

The cover features a background image of a modern, multi-story building with large windows and a flat roof. A yellow horizontal band runs across the middle of the building. In the foreground, there is a paved area with a few parked cars. The overall lighting is somewhat dim, suggesting an overcast day or dusk. A solid yellow vertical bar is positioned on the left side of the cover, partially overlapping the text area.

Non-Safe Simplifier Gateway Manual

Usman Mehmood & Bastian Jürgens

Software & Hardware Architect

SSP – Safety System Products

Contents

1	Application Description	1
2	How to Use	1
2.1	Modes and Configurations	1
2.2	Configurable Parameters	1
2.3	Default Values.....	1
2.4	CLI Arguments	2
2.5	Config File	2
2.6	Wrong Configuration	2
3	PLC Interface Data Format.....	3
3.1	Packet Structure	3
3.1.1	Packet Header.....	3
3.1.2	Packet Body	3
3.2	Packet Size and Speed	3
4	Updating Config and Software.....	4
5	PLC Implementation (Siemens TIA V16 or higher)	5
5.1	Opening the library and pasting the blocks and data types into the PLC program	5
5.2	Call the Function Block	6
5.3	User Data	7
6	Necessary Hardware (Example).....	8
7	Example with Simplifier Manager showing USB Inputs and Outputs.....	9
8	Common Errors.....	10

1 Application Description

The Non-Safe Simplifier Gateway is a device that acts as a gateway between multiple Safety Simplifiers and a PLC. It takes USB commands from the PLC and returns information about all connected Simplifiers as well as their global memories.

2 How to Use

2.1 Modes and Configurations

The device has two interfaces. A USB interface for the Safety Simplifier, and a TCP/IP interface for the PLC. Both can be simulated by passing a CLI argument.

If the USB interface is not simulated, a Safety Simplifier must be connected to the gateway.

If the PLC interface is not simulated, a Siemens PLC must be connected and programmed with an appropriate application. It needs to have the same IP address, rack number, DB number and slot number as passed to this application (or vice-versa).

The application can also take verbosity level, cycle time, communication timeout, and IP address of the Gateway as arguments.

2.2 Configurable Parameters

The following parameters of the app can be configured by the user:

- PLC's IP address, over which the TCP/IP connection takes place.
- Rack number of the PLC.
- Slot number of the PLC.
- DB number of the PLC's receive DB area.
- PLC interface's simulation status
- Safety Simplifier's connection IP address
- Safety Simplifier's connection port
- Safety Simplifier interface's simulation status
- Cycle time for data exchange with the Simplifier Gateway
- Timeout for data exchange with the Simplifier Gateway
- Number of retries in case of a timeout before restarting.
- Ratio of messages with global memories to messages with global memories and device data
- Simplifier Gateway's IP address (only if it is running on Linux)
- Console logs verbosity

2.3 Default Values

Parameter	Interface	Type	Default Value
Address	PLC	string	127.0.0.1
Rack	PLC	int	0
Slot	PLC	int	1
DB number	PLC	Int	1
Simulated	PLC	bool	false
Address	Simplifier	string	127.0.0.1
Port	Simplifier	int	5001
Simulated	Simplifier	bool	false
Cycle Time	Gateway	int	100 ms

Timeout	Gateway	int	1000 ms
Retries	Gateway	Int	3
Message Ratio	Gateway	int	20
Address	Gateway	string	192.168.2.201
Verbosity	Gateway	string	info

2.4 CLI Arguments

All the configurable parameters can be set with command line arguments.

- `-smp <sim> ip <string> port <int>` for setting Simplifier parameters
- `-plc <sim> ip <string> rack <int> slot <int> db <int>` for setting PLC parameters
- `-freq <int>` for setting the cycle time of data exchanged in milliseconds
- `-timeout <int>` for setting the communication timeouts in milliseconds.
- `-retries <int>` for setting number of retries in case of a timeout before restarting the app
- `-ip <string>` for setting the IP address of the Gateway itself if it is a Linux machine
- `-ratio <int>` for setting the ratio of GlobalMemory messages per DeviceData message
- `-v <debug/info/warning>` for setting the console verbosity level.

