI Tool

- ❖ Windows tool for downloading IMG, and access data from WE-I Plus

7. image_gen_linux_tool

- ❖ The converting 「ELF」 and 「MAP」 files to **flash image** files.

IV. Connect to Azure IoT Central

1. Create an application

Please refer to [Quickstart - Create an Azure IoT Central application](#) to create a 「**Custom application**」 template.

2. Create a device template

Please refer to [Create a device template from the device catalog](#) to create the 「**Himax-AIoT-NB-G2 device template**」.

3. Add a device

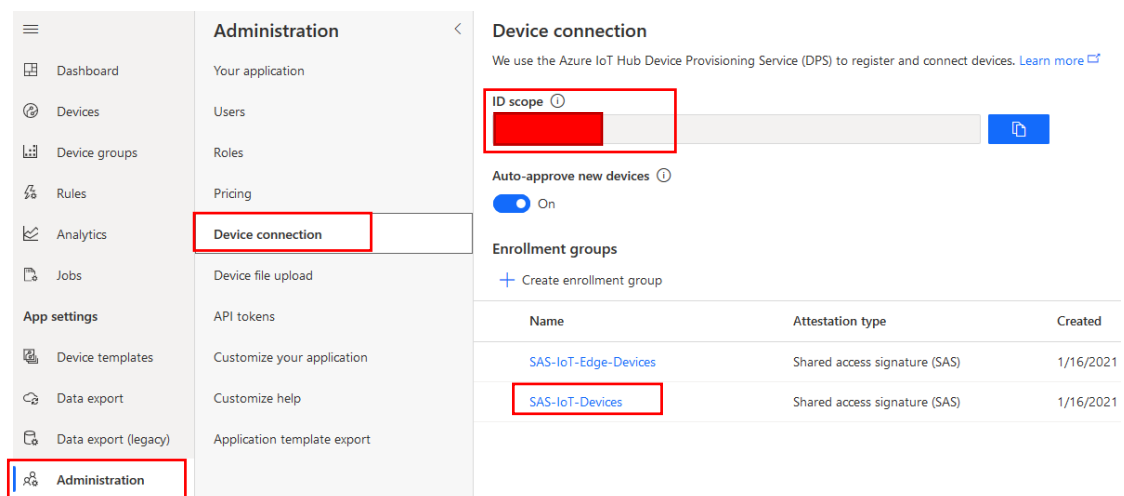
Add a new device under 「**Himax-AIoT-NB-G2 device template**」.
Make a note of the Device ID.

4. Get connection information

ID scope: In your IoT Central application, navigate to Administration > Device Connection. Make a note of the ID scope value.

Group primary key: In your IoT Central application, navigate to Administration > Device Connection > SAS-IoT-Devices.

Make a note of the shared access signature Primary key value.



Administration

Your application

Users

Roles

Pricing

Device connection

Device file upload

API tokens

Customize your application

Customize help

Application template export

Administration

Device connection

We use the Azure IoT Hub Device Provisioning Service (DPS) to register and connect devices. [Learn more](#)

