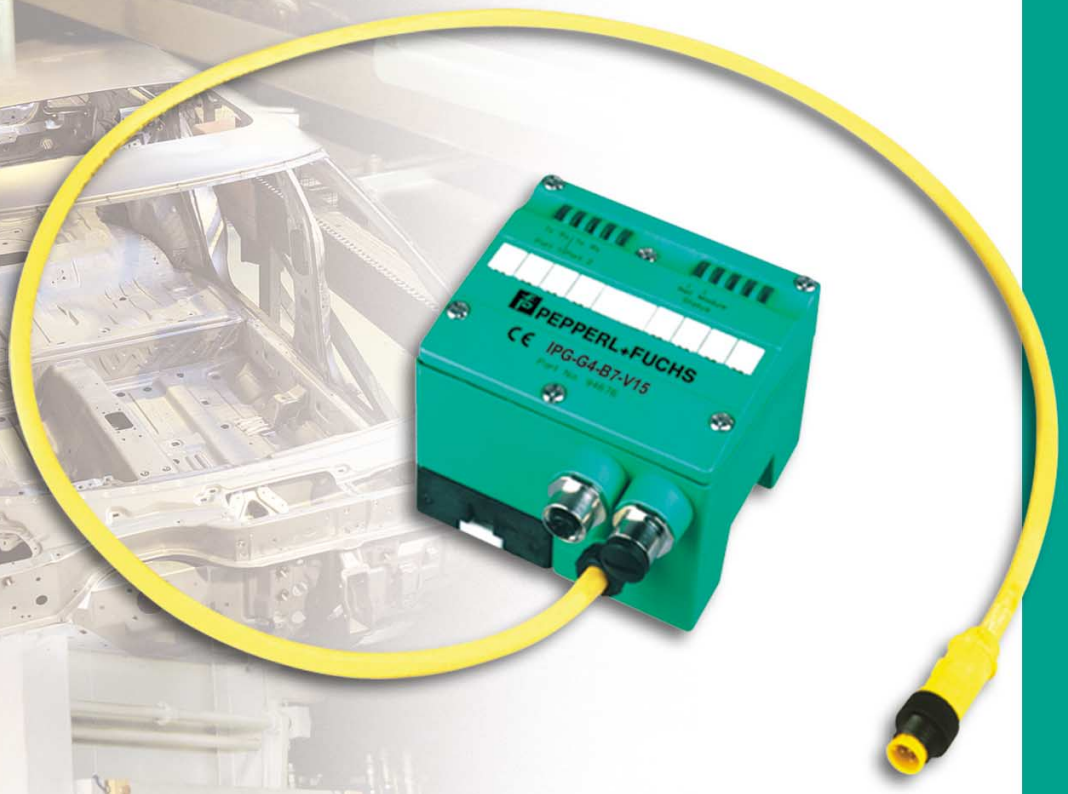
FACTORY AUTOMATION

IPG-G4-B7-V15

User Manual



PEPPERL+FUCHS

Released EDM checkout 2005-AUG-15

IPG-G4-B7-V15
Table of Contents

Issued Apr/2003

Released EDM checkout 2005-AUG-15

Notes

1 Symbols

WARNING, CAUTION, and NOTE

Warning

This symbol is a warning for danger. If it goes unheeded, there is a potential for death or injury to person and damage to equipment.

Attention

This symbol warns of a possible fault. If unheeded, the device or the system may develope a fault or fail completely.

Note

This symbol indicates important information.

2 Safety

Intended Use

The protection of operating personnel is not guaranteed if the equipment is used for a purpose for which it is not intended.

Warning

The IPG-G4-B7-V15 is to be operated only by authorized personnel in accordance with these operating instructions.

Safety Considerations

The use of this device in a manner inconsistent with these operating instructions will void any guarantee of safety for device and system function.

Warning

Connection and maintenance of this device while voltage is applied should only be performed by trained technicians.

If a fault cannot be corrected, the device should be removed from service and steps should be taken to ensure that it is not reinstalled.

Device repairs or modifications should only be performed by the manufacturer. Unauthorized operation, repair or modification of the device will void any guarantees. The operator is responsible for complying with local safety regulations.

3 Product Description

Shipment Contents

The IPG-G4-B7-V15 DeviceNet node adapter includes the following items:

- 1 meter micro Devicenet pigtail
- 1 Mounting base
- 1 M12 dust cap
- 1 Cover with six screws
- 2 Micro or M12 quick disconnects for read head connection
- 1 CD includes eds file and icon file, user manual in PDF-format

Application Suitability

The system is suited for the following applications including:

- Automation
- Material flow control
- Acquisition of operating data
- Access control
- Identification of storage vessels, pallets, parts bins, refuse bins, tanks, or containers

Device Characteristics and Operating Principles

- Bus interfaces with galvanically isolated voltage supply
- Devicenet connection using the 5-pin micro pigtail
- Display LEDs (on the cover of IPG-G4-B7-V15)

DeviceNet:	Green/red	Network
	Green/red	Module
Serial:	Green/red	send/receive port 1
	Green/red	send/receive port 2

Required Accessories

- Read heads (max 2) IPT-FP
- Read head mounting base (max 2) U-P4-R4-V15
- Read head cable (1 for each head)

5 m	V15-G-YE5M-PVC-V15-G-ABG
10 m	V15-G-YE10M-PVC-V15-G-ABG
25 m	V15-G-YE25M-PVC-V15-G-ABG

Issued Apr/2003

4 Installation

Storage and transportation

The device should be packed appropriately to avoid damage due to impact and dampness. The original packing affords the best protection. In addition, do not exceed the environmental conditions (see pages 39-41 for Technical Data).

Unpacking

Examine the contents to ensure the order is complete and undamaged. Report any damage to the shipping company.

Retain the original packaging in case the unit has to be packed for storage or shipping.

Any questions should be directed to Pepperl+Fuchs.

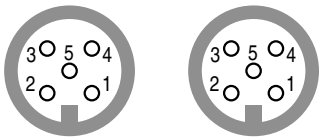
Connection

The RS 485 connection is made using cable V15-G-YExx-PVC-V15-G-ABG. See below for pin connections.

The RS485 address DIP switches on the U-P4-R4-V15 should be set to 1, which means that DIP switch 5 is on and the rest are off. No other combination is supported.

IPG-G4-B7-V15

Female Connection

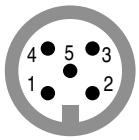


Port 1 **Port 2**

1	24V (+)
2	A
3	24V (–)/ground
4	B
5	Shield

U-P4-R4-V15

Male Receptacle



1	24V (+)
2	A
3	24V (–)/ground
4	B
5	Shield

Baud rates

Baud rates of 125 kbits, 250 kbits and 500 kbits per second are supported and are settable by the "baud" rotary switch.

- 0 = 125 kbits
- 1 = 250 kbits
- 2 = 500 kbits
- 3-9 = Software configuration

Issued Apr/2003

5 Cable Connections

Cable Length

- RS485 connection from base to DeviceNet node adapter (1000 m)
- DeviceNet maximum cable length:

  125 kbaud = 500 m

  250 kbaud = 250 m

  500 kbaud = 100 m

Termination

RS485 connection requires termination switch to be active for long cable runs.

DeviceNet requires no termination on drop lines. Termination is, however, required at the start and end of the trunk line.

EMC Shielding

Cable shielding discharges electromagnetic interference. Both ends of the cable shielding must be connected to ground with low resistance and low inductance. Use recommended cable for connection.

When using leads with dual shielding, e.g. wire mesh and metal foil, both forms of shielding must be connected at the ends of the cable with low impedance.

Note

Power supply cables radiate a significant amount of interference, e.g. the start current of a three-phase electric motor. For this reason, avoid installing the supply cables in parallel with the data/signal; cables or within the same cable duct.

Connect mesh shield/drain wire to Pin 5 on both the read head, U-P4-R4-V15 and IPG-G4-B7-V15.

6 Response time

The response time of an expanded IDENT-I System P identification system on DeviceNet is an important operational consideration. However, the time it takes for data to reach the computer or PLC from the time a data carrier enters the commuicators read zone can be determined based on a number of parameters.

Important factors that impact response time include:

- Host type, e.g. PLC or PC
- DeviceNet node adapter baud rate
- Number of DeviceNet stations
- Code/data carrier type
- Type of access to the read/write communication objects
- Type of commands used with the communicator
- User program structure

In dealing with larger projects data transfer from the DeviceNet node adapter to the host should be tested under safe conditions to familiarize the engineer with the entire system response.

There is a whole series of configuration tools that enable the DeviceNet user to plan and operate a network, without detailed knowledge of the underlying communication processes.

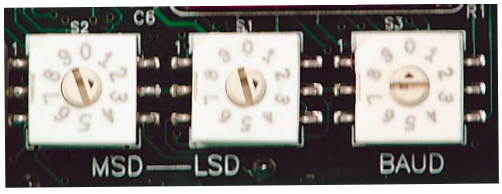
Startup

In preparation for commissioning, the device address (MAC ID) and baud rate must be set using the rotary switches, marked MSD "Most Significant Digit"and LSD "Least Significant Digit" for the MACID and BAUD for the baud rate. These are located near the lower section near to the connection terminals.

Settings

Select a node address between 0 and 63 that is not occupied by another bus station and set this using the rotary switches. Addresses above 63 make the node address software settable.

Select the baud rate on rotary switch marked BAUD; see Baud Rate on page 9.



Issued Apr/2003

7 Commands

Software

Commands consist of the command code, port number, number of words, start address and write Data. This information is entered into the "Output" data field of the master.

A response is read from the master's input data field and consists of the command code's echo, status, execution counter and read data.

A number of commands do not use all the parameters and data fields. These fields should have "0" as a value. The input and output fields are constructed as follows:

Output data field:

Byte 0	Command code
Byte 1	WordNum/APort/PNum/Tbit
Byte 2	Parameter
Byte 3	Parameter
Byte 4	Write data
Byte 5	Write data
Byte 6	Write data
Byte 7	Write data
Byte 8	Write data

Input data field:

Byte 0	Command code (echo)
Byte 1	WordNum/APort/PNum/Tbit (echo)
Byte 2	Status
Byte 3	Execution counter
Byte 4	Read data
Byte 5	Read data
Byte 6	Read data
Byte 7	Read data
Byte 8	Read data

In order to send a new command to the device, the DeviceNet user must enter data into the the output data field of the DeviceNet master. The new command is executed when the output data has changed. If the same command is to be executed a number of times, the toggle bit must be inverted so that the device recognizes that a new command must be processed.

When a new command is detected, the "Status" is set to FFh and the execution counter is set to 00h. When the command executes, the status changes from FFh to a valid status response. The execution counter increments each time a read or write completes. An execution counter increments of FFh will increment to 00h on the next successful read or write.

After processing the commands through the IDENT system, the "Status" is output in accordance with the Status/Error messages table on page 36.

The port bit will be high of low depending on which port is active. Correspondingly, the toggle bit of the response is the same as the toggle bit of the new command.

New response data will over write the old. The poll retension timer in the EDS file can be used to determine the minimum length of time the data should be retained before it can be overwritten. The basic setting for the timer value is 0 ms. This means data will be updated as soon as it is received.

The buffered and enhanced buffered commands are executed repeatedly as long as commands remain in the output data field.

Overview of commands with command code in hex and decimal representation
System commands

Command code	Command description	Abbrev.	Page
2h/2d	quit	qu	13
4h/4d	change tag	ct	14
16h/22d	reset	rs	15
3h/3d	version	ve	16

Standard read/write commands
Fixed code

Command code	Command description	Abbrev.	Page
1h/1d	Single read fixed code	sf	17
8h/8d	Auto read fixed code	af	18
9h/9d	Buffered read fixed code	bf	19
1Dh/29d	Enhanced read fixed code	ef	20

Overview of commands

Read data

Command code	Command description	Abbrev.	Page
10h/16d	single read	sr	22
20h/32d	auto read	ar	23
30h/48d	buffered read	br	24
19h/25d	enhanced buffered read	er	25

Write data

Command code	Command description	Abbrev.	Page
40h/64d	single write	sw	26
50h/80d	auto write	aw	27
60h/96d	buffered write	bw	28
1Ah/26d	enhanced buffered write	ew	29

Write fixed code IPC10

Command code	Command description	Abbrev.	Page
1Fh/31d	single write fixed code	sx	31
64h/100d	auto write fixed code	ax	32
65h/101d	buffered write fixed code	bx	33
24h/36d	enhanced write fixed code	ex	34

System Commands

quit (qu):

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (2h/2d)	0	0	0	0	0	0	1	0
Byte 1	APort/PNum/Tbit	0	0	0	0	A	P	0	T
Byte 2	unused	-	-	-	-	-	-	-	-
Byte 3	unused	-	-	-	-	-	-	-	-
Byte 4	unused	-	-	-	-	-	-	-	-
Byte 5	unused	-	-	-	-	-	-	-	-
Byte 6	unused	-	-	-	-	-	-	-	-
Byte 7	unused	-	-	-	-	-	-	-	-
Byte 8	unused	-	-	-	-	-	-	-	-

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (2h/2d)	0	0	0	0	0	0	1	0
Byte 1	APort/PNum/Tbit	0	0	0	0	A	P	0	T
Byte 2	Status	<Status>							
Byte 3	unused	-	-	-	-	-	-	-	-
Byte 4	unused	-	-	-	-	-	-	-	-
Byte 5	unused	-	-	-	-	-	-	-	-
Byte 6	unused	-	-	-	-	-	-	-	-
Byte 7	unused	-	-	-	-	-	-	-	-
Byte 8	unused	-	-	-	-	-	-	-	-

Note: All dashes (-) are assumed to 0 (zero).

System Commands

change tag (ct):

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (4h/4d)	0	0	0	0	0	1	0	0
Byte 1	Reserved/Toggle bit	0	0	0	0	0	0	0	T
Byte 2	<ID Tag type>	<Tagtype> (High byte)							
Byte 3	<ID Tag type)	<Tagtype> (Low byte)							
Byte 4	unused	-	-	-	-	-	-	-	-
Byte 5	unused	-	-	-	-	-	-	-	-
Byte 6	unused	-	-	-	-	-	-	-	-
Byte 7	unused	-	-	-	-	-	-	-	-
Byte 8	unused	-	-	-	-	-	-	-	-

Response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (4h/4d)	0	0	0	0	0	1	0	0
Byte 1	Reserved/PNum/Toggle bit	0	0	0	0	0	P	0	T
Byte 2	Status	<Status>							
Byte 3	unused	-	-	-	-	-	-	-	-
Byte 4	unused	-	-	-	-	-	-	-	-
Byte 5	unused	-	-	-	-	-	-	-	-
Byte 6	unused	-	-	-	-	-	-	-	-
Byte 7	unused	-	-	-	-	-	-	-	-
Byte 8	unused	-	-	-	-	-	-	-	-

This command is used to inform the communicators which tag type to communicate with. On first start-up, the IPG-G4-B7-V15 instructs the IPT-FP heads to type 30h/33h.
The following types of data carriers are currently supported:

<Tagtype> High byte	<Tagtype> Low byte	Description	Access	<WordAdr>	Bits
30h	32h	IPC02	fixcode		40
30h	33h	IPC03	R/W(fixcode)	0 ... 1Dh	928 (32)
31h	30h	IPC10	R/W	00h	96

The last selected setting is stored in the IPG-G4-B7-V15. On start-up, the IPT-FP heads are instructed to use that tag type.

System Commands

reset (rs):

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (16h/22d)	0	0	0	1	0	1	1	0
Byte 1	Aport/PNum/Tbit	0	0	0	0	A	P	0	T
Byte 2	unused	-	-	-	-	-	-	-	-
Byte 3	unused	-	-	-	-	-	-	-	-
Byte 4	unused	-	-	-	-	-	-	-	-
Byte 5	unused	-	-	-	-	-	-	-	-
Byte 6	unused	-	-	-	-	-	-	-	-
Byte 7	unused	-	-	-	-	-	-	-	-
Byte 8	unused	-	-	-	-	-	-	-	-

Response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (16h/22d)	0	0	0	1	0	1	1	0
Byte 1	Aport/PNum/Tbit	0	0	0	0	A	P	0	T
Byte 2	Status	<Status>=2							
Byte 3	unused	-	-	-	-	-	-	-	-
Byte 4	unused	-	-	-	-	-	-	-	-
Byte 5	unused	-	-	-	-	-	-	-	-
Byte 6	unused	-	-	-	-	-	-	-	-
Byte 7	unused	-	-	-	-	-	-	-	-
Byte 8	unused	-	-	-	-	-	-	-	-

System settings of the IPG-G4-B7-V15 are reloaded from the nonvolatile memory.

System Commands

version (ve):

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (3h/3d)	0	0	0	0	0	0	1	1
Byte 1	Aport/PNum/Tbit	0	0	0	0	A	P	0	T
Byte 2	unused	-	-	-	-	-	-	-	-
Byte 3	unused	-	-	-	-	-	-	-	-
Byte 4	unused	-	-	-	-	-	-	-	-
Byte 5	unused	-	-	-	-	-	-	-	-
Byte 6	unused	-	-	-	-	-	-	-	-
Byte 7	unused	-	-	-	-	-	-	-	-
Byte 8	unused	-	-	-	-	-	-	-	-

Response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (3h/3d)	0	0	0	0	0	0	1	1
Byte 1	Aport/PNum/Tbit	0	0	0	0	A	P	0	T
Byte 2	Status	<Status>							
Byte 3	unused	-	-	-	-	-	-	-	-
Byte 4	version	<Data>							
Byte 5	version	<Data>							
Byte 6	unused	-	-	-	-	-	-	-	-
Byte 7	unused	-	-	-	-	-	-	-	-
Byte 8	unused	-	-	-	-	-	-	-	-

The version of the read head is returned. For example, Version 5.09 = 0509h.

Read Fixed Code Commands (sf, af, bf, ef)

Fixed code commands are used to access permanently stored information on the tag. An IPC03 tag will return 4 bytes of fixed code and an IPC02 will return 5 bytes of fixed code. These commands are also used after the X commands (write fixed code commands on page 32) have configured the IPC02 type. Except for the IPC10 type tags where you can program your own number into it, all tags are guaranteed to have a unique fixed code.

Single read fixed code (sf):

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (1h/1d)	0	0	0	0	0	0	0	1
Byte 1	APort/PNum/Tbit	0	0	0	0	A	P	0	T
Byte 2	unused	-	-	-	-	-	-	-	-
Byte 3	unused	-	-	-	-	-	-	-	-
Byte 4	unused	-	-	-	-	-	-	-	-
Byte 5	unused	-	-	-	-	-	-	-	-
Byte 6	unused	-	-	-	-	-	-	-	-
Byte 7	unused	-	-	-	-	-	-	-	-
Byte 8	unused	-	-	-	-	-	-	-	-

Response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (1h/1d)	0	0	0	0	0	0	0	1
Byte 1	APort/PNum/Tbit	0	0	0	0	A	P	0	T
Byte 2	Status	<Status>							
Byte 3	Execution counter	<Execution counter>							
Byte 4	ID-Code 00 .. FFh	<ID-Code 4>/<...3>							
Byte 5	ID-Code 00 .. FFh	<ID-Code 3>/<...2>							
Byte 6	ID-Code 00 .. FFh	<ID-Code 2>/<...1>							
Byte 7	ID-Code 00 .. FFh	<ID-Code 1>/<...0>							
Byte 8*	ID-Code 00 .. FFh	<ID-Code 0>							

* only IPC02

One attempt is made to read a fixed code.

Auto read fixed code (af):

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (8h/8d)	0	0	0	0	1	0	0	0
Byte 1	APort/PNum/Tbit	0	0	0	0	A	P	0	T
Byte 2	unused	-	-	-	-	-	-	-	-
Byte 3	unused	-	-	-	-	-	-	-	-
Byte 4	unused	-	-	-	-	-	-	-	-
Byte 5	unused	-	-	-	-	-	-	-	-
Byte 6	unused	-	-	-	-	-	-	-	-
Byte 7	unused	-	-	-	-	-	-	-	-
Byte 8	unused	-	-	-	-	-	-	-	-

Response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (8h/8d)	0	0	0	0	1	0	0	0
Byte 1	APort/PNum/Tbit	0	0	0	0	A	P	0	T
Byte 2	Status	<Status>							
Byte 3	Execution counter	<Execution counter>							
Byte 4	ID-Code 00 .. FFh	<ID-Code 4>/<...3>							
Byte 5	ID-Code 00 .. FFh	<ID-Code 3>/<...2>							
Byte 6	ID-Code 00 .. FFh	<ID-Code 2>/<...1>							
Byte 7	ID-Code 00 .. FFh	<ID-Code 1>/<...0>							
Byte 8*	ID-Code 00 .. FFh	<ID-Code 0>							

* only IPC02

A read attempt is made until a fixcode is successfully read.

Buffered read fixed code (bf):

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (9h/9d)	0	0	0	0	1	0	0	1
Byte 1	APort/PNum/Tbit	0	0	0	0	A	P	0	T
Byte 2	unused	-	-	-	-	-	-	-	-
Byte 3	unused	-	-	-	-	-	-	-	-
Byte 4	unused	-	-	-	-	-	-	-	-
Byte 5	unused	-	-	-	-	-	-	-	-
Byte 6	unused	-	-	-	-	-	-	-	-
Byte 7	unused	-	-	-	-	-	-	-	-
Byte 8	unused	-	-	-	-	-	-	-	-

Response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (9h/9d)	0	0	0	0	1	0	0	1
Byte 1	APort/PNum/Tbit	0	0	0	0	A	P	0	T
Byte 2	Status	<Status>							
Byte 3	Execution counter	<Execution counter>							
Byte 4	ID-Code 00 .. FFh	<ID-Code 4>/<...3>							
Byte 5	ID-Code 00 .. FFh	<ID-Code 3>/<...2>							
Byte 6	ID-Code 00 .. FFh	<ID-Code 2>/<...1>							
Byte 7	ID-Code 00 .. FFh	<ID-Code 1>/<...0>							
Byte 8*	ID-Code 00 .. FFh	<ID-Code 0>							

* only IPC02

The fixcode is continuously read. Only changing data (i.e. new tag data) is transmitted over the interface.

Enhanced buffered read fixed code (ef):

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (1Dh/29d)	0	0	0	1	1	1	0	1
Byte 1	APort/PNum/Tbit	0	0	0	0	A	P	0	T
Byte 2	unused	-	-	-	-	-	-	-	-
Byte 3	unused	-	-	-	-	-	-	-	-
Byte 4	unused	-	-	-	-	-	-	-	-
Byte 5	unused	-	-	-	-	-	-	-	-
Byte 6	unused	-	-	-	-	-	-	-	-
Byte 7	unused	-	-	-	-	-	-	-	-
Byte 8	unused	-	-	-	-	-	-	-	-

Response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (1Dh/29d)	0	0	0	1	1	1	0	1
Byte 1	APort/PNum/Tbit	0	0	0	0	A	P	0	T
Byte 2	Status	<Status>							
Byte 3	Execution counter	<Execution counter>							
Byte 4	ID-Code 00 .. FFh	<ID-Code 4>/<...3>							
Byte 5	ID-Code 00 .. FFh	<ID-Code 3>/<...2>							
Byte 6	ID-Code 00 .. FFh	<ID-Code 2>/<...1>							
Byte 7	ID-Code 00 .. FFh	<ID-Code 1>/<...0>							
Byte 8*	ID-Code 00 .. FFh	<ID-Code 0>							

* only IPC02

This command functions the same as the buffered read fixcode command; however, "Status 5" (no tag) is received when the tag leaves the read zone.

Read/Write Commands (sr, ar, br, er, sw, aw, bw, ew)

When the length of a read or write command is set to 01h in <WordNum>, a specific area on the tag is used. This length is fixed, which means the start address needs to be changed in order to manipulate other 32 bit words on the tag. The number of double words on the tag may vary depending on the tag type and can range from 1 to 29 double words. These different locations on the tag are accessed by changing the word address, <WordAddr>, parameter and then reissuing the node adapter command.

Preconfigured Read Commands (sr, ar, br, er)

If a read command is issued with a length, <WordNum>, of 00h and a start address, <WordAddr>, of 0000h then the communicator is put into preconfigured mode. This communicator in this mode assumes the start address and length are stored on the tag and waits for data. In order to use this feature the tag needs to have these two parameters, start address and length, stored on the tag. This can be done using P+F configuration software and an IDENT-I System P communicator connected to a PC. Even though the tag is preconfigured, it can still be accessed normally using standard read/ write commands with a length of 0001h.

Reasons for Using Preconfigured Read Commands

Because the tag knows what to send to the communicator, data transmission is only in one direction. This means the tag does not need to interpret a command from the communicator, but merely sends information back.

The read range is increased, because the data transmission is unidirectional. This increase in range as well as the increase in the read speed because of the quick command execution allows for a user to "read on the fly" at much faster speeds. Please contact P+F for details on preconfiguring tags.

Read/write commands

Single read (sr):

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (10h/16d)	0	0	0	1	0	0	0	0
Byte 1	WordNum/APort/PNum/Tbit	<WordNum>				A	P	0	T
Byte 2	Word address	<WordAddr> (High Byte)							
Byte 3	Word address	<WordAddr> (Low Byte)							
Byte 4	unused	-	-	-	-	-	-	-	-
Byte 5	unused	-	-	-	-	-	-	-	-
Byte 6	unused	-	-	-	-	-	-	-	-
Byte 7	unused	-	-	-	-	-	-	-	-
Byte 8	unused	-	-	-	-	-	-	-	-

Number of words, <WordNum>, can only be <0001> or <0000>.

Response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (10h/16d)	0	0	0	0	1	0	0	0
Byte 1	WordNum/APort/PNum/Tbit	<WordNum>				A	P	0	T
Byte 2	Status	<Status>							
Byte 3	Execution counter	<Execution counter>							
Byte 4	data 00 .. FFh	<Data>							
Byte 5	data 00 .. FFh	<Data>							
Byte 6	data 00 .. FFh	<Data>							
Byte 7	data 00 .. FFh	<Data>							
Byte 8	unused	-	-	-	-	-	-	-	-

One attempt is made to read one 32-bit word starting at address <WordAddr>.

Auto read (ar):

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (20h/32d)	0	0	1	0	0	0	0	0
Byte 1	WordNum/APort/PNum/Tbit	<WordNum>				A	P	0	T
Byte 2	Word address	<WordAddr> (High Byte)							
Byte 3	Word address	<WordAddr> (Low Byte)							
Byte 4	unused	-	-	-	-	-	-	-	-
Byte 5	unused	-	-	-	-	-	-	-	-
Byte 6	unused	-	-	-	-	-	-	-	-
Byte 7	unused	-	-	-	-	-	-	-	-
Byte 8	unused	-	-	-	-	-	-	-	-

Number of words, <WordNum>, can only be <0 0 0 1> or <0 0 0 0>.
Response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (20h/32d)	0	0	1	0	0	0	0	0
Byte 1	WordNum/APort/PNum/Tbit	<WordNum>				A	P	0	T
Byte 2	Status	<Status>							
Byte 3	Execution counter	<Execution counter>							
Byte 4	data 00 .. FFh	<Data>							
Byte 5	data 00 .. FFh	<Data>							
Byte 6	data 00 .. FFh	<Data>							
Byte 7	data 00 .. FFh	<Data>							
Byte 8	unused	-	-	-	-	-	-	-	-

A read attempt is made until successful to read one 32-Bit word starting at <WordAddr>.

Buffered read (br):

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (30h/48d)	0	0	1	1	0	0	0	0
Byte 1	WordNum/APort/PNum/Tbit	<WordNum>				A	P	0	T
Byte 2	Word address	<WordAddr> (High Byte)							
Byte 3	Word address	<WordAddr> (Low Byte)							
Byte 4	unused	-	-	-	-	-	-	-	-
Byte 5	unused	-	-	-	-	-	-	-	-
Byte 6	unused	-	-	-	-	-	-	-	-
Byte 7	unused	-	-	-	-	-	-	-	-
Byte 8	unused	-	-	-	-	-	-	-	-

Number of words, <WordNum>, can only be <0 0 0 1> or <0 0 0 0>.
Response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (30h/48d)	0	0	1	1	0	0	0	0
Byte 1	WordNum/APort/PNum/Tbit	<WordNum>				A	P	0	T
Byte 2	Status	<Status>							
Byte 3	Execution counter	<Execution counter>							
Byte 4	data 00 .. FFh	<Data>							
Byte 5	data 00 .. FFh	<Data>							
Byte 6	data 00 .. FFh	<Data>							
Byte 7	data 00 .. FFh	<Data>							
Byte 8	unused	-	-	-	-	-	-	-	-

Continuous attempts are made to read one 32-Bit word from address <WordAddr>. Only new data is transmitted over the interface. This occurs when a new data carrier is read.

Enhanced buffered read (er):

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (19h/25d)	0	0	0	1	1	0	0	1
Byte 1	WordNum/APort/PNum/Tbit	<WordNum>				A	P	0	T
Byte 2	Word address	<WordAddr> (High Byte)							
Byte 3	Word address	<WordAddr> (Low Byte)							
Byte 4	unused	-	-	-	-	-	-	-	-
Byte 5	unused	-	-	-	-	-	-	-	-
Byte 6	unused	-	-	-	-	-	-	-	-
Byte 7	unused	-	-	-	-	-	-	-	-
Byte 8	unused	-	-	-	-	-	-	-	-

Number of words, <WordNum>, can only be <0 0 0 1> or <0 0 0 0>.
Response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (19h/25d)	0	0	0	1	1	0	0	1
Byte 1	WordNum/APort/PNum/Tbit	<WordNum>				A	P	0	T
Byte 2	Status	<Status>							
Byte 3	Execution counter	<Execution counter>							
Byte 4	data 00 .. FFh	<Data>							
Byte 5	data 00 .. FFh	<Data>							
Byte 6	data 00 .. FFh	<Data>							
Byte 7	data 00 .. FFh	<Data>							
Byte 8	unused	-	-	-	-	-	-	-	-

Continuous attempts are made to read one 32-Bit word from address <WordAddr>. Only new data is transmitted over the interface. Status "5" (no tag) is received when the data carrier leaves the read zone.

Single write (sw):

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (40h/64d)	0	1	0	0	0	0	0	0
Byte 1	WordNum/APort/PNum/Tbit	<WordNum>				A	P	0	T
Byte 2	Word address	<WordAddr> (High Byte)							
Byte 3	Word address	<WordAddr> (Low Byte)							
Byte 4	Data	<Data>							
Byte 5	Data	<Data>							
Byte 6	Data	<Data>							
Byte 7	Data	<Data>							
Byte 8	unused	-	-	-	-	-	-	-	-

Number of words, <WordNum>, can only be <0 0 0 1>.
Response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (40h/64d)	0	1	0	0	0	0	0	0
Byte 1	WordNum/APort/PNum/Tbit	<WordNum>				A	P	0	T
Byte 2	Status	<Status>							
Byte 3	Execution counter	<Execution counter>							
Byte 4	unused	-	-	-	-	-	-	-	-
Byte 5	unused	-	-	-	-	-	-	-	-
Byte 6	unused	-	-	-	-	-	-	-	-
Byte 7	unused	-	-	-	-	-	-	-	-
Byte 8	unused	-	-	-	-	-	-	-	-

A write attempt is made from address <WordAddr> to write one 32-Bit word to <WordAddr>.

Auto write (aw):

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (50h/80d)	0	1	0	1	0	0	0	0
Byte 1	WordNum/APort/PNum/Tbit	<WordNum>				A	P	0	T
Byte 2	Word address	<WordAddr> (High Byte)							
Byte 3	Word address	<WordAddr> (Low Byte)							
Byte 4	Data	<Data>							
Byte 5	Data	<Data>							
Byte 6	Data	<Data>							
Byte 7	Data	<Data>							
Byte 8	unused	-	-	-	-	-	-	-	-

Number of words, <WordNum>, can only be <0 0 0 1>.

Response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (50h/80d)	0	1	0	1	0	0	0	0
Byte 1	WordNum/APort/PNum/Tbit	<WordNum>				A	P	0	T
Byte 2	Status	<Status>							
Byte 3	Execution counter	<Execution counter>							
Byte 4	unused	-	-	-	-	-	-	-	-
Byte 5	unused	-	-	-	-	-	-	-	-
Byte 6	unused	-	-	-	-	-	-	-	-
Byte 7	unused	-	-	-	-	-	-	-	-
Byte 8	unused	-	-	-	-	-	-	-	-

A continuous attempt is made to write 32-Bit words to <WordAddr>.

Buffered write (bw):

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (60h/96d)	0	1	1	0	0	0	0	0
Byte 1	WordNum/APort/PNum/Tbit	<WordNum>				A	P	0	T
Byte 2	Word address	<WordAddr> (High Byte)							
Byte 3	Word address	<WordAddr> (Low Byte)							
Byte 4	Data	<Data>							
Byte 5	Data	<Data>							
Byte 6	Data	<Data>							
Byte 7	Data	<Data>							
Byte 8	unused	-	-	-	-	-	-	-	-

Number of words, <WordNum>, can only be <0 0 0 1>.
Response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (60h/96d)	0	1	1	0	0	0	0	0
Byte 1	WordNum/APort/PNum/Tbit	<WordNum>				A	P	0	T
Byte 2	Status	<Status>							
Byte 3	Execution counter	<Execution counter>							
Byte 4	unused	-	-	-	-	-	-	-	-
Byte 5	unused	-	-	-	-	-	-	-	-
Byte 6	unused	-	-	-	-	-	-	-	-
Byte 7	unused	-	-	-	-	-	-	-	-
Byte 8	unused	-	-	-	-	-	-	-	-

A continuous attempt is made until successful to write one 32-Bit word to tag address <WordAddr>. After each successful attempt, the response is transmitted and the head automatically begins continuous reading. If a new data carrier is recognized or the same data carrier leaves the read/write zone and returns the write command executes again.

Enhanced buffered write (ew):

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (1Ah/26d)	0	0	0	1	1	0	1	0
Byte 1	WordNum/APort/PNum/Tbit	<WordNum>				A	P	0	T
Byte 2	Word address	<WordAddr> (High Byte)							
Byte 3	Word address	<WordAddr> (Low Byte)							
Byte 4	Data	<Data>							
Byte 5	Data	<Data>							
Byte 6	Data	<Data>							
Byte 7	Data	<Data>							
Byte 8	unused	-	-	-	-	-	-	-	-

Number of words, <WordNum>, can only be <0 0 0 1>.
Response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (1Ah/26d)	0	0	0	1	1	0	1	0
Byte 1	WordNum/APort/PNum/Tbit	<WordNum>				A	P	0	T
Byte 2	Status	<Status>							
Byte 3	Execution counter	<Execution counter>							
Byte 4	unused	-	-	-	-	-	-	-	-
Byte 5	unused	-	-	-	-	-	-	-	-
Byte 6	unused	-	-	-	-	-	-	-	-
Byte 7	unused	-	-	-	-	-	-	-	-
Byte 8	unused	-	-	-	-	-	-	-	-

This command functions in the same way as the buffered write command, except that Status "5" (no tag) is received when the data carrier leaves the read write zone.

IPC10 Commands

An IPC10 tag can be formatted to be a read/write tag with one, 32 bit word, or to emulate an IPC02. The first time the tag is used, it is permanently formatted and cannot be changed.

Standard Commands (sr, ar, br, er, sw, aw, bw, ew)

When writing data to this tag, the start address, <WordAddr>, must always be 00h and the length, <WordNum> always 1h. Because the tag is not preconfigured, it will be formatted the first time data is written to it. The data can be changed, but the address and the length are now fixed. When reading this type of tag, the start address, <WordAddr>, must be 00h and the length, <WordNum> always 0h.

Write Fixed Code Commands (sx, ax, bx, ex)

The IPC10 can also be programmed to emulate an IPC02 fixed code tag using the fixed code commands (X commands). The 'ct' (change tag command on page 16) command needs to be issued to change the tag type to an IPC10. Once the tag type IPC10 is chosen, five bytes of <Data> are written to the tag using one of the X commands. Once written, the tag can be read as type IPC10 or IPC02 using read fixed code commands: sf, af, bf and ef. **Note: Once written, the data cannot be changed.**

Single write fixed code (sx):

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (1Fh/31d)	0	0	0	1	1	1	1	1
Byte 1	Fixlen/APort/PNum/Tbit	<Fixlen>				A	P	0	T
Byte 2	Fix type	<Fixtyp>							
Byte 3	Fix type	<Fixtyp>							
Byte 4	data 00 .. FFh	<Data>							
Byte 5	data 00 .. FFh	<Data>							
Byte 6	data 00 .. FFh	<Data>							
Byte 7	data 00 .. FFh	<Data>							
Byte 8	data 00 .. FFh	<Data>							

Response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (1Fh/32d)	0	0	0	1	1	1	1	1
Byte 1	Fixlen/APort/PNum/Tbit	<Fixlen>				A	P	0	T
Byte 2	Status	<Status>							
Byte 3	Execution counter	<Execution counter>							
Byte 4	unused	-	-	-	-	-	-	-	-
Byte 5	unused	-	-	-	-	-	-	-	-
Byte 6	unused	-	-	-	-	-	-	-	-
Byte 7	unused	-	-	-	-	-	-	-	-
Byte 8	unused	-	-	-	-	-	-	-	-

One attempt is made to write a fixed code. <Fixtype> is always 30h/32h for IPC10 and <Fixlen> is always 5h, since 5 bytes must always be written.

Auto write fixed code (ax):

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (64h/100d)	0	1	1	0	0	1	0	0
Byte 1	Fixlen/APort/PNum/Tbit	<Fixlen>				A	P	0	T
Byte 2	Fix type	<Fixtyp>							
Byte 3	Fix type	<Fixtyp>							
Byte 4	data 00 .. FFh	<Data>							
Byte 5	data 00 .. FFh	<Data>							
Byte 6	data 00 .. FFh	<Data>							
Byte 7	data 00 .. FFh	<Data>							
Byte 8	data 00 .. FFh	<Data>							

Response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (64h/100d)	0	1	1	0	0	1	0	0
Byte 1	Fixlen/APort/PNum/Tbit	<Fixlen>				A	P	0	T
Byte 2	Status	<Status>							
Byte 3	Execution counter	<Execution counter>							
Byte 4	unused	-	-	-	-	-	-	-	-
Byte 5	unused	-	-	-	-	-	-	-	-
Byte 6	unused	-	-	-	-	-	-	-	-
Byte 7	unused	-	-	-	-	-	-	-	-
Byte 8	unused	-	-	-	-	-	-	-	-

An attempt is made until successful to write a fixed code. <Fixtype> is always 30h/32h for the IPC10 and <Fixlen> is always 5h, since 5 bytesmust always be written.

Buffered write fixed code (bx):

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (65h/101d)	0	1	1	0	0	1	0	1
Byte 1	Fixlen/APort/PNum/Tbit	<Fixlen>				A	P	0	T
Byte 2	Fix type	<Fixtyp>							
Byte 3	Fix type	<Fixtyp>							
Byte 4	data 00 .. FFh	<Data>							
Byte 5	data 00 .. FFh	<Data>							
Byte 6	data 00 .. FFh	<Data>							
Byte 7	data 00 .. FFh	<Data>							
Byte 8	data 00 .. FFh	<Data>							

Response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (65h/101d)	0	1	1	0	0	1	0	1
Byte 1	Fixlen/APort/PNum/Tbit	<Fixlen>				A	P	0	T
Byte 2	Status	<Status>							
Byte 3	Execution counter	<Execution counter>							
Byte 4	unused	-	-	-	-	-	-	-	-
Byte 5	unused	-	-	-	-	-	-	-	-
Byte 6	unused	-	-	-	-	-	-	-	-
Byte 7	unused	-	-	-	-	-	-	-	-
Byte 8	unused	-	-	-	-	-	-	-	-

One attempt is made to write a fixed code. After each successful write, the response is sent and the system waits until a new data carrier is within the detection range. Then the command starts again from the beginning. <Fixtype> is always 30h/32h for the IPC10 and the <Fixlen> is always 5h, since 5 bytes must always be written.

Enhanced buffered write fixed code (ex):

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (24h/36d)	0	0	1	0	0	1	0	0
Byte 1	Fixlen/APort/PNum/Tbit	<Fixlen>				A	P	0	T
Byte 2	Fix type	<Fixtyp>							
Byte 3	Fix type	<Fixtyp>							
Byte 4	data 00 .. FFh	<Data>							
Byte 5	data 00 .. FFh	<Data>							
Byte 6	data 00 .. FFh	<Data>							
Byte 7	data 00 .. FFh	<Data>							
Byte 8	data 00 .. FFh	<Data>							

Response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Command code (24h/36d)	0	0	1	0	0	1	0	0
Byte 1	Fixlen/APort/PNum/Tbit	<Fixlen>				A	P	0	T
Byte 2	Status	<Status>							
Byte 3	Execution counter	<Execution counter>							
Byte 4	unused	-	-	-	-	-	-	-	-
Byte 5	unused	-	-	-	-	-	-	-	-
Byte 6	unused	-	-	-	-	-	-	-	-
Byte 7	unused	-	-	-	-	-	-	-	-
Byte 8	unused	-	-	-	-	-	-	-	-

The command functions the same as the buffered write fixed code command. <Fixtype> is always 30h/32h for IPC10 and <Fixlen> is always 5h, since 5 bytes must always be written. A status 5 is sent when no tag is present.

Key:

<Status>:	8 bit 00h - FFh
<Tagtype>:	16 bit 0000h - FFFFh
<WordAddr>:	Double word start address from 0000h - FFFFh depending on data carrier type
<WordNum>:	Number of double words to be read or written, 4 bit range. Possible values from 0h to Fh depending on the data carrier type The following applies to the IPC03 and IPC10: The number of words 0h is used with word address 0000h for reading the data range preset on the data carrier (preconfigured read).
<Execution Counter>:	No. of times data has been read/written, 8 bit from 00h - FFh
<Data>:	Tag data
<ID-Code>:	4 bytes 5 bytes for IPC02
<A>:	All ports bit; 1=all ports
<P>:	Ports bit; 0=port 1, 1=port 2
<T>:	Toggle bit; change to issue same command again
<Fixlen>:	No. of bytes to write; always 5h $(0101)_b$
<Fixtyp>:	Type of tag to make; always IPC02 30h/32h $(00110000)_b$, $(00110010)_b$

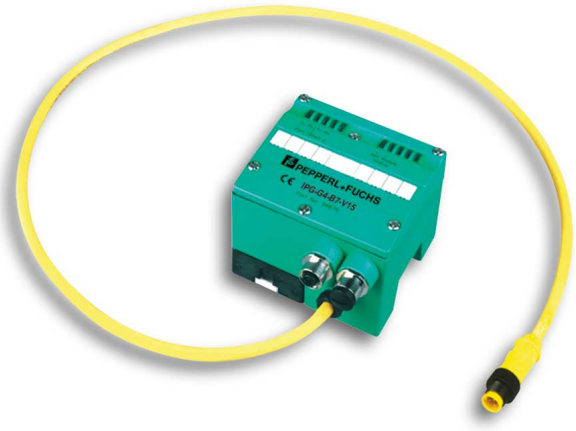
Error/Status Messages:

Status	Meaning
00h	Command has been executed without error.
02h	Switch-on message, reset has been executed.
03h	Reserved
04h	Incorrect or incomplete command sent to read head
05h	Read/write error, no data carrier
06h	Read head, IPT-FP missing
08h	Reserved
09h	Reserved
0Ah	Reserved
0Bh	Reserved
0Ch	Reserved
0Dh	Reserved
0Eh	Reserved
0Fh	Reserved
10h	Reserved
FFh	Command being processed
40h	Corrupt data, wrong checksum from the read head

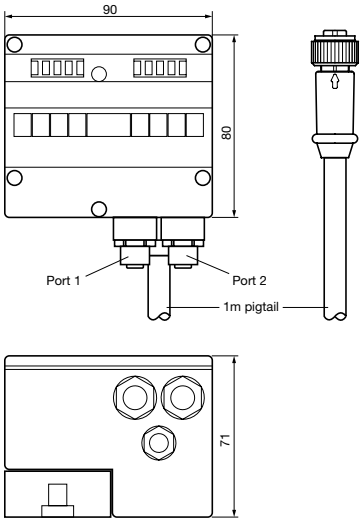
Technical data

Bus interface:	IPG-G4-B7-V15 DeviceNet Bus Coupler
Baud rate	125; 250; 500 kbits/s
Address setting	rotary switch
Bus termination	rotary switch
Electrical data:	
Operating voltage	11-26 V
Current Consumption	≤ 24 V @ 100 mA; ≤ 11 V @ 160 mA
Mechanical data (lower section IPG-G4-B7-V15):	
Housing material	Aluminum, black anodized
Operating temperature	248 Kelvin to 343 Kelvin (-25°C to +70°C)
Storage temperature	233 Kelvin to 358 Kelvin (-40°C to +85°C)
Protection class per EN60529	IP 67 with the IPT-FP PG9 cable glands that are not used must be sealed with the plugs provided. The plugs are suitable for both the standard cable glands and the EMC cable glands.
Connections	
Bus cable:	1-meter male pigtail
Voltage supply:	Supplied with five-conductor DeviceNet cable

IPG-G4-B7-V15 Housing



Dimensions (mm)



Issued Apr/2003

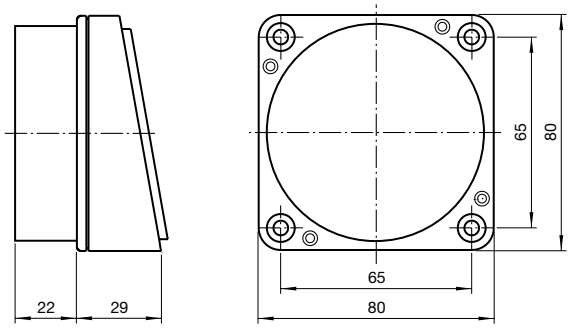
Technical data

Read/write head	IPT-FP
Mechanical data:	
Housing material	PTB
Operating temperature	248 Kelvin to 343 Kelvin (-25°C to +70°C)
Storage temperature	233 Kelvin to 358 Kelvin (-40°C to +85°C)
Protection class per EN60529	IP 67 with the U-P4-R4-V15
Voltage supply	via U-P4-R4-V15

Read/write distances are determined by data carrier. See data carrier data sheet for ranges.

IPT-FP Housing

IPT-FP Dimensions (mm)

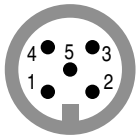


Issued Apr/2003

Technical data

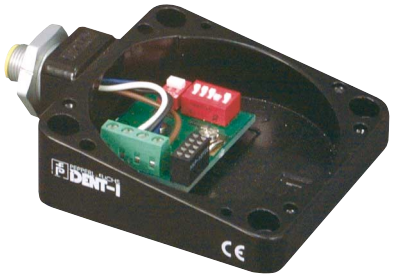
Serial interface:	U-P4-R4-V15 read/write head base
Baud rate	1200, 2400, 4800, 9600, 19200 kbit/s
Total cable length	1200 m
Electrical data:	
Operating voltage U_B	20 -30 VDC, PELV
Current Consumption	< 300 mA
Remaining ripple	± 10% at U_B=30 V
Mechanical data:	
Housing material	Aluminum
Operating temperature	248 Kelvin to 343 Kelvin (-25°C to +70°C)
Storage temperature	233 Kelvin to 358 Kelvin (-40°C to +85°C)
Protection class per EN60529	IP 67
Connection:	
Method of connection	5-pin, M12 quick disconnect
Interface cable	2-wire acc. to RS485
Power supply	up to 3 x 1.5 mm²

Male Receptacle

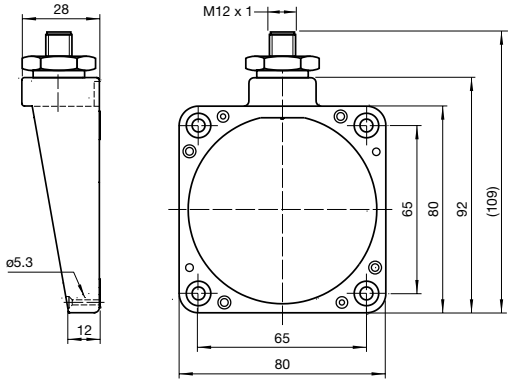


1 24V (+)
2 A
3 24V (–)/ground
4 B
5 Shield

U-P4-R4-V15 Housing



U-P4-R4-V15 Dimensions (mm)



General Information

The IPG-G4-B7 node adapter device operates as a slave on the DeviceNet network. The unit supports Explicit Messages and Polled I/O Messages of the predefined master/slave connection set. It does not support the Explicit Unconnected Message Manager (UCMM).

The unit links the P+F identification systems IDENT-I System P to DeviceNet. Connection between the node adapter and the Ident-I System P Communicator is established via 2-wire RS485 serial communication. The IPG-G4-B7 has two UART's, such that up two Ident-I System P Communicators can be connected.

The Communicators are auto-configured to operate at 19.2Kbaud at power up. At power up of the IPG-G4-B7 node adapter the System P Communicators must be set to 9600 baud or 19.2Kbaud for this to work. If a Communicator is not found during the power up phase it can not be added to the system until the IPG-G4-B7 it is powered down and back up again. As factory default the node adapter defaults to the IPC03 tag types and must be changed to support IPC02 or IPC10 type tags. Once changed the new settings are stored in non-volatile RAM on the IPG-G4-B7. At power up the Communicators are configured to match those settings.

The IPG-G4-B7 node adapter has two bicolor DeviceNet LED's and four bicolor LED's for the serial channel ports that communicate to the Communicators. There is a Transmit and Receive indicator for each serial port. A green light will flash when the IPG-G4-B7 node adapter is sending or receiving a character over the RS485 channel.

Note about Table Values

The tables throughout this document provide data values for various attributes. Note that values in parenthesis are actually notes that will describe the information. They can be found following the table.

DeviceNet Message Types

As a group 2 slave device the IPG-G4-B7 supports the following message types.

CAN IDENTIFIER	GROUP 2 Message Type
10xxxxxx111	Duplicate MACID Check Message
10xxxxxx110	Unconnected Explicit Request Message
10xxxxxx101	Master I/O Poll Command Message
10xxxxxx100	Master Explicit Request Message

xxxxxx = Node Address

DeviceNet Class Services

As a group-2 slave device the IPG-G4-B7 supports the following class services and instance services.

SERVICE CODE	SERVICE NAME
05 (0x05)	Reset
14 (0x0E)	Get Attribute Single
16 (0x10)	Set Attribute Single
75 (0x4B)	Allocate Group 2 Identifier Set
76 (0x4C)	Release Group 2 Identifier Set

DeviceNet Object Classes

The IPG-G4-B7 device supports the following DeviceNet object classes.

CLASS CODE	OBJECT TYPE
01 (0x01)	Identity
02 (0x02)	Router
03 (0x03)	DeviceNet
04 (0x04)	Assembly
05 (0x05)	Connection
64 (0x40)	User defined Tag Object

Identity Object Class Code: 01 (0x01)

The Identity Object is required on all devices and provides identification of and general information about the device.

Identity Object Class Attributes

Attribute	Access	Name	Type	Value
1	Get	Revision	UINT	1
2	Get	Max Object Instance	UINT	1
6	Get	Max Class Identifier	UINT	7
7	Get	Max Instance Attribute	UINT	7

Identity Object, Instance 1 Attributes

Attribute	Access	Name	Type	Value
1	Get	Vendor	UINT	57
2	Get	Product Type	UINT	00
3	Get	Product Code	UINT	5696
4	Get	Revision	STRUCT OF	
		Major Revision	USINT	2
		Minor Revision	USINT	021
5	Get	Device Status	UINT	(1)
6	Get	Serial Number	UINT	(2)
7	Get	Product Name	STRUCT OF	
		Length	USINT	9
		Name	STRING [6]	IPG-G4-B7

Common Services

Service Code	Class	Instance	Service Name
05 (0x05)	No	Yes	Reset
14 (0x0E)	Yes	Yes	Get_Attribute_Single

(1) Device Status

bit 0	owned	0=not owned 1=owned (allocated)
bit 1	reserved	0
bit 2	configured	0
bit 3	reserved	0
bit 4-7	vendor specific	0
bit 8	minor cfg fault	0=no fault 1=minor fault
bit 9	minor dev.fault	0=no fault 1=minor device fault
bit 10	major cfg.fault	0=no fault 1=major cfg. fault
bit 11	major dev.fault	0=no fault 1=major device fault
bit 12-15	reserved	0

(2) Unique Serial Number

Router Object Class Code: 02 (0x02)

The Message Router Object provides a messaging-connection point through which a Client may address a service to any object class or instance residing in the physical device.

Router Object Class Attributes

Attribute	Access	Name	Type	Value
1	Get	Revision	UINT	1
6	Get	Max Class Identifier	UINT	7
7	Get	Max Instance Attribute	UINT	2

Router Object, Instance 1 Attributes

Attribute	Access	Name	Type	Value
2	Get	Number of Connections	UINT	2

Common Services

Service Code	Class	Instance	Service Name
14 (0x0E)	Yes	Yes	Get_Attribute_Single

DeviceNet Object Class Code: 03 (0x03)

DeviceNet Object Class Attributes

Attribute	Access	Name	Type	Value
1	Get	Revision	UINT	2

Router Object, Instance 1 Attributes

Attribute	Access	Name	Type	Value
1	Get/Set	MACID	USINT	(1)
2	Get/Set	Baud Rate	USINT	(2)
3	Get/Set	Bus Off Interrupt	BOOL	(3)
4	Get/Set	Bus Off Counter	USINT	(4)
5	Get	Allocation Information	STRUCT of	(5)
		Choice Byte	BYTE	
		Master Node Addr.	USINT	
6	Get	MAC ID Switch changed	BOOL	0 = No Change 1= Changed since last Reset or Power-up.
7	Get	MAC ID Switch Value	BOOL	0 = No Change 1= Changed since last Reset or Power-up.
8	Get	MAC ID Switch Value	USINT	Range 0 - 63
9	Get	Baud Rate Switch Value	USINT	Range 0 - 2

(1) Settable only if the MacID switches are set to a value greater than 63. Value returned will be switch value if less than 64 or the last value set.

(2) Settable only if the Baud Rate switch is set to a value greater than 2. Value returned will be switch value if less than 4 or the last value set.

Switch/Value	Speed
0	125 kbits
1	250 kbits
2	500 kbits
3	Software settable

(3) Bus Off Interrupt (BOI) determines action if Bus Off state encountered. Following values are supported:

BOI	Action
0	Hold chip in OFF state (default)
1	If possible reset CAN chip

(4) Bus Off Counter will be forced to 0 whenever set regardless of the data value provided.

(5) Allocation_byte

bit 0 explicit set to 1 to allocate
bit 1 polled set to 1 to allocate
bit 2 strobed (not supported)
bit 3-7 reserved (always 0)

Common Services

Service Code	**Class**	**Instance**	**Service Name**
14 (0x0E)	Yes	Yes	Get_Attribute_Single
16 (0x10)	No	Yes	Set_Attribute_Single
75 (0x4B)	No	Yes	Allocate Master/Slave
76 (0x4C)	No	Yes	Release Master/Slave

Assembly Object Class Code: 04 (0x04)

The Assembly Objects bind attributes of multiple objects to allow data to or from each object to be sent or received over a single connection.

Assembly Object Class Attributes

Attribute	Access	Name	Type	Value
1	Get	Revision	UINT	2
2	Get	Max Class ID	UINT	2

Assembly Object, Instance 1 Attributes (input assembly)

Attribute	Access	Name	Type	Value
3	Get	Communicator output stream	see notes	(1)

Assembly Object, Instance 2 Attributes (output assembly)

Attribute	Access	Name	Type	Value
3	Get/Set	Communicator input stream	see notes	(2)

Common Services

Service Code	Class	Instance	Service Name
14 (0x0E)	Yes	Yes	Get_Attribute_Single
16 (0x10)	Yes	Yes	Set_Attribute_Single

(1) The IPG-G4-B7 will produce 9-bytes. The format is shown below, where the bottom row is the number of the byte transmitted in the DeviceNet response:

Cmnd	HLT	Status	EC	Data1	Data2	Data3	Data4	Data5
1	2	3	4	5	6	7	8	9

Cmnd = Command the head (head refers to a Communicator) is given (see command table)
HLT = Head # /Length/ Toggle Flag- HLT: 8 bit value

L3	L2	L1	L0	B	port	H	Tog
7	6	5	4	3	2	1	0

Where, L3-0 = length to read/write in double words (currently 0 or 1)
L3 is the MSbit and L0 is the LSbit.

H = Head number (0=head1,1=head2)
Port = 0 for port 1
= 1 for port 2
B = broadcast command (set this bit to send command to all available heads)
Tog = Toggle flag

Status = Head status information- this is the same status in ASCII as the Communicator returns on the RS485 side, except for a few special values listed below:
FF = command has not been processed yet.
06 = RS485 serial communication problem.
EC = Execution Counter
Data1 = MSB of data. There are at least four bytes of data returned from a tag read.
Data5 = LSB of data. Note that this is set to zero unless reading an IPC02 fix-code value.

(2) The IPG-G4-B7 will consume 9 bytes of information for the output assembly. The format is shown below where the bottom row is the number of the byte received on a DeviceNet request:

Cmnd	HLT	Param1	Param2	Data1	Data2	Data3	Data4	Data5
1	2	3	4	5	6	7	8	9

Cmnd = Command given to the Communicator (see the Command Table)
HLT = This is the head number, Length, and Toggle flag explained above.
Param1 = MSB of the Tag address to read from or write to, or other parameter.
Param2 = LSB of the Tag address to read from or write to, or other parameter.
Data1-5 = Data to be written to a tag during a write command (MSB-LSB). Note that when issuing a 'ct' (change tag type command) that Param1-2 locations contain the Tag Type in ASCII (i.e. param1=30hex param2=32hex for IPC02 tags). Data1-5 are

'don't cares'. When using the version command, Data1-4 contains the hex converted date and Data5 is a 'don't care'.

Refer to the Communicator Object class 64 for further information.

Connection Object Class Code: 05 (0x05)

The Connection Objects manage the characteristics of each communication connection. As a Group II Only Slave device the unit supports one explicit message connection and a POLL message connection.

Connection Object Class Attributes

Attribute	Access	Name	Type	Value
1	Get	Revision	UINT	1

Connection Object, Instance 1 Attributes (Explicit Message)

Attribute	Access	Name	Type	Value
1	Get	State	USINT	(1)
2	Get	Instance Type	USINT	0 = Explicit Message
3	Get	Transport Class Trigger	USINT	0x83
4	Get	Production Connection	UINT	(2)
5	Get	Consumed Connection	UINT	(2)
6	Get	Initial Comm. Char.	USINT	0x21
7	Get	Production Size	UINT	20
8	Get	Consumed Size	UINT	20
9	Get/Set	Expected Packet Rate	UINT	Default 2500 msec
12	Get/Set	Timeout Action	USINT	(3)
13	Get	Prod. Path Length	USINT	0
14	Get	Production Path		(null)
15	Get	Cons. Path Length	USINT	0
16	Get	Consumed Path		(null)
17	Get	Production Inhibit	UINT	0

Connection Object, Instance 2 Attributes (POLL connection)

Attribute	Access	Name	Type	Value
1	Get	State	USINT	(1)
2	Get	Instance Type	USINT	1 = I/O Message
3	Get	Transport Class Trigger	USINT	0x82
4	Get	Production Connection	UINT	(2)
5	Get	Consumed Connection	UINT	(2)
6	Get	Initial Comm. Char.	USINT	0x1
7	Get	Production Size	UINT	9 bytes
8	Get	Consumed Size	UINT	9 bytes
9	Get/Set	Expected Packet Rate	UINT	Default 2500 msec
12	Get/Set	Timeout Action	USINT	(3)
13	Get	Prod. Path Length	USINT	6
14	Get	Produced Path	STRUCT of	
		Log. Seg., Class	USINT	0x20
		Class Number	USINT	0x04
		Log.Seg., Instance	USINT	0x24
		Instance Number	USINT	0x01
		Log.Seg., Attribute	USINT	0x30
		Attribute Number	USINT	0x03
15	Get	Cons. Path Length	USINT	6
16	Get	Consumed Path	STRUCT of	
		Log. Seg., Class	USINT	0x20
		Class Number	USINT	0x04
		Log.Seg., Instance	USINT	0x24
		Instance Number	USINT	0x02
		Log.Seg., Attribute	USINT	0x30
		Attribute Number	USINT	0x03
17	Get	Production Inhibit	UINT	0

Common Services

Service Code	Class	Instance	Service Name
05 (0x05)	Yes	Yes	Reset
14 (0x0E)	Yes	Yes	Get_Attribute_Single
16 (0x10)	No	Yes	Set_Attribute_Single

(1) Connection States:

0 = non-existent

1 = configuring

3 = established
4 = timed out

(2) Connection ID's:

Connection 1 Produced Connection ID: 10xxxxxx011
Connection 1 Consumed Connection ID: 10xxxxxx100
Connection 2 Produced Connection ID: 01111xxxxxx
Connection 2 Consumed Connection ID: 10xxxxxx101

xxxxxx = Node Address.

(3) Watch Dog TimeOut Activity:

0 = Timeout (Explicit Messaging default)
1 = Auto Delete
2 = Auto Reset (I/O Message default)
3 = Deferred Delete

(4) If no data is available during the poll response a 0 length (null) packet is returned.

User Defined Communicator Object Class Code: 64 (0x40)

The Communicator Object model links the RS485 P+F IDENT-I System P Communicator to DeviceNet. There is 1 instances for this object.

Display Class Attributes

Attribute	Access	Name	Type	Value
1	Get	Revision	UINT	1
2	Get	Max Object Instance	UINT	1
6	Get	Max Class Identifier	UINT	15
7	Get	Max Instance Attribute	UINT	1

Communicator Object, Instance 1 Attributes

Attribute	Access	Name	Type	Value
3	Get/Set	E_Length	USINT	(3)
4	Get/Set	E_HeadNo	Struct of	(4)
5	Get/Set	Command + Head# + Params	Struct of	(5)
		Command	USINT	
		Head number (in hex)	USINT	
		Paramater1	USINT	
		Paramater2	USINT	
6	Get/Set	Tag Type (in hex)	USINT	(6)
7	Get/Set	Initializing Wait Time	USINT	(7)
8	Get/Set	Tag Time Out (used with 'ci')	USINT	(8)
9	Get	Version Date	See notes	(9)
10	Get/Set	IPG-G4-B7 Error Code	USINT	(10)
11	Get	Head State	USINT	(11)
12	Get/Set	Mode Byte (bit mapped)	USINT	(12)
15	Get/Set	Diagnostic Byte (bit mapped)	USINT	(15)
16	Get/Set	Poll retention time (msec)	USINT	(16)
17	Get/Set	Write Response time (msec)	**UINT**	(17)
18	Get/Set	Read Response time (msec)	USINT	(18)

(3) E_Length: the commands that are issued to the Communicator through this class (explicitly) will use the value set here as the length that is sent to the Communicator. Currently values of zero and one are allowed. The value is in hex

(4) E_HeadNo: the commands that are issued to the Communicator through this class (explicitly) will use the value set here as the head number that is used to determine what Communicator the command is meant for. Head numbers 1 and 2 are on port-1, head numbers 3 and 4 are on port-

2. Any number given that is greater than 4 will be assumed to be a broadcast command that is sent to all available heads.

(5) Command + Head # + Params: commands that require address, but not data, may be set explicitly with this attribute. See the assembly object for the structure of the parameters. This attribute will use the E_Length for the length. The format is as follows:

command	HeadNo (hex)	Param1	Param2
1	2	3	4

(6) TagType: IPC02 = 2
IPC03 = 3
IPC10= 10 (0A hex)

(7) Initializing Wait Time: the time (in msec's) the IPG-G4-B7 will wait for a Communicator to respond when attempting to establish communicate to each Communicator. For example a value of 14 hex = 20 decimal = 200 msec wait time.

(8) TagTime Out: the time out value for the Communicator is a three-digit number. The first digit may be set to a value of 0 to 9 (in hex), which corresponds to a value of 100 to 900msec. For example 01 hex = 100 msec, 02 hex = 200 msec…09 hex = 900 msec. This is sent in the 'ci' command at startup.

(9) Version date: the version date of the Communicator firmware may be gotten as an eight-byte value in ASCII format. An example is 14.01.98, which represents January 14th, 1998. The version command must be given before the information is available.

(10) The IPG-G4-B7 will record an error code from the following list. The error may be cleared by setting the attribute:

Error Codes
0 = No error.
1 = RS485 Response Buffer full.
2 = The node adapter has timed out when waiting for a response from the Communicator.
3 = HeadNoError- the node adapter received a head number other than 1-4.
4 = Unknown response- the node adapter received a response that it did not understand.
5 = Check Sum Error – if a check sum error occurs it is latched here.

(11) Head State: the node adapter internal state of the head may be read using this attribute. These are the states that the node adapter assigns for each head and are listed in the State table.

(12) Mode Byte: the mode byte is retained in EEPROM and is a bit-mapped value retained when the IPG-G4-B7 is powered down. The default value is zero. The format is:

Mode Byte

X	x	x	x	ChkSumFlash	ReqRst	RedLedEn	ChkSumDisable
7	6	5	4	3	2	1	0

If bit-1 is set, Check Sum checking will be disabled and a #CR is sent instead of the check sum and ETX. If bit-1 is set, The RED LED's are enabled for errors on the both communication channels these are the same LED's that flash green when a character is transmitted and received. If enabled, the can be cleared by explicitly setting the status byte or power cycling the unit. If bit-2 is set, zeros are not placed in the response when a new command is sent. If bit-3 is set the RED LED's will not flash if a check sum error is received.

(13), (14) – Skipped on purpose.

(15) Diagnostic Byte: a bit-mapped debugging byte stored in RAM only. The default is zero and any other values placed in it will not be retained when the unit is powered down.

(16) Poll Retention Time: this value determines the minimum time a response from a Communicator is left on the DeviceNet response. The value is set in msec.

(17) Write Response Time: this is the time that the IPG-G4-B7 will wait for a 485 response form the Communicator (head) after a write command is issued to it. The value is a two byte (one word) value in msec.

(18) Read Response Time: this is the time that the IPG-G4-B7 will wait for a 485 response form the Communicator (head) after a write command is issued to it. The time is a two byte (one word) value in msec.

Common Services

Service Code	**Class**	**Instance**	**Service Name**
5 (0x05)	No	Yes	*Reset
14 (0x0E)	Yes	Yes	Get_Attribute_Single
16 (0x10)	No	Yes	Set_Attribute_Single

*A Reset to the Communicator object may be done with the following data:

No data	=	will reinitialize the Communicators only
0	=	will reinitialize the Communicators only
1	=	reinitializes/sets InitTime, TimeOut, and ModeByte to factory defaults

2 = reinitializes/sets all Communicators to IPC02
3 = reinitializes/sets all Communicators to IPC03

Further information about the polling and assembly of Tag data is provided in the Assembly object, class-04.

Command Table

Command	Command Code (hex)	<Number of Double Word>	Head Number (bit pattern)	Parameter		Data				
				1	2	1	2	3	4	5
ve	03	-	000 to 1xx	-	-	-	-	-	-	-
qu	02	-	000 to 1xx	-	-	-	-	-	-	-
rs	16	-	000 to 1xx	-	-	-	-	-	-	-
ct	04	-	000 to 1xx	Low Byte	High Byte	-	-	-	-	-
sf	01	-	000 to 1xx	-	-	-	-	-	-	-
af	08	-	000 to 1xx	-	-	-	-	-	-	-
bf	09	-	000 to 1xx	-	-	-	-	-	-	-
ef	1D	-	000 to 1xx	-	-	-	-	-	-	-
sr	10	1 or 0	000 to 1xx	ADR High	ADR Low	-	-	-	-	-
ar	20	1 or 0	000 to 1xx	ADR High	ADR Low	-	-	-	-	-
br	30	1 or 0	000 to 1xx	ADR High	ADR Low	-	-	-	-	-
er	19	1 or 0	000 to 1xx	ADR High	ADR Low	-	-	-	-	-
sw	40	1	000 to 1xx	ADR High	ADR Low	Data4	Data3	Data2	Data1	-
aw	50	1	000 to 1xx	ADR High	ADR Low	Data4	Data3	Data2	Data1	-
bw	60	1	000 to 1xx	ADR High	ADR Low	Data4	Data3	Data2	Data1	-
ew	1A	1	000 to 1xx	ADR High	ADR Low	Data4	Data3	Data2	Data1	-
sx	1F	5	000 to 1xx	FixType1	FixType2	Data4	Data3	Data2	Data1	Data0
ax	64	5	000 to 1xx	FixType1	FixType2	Data4	Data3	Data2	Data1	Data0
bx	65	5	000 to 1xx	FixType1	FixType2	Data4	Data3	Data2	Data1	Data0
ex	24	5	000 to 1xx	FixType1	FixType2	Data4	Data3	Data2	Data1	Data0

Internal IPG-G4-B7 States

These are the internal states of each Communicator. This information is not usually needed for normal operations and is intended primarily for debugging. These states are obtained from the state attribute in the Communicator Object.

```
NotInit         equ     0
NotDet9600      equ     1       ; not detected
Init9600        equ     2       ; detected at 9600
InitReset       equ     3       ; Reset detected at 9600
Init19200       equ     4       ; detected at 19200
NotDet          equ     equ     5       ; not detected
Ready           equ     6       ; detected and available at 19200
ComFaulted      equ     7       ; faulted after detection

state to tell xmitter to send cmd
Up_ct           equ     8
Up_ve           equ     9
Up_no           equ     10
Up_qu           equ     11
Up_ci           equ     12
Up_rs           equ     13
Up_qu_nr        equ     14      ; quit with out updating DeviceNet response

Up_sf           equ     16
Up_af           equ     17
Up_bf           equ     18
Up_sr           equ     19
Up_ar           equ     20
Up_br           equ     21
Up_sw                   equ     22
Up_aw                   equ     23
Up_bw                   equ     24
Up_ef           equ     26
Up_er           equ     27
Up_ew                   equ     28
Up_sx           equ     29
Up_ax           equ     30
Up_bx           equ     31
Up_ex           equ     32

command states  - state after cmd is sent, waiting for response
Cmd_ct          equ     40
Cmd_ve          equ     41
Cmd_no          equ     42
Cmd_qu          equ     43
```

```
Cmd_ci      equ   44
Cmd_qu_nr   equ   45     ; quit with out updating DeviceNet response
Cmd_sr      equ   48
Cmd_ar      equ   49
Cmd_br      equ   50
Cmd_sf      equ   51
Cmd_af      equ   52
Cmd_bf      equ   53
Cmd_sw      equ   54
Cmd_aw      equ   55
Cmd_bw      equ   56
Cmd_rs      equ   57     ; not used goes to Rsp_rs
Cmd_ef      equ   58
Cmd_er      equ   59
Cmd_ew      equ   60

Cmd_sx      equ   61
Cmd_ax      equ   62
Cmd_bx      equ   63
Cmd_ex      equ   64

command states  - state after 1st response- need to issue 'gd'
Rsp_sr      equ   70     ; use to start at 64 changed 2-24-00
Rsp_ar      equ   71
Rsp_br            equ   72

Rsp_sf      equ   73
Rsp_af      equ   74
Rsp_bf            equ   75

Rsp_sw      equ   76
Rsp_aw      equ   77
Rsp_bw      equ   78

Rsp_rs      equ   79
Rsp_ve      equ   80
Rsp_ci      equ   81

Rsp_ef      equ   82
Rsp_er      equ   83
Rsp_ew      equ   84

Rsp_sx            equ   85
Rsp_ax      equ   86
Rsp_bx      equ   87
Rsp_ex      equ   88
```

```
command states  - after 'gd' issued
gd_sr           equ     128
gd_ar           equ     129
gd_br           equ     130
gd_sf           equ     131
gd_af           equ     132
gd_bf           equ     133
gd_sw           equ     134
gd_aw           equ     135
gd_bw           equ     136
gd_rs           equ     137

gd_ef           equ     140
gd_er           equ     141
gd_ew           equ     142

gd_sx           equ     141
gd_ax           equ     142
gd_bx           equ     143
gd_ex           equ     143
```