SIEMENS

Library description • 01/2015

Open user communication to 3rd party control system

STEP 7 (TIA Portal), S7-1200/S7-1500, Allen-Bradley

Warranty and liability

Note

> The Application Examples are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. The Application Examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for ensuring that the described products are used correctly. These application examples do not relieve you of the responsibility to use safe practices in application, installation, operation and maintenance. When using these Application Examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these Application Examples at any time without prior notice.
> If there are any deviations between the recommendations provided in these application examples and other Siemens publications – e.g. Catalogs – the contents of the other documents have priority.

We do not accept any liability for the information contained in this document.

Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act ("Produkthaftungsgesetz"), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract ("wesentliche Vertragspflichten"). The damages for a breach of a substantial contractual obligation are, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.

Any form of duplication or distribution of these Application Examples or excerpts hereof is prohibited without the expressed consent of the Siemens AG.

Security information

> Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, equipment and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens' products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.
>
> For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Third-party products that may be in use should also be considered. For more information about industrial security, visit http://www.siemens.com/industrialsecurity.
>
> To stay informed about product updates as they occur, sign up for a product-specific newsletter. For more information, visit http://support.automation.siemens.com.

Table of contents

1 Library Overview

Main topics

This document describes the "T_COMM" block library. The block library provides you with tested code with clearly defined interfaces. They can be used as a basis for your task to be implemented.

A key concern of the document is to describe

- the blocks of the block libraries.
- the functionality implemented through the block.

With the libraries you can implement the communication between an S7-1500/S7-1200 controller and a CLX/GLX controller. Therefore two libraries are provided; one for the use with an S7, one for the use with a CLX/GLX controller.

Furthermore, this documentation shows possible fields of application and helps you to integrate the library into your STEP 7 project using step-by-step instructions.

1.1   User scenario

Possible application for the "T_COMM" library

To realize a communication a between Siemens S7-1500 or S7-1200 controller and a 3rd party controller this application example demonstrates the use of the built-in interfaces and program functions (without 3rd party adapter or gateway).

The library includes an FB for the Siemens controller and an AOI for the Rockwell Automation controller which is handling the communication establishment and data exchange.

Standard communication instructions are used inside the FB and AOI, however all required logic is encapsulated and interfaces to the user program are provided for easy configuration and quick setup.

Figure 1-1 Application example
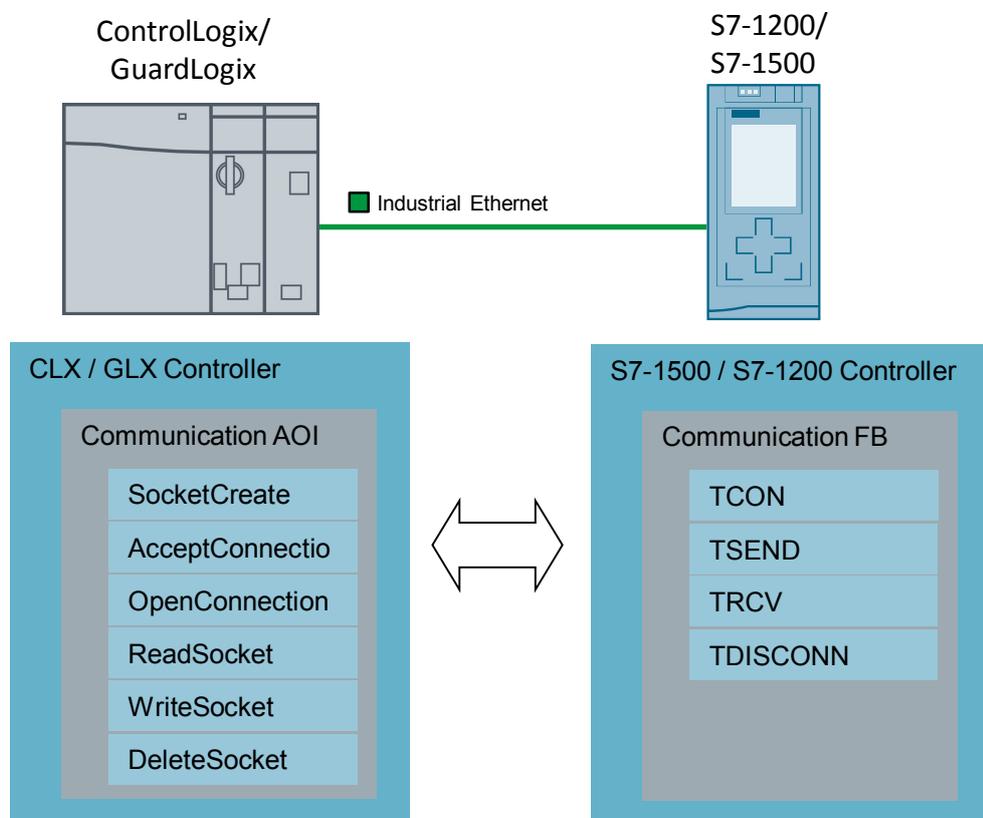
Advantages

The "T_COMM" library offers the following advantages:

- function block for an easy startup with the communication configuration.
- easy-to-use and easy-to-adapt program block.
- different options (server/client, send/receive) for the communication are available.

1.2 Required Hardware and Software Components

The application was generated with the following components:

Hardware components

Table 1-1

Component	No.	MLFB / order number	Note
Siemens CPU 1511-1 PN, FW v1.6	1	6ES7 511-1AK00-0AB0	
Siemens CPU 1215C DC/DC/DC, FW v4.0	1	6ES7 215-1AG40-0XB0	
Rockwell Automation GuardLogix Controller	1	1756-L72S and 1756-L7SP FW 21.11	Alternatively any ControlLogix CPU 1756-L7x can be used
Rockwell Automation EthernetBridge	1	1756-EN2T FW v.10	Alternatively 1756-En2xx Module FW minimum v5.007 or 1756-EWEB, FW minimum v4.006 can be used

Standard software components

Table 1-2

Component	No.	MLFB / order number	Note
SIMATIC STEP 7 Professional V13	1	6ES7822-1AA03-0YA5	
Rockwell Automation Studio 5000 Logix Designer Version 21.03	1		

Sample files and projects

The following list includes all files and projects that are used in this example.

