Neue Funktionen in ibaLogic v5.3.0

Author: iba AG Fürth

Date: 16/03/2018

Inhaltsverzeichnis

1 Bereinigung der Berechnungsreihenfolge

Die Berechnungsreihenfolge bei ibaLogic ergibt sich automatisch aus der Logik der verschalteten Bausteine.
So müssen z.B für einen Ausgang alle vorherigen Bausteine, beginnend beim ersten Baustein, durchgerechnet werden.

Eine Ausnahme hiervon ergibt sich bei Unklarheiten wie parallelen Zweigen oder Rückkopplungen (Feedbacks). Da hier nicht klar ist, wo der „Anfang“ der Kette besteht, wird das Prinzip angewandt, dass der erste berechnete Bausteine auf Grund seiner Position gesucht wird. In der Reihenfolge von oben nach unten (Y-Position) und von links nach rechts (X-Position).
Dies hat zur Konsequenz, dass unter Umständen eine Verschiebung von Bausteinen zu einer anderen Berechnungsreihenfolge führt.
Dies war schon immer in allen bisherigen ibaLogic Versionen so gegeben.
Probleme mit diesem Algorithmus sind in der Zeit seit ibaLogic V3 auch nicht bekannt.

Im Zuge eines Fehlers bei einer bestimmten Konstellation in Feedbacks, wurde der Algorithmus der Berechnungsreihenfolge korrigiert.
Dies führt nun in diesen Fällen zu einer anderen Berechnungsreihenfolge ab der Version 5.3.0.
Ein grundsätzlich anderes Verhalten von ibaLogic ist in der Regel nicht zu erwarten. Es könnte aber in Einzelfällen bei takt-relevanten Tricks und Berechnungen zu einem anderen Verhalten kommen.

Um das Berechnungsverhalten für den Benutzer transparenter zu machen wurden folgende Neuerungen in ibaLogic 5.3.0 realisiert. Die einzelnen Punkte sind auch in den nächsten Kapiteln im Detail erläutert:

1) Es gibt die ibaLogic Zwischen-Version 5.2.4, mit der Möglichkeit sich die Berechnungsreihenfolge als Txt-Dateien (pro Task) zu exportieren. Damit kann man den Unterschied zwischen alter und neuer Berechungsreihenfolge durch Datei-Vergleich klar sehen
2) Es gibt eine neue Ansicht der FEEDBACKs, so dass diese einfach aufgefunden werden könnnen.

   Die Problematik der Feedbacks kann aufgelöst werden. Dazu wurde ein neuer Baustein FEEDBACKBREAKER eingeführt. Er entspricht den in der ibaLogic V3 bekannten MOVE Baustein und bestimmt das Rückführ-Signal eines Feedbacks. Damit werden Feedbacks praktisch aufgelöst.
3) Um auch das generelle Verhalten der Task mit Berechungsreihenfolge und eventuellen Unterbrechungen / Verschiebungen / Laufzeiten besser transparent zu haben, wurde ein neues Tool IBALOGIC V5 TIMING DIAGNOSE eingeführt. Dieses Tool schreibt alle Task-Zustände wie LÄUFT/ RUHT / UNTERBROCHEN mit µs Auflösung als Dat-Datei mit. Damit ist das exakte Verhalten der Tasks analysierbar.

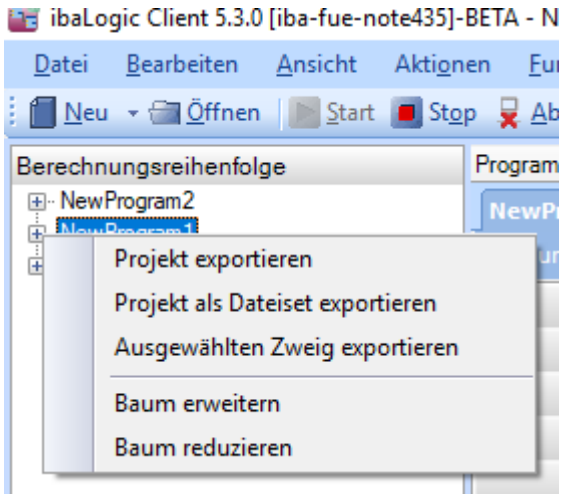
1.1 Export der Berechnungsreihenfolge

In der Ansicht Berechungsreihenfolge gibt es ein neues Kontext-Menü, welches mit der rechten Maustaste aufrufbar ist. Dort hat man die Möglichkeit die Berechungsreihenfolge als Textdatei zu exportieren. Diese Funktion ist ab der Version 5.2.4 vorhanden.

