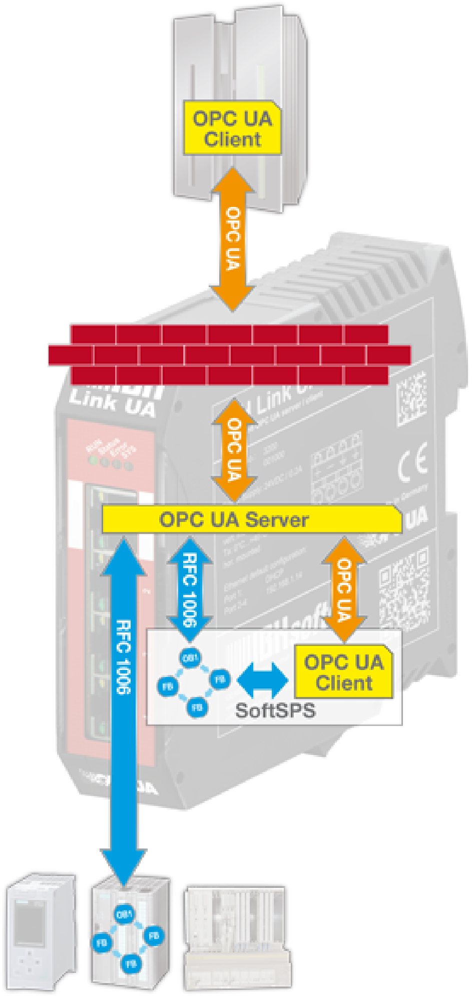
Introduction

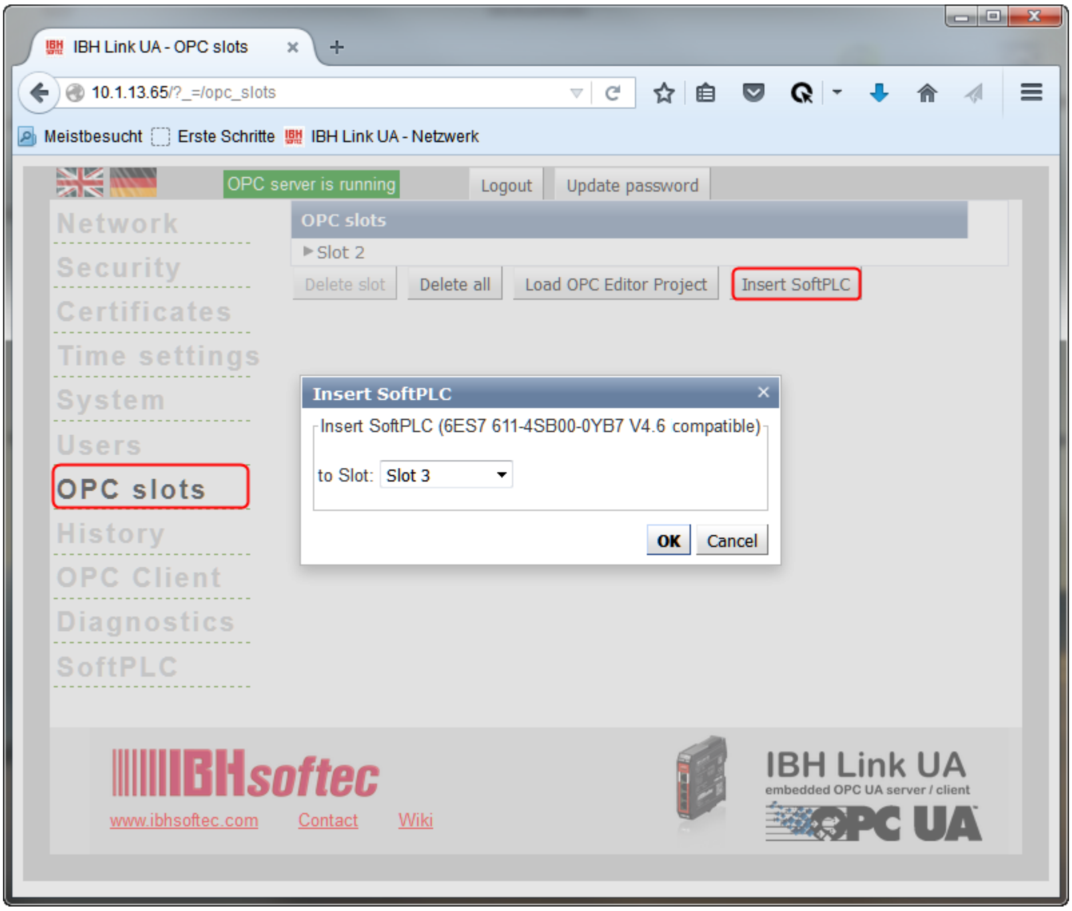
The IBH OPC UA Server/Client additional contains a SoftPLC. The SoftPLC is able to read and write variables from the OPC UA server. The integrated SoftPLC can be used for data preprocessing and supports the programming languages LAD, FBD, STL, SCL or S7-GRAPH.



Commissioning

Activate the integrated SoftPLC

First, the SoftPLC needs to be activated using the web interface:



Insert SoftPLC

Insert SoftPLC (6ES7 611-4SB00-0YB7 V4.6 compatible)

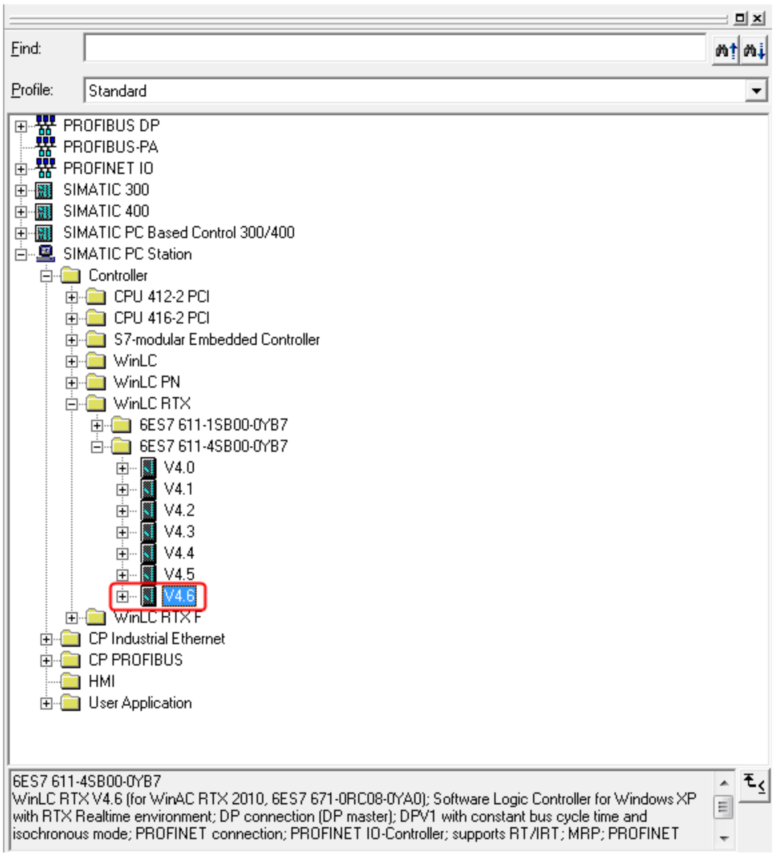
to Slot: Slot 3

OK Cancel

Define SoftPLC within PLC project

Therefore on the defined slot of the PC station a SoftPLC must be added.

SIMATIC Manager:

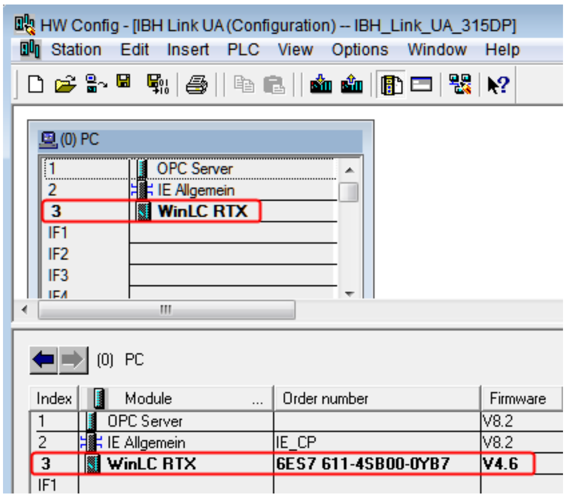


Find:

Profile: Standard

```
PROFIBUS DP
PROFIBUS-PA
PROFINET IO
SIMATIC 300
SIMATIC 400
SIMATIC PC Based Control 300/400
SIMATIC PC Station
  Controller
    CPU 412-2 PCI
    CPU 416-2 PCI
    S7-modular Embedded Controller
    WinLC
    WinLC PN
    WinLC RTX
      6ES7 611-1SB00-0YB7
      6ES7 611-4SB00-0YB7
        V4.0
        V4.1
        V4.2
        V4.3
        V4.4
        V4.5
        V4.6
    WinLC RTX F
  CP Industrial Ethernet
  CP PROFIBUS
  HMI
  User Application
```

6ES7 611-4SB00-0YB7
WinLC RTX V4.6 (for WinAC RTX 2010, 6ES7 671-0RC08-0YA0); Software Logic Controller for Windows XP with RTX Realtime environment; DP connection (DP master); DPV1 with constant bus cycle time and isochronous mode; PROFINET connection; PROFINET IO-Controller; supports RT/IRT; MRP; PROFINET



HW Config - [IBH Link UA (Configuration) -- IBH_Link_UA_315DP]

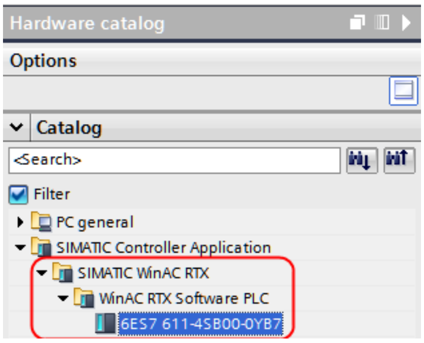
(0) PC

1	OPC Server
2	IE Allgemein
3	WinLC RTX
IF1	
IF2	
IF3	
IF4	

(0) PC

Index	Module	Order number	Firmware
1	OPC Server		V8.2
2	IE Allgemein	IE_CP	V8.2
3	WinLC RTX	6ES7 611-4SB00-0YB7	V4.6
IF1			

TIA Portal:



Hardware catalog

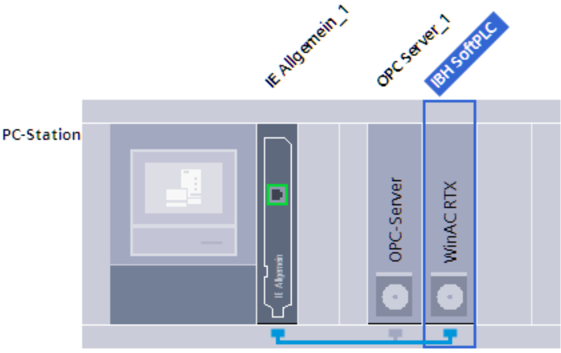
Options

Catalog

<Search>

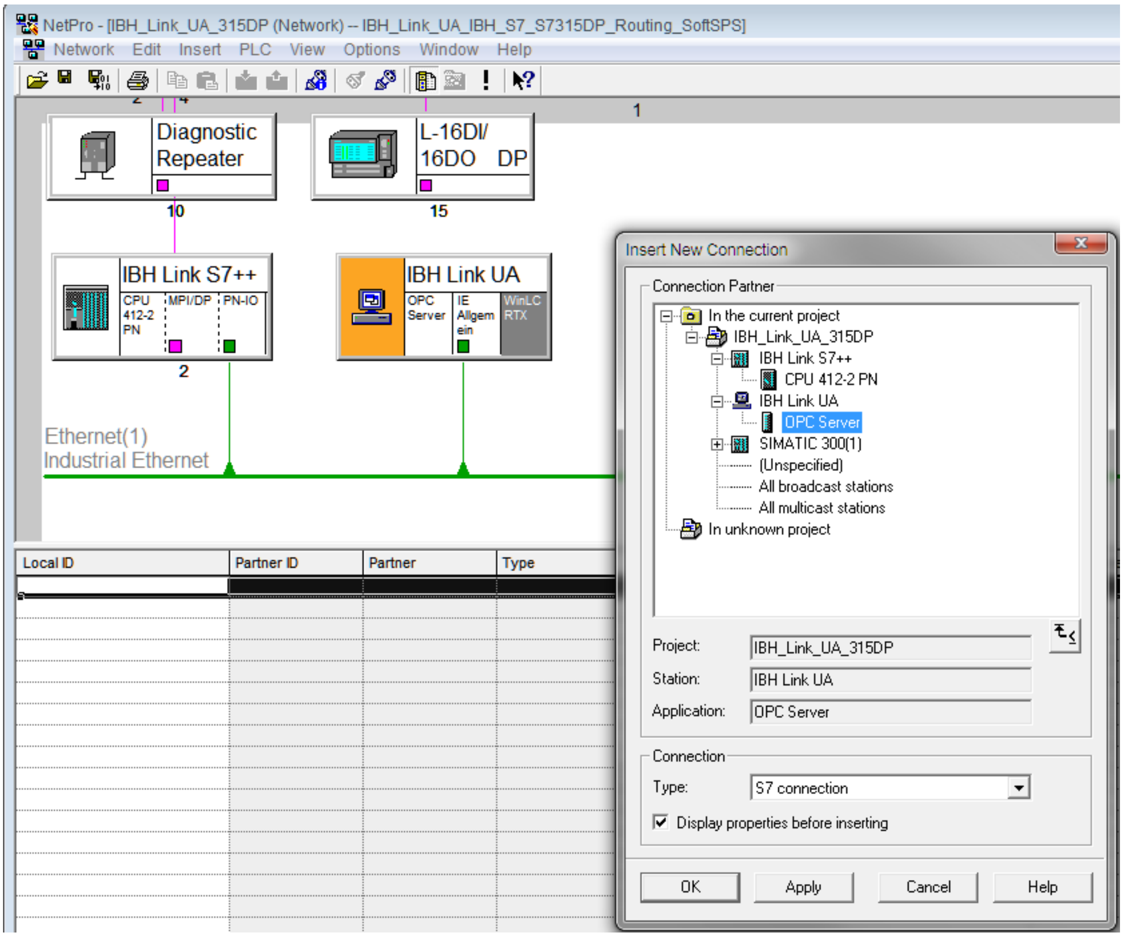
☑ Filter

- PC general
- SIMATIC Controller Application
  - SIMATIC WinAC RTX
    - WinAC RTX Software PLC
      - 6ES7 611-4SB00-0YB7



A S7 connection between the SoftPLC and the OPC UA Server must be established:

SIMATIC Manager:



Insert New Connection

Connection Partner

- In the current project
  - IBH_Link_UA_315DP
    - IBH Link S7++
      - CPU 412-2 PN
    - IBH Link UA
      - OPC Server
    - SIMATIC 300(1)
    - (Unspecified)
    - All broadcast stations
    - All multicast stations
- In unknown project

Project: IBH_Link_UA_315DP

Station: IBH Link UA

Application: OPC Server

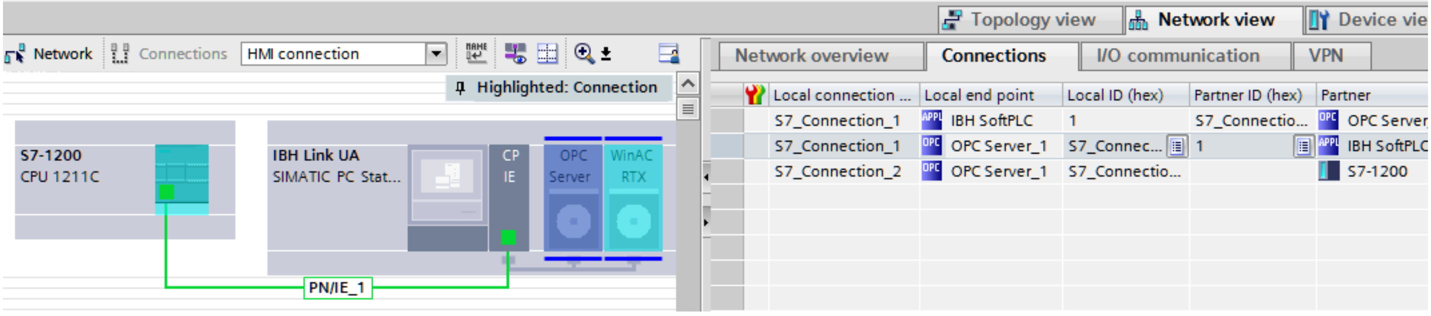
Connection

Type: S7 connection

☑ Display properties before inserting

Local ID	Partner ID	Partner	Type

TIA:

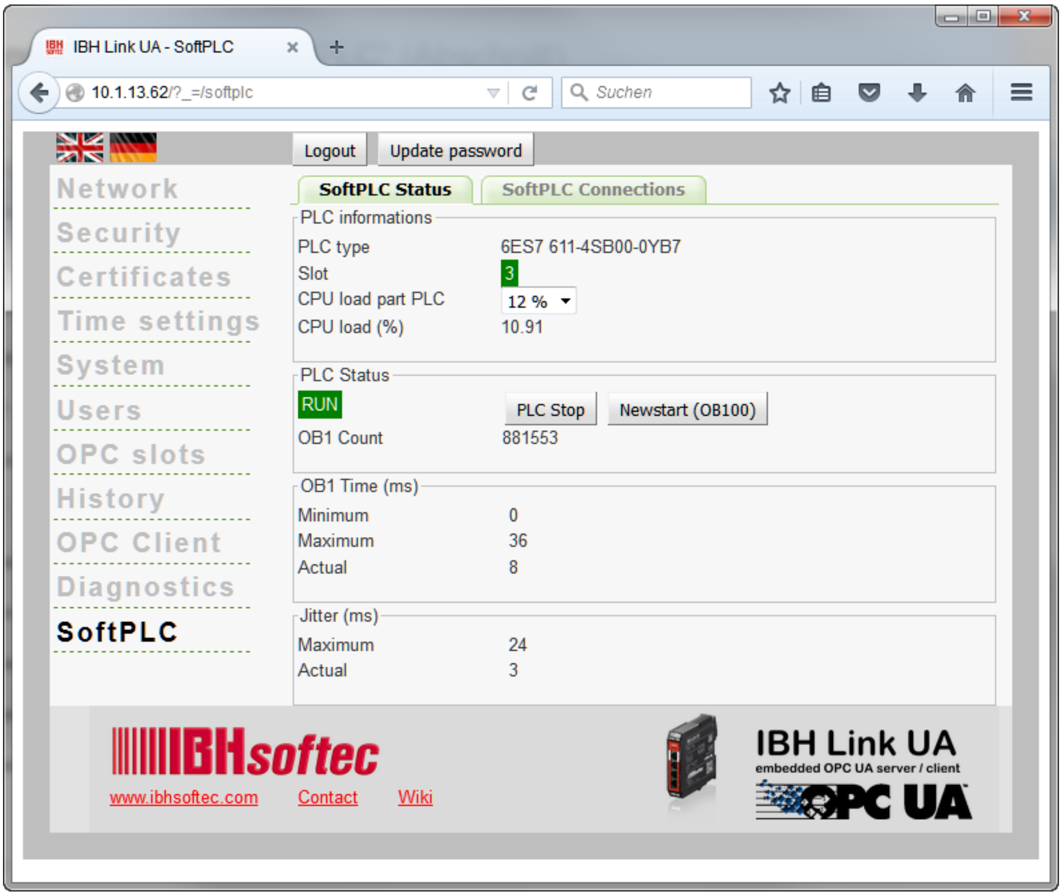


Local connection ...	Local end point	Local ID (hex)	Partner ID (hex)	Partner
S7_Connection_1	IBH SoftPLC	1	S7_Connectio...	OPC Server
S7_Connection_1	OPC Server_1	S7_Connec...	1	IBH SoftPLC
S7_Connection_2	OPC Server_1	S7_Connectio...		S7-1200

The SoftPLC supports the programming languages LAD, FBD, STL, SCL or S7-GRAPH.

SoftPLC Status and Settings

The **SoftPLC Status** can be monitored on the web interface:



PLC informations

PLC type 6ES7 611-4SB00-0YB7

Slot 3

CPU load part PLC 12 %

CPU load (%) 10.91

PLC Status

RUN

OB1 Count 881553

OB1 Time (ms)

Minimum 0

Maximum 36

Actual 8

Jitter (ms)

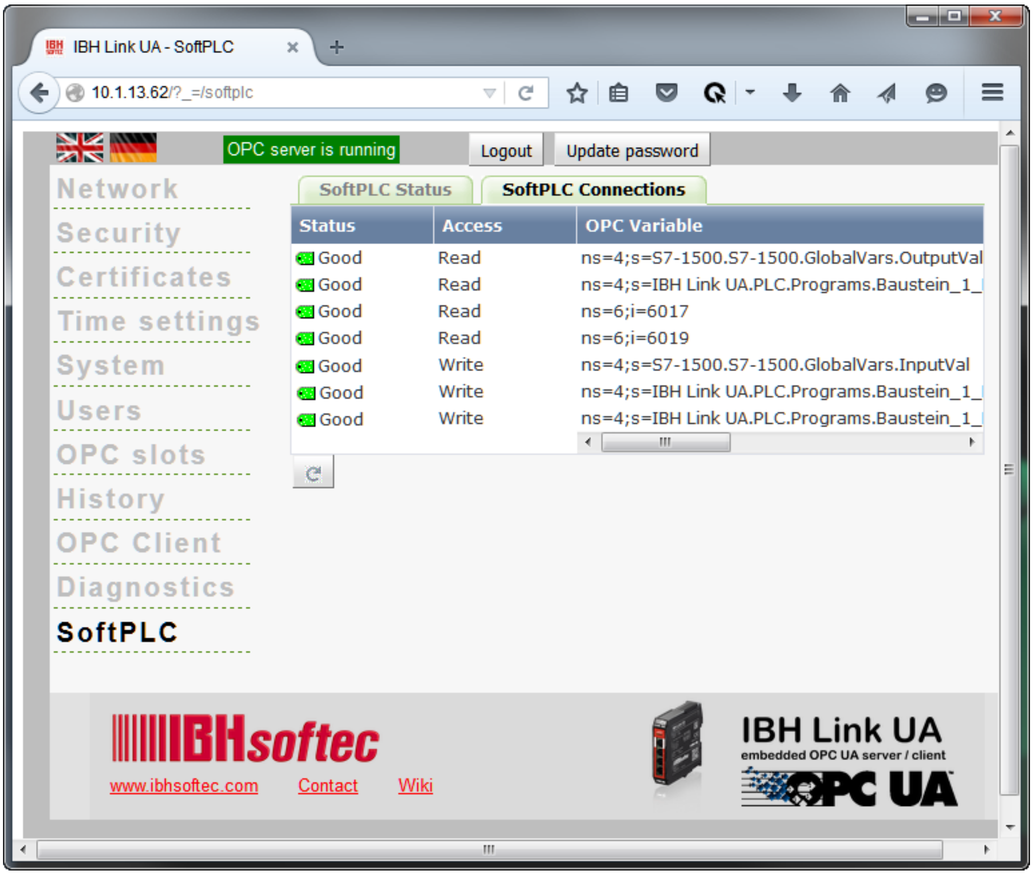
Maximum 24

Actual 3

The performance of the SoftPLC can be adjusted:

CPU Share PLC	processing time (1000 mixed instructions)
50%	apx. 360 µs
33%	apx. 550 µs
25%	apx. 720 µs
20%	apx. 900 µs
12%	apx. 1800 µs

Furthermore, the status of the **SoftPLC Connections** is available.:



IBH Link UA - SoftPLC

10.1.13.62/?_=/softplc

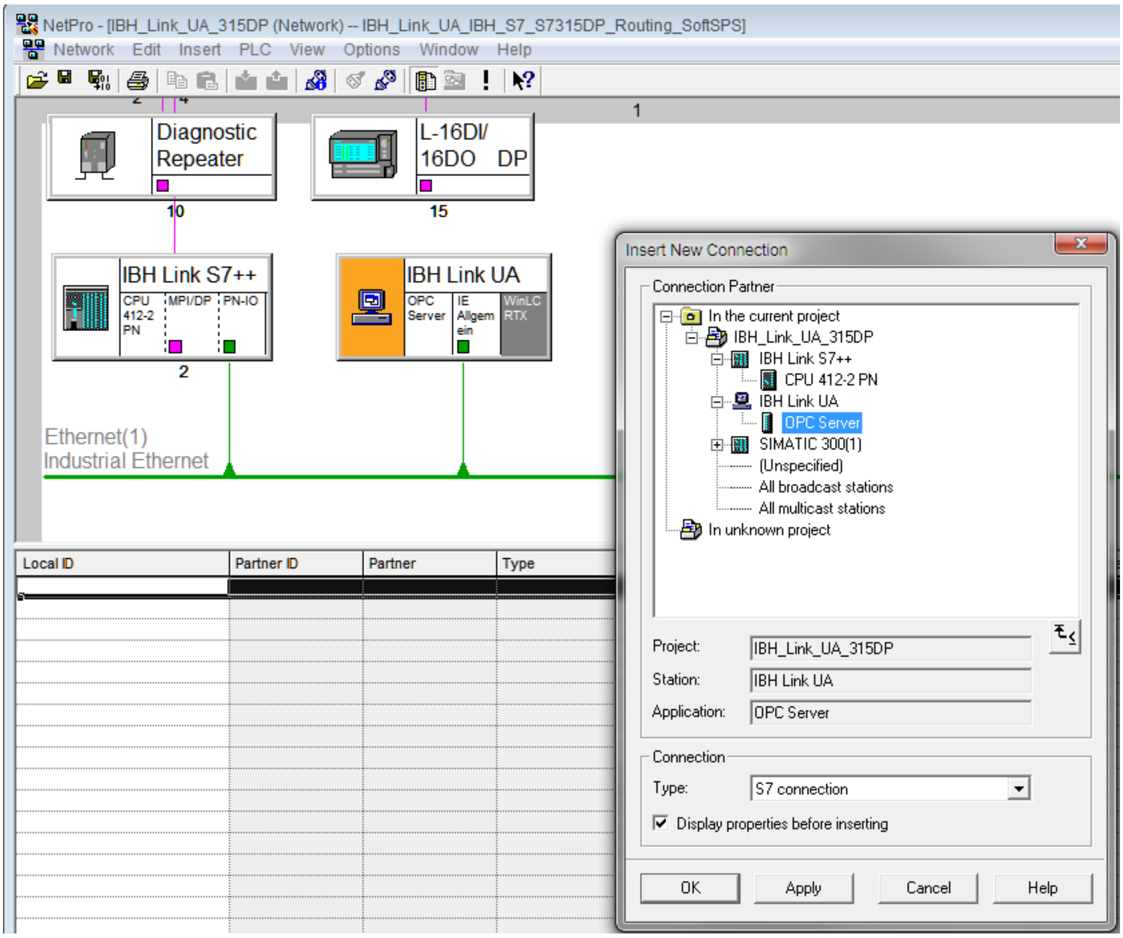
OPC server is running

Status	Access	OPC Variable
Good	Read	ns=4;s=S7-1500.S7-1500.GlobalVars.OutputVal
Good	Read	ns=4;s=IBH Link UA.PLC.Programs.Baustein_1_
Good	Read	ns=6;i=6017
Good	Read	ns=6;i=6019
Good	Write	ns=4;s=S7-1500.S7-1500.GlobalVars.InputVal
Good	Write	ns=4;s=IBH Link UA.PLC.Programs.Baustein_1_
Good	Write	ns=4;s=IBH Link UA.PLC.Programs.Baustein_1_

Server Function

A S7 connection between the SoftPLC and the OPC UA Server must be established::

SIMATIC Manager:



NetPro - [IBH_Link_UA_315DP (Network) -- IBH_Link_UA_IBH_S7_S7315DP_Routing_SoftSPS]

Insert New Connection

Connection Partner

- In the current project
  - IBH_Link_UA_315DP
    - IBH Link S7++
      - CPU 412-2 PN
    - IBH Link UA
      - OPC Server
    - SIMATIC 300(1)
    - (Unspecified)
    - All broadcast stations
    - All multicast stations
- In unknown project

Project: IBH_Link_UA_315DP

Station: IBH Link UA

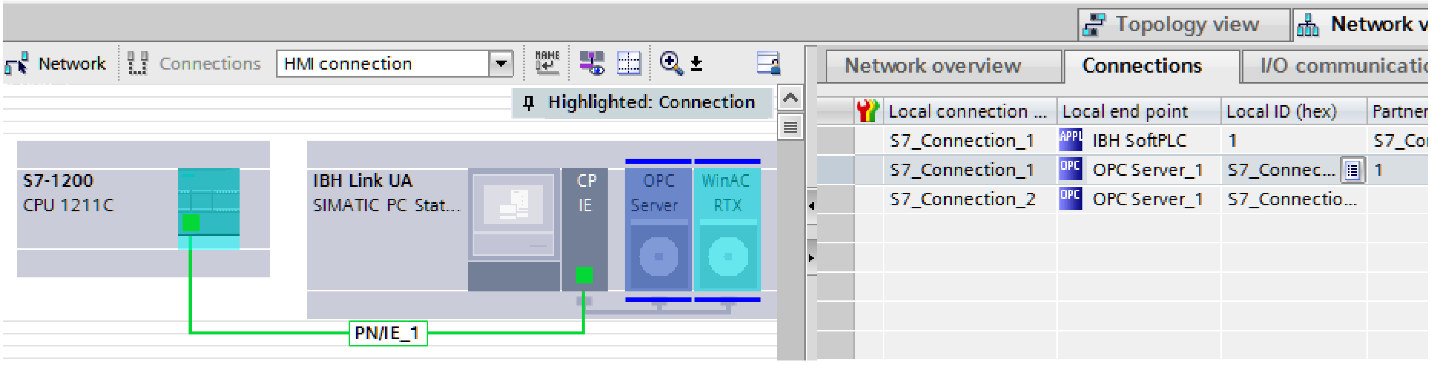
Application: OPC Server

Connection

Type: S7 connection

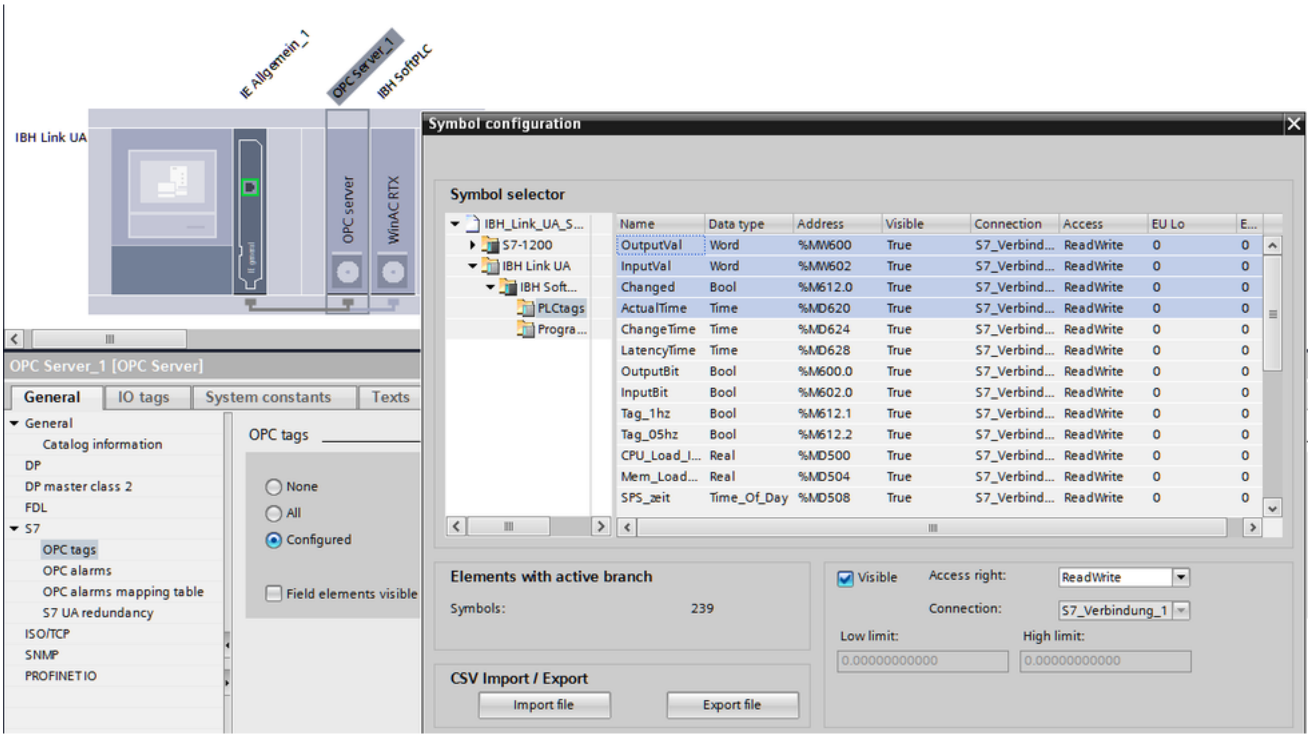
☑ Display properties before inserting

TIA:



Local connection ...	Local end point	Local ID (hex)	Partner
S7_Connection_1	IBH SoftPLC	1	S7_Co
S7_Connection_1	OPC Server_1	S7_Connec...	1
S7_Connection_2	OPC Server_1	S7_Connectio...	

