New Features in ibaLogic v5.3.0

Author: iba AG Fürth

Date: 16/03/2018

Content

1 Clean up of the evaluation order

The evaluation order in ibaLogic automatically results from the logic of the interconnected blocks.
For example, for an output, all previous blocks must be calculated, starting with the first block.

An exception to this is an ambiguity caused by parallel branches or feedbacks. Since it is not clear where the "beginning" of a feedback chain is, the first calculated block is searched for on the basis of its position. The search progresses from top to bottom (Y position) and left to right (X position).
The consequence of this is that moving blocks may lead to a different order of evaluation.
This has always been the case in all previous ibaLogic versions.
Problems with this algorithm are also unknown in the time since ibaLogic V3.

Because of a mistake in a certain constellation in feedbacks, the algorithm of the evaluation order was corrected
This now leads to a different kind of evaluation order from version 5.3.0.
A fundamentally different behavior of ibaLogic is not to be expected. However, in individual cases it could lead to a different behavior for timing tricks and calculations.

To make the evaluation behavior more transparent for the user, the following new features have been implemented in ibaLogic 5.3.0. The individual points are also explained in detail in the following chapters:

1) There is the ibaLogic intermediate version 5.2.4, with the possibility to export the evaluation order as txt files (per task). This makes it easy to see the difference between the old and new evaluation order through file comparison

2) There is a new view of FEEDBACKs so they can easily be found.

4) The problem of feedbacks can be resolved. For this purpose, a new function block FEEDBACKBREAKER was introduced. It corresponds to the MOVE block known from ibaLogic V3 and determines the feedback point of a feedback. This virtually eliminates feedback.

5) In order to better understand the general behavior of the task with evaluation order and possible interrupts / time shifts / running times, a new tool IBALOGIC V5 TIMING DIAGNOSTIC was introduced. This tool records all task states, such as RUN / REST / INTERRUPTED, with μs resolution as a dat file. This makes it possible to analyze the exact behavior of the tasks.

2 Export of the evaluation order

In the evaluation order view, there is a new context menu, which can be called up with the right mouse button. There you have the possibility to export the evaluation order as a text file. This function is available as of version 5.2.4.

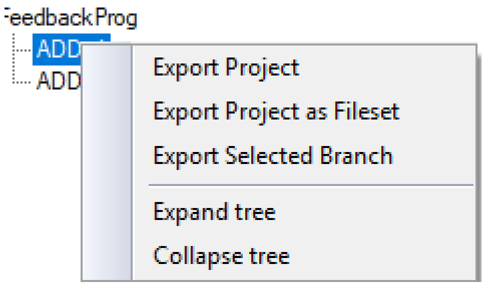
The 5.2.4 still uses the "old" algorithm for the evaluation. Therefore you can compare an export from the 5.2.4 with an export of a 5.3.0.

You can choose between different export forms:

Export Project: Export the entire evaluation order to a single file

Export Project as File Set: Each task is exported to a single file

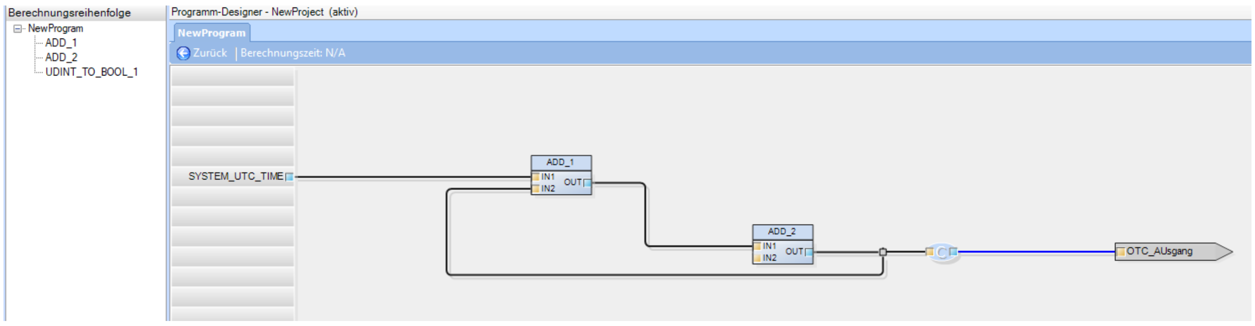
Export Selected Branch: Only the selected branch is exported to a file. This allows you to export only one task / macro etc.

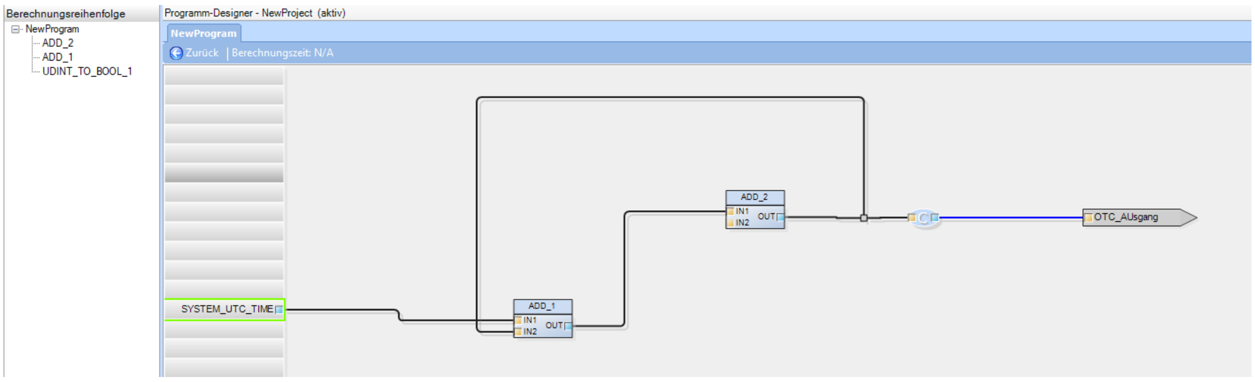


3 Feedbacks and their resolution

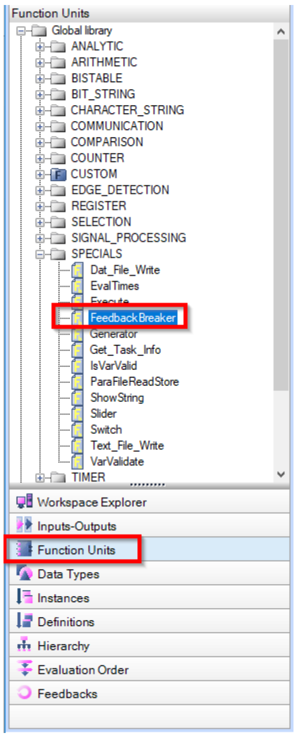
Feedbacks are handled in ibaLogic so that the function block that comes first is calculated first. "First" means going through the program from top to bottom and from left to right.

Therefore, depending on the layout, a different order of evaluation results (see left):

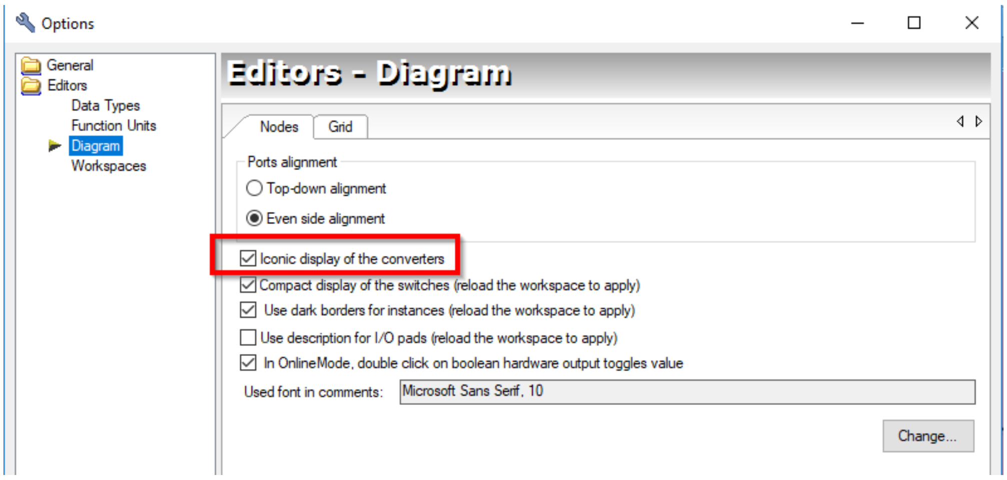




This behavior can now be fixed.
For this you must determine the feedback point.
There is the function block FEEDBACKBREAKER.



The gaphical representation of the block is bound to this option:


Options

- General
- Editors
  - Data Types
  - Function Units
  - Diagram
  - Workspaces

Editors - Diagram

Nodes | Grid

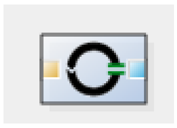
Ports alignment
- ( ) Top-down alignment
- (●) Even side alignment

- [x] Iconic display of the converters
- [x] Compact display of the switches (reload the workspace to apply)
- [x] Use dark borders for instances (reload the workspace to apply)
- [ ] Use description for I/O pads (reload the workspace to apply)
- [x] In OnlineMode, double click on boolean hardware output toggles value

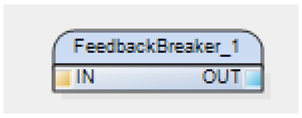
Used font in comments: Microsoft Sans Serif, 10

Change...

If this option is set (default setting):

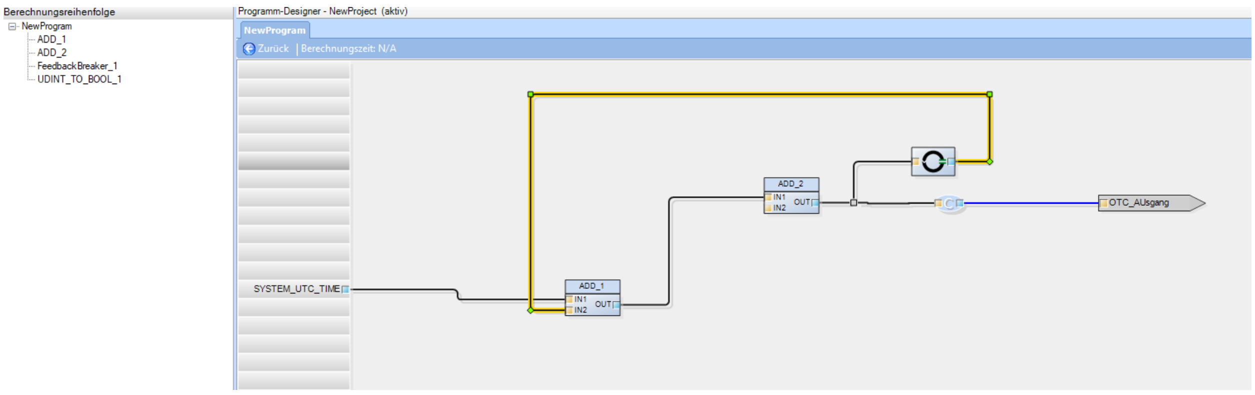


If this option is not set:

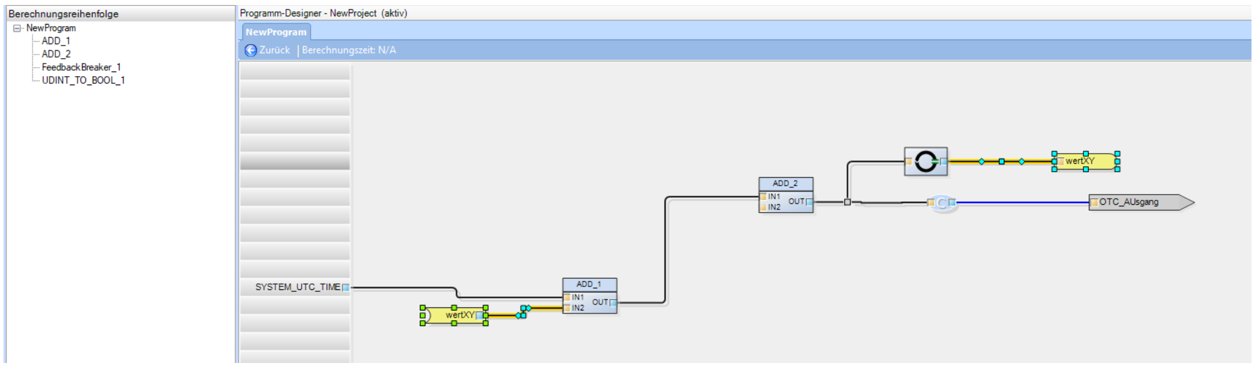


This FeedbackBreaker must be placed as a feedback point.

The block marks the end point of the feedback, so now the order is clearly defined.

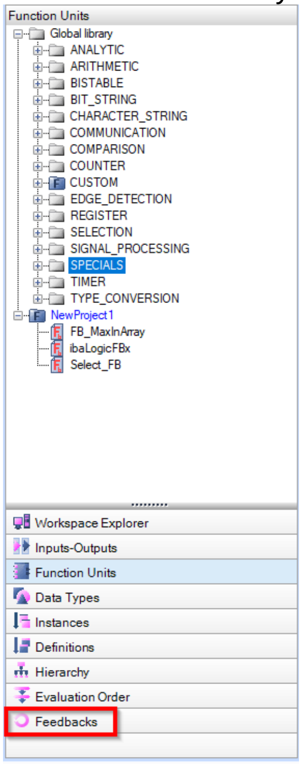


For a better understanding and for a better overview, it is recommended to work with IntraPage connectors. The layout looks like this:

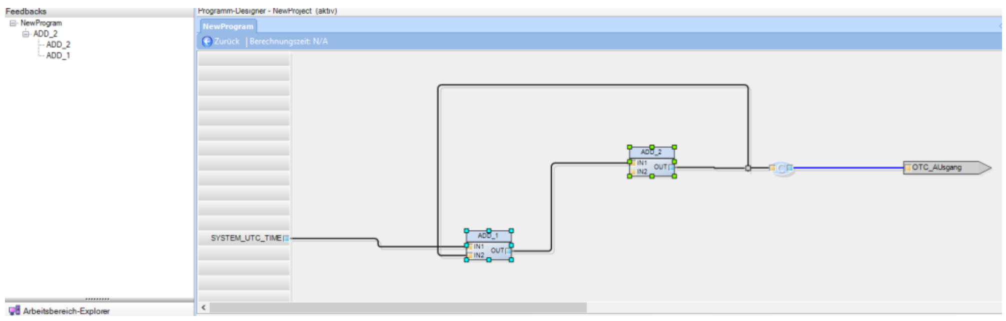

Berechnungsreihenfolge
NewProgram
ADD_1
ADD_2
FeedbackBreaker_1
UDINT_TO_BOOL_1
Programm-Designer - NewProject (aktiv)
NewProgram
Zurück
Berechnungszeit: N/A
SYSTEM_UTC_TIME
wertXY
ADD_1
IN1
IN2
OUT
ADD_2
IN1
IN2
OUT
wertXY
OTC_AUsgang

3.1 Directory of all feedbacks

To find the feedbacks in your layouts, ibaLogic has a new view:



In this view, you will find all the feedback present in the project.
By double-clicking on the feedback, the corresponding blocks are marked.



By this you can see the network and find the feedback point.
In newer layouts you should always work with the FEEDBACKBREAKER block.

If you want to eliminate a feedback in older large layouts, it may be difficult to find the feedback point. If you know your program, you may know where feedback is located and then simply eliminate it. But if you have no direct knowledge, you have to sift through the program.

Here it is advisable to use the PROGRAM OVERVIEW window.

Method:

- mark feedback blocks (by double-clicking on the feedback in the feedback directory)

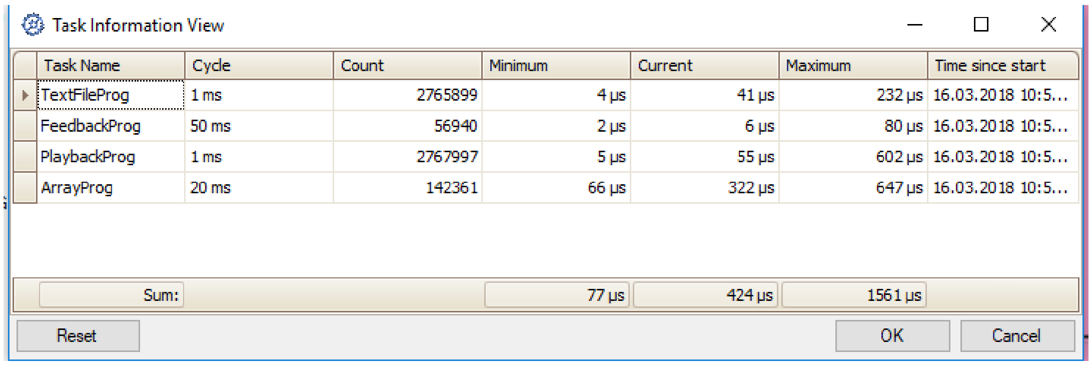
- Now you can navigate in the overview window and search for the blocks. By means of the markings you can also determine which is the last building block.

- You can also jump to the individual blocks of the feedback by double-clicking them. If you look at the overview window, you might see which function block is at the bottom of the layout. Often the feedback has an output at its end.

- You can also select a line while pressing the ALT key. This selects the complete connection, even beyond IPCs. By scrolling through the overview window, you can see if this line reappears at the top of the layout. This can help finding a feedback.

4 Diagnosing the task evaluation order

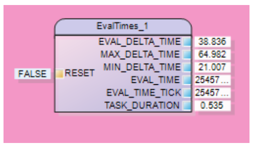
So far, the following tools / displays have been available for the diagnosis of tasks and their execution:
- The utilization ratio / evaluation time display per task in the task header
- The overview of all tasks in the task information (VIEW TASK INFORMATION)



Task Information View

Task Name	Cycle	Count	Minimum	Current	Maximum	Time since start
TextFileProg	1 ms	2765899	4 µs	41 µs	232 µs	16.03.2018 10:5...
FeedbackProg	50 ms	56940	2 µs	6 µs	80 µs	16.03.2018 10:5...
PlaybackProg	1 ms	2767997	5 µs	55 µs	602 µs	16.03.2018 10:5...
ArrayProg	20 ms	142361	66 µs	322 µs	647 µs	16.03.2018 10:5...
Sum:			77 µs	424 µs	1561 µs	

- Use of the EVALTIMES function block

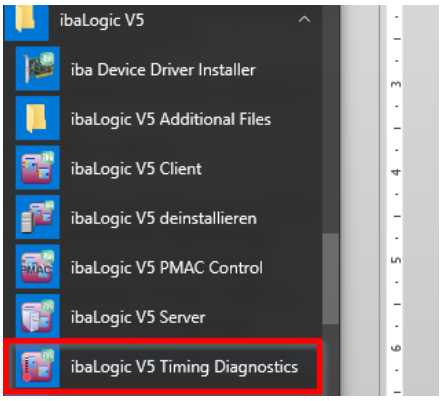


Note: The EVALTIMES module is also suitable for measuring transit times within a program. If you place two EVALTIMES blocks into the data flow, you can determine the duration of all intermediate blocks by the difference of the EVAL_TIME outputs.

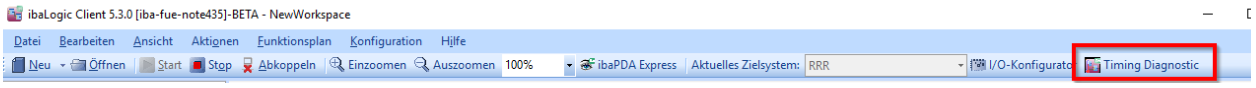
With ibaLogic 5.3.0 you now have an even more accurate diagnostic option. There is the ibaLogic V5 timing diagnostic tool.

This tool is licensed and requires a dongle activation.

The tool can be called from the Windows start menu (if ibaLogic has been installed on the computer):


ibaLogic V5
iba Device Driver Installer
ibaLogic V5 Additional Files
ibaLogic V5 Client
ibaLogic V5 deinstallieren
ibaLogic V5 PMAC Control
ibaLogic V5 Server
ibaLogic V5 Timing Diagnostics

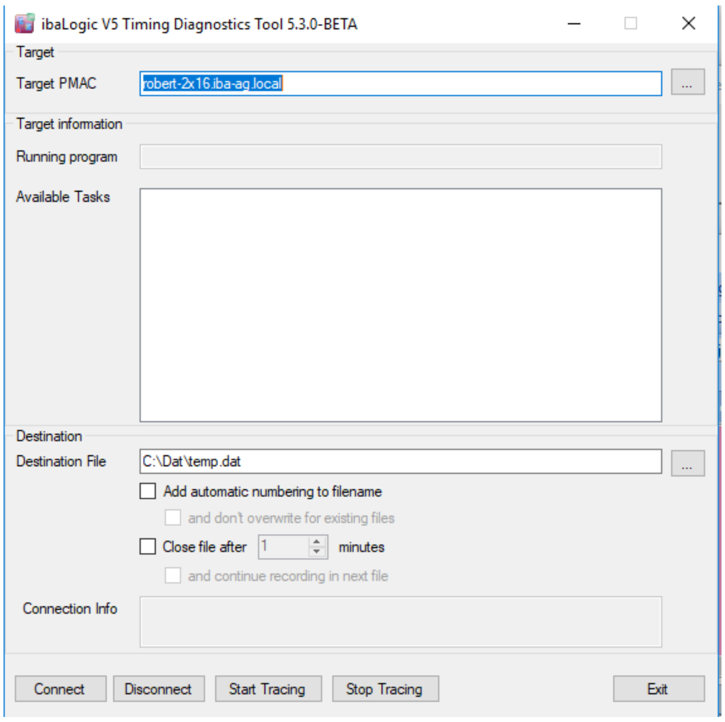
Or directly from an ibaLogic client.


ibaLogic Client 5.3.0 [iba-fue-note435]-BETA - NewWorkspace
Datei Bearbeiten Ansicht Aktionen Funktionsplan Konfiguration Hilfe
Neu Öffnen Start Stop Abkoppeln Einzoomen Auszoomen 100% ibaPDA Express Aktuelles Zielsystem: RRR I/O-Konfigurato Timing Diagnostic

This tool writes a dat file with all task state changes.

Method:

- Start tool


ibaLogic V5 Timing Diagnostics Tool 5.3.0-BETA

Target

Target PMAC: robert-2x16.iba-ag.local

Target information

Running program:

Available Tasks:

Destination

Destination File: C:\Dat\temp.dat

☐ Add automatic numbering to filename

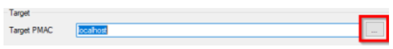
☐ and don't overwrite for existing files

☐ Close file after 1 minutes

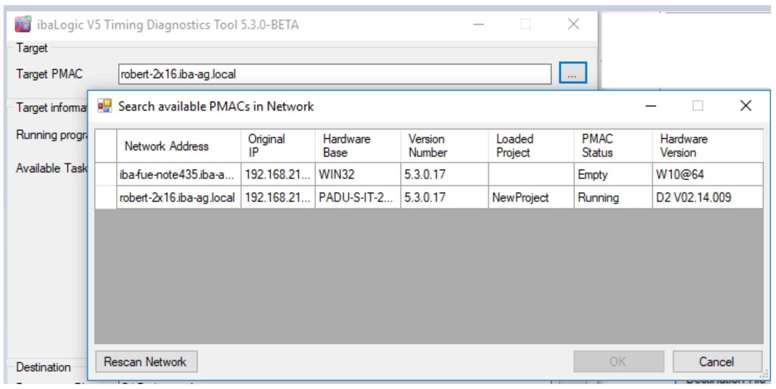
☐ and continue recording in next file

Connection Info:

- Connect to the corresponding PMAC. When starting from ibaLogic, the current target PMAC is preselected. Alternatively, the network search can be used by this button.


Target

Target PMAC: localhost


ibaLogic V5 Timing Diagnostics Tool 5.3.0-BETA

Target

Target PMAC: robert-2x16.iba-ag.local

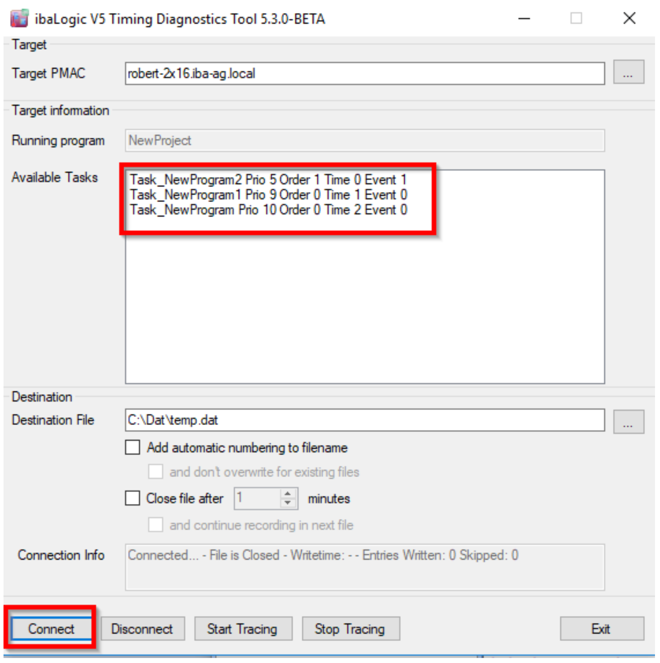
Search available PMACs in Network

	Network Address	Original IP	Hardware Base	Version Number	Loaded Project	PMAC Status	Hardware Version
	iba-fue-note435.iba-a...	192.168.21...	WIN32	5.3.0.17		Empty	W10@64
	robert-2x16.iba-ag.local	192.168.21...	PADU-S-IT-2...	5.3.0.17	NewProject	Running	D2 V02.14.009

- Enter a file name for the data file or via the folder selection. Select path and file.

- Use the options to specify the storage of the data files.

- You can now connect to a running PMAC by **CONNECT**. The tasks found are displayed in the task overview.



Target PMAC: robert-2x16.iba-ag.local

Running program: NewProject

Available Tasks:

```
Task_NewProgram2 Prio 5 Order 1 Time 0 Event 1
Task_NewProgram1 Prio 9 Order 0 Time 1 Event 0
Task_NewProgram Prio 10 Order 0 Time 2 Event 0
```

Destination File: C:\Dat\temp.dat

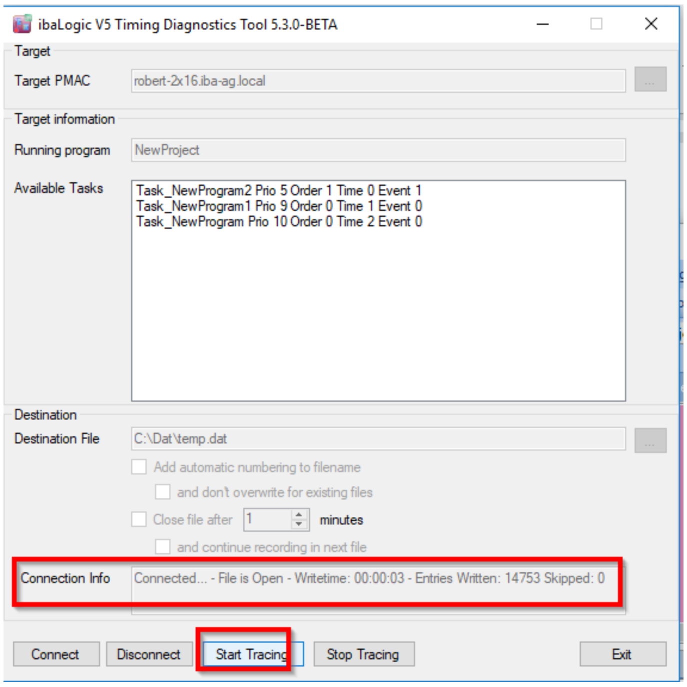
Close file after: 1 minutes

Connection Info: Connected... - File is Closed - Writetime: - - Entries Written: 0 Skipped: 0

- **Disconnect** disconnects the connection to the PMAC

- A recording is started with **START TRACING**.

Note: If no PMAC is yet connected, START TRACING automatically performes a CONNECT.



Target PMAC: robert-2x16.iba-ag.local

Running program: NewProject

Available Tasks:

```
Task_NewProgram2 Prio 5 Order 1 Time 0 Event 1
Task_NewProgram1 Prio 9 Order 0 Time 1 Event 0
Task_NewProgram Prio 10 Order 0 Time 2 Event 0
```

Destination File: C:\Dat\temp.dat

Close file after: 1 minutes

Connection Info: Connected... - File is Open - Writetime: 00:00:03 - Entries Written: 14753 Skipped: 0

The CONNECTION INFO shows if data is recorded. The data counter WRITTEN is constantly increasing. The parameter WRITETIME shows the current duration of the active dat file.

- **STOP TRACING** stops the recording and the written dat file can be viewed with ibaAnalyzer

5 View Task Diagnosis dat file in ibaAnalyzer

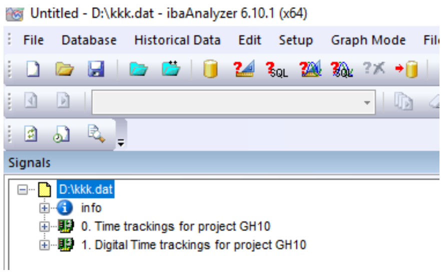
Each task can have three states:

State	Analog value	Digital value
pause	0	0
running	1	1
interrupted	-1	1

These values are recorded in 1 µs accuracy in the data file.

If all tasks have the same priority they cannot be interrupted. For this case there is also the recording as a digital value. This makes it easier to evaluate the runtimes by double-clicking etc.

When opening the data file in ibaAnalyzer you will get the following view:



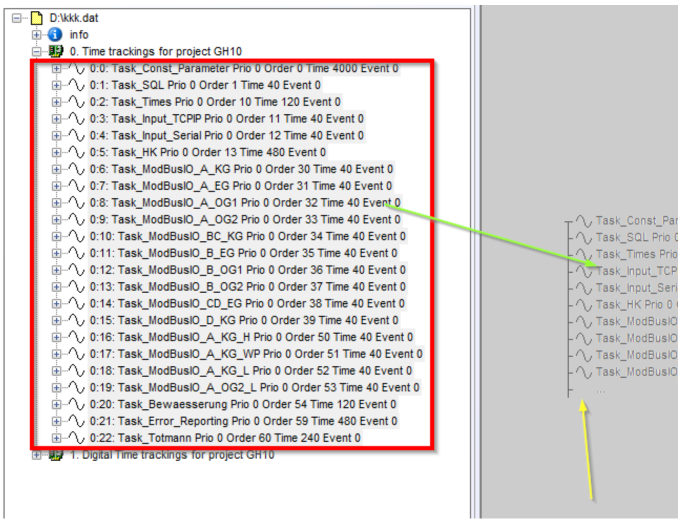
D:\kkk.dat
info
0. Time trackings for project GH10
1. Digital Time trackings for project GH10

You can see the analog and the digital tree.

Display the analog values

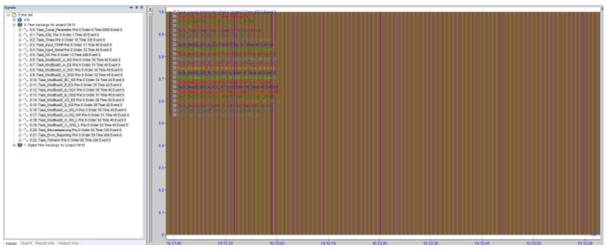
For these values the following procedure is recommended:

Mark all signals and drag them into the view via Darg & Drop (green arrow). Before releasing, press the SHIFT key and keep it pressed. By this, all signals are placed into the graph with the same scale.



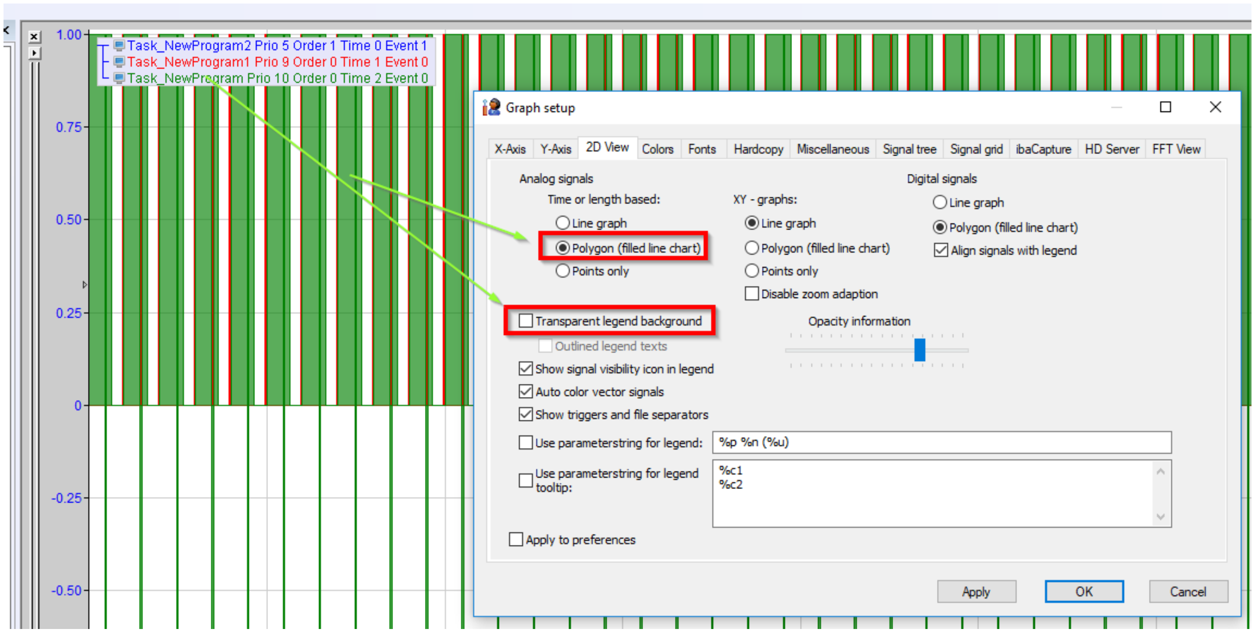
D:\kkk.dat
info
0. Time trackings for project GH10
0:0: Task_Const_Parameter Prio 0 Order 0 Time 4000 Event 0
0:1: Task_SQL Prio 0 Order 1 Time 40 Event 0
0:2: Task_Times Prio 0 Order 10 Time 120 Event 0
0:3: Task_Input_TCPIP Prio 0 Order 11 Time 40 Event 0
0:4: Task_Input_Serial Prio 0 Order 12 Time 40 Event 0
0:5: Task_HK Prio 0 Order 13 Time 480 Event 0
0:6: Task_ModBusIO_A_KG Prio 0 Order 30 Time 40 Event 0
0:7: Task_ModBusIO_A_EG Prio 0 Order 31 Time 40 Event 0
0:8: Task_ModBusIO_A_OG1 Prio 0 Order 32 Time 40 Event 0
0:9: Task_ModBusIO_A_OG2 Prio 0 Order 33 Time 40 Event 0
0:10: Task_ModBusIO_BC_KG Prio 0 Order 34 Time 40 Event 0
0:11: Task_ModBusIO_B_EG Prio 0 Order 35 Time 40 Event 0
0:12: Task_ModBusIO_B_OG1 Prio 0 Order 36 Time 40 Event 0
0:13: Task_ModBusIO_B_OG2 Prio 0 Order 37 Time 40 Event 0
0:14: Task_ModBusIO_CD_EG Prio 0 Order 38 Time 40 Event 0
0:15: Task_ModBusIO_D_KG Prio 0 Order 39 Time 40 Event 0
0:16: Task_ModBusIO_A_KG_H Prio 0 Order 50 Time 40 Event 0
0:17: Task_ModBusIO_A_KG_WP Prio 0 Order 51 Time 40 Event 0
0:18: Task_ModBusIO_A_KG_L Prio 0 Order 52 Time 40 Event 0
0:19: Task_ModBusIO_A_OG2_L Prio 0 Order 53 Time 40 Event 0
0:20: Task_Bewaesserung Prio 0 Order 54 Time 120 Event 0
0:21: Task_Error_Reporting Prio 0 Order 59 Time 480 Event 0
0:22: Task_Totmann Prio 0 Order 60 Time 240 Event 0
1. Digital Time trackings for project GH10

The result looks like this.



For a better overview you should now change some settings for the graph:

With the right mouse button you get the SETUP menu.



Task_NewProgram2 Prio 5 Order 1 Time 0 Event 1
Task_NewProgram1 Prio 9 Order 0 Time 1 Event 0
Task_NewProgram Prio 10 Order 0 Time 2 Event 0

Graph setup

X-Axis | Y-Axis | 2D View | Colors | Fonts | Hardcopy | Miscellaneous | Signal tree | Signal grid | ibaCapture | HD Server | FFT View

Analog signals

Time or length based:
- ( ) Line graph
- (●) Polygon (filled line chart)
- ( ) Points only

XY - graphs:
- (●) Line graph
- ( ) Polygon (filled line chart)
- ( ) Points only
- ☐ Disable zoom adaption

Digital signals
- ( ) Line graph
- (●) Polygon (filled line chart)
- ☑ Align signals with legend

☐ Transparent legend background
☐ Outlined legend texts
☑ Show signal visibility icon in legend
☑ Auto color vector signals
☑ Show triggers and file separators

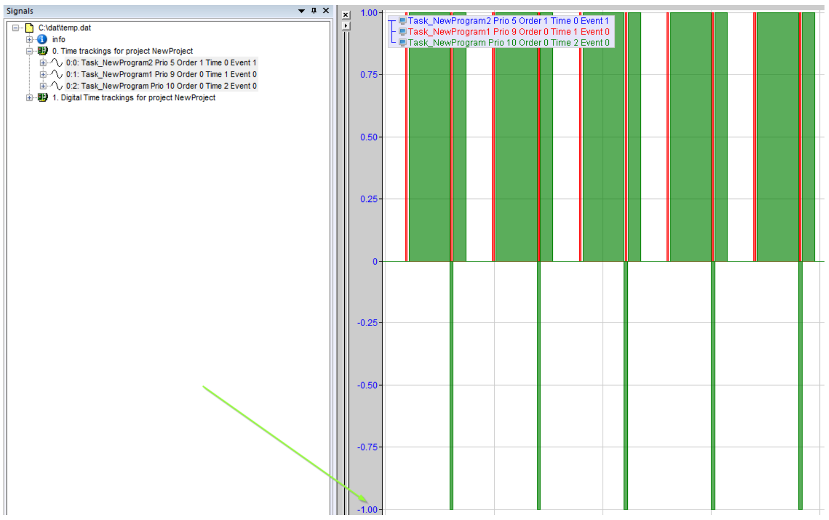
Opacity information

☐ Use parameterstring for legend: %p %n (%u)
☐ Use parameterstring for legend tooltip: %c1 %c2

☐ Apply to preferences

Apply | OK | Cancel

Important is the “polygon” setting. You get this view when you zoom in.



Signals
C:\dat\temp.dat
info
0. Time trackings for project NewProject
0:0: Task_NewProgram2 Prio 5 Order 1 Time 0 Event 1
0:1: Task_NewProgram1 Prio 9 Order 0 Time 1 Event 0
0:2: Task_NewProgram Prio 10 Order 0 Time 2 Event 0
1. Digital Time trackings for project NewProject

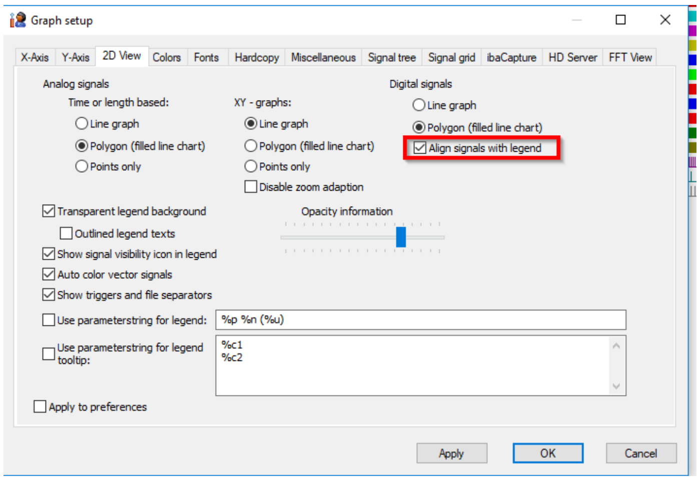
Task_NewProgram2 Prio 5 Order 1 Time 0 Event 1
Task_NewProgram1 Prio 9 Order 0 Time 1 Event 0
Task_NewProgram Prio 10 Order 0 Time 2 Event 0

Each task is now visible with its evaluation duration and its time of evaluation in relation to all other tasks. Values of -1 indicate that tasks have been interrupted here due to their priority. If there are no interruptions, the digital display is easier to handle. (See next chapter)

Display of digital values

If you only use one priority in all tasks, you can also use the digital values because there is no interruption.

With drag & drop and SHIFT key place the signals into the view.



Graph setup

X-Axis | Y-Axis | 2D View | Colors | Fonts | Hardcopy | Miscellaneous | Signal tree | Signal grid | ibaCapture | HD Server | FFT View

Analog signals

Time or length based:
- ○ Line graph
- ● Polygon (filled line chart)
- ○ Points only

XY - graphs:
- ● Line graph
- ○ Polygon (filled line chart)
- ○ Points only
- ☐ Disable zoom adaption

Digital signals
- ○ Line graph
- ● Polygon (filled line chart)
- ☑ Align signals with legend

☑ Transparent legend background

☐ Outlined legend texts

☑ Show signal visibility icon in legend

☑ Auto color vector signals

☑ Show triggers and file separators

Opacity information

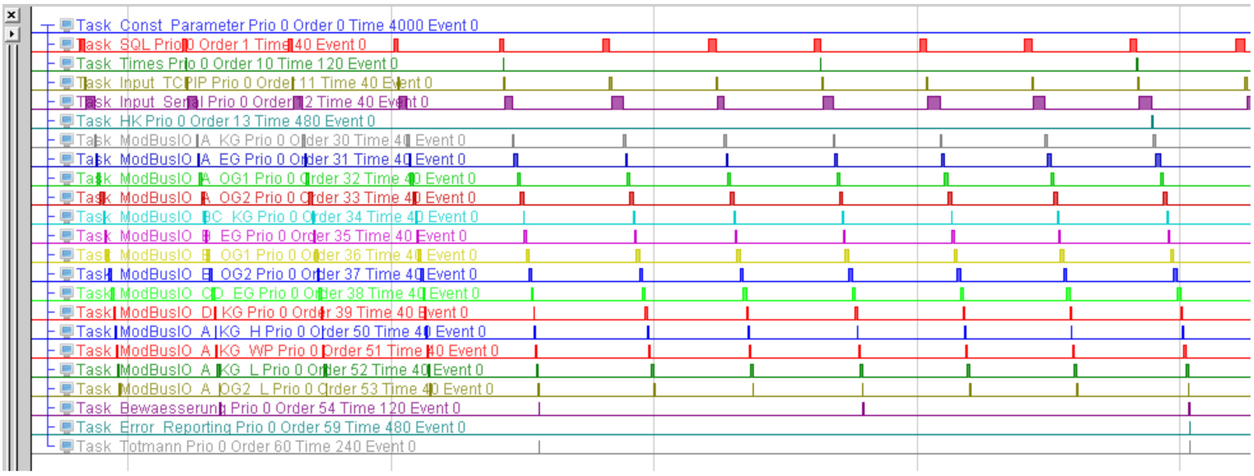
☐ Use parameterstring for legend: %p %n (%u)

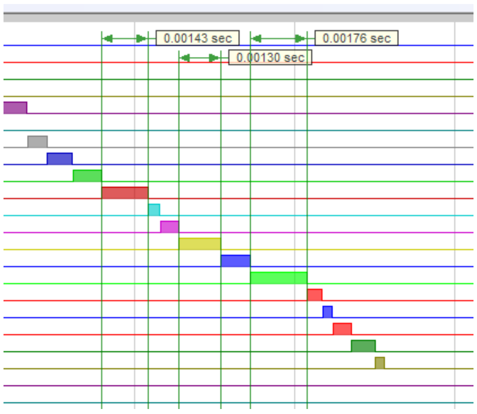
☐ Use parameterstring for legend tooltip: %c1 %c2

☐ Apply to preferences

Apply | OK | Cancel

Set the signals to align with legend and you get this view:





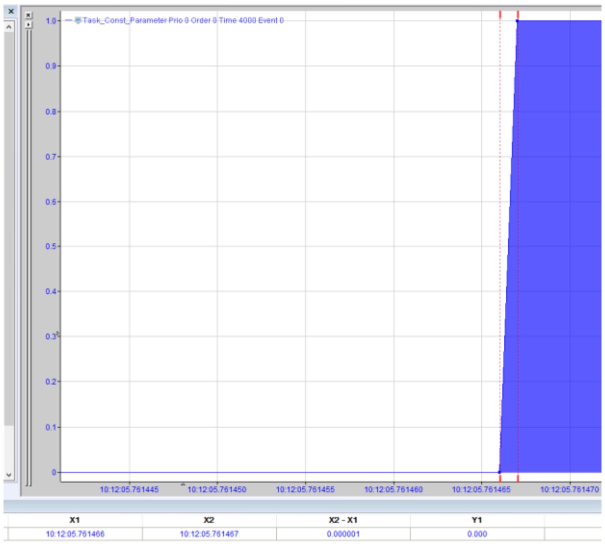
Double-clicking on the respective high state displays the runtime directly. Double-clicking on low state can display the distances.

Note: The recording is done with non-equidistant values. Only the status changes are recorded. Therefore, the signals are not suitable to be used in formulas in ibaAnalyzer, as this can only process equidistant signal values.

However, if it is necessary, you can create equidistant signals by the function RESAMPLE.

Note: Normally, a change of state would make the analog signal look like drawing a slanting line from the last state to the new state. This would make an analysis of the values cumbersome. Therefore, two values are always recorded for each status change.

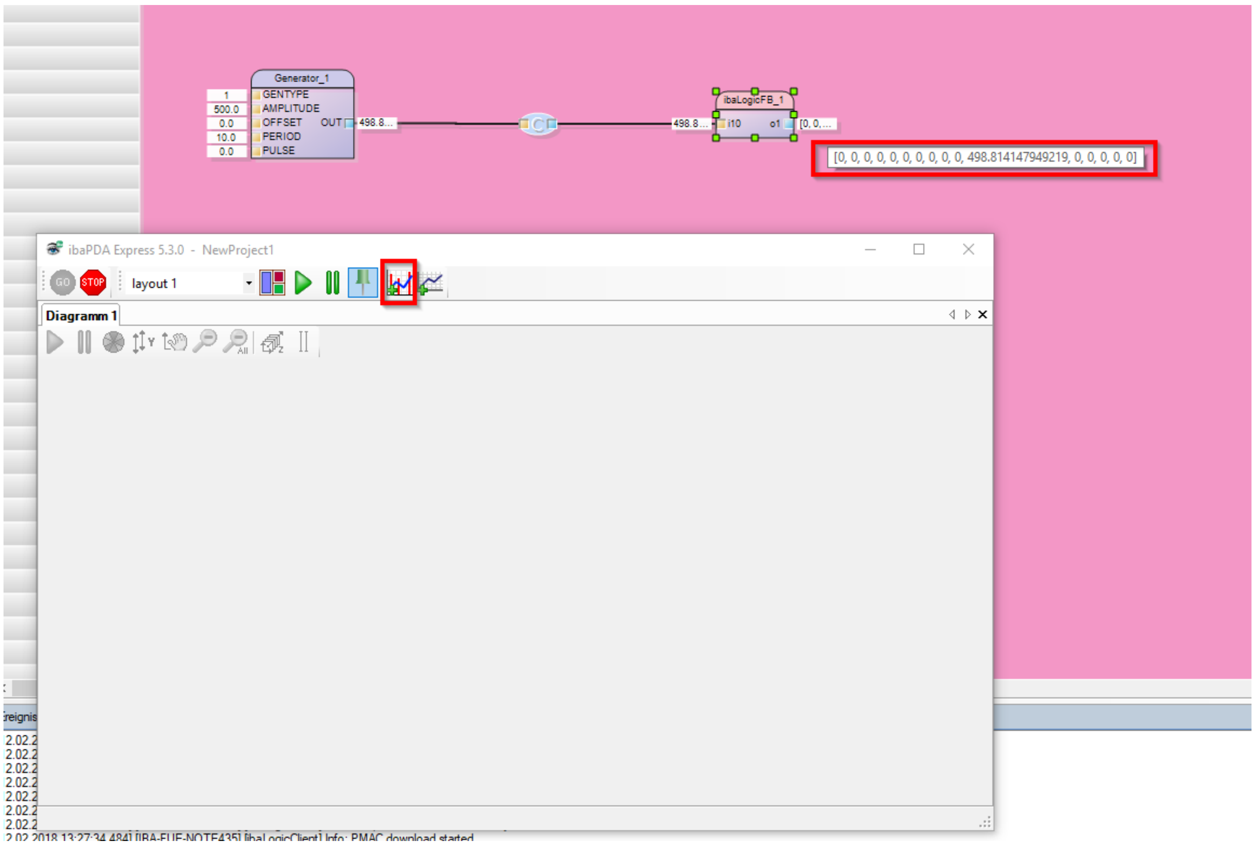
1 µs before each new state the previous state is repeated. This results in a steep slope that you see only when zooming in extremely. But for the normal view, the analog states can be clearly represented.



X1	X2	X2 - X1	Y1
10:12:05.761466	10:12:05.761467	0.000001	0.000

6 Display of arrays in ibaPDAExpress

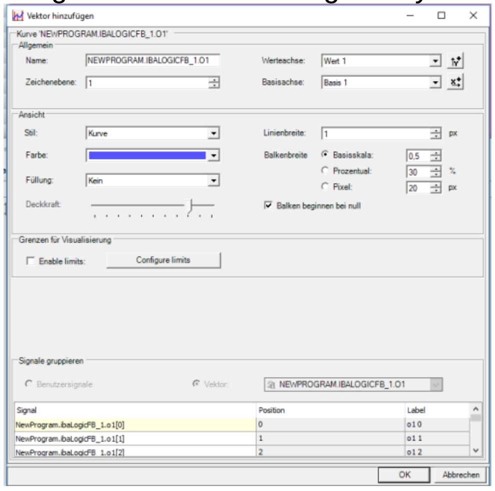
The display of arrays in ibaLogic was previously only possible by using a special function block, which separates the array into corresponding outputs.
To make array values easier to read, you can now drag arrays into ibaPDAExpress.
The tooltip of an array also displays the current values, but it is very confusing for larger arrays.
In the ibaPDAExpress you can now open a DIAGRAM as an array display.



```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 498.814147949219, 0, 0, 0, 0, 0]
```

```
2.02.2018 13:27:34.484] [IBA-FUE-NOTE435] [ibaLogicClient] Info: PMAC download started
```

Drag the value to the diagram by holding down the ALT key.



Vektor hinzufügen

Kurve 'NEWPROGRAM.IBALOGICFB_1.O1'

Allgemein

Name:	NEWPROGRAM.IBALOGICFB_1.O1	Werteachse:	Wert 1
Zeichenebene:	1	Basisachse:	Basis 1

Ansicht

Stil:	Kurve	Linienbreite:	1 px
Farbe:	🟦	Balkenbreite	[✓ Basisskala:] 0,5
Füllung:	Kein		Prozentual: 30 %
Deckkraft			Pixel: 20 px

[✓] Balken beginnen bei null

Grenzen für Visualisierung

☐ Enable limits: Configure limits

Signale gruppieren

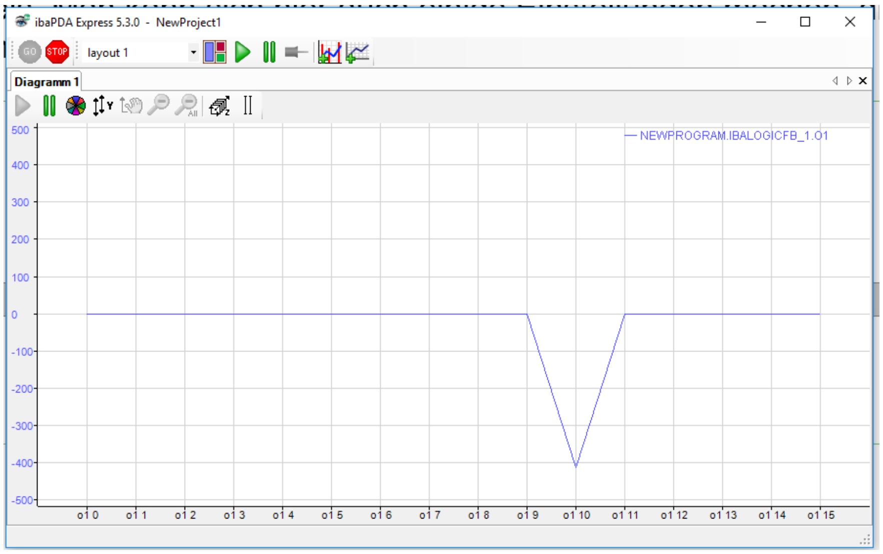
Benutzersignale [✓ Vektor:] NEWPROGRAM.IBALOGICFB_1.O1

Signal	Position	Label
NewProgram.ibaLogicFB_1.o1[0]	0	o1 0
NewProgram.ibaLogicFB_1.o1[1]	1	o1 1
NewProgram.ibaLogicFB_1.o1[2]	2	o1 2

OK Abbrechen

This dialog can be just confirmed.

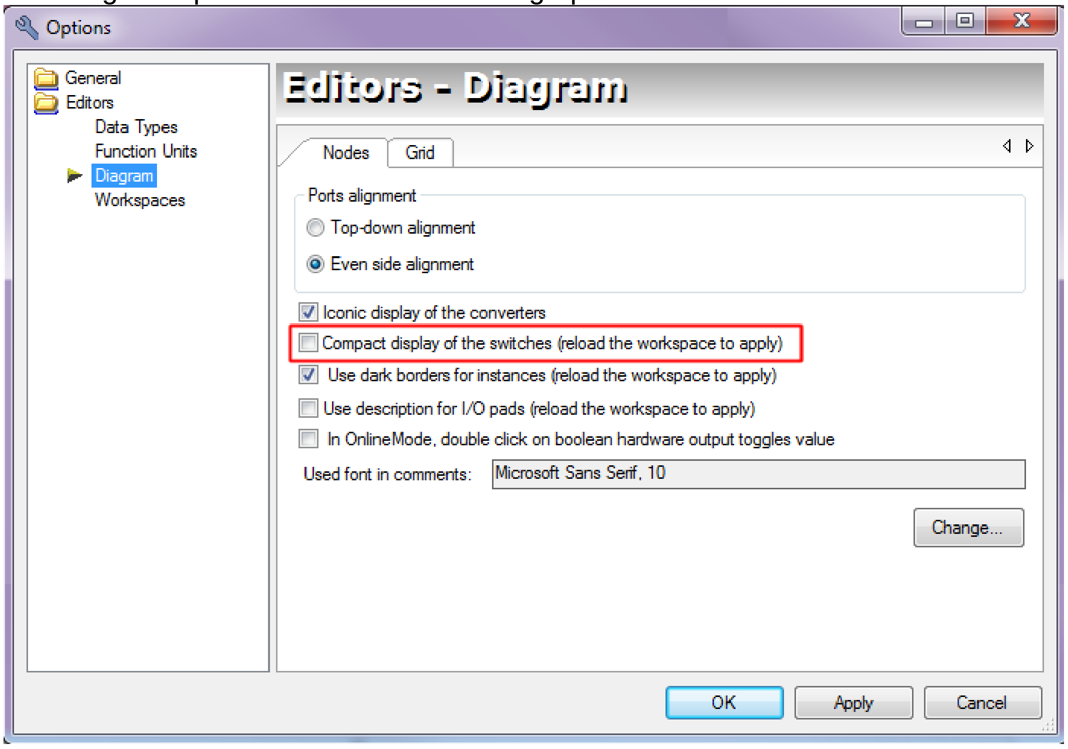
Note: You can also make some settings here, which can be found in the DIAGRAM description of ibaQPanel.



Now you get a representation of the array. Starting on the left side with the lowest index and on the right side with the highest array index (here: 15).
Zooming and other known practices are available.

7 Switch as a compact symbol

It is now possible to display the switch as a compact symbol.
The diagram option contains the following option:

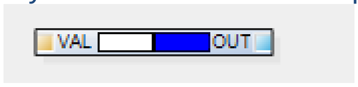


Ports alignment
Top-down alignment [✓ Even side alignment]

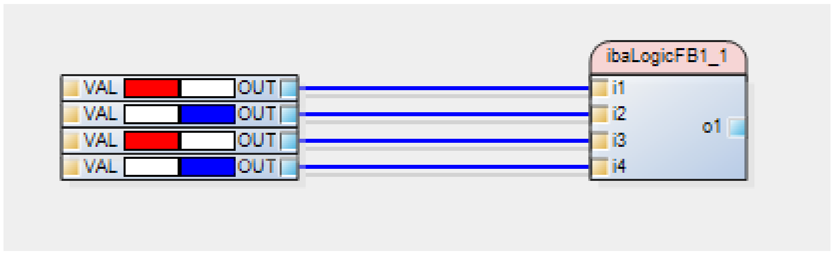
[✓] Iconic display of the converters
[ ] Compact display of the switches (reload the workspace to apply)
[✓] Use dark borders for instances (reload the workspace to apply)
[ ] Use description for I/O pads (reload the workspace to apply)
[ ] In OnlineMode, double click on boolean hardware output toggles value

Used font in comments: Microsoft Sans Serif, 10

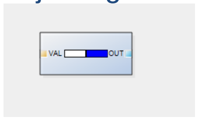
If you have activated this option, the switches are now displayed as follows:



Keep in mind that the inner area is the switching area. To move the switch therefore select it with the mouse in the area of the inscription VAL / OUT .
The compact representation now allows to place the switches opposite to the connectors of a block.



To change existing switches to compact display, the workspace must be reopened once the option has been set.
The "old switches" are still displayed in their size unchanged and must be corrected manually by adjusting the height.



8 Extended playback

So far it was possible to use a playback file as data source.

The extended playback can now:

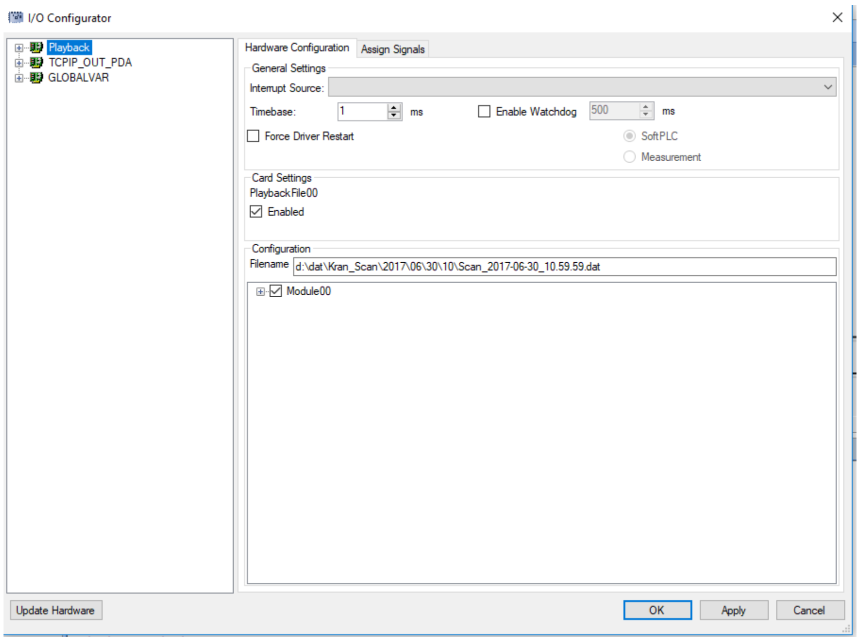
- Play various playback files controlled. The playback files should come from the same data source so that signals are always unique.

- Control the playback file by

- Pause
- Replay (start again from the beginning)

<u>Note: When loading old projects with active playback for the first time, the I / O configurator must be opened once and "Update Hardware" must be pressed.</u>

Creating a first DatFile as before:


I/O Configurator

Playback
TCPIP_OUT_PDA
GLOBALVAR

Hardware Configuration | Assign Signals

General Settings
Interrupt Source:
Timebase: 1 ms
☐ Enable Watchdog 500 ms
☐ Force Driver Restart
● SoftPLC
○ Measurement

Card Settings
PlaybackFile00
☑ Enabled

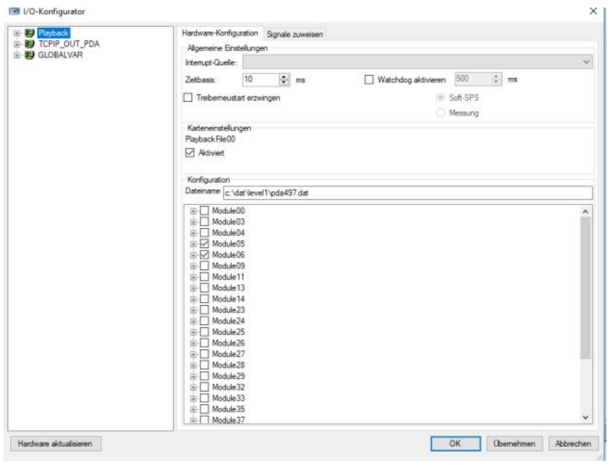
Configuration
Filename d:\dat\Kran_Scan\2017\06\30\10\Scan_2017-06-30_10.59.59.dat

☑ Module00

Update Hardware | OK | Apply | Cancel

- Enter path and filename after activating playback

- Push APPLY


I/O-Konfigurator

Playback
TCPIP_OUT_PDA
GLOBALVAR

Hardware-Konfiguration | Signale zuweisen

Allgemeine Einstellungen
Interrupt-Quelle:
Zeitbasis: 10 ms
☐ Watchdog aktivieren 500 ms
☐ Treiberneustart erzwingen
● Soft-SPS
○ Messung

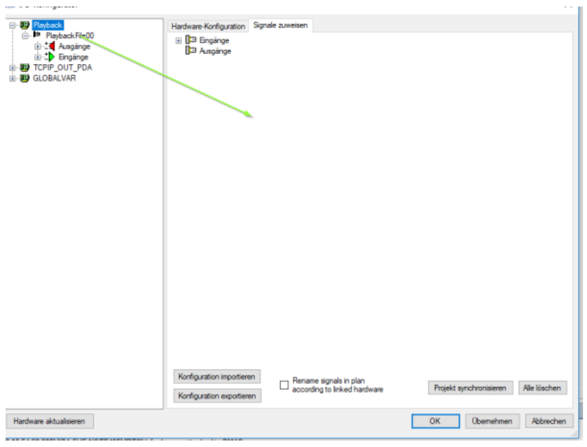
Karteneinstellungen
PlaybackFile00
☑ Aktiviert

Konfiguration
Dateiname c:\dat\level1\pda497.dat

☐ Module00
☐ Module03
☐ Module04
☑ Module05
☑ Module06
☐ Module09
☐ Module11
☐ Module13
☐ Module14
☐ Module23
☐ Module24
☐ Module25
☐ Module26
☐ Module27
☐ Module28
☐ Module29
☐ Module32
☐ Module33
☐ Module35
☐ Module37

Hardware aktualisieren | OK | Übernehmen | Abbrechen

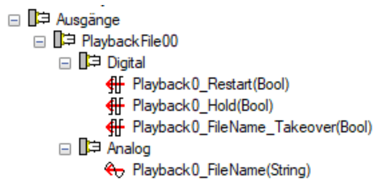
- Select modules and / or signals
- Push again APPLY



- Then make the signal assignment by drag & drop.
- Exit the dialog with OK

Now you have all signals available.

For the outputs there are:



Ausgänge
PlaybackFile00
Digital
Playback0_Restart(Bool)
Playback0_Hold(Bool)
Playback0_FileName_Takeover(Bool)
Analog
Playback0_FileName(String)

Restart: a rising edge at this output restarts the playback file

Hold: a high signal on this output to pause playing the file

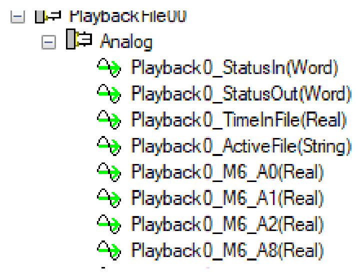
To play different playback files, the outputs **TAKEOVER** and **FILENAME** are used.

FileName to select the playback file with path and name

FileName_Takeover A rising edge takes over the file specified in FileName for playback and starts at the beginning of the file

Note: You do not need to connect these last two outputs if you only want to play the dat file as defined in the IO configuration.

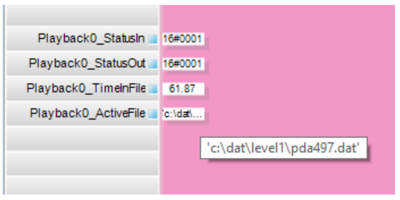
inputs:



PlaybackFile00
Analog
Playback0_StatusIn(Word)
Playback0_StatusOut(Word)
Playback0_TimeInFile(Real)
Playback0_ActiveFile(String)
Playback0_M6_A0(Real)
Playback0_M6_A1(Real)
Playback0_M6_A2(Real)
Playback0_M6_A8(Real)

The inputs indicate the status for the inputs / outputs.

They should be 1, everything else indicates an error (e.g. file not found).



Playback0_StatusIn	16#0001
Playback0_StatusOut	16#0001
Playback0_TimeInFile	61.87
Playback0_ActiveFile	'c:\dat...

'c:\dat\level1\pda497.dat'

TimeInFile displays the current time in the playback file.

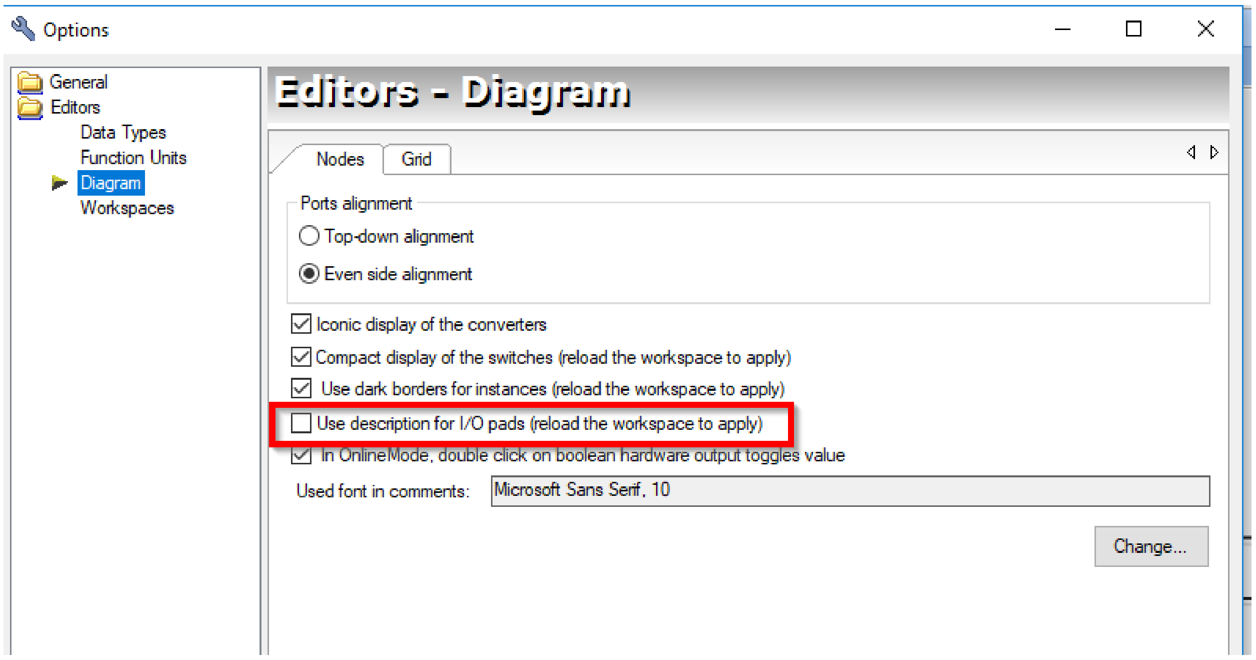
ActiveFile displays the path and name of the currently playing playback file as feedback

9 Optional display signal name or signal description

Previously, it was only possible to display the signal name from the IO configuration for the IO signals in the plan. For playback files or other configuration, the actual signal name is hidden in the signal description and so far was only visible in the tooltip.

The reason was that signal names must be IEC compliant and certain signal import or playback signal names do not meet this standard. Therefore, the signal name is generated generically and the actual signal name is moved into the description.

It is now possible in the options to display the original name instead of the signal name. This only affects the display in the plan, for evaluation or display in PdaExpress still the IEC compliant signal name is used.



Ports alignment
- ○ Top-down alignment
- ● Even side alignment

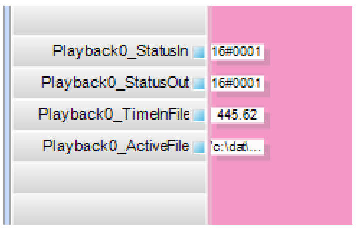
- ☑ Iconic display of the converters
- ☑ Compact display of the switches (reload the workspace to apply)
- ☑ Use dark borders for instances (reload the workspace to apply)
- ☐ Use description for I/O pads (reload the workspace to apply)
- ☑ In OnlineMode, double click on boolean hardware output toggles value

Used font in comments: Microsoft Sans Serif, 10

If necessary, reload the workspace for the option to take effect after the changeover.

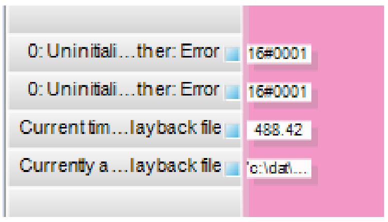
Example:

Option not selected:



Playback0_StatusIn	16#0001
Playback0_StatusOut	16#0001
Playback0_TimeInFile	445.62
Playback0_ActiveFile	'c:\dat\...

Option selected:



0: Uninitiali…ther: Error	16#0001
0: Uninitiali…ther: Error	16#0001
Current tim…layback file	488.42
Currently a…layback file	'c:\dat\...

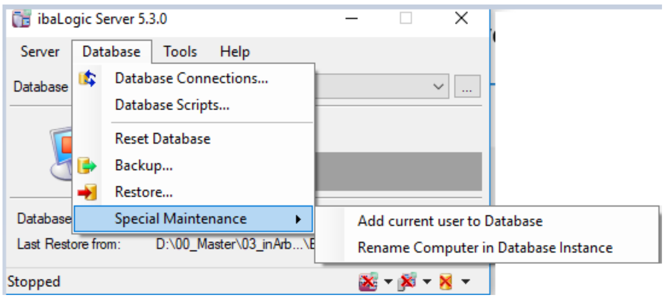
In order to represent longer names, a representation typical for Microsoft was chosen. The first and last characters are displayed, the rest is suppressed by .... Again, the tooltip shows the full name.

This representation was chosen because often at the beginning of a label the big context and at the other end the detail can be found. In between are often identical names that can be suppressed.

10 Auxiliary functions for changing the computer name or database

If you change the computer name of your ibaLogic computer or if you are logged in with another user than the one who installed ibaLogic, then you may not be able to start ibaLogic or set it online, etc.

There are two new functions for the ibaLogic Server:



Ad current user to database: The current user is added to the database so that he has access to the workspaces.

Rename computer in database instance: If you have changed the computer name, or changed the workgroup or domain, you can adjust the database to the computer name

11 Positioning element

The positioning of elements can now also be done with the arrow keys.

The following applies:

- If nothing is selected, the arrows are used to move the plan

- If one or more elements are selected, then these will be moved with the arrow keys.

If you additionally hold down the CTRL key, movement is done with a fine resolution.

This facilitates and accelerates accurate positioning of the components, for example, to get straight connecting lines.

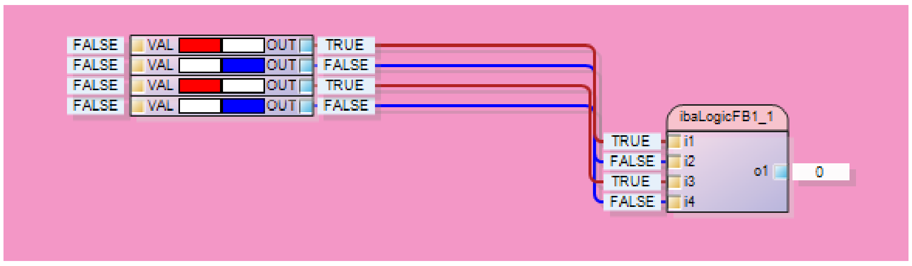
Connection nodes can also be moved in this way.

12 Improved routing

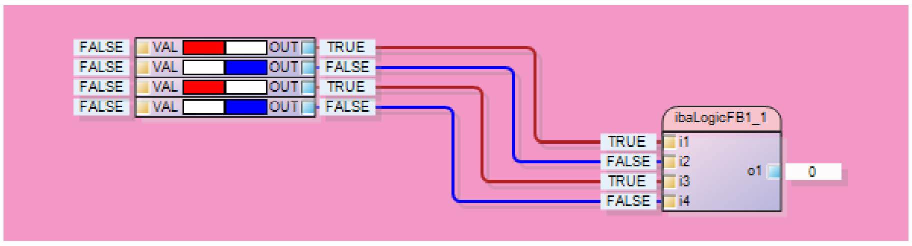
A poor feature of the routing had been that connections corrected by hand were immediately re-routed, if you move a block only a tiny bit. An improvement has now been implemented:
As long as you do not move the block beyond the manually placed corner point, the lines remain as they were placed before.

Example:

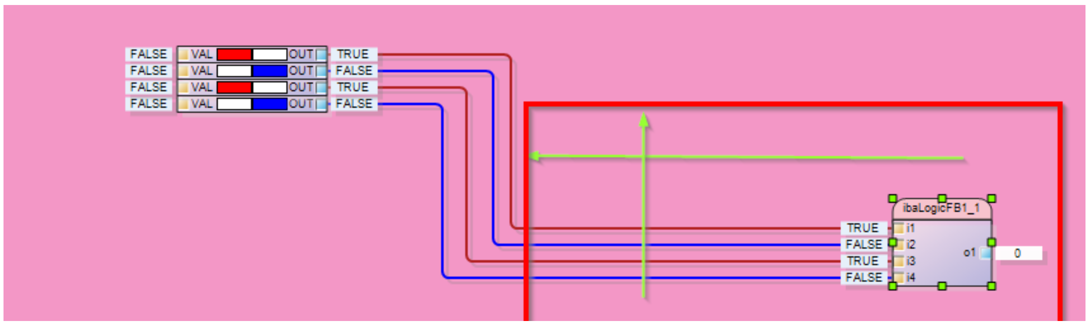
This routing is given by auto-routing:



Each line is now corrected manually:



Now you can move the block within the marked frame without having to re-route lines.



Only moving the block horizontally further to the left or vertically higher than the corner points will re-route all the lines.

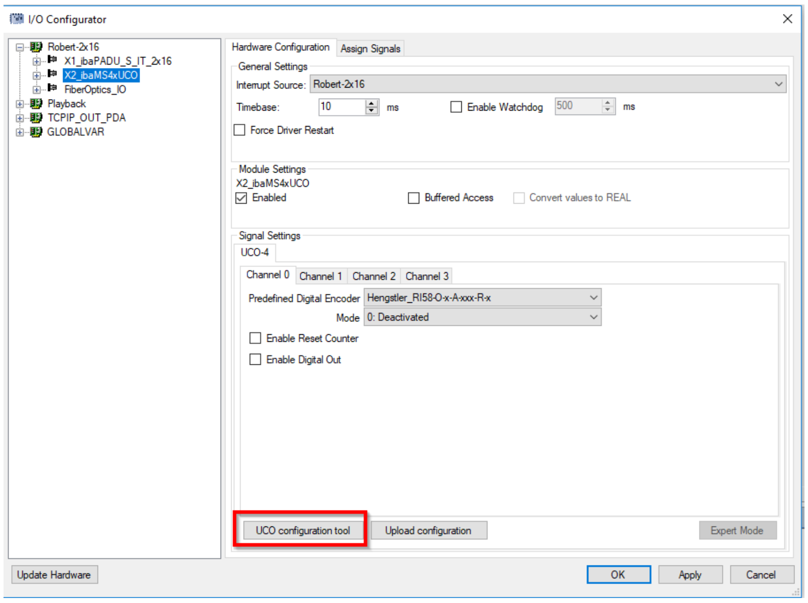
This facilitates the fixation of lines, even if it is not yet optimal.

13 Hint on operation with ATL / CTRL etc

If you press the CTRL / ALT / SHIFT key, a hint will appear in the bottom line indicating which options these keys offer.

14 UCO Tool

For the ibaMS4xUCO module, the configuration tool known from ibaPDA has been integrated.

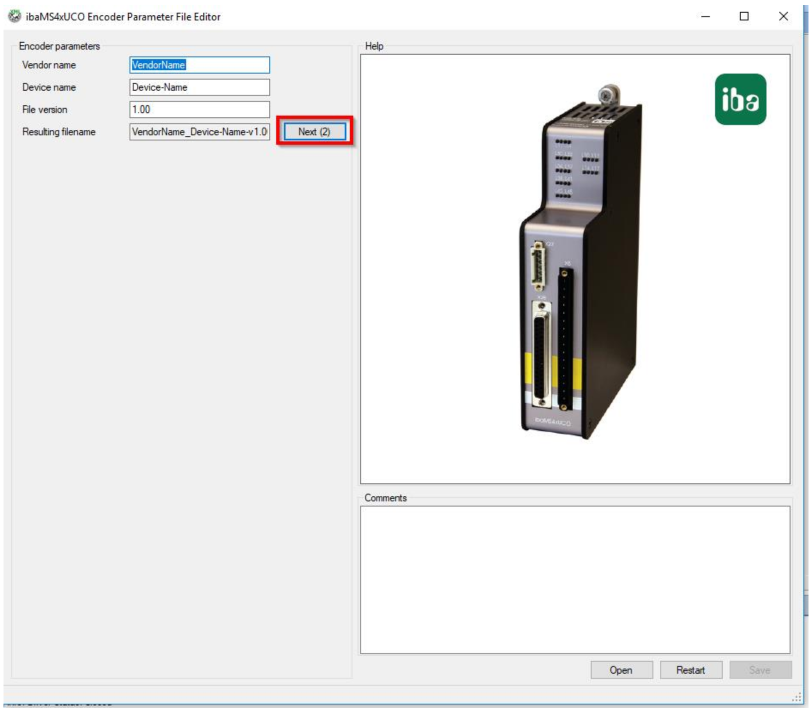


Interrupt Source: Robert-2x16
Timebase: 10 ms
☐ Enable Watchdog 500 ms
☐ Force Driver Restart

X2_ibaMS4xUCO
☑ Enabled
☐ Buffered Access
☐ Convert values to REAL

Predefined Digital Encoder Hengstler_RI58-O-x-A-xxx-R-x
Mode 0: Deactivated
☐ Enable Reset Counter
☐ Enable Digital Out

With this you have an assistant for creating UCO settings. With NEXT you progress through the configuration.

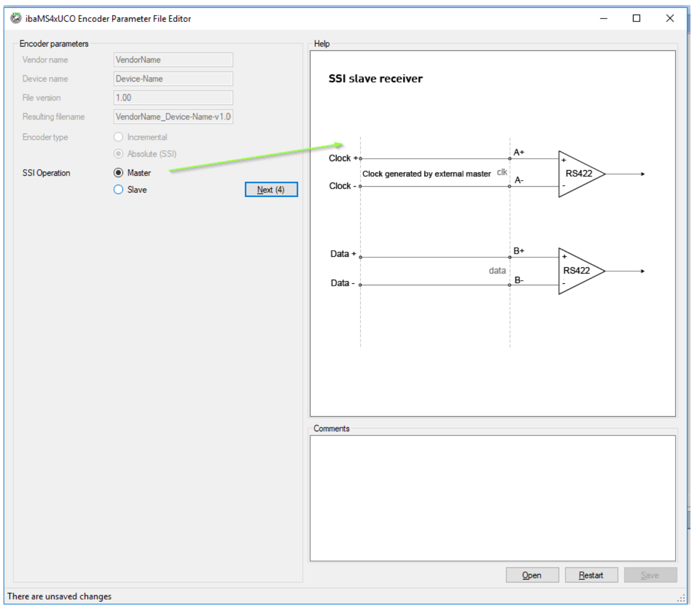


ibaMS4xUCO Encoder Parameter File Editor

Encoder parameters

Vendor name	VendorName
Device name	Device-Name
File version	1.00
Resulting filename	VendorName_Device-Name-v1.0

An explanation will always be provided on the right side for each item.



ibaMS4xUCO Encoder Parameter File Editor

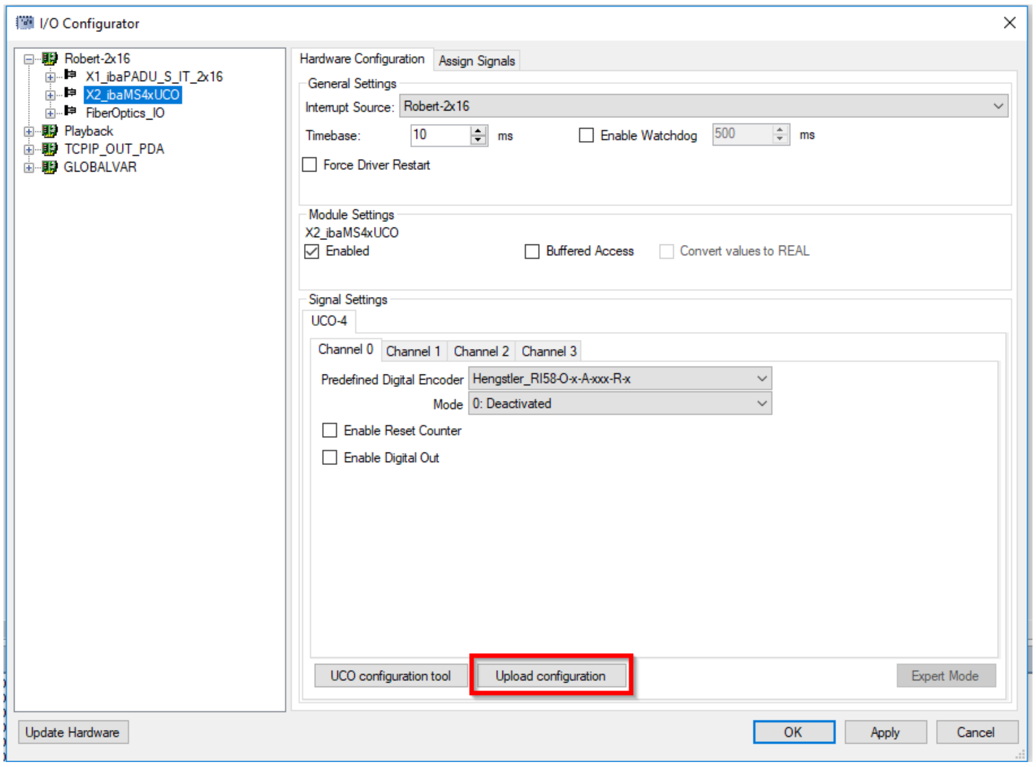
Encoder parameters

Vendor name	VendorName
Device name	Device-Name
File version	1.00
Resulting filename	VendorName_Device-Name-v1.0
Encoder type	Incremental [✓ Absolute (SSI)]
SSI Operation	[✓ Master] Slave

Help: SSI slave receiver

There are unsaved changes

If you are done with the configuration, you save it in an XML file (SAVE)



General Settings
Interrupt Source: Robert-2x16
Timebase: 10 ms
☐ Enable Watchdog 500 ms
☐ Force Driver Restart

Module Settings
X2_ibaMS4xUCO
☑ Enabled ☐ Buffered Access ☐ Convert values to REAL

Signal Settings
Predefined Digital Encoder Hengstler_RI58-O-x-A-xxx-R-x
Mode 0: Deactivated
☐ Enable Reset Counter
☐ Enable Digital Out

This XML is automatically sent to the connected PaduS-IT-2x16. If you want to use it for other devices, too, the file can be selected with UPLOAD CONFIGURATION and stored in the UCO module.

15 DAQ-S / DAQ-C / DAQ

This version of ibaLogic can be installed and started on the devices ibaDAQ-S / ibaDAQ-C and ibaDAQ. The hardware interfaces of the main module and the additional modules are NOT available to ibaLogic. They are exclusively used by ibaPDA. As usual, ibaLogic can communicate with ibaPDA or also use the TCP / IP connections of the devices.

16 IL5 name

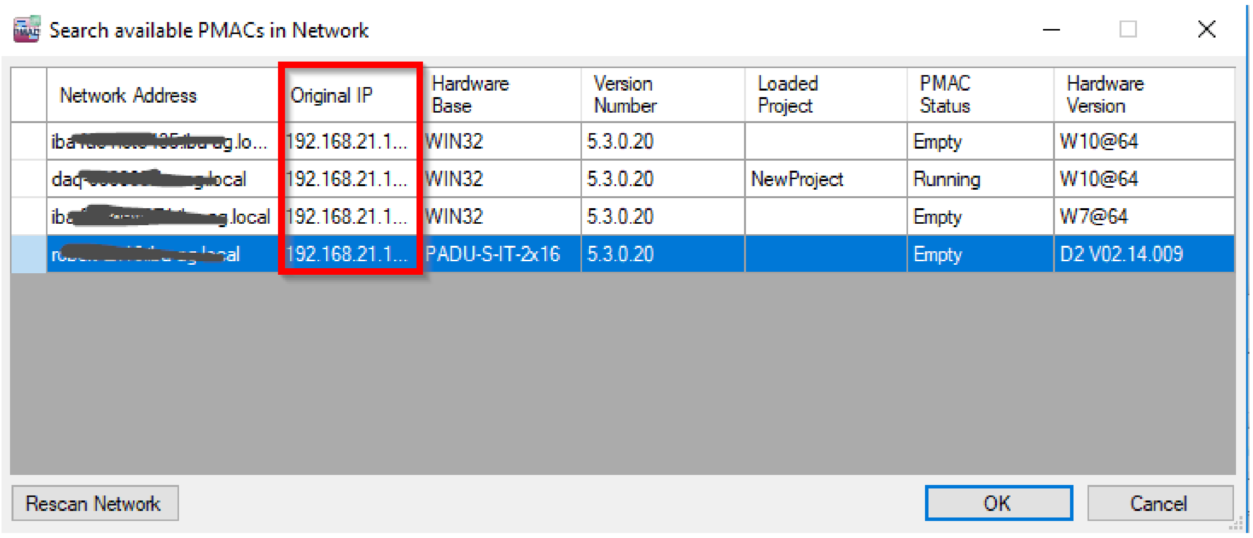
Export files now have the extension .IL5 and are thus adjusted to the ibaLogic version. Older .IL4 files are still recognized and can be imported.

17 Improved disconnect / reconnect and detection of changes

When closing the client or disconnecting the client from the PMAC and subsequently reopening or reconnecting it, sometimes the only option for available was to restart the project. Several improvements have been implemented now, significantly improving the reconnection feature.

18 PMAC search window with IP address

The search window for PMACs, which can be found in the target system configuration or the timing diagnostic tool, has been supplemented with the IP address. This allows you to manually use the IP address of the target system instead of the name to connect to it, especially in networks where the name resolution does not work correctly.

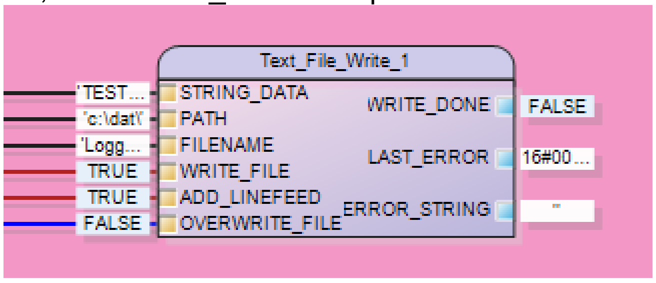

Search available PMACs in Network

Network Address	Original IP	Hardware Base	Version Number	Loaded Project	PMAC Status	Hardware Version
iba[redacted]g.lo...	192.168.21.1...	WIN32	5.3.0.20		Empty	W10@64
daq[redacted]local	192.168.21.1...	WIN32	5.3.0.20	NewProject	Running	W10@64
ib[redacted]g.local	192.168.21.1...	WIN32	5.3.0.20		Empty	W7@64
r[redacted]al	192.168.21.1...	PADU-S-IT-2x16	5.3.0.20		Empty	D2 V02.14.009

19 Text_File_Write function block

In the past, sometimes the LogFileWrite DLL was used to write text files for logging.

The Text_File_Write function block replaces the DLL. Please note that the file write is done asynchronly to the task, the "WRITE_DONE" output indicates that the last write command has successfully ended.



STRING_DATA: Ascii data to be written to the file

PATH: Path to the output file

FILENAME: Name of the output file

WRITE_FILE: TRUE = write, so with constant TRUE the string_data is written in each cycle, if the last write command was terminated successfully.

ADD_LINEFEED: A line feed is automatically appended to the STRING_DATA text. This will write each entry into a new line.

OVERWRITE_FILE: FALSE = All data will be appended to the existing content in the text file. TRUE = Only the last text will be written to the file as it will be completely overwritten

WRITE_DONE: TRUE when file is written
LAST_ERROR: Hex error code
ERROR_STRING: error message as a string