How to Build an IoT Application with Moxa's ThingsPro and AWS IoT Service

Moxa Technical Support Team <u>support@moxa.com</u>

Contents

1	Introduction	2
2	Application Scenario	2
3	Prerequisites	3
4	Solution	3
5	How to Get the AWS IoT Parameters for ThingsPro	10
6	Additional Reading	19

Copyright © 2018 Moxa Inc.

Released on February 02, 2018

About Moxa

Moxa is a leading provider of edge connectivity, industrial computing, 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.

How to Contact Moxa

Tel: 1-714-528-6777 Fax: 1-714-528-6778



Using Moxa ThingsPro and AWS IoT Service

1 Introduction

Moxa's ThingsPro[™] 2.0 Suite simplifies the development of IoT applications and facilitates data acquisition as well as remote device management. ThingsPro provides Modbus communications, data acquisition, wireless networking, and device management, in a few simple steps, allowing users to focus on developing applications instead of complex system integration.



For users who develop data acquisition and asset management software programs on their own, ThingsPro provides the ability to transfer field data to the AWS IoT service without requiring any additional programming on the gateway. ThingsPro 2.0 includes generic Modbus protocol and AWS IoT support, which you can use to easily configure protocol data polling tables and AWS IoT connection settings. You can then upload the data collected to the AWS IoT service. In this article, we describe how to configure the MODBUS device template in ThingsPro, retrieve data from ThingsPro, and upload the data to a remote HTTP server.

2 Application Scenario

Set up an environment to send real-time device data updates to the AWS IoT service using ThingsPro 2.0. Here, ThingsPro is used as a Modbus data logger and the AWS function provided by ThingsPro is used to configure the connection to the AWS IoT service.

3 Prerequisites

1. Moxa UC-8112 gateway computer with Internet access, and ThingsPro 2.0 Gateway and SD card installed.

For information about setting up a Moxa UC-8112, refer to the *ThingsPro User's Manual*.

- 2. PC/notebook with Chrome browser and Microsoft Device Explorer installed
- 3. Modbus device (e.g., Moxa ioLogik-E2242)
- 4. <u>AWS IoT configuration: thing name, CA certificate, device certificate, device private key,</u> <u>mqtt topics, and mqtt endpoint</u>

4 Solution

Complete the following steps to create your IoT application using ThingsPro and AWS IoT service:

1. Choose an equipment template.

a. In the ThingsPro Gateway web page, click \equiv (Menu).

						ADMIN •	• ●	٠
	Dia topy 10 v	(877 CPU (%)					
Device I	nformation	C	Maintenance					
	Hostname	Moxa 🧨	Upgrade system	P				
1.0	Device name	ThingsPro 🧨	Export system config	۵				
р	Software version	2.0 Build 17031708	Import system config	Ð				
O	Uptime	in 29 minutes	Reboot system	Ģ				
٥	Memory size	246 MB	Export syslog	6				
Online								

Copyright $\ensuremath{\mathbb{C}}$ 2018 Moxa Inc.

Using Moxa ThingsPro and AWS IoT Service

b. In the ThingsPro Gateway menu, click on the **Modbus & Logging** link.

A admi admin@	n pmoxa.com
Menu	
	Home
\$	Gateway 👻
	User Account
<>	User Programs
0	MODBUS & Logging
‡‡‡	Modbus Slave
	Applications 👻

• Online

c. Open the EQUIPMENT TEMPLATE tab

d. Browse through the **Template List** to locate the template corresponding to the Modbus device or create a new template.

For details on creating a new template, refer to the ThingsPro User's Manual.

EQUIPMENT TEMPLATE		MODBUS DEVICE LOG UPLOAD			
Template Management				C	
Template List				± =+	
Name 🔨	Tag Count	Template Action		Tag Action	
ioLogik-E1210	16	Ē		•	
ioLogik-E1211	31		/ =+	•	
ioLogik-E1212	32		∕ =+	۵ ک	
ioLogik-E1213	28		∕ =+	6	
ioLogik-E1214	18		. ₹+	6	
ioLogik-E1240	8	6	/ =+	D	
ioLogik-E1241	8		/ =+	© •	
ioLogik-E1242	20		/ ≡+	Ē 🕨	
ioLogik-E1260	6	6	▶ =+	ē 🕨	

In this article, we use the Modbus device **E2242**, which is already listed in the **Template** List.

