How to connect an IoT-device to Google Cloud Platform using WireQueue MQTT Toolkit

This guide explains how you can use WireFlow's MQTT Toolkit to enable your IoT-devices to communicate with/via "Google Cloud Platform" (GCP) using the MQTT protocol.

It is important to note that GCP is not a standard MQTT-broker, but rather a huge internet-based infrastructure-application that supports MQTT as a means of communication for connected devices. This means, for example, that it is not possible to just switch from a standard MQTT-broker to GCP, just as easily as it would be to switch to another standard broker. More on this in section *"PART II: Understanding Google Cloud Platform (GCP) as an MQTT broker"* on page 7.

The document assumes you already have created a GCP-account and that you have full administrator access to it.

The instructions are "kind of" following the Quickstart-guide that can be found here:

https://cloud.google.com/iot/docs/quickstart/

...but with more details and informative screenshots.

Note: *Details, urls, methods, page layouts etc. may change on GCP after this paper was written (February of 2021).*

There is also a LabVIEW program available for download that you can use to explore how to program your device with the MQTT Toolkit to connect it with GCP. The program is set up using the same resource IDs that is used in this document, so if there is no specific reason not to, it might be a good idea to use the same naming when you set up your devices in GCP. But it is not necessary.

See *"PART III: An example-program"* on page 9 for more information about the example-program.

WireFlow AB

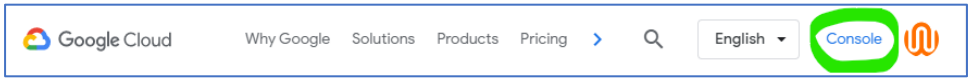
Theres Svenssons gata 10
SE-417 55 Göteborg
Sweden

www.wireflow.se

Application note no. 20
AB0005-128 rev A

PART I: Set up a device on Google Cloud Platform

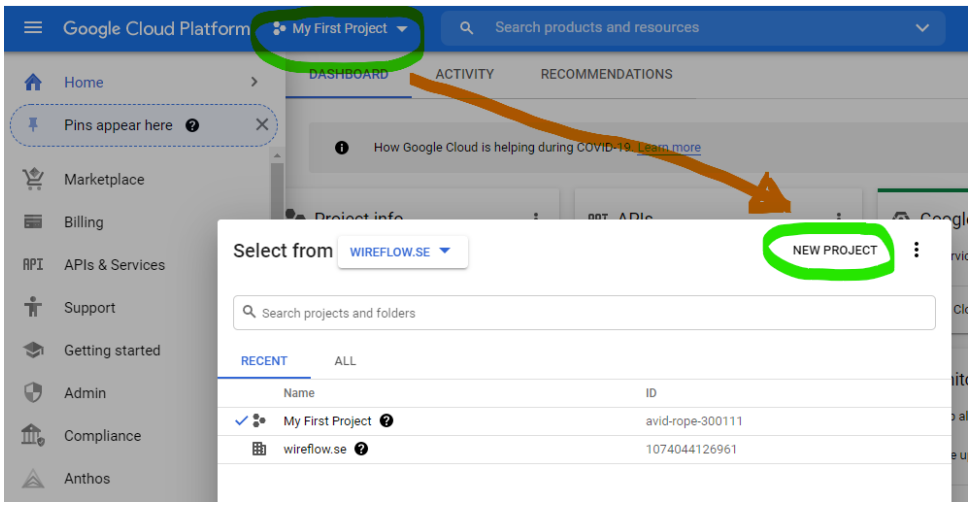
1) **Log in** to the GCP-account here: https://cloud.google.com/
2) **Go to Console**: Press “Console”



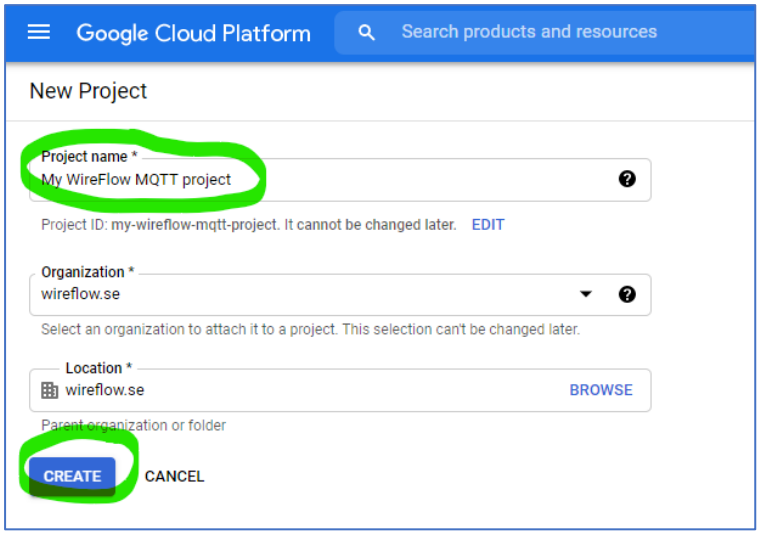
3) **Create a project**:

   a) Press the Project selector

   b) In the window that appears, press “NEW PROJECT”



   c) In the new window that appears, enter a Project name and press create:



WireFlow AB

Theres Svenssons gata 10
SE-417 55 Göteborg
Sweden

www.wireflow.se

Application note no. 20
AB0005-128 rev A

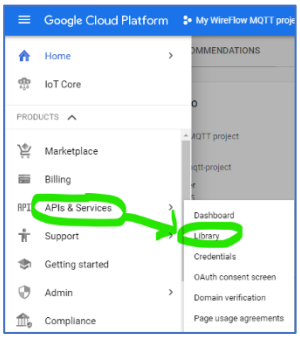
4) Enable APIs

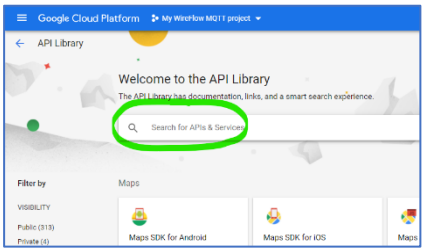
Make sure that these APIs are enabled:

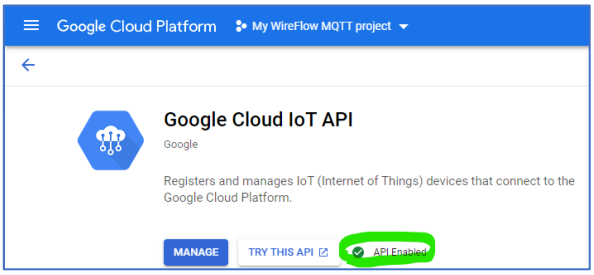
- "Google Cloud IoT API"
- "Cloud Pub/Sub API"

To do this:

a) Find "API's & Services" in the navigation menu to the right and select "Library" in the sub-menu.
b) Here you can search between all API's.
c) Search for the API's, open them and make sure that they have been enabled.





