New features in ibaPDA v6.20.0

1 AB Ethernet interface

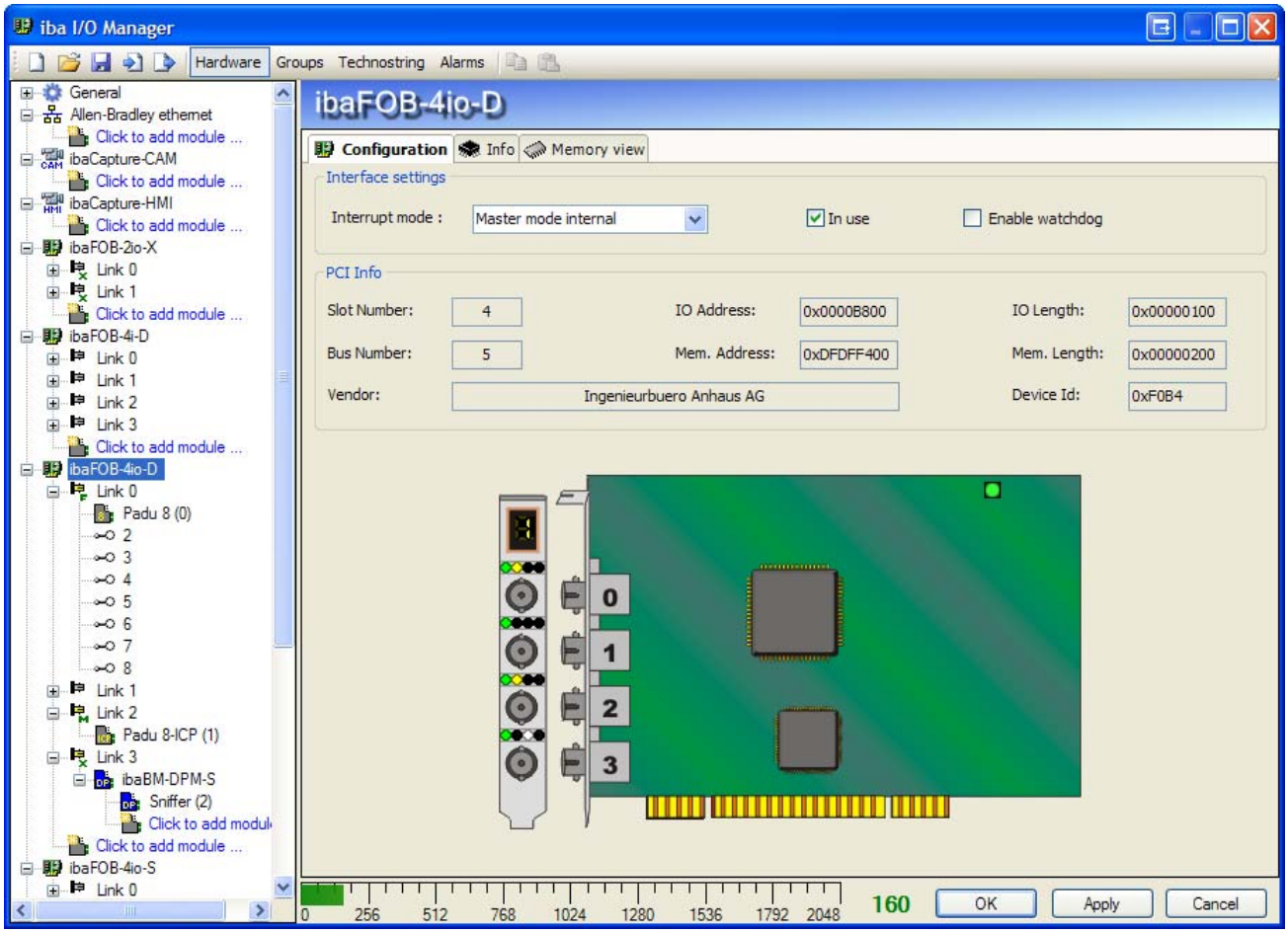
See separate document

2 Generic TCP/IP interface

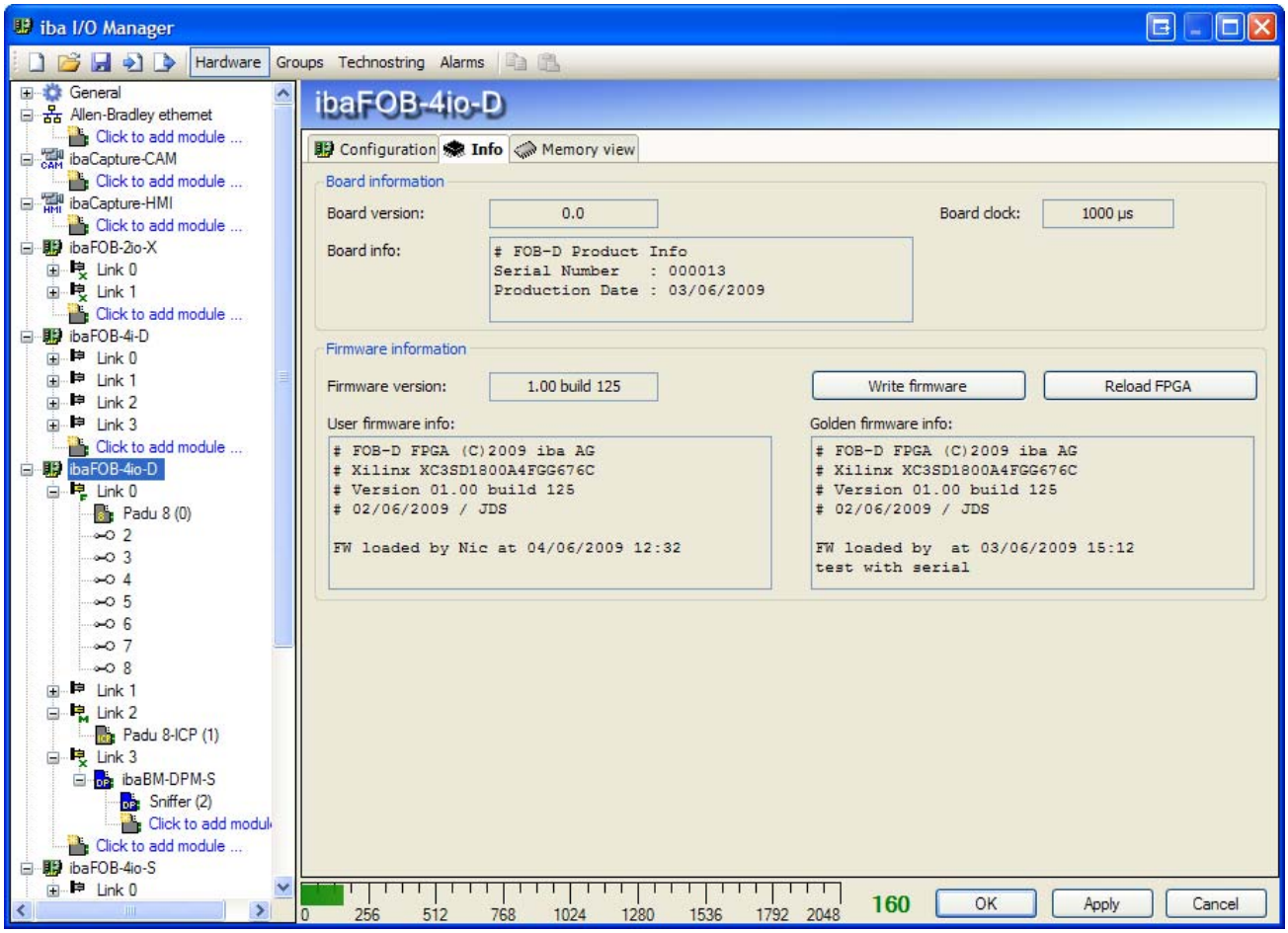
See separate document sw_ibaPDA-TCPIP-Generic_v1.0_en_A4.pdf