Table 1-3

Component	Note
S7_T_COMM_FB_v0.1.zip	This ZIP file includes a library with T_COMM FB v.0.1 and a sample data DB.
RA_T_COMM_AOI_v1.0.zip	This ZIP file includes the AOI which can be imported to Automation Studio 5000.
108740380_TCPCommRA_lib_en.pdf	This document.

1.3 Library resources

What will you find in this section?

The following section gives you an overview of the size of the blocks of "T_COMM" library in the main memory.

Overall size

The overall size of all blocks of the "T_COMM" library in STEP 7 in the work memory is 1362 Bytes.

Size of the individual blocks of the STEP 7 library

Table 1-4

Block	Symbol	Size in load memory	Size in work memory
FB 1	T_COMM	18673 Bytes	984 Bytes
DB 4	DB_DATA	1772 Bytes	378 Bytes

Size of the library in CompactLogix and GuardLogix

Table 1-5

Controller	I/O memory	Data and Logic Memory
CompactLogix	3 936 bytes	139 216 bytes
GuardLogix	41 208 bytes	156 540 bytes

2 Blocks of the Library "T_COMM"

What will you find in this section?

This chapter lists (chapter 2.1) and explains all blocks of the "T_COMM" library for STEP 7 (chapter 2.2) and for the Automation Studio 5000 (chapter 2.3).

2.1 List of the blocks

The following table lists all blocks of the "T_COMM" library.

Table 2-1

Block	Symbol	Engineering Framework
FB1	T_COMM	STEP 7
DB4	DB_DATA	STEP 7
AOI	T_COMM	Automation Studio 5000
Data type	Message_ErrorRecord	Automation Studio 5000
Data type	SocketComm	Automation Studio 5000
Data type	SocketComm_AcceptResponse	Automation Studio 5000
Data type	SocketComm_CreateParams	Automation Studio 5000
Data type	SocketComm_OpenConnParams	Automation Studio 5000
Data type	SocketComm_ReadParams	Automation Studio 5000
Data type	SocketComm_ReadResponse	Automation Studio 5000
Data type	SocketComm_SockAddr	Automation Studio 5000
Data type	SocketComm_WriteParams	Automation Studio 5000

2.2 Explanation of the FB "T_COMM" for STEP 7

The following chapter explains the FB "T_COMM" for the engineering framework STEP 7.

2.2.1 Functionality

Figure 2-1
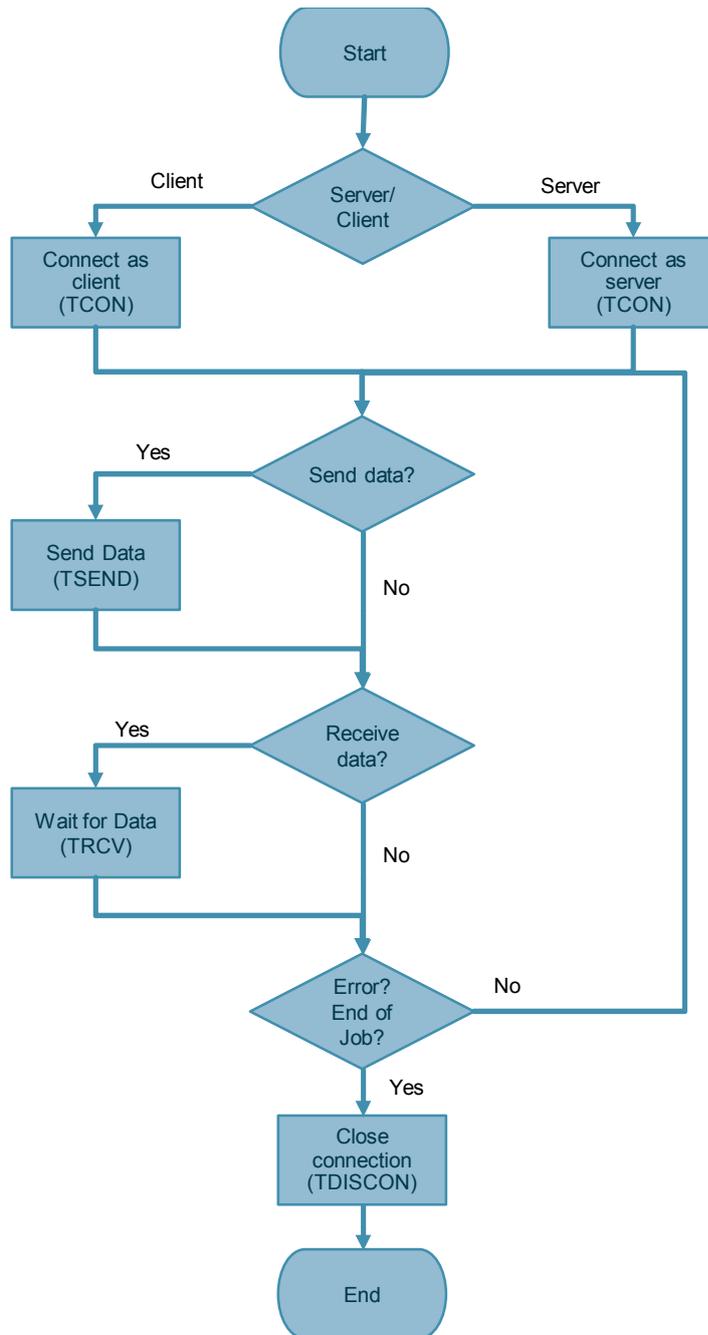
1. Connect to communication partner
   - If configured as server: Execute TCON with the partner configuration set to 'active'.
   - If configured as client: Execute TCON with the partner configuration set to 'passive'.
2. Execute TSEND if configured to send data
   - Sending data to the communication partner over the connection established in step 2
   - Continuous configuration:
     - YES: TSEND is repeated with short delay continuously until disabled from the user program or error occurs.
     - NO: TSEND is executed only once.
3. Execute TRCV if configured to receive data
   - Waiting for incoming data and receiving data if the data amount specified in "DataLen" is received. The connection established in step 2 is used.
   - Continues configuration:
     - YES: TRCV is repeated after data has been received until disabled from the user program or error occurs.
     - NO: TRCV is executed only once.
4. Close connection
   - Execute TDISCON to close connection if any error occurs, send and receive once is finished (Continuous = FALS) or disabled from the user program using the input of the instruction "T_COMM".