ID scope ⓘ

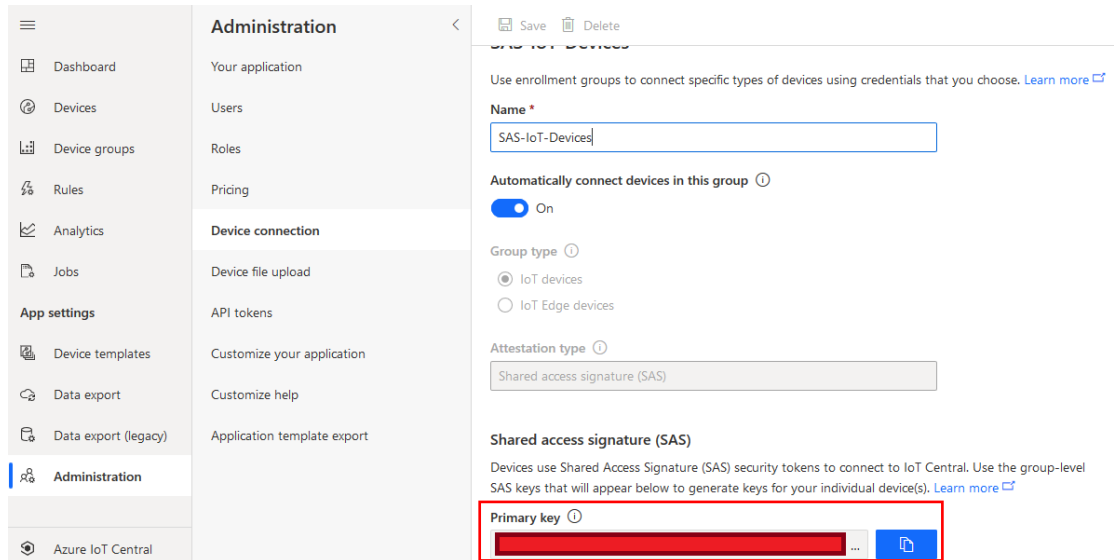
Auto-approve new devices ⓘ

On

Enrollment groups

+ Create enrollment group

| Name | Attestation type | Created |
|--------------------------------------|-------------------------------|-----------|
| SAS-IoT-Edge-Devices | Shared access signature (SAS) | 1/16/2021 |
| SAS-IoT-Devices | Shared access signature (SAS) | 1/16/2021 |



The screenshot shows the Azure IoT Central Administration interface. On the left is a navigation menu with options like Dashboard, Devices, Device groups, Rules, Analytics, Jobs, App settings, and Administration. The 'Administration' section is expanded, showing sub-options like Your application, Users, Roles, Pricing, Device connection, Device file upload, API tokens, Customize your application, Customize help, and Application template export. The 'Device connection' sub-section is selected. The main area displays settings for an enrollment group named 'SAS-IoT-Devices'. It includes a toggle for 'Automatically connect devices in this group' which is turned 'On'. The 'Group type' is set to 'IoT devices'. The 'Attestation type' is 'Shared access signature (SAS)'. Under the 'Shared access signature (SAS)' section, there is a 'Primary key' field which contains a redacted key value and a copy icon. A red rectangle highlights the 'Primary key' field.

Use the Cloud Shell to generate a device specific key from the group SAS key you just retrieved using the Azure CLI

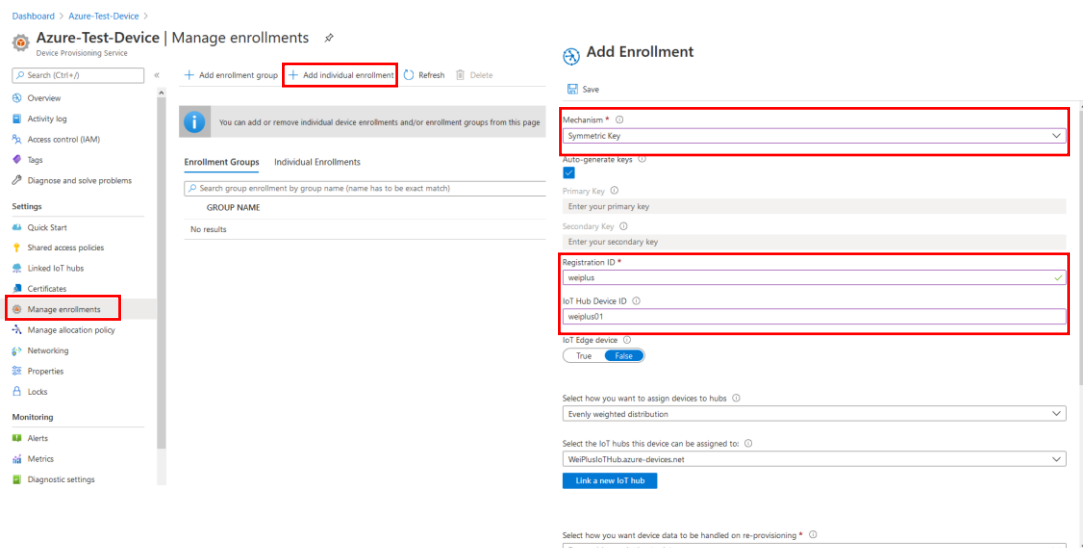
- `az extension add --name azure-iot`
- `az iot central device compute-device-key --device-id sample-device-01 --pk`

Make a note of the generated device key, and the ID scope for this application and flash it on the device.

