

echocollect



Version: EN-1606-1.01

© Copyright 2014 - 2019 Softing Industrial Automation GmbH



Disclaimer of liability

The information contained in these instructions corresponds to the technical status at the time of printing of it and is passed on with the best of our knowledge. The information in these instructions is in no event a basis for warranty claims or contractual agreements concerning the described products, and may especially not be deemed as warranty concerning the quality and durability pursuant to Sec. 443 German Civil Code. We reserve the right to make any alterations or improvements to these instructions without prior notice. The actual design of products may deviate from the information contained in the instructions if technical alterations and product improvements so require.

It may not, in part or in its entirety, be reproduced, copied, or transferred into electronic media.

Softing Industrial Automation GmbH

Richard-Reitzner-Allee 6
85540 Haar / Germany
<http://industrial.softing.com>



+ 49 89 4 56 56-0



+ 49 89 4 56 56-488



info.automation@softing.com

support.automation@softing.com

The latest version of this manual is available in the Softing download area at:
<http://industrial.softing.com/en/downloads.html>

Table of Contents

Chapter 1	Help Overview.....	9
Chapter 2	echocollect.....	10
2.1	Introduction	10
2.2	Scope of Delivery.....	11
2.3	Features	11
2.4	Specifications	13
2.5	Hardware	14
2.6	Getting Started.....	18
Chapter 3	NetCon – General.....	20
3.1	Overview	20
3.2	System Requirements.....	21
3.3	License Conditions.....	21
Chapter 4	Basics.....	24
4.1	NetCon echo	24
4.2	Logger	24
4.3	ARP	25
4.4	PC-H1 Driver	25
Chapter 5	PLC Connections.....	26
5.1	S7 – General Information.....	26
5.2	S5 – General Information.....	26
5.3	Modbus – General Information.....	27
5.4	CLX – General Information.....	27
5.5	PLC-5 / SLC – General Information.....	27
5.6	MELSEC-Q – General Information.....	28
5.7	Send/Receive – General Information.....	28
5.8	OPCpipe – General Information.....	29
5.9	AS511	29
5.10	Telnet	30
5.11	3964(R)	30
5.11.1	RK 512	30
5.12	MPI	30
5.13	Modbus RTU	31
5.14	DF1	32
Chapter 6	Operation and Configuration.....	33

6.1	How to Proceed.....	33
6.2	Basic Configuration.....	33
6.3	Licensing	34
Chapter 7	Windows.....	35
7.1	Softing NetCon Configuration.....	35
7.1.1	Select a Station in Your Network	36
7.1.2	Edit Access to a Station in Your List	37
7.1.3	Station Not Found	38
7.1.4	Offline Configuration	38
7.2	NetCon Main Window.....	39
7.3	Connections List.....	40
7.4	Logger	41
7.5	Collect Frame Tables.....	42
Chapter 8	Connection.....	43
8.1	New Connection.....	43
8.1.1	New Logic Table Connection	44
8.1.2	New Logic Table Connection – Serial	45
8.2	Connection Settings.....	46
8.2.1	Network Protocol	46
8.2.2	Network PLC Protocol	46
8.2.3	TCP/IP Connect Parameters	47
8.2.3.1	TCP/IP Special Settings.....	48
8.2.3.2	IP Address	49
8.2.3.3	Port	50
8.2.3.4	TSAP	50
8.2.3.5	TSAP for S7 Connections.....	51
8.2.3.6	PLC Header.....	52
8.2.4	H1 Connect Parameters	53
8.2.5	Collect Destination Connection Config.	54
8.2.5.1	Server Connections.....	55
8.2.5.2	File Upload Connection Edit.....	56
8.2.6	S7 Protocol Settings	56
8.2.7	echocollect Connection: Modbus Protocol	56
8.2.8	echochange Connection: CLX Protocol	56
8.2.9	echochange Connection: SLC / PLC-5 Protocol	56
8.2.10	Melsec-Q Protocol	57
8.2.11	OPCpipe Parameters	57

8.2.12	Collect Memory Target	58
8.2.13	OPC UA Server Settings	58
8.2.13.1	OPC UA Certificates.....	59
8.2.14	Serial Protocol	59
8.2.14.1	Telnet Connection (Send/Receive).....	60
8.2.14.2	3964 and 3964R Connection.....	61
8.2.14.3	AS511 (Pg S5) Connection.....	62
8.2.14.4	MPI Connection.....	62
8.2.14.5	Modbus RTU Connection.....	62
8.2.14.6	DF1 Connection.....	63
8.2.14.7	Serial Line Parameters.....	64
8.2.15	Database Connect Parameters	65
8.2.15.1	MySQL Connection Flags.....	68
8.2.15.2	Oracle Connection Settings.....	68
8.2.15.3	MsSQL/Sybase Connection Settings.....	68
8.2.16	MQTT Broker Configuration	69
8.2.16.1	Connection Settings.....	69
8.2.16.2	Communication Settings.....	69
	MQTT Broker Certificates.....	70
	MQTT Broker Certificate Validation.....	70
	MQTT Client Certificate.....	70
8.2.16.3	MQTT Topic Settings	71
8.2.16.4	MQTT Topic Selection.....	71
8.2.16.5	MQTT Last Will Settings.....	71
8.2.17	Memory Table Definition	72
8.2.18	Mail Connect Parameters	73
8.2.19	Collect Slave Config. Edit	73
8.3	Edit Connection.....	74
8.4	Copy Connection.....	75
8.5	Delete Connection.....	75
8.6	Disable Connection.....	75
Chapter 9	Main Menu.....	76
9.1	File	77
9.2	Connection	77
9.3	Diagnostics	78
9.3.1	Device Connection Diagnostics (1 Connection)	79
9.3.2	MPI Ring Diagnostics	79
9.3.3	TCP/IP Stati	79
9.3.4	Logger	79

9.3.5	Logger Settings	80
9.3.6	Serial Diagnostics	81
9.3.7	Using the Diagnostic Text Output System	81
9.4	Collect	81
9.5	Station	82
9.5.1	Station Password	82
9.5.2	Station Parameters	82
9.5.3	MPI Station Settings	84
9.5.3.1	MPI Station Settings.....	84
9.5.3.2	MPI Station Extended Settings.....	84
9.5.4	Device SSH Settings	85
9.6	Settings	86
9.6.1	H1 System Settings	86
9.6.2	TCP/IP System Settings	87
9.6.3	Device System Settings	89
9.6.4	Diagnose of Default Connections	89
9.6.5	Clock Settings	90
9.6.6	Device Service Settings	90
9.6.7	Reboot	90
9.7	Help	91
9.7.1	Versions	91
9.7.1.1	Hardware Versions.....	91
9.7.2	About	91
9.7.3	License Overview	91
9.7.4	Licenses	92
Chapter 10	Collect Frame Menu.....	93
10.1	File	94
10.2	Table	95
10.2.1	Collect Create Frame	95
10.3	Elements	96
10.3.1	Collect Edit Frame Entry	96
10.3.1.1	Collect Browsing.....	97
10.3.1.2	Collect Edit Array Settings.....	98
10.3.1.3	Collect Edit Date Entry.....	98
10.3.2	Collect Edit Type Entry	98
10.3.3	Collect Indexed Text List	99
10.3.4	Collect Database Naming	99

10.3.5	Collect Browsing Databases - Tables	100
10.3.6	Collect Browsing Databases - Columns	100
10.3.7	Collect Browsing Databases - Functions/Procedures	100
10.4	Online Diag	100
10.5	Special Settings.....	101
10.5.1	Collect PLC Confirm	102
10.5.2	Collect Trigger Settings	102
10.5.3	Collect Edit Normalisation Rule	102
10.5.4	Collect Edit Rule	103
10.5.5	Collect Memory Target	103
10.5.6	Collect Edit Frame Chaining	103
10.5.6.1	Collect Frames List.....	103
10.5.7	Select Address	103
10.5.8	Collect Indexed Text Table Edit	104
10.5.8.1	Collect Indexed Text Edit.....	104
10.5.8.2	Collect Add Indexed Text.....	104
10.5.9	Collect Symbol Table Edit	104
10.5.9.1	Collect New Symbol Table.....	105
10.5.9.2	Collect Symbol Table Edit.....	105
10.5.9.3	Collect Symbol Edit.....	105
10.5.9.4	S7 Symbol Import.....	106
10.5.10	Collect Address Book	106
10.5.10.1	Collect Add Address Book Entry.....	106
10.5.11	Collect Code Generator Settings	106
10.6	Edit Mode	107
10.6.1	Collect Display Settings	107
10.7	Help	108
Chapter 11	Item Syntax.....	109
11.1	S7 Item Syntax.....	110
11.2	S5 Item Syntax.....	113
11.3	Modbus Item Syntax.....	117
11.4	PLC-5 / SLC Item Syntax.....	121
11.5	CLX Item Syntax.....	123
11.6	MELSEC-Q Item Syntax.....	125
11.7	Send/Receive Item Syntax.....	128
11.8	Extended Raw Data Item Syntax.....	130
11.9	OPCpipe Item Syntax.....	132
11.10	Special Syntax for Trigger/Normalisation.....	134

11.11	Virtual Collect Item Syntax.....	137
11.12	Arrays	141
11.13	Suffixes	142
Index	145

1 Help Overview

NetCon echo for echocollect

[echocollect](#)

[NetCon - General](#)

[Basics](#)

[PLC Connections](#)

[Operation and Configuration](#)

[Windows](#)

[Connection](#)

[Main Menu](#)

[Collect Frame Menu](#)

[Item Syntax](#)

2 echocollect

This chapter covers the following topics:

- [Introduction](#)
- [Scope of Delivery](#)
- [Features](#)
- [Specifications](#)
- [Hardware](#)
- [Getting Started](#)

2.1 Introduction

echocollect is a device from the “echo” family that is mountable on the top hat rail and features two Ethernet ports (allowing separation of production network from office network), which can be integrated directly into the production network. echocollect can be optionally equipped with a serial interface in order to collect data from PLCs that have no Ethernet connections.

How echocollect works: First, all the relevant data is gathered from the distributed PLCs according to configurable trigger conditions. This data is then stored in a process image, which can be timestamped, if required. The process image can also be extended with database commands right away, which can then be sent, in its entirety, directly to the database computer via the network. To buffer the data in case of interruptions, echocollect saves all the gathered data to a memory card. This ensures that no process data from the plant will be lost. In the opposite direction, data packets such as recipes can be transferred from the database to echocollect. From there, the relevant variables are then extracted and distributed to the assigned PLCs.

If the operating system of the echocollect detects a problem with the file system of the memory card, then it stops to write to the memory card. Please replace the memory card in such cases.

The system can also be used as a data logger, for monitoring the operating state of a plant. The basic version is able to collect data from up to five networked PLCs and store it in up to ten process images, as desired. The process images can be independently transmitted to different databases, which can also be on different destination systems. Using configurable triggers, specific conditions in connected plant areas can be quickly identified and the relevant data can be immediately transmitted if an event occurs. In any case echocollect works non-reactive and does not interfere with the programs of the existing PLCs.

2.2 Scope of Delivery

Before you commission the echocollect device, please make sure that the echocollect package contains the following elements:

- echocollect e (Ethernet) or echocollect r / p / m (serial and Ethernet)
- CD-ROM with NetCon echo configuration software
- Demo software
- Manuals
- echocollect Quick Start
- 2GB SD memory card
- 24V male connector

NOTE:

An AC adapter is not included in delivery. Please order it separately under the following order number:

AC adapter, 24V DC, 700-5596-04

Please also indicate the required plug type (Euro, UK, US) in your order.

If you want to use your own AC adapter, please observe the polarity of the 24V connector (positive pole at the top, negative pole at the bottom).

2.3 Features

echocollect interconnects PLCs with databases and serves as data logger. The first version of the device – echocollect e – is used when PLCs are controlled exclusively via the network. If the PLCs are connected serially or via MPI, the second version – echocollect se – is used.

echocollect gathers the process data and transfers it into a database structure (SQL). It is irrelevant which ERP system is used. The data collector provides the data in the required format. The device featuring two Ethernet interfaces also acts as a gateway and provides the possibility to separate the office network from the plant network. The allocation is performed using tables, thus allowing OPC independent operation.

echocollect e

Network access

- Converts data from different Ethernet protocols
- Bridges two separate networks using different protocols
- Connects computers with standard TCP/IP to Industrial Ethernet
- Converts the following Ethernet protocols

Supported transport protocols

- TCP
- UDP
- IP
- RFC 1006
- PLC Header
- ISO (H1)

Supported application protocols

- Send / Receive (raw data)
- Modbus over TCP
- EthernetIP

- CSP/AB Ethernet (SLC 500)

echocollect r/p/m

Fieldbus and serial access via standard interfaces or MPI:

Devices originally not designed for networking (and thus without network adapter) are able to communicate via network. The serial device is connected to echocollect via its own serial interface. The echo device is connected to Ethernet via the network adapter. This way serial devices can be reached on the network. echocollect is very versatile and supports a wide range of serial communication protocols:

- Telnet
- AS511
- MPI/PPI
- Modbus RTU

On the Ethernet side, echocollect supports the transport protocols TCP and ISO (H1). In addition, the higher-level protocols S7, S5, Modbus over TCP, EtherNet/IP are implemented. The device supports complex applications such as PLC programming or process data visualization.

Echocollect is designed to provide the best possible solution to the special demands of industrial automation.

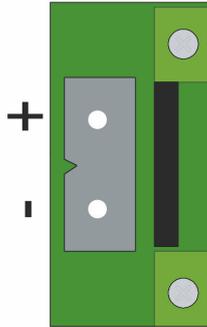
2.4 Specifications

Parameter	echocollect e	echocollect r / p / m
Serial addressable PLCs		<ul style="list-style-type: none"> • Simatic S7 (MPI) • Simatic S5 (AS 511) • PLCs with RK512, 3964(R) • PLCs with Modbus RTU
Network addressable PLCs	<ul style="list-style-type: none"> • Simatic S7 • Rockwell CLX • Rockwell SLC-500 • Simatic S5 • PLCs with Modbus over TCP • Wago, Phoenix, Beckhoff, etc. 	<ul style="list-style-type: none"> • Simatic S7 • Rockwell CLX • Rockwell SLC-500 • Simatic S5 • PLCs with Modbus over TCP • Wago, Phoenix, Beckhoff, etc.
Ethernet interface	2 x TP (RJ45), 10/100 MBit	2 x TP (RJ45), 10/100 MBit
Serial interfaces		<ul style="list-style-type: none"> • RS232 • RS485 • TTY 20mA • MPI
Power supply	24V DC, 1.1A (± 20 %)	24V DC, 1.1A (± 20 %)
Storage temperature	-20°C .. 70°C	-20°C .. 70°C
Operating temperature	5°C .. 55°C	5°C .. 55°C
Dimensions (in mm)	H = 131, W = 47, L = 111	H = 131, W = 110, L = 111
Mounting	35 mm top hat rail (DIN rail)	35 mm top hat rail (DIN rail)
Protection class of housing	IP20	IP20
AC adapter	External	External
Relative humidity	10 % ... 95 % (non-condensing)	10 % ... 95 % (non-condensing)
Altitude	must not exceed 2,000 m	must not exceed 2,000 m
Location	Indoor use only, no direct sunlight	Indoor use only, no direct sunlight

2.5 Hardware

Power supply

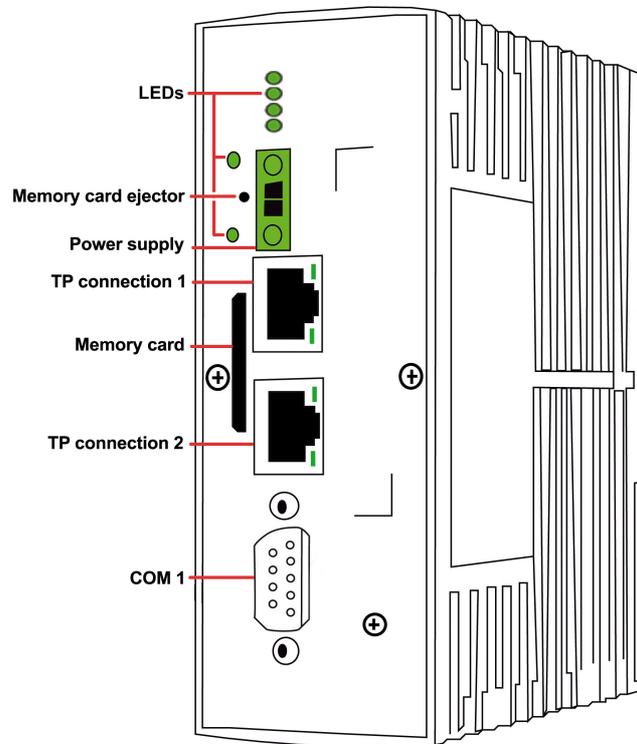
An AC adapter 230V / 24V is not included in delivery. Please order it separately. If you want to use your own AC adapter, please observe the polarity of the 24V DC connector (positive pole at the top, black wire of the cable, negative pole at the bottom, white wire of the cable).



Memory card

To eject the memory card, simply take the card out of the slot. Please do not insert any objects in the "SD eject" hole.

echocollect e

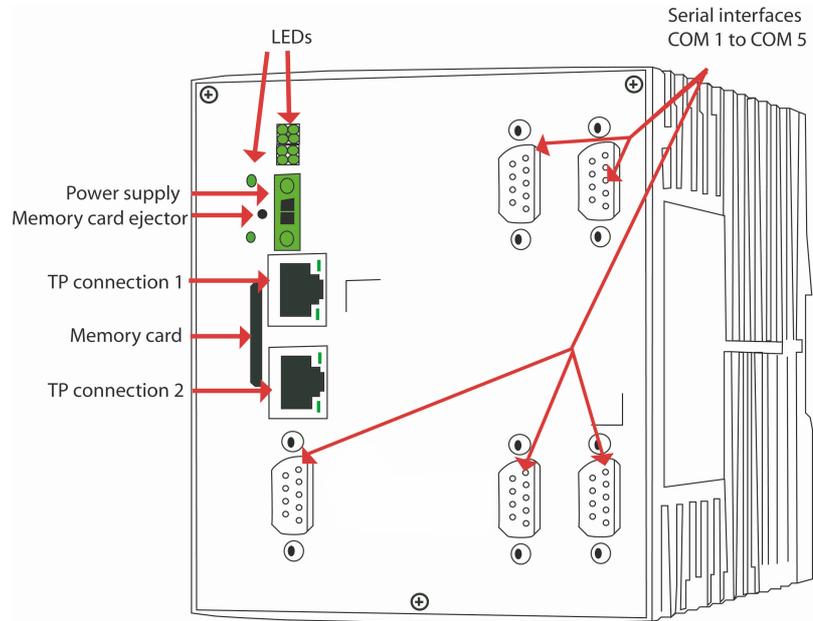


LEDs

The 5 LEDs have the following meanings:

LED	Meaning for echocollect e
Power	echo device correctly connected to the power supply
SD	SD card inserted
Conn	Indicates that a configured connection has been established
Eth 1	Data transfer to TP1
Eth 2	Data transfer to TP2
COM1	TTY Serial Port

echocollect r/ p / m



Serial interface

The serial connections are provided for connecting echocollect to a device with serial interface, e.g. a SIMATIC S7, a SIMATIC S5, etc.

Depending on the device version, different interface versions are possible. The following interfaces are available:

- MPI
- TTY (20mA)
- RS232
- RS485, non-floating

PIN No. on D-Sub	MPI	TTY (20mA)	RS232	RS485
1	- not assigned -	20mA_1	DCD	- not assigned -
2	- not assigned -	20mA_2	RxD	- not assigned -
3	Data B	TxD	TxD	Data+
4	RTS AB	RxD	DTR	- not assigned -
5	GND	GND	GND	GND
6	+5V	-RxD	DSR	- not assigned -
7	- not assigned -	GND	RTS	Term+
8	Data A	GND	CTS	Term-
9	- not assigned -	-TxD	-TxD	Data-

LEDs

The LEDs have the following meanings:

LED	echocollect se
Conn	Indicates that a configured connection has been established
Eth 1	Data transfer to TP1
Eth 2	Data transfer to TP2
Power	echocollect correctly connected to the power supply
Com1-Com5	Payload is being transmitted on the serial port
SD	SD card inserted

Intended Use

echocollect is a gateway device from the “echo” family that is mountable on the top hat rail and features two Ethernet ports (allowing separation of production network from office network), which can be integrated directly into the production network.



The echocollect is only approved for intended and appropriate use. In case of a non-compliant use, an impairment to the safety protection may be caused.

Installation

echocollect shall be installed within another enclosure that supplies the safety aspects protecting the operator from mechanical hazards.



Note

The installation orientation must be as displayed in the figure above.



Note



This symbol shows the terminal where functional earth shall be connected to the device.



This symbol indicates that only DC (direct current) can be connected



Note

Consider during installation and positioning that disconnecting from supply source is easily possible.

Maintenance

According to the local conditions and in compliance with the environmental rules, the operator is responsible to visually inspect the system and to remove dust settlements in a regularly interval (every 6 months). Do not use liquids for cleaning.



Note

Do not open the housing of the echocollect. It does not contain any parts that need to be maintained or repaired. In the event of a fault or defect, remove the device and return it to the vendor.

Opening the device will void the warranty!

2.6 Getting Started

Before working with the device, you need to carry out the following steps:

1. Mounting
2. Connection to the AC adapter
3. Connection to the Ethernet network
4. Installation of the NetCon configuration software
5. Connection of PC and device
6. Basic configuration

Mounting

- Install or remove the device only when it is switched off
- Horizontal mounting
- Keep the device away from heat and electrical interference
- Do not cover the cooling slots on the top and bottom
- Leave enough space for ventilation

The echo device can be mounted on a top hat rail. Additional accessories are not required for top hat rail mounting. The device is provided with a snap lock for easy mounting on a top hat rail (DIN rail). Do the following:

1. Hang the rear panel of the device on the DIN rail.
2. Swivel the device down in the direction of the DIN rail. Make sure that the device properly snaps into place and is securely fastened to the rail.

Connection to the AC adapter

Connect the AC adapter with the device and then plug the adapter into the AC mains power supply.

NOTE:

The device has no ON/OFF switch. It switches on automatically when the AC adapter is plugged into the mains power supply.

Wait until only the green Power LED is on. This means that the device is ready for operation.

Connection to the Ethernet network

The device has an RJ45 socket for Ethernet connection.

- Connect one end of a TP cable with the TP socket (RJ45 socket) on the front panel of the device. Plug the other end into an Ethernet hub/switch connected to your network.
- The LEDs Eth 1 and/or Eth 2 should blink now. If they don't, please check the network connection and cabling.

Installation of the configuration software

Using the NetCon echo configuration software, you define the basic configuration of the device and set the parameters for your connections. The defined parameter settings are either transferred directly to the device or stored locally for later transmission. The configuration software allows you to monitor and diagnose configured connections. To configure the device, you will need a PC on which the NetCon echo configuration software is installed. PC and device have to be connected either directly or indirectly (see chapter 2.5). To install the configuration software, do the following:

1. Insert the CD-ROM in your CD drive.
2. Select the "echocollect" installation.
3. You will be automatically guided through the installation process.

Connection of PC and device

A connection between PC and device is required for:

- Online configuration
- Transfer of a parameter file from PC to device and vice versa
- Diagnostic and test functions
- Programming of the S5 / S7 via network

The connection between PC and device can be established indirectly via bus.

Basic configuration

The echo device is configured with the NetCon software.

Resetting the device to default settings

- Start the configuration software.
- Select "Offline".
- Enter the file name "netfile.net". This file must be located in the main directory of the SD card.
- Set the Station Parameters to the desired values. For the MAC address, you can either leave 00 00 00 00 00 00 – in this case, the hardware address of the device will be used – or you can enter the desired MAC address.
- Exit the configuration software.
- Insert the card in the device.
- Restart the device.

The device has the specified parameters; all the connections have been removed. You can use any configuration file you have created.

NOTE:

Remove the card from the device. If you leave it inserted, all the parameter settings will be reset every time you switch on the device.

3 NetCon – General

This chapter provides general information about NetCon:

- [Overview](#)
- [System Requirements](#)
- [License Conditions](#)

3.1 Overview

echo

The devices of the **echo** product family allow you to exchange data between field devices (PLC) from many different manufacturers as well as communication servers, such as OPC servers or DDE servers, databases and other communication partners. The access path is routed via Ethernet TCP/IP, RFC 1006, ISO (H1), MPI, Profibus, RS232, RS485, RS422 or TTY (20mA).

NetCon echo

The user interface for device configuration and diagnostics is provided by the **NetCon** software application. The GUI connects to the devices via TCP/IP (port 982, 997) or via H1.

Access protection

A password can be set to prevent unauthorized modification of the configuration. Accessing devices with multiple NetCons at the same time is not recommended. Because it causes the danger of interfere with each other NetCon user.

Logger

The logger is integrated in the devices. The texts are logged in the device. Using NetCon, you can monitor and configure the logger.

See also [Logger](#).

Help

NetCon provides a context-sensitive help system. You can call the online help from any dialog box either by pressing the **F1 key** or clicking the **Help** button. This displays a help page with a detailed description of the elements and input options in that dialog box.

Manual

Every device comes with a manual in PDF format. It includes the chapters describing the hardware and commissioning, and the online help of the NetCon software. Free PDF readers are available on the Internet from <http://www.adobe.com> or <http://www.foxitsoftware.com>.

3.2 System Requirements

To use NetCon, the following minimum system requirements must be met:

Processor:	Intel Pentium II 200 MHz processor
RAM:	Minimum: 64 MB RAM
Hard disk capacity:	500 MB memory space on hard drive
Operating system:	Windows XP Windows Server 2003, 32 Bit Windows Server 2008, 32 Bit or 64 Bit Windows Server 2008 R2, 64 Bit Windows 7, 32 Bit or 64 Bit

3.3 License Conditions

Please read the following conditions carefully before you install the software product. You must declare that you accept the following conditions either when you buy the software product or, at the latest, when it is installed. If you do not accept these conditions, send the software and the manual back to us within 14 days starting with the date of the invoice. The purchase price can only be reimbursed within the specified time frame.

I. Preamble

The object of the contract is the computer program (called "SOFTWARE PRODUCT" hereafter) stored on the data medium. The LICENSER states that the state of technology makes it impossible to make computer software so that it will work correctly in all applications and combinations. Thus the object of the contract is only a piece of software which works correctly in the sense of the program description and user's guide. The SOFTWARE PRODUCT including all contents (e.g., figures, graphics, text and sample applications) is and will remain the property of the LICENSER and is thus protected by copyright. The LICENSER is willing to give user's rights as defined in this contract in return for an appropriate licensing fee.

This license contract is concluded between LICENSEE and the LICENSER by the acceptance of the SOFTWARE PRODUCT by the LICENSEE.

II. Terms

LICENSER

Softing Industrial Automation GmbH
Richard-Reitzner-Allee 6
D-85540 Haar

LICENSEE

Natural or legal persons who purchase this license for the purpose of using the SOFTWARE PRODUCT.

THIRD PARTIES

Other natural or legal persons.

III. Scope

1. The SOFTWARE PRODUCT is exclusively licensed for use by the LICENSEE. The LICENSEE may sell the SOFTWARE PRODUCT to THIRD PARTIES under the condition that the THIRD PARTY agrees to this software licensing contract. In this case, the software license contract is invalid for the LICENSEE and the utilization rights to the SOFTWARE PRODUCT are cancelled for him since these rights have been passed on to a THIRD PARTY who has now become the LICENSEE him/herself.
2. The SOFTWARE PRODUCT may only be used on a single computer. The LICENSEE is obligated to purchase a separate license for every computer on which the SOFTWARE PRODUCT is used. A different agreement between LICENSEE and LICENSER applies to the purchase of multiple and network licenses.
3. The duplication of the SOFTWARE PRODUCT and all related documentations is forbidden except for the creation of an archive copy for exclusive use by the LICENSEE.
If the LICENSEE's original data medium becomes defective, it can be exchanged at the LICENSER for a replacement data medium. The cost of the exchange will be carried by the LICENSEE.

4. The SOFTWARE PRODUCT is delivered on data medium as a demo version with restricted function scope.
The demo version may not be used for commercial purposes. After payment of the licensing fee for the particular desired version, the LICENSEE will receive a code number from the LICENSER for the release of the corresponding SOFTWARE PRODUCT.
5. If a new program version is given to the LICENSEE due to the purchase of an update or for any other reason, the user's rights for the earlier program versions become void. This means that you may only work with the latest version.
6. The LICENSEE and THIRD PARTIES are forbidden to modify the SOFTWARE PRODUCT or reverse engineer it (i.e., de-compile or disassemble it).
The LICENSEE is liable for all damages which occur because of the violation of these conditions.

IV. Limited Warranty

The data carriers on which the SOFTWARE PRODUCT was delivered are free from material and manufacturer's flaws during normal use during a period of time of 6 months starting on the delivery date. The data carriers are free of viruses according to the LICENSER. If, however hidden viruses have snuck in, the LICENSER will not be liable for any subsequent damages which may occur.

1. The preceding limited warranty does not cover data carriers which have been damaged by chance or misuse or manipulation of unauthorized parties (people other than employees of the LICENSER).
2. During the stated guarantee period, the LICENSER is obligated to replace a defective data carrier if this was sent to the LICENSER with a copy of the invoice. Other claims, in particular damage claims, will not be honored subject to para. 4 and Roman numeral V.
3. No further guarantee claims will be allowed. In particular, no guarantee claims will be accepted for program content and its freedom from errors or suitability for certain purposes. The responsibility for software and hardware selection, for installation, use, expected results and data protection and data backup with backup copies is the exclusive business of the LICENSEE.
4. Exceptions to the preceding warranty restrictions are promised characteristics. These promises must be in writing to be valid and accompanied by the signature of the authorized representative of the LICENSER. The LICENSER is only liable for damage claims in accordance with Roman numeral V.

V. RESTRICTION OF LIABILITY

1. The LICENSER is not liable for damages unless the damage is caused by intent or gross negligence of the LICENSER.
2. Liability due to any characteristics which may have been promised by the LICENSER is not affected by this. Liability for subsequent damages due to deficiencies which were not included in the promises of the LICENSER are excluded.
3. No liability is assumed for calculable damages, in particular lost profit.
4. Any damage claims are limited to the amount of the damage whose possible occurrence the LICENSER logically had to expect under the circumstances known at that time when the contract was concluded. In any case, liability is limited to the amount of twice the purchase price that was paid (license fee), regardless of whether claims pertaining to contract law, damage claims or other liability claims are concerned.

VI. Validity and Conclusion

1. The license given to the LICENSEE is valid until it is terminated by the licensee or the LICENSER.
2. The LICENSEE can terminate the license at all times by returning the SOFTWARE PRODUCT including the archivation copy and all related documentation to the LICENSER.
If the return is not due to a warranty claim, the license fee will not be reimbursed.
3. No liability is assumed for calculable damages, in particular lost profit.
4. The LICENSEE terminates the license by resale to a THIRD PARTY in accordance with III, para. 1.

VII. Applicable Laws

1. Regarding the laws which may apply, this license is subject to the laws of the Federal Republic of Germany and the exclusive adjudication of the German courts of law.
2. Court of jurisdiction for all legal disputes resulting from the contractual relationship and its creation and effectiveness for general merchants is Nuremberg. However, the LICENSER has the right to take the LICENSEE to court at his location.
3. Place of execution for all obligations from this contractual relationship is Nuremberg.

4 Basics

This chapter covers the following topics:

- [NetCon echo](#)
- [PLC Connections](#)
- [Logger](#)
- [ARP](#)
- [PC-H1 Driver](#)

4.1 NetCon echo

This is the graphical user interface (GUI) used to configure and diagnose the echo devices. The GUI connects to the device via TCP/IP (port 982 or 997).

Starting the program

To start the program, proceed as follows:

- Select **Start - (All) Programs - Softing - NetCon echo - NetCon echo**.
The start window opens, allowing you to configure your echo device via TCP/IP or via H1 (provided the H1 protocol driver has been installed). Then, the [Select a Station in Your Network](#) window opens where you can establish a connection to the desired device. You can also perform the configuration **offline** and load the configuration data later into your echo device, but usually, the configuration is done online.

Password

To prevent parameters and settings from being changed, you can set a password. To do so, select **Station - Station Password**. If a password has been set, each user will be prompted for the password when trying to save or change parameters. Read-only access is not possible without entering the password.

If you forget the password, the only workaround is to transfer the parameters from the external SD card. In the root directory of the SD card, a file named `\netfile.net` is stored. This file contains either the known password or it is empty. Then, insert the card into the device and disconnect it from the power supply.

Then, switch the device on again. While the device starts up, its parameters are overwritten with the ones from the card. There is no other remedy.

Operation and Configuration

For more information on the NetCon graphical inter user interface, see the [Operation and Configuration](#) section.

Logger

In NetCon, you can configure the [Logger](#) feature for the device. The Logger content is displayed in the [Logger](#) window.

4.2 Logger

The Logger feature allows you to log the device's communication over the configured connection. You can choose to keep the log data only temporarily in the RAM or save it to a log file. The Logger records the selected data traffic of a communication going over a particular connection. Such a log can be very useful, especially for troubleshooting. Each event is recorded with a timestamp, the name of the application and the actual message.

In NetCon, the [Logger](#) window displays the entries that have been recorded.

To set the parts of the communication to be logged, open the [Logger Settings](#) dialog box.

To specify whether to create the log in the RAM only or to also save it to a file, click [Settings](#) in the Logger dialog box.

4.3 ARP

The **Address Resolution Protocol (ARP)** is a network protocol which allows mapping network addresses to hardware addresses.

To map an IP address to a MAC address, an ARP request is used. After encapsulating this request in a frame, it is sent out as a broadcast. All stations in the network receive and evaluate this request. They compare the IP address submitted in the ARP request with their own IP address. The station whose the IP address matches the one in the request replies by sending an ARP response.

The response is sent specifically to the station which had sent out the request, it contains the requested MAC address.

The MAC address is added to an ARP cache along with the associated IP address. The ARP cache is a temporary memory which is part of the respective station's RAM. Each time another IP address needs to be resolved to a MAC address, the station first checks whether it can do so using the ARP cache and thus without sending a new ARP request. If this is not possible, it sends an ARP request. Thus, ARP resolution is advantageous as it saves time and reduces the network load.

If an entry in the ARP cache is not accessed for a certain period of time, it will be removed. This behavior ensures that changes in the network can be detected and reflected by sending new ARP requests.

See also: ARP Cache Stati

4.4 PC-H1 Driver

To configure the echo devices via H1, you need the H1 driver (INAT H1 ISO protocol). This driver is installed with the software and is available for NetCon.

If the PC has multiple network adapters installed, the H1 driver should only be bound to the network adapter used for configuration.

A detailed description is included in the PC-H1 program folder.

5 PLC Connections

This chapter covers the following topics:

- [S7 – General Information](#)
- [S5 – General Information](#)
- [Modbus – General Information](#)
- [CLX – General Information](#)
- [PLC-5 / SLC – General Information](#)
- [MELSEC-Q – General Information](#)
- [Send / Receive – General Information](#)
- [OPCpipe – General Information](#)

Serial Connections:

- [AS511](#)
- [Telnet](#)
- [3964 \(R\)](#)
 - [RK 512](#)
- [MPI](#)
- [Modbus RTU](#)
- [DF1](#)

5.1 S7 – General Information

The S7 protocol is used to communicate with S7 PLCs and with CPs capable of using the S7 protocol. The S7 protocol can be used in conjunction with TCP/IP, H1, or MPI.

Read (Fetch active) and Write (Write active)

- When communicating with S7 PLCs, it is generally possible to both read and write data over a single connection.
- Usually, the parameterization of the S7 protocol is based on RFC 1006.
- Besides the S7 PLC address (IP address or MAC address), a read and write TSAP is required. When using MPI, enter the MPI address of the CPU.
- To read and write data from and to an S7 PLC (layer 7 communication), the following can be used:
 - non-parameterizable connections
 - parameterizable connections

Non-parameterizable connections

A connection will only be established on the echo side and NOT on the PLC side.

Default connections are required that are handled via standard TSAPs or default ports.

Only a limited number of default connections are available. If these are not sufficient, you need to use a "parameterizable connection".

Default connections with Siemens CPs are handled via RFC 1006 or H1 (see [Standard TSAPs](#) and [TSAP for S7 Connections](#)).

Parameterizable connections

The connection must be parameterized on both sides of the communication (one Fetch/Write active connection on the echo side AND one Fetch/Write passive connection in the S7 CP)

If you use echolink for the communication, parameterizable connections are required.

5.2 S5 – General Information

The S5 protocol is used to communicate with S5 PLCs (S5 AP headers) and with CPs capable of using the S5 protocol.

The S5 protocol can be used in conjunction with TCP/IP, H1 or for serial communication with AS511.

Read (Fetch active):

- To read data actively from an S5 PLC, you need to establish a read connection in the echo device.
- Besides the S5 PLC address (IP address or MAC address), a read port or a read TSAP must be specified.
- The read connection needs to be parameterized on both sides of the communication (Fetch active connection in the OPC server, Fetch passive connection in the S5).

Write (Write active):

- If you want to write data to the S5 PLC as well, a write connection needs to be established besides the read connection. When using S5 via TCP/IP or via AS511, a single connection may be used for this purpose.
- For the write connection, a write port or a write TSAP must be specified.
- The write connection needs to be parameterized on both sides of the communication (Write active connection in the echo device, Write / Receive passive connection in the S5).
- Newer S5 TCP/IP cards and echo devices support reading and writing over a single connection.

5.3 Modbus – General Information

The Modbus over TCP protocol is used for the communication with PLCs from Wago, Groupe Schneider, Beckhoff and all CPs capable of using the Modbus over TCP protocol (Modicon, Ethernet terminals from Phoenix, Wago, Beckhoff and many more).
The Modbus protocol only works in conjunction with TCP/IP.

Read (Fetch active) and Write (Write active):

- When communicating with Modbus PLCs, it is generally possible to both read and write data over a single connection.
- If default port 502 is used, you only need to parameterize the connection on the echo device side. On the PLC side, no other settings are required.

NOTE:

For the communication via Modbus over TCP, a specific [Modbus Item Syntax](#) is used.

5.4 CLX – General Information

The CLX protocol is used for the communication with ControlLogix / Compact Logix PLCs from Rockwell Automation.
The CLX protocol only works in conjunction with TCP/IP.

Read (Fetch active) and Write (Write active):

- When communicating with CLX PLCs, it is generally possible to both read and write data over a single connection.
- If default port 44818 is used, you only need to parameterize the connection on the echo device side. On the PLC side, no other settings are required.

NOTE:

To address item IDs, you need to specify their symbolic names in the CLX.

Alias browsing:

Once the CLX connection has been established, symbol information will be read from the PLC and displayed for selection in the corresponding dialog boxes.

5.5 PLC-5 / SLC – General Information

This protocol is used for the communication with PLC-5 and SLC PLCs from Rockwell Automation Allen-Bradley.
The PLC-5 / SLC protocol only works in conjunction with TCP/IP.
We do not support the SLC firmware version which uses the function codes A2 and AA. In

such cases a firmware update of the SLC is recommended.

Read (Fetch active) and Write (Write active):

- When communicating with SLC or PLC-5 PLCs, it is generally possible to both read and write over a single connection.
- If default port 2222 is used, you only need to parameterize the connection on the echo device side. On the PLC side, no other settings are required.

NOTE:

For PLC-5 and SLC communication, a specific [PLC-5 / SLC Item Syntax](#) is used.

5.6 MELSEC-Q – General Information

The MELSEC-Q protocol is used for the communication with Mitsubishi PLCs from the MELSEC-Q series.

The MELSEC-Q protocol only works in conjunction with TCP/IP.

Read (Fetch active) and Write (Write active):

- When communicating with MELSEC-Q PLCs, it is generally possible to both read and write data over a single connection.
- The connection needs to be parameterized on both sides of the communication (one connection in the echo device and one connection in the MELSEC-Q PLC).
- By default, port 8192 is used. The same port number must be used on both sides of the connection.
- If the MELSEC-Q PLC uses non-parameterizable connections with fixed ports (“default connections”), you only need to parameterize the connection on the echo device side. On the PLC side, no other settings are required. Please make sure that the echo device uses the same ports as the MELSEC-Q PLC.

NOTE:

In Mitsubishi PLCs, addresses are often given in HEX notation (in this case, you need to convert the port numbers to decimal notation for the echo device side).

NOTE:

For communication with Mitsubishi PLCs, a specific [Melsec-Q Item Syntax](#) is used.

5.7 Send/Receive – General Information

In a Send / Receive communication, the data is transmitted without application headers, i.e. as raw data. Any station that supports the Send / Receive direct interface can be used as the communication partner. The Send / Receive protocol works both with TCP/IP and H1.

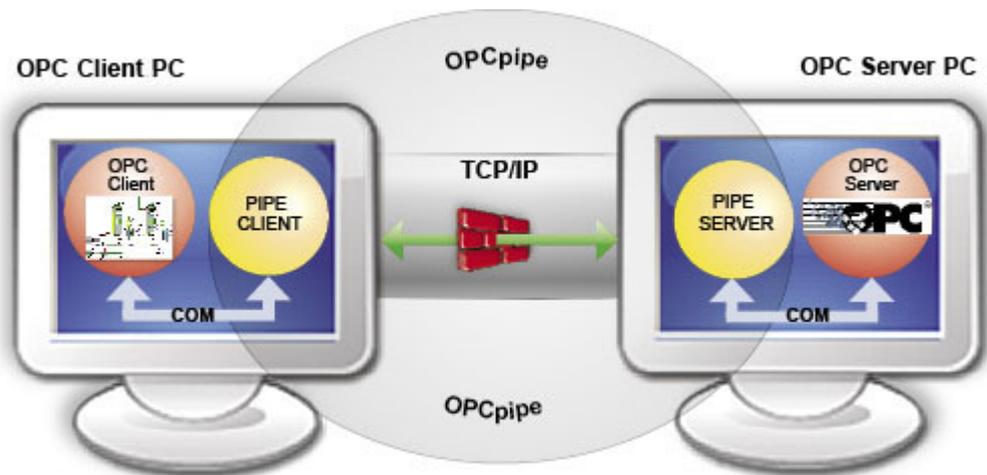
Receiving and sending data

- It is generally possible to both send and receive over a single connection.
- Besides the address of the Send / Receive PLC (IP address or MAC address), you need to specify a port or TSAP.

NOTE:

For Send / Receive communication, a specific [Send/Receive Item Syntax](#) and a specific [Extended Raw Data Item Syntax](#) are used..

5.8 OPCpipe – General Information



OPCpipe is a tunnel protocol that tunnels the data exchanged in an OPC communication between a client and a server.

OPCpipe comprises two parts:

OPCpipe server: Runs on the same computer as the OPC server or on the echo device.

OPCpipe client: Runs on the same computer as the OPC client.

The OPCpipe client receives the request from the OPC client and converts the OPC communication to a “normal” TCP/IP communication. The data can thus be transmitted in a tunnel to the destination computer or echo device via the network by using a standard TCP/IP connection. There, the OPCpipe server receives the data, “decrypts” it by converting it back to OPC communication and forwards the data to the OPC server or the protocol logic of the echo device. The server or echo device performs the requested action and sends the data back to the OPC client. The tunnel mechanism works the same way in both directions.

Establishing an OPCpipe connection

The OPCpipe connection must be configured both on the server side (computer or echo device) and the client side (computer where the OPC client is running).

For this purpose, the OPC server with OPCpipe functionality must be installed on both sides of the communication. Usually, the OPCpipe client initiates the active connection. The default port number is 4444.

5.9 AS511

AS511 is a transmission procedure from Siemens that couples programming devices (PGs) and operator panels (OP) to a Simatic S5 PLC.

- For this communication, the serial AS511 protocol is used.
- PG / OP communication is suitable for typical PG functionality such as loading of programs and configuration data, running test and diagnostic functions, and operating and monitoring an installation via OPs.
- Each S5 PLC has a PG interface and “knows” the AS511 protocol. The partner station of the S5 (a PG / OP) can actively read data and actively write data to the S5. This means that addresses need to be passed: the PG / OP indicates which data will be read from where and where data will be written to. In both cases, the S5 is passive. It responds to active requests by providing the data requested or allowing data to be written.
- Usually, the S5 itself is NOT capable of assuming an active role when communicating over the serial

AS511 protocol. It can neither request data actively nor send data actively.

5.10 Telnet

Telnet (Terminal over Network, Teletype Network) is the first service that was implemented on the Internet. With Telnet, you can access a computer connected to the network by running a terminal session (also referred to as remote login). The Telnet service uses TCP port 23 and is specified in RFC 854. With the standard remote login service, you can use computers in the network as if your own keyboard and terminal were connected directly to a remote computer, aside from longer response times in case of slow network connections. Today, all current operating systems have a Telnet client program. In most cases, this program features a simple text entry window. As soon as the connection has been established, all commands entered into this window are sent to the server and executed. All characters coming from the server are displayed in the client program window.

Using the TELNET server, you can establish a terminal connection over the network. This is helpful in cases where you cannot use the serial interface because the device is remote or the interface is used for other devices.

5.11 3964(R)

The Siemens procedures 3964 / 3964R are asynchronous, bit-serial transmission standards for point-to-point coupling, i.e. the data exchange between two automation devices (two CPUs) or between an automation device and another coupling partner over a serial connection. The protocols are associated with layer 2 (data link layer) of the ISO Reference Model.

3964 procedures allow the SEND and RECEIVE job types. On the sending side, the data to be sent must be part of a data block. The receiving side must indicate a data block to hold the received data. In order to use the 3964 / 3964R procedure, the communication partners need to be configured appropriately. These protocol parameters (baud rate, parity, bits per character, stop bits, mode (3964 or 3964R), priority, character delay time, acknowledgment delay time, connection attempts, number of send attempts) are either specified directly in the CPU application program or in the CP.

All these parameters must be identical for both partners. EXCEPTION: The priority (0 lower, 1 higher) MUST NOT be identical to allow for solving initialization conflicts.

5.11.1 RK 512

The Siemens RK 512 procedure is an asynchronous, bit-serial transmission standard for point-to-point coupling, i.e. the data exchange between two automation devices (two CPUs) or between an automation device and another coupling partner over a serial connection.

It is based on the 3964R protocol.

See also

- [3964\(R\)](#)

5.12 MPI

MPI = Multi-Point Interface

In the SIMATIC S7 PLC, the AS511 programming device interface of the SIMATIC S5 PLC has been replaced by the multi-point capable interface MPI (for S7-300 and S7-400). All CPUs in S7-300 and S7-400 PLCs as well as PGs and OPs have an MPI interface. The following devices can be connected to an MPI:

- Programming devices (PG / PC)
- Operating and monitoring devices (OPs and TDs)
- Other SIMATIC S7 PLCs

While the AS511 interface only allows point-to-point communication, MPI enables you to set up a bus. MPI

provides the following options:

- Programming via Ethernet. Your device routes this data to the corresponding COM port. All PLCs connected to the MPI bus are accessible. PG routing is also possible, i.e. programming across network boundaries.
- Fetch / Write communication
- Send / Receive communication

Structure of the MPI subnet

A segment is the bus line between two terminating resistors.

- A segment can include up to 32 devices.
- If more than 32 devices are to be included, the bus segments need to be coupled using RS485 repeaters.
- Each RS485 repeater used reduces the maximum number of other devices per bus segment. This means that if a bus segment contains an RS485 repeater, a maximum of 31 other devices may be included in the bus segment. The number of RS485 repeaters, however, does not affect the global maximum number of devices on the bus.
- The individual devices of the subnet are interconnected by bus connectors and the PROFIBUS bus cable.
- A bus line needs to be terminated using the correct impedance. For this purpose, the terminating resistor is enabled on the first and the last device of a subnet or segment by setting the terminating resistor to “On” at the bus connector.

Addressing MPI subnet devices

To allow intercommunication between the devices in an MPI subnet, they are assigned a unique address (MPI address). In addition, each station is informed of the “highest MPI address” in the MPI subnet. Valid MPI addresses are from 0 to 126. The following default addresses have been defined:

PG = 0
OP = 1
CPU = 2

- All MPI addresses in a subnet must be unique.
- Even if only a single CPU will be addressed via the programming software, the MPI address must be used to establish the communication.
- The highest MPI address must be \geq the largest actual MPI address. This setting must be the same for all devices (exception: a PG is connected to multiple devices).

5.13 Modbus RTU

The serial Modbus protocol is an open communication protocol based on a master/slave architecture. In the ISO/OSI Reference Model, it is associated with layer 2.

Please note that a server is not the same as a slave. The term “server” is related to layer 7, while the term “slave” is used in connection with layer 2. A device that basically has a slave functionality is not capable of initiating a sending transaction. A server, however, can autonomously send a message on the bus when a particular event occurs.

Modbus is used on different transmission media. It is very common to implement Modbus on the RS485 bus physics.

- Typical master devices are host computers and programming devices. Typical slaves are programmable controllers.
- In a master/slave system, the bus consists of an active device (master) and multiple passive devices (slaves).

- Communication is exclusively controlled by the master. As long as the master does not request the slaves to do so, they will not send any data; nor will they communicate with other slaves.

The master uses one of the following modes for its request:

1. Polling / Unicast:

The master addresses its request to a specific slave. After having received and processed the request, the slave sends a response to the master.

In this mode, the transaction consists of 2 messages: a request from the master and a response from the slave. If you want to address a slave, it needs a unique address.

2. Broadcast

The master sends a request to all stations in the network (address 0 is used as the broadcast address). Broadcast requests are invariably Write commands. All devices must accept broadcast requests. NO response is sent to the master. In this mode, the transaction consists of 1 message: a request from the master.

Addressing

In order to enable the master to send its frames to the corresponding slaves, they must be assigned unique addresses. No address needs to be configured for the master. This address will be transmitted in the address field of the Modbus frame.

Address	Meaning
0	Broadcast
1-247	Slave addresses
248-255	reserved

5.14 DF1

The DF1 protocol is the serial protocol for Rockwell PLCs using RS232.

The individual items are addressed through file numbers. In the Control Logix and Compact Logix systems, these file numbers are assigned to items via access tables.

6 Operation and Configuration

This chapter covers the following topics:

- [How to Proceed](#)
- [Basic Configuration](#)
- [Licensing](#)

6.1 How to Proceed

The devices are very easy to parameterize in just a few steps. To set up communication, proceed as follows:

Basic Configuration

To be able to communicate with the echo device via TCP/IP, the device needs an IP address. How to assign the address is described in the [Basic Configuration](#) section.

Configuring the connection

For every hardware from which you want to read data or to which you want to write data, you create a connection to the echo device by using the NetCon software. This connection defines the access path to the hardware, i.e. the channel between the two communication partners. All the parameters needed for the communication between the echo device and the hardware are defined in the respective dialog boxes. The protocols that are supported by the communication partner (e.g. S7 protocol and RFC 1006) are provided for selection. To access specific hardware, you specify the destination addresses (e.g. IP address and port numbers, or MAC address and TSAP). You can also choose whether or not writes are basically allowed over this connection.

See the [New Connection](#) section.

Logger

The [Logger](#) is provided for recording error events that have occurred in the device.

6.2 Basic Configuration

During the basic configuration, the device is provided with all the parameters needed for communication. The device is registered on the network.

- Start the NetCon configuration software.
- Select the [type of configuration](#) (usually IP).

NOTE:

The station you are configuring via H1 or IP has to be in the same network as the configuration computer. Stations connected behind a router cannot be discovered.

Configuration via H1 is only available if the INAT PC H1 protocol driver is installed.

Configuration via TCP/IP or H1

- The [Select a Station in Your Network](#) dialog box appears. The new echo device is displayed together with its MAC address.
- From the MAC address, you can determine the station to be configured. Double-click the station or select it and then click OK.
- The [Station Parameters](#) dialog box appears. Edit the settings and confirm your entries with OK.
- The configuration is complete.

NOTE:

If you want to use both network connections provided by the echo device, both configurations should be configured.

6.3 Licensing

The echo devices are delivered with the functionality that was specified in the order. If, during commissioning or later when adding extensions, you find that you need to expand the functionality, you can do this by activating a license.

In the [License Overview](#) dialog box, you can see which functionalities are licensed and which not. The NetCon displays a License Request Code that you can send to Softing Industrial Networks GmbH. You will then receive an activation code that enables the desired functionality. How this is done is described in the [License Overview](#) and [Licenses](#) sections.

Please note the [license conditions](#).

7 Windows

The program GUI basically consists of the following main windows:

- [Softing NetCon Configuration](#)
- [NetCon main window](#)
- [Connections list](#)
- [Logger](#)
- [Collect Frame Tables](#)

Other functions are available from dialog boxes described in the [Connections List](#), [Main Menu](#) and [Collect Frame Menu](#) chapters.

7.1 Softing NetCon Configuration

The configuration window opens when you start NetCon by selecting **Start - (All) Programs - Softing - NetCon echo - NetCon echo**.

NetCon is required to configure the echo devices. The software is used to make basic settings, create and diagnose connections and display data recorded by the Logger feature. To configure NetCon for a device, use one of the following methods:

Configuration via TCP/IP

A socket connection to the device is established via TCP/IP. Any entries you make are transmitted directly to the device and stored there. This method allows you to configure the device from anywhere within the TCP/IP network.

Configuration via H1

A configuration via H1 is only possible if the H1 protocol driver is installed. An H1 connection is established to the device you want to configure. Any entries you make are transmitted directly to the station and stored there. This method allows you to configure the station from anywhere within the H1 network.

Offline configuration

You can create a configuration file offline on the PC and transfer it to the device. This is not the common way of configuration. It is recommended to configure the device online until everything works properly. Then, transfer the parameters from the device to the PC and save them there.

Language selection (buttons)

Use the buttons to select the desired user interface language. After a restart, NetCon will display all texts in the selected language. Please note that this does not toggle the texts displayed in the Logger.

7.1.1 Select a Station in Your Network

When you perform a **configuration via TCP/IP** or a **configuration via H1** the **Select a Station in Your Network** window opens.

All stations with an arrow (->) at the beginning of the row are currently online in the network. They are detected automatically by NetCon. The software displays all echo devices and PCs with servers.

NOTE:

Stations connected behind a router cannot be discovered automatically. For these, you need to define a direct connection using the [New](#) button.

Selecting the desired station

When you double-click the desired station (or click the OK button), NetCon will connect to the corresponding station. The connection attempt has a timeout of about 3 seconds.

Successful connection: the [Connections](#) main window opens.

Unsuccessful connection: If there is no reply from the station within 3 seconds, the connection attempt aborts.

New

If the station is not listed in the automatic online display (parameterization via IP) because it is located outside your network, you can click this button to configure a direct connection to this station in the [Edit Access to a Station in Your List](#).

Edit

Using this button, you can change the parameters for the station you created using the [Edit Access to a Station in Your List](#) dialog box.

Delete

Using this button, you can delete the parameters for the station. A prompt asks you to confirm that you really want to delete the parameters. **Deletions cannot be undone.**

Scan for stations

Starts the station scan and the automatic discovery of online stations and servers. This scan will only find stations that are accessible in the local network.

Execute station scan on start

Automatically starts the station scan when you open the dialog box. Since this may take a long time if many stations are online, you can disable this function.

Use Timeout

It is recommended to enable this option only if the default connection timeout is too short or the transmission is very slow. By default, this function is disabled.

Timeout

You can use the connection timeout option to change the default timeout (3 seconds) specifically for this connection. This is useful e.g. for Internet connections for which heartbeat monitoring (sending of life time acks) has been disabled. This also increases the response time in case of connection problems (such as a broken cable).

7.1.2 Edit Access to a Station in Your List

If a connection to the desired station was not possible and the [Station Not Found](#) dialog box appears, you can configure a direct connection to this station there. The entry will then be added to the list of available stations and displayed in the [Select a Station in Your Network](#) dialog box. After the station scan has been performed, an arrow symbol (->) preceding a station indicates that this station is available online.

Station Name

The name to be displayed in the list.

H1

Establishes a connection via the H1 protocol. Specify the MAC address and the TSAP.

TCP/IP

Establishes a connection via the TCP/IP protocol. Specify the IP address and the port number.

Extended

Extends the dialog box so that you can enter additional parameters.

H1 Settings

MAC address

The MAC address of the station to be configured and diagnosed.

Use Default TSAP / Special TSAP

Here, you can indicate the TSAP for H1 or RFC 1006 connections.

TCP/IP Settings

IP Address

Here, you can indicate the IP address or the DNS name of the station to be configured and diagnosed. 127.0.0.1 addresses the local device.

Use Default Port / Special Port

The OPC server uses port **982**.

The echo devices use port **997**, newer ones use **982**.

If you want to configure and diagnose the station using RFC 1006, you need to enter the TSAP.

Standard

Reduces the dialog box showing only a few parameters.

Optional Product Select

These entries will be ignored.

7.1.3 Station Not Found

The dialog box shows the message **A timeout occurred**.

If a station cannot be found, this can be due to the following reasons:

OPC server

- The server service has stopped.

Configuration via TCP/IP or Configuration via H1

- The selected station is switched off.
- A network cable is not properly connected or faulty.
- A switch, hub, router or gateway is switched off.
- The station is connected behind a router and therefore not accessible.
- On the operating PC, the protocol to be used for connecting to the station (H1 or TCP/IP) has not been installed or configured properly.
- The H1 or TCP/IP protocol on the remote network station has not been installed or configured properly.
- A station (e.g. a reference router) has disabled the forwarding of the frames.
- The firewall has not been configured properly.

NOTE:

Stations connected behind a router cannot be discovered automatically. For these, you need to define a direct connection using the New button (see [Edit Access to a Station in Your List](#)).

Serial connection

- The connection cable is not plugged in.
- The pins of the connection cable are not assigned correctly. The connection to the station only uses three wires: Send data, receive data, signal ground.
- The interface adapter installed in your PC is faulty.
- The interface of your PC is used by another task.

7.1.4 Offline Configuration

In the **Choose a Product** dialog box, you can select the device to be configured offline. The dialog boxes and menus may vary depending on the functionality of the device.

In an offline parameterization of a device, the parameter data is stored in a parameter file on the PC. The data can later be transferred to the device.

- After clicking the **Configuration Offline** button, the **Open File...** window opens. Here, you can either open an existing parameter file or enter a new file name.
- After you have entered a new file name, the following dialog box opens: **Choose a Product**.
- Here, select the device to be configured.
- Selecting the device closes the dialog and enables you to make further entries.

7.2 NetCon Main Window

The main window consists of two parts. The left panel shows all online stations. The right panel shows the parameterized connections of the station selected in the left panel, i.e. the [Connections](#) list. When you select a station in the left panel, the right panel shows the associated connections.

Double-clicking a connection opens the dialog box where you can edit the connection.

Right-clicking opens a context menu that provides more functions.

The menu bar provides all functions required to parameterize the echo devices. The menus are described in the [Main Menu](#) and [Collect Frame Menu](#) chapters.

Frequently used functions are available on a toolbar where they can be accessed by a single click.

Press the **ESC** key to exit the window.

7.3 Connections List

The Connections list of the device is displayed in the main window. It shows an overview of the configured connections including their parameters. The connections are displayed in different colors to indicate their current status: active (**black**) or inactive (**gray**).

To edit a connection, double-click it. To do so, you need master access.

You can toggle the main window between the Connections list and Online Diagnostics by clicking the **Diagnosis - Show All Connections** menu item or the corresponding icon from the toolbar.

Clicking the table header sorts the table by this column in ascending order. Another click on the table header changes the sort order to descending.

The individual columns have the following meanings:

Column	Description
Type	Displays the transport protocol used: - TCP/IP - H1 - Iso TCP (RFC 1006) - Collect Type If the connection is inactive, its name is followed by (off) and grayed out.
Connection Name	The name of the connection.
Job	Shows the application protocol used.
NetProt	Shows the network protocol used and whether the connection has been established actively or passively: TCP/IP: Client [active] / Server [passive] H1: Act [active] / Pass [passive]
Dest. Address	Shows the address of the destination hardware (IP address, IP name or Ethernet address) which can be 0 if the connection has been established passively.
Parameters	This column displays some connection parameters: - Port number - Transport protocol TCP or UDP - Local TSAP , Dest TSAP for RFC 1006 and H1 connections
Parameter Details	Shows additional connection parameters and details, including: - OPCpipe access: Read and write: RW , read only: RO

7.4 Logger

A toolbar in the top part of the window provides the following functions:

Set Marker

Adds a marker text including the current timestamp to the list.
"-----"

Clear

Deletes the entries stored in the RAM. If you have selected to write the log data to a file, the RAM content will be saved to the file before deletion.

Settings

Opens the following dialog box: [Logger Settings](#)

Auto Scroll

Cyclically refreshes the window contents and automatically scrolls to the last line containing the most recent entry.

Suspend

Clicking this button suspends logging. This allows the user to diagnose the current content without overwriting older entries.

NOTE:

No more entries will be recorded.

Close

Closes the Logger dialog box. This does not affect the logging process. Logging continues even if the window is closed.

Master / Slave

Master / Slave shows the current mode of NetCon parameterization. In slave mode, it is not possible to make any settings in the Logger. The mode is selected automatically with the first NetCon instance always being the master; any other instances will run in Slave mode.

As a general rule, it is not recommended to access an echo device by more than one NetCon instance.

Operation

In this window, you can also select entries with the mouse or by pressing the space bar, and copy them to the clipboard. To access this function, master access is required.

Right-clicking then opens a **context menu** which provides the following functions:

Copy

The selected entries are copied to the clipboard and can be pasted into other programs.

Clear Logger

Deletes the entries stored in the RAM. If you have selected to write the log data to a file, the RAM content will be saved to the file before deletion.

Clear Selection

All selected entries are deselected.

To define which parts of the communication will be logged, open the [Logger Settings](#) dialog box. This function is only available in echocollect.

To specify whether to create the log in the RAM only or to also save it to a file, click the [Settings](#) button in the Logger window.

For more information on the logging function in the device, see the [Logger](#) section.

7.5 Collect Frame Tables

Collect frame tables describe the assignment of a PLC to another PLC and from a PLC to a database (e.g. SQL).

The window shows a list of the configured collect tables in the left panel and the graphical representation of the selected table in the right panel. This window helps you with creating and changing the connection parameters.

Edit Mode

The menu items under [Edit Mode](#) allow you to toggle between the graphical and the tabular representation.

Graphical representation	The graphical representation allows you to configure the assignment or the SQL request. It does not accept wrong entries. In the Collect Display Settings dialog box, you can configure the window.
Tabular representation	The tabular representation displays the complete frame including all syntax items, thus allowing you to review the frame. Use the buttons below the tables to edit, delete or move the syntax items.

Online Diagnostics

The **Online Diag - Online Diagnostics** menu item toggles the window to the online mode which displays the connection status and the item values. This allows you to check and monitor the functioning of the Collect frame table.

Operation

Each element can be manipulated by **right-clicking**. A context menu opens which provides the available functions.

Menu

The window has a specific menu which is described in more detail in the [Collect Frame Menu](#) chapter. For the description of frequently used functions, refer to the following sections:

- [Collect Frame Menu](#)
- [Collect Create Frame](#)
- [Add Element in Read Table](#)
- [Add Hidden Trigger](#)
- [PLC Confirm](#)
- [Trigger Settings](#)
- [Edit Normalisation Rule](#)
- [Memory Target](#)
- [Frame Chaining](#)
- [Mail Target](#)
- [Collect Indexed Text Table](#)
- [Collect Add Symbol Entry](#)
- [Collect Address Book](#)

Toolbar

A toolbar located below the menu provides frequently used functions.

How to couple multiple PLCs

As with databases, a read entry fetches data from the process. The data may be supplied by multiple PLCs. The collected data is transmitted as a frame to the destination PLC. Write entries fetch data from a PLC and distribute it to one or more PLCs. Complex data can be edited using chained tables which are logically linked.

8 Connection

This chapter describes the dialog boxes used to handle connections. Other dialog boxes are described in the [Windows](#), [Main Menu](#) and [Collect Frame Menu](#) chapters.

This chapter covers the following topics:

- [New Connection](#)
- [Connection Settings](#)
- [Edit Connection](#)
- [Copy Connection](#)
- [Delete Connection](#)
- [Disable Connection](#)

8.1 New Connection

New Connection

For each hardware system to be accessed (i.e. with which a communication needs to be established), you must create at least one connection in the device. This connection defines the access path to the hardware, i.e. the channel between the two communication partners.

In this dialog box, you can specify the following:

Connection Name

You can choose any connection name you like. It may contain only letters and numbers. The connection name you specify here can be selected in other dialog boxes.

Connection Type

Here, you can select the type of the connection to be created. This is a cascading dialog box. When you select an image or a button, other options will appear. When you select one of those options, another level may be displayed.

Logic Table Connection	Use this setting to select the destination or source connection of the echo device. In addition, you can select the type of protocol.
Database	Connection to a database. In the next dialog box, Database Connect Parameters , you can select the database to be used.
File	A local file will be created.
E-mail	Enables you to connect to the e-mail server. In the next dialog box, Mail Connect Parameters , you can select the desired connection parameters.
PLC/Raw	Connection to a PLC. It is possible to establish this connection as a raw data connection. Then, you can select the access path by clicking Ethernet or Serial . In the next dialog box, Network Protocol or Serial Protocol , you can specify the desired parameters.
Memory	The values are stored in a memory table in the device.
OPCpipe Client	Connection to another device or PC where an OPCpipe server connection has been configured. In the next dialog boxes, TCP/IP Connect Parameters and OPCpipe Parameters , you can specify the desired parameters.
Protocol Convert	Select this setting if you want to connect a device with a serial interface to an Ethernet network.
Ethernet - Serial/Bus	The connection will be used for protocol conversion. In the next dialog box, Serial Protocol , you can select the desired serial

	interface and the serial protocol to be used. Then, use the Network Protocol and TCP/IP Connect Parameters dialog boxes to select and configure the Ethernet connection.
Other Connection	Using this setting, the echo device could to offer connection to OPC UA clients and OPC servers.
OPC UA Server	OPC UA server allows OPC UA clients to access data for all connections which have symbolic item names defined.
MQTT Publisher	MQTT Publisher sends data to a MQTT Broker.
OPCpipe Server	OPCpipe connection that can read data from all connections where OPCpipe access is permitted.

8.1.1 New Logic Table Connection

In this dialog box, you can enter a new connection and its parameters.

This dialog box will be displayed for any type of connection. Depending on the connection type, different options are available.

The dialog box includes 4 groups.

Top group:

Connection Name

This field displays the connection name which can be changed here.

Connection Active

Shows whether the connection is active or not. If not, you cannot register items and there will be no connection to the PLC. This feature allows you to disable a connection temporarily without deleting it so that you can re-enable it later without having to enter all parameters again.

Write allowed

For some connections, you can disable the Write function.

OPCPipe Access allowed

Using this option, you can allow an OPCpipe server connection to access the connection configured here. Thus, it is possible to access the PLC from a remote PC via this connection.

OPCPipe Write allowed

Using this option, you can enable or disable Write access via OPCpipe.

Left group:

Here, you can specify the Ethernet parameters for the connection. These are the same as in the [TCP/IP Connect Parameters](#) or in the [H1 Connect Parameters](#).

Right group:

Here, you can enter the parameters for setting the speed. These are the same as in the [Collect Slave Config. Edit](#) dialog box.

Poll interval ... ms

Here, you can specify the poll rate the device will use for reading data from the PLC.

Lower group:

You can enter the protocol parameters for the connection in the left panel. They are the same as in the dialog boxes of the corresponding protocol settings.

The right panel shows a summary of these parameters.

8.1.2 New Logic Table Connection – Serial

In this dialog box, you can enter a new serial connection and its parameters.

This dialog box will be displayed for any type of connection. Depending on the connection type, different options are available.

The dialog box includes 4 groups.

Top group:

Connection Name

This field displays the connection name which can be changed here.

Connection Active

Shows whether the connection is active or not. If not, you cannot register items and there will be no connection to the PLC. This feature allows you to disable a connection temporarily without deleting it so that you can re-enable it later without having to enter all parameters again.

Write allowed

For some connections, you can disable the Write function.

OPCPipe Access allowed

Using this option, you can allow an OPCpipe server connection to access the connection configured here. Thus, it is possible to access the PLC from a remote PC via this connection.

OPCpipe Write allowed

Using this option, you can enable or disable Write access via OPCpipe.

Left group:

Here, you can enter the protocol-specific serial parameters for the connection. They are the same as in the serial connection dialog boxes.

Right group:

Here, you can enter the parameters for setting the speed. These are the same as in the [Collect Slave Config. Edit](#) dialog box.

Poll interval ... ms

Here, you can specify the poll rate the device will use for reading data from the PLC.

Lower group:

You can enter the serial line settings in the left pane. They are the same as in the [Serial Line Parameters](#) dialog box.

The right panel shows a summary of these parameters.

8.2 Connection Settings

This section describes all dialog boxes that open when you edit a connection. The number of dialog boxes varies depending on the echo device to be configured.

8.2.1 Network Protocol

Name of the Connection

Here, you can edit the connection name, if required. All characters except brackets may be used.

Expert mode

If the expert mode is disabled, the available settings that are typically not used, will be hidden. Only enable this mode if you want to make special settings. By default, this mode is off.

Transport Protocol

Here, you can select the transport protocol to be used when communicating via the Ethernet network.

Transport Protocol	
TCP/IP	The Transmission Control Protocol/Internet Protocol is capable of routing and is often used today for the communication with PLCs and field devices. For an RFC 1006 (ISO on TCP) communication, select TCP/IP.
H1	ISO H1 protocol to connect to Siemens S5 or S7 PLCs that do not support TCP/IP.

Application Protocol

Here, you can select the application protocol.

Application Protocol	PLC
Send/Receive	Used for the communication with other PLCs or devices that do not support any of the protocols listed below. Only payload data (raw data) will be transmitted, but no protocol information.
S5 AP	Communication with Siemens S5 CPs or echo devices.
S7	S7 protocol used to connect to Siemens S7-400, S7-300 or S7-200 PLCs or echo devices.
Modbus TCP	Communication with PLCs or devices supporting the Modbus TCP protocol, such as Wago, Beckhoff, Groupe Schneider, Modicon ...
CLX - Ethernet/IP	Communication with Allen-Bradley ControlLogiX, CompactLogiX, SoftLogiX
Rockwell Slc / Plc5	Communication with Allen-Bradley PLC-5 and SLC
Melsec-Q	Communication with Mitsubishi Melsec-Q

OK

Click OK to apply your selection.

8.2.2 Network PLC Protocol

Name of the Connection

Here, you can edit the connection name, if required. All characters except brackets may be used.

Plc Protocol

Here, you can select the PLC application protocol.

Plc Protocol	PLC
Send/Receive	Used for the communication with other PLCs or devices that do not

	support any of the protocols listed below. Only payload data (raw data) will be transmitted, but no protocol information.
S5 AP	Communication with Siemens S5 CPs or echo devices.
S7	S7 protocol used to connect to Siemens S7-400, S7-300 or S7-200 PLCs, or echo devices
Modbus TCP	Communication with PLCs or devices supporting the Modbus TCP protocol, such as Wago, Beckhoff, Groupe Schneider, Modicon ...
CLX - Ethernet/IP	Communication with Allen-Bradley ControlLogiX, CompactLogiX, SoftLogiX
Rockwell Slc / Plc5	Communication with Allen-Bradley PLC-5 and SLC
Melsec-Q	Communication with Mitsubishi Melsec-Q

OK

Click OK to apply your selection.

8.2.3 TCP/IP Connect Parameters

In this dialog box, you can edit the TCP/IP parameters.

Name of the Connection

You can choose any connection name you like. All characters except brackets may be used.

Destination IP Address (or name)

Here, you can specify the IP address of the destination station (IP address of the PLC) or the DNS name of the destination station. If the device was configured for DNS and a DNS server is available in the network, you can also enter the symbolic name of the destination station.

For information on the structure of IP addresses, please refer to the [IP Address](#) section.

Port

Port numbers are addresses that are used on the transport layer in order to address applications. Port numbers are required for TCP connections. Ports are similar to the TSAPs used for RFC 1006 and H1 connections. Each port number is a 16-bit number in the range from 1 to 65535. To establish a connection, please note the following:

NOTE:

You can only establish a connection if the port number is identical on both sides.

[More detailed information on ports](#)

Type:

Here, you can specify whether your own station will actively initiate the connection attempt or will wait passively for the destination station to establish the connection. Please make sure that different values are selected on both sides of the connection.

Client (Active)	The station will actively initiate the connection attempt.
Server (Passive)	The station will wait for the destination station to establish the connection.

Protocol

The TCP and UDP protocols are available for selection. TCP is a secured protocol.

PLC Header

Enable this option for the communication with echo devices.

For details on the structure of the PLC header, refer to the [PLC Header](#) section.

Life Data Acks

If you enable this option, payload frames without content (only headers) are transmitted to keep a

connection alive that is not used cyclically (heartbeat monitoring). Both communication partners must support this feature. For the OPCpipe communication, it is recommended to enable this option.

RFC1006

If you enable the RFC1006 option, H1 frames will be "wrapped" in a TCP/IP frame for transport. To configure the TSAPs, click the **RFC1006 TSAPs** button.

Own TSAP, Dest TSAP

TSAPs (Transport Service Access Points) are addresses that are used on the transport layer in order to address applications. TSAPs are required for RFC 1006 connections. To connect, enter the local TSAP into the Own TSAP field and the TSAP of the communication partner into the Dest TSAP field. In this context, please note the following:

NOTE:

In order to be able to establish the connection, the value in the Own TSAP field of one system must match the Dest TSAP value in the other system, i.e. they must match crosswise.

[More detailed information on TSAPs
TSAPs for S7 Connections](#)

8.2.3.1 TCP/IP Special Settings

Use PLC Header

The PLC header can be used for communication with other Softing devices where the PLC header can also be enabled.

See also [PLC Header](#).

Life Data Acks

If you enable this option, payload frames without content are transmitted to keep a connection alive that is not used cyclically (heartbeat monitoring). Both communication partners must support this feature. For the communication with S7 PLCs, we recommend to disable this option.

Life Acks as TCP/IP Standard

By enabling this option, you can activate connection monitoring, a function that is poorly supported in many socket libraries. Life acks are frames that pass the connection status. It is recommended to enable this setting (default setting). If a WAN connection is used, you might want to disable this function for cost reasons.

Ignore TCP End Check

This setting only makes sense for Receive Direct connections.

Option enabled:

Reading in a TCP/IP frame disables its end tag. Thus, the reception buffer passes exactly the amount of data the PLC user program had requested. When data is read cyclically, this may cause problems if different data lengths were configured for the "Send Direct" and "Receive Direct" (user program) jobs.

Option disabled:

(default setting, recommended)

Excess data bytes will be ignored. This setting is only relevant to the currently edited connection (Receive Direct connection).

Life Acks with Previous Received Data

Same function as for "Life Acks as TCP/IP Standard"

If you enable this option, the data byte received last will be sent back, but with a wrong sequence number so that the other communication partner replies by sending a life ack with the correct sequence number.

Send an Ack Immediately after Received Data

It is recommended to enable this parameter if data is transmitted cyclically in a LAN. In most cases, the TCP/IP protocol collects multiple data blocks before sending an acknowledgment. If you enable this setting,

the acknowledgment (ACK) is sent immediately. This increases the data rate when data blocks are sent cyclically in short intervals.

Send a Life Data Ack after the last Frame in a Sequence

This option is only available if PLC Header or RFC 1006 is used for communication.

Many socket implementations (socket libraries) do not send the IP acknowledgment frame. If you enable this function, a life data ack will be returned as an acknowledgment.

Do not Wait for Send Acknowledge

This option is only available if PLC Header or RFC 1006 is used for communication.

Option enabled:

The sending station initiates frames without waiting for the frame acknowledgment (ACK) relating to the previous frame.

Option disabled:

(default setting)

The sending station waits for the frame acknowledgment (ACK) relating to the previous frame before sending a new frame.

End Connection with FIN Instead of RST

Option enabled:

As a reply to the end connection frame (FIN), a corresponding end connection frame is sent.

Option disabled:

When a station receives an end connection frame, it resets the connection.

Use the same Port Number for both Ports

This setting sets both the source and the destination ports to the port selected in the parameterization. This disables the automatic setting of the source port to a value > 1024 in some devices (see also [TCP/IP System Settings](#)). This setting is required for the communication with CPs that do not correctly handle frame traffic via UDP.

Big Endian Format in the PLC Header

In the PLC header, the sequence number will be transmitted in the MOTOROLA format (big-endian). Usually (flag not set) the INTEL format (little-endian) is used to transmit the sequence number in the PLC header.

Change Connection Timeout

Using this setting, you can specifically change the timeout setting for connection monitoring that will end a connection when parameterizing a station remotely. This makes sense in cases where you disabled the transmission of life data acks for an Internet connection, for example. Please note that increasing this value will also increase the response time in case of connection problems (such as a broken cable).

Change Connection Memory

This setting allows you to change the memory size reserved for a connection. The minimum memory size for a connection is 1460 bytes (maximum number of payload for Ethernet connections). For a broadcast Receive connection (UDP), this setting might not be sufficient. If the partner station sends data faster than the PLC can accept it, the data will be buffered in this memory. The UDP data will only be discarded if this memory overflows.

8.2.3.2 IP Address

Basics

To establish the communication between two technical devices, each device must be capable of sending data to the other device. This data can only be received by the intended remote station if it has been addressed properly. In IP networks, this is ensured by specifying an IP address.

An OPC server is able to address a PLC directly by its IP address. Example: 192.168.1.20

It is also possible to address a PLC by its name. For this purpose, specify the name of the PLC and the corresponding IP address and enter the domain server in the TCP/IP settings of the server. For a domain name (e.g. "PLC1"), the server queries the name server to get the IP address and then addresses the PLC

directly using its IP address (192.168.1.20).

IP Address

Each IP data packet starts with an IP header. This is an information section used for transmission on the IP layer. This header also includes two fields where the IP addresses of both the sender and the receiver are entered before transmitting the data packet. Routing is done on layer 3 of the OSI model, the network layer.

Structure

IPV4

The IPv4 addresses that have been used predominantly since the introduction of Internet Protocol Version 4 consist of 32 bits, i.e. 4 octets (bytes). This means that it is possible to represent $2^{32} = 4,294,967,296$ addresses. In dotted decimal notation, the four octets are written as four integers, from 0 to 255, in decimal format, separated by dots.

Example: 192.168.1.20.

IPV6 – new version with a bigger address space

Due to the rapidly increasing IP address demand, it was foreseeable that the usable address space provided by IPv4 would soon be exhausted. The IANA address pool was depleted on February 03, 2011. This was the main reason for developing the IPv6 protocol. IPv6 uses 128 bits to store addresses. This means that $2^{128} = 256^{16}$ ($= 340,282,366,920,938,463,374,607,431,768,211,456 \approx 3.4 \cdot 10^{38}$) addresses can be represented. This number is sufficient to provide each square millimeter of the surface of the earth with at least $665,570,793,348,866,944 (= 6.65 \cdot 10^{17})$ IP addresses.

Since a decimal representation with ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd would neither be very transparent nor handy, IPv6 addresses are usually represented in hexadecimal format. To further simplify this representation, every two address octets are grouped and separated by colons. XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX

Example: 2001:0db8:85a3:0000:0000:8a2e:0370:7344

To further shorten the string, leading zeros in a block can be omitted. A sequence of blocks consisting only of zeros can be replaced with :: but only once in each IPV6 address.

Example: 2001:db8:85a3::8a2e:370:7344

For IPV6, usually names are specified instead of the IP addresses.

8.2.3.3 Port

Port numbers are addresses that are used on the transport layer in order to address applications. Port numbers are required for TCP / UDP connections.

The port number is a 16 bit number from 0 to 65535.

Certain applications use fixed port numbers that have been assigned by the IANA and that are generally known. Usually, they are between 0 and 1023. They are referred to as well-known ports. The registered ports are between port 1024 and port 49151. Application providers may register ports for their proprietary protocols.

The remaining ports (49152 through 65535) are referred to as dynamic and/or *private ports*. They can be used variably because they are not registered and do not belong to any application.

8.2.3.4 TSAP

TSAP = Transport **S**ervice **A**ccess **P**oint

On the transport layer, ISO (H1) uses so-called TSAPs to address applications. These connection endpoints are required both for ISO (H1) and RFC 1006 connections.

Parameters for ISO (H1) connections		Parameters for RFC 1006 connections	
Station A	Station B	Station A	Station B
Own TSAP A	Own TSAP B	Own TSAP A	Own TSAP B
Dest TSAP B	Dest TSAP A	Dest TSAP B	Dest TSAP A
MAC Address A	MAC Address B	IP Address A	IP Address B

In order to establish the connection, the value in the Own TSAP field of communication partner A must match the value in the Dest TSAP field of communication partner B, i.e. they must match crosswise. Accordingly, the Dest TSAP field of station A must match the Own TSAP field of station B.

Own TSAP (station A) = Dest TSAP (station B)
Dest TSAP (station A) = Own TSAP (station B)

This condition can easily be met if you set the same value for the Own TSAP and Dest TSAP fields. These values may be identical.
If multiple connections are to be established between 2 stations, the TSAPs for the individual connections must be different.
To exactly specify a connection, the combination of TSAP + MAC address (or IP address) must be unique.

Rules for entering TSAPs for S7 connections

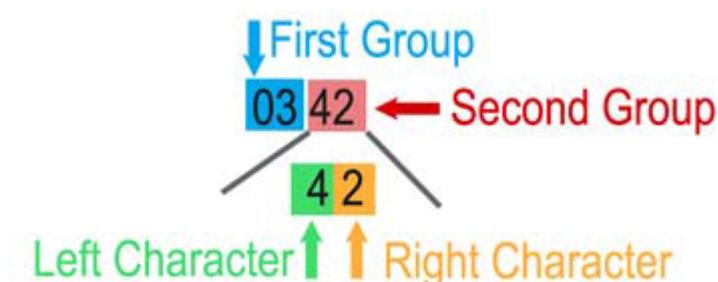
- TSAPs have a minimum length of 2 bytes and a maximum length of 8 bytes
- TSAPs can be entered as hex or ASCII characters.
- The TSAPs for S7 connections have a special meaning. See [TSAP for S7 Connections](#).

Rules for entering TSAPs for S5 connections

- TSAPs have a minimum length of 2 bytes and a maximum length of 8 bytes for TCP/IP, and 16 bytes for H1.
- TSAPs can be entered as hex or ASCII characters.
- The TSAPs for S5 connections do not have a special meaning. They are selected arbitrarily.

8.2.3.5 TSAP for S7 Connections

For non-parameterizable connections (= default connections), so-called standard TSAPs are used. For these, the following rules apply:



First group

contains device IDs for which resources are provided in the S7:

- 01: PG or PC
- 02: OS (operating or monitoring device)
- 03: Others, such as OPC server, Simatic S7 PLC...

Second group

contains the addresses of these components

Left character (bits 7...4):

Rack number multiplied by 2

Right character (bits 3...0):

CPU slot (< 16). S7-300 always uses slot 2

The standard TSAPs **MUST** be used on the PLC side (Dest TSAP field of the echo device). The local TSAP of the echo device (Own TSAP field) may be selected freely, but should have the same format. We recommend to use 01 01 in the Own TSAP field.

Examples:

03 02 Communication with the S7 CPU in rack 0, slot 2

03 43 Communication with the S7 CPU in rack 2, slot 3

03 2E Communication with the S7 CPU in rack 1, slot 14

NOTE:

To enter the binary standard TSAPs, use the HEX field (left field).

S7-200

The S7-200 is usually addressed with the TSAP 10 00 (binary). The own TSAP is set to 10 00 by default.

S7-300, S7-400

The S7-300 and 400 is usually addressed with the TSAP 03 02 (binary). The own TSAP is set to 01 01 by default.

S7-1200, S7-1500

The S7-1200 and 1500 is usually addressed with the TSAP 02 01 (binary). The own TSAP is set to 01 01 by default.

8.2.3.6 PLC Header

The data-stream oriented TCP/IP protocol is capable of grouping multiple short data units into a larger unit. This increases the data throughput of the network. However, it requires a header for the protocol above TCP. This corresponds to the procedure common in other protocols (FTP, HTTP).

For this purpose, the 8-byte INAT PLC header was developed.

Only use the PLC header if the communication partner supports it. Otherwise, the connection will fail!

PLC Header Format

Byte no.	Meaning
Byte 0	0x4d 'M'
Byte 1	0x4b 'K'
Byte 2***	Number of payload bytes following the header (LSB*).
Byte 3***	Number of payload bytes following the header (MSB**).
Byte 4***	Bit 0 = 1, if other frames follow
Byte 5***	0

Byte 6***	SeqNo. LSB*
Byte 7***	SeqNo. MSB**
Datalen in bytes	Payload

*LSB: Least (Low er) significant byte

**MSB: Most significant byte

*** Bytes 2 / 3, bytes 4 / 5 and bytes 6 / 7 together form the "short" data value. They are represented in the INTEL format

Acknowledges

If DataLen equals 0, the frame does not contain payload, but a life data ack. Data acknowledgments allow connection monitoring, a feature that TCP/IP, as a wide area protocol, does not include inherently. The default times for connection monitoring are the same as for the H1 protocol. This makes the S5 TCP/IP system H1-compatible from the PLC or PC perspective.

Sequence Numbers

Bytes 6 and 7 contain a sequence number that is 0 when establishing the connection and will be incremented by 1 each time payload is sent. This frame counter additionally secures the data transmission. If live data acks are sent, the sequence numbers are not incremented and Datalen is 0.

Fetch and Write Connections

For Fetch and Write jobs, the first 16 data bytes at the start of a job correspond to the SINEC AP header. The SINEC AP header is also used for H1 communication.

Sending / Receiving Data

When sending data over the S5 TCP/IP, a frame can include a maximum payload of 512 bytes. This maximum value is preset by the default tile block size. Received data packets can contain up to 1460 bytes. These limits are monitored automatically by the TCP/IP protocol so that no monitoring is required on the user side.

Transmission without Frame Header

The header at the beginning of the frame may be disabled. In this case, the application programs on both sides are responsible for monitoring. In this context, please note the following:

1. In particular with Send Direct and Receive Direct jobs, certain time limits until frame receipt must not be exceeded. If these time limit were ignored, the internal buffers would be full, e.g. due to requests. Thus, it would be impossible to synchronize request and response.
2. A certain blocked data transmission mechanism must be used so that it is possible to recognize the end of payload.
3. On the recipient side, you need to make sure that the frames from the reception buffer have been read before the partner station sends the next frame.

It is mandatory to set up connection monitoring in the application program.

8.2.4 H1 Connect Parameters

In this dialog box, you can edit the H1 parameters.

Name of the Connection

You can choose any connection name you like. All characters except brackets may be used.

Adapter

Here, you can specify the number of the Ethernet adapter used to establish the H1 connection.

Destination MAC

Each Ethernet station is addressed using a unique MAC address (also referred to as Ethernet address or hardware address). Enter the Ethernet address of the destination station here. The destination address consists of 6 bytes (to be entered in hexadecimal notation).

NOTE:

In a single network, each station must have a unique Ethernet address.

Own TSAP, Dest TSAP

TSAPs (Transport Service Access Points) are addresses that are used on the transport layer in order to address applications. To connect, enter the local TSAP into the Own TSAP field and the TSAP of the communication partner into the Dest TSAP field. In this context, please note the following:

NOTE:

In order to be able to establish the connection, the value in the Own TSAP field of one system must match the Dest TSAP value in the other system, i.e. they must match crosswise.

[More detailed information on TSAPs
TSAPs for S7 Connections](#)

CR Parameters

With ISO (H1) connections, a transport connection is established. For this purpose, an active transport instance sends a CR TPDU (Connection Request Transport Protocol Data Unit) signaling that it wants to connect. This CR TPDU is used to send some parameters to the partner, such as the desired TPDU size, the TPDU format, and others. Since there is no standard CR parameter definition, please refer to the operating instructions of the destination system to find out which parameters you need to specify here, if any. If no information is available, do not enter any CR parameters.

Type (active / passive)

Here, you can specify whether your own station will actively initiate the connection attempt or wait passively for the partner station to establish the connection. Please make sure that different values are selected on both sides of the connection. Usually, the PC will be parameterized as active. This is the default.

Protocol

Using this option, you can specify whether frames on this connection will go to all devices (Broadcast), whether a certain group of stations should be addressed (Multicast), whether a secured connection will be used (Normal) or whether the data will be transmitted via an unsecured connection (Datagram). Usually, you select "Normal" here.

Priority

The line priority can range from 0 (highest priority) to 4 (lowest priority). 0 and 1 are so-called express priorities, 2 and 3 are normal priorities. Priority 4 is only used on rare occasions because it causes the connection to be reestablished for each send transaction. On the other hand, if it is only used intermittently, it puts less load on the network than the other priorities because the line will not be monitored (the connection is closed after each send transaction). Please note that when using express priorities, the transmission will not be faster than with normal priorities. For some PLCs, however, the data will be transferred to the RAM using an interrupt if you select priority 0. This may result in a faster overall data transmission. For priorities 0 and 1, the maximum data length is 16 bytes. Usually, Prio 3 is used here.

8.2.5 Collect Destination Connection Config.

Connection Name

Here, you can edit the connection name, if required.

Drop first value after slave connection reestablishing

Enable this option if a first change of the trigger status should be ignored after the connection has been established, for instance in case of a trigger on value change. When a connection is established, the value changes from Fail to Valid. This change can thus be ignored.

When to store frames

Don't store frames	
Store frames for resending after network failure	The data will only be buffered if the network connection to the database is faulty. This requires a writeable sd-card with enough storage space inserted into the device.
Store frames for logging	The data will always be saved. If you select this option, another

	dialog box opens where you can enter the logging file settings and the server connection. This determines how the data will be transmitted to a server. This requires a writeable sd-card with enough storage space inserted into the device.
--	---

Logging File Settings

Filename	You can choose any file name you like. If you do not specify an extension, ".dat" will be appended.
File Name Example	Shows the path and file name in the echo device.
Max disc space used (mb)	Limits the memory space for log files in the echo device.
Create subdir	Creates a subfolder using the file name.
Write interval	Specifies the interval for writing to the file. Caution: the number of writes to the integrated FLASH is limited. For this reason, set the value as high as possible.
When to create a new file	This option specifies that after a certain time, a new file will be created. This enables the next option.
How to number files	A number or the creation date is appended to the file name.

File Upload

File Upload	This option enables the transmission of files to a server using a server connection. Otherwise, they remain on the echo device.
Connection	Here, you can select a server connection.
Add	Here, you can add, edit or delete a server connection.
Upload Path	This path will be appended to the server connection path.
Delete file after upload	The local file will be sent to the server and then deleted locally.
Append to File (after append on server the local file will be deleted)	The local file is appended to the file on the server and then deleted locally.
File Name Example	Shows the path and file name on the server.

8.2.5.1 Server Connections

All functions that copy files to the network server will use these connections. Examples: CSV files in echocollect.

This dialog box shows the list of existing connections.

New Entry

Click this button to add a new entry.

Delete Entry

This button deletes the entry selected in the list once you confirm the corresponding prompt.

Edit Entry

Using this button, you can edit the selected entry. The [File Upload Connection Edit](#) dialog box opens.

8.2.5.2 File Upload Connection Edit

All functions that copy files to the network server use connections configured in this dialog box. Examples: CSV files in echocollect. To create a file server connection, specify the login data for the file server here.

8.2.6 S7 Protocol Settings

Depending on the PLC family used, select **S7 1200/1500**, **S7 300/400** or **S7 200**. Changing the PLC family updates the TSAP settings with default values, as described in [TSAP for S7 Connections](#).

The Use Short TPDU setting is required for some S7-compatible PLCs. Only use this option if the data exchange with the PLC does not work properly. This option may slow the communication.

8.2.7 echocollect Connection: Modbus Protocol

In this dialog box, you can edit the specific parameters for the connection to Modbus PLCs.

	Description	Default
Name of the Connection	This field displays the connection name which can be changed here.	
Slave address	The slave address entered here will be transmitted in the Modbus frame.	1
Start Address 0	The starting address in the frame will be transmitted as specified.	Yes
Start Address 1	Some Modbus devices expect a starting address in the frame that is 1 less than the one you entered. The first address is 1.	No
Use Byte Swap	Swaps the bytes within a word	Off
Use Word Swap	Swaps the words within a double word (32 bits)	Off
Use Write Single Coil (05)	For writes, the Modbus opcode "Write Single Coil 05" is used instead of "Write Multiple Coils 15".	Off
Use Write Single Register (06)	For writes, the Modbus opcode "Write Single Register 06" is used instead of "Write Multiple Registers 16".	Off
User Item Unit ID	The slave address will be indicated in the item syntax. Example: Id2.R2 For more details, see Modbus Item Syntax	1

8.2.8 echochange Connection: CLX Protocol

In this dialog box, you can edit the specific parameters for a connection to Allen-Bradley ControlLogiX PLCs.

Name of the Connection

Here, you can edit the connection name.

CPU Slot

Here, you can specify the ControlLogiX CPU slot.

8.2.9 echochange Connection: SLC / PLC-5 Protocol

In this dialog box, you can edit the specific parameters for a connection to Allen-Bradley PLC-5 and SLC PLCs.

Name of the Connection

Here, you can edit the connection name.

CPU Slot

Here, you can specify the PLC-5 or SLC CPU slot.

8.2.10 Melsec-Q Protocol

In this dialog box, you can edit the specific parameters for a connection to Mitsubishi Melsec-Q PLCs.

You need to set these parameters if a Mitsubishi Net has been set up and the hardware supports it.

	Description	Hex	Decimal
Destination Module	Network address	0x3FF	1023
Monitor Timer	Monitors the time within which the response should be received.	0x100	256
Network Number	Network ID in the hardware configuration of the programming software	0	0
PLC Number	Another intermediate PLC number	0xFF	255
PC Number	Only relevant if a permission management system has been set up, allowing or denying PC access.	0xFF	255
Destination Station	Module address of another module in a PLC.	0	0

The effect of these values depends on the hardware used.

8.2.11 OPCpipe Parameters

Name of the Connection

Each connection must have a unique name that allows you to quickly identify it later in the Connections list.

For OPCpipe, some **security settings** can be specified.

Password

When you establish a connection, the password is used for access control. It is encrypted during transmission over the network.

Connection Active

You can disable a connection if it is not intended for communication with a PLC. Thus, it is possible to disable it temporarily without having to delete and redefine it.

Write allowed

If this option is disabled, this connection is read-only.

Memory limit

Each OPCpipe frame can have a size of up to 4 gigabytes. Usually, frames of that size are not required. Since OPCpipe must provide the memory for such frames, you can restrict this value in order to prevent reduced performance. This value should be selected in a way that it will not cause a frame of maximum length to be truncated.

If at runtime, this limit is exceeded, the data will be discarded and the communication partner will receive a corresponding message.

Send timeout

If the acknowledgment mode is used, this is the time within which the acknowledgment must arrive; otherwise an error is generated.

Use Send Acknowledge

In OPCpipe, you can use the acknowledgment mode that sends an acknowledgment to the sending station when data has been received.

This mode slows down communication and should only be used in cases where you need to make sure that all data has been received.

The mode should only be used if the destination station supports the acknowledgment mode as well, notifying the partner that a secured transmission has taken place.

8.2.12 Collect Memory Target

If you use a memory table as the destination, you can enter the corresponding item name in the **Item** box or click the **>> button** to open the [Collect Browsing](#) window where you can insert the desired name after browsing to it.

8.2.13 OPC UA Server Settings

Server Endpoint

The echo collect device contains a UA discovery server. The string specified as "Server Endpoint" is the string delivered to UA clients requesting a discovery

Server Namespace

When started up, the UA server in the echo provides multiple namespaces. This info field displays the namespace containing the PLC items.

OPC UA Server Authentication Settings

You may choose the authentication settings of the OPC UA server within the device from the following policies:

- **Anonymous:** Each client may access the server.
- **User name and password:** Clients which are authenticated by valid user name and password may access the data of the server.
- **Certificate:** Clients which are authenticated by a trusted certificate may access data of the server; see [OPC UA Certifikates](#).

OPC UA Server Security Settings

You may choose which security settings of the OPC UA secure channel are offered by the OPC UA server within the EchoCollect to OPC UA clients.

The following message security modes could be activated:

- None
- Sign: Messages are digitally signed to protect against manipulation.
- Sign and Encrypt: Messages are digitally signed to protect against manipulation and encrypted.

The following message security/encryption policies could be activated:

- None: To be used if no message security is required.
- Basic128Rsa15: To be used if medium message security is required. The certificate of the client needs to be trusted, see [OPC UA Certifikates](#).
- Basic256: To be used for high message security. The certificate of the client needs to be trusted, see [OPC UA Certifikates](#).

On session establishment the OPC UA clients choose one of your preselected settings. So if you enable all possible settings, the client may choose from the following matrix:

	None	Sign	Sign & Encrypt
None	Available	Not available	Not available
Basic128Rsa15	Not available	Available	Available
Basic256	Not available	Available	Available

8.2.13.1 OPC UA Certificates

For a client certificate to become trusted (from the EchoCollect point of view) the following conditions need to be full filled:

1. The certificate is digitally signed and the whole chain of certificates used for signing is available to the EchoCollect. Either stored in the CA or the certs folder (see below).
2. A certificate which has digitally signed the client certificate is stored in the certs folder within the EchoCollect. So self-signed certificates need to be stored in the certs folder to become trusted. And all certificates which are signed by a certificate stored in the certs folder become trusted; unless the certificates are known to be revoked.
3. The finger print of the client certificate is NOT stored in any certificate revokation list in the crl folder of the EchoCollect.
4. For user authentication it is additionally checked. that the certificate is not stored within the rejected folder within the EchoCollect.

Certificate Management

To ease the task of certificate administration, the OPC UA server within the EchoCollect stores each new client certificate in the newclientcerts subfolder in the binary .der format. The base part of the filename equals the SHA1 hash of the certificate.

So the recommended way to maintain client certificates within the EchoCollect is as follows:

1. Try to connect with each client which should become a trusted client to the OPC UA server within the EchoCollect. (This first connection may fail, depending on current authentication settings and missing certificates.)
2. Insert a writeable SD-card into the EchoCollect.
3. Use the menu entry "Store configuration to SD-card" of the NetCon.
4. After the NetCon indicated, that the configuration has been successfully written to the SD-card, the SD-card contains the following subfolders:
 - OPC_UA_PKI/certs
The subfolder contains all trusted client certificates in binary .der format. It also contains the server certificate in human readable .pem format.
 - OPC_UA_PKI/CA
The subfolder contains the certificates of Certificate Authorities, which are needed to verify the digitally signatures of not self-signed certificates.
 - OPC_UA_PKI/crl
The subolder contains the certificate revokation lists of the certificate authorities, which are used to invalidate digital signatures.
 - OPC_UA_PKI/newclientcerts
Any new client certificate is stored into this subfolder by the EchoCollect.
 - OPC_UA_PKI/rejected
The subfolder contains blacklisted certificates. Clients which use this certificates could not access the server, even if the certificate which has signed them is still in the trusted certs subfolder.
5. Insert the SD-card into a card-read, connected to your PC.
6. Move the client certificates which you want to be trusted into the OPC_UA_PKI/certs folder. Compare the SHA1 hash values of the certificates with the file names to be sure, that you really trust those certificates.
7. Copy any certificate, which is needed to verify the chain of certificates into the OPC_UA_PKI/CA folder.
8. Move any client certificate, which you like to blacklist into the OPC_UA_PKI/rejected folder.
9. Insert the SD card back into the EchoCollect and reboot the device.

8.2.14 Serial Protocol

COM Number

In the left panel of this dialog box, you can select the desired COM number. The graphic shows the port that will be used.

Depending on the device hardware, the available interfaces are enabled or disabled (grayed out).

Serial / PLC

Here, you can select the protocol to be used for the serial interface.

- Telnet
- 3964R
- AS 511 (S5 PG)
- MPI/PPI
- Modbus RTU
- DF1

8.2.14.1 Telnet Connection (Send/Receive)

In this dialog box, you can set the parameters for a Send/Receive connection and for a Telnet connection. Both are quite similar.

Name of the Connection

This field displays the connection name which can be changed here.

Connection Active

Shows whether the connection is active or not. If not, you cannot register items and there will be no connection to the PLC. This feature allows you to disable a connection temporarily without deleting it so that you can re-enable it later without having to enter all parameters again.

Standard, as fast as it can

Data will be transmitted as fast as possible without taking any length information into account.

Fixed Frame Length [in bytes]

When specifying a fixed length, characters will be read until the indicated fixed frame length (maximum: 2 KB) is reached. Then, these characters will be sent in a single frame.

Len Information in Frame: 8 Bit

The length of an item is indicated in an 8-bit value. The offset is specified below.

Len Information in Frame: 16 Bit (Little Endian) (INTEL format)

There are two different byte ordering methods, i.e. the representation or order of the bits/bytes in a word: Little-endian and big-endian. All components that should communicate with each other must invariably use the same method. In the big-endian method, bytes are ordered "from left to right". In the little-endian method, however, they are ordered "from right to left". INTEL, VAX, DEC and others are little-endian systems.

Len Information in Frame: 16 Bit (Big Endian) (Motorola format)

Byte ordering here is "from left to right". Motorola, SPARC, IBM and others are big-endian systems.

Len element is at Offset

If the frame contains length information, this value indicates the byte offset address from which on length information can be found.

Use End Byte

If the frame contains a fixed end character, you can specify this value (in decimal notation) here. Next to it, the value is displayed in hex and ASCII notations.

Wait a Time for New Data

After each character, the waiting time will be checked. When the waiting time is over and no other characters were received, the collected data will be transmitted in a single frame.

Pad Frame with Zeros to 2 Byte Boundary

In this case, a frame is padded with zeros to obtain an even 16 bit value. This may be required for S5, Modicon or Pilz PLCs where the smallest unit of a data block (DB) is 16 bits.

Pad Frame with Zeros to 4 Byte Boundary

In this case, a frame is padded with zeros to obtain an even 32 bit value. This can be useful, e.g., with CNC PLCs where the smallest unit is 4 bytes.

Serial

This button opens the [Serial Line Parameters](#) dialog box where you can set the parameters.

8.2.14.2 3964 and 3964R Connection

Name of the Connection

Here, you can enter or edit the connection name.

Connection Active

You can disable the connection without deleting it. This way, the connection parameters will be retained. You can then re-enable the connection later, using the same parameters as before.

Special Settings

These settings are only available if procedure RK512 is used.

Fetch / Write Connection

Select a Fetch / Write connection if you would like to read or write data actively (Fetch Active / Passive, Write Active / Passive job types)

3964 Base Settings

No RK512

can be selected if the procedure 3964 or 3964R is used. In this case, the transmission is a raw data transmission. No other settings are required.

RK512 active

The activity is initiated from the Ethernet side, i.e. the Fetch Active, Write Active job is sent from the Ethernet station to the serial side (e.g. read data from the PLC actively using an OPC communication).

RK512 passive

The activity is initiated from the serial side, i.e. the Fetch Active, Write Active job is sent from the serial station to the Ethernet station.

RK512 symmetrical

The activity is initiated from both sides, i.e. both the serial side and the Ethernet side perform active jobs.

PLC Coordinate Flag in RK512

Coordinate flags are only relevant to the RK512 procedure. The 3964 / 3964R procedures do not use these flags.

Using a coordinate flag

If a coordinate flag is to be used, you can enter the byte and bit nos. of the flag here. Each coordinate flag occupies a flag bit in the flag area of the CPU. For information on the value range for the coordinate flags (byte and bit nos.), please refer to your CPU manual. Please make sure that the link partner will be able to evaluate this coordinate flag, i.e. that it has been specified in the PLC program. Please also make sure that the value entered in the device is not used elsewhere in the PLC program.

Using no coordinate flag

If you do not want to use a coordinate flag, please refer to your CPU manual to find out how to proceed. Many systems specify flag 255, bit 15 as the coordinate flag range to show that no coordinate flag is being used. This is also the default value in your echo device.

CPU

Here, you can specify the CPU number the coordinate flag is associated with. This defaults to the value "1". If you do not use a coordinate flag, also specify the CPU as unused. In many systems, the value "15" serves this purpose. Here again, please refer to your CPU manual to find the applicable value.

Serial

This button opens the [Serial Line Parameters](#) dialog box where you can set the serial parameters for the line. Please note that these parameters must be identical for both communication partners.

8.2.14.3 AS511 (Pg S5) Connection

In this dialog box, you can set the parameters for a connection using the [AS511 protocol](#) on the serial side.

Name of the Connection

All parameterization data is associated with the connections. You can identify each connection by its connection name. If required, you may edit it here.

Net Parameters

Here, the parameters set so far for the network side of the device are displayed.

Serial Parameters

Here, the parameters set so far for the serial side of the device are displayed.

Connection Active

You can disable the connection without deleting it. This way, the connection parameters will be retained. You can then re-enable the connection later, using the same parameters as before.

Fetch / Write Connection

Select the Fetch / Write Connection option if you want the S5 PLC to receive data passively (Write Passive) or send data passively (Fetch Passive).

Fixed serial parameters

Serial standard PG interfaces are operated with 9,600 bauds, 8 bits, even parity and 1 stop bit. Due to the design of the serial PG interfaces, the corresponding COM port of the device is invariably set to these values.

8.2.14.4 MPI Connection

Name of the Connection

Here, you can enter or edit the connection name.

Connection Active

You can disable the connection without deleting it. This way, the connection parameters will be retained. You can then re-enable the connection later, using the same parameters as before.

Destination MPI Address

This is the MPI address of the communication partner you want to communicate with. The default is 2. For some MPI adapters, you can enter 0 here to allow automatic determination of the address.

CPU Slot

This is the CPU slot in the PLC. The default is 2. For some MPI adapters, you can enter 0 here to allow automatic determination of the slot.

Type of Service

Here, you can specify the desired service type. Depending on the CPU type, the number of connections available for the selected service may vary.

Fetch / Write Connection

Select the Fetch / Write Connection option if the S7 PLC connected to the echo device will passively respond to requests sent from the echo device. This is for example the case for an OPC communication.

8.2.14.5 Modbus RTU Connection

Name of the Connection

All parameterization data is associated with the connections. You can identify each connection by its connection name. If required, you may edit it here.

Net Parameters

Here, the parameters set so far for the network side of the device are displayed.

Serial Parameters

Here, the parameters set so far for the serial side of the device are displayed.

Connection Active

You can disable the connection without deleting it. This way, the connection parameters will be retained. You can then re-enable the connection later, using the same parameters as before.

Fetch / Write Connection

Select a Fetch / Write connection if you want to read data from a Modbus slave and/or write them to the Modbus slave.

Modbus Station Address (Slave Address)

Here, you can specify the slave address of the serial Modbus device. It will be transmitted in each frame.

Factor of Address in Blocks

Not implemented for RTU.

Use Multiplier Factor

Not implemented for RTU.

First Element is 1

Disabled: The starting address in the frame will be transmitted as specified. The first address is 0.

Enabled: Some Modbus devices expect a starting address in the frame that is 1 less than the one you entered. The first address is 1.

Use Modbus TCP Slave Address

It is not necessary to set the slave address of the serial Modbus device explicitly if the Ethernet client uses the Modbus-on-TCP protocol for communication. In this case, the Modbus header of the request contains the slave address of the destination device (Unit Identifier Field). The echo device will then use this identifier to forward the request to the appropriate slave device. If the partner station on the Ethernet does not use the Modbus-on-TCP protocol, you must indicate the slave address in the Modbus Station Address (Slave Address) field.

Serial

This button opens the [Serial Line Parameters](#) dialog box where you can set the parameters.

8.2.14.6 DF1 Connection**Name of the Connection**

All parameterization data is associated with the connections. You can identify each connection by its connection name. If required, you may edit it here.

Net Parameters

Here, the parameters set so far for the network side of the device are displayed.

Serial Parameters

Here, the parameters set so far for the serial side of the device are displayed.

Connection Active

You can disable the connection without deleting it. This way, the connection parameters will be retained. You can then re-enable the connection later, using the same parameters as before.

Station Slave Address

Here, you can specify the destination address.

Protocol

You can select either Half Duplex or Full Duplex.

Error Checking

You can select either BCC or CRC.

Fetch / Write Connection

Select a Fetch / Write connection if you would like to read or write data actively (Fetch Active / Passive, Write Active / Passive job types)

Serial

This button opens the [Serial Line Parameters](#) dialog box where you can set the parameters.

8.2.14.7 Serial Line Parameters

Line

Shows the selected serial interface.

Baud Rate

The baud rate (also referred to as modulation rate) indicates the number of state changes in the transmitted signal per second. You can set baud rates from 75 to 115200. The same baud rate must be used for the sending and the receiving stations.

Parity

The same parity must be used for the sending and the receiving stations.

Even

For even parity, the number of 1s (including the parity bit) is even.

Odd

For odd parity, the parity bit is set in a way to obtain an odd number of 1s (including the parity bit).

None

Select this option if you do not want to use the parity check.

Mark

Select this option if you want the parity bit to be permanently 1, i.e. if your device expects a 1 parity bit.

Space

Select this option if you want the parity bit to be permanently 0, i.e. if your device expects a 0 parity bit.

Protocol

Here, you can select the protocol mode to be used. The devices use it to communicate whether they are ready to send / receive data. There are two types of handshake: software handshake and hardware handshake.

Hardware handshake means that synchronization is done via electrical wires. V.24 typically uses either RTS / CTS or DTR / DSR for signaling.

Software handshake means that synchronization is done by transmitting control characters. XON / XOFF are the most common software handshake characters.

RTS / CTS

RTS= Request To Send, signal of the V.24 interface

CTS= Clear To Send, signal of the V.24 interface

DTR / DSR

DTR= Data Terminal Ready (to send), signal of the V.24 interface

DSR= Data Set Ready (for operation), signal of the V.24 interface

XON / XOFF

The communication partners add ASCII characters to the communication data stream. These control characters in the data stream indicate whether the data flow is possible or not:

XON: data can be received

XOFF: data cannot be received

RS485

Select RS485 if the serial device is connected to the echo device via an RS485 cable.

DTR RTS = 1

Select DTR RTS = 1 if DTR or RTS default to 1

None

Select None if you do not want to use any of the protocols listed above for the serial line.

Bits

Indicate the number of bits per character here: 5, 6, 7 or 8

Stop Bits

Duration of the stop bit transmission, relative to the transmission time of an information bit. For asynchronous, serial lines, the transmission of a character is terminated by sending a stop bit. Specify here whether 1 or 2 bits will be used for termination.

Timeout

This option specifies the time after which the device will no longer wait for a response and assume the partner is not present.

The value given in the field is not an absolute time but instead a factor that can be used to extend the standard timeout by the given factor.

The timeout (in ms) for the reply of the modbus RTU device connected to the echo will be calculated as follows:

$$\begin{aligned} \text{timeout} &= \max \left\{ 120\text{ms}, \text{factor} * \frac{2 * (3.5 * 11) + 2 * (256 * 11)}{\text{Baudrate} [\text{bit/s}]} \right\} = \\ &= \max \left\{ 120\text{ms}, \text{factor} * \frac{5709}{\text{Baudrate} [\text{bit/s}]} \right\} \end{aligned}$$

For example, for *baudrate* = 19200 and *factor* = 1 we get the *timeout* = 297 ms

8.2.15 Database Connect Parameters**ConnectionName**

Here, you can edit the connection name, if required.

Database Type

Here, you can select the database type: MySql, Oracle, Ms-SQL, Sybase, DB/2.

IP Settings**Destination IP Address**

When establishing an active connection, you must enter the destination IP address. For passive connections, we recommend to enter the wildcard address (0.0.0.0).

If you have selected the special setting use tnsnames.ora entry for an Oracle database, then it is possible to use such an entry by double clicking on it to fill the destination address field.

Port

When establishing a connection, the default port for the corresponding database will be suggested. You can change it, if required.

Life Data Acks

Life data ACKs are frames that enable heartbeat monitoring. They are sent to keep up a database connection, even if no frames with payload are being exchanged. It is recommended to enable this function. If a WAN connection is used, you can disable this function for cost reasons. Otherwise, the TCP/IP connection will be kept up permanently.

Username

Here, you can enter the user name with which the device logs on to the database. Not all systems require a user name. To get the correct login data for the database, please ask the system administrator.

Password

Enter the password assigned to the user. Not all systems require a password. To get the correct login data for the database, please ask the system administrator.

Database

Here, you can enter the database name. For some database types, this is the database schema (MySQL), others use this information as the starting point, still others only have one fixed database. Depending on the permissions allocated, database requests reset this database name.

Special Settings

This button gives access to special parameter settings for the selected database. Usually, these parameters are not required. If problems are encountered, you can compare these parameters with the database requirements.

General Information about Oracle Database Connections

Oracle Databases are connected using the Instant Client Light Version 10.2.0.5.0. It can connect to databases having one of the following database character sets. An error is returned if a character set other than those in the list is used as the client or database character set.

- US7ASCII
- WE8DEC
- WE8MSWIN1252
- WE8ISO8859P1
- WE8EBCDIC37C
- WE8EBCDIC1047
- UTF8
- AL32UTF8 (generally recommended)

The echo collect implements an oracle client with the language environment set to `NLS_LANG=AMERICAN_AMERICA.AL32UTF8`

Oracle Database compatibility follows the support scheme of the Instant Client Library. Supported database versions:

- 11.1.0
- 10.2.0
- 10.1.0

deprecated but still supported:

- 9.2.0
- 8.1.7

Special characters like german Umlaute are not allowed for names of tables, columns, service name, login data. Refer to the documentation of oracle for allowed characters

Limits of oracle database queries

- maximum number of elements requested in a "select": 240

- string size: 255 (+null terminator) characters
- maximum sql query size: 4096 characters

Supported sql commands in Oracle SQL Queries

fully supported are "SELECT", "INSERT", "UPDATE". Stored procedures and functions using "EXEC" have limitations. Procedures are supported with "call by value". "Call by reference" for return values in procedures are not supported. Functions can't be called by just using "EXEC" but using a "SELECT" additionally. Returning arrays from functions are not supported, only scalars can be returned.

General Information about MySQL Database Connections

For MySQL connections the maximum SQL query size is 2100 characters.

8.2.15.1 MySQL Connection Flags

You can set the following parameters:

MySQL Connection Flags	
Enable Multiple Statement Execution	The performance will increase, provided your database administrator allows you to use this command.
Enable Immediate Commit	If this setting is enabled, the performance will be faster, but in case of network problems, the echo device might not realize that a database command has already been executed on the server (because the connection is lost). If this command is then retransmitted, the record will be stored twice in the database. If you disable this setting, 2 frames will be exchanged for each command (the command itself + separate commit). This will fix the problem, but the performance will be slower.
Enable Compression	The network load will be reduced and the performance will increase.
Enable SSL Encryption	The connection is more secure, but slower. Here again, the database administrator must enable this function for you.

8.2.15.2 Oracle Connection Settings

You can set the following parameters:

Connection Method	
SID SERVICE_NAME	Depending on the configuration of your database listener service, you need to select SID or SERVICE_NAME as connection method.
USE tnsnames.ora file entry	As an alternativ to the manual setting of IP address, port number and SID or SERVICE_NAME the connection settings from an entry of the tnsnames.ora file could be used. This allows for example the connection to a redundant database net. For details please refer to the Oracle documentation.

The buttons "Upload tnsnames.ora file into device" and "Delete tnsnames.ora file from device" allows you to store a tnsnames.ora file in the device or to remove the file from the device. For this to work, it is required, that for the ftp web user, an user name and a password has been assigned.

8.2.15.3 MsSQL/Sybase Connection Settings

You can set the following parameters:

MsSQL/Sybase Connection Settings	
Version 4.2	for Sybase before 10.0 for MS-SQL before 7.0
Version 5.0	for Sybase 10.0 and later
Version 7.0	for MS-SQL 7.x
Version 8.0	for MS-SQL 2000 and later
Autocommit setting	
Explicitly enabled	Automatic commit is on.

Explicitly disabled	Automatic commit is off.
Not explicitly set	Automatic commit uses the default behavior.

8.2.16 MQTT Broker Configuration

The MQTT protocol knows the following peers:

1. **MQTT Broker:** The MQTT Broker is the central instance in an MQTT network. The other subscribers establish each a TCP or SLL/TLS connection to the broker. Depending on the MQTT Broker configuration an authentication with user name and passwords or an SSL certificate is required. The MQTT Broker receives data from the MQTT Publishers. If a certain publisher does not have the required write permissions, the corresponding date is discarded by the broker. Otherwise the broker provides the data to all MQTT Subscribers that have subscribed to this data.
2. **MQTT Publisher:** The MQTT Publisher create the MQTT address space (topics) and fill this space with content (data). The MQTT Publisher sends this data to the MQTT Broker. This is exactly the role of the MQTT Publisher Module.
3. **MQTT Subscriber:** The MQTT Subscribers subscribe to MQTT Topics. In this context they are allowed to use the wild card characters '+' and '#'. Thus, an MQTT Subscriber that only uses the character '#' for topic subscription, subscribes all data from a broker.

The data format is not specified by the MQTT protocol specification but can be specified by the respective MQTT publisher. The MQTT Publisher Module uses strings as data format.

8.2.16.1 Connection Settings

This page allows to specify connection settings such as e.g. the unique connection name for identifying the connection.

Connection Name

A unique identifier for the current connection. In case the provided name is already used by other connection, the user use is not allowed to proceed to the next wizard page.

Client ID

The client ID must be unique for the respective MQTT Broker. If two MQTT clients are using the identical client ID, the broker is disconnected. The client (as e.g. echocollect) cannot recognize that the client ID is already used by another client.

The MQTT Broker can be configured in a way that it only accepts connections with specific client IDs.

Connection active

This flag specifies if the current connection will be activated or not.

8.2.16.2 Communication Settings

This page allows to parameterize the communication and authentication of the MQTT Broker.

Broker URI

The Broker URI is composed of the transport protocol, the Full Qualified Domain name (FQDN, consisting of host name and domain name [see wikipedia]) optionally followed by the port number separated by a colon. For the transport protocol, the following variants could be chosen:

1. `tcp://` MQTT uses the plain TCP. MQTT brokers usually offer this service at port 1883.
2. `ssl://` MQTT uses SSL or TLS security upon TCP. MQTT brokers usually offer this service at port 8883.

The FQDNS can be replaced by the corresponding IPv4- or IPv6-address.

Authentication settings

Depending on the MQTT Broker configuration echocollect may need to identify itself as an MQTT client with user name and password or by a client certificate.

In this case you need to select the entry "Username and Password" in "User Identity" Then fill out the input fields for user name and password accordingly. If needed activate the "Use Client Certificate" check box and

assign a corresponding client certificate file through the [MQTT Client Certificate dialog](#).
If no identification is required, leave the entry "Anonymous" in "User Identity".

MQTT Publish Options

- **MQTT QoS:**

Level 0: The MQTT message is published with the safety of the current TCP connection.

Level 1: A MQTT message is repeated in the next TCP connection, if the delivery of the message in the current TCP connection fails.

Level 2: MQTT protocol uses confirmation of confirmations to ensure that a message is delivered exactly once.

If the device is equipped with a writable SD-card and the setting "Publish only on value change" (see below) is active, MQTT messages with QoS1 and 2 are stored on the SD-card while the broker is unreachable. Those messages will be delivered to the broker, as soon as a connection to the broker could be established.

The storage to SD-card is limited by the `vfat` filesystem of the SD-card to 16335 files per directory.

- **Enable MQTT Retain:** By setting the Retain flag the MQTT Broker is instructed to save the most recent data value for the topic. (Depending on the configuration the broker saves the data into the RAM or persistently into the file system/data base.) Data values without Retain flag are only transferred from the MQTT Broker to those MQTT Subscribers that are registered in that moment by the corresponding broker and have subscribed the corresponding topic. This check box allows to specify whether the MQTT Publisher Module sets the Retain flag for the respective topic or not.
- **Minimum publishing interval [s]:** The same topic is not published with a faster interval than this setting in seconds. This setting could protect the MQTT broker and MQTT subscribers from a flood of too many publish messages of a single topic. If the polling interval is slower than this setting, than the polling interval overrules this setting.
- **Publish only on value change:** If this setting is activate, a data value would only be published to the MQTT broker, if the value has changed and the minimum publishing interval has been expired.

8.2.16.2.1 MQTT Broker Certificates

This dialog provides options to facilitate the secure communication between client and external broker by certificate handling.

Download MQTT broker Certificate chain into the MQTT publisher trusted folder

By pressing the receive broker certificate button the user initiates a server connection and implicitly a certificate exchange. Once the server certificate chain is obtained it is presented to the user into [certificate validation](#) to be investigated. If the user accepts the certificate chain it would then added to the trusted folder and assigned to the connection. Otherwise the certificate chain is removed from the file system.

Import MQTT broker Certificate chain into the MQTT publisher trusted folder

By using the import button the user is prompted to select a broker certificate chain (*.crt* or *.pem* format) that will be added to the trusted folder. This operation basically can substitute Receive server certificate operation.

8.2.16.2.1.1 MQTT Broker Certificate Validation

This dialog displays the received certificate chain of the MQTT broker.

The user may accept or discard the received certificate chain. If the user accepts the certificate chain it would then added to the trusted folder and assigned to the connection. Otherwise the certificate chain is removed from the file system.

8.2.16.2.2 MQTT Client Certificate

This dialog allows to select the client certificate, which is intended to identify the client by the MQTT broker.

This dialog provides the following input fields:

1. The file in PEM format containing the public certificate chain of the client. It may also include the private key of the client.
2. If not included in the KeyStore above, this setting points to the file in PEM format containing the private key of the client.

3. The password to load the privateKey of the client, if encrypted.

8.2.16.3 MQTT Topic Settings

This page allows to configure the settings of the topics which are going to be published to the MQTT broker. The following settings are provided:

- **MQTT Root Topic:** The specified root topic is prefixed to all MQTT topic names which are going to be published. If this setting is not empty, than a trailing slash ("/") is inserted between the root topic part and the automatically generated topic name parts.
- **Hierarchy:**
 - Full PLC hierarchy: The full PLC hierarchie will be translated into MQTT hierarchy.
 - Flattened PLC hierarchy: The PLC hierarchy is flattened into just one MQTT hierarchical level.
 - Suppressed PLC Symbols: The PLC symbol names are not part of the MQTT topic name(s). Use this setting, if the MQTT broker allows only to publish to one single MQTT topic.
- **MQTT Suffix Topic:** The specified suffix topic is appended to each MQTT topic to publish. Depending on the requirements of your MQTT broker and application, you may need to begin the suffix topic with a leading slash ("/").
- **Publish Format:** The user could define the format string of the data to publish. The following, case sensitive keywords will be replaced with the corresponding data:
 - a. @VALUE@: The value of the PLC item.
 - b. @TIME@: The timestamp of the PLC item.
 - c. @QUALITY@: The quality attribute of the PLC item.
 - d. @ITEM@: The symbolic name of the PLC source item.
- **Collect PLC values in one MQTT message:** To reduce amount of MQTT messages, up to 10 PLC values could be transmitted in one MQTT message.
- **PLC Value marker:** In these input fields you could specify the start-sequence, the delimiter-sequence and the end-sequence of the grouped MQTT message.

8.2.16.4 MQTT Topic Selection

Usually a PLC project contains a huge number of PLC items. But usually only a small subset of these items are of interest for publishing to MQTT.

For performance reasons you should activate only the items which you are interested!

This page allows to select the PLC items which should be published to the MQTT broker.

Therefore this page provides you a list of all imported scalar PLC items and a list of all PLC items selected for publishing.

Filter

For a better overview, you may set a filter into the "Filter" input field. The filter is automatically applied to both lists while typing.

The filter input field is internally used as case insensitive regular expression. So for example the following filters could be applied:

- You are interested in all PLC items which contain the string "bool", then use the string *bool* as filter.
- You are only interested in all PLC items which end in the String "bool", then use the string *bool\$* as filter.
- You are interested in all PLC items which contain the string "counter" or the string "timer", then use the string *counter|timer* as filter.
- You are interested in all PLC items which contain the string "input" followed by a number, then use the string *input\d+* as filter.

Activate or deactivate Items

You could activate items for publishing by marking them in the 'Available Items' list and add them to the 'Selected Items' list with the 'Add Items to selected Item list'.

You could deactivate items for publishing by marking them in the 'Selected Items' list and removing them with the 'Remove Items from selected Item list'.

8.2.16.5 MQTT Last Will Settings

This page allows to perform the "Last Will and Testament" settings of the MQTT connection.

When establishing a connection, MQTT clients such as the echocollect MQTT Publisher can transfer a topic name and a message.

If the Broker detects that the connection to the client does not exist any longer (e.g. because a network switch has failed), then the Broker publishes this message within the topic.

Complete name of the Last Will topic including all hierarchy levels.

The individual hierarchy levels are separated by the '/' character. They must be entered for the Last-Will topic.

Message

Enter the message into this field that is to be published by the Broker as testament.

Retain

By setting the Retain flag the MQTT Broker is instructed to save the most recent data value for the topic.

Depending on the Broker configuration it saves the data in RAM or persistently in the file system/database.

Data values without Retain flag are only transferred from the MQTT Broker to those MQTT Subscribers that are registered in that moment by the corresponding broker and have subscribed the corresponding topic.

Using Last Will and Testament

This check box allows to specify whether the "Last Will and Testament" MQTT connection options are used or not. If they are used, additional configuration fields are available.

8.2.17 Memory Table Definition

A memory table collects data. It can be used to perform calculations. For this purpose, you need to define the data type of the entries and the number of items.

These memory tables are often populated with data every second. Thus, the memory might contain e.g. a list of temperature values that can be displayed as a curve on a web site. In addition, you can specify calculations to be performed, such as the maximum, minimum or average values. The data and the associated calculations can be stored in other allocation tables, e.g. in a database. This avoids complex calculations in stored procedures within the database.

Data can be read from or written to memory tables.

In this dialog box, you can also enable access via OPCpipe.

Name of the Connection

This field displays the connection name which can be changed here.

Connection Active

Shows whether the connection is active or not. If not, you cannot register items and there will be no connection to the PLC. This feature allows you to disable a connection temporarily without deleting it so that you can re-enable it later without having to enter all parameters again.

OPCPipe Access allowed

Using this option, you can allow an OPCpipe server connection to access the connection configured here. Thus, it is possible to access the PLC from a remote PC via this connection.

OPCPipe Write allowed

Using this option, you can enable or disable Write access via OPCpipe.

Memory is Ring Buffer

When you use the push command to write data, an item will be inserted at the beginning. R0 reads the oldest item. RR0 reads the newest item.

Function Calculation

The device automatically calculates a value using the selected function and suggests the value as a variable.

Function Calculation	Variable	Description
Sum Value	sum	Calculates the sum of all values in the table.
Average Value	avg	Calculates the average of all values in the table.
Minimum Value	min	Finds the lowest value in the table.
Maximum Value	max	Finds the highest value in the table.

Element Type

Allows you to set the data type for the entire memory table.

Size

Indicates the number of items in the table.

To access a memory table, a separate item syntax is available that is described in the [Virtual Collect Item Syntax](#) section.

8.2.18 Mail Connect Parameters

In this dialog box, you can edit the e-mail parameters.

For each e-mail connection, you need to specify the IP address or domain name of the mail server.

For details on the parameters, please ask your mail operator or your e-mail provider or look them up in your e-mail contract.

Connection Name

You can choose any connection name you like. All characters except brackets may be used.

IP Settings**Destination IP Address (or name)**

Here, you can specify the IP address of the destination station (IP address of the PLC) or the DNS name of the destination station, i.e. of the mail server. If the device was configured for DNS and a DNS server is available in the network, you can also enter the symbolic name of the destination station.

For information on the structure of IP addresses, please refer to the [IP Address](#) section.

Port

The default mail sending method (SMTP) often uses port 25.

Sender Address

Here, you can specify the address for replies.

Authentication

Depending on the mail server, login may be required.

Username and Password

These are the login data.

Login type

Select Auto Select or, if known, please specify.

The following protocols can be selected:

- PLAIN
- LOGIN
- CRAM-MD5

8.2.19 Collect Slave Config. Edit**Optimizing****Max. read gap:**

Specifies whether and to what extent data requests will be combined. In most cases, it makes more sense to read larger data blocks, which sometimes include unnecessary data, instead of multiple small requests. This depends on the PLC but, above all, on the connection speed.

Don't combine different types

If this option is selected, requests for different data types will not be combined in the PLC protocols.

Example for S7: The request includes mb0.2 and mw2.

Option disabled: The request is sent with mb0.4, i.e. the device combines the requests.

Option enabled: Two requests are sent in a single frame: mb0.2 and mw2.

Handling details

Disable multiple plc areas in one request

The High Performance Protocol allows you to perform multiple different block requests in a single PLC frame. A few PLCs provide this option but fail to process the data accordingly; in these rare cases, you should enable this option.

Keep time sequence for write

If you enable this option, all data will be written sequentially to the PLC. They will never be combined to a single frame. This means that processing may become slow. Only enable this option if this is required by the PLC program.

Timeout

Application Timeout

Resets the connection if the PLC does not respond to a request within the time specified.

Reconnect Error Timeout

If you set a value here, the connection will only be considered as faulty after the time specified. If the connection is reestablished within this time, no error will be recorded. The default is 0 -> immediately detect as an error.

8.3 Edit Connection

Name of the Connection

This field displays the connection name which can be changed here.

Connection Active

Shows whether the connection is active or not. If not, you cannot register items and there will be no connection to the PLC. This feature allows you to disable a connection temporarily without deleting it so that you can re-enable it later without having to enter all parameters again.

Write allowed

For some connections, you can disable the Write function.

OPCPipe Access allowed

Using this option, you can allow an OPCpipe server connection to access the connection configured here. Thus, it is possible to access the PLC from a remote PC via this connection.

OPCPipe Write allowed

Using this option, you can enable or disable Write access via OPCpipe.

Poll interval ... ms

Here, you can specify the poll rate the device will use for reading data from the PLC.

Buttons

Depending on the connection type, you can edit further parameters.

Button	Dialog box that opens
Network Protocol	Network PLC Protocol
Network Parameters	TCP/IP Connect Parameters or H1 Connect Parameters or Mail Connect Parameters
OPCpipe Parameters	OPCpipe Parameters
Serial Parameters	The dialog box for editing the serial parameters opens. For a description, see the Serial Protocol section.
Memory Settings	Memory Table Definition

Performance Settings	Collect Slave Config. Edit
Collect Settings	Collect Destination Connection Config.
Protocol Parameters	Opens the dialog box for protocol-specific parameters.

The number of available options depends on the type of connection.

8.4 Copy Connection

It is possible to copy a connection. All parameters (except the connection name) of the highlighted connection will be copied. Make sure to specify the new connection name before saving the connection. Then, the connection will be displayed in the [Connections](#) list.

8.5 Delete Connection

The highlighted connection will be deleted. This process cannot be undone!
As an alternative to deleting, you can disable a connection using the [Switch on/off](#) option.

8.6 Disable Connection

To disable a connection, do one of the following:

- Menu: **Connection > Switch on/off**
- Highlight the connection, **right-click** and select **Switch on/off**

In the [Connections](#) list, the word (**off**) is displayed after the type.

You can disable a connection without losing the connection parameters. Later, you can re-enable the connection using the same parameters as before. By default, connections are enabled.

9 Main Menu

The menu provides the following options:

- [File](#)
- [Connection](#)
- [Diagnostics](#)
- [Collect](#)
- [Station](#)
- [Settings](#)
- [Help](#)

9.1 File

The File menu provides the following functions:

Load configuration from file

This menu item is used for loading the parameter file to the device.

- Click this menu item to open the Windows dialog box: Open File...
- Select the desired file and click the **Open** button.
- A new dialog box appears: Copy from File into the Device
- Here you can choose if you want to overwrite the MAC address, the IP address and the IP settings in the device.
- When you click the **OK** button, the parameters are irreversibly overwritten in the device.

Store configuration to file

Using this menu item, you can save all the device's parameters to a parameter file.

Store configuration to SD-Card

Using this menu item, you can save all the device's parameters to a sd-card in the device.

Attention: The configuration stored on the SD-card will replace the current configuration when the device restarts.

Refresh

Reestablishes the configuration connection to the device. If a password has been set, the password needs to be re-entered to allow editing.

This function is particularly useful for serial links. It allows rereading the list of connections from the module. This is necessary, for example, if the connection cable is plugged into a different module. The Connections list is cleared and rebuilt with the current connections.

Print

For documentation purposes, you can output a list of the connections.

The list is printed on the system's default printer.

Close Window

Closes the configuration of the device and switches to the [Configuration](#) window.

9.2 Connection

This menu provides the following options:

Switch on/off

You can disable a connection if it is not intended for communication. Thus, it is possible to disable it temporarily without having to delete and redefine it. See [Disable Connection](#).

To use this function, master access is required.

New Connection

Here you can enter a new connection. The [New Connection](#) dialog box appears. First, select the connection name and type. Then enter the required parameters.

When you save the connection, the connection parameters take effect immediately.

To use this function, master access is required.

Edit Connection

Here you can edit the parameters of the selected connection. The [Edit Connection](#) dialog box appears.

To use this function, master access is required.

Copy Connection

You can copy the connection. The [Copy Connection](#) dialog box appears. Before clicking Save, please keep in mind to rename the copied connection.

To use this function, master access is required.

Delete Connection

The connection is deleted after you confirm the prompt in the [Delete Connection](#) dialog box.

To use this function, master access is required.

9.3 Diagnostics

This menu provides the following options:

Show one Connection

The connection selected in the Connections list can be monitored. The [Connection Diagnostics](#) dialog box appears.

Signal Device

When you click this menu item, all the LEDs will blink for about 2-3 seconds. This shows you which device the NetCon is connected to.

MPI Ring Diagnostics

Opens the [MPI Ring Diagnostics](#) dialog box, which provides diagnostics for the connected MPI bus.

IP/NetStat

Opens the [IP/NetStat](#) dialog box, which displays an overview of the connections entered in the runtime system.

Start Capture Network

If the device is equipped with a SD-card, then this menu entry starts the capture of the data traffic at both Ethernet interfaces. The capture ends after ca. 25 hours, the stop command or when the device restarts. If there is already one capture running, then no additional is started.

Stop Capture Network

This menu entry terminates a previously started capture of network traffic.

Serial Diagnostics

The menu item opens the [Seriell](#) dialog box, which provides diagnostics for the serial lines. All messages and transferred bytes are logged in this window. Using the check boxes at the bottom, you can enable the individual lines.

See also [Using the Diagnostic Text Output System](#).

Logger

Opens the [Logger](#) window where the recorded messages are displayed.

This window can be opened in parallel to the software application. You can continue to configure and use the software while the Logger window is open.

Logger Settings

Opens the [Logger Settings](#) dialog box. There you can select the events that will be recorded by the Logger. The file and memory settings for the Logger are made under Settings in the Logger window.

Free place on sd-card

If a memory card is inserted, the dialog box shows the size of the memory card and the free memory space (version 7.0 and higher).

9.3.1 Device Connection Diagnostics (1 Connection)

The diagnostics window is divided into two parts:

- the network side
- the Collect or serial side

The number of transmitted frames and bytes as well as the number of received frames and bytes are shown for each side.

In addition, the window shows you the number of disconnects and the connection status.

For the Collect/serial side, you can also view the last byte transmitted or received.

For passive connections with multiple endpoints, like OPC UA and OPC pipe, the connection status is considered OK, when the listen socket is successfully opened.

For passive connections with only one endpoint, the connection status is OK, when the other side has successfully established the connection.

The contents are cyclically refreshed.

9.3.2 MPI Ring Diagnostics

The dialog box displays a list of the stations that are connected to the MPI bus, and provides diagnostics.

As long as the device has not established a connection over MPI-bus, the device uses MPI only passive.

While in passive mode, the own MPI address of the device is not used.

Use **Port** to select the relevant interface.

The dialog box will then show the MPI bus settings.

Underneath you will find a list of the individual stations together with their MPI addresses and states.

Refresh

Press this button to refresh the display.

9.3.3 TCP/IP Stati

The dialog box displays a list of the TCP/IP connections and provides diagnostics.

The individual columns show details about the configuration and status.

The contents are cyclically refreshed.

To scroll through the list, use the "**Page...**" buttons.

9.3.4 Logger

Here you can select different options to specify the information you want to log. The log data is displayed in the Logger window.

Error

Error	
Error	Log all types of errors in the Logger, except PLC errors.
PLC Element Error	If an area in the PLC does not exist or if a requested area is too small, a corresponding entry is made in the Logger. This also applies to all other errors generated by the PLC.
SQL Error	Logs all types of SQL connection errors in the Logger.
SQL Error Log File	The file name for logging SQL errors

Status

The reparameterization of connections and the login for reparameterization are recorded by the Logger.

Status	
Connection Parameterization	An entry is logged whenever connection parameters are changed and saved.
Connection Status Change	An entry is logged whenever the status of a connection changes.

PLC Data

The parameters of the respective PLC requests from the server to the PLC are recorded by the Logger.

PLC data	
Send	Content of the frames in write direction. The area that is written to is logged.
Send Data	The data content of the frames in write direction is logged.
Receive	Content of the frames in read direction. The area that is read is logged.
Receive Data	The data content of the frames in read direction is logged.

SQL Data

The requests and data of the Collect destination connection are recorded by the Logger. This can be a SQL connection, but it can also be of any other type. The direction is selectable.

SQL data	
Send	Content of the frames in write direction. The area that is written to is logged.
Send Data	The data content of the frames in write direction is logged.
Receive	Content of the frames in read direction. The area that is read is logged.
Receive Data	The data content of the frames in read direction is logged.

9.3.5 Logger Settings

In the Logger Settings dialog box, you can choose to keep the log data only temporarily in the RAM or save it to a log file. For this, you can specify the logging intervals as well as the directory and the file to which the log data will be saved.

Cache Size

Specify how much RAM you want to allocate to data logging. The cache acts as a ring buffer. If you increase the cache size, more entries will be buffered in the RAM.

Use File Logging

Select this check box if you also want to save the data to a file.

File Settings

Maximum Disk Space

Specifies how much disk space may be occupied by log data. When the value you set here is reached, the old files are deleted. This setting serves to prevent the log files from taking up all the hard disk space.

NOTE:

When the selected setting is reached, the Logger will automatically delete files from the directory without a confirmation prompt.

Directory

Specifies the directory in which you want to save the log files.

File Prefix

The file name is made up of the prefix followed by a sequential number and the date and time.

Save Every ...

Specifies the interval at which the file will be saved if the cache in the RAM has not yet completed one loop. When the ring buffer is full and the write interval time has not been reached, the ring buffer loops around more quickly and the old data in the ring buffer is being overwritten.

NOTE:

When you save to a flash drive, the number of writes is limited. The fewer writes, the longer the lifetime of the flash drive will be.

New File Every ...

Specifies when a new file will be started. This facilitates the evaluation.

File Name Example

The display field shows an example of a file. Here you can see where the log files are located and what the file names look like.

9.3.6 Serial Diagnostics

The dialog box shows the data logged for the serial driver. You can use this dialog box for serial communication diagnostics. The communication can be displayed either directly in the window or saved to a file. The file name is selectable. The file can be saved as a TXT or HTML file. You can choose to read once or cyclically (continuous debug in window or continuous debug in file).

In the [Using the Diagnostic Text Output System](#) section, you will find a description of how to use the dialog box.

9.3.7 Using the Diagnostic Text Output System

In the diagnostic text window, you can view and monitor the information you need for diagnostics.

When you open the window, the system's diagnostics memory is read, displayed and deleted. To display more data, click "**New read**" repeatedly. This clears the old window contents.

For a continuous display, click "**Continuous Debug in Window**". In this case, additional diagnostic data is appended to the existing contents.

To create a file, click "**In File**". This writes the current window contents to a file.

The "**Cont. Debug in File**" button creates a plain text file that will grow in length.

Using the "**Cont. Color Debug in File**" button, you can create a special-format file that will grow in length.

You can load these files again as you need them (the file icon opens the file selection). These files also contain the color information and the current timestamps for later evaluation.

You can filter individual contents during logging by selecting the relevant check boxes. The filter will only be enabled if the "**Direct**" check box is also selected.

To add timestamps to the files, select the **Timestamp** check box.

If you select the **Html** check box and write data to the file, the file is created in the format that can be viewed with a web browser. Please note that some web browsers may not be able to process very large HTML files.

9.4 Collect

The Collect menu provides the following functions:

Collect Frame Table

The menu item opens the [Collect Frame](#) window where you can enter and manage the Collect frame tables.

Collect Text Lists

Here you can create lists of texts that are used by the indexing of data. For example, an error message is displayed instead of a number. The dialog box is opened.

Collect Symbol Table

The menu item opens the [Collect Symbol Table Edit](#) dialog box where you can enter and manage the echocollect symbol tables.

Collect Mail Address Book

The menu item opens the [Collect Address Book](#) dialog box where you can manage e-mail addresses for automatic notifications.

All the menu items can also be accessed from the [Collect Frame](#) window.

9.5 Station

This menu provides the following options:

Station Password

Opens the [Change the Password](#) dialog box for creating, deleting or entering a password. Use this dialog box to specify whether a user is allowed to modify the parameters after the program start, or if the user has to log on with the password first.

Own Station 1

Opens the [Station Parameters](#) dialog box for displaying and entering the station parameters for the first network adapter.

Own Station 2

Opens the [Station Parameters](#) dialog box for displaying and entering the station parameters for the second network adapter.

MPI Station Values

Opens the [MPI Station Settings](#) dialog box for displaying and entering the MPI station parameters.

SSH

Opens the [Device SSH Settings](#) dialog box for displaying and entering the SSH parameters (Secure Shell, network protocol for encrypted connections).

SSH allows you to establish an encrypted VPN connection to the device via the Internet. It also offers the possibility to reach the device through a firewall, usually via TCP/IP port 22.

Set Original MAC Addresses

Using this menu item, a MAC address that may have been changed can be reset to the hardware address. Please keep in mind, however, that in other devices communicating with this echo device the communication will be interrupted until the ARP cache is updated or manually deleted.

On a Windows PC, this is done with the command:

```
arp -d *
```

9.5.1 Station Password

In this dialog box, you can set or delete the password. It also appears when a user logs on.

You can protect the configuration from modification by assigning a password.

To do this, open the dialog box by selecting the **Station** menu and clicking **Station Password**. Enter the password and repeat it in the Retype Password field. NetCon is now logged on and you can modify the configuration.

9.5.2 Station Parameters

This dialog box shows details of the local station.

Some values cannot be changed. The dialog box displays the determined information.

Station Name

You can freely select a name for the device. This name will be displayed when you run a station scan in the [Select a Station in Your Network](#) dialog box.

Station Password

If this dialog has been opened from the station search, then the input field for the password input appears. If a password has been assigned to the station, then this password needs to be entered here. In order to change the ip-address , etc. Otherwise this field must remain empty.

MAC Address

The MAC address of the network adapter. Every station in the Ethernet network has a unique Ethernet address. This address can be changed. To restore the original setting after a change to the MAC address, select **Station – Set Original MAC Addresses**.

Use DHCP

Here you can enable automatic configuration via a DHCP server. This check box should only be selected if a DHCP server exists in the local network.

IP Address

This entry specifies the station's address when the TCP/IP protocol is used, see [IP Addresses](#).

Subnet Mask

This parameter determines which station requests to the active network are allowed to pass. Addresses in which the masked parts differ are transmitted to the routers.

Domain Name

The domain name is used for symbolic requests in which no domain has been explicitly specified. This is usually the case with names that have no dot.

DNS Addresses

These addresses specify the servers that resolve the symbolic Internet names to the IP number addresses in the network.

Router Addresses

Routers are used if the station should also communicate with stations located outside its own network. The IP-Address of the router must be within the own local network, because the router connects this local network with other networks or the internet.

We distinguish between dedicated routers and the default gateway:

- For dedicated routers the IP-address of the router within the local network, the net-address and subnet mask of the destination network needs to be provided.
- For the standard gateway only the IP-address needs to be provided. In the usual case, there should be only one default gateway.

Routing examples

Assume that the device has the following configuration:

- IPv4 Address of the 1st interface: 10.123.45.78
- Subnet mask of the 1st interface: 255.0.0.0
- Default gateway of the 1st interface: 10.0.0.1
- IPv4 Address of the 2nd interface: 192.168.1.2
- Subnet mask of the 2nd interface: 255.255.255.0
- Default gateway of the 2nd interface: not set
- Dedicated route at the 2nd interface: Address of the router: 192.168.1.1, net: 192.168.0.0, subnet mask: 255.255.0.0

And the device should reach the following destination addresses over IPv4: ntp.pool.org within the internet, 10.0.0.2, 192.168.1.3 und 192.168.47.11.

These destination could be reached as follows:

1. ntp.pool.org should be resolved to an address within the internet. The device attempts to reach this destination through the default gateway 10.0.0.1 at the 1st interface.
2. Destination 10.0.0.2 is within the local net of the 1st interface. Therefore it could be reached directly.
3. Destination 192.168.1.3 is within the local net of the 2nd interface. Therefore it could be reached directly.
4. Destination 192.168.47.11 is not within any of the two local networks. But it could be reached through the router 192.168.1.1 of the 2nd interface.

9.5.3 MPI Station Settings

The dialog box shows the ports of the echo device. When you select a port, the [MPI Station Settings](#) dialog box appears, where the MPI parameters of the port are displayed. Ports that are not suitable for the hardware used are not available for selection (grayed out).

9.5.3.1 MPI Station Settings

The dialog box shows the station settings for the MPI bus and allows configuring the parameters.

COM

Displays the selected COM port.

MPI S7-300 or S7-400 / PPI S7-200

Here you can choose whether the device is to communicate with an S7-400 / 300 via the MPI protocol or with an S7-200 via the PPI protocol. While MPI is provided on-board, a serial-to-PPI converter is required for PPI communication.

Serial Baud Rate

Here you can enter the baud rate for the **serial side**, if the connection between the S7 PLC and the device uses an RS232 MPI/PPI converter.

MPI Baud Rate

Here the baud rate of the MPI bus is specified.

Own MPI Address

Here you can enter the station's own MPI address.

Max MPI Address

Here the highest MPI address is specified.

Defaults

The entries are overwritten with the most commonly used values.

Extended Settings

Opens the [MPI Station Extended Settings](#) dialog box where you can configure additional settings for the MPI bus.

9.5.3.2 MPI Station Extended Settings

In this dialog box, you can configure the extended settings for the MPI bus.

Own MPI Address

Here you can enter the station's own MPI address.

Max MPI Address

Here the highest MPI address in the MPI ring is specified.

Slot Time (TSL)

The slot time is the maximum time that the sender will wait for a response from the addressed station.

Minimum Station Delay of Responders (min TSDR)

(Minimum Station Delay of Responders (min TSDR))

This parameter specifies the minimum time that a remote receiver has to wait before sending an acknowledgment.

Maximum Station Delay of Responders (max TSDR)

Maximum Station Delay of Responders (max TSDR)

This parameters specifies the minimum time that a sender has to wait before transmitting a frame after the end of the previously received frame.

Quiet Time (TQUI)

Time between 2 requests.

Setup Time (TTR)

This parameter specifies the time that may elapse between the receipt of a data frame and the response to that frame.

Target Rotation Time (TTR)

This parameter specifies the target token rotation time. Every master compares the target token rotation time with the actual token rotation time. The difference determines how much time is available to the MPI station to transmit its own data frames.

GAP Update Factor

Here you can enter the GAP update factor. GAP means the distance from the local MPI address to the next MPI address. The update factor specifies the number of token rotations after which the master will check if there is another master in the GAP. Example: GAP = 5 ==> After about 5 token rotations, each master will check whether there is a new master between its own MPI address and the MPI address of the next master.

Maximum Retry Limit (MRL)

Maximum number of retries.

NOTE:

Please keep in mind that the values you enter here are interdependent (exception: MPI addresses). Changes should be made with caution.

Defaults

The entries are overwritten with the most commonly used values.

9.5.4 Device SSH Settings

The dialog box allows displaying and entering the SSH parameters (Secure Shell, network protocol for encrypted connections).

SSH allows you to establish an encrypted VPN connection to the device via the Internet. It also offers the possibility to reach the device through a firewall, usually via TCP/IP port 22.

In this way, you can access a connected PLC or parameterize the device.

SSH Base Settings**Use SSH**

Enables access using SSH.

SSH Port

Port number at which the device is accessed via the SSH connection.

Encryption Bits

Specifies the number of bits used for encryption. Normally 1024 bits are used.

The following rule applies: The higher the number of bits, the more secure the connection – and the slower the line speed. The default value of 1024 suits most purposes.

SSH User Accounts

Displays a list of authorized users.

At least one user and one password have to be entered here. The minimum length of the password is 6 characters.

New User, Delete User

Allows managing the user entries.

Edit User

Opens the Edit User dialog box where you can enter the user data.

NOTE:

SSH encryption must be enabled by the database administrator.

SSH Certificates

Currently not used.

9.6 Settings

This menu provides the following options:

H1 System Values

Opens the [H1 System Settings](#) dialog box where you can edit the H1 settings.

IP System Values

Opens the [TCP/IP System Settings](#) dialog box where you can edit the TCP/IP settings.

Delete Connections and Tables

Deletes all user entries and settings except for the station parameters in the [Station Parameters](#) dialog box for the existing network adapters. The device is reset but remains accessible via NetCon.

PLC System Values

Opens the [Device System Settings](#) dialog box where you can edit the PLC system settings.

Clock Settings

Opens the [Clock Settings](#) dialog box where you can change the clock settings.

File Upload Connections

Opens the [File Upload Connections](#) dialog box where you can change the server connections.
Modbus Table

Service Settings

Opens the [Device Service Settings](#) dialog box where you can edit the services as well as the access permissions for the services.

Firmware Update

This menu item is used for loading a firmware file to the device. If an update is required, you will receive a new firmware version from Softing Industrial Networks GmbH.

For compatibility the major and minor version number of the Firmware and the NetCon should match. See Firmwareupdate of echo devices.

Reboot

Use this menu item to reboot the device. All connections are closed.

Parameter in Flash

Here you can transfer the parameter settings to an external memory card inserted in the device. When the transfer is complete, the memory card should be removed and kept in a safe place. If necessary, you can then insert the card into the device and reboot it. The parameters are then loaded to the device.

9.6.1 H1 System Settings

The H1 system parameters represent the operating parameters of layer 4. The settings should only be changed in special cases. Please contact your system administrator before you make any changes to these settings.

The dialog box provides the following settings.

Connect Request Time Interval (fast)

This parameter specifies the time interval between two Connect Request attempts if the connection cannot be established. After the maximum number of fast CRs is reached, the slow CR time interval will be used.

Connect Request Time Interval (slow)

Here you specify the time interval between two Connect Request attempts if the connection cannot be established and the maximum number of fast CRs has been exceeded.

Number of Fast CRs

Here you define after how many failed Connect Request attempts the time interval between the attempts is increased so that the attempts are performed less frequently. This reduces the network load caused by unsuccessful attempts to establish connections. Whenever an established connection is closed, the system tries to reconnect.

Time Between Send Retries

Time interval after which a frame is retransmitted.

Time Until Connection Is Lost

Specifies the time after which a connection error is detected if the partner station no longer sends any frames. When this timeout expires, the connection has to be reestablished.

Timeout Between the Acknowledges

Specifies the time after which an acknowledgment is to be sent if no data traffic is taking place.

Maximum Credit

This parameter specifies how many frames the connection can buffer. With H1, this is usually 1.

Maximum Frame Length

Specifies the maximum length of a frame. The value is a power of two.

Class Options

The description for this parameter is provided in the OSI standard specification sheets.

Protocol Options

The description for this parameter is provided in the OSI standard specification sheets.

Timeout Wait

Not relevant for echo devices. (-1 is forever or until finished).

OK

The dialog box is closed and the data is applied.

Cancel

The dialog box is closed without applying the data.

Default

Resets all the input fields to the defaults.

9.6.2 TCP/IP System Settings

The TCP/IP system parameters represent the operating parameters in the TCP/IP core. The settings should only be changed in special cases. Please contact your system administrator before you make any changes to the settings.

Connect Request Time Interval (fast)

Short time interval between two CRs (Connect Requests).

The short time interval is used until the value set under Number of Fast CRs is reached.

Connect Request Time Interval (slow)

Long time interval between two CRs (Connect Requests). The long time interval is used after the value set under Number of Fast CRs was reached. This reduces the frequency of Connect Request attempts.

Number of Fast CRs

Specifies after how many failed Connect Request attempts the time interval between the attempts is increased so that the attempts are performed less frequently. As soon as the value specified here is reached, the Connect Request Time Interval (slow) is used between two Connect Request attempts. This allows reducing the network load.

Time Until Connection Is Lost

Time without data communication after which the connection is considered broken. Since TCP/IP as a WAN protocol normally does not use life acks, the connection is also broken if no data has been transmitted during the specified time.

Timeout Between the Acknowledges

Specifies the time after which an acknowledgment (ACK) is to be sent if no data traffic is taking place. For TCP/IP this value can be 0, which disables the life acks.

Time of the Acknowledge After Received Data

Received data is acknowledged after this time if no data frame has been returned. The shorter the time, the faster the data traffic, which is only in one direction.

Number of Retries

Specifies the number of retransmissions if no acknowledgment arrives from the receiver (confirming receipt of the data). When the specified number is reached, the connection is considered broken.

Timeout of an ARP Entry

Specifies how long entries in the ARP cache are valid if they have not been accessed. If an entry in the ARP cache is not accessed for a certain period of time, it will be removed. This behavior ensures that changes in the network can be detected and reflected by sending new ARP requests.

Timeout for DNS

Specifies the maximum time for converting a name to an IP address. When the timeout expires, the DNS request is retransmitted.

Time Between Two Life Data Acks

Time without data communication after which the connection is considered broken. Since TCP/IP as a WAN protocol normally does not use life data acks, the connection is also broken if no data has been transmitted during the specified time.

Maximum Frame Length

TPDU = Transport Protocol Data Unit

Specifies the maximum number of bytes that can be transmitted in a frame.

Start Number of the TCP Port Pool

Used for TCP connections in which a port has not been specified. If a port is parameterized as 0, a port number is generated. The numbers that are used start at the value given here.

Start Number of the UDP Port Pool

Used for UDP connections in which a port has not been specified. If a port is parameterized as 0, a port number is generated. The numbers that are used start at the value given here.

Factor for Timeout If No Header Is Used

All the timeouts given above are multiplied by this factor if no PLC header has been selected.

Factor for Timeout If RFC1006 Is Used

All the timeouts given above are multiplied by this factor if the RFC 1006 header has been selected.

Allow ICMP Broadcast

By default, echo modules ignore smurph attacks from the Internet (in other words, they do not reply). If ICMP broadcast frames are to be used in the LAN of a plant, you can select this check box to cause the echo module to reply.

Default

Resets all the input fields to the defaults.

9.6.3 Device System Settings

Any changes to these parameters will take effect immediately.

Free bytes of the S5 AP Ack aren't mirrored

In the acknowledgment frame of the S5 AP header, not all the bytes are relevant. With this check box, you can choose whether you want these bytes to be filled with the bytes of the request, or set to 0.

Use Active PG Function

Select this check box to enable the active PG function for S5 PLC access. This allows the programming via Ethernet by using a serially connected PG.

Reset Active PG

Use this button to reset the PG interface of the device without having to restart the PLC it is connected to. With this function you can reduce the connection timeout of several minutes after serial PG access.

Convert Protocol Start Address on Port(s):

If the echo device uses protocols of varying granularity, you can use this parameter to convert the address; e.g. for data blocks with S5 and S7 (word or byte oriented).

Reset Diagnostics

Resets all internal diagnostic counters to 0.

9.6.4 Diagnose of Default Connections

If you are using default connections, you can diagnose them here. Up to 7 default connections can be used (2 x port 990, 5 x port 991). Each default connection is displayed with the following parameters:

Type

Always IP because default connections are only available for TCP/IP.

Name of the Connection

The connection name is numbered from Default 1 to Default 7.

Job

The job numbers are assigned automatically.

Anzw

The display word in hexadecimal code

Anzw Text

The display word in plain text

Network

Displays the current status of the connection. The following messages can be displayed: OK, Wait for ACK, No connection

Send + Rec

Here the number of transmitted and/or received frames is displayed.

9.6.5 Clock Settings

How to Set the Clock

To set the internal clock, enter the hour, minute, second, day, month and year. Then click the Set button to apply the displayed values to the echo device. The day of the week is then shown automatically. The clock continues to run as long as no entries are made. Entries stop the refreshing of the display. If NTP is selected, the clock is set automatically.

Time Location

Select the correct time zone.

NTP Settings

Use NTP Service

Select this check box to obtain the time settings over the Internet.

Many institutions provide master clocks based on UTC time (Coordinated Universal Time). To provide the local time, select the corresponding time zone. In this case, the clock will also be adjusted to daylight saving time and take into account the leap seconds that occur occasionally at the end of a year.

URL

Here you enter the address of the NTP server.

Poll Time in Minutes

The interval at which time will be synchronized. The clock is often synchronized once an hour – 60 minutes.

9.6.6 Device Service Settings

In this dialog box, you can enable different services and edit the associated parameters.

Web Settings

Use Web Server

Enable or disable the web server in the device.

FTP Web User Name:/FTP Web Password:

This user account is only used for FTP access to the web pages of the web server in the echo device. Using this account, you can upload web pages and contents to the device for display in the web server. In addition, you can organize and delete web pages and contents. If no user and password is specified, FTP access to the web pages is denied.

Memory Card FTP Account

To access the SD card in the device from an FTP client, you can enter the FTP user and password here. This account provides access to log files created by the Logger as well as to Collect frame tables.

9.6.7 Reboot

With the **Reboot** menu item you can reboot the device via software (warm start). When doing so, you may be prompted for a reboot password. The reboot password is determined in the following way:

- The station password defined in the [Station Password](#) dialog box.

In devices with a version earlier than V7.0, the reboot password is determined in the following way:

- Open the **Help** menu and select **Versions**.
- In the Versions dialog box, the value **Version Param Server** is used as the password in the following format: xyyzzz.
- Example: 3.08, Build 22 ==> the password is 308022

During rebooting, all connections are closed and then reestablished.

9.7 Help

This menu provides the following options:

Help

Starts the [Help Overview](#) of the online help.

Versions

Opens the [Versions](#) dialog box displaying the versions of all the program modules of the device.

Device Licenses

Opens the [License Overview](#) dialog box where you can view and edit the device licenses.

About

Opens the [About](#) dialog box displaying the manufacturer and the versions of the NetCon software.

9.7.1 Versions

The dialog box shows the software versions of the echo device and all of its individual components.

Using the -> **Clipboard** button, you can copy the content to the clipboard and paste it into an e-mail or word processing application for further processing.

You will need these version details if you have any **questions** or require **support** for the product. In addition, you will need the hardware versions from the [Hardware Versions](#) dialog box and the NetCon versions from the [About](#) dialog box.

9.7.1.1 Hardware Versions

The dialog box shows the hardware versions of the echo device and all of its individual components.

Using the -> **Clipboard** button, you can copy the content to the clipboard and paste it into an e-mail or word processing application for further processing.

You will need these version details if you have any **questions** or require **support** for the product. In addition, you will need the software versions from the [Versions](#) dialog box and the NetCon versions from the [About](#) dialog box.

9.7.2 About

The dialog box shows the copyright, the versions of the NetCon configuration software, and the contact details for questions, suggestions or support.

You will need these version details if you have any **questions** or require **support** for the product. In addition, you will need the hardware versions from the [Hardware Versions](#) dialog box and the software versions from the [Versions](#) dialog box.

[License Conditions](#)

9.7.3 License Overview

Displays a list of the available modules with order number and license status. A function can be enabled by entering an activation code. This allows adding functionality to the device without having to return it to the manufacturer.

Show License

To edit an entry, double-click it or use the button. The [Licenses](#) dialog box appears. To edit licenses, master access is required.

Refresh List

This button reads the license status from the device and refreshes the displayed status.

9.7.4 Licenses

In the dialog box, you can enable the functionality of a software component provided in the device.

Please follow the steps below to license the product:

- Enter the **Company Name** and the **Customer Name** to make the license unique.
- The **Authorisation Code** is not relevant to echo devices. For OPC servers, please enter the authorization code that is given in the delivery note.
- Then click the **-> Clipboard** button. The data needed for requesting the license is copied to the clipboard and can be pasted into an e-mail or word processing application. For example:

```
-----  
Product       : COM5 MPI  
Order No      : 200-7120-01  
Customer      : Softing Industrial Networks GmbH  
User          : Dipl. Inform. Thomas Muster  
Authorization :  
Request       : 2E2380H0CH8SFMH  
Confirm       :  
=====
```

NOTE:

The License Request Code is unique for the device!

- Please send this data by fax to +49 911 54427-27 or e-mail it to info-in@softing.com.
- Softing Industrial Networks GmbH will send you a **License Confirm Code** for product activation.
- Enter the received code in the **License Confirm Code** field.
- The **License Status** field indicates whether the input is OK and the function is enabled.
- The device will only accept valid **License Confirm Codes**.

10 Collect Frame Menu

The [Collect Frame](#) window has its own menu providing the following menu items:

- [File](#)
- [Table](#)
- [Elements](#)
- [Online Diag](#)
- [Special Settings](#)
- [Edit Mode](#)
- [Help](#)

10.1 File

The File menu provides the following functions:

Export

Using this menu item, you can export the stored tables to a parameter file.

- Click this menu item to open the Windows dialog box: Write File...
- Select the desired file and click the **Save** button. The parameters are written to the file.

Import

This menu item is used for loading the tables from a parameter file into the device.

- Click this menu item to open the Windows dialog box: Open File...
- Select the desired file and click the **Open** button.
- A new dialog box appears: Collect Create Frame
- Here you can see a list of the tables that are contained in the file and are going to be created in the echo device.
- When you click the **OK** button, the parameters will be irreversibly overwritten in the device.

Refresh

Reestablishes the configuration connection to the device. If a password has been set, the password needs to be re-entered to allow editing.

Save

Applies the changes made to the tables and loads them to the device.

Save and Exit

Applies the changes to the tables, loads them to the device and closes the window.

Print

The selected table is printed on the system's default printer.

Cancel

Closes the configuration of the tables and switches to the [Connections](#) window. Changes made to the table are applied or discarded after a confirmation prompt.

10.2 Table

The Table menu provides the following functions:

Add Table

A new table is added to the [Collect Create Frame](#) dialog box.

Copy Table

A copy of the selected table is added to the [Collect Create Frame](#) dialog box. You can then rename the copied table.

Delete Table

The selected table is deleted after you confirm the prompt.

Rename Table

The selected table in the [Collect Create Frame](#) dialog box is displayed for renaming. You can then change the table name.

Switch On/Off

You can disable a table if you do not want it to be edited. Thus, it is possible to disable it temporarily without having to delete and redefine it.

10.2.1 Collect Create Frame

In the dialog box, you can set the properties of a table.

Collect Create Frame	
Frame Table Name	The file name can be entered or edited.
Frame Enabled	Deselecting this check box disables the table. In this case, it cannot be edited.
Flash Write Enabled	If the device has an external memory card, this check box enables writing to the card.
Select a Connection	The destination connection can be selected.
Add	A new connection can be entered, which can then be selected. The New Connection dialog box appears.

10.3 Elements

The Elements menu provides the following functions:

Add Element in Read Table

A new element is added to the read table in the [Collect Database Naming](#) or [Collect Add Frame Entry](#) dialog box.

Add Element in Write Table

A new element is added to the write table in the [Collect Add Frame Entry](#) dialog box. Please note that the write connection is only supported for database connections.

Add Hidden Trigger

A new element is added to the read table in the [Collect Add Frame Entry](#) dialog box. The hidden trigger is needed if you do not want the trigger value to be included in the transmitted frame.

10.3.1 Collect Edit Frame Entry

In the dialog box, you can add or edit a Collect frame entry. Depending on the entry type you select, the dialog box changes to allow setting the type-specific parameters.

Entry Type:

Topic / Item	Specifies a PLC connection and a PLC item. The item syntax is described in the Item Syntax chapter.
Frame Counter	All frames are sequentially numbered.
Static Text	Constant text
Symbolic Item	Specifies a PLC item stored in the symbol table. How to create symbols is described in the Collect Symbol Table Edit section.
Frame Timestamp	Time of frame generation.
Static Binary Data	Binary data

[Trigger](#) (for Topic/Item and Symbol)

Here you can choose if you want to use the entry as a trigger or if

- no trigger is to be used
- the trigger is to fire on value change, or
- the complex trigger is to be applied.

See also

[Special Syntax for Trigger/Normalisation](#).

[Normalisation](#) (for Topic/Item and Symbol)

No Normalisation

The value is used as it was created at the source.

Automatic Type Conversion

If binary data is generated (e.g. in order to write to the PLC), this setting automatically converts the data type. Example: A byte is read, but is to be written to a word element. If this setting is selected, the data type will be converted. If deselected, only a single byte will be transmitted, which would lead to an error. The setting should be deselected, for instance, if you fetch two single bytes from somewhere and want to write both to one word element. In this case, the automatic conversion would separately convert each byte to a word and thus generate 2 words, which would be incorrect.

Normalisation

Performed based on a special rule that can be entered in the [Collect Edit Normalisation Rule](#) dialog box.

Example: Basic arithmetic (e.g. *10, +4, /100, ...)

Bit masking

Boolean conversion (e.g. true = 5, false = 7)

Allows performing arithmetic or logical operations on the value.

See also

[Special Syntax for Trigger/Normalisation.](#)

Error Handling

Defines the value to be used in the case of an error (e.g. read error, connection interruption).

- Zero
- Empty value
- Last valid value

Counter setting (for Frame Counter):

Defines what will be counted

- Only this frame
- All frames for this (destination) connection
- All frames for this trigger source connection

Entry Enabled

The entry can be disabled. In this case, it will not be taken into account.

Raw Data Always New

If raw data is read, you can select this check box to force that each frame must contain new data and, if in doubt, wait for it to arrive. If the check box is not selected, the same raw data can be used for multiple frames (e.g. if raw data only comes when a value has changed somewhere). For trigger values, the device behaves as if this check box was permanently selected; the trigger only fires on receipt of new data.

Escape automatically

This option enables to escape single quote ' characters by duplication.

A database expects quoted text when handling non number data types. If the source data contains a single quote ' character and is forwarded 1:1 to a non number data type of the database the SQL statement will be invalid. With this option single quote characters are escaped resulting in a valid SQL statement.

10.3.1.1 Collect Browsing

CLX connections

For CLX connections, the items are read from the PLC and represented in a tree view.

Each branch can be expanded by clicking the small plus symbol "+" next to it.

For selectable items, the respective read/write access is displayed in the **Access** column.

The **Parameter** column shows the data type, while the Number column indicates the number of array elements.

If the PLC is switched off or not available, the list is empty.

Memory Table

The memory table shows the items that can be used.

10.3.1.2 Collect Edit Array Settings

Element Separator

Allows separating the array elements in the frame (e.g. a comma in SQL statements). The entry of a separator only makes sense for read items.

Array Length

Defines the length of an array and limits the amount of data, irrespective of the data length received or to be sent.

10.3.1.3 Collect Edit Date Entry

In the dialog box, you can select the desired timestamp format.

Length of Field

Specifies the number of characters output with the timestamp. It is recommended to use the Automatic setting. If a fixed length is entered, you can modify the Fitting and Boundary settings.

Generation Format

Specifies the format in which the timestamp will be created. Various ASCII and binary formats are provided for selection.

Display Format

Specifies the format in which the timestamp will be created. Various time bases are available.

Custom Date Format

The timestamp can be created from the predefined reserved characters between which any other ASCII characters can be included.

Example: "The frame was created on YYYY-MM-DD at hh:mm."

Unix Timestamp

A 32-bit value containing the number of seconds since 1970-01-01 00:00:00.

Unix Timestamp with Microseconds

A 32-bit floating point value containing the number of seconds since 1970-01-01 00:00:00 including the microseconds as decimal places.

Boundary

Indicates whether the text is left or right aligned. Can only be set if a fixed Length of Field is entered.

Fitting

Specifies the padding bytes.

10.3.2 Collect Edit Type Entry

Here you can make the following settings:

Length of Number

Fixed length or automatic, which is determined by the length at runtime.

Float Precision

The number of digits for floating point values (sum of digits before and after the decimal dot).

Use Comma

When deselected, a point is used.

Generation Format

ASCII

Converts binary PLC data to ASCII strings (e.g. for insertion into a database or text file). Here you can choose between decimal, hexadecimal and octal representation.

UTF8

Converts text to UTF8 format.

Binary

Passes binary PLC data in binary format (useful for generation of binary frames or for output of PLC strings). Depending on the type of representation, you can choose Motorola or Intel format.

Text File Index

Converts the value based on a text list (e.g. for error messages). In the [Collect Indexed Text List](#) dialog box, you can select the indexed text list you want to use.

Hidden (Trigger Only)

No output, the element is only used as a trigger.

Representation Format

Decimal, Hexadecimal or Octal

Fitting

The selected padding option is effective if a fixed length is set.

You can choose between blank space, decimal zero or binary zero for the padding.

Boundary

You can choose between left and right alignment.

Example 1:

Number = 4 (value, decimal)

Generation Format = Ascii

Representation Format = Hexadecimal

Output = "21"

Example 2:

Number = 4 (value, decimal)

Generation Format = Binary, Standard format

Output = 4 (as value)

10.3.3 Collect Indexed Text List

The dialog box allows selecting the indexed text lists.

The following dialog boxes are provided for editing:

- [Collect Indexed Text Table Edit](#)
- [Collect Indexed Text Edit](#)
- [Collect Add Indexed Text](#)

10.3.4 Collect Database Naming

In the dialog box, you can name the Collect database elements. If the echocollect device has an online connection to the database and the database is running, everything can be determined online. The >> buttons display the available parameters in a selection list.

The basic functions of the database, such as INSERT INTO or SELECT are provided in the drop-down list box.

In Table mode, you can also enter your own functions. This can be selected in the [Edit Mode](#) dialog box.

If there is no connection to the database or if the permissions granted in the database do not allow the reading of functions and tables, an error message is generated. In this case, get in touch with the database administrator. If the problem persists, you can manually make the entries for the Database, Table and Column Name fields.

Create Quotes For Values

If selected, the code generator will generate quotation marks for this entry. Otherwise, the setting selected in the [Collect Code Generator Settings](#) dialog box will be used.

10.3.5 Collect Browsing Databases - Tables

The dialog box displays the list of available tables of the database.

Element

Name of the table

If not all functions are displayed, the database permissions do not allow it.

Select the desired table and click **OK** to confirm.

10.3.6 Collect Browsing Databases - Columns

The dialog box displays the list of columns used in the selected table of the database.

Element

Name of the column

Description

Data type used in the database

Select the desired column and click **OK** to confirm.

10.3.7 Collect Browsing Databases - Functions/Procedures

The dialog box displays the available database functions and procedures.

Element

Names of the functions and procedures

Each function can be expanded to display the associated function parameters.

Parameter

Shows whether procedure or function. When expanded, the parameter / data type is displayed. With some databases, the data type cannot be determined; in this case, only the text parameter is displayed in the column.

If not all functions are displayed, the database permissions do not allow it.

Select the desired table and click **OK** to confirm.

10.4 Online Diag

The Online Diag menu provides the following functions:

Test Send

Using Test Send, you can trigger manually and transmit the configured frame. For example, you can use this function to test database commands without actively interfering with the system's PLC program.

Online Diagnostics

Switches the representation in the [Collect Frame](#) window. You can view the represented connections and entries together with the associated states and/or values.
Using this function, you can monitor the proper functioning of the table. From the color coding, you can see at a glance whether everything is OK or whether faults have occurred. Errors are displayed in **red print**.

Logger

Opens the [Logger](#) window where the recorded messages are displayed.
This window can be opened in parallel to the software application. You can continue to configure and use the software while the Logger window is open.

Logger Settings

Opens the [Logger Settings](#) dialog box. There you can select the events that will be recorded by the Logger. The file and memory settings for the Logger are made under Settings in the Logger window.

10.5 Special Settings

This menu item provides the following options:

PLC Confirm

Opens the [Collect PLC Confirm](#) dialog box where you can enter and edit the acknowledgment for an action.

Trigger Settings

Opens the [Collect Trigger Settings](#) dialog box where you can enter and edit the trigger for the selected table.

PLC Target

Opens the [Collect Memory Target](#) dialog box where you can enter and edit the destination in the PLC for the selected table.

Chain Table

Opens the [Collect Edit Frame Chaining](#) dialog box where you can link multiple tables into a chain.

Mail Target

Opens the [Select Address](#) dialog box where you can enter and edit the e-mail destination for the selected table.

Enable Write

Enables the Write function and allows entering a write job. Write jobs are only possible for database connections.

Collect Text Lists

Opens the [Collect Indexed Text Table Edit](#) dialog box for managing indexed text lists.

Collect Symbol Table

Opens the [Collect Symbol Table Edit](#) dialog box for managing the symbol table.

Collect Mail Address Book

Opens the [Collect Address Book](#) dialog box for managing entries in the e-mail address book.

Code Generator

Opens the [Collect Code Generator Settings](#) dialog box where you can edit the parameters for the SQL code generator.

Logger Settings

Opens the [Logger Settings](#) dialog box where you can select the Logger texts you want to log.

10.5.1 Collect PLC Confirm

An acknowledge is send to the PLC if the frame table was successfully executed. The frame table is successfully executed when the frame has been queued for transmission to the destination.

The acknowledgment is often used for resetting triggers, but can also be used for implementing a handshake.

The dialog box allows configuring the acknowledgment:

Use PLC Confirm

Enables or disables the PLC acknowledgment.

Entry Type

Allows selection from Topic/Item or an element of the symbol table.

Confirm Value

Allow selecting the value to be written to the respective acknowledgment element.

10.5.2 Collect Trigger Settings

In this dialog box, you can set the trigger settings; in other words, you can define when to execute a job.

Use Trigger Items

If the **Trigger** option has been enabled for the read items, you can enable and disable the use of the trigger items here.

With the radio buttons, you can specify whether to trigger as soon as one element changes or only after all elements have changed.

Trigger Combination

Here you can choose the logical operation to be performed on Trigger Items and Timeout. It defines whether the trigger event is initiated if one condition is true or if all conditions are true.

Use Timeout

If you want to trigger within a fixed time interval, you can enter a timeout in milliseconds here.

Use Custom Rule

Custom Rules can be defined for trigger and normalisation settings. While the standard rules can be simply selected from a list, it is also possible to create new rules as described in the [Special Syntax for Trigger/Normalisation](#) section and tailor them to your specific requirements.

10.5.3 Collect Edit Normalisation Rule

Normalisation allows you to convert and limit a value.

Normalisation

You can use the **Current Value** or the **Previous Value** for the calculation.

The list displays the sequence of calculations. The conversion is performed step by step. Entries are created with the **New** button and removed with **Delete**.

The conversion steps are calculated top down.

To move individual elements, click the **Up** and **Down** buttons.

The conversion is performed using the specified **Data Type**.

Add Automatic Typecast

If binary data is generated (e.g. in order to write to the PLC), this setting automatically converts the data type. Example: A byte is read, but is to be written to a word element. If this setting is selected, the data type will be converted. If deselected, only a single byte will be transmitted, which would lead to an error. The setting should be deselected, for instance, if you fetch two single bytes from somewhere and want to write both to one word element. In this case, the automatic conversion would separately convert each byte to a word and thus generate 2 words, which would be incorrect. If the check box is selected, the conversion is performed after the entire calculation has been completed, otherwise not.

Boolean Conversion

Boolean values can be represented in different formats and with different values.

Use Boundaries

The value range can be limited.

Custom Rule

If you want to create a nested calculation rule that cannot be defined using the other settings in this dialog box, you can define a Custom Rule for this purpose. The [Collect Edit Rule](#) dialog box facilitates the definition of custom rules. For details, please see the [Special Syntax for Trigger/Normalisation](#) section.

10.5.4 Collect Edit Rule

In the dialog box, you can define a custom rule. The settings you make in the top section of the dialog box are inserted at the cursor position in the input field at the bottom when you click the >> button.

The [Special Syntax for Trigger/Normalisation](#) section provides a detailed description of the syntax used for the calculation.

10.5.5 Collect Memory Target

In the dialog box, you specify where in the PLC you want the data to be transmitted to; in other words, you define the exact memory location.

Use Memory Target

Enables or disables writing to the memory target.

Item

The item is specified using the corresponding [item syntax](#).

10.5.6 Collect Edit Frame Chaining

Tables can be chained. In this dialog box, you can define if you want another table processed next, and choose which one.

Execute this frame next

Enables chaining.

>> button

Opens the [Collect Frames List](#) dialog box where you can select the next table to be processed.

10.5.6.1 Collect Frames List

This dialog box shows the list of existing tables. Select the desired table and click **OK** to confirm.

10.5.7 Select Address

In the dialog box, you can select to which recipients from the mail address book you want to send the e-mail.

Use Mail Targets

Select this check box if you want the e-mail to be sent.

Mail Subject

Here you can enter text that will be inserted into the subject line of the e-mail. If you leave the field blank, the table name will be inserted into the subject line.

Select Recipient Address

Displays the entries from the mail address book.

Each entry can be enabled or disabled.

10.5.8 Collect Indexed Text Table Edit

The Indexed Text function in echocollect allows the conversion of numbers to events or texts. These events can then be forwarded as text files, e.g. "1=All OK", "2=Machine stopped", etc. The dialog box shows the list of indexed text tables.

The **New List** and **Delete List** buttons are used for managing the text tables.

Edit List

Opens the [Collect Indexed Text Edit](#) dialog box where you can enter the index numbers for the selected list.

10.5.8.1 Collect Indexed Text Edit

The dialog box shows the texts of an indexed text table and allows creating, editing and deleting texts. To edit an entry in the list, double-click it.

List Name

This field displays the list name which can be changed here.

Add

Opens the [Collect Add Indexed Text](#) dialog box where you can add a text.

<- Delete

Deletes the selected entry from the list.

Edit

Opens the [Collect Add Indexed Text](#) dialog box where you can edit the selected entry.

Default Text

This text is output when no text has been entered for a number.

Use Number as Text

If no text has been entered for a number, the corresponding number is output instead of the default text.

10.5.8.2 Collect Add Indexed Text

In the dialog box, you can add or edit a text from the indexed text table. When you click **OK**, the software checks whether the number already exists. If it does, an error message appears.

10.5.9 Collect Symbol Table Edit

The dialog box shows the list of connections for which symbol tables have been created.

Topic

The list of symbol tables

Add

Opens the [Collect New Symbol Table](#) dialog box where you can add a symbol table.

Edit

Opens the [Collect Symbol Table Edit](#) dialog box for editing a symbol table.

Delete

Deletes the entire symbol table after you confirm the displayed prompt.

Load

Loads the symbols from a file. This does not have anything to do with the symbol import from a PLC.

Save

Saves the symbols to a file. This does not have anything to do with the symbol import from a PLC.

Importing symbols from PLC projects

To import symbols from projects in a PLC ([Siemens S7](#) and Rockwell CLX) use the button "Symbol Import".

10.5.9.1 Collect New Symbol Table

This dialog enables you to create a new symbol table.

New Topic

Here you can either enter a new topic (connection) or open the list box to select an existing topic. If you enter a new name, you should then create a connection of the same name to be able to use the symbol table.

10.5.9.2 Collect Symbol Table Edit

The dialog box shows the list of symbols for the connection displayed above. To edit an entry in the list, double-click it.

Connection Name

Name of the symbol table.

Symbol

The symbolic name by which the item is addressed.

Item

The absolute address in the PLC that can be entered using the correct [item syntax](#).

Add

Opens the [Collect Symbol Edit](#) dialog box where you can add a symbol. If the connection is an S7 connection, you can import the symbol from a PLC program by using the Symbol Import feature.

Edit

Opens the [Collect Symbol Edit](#) dialog box for editing the selected symbol.

Delete

The symbol is deleted without a confirmation prompt.

List of symbols

To import symbols from projects in a PLC ([Siemens S7](#) and Rockwell CLX) use the button "Symbol Import".

10.5.9.3 Collect Symbol Edit

This dialog enables you to create a new symbol. If the connection is an S7 connection, you can import the symbol from a PLC program by using the Symbol Import feature.

Connection Name

Name of the connection for which you want to create the symbol.

Symbol Name

The symbolic name by which the item is addressed.

Item

The absolute address in the PLC that can be entered using the correct [item syntax](#).

Data Type

If you want to perform a data type conversion, select the corresponding data type here; otherwise the **Unused** setting should be selected.

Array Length

If the element is an array, you can limit its length here.

Symbol Import

To import symbols from projects in a PLC ([Siemens S7](#) and Rockwell CLX) use the button "Symbol Import".

Comment

Any comment on the symbol.

10.5.9.4 S7 Symbol Import

In the dialog box, you can import a symbol from an existing S7 PLC project.

For this the symbols of the S7 project needs to be exported into a .sdfi file with the Softing dataFEED Exporter.

For this to work, do as follows:

1. Please launch the Softing dataFEED Exporter on the PC on which the SIEMENS TIA or Siemens STEP7 software with the project to import is installed.
2. Please follow the instructions within the Softing dataFEED Exporter.
3. Please choose the .sdfi file, which has been generated in the previous step in the Symbol File Import dialog of the NetCon.
4. Activate or deactivate the following checkboxes to your needs:
 - Clear symbol table before import:
Please activate this checkbox, if you like to replace all old symbols of the current connection with the symbols from the .sdfi file.
If you like to renew existing symbols and add new symbols from the .sdfi file, then please deactivate the checkbox. Symbols with the same name will be overwritten.
 - Resolve Array
If you like to address array elements with an own symbolic name, then please activate the checkbox. This option leads to a large increase of symbols. So the licence- and performance-limits could be reached. Therefor it is recommended, to enable this option only if it is really needed.
5. Please use the "OK" button to finalize the import.

10.5.10 Collect Address Book

The dialog box shows the list of stored e-mail addresses as an address book.

Up to 4 address book entries will be used.

To edit an entry in the list, double-click it.

Use the **Add** and **Delete** buttons to manage the entries.

Edit

Opens the [Collect Add Address Book Entry](#) dialog box where you can edit the selected entry.

In the [Mail Target](#) dialog box, you can select the addresses entered here.

10.5.10.1 Collect Add Address Book Entry

The dialog box enables you to add or edit an entry in the Mail Address Book

Name

Displayed for selection in other tables.

Address

Used for sending e-mails. Please only enter valid e-mail addresses.

Comment

The comment is only used internally for display.

10.5.11 Collect Code Generator Settings

This dialog box enables you to configure specific behavior of the SQL/CSV generator.

Generate Quotes Around SQL Names

If selected, all SQL names are enclosed in database-specific characters:

- MySQL, DB2: `name`
- MsSQL: [name]
- Oracle: "name"

This also allows the use of reserved symbols as names.

Generate Quotes Around SQL Values

If selected, all SQL values are enclosed in database-specific characters (see above). The generated value may look like this, for example: 'value'

- Always
- Never
- or Automatic (that is, if character strings or date values have been detected)

CSV Separator

The delimiter between values in generated CSV files.

Float Separator

Determines, for example, whether 3.14 or 3,14 will be used as the default value for new items. The setting can be changed separately for each item at any time.

The generated quotation marks and the CSV separator can be adjusted later in Expert mode, if required.

10.6 Edit Mode

The menu item toggles the representation of the [Collect Frame](#) window. It provides the following menu items:

Graphical

Switches the [Collect Frame](#) window to Graphical mode.

The mode provides assistance, browsing and syntax generator, but not all the functionality is available.

Table

Switches the [Collect Frame](#) window to Table mode.

The mode is designed for advanced users.

The system generator is disabled; you have to put commands together yourself and can make use of the full range of database commands.

Graphical Settings

Opens the [Collect Display Settings](#) dialog box where you can configure the settings for the Graphical mode.

10.6.1 Collect Display Settings

This dialog box enables you to modify the appearance of the Graphical display and the Table display of the [Collect Frame](#) window.

Straight Lines Instead of Cornered Lines	The lines connect the elements either directly or by horizontal and vertical lines.
Drop Syntax Generated Elements	If deselected, all syntax elements are displayed in the table view.
Show Items in PLC Symbol	
Show Destination Separate	
Save Window Positions	Saves the position and size of the Collect Frame window and displays the window at the same position when it is reopened.
Snap Size	Specifies how close the mouse pointer has to be to the elements on screen until touching them.
Max Text Display Len	Specifies how many text characters will be represented in the tables.

10.7 Help

Opens the help pages that assist you in defining [Collect Frame](#) tables.

11 Item Syntax

This chapter describes the item syntax that can be used for the creation of items (tags) for the respective connection (access path).

All names within the Item syntax may only contain letters and digits. All other characters like the point '.' have a special meaning (e.g. to split the names of an item). Therefore the usage of such special characters (e.g. '=') lead to unwanted results.

- [S7 Item Syntax](#)
- [S5 Item Syntax](#)
- [Modbus Item Syntax](#)
- [PLC-5 / SLC Item Syntax](#)
- [CLX Item Syntax](#)
- [MELSEC-Q Item Syntax](#)
- [Send/Receive Item Syntax](#)
- [Extended Raw Data Item Syntax](#)
- [OPCpipe Item Syntax](#)
- [Special Syntax for Trigger/Normalisation](#)
- [Item Syntax for Virtual Collect](#)

The following sections are generally applicable:

- [Arrays](#)
- [Suffixes](#)

11.1 S7 Item Syntax

The S7 item syntax has the following structure:

[Group.]<Area><DataType><StartAddress>[.ArraySize][Suffix]

If the data type is BOOL, the bit number must be specified:

[Group.]<Area><DataType><StartAddress><.BitNumber>[.ArraySize][Suffix]

Key: <> mandatory [] optional

<Area>

	Syntax	Orientation ¹	>Access rights	Notes
Data Blocks	DBx. V synonym for DB1	BYTE	Read / Write	Blocks require the entry of a block number x (x = 1 to 65535). The block number must be followed by a dot or comma.
Instance Data Block	Dlx.	BYTE	Read / Write	
Flag	M or F	BYTE	Read / Write	
Timer	T	WORD	Read	
Counter	Z or C	WORD	Read / Write	
Input	E or I	BYTE	Read	
Output	A or O or Q	BYTE	Read / Write	
Peripheral Input	PE or PI	BYTE	Read	
Peripheral Output	PA or PO or PQ	BYTE	Read / Write	

¹BYTE-oriented means that for each physical address one byte is addressed.

WORD-oriented means that for each physical address one word (16 bits) is addressed.

<DataType>

Type	Syntax	DB / DI	M	I/O	PI / PO	C/T	With array	Appropriate suffixes	OPC data type
BIT VT_BOOL	X	DB5.X4.3	MX1.3	-----	-----	-----	-----	-----	
BIT VT_BOOL		DB5.4.3	M1.3	E4.3 I4.3 A4.3 O4.3	PE4.5 PI4.5 PA1.3 PO1.3	-----	-----	-----	
BYTE VT_UI1	B Byte	DB5.B2 DB5.Byte 2	MB4 MByte4 FB4 FByte4	EB4 EByte4 IB4 IByte4 AB5 AByte5 OB5 OByte5	PEB4 PEByte4 PIB4 PIByte4 PAB5 PAByte5 POB5 POByte5	-----	DB5.B2.4 DB5.Byte 2.4 MB4.3 MByte4.3 FB4.4 FByte4.4 POB5.3 etc.	BA BCD RI RU	VT_BOOL VT_ARRAY VT_I2 VT_R4 VT_R4
WORD VT_UI2 VT_I4 (unsigned)	W Word	DB5.W3 DB5.Word d3	MW4 MWord4 FW4 FWord4	EW4 EWord4 IW4 IWord4 AW5 AWord5 OW5	PEW4 PEWord4 PIW4 PIWord4 PAW5 PAWord5 POW5	C5 Z5 T5	DB5.W3. 2 DB5.Wor d3.2 MW4.2 EWord4.2 PAWord5	BA BCD D DT DU KF KT	VT_BOOL VT_ARRAY VT_I4* / VT_UI2** VT_BSTR

				OWord5 QWord5	POWord5		.3 C5.3 Z5.10 T5.2 etc.	RI RU S5T SWAP TR	VT_BSTR VT_BSTR VT_I2 VT_BSTR VT_R4 VT_R4 VT_BSTR VT_I4* / VT_UI2** VT_R4
INT VT_I2	I Int	DB5.I3 DB5.Int3	MI4 MInt4 FI4 FInt4	EI4 EInt4 II4 IInt4 AI5 AInt5 OI5 OInt5	PEI4 PEInt4 PII4 PIInt4 PAI5 PAInt5 POI5 POInt5	-----	DB5.I3.2 DB5.Int3. 2 MI4.4 FInt4.3 AInt5.3 OI5.2 OInt5.5 POInt5.4 etc.	BA BCD DT DU KT RI RU S5T SWAP TR	VT_BOOL VT_ARRA Y VT_I4 VT_BSTR VT_BSTR VT_BSTR VT_R4 VT_R4 VT_BSTR VT_I4 VT_R4
REAL VT_R4	R Real	DB5.R2 DB5.REA L2	MR4 MREAL4	ER4 EReAL4 IR4 IREAL4 AR4 AREAL4 OR4 OREAL4	PER4 PEReAL4 PIR4 PIREAL4 PAR4 PAREAL4 POR4 POREAL 4	----	DB5.R2.2 DB5.REA L2.4 MR4.5 ER4.4 AREAL4. 5 PER4.2 PAR4.7 POR4.3 etc.	-----	-----
STRING VT_BSTR	S **** String	DB5.S1.2 DB5.Strin g1.2	MS2.3 MString2. 3	-----	-----	-----	-----	KA	VT_BSTR
S7- STRING VT_BSTR	G ****	DB5.G2.2	MG2.3	-----	-----	-----	-----	KA	VT_BSTR
DOUBLE WORD VT_R8 VT_UI4 (unsigned)	DW DWord	DB5.DW3 DB5.DWo rd3	MDW4 MDWord4 FDW4 FDWord4	EDW4 EDWord4 IDW4 IDWord4 ADW5 ADWord5 ODW5 ODWord5 QDW5	PEDW4 PEDWord 4 PIDW4 PIDWord4 PADW5 PADWord 5 PODW5 PODWord 5	-----	DB5.DW3 MDW4.2 FDW4.2 EDW4.4 ADWord5 .8 PEDW4.8 PADW5.4 PODW5.4 etc.	BA BCD KF KG RI RU S5T SWAP T TOD	VT_BOOL VT_ARRA Y VT_I4 VT_I4 VT_R4 VT_R4 VT_R4 VT_I4 VT_R4 VT_BSTR VT_BSTR
DOUBLE INT VT_I4	D DI DInt	DB5.D3** * DB5.DI3 DB5.DInt3	MDI4 MDInt4 FDI4 FDInt4	EDI4 EDInt4 IDI4 IDInt4	PEDI4 PEDInt4 PIDI4 PIDInt4	-----	DB5.D3.2 DB5.DI3.2 DB5.DInt3 .4	BA BCD KG RI	VT_BOOL VT_ARRA Y

				ADI5 ADInt5 ODI5 ODInt5 QDI5	PADI5 PADInt5 PODI5 PODInt5		MDI4.5 EDI4.5 ADI5.2 PEDI4.5 PADInt5.2 etc.	RU SWAP T TOD	VT_I4 VT_R4 VT_R4 VT_R4 VT_I4 VT_BSTR VT_BSTR
--	--	--	--	--	--------------------------------------	--	--	------------------------	---

- * If "Standard Types only" is selected under Data Types (server settings)
- ** If "Unsigned Types also" is selected under Data Types (server settings)
- *** Caution! Not to be confused with the S5 syntax for bit (DB5D1.1)
- **** SF can be used to force an S5 string, and SS to force an S7 string

Notes on counters and timers

Counters and timers are always addressed as words. Therefore, no data type needs to be specified! The "T" or "C/Z" area is followed directly by the start address. Timers are read-only! Counters can be read and written to.

Timer values are displayed in seconds; example: T = 0.7 => T = 0.7s = 700ms.

Counters are represented in decimal notation (0 to 999).

<StartAddress>

The start address indicates the address at which reading or writing will start.

Example: DB5.DW6: Double word 6 of data block 5 is the start address

Example: MB17: Flag byte 17 is the start address

If the start address is a specific bit, the bit number must be indicated, as well.

<.BitNumber>

The bit number must always be specified if the data type is BOOL.

Example: I4.3: Bit 3 of input byte 4 – here an input bit is addressed

Example: MX12.1: Bit 1 of flag byte 12 – here a flag bit is addressed

[.ArraySize]

An array is a series of identical elements. An array combines multiple units of the same data type to form a single unit. For example: If you would like to read multiple words from a data block, this will be an array of words. To create an array, append the length of the array to the standard syntax, separated by a dot.

Example: DB10.REAL2.5.

More information about arrays is provided [here](#).

[Suffix]

A suffix allows representing a value in a different format. To find out more about suffixes, click [here](#).

11.2 S5 Item Syntax

The S5 item syntax has the following structure:

```
[Group.]<Area><DataType><StartAddress>[.ArraySize][Suffix]
```

If the data type is BOOL, the bit number must be specified:

```
[Group.]<Area><DataType><StartAddress><.BitNumber>[.ArraySize][Suffix]
```

Key: <> mandatory [] optional

[Group.]

Operand areas can be grouped. If the periphery contains gaps (e.g. A10.1 and A10.3), these gaps can be skipped by forming different groups. In this case, the gap will not be included in the request. For this purpose, specify the group name followed by a dot and the item name (GroupName.ItemName). The group name is composed of "G" and the number.

Example:

G1.A10.1

G2.A10.3

The gap is skipped. The items are not combined, but requested individually.

<Area>

	Syntax	Orientation ¹	Access rights	Notes
Data Blocks	DBx.	WORD	Read / Write	Blocks require the entry of a block number x (x = 1 to 65535).
Extended Block	DXx.	WORD	Read / Write	
Flag	M or F	BYTE	Read / Write	
Timer	T	WORD	Read	
Counter	Z or C	WORD	Read	
Input	E or I	BYTE	Read	
Output	A or O or Q	BYTE	Read / Write	
Peripheral	P	BYTE	Read / Write	
Extended Peripheral	OB	BYTE	Read / Write	
System Area		WORD	Read	
Absolute Addressed Memory Cells	AS	WORD	Read	

¹BYTE-oriented means that for each physical address one byte is addressed.

WORD-oriented means that for each physical address one word (16 bits) is addressed.

<DataType> for data blocks and extended data blocks

	Syntax	Example	With array	Appropriate suffixes	OPC data type
BIT VT_BOOL	D	DB5D4.12	-----	-----	-----
BYTE VT_UI1	DB	DB5DB3	DB5DB3.5	BA	VT_BOOL VT_ARRAY
LEFT BYTE VT_UI1	DL	DB5DL4	DB5DL4.2	BA	VT_BOOL VT_ARRAY
RIGHT BYTE VT_UI1	DR	DB5DR2	DB5DR2.5	BA	VT_BOOL VT_ARRAY
WORD VT_UI2** VT_I4(*)	DW	DB5DW4	DB5DW4.5	BA BCD D DT DU KF KT RI RU S5T SWAP TR	VT_BOOL VT_ARRAY VT_I4* / VT_UI2** VT_BSTR VT_BSTR VT_BSTR VT_I2 VT_BSTR VT_R4 VT_R4 VT_BSTR VT_I4(*) / VT_UI2** VT_R4
DOUBLE WORD VT_R4(*) VT_UI4**	DD	DB5DD3	DB5DD3.2	BA BCD IEEE KF KG RI RU SWAP T TOD	VT_BOOL VT_ARRAY VT_I4 VT_R4 VT_I4 VT_R4 VT_R4 VT_R4 VT_R4 VT_R4 VT_I4 VT_BSTR VT_BSTR
S7-STRING VT_BSTR	S	DB5S2.3	-----	-----	-----

* If "Standard Types only" is selected under Data Types (server settings)

** If "Unsigned Types also" is selected under Data Types (server settings)

<DataType> for all other areas

	Syntax	M	I/O	P / OB	C / T / RS / AS	With array	Appropria te suffices	OPC data type
BIT VT_BOOL		M4.3 F4.3	E4.3 I4.3 A4.3 Q4.3	-----	-----	-----	-----	-----
BYTE VT_UI1	B	MB4 FB4	EB4 IB4 AB5 QB5	PB4 OB4	-----	MB4.3 FB4.4 EB4.4 IB4.5 AB5.2 QB5.5 PB4.2 OB4.3	-----	-----
WORD VT_UI2 VT_I4	W	MW4 FW4	EW4 IW4 AW5 QW5	PW2 OW2	C5 T5 RS4 BS4 AS5	MW4.2 FW4.4 EW4.5 IW4.2 AW5.5 QW5.2 PW2.10 OW2.3 C5.5 T5.3 RS4.2 BS4.7 AS5.2	BA BCD D DT DU KF KT RI RU S5T SWAP TR	VT_BOOL VT_ARRA Y VT_I4* / VT_UI2** VT_BSTR VT_BSTR VT_BSTR VT_I2 VT_BSTR VT_R4 VT_R4 VT_BSTR VT_I4* / VT_UI2** VT_R4
DOUBLE VT_R4(*) VT_UI4**	D	MD4 FD4	ED4 ID4 AD5 QD5	PD5 OD5	-----	MD4.4 FD4.2 ED4.6 ID4.2 AD5.6 QD5.2 PD5.4 OD5.9	BA BCD IEEE KF KG RI RU SWAP T TOD	VT_BOOL VT_ARRA Y VT_I4 VT_R4 VT_I4 VT_R4 VT_R4 VT_R4 VT_I4 VT_BSTR VT_BSTR
STRING VT_BSTR	S	MS4.5 FS4.5	-----	-----	-----	-----		-----

* If "Standard Types only" is selected under Data Types (server settings)

** If "Unsigned Types also" is selected under Data Types (server settings)

Notes on counters and timers

Counters and timers are always addressed as words. Therefore, no data type needs to be specified! The “T” or “C/Z” area is followed directly by the start address. Timers are read-only! Counters can be read and written to.

Timer values are displayed in seconds; example: T = 0.7 => T = 0.7s = 700ms.

Counters are represented in decimal notation (0 to 999).

<StartAddress>

The start address indicates the address at which reading or writing will start.

Example: DB5DD6: Double word 6 of data block 5 is the start address

Example: MB17: Flag byte 17 is the start address

If the start address is a specific bit, the bit number must be indicated, as well.

<.BitNumber>

The bit number must always be specified if the data type is BOOL.

Example: I4.3: Bit 3 of input byte 4: here an input bit is addressed

Example: M12.1: Bit 1 of flag byte 12: here a flag bit is addressed

[.ArraySize]

An array is a series of identical elements. An array combines multiple units of the same data type to form a single unit. For example: If you would like to read multiple words from a data block, this will be an array of words. To create an array, append the length of the array to the standard syntax, separated by a dot.

Example: DB10DW3.4.

More information about arrays is provided [here](#).

[Suffix]

A suffix allows representing a value in a different format. Without a suffix, the S5 syntax returns an unsigned data type. To receive a signed data type, a suffix must be specified.

Example: DB10DW3.4BCD.

More information about suffixes is provided [here](#).

11.3 Modbus Item Syntax

The Modbus item syntax has the following structure:

```
[Group.]<Area><DataType><StartAddress>[.ArraySize][Suffix]
```

If the data type is BOOL, the bit number must be specified:

```
[Group.]<Area><DataType><StartAddress><.BitNumber>[.ArraySize][Suffix]
```

If the option to use the Unit Identifier individually with the item syntax was selected for the connection, the following structure will be used:

```
[UnitID.][Group.]<Area><DataType><StartAddress><.BitNumber>[.ArraySize][Suffix]
```

Key: <> mandatory [] optional

[Group.]

Operand areas can be grouped.

Area: **0 - 65535**. If the option to use the Unit Identifier individually with the item syntax was selected, the area is reduced to **0 - 255**.

If the periphery contains gaps, the gaps can be skipped by forming different groups. In this case, the gap will not be included in the request.

For this purpose, specify the group name followed by a dot and the item name (GroupName.ItemName).

The group name is composed of G and the number.

Example:

G1.40001

G2.R2

G3.S20.30

[UnitID.]

The Unit Identifier for this item is indicated in the item syntax.

For this purpose, specify the UnitID name followed by a dot and the item name (UnitIDName.ItemName).

The unit ID name is composed of the ID and the number.

Area: **0 - 255**.

If you specify a group, the area of the group number is reduced to 0 - 255.

If the Unit ID is missing in the item syntax, the parameterized Unit ID is transmitted to the PLC.

Example:

Id1.40001

Id2.R2

Id3.S20.30

Id1.G2.R3 UnitId + Group

<Area>

	Syntax mnemonics	Mnemonic number	Orientation*	Access rights
Discrete Inputs	I E DI DE	1xxxxx	BIT	Read
Discrete Outputs	A O Q DA DO DQ	0xxxxx	BIT	Read / Write
Input Register	ER IR	3xxxxx	WORD	Read
Register (Holding Register)	R HR	4xxxxx	WORD	Read / Write
Discrete Inputs Octal **	J	-	BIT	Read
Discrete Outputs Octal **	P	-	BIT	Read / Write

* BIT-oriented means that for each physical address one bit is addressed.

WORD-oriented means that for each physical address one word (16 bits) is addressed.

** The start address is entered in octal notation; as a result, the numbers 8 and 9 are invalid. The address is internally managed in decimal notation and must be taken into account in the Logger and the status.

Discrete inputs and outputs occupy 1 bit in the PLC. They are treated as an 8-bit value in read and write operations. **This means, that writing of single bits is not supported.**

Areas can be addressed using either the above character string or a number. In other words, to address a discrete input you can use the abbreviation "I" or the number "1".

<DataType>

	Syntax	Syntax R	Syntax I / O	Syntax ER	With array	Appropriate suffixes	OPC data type
BIT VT_BOOL	X	RX5.2 HRX5.2 4X5.2	E255 I255 DE255 DI255 125543	---	---	---	---
INT**** VT_I2 (signed)	None I	R50 HR50 400050 RI50 HRI50 4I50	----	ER120 IR120 3I2034 ERI120 IRI120 3I12034	R50.2 HR50.2 400050.2 ER120.2 IR120.2 3I2034.2	BCD D WDT* KT RI RU S5T SWAP BA	VT_I2 VT_BSTR VT_BSTR VT_BSTR VT_R4 VT_R4 VT_BSTR VT_I2 VT_BOOL VT_ARRAY
WORD**** VT_UI2 (unsigned)	W	RW50 HRW50 4W50	----	ERW120 IRW120 3W12034	RW50.2 HRW50.2 4W00050.2 ERW120.2 IRW120.2 3W12034.2	BCD D WDT* KT RI RU S5T SWAP BA	VT_I2 VT_BSTR VT_BSTR VT_BSTR VT_R4 VT_R4 VT_BSTR VT_I2 VT_BOOL VT_ARRAY
DOUBLE INT**** VT_I4 (signed)	D DI	RD50 HRD50 4D50 RDI50 HRDI50 4DI50	----	ERD120 IRD120 3D12034 ERDI120 IRDI120 3DI12034	RD50.2 HRD50.2 4D00050.2 ERD120.2 IRD120.2 3D12034.2	BCD D WDT* KT RI RU S5T SWAP BA	VT_I2 VT_BSTR VT_BSTR VT_BSTR VT_R4 VT_R4 VT_BSTR VT_I2 VT_BOOL VT_ARRAY
DOUBLE WORD**** VT_UI4 (unsigned)	DW	RDW50 HRDW50 4DW50	----	ERDW120 IRDW120 3DW12034	RDW50.3 HRDW50.3 4DW50.3 ERDW120.3 IRDW120.3 3DW12034.3	BCD SWAP KG IEEE TR RI RU T WDT*** BA	VT_I4 VT_I4 VT_R4 VT_R4 VT_R4 VT_R4 VT_R4 VT_BSTR VT_BSTR VT_BOOL VT_ARRAY
REAL VT_R4	R	RR5	----	ERR5	RR5.3 ERR5.2	---	---
STRING VT_BSTR	S	RS5.4	----	ERS5.4	---	KA SWAP	VT_BSTR VT_BSTR

* If "Standard Types only" is selected under Data Types (server settings)

** If "Unsigned Types also" is selected under Data Types (server settings)

*** If the suffix WDT (Wago Date and Time) is specified, 4 registers (8 bytes) will be read

**** It is manufacturer-specific, which byte is the first and second in the word / double-word. There exists the possibility to create two connections: (1) One for the Word/DWord access and use the Option "word swap". (2) The second one for the byte access, using the option "byte swap".

<StartAddress>

The start address indicates the address at which reading or writing will start.

Example: ER120: Input Register 120

If the start address is a specific bit, the bit number must be indicated, as well.

<.BitNumber>

The bit number must always be specified if the data type is BOOL.

Example: HRX5.2: Bit 2 of Holding Register 5

[.ArraySize]

Arrays are formed to combine multiple units of the same data type to a single unit.

Example: HRD50.3 [More about arrays](#)

[Suffix]

A suffix allows representing a value in a different format.

Example: HRD50.3WDT (representation in Wago Date and Time Format)

[More about suffixes](#)

Strings

To support strings within a Modbus PLC, we allow to interpret a range of Modbus registers as a string with 8 bit US-ASCII encoding. For example the syntax RS100.8 maps the modbus registers 100 to 108 to a string with 8 8-bit characters.

11.4 PLC-5 / SLC Item Syntax

The item syntax for PLC-5 and SLC has the following structure:

```
<Area>[FileNumber]<StartAddress>[.ArraySize][Suffix]
```

Key: <> mandatory [] optional

<Area>

File Type	Syntax	Orientation*	Access rights	Default file number	Address format
Output	O	BIT	Read / Write	0	Octal
Input	I	BIT	Read	1	Octal
Integer	N	WORD	Read / Write	7	Decimal
Binary	B	WORD	Read / Write	3	Decimal
Float	F	DOUBLE WORD	Read / Write	8	Decimal
String	ST	SLC string	Read / Write	3	Decimal
SFC Status	SC	WORD	Read / Write	3	Decimal
Status	S	WORD	Read / Write	2	Decimal
Timer	T	WORD	Read / Write	4	Decimal
Counter	C	WORD	Read / Write	5	Decimal
Control	R	WORD	Read / Write	6	Decimal
ASCII	A	WORD	Read / Write	3	Decimal

* BIT-oriented means that for each physical address one bit is addressed.

WORD-oriented means that for each physical address one word (16 bits) is addressed.

DOUBLE WORD-oriented means that for each physical address one double word (32 bits) is addressed.

[FileNumber]

The file number is optional. If it is not specified, the default file number is used. See the **Default file number** column in the table.

<StartAddress>

The start address indicates the address at which reading or writing will start. It can comprise 2 constituents: Wort (float) number and, if a single bit is addressed, the bit number. When addressing a bit, the word number can be omitted. The word number or float number is initiated by a colon (:). The bit number is preceded by a slash (/) <:word> or <:float> or </bit>. After the colon, you can specify a format. If no format is specified, the data format from the **Orientation** column is applied.

Format	Syntax	Item format
String	S	STRING
High Byte	H	WORD
Low Byte	L	WORD
Byte	B	WORD

The addresses are indicated in octal notation for some areas and in decimal notation for others. See column **Address format**

For the start address, indicate either the word number, the float number or the bit number:

<:word> or <:float> or </bit>

To address a specific bit within a word, use the following syntax:

<:word/bit>

NOTE:

When writing bits, the whole word is written!

Syntax	Description
O:0	Word 0 in output file 0
O:0/12	Bit 10 (12 octal = 10 decimal) in output file 0
O/12	Bit 10 (12 octal = 10 decimal) in output file 0
I:37	Word 31 (37 octal = 31 decimal) in input file 1
I4:37/2	Bit 2 in word 31 ((37 octal = 31 decimal) in input file 4
I:1/0	Bit 0 in word 1 in input file 1
B3/26	Bit 26 in binary file 3
B12:5.15	Array of 15 words in binary file 12, starting at word 5
B12:5/15	Bit 15 in word 5 of binary file 12
F8:0	Float 0
F9:10.16	Array of 16 double words in float file 9, starting at double word 10
N23:4	Word 4 of integer file 23
N23:4.10	Array of 10 words in integer file 23, starting at word 4
N23:4/2	Bit 2 in word 4 of integer file 23 = bit 66 in integer file 23
N23/66	Bit 66 in integer file 23

[.ArraySize]

Arrays are formed to combine multiple units of the same data type to a single unit. Arrays are only possible for word areas and float areas.

Examples: N23:4.10

[Suffix]

A suffix allows representing a value in a different format. If no suffix is specified, the formats from the **Orientation** column are applied.

[More about suffixes](#)

11.5 CLX Item Syntax

The following item syntax is required for CLX.

Reading from the CLX

The CLX item syntax for reading has the following structure:

```
<TagName>
```

If the data type is **array**, the CLX item syntax has the following structure:

```
<TagName>[DataType,StartAddress.Length]
```

If the data type is **string**, the CLX item syntax has the following structure:

```
<TagName><DataType>
```

Key: <> mandatory [] optional

Writing to the CLX

When writing, the syntax must always include the data type.

If the item is a **string** or a **simple item (not an array)**, the following item syntax is required:

```
<TagName><DataType>
```

When writing to **arrays**, the CLX item syntax has the following structure:

```
<TagName><DataType><,StartAddress><.Length>
```

Key: <> mandatory [] optional

<DataType>

Type	Syntax
BIT	X
INT8 UINT8	c B
INT16 UINT16	I W
INT32 UINT32	DI DW
REAL32	R
STRING	S
INT64 UINT64	QI QW

<DataType>,<StartAddress>.<Length>

The **start address** indicates the address at which reading or writing will start in an array. If no **length** is specified, only a single element will be addressed.

ATTENTION:

The data type, the start address and the length must be given in square brackets.

Array element:

Example for reading: VAR1[C,0.10]: 10 elements are read from VAR1 (an array of type INT8), starting at start address 0 (index of the array).

Example for writing: VAR2[I,5.8]: 8 elements are written to VAR2 (an array of type INT16), starting at start address 5 (index of the array).

No array element:

Example for reading: VAR1: VAR1 is read, which contains only a single value.
Example for writing: VAR2[R]: A value is written to VAR2, which of type REAL (32 bits).

BOOL ARRAY:

In a Boolean array, only one bit is written or read, starting at a start address.

Example for writing: VAR3[x,15]: VAR3 is a Boolean array; the start address 15 (index of the array) is set (TRUE) or reset (FALSE).
VAR3[x,29]: VAR3 is a Boolean array; the start address 29 (index of the array) is set (TRUE) or reset (FALSE).

String:

Example for reading strings: VAR1[s]: The string is read.
Example for writing to strings: VAR2[s]: The VAR2 character is written to.

NOTE:

Structure arrays and string arrays are not supported. Access to Bool Arrays is not supported, only the Access to one element of a Bool Array

11.6 MELSEC-Q Item Syntax

You can choose between two different syntax variants for creating items:

1. Simple variant:

```
[Group.]<Area><StartAddress>[.ArraySize][Suffix]
```

2. Extended variant:

```
[Group.]<Area><.Type><StartAddress>[.ArraySize][Suffix]
```

Key: <> mandatory [] optional

PLEASE NOTE:

- When using the extended variant, a dot or comma must be entered between the <Area> and the <Type>. If the dot or comma is missing, the syntax will correspond to the syntax of the simple variant. The <Type> will then be the default type (which is BIT for bit areas and WORD for word areas – see <Area> table).
- If the start address of the area is represented in HEX, all numerical entries for this area will also be in HEX
- You can change between the number formats HEX and Decimal by using the following prefixes:
Conversion HEX -> DEC: Enter 0d (number zero + letter d) followed by the decimal address
Conversion DEC -> HEX: Enter 0x (number 0 + letter x) followed by the hexadecimal address
- If words, double words or strings are registered in a bit area, the start address is a bit address and only possible at word boundaries (DEC: 0/16/32... or HEX: 0/10/20...)

Example: Y.D10.3

- **The writing of single bits and bytes is not supported!**
- Bit masking is possible. The bit mask is defined before the suffix (see Bit Masking).
- Bit arrays in bit areas are not possible.
- Bit arrays in WORD areas with HEX representation, are not possible because of the suffix BA (BitArray) is interpreted as a hexadecimal number.

[Group.]

Areas can be divided into groups of consecutive addresses.

If the periphery contains gaps, the gaps can be skipped by forming groups for reading and writing. The addresses before the gap are combined to one group, the addresses after the gap form another group. In this way, the gap will not be included in the request.

To achieve this, enter the letter G and the group number followed by a dot and the item name (GNo.ItemName)

Example:

G1.Y10

G2.Y30

Areas in different groups are not combined, but requested individually.

<Area>

	Syntax	Orientation*	Representation of the start address
Special Relay	SM	BIT	Decimal
Special Register	SD	WORD	Decimal
Input Relay	X	BIT	HEX
Output Relay	Y	BIT	HEX
Internal Relay	M	BIT	Decimal
Latch Relay	L	BIT	Decimal
Annunciator	F	BIT	Decimal
Edge Relay	V	BIT	Decimal

Link Relay	B	BIT	HEX
Data Register	D	WORD	Decimal
Link Register	W	WORD	HEX
Timer Contact	TS	BIT	Decimal
Timer Coil	TC	BIT	Decimal
Timer Current Value	TN	WORD	Decimal
Retentive Timer Contact	SS	BIT	Decimal
Retentive Timer Coil	SC	BIT	Decimal
Retentive Timer Current Value	SN	WORD	Decimal
Counter Contact	CS	BIT	Decimal
Counter Coil	CC	BIT	Decimal
Counter Current Value	CN	WORD	Decimal
Special Link Relay	SB	BIT	HEX
Special Link Register	SW	WORD	HEX
Step Relay	S	BIT	Decimal
Direct Input	DX	BIT	HEX
Direct Output	DY	BIT	HEX
Index Register	Z	WORD	Decimal
File Register (Normal Access by Block Switching)	R	WORD	Decimal
File Register (Serial No. Access)	ZR	WORD	Decimal

* BIT-oriented means that for each physical address one bit is addressed. WORD-oriented means that for each physical address one word (16 bits) is addressed.

<.Type> <.Type>

	Syntax	Simple syntax	Example for bit area	Example for word area	With array bit area	With array word area	Appropriate suffixes	OPC data type
BIT VT_BOOL (readonly)	X	DY1	----	D.X1.2	----	----	----	----
BIT**** VT_BOOL (readonly)	----	DY1	----	D.1.2	----	----	----	----
BYTE VT_UI2 (readonly)	B BYTE	----	Y.B10 Y.BYTE10	R.B1 R.BYTE1	Y.B10.5 Y.BYTE10.5	R.B1.3 R.BYTE1.3	----	----
INT VT_I2	I INT	Y.I10 Y.INT10	Y.I10 Y.INT10	R. I 2 R. INT 2	Y.I10.3 Y.INT10.3	R.I2.3 R.INT2.3	BCD D WDT* KT RI RU S5T SWAP BA	VT_I2 VT_BSTR VT_BSTR VT_BSTR VT_R4 VT_R4 VT_BSTR VT_I2 VT_BOOL VT_ARRAY
WORD VT_UI2	W WORD	R20	Y.W10 Y.WORD	R.W2 R.WORD2	Y.W10.3 Y.WORD	R.W2.3 R.WORD2.	BCD D	VT_I2 VT_BSTR

			10		10.3	3	WDT* KT RI RU S5T SWAP BA	VT_BSTR VT_BSTR VT_R4 VT_R4 VT_BSTR VT_I2 VT_BOOL VT_ARRAY
DOUBLE WORD VT_UI4	D DW DWORD	----	Y.D10 Y.DWORD D10	R.D2 R.DWORD 2	Y.D10.3 Y.DWORD D10.3	R.D2.3 R.DWORD 2.3	BCD SWAP KG IEEE TR RI RU T WDT*** BA	VT_I4 VT_I4 VT_R4 VT_R4 VT_R4 VT_R4 VT_R4 VT_BSTR VT_BSTR VT_BOOL VT_ARRAY
DOUBLE INT VT_I4	DI DINT	----	Y.DI10 Y.DINT10	R.DI2 R.DINT2	Y.DI10.3 Y.DINT10 .3	R.DI2.3 R.DINT2.3	BCD SWAP KG IEEE TR RI RU T WDT*** BA	VT_I4 VT_I4 VT_R4 VT_R4 VT_R4 VT_R4 VT_R4 VT_BSTR VT_BSTR VT_BOOL VT_ARRAY
REAL VT_R4	R REAL	----	Y.R10 Y.REAL1 0	R.R2 R.REAL2	Y.R10.3 Y.REAL1 0.3	R.R2.3 R.REAL2.3	----	----
STRING VT_BSTR	S STRING	----	Y.S10.20 Y.STRING G10.20	R.S2 R.STRING 2.20	----	----	KA SWAP	VT_BSTR VT_BSTR

* If "Standard Types only" is selected under Data Types (server settings)

** If "Unsigned Types also" is selected under Data Types (server settings)

*** If the suffix WDT (Wago Date and Time) is specified, 4 registers (8 bytes) will be read

**** Caution: For HEX addresses, it is recommended to choose the variant with the X

[.ArraySize]

Arrays are formed to combine multiple units of the same data type to a single unit.

Examples:

X10.5

D20.300

[More about arrays](#)

[Suffix]

A suffix allows representing a value in a different format.

Example: D20.300KF

[More about suffixes](#)

11.7 Send/Receive Item Syntax

The Send/Receive item syntax has the following structure:

<Job><DataType><StartAddress>[.ArraySize][Suffix]

If the data type is BOOL, the bit number must be specified:

<Job><DataType><StartAddress><.BitNumber>[.ArraySize][Suffix]

Key: <> mandatory [] optional

<Job>

	Syntax
SEND	S
RECEIVE	R

<DataType>

	Syntax	Syntax Send	Syntax Receive	With array	Appropriate suffixes	OPC data type
BIT VT_BOOL	X	SX1.2	RX1.2	----	----	----
BIT VT_BOOL	----	S1.2	R1.2	----	----	----
BYTE VT_UI1	B BYTE	SB4 SBYTE4	RB5 RBYTE5	SB4.4 SBYTE4.5 RB5.10 RBYTE5.3	BA BCD RI RU	VT_BOOL VT_ARRAY VT_I2 VT_R4 VT_R4
CHAR VT_BSTR	C CHAR	SC4 SCHAR4	RC5 RCHAR5	SC4.2 SCHAR4.2 RC5.4 RCHAR5.4	DT BA	VT_BSTR VT_BOOL VT_ARRAY
WORD VT_I4 (VT_UI2*)	W WORD	SW10 SWORD10	RW10 RWORD10	SW10.2 SWORD10.2 RW10.4 RWORD10.4	BA BCD D KF KT RI RU S5T SWAP TR	VT_BOOL VT_ARRAY VT_I2 VT_BSTR VT_I2 VT_BSTR VT_R4 VT_R4 VT_BSTR VT_I4 (VT_UI2*) VT_R4
INT VT_I2	I INT	SI4 SINT4	RI6 RINT6	SI4.2 SINT4.2 RI6.10 RINT6.10	BA BCD KT RI RU S5T SWAP TR	VT_BOOL VT_ARRAY VT_I4 VT_BSTR VT_R4 VT_R4 VT_BSTR VT_I4 VT_R4
DOUBLE WORD	D DW	SD6 SDW6	RD6 RDW6	SD6.2 SDW6.2	BA BCD	VT_BOOL VT_ARRAY

VT_R8 (VT_UI4*)	DWORD	SDWORD6	RDWORD6	SDWORD6.2 RD6.5 RDW6.5 RDWORD6.5	KF KG RI RU SWAP T	VT_I4 VT_I4 VT_R4 VT_R4 VT_R4 VT_R8 (VT_UI4*) VT_BSTR
DOUBLE INT VT_I4	DI DINT	SDI6 SDINT6	RDI2 RDINT2	SDI6.2 SDINT6.2 RDI2.5 RDINT2.5	BA BCD KG RI RU SWAP T	VT_BOOL VT_ARRAY VT_I4 VT_R4 VT_R4 VT_R4 VT_I4 VT_BSTR
REAL VT_R4	R REAL	SR2 SREAL2	RR2 RREAL2	SR2.2 SREAL2.2 RR2.5 RREAL2.5	----	----
STRING VT_BSTR	S STRING	SS5.2 SSTRING5.2	RS5.2 RSTRING5.2	----	KA DT	VT_BSTR VT_BSTR
S7-STRING VT_BSTR	G	SG5.2	RG5.2	----	KA DT	VT_BSTR VT_BSTR

* If "Standard Types only" is selected under Data Types (server settings)

** If "Unsigned Types also" is selected under Data Types (server settings)

<StartAddress>

The start address indicates the byte address where sending or receiving will start.

Example: SWORD10: Word 10 is the start address

If the start address is a specific bit, the bit number must be indicated, as well.

<.BitNumber>

The bit number must always be specified if the data type is BOOL.

Example: SX1.2: Bit 2 of byte 1 is the start address

[.ArraySize]

Arrays are formed to combine multiple units of the same data type to a single unit.

Example: RWORD10.4

More about [arrays](#)

[Suffix]

A suffix allows representing a value in a different format.

[More about suffixes](#)

11.8 Extended Raw Data Item Syntax

To transmit raw data without a protocol, the following extended item syntax is required:

```
#[+<startoffset>][.<NumberOfCharacters>][*<Encoding>]
#[+<startoffset>]?<EndCharacter>[*<Encoding>]
#[+<startoffset>]X<EndCharacter Hex>[*<Encoding>]
```

CSV

The frames are structured like CSV files. There are separators, which are often defined differently within a frame. Blank spaces before and after a separator are ignored.

```
#?<Space>-<Field>[+<startoffset>][.<NumberOfCharacters>][*<Encoding>]
#X<Space Hex>-<Field>[+<startoffset>][.<NumberOfCharacters>][*<Encoding>]
```

Optional start and end characters

Used for serial transmission; often included in transmission over the network. If necessary, they can be filtered out using the optional start offset.

Rules for the item syntax:

Characters	Description
#	First character
?,x	Separator (with field number) or end character (without field number)
-	Field number
+	Start offset. 0 means no start offset.
.	Length. 0 is invalid.
*	Encoding: uint8, int16, float32, string, ...
?-+.*	May be used in any order.
suffix SWAP	Swaps binary values from Intel to Motorola and vice versa

Encoding

Data format

Format	Signed	Syntax
8-bit	No	*uint8
8-bit	Yes	*int8
16-bit	No	*uint16
16-bit	Yes	*int16
32-bit	No	*uint32
32-bit	Yes	*int32
64-bit	No	*uint64
64-bit	Yes	*int64
32-bit floating point	Yes	*float *float32
64-bit floating point	Yes	*double *float64
Character string	-	*string *str

Encoding

Encoding	Syntax	Data	Available for
Binary	*uint16 *uint16bin	2 binary bytes	All
Hexadecimal	*uint16hex	1-4 hex digits	*uint8-64, *int8-64, *float32-64
Decimal	*uint16dec	1-5 decimal digits	*uint8-64, *int8-64, *float32-64
Octal	*uint16oct	1-6 octal digits	*uint8-64, *int8-64
BCD	*uint16bcd	BCD encoded	*uint8-64, *int8-64

NOTE:

The number of bytes/characters will, of course, vary depending on the data format. The values given above are examples of 16-bit data.

11.9 OPCpipe Item Syntax

In the Collect frame table, the OPCpipe addressing is handled as follows:

In the **Topic** field, enter the OPCpipe client access path.

In the **Item** field, enter the destination access path and the item name, separated by a question mark.

Topic field:

<OPCpipe Client AccessPath>

Item field:

<DestinationAccessPath>?<Item>

Key: <> mandatory [] optional

<OPCpipe Client Access Path>

Specifies the access path in the OPCpipe client, which is used for addressing the corresponding OPCpipe server.

<DestinationAccessPath>

Specifies the access path configured in the OPCpipe server as the connection to the PLC. For this access path, you need to enable the following option in the [connection](#) parameters: **OPCpipe Write allowed**.

<Item>

Specifies the real item name or the corresponding symbolic name.

Please note the following when registering items via OPCpipe:

- When registering an item via OPCpipe, the desired data type must be indicated. It can be omitted if
 - the item is known and can be browsed by the OPCpipe client, or
 - the access path of the OPCpipe client starts with specific letters that implicitly address the type of the access path, and if the data type can be determined from the item syntax:

Abbreviation	Item Syntax
S7	S7 Item Syntax
S5	S5 Item Syntax
MOD	Modbus Item Syntax
SLC PLC5	PLC-5 / SLC Item Syntax
MELS	MELSEC-Q Item Syntax
SR	Send/Receive Item Syntax

- For memory tables, the prefix **S7** can be used.
- It is not possible to access item "groups" (.Gx.) from the PLC using OPCpipe.

- The following data types are allowed when registering items via OPCpipe:

Simple data types	Array data types
VT_BOOL	VT_ARRAY VT_BOOL
VT_I1	VT_ARRAY VT_I1
VT_I2	VT_ARRAY VT_I2
VT_I4	VT_ARRAY VT_I4
VT_UI1	VT_ARRAY VT_UI1
VT_UI2	VT_ARRAY VT_UI2
VT_UI4	VT_ARRAY VT_UI4
VT_R4	VT_ARRAY VT_R4
VT_BSTR	

11.10 Special Syntax for Trigger/Normalisation

For triggers and normalization, custom rules can be defined. In this way, you can define expressions representing rules that cannot be selected using standard parameterization.

The following syntax is required:

```
<Command>(<Parameter>, <Parameter>, ...)
```

<Command> is either a special character or an identifier consisting of letters and numbers.

<Parameter> is either a number or another rule.

There are commands with none, one or more than one parameter. The table below provides an overview.

Command	Brief description	Number of parameters	Example	Explanation of command and example
& ^	Logical operators AND, OR, XOR	Two	&(i(0),6s(7))	Calculates the bitwise operation on the two arguments; the types are retained. The example masks the lower 3 bits of the 16-bit value i(0).
= ! < > { }	Relational operators EQUAL, UNEQUAL, GREATER THAN, LESS THAN, GREATER OR EQUAL, LESS OR EQUAL	Two	<(i(0),6s(20))	Compares the two arguments and returns the result as an 8-bit value (0 or 1). The example checks whether the 16-bit item i(0) is less than 20.
+ - * / %	Arithmetic operators ADDITION, SUBTRACTION, MULTIPLICATION, DIVISION, MODULO	Two	+(i(0),6s(5))	Calculates the arithmetic result of the arguments; the return value is of the same type. In the example, the number 5 is added to the 16-bit item i(0).
~	Unary operator NOT	One	~(i(0))	Calculates the bitwise negation of the argument. The type is retained.
n	Arithmetic negation	One	n(i(0))	Calculates the arithmetic negation of the argument.
a	Absolute value	One	a(i(0))	Removes the algebraic sign.
i o h	Item value, old item value (last cycle), first item value (last send)	One number	i(3)	Reads item values periodically from the PLC. The item is: <ul style="list-style-type: none"> • 0 for the item of the current frame table line • an item ID, corresponding to the line number in the frame table The example refers to the third line of the current frame table.
c u d f t	Abbreviations of tests on item values: Value changed, value increased (up), value	One number	c(0)	The same explanation applies as for i(x).

	decreased (down), value is zero (false), value is not zero (true)			
g b	Checks the item status GOOD (g) or BAD (b)	One number	g(0)	An item status is bad, for example, if there are problems with the PLC connection or if the item is not available In the example, a check is performed to see if the item is OK
8s 8u	8-bit constant values signed/unsigned	One number	8u(10)	Any desired values are represented. The number is in decimal notation Here, an unsigned 8-bit integer constant "10" is created.
6s 6u 2s 2u 4s 4u	16, 32 and 64-bit integer constants	One number	6s(5)	See 8s/8u
2f 4f	32 and 64-bit IEEE floating point values	One number	2f(12.68)	See 8s/8u. Fractional digits are separated by dot "."
x8s x8u ... x4f	Type conversion	One	x8u(/(i(0),6u(256)))	Type conversion between data types. In the example, the item i(0) is divided by 256 and converted to unsigned int 8
y0 y1 yp ye yc	Special constants 0,1,pi, Euler's constant, speed of light	None	*(i(0),yp)	Abbreviations of general constants. Other notations would be: y0 = 8u(0) y1 = 8u(1) yp = 4f(3.1416.....) ye = 4f(2.718....) yc = 2u(299792458) In the example, the circumference of a circle with a diameter of i(0) is calculated.
?	Conditional operator	Three	?(u(0),8u(12),8u(5))	The first parameter is evaluated as TRUE or FALSE. The second parameter will be returned if TRUE, the third parameter if FALSE. In the example, if the item value has increased, the value 12 will be output, otherwise 5.
Min, Max	Minimum or maximum of two values	Two	Max(i(0),8u(100))	The example returns the item i(0) if it is greater than 100, otherwise 100

The following commands apply only to trigger rules and not to normalization rules:

Command	Brief description	Parameter	Example	Explanation of command and example
SimpleOr, SimpleAnd	Logical combination of all	None		These rules are used to specify how the trigger items

	trigger elements in a frame			will be used
SendTime	Time since the last message	None		Time in milliseconds since the last time the frame was sent
ConnOk, ConnBad	Connection test	Connection name		This rule triggers once when the connection is established (ConnOk) or breaks (ConnBad)
ConnErr	Tests the connection for data errors	Connection name		This rule triggers when a data error occurs. The reasons can be: <ul style="list-style-type: none"> • For PLC connections: invalid DB, block too short, ... • For Pipe connections: quality values other than GOOD or LAST KNOWN VALUE • For database connections: all SQL errors

Complex examples

Two values are read from the PLC and the sum is delivered to the database:

First line: Topic/Item, Hidden

Second line: Topic/Item, Normalisation: +(i(0),i(1))

Addition of three values

First and second lines: Topic/Item, Hidden

Third line: Topic/Item, Normalisation: +(+(i(0),i(1)),i(2))

Check for value change, while ignoring status changes

&! (i(0),o(0)),c(0))

11.11 Virtual Collect Item Syntax

Access to ring buffer

R[Direction]<StartAddress>[.ArraySize]

Reads elements from the ring buffer. Elements that have not yet been written are invalid until there are enough values in the ring buffer. If no ring has been configured (in other words, if no ring functions have been selected), this syntax directly accesses the memory.

Access: read and write

Data type: as configured.

<Direction>

Indicates in which (temporal) direction the ring will be addressed. If no direction is specified, "Forward" is assumed.

	Syntax	Example		With array	
Name	FF cV	R0 RF0 RV0	They all read the oldest element	R0.10	The 10 oldest elements (the oldest first)
Reads the newest element	RR0.10	The 10 newest elements (the newest first)			

<StartAddress>

The start address in the ring; the unit is one ring element.

Example: R0 reads the oldest ring element. RR1 reads the second newest element.

[ArraySize]

The number of elements to be read. If no array size is specified, 1 is assumed.

Example: R10.2 reads 2 elements, starting at address 10

NOTE:

When writing to elements that do not yet exist in the ring:

- **With Rx:** the ring is increased (maximum up to the defined size), elements in-between are initialized with 0
Example: Previously 1 (R0), 2, 3, 4, 5 (R4) -> write 8 to R7 -> afterwards 1 (R0), 2, 3, 4, 5, 0, 0, 8 (R7)
- **With RRx:** Error message because the ring cannot be increased into the future
Example: Previously 1 (R0 = RR4), 2, 3, 4, 5 (R4 = RR0) -> writing to RR5 not possible

If no ring has been configured (in other words, if no ring functions have been selected), these restrictions do not apply; all values are initialized with 0 and are granted unrestricted read and write permissions.

Access to ring functions

All of these functions are simply entered as names, without parameters. Functions that are not selected have undefined values. If no function is selected, access will always generate an error message.

push	Adds a new ring element. If the ring is full already, all the existing elements move 1 forward. The oldest element is removed. Access: write-only Data type: as configured.
sum	Calculates the sum of all elements in the ring. Access: read-only Data type: as configured.
avg	Calculates the average of all elements in the ring. Access: read-only Data type: as configured.
min	Calculates the minimum of all elements in the ring. Access: read-only Data type: as configured.
max	Calculates the maximum of all elements in the ring. Access: read-only Data type: as configured.
count	Returns the number of elements currently stored in the ring. Access: read-only Data type: WORD

Direct memory access

M<DataType><StartAddress>[.ArraySize]

or with bit access

M<DataType><StartAddress><.BitNumber>

Reads the memory directly. Elements that have never been used have the start value 0. This syntax only makes sense for connections without ring management because when using the ring R0 does not always coincide with the first value (MW0).

Access: read and write

Data type: as specified.

<DataType>

Name	Syntax	Example	With array
Bit	X	MX2.3 M2.3	--
Byte (8-bit unsigned)	B Byte	MB2 MByte2	MB2.3
Char (8-bit signed)	C Char	MC2 MChar2	MC2.3
Word (16-bit unsigned)	W Word	MW2 MWord2	MW2.3
Int (16-bit signed)	I Int	MI2 Mint	MI2.3
Double Word (32-bit unsigned)	D DW DWord	MD2 MDW2 MDWord2	MD2.3
Double Int (32-bit signed)	DI DInt L Long	MDI2 MDInt2 ML2 MLong2	MDI2.3
Quad Word (64-bit unsigned)	Q QW QWord	MQ2 MQW2 MQWord2	MQW2.3
Quad Int (64-bit signed)	QI QInt	MQI2 MQInt2	MQI2.3
Real (32-bit floating point)	R Real	MR2 MReal2	MR2.3
Double Real (64-bit floating point)	DR DReal Double	MDR2 MDReal2 MDouble2	MDR2.3

<StartAddress>

The start address in the memory where reading will start. Always given as a byte address.

[ArraySize]

The number of elements to be read. If no array size is specified, 1 is assumed. Not allowed for bit elements.

<BitNumber>

The bit number to be read (0 to 7)

11.12 Arrays

An array is a series of identical elements. An array combines multiple units of the same data type to form a single unit. To create an array, append the length of the array to the standard syntax, separated by a dot.

Arrays are NOT possible for:
BOOL DATE AND TIME DATE TIME TIME OF DAY STRINGS (already represent an array of characters) as well as for all suffixes that are String or Bool towards the client side: BA (already represents an array of bits) KA S5T KT D T DT DU TOD

11.13 Suffixes

A suffix allows representing a value in a different format.

Suffixes	Syntax	Applicable to	Area	Data type	Variant data type	Comment
BitArray	BA	Byte Word Int DWord DInt	Number of bits: Size in bytes multiplied by 8 Size in words multiplied by 16 Size in Ints multiplied by 16 Size in DWords multiplied by 32 Size in DInt multiplied by 32	BOOLE AN BOOLE AN BOOLE AN BOOLE AN BOOLE AN	VT_BOOL ARRAY VT_BOOL ARRAY VT_BOOL ARRAY VT_BOOL ARRAY VT_BOOL ARRAY	With the BA suffix, the data stored in the PLC is represented as an array of bits.
BCD	BCD	Byte Word Int DWord DInt	Byte: 0 to 99 Word: 0 to 99 Int: 0 to 999 DWord: 0 to 9999999 DInt: 0 to 9999999	SHORT SHORT	VT_I2 VT_I2 VT_I4 VT_I4 VT_I4	With the BCD suffix, the data stored in the PLC is represented as an unsigned, binary-encoded value. For example, the decimal value "65535" will be represented as "9999".
Date	D	Word	1990-01-01 to 2168-12-31	STRING	VT_BSTR	<p>The D suffix is used to represent the data stored in the PLC as the DATE data type. The DATE data type occupies one word. The contents correspond to the number of days since January 1, 1990. The representation contains the year, the month and the day, each separated by a hyphen. September 1, 2006 is represented as 2006-09-01. The value range is from 0 (0 days since Jan. 1, 1990: 1990-01-01) to 65378 (65378 days since Jan. 1, 1990: 2168-12-31).</p> <p>Rules for using the D suffix:</p> <ul style="list-style-type: none"> The years from 1990 through to 2089 can be given as 2-digit or 4-digit numbers. From 2090 onwards, the years must be given as 4-digit values. 90 to 99 ==> 1990 to 1999 00 to 89 ==> 2000 to 2089 The months and days can be given as 1 or 2-digit numbers. All characters except numbers are allowed for the separators (e.g. 89/01/17). You can use any number of separators. <p>This suffix is not supported for OPCpipe.</p>
Date	DT	Word, Int	1990-1-1-	STRING	VT_BSTR	With the DT suffix, the data

and Time			00:00:00.000 to 2089-12-31-23:59:59.999***			stored in the PLC is represented as a composite DATE_AND_TIME data type. The DATE_AND_TIME data type occupies 8 bytes (64 bits). The representation contains the year, the month, the day, the hour, the minute, the second and the millisecond. Special attention must be given to the separators (semicolon, colon and dot)!
Date and Time (SQL)	DU	Word, Int	1990-01-01 00:00:00.000 to 2089-12-31 23:59:59.999***	STRING	VT_BSTR	With the DU suffix, the data stored in the PLC is represented as a composite DATE_AND_TIME data type in SQL compatible format. The time of day is not preceded by a hyphen. This suffix is not supported for OPCpipe .
ASCII to Hex	KA	String, S7 string	HEX: 0 to 9, A to F	STRING	VT_BSTR	With the KA suffix, the data stored in the PLC is represented as HEX characters.
Signed	KF	Byte (LByte, RByte, S5), Word (S5), DWord (S5)	Byte: -128 to 127 Word: -32768 to 32767 DWord: -2147483648 to 2147483647	CHAR SHORT LONG	VT_I2 VT_I2 VT_I4	With the KF suffix, the data stored in the PLC is represented as a signed fixed point number.
S5-KG	KG	DWord, DInt	0.1469368E-38 to 0.1701412E39 Attention: The KG area in the PLC is larger than in the PC!	REAL	VT_R4	With the KG suffix, the data stored in the PLC is treated as a 4-byte floating point number. This suffix is not supported for OPCpipe .
S5-KT-Format	KT	Word, Int	000.0 to 999.3	STRING	VT_BSTR	With the KT suffix, the data stored in the PLC is represented as a 2-byte time constant. The time base is included in addition to the time value. The value range is from 000.0 to 999.3
Signed to Real	RI	Byte, Word, Int, DWord, DInt	Converts an item to a real value and rounds the value while writing	REAL	VT_R4	With the RI suffix, the data stored in the PLC is represented as a signed REAL data type. The value is rounded while writing to the PLC. Example: db9dw0RI – the value 10.7 is rounded to 11.
Unsigned to Real	RU	Byte, Word, Int, DWord, DInt	Converts an item to a real value and rounds the value while writing	REAL	VT_R4	With the RU suffix, the data stored in the PLC is represented as an unsigned REAL data type. The value is rounded while writing to the PLC.
S5 Time	S5T	Word, Int	0ms to 2h46m30	STRING	VT_BSTR	With the S5T suffix, the data stored in the PLC is represented as S5TIME (SIMATIC time). The S5TIME data type occupies one

						16-bit word and is the product of the time value and the time interval (time base). The time duration is given in hours, minutes, seconds and milliseconds. The BCD number format is used for internal representation. The value range is from 0 ms to 2h46m30s. The smallest value is 10ms. This suffix is not supported for OPCpipe .
Swap Bytes	SWAP	Word Int DWord DInt	Swaps high byte and low byte		VT_I2 VT_I4*/UI2** VT_I4, VT_I4, VT_I4	With the SWAP suffix, the high-order and low-order bytes of the data stored in the PLC are swapped.
Time	T	DWord	- 24D_20H_31M_23 S_648MS to 24D_20H_31M_23 S_647MS**	STRING	VT_BSTR	With the T suffix, the data stored in the PLC is represented as the TIME data type. The TIME data type occupies one double word (32 bits). The representation contains the days (d), hours (h), minutes (m), seconds (s) and milliseconds (ms). Milliseconds can be omitted. The value range is from - 2147483648 (- 24d20h31m23s648ms) to 2147483647 (24d20h31m23s647ms)
Time of Day	TOD	DWord, DInt	0:0:0.0 to 23:59:59.999		VT_BSTR	With the TOD suffix, the data stored in the PLC is represented as the TIME_OF_DAY data type. The TIME_OF_DAY data type occupies one double word (32 bits). The representation contains the hours:minutes:seconds and .milliseconds. Milliseconds can be omitted. The value range is from 0:0:0.0 to 23:59:59.999 This suffix is not supported for OPCpipe .
TimeReal	TR	Word, Int	0.01 to 9990.0	REAL	VT_R4	With the TR suffix, the data stored in the PLC is represented as the TIME REAL data type. The value range is from 0.01 to 9990.0 (s)

* If "Standard Types only" is selected under Data Types (server settings)

** If "Unsigned Types also" is selected under Data Types (server settings)

*** ms can be omitted.

Index

- 3 -

3964R 61

- A -

A timeout occurred. 38
 About 91
 Access Path 43, 46, 74, 75, 77
 Additional features 24
 Allen Bradley 27, 56
 Allen-Bradley 121
 ARP 25
 Arrays 141
 AS511 29

- B -

Basic Configuration 33
 Basics 24

- C -

Change the Password 82
 Clock Settings 90
 CLX 27, 46, 56
 Collect Connection 99
 Collect Connection Configuration 54
 Collect Create Frame Entry 95
 Collect Edit Array Settings 98
 Collect Edit Rule 134
 Collect Edit Type Entry 98
 Collect Frame Entry 96
 Collect Frame Table Edit 42
 Collect Indexed Text 104
 Collect Indexed Text – Add 104
 Collect Indexed Text – Table Edit 104
 Collect Trigger Settings 102
 CollectEditTriggerEntry 102
 CollectPLCConfirm.htm 102
 Configuration 35, 43
 Connection 43, 46, 74, 75
 Connection menu 77
 Connection: AS511 62
 Connection: MPI 62
 Connection: Telnet 60
 Connetion: Modbus RTU 62

ControlLogiX 27, 56
 Copy Connection 75
 CR Parameters 53

- D -

Database Connect Parameters 65
 Default connections, monitoring 89
 Delete 75
 Delete Connection 75
 Destination MAC 53
 Device SSH Settings 85
 Device System Settings 89
 Diagnose of Default Connections 89
 Diagnostics for echolink connection 79
 Diagnostics menu 78
 Dialog boxes 35, 43
 Disable connection 75

- E -

echochange connection to SLC / PLC-5 56
 echochange Connection: Clx Protocol 56
 echocollect Connection: Modbus Protocol 56
 echolink connection diagnostics.htm 79
 Edit Access to a Station in Your List 37
 Edit Connection 74
 Edit H1 parameters 53
 Entry Type 96

- F -

F\$\$\$Connection not found 38
 File menu 77

- H -

H1 Connect Parameters 53
 H1 System Settings 86
 Help menu 91
 Hilfeüberblick 9
 How to Proceed 33

- I -

INAT OPC-Server – General 20
 IP address 49, 82
 IP address syntax 49
 IP System Settings 87
 Item syntax 109, 110, 113, 117, 121, 125, 128

- L -

License 91, 92
License conditions 91
License Overview 91
Licenses 92
Licensing 34
Line 53
Lizenzbedingungen 21
Logger 20, 24, 41, 79, 80
Logger configuration 79
Logger Settings 80

- M -

MAC address 82
Mail connection 73
Master/Slave 24
MELSEC-Q 28, 46, 57, 125
MELSEC-Q item syntax 125
Melsec-Q Protocol 57
Menu 76, 77, 78, 82, 86, 91
Mitsubishi 28, 57, 125
Modbus 27, 46, 56, 117
Modbus item syntax 117
Modbus RTU 62
MPI 62
MPI ring monitoring 79
MPI ring status 79
MPI Station Extended Settings 84
MPI Station Settings 84
MPI Station Values 84
Multicast 53

- N -

NetCon configuration 35
NetCon OPC 24
netLINK 46
Network Parameters 74
Network PLC Protocol 46
Network Protocol 46
New Access Path 43
New Connection 43, 46
New station 37

- O -

Offline parameterization 38
On/Off:connection 75

OPCpipe 29, 43, 46, 132
OPCpipe client 43, 132
OPCpipe item syntax 132
OPCpipe server 43, 132
Operation and Configuration 33
Overview 20

- P -

Password 24, 82
PLC connections 26
PLC Header 52
PLC-5 27, 46, 121
PLC-5 / SLC item syntax 121
Port 50
Port number 50
Priority 53
Protocol selection 46
Protocol settings 56, 57

- R -

Reboot 90
Reboot password 90
Restart 90
Rockwell 27, 56, 121

- S -

S5 26, 46
S7 26, 46, 51, 110, 113
S7 item syntax 110, 113
Select a connection 39
Select a Station in Your Network 36
Select station 36
Selecting a Collect connection 99
Send / Receive 28, 46
Send/Receive 128
Send/Receive item syntax 128
Serial Diagnostics 81
Serial Parameterization 64
Server Connections 55
Settings menu 86
Simulation 74
SLC 27, 46, 121
Specific settings for TCP/IP 48
Starting the program 24
Station menu 82
Station not found 38
Station parameters 82
Station Password 82
Station selection in the network 36

Suffixes 142
Switch off 75
Switch off (connection) 75
System requirements 21

- T -

TCP/IP Stati 79
TCP/IP System Settings 87
Telnet 60
Trigger 96
TSAP 50, 51
Type 53

- V -

Versions 91

- W -

Warm start 90
Windows 35, 43

This page is intentionally left blank.

Softing Industrial Automation GmbH

Richard-Reitzner-Allee 6
85540 Haar / Germany
<http://industrial.softing.com>

 Tel: + 49 89 45 656-0
 Fax: + 49 89 45 656-488
 info.automation@softing.com