Think Automation and beyond...

Connecting FC6A Plus to AWS IoT Core

Don Pham

August 18, 2021

Revision 1.01

Overview

- This document is a tutorial for explaining how to connect FC6A Plus to AWS IoT Core.

- What is IDEC FC6A Plus?
  - FC6A Plus is a Micro PLC with IoT features such as Web Server and AWS Cloud Connectivity via MQTT.

- Product Family
  - IDEC also have FC6A All-in-One as FC6A family. This series provides basic PLC features.(FC6A Plus has additional IoT features.)

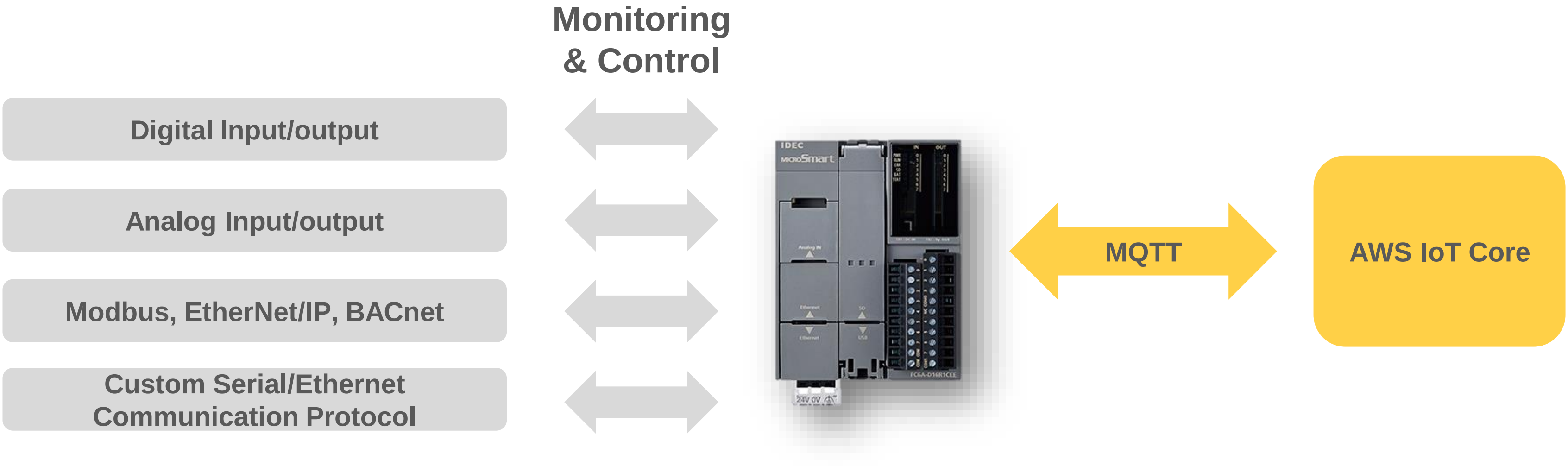
FC6A Plus

FC6A All-in-One

Overview

FC6A Plus Features

- The FC6A Plus provides basic PLC function to control Machine and Equipment. In addition, it has powerful communication features.
- It supports the MQTT protocol to connect your machine/equipment to AWS IoT Core, and support other communication protocols such as Modbus, EtherNet/IP, BACnet.



Monitoring & Control

Digital Input/output

Analog Input/output

Modbus, EtherNet/IP, BACnet

Custom Serial/Ethernet Communication Protocol

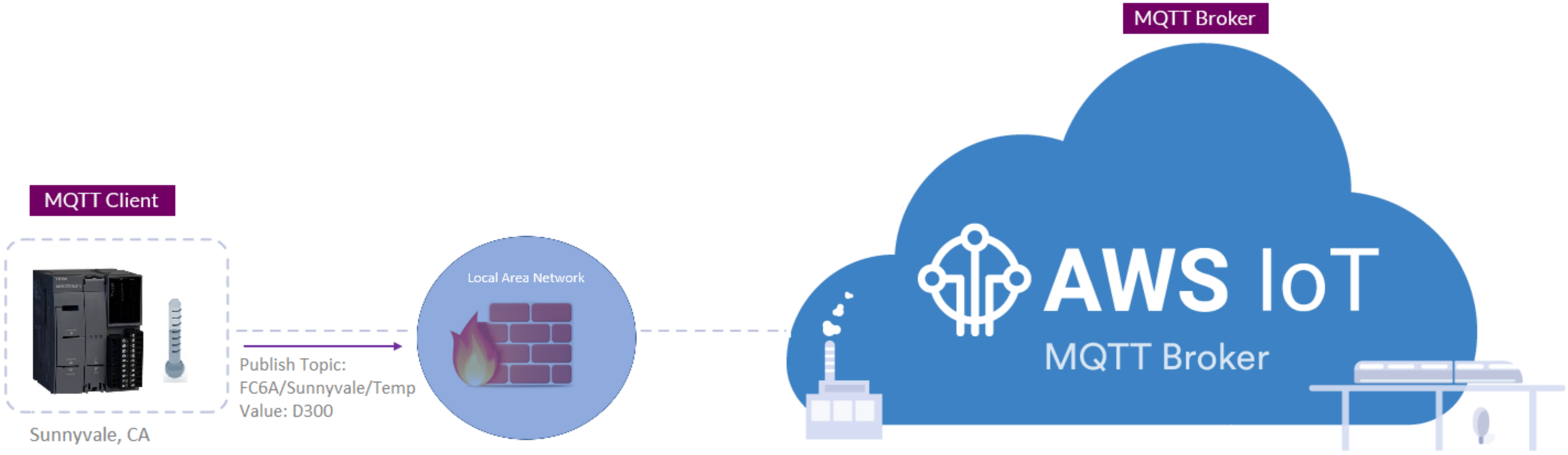
MQTT

AWS IoT Core

Example 1 - Publish

- FC6A Plus CPU is configured as Publisher

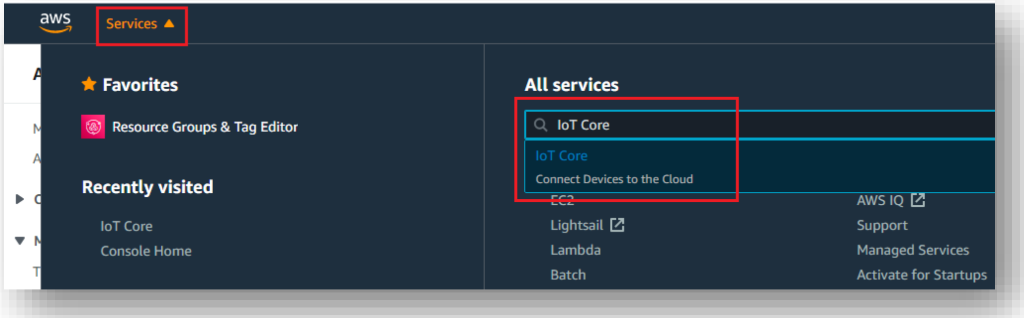
- AWS IoT Core console being used as Subscriber to verify the communication



AWS IoT Core configuration

AWS IoT Core configuration

1. Login AWS account
2. Click Services, search *IoT Core*, and select



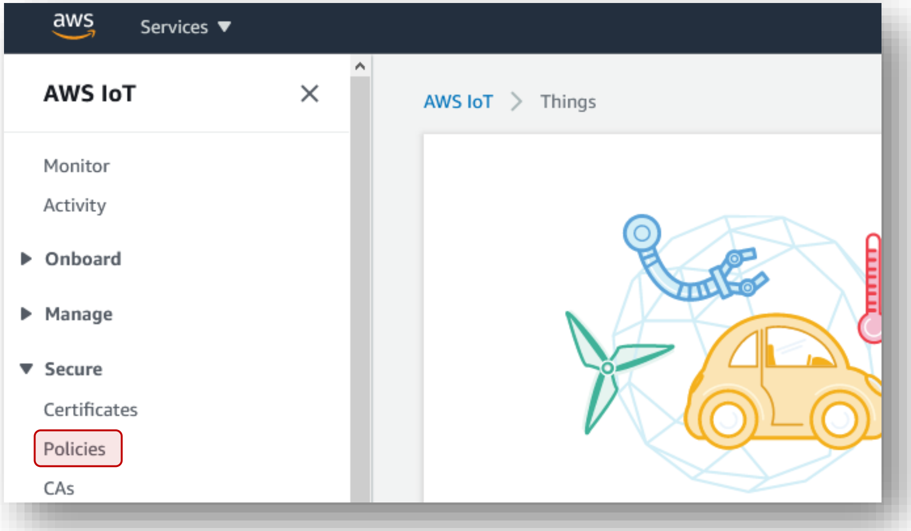
3. In AWS IoT Core, we'll configure 3 things
   A. Create "**Policies**"
   B. Define "**Things**" and Create "**Certificates**"
   C. Confirm Endpoint (IP Address/Host Name)

Reference
If you don't have AWS account, refer to the instructions at
https://docs.aws.amazon.com/iot/latest/developerguide/setting-up.html

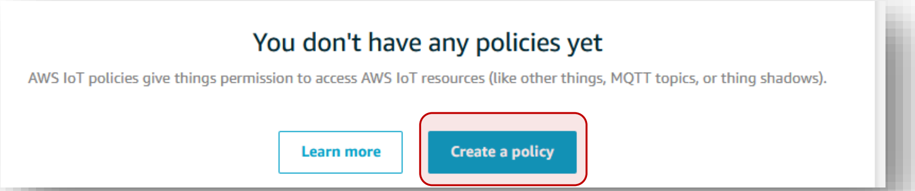
AWS IoT Core configuration

Create Policies

4. Under Secure, select *Policies*



5. Click *Create a Policy*



You don't have any policies yet

AWS IoT policies give things permission to access AWS IoT resources (like other things, MQTT topics, or thing shadows).

