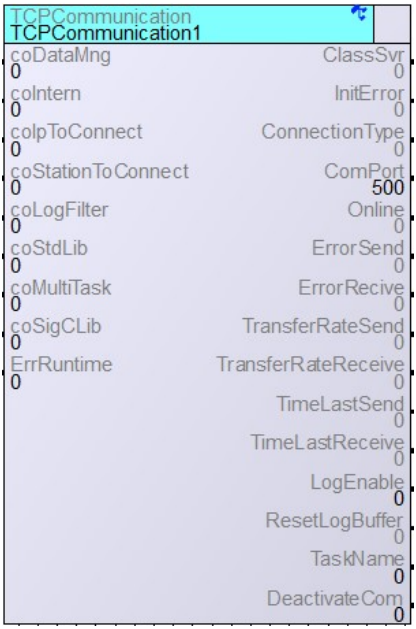
TCPCommunication



The class is used to exchange data between two CPUs or terminals.

Can be used to send data between two CPUs.

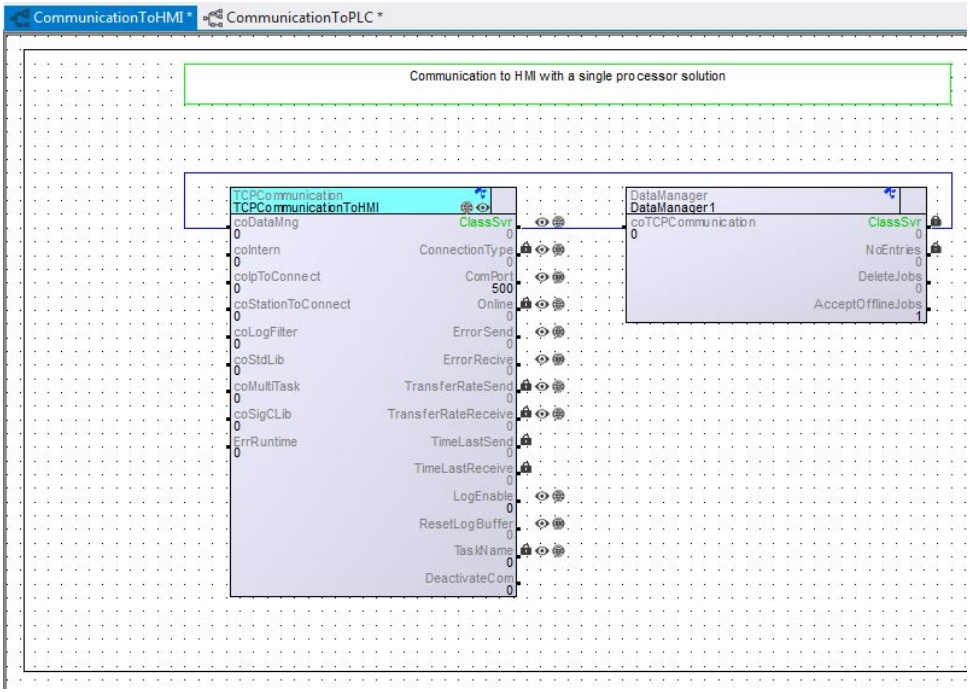
The class cannot operate independently, the DataManager class is required for correct function.

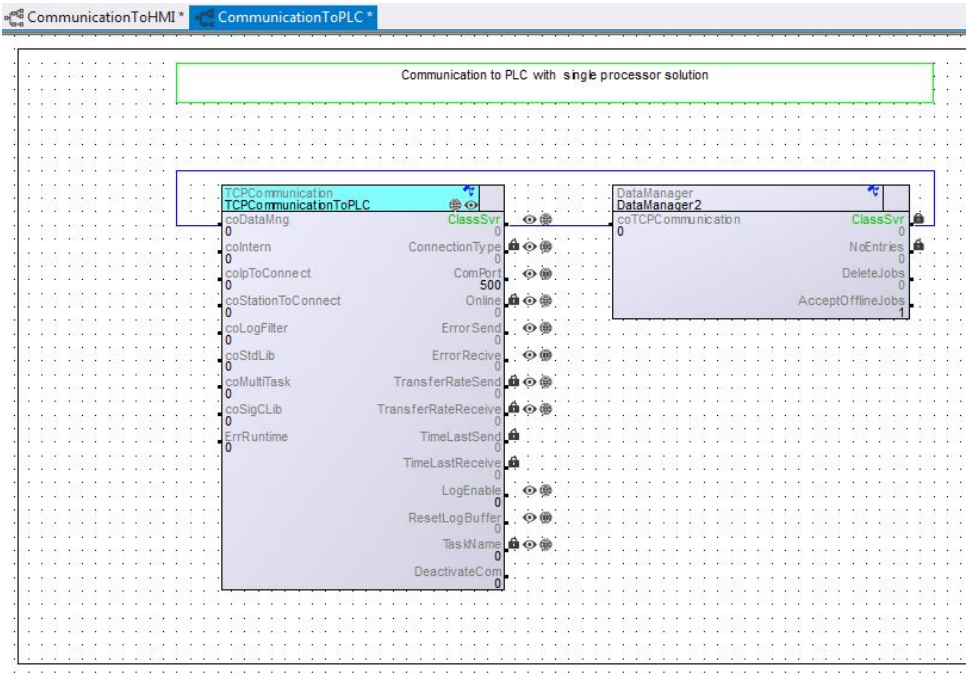
Depending on the clients connected, the class operates as
- TCPIP client
- TCPIP server
- Internal connection