Example:

```
./SimplifierGateway -smp ip 127.0.0.1 port 50 -plc ip 127.0.0.1 rack 0 slot 1 db 1
-freq 500 -timeout 1000 -retries 3 -ratio 20 -v debug
```

2.5 Config File

To configure the app without using CLI arguments, a configuration file named `config.txt` can be placed in the app directory, which the app tries to read if no arguments are passed to it.

Shown below is the example and format of this file.

```
Verbosity          debug          # Verbosity level of console logs
MessageRatio       10             # Ratio of GlobalMemory messages
CycleTime          1000           # Cycle time of the Gateway
Timeout            1500           # Timeout for Simplifier and PLC
Retries            3              # Retry attempts before restarting app
Address            192.168.2.202  # IP address of the gateway

PlcParameters_IsSimulated TRUE      # PLC interface is simulated or not
PlcParameters_Address 192.168.2.200 # IP address of PLC
PlcParameters_Rack    0             # Rack number of PLC
PlcParameters_Slot    1             # Slot number of PLC
PlcParameters_DbNumber 1            # DB number for send / receive block

SmpParameters_IsSimulated FALSE     # Simplifier interface is simulated or not
SmpParameters_Address 127.0.0.1     # IP address of Simplifier
SmpParameters_Port    5001          # Port number of Simplifier
```

The text after `#` symbol is a comment, and is ignored by the app.

2.6 Wrong Configuration

If a parameter is not present in the config file or the CLI arguments, its default value is assumed. If a wrong (name or type) parameter is passed, the application will display the help menu and exit. If a wrong PLC or Simplifier related parameter is passed while the interface is simulated, it is ignored, and the default value is assumed.

3 PLC Interface Data Format

This communication takes place by writing to and reading from the Data Blocks in a Siemens PLC and using it as a "shared memory".

3.1 Packet Structure

The data sent to the PLC is arranged in the following format. u16 is unsigned 16-bit, u32 is unsigned 32-bit, s32 is signed 32-bit. Their endianness is the same as a Siemens PLC.

3.1.1 Packet Header

The packet's "header" contains the information common to all Simplifiers connected to the network.

	Name	Size	Type	Start	End	Description
1	Message count	4	u32	0	3	Number of messages exchanged
2	Node count	2	u16	4	5	Number of nodes in data sent
3	Gateway errors	2	u16	6	7	Gateway errors (unused for now)
4	USB outputs	2	u16	8	9	USB outputs from Simplifier
5	CRC Header	2	byte array	10	11	CRC of Global memories
6	CRC Device data	2	byte array	12	13	CRC of Simplifier data array
7	Global memories	32	byte array	14	45	Global memories

3.1.2 Packet Body

The packet's "body" is information unique to each Simplifier in the network, also referred to as the Simplifier data array.

	Name	Size	Type	Start	End	Description
8	Restart count	4	s32	46	49	Soft reset times of the Simplifier
9	Voltage	8	double	50	57	Input voltage of the Simplifier
10	IO status	2	u16	58	59	ON/OFF status of IOs in the Simplifier
11	IO errors	2	u16	60	61	reserved
12	Internal signals	16	byte array	62	77	Signals used for syncing data
13	CPU1 major	4	s32	78	81	CPU 1 firmware version major
14	CPU1 minor	4	s32	82	85	CPU 1 firmware version minor
15	CPU2 major	4	s32	86	89	CPU 2 firmware version major
16	CPU2 minor	4	s32	90	93	CPU 1 firmware version minor
17	RTC	4	s32	94	97	Time since the Simplifier Started
18	CAN access	2	u16	98	99	reserved
19	Radio access	2	u16	100	101	reserved
20	Configuration ID	4	u32	102	105	reserved
21	Temperature	8	double	106	113	Temperature of Simplifier
22	Node index	4	s32	114	117	Index of Simplifier (starts with 0)

When multiple Simplifiers are connected in the network, their information is serially appended in the same pattern. For example, for the second Simplifier, its restart count would be placed at byte 118, after the node index of the first Simplifier.

3.2 Packet Size and Speed

Naturally larger packets take longer to transmit and therefore can slow the SimplifierGateway down. Since the information in the packet body is not as critical or important as the information in the header, it is not transmitted as frequently, greatly improving the transmission speed.

The ratio of transmission of packet header to packet body is set by the `MessageRatio` parameter.