Using Moxa ThingsPro and AWS IoT Service

- 2. Create an instance of the Modbus device.
 - a. Select the **MODBUS DEVICE** tab.
 - b. Configure the Modbus device with a communication interface (serial or Ethernet).
 Because the connected device (E2242) uses Modbus TCP protocol, click on the
 MODBUS/TCP tab.

								A	ADMIN 👻
EQUIPMEN	T TEMPLATE		MODBUS DEVICE]			LOG UP	LOAD	
Modbus Management		*							C
MODBUS / RTU	MODBUS / TCP	ĸ							
TCP List									=+
Name 个	Interval	Port	Host IP						
				Page:	1 •	1 - 2 of 2	К	< >	×

- i. Click =+ (Add TCP interface).
- ii. Fill in the TCP interface settings and click **SAVE**.

Interface Name * E2242	
EZ242	
Host IP *	
192.168.4.123	
Port *	
502	
Interval *	
1000	
Response Timeout *	
5000	\$

The Modbus/TCP interface for E2242 is added to the **TCP List**.

Using Moxa ThingsPro and AWS IoT Service

iii. Click on the (Add connected device) button corresponding to E2242 to add a connected device.

EQUIPMEN	IT TEMPLATE		MODBUS DEVICE		LOG U	JPLOAD
Modbus Management						G
MODBUS / RTU	MODBUS / TCP					
TCP List						≡+
Name 🛧	Interval	Port	Host IP			
E2242	1 sec	502	192.168.4.123		0	/ =+

- iv. Fill in the **Device Name**.
- v. Select a **Template** to combine MODBUS/TCP interface with the Modbus device.
- vi. Specify the **Unit ID** for the device and click **SAVE**.

Device Name + E2242	
Template + ioLogik-E2242	
Unit ID +	

ThingsPro will now start continuously polling data from the device.

To check the status, click <a>(Show connected devices).

If the **Request Status** is green, ThingsPro is able to send requests to the Modbus TCP device and receive data.

Using Moxa ThingsPro and AWS IoT Service

- 3. Use the AWS IoT service for real-time data update.
 - a. In the ThingsPro Gateway menu, click on the **AWS IoT** link in the **Applications** section.



Copyright $\ensuremath{\mathbb{C}}$ 2018 Moxa Inc.

b. Fill in the AWS IoT service details.

Follow the steps in "How to get AWS IoT parameters for ThingsPro" to get the AWS IoT service parameters.

WS IoT	(
Enable	
Connection Status	č
Target Host *	
This field is required	
Port* 8883	
Topic *	
This field is required	
Client ID *	
This field is required	
My Thing Name *	
This field is required	
Root CA File E.g.: ****-G5.pem	SELECT
Root GA File E.g. ****453.pem	
Certificate File E.g.: ****-certificate.pem.crt	SELECT
certificate Prie E.g.,certificate.pem.on	
Private Key File E.g.: ****-private.pem.key	SELECT
SELECT TAGS	
Logging data when network is disconnected	
	SAVE
	SAVE

- c. Click **SELECT TAGS** to select the tags that you want to upload to AWS IoT.
- d. Click **SAVE**.

Using Moxa ThingsPro and AWS IoT Service

Edit u	pload tag	setting			×
Select e d 2242	*				
2 it	ems are s	elected			
		Log On Change		Unit	Data Type
	Name 个		Description		
	AI_0				int16
	AI_1				int16
	AI_2				int16
	DI_0				boolean
	DI_1				boolean
	DI_2				boolean
		Page:	1 🔻 1-6 o	f6 K ≺	> >
					SAVE

e. If you need to log data when network is down, check the Logging data when network is disconnected, select the log File destination, and specify the Max. Storage for Log value.

Your data will be logged in a data file. Once the network connection comes up, the data in the log file will be transmitted to AWS.

Logging data when network is disconnected	
File destination	
Internal(/var/mxc)	*
Max. Storage for Log (MB) *	
2000	
Vax. Days for Log	3.2 days
	SAVE

- f. Select the **Enable** option
- g. Click SAVE

The AWS IoT interface will use the information provided to create a connection with the AWS IoT service. Once the connection is established, data can be uploaded to the AWS IoT service in real time.