V. Integration with Azure IoT Explorer

1. [Apply to Azure account](#)
2. [Set up the IoT Hub Device Provisioning Service with the Azure portal](#)
3. Create an enrollment device
 - 3-1. Sign in to the [Azure portal](#) and select the **All resources** in left-hand and choose your Device Provisioning service (DPS) instance.
 - 3-2. [Create a device enrollment.](#)
 - **Mechanism:** Select Symmetric Key as the identity attestation Mechanism.
 - **Auto-generate keys:** Check this box.
 - **Registration ID:** Enter a registration ID to identify the enrollment. Use only lowercase
 - alphanumeric and dash ('-') characters. For example, **weiplus**.
 - **IoT Hub Device ID:** Enter a device identifier. For example, **weiplus01**.

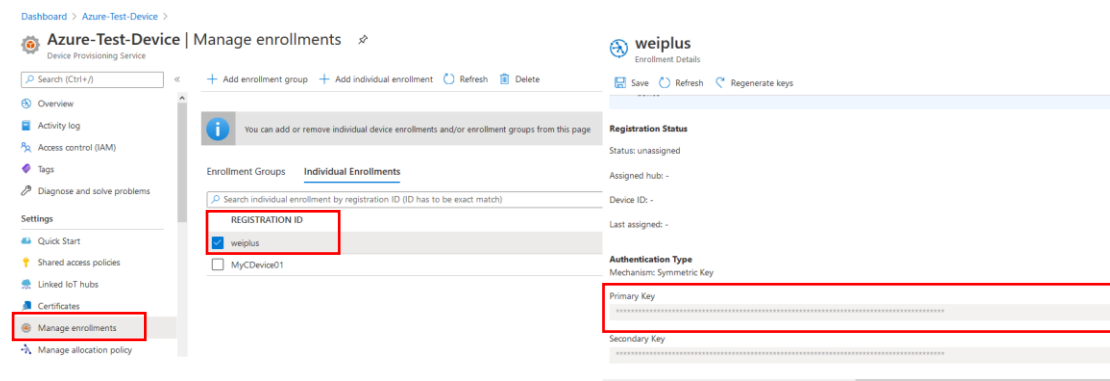
Choose your DPS → Manage enrollments → + Add individual enrollment



The screenshot displays the Azure IoT Hub Device Provisioning Service (DPS) interface. On the left, the 'Manage enrollments' option is selected in the sidebar. The main panel shows the 'Add Enrollment' form. The 'Mechanism' is set to 'Symmetric Key'. The 'Auto-generate keys' checkbox is checked. The 'Registration ID' is 'weiplus' and the 'IoT Hub Device ID' is 'weiplus01'. The 'IoT Edge device' checkbox is unchecked. The 'Add Enrollment' button is visible at the bottom of the form.