Die 5.2.4 verwendet noch den „alten“ Algorythmus für die Berechung. Daher kann man einen Export aus der 5.2.4 mit einem Export einer 5.3.0 im Datei-Vergleich untersuchen.

Man kann zwischen verschiedenen Export-Formen auswählten:

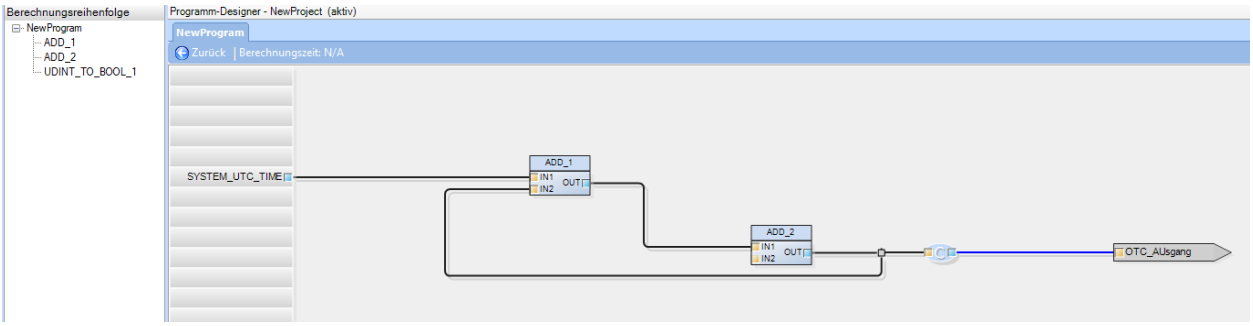
Projekt exportieren: Die gesamte Berechnungsreihenfolge in eine einzige Datei exportieren
Projekt als Dateiset exportieren: Jede einzelne Task wird in eine einzelnen Datei exportiert
Ausgewählten Zweig exportieren: Nur der anwählte Zweig wird in eine Datei exportiert. Dies ermöglicht es, geziehlt nur eine Task / einen Makro etc zu exportieren.

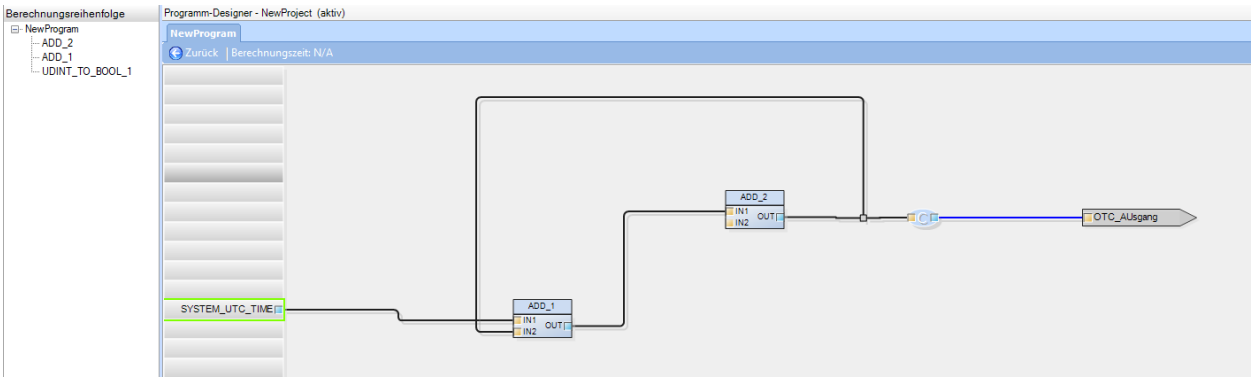


1.2 Rückkopplungen und deren Auflösung

Rückkopplungen in ibaLogic werden so behandelt, dass der Baustein, der zuerst kommt, zuerst berechnet wird.
„Zuerst“ heißt, wenn man von oben nach unten und von links nach rechts das Programm durchgeht.

