

Azure connectivity: Get started document

platform	device	language
Proprietary	DIRIS Digiware M-xx DIRIS Digiware D-xx	Not required

- [Introduction](#)
 - [About this document](#)
- [Step 1: Prerequisites](#)
- [Step 2: Prepare your DIRIS Digiware M/D gateway](#)
 - [System wiring](#)
 - [Configuration with DPS \(Device Provisioning Service\)](#)
 - [DPS with SAS Token authentication](#)
 - [DPS with X.509 certificate authentication](#)
- [Step 3: Integration with Azure IoT Explorer](#)
 - [Connect Azure IoT Explorer to the Azure IOTHub](#)
 - [View telemetry messages](#)
- [Step 4: Connect to Azure IoT Central](#)
- [Step 5: Additional Information](#)
- [Step 6 : Additional Links](#)

Introduction

About this document

This document describes how to connect the SOCOMEC DIRIS Digiware M/D gateways with Azure IoT. This multi-step process includes:

- [Configuring Azure IoT Hub](#)
- [Registering your IoT device](#)
- [Provisioning your devices on Device Provisioning service \(DPS\)](#)
- [Configuring Azure IoT connectivity on device](#)

Step 1: Prerequisites

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

- [Setup your IoT hub](#)
- [Provision your device over DPS](#)



The DIRIS Digiware M/D gateways integrate a firmware natively compatible with Azure connectivity. There is no need to compile source code or libraries, and configuration can be done easily through the integrated web server.

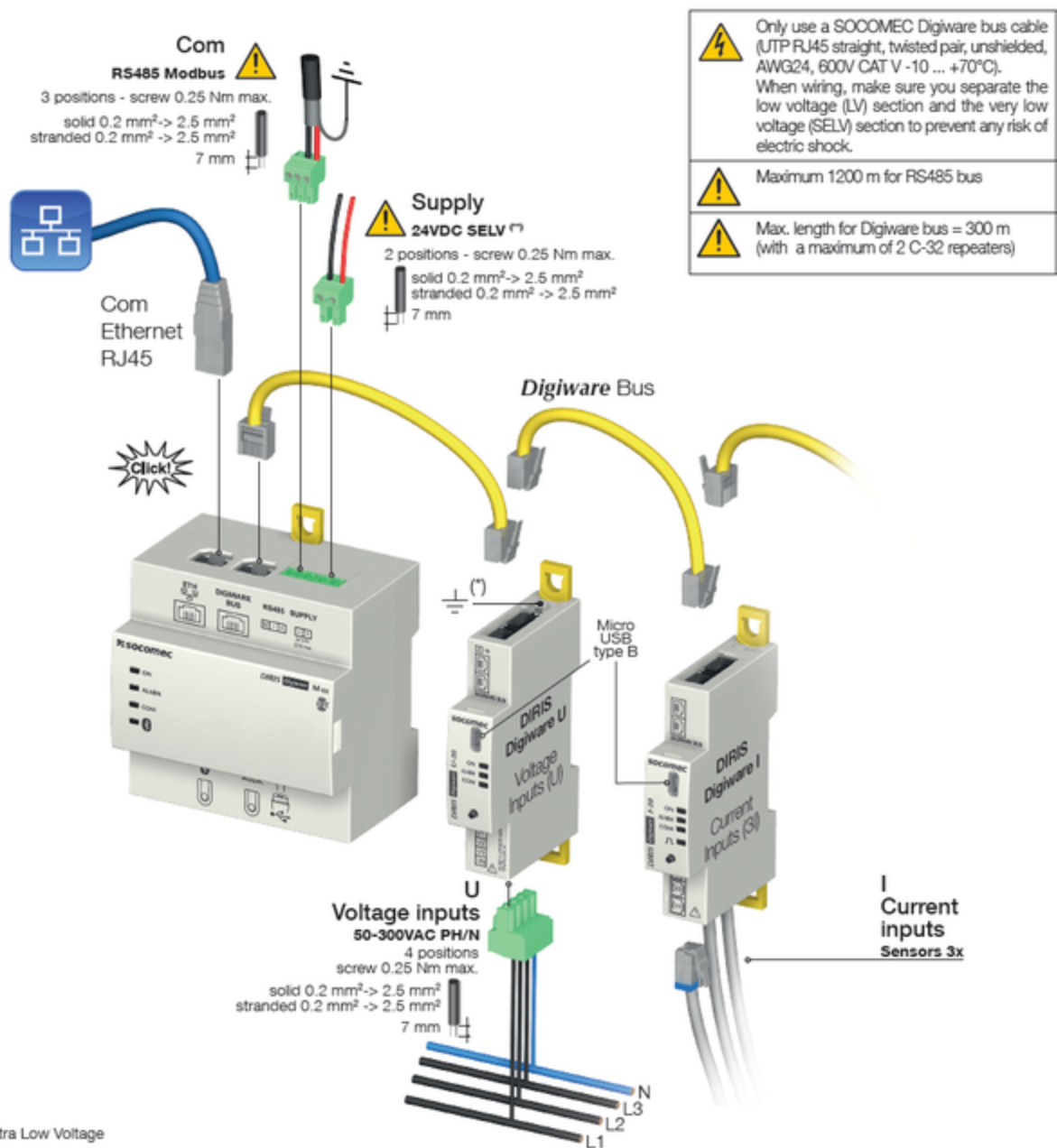
Step 2: Prepare your DIRIS Digiware M/D gateway

