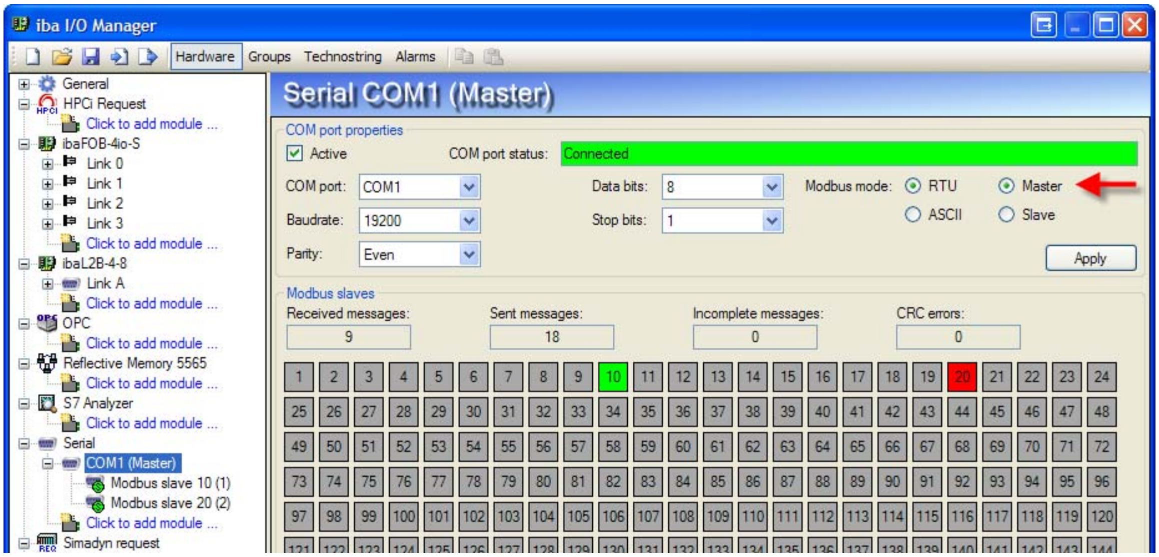
New features in ibaPDA v6.15.0

Changed: 2008-02-07 : dkopp

1 Modbus serial master

The modbus COM port can be switched between master and slave mode via radio buttons.



Serial COM1 (Master)

COM port properties

☑ Active COM port status: Connected

COM port: COM1 Data bits: 8 Modbus mode: ● RTU ○ ASCII ● Master ○ Slave

Baudrate: 19200 Stop bits: 1

Parity: Even

Modbus slaves

Received messages: 9 Sent messages: 18 Incomplete messages: 0 CRC errors: 0

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96
97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144

If the COM port is in slave mode then pda responds to requests from the modbus master. If the COM port is in master mode then pda will periodically send requests to the modbus slaves that have been configured.

The screenshot shows pda as modbus master with 2 configured slaves: 10 and 20. A slave can be in following states:

State	**Master mode**	**Slave mode**
Connected (green)	The slave responds to the periodic requests from pda.	The master is sending periodic requests to pda.
Disconnected (red)	The slave does not respond to periodic requests from pda.	The master doesn't send any requests to this slave.
Disabled (gray)	This slave is not configured.	This slave is not configured.

See the document about ibaPDA 6.14.0 for more information about the serial interface and the modbus slave module.

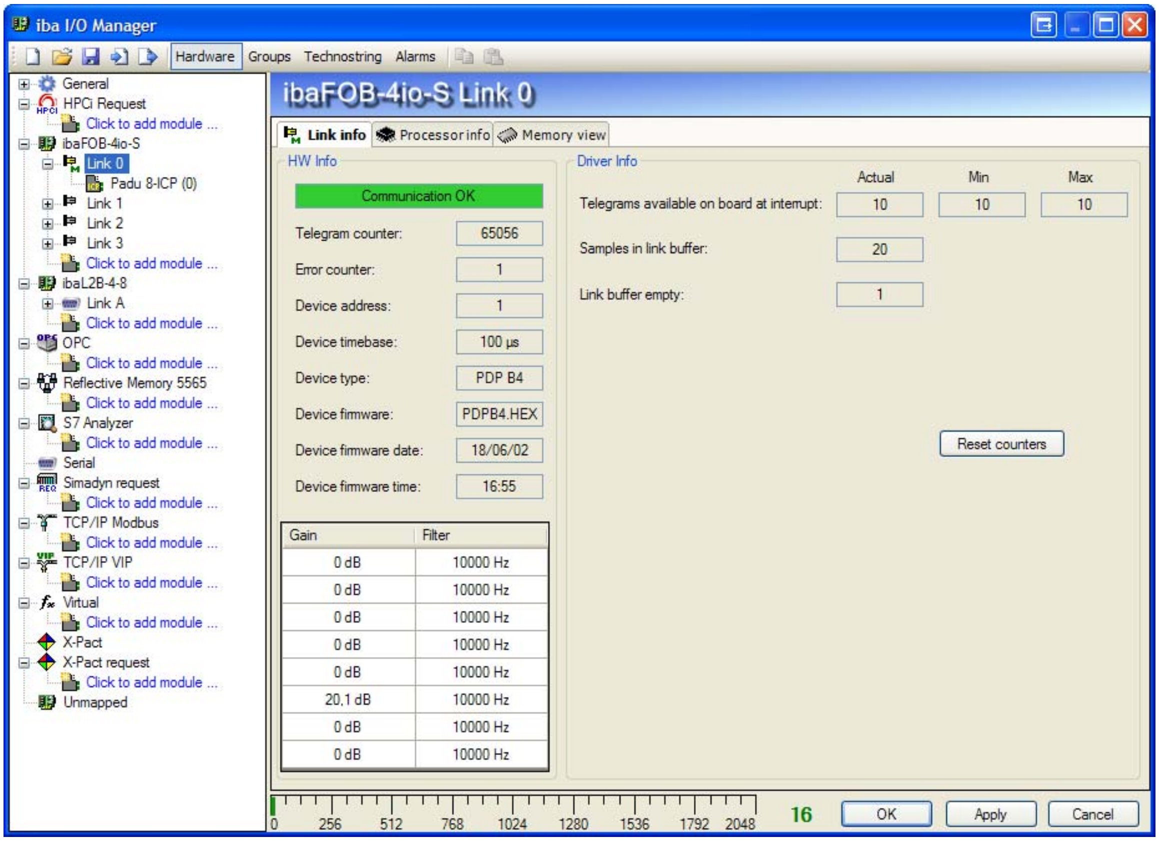
2 X-Pact request interface

See separate document sw_man_ibaPDA-XPact-Request_V1.0_en_A4.doc

3 FOB-M mode

ibaPDA supports *synchronous* FOB-M mode. This mode is only supported on ibaFOB-4io-S and ibaFOB-io-S boards. It is not supported on old FOB boards and on FOB-X boards.