Once the AWS Client App successfully connects to the AWS service, the connection status turns green as shown below:



5 How to Get the AWS IoT Parameters for ThingsPro

To get the AWS IoT parameter for ThingsPro, you must log in into the AWS IoT service and do the following:

1. Click on the **AWS IoT** link.

🎁 Services 🗸 Resource Group	os v 1 k					
Fin	/S services d a service by name (for example, EC Recently visited services	2, S3, I	Elastic Beanstalk).			۹
Ŵ	AWS IoT Il services	١	EC2	Ŷ	IAM	
¢	Compute EC2 EC2 Container Service	60	Developer Tools CodeCommit CodeBuild	Ŵ	Internet of Things AWS IoT	
k	Lightsail Elastic Beanstalk Lambda		CodeDeploy CodePipeline	2	Game Development Amazon GameLift	
	Storage S3 EFS		Management Tools CloudWatch CloudFormation CloudTrail Config		Mobile Services Mobile Hub Cognito Device Farm Mobilo Apalvtics	

Copyright $\ensuremath{\mathbb{C}}$ 2018 Moxa Inc.

You will see the AWS IoT Dashboard.



2. Click on the **connect** link and then the **Get started** button.

🎁 Services 🗸 Resource	Groups 🤟 🛠		û andy.chu @ moxasys ∗ Oregon •	Support +
Welcome to the new AWS IoT Console.	Take me to the old console Dismiss			
AWS IOT		Connect to AWS IoT		Û
Image: Connect Image: Conne Image: C				
	Configuring a device	AWS IoT Button	AWS IoT Starter Kit	
	Connect a device or your computer to AWS IoT using the connection wizard for AWS IoT Device SDKs.	The AWS IoT Button is a single-purpose device that sends a message to AWS IoT with a press of a button.	Browse AWS IoT Starter Kits that were made for connecting to AWS IoT and getting started with the service.	
Settings	Get started	Configure a button Don't have a button? Buy one	Browse starter kits	

3. Choose the Linux/OSX platform and select the Python SDK.

🎁 Services 🗸 Resourc	ce Groups 🗸 🔥		🗘 🛛 andy.chu @ moxasy	s 🕶 Oregon 👻 S	upport ~
Welcome to the new AWS IoT Consol	e. Take me to the old console Dismiss				
_					ņ
÷	How are you connecting to AWS IoT?				-
	Select the platform and SDK that best suits how you are connecting Choose a platform	to AWS IoT.			
	Linux/OSX	Windows	>		
	Choose a AWS IoT Device SDK Connect in 15 minutes or less with these SDKs and a quick setup scr	ipt.			
	Node.js	Python	>		

4. Click Get started

For a services → Resource Groups ★ Welcome to the new AWS IoT Console. Take me to the or	tid console Dismiss	⊉ andy.chu @ moxas+ Oregon - Support -
×	Connect to AWS IoT	Ф
-	Connecting to AWS IoT requires the creation and configuration of several components with the service. The following steps will help do this for you. You will:	
	Register a device A thing is the representation and record of your physical device in the cloud. Any physical device needs a thing record in order to work with AWS IoT.	
	Download a connection kit The connection kit includes some important components: security credentials, the SDK you chose, and a sample project. You can work with these individually.	
	Configure and test your device Using the connection kit, you will configure your device by transferring files and running a script, and test that it is connected to AWS IoT correctly.	
	Some prerequisites to consider, the device should have Python and Git installed and a TCP connection to the public internet on port 8883.	
	Want to learn more about the components of AWS IoT? Try the interactive overview Get started	

- Fill in the thing name. e.g.:8112.
 This name will be used in the Client ID and My Thing name in the ThingsPro AWS IoT connection UI.
- 6. Click **Next step**.

CONNECT TO AWS IOT Register a thing STEP 1/3	
A thing is the representation and record of your physical device in the cloud. Any physical device needs a thing to work with AWS IoT. Creating a thing will also create a thing shadow.	Choose an existing thing instead?
Name 8112	
Show optional configuration (this can be done later) -	
Cancel	Back Next step

- 7. Click on the Download connection kit link
- 8. After the download has finished, click **Next step**.

CONNECT TO AWS IOT Download a connection kit STEP 2/3	
The following AWS IoT resources	will be created:
A thing in the AWS IoT registry	8112
A policy to send and receive messages	8112-Policy Preview police
The connection kit contains: A certificate and private key	8112.cert.pem, 8112.private.key
AWS IOT Device SDK	Python SDK
A script to send and receive messages	start.sh
Before your device can connect ar	nd publish messages, you will need to download the connection kit.
Download connection kit for Linux/OSX	
Cancel	Back Next step

9. Click Done.

Note: You do not need to perform steps 1 to 3 under "Configure and test your device".