Establishing Internal Communication

In order to establish an internal communication (1 CPU solution), the clients of the two objects have to be connected to the ClassSvr of the other side. For establishing an external communication, these clients must not be connected.

HMI and PLC component are on the same processor:



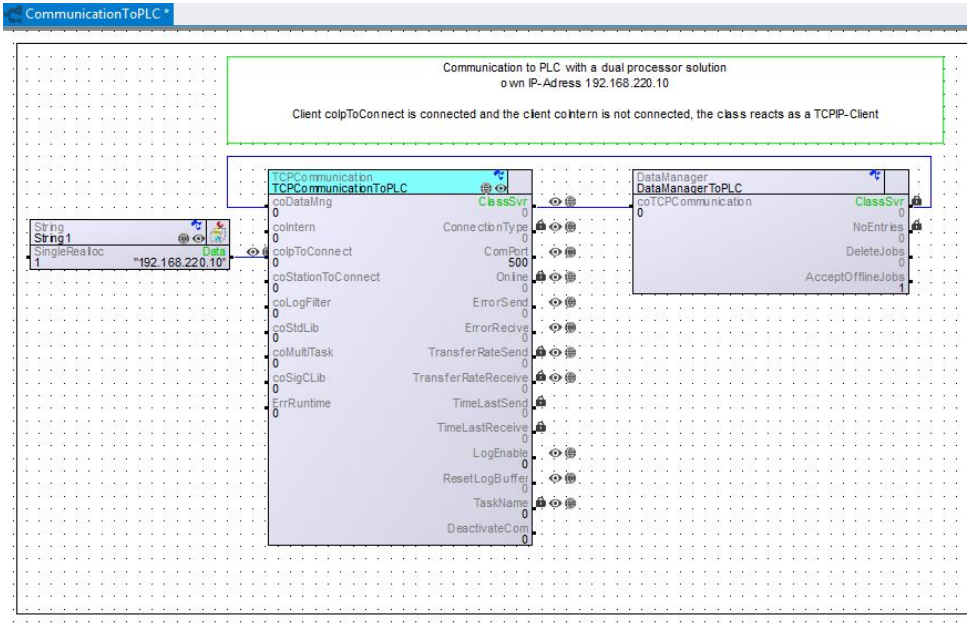


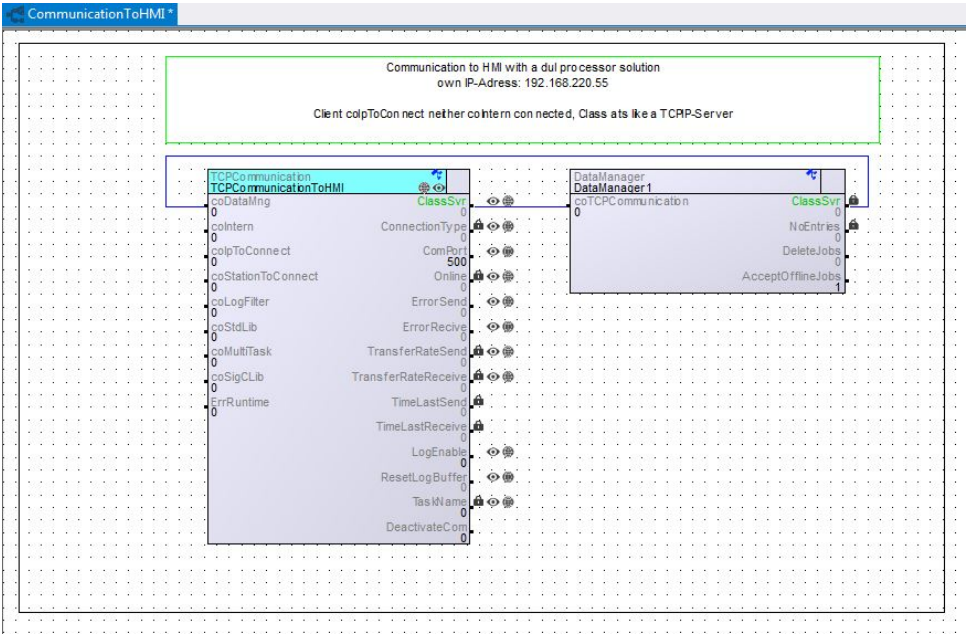
Configuring a TCPIP Client resp. Server

If the client coIpToConnect is connected to a string object containing the IP address of the TCPIP server, this block of the communication acts as a TCPIP client. It has to be cared, that the client coIntern must not be connected in a 2 CPU solution.

If the object should act as a TCPIP server, the clients coIpToConnect and coIntern may not be connected.

Example:
HMI component located on processor with the IP 192.168.220.10
PLC component located on processor with the IP 192.168.220.55


CommunicationToPLC *
Communication to PLC with a dual processor solution
own IP-Adress 192.168.220.10
Client colpToConnect is connected and the client cointern is not connected, the class reacts as a TCPIP-Client
String
String1
SingleRealloc
Data
"192.168.220.10"
TCPCommunication
TCPCommunicationToPLC
coDataMng
coIntern
colpToConnect
coStationToConnect
coLogFilter
coStdLib
coMultiTask
coSigCLib
ErrRuntime
ClassSvr
ConnectionType
ComPort
500
Online
ErrorSend
ErrorRecive
TransferRateSend
TransferRateReceive
TimeLastSend
TimeLastReceive
LogEnable
ResetLogBuffer
TaskName
DeactivateCon
DataManager
DataManagerToPLC
coTCPCommunication
ClassSvr
NoEntries
DeleteJobs
AcceptOfflineJobs
1



Interfaces

Clients

coDataManager	Connection to the data manager Object channel: DataManager (via this client, the manager reads the data to send)	
	Data type	Object channel to DataManager
coIntern	Must be internally connected (single processor solution) to the other TCPCommunication object (to ClassSvr).	
	Data type	Object channel to TCPCommunication
coIpToConnect	Needed when the TCPIP client should function as a class. Connection to an object of the String or StringRAM class (IP address of the desired TCPIP server).	