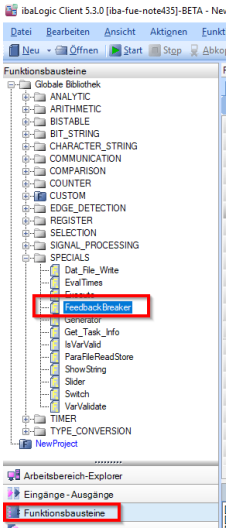
Daher ergibt sich je nach Anordung eine andere Berechungsreihenfolge(siehe links):



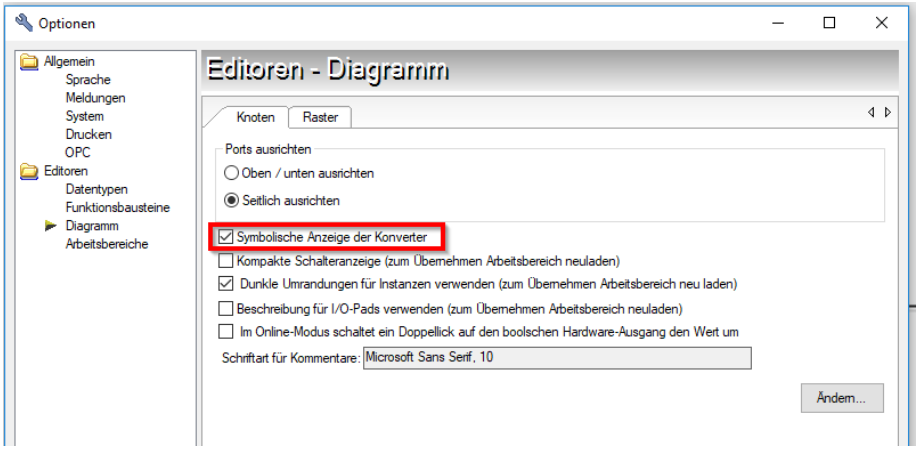


Dieses Verhalten kann nun fest definiert werden.
Dazu muss man den Punkt der Rückkopplung bestimmen.

Es gibt dazu den Baustein FEEDBACKBREAKER.

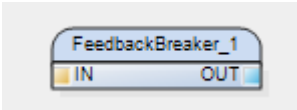


Die Darstellung des Bausteins ist an diese Option gebunden:



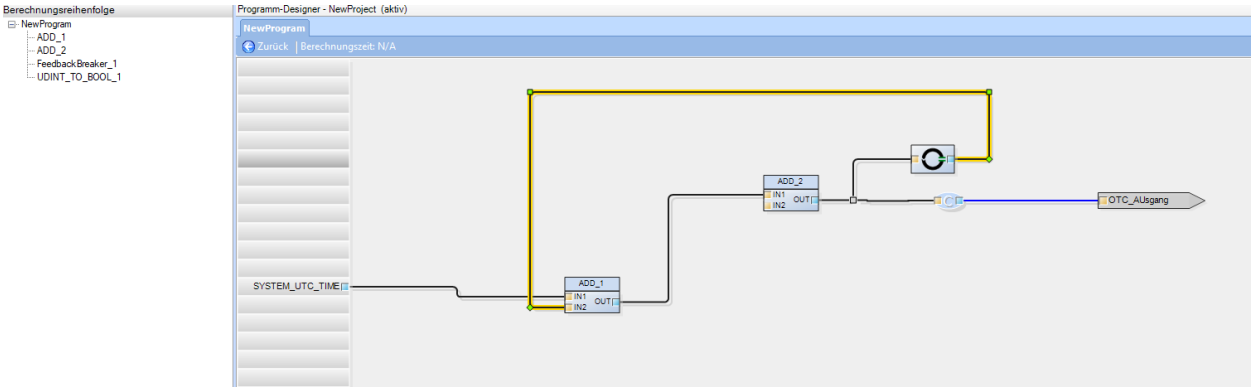
Ist diese Option gesetzt (Default Einstellung):

Ist diese Option nicht gesetzt:

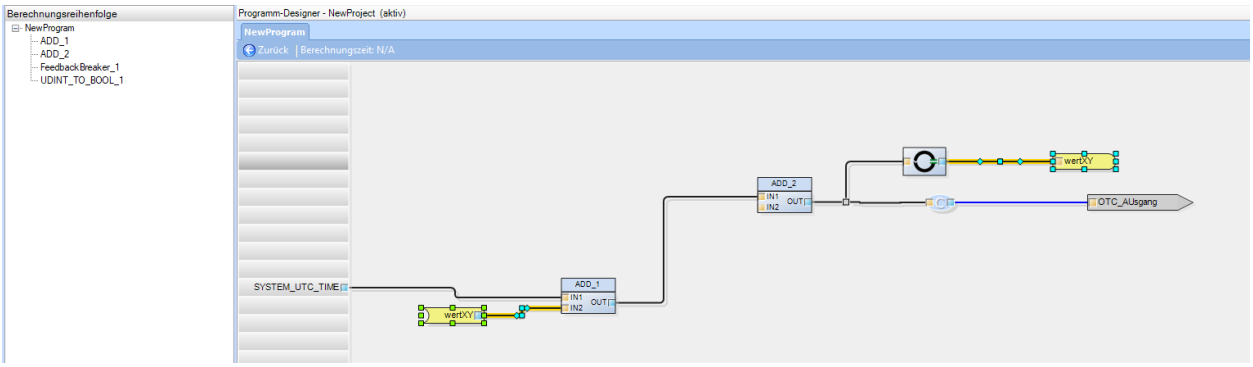


Dieser FeedbackBreaker muss als Rückkopplungs-Punkt eingebaut werden.

Damit ist dies als Endpunkt der Rückkopplung gekennzeichnet und die Reihenfolge ist klar defniniert.

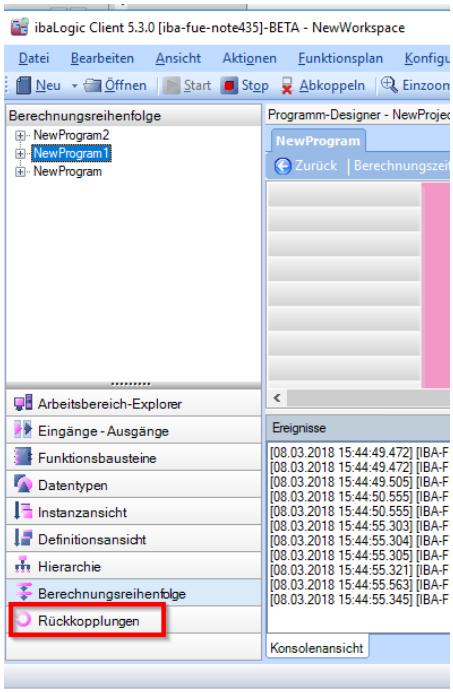


Zum besseren Verständnis und auch zur besseren Übersicht empfiehlt es sich mit IntraPage-Konnektoren zu arbeiten. Das Layout sieht dann so aus:



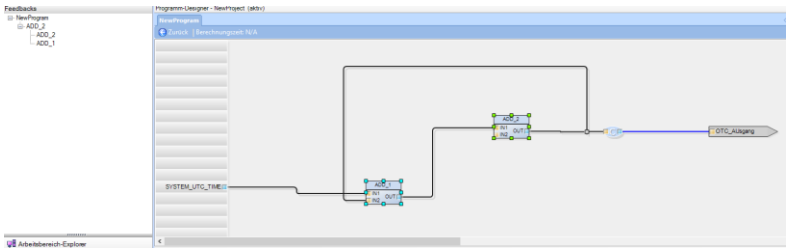
1.3 Verzeichnis aller Rückkopplungen

Um in seinen Layouts die Rückkopplungen zu finden, gibt es eine neue Ansicht:



In dieser Ansicht findet man alle im Projekt vorhandenen Rückkopplungen.

Durch Doppelklick auf die Rückkopplung werden die zugehörigen Bausteine markiert.



So kann man das Netzwerk sehen und den Rückkopplungspunkt finden.

In neueren Layouts sollte man immer mit dem FEEDBACKBREAKER Baustein arbeiten.

Will man in älteren großen Layouts die Rückkopplungen eliminieren, kann es unter Umständen schwierig sein, den Rückkopplungspunkt zu finden.

Wenn man sein Programm kennt, weiß man eventuell wo sich Rückkopplungen befinden und kann diese dann einfach eliminieren.

Hat man aber keine direkten Kennnisse, muss man das Programm durchforsten.
Hier empfiehlt es sich das Fenster PROGRAMMÜBERSICHT zur Hilfe zu nehmen.

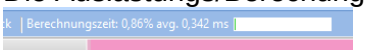
Vorgehensweise:

- Rückkopplungs-Bausteine markieren (durch Doppelklick auf die Rückkopplung im Rückkopplungsverzeichnis)
- Jetzt kann man im Übersichtsfenster navigieren und die Bausteine suchen. Anhand der Markierungen kann man auch sehen, welches der letzte Baustein ist.
- Man kann auch zu den einzelnen Bausteinen des Feedbacks springen, indem man sie doppelt anklickt. Wenn man dabei das Übersichtsfenster beobachtet, sieht man vielleicht, welcher Baustein ganz unten im Layout ist. Der hat oft die Rückkopplung als Ausgang.
- Ebenso kann man eine Linie mit gedrückter ALT Taste anwählen. Damit wird die Linie über IPCs hinaus komplett markiert. Durch scrollen über das Übersichtsfenster kann man dann sehen, ob diese Linie oben im Layout wieder auftaucht. Damit hätte man auch eine Rückkopplung gefunden.

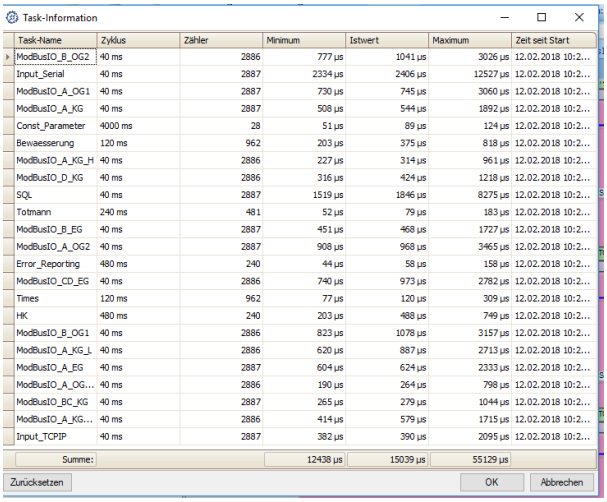
2 Diagnose der Task-Berechnungsreihenfolge

Für die Diagnose der Tasks und deren Abarbeitung hatte man bisher folgende Tools/Anzeigen zur Verfügung:

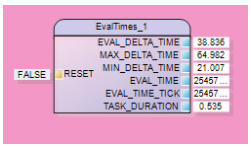
- Die Auslastungs/Berechungszeit-Angabe pro Task im Task-Kopf



- Die Übersicht über alle Task in der Task-Information (ANSICHT-TASKINFORMATION)



- Einsatz des EVALTIMES Baustein

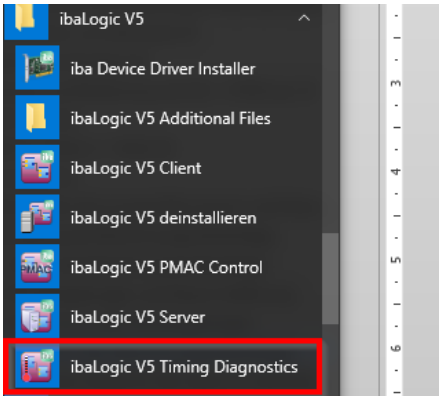


Hinweis: Der EVALTIMES Baustein eignet sich auch zum Messen von Laufzeiten innerhalb eines Programm. Wenn man zwei EVALTIMES Bausteine in den Datenfluss einbindet, kann man durch Differenz-Bildung die Laufzeit aller dazwischen liegenden Bausteine ermitteln.

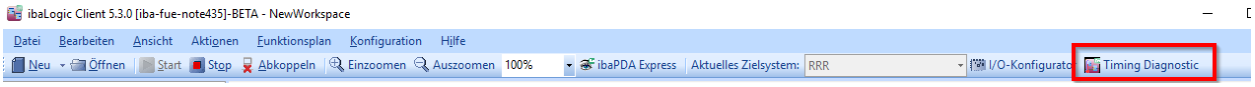
Mit ibaLogic 5.3.0 hat man nun eine noch genauer Diagnose-Möglichkeit. Es gibt das **ibaLogic V5 Timing Diagnose-Tool.**

Dieses Tool ist lizenzpflichtig und benötigt eine Dongle-Freischaltung.