3 ibaFOB-4io-D interface

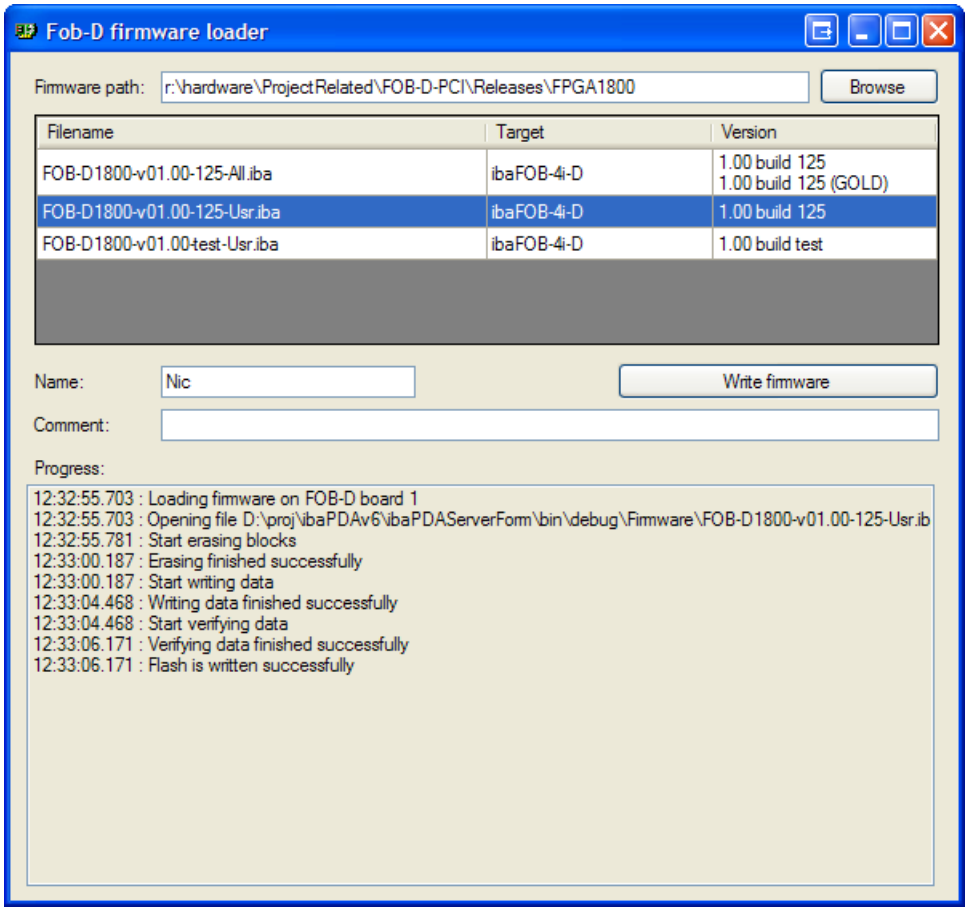
The new FOB-D card is supported in ibaPDA. There are maximum 8 FOB-D boards supported. The FOB-D board can handle all FOB protocols: 2 Mbit/s (old F-mode), 3.3Mbit/s (F-mode), 5Mbit/s (M-mode synchronous and asynchronous) and 32 Mbit/s (X-mode). In pda all the modules that could be added to FOB-F, FOB-S and FOB-X boards can also be added to the FOB-D board. HPCi request, X-Pact request and MMC request work with the FOB-D board.



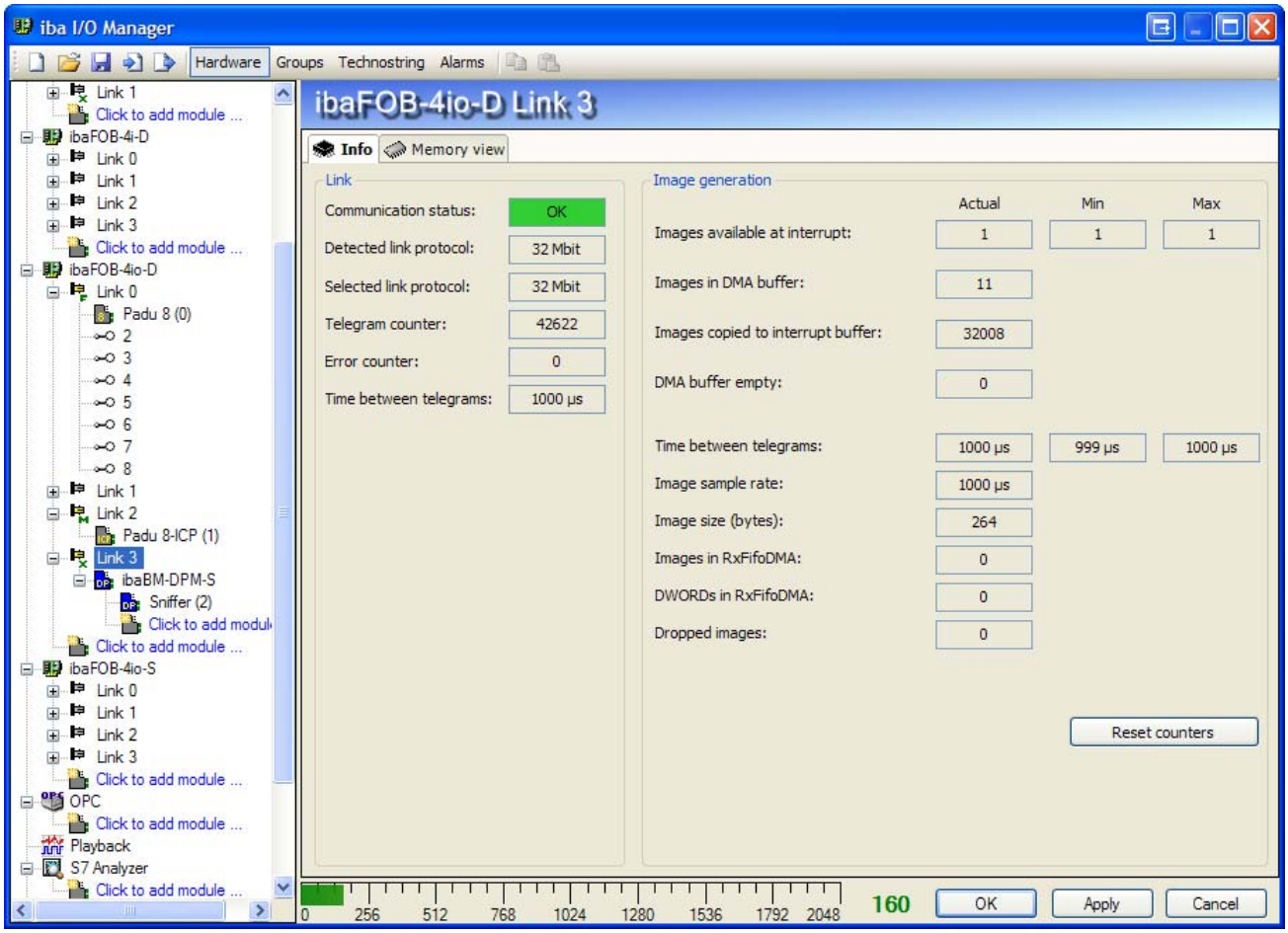
On the FOB-D interface you can setup the interrupt mode and the in use flag like on the other FOB boards. The board watchdog can also be enabled. When the watchdog is enabled then the board will put an alarm on its alarm connector when the acquisition isn't running. When the alarm is on then the red leds of all links will burn and the RGB led will also have a red color.



On the info tab you can see information about the board and the loaded firmware. You can reload the FPGA and you can update the firmware.



When you enter a directory then the grid will display a list of firmware files found in that directory. The grid shows the name of the file, the target board and the firmware version(s) in the file. The file can contain the user firmware or the factory (golden) firmware or both. Normally the golden firmware is only written at production time. Push the write firmware button to write the firmware to the board.



When you click on a link then you get information about this link. The information on the left describes the FO communication. The displayed information will depend on the current protocol on the FO link.

Here is a short description of the FO communication information that is displayed for all protocols:

Communication status: OK when the FO communication is ok. This means that the telegrams that are being received correspond with the mode that is configured on the link. The link mode is determined by the module that is attached to the link, e.g. if a Padu 8 module is configured then the link will be put in 3.3 Mbit/s mode. If a Padu 8-ICP module is configured then the link will be put in 5Mbit/s mode.

Detected link protocol: This is the link protocol that the board detects. This can be 2 Mbit/s, 3.3 Mbit/s, 5 Mbit/s, 32 Mbit/s or ? (no device connected).

Selected link protocol: This is the link protocol that the link is configured in. It is determined by the module that is attached to the link.

Telegram counter: Counter of correctly received telegrams.

Error counter: Counter of received telegrams that have errors (e.g. incorrect checksum). If this counter is changing then it means the FO communication isn't working correctly.

Time between telegrams: The time between the last 2 correctly received telegrams.