4 Updating Config and Software

The Simplifier Gateway utilizes our in-house developed tool, SecureBootloader for managing and applying updates. Update files can be put in a USB stick, which can then be connected to the Gateway. After that, simply restarting the Gateway would trigger the update process and the updated Gateway application would run.

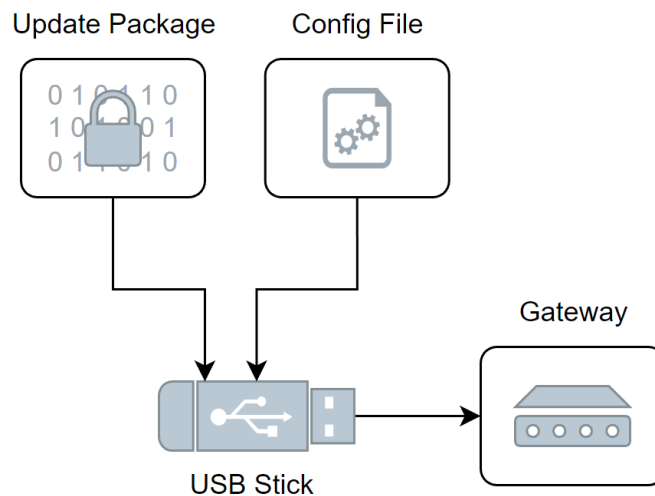
Update files are:

- Gateway app update package
- Config file

The update package and the config file can be obtained from SSP. The config file can also be created by the customer and put into the USB as is. Whichever file is present in the USB would be applied to the Gateway during the update process.

It should be noted that to be detected by the updating system, the name of the USB stick must be `SSP_UPDATE`, the name of the update package must be `ssp_secure_update`, and the name of the config file must be `config.txt`. Although having additional files in the USB stick should not affect the update process, it is strongly recommended to only have the two update files.

If the update process is triggered by connecting the USB and restarting the Gateway, it generally takes 2-3 seconds to complete. After that, the USB stick can be safely removed.



5 PLC Implementation (Siemens TIA V16 or higher)

The implementation of the Simplifier Gateway on the PLC side is linked to a library provided by SSP.

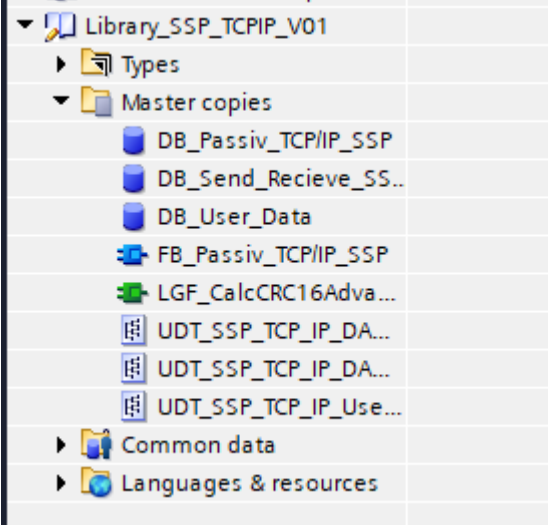
This library contains all blocks and data types to guarantee the data transfer and to read out the evaluated data.

In the following the necessary working steps are explained:

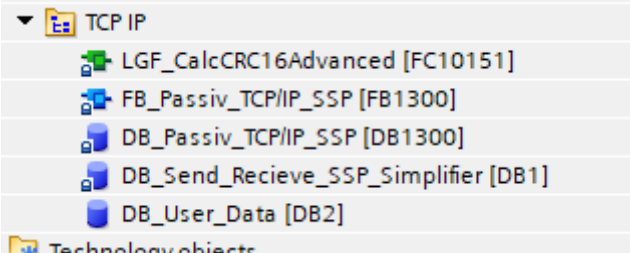
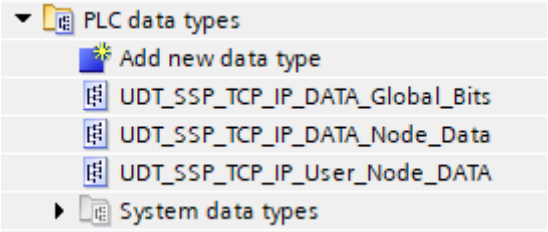
5.1 Opening the library and pasting the blocks and data types into the PLC program

Unpack the library and open it via the tab in your TIA software.

