
Robotix-Academy Roadshow 2022

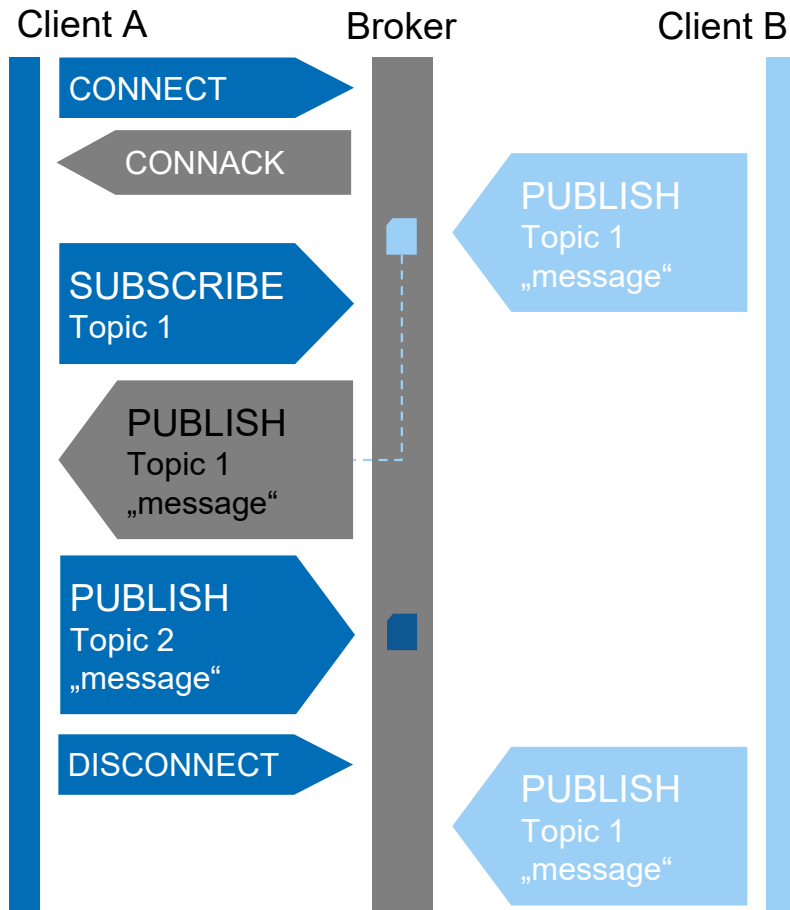
MQTT on Universal Robots – Sending Robot Poses

Stefan Marx M.Sc.

