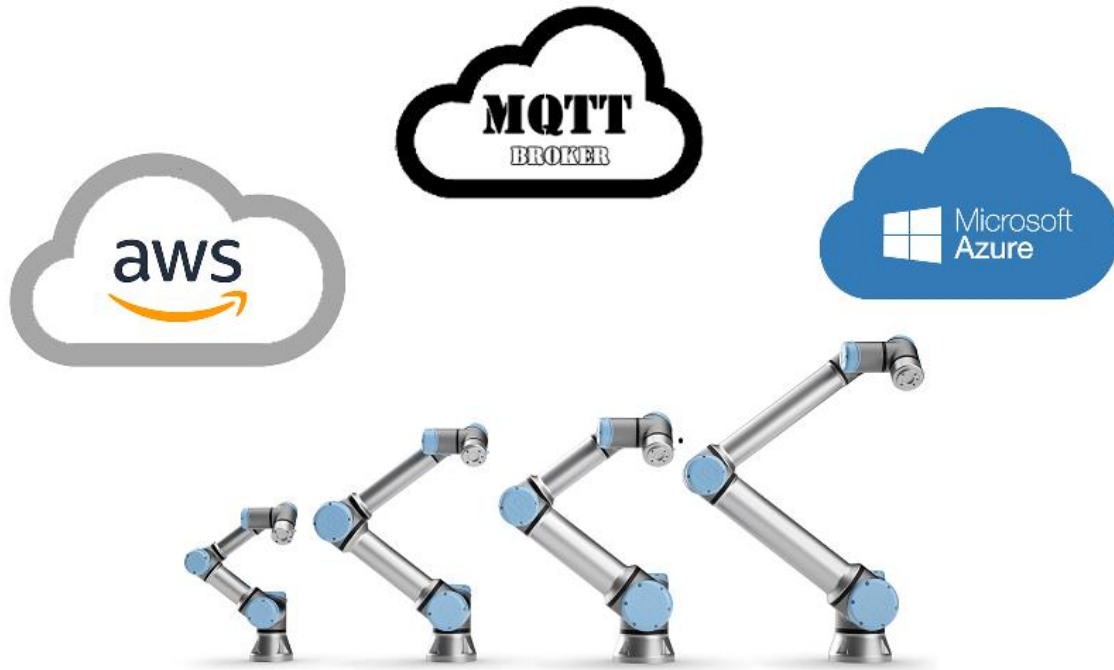


Introduction

The Cloud Connector is an URCap made for Universal Robot robots. It makes it possible to publish and subscribe data from the robot to cloud services.



This user manual describes how to use the Cloud Connector.

Download of software

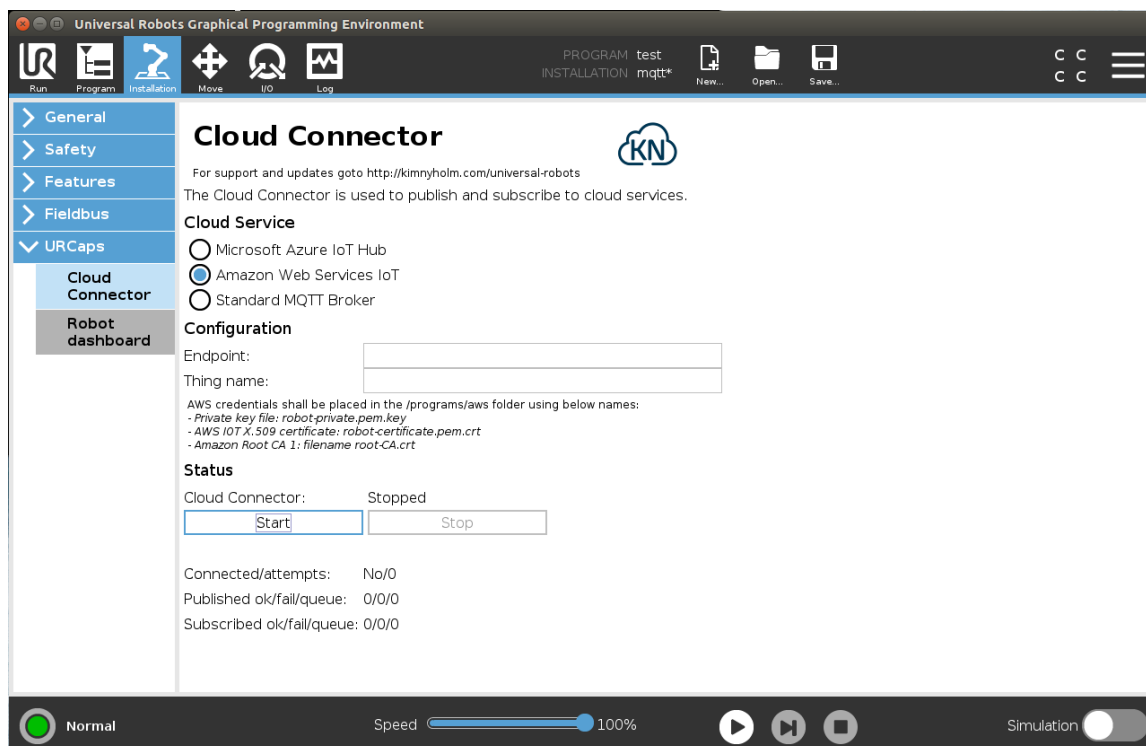
Go to <http://kimnyholm.com/universal-robots/> to download the latest version of the Cloud Connector URCap.

Installation

To install the Cloud Connector follow the steps below:

- Copy the downloaded URCap to the /programs directory of the robot
- In Polyscope go into **Setup robot** and from there **URCaps** and install the URCap.

- After restart of Polyscope the **Cloud Connector** is present in the Installation



Usage

When using the Cloud Connector, the user should understand:

- Configuration of the cloud service
- Programming the robot
- Running the program

Using Microsoft Azure

The Cloud Connector links to IoT Edge devices placed in IoT Hubs.

To connect the robot with an Azure IoT device, the “Primary Connection String” of the IoT device shall be configured in the Cloud Connector.

Data from the robot is send as telemetry data to Azure.

Topics are used to identify the telemetry attributes.

Data from the robot is send using device-to-cloud messages. Data from the robot publishes on a topic, which is the name of the attribute of the Azure IoT device.

Data from an Azure IoT device is send using cloud-to-device messages. Data from an Azure IoT device is received by subscribing to a topic, which is the name of the attribute of the IoT device.

Using Amazon Web Services

The Cloud Connector links to Amazon IoT Core using MQTT.

To connect the robot with Amazon IoT, the credentials of an Amazon IoT Thing is required:

- Private key for thing placed in file `/programs/aws/robot-private.pem.key`



-
- Certificate for thing placed in file /programs/aws/robot-certificate.pem.crt
 - Endpoint for thing
 - Name of thing
 - Root ca 1 certificate for amazon placed in file /programs/aws/root-CA.crt

The private key and certificate is generated under the things “Security”

The endpoint of the thing can be found under the things “Interact”

The root certificate can be found at

<https://www.amazontrust.com/repository/AmazonRootCA1.pem>

Two different Amazon modes are available, dependent on whether useShadow is enabled.

- When useShadow is enabled, the robot will publish data to the *reported* property of the shadow and subscribe to properties of the *desired* property of the shadow
- When useShadow is disabled, the robot will publish and subscribe data directly to the specified topic. Any valid topic can be used.

Using MQTT Broker

The Cloud Connector links to MQTT brokers using MQTT (no SSL).

To connect the robot with the MQTT broker, the following are required:

- MQTT host (DNS name or IP)
- MQTT port
- Broker username
- Broker password

Data from the robot is send as MQTT data to the broker. Any valid topic can be used to send data.

Data from the robot is received as MQTT data from the broker. Any valid topic can be subscribed.

Configuration and test of cloud service

Configuration requires the Cloud Connector to be stopped.

When starting the Cloud Connector, it will attempt to connect to the cloud service without sending any data.

The “Status” part of the installation tab shall show that the Cloud Connector it connected to the cloud service. If it is not connected, either the configuration is wrong or there are network issues.

Programming the robot

The Cloud Connector is integrated into Polyscope as functions. The following functions are available:

- CloudPublish(topic, value)
- CloudSubscribe(topic)
- CloudUnsubscribe(topic)
- CloudGetMessage(topic)
- CloudGetMessageCount(topic)
- CloudIsConnected()



All functions are non-blocking.

See the annex API reference for details.

Running the program

Every time the robot program is started, the Cloud Connector will shut down any existing connection and establish a new connection.

All existing subscriptions will be released, and all statistics will be reset.

All messages are send using QoS 0.



API Reference

CloudGetMessage(topic)

Description

Retrieves a queue message for the specified topic.

Parameters

- topic: a string representing an attribute (for Azure) or a topic (for Amazon and MQTT)

Returns

- value received on the topic. The type of the value depends on the message. The type can be integer, float, bool, string, pose and none

Notes

The return type of this function depends on the message returned.

When used in a context, where another type than the one returned is expected, the robot program will create an exception and stop. E.g. when assigning to an integer variable, be sure that the topic always returns an integer.

If there are no messages available for the specified topic “none” is returned. Assigning none to other types will result in an exception and stop. E.g. be sure that data is available when using this function. This can be checked with the function CloudGetMessageCount(topic)

CloudGetMessageCount(topic)

Description

Retrieves number of queued message for the specified topic.

Parameters

- topic: a string representing an attribute (for Azure) or a topic (for Amazon and MQTT)

Returns

- integer: the number of queued messages for the specified topic or -1 if the subscription does not exist, see CloudSubscribe(topic)

CloudIsConnected()

Description

Checks if the cloud service is connected.

Parameters

- none

Returns

- bool: true if cloud service is connected, otherwise false



CloudPublish(topic, value)

Description

Publishes data from the robot to the cloud service. Be aware how data is handled by the cloud service as described previously.

Parameters

- topic: a string representing an attribute (for Azure) or a topic (for Amazon and MQTT)
- value: the result of an expression in Polyscope. The type depends on the expression.

Returns

- a string with a unique message id

Notes

If the cloud service is not connected, the message will be dropped.

CloudSubscribe(topic)

Description

Subscribe to the specified topic.

Parameters

- topic: a string representing an attribute (for Azure) or a topic (for Amazon and MQTT)

Returns

- string: "OK" if subscription has been issued, otherwise an error description

Notes

It may take some time before the subscription is activated by the cloud service. Until the subscription is activated CloudGetMessageCount(topic) will return -1

CloudUnsubscribe(topic)

Description

Unsubscribe the specified topic.

Parameters

- topic: a string representing an attribute (for Azure) or a topic (for Amazon and MQTT)

Returns

- string: "OK" if un-subscription succeeded, otherwise an error description



Data types and message formatting

All data is sent as strings. Below is shown examples of how the different data types are formatted.

- Integer: 7
- Float: 7.0
- Bool: true
- String: "Some string"
- Array (of integers): [0,7]
- Pose: {"rx":3,"ry":-1,"rz":-2,"x":0,"y":1,"z":2}



Additional information and support

For more information see <http://kimnyholm.com/universal-robots/> or contact support@kimnyholm.com.