

VERSION 1.0
JANUARY 5, 2024

MQTT CONNECTOR PROFESSIONAL

APPLICATION NOTE 03

Publish data to Microsoft Azure IoT Hub with the URcap v. 1.8.0



4EACH S.R.O.
WWW.4EACH.CZ

APPLICATION NOTE #3

In this application note, we will go through the process of publishing robot data to the Microsoft Azure IoT hub using the MQTT Connector Professional URCap extension. We will cover the setup on the Azure side, importing the correct certificate and message sending from the robot user program.

In this tutorial, we will learn how to setup the Azure services and the robot for sending device-to-cloud messages.

NOTE

All Azure resources used in this application note are not available anymore. Resources were created temporarily only for the purpose of explaining this process.

CONTENTS

Application note #3	1
Setting up Azure resources	2
Resource group	2
IoT hub	3
Shared access policies	7
Device explorer	8
Robot installation	11
Robot program	12
Footnotes	13

SETTING UP AZURE RESOURCES

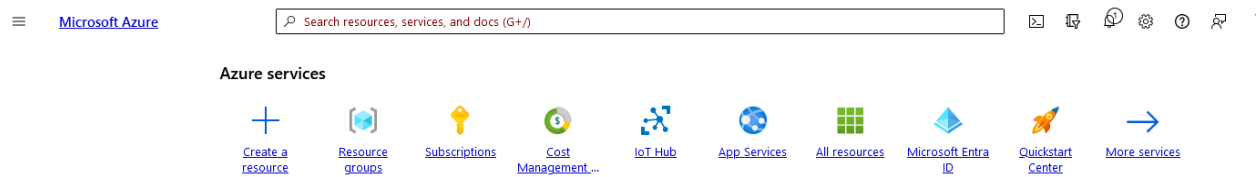
RESOURCE GROUP

First, you must have an Azure account and an active subscription. This process is out of the scope of this document. Microsoft offers some Free subscription plans. You can learn more on this website:

<https://azure.microsoft.com/en-us/free>

When your subscription is ready, we strongly recommend creating the Resource group for better resource management .

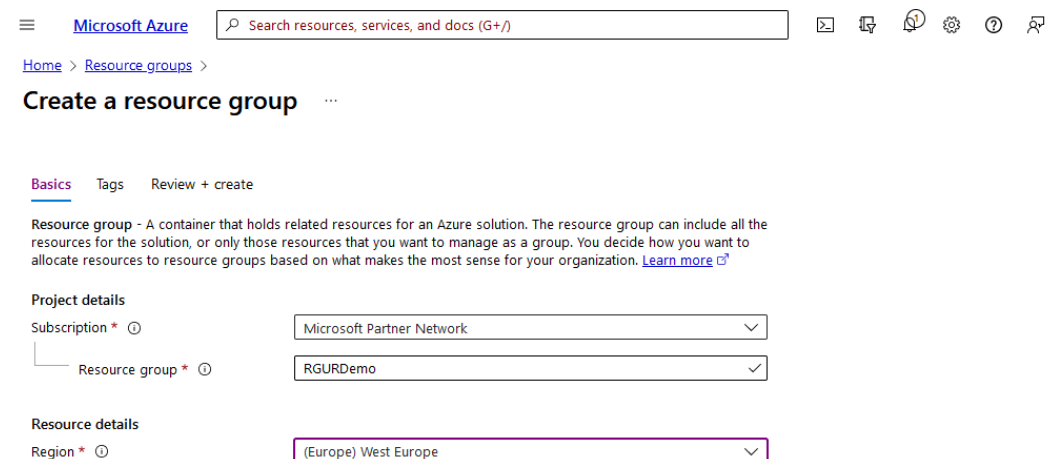
Int the <https://portal.azure.com> dashboard, please select the Resource groups option:



And create the resource group by the option Create:



You can define any name for the Resource group. While creating the Resource group you have to define the Region. Please select the Region which is close to your application. Selection of the far region can cause some issues due to higher response time during the communication.



In our application note we will use the resource group with the name **RGURDemo**:

Microsoft Azure Search resources, services, and docs (G+)

Home > Resource groups >

Create a resource group

Validation passed.

Basics Tags **Review + create**

Basics

Subscription	Microsoft Partner Network
Resource group	RGURDemo
Region	West Europe

Tags

None

IOT HUB

In the resource group RGURDemo, please select the **Create** option:

Microsoft Azure Search resources, services, and docs (G+)

Home > Resource groups >

RGURDemo Resource group

Search < + Create Manage view Delete resource group Refresh Export to CSV Open query

The Microsoft will bring you to a Marketplace with a lot of components. Please use the search function to locate the **IoT Hub** service and use the **Create** function.