The library contains:

		
DB_Passiv_TCP/IP_SSP	3 data types	
DB_Send_Recieve_SS...		
DB_User_Data	1 function	
FB_Passiv_TCP/IP_SSP		
LGF_CalcCRC16Adva...	1 function block with instance data block	
UDT_SSP_TCP_IP_DA...		
UDT_SSP_TCP_IP_DA...	2 global data blocks	
UDT_SSP_TCP_IP_Use...		
Common data		
Languages & resources		

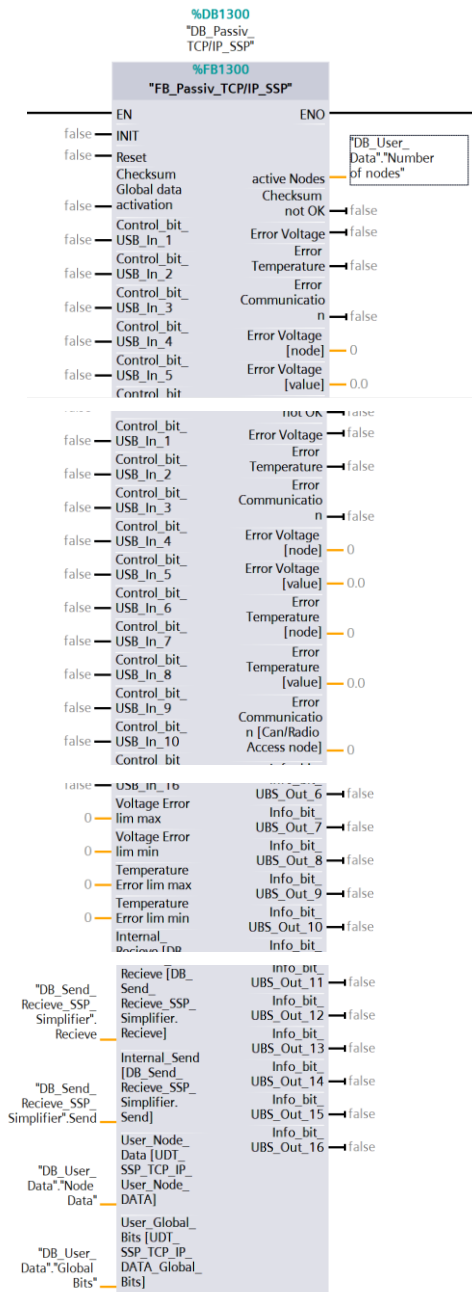
Add all into your project:

		
LGF_CalcCRC16Advanced [FC10151]		
FB_Passiv_TCP/IP_SSP [FB1300]		
DB_Passiv_TCP/IP_SSP [DB1300]		
DB_Send_Recieve_SSP_Simplifier [DB1]		
DB_User_Data [DB2]		
		
UDT_SSP_TCP_IP_DATA_Global_Bits		
UDT_SSP_TCP_IP_DATA_Node_Data		
UDT_SSP_TCP_IP_User_Node_DATA		
System data types		

It's important to choose the same number for "DB_Send_Recieve_SSP_Simplifier" as in the config file on the Gateway itself, to make sure that the connection can be established correctly.

5.2 Call the Function Block

Next step is to call the function block. Given below is an example in OB1 network:



INPUT Declaration:

INIT [Bool]

- Initialization for whole FB Data

Reset [Bool]

- Rest Errors and Error values

Checksum Global Data activation [Bool]

- Activate CRC test for Global Data (16x16 Bit)

Control_bit_USB_In_x (1-16) [16x Bool]

- 16 Inputs directly to connected Simplifier

Min / Max Values for Errors [5x Int]

- Limits for Errors

INOUT Declaration:

Internal Receive [UDT]

- Data type for receive area

Internal Send [UDT]

- Data type for send area

User Node Data [UDT]

- Data type for User Data 1

User Global Data [UDT]

- Data type for User Data 2

OUTPUT Declaration:

Active Nodes [INT]

- Number of nodes in Simplifier configuration

Checksum NOK [Bool]

- Indicator for CRC Checksum NOK

Error indications [3x Bool]

- Error bits for [°C], [V] and communication

Error values & node numbers [INT/Real/INT/Real/INT]

- Values for Error bits

Info_Bit_USB_Out_x (1-16) [16x Bool]

- 16 Outputs directly from connected Simplifier

5.3 User Data

The final step is getting the user data. In the pictures below, a part of the user data can be seen.