System wiring

Please refer to the user manual of the device to get all the details of commissioning. You can also look at [Configure your DIRIS Digiware system](#) video.



Before enabling the Azure connectivity from your gateway, please verify that all devices are correctly connected and configured (you can check it with the diagnostic page of the integrated web server).



Configuration with DPS (Device Provisioning Service)

i The configuration through DPS is provided in the Socomec gateways to ensure the complete compatibility with Azure platform.

The gateways can also be declared directly in the Azure IOTHub without need of DPS. In that case the provisioning mode "Manual" should be selected in the integrated web server (see below).

DPS with SAS Token authentication

First step is to retrieve the parameters from your DPS instance.

- **ID Scope** of the DPS
- **Registration ID** of the enrollment
- **Primary key** of the enrollment

Home >



ProvisioningCTD

Device Provisioning Service

Search (Ctrl+/)

Move Delete Refresh

Overview

- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems

Settings

- Quick Start
- Shared access policies
- Linked IoT hubs
- Certificates
- Manage enrollments
- Manage allocation policy
- Networking
- Properties
- Locks

Monitoring

- Alerts
- Metrics
- Diagnostic settings
- Logs

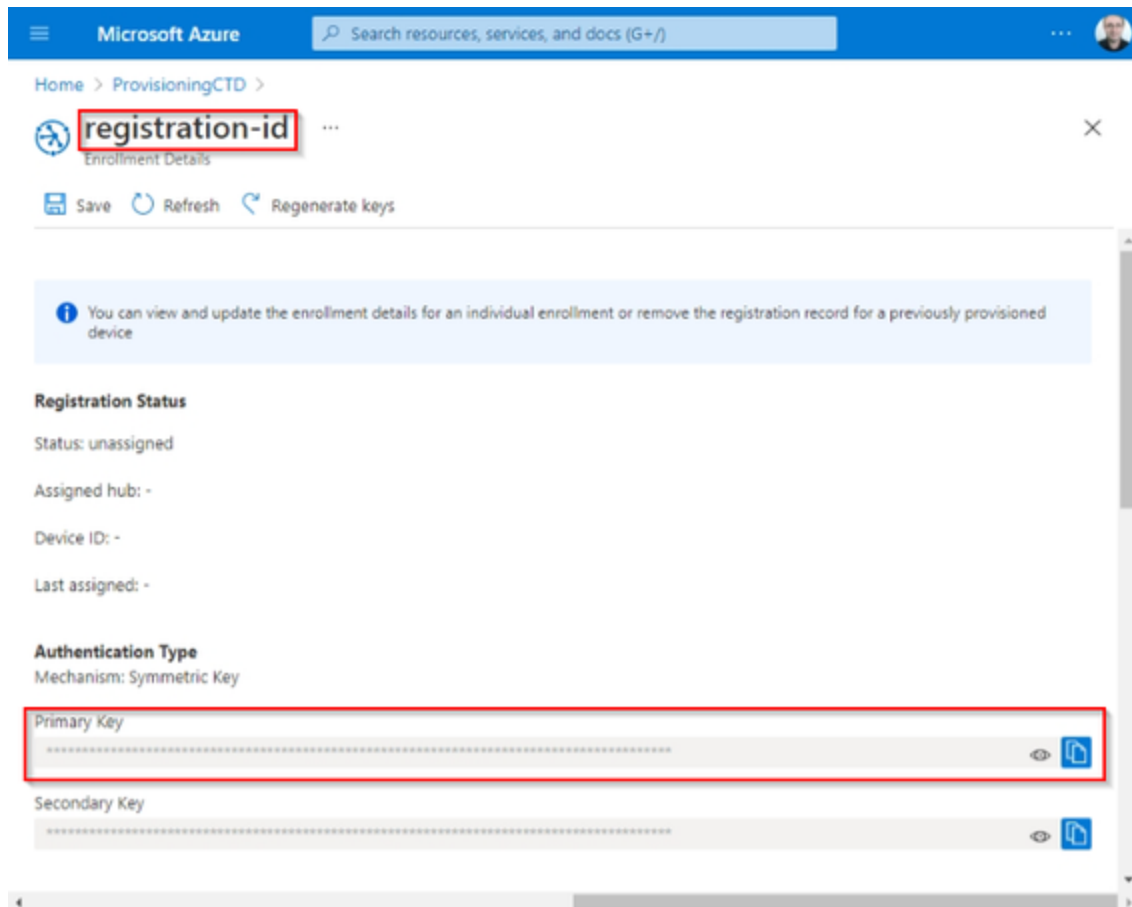
Essentials

JSON View

Resource group (Move)	: IOTDEV_CTD
Status	: Active
Location	: West Europe
Subscription (Move)	: CSP Azure IOT
Subscription ID	: 1adff130-f187-49a4-94ce-f88a07c24...
Service endpoint	: ProvisioningCTD.azure-devices-provi...
