Citect for Windows
Driver Specification
Klöckner Moeller PS316/PS416 and PS4200 Drivers

Author	Date	Comment
Terence White	07 April 1997	Original
Allen Hobson	10 April 1997	Update
Bryan Steer	18 September	Updated document
Trevor Hudson	10 August 2000	Link to his file

Contents

1. INTRODUCTION 5

1.1 Scope. 5

1.2 Outline. 5

2. QA 6

2.1 Developers Guidelines 6

2.1.1 Accredited Drivers. 6

2.1.2 Independent Drivers. 6

2.2 Accreditation process 7

3. TARGET DEVICE(S) AND PROTOCOL 8

3.1 Introduction 8

3.2 Device Manufacturer 8

3.3 Device Definition 8

3.4 Communications Method 8

3.5 Communications/Hardware Configuration PRG Interface 8

PS416 Communications. 8

PS316 Communications. 8

PS 4200 Communications. 8

3.5.1 Wiring Diagrams 9

Pin assignments 9

3.5.2 I/O Device Settings 11

3.5.3 Software Setup 11

3.6 Maximum Request Length 11

4. PROTOCOL REQUIREMENTS 12

4.1 Introduction 12

4.2 Initialising the Board 12

4.3 Initialising the Port 12

4.4 Initialising the IO Device 12

4.5 IO Device Online Test 12

4.6 State Flow Description 12

4.7 Message Structure 12

4.8 Data Format 13

4.9 Check Sum 13

4.10 Error Handling **13**

5. USER INTERFACE **14**

5.1 Introduction **14**

5.2 Driver Name **14**

5.3 Boards Form Error! Bookmark not defined.

5.3.1 Board Type **Error! Bookmark not defined.**

5.3.2 Address **Error! Bookmark not defined.**

5.3.3 IO Port **Error! Bookmark not defined.**

5.3.4 Interrupt **Error! Bookmark not defined.**

5.3.5 Special Opt **Error! Bookmark not defined.**

5.4 Ports Form **14**

5.4.1 Baud Rate 14

5.4.2 Data Bits 14

5.4.3 Stop Bits 14

5.4.4 Parity 14

5.4.5 Special Opt 14

5.5 IO Devices Form **15**

5.5.1 Protocol 15

5.5.2 Address 15

5.6 Pulldown lists Help **15**

5.7 IO Device Variable Types **15**

5.8 PROTDIR.DBF **15**

5.9 Parameters and INI options **16**

5.9.1 Standard Parameters 16

5.9.2 Driver Specific Parameters 16

5.10 Driver Specific Errors **16**

5.11 Driver Error Help **16**

5.12 Debug Messages **16**

5.13 Stats Special Counters **16**

5.14 Hints and Tips **16**

6. BASIC TESTING **17**

6.1 Introduction **17**

6.2 Procedure **17**

7. PERFORMANCE TESTING **18**

7.1 Introduction **18**

7.2 Calculating the Blocking Constant **18**

8. REFERENCES AND CONTACTS 19

8.1 References 19

8.2 Contacts 19

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 full testing of the driver. Once complete, 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.

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 Ci Technologies 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 cooperation 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 is 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 Ci Technologies 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:
1. Drivers may be written by or for an end user giving them an edge over their opposition by using Citect.
2. 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.	Terence White	97/04/07
2	Specification reviewed and accepted by Ci Technologies Driver Development.	Bruce Kinchin	17-3-98
	At this checkpoint coding is ready to be commenced.		
3	Driver coded.	Bryan Steer	97/9/18
4	Code and specification reviewed and accepted by Ci Technologies Driver Development.		
5	Testing with connection project, and performance test.		
6	Driver integrated into Citect source and built.		
7	Documentation is written.		
8	Documentation reviewed.		
	At this checkpoint coding is done and the driver is available as a beta.		
9a	Full testing is carried out.		
9b	Performance testing is carried out.		
9c	Specification and documentation updated from testing/performance tests		
	At this checkpoint the testing is complete.		
10a	Review for completeness by developer, tester, documenter and CIT Driver Development		
10b	Add driver to install disks		
10c	Add driver to drivers database		
10d	Support notified of new driver for training purposes		
11	Sales notified of new driver		
	The driver is now finished.		

The hand over of a driver requires that all 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

Klöckner Moeller

3.3 Device Definition

PS316, PS416 or PS4200

3.4 Communications Method

Serial via PLC programming port.

3.5 Communications/Hardware Configuration PRG Interface

PS416 Communications.

The PS416 is connected to the CITECT PC via the programming port on the PLC. This port is known as the PRG interface. The PRG Interface can be set as an RS232 or RS485 interface according to the application at hand.

PS316 Communications.

The programming port of the PLC is a RS485 port. The CITECT PC can be connected to the programming port of the PLC by means of a RS232 to RS485 converter.

PS 4200 Communications.

The programming port of these PLCs is a RS232 port. The CITECT PC can be connected to the programming port of the PLC.

Interface Parameters:

The interface parameters for the PLC are set up via the Klöckner Moeller programming software.

The Data format is defined as:

Settings		Value
Data bit	=	8
Parity	=	None
Stop Bit	=	2
Baud Rate	=	2400, 4800, 9600 or 19200 baud.

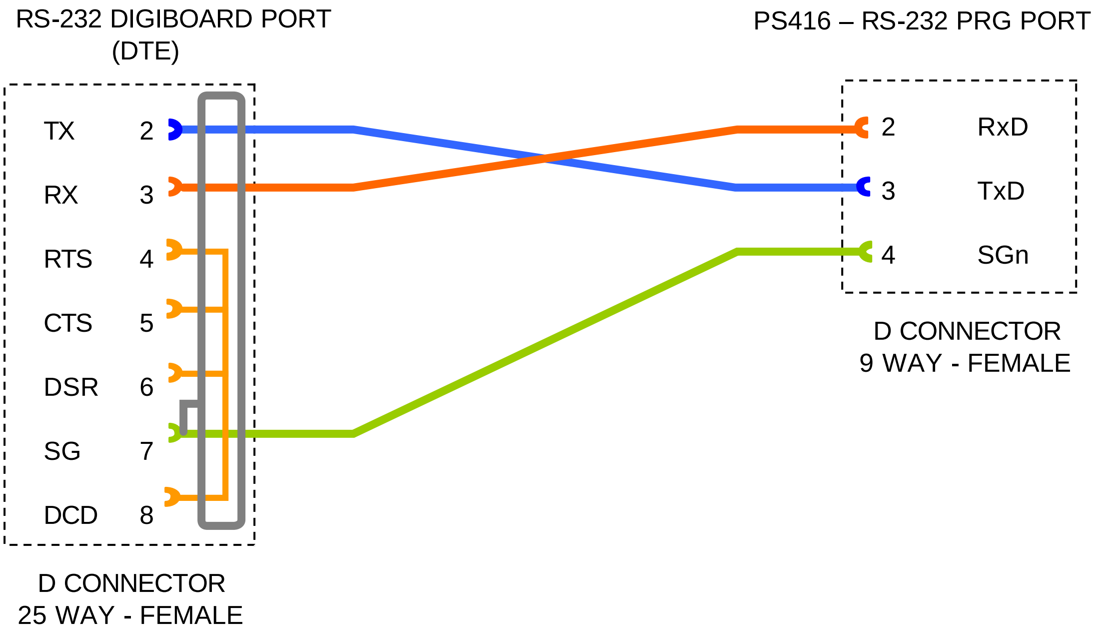
3.5.1 Wiring Diagrams

Pin assignments

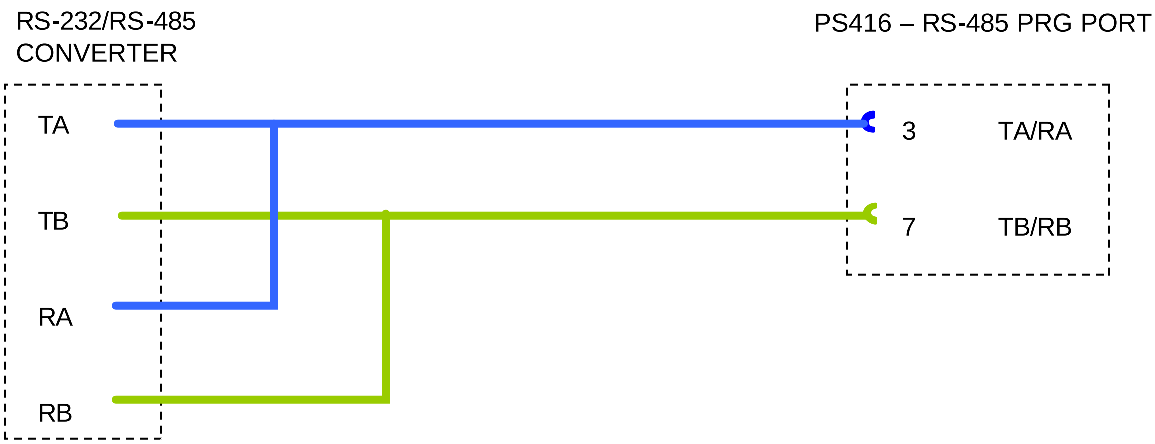
The table below shows the pin assignments for the programming port of each PLC.

Pin No.	PS416	PS4200	PS316
1	Not Connected	Not Connected	TB/RB
2	RxD	RxD	Screen
3	TxD / TA /RA	SGnd	Power Ground
4	Power Ground	Not Connected	TA/RA
5	SGnd	TxD	+ 9 Volts
6	Not Connected	-	-
7	TB/RB	-	-
8	Power Ground	-	-
9	+ 5 Volts	-	-

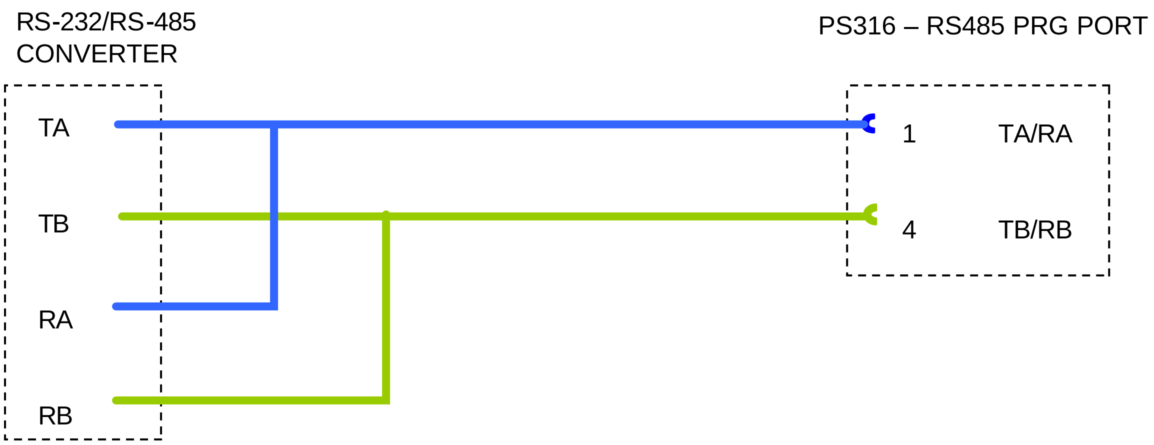
PS416 – RS232 Interface



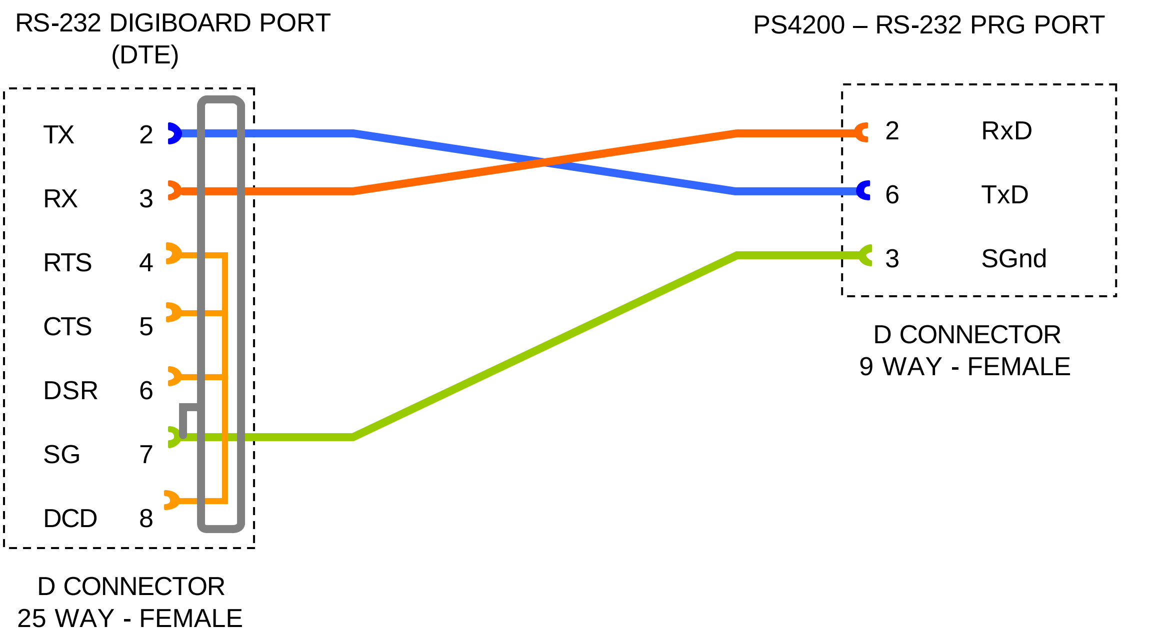
PS416 – RS485 Interface



PS316 – RS485 Interface



PS4200 – RS485 Interface



3.5.2 I/O Device Settings

The address of the unit must be set up on the PLC address switches.

The table below illustrates the switch positions:

Address	Switch							
	1	2	3	4	5	6	7	8
1	1	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0
3	1	1	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0
5	1	0	1	0	0	0	0	0
6	0	1	1	0	0	0	0	0
7	1	1	1	0	0	0	0	0
8	0	0	0	1	0	0	0	0
9	1	0	0	1	0	0	0	0
10	0	1	0	1	0	0	0	0
11	1	1	0	1	0	0	0	0
12	0	0	1	1	0	0	0	0
13	1	0	1	1	0	0	0	0
14	0	1	1	1	0	0	0	0
15	1	1	1	1	0	0	0	0
16	0	0	0	0	1	0	0	0
17	1	0	0	0	1	0	0	0
18	0	1	0	0	1	0	0	0
19	1	1	0	0	1	0	0	0
20	0	0	1	0	1	0	0	0
21	1	0	1	0	1	0	0	0
22	0	1	1	0	1	0	0	0
23	1	1	1	0	1	0	0	0
24	0	0	0	1	1	0	0	0
25	1	0	0	1	1	0	0	0
26	0	1	0	1	1	0	0	0
27	1	1	0	1	1	0	0	0
28	0	0	1	1	1	0	0	0
29	1	0	1	1	1	0	0	0
30	0	1	1	1	1	0	0	0
31	1	1	1	1	1	0	0	0

3.5.3 Software Setup

The Klockner Moeller protocol will not allow a SCADA to communicate directly with the inputs or outputs. Thus CITECT can only communicate with the vaiable registers called Marker words. Because of this the PLC must be programmed so that I/O addresses that need to be addressed by CITECT must be mapped to Marker words (M words)

3.6 Maximum Request Length

32 Integers, 64 Bytes or 512 Digitals

4. Protocol Requirements

4.1 Introduction

This section documents all the requirements of the protocol itself.

4.2 Initialising the Board

Nothing special

4.3 Initialising the Port

Nothing special

4.4 Initialising the IO Device

Nothing special

4.5 IO Device Online Test

The driver reads the status word, which is at addresses 0x5200 to 0x5201

4.6 State Flow Description

Any request received from Citect places the Citect constructed Driver Control Block (DCB) on an input queue. This request will then be handled asynchronously, during the drivers tick period.

On tick and if the communication's state is idle the driver will send a Read or Write request. Once a request is sent the DCB is removed from the input queue and placed on an output queue.

The state of the receive buffer on the communications port is checked and the port interrupt will pass the control to the driver if new data is received. If a message is received it will be checked and compared with the DCB from the output queue. If this message is the response to the request in the DCB, the DCB's buffer is filled with the received data and posted back to Citect.

4.7 Message Structure

Commands: Read from data memory, write to data memory:

STX	User No.	Control Byte	Start Add. MSB	Start Add. LSB	End Add. MSB	End Add. LSB	LRC

Where:

STX	= 0Xc
User No	= 0
Control Byte	= 0x82 – Read Request
	= 0x83 – Write Request
Start Address	= The start address of the Marker block to be read
End Address	= The end address of the Marker block to be read
LRC	= Longitudinal Redundancy Check

Data Block: Data read or written to memory:

STX	User No.	Control Byte	No. of Bytes	Data Byte 1	Data Byte ………..	Data Byte n	LRC

Where:

STX	= 0Xc
User No	= 0
Control Byte	= 0x14 – Data Block
No. of Bytes	= The number of data bytes in the message (must be an even number)
Data Byte x	= The actual data in the message
LRC	= Longitudinal Redundancy Check

4.8 Data Format

The protocol allows for the sending and receiving of an even amount of bytes only in data blocks. Thus, odd number of bytes and Digital's are the responsibility of the driver.

4.9 Check Sum

The last byte of the packet is a check byte. The check byte is created over an XOR link. The calculation is made starting from the byte following the STX.

4.10 Error Handling

Three consecutive timeouts will put the unit offline

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 debugs trace messages and the Stats.Special counters are documented.

5.2 Driver Name

KLOCK

5.3 Boards Form

5.3.1 Board Type

COMX

5.3.2 Address

0

5.3.3 IO Port

BLANK

5.3.4 Interrupt

BLANK

5.3.5 Special Opt

BLANK

5.4 Ports Form

5.4.1 Baud Rate

2400, 4800, 9600 or 19200 baud

5.4.2 Data Bits

8 bits only

5.4.3 Stop Bits

2 stop bits only

5.4.4 Parity

No parity only

5.4.5 Special Opt

None

5.5 IO Devices Form

5.5.1 Protocol

Klockl316 for the PS316 PLC
Klock416 for the PS416 PLC
Klock4200 for the PS4200 PLC

5.5.2 Address

The address of the IO unit . See section 3.5.2 for address settings on the PLC

5.6 Pulldown lists Help

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

TYPE	DATA	FILTER
PROTOCOL	**KLOCK316**	none
PROTOCOL	**KLOCK416**	none
PROTOCOL	**KLOCK4200**	none

5.7 IO Device Variable Types

IO Device Type	**Citect data format**	**Citect data types**	**Description/Special Usage/Limitations/ Valid Ranges**
Marker words	Mxxx	INT	Maximum of 32 read/write
Marker Digital's	Mxxx.zz	DIGITAL	Read/write as words, maximum read 32 words, maximum write 1 bit

Where:

Xxx Where xxx denotes an internal memory register called a Marker. In the PS416 and PS316 xxx is a number from 0 to 2172 . In the PS4200, xxx must be a even number between 0 and 4096.

Zz Is a number from 0 to 15 denoting the specific bit of the internal marker xxx.

5.8 PROTDIR.DBF

TAG	**FILE**	**BIT_BLOCK**	**MAX_LENGTH**	**OPTIONS**
KLOCK316	KM316	512	512	0x8803
KLOCK416	KM416	512	512	0x8803
KLOCK4200	KM416	512	512	0x8803

5.9 Parameters and INI options

5.9.1 Standard Parameters

Block	32
Delay	0
MaxPending	2
Polltime	0
Timeout	1000
Retry	1
WatchTime	30

5.9.2 Driver Specific Parameters

None

5.10 Driver Specific Errors

Driver Error Code (Hexadecimal)	Mapped to (Generic Error label)	Meaning of Error Code
None		

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
None						

5.12 Debug Messages

Bad data	Data packet received is not in the expected format
Bad xmit	An STX header is not received
Receive not ok	The data received is not recognisable
Invalid DCB	The DCB id not valid

5.13 Stats Special Counters

Number	Label	Purpose/Meaning of this counter
0	**Total Cmds**	**Number of messages sent**
1	**Total Req**	**Number of messages received**

5.14 Hints and Tips

None

6. Basic Testing

6.1 Introduction

The programmer will perform a minimum level of testing, which 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 by backed up and supplied to the Citect Testing department.

6.2 Procedure

Basic testing should cover the following points:

- On startup the IO Device comes online without errors.
- The driver supports IO Devices of addresses as documented in the specification.
- The driver reports the IO Device offline when the IO Device is a) powered down, b) disconnected.
- The driver will re-establish communication with the IO Device after a) power cycle, b) disconnection/ reconnection.
- Confirm that retries (if supported) and error reporting operate correctly.
- The driver reads all the device data types documented as readable in this specification.
- The driver writes to all the device data types documented as write-able in this specification.
- The driver reads and writes all data formats supported by the protocol, ie. DIGITAL, INT, LONG, REAL, BCD, LONG_BCD.
- Test the limit of the IO Devices request size, this should be done for at least DIGITAL and an INT data formats.
- Let the driver run over night and check that no retries or other errors have occurred.
- 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.

\\SYD-FILE1\DATA1\CITECT\DRIVERS\SPEC\Klock Testing.doc

7. Performance Testing

7.1 Introduction

This section outlines the tests, which 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

The Performance test procedure is documented in the driver development kit in Appendix A, 'Calculating the Block Constant'. The results of the performance test are recorded here.

block size to read [words]	Average response time [mS]
1	
{25% of maximum}	
{50% of maximum}	
{75% of maximum}	
{maximum}	

From these results the overhead and rate are determined and the ideal blocking constant is calculated

Overhead [mS] =

Word Rate [words / mS] =

Blocking constant [words] =

Note that the calculated blocking constant must now be set by the programmer in the Constants structure (the Block field) in bytes and in the PROTDIR.DBF (the BIT_BLOCK field) in bits.

8. References and Contacts

8.1 References

SUCOM-A Protocol Description © Klöckner Moeller

8.2 Contacts

Technical information:
Xycom South Africa
+27 11 462-6671
FAX +27 11 462-5159
EMAIL xycom@pixie.co.za