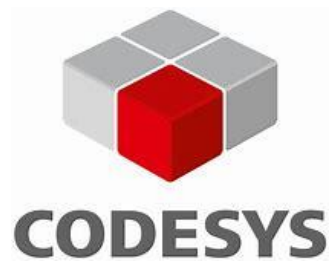


Description of

# CODESYS Integration

for

## IS1+ field stations



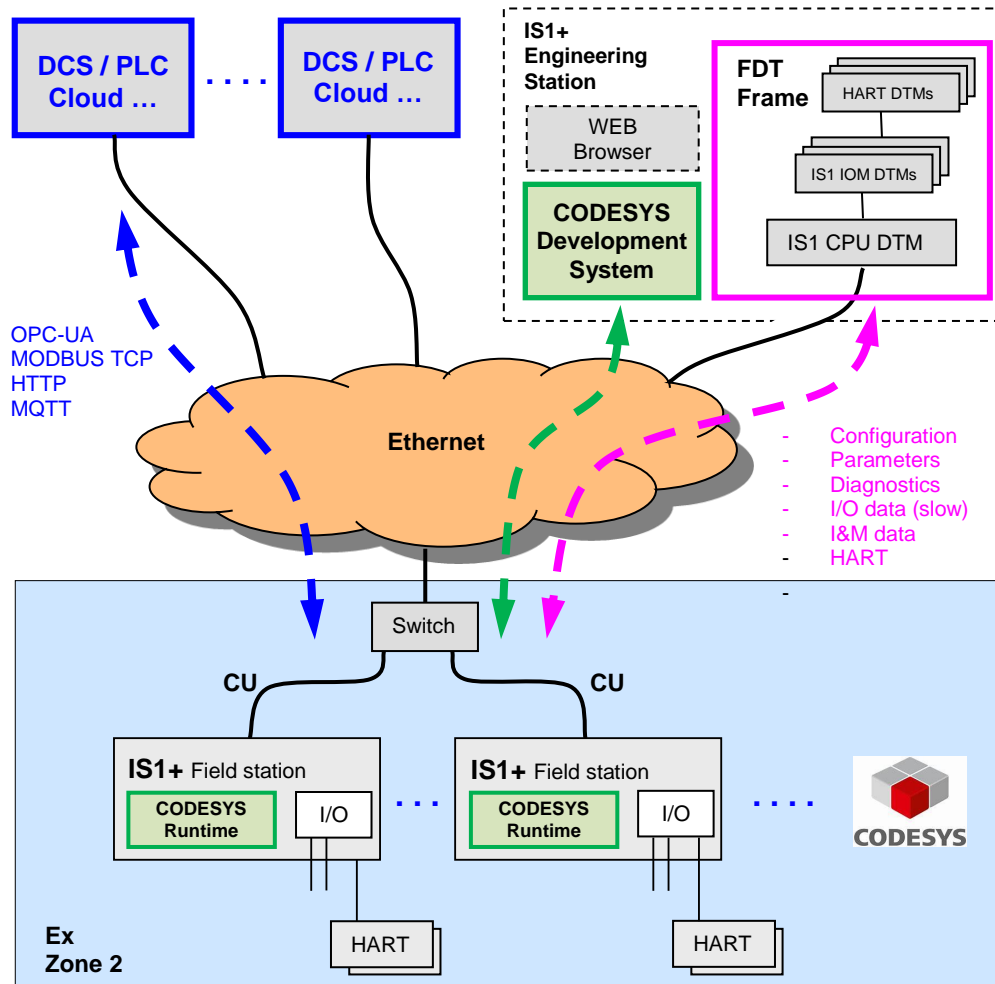
## CODESYS Integration

### Content:

1	System overview .....	3
2	System requirements.....	4
2.1	Demo Modus.....	4
3	IS1+ Configuration.....	5
3.1	IP Address of IS1+ CPU.....	5
3.2	LED display IS1+ CPU .....	5
3.3	WEB Server.....	6
4	CODESYS System .....	7
4.1	Installation .....	7
4.2	Project engineering .....	8
4.2.1	Connect IS1+ Field station.....	8
4.2.2	Manual I/O-Module configuration.....	9
4.2.3	Automatic I/O-Module configuration .....	11
4.2.4	Symbolic Variable Names.....	11
4.2.5	Counter-/Frequency channel allocation to CODESYS DUM Data structure .....	12
4.2.6	Login .....	13
4.2.7	Online Config Mode .....	13
4.2.8	HART Variables (HV).....	13
4.2.9	Start.....	14
4.2.10	Status and Diagnostic.....	14
4.2.11	MODBUS TCP Master / Slave Device.....	15
4.2.11.1	add Ethernet port to IS1+ CPU .....	16
4.2.11.2	add Modbus TCP Master or Slave Device .....	16
4.2.11.3	Modbus TCP Master (Client):.....	17
4.2.11.4	add MODBUS TCP slaves to Master .....	18
4.2.11.5	Modbus TCP Slave Device (Server): .....	20
4.2.12	OPC-UA Server .....	21
4.3	'Hello World' in IEC 61131-3 .....	23
4.4	System Analysis .....	24
4.4.1	Cycle time .....	24
4.4.2	Memory Utilization .....	25
4.5	Update.....	26
5	Security.....	27
6	Liability.....	27
7	List of abbreviations.....	27
8	Version changes.....	28
9	Literature references .....	28
10	Support Address.....	28

## CODESYS Integration

### 1 System overview



As an off-the-shelf explosion protected unit, the IS1+ field station can be installed directly in the potentially explosive atmosphere (Zone 1 or Zone 2). It can also be installed in the safe area. The picture above shows a Zone 2 solution with CODESYS.

The IS1+ 9442 CPU can optionally be extended by an integrated CODESYS Runtime System. CODESYS is an IEC 61131-3 automation software for project engineering of control systems. This allows automation tasks to be processed locally in IS1+ CPUs. In this case, the RIO typical connection of IS1+ to higher-level PLCs via AS-Bus protocols (PROFIBUS, PROFINET, EtherNet/IP, MODBUS TCP) is deactivated. Higher-level systems can, for example, be connected to the CODESYS Runtime System in the IS1+ CPU via OPC-UA.

This document describes the project engineering of IS1+ field stations in the CODESYS Development System as well as their functions in the CODESYS Runtime System within an IS1+ CPU.

Knowledge of the general handling of CODESYS is recommended.

[Literature references](#)

[Support Address](#)

[List of abbreviations](#)

## CODESYS Integration

## 2 System requirements

### Hardware requirements:

- IS1+ Field station with CPU 9442/35-10-00, Socket 9496/.. and Power Module PM 9445/.. 9442/xxxx with CODESYS Licency.

### Software requirements:

- 9442 CPU Firmware V2.xx.yy-zzzz

The above mentioned CPU firmware contains a CODESYS Runtime System V3.5.15.0 A CODESYS Development System Version 3.5.15 or higher must be used.

This IS1+ CPU variant is equipped with a CODESYS license sticker with a unique CODESYS license number.

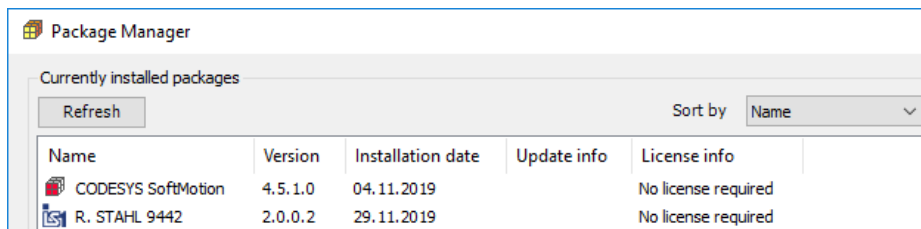
IS1+ CPU Firmware Revision	Usage
V1.xx.yy-zzzz	IS1+ Remote I/O
V2.xx.yy-zzzz	IS1+ CODESYS PLC

Operation with CODESYS License only with IS1+ CPU Type 9442 /xxxx .....

- Driver Package for integration of IS1+ into the CODESYS Development System  
File: R. STAHL IS1+ CODESYS V2.xx.yy.zzzz.package

For compatibility reasons, the version numbers of IS1+ CPU Firmware and IS1+ CODESYS Package must match except for the rear part (.zzzz)

The currently installed IS1+ Package version can be displayed or uninstalled in Tools -> Package Manager.



### 2.1 Demo Modus

The above mentioned IS1+ 9442 CPU firmware variant V2.xx.yy-zzzz is optionally available and can be loaded into all IS1+ 9442 CPUs for demo and evaluation purposes even without previous CODESYS extension.

In this case the CODESYS Runtime System works in demo mode with a limited runtime of 2 hours with full functionality. Then the runtime system is switched off and a powercycle is required for restarting.

WebVisu works in demo mode for about 30 minutes.

## CODESYS Integration

### 3 IS1+ Configuration

The protocol selection switch 'S1: Prot' in the socket of the corresponding IS1+ field station is to be set to Position 6 = MODBUS TCP.

Depending on the I/O modules to be used, a suitable configuration and parameterization of the IS1+ field station must be created using the tools IS1 DTM or I.S.Wizard and loaded into the 9442 CPU. A 9442 CPU with MODBUS TCP must be selected in the tool during configuration.

Description of the configuration of I/O module operating modes, parameters, scaling, etc. see document: **Operating instructions MODBUS TCP for IS1+ field stations.**

Note: The AS-Bus interface with MODBUS TCP is deactivated in combination with CODESYS. The CPU selection with MODBUS TCP is for internal configuration only. Operation of the AS-Bus protocols in parallel with CODESYS, is not possible. Access via the AS-Bus protocols is disabled. In this case, output data can only be written via the user program in the CODESYS Runtime System.

From the list of I/O modules configured in the 9442 CPU, the data structure of all I/O signals in the CODESYS project of the Development System can be generated automatically.

See [Automatic I/O-Module configuration](#)

#### 3.1 IP Address of IS1+ CPU

Two separate IP addresses for the Ethernet communication are available for the 9442 IS1 CPU:

- **IP-AS:** Realtime bus to automation system (MODBUS TCP, PROFINET, EtherNet/IP,) When using the CODESYS system, this interface is **not used** and settings have no function.
- **IP-SB:** Service Bus Functions: Web-Server, IS1 DTM, HART, Standard TCP Traffic, SW-Update. This IP address must be used to connect the CODESYS Development System.

The IP-SB address can be set using the tool IS1+ Detect or via the IS1+ Web Diagnostics.

#### 3.2 LED display IS1+ CPU

CODESYS operation without AS protocol:

AS EXCH LED	On: PLC RUN	
	Off: PLC Stop	
STATUS LED	Function tbd	(Spare LED_Res -> first CPUs covered by foil).

Note: IS1+ CPU Firmware Download via IS1+ Web Page is blocked at PLC RUN (AS EXCH = On)

In addition, the description of LED displays of the 9442 CPU according to the operating manual applies.

## CODESYS Integration

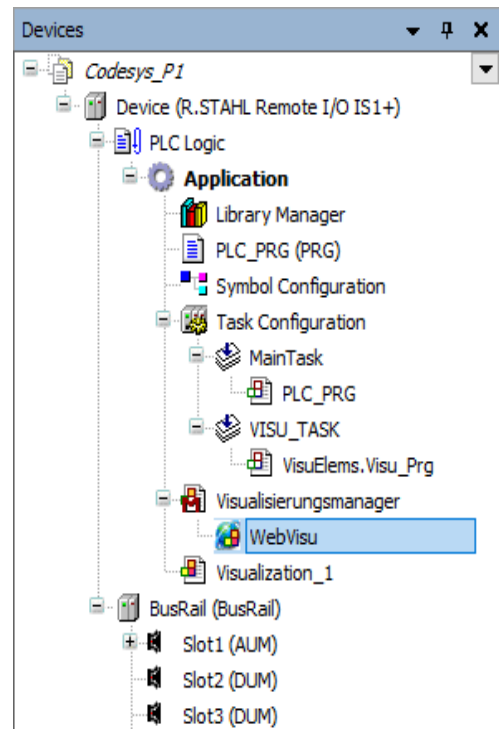
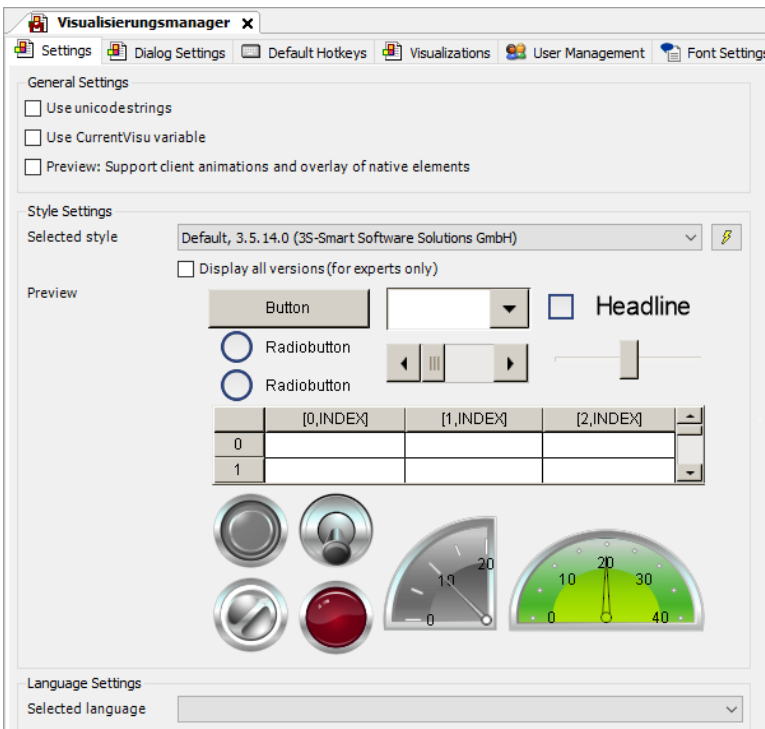
### 3.3 WEB Server

Two separate WEB servers are integrated in an IS1+ CPU with CODESYS:

WEB Server	Access	Function
IS1+ Web Diagnostic	IP-SB in Webbrowser	Diagnosis of IS1+ I/O modules and CPU hardware Change the IP address of the IS1+ CPU. Firmware Download IS1+ CPU.
CODESYS Web Visualisation	IP-SB:8080 <a href="http://localhost:8080/webvisuA.htm">http://localhost:8080/webvisuA.htm</a> (webvisuA = Projected name)	User specific web visualization which can be projected using the CODESYS Development System.

CODESYS WebVisu allows the user to create his own websites in the development system which can be downloaded into the IS1+ Runtime System.

Symbolic variables can be linked directly to finished graphic objects from a library with little effort.



WebVisu can be optionally configured in the CODESYS project

-> Application -> Add Objekt -> Visualisation

Both web servers can be operated in parallel.

Details CODESYS WebVisu: <https://www.codesys.com/products/codesys-visualization/webvisu.html>

WebVisu runs in demo mode without CODESYS license and is switched off 30 minutes after powercycle.

## CODESYS Integration

### 4 CODESYS System

A CODESYS Runtime System V 3.5 can optionally be integrated in IS1+ 9442 CPUs.

For PLC project engineering according to IEC 61131, a CODESYS Development System V3.5 or higher must be used. Various automation languages are supported:

- IL (Instruction List),
- ST (Structured Text)
- LD (Ladder Diagram),
- FBD (Function Block Diagram),
- SFC (Sequential Function Chart),
- CFC (Continuous Function Chart)

The Development System can be downloaded free of charge from the CODESYS Store:

<https://store.codesys.com/codesys.html>

Project engineering created with the CODESYS Development System can be loaded into Runtime Systems in IS1+ CPUs and executed there.

Supported CODESYS options:

- CODESYS WebVisu
- CODESYS OPC UA Server
- Modbus TCP Server
- Modbus TCP Client

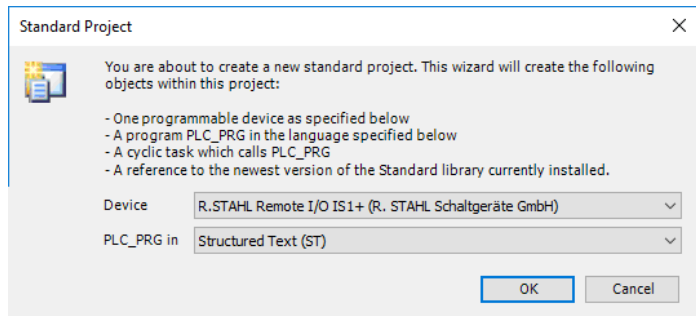
#### 4.1 Installation

- Download CODESYS Development System and install it on your PC.
- Install R. STAHL CODESYS Package on your PC.  
Start package installation by double-clicking on File: R. STAHL IS1+ CODESYS Bx.y.z.package  
- Select 'Typical Installation'.
- After completion of both installations start CODESYS Development System on PC

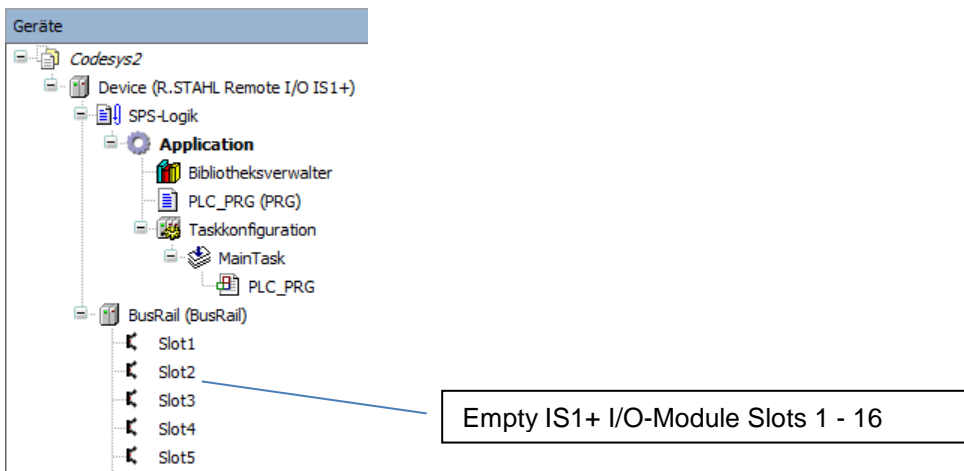
## CODESYS Integration

### 4.2 Project engineering

- Create a project in Codesys Development System.
- Select IS1+ RIO as device and desired programming language



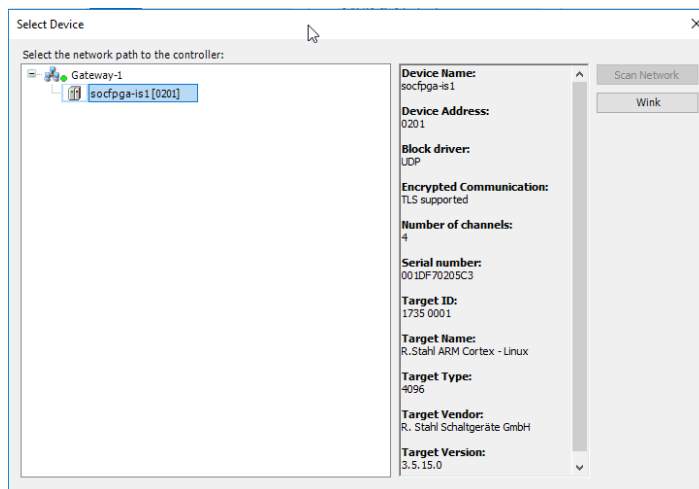
An IS1+ field station is created in the project tree. The BusRail with 16 empty slots for I/O modules is located parallel to the Device.



#### 4.2.1 Connect IS1+ Field station

Setting of the IS1+ IP-SB address in the CODESYS project:

- Function 'Edit object' (Right Mouse on Device) opens the Communication window see below.
- Select 'Scan Network'



- > Select IS1+ CPU from list of found devices.
- Serial number = MAC Address of the IS1+ CPU
  - 'Wink' activates LED displays on the IS1+ CPU for 5 seconds



## CODESYS Integration

- Alternatively: Enter IP-SB address and connect to IS1+ field station.

Green = IS1+ Field station is connected.

Optional: IP-SB Address of the IS1+ CPU

### 4.2.2 Manual I/O-Module configuration

**Plug device** (right mouse click on Busrail in device tree)

Manually insert I/O modules into the project tree according to the IS1+ configuration.

All different I/O module types of the IS1+ system are mapped to two universal data structures:

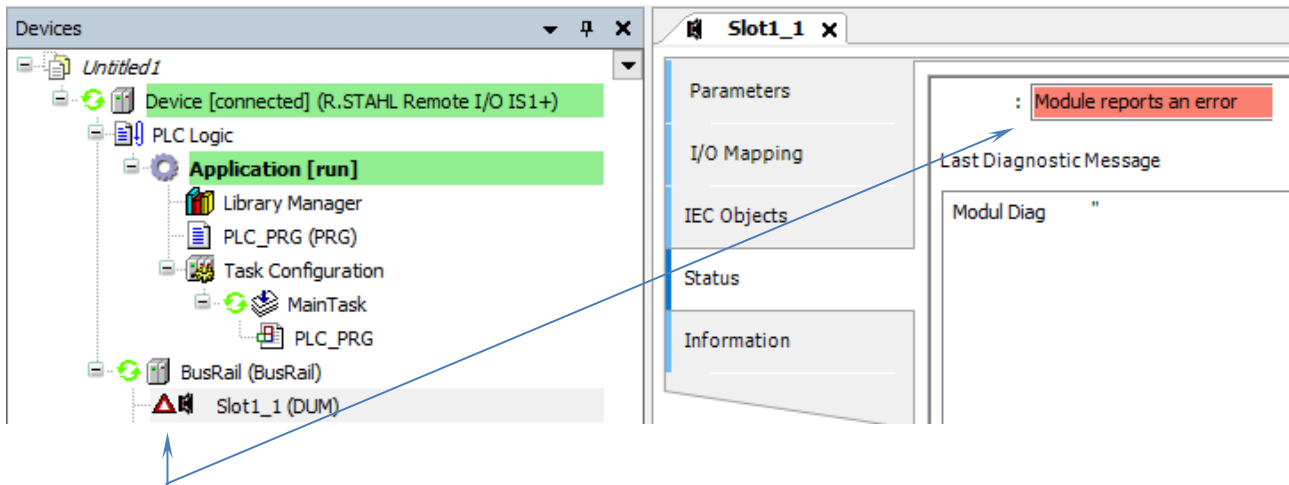
- AUM Analog Universal Module
- DUM Digital Universal Module

**Attention:** Windows remain open and further IOMs can be inserted by selecting another IOM slot in the tree. Close the window manually after completing the IOM configuration.

Depending on the real I/O module type used and the configured operating mode, parts of the generated data structure may remain unused.

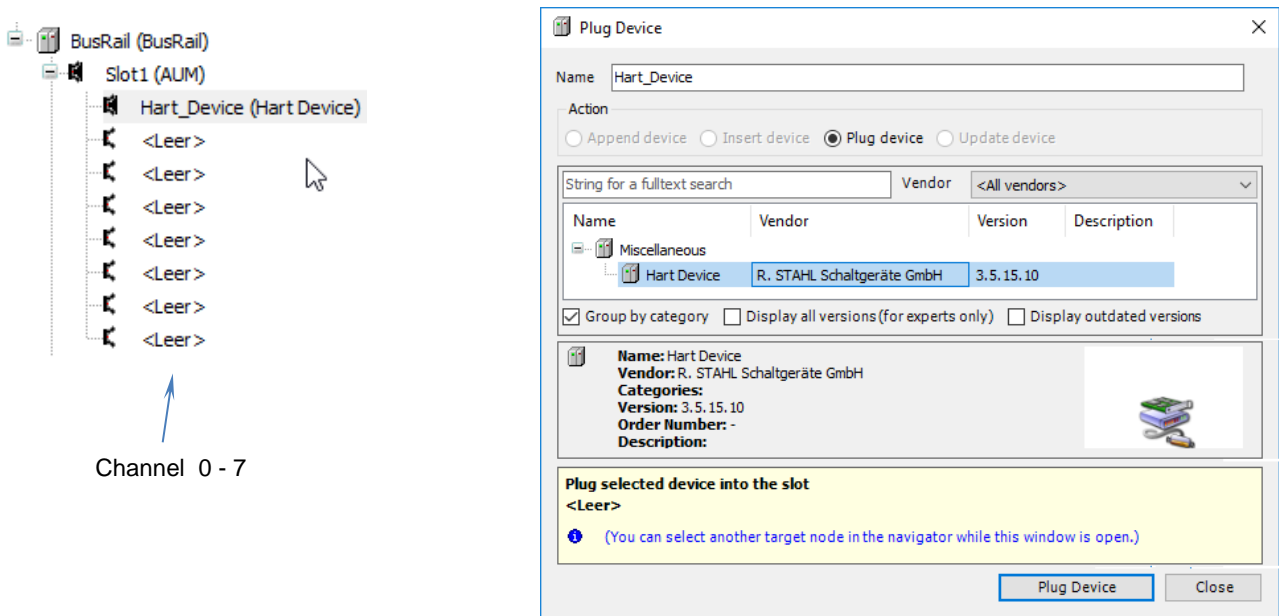
Example: If a DI module without counter/frequency function has been configured, the corresponding signals of the data structure remain unused.

## CODESYS Integration



An error is reported in online mode if the I/O module configuration and data structure (AUM, DUM) are incorrectly assigned.

Optionally plug **HART Devices** into channels of an IS1+ HART module:



The device name (default = HART\_Device) can be configured.

A data structure is created with the four possible HART variables (PV, SV, TV, QV) of a HART field device including Value, Unit and DV status as well as EXTDEVSTATUS of the device.

The update of these HART variables is independent of the number of HART variables (+4HV, +8HV) configured via the operating mode in the IS1+ HART module.

## CODESYS Integration

### 4.2.3 Automatic I/O-Module configuration

**Scan for devices** (right mouse click on BusRail in the device tree).

All I/O modules configured in the IS1+ field station are automatically transferred to the device tree in the CODESYS project regardless of the IOM type plugged in.

HART devices which are connected to IS1+ IOMH and found via the HART Livelist are also automatically integrated in the project tree.

When the I/O modules are created in the device tree, an I/O image with the associated signals and data structures of the module is automatically generated. Variable names are empty by default.

**Attention!** Existing module names in the device tree are overwritten with default values and must be changed manually afterwards if necessary!

### 4.2.4 Symbolic Variable Names

Symbolic variable Names for use in the PLC program as well as unit and description are empty by default and can be configured project-specifically for each signal.

This data can also be processed externally using the functions:

- Export mappings to CSV
- Import mappings from CSV

**Attention on DO signals.**

Here, only either the integer variable or the individual bits can be assigned to a symbolic name so that the write access is unique. In the event of an error, the symbols of the bits are discarded during import.

Variable	Mapping	Channel	Address	Type	Unit	Description
DO_S2_0	Bit0	%QX16.0	BOOL			
DO_S2_1	Bit1	%QX16.1	BOOL			
DO_S2_2	Bit2	%QX16.2	BOOL			
DO_S2_3	Bit3	%QX16.3	BOOL			
DO_S2_4	Bit4	%QX16.4	BOOL			
DO_S2_5	Bit5	%QX16.5	BOOL			
DO_S2_6	Bit6	%QX16.6	BOOL			
DO_S2_7	Bit7	%QX16.7	BOOL			

Variable	Mapping	Channel	Address	Type	Unit	Description
Stat_S1	Status	%IB0	BYTE			
Stat_S1_0	Bit0	%IX0.0	BOOL			
Stat_S1_1	Bit1	%IX0.1	BOOL			
Stat_S1_2	Bit2	%IX0.2	BOOL			
Stat_S1_3	Bit3	%IX0.3	BOOL			
Stat_S1_4	Bit4	%IX0.4	BOOL			
Stat_S1_5	Bit5	%IX0.5	BOOL			
Stat_S1_6	Bit6	%IX0.6	BOOL			
Stat_S1_7	Bit7	%IX0.7	BOOL			
AI_S1_0	AI0	%IW1	INT			Pressue P1
AI_S1_1	AI1	%IW2	INT			
AI_S1_2	AI2	%IW3	INT			
AI_S1_3	AI3	%IW4	INT			
AI_S1_4	AI4	%IW5	INT			
AI_S1_5	AI5	%IW6	INT			
AI_S1_6	AI6	%IW7	INT			
AI_S1_7	AI7	%IW8	INT			
AO_S1_0	AO0	%QW0	INT			
AO_S1_1	AO1	%QW1	INT			
AO_S1_2	AO2	%QW2	INT			
AO_S1_3	AO3	%QW3	INT			
AO_S1_4	AO4	%QW4	INT			
AO_S1_5	AO5	%QW5	INT			
AO_S1_6	AO6	%QW6	INT			
AO_S1_7	AO7	%QW7	INT			

## CODESYS Integration

### 4.2.5 Counter-/Frequency channel allocation to CODESYS DUM Data structure

DI_S4_C8	CF8	%IW31	UDINT
DI_S4_C9	CF9	%IW32	UDINT
DI_S4_C10	CF10	%IW33	UDINT
DI_S4_C11	CF11	%IW34	UDINT
DI_S4_C12	CF12	%IW35	UDINT
DI_S4_C13	CF13	%IW36	UDINT
DI_S4_C14	CF14	%IW37	UDINT
DI_S4_C15	CF15	%IW38	UDINT
DO_S4	DO	%QW12	WORD
Bit0		%QX24.0	BOOL
Bit1		%QX24.1	BOOL
Bit2		%QX24.2	BOOL
Bit3		%QX24.3	BOOL
Bit4		%QX24.4	BOOL
Bit5		%QX24.5	BOOL
Bit6		%QX24.6	BOOL
Bit7		%QX24.7	BOOL
Bit8		%QX25.0	BOOL
Bit9		%QX25.1	BOOL
Bit10		%QX25.2	BOOL
Bit11		%QX25.3	BOOL
Bit12		%QX25.4	BOOL
Bit13		%QX25.5	BOOL
Bit14		%QX25.6	BOOL
Bit15		%QX25.7	BOOL
DI_S4_CR	CNTR Reset	%QB26	BYTE
Bit0		%QX26.0	BOOL
Bit1		%QX26.1	BOOL
Bit2		%QX26.2	BOOL
Bit3		%QX26.3	BOOL
Bit4		%QX26.4	BOOL
Bit5		%QX26.5	BOOL
Bit6		%QX26.6	BOOL
Bit7		%QX26.7	BOOL
DI_S4_CS	CNTR Stop	%QB27	BYTE
Bit0		%QX27.0	BOOL
Bit1		%QX27.1	BOOL
Bit2		%QX27.2	BOOL
Bit3		%QX27.3	BOOL
Bit4		%QX27.4	BOOL
Bit5		%QX27.5	BOOL
Bit6		%QX27.6	BOOL
Bit7		%QX27.7	BOOL

**32 bit up/down counter:**  
 If a channel pair is configured as 32 bit up/down counter, the counter value is mapped into the first of the two counter variables (CFx). The second of the two counter variables (CFx+1) is not updated and is always = 0.

**Allocation counter (CNTR) Reset / STOP:**

Bit	CNTR channel
0	8
1	9
2	10
3	11
4	12
5	13
6	14
7	15

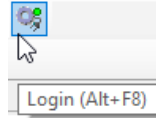
configured operation mode	Channels supporting CF function	Channels without CF function	CNTR Reset / STOP Bit allocation
DIM + 2 CF	14 + 15	0 – 13	6 - 7
DIM + 6 CF	10 - 15	0 – 9	2 - 7
DIM + 8 CF	8 - 15	0 – 7	0 - 7

Further details of the counter / frequency function see document: **Operation Manual MODBUS TCP for IS1+ Field stations.**

## CODESYS Integration

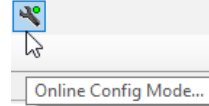
### 4.2.6 Login

This loads the I/O configuration and the PLC program into the runtime environment of the IS1+ field station.



### 4.2.7 Online Config Mode

Manual Loopcheck. **No** PLC program is executed. All input values are updated cyclically. Output values can be written manually.



Variable	Mapping	Channel	Address	Type	Current Value	Unit	Description
Stat_S1		Status	%IB0	BYTE	255		
AI_S1_0		AI0	%IW1	INT	12000		Pressue P1
AI_S1_1		AI1	%IW2	INT	5000		
AI_S1_2		AI2	%IW3	INT	0		
AI_S1_3		AI3	%IW4	INT	3428		
AI_S1_4		AI4	%IW5	INT	0		
AI_S1_5		AI5	%IW6	INT	0		
AI_S1_6		AI6	%IW7	INT	0		
AI_S1_7		AI7	%IW8	INT	0		
AO_S1_0		AO0	%QW0	INT	12000		
AO_S1_1		AO1	%QW1	INT	5000		
AO_S1_2		AO2	%QW2	INT	0		
AO_S1_3		AO3	%QW3	INT	3428		
AO_S1_4		AO4	%QW4	INT	0		
AO_S1_5		AO5	%QW5	INT	0		
AO_S1_6		AO6	%QW6	INT	0		
AO_S1_7		AO7	%QW7	INT	0		

### 4.2.8 HART Variables (HV)

Variable	Mapping	Channel	Address	Type	Current Value	Unit	Description
HV_S8_7_P		PV Struct	%ID86				
		Value	%ID86	REAL	1.237		
		DV Status	%IB348	BYTE	192		
		Unit	%IB349	BYTE	45		
		SV Struct	%ID88				
HV_S8_7_S		Value	%ID88	REAL	59		
		DV Status	%IB356	BYTE	192		
		Unit	%IB357	BYTE	41		
		TV Struct	%ID90				
HV_S8_7_T		Value	%ID90	REAL	NaN		
		DV Status	%IB364	BYTE	0		
		Unit	%IB365	BYTE	0		
		QV Struct	%ID92				
		Extended Device Status	%IB376	Enumeration of BYTE	Normal		

Extract from HCF / FCG Spec-183 Tab. 5.2:

Unit Code [dez]	Unit
7	bar
8	mbar
32	°C
36	mV
37	Ohm
39	mA
41	Liter
45	Meter
49	mMeter
58	V
59	pH
127	kW
128	kWh
163	kOhm

#### DV Status:

DV Status = good (0xc0 = 192) if HART Variable is readable.  
 DV Status = bad (0x00) if HART Variable is not readable.

#### EXTDEVSTATUS:

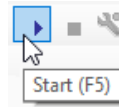
Contains NAMUR NE107 compatible condensed status information. Fully readable only by devices with HART 7 or higher. From HART 6 only bits 0 and 1 are included. Updated status info is delivered if the HART Field Device (HFD) is available and EXTDEVSTATUS is supported. If the HFD is not reachable or EXTDEVSTATUS is not supported by the HFD, the IOMH will set:

- EXTDEVSTATUS = Good / Normal (0x00) if HV variable readable.
- EXTDEVSTATUS = Failure (0x08) if HV variable not readable.

## CODESYS Integration

### 4.2.9 Start

The cyclic PLC operation is started.  
Outputs are controlled via the PLC program.



### 4.2.10 Status and Diagnostic

In online mode, module errors are displayed in the device tree and textually in the device parameters.

The screenshots illustrate the diagnostic interface. The first shows the 'Parameters' window for Slot 2, where the 'Modul Diag' parameter has the value 'IOM does not respond'. The second shows Slot 12 with 'Maintenance request' in the 'Modul Diag' parameter. The third shows the 'Last Diagnostic Message' window for Slot 12, displaying 'Modul Diag: Maintenance request'.

Signal errors are indicated by the signal status and can be evaluated by the PLC program, displayed in the WebVisu and reactions can be programmed.

Variable	Mapping	Channel	Address	Type	Current Value
Stat_S1		Status	%IB0	BYTE	253
Stat_S1_0		Bit0	%IX0.0	BOOL	TRUE
Stat_S1_1		Bit1	%IX0.1	BOOL	FALSE
Stat_S1_2		Bit2	%IX0.2	BOOL	TRUE
Stat_S1_3		Bit3	%IX0.3	BOOL	TRUE
Stat_S1_4		Bit4	%IX0.4	BOOL	TRUE
Stat_S1_5		Bit5	%IX0.5	BOOL	TRUE
Stat_S1_6		Bit6	%IX0.6	BOOL	TRUE
Stat_S1_7		Bit7	%IX0.7	BOOL	TRUE
AI_S1_0		AI0	%IW1	INT	9783
AI_S1_1		AI1	%IW2	INT	0

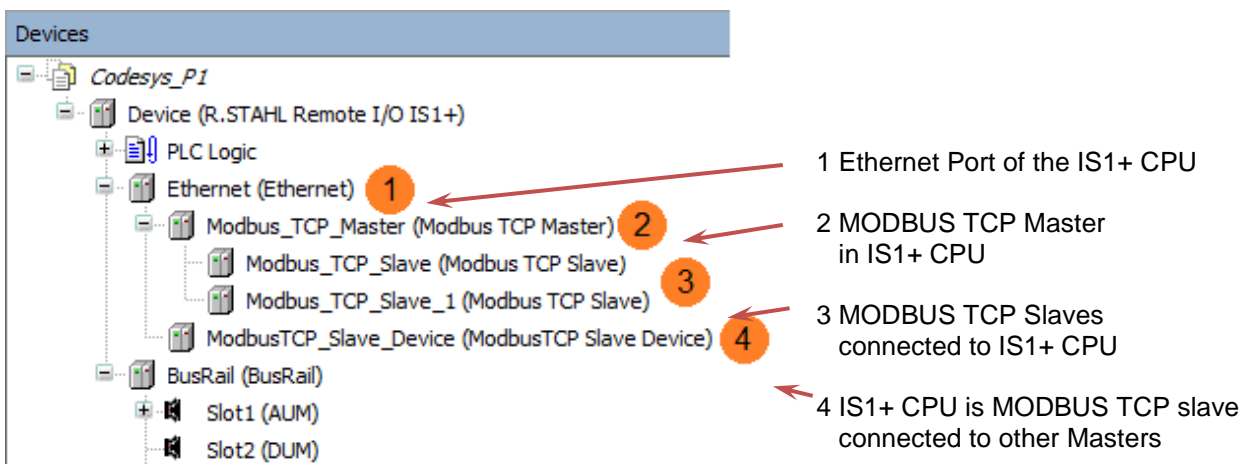
## CODESYS Integration

### 4.2.11 MODBUS TCP Master / Slave Device

A MODBUS TCP Master (Client) and a MODBUS TCP Slave Device (Server) are integrated in the IS1+ CODESYS CPU. The complete MODBUS configuration is done directly in the CODESYS Development System.

The MODBUS Configurator consists of editors for the following device categories, which are hierarchically suspended in the device tree:

1. **Ethernet:**  
The settings of the Ethernet adapter such as IP address, subnet mask etc. are configured here.
2. **Modbus TCP Master (Client):**  
A TCP Master mode can be inserted under the Ethernet node. Modbus-specific communication settings can be defined here, e.g. the "Response Timeout" for determining how long the response of a Modbus TCP slave is waited for.
3. **Modbus TCP Slave:**  
Several Modbus TCP Slave devices can be inserted under the Modbus TCP Master. Here the slave address and a series of Modbus commands (incl. associated I/O mapping) are defined, which are then processed by the driver and exchanged with this Modbus TCP Slave.
4. **Modbus TCP Slave Device:**  
IS1+ CPU works as MODBUS Slave Device. Data can be read and written from external MODBUS TCP Master devices.



The Modbus driver supports all Modbus function codes for the exchange of I/O data:

- FC 01 - Read Coils
- FC 02 - Read Discrete Inputs
- FC 03 - Read Holding Registers
- FC 04 - Read Input Registers
- FC 05 - Write Single Coil
- FC 06 - Write Single Register
- FC 15 - Write Multiple Coils
- FC 16 - Write Multiple Registers
- FC 23 - Read/ Write Multiple Registers

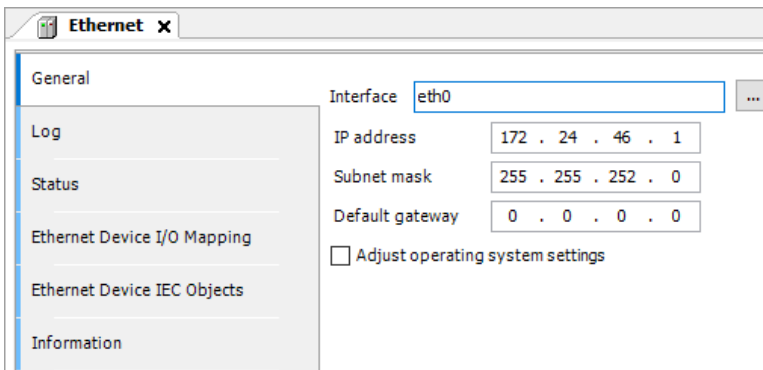
Maximum 32 Modbus TCP slaves are supported.

For details on the MODBUS TCP configuration see 'Help' in the CODESYS Development System.

## CODESYS Integration

### 4.2.11.1 add Ethernet port to IS1+ CPU

Device -> Add Device -> Ethernet

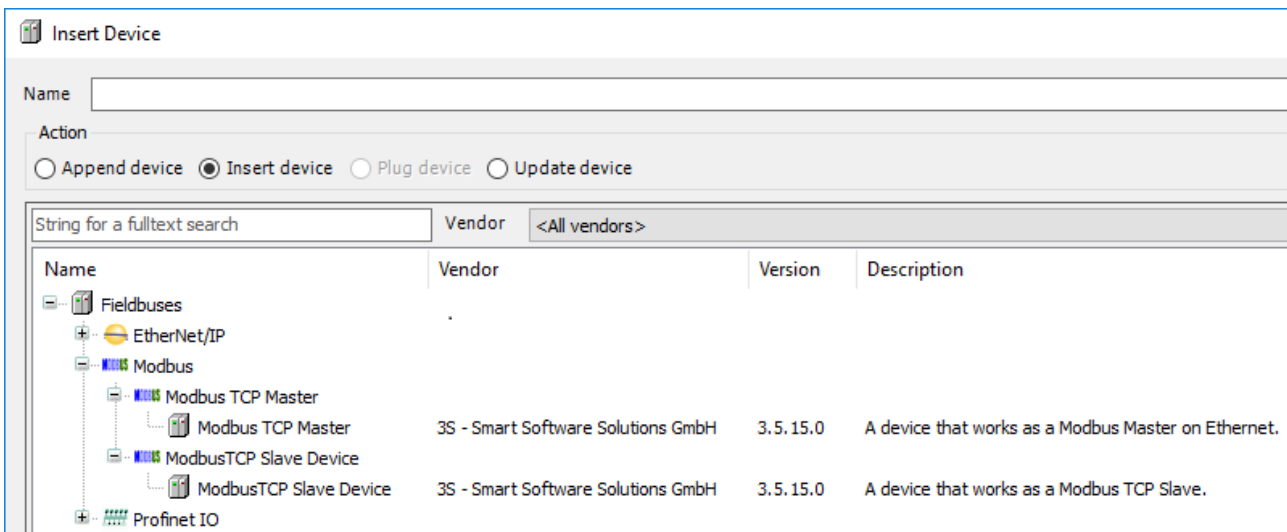


Select Interface = eth0 with existing connection to IS1+ CPU.

With this MODBUS TCP uses the set IP-SB address of the IS1+ CPU.

### 4.2.11.2 add Modbus TCP Master or Slave Device

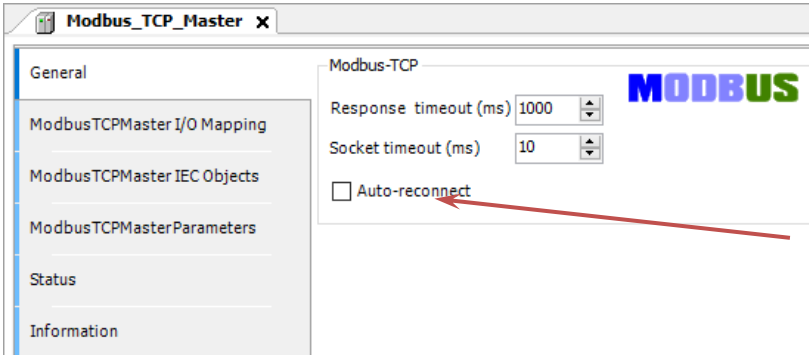
Ethernet -> Insert Device -> Modbus TCP Master / Slave Device



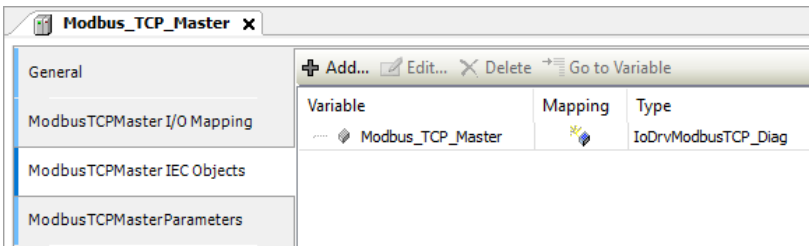


## CODESYS Integration

### 4.2.11.3 Modbus TCP Master (Client):



Select **Auto-reconnect** for automatic restart of cyclic connections after faults.



This screenshot shows the parameter table for the 'Modbus\_TCP\_Master' configuration. The table lists various parameters with their types, current values, default values, units, and descriptions.

Parameter	Type	Value	Default Value	Unit	Description
ExtendedChannelConfig	BOOL	true	true		Use the new Channel-Config format
OptimizationOn	BOOL	TRUE	TRUE		the driver optimizes the io update
Socket Timeout	UDINT	10	10		Socket Timeout in milliseconds
ResponseTimeOut	UDINT	1000	1000		Response time in milliseconds
AutoReconnect	BOOL	FALSE	FALSE		auto-confirm error and re-establish TCP connection
ModbusTCP Slave Instance					Implicit Function Block for Modbus Slaves.
FBType	STRING	'ModbusTCPSlaveUnit'	'ModbusTCPSlaveUnit'		
FBDiagType	STRING	'ModbusTCPSlaveUnit...'	'ModbusTCPSlaveUnit...'		
InitMethodName	STRING	'Initialize'	'Initialize'		

## CODESYS Integration

### 4.2.11.4 add MODBUS TCP slaves to Master

Modbus TCP Master -> Add Device -> Modbus TCP Slave

**Modbus\_TCP\_Slave**

General

Modbus Slave Channel

Modbus Slave Init

ModbusTCPslave Parameters

Modbus-TCP

Slave IP address: 192 . 168 . 0 . 1

Response timeout (ms): 1000

Port: 502

**Modbus\_TCP\_Slave**

Name	Access Type	Trigger	READ Offset	Length	Error Handling	WRITE Offset	Length	Comment
0 Channel 0	Read Input Registers (Function Code 04)	Cyclic, t=100ms	16#0000	4	Keep last Value			
1 Channel 1	Write Multiple Registers (Function Code 16)	Cyclic, t=100ms				16#000A	2	

Buttons: Move Up, Move Down, Add Channel..., Delete..., Edit...

#### Add Channel:

**ModbusChannel**

Channel

Name: Channel 2

Access type: Read Holding Registers (Function Code 3)

Trigger: Cyclic Cycle time (ms): 100

Comment:

READ Register

Offset: 0x0000

Length: 1

Error handling: Keep last Value

WRITE Register

Offset: 0x0000

Length: 1

Buttons: OK, Cancel

**Modbus\_TCP\_Slave**

Variable	Mapping	Channel	Address	Type	Unit	Description
		Channel 0	%IW0	ARRAY [0..3] OF WORD		Read Input Registers
		Channel 0[0]	%IW0	WORD		0x0000
		Channel 0[1]	%IW1	WORD		0x0001
		Channel 0[2]	%IW2	WORD		0x0002
		Channel 0[3]	%IW3	WORD		0x0003
		Channel 1	%QW0	ARRAY [0..1] OF WORD		Write Multiple Registers
		Channel 1[0]	%QW0	WORD		0x000A
		Channel 1[1]	%QW1	WORD		0x000B

## CODESYS Integration

Parameter	Type	Value	Default Value	Unit	Description
◆ NewChannelConfig	BOOL	true	true		Use the new Channel-Config format
◆ Unit-ID	USINT	255	16#FF		Unit-ID of the Device
◆ ResponseTimeout	DWORD	1000	1000		Maximum time for a Slave to respond in ms
◆ IPAddress	ARRAY[0..3] OF BYTE	[192, 168, 0, 1]	[192, 168, 0, 1]		Configure IP Address of TCP Slave.
◆ Port	UINT	502	502		Port where the slave is listening
◆ Channel 0					ChannelConfig
◆ Function Code	UINT	4			
◆ Read Offset	UINT	16#0000			
◆ Read Length	UINT	4			
◆ Write Offset	UINT	0			
◆ Write Length	UINT	0			
◆ Trigger	Enumeration of USINT	CYCLIC			
◆ Cycle Time	DWORD	100			
◆ Error Handling	Enumeration of BOOL	Keep last value			
◆ EnableRegisterBi...	BOOL				
◆ Channel 1					ChannelConfig
◆ Function Code	UINT	16			
◆ Read Offset	UINT	0			
◆ Read Length	UINT	0			
◆ Write Offset	UINT	16#000A			
◆ Write Length	UINT	2			
◆ Trigger	Enumeration of USINT	CYCLIC			
◆ Cycle Time	DWORD	100			
◆ Error Handling	Enumeration of BOOL	Keep last value			
◆ EnableRegisterBi...	BOOL				
◆ ConfigVersion	UDINT	16#03050B00	16#03050B00		

## CODESYS Integration

### 4.2.11.5 Modbus TCP Slave Device (Server):

**ModbusTCP\_Slave\_Device**

**General**

Modbus TCP Slave Device I/O Mapping

Modbus TCP Slave Device IEC Objects

Information

**Configured Parameters**

Watchdog 500 (ms)

Slave port 502

Unit ID

Holding registers 10 (%IW)  Writeable

Input registers 10 (%QW)

**Data Model**

**StartAddresses**

Coils 0

Discrete inputs 0

Holding register 0

Input register 0

Holding- and input register data areas overlay

**ModbusTCP\_Slave\_Device**

**General**

Modbus TCP Slave Device I/O Mapping

Modbus TCP Slave Device IEC Objects

Information

**Find** **Filter** **Show all** **Add FB for IO Channel...** **Go to Ir**

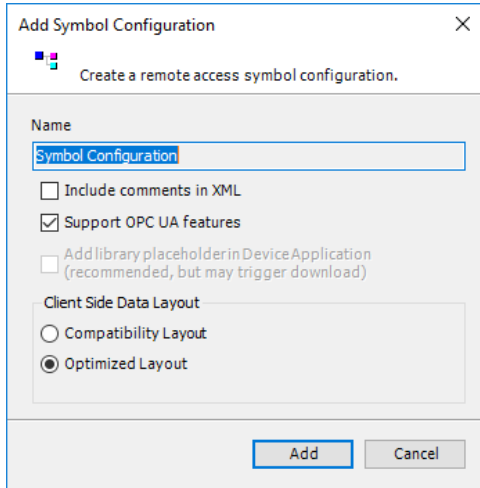
Variable	Mapping	Channel	Address	Type	Unit	Description
Application.AI_S1_5		Inputs	%IW0	ARRAY [0..9] OF WORD		Modbus Holding Registers
		Inputs[0]	%IW0	WORD		
		Inputs[1]	%IW1	WORD		
		Inputs[2]	%IW2	WORD		
		Inputs[3]	%IW3	WORD		
		Inputs[4]	%IW4	WORD		
		Inputs[5]	%IW5	WORD		
		Inputs[6]	%IW6	WORD		
		Inputs[7]	%IW7	WORD		
		Inputs[8]	%IW8	WORD		
		Inputs[9]	%IW9	WORD		
		Outputs	%QW0	ARRAY [0..9] OF WORD		Modbus Input Registers
		Outputs[0]	%QW0	WORD		
		Outputs[1]	%QW1	WORD		
		Outputs[2]	%QW2	WORD		
		Outputs[3]	%QW3	WORD		
		Outputs[4]	%QW4	WORD		
		Outputs[5]	%QW5	WORD		
		Outputs[6]	%QW6	WORD		
		Outputs[7]	%QW7	WORD		
		Outputs[8]	%QW8	WORD		
		Outputs[9]	%QW9	WORD		

## CODESYS Integration

### 4.2.12 OPC-UA Server

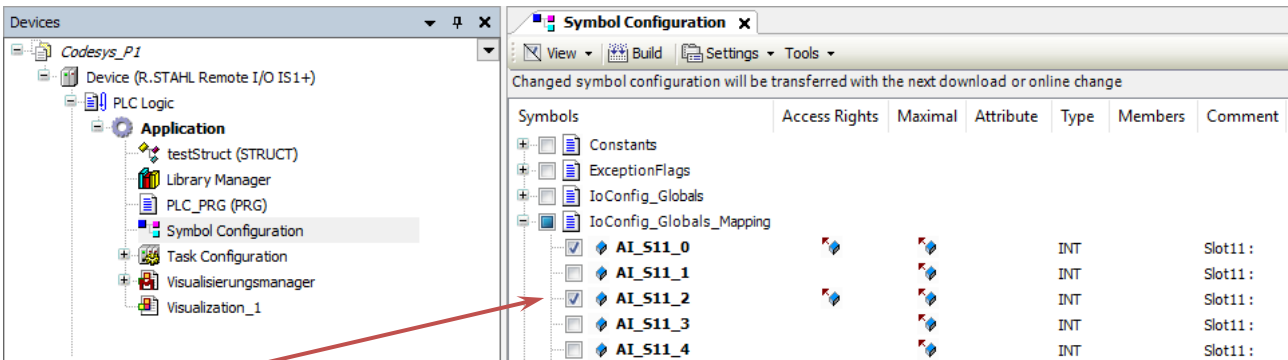
An OPC UA server is integrated in the IS1+ 9442 Codesys CPU.

Project engineering: Application -> Add Object -> Add Symbol Configuration



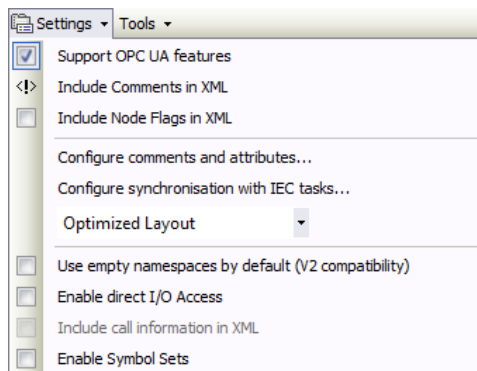
Support OPC UA features is activated by default.

A list of all symbols and data structures used in the project is generated automatically.



Symbols and data structures that are to be transferred via OPC-UA must be selected here.

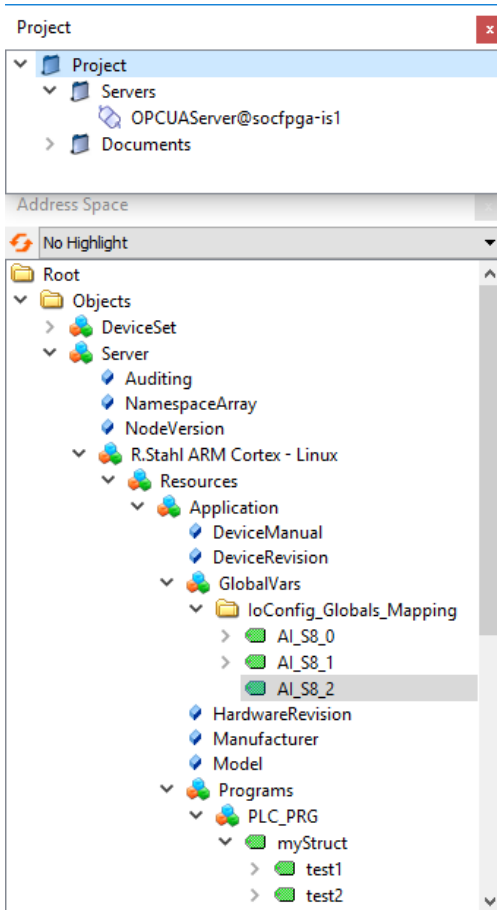
Optional functions are available under 'Settings':



## CODESYS Integration

After compiling and downloading the project into an IS1+ Codesys station, the selected data are available in the OPC-UA server in the IS1+ CPU.

This data can be accessed using OPC-UA clients.



Example of data access using UaExpert (OPC-UA Client from Unified Automation GmbH)

#	Server	Node Id	Display Name	Value	Datatype	Source Timestamp	Server Timestamp	Statuscode
1	OPCUAServer@socfpga-is1	NS4 String var R.Stahl ARM Cortex - Linux.Application.PLC_PRG.myStruct.test1	test1	1	Int16	11:54:17.608	11:54:17.608	Good
2	OPCUAServer@socfpga-is1	NS4 String var R.Stahl ARM Cortex - Linux.Application.PLC_PRG.myStruct.test2	test2	2	Int16	11:54:17.608	11:54:17.608	Good
3	OPCUAServer@socfpga-is1	NS4 String var R.Stahl ARM Cortex - Linux.Application.IoConfig_Globals_Mapping.AI_S8_0	AI_S8_0	-2	Int16	12:02:13.091	12:02:13.091	Good
4	OPCUAServer@socfpga-is1	NS4 String var R.Stahl ARM Cortex - Linux.Application.IoConfig_Globals_Mapping.AI_S8_1	AI_S8_1	-6911	Int16	12:02:10.355	12:02:10.355	Good
5	OPCUAServer@socfpga-is1	NS4 String var R.Stahl ARM Cortex - Linux.Application.IoConfig_Globals_Mapping.AI_S8_2	AI_S8_2	-32762	Int16	11:55:41.054	11:55:41.054	Good

## CODESYS Integration

### 4.3 'Hello World' in IEC 61131-3

Here is a simple Mini-PLC program in Structured Text (ST) as an example:  
A DI input signal (DI\_S4\_8) is mapped to a DO output (DO\_S5\_0).

```

PLC_PRG x
1 PROGRAM PLC_PRG
2 VAR
3
4 END_VAR
-----
1 DO_S5_0 := DI_S4_8;
    
```

```

1 DO_S5_0 FALSE := DI_S4_8 FALSE;
2 DO_S15 26880 := DO_S15 26880 +5;
3 test: 'Abcd1234' := 'Abcd1234';
4 RETURN
    
```

Variable values for debugging are displayed online.

Example: Creating and applying a data structure (data unit type DUT):

Add DUT ×

Create a new data unit type

Name  
TestStruct1

Type

Structure

Extends  ...

Enumeration

Textlistsupport

Alias

Base type  >

Union

```

TestStruct1 x
1 TYPE TestStruct1 :
2 STRUCT
3     test1: INT;
4     test2: INT;
5 END_STRUCT
6 END_TYPE
    
```

```

PLC_PRG x
1 PROGRAM PLC_PRG
2 VAR
3     test: STRING;
4     myStruct : TestStruct1;
5 END_VAR
-----
1 DO_S5_0 := DI_S4_8;
2 DO_S15 := DO_S15+5;
3 test := 'Abcd1234';
4 myStruct.test1 := 1;
5 myStruct.test2 := 2;
    
```

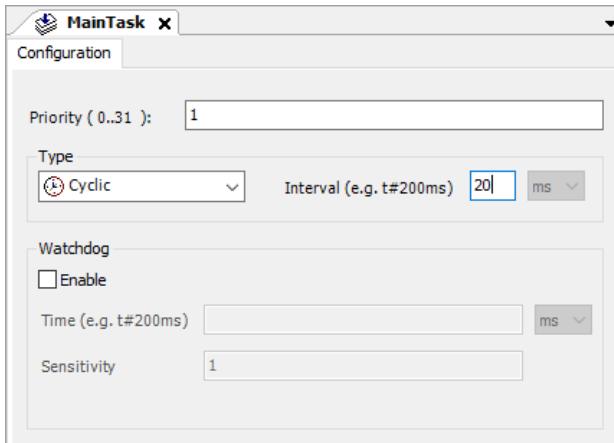
## CODESYS Integration

### 4.4 System Analysis

#### 4.4.1 Cycle time

##### Application -> Task configuration -> Maintask

The time behaviour of the program execution is set here.



##### Application -> Task configuration

Shows current and statistical data about the real time behavior of the configured tasks.

Task Configuration											
Task	Status	IEC-Cycle Count	Cycle Count	Last Cycle Time (µs)	Average Cycle Time (µs)	Max. Cycle Time (µs)	Min. Cycle Time (µs)	Jitter (µs)	Min. Jitter (µs)	Max. Jitter (µs)	
MainTask	Valid	3972	3972	665	580	1092	507	476	-227	249	
VISU_TASK	Valid	1617	1617	65	112	1210	54	2440	-1219	1221	

Right mouse click on line -> Reset drag pointers for min and max values.

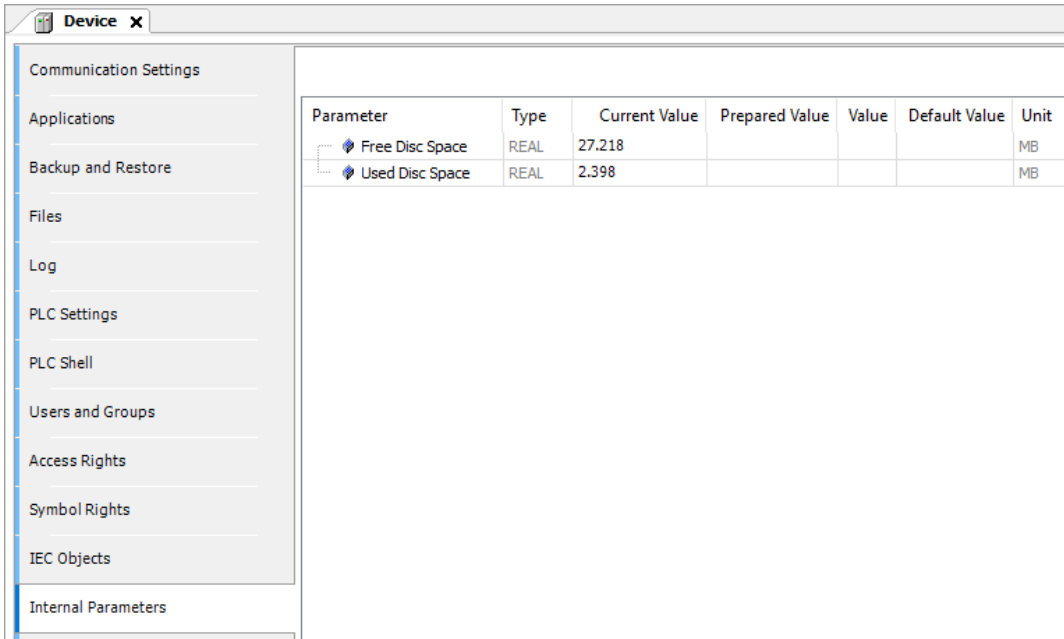


## CODESYS Integration

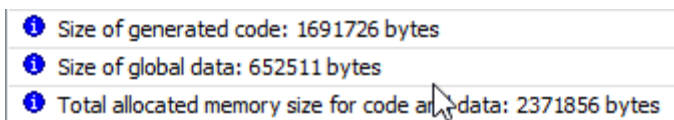
### 4.4.2 Memory Utilization

#### FLASH:

A total of 27 MB FLASH memory is available for program code and visualizations.  
 Display at 'Internal Parameters' of the IS1+ RIO:



In addition, after compilation and download of program code and visualizations, the storage space occupied by CODESYS is displayed in ' Messages':



#### RAM:

Display at ' Messages':



## CODESYS Integration

### 4.5 Update

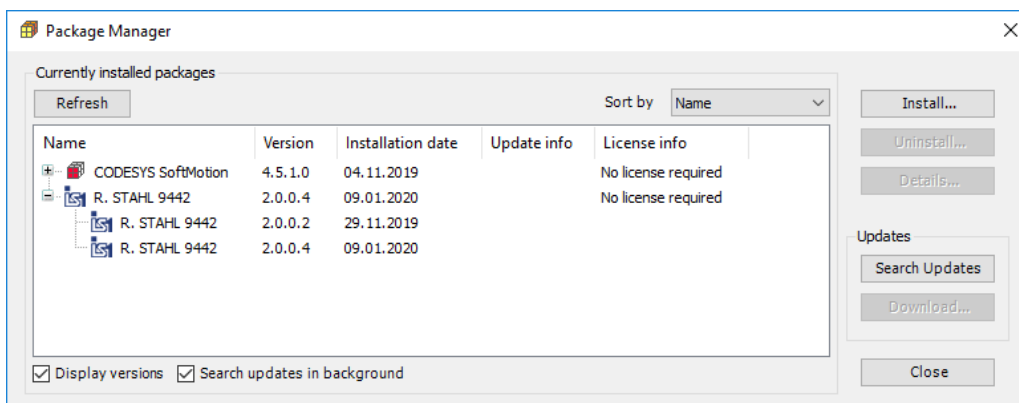
#### IS1+ 9442 CPU Firmware Update:

Firmware updates of the IS1+ 9442 CPU can be loaded via the IS1+ web server if required. For this purpose, the PLC must first be set to the Stop state via the Codesys Development Tool. An IS1+ CPU firmware download via the IS1+ web page is blocked when the PLC is in RUN mode. (LED 'AS EXCH' = On)

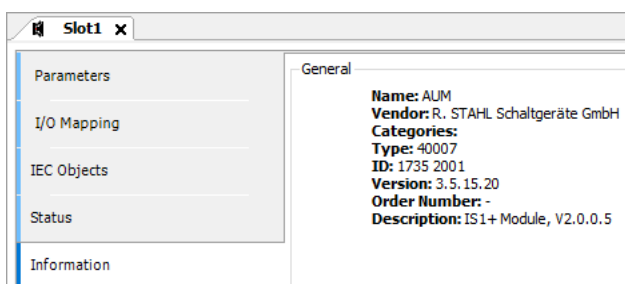
#### IS1+ Codesys package:

A suitable IS1+ Codesys Package is delivered together with a IS1+ 9442 Codesys Firmware File (.SWU). The IS1+ Package version currently installed in the Codesys Development Tool can be displayed in Tools -> Package Manager.

Older parallel installed Package Versions can be displayed in the Package Manager via 'Display Versions'. If the new IS1+ package version is not yet installed in the Codesys Development Tool, the new package supplied must be installed.



A newly installed IS1+ package version is not always automatically transferred to existing Codesys configurations. Check the package version used in the project under 'Information -> Description' at Device, BusRail and I/O module level.



If required, after the package installation in the project, the function 'Update Device' has to be called up at device level and for each configured I/O module.

Instead of manually calling the function 'Update Device' for all configured I/O modules, the attached script 'updateTree.py' can be started, which automates the update of the I/O modules.  
call: Tools -> Scripting -> Execute Script File

After the update has been completed, the project must be recompiled and loaded into the IS1+ CPUs.

## CODESYS Integration

### 5 Security

R. STAHL Security Whitepaper. .... TBD

CODESYS Security Hints:

<https://www.codesys.com/security/codesys-security.html>

CODESYS Security Whitepaper:

<https://de.codesys.com/fileadmin/data/customers/security/CODESYS-Security-Whitepaper.pdf>

### 6 Liability

Exclusion of liability of R. STAHL for consequences of faulty PLC project planning .... TBD .....

### 7 List of abbreviations

AS	<b>A</b> utomation <b>S</b> ystem
AIM	<b>A</b> nalogue <b>I</b> nput <b>M</b> odule
AIMH	<b>A</b> nalogue <b>I</b> nput <b>M</b> odule + <b>H</b> ART
AUMH	<b>A</b> nalog <b>U</b> niversal <b>M</b> odule AI/AO with <b>H</b> ART
SAIMH	<b>S</b> afety <b>A</b> nalogue <b>I</b> nput <b>M</b> odule + <b>H</b> ART (PROFIsafe)
AOM	<b>A</b> nalogue <b>O</b> utput <b>M</b> odule
DIM	<b>D</b> igital <b>I</b> nput <b>M</b> odule
DIOM	<b>D</b> igital <b>I</b> nput <b>O</b> utput <b>M</b> odule
DOM	<b>D</b> igital <b>O</b> utput <b>M</b> odule
DOMR	<b>D</b> igital <b>O</b> utput <b>M</b> odule <b>R</b> elays
DOMV	<b>D</b> igital <b>O</b> utput <b>M</b> odule <b>V</b> alves
HW	Hardware
IOM	General description of <b>I/O</b> <b>M</b> odule
MQTT	<b>M</b> essage <b>Q</b> ueuing <b>T</b> elemetry <b>T</b> ransport
OPC-UA	<b>O</b> pen <b>P</b> latform <b>C</b> ommunications - <b>U</b> nified <b>A</b> rchitecture
PM	<b>P</b> ower <b>M</b> odule
SW	<b>S</b> oftware
TIM	<b>T</b> emperature <b>I</b> nput <b>M</b> odule
FCG	<b>F</b> ield <b>C</b> omm <b>G</b> roup (former HART Communication Foundation HCF)
HART	<b>H</b> ighway <b>A</b> ddressable <b>R</b> emote <b>T</b> ransducer
PV	<b>P</b> rimary <b>V</b> ariable
SV	<b>S</b> econdary <b>V</b> ariable
TV	<b>T</b> ertiary <b>V</b> ariable
QV	<b>Q</b> uaternary <b>V</b> ariable



## CODESYS Integration

### 8 Version changes

Version Operating Instructions	9442 CPU Firmware	Extensions / Changes
B0.11	V1-0-xx-yyyy	Work
		First released version

### 9 Literature references

<https://www.codesys.com/>

<https://en.wikipedia.org/wiki/CODESYS>

[https://en.wikipedia.org/wiki/IEC\\_61131](https://en.wikipedia.org/wiki/IEC_61131)

### 10 Support Address

#### IS1+ Support:

**R. STAHL Schaltgeräte GmbH**

Business Unit Automation Interface and Solutions

eMail: [support.automation@r-stahl.com](mailto:support.automation@r-stahl.com)

Support information: <http://www.r-stahl.com>

Service Hotline IS1+: +49 (7942) 943-4123

Telefax : +49 (7942) 943-40 4123

**CODESYS Support:** <https://www.codesys.com/support-training.html>

**CODESYS Forum:** <https://forum.codesys.com/>