2.2.2 Parameters

Figure

Figure 2-2
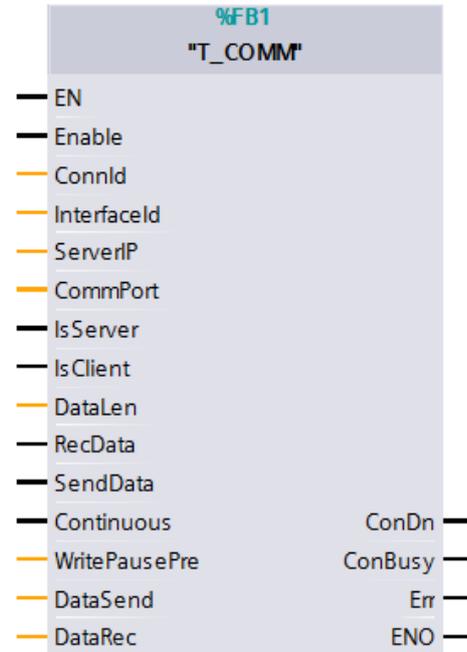
Table 2-2

Name	Type	Description
Enable	IN: BOOL	Communication Enable To be set from the user program to start sending or receiving data.
ConnId	IN: CONN_OUC	Connection ID must be unique for the controller. Define any available connection ID number.
InterfaceId	IN: HW_ANY	For S7-1500 = 64 For S7-1200 = 1
ServerIp	IN: IP_V4	In case FB is used as client, connect tag which holds the IP address of the server. If FB is used as server, leave parameter unconnected.
CommPort	IN: UINT	Communication Port
IsServer	IN: BOOL	Act as Server Set TRUE if server functionality is used (accepting incoming connection). If "IsServer" is set to TRUE, "IsClient" must be FALSE.

Name	Type	Description
IsClient	IN: BOOL	Act as Client Set TRUE if client functionality is used (create a outgoing connection). If "IsClient" is set to TRUE, "IsServer" must be FALSE.
DataLen	IN: UINT	Length of data to read or write [bytes] No. of bytes to be sent or received. The maximum is 100. If more data is to be used, the FB needs to be modified.
RecData	IN: BOOL	Receive Data Enables receiving data.
SendData	IN: BOOL	Send Data Enables sending data.
Continuous	IN: BOOL	Continuously send and receive data Default: TRUE
WritePausePre	IN: TIME	Pause preset between sending data finished and next start of sending data [ms]
DataSend	INOUT: Array[0..99] of Byte	Data which will be sent. Must be a 100 element array. But sending data can be limited by the "DataLen" parameter
DataRec	INOUT: Array[0..99] of Byte	Data which are received. Must be a 100 element array but the amount of data which really are received is defined by the "DataLen" parameter
ConDN	OUT: BOOL	Connection established
ConBusy	OUT: BOOL	If configured as server: Listening, waiting for incoming connection If configured as client: Waiting that server is accepting the connection
Err	OUT: BOOL	Error present

2.3 Explanation of the AOI "T_COMM" Automation Studio 5000

2.3.1 Functionality

Figure 2-3

```
            ┌─────────────┐
            │    Start    │
            └──────┬──────┘
                   │
            ┌──────▼──────┐
            │ Create socket│
            │(SocketCreate)│
            └──────┬──────┘
                   │
        Client ◄── Server/ ──► Server
              ┌──  Client  ──┐
    ┌─────────▼──┐       ┌───▼────────┐
    │Connect as  │       │Connect as  │
    │  client    │       │  server    │
    │(OpenConn.) │       │(AcceptConn)│
    └────────────┘       └────────────┘
```

Send data? — Yes → Send Data (WriteSocket) — No

Receive data? — Yes → Wait for Data (ReadSocket) — No

Error? End of Job? — No / Yes

Close connection (DeleteSocket)