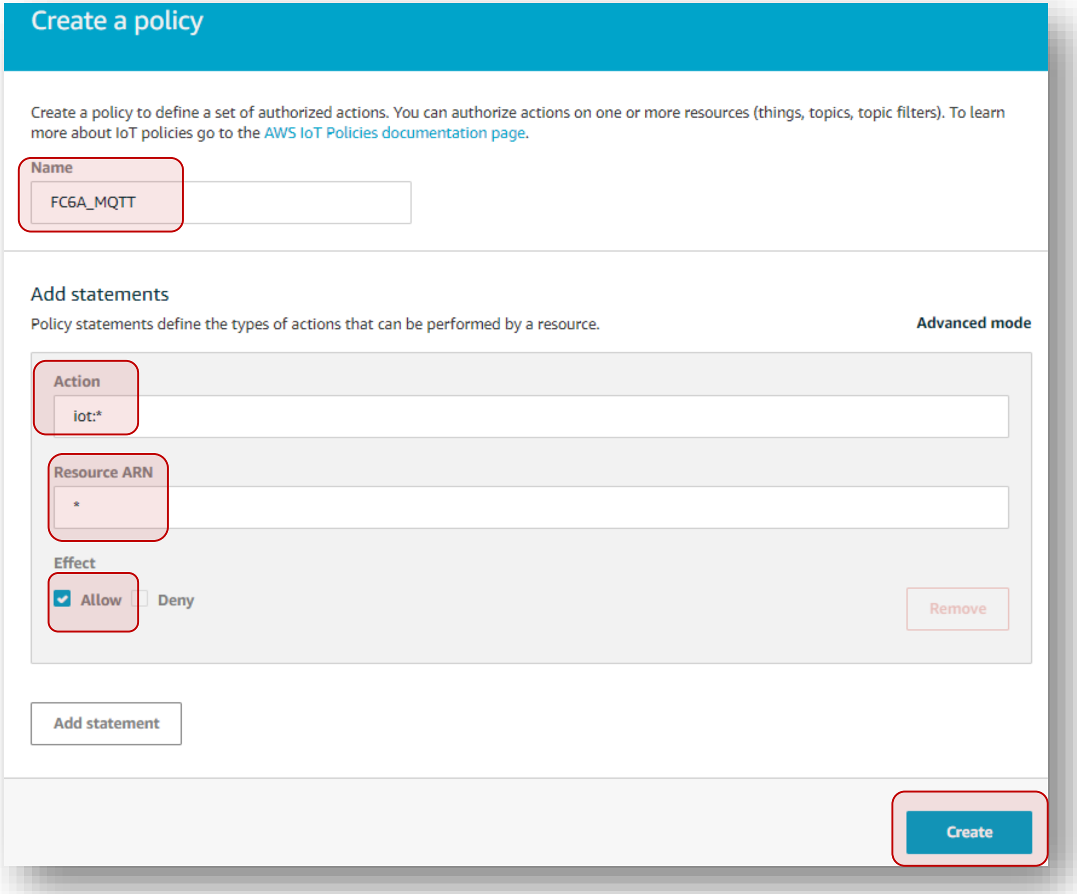
AWS IoT Core configuration

Create Policies

6. Configure the following parameters as followed:
   - Name: any name (FC6A_MQTT)
   - Action: **iot:***
   - Resource ARN: *
   - Effect: Check ***Allow***

7. Click Create

__NOTE__ – The examples in this document are intended only for dev environments. All devices in your fleet must have credentials with privileges that authorize only intended actions on specific resources. The specific permission policies can vary for your use case. Identify the permission policies that best meet your business and security requirements. For more information, refer to [Example policies](#) and [Security Best practices](#).



Create a policy

Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters). To learn more about IoT policies go to the AWS IoT Policies documentation page.

Name: FC6A_MQTT

Add statements

Policy statements define the types of actions that can be performed by a resource.

Action: iot:*

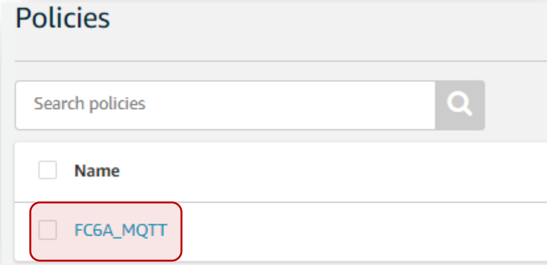
Resource ARN: *

Effect: ☑ Allow ☐ Deny

AWS IoT Core configuration

Create Policies

8. Once policy is created you can see and check by clicking on the policies names

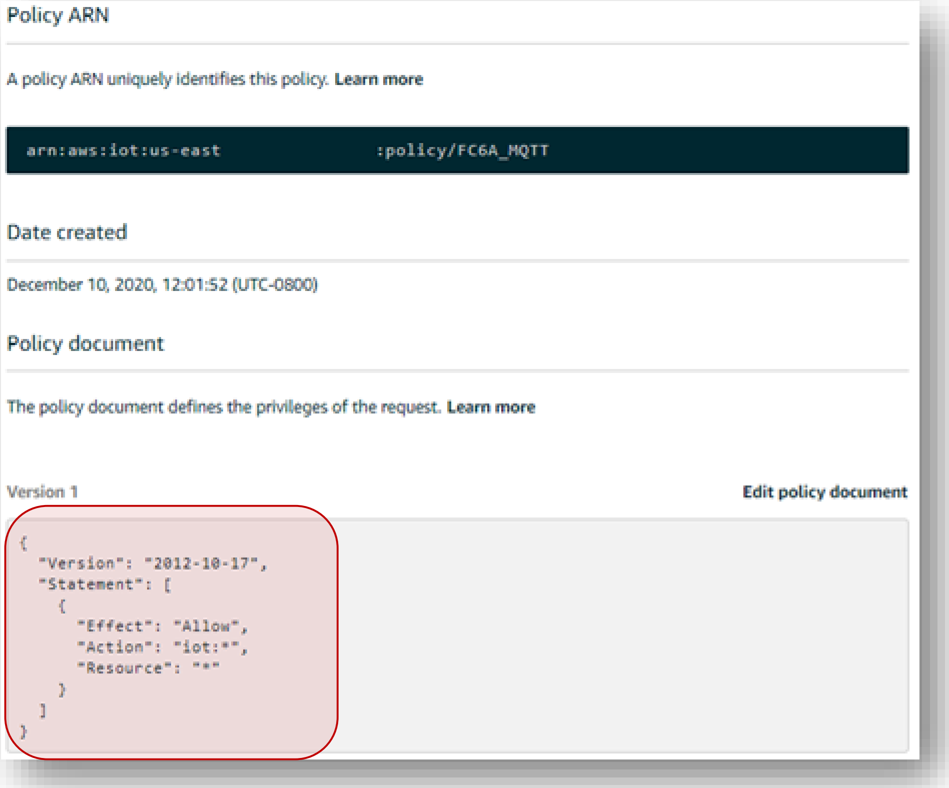


Policies

Search policies

□ **Name**

□ FC6A_MQTT



Policy ARN

A policy ARN uniquely identifies this policy. **Learn more**

```
arn:aws:iot:us-east[redacted]:policy/FC6A_MQTT
```

Date created

December 10, 2020, 12:01:52 (UTC-0800)