	Data type	UDINT
coLogFilter	Connection to the TCPCommunicationLogFilter.	
	Data type	Object channel to TCPCommunicationLogFilter

coStdLib	Connection to the _StdLib operating system interface (established automatically) Data type: Object channel to _StdLib
coMultiTask	Connection to the operating system interface _MultiTask class (created automatically). Data type: Object channel to _MultiTask
coSigCLib	Connection to the operating system interface SigClib (created automatically). Data type: Object channel to SigClib
ErrRuntime	If the parallel communication task is not called for a longer time, an error is output here. Data type: DINT
SizeOfComBuffers	Size of Rx as well as Tx buffer in bytes. Data type: DINT
CyclicTaskIntern	Configuration, which task should be used for an internal connection 0 parallel communication task is created >0 communication is handled in the cyclic task Here, you must care for the set cyclic time! (default = 100 ms) Data type: DINT

Servers

ClassSvr	Object server of the class. The coTCPCommunication client of the DataManager must be connected here. The TCPCommunicationLogViewer class can also be connected here, in order to display and export log entries.			
	Unit	-	Data type	DINT
	Value range	-	Write protected	FALSE
	Default value	-	Retentive	FALSE
InitError	If this server is unequal to 0, a severe error occurred during the initialization of the class. A communication cannot be started. -1 The parallel communication task could not be started -2 A required CIL Interface pointer could not be gathered -3 Error while allocating memory			
	Unit	-	Data type	DINT
	Value range	-3 … 0	Write protected	TRUE
	Default value	-	Retentive	FALSE
ConnectionType	Shows the type of configured connection.			
	Unit	-	Data type	t_e_ConnectionType
	Value range	0-4	Write protected	TRUE
	Default value	-	Retentive	FALSE
ComPort	Here, the TCP port over which to communicate must be entered. When sending and receiving, the same port must be specified. When communicating with several CPUs, different ports must be assigned for the different CPUs.			