The information on the right describes the image generation. An image is a collection of bytes that the board writes into the PC system memory via DMA. This image contains all the data of the measured signals on that link. Here is a short description of the image generation information:

Images available at interrupt: These counters show how many images where available in the DMA buffer when the last interrupt fired. This value should normally correspond with the image sample rate divided by the interrupt time.

Images in DMA buffer: This is the number of images that are in the DMA buffer. This number should remain constant. If this number starts increasing then something is wrong. This can happen if e.g. an interrupt is missed.

Images copied to interrupt buffer: This counter shows how many images have been retrieved from the DMA buffer and have been processed by ibaPDA. This counter should be constantly increasing.

DMA buffer empty: This counter increments each time the DMA buffer is empty when the interrupt fires. The driver will use the value 0 for all signals that are on this link when this happens. This can happen if the FO link is disconnected.

Time between telegrams: The time between the last 2 correctly received telegrams. This is the same as the time in the FO communication information but the driver maintains the minimum and maximum values. There shouldn't be much difference between the minimum and maximum values.

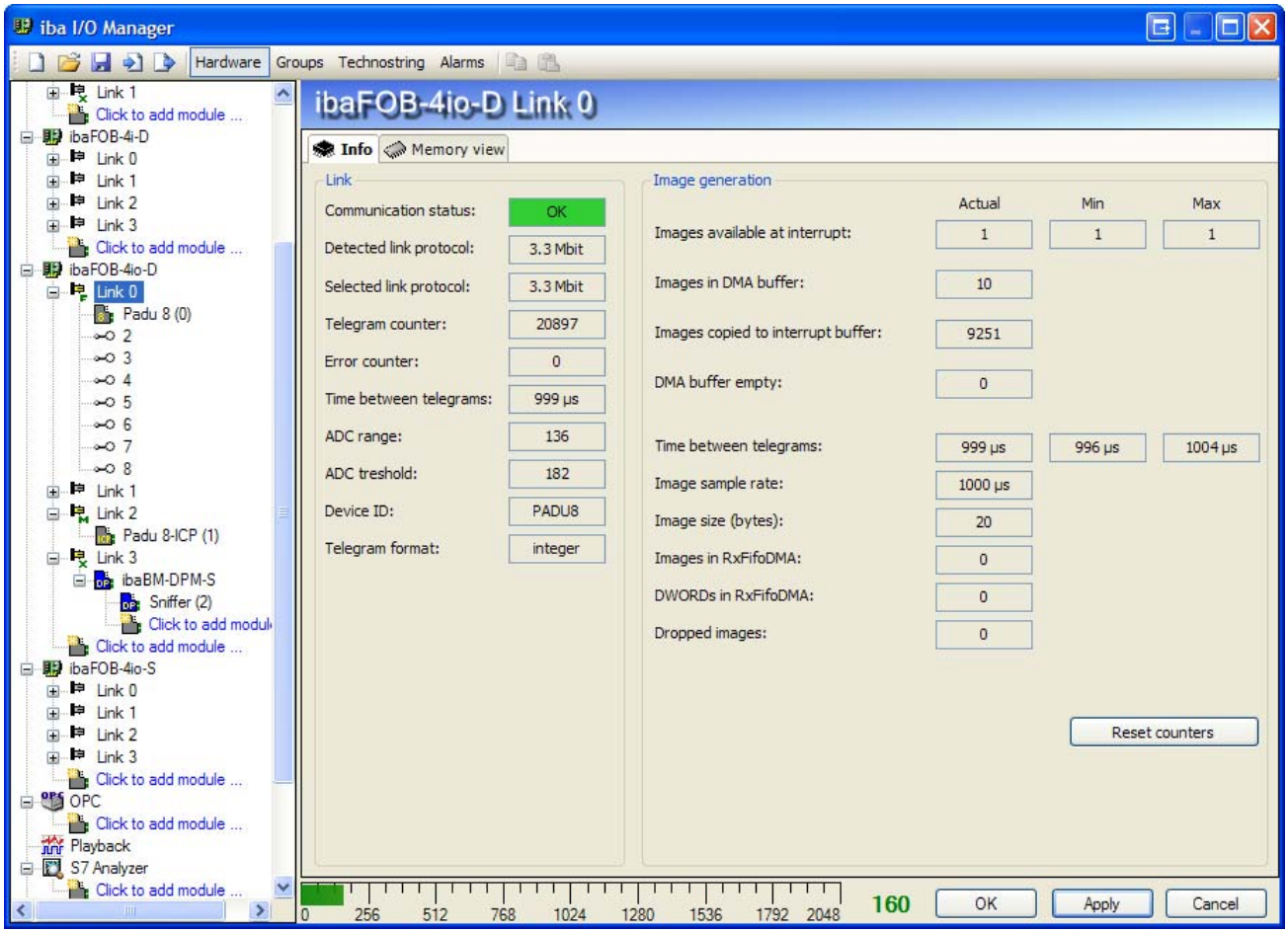
Image sample rate: The rate at which the board writes images to the DMA buffer. This should be faster than or equal to the fastest timebase of the modules connected to this link.

Image size: This is the size of the image in bytes. If you multiply the image size with the image sample rate then you know how many bytes per second are transferred by this link over the PCI bus.

Images in RxFifoDMA: This is the number of images that are in the board's DMA fifo waiting to be transferred over the PCI bus. Normally this value should be 0 or 1. If this value rises then this means that the PCI bus is overloaded.

DWORDs in RxFifoDMA: This is the same counter as the images in RxFifoDMA. The only difference is that this is now expressed in DWORDs.

Dropped images: This counter increments when the board's DMA fifo is full and an additional image arrives. If this happens then something is seriously wrong. This means that the board is unable to transfer images over the PCI bus.



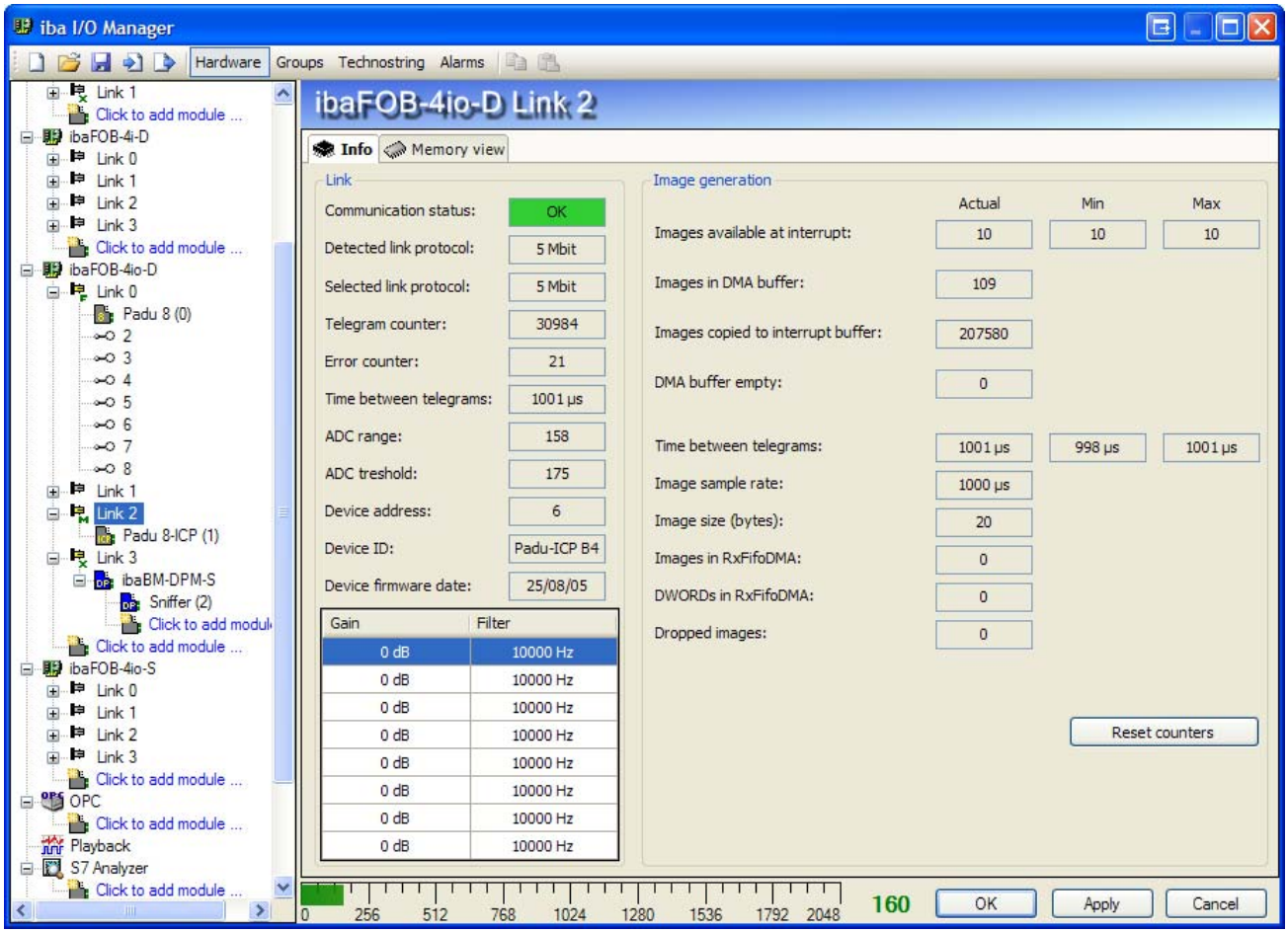
The screenshot shows the information you see when the link is in 2 Mbit/s or 3.3 Mbit/s mode. There are 4 items that are 2 and 3.3 Mbit/s specific:

ADC range: This is the difference between the maximum value and the minimum value received from the FO A/D converter. This can be maximum 255. The higher this value is the stronger the FO input signal is.