Microsoft Azure Search resources, services, and docs (G+)

Home >

Marketplace

Get Started

Service Providers

Management

Private Marketplace

Private Offer Management

My Marketplace

Favorites

My solutions

Recently created

Private plans

Categories

Internet of Things (67)

Analytics (31)

AI + Machine Learning (13)

Search iot hub

Publisher name: All Product Type: All Publisher Type: All

Azure services only

Operating System: All Pricing: All

New! Get AI-generated suggestions for your search. View suggestions

Showing 1 to 20 of 93 results for 'iot hub'. Clear search

<p>IoT Hub</p> <p>Microsoft</p> <p>Azure Service</p> <p>Connect, monitor and manage IoT devices</p> <p>Create</p>	<p>IoT Hub Device Provisioning Service</p> <p>Microsoft</p> <p>Azure Service</p> <p>Seamless, zero-touch registration of devices to IoT Hub with security that begins at the device and ends with our cloud</p> <p>Create</p>	<p>Device Update for IoT Hub</p> <p>Microsoft</p> <p>Azure Service</p> <p>Securely and Reliably update your devices with Device Update for IoT Hub.</p> <p>Create</p>	<p>SIEMENS</p> <p>Insights Hub</p> <p>Siemens DI Software</p> <p>SaaS</p> <p>Insights Hub is an ir a service solution wi analytics and AI</p> <p>Subscribe</p>
--	--	--	--

During the IoT Hub creation, some parameters are important:

The screenshot shows the 'Basic' tab of the IoT Hub creation wizard in the Microsoft Azure portal. The breadcrumb navigation is 'Home > Marketplace > IoT Hub >'. The page title is 'IoT hub' with a Microsoft logo. Below the title are tabs for 'Basics', 'Networking', 'Management', 'Add-ons', 'Tags', and 'Review + create'. A message states: 'Create an IoT hub to help you connect, monitor, and manage billions of your IoT assets. [Learn more](#).' The 'Project details' section instructs to 'Choose the subscription you'll use to manage deployments and costs. Use resource groups like folders to help you organize and manage resources.' The 'Subscription' dropdown is set to 'Microsoft Partner Network' and the 'Resource group' dropdown is set to 'RGURDemo', with a 'Create new' link below. The 'Instance details' section includes: 'IoT hub name' set to 'IoTHubURDemo', 'Region' set to 'West Europe', and 'Tier' set to 'Free'. A note for the Free tier states: 'Free trial explores the app with live data. Trials cannot scale or be upgraded later.' with a 'Compare tiers' link. The 'Daily message limit' is set to '8 000 (0 €/month)'.

Please select the existing Subscription and the created Resource group.

When you define the IoT hub name, please note, the name must be **globally** unique. This name will become the hostname in the azure-devices.net domain. For our needs we have defined the IoT Hub name as **IoTHubURDemo**.

For this application note we've selected the **Free tier** of the IoT Hub. Please note, the Free tier **can't** be upgraded to the higher tier in the future, and it is allowed to have only one Free tier instance of the IoT Hub in the subscription.

For the demonstration needs of this application note we have defined the **public access** to the IoT Hub in the networking configuration:

The screenshot shows the 'Networking' tab of the IoT Hub creation wizard. The breadcrumb navigation is 'Home > Marketplace > IoT Hub >'. The page title is 'IoT hub' with a Microsoft logo. Below the title are tabs for 'Basics', 'Networking', 'Management', 'Add-ons', 'Tags', and 'Review + create'. A message states: 'You can connect to your IoT hub either publicly via its public hostname or privately using a private endpoint. [Learn more](#).' The 'Connectivity configuration' section has two radio buttons: 'Public access' (selected) and 'Private access (Recommended)'. A note below states: 'You can change this or configure another connectivity method after this resource has been created. [Learn more](#)'.

