

Getting Started With the Google Cloud Platform SDK on ThingsPro 2.0

Moxa Technical Support Team
support@moxa.com

Contents

- 1 Introduction..... 2**
- 2 Application Scenario 2**
- 3 Prerequisites..... 3**
- 4 Solution 3**
 - 4.1 Set up ThingsPro Gateway to read Modbus data from "Constrained Devices" 4
 - 4.2 Install the Google Cloud Platform SDK on the ThingsPro Gateway..... 7
 - 4.3 Publish and Subscribe test data to/from the Google Cloud Platform Pub/Sub Service 9
 - 4.4 Publish and Subscribe ThingsPro Modbus data to/from the Google Cloud Platform Pub/Sub service 13
- 5 Additional Reading 17**

Copyright © 2017 Moxa Inc.

Released on October 02, 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 info@moxa.com.

How to Contact Moxa

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



1 Introduction

The purpose of this tech note is to provide you with step-by-step instructions on how to run the Google Cloud Platform SDK on ThingsPro.

The instructions in this document are suitable for developers who are already familiar with the Google Cloud Platform console. If you are not familiar with the technology, please visit:

<https://cloud.google.com/solutions/iot-overview>

2 Application Scenario

The UC-8112-LX is an embedded computer with the Debian Linux distribution preinstalled. ThingsPro Gateway, which is part of the ThingsPro Suite, is a software package that acts as a protocol gateway between the IT & OT worlds. On the IT side, it supports a variety of IT protocols such as Microsoft Azure, Wonderware Online Insight, AWS, Generic MQTT, HTTP, Sparkplug, and M2X cloud server. On the OT side, the ThingsPro Gateway supports the Modbus protocol and provides the programming environment for users to develop their own proprietary OT protocol. In this document, we will show you how to leverage ThingsPro's Modbus engine to read data from Modbus devices and transmit this data to the Google Cloud Platform so that it can be used by IT applications.

Figure 1 shows an overview of the Google Cloud Platform architecture. ThingsPro Gateway, in its role as the "gateway device", reads Modbus data from "Constrained Devices". The data is then published to the "Ingestion" function block in the Google Cloud Platform, which uses the "Cloud Pub/Sub" function to manipulate data and publish it to the "Pipelines" function block. To learn more about the publisher/subscriber model of the Google Cloud platform, refer to https://cloud.google.com/solutions/iot-overview#pipeline_processing_tasks.

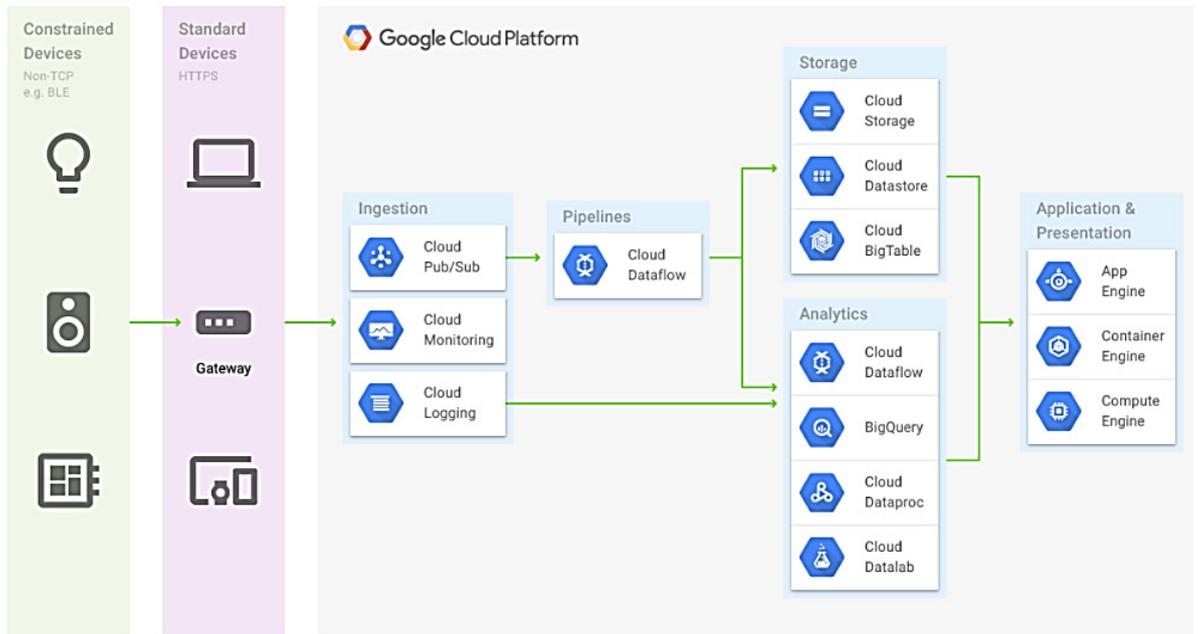


Figure 1

3 Prerequisites

- An eligible* Moxa device (e.g., UC-8112-LX) with the ThingsPro Gateway software installed
- A Google Cloud Platform user account

*Refer to the ThingsPro datasheet for a list of eligible devices.