Global device endpoint	: global.azure-devices-provisioning.net
ID Scope	: 0ne0031AC78
Pricing and scale tier	: S1
Tags (Edit)	: Click here to add tags

Quick Links

- [Azure IoT Hub Device Provisioning Service Documentation](#)
- [Learn more about IoT Hub Device Provisioning Service](#)
- [Device Provisioning concepts](#)
- [Pricing and scale details](#)



Then you can go into the integrated web server of the Socomec's gateway to enter the settings accordingly. Log in the page with profile “**Administrator**” or “**Cyber**”, then go in **Settings>Protocols>Data Push**.

The screenshot shows the 'Azure Cloud connection' settings window. It has a blue header with a title bar and two buttons (a red 'X' and a green checkmark). The 'Enable Cloud' toggle is turned on. Below it, there are several settings: 'Provisioning mode' is set to 'Provisioning service', 'Id Scope' is '0ne0031AC78', 'Registration Id' is 'registration-id', 'Authentication' is 'Symmetric key', and 'Primary key' is masked with dots. The 'Synchronisation status' section shows 'State' as 'Not started'. At the bottom, there is a large blue button labeled 'Synchronize'.

Validate settings and click on “**Synchronize**” to start the synchronisation process with the DPS.

The push of telemetry data into the Azure IoT Hub will start automatically, according to the devices capabilities (energy meters, power measurements, ...).

DPS with X.509 certificate authentication

The configuration is similar than SAS Token authentication. Get the parameters from your DPS and set the Socomec's device parameters.