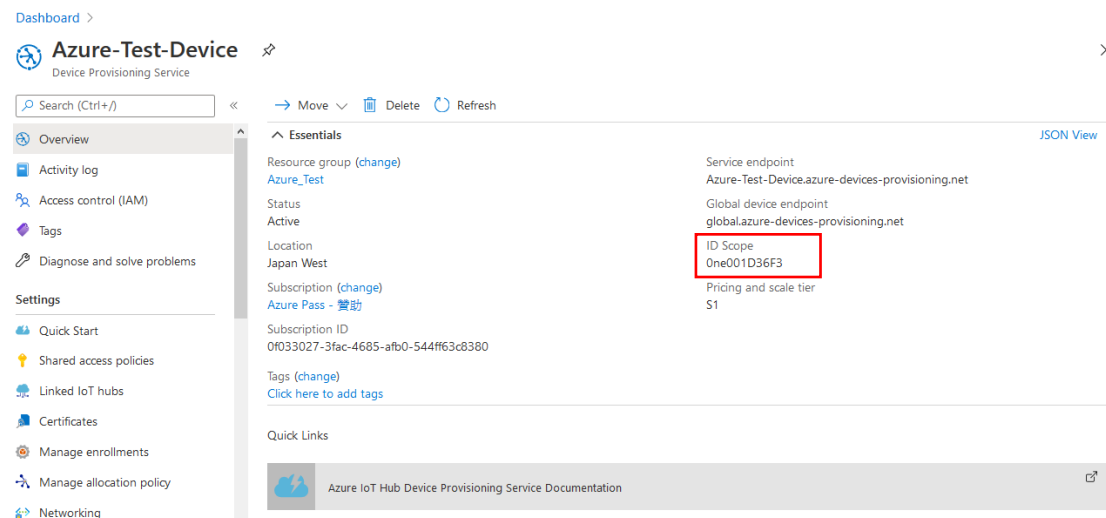
- Copy your 「**REGISTRATION ID**」 and 「**Primary Key**」 and go to back DPS overview copy 「**ID Scope**」 for generate connect string in firmware(sample_config.h)

Choose your DPS → Manage enrollments → Individual Enrollments → Choose your REGISTRATION ID → Copy Primary Key



The screenshot shows the 'Manage enrollments' page in the Azure IoT Hub Device Provisioning Service. The left sidebar has 'Manage enrollments' selected. The main area shows 'Individual Enrollments' for the 'weipius' enrollment group. The 'REGISTRATION ID' is highlighted in a red box. The 'Primary Key' is also highlighted in a red box.

Choose your DPS → Overview → Copy ID Scope



The screenshot shows the 'Overview' page in the Azure IoT Hub Device Provisioning Service. The left sidebar has 'Overview' selected. The main area shows the 'Essentials' section with the 'ID Scope' highlighted in a red box.

Replace the text in {} to your ID Scope, REGISTRATION ID, Primary Key in **sample_config.h** EX: #define ID_SCOPE "0ne001D36F3"

```
/* Required when DPS is used. */
#ifndef ENDPOINT
#define ENDPOINT "global.azure-devices-provisioning.net"
#endif /* ENDPOINT */

#ifndef ID_SCOPE
#define ID_SCOPE "{ID Scope}"
#endif /* ID_SCOPE */

#ifndef REGISTRATION_ID
#define REGISTRATION_ID "{REGISTRATION ID}"
#endif /* REGISTRATION_ID */

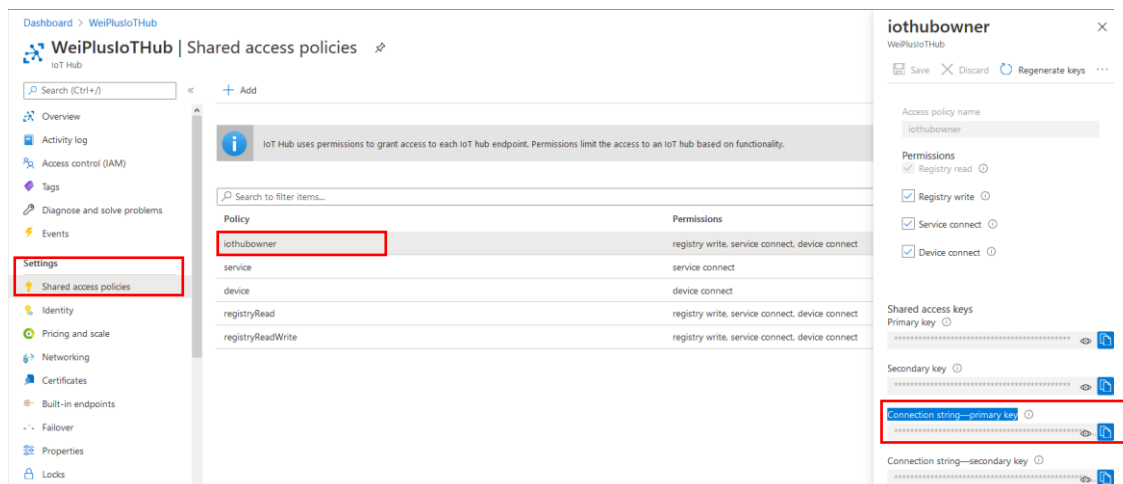
#endif /* ENABLE_DPS_SAMPLE */

/* Optional SYMMETRIC KEY. */
#ifndef DEVICE_SYMMETRIC_KEY
#define DEVICE_SYMMETRIC_KEY "{Primary Key}"
#endif /* DEVICE_SYMMETRIC_KEY */
```