End
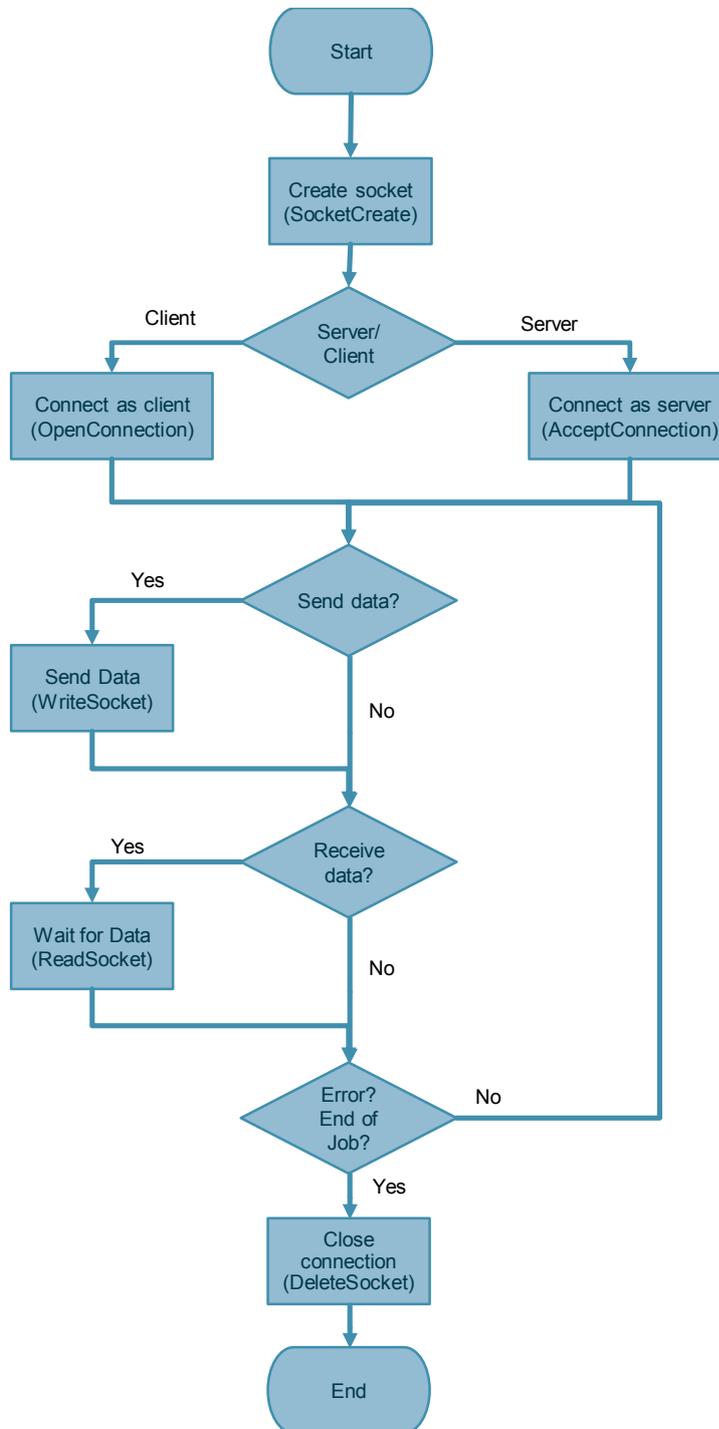
1. Send "SocketCreate" Message instruction
   - Create an outgoing connection or start listening for incoming connection
   - If configured as server, send message instruction "AcceptConnection" to start listening for incoming connection.
   - If configured as client, send message instruction "OpenConnection" to establish a connection with the server

2. Send message instruction "WriteSocket" if configured to send data
   - Sending data to the communication partner over the connection established in step 2
   - Continuous configuration:
     o YES: message instruction is repeated with short delay continuously until disabled from the user program or error occurs.
     o NO: message instruction is executed only once

3. Send message instruction "ReadSocket" if configured to receive data
   - Waiting for incoming data and receiving data if the data amount specified as "DataLen" is received. Connection established in step 2 is used.
   - Continuation configuration:
     o YES: message instruction is repeated after data have been received until disabled from the user program or error occurs.
     o NO: message instruction is executed only once

4. Close connection
   - Send message instruction "DeleteSocket" to close connection if any error occurs, send and received once is finished (Continuous = FALSE) or disabled from the user program.

2.3.2 Parameters

Figure 2-4
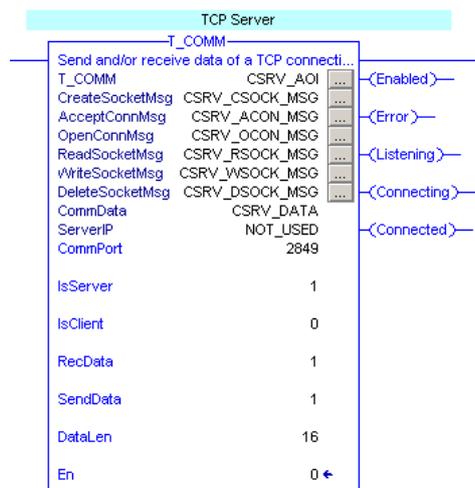
Table 2-3

Name	Type	Description
CreateSocketMsg	MESSAGE	CreateSocket Message Instruction *Requires a controller scope tag as parameter*
AcceptConnMsg	MESSAGE	AcceptConnection Message Instruction *Requires a controller scope tag as parameter*
OpenConnMsg	MESSAGE	OpenConnection Message Instruction *Requires a controller scope tag as parameter*
ReadSocketMsg	MESSAGE	ReadSocket Message Instruction *Requires a controller scope tag as parameter*
WriteSocketMsg	MESSAGE	WriteSocket Message Instruction *Requirse a controller scope tag as parameter*
DeleteSocketMsg	MESSAGE	WriteSocket Message Instruction *Requires a controller scope tag as parameter*
CommData	SocketComm	Communication Data and Settings *Requires tag as parameter* Data structure of the type "SocketComm" which contains configuration and data to be sent and received.
ServerIP	STRING	Communication Data and Settings *Requires tag as parameter* String variable which contains the server IP address e.g. "10.88.81.10". If no client functionality is required, create a new string tag but leave the value empty.
CommPort	DINT	Communication Port *Configure value directly at the AOI instruction block*
IsServer	BOOL	Act as Server *Configure value directly at the AOI instruction block* Set TRUE if server functionality should be used (accepting incoming connection). If "IsServer" is set to TRUE, "IsClient" must be FALSE.
IsClient	BOOL	Act as Client *Configure value directly at the AOI instruction block* Set TRUE if client functionality should be used (create a outgoing connection). If "IsClient" is set to TRUE, "IsServer" must be FALSE.

Name	Type	Description
RecData	BOOL	Receive Data *Configure value directly at the AOI instruction block* Enables receiving data.
SendData	BOOL	Send Data *Configure value directly at the AOI instruction block* Enables sending data.
DataLen	DINT	Length of data to read or write [bytes] *Configure value directly at the AOI instruction block* No. of bytes to be sent or received. The maximum is 100 (length of array in the SocketComm structure). If more data is to be used, the UDT needs to be modified.
En	optional: BOOL	Communication Enable To be set from the user program to start sending or receiving data.
ServiceTimout	optional: DINT	Service Timout [ms] Default: 1500000
MsgTimout	optional: DINT	Message Instruction Timeout [ms] Default: 1800000
WritePausePre	optional: DINT	Pause between writing to socket [ms] Default: 100
Continuous	optional: BOOL	Continuously send and receive data Default: TRUE
ReTryOpenConn	optional: BOOL	Retry OpenConnection Default: TRUE
Enabled	Output: BOOL	Communication logic enabled
Error	Output: BOOL	Error present
Listening	Output: BOOL	Listening for incoming connection
Connecting	Output: BOOL	Waiting for outgoing connection to be accepted
Connected	Output: BOOL	Connected (incoming connection received or outgoing connection accepted)
Busy	Output: BOOL	Waiting for connection to be closed

2.3.3 Configuration

Message instruction configuration

The AOI uses 6 message instructions. Because the AOI is prepared for server / client functionality and can send / receive data, all message instructions must be initialised, regardless of wether they will be used during runtime.

