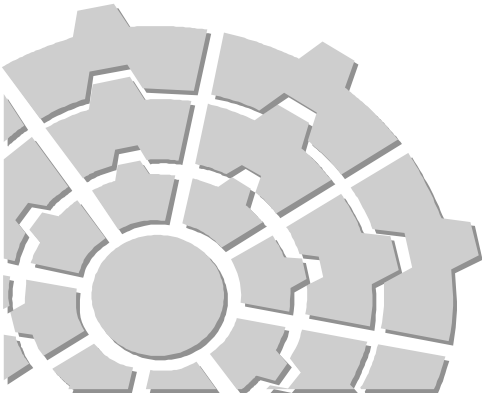
DeltaLink SINEC_H1
Sinec H1 Bus interface software for FactoryLink

User Manual



Version 5

Printed: 6 aug, 1997

Table of Contents

This page is left blank intentionally.

1. Introduction

Thank you for buying this driver! We hope you will enjoy using this product.

1.1. Scope of this document

This manual is written for a technician who is familiar with both the FactoryLink® IV software and the Siemens S5 or Siemens TI Programmable Logic Controllers (PLC's). This document can be used both as a training manual as well as a reference manual.

> ***Note:*** *Please check the contents of the shipment with the list as described in the next chapter.*

The first section of this manual deals with the installation of hardware and software in your FactoryLink workstation. This part is split into a platform independent and a platform specific part. Please read carefully through both parts to make sure both hardware and software are installed correctly.

The second part explains the operation principles of the communication with the Siemens PLC and the Decoder. Here all terms and definitions are explained to the reader. It explains terms like "Dataset" and "Virtual Connections'. This part should be read by both the PLC programmer and the FactoryLink programmer to make sure that the optimum performance can be achieved.

The third part explains the exact tables associated with this driver. This part is useful only to FactoryLink programmers and can be used as a reference. This part is also an example of how to use this driver with the Decoder. It shows the entries made for the pre-configured demo which comes with this package. The demo program can be used to check if the communication is working without making a complete application.

The last part are the appendices which contain summarised data.

DeltaLink bv, Pioenroosstraat 26, 5241AB Rosmalen, The Netherlands,

tel + 31 73 523 1234
fax +31 73 523 1222

This page is left blank intentionally.

2. Contents of shipment

Please check the package you received with the checklist below. Should there be an item missing contact DeltaLink bv to correct the problem. There is a limit of 90 days after shipment to report problems!

This package includes the following:

❶ A Sinec H1 driver build contained on the FactoryLink Device Inteface CDROM media or on a diskette(s) labelled "DeltaLink Sinec H1 Protocol Driver"

❷ This manual.

You should also have:

- ☑ A IMX based decoder e.g. DeltaLink decoder.
- ☑ The correct hardware and software for the OSI Ethernet stack on your platform. (See also chapter 3)
- ☑ Siemens or TI PLC with H1 communication board and software to program both the CPU and the CP boards.
- ☑ An Ethernet-network with at least both the PLC and the FactoryLink station connected to it.

This page is left blank intentionally.

3. Installation

3.1. Installation of the FactoryLink software

To install the FactoryLink task and its related tables please follow the following steps.

Before installing

Before installing the driver on the system, FactoryLink must have been installed error free. It is very important that all the environment settings are made for the FactoryLink system such as the *FLINK*, *FLOPT* etc.

First:

Copy the files from the "DeltaLink SINEC_H1" media to the appropriate directory. This will be done automatically by running the install utility placed on the installation media. The installation procedure differs for UNIX and W95-OS/2-NT systems.

For W95-OS/2-NT systems follow the next procedure:

```
a:\↵
install↵
```

For UNIX systems first an install directory must be created. The files on the install media must be first copied to the install directory using the *tar* command. From this directory the install utility can be run.

For UNIX systems follow the next procedure:

```
mkdir install
cd install
tar xv↵
install↵
```

Second:

After you installed the software you need to activate the tables in the FactoryLink Configuration Manager (FLCM). The installation automatically appends the *sinec_h1.ac* entry into the *{FLINK}/ac/titles* file[1]. The place of this entry is also the place where the option appears in the FLCM Main Menu. Therefore check the validity of the entry and move it to the place where you want to appear it in the Configuration Manager. The entry must match the sinec_h1 entry in the following table:

```
file: {FLINK}/ac/titles:

...
windhdr.ac
decoder.ac
sinec_h1.ac
spool.ac
...
```

Third:

To make sure all the Configuration Tables (CT's) are generated after a change, the install utility automatically adds the sinec_h1 entry at the end of the *{FLINK}/ctgen/ctlist* file. The place of this entry is not important. Check if this entry has the same format as in the next table.

[1] {FLINK} is the working directory for the FactoryLink programs.

```
file: {FLINK}/ctgen/ctlist:

...
recipe: rcphdr rcpovr
decoder: decoderm decoderp decoderd decodert
sinec_h1: sinech1m sinech1x sinech1p sinech1d
iml: imltags imltrig
...
```

Fourth:

To enable the help functionality for the Siemes Sinec H1 Protocol Driver tables in the Configuration Manager, the installation utility reindexes the 'help-index' for the Configuration Manager. If desired reindexing of the 'help-index' can be started from the command line prompt.

```
mkhelp↵
```

Fifth:

The FactoryLink Configuration Manager uses a map file, {FLINK}/ac/ac2ct.map, to be aware of the different configuration tables which can be located behind one entry in the main menu. Upon startup, the Configuration Manager reads the map file instead of all the table configuration files, mainly because reading all these files at once takes too much time. An account manger is present to update this conversion file, and can be started from the command line.

```
acctmgr  -c -d -t{FLINK}/ac/titles↵
```

Sixth:

The protocol driver must, with the FactoryLink Configuration Manager (FLCM), be entered in the System Configuration table. An entry of an existing task which will not be used at run-time can be overwritten or a new entry can be created with (as a minimum) the following data:

Task Name	*Description ....*	*Executable File*
SINEC_H1	***Siemens Sinec H1 Protocol Driver***	***bin/sinec_h1***

The Task Name and name of the executable file are fixed and should not be altered by the user.

This completes the installation of the FactoryLink (software) parts.

3.2. Installation of the Ethernet Stack

It is necessary to have an Ethernet stack loaded which can be used by the driver to communicate with the Sinec H1 network. This stack is based on an OSI transport layer 4 protocol. Depending on the platform the stack can be different. Please choose the correct platform and options and make sure your stack is loaded correctly.

A platform dependent description of how to set-up an Ethernet stack for the Sinec H1 network can be found in *Appendix A*.

3.3. Installation of the protection

The Siemens Sinec H1 protocol driver is protected via the DeltaLink option file. This file contains authorization sequence codes for DeltaLink modules. The protection is linked to the serial number of the FactoryLink package.

3.3.1. The DeltaLink option file

The installation media of the Sinec H1 driver contains an option file, named ‘delta.opt’, in the 'opt' directory. This file contains the unique authorization sequence which enables the Sinec H1 driver to run. The install utility automatically copies the authorization sequence into the {FLOPT}/delta.opt file. It is also possible to enter the authorization sequence manually into the {FLOPT}/delta.opt file. For more information on the delta.opt file refer to *Appendix B*.

Note that the task will only run on the FactoryLink system with the same serial number. The *delta.opt* file on the installation diskette contains, for reference, the serial number of FactoryLink.

3.3.2. Demo installation

It is possible to install the Sinec H1 driver without an authorization code. This will be done from a normal installation media. In this case the task will start up but only runs in so called 'demo' mode. This means that the driver runs only for a limited period of time (five hours). After this period has expired the task will shutdown and can not be restarted before the complete FactoryLink system has been restarted.

After installation of this demo version an authorization code can be ordered and installed which enables the task to run without the time and restart limitation. The authorization code must be entered manually in the {FLOPT}/delta.opt file. For more information on entering the authorization code in the delta.opt file refer to *Appendix B*.

The limitations of a demo version of the Siemens Sinec H1 driver are:

- five hours of consecutive run-time
- not restartable (FactoryLink must be restarted)

4. Principle

4.1. The decoder RAP Ddriver principle

RAPD stands for Rapid Application Protocol Driver. The RAPD principle was adopted so that protocol drivers can be easily and rapidly configured for a FactoryLink application. RAPD is based on the Intertask Mail Exchange Standard or IMX, which defines a way for a protocol driver task to communicate with an I/O Translator task (e.g. Decoder, high speed logger). The RAPD system consists of a protocol driver which communicates with external devices (RTUS, PLCs, etc.) and a translator which controls data storage (going to and coming from a protocol driver) in the FactoryLink real-time database. All data collected by the protocol driver is referenced as contiguous blocks or ranges within the device. This enables communications between the driver and a device to be very efficient. All data is referenced between the driver and the translator in terms of datasets. Datasets, described in the next section, define memory regions or locations of data within a device.
The protocol driver and the IOX communicate with one another via FactoryLink mailbox tags, according to the IMX standard. Every task (translator and protocol drivers) has its own mailbox, so for full communication between a translator and a protocol driver a mailbox database element for every task has to be defined. The IMX standard is especially designed for the following situation. To use one decoder and several protocol drivers. For example the decoder together with the Sinec H1 protocol driver and the Modbus Plus protocol driver. Aside from storage duties, the translator provides data conversions (i.e. analog, IEEE conversions, etc.) for I/O data to/from a protocol driver.

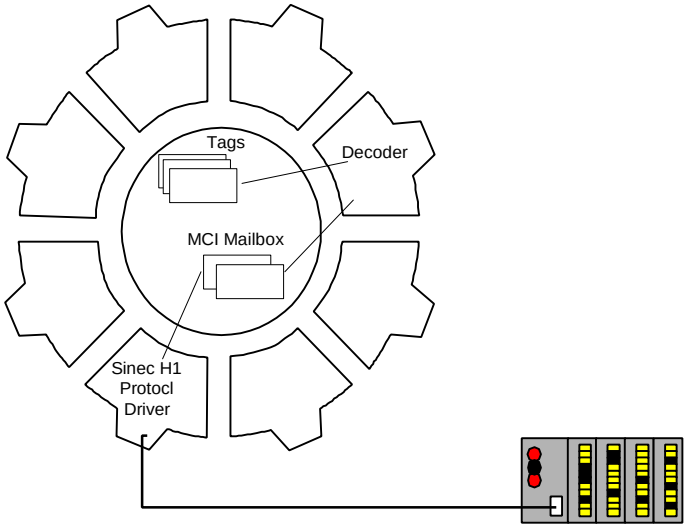


Figure 4.1.1 The RAPD principle.

4.2. H1 Communication

4.2.1. Datasets

A dataset is a (contiguous) area of data in the PLC. The complete area can be read or written to, with one command (trigger). Except actions on the complete area, the protocol driver is also capable of addressing specific elements (or a group of elements) in the data area. The decoder takes care of mapping the PLC-data to/from FactoryLink Tags (including conversions).
A dataset consist out basic elements, which are all of the same size, for example all words or all bytes. This size corresponds with the size definition of the data area in the PLC and is called the boundary of a dataset. The boundaries for the different data areas used with the Sinec H1 driver are listed in the tables below.

Siemens S5	
PLC data type	Boundary
Data block	Word
Flags	Byte
Input	Byte
Output	Byte
Counter	Word
Timer	Word
System word	Word
Peripheral byte	Byte
Extended Peripheral byte	Byte
Expanded data block	Word

Siemens S7	
PLC data type	Boundary
Data block	Byte
Flags	Byte
Input	Byte
Output	Byte

Siemens TI	
PLC data type	Boundary
Data block	Byte
Flags	Byte
Input	Byte
Output	Byte
Variable	Word
Constant	Word
Discrete Input	Byte
Discrete Output	Byte
Control Register	Byte
Discrete Input Packed	Byte
Discrete Output Packed	Byte
Control Register Packed	Byte
Word Input	Word
Word Output	Word
Timer or Counter Preset	Word
Timer or Counter Current	Word
Drum Step Preset	Word
Drum Step Current	Word
Drum Counter Preset	Word
System Status words	Word
Drum Current count	Long
Variable Memory	Word
Constant Memory	Word
Loop Status	Word
Loop Mode	Word
Loop Gain	Long

Siemens TI	
PLC data type	Boundary
Loop Reset time	Long
Loop Rate time	Long
Loop High Alarm Limit	Long
Loop Low Alarm Limit	Long
Loop Process Variable	Long
Loop High Process Var.	Long
Loop Low Process Var.	Long
Loop Orange Deviation	Long
Loop Yellow Deviation	Long
Loop Sample Rate	Long
Loop Set Point	Long
Loop Output	Long
Loop V-flags	Long
Loop Control Flags	Word
Loop Ramp/Soak Status	Word
Loop Error	Long
Loop Bias	Long
Loop Process Variable High High Alarm	Long
Loop Process Variable Low Low Alarm	Long
Loop Rate Change Alarm	Long
Loop Setpoint High Limit	Long
Loop Setpoint Low Limit	Long
Loop Alarm Deadband	Word
Loop Raw High Alarm Limit	Word
Loop Raw Low Alarm Limit	Word
Loop Raw Process Variable	Word
Loop Raw Orange Dev Alarm Limit	Word
Loop Raw Yellow Dev Alarm Limit	Word
Loop Raw output	Word
Loop Raw setpoint	Word
Loop Raw Error	Word
Loop Raw High/High Alarm Limit	Word
Loop Raw Low/Low Alarm Limit	Word
Loop Raw Alarm Deadband	Word
Loop Raw Bias	Word
Loop Raw Setpoint Low Limit	Word
Loop Raw Setpoint High Limit	Word
Loop Most-sig word loop C-flags	Word
Loop Least-sig word loop C-flags	Word
Loop Derative Gain Limiting Coef	Long
Loop Ramp Soak Step Number	Word
Loop Alarm Acknowledge Flags	Word
High Alarm Limit	Long
Alarm Limit	Long
Process Variable	Long
Process Variable High Limit	Long
Process Variable Low Limit	Long
Orange Deviation Alarm Limit	Long
Yellow Deviation Alarm Limit	Long
Sample Rate in seconds	Long
Alarm Setpoint	Long
Alarm Variable Flags	Word
Alarm Control Flag	Word
Alarm Error	Word
High High Alarm Limit	Word
Low Low Alarm Limit	Word
Rate of Change Alarm Limit	Word
Setpoint High Limit	Word
Setpoint Low Limit	Word