5. Connect to your IoT Hub

If you the first time you run Azure IoT explorer, you are need to add your IoT hub connection string. After you **add connection string**, Click **View devices** in this hub.

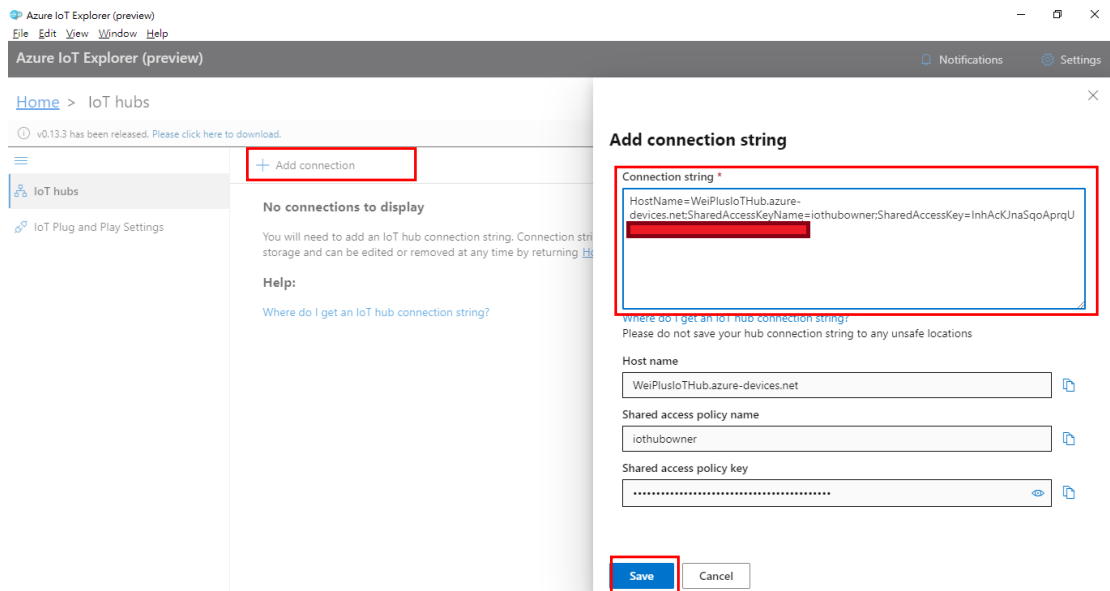
- **IoT Hub Connection string:** Sign in Azure Portal → All resource → Choose your IoT Hub → Setting, Shard access policies in left-hand → iothubowner → copy Connection string-primary key in right-hand



The screenshot displays the Azure IoT Hub 'Shared access policies' configuration page. On the left, the 'Settings' menu is open, and 'Shared access policies' is selected. The main content area shows a table of policies. The 'iothubowner' policy is highlighted with a red box. The right sidebar shows the details for the 'iothubowner' policy, including its permissions (Registry read, Registry write, Service connect, Device connect) and shared access keys. The 'Connection string—primary key' is highlighted with a red box.

