
platform: Windows IoT Enterprise

device: VCO-6131E-4M2 Edge Inference Computer

language: C#

Contents

Introduction	1
Step 1: Prerequisites	2
Step 2: Prepare your Device	2
Create a device enrollment.....	2
Step 3: Build SDK and Run Samples	3
Set up Environment Variables.....	3
Build and Run Sample	6
Step 4: Integration with Azure IoT Explorer.....	6
Step 7: Additional Links.....	11

Introduction

About this document

This document describes how to connect VCO-6131E-4M2 Edge Inference Computer device running Windows IoT Enterprise with Azure IoT SDK. This multi-step process includes:

- Configuring Azure IoT Hub
- Registering your IoT device
- Provisioning your devices on Device Provisioning service
- Build and deploy Azure IoT SDK on device

Designed around 8th/9th Gen. Intel® Core™ processing and rich NVMe storage support, the VCO-6131E-4M2-P AI Edge Inference Computer enables hyperintelligence in challenging settings. The computer is purpose-built to train and apply AI, machine learning and inference analysis algorithms in harsh industrial, mobile and remote areas. Four hot-swappable NVMe trays allow high-speed storage to feed CPUs and integrated GPU performance accelerators for data immediacy to support intelligence applications at the rugged edge. The AI Edge Inference Computer supports 10GbE, 4G LTE, 5G, Wi-Fi 6 for fast wireless connectivity. Four USB 3.2 Gen 2 ports provide 10 Gbps connection to high-speed peripherals. A hot-swappable GPU blower fan streamlines maintenance and replacement. TPM 2.0 provides hardware security and enables cybersecurity features.

Premio AI Edge Inference Computers certified for Microsoft Azure IoT enable advanced intelligence for insights and optimization in harsh surroundings.

Step 1: Prerequisites

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

1. [Setup your IoT hub and Provision your device over DPS](#)
2. Make sure you have the [.NET Core 2.1 SDK or later](#) installed on your machine.
3. Install [git](#). Confirm git is in your PATH by typing git version from a command prompt.
4. Download and install the [Azure IoT Hub Explorer](#).

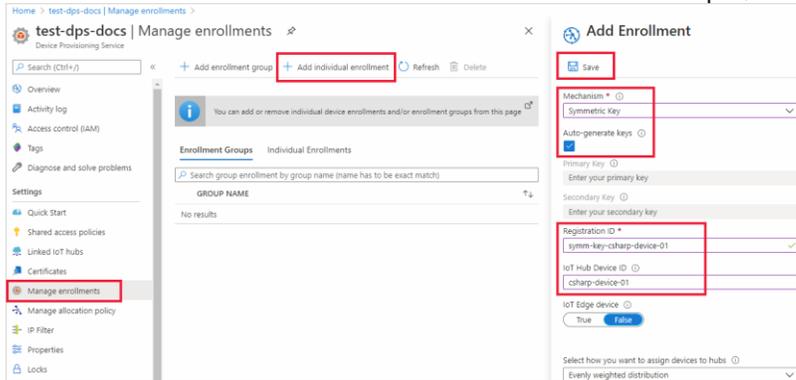
Step 2: Prepare your Device

- Connect power to device
- Connect device to the Internet via Ethernet, Wi-Fi, or cellular
- Connect device to a display
- Power on and initialize Windows IoT operating system

Create a device enrollment

1. Sign in to the [Azure portal](#), select the **All resources** button on the left-hand menu and open your Device Provisioning service (DPS) instance.
2. Select the **Manage enrollments** tab, and then select the **Add individual enrollment** button at the top.
3. In the **Add Enrollment** panel, enter the following information, and press the **Save** button.
 - **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, **symm-key-csharp-device-01**.
- **IoT Hub Device ID:** Enter a device identifier. For example, **csharp-device-01**.

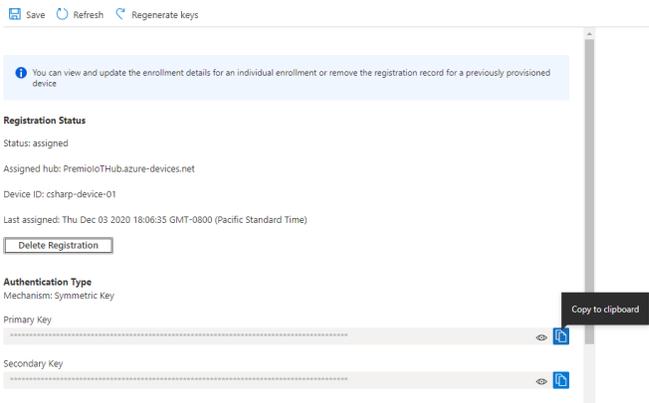


4. Once you have saved your enrollment, the **Primary Key** and **Secondary Key** will be generated and added to the enrollment entry. Your symmetric key device enrollment appears as **symm-key-csharp-device-01** under the *Registration ID* column in the *Individual Enrollments* tab.
5. Open the enrollment and copy the value of your generated **Primary Key** and **Secondary Key**. You will use this key value and the **Registration ID** later when you add environment variables for use with the device provisioning sample code.

Step 3: Build SDK and Run Samples

Set up Environment Variables

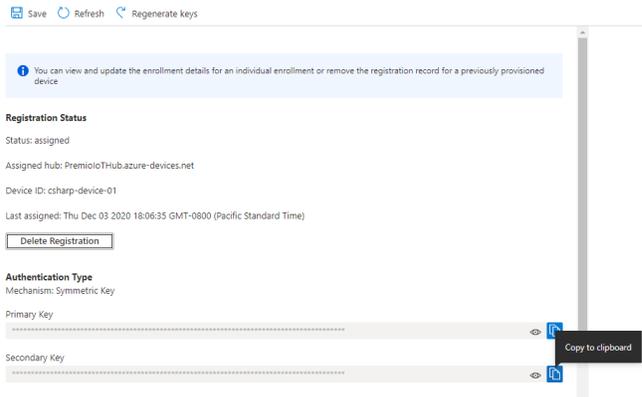
1. You will then need to set up the proper environment variables based on the information from the “Create a device enrollment” section. First copy the Primary Symmetric Key. You can go to your Azure portal and go to the Device Provisioning Service that you created. Then go to Manage enrollments and select the Individual Enrollments tab. You should see the device that you registered. Click on the device and then scroll down until you see the Primary Key. Copy that value as shown below:



- Go back to your command prompt and type out the following command:
`setx PRIMARY_SYMMETRIC_KEY "{paste in your primary key}"`

Be sure to remove the curly braces but paste in the primary key value within double quotes.

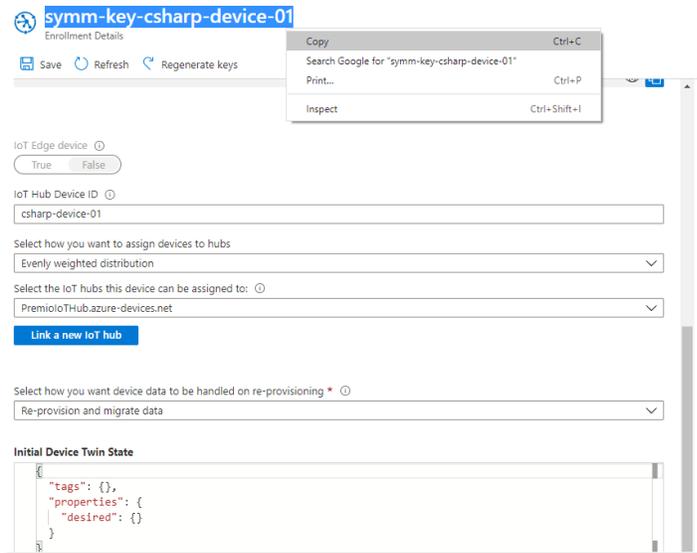
- Go back to the Azure portal screen and copy the Second Key as shown below:



- Go back to your command prompt and type out the following command:
`setx SECONDARY_SYMMETRIC_KEY "{paste in your secondary key}"`

Be sure to remove the curly braces but paste in the secondary key value within double quotes.

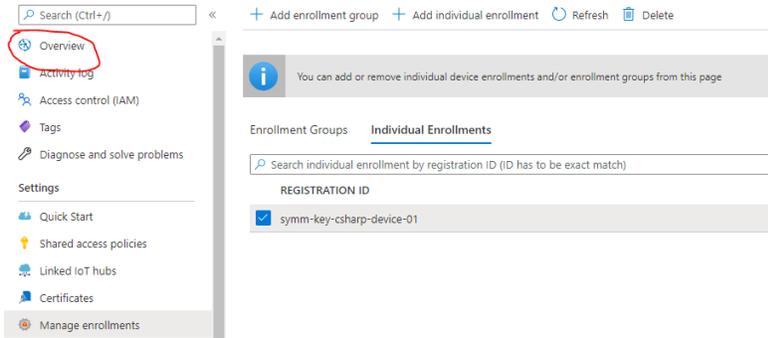
- Go back to your Azure portal and copy the Registration ID. Here we use the same Registration ID that we used in the "Create a device enrollment" section.



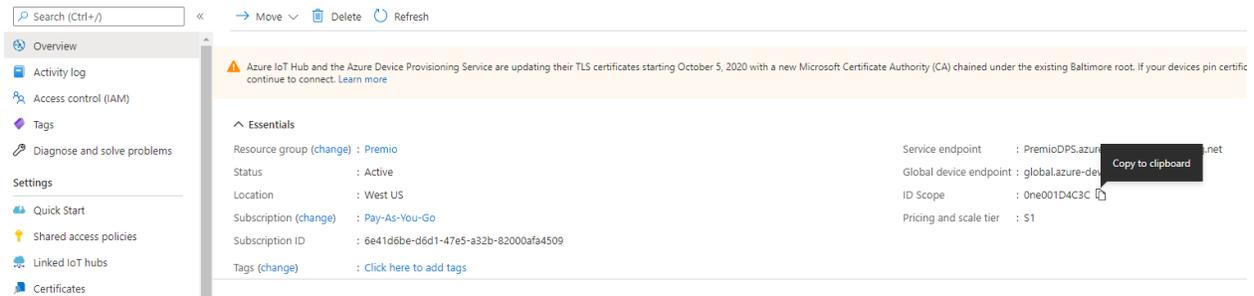
- Go back to your command prompt and type out the following command:
`setx PROVISIONING_REGISTRATION_ID "{paste in the device registration ID}"`

Be sure to remove the curly braces but paste in the registration ID value within double quotes.

- 7. Go back to your Azure portal and close the enrollment details of the device and go back to the Device Provisioning Service that you created. Next, click on the Overview link.



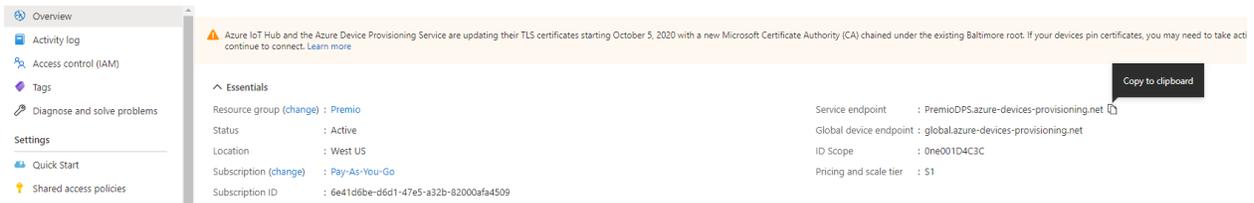
- 8. In the Overview screen, copy the ID Scope value as shown below.



- 9. Go back to your command prompt and type out the following command:
`setx DPS_IDSCOPE "{paste in the ID Scope}"`

Be sure to remove the curly braces but paste in the ID Scope value within double quotes.

- 10. Go back to the Azure portal and copy the Service endpoint value.



- 11. Go back to your command prompt and type out the following command:
`setx DPS_SERVICE_ENDPOINT "{paste in the Service endpoint}"`

Be sure to remove the curly braces but paste in the Service endpoint value within double quotes.

12. Close the command prompt window. (In order for the environment variables to work, a new command prompt window must be opened).

Build and Run Sample

1. If you have not done so yet, download and install [.NET Core 2.1 SDK or later](#) onto your device.
2. Open up a new command prompt. Navigate to the area on your drive of where you wish to download the source code. Next, download the source code of the PremioIoTHubTester by running the following command from a command prompt:

```
git clone https://premiopmg@dev.azure.com/premiopmg/PremioIoTHubTester/ git/PremioIoTHubTester
```

3. CD into the PremioIoTHubTester directory and execute the following command from the command prompt:

```
dotnet run
```

Your firewall settings may block network access to the PremioIoTHubTester. Be sure to allow the program access.

4. If the program is running successfully, you should see the command prompt display the messages as shown below:

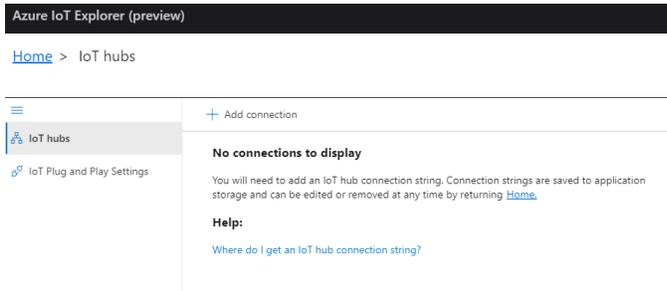
```
D:\PremioIoTHubTester>dotnet run
Provisioning Device Client...
RegistrationID = symm-key-csharp-device-01
ProvisioningClient RegisterAsync . . . Assigned
ProvisioningClient AssignedHub: PremioIoTHub.azure-devices.net; DeviceID: csharp-device-01
Creating Symmetric Key DeviceClient authentication
Opening client...
Sending message to test connection...
Connection testing complete.

Enter selection:
1 - Send messages from device to cloud (D2C)
2 - Read messages from cloud to device (C2D)
M or Return for Menu
0 - Exit
>
```

Step 4: Integration with Azure IoT Explorer

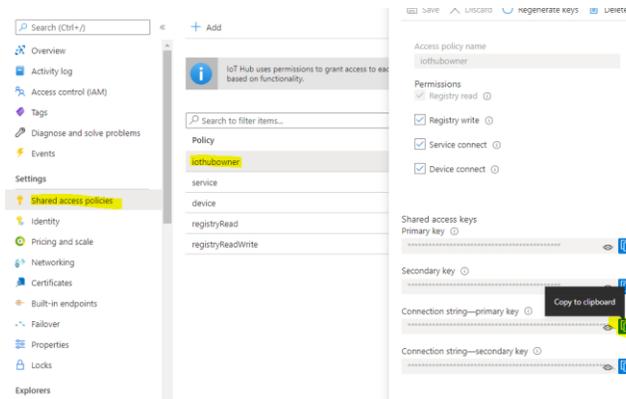
1. If you have not yet done so, download and install the [Azure IoT Explorer](#). It is suggested for simplicity, download and install the latest Azure.IoT.Explorer.preview.x.x.x.msi file.

2. After installing the Azure IoT Explorer, you should see the following screen.



3. Click on Add connection.

4. You will need to add the connection string to your IoT Hub. Go back to your Azure portal and then navigate to your IoT Hub. Click on Shared access policies and then iothubowner. You should see on the right pane the Connection string-primary key. Copy that value.



5. Go back to the IoT Explorer and paste the connection string into the Connection string text box and then click Save.

Add connection string

Connection string *

```
HostName=PremioIoTHub.azure-devices.net;SharedAccessKeyName=iothubowner;SharedAccessKey=
```

Where do I get an IoT hub connection string?
Please do not save your hub connection string to any unsafe locations

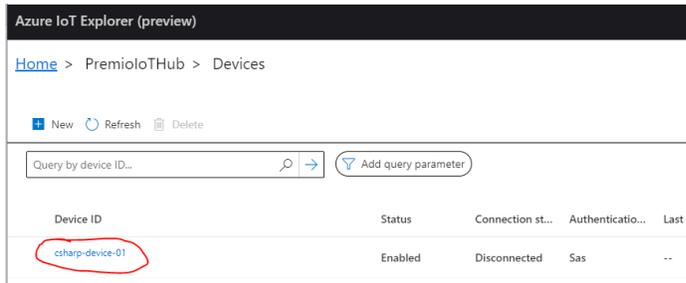
Host name
PremioIoTHub.azure-devices.net

Shared access policy name
iothubowner

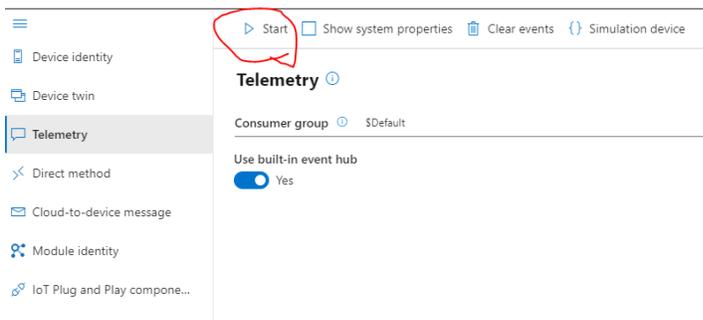
Shared access policy key
.....

Save Cancel

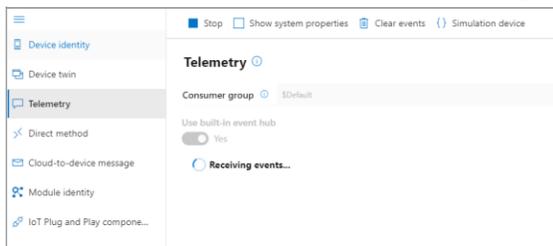
6. You should see the device that was provisioned. Click on the device as shown below:



7. We will now attempt to send messages from the device to the cloud. On the left pane, click on Telemetry. Afterwards, click start as shown below:



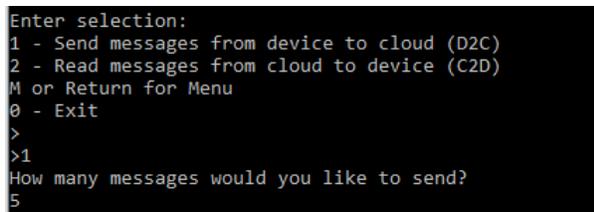
8. You should see the IoT Explorer listening for messages as shown below:



9. Go back to your command prompt that is running the PremioIoTHubTester program. Type in selection 1 and press Enter.



10. Let's send 5 messages from the device to the cloud. Type 5 and then press Enter.

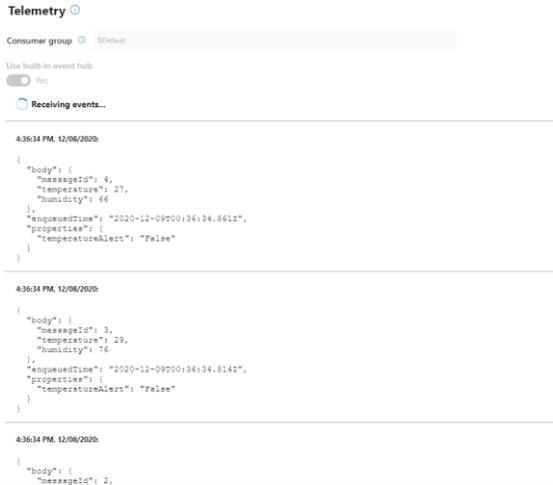


11. You should see something similar to what’s shown below. Five messages should have been sent to the cloud.

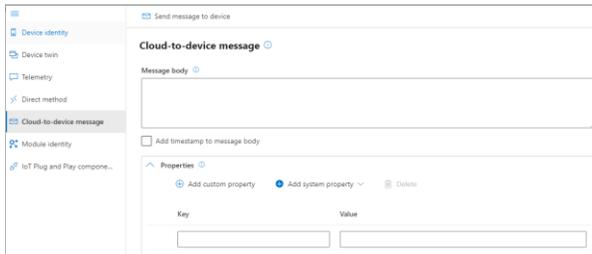
```
How many messages would you like to send?
5
Device sending 5 messages to IoT Hub...

12/8/2020 4:36:35 PM> Sending message: 0, data: [{"messageId":0,"temperature":29,"humidity":71}]
12/8/2020 4:36:35 PM> Sending message: 1, data: [{"messageId":1,"temperature":23,"humidity":72}]
12/8/2020 4:36:35 PM> Sending message: 2, data: [{"messageId":2,"temperature":34,"humidity":75}]
12/8/2020 4:36:35 PM> Sending message: 3, data: [{"messageId":3,"temperature":29,"humidity":76}]
12/8/2020 4:36:35 PM> Sending message: 4, data: [{"messageId":4,"temperature":27,"humidity":66}]
```

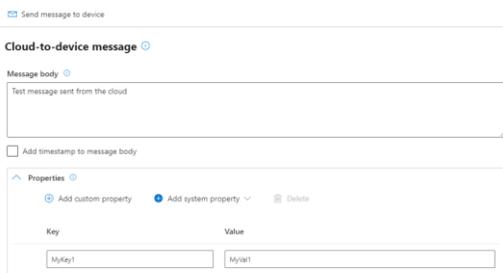
12. Go back to the Azure IoT Explorer. You should see the same messages there. This indicates that the IoT Hub did indeed receive the 5 messages sent from the device.



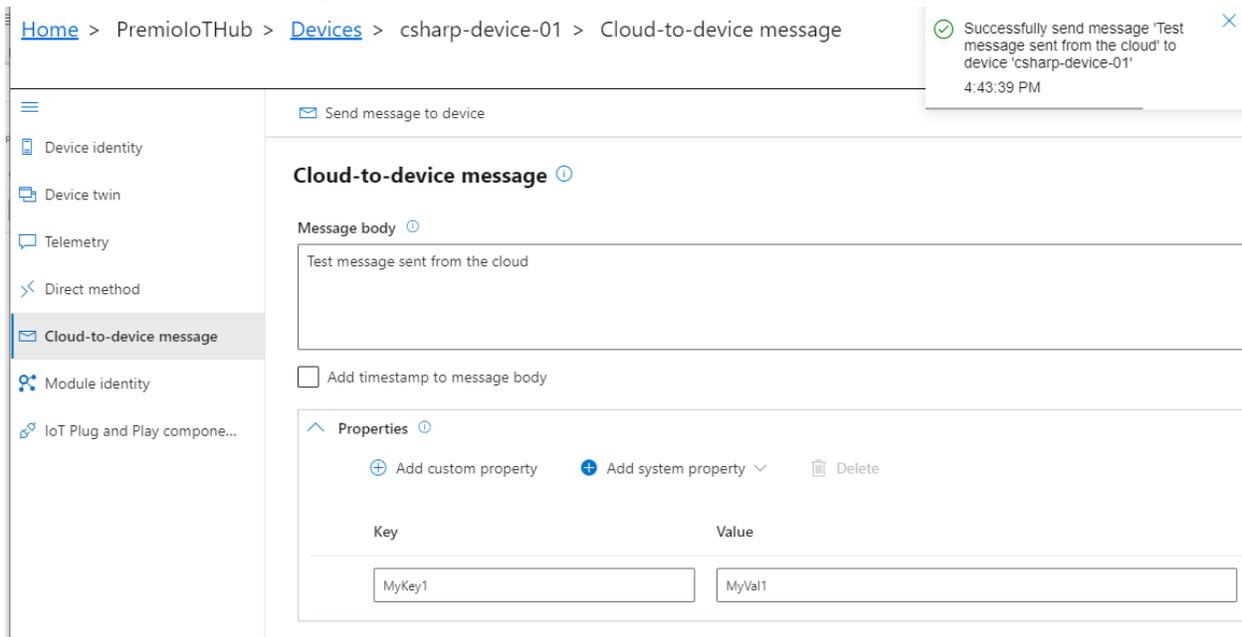
13. Let’s send messages from the cloud to the device. In the Azure IoT Explorer, go to Cloud-to-device message which is located on the left pane. You should see the following:



14. In the Message body, type in “Test message sent from the cloud”. In the Key field, type in “MyKey1”. For the Value field, type in “MyVal1”. It should look similar to below:



15. Next, click “Send message to device”. You should receive the following:



16. Go back to the command prompt that is running the PremioloTHubTester. To receive the messages from the cloud onto the device, type in option 2 and press Enter as shown below:

```
>2
Use the Azure Portal IoT Hub blade or Azure IoT Explorer to send a message to this device.
The device will then process each message and mark them completed.
Once there are no message left to process it will quit.
Press Enter to start processing messages.
```

17. Press Enter to actually receive and process the messages. After pressing Enter you should receive the following:

```
Press Enter to start processing messages.
Checking for messages (takes about a minute)...
12/8/2020 4:44:17 PM> Received message: Test message sent from the cloud
Property[0> Key=MyKey1 : Value=MyVal1
12/8/2020 4:44:47 PM> No messages left to process.
The device completed listening for messages.
Press Enter to continue.
```

18. Once it has completed processing the message(s), Press Enter.

Step 7: Additional Links

Please refer to the below links for additional information

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