

Language: c

Connect WE-I Plus device to your Azure IoT services

Contents

I. Introduction	2
II. Prerequisites	3
III. Prepare the Device	3
IV. Integration with Azure IoT Explorer	5
V. Connect to Azure IoT Central.....	11
VI. Additional Links	12

I. Introduction

About this document

This document describes how to connect WE-I Plus 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.

Industrial Ultra Long Battery Life AIoT Devices WE-I Plus is an ultralow power AI accelerator-embedded ASIC platform solution for application developers to develop and deploy convolutional neural networks (CNN)-based machine learning (ML) models on AIoT applications including smart home appliances and surveillance systems.

Flexible and optimized computation architecture could make AIoT devices perfectly fit blooming IoT applications with intelligence. WE-I can enable IoT devices to be smart with following excellent features:

- A programmable processor with an enhanced DSP
- Power-efficient image and JPEG hardware accelerator for real-time motion detection, object detection, and image processing
- Optimized SRAM size to support simultaneously ultralow power vision

Applications

- Smart home appliance
- Intelligent home security & safety
- Wise city & building
- Brilliant consumer devices

II. Prerequisites

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

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

III. Prepare the Device

■ Hardware Environmental setup

Prepare WE-I Plus and connect to PC using micro USB.

Prepare a NB-IOT SIM Card for connect Cloud.

■ Software Environmental setup

Pre-request Software :

- 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.
- Curl command
 - ❖ Installing curl for Ubuntu Linux.
`sudo apt update`
`sudo apt upgrade`
`sudo apt install curl`
- Image Generator tool
 - ❖ Windows tool for converting ELF and MAP files to IMG files
- FT4222H Interface A/B Driver
 - ❖ Windows USB driver for downloading IMG (and other operation) via debug board (FT4222 inside).
- HMX_FT4222H_GUI Tool
 - ❖ Windows tool for downloading IMG, and access data from WE-I Plus
- Terminal - Tera Term
 - ❖ Display log message.
- [Azure IoT Hub Explorer](#)

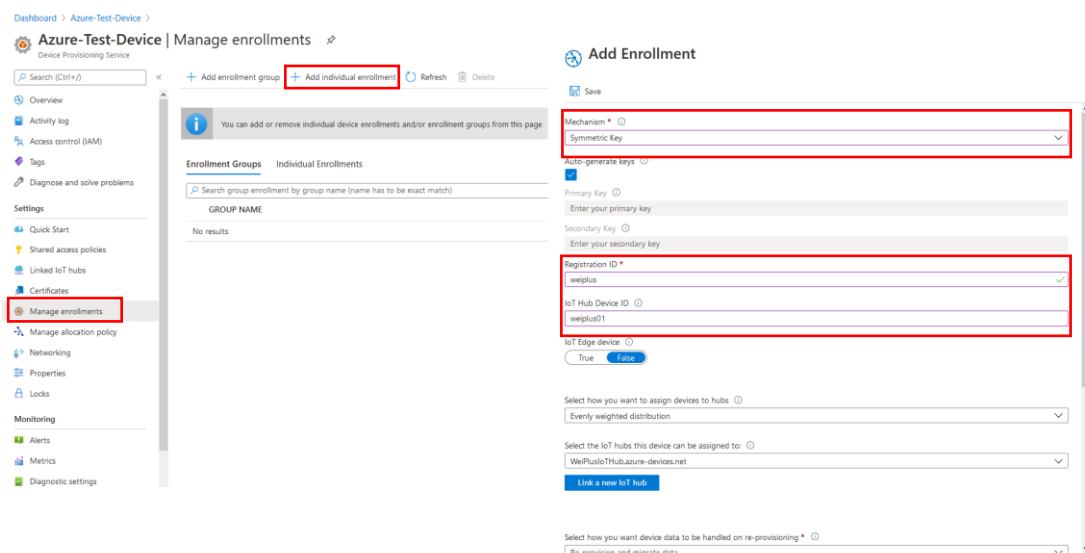
- ❖ Use the tool to interact with and test your IoT Plug and Play devices.

