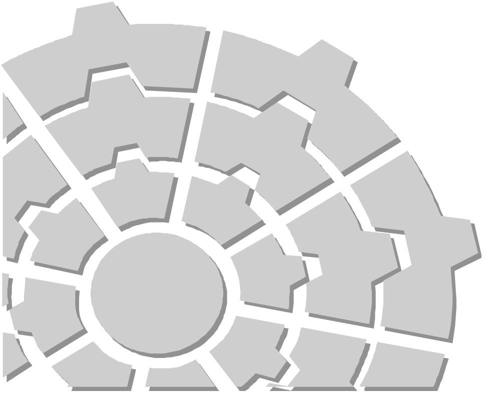
DeltaLink MB_PLUS
Modbus Plus interface software for FactoryLink

User Manual



Version 2

Printed: 6 aug, 1997

Table of Contents

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 AEG-Modicon 984 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 AEG-Modicon PLC and the basic 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 basic 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 + 31 073-5231234
fax +31 073-5231222

This page is left intentionally blank

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:

1 diskette labelled "DeltaLink MB_PLUS driver version 2.xx"

Files:	\AC\MB_PLUS.AC	
	\BIN\MB_PLUS.EXE	
	\BIN\MBPLUS.DLL	(Windows)
	\BIN\VPXD.386	(Windows)
	\BIN\SA85.SYS	(Windows)
	\LIB\SA85.DLL	(OS/2)
	\BIN\MPHOST.SYS	(OS/2)
	\BIN\MPHTICK.EXE	(OS/2)
	\BIN\DOWOS2.SYS	(OS/2)
	\CTGEN\MB_PLUS.CTG	
	\KEY\MPTYP.KEY	
	\MSG\MB_PLUS.HLP	
	\MSG\MB_PLUS.TXT	
	\OPT\DELTA.OPT	
	\MPS\MB_PLUS.MPS	
	\INSTALL	(Unix)
	\INSTALL.BAT	(Windows)
	\INSTALL.CMD	(OS/2)
	\INST_SEQ.EXE	
	\FL_BUILD.ID	
	\FLXMEDIA	(Windows-OS\2)
	\MB_PLUS.$$$	

❷ 1 DeltaLink hardware key containing the MB_PLUS option.

❸ This manual (Which seems to be present).

You should also have:

☑ A MCI[1] based decoder e.g. DeltaLink Basic Decoder

☑ AEG-Modicon 984 PLC with Modbus Plus connection with software to program the CPU.

☑ A Modbus Plus network with at least both the PLC and the FactoryLink station connected to it.

[1] Mailbox Communication Interface (MCI): A specific DeltaLink development, for inter-task communication in FactoryLink.

This page is left intentionally blank

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 Modbus Plus 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 MB_PLUS driver" diskette to the appropriate directory. This will be automatically done by running the install utility placed on the installation diskette.. To invoke this utility please enter from the prompt:

```
A: ↵
INSTALL↵
```

Second:

After you installed the software you need to activate the tables in the FactoryLink Configuration Manager (FLCM). The installation automatically appends the *mb_plus.ac* entry into the *{FLINK}/AC/titles* file[2]. 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 *mb_plus.ac* entry in the following table:

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

...
spool.ac
decoder.ac
mb_plus.ac
sinec_h1.ac
itimer.ac
etimer.a
...
```

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

Third:

To make sure all the Configuration Tables (CT's) are generated after a change, the install utility automatically adds the *mb_plus* 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:

```
file: {FLINK}/ctgen/ctlist:
...
rp: rptovr rpthdr
mb_plus: mb_plusm mb_plusx mb_plusp mb_plusd
decoder: decoderm decoderp decoderd decodert
timer: itimer etimer
...
```

Fourth:

To enable the help fields in the MB_PLUS tables type the following command:

```
mkhelp↵
```

Fifth:

The Modbus Plus task must be entered in the FactoryLink Configuration Manager (FLCM) 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 made with the following data:

Task Name	*Description*	*....*	*Executable File*
MB_PLUS	***DeltaLink Modbus Plus Protocol driver***		***bin/mb_plus***

The *Task Name* and *Executable File* name are fixed and should not be altered by the user.

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

3.2. Installation of the SA85 communication stack

It is necessary to have a SA85 Modbus Plus stack loaded which can be used by the driver to communicate with the AEG-Modicon Modbus Plus network. 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 Modbus Plus stack can be found in *Appendix A*.

3.3. Installation of the protection

The Modbus Plus driver has been protected on two different ways. The first method is via the DeltaLink option file which must reside in the *{FLOPT}*[3] directory. This file contains authorisation sequence codes for DeltaLink modules. This protection is linked to the serial number of the FactoryLink package. The second method is via a hardware key which enables DeltaLink modules to run. The end-user can choose which method of protection to use.

3.3.1. The DeltaLink option file

The installation disk of the Modbus Plus driver contains a *delta.opt* file in the *opt* directory. In case for protection with an option file has been chosen then this file contains the unique authorisation sequence which enables the Modbus Plus driver to run. The install utility automatically copies the authorisation sequence to the *{FLOPT}\delta.opt* file. In case the serial number was not known at the time of delivery the *delta.opt* file on the installation diskette will be empty. The authorisation code has to be requested at DeltaLink and manually added to the *{FLOPT}\delta.opt* file. For more information on the format of the *delta.opt* file refer to *Appendix B*.

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

3.3.2. The hardware key

The protection key is a DeltaLink key that is plugged into any parallel printer port of the computer. This key contains authorisation codes for different DeltaLink modules, purchased by the customer. These authorisation codes are not related to the FactoryLink system serial number. This way this key can be used on any FactoryLink system.

The DeltaLink keys are normally used in combination with a Usdata FactoryLink protection key. Do not unnecessary connect a printer to the keys. Use separate ports for printers when ever possible. Platform specific information for the protection key can be found in *Appendix C*.

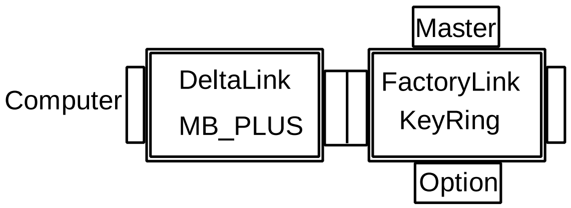


Figure 3.1.1: Protection key.

[3] For installation of authorization codes the {FLOPT} environment variable must be set

3.3.3. Demo installation.

It is possible to install the Modbus Plus driver without an authorisation code or protection key. This will be done from a Modbus Plus demo diskette. A Modbus Plus demo diskette contains the complete installation of the driver except for an authorisation code in the *delta.opt* file. In this case the driver will start up but only run in demo mode. This means that the Modbus Plus driver runs on limited resources[4] such as number of devices and a period of time (usual one hour). After this period has expired the driver will shutdown and can not be restarted before the total FactoryLink system has been restarted.

After installation of this demo diskette an authorisation code or protection key can be ordered and installed which enables the driver to run with full functionality. The hardware key needs only to be connected to the parallel port. The authorisation code must be entered manually in the *{FLOPT}\delta.opt* file. For more information on entering the authorisation code in the delta.opt file refer to *Appendix B*.

The limitations of a demo version of the Modbus Plus driver are:

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

3.3.4. Lite installation.

The DeltaLink Modbus Plus driver can also be purchased in a lite version. The installation package of this lite version is exactly the same as the full version. The protection mechanism enables the lite or the full version to run (authorisation code or hardware key). The lite version differs from the full version in its resources which are limited. The following resource limitation apply to the Modbus Plus Lite version:

- maximum of one decoder supported
- maximum two PLC's supported
- maximum of 16 datasets supported

All other functionality's are exactly the same as the full version of the Modbus Plus driver.

[4] If an application is started with more resources then this will result in errors and not correct functioning of the application

4. Principle

4.1. Protocol-driver / decoder principle

DeltaLink designed its own method for implementing high-performance communication drivers. In principal it consists of two parts, each part being a specific FactoryLink task. The first task is the protocol driver which actually communicates with the PLC(s). It knows how to 'speak' with a PLC and can get data to and from the PLC. The data is always a contiguous block of words (or bytes). The second task is the decoder which is identical for all communication protocol drivers. The decoder can be considered as an unique user-interface for configuring communications inside FactoryLink. Raw data from the PLC are for example converted before writing the values to Real Time DataBase (RTDB) elements (TAGs). On the other hand TAG values are converted in a way that they can be handled by the protocol driver, for actual writing to the PLC.

The protocol driver and the decoder communicate through a FactoryLink TAG of type mailbox using the MCI[5] protocol. The here explained principle is especially designed for the following situation: to use one decoder and several protocol drivers. For example the basic decoder together with Modbus Plus protocol driver and the Sinec H1 protocol driver.

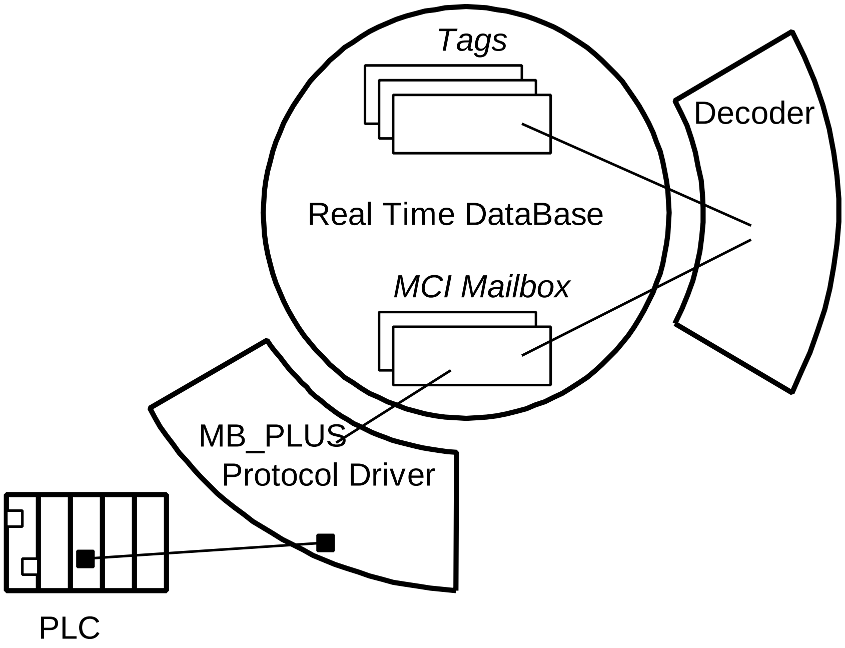


Figure 4.1.1 The Protocol-driver/Decoder Principle.

4.2. 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. The size of an basic-element of a dataset depends on the type of data that remains in the PLC.

[5] Mailbox Communication Interface (MCI): A specific DeltaLink development, for inter-task communication in FactoryLink.

4.3. Modbus Plus Communication

The main and only task of the DeltaLink Modbus Plus protocol driver is to communicate with AEG-Modicon 984 PLC's. The Modbus Plus driver can perform a range of different functionality's which are all performed asynchronous and independent of each other with different PLC's at the same time. Some characteristics and the functionality's of the Modbus Plus driver will be discussed in the next section.

4.3.1. Connections

Before exchanging data with a remote PLC a path connection must be established. The Modbus Plus protocol supports four different types of connections:

Data Master (DM) Path

This type of path is opened for data reads and writes, and for get and clear remote statistics, as they are originated in the device.

Data Slave (DS) Path

This type of path is opened for data reads and writes as they are received by the device.

Program Master (PM) Path

This type of path is opened for programming commands as they are originated in the device.

Program Slave (PS) Path

This type of path is opened for programming commands as they are received by the device.

The Modbus Plus driver uses the *DM* path for executing read and write commands and the *DS* path for the unsolicited receive function. The quantity of the paths is limited on the SA85 adapter. Each type of path has a maximum of eight. This means that a maximum of eight PLC's per SA85 adapter with full functionality can be connected. It is possible to install two SA85 adapters in the same system which doubles the number of total paths.

For information on addressing on the Modbus Plus network through path's refer to the Modicon Modbus Plus documentation (for example Modicon IBM Host Based Devices User's Guide).

4.3.2. Read and Block Write

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

4.3.3. 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 in 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 slow and unpredictable 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.

Unsolicited data is transferred via a data slave (DS) path. In the Modbus Plus panels a DS path can be assigned to a remote PLC. It is important to know that this path can only be used with the specific PLC. This means that the PLC must transmit over the correct DS path. In case the PLC transmits data over another DS path the Modbus Plus driver will not accept the data for the specific PLC. In the worst case it will accept the data as if it were coming from the logical PLC with the specified DS path.

4.3.4. 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 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 the data must be read first, patched with exception data and written back. If this is not done then data will be unintentionally overwritten in the PLC. When this is the case depends on the data type of the dataset.

For example: if a byte left (BL) in a Holding Register (HREG) has to be set then the specific data word has to be read first because the size of a Holding Register is a word. The BL has to be patched into the word and written back.

The reading of data before writing is not performance friendly and can be solved by using the encoded write which will be discussed in the next section.

4.3.5. 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 range of Holding Registers. 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 Modbus Plus 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 off the lay-out of an encoded write command refer to appendix F.

The location where the encoded write command will be placed must be specified in the Encode Address field of the *DeltaLink Modbus Plus Device definition* panel.

4.3.6. Global read and write

The Modbus Plus protocol supports the global data type. Global data is a limited package of maximum 32 words which can be can be sent by a node on the network together with the token on the network. The data from that node can be received by every other node on the network. Because the data travels together with the token it is a very fast and determined means of communication.

The DeltaLink Modbus Plus driver supports the reading and writing of global data. Just specify a dataset in the protocol driver of the type of global data[6] (GLOB). Every time the decoder writes data to this dataset a global write will be performed on the network.

The reading of global data can be performed on two different ways: as a normal read or as an unsolicited receive. In the first case the decoder initiates a normal read on the global dataset. The protocol driver will perform the normal read and forward the data to the decoder. In the second case the **-g** option must be specified in the program arguments of the protocol driver and the dataset must be marked as unsolicited receive in the decoder. The protocol driver will now continuously read the global data and check it for changes. In case the global data has changed then the dataset will be forwarded to the decoder. From point of view of the decoder the global read will act as an unsolicited receive.

The difference between reading or unsolicited receiving global data lies in performance versus load of the system. In the case of a read the performance of getting global data won’t be as optimal as with the unsolicited receive but on the other hand the load will be less. With the unsolicited receive the global data will be received as fast as possible but the load will be more because the protocol driver is constantly receiving and comparing global data.

[6] The addressing within global data always starts with the addess 1.

4.4. FactoryLink domain selection

The standard domain for the Modbus Plus 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 started by the shared runtime-manager.

Remark: The protocol driver and the basic decoder must be in the same domain (either SHARED or USED).

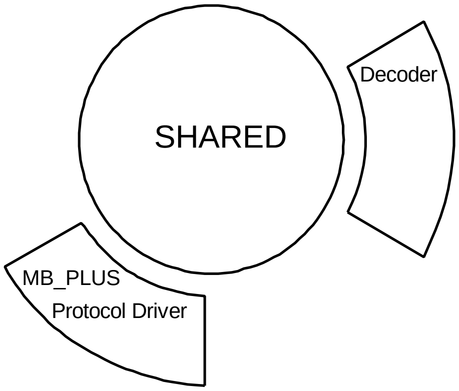


Figure 4.4.1 Standard domain selection.

This page is left intentionally blank

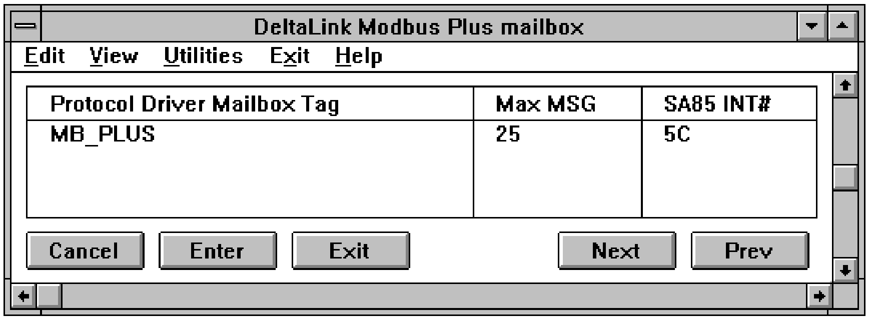
5. Configuration tables

In the Configuration Manager Main Menu, select **DeltaLink Modbus Plus 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 Chapter 5 of the FactoryLink Software System User Manual.

5.1. Modbus Plus mailbox

From the display of all the panels, select the *DeltaLink Modbus Plus mailbox* panel.



Protocol Driver Mailbox Tag	Max MSG	SA85 INT#
MB_PLUS	25	5C

Figure 5.1.1 DeltaLink Modbus Plus mailbox panel.

The DeltaLink Modbus Plus mailbox panel allows the user to initialise one Mailbox Tag for the Modbus Plus protocol driver. Only one mailbox is needed for full operation of MCI. Specify the following information.

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

  valid entry: Tag name of the decoder mailbox element of type 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 50 messages. There is not really a maximum for the number of messages. It is limited by the size of an integer value, and practically by the amount of available memory. The memory needed for a request depends on the size of the configured datasets.

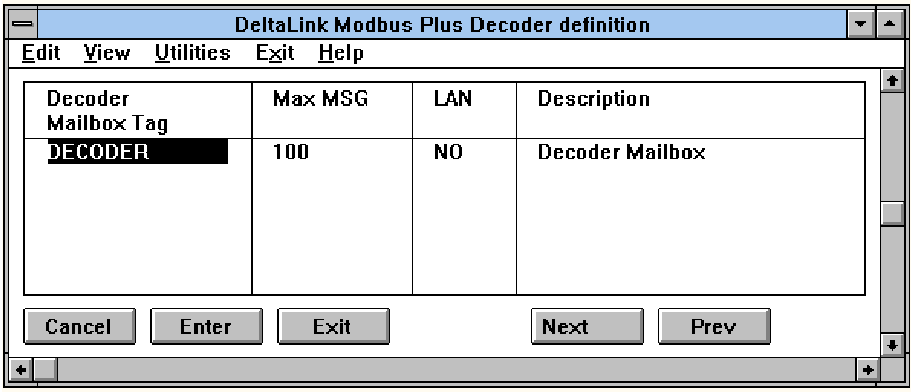
  valid entries: Minimum value is 1.

- **SA85 INT#**
  Software interrupt number to use with the SA85.SYS device driver. This value must correspond to the value specified with the /S option with the SA85.SYS driver in the config.sys file. The default interrupt used id 5Ch.

  valid entries: Hexadecimal number 00h to FFh.

5.2. Decoder definition

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


DeltaLink Modbus Plus Decoder definition
Edit View Utilities Exit Help
Cancel
Enter
Exit
Next
Prev

Decoder Mailbox Tag	Max MSG	LAN	Description
DECODER	100	NO	Decoder Mailbox

Figure 5.2.1 DeltaLink Modbus Plus Decoder definition.

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

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

valid entry: Required.
description: Standard FactoryLink tag name of type 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.

valid entry: Required/default: 100.
valid entries: 1 .. 9999.

♦ **LAN**
The protocol driver and decoder can reside on separate FactoryLink workstations which communicate through the LAN option of FactoryLink. The developer defines if LAN communication is to be used for the decoder - protocol driver combination.

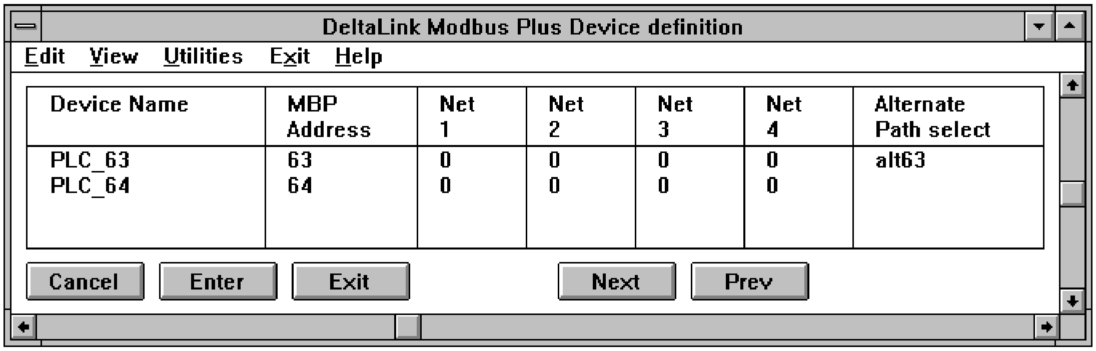
valid entry: Field is in development.
valid entries: YES or NO.

¨ **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 Modbus Plus Device definition* panel.



Device Name	MBP Address	Net 1	Net 2	Net 3	Net 4	Alternate Path select
PLC_63	63	0	0	0	0	alt63
PLC_64	64	0	0	0	0	

Figure 5.3.1 DeltaLink Modbus Plus Device definition.

The DeltaLink Modbus Plus Device definition panel allows the user to specify the logical devices for the communication. Specify the following information.

♦ **Device Name**
Logical name assigned by the user to represent a particular communication device.

valid entries: Alphanumeric string of up to 16 characters.

♦ **MBP Address**
The Modbus Plus Local Area Network node address. Up to 64 nodes can be present on a single network, each with a unique address between 1 and 64.

valid entries: decimal number, 1 to 64

¨ **Net 1**
The Net 1 field specifies the routing address 1. For more information on Modbus Plus Message Routing information refer to the Modicon Modbus Plus documentation.

valid entries: decimal number, 0 to 64

¨ **Net 2**
The Net 2 field specifies the routing address 2. For more information on Modbus Plus Message Routing information refer to the Modicon Modbus Plus documentation.

valid entries: decimal number, 0 to 64

¨ **Net 3**
The Net 3 field specifies the routing address 3. For more information on Modbus Plus Message Routing information refer to the Modicon Modbus Plus documentation.

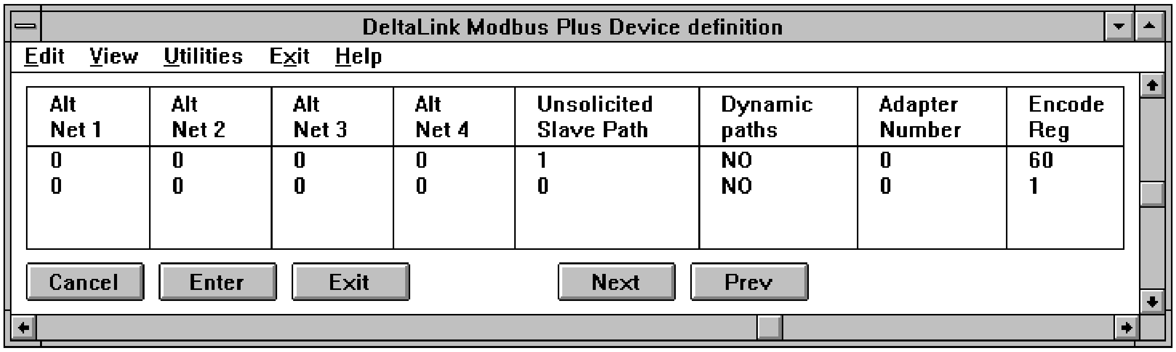
valid entries: decimal number, 0 to 64

- **Net 4**
  The Net 4 field specifies the routing address 4. For more information on Modbus Plus Message Routing information refer to the Modicon Modbus Plus documentation.

  valid entries: decimal number, 0 to 64

- **Alternate path select**
  The alternate path select tag must be used to switch between the normal routing path and the alternative routing path. If the tag holds the value 0 the normal routing path will be used. If the tag holds the value 1 the alternative routing path will be used.

  valid entries: Standard FactoryLink tag name of type DIGITAL.



Alt Net 1	Alt Net 2	Alt Net 3	Alt Net 4	Unsolicited Slave Path	Dynamic paths	Adapter Number	Encode Reg
0	0	0	0	1	NO	0	60
0	0	0	0	0	NO	0	1

Figure 5.3.2 DeltaLink Modbus Plus Device definition.

- **Alt Net 1**
  The Alt Net 1 field specifies the alternative routing address 1. For more information on Modbus Plus Message Routing information refer to the Modicon Modbus Plus documentation.

  valid entries: decimal number, 0 to 64

- **Alt Net 2**
  The Alt Net 2 field specifies the alternative routing address 2. For more information on Modbus Plus Message Routing information refer to the Modicon Modbus Plus documentation.

  valid entries: decimal number, 0 to 64

- **Alt Net 3**
  The Alt Net 3 field specifies the alternative routing address 3. For more information on Modbus Plus Message Routing information refer to the Modicon Modbus Plus documentation.

  valid entries: decimal number, 0 to 64

- **Alt Net 4**
  The Alt Net 4 field specifies the alternative routing address 4. For more information on Modbus Plus Message Routing information refer to the Modicon Modbus Plus documentation.

  valid entries: decimal number, 0 to 64

- **Unsolicited Slave Path**
  The unsolicited slave path must be used to select a communication channel for the unsolicited receive function. There are 8 channels available per SA85 adapter in the of 1 to 8. In case 0 is specified then the unsolicited receive function for the specific device is not supported.

  valid entries: decimal number, 0 to 8

- **Dynamic paths**
  The dynamic paths field specifies if to maintain permanent (option No) or dynamic (option Yes) connection. In the case of permanent connections the paths are opened at start up for the specific device. With dynamic connections no connection is established until a request for communication is received. Then a path will be opened and after the request is handled the path will be closed. This option doesn't apply to the unsolicited receive functionality.

  valid entries: Yes or No

- **Adapter Number**
  The adapter number field specifies the SA85 adapter through which the communication of the device will flow. A maximum of 2 SA85 adapters in one system is possible at the moment.

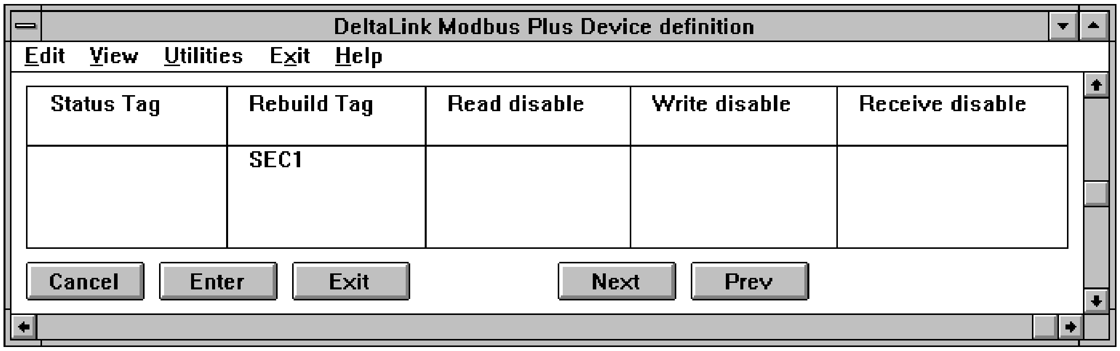
  valid entries: decimal number, 0 - 1.

- **Encode Reg**
  Holding Register (HREG) address for encoded write commands. Coded information is placed in this Holding Register area, encoding is performed by the PLC. A standard PLC program is provided by DeltaLink for decode actions inside the PLC, for specific information see also *Appendix E*.

  valid entries: decimal number, 0 - 255.

- **Status Tag**
  Real-time database element set by the Modbus Plus 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. For a detailed list of communication errors see *Appendix C*.

  valid entry: Standard FactoryLink tag name of type ANALOG.



Status Tag	Rebuild Tag	Read disable	Write disable	Receive disable
	SEC1			

Figure 5.3.3 DeltaLink Modbus Plus Device definition.

- **Rebuild Tag**
  Real-time database element set by the user to trigger a rebuild of the communication with a logical device. The rebuild is performed in case the database element has the value ON and the change flags are ON. Used in case there should have been error on the virtual communication line.

  valid entry: Standard FactoryLink tag name of type DIGITAL.

- **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 the status of the tag is ON.

  valid entry: Standard FactoryLink tag name of type 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 the status of the tag is ON.

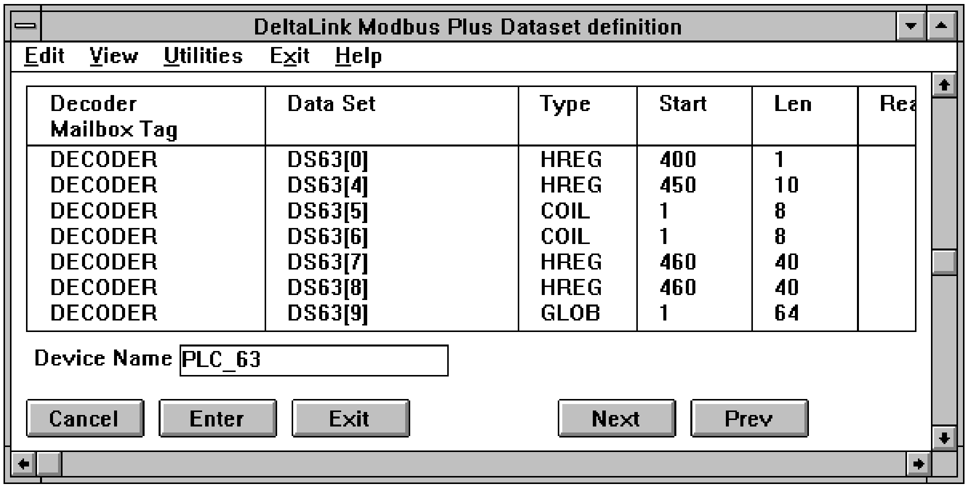
  valid entry: Standard FactoryLink tag name of type DIGITAL.

- **Receive disable**
  Real-time digital database element used to enable/disable unsolicited 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 the status of the tag is ON.

  valid entry: Standard FactoryLink tag name of type DIGITAL.

5.4. Dataset definition

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


DeltaLink Modbus Plus Dataset definition
Edit View Utilities Exit Help
Cancel
Enter
Exit
Next
Prev

Decoder Mailbox Tag	Data Set	Type	Start	Len	Rea
DECODER	DS63[0]	HREG	400	1	
DECODER	DS63[4]	HREG	450	10	
DECODER	DS63[5]	COIL	1	8	
DECODER	DS63[6]	COIL	1	8	
DECODER	DS63[7]	HREG	460	40	
DECODER	DS63[8]	HREG	460	40	
DECODER	DS63[9]	GLOB	1	64	

Device Name PLC_63

Figure 5.4.1 DeltaLink Modbus Plus Dataset definition.

The DeltaLink Modbus Plus Device definition panel allows the user to specify datasets for a particular logical device. Different logical device names are selected in the panel *DeltaLink Modbus Plus Device definition*. Specify the following information.

- **Data Set**
  Tag name representing a (unique) logical name for a data set. The name of the logical device is displayed in the field *Device Name*.

  valid entry: Standard FactoryLink tag name of type DIGITAL.

- **Type**
  Type definition of the data block in the logical device (PLC). The data blocks can be of type Coil, Digital Input, Analog I/O registers, Holding Registers and Global Data.

  valid entries:

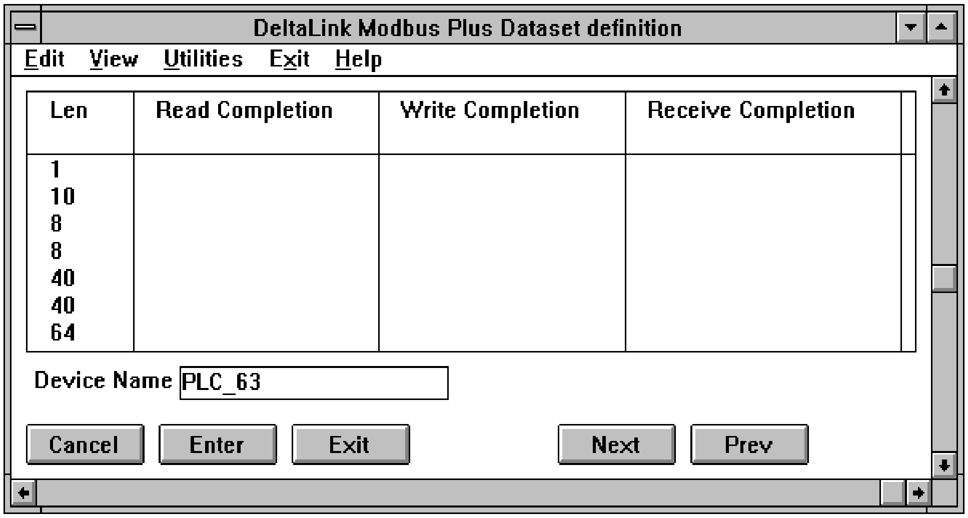
Type	Description
C	Digital output (0x range, also COIL)
D	Digital input (1x range, also DINP)
I	Analog I/O registers (3x range, also IREG)
H	Holding registers (4x range, HREG)
G	Global data (also GLOB)

- **Start**
  Start address number of the dataset.

  valid entries: 0 - 9999.

- **Len**
  Length of the dataset in bytes or words, depending on the type of the dataset. The length will always be specified in the boundary size of a data type. For example the HREG type has a word boundary so the length will be specified in words. A maximum of 125 words can be transferred by the Modbus Plus protocol. This means that the maximum size of a dataset is 125 words.

  valid entries: 1 - 9999, maximum length depends on type of data set.



Len	Read Completion	Write Completion	Receive Completion
1			
10			
8			
8			
40			
40			
64			

Device Name PLC_63

Figure 5.4.1 DeltaLink Modbus Plus Dataset definition.

- **Read completion**
  Real-time database element used to indicate that a read command is complete. This element, if entered, is forced ON by the Modbus Plus protocol driver task and can be referenced by any other FactoryLink task.

  valid entry: Standard FactoryLink tag name of type DIGITAL.

- **Write completion**
  Real-time database element used to indicate that a write command is complete. This element, if entered, is forced ON by the Modbus Plus protocol driver task and can be referenced by any other FactoryLink task.

  valid entry: Standard FactoryLink tag name of type DIGITAL.

- **Receive completion**
  Real-time database element used to indicate that an unsolicited receive command is complete. This element, if entered, is forced ON by the Modbus Plus protocol driver task and can be referenced by any other FactoryLink task.

  valid entry: Standard FactoryLink tag name of type DIGITAL.

5.5. Modbus Plus program arguments

The Modbus Plus task must be entered in the FactoryLink Configuration Manager (FLCM) System Configuration table. In this table for Modbus Plus driver the entry program arguments can be specified. The following arguments are valid for the Modbus Plus driver:

Argument	Description
-w	Exception write completion
-g	Unsolicited receive of global data

This page is left intentionally blank

Appendix A. Communication hardware installation

A.1 Windows 3.1 with the SA85 board.

Before installing the SA85 adapter a network node address and memory window base address must be set in a set of hardware switches. The node address identifies the SA85 for tokens and messages on the Modbus Plus network. The memory address defines a 2K bytes area in your computer that will be used as a buffer between the SA85 and your application.

Before using the SA85 adapter the SA85.SYS device driver must be installed. The software settings must match the hardware settings. The following is an example entry into your config.sys file:

```
file: \config.sys
...
device=c:\flwin\bin\sa85.sys /N0 /MCE00 /S5C /R2
...
```

The device command line options are:

/M Selects the memory window base address. Example: /MD080 = memory address D0800; /MC000 = C0000. The default is memory address D0000.

/N Selects a specific device within the host. Example: /N0 = device 0; /N1 = device 1. The default is device 0.

/S Selects the software interrupt vector. Example: /S5B = interrupt 5B. The default is interrupt 5C.

/R2 Defines driver revision 2.0. Required for normal operation.

After the hardware and software settings are complete the SA85 is ready to communicate. For more information on installing the SA85 refer to the Modicon IBM Host Based Devices User's Guide (GM-HBDS-001).

Attention: Make sure, if you use a memory manager, that you exclude the area used by the SA85. In the default configuration this area is **D000-D080**.

A.2 Windows NT with the SA85 board.

The hardware installation of the SA85 adapter on the Windows NT platform is identical as on the Windows platform so we'll concentrate to the software installation.

The DeltaLink Modbus Plus driver uses the SA-85 device driver of CYBERLOGIC and the standard SA-85 Modbus Plus adapter of Modicon. The device driver is very easy to install. Just place the disk labaled *Local NETLIB library for ModConnect Modbus Plus Interface Adapters* in the floppy driver and run the install utility. Follow the directions on the screen.

In case after the installation parameters have to be changed then this can be done by running the program *config* which is placed in the installation directory of the device driver. In case icons are generated at installation then the config utility can also be run from the Cyberlogic program group.

After the system has been rebooted a check can be made if the Modbus Plus device driver is running. This can be done by choosing the *devices* icon in the *Control Panel*. In the list there must be a MBPLUS entry. This entry represents the Modbus Plus device driver. The status of this entry must be *started* and the start-up field must be *automatic*. The start-up method automatic is preferable because every time the system boots, the Modbus Plus driver will be started.

After these steps the system is ready to run the DeltaLink Modbus Plus driver.

A.3 OS/2 with the SA85 board.

The hardware installation of the SA85 adapter on the OS/2 platform is identical as on the Windows platform so we'll concentrate to the software installation.

Before communication through the SA85 adapter is possible the following entries must be made in the config.sys (for example):

```
file: \config.sys
...
device=c:\flos2\bin\mphost.sys /N0 /MCE00
run=c:\flos2\bin\mphtick.exe /N0
...
```

The following entries should also be present in your config.sys file:

```
file: \config.sys
...
threads=128
memman=SWAP, MOVE
maxwait=3
timeslice=32,100
...
```

The device command line options are:

/M Selects the memory window base address. Example: /MD080 = memory address D0800; /MC000 = C0000. The default is memory address D0000.

/N Selects a specific device within the host. Example: /N0 = device 0; /N1 = device 1. The default is device 0.

After these entries are made and the system is rebooted the SA85 adapter is ready for communication.

A.4 HP-UX with the SA85 board.

The DeltaLink Modbus Plus protocol driver uses the SA85 communication driver of Commercial Timesharing Inc. also known as CTI. The SA85 driver of CTI uses the SA85 communication board of Modicon which is also used on the Windows (3.1 and NT) and OS/2 platforms. The SA85 board is an ISA based board which means that the HP hardware must contain an ISA or an EISA slot.

There is a major difference in jumpering the SA85 board with HP-UX and PC platforms. Normally the Polled Mode Jumper must be set to Polled Mode but with HP-UX the jumper must be installed on one of the IRQ interrupts. The other board settings can be choosen freely, the same as with other platforms.

The installation of the SA85 board driver will be discussed now. The HP-UX operating system provides a utility for configuring EISA slots called *eisa_config*. This utility will be used for the installation. Following steps must be taken:

- login as super-user on the system
- run *update* and install the driver from the tape

  A. select input from the appropriate tape device
  B. select the SA85 driver package for installation
  C. integrate the driver into the */etc* and */usr* file systems
  D. run SAM (use *dfile /etc/conf/dfile*) to build a new kernel and *reboot*

- configure the EISA slot and save into non-volatile memory

  A. run the *eisa_config* utility
  B. use *show* to determine the slot for installation. This may require that you move another board first.
  C. Add `!MOD0850` to the desired slot X.
  D. show board X to examine its current configuration.
  E. Change the configuration as desired. For example, to set the board IRQ 5 and memory to D4000h:

change X f1 ch4	Interrupt IRQ 5
change X f2 ch6	Memory address D4000h

  F. quit and save the configuration
  G. *shutdown -h* to shutdown and halt the system, jumper and install the SA-85 board and reboot the system.

While the system is rebooting, a message should be displayed that the SA-85 is initializing with all the configured addresses. If this is the case then the SA-85 has been installed correctly. The */users/cti/sa85/utils* directory contains a program *sa85_diag* which can be used for checking the installation and monitoring the Modbus Plus network.

For more details on jumpering the SA-85 refer to the *Modicon IBM Host Based Devices User's Guide* delivered with the SA-85 board. For more details on installing the SA-85 driver software refer to the *Modicon SA85 Driver for Unix* manual delivered with the CTI driver.

Appendix B. The delta.opt file

In order to run the Modbus Plus driver permanently with full functionality either an authorisation sequence or a protection hardware key with the Modbus Plus option is needed. This appendix describes the authorisation sequence together with the option file.

The authorisation sequence of every independent DeltaLink module must reside in the *{FLOPT}/delta.opt* file in order to take effect. In case the Modbus Plus driver has been ordered with an authorisation sequence then this sequence resides on the install diskette (in the *\opt\delta.opt* file) and will be automatically added to the *{FLOPT}/opt file* with the installation. In case the Modbus Plus driver has been previously installed from a demo diskette and the authorisation sequence has been purchased later then the sequence must be added manually to the *{FLOPT}\delta.opt* file.

The format of the delta.opt file consists of two parts. The first part is the comment header. This part remains always at the beginning of the file. Every line of the comment part starts with an '*' character. The second part contains the authorisation sequences of every independent DeltaLink module. Every line must hold one sequence code and must apply to a strict format.

An example of the option file looks like this:

```
*
* Copyright 1994 DeltaLink bv. All Rights Reserved
*
* DeltaLink bv
* Saffierborch 6c
* NL-5241 LN Rosmalen
* The Netherlands
* Tel: (int) 31 4192 20025
* Fax: (int) 31 4192 15451
*
*
* FactoryLink Serial Number: 12345OS2
*
* DeltaLink module option:
*
* MODULE = taskname of always 8 characters with trailing spaces
* .      = <space>
* X      = authorization code supplied by DeltaLink
*
* MODULE..XXXX.XXXX.XXXX.XXXX.XXXX.XXXX.XXXX.XXXX.XXXX
MB_PLUS   2D6F E047 2534 036E 1215 EC80 92F8 1EFC C4C0
DECODER   B623 C712 D3A7 A08D 8F96 9F14 017B 7D7C F073
```

In case a full installation diskette with authorisation diskette has been ordered then the header in the option file on the diskette contains the serial number of FactoryLink. The Modbus Plus driver will run only on the FactoryLink package with this serial number. If the serial number is not listed in the *{FLOPT}\delta.opt* file due to a previous demo installation then this number can be added in the header of this file.

The authorisation code must exactly match the format as listed in the header. If this is not the case the module will not recognise the authorisation sequence and start up in demo mode.

The format of an authorisation sequence line is as follows:

MODULE<s><s>XXXX<s>XXXX<s>XXXX<s>XXXX<s>XXXX<s>XXXX<s>XXXX<s>XXXX<s>XXXX<CR><LF>

The **MODULE** field contains the DeltaLink module name in this case Modbus Plus. This field must always be 8 characters long. If the module name is shorter then 8 characters then the name must be filled out with spaces to 8 characters.

After the MODULE field one space must be entered;

After the space field 9 records must be specified with the authorisation code. One record is build up of one leading space (ASCII 0x20) and four sequence codes. The sequence codes must be entered exactly as specified by DeltaLink.

After the authorisation code records a carriage return (ASCII 0x13) and linefeed (ASCII 0x10) must follow.

There may be no empty lines between the specification of more than one module. To add an authorisation sequence a normal editor can be used. If all modules with the right authorisation codes are specified according to the format described above then the modules will start with full functionality.

Appendix C. The protection key

B.1 Windows 3.1

The protection key is placed in a parallel port of the FactoryLink workstation. The 386 enhanced mode of Windows 3.1 'manages' all accesses to the parallel port while it is running. To insure conflict free operation, a Windows device driver must be included in the Windows SYSTEM.INI file.

Port management is in force whenever the 386 enhanced mode of Windows is active in the machine. Non-Windows applications running in a protected mode DOS-shell can conflict with other tasks trying to use the parallel port.

Copy the VPXD.386 device driver from the directory {FLINK}\bin (or the install diskette of the protocol driver) to the \WINDOWS\SYSTEM directory. Add the following line to the SYSTEM.INI file under the section headed [386Enh].

```
device=VPXD.386
```

Remark: Some printers tie their control lines to ground when they are powered down. This can negatively effect the protection keys ability to operate. While an attached printer is not required for operation, if one is present it should be powered up and on-line.

B.2 OS/2

The protection key is placed in a parallel port of the FactoryLink workstation. The OS2 operating system 'manages' all accesses to the parallel port while it is running. To insure conflict free operation, an OS2 device driver must be included in the CONFIG.SYS file.

Copy the DOWOS2.SYS device driver from the directory {FLINK}\bin (or the install diskette of the protocol driver) to the \OS2\DRIVERS directory. Add the following line to the CONFIG.SYS file.

```
DEVICE=C:\OS2\DRIVERS\DOWOS2.SYS
```

In case the device driver is not installed at all (or incorrectly) any task protected with a DeltaLink key will fail to startup. An error message will be generated indicating that the protection device failed.

Remark: Some printers tie their control lines to ground when they are powered down. This can negatively effect the protection keys ability to operate. While an attached printer is not required for operation, if one is present it should be powered up and on-line.

This page is left intentionally blank

Appendix D. Modbus Plus 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 DelatLink Modbus Plus protocol driver in the run-time manager. The errors can be generated from different parts of the driver which will be listed here:

Modbus Plus protocol errors

Error #	Description
1	Bad send or status buffer size
3	invalid NetBIOS command
5	Command time-out has expired
6	Receive buffer not big enough
8	Bad value in NCB_LSN
9	LAN card doesn't have enough memory
10	This session is closed
11	Command has been closed
13	Name already exists for this PC
14	Local name table is full
15	Can't delete name - used in session
17	Local session table is full
18	Remote PC not listening for call
19	Bad value in NCB_NUM field
20	No answer to CALL or no such remote
21	No such name in local name table
22	Name is in use elsewhere on net
23	Name incorrectly deleted
24	Session aborted abnormally
25	2 or more identical names in use!
26	Bad NetBIOS packet on network
33	network card is busy
34	Too many NetBIOS commands queued
35	bad NCB_LANA_NUM - must be 0 or 1
36	command finished while cancelling
37	Command can't be cancelled
255	Still processing command

Error #	Description
201	Illegal function for the addressed slave
202	Illegal data address within the information field for the addresses slave
203	Illegal data value in the information field for the addresses slave
206	Busy, the function just requested cannot be performed at this time because a long-duration PROGRAM command is being processed.

Internal MP communication layer errors.

Error #	Description
210	bad addressing path specified (master/slave)
211	no adapter present in local system
212	wrong plc type to write to
213	NETLIB initialazation failed
214	Adapter diagnostics failed
215	non existing MODICON type
220	no encoding for this request
0x8012	linear memory unavailable (stack)
0x8013	physical memory unavailable (stack)
0x8014	backing store unavailable (stack)
0x8021	invalid value (CX too large)

Mailbox Communication Interface errors

Error #	Description
100	bad message type
101	message with dataset control tag not found in queue
102	no messages available to query
103	bad receive mailbox tag
104	bad mailbox send tag
105	bad dataset control tag

Protocol driver errors

Error #	Description
500	invalid addressing method on bit boundary
501	exception write not possible to destination
502	Modbus Plus driver has been shutdown

FactoryLink errors

All FactoryLink error start with an offset of 400 decimal. For a complete description for each individual error refer to the FactoryLink documentation. For example: error #402 can be found in the FactoryLink manual as error 2 (402 - 400) which means no more memory available (FLE_OUT_OF_MEMORY).

Error #	Description
401	Internal error
402	Out of memory
403	Operating system error
404	Initialisation not successfull
405	Initialisation not successfull
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	Wakeup failed
433	No signals
434	Signalled
435	Not a mailbox
436	No messages
437	Access denied
438	Attribute failure
439	Invalid attribute
440	Attribute not defined
441	Application exists
442	RTDB does not exist
443	No task bit
444	Not a lite task

This page is left intentionally blank

Appendix E. 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 *MB_PLUS*. The error messages that may be displayed are as follows:

Can't open CT file

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

No triggers defined

The protocol driver configuration table file does not contain an expected trigger. This is a fatal error.

No DataSets defined

The protocol driver configuration table file does not contain a valid dataset definition. This is a fatal error.

Out of RAM

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

Invalid tag number

The protocol driver encountered an invalid TAG number. This is a fatal error.

Bad tag %s %s %d

The protocol driver encountered an incorrect TAG definition. The TAG number, segment and offset of the specific TAG are displayed.

PD write not in range, %s db %d start %d len %d

Write command from the protocol driver is not in range with the dataset in the PLC. Specified are the address number in the dataset, the DB number corresponding with the dataset, start address and length of the dataset.

MCI unknown msg type %d

MCI error, internal error of the protocol driver task. Specified is the number of an unrecognised command.

MCI Bad MSG Control TAG table %s record %d type %d

MCI error, internal error of the protocol driver task. Specified are table number, record number and type number.

MCI Duplicate MSG Ctrl TAG table %s record %d type %d

MCI error, internal error of the protocol driver task. Specified are table number, record number and type number.

MCI Bad Mailbox TAG table %s record %d type %d

MCI error, internal error of the protocol driver task. Specified are table number, record number and type number.

MCI DS for %s function not found, MsgCtrl %d

MCI error, internal error of the protocol driver task, missing dataset for the task. Specified are function and the message control TAG.

Com function %s on %s, %s, %d

Modbus Plus communication error. The error number is equal to error numbers in *Appendix C*.

Com unknown DataSet type

Modbus Plus communication error, an unknown type is specified for conversion of a dataset element.

Internal error: unknown state of job %s

Internal error of the protocol driver task, contact DeltaLink bv for support.

Appendix F. Encoded Write Lay-Out

The encoded write functionality writes a command to the PLC which must decode and perform this command. This way all elements can be written at once without reading them before writing. This will increase the performance of the write functionailty especially in case of exception bit writes.

The encoded write function will write the command to a ceratin address in the PLC. This must be a HREG address and must be specified in the *DeltaLink Modbus Plus Device definition* panel in the *Encoded Reg* field. The command exists of maximum 5 holding registers an has the following format:

Origination	Variable type
Address	
Bit number	Change flag
Variable	
Variable	

The origination field of the first register can be of the following type:

'O'	output or coils elements 0xxxx
'I'	input elements 1xxxx
'A'	analog registers 3xxxx
'H'	holding registers 4xxxx

The origination indicates the destination type of the variable. Note that the values are ASCII values.

The variable type can be one of the following types:

'F'	bit operation
'B'	byte operation
'L'	left byte operation
'R'	right byte operation
'W'	word operation
'D'	long operation

The variable type indicates the source type of the variable. The length of the variable in the encoded command depends on this value. This value is also specified in ASCII code.

The address element reserves one word in the encoded command and specifies the destination address of the variable.

The bit number specifies the destination address of the bit in case a ‘F’ operation has been specified in the variable type field. This field is dummy in all other cases.

The change flag will always be set to the value 0xFF. This field must be used by the PLC program to detect if an encoded write command has been received. This filed will be reset by the PLC program for receiving the next command.

The variable field holds the value of the variable according to the variable type field. The length of the variable can be maximum two holding registers which corresponds to the ‘D’ type.

This page is left intentionally blank