Siemens TI	
PLC data type	Boundary
Alarm Deadband	Word
Raw High Alarm Limit	Word
Raw Low Alarm Limit	Word
Raw Process Variable	Word
Raw Orange Deviation Alarm Limit	Word
Raw Yellow Deviation Alarm Limit	Word
Alarm Raw Setpoint	Word
Raw Alarm Deadband	Word
Alarm Raw Error	Word
Raw High-High Alarm Limit	Word
Raw Low-Low Alarm Limit	Word
Raw Setpoint Low Limit	Word
Raw Setpoint High Limit	Word
Most-sig word Alarm C-flags	Word
Least-sig word Alarm C-flags	Word
Alarm Acknowledge Flags	Word
VME (TI575 only) A24 space	Word
VME (TI575 only) 16 space	Word
App Z global variables	Word
App Y global variables	Word
App X global variables	Word
App W global variables	Word
App V global variables	Word
App U global variables	Word
App T global variables	Word
App S global variables	Word
App R global variables	Word
App Q global variables	Word
App P global variables	Word
App O global variables	Word
App N global variables	Word
App M global variables	Word
App L global variables	Word
App K global variables	Word
App J global variables	Word
App I global variables	Word
App H global variables	Word
App G global variables	Word
App F global variables	Word
App E global variables	Word
App D global variables	Word
App C global variables	Word
App B global variables	Word
App A global variables	Word
Local App global variables	Word

4.2.2. Virtual connection

Data exchange between a FactoryLink workstation and a PLC is performed by using the *read*- and *write*-services of e.g. the Siemens CP535/CP143 communication processor. The protocol driver task uses the *read*-service to read (or fetch) data from the PLC, and it uses the *write*-service to write data to the PLC. Both actions can be initiated by the driver task, also the PLC may use the *write*-service for sending data to the FactoryLink station, the unsolicited receive for the protocol driver.

Every *read/write/receive*-service is associated with a so called Virtual Circuit (VC). A Virtual Circuit is a bi-directional point-to-point communication link through the network. Each Virtual Circuit requires a Transport Service Access Point Identifier (TSAP-ID). To set-up a Virtual Connection between two stations, the following information is needed for each station.

- Local Ethernet address.
- Local TSAP.
- Remote Ethernet address.
- Remote TSAP.

The Sinec H1 protocol supports different types of read and write services, the ones supported are described in the next paragraphs. In the table below are all the possible services listed together with the information if they are supported or not.

Read/Write Service	Supported
Block read	Yes
Block write	Yes
Exception write	Yes
Encoded write	Yes
Unsolicited Receive	Yes

4.2.3. Block read and block write

The block read and block write commands are performed on a contiguous block of data in the PLC, the dataset. Both commands are initiated by the FactoryLink station. The decoder sends requests to the Sinec H1 protocol driver for reading or writing an entire dataset. For configuring the decoder refer to the manual of the decoder.

4.2.4. Exception write

An exception write is initiated by the FactoryLink workstation, and writes data to the PLC. Not a complete dataset but one element from the dataset is written to the PLC. The protocol driver will receive a request for an exception write from the decoder. The first action the driver will take is to check if the write can be accomplished with one write or must be read first.

In case the exception data element size is smaller than the size of the PLC data type element (boundary) the data must be read first, patched with exception data and written back. If this is not done, data will be unintentionally overwritten in the PLC. Whether this is the case depends on the data type of the dataset.

For example a Siemens S5 PLC: if a data left (DL) in a Data Block (DB) has to be set then the specific data word has to be read first because the smallest element in a Data Block is a word. The DL has to be patched into the word and written back.

In the above mentioned example it is possible, that the device changes the contents of the word, after the protocol driver has read the word, but before the new value is written. This means that after the write of the protocol driver the content change of the PLC is lost. For applications which require that single elements can be accessed both by the PLC and the protocol driver, an encoded write can be used. The encoded write will be discussed in the next section.

4.2.5. Encoded write

The encoded write function is almost similar to the exception write in the fact that both functions write single data elements. The difference is that the exception write accesses directly the destination address in the PLC, whereas the encoded write composes an encoded write command and writes this command to a location in a certain data block. The PLC program reads and decodes this encoded write command and accesses the actual destination location.

The advantage of this method is that for every element to write only one write command is needed because the PLC program does the actual operation on the data element (no reading before writing). Another advantage is that the PLC program can control all encoded writes because they all are written

to one location in the PLC. For a complete description and the lay-out of an encoded write command refer to *Appendix E* and *Appendix F* .

The location where the encoded write command will be placed is part of the definition of a logical device, a detailed description is in the next chapter.

4.2.6. Unsolicited receive

In case of an unsolicited receive the PLC sends a dataset to the FactoryLink workstation, without a specific request from that workstation. This means that the action is initiated by the PLC. The protocol driver will check if the received data is configured in one of the datasets. If this is the case, the data will be sent with the dataset information to the decoder which converts and updates the data. If no dataset is specified regarding to the received data, it will be discarded.

The unsolicited receive function is a very powerful functionality for fast communication because no request has to be placed by the protocol driver. This way less overhead is involved in communication. Data which alters unpredictably such as alarms are very suitable for the unsolicited receive function. Some extra programming effort has to be done in the PLC because the PLC is in charge of sending the data.

4.3. FactoryLink domain selection

The standard domain for the Siemens Sinec H1 protocol driver is the SHARED domain. The protocol driver communicates with a dedicated piece of hardware, there fore only one task should be able to access the hardware. If only one program accesses the hardware, the task should be located in the shared domain and therefore started by the shared runtime-manager.

Important: The protocol driver and the decoder must be in the same domain (either SHARED or USER).

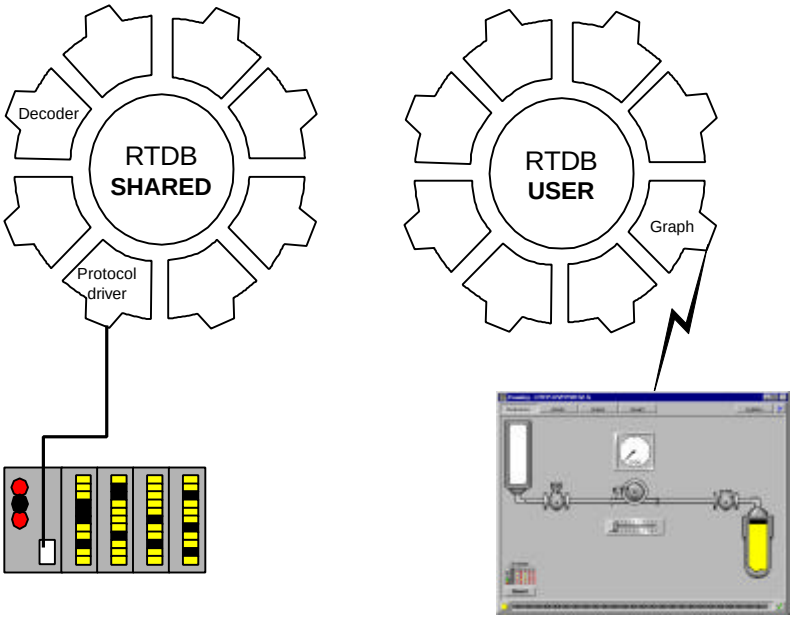


Figure 4.3.1 Standard domain selection.

This page is left blank intentionally.

5. Configuration tables

In the Configuration Manager Main Menu, select **DeltaLink Sinec H1 Protocol Driver**. Four tables appear, with the titles of all panels visible for direct access. To access a specific panel position the cursor on a visible area and press the left mouse-button, or use the Next/Prev buttons .

Note: For general information about entering data in FactoryLink configuration tables, refer to the FactoryLink Fundamentals Manual.

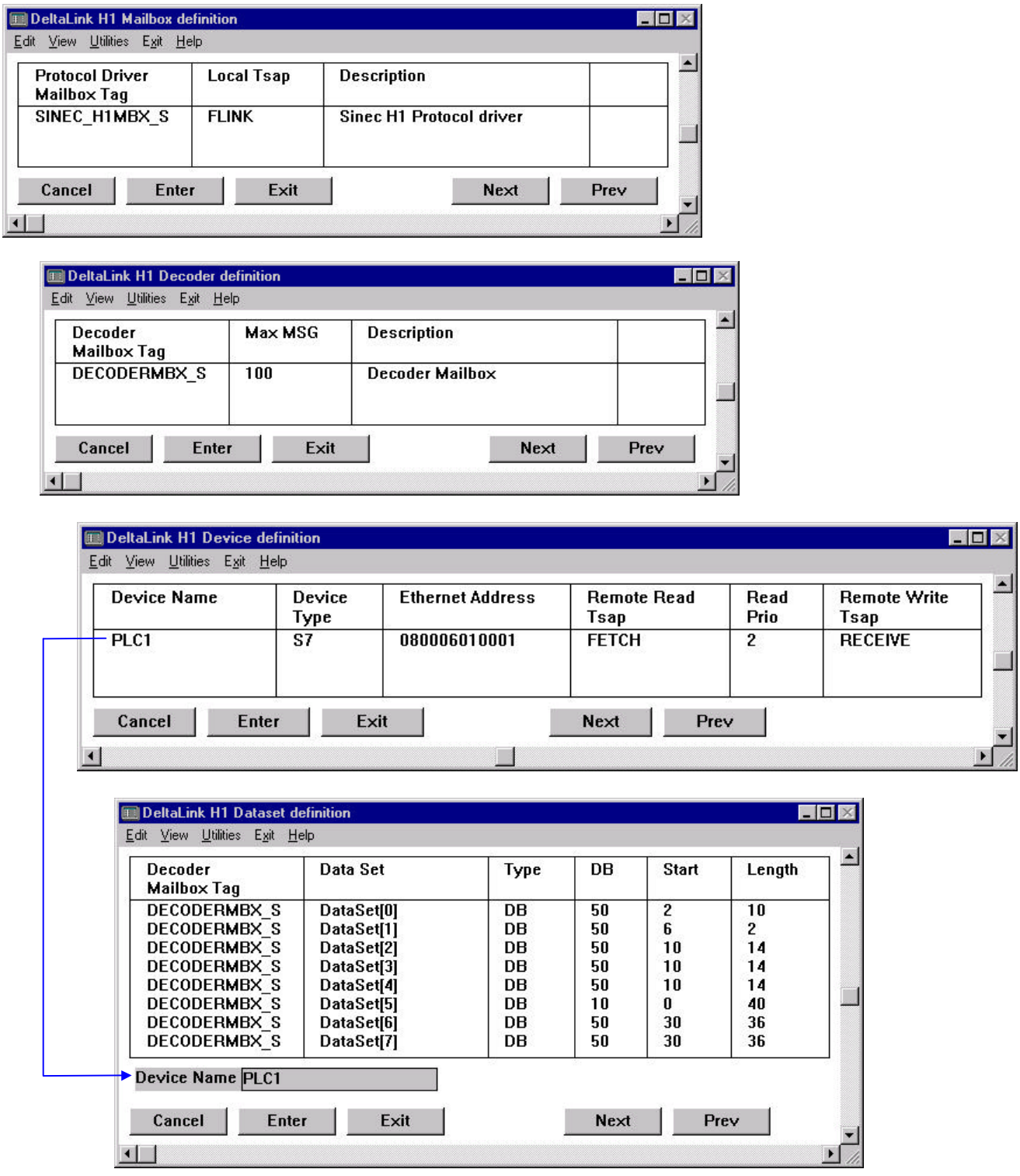


Figure 5.0.1 Siemens Sinec H1 Configuration panels.

5.1. Mailbox definition

From the display of all the panels, select the *DeltaLink H1 Mailbox Definition* panel.

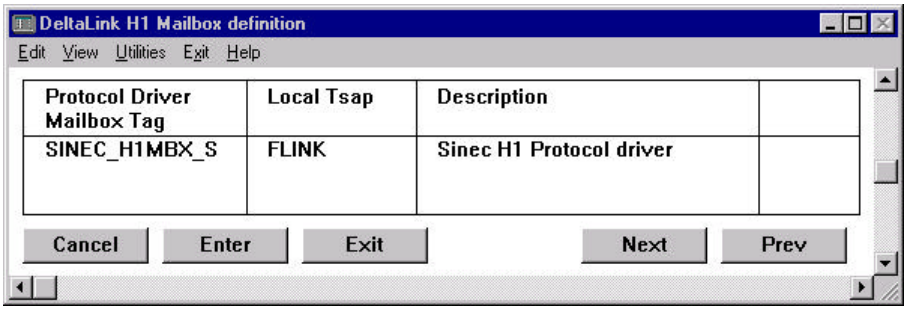


Figure 5.1.1 DeltaLink H1 Mailbox Definition.

The Siemens H1 mailbox definition panel allows the user to initialise one Mailbox Tag for the Sinec H1 protocol driver. Only one Mailbox is needed for full performance of IMX. Specify the following information.