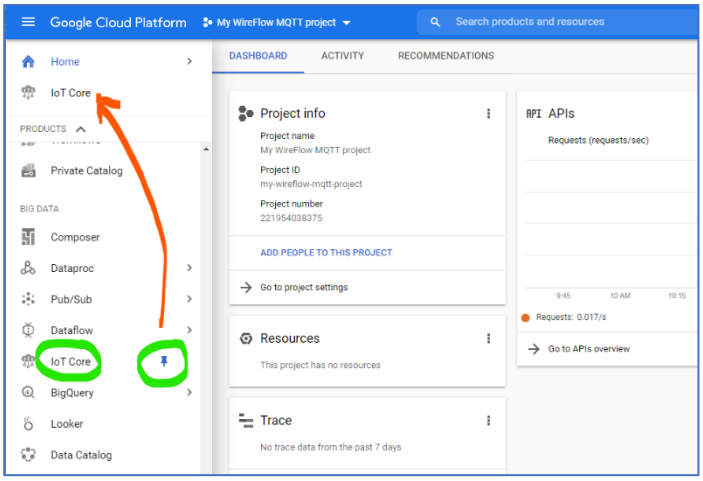


5) Set up a "Device Registry".

a) Go to "IoT Core"-page

Press the "IoT Core"-entry far down in the left-side menu-panel.

Tip: To get quick access to a menu-item in the long list, you can pin it as shown in the picture. This will make them appear at the top of the list.



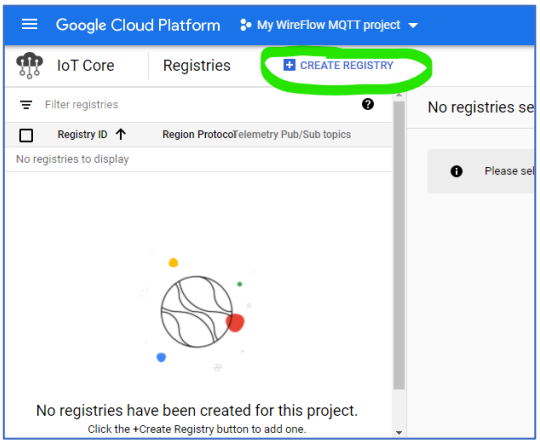
WireFlow AB

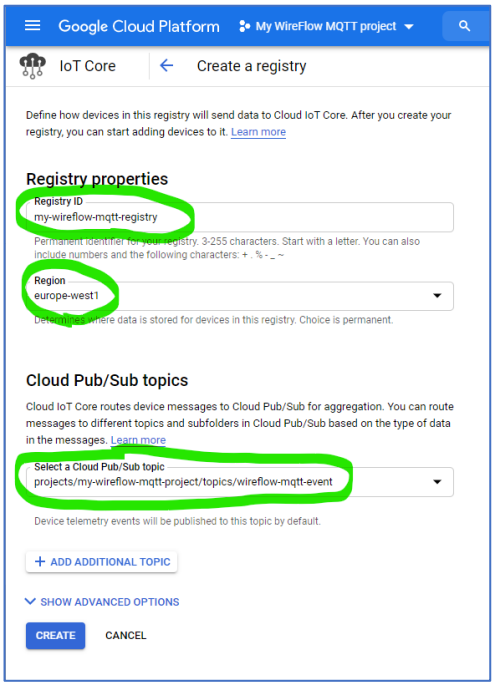
Theres Svenssons gata 10
SE-417 55 Göteborg
Sweden

www.wireflow.se

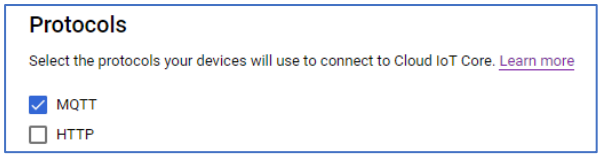
Application note no. 20
AB0005-128 rev A

6) Create a “Device Registry”

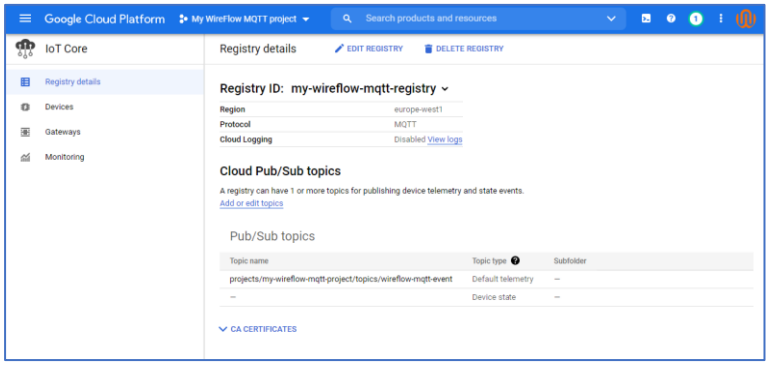




a) Name it (in our example: *“my-wireflow-mqtt-registry”*)
b) Select a region close to you (*“europe-west1”*)
c) Create a topic (*"wireflow-mqtt-event”*)
d) The Device state topic and Certificate value fields are optional, so leave them blank.
e) Press *“SHOW ADVANCED OPTIONS”*
f) Under *“Protocols”* make sure that MQTT is enabled and that HTTP is disabled



g) Press *“CREATE”* to complete the “Device Registry” creation.
h) You have now created a “Device Registry” with these properties:

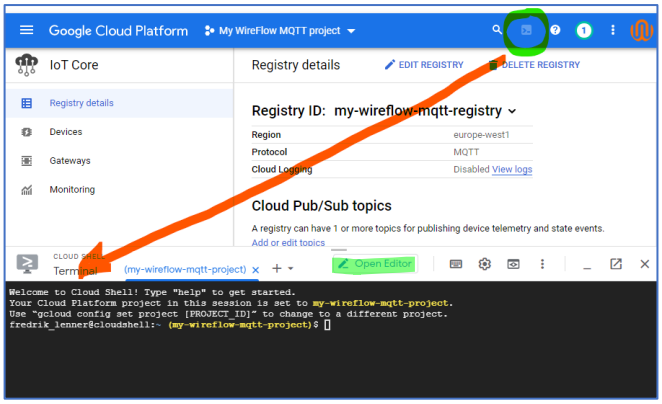


7) Generate a device key pair.
   a) Activate the “Cloud Shell” by pressing the button to the right in the upper menu bar This will show a console window at the bottom of the page.

WireFlow AB

Theres Svenssons gata 10
SE-417 55 Göteborg
Sweden

www.wireflow.se

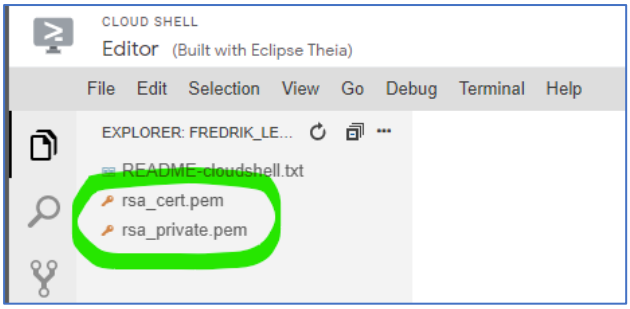


b) Create key

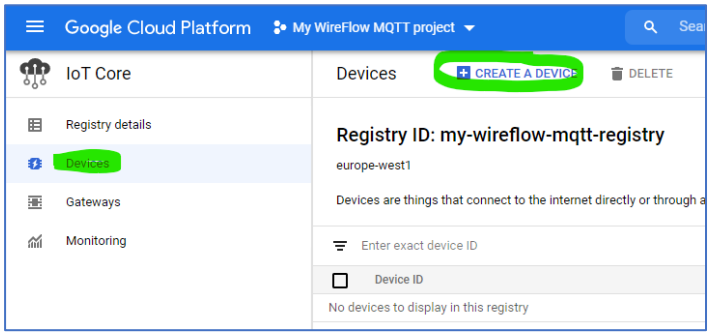
In the shell, write this multi-line commend to create a RS256 key:

```
openssl req -x509 -newkey rsa:2048 -keyout rsa_private.pem -nodes \
    -out rsa_cert.pem -subj "/CN=unused"
```

To verify, you can open the "Editor", see picture in point a) above.
There you should now see the two keys like in the picture below.



8) Add a device to the "Device Registry".
   a) On the **Registries** page, select "*my-wireflow-mqtt-registry*".
   b) Select the Devices tab and click Create a device.

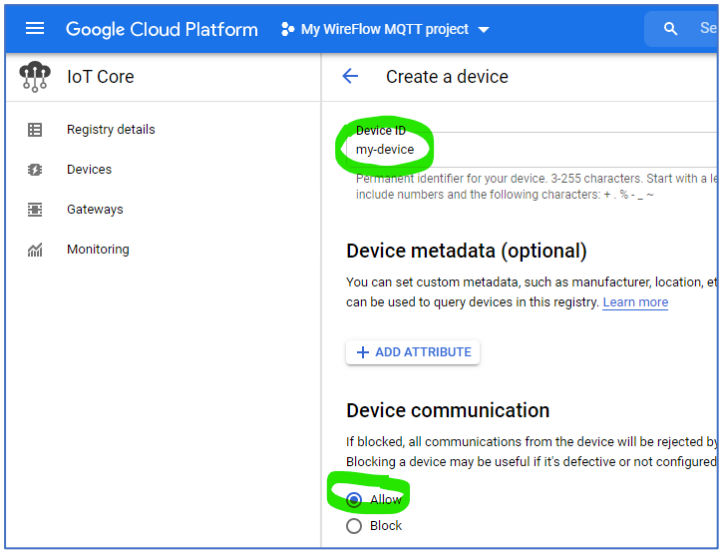


   c) Enter (for example) "*my-device*" for the Device ID.
   d) Unfold the *"COMMUNICATION, CLOUD LOGGIN …"*-section
   e) Select Allow for Device communication.

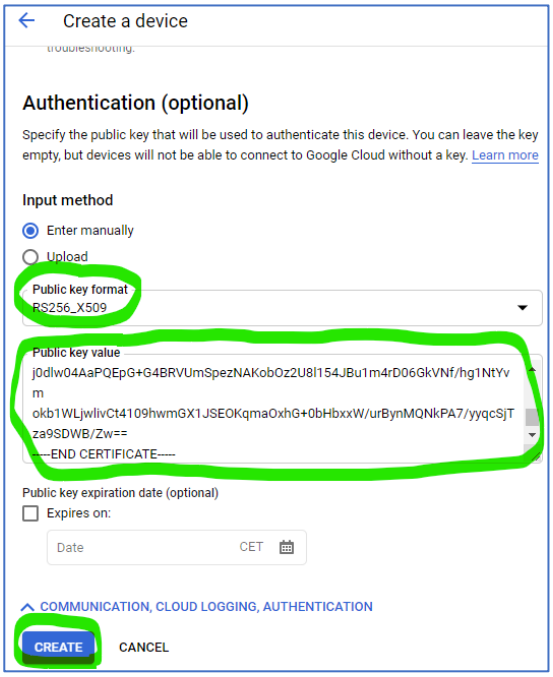
WireFlow AB

Theres Svenssons gata 10
SE-417 55 Göteborg
Sweden

www.wireflow.se



f) Add the public key generated earlier to the Authentication fields.
   - Copy the contents of rsa_cert.pem to the clipboard. Make sure to include the lines that say *-----BEGIN CERTIFICATE-----* and *-----END CERTIFICATE-----*.
   - Select RS256_X509 for the Public key format.
   - Paste the public key in the Public key value box.



g) The Device metadata field is optional; leave it blank.

h) Click Create.

i) You have just added a device to your registry. The RS256_X509 key appears on the Device details page for your device.