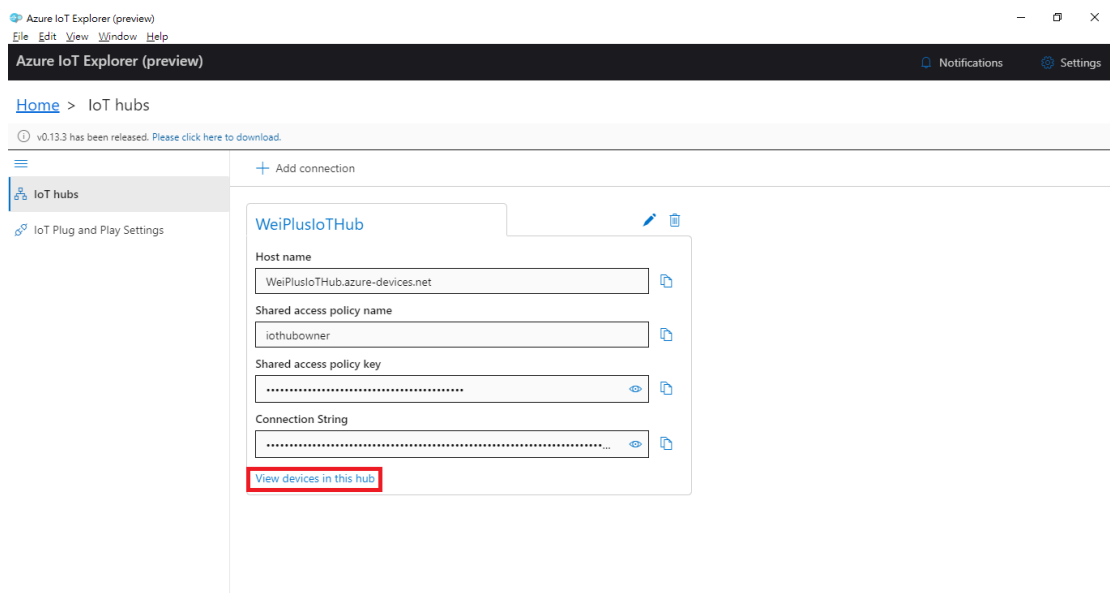
■ Azure IoT Explorer

Add connection string → Paste your connection string → Save



The screenshot shows the Azure IoT Explorer (preview) interface. On the left sidebar, the 'IoT hubs' section is selected. The main area displays 'No connections to display' with a message explaining that a connection string is needed. A red box highlights the '+ Add connection' button. On the right, the 'Add connection string' dialog is open. It contains a text area for the 'Connection string' with the value 'HostName=WeiPlusIoTHub.azure-devices.net;SharedAccessKeyName=iothubowner;SharedAccessKey=lnhAcKUnaSqaAprqU'. Below this, there are input fields for 'Host name', 'Shared access policy name', and 'Shared access policy key'. A red box highlights the 'Save' button at the bottom of the dialog.

View devices in this hub



The screenshot shows the Azure IoT Explorer (preview) interface. The left sidebar shows the 'IoT hubs' section. The main area displays the details for the 'WeiPlusIoTHub'. It includes input fields for 'Host name', 'Shared access policy name', and 'Shared access policy key'. Below these fields is a 'Connection String' field. A red box highlights the 'View devices in this hub' button at the bottom of the hub details card.

Choose IoT Plug and Play components in your connected device

Azure IoT Explorer (preview)

File Edit View Window Help

Azure IoT Explorer (preview) Notifications Settings

Home > WeiPlusIoTHub > Devices

New Refresh Delete

| ID | Enabled | Disconnected | SelfSigned | -- | |
|---------------------------------------------------------------------------------------------------------------------------------|---------|--------------|------------|----|------------------------------|
| 8e5a33bb9cfceec62c01ed5bca57aee15eb0249e7dc7d4f103673823f25e0eefde693d3b2da1d90a6cad9462011772a112676759ed0aed8ba7d6c8a1558e955 | Enabled | Disconnected | SelfSigned | -- | |
| watermeter01 | Enabled | Disconnected | Sas | -- | dtm:himax:weiplus:2 |
| dae230b7e0ad36de70421ba0631f0ad3ed27912754b853f93d1ee71f25e6577d505ed221766976f2bcae9f5010e650ce36f5453731a247ba0fecaff56305e0f | Enabled | Disconnected | SelfSigned | -- | |
| 0f613d2365ba0b1d557ba7839def128631fd59405264fb1d99932b156693cda2e8e6b3e2533d687473f21bf914d79bd156929b140ce96f6064dd82d2bf | Enabled | Disconnected | SelfSigned | -- | |
| weipus01 | Enabled | Connected | Sas | -- | dtm:himax:himax_aiot_nb_g2:2 |
| MyCDevice02 | Enabled | Disconnected | Sas | -- | dtm:com:example:Thermostat:1 |
| 6da94bc0af5f88cdd97898fbcecc42895b5c06d198483e347fc832554ba6acbfaf9d10ed1edda964a54492598a77ca9af765e24e3fd9e5eefef1fb74796e77 | Enabled | Disconnected | SelfSigned | -- | |