ADC treshold: This is the threshold used by the board to determine if a 1 or a 0 is received from the FO A/D converter. This threshold is automatically adjusted depending on the received ADC range.

Device ID: This is the ID of the last device in the FO chain connected to this link.

Telegram format: This is the format of the analog data that is transferred in the telegram. The possible values are integer, real and S5 real.



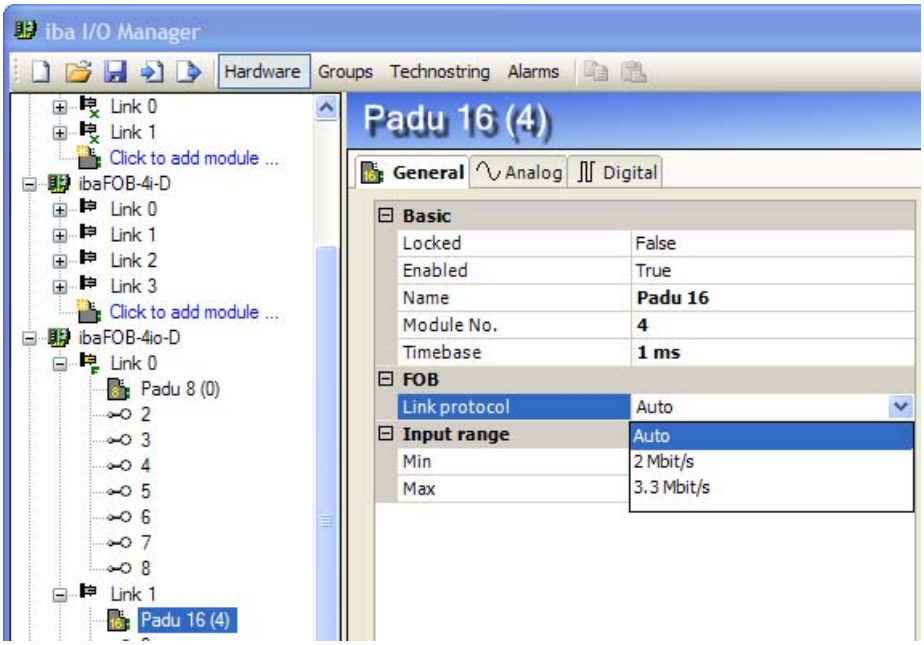
The screenshot shows the information you see when the link is in 5 Mbit/s mode. The ADC range, ADC threshold, device address and device ID are already explained for 3.3 Mbit/s.

Device firmware date: The firmware date of the connected device.

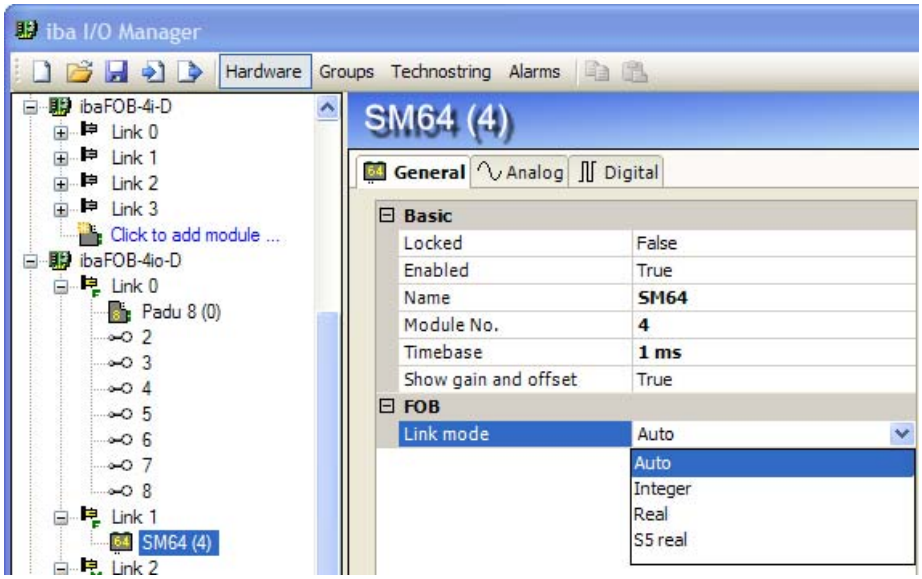
Gain and filter grid: The gains and filters that are configured in the device. This only applies to the Padu 8-ICP device.

All the modules that can be connected to the FOB-D are the same as the modules that can be connected to FOB-F, FOB-S and FOB-X. There are only 2 changes made.

The Padu 16 and Padu 32 devices with a serial number lower than 1000 use the 2 Mbit/s protocol. The newer devices use the 3.3 Mbit/s protocol. On previous boards the board automatically converted 2 Mbit/s telegrams into 3.3 Mbit/s telegrams. The FOB-D board doesn't do this so pda needs to know what the link protocol is going to be. That is why the padu 16 and padu 32 modules have an extra property called **link protocol** when they are connected to FOB-D. This property can be set to Auto, 2 Mbit/s or 3.3 Mbit/s. If it is set to Auto then the protocol is determined automatically when the acquisition is started. This will only work when the FO link is connected at the start of the acquisition.

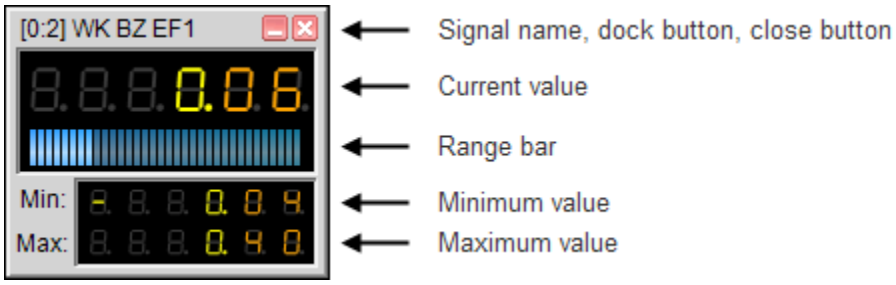


The second change is on the SM64 module. The SM64 device can be put into an S5 system. The SM64 will then send S5 reals instead of standard IEEE 754 reals. The previous boards automatically converted S5 reals into IEEE 754 reals. The FOB-D has to be configured to do this. That is why pda needs to know the incoming telegram format. The link mode property is extended with the S5 real format when an SM64 is connected to a FOB-D link.