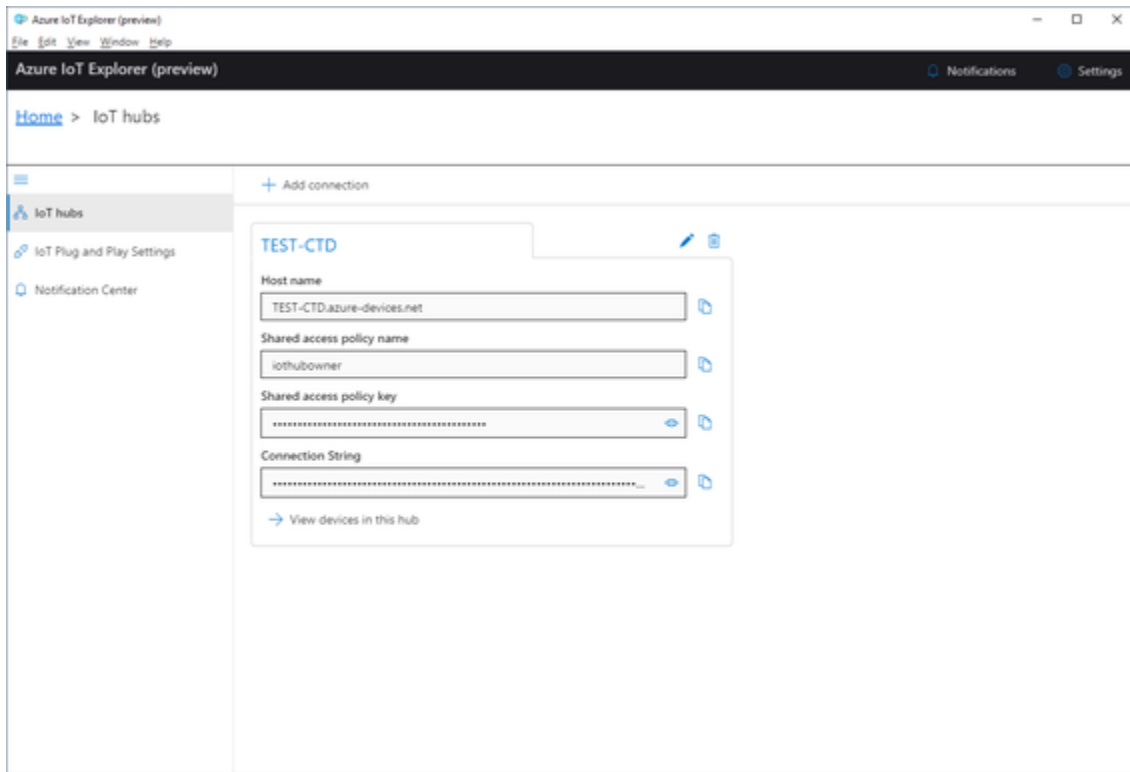
- **ID Scope** of the DPS
- **Registration ID** of the enrollment
- **Certificate** and **private key** generated from the certificate/key of the enrollment

