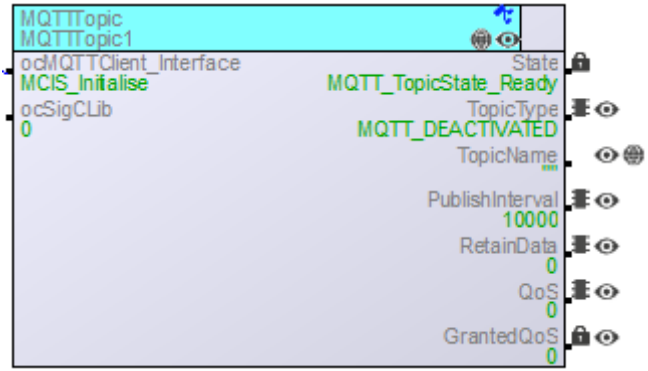
MQTTTopic



Diese Klasse dient als Basisklasse um sich an „MQTT Topics“ anzumelden („Subscribe“), oder Daten an ein bestimmtes „MQTT Topic“ zu senden („Publish“).

Hinweise für den Benutzer

Folgende private Methoden müssen vom Anwender überschrieben werden:

User_SubscribeData()	Hier werden die Daten zum aktuellen angemeldeten Topic übergeben.
User_GetPublishData()	Der Rückgabewert muss auf die Daten gesetzt werden, welche zum definierten Topic gesendet werden sollen.

Folgende private Methoden können zusätzlich vom Anwender überschrieben werden:

User_ClientConnected()	Wird aufgerufen, wenn der MQTT Client zum Server eine Verbindung erstellt hat.
User_ClientDisconnected()	Wird aufgerufen, wenn die Verbindung vom MQTT Client zum Server getrennt wurde.
User_ErrorOccured()	Wird aufgerufen, wenn in der Routine ein Fehler aufgetreten ist.
User_SubscribedOK()	Wird aufgerufen, wenn die Anmeldung zum definierten Topic erfolgreich war.
User_UnsubscribedOK()	Wird aufgerufen, wenn die Abmeldung vom definierten Topic, erfolgreich war.
User_PublishDone()	Wird aufgerufen, wenn die Daten zum definierten Topic, erfolgreich gesendet wurden.

Für weitere Information, siehe die Schnittstellen-Beschreibung.

Schnittstellen

Clients

ocMQTTClient_Interface	Objektkanal zur Klasse MQTTClient_Interface.	
	Datentyp	MQTTClient_Interface::t_e_MQTTClientStates
ocSigCLib	Objektkanal zur Klasse SigCLib. Muss nicht verbunden werden.	
	Datentyp	DINT

Server

State

Dieser Server kann zum Aufrufen der globalen Methoden der Klasse benutzt werden. Zeigt den aktuellen Schritt des internen Schrittschaltwerks der Routine an.

MTS_Init	Initialisierung des Topics
MTS _WaitForConnection	Es wird gewartet, bis der Client eine Verbindung zum Server aufgebaut hat.
MTS _Ready	Klasse ist bereit für den im Server TopicTyp definierten Vorgang
MTS _Subscribe	Es wird eine Anmeldung zum definierten Topic ausgeführt
MTS _WaitForSubscribe	Es wird auf die Rückmeldung zum Anmeldevorgang gewartet
MTS _Subscribed	Die Anmeldung war erfolgreich
MTS _Unsubscribe	Es wird eine Abmeldung vom definierten Topic ausgeführt
MTS _WaitForUnsubscribe	Es wird auf die Rückmeldung zum Abmeldevorgang gewartet
MTS _Unsubscribed	Die Abmeldung war erfolgreich
MTS _WaitForPublishInterval	Es wird auf die im Server PublishInterval definierte Zeitverzögerung gewartet, bevor Daten zum Server gesendet werden.
MTS _WaitForPublishDataChange	Es wird auf eine Änderung der Benutzerdefinierten Daten gewartet, bevor Daten zum Server gesendet werden.
MTS _WaitForPublishCommand	Es wird auf ein manuelles Anstoßen vom Benutzer gewartet, bevor Daten zum Server gesendet werden.
MTS _Publish	Es wird der Sendevorgang für die Benutzerdefinierten Daten angestoßen.
MTS _WaitForPublishDone	Es wird auf eine Rückmeldung zum Sendevorgang gewartet.
MTS _Error	Während der Routine ist ein Fehler aufgetreten.
MTS _Error_WaitForReset	Es wird darauf gewartet, dass der Anwender die Klasse aus dem Fehler zurücksetzt.