IoT Plug and Play components → Interface

Azure IoT Explorer (preview)

File Edit View Window Help

Azure IoT Explorer (preview) Notifications Settings

Home > WeiPlusIoTHub > Devices > weipus01 > IoT Plug and Play components > DEFAULT_COMPONENT > Interface

Interface Properties (read-only) Properties (writable) Commands Telemetry

Refresh Back

You model definition has been resolved from: Configurable Repository [Configure](#)

Interface Id

dtm:himax:himax_aiot_nb_g2:2

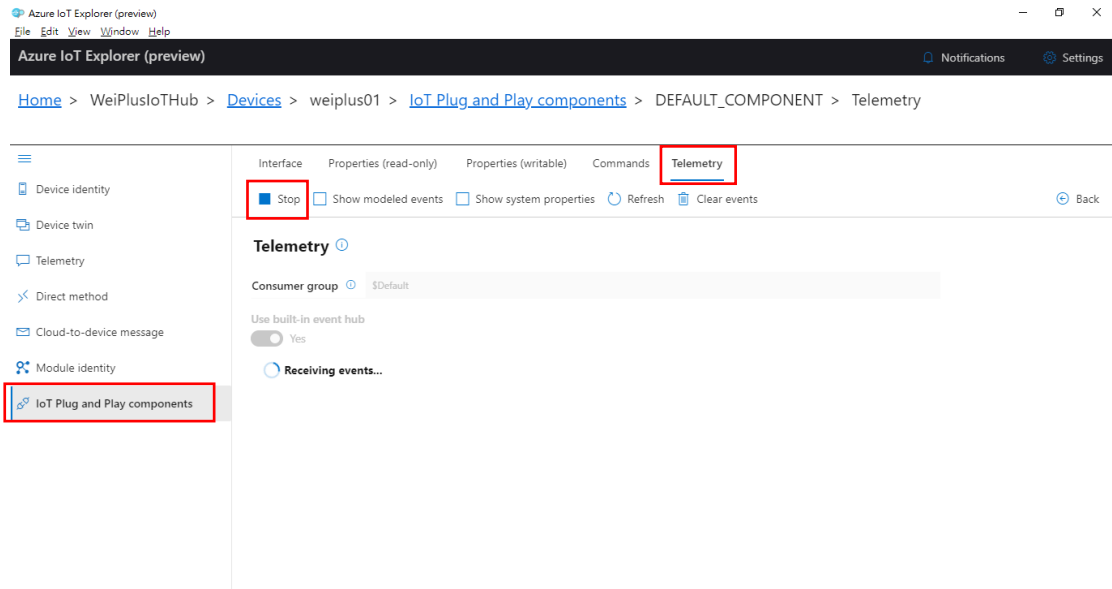
Name

Himax-AIoT-NB-G2

Description

```
--
1 {
2   "@context": "dtm:dts1:context:2",
3   "@id": "dtm:himax:himax_aiot_nb_g2:2",
4   "@type": "Interface",
5   "displayName": "Himax-AIoT-NB-G2",
6   "contents": [
7     {
8       "@type": "Telemetry",
9       "name": "human",
10      "displayName": "human",
11      "description": "Current detected human number"
12    }
13  ]
14 }
```

IoT Plug and Play components → Telemetry → Click Start → Receiving events...