Please select the Shared access policy + RBAC permission model:

Microsoft Azure Search resources, services, and docs (G+/)

Home > Marketplace > IoT Hub >

IoT hub

Microsoft

Basics Networking **Management** Add-ons Tags Review + create

Role-based access control

Change the permission model to Azure role-based access control (RBAC) only, or to a combination of shared access policies and RBAC. [Learn more](#)

Permission model

☐ RBAC only

☒ Shared access policy + RBAC

To manage the elements within an instance, a user needs access to IoT Hub data APIs. Select the suggested role below to grant yourself full access to the APIs. You can also use Access Control (IAM) to choose appropriate roles later. [Learn more](#)

Assign me ☐ IoT Hub Data Contributor role

Scale

Device-to-cloud partitions * 2

Preview mode

Turning preview mode on means this IoT hub can't be used for production environments. Some existing features may not work. This setting can't be changed later. [Learn more](#)

Try new features like MQTT 5, TLS maximum fragment length negotiation, and ECC server cert. [Learn more](#)

Preview mode ☐ On ☒ Off

The Free tier IoT hub does not allow any Add-ons:

Microsoft Azure Search resources, services, and docs (G+/)

Home > Marketplace > IoT Hub >

IoT hub

Microsoft

Basics Networking Management **Add-ons** Tags Review + create

The following features are optional and billed separately. Microsoft recommends enabling them to ensure the most robust protections and capabilities to secure and update your fleet of devices are available. [Learn more](#)

Device Update for IoT Hub

Device Update for IoT Hub is an additional service that enables you to deploy over-the-air updates for your IoT devices. You will be charged separately for this service. See [Azure pricing](#) for more details.

Enable Device Update for IoT Hub ☐ This add-on is not supported for the selected hub tier.

Defender for IoT

Microsoft [Defender for IoT](#) is a separate service which adds an extra layer of threat protection for Azure IoT Hub, IoT Edge, and your devices. You will be charged separately for this service. Defender for IoT may process and store your data within a different geographic location than your IoT Hub. [Learn more](#)

Enable Defender for IoT ☐ 0,001 € per device per month This add-on is not supported for the selected hub tier.

We are not going to use any pre-defined Tags for this demonstration:

Microsoft Azure Search resources, services, and docs (G+)

Home > Marketplace > IoT Hub >

IoT hub

Microsoft

Basics Networking Management Add-ons **Tags** Review + create

Tags are name/value pairs. To categorize resources and consolidate billing, apply the same tag to multiple resources and resource groups. Your tags will update automatically if you change your resources. [Learn more](#)

Name	Value	Resource
<input type="text"/>	<input type="text"/>	IoT Hub

On the last page, you can review the IoT Hub configuration and you can start the deployment process by clicking the **Create** button:

Microsoft Azure Search resources, services, and docs (G+)

Home > Marketplace > IoT Hub >

IoT hub

Microsoft

Basics Networking Management Add-ons Tags **Review + create**

Pricing

IoT hub **0 € EUR** per month [Change basics](#)

Add-ons total [Change add-ons](#)

Basics

Subscription	Microsoft Partner Network
Resource group	RGURDemo
IoT hub name	IoTHubURDemo
Region	West Europe
Disaster recovery enabled	Yes
Tier	Free
Daily message limit	8 000 (0 €/month)

Networking

Connectivity configuration	Public access
Private endpoint connections	None
Allow public network access	Enabled

Management

Tier	F1
Number of F1 IoT hub units	1
Device-to-cloud partitions	2
Enable Defender for IoT	Disabled
Preview mode	Off

Device Update for IoT Hub

Disabled

Tags

[Create](#) [< Previous: Tags](#) [Next >](#) [Automation options](#)

Since the deployment is successfully finished, you can find your new IoT Hub **IoTHubURDemo** in the resource group **RGURDemo**.

SHARED ACCESS POLICIES

Open your new IoT Hub **IoTHubURDemo** in the resource group **RGURDemo** and in the left blade find Shared access policies in the Security settings section.

The screenshot displays the Azure IoT Hub 'Shared access policies' interface. The left-hand navigation pane shows the 'Shared access policies' option selected under the 'Security settings' category. The main content area is divided into two sections: 'Connect using shared access policies' and 'Manage shared access policies'. The 'Manage shared access policies' section contains a table listing the policies.

Policy Name	Permissions
iothubowner	Registry Read, Registry Write, Service Connect, Device Connect
service	Service Connect
device	Device Connect
registryRead	Registry Read
registryReadWrite	Registry Read, Registry Write

The right-hand pane shows the details for the 'iothubowner' policy. It includes options to regenerate keys and swap keys. The 'Primary key' and 'Secondary key' fields are visible, along with 'Primary connection string' and 'Secondary connection string' fields. A 'Copy to clipboard' button is present next to the primary connection string. The 'Permissions' section shows a list of permissions with checkboxes: Registry Read (checked), Registry Write (checked), Service Connect (checked), and Device Connect (checked).

From the list of policies select the existing `iothubowner` policy. On the right blade, copy the primary connection string to the clipboard and keep it for future use.

