Connect PalThree device to your Azure IoT services

Table of Contents

Introduction

About this document

This document describes how to connect PalThree 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.

You shouldn't take more than 30 minutes to get through this guide.

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

Hardware

- A PalThree

- USB 2.0 A male to Micro USB male cable
- Ethernet cable RJ45 terminated

Software and tools

- Visual Studio 2022 (2017 should be OK to)
- .NET nanoFramework VS extension installed in Visual Studio
- nanoff tool installed

Prepare the Device.

Hardware

Make sure to connect your device to a PSU capable of delivering 9-12V, 1A, use the RJ45 cable to connect Ethernet and a micro USB cable to a PC.

The PalThree already has .NET nanoFramework firmware installed. You may want to check if it's running the latest version.

1. Install nanoff tool if you haven't done so before.

```
dotnet tool install -g nanoff
```

2. Run the tool to update the firmware

```
nanoff --update --target ORGPAL_PALTHREE
```

Clone the repo for the quickstart

Clone the following repo to download the sample code for this quick start.

```
gh repo clone github.com/OrgPal/Orgpal-Azure-IoT.git
```

Open the solution in Visual Studio

Open the PalThree-Azure-IoT solution in Visual Studio.

Store Azure Root CA certificate in the device

Uploading the Azure Root CA to the device make things easier (and simpler). Please follow the instructions on how to do this.

Create the X.509 certificates for the device

If you need to create device a certificate for your device you have follow the instructions here very closely.

Setup a DPS

Follow the commands to create a Resource Group, an Iot Hub and a Device Provisioning Service. We'll be using X.509 certificate with group enrolment. Please adjust according to your preferences, Azure subscription or geographical location.

Upload a CA certificate PEM file to an Azure IoT Hub Device Provisioning Service instance.

```
az iot dps certificate create --dps-name MyDps --resource-group MyResourceGroup --
name MyCertificate --path /certificates/Certificate.pem
```

Generate a verification code for a certificate in an Azure IoT Hub Device Provisioning Service instance. This verification code is used to complete the proof of possession step for a certificate. Use this verification code as the CN of a new certificate signed with the root certificates private key.

```
az iot dps certificate generate-verification-code --dps-name MyDps --resource-
group MyResourceGroup --name MyCertificate --etag AAAAAAAAAAA=
```

Verify a certificate by uploading a verification certificate containing the verification code obtained by calling generate-verification-code. This is the last step in the proof of possession process.

```
az iot dps certificate verify --dps-name MyDps --resource-group MyResourceGroup --
name MyCertificate --path /certificates/Verification.pem --etag AAAAAAAAAAA=
```

Generate a derived device SAS key for an enrollment group in an Azure IoT Hub Device Provisioning Service.

```
az iot dps enrollment-group compute-device-key -g My_Resource_Group --dps-name
MyDps --enrollment-id My_enrollment_Id --registration-id {registration_id}
```

Create an enrollment group in an Azure IoT Hub Device Provisioning Service.

```
az iot dps enrollment-group compute-device-key -g My_Resource_Group --dps-name
MyDps --enrollment-id My_enrollment_Id --registration-id {registration_id}
```

Create an enrollment group 'My_enrollment_group' in the Azure IoT provisioning service 'MyDps' in the resource group 'My_Resource_Group' using an intermediate certificate as primary certificate'.

```
az iot dps enrollment-group create -g My_Resource_Group --dps-name MyDps --
enrollment-id My_enrollment_group --certificate-path /certificates/Certificate.pem
```

Connect to Azure IoT Central (Simple)

To configure a device to 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.

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 replace it on the C# demo application.

Add a device from template: In your IoT Central application, navigate to Devices > New > Set Device name > Set Device ID > Device template select palthree_demo_0 > Create.

Connect device to IoT Central

In your IoT Central application, navigate to Devices > Select your device > Connect. Make a note of ID scope, Device ID and Primary key and and replace it on the C# demo application.

Integration with Azure IoT Explorer

Enter the connection string from Azure CLI and press Save. You will see a device symm-key-device.

```
az iot hub show-connection-string --name my-sample-hub --key primary --query
connectionString -o tsv
```

Telemetry Press the device name and choose the Telemetry Page. Press Start and wait a moment. You will see the telemetry data on the screen.

Property and Command You can show the device properties by clicking on the appropriate name. For example to check the firmware version click `fwVersion`. You can send commands to the device by clicking on the appropriate name. For example to blink the LED1 on the PalThree board click `blinkLed`.

Additional Links

Please refer to the below link for additional information

- 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
- Azure IoT library for .NET nanoFramework