

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

Moxa Technical Support Team
support@moxa.com

Contents

- 1 Introduction..... 2**
- 2 Application Scenario 2**
- 3 Prerequisites..... 3**
- 4 Solution 3**
- 5 Additional Reading 8**

Copyright © 2017 Moxa Inc.

Released on March 31, 2017

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.

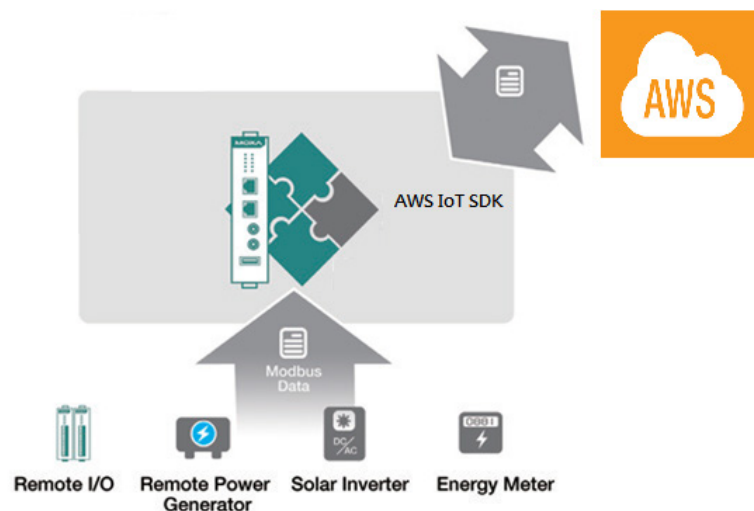
How to Contact Moxa

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



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

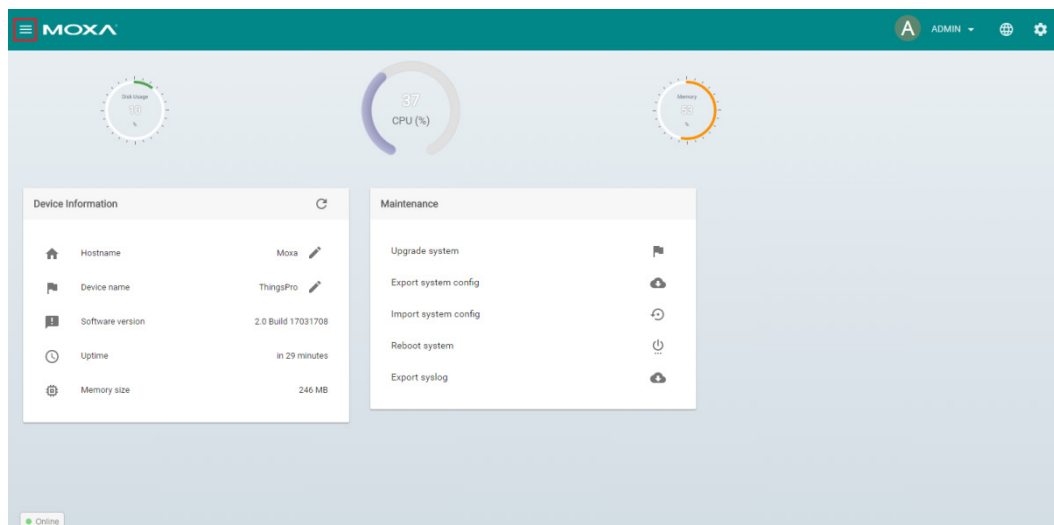
1. Moxa UC-8112 gateway computer with Internet access, and ThingsPro 2.0 Gateway and SD card installed.
For information about setting up a Moxa UC-8112, refer to the *ThingsPro User's Manual*.
2. PC/notebook with Chrome browser and [Microsoft Device Explorer](#) installed
3. Modbus device (e.g., Moxa ioLogik-E2242)
4. [AWS IoT configuration: thing name, CA certificate, device certificate, device private key, mqtt topics, and mqtt endpoint](#)