We have now set up a device on the Google IoT Core that is ready for interaction with WireFlows MQTT Toolkit!

WireFlow AB

Theres Svenssons gata 10
SE-417 55 Göteborg
Sweden

www.wireflow.se

PART II: Understanding Google Cloud Platform (GCP) as an MQTT broker

GCP is not a vanilla MQTT broker, but rather a cloud-based server application that can be set up to utilize the MQTT protocol for interaction with remote clients/devices.

GCP's implementation restricts some of the aspects of MQTT and enforces some rules with regards to specific topics and how they may be used. More about this below.

About topics

When it comes to topics, GCP has specified four mandatory topics that should be used for different categories of information. GCP also specifies the directionality of each topic.

The table below lists those mandatory topics.

Category	Direction	Topic-path	Description
Telemetry data	Device => Cloud	/devices/DEVICE_ID/events	All event-data (for example, measurements) sent from devices to the cloud. Telemetry data sent from a device to the cloud is called "device telemetry event"-data. It is possible to create sub-topics under the "events"-node and publish to them.
Device state	Device => Cloud	/devices/DEVICE_ID/state	An arbitrary, user-defined blob of data that describes the status of the device.
Device configutration	Cloud => Device	/devices/DEVICE_ID/config	An arbitrary, user-defined blob of data used to modify a device's settings.
Command	Cloud => Device	/devices/DEVICE_ID/commands/#	Commands for controlling the behaviour of a device

Note that each "Category"-entry in the list is a clickable link to the GCP-documentation on the Internet.

If you, for example, want an IoT-device to publish some kind of sensory data to the GCP, this must be done to a sub-topic in the "events"-topic since it falls in the "Telemetry-data"-category. If you want to publish several different types of sensor-values from a device, let's say temperature and air pressure, it is possible to create a sub-topic for each of these parameters in the "events"-topic. The same is true for each of the mandatory category-topics.

WireFlow AB

Theres Svenssons gata 10
SE-417 55 Göteborg
Sweden

www.wireflow.se

About device authentication

When an IoT-device connects to GCP, it is required to supply a so-called "JSON Web Token", or "JWT". "JWT" is a standard for safe authentication over internet. The validity of a JWT in GCP is limited to last for max 24 hours, before which it must be renewed. If not renewed, the connection to GCP will break down. It is possible to refresh the token without disconnecting first, this is described in the link at the end of this section.

If the connection is lost, a new JWT must be generated and used for the re-login. This means that your LabVIEW code must be monitoring the validity state of the registered token to detect when a refresh is required or if a new JWT must be generated.

In GCP's documentation there is a code example showing how to generate JWT's using different languages. WireFlow has created a VI that calls Python code to do this. The JWT-generator VI is part of the example-code that is supplied with this white paper. Note that it requires Python to be installed as well as a package called "PyJWT".

GCP's use of JWT's is described here.

Some useful connection settings

Setting	Value	Comment
Server address	mqtt.googleapis.com	
Port	8883 or 443	
Client ID	projects/*PROJECT_ID*/locations/*REGION*/registries/*REGISTRY_ID*/devices/*DEVICE_ID*	"Blue values" are to be replaced with values suitable for your application
Username	A non-empty, irrelevant string.	We use "IGNORE" to signal that it is irrelevant.
Password	A JWT	There are detailed descriptions about how to generate JWTs on the GCP home page.

WireFlow AB

Theres Svenssons gata 10
SE-417 55 Göteborg
Sweden

www.wireflow.se

PART III: An example-program

WireFlow has developed an example-program that shows how our MQTT toolkit can be used in your LabVIEW application to communicate with the Google Cloud Platform (GCP). It can be downloaded from our website and is called *"AB0005-129 AN20 Google IoT How-To_ExampleProgram.zip"*.

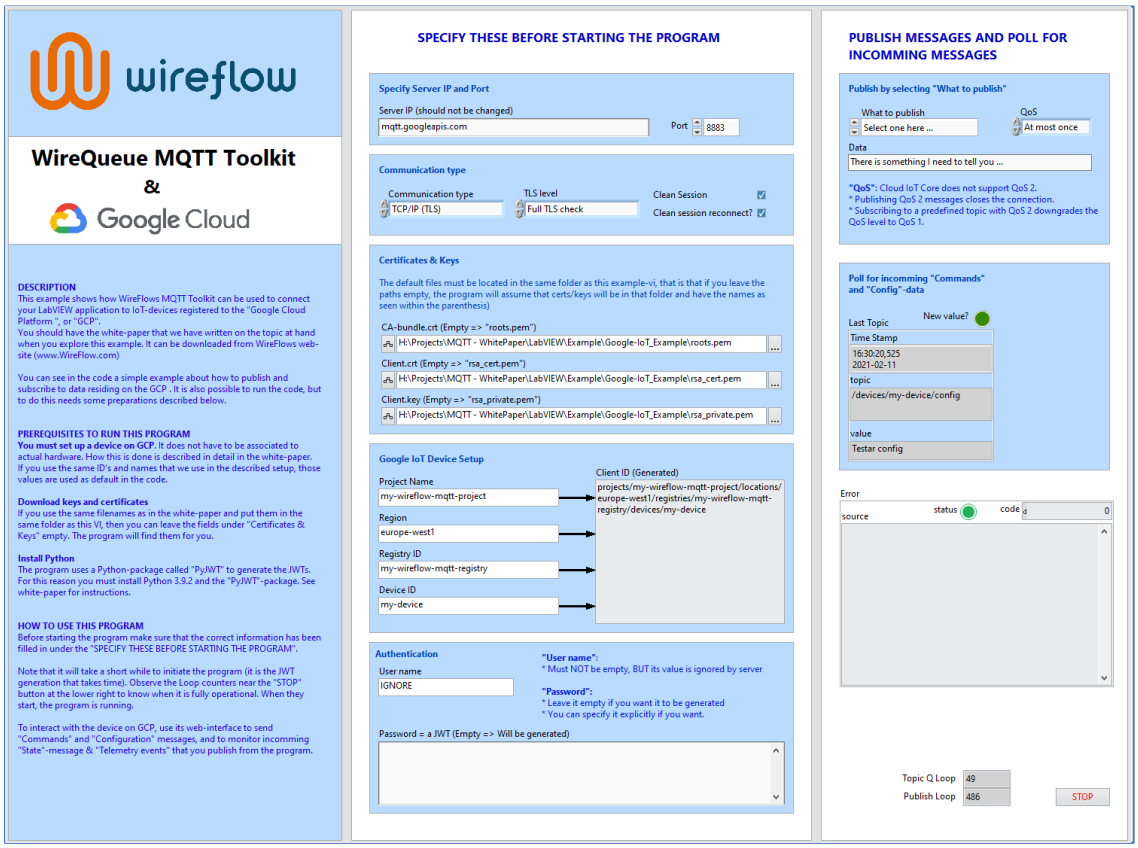
The program has been created for a temporary GCP project and requires you to set up your own project if you want to connect to GCP and try out the functionality. But even if you do not set up an account, you can look at the code and get a grip on how to do things.