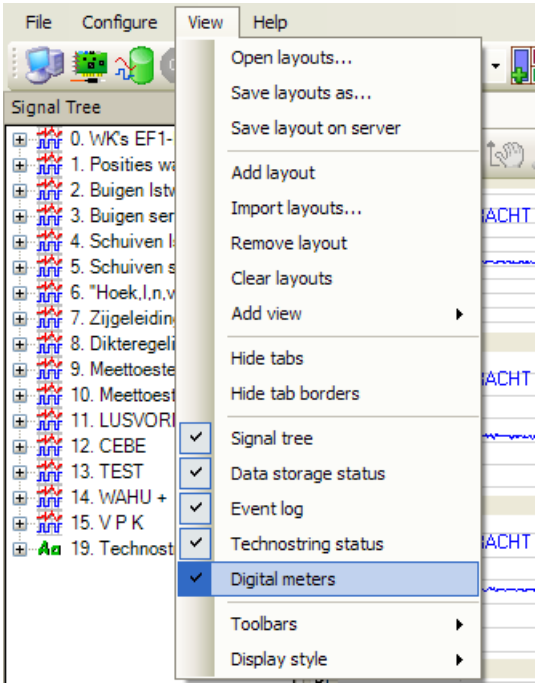


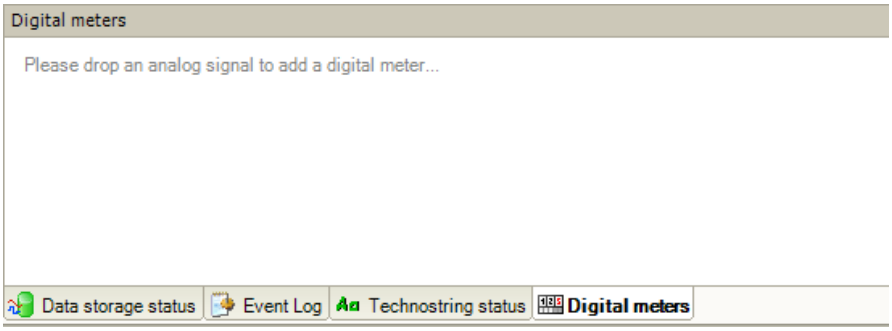
Similar to the FobFLinkStatus, FobMLinkStatus and FobFastLinkStatus, there is now a FobDLinkStatus function. It returns the status of a FOB-D link no matter which protocol is currently being used.

4 Digital meters



In v6.20.0 the digital meters from pda version 5 are back. A digital meter shows the actual value of an analog signal. Optionally it can show the minimum and maximum value of the signal. It can also show a range bar that shows the position of the current value between the minimum and maximum value of the signal. The digital meters can be floating or they can be docked in the digital meters panel. The digital meters panel can be shown via the View menu.

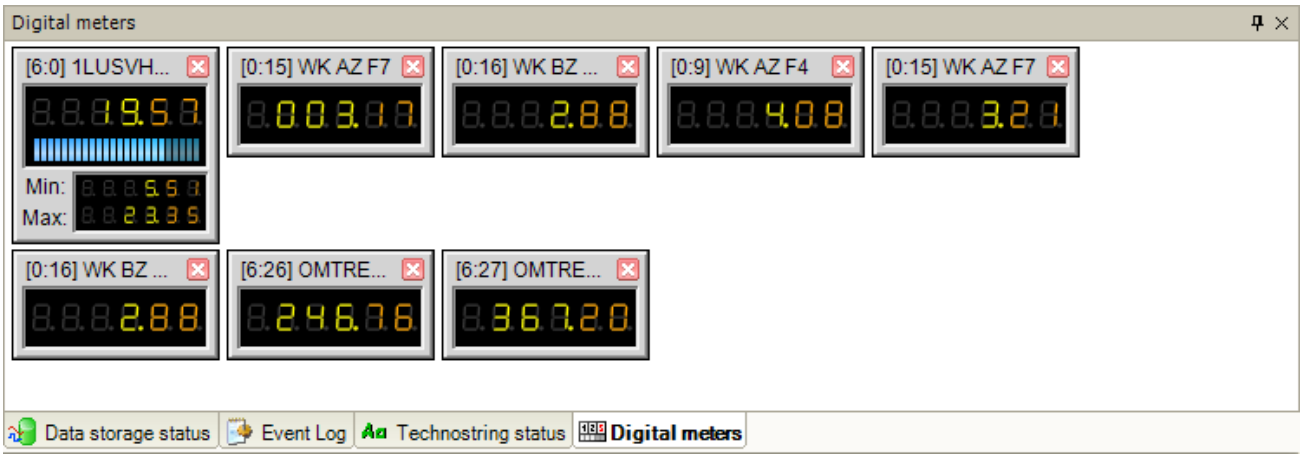




There are 2 ways to create a digital meter:

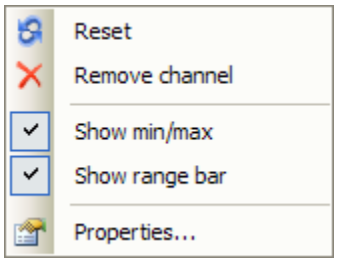
- Right-click an analog signal in the signal tree and select “Add to digital meters” from the context menu
- Drag an analog signal from the signal tree and drop it on the digital meters panel

The digital meter panel arranges the meters in a flow layout. This means that the meters are arranged from left to right. When there is no more place left at the right then a second row is started.

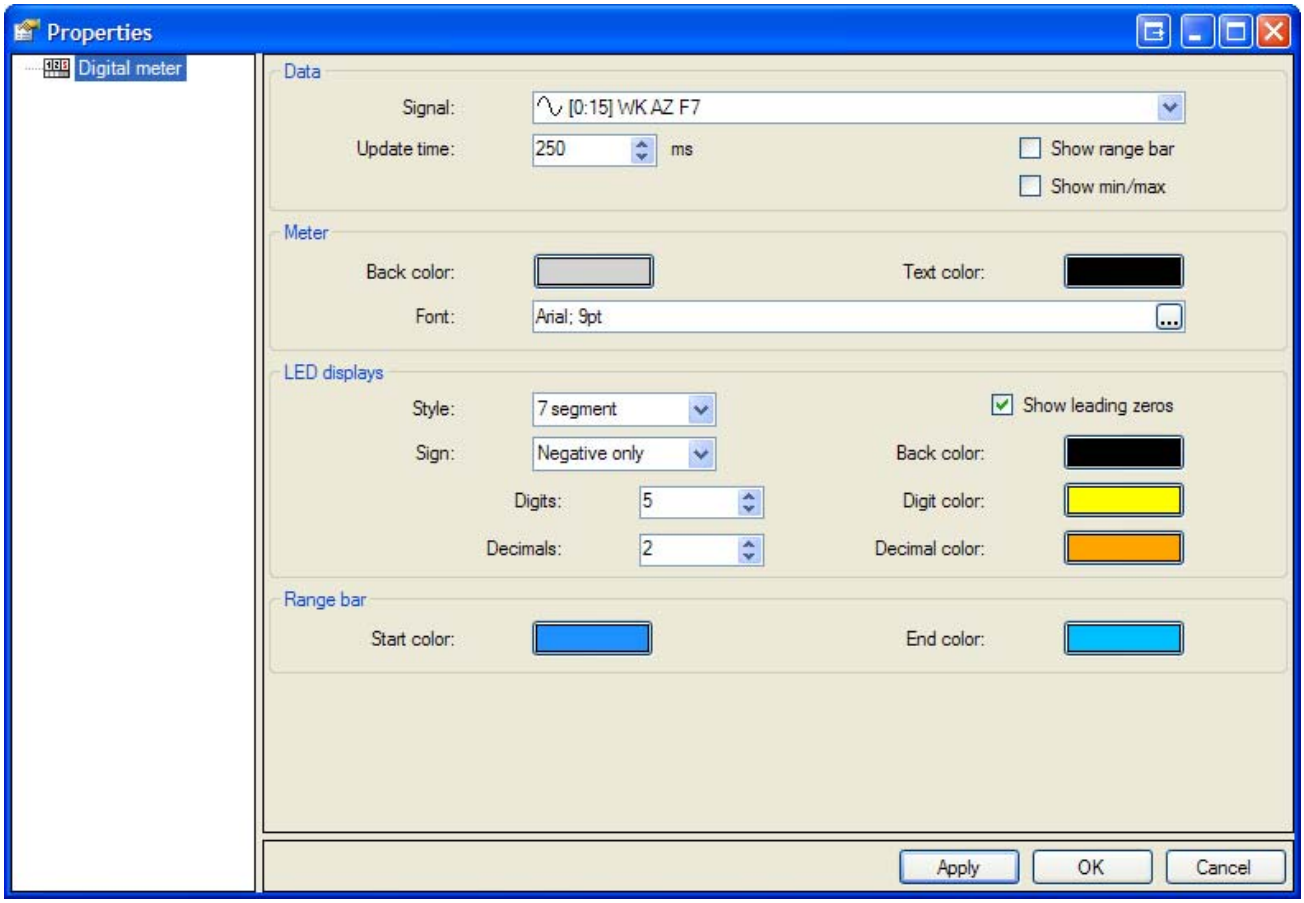


The order of the meters in the panel can be changed by dragging the meters. You can also drag a meter from the panel and drop it outside the panel. The meter will be floating then. The meter can be docked again by dragging it back on to the panel or by clicking the dock button. You can remove a digital meter by clicking its close button.