	Unit	-	Data type	_FSM_TCP_USER
	Value range	max. UDINT	Write protected	TRUE
	Default value	configurable	Retentive	FALSE
Online	Online connection status. 0 offline 1 online			
	Unit	-	Data type	DINT
	Value range	0-1	Write protected	TRUE
	Default value	-	Retentive	FALSE
ErrorSend	When incremented, an error should occur when sending.			
	Unit	-	Data type	DINT
	Value range	max. DINT	Write protected	FALSE
	Default value	-	Retentive	FALSE
ErrorRecive	When incremented, an error should occur when receiving.			
	Unit	-	Data type	_FSM_TCP_USER
	Value range	max. DINT	Write protected	TRUE
	Default value	-	Retentive	FALSE
TransferRateSend	Communication speed in [Bits/s] (Upload)			
	Unit	Bits/s	Data type	DINT
	Value range	-	Write protected	TRUE
	Default value	-	Retentive	FALSE
TransferRateRecive	Communication speed in [Bits/s] (Download)			
	Unit	Bits/s	Data type	DINT
	Value range	-	Write protected	TRUE
	Default value	-	Retentive	FALSE
TimeLastSend	Time required by the last TCP packet sent [µs].			
	Unit	µs	Data type	UDINT
	Value range	-	Write protected	TRUE
	Default value	-	Retentive	FALSE
TimeLastRecive	Time to completion needed by the last TCP packet received [µs].			
	Unit	µs	Data type	UDINT
	Value range	-	Write protected	TRUE
	Default value	-	Retentive	FALSE
LogEnable	This server can be used to activate Logging. So that the TCPCommunicationLogViewer can export the data into an Excel file, logging must be disabled. 0 Logging deactivated 1 Logging active			
	Unit	-	Data type	DINT
	Value range	0-1	Write protected	FALSE
	Default value	-	Retentive	FALSE

ResetLogBuffer	Writing to this server clears the internal log buffer. This function can only be performed when the LogEnable server is set to 0.

Unit	-	Data type	DINT
Value range	-	Write protected	FALSE
Default value	-	Retentive	FALSE

TaskName	With this server string functions can be used

Unit	-	Data type	UDINT
Value range	-	Write protected	TRUE
Default value	0	Retentive	FALSE

DeactivateCom	Here, the entire communication can be disabled.
	DeactivateCom.Write(input := 1); // communication is deactivated
	DeactivateCom.Write(input := 0); // communication is activated again

Unit	-	Data type	DINT
Value range	0-1	Write protected	FALSE
Default value	0	Retentive	FALSE

Global Methods

Init	Not for the user.
GetOnlineState	Here, the status of the TCP/IP connection can be read. ▶ none ◀ retcode 0 Offline 1 Online
SetParameter	Function to set different communication parameters. ▶ ParaNo parameter number ▶ ParaVal value to be set ◀ retcode: 0 value successfully changed -1001 invalid parameter number Constants for parameter numbers: TCPCom_ParaWR_ComPort communication port TCPCom_ParaWR_AliveSignalTime time interval for sending Alive packets TCPCom_ParaWR_MissingAliveError error threshold for lost Alive packets: after X missing packets an error is triggered and the connection is reestablished. TCPCOM_ParaWR_TaskPriority task priority for the TCP task (must be set after the 1st and before the last run of Init()!)
ReadParameter	Function to read different communication parameters.

▶ ParaNo parameter number
▶ pParaVal to this address the value is written to
◀ retcode
0 value successfully changed
-1001 invalid parameter number
-1002 for pParaVal a NIL pointer was transferred

