platform: {Windows10}
device: {Nise70}
language: {C#}

Connect Nise70 device to your Azure IoT services

Table of Contents

Introduction

About this document

This document describes how to connect Nise70 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 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.

Nise70 Key Features:

- Onboard Intel® Celeron® 6305E Tiger Lake-UP3 processor
- 4 x HDMI, 3 x USB 3.0, 1 x USB 2.0, 3 x LANs, 2 x COMs
- Onboard TPM 2.0 chip
- 1 x M.2 Socket for storage/4G LTE/5G
- 1 x mini-PCIe socket support mSATA/Wi-Fi/BT/4G LTE
- Support 12 to 24V DC input

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

- Prepare Nise70,and install Win10 IoT Enterprise.
- Power on the Nise70.
- Connect to the network.

Software Environmental setup

- Download the source code from this GitHub and check the “PNP_Xcare_Nise70” folder
- Install Visual Studio.
- Open the project

Prepare IoT Hub and DPS configuration

Please refer to this tutorial to complete the following procedures :

1. Use Azure commands or Azure portal to create a Resource Group which include Iot Hub and a Device Provisioning Service
2. To link the DPS instance to your IoT hub
3. To create your device by individual device enrollment in your DPS instance.
4. Make a note of the DPS information (DPS endpoint/Registration ID/ID Scope/Symmetric key).

Run the sample
Under the “PNP_Xcare_Nise70” folder, open the project and set debug parameter:
"(-s dps –i {scopeId} –d {deviceID} –k {primarykey} –e {endpoint}"

Connect to Azure IoT Central

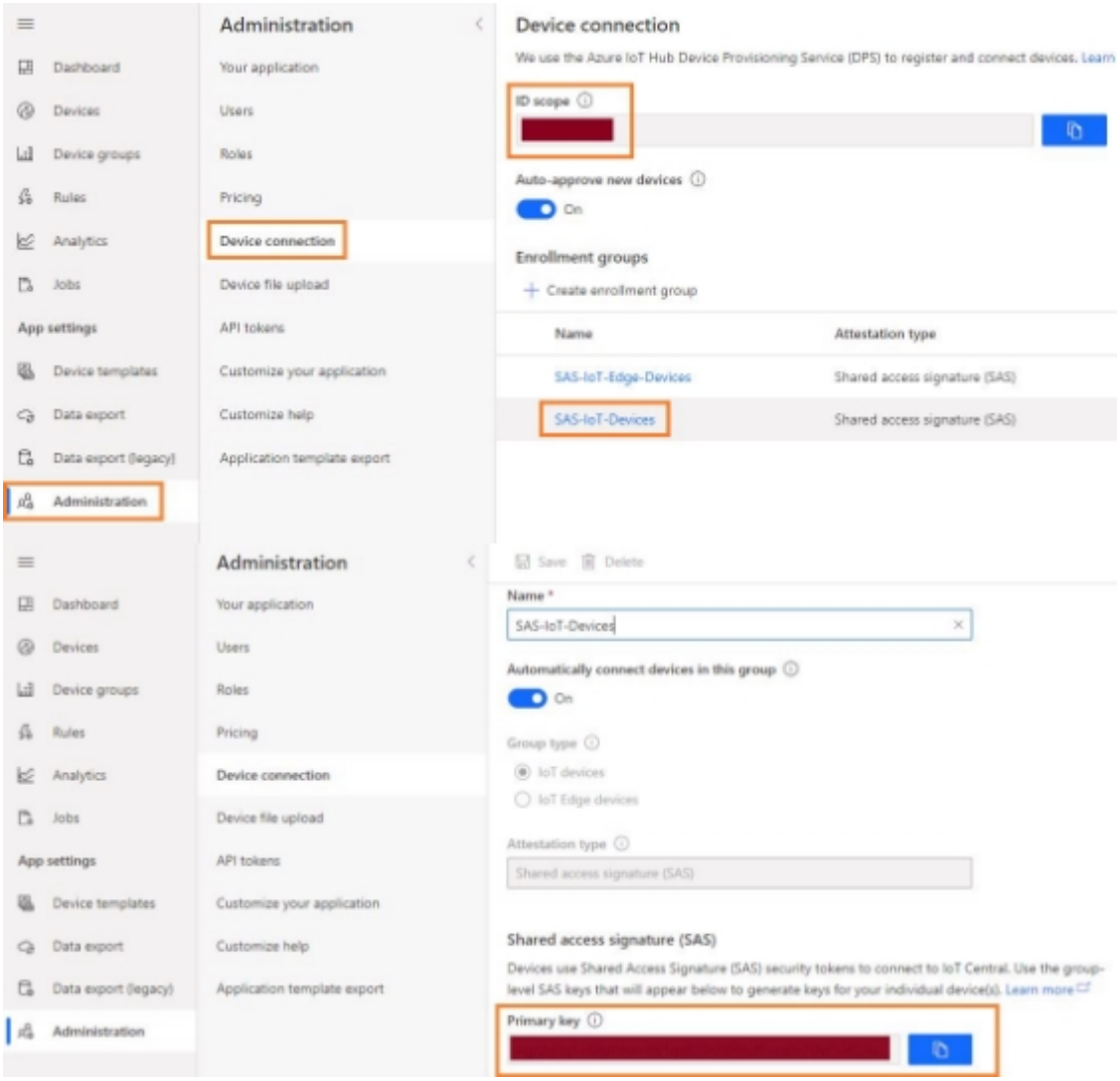
1. Create an application
   Please refer to this tutorial to create a “Custom application” template.

2. Create a device template from the device catalog
   Please refer to this tutorial to create the Nise70 device template.

3. Add a device
   Add a new device under Nise70 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.



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 <the group SAS primary key value>

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

Integration with Azure IoT Explorer (Advanced)

This section is optional for Advanced setup

- Include the steps on how to connect the IoT Plug and Play Device to Azure IoT Explorer
- Include screenshots and comments on how IoT Explorer shows/visualize telemetry , commands and properties coming from your IoT Plug and Play device.
- Include the steps on how to interact with devices (telemetry, commands properties)
- Ensure to attach the screenshot on consuming the device models available in public repository (not local folder) when using Azure IoT Explorer

Additional information

Put any additional information here such as alternative paths to deploy device application etc.

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