

# Connect to Azure IoT Hub with the MGate 5105 Industrial Protocol Gateway

Moxa Technical Support Team  
[support@moxa.com](mailto:support@moxa.com)

## Contents

- 1. Introduction..... 2**
- 2. System Topology..... 2**
- 3. Prerequisites..... 3**
  - 3.1 Modbus Slave Tool..... 3
  - 3.2 Device Explorer..... 3
  - 3.3 Create Azure IoT Hub and Device ..... 3
- 4. MGate 5105 Settings..... 4**
  - 4.1 Protocol Conversion..... 4
  - 4.2 Modbus RTU Master Settings..... 5
  - 4.3 Azure Device Settings..... 5
  - 4.4 Serial Settings .....11
- 5. Modbus Slave Tool Setting ..... 11**
- 6. Device Explorer Settings ..... 12**
- 7. Communication Test ..... 13**
  - 7.1 Device-to-cloud message ..... 13
  - 7.2 Cloud-to-device message ..... 14

---

Copyright © 2019 Moxa Inc.

Released on March 30, 2019

### About Moxa

Moxa is a leading provider of edge connectivity, industrial networking, and network infrastructure solutions for enabling connectivity for the Industrial Internet of Things. With over 30 years of industry experience, Moxa has connected more than 50 million devices worldwide and has a distribution and service network that reaches customers in more than 70 countries. Moxa delivers lasting business value by empowering industry with reliable networks and sincere service for industrial communications infrastructures. Information about Moxa’s solutions is available at [www.moxa.com](http://www.moxa.com).

### How to Contact Moxa

Tel: +886-2-8919-1230  
Fax: +886-2-8919-1231



## 1. Introduction

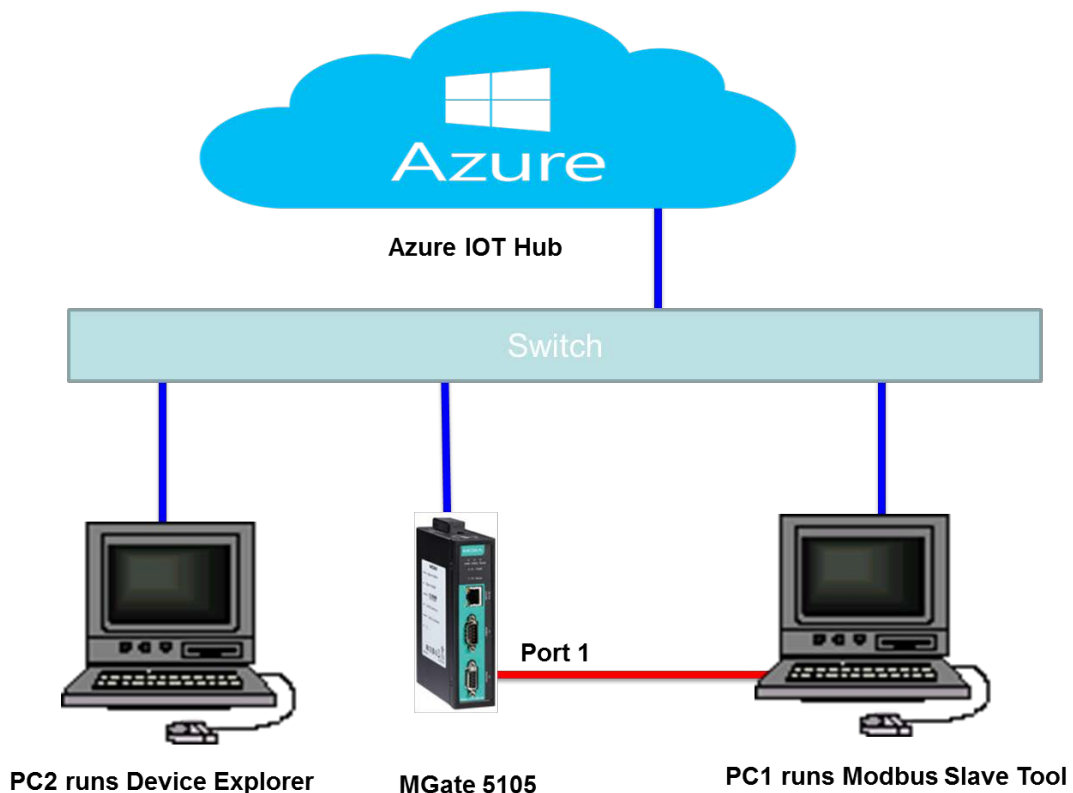
The MGate 5105 performs easy protocol conversions between Modbus RTU/ASCII, Modbus TCP, and EtherNet/IP protocols. From Firmware Version 4.0, it can support data that is collected from these fieldbus protocol devices and pushed to a cloud server. The cloud server could be Microsoft Azure, Alibaba Cloud, or MQTT Broker.

This document demonstrates how to use the MGate 5105 that connects to the Microsoft Azure IoT Hub and publishes fieldbus data messages to this IoT Hub or receives messages from it.

## 2. System Topology

Figure 1 illustrates the system topology. PC1 runs Modbus Slave tool to act as a Modbus RTU device. It connects to the MGate 5105's Port 1. The MGate 5105 acts as an Azure IoT device and connects to Azure IoT Hub. PC2 runs Device Explorer to monitor Azure IoT devices in the IoT Hub.

< Figure 1. System Topology >



### 3. Prerequisites

#### 3.1 Modbus Slave Tool

[Modbus Slave](#) is a very popular Modbus slave simulator for testing and debugging of your modbus devices, which support Modbus RTU/ASCII and Modbus TCP/IP.

Download from website: <http://www.modbustools.com/download.html>

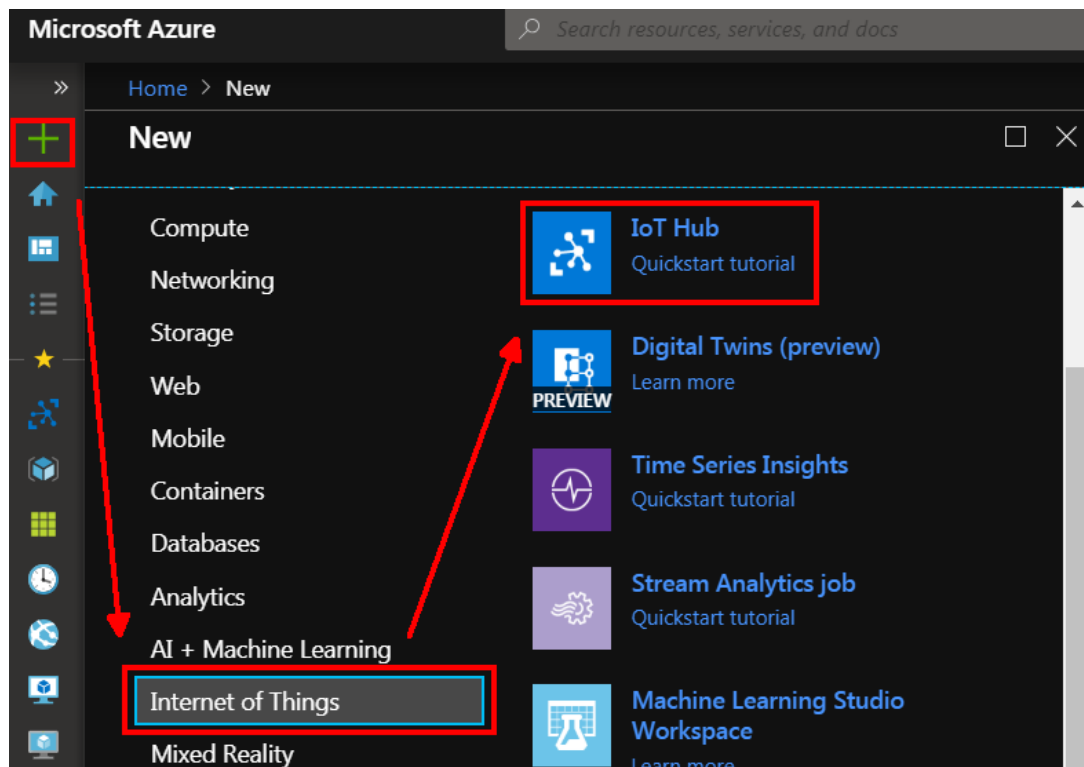
#### 3.2 Device Explorer

The Device Explorer is the gadget released by Azure to monitor “Azure IoT Hub” service. You can download a prebuilt version of the Device Explorer application. The latest release can be found at the following website:

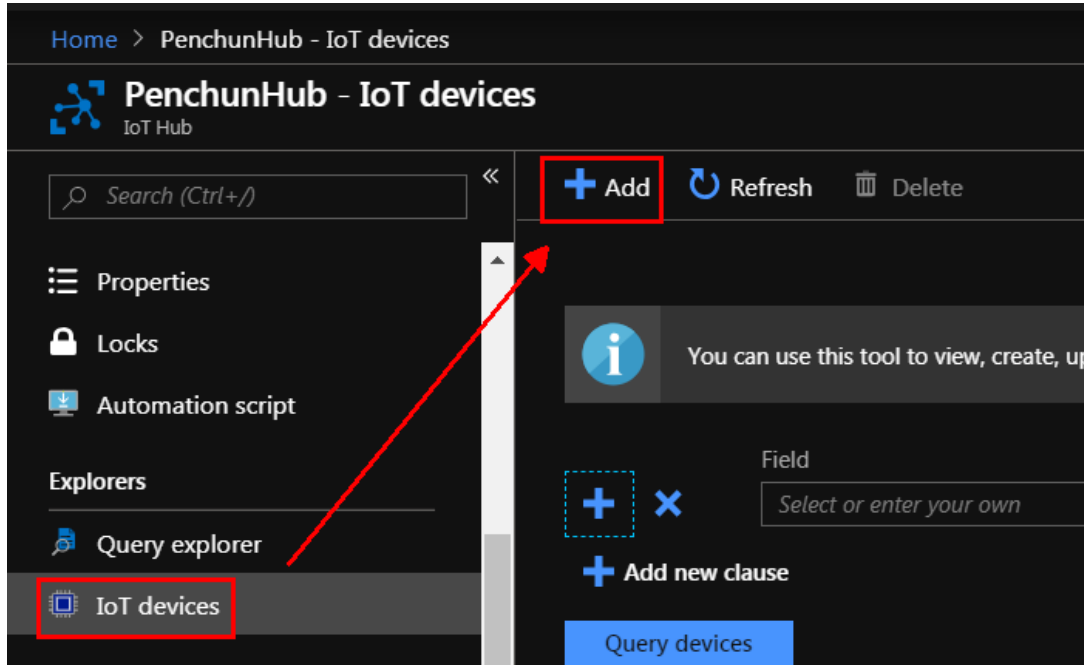
<https://github.com/Azure/azure-iot-sdk-csharp/releases/tag/2019-1-4>

#### 3.3 Create Azure IoT Hub and Device

1. Use Microsoft user account to log in to Azure Portal  
Azure Portal website: <https://portal.azure.com>
2. Create a New IoT Hub: New → Internet of Things → IoT Hub



- When a new IoT Hub is created, click it to create a new IoT Device: Explorers → IoT device → Add.



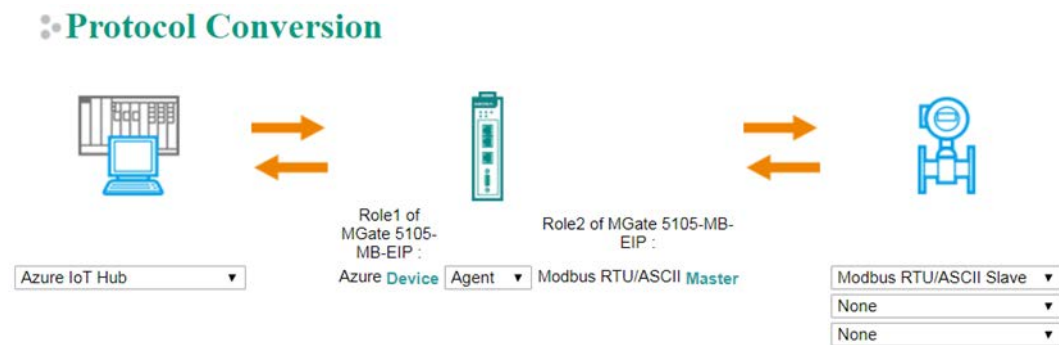
## 4. MGate 5105 Settings

Log in to MGate 5105’s web console, then do the following settings:

### 4.1 Protocol Conversion

Under “Protocol Conversion” settings, we choose “Azure Device” as Role 1. In the fieldbus site, we can choose the following protocols: Modbus RTU/ASCII Slave, Modbus TCP Server, or EtherNet/IP Adapter, and multiple combinations are allowed for settings in Role 2. In this demonstration, we choose “Modbus RTU/ASCII Slave”.

Set as below:



### 4.2 Modbus RTU Master Settings

1. In the "Modbus RTU/ASCII Master Settings" web page, under Mode, we choose "RTU" and keep the "Master Settings" as the default setting.
2. Add a "Read1" modbus command to send a function code 03, quantity as 1 command, Endian Swap as Byte. Its "Poll interval" is 1000 ms.
3. Add a "Write1" modbus command to send a function code 06 command, Endian Swap as Byte. Its command "Trigger" is by "Data Change".

Set as below:

Role: Master  
 Mode: RTU

**Master Settings**

Initial delay: 0 (0 - 30000 ms)  
 Max. retry: 3 (0 - 5)  
 Response timeout: 1000 (10 - 120000 ms)  
 Inter-frame delay: 0 (10 - 500 ms, 0: default)  
 Inter-character timeout: 0 (10 - 500 ms, 0: default)

**Modbus Commands**

Index	Name	Slave ID	Function	Address / Quantity	Trigger	Poll Interval	Endian Swap
1	Read1	1	3	Read address 0, Quantity 1	Cyclic	1000	Byte
2	Write1	1	16	Write address 0, Quantity 1	Data Change	N/A	Byte

### 4.3 Azure Device Settings

1. Device connection string:  
 In the Basic Settings → Device connection string, you must fill in your IoT device connection string. It can be found at "IoT Device details" within Azure Portal as below:

Home > PenchunHub - IoT devices > Device details

**Device details**  
 MGate5105

Save Message to device Direct method Device twin Add module identity Regenerate keys Refresh

Device Id: MGate5105

Primary key: i4sJeh2RXd91CjKa8+6pDix9EC7bNYrs+tr5wUy3on8=

Secondary key: cpDMo69CGEo+QpzKkXmc5UhRaVzaPylCW30SKWPgDWk=

Connection string (primary key): **HostName=PenchunHub.azure-devices.net;DeviceId=MGate5105;SharedAccessKey=i4sJeh2...**

Connection string (secondary key): HostName=PenchunHub.azure-devices.net;DeviceId=MGate5105;SharedAccessKey=cpDMo...

Set as below:

<b>Role</b>	Device
<b>Basic Settings</b>	
<b>Device connection string</b>	HostName=PenchunHub.azure-devices.net;DeviceId=MGate5105;Sha

2. Device-to-cloud Messages:

Click the **Add** button to create a Device-to-cloud Message, then click it to edit message settings.

**Device-to-cloud Messages**

+ Add   Edit   Delete

<b>Message ID</b>
-------------------

In "Pair Settings", we click "Message ID" to edit its Name as "Read1".

Pair Settings

Type

- Message ID
- Message Version
- Gateway ID
- Date Time
- Tag Status Monitoring

Conditional User Information

Message ID

Name: msgID

Value: Read1

OK

Cancel

Click "Date Time" to enable "dateTime" padding in the message.

Pair Settings

Type

- Message ID
- Message Version
- Gateway ID
- Date Time
- Tag Status Monitoring

Conditional User Information

Constant User Information

Date Time

Pair: Enable

Name: dateTime

Value: Example: 1990-01-02T03:04:05+06:00

OK

Cancel

Click **Add → Module** to create a new module.

Pair Settings

+ Add ✎ Edit 📄 Clone 🗑 Delete

Type	Module	Name
Message ID	Tag	msgID
Message Version		msgVer
Gateway ID		gwID
Date Time		dateTime
Tag Status Monitoring		validTag

Select "ModuleR1" under Name.

**Module**

**Name**

OK  
Cancel

Choose "ModuleR1", then click **Add → Tag**.

+ Add ✎ Edit 📄 Clone 🗑 Delete

Type	Module	Name
Message ID	Tag	msgID
Message Version		msgVer
Gateway ID		gwID
Date Time		dateTime
Tag Status Monitoring		validTag
- Module		ModuleR1

Create a Protocol Tag as below:

**Protocol Tag**

Name

Data unit

Unit quantity

Endian swap

Onchange trigger

Trigger deadband

3. Cloud-to-device Messages:

Click the **Add** button to create a Cloud-to-device message and then click it to edit message settings.

**Cloud-to-device Messages**

**Message ID**

In "Pair Settings", we click "Message ID" to edit its Name as "Write1".

**Message ID**

Pair Settings

Type

Message ID

**Message ID**

Name

Value

Click **Add → Module** to create a new module.

**Pair Settings**

Type		Name
Message ID	Module	msgID
Message Version	Tag	msgVer
Gateway ID		gwID



Select "ModuleW1" as Name.

Module

Name

OK Cancel

Choose ModuleW1 and then click **Add → Tag**.

Type	Module	Name
Message ID		msgID
Message Version		msgVer
Gateway ID		gwID
- Module		ModuleW1

+ Add Edit Clone Delete

Tag

Create a Protocol Tag as below:

Protocol Tag

Name

Data unit

Unit quantity

Endian swap

OK Cancel

### 4.4 I/O Data Mapping


When the protocol settings are done, you just need one more step of I/O Data mapping for protocol configuration. You can click the **Make a proposal** button for auto mapping in both the **Azure IoT Hub → Fieldbus Slave** direction and **Azure IoT Hub ← Fieldbus Slave** direction.

#### ⚙️ I/O Data Mapping


Data flow direction    Azure IoT Hub --> Fieldbus Slave ▾

---


Mapping address arrangement    Make a proposal!




Your device :  
Azure IoT Hub




**write**





**write**



Your device :  
Fieldbus Slave


**Role 1 of MGate 5105-MB-**  
EIP :  
Azure Device

Name	Internal Address	Data Size
Write1.ModuleW1.TagW1	N/A .. N/A	2


**Role 2 of MGate 5105-MB-**  
EIP :  
Fieldbus Master

Protocol	Name	Internal Address	Data Size
Unselected	Unselected	N/A .. N/A	0


The mapping result is as below:




Your device :  
Azure IoT Hub




**write**





**write**




Your device :  
Fieldbus Slave

**Role 1 of MGate 5105-MB-**  
EIP :  
Azure Device


Name	Internal Address	Data Size
Write1.ModuleW1.TagW1	0 .. 1	2

**Role 2 of MGate 5105-MB-**  
EIP :  
Fieldbus Master


Protocol	Name	Internal Address	Data Size
Modbus RTU/ASCII Master	Write1	0 .. 1	2




Your device :  
Azure IoT Hub




**read**





**read**



Your device :  
Fieldbus Slave

**Role 1 of MGate 5105-MB-**  
EIP :  
Azure Device

Name	Internal Address	Data Size
Read1.ModuleR1.TagR1	0 .. 1	2

**Role 2 of MGate 5105-MB-**  
EIP :  
Fieldbus Master

Protocol	Name	Internal Address	Data Size
Modbus RTU/ASCII Master	Read1	0 .. 1	2

## 4.5 Serial Settings

Serial Port1 connects to Modbus RTU device, so you must set the serial parameters of Port1.

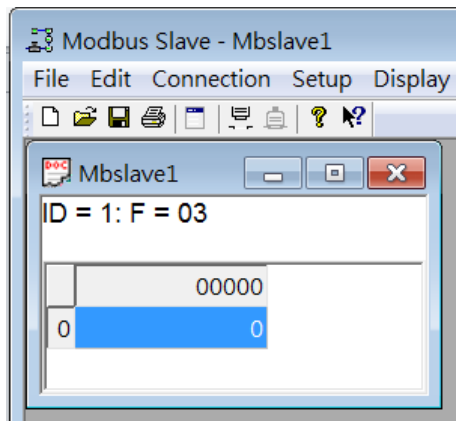
Set as below:

### Serial Settings

Port	Baud rate	Parity	Data bit	Stop bit	Flow control	FIFO	Interface	RTS on delay	RTS off delay
1	115200	Even	8	1	None	Enable	RS-232	0	0

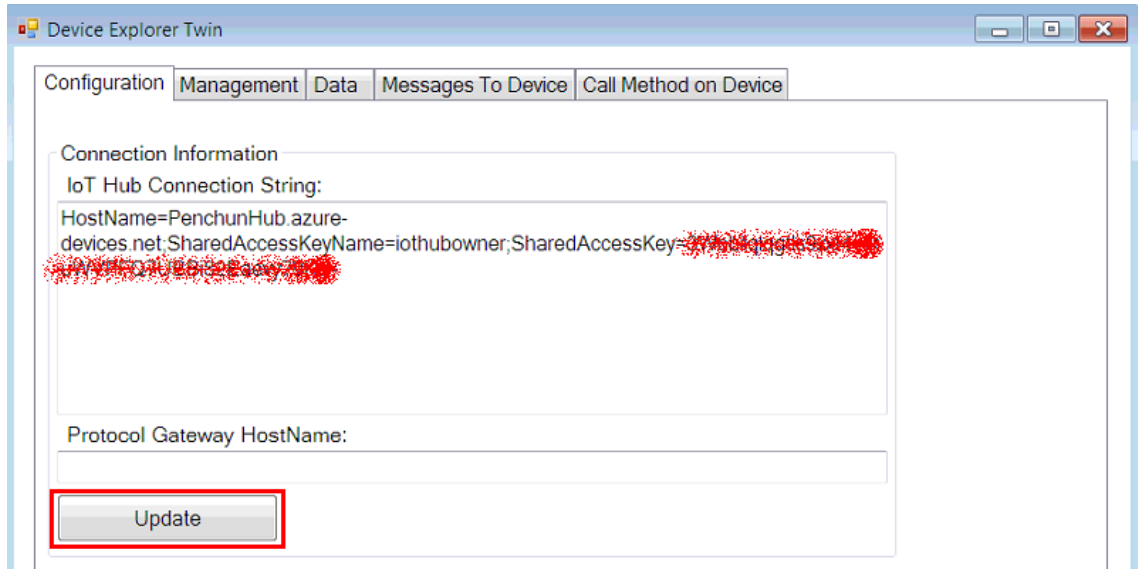
## 5. Modbus Slave Tool Settings

PC1 runs **Modbus Slave tool** and connects to MGate 5105's Serial Port. Add the Modbus definition below:

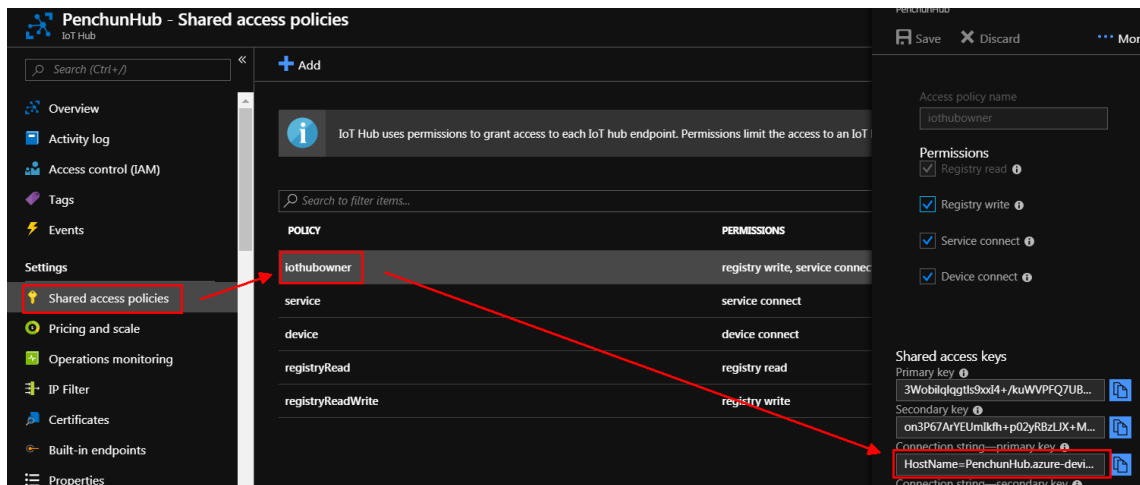


## 6. Device Explorer Settings

PC2 runs Device Explorer. In Configuration Tab, fill in IoT Hub Connection String.

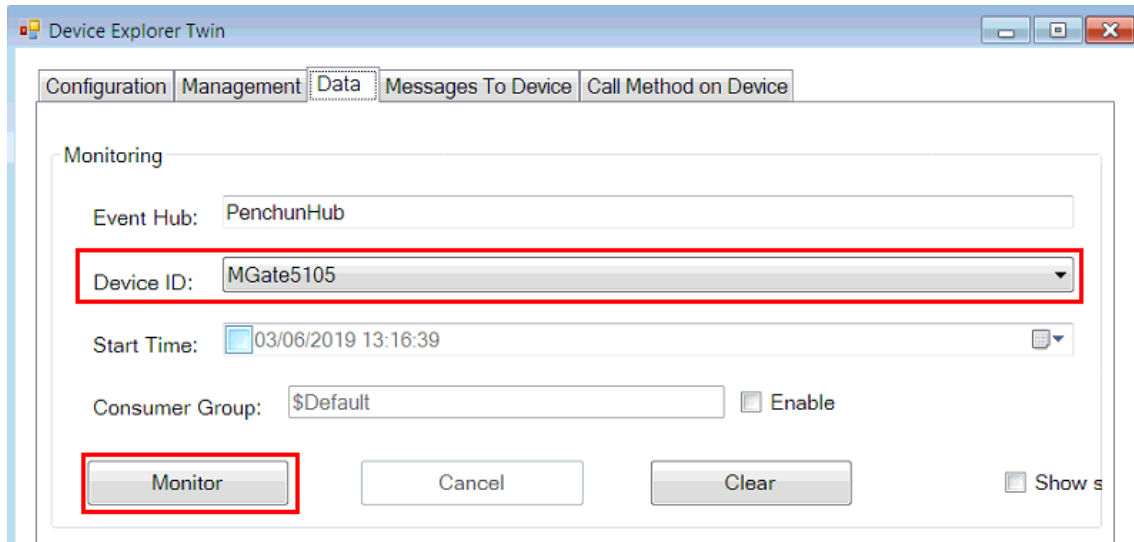


This string can be found under "IoT Hub Shared access policies" in Azure Portal as below:



After filling in the IoT Hub Connection String, click "Update" and it will connect to Azure IoT Hub.

In Data Tab, choose your IoT Device from Device ID dropdown list. Then click "Monitor" to monitor Device-to-cloud message.



## 7. Communication Test

### 7.1 Device-to-cloud message

Set Device-to-cloud message Trigger as "no cyclic sending" and "by specific tag change":

**Trigger**

**Cyclic sending intervals**  (1000 - 86400000 ms, 0 for disable)

**Tag changes**

To enable Onchange trigger with Trigger deadband to 0 on TagR1.

**Protocol Tag**

**Name**

**Data unit**

**Unit quantity**

**Endian swap**

**Onchange trigger**

**Trigger deadband**

So when the MGate 5105 gets Modbus RTU device Register0's value changed, it triggers to publish message to IoT Hub.

Now, you can try to update Modbus Register0's value to 1. In Device Explorer's Event Hub Data, it gets TagR1's value shown as 1 and with dateTime padding.

```
Event Hub Data
Receiving events...
2019-03-26 4:06:24 PM> Device: [MGate5105], Data:[{"msgID":"Read1","msgVer":"1.0","gwID":"MGateAzure","ModuleR1":
{"TagR1":1},"dateTime":"2019-03-26T16:04:13+08:00"}]
```

### 7.2 Cloud-to-device message

Device Explorer can be used to send messages to device. You can follow below steps:

1. Click Cloud-to-device message "View JSON" button.

Pair Settings

+ Add   Edit   Clone   Delete

Type	Name
Message ID	msgID
Message Version	msgVer
Gateway ID	gwID
- Module	ModuleW1
Protocol Tag	TagW1

View JSON   Ok   Cancel

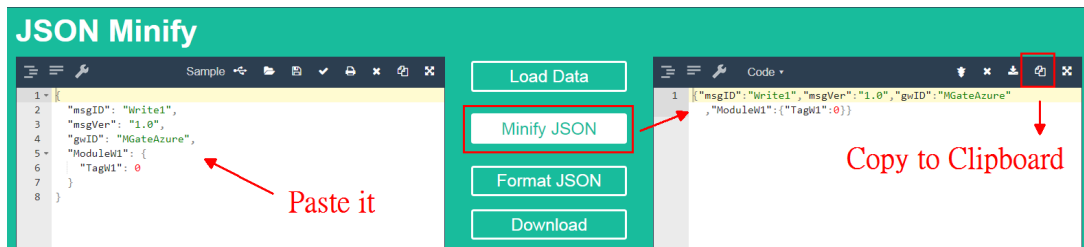
Copy Cloud-to-device message JSON format:



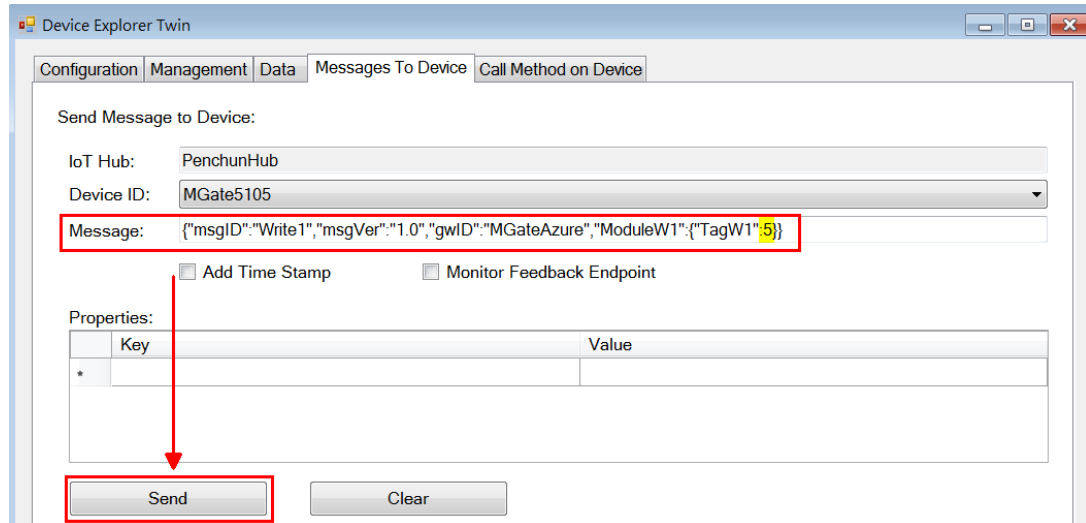
- 2. The copied message has a lot of space and line feed. We can use tool to compact it. Below is a free online tool:

<https://jsonformatter.org/json-minify>

Paste the message on the left side column, then click "Minify JSON". It will show a compact JSON format message on the right side column. Click "Copy to Clipboard".



3. Paste it on the Message field of Device Explorer, and modify TagW1 value to 5, then click "Send".



4. Check Modbus Slave tool that Register0's value is updated to 5.

