

VERSION 1.0
DECEMBER 14, 2017

MQTT CONNECTOR PROFESSIONAL

API REFERENCE
Developer guide



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

MQTT CONNECTOR PROFESSIONAL API REFERENCE

Documentation valid for URCap version 1.0.0

MQTT_INITIALIZE

initializes connection details using credentials.

It takes the following arguments:

- Hostname – the hostname or IP address of the remote broker [string]
- Port - the network port of the server host to connect to [int]
- User – username to access to remote broker [string]
- Password - password to access to remote broker [string]

Note, that some brokers will accept anonymous users, even though they provide username and password.

MQTT_INITIALIZE_ANONYMOUS

initializes connection details using anonymous access (without username and password).

It takes the following arguments:

- Hostname – the hostname or IP address of the remote broker [string]
- Port - the network port of the server host to connect to [int]

MQTT_SET_MAX_QUEUE

Sets maximum outgoing messages queue length. This is important in case of connection loss, because every queued message is send after connection is restored. This can cause temporary network overload. Parameter with value 0 sets unlimited queue length. When the queue is full, any further outgoing messages would be dropped.

- Length - maximum outgoing messages queue length [int]

MQTT_SET_LAST_WILL

Set a Will to be sent to the broker. If the client disconnects without calling disconnect(), the broker will publish the message on its behalf.

- Topic - the topic that the will message should be published on [string]
- Message - the message to send as a will [string]
- Qos - the quality of service level to use for the will [int]
- Retained - if set to True, the will message will be set as the “last known good”/retained message for the topic [bool]

Note, that this function must be called after initialize function and before connect function.

MQTT_CONNECT

Connects the client to a broker, using default timeout period of 5 seconds.

MQTT_CONNECT_TIMEOUT

Connects the client to a broker, using custom timeout period defined by parameter.

- Timeout - Ensures maximum blocking duration of function call. Time is defined in milliseconds, and changes in 100ms increments. [int]

MQTT_SUBSCRIBE

Subscribe the client to one topic.

- Topic - subscription topic to subscribe to [string]

MQTT_UNSUBSCRIBE

Unsubscribe the client from one topic.

- Topic - subscription topic to unsubscribe from [string]

MQTT_UNSUBSCRIBE_ALL

Unsubscribe the client from all topics.

MQTT_GET_MESSAGE

Function used to lookup incoming messages buffer. Returns latest received value from given topic.

- Topic – Specifies topic to search for [string]

Note, that this function has different communication interface than other methods. It doesn't return OK status code, but actual message payload. If error is present, error codes are returned.

MQTT_PUBLISH

This function causes a message to be sent to the broker and subsequently from the broker to any clients subscribing to matching topics, using default timeout period of 1 second. It takes the following arguments:

- Topic - the topic that the will message should be published on [string]
- Message - the message to send as a will [string]
- Qos - the quality of service level to use for the will [int]
- Retained - if set to True, the will message will be set as the "last known good"/retained message for the topic [bool]

MQTT_PUBLISH_TIMEOUT

This function causes a message to be sent to the broker and subsequently from the broker to any clients subscribing to matching topics, using custom timeout period defined by parameter. It takes the following arguments:

- Topic - the topic that the will message should be published on [string]
- Message - the message to send as a will [string]
- Qos - the quality of service level to use for the will [int]
- Retained - if set to True, the will message will be set as the “last known good”/retained message for the topic [bool]
- Timeout - Ensures maximum blocking duration of function call. Time is defined in milliseconds, and changes in 100ms increments. [int]

MQTT_DISCONNECT

Disconnects from the broker cleanly. Using disconnect() will not result in a will message being sent by the broker. Disconnect will not wait for all queued message to be sent.

DEBUGGING PRACTICE AND ERROR CODES

To ease programing, error troubleshooting and general ease of use, MQTT Connector uses error codes as return values for most of its functions. This can be helpful in situations where client-broker connectivity is not stable, or during initial setup.

General recommended usage

- Compare return values against constants or predefined variables using IF statement. Return values should be compared against error codes table.
- Store return values in variables. Even though value is never compared against error codes, it can be helpful to view variable values in Polyscope graphical interface.

ERROR CODE TABLE

<i>Function</i>	<i>Description</i>	<i>Status</i>	<i>Error code</i>
initialize	INITIALIZATION OK	OK	OK-CODE-010
	UNKNOWN ERROR	NG	NG-CODE-011
connect	MQTT CONNECT OK	OK	OK-CODE-020
	NO ROUTE TO HOST	NG	NG-CODE-021
	FAILED – timeout	NG	NG-CODE-022
	UNKNOWN ERROR	NG	NG-CODE-023
disconnect	MQTT DISCONNECT OK	OK	OK-CODE-030
	UNKNOWN ERROR	NG	NG-CODE-031
subscribe	SUBSCRIBE OK	OK	OK-CODE-040
	SUBSCRIBE FAILED – already exists	NG	NG-CODE-041
	UNKNOWN ERROR	NG	NG-CODE-042
unsubscribe	UNSUBSCRIBE OK	OK	OK-CODE-050
	UNSUBSCRIBE FAILED – no such subscription	NG	NG-CODE-051
	UNKNOWN ERROR	NG	NG-CODE-052
unsubscribe_all	UNSUBSCRIBE ALL OK	OK	OK-CODE-060
	UNKNOWN ERROR	NG	NG-CODE-061
get_message	<i>MESSAGE</i>		
	NO SUCH SUBSCRIPTION	NG	NG-CODE-071
	FALSE PARAMS	NG	NG-CODE-072
	UNKNOWN ERROR	NG	NG-CODE-073
publish	SUCCESS	OK	OK-CODE-080
	FAILED	NG	NG-CODE-081
	SEND FAILED – broker disconnected	NG	NG-CODE-082
	FALSE PARAMS	NG	NG-CODE-083
	UNKNOWN ERROR	NG	NG-CODE-084
set_max_queued_messages	SET OK	OK	OK-CODE-090
	UNKNOWN ERROR	NG	NG-CODE-091
set_last_will	SET OK	OK	OK-CODE-100
	UNKNOWN ERROR	NG	NG-CODE-101