Azure IoT Explorer (preview)

File Edit View Window Help

Azure IoT Explorer (preview) Notifications Settings

Home > WeiPlusIoT Hub > Devices > weiplus01 > IoT Plug and Play components > DEFAULT_COMPONENT > Telemetry

Interface Properties (read-only) Properties (writable) Commands **Telemetry**

Stop Show modeled events Show system properties Refresh Clear events Back

Telemetry

Consumer group \$Default

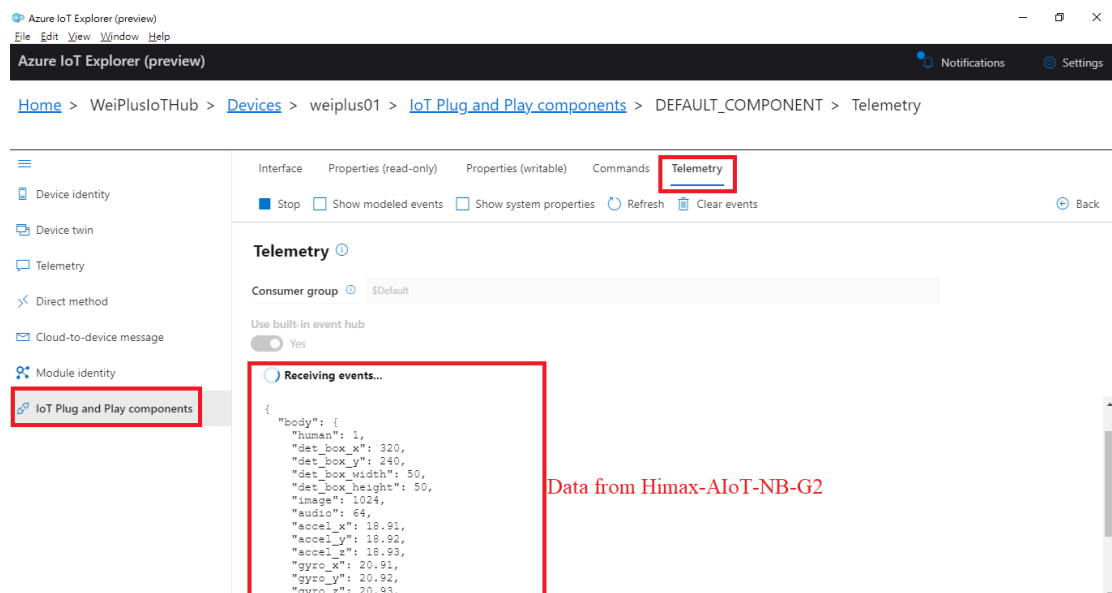
Use built-in event hub

Yes

Receiving events...

IoT Plug and Play components

Data from Himax-AIoT-NB-G2



Azure IoT Explorer (preview)

File Edit View Window Help

Azure IoT Explorer (preview) Notifications Settings

Home > WeiPlusIoT Hub > Devices > weiplus01 > IoT Plug and Play components > DEFAULT_COMPONENT > Telemetry

Interface Properties (read-only) Properties (writable) Commands **Telemetry**

Stop Show modeled events Show system properties Refresh Clear events Back

Telemetry

Consumer group \$Default

Use built-in event hub

Yes

Receiving events...

```
{
  "body": {
    "human": 1,
    "det_box_x": 320,
    "det_box_y": 240,
    "det_box_width": 50,
    "det_box_height": 50,
    "image": 1024,
    "audio": 64,
    "accel_x": 18.91,
    "accel_y": 18.92,
    "accel_z": 18.93,
    "gyro_x": 20.91,
    "gyro_y": 20.92,
    "gyro_z": 20.93
  }
}
```

Data from Himax-AIoT-NB-G2

IoT Plug and Play components

Note:

Connect DPS refer to 「[Communicate with your DPS using the MQTT protocol](#)」

Connect IoT hub refer to 「[Communicate with your IoT hub using the MQTT protocol](#)」

VI. Additional Links

- [Manage cloud device messaging with Azure-IoT-Explorer](#)
- [Import the Plug and Play model](#)
- [Configure to connect to IoT Hub](#)
- [How to use IoT Explorer to interact with the device](#)

Note: If you need any more information please reference the link.

- <https://www.himax.com.tw/products/intelligent-sensing/always-on-smart-sensing/>
- <https://www.himax.com.tw/products/intelligent-sensing/always-on-smart-sensing/inquiry-form/>