MQTT Client Protocol (Message Queue Telemetry Transport)

Supported device types and versions

The protocol is an implementation of the MQTT 3.1.1 standard (October 2014). MQTT protocol is a client/server protocol of subscribe/publish type. It is simple, has little overhead and is easy to implement. It is used for M2M communication (Machine to Machine) and in the IoT context (Internet of Things). D2000 KOM implements the client part of the protocol. The protocol je implemented on a TCP/IP line.
For transfer of LoRaWAN data encapsulated within the MQTT protocol, see LoRaWan protocol description.

Each PUBLISH message contains a topic (Topic), data (Payload) and level of confirmation (QoS). PUBLISH messages can be sent both by the client and the server. The clients at the beginning of the communication will use the SUBSCRIBE message to indicate what topics (parameter of Topic Filter protocol) they are interested in.
The protocol defines the following levels of confirmation of PUBLISH messages - QoS (Quality of Service):

- **QoS_0** - PUBLISH message is not confirmed, it may be lost
- **QoS_1** - PUBLISH message is confirmed by other side's PUBACK, it may be duplicate
- **QoS_2** - PUBLISH message is confirmed by other side's PUBREC which is then confirmed back by the PUBREL message and that one by a final PUBCOMP message.

The level of confirmation of the messages sent by the D2000 KOM process is defined by the protocol parameter Publish QoS . D2000 KOM process considers the writing of the output tag to be successfully finished depending on the QoS:

- **QoS_0** - after the data is successfully sent via TCP connection
- **QoS_1** - after receiving PUBACK
- **QoS_2** - after receiving PUBCOMP

The MQTT communication starts with the CONNECT message sent by client (D2000 KOM). Message contains User Name, Password and other parameters, from which only Clean Session Flag and Client ID can be modified (parameter *Will Flag* is not used, as well as *Will QoS* and *Will Retain*, parameter *Keep Alive* is set to 0). Server replies with CONNACK message with a return code that contains information about the success of connect operation.

Then client sends SUBSCRIBE message with filter of topics (Topic Filter parameter), specifying which topics it is interest in, and with the required maximum level of confirmation (parameter Subscribe QoS).

The server responds with a return code that contains information about the success and maximum QoS that was assigned to the requested topics.

Then follows a phase of communication, during which both the client and the server send PUBLISH messages (the client with any topic, the server with topics relating to the filter of topics of the received SUBSCRIBE message) and confirm them according to the value of the QoS parameter of the received PUBLISH messages.

If the server does not send a message for longer than Ping Interval seconds, the client sends the PING request message, to which the server must respond with the PING response message (within the time specified by the Reply Timeout parameter).
If parameters change on the line, the connection is closed and re-created.

The communication has been tested with the MQTT server www.TheThings.network.

Communication line configuration

- Communication line category: TCP/IP-TCP.
- Host: IP address of MQTT server (or redundant addresses separated by a coma or semicolon).
- Port: default port number is 1883 or 8883 for the encrypted SSL/TLS variant.
- Line number: unused, set the value to 0.

Note: Default port for the MQTT protocol is 1883 or 8883 for SSL/TLS version. D2000 KOM does not contain implementation of the SSL/TLS protocol variant, but it is possible to configure it by using the stunnel utility http://www.stunnel.org working in a client mode (client = yes). Stunnel running on the same computer as the D2000 KOM should listen to the 1883 local port and after connecting of D2000 KOM process to the port should encrypt the communication using SLL/TLS and send to the target MQTT server (typically on port 8883).

Communication line parameters

Dialog link configuration - **Protocol parameters** tab.
They affect some optional protocol parameters. The following protocol line parameters can be entered:

Table 1

Parameter	Description	Unit / size	Default value
Full Debug	Activates detailed debug information about sending and receiving values.	YES /NO	NO
User Name	User name used in CONNECT message to connect to the MQTT server.	-	
Password	Password used in CONNECT message to connect to the MQTT server.	-	
Topic Filter	The name of one topic or a multiple topic filter sent within the SUBSCRIBE message. Using the filter the MQTT client specifies topics, within which it wants to receive messages. **Note:** topics are hierarchically sorted, a slash (/) is used as the separator, a plus (+) is used as a one-level mask, a hash (#) character is used a mask for multiple levels. Examples of filter: a/b , level1/+ , # , +/+/+/up	-	#
Subscribe QoS	The desired maximum level of validation (QoS) sent within the SUBSCRIBE message. The MQTT server can then send PUBLISH messages with such or lower level of confirmation (but not higher). PUBLISH messages sent by the MQTT server will be confirmed by the D2000 KOM process according to the level of confirmation specified in them. The higher the level of confirmation, the more messages between the client and the server are exchanged (1 at QoS_0, 2 at QoS_1 and 4 at QoS_2).	QoS_0 QoS_1 QoS_2	QoS_1
Client ID	Unique client identifier (Client Identifier) sent within the CONNECT message. **Note:** it is possible to enter a blank string - in which case the server can assign a unique name to the client (if it supports such functionality) or return an error. However, if the Client ID is not specified, the Clean Session Flag parameter settings will be ignored (as the server will assign a unique name each time). The tested MQTT server (thethings.network) returned an error if the Client ID was blank and Clean Session Flag=NO.	-	
Clean Session Flag	Parameter Clean Session Flag of the CONNECT message. The *No* value means that the server uses the current session state (connection) - e. g. after collapse and recovery of the TCP connection. This means that all unconfirmed PUBLISH messages with QoS_1 and QoS_2 are resent (optionally also QoS_0, depending on the implementation). The Yes value means that the session is re-created and unconfirmed PUBLISH messages are not repeated.	YES /NO	NO
Publish QoS	Level of confirmation (QoS) used to send PUBLISH messages through the D2000 KOM process. Sending the PUBLISH message is the outcome of writing into the output tag with OUT_DATA address. The higher the confirmation level, the more messages between the client and server are exchanged (1 for QoS_0, 2 for QoS_1 and 4 for QoS_2).	QoS_0 QoS_1 QoS_2	QoS_0
Ping Interval	If the MQTT server did not send any message during the specified time interval, the D2000 KOM process sends a PING request and waits for a PING response response (until time Reply Timeout). A value of 0 turns off sending the PING request messages. Parameter allows detection of TCP connection failure.	sec	60
Reply Timeout	If the MQTT server does not respond to the SUBSCRIBE, UNSUBSCRIBE, and PING requests within the required time or the D2000 KOM process fails to read a complete message (and only part of it is read), the D2000 KOM process declares an error, closes the connection, and opens it again. Value 0 turns off the timeout. The parameter enables handling of problematic behavior of the MQTT server.	sec	20
Wait Timeout	Timeout of a single reading from TCP connection. D2000 KOM repeats reading of spontaneous data Max. Wait Retry times and if no data is read, the reading is timeouted and finished (and may be followed by a further reading or writing). By lowering Wait Timeout and Max. Wait Retry parameters, it is possible to achieve a faster writing response of D2000 KOM process at the expense of a higher CPU load when the MQTT server has no data.	sec	0.100
Max. Wait Retry	Number of repetitions of reading from TCP connection. See description of the Wait Timeout parameter.	-	3

Communication station configuration

- Communication protocol "**MQTT Client Protocol**".
- Station address: station address is a particular topic (Topic) or a character # representing all topics. If there is a station with address # on the line, all messages are directed to its I/O tags. Otherwise, KOM looks for a station with the address corresponding to the Topic field in the PUBLISH message received from the MQTT server.
  Note: station address doesn't support wildcard characters "+" corresponding to mask of Topic Filter parameter (yet).
- Polling parameters on the *Time parameters* tab - recommended value is Delay=0.

I/O tag configuration

Possible value types of I/O tags: **Ci**, **Co**, **TxtI**, **TxtO**.

Type of I/O tag	Address	Description

I/O tags for reading data sent by MQTT server through PUBLISH message. **Note:** values of I/O tags are set by D2000 KOM process in the order IN_TOPIC, IN_DATA and IN_ID. It is not necessary for configuration to contain all three I/O tags.		
TxtI	IN_TOPIC	Topic (Topic) of received PUBLISH message.
TxtI	IN_DATA	Data (Payload) of received PUBLISH message.
Ci	IN_ID	Identifier of packet (Packet Identifier) of PUBLISH message that depends on the level of validation (QoS). For messages sent with QoS_0, the identifier is zero, for QoS_1 and QoS_2, it is a positive 16-bit number. **Note:** if MQTT server sends also messages with the QoS_0 level of validation and the ACK_ID I/O tag is configured, then we recommend to activate the option *New value when changing time* in the *Filter* tab, so that repeated writing of the value 0 will cause a new value that differs only with time stamp to be generated.
I/O tag to confirm a received data to MQTT server.		
Co	ACK_ID	If an output I/O tag with ACK_ID address is defined, the D2000 KOM expects confirmation of the processing of each message by writing a copy of value of the IN_ID tag. Only after then it sets values from the next received PUBLISH message (if it was received in the meantime) into the IN_TOPIC, IN_DATA and IN_ID I/O tags (in this order). In case of the QoS_0 level of confirmation, it is therefore necessary to repeatedly set the value of I/O tag ACK_ID to 0. If the I/O tag ACK_ID does not exist, the values are written into the IN_TOPIC, IN_DATA and IN_ID I/O tags immediately after the PUBLISH message is received and processed. **Note:** for the messages received with the QoS_0 level of validation, no confirmation is sent to MQTT server, only the values of r eceived PUBLISH message will be published.
I/O tags for sending values to the MQTT server through PUBLISH message. **Note:** in order for the D2000 KOM process to send the PUBLISH messages to the MQTT server, both I/O tags must be defined within one station.		
TxtO	OUT_TOPIC	Topic of the PUBLISH message being sent.
TxtO	OUT_DATA	Data (Payload) of the PUBLISH message being sent. **Note:** sending the message is performed out as result of writing to the OUT_DATA I/O tag (i.e. if the Topic does not change then it is sufficient to set the OUT_TOPIC point once - e.g. by using default value).

Literature

Links
Official website of MQTT protocol http://mqtt.org

Specifications and Standards
MQTT 3.1.1 specification http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html
ISO/IEC 20922:2016 http://www.iso.org/standard/69499.html

Descriptions of Data Formats and API
www.loriot.io - Application API Data Format https://www.loriot.io/home/documentation.html#docu/app-data-format
www.thethingsnetwork.org - API Reference https://www.thethingsnetwork.org/docs/applications/mqtt/api.html

Document revisions

- Ver. 1.0 - August 8th, 2017 - document creation.

Súvisiace stránky:

Communication Protocols