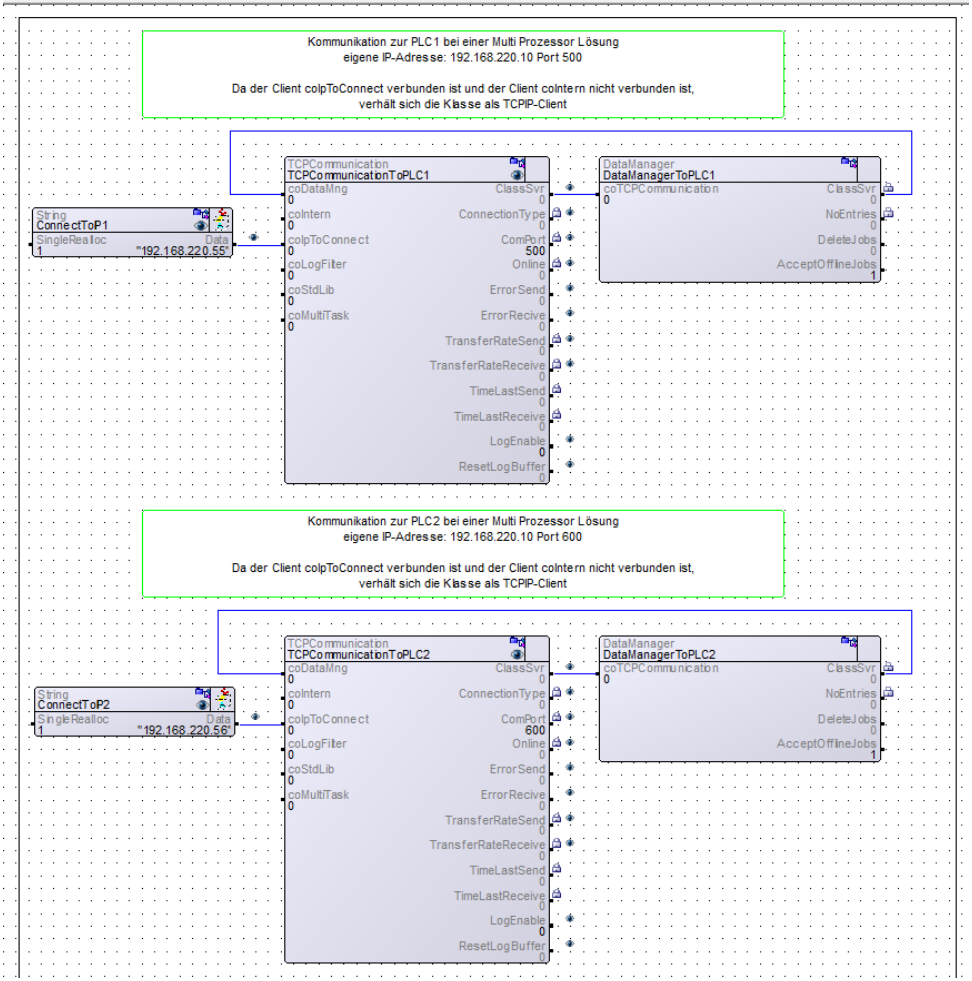
Constants for parameter numbers:
TCPCom_ParaRD_ComPort communication port
TCPCom_ParaRD_AliveSignalTime time interval for sending Alive packets
TCPCom_ParaRD_MissingAliveError error threshold for lost Alive packets: after X missing packets an error is triggered and the connection is reestablished.
TCPCom_ParaRD_TaskPriority task priority for the TCP task

Example of a 3-Processor Solution

HMI component located on processor with the IP 192.168.220.10
PLC1 component located on processor with the IP 192.168.220.55
PLC2 component located on processor with the IP 192.168.220.56

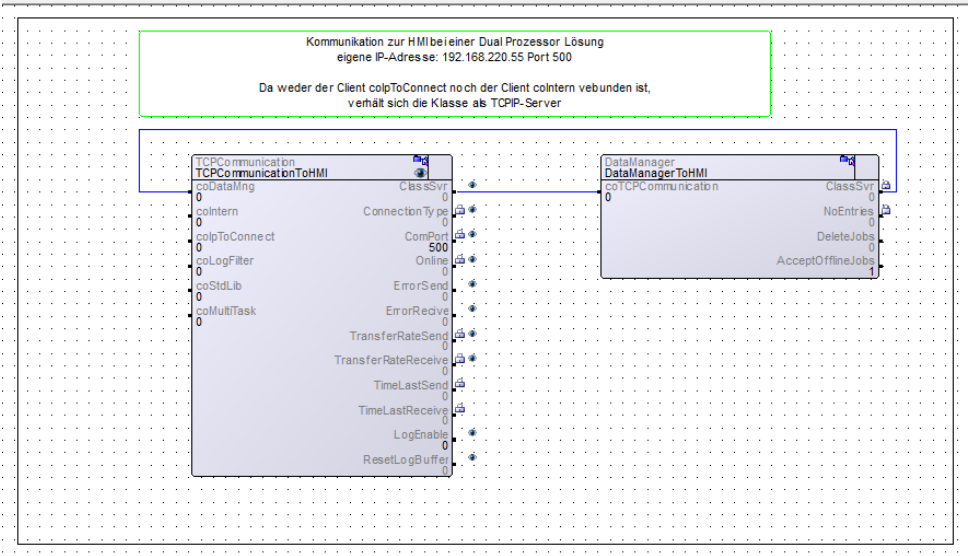
HMI Component

CommunicationToPLC *



PLC1 Component

CommunicationToHMI *



PLC2 Component