Azure SDK Version: [Azure SDK for Embedded C. release 1.0.0 on 22 Sep 2020](#)

IV. 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**.

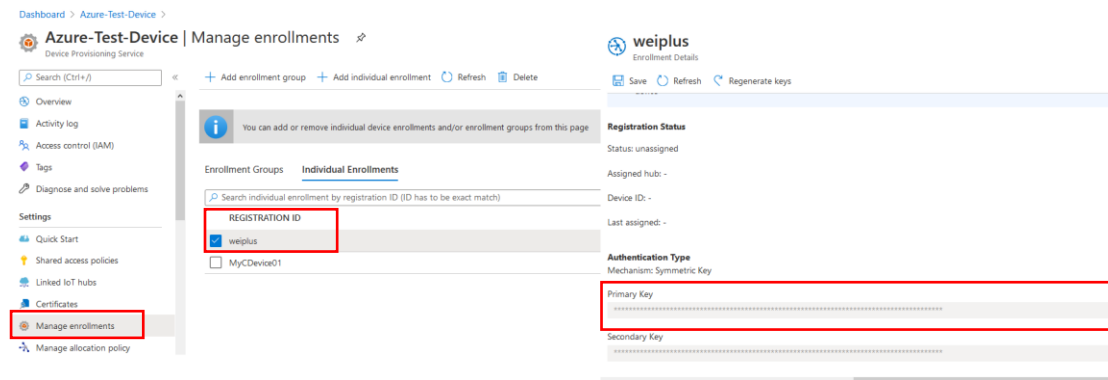
Choose your DPS → Manage enrollments → + Add individual enrollment



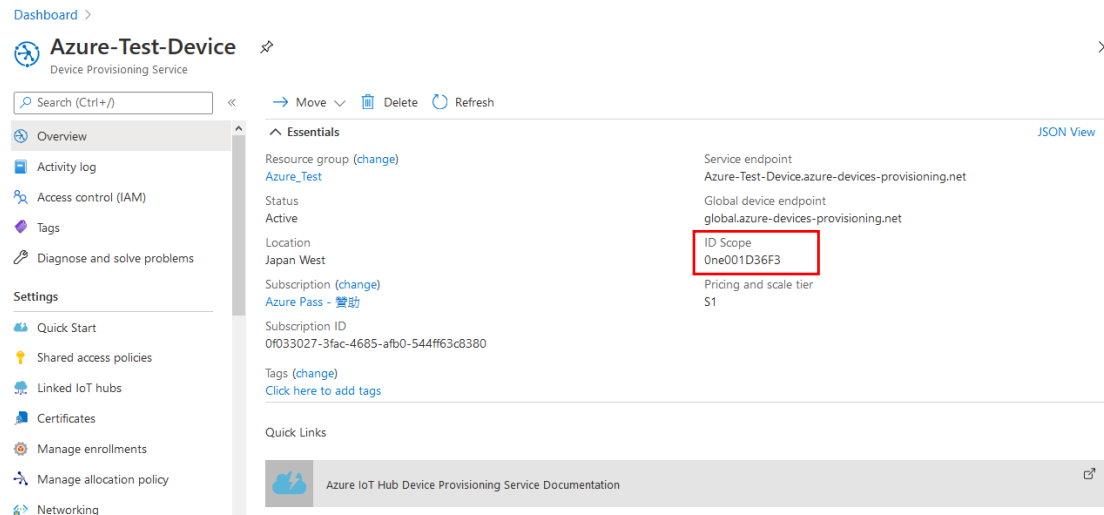
The screenshot displays the Azure IoT Hub Device Provisioning Service (DPS) interface. On the left, the 'Manage enrollments' option is selected in the sidebar. The main panel shows the 'Add Enrollment' form. The 'Mechanism' is set to 'Symmetric Key'. The 'Auto-generate keys' checkbox is checked. The '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.

4. 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



Choose your DPS → Overview → Copy ID Scope



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 */

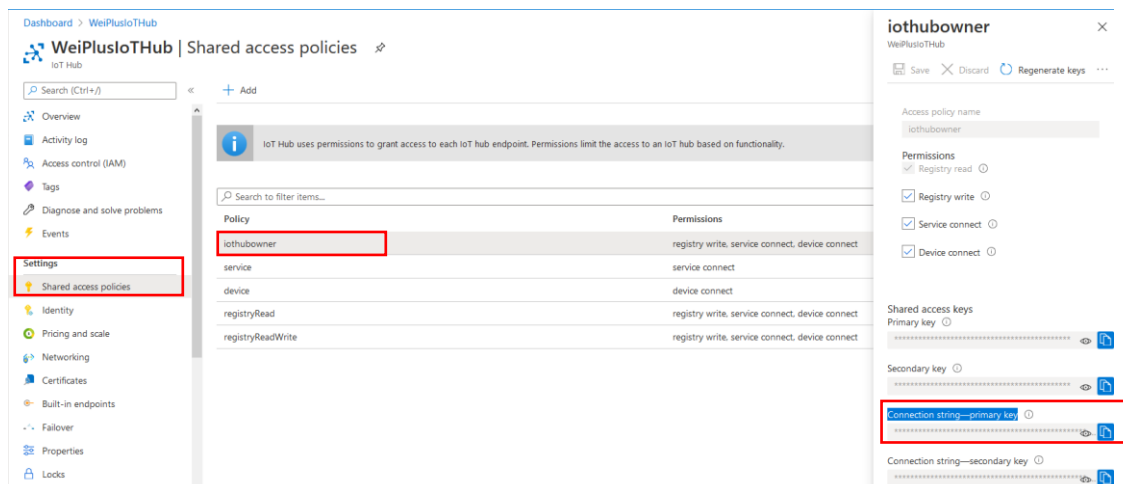
#endif /* ENABLE_DPS_SAMPLE */

/* 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**.

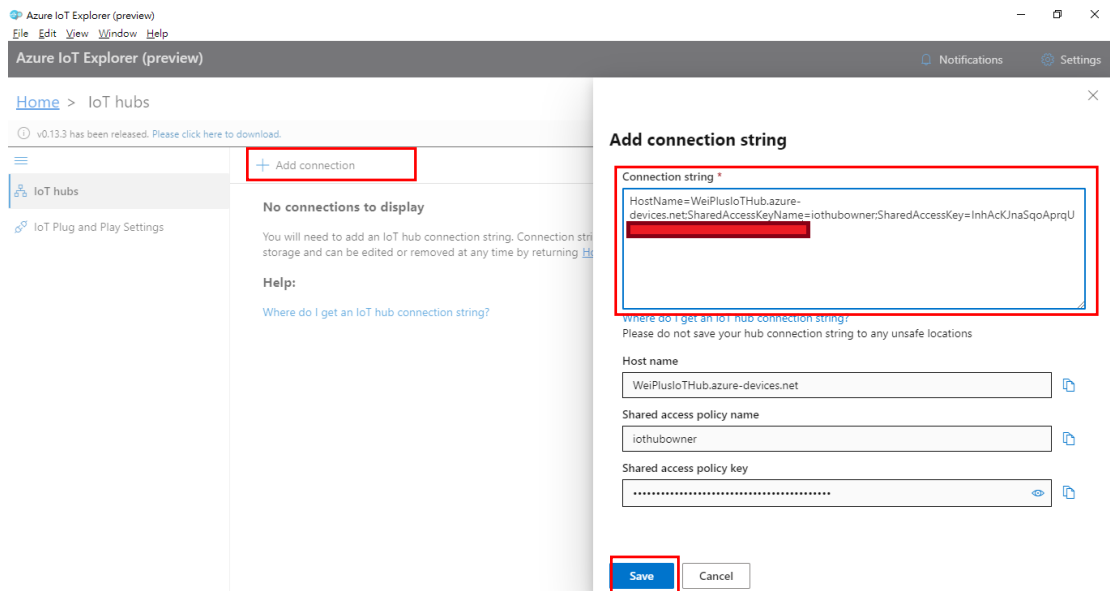
- **IoT Hub Connection string:** Sign in Azure Portal → All resource → Choose your IoT Hub → Setting, Shard access policies in left-hand → iothubowner → copy Connection string-primary key in right-hand



The screenshot displays the Azure IoT Hub management interface. On the left, the 'Settings' menu is expanded, and 'Shared access policies' is selected. The main content area shows a table of shared access policies. The 'iothubowner' policy is highlighted with a red box. The right-hand pane provides details for the 'iothubowner' policy, including its permissions (Registry read, Registry write, Service connect, Device connect) and the 'Connection string—primary key', which is also highlighted with a red box.

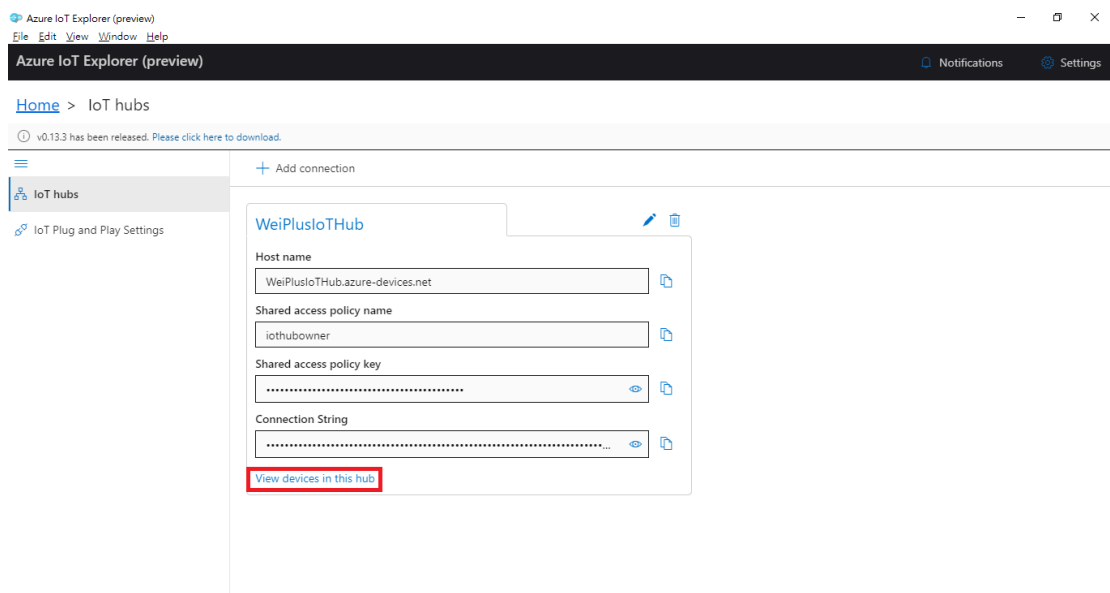
■ Azure IoT Explorer

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' and a '+ Add connection' button. A modal dialog titled 'Add connection string' is open on the right. 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'. At the bottom of the dialog, there are 'Save' and 'Cancel' buttons. The 'Save' button is highlighted with a red box.

View devices in this hub



The screenshot shows the Azure IoT Explorer (preview) interface. The 'IoT hubs' section is selected in the left sidebar. The main area displays a card for the 'WeiPlusIoTHub'. The card contains the following information: Host name (WeiPlusIoTHub.azure-devices.net), Shared access policy name (iothubowner), Shared access policy key (masked), and Connection String (masked). At the bottom of the card, there is a button labeled 'View devices in this hub', which is highlighted with a red box.

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 > WeiPlusIoT Hub > Devices

New Refresh Delete

Device ID	Status	Connection state	Authentication type	Last status up...	IoT Plug and ...	Edge device
b6091d59f6c8cad40356af3f1cca89cbb9961999ktb5eeb5b388a2213036aafda87283b27671149bd52b4a834af805e30d8865efbaace591a70d912b78f060	Enabled	Disconnected	SelfSigned	--		
8e5a33bb9cfceec62c01ed5bca57aae15eb0249e7dc7def103673823f25e0eefde693d3b2da1d90d6cad9462011772a112676759ed0aed8ba7d6c8a1558e955	Enabled	Disconnected	SelfSigned	--		
dae230b7e0ad36de70421ba0631f0ad3ed27912754b853f93d1ee7f1f25e6577d505e0d221766976f2bcae9f5010e050ce36f5453731a247ba0fecaff56305e0f	Enabled	Disconnected	SelfSigned	--		
0f613f2365ba9b1d557ba7839def128631fd59405264fb1d99932b156693cda2e8e6b3e2533d3d6874773f21bf914d79bd1569f29b140ce96f6064d8d2d2bf	Enabled	Disconnected	SelfSigned	--		
weiplus01	Enabled	Connected	Sas	--	dtm:himax:weiplus:1	
MyCDevice02	Enabled	Disconnected	Sas	--	dtm:com:example:Thermostat:1	
6da4bc0af5f88cdd97898fbcecc42895b5c06d198483e347fcf832554ba6acbfaf9d10ed1edda964a54492598a77ca9af765e24e3fd9e5eefef1fb74796e77	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 > WeiPlusIoT Hub > Devices > weiplus01 > 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:weiplus:1