The following requirements must be met to <u>run</u> the example:

- LabVIEW 2020 installed.
- VIPM-package "WF WireQueue-MQTT" installed with an activated license.
- Python 3.9.1 installed with the same bitness as your LabVIEW-installation.
- Python package "PyJWT" installed with the "Cryptographic Dependencies"-option
  Use this command: pip install pyjwt[crypto]
- A device configured in GCP.
- Generated and downloaded key-pair put in the same directory as the LabVIEW-example code.

The Program has been configured to work with the GCP-configuration described in this document, so if you use the same ID's and names, the program will work right away if the prerequisites are fulfilled.

Below is a screenshot of the programs front panel.



WireFlow AB

Theres Svenssons gata 10
SE-417 55 Göteborg
Sweden

www.wireflow.se

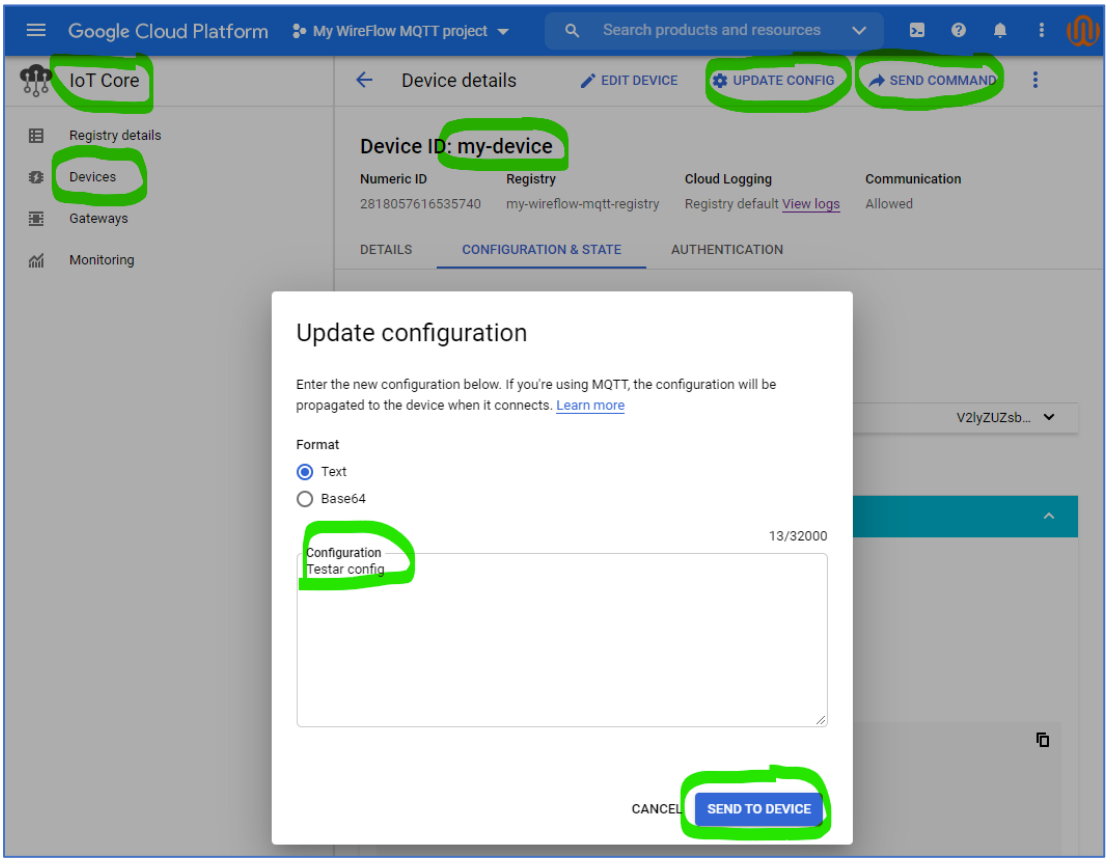
The program has instructions on its front panel and in the diagram, so it will not be described in too much detail here. Some things are worth mentioning though.

- The program connects to the specified GCP-project with the specified configuration. To change the settings the program must be stopped, re-configured and re-started. Use the “STOP”-button in the lower right corner to stop the program.
- The program will operate as an IoT-device publishing its data (“Telemetry-events”) to GCP.
- It polls for incoming “Commands” and “Configuration-events”.
- The user can publish “Telemetry--events” and “State-events” to GCP.
- To confirm that the data is actually exchanged between the program and your Google IoT-device, you can use the GCP console.
- **Note** that the program does not have any connection status monitoring, and if the connection is lost, or if the JWT gets outdated, you must restart the program.