Within the properties of the OPC Server at <**General**> <**S7**> <**OPC tags**> the OPC Tags can be selected:



OPC tags

○ None
○ All
● Configured

□ Field elements visible

Name	Data type	Address	Visible	Connection	Access	EU Lo	E...
OutputVal	Word	%MW600	True	S7_Verbind...	ReadWrite	0	0
InputVal	Word	%MW602	True	S7_Verbind...	ReadWrite	0	0
Changed	Bool	%M612.0	True	S7_Verbind...	ReadWrite	0	0
ActualTime	Time	%MD620	True	S7_Verbind...	ReadWrite	0	0
ChangeTime	Time	%MD624	True	S7_Verbind...	ReadWrite	0	0
LatencyTime	Time	%MD628	True	S7_Verbind...	ReadWrite	0	0
OutputBit	Bool	%M600.0	True	S7_Verbind...	ReadWrite	0	0
InputBit	Bool	%M602.0	True	S7_Verbind...	ReadWrite	0	0
Tag_1hz	Bool	%M612.1	True	S7_Verbind...	ReadWrite	0	0
Tag_05hz	Bool	%M612.2	True	S7_Verbind...	ReadWrite	0	0
CPU_Load_I...	Real	%MD500	True	S7_Verbind...	ReadWrite	0	0
Mem_Load...	Real	%MD504	True	S7_Verbind...	ReadWrite	0	0
SPS_zeit	Time_Of_Day	%MD508	True	S7_Verbind...	ReadWrite	0	0

Elements with active branch

Symbols: 239

☑ Visible

Access right: ReadWrite

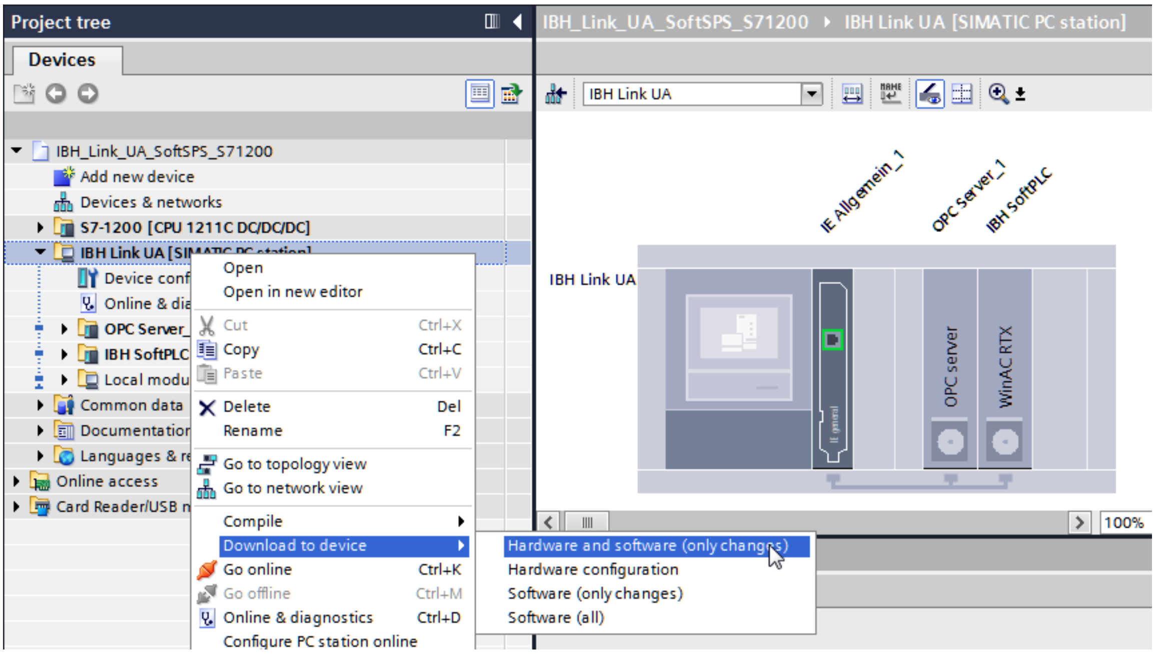
Connection: S7_Verbindung_1

Low limit: 0.00000000000

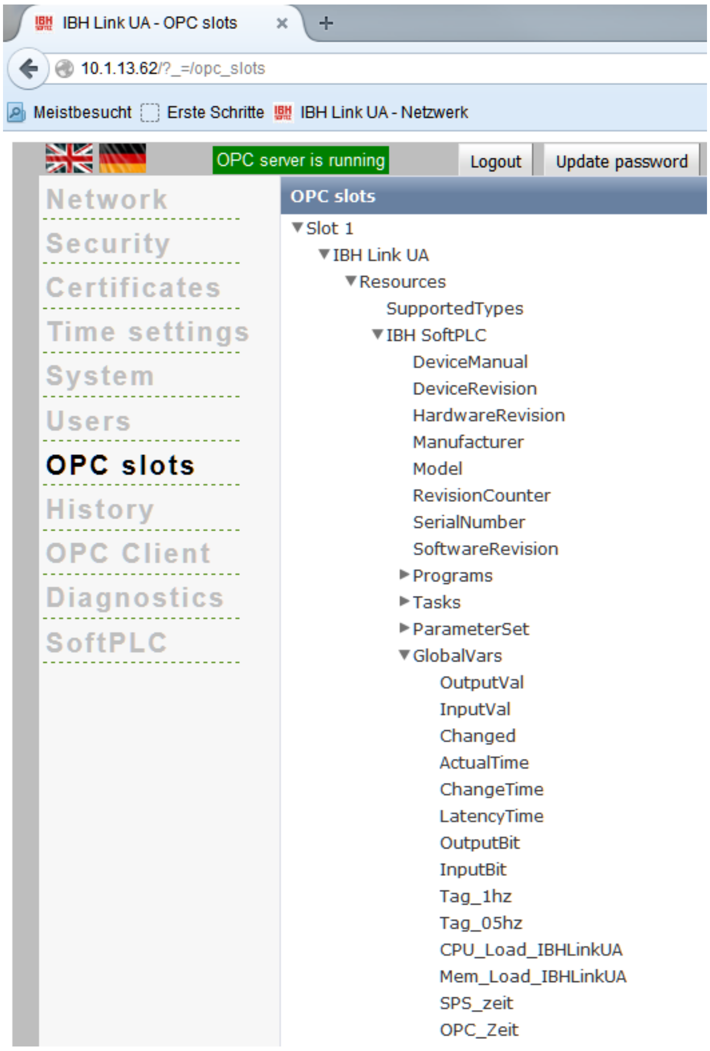
High limit: 0.00000000000

CSV Import / Export

Now the configuration needs to be downloaded to the IBH OPC UA Server/Client:



The configuration is now completed and the tags from the **SoftPLC** within the IBH OPC UA Server/Client are available :



IBH Link UA - OPC slots

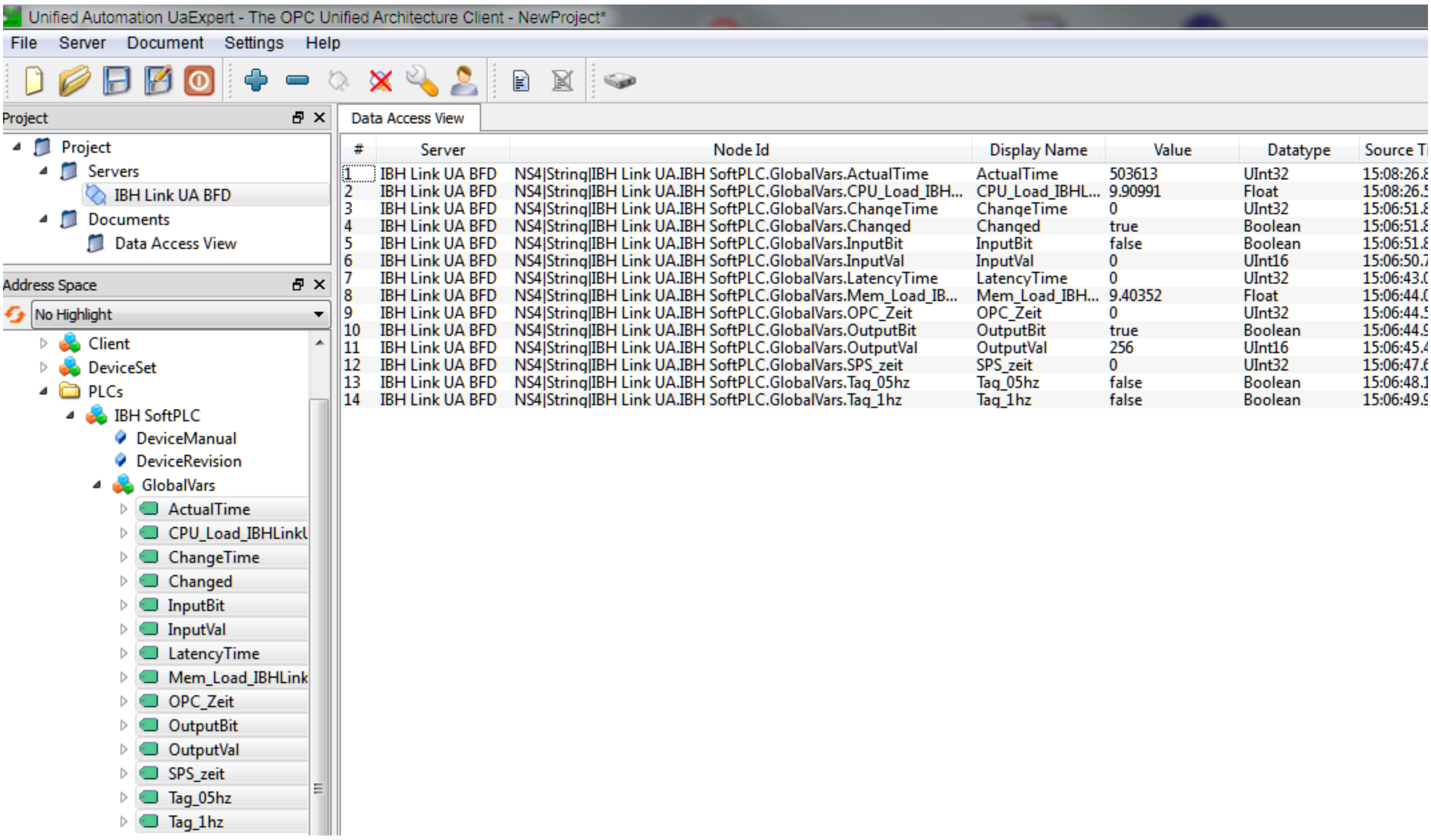
10.1.13.62/?_=/opc_slots

Meistbesucht · Erste Schritte · IBH Link UA - Netzwerk

OPC server is running · Logout · Update password

OPC slots

```
▼Slot 1
  ▼IBH Link UA
    ▼Resources
       SupportedTypes
      ▼IBH SoftPLC
         DeviceManual
         DeviceRevision
         HardwareRevision
         Manufacturer
         Model
         RevisionCounter
         SerialNumber
         SoftwareRevision
        ▶Programs
        ▶Tasks
        ▶ParameterSet
        ▼GlobalVars
           OutputVal
           InputVal
           Changed
           ActualTime
           ChangeTime
           LatencyTime
           OutputBit
           InputBit
           Tag_1hz
           Tag_05hz
           CPU_Load_IBHLinkUA
           Mem_Load_IBHLinkUA
           SPS_zeit
           OPC_Zeit
```



Unified Automation UaExpert - The OPC Unified Architecture Client - NewProject*

Project

```
Project
  Servers
    IBH Link UA BFD
  Documents
    Data Access View
```

Address Space

No Highlight

```
Client
DeviceSet
PLCs
  IBH SoftPLC
    DeviceManual
    DeviceRevision
    GlobalVars
      ActualTime
      CPU_Load_IBHLinkU
      ChangeTime
      Changed
      InputBit
      InputVal
      LatencyTime
      Mem_Load_IBHLink
      OPC_Zeit
      OutputBit
      OutputVal
      SPS_zeit
      Tag_05hz
      Tag_1hz
```

Data Access View

#	Server	Node Id	Display Name	Value	Datatype	Source Ti
1	IBH Link UA BFD	NS4|String|IBH Link UA.IBH SoftPLC.GlobalVars.ActualTime	ActualTime	503613	UInt32	15:08:26.8
2	IBH Link UA BFD	NS4|String|IBH Link UA.IBH SoftPLC.GlobalVars.CPU_Load_IBH...	CPU_Load_IBHL...	9.90991	Float	15:08:26.5
3	IBH Link UA BFD	NS4|String|IBH Link UA.IBH SoftPLC.GlobalVars.ChangeTime	ChangeTime	0	UInt32	15:06:51.8
4	IBH Link UA BFD	NS4|String|IBH Link UA.IBH SoftPLC.GlobalVars.Changed	Changed	true	Boolean	15:06:51.8
5	IBH Link UA BFD	NS4|String|IBH Link UA.IBH SoftPLC.GlobalVars.InputBit	InputBit	false	Boolean	15:06:51.8
6	IBH Link UA BFD	NS4|String|IBH Link UA.IBH SoftPLC.GlobalVars.InputVal	InputVal	0	UInt16	15:06:50.7
7	IBH Link UA BFD	NS4|String|IBH Link UA.IBH SoftPLC.GlobalVars.LatencyTime	LatencyTime	0	UInt32	15:06:43.0
8	IBH Link UA BFD	NS4|String|IBH Link UA.IBH SoftPLC.GlobalVars.Mem_Load_IB...	Mem_Load_IBH...	9.40352	Float	15:06:44.0
9	IBH Link UA BFD	NS4|String|IBH Link UA.IBH SoftPLC.GlobalVars.OPC_Zeit	OPC_Zeit	0	UInt32	15:06:44.5
10	IBH Link UA BFD	NS4|String|IBH Link UA.IBH SoftPLC.GlobalVars.OutputBit	OutputBit	true	Boolean	15:06:44.9
11	IBH Link UA BFD	NS4|String|IBH Link UA.IBH SoftPLC.GlobalVars.OutputVal	OutputVal	256	UInt16	15:06:45.4
12	IBH Link UA BFD	NS4|String|IBH Link UA.IBH SoftPLC.GlobalVars.SPS_zeit	SPS_zeit	0	UInt32	15:06:47.6
13	IBH Link UA BFD	NS4|String|IBH Link UA.IBH SoftPLC.GlobalVars.Tag_05hz	Tag_05hz	false	Boolean	15:06:48.1
14	IBH Link UA BFD	NS4|String|IBH Link UA.IBH SoftPLC.GlobalVars.Tag_1hz	Tag_1hz	false	Boolean	15:06:49.9

Client Function

Cyclic Read and Write

The SoftPLC is able to read and write the OPC variables cyclically.
The configuration will be done within the SFB 8 (USEND) and called during the warm start (OB100).

Parameter	Declaration	Data type	Storage area	description
REQ	INPUT	BOOL	E, A, M, D, L	Not evaluated
ID	INPUT	WORD	M, D, Const.	Fixed value: 65400
R_ID	INPUT	DWORD	E, A, M, D, L, Const.	Mode Low Byte: 0: Read(Variable as XML String) 1: Write(Variable as XML String) 2: Read(PLC Variable) 3: Write(PLC Variable) 4: Read(Special Variable) 5: Write(Special Variable) 6: Read(Server Variable) 7: Write(Server Variable) Flag 14:While writing the Any Pointer points to DATE_AND_TIME Flyg 15: While writing the Any Pointer points to a STRING High Word: Sampling in milliseconds
DONE	OUTPUT	BOOL	E, A, M, D, L	is not set

ERROR	OUTPUT	BOOL	E, A, M, D, L	ERROR=0:
				The parameters were passed correctly
				ERROR=1:
				The parameters were passed not correctly

STATUS	OUTPUT	WORD	E, A, M, D, L	0x0000 on success, 0x8090 on error.
SD_1	IN_OUT	ANY	D	Pointer to the OPC Variable.
SD_2	IN_OUT	ANY	E, A, M, D	Pointer to the OPC Variable for the value of the OPC Variable. Allowed is Data Type BOOL, BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, DATE_AND_TIME. **Note**: If the ANY pointer points to a DB, the DB must always be specified (f.i.: P# DB10.DBX5.0 Byte 0).
SD_3	IN_OUT	ANY	E, A, M, D	Pointer to the PLC Variable for the status of the Variable. Allowed is Data Type DWORD **Note:** If the ANY pointer points to a DB, the DB must always be specified (f.i.: P# DB10.DBX5.0 Byte 10).
SD_4	IN_OUT	ANY	D	Pointer to the PLC Variable for the Time Stamp. Allowed is Data Type DATE_AND_TIME. **Note:** If the ANY pointer points to a DB, the DB must always be specified (f.i.: P# DB10.DBX5.0 Byte 10).

The OPC Variables are defined as follows:

Mode 0 and **1: ´ns=<Namespace>;s=<Identifier>´**

or: **ns=<Namespace>;i=<Numeric Identifier>´**

All OPC Variables can be read using this mode. The IBH Link UA knows the following Namespaces

Namespace	Area
0	General OPC Server Variables
1, 2, 3, 5	No evaluable variables
4	All PLC specific OPC variables
6	IBH Link UA Special Variables

Mode 2 and **3: ´<Identifier>´**

The identifier is formed as follows:
<Station name>.<PLC Name>.<GlobalVars>.<Variable name from Variables table>
or:
<Station name>.<PLC Name>.<Programs>..<Variable name>

Mode 4 and **5: <Numeric Identifier>**

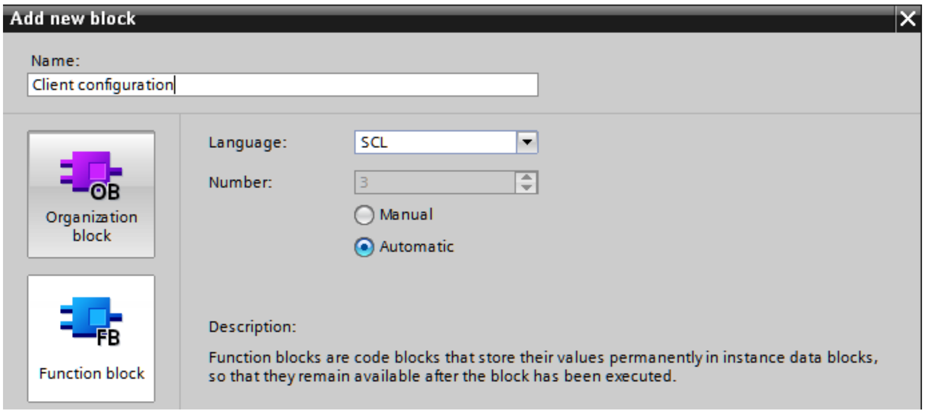
Number of the Special Variable from Namespace 6.

Modus 6 and **7: <Numeric Identifier>**

Number of the Server Variable from Namespace 0.

Sample TIA Portal

Create a function block for the client configuration:


Add new block

Name: Client configuration

Language: SCL

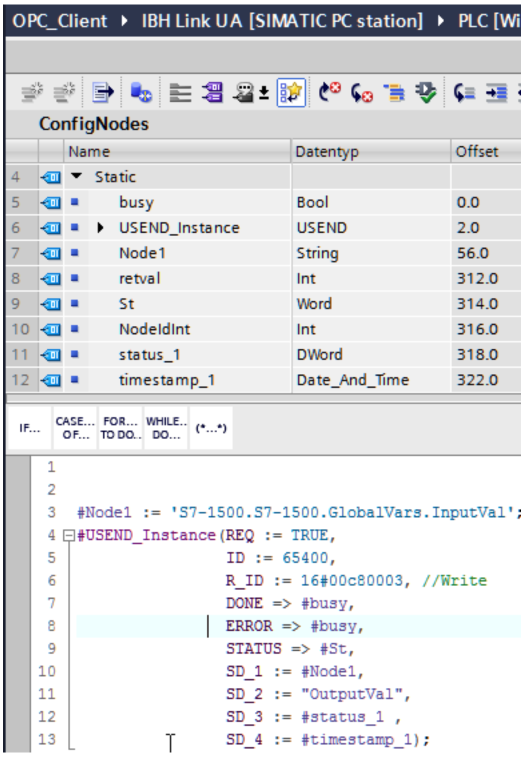
Number: 3

○ Manual

● Automatic

Description:

Function blocks are code blocks that store their values permanently in instance data blocks, so that they remain available after the block has been executed.


OPC_Client ▸ IBH Link UA [SIMATIC PC station] ▸ PLC [Wi

ConfigNodes

	Name	Datentyp	Offset
4	▾ Static		
5	busy	Bool	0.0
6	▸ USEND_Instance	USEND	2.0
7	Node1	String	56.0
8	retval	Int	312.0
9	St	Word	314.0
10	NodeIdInt	Int	316.0
11	status_1	DWord	318.0
12	timestamp_1	Date_And_Time	322.0

```


#Node1 := 'S7-1500.S7-1500.GlobalVars.InputVal';
#USEND_Instance(REQ := TRUE,
                ID := 65400,
                R_ID := 16#00c80003, //Write
                DONE => #busy,
                ERROR => #busy,
                STATUS => #St,
                SD_1 := #Node1,
                SD_2 := "OutputVal",
                SD_3 := #status_1 ,
                SD_4 := #timestamp_1);
```

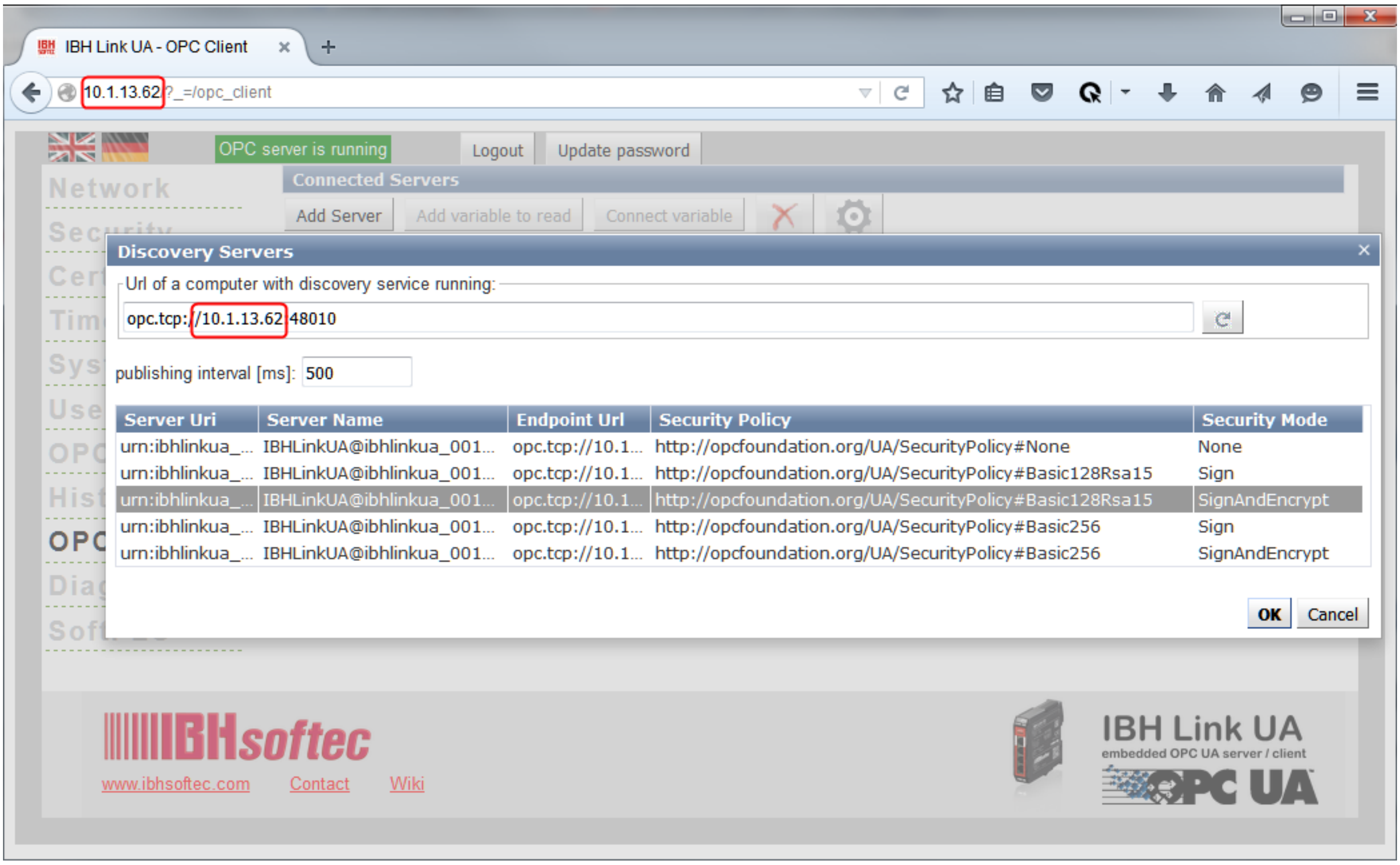
The PLC Variable **OutputVal** will be written cyclically into the OPC Variable **S7-1500.S7-1500.GlobalVars.InputVal**.
The sampling rate is 16#00c8 or 200 milliseconds.
The OPC Status is written in the same cycle as **Output Val** into the variable **# Status_1** and the time stamp of the cycle into the variable **# timestamp_1** .
Now the configuration block must be called within OB100.

Server Bridge

The basic procedure is described at IBH OPC UA Client Functions .

The server, between which a communication should be established, need to be defined.

First, the own server URL of the IBH OPC UA Server/Client is specified:



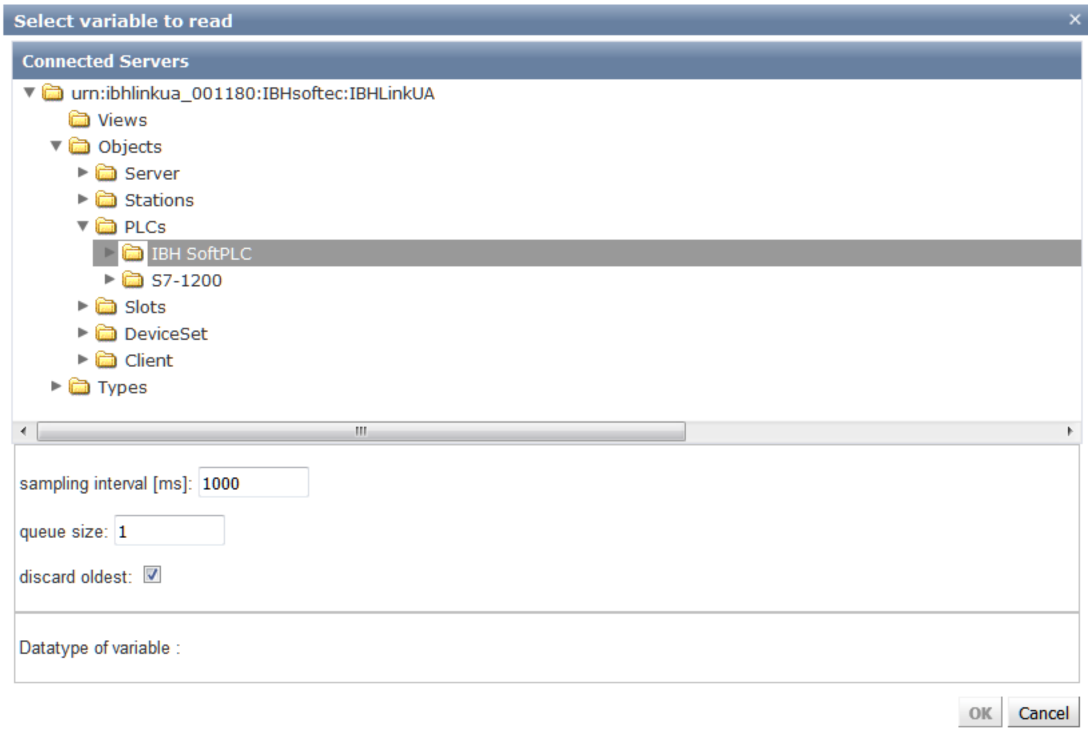
Discovery Servers

Url of a computer with discovery service running: opc.tcp://10.1.13.62:48010

publishing interval [ms]: 500

Server Uri	Server Name	Endpoint Url	Security Policy	Security Mode
urn:ibhlinkua_...	IBHLinkUA@ibhlinkua_001...	opc.tcp://10.1...	http://opcfoundation.org/UA/SecurityPolicy#None	None
urn:ibhlinkua_...	IBHLinkUA@ibhlinkua_001...	opc.tcp://10.1...	http://opcfoundation.org/UA/SecurityPolicy#Basic128Rsa15	Sign
urn:ibhlinkua_...	IBHLinkUA@ibhlinkua_001...	opc.tcp://10.1...	http://opcfoundation.org/UA/SecurityPolicy#Basic128Rsa15	SignAndEncrypt
urn:ibhlinkua_...	IBHLinkUA@ibhlinkua_001...	opc.tcp://10.1...	http://opcfoundation.org/UA/SecurityPolicy#Basic256	Sign
urn:ibhlinkua_...	IBHLinkUA@ibhlinkua_001...	opc.tcp://10.1...	http://opcfoundation.org/UA/SecurityPolicy#Basic256	SignAndEncrypt

Furthermore, the desired security level can be chosen.
Now the variables of the SoftPLC can be selected for reading:



Select variable to read

Connected Servers

- urn:ibhlinkua_001180:IBHsoftec:IBHLinkUA
  - Views
  - Objects
    - Server
    - Stations
    - PLCs
      - IBH SoftPLC
      - S7-1200
    - Slots
    - DeviceSet
    - Client
  - Types

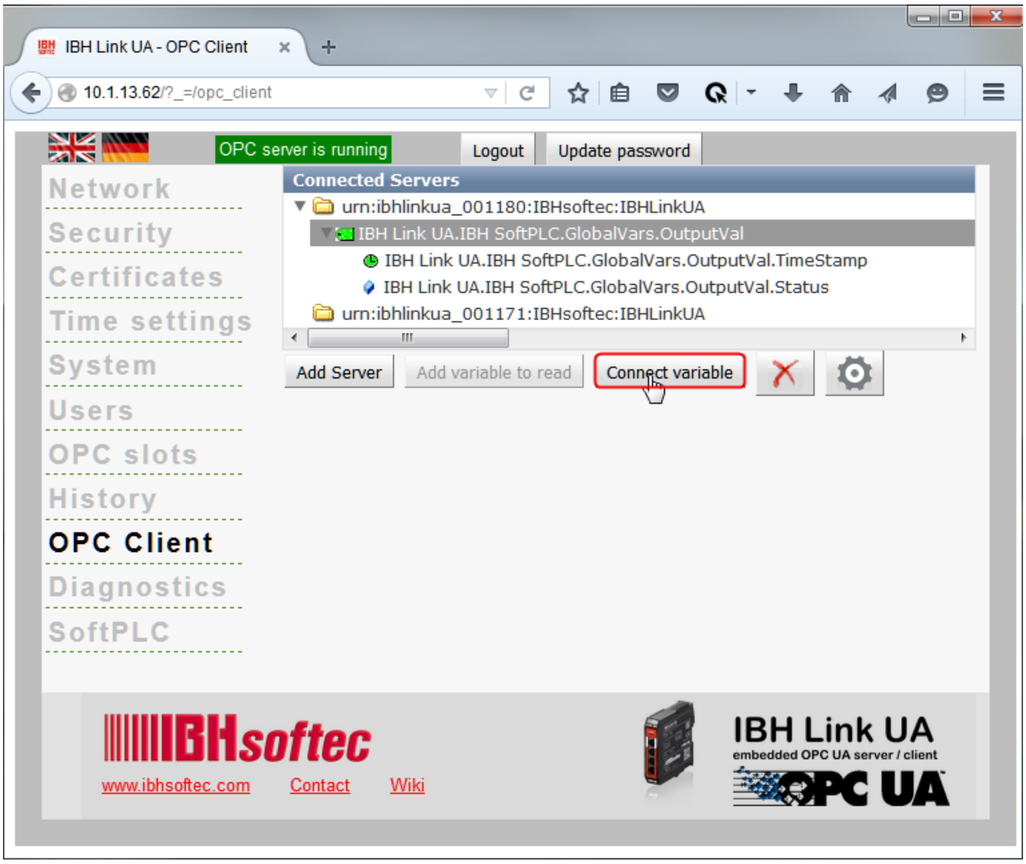
sampling interval [ms]: 1000

queue size: 1

discard oldest: ☑

Datatype of variable :

and can be connected to a variable of another OPC UA Server:



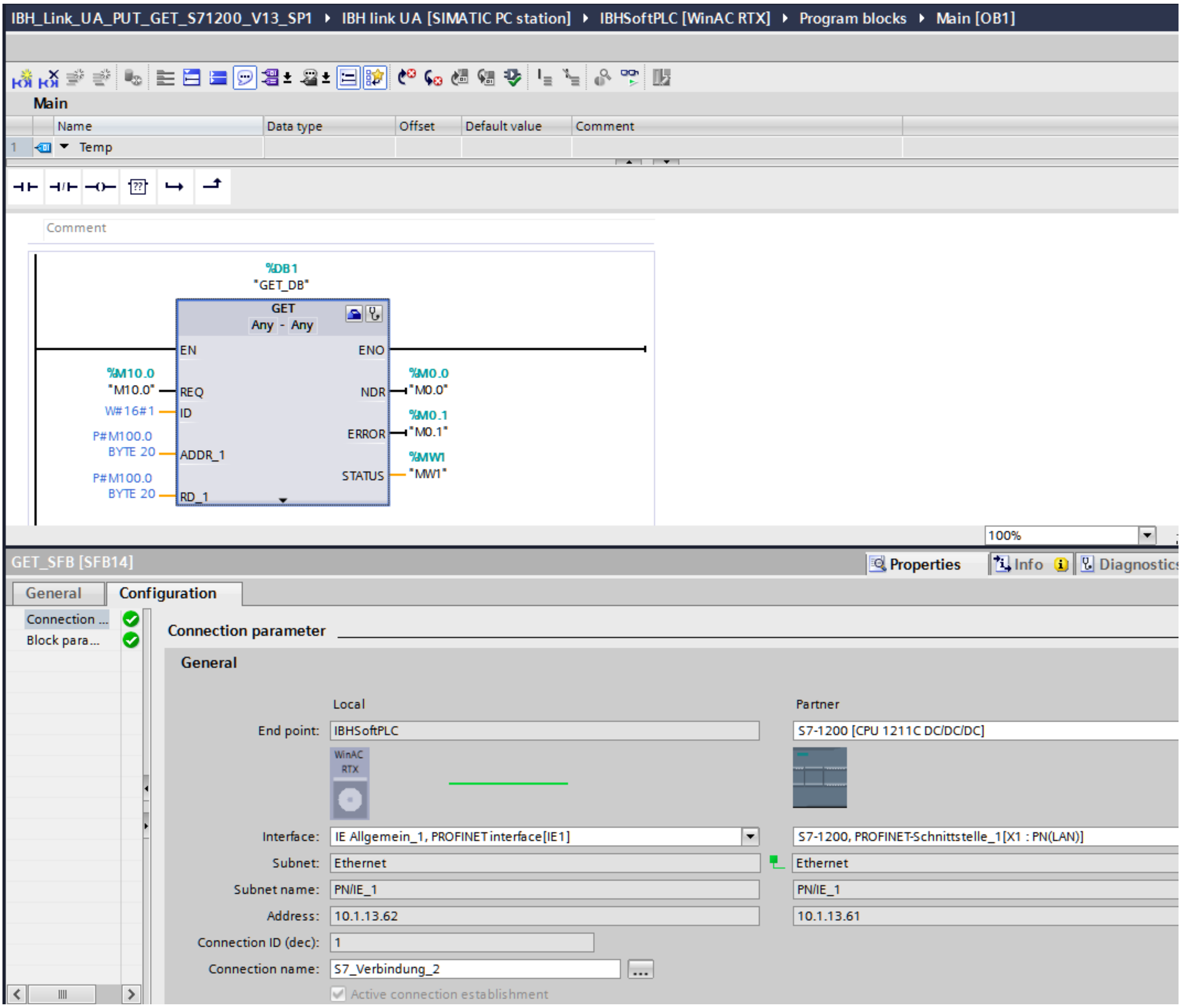
IBH Link UA - OPC Client
10.1.13.62/?_=/opc_client
OPC server is running
Logout
Update password
Network
Security
Certificates
Time settings
System
Users
OPC slots
History
OPC Client
Diagnostics
SoftPLC
Connected Servers
urn:ibhlinkua_001180:IBHsoftec:IBHLinkUA
IBH Link UA.IBH SoftPLC.GlobalVars.OutputVal
IBH Link UA.IBH SoftPLC.GlobalVars.OutputVal.TimeStamp
IBH Link UA.IBH SoftPLC.GlobalVars.OutputVal.Status
urn:ibhlinkua_001171:IBHsoftec:IBHLinkUA
Add Server
Add variable to read
Connect variable
IBHsoftec
www.ibhsoftec.com
Contact
Wiki
IBH Link UA
embedded OPC UA server / client
OPC UA

PUT / GET Functions

Configured Connection

The PUT(SFB 15) and GET(SFB 14) blocks can be used to read and write data directly from a S7 controller with the SoftPLC integrated in the IBH Link UA. Data can be read or written from freely addressable data areas of the partner PLC.

For this purpose, a S7 connection from the IBH SoftPLC to the desired PLC can be configured within the Simatic Manager or the TIA Portal:



Connection parameter

General

	Local	Partner
End point:	IBHSoftPLC	S7-1200 [CPU 1211C DC/DC/DC]
Interface:	IE Allgemein_1, PROFINET interface[IE1]	S7-1200, PROFINET-Schnittstelle_1[X1 : PN(LAN)]
Subnet:	Ethernet	Ethernet
Subnet name:	PN/IE_1	PN/IE_1
Address:	10.1.13.62	10.1.13.61
Connection ID (dec):	1	
Connection name:	S7_Verbindung_2	

☑ Active connection establishment

Download: **TIA Sample: IBH Link UA SoftPLC PUT GET S7-1200**

SFB15 PUT Special function

The SFB15 PUT can also be used to enter own S7 connections on top of the connections already made with STEP7 or TIA. To do this, the SFB15 PUT with the special ID 65400 is called within the OB100("Complete restart"). The ADDR and SD fields are then used to specify the connection information.

Usage: To access another PLC on Rack 0 Slot 2, only the IP address and the number of the new local ID are required.

```
    CALL  PUT , "InstanceDbPut"
    REQ : = TRUE
    ID : = 65400
    DONE : = "DoneFlag"
    ERROR : = "ErrorFlag"
    STATUS : = "StatusWord"
    ADDR_1 : = P#DB10.DBX0.0 BYTE 20  // IP address or host name as character array in the form "192.168.1.22"
    ADDR_2 : =
    ADDR_3 : =
    ADDR_4 : = P#DB10.DBX24.0 WORD 1  // If another port than 102 is used, it can be specified as WORD
    SD_1 : = P#DB10.DBX26.0 WORD 1          // Number of the new, free local ID as WORD
    SD_2 : =
    SD_3 : =
    SD_4 : =
```

Usage: To access another PLC on Rack 0 Slot 1 (wthin our sample a S7-1200), not only the IP address and the number of the new local ID are required, in this case also the TSAPs need to be specified.

```
    CALL  PUT , "InstanceDbPut"
    REQ : = TRUE
    ID : = 65400
    DONE : = "DoneFlag"
    ERROR : = "ErrorFlag"
    STATUS : = "StatusWord"
```

```
     ADDR_1 : = P#DB10.DBX0.0 BYTE 20  // IP address or host name as character array in the form "192.168.1.22"
     ADDR_2 : = P#DB10.DBX20.0 BYTE 2  // Remote TSAP as byte array in the form b#16#1 b#16#1
     ADDR_3 : = P#DB10.DBX22.0 BYTE 2  // Local  TSAP as byte array in the form b#16#1 b#16#0
     ADDR_4 : = P#DB10.DBX24.0 WORD 1  // If another port than 102 is used, it can be specified as WORD
     SD_1 : = P#DB10.DBX26.0 WORD 1          // Number of the new, free local ID as WORD
     SD_2 : =
     SD_3 : =
     SD_4 : =
```

Download: **S7 for Windows® Sample: IBH Link UA SoftPLC PUT Special function S7-1200**

Download: **Step7 Sample: IBH Link UA SoftPLC PUT Special function S7-1200**

Instruction set

Bit logic instructions

Instruction	Operand	Description
A		AND with a scan to "1"“
AN		AND with a scan to "0"
O		OR with scan to "1"
ON		OR with scan to "0"
X		Exclusive-OR with a scan to "1"
XN		Exclusive-OR with a scan to "0"
	I	of an input
	Q	of an output
	M	of a memory
	L	of a local data bit
	T	of a timer
	C	of a counter
	DBX	of a data bit
	DIX	of an instant data bit
==0		result equal zero
<>0		result unequal zero
>0		result greater than zero
>=0		result greater or equal zero
<0		result less than zero
<=0		result less or equal zero
UO		invalid result
OV		overflow
OS		overflow (saving)
BR		binary result

Other boolean logic instructions

Operation	Operand	Description
A(		AND left parenthesis
AN(		AND NOT left parenthesis
O(		OR left parenthesis
ON(		OR NOT left parenthesis
X(		Exclusive OR left parenthesis
XN(		Exclusive OR NOT left parenthesis
)		Right parenthesis
O		OR-ing AND operations
NOT		Negate RLO
SET		Set RLO to "1"
CLR		Set RLO to "0"
SAVE		Save RLO to the BR bit

Operation	Operand	Description
=		Assign value of RLO
S		Set
R		Reset
FP		Positive edge detection
FN		Negative edge detection
	I	of an input bit
	Q	of an output bit
	M	of a memory bit
	L	of a local data bit
	DBX	of a data bit
	DIX	of a instance data bit

Master Control Relay

Operation	Operand	Description
MCRA		Activate MCR
MCRD		Deactivate MCR
MCR(		Open MCR zone
)MCR		Close MCR zone

Load / Transfer instructions

Operation	Operand	Description
L		Load operand
T		Transfer operand
	IB	Input byte
	IW	Input word
	ID	Input double word
	QB	Output byte
	QW	Output word
	QD	Output double word
	MB	Memory byte
	MW	Memory word
	MD	Memory double word
	LB	Local data byte
	LW	Local data word
	LD	Local data double word
	DBB	Data block byte
	DBW	Data block word
	DBD	Data block double word
	DIB	Instance data block byte
	DIW	Instance data block word
	DID	Instance data block double word
	STW	Status word
L	const	Load constant value
L	#Pointer	Load pointer
L	T	Timer
LC	T	Timer (BCD coded)
L	Z	Counter
LC	Z	Counter (BCD coded)
L	DBNO	Data block number
L	DBLG	Data block length
L	DINO	Instance data block number
L	DILG	Instance data block length

Peripheral access instructions

Peripheral access is not possible with the integrated SoftPLC.

Accumulator instructions

Operation	Operand	Description
PUSH		Push accumulators upwards
POP		Push accumulators downwards
ENT		Push accumulators (without A1) A2->A3, A3->A4
LEAVE		Push accumulators (without A1) A4->A3, A3->A2
TAK		Change ACCU1 with ACCU2
CAW		Change ACCU1 bytes 0 and 1
CAD		Change ACCU1 bytes 0 and 3, 1 and 2

Shift and rotating instructions

Operation	Operand	Description
SLW		Shift ACCU1-L left (word shift)
SLD		Shift ACCU1 left (double word shift)
SRW		Shift ACCU1-L right (word shift)
SRD		Shift ACCU1 right (double word shift)
SSI		Shift ACCU1-L right (word shift with sign)
SSD		Shift ACCU1 right (double word shift with sign)
RLD		Rotate ACCU1 left (double word)
RLDA		Rotate ACCU1 left through CC1
RRDA		Rotate ACCU1 right through CC1
RRD		Rotate ACCU1 right (double word)
	const	Number of places to shift/rotate

	-	Number of places to shift/rotate located in ACCU2

Timer instructions

Operation	Operand	Description
SP	T	Start as impulse
SE	T	Start as extended impulse
SD	T	Start as ON-delay
SS	T	Start as saving ON-delay
SF	T	Start as OFF-delay
R	T	Reset timer
FR	T	Enable timer

Counter instructions

Operation	Operand	Description
CU	C	Count up (increment)
CD	C	Count down (decrement)
S	C	Set counter
R	C	Reset counter
FR	C	Enable counter

Word instructions

Operation	Operand	Description
AW		AND ACCU1-L (AND word)
AD		AND ACCU1 (AND double word)
OW		OR ACCU1-L (OR word)
OD		OR ACCU1 (OR double word)
XOW		Exclusive-OR ACCU (XOR word)
XOD		Exclusive-OR ACCU (XOR double word)
	const	with a word or double word constant
	-	with ACCU2

Arithmetic instructions

Operation	Operand	Description
==I		Integer values equal
<>I		Integer values unequal
>I		Integer values greater
>=I		Integer values greater or equal
<I		Integer values less
<=I		Integer values less or equal
==D		Double integer values equal
<>D		Double integer values unequal
>D		Double integer values greater
>=D		Double integer values greater or equal
<D		Double integer values less
<=D		Double integer values less or equal
==R		Real values equal
<>R		Real values unequal
>R		Real values greater
>=R		Real values greater or equal
<R		Real values less
<=R		Real values less or equal
SIN		Sine of a real value
COS		Cosine of a real value
TAN		Tangent of a real value
ASIN		Arcsine of a real value
ACOS		Arccosine of a real value
ATAN		Arctangent of a real value
SQR		Square a real value
SQRT		Square root a real value
EXP		e to the power of a real value
LN		Natural logarithm of a real value
+I		Integer addition
-I		Integer subtraction
*I		Integer multiplication
/I		Integer division
+D		Double-integer addition
-D		Double-integer subtraction

*D		Double-integer multiplication
/D		Double-integer division
+R		Real addition
-R		Real subtraction
*R		Real multiplication
/R		Real division
MOD		Double integer division (modulo)
+	const	Add a constant
+P#	const	Add a pointer
DEC		decrement ACCU1-LL (one byte)
INC		increment ACCU1-LL (one byte)

Data type conversion instructions

Operation	Operand	Description
ITD		Convert integer to double integer
ITB		Convert integer to BCD
DTB		Convert double integer to BCD
DTR		Convert double integer to real
BTI		Convert BCD to integer
BTD		Convert BCD to double integer
RND		Convert real to double integer (round)
RND+		Convert real to double integer (round up)
RND-		Convert real to double integer (round down)
TRUNC		Convert real to double integer (truncate)
INVI		Invert ACCU1-L (integer)
INVD		Invert ACCU1 (double integer)
NEGI		Negate ACCU1-L (integer)
NEGD		Negate ACCU1 (double integer)
NEGR		Negate ACCU1 (real)
ABS		Absolute value (real)

Jump instructions

Operation	Operand	Description
JU	goal	Jump unconditional
JC	goal	Jump if RLO = 1
JCB	goal	Jump if RLO = 1 (save RLO)
JCN	goal	Jump if RLO = 0
JNB	goal	Jump if RLO = 0 (save RLO)
JBI	goal	Jump if BR = 1
JNBI	goal	Jump if BR = 0
JZ	goal	Jump if compare result = 0
JN	goal	Jump if compare result != 0
JP	goal	Jump if compare result > 0
JPZ	goal	Jump if compare result >= 0
JM	goal	Jump if compare result < 0
JMZ	goal	Jump if compare result <= 0
JUO	goal	Jump if compare result "Unordered Math Instruction"
JO	goal	Jump on overflow
JOS	goal	Jump on saving overflow
JL	goal	Jump distributor
LOOP	goal	Loop programming (decrement ACCU1-L and jump if != 0

Block call instructions

Operation	Operand	Description
CALL FB		Unconditional call of an FB with parameter transfer
CALL FC		Unconditional call of an FC with parameter transfer
CALL SFB		Unconditional call of an SFB with parameter transfer
CALL SFC		Unconditional call of an SFC with parameter transfer
UC FB		Unconditional call of an FB without parameter transfer
CC FB		Conditional call of an FB without parameter transfer
UC FC		Unconditional call of an FC without parameter transfer
CC FC		Conditional call of an FC without parameter transfer
BEU		End block unconditionally
BEC		End block conditionally (RLO = 1)
BE		End block
OPN	DB	Open data block
	DI	Open instance data block
TDB		Exchange data block registers

Indirect addressing instructions

Operation	Operand	Description
LAR1/LAR2		Load AR1/AR2
	MD	with memory double word
	LD	with local data double word
	DBD	with data block double word
	DID	with instance data block double word
LAR1	-	Load AR1 with ACCU1
LAR2	-	Load AR2 with ACCU1
LAR1	AR2	Load AR1 with AR2
LAR1	P#	Load AR1 with pointer
LAR2	P#	Load AR2 with pointer
TAR1/TAR2		Transfer in AR1/AR2
	MD	in memory double word
	LD	in local data double word
	DBD	in data block double word
	DID	in instance data block double word
TAR1	-	Transfer AR1 in ACCU1
TAR2	-	Transfer AR2 in ACCU1
TAR1	AR2	Transfer AR1 in AR2
TAR		Swap AR1 with AR2
+AR1		Add ACCU1 to AR1
+AR2		Add ACCU1 to AR2
+AR1	P#	Add pointer to AR1
+AR2	P#	Add pointer to AR2

Program display and null operation instructions

Operation	Operand	Description
NOP	0	Null operation instruction
NOP	1	Null operation instruction
BLD	const	Program display instruction (null operation)

Integrated organisation blocks

The organisation blocks, listed in the chart below, are executed, if they are programmed:

OB no	Short Description
OB 1	Main program
OB 10 – OB 17	Time alarm
OB 20 – OB 23	Delay alarm
OB 30 – OB 38	Time blocks (cyclic call)
OB 80 – OB 87	Error blocks
OB 90	Background execution
OB 100/OB 101	Warm or hot restart
OB 121/OB 122	Programming faults

Integrated system functions

All system functions (SFC) relevant for a Software PLC are integrated in *S7-SoftPLC*.

The chart below shows the available SFCs(in numeric listing)

SFC no	SF name	Short Description
SFC 0	SET_CLK	Set system clock
SFC 1	READ_CLK	Read system clock
SFC 6	RD_SINFO	Read start information of actual OB
SFC 20	BLKMOV	Copy PLC-variable in target-area
SFC 21	FILL	Preoccupy PLC-variable in target-area
SFC 22	CREAT_DB	Create data block
SFC 23	DEL_DB	Delete data block
SFC 24	TEST_DB	Test attributes of a data block
SFC 25	COMPRESS	Compress load memory
SFC 28	SET_TINT	Set time alarm
SFC 29	CAN_TINT	Stop time alarm (do not execute)
SFC 30	ACT_TINT	Enable time alarm
SFC 31	QRY_TINT	Status of time alarm
SFC 32	SRT_DINT	Start delay alarm
SFC 33	CAN_DINT	Stop delay alarm
SFC 34	QRY_DINT	Status query of delay alarm

SFC 36	MSK_FLT	Mask a synchron error occurrence
SFC 37	DMSK_FLT	Demask a synchron error occurrence
SFC 38	READ_ERR	Read out the occurrence status register
SFC 39	DIS_IRT	Locking the asynchron and alarm error processing
SFC 40	EN_IRT	Enable the asynchron and alarm error processing
SFC 41	DIS_AIRT	Delay high-priority asynchron and alarm error processing
SFC 42	EN_AIRT	Enable high-priority asynchron and alarm error processing
SFC 43	RE_TRIGR	Retrigger the Watchdog
SFC 46	STP	CPU operating state STOP
SFC 51	RDSYST	Read out system status list (SSL)
SFC 64	TIME_TCK	Timer Tick
SFC 81	UBLKMOV	Block Move, that can not be stopped

Integrated system blocks

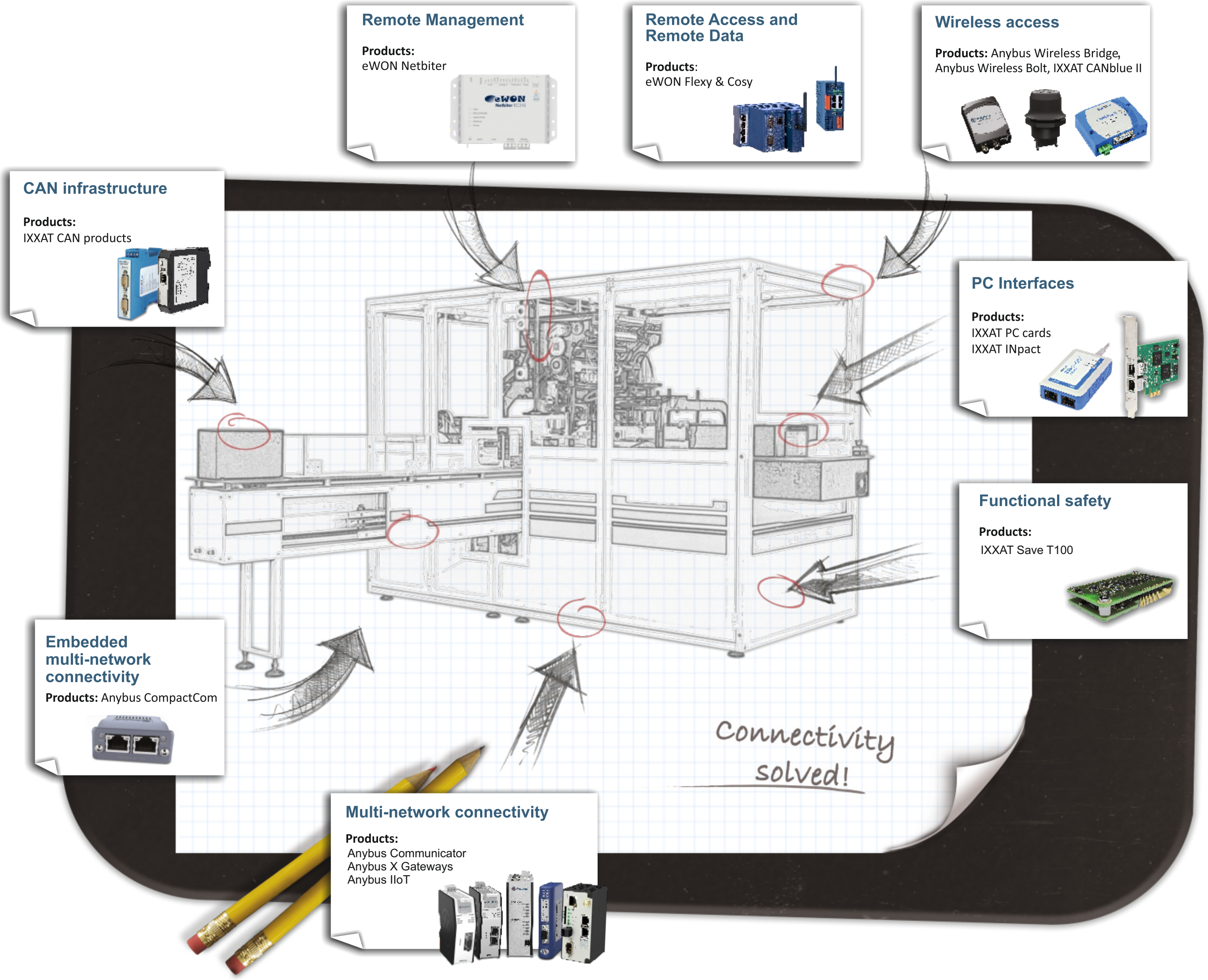
The system blocks (SFB) relevant for a software PLC are integrated in *S7-SoftPLC*.

The following system function blocks are supported:

SFB no	SFB name	Short Description
SFB 0	CTU	Count up
SFB 1	CTD	Count down
SFB 2	CTUD	Count up and down
SFB 3	TP	Generate pulse
SFB 4	TON	Generate ON-delay
SFB 5	TOF	Generate OFF-delay
SFB 8	USEND	OPC UA Client Configuration
SFB 32	DRUM	Implement sequencer

Technical Data

Load memory	5 MB
Main memory	> 100 MB
Blocks	16.384
Flags (Bit)	131.072
Timer	2048
Counter	2048
Digital I/O	32.768 Bit
	processing times 1)
CPU Chare PLC 50%	apx. 360 µs
CPU Chare PLC 33%	apx. 550 µs
CPU Chare PLC 25%	apx. 720 µs
CPU Chare SPS 20%	apx. 900 µs
CPU Chare SPS 12%	apx. 1800 µs



Remote Management

Products:
eWON Netbiter

Remote Access and Remote Data

Products:
eWON Flexy & Cosy

Wireless access

Products: Anybus Wireless Bridge, Anybus Wireless Bolt, IXXAT CANblue II

CAN infrastructure

Products:
IXXAT CAN products

PC Interfaces

Products:
IXXAT PC cards
IXXAT INpact

Functional safety

Products:
IXXAT Save T100

Embedded multi-network connectivity

Products: Anybus CompactCom

Multi-network connectivity

Products:
Anybus Communicator
Anybus X Gateways
Anybus IIoT

¿Qué aportamos?

Productos tecnológicos de la más alta calidad aplicados en la industria. Un equipo técnico experto en dar soluciones completas con garantía de funcionamiento.

- Más de 30 años de experiencia en el diseño e integración de sistemas de control y comunicación industrial.
- Expertise en redes industriales y buses de campo.
- Telecomunicaciones, Internet, Ethernet TCP/IP.
- El Internet de las cosas (IoT), y el Internet Industrial de las cosas (IIoT).

er-soft
Industrial Control &
Communication Systems

Todo en Redes Industriales, Buses de Campo y Control de Procesos.

ER-Soft, S.A.
C/ Comunidad de la Rioja 3, 28231 Las Rozas de Madrid
E-mail: info@er-soft.com
Tel.: +34 916 408 408
www.er-soft.com

2018/19