Citect for Windows, Version 6.xx, 7.xx, 8.xx

JControl driver, User information

begcomm Communication AB
Brunnehagen 109
S-417 42 Göteborg
Phone +46 70 258 50 01
Email info@begcomm.com
www.begcomm.com

Contents

1. User information

1.1 Device Application notes

1.1.1 Device Manufacturer

	Detail
Manufacturer	Johnson Controls
Device name	JControl
Communications method	Serial RS232 direct or RS485 multidrop via converter IU-9100. You can also communicate via a TCP or UDP gateway to RS232/RS485. The driver acts as master.

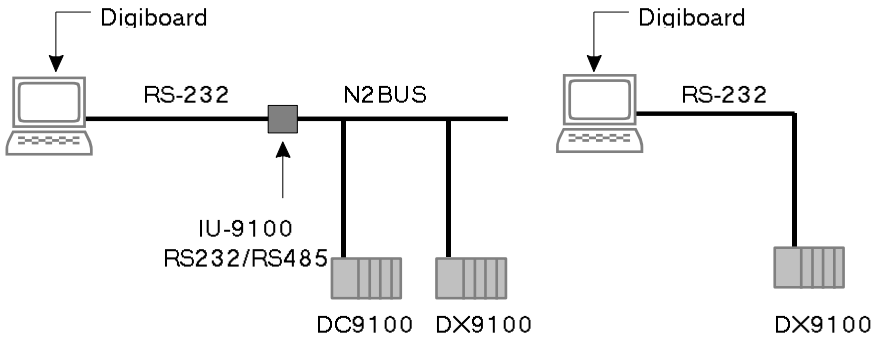
1.1.2 Device Definition

Johnson Controls has informed that the protocol is the same for all units below. The only difference is that the item addresses for similar signals are different between the models.

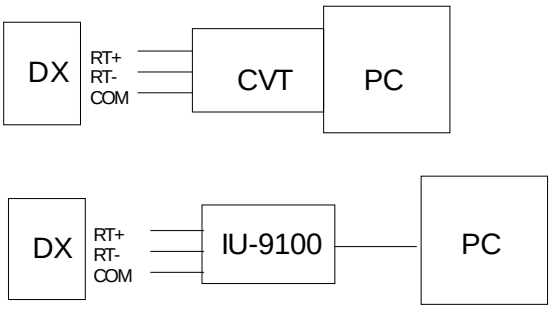
Models
DX-9100
DC-9100
XT/XP-9100
TC-9100
SC-9100
DR-9101

1.2 Communications/Hardware Configuration

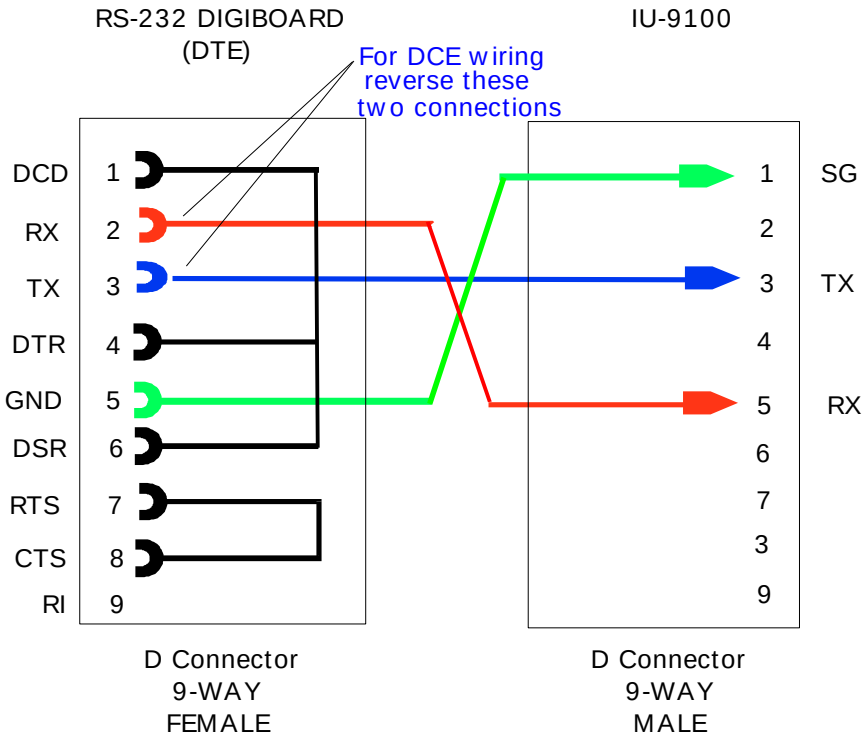
The serial method of communication to the Johnson family of intelligent digital and analogue input/output controllers, uses the JCONTROL protocol. Using this method you can connect to single controllers or to multiple controllers as in the following diagrams:

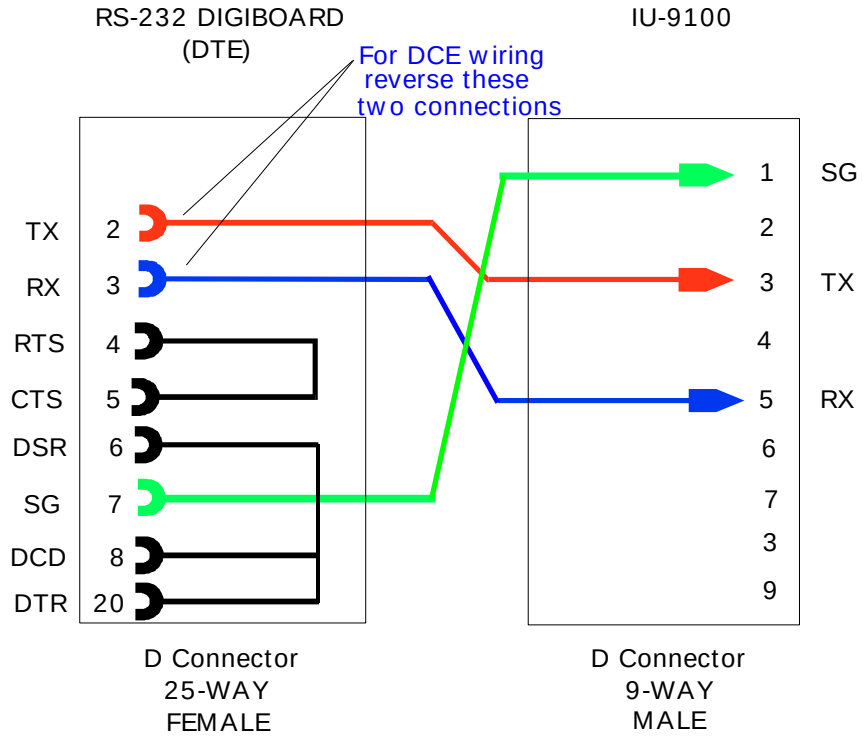


1.2.1 Connection details for download/upload of DX-9100

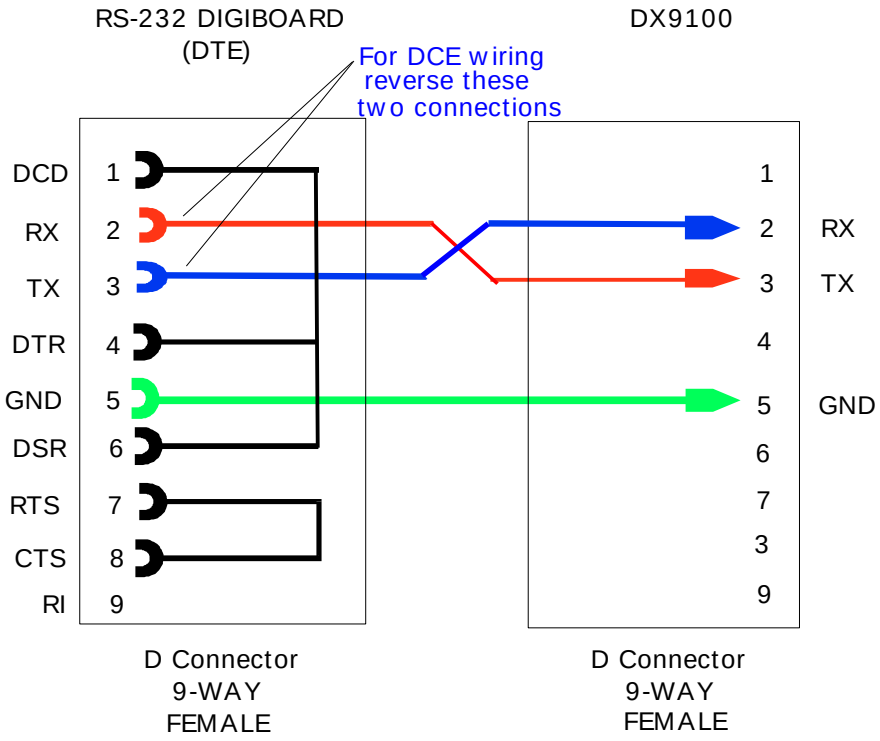


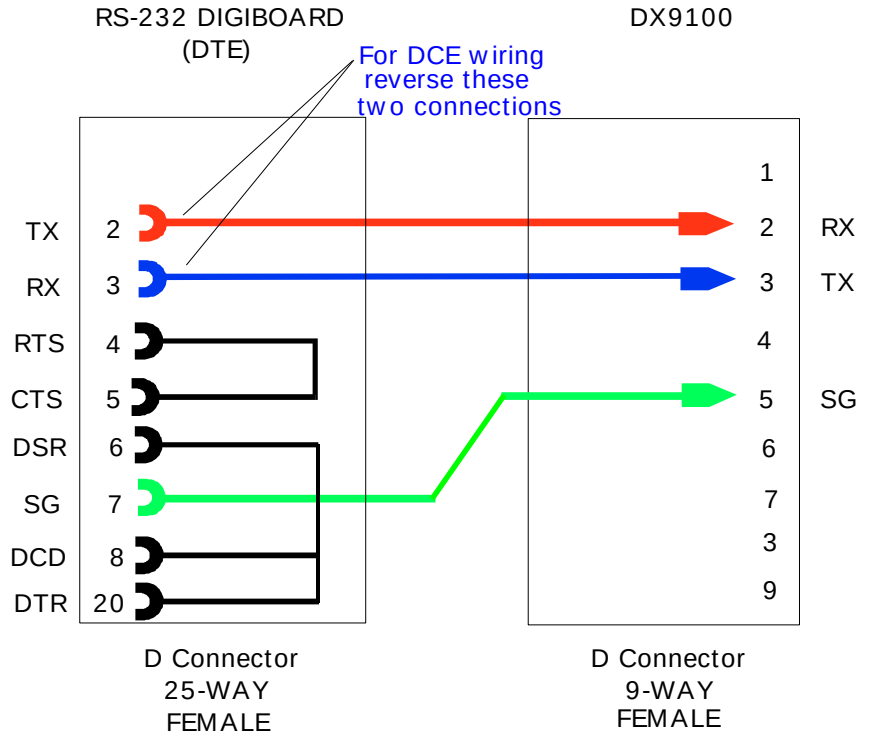
1.2.2 Wiring Diagram IU-9100 to PC Connection





1.2.3 Wiring Diagram Direct Download DX9100





1.2.4 I/O Device Settings

The Johnson factory sets all communications parameters in the units.

1.3 Special Requirements

The JCONTROL driver for Johnson Control equipment deliveries by begcomm Communication AB, Brunnehagen 109, 417 42 Göteborg, Sweden. All copyrights and licenses belong to begcomm Communication AB.

1.4 Maximum Request Length

The driver works with single items. The item structure especially in the DX series is not good for block readings.

2. Communication forms

2.1 Serial communication

2.1.1 Boards form

Field	Default	Allowable values
Board Name	This field is user defined.	
Board Type	COMX	
Address	0	
I/O Port	BLANK	
Interrupt	BLANK	
Special Opt	BLANK	
Comment	This field is user defined and is not used by the driver.	

2.1.2 Ports form

Field	Default	Allowable values
Port Name	This field is user defined.	
Port number		
Board name	Refers to the board previously defined in ‘boards’ form.	
Baud rate	9600	
Data bits	8	
Stop bits	1	
Parity	NONE	
Special Opt	BLANK	
Comment	This field is user defined and is not used by the driver.	

2.2 Ethernet communication

2.2.1 Boards form

Field	Default	Allowable values
Board Name	This field is user defined.	
Board Type	TCPIP	
Address	0	
I/O Port	BLANK	
Interrupt	BLANK	
Special Opt	BLANK	
Comment	This field is user defined and is not used by the driver.	

2.2.2 Ports form for TCP

Field	Default	Allowable values
Port Name	This field is user defined.	
Port number		
Board name	Refers to the board previously defined in ‘boards’ form.	
Baud rate	BLANK	Not used by the driver
Data bits	BLANK	Not used by the driver
Stop bits	BLANK	Not used by the driver
Parity	BLANK	Not used by the driver
Special Opt	-i192.168.100.136 –p10001 -t	
Comment	This field is user defined and is not used by the driver.	

2.2.3 Ports form for UDP

Field	Default	Allowable values
Port Name	This field is user defined.	
Port number		
Board name	Refers to the board previously defined in ‘boards’ form.	
Baud rate	BLANK	Not used by the driver
Data bits	BLANK	Not used by the driver
Stop bits	BLANK	Not used by the driver
Parity	BLANK	Not used by the driver
Special Opt	-i192.168.100.136 –p10001 -u	
Comment	This field is user defined and is not used by the driver.	

2.3 I/O Devices form

2.3.1 JControl N2

Field	Default	Allowable values
Name	This field is user defined and is not used by the driver.	
Number	Must be unique.	
Address	241	The range is between 0 and 255
Protocol	JCONTROL	
Port name	Refers to the port previously defined in ‘ports’ form.	
Comment	This field is user defined and is not used by the driver.	

2.3.2 JControl N2Open

Field	Default	Allowable values
Name	This field is user defined and is not used by the driver.	
Number	Must be unique.	
Address	241	The range is between 0 and 255
Protocol	JCONTROL2	
Port name	Refers to the port previously defined in ‘ports’ form.	
Comment	This field is user defined and is not used by the driver.	

3. Variable Types

3.1 Data types JControl

IO Device Type	Citect data format	Citect data types	Description/Special Usage/Limitations/ Valid Ranges
Single Item	SIN*x*	REAL	Read / Write. Johnson Control special floating point number (16 bit).
Single Item	SIB*x*	BYTE	Read / Write.
Single Item	SIB*x.y*	DIGITAL	Read / Write.
Single Item	SIW*x*	UINT	Read / Write.
Single Item	SIW*x.z*	DIGITAL	Read / Write.
Single Item	SIL*x*	LONG	Read / Write. Allowed value is maximum 2,147,483,647
Real Time Clock	RTC*t*	BCD	Read / Write.
Daylight Savings	DS*d*	BCD	Read / Write.
Exception Days	ED*e*	BCD	Read / Write.
Time Schedule	TS*m.w*	BCD	Read / Write.
Time Schedule	TS*m.w*	UINT	Read / Write.

Where:

x	Item number, 0 - FFFF
y	Bit number, 1 - 8
z	Bit number, 1 - 16
t	Item number, 0 - 7
d	Item number, 0 - 1
e	Item number, 0 - 3B
m	Time schedule module, 0 - 7
w	Item in time schedule module, 0 - 1C

3.2 Examples data types JControl

3.2.1.1 Single Item

Data Type BYTE
Address SIB0
Comment Item for Device Model (All models)

Data Type UINT
Address SIW21
Comment Status Word 1 in DC9100 (Digital input)

Data Type DIGITAL
Address SIW21.1
Comment Logic Input 1 (DC9100)

Data Type DIGITAL
Address SIB6.1
Comment Digital Input 1 in (DX9100)

Data Type REAL
Address SIN4C7
Comment Analog Input Value 1 (DX9100)

Data Type LONG
Address SILD
Comment DI1 Pulse Counter (DX9100)

3.2.1.2 Real Time Clock

Data Type BCD
Address RTC6
Comment Real Time Clock Minutes (DX9100)

3.2.1.3 Daylight Savings

Data Type BCD
Address DS0
Comment Daylight Savings Start date (DX9100)

3.2.1.4 Exception Days

Data Type BCD
Address ED2
Comment Exception Days Start Date #02 (DX9100)

3.2.1.5 Time Schedule

Data Type BCD
Address TS1.1B
Comment Time Schedule module 1, End Time Event #08 (DX9100)

Data Type UINT
Address TS1.1C
Comment Time Schedule module 1, Enable Days Event #08 (DX9100)

3.3 Data types for JControl N2Open

IO Device Type	Region	Tag type	Citect data format	Citect data types	Description
Analog Input	1	Current Value	AI*x*	REAL	Read
Analog Input	1	Object Configuration	AIC*x*	BYTE	Read / Write
Analog Input	1	Object Configuration	AIC*x.z*	DIGITAL	Read
Analog Input	1	Object Status	AIS*x*	BYTE	Read
Analog Input	1	Object Status	AIS*x.z*	DIGITAL	Read
Analog Input	1	Low Alarm Limit	AILAL*x*	REAL	Read / Write
Analog Input	1	Low Warning Limit	AILWL*x*	REAL	Read / Write
Analog Input	1	High Warning Limit	AIHWL*x*	REAL	Read / Write
Analog Input	1	High Alarm Limit	AIHAL*x*	REAL	Read / Write
Analog Input	1	Differential	AID*x*	REAL	Read / Write
Analog Input	1	Override value	AIO*x*	REAL	Write
Analog Input	1	Override release	AIR*x*	DIGITAL	Write
Binary Input	2	Object Configuration	BIC*x*	BYTE	Read / Write
Binary Input	2	Object Configuration	BIC*x.z*	DIGITAL	Read
Binary Input	2	Object Status	BIS*x*	BYTE	Read
Binary Input	2	Object Status	BIS*x.z*	DIGITAL	Read
Binary Input	2	Override value	BIO*x*	BYTE	Write
Binary Input	2	Override release	BIR*x*	DIGITAL	Write
Analog Output	3	Current Value	AO*x*	REAL	Read
Analog Output	3	Object Configuration	AOC*x*	BYTE	Read / Write
Analog Output	3	Object Configuration	AOC*x.z*	DIGITAL	Read
Analog Output	3	Object Status	AOS*x*	BYTE	Read
Analog Output	3	Object Status	AOS*x.z*	DIGITAL	Read
Analog Output	3	Override value	AOO*x*	REAL	Write
Analog Output	3	Override release	AOR*x*	DIGITAL	Write
Binary Output	4	Object Configuration	BOC*x*	BYTE	Read / Write
Binary Output	4	Object Configuration	BOC*x.z*	DIGITAL	Read
Binary Output	4	Object Status	BOS*x*	BYTE	Read
Binary Output	4	Object Status	BOS*x.z*	DIGITAL	Read
Binary Output	4	Minimum On-Time(sec)	BOMINON*x*	INT	Read / Write
Binary Output	4	Minimum Off-Time(sec)	BOMINOFF*x*	INT	Read / Write
Binary Output	4	Maximum Cycles/Hour	BOMAXCH*x*	INT	Read / Write
Binary Output	4	Override value	BOO*x*	BYTE	Write
Binary Output	4	Override release	BOR*x*	DIGITAL	Write
Internal Float value	5	Current Value	ADF*x*	REAL	Read / Write
Internal Float value	5	Object Status	ADFS*x*	BYTE	Read
Internal Float value	5	Object Status	ADFS*x.z*	DIGITAL	Read
Internal Float value	5	Override value	ADFO*x*	REAL	Write
Internal Float value	5	Override release	ADFR*x*	DIGITAL	Write
Internal Integer value	6	Current Value	ADI*x*	INT	Read / Write
Internal Integer value	6	Current Value	ADI*x.y*	DIGITAL	Read
Internal Integer value	6	Object Status	ADIS*x*	BYTE	Read
Internal Integer value	6	Object Status	ADIS*x.z*	DIGITAL	Read
Internal Integer value	6	Override value	ADIO*x*	INT	Write
Internal Integer value	6	Override release	ADIR*x*	DIGITAL	Write
Internal Byte value	7	Current Value	BYT*x*	BYTE	Read / Write
Internal Byte value	7	Current Value	BYT*x.z*	DIGITAL	Read
Internal Byte value	7	Object Status	BYTS*x.*	BYTE	Read
Internal Byte value	7	Object Status	BYTS*x.z*	DIGITAL	Read
Internal Byte value	7	Override value	BYTO*x*	BYTE	Write
Internal Byte value	7	Override release	BYTR*x*	DIGITAL	Write
Synch Time Command (*Not implemented in this version)*			TIME	INT	Write

3.4 Old N2Open data types for backward compatibility

IO Device Type	Region	Tag type	Citect data format	Citect data types	Description
Internal Float value	5	Current Value	IPADF*x*	REAL	Read / Write
Internal Float value	5	Override value	OIPADF*x*	REAL	Write
Internal Integer value	6	Current Value	IPADI*x*	INT	Read / Write
Internal Integer value	6	Current Value	IPADI*x.y*	DIGITAL	Read
Internal Integer value	6	Override value	OIPADI*x*	INT	Write
Internal Byte value	7	Current Value	IPBD*x*	BYTE	Read / Write
Internal Byte value	7	Current Value	IPBD*x.z*	DIGITAL	Read
Internal Byte value	7	Override value	OIPBD*x*	BYTE	Write

Where:

x Item number, 1 - 255
y Bit number, 0 - 15
z Bit number, 1 - 8

3.5 Examples data types N2Open

3.5.1 Analog Input (Region 1)

3.5.1.1 Current Value

Data Type REAL
Address AI1
Comment

3.5.1.2 Object Configuration

Data Type BYTE
Address AIC1
Comment

Data Type DIGITAL
Address AIC1.1
Comment

3.5.1.3 Object Status

Data Type BYTE
Address AIS1
Comment

Data Type DIGITAL
Address AIS1.1
Comment

3.5.1.4 Low Alarm Limit

Data Type REAL
Address AILAL1
Comment

3.5.1.5 Low Warning Limit

Data Type REAL
Address AILWL1
Comment

3.5.1.6 High Warning Limit

Data Type REAL
Address AIHWL1
Comment

3.5.1.7 High Alarm Limit

Data Type REAL
Address AIHAL1
Comment

3.5.1.8 Differential

Data Type REAL
Address AID1
Comment

3.5.1.9 Override value

Data Type REAL
Address AIO1
Comment

3.5.1.10 Override release

Data Type DIGITAL
Address AIR1
Comment: Write value 1 for reset

3.5.2 Binary Input (Region 2)

3.5.2.1 Object Configuration

Data Type BYTE
Address BIC1
Comment

Data Type DIGITAL
Address BIC1.1
Comment

3.5.2.2 Object Status

Data Type BYTE
Address BIS1
Comment

Data Type DIGITAL
Address BIS1.1
Comment

3.5.2.3 Override value

Data Type BYTE
Address BIO1
Comment

3.5.2.4 Override release

Data Type DIGITAL
Address BIR1
Comment: Write value 1 for reset

3.5.3 Analog Output (Region 3)

3.5.3.1 Current Value

Data Type REAL
Address AO1
Comment

3.5.3.2 Object Configuration

Data Type BYTE
Address AOC1
Comment

Data Type DIGITAL
Address AOC1.1
Comment

3.5.3.3 Object Status

Data Type BYTE
Address AOS1
Comment

Data Type DIGITAL
Address AOS1.1
Comment

3.5.3.4 Override value

Data Type BYTE
Address AOO1
Comment

3.5.3.5 Override release

Data Type DIGITAL
Address AOR1
Comment: Write value 1 for reset

3.5.4 Binary Output (Region 4)

3.5.4.1 Object Configuration

Data Type BYTE
Address BOC1
Comment

Data Type DIGITAL
Address BOC1.1
Comment

3.5.4.2 Object Status

Data Type BYTE
Address BOS1
Comment

Data Type DIGITAL
Address BOS1.1
Comment

3.5.4.3 Minimum On-Time(sec)

Data Type INT
Address BOMINON1
Comment

3.5.4.4 Minimum Off-Time(sec)

Data Type INT
Address BOMINOFF1
Comment

3.5.4.5 Maximum Cycles/Hour

Data Type INT
Address BOMAXCH1
Comment

3.5.4.6 Override value

Data Type BYTE
Address BOO1
Comment

3.5.4.7 Override release

Data Type DIGITAL
Address BOR1
Comment: Write value 1 for reset

3.5.5 Internal Float values (Region 5)

3.5.5.1 Current Value

Data Type REAL
Address ADF1
Comment

3.5.5.2 Object Status

Data Type BYTE
Address ADFS1
Comment

Data Type DIGITAL
Address ADFS1.1
Comment

3.5.5.3 Override value

Data Type REAL
Address ADFO1
Comment

3.5.5.4 Override release

Data Type DIGITAL
Address ADFR1
Comment: Write value 1 for reset

3.5.6 Internal Integer values (Region 6)

3.5.6.1 Current Value

Data Type INT
Address ADI1
Comment

Data Type DIGITAL
Address ADI1.1
Comment

3.5.6.2 Object Status

Data Type BYTE
Address ADIS1
Comment

Data Type DIGITAL
Address ADIS1.1
Comment

3.5.6.3 Override value

Data Type INT
Address ADIO1
Comment

3.5.6.4 Override release

Data Type DIGITAL
Address ADIR1
Comment: Write value 1 for reset

3.5.7 Internal Byte values (Region 7)

3.5.7.1 Current Value

Data Type BYTE
Address BYT1
Comment

Data Type DIGITAL
Address BYT1.1
Comment

3.5.7.2 Object Status

Data Type BYTE
Address BYTS1
Comment

Data Type DIGITAL
Address BYTS1.1
Comment

3.5.7.3 Override value

Data Type BYTE
Address BYTO1
Comment

3.5.7.3.1 Override release

Data Type DIGITAL
Address BYTR1
Comment: Write value 1 for reset

3.6 Supported N2Open Commands

3.6.1 Read Analog Input Command

The read analog input command is used to retrieve any of the thirteen (13) attributes pertaining to specified analog input object.