4 Solution

To transfer Modbus data from ThingsPro to the Google Cloud platform, do the following:

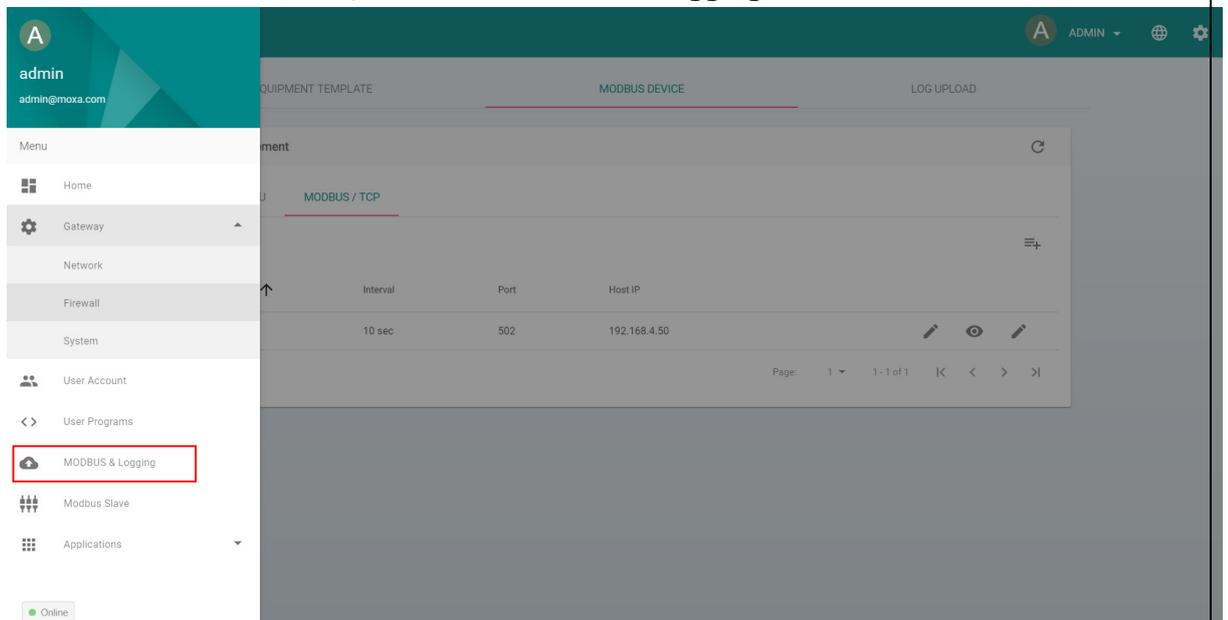
1. Set up ThingsPro Gateway to read Modbus data from "Constrained Devices"
2. Install the Google Cloud Platform SDK on the ThingsPro Gateway
3. Publish and Subscribe test data to/from the Google Cloud Platform Pub/Sub Service
4. Publish and Subscribe ThingsPro Modbus data to/from the Google Cloud Platform Pub/Sub service

4.1 Set up ThingsPro Gateway to read Modbus data from “Constrained Devices”

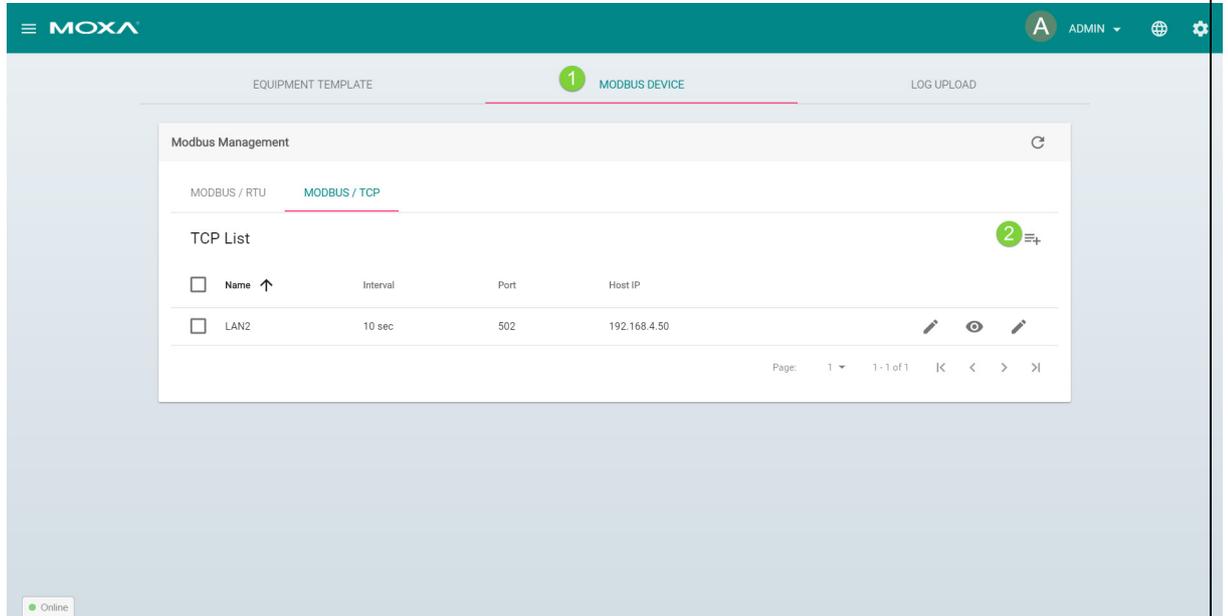
First you need to connect devices to the ThingsPro Gateway. In our example, we use a Moxa ioLogik-E1242 device.