Policy document

The policy document defines the privileges of the request. **Learn more**

Version 1

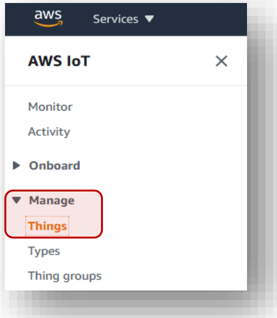
Edit policy document

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:*",
      "Resource": "*"
    }
  ]
}
```

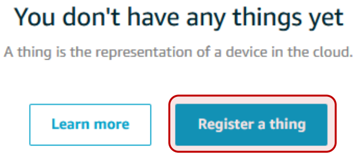
AWS IoT Core configuration

Define Things

9. Under Manage, click *Things*



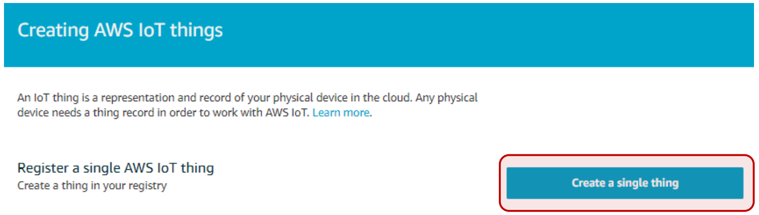
10. Click *Register a Thing*



You don't have any things yet

A thing is the representation of a device in the cloud.

11. Click *Create a single thing*



Creating AWS IoT things

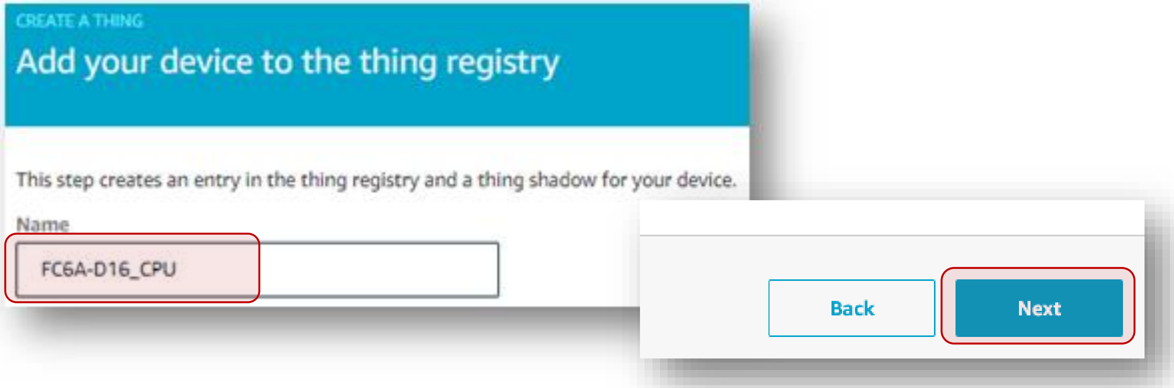
An IoT thing is a representation and record of your physical device in the cloud. Any physical device needs a thing record in order to work with AWS IoT. Learn more.

Register a single AWS IoT thing
Create a thing in your registry

AWS IoT Core configuration

Define Things

12. Give it a name and click *Next*



CREATE A THING

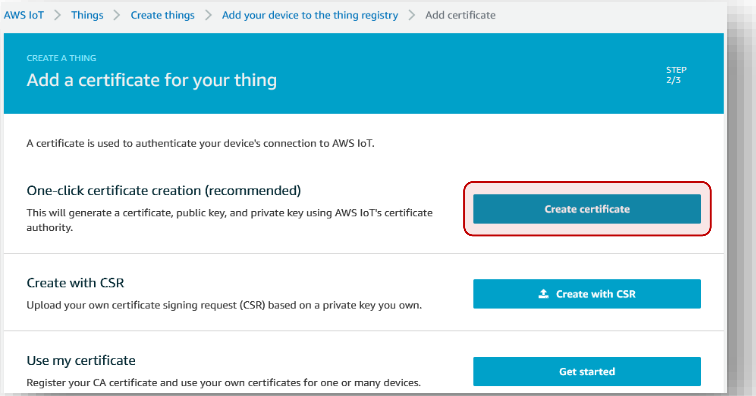
Add your device to the thing registry

This step creates an entry in the thing registry and a thing shadow for your device.

Name

FC6A-D16_CPU

13. Under One-click certificate creation, click *Create Certificate*



AWS IoT > Things > Create things > Add your device to the thing registry > Add certificate

CREATE A THING

Add a certificate for your thing

STEP 2/3

A certificate is used to authenticate your device's connection to AWS IoT.

One-click certificate creation (recommended)

This will generate a certificate, public key, and private key using AWS IoT's certificate authority.

Create with CSR

Upload your own certificate signing request (CSR) based on a private key you own.

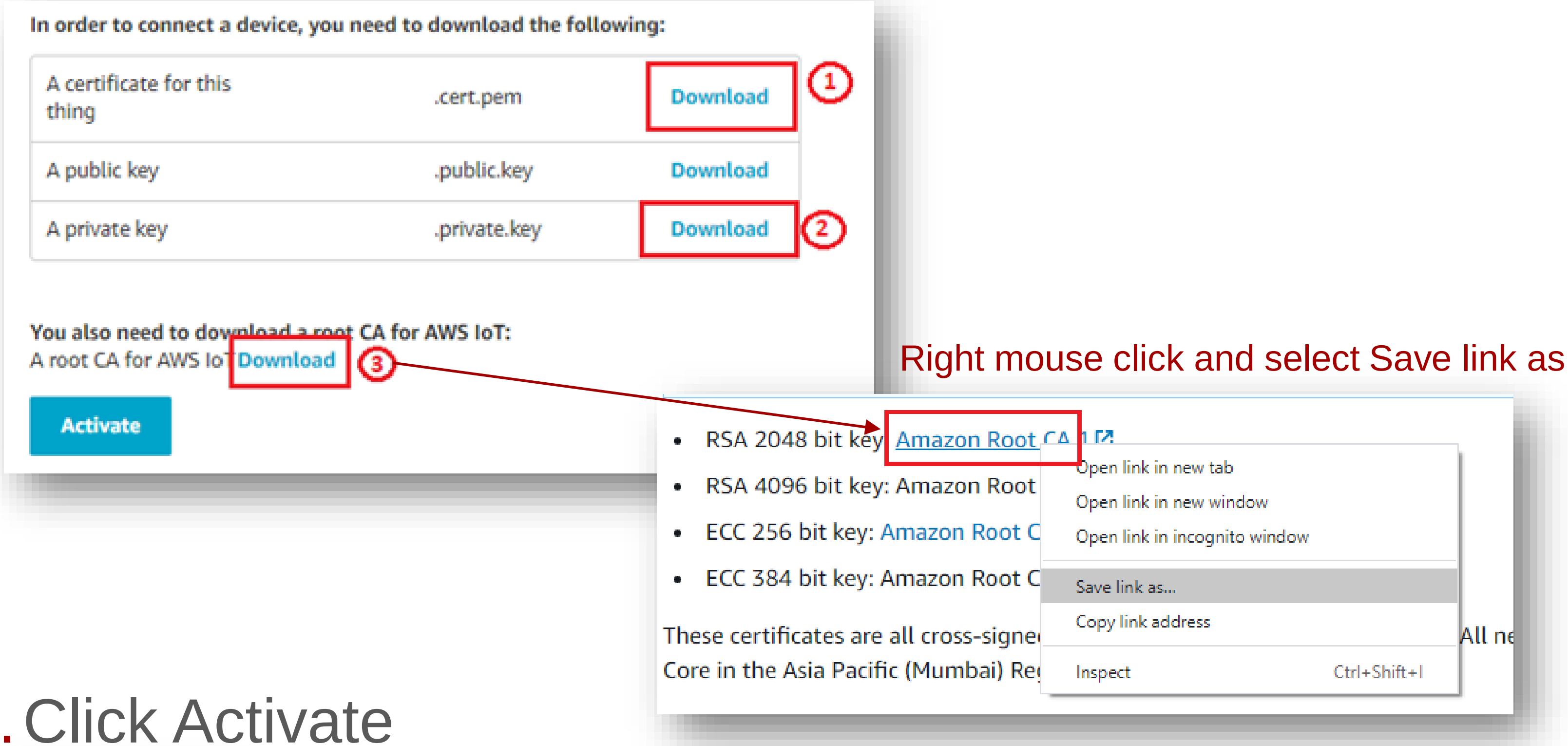
Use my certificate

Register your CA certificate and use your own certificates for one or many devices.

AWS IoT Core configuration

Define Things

14. Download and save the following 3 files



In order to connect a device, you need to download the following:

A certificate for this thing	.cert.pem	Download ①
A public key	.public.key	Download
A private key	.private.key	Download ②

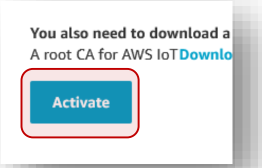
You also need to download a root CA for AWS IoT:
A root CA for AWS IoT Download ③

