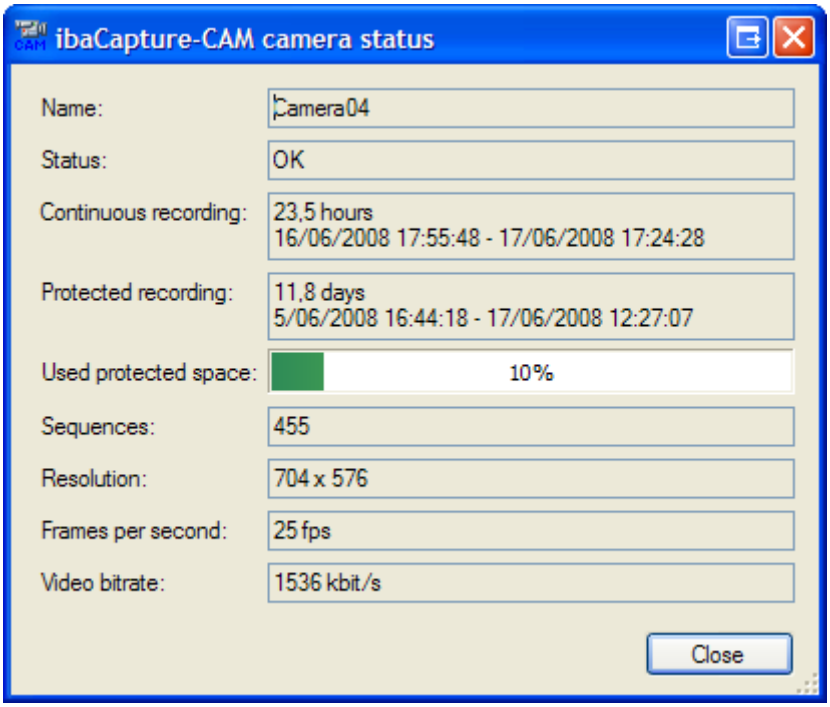
New features in ibaPDA v6.17.0

1 ibaCapture-CAM

ibaCapture-CAM is a new interface in ibaPDA to record video in sync with the data recording. ibaPDA connects to a video server that does the actual video recording. The video server sends synchronization information to ibaPDA and ibaPDA writes this information in the dat file. Later on ibaAnalyzer uses that information to display the video images that correspond to a specific timestamp.

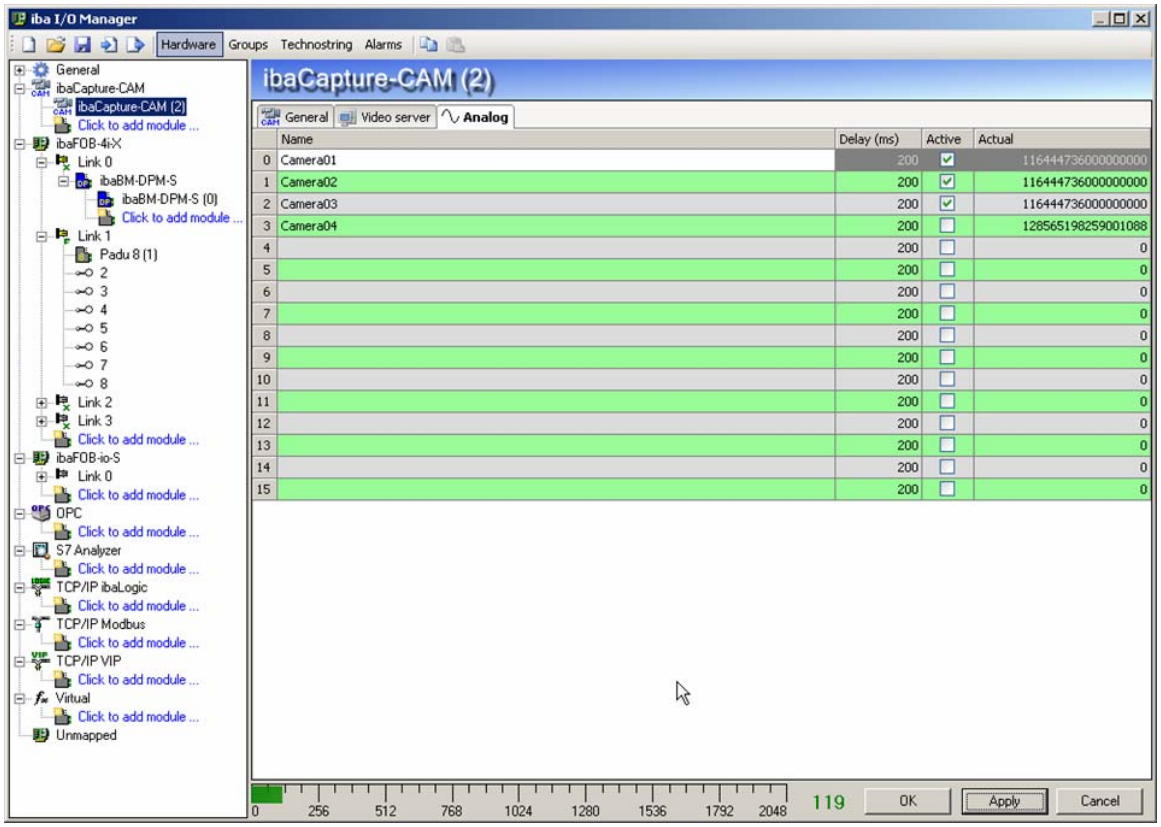
In ibaPDA you have to add an ibaCapture-CAM module. On the video server tab you can specify the IP address or hostname of the video server and the port number that is configured on the video server. There is a search button that searches the network for available video servers. Click the connect button to connect to the video server. If the ibaCapture-CAM player is installed then you will see the live video of the selected camera(s) on the I/O manager. You can display up to 16 cameras by using the 1x1, 2x2, 3x3 and 4x4 toolbar buttons.

When you doubleclick on a camera then a status dialog is shown that provides information about the selected camera.

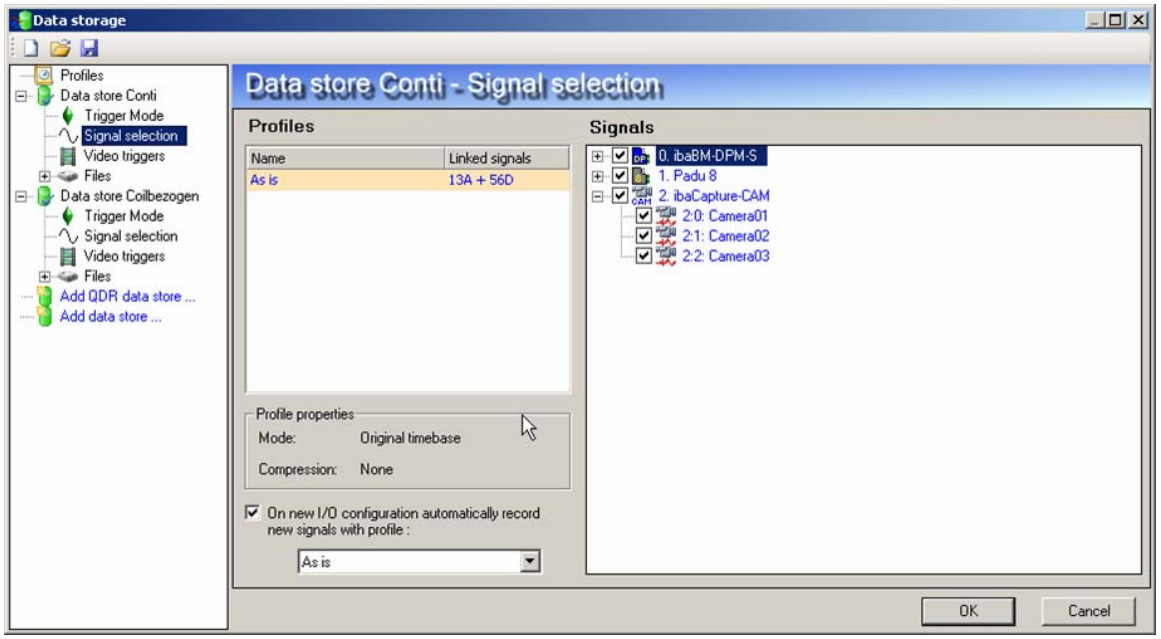


The video server records video continuously in a pre-allocated ringbuffer on its harddisk. The continuous recording item shows how much time is stored in that ringbuffer. ibaPDA can mark certain video sequences as protected by using video triggers. The video triggers are defined in the datastore configuration. The protected video sequences are not overwritten by the continuous recording. The protected recording item shows how much time is marked as protected. The progress bar shows how much of the reserved space for protected sequences is in use. The number of protected sequences is also displayed. If the protected space is full then the oldest protected sequences are automatically unprotected and they are returned to the continuous recording ringbuffer.