To establish a connection between the ThingsPro Gateway and the ioLogik-E1242 device, do the following:

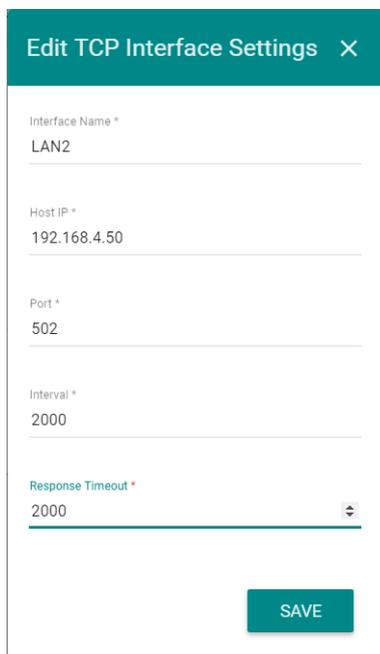
1. Login into the ThingsPro Gateway web console.
2. Click  Main Menu, and select **MODBUS & Logging**



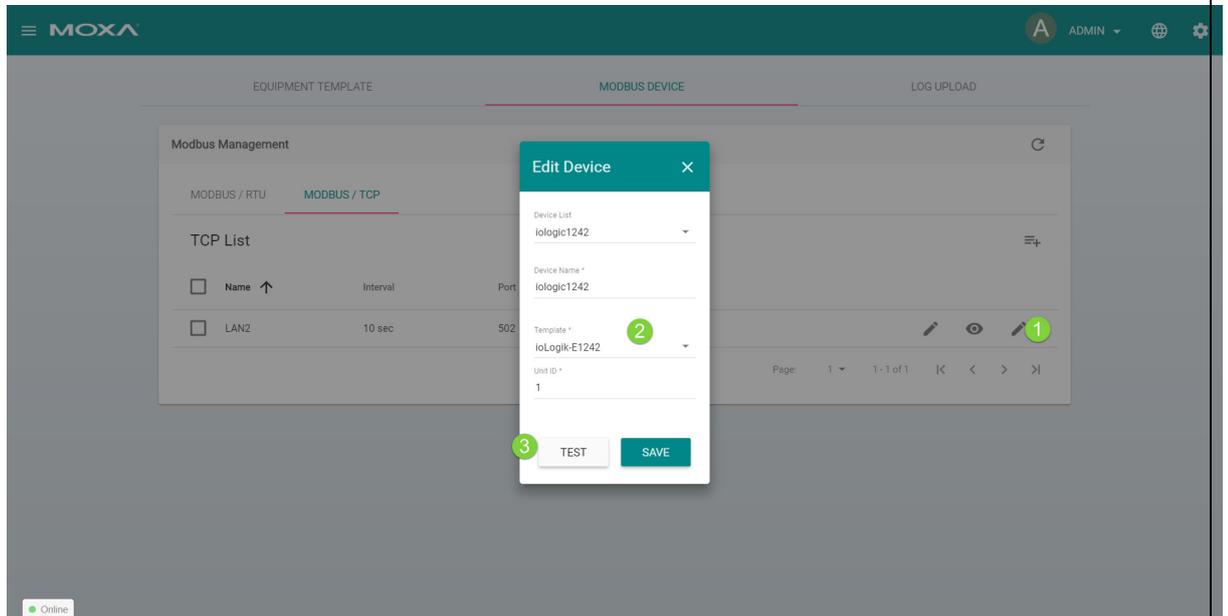
3. Create a new Modbus TCP polling rule.



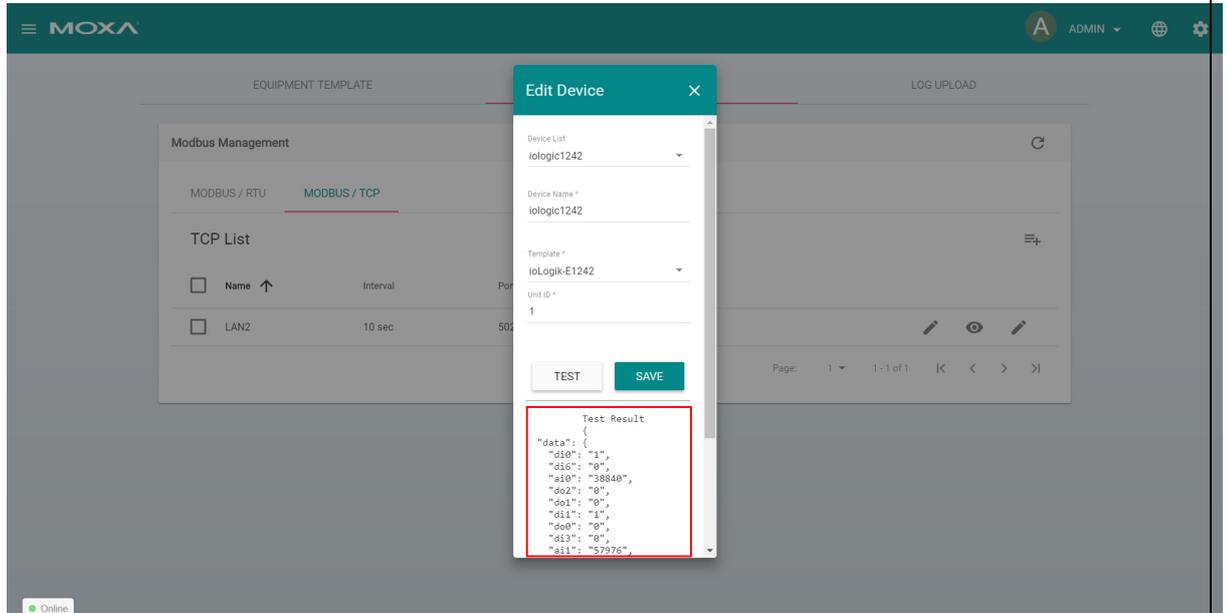
In this example, the iLogik-E1242 Modbus device is set up with the IP address 192.168.4.50. The query interval for the device is set to 2000 milliseconds.