3.6.1.1 Object Status

Bit	Description	Description
0	Reliable (0) / unreliable (1)	
1	Override active (1)	
2	Out of range – high (1)	
3	Out of range – low (1)	
4	COS status	
5	COS status	
6	COS status	
7	unused	
Where bits 4, 5, 6 are	000	normal
Where bits 4, 5, 6 are	001	JCI use only
Where bits 4, 5, 6 are	010	Not available
Where bits 4, 5, 6 are	011	Low warning
Where bits 4, 5, 6 are	100	Low alarm
Where bits 4, 5, 6 are	101	High warning
Where bits 4, 5, 6 are	110	High warning

Note: When either attribute 2 or 3 is requested, the N2 device responds with both attributes 2 and 3.

3.6.2 Read Binary Input Command

The read binary input command is used to retrieve any of the four (4) attributes of the specified binary input object.

3.6.2.1 Object Status

Bit	Description
0	Reliable (0) / unreliable (1)
1	Override active (1)
2	unused
3	unused
4	Normal (0) / alarm (1)
5	JCI use only
6	Current state
7	unused

3.6.3 Read Analog Output Command

The read analog output command is used to retrieve any of the five (5) attributes of the specified analog output object.

3.6.3.1 Object Status

Bit	Description
0	Reliable (0) / unreliable (1)
1	Override active (1)
2	JCI use only
3	JCI use only
4	unused
5	unused
6	unused
7	unused

Note: When either attribute 2 or 3 is requested, the N2 device responds with both attributes 2 and 3.

3.6.4 Read Binary Output Command

The read binary output command is used to retrieve any of the eight (8) attributes of the specified binary output object.

3.6.4.1 Object Status

Bit	Description
0	Reliable (0) / unreliable (1)
1	Override active (1)
2	unused
3	unused
4	JCI use only
5	JCI use only
6	unused
7	unused

3.6.5 Read Internal Parameter Command

The read internal parameter command retrieves the value attribute of the internal parameter objects.

3.6.5.1 Object Status

Bit	Description
0	Reliable (0) / unreliable (1)
1	Override active (1)
2	unused
3	unused
4	unused
5	unused
6	unused
7	unused

Note: When either attribute 1 or 2 is requested, the N2 device responds with both attributes 1 and 2.

3.6.6 Write Analog Input Command
The write analog input command is used to change an attribute of the specified analog input object (with the exception of attributes 2 and 3, object status, and current value).

3.6.7 Write Binary Input Command
The write binary input command is used to change an attribute of the specified binary input object (with the exception of attributes 2 and 4, object status, and accumulator value).

3.6.8 Write Analog Output Command
The write analog output command is used to change an attribute of the specified analog output object (with the exception of attributes 2 and 3, object status, and current value).

3.6.9 Write Binary Output Command
The write binary output command is used to change an attribute of the specified binary output object (with the exception of attribute 2, object status).

3.6.10 Write Internal Parameter Command
The write internal parameter is used to change the value attribute of internal parameter objects.

3.6.11 Override Analog Input Command
The override analog input command is used to send an override value to the analog input object to be used in place of the normal analog input. The override value becomes the object's current value. Generally, the override value remains active as long as the master continues polling the N2 device. In standalone operation, or should the polling stop, the object reverts to the object's input value after a specified period (normally 10 minutes).

3.6.12 Override Binary Input Command
The override binary input command is used to send an override value to the binary input object to be used in place of the current binary state. The override value becomes the object's current value. Generally, the override value remains active as long as the master continues polling the N2 device. In standalone operation, or should the polling stop, the object reverts to the object's input value after a specified period (normally 10 minutes).

3.6.13 Override Analog Output Command
The override output command is used to send an override value to the analog output object to be used in place of its current value attribute. The override value becomes the object's current value. Generally, the override value remains active as long as the master continues polling the N2 device. In standalone operation, or should the polling stop, the object reverts to the object's input value after a specified period (normally 10 minutes).

3.6.14 Override Binary Output Command
The override binary output command is used to send an override value to the binary output object to be used in place of its current state. The override value becomes the object's current value. If an attempt is made to override a binary output that is not allowed by the N2 device, an error will occur. Generally, the override value remains active as long as the master continues polling the N2 device. In standalone operation, or should the polling stop, the object reverts to the object's input value after a specified period (normally 10 minutes).

3.6.15 Override Internal Parameter Command
The override internal parameter command is used to change any internal parameter value. The override value becomes the objects current value.

3.6.16 Override Release Request
This message commands the N2 device to release a previously overridden data value. Once a value has been released the local value is be used.

3.7 Hints, Tips, and Frequently asked questions

- Take care to use bit writing if you have a PLC program running in the DUC. The driver is first reading the word and masking the bit and afterward sending it back to the unit. If the PLC program make a change in the same word during the time the driver is manipulating this word the PLC changing can be overwritten.
- The Johnson Controls floating point values uses only sixteen bit. Use therefore maximum only one decimal in the presentation.
- If you want to set output 3 to 8 in DX-9100 direct from Citect you have to first enable the output. You have also to put the bit for Supervisory System Active SUP W(16) to active state and refreshes the DX9100 with 120 minutes time-out. The best way to do this is from Cicode. This is a watchdog function, which belongs to the security system. The project developer has the responsibility for this procedure, not the driver itself.
- You can change the Counters size in DX-9100 between 16 bits or 32 bits direct from Citect with B(4) in DXS1.
- With tcpip it can sometimes be good to increase the timeout parameter to a very high value e.g. 12000. The reason for this is that if you should have problem with disturbances the tcpip part can make retries before Citect make a timeout. This is typical for protocols without a synchronize mechanism

4. Driver reference

	Detail
Driver name	JCONTROL
Maximum array size[1]	256

4.1 Driver generated error codes

Driver Error Code (Hexadecimal)	Mapped to (Generic Error label)	Meaning of Error Code
101	**GENERIC_INVALID_COMMAND**	Not existing command mode
102	**GENERIC_INVALID_DATA**	Data not matching the item or function type
181	**GENERIC_BAD_PARAMETER**	Not existing item or function
182	**GENERIC_CMD_CANCELED**	Temporarily impossible to access the item
183	**GENERIC_ACCESS_VOILATION**	Not programmable item
184	**GENERIC_INVALID_DATA**	Table programmed with illegal items
185	**GENERIC_INVALID_DATA**	Trend programmed with illegal item
186	**GENERIC_INVALID_COMMAND**	Invalid Functional Module
187	**GENERIC_ADDRESS_RANGE_ ERROR**	Exceeding Addressing Range
188	**GENERIC_NO_RESPONSE**	Undefined Address after gate
189	**GENERIC_NO_RESPONSE**	No answer from Device after gate
18A	**GENERIC_ACCESS_VOILATION**	Password Protection Active
190	**GENERIC_HARDWARE_ERROR**	I=C-bus Error
191	**GENERIC_HARDWARE_ERROR**	Hardware not available
192	**GENERIC_BAD_PARAMETER**	Illegal item number
193	**GENERIC_INVALID_DATA**	Counters unreliable
194	**GENERIC_HARDWARE_ERROR**	Power supply unreliable

[1] Equivalent to ‘Maximum Request Length’

4.2 Error codes in N2Open

Defined codes	Description of error
00	Device has reset and is waiting for the “Identify Yourself” command.
01	Undefined Command: command not understood by addressed device.
02	Checksum error.
03	Input buffer overrun: message longer than maximum device expects to receive.
05	Data field error: size of message not correct for command type.
10	Invalid Data: one of the fields contains a value that is out of the expected range. A vendor device should return this code if a requested point does not exist or a commanded value is out of range.
11	Invalid command for data type: command not appropriate for this field or record.
12	Command not accepted: due to problems with the device, the command is ignored. The master should then use the Status Update Request command (described in section Status Update Request) to determine the problem.

4.3 Standard Parameters

Parameter	Default	Allowable values	Description
Block (bytes)	4		(Not used by the driver)
Delay (mS)	5		(30000/BR mS (BR=Baud Rate bit/sec))
MaxPending	2	2	
Polltime (mS)	0	0	
TimeOut (mS)*	1000	1000 to 32000	
Retry	1	0 to 3	
WatchTime (Sec)	30		

4.3.1 Block
The optimum size of data for a read request in bytes.

4.3.2 Delay
The period, in milliseconds, to wait between receiving a response and sending the next command.

4.3.3 MaxPending
The maximum number of pending commands that the driver holds ready for immediate execution on each channel.

4.3.4 Polltime
The polling time (in milliseconds). If the PollTime is set to 0 then the driver will work in interrupt mode.

4.3.5 Timeout
Specifies how many milliseconds to wait for a response before regarding a request as having failed

4.3.6 Retry
The number of times to retry a request which has timed out after no response.

4.3.7 WatchTime
The frequency that the driver uses to check the communications link to the I/O Device.

4.4 Driver Specific Parameters

Parameter	Default	Allowable values	Description
RegisterTcpip	0	0=serial 1=tcpip -1=messagebox	Obsolete from version 3.02.00.001

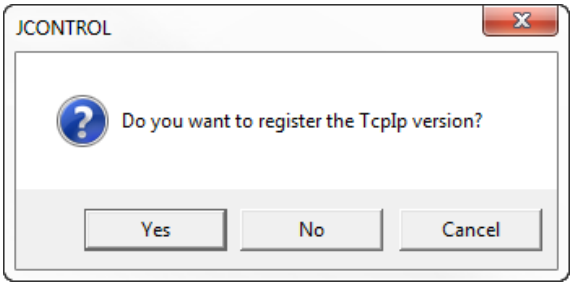
4.4.1 RegisterTcpip (Obsolete from version 3.02.00.001)

The Citect parameter RegisterTcpip (Obsolete from version 3.02.00.001) must be set to 1 or -1 if it shall be possible to use tcpip. Default is 0 for serial. If you are setting this parameter to -1 you will see the following messagebox

If you choose **Yes** you have to continue to fill in your tcpip code you have got from Beijer Electronics AB. If you choose **No** you have to put in the serial code or if you choose Cancel you can choose Demo mode. If you don't want to have this extra messagebox you can set the parameter [JCONTROL]

RegisterTcpip to 0 or to1 in Citect.ini there 0 forces your choice to serial and 1 to tcpip.

In runtime you can read out the type of registration in the statistics parameter 18 = 1 "TcpIp registered" or 19=1 "Serial registered" in the Kernel driver window.



4.5 Driver generated statistics

Number	Label	Description
0	Frames Transmitted	Number of frames transmitted
1	Frames Received	Number of frames received
2	Received Interrupts	Number of interrupts
3	Write Requests	Number of write requests
4	Read Requests	Number of read requests
5	Frame Accepted	Number of accepted received frames
6	Frame With Error	Number of received frames with errors
7	Read Modify Write	Number of errors in received frames
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18	TcpIp registered	Value = 1 when tcpip version is registered (Obsolete from version 3.02.00.001)
19	Serial registered	Value = 1 when serial version is registered (Obsolete from version 3.02.00.001)

4.6 Debug messages JControl

4.6.1 Initialisation

Item 0 is used for initialisation. In the answer it is possible to read what type of unit and version is used. In the example below the answer is 15 and 12 for the two multidropped units. (15H = DX9100 version 2.x and 12H = DC9100 version 2)

Sat Feb 19 21:17:12 2000 31:27:03.825 Transmit Length 14
>0180007A0405<d>
Sat Feb 19 21:17:12 2000 31:27:03.867 Receive Length 10
A15D51545<d>
Sat Feb 19 21:17:12 2000 31:27:03.955 Transmit Length 14
>0080007E0105<d>
Sat Feb 19 21:17:12 2000 31:27:03.987 Receive Length 10
A12DB124C<d>

4.6.2 Reading

In the example item 4C7 (C704) is readout and the item address is the value for analogue input 1.

Sat Feb 19 21:20:19 2000 31:30:10.903 Transmit Length 16
>0184C704892583<d>
Sat Feb 19 21:20:19 2000 31:30:10.942 Receive Length 12
A58554C06B4<d>

4.6.3 Writing

Sat Feb 19 21:25:56 2000 31:35:47.978 Transmit Length 20
>01D801050013A4684A<d>
Sat Feb 19 21:25:56 2000 31:35:48.015 Receive Length 2
A<d>

4.6.4 Error

Here we are trying to read a non-existing item 7F00. The answer is NAK and the errorcode is 81.

Sat Feb 19 21:32:27 2000 31:42:18.681 Transmit Length 16
>0184007FE9118A<d>
Sat Feb 19 21:32:27 2000 31:42:18.714 Receive Length 4
N81<d>
Sat Feb 19 21:32:27 2000 31:42:18.714 Error: Bad user parameters
READ 001d PORT2_BOARD1 IODev SIW7F00(32512) 1
Generic 000029 Driver 00000385 (0x00000181)

4.7 Debug messages JControl N2Open

4.7.1 Initialisation

“Identify Device Type Command” for unit 140 is used for initialisation.

Transmit Length 7 >8CFC1<d>
Receive Length 6 A1061<d>

4.7.2 Reading

Reading ADF23 for unit 140 has value 44.

Transmit Length 12 >8C151602AA<d>
Receive Length 14 A0042300000E9<d>

4.7.3 Writing

Writing value 52 to ADF23 unit 140.

Transmit Length 18 >8C251642500000D4<d>
Receive Length 2 A<d>

5. Analysis

5.1 IO Device Online Test

The item 0 is read out from the device at initialization. The driver makes the standard control about checksums, errors and so on. If the message is accepted the driver will tell Citect the device is online. In the answer from the unit is the device code laying. It's easy to see the type and revision for the unit. The digits after A in the protocol is the device code A15D51545<d> e.g. 15 stand for DX9100 Revision 2.x

Device Code	Description	Revision
01H	DR9100 - Room Controller	1
11H	DR9100 - Room Controller	2
02H	DC9100 - Plant Controller	1
12H	DC9100 - Plant Controller	2
03H	IM9100 - Modem Interface	1
04H	DO9100 - Digital Optimizer	1
05H	DX9100 - Digital Controller	1.x
15H	DX9100 - Digital Controller	2.x
25H	DX9100 - Digital Controller	3
06H	TC9100 - Room Controller	1
08H	XT9100 - Extension Module	1
18H	XTM905 - Extension Module	1

5.2 Message Structure JControl

5.2.1 Single item read message format:

">"	**ADR**	**CMD**	**ITEM**	**BCC**	**CHECKSUM**	**"Cr"**

Positive answer:

A	**DATA**	**BCC**	**CHECKSUM**	**"Cr"**

Negative answer:

N	**ERROR**	**"Cr"**

5.2.2 Single item write message format:

">"	**ADR**	**CMD**	**ITEM**	**DATA**	**BCC**	**CHECKSUM**	**"Cr"**

Positive answer:

A	**"Cr"**

Negative answer:

N	**ERROR**	**"Cr"**

5.2.3 Functional module read message format:

">"	**ADR**	**CMD**	**MOD**	**INDEX**	**BCC**	**CHECKSUM**	**"Cr"**

Positive answer:

A	**DATA**	**BCC**	**CHECKSUM**	**"Cr"**

Negative answer:

N	**ERROR**	**"Cr"**

5.2.4 Functional module write message format:

">"	**ADR**	**CMD**	**MOD**	**INDEX**	**DATA**	**BCC**	**CHECKSUM**	**"Cr"**

Positive answer:

A	**"Cr"**

Negative answer:

N	**ERROR**	**"Cr"**

Where:

>	Start Command Character
ADR	Two digits hexadecimal transmitted as two ASCII characters, indicates which of the 256 System 91 units on the serial link is being addressed, ranging from 00 to 0FFH.
CMD	Two digits hexadecimal Command Code.
ITEM	Two digits hexadecimal or one word hexadecimal indicating which of the possible items, defined in the All Items List, is interested in the transaction. Address up to 0FFFFH.
DATA	N bytes data according to item or function type
BCC	Block check sum word obtained executing of all transmitted ASCII from ADR to DATA included.
CHECKSUM	Checksum byte obtained executing the sum, modulo 256, of all transmitted ASCII from ADR to BCC included.
Cr	Termination character, ASCII '0DH'
ERROR	Error code see section 1.7.1
MOD	One byte hexadecimal indicating which Functional Module is interested in the transaction.
INDEX	One to three bytes hexadecimal for selection of parameters within the functional module; depending on the selected functional module some of the digits are not used or assume special meanings.

5.3 Pages

Different units uses different pages. DC9100 uses page 0 to 3. DX9100 uses page 0 to 3 and the extended area. DX9100 can use the whole range from address 0000 to FFFF together with the command 84/C4. The driver looks at the item address and changes the read/write command by itself. It's therefore possible to mix e.g. DC and DX units on the same multidropline.

	Item address	**Read Command**	**Write Command**
Page 0	00 to FF	80	C0
Page 1	100 to 1FF	81	C1
Page 2	200 to 2FF	82	C2
Page 3	300 to 3FF	83	C3
Extended	400 to FFFF	84	C4

6. DBFs

6.1 Help.dbf

TYPE	DATA	FILTER
PROTOCOL	JCONTROL	
PROTOCOL	JCONTROL2	

6.2 Protdir.dbf

TAG	FILE	BIT_BLOCK	MAX_LENGTH	OPTIONS
JCONTROL	JCONTROL	256	256	0x0DBf
JCONTROL2	N2OPEN	256	256	0x0DBf

6.3 JControl.dbf Entries

TEMPLATE	UNIT_TYPE	RAW_TYPE	BIT_WIDTH	LOW	HIGH	COMMENT
SIN%<16X	0x00000001	2	32	0	65535	Single Item Real, 0 - FFFF
SIB%<16X[.%u(1,1,8)]	0x00000002	8	8	0	65535	Single Item Byte, 0 - FFFF
SIW%<16X[.%u(1,1,16)]	0x00000003	1	16	0	65535	Single Item Integer, 0 - FFFF
SIL%<16X	0x00000004	4	32	0	65535	Single Item Long, 0 - FFFF
RTC%<16X	0x00000005	3	16	0	7	Real Time Clock, 0 - 7
DS%<16X	0x00000006	3	16	0	1	Daylight Savings, 0 - 1
ED%<16X	0x00000007	3	16	0	59	Exception Days, 0 - 3B
TS%<16.%+X%*256	0x00000008	3	16	0	28	Time Schedule, 0 - 7. 0 - 1C
TS%<16.%+X%*256	0x00000008	1	16	0	28	Time Schedule, 0 - 7. 0 - 1C

6.4 N2Open.dbf Entries

TEMPLATE	UNIT_TYPE	RAW_TYPE	BIT_WIDTH	LOW	HIGH	COMMENT	
AI%<16U	0x00000010	2	32	1	255	Current Value	Real, R
AIC%<16U[.%u(1,1,8)]	0x00000011	8	8	1	255	Object Configuration	Byte, R/W
AIS%<16U[.%u(1,1,8)]	0x00000012	8	8	1	255	Object Status	Byte, R
AILAL%<16U	0x00000013	2	32	1	255	Low Alarm Limit	Real, R/W
AILWL%<16U	0x00000014	2	32	1	255	Low Warning Limit	Real, R/W
AIHWL%<16U	0x00000015	2	32	1	255	High Warning Limit	Real, R/W
AIHAL%<16U	0x00000016	2	32	1	255	High Alarm Limit	Real, R/W
AID%<16U	0x00000017	2	32	1	255	Differential	Real, R/W
AIO%<16U	0x00000018	2	32	1	255	Override	Real, W
AIR%<16U	0x00000019	0	1	1	255	Release	Dig, W
BIC%<16U[.%u(1,1,8)]	0x0000001a	8	8	1	255	Object Configuration	Byte, R/W

BIS%<16U[.%u(1,1,8)]	0x0000001b	8	8	1	255	Object Status	Byte, R
BIO%<16U	0x0000001c	8	8	1	255	Override	Byte, W
BIR%<16U	0x0000001d	0	1	1	255	Release	Dig, W
AO%<16U	0x0000001e	2	32	1	255	Current Value	Real, R
AOC%<16U[.%u(1,1,8)]	0x0000001f	8	8	1	255	Object Configuration	Byte, R/W
AOS%<16U[.%u(1,1,8)]	0x00000020	8	8	1	255	Object Status	Byte, R
AOO%<16U	0x00000021	2	32	1	255	Override	Real, W
AOR%<16U	0x00000022	0	1	1	255	Release	Dig, W
BOC%<16U[.%u(1,1,8)]	0x00000023	8	8	1	255	Object Configuration	Byte, R/W
BOS%<16U[.%u(1,1,8)]	0x00000024	8	8	1	255	Object Status	Byte, R
BOMINON%<16U	0x00000025	1	16	1	255	Minimum On-Time(sec)	Int,R/W
BOMINOFF<16U	0x00000026	1	16	1	255	Minimum Off-Time(sec)	Int,R/W
BOMAXCH%<16U	0x00000027	1	16	1	255	Maximum Cycles/Hour	Int,R/W
BOO%<16U	0x00000028	8	8	1	255	Override	Byte, W
BOR%<16U	0x00000029	0	1	1	255	Release	Dig, W
ADF%<16U	0x0000002a	2	32	1	255	Current Value	Real, R/W
ADFS%<16U[.%u(1,1,8)]	0x0000002b	8	8	1	255	Object Status	Byte, R
ADFO%<16U	0x0000002c	2	32	1	255	Override	Real, W
ADFR%<16U	0x0000002d	0	1	1	255	Release	Dig, W
ADI%<16U[.%u(0,0,15)]	0x0000002e	1	16	1	255	Current Value	Int, R/W
ADIS%<16U[.%u(1,1,8)]	0x0000002f	8	8	1	255	Object Status	Byte, R
ADIO%<16U[.%u(0,0,15)]	0x00000030	1	16	1	255	Override	Int, W
ADIR%<16U	0x00000031	0	1	1	255	Release	Dig, W
BYT%<16U[.%u(1,1,8)]	0x00000032	8	8	1	255	Current Value	Byte, R/W
BYTS%<16U[.%u(1,1,8)]	0x00000033	8	8	1	255	Object Status	Byte, R
BYTO%<16U[.%u(1,1,8)]	0x00000034	8	8	1	255	Override	Byte, W
BYTR%<16U	0x00000035	0	1	1	255	Release	Dig, W
TIME	0x00000036	1	16	0	0	Sync Time Command	Int, W
IPADF%<16U	0x00000037	2	32	1	255	Current Value ADF	Real, R/W
OIPADF%<16U	0x00000038	2	32	1	255	Override	Real, W
IPADI%<16U[.%u(0,0,15)]	0x00000039	1	16	1	255	Current Value ADI	Int, R/W
OIPADI%<16U[.%u(0,0,15)]	0x0000003a	1	16	1	255	Override	Int, W
IPBD%<16U[.%u(1,1,8)]	0x0000003b	8	8	1	255	Current Value BD	Byte, R/W
OIPBD%<16U[.%u(1,1,8)]	0x0000003c	8	8	1	255	Override	Byte, W

6.5 PROTERR.dbf Entries

PROTOCOL	MASK	ERROR	MESSAGE	REFERENCE	ACTION	COMMENT
JCONTROL	**0**	**101**	Not existing command mode			
JCONTROL	**0**	**180**	Data not matching the item or function type			
JCONTROL	**0**	**181**	Not existing item or function			
JCONTROL	**0**	**182**	Temporarily impossible to access the item			
JCONTROL	**0**	**183**	Not programmable item			
JCONTROL	**0**	**184**	Table programmed with illegal items			
JCONTROL	**0**	**185**	Trend programmed with illegal item			
JCONTROL	**0**	**186**	Invalid Functional Module			
JCONTROL	**0**	**187**	Exceeding Addressing Range			
JCONTROL	**0**	**188**	Undefined Address after gate			
JCONTROL	**0**	**189**	No answer from Device after gate			
JCONTROL	**0**	**18A**	Password Protection Active			
JCONTROL	**0**	**190**	I=C-bus Error			
JCONTROL	**0**	**191**	Hardware not available			
JCONTROL	**0**	**192**	Illegal item number			
JCONTROL	**0**	**193**	Counters unreliable			
JCONTROL	**0**	**194**	Power supply unreliable			
JCONTROL	**FF**	**100**	*JCONTROL error	DC page 2.4, DX page 6-5, TC page 3-3, XT page 10/35		

7. References and Contacts

7.1 Contacts

begcomm Communication AB
Brunnehagen 109
S-417 47 Göteborg
Phone: +46(0)70-5305769
Email: info@begcomm.com
Website: www.begcomm.com

7.2 Documents

- Johnson Controls DX9100 Specifications Rev 3.14 1-Sep-1997
- Johnson Controls DC9100 Specifications Ver 1.03 16-Jun- 1988
- Johnson Controls TC9100 Specifications Rev 3.0 22-Dec-1995
- Johnson Controls XT9100 Design specification Ver 3.6 17-Feb-1992
- METASYS N2 SYSTEM PROTOCOL SPECIFICATION FOR VENDORS 05/10/99

7.3 Driver Version History

Version	Modified By	Details
2.00.00.001	Bertil Göransson	Original
2.00.01.001	Bertil Göransson	Cleaning of receive buffer before transmit
2.00.01.002	Bertil Göransson	New ProtocolID
2.01.00.001	Bertil Göransson	TcpIp implemented with special registration facility
2.02.00.001 B1	Bertil Göransson	N2Open implemented. Only Internal Parameters Read/Write
2.02.00.002 B1	Bertil Göransson	Override Internal Parameters Read/Write for N2Open implemented
2.02.01.001	Bertil Göransson	Updated for Citect Citect V6
2.02.02.001	Bertil Göransson	TimeChannels (TS) updated for Citect V7
3.00.00.001	Bertil Göransson	Internal information changed to begcomm. New GUID
3.01.00.001	Bertil Göransson	Ported to VS2010
3.02.00.001	Bertil Göransson	Changed license system to only one license type for both TCP and serial communication. N2Open is implemented in the documentation. N2Open is not undocumented anymore.
3.02.00.002	Bertil Göransson	Serial ID implemented in the Protection project. Ported to VS2017.
4.00.00.001	Bertil Göransson	Increased N2Open functionality