- **Protocol Driver Mailbox Tag**
  Tag name of the Sinec H1 protocol mailbox element that the developer defines, to be referenced by a decoder task (**N.B.** IMX must be supported by the decoder task). The decoder task uses this mailbox to send requests to the Sinec H1 protocol driver.

  entry: Required.
  entry type: Standard FactoryLink tag name.
  valid entry: MAILBOX.

- **Local Tsap**
  Local TSAP identifier of the FactoryLink workstation. The entry made here must be identical with the remote TSAP identifier of the connected device(s).

  entry: Required.
  entry type: Alphanumeric string.
  valid type: String of up to 8 characters.

- **Description**
  Description of the Sinec H1 protocol driver mailbox element defined by the developer.

  valid entry: Output only.

5.2. Decoder definition

From the display of all the panels, select the *DeltaLink H1 Decoder definition* panel.

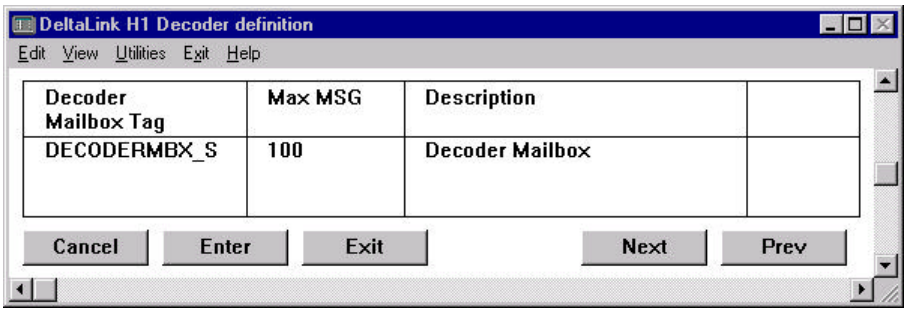


Figure 5.2.1 DeltaLink H1 Decoder Definition.

The Sinec H1 Decoder Definition panel allows the user to specify one or more decoder(s). Every decoder mailbox tag specifies a different decoder. Only one mailbox tag, for every decoder, is needed for full performance of IMX. Specify the following information.

- **Decoder Mailbox Tag**
  Tag name of a decoder mailbox element, to be referenced by the Sinec H1 protocol driver task (**N.B.** IMX must be supported by the decoder task). The decoder task uses this mailbox to receive data from the Sinec H1 protocol driver.

  entry: Required.
  entry type: Standard FactoryLink tag name.
  valid entry: MAILBOX.

- **Max MSG**
  The maximum number of requests for the protocol driver, which can be queued in the mailbox tag. Recommended is a value of 100 messages. The number of messages is limited by the size of an integer value (9999), and practically by the amount of available memory. The memory needed for a request depends on the size of the configured datasets.

  entry: Required / Default: 100.
  entry type: Decimal number.
  valid entry: Positive integer 1 .. 9999.

- **Description**
  Description of the decoder mailbox element defined by the developer.

  valid entry: Output only.

5.3. Device definition

From the display of all the panels, select the *DeltaLink H1 Device Definition* panel.

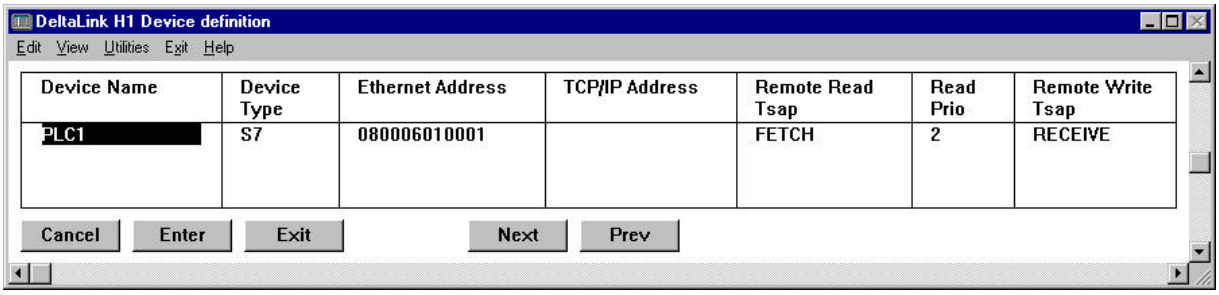


Figure 5.3.1 DeltaLink H1 Device Definition.

The DeltaLink H1 Device definition panel allows the user to specify logical devices, for the communication. Different logical device names may be used for one communication device (the same Ethernet address). If the same Ethernet address is used make sure that the next entries are not identical: Device Name, Remote TSAPs. Specify the following information.

- **Device Name**
  Logical name assigned by the user to represent a particular communication device. This field is used as a selection criterion for the next panel: *DeltaLink H1 Dataset Definition*

  entry: Required.
  entry type: Alphanumeric string.
  valid type: String of up to 16 characters.

- **Device Type**
  PLC type of the logical device. This type specifies wich dataset types are present in the PLC.

  entry: Required.
  entry type: Alphanumeric string.
  valid entries: S5, S7, S7RW, SYM, TI.
  description:

PLC type	Description
S5	Siemens S5 PLC, standard H1
S7	Siemens S7 PLC, with handling blocks, read/write information is supported
S7RW	Siemens S7 PLC, without handling blocks, no read/write information
SYM	Symadyn controller, no read/write information
TI	Siemens Texas Instruments PLC

- **Ethernet address**
  The PLC Ethernet address. A six byte (00 - FF) address to identify the PLC in the Ethernet network for example 080006010001. This address must be specified in case the H1-TP4 protocol is used for communication with the PLC. This field can be left blank in case the H1-TCP protocol is used and the next field has to be specified. It is also possible to change the MAC address dynamically in case a tag has been specified. When defining a tag then a default value as a MAC address must be specified.

  entry: Optional.
  entry type: Hexadecimal string or TAG.
  valid entry: String of up to 12 characters.

- **TCP/IP address**
  The PLC TCP/IP address. This must address according to the syntax of the Internet Protocol i.e. 192.0.0.0. This address must be specified in case the H1-TCP protocol is used for communication with the PLC. This field can be left blank in case the H1-TP4 protocol is used and the previous field has to be specified. It is also possible to change the IP address dynamically in case a tag has been specified. When defining a tag then a default value as a IP address must be specified.

  This option is not supported under windows NT at the moment.

  entry: Optional.
  entry type: ASCII string or TAG .
  valid entry: IP address, 4 digits separated by a dot.

- **Remote Read Tsap**
  Remote TSAP identifier for a fetch connection. Must be identical to the local Fetch Tsap of the logical device.

  entry: Required /default: FETCH.
  entry type: Alphanumeric string.
  valid type: String of up to 8 characters.

- **Read Prio**
  Priority of the read commands.

  Prio 2 Normal data transfer, with maintained connection to a remote node. The data buffers are dynamically build, during data transfer.
  Prio 4 Normal data transfer, with connection establishment and termination. The connection and data buffer are build when data is transferred, and are terminated at the end of the data transfer.

  entry: Required / default: 2.
  entry type: Decimal number.
  valid entry: Positive integer 2, 4.

- **Remote Write Tsap**
  Remote TSAP identifier for a write connection. Must be identical to the local Receive Tsap of the logical device.

  entry: Required /default: RECEIVE.
  entry type: Alphanumeric string.
  valid type: String of up to 8 characters.

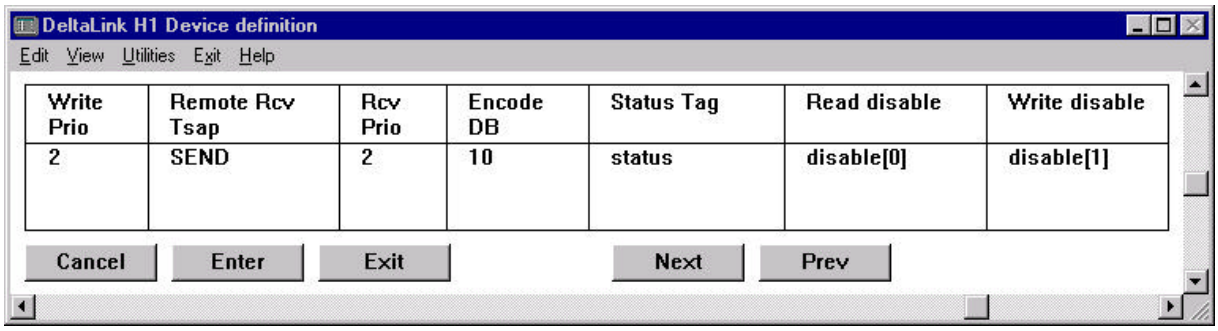


Figure 5.3.2 DeltaLink H1 Device definition.

- **Write Prio**
  Priority of the write commands. For a description of the priority levels see *Fetch Prio*.

entry: Required / default: 2.
entry type: Decimal number.
valid entry: Positive integer 2, 4.

- **Remote Rcv Tsap**

Remote TSAP identifier for a unsolicited receive connection. Must be identical to the local Send Tsap of the logical device.

entry: Required /default: SEND.
entry type: Alphanumeric string.
valid type: String of up to 8 characters.

- **Rcv Prio**

Priority of the unsolicited receive commands. For a description of the priority levels see *Read Prio*. The RCV Prio field will not be used in the protocol driver because the PLC Communcations card settings determines what kind of priority this connection will have. This field must be used as an information field for the developer.
The unsolicited receive connection used with PRIO 4 is a special case because in principal Siemens doesn't support a SEND command with PRIO 4 and 'Read/Write yes' configured for the Siemens S5 range (*Appendix F*). As a consequence DeltaLink has developed a Function Block (FB 2) for S5 to circumvent this problem. This PRIO 4 has been limited for sending Data Blocks only (DBs). When with the SEND command a PRIO 4 connection has been chosen, this FB 2 must be used in the STEP5 program for sending data with the unsolicited receive function in the protocol driver. For the normal PRIO 2 connection no additional software is needed in the PLC. For a description of the PRIO 4 FB2 refer to *Appendix F*.

entry: Required / default: 2.
entry type: Decimal number.
valid entry: Positive integer 2, 4.

- **Encode DB**

Data Block (DB) number for encoded write commands. Coded information is placed in this DB, encoding is performed by the PLC. A standard function block (FB10) is provided by DeltaLink for decode actions inside the PLC, for specific information see also *Appendix F* and *Appendix G*.

entry: Optional.
entry type: Decimal number.
valid entry: Positive integer 1 .255.

- **Status Tag**

Real-time database element written to by the Sinec H1 protocol driver task to indicate the status of a communication command. The status element can be referenced by any task to handle communication error situations. A value of zero indicates no error (success), any other values represents an error, for a detailed list of communication errors see *Appendix D*.

entry: Optional.
entry type: Standard FactoryLink tag name.
valid entry: ANALOG.

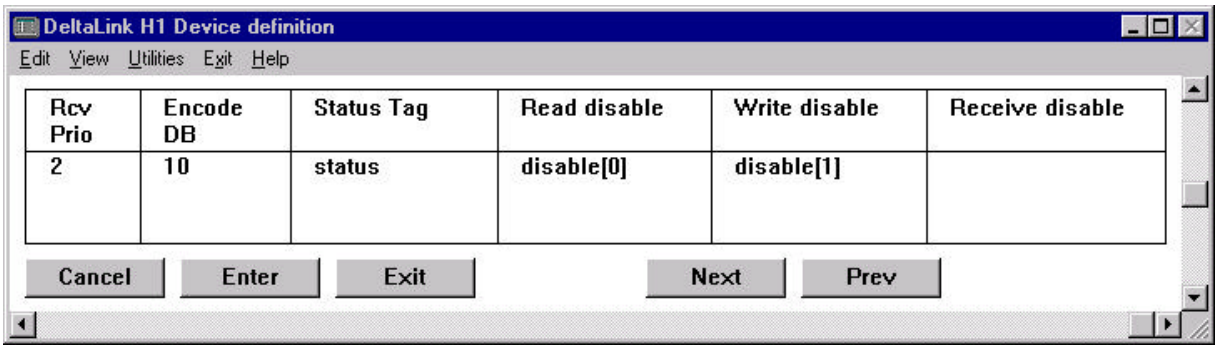


Figure 5.3.3 DeltaLink H1 Device definition.

- **Read disable**
  Real-time digital database element used to enable/disable read commands for the logical device. Read commands are enabled in case there is no tag defined, or the status of the digital tag is OFF. Read commands are disabled if a tag is defined and the status of the tag is ON.

  entry: Optional.
  entry type: Standard FactoryLink tag name.
  valid entry: DIGITAL.

- **Write disable**
  Real-time digital database element used to enable/disable write commands for the logical device. Write commands are enabled in case there is no tag defined, or the status of the digital tag is OFF. Write commands are disabled if a tag is defined and the status of the tag is ON.

  entry: Optional.
  entry type: Standard FactoryLink tag name.
  valid entry: DIGITAL.

- **Receive disable**
  Real-time digital database element used to enable/disable receive commands for the logical device. Receive commands are enabled in case there is no tag defined, or the status of the digital tag is OFF. Receive commands are disabled if a tag is defined and the status of the tag is ON.

  entry: Optional.
  entry type: Standard FactoryLink tag name.
  valid entry: DIGITAL.

5.4. Dataset definition

From the display of all the panels, select the *DeltaLink H1 Dataset definition* panel.

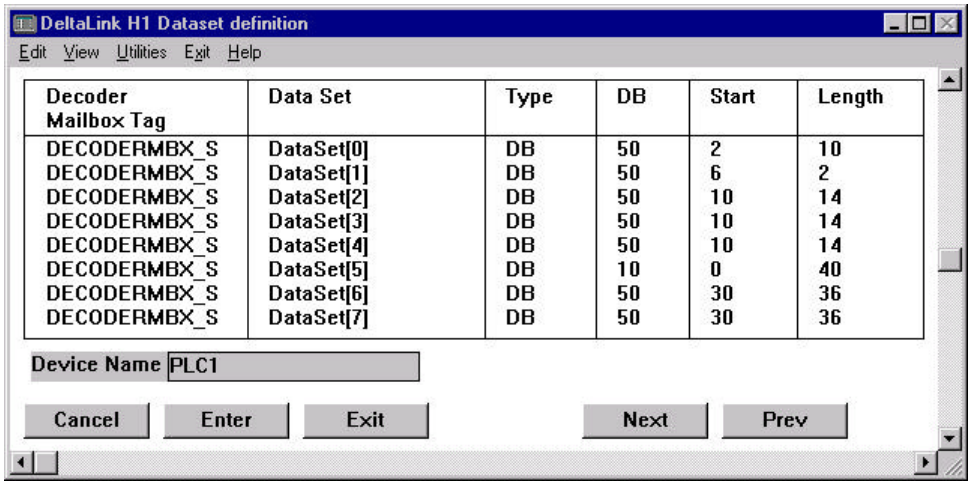


Figure 5.4.1 DeltaLink H1 Dataset definition.

The Sinec H1 Dataset Definition panel allows the user to specify datasets for a particular logical device. Different logical device names are selected in the panel *DeltaLink H1 Device Definition*. The name of the logical device is displayed in the field *Device Name*, at the bottom of the table. Specify the following information.

- **Decoder Mailbox Tag**
  Tag name of a decoder mailbox element, to be referenced by the Sinec H1 protocol driver task (**N.B.** IMX must be supported by the decoder task). The decoder task uses this mailbox to receive data from the Sinec H1 protocol driver.

  entry: Required.
  entry type: Standard FactoryLink tag name.
  valid entry: MAILBOX.

- **Dataset**
  Tag name representing a (unique) logical name for a dataset. The tag is used internally in the decoder and protocol driver for activating a dataset read from the external device. A dataset is referenced, internally by the decoder and the protocol driver, by the tag name. Therefore the tag specified in this field must be unique.

  entry: Required.
  entry type: Standard FactoryLink tag name.
  valid entry: DIGITAL.

- **Type**
  Type definition of the data area in the logical device (PLC). Note that exception write behaviour depends on the type of data area. To know how an exception write is executed, one should regard the boundary of the data area and the element size involved with the exception write.

  entry: Required.
  entry type: Alphanumeric string.
  valid entries: DB, FB, IB, OB, CW, TW, SW, PB, EB, DX.
  description:

Siemens S5 type	
Data type entry	Description
DB	Data Block

Siemens S5 type	
Data type entry	Description
FB	Flag byte
IB	Input Byte
OB	Output byte
PB	peripheral byte
CW	counter word
TW	timer word
SW	system data word
DX	expanded data block
DE	Data Block in external memory
EB	expanded peripheral byte

Siemens S7 type	
Data type entry	Description
DB	Data Block
FB	Flag byte
IB	Input Byte
OB	Output byte

Siemens TI type	
Data type entry	Description
V	Variable
K	Constant
X	Discrete Input (X)
Y	Discrete Output (Y)
C	Control Register (CR)
XP	Discrete Input Packed (X)
YP	Discrete Output Packed (Y)
CP	Control Register Packed (CR)
WX	Word Input (WX or 7MT)
WY	Word Output (WY)
TCP	Timer or Counter Preset (TCP)
TCC	Timer or Counter Current (TCC)
DSP	Drum Step Preset (DSP)
DSC	Drum Step Current (DSC)
DCP	Drum Counter Preset (DCP)
STW	System Status words
DCC	Drum Current count
VF	Variable Memory
KF	Constant Memory
LS	Loop Status
LM	Loop Mode
LKC	Loop Gain
LTI	Loop Reset time (minutes)
LTD	Loop Rate time (minutes)
LHA	Loop High Alarm Limit
LLA	Loop Low Alarm Limit
LPV	Loop Process Variable
LPVH	Loop High Process Variable
LPVL	Loop Low Process Variable
LODA	Loop Orange Deviation
LYDA	Loop Yellow Deviation
LTS	Loop Sample Rate (seconds)
LSP	Loop Set Point
LMN	Loop Output (percent)
LVF	Loop V-flags
LCF	Loop Control Flags
LRSF	Loop Ramp/Soak Status
LERR	Loop Error
LMX	Loop Bias

Siemens TI type	
Data type entry	Description
LHHA	Loop Process Variable High High Alarm
LLLA	Loop Process Variable Low Low Alarm
LRCA	Loop Rate of Change Alarm
LSPH	Loop Setpoint High Limit
LSPL	Loop Setpoint Low Limit
LADB	Loop Alarm Deadband
LHAR	Loop Raw High Alarm Limit
LLAR	Loop Raw Low Alarm Limit
LPVR	Loop Raw Process Variable
LODAR	Loop Raw Orange Dev Alarm Limit
LYDAR	Loop Raw Yellow Dev Alarm Limit
LMNR	Loop Raw output
LSPR	Loop Raw setpoint
LERRR	Loop Raw Error
LHHAR	Loop Raw High/High Alarm Limit
LLLAR	Loop Raw Low/Low Alarm Limit
LADBR	Loop Raw Alarm Deadband
LMXR	Loop Raw Bias
LSPLR	Loop Raw Setpoint Low Limit
LSPHR	Loop Raw Setpoint High Limit
LCFH	Loop Most-sig word loop C-flags
LCFL	Loop Least-sig word loop C-flags
LKD	Loop Derative Gain Limiting Coef
LRSN	Loop Ramp Soak Step Number
LACK	Loop Alarm Acknowledge Flags
AHA	High Alarm Limit
ALA	Alarm Limit
APV	Process Variable
APVH	Process Variable High Limit
APVL	Process Variable Low Limit
AODA	Orange Deviation Alarm Limit
AYDA	Yellow Deviation Alarm Limit
ATS	Sample Rate in seconds
ASP	Alarm Setpoint
AVF	Alarm Variable Flags
ACF	Alarm Control Flag
AERR	Alarm Error
AHHA	High High Alarm Limit
ALLA	Low Low Alarm Limit
ARCA	Rate of Change Alarm Limit
ASPH	Setpoint High Limit
ASPL	Setpoint Low Limit
AADB	Alarm Deadband
AHAR	Raw High Alarm Limit
ALAR	Raw Low Alarm Limit
APVR	Raw Process Variable
AODAR	Raw Orange Deviation Alarm Limit
AYDAR	Raw Yellow Deviation Alarm Limit
ASPR	Alarm Raw Setpoint
AADBR	Raw Alarm Deadband
AERRR	Alarm Raw Error
AHHAR	Raw High-High Alarm Limit
ALLAR	Raw Low-Low Alarm Limit
ASPLR	Raw Setpoint Low Limit
ASPHR	Raw Setpoint High Limit
ACFH	Most-sig word Alarm C-flags
ACFL	Least-sig word Alarm C-flags
AACK	Alarm Acknowledge Flags
VMM	VME (TI575 only) A24 space
VMS	VME (TI575 only) 16 space

Siemens TI type	
Data type entry	Description
GZ	App Z global variables
GY	App Y global variables
GX	App X global variables
GW	App W global variables
GV	App V global variables
GU	App U global variables
GT	App T global variables
GS	App S global variables
GR	App R global variables
GQ	App Q global variables
GP	App P global variables
GO	App O global variables
GN	App N global variables
GM	App M global variables
GL	App L global variables
GK	App K global variables
GJ	App J global variables
GI	App I global variables
GH	App H global variables
GG	App G global variables
GF	App F global variables
GE	App E global variables
GD	App D global variables
GC	App C global variables
GB	App B global variables
GA	App A global variables
G	Local App global variables

- **DB**

  Data block (DB), External Data block (DE) or Data block eXtended (DX) number. Only used if the *Type* is DB, DE or DX.
  In case a TI data type has been specified this number will be used for addressing pages of 64Kb within the data type. Usually this will be set to zero. For example, to address variable with address 65536 in V memory specify DBNR 1 and start address 1.

  entry: Optional.
  entry type: Decimal number.
  valid entry: Positive integer 1 ..255.

- **Start**

  Start address number of the dataset.

  entry: Required / Default: 0.
  entry type: Decimal number.
  valid entry: Positive integer 0 ..2048.

- **Length**

  Length of the dataset in bytes or words, depending on the type of the dataset. For the types DB, DE and DX the length is expressed in words, the data boundary is words. For all other types the data boundary is byte, which means that the length is expressed in bytes. Maximum length depends on the type of data set and PLC type, for further details see the Siemens manuals.

  entry: Required / Default: 1.
  entry type: Decimal number.
  valid entry: Positive integer 1 ..2048.

This page is left blank intentionally.

Appendix A. Communication OSI protocol stack installation

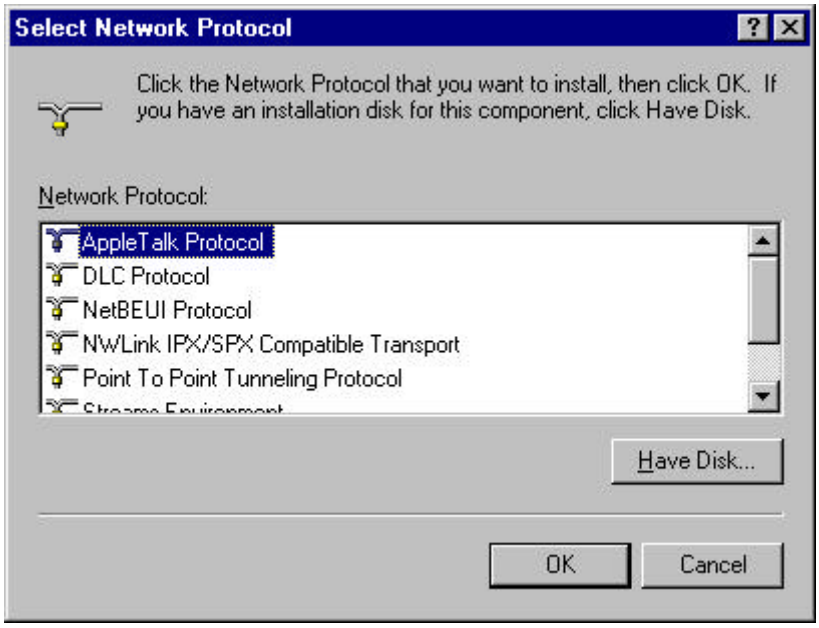
A.1 OSI protocol stack for Windows NTI

On the Windows NT platform any Ethernet card with support for Windows NT can be used. This card can be used to run the H1 stack but also in combination with other stacks simultaneously. The stack used for the H1 communication is called the Windows NT TP4 stack. The next section gives a short description of the installation of the stack and how to configure it. This configuration is very important because otherwise H1 communication will not work.

Installation of stack

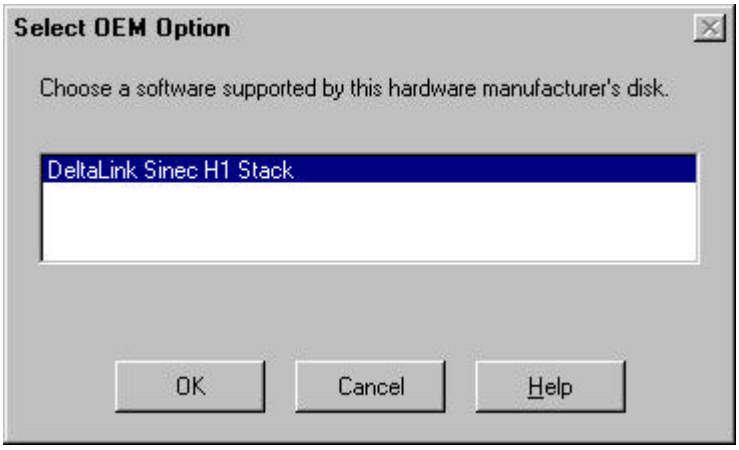
The TP4 stack delivered with the Sinec H1 driver must be installed through the ***Control Panel Network*** applet. The user must have administrator privileges to install the TP4 stack. In this applet the following steps must be taken:

- From the control panel, select the **Network** applet and the system presents the network settings dialog box.
- Choose the **Protocols** tab in the applet and within this tab choose the **add** button for adding another protocol.

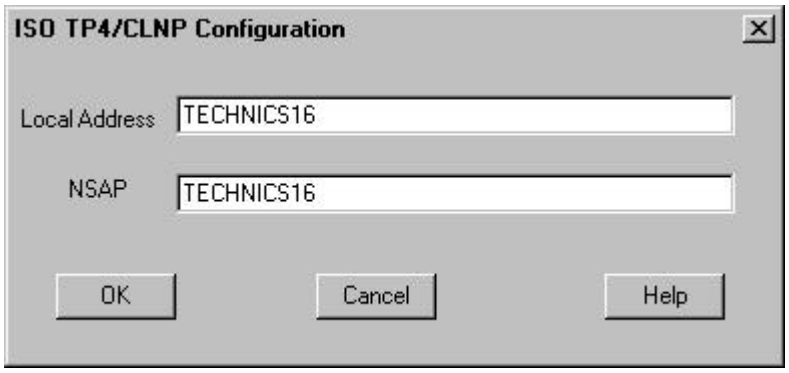


- In this applet choose the have disk button and specify the location of the OSI protocol stack. This can be on floppy or on CDROM or hard-disk. In case of the CDROM it is advised to copy first the complete stack to hard-disk and install from there.

- The following pop up window must appear after specifying the right path and pressing the OK button. Press the OK button and the protocol stack will start to install.



- Before closing the network applet the install procedure will come up with the following window. The local address and the NSAP will be set on default to your computer name and need not to be changed. Pres the OK button to finish the install procedure and reboot the system to activate the protocol stack.



Configuration

For use with the Sinec H1 driver the stack must be configured. This configuartion will be done automatically when installing the Sinec H1 module through the **set_parm** utility, which resides in the **bin** directory of the sinec H1 install directory. This utility configures a few parameters in the registry of NT. The following entries in the registry are altered and can be checked through the **regedit** utility:

HKEY_LOCAL_MACHINE/System/CurrentControlSet/Services/IsoTp/Parameters

In this directory there are three sub-directories: CLNP, ESIS and TP4. In the directory **CLNP** the following parameter is changed so it has the following value:

NullNetworkLayer:REG_DWORD=0x1

In the directory **TP4** the following parameter is changed so it has the following value:

PropNoChksum:REG_DWORD=0x1

The set_parm utility will stop the protocol stack, make these in the registry and automatically start the protocol stack again. That's why the preferred order of installing is first the protocol stack, reboot the system and then install the Sinec H1 driver.

Appendix B. Command line parameters

The protocol driver accepts several command line parameters, these can be configured with the configuration manager in the 'System Configuration' table, column 'Program Arguments'. An argument consists out of first a minus sign ('-'), followed by the a letter specifying the option. After the letter an optional number can be present, if this is supported by the option.

Option	Description
-d*n*	Debug option, the level of debug information is set with the number *n*. The range of this number is from 1 until 9. If no number is specified the default level will be 1. The debug output will be visible in the 'window' of the protocol driver, that of the run time manager.
-l*n*	Same as the previous option, difference is the output device. For this option an ASCII log file is generated, being the file: {FLAPP}/{FLNAME}/{FLDOMAIN}/log/sinec_h1.log
-m*n*	Defines the size of the pipe buffer for buffering IMX commands from the loxlator. The size is specified in kilobytes. The default size 32 Kb which means a value of 32.
-s*n*	Starts and stops the TP4 stack. The parameter specifies a sleep value in seconds before starting the stack after starting the sinec h1 driver. When stopping the sinec h1 driver the stack will also be stopped.
-c	Automatically disables the connection rebuild after an error on the connection. The disable tag must be forced changed to 0 again for enabling rebuild of the connection. At the first IMX command on the connection, the connection will be tried to rebuild. This option must be used in case PLC's are configured but not connected to the network or when frequently connections are lost on the network with a high communication load on that connection.
-LS*n*	Local Station id with number *n*, needed if the protocol driver and decoder reside on different nodes (or FactoryLink workstations). This must be an unique (station) number for the network.

This page is left blank intentionally.

Appendix C. Sinec H1 protocol driver error codes

The error code is returned to the user in a user-defined status tag. These error codes will also be printed with the message of the DeltaLink Sinec H1 protocol driver in the run-time manager. The errors can be generated from different parts of the driver which will be listed here:

Error #	XTI layer communication error
4	Socket marked as non-blocking
9	Invalid flags for socket
13	Access error
14	Socket failure
22	Unable to establish connection, partner not found, invalid TSAP
35	Blocking failed
36	Blocking call in progress
37	Command already complete
38	Descriptor is not a socket
39	Destination address required
40	Data size too large
41	Protocol is of wrong type for this socket
42	Protocol option not supported for this socket type
43	Protocol is not supported
44	Socket type not supported by this address family
45	Option not supported
46	Address family not supported by this protocol
47	Address family not supported by this protocol
48	Address is in use
49	Address not available from local machine
50	Network subsystem is down
51	Network cannot be reached
52	Connection has been dropped
53	Connection has been aborted
54	Connection has been reset
55	No buffer space available
56	Socket is already connected
57	Socket is not connected
58	Socket has been shut down
59	To many references
60	Command timed out
61	Connection refused
91	Network subsystem not ready
92	Version not supported
93	WSAStartup() has not been successfully called
101	Unknown data area
102	Encode value not supported
103	Invalid function code
104	The size off the PG packet was too small
105	Undefined encode destination
106	Undefined exception write
107	No memory available any more
108	Invalid data area
109	Length for read/write is zero
110	Protocol error detected by PLC

Error #	H1 data package errors
201	Unidentified data received.
202	Encode value not supported.
203	Unknown encode type.
204	The size of the H1 packet was too small
205	Error in received H1 protocol header

Error #	H1 data package errors
300	No error.
301	Incorrect Q/Z type at handling block.
302	Area not present in AG.
303	Area in AG too small.
304	AKD Error in the AG.
305	Error with condition code word.
306	No valid ORG format.
307	No free data buffer.
308	No free transport links.
309	Remote error with Read/Write.
310	Data Link error.
311	Handshake error.
312	Initiation error.
313	Abort after reset.
314	Job with boot strap function.
315	Job not present.
316	Area in AG too small.

Error #	FactoryLink errors
401	Internal error
402	Out of memory
403	Operating system error
404	Initialization not successful
405	Initialization not successful
406	Incorrect function
407	Incorrect argument
408	Incorrect data
409	Bad tag
410	Null pointer assignment
411	Change flag not set
412	Procedure table full
413	Bad procedure name
414	Bad user name
415	Bad option
416	Incorrect checksum
417	No options
418	No key
419	Bad key
420	No port available
421	Port busy
422	FL already active
423	No lock
424	Lock failed
425	Lock expired
426	Wait failed
427	Termination flag set
428	Q-size to big
429	Q-size changed
430	No tag list
431	Tag list changed
432	Wake up failed
433	No signals
434	Signaled
435	Not a mailbox
436	No messages
437	Access denied
438	Attribute failure
439	Invalid attribute
440	Attribute not defined
441	Application exists

Error #	FactoryLink errors
442	RTDB does not exist
443	No task bit
444	Not a lite task

Error #	IMX errors
450	Bad message type
451	Message with dataset control tag not found in queue
452	No messages available to query
453	Bad receive mailbox tag
454	Bad mailbox send tag
455	Bad dataset control tag
456	Message cannot be adjusted
457	Operation to big for variable
458	Unknown boundary
459	Function not supported
460	No message for this index present
461	The remote dataset Is not defined on this system
462	The received dataset was not registered
463	The message is not queued
464	Message is rejected due error in the remote IMX
465	Illegal method of addressing bits on bit boundary
466	Element cannot be written
467	Invalid buffer specified
468	Block write function impossible
469	Maximum number of messages in MBX reached
470	No memory left
471	Error registering standard dataset
472	Error writing message in pipe
473	Not supported IMX message
474	Error creating pipe
475	Error starting threat
476	Error server connecting to pipe
477	Error child connecting to pipe
478	Error reading the pipe
479	Error number of bytes read from pipe
480	Error writing into the pipe
481	Error number of bytes written into the pipe
482	Error reading dataset from the mailbox
483	No communication buffers assigned
484	Error decrementing semaphore
485	Error writing to pipe, no space left
486	Error querying no. of messages for decoder
487	Max. No. of messages in decoder MBX reached

Error #	Sinec h1 driver status / errors
350	Read functionality has been disabled by the user
351	Write fnctionality has been disabled by the user
353	Invalid addressing method on bit boundary
354	Exception write not possible to destination
355	SINEC_H1 driver has shutdown normally
356	an unknown error occured in the comm layer
357	No data has been received
500	invalid addressing method on bit boundary
501	exception write not possible to destination
502	Sinec H1 driver has shutdown
503	an unknown error occurred in the comm layer

Appendix D. Messages

If an error condition occurs in the protocol driver task during run-time mode, a message to that effect will appear on the runtime manager graphics screen to the right of "SINEC_H1". The error messages that may be displayed are as follows:

Running in DEMO mode

The task did not found the DeltaLink protection code, when starting up. The option file could not be present or damaged. This is a non fatal error, however the task continues running in DEMO mode. The DEMO has a limitation: timed for 5 hours. After shutdown the task can not be restarted without stopping the complete FactoryLink application.

Demo shutdown, licensed to run 5 hours

The task did not found the DeltaLink protection code, when starting up and continued running in DEMO mode. The DEMO has a limitation: timed for 5 hours. After shutdown the task can not be restarted without stopping the complete FactoryLink application.

Demo restart prohibited, restart FactoryLink

The task did not found the DeltaLink protection code, when starting up. The option file could not be present or damaged. This is a non fatal error, however the task continues running in DEMO mode. The DEMO has a limitation: timed for 5 hours. After shutdown the task can not be restarted without stopping the complete FactoryLink application.

Out of RAM

There is not enough RAM memory to load the complete configuration and/or task. This is a fatal error.

Error (%d) reading RTDB element

An error occurred reading a tag value. The error number is displayed.

Error (%d) forcing RTDB element

An error occurred forcing a tag value. The error number is displayed.

Error (%d) writing RTDB element

An error occurred writing a tag value. The error number is displayed.

Error (%d) retrieving event

An error occurred while waiting for an event to happen. The error number is displayed.

Can't open CT file

The task was unable to open the configuration table file, generally because it does not exist. This is a fatal error.

No triggers defined

The task was able to open the configuration table file, but contained not enough information to continue running. This is a fatal error.

Error reading CT index

An error occurred reading the index of the CT file. This is a fatal error.

Error reading CT header

An error occurred reading the header of a table in the CT file. This is a fatal error.

Error reading CT record

An error occurred reading a record of a table in the CT file. This is a fatal error.

Initialization critical parameters failed

Use of critical sections will not be possible, but is still required for full functionality of the protocol driver. This is a fatal error.

Event not registered to FL kernel
Tag event is not registered to the FactoryLink kernel. This is a fatal error.

No decoders defined
There are no decoders (or mailbox) defined. This is a fatal error.

No datasets defined
No dataset is defined, the protocol driver has nothing to do. Therefore this considered to be a fatal error.

Unknown decoder table %s record %d type %d
The protocol driver configuration table file contains an unknown decoder mailbox tag, for the specified table, record number and type of dataset. This is a fatal error.

IMX Bad Mailbox TAG table %s record %d type %d
IMX error, internal error of the protocol driver task. Specified are table number, record number and type number.

IMX (%d) Initialization failed
IMX initialization failed, the error number is reported. This is a fatal error.

IMX (%d) dataset registration failed %s
Registration of the reported dataset failed, the cause can be determined by the error number. This is a fatal error.

IMX semaphore creation failed
Creation of semaphores, needed for IMX failed. This is a fatal error.

IMX failed to start thread
The protocol driver could not start a thread. This is a fatal error.

IMX maximum msg for decoder %d reached
The reported decoder failed to process the queued messages, the configured maximum length of the queue is reached. The protocol driver stops queuing until the number decreases.

IMX (%d) send error to mailbox %s
The protocol driver could not queue a message for the reported decoder due to an error, the corresponding error number is reported.

IMX (%d) reply to mailbox %s
The protocol driver could not queue a message for the reported decoder due to an error, the corresponding error number is reported.

Com building device failed
Communication error building of a communication device failed. This is a fatal error.

Com unknown DataSet of type %d
Unknown dataset type specified, the type invalid type id is reported. This is a fatal error.

Com init comm. layer %d
An error occurred during initialising the communication layer Specified is the error number.

Com (%d) connecting '%s' %s
An error occurred during connecting to a device. Specified are the device name, functionality and the error number.

Com (%d) listenening %s %s

An error occurred during listening. Specified is the error number.

Com (%d) accepting connection %s

An error occurred during accepting a connection to a device. Specified are the device name and the error number.

Com (%d) disconnecting '%s' %d

An error occurred during disconnecting from a device. Specified are the device name, functionality and the error number.

Com (%d) read %s

An error occurred during reading from a device. Specified are the device name and the error number.

Com (%d) reading for exception %s

An error occurred during reading from a device for an exception write. Specified are the device name and the error number.

Com (%d) encode write %s

An error occurred during writing for an encoded write to a device. Specified are the device name and the error number.

Com (%d) write %s

An error occurred during writing to a device. Specified are the device name and the error number.

Com (%d) receive %s

An error occurred during receiving from a device. Specified are the device name and the error number.

Com (%d) reject connection %s, %s

An incoming connection request is rejected. Specified is the error number.

This page is left blank intentionally.

Appendix E. Siemens S5-PLC

E.1 Encoded write handling

The Sinec H1 protocol driver is capable to perform an encoded write. An encoded write causes the protocol driver to place coded information in a data block in the PLC instead of directly updating or changing a variable.

The information, placed in the PLC by an encoded write, should be decoded by a PLC-program in order to update/change the correct variable. A standard function block (FB10) is provided by DeltaLink for decode actions inside the PLC. The standard function block FB10 is called "ENCO-WRT", and uses FB11, FB12 and the data block opened just before calling the function block FB10. The data block opened before calling the function block FB10 must contain the information from an encoded write. The protocol driver places the information in a special way in the encode DB. Internally the next flag bytes are used: FB240 .. FB255.

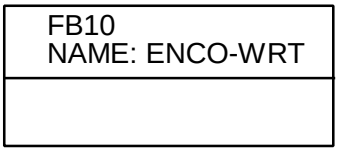


Figure E.1.1 Encoded Write Function Block.

Encode DB		
Data word	Type	Description
DW0	KF	Internal use FB10
DW1	KY	Destination, destination type
DW2	KY	DB/DX no., address no.
DW3	KY	Bit no., Change flags
DW4	KF	Value, low word
DW5	KF	Value, high word
DW6	KH	Internal use FB10
DW7	KH	Internal use FB10
DW8	KH	Internal use FB10
DW9	KH	Internal use FB10
DW10	KF	Error code from FB10
DW11	KY	Copy of DW1 for an error
DW12	KY	Copy of DW2 for an error
DW13	KY	Copy of DW3 for an error
DW14	KF	Copy of DW4 for an error
DW15	KF	Copy of DW5 for an error
DW16	KH	Internal use FB10
DW17	KH	Internal use FB10
DW18	KH	Internal use FB10
DW19	KH	Internal use FB10
DW20	KH	Internal use FB10

Figure E.1.2 Encode Data Block.

Destination: the information in this byte indicates the destination of the value which should be updated.

Value		**Description**
M	=	Flag type.
D	=	Data block.
X	=	Data block eXtended.

T	=	Timer.
Z	=	Counter.
A	=	Output.
E	=	Input.
P	=	Peripheral output.
Q	=	Extended periphery output.
S	=	System address.

Destination type: the information in this byte indicates the destination type of the value which should be updated.

Value		Description
F	=	Flag for DB, DX, F, A or E.
B	=	Byte for DB, DX, F, A, E, P or Q.
W	=	Word for DB, DX, F, A, E, P or Q.
L	=	Data left for DB or DX.
R	=	Data right for DB or DX.
D	=	Double word, for long integer or float.

DB/DX number: The information in this byte is the number of a DB or DX. If the destination is not D and not X the value in this byte is irrelevant. The value is between 0 and 255.

Address number: address number of the byte, word, data word, timer or counter. The value is between 0 and 255.

Bit number: the number of the bit in a word or a flag. If the destination type is not flag, the value is irrelevant.

Change flags: For every encoded write the protocol driver writes the value 255 in this byte. Function block FB10 uses this value to detect if there is new information.

Value: value may be up to words, depending on the type of data. For flag and byte values DR4 is used. For word values DW4 is used. And for long integers and float DW4 plus DW5 is used.

Error code: the different error codes can be:

Error	Description
1	Unknown destination.
2	Unknown destination type for destination is D or X.
3	Incorrect bit number for destination is D or X.
4	Unknown destination type for destination is M.
5	Incorrect bit number for destination is M.
6	Unknown destination type for destination is A.
7	Incorrect bit number for destination is A.
8	Unknown destination type for destination is E.
9	Incorrect bit number for destination is E.
10	Unknown destination type for destination is P.
11	Unknown destination type for destination is Q.

E.2 Unsolicited receive PRIO 4

The communication cards CP535 and CP143 are not capable of sending data including read / write information with PRIO 4. The consequences are that the PLC can not send data with PRIO 4 to a FactoryLink workstation to perform an unsolicited receive job. To overcome this problem DeltaLink included a function block (FB2) in the shipment for your H1 driver.
The function block FB2 (UNSOL4) activates a send command on the CP535 or CP143 for PRIO 4.

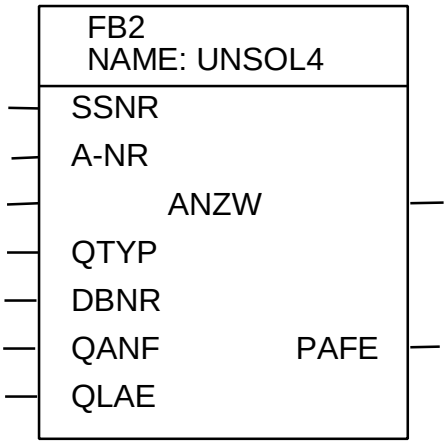


Figure F.1 Unsolicited Receive Function Block.

The function block FB2 uses data block DB11, Flag word MW12. The contents of flag word MW12 and data block DB11 should not be altered by the user. The function block uses also the standard SIEMENS communication functions for SEND and CONTROL (for example: if the PLC type is S-135U, the function blocks will be FB120 and FB123). The parameters for FB2 are identical to the parameters used for the standard SIEMENS communication functions.

Communication port number **SSNR**:
A constant of type KY = x, y.

Command number **A-NR**:
A constant of type KY = x, y.
x = 0 : Value of x is meaningless and will not be evaluated.
y >< 0 : Number of the command for a send-direct. The value zero is not allowed.

Command status **ANZW**:
Status word for the send command, only a flag word is allowed. The use of a data word will give unpredictable results. For a detailed description of the status word see the SIEMENS manual.

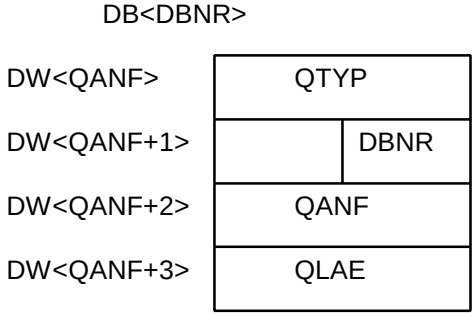


Figure F.2 Indirect parameters.

Source type **QTYP**:
A constant of type KC:

Allowed entries 'DB' or 'XX', meaning that the next parameters are direct or indirect.
For indirect parameters QTYP is 'XX', DBNR is the number of the data block where the actual parameters are stored, the start address for the parameters is set with QANF.

Data block number **DBNR**:
A constant of type KY = x, y.
x = 0 : Value of x is meaningless and will not be evaluated.
y >< 0 : Number of the data block containing the source data, which will be send.

Source start address **QANF**:
A constant of type KF, indicating the data word number of the start address for the source data-area. The value of **QANF** must be greater or equal to 8. The eight data words prior to QANF are used internally by FB2. These data words will be overwritten by the function block FB2.

Source length **QLAE**:
A constant of type KF, indicating the length in data words of the source data-area.

Error byte **PAFE**:
Indication of parameter errors. Valid entries are a flag byte or an output byte. For a detailed description of the error byte see the SIEMENS manual.

E.3 CP143/CP535 Configuration

This section displays the configuration of a CP535 or CP143 card.

```
   VERBINDUNGSBAUSTEIN

VOM EIGENEN AG:

SSNR          :   0         ANR               :  1
AUFTRAGSART   : FETCH       AKTIV/PASSIV (A/P): P


ZUM FREMDEN AG:

BUKZ:        STKZ:        BGKZ:
ETHERNETADRESSE  : 00AA000682CE H    SSNR:         ANR:

MULTICAST (J/N): N   MULTICASTKREIS:      ETHERNETADRESSE:              H

DATAGRAMM (J/N): N
PRIORITAET     : 2   READ/WRITE (J/N): J


QUELLE/ZIEL    :                LAENGE :
ANZEIGENWORT   :
INTERPRETER    :                ADRESSE:      :      H


EIGENER TSAP-ID: LAENGE: 5    HEXA: 46 45 54 43 48          ASC: FETCH
FREMDER TSAP-ID: LAENGE: 5    HEXA: 46 4C 49 4E 4B          ASC: FLINK
FREMDER NSAP-ID: LAENGE: 12   ASC :

ANZAHL DER AUFTRAEGE PRO TSAP: 1
```

```
   VERBINDUNGSBAUSTEIN

VOM EIGENEN AG:

SSNR          :   0         ANR               :  2
AUFTRAGSART   : RECEIVE     AKTIV/PASSIV (A/P): P


ZUM FREMDEN AG:

BUKZ:        STKZ:        BGKZ:
ETHERNETADRESSE  : 00AA000682CE H    SSNR:   0     ANR:

MULTICAST (J/N): N   MULTICASTKREIS:      ETHERNETADRESSE:              H

DATAGRAMM (J/N): N
PRIORITAET     : 2   READ/WRITE (J/N): J


QUELLE/ZIEL    :                LAENGE :
ANZEIGENWORT   :
INTERPRETER    :                ADRESSE:      :      H


EIGENER TSAP-ID: LAENGE: 7    HEXA: 52 45 43 45 49 56 45    ASC: RECEIVE
FREMDER TSAP-ID: LAENGE: 5    HEXA: 46 4C 49 4E 4B          ASC: FLINK
FREMDER NSAP-ID: LAENGE: 12   ASC :

ANZAHL DER AUFTRAEGE PRO TSAP: 1
```

```
   VERBINDUNGSBAUSTEIN

VOM EIGENEN AG:

SSNR          :   0         ANR               :  3
AUFTRAGSART   : SEND        AKTIV/PASSIV (A/P): P


ZUM FREMDEN AG:

BUKZ:        STKZ:        BGKZ:
ETHERNETADRESSE  : 00AA000682CE H    SSNR:   0     ANR:  3

MULTICAST (J/N): N   MULTICASTKREIS:      ETHERNETADRESSE:              H

DATAGRAMM (J/N): N
PRIORITAET     : 2   READ/WRITE (J/N): J


QUELLE/ZIEL    :                LAENGE :
ANZEIGENWORT   :
INTERPRETER    :                ADRESSE:      :      H


EIGENER TSAP-ID: LAENGE: 4    HEXA: 53 45 4E 44             ASC: SEND
FREMDER TSAP-ID: LAENGE: 5    HEXA: 46 4C 49 4E 4B          ASC: FLINK
FREMDER NSAP-ID: LAENGE: 12   ASC :

ANZAHL DER AUFTRAEGE PRO TSAP: 1
```


E.4 PLC Software example

```
OB 1

NETZWERK 1          0000
            :A   DB  10              Call encode datablock
            :SPA FB   1              Communication link with FL
     NAME   :SIE-LINK
            :O   M    0.0
            :ON  M    0.0
            :A   DB  10              Call encode datablock
            :L   DR   3
            :L   KF +0
            :><F                     Encoded write performed?
            :SPA FB 244              Unsollicited receive
     NAME   :SEND
     SSNR   :      KY 0,0
     A-NR   :      KY 0,3
     ANZW   :      MW 110
     QTYP   :      KC RW
     DBNR   :      KY 0,100
     QANF   :      KF +0
     QLAE   :      KF -1
     PAFE   :      MB 114
            :SPA FB  10              Decode information from FL
     NAME   :ENCO-WRT
            :BE
```

```
DB100

     0:           KC    =     'DB';        Source type
     1:           KF    =     +00010;      Source DB number
     2:           KF    =     +00001;      Source start address
     3:           KF    =     +00005;      Source length
     4:           KC    =     'DB';        Destination type
     5:           KF    =     +00010;      Destination DB number
     6:           KF    =     +00001;      Destination start addres
     7:           KF    =     +00005;      Destination length
     8:           KF    =     +00000;
     9:           KF    =     +00000;
    10:           KF    =     +00000;
```

```
FB 1

NETZWERK 1          0000
NAME :SIE-LINK

            :O   M    0.0
            :ON  M    0.0                 VKE == 1
            :SPA FB 244                   Send all
     NAME   :SEND
     SSNR   :      KY 0,0
     A-NR   :      KY 0,0
     ANZW   :      MW 100
     QTYP   :      KC NN
     DBNR   :    KY 0,0
     QANF   :    KF +0
     QLAE   :    KF +0
     PAFE   :    MB 104
            :
            :O   M    0.0
            :ON  M    0.0                 VKE == 1
            :SPA FB 245                   Receive all
     NAME   :RECEIVE
     SSNR   :      KY 0,0
```

```
            A-NR   :      KY 0,0
            ANZW   :      MW 105
            ZTYP   :      KC NN
            DBNR   :      KY 0,0
            ZANF   :      KF +0
            ZLAE   :      KF +0
            PAFE   :      MB 109
                   :***

      NETZWERK 2           001F
                   :BE
```


Appendix F. Siemens S7-PLC

F.1 S7 H1 protocol

The Sinec H1 protocol driver is capable of communicating with a PLC from the Siemens S7 range. The S7 range of PLC's supports H1 communication with the following restriction: No read/write header information is incorporated in the transmitted data packages. The consequence are: the read functionality is no longer supported, for write and receive only one dataset (for each functionality) is useful for every device. When defining more then one for e.g. the write functionality means that all data is written to the same location in the PLC. When using this type of communication, the device type specified in the Sinec H1 protocol driver tables must be 'S7RW'!

A solution for this shortcoming is to use the S7 function blocks (FB5, FB6, FB7) which implements the H1 protocol on the PLC side. To enable the read functionality FB5 should be called once every PLC cycle, the same applies to the write functionality: call FB6 every PLC cycle. The last function block FB7 can be used for the unsolicited receive.

All the functions call internally the following functions:

SFC20, SFC64	These are standard function blocks.
FC5, FC6	These function block are part of the NCM S7 industrial ethernet package and implement the basic receive and send on the PLC side.

The functions blocks (FB5, FB6, FB7) for implementing the H1 protocol on a S7 PLC, use the function blocks FC5, FC6, which are part of the NCM S7 package. These two functions implement the read (FC5) and the write (FC6) service on a connection. There is one important limitation for both function blocks: the maximum length of the data package is 240 bytes! From these 240 bytes are 16 bytes needed to implement the H1 protocol, this result in an effective data length of **224** bytes**!**
When defining datasets for the Sinec H1 protocol driver this length of 224 bytes is the maximum length of a dataset, even if the range inside the PLC can be larger.. Trying to write or read a dataset with a length exceeding the 224 bytes will result in an error.
The supported data areas in the S7 PLC, including the maximum addresses and lengths are summarised in the table below.

Supporte Data areas Siemens S7			
Data area	Max. Db no.	Max. address	Description
DB	255	65534	Data block
FB		2047	Flag byte
IB		8191	Input byte
OB		8191	Output byte

FB5: FL_READ

This function block is used to implement the read functionality for the driver, when this function is used the PLC device type definition should be set to 'S7' or 'S5'. The choice between 'S5' and 'S7' depends on the desired addressing mode of data blocks. When using 'S5' data blocks are addressed with words, the 'normal' addressing mode for the S5 PLC range. Specifying 'S7' means that the driver confirms to the addressing mode on bytes for a data block in a S7 PLC. Note that this information also must be specified as a parameter for the function block.

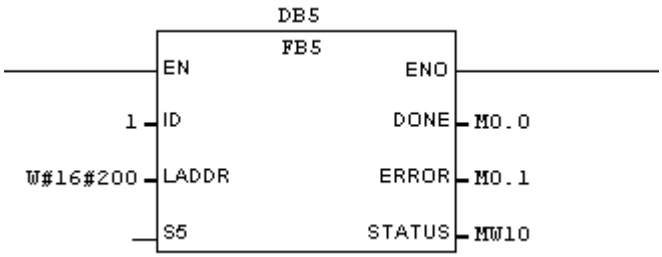


Figure G.1.1 Function block FB5.

Parameters:

Instance DB: The function block needs an instance DB, specify the desired block number.

ID: Id of the virtual connection which will be used, id can be found in the connection definition table.

LADDR: Local address of the virtual connection which will be used, address can be found in the connection definition table.

S5: Boolean to specify if S5 or S7 addressing mode is to be used. A value of TRUE means that the S5 addressing mode is used. S7 addressing mode is used when the value is FALSE.

DONE: Output flag set to TRUE for one cycle on completion of a read job, without an error.

ERROR: Output flag set to TRUE for one cycle on completion of a read job, with an error.

STATUS: Output word set to zero on completion of a read job, without an error. If a job ends with an error, the word contains an error code from the one of the following function blocks: SFC20, FC5 or FC6

FB6: FL_WRITE

This function block is used to implement the write functionality for the driver, when this function is used the PLC device type definition should be set to ‘S7’ or ‘S5’. The choice between ‘S5’ and ‘S7’ depends on the desired addressing mode of data blocks. When using ‘S5’ data blocks are addressed with words, the ‘normal’ addressing mode for the S5 PLC range. Specifying ‘S7’ means that the driver confirms to the addressing mode on bytes for a data block in a S7 PLC. Note that this information also must be specified as a parameter for the function block.

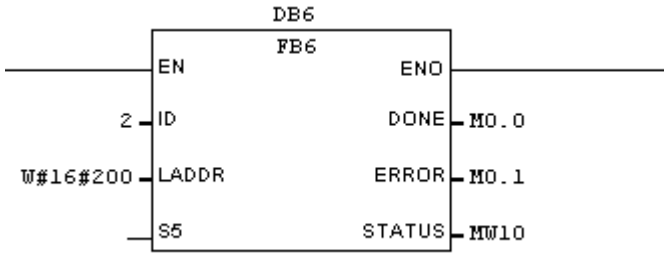


Figure G.1.2 Function block FB6.

Parameters:

Instance DB: The function block needs an instance DB, specify the desired block number.

ID: Id of the virtual connection which will be used, id can be found in the connection definition table.

LADDR: Local address of the virtual connection which will be used, address can be found in the connection definition table.

S5: Boolean to specify if S5 or S7 addressing mode is to be used. A value of TRUE means that the S5 addressing mode is used. S7 addressing mode is used when the value is FALSE.

DONE: Output flag set to TRUE for one cycle on completion of a write job, without an error.
ERROR: Output flag set to TRUE for one cycle on completion of a write job, with an error.
STATUS: Output word set to zero on completion of a write job, without an error. If a job ends with an error, the word contains an error code from the one of the following function blocks: SFC20, FC5 or FC6

FB7: FL_SEND

This function block is used to implement the unsolicited receive functionality for the driver, when this function is used the PLC device type definition should be set to ‘S7’ or ‘S5’. The choice between ‘S5’ and ‘S7’ depends on the desired addressing mode of data blocks. When using ‘S5’ data blocks are addressed with words, the ‘normal’ addressing mode for the S5 PLC range. Specifying ‘S7’ means that the driver confirms to the addressing mode on bytes for a data block in a S7 PLC. Note that this information also must be specified as a parameter for the function block. Additional parameters include a start condition and a definition of the data area which should be send to the FactoryLink workstation. If the start condition is false, no data transmission will be initiated, but the status of a pending job will be reported on the output parameters.

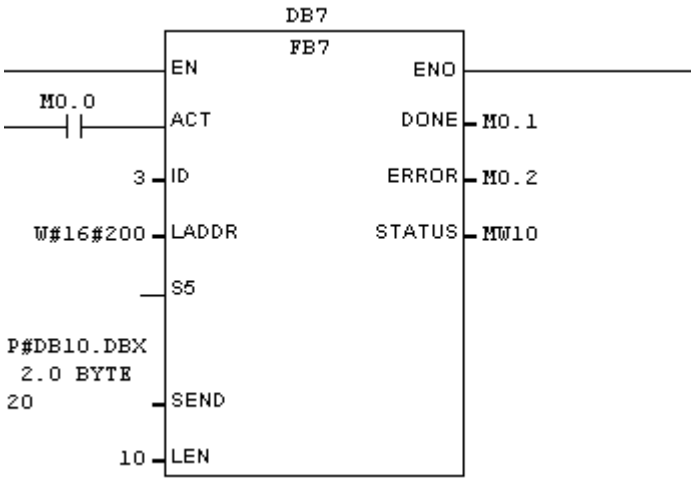


Figure G.1.2 Function block FB7.

Parameters:

Instance DB: The function block needs an instance DB, specify the desired block number.

ACT: Activate send command, in case the value is TRUE a send command is started.
ID: Id of the virtual connection which will be used, id can be found in the connection definition table.
LADDR: Local address of the virtual connection which will be used, address can be found in the connection definition table.
S5: Boolean to specify if S5 or S7 addressing mode is to be used. A value of TRUE means that the S5 addressing mode is used. S7 addressing mode is used when the value is FALSE.
SEND: Any pointer to identify the area to send to the FactoryLink workstation. Specify the start address and the maximum length of this area. The actual length to send is specified with the next parameter. Note that the maximum length must be greater or equal to the maximum length.
LEN: Actual length in bytes to send.

DONE: Output flag set to TRUE for one cycle on completion of a write job, without an error.

ERROR:	Output flag set to TRUE for one cycle on completion of a write job, with an error.
STATUS:	Output word set to zero on completion of a write job, without an error. If a job ends with an error, the word contains an error code from the one of the following function blocks: SFC20, SFC64, FC5 or FC6

F.2 Encoded write handling

The Sinec H1 protocol driver is capable to perform an encoded write. An encoded write causes the protocol driver to place coded information in a data block in the PLC instead of directly updating or changing a variable.

The information, placed in the PLC by an encoded write, should be decoded by a PLC-program in order to update/change the correct variable. A standard function block (FB10) is provided by DeltaLink for decode actions inside the PLC. The standard function block FB10 is called "ENCODED", and uses SFC20 and an instance data block. The instance data block should contain the information from an encoded write. The protocol driver places the information in a special way in the encode DB.

Encode DB		
Data byte	Type	Description
DBB0	KF	Reserved
DBB0	KF	Reserved
DBB2	KY	Destination
DBB3	KY	destination type
DBB4	KY	DB no..
DBB5	KY	address no.
DBB6	KY	Bit no.
DBB7	KY	Change flags
DBB8	KY	Value
DBB9	KY	Value
DBB10	KY	Value
DBB11	KY	Value

Figure F.2.1 Encode Data Block.

Destination: the information in this byte indicates the destination of the value which should be updated.

Value		Description
M	=	Flag type.
D	=	Data block.
A	=	Output.
E	=	Input.

Destination type: the information in this byte indicates the destination type of the value which should be updated.

Value		Description
F	=	Flag for DB, F, A or E.
B	=	Byte for DB, F, A or E.
W	=	Word for DB, F, A or E.
D	=	Double word, for long integer or float.

DB number: The information in this byte is the number of a DB. If the destination is not D the value in this byte is irrelevant. The value is between 0 and 255.

Address number: address number of the byte, word, data word, timer or counter. The value is between 0 and 255.

Bit number: the number of the bit in a word or a flag. If the destination type is not flag, the value is irrelevant.

Change flags: For every encoded write the protocol driver writes the value 255 in this byte. Function block FB10 uses this value to detect if there is new information.

Value: value may be up to longs, depending on the type of data. For flag and byte values DBB8 is used. For word values DBW8 is used. And for long integers and float DBW8 plus DBW10 is used.

FB10 ENCODED

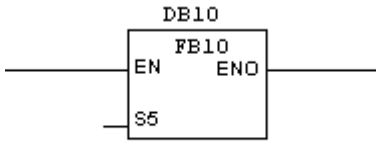


Figure F.2.2 Function Block FB10.

Parameters:

Instance DB: The function block needs an instance DB, specify the desired block number.

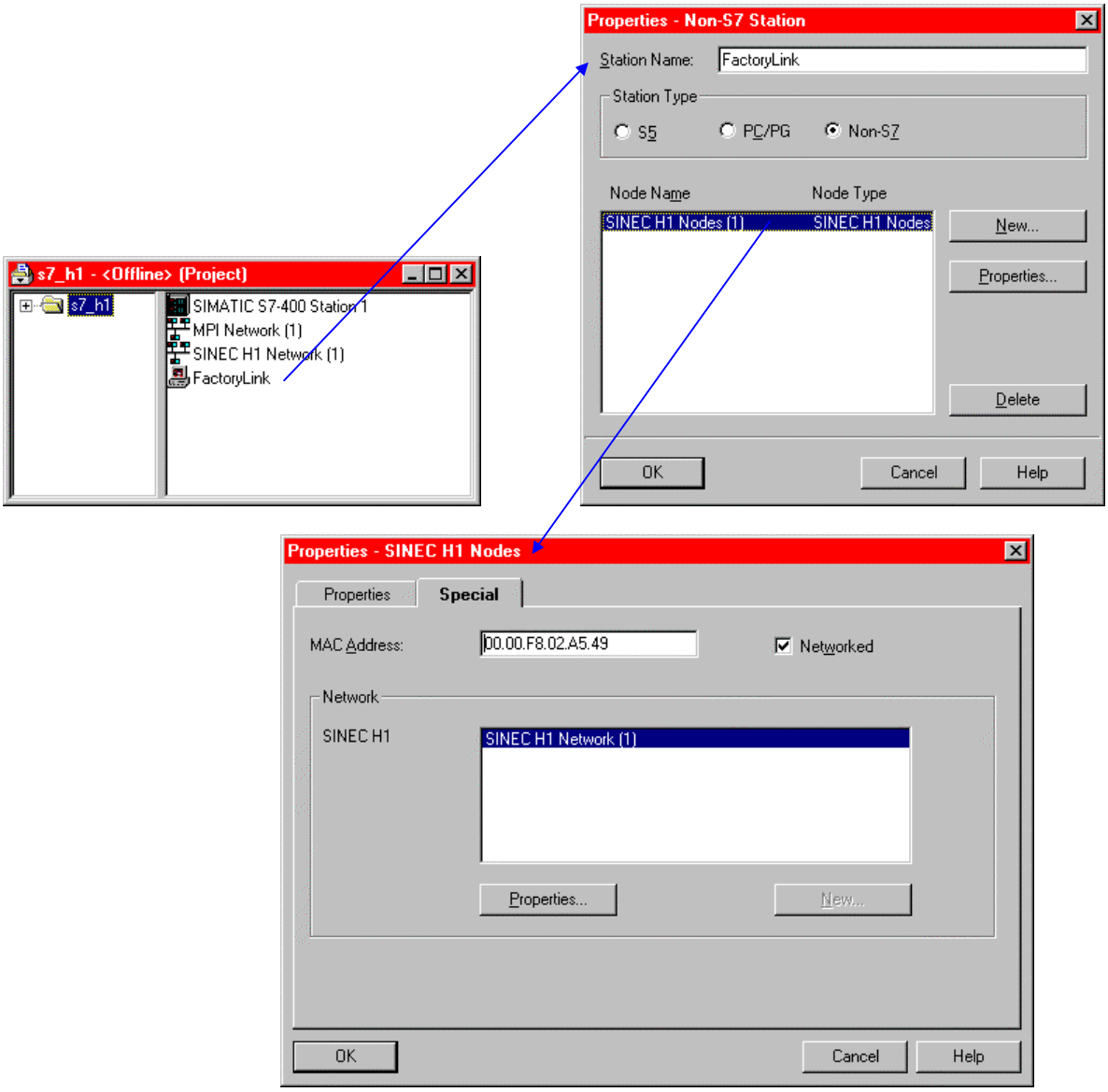
S5: Boolean to specify if S5 or S7 addressing mode is to be used. A value of TRUE means that the S5 addressing mode is used. S7 addressing mode is used when the value is FALSE.

F.3 CP443 Configuration

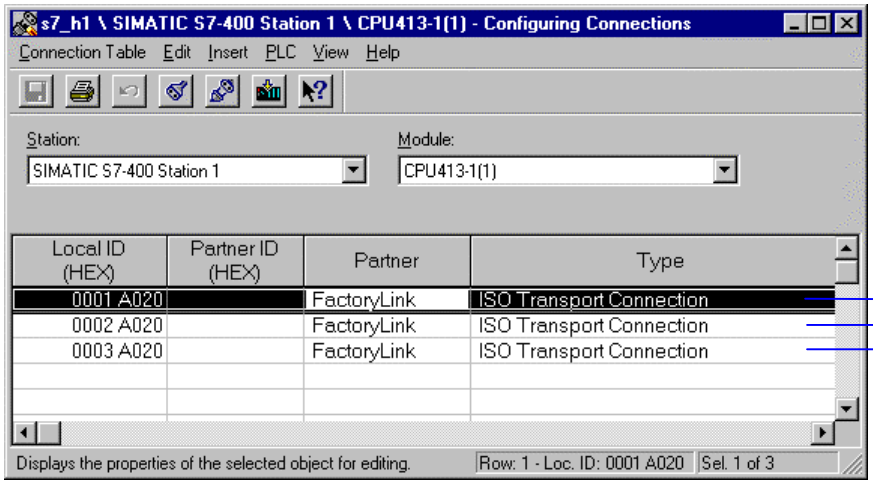
This section displays the configuration of a CP443 card. The following should be defined as a minimum on the S7 (assuming the S7 system configuration is correct):

1. Remote station, this includes the MAC address of the FactoryLink workstation.
2. A minimum of three connections must be defined on the CP 443 card. A connection together with one of the function blocks FB5, FB6 or FB7 gives the functionality the driver needs: that is read, write or unsolicited receive!

Remote station definition

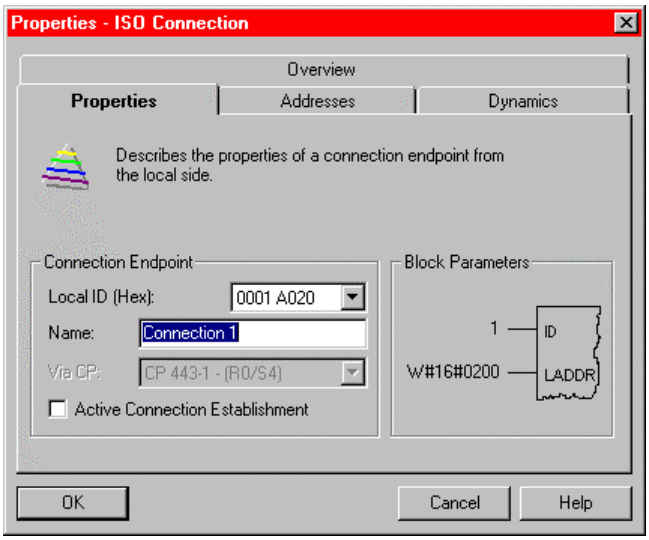


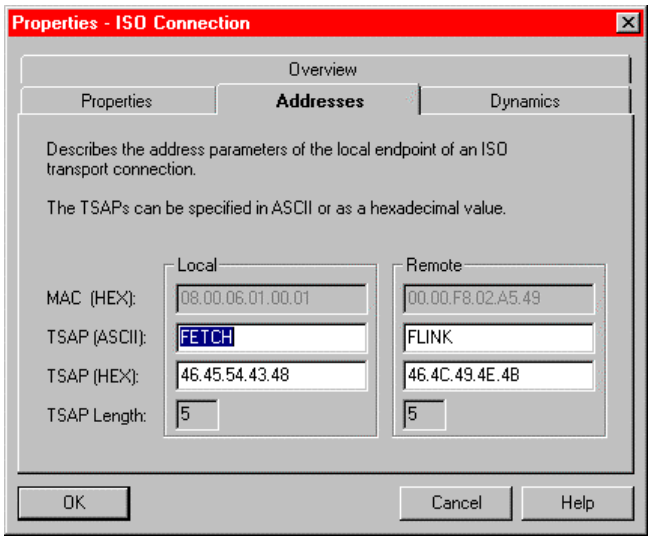
Connection (TSAP) definition

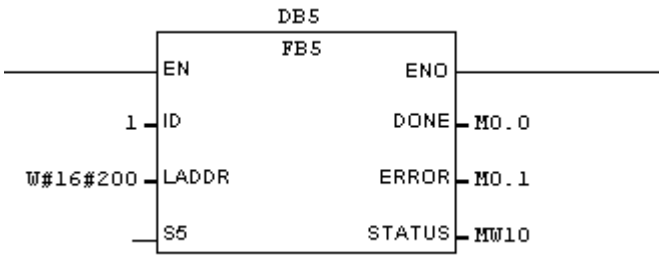


Sinec H1 driver functionality	Function block
Read	FB5
Write	FB6
Unsolicited Receive	FB7

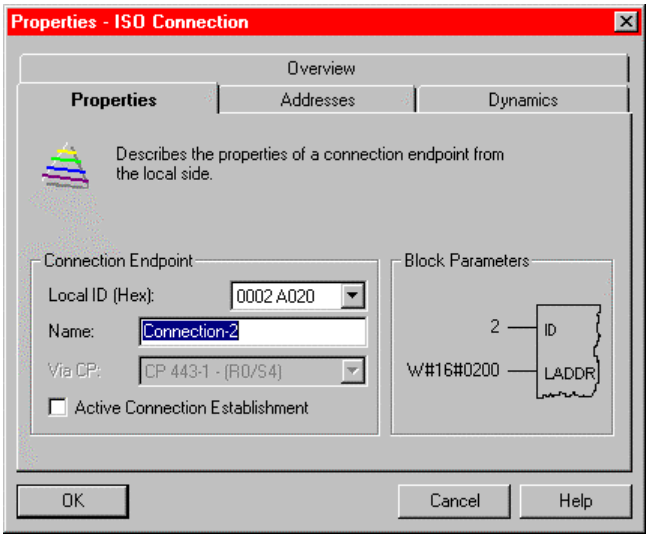
Read connection

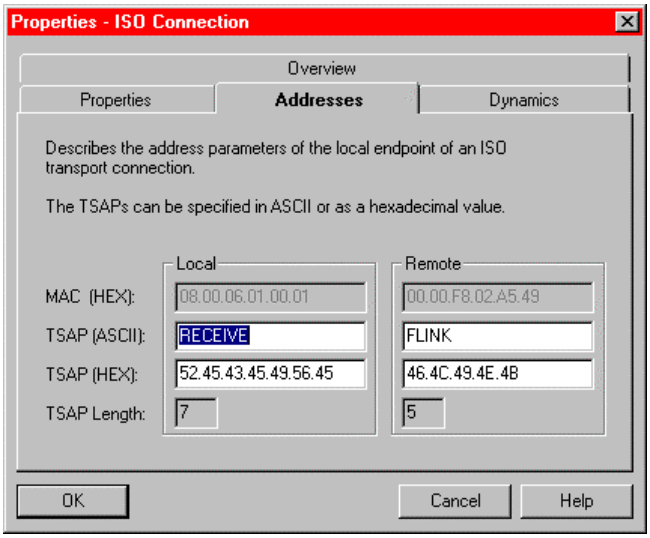


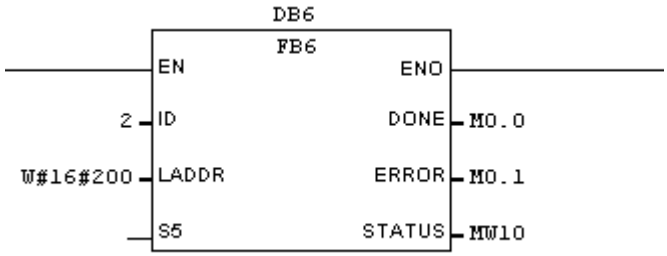




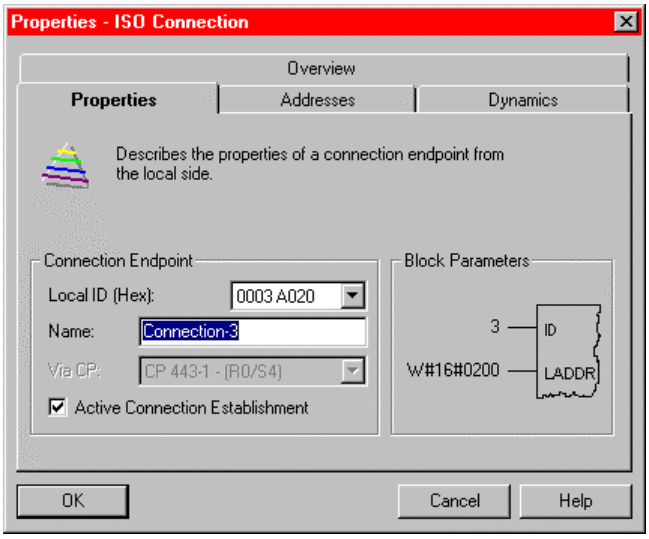
Write connection

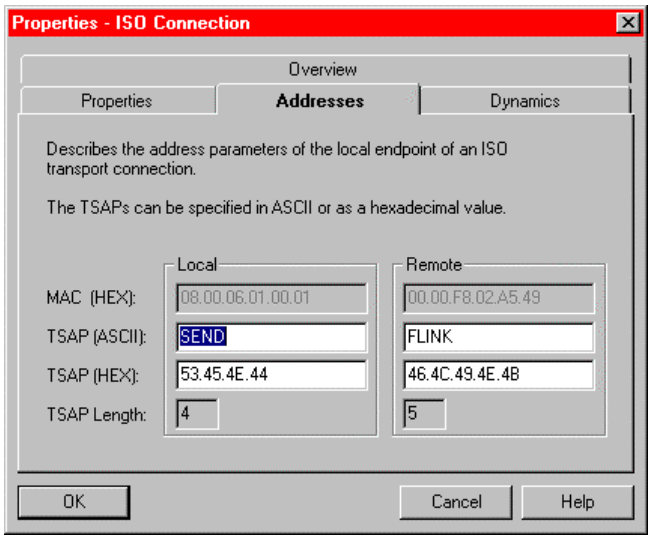


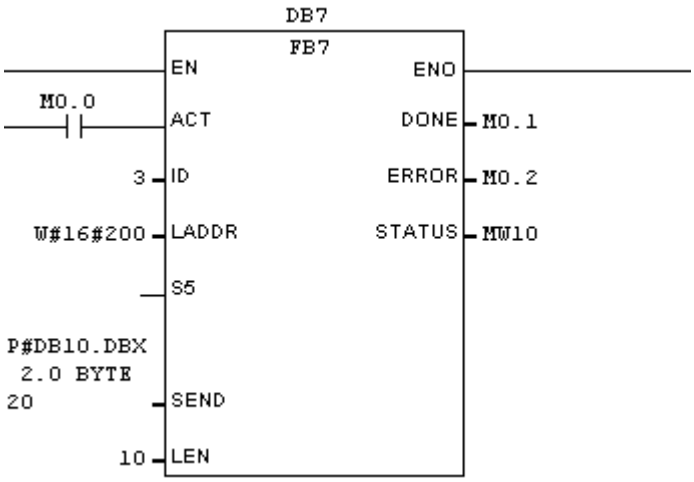




Unsolicited receive connection







F.4 PLC Software example

```
ORGANIZATION_BLOCK OB 1
TITLE =S7 H1 demo
VERSION : 0.1


VAR_TEMP
  OB1_EV_CLASS : BYTE ;           //Bits 0-3 = 1 (Coming event), Bits 4-7 = 1
(Event
                                  //class 1)
  OB1_SCAN_1 : BYTE ;             //1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n
of
                                  // OB 1)
  OB1_PRIORITY : BYTE ;           //1 (Priority of 1 is lowest)
  OB1_OB_NUMBR : BYTE ;           //1 (Organization block 1, OB1)
  OB1_RESERVED_1 : BYTE ;         //Reserved for system
  OB1_RESERVED_2 : BYTE ;         //Reserved for system
  OB1_PREV_CYCLE : INT ;          //Cycle time of previous OB1 scan (milliseconds)
  OB1_MIN_CYCLE : INT ;           //Minimum cycle time of OB1 (milliseconds)
  OB1_MAX_CYCLE : INT ;           //Maximum cycle time of OB1 (milliseconds)
  OB1_DATE_TIME : DATE_AND_TIME ; //Date and time OB1 started
END_VAR
BEGIN
NETWORK
TITLE =Handling of basic communication functions

      CALL FC     1 ;

NETWORK
TITLE =Timer/Counter for H1 demo

      OPN   DB    50;
      L     T      1;
      T     DBW    2;
      T     MW     2;
      L     C      1;
      T     DBW    4;
      T     MW     4;
      AN    T      1;
      L     W#16#999;
      SD    T      1;
      A     M    100.2;
      CU    C      1;
      L     C      1;
      L     999;
      >=I   ;
      R     C      1;
END_ORGANIZATION_BLOCK
```

```
FUNCTION FC 1: VOID
TITLE =S7 H1
//The three functionality's for H1 communication with FactoryLink are handled
//here:
//1. Read
//2. Write
//3. Unsolicited receive
NAME : H1
VERSION : 0.1

BEGIN
NETWORK
TITLE =FactoryLink Read
//1. FactoryLink read: addressing information is evaluated and the requested data
//   is send to the FactoryLink work station.
      CALL FB     5 , DB     5 (    // FactoryLink read, send requested data
           ID                       := 1,
```



```
           LADDR                    := W#16#200,
           S5                       := FALSE,// S7 addressing mode
           DONE                     := M    100.2,
           ERROR                    := M    100.3,
           STATUS                   := MW   110);

NETWORK
TITLE =FactoryLink Write
//2. FactoryLink write: data including addressing info is received, addressing
//   info is evaluated, and data is set on the correct location.
      CALL FB     6 , DB     6 (    // FactoryLink write, move received data
           ID                       := 2,
           LADDR                    := W#16#200,
           S5                       := FALSE, // S7 addressing mode
           DONE                     := M    100.4,
           ERROR                    := M    100.5,
           STATUS                   := MW   112);

NETWORK
TITLE =FactoryLink unsolicited Receive
//3. FactoryLink unsolicited receive: whenever there is received an encoded write,
//   the addressing information of this write is send back to the FactoryLink
//   workstation.
      L     DB10.DBB    7;          // Check if encoded write is received
      L     0;
      <>I   ;
      =     M    100.7;             // Encoded write received, send info back..
      CALL FB     7 , DB     7 (    // Send encoded write info to FactoryLink
           ACT                      := M    100.7,
           ID                       := 3,
           LADDR                    := W#16#200,
           S5                       := FALSE, // S7 addressing mode
           SEND                     := P#DB10.DBX 2.0 BYTE 10,// Origin
           LEN                      := 10, // Length of data in bytes
           DONE                     := M    101.0,
           ERROR                    := M    101.1,
           STATUS                   := MW   114);

      CALL FB    10 , DB    10 (    // Evaluation of encoded write
           S5                       := FALSE); // S7 addressing mode

END_FUNCTION
```

This page is left blank intentionally.