4. Apply the Modbus TCP polling engine rule that you just created to a predefined ioLogik-E1242 Modbus template.



5. Click **TEST** to verify that the Modbus device is polling data.



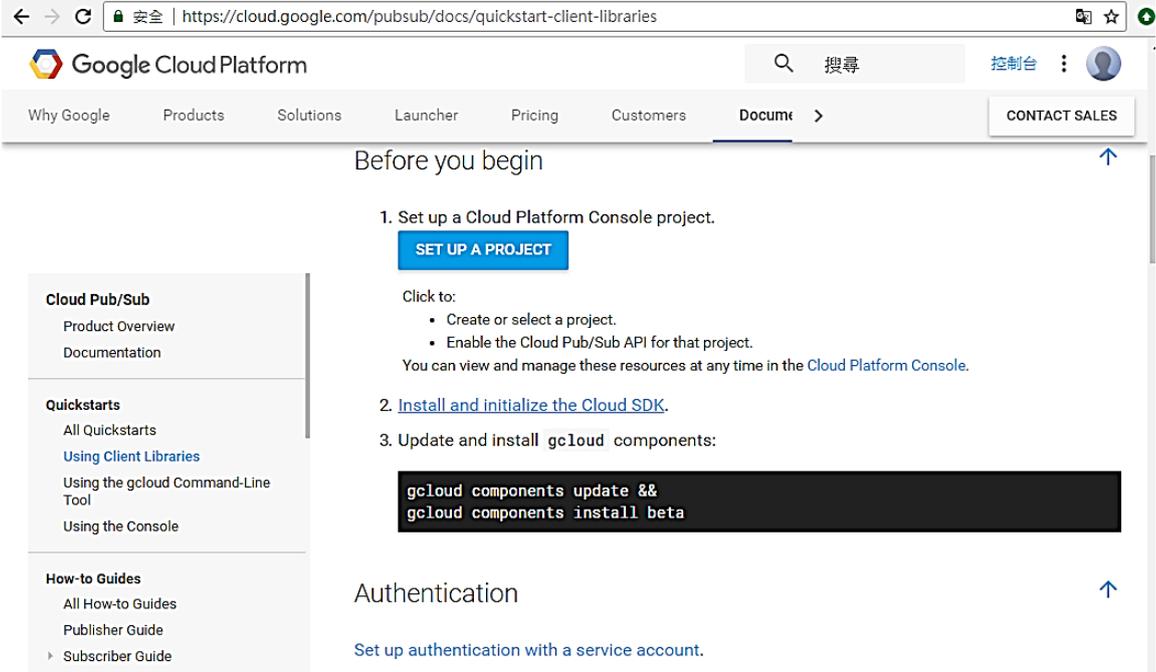
If the test result shows no errors, the Modbus device is set up correctly.

If the test result shows errors, check if the:

- Modbus device is set up with an incorrect IP addresses
- Router firewall is blocking the Modbus protocol
- Modbus device allows only 1 TCP session

4.2 Install the Google Cloud Platform SDK on the ThingsPro Gateway

The Google Cloud Platform SDK is available for download from the Google web site (<https://cloud.google.com/pubsub/docs/quickstart-client-libraries>). You will need a Google account to use the Pub/Sub service. Once you are set up with an account, follow the instructions given in the "Before you begin" section as shown in Figure 2.



The screenshot shows the Google Cloud Platform website at the URL <https://cloud.google.com/pubsub/docs/quickstart-client-libraries>. The page title is "Before you begin". The main content area lists three steps:

1. Set up a Cloud Platform Console project.
A blue button labeled "SET UP A PROJECT" is visible.
Click to:
 - Create or select a project.
 - Enable the Cloud Pub/Sub API for that project.You can view and manage these resources at any time in the Cloud Platform Console.
2. [Install and initialize the Cloud SDK.](#)
3. Update and install `gcloud` components:
A terminal snippet shows the commands:

```
gcloud components update &&  
gcloud components install beta
```

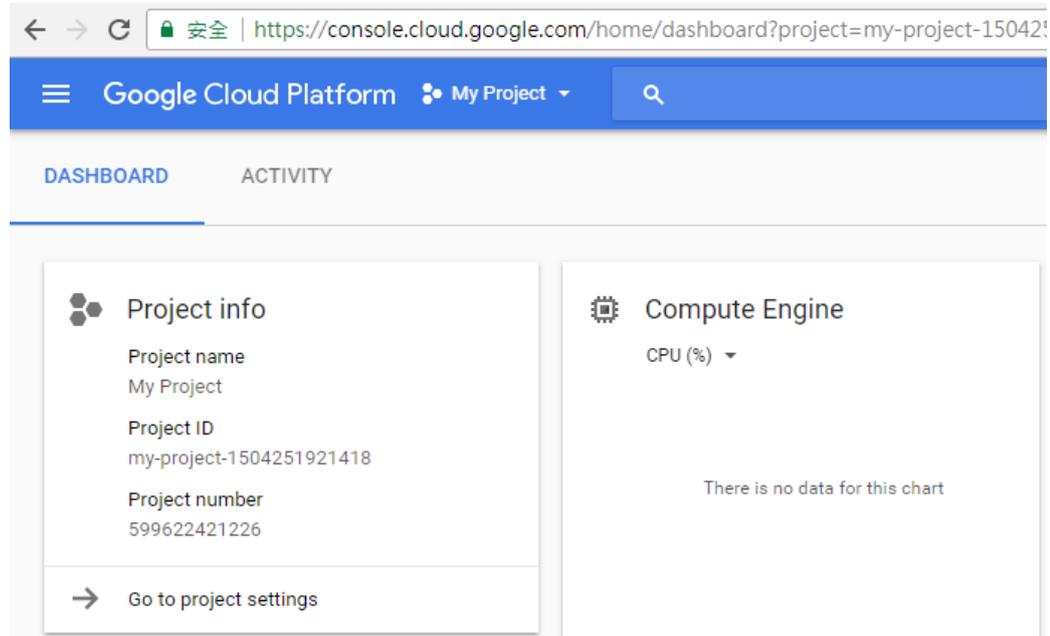
Below the steps, there is a section for "Authentication" with a link: "Set up authentication with a service account."