Name
 --

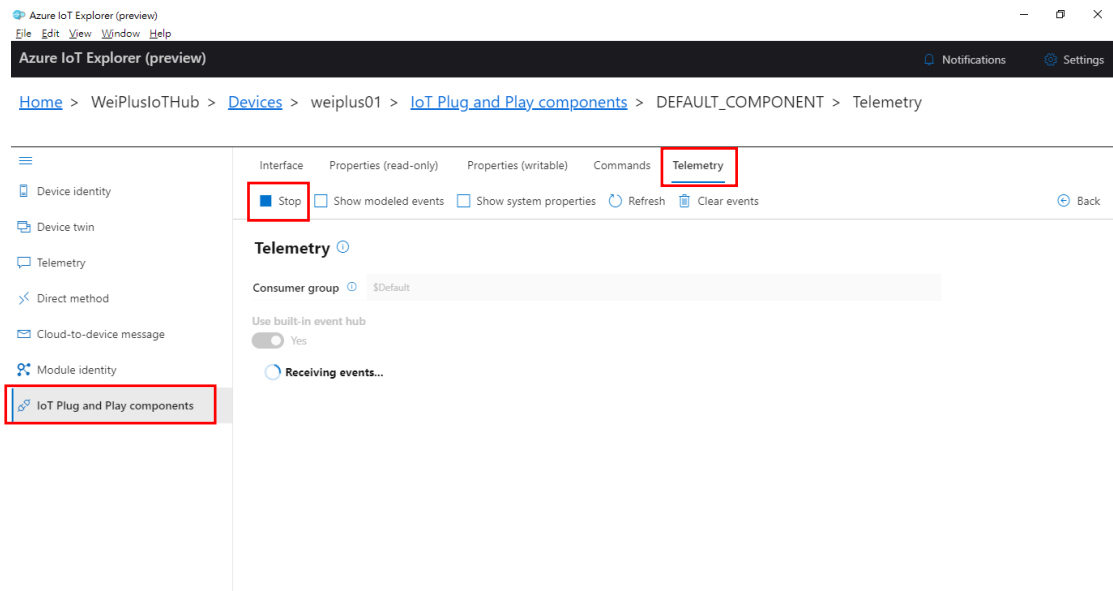
Description
 --

```

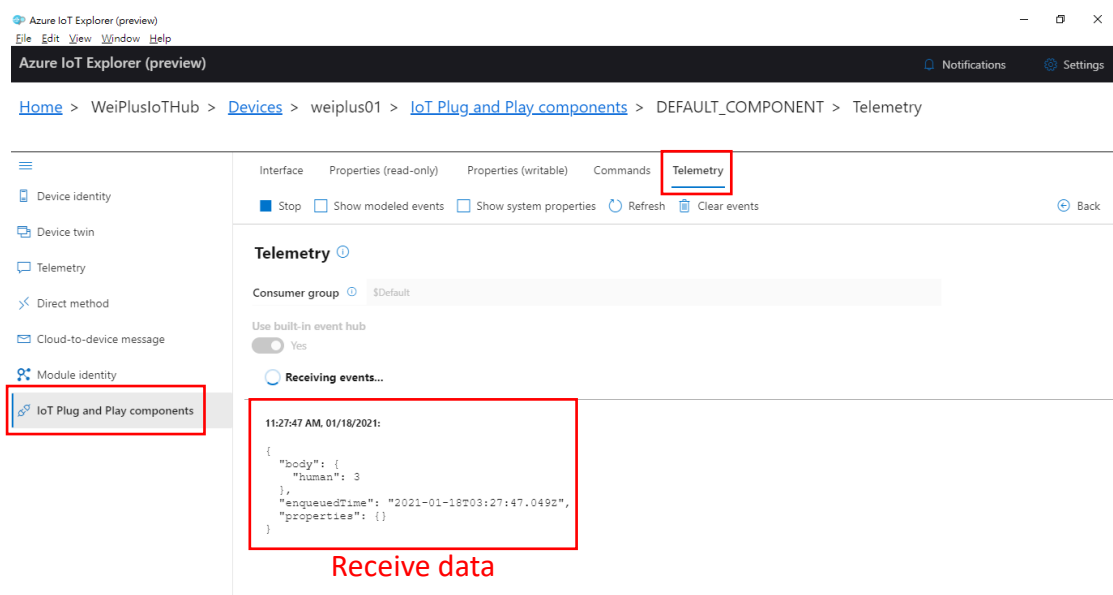
2  "id": "dtm:himax:weiplus:1",
3  "econtext": "dtm:dtul:context;2",
4  "type": "Interface",
5  "contents": [
6    {
7      "type": "Telemetry",
8      "displayName": {
9        "en": "human_presence"
10     },
11     "description": "Current detected human number.",
12     "name": "human",
13     "schema": "Integer"
  