Right mouse click and select Save link as

- RSA 2048 bit key: Amazon Root CA 1
- RSA 4096 bit key: Amazon Root
- ECC 256 bit key: Amazon Root C
- ECC 384 bit key: Amazon Root C

These certificates are all cross-signe ... All ne
Core in the Asia Pacific (Mumbai) Re

15. Click Activate

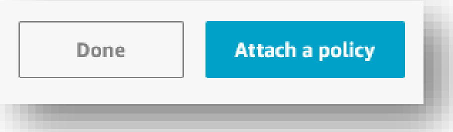


You also need to download a
A root CA for AWS IoT Downlo

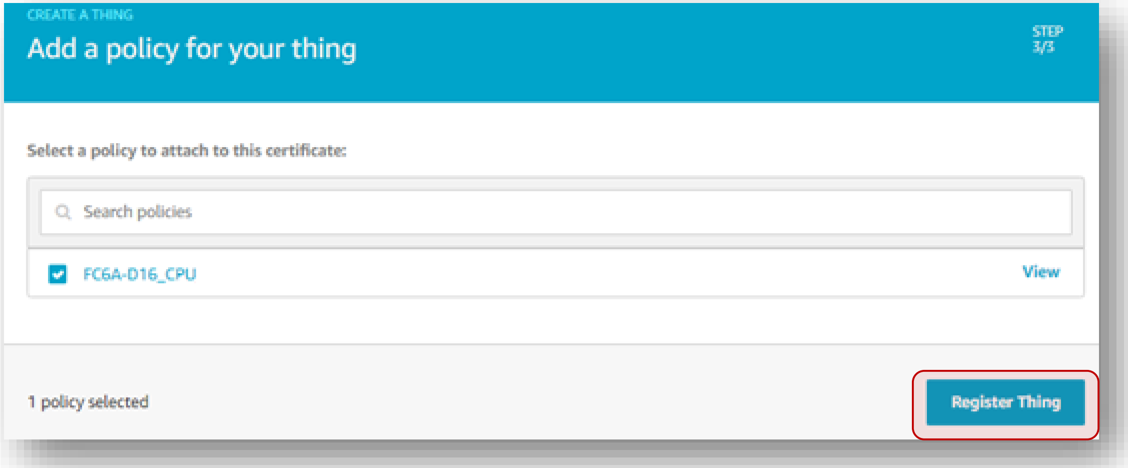
AWS IoT Core configuration

Define Things

16. Click *Attach a policy*



17. Check the “FC6A-D16_CPU” box and click Register Thing


CREATE A THING

Add a policy for your thing

STEP 3/3

Select a policy to attach to this certificate:

Search policies

☑ FC6A-D16_CPU View

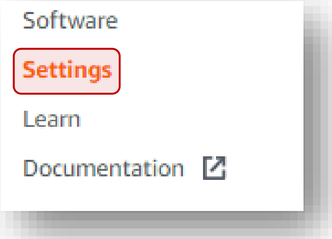
1 policy selected

Register Thing

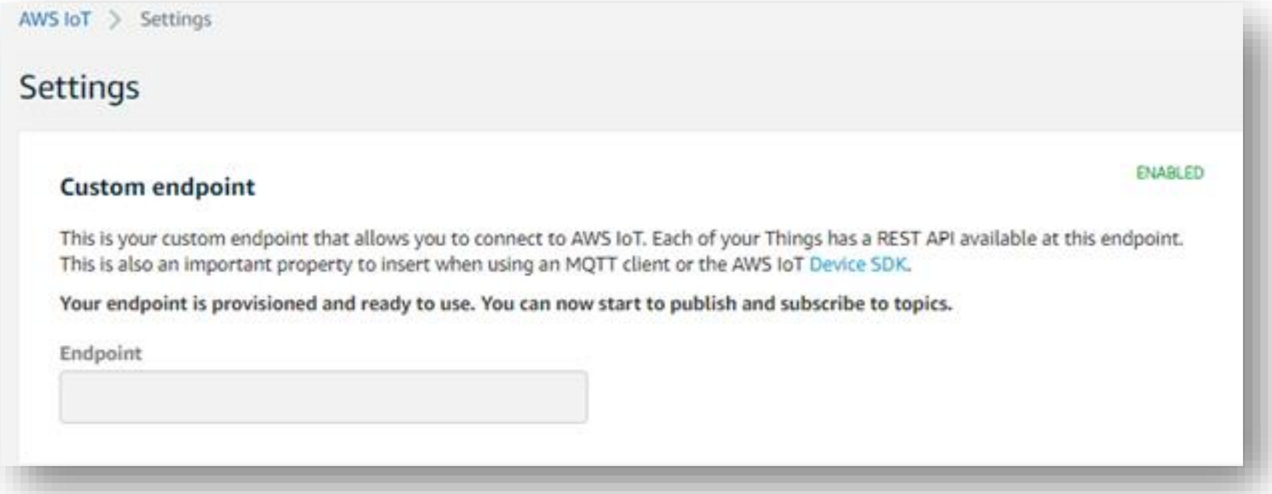
AWS IoT Core configuration

Confirm Endpoint (IP Address/Host Name)

18. Click *Settings*



19. Confirm the Endpoint



Settings

Custom endpoint

ENABLED

This is your custom endpoint that allows you to connect to AWS IoT. Each of your Things has a REST API available at this endpoint. This is also an important property to insert when using an MQTT client or the AWS IoT Device SDK.

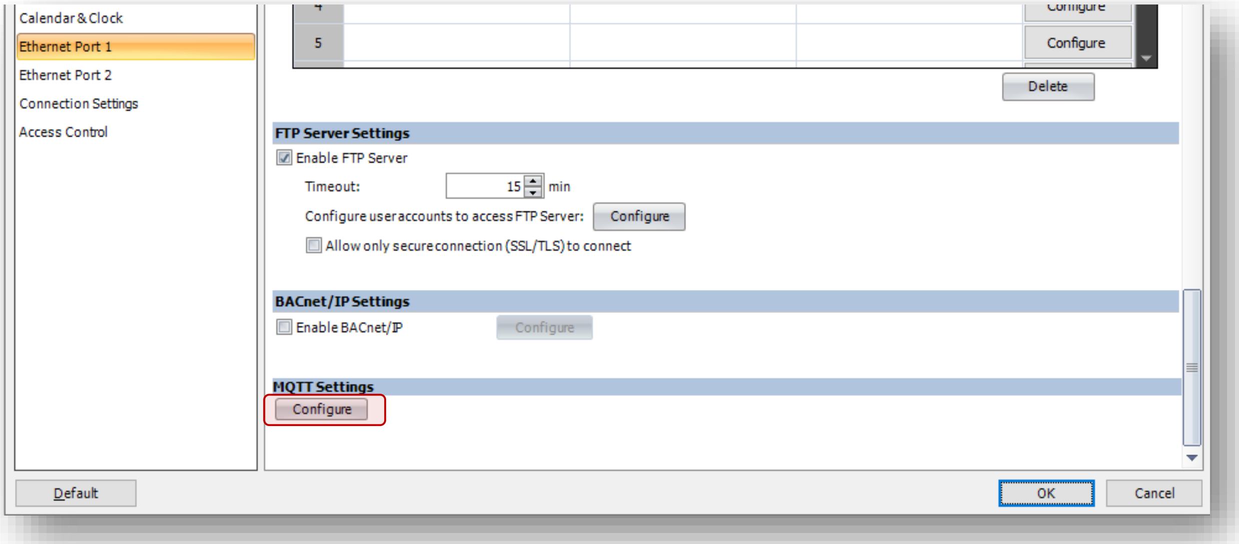
Your endpoint is provisioned and ready to use. You can now start to publish and subscribe to topics.

Endpoint

WindLDR configuration

WindLDR configuration

1. Launch WindLDR version 8.17.00 or later
2. Under Configuration tab, click *Ethernet Port 1*
3. Under MQTT Settings, click *Configure*



FTP Server Settings

☑ Enable FTP Server

Timeout: 15 min

Configure user accounts to access FTP Server:

□ Allow only secure connection (SSL/TLS) to connect

BACnet/IP Settings

□ Enable BACnet/IP

MQTT Settings

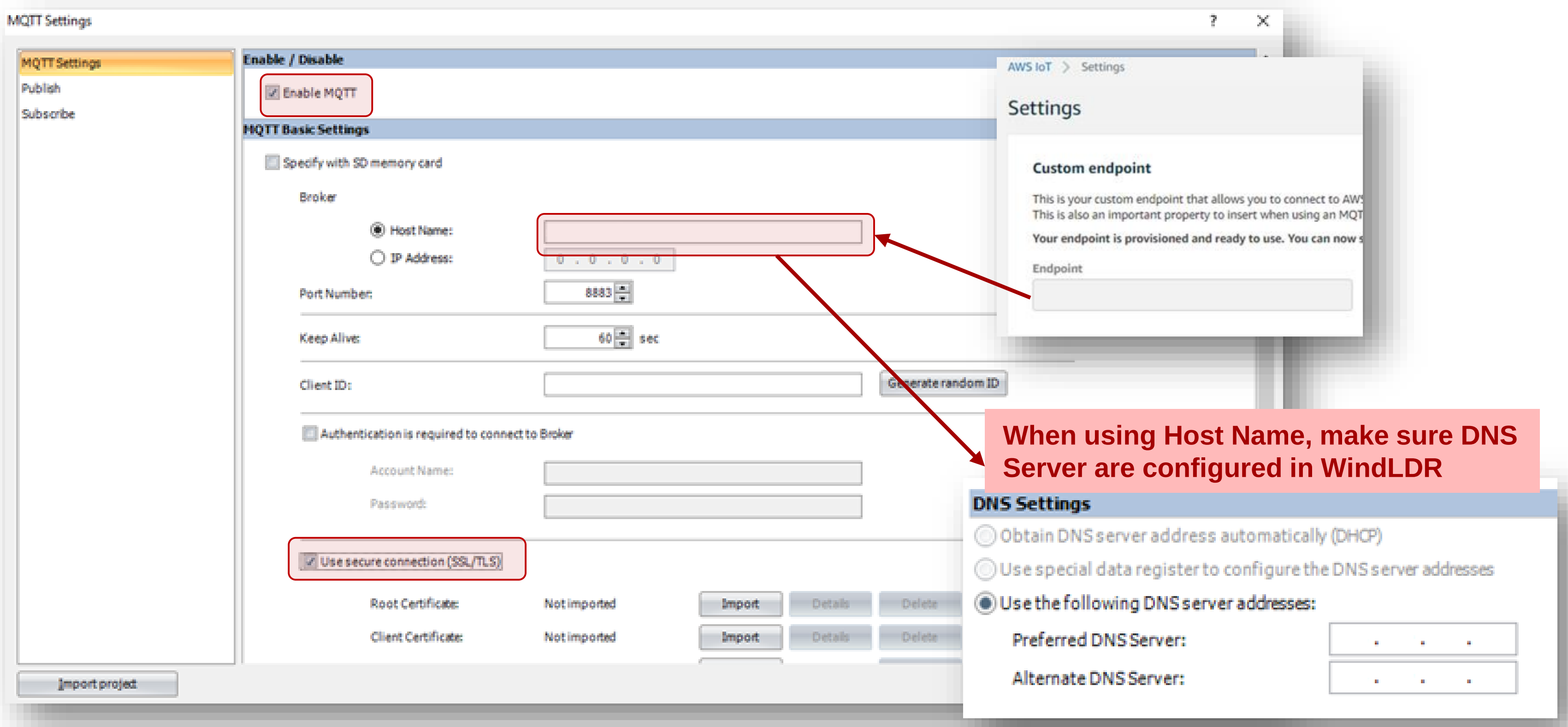
Reference:

- For WindLDR software, refer to the following website. https://us.idec.com/idec-us/en/USD/Software/WindLDR-PLC-Software/c/WindLDR
- If your software version is not greater than 8.17.00, update your software. https://us.idec.com/idec-us/en/USD/Software-Downloads-Automation-Organizer

WindLDR configuration

4. Configure the following:
   - Check the box Enable MQTT
   - Host Name = Endpoint in AWS IoT Core