▼ Static	
Number of nodes	Int 0
▼ Node Data	"UDT_SSP_TCP_IP_User_Node_DATA"
▼ Data	Array[1..16] of Struct
Data[1]	Struct
Restart_Count	DInt 0
Voltage [V]	Real 0.0
CPU1_Version	Real 0.0
CPU2_Version	Real 0.0
RTC [not used]	DInt 0
Can_Radio_Access [0=off, 1=Can, 2=Radio, 3=both]	Int 0
Configuration_ID [not used]	UDInt 0
Temperature [°C]	Real 0.0
Node_Index [not used]	DInt 0
Data[2]	Struct
Data[3]	Struct
Data[4]	Struct
Data[5]	Struct
Data[6]	Struct
Data[7]	Struct
Data[8]	Struct
Data[9]	Struct
Data[10]	Struct
Data[11]	Struct
Data[12]	Struct
Data[13]	Struct
Data[14]	Struct
Data[15]	Struct
Data[16]	Struct

▼ Global Bits	"UDT_SSP_TCP_IP_DATA_Global_Bits"
Global Memory	Array[1..16, 1..16] of Bool
Global Memory(1..1)	Bool false
Global Memory(1..2)	Bool false
Global Memory(1..3)	Bool false
Global Memory(1..4)	Bool false
Global Memory(1..5)	Bool false
Global Memory(1..6)	Bool false
Global Memory(1..7)	Bool false
Global Memory(1..8)	Bool false
Global Memory(1..9)	Bool false
Global Memory(1..10)	Bool false
Global Memory(1..11)	Bool false
Global Memory(1..12)	Bool false
Global Memory(1..13)	Bool false
Global Memory(1..14)	Bool false
Global Memory(1..15)	Bool false
Global Memory(1..16)	Bool false
Global Memory(2..1)	Bool false
Global Memory(2..2)	Bool false
Global Memory(2..3)	Bool false
Global Memory(2..4)	Bool false
Global Memory(2..5)	Bool false
Global Memory(2..6)	Bool false
Global Memory(2..7)	Bool false
Global Memory(2..8)	Bool false
Global Memory(2..9)	Bool false
Global Memory(2..10)	Bool false
Global Memory(2..11)	Bool false
Global Memory(2..12)	Bool false
Global Memory(2..13)	Bool false
Global Memory(2..14)	Bool false
Global Memory(2..15)	Bool false
Global Memory(2..16)	Bool false
Global Memory(3..1)	Bool false
Global Memory(3..2)	Bool false
Global Memory(3..3)	Bool false
Global Memory(3..4)	Bool false

Every Simplifier has its own Data Area for its node data, which can be copied from here or directly evaluated as values out of the DB.

Additionally, all the memory flags can also be found here. They are shown here in a two-dimensional array (Simplifier number / Memory position) with a refresh rate that was previously set in the Simplifier Gateway.

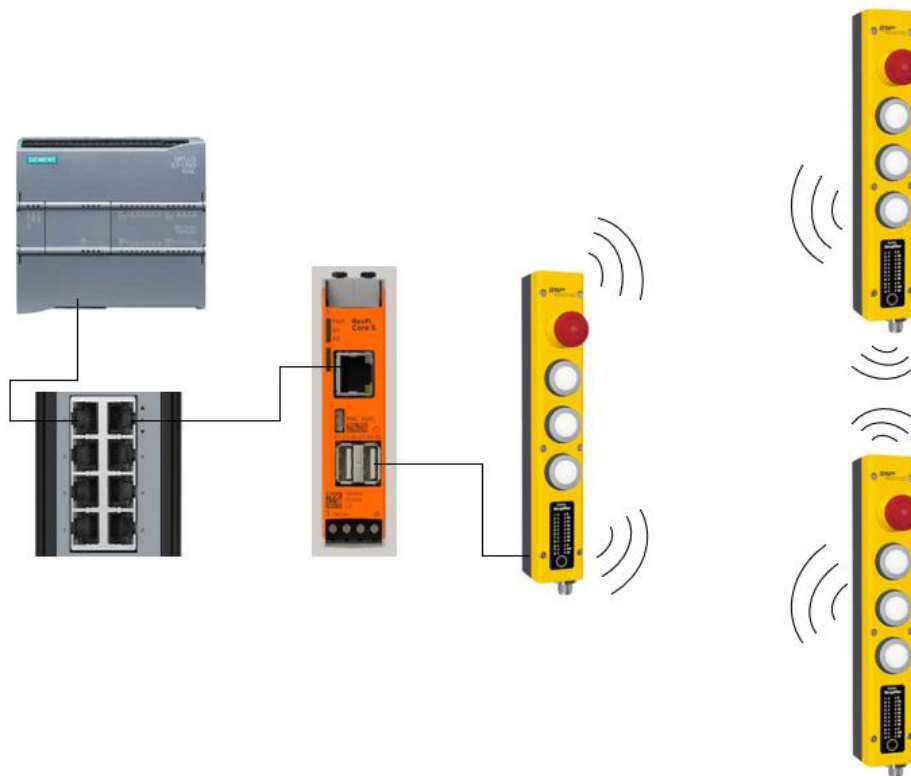
6 Necessary Hardware (Example)