DEVICE EXPLORER

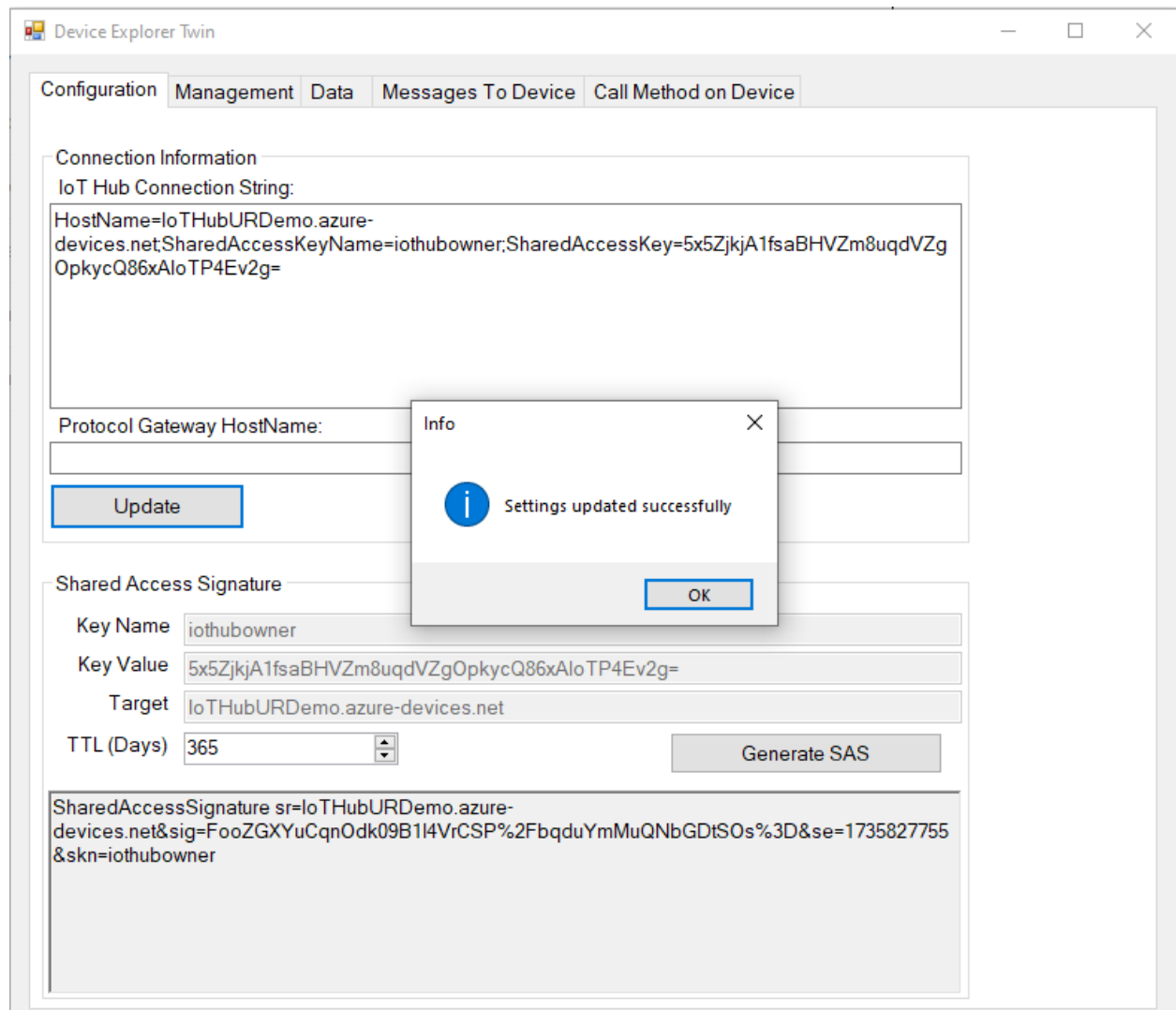
For the Device definition and monitoring we are going to use the Device explorer utility from the Azure IoT SDK.

Please download the utility from the <https://github.com/Azure/azure-iot-sdks/releases>

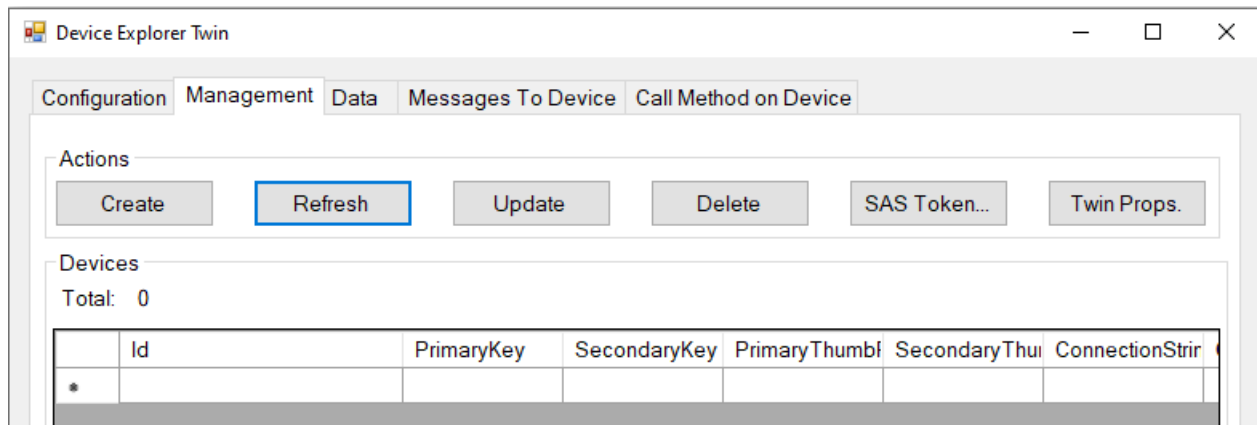
website. Point to the Azure IoT SDKs section and scroll down to the assets. The recent version of the SetupDeviceExplorer.msi file should be available here.

The version of the utility from the time of this application note should be directly available from <https://github.com/Azure/azure-iot-sdks/releases/download/2016-11-17/SetupDeviceExplorer.msi>

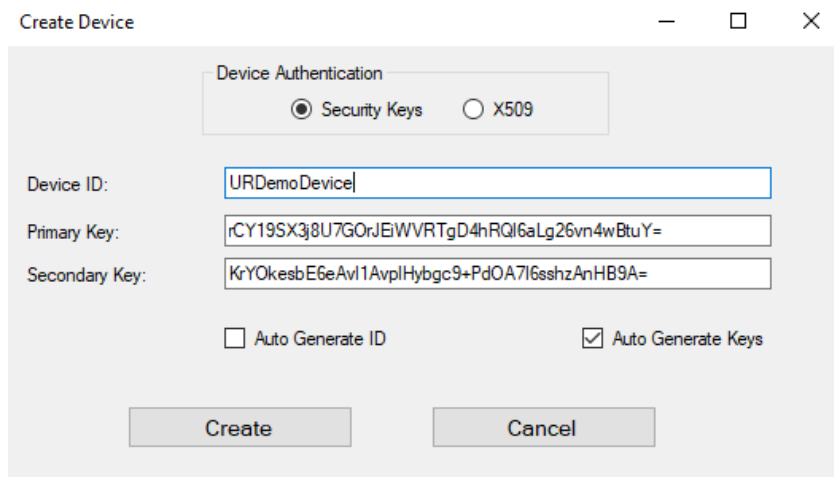
Insert the connection string of your IoT Hub to the Configuration tab of the Device explorer and press Update:



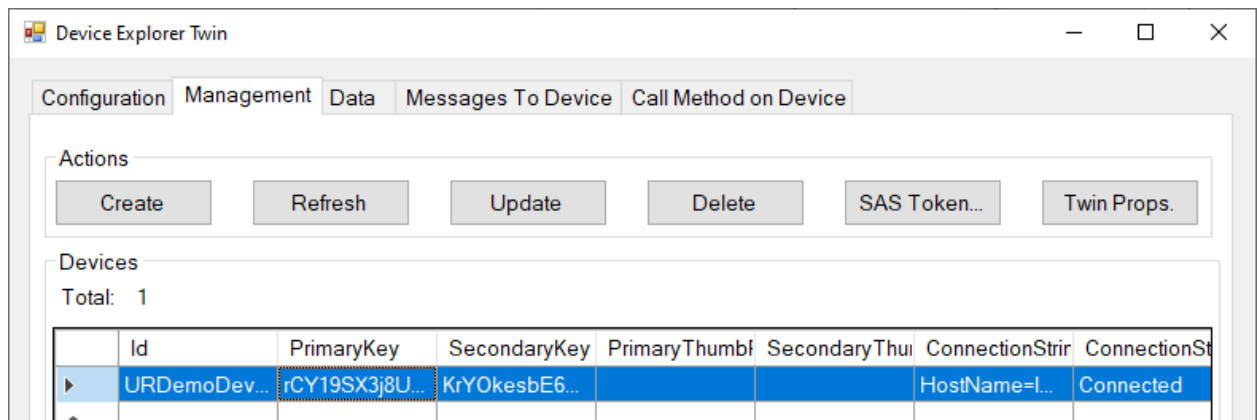
Go to the Management Tab and create the device by pressing the `Create` button:



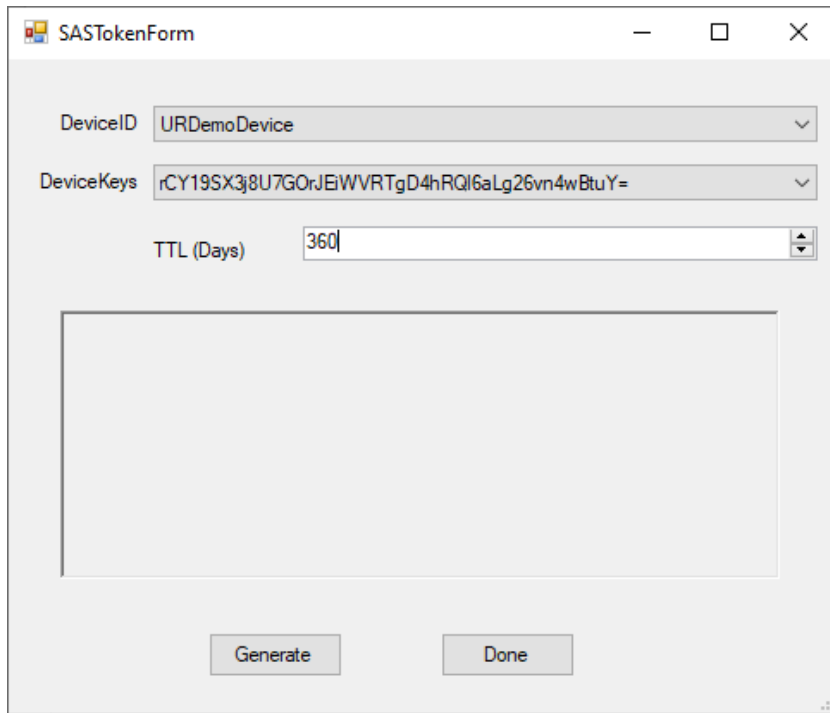
Create the device in the dialog. Define only the name of the device and keep default values in other dialog fields. We are going to use the name **URDemoDevice**.



Since The device is created, it should be available in the list on the management Tab of the Device Explorer.



Select the SAS Token... function for generating the token:



In the SASTokenForm, please, enter the TTL value for the device.

NOTE

Please don't leave the TTL field on its default 0 value, it will cause the connectivity issues.

After choosing the Generate function, please copy the generated text (token) and keep it for future use.