   - Check the box Secure connection (SSL/TLS)

     Note: When this box is checked, Port Number switched to 8883. Make sure this Port Number is open if the FC6A is connected to a company local area network behind a firewall



MQTT Settings

- MQTT Settings
- Publish
- Subscribe

Import project

Enable / Disable

☑ Enable MQTT

MQTT Basic Settings

□ Specify with SD memory card

Broker

● Host Name:

○ IP Address: 0 . 0 . 0 . 0

Port Number: 8883

Keep Alive: 60 sec

Client ID:

Generate random ID

□ Authentication is required to connect to Broker

Account Name:

Password:

☑ Use secure connection (SSL/TLS)

Root Certificate: Not imported — Import, Details, Delete

Client Certificate: Not imported — Import, Details, Delete

AWS IoT > Settings

Settings

Custom endpoint

This is your custom endpoint that allows you to connect to AWS
This is also an important property to insert when using an MQT

Your endpoint is provisioned and ready to use. You can now s

Endpoint

When using Host Name, make sure DNS Server are configured in WindLDR

DNS Settings

○ Obtain DNS server address automatically (DHCP)

○ Use special data register to configure the DNS server addresses

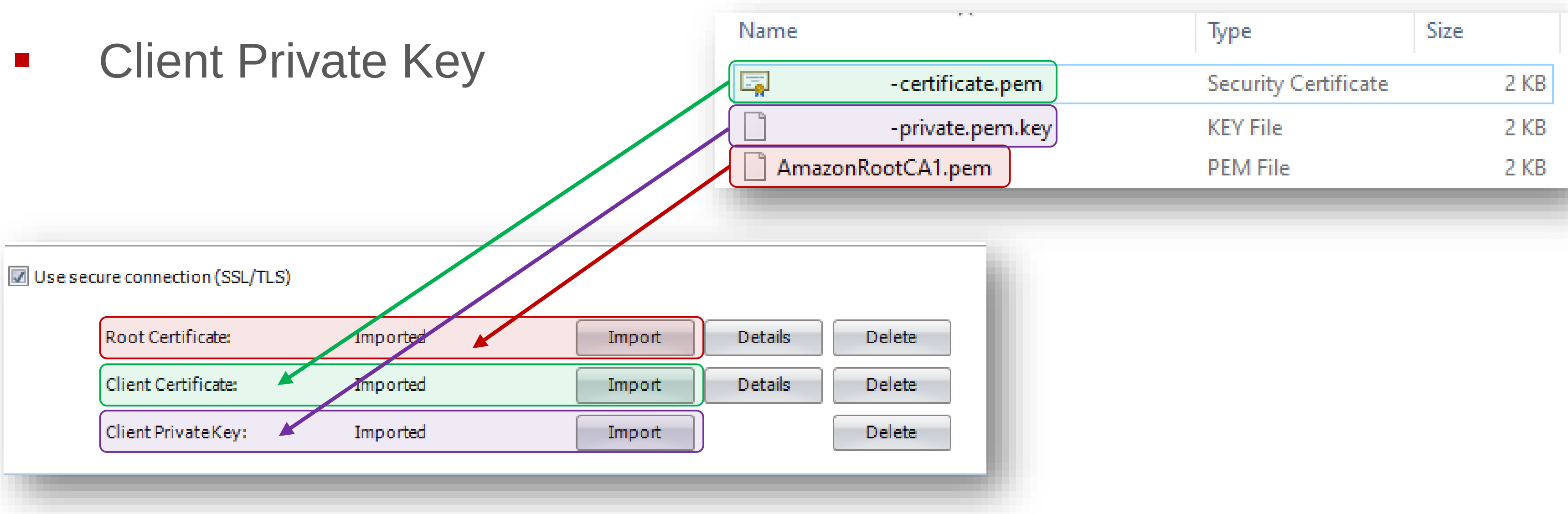
● Use the following DNS server addresses:

Preferred DNS Server: . . .

Alternate DNS Server: . . .

WindLDR configuration

5. Click Import and locate the following files (refer to steps 14-15 on page 9)
   - Root Certificate
   - Client Certificate
   - Client Private Key



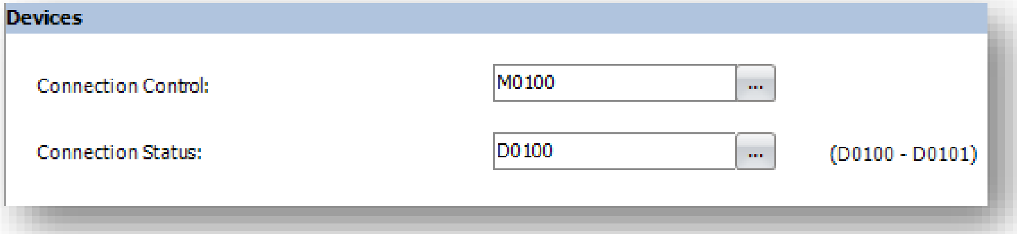
Name	Type	Size
-certificate.pem	Security Certificate	2 KB
-private.pem.key	KEY File	2 KB
AmazonRootCA1.pem	PEM File	2 KB

☑ Use secure connection (SSL/TLS)

Root Certificate:	Imported	Import	Details	Delete
Client Certificate:	Imported	Import	Details	Delete
Client Private Key:	Imported	Import		Delete

WindLDR configuration

6. Configure Connection Control and Status registers
   - Connection Control: MQTT enable/disable connection bit (M100)
   - Connection Status: MQTT connection status registers (D100, used two registers)

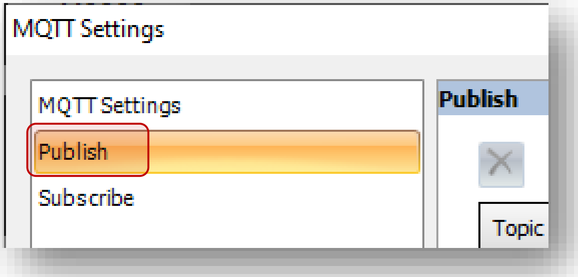


Devices

Connection Control: M0100

Connection Status: D0100 (D0100 - D0101)

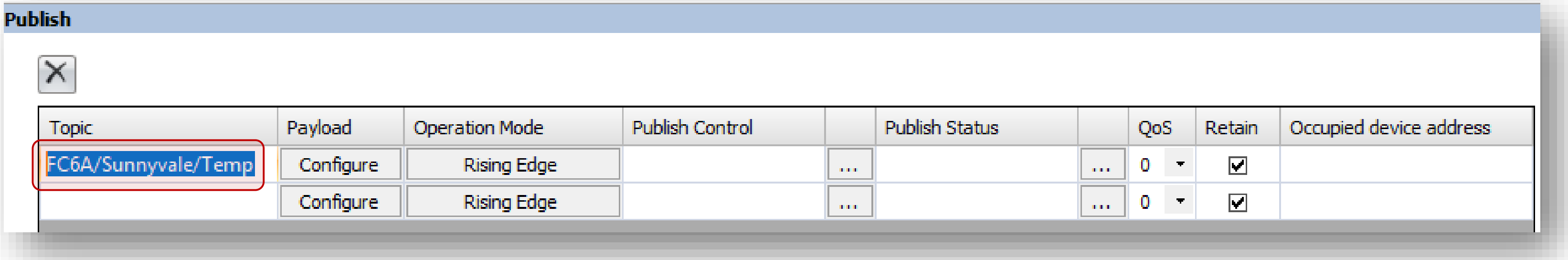
7. Click *Publish*