The left sidebar contains navigation links for "Cloud Pub/Sub", "Quickstarts", and "How-to Guides".

Figure 2

Step 1: Set up a Project

In this example, we create a new project named "My Project". A project ID "my-project-1504251921418" is assigned automatically when the project is created.

**Figure 3****STEP 2: Install and initialize the Cloud SDK**

1. Connect the console cable to the ThingsPro Gateway and set the baudrate to 115200/8N1
2. Login with the username `moxa` and password `moxa`
3. Switch to the root user using the command:

```
# sudo su (default password "moxa")
```
4. Make sure your ThingsPro Gateway can access Internet and run the following commands to install the Google Cloud SDK:

```
# apt-get update && apt-get install lsb-release  
# export CLOUD_SDK_REPO="cloud-sdk-$(lsb_release -c -s)"  
# echo "deb http://packages.cloud.google.com/apt $CLOUD_SDK_REPO main" |  
sudo tee -a /etc/apt/sources.list.d/google-cloud-sdk.list  
# curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo  
apt-key add -  
# sudo apt-get update && sudo apt-get install google-cloud-sdk
```

Wait for about 10 minutes for the installation process to complete. Then, run the following command:

```
# pip install virtualenv
```

5. Once you set up the Google Cloud SDK, run the following commands on the ThingsPro Gateway console to download the Python pub/sub example code.

```
# apt-get install git -y
# cd /home/moxa
# mkdir Google
# cd Google
# git clone
https://github.com/GoogleCloudPlatform/python-docs-samples.git
# cd python-docs-samples/pubsub/cloud-client
```

For details, visit:

<https://github.com/GoogleCloudPlatform/python-docs-samples/tree/master/pubsub/cloud-client>

4.3 Publish and Subscribe test data to/from the Google Cloud Platform Pub/Sub Service

To be able to publish/subscribe data to/from the Google Pub/Sub service, you must first create a new topic in the Google Cloud Platform Pub/Sub Service web console as shown in Figure 4.

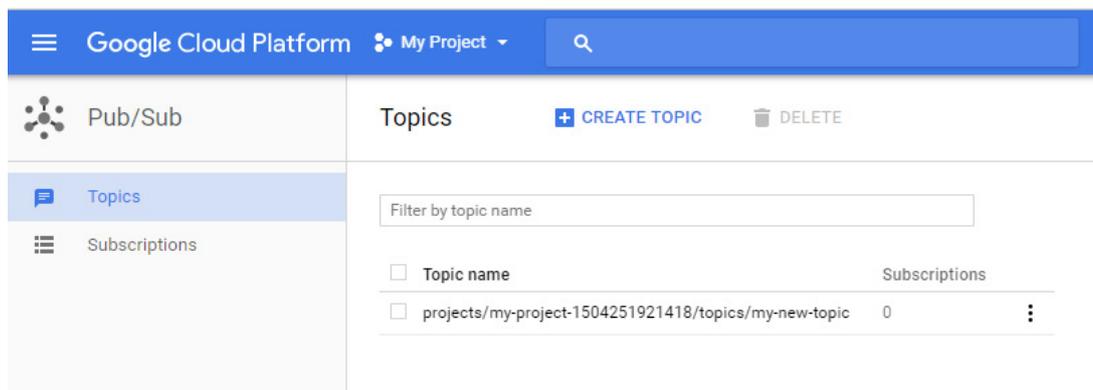


Figure 4

Once you successfully create a new topic (**my-new-topic**, in this example), you will be able to publish and subscribe messages to and from that topic.

Before using the Google Cloud Pub/Sub Python samples, you need to follow the instructions available at the following link to set up an authentication process:

<https://github.com/GoogleCloudPlatform/python-docs-samples/tree/master/pubsub/cloud-client>

In this document, we use Google Cloud SDK and ThingsPro's default credentials to set up an authentication process to publish messages and subscribe to messages.

For details, visit:

<https://github.com/GoogleCloudPlatform/python-docs-samples/tree/master/pubsub/cloud-client>

Now, run the following command:

```
# gcloud auth application-default login
```

A secure link is displayed.

```
root@Moxa:/home/moxa# gcloud auth application-default login
Go to the following link in your browser:

https://accounts.google.com/o/oauth2/auth?redirect_uri=urn%3Aietf%3Aawg%3Aoauth%3A2.0%3Aaob&prompt=select_account&response_type=code&client_id=764086051850-6qr4p6gpi6hn506pt8ejuq83di341hur.apps.googleusercontent.com&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform&access_type=offline
```

Access the link to view the authorization code. Copy the authorization code and paste it in the console.

