

# TCPCCommunication

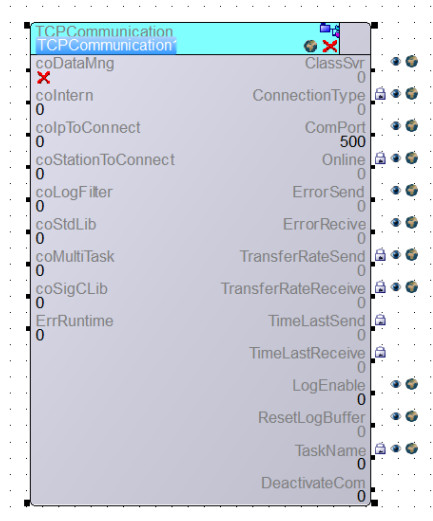
Die Klasse dient dem Datenaustausch zwischen zwei CPUs bzw. Terminals.

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



## Schnittstellen

### Clients

<b>coDataManager</b>	Verbindung zum Daten-Manager. Objekt Kanal: DataManager (über diesen Client werden die zu sendenden Daten vom Manager ausgelesen)
<b>coIntern</b>	Ist bei interner Verbindung (Singleprozessorrösung) mit dem anderen TCPCCommunication-Objekt zu verbinden (auf Server ClassSvr).
<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).
<b>coLogFilter</b>	Verbindung zur Klasse TCPCCommunicationLogFilter.
<b>coStdLib</b>	Verbindung zur Betriebssystemschnittstelle _StdLib (wird automatisch erstellt).
<b>coMultiTask</b>	Verbindung zur Betriebssystemschnittstelle _MultiTask (wird automatisch erstellt).
<b>coSigCLib</b>	Verbindung zur Betriebssystemschnittstelle SigCLib (wird automatisch erstellt).
<b>ErrRuntime</b>	Wird der parallele Kommunikationstask zu lange nicht aufgerufen, wird hier ein Fehler ausgegeben.

## Server

<b>ClassSvr</b>	Objekt-Server der Klasse. Der Client coTCPCommunication des DataManager muss hierher verbunden werden. Die Klasse TCPCommunicationLogViewer kann ebenfalls hierher verbunden werden, um die Loggeinträge anzuzeigen und zu exportieren.
<b>ConnectionType</b>	Zeigt die Art der konfigurierten Verbindung an.
<b>ComPort</b>	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 CPU's kommuniziert, müssen unterschiedliche Ports für die unterschiedlichen CPU's vergeben werden.
<b>Online</b>	Status der Online Verbindung . 0 ... Offline 1 ... Online
<b>ErrorSend</b>	Wird inkrementiert, sollte beim Senden ein Fehler auftreten.
<b>ErrorRecv</b>	Wird inkrementiert, sollte beim Empfangen ein Fehler auftreten.
<b>TransferRateSend</b>	Geschwindigkeit der Kommunikation in [Bit/s] (Upload)
<b>TransferRateRecv</b>	Geschwindigkeit der Kommunikation in [Bit/s] (Download)
<b>TimeLastSend</b>	Zeit, welche das zuletzt gesendete TCP-Paket benötigt hat [us].
<b>TimeLastRecv</b>	Zeit, welche das zuletzt empfangene TCP-Paket benötigt hat, bis es vollständig empfangen wurde [us].
<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
<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.
<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