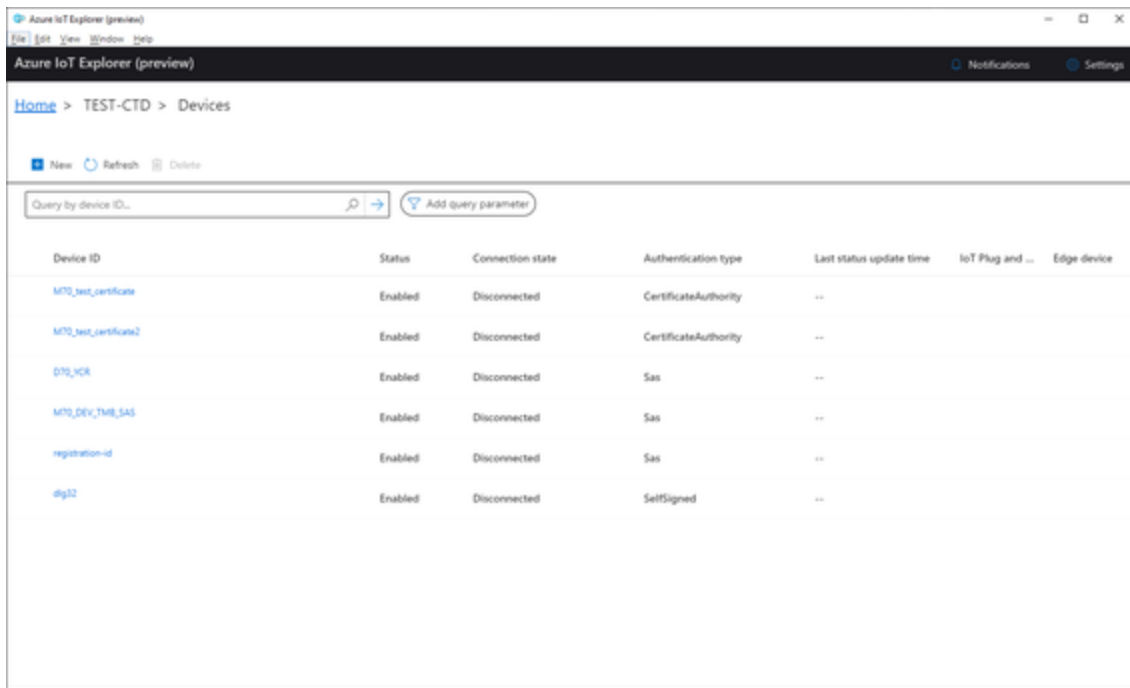
Step 3: Integration with Azure IoT Explorer

The Socomec gateways are compatible with Azure IoT Explorer. However, there is no "Direct method" or "Cloud to Device" capabilities on these gateways.

 Please refers to [Azure IoT Explorer documentation](#) to get all details about this software.

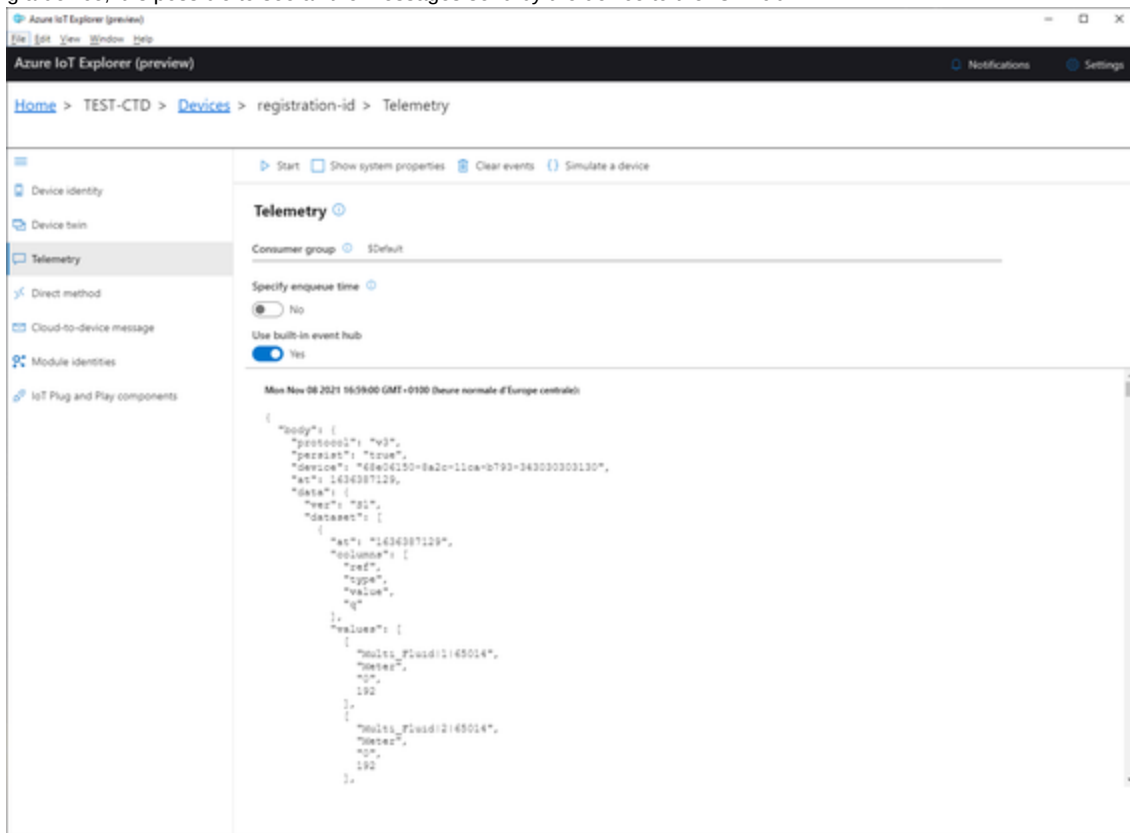
Connect Azure IoT Explorer to the Azure IOTHub