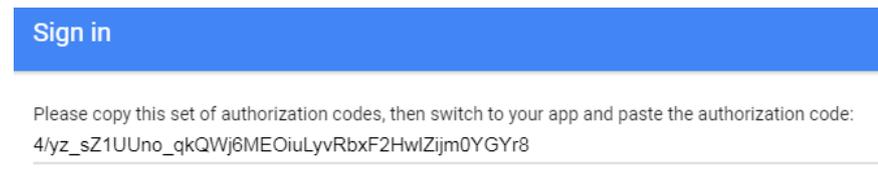


Figure 5

After you have entered the verification code in the console, the credentials will be generated and stored in the ThingsPro Gateway for Pub/Sub samples as shown in Figure 6.

```
root@moxa:/home/moxa# gcloud auth application-default login
Go to the following link in your browser:

  https://accounts.google.com/o/oauth2/auth?redirect_uri=urn%3Aietf%3Aawg%3Aoauth%3A2.0%3Aaob&prompt=select_account&response_type=code&client_id=764086051850-6qr4p6gpi6hn506pt8ejuq83di341hur.apps.googleusercontent.com&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform&access_type=offline

Enter verification code: 4/yz_sZ1UUno_qkQWj6MEOiuLyvRbxF2HwlZijm0YGYr8

Credentials saved to file: [/root/.config/gcloud/application_default_credentials.json]

These credentials will be used by any library that requests
Application Default Credentials.
root@moxa:/home/moxa#
```

Figure 6

Install the required Python packages in the virtual environment using the following commands:

```
# cd /home/moxa/Google/python-docs-samples/pubsub/cloud-client
# virtualenv env
# source env/bin/activate
(env) # pip install -r requirements.txt
```

```
(env) root@moxa:/home/moxa/Google/python-docs-samples/pubsub/cloud-client# pip install -r requirements.txt
Collecting google-cloud-pubsub==0.28.2 (from -r requirements.txt (line 1))
  Downloading google_cloud_pubsub-0.28.2-py2.py3-none-any.whl (80kB)
    100% |#####| 81kB 651kB/s
Collecting googleapis-common-protos[grpc]<2.0dev,>=1.5.2 (from google-cloud-pubsub==0.28.2->-r requirements.txt (line 1))
  Downloading googleapis-common-protos-1.5.2.tar.gz
Collecting google-gax<0.16dev,>=0.15.13 (from google-cloud-pubsub==0.28.2->-r requirements.txt (line 1))
  Downloading google-gax-0.15.14.tar.gz (109kB)
    100% |#####| 112kB 381kB/s
Collecting psutil<6.0dev,>=5.2.2 (from google-cloud-pubsub==0.28.2->-r requirements.txt (line 1))
  Downloading psutil-5.2.2.tar.gz (348kB)
    100% |#####| 358kB 332kB/s
Collecting google-cloud-core<0.28dev,>=0.27.0 (from google-cloud-pubsub==0.28.2->-r requirements.txt (line 1))
  Downloading google_cloud_core-0.27.1-py2.py3-none-any.whl (50kB)
    100% |#####| 51kB 857kB/s
Collecting grpcio<2.0dev,>=1.0.2 (from google-cloud-pubsub==0.28.2->-r requirements.txt (line 1))
```

To subscribe to the message from ThingsPro Gateway, you need to subscribe to the topic that you just created using the command:

```
(env) # python ./subscriber.py my-project-1504251921418 create my-new-topic my-sub
```

```
(env) root@Moxa:/home/moxa/Google/python-docs-samples/pubsub/cloud-client# python ./subscriber.py my-project-1504251921418 create my-new-topic my-sub
Subscription created: name: "projects/my-project-1504251921418/subscriptions/my-sub"
topic: "projects/my-project-1504251921418/topics/my-new-topic"
push_config {
}
ack_deadline_seconds: 10
(env) root@Moxa:/home/moxa/Google/python-docs-samples/pubsub/cloud-client#
```

Now you are ready to receive messages through your subscription ("my-sub") from the topic "my-new-topic".