Use the following tables to initialize the message instructions:

Socket Create message instruction - Configuration

Field	Value
Message Type	CIP Generic
Service Type	Socket Create
Instance	n/a
Source Element	<CommDataTag>.CreateParams
Source Length	12 (fixed, not editable)
Destination Element	<CommDataTag>.CreateInstance

AcceptConnection message instruction - Configuration

Field	Value
Message Type	CIP Generic
Service Type	AcceptConnection
Instance	0 (will be updated automatically by the program)
Source Element	<CommDataTag>.AcceptTimeout
Source Length	4 (fixed, not editable)
Destination Element	<CommDataTag>.AccpetResponse

OpenConnection message instruction - Configuration

Field	Value
Message Type	CIP Generic
Service Type	OpenConnection
Instance	0 (will be updated automatically by the program)
Source Element	<CommDataTag>.OpenParams
Source Length	1 (will be updated automatically by the program)
Destination Element	n/a

ReadSocket message instruction - Configuration

Field	Value
Message Type	CIP Generic
Service Type	ReadSocket
Instance	0 (will be updated automatically by the program)
Source Element	<CommDataTag>.ReadParam
Source Length	8 (fixed, not editable)
Destination Element	<CommDataTag>.ReadData

WriteSocket message instruction - Configuration

Field	Value
Message Type	CIP Generic
Service Type	WriteSocket
Instance	0 (will be updated automatically by the program)
Source Element	<CommDataTag>.WriteParams

Source Length	16 (will be updated automatically by the program)
Destination Element	<CommDataTag>.WriteResponse

DeleteSocket message instruction - Configuration

Field	Value
Message Type	CIP Generic
Service Type	DeleteSocket
Instance	0 (will be updated automatically by the program)
Source Element	n/a
Source Length	n/a
Destination Element	n/a

Message Instruction communication configuration

The communication path must be configured for all message instructions in the same way. This must be done for each message instruction together with the general configuration (the Automation Center will display an error if the configuration is closed with an empty communication path field).

The communication path is the path to the Ethernet card which will be used for communication (not to the target device). The Ethernet card can be selected from the I/O configuration tree (if configured) or the path can be manually defined (in the example: 1, 2 – 1 = From the controller to the backplane, 2 = card in slot 2).

Figure 2-5
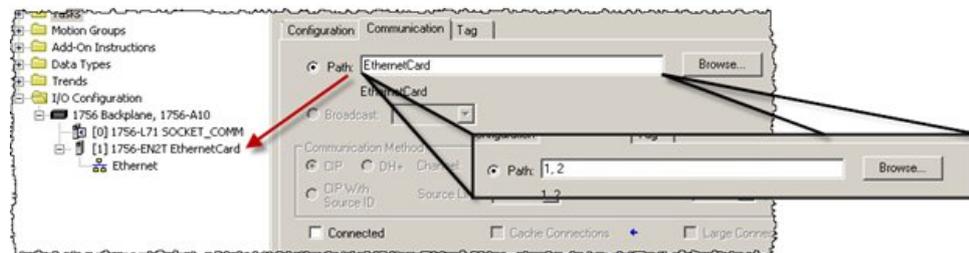
2.3.4   User data

The user data which are to be sent or received are part of the configuration and data structure. In this sample application, there is an array of 100 byte which can be used for user data. The number of elements which are used must match the "DataLen" parameter.

Figure 2-6

To use more data than 100 byte, the UDT "SocketComm_WriteParams" and/or "SocketComm_ReadResponse" need to be changed.

Figure 2-7

Service	Unconnected Size	Standard Connection Size	Large Connection Size
ReadSocket	484 bytes	484 bytes	3984 bytes
WriteSocket	462 bytes	472 bytes	3972 bytes

Depending on the change, the message instructions for ReadSocket and/or WriteSocket need to be switched to "Large Connection"

3 Working with the Library

3.1 Importing library to STEP 7 Project

The following table shows you how to import the library into a STEP 7 (TIA Portal) project.