The FOB link is switched to FOB-M mode automatically when you add a Padu8-M, Padu8-ICP or Padu16-M module to the link. The following screenshot shows a link in FOB-M mode.



ibaFOB-4io-S Link 0

HW Info

Communication OK

Telegram counter: 65056

Error counter: 1

Device address: 1

Device timebase: 100 µs

Device type: PDP B4

Device firmware: PDPB4.HEX

Device firmware date: 18/06/02

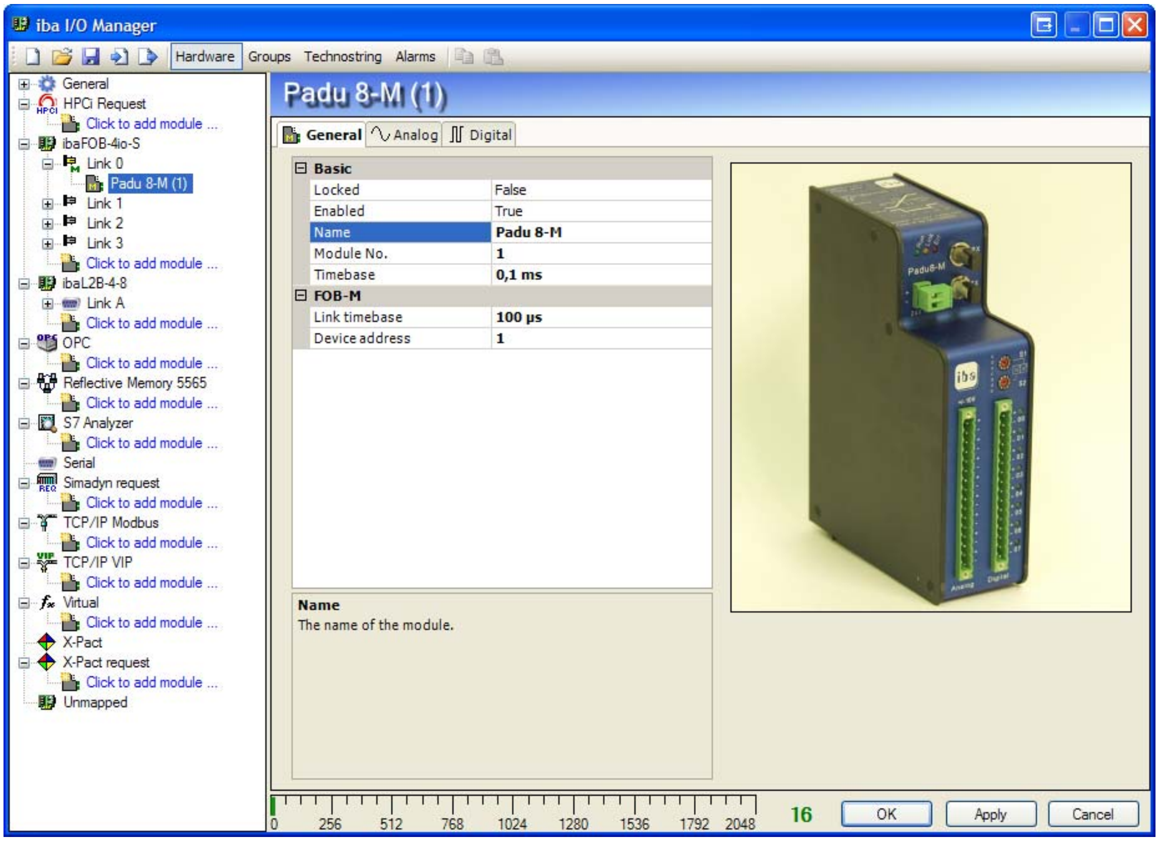
Device firmware time: 16:55

Gain	Filter
0 dB	10000 Hz
0 dB	10000 Hz
0 dB	10000 Hz
0 dB	10000 Hz
0 dB	10000 Hz
20,1 dB	10000 Hz
0 dB	10000 Hz
0 dB	10000 Hz

Driver Info

	Actual	Min	Max
Telegrams available on board at interrupt:	10	10	10
Samples in link buffer:	20		
Link buffer empty:	1		

You can see the status of the link, some counters, some information about the connected device and the configured gains and filters. The driver info groupbox contains some counters that are only for internal use (debugging the driver).



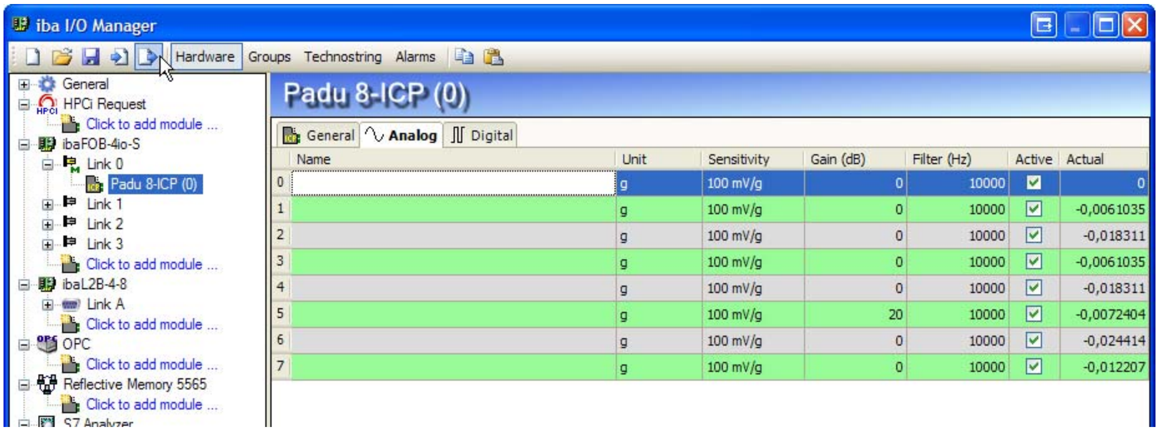
Basic	
Locked	False
Enabled	True
Name	Padu 8-M
Module No.	1
Timebase	0,1 ms
FOB-M	
Link timebase	100 µs
Device address	1

Name
The name of the module.

ibaPDA only supports the synchronous FOB-M mode. This means that all FOB-M devices connected to the pda system all use the same clock. The FOB-M devices sample their inputs all at the same time. You have to specify the period of this clock by filling in the link timebase on a FOB-M module. All FOB-M modules use the same link timebase. If you want to measure one of the devices at a slower rate then you can set the timebase property of the FOB-M module to a multiple of the link timebase.

