Advant® OCS
with Master software

VIP, Vendor Internet Protocol TCP/IP

Advant® Controller 450RMC

User’s Guide

NOTICE

The information in this document is subject to change without notice and should not be construed as a commitment by ABB Industrial Systems AB. ABB Industrial Systems AB assumes no responsibility for any errors that may appear in this document.

In no event shall ABB Industrial Systems AB be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB Industrial Systems AB be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without ABB Industrial Systems AB's written permission, and the contents thereof must not be imparted to a third party nor be used for any unauthorized purpose.

The software described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license.

3BSE001264/C

3BSE 014 823 R0001

TABLE OF CONTENTS

Chapter 1 - Introduction

1.1 General 9
1.2 Purpose of the manual 9
1.3 Abbreviations 9

Chapter 2 - Configuration

2.1 General 11
2.2 Network Configuration 12
2.3 Node Configuration 12
2.4 Link Configuration 12
2.5 Channel Configuration 12
2.6 Application Message Configuration 13
2.7 MasterBus 300 Routing 14

Chapter 3 - Hardware and Strapping

3.1 General 15
3.2 Communication Interface CI546 15
3.2.1 Strappings on CI546 16
3.3 Communication Interface PM510 17
3.3.1 Submodule CS513 18
3.3.2 Strappings on CS513 18

Chapter 4 - Performance and Capacity

4.1 Capacity 19
4.2 CPU Load In CI546 19
4.3 CPU Load In PM510 19
4.4 CPU Load In Main CPU with Local Network Interface 20
4.5 CPU Load In Main CPU with Remote Network Interface 21
4.6 CPU Load In MB300 Transit Node 22

Chapter 5 - PC element library

5.1 PC Element Execution Times 24
5.2 Memory requirement 25

Chapter 6 - Error Handling and Error Codes

6.1 General 27
6.2 Error Codes 27

PC Element TIME-CON 31

PC Element TIME-SET 33

PC Element VIP-CHAN 35

PC Element VIP-LINK 39

PC Element VIP-NETW 43

PC Element VIP-NODE 45

PC Element VIP-R 47

PC Element VIP-W 51

Appendix A - Protocols And Message Formats

A.1 General 55
A.2 Data Transmit Ordering 55
A.3 Structure of Header Part, Protocol 0 56
A.4 Structure of Data Part 56
A.5 Message Formats of Data Part 57
A.5.1 16-bit Integer data 57
A.5.2 16-bit Packed Boolean data 57
A.5.3 32-bit Packed Boolean data 57
A.5.4 32-bit Integer data 57
A.5.5 Real data 58
A.5.6 Group data 59
A.5.6.1 General 59
A.5.6.2 Format 0 59
A.5.6.3 Format 1 59
A.5.7 Byte Array data 60
A.5.7.1 General 60
A.5.7.2 Example 60
A.5.7.3 Format 0 60
A.5.7.4 Format 1 61
A.5.7.5 Format 2 61
A.5.7.6 Format 3 61

Appendix B - Internet Concepts

B.1 Overview 63
B.2 Internet Protocol (IP) 63
B.3 User Datagram Protocol (UDP) 63
B.4 Transmission Control Protocol (TCP) 64
B.5 Sockets 64

TABLES

Table 5-1. PC Element Library 23
Table 5-2. PC Element Execution Times 24
Table 5-3. PC Element Memory Requirement 25
Table 6-1. Network Errors 27
Table 6-2. Node Errors 28
Table 6-3. Link Errors 28
Table 6-4. Channel Errors 29

ILLUSTRATIONS

Figure 2-1. Configuration Tree 11
Figure 2-2. MB300 Routing Example 14
Figure 3-1. CI546 15
Figure 3-2. CI546 Strap Groups 16
Figure 3-3. Rack Layout with TCP/IP interface 17
Figure 3-4. CS513 18
Figure 3-5. CS513 Strap Groups for CS513 18
Figure 4-1. CI546 Processor Load 19
Figure 4-2. Main CPU Load with Local TCP/IP Network Interface. 20
Figure 4-3. Main CPU Load with Remote TCP/IP Interface and MB300 in SC510 21
Figure 4-4. Main CPU Load with Remote TCP/IP Interface and MB300 in SC520 21
Figure 4-5. Transit Load In Main CPU with MB300 in SC520 22
Figure A-1. VIP Message 55
Figure A-2. Multi Byte Order 55
Figure A-3. Message Header, Protocol 0 56
Figure A-4. Data Part 56
Figure A-5. Group Data Format 0 59
Figure A-6. Group Data Format 1 59
Figure A-7. Byte Array Example 60
Figure A-8. Byte Array Format 0 60
Figure A-9. Example, Byte Array Format 0 60
Figure A-10. Byte Array Format 1 61
Figure A-11. Example, Byte Array Format 1 61
Figure A-12. Byte Array Format 2 61
Figure A-13. Example, Byte Array Format 2 61
Figure A-14. Byte Array Format 3 61
Figure A-15. Example, Byte Array Format 3 61

Chapter 1 Introduction

1.1 General

The Vendor Internet Protocol (VIP) function is intended to be used for communication between AC 450RMC controllers and external computers supplied by vendors outside ABB, but can also be used for internal communication between AC 450RMC controllers.

The VIP function supports Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) built on the Internet Protocol (IP) for ethernet, i.e. IEEE 802.3 CSMA/CD medium access control.

1.2 Purpose of the manual

The purpose of this manual is to provide an application engineer with information regarding possibilities and facilities provided by the Vendor Internet Protocol (VIP) function.
The document will also serve as a guide for configuration of communication links and exchanging of application messages with external computers.

It is assumed that the reader of this document is familiar with application programming regarding database handling and PC programming.

1.3 Abbreviations

IP	Internet Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VIP	Vendor Internet Protocol
RMC	Rolling Mill Control
AC 450RMC	Advant Controller 450RMC

Chapter 2 Configuration

2.1 General

Configuration of communication links is done by means of a data base element and the PC elements below.

- VIP-NETW Configuration of network
- VIP-NODE Configuration of nodes
- VIP-LINK Configuration of links
- VIP-CHAN Configuration of channels

A configuration can be illustrated as a tree structure according to Figure 2-1.

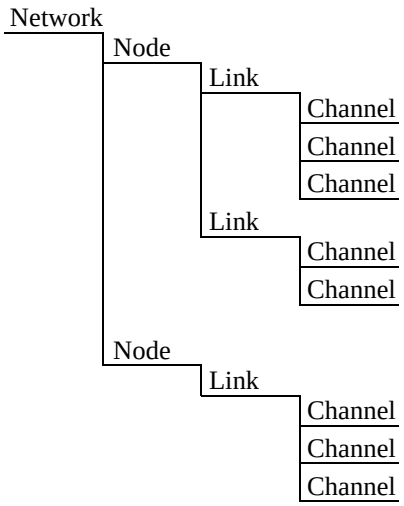


Figure 2-1. Configuration Tree

2.2 Network Configuration

DB element CI532 and PC element VIP-NETW is used for configuration of the network. The element constitutes the interface to the MasterNet function that will associate the network interface with a MasterNet network number.

The PC element VIP-NETW is used for initialization of the DB element and applying an internet address and a host name for the network interface. Internet address for gateway and remote network can also be specified for the network.

NOTE

A CI532 DB element must be created for each network interface card. The IMPLEMENTED property must be set to 0 by the user to allow the PC element VIP-NETW to initialize the DB element.

2.3 Node Configuration

The PC element VIP-NODE is used for associating a MasterNet node number for each external computer connected the same MasterNet network. An internet address is specified for each node number. Optionally a host name may be specified for each node.

2.4 Link Configuration

The PC element VIP-LINK is used for configuration of links to an external computer. The parameter listed below are used for definition of a link.

- Port (service), well defined and known by both end of the link
- Protocol
- Direction of the link: send, receive or bidirectional
- Transport protocol type: TCP or UDP
- Client or server for TCP links

2.5 Channel Configuration

The PC element VIP-CHAN is used for configuration of message channels for a link. Each channel is associated with a specific application message, i.e. one channel is created for each message type. A sending channel is used as an interface to the transmitting PC element VIP-W. A receiving channel is used as an interface to PC element VIP-R. A message queue is confederated for each receiving channel to be able to queue application messages when event driven communication is used.

2.6 Application Message Configuration

Configuration of an application messages is implicitly defined by PC element VIP-W for sending messages and VIP-R for receiving messages. Each PC element is connected to a message channel created by PC element VIP-CHAN.

The order, and the data types that are supported by the PC elements are listed below.

- 16 bit integer values
- 32 bit integer values
- Real values
- Group data arrays
- Byte array arrays

The total number of data inputs/outputs is limit to 45 terminals per PC element. PC elements can be linked to handle larger application messages.

The format of the message is specified by means of PC element VIP-CHAN element, e.g. byte arrays can be null terminated or the size of the array may be included.

The interpretation of an application message is implicitly defined by the configuration of the corresponding VIP-R PC element. The order of the data item in the message is assumed to be the same order as the order of the data output terminals of the PC element.

2.7 MasterBus 300 Routing

VIP messages can be transmitted/received in any AC450 node that is connected to the same MB300 network as the node containing the TCP/IP interface module.

Node 2 in Figure 2-2 below act as a transit node for node 1,3 and 4. Network 11 is used for node 3 and 4, and network 12 is used for node 1. Node 2 is not directly connected to network 13 and can not act as a transit node for node 5.

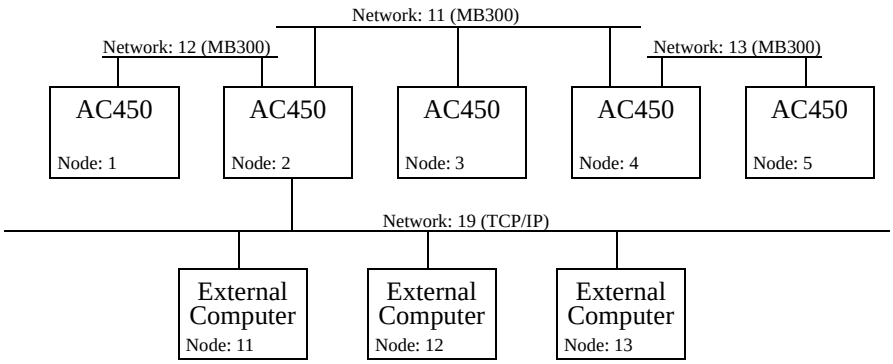


Figure 2-2. MB300 Routing Example

The routing scheme is determined by PC element VIP-CHAN. A message received from an external node is routed to all nodes containing a PC element VIP-CHAN defining the channel associated with the message. A node sending a message to an external computer must contain the VIP-CHAN element defining the sending channel.