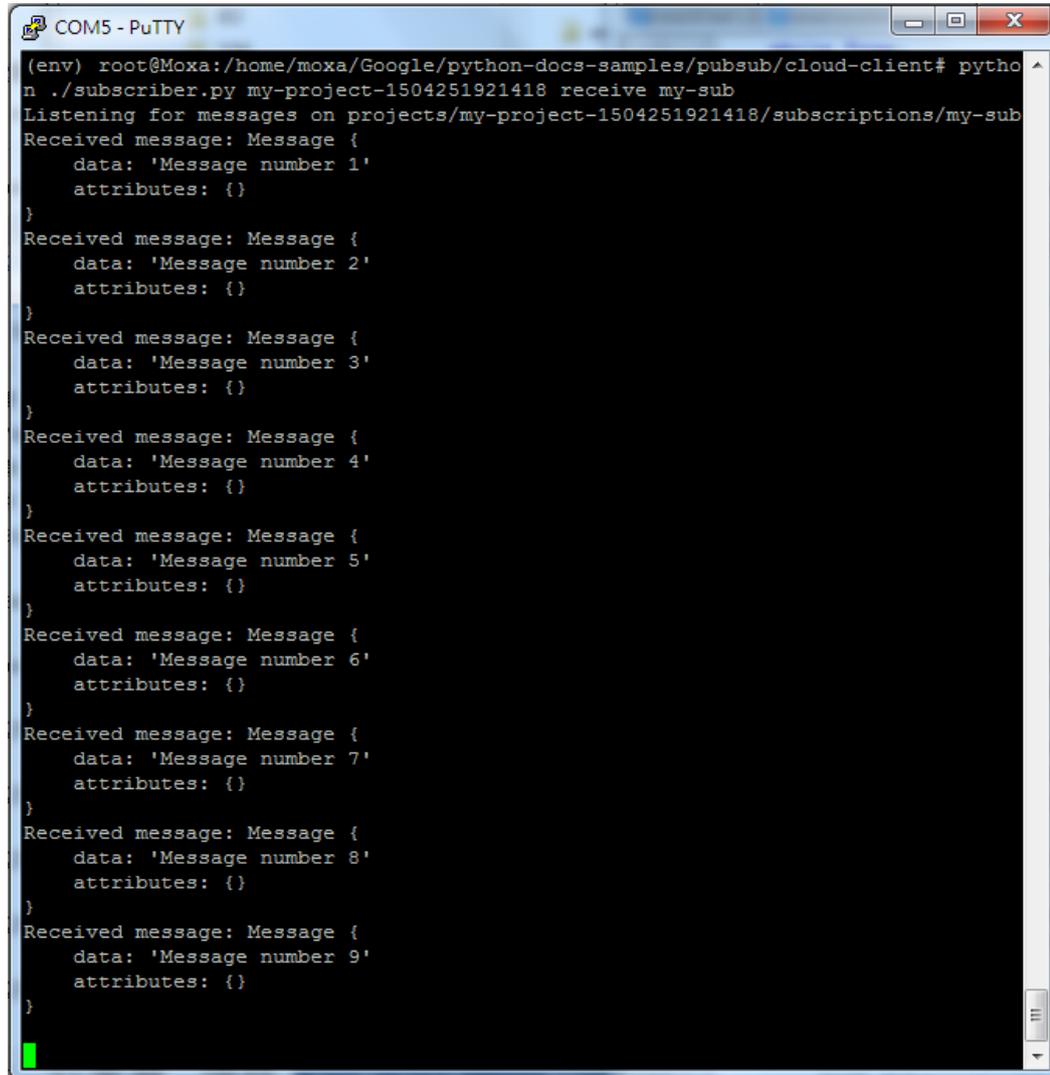
Use the following command to publish messages to the "my-new-topic" topic.

```
(env) # python ./publisher.py my-project-1504251921418 publish my-new-topic
```

```
(env) root@Moxa:/home/moxa/Google/python-docs-samples/pubsub/cloud-client# python ./publisher.py my-project-1504251921418 publish my-new-topic
Published messages.
(env) root@Moxa:/home/moxa/Google/python-docs-samples/pubsub/cloud-client#
(env) root@Moxa:/home/moxa/Google/python-docs-samples/pubsub/cloud-client#
```

To read the messages that you just published to the topic, use the command:

```
(env) # python ./subscriber.py my-project-1504251921418 receive my-sub
```



```
COM5 - PuTTY
(env) root@Moxa:/home/moxa/Google/python-docs-samples/pubsub/cloud-client# python
n ./subscriber.py my-project-1504251921418 receive my-sub
Listening for messages on projects/my-project-1504251921418/subscriptions/my-sub
Received message: Message {
  data: 'Message number 1'
  attributes: {}
}
Received message: Message {
  data: 'Message number 2'
  attributes: {}
}
Received message: Message {
  data: 'Message number 3'
  attributes: {}
}
Received message: Message {
  data: 'Message number 4'
  attributes: {}
}
Received message: Message {
  data: 'Message number 5'
  attributes: {}
}
Received message: Message {
  data: 'Message number 6'
  attributes: {}
}
Received message: Message {
  data: 'Message number 7'
  attributes: {}
}
Received message: Message {
  data: 'Message number 8'
  attributes: {}
}
Received message: Message {
  data: 'Message number 9'
  attributes: {}
}

```

4.4 Publish and Subscribe ThingsPro Modbus data to/from the Google Cloud Platform Pub/Sub service

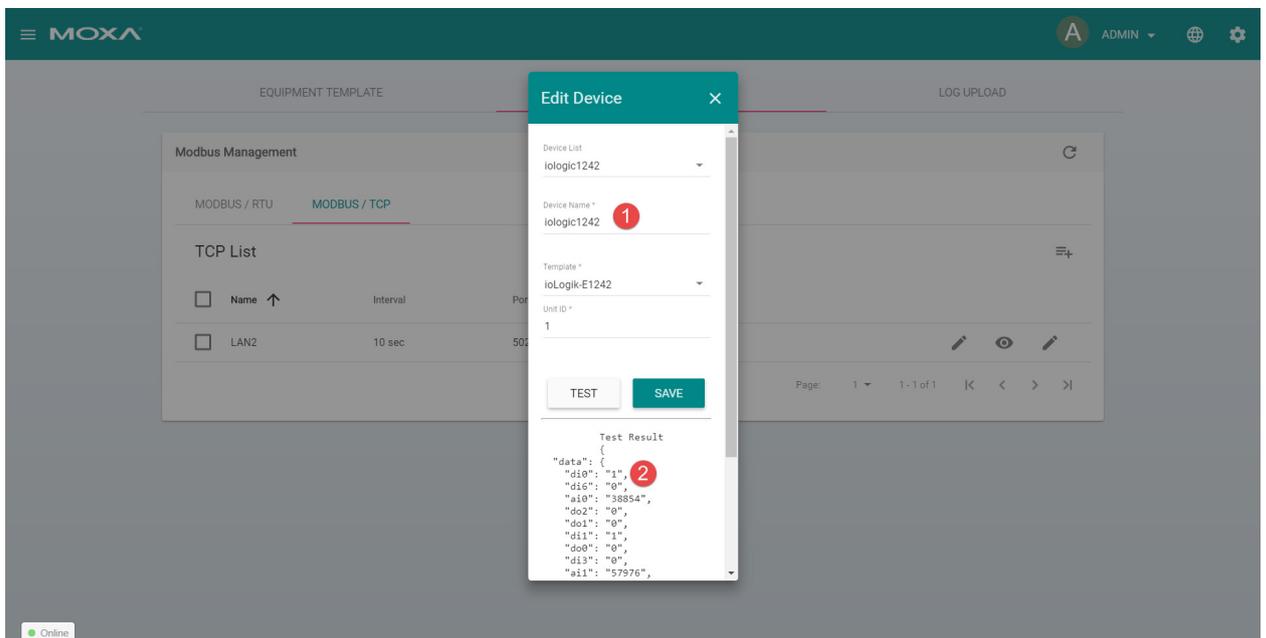
In this section, we use the ThingsPro example code, which is already available on the platform, to read Modbus data queried by the ThingsPro Modbus engine that we have set up in the previous section.

Getting Started with the Google Cloud Platform SDK on ThingsPro 2.0

1. Copy the ThingsPro Modbus example code ("subscribe.py") to the current directory and rename it to "thingspro.py".

```
jimmy@jimmy-desktop: ~  
(env) root@moxa:/home/moxa/Google/python-docs-samples/pubsub/cloud-client# cp /usr/share/mxdaf/libmxdaf_py/sample/subscribe.py thingspro.py  
(env) root@moxa:/home/moxa/Google/python-docs-samples/pubsub/cloud-client#
```

2. Change the Modbus query example code based on your ThingsPro Modbus TCP web console setup.
For the example in this document, used the Equipment name (1: iologic1214) and tag name (2: di0) values and edit the `thingspro.py` program in the web console as shown below:



```
jimmy@jimmy-desktop: ~  
import time  
from libmxidaf_py import TagV2  
  
tagv2 = TagV2.instance()  
  
def on_tag_callback(equipment_name, tag_name, tag):  
    print "{}: {}: {}: {}: {}".format(  
        equipment_name,  
        tag_name,  
        tag.at(),  
        tag.value(),  
        tag.unit())  
  
tagv2.subscribe_callback(on_tag_callback)  
  
tagv2.subscribe(  
    "iologic1242", 1  
    "di0" 2  
  
while(True):  
    time.sleep(0.1)  
~  
~  
~
```

3. Run the thingspro.py program.

The Modbus engine will poll data at the intervals specified.

```
(env) root@Moxa:/home/moxa/Google/python-docs-samples/pubsub/cloud-client# python thingspro.py  
iologic1242:di0:2017-09-18T10:03:44.008Z:1:  
iologic1242:di0:2017-09-18T10:03:46.003Z:1:
```

4. Now merge Google publisher.py example code with the thingspro.py program as shown below:

```

import argparse ①
import time
from libxidaf_py import TagV2
from google.cloud import pubsub_v1 ②

tagv2 = TagV2.instance()

def publish_messages(project, topic_name, data): ③
    """Publishes multiple messages to a Pub/Sub topic."""
    publisher = pubsub_v1.PublisherClient()
    topic_path = publisher.topic_path(project, topic_name)

    # for n in range(1, 10):
    #     data = u'Message number {}'.format(n)
    #     # Data must be a bytestring
    #     data = data.encode('utf-8')
    #     publisher.publish(topic_path, data=data)

    print('Published messages.')

def on_tag_callback (equipment_name, tag_name, tag):
    print "{}:({}):({}):({})".format(
        equipment_name,
        tag_name,
        tag.at(),
        tag.value(),
        tag.unit())

④ publish_messages("my-project-1504251921418", "my-new-topic", "tag: {}, timestamp: {}, value: {}".format(tag_name, tag.at(), tag.value()))

tagv2.subscribe_callback(on_tag_callback)

tagv2.subscribe(
    "iologic1242",
    "di0")

while(True):
    time.sleep(0.1)

```

5. Run the `thingspro.py` program.

Each time the Modbus engine reads data, it publishes it to the Google Pub/Sub service (3 above). The following image shows the data payloads being sent.

```

(env) root@Moxa:/home/moxa/Google/python-docs-samples/pubsub/cloud-client# python thingspro.py
iologic1242:di0:2017-09-19T02:46:06.007Z:1:
Published messages.
iologic1242:di0:2017-09-19T02:46:08.000Z:1:
Published messages.
iologic1242:di0:2017-09-19T02:46:10.007Z:1:
Published messages.

```

6. To read the data from the Google Pub/Sub service, run the `subscriber.py` program.

```

(env) root@Moxa:/home/moxa/Google/python-docs-samples/pubsub/cloud-client# python ./subscriber.py my-project-1504251921418 receive my-sub
Listening for messages on projects/my-project-1504251921418/subscriptions/my-sub
Received message: Message {
  data: 'tag: di0, timestamp: 2017-09-19T02:46:10.007Z, val...'
  attributes: {}
}
Received message: Message {
  data: 'tag: di0, timestamp: 2017-09-19T02:46:08.000Z, val...'
  attributes: {}
}
Received message: Message {
  data: 'tag: di0, timestamp: 2017-09-19T02:46:06.007Z, val...'
  attributes: {}
}

```

The `subscriber.py` program is used to confirm that your data has been published to the Google Pub/Sub service. Once your data is available to the Pub/Sub service, it can be used by Google Cloud Data Analytics Services through the Data Flow service.

5 Additional Reading

<https://console.cloud.google.com>

https://cloud.google.com/solutions/iot-overview#pipeline_processing_tasks

<https://cloud.google.com/pubsub/docs/quickstart-client-libraries>

[https://github.com/GoogleCloudPlatform/python-docs-samples/tree/master/pubsub/
cloud-client](https://github.com/GoogleCloudPlatform/python-docs-samples/tree/master/pubsub/cloud-client)

<http://www.moxa.com/product/uc-8100.htm>