To send a “configuration”- or “command”-message from the console

Go to:

“IoT Core” => Select the right registry => “Devices” => “my-device” => Press “UPDATE CONFIG” => “Update configuration”-wind will show => Enter data => Press “SEND TO DEVICE”.



If everything worked, the message you wrote should appear in the program in the “Poll for incoming …”-area.

To send a Command do the same, but instead press the “SEND COMMAND”-button in the console.

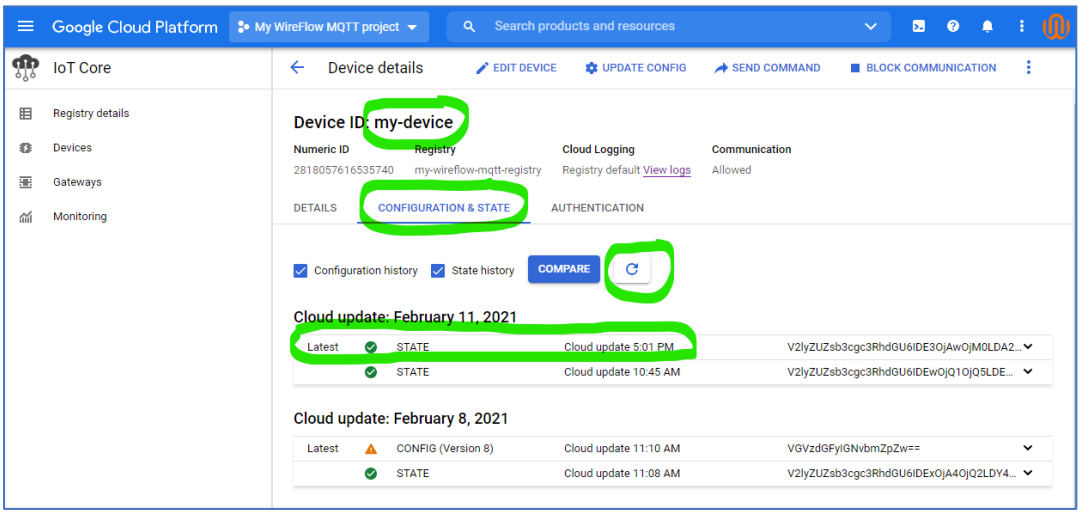
WireFlow AB

Theres Svenssons gata 10
SE-417 55 Göteborg
Sweden

www.wireflow.se

To verify that published “State”-messages reaches GCP

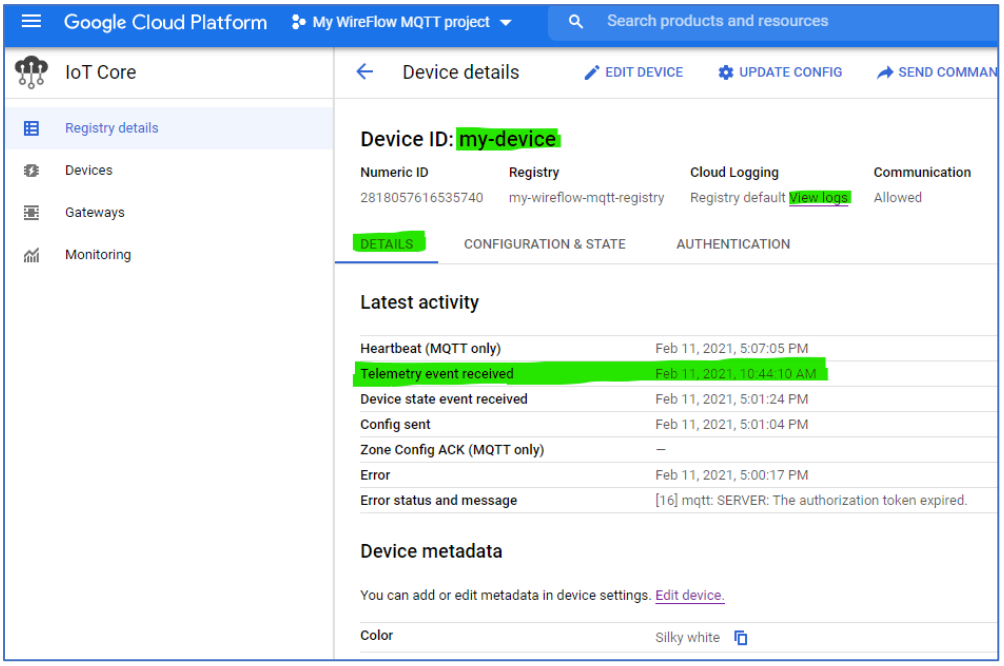
1. Go to “my-device”
2. Select the “CONFIGURATION & STATE”-tab.
3. If needed, press the refresh-button
4. Check that there is a new “STATE”-message in the list



To verify that published “Telemetry-events” reaches GCP

Just check the latest message

1. Go to “my-device”
2. Select the “DETAILS”-tab.
3. Check that the latest “Telemetry event received” have a correct timestamp.



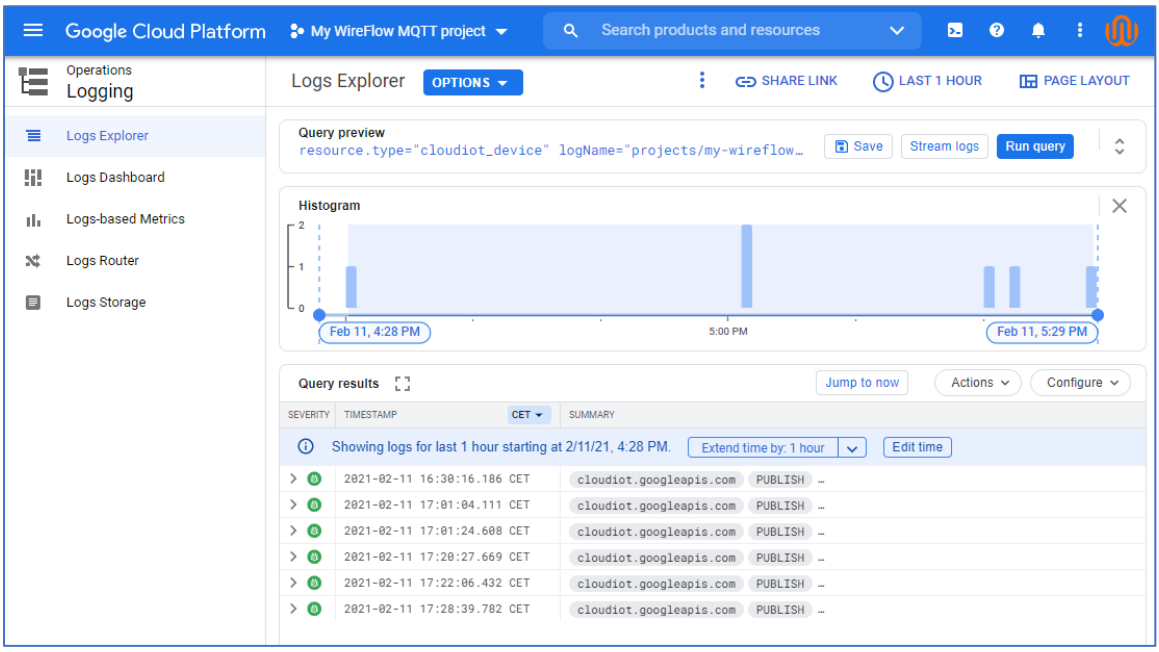
WireFlow AB

Theres Svenssons gata 10
SE-417 55 Göteborg
Sweden

www.wireflow.se

Check the log

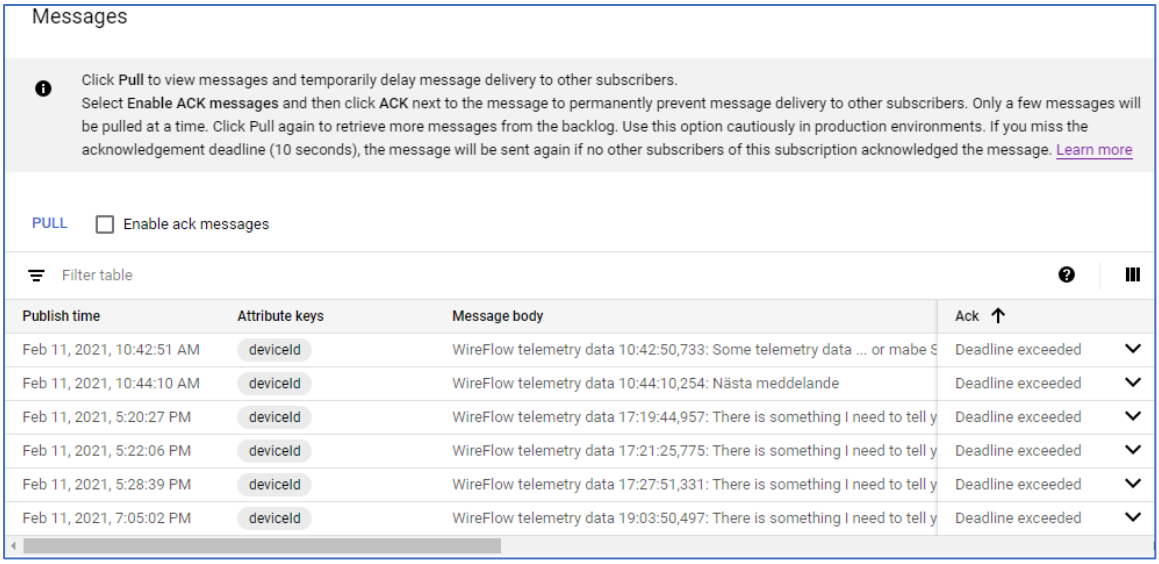
To get a more comprehensive overview of received telemetry-events you can press the "View logs"-link which will take you to the "Logs explorer", see below.



Here you can view all activities, like for example the telemetry-events. Use queries and time filters to view the activity that you are interested in.

Create a dedicated subscription

A more readable method is to create a subscription in GCP for the "wireflow-mqtt-event"-topic, let us call it "wireflow-mqtt-event-subscription". This would enable you to a clean list just containing messages from the program published to this specific topic. See below.