CONNECT TO AWS Configure and STEP 3/3	IOT test your device		
To configure and t	est the device, perform the follow	wing steps.	
	connection kit on the device vice_package.zip		
Step 2: Add exect			
	art script. Messages from you	r thing will appear below	
Waiting for mess	ages from your device		
Cancel			Back Done
ight © 2018 Moxa I	nc.		Page 14 of

Using Moxa ThingsPro and AWS IoT Service

- 10. Extract the files from the **connect_device_package.zip** file that you just downloaded. The files include information on the private key, certificate file, and public key.
 - 📄 8112.cert.pem 🔊 8112.private.key
 - 🗿 8112.public.key
 - 📄 start.sh
- 11. Download the AWS root-CA file from following URL: <u>https://www.symantec.com/content/en/us/enterprise/verisign/roots/VeriSign-Class%203-</u> <u>Public-Primary-Certification-Authority-G5.pem</u>
- 12. Copy the content and save it in a root-CA.crt file together with the private key and certificate file. The root-CA.crt , private key, and certificate file will be used in the ThingsPro AWS IoT connection UI



 Go back to the AWS IoT dashboard and click on the **Registry and Things** link. You will see the "thing" that you just created.

Using Moxa ThingsPro and AWS IoT Service



- 14. Click on the thing that you created.
- 15. To fill in the AWS IoT parameters, do the following:
 - Click on the **Interact** link

Using Moxa ThingsPro and AWS IoT Service

THING 8112 NO TYPE		Actions -
Details	Thing ARN	
Security	A thing Amazon Resource Name uniquely identifies this thing.	
Shadow	556337455073:thing/8112	
Activity	Туре	Edit
	Q No type	
	0 Attributes	Edit

• The **Rest API Endpoint** is the **target host** of the ThingsPro AWS IoT connection UI.

3112		
IO TYPE	Acti	ions -
Details	This thing already appears to be connected. Connect a d	evice
Security		
Shadow	HTTPS	
Interact	Update your Thing Shadow using this Rest API Endpoint. Learn more	
Activity	el. 2mi intus-west-2. amazonaws.com	
	MQTT Use topics to enable applications and things to get, update, or delete the state information for a Thing	1
	Use topics to enable applications and things to get, update, or delete the state information for a Thing (Thing Shadow) Learn more	
	Update to this thing shadow	
	\$aws/things/8112/shadow/update	
	Update to this thing shadow was accepted	
	\$aws/things/8112/shadow/update/accepted	
	Update this thing shadow documents	

Using Moxa ThingsPro and AWS IoT Service

• The **update to this thing shadow** is the **Topic** of ThingsPro AWS IoT connection UI.

Details	This thing already appears to be connected. Connect a dev
Security	HTTPS
Shadow	
Interact	Update your Thing Shadow using this Rest API Endpoint. Learn more
Activity	adw9mfri5w1h2.iot.us-west-2.amazonaws.com
	MQTT
	MQTT Use topics to enable applications and things to get, update, or delete the state information for a Thing (Thing Shadow) Learn more
	Use topics to enable applications and things to get, update, or delete the state information for a Thing (Thing Shadow) Learn more Update to this thing shadow
	Use topics to enable applications and things to get, update, or delete the state information for a Thing (Thing Shadow) Learn more
	Use topics to enable applications and things to get, update, or delete the state information for a Thing (Thing Shadow) Learn more Update to this thing shadow

With above information, you should be able to fill up all required information for connecting to AWS IoT service.

Enable	
Update on chagne	
'arget host " xxxxxxxx.iot.us-west-2.amazonaws.com	
^{fort *}	
^{iopic *} Saws/things/test/shadow/update	
Client ID *	
Root CA File ex: ****-G5.pem	SELECT
Root CA File ex: ****-G5.pem	SELECT
Root CA File ex: ****-G5.pem root-CA.crt Certificate File ex: ****-certificate.pem.crt	SELECT
My Thing Name * test Root CA File ex: ****-G5.pem root-CA.crt Certificate File ex: ****-certificate.pem.crt certificate.pem.crt Private Key File ex: ****-private.pem.key private.pem.key	

6 Additional Reading

Getting Started with AWS IoT