Einheit	-	Datentyp	t_e_MQTT_Topic State
Wertebereich	-	Write Protected	TRUE
Defaultwert	-	Retentive	FALSE

TopicType

Hier kann die Art, wie das MQTT Topic, welches im Server TopicName definiert wurde, behandelt werden soll.

MQTT_DEACTIVATED	Klasse ist deaktiviert. Wir sind weder an einem Topic angemeldet, noch werden Daten an ein Topic gesendet.
MQTT_SUB	Es soll eine Anmeldung zum definierten Topic erfolgen
MQTT_PUB_POLL	Zum definierten Topic soll in einem bestimmten Zeitintervall, welches man am Server PublishInterval einstellen kann, die benutzerdefinierten Daten gesendet werden.
MQTT_PUB_ONCHANGE	Wird eine Änderung der benutzerdefinierten Daten erkannt, dann sollen diese zum definierten Topic gesendet werden.
MQTT_PUB_MAN	Manuelles anstoßen zum Senden der Daten an das definierte Topic. Das Anstoßen kann durch den Aufruf der Methode DoManualPublish() angestoßen werden.

Einheit	-	Datentyp	t_e_MQTT_Topic Type
Wertebereich	0-4	Write Protected	FALSE
Defaultwert	einstellbar	Retentive	SRAM

TopicName

Hier muss der Name des Topics definiert werden.

Einheit	-	Datentyp	UDINT Objektkanal zur Klasse String-RAM
Wertebereich	-	Write Protected	FALSE
Defaultwert	-	Retentive	SRAM

PublishInterval

Ist der Topic Typ auf MQTT_PUB_POLL (Server TopicType) eingestellt, kann hier das Zeitintervall zum Senden der Daten definiert werden.

Einheit	[ms]	Datentyp	UDINT
Wertebereich	-	Write Protected	FALSE
Defaultwert	10 sek	Retentive	SRAM

RetainData

Hier wird definiert ob die Daten, welche zum Server zu einem bestimmten Topic gesendet werden (Publish), gespeichert werden sollen. Meldet sich ein neuer Client an dieses Topic an, so werden ihm sofort die letzten gespeicherten Daten gesendet. Ansonsten bekäme der neue Client nur Daten, sobald wieder jemand etwas auf dieses Topic sendet.

0…Daten werden nicht im Server gespeichert
1…Daten werden im Server gespeichert.

Einheit	-	Datentyp	UDINT
Wertebereich	0-1	Write Protected	FALSE
Defaultwert	0	Retentive	SRAM

QoS

Hier kann die „Quality of Service“ für die Datenübertragung eingestellt werden.
Diese bezieht sich immer zwischen Client und Server (Broker).

0…Das Senden der Daten erfolgt höchstens einmal. Der Empfänger bestätigt nicht den Empfang der Daten. Bietet dieselbe Garantie wie das darunter liegende TCP Protokoll.

1…Das Senden der Daten erfolgt mindestens einmal. Der Sender wartet auf eine Bestätigung des Empfängers. Erfolgt diese nicht in einer gewissen Zeit, so werden die Daten nochmal gesendet. Es besteht die Möglichkeit das die gleichen Daten öfter gesendet und empfangen werden.

2…Das Senden der Daten erfolgt exakt einmal. Hier ist sichergestellt, dass der Empfänger die Nachricht genau einmal bekommt.

Hinweis: Umso höher der „Quality of Service“ eingestellt ist, umso länger dauert auch die Abarbeitung der Sende- und Empfangsroutine.

Einheit	-	Datentyp	DINT
Wertebereich	0-2	Write Protected	FALSE
Defaultwert	0	Retentive	SRAM

GrantedQoS

Nur relevant im Topic Typ MQTT_SUB.
Hier wird die erteilte „Quality of Service“ zum angemeldeten (subscribed) Topic angezeigt. Diese wird vom Server gesetzt. Die hier angezeigte Qualität ist maximal die im Server QoS definierte, kann aber nach Einstellungen des MQTT Servers auch kleiner sein.

Einheit	-	Datentyp	DINT
Wertebereich	0-2	Write Protected	TRUE
Defaultwert	-	Retentive	FALSE

Globale Methoden

<table>
<tr><td>Init</td><td>Initialisierung der Klasse. Im ersten Init-Durchlauf werden die Timeouts für die Vorgänge Subscribe, Unsubscribe und Publish gesetzt.</td></tr>
<tr><td>CyWork</td><td>Aufruf der internen Routine.</td></tr>
<tr><td>PubSubData_Callback</td><td>Callback Methode für die MQTTClient_Interface Klasse für verschiedene Vorgänge. Wird bei den Vorgängen Subscribe, Unsubscribe und Publish übergeben.
<table>
<tr><td>► pThis</td><td>Pointer auf das Objekt</td></tr>
<tr><td>► MsgType</td><td>Art der Nachricht:
<table>
<tr><td>PSRC_Subscribed</td><td>Rückmeldung zum Anmeldevorgang.</td></tr>
<tr><td>PSRC_Unsubscribed</td><td>Rückmeldung zum Abmeldevorgang</td></tr>
<tr><td>PSRC_PublishReceived</td><td>Daten zu einem angemeldeten Topic empfangen</td></tr>
<tr><td>PSRC_Published</td><td>Rückmeldung zu einem Sendevorgang</td></tr>
</table></td></tr>
<tr><td>► iMid</td><td>ID des Datenpakets</td></tr>
<tr><td>► iGrantedQoS</td><td>Erteilte QoS vom Server</td></tr>
<tr><td>► pMessage</td><td>Pointer auf die Daten</td></tr>
</table></td></tr>
<tr><td>PubSubData</td><td>Wird von der Methode PubSubData_Callback aufgerufen.
<table>
<tr><td>► MsgType</td><td>Art der Nachricht:
<table>
<tr><td>PSRC_Subscribed</td><td>Rückmeldung zum Anmeldevorgang.</td></tr>
<tr><td>PSRC_Unsubscribed</td><td>Rückmeldung zum Abmeldevorgang</td></tr>
<tr><td>PSRC_PublishReceived</td><td>Daten zu einem angemeldeten Topic empfangen</td></tr>
<tr><td>PSRC_Published</td><td>Rückmeldung zu einem Sendevorgang</td></tr>
</table></td></tr>
<tr><td>► iMid</td><td>ID des Datenpakets</td></tr>
<tr><td>► iGrantedQoS</td><td>Erteilte QoS vom Server</td></tr>
<tr><td>► pMessage</td><td>Pointer auf die Daten</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>DoManualPublish</td><td>Ist der Topic Typ MQTT_PUB_MAN eingestellt, kann durch den Aufruf dieser Methode das senden der Daten zum Server angestoßen werden.

<table><tr><td>◄ outSuccess</td><td>True…Manuelles Senden wurde angestoßen
False…Manuelles Senden konnte nicht angestoßen werden.</td></tr></table></td></tr>
<tr><td>SetTimeouts</td><td>Mit dieser Methode können die Timeouts für die Subscribe,Unsubscribe und Publish Vorgänge gesetzt werden.

<table><tr><td>► inTimeInfo</td><td>Datentypstruktur t_s_MQTT_TopicTimeouts
Elemente:<table><tr><td>SubscribingTimeout</td><td>Timeout für den Anmeldevorgang [ms]</td></tr><tr><td>UnsubscribingTimeout</td><td>Timeout für den Abmeldevorgang [ms]</td></tr><tr><td>PublishTimeout</td><td>Timeout für das Senden der Daten [ms]</td></tr></table></td></tr></table></td></tr>
<tr><td>ResetError</td><td>Im Falle das ein Fehler auftritt und der Server State auf „MTS _Error_WaitForReset“ steht, kann mit dieser Methode der Fehler wieder zurückgesetzt werden.</td></tr>
<tr><td>WriteTopicName</td><td>Mit dieser Methode kann der Topic Name geändert werden.

<table><tr><td>► inPtrTopicString</td><td>Zeiger auf den String mit dem neuen Topic Namen.</td></tr><tr><td>◄ outSuccess</td><td>True…Ändern des Topic Namens erfolgreich
False…Ändern des Topic Namens fehlgeschlagen</td></tr></table></td></tr>
<tr><td>ReadTopicName</td><td>Mit dieser Methode kann der aktuelle Topic Name gelesen werden.

<table><tr><td>► inDstPtrTopicString</td><td>Zeiger auf den Zielstring, wo der Name gespeichert werden soll.</td></tr><tr><td>► inLenOfTopicString</td><td>Länge des Zielstrings</td></tr><tr><td>◄ outSuccess</td><td>True…Name wurde erfolgreich kopiert
False…Kopieren des Namens fehlgeschlagen</td></tr></table></td></tr>
<tr><td>GetTopicNameLength</td><td>Liefert die Länge des Topic Namens zurück. Die 0-Terminierung ist nicht inkludiert.

<table><tr><td>◄ outTopicLength</td><td>Länge des Topic Namens ohne 0-Terminierung.</td></tr></table></td></tr>
</table>

Config_SetParameter	Diese Methode kann verwendet werden, um Parameter in der MQTTTopic-Klasse zu konfigurieren, eine Liste der Parameter finden Sie in der Beschreibung der Aufzählung.	
	► Parameter	Aufzählungswert, der den zu setzenden Parameter angibt.
	► Value	Der Wert, der für den gewählten Parameter eingestellt werden soll.
	◄ retCode	Das Ergebnis der Wertsetzoperation, dieser Wert unterscheidet sich je nach gewähltem Parameter.

Private Methoden

Hier werden nur die für den Anwender relevanten Methoden beschrieben.

<table>
<tr><td>User_ClientConnected</td><td>Wird aufgerufen, wenn der MQTT Client eine Verbindung zum Server aufbaut hat. Kann vom Benutzer überschrieben werden.</td></tr>
<tr><td>User_ClientDisconnected</td><td>Wird aufgerufen, wenn die Verbindung zwischen MQTT Client und Server getrennt wurde. Kann vom Benutzer überschrieben werden.</td></tr>
<tr><td>User_ErrorOccured</td><td>Wird aufgerufen, wenn ein Fehler in der internen Routine auftritt. Kann vom Benutzer überschrieben werden.
Tritt ein Fehler auf kann zusätzlich die Variable „ErrorOccuredState“ überprüft werden, um den Schritt der internen Routine, in der der Fehler auftrat, zu ermitteln.</td></tr>
<tr><td>User_SubscribedOK</td><td>Wird aufgerufen, wenn die Anmeldung zum definierten Topic erfolgreich war. Kann vom Benutzer überschrieben werden.
<table><tr><td>► pData</td><td>Zeiger auf den Payload der MQTT Nachricht, falls Daten vorhanden sind. Ansonsten ist der Zeiger NIL.</td></tr><tr><td>► SizeOfData</td><td>Größe des Payload in Byte.</td></tr></table></td></tr>
<tr><td>User_UnsubscribedOK</td><td>Wird aufgerufen, wenn die Abmeldung vom definierten Topic, erfolgreich war. Kann vom Benutzer überschrieben werden.
<table><tr><td>► pData</td><td>Zeiger auf den Payload der MQTT Nachricht, falls Daten vorhanden sind. Ansonsten ist der Zeiger NIL.</td></tr><tr><td>► SizeOfData</td><td>Größe des Payload in Byte.</td></tr></table></td></tr>
<tr><td>User_GetPublishData</td><td>Wird aufgerufen, wenn Daten zum definierten Topic gesendet werden sollen. Muss vom Anwender überschrieben werden, um seine spezifischen Daten zu übergeben.
<table><tr><td>◄ outPublishData</td><td>Datentypstruktur t_s_MQTT_TopicPublishData
Elemente:<table><tr><td>PointerToData</td><td>Zeiger auf die Daten, die gesendet werden sollen.</td></tr><tr><td>SizeOfData</td><td>Größe der Daten in Byte.</td></tr></table></td></tr></table></td></tr>
</table>

<table>
<tr><td>User_PublishDone</td><td>Wird aufgerufen, wenn die Daten zum definierten Topic, erfolgreich gesendet wurden. Kann vom Benutzer überschrieben werden.

<table><tr><td>► pData</td><td>Zeiger auf den Payload der MQTT Nachricht, falls Daten vorhanden sind. Ansonsten ist der Zeiger NIL.</td></tr><tr><td>► SizeOfData</td><td>Größe des Payload in Byte.</td></tr></table></td></tr>
<tr><td>User_SubscribeData</td><td>Wird aufgerufen, wenn Daten vom angemeldeten Topic empfangen wurden.
Muss vom Anwender überschrieben werden, um die Daten zu verwerten.

Hinweis: Die übergebenen Parameter sind nur im aktuellen Aufruf der Methode gültig.

<table><tr><td>► pData</td><td>Zeiger auf den Payload der MQTT Nachricht, falls Daten vorhanden sind. Ansonsten ist der Zeiger NIL.</td></tr><tr><td>► SizeOfData</td><td>Größe des Payload in Byte.</td></tr></table></td></tr>
</table>

Aufzählungen

t_e_ConfigParameters:

CP_SendAtStart	Wenn TopicType = MT_Publish_OnChange: Dieser Wert gibt an, ob das Topic beim Start gesendet wird oder ob sich der Wert zwischen dem Zeitpunkt der letzten Übertragung und dem Start geändert hat. Das bedeutet, wenn aktiv, werden die Daten immer gesendet, wenn das Thema neu initialisiert wird (auch bei einem Neustart der Anwendung), wenn nicht aktiv, wartet die Klasse auf die erste Änderung, bevor die Daten gesendet werden. 0 = Inaktiv (Default) 1 = Aktiv