NOTE

The real time performance in the transit node and on the MB300 network cable should be considered when large amount of transit traffic is used.
Refer MasterNet™ User's Guide.

Chapter 3 Hardware and Strapping

3.1 General

The modules below are available as TCP/IP network interface for the VIP function.

- CI546
- PM510

The PM510 module is supported for backward compatibility. PM510V with 16 MByte RWM is not supported.

3.2 Communication Interface CI546

CI546 is a small circuit board adapted to the Module Interconnection Bus (MIB) and intended for insertion in SC510 and SC520. The board contains:

- a CPU (MC68360)
- an integrated Local Area Communication Controller which realizes the physical level of communication defined in IEEE 802.3 standard
- RWM
- PROM which contains the TCP/IP communication stack.

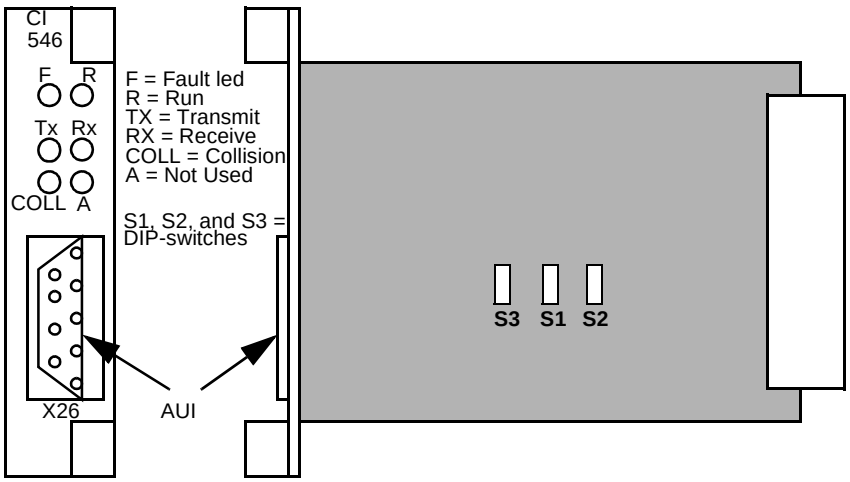


Figure 3-1. CI546

DIP-Switches

The DIP-switch S1 is used for node number. The DIP-switches S2, and S3 are NOT used.

CAUTION

Always disconnect the transceiver cable (DSTK 128/129, TK 576Vxxx) from the X26 contact of CI546 before inserting or removing CI546.

NOTE

The SC510/SC520 carrier board holding the CI546 submodule must not be shared with other main CPUs.

3.2.1 Strappings on CI546

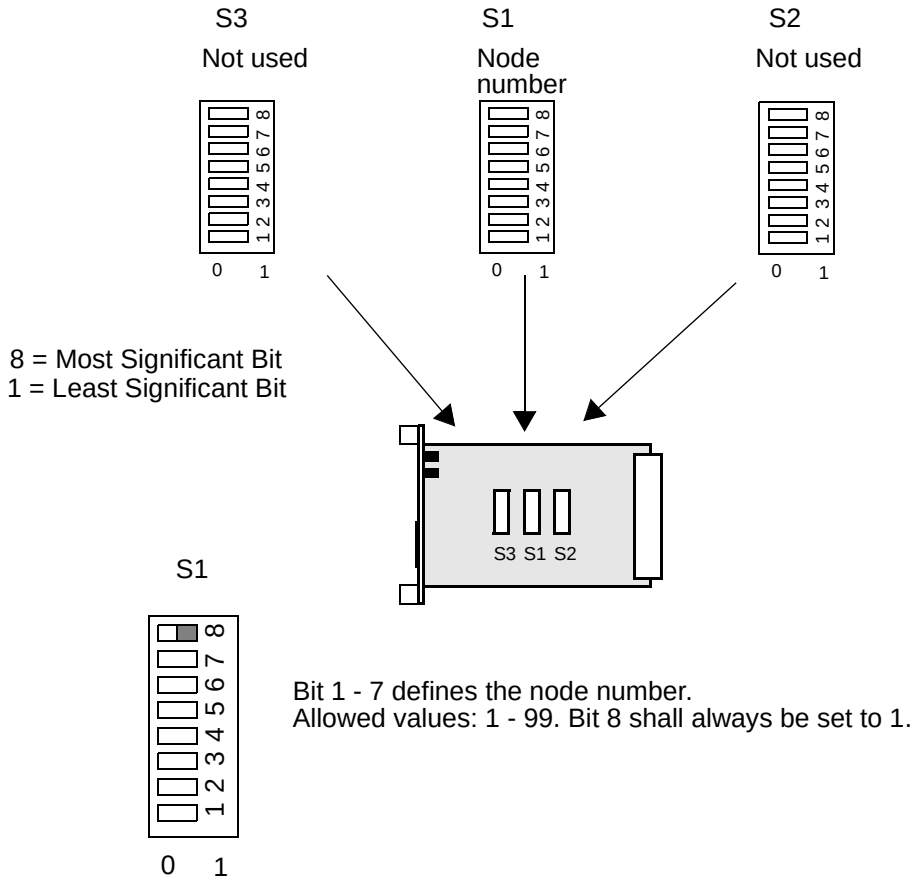


Figure 3-2. CI546 Strap Groups

3.3 Communication Interface PM510

Using processor module PM510 for the VIP function requires following hardware components for each VIP network:

- Processor module PM510, used for execution of the protocol stack. The MODE switch on the front panel shall always be in position CLEAR.
- Submodule CS513, used as network interface
- Submodule MB510, used as carrier for the flash prom card
- Flash prom card containing software for the TCP/IP communication stack
- Carrier board SC510, used as holder for the submodules.

Each VIP network will occupy two slots in the central rack. The carrier board SC510 must be located just to the right of the PM510 module.

The PM510 module must be placed to the right of the CI540 module to prevent the main CPU from catching the CS513 module used for TCP/IP.

The base system QC04-BAS04 is required for the main PM510 boards to get the boot function for the TCP/IP CPU to work properly. The contents of the flash prom will be loaded at each start-up of the PM510 module.

Figure 3-3 shows how to place the TCP/IP hardware in a RF520 rack. Slot 7 and 8 are used for the TCP/IP interface.

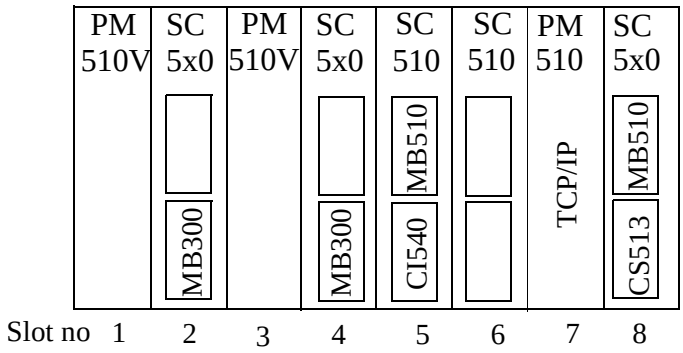


Figure 3-3. Rack Layout with TCP/IP interface

3.3.1 Submodule CS513

CS513 is a small circuit board adapted to the Module Interconnection Bus (MIB). The board contains an Integrated Local Area Communication Controller which realizes the physical level of communication defined in the IEEE 802.3 standard.

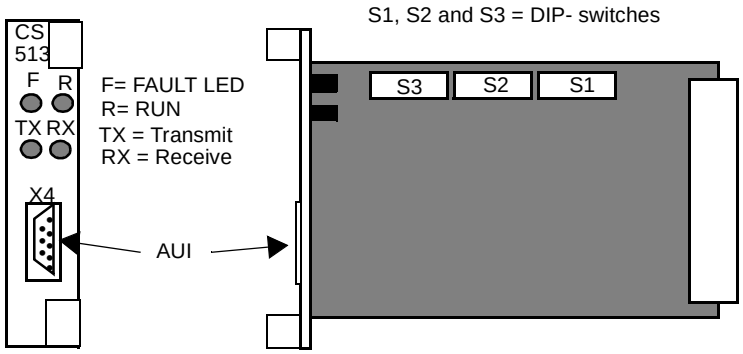


Figure 3-4. CS513

DIP-Switches

The DIP-switch S2 is used for node number. The DIP-switches S1, and S3 are NOT used.

3.3.2 Strappings on CS513

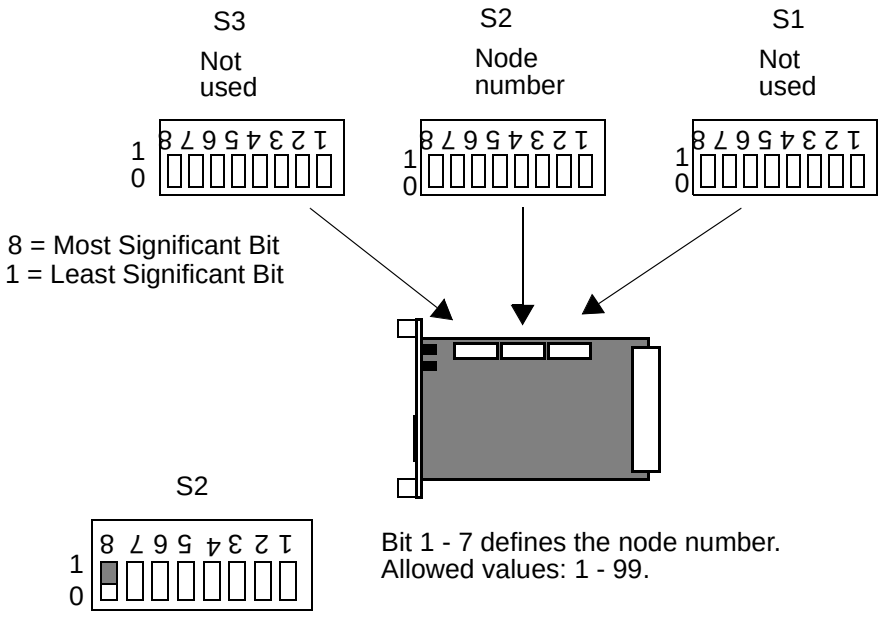


Figure 3-5. CS513 Strap Groups for CS513

Chapter 4 Performance and Capacity

4.1 Capacity

Number of networks:	9 per AC450
Number of external nodes:	20 per network
Number of links:	50 per network
Number of channels:	200 per network
Application message size:	65535 bytes (TCP) 9500 bytes (UDP)

4.2 CPU Load In CI546

Figure 4-1 shows the CPU load in the CI546 module. The total load for sending and receiving messages should not exceed 80 percent.

