

VERSION 1.0
DECEMBER 14, 2017

MQTT CONNECTOR PROFESSIONAL

APPLICATION NOTE 02
Subscribe data example



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

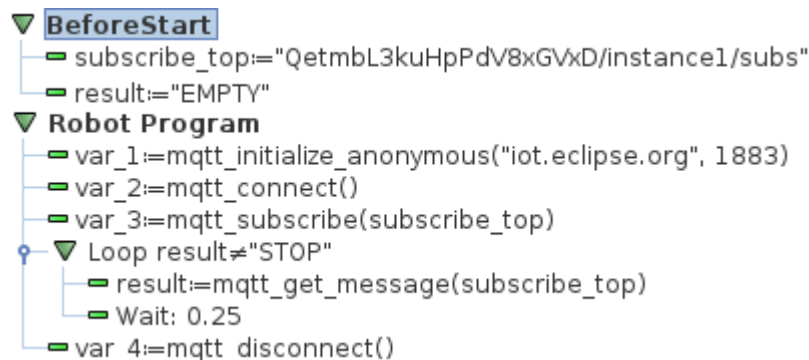
APPLICATION NOTE #2

In this series of application notes, we will go through basic usage of MQTT Connector Professional URCap extension. We will cover messaging fundamentals, like message sending and subscribing.

In this tutorial, we will learn how to subscribe and receive messages from broker.

Usage of this URCap is realized through calling each function using URScript. This can be done using “Script Code” control, or as expression for variable assignment. Later has significant advantage, because you can easily keep track of the returned value. In most cases, return value consist of status code, which can be compared to gain the ability to react to unexpected errors, such as loss of connection. Consult status code chart for more details.

PROGRAM BODY



NOTE DESCRIPTION

Before start sequence initializes variables, to be later used in program body.

- subscribe_top – MQTT topic to subscribe to as string
- result – init value for subscribe variable as string

ROBOT PROGRAM

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

mqtt_initialize_anonymous initializes connection details. In this instance, anonymous access is used. Function takes two parameters, MQTT broker **address** and **port**. Consult API documentation for details about access using credentials.

mqtt_connect– opens network connection between robot and MQTT broker. Function takes no parameters, which sets maximum blocking duration of function call to default value (consult documentation on exact value).

mqtt_subscribe - Subscribes the client to one topic. Function takes one parameter, **subscribe topic**. Without subscription, no message will be ever received. As parameter, we use subscribe_top variable.

Afterwards, program jumps into loop, which runs as long as result variable is not “STOP”. In every iteration, following happens:

mqtt_get_message – reads message from incoming messages queue. Function takes one parameter:

- **topic** – topic to identify desired message

This function returns last received value for given topic. If no message has yet arrived, ERROR code is returned. Consult API documentation for details.

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 function 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.