View telemetry messages

After selecting a device, it is possible to see all the messages send by the device to the IOTHub.

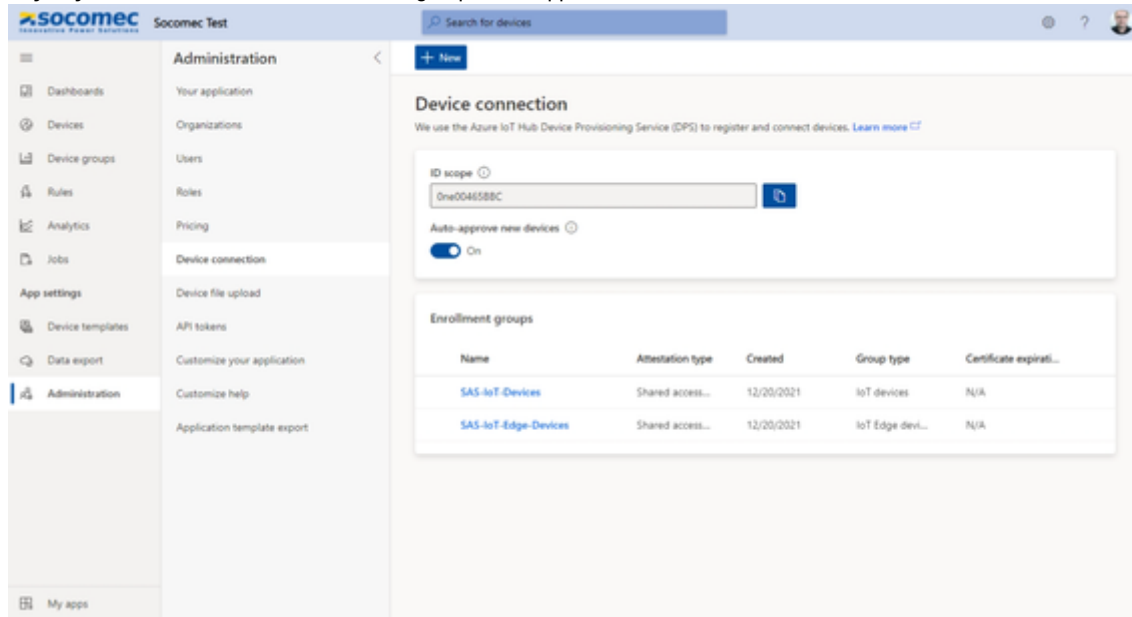


Step 4: Connect to Azure IoT Central

The Socomec gateways are fully compatible with Azure IoT Central.

