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

JControl driver, User information

begcomm Communication AB
Brunnehagen 109
S-417 47 Göteborg
Phone +46 70 258 50 01
Email info@begcomm.com
www.begcomm.com

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 V6
2.02.02.001	Bertil Göransson	TimeChannels (TS) updated for V7
3.00.00.001	Bertil Göransson	Internal information changed to begcomm. New GUID
3.01.00.001	Bertil Göransson	Ported to VS2010

Contents

1. INTRODUCTION 6
   1.1 Scope. 6
   1.2 Outline. 6
2. QA 7
   2.1 Developers Guidelines 7
      2.1.1 Accredited Drivers. 7
      2.1.2 Independent Drivers. 7
   2.2 Accreditation process 8
3. TARGET DEVICE(S) AND PROTOCOL 9
   3.1 Introduction 9
   3.2 Device Manufacturer 9
   3.3 Communications Method 9
   3.4 Communications/Hardware Configuration 9
      3.4.1 Connection details for download/upload of DX-9100 10
      3.4.2 Wiring Diagram IU-9100 to PC Connection 10
      3.4.3 Wiring Diagram Direct Download DX9100 11
      3.4.4 I/O Device Settings 12
   3.5 Special Requirements 12
   3.6 Maximum Request Length 12
4. PROTOCOL REQUIREMENTS 13
   4.1 Introduction 13
   4.2 Initialising the Board 13
   4.3 Initialising the Port 13
   4.4 IO Device Online Test 13
   4.5 State Flow Description 13
   4.6 Message Structure 13
   4.7 Data Format 14
   4.8 Pages 14
   4.9 Floating Point Numbers 15
   4.10 Check Sum 16
      4.10.1 Block Check Sum Word 16
   4.11 Error Handling 16
5. USER INTERFACE 17
   5.1 Introduction 17
   5.2 Driver Name 17

5.3 Boards Form 17
5.3.1 Board Type 17
5.3.2 Address 17
5.3.3 IO Port 17
5.3.4 Interrupt 17
5.3.5 Special Opt 17
5.4 Ports Form 17
5.4.1 Baud Rate 17
5.4.2 Data Bits 17
5.4.3 Stop Bits 17
5.4.4 Parity 17
5.4.5 Special Opt 17
5.5 IO Devices Form 18
5.5.1 Protocol 18
5.5.2 Address 18
5.6 Pull-down lists Help 18
5.7 IO Device Variable Types 18
5.7.1 Formats and types 18
5.7.2 JControl.dbf Entries 19
5.7.3 Examples 19
5.8 Protdir.dbf 19
5.9 Parameters and INI options 20
5.9.1 Standard Parameters 20
5.9.2 Driver Specific Parameters 20
5.10 Driver Specific Errors 21
5.11 Driver Error Help 22
5.12 Debug Messages 22
5.13 Stats Special Counters 23
5.14 Hints and Tips 23
6. BASIC TESTING 24
6.1 Introduction 24
6.2 Procedure 24
7. PERFORMANCE TESTING 25
7.1 Introduction 25
7.2 Calculating the Blocking Constant – Not applicable 25
8. REFERENCES AND CONTACTS 26
8.1 References 26
8.2 Contacts 26

9. APPENDIX 1 – SOFTWARE PROTECTION 27

9.1 Unregistered driver 27

9.2 Citect Dongle not found 28

9.3 Incorrect registration 28

9.4 Registration wizard 29

9.5 Demo Mode 30

9.6 Disable the Driver 31

1. Introduction

1.1 Scope.

This document follows the development of the new driver. It serves as a functional specification, design specification and test specification.

1.2 Outline.

The specification is broken down into the following sections:

Section 1 - Introduction.
This section defines the scope of a board driver specification and outlines the items addressed by the specification.

Section 2 - Quality Assurance.
The QA section defines the requirements and procedures for Quality Assurance Accreditation. It is important you read this if you want your driver integrated into Citect.

Section 3 - Physical Communication Method.
The Physical Communication Method section defines the physical communication method supported, hardware/software suppliers, how the method is setup, any wiring diagrams involved etc.

Section 4 - Protocol Requirements.
The Protocol Requirements section details the technical considerations required or incorporated by the driver.

Section 5 - User Interface
The User Interface section defines how the user will see and setup the driver in Citect.

Section 6 - Basic Testing.
The Basic Testing section defines the items that should be addressed in basic testing by the developer.

Section 7 - Performance Testing.
The Citect Testing Department of Ci Technologies uses the Performance Testing section in the full testing of the driver. Once completed, this will provide details on the reliability and stability of the driver, and point out where the driver needs to be improved.

Section 8 - References and Contacts.
The References and Contacts section should be used as a record of reference materials and contacts used in developing this driver.

Section 9 - Softwareprotection.
The References and Contacts section should be used as a record of reference materials and contacts used in developing this driver

2. QA

2.1 Developers Guidelines

These guidelines are meant as a rough indication of what options there are for developing Citect drivers and the advantages of these options. It is not a technical discussion of options, rather a marketing guideline.

Drivers fall into two categories, Accredited and Independent.

2.1.1 Accredited Drivers.

Accredited drivers are those drivers that have been put through the CIT Driver QA Scheme and have passed all stages of this accreditation process. It is a precondition to becoming accredited that these drivers will be included with Citect in a normal release.

Accreditation has the following advantages:
1. The driver will be included in the product and a certificate stating this driver has achieved Accreditation will be sent to the developer.
2. Accredited drivers will be honoured as part of the product in terms of Citect Support and receive full co-operation between Citect Support personnel and the developer. On the other hand, independent driver problems will immediately be referred on to the original developer.
3. Help documentation and Express Wizards are provided, free of charge, for all Accredited drivers. Help documentation for Independent drivers is the responsibility of the developer.
4. Accreditation is included in the cost of the DDK. A high level of quality is expected and if this is not met the driver will not be accredited.
5. Citect Customers see value in Accredited drivers, as there is some assurance that the driver will operate as documented. Some customers may only accept Accredited drivers.

2.1.2 Independent Drivers.

Independent drivers are those that have not completed or are not intended to complete the Accreditation process. These drivers will not be included in Citect, nor will Citect Support personnel give them any support. We would request all drivers be sent to CIT regardless, even if they are not to be included in the product. If this is done, we can try to ensure compatibility with future versions of Citect.

Independent Drivers have the following advantages:
Drivers may be written by or for an end user giving them an edge over their opposition by using Citect. Drivers may be developed as part of a package offered by System Integrators or including pre-configured packages etc., thereby maintaining the intellectual and financial investment. This would be similar to value added or OEM style marketing.

2.2 Accreditation process

The following check list defines the QA steps for generating a new driver. This procedure must be followed for drivers to be integrated into Citect. It is advisable to ensure that items before each checkpoint are complete before proceeding to avoid rework if changes are required.

	Description	**Person**	**Date**
1	This specification document is written.	BeG	990413
2	Specification reviewed and accepted by CIT Driver Development.		
	At this checkpoint coding is ready to be commenced.		
3	Driver 60% coded.	BeG	981215
4	Pre-review of code by CIT Driver Development.		
	At this checkpoint final coding and documentation is ready to be commenced.		
5	Driver coded.	BeG	990130
6	Code and specification reviewed and accepted by CIT Driver Development.		
7	Testing with connection project, and performance test.	BeG	990215
8	Driver integrated into Citect source and built.		
9	Documentation is written.	BeG	990430
10	Documentation reviewed.	BeG	000219
	At this checkpoint coding is done and the driver is available as a beta.		
11a	Full testing is carried out.	BeG	990602
11b	Performance testing is carried out.	BeG	990602
11c	Specification and documentation updated from testing/performance tests	BeG	990415
	At this checkpoint the testing is complete.		
12a	Review for completeness by developer, tester, documenter and CIT Driver Development		
12b	Add driver to install disks		
12c	Add driver to drivers database		
12d	Support notified of new driver for training purposes		
13	Sales notified of new driver		
	The driver is now finished.		

The hand over of a driver requires that all of the above steps are completed and checked off.

3. Target Device(s) and Protocol

3.1 Introduction

This section defines the types of I/O Devices that are targeted by this driver.

3.2 Device Manufacturer

Johnson Controls

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

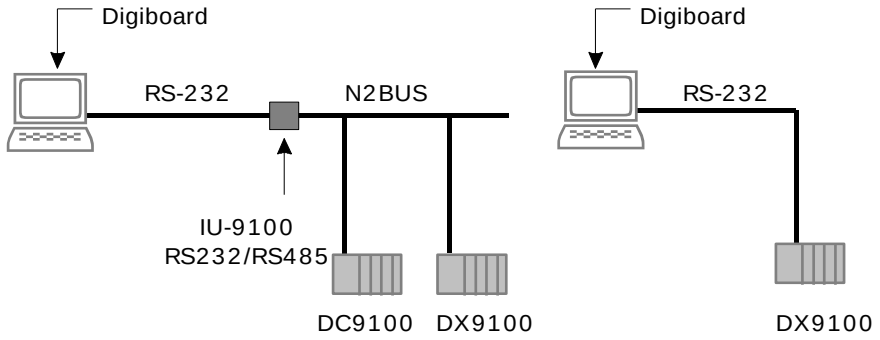
DX-9100 Tested by Autic
DC-9100 Tested by Autic
XT/XP-9100
TC-9100
SC-9100
DR-9101

3.3 Communications Method

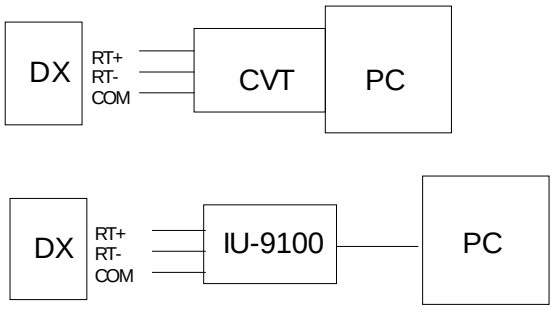
Serial RS232 direct or RS485 multidrop via converter IU-9100. The driver acts as master.

3.4 Communications/Hardware Configuration

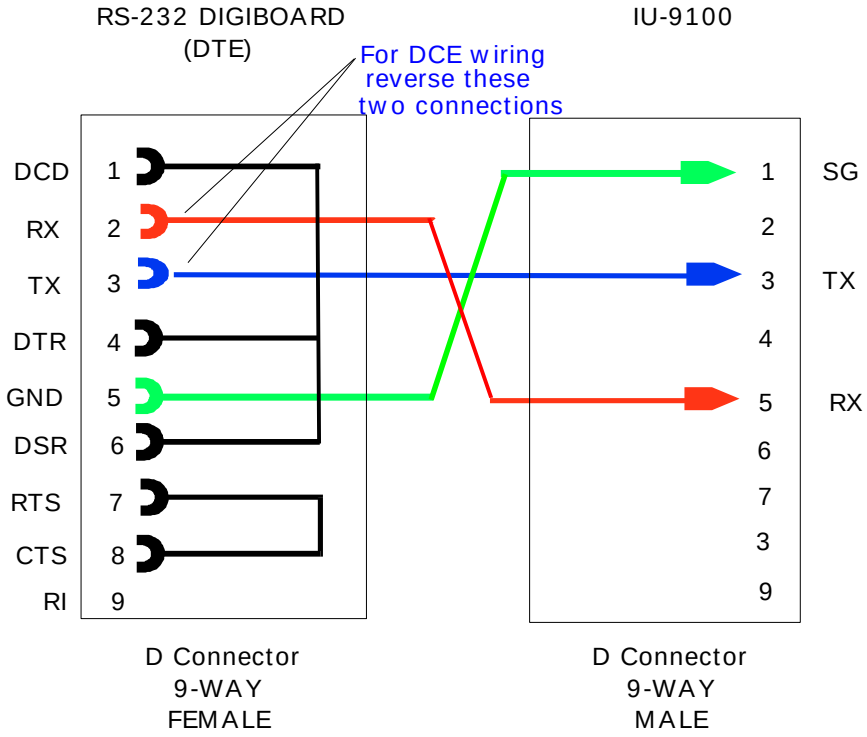
The serial method of communication to the Johnson family of intelligent digital and analog input/output controllers, uses the JCONTROL protocol. Using this method you can connect to single controllers or to multiple controllers as in the following diagram:

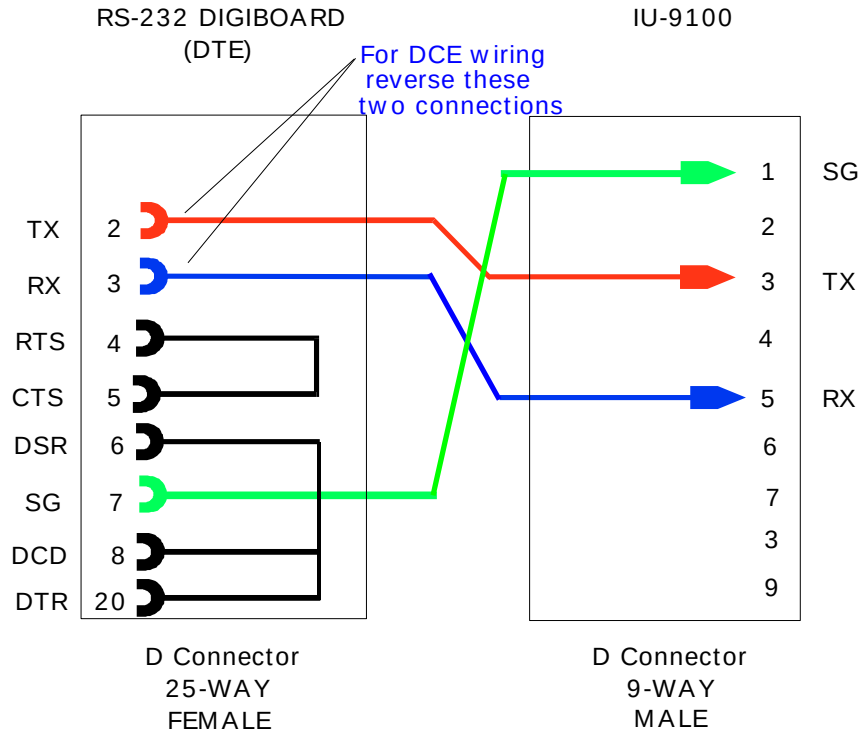


3.4.1 Connection details for download/upload of DX-9100

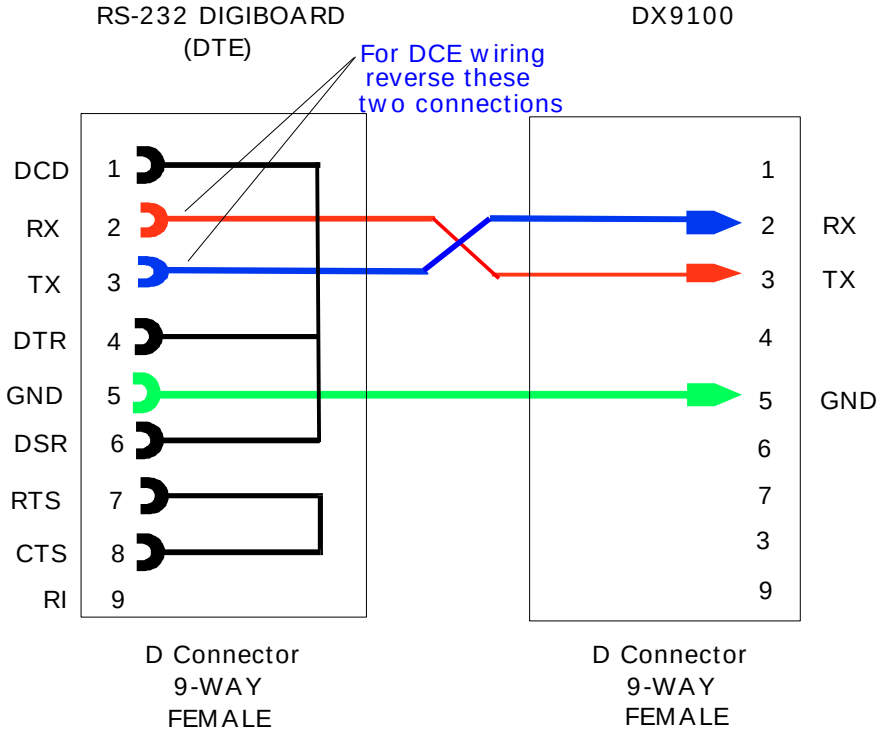


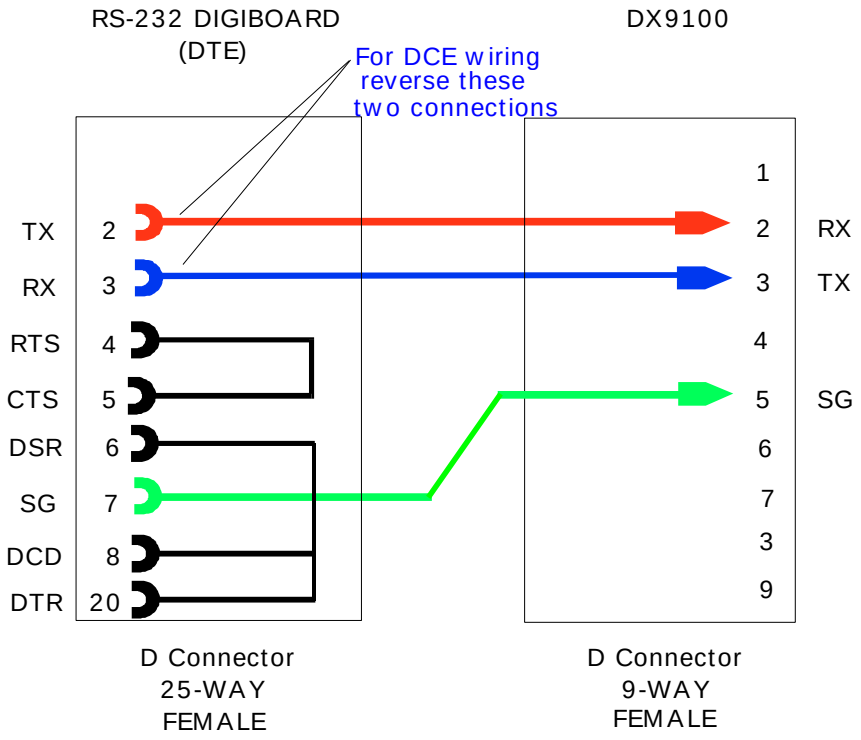
3.4.2 Wiring Diagram IU-9100 to PC Connection





3.4.3 Wiring Diagram Direct Download DX9100





3.4.4 I/O Device Settings

The Johnson factory sets all communications parameters in the units.

3.5 Special Requirements

The JCONTROL driver for Johnson Control equipment deliveries by begcomm Communication AB, Brunnehagen 109, 417 47 Göteborg, Sweden. All copyrights and licenses belongs to begcomm Communication AB.

3.6 Maximum Request Length

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

4. Protocol Requirements

4.1 Introduction

This section documents all the requirements of the protocol itself.

4.2 Initialising the Board

No special initialisation is necessary.

4.3 Initialising the Port

No special initialisation is necessary.

4.4 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

4.5 State Flow Description

The communications structure is a typical Request/Response structure.

4.6 Message Structure

Single item read message format:

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

Positive answer:

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

Negative answer:

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

Single item write message format:

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

Positive answer:

A	**“Cr”**

Negative answer:

N	**ERROR**	**“Cr”**

Functional module read message format:

“>”	**ADR**	**CMD**	**MOD**	**INDEX**	**BCC**	**CHECKSUM**	**“Cr”**

Positive answer:

A	**DATA**	**BCC**	**CHECKSUM**	**“Cr”**

Negative answer:

N	**ERROR**	**“Cr”**

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 fromm 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 5.12
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.

4.7 Data Format

The data format is 8-bit ascii. Any hexadecimal character is valid.

4.8 Pages

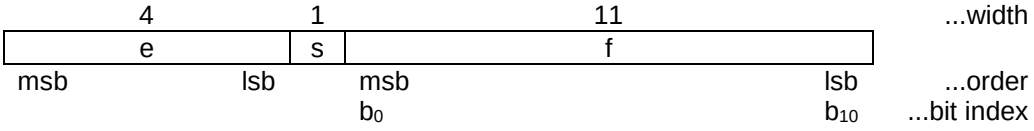
Different units uses different pages. DC9100 uses page 0 to 3. DX9100 uses page 0 to 3 and the extended area. DX9100 can use the whole range from address 0000 to FFFF together with the command 84/C4. The driver looks at the item address and changes the read/write command by it self. It’s therefore possible 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

4.9 Floating Point Numbers

The protocol uses a floating point number witch is defined by two bytes, not four bytes IEEE 32 as Citect uses. The following format (in the communication buffer will be in the sequence from the least significant byte to the most significant byte) the 16 bits string is characterized by three components, an exponent, a sign and a significant. Its numerical value is the signed product of its significant and two raised to the power of its exponent.

2 Bytes Floating Point Format



where: e 4 bits exponent. Any integer between 0 and 15, inclusive
s 1 bit sign (0 = positive, 1 = negative)
f 11 bits fractional significant with an implied binary point and the format: $\mathbf{0.b_0b_1...b_{10}}$

The numerical value of a floating point string can be calculated as:

Numerical Value = $(-1)^s\ 2^e\ (0.b_0b_1...b_{10})$

The numerical value of the n^{th} bit (b_n) in the significant field (n from 0 to 10) is:

Significant Value(b_n) = $\mathbf{2^{-(n+1)}}$

A number is zero when all bits of the fractional significant are 0.

4.10 Check Sum

The protocol uses two checksum controls. First comes the block check sum and after that the checksum witch uses a normal summary model.

4.10.1 **Block Check Sum Word**

Each message contains a Block Check Sum word BCC, composed by two bytes ordered as BCC1 & BCC2 which are calculated, using one's complement arithmetic (a carry resulting from a two byte sum is added to the result), with the following procedure:

a) Initialize S1(0)=0 and S2(0)=0
b) Perform partial calculation using all message's N bytes B(n) starting form first byte B(1)
   S1(n) = S1(n-1) + B(n)
   S2(n) = S2(n-1) + S1(n)
c) Calculate checksum bytes as
   BCC1 = - (S1(N) + S2(N))
   BCC2 = S2(N)

4.11 Error Handling

Three consecutive timeouts will put a unit off line.

5. User Interface

5.1 Introduction

This section defines how the user will see the driver. This relates directly to how the Citect forms need to be filled out and any special INI options. For the kernel, the debug trace messages and the Stats.Special counters are documented.

5.2 Driver Name

JCONTROL

5.3 Boards Form

5.3.1 Board Type

For serial communication choose **COMX** as the board driver.

For communication over tcp/ip choose **TCPIP** as the board driver.

5.3.2 Address

0

5.3.3 IO Port

N/A

5.3.4 Interrupt

N/A

5.3.5 Special Opt

N/A

5.4 Ports Form

5.4.1 Baud Rate

9600 bit/sec (Factory set)

5.4.2 Data Bits

8 (Factory set)

5.4.3 Stop Bits

1 (Factory set in Johnson units)

5.4.4 Parity

NONE (Factory set)

5.4.5 Special Opt

None for serial communication
For tcp/ip mode the IP address and port number has to be filled in.
Eg. -i192.168.100.35 -p8001

5.5 IO Devices Form

5.5.1 Protocol

JCONTROL

5.5.2 Address

The address format looks like 241 etc. The address format is decimal and the range is between 0 and 255.

5.6 Pull-down lists Help

The following entries should be included in the Citect HELP.DBF spec file.

TYPE	DATA	FILTER
PROTOCOL	**JCONTROL**	

5.7 IO Device Variable Types

5.7.1 Formats and types

IO Device Type	Citect data format	Citect data types	Description/Special Usage/Limitations/ Valid Ranges
Single Item	SINx	REAL	Read / Write. Johnson Control special floating point number (16 bit).
Single Item	SIBx	BYTE	Read / Write.
Single Item	SIBx.y	DIGITAL	Read / Write.
Single Item	SIWx	UINT	Read / Write.
Single Item	SIWx.z	DIGITAL	Read / Write.
Single Item	SILx	LONG	Read / Write. Allowed value is maximum 2,147,483,647
Real Time Clock	RTCt	BCD	Read / Write.
Daylight Savings	DSd	BCD	Read / Write.
Exception Days	EDe	BCD	Read / Write.
Time Schedule	TSm.w	BCD	Read / Write.
Time Schedule	TSm.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

5.7.2 JControl.dbf Entries

TEMPLATE	UNIT_TYPE	RAW_TYPE	BIT_WIDTH	LOW	HIGH	COMMENT
SIN%<16X	1	2	32	0	65535	Single Item Real, 0 - FFFF
SIB%<16X[.%u(1,1,8)]	2	8	8	0	65535	Single Item Byte, 0 - FFFF
SIW%<16X[.%u(1,1,16)]	3	1	16	0	65535	Single Item Integer, 0 - FFFF
SIL%<16X	4	4	32	0	65535	Single Item Long, 0 - FFFF
RTC%<16X	5	3	16	0	7	Real Time Clock, 0 - 7
DS%<16X	6	3	16	0	1	Daylight Savings, 0 - 1
ED%<16X	7	3	16	0	59	Exception Days, 0 - 3B
TS%<16.%+X%*256	8	3	16	0	28	Time Schedule, 0 - 7. 0 - 1C
TS%<16.%+X%*256	9	1	16	0	28	Time Schedule, 0 - 7. 0 - 1C

5.7.3 Examples

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)

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

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

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

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)

5.8 Protdir.dbf

TAG	FILE	BIT_BLOCK	MAX_LENGTH	OPTIONS

JCONTROL	JCONTROL	256	256	0x0DB

5.9 Parameters and INI options

5.9.1 Standard Parameters

Block	Not used by driver. Set at 4.
Delay	5mS (30000/BR mS (BR=Baud Rate bit/sec))
MaxPending	2
Polltime0	
Timeout	1000mS
Retry	1
WatchTime	30s

5.9.2 Driver Specific Parameters

RegisterTcpip	0 default 0 serial 1 tcp/ip -1 messagebox

5.10 Driver Specific Errors

Driver Error Code (Hexadecimal)	Mapped to (Generic Error label)	Meaning of Error Code
101	**GENERIC_INVALID_COMMAND**	Not existing command mode
180	**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 unrelible

5.11 Driver Error Help

The following entries should be included in the Citect PROTERR.DBF spec file.

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 unrelible			
JCONTROL	**FF**	**100**	*JCONTROL error	DC page 2.4, DX page 6-5, TC page 3-3, XT page 10/35		

5.12 Debug Messages

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>

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>

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>

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)

5.13 Stats Special Counters

Number	Label	Purpose/Meaning of this counter
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
.		
.		
18	TcpIp registered	Value = 1 when tcpip version is registered
19	Serial registered	Value = 1 when serial version is registered

5.14 Hints and Tips

- 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, witch 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 eg 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.

6. Basic Testing

6.1 Introduction

The programmer will perform a minimum level of testing that is outlined here.

A sample Project is available which can be used as a starting point for the programmer's test Project. When the programmer has completed basic testing and debugging this Project should be backed up and supplied to the Citect Testing department.

6.2 Procedure

The following are points that should be covered by basic testing.

- On startup the IO Device comes online without errors.
  -OK

- The driver supports IO Devices of addresses as documented in the specification.
  -OK

- The driver reports the IO Device offline when the IO Device is a) powered down, b) disconnected.
  -OK

- The driver will re-establish communication with the IO Device after a) power cycle, b) disconnection/ reconnection.
  -OK

- Confirm that retries (if supported) and error reporting operate correctly.
  -OK

- The driver reads all the device data types documented as readable in this specification.
  -OK

- The driver writes to all the device data types documented as write-able in this specification.
  -OK

- The driver reads and writes all data formats supported by the protocol, i.e. DIGITAL, BYTE, INT, LONG, REAL, BCD.
  -OK

- Test the limit of the IO Devices request size, this should be done for at least DIGITAL and an INT data formats.
  -Not applicable of this driver.

- Let the driver run over night and check that no retries or other errors have occurred.
  -OK

- If a multi-drop or network protocol and if the hardware is available then the protocol should be tested with more than one IO Device connected.
  -OK (Also tested with one DC9100 and one DX9100 multidropped)

7. Performance Testing

7.1 Introduction

The following are tests that give some indication of the driver's performance. The programmer needs to perform these tests since the results feed back into the Constants structure and the PROTDIR.DBF.

7.2 Calculating the Blocking Constant – Not applicable

Due to the nature of especially the DX unit the driver doesn't support blocking. The item datatypes is very mixed in consecutive order.

8. References and Contacts

8.1 References

- 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

8.2 Contacts

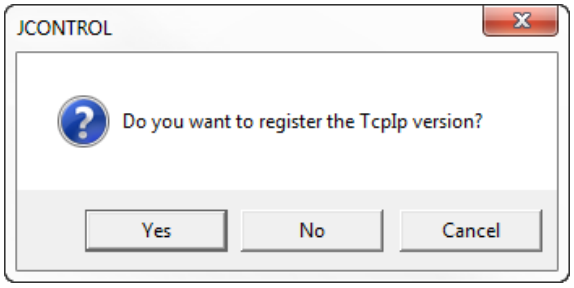
www.begcomm.com

info@begcomm.com

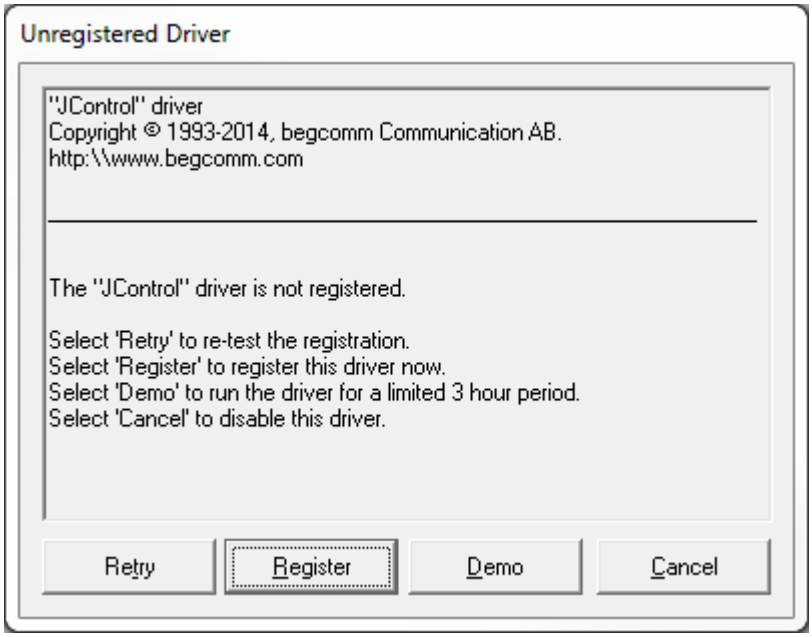
9. Appendix 1 – Software Protection

9.1 Unregistered driver

The Citect parameter RegisterTcpip has to 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 se 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=0 or =1 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.

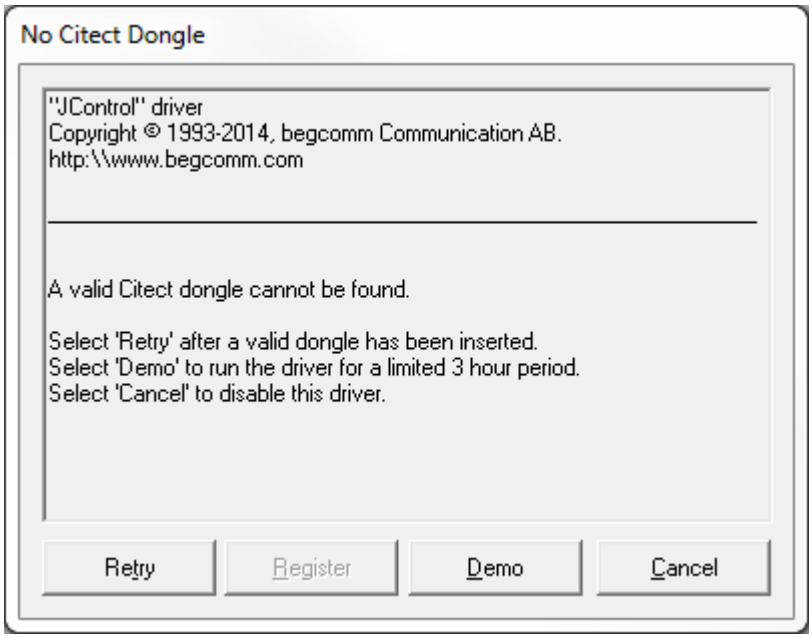


When a driver is run for the first time, or each time an unregistered driver runs, an 'Unregistered driver' dialog box will be displayed. This dialog box will prompt you to either:
a) Let the driver re-test the registration settings
b) Register the driver
c) Run in a demo mode
d) Disable the driver



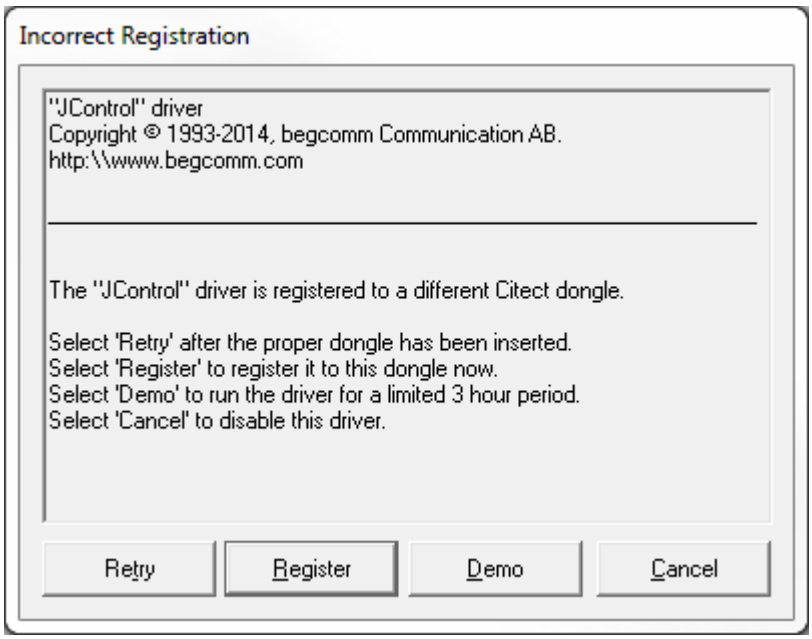
9.2 Citect Dongle not found

If there is no Citect dongle attached to local parallel port, or the dongle cannot be detected, then the 'No Citect Dongle' dialog box will be displayed. This dialog box will prompt you to either:
a) Insert a valid Citect dongle and retry
b) Run in a demo mode
c) Disable the driver



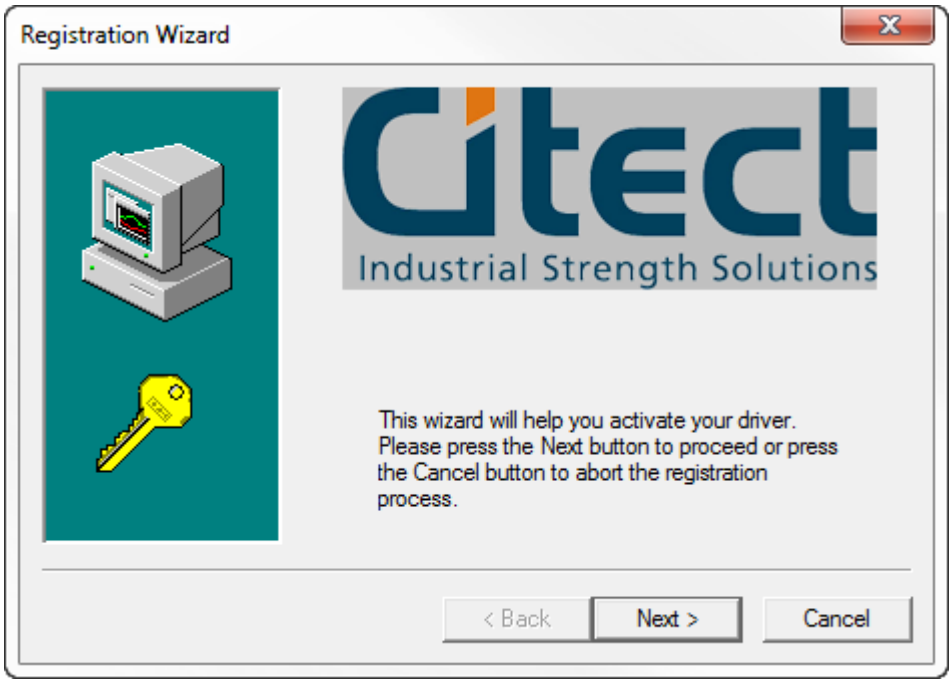
9.3 Incorrect registration

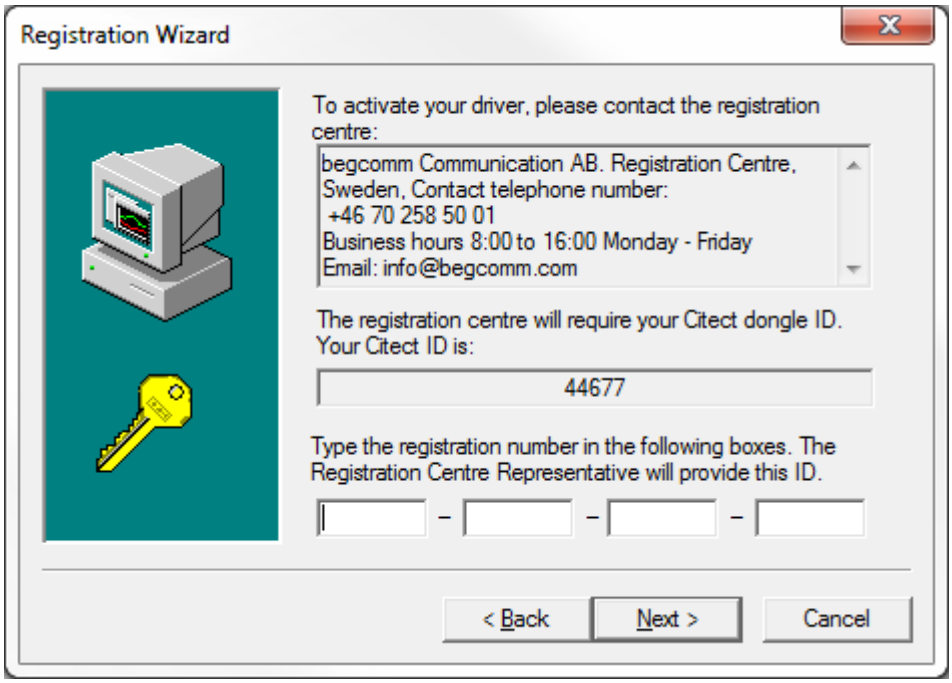
If the Citect dongle does not match the registration key, then the 'Incorrect registration' dialog box will be displayed. This dialog box will prompt you to either:
a) Replace the dongle and retry
b) Register the driver to the current attached dongle
c) Run in a demo mode
d) Disable the driver