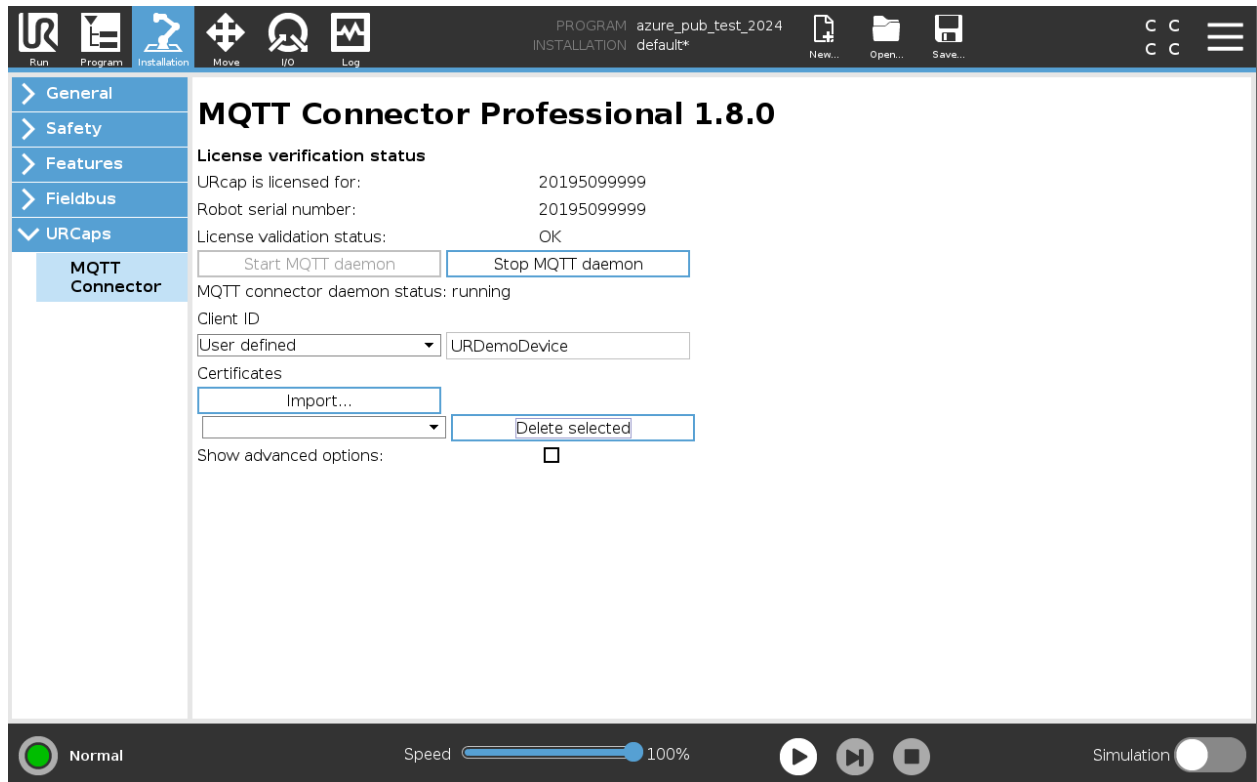
ROBOT INSTALLATION

NOTE

As we are going to transfer complex string values to the robot, we recommend creating the simple Installation and robot user program in the URSim environment and transfer it to the real robot. You can download the Simulator (URSim) on the website <https://www.universal-robots.com/download>.

As a first step in the robot environment is to use the device name as an MQTT Client Id in the installation Tab of the MQTT Connector URCap.

Please Select the User defined Client Id and enter the name of the device same as it is defined in the Device explorer Utility. In our case we use the **URDemoDevice**.

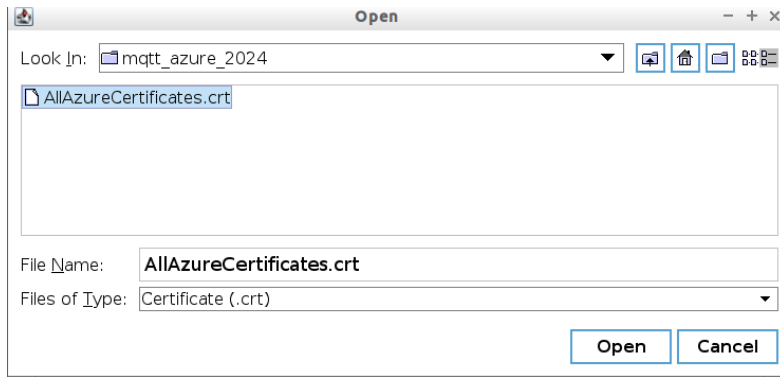


The second step is importing the certificate set for the TLS Communication with MS Azure. Chose the `Import...` option and select the Certificate file.

NOTE

- You can download the `AllAzureCertificates.crt` file from the Documentation section of the product website <https://4each.cz/mqtt-connector-professional>.

Find the certificate file on the filesystem of the robot in the Import dialog.

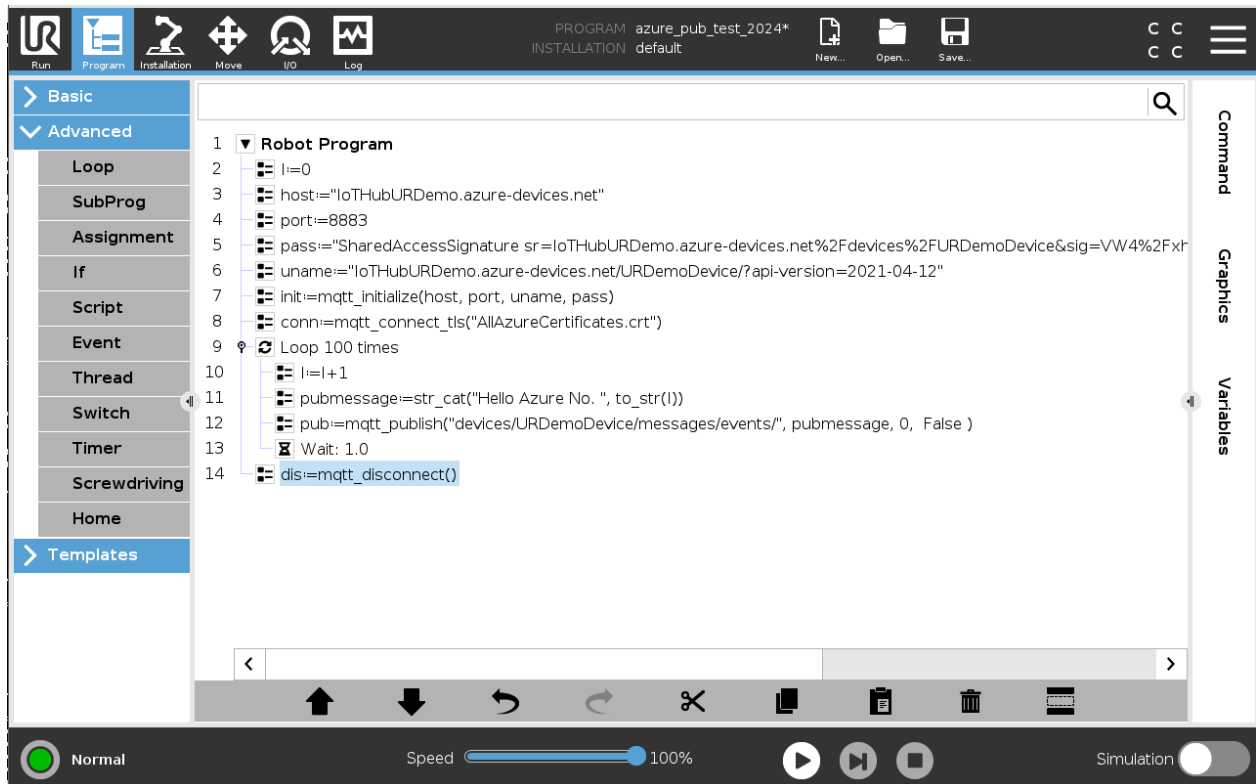
**NOTE**

- You can transfer files to the robot by using SFTP or SCP protocol (we usually use the WINSCP utility for file transfers. Available on the <https://winscp.net> website).
- The location for user programs in the URSim environment is /home/ur/ursim-current/Programs.UR10
- The location for user programs in the real robot is /programs

After the successful import, the certification file should be visible in the list.

**ROBOT PROGRAM**

The main program body is as following (note that robot program loops forever property is disabled)



The following parameters are required to successfully initialize the connection:

- **Host:** The address of the host consists of the name of the IoT Hub and the azure-devices.net domain. In our case the host is defined as ioTHubURDemo.azure-devices.net. The parameter is type of string, so it is necessary to put the value in double quote ("").
- **Port:** The port of the IoT Hub is 8883 (standard MQTT port for TLS connectivity)
- **Username:** The username should follow this pattern {iotHub-hostname}/{device-id}/?api-version=2021-04-12, where {iotHub-hostname} is the full CName of the IoT Hub. In our case the username should look like ioTHubURDemo.azure-devices.net/URDemoDevice/?api-version=2021-04-12
- **Password:** Should be part of the SAS Token we've kept from the Device Explorer. The pattern of the password is SharedAccessSignature sig={signature-string}&se={expiry}&sr={URL-encoded-resourceURI}. Let's consider the following SAS token:

```
HostName=IoTHubURDemo.azure-devices.net;DeviceId=URDemoDevice;SharedAccessSignature=SharedAccessSignature sr=IoTHubURDemo.azure-devices.net%2Fdevices%2FURDemoDevice&sig=VW4%2Ffxhd53JosF9SRyMZ3pveDLIONAQQiDR%2Bt9FBGtLA%3D&se=1735549327
```

After removing the grey part, the rest of the SAS token should be used as a password.

All the parameters mentioned above are used in the standard MQTT Connector function **mqtt_initialize**.

mqtt_connect_tls— opens network connection between robot and MQTT broker. Function takes one string parameter with the name of the certificate file. In our case "AllAzureCertificates.crt".

Afterwards, program jumps into the testing loop, which contains the **mqtt_publish** function.

The **mqtt_publish** function expects following parameters:

- **Topic:** The Topic parameter should fit the following pattern devices/{device-id}/messages/events/.

In our case the Topic parameter should look like
devices/URDemoDevice/messages/events/

- **Payload:** Some data to be sent to the MS Azure. In our case we are sending String "Hello Azure No." with the iterator number of the loop.

mqtt_disconnect - Disconnects from the broker cleanly. It is advised to use disconnect function to clean up network resources and to let broker know, that client has disconnected voluntarily.

FOOTNOTES

- Program doesn't check whether broker is connected or not. It tries to connect, and then loops forever. However, if connection does become available in the future, it connects automatically, i.e. it performs automatic reconnect.
- Return values from functions are stored in variables, but never used. They are set like this for clarity and ease of debugging thanks to variables lookup table in Polyscope.