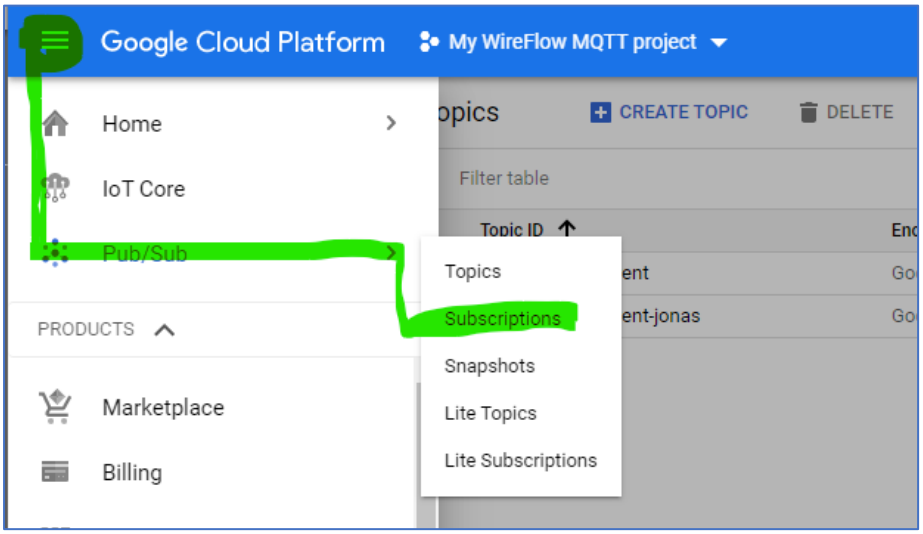
WireFlow AB

Theres Svenssons gata 10
SE-417 55 Göteborg
Sweden

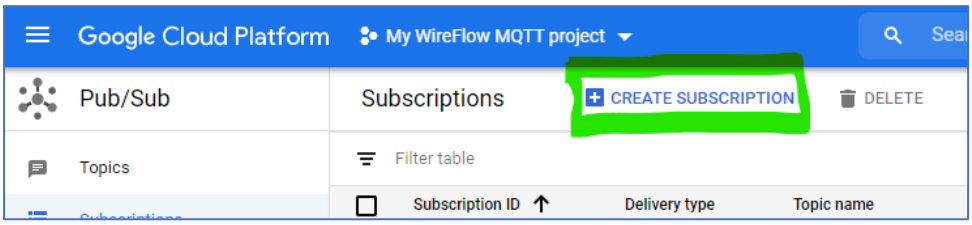
www.wireflow.se

Application note no. 20
AB0005-128 rev A

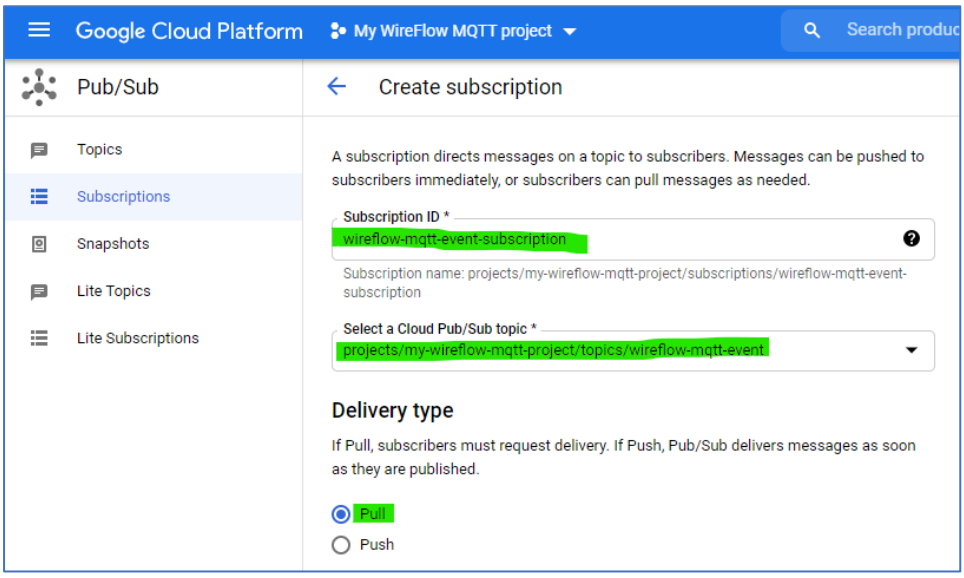
To create a subscription, go to Pub/Sub and select “Subscriptions”



Select “CREATE SUBSCRIPTION”



1. Enter Subscription ID = “wireflow-mqtt-event-subscription”
2. Select “wireflow-mqtt-event” as Pub/Sub-topic
3. Press the “CREATE” button at the bottom of the page.



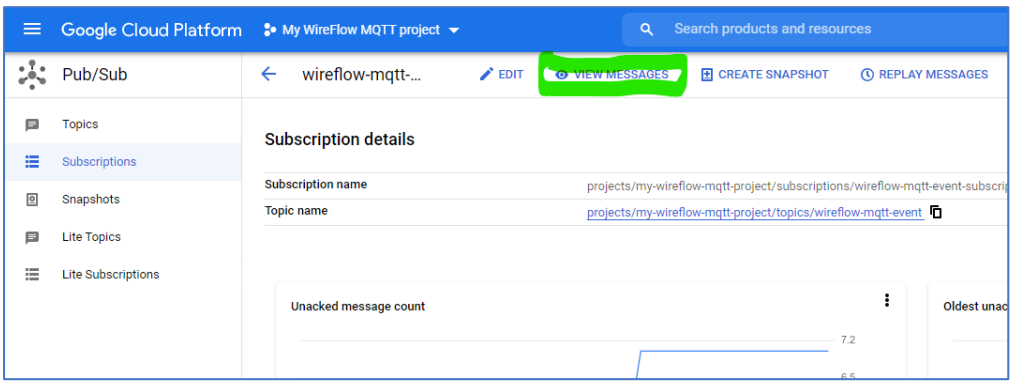
WireFlow AB

Theres Svenssons gata 10
SE-417 55 Göteborg
Sweden

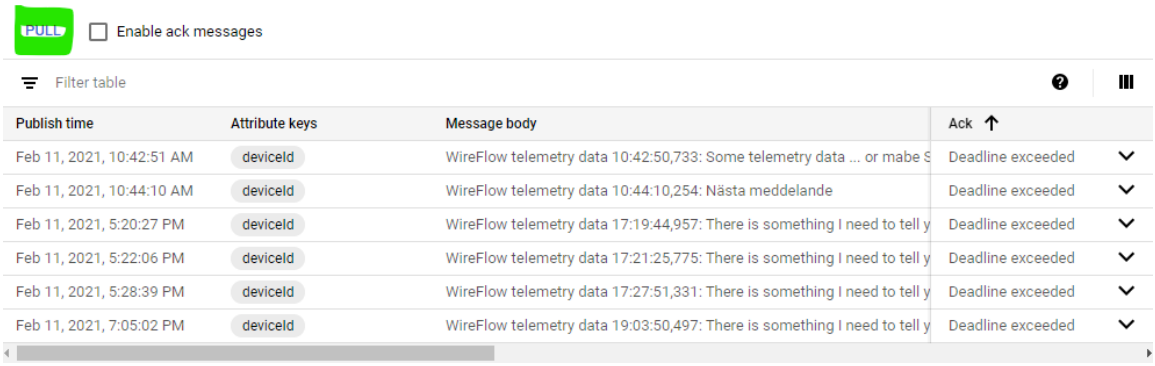
www.wireflow.se

“wireflow-mqtt-event-subscription” should now appear in the Subscriptions”-list.

Press its name and you will get to the “Subscription details”-page:



Press “VIEW MESSAGES”



Press “PULL” to obtain a list of “telemetry-event”-messages received from the program.

WireFlow AB

Theres Svenssons gata 10
SE-417 55 Göteborg
Sweden

www.wireflow.se