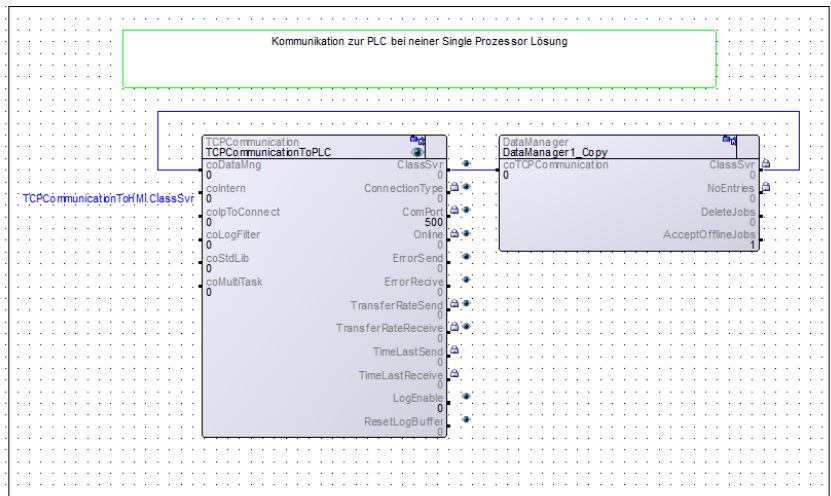
## Globale Methoden

<b>Init</b>	Nicht für den Benutzer.
<b>GetOnlineState</b>	<p>Hier kann der Status der TCP/IP-Verbindung ausgelesen werden.</p> <p><i>Parameter:</i> ► keine                    ◄ retcode 0 ... Offline                                    1 ... Online</p>
<b>SetParameter</b>	<p>Über diese Funktion kann der Kommunikationsport eingestellt werden.</p> <p><i>Parameter:</i> ► ParaNo: immer TCPCom_ParaWR_ComPort                    ► ParaVal: Wert, der eingestellt werden soll                    ◄ retcode:     0 ... Wert erfolgreich geändert                                    -1001 ... Ungültige Parameternummer</p>
<b>ReadParameter</b>	<p>Über diese Funktion kann der Kommunikationsport abgefragt werden.</p> <p><i>Parameter:</i> ► ParaNo: immer TCPCom_ParaWR_ComPort                    ► pParaVal: auf diese Adresse wird der Wert geschrieben                    ◄ retcode:     0 ... Wert erfolgreich geändert                                    -1001 ... Ungültige Parameternummer                                    -1002 ... für pParaVal wurde ein NIL Pointer übergeben</p>