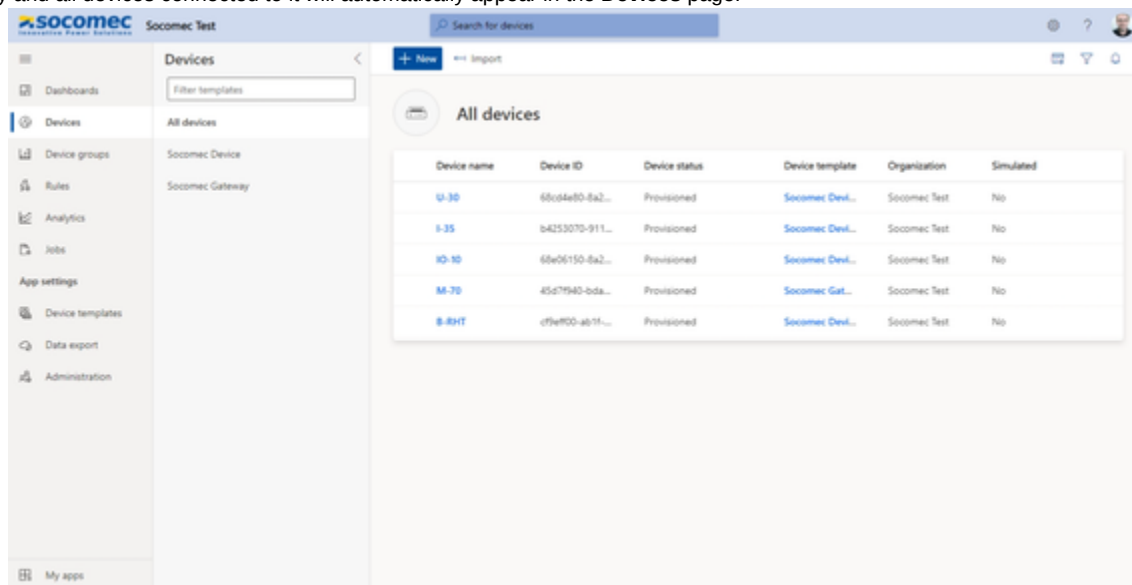
After creating an application into Azure IOT Central, get the connection parameters from **Administration>Device connection**:

- **ID Scope** of the application
- **Primary key** from SAS-IoT-Devices enrollment group of the application



Then you can go into the integrated web server of the Socomec's gateway to enter the settings accordingly. Validate settings and click on “**Synchronize**” to start the synchronisation process with the application DPS.

The gateway and all devices connected to it will automatically appear in the **Devices** page:



The push of **telemetry** and **property** data into the Azure IoT Hub will start automatically, according to the devices capabilities (energy meters, power measurements, ...). The **device templates** are retrieved from the public space, so the data will be automatically decoded by Azure IOT Central application.

The screenshot shows the Socomec Test application interface. The top navigation bar includes the Socomec logo, the title 'Socomec Test', a search bar for devices, and user controls. The left sidebar contains navigation links for Dashboards, Devices, Device groups, Rules, Analytics, Jobs, App settings, Device templates, Data export, and Administration. The main content area displays the 'Devices' section for a specific device 'I-35'. It shows the device is 'Disconnected' with the last data received on 1/13/2022 at 11:21:00 AM. Below this, there are tabs for 'Raw data' and 'Mapped aliases'. The 'Raw data' tab is active, showing a table of data points.

Timestamp ↓	Message type	Event creation time	Analogic alarm 1	Analogic alarm 2	Analogic alarm 3	Analogic alarm 4	Ea- Load 1	Ea- Load 2
1/13/2022, 11:2...	Telemetry							
1/13/2022, 11:2...	Telemetry							
1/13/2022, 11:2...	Telemetry						100230000	101693000
1/13/2022, 11:2...	Device disconn...							
1/13/2022, 11:1...	Telemetry							
1/13/2022, 11:1...	Device disconn...							
1/13/2022, 11:1...	Telemetry							
1/13/2022, 11:1...	Telemetry							
1/13/2022, 11:1...	Telemetry						100230000	101693000
1/13/2022, 11:1...	Device disconn...							
1/13/2022, 11:1...	Telemetry							

You can now customize the application with dashboards, analytics, etc...

Step 5: Additional Information

The Socomec's gateway send the message in JSON format. The complete description is available on demand, for any information please contact [Socomec's support team](#).

Step 6 : Additional Links

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

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