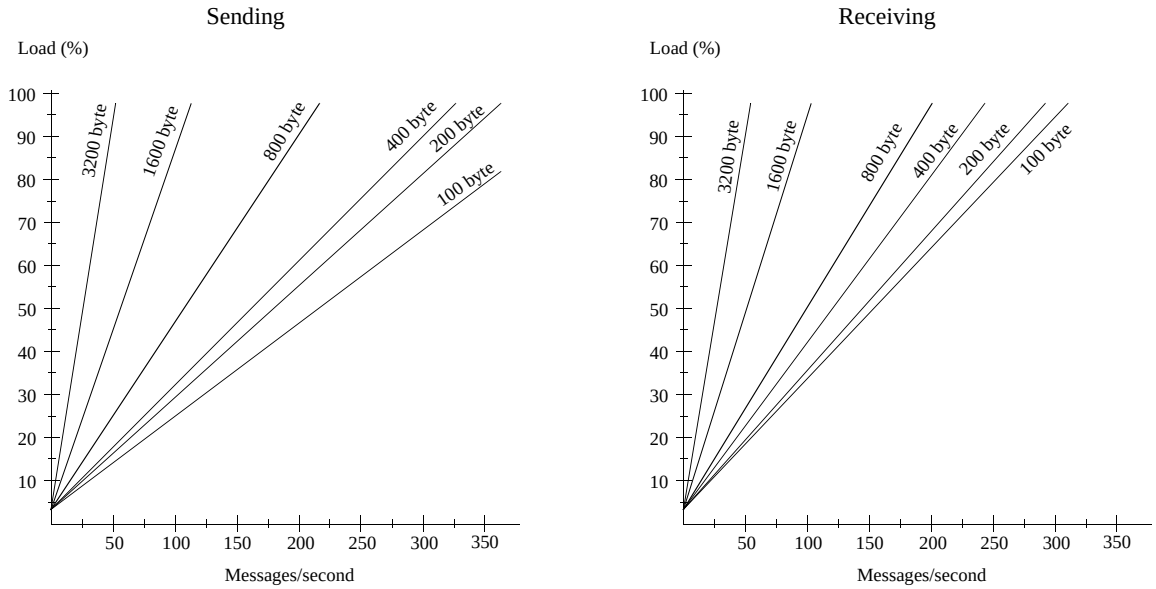


Figure 4-1. CI546 Processor Load

4.3 CPU Load In PM510

The CPU load in PM510 as network interface can be calculated as 1.1 x $Load_{CI546}$

4.4 CPU Load In Main CPU with Local Network Interface

Figure 4-2 shows the load in the main CPU with a local TCP/IP network interface. The receiving diagram shows the load exclusive the PC element load, i.e the load for PC element VIP-R should be added to get the total receive load.

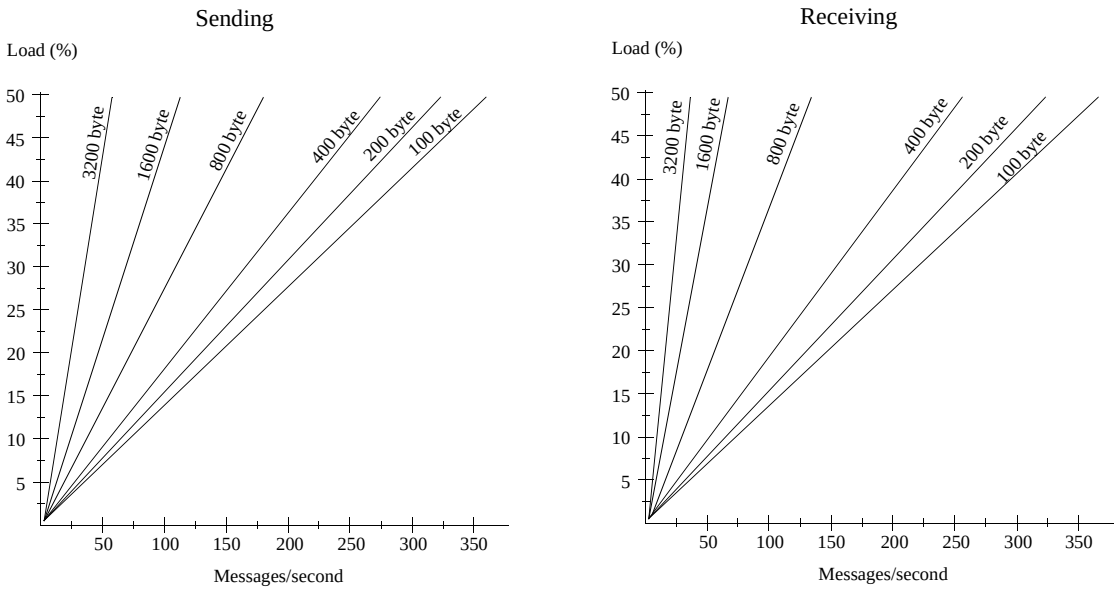


Figure 4-2. Main CPU Load with Local TCP/IP Network Interface.

4.5 CPU Load In Main CPU with Remote Network Interface

Figure 4-3 shows the load in the main CPU with a remote TCP/IP network interface with MB300 interface in carrier board SC510. Figure 4-4 shows the load with MB300 interface in carrier board SC520.

The receiving diagram shows the load exclusive the PC element load, i.e the load for PC element VIP-R should be added to get the total receive load.

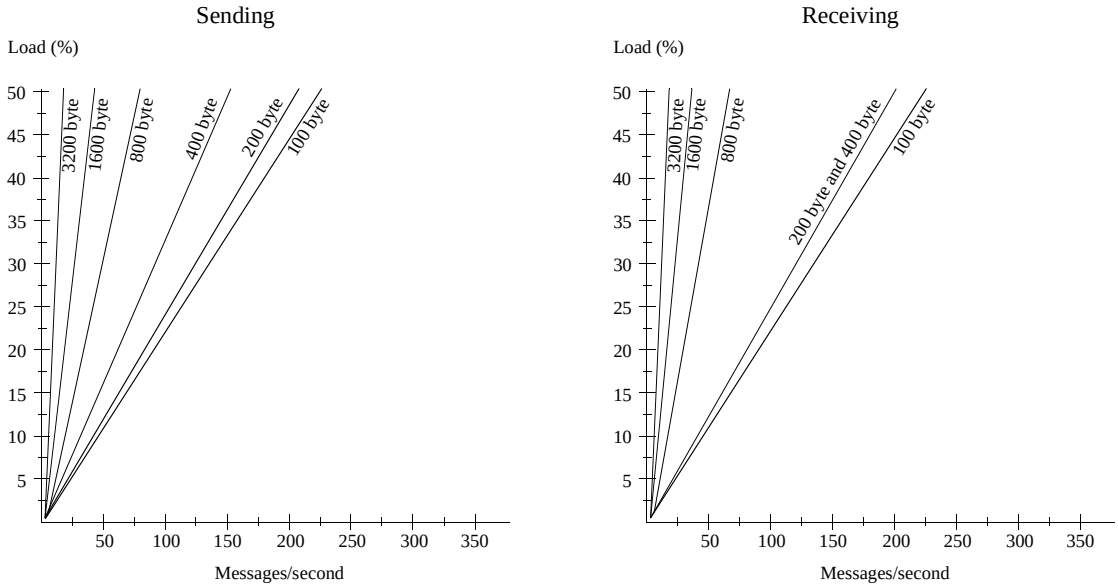


Figure 4-3. Main CPU Load with Remote TCP/IP Interface and MB300 in SC510

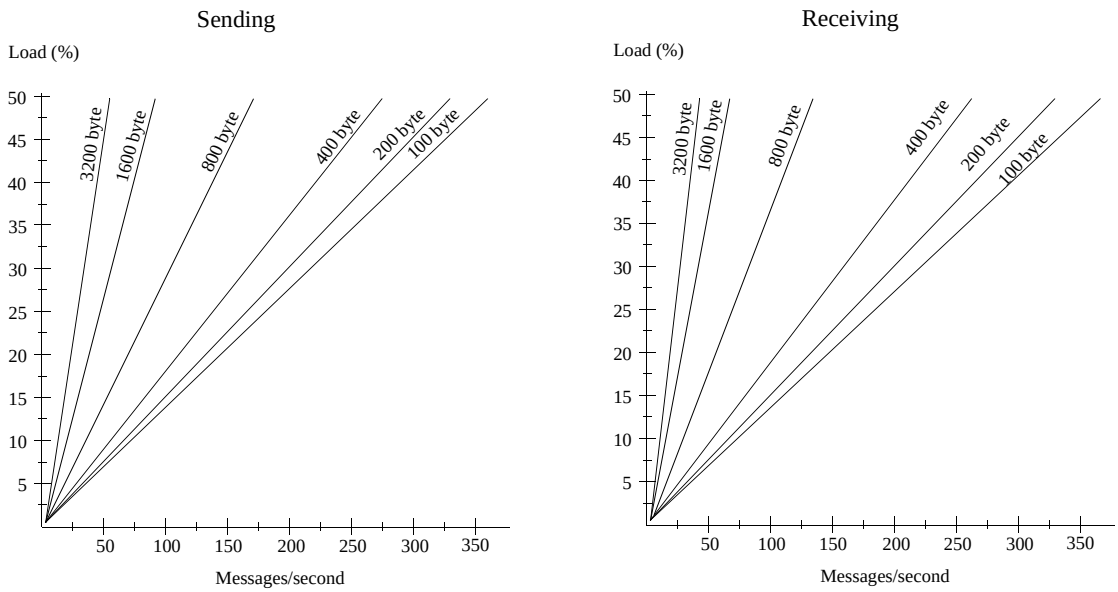


Figure 4-4. Main CPU Load with Remote TCP/IP Interface and MB300 in SC520

4.6 CPU Load In MB300 Transit Node

Figure 4-5 shows the load in the main CPU when it is used as a transit node between TCP/IP and MB300 networks. Carrier board SC520 is used for the MB300 interface.

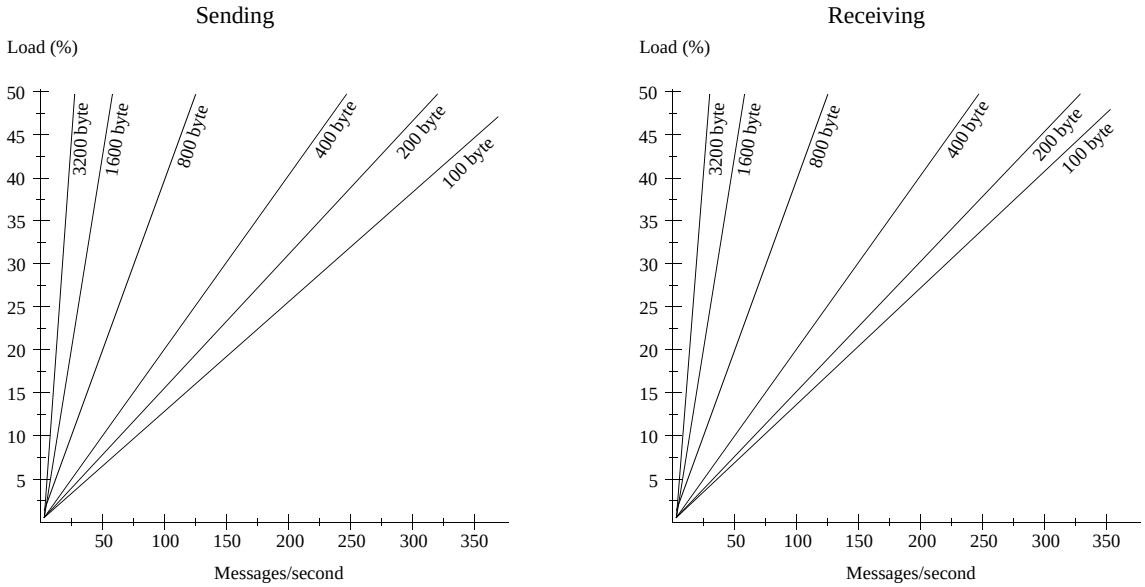


Figure 4-5. Transit Load In Main CPU with MB300 in SC520

Chapter 5 PC element library

Table 5-1 briefly describes the PC elements.

Element	Description
VIP-NETW	Used for configuration and supervision of a VIP network in order to communicate via TCP/UDP links to external computers.
VIP-NODE	Used for configuration and supervision of nodes connected to a VIP network.
VIP-LINK	Used for configuration and supervision of TCP and UDP links to an external computer.
VIP-CHAN	Used for configuration and supervision of channels for a TCP/UDP link.
VIP-R	Provides the application programme with an interface for receiving data on a TCP/UDP link from an external computer.
VIP-W	Provides the application programme with an interface for sending data on a TCP/UDP link to an external computer.
TIME-CON	Used for conversion of date and time into internal system time representation.
TIME-SET	Used for setting of date and time in the internal system time representation.

Table 5-1. PC Element Library

5.1 PC Element Execution Times

Table 5-2 below shows the execution times for PC elements with AC 450.

Element	Typical execution time (µs)	Notes
TIME-CON	32	
TIME-SET	5	
VIP-CHAN	20 + 4·C1	
VIP-LINK	30 + 15·C1	
VIP-NETW	28	
VIP-NODE	17 + 15·C1	
VIP-R	72 (if no message is available). Otherwise: 150 + 2·C1 + 3.4(C2+C3) + (40 + 0.4·C7)·C8 + (22 + 1.6·C5)·C6	Add load for network handler and for VIP message handler. Please see "VIP User's Guide".
VIP-W	200 + 30x + 1.6·C1 + 2.8·(C2+C3) + (30+1.5·C5)·C6 + (30+1.5·C7)·C8 + 244y + 0.95z;	x=1 if C1+C2+C3>0; Otherwise x=0; y= number of buffers required for message = = (message size)/498; z= message size = 2·C1 + 4·(C2+C3) + (2+n·C5)·C6 + (2 + C7)C8; n = 2 if C4 = I. n = 4 if C4 = IL or R.

Table 5-2. PC Element Execution Times

5.2 Memory requirement

Table 5-3 below shows the memory requirement for the PC elements. About 300 Kilobyte additional RWM memory is required besides the memory used for the PC elements that will be allocated only when the VIP function is used in a node.

Element	PC statement (bytes)	Local data area (bytes)	Note
TIME-CON	44	10	
TIME-SET	20	14	
VIP-CHAN	22 + 24·C1	16 + 4·C1	
VIP-LINK	20 + 26·C1	16 + 4·C1	
VIP-NETW	46	40	
VIP-NODE	18 + 22·C1	16 + 4·C1	
VIP-R	52 + 6(C1+C2+C3) + 8(C6+C8 +2·C9)	36+2·C1+4(C2+C3) + C6(2+n·C5)+C8(2+C7) + C9(4+2·C6+2·C8)	n = 2 if C4 = I; n = 4 if C4 = IL or R;
VIP-W	56+2(C1+C2+C3+C6+ C8+2·C9)	30	

Table 5-3. PC Element Memory Requirement

Chapter 6 Error Handling and Error Codes

6.1 General

The error types are divided into four categories as below.

- Network errors
- Node errors
- Link errors
- Channel errors

Network errors are shown on all configuration PC elements, i.e VIP-NETW, VIP-NODE, VIP-LINK and VIP-CHAN.

Node errors are shown on PC element, VIP-NODE, VIP-LINK and VIP-CHAN.

Link errors are shown on PC element VIP-LINK and VIP-CHAN.

Channel errors are shown on PC element VIP-CHAN only.

6.2 Error Codes

The tables below describes all the error codes indicated on the VIP elements.

Code	Description
101	Network unknown. Board removed or network not configured, refer PC element VIP-NETW and DB element CI532.
102	Network configuration in progress.
103	Network connection broken.
104	Network parameter error.
105	Network insufficient resources.
106	Network already configured.

Table 6-1. Network Errors

Code	Description
201	Node unknown, refer PC element VIP-NODE.
202	Node configuration in progress.
203	Node connection broken.
204	Node parameter error.
205	Node insufficient resources.
206	Node already configured.

Table 6-2. Node Errors

Code	Description
301	Link unknown, refer PC element VIP-LINK.
302	Link configuration in progress.
303	Link connection broken.
304	Link parameter error.
305	Link insufficient resources.
306	Link already configured.
307	Port and protocol conflict. Using different protocols for the same port is not allowed.
308	Unknown protocol specified.

Table 6-3. Link Errors

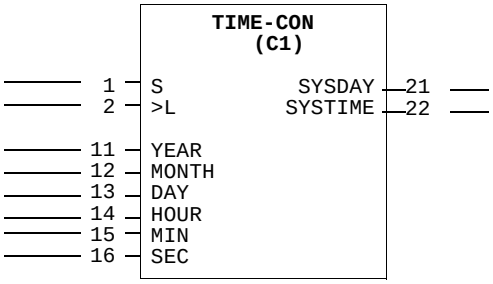
Code	Description
401	Channel unknown, refer PC element VIP-CHAN.
402	Channel configuration in progress.
403	Channel connection broken.
404	Channel parameter error.
405	Channel insufficient resources.
406	Channel already configured.
407	Invalid use of FIRST and LAST inputs, refer PC element VIP-W/VIP-R.
408	Channel receive queue is empty
409	Channel queue overflow
410	End of message detected before all outputs of VIP-R is updated, i.e. configuration of PC element does not fit the message.
411	Lack of message buffers for channel queue. Received message is dropped.
412	Transmit overflow. Sent message is dropped.
413	Receive overflow. Received message is dropped.
414	Receive overflow.
415	Transmit overflow.
416	Lack of memory for allocation of channel pool.
417	End of channels.
418	Channel lock-up table full.
419	Channel receive time-out.
420	Illegal message length.
421	Illegal buffer length.
422	Illegal direction.

Table 6-4. Channel Errors

Time Conversion TIME-CON

Summary

TIME-CON is used for conversion of date and time into internal system time representation.



Call TIME-CON(C1)

Call Parameter

Parameter	Significance	Permissible values
C1	Data type of date and time inputs	I, IL

Terminal Description

Number	Name	Type	Description
1	S	IB	**S**et activates continually the conversion of data.
2	L	IB	**L**oad activates the conversion of data when it goes from "0" to "1".
11	YEAR	C1	The **YEAR** input value must be greater (or equal) to 1980:
12	MONTH	C1	The **MONTH** input value must be within "1" up to "12".
13	DAY	C1	The **DAY** input value must be within "1" up to "maximum day of month".
14	HOUR	C1	The **HOUR** input value must be within "0" up to "23".
15	MIN	C1	The **MIN**ute input value must be within "0" up to "59".
16	SEC	C1	The **SEC**ond input value must be within "0" up to "59".
21	SYSDAY	OL	Internal representation of **SYS**tem **DAY**. Day number since 1980-01-01.
21	SYSTIME	OL	Internal representation of **SYS**tem **TIME**. Number of 0.1 milliseconds since midnight.

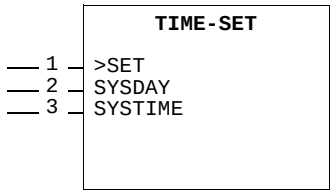
Function

The element TIME-CON converts date and time into system internal time representation.
Conversion is enabled when input terminal S is set to "1" or when input terminal L goes from "0" to "1".

Time Setting TIME-SET

Summary

TIME-Set is used for setting of date and time in the internal system time representation.



Terminal Description

Number	Name	Type	Description
1	SET	IB	Sets date and time when **SET** goes from "0" to "1".
2	SYSDAY	IL	Internal representation of **SYS**tem **DAY**. Day number since 1980-01-01.
3	SYSTIME	IL	Internal representation of **SYS**tem **TIME**. Number of 0.1 milliseconds since midnight.

Function

The element TIME-SET sets date and time in the system internal time representation.
Conversion is enabled when input terminal SET goes from "0" to "1".

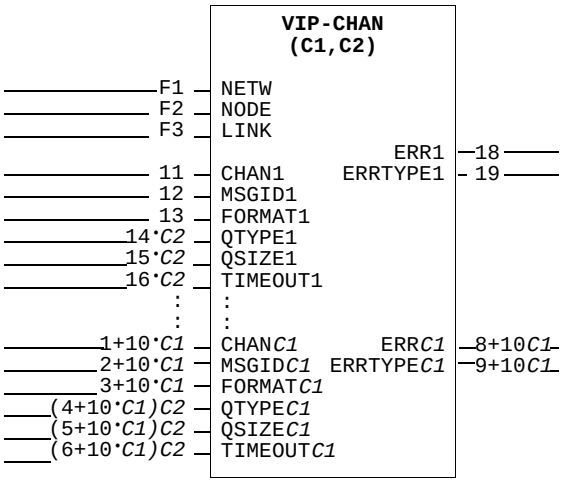
Vendor Internet Protocol-Channel VIP-CHAN

Summary

VIP-CHAN is used for configuration and supervision of channels for a VIP,Vendor Internet Protocol link.

(Inputs 14, 15 and 16 only exist for C2=1;
i.e. receive channels.)

Call VIP-CHAN(C1,C2)



Call Parameters

Parameter	Significance	Permissible values
C1	Number of channels to configure	1.....9
C2	Channel type	0=Send. 1=Receive

Terminal Description

Number	Name	Type	Description
F1	NETW	FI	MasterNet **NETW**ork configured by means of PC element VIP-NETW.
F2	NODE	FI	MasterNet **NODE** configured by means of PC element VIP-NODE.
F3	LINK	FI	**LINK** configured by means of PC element VIP-LINK.
10·ξ + 1	CHANξ	II	**CHAN**nel identity. Must be unique within the whole system configuration. ‡
10·ξ + 2	MSGIDξ	IL	**M**e**S**sa**G**e **ID**entity included in the message to distinguish between messages. ‡
10·ξ + 3	FORMATξ	II	Message **FORMAT** used for the channel.
(10·ξ + 4)*C2*	QTYPEξ	IB	**Q**ueue **TYPE** for receiving channel. Set to "1" if overwrite mode is used for the queue.
(10·ξ + 5)*C2*	QSIZEξ	II	**Q**ueue **SIZE** = maximum number of messages that can be queued when receiving. ‡
(10·ξ + 6)*C2*	TIMEOUTξ	IL	**TIME**-**OUT** time used for supervision for cyclically received messages.
10·ξ + 8	ERRξ	OB	**ERR**or on channel ξ
10·ξ + 9	ERRTYPEξ	OI	**ERR**or **TYPE** on channel ξ

ξ is a variable which varies from 1 up to *C1*.
Terminals 10·ξ + 4, 10·ξ + 5 and 10·ξ + 6 only exist for receiving channels; i.e. when C2=1.

‡ Admitted range for CHANξ: 1....65535; for MSGIDξ: 0....65535; and forQSIZEξ: 1.....10.

Function

The element is used for configuration of message channels used for exchanging application messages with an external computer associated with a MasterNet network, a node number and a link identity.

Input terminal CHAN defines the identity of the local channel that is used for the connection of the PC-element VIP-W or VIP-R to the channel.

MSGID, Message identity
Input terminal MSGID defines the message identity that must be well defined and known by both end parts of the link. The message identity is included in the header part of the message transferred on the link and is used to separate between messages on the same link.

FORMAT, Message format
Input terminal FORMAT defines the message format used for the channel.

The message format is specified as a code consisting of three digits *XYZ*.

The least significant digit *Z* determines the format used for byte array data usually used for character data.
The second least digit *Y* determines the format for group data in a message.
The most significant digit *X* defines the format for simple data such as integer data and real data.

The format selected for byte array data and group data affect the date representation in the actual message sent on the link. The selected format for simple data does not affect the data representation.

Data type	Format	Description
Int16 Int32 Real	**0***YZ*	PC element VIP-R indicates *illegal message length* if the message does not contain enough with data to update all Int16, Int32 and Real outputs.
	1*YZ*	PC element VIP-R will not indicate error when the message is too short. Outputs are set to 0 if there is no corresponding data in the received message.
Group data	*X***0***Z*	A 2-byte header preceeds the data for each group data array. The header contains number of elements in the array. The number of elements in the array sent by PC element VIP-W is determined by call parameter C5 or by the corresponding element in the GSIZE input if C9 is set to 1. PC element VIP-R sets remaining elements in the group data outputs to 0 if there are no corresponding data in the message. If C9 is set to 1 the length of the group data array is indicated in the GSIZE output.
	*X***1***Z*	No header is used for the group data. The number of elements in the vector sent by PC element VIP-W is determined by call parameter C5 or by the corresponding element in the GSIZE input if C9 is set to 1. PC element VIP-R reads C5 number of vector elements from the message for each group data vector. If the message is too short the remaining elements in the group data outputs are set to 0. If C9 is set to 1 the length of the group data vectors is indicated in the GSIZE output.
Byte array data	*XY***0**	A 2-byte header preceeds the data for each byte array. The header contains the number of bytes/characters in the array. The number of bytes in the array sent by PC element VIP-W is determined by call parameter C7 or by the corresponding element in the ASIZE input if C9 is set to 1. PC element VIP-R sets remaining elements in the byte array outputs to ASCII blanks if there are no corresponding data in the message. If C9 is set to 1 the length of the byte array is indicated in the ASIZE output.

Data type	Format	Description
	*XY***1**	No header is used for the byte arrays. The number of bytes/characters in the array sent by PC element VIP-W is determined by call parameter C7 or by the corresponding element in the ASIZE input if C9 is set to 1. PC element VIP-R reads C7 number of bytes from the message for each byte array. If the message is too short the remaining bytes in the byte array outputs are set according to the ASCII input. If C9 is set to 1 the length of the byte arrays is indicated in the ASIZE output.
	*XY***2**	Byte arrays are treated as null terminated strings with variable length. PC element VIP-W removes trailing space characters and puts a null character at the end of the string. PC element VIP-R reads max C7 number of characters from the message for each byte array. If the string is shorter than C7 the remaining characters in the byte array outputs are set according to the ASCII input. If C9 is set to 1 the length of the byte arrays is indicated in the ASIZE output.
	*XY***3**	Byte arrays are treated as null terminated strings with fixed length. PC element VIP-W send C7 number of characters ended with a null character. PC element VIP-R reads C7 number of bytes from the message for each byte array. *Illegal message length* is indicated if the string is shorter than C7 characters.

QTYPE, Queue type and QSIZE, Queue size
Input terminal QSIZE determines the number of messages that can be queued in the receiving channel.
Input terminal QTYPE defines the the type of the queue used.

If type "0" is selected the VIP-CHAN element will indicate *Queue Overflow* if messages have been dropped due to a full queue.

If type "1" is selected the oldest (first) message is cancelled if the queue is full and a new message is inserted in its place.
No overflow indication is generated in this case (for type 1).

Queue type "0" is usually used for event driven messages. To avoid losing messages due to queue overflow
both the sender's cycle time and the receiving VIP-R element's cycletime should be considered when selecting queue size.

Queue type "1" is normally used for cyclic messages containing HMC,Human Machine Communication data etc.
For type "1" the queue size is normally dimensioned to only one position in order always to get only the latest message.

TIMEOUT, Time-out supervision
Input terminal TIMEOUT determines the maximum time to wait for a message before a time-out indication is generated.
The alarm is given by the VIP-R element connected to the channel.
Inside the VIP-CHAN element this TIMEOUT value is set to the nearest higher multiple of CYCTIME of PC element VIP-R.

If TIMEOUT is set to zero the supervision is disabled.

Re-configuration

After a configuration has been accepted, i.e. corresponding ERR output terminal is set to zero,
no parameter change attempts will succeed until next system initialization.

Error handling

The output terminal ERR is set when an error is detected.
For linked elements any detected error is signalized to succeeding chain elements.
In linked chains the ERR output of the last element must therefore be studied .

Output terminal ERRTYPE gives the error code:

Code Meaning

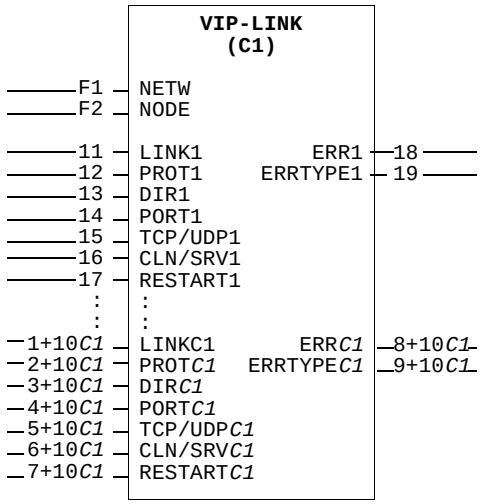
101 Unknown network. Board removed or not configured. Check PC element VIP-NETW and database element CI532.
102 Netword configuration in progress.
103 Network connection broken.
104 Network parameter error.
105 Network insufficient resources.
106 Network already configured.
201 Unknown node. Check PC element VIP-NODE.
202 Node configuration in progress.
203 Node connection broken.
204 Nope parameter error.
205 Node insufficient resources.
206 Node already configured.
301 Unknown link. Check PC element VIP-LINK.
302 Link configuration in progress.
303 Link connection broken.
304 Link parameter error.
305 Link insufficient resources.
306 Link already configured.
307 Port-protocol conflict. It's not allowed to use different procotols on the same port.
308 Illegal protocol, protocol not implemented.
401 Unknown channel. Check PC element VIP-CHAN.
402 Channel configuration in progress.
403 Channel connection broken.
404 Channel parameter error.
405 Channel insufficient resources.
406 Channel already configured.
407 Invalid use of FIRST or LAST terminal inputs. Check elements VIP-W and VIP-R.
408 Channel receive queue is empty.
409 Channel queue overflow.
410 Configuration error. End ef message detected before all outputs of VIP-R are updated.
411 Received message is deleted due to insufficient buffers for the channel queue.
412 Transmit overflow. Sent message is discarded.
413 Receive overflow. Received message is dumped.
414 Receive overflow. Received message is scrapped.
415 Transmit overflow. Sent message is aborted.
416 Insufficient memory space for allocation of channel pool.
417 End of channels.
418 Channel lock-up table full.
419 Receive channel time-out.
420 Illegal message length.
421 Illegal buffer length.
422 Illegal direction.

Vendor Internet Protocol-Link VIP-LINK

Summary

VIP-LINK is used for configuration and supervision of TCP* and UDP* links to an external computer.

*/(Transmission Control Protocol, User Datagram Protocol)



Call VIP-LINK(C1)

Call Parameter

Parameter	Significance	Permissible values
C1	Number of links to configure	1.....5

Terminal Description

Number	Name	Type	Description
F1	NETW	FI	MasterNet **NETW**ork configured by means of PC element VIP-NETW.
F2	NODE	FI	MasterNet **NODE** configured by means of PC element VIP-NODE.
10·ξ + 1	LINKξ	II	**LINK** identity. Must be unique for the node. Set within 1.....32767.
10·ξ + 2	PROTξ	II	Specifies the vendor **PROT**ocol used for the link. ‡
10·ξ + 3	DIRξ	IB	Link **DIR**ection. 1 = Send. 0 = Receive. 2 = Bidirectional.
10·ξ + 4	PORTξ	IL	**PORT** number (service). Set within 5001.....65535.
10·ξ + 5	TCP/UDPξ	IB	Link transport protocol. **TCP**=0. **UDP**=1.
10·ξ + 6	CLN/SRVξ	II	**CL**ie**N**t/**S**e**RV**er. Only for TCP links. Set to 1 when used as a connection server.
10·ξ + 7	RESTARTξ	IB	**RESTART**ing of TCP link LINKξ.
10·ξ + 8	ERRξ	OB	**ERR**or on link ξ
10·ξ + 9	ERRTYPEξ	OI	**ERR**or **TYPE** on link ξ

ξ is a variable which varies from 1 up to *C1*.

‡ Protocol numbers from 0 up to 19 are reserved for RMC standard products.

Function

The element is used for configuration of links to an external computer associated with a MasterNet network and a node number.

Protocol
Input terminal PROT specifies the vendor protocol to be used for the link.
Protocol numbers 0 to 19 are reserved for standard AC450 RMC protocols.

Protocol description is given in "VIP User's guide".

Direction
Input terminal DIR specifies the link direction that can be defined as "sending", "receiving" or "bidirectional".

Port (Service)
Each link is associated with a port (5001.....65535) that must be well defined and known by both ends of the link.
Several links can be associated with the same port (service) provided that the same protocol is used for all the links.

Transport protocol
Input terminal TCP/UDP defines the link transport protocol.
TCP, Transmission Control Protocol links are reliable with confirmation of transmitted data.
TCP links are normally used for event driven data.
UDP, User Datagram Protocol links are unconfirmed and shall normally be used only for cyclic messages.

Client/Server
When establishing TCP links one part acts as connection server for a well defined port and the other part acts as client.
A connection server listen to connection requests from clients.
When a request is accepted by the server the actual application message transfer can begin.
A client is normally seen as the sending part who wants the server to make some service on the data;
e.g. displaying, controlling etc.

Input terminal CLN/SRV is set to one if this end of the TCP link acts as connection server.
The link direction is independant of whether this end is server or not, e.g. the link can be used both for sending and/or for receiving.

Re-configuration

After a configuration has been accepted, i.e. corresponding ERR output terminal is set to zero,
no parameter change attempts will succeed until next restart of the network interface.

Error handling

The output terminal ERR is set when an error is detected.

Output terminal ERRTYPE gives the error code:

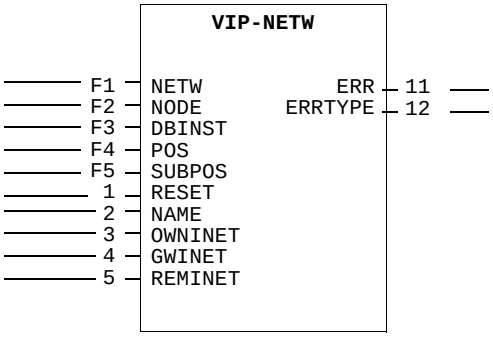
Code	Meaning
101	Unknown network. Board removed or not configured. Check PC element VIP-NETW and database element CI532.
102	Network configuration in progress.
103	Network connection broken.
104	Network parameter error.
105	Network insufficient resources.
106	Network already configured.
201	Unknown node. Check PC element VIP-NODE.
202	Node configuration in progress.
203	Node connection broken.
204	Nope parameter error.
205	Node insufficient resources.
206	Node already configured.
301	Unknown link. Check PC element VIP-LINK.
302	Link configuration in progress.
303	Link connection broken.
304	Link parameter error.
305	Link insufficient resources.
306	Link already configured.
307	Port-protocol conflict. It's not allowed to use different procotols on the same port.
308	Illegal protocol, protocol not implemented.

Vendor Internet Protocol-Network VIP-NETW

Summary

VIP-NETW is used to create a configuration and supervision of a VIP, Vendor Internet Protocol network in order to communicate on a TCP/UDP* link to an external computer.

*/(Transmission Control Protocol/ User Datagram Protocol)



Call VIP-NETW

Terminal Description

Number	Name	Type	Description
F1	NETW	FI	Master **NETW**ork number (1....99) to be assiciated with network interface board.
F2	NODE	FI	MasterNet **NODE** number of own node.
F3	DBINST	FI	**D**ata**B**ase **INST**ance. Logical record number of CI532 (MVI) database element.
F4	POS	FI	TCP/IP processor carrier board **POS**ition.
F5	SUBPOS	FI	TCP/IP processor board **SUBPOS**ition on the carrier board. SUBPOS is to be set to zero If PM510 is used.
1	RESET	IB	Input for **RESET**ing of the TCP/IP processor board.
2	NAME	IA20	Own host **NAME**.
3	OWNINET	IA20	**OWN I**nter**NET** address of network interface 1.
4	GWINET	IA20	**G**ate**W**ay **I**nter**NET** address. Set to empty string if not used.
5	REMINET	IA20	**REM**ote network of gateway **I**nter**NET** address. Set to empty string if not used.
11	ERR	OB	**ERR**or
12	ERRTYPE	OI	**ERR**or **TYPE**

Function

The element associates a MasterNet network number with the TCP/IP network interface.
The internet address of the network interface is specified on the OWNINET input terminal.
If a gateway is used for reaching any of the external computers the Internet address of the gateway is specified on input terminal GWINET and the remote network on input terminal REMINET.

Input terminal DBINST specifies the instance number of the corresponding CI532 database element.

Note:
Please observe that the property terminal IMPLEMENTED of the corresponding CI532 database element must be set to zero to allow for the VIP-NETW element to initialize the CI532 database element.

Re-configuration

After a configuration has been accepted, i.e. corresponding ERR output terminal is set to zero, no parameter change attempts will succeed until next restart of the network interface.

Error handling

The output terminal ERR is set when an error is detected.

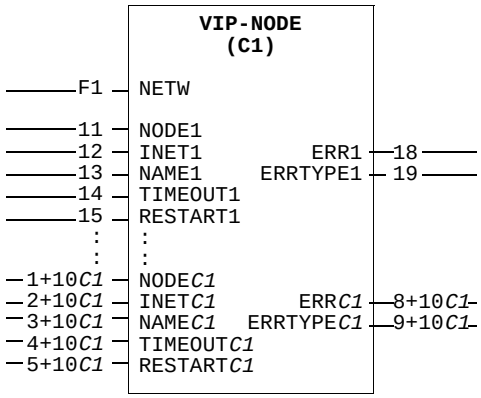
Output terminal ERRTYPE gives the error code:

Code	Meaning
101	Unknown network. Board removed or not configured. Check PC element VIP-NETW and database element CI532.
102	Network configuration in progress.
103	Network connection broken.
104	Network parameter error.
105	Network insufficient resources.
106	Network already configured.

Vendor Internet Protocol-Node VIP-NODE

Summary

VIP-NODE is used for configuration and supervision of nodes connected to a Vendor Internet Protocol network.



Call VIP-NODE(C1)

Call Parameter

Parameter	Significance	Permissible values
C1	Number of nodes to configure	1.....9

Terminal Description

Number	Name	Type	Description
F1	NETW	FI	MasterNet **NETW**ork configured by means of PC element VIP-NETW.
10·ξ + 1	NODEξ	II	MasterNet **NODE** number (1....99) to be associated with external computer ξ.
10·ξ + 2	INETξ	IA20	**I**nter**NET** address of external computer number ξ.
10·ξ + 3	NAMEξ	IA20	Host **NAME** of the external computer number ξ. Set to empty string if not used.
10·ξ + 4	TIMEOUTξ	IT	**TIME**-**OUT** time for supervision of external computer ξ. Set to zero if not used.
10·ξ + 5	RESTARTξ	IB	Input for **RESTART**ing of all TCP links for NODEξ.
10·ξ + 8	ERRξ	OB	**ERR**or on node ξ.
10·ξ + 9	ERRTYPEξ	OI	**ERR**or **TYPE** on node ξ.

ξ is a variable which varies from 1 up to *C1*.

Function

The element associates a MasterNet node number with each external computer connected to the same MasterNet network.
The internet address of the external node is specified on the INET input terminal.
Optionally a host name may be specified on input terminal NAME.

Supervision
Input terminal TIMEOUT specifies the maximal time to wait for a message from the external node before closing all links to it.
The supervision is disabled by setting TIMEOUT to zero.

Re-configuration

After a configuration has been accepted, i.e. corresponding ERR output terminal is set to zero,
no parameter change attempts will succeed until next restart of the network interface.

Error handling

The output terminal ERR is set when an error is detected.

Output terminal ERRTYPE gives the error code:

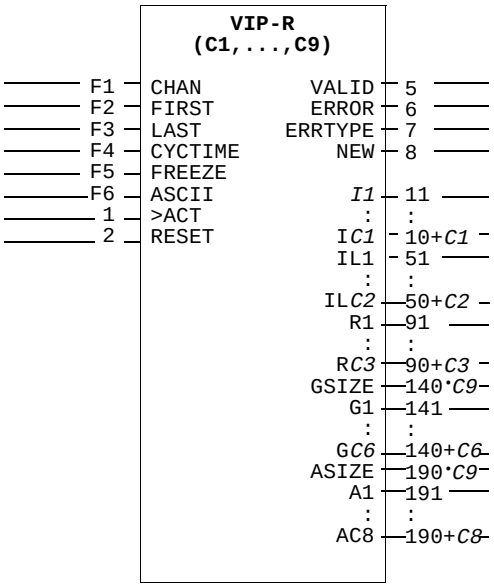
Code	Meaning
101	Unknown network. Board removed or not configured. Check PC element VIP-NETW and database element CI532.
102	Networld configuration in progress.
103	Network connection broken.
104	Network parameter error.
105	Network insufficient resources.
106	Network already configured.
201	Unknown node. Check PC element VIP-NODE.
202	Node configuration in progress.
203	Node connection broken.
204	Nope parameter error.
205	Node insufficient resources.
206	Node already configured.

Vendor Internet Protocol-Read VIP-R

Summary

VIP-R provides the application programme with an interface for receiving data on a TCP/UDP* link from an external computer.

*/(Transmission Control Protocol/ User Datagram Protocol)



Call VIP-R(C1,C2,C3,C4,C5,C6,C7,C8,C9)

Call Parameters

Parameter	Significance	Permissible values	Default value
C1	Number of integer data outputs	0.....40	0
C2	Number of integer long data outputs	0.....40	0
C3	Number of real data outputs	0.....40	0
C4	Group data type	I, IL, R	-
C5	Number of elements in group data outputs	0.....255	0
C6	Number of group data outputs	0.....40	0
C7	Max size of byte array data outputs	0.....255	0
C8	Number of byte array data outputs	0.....40	0
C9	Selection of group data outputs for obtaining number of elements included in the message of the group data outputs G1...G*C6* and byte array data outputs A1..A*C8*.	0.....1	0

Terminal Description

Number	Name	Type	Description
F1	CHAN	FIL	**CHAN**nel Identity. 1....65535.
F2	FIRST	FB	This is the **FIRST** part of a message
F3	LAST	FB	This is the **LAST** part of a message
F4	CYCTIME	FIT	**CYC**le **TIME** interval for cyclic messages. CYCLIC=0 disables cyclic sending.
F5	FREEZE	FB	**FREEZE** data outputs when error. FREEZE=0 will clear outputs for all errors.
F6	ASCII	FB	Fill byte array outputs with **ASCII** blanks at reset.
1	ACT	IB	**ACT**ivation of receiving at positive edge.
2	RESET	IB	**RESET** data outputs
5	VALID	OB	**VALID** receiving of data. Set to "1" for successful reception.
6	ERR	OB	**ERR**or
7	ERRTYPE	OI	**ERR**or **TYPE**
8	NEW	OB	Output is set during one cycle when a **new** message is available on data outputs.
11.....10+*C1*	I1.....I*C1*	OI	**I**nteger data outputs
51.....60+*C2*	IL1....IL*C2*	OIL	**I**nteger **L**ong data outputs
91.....90+*C3*	R1....R*C3*	OR	**R**eal data outputs
140·*C9*	GSIZE	OGI*C6*	Optional **G**roup data **SIZE** output for obtaining number of elements included in the message of the group data outputs G1....G*C6*.
141.....140+*C6*	G1....G*C6*	OG*C4C5*	**G**roup data outputs
190·*C9*	ASIZE	OGI*C8*	Optional **A**rray group data **SIZE** output for obtaining number of elements included in the message of the byte array outputs A1....A*C8*.
191.....190+*C8*	A1....A*C8*	OIA*C7*	Byte **A**rray data outputs

Function

Source address

The source address for the application message is implicitly defined by the channel identity CHAN.
The channel is associated with a link to a remote computer by means of the PC elements VIP-NODE, VIP-LINK and VIP-CHAN.

Element linking

Several elements can be linked together to form one single message.
The order of the data values in the message is determined by the execution order.
Input FIRST is set to one for the first element in the chain
and input terminal LAST is set to one for the last element in the chain.

Execution control

Message sending can be periodic and/or event driven. (I.e. a message can be both cyclic and event driven at the same time.) CYCTIME specifies the time interval used for periodic reading. Inside the PC element this value is set to the nearest higher multiple of CYCTIME defined for the first element in the chain (the header element). CYCTIME=0 disables cyclic sending. For linked elements the sending interval is decided by the header element.

The ACT input is used for event controlled reading. Reading is activated at positive edge when ACT changes 0-->1 i.e. from zero to one. This input of a linked chain controls the whole chain.

Error handling

The output terminal ERR is set when an error is detected.
For linked elements any detected error is signalized to succeeding chain elements.
In linked chains the ERR output of the last element must therefore be studied .

Output terminal ERRTYPE gives the error code:

Code Meaning

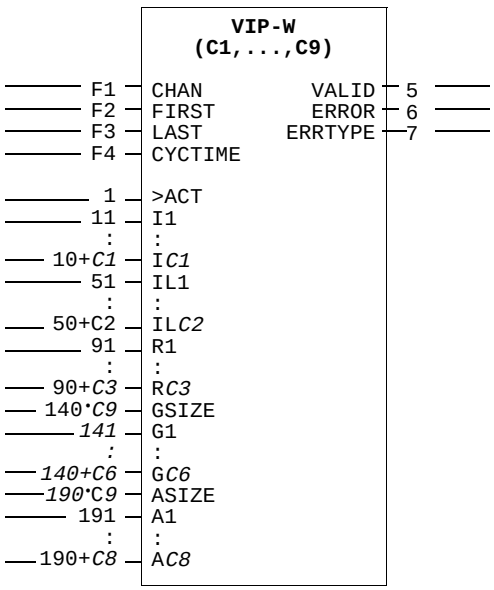
101 Unknown network. Board removed or not configured. Check PC element VIP-NETW and database element CI532.
102 Netword configuration in progress.
103 Network connection broken.
104 Network parameter error.
105 Network insufficient resources.
106 Network already configured.
201 Unknown node. Check PC element VIP-NODE.
202 Node configuration in progress.
203 Node connection broken.
204 Nope parameter error.
205 Node insufficient resources.
206 Node already configured.
301 Unknown link. Check PC element VIP-LINK.
302 Link configuration in progress.
303 Link connection broken.
304 Link parameter error.
305 Link insufficient resources.
306 Link already configured.
307 Port-protocol conflict. It's not allowed to use different procotols on the same port.
308 Illegal protocol, protocol not implemented.
401 Unknown channel. Check PC element VIP-CHAN.
402 Channel configuration in progress.
403 Channel connection broken.
404 Channel parameter error.
405 Channel insufficient resources.
406 Channel already configured.
407 Invalid use of FIRST or LAST terminal inputs. Check elements VIP-W and VIP-R.
408 Channel receive queue is empty.
409 Channel queue overflow.
410 Configuration error. End ef message detected before all outputs of VIP-R are updated.
411 Received message is deleted due to insufficient buffers for the channel queue.
412 Transmit overflow. Sent message is discarded.
413 Receive overflow. Received message is dumped.
414 Receive overflow. Received message is scrapped.
415 Transmit overflow. Sent message is aborted.
416 Insufficient memory space for allocation of channel pool.
417 End of channels.
418 Channel lock-up table full.
419 Receive channel time-out.
420 Illegal message length.
421 Illegal buffer length.
422 Illegal direction.

Vendor Internet Protocol-Write VIP-W

Summary

VIP-W provides the application programme with an interface for sending data on a TCP/UDP* link to an external computer.

*/(Transmission Control Protocol/ User Datagram Protocol)



Call VIP-W(C1,C2,C3,C4,C5,C6,C7,C8,C9)

Call Parameters

Parameter	Significance	Permissible values	Default value
C1	Number of integer data inputs	0.....40	0
C2	Number of integer long data inputs	0.....40	0
C3	Number of real data inputs	0.....40	0
C4	Group data type	I, IL, R	-
C5	Number of elements in group data inputs	0.....255	0
C6	Number of group data inputs	0.....40	0
C7	Max size of byte array data inputs	0.....255	0
C8	Number of byte array data inputs	0.....40	0
C9	Selection of group data inputs for controlling number of elements to be included in the message of the group data inputs G1...G*C6* and byte array data inputs A1..A*C8*.	0.....1	0

Terminal Description

Number	Name	Type	Description
F1	CHAN	FIL	**CHAN**nel Identity
F2	FIRST	FB	This is the **FIRST** part of a message
F3	LAST	FB	This is the **LAST** part of a message
F4	CYCTIME	FIT	**CYC**le **TIME** interval for cyclic messages. CYCLIC=0 disables cyclic sending.
1	ACT	IB	**ACT**ivation of sending (at positive edge)
5	VALID	OB	**VALID** sending of data. Set to "1" for successful sending.
6	ERR	OB	**ERR**or
7	ERRTYPE	OI	**ERR**or **TYPE**
11.....10+*C1*	I1.....I*C1*	II	**I**nteger data inputs
51.....60+*C2*	IL1....IL*C2*	IIL	**I**nteger **L**ong data inputs
91.....90+*C3*	R1....R*C3*	IR	**R**eal data inputs
140·*C9*	GSIZE	IGI*C6*	Optional **G**roup data **SIZE** input for controlling number of elements to be included in the message of the group data inputs G1....G*C6*.
141.....140+*C6*	G1....G*C6*	IG*C4C5*	**G**roup data inputs
190·*C9*	ASIZE	IGI*C8*	Optional **A**rray group data **SIZE** input for controlling number of elements to be included in the message of the byte array inputs A1....A*C8*.
191.....190+*C8*	A1....A*C8*	IA*C7*	Byte **A**rray data inputs

Function

Destination

The destination address for the application message is implicitly defined by the channel identity CHAN.
The channel is associated with a link to a remote computer by means of the PC elements VIP-NODE, VIP-LINK and VIP-CHAN.

Element linking

Several elements can be linked together to form one single message.
The order of the data values in the message is determined by the execution order.
Input FIRST is set to one for the first element in the chain
and input terminal LAST is set to one for the last element in the chain.

Execution control

Message sending can be periodic and/or event driven. (I.e. a message can be both cyclic and event driven at the same time.) CYCTIME specifies the time interval used for periodic sending. Inside the PC element this value is set to the nearest higher multiple of CYCTIME defined for the first element in the chain (the header element). CYCTIME=0 disables cyclic sending. For linked elements the sending interval is decided by the header element.

The ACT input is used for event controlled sending. Sending is activated at positive edge when ACT changes 0-->1 i.e. from zero to one. This input to a header element of a linked chain controls the whole chain.

Error handling

The output terminal ERR is set when an error is detected.
For linked elements any detected error is signalized to succeeding chain elements.
In linked chains the ERR output of the last element must therefore be studied .

Output terminal ERRTYPE gives the error code:

Code Meaning

101 Unknown network. Board removed or not configured. Check PC element VIP-NETW and database element CI532.
102 Networld configuration in progress.
103 Network connection broken.
104 Network parameter error.
105 Network insufficient resources.
106 Network already configured.
201 Unknown node. Check PC element VIP-NODE.
202 Node configuration in progress.
203 Node connection broken.
204 Nope parameter error.
205 Node insufficient resources.
206 Node already configured.
301 Unknown link. Check PC element VIP-LINK.
302 Link configuration in progress.
303 Link connection broken.
304 Link parameter error.
305 Link insufficient resources.
306 Link already configured.
307 Port-protocol conflict. It's not allowed to use different procotols on the same port.
308 Illegal protocol, protocol not implemented.
401 Unknown channel. Check PC element VIP-CHAN.
402 Channel configuration in progress.
403 Channel connection broken.
404 Channel parameter error.
405 Channel insufficient resources.
406 Channel already configured.
407 Invalid use of FIRST or LAST terminal inputs. Check elements VIP-W and VIP-R.
408 Channel receive queue is empty.
409 Channel queue overflow.
410 Configuration error. End ef message detected before all outputs of VIP-R are updated.
411 Received message is deleted due to insufficient buffers for the channel queue.
412 Transmit overflow. Sent message is discarded.
413 Receive overflow. Received message is dumped.
414 Receive overflow. Received message is scrapped.
415 Transmit overflow. Sent message is aborted.
416 Insufficient memory space for allocation of channel pool.
417 End of channels.
418 Channel lock-up table full.
419 Receive channel time-out.
420 Illegal message length.
421 Illegal buffer length.
422 Illegal direction.

Appendix A Protocols And Message Formats

A.1 General

This chapter describes the protocols and message formats available for the VIP function.

A VIP message consists of a header part and a data part as shown in Figure A-1. The protocol defines the message header and the format specifies representation of data in the data part.

The packet format is the same for both TCP and UPD data transfer.

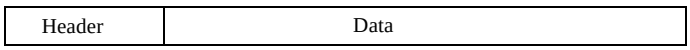


Figure A-1. VIP Message

A.2 Data Transmit Ordering

Multi byte data is sent/received with the most significant byte first according to Figure A-2. The external computer shall send in the same format to the Advant Controller system.

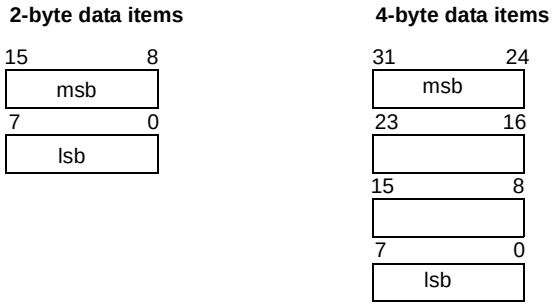


Figure A-2. Multi Byte Order

A.3 Structure of Header Part, Protocol 0

The message length field contains the total length of the message in bytes including the header. The message identity field is used as an identifier for the message in order to distinguish between messages on the same link

The total length of a message is limited to:

- 65535 bytes for TCP messages.
- 9500 bytes for UDP messages.

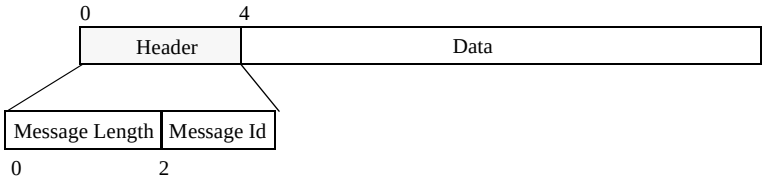


Figure A-3. Message Header, Protocol 0

A.4 Structure of Data Part

The data part of a VIP message contains an arbitrary number of data blocks. Each data block corresponds to one PC element VIP-W or VIP-R. The order of the data types within each data block, shown below, is the same order as the terminal are grouped on the PC elements VIP-W and VIP-R.

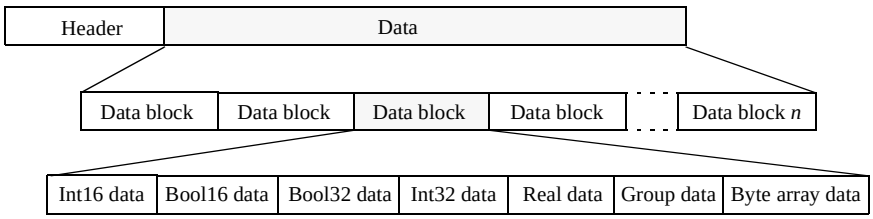


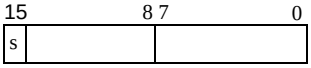
Figure A-4. Data Part

A data block can contain up to 40 data items of each data type. The total number of data items within a data block must however not exceed 45 items.

A.5 Message Formats of Data Part

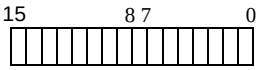
This section describes the available message formats and their representation. Refer PC element VIP-CHAN for details regarding format handling done by PC element VIP-W and VIP-R.

A.5.1 16-bit Integer data



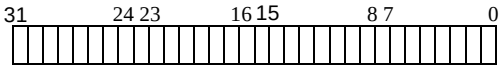
Size: 2 byte
Format: Signed 2-complement
Range: -32768 to 32767

A.5.2 16-bit Packed Boolean data



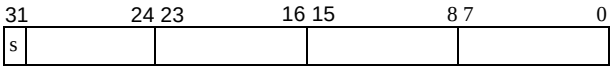
Size: 2 byte
True: bit value = 1
False: bit value = 0

A.5.3 32-bit Packed Boolean data



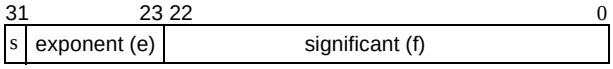
Size: 4 byte
True: bit value = 1
False: bit value = 0

A.5.4 32-bit Integer data



Size: 4 byte
Format: Signed 2-complement
Range: -2 147 438 648 to 2 147 438 647

A.5.5 Real data



Size:		4 byte
Interpretation of sign:		
	positive:	0
	negative:	1
Normalized numbers:		
	Interpretation of normalized e:	unsigned integer
	bias of e:	127
	range of e:	0 < e < 255
	interpretation of significant:	1.f
	relation to representation of real number:	$(-1)^{sign} * 2^{(exponent-127)} * 1.significand$
	range of significant:	1.0 <= significant < 2.0 with the integer part implicit
zeroes:		
	e =	0
	f =	0

A.5.6 Group data

A.5.6.1 General

Group data is a collection of data elements (1...255) with one of the data types below. The data type of the group data elements must be of the same type for all arrays within the same data block.

- 16-bit integer data (2 bytes)
- 32-bit integer data (4 bytes)
- Real data (4 bytes)

A.5.6.2 Format 0

Each array is preceded by a 2-byte header containing number of elements in the array as shown in Figure A-5. The size of the arrays may vary dynamically but must not exceed 255 elements per group data array.

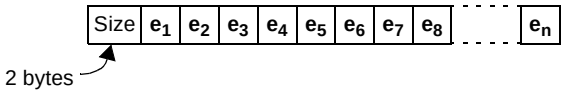


Figure A-5. Group Data Format 0

NOTE

The input GSIZE on PC element VIP-W is used to control the size of the group data arrays and output GSIZE on PC element VIP-R is used to obtain number of elements received for a group data array.

A.5.6.3 Format 1

A group data with this format can also be seen as a number of single elements, e.g. a group data array with 10 32-bit integer values can also be seen as 10 single 32-bit integer values.

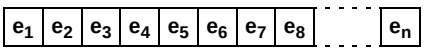


Figure A-6. Group Data Format 1

NOTE

The input GSIZE on PC element VIP-W is used to control the size of the group data arrays and output GSIZE on PC element VIP-R is used to obtain number of elements received in a group data array.

A.5.7 Byte Array data

A.5.7.1 General

Byte Array data is a collection of bytes, usually characters.

Representation of characters:

Size:	1 byte
Format:	7-bit ASCII
Range:	0 to 127

A.5.7.2 Example

Figure A-7 below shows an example with one byte array input terminal, A1, with a maximal size of 10 bytes. The representation of input string 'ABCD' in the message is shown below for each format.

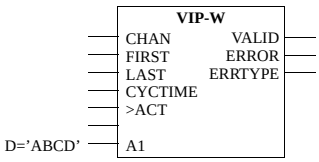


Figure A-7. Byte Array Example

A.5.7.3 Format 0

Each byte array is preceded by a 2-byte header containing number of elements in the array as shown in Figure A-8. The size of the array may vary dynamically but must not exceed 255 bytes per array. The array is padded with white spaces.

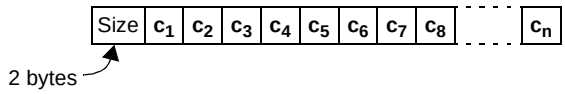


Figure A-8. Byte Array Format 0

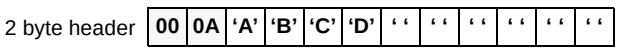


Figure A-9. Example, Byte Array Format 0

A.5.7.4 Format 1

The size should not exceed 255 bytes if it should be treated as a separate string. The array is padded with white spaces.

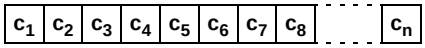


Figure A-10. Byte Array Format 1

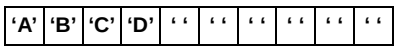


Figure A-11. Example, Byte Array Format 1

A.5.7.5 Format 2

The byte array is used for character data and is terminated with a null character. The size of the array may vary dynamically but should not exceed 255 bytes. No padding is done.

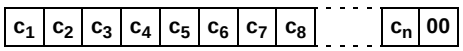


Figure A-12. Byte Array Format 2

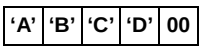


Figure A-13. Example, Byte Array Format 2

A.5.7.6 Format 3

The byte array is used for character data and is terminated with a null character. The size of the array is fixed and should not exceed 255 bytes. The array is padded with white spaces.

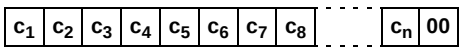


Figure A-14. Byte Array Format 3

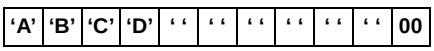


Figure A-15. Example, Byte Array Format 3

Appendix B Internet Concepts

B.1 Overview

Networking protocols are normally developed in layers, with each layer responsible for a different facet of the communication. TCP/IP is normally considered to be a 4-layer system, as shown in the figure below.

Application	Telnet, FTP, etc.
Transport	TCP, UDP
Network	IP
Link	Device driver and interface card

The Internet is a layered protocol model, where the higher layers provides the most complex functions

B.2 Internet Protocol (IP)

IP is the base network protocol of the Internet protocol family. With IP, each host (computer) in the network has a unique four-byte Internet address. IP accepts packets addressed to a particular host and tries to deliver them. If multiple networks are connected by gateways, IP will forward a package from gateway to gateway until the packet reaches a network where it can be delivered directly. IP will also break up and reassemble packets to fit the packet size of the physical network. However, IP makes no guarantees that packets will be delivered to the destination correctly. Although is possible to access IP directly, most applications will use one of the higher-level protocols such as UDP or TCP.

B.3 User Datagram Protocol (UDP)

UDP provides a simple *datagram*-based process-to-process communication mechanism. UDP extends the message address to include a *port address* in addition to the host Internet address, where a port address identifies one of several distinct destinations within a single host. Thus UDP accepts messages addressed to a particular port on a particular host, and tries to deliver them, using IP to transport the messages between the hosts. Like IP, UDP makes no guarantees that messages will be delivered correctly or even delivered at all.

B.4 Transmission Control Protocol (TCP)

TCP provides reliable, flow-controlled, two-way, process-to-process transmission of data. TCP is a *connection*-based communication mechanism. This means that before data can be exchanged via TCP, the two communicating processes must first establish a connection via a distinct connection phase. Data is then sent and received as a byte stream at both ends. Like UDP, TCP extends the connection address to include a port address in addition to the Internet address. That is, a connection is established between a particular port in one host and a particular port in another host. TCP *guarantees* that the delivery of data will be correct, in proper order, and without duplication.

B.5 Sockets

The direct user interface to Internet protocol suite is via *sockets*. A socket is an end-point for communications which gets "bound" to a UDP or TCP port within the host.

A process can create a datagram socket (UDP) and bind it to a particular port number. Other processes, on any host in the network, can then send messages to that socket by specifying the host internet and the port number.

Similarly, a process can create a stream socket (TCP) an bind it to a particular port number. Another process, on any host in the network, can then create another stream socket and request that it be connected to the first socket by specifying its host Internet address and port number. Once the two TCP sockets have been thus connected, there is a "virtual circuit" set up between them allowing error-free socket-to-socket communication.

3BSE 014 823 R0001

December 1997