Citect for Windows, Version 5.xx, 6.xx

PRV Driver, User information

Beijer Electronics AB
Box 426
S-201 24 Malmö
Phone +46 40 358600
Telefax +46 40 932301
www.beijer.se

Contents

1. QA

1.1 Introduction

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

1.2 Procedure for generating a new driver

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.

	Description	**Person**	**Date**
1	This specification document is written.	TR/BeG	1999.09.03
2	Specification reviewed and accepted by R&D department.		
3	Driver coded.	TR	1999.08.25
4	Code and specification reviewed and accepted by R&D department.		
5	Testing with connection project, and performance test.	TR	1999.09.03
6	Driver integrated into Citect source and built.		
7	Documentation is written (HLP or MVB files)		
	At this checkpoint coding is done and the driver is available as a beta.		
8a	Full testing is carried out.	TR/BeG	
8b	Performance testing is carried out.	TR/BeG	
8c	Specification and documentation updated from testing/performance tests	TR/BeG	
	At this checkpoint the testing is complete.		
9a	Review for completeness by developer, tester, documentor and R&D staff		
9b	Add driver to install disks	BeG	2000.06.22
9c	Add driver to protocols database		
9d	Support notified of new driver for training purposes	BeG	2000.06.22
10	Sales notified of new driver	BeG	2000.06.22
	The driver is now finished.		

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

2. Target Device(s) and Protocol

2.1 Introduction

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

2.2 Device Manufacturer

Siemens Building Technologies AB

141 87 HUDDINGE
SWEDEN

Phone: +8 578 410 00

The former name of this manufacturer was Landis & Gyr AB / Landis & Staefa AB / Siemens Landis & Staefa AB.

2.3 Device Definition

PRV1
PRV2
EKL-X

These devices are the digital substations (Digital Under Central, DUC) that Siemens and their partners configure and program for different building automation projects. They have a configurable number of I/O's (digital input, analog input, analog output etc).

The device are further on named 'PRV'.

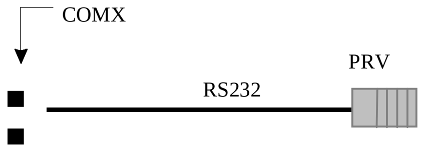
2.4 Communications Method

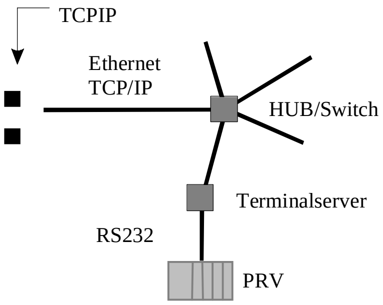
Serial RS-232, directly connected. This includes the Citect COMX or TCPIP driver, for access through terminal servers. The driver act as the master. Future versions of this driver will have possibilities to communicate with multidrop nodes.

2.5 Communications/Hardware Configuration

The PC is connected to the devices using a standard RS-232 port, RS-232 digiboard or similar equipment. Alternatively a terminal server and a hub/switch may be used.

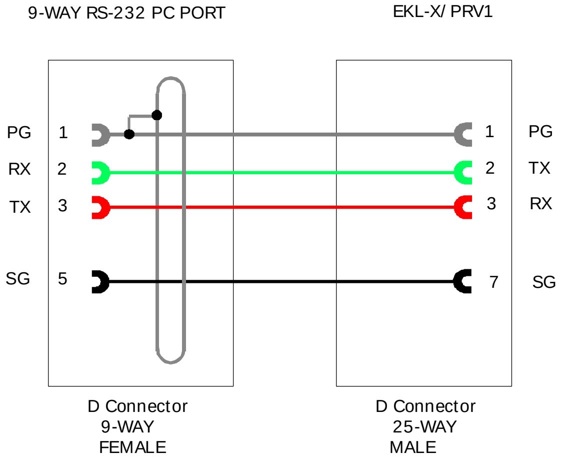
The PRV1 and PRV2 devices use 24 VAC, 50 Hz, and the EKL-X device 220 VAC, 50 Hz.

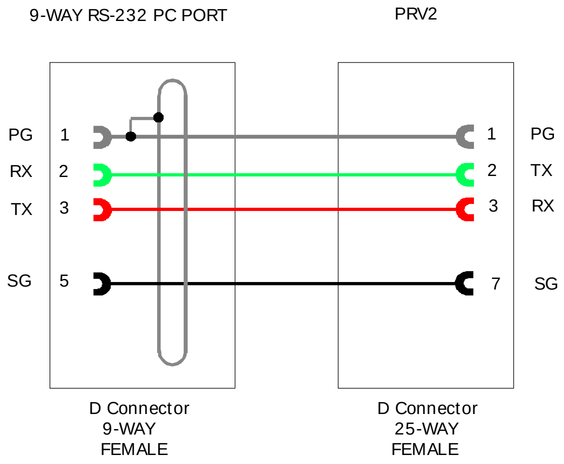




2.5.1 PC-PRV Cable

Use a standard data cable as follows:





If the distance between the device and the PC is above 15 m a short range modem should be used.

2.5.2 I/O Device Settings

The driver supports any baud rate available by the devices, but depending of the device load, number of programs, version etc the practical limit varies between 1200-9600 bps. The device response time is about 1200 ms at 1200 bps (EKL-X), and is decreasing to about 200 ms at 9600 bps (PRV2) for a single message, which is a request of about 5-6 values.

The devices are polled from the PC and responds with data on request. Some points should be event based. They will respond quicker and minimize the communication overhead, i.e. digital inputs, analog inputs with alarm limits etc.

The PRV could be set up using the Hyperterminal. This should be done of a person with knowledge of the PRV.

Connect the PRV directly to a COM-port, and set up the Hyperterminal parameters to the desired settings, i.e. correct COM-port (Direct COM1 etc), baud rate, no flow control etc.

First, set up the baud rate that should be used by the driver. Begin with 1200 bps for EKL-X/PRV1 devices, and 4800 bps for PRV2 devices. Use the command:

```
TTY2.BD
```

and the device will respond with the actual baud rate. Use:

```
TTY2.BD=1200
```

```
INITTY2
```

etc for initializing the new baud rate in the device. Now Hyperterminal must be changed to reflect the new settings.

To be able to use event the Event Task must be updated with a couple of program rows to echo out new values. Further on, the desired event based points must have their EVE-parameter set to 1. Polled points always have '/P' added to the tag address, i.e.

$040.HW Event based, need the EVE parameter set to 1. This type of points will be read once and will only wait for messages from the PRV that indicates that they are updated. There will be a slow cyclic update of these points anyway. It defaults to once per 30 minutes. The values will never be deleted from the cache once they are read.

If the PRV is heavy loaded these event based points will slow down the device further. Generally all digital inputs (module of ML type) and analog inputs (module of MW type) with alarm limits should be event based, i.e. all points that the Alarms Server will poll, this will reduce communications overhead to a minimum. If the device is of a PRV2 type, generally all used points could be event based. This will significally speed up the picture update time.

$040.HW/P Polled, does not need the EVE parameter set to 1, but will be significally slower is response when the picture is loaded. As a default these values are kept in the cache for 30 second before they are deleted, i.e. if you switch back to a previous picture before this time, the values will be read from the cache immediately. The cache will be updated in a cyclic fashion, with new requests as high priority.
This type of value should never be used in the Alarms Database (by the Alarms Server) because it will force the driver to push communications with the PRV. This will slow down program execution of this device, and could cause some loops to get very slow with unpredictable result. Remember that this equipment is based on a 20 year old construction.

First of all, set the EVE parameter of the points that should be event based:

```
$000.EVE=1
$040.EVE=1
$073.EVE=1
```

etc

Then change the event task (TSK253) must be changed to make an output of the point values when they change. The event task is started when any point changes at its EVE parameter is set to 1.

TSK253	To get into the editor mode
LIST	To view the program

New rows:

```
...
1020 EA=EVADR
1030 PRINT "<*"@EA.ADR"="@EA.HW";"@EA.ESTA
...
```

If the program is entered into a PRV2, the row should be entered as follows

```
1030 PRINT "<*"@EA.TA"="@EA.HW";"@EA.ERSTA
```

These rows must be entered in an appropriate place, and this must be made by a person that is an experienced PRV programmer to avoid problems with other programs.

The system parameter TSKPRP should be set to 2 for TSK253, and TSKTTY must be set to the appropriate port, i.e.

```
[253]TSKPRP=2
[253]TSKTTY=2
```

for TTY2 (X2).

To further increase performance, similar rows should be entered in the Alarm Event Task (TSK252). The address is get from PRIADR, i.e.

```
...
1020 PA=PRIADR
1030 PRINT "<*"@ PA.ADR"="@ PA.HW";"@ PA.ESTA
...
```

etc. The TSKPRP and TSKTTY settings should be the same as for TSK253. The TSK252 is only run by changes to points that has the PRIO set (not 0).

Always check the Kernel when a new device is compiled and runs in the project. Do a DEBUG xxxx ALL at the appropriate port, to see what happens. There should be no communication in normal mode (i.e. no picture is shown). Alternatively, the Driver Statistics, parameter *Active Polls* (use 'V' to show the verbose info), shows the number of active cached points that are polled.