Das Tool kann aus der Windows-Menüleiste aufgerufen werden (wenn ibaLogic auf dem Rechner installiert wurde):



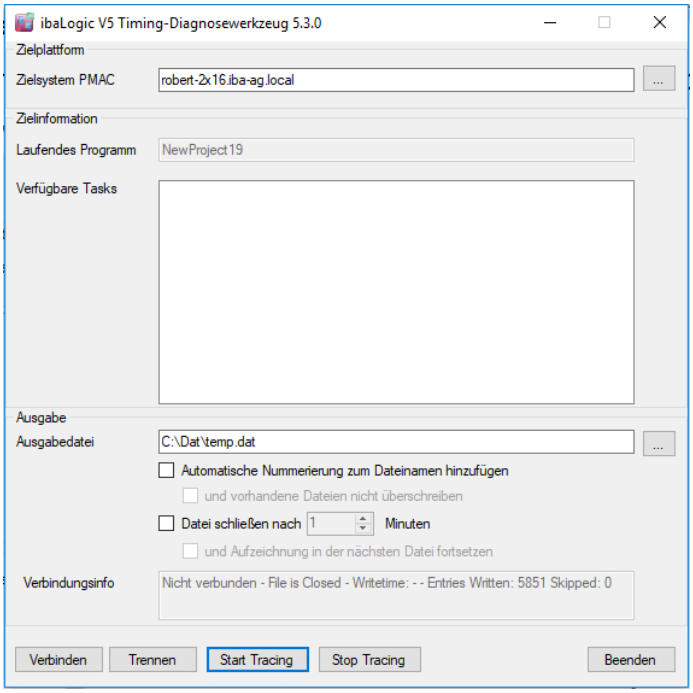
Oder auch direkt aus einem ibaLogic-Client heraus.



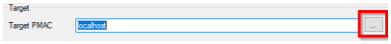
Dieses Tool schreibt eine Dat-Datei mit allen Task-Zustandsänderungen.

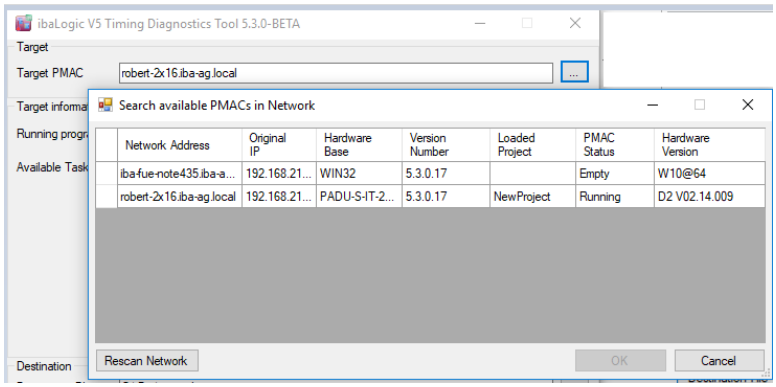
Vorgehensweise:

- Tool aufrufen

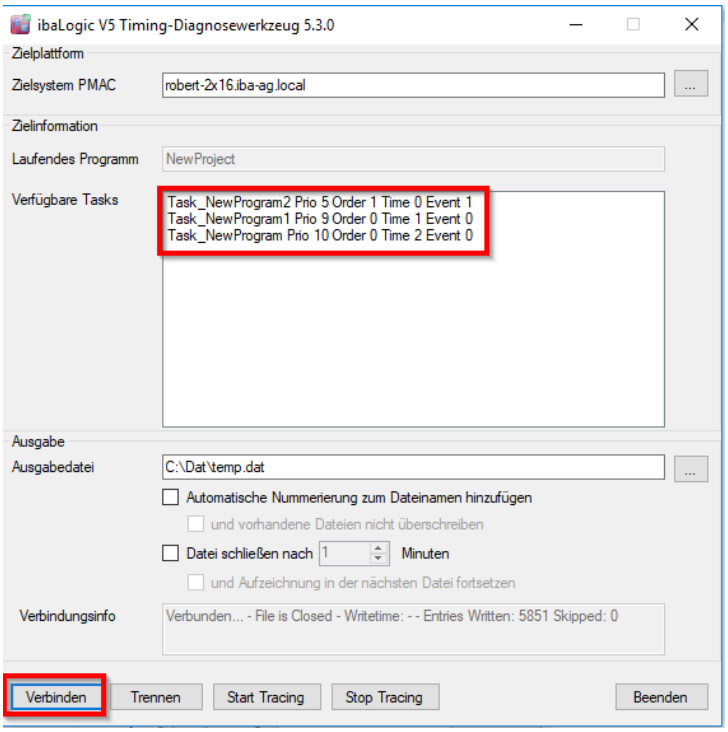


- Verbindung mit dem entsprechenden PMAC aufnehmen. Beim Start aus ibaLogic heraus wird der aktuelle Ziel-PMAC vorausgewählt. Über den Button erreicht man alternativ die Netzwerksuche.