```

IoT Plug and Play components

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



Receive data from WE-I Plus



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](#)」

V. 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 「**WE-I Plus device template**」.

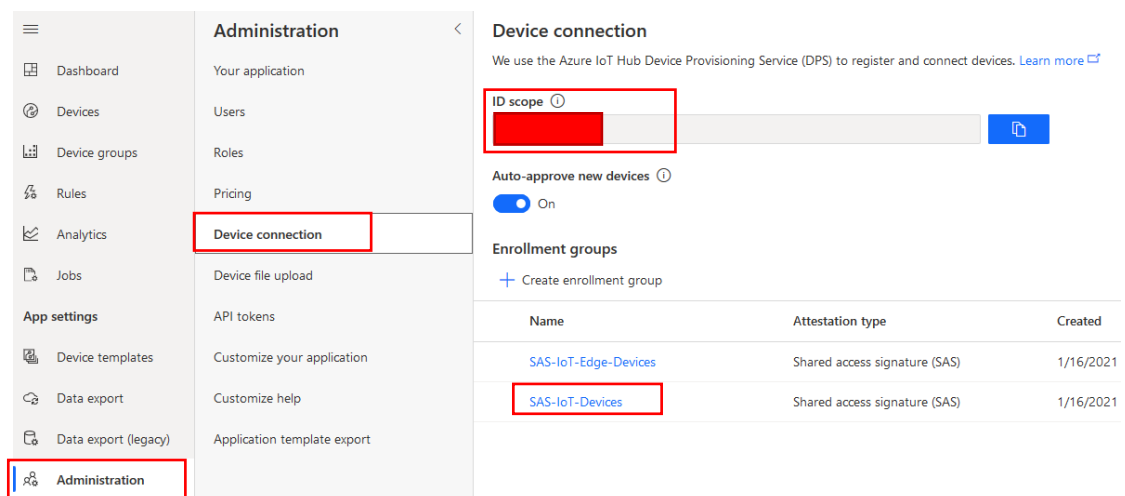
3. Add a device

Add a new device under 「**WE-I Plus device template**」. Make a note of the Device ID.

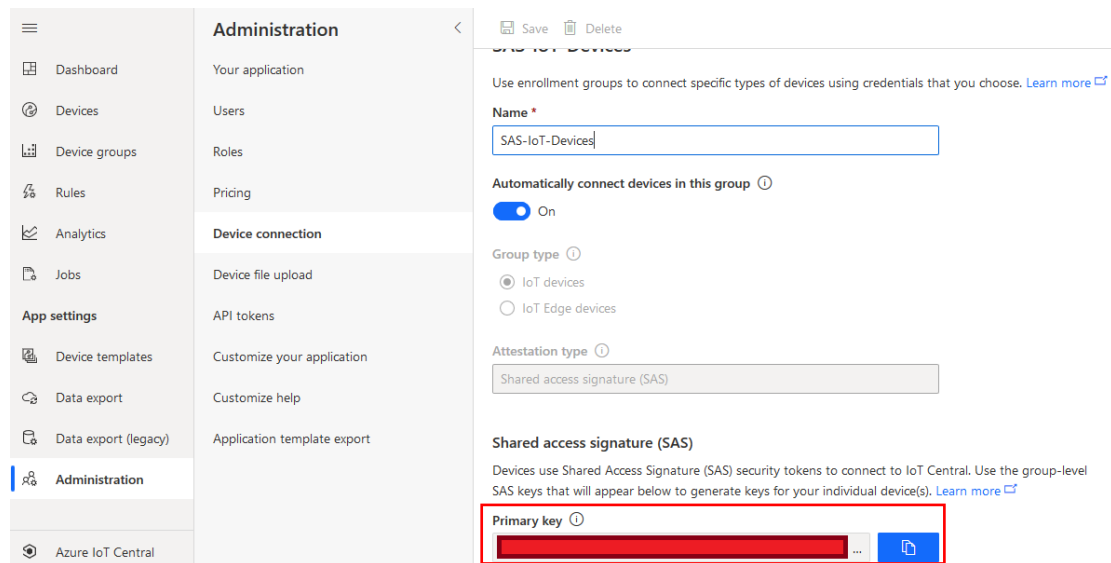
4. Get connection information

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.



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



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.

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/>