

# TCPCommunication

TCPCommunication	
TCPCommunication1	
coDataMng	ClassSvr
0	0
coIntern	InitError
0	0
coIpToConnect	ConnectionType
0	0
coStationToConnect	ComPort
0	500
coLogFilter	Online
0	0
coStdLib	ErrorSend
0	0
coMultiTask	ErrorRecive
0	0
coSigCLib	TransferRateSend
0	0
ErrRuntime	TransferRateReceive
0	0
	TimeLastSend
	0
	TimeLastReceive
	0
	LogEnable
	0
	ResetLogBuffer
	0
	TaskName
	0
	DeactivateCom
	0

Die Klasse dient dem Datenaustausch zwischen zwei CPUs bzw. Terminals.

Kann verwendet werden, um Daten zwischen zwei CPUs zu verschicken.

Die Klasse alleine ist nicht funktionsfähig. Sie benötigt die Klasse DataManager, um korrekt zu funktionieren.

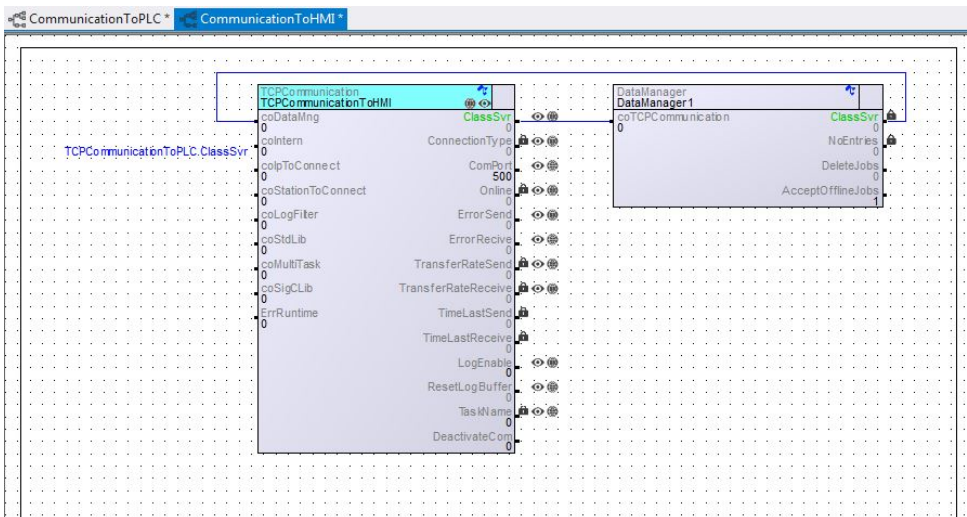
Je nachdem welche Clients verbunden werden, arbeitet die Klasse als

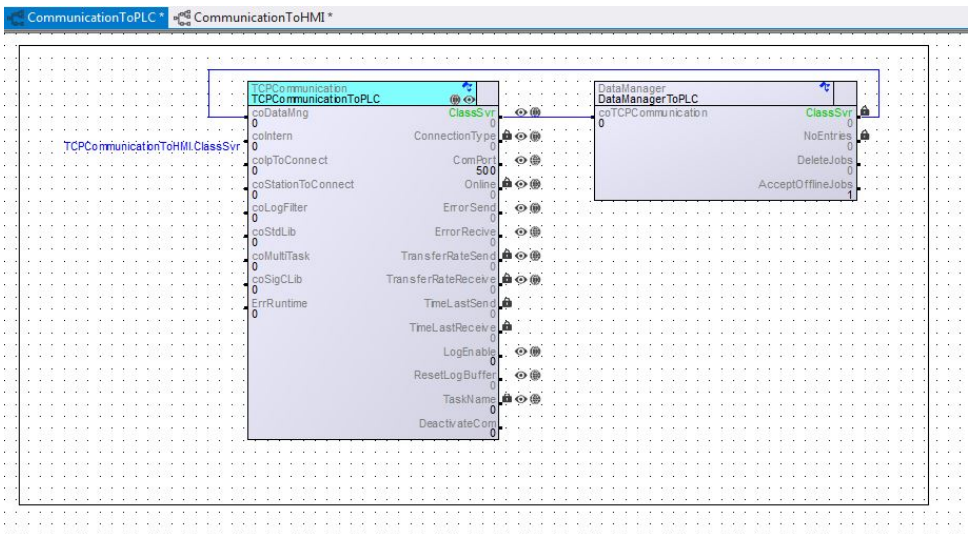
- TCPIP Client
- TCPIP Server
- interne Verbindung

## Interne Kommunikation aufbauen

Um eine interne Verbindung (1-CPU Lösung) herzustellen, müssen die Clients der beiden Objekte mit den ClassSvr der Gegenseite verbunden werden. Will man eine externe Kommunikation aufbauen, dürfen diese Clients nicht verbunden werden.

HMI- und PLC-Teil befinden sich auf einem Prozessor:





## Einrichten eines TCPIP-Clients bzw. Servers

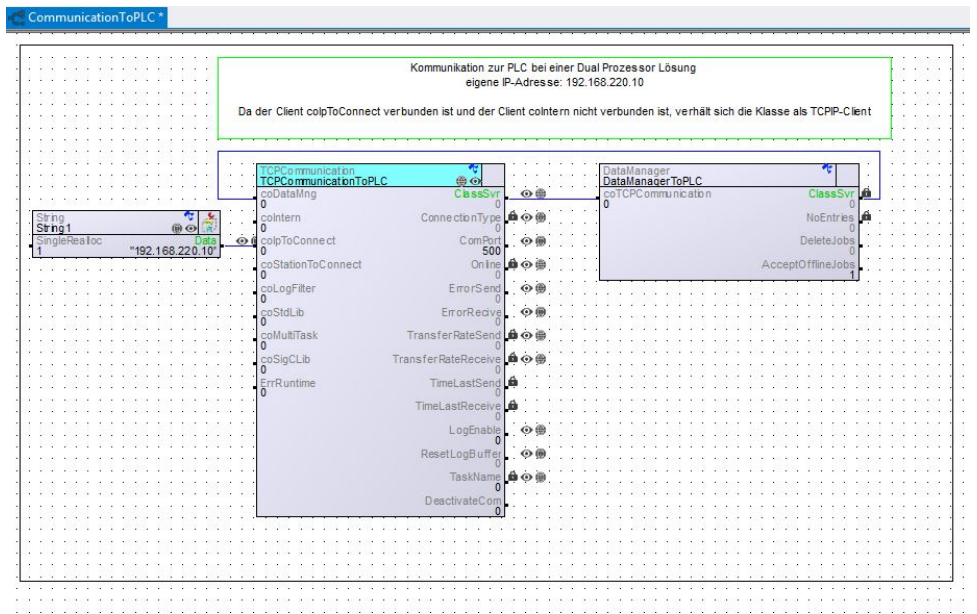
Verbindet man den Client `colpToConnect` mit einem Stringobjekt, welches die IP-Adresse des TCPIP-Servers beinhaltet, verhält sich dieser Block der Kommunikation als TCPIP-Client. Zu beachten ist, dass der Client `colIntern` bei einer 2 CPU-Lösung nicht verbunden werden darf.

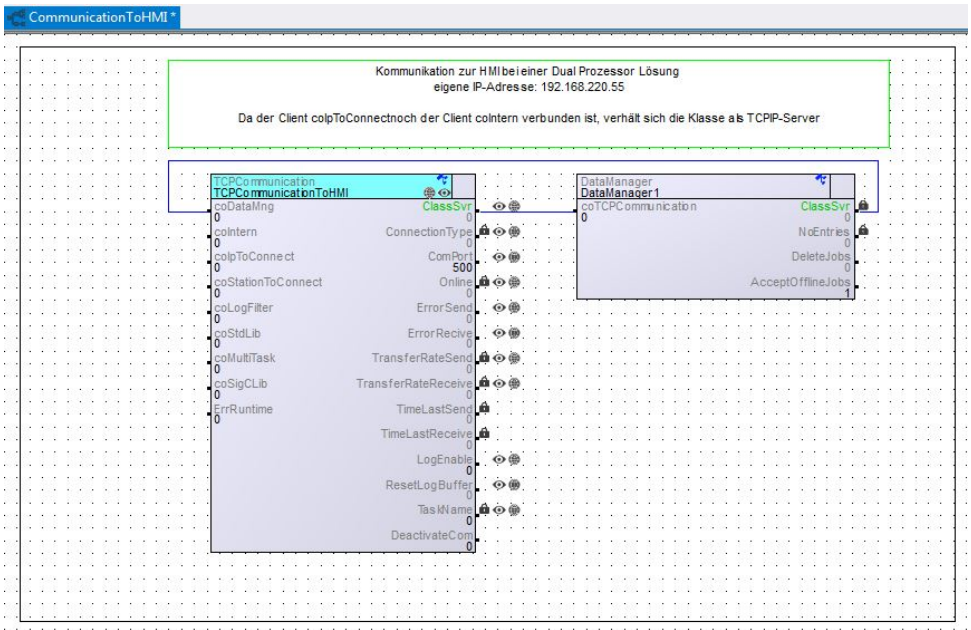
Möchte man, dass das Objekt als TCPIP-Server fungiert, müssen keine Verbindungen an den Clients `colpToConnect` und `colIntern` erstellt werden.

Beispiel:

HMI-Teil befindet sich auf dem Prozessor mit der IP 192.168.220.10

PLC-Teil befindet sich auf dem Prozessor mit der IP 192.168.220.55





## Schnittstellen

### Clients

<b>coDataManager</b>	Verbindung zum Daten-Manager. Objekt Kanal: DataManager (über diesen Client werden die zu sendenden Daten vom Manager ausgelesen)
	Datentyp: Objektkanal zur DataManager
<b>colIntern</b>	Ist bei interner Verbindung (Singleprozessorslösung) mit dem anderen TCPCommunication-Objekt zu verbinden (auf Server ClassSvr).
	Datentyp: Objektkanal zur <a href="#">TCPCommunication</a>
<b>colpToConnect</b>	Wird benötigt, wenn die Klasse als TCPIP-Client arbeiten soll. Verbindung auf ein Objekt der Klasse String oder StringRAM (IP-Adresse des TCPIP-Servers auf den man sich verbinden will).
	Datentyp: UDINT
<b>coLogFilter</b>	Verbindung zur Klasse TCPCommunicationLogFilter.
	Datentyp: Objektkanal zur TCPCommunicationLogFilter

<b>coStdLib</b>	Verbindung zur Betriebssystemschnittstelle _StdLib (wird automatisch erstellt).
Datentyp	Objektkanal zur _StdLib
<b>coMultiTask</b>	Verbindung zur Betriebssystemschnittstelle _MultiTask (wird automatisch erstellt).
Datentyp	Objektkanal zur _MultiTask
<b>coSigCLib</b>	Verbindung zur Betriebssystemschnittstelle SigCLib (wird automatisch erstellt).
Datentyp	Objektkanal zur SigCLib
<b>ErrRuntime</b>	Wird der parallele Kommunikationstask zu lange nicht aufgerufen, wird hier ein Fehler ausgegeben.
Datentyp	DINT
<b>SizeOfComBuffers</b>	Größe von sowohl Rx als auch Tx Puffer in Bytes.
Datentyp	DINT
<b>CyclicTaskInterval</b>	Konfiguration, welcher Task bei einer internen Verbindung verwendet werden soll 0 paralleler Kommunikationstask wird angelegt >0 Kommunikation wird im Cyclic-Task behandelt Hier ist auf die eingestellte Cyclic-Time zu achten! (Standard = 100 ms)
Datentyp	DINT






## Server

ClassSvr	Objekt-Server der Klasse. Der Client coTCPCommunication des DataManager muss hierher verbunden werden. Die Klasse TCPCommunicationLogViewer kann ebenfalls hierher verbunden werden, um die Logeinträge anzuzeigen und zu exportieren.			
	Einheit	-	Datentyp	DINT
	Wertebereich	-	Write Protected	FALSE
	Defaultwert	-	Retentive	FALSE
InitError	Ist dieser Server ungleich 0, ist während der Initialisierung der Klasse ein schwerwiegender Fehler aufgetreten. Die Klasse ist nicht funktionsfähig. -1 Paralleler Task konnte nicht gestartet werden -2 Ein benötigtes CIL-Interface ist nicht verfügbar -3 Benötigter Speicher nicht verfügbar			
	Einheit	-	Datentyp	DINT
	Wertebereich	-3 ... 0	Write Protected	TRUE
	Defaultwert	-	Retentive	FALSE
ConnectionType	Zeigt die Art der konfigurierten Verbindung an.			
	Einheit	-	Datentyp	t_e_ConnectionType
	Wertebereich	0-4	Write Protected	TRUE
	Defaultwert	-	Retentive	FALSE
ComPort	Hier muss der TCP-Port über welchen kommuniziert werden soll, eingetragen werden. Beim Senden und Empfangen muss der gleiche Port angegeben werden.			

	Wird mit mehreren CPUs kommuniziert, müssen unterschiedliche Ports für die unterschiedlichen CPUs vergeben werden.				
	Einheit	-	Datentyp	_FSM_TCP_USER	
	Wertebereich	max. UDINT	Write Protected	TRUE	
	Defaultwert	einstellbar	Retentive	FALSE	
<b>Online</b>	Status der Online Verbindung. 0        Offline 1        Online				
	Einheit	-	Datentyp	DINT	
	Wertebereich	0-1	Write Protected	TRUE	
	Defaultwert	-	Retentive	FALSE	
<b>ErrorSend</b>	Wird inkrementiert, sollte beim Senden ein Fehler auftreten.				
	Einheit	-	Datentyp	DINT	
	Wertebereich	max. DINT	Write Protected	FALSE	
	Defaultwert	-	Retentive	FALSE	
<b>ErrorReceive</b>	Wird inkrementiert, sollte beim Empfangen ein Fehler auftreten.				
	Einheit	-	Datentyp	_FSM_TCP_USER	
	Wertebereich	max. DINT	Write Protected	TRUE	
	Defaultwert	-	Retentive	FALSE	
<b>TransferRateSend</b>	Geschwindigkeit der Kommunikation in [Bit/s] (Upload)				
	Einheit	Bit/s	Datentyp	DINT	
	Wertebereich	-	Write Protected	TRUE	
	Defaultwert	-	Retentive	FALSE	
<b>TransferRateReceive</b>	Geschwindigkeit der Kommunikation in [Bit/s] (Download)				
	Einheit	Bit/s	Datentyp	DINT	
	Wertebereich	-	Write Protected	TRUE	
	Defaultwert	-	Retentive	FALSE	
<b>TimeLastSend</b>	Zeit, welche das zuletzt gesendete TCP-Paket benötigt hat [µs].				
	Einheit	µs	Datentyp	UDINT	
	Wertebereich	-	Write Protected	TRUE	
	Defaultwert	-	Retentive	FALSE	
<b>TimeLastReceive</b>	Zeit, welche das zuletzt empfangene TCP-Paket benötigt hat, bis es vollständig empfangen wurde [µs].				
	Einheit	µs	Datentyp	UDINT	
	Wertebereich	-	Write Protected	TRUE	
	Defaultwert	-	Retentive	FALSE	
<b>LogEnable</b>	Hiermit kann das Logging aktiviert werden. Damit die Klasse TCPCommunicationLogViewer die Daten in ein Excel File exportieren kann, muss das Logging deaktiviert werden. 0        Logging ist deaktiviert 1        Logging ist aktiv				
	Einheit	-	Datentyp	DINT	

	Wertebereich	0-1	Write Protected	FALSE
	Defaultwert	-	Retentive	FALSE
<b>ResetLogBuffer</b>	Durch das Schreiben auf diesen Server wird der interne Log Buffer gelöscht. Kann nur ausgeführt werden, wenn der Server LogEnable auf 0 steht.			
	Einheit	-	Datentyp	DINT
	Wertebereich	-	Write Protected	FALSE
	Defaultwert	-	Retentive	FALSE
<b>TaskName</b>	Über diesen Server können String Funktionen benutzt werden			
	Einheit	-	Datentyp	UDINT
	Wertebereich	-	Write Protected	TRUE
	Defaultwert	0	Retentive	FALSE
<b>DeactivateCom</b>	Hier kann die gesamte Kommunikation deaktiviert werden. DeactivateCom.Write(input := 1); // Kommunikation wird deaktiviert DeactivateCom.Write(input := 0); // Kommunikation wird wieder aktiviert			
	Einheit	-	Datentyp	DINT
	Wertebereich	0-1	Write Protected	FALSE
	Defaultwert	0	Retentive	FALSE

## Globale Methoden

<b>Init</b>	Nicht für den Benutzer.
<b>GetOnlineState</b>	<p>Hier kann der Status der TCP/IP-Verbindung ausgelesen werden.</p> <p>  keine   retcode </p> <p> 0            Offline  1            Online </p>
<b>SetParameter</b>	<p>Funktion zum Setzen diverser Kommunikationsparameter.</p> <p>  ParaNo Parameternummer   ParaVal Wert, der eingestellt werden soll   retcode </p> <p> 0            Wert erfolgreich geändert  -1001       Ungültige Parameternummer </p> <p>Konstanten für Parameternummern:</p> <p> TCPCom_ParaWR_ComPort            Kommunikationsport  TCPCom_ParaWR_AliveSignalTime    Zeitintervall für das Senden von Alive Paketen  TCPCom_ParaWR_MissingAliveError    Fehlerschwelle für verlorene Alive Pakete: Nach X fehlenden Paketen wird ein Fehler ausgelöst und die Verbindung neu aufgebaut. </p>



	TCPCom_ParaWR_TaskPriority Task-Priorität für den TCP Task (muss nach dem 1. und vor dem letzten Init() Durchlauf gesetzt werden!)
<b>ReadParameter</b>	<p>Funktion zum Auslesen diverser Kommunikationsparameter.</p> <ul style="list-style-type: none"> <li>▶ ParaNo immer TCPCom_ParaRD_ComPort</li> <li>▶ pParaVal auf diese Adresse wird der Wert geschrieben</li> <li>◀ retcode                         <ul style="list-style-type: none"> <li>0 Wert erfolgreich geändert</li> <li>-1001 Ungültige Parameternummer</li> <li>-1002 für pParaVal wurde ein NIL Pointer übergeben</li> </ul> </li> </ul> <p>Konstanten für Parameternummern:</p> <ul style="list-style-type: none"> <li>TCPCom_ParaRD_ComPort Kommunikationsport</li> <li>TCPCom_ParaRD_AliveSignalTime Zeitintervall für das Senden von Alive Paketen</li> <li>TCPCom_ParaRD_MissingAliveError Fehlerschwelle für verlorene Alive Pakete: Nach X fehlenden Paketen wird ein Fehler ausgelöst und die Verbindung neu aufgebaut.</li> <li>TCPCom_ParaRD_TaskPriority Task-Priorität für den TCP Task</li> </ul>

## Beispiel für eine 3-Prozessor-Lösung

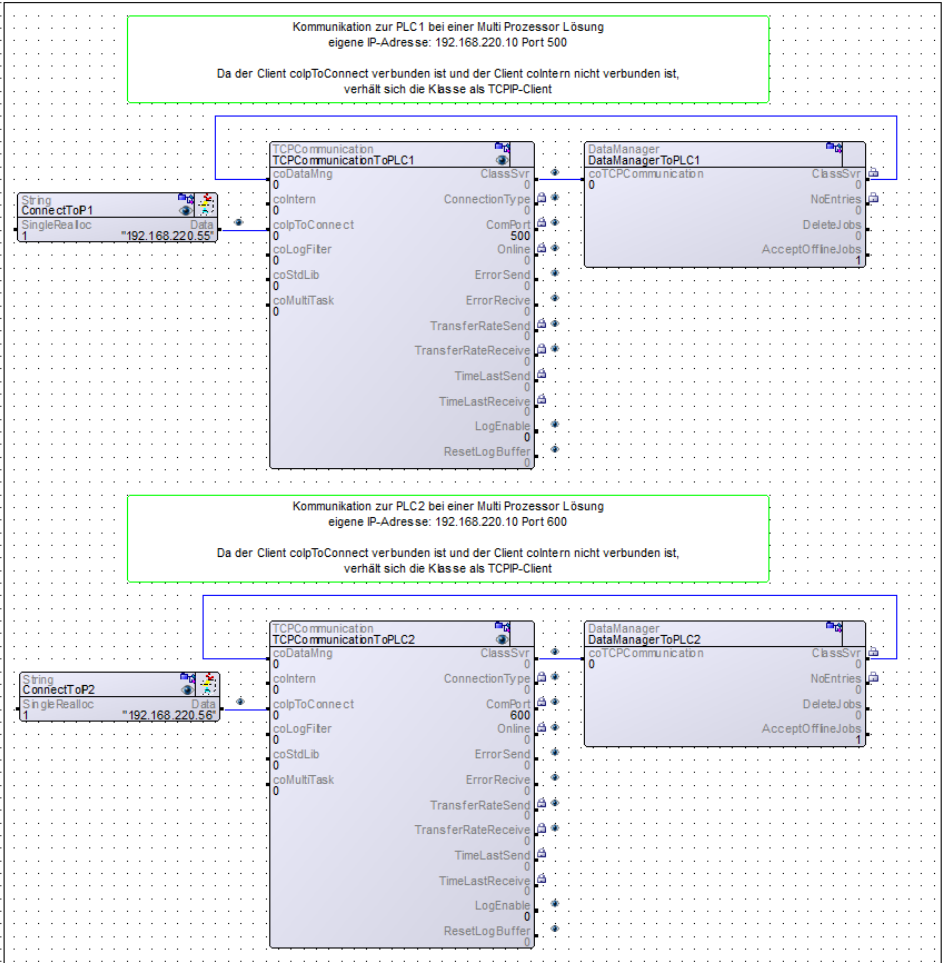
HMI-Teil befindet sich auf dem Prozessor mit der IP 192.168.220.10

PLC1-Teil befindet sich auf dem Prozessor mit der IP 192.168.220.55

PLC2-Teil befindet sich auf dem Prozessor mit der IP 192.168.220.56

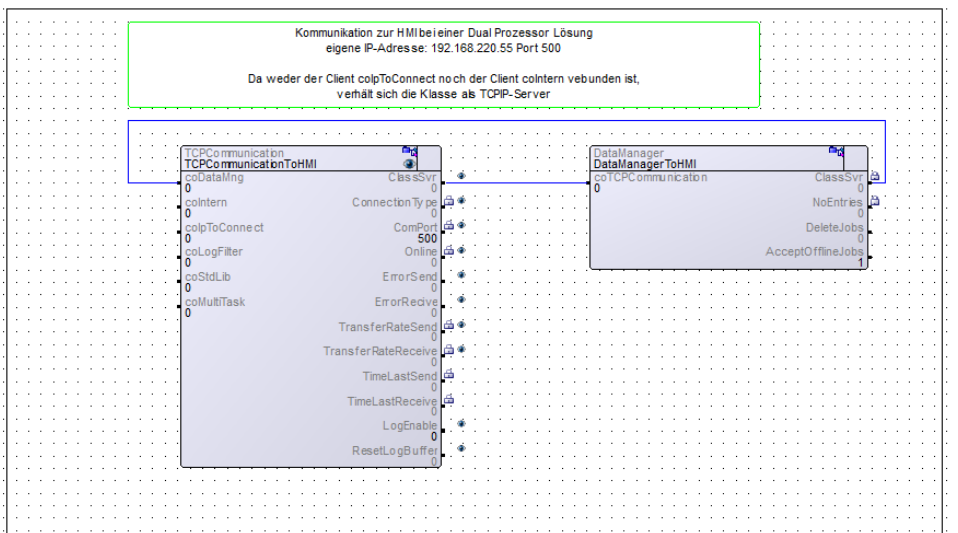
HMI-Teil

CommunicationToPLC \*



## PLC1-Teil

CommunicationToHMI \*



PLC2-Teil

CommunicationToHMI