Table 3-1

Nr.	Action
1.	Open STEP 7 and open a STEP 7 project with an S7-PLC.
2.	Open the task card "Library".
3.	Click the "Open" button.
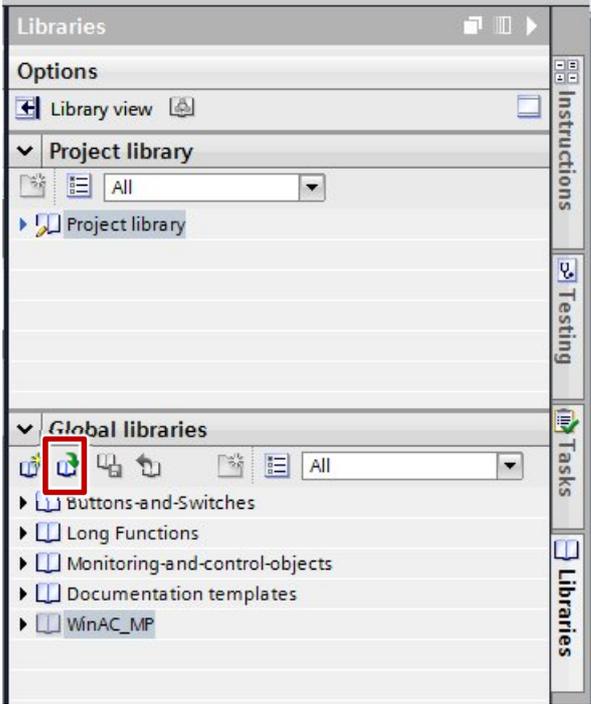
Nr.	Action
4.	Browse to the directory where the library data of the download are located.
5.	Copy the function and the data block per Drag&Drop to your PLC project.
6.	Now you can add the function block to your application program.
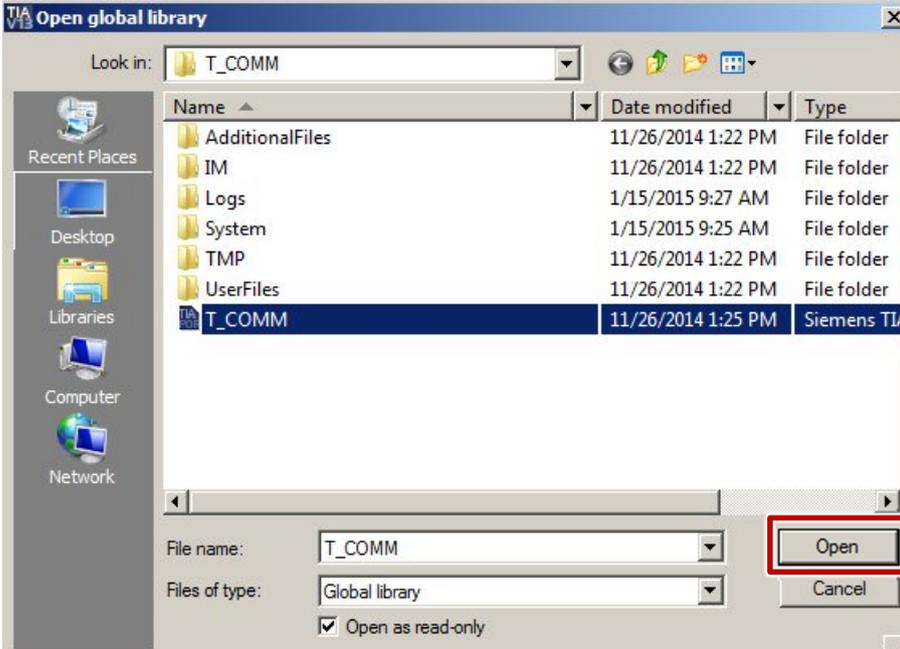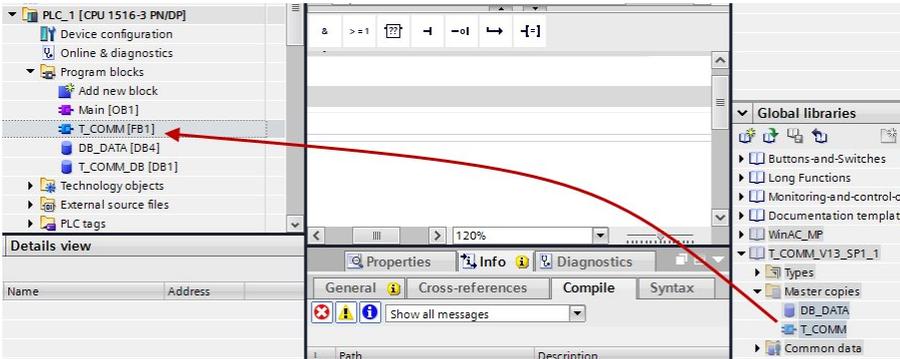
## 3.2 Importing AOI to Studio 5000 Project

The following table shows you how to import the AOI into an Automation Studio 5000 project.

Table 3-2

Nr.	Action
1.	Open Studio 5000 and open a controller project or create an empty project.
2.	Right click on "Add-On Instructions" in the project tree and select "Import Add-On instruction …"
3.	Browse to the Add-On Instruction file, select "T_COMM" and select "Import…"
4.	Confirm the import dialog by pressing "OK" If the project is empty or the project does not contain the AOI, there will be no report about conflicts etc. If there is a previous version of the Add-On Instruction present, a report will be shown with details about modifications in the program.
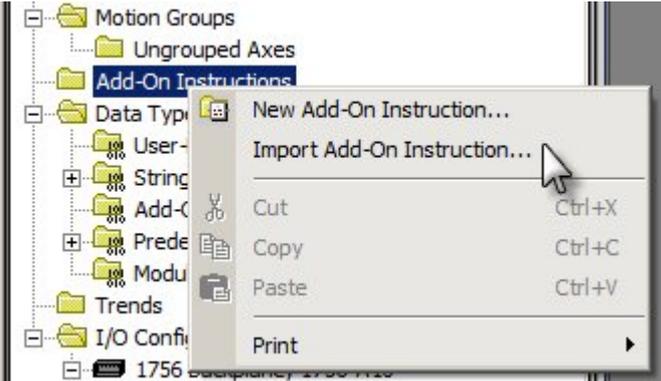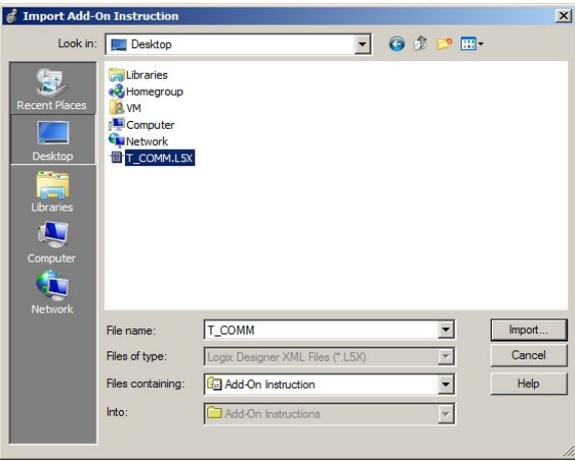
Nr.	Action
5.	The AOI will be imported. Also all UDTs required for the AOI will be created.
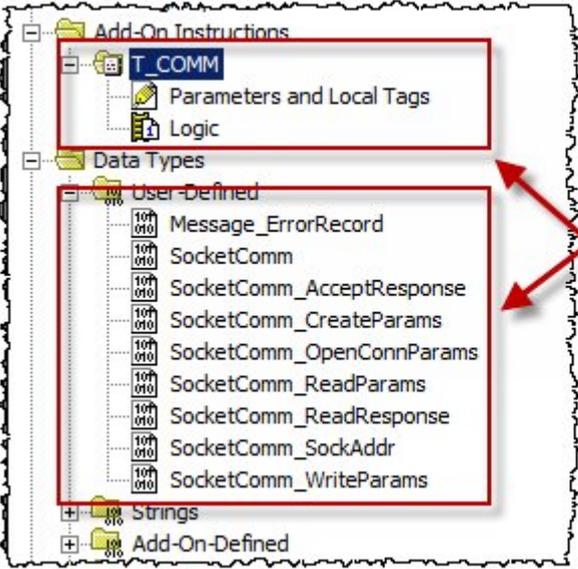
3.3 Scenario A: S7 controller is the active partner

In this scenario, the S7 controller is the active partner, which means that it will act as a client.

Set the following parameters.

The settings for IP address and the port to use are depending on the individual setup.

Table 3-3

No.	Instruction	Remarks
1.	Set FB input "IsServer" to FALSE	on FB instance in the S7 CPU
2.	Set FB input "IsClient" to TRUE	on FB instance in the S7 CPU
3.	Set AOI input "IsServer" to TRUE	on AOI instance in the Logix CPU
4.	Set AOI input "IsClient" to FALSE	on AOI instance in the Logix CPU
5.	Set AOI input "En" to TRUE to start listening	on AOI instance in the Logix CPU
6.	Set FB input "Enable" to TRUE to start connection	on FB instance in the S7 CPU
7.	Write data to the tag connected to the "DataSend" interface	on FB instance in the S7 CPU
8.	Read the data from user data structure	on AOI instance in the Logix CPU
9.	Write data to the send data part of the user data	on AOI instance in the Logix CPU
10.	Read data from the tag connected to "DataRec" interface	on FB instance in the S7 CPU
11.	Download the programs both to the S7-PLC and the Logix PLC.	over STEP 7 / Automation Studio (see chapter 3.6)
12.	Data exchange will run continuously until the enable signal is reset on client or server side	

3.4 Scenario B: S7 controller is the passive partner

In this scenario, the S7 controller is the passive partner, which means that it will act as a server listening for incoming connections.

Table 3-4

No.	Instruction	Remarks
1.	Set FB input "IsServer" to TRUE	on FB instance in the S7 CPU
2.	Set FB input "IsClient" to FALSE	on FB instance in the S7 CPU
3.	Set AOI input "IsServer" to FALSE	on AOI instance in the Logix CPU
4.	Set AOI input "IsClient" to TRUE	on AOI instance in the Logix CPU
6.	Set FB input "Enable" to TRUE to start listening	on FB instance in the S7 CPU
5.	Set AOI input "En" to TRUE to start connection	on AOI instance in the Logix CPU
7	Write data to the tag connected to the "DataSend" interface	on FB instance in the S7 CPU
8.	Read the data from user data structure	on AOI instance in the Logix CPU
9.	Write data to the send data part of the user data	on AOI instance in the Logix CPU
10.	Read data from the tag connected to "DataRec" interface	on FB instance in the S7 CPU
11.	Download the programs both to the S7-PLC and the Logix PLC.	over STEP 7 / Automation Studio (see chapter 3.6)
12.	Data exchange will run continuously until the enable signal is reset on client or server side	

3.5 Further scenarios

There are more scenarios possible which depends on the configuration of the program block

- Only send data from server to client
- Only send data from client to server
- Send and receive data once
- Send data only once from server to client
- Send data only once from client to server

These use cases depend on the configuration of the "RecData", "SendData" and "Continuous" parameters.

3.6 Download the user program

3.6.1 STEP 7

To download the user program to the S7-PLC please follow the instructions of the following table.

Table 3-5

No.	Action
1.	Open your STEP 7 project. Click on the S7-PLC in the project tree.
2.	Click on the main menu entry "Online > Download to device".
3.	After choosing the network access Type of the PG/PC interface: PN/IE PG/PC interface: Intel(R) PRO/1000 MT Network Connection Connection to interface/subnet: Direct at slot '1 X1' 1st gateway: click on 'Start Search' to find your PLC in the network. Show all compatible devices Address / Target device ccess address / — Start search
4.	Click on the PLC and then click on "Load" Confirm with the following dialogues.
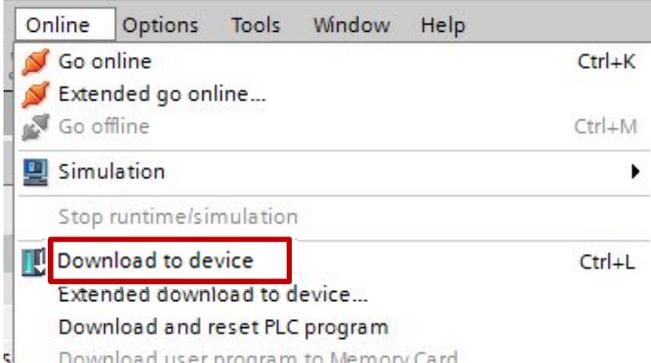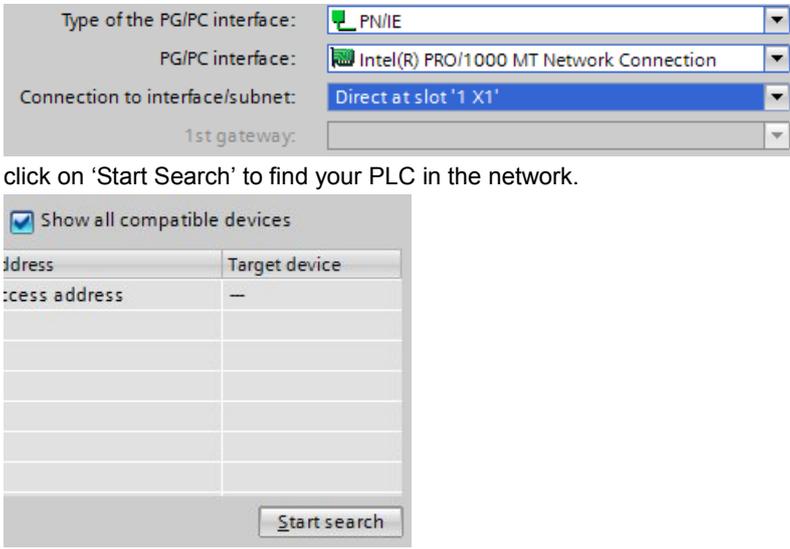
3.6.2 Automation Studio

To download the user program to the CLX/GLX controller follow the instructions of the following table.

Table 3-6

No.	Action
1.	Open your project in the Automation Studio.
2.	Click on "Download".

| No.	Action
3.	Activate the checkbox "Download Project Documentation and Extended Properties". Click then on „Download".
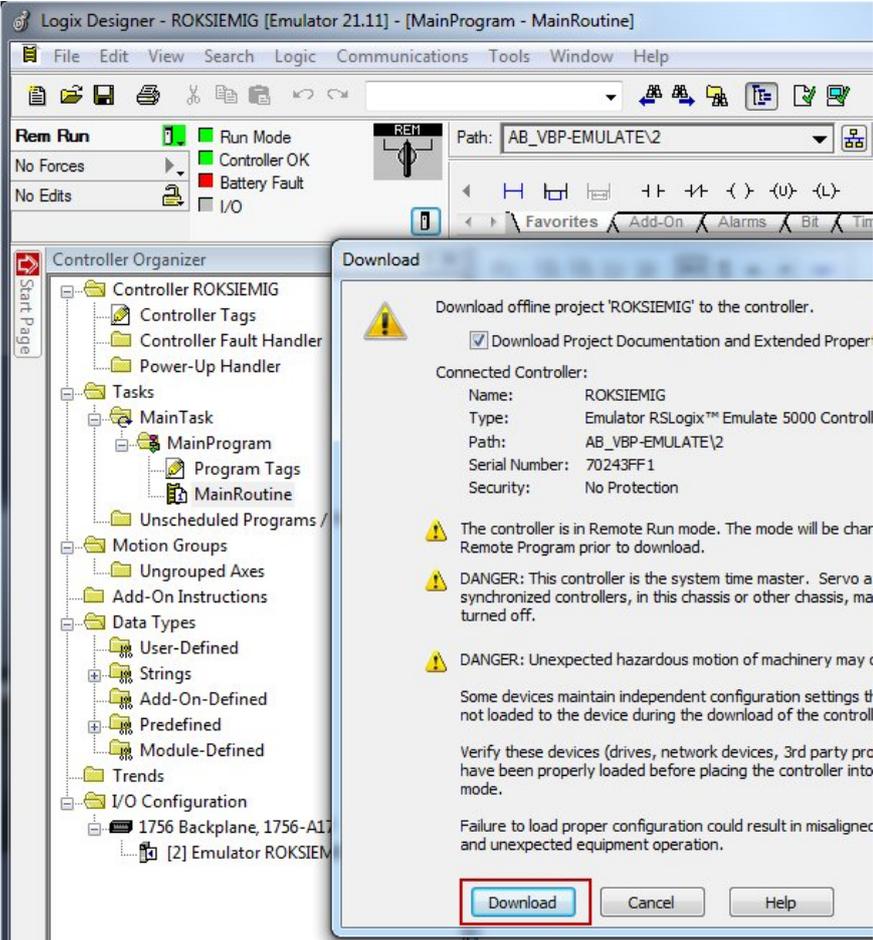
4 Further Notes, Tips and Tricks, etc.

4.1 Error Codes of open user communications instructions

For debugging purposes, the last error codes are stored in the instance DB of the communication function block:

Figure 4-1
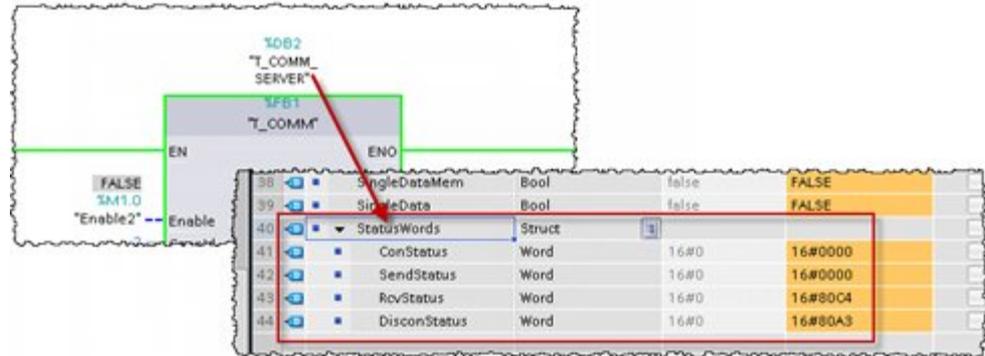
Error codes can be found in the TIA portal online help.

4.2 Error Codes of message instruction

For debugging purposes, the last error codes are stored in the instance tag of the communication Add-On instruction:

Figure 4-2

Error codes can be found in the Studio 5000 online help and in the "EtherNet/IP Socket Interface" application technique manual.

5 Literatur

This list is not complete and only represents a selection of relevant information.

Table 5-1

	Subject	Title
\1\	EtherNet/IP Socket Interface Application Technique	http://literature.rockwellautomation.com/idc/groups/literature/documents/at/enet-at002_-en-p.pdf
	Link to this application example	http://support.automation.siemens.com/WW/view/en/108740380

6 History

Table 6-1

Version	Date	Modifications
V1.0	02/2015	First version