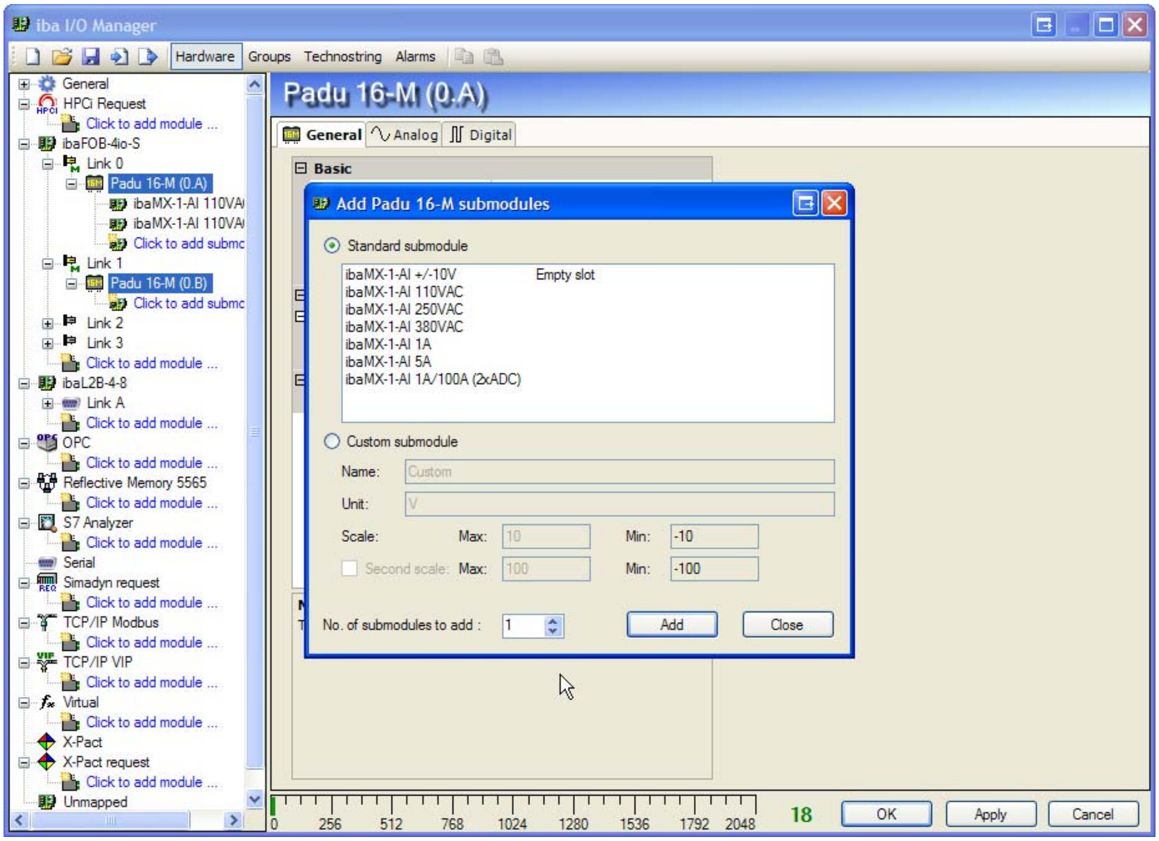
You also have to specify the device address on the Padu8-M and Padu8-ICP modules.

The Padu8-ICP module supports gains and filters on the analog inputs. These can be specified in the analog signal grid.



	Name	Unit	Sensitivity	Gain (dB)	Filter (Hz)	Active	Actual
0		g	100 mV/g	0	10000	☑	0
1		g	100 mV/g	0	10000	☑	-0,0061035
2		g	100 mV/g	0	10000	☑	-0,018311
3		g	100 mV/g	0	10000	☑	-0,0061035
4		g	100 mV/g	0	10000	☑	-0,018311
5		g	100 mV/g	20	10000	☑	-0,0072404
6		g	100 mV/g	0	10000	☑	-0,024414
7		g	100 mV/g	0	10000	☑	-0,012207

The Padu16-M module has 2 links. Each link can contain up to 8 submodules. In ibaPDA you can select the submodules by clicking the “Click to add submodule” node in the tree. This will open a small wizard where you can add the standard submodules or add a custom one.



● Standard submodule

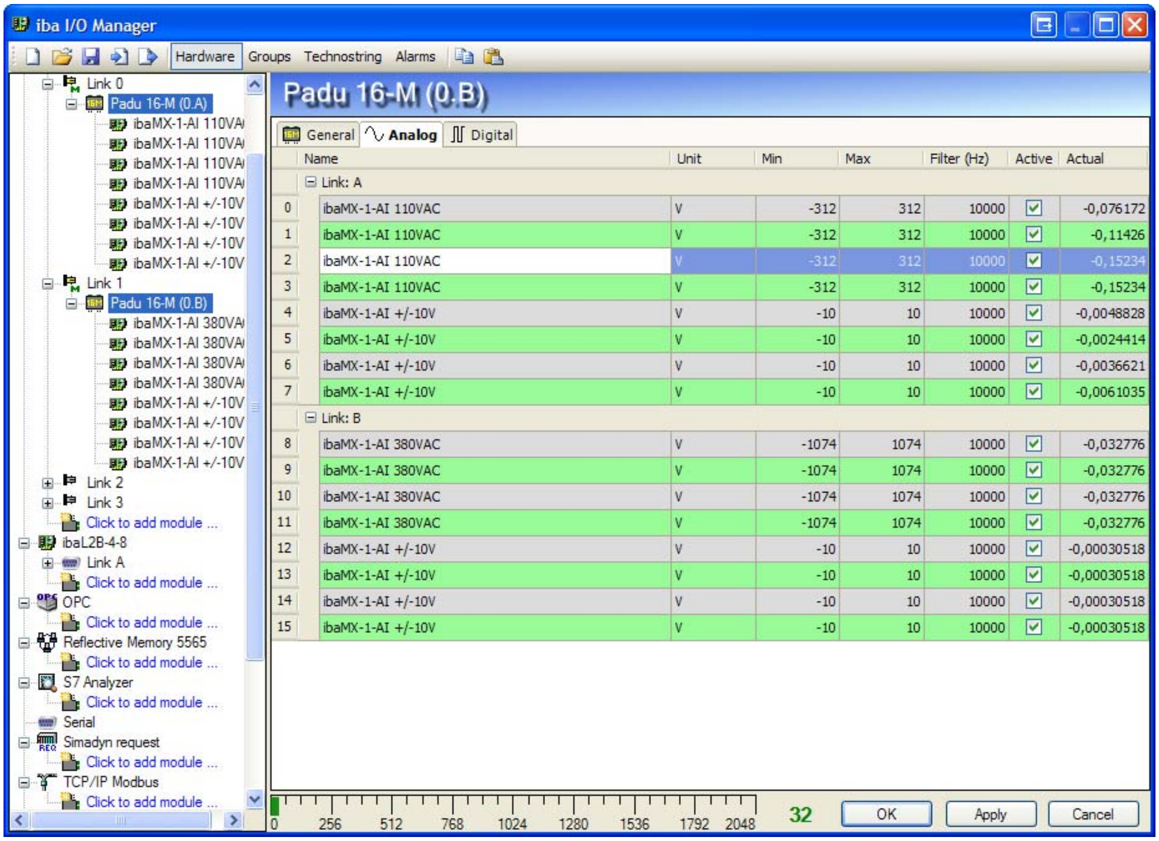
ibaMX-1-AI +/-10V Empty slot
ibaMX-1-AI 110VAC
ibaMX-1-AI 250VAC
ibaMX-1-AI 380VAC
ibaMX-1-AI 1A
ibaMX-1-AI 5A
ibaMX-1-AI 1A/100A (2xADC)

○ Custom submodule

Name: Custom
Unit: V
Scale: Max: 10 Min: -10
□ Second scale: Max: 100 Min: -100

No. of submodules to add : 1

Each submodule corresponds to 1 analog input. The correct scaling is applied automatically.



	Name	Unit	Min	Max	Filter (Hz)	Active	Actual
	⊟ Link: A						
0	ibaMX-1-AI 110VAC	V	-312	312	10000	☑	-0,076172
1	ibaMX-1-AI 110VAC	V	-312	312	10000	☑	-0,11426
2	ibaMX-1-AI 110VAC	V	-312	312	10000	☑	-0,15234
3	ibaMX-1-AI 110VAC	V	-312	312	10000	☑	-0,15234
4	ibaMX-1-AI +/-10V	V	-10	10	10000	☑	-0,0048828
5	ibaMX-1-AI +/-10V	V	-10	10	10000	☑	-0,0024414
6	ibaMX-1-AI +/-10V	V	-10	10	10000	☑	-0,0036621
7	ibaMX-1-AI +/-10V	V	-10	10	10000	☑	-0,0061035
	⊟ Link: B						
8	ibaMX-1-AI 380VAC	V	-1074	1074	10000	☑	-0,032776
9	ibaMX-1-AI 380VAC	V	-1074	1074	10000	☑	-0,032776
10	ibaMX-1-AI 380VAC	V	-1074	1074	10000	☑	-0,032776
11	ibaMX-1-AI 380VAC	V	-1074	1074	10000	☑	-0,032776
12	ibaMX-1-AI +/-10V	V	-10	10	10000	☑	-0,00030518
13	ibaMX-1-AI +/-10V	V	-10	10	10000	☑	-0,00030518
14	ibaMX-1-AI +/-10V	V	-10	10	10000	☑	-0,00030518
15	ibaMX-1-AI +/-10V	V	-10	10	10000	☑	-0,00030518

ibaPDA has a new function: FobMLinkStatus(‘BoardNr’, ‘LinkNr’). This function returns the status of the FOB link in FOB-M mode:

- 0 = link not active
- 1 = link ok
- 2 = link broken

This function can be used to write the status of the link to the dat file or to generate alarms.

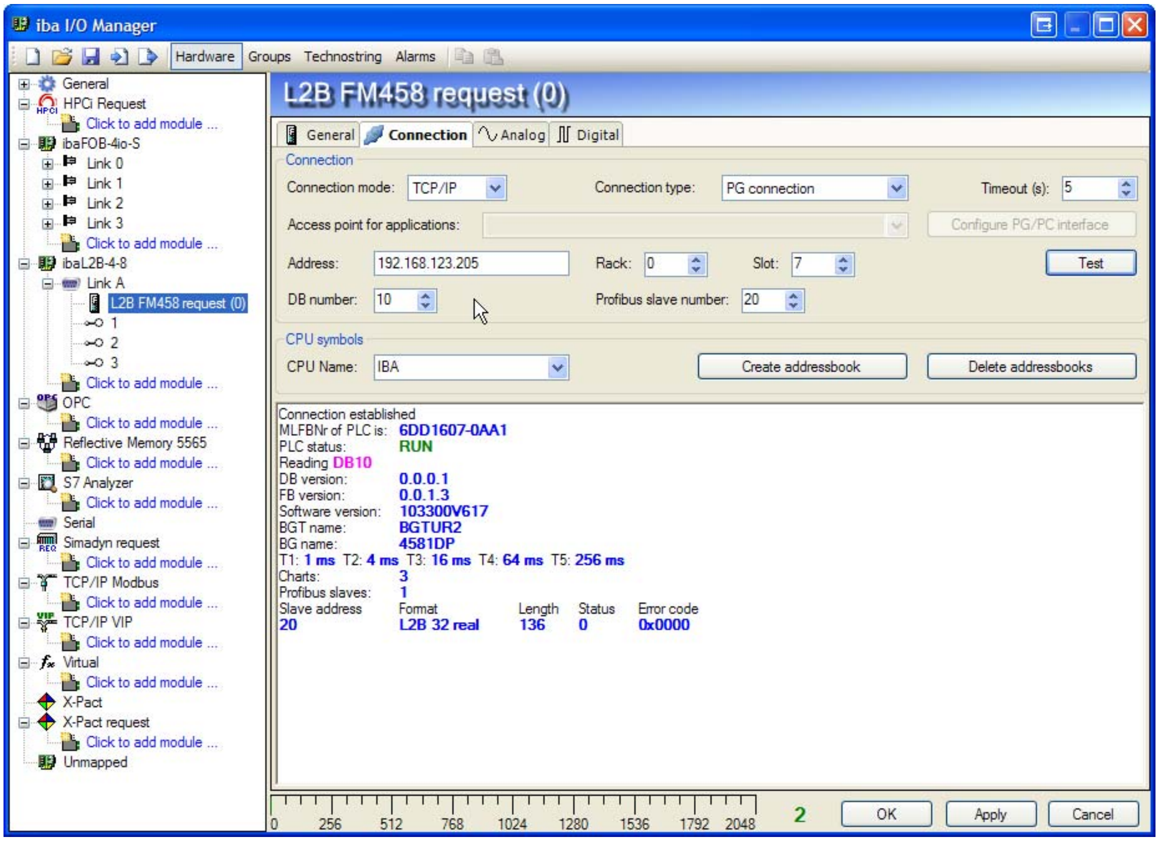
4 FM458 request

The FM458 request system uses 2 types of connections between PC and FM458:

- Control connection via TCP/IP or via a PC/CP connection (MPI, profibus, ISO or TCP/IP)
- Data connection via profibus

The control connection is used to create the addressbook and to request signals. The data connection is used to cyclically transfer the data for the requested signals. Currently only the L2B board is supported for the data connection. In the future ibaPDA will also support the ibaBM-DPM-S.

You can add a L2B FM458 request module to an L2B board. On the connection tab you can setup the control and data connection to the FM458.



Connection

Connection mode: TCP/IP   Connection type: PG connection   Timeout (s): 5

Access point for applications:

Address: 192.168.123.205   Rack: 0   Slot: 7

DB number: 10   Profibus slave number: 20

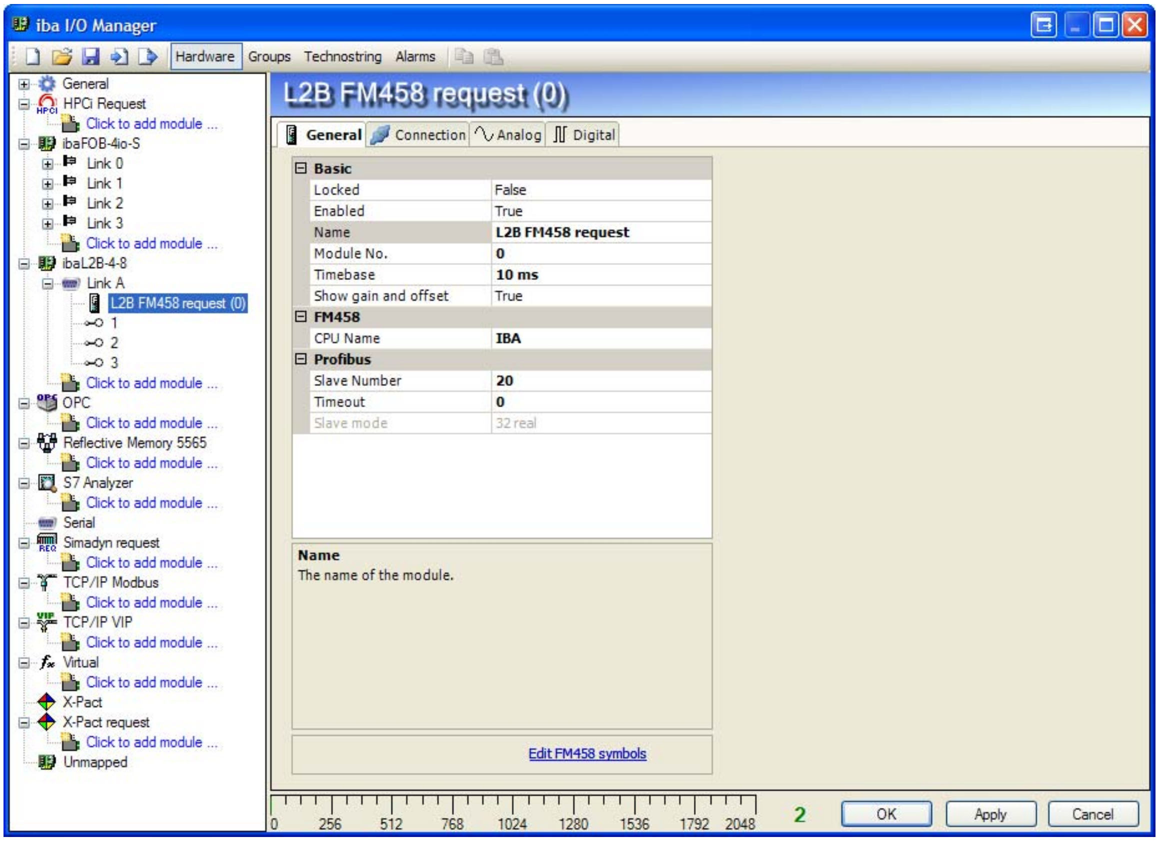
CPU symbols

CPU Name: IBA

```
Connection established
MLFBNr of PLC is:   6DD1607-0AA1
PLC status:         RUN
Reading DB10
DB version:         0.0.0.1
FB version:         0.0.1.3
Software version:   103300V617
BGT name:           BGTUR2
BG name:            4581DP
T1: 1 ms  T2: 4 ms  T3: 16 ms  T4: 64 ms  T5: 256 ms
Charts:             3
Profibus slaves:    1
Slave address       Format          Length   Status   Error code
20                  L2B 32 real     136      0        0x0000
```

First you can choose between TCP/IP and PC/CP. If you choose PC/CP then you will have to select the access point. You can then fill in address, rack and slot number. You will also have to fill in the DB number that will be used for the communication. Now you can click the Test button to test the connection. If an FM458 is connected then ibaPDA will check that the DB is valid and it will also display some diagnostic information. The diagnostic information contains the profibus slave number. This slave number is automatically copied to the connection properties. See the S7 analyzer documentation for more information about the connection properties.

If the communication with the FM458 is OK then you can create the addressbook. You first have to give the FM458 a symbolic name. The CPU name combobox contains a list of the already available FM458s. You can also type in a new name. Click the create addressbook button to create the addressbook.



Basic

Locked	False
Enabled	True
Name	L2B FM458 request
Module No.	0
Timebase	10 ms
Show gain and offset	True

FM458

CPU Name	IBA

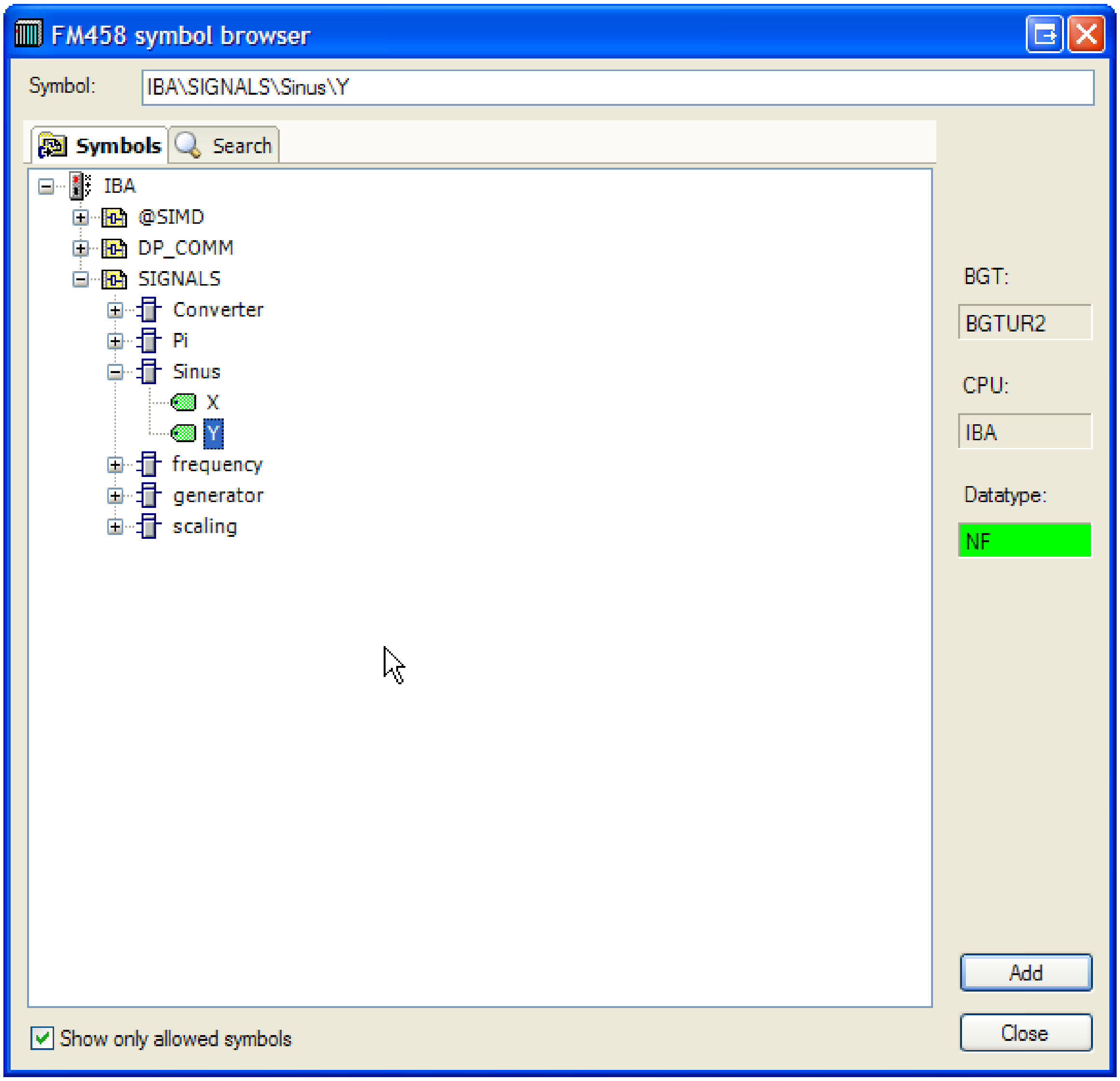
Profibus

Slave Number	20
Timeout	0
Slave mode	32 real

Name
The name of the module.

Edit FM458 symbols

On the general tab you can click the “Edit FM458 symbols”. This will open the FM458 symbol browser where you select the signals you want to measure.



Symbol: IBA\SIGNALS\Sinus\Y

BGT: BGTUR2

CPU: IBA

Datatype: NF

☑ Show only allowed symbols

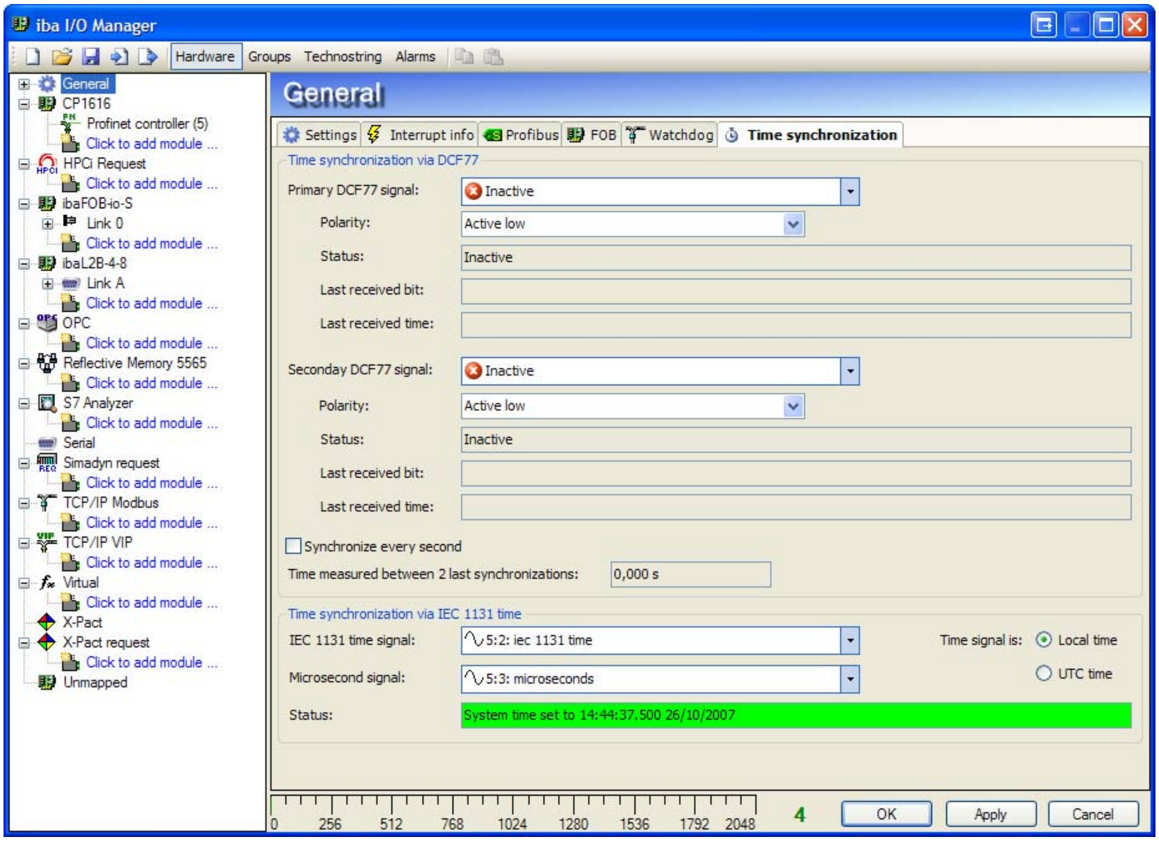
The symbol browser looks and acts the same as the TDC symbol browser.

When you have made your selection of signals then you can start the measurement. During validation you will first see the status of the profibus slave and then you will see the request communication between pda and the FM458.

The signals are transferred from the FM458 to ibaPDA in their native datatypes. There is no conversion to float done by the agent.

5 IEC 1131 time synchronization

On the general node there is a new time synchronization tab. There are 2 types of time synchronization supported: DCF77 and IEC 1131 time. Both types can use any signal for the time information. It doesn’t matter via which interface the signals are being measured.



Time synchronization via DCF77

Primary DCF77 signal: Inactive
Polarity: Active low
Status: Inactive
Last received bit:
Last received time:

Seconday DCF77 signal: Inactive
Polarity: Active low
Status: Inactive
Last received bit:
Last received time:

☐ Synchronize every second
Time measured between 2 last synchronizations: 0,000 s

Time synchronization via IEC 1131 time

IEC 1131 time signal: 5:2: iec 1131 time
Microsecond signal: 5:3: microseconds
Time signal is: ● Local time ○ UTC time
Status: System time set to 14:44:37.500 26/10/2007

Pda needs 2 time signals in IEC 1131 mode:

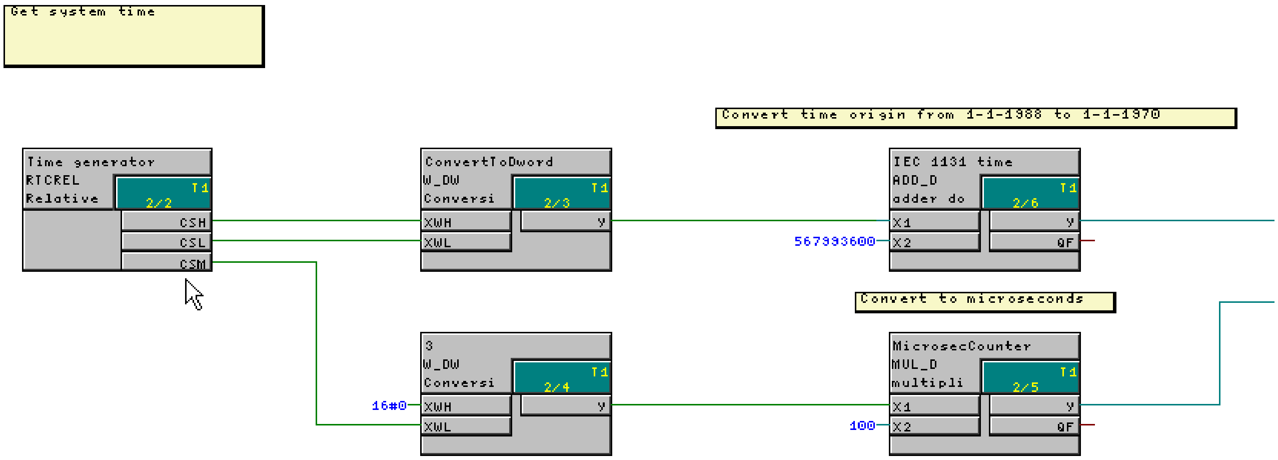
- A DWORD with the IEC 1131 time. This time represents the number of seconds that have passed since 00:00 on January 1st 1970.
- A DWORD with a microsecond counter that goes from 0 to 999999. The counter is reset at the start of every second.

You also have to specify if the received time is a local time or a UTC time. If it is a local time then pda will take that time without any conversions. If it is a UTC time then pda will convert it to a local time on the PC according to the time zone settings of the PC.

The status field shows the current status of the time synchronization. The time synchronization only works when the acquisition is running.

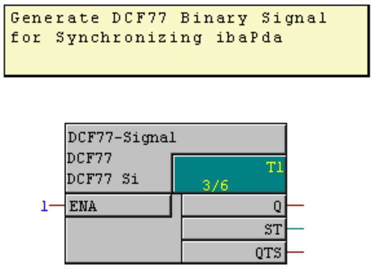
You should use the “Use system time” option on the datastore when time synchronization is active. Otherwise the dat file start time will not be accurate.

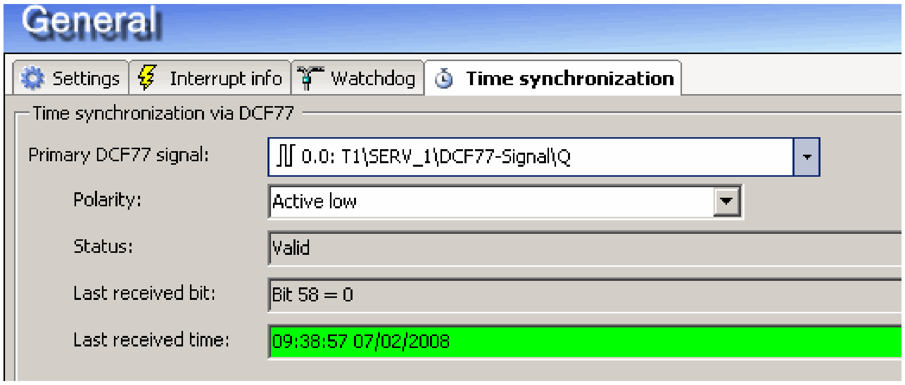
Example for FM458:



Example for Simadyn (CFC) and Simatic TDC:

Use the DCF77 function block to generate a binary decoded time signal. The functionblock can be delivered by iba AG.





Time synchronization via DCF77

Primary DCF77 signal: 0.0: T1\SERV_1\DCF77-Signal\Q

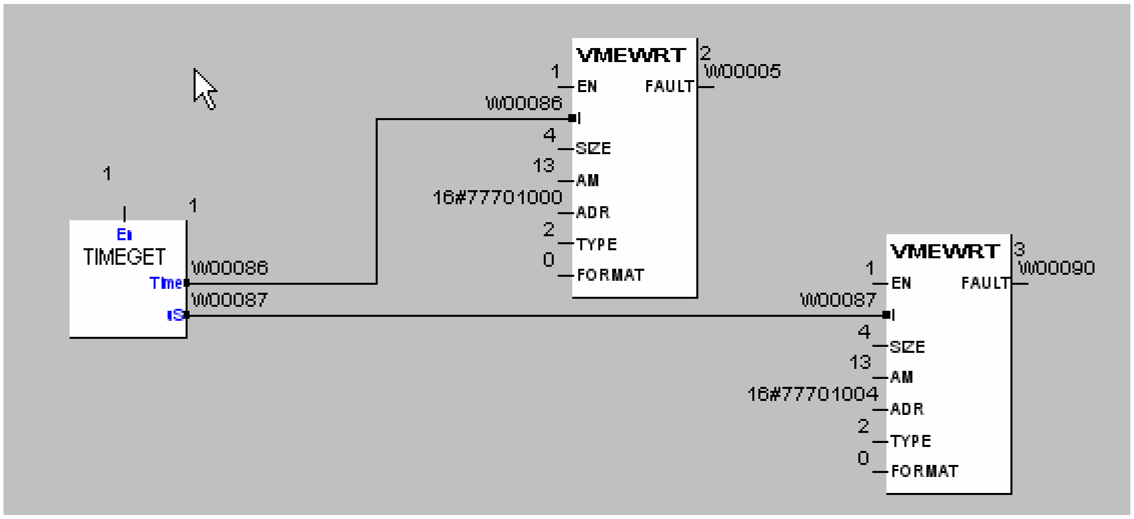
Polarity: Active low

Status: Valid

Last received bit: Bit 58 = 0

Last received time: 09:38:57 07/02/2008

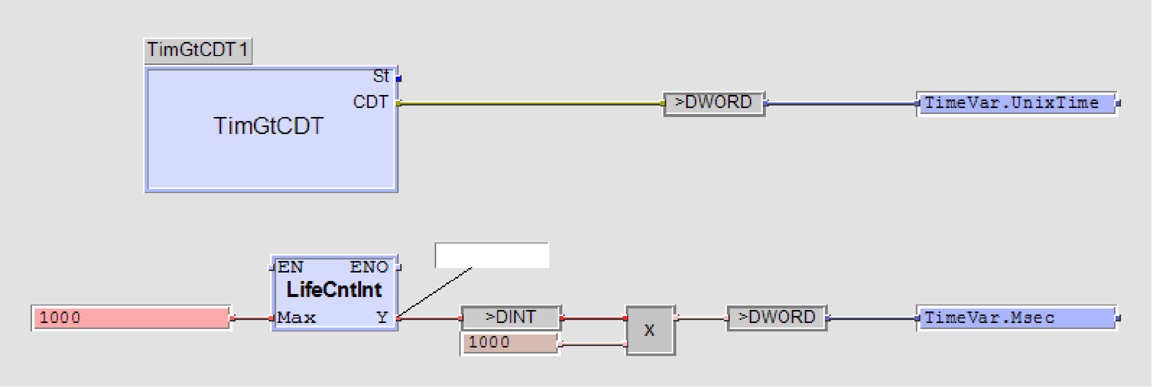
Example for HPCi:



Example for Simotion:

```
PROGRAM pda_timesync

 VAR
 systemtime: DT;
 myRetTime : TOD;
 myRetDate : DATE;
 myRtc :rtc;

 timeAsDWORD: DWORD;
 timeInSecSince1992: DWORD;
 secondsToday : DWORD;

 timeArray: ARRAY[0..3] OF BYTE;
 dateArray: ARRAY[0..3] OF BYTE;
 END_VAR

 //Get current system time
 myRtc(set:=0,read:=1,pdt:=DT#0001-01-01-0:0:0);
 systemtime := myRtc.cdt;

 //Extract time of day from system time and convert it to a DWORD
 //(milliseconds since midnight)
 myRetTime:= DT_TO_TOD( systemtime);
 timeArray:= ANYTYPE_TO_LITTLEBYTEARRAY(myRetTime,0);
 timeAsDWORD := LITTLEBYTEARRAY_TO_ANYTYPE(timeArray,0);

 //Extract date from system time and convert it to a DWORD
 //(seconds since 1 jan 1992)
 myRetDate:= DT_TO_DATE(systemtime);
 dateArray:= ANYTYPE_TO_LITTLEBYTEARRAY(myRetDate,0);
 timeInSecSince1992 := LITTLEBYTEARRAY_TO_ANYTYPE(dateArray,0);

 //Create IEC 1131 datetime: seconds since 1 jan 1970
 secondsToday := timeAsDWORD /1000 ; // get seconds from today
 pn_pda_unixtime := timeInSecSince1992 * 24*3600 + 694224000 + secondsToday;

 //Create microsecond counter
 pn_pda_usec := (timeAsDWORD MOD 1000) * 1000 ; // from msec to usec

 END_PROGRAM
```

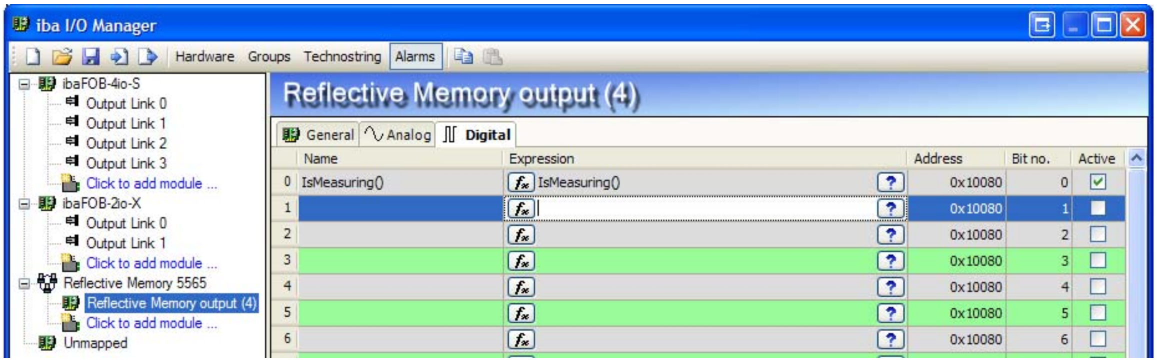
Example for X-Pact:



Use the TimeGtCDT block to get the system time and the ANY_TO_DWORD2 block to convert it into IEC 1131 format. The DWORD is then written on the reflective memory board. The LifeCntInt block can be used to generate a microsecond counter. The counter is also written on the reflective memory board.

6 Reflective memory output module

Pda can write analog and digital values on the reflective memory via a reflective memory output module.



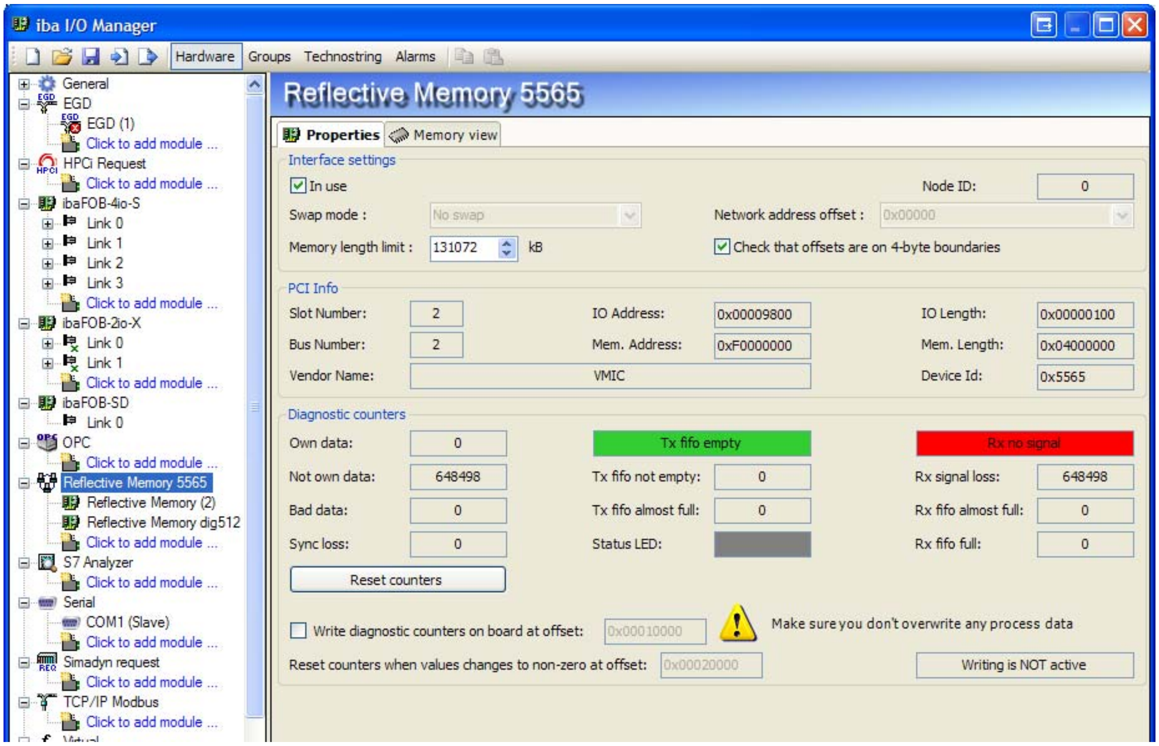
Reflective Memory output (4)

	Name	Expression	Address	Bit no.	Active
0	IsMeasuring()	IsMeasuring()	0x10080	0	☑
1			0x10080	1	□
2			0x10080	2	□
3			0x10080	3	□
4			0x10080	4	□
5			0x10080	5	□
6			0x10080	6	□

The values that are written to the reflective memory board are the results of expressions. Like with all output modules the data is written at a maximum rate of 50ms.

7 Diagnostic counters on VMIC-5565 board

Pda monitors the status registers on the VMIC-5565 board. Pda calculates some counters based on some bits in the status registers. The calculations follow the specification received from Alcoa.



Reflective Memory 5565

Interface settings

☑ In use

Node ID: 0

Swap mode : No swap

Network address offset : 0x00000

Memory length limit : 131072 kB

☑ Check that offsets are on 4-byte boundaries

PCI Info

Slot Number: 2 | IO Address: 0x00009800 | IO Length: 0x00000100

Bus Number: 2 | Mem. Address: 0xF0000000 | Mem. Length: 0x04000000

Vendor Name: VMIC | Device Id: 0x5565

Diagnostic counters

Own data:	0	Tx fifo empty		Rx no signal	
Not own data:	648498	Tx fifo not empty:	0	Rx signal loss:	648498
Bad data:	0	Tx fifo almost full:	0	Rx fifo almost full:	0
Sync loss:	0	Status LED:		Rx fifo full:	0

Reset counters

□ Write diagnostic counters on board at offset: 0x00010000

Make sure you don't overwrite any process data

Reset counters when values changes to non-zero at offset: 0x00020000

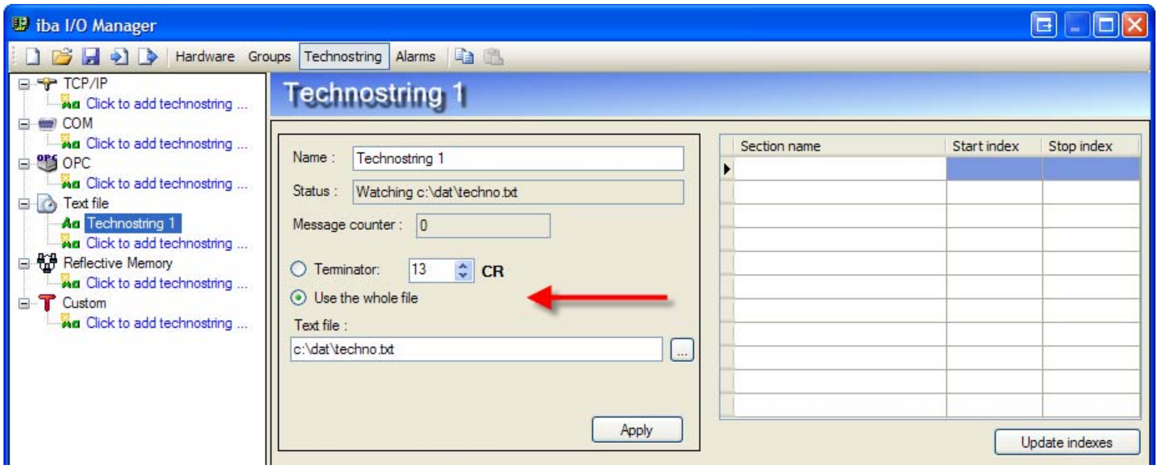
Writing is NOT active

The counters are always displayed on the reflective memory interface. Pda can also write them back to the reflective memory board. Pda only writes them when the acquisition is running. When pda writes to the reflective memory then it also monitors a DWORD at the reset offset. When the DWORD changes to a non-zero value then the counters are reset.

8 EGD

See separate EGD description: sw_COMM_ibaPDA-EGD_v1.0_en_A4.pdf

9 Text file technostring


iba I/O Manager
Hardware
Groups
Technostring
Alarms
TCP/IP
Click to add technostring ...
COM
Click to add technostring ...
OPC
Click to add technostring ...
Text file
Technostring 1
Click to add technostring ...
Reflective Memory
Click to add technostring ...
Custom
Click to add technostring ...
Technostring 1
Apply
Update indexes

Name : Technostring 1

Status : Watching c:\dat\techno.txt

Message counter : 0

○ Terminator: 13 CR

● Use the whole file

Text file : c:\dat\techno.txt

Section name	Start index	Stop index

The text file technostring now has 2 ways to end the string. It can use a terminator or it can just use the whole file.