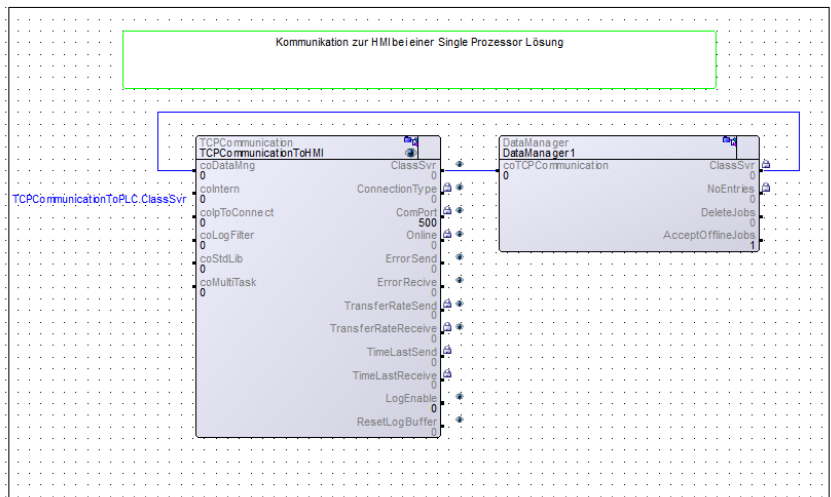
## Beispiel für eine 1-Prozessor-Lösung

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

CommunicationToHMI \* CommunicationToPLC \*



CommunicationToHMI \* CommunicationToPLC \*



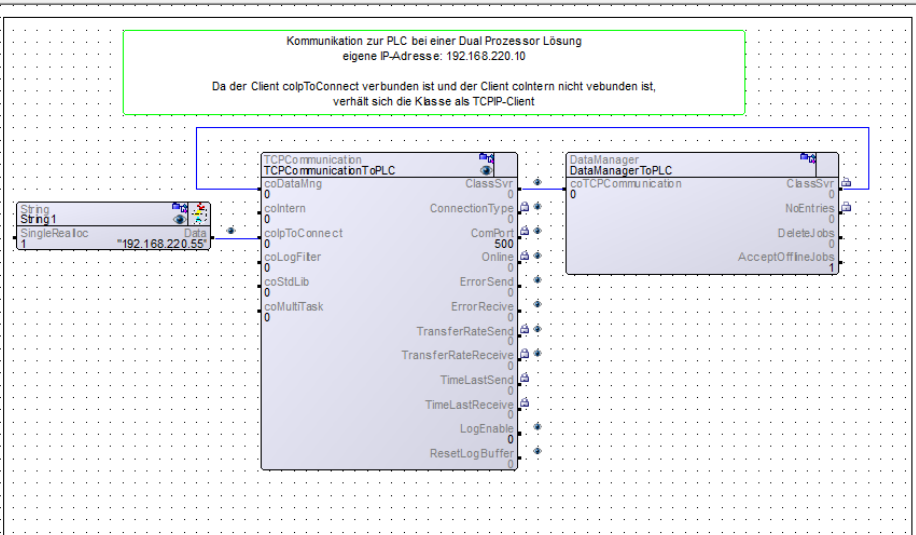
## Beispiel für eine 2-Prozessor-Lösung

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

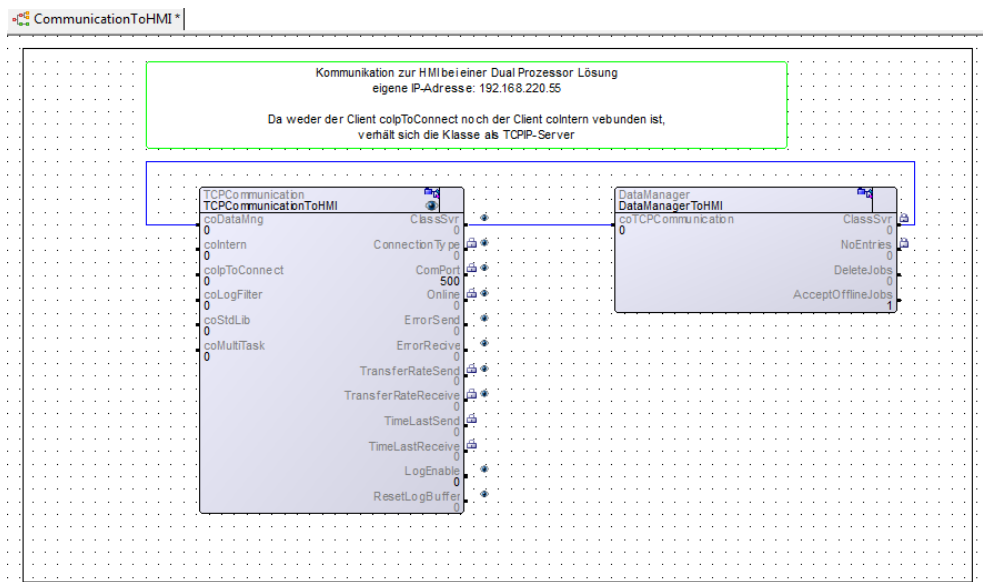
PLC-Teil befindet sich auf dem Prozessor mit der IP 192.168.220.55

HMI-Teil:

CommunicationToPLC\*



## PLC-Teil:



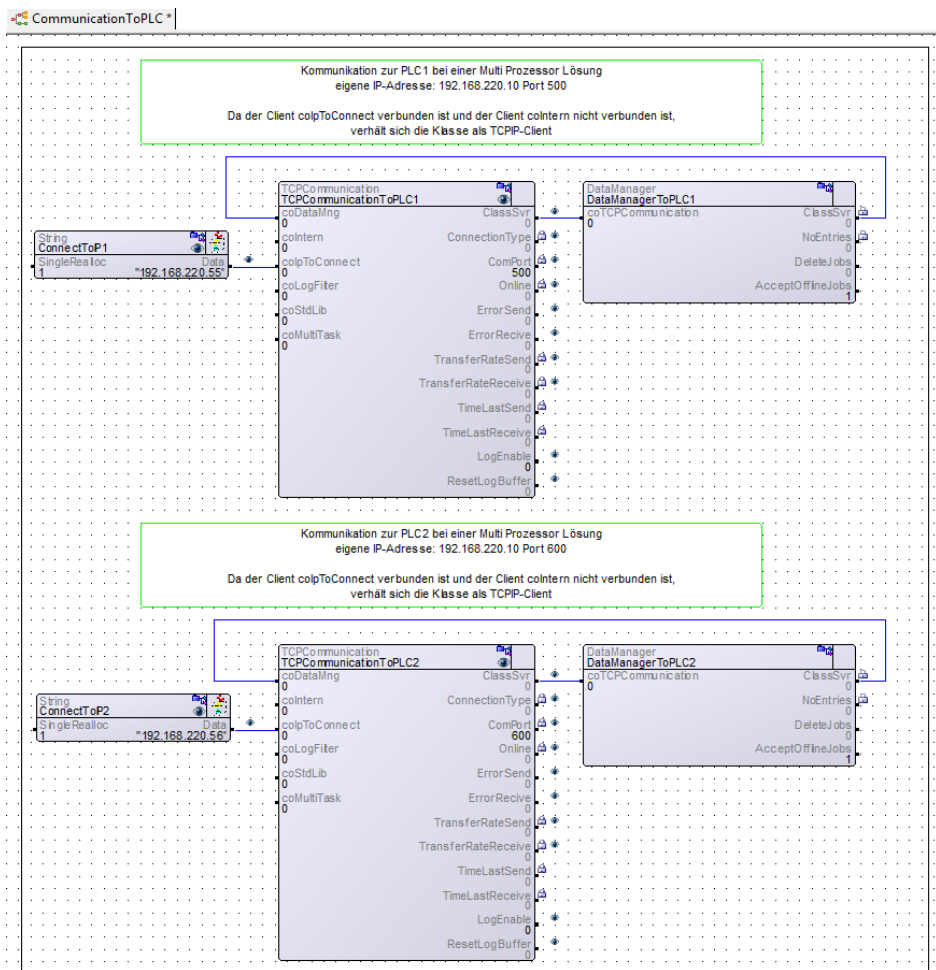
## Beispiel für eine 3-Prozessor-Lösung

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

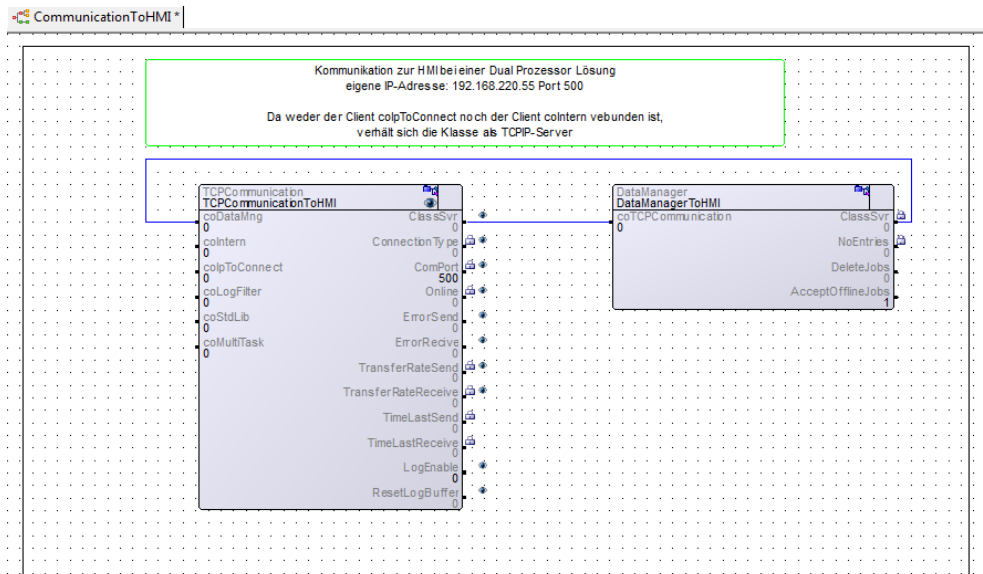
PLC1-Teil befindet sich auf dem Prozessor mit der IP 192.168.220.55

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

HMI-Teil:



## PLC1-Teil:





## PLC2-Teil:

