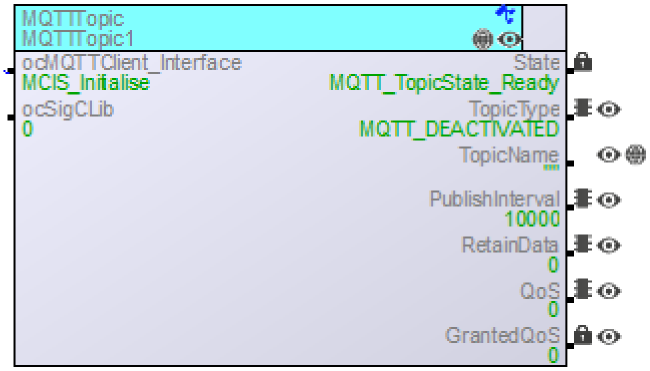
MQTTTopic



This class serves as a base class to subscribe to "MQTT Topics", or to send data to a specific "MQTT Topic" ("Publish").

Notes for the User

The following private data must be overwritten by the user:

User_SubscribeData()	Here the data for the currently logged on topic is transferred.
User_GetPublishData()	The return value must be set to the data to be sent to the defined topic.

The following private data can also be overwritten by the user:

User_ClientConnected()	Is called when the MQTT client has established a connection to the server.
User_ClientDisconnected()	Is called when the connection from the MQTT client to the server is terminated.
User_ErrorOccured()	Is called if an error has occurred in the routine.
User_SubscribedOK()	Is called if the logon to the defined topic was successful.
User_UnsubscribedOK()	Is called if the logoff from the defined topic was successful.
User_PublishDone()	Is called when the data for the defined topic has been successfully sent.

For further information refer to the interfacess description.

Interfaces

Clients

ocMQTTClient_Interface	Object channel to the MQTTClient_Interface class.	
	Data type	MQTTClient_Interface::t_e_MQTTClientStates
ocSigCLib	Object channel to the SigCLib class. Does not have to be connected.	
	Data type	DINT

Server

State	This server can be used to call the global methods of the class. Shows the current step of the routine's internal stepping mechanism.

MTS_Init	Initialization of the topic
MTS _WaitForConnection	It is waited until the client has established a connection to the server.
MTS _Ready	Class is ready for the operation defined in the server TopicType
MTS _Subscribe	A logon to the defined topic is executed.
MTS _WaitForSubscribe	The system waits for confirmation of the logon process.
MTS _Subscribed	The logon was successful
MTS _Unsubscribe	A logoff from the defined topic is executed.
MTS _WaitForUnsubscribe	The system waits for confirmation of the logoff process.
MTS _Unsubscribed	The logoff was successful
MTS _WaitForPublishInterval	It waits for the time delay defined in the server PublishInterval before sending data to the server.
MTS _WaitForPublishDataChange	It waits for a change to the user-defined data before sending data to the server.
MTS _WaitForPublishCommand	It waits for a manual trigger from the user before sending data to the server.
MTS _Publish	The send process for the user-defined data is triggered.
MTS _WaitForPublishDone	It waits for a response to the transmission process.
MTS _Error	An error has occurred during the routine.
MTS _Error_WaitForReset	It waits for the user to reset the class from the error.

Unit	-	Data type	t_e_MQTT_Topic State
Value range	-	Write Protected	TRUE
Default value	-	Retentive	FALSE

TopicType

Here you can specify how the MQTT topic defined in the server TopicName should be handled.

MQTT_DEACTIVATED	Class is deactivated We are not logged into a topic, nor are data sent to a topic.
MQTT_SUB	You want to log on to the defined topic.
MQTT_PUB_POLL	The user-defined data should be sent to the defined topic in a certain time interval, which can be set on the server PublishInterval.
MQTT_PUB_ONCHANGE	If a change of the user-defined data is detected, then these should be sent to the defined topic.
MQTT_PUB_MAN	Trigger manually to send the data to the defined topic. Can be triggered by calling the method DoManualPublish().

Unit	-	Data type	t_e_MQTT_Topic Type
Value range	0 – 4	Write Protected	FALSE
Default value	adjustable	Retentive	SRAM

TopicName

The name of the topic must be defined here.

Unit	-	Data type	UDINT Object channel for the StringRAM class.
Value range	-	Write Protected	FALSE
Default value	-	Retentive	SRAM

PublishInterval

If the topic type is set to MQTT_PUB_POLL (server TopicType), the time interval for sending the data can be defined here.

Unit	ms	Data type	UDINT
Value range	-	Write Protected	FALSE
Default value	10 s	Retentive	SRAM

RetainData

Here you define whether the data which is sent to the server for a certain topic (Publish) should be stored. If a new client logs on to this topic, the last stored data is immediately sent to it.
Otherwise the new client would only get data as soon as someone sends something to this topic again.

0...Data is not stored in the server

1...Data is stored in the server.

Unit	-	Data type	UDINT
Value range	0-1	Write Protected	FALSE
Default value	0	Retentive	SRAM

QoS

The "Quality of Service" for data transmission can be set here.

This always means between client and server (broker).

0...The data is sent at most once. The receiver does not acknowledge reception of the data. Provides the same guarantee as the underlying TCP protocol.

1...The data is sent at least once. The sender is waiting for confirmation from the receiver. If this is not done within a certain time, the data is sent again. It is possible to send and receive the same data more often.

2...The data is sent exactly once. This ensures that the recipient receives the message exactly once.

Note: The higher the "Quality of Service" is set, the longer the processing of the send and receive routine will take.

Unit	-	Data type	DINT
Value range	0-2	Write Protected	FALSE
Default value	0	Retentive	SRAM

GrantedQoS

Only relevant in the topic type MQTT_SUB.

The "Quality of Service" for the subscribed topic is displayed here. It is set by the server. The quality displayed here is at most the quality defined in the server QoS, but can also be smaller after settings of the MQTT server.

Unit	-	Data type	DINT
Value range	0-2	Write Protected	TRUE
Default value	-	Retentive	FALSE

Global Methods

Init	Class initialization method. In the first Init run, the timeouts for the Subscribe, Unsubscribe, and Publish processes are set.				
CyWork	Call of the internal routine.				
PubSubData_Callback	Callback method for the MQTTClient_Interface class for various operations. Is transferred for the Subscribe, Unsubscribe and Publish processes.				
		► pThis	Pointer to the object.		
		► MsgType	Message type:		
				PSRC_Subscribed	Confirmation of the logon process.
				PSRC_Unsubscribed	Confirmation of the logoff process.
				PSRC_PublishReceived	Receive data for a logged on topic
				PSRC_Published	Confirmation of a send process
		► iMid	Data packet ID		
		► iGrantedQoS	QoS granted by the server		
		► pMessage	Pointer to the data.		
PubSubData	Called by the PubSubData_Callback method.				
		► MsgType	Message type:		
				PSRC_Subscribed	Confirmation of the logon process.
				PSRC_Unsubscribed	Confirmation of the logoff process.
				PSRC_PublishReceived	Receive data for a logged on topic
				PSRC_Published	Confirmation of a send process
		► iMid	Data packet ID		
		► iGrantedQoS	QoS granted by the server		
		► pMessage	Pointer to the data.		

DoManualPublish	If the topic type MQTT_PUB_MAN is set, the sending of data to the server can be triggered by calling this method.		
	◄ outSuccess	True...Manual sending has been triggered. False...Manual sending could not be triggered.	
SetTimeouts	With this method the timeouts for the Subscribe, Unsubscribe and Publish processes can be set.		
	► inTimeInfo	Data type structure t_s_MQTT_TopicTimeouts **Elements:**	
		SubscribingTimeout	Timeout for the logon process [ms]
		UnsubscribingTimeout	Timeout for the logoff process [ms]
		PublishTimeout	Timeout for the send data process [ms]
ResetError	If an error occurs and the server state is set to "MTS _Error_WaitForReset", the error can be reset with this method.		
WriteTopicName	Use this method to change the current topic name.		
	► inPtrTopicString	Pointer to the string with the new topic name	
	◄ outSuccess	True...Topic name changed successfully True...Topic name change failed	
ReadTopicName	Use this method to read the current Topic Name.		
	► inDstPtrTopicString	Pointer to the target string, where the name should be stored.	
	► inLenOfTopicString	Length of the target string	
	◄ outSuccess	True…Copying of the name successful False...Copying of the name failed	
GetTopicNameLength	Returns the length of the topic name. The 0 termination is not included.		
	◄ outTopicLength	Length of the topic name without 0 termination.	

Config_SetParameter	This method can be used to configure parameters in the MQTTTopic class, for a list of parameters, see the description of the Enumeration.	
	► Parameter	Enumeration value indicating the parameter to be set.
	► Value	The value to be set for the chosen parameter.
	◄ retCode	The result of the value set operation, this value differs based on the chosen parameter.

Private Methods

Only the methods relevant to the user are described here.

User_ClientConnected	Is called when the MQTT client has established a connection to the server. Can be overwritten by the user.		
User_ClientDisconnected	Is called when the connection between MQTT client and server is terminated. Can be overwritten by the user.		
User_ErrorOccured	Called when an error occurs in the internal routine. Can be overwritten by the user. If an error occurs, the variable "ErrorOccuredState" can also be checked to determine the step of the internal routine in which the error occurred.		
User_SubscribedOK	Is called if the logon to the defined topic was successful. Can be overwritten by the user.		
	► pData	Pointer to the payload of the MQTT message if data is available. Otherwise the pointer is NIL.	
	► SizeOfData	Size of the payload in bytes.	
User_UnsubscribedOK	Is called if the logoff from the defined topic was successful. Can be overwritten by the user.		
	► pData	Pointer to the payload of the MQTT message if data is available. Otherwise the pointer is NIL.	
	► SizeOfData	Size of the payload in bytes.	
User_GetPublishData	Is called if data is to be sent to the defined topic. Must be overwritten by the user to send his specific data.		
	◄ outPublishData	Data type strucure t_s_MQTT_TopicTimeouts **Elements:**	
		PointerToData	Pointer to the data that should be sent.
		SizeOfData	Data size in bytes
User_PublishDone	Is called when the data for the defined topic has been successfully sent. Can be overwritten by the user.		
	► pData	Pointer to the payload of the MQTT message if data is available. Otherwise the pointer is NIL.	

	► SizeOfData — Size of the payload in bytes.
User_SubscribeData	Is called when data has been received from the logged in topic. Must be overwritten by the user to evaluate the data. **Note:** The transferred parameters are only valid in the current call of the method. ► pData — Pointer to the payload of the MQTT message if data is available. Otherwise the pointer is NIL. ► SizeOfData — Size of the payload in bytes.

Enumerations

t_e_ConfigParameters:

CP_SendAtStart	When TopicType = MT_Publish_OnChange: This value indicates whether the Topic is sent at start regradless whether the value has changed between the time of the last transmission and start. This means, when active the data will always be sent when the topic rei-nitializes (also at Application restart), if not active the class waits for the first change before sending the data. 0 = Inactive (Default) 1 = Active