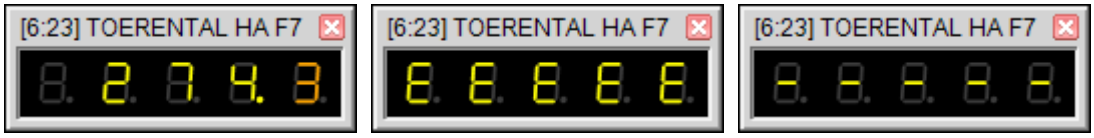
If you right-click a meter then its context menu is displayed.



The reset command will reset the minimum and maximum values. The properties command shows the properties form. In this form you can select the displayed signal, the font, the colors, the formatting, …



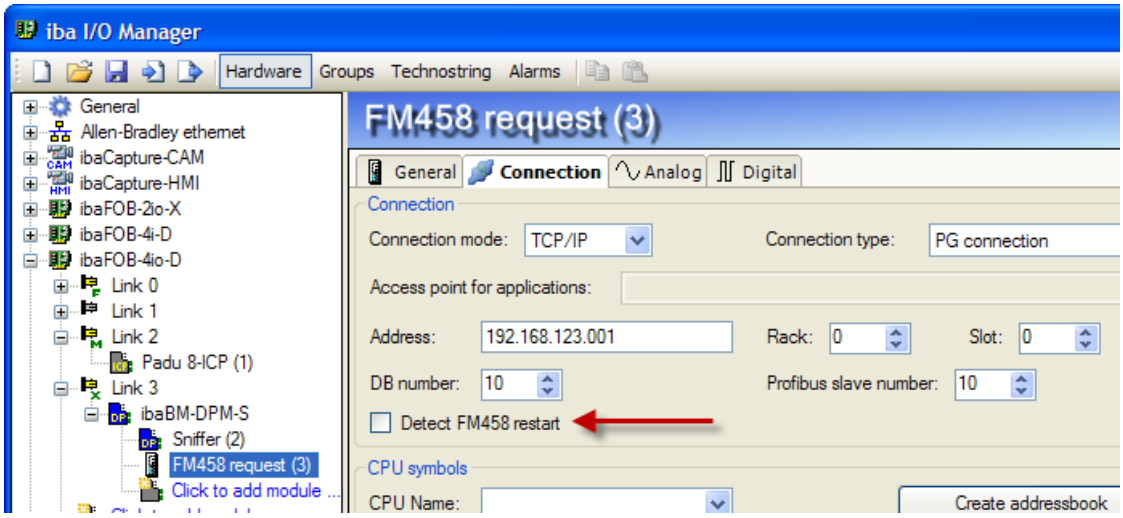
In the properties form you can setup how many digits are displayed. You can also specify how many of these digits are used as decimals. If the signal value is larger than can be displayed by the number of digits then the meter displays E for all its digits. This means overflow. If the signal is no longer valid then the meter displays – for all its digits.



The digital meters, their properties and their positions are saved per layout.

5 Small changes

5.1 FM458 restart detection



If this option is enabled then the acquisition is automatically restarted when pda detects that an FM458 has restarted. This is a global option. If you enable this on 1 FM458 request module then it is enabled on all FM458 request modules.

5.2 Sisteam generic module

Similar to the generic VIP and modbus modules you can now create a generic Sisteam module. The module indexes start at 200. The maximum data size of a Sisteam telegram is 1024 bytes.

5.3 Playback improvements

Playback now also supports signals with a timebase smaller than 1 ms.

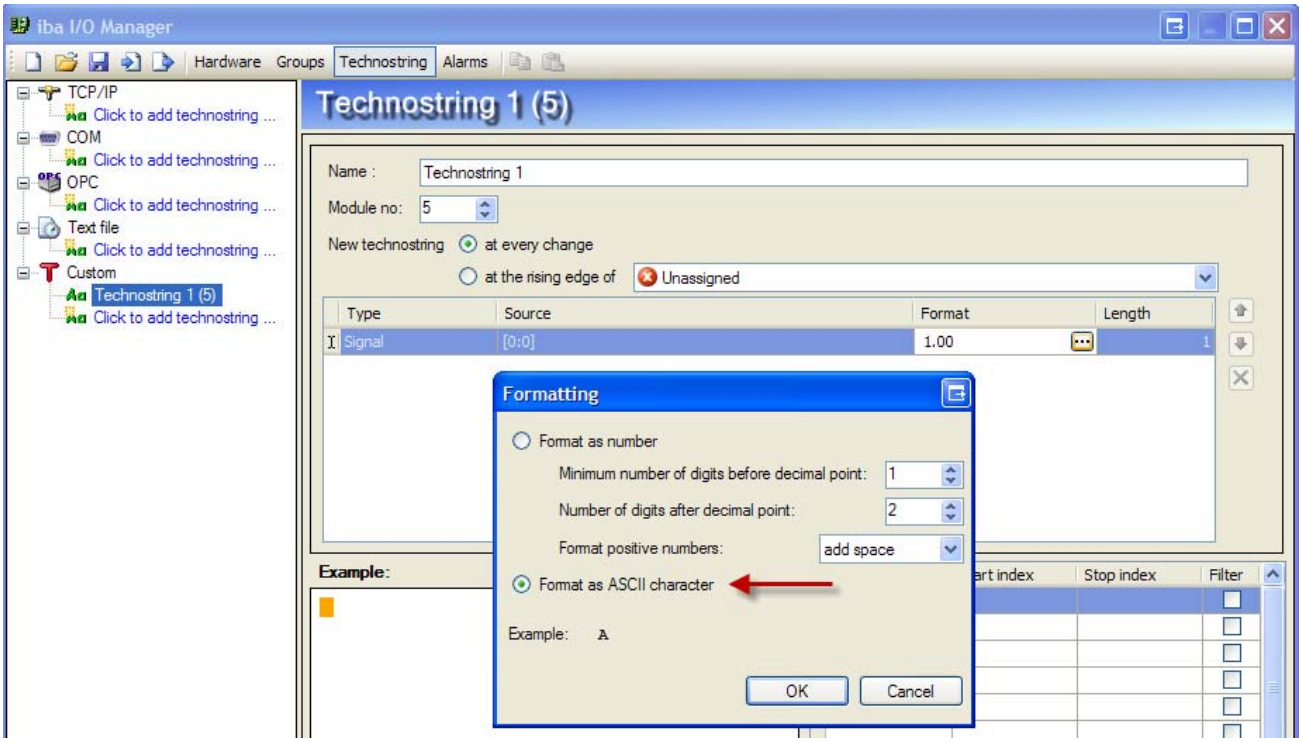
The loading of dat files is now more tolerant to ‘incorrect’ dat files. These are dat files not written conform the standard. ibaScope for example writes incorrect signal numbers for digital signals in the dat file. Playback will not be able to playback all digital signals from a dat file written by ibaScope.

5.4 Integer, real and generic TCP/IP modules

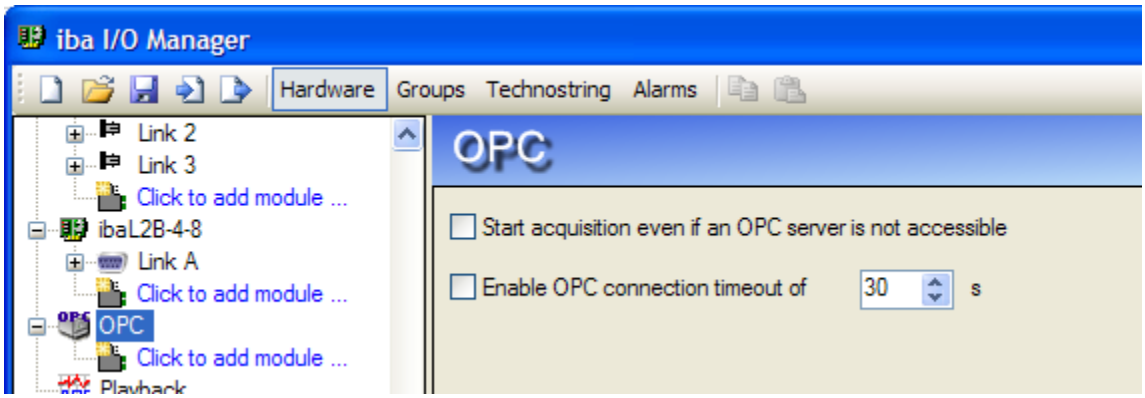
The integer, real and generic TCP/IP modules of 1 TCP/IP protocol can’t overwrite each others’ data. In previous versions VIP modules with module indexes 0 (=integer) and 100 (=real) would overwrite each others’ data. This has lead to numerous customer support calls. Now this configuration is allowed and the modules will get the correct data.

5.5 Signal value as ASCII character in custom technostring

A signal value can now be interpreted as an ascii character in the custom technostring. Select Format as ASCII character in the formatting form.



5.6 OPC connection timeout



There is an option to set the OPC connection timeout. If the connection to an OPC server doesn't succeed within this time then ibaPDA continues. In previous versions a failing OPC server could hang ibaPDA.

5.7 HPCi lite supports multiple DGM200P boards

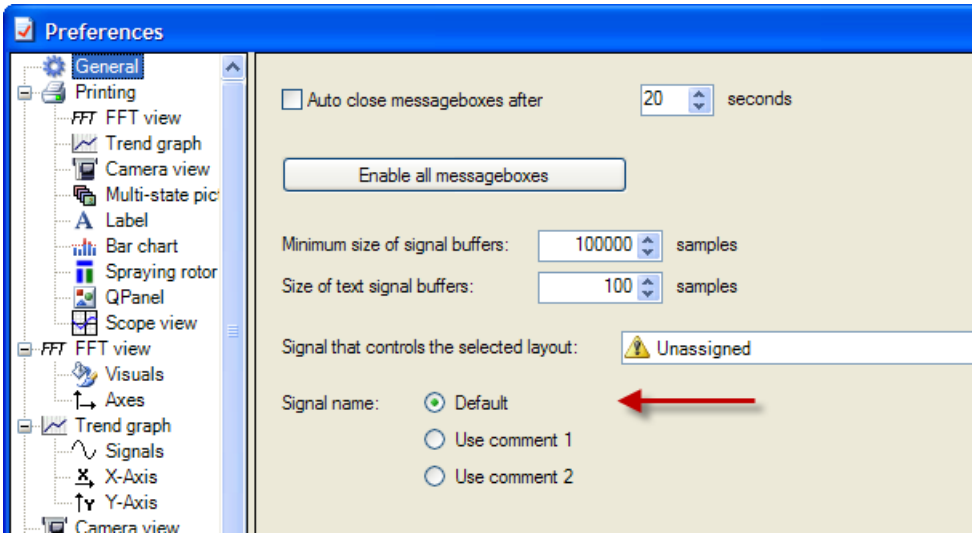
HPCi lite now supports a CC100 addressbook per DGM200P board. In previous versions there was only 1 global CC100 addressbook that was used for all DGM200P boards. The CC100 addressbook names are fixed. They must be cc100.tsv for board 0, cc100_1.tsv for board 1, cc100_2.tsv for board 2 and finally cc100_3.tsv for board 3. If there is no specific addressbook available for a board then addressbook cc100.tsv is used. The addressbooks must all be in the same directory as the toc.ini file.

6 Client changes

6.1 Alternative signal names

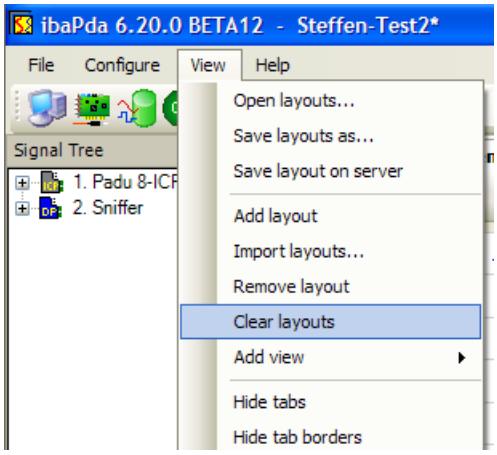
The signal name shown in the signal tree and in the different views can come from 3 sources:

- The signal name itself (default)
- Comment 1
- Comment 2



Open the preferences form to change the signal name source. This feature can be used to give signals different names in different languages. An integrator can use English signal names and his Chinese customer can use Chinese signal names.

6.2 Clear layouts menu item



The clear layouts menu item removes all the layouts and clears the current layout.

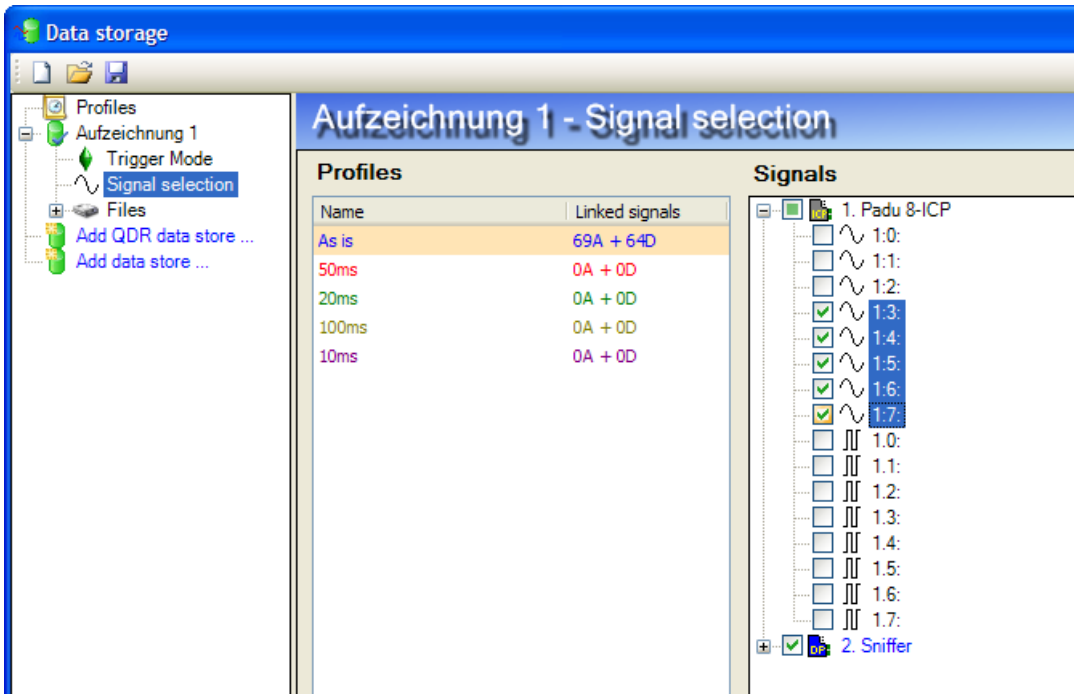
6.3 Trend graph improvements

In the previous versions the signal bars on a trend graph were only updated when the trend graph scrolled. If the trend graph showed a large X-range then the signal bars would only repaint every few seconds. This wasn’t very usefull. Now the signal bars are repainted each time there is new data.

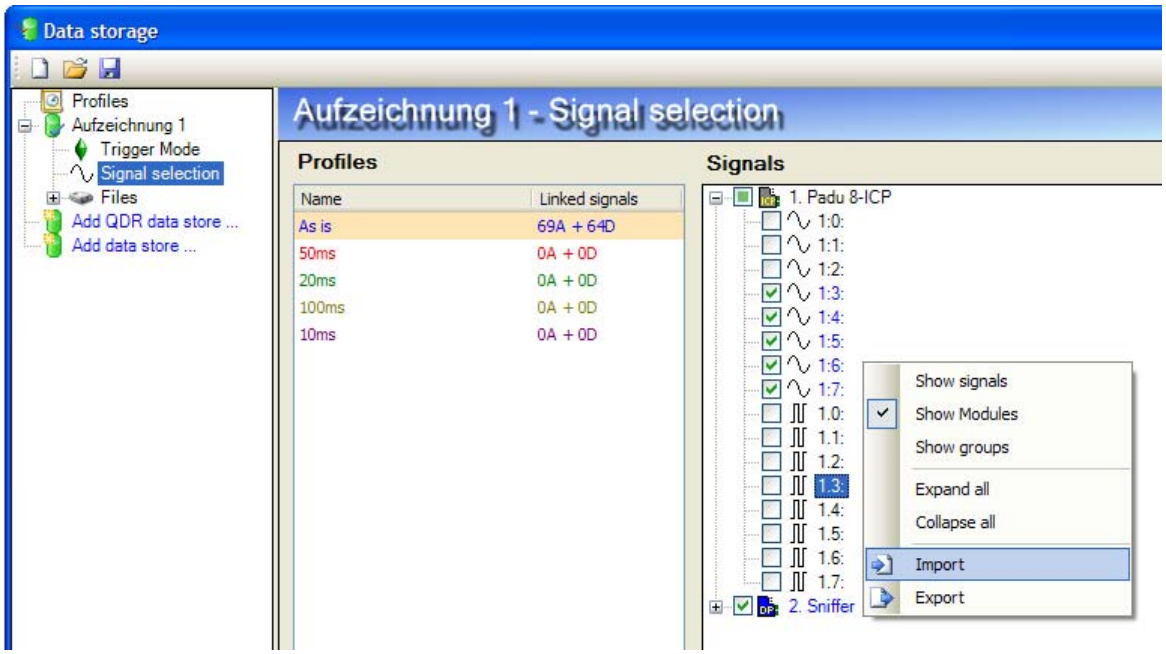
The range of the X-axis in the trend graph can be controlled by an analog signal. This feature was previously limited to the trend graph on a Q-Panel.

The legend tooltip now shows both signal comments instead of only the first one.

7 Data storage configuration changes



The signaltree in the signal selection of the data storage configuration form now supports multiselect. You can select multiple nodes via SHIFT or CTRL. If you then click on the checkbox of one of the selected signals then all the checkboxes of all the selected signals will change.



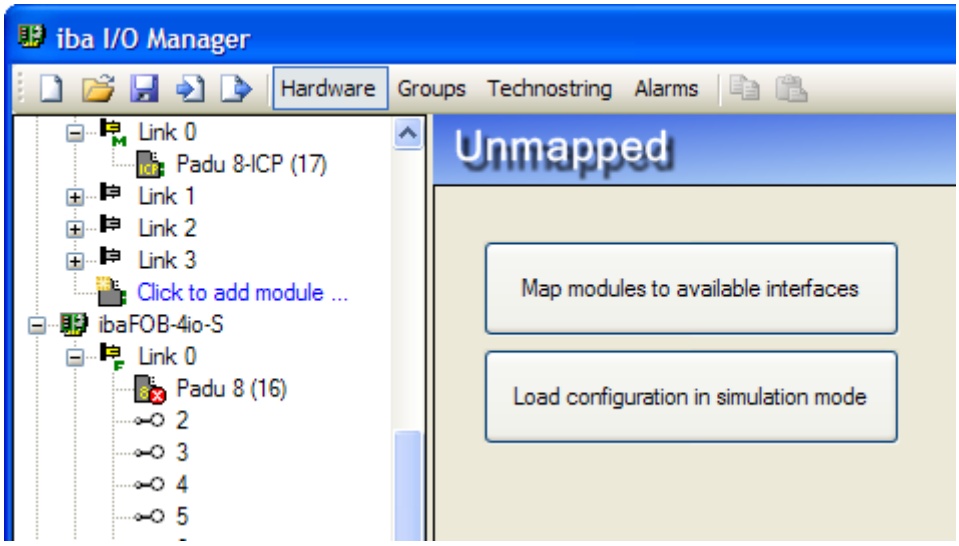
The signal selection can also be exported to a text file and it can be imported back. The text file is a tab separated file with 3 columns: signal number, signal name, profile. There is a row for every signal in the I/O configuration. Right-click the signal tree to show the context menu.

8 Iba internal features

These features are only meant for iba internal use.

8.1 Opening I/O configuration in simulation mode

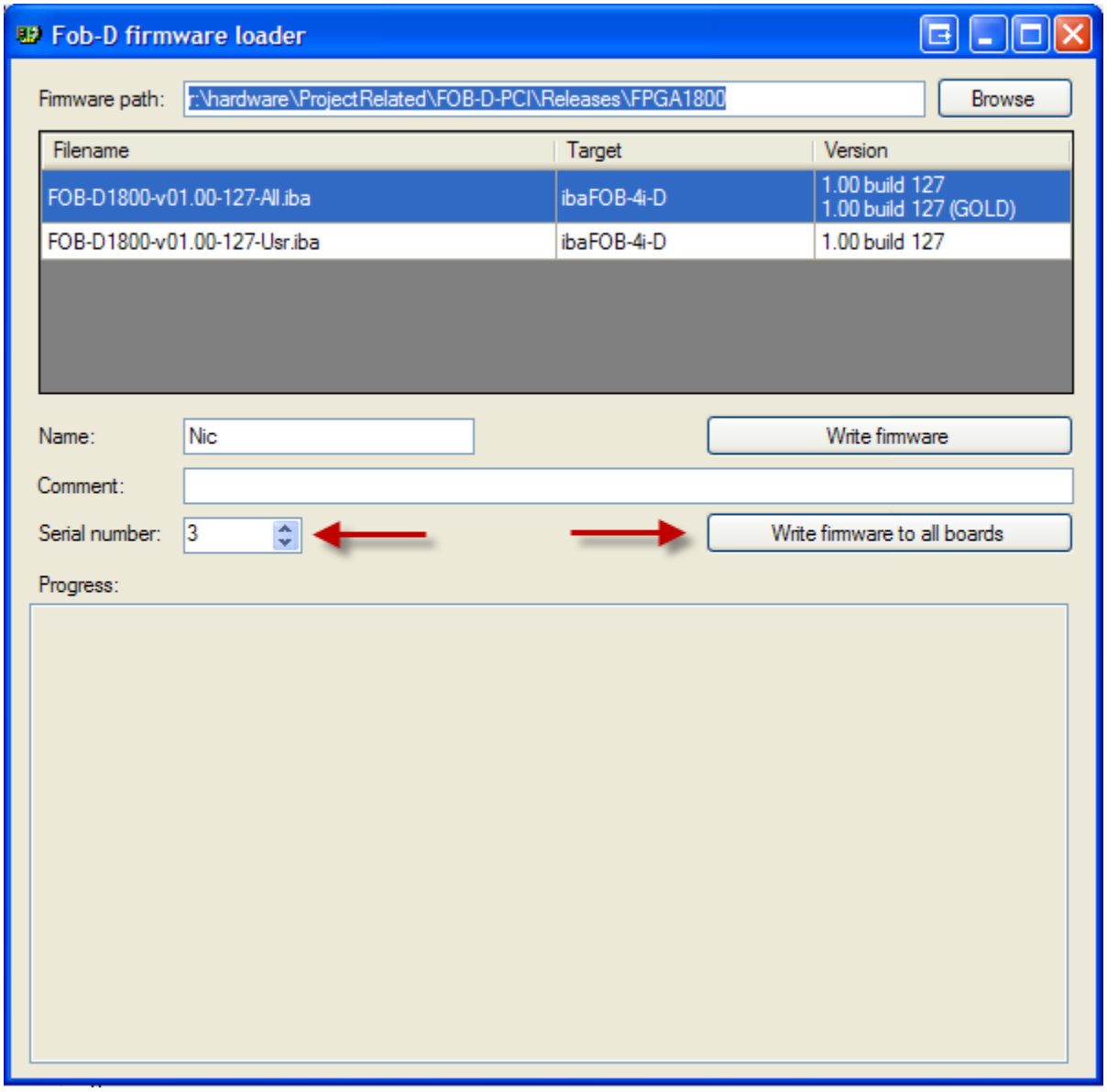
When you normally load an I/O configuration then pda will change the configuration by adding or removing interfaces depending on the boards and license bits currently present on the pda server. This can be quite annoying if you want to view the configuration of a customer. In pda 6.20.0 there is a new button on the unmapped interface. When you click the ‘load configuration in simulation mode’ button then it will show a file open dialog. Select the .io file you want to view. The .io file will be loaded without making changes to its content. So you will see all the interfaces and modules in the .io file no matter which boards or license bits the pda server has. You won’t be able to apply such a configuration. The ok and apply buttons are disabled.



8.2 Setting serial number of FOB-D boards

There is a new command line switch for the pda client. If you start it via ‘ibaPda.exe /production’ then you will be able to write the serial number of FOB-D boards.

The FOB-D firmware loader form in production mode contains 2 extra items. A serial number input and a ‘Write firmware to all boards’ button. The serial number is automatically incremented after it is written to a board. The value of the serial number is also stored in the registry so when you restart pda it will show the last number. With the new button you can write the same firmware and incrementing serial numbers to all FOB-D boards in the PC.


Fob-D firmware loader
Firmware path:
r:\hardware\ProjectRelated\FOB-D-PCI\Releases\FPGA1800
Browse
Filename
Target
Version
FOB-D1800-v01.00-127-All.iba
ibaFOB-4i-D
1.00 build 127
1.00 build 127 (GOLD)
FOB-D1800-v01.00-127-Usr.iba
ibaFOB-4i-D
1.00 build 127
Name:
Nic
Write firmware
Comment:
Serial number:
3
Write firmware to all boards
Progress: