

Connect Armadillo-IoT GW G3M1 basic device to your Azure IoT services

Table of Contents

- [Introduction](#)
- [Prerequisites](#)
- [Prepare the Device](#)
- [Connect to Azure IoT Central](#)
- [Additional Links](#)

Introduction

About this document

This document describes how to connect Armadillo-IoT GW G3M1 basic to Azure IoT Hub using the Azure IoT Explorer 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 be 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.

Armadillo-IoT GW G3M1's "basic" configuration has capabilities:

1. Sense the temperature of CPU and send it as telemetry.
2. Reboot by Cloud2Device message.

Prerequisites

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

For Azure IoT Central

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

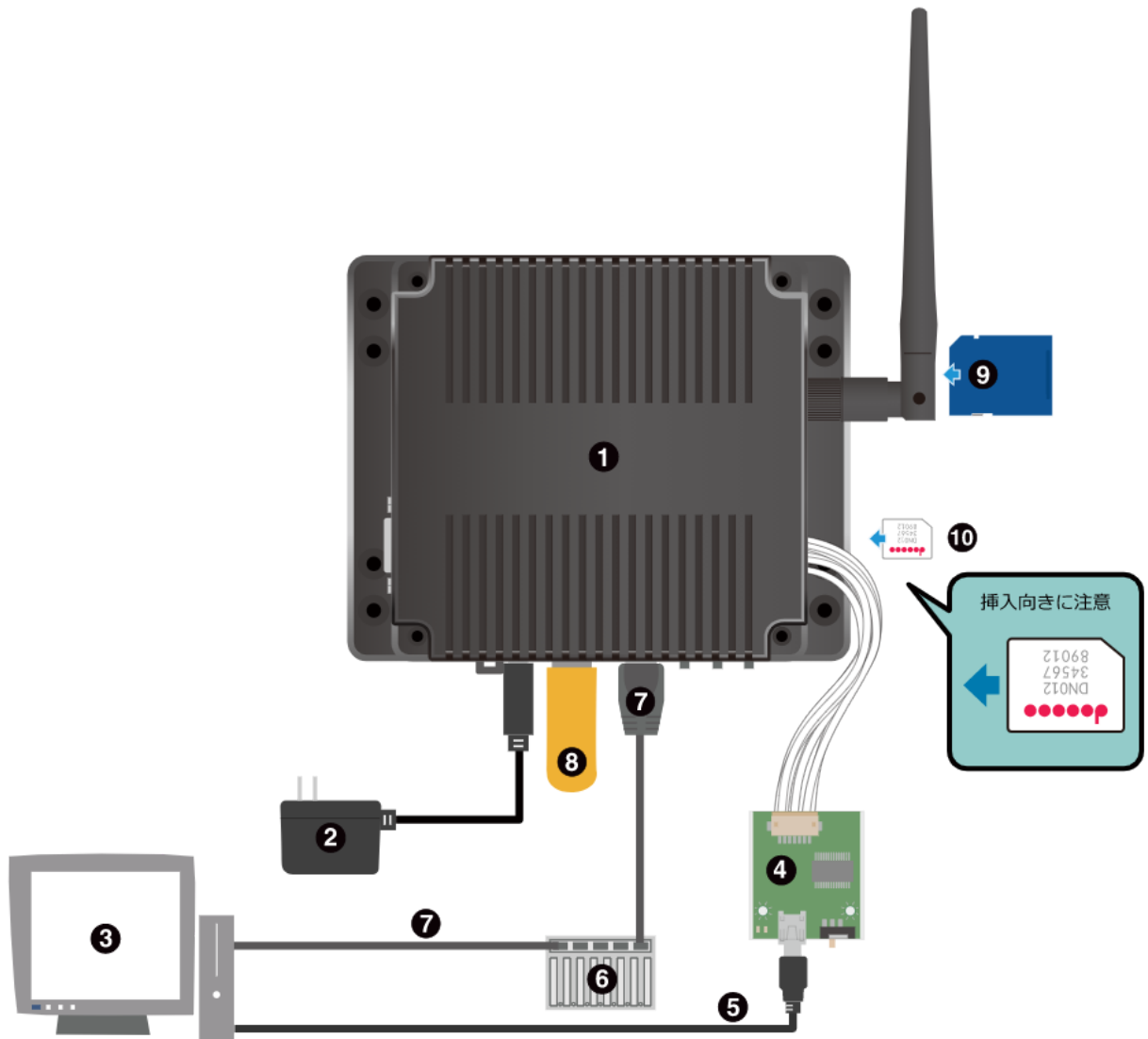
For Azure IoT Hub

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

Prepare the Device.

Hardware Environmental Setup (Estimated time: 5-20 minutes)

- Prepare Armadillo-IoT GW G3M1 device. See below figure. (1) is the device.



- (1): Armadillo-IoT GW G3M1
- (2): power adaptor (accessory of the device)
- (3): your PC
- (4): USB-Serial adaptor (accessory of the device)
- (5): USB cable (accessory of the device)
- (6): LAN Hub
- (7): LAN cable
- (8): USB memory
- (9): SD memory card

(10): microSIM card

- Connect PC to Armadillo-IoT GW G3M1 device with USB-Serial adaptor and USB cable. (Above figure's (4) and (5). The detailed description is [here \(in Japanese\)](#))
- Prepage the software for serial communication such as Tera Term (for Windows), minicom (for Linux). The detailed descriptions is [here \(in Japanese\)](#)
- Connect power adaptor with the device (Above figure's (2)).
- Connect LAN cable with the device (Above figure's (7)). You can using Wifi too, see [this guide \(in Japanese\)](#)

Software Environmental Setup (Estimated time: 60-80 minutes)

Armadillo-IoT GW G3M1's "basic" configuration has capabilities:

1. Sense the temperature of CPU and send it as telemetry.
2. Reboot by Cloud2Device message.

The setup instruction is:

- Login to the device from PC using the software for serial communication.
- Update the installed package using apt. (*Estimated time: 2 minutes*)

```
# apt update
# apt upgrade
```

- Install Python 3.8 or other version higher than 3.7.3 via build from the source (*Estimated time: 30 minutes*)

```
# apt install build-essential zlib1g-dev libncurses5-dev libgdbm-dev libnss3-dev
libssl-dev libreadline-dev libffi-dev curl libbz2-dev
# mkdir work
# cd work
# wget https://www.python.org/ftp/python/3.8.9/Python-3.8.9.tgz
# tar zxvf Python-3.8.9.tgz
# cd Python-3.8.9
# ./configure
# make -j 2

# make altinstall
# pip3.8 install --upgrade pip
```

- Install Azure IoT device SDK for Python

```
# pip3.8 install azure-iot-device
```

- Install rngd for prevent [this problem](#)

```
# apt install rng-tools
```

- Install git for setup the devie code

```
# apt install git
```

- Install the device application

```
# cd
# cd work
# git clone https://github.com/atmark-techno/Azure-IoT-samples.git
```

Prepare IoT Hub and DPS configuration

Please refer to [this tutorial](#) to complete the following procedures:

1. Use [Azure CLI](#) or Azure portal to create a Resource Group, and an IoT Hub, a DPS (Device Provisioning Service). Azure CLI can be installed and run on your PC. Please refer [this guide](#) to install it. When you use Azure CLI locally (i.e. on your PC), you should [sign in](#) at first.
2. Link the DPS instance to your IoT Hub instance.
3. Create a device by individual device enrollment in your DPS instance.
4. Make a note of the DPS information (DPS ID Scope, Device ID, Symmetric Key)
5. Set the DPS information got from the former step in work/Azure-IoT-samples/Armadillo-IoT_GW/config.json

```
{
  "auth": {
    "IOTHUB_DEVICE_DPS_ENDPOINT": "global.azure-devices-provisioning.net",
    "IOTHUB_DEVICE_DPS_ID_SCOPE": "",
    "IOTHUB_DEVICE_DPS_DEVICE_ID": "",
    "IOTHUB_DEVICE_DPS_DEVICE_KEY": ""
  },
  "send_interval": 2,
  "thermalsense_interval": 2,
  "disable_reboot": false
}
```

Run the device application

```
# cd Azure-IoT-samples/Armadillo-IoT_GW
# python3.8 ./azure_basic.py
```

Try to using Azure IoT explorer with your IoT Hub. You can see the status of your device and telemetry, etc. Please refer to [this guide](#).

If you'd like to stop the application, type 'Q' and Enter.

Connect to Azure IoT Central

1. Create an application

You can create an IoT Central application using the Azure CLI.

```
az iot central app create -n <app name> -g <resource group name> -s <subdomain name; equal with appname> -p ST0 -l <location; e.g. "japan">
```

Or please refer to [this tutorial](#) to create a "Custom application" template. After create the IoT Central app, you can navigate to it from the Resource Group at Azure portal.

2. Get connection information

To configure a device connect to Azure IoT Central you need the following

- 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.

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 --primary-key <the group SAS primary key value> --device-id <device ID>
```

Make a note of the generated device key, and the ID scope for this application and set it to the config.json file.

3. Run the device application

If running, stop the device application, and start it with new setting (for IoT Central app). When the device application start sending telemetry, you will see the device at your IoT Central App's "Armadillo-IoT GW G3M1 basic" device group. See the graph of telemetry data and properties with IoT Central App's view.

Additional information

With Azure CLI, you can change the writable property of the device which connecting your IoT Central app. Try following procedure:

1. Get your access token for IoT Central and note it. Note the value of 'accessToken' from the command response.

```
az account get-access-token --resource https://apps.azureiotcentral.com
```

2. If not running, run the device application
3. Invoke REST API with access token you retrieved at former step

```
az rest --method PATCH --url https://{IoT Central app
name}.azureiotcentral.com/api/preview/devices/{Device ID}/properties --body "
{"thermalsense_interval": 4}" --headers "{\"Authorization\": \"Bearer <access
token>\"}"
```

It must be replace '{IoT Central app name}', '{Device ID}', '{access token}' with the name of your IoT Central app, the device ID, and access token respectively. When the REST API invocation succeeded, you will see that the interval of the telemetry changed to 4[sec] from 2[sec](setting value with config.json).

Additional Links

Please refer to the below link for additional information for Plug and Play

- [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](#)