Never create the same point both polled and event based although this is possible, this may cause disruption in the error handling if the point is bad.

2.6 Contacts

www.beijer.se

support@beijer.se

3. Protocol Requirements

3.1 Introduction

This section documents all the requirements of the protocol itself.

3.2 Initialising the Port

No special initialisation is necessary.

3.3 IO Device Online Test

The drivers’ response to CTDRV_INIT_UNIT and CTDRV_STATUS_UNIT is to issue a Ctrl-E transmission to the device. This will return the text 'Ctrl/E' and exit any programming modes in the device.

3.4 Message Structure

The message structure has the following formats :

Driver request:

"<?xx="$000.HW";"$001.HW";"$002.HW";"

Valid response:

<?xx=23.3;12.5;19.71;

The request may be up to 75 bytes wide. Response messages may be up to 80 bytes wide. The driver always calculates the length of a request and limits it to 75 bytes, or 10 values. Generally a request is about 5-6 values.

<?	Indicates to the driver (in the response) that this is a question
xx	The packet number. This is always increased by one. At 99 it is rolled over to 00.
=	Indicates to the driver that the following text is the first value that is requested.
$000.HW..$002.HW	The requested points.
";"	Value delimiter.

3.4.1 Event messages

<*$000=23.3;1;

<*	Indicates to the driver that this is an event
$000	The event address. The cache is scanned for this address. If the address is found, the parameter is checked. Only the main value is updated (i.e. ML, MW, SB, ZW, ST and HW).
=	Indicates to the driver that the following text is the new value.
23.3	The new value.
";"	Value delimiter.
1	The error status (ESTA/ERSTA). If this parameter at this address is found in

	the cache it will be updated.

3.5 Data Format

The data format is 8-bit ascii. Any ascii character from 0x20 - 0x7e is valid in the message.

3.6 Validity check

The responses are checked for validity (i.e. they must be numbers if it is a cached value). The packet number must be correct. Check the *Bad Packets* parameter in the statistics. This should always be very low. Not more than one bad packet per 1000 should be allowed. If there is a high number of re-transmission, the baud rate should be lowered. If there are a lot of bad packets, the baud rate should be lowered and the idle time between messages should be increased.

3.7 Extended Protocol Description

The fully protocol description is found in the *PRV Programming Manual*, available by Siemens Building Technologies.

3.8 Error Handling

3 consecutive timeouts at read or write requests will put the device off-line (defaulted to 2 retries). The reads could be made by cached values, or directly by Citect.

If the device is off line, it will still be polled until the cached values are erased. The default time for this is 30 seconds, but could be changed in CITECT.INI (*CacheLiveTime*).

The device will return to the on line state as soon as the driver gets any response from it. Citect will put the device on line when next init unit command is sent. This is done every 30 seconds and is called *WatchTime*. That parameter could be changed in CITECT.INI.

3.9 Bad points

If a point is not defined (i.e. ERROR 87) it will cause the device to not return any values at all immediately following the bad point. The driver marks the point as bad and ignores it for 11 consecutive reads. The reason why 11 reads is chosen is that a new request for the other points in the same request as the bad point will be built and sent. If there are more bad points the device will only respond with the first bad point, therefore it is necessary to keep the first bad point away for at least 10 reads which is the maximum points per request. For event based values, the device will send a DIAG message with the point address. This will be treated in a similar way. When the point is ok again (i.e. sensor fault) it will be reported back to Citect as a normal point.

The ERROR and DIAG numbers are treated as follows:

83-91	Address Out Of Range. Will cause a hardware alarm. The device will not go off-line.
92-99	Bad Data. Will cause a hardware alarm. The device will not go off-line.

All other ERROR messages and DIAGs are ignored. If the device does not respond, retries are made (see *Error Handling*).

4. General Driver description

4.1 Cache

This driver is of the ‘Front-End-Back-End’ type, and therefore has a cache list of all polled and event based values. Once a value is questioned by Citect (because a new picture is opened, the Trend Server request new values, the Alarms Server is starting etc) a cache element is requested for it. If there is any free cache elements in the cache list, that cache element is used. If there is no free cache elements left, the driver will reallocate the cache memory to fit 16 new cache elements. Cache elements are never freed (i.e. memory usage never shrinks), when cache elements are deleted they are only marked as 'free' and will be reused next time it is needed.

Event based values are only read once and will never be marked as free, even if Citect never more asks for them. This is because the driver keep track of incoming messages, and updates the cached values. Next time Citect ask for the value, it will be read from the cache which will significally improve speed. As described above, the event based values will be updated with a default interval of 30 minutes, to ensure data integrity.

Polled values will be read as fast as possible. The driver will continue to read polled values in a never ending loop, until the *CacheLiveTime* has expired for them. Every time Citect ask for a value, the *CacheLiveTime* timer is reset, which will cause the cache element to never expire unless Citect stops asking for it. The default cache live time is 30 seconds.

The size of the cache memory is 16 bytes per value. This is allocated from the linear 32-bit memory space.

This means that Citect indicates a very high channel usage, though the default is to handle 10 outstanding DCB’s at one time.

4.2 One Shot

Some values are never cached, these values are named 'One Shots'. They are read directly from the PRV and are never passed through any temporary memory storage. These values are always read as one per message, because the PRV will not handle any multiple requests.

Example of this type are SYSVER, TIME, DATE etc.

The driver will split the requests if there are both cached values and one shots that should be read, i.e. every second request will be a one shot request. The other requests will be for the cache values. That means that both types have equal priority.

4.3 First time reads

When Citect starts and the Alarm Server begins to request values, a first explicit poll is made of the values requested, even though they are defined as event based. This is because the driver has no idea of the previous state of the channel. Once the first poll is made, the normal event handling takes over, and the driver responds with values directly from the cache. This will cause the response to be poor immediately following a startup until the cache is filled.

4.4 Request Performance

Due to the nature of the PRV protocol, and the possibility to cached reads, Citect automated blocked reads are turned off. The medium response time for such a request is about 0,2-1,2 seconds per message. Every message contains about 5 values which gives a 'response per value' time of about 50-250 ms. For one shots (see below) the response time will be 0,2-1,2 seconds.

4.5 PRV Performance

The PRV protocol has a fairly simple master slave structure, and communicates with the maximum speed of 1200-9600 bps. This means that it has a generally bad performance when polled values are used.

4.6 Writes

Write requests are never cached, they are performed at once. However, Citect may push a lot of write DCB's at the same address if the user use a Slider etc to set the value, and let it use 'Continous update of tag'. The driver could get a lot of consecutive write requests to the same address, and will handle them in the order they arrived.

4.7 Citect Cache

Citect do not need to do any cache, it will be handled by the driver.

4.8 PRV.DBF

The database specification contains a dynamic number of parameters. The first one is explained:

TEMPLATE	UNIT_TYPE	RAW_TYPE	BIT WIDTH	LOW	HIGH	COMMENT
$%O(0).HW%*256	0x0000	2	32	0	377	<7/$%03o.HW>HauptWert
$%O(0).HW/P%*256	0x8000	2	32	0	377	<7/$%03o.HW>HauptWert

This row describes the converted Main Value, which is valid for all physical points. Both the event based and the polled version are shown above.

4.8.1 TEMPLATE

The address of the parameter is multiplied by 256 to avoid Citect blocking. Blocking is made internally by the driver, because the internal cache will have lower performance if blocking was made by Citect.

The address is always entered in octal format, which is standard for the PRV, i.e. '$077.HW' etc.

4.8.2 UNIT_TYPE

The PRV has a lot of different parameters, of which just a few are in the PRV.DBF file. New may be entered without changing the driver DLL. The bits 0-11 of the UNIT_TYPE are used as a counter for the different parameters which gives the possibility to handle up to 4096 different parameters.

Bits 12-15 are used as flags to indicate:

Bit 15 Polled. If set this value will be polled when cached (Citect asks for it).
Bit 14 One Shot. This value will never go into the cache.
Bit 13 No Address. The request will never be created with any address.

The lower 12 bits are always masked of when the templates are read in (see below) and cross checked. If there are any unit types that are the same, the first found request template will be used. This will also have the effect that if two tags are set up with the same unit type and address, and one is polled and the other event based, the first type will take precedence. Therefore the event based unit types should be defined first in the PRV.DBF.

To set up two tags in this way should be considered as an engineering fault, but is not checked for.

4.8.3 RAW_TYPE, BIT_WIDTH

The most used raw type in the PRV is real. Almost every one shots are strings.

4.8.4 LOW, HIGH

Though the PRV always use octal addressing, the limits are set to 0 respective 377. The upper limit is however depending of the device type.

4.8.5 COMMENT

The comment row contains a template for the message build function, which describes the look of the request.

<7/$%03o.HW>

The first field, 7, is the normal request length. This is used to keep track of the message length, so it does not exceed 75 characters.

The second field, $%03o.HW, is the template, and it is used in a sprintf C function to create a request. A flag in the UNIT_TYPE indicates if the address should be used. Some parameters does not require the address (i.e. TIME etc).

4.9 Request Template Allocation

At driver startup, the DLL checks if it has been loaded before. If it has not, it will allocate a global memory array for the request templates. These will then be read from the PRV.DBF file and inserted into the memory array. This memory array is then used by all instances of the DLL to avoid unnecessary memory usage, and will be freed when the last channel is closed.

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

PRV

5.3 Ports Form

5.3.1 Baud Rate

9600 bps, or lower. The baudrate is set in the different devices.

5.3.2 Data Bits

8

5.3.3 Stop Bits

1

5.3.4 Parity

None

5.3.5 Special Opt

None

5.4 IO Devices Form

5.4.1 Protocol

PRV

5.4.2 Address

The address field has different formats depending of the device hardware setup. The field has the following format:

[address] [-dnnn] [-tnnn]

address The device numeric address or ‘SYSTEM’. If a multidrop RS-485 network with KD485-ADE is used, the address is needed. The address spans from 1 to 126. Only up to eight devices are supported by each RS-485 network. If more devices will be connected, then the network must be splitted into two or more. The performance will decrease if many devices are connected to the same network.

-dnnn This parameter tells the driver that it should use another delay between messages instead of the default 500 ms delay. This may be done to improve speed to devices that has low loading, or are of a faster type (PRV2). This parameter could also be used to increase the delay.

-tnn The timeslot timeout in the KD485-ADE unit, if a multidrop RS-485 network is used. This timeout must match the KD485-ADE timeout exactly, otherwise data corruption will occur.

Example:

10 -t50 -d300 Multidrop RS-485 network. The KD485-ADE timeslot is 50 ms, and its address is 10. The message delay is 300 ms.

-d400 Single PRV connected directly to a COM-port or a terminal server TCP port. The message delay is 400 ms.

System This device is used to communicate with the PRV when it is up- and downloading TSC's (using PRVView). The reason is that the communication is shut down for that device during the transparent session. PRVView uses the system device 'transparent' tag that points to the PRV device, though the system device never goes off line.

Multidrop RS-485 network with KD485-ADE will be implemented at request, i.e. the above [-dnnn] [-tnnn] are not used for now.

5.5 Pulldown lists Help

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

TYPE	DATA	FILTER
PROTOCOL	**PRV**	

5.6 IO Device Data Types and Formats

IO Device Data Type	**Valid Range**	**Citect Data Type**	**Description**
Float	3.4E +/- 38	REAL	7 digits
Integer	-32768 - +32767	INT	
Boolean	0 – 1	DIGITAL	
String		STRING	Mostly used for one shot values, as TIME and DATE etc.

5.7 IO Device Function Blocks and their Parameters (PRV.DBF)

IO Device Type	Data type	PRV Description
$xxx.HW $xxx.HW/P	Real	The physical point xxx's main value. This addressing format is valid for all physical point types and should be used as often as possible, though it minimizes the risc of addressing faults. The only reason to use any of the MW/ST/ML/SB/ZW parameters, are when some other data type than real is desired. For the MW/ST/ZW types, which always are real, the HW parameter should be used.
$xxx.MW $xxx.MW/P	Real	An analog inputs' main value.
$xxx.ST $xxx.ST/P	Real	An analog outputs' main value.
$xxx.ML $xxx.ML/P	Digital	Digital inputs' main value.
$xxx.SB $xxx.SB/P	Integer	Digital outputs' main value. Depending of the module type used, this could be other values than 0 and 1 (i.e. three-stage modules etc).
$xxx.ZW $xxx.ZW/P	Real	Counter inputs' main value.
VIPxx.VP VIPxx.VP/P	Real	Virtual point. Used internally in the PRV programs to set point values, calculated states, transporting the outdoor temp to any other device etc.
RGBxx.SPV	Real	Regulator blocks' set point value.
RGBxx.PR	Real	Regulator blocks' process value.
RGBxx.COU	Real	Regulator blocks' calculated output.
RGBxx.KINT	Real	Regulator blocks' integration value.
RGBxx.KREG	Real	Regulator blocks' amplification value.
RGBxx.KDIF	Real	Regulator blocks' derivation value.
PLTxx.PSTA PLTxx.PSTA/P	Integer	Plant points status.
PLTxx.MAN	Digital	If set to 1 tells the PLT point that it should use the MCOM parameter to calculate the PSTA (Manual Command).
PLTxx.MCOM	Integer	Manual Command. Will be transferred to the PSTA when the MAN parameter is set to 1. If MAN is 0, then this parameter will reflect the PSTA.
PLTxx.PAR1 PLTxx.PAR2 PLTxx.PAR3 PLTxx.PAR4 PLTxx.PAR5 PLTxx.PAR6 PLTxx.PAR7 PLTxx.PAR8	Real	Plant points parameters. Only PAR1-4 are valid for EKL-X / PRV1.
ZONxx.HW ZONxx.HW/P	Real	Zone points main value (Set Point). Uses to be set by the WDC.
ZONxx.PDFV	Real	Zone points default value.
ZONxx.OCC	Integer	Zone points occupation status.
$xxx.ESTA $xxx.ESTA/P	Integer	Any points error status (EKL-X / PRV1)
$xxx.ERSTA $xxx.ERSTA/P	Integer	Any points error status (PRV2).
$xxx.AST	Integer	Inputs' state where it should report an error condition in ESTA (EKL-X / PRV1).
$xxx.EST	Integer	Inputs' state where it should report an error condition in ERSTA (PRV2).
$xxx.LOL $xxx.HIL	Real	Inputs' low and high limits, before they report an error condition in the ESTA/ERSTA parameter. The delay is always considered before any

		error condition occurs/resets.
$xxx.DL01 $xxx.DL10	Integer	Inputs' delay before the input generate an error (ESTA change) for 0->1 and 1->0 transitions (EKL-X / PRV1). This is not in seconds.
$xxx.DEL01 $xxx.DEL10	Integer	Inputs' delay in seconds before the input generate an error (ERSTA change) for 0->1 and 1->0 transitions (PRV2).
$xxx.INV	Digital	Digital inputs' (ML) inverted state. If set to 1 the ML input at the same address will be inverted.
@RWIyy.APHS	Integer	RWI Handswitch.
@RWIyy.SPH	Real	RWI Setpoint Heating.
@RWIyy.SPC	Real	RWI Setpoint Cooling.
@RWIyy.ASPH	Real	RWI Calc Setpoint Heating.
@RWIyy.ASPC	Real	RWI Calc Setpoint Cooling.
@RWIyy.TPSAMI	Real	RWI Supply Temp Min.
@RWIyy.TPSAMA	Real	RWI Supply Temp Max.
@RWIyy.TPR	Real	RWI Temp Room.
@RWIyy.TPSA	Real	RWI Temp Supply.
@RWIyy.TPOS	Real	RWI Temp Outdoor.
@RWIyy.TPFR	Real	RWI Temp Frost.
@RWIyy.AOCOOL	Real	RWI Analog Output Cool.
@RWIyy.AOHREC	Real	RWI Analog Output Heat Recovery.
@RWIyy.AOHEAT	Real	RWI Analog Output Heat.
@RWIyy.DOHEAT	Digital	RWI Pump Heat.
@RWIyy.DOCOOL	Digital	RWI Pump Cool.
@RWIyy.DOFAN1	Digital	RWI Fan Step 1.
@RWIyy.DOFAN2	Digital	RWI Fan Step 2.
@RWIyy.ALARMA	Digital	RWI A-alarm.
@RWIyy.ALARMB	Digital	RWI B-alarm.
@RWIyy.DIFLOW	Digital	RWI Flow Indicator.
@RWIyy.DIOLH	Digital	RWI Pump Failure.
@RWIyy.DIOLFN	Digital	RWI Fan Failure.
@RWIyy.DIAUX1	Digital	RWI Aux Input 1.
@RWIyy.DIAUX2	Digital	RWI Aux Input 2.
@RWIyy.DIFIRE	Digital	RWI Fire Detector.
@RWIyy.DITC1	Digital	RWI External Time Switch.
@RWIyy.SOMOD	Digital	RWI Op mode.
@RWIyy.SOPST	Digital	RWI Op state.
@RWIyy.AOPST	Digital	RWI Op state.
@RWIyy.WHY	Digital	RWI Reason.
@RWIyy.EXSPS	Digital	RWI SPV Correction.
PLTxx.MACT	Digital	If set to 1 tells the PLT point that it should use the MCOM parameter to calculate the PSTA (Manual Command).
@RWIyy.ERSTA	Integer	RWI Error State
@Diyy.DI	Integer	DI Input
@Diyy.ERSTA	Integer	DI Error State
@DOyy.DO	Integer	DO Output
@DOyy.RDO	Integer	DO Returned state
@DOyy.ERSTA	Integer	DO Error State
@AIyy.AI	Real	AI Input
@AIyy.ERSTA	Integer	AI Error State
@AOyy.AO	Real	AO Output
@AOyy.ERSTA	Integer	AO Error State
TRANSPARENTzz	String	The transparent channel for PRVView
TIME	String	The devices' time. The format is HH:MM:SS. I.e. 23:40:36.
DATE	String	The devices' date. The format is DD-MMM-YY. I.e. 14-OCT-99.
SYSVER	String	The devices' program version number, i.e. 5472/5480 etc for EKL-X and 6 etc for PRV. One shot. Polled.
SYSVER/R	Real	The devices' program version number, i.e. 5472/5480 etc for EKL-X

		and 6 etc for PRV. One shot. Polled.
SYSVER/I	Integer	The devices' program version number, i.e. 5472/5480 etc for EKL-X and 6 etc for PRV. One shot. Polled.

/P Indicates that the value is polled, not event based. This is only true for HW/ML/SB/MW/ST/ZW/ESTA/ERSTA. All other points are always polled, even though /P is missing. This is for engineering conveniance.

xxx Physical point address. In octal addressing format, i.e. 000-007, 010-017, 170-177 etc.

yy RWI (AeroGyr) FLN address.

zz The PRV device number that is to be used in PRVView communication through this tag. This tag(s) should *always* be created at the system device.

It is possible to add own parameters to this database. They should then range from UnitType 0x8800..8ffff (about 2048 different parameters). See the PRV.DBF section above for a thorough description of the UnitType.

5.8 PROTDIR.DBF

TAG	FILE	BIT_BLOCK	MAX_LENGTH	OPTIONS
PRV	PRV	16	256	0x047

5.9 Parameters and INI options

5.9.1 Standard Parameters

Block	1	(Citect blocking is always prohibited)
Delay	200 ms	
MaxPending	20	
Polltime	0 ms	(interrupt driven)
Timeout	3000 ms	
Retry	2	
WatchTime	30 s	

5.9.2 Driver Specific Parameters

All PRV specific parameters are located in the section ‘PRV'

PARAMETER	DEFAULT	DESCRIPTION
CacheLiveTime	30000 (ms)	The time a cache element should be alive in the cache when Citect stops asking for it before it is marked as unused. This is only true for polled values. Event based values never expire.
CacheCleanTime	500 (ms)	The time that elapses between cache cleaning. When this timer

		expires, the driver scans through all cache elements and decreases the CacheLiveTime timer for the polled cache elements. The CacheCleanTimer event also will do a reply for DCB's directly from the cache, if the values that Citect asks for are valid and in the cache.
EventRefreshTime	1800000 (ms)	The event based values will be cyclic updated with this interval. The default is 30 minutes. If this is set to a very low value, all event based values will be polled, just like normally polled values.
AutoEvent	1	N*ot used for the moment.*
SetRTCTime	02:30:00 00:30	The time when the RTC (date and time) will be set in the device. If set to an invalid time (>23:59:59), no RTC updates will be made. If this parameter is entered using a HH:MM format, i.e. 00:30, a interval is used instead of a time of day. The time expirations are checked every minute, and could differ somewhat from desired time Ex. if the RTC should be set at 02:30:00 and the expiration check is made 02:30:20 then the time will be set 02:30:20. The time will of course be set to the actual time (02:30:20). *This is not used for the moment, and the RTC is not set automatically by the driver.*
SetRTCAtInit	0	If set to 1, it will cause the driver to send out a time update before the devices are set on line. *This is not used for the moment, and the RTC is not set automatically by the driver.*
ConnectTimeOut	15000	The time the driver will wait for a TCP connection to be established before the connection will be considered failed.
KeepAliveTime	0	The time the driver will wait for any response or packet whatsoever from the PLC before it decides to send a Ctrl/E to see if the PLC still is connected. If only event based values are read by Citect from the driver cache, no communincation between Citect and the PLC occurs unless the PLC sends any event updates. During this condition there is no way for the driver to know if the PLC still is connected to Citect unless it sends a cyclic KeepAlive message. The time 0 means that no KeepAlive Ctrl/E's are sent at all.
ClearCacheAt-TransparentExit	1	If set to 1, it will cause the driver to re-read all cached values before starting the normal poll sequence. When the transparent mode is entered (using the external time switch program) all events from the DSS will be transferred directly to the external program and will be lost, therefore, a re-read is necessary.
AutoEvent	1	Not used in this version
ExtendedTrace	0	If set to 1, it will cause the driver to echo all AppendDCB calls and Reply callbacks in the Kernel Main window. This is true only if any debug mode is entered.
EventTrace	0	Views extended info in Kernel for an incoming event when the debug mode is set to ALL.
ShowTraceInHex	0	If set to 1, it will change the Citect default trace format (ascii/hex mixed) to only hex with the ascii characters in a separate column.
WriteTraceLog	0	If set to 1, the driver will create log files in the root of C: and write all enter/exit steps in any procedure in the driver. The log file name is set to C:\PRV_*nnnnnn*.LOG, where *nnnnnn* is the channel number (in no particular order). *This parameter should only be used if advised by the support, and*

		will significally slow down the driver. No checking of file sizes are made, and they may fill up the disk space with no furhter notice.
MaxRequestsPer-Row	10	Sets the maximum of parameters that are asked in each packet. The default is seldom reached, mostly each packet end up with 5-6 parameters. *This parameter should only be used if advised by the support, it could significally slow down the driver.*
IgnoreAllErrors	0	If set to 1, it will cause the driver to not recognise any DIAG or ERROR message from the PRV. The driver will behave like it never got any packet in return and will ask until the retry counter has expired. *This parameter should only be used if advised by the support, it could significally slow down the driver.*
NeverSetUnitOff-Line	0	If set to 1, it will cause the driver to never return any unit offlines, not even if the unit does not respond to normal data requests. This parameter does not affect the *INIT_UNIT* behaviour. *This parameter should only be used if advised by the support, it could significally slow down the driver.*

5.10 Driver Specific Errors

Driver Error Code	Mapped to (Generic Error label)	Meaning of Error Code

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

5.12 Debug Messages

Shows the outgoing/incoming data packets in hex. If error report level is set to ERROR the driver will report errors, i.e. if any erroneous packets are received.

5.13 Stats Special Counters

Number	Label	Purpose/Meaning of this counter
0	DCB Requests	The total number of read requests made.
1	Inbuffer max	Maximum used bytes in the input buffer for the CPU service

		function (before they have been handled by the packet receive routine).
2	Invalid packets	Packets received that was not valid (invalid packet number etc).
3	Packets received	Total number of received packets.
4	Cache hits	Number of DCB's that has been read directly from the Cache (did not cause a physical read).
5	Cache hit rate (%)	The hit rate (Cache hits / DCB Requests x 100).
6	Sent packets	Number of sent packets.
7	Resent packets	Resents packets caused by timeout.
8	One shots	Number of packets that was sent due to a one shot request.
9	Actual cache alloc's	The number of cache elements that are in use just for the moment. These include event based cache elements that never expire.
10	Trashed elements	Cache elements that where erased from Citect because there was to slow response from the driver. This elements where erased before they returned any value (default timeout is 30 seconds).
11	Outstanding DCB's	The momentary value of outstanding DCB's that is serviced by the driver (default max is 20).
12	Max outst. DCB's	The top value of DCB's that has been requested by Citect to be serviced by the driver at the same time.
13	Phys. read time (ms)	The time the PRV consumed to service the last request.
14	Total DCB's	Total number of DCBs ever served.
15	Bad Data	If a incoming packet had correct packet number and preambles/postambles, but had a bad data field, this counter will increase. This counter also increases if an event or poll for a bad point is made (i.e. sensor failure, not configured etc)
16	Total cache alloc's	The total number of cache allocations that ever was made.
17	Cache Mem Size	The total size of the unit cache. This grows in increments of 32 cache elements and never shrinks.
18	Sent trp bytes	The number of bytes sent using the transparent mode (i.e. time switch program)
19	Received trp bytes	The number of bytes received using the transparent mode (i.e. time switch program)

5.14 Hints and Tips

It is strongly recommended to use event based values if this is possible. Take care so the PRV don't get to busy handling just communication.

A very common problem is the addressing fault. If the PRV gets a request message of 5 values, and one of them doesn't exist physically or are of some other type, the PRV will respond with an error code indicating that there are some kind of addressing fault. The driver is not able to sort out the value that caused the fault because the PRV does not answer at any of the requested values. The DCB response to Citect will be the DRIVER_ADDRESS_OUT_OF_RANGE error.

By using the Kernel DEBUG xxxx ERROR command the erroneous request answers will be shown, use them to track the bad address.

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 programmers 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

The following are points 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 (Statistics report)

- 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 writeable in this specification.

  - OK

- The driver reads and writes all data formats supported by the protocol, ie DIGITAL, INT, REAL, STRING.

  - 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 to this driver.

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

  - OK

- If a multidrop or network protocol and if the hardware is available then the protocol should be tested with more than one IO Device connected.

  - not yet tested

7. Performance Testing

7.1 Introduction

Tests which give some indication of the drivers 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

Because this driver is of the ‘Front-End-Back-End’ type, and due to the nature of the PRV protocol, it does not support Citect blocking. The channel read always read 16 channels, wich is just about the maximum for one request. This has been tested and found to be the fastest way to get required data from the device.

8. References

8.1 References

The fully protocol description is found in the *PRV Programming Manual*, available by Siemens Building Technologies.