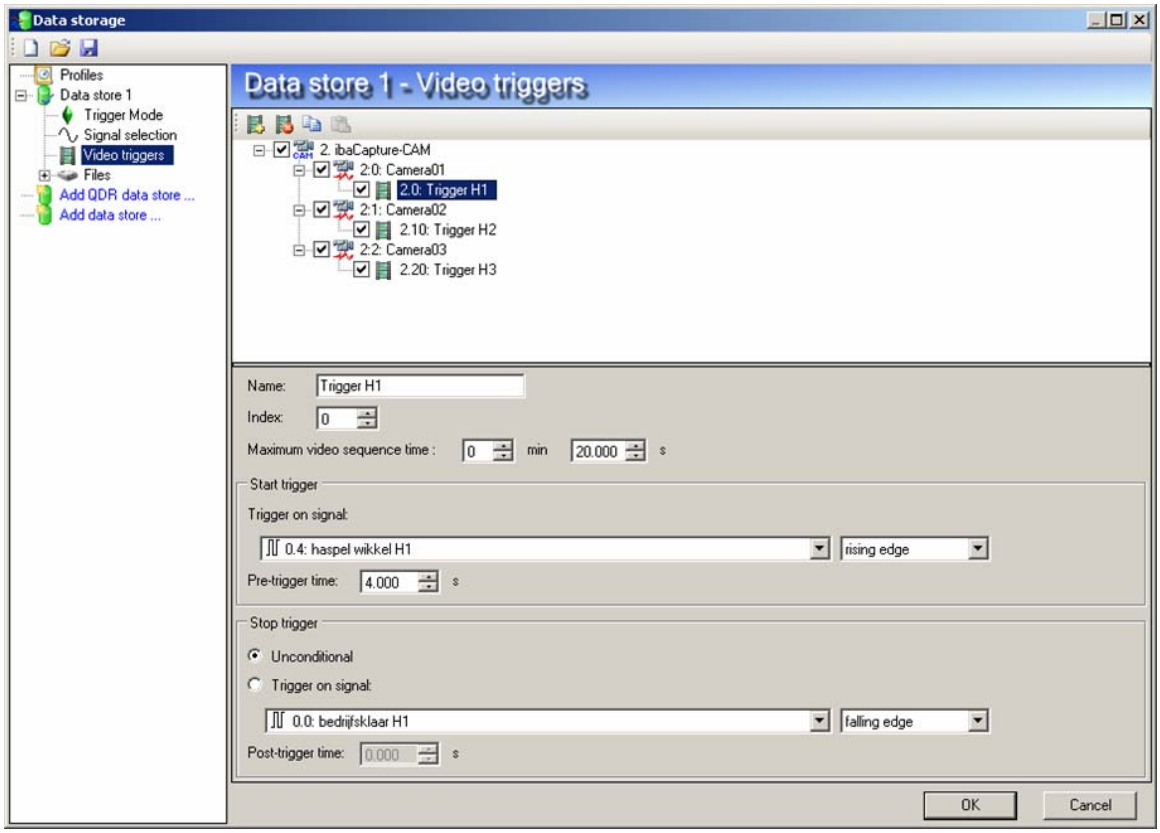
On the analog tab you can configure the cameras that are connected to the video server. Each analog signal corresponds to 1 camera. One video server can support maximum 16 cameras. You can give each camera a name. There is always a delay between the actual video recording and the time that the synchronization information is received by ibaPDA. This delay is caused by the MPEG4 video compression that is done on the video server and by network latency. You can specify this delay per camera/signal. When the measurement is running then the actual column shows the received synchronization information per camera.



In the signals node of the datastore you can see the camera synchronization signals. When you select a camera signal then the synchronization info will be saved in the dat file. ibaAnalyzer will show the camera in its signal tree. When you doubleclick the camera in ibaAnalyzer then ibaAnalyzer will retrieve the video from the continuous recording ringbuffer of the video server. If the video is already overwritten then ibaAnalyzer will give an error.



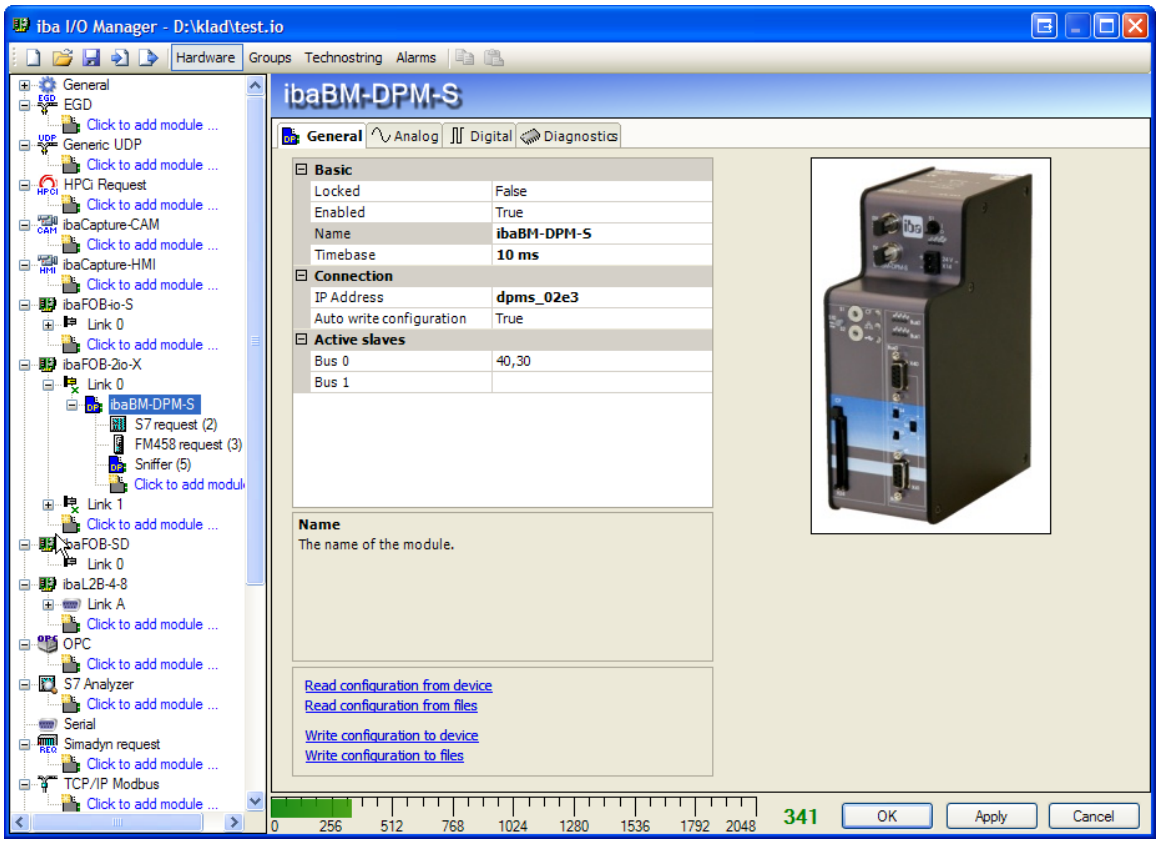
If you want to save specific video sequences for a longer period of time then you have to use video triggers. Select the video triggers node in the tree.



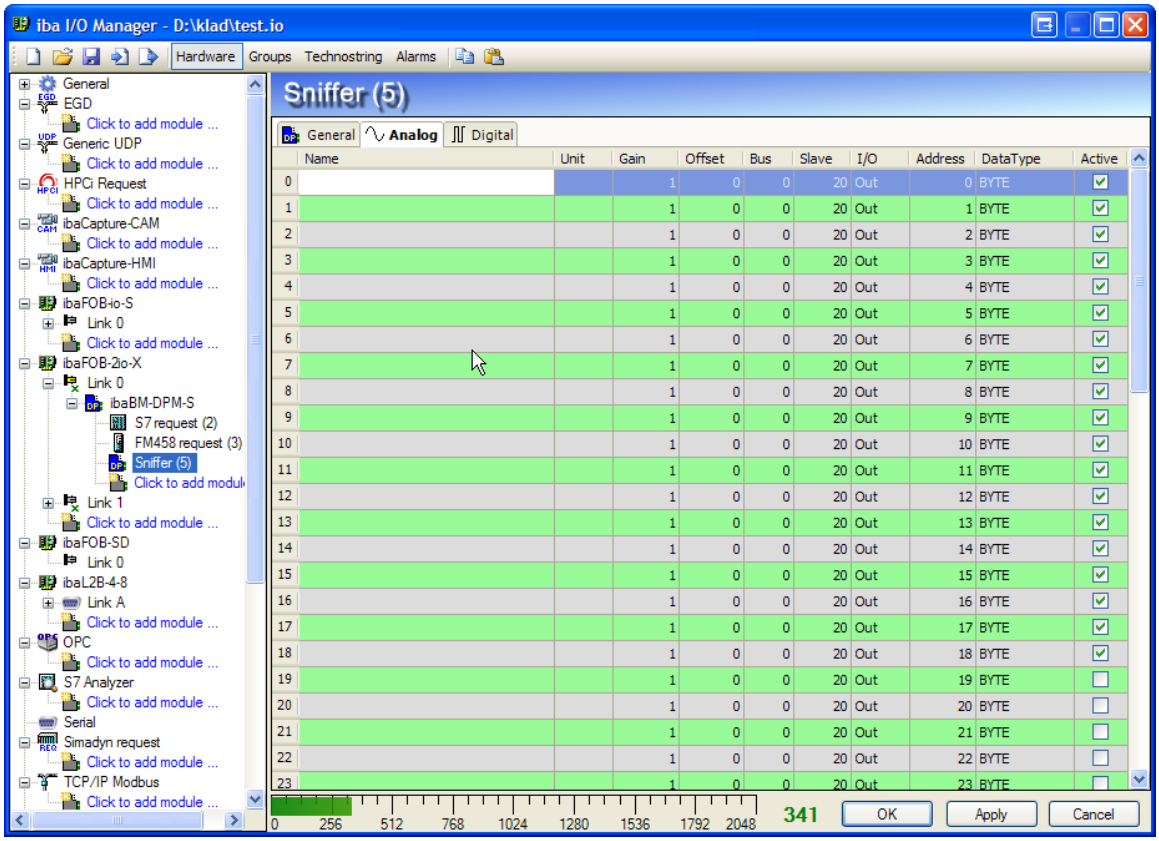
The tree shows all video servers and the active cameras connected to them. You can add a video trigger to a camera by selecting the camera and clicking the add button on the toolbar. You can also use the context menu to add one. A camera can have up to 10 video triggers. You have to define a start trigger and optionally a pretrigger time. The video sequence stops when it reaches the maximum video sequence time you specified or when a stop trigger occurs. The stop trigger is optional.

2 ibaBM-DPM-S

The profibus sniffer module has been redesigned. You now have to first add an ibaBM-DPM-S device. You can then add different submodules. There are currently 3 kinds of submodules: sniffer, FM458 request and S7 request.

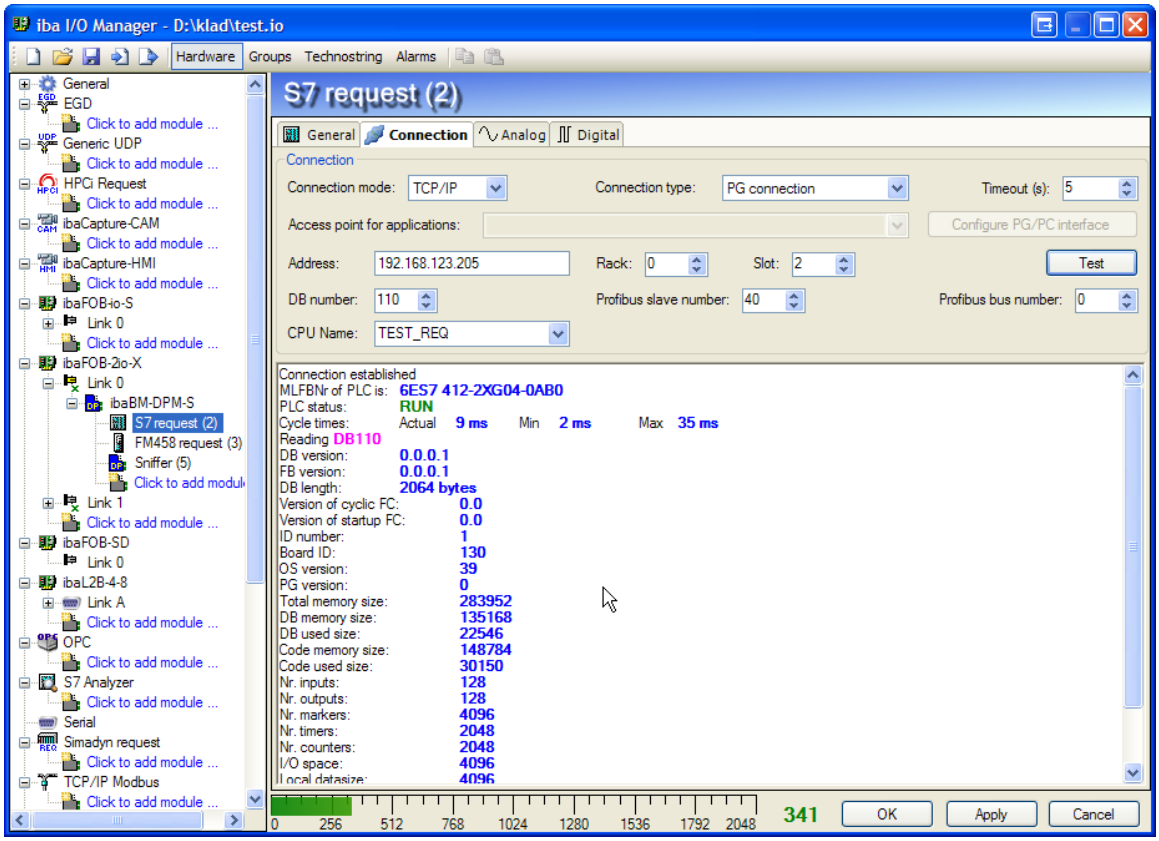


2.1 Sniffer submodule



The sniffer submodule is in principle the same as the old ibaBM-DPM-S module. You can specify a number of active slaves. You can specify the number of analog and digital signals. You have to set the bus number, slave number, direction, address, datatype and bit number for the analog and digital signals.

2.2 S7 request submodule

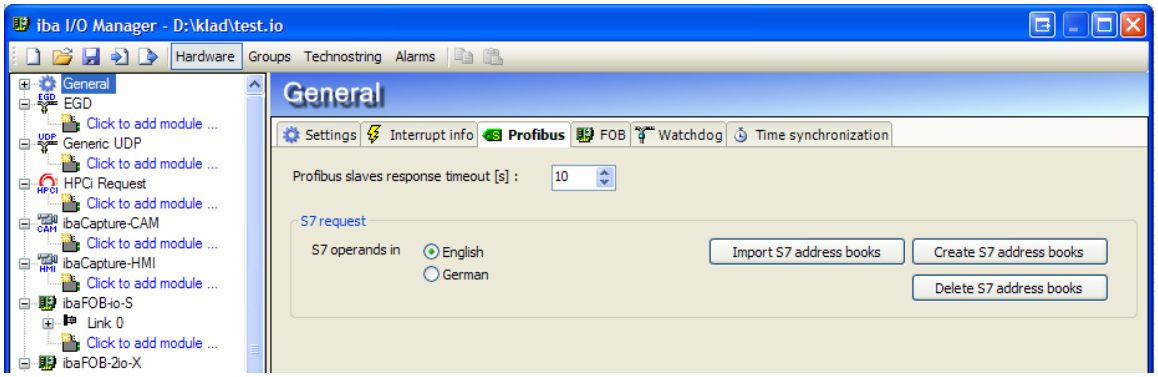


The S7 request module can measure all operands from an S7. It requires 2 connections to the S7. A control connection to send the list of operands and a data connection that is used by the S7 to send the values of the operands to ibaPDA.

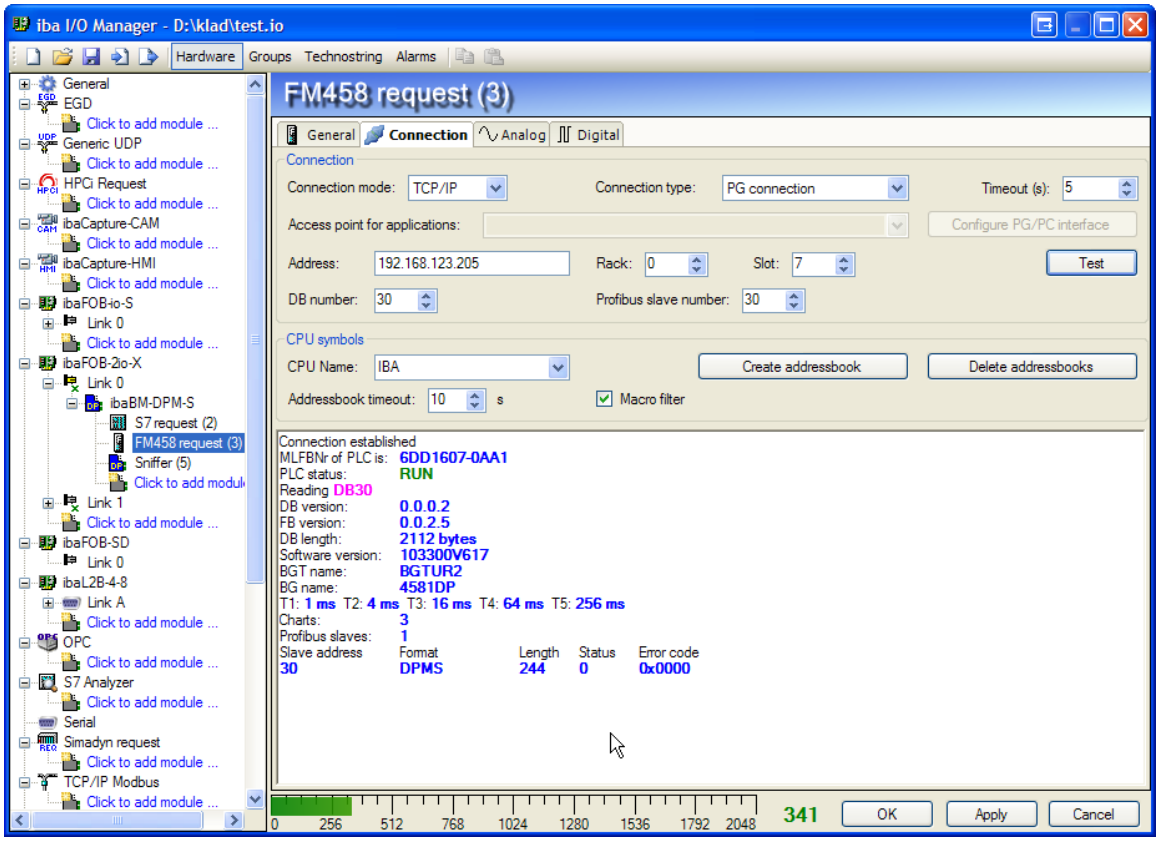
The control connection uses TCP/IP or PC/CP similar to the S7 analyzer interface. ibaPDA uses the connection to read and write a DB in the S7. This DB is used to exchange messages between ibaPDA and the request agent on the S7. On the connection tab you have to configure the connection and the DB number. When you click the Test button ibaPDA tries to connect to the S7 and read some diagnostic information.

The data connection is done via profibus. You have to specify the slave number of profibus slave. You also need to specify to which bus of the DPMS the S7 is connected. The diagnostic information read from the S7 contains the profibus slave that is configured in the S7. This value is automatically copied to the connection properties. The size of the profibus slave is also reported in the diagnostic information. The size can be decided by the user and it is maximum 244 bytes. 8 bytes are reserved for digital signals. The rest of the bytes can be used to transfer analog signals. The analog signals are transferred in their native datatype. There is no conversion to float.

If you want to create addressbooks then select the general node in the tree and go to the profibus tab. There you can manage the S7 addressbooks.



2.3 FM458 request submodule

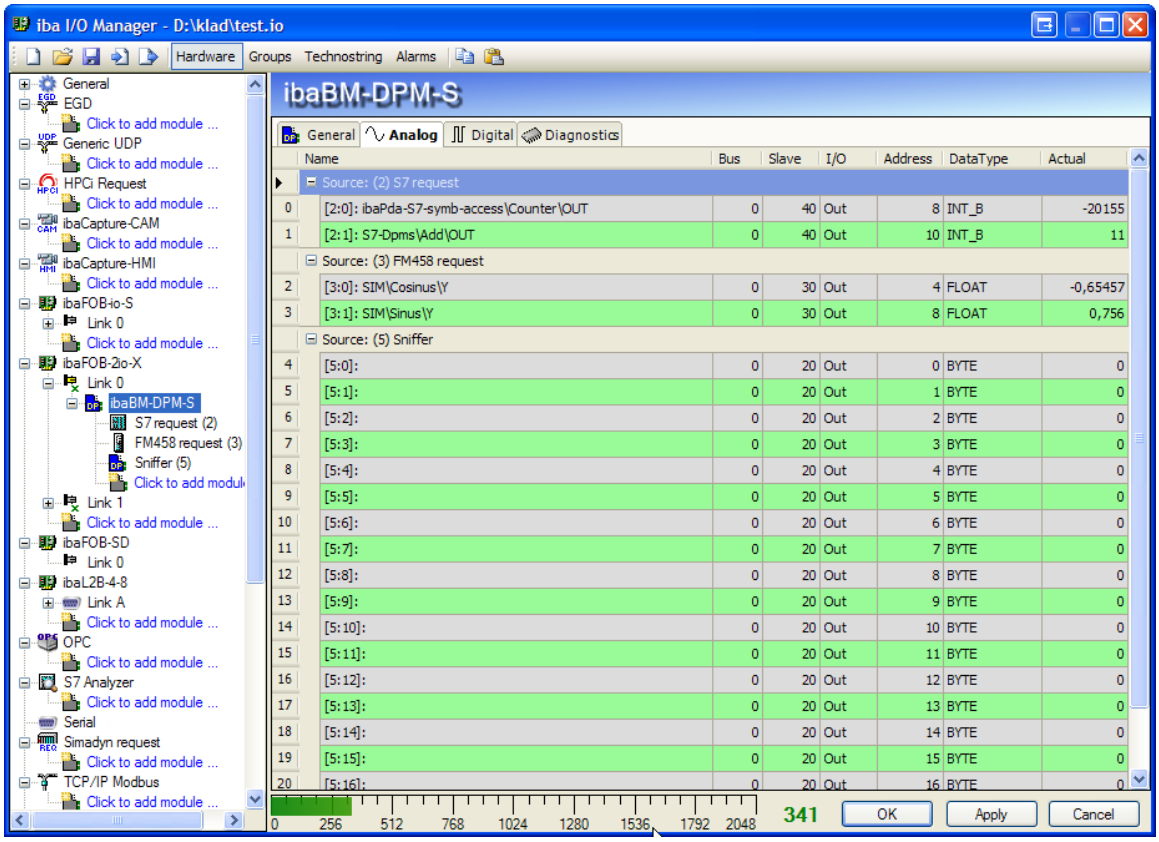


The FM458 request module is almost identical to the FM458 request module on L2B. The only difference is that the DPM-S module uses an active profibus slave on the DPM-S for the data transfer and the L2B module uses a slave on the L2B board. The DPM-S variant can use up to 244 bytes for transferring data. The L2B module is limited to 128 bytes for analog signals and 4 bytes for digital signals.

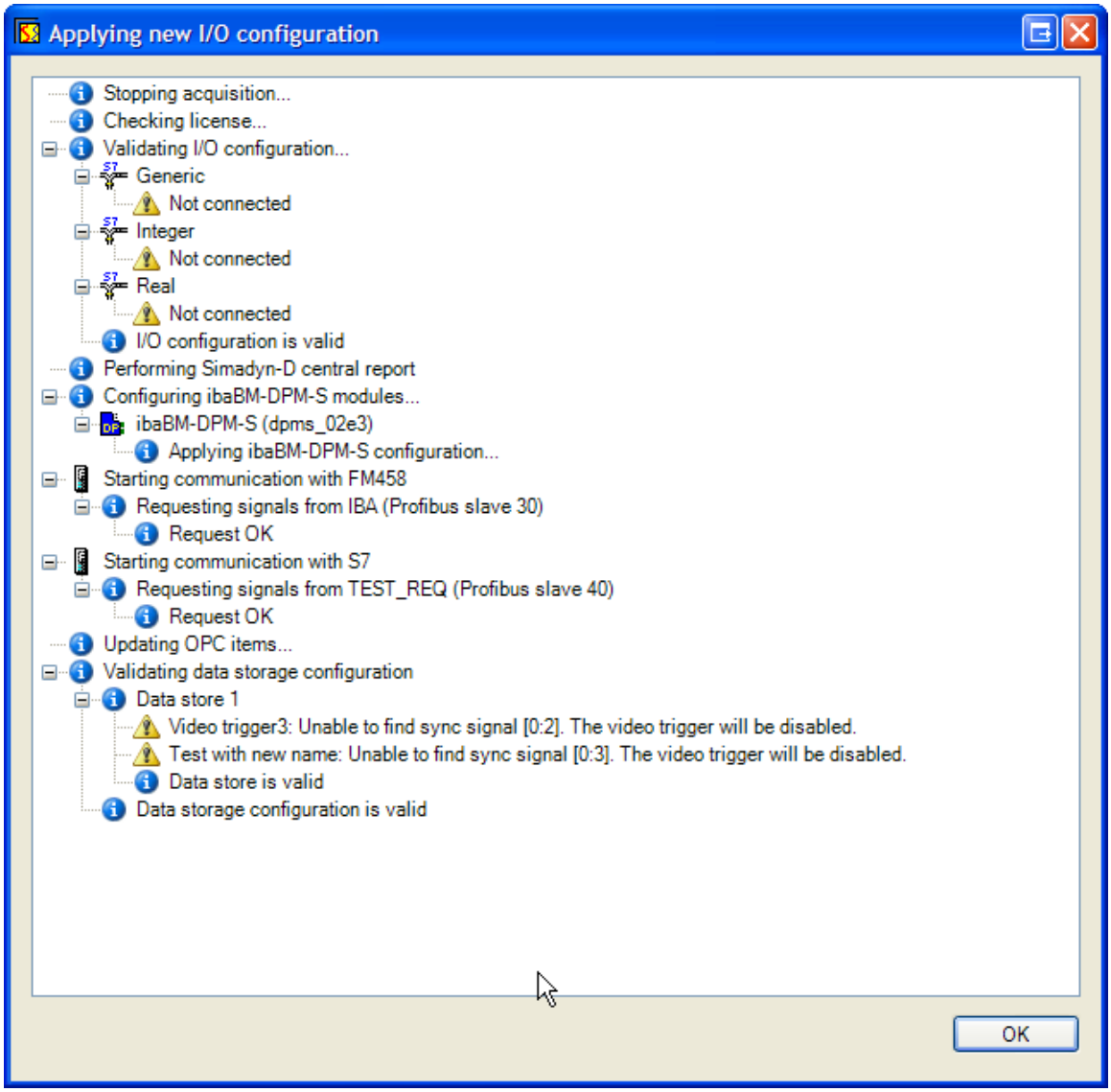
2.4 DPM-S configuration

The configuration of the DPM-S is automatically created by ibaPDA out of the submodules. The actual configuration can be seen on the ibaBM-DPM-S device. On the general tab you can see the

active slaves that are enabled. On the analog tab you see which analog signals are requested. They are grouped by their submodules. You also see the actual value of the signals. The digital tab is similar to the analog tab.

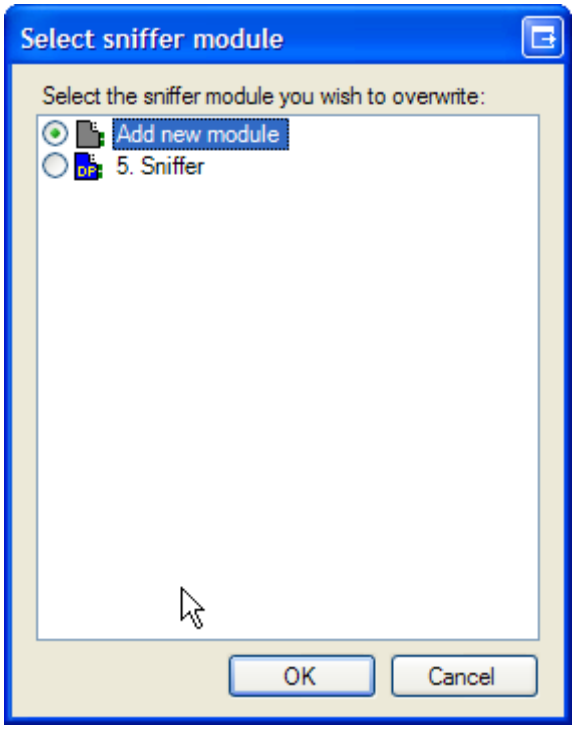


This configuration is automatically generated at the start of the measurement. If there are request modules then this configuration is always transferred to the DPM-S at the start of the measurement. If there are only sniffer modules then you can decide if you want to transfer the configuration at the start of the measurement or not by setting the auto write configuration property on the ibaBM-DPM-S device. The following screenshot shows the transfer of the DPM-S configuration and afterwards the request process on FM458 and S7.



You can also manually generate the configuration by clicking the write configuration hyperlinks on the ibaBM-DPM-S device.

You can read the configuration from a device by clicking the read configuration hyperlinks on the ibaBM-DPM-S device. ibaPDA will read the configuration and create a sniffer module that corresponds to the configuration. ibaPDA can’t generate the request modules from the configuration. If there are already sniffer modules defined then ibaPDA asks if you want to overwrite one of the sniffer modules or if you want to create a new one.


Select sniffer module
Select the sniffer module you wish to overwrite:
Add new module
5. Sniffer
OK
Cancel

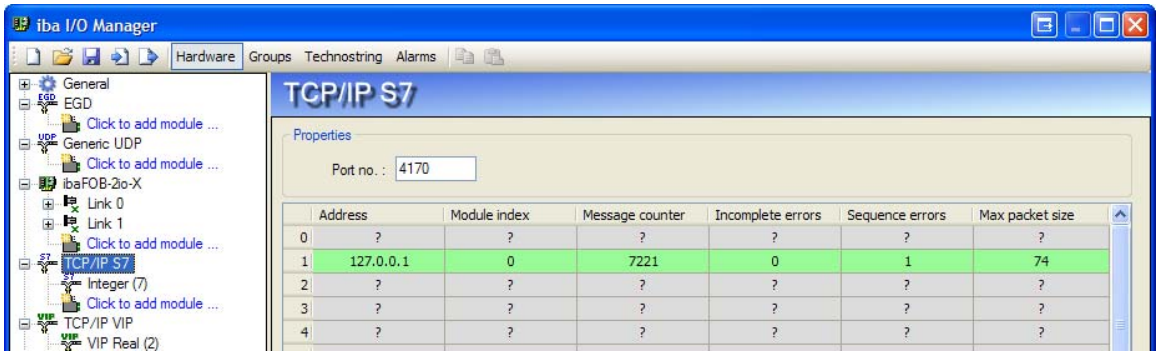
3 S7 TCP/IP interface

The S7 TCP/IP interface is very similar to the VIP interface. ibaPDA listens to a TCP port number. The port number is 4170 by default but it can be changed by the user. The TCP packets that arrive on the port must have the following structure.

Offset	Size	Datatype	Description
0	2	INT	Message length: The total length of the message including this 6 byte header.
2	2	INT	Module index: Number between 0..63 or 100..163. This determines the address in ibaPDA where the message is saved
4	2	WORD	Sequence counter: Number that increments after every send.
6	1 … 512	Any	Data: The actual data that ibaPDA needs to measure. The size and structure of the data can be chosen freely. The size is limited to 512 bytes.

The header (message length, module index and sequence counter) has to be in network byte order which is big endian byte order! If you are sending from an S7 CPU then this is no problem because the S7 uses big endian byte order internally.

If you have the S7 TCP/IP license then the S7 TCP/IP interface will appear in the I/O manager.



On the interface you can set the port number. You can also look at the connection diagnostics similar to all other TCP/IP interfaces.

There are 3 module types supported on the S7 TCP/IP interface:

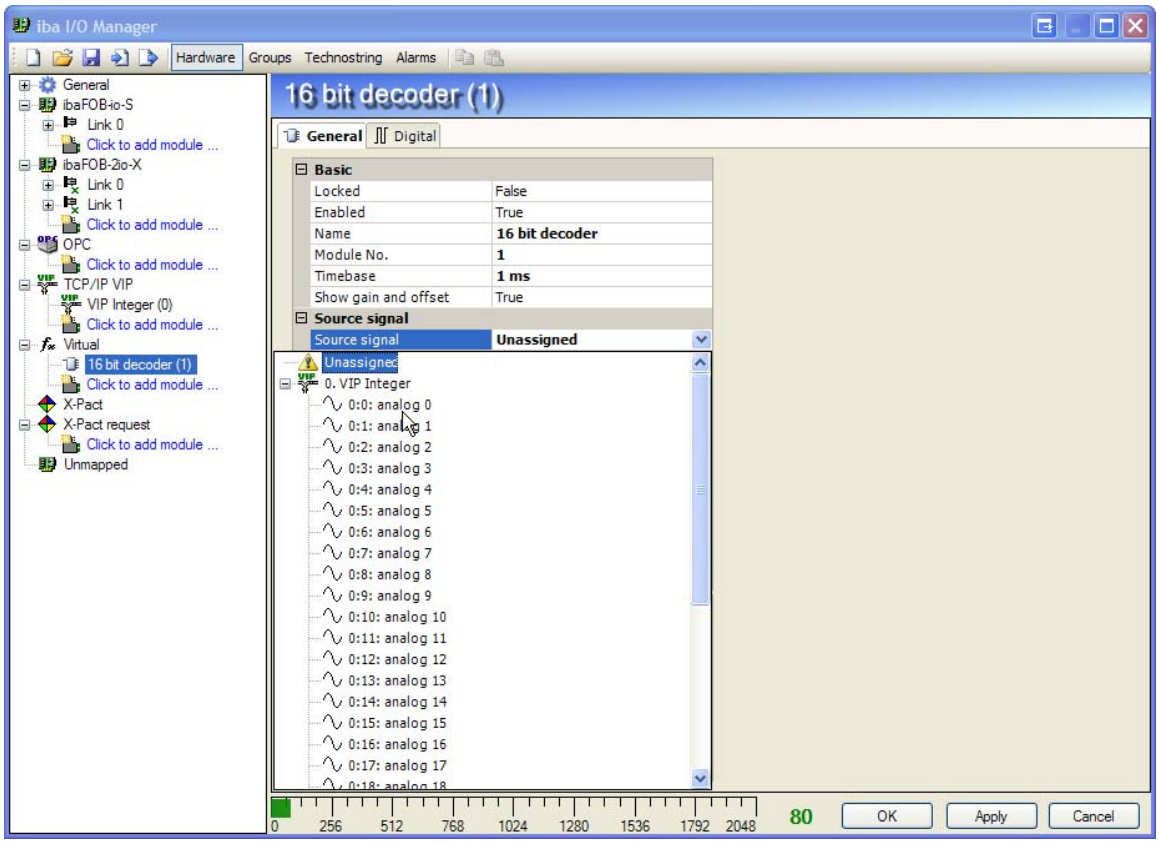
- Integer module: 32 integers + 32 digitals (same message layout as VIP integer)
- Real module: 32 reals + 32 digitals (same message layout as VIP real)
- Generic module: user can configure the number of analog and digital signal and their address and datatype.

4 Generic UDP interface

See separate document sw_COMM_ibaPDA-Generic_UDP_v1.0_en_A4.doc

5 16 bit decoder module

The 16 bit decoder module can be used to create 16 digital signals from 1 analog signal. The analog signal is converted to a 16 bit integer and each digital signal corresponds to one of the bits. The analog signal can come from any other module.



On the general tab of the decoder module you have to select the source signal. The timebase of the decoder module will always be the same as the timebase of the source signal. On the digital tab you can name the 16 digital signals.

6 New functions

There are 2 new functions that can be used in expressions: Truncate and SampleAndHold.

6.1 Truncate('expr')

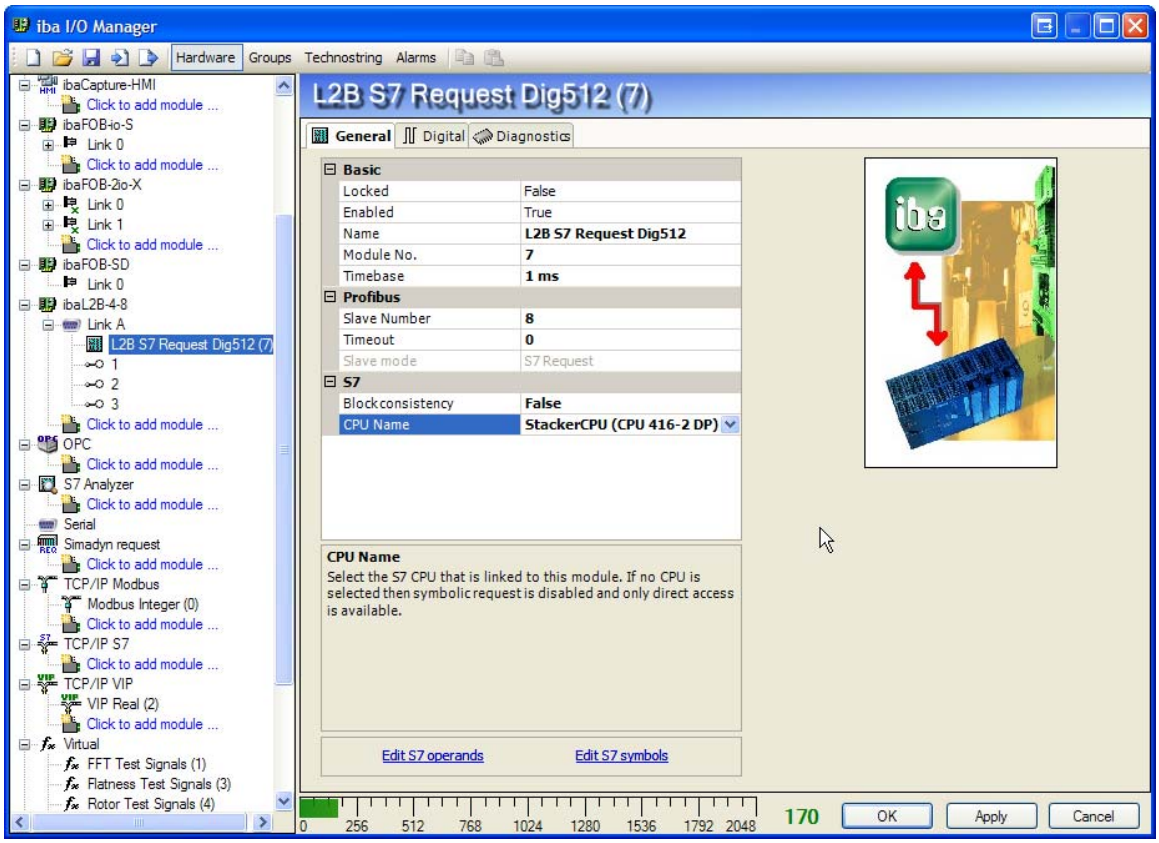
This function returns the integral part of a floating-point value.

6.2 SampleAndHold('expr','sample')

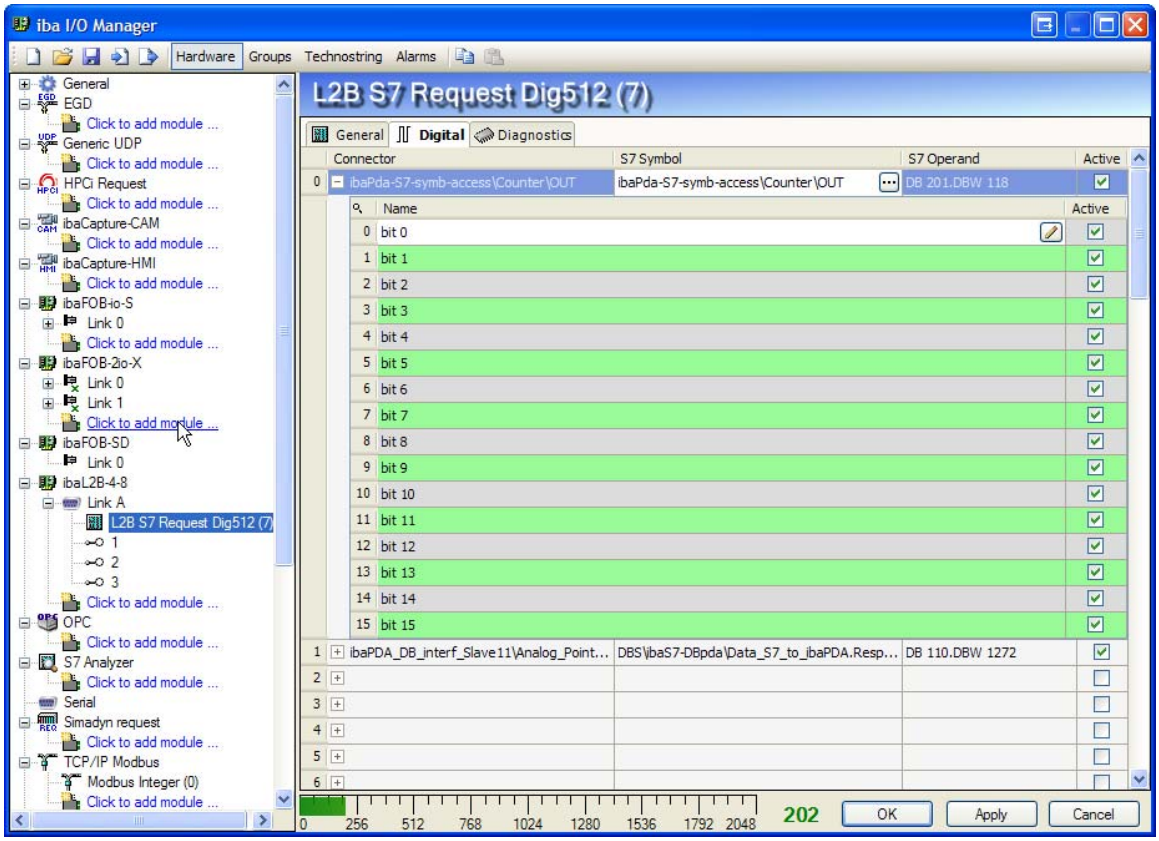
This is a sample and hold function. The output follows 'expr' when 'sample' is 1. It remains constant when 'sample' is 0.

7 S7 changes

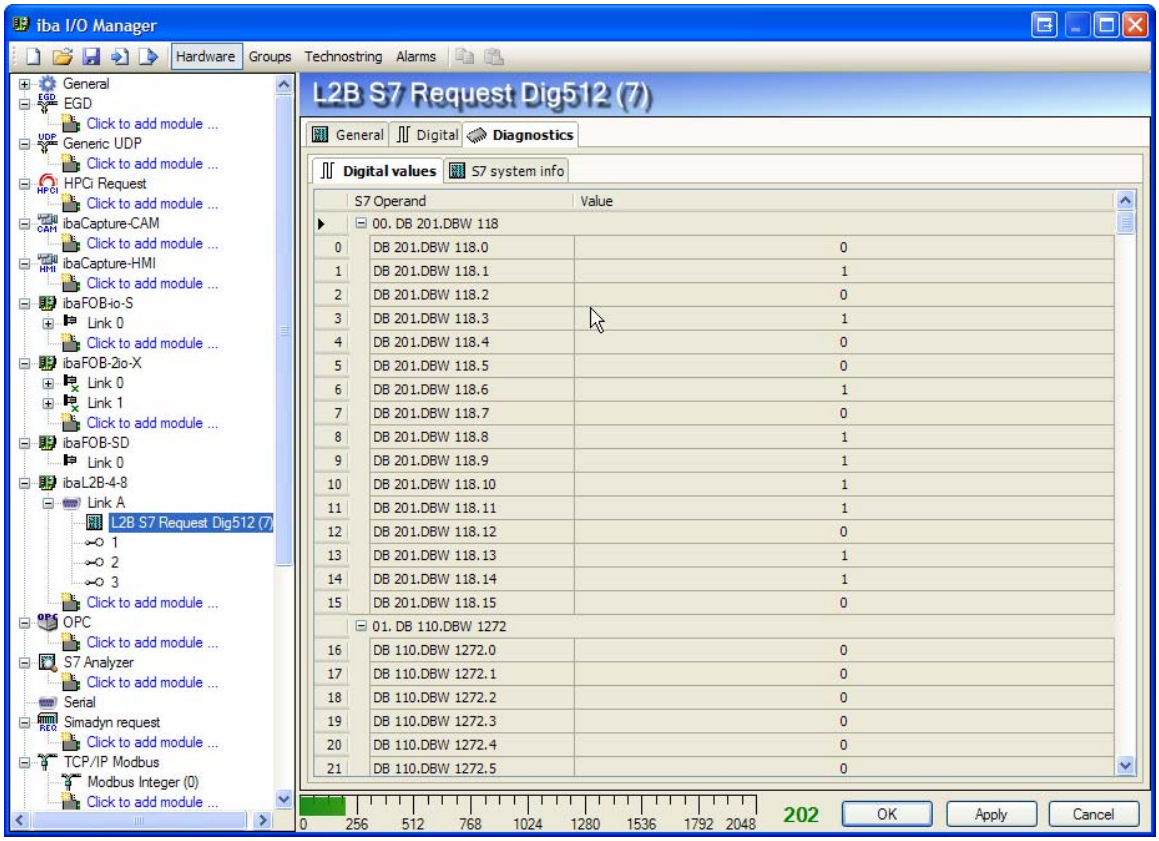
There is a new S7 request module type: L2B S7 request dig512. This module type is similar to an S7 request module on L2B. This module can request 32 word or integer operands. The 16 bits of each operand are measured as 16 digital signals. So this module has 32x16=512 digital signals.



The general tab looks and acts the same as a normal S7 request module.



On the digital tab you have to select an S7 operand or S7 symbol in the main grid. On the sub grid you can give a name to each of the 16 digital signals.

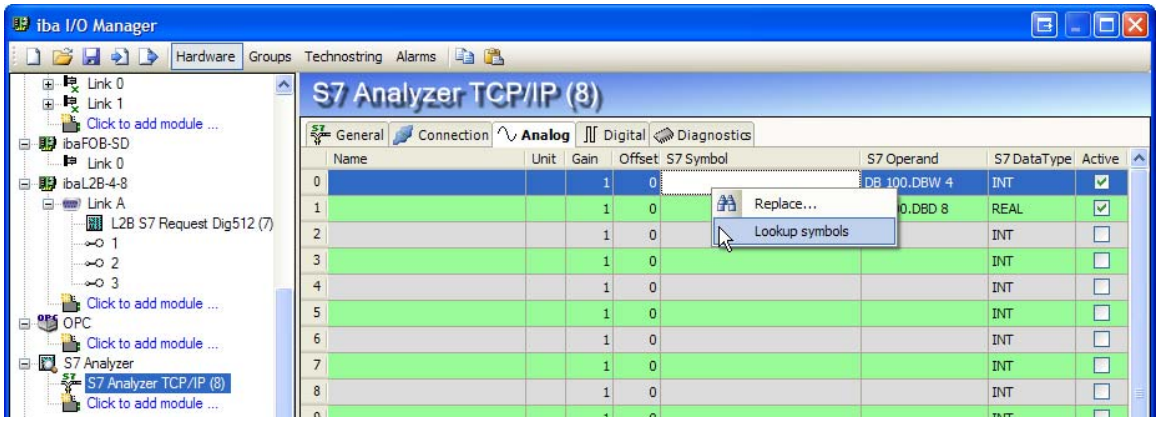


The diagnostics tab shows the values of all 512 digital signals. The signals are grouped by their operand.

If you click the S7 operand column header then ibaPDA will fill in incrementing S7 operands. This works on all module types that have an S7 operand column.

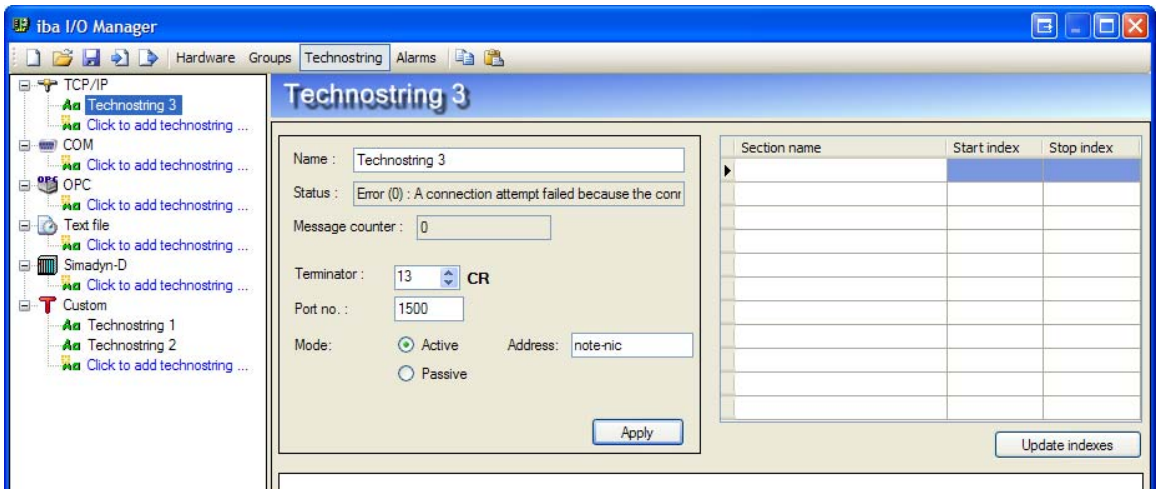
The S7 operand is now filtered. If you open the editor via the analog tab then you will only see the analog operands. If you open it via the digital tab then you will only see the digital operands.

ibaPDA can lookup the symbol that corresponds with an operand. Right-click on the signal grid and select Loopup symbols. ibaPDA will first search the symbol table, then CFC and finally the DBs for the operand.

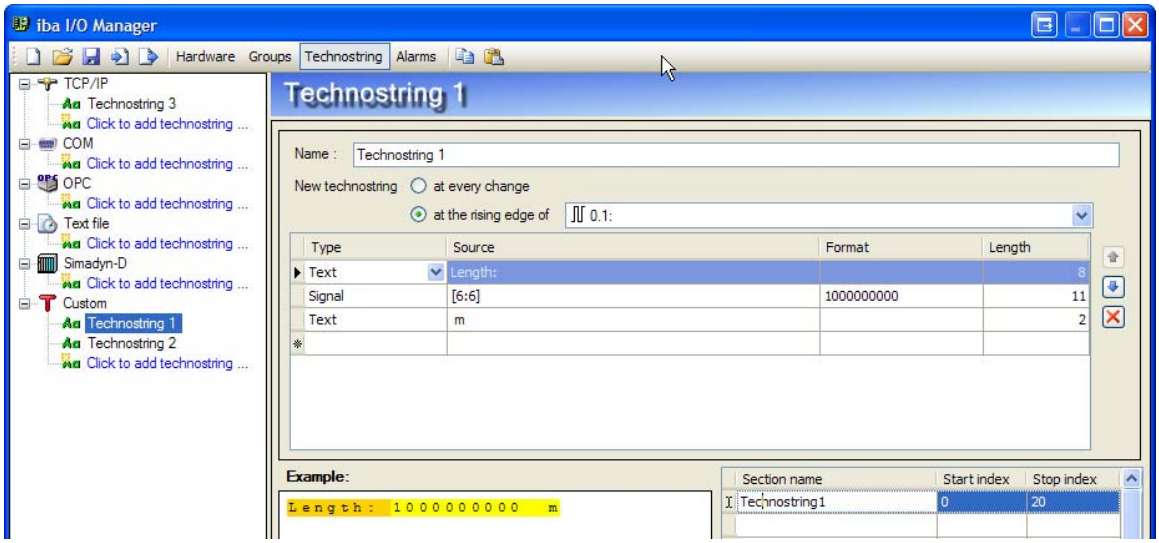


8 Technostring changes

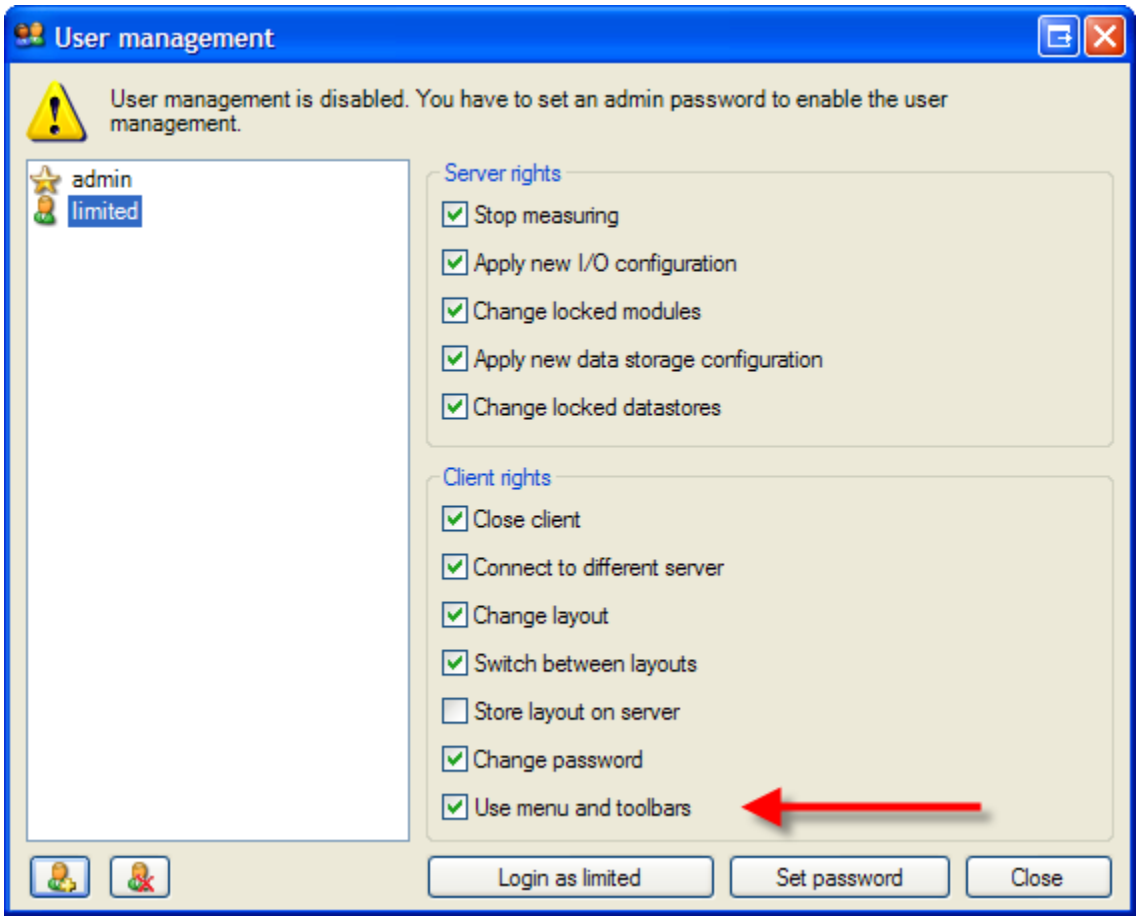
The TCP/IP technostring now supports 2 connection modes: active and passive. In active mode ibaPDA will try to connect to a configurable port on a configurable address. In passive mode ibaPDA listens on a configurable port.



The custom technostring now supports 2 modes. In the first mode a new technostring in generated whenever there is a change in one of the inputs. In the second mode a new technostring is generated at the rising edge of a digital signal.



9 User management changes



There is a new client right in ibaPDA v6.17.0: Use menu and toolbars. If you don't have this right then the menu and all the toolbars are disabled.

At all times you can open the user management dialog with the new keyboard shortcut CTRL+U.