The minimal system consists of a Simplifier Gateway in the middle of the structure and both communication ends.

On one side, there is a Siemens PLC of the 12xx or 15xx series and on the other side a SSP Safety Simplifier.

The PLC side is connected to the gateway at its RJ45 socket with a conventional ProfNet compatible cable to the PLC network. The Simplifier side is connected via a USB connection.

The USB connector is located on the PCB inside the Simplifier. This must be connected to one of the USB ports of the gateway.

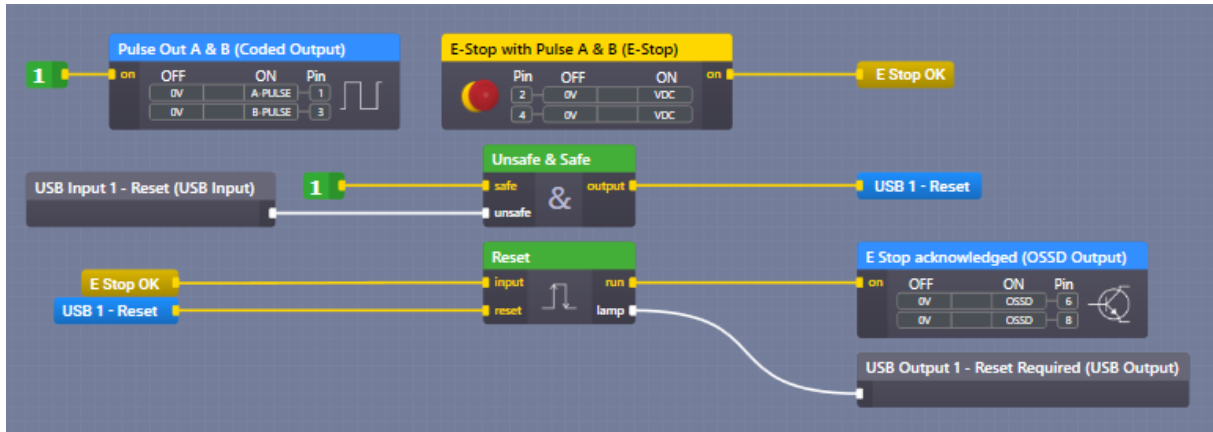


7 Example with Simplifier Manager showing USB Inputs and Outputs

In the picture below, the Safety Simplifier Manager can be seen.

The USB inputs and USB outputs are shown below.

These are easy to use control signals and information in addition to the flag pattern that is mapped in the User Data DB.



USB Input used for Reset of a very easy external pulsed controlled E-Stop.

USB Out used for controlling a message / LED / HMI, to show that an operator must intervene.

8 Common Errors

- All data in the configuration file, which is transferred to the gateway via USB stick, must be valid:
 - Clock time > Timeout
 - DB number < 60000
 - Rack and slot number are given by TIA
 - These are normally "Rack = 0" and "Slot = 1".
- After you have loaded the configuration file, the gateway must be restarted.
- Note that the connection setup takes some time until the gateway communicates.
- In the Siemens configuration the "Put / Get communication" must be enabled, so that the gateway can communicate.
- Make sure that the function "Support simulation during block compilation" is selected at Siemens if you use PLC SIM.