MQTT Settings

WindLDR configuration

8. Under topic, create a topic you want to publish (example FC6A/Sunnyvale/Temp)



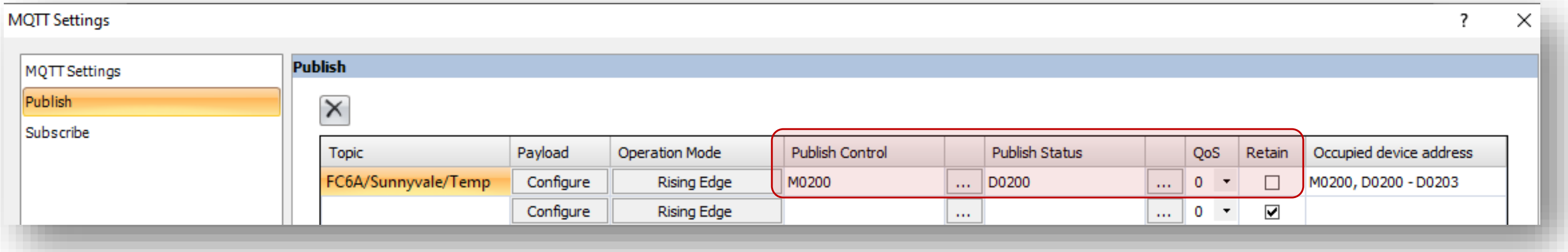
Publish

Topic	Payload	Operation Mode	Publish Control		Publish Status		QoS	Retain	Occupied device address
FC6A/Sunnyvale/Temp	Configure	Rising Edge		...		...	0	☑	
	Configure	Rising Edge		...		...	0	☑	

9. Configure Publish Control and Status

- Publish Control: Enable topic bit (M200)
- Publish Status: Status registers (D200, used four registers)

Note: **Make sure Retain is Unchecked**



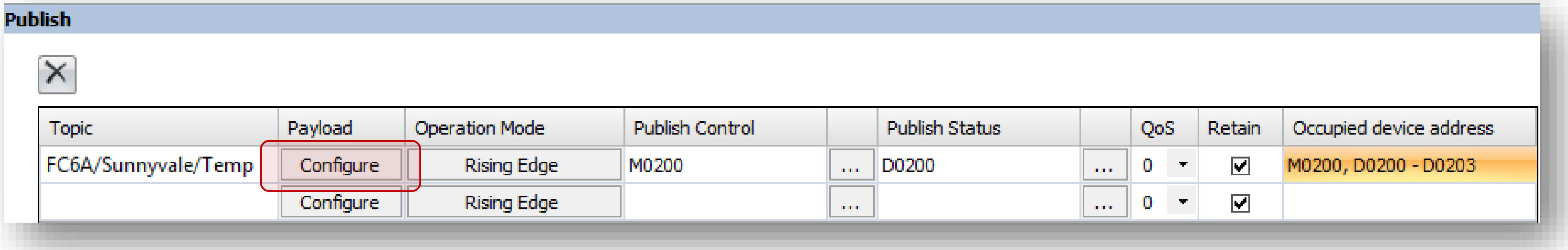
MQTT Settings

Publish

Topic	Payload	Operation Mode	Publish Control		Publish Status		QoS	Retain	Occupied device address
FC6A/Sunnyvale/Temp	Configure	Rising Edge	M0200	...	D0200	...	0	☐	M0200, D0200 - D0203
	Configure	Rising Edge		...		...	0	☑	

WindLDR configuration

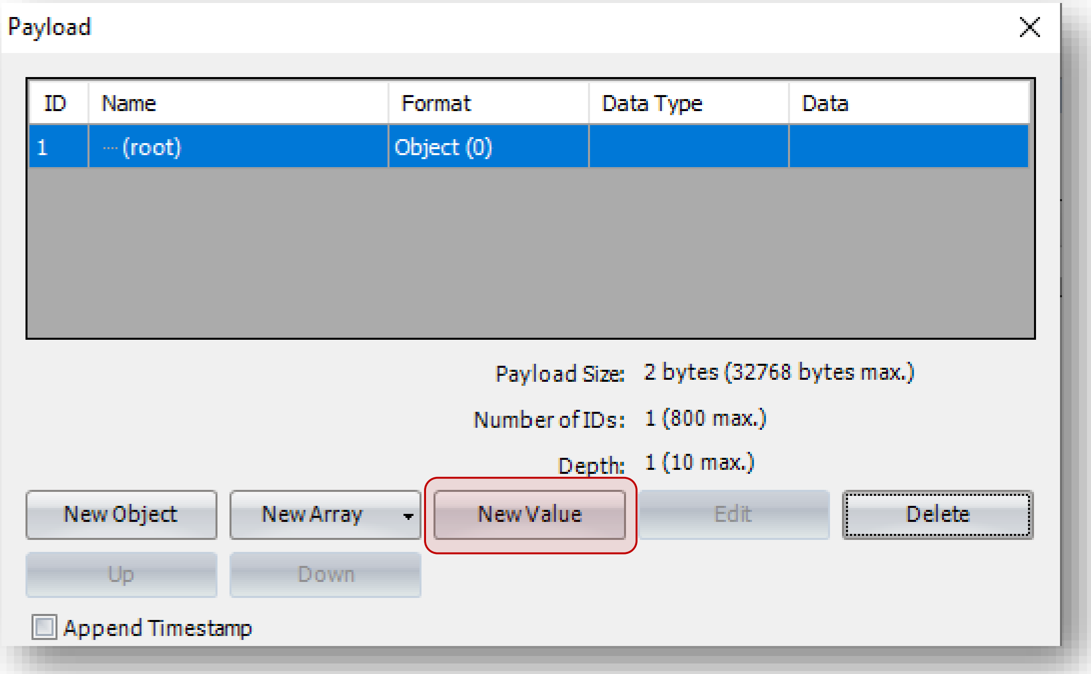
10. Under Payload, click on *Configure*



Publish

Topic	Payload	Operation Mode	Publish Control		Publish Status		QoS	Retain	Occupied device address
FC6A/Sunnyvale/Temp	Configure	Rising Edge	M0200	...	D0200	...	0	☑	M0200, D0200 - D0203
	Configure	Rising Edge		...		...	0	☑	

11. In the Payload dialog, click *New Value*



Payload

ID	Name	Format	Data Type	Data
1	(root)	Object (0)		

Payload Size: 2 bytes (32768 bytes max.)

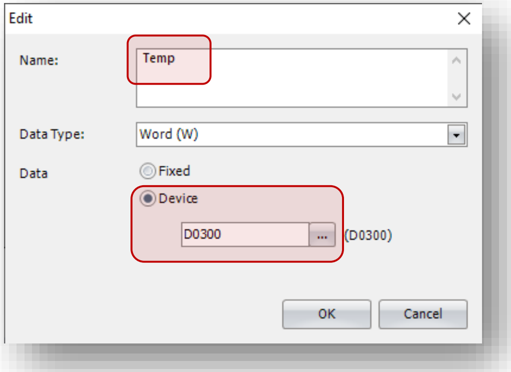
Number of IDs: 1 (800 max.)

Depth: 1 (10 max.)

□ Append Timestamp

WindLDR configuration

12. Select Device and enter D300. Change the name to Temp

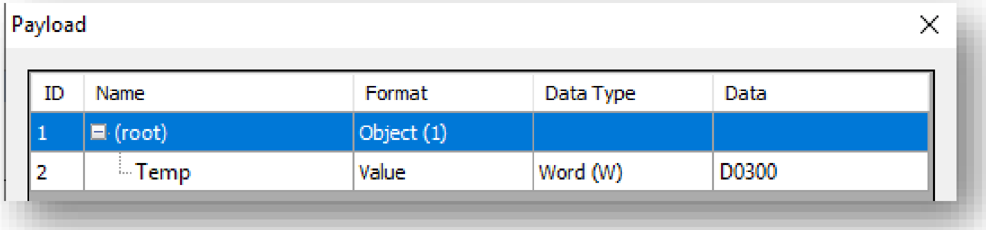


Name: Temp

Data Type: Word (W)

Data
○ Fixed
● Device
D0300 (D0300)

13. Click OK to complete



ID	Name	Format	Data Type	Data
1	⊟ (root)	Object (1)		
2	Temp	Value	Word (W)	D0300

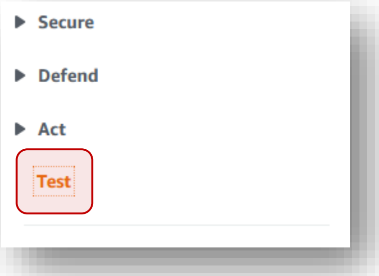
14. Download project.
    - Note:
        - Make sure the PLC firmware version is greater than 1.80.
        - The downloaded firmware version can be checked from “Monitor” -> “Status”.
        - If the firmware version is old, download the firmware from “Online” -> “Download” with “Download system software” option.

Testing

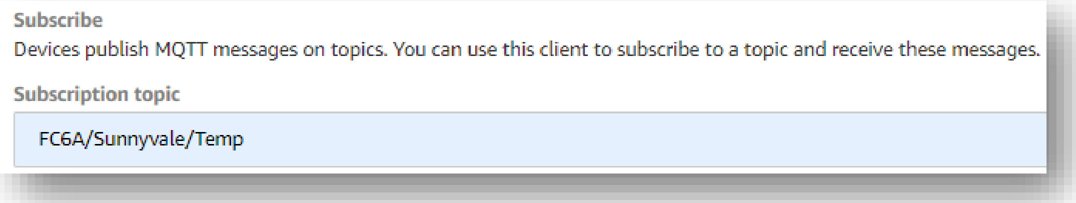
AWS IoT Core – Test

Subscribe to Topic

1. In AWS IoT Core console, click *Test*