4 Solution

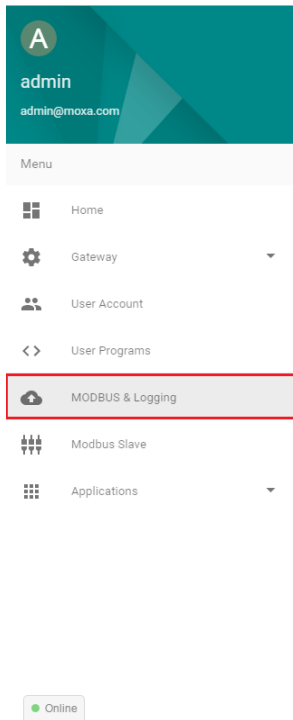
Complete the following steps to create your IoT application using ThingsPro and AWS IoT service:

1. Choose an equipment template.

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

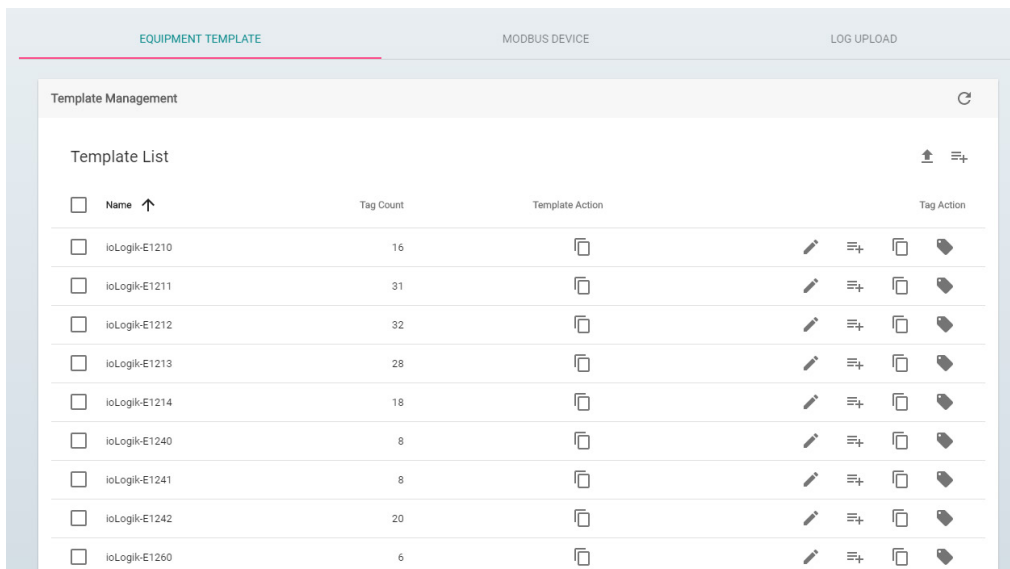


- b. In the ThingsPro Gateway menu, click on the **Modbus & Logging** link.



- 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.

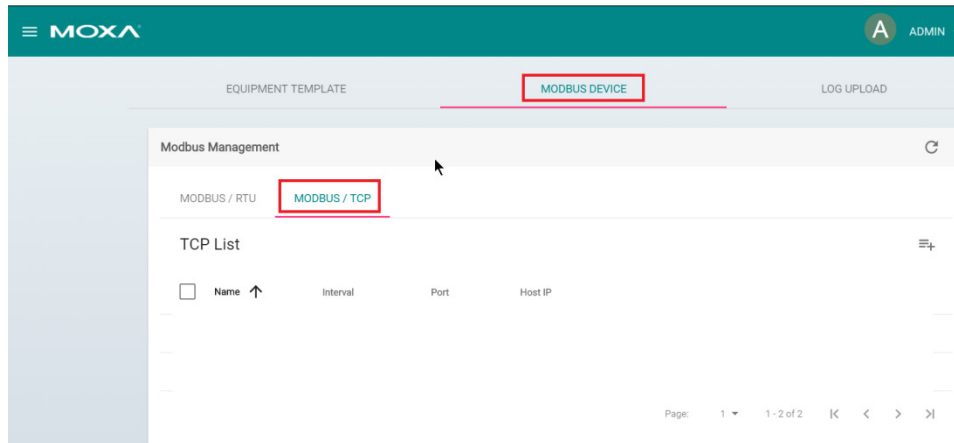
For details on creating a new template, refer to the *ThingsPro User's Manual*.




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

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.



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

Edit TCP Interface Settings ×

Interface Name *
E2242

Host IP *
192.168.4.123


Port *
502

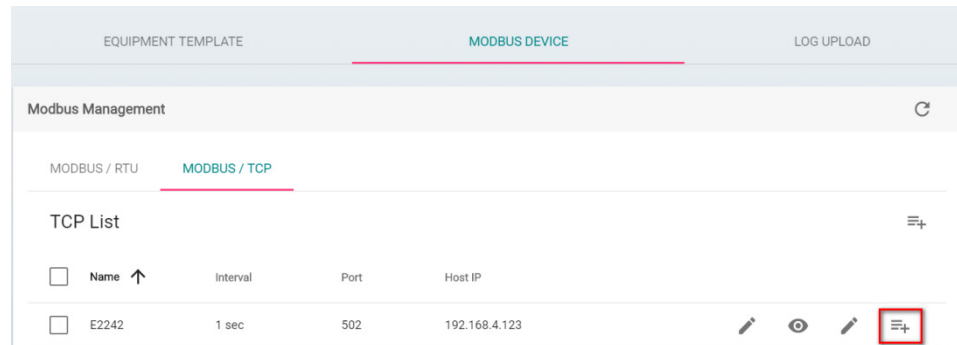
Interval *
1000

Response Timeout *
5000

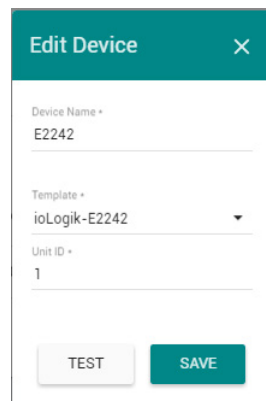
SAVE

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

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



- 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**.



Edit Device [X]

Device Name *
E2242

Template *
ioLogik-E2242

Unit ID *
1

TEST SAVE

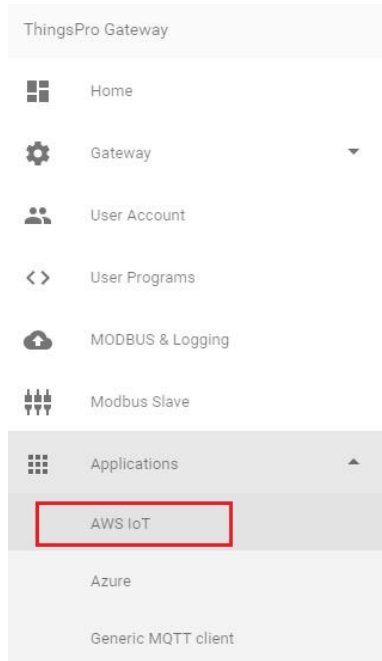
ThingsPro will now start continuously polling data from the device.

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

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

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.



b. Fill in the AWS IoT service details.

AWS IoT

Enable

Update on change

Target Host *

Port*
8883

Topic *

Client ID *

My Thing Name *

Root CA File E.g.: ****-G5.pem

Certificate File E.g.: ****-certificate.pem...

Private Key File E.g.: ****-private.pem.key

c. Select the **Enable** option

d. 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.

5 Additional Reading

[Getting Started with AWS IoT](#)