ZeMA gGmbH
Zentrum für Mechatronik und Automatisierungstechnik

Liège, 10.03.2022

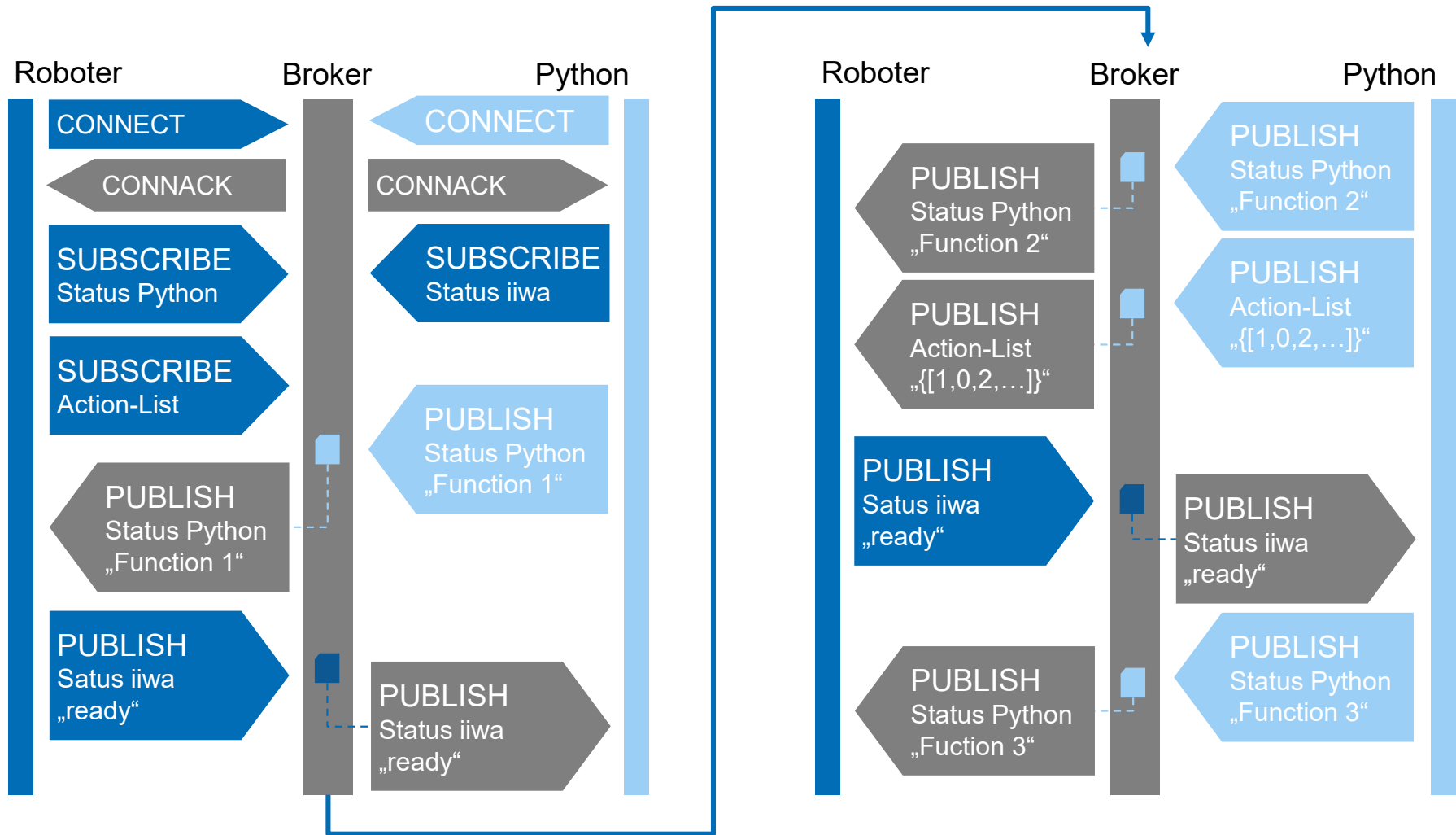
MQTT



- MQTT (Message Queuing Telemetry Transport)
 - network protocol for M2M-communication
 - client-server-principle
 - server (broker) manages message traffic
 - sender and receiver connect to broker
 - topic declares subject of a message
- Properties:
 - QoS (Quality of Service): (0, 1, 2)
 - Last Will
 - Retained Messages
- MQTT Server can be operated locally
 - e.g. [Mosquitto Broker](#)

Quellen: <http://mqtt.org/>, <https://www.informatik-aktuell.de>, angelehnt an <https://de.wikipedia.org/wiki/MQTT>

MQTT



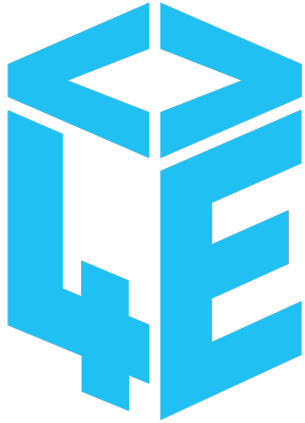
JSON

■ JSON (JavaScript Object Notation)

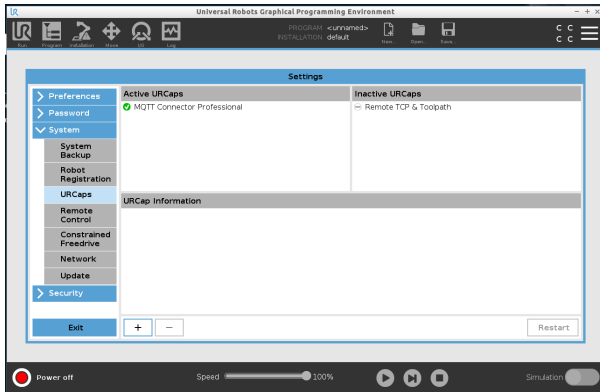
- parsers in common programming languages
- Data exchange between different languages and systems
- Structure like a library
 - {<key>: <data>}
- Different data types

Data type	Example
String	{ "Name": "Robot" }
Number	{ "Number": 360 }
Boolean	{ "Boolean": false }
Null	{ "Not needed": null }
Object	{ "Specifications": { "name": "iiwa", "mass": 28, "cobot": true } }
Array	{ "Robots": [90, 28, 35, 105, 180, 3, 14] }

MQTT on Universal Robots



- URCap von 4EACH (<https://4each.cz/mqtt-connector-professional>)
- Different functions available



A screenshot of the Universal Robots Graphical Programming Environment main interface. The 'Command' tab is selected, showing a list of MQTT functions. The 'Assignment' block is active, showing a variable 'var_1' assigned to the expression 'f(x)'. The interface includes a toolbar with icons for Run, Program, Installation, Move, I/O, and Log. The 'Command' list includes: mqtt_initialize_anonymous(<hostname>,<port>), mqtt_initialize(<hostname>,<port>,<user>,<passwor...), mqtt_set_max_queue(<length>), mqtt_set_last_will(<topic>,<message>,<qos>,<retain...), mqtt_set_publish_on_stop(<topic>,<message>,<qos...), mqtt_connect(), mqtt_connect_timeout(<timeout>), mqtt_connect_tls(<certfile>), mqtt_publish(<topic>,<message>,<qos>,<retained>), mqtt_publish_timeout(<topic>,<message>,<qos>,<re...), mqtt_publish_json(<topic>,<qos>,<retained>), mqtt_publish_json_timeout(<topic>,<qos>,<retained>...), mqtt_json_message_clear(), mqtt_json_message_add(<name>,<value>), mqtt_subscribe(<topic>), and mqtt_unsubscribe(<topic>). The interface also features a keyboard-like control panel with buttons for logical operations (False (LO), xor, not), arithmetic operations (/, *), comparison operators (<, >, =), and a numeric keypad (0-9, ., -). A 'Submit' button is also present.

Thank you for your attention

Contact

Stefan Marx

s.marx@zema.de

+49 (0)681 - 85 787 - 553