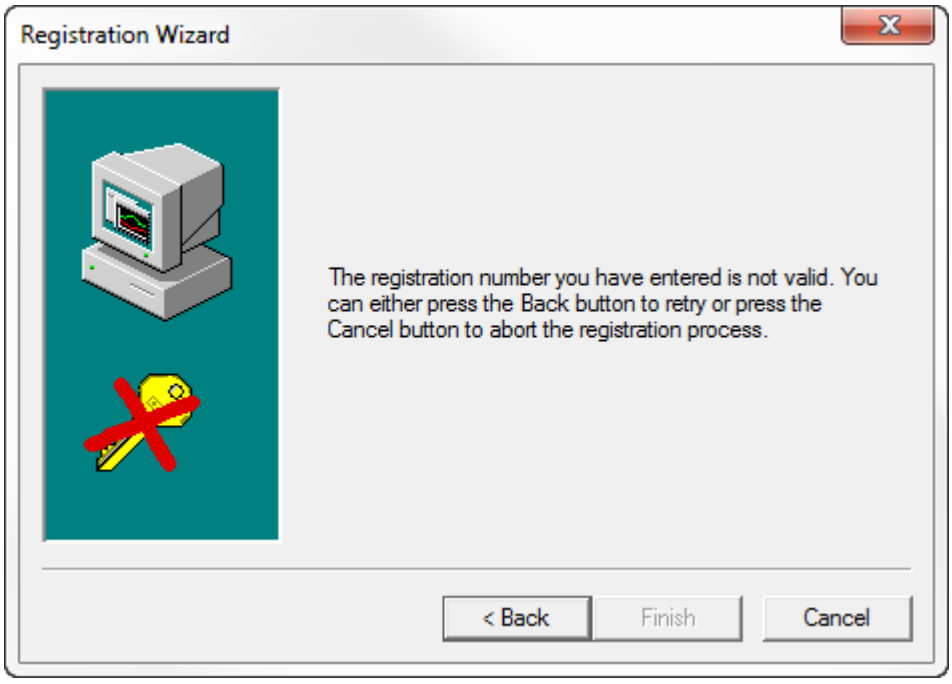


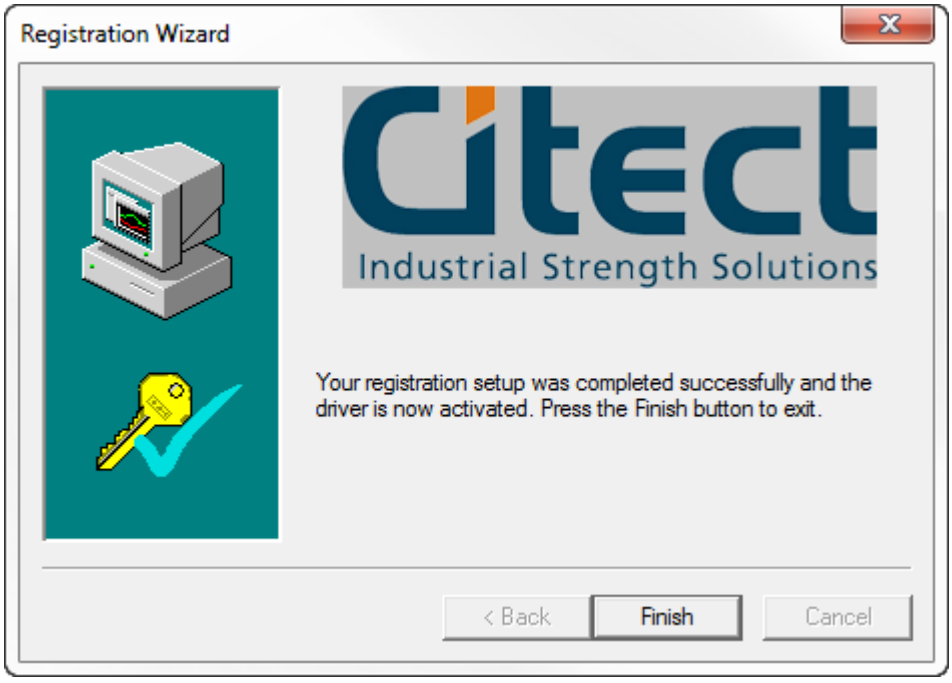
9.4 Registration wizard

If the user selects the ‘Register’ button on the dialog box, the ‘Registration Wizard’ will be activated which guides you through the registration process. If you have entered an invalid registration number, then a fail message will be displayed as shown, otherwise a successful message will be displayed and the registration number will be saved in the group [DRIVER_REGISTRATION] in Citect.ini file.









If you cancel the registration process, the ‘Unregistered Driver’ dialog box will remain open. If the ‘Finish’ button is pressed (process finished successfully), the dialog box will close automatically.

9.5 Demo Mode

If you choose the ‘Demo’ button from the dialogue, then the driver will run in demo mode for period of time.

9.6 Disable the Driver

If you choose the ‘Cancel’ button from the dialog, the driver will be put in ‘Channel Offline’ mode. This means the driver will report the ‘Channel Offline’ hardware alarm whenever the ‘WatchTime’ parameter triggers, which is normally every 30 seconds.