2. Under Subscription topic, enter the topic we configured in WindLDR (FC6A/Sunnyvale/Temp)



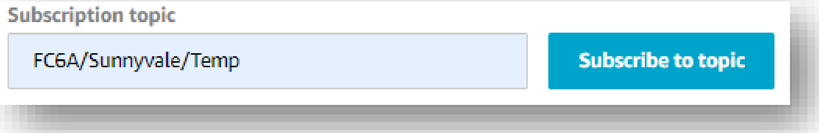
Subscribe

Devices publish MQTT messages on topics. You can use this client to subscribe to a topic and receive these messages.

Subscription topic

FC6A/Sunnyvale/Temp

3. Click *Subscribe to topic*



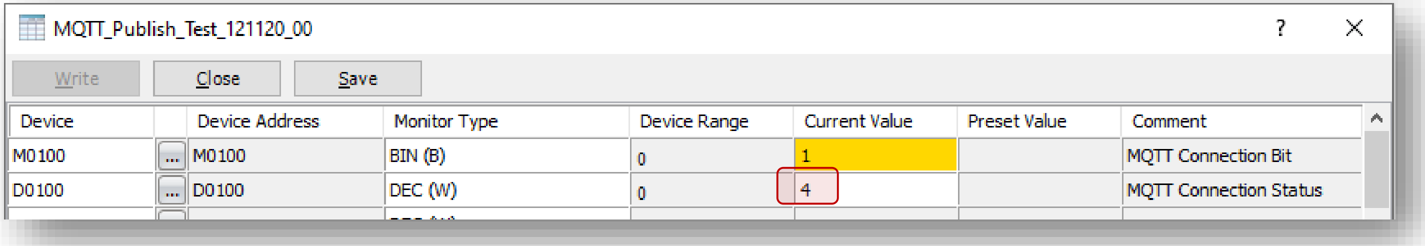
Subscription topic

FC6A/Sunnyvale/Temp

AWS IoT Core – Test

WindLDR monitor mode

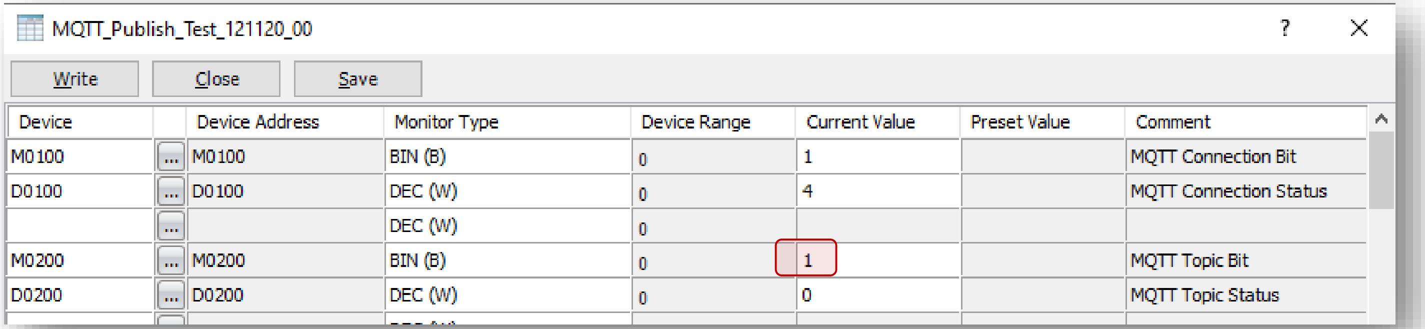
4. Turn On M100 MQTT connection bit
5. Wait until D100 value is 4 (connected)



MQTT_Publish_Test_121120_00

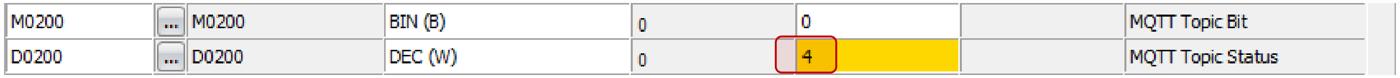
Device		Device Address	Monitor Type	Device Range	Current Value	Preset Value	Comment
M0100	...	M0100	BIN (B)	0	1		MQTT Connection Bit
D0100	...	D0100	DEC (W)	0	4		MQTT Connection Status

6. Turn On M200 MQTT topic bit. D200 returned a value of 4 if successful



MQTT_Publish_Test_121120_00

Device		Device Address	Monitor Type	Device Range	Current Value	Preset Value	Comment
M0100	...	M0100	BIN (B)	0	1		MQTT Connection Bit
D0100	...	D0100	DEC (W)	0	4		MQTT Connection Status
	...		DEC (W)	0			
M0200	...	M0200	BIN (B)	0	1		MQTT Topic Bit
D0200	...	D0200	DEC (W)	0	0		MQTT Topic Status

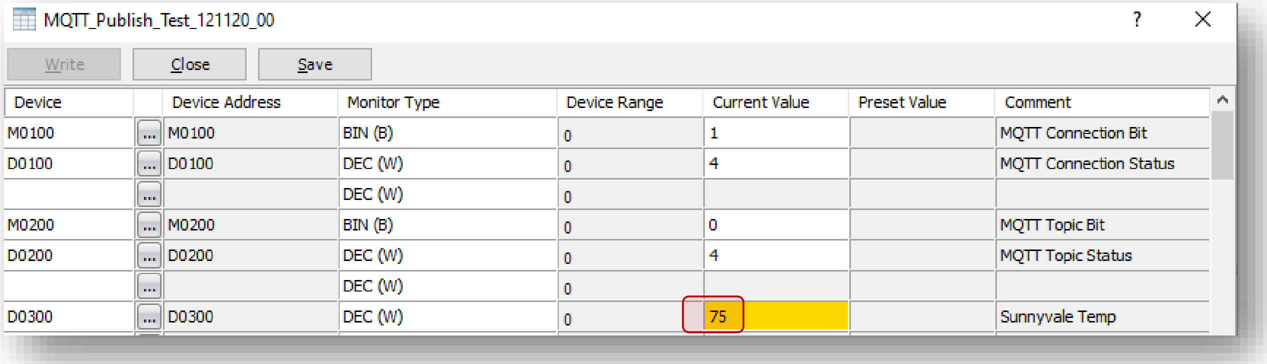


Device		Device Address	Monitor Type	Device Range	Current Value	Preset Value	Comment
M0200	...	M0200	BIN (B)	0	0		MQTT Topic Bit
D0200	...	D0200	DEC (W)	0	4		MQTT Topic Status

AWS IoT Core – Test

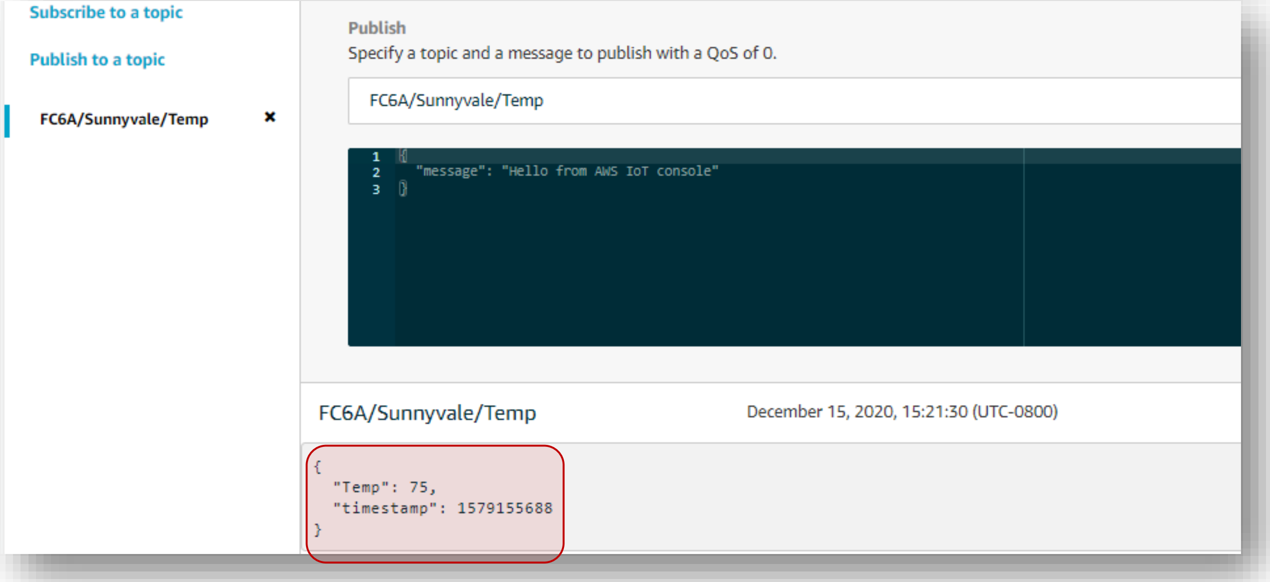
Subscribe to Topic

7. The value in D300 will be seen in AWS console



MQTT_Publish_Test_121120_00

Device		Device Address	Monitor Type	Device Range	Current Value	Preset Value	Comment
M0100	...	M0100	BIN (B)	0	1		MQTT Connection Bit
D0100	...	D0100	DEC (W)	0	4		MQTT Connection Status
	...		DEC (W)	0			
M0200	...	M0200	BIN (B)	0	0		MQTT Topic Bit
D0200	...	D0200	DEC (W)	0	4		MQTT Topic Status
	...		DEC (W)	0			
D0300	...	D0300	DEC (W)	0	75		Sunnyvale Temp



Publish

Specify a topic and a message to publish with a QoS of 0.

FC6A/Sunnyvale/Temp

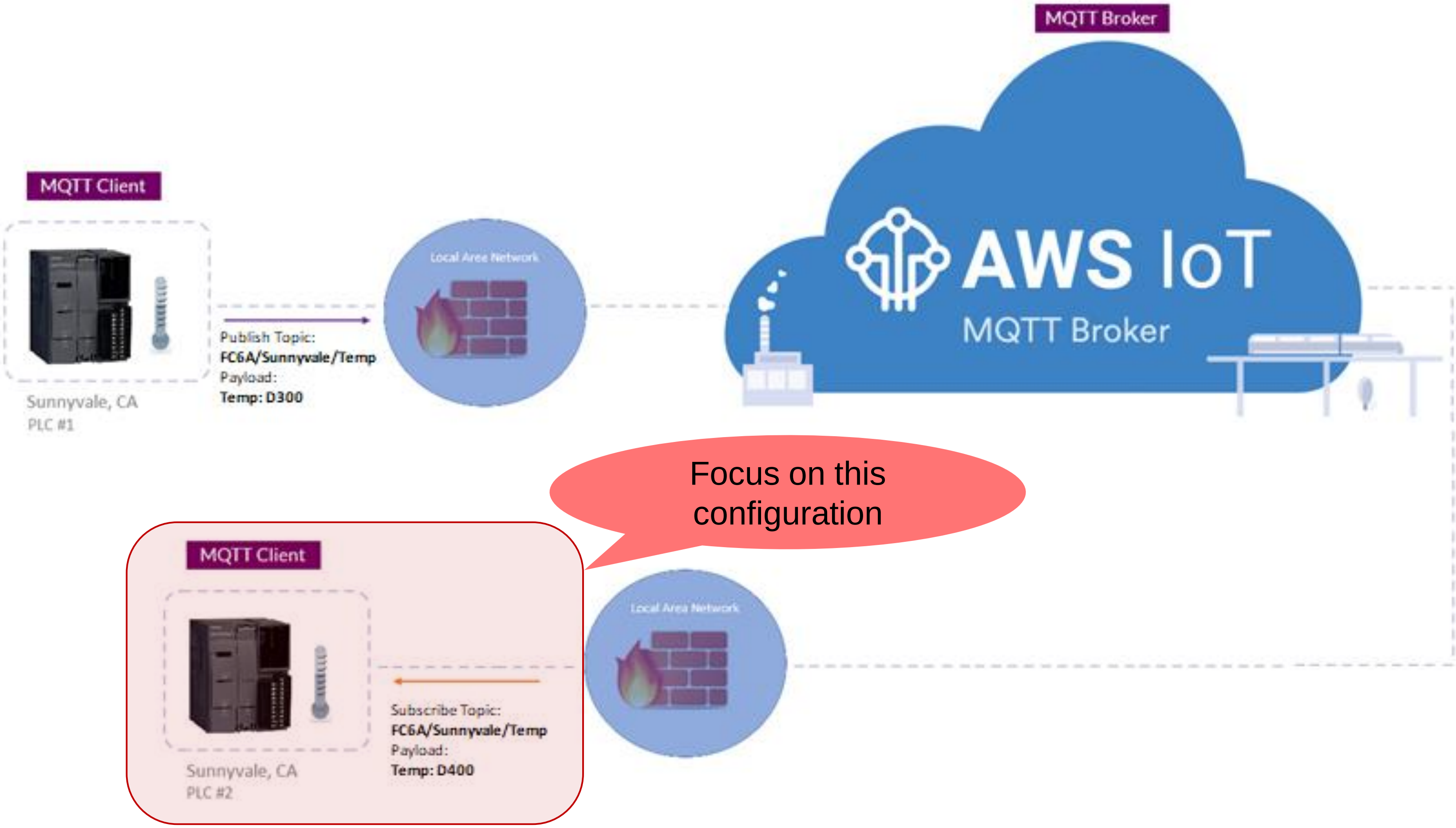
```
{
  "message": "Hello from AWS IoT console"
}
```

FC6A/Sunnyvale/Temp December 15, 2020, 15:21:30 (UTC-0800)

```
{
  "Temp": 75,
  "timestamp": 1579155688
}
```

Example 2 - Subscribe

- One FC6A Plus CPU is configured as Publisher
- A second FC6A Plus CPU is configured as Subscriber

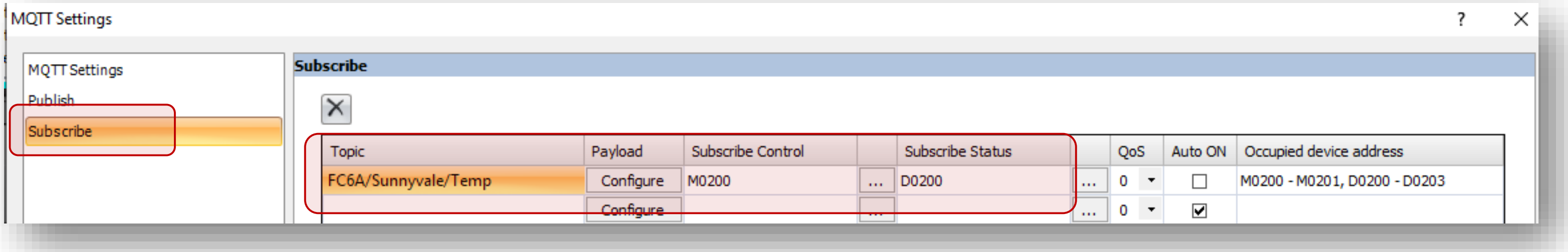


Focus on this configuration

Subscribe

Subscribe to Topic

1. AWS IoT Core
   - ➢ Repeat step 1-19 on page 4-12
2. WindLDR
   - ➢ Repeat step 1-6 on page 14-17
3. Click *Subscribe* and enter the following:
   - Topic: FC6A/Sunnyvale/Temp
   - Subscribe Control: Enable bit (M200, used two bits)
   - Subscribe Status: Status registers (D200, used four registers)

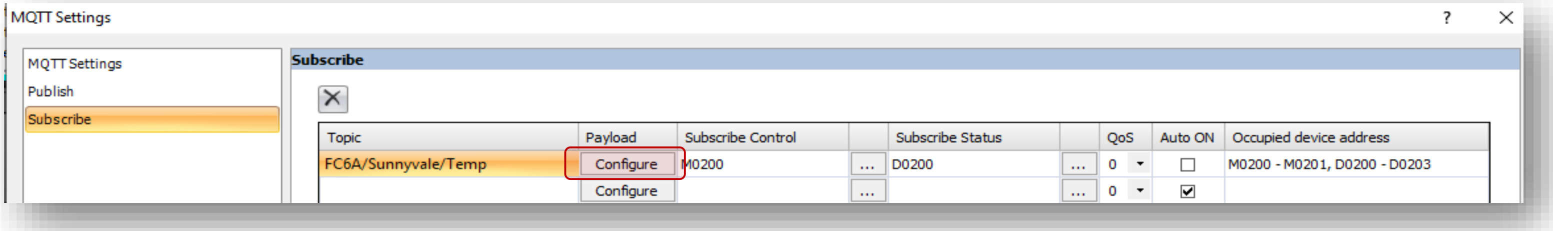


Topic	Payload	Subscribe Control		Subscribe Status		QoS	Auto ON	Occupied device address
FC6A/Sunnyvale/Temp	Configure	M0200	...	D0200	...	0	□	M0200 - M0201, D0200 - D0203
	Configure		...		...	0	☑	

Subscribe

Subscribe to Topic

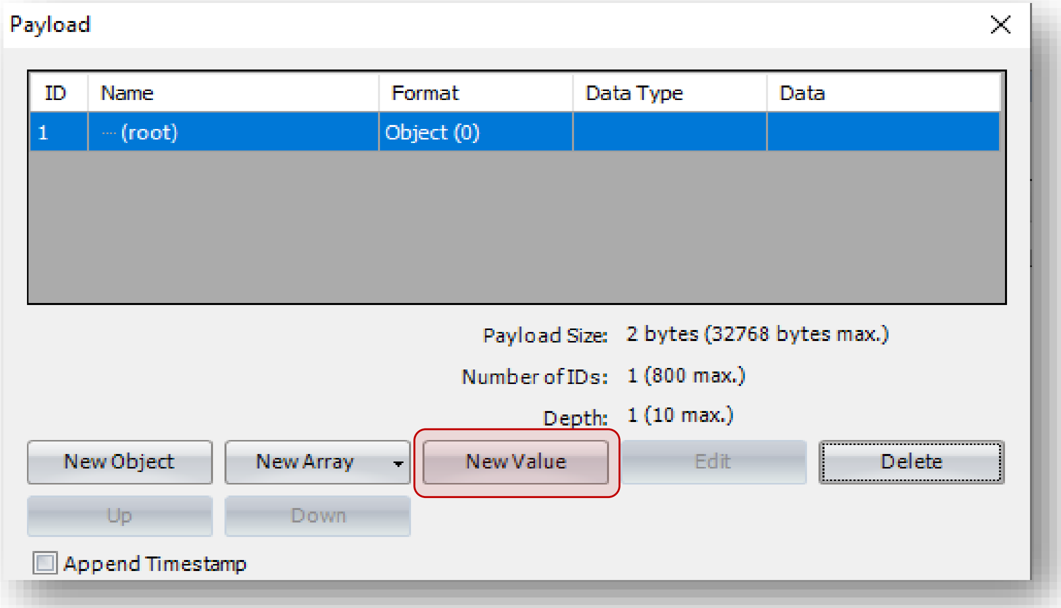
4. Under Payload, click *Configure*



Subscribe

Topic	Payload	Subscribe Control		Subscribe Status		QoS	Auto ON	Occupied device address
FC6A/Sunnyvale/Temp	Configure	M0200	...	D0200	...	0	□	M0200 - M0201, D0200 - D0203
	Configure		...		...	0	☑	

5. In the Payload dialog, click *New Value*



ID	Name	Format	Data Type	Data
1	(root)	Object (0)		

Payload Size: 2 bytes (32768 bytes max.)

Number of IDs: 1 (800 max.)

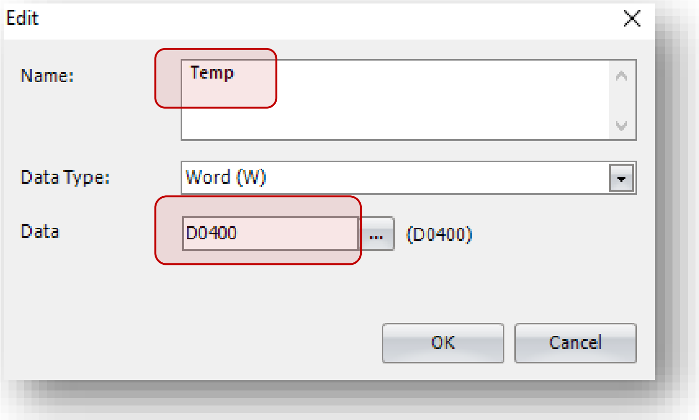
Depth: 1 (10 max.)

□ Append Timestamp

Subscribe

Subscribe to Topic

7. Enter Name (Temp) and Data (D400)

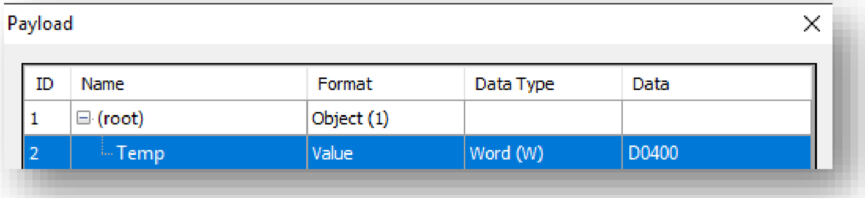

Edit

Name: Temp

Data Type: Word (W)

Data: D0400 (D0400)

8. Click OK

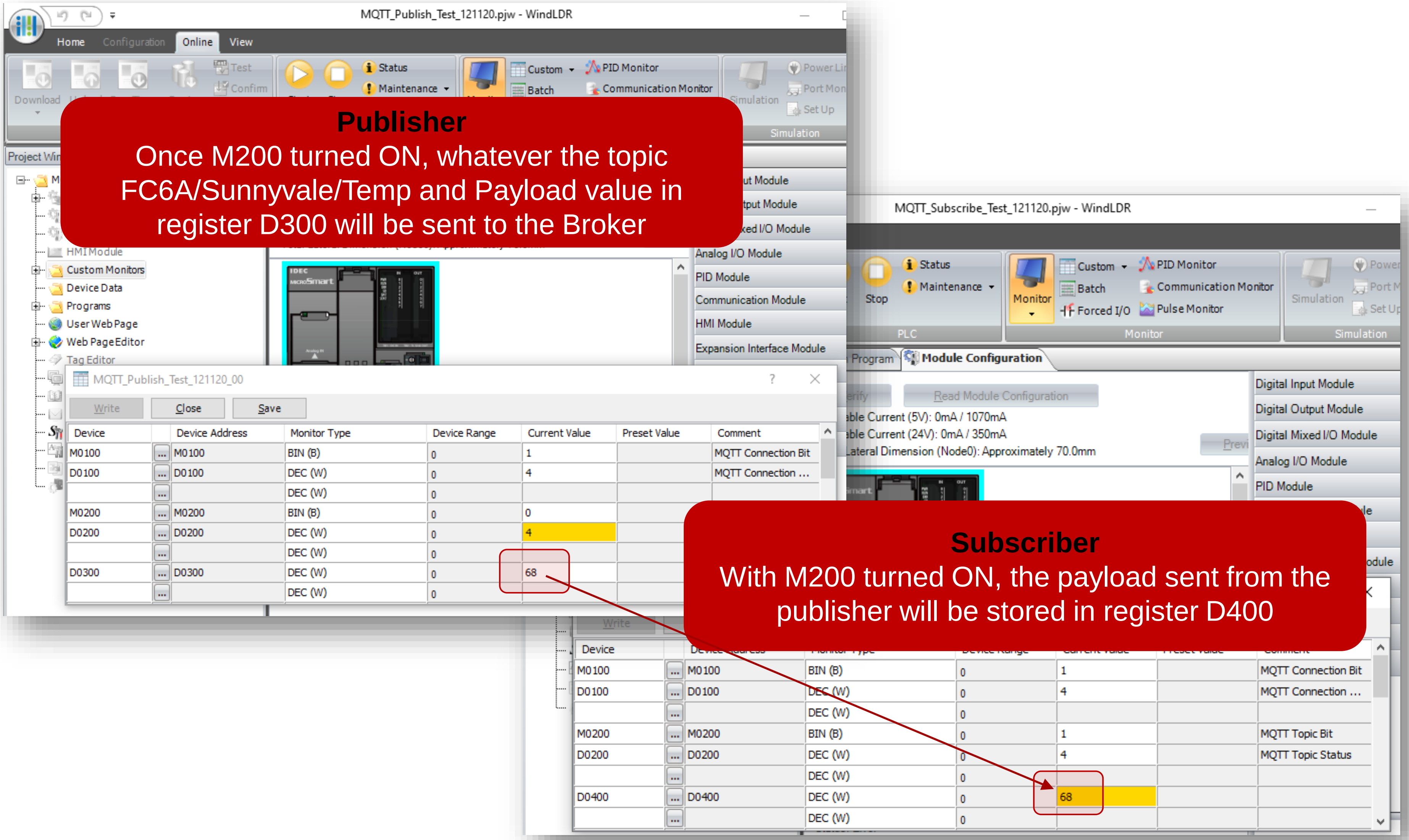

Payload

ID	Name	Format	Data Type	Data
1	⊟ (root)	Object (1)		
2	Temp	Value	Word (W)	D0400

9. Download project.

Subscribe

Test



MQTT_Publish_Test_121120.pjw - WindLDR

MQTT_Publish_Test_121120_00

Device	Device Address	Monitor Type	Device Range	Current Value	Preset Value	Comment
M0100	M0100	BIN (B)	0	1		MQTT Connection Bit
D0100	D0100	DEC (W)	0	4		MQTT Connection ...
		DEC (W)	0			
M0200	M0200	BIN (B)	0	0		
D0200	D0200	DEC (W)	0	4		
		DEC (W)	0			
D0300	D0300	DEC (W)	0	68		
		DEC (W)	0			

Publisher
Once M200 turned ON, whatever the topic FC6A/Sunnyvale/Temp and Payload value in register D300 will be sent to the Broker

MQTT_Subscribe_Test_121120.pjw - WindLDR

able Current (5V): 0mA / 1070mA
able Current (24V): 0mA / 350mA
ateral Dimension (Node0): Approximately 70.0mm

Device	Device Address	Monitor Type	Device Range	Current Value	Preset Value	Comment
M0100	M0100	BIN (B)	0	1		MQTT Connection Bit
D0100	D0100	DEC (W)	0	4		MQTT Connection ...
		DEC (W)	0			
M0200	M0200	BIN (B)	0	1		MQTT Topic Bit
D0200	D0200	DEC (W)	0	4		MQTT Topic Status
		DEC (W)	0			
D0400	D0400	DEC (W)	0	68		
		DEC (W)	0			

Subscriber
With M200 turned ON, the payload sent from the publisher will be stored in register D400

Troubleshooting

Troubleshooting

Error Code and Details

- If the FC6A Plus cannot connect to AWS, Check “MQTT connection status registers”. (In this tutorial, D100 is set as status register)

Status Register Device Address + 0 (D100)	
Status Code	**Status**
0 (0x0000)	Initial status (disconnected)
2 (0x0002)	Connecting
4 (0x0004)	Connected
8 (0x0008)	Disconnecting
16 (0x0010)	Connection processing error
32 (0x0020)	Disconnection processing error

Status Register Device Address + 1 (D101)	
Error Code	**Error Details**
1 (0x0001)	The Ethernet cable is disconnected or broken and the Plus CPU module cannot connect to the network properly
2 (0x0002)	When the Specify with SD memory card check box is selected, authentication information was not downloaded from the SD memory card or reading the downloaded authentication information failed
16 (0x0010)	An unknown packet was received
32 (0x0020)	An invalid MQTT packet was received
64 (0x0040)	Keep alive timeout error
80 (0x0050)	Packet could not arrive at destination host
96 (0x0060)	MQTT packet receive timeout error
112 (0x0070)	TLS error
256 (0x0100)	Broker connection refused (unacceptable MQTT protocol version)
512 (0x0200)	Broker connection refused (invalid client ID)
768 (0x0300)	Broker connection refused (broker unavailable)
1024 (0x0400)	Broker connection refused (invalid account name or password)
1280 (0x0500)	Broker connection refused (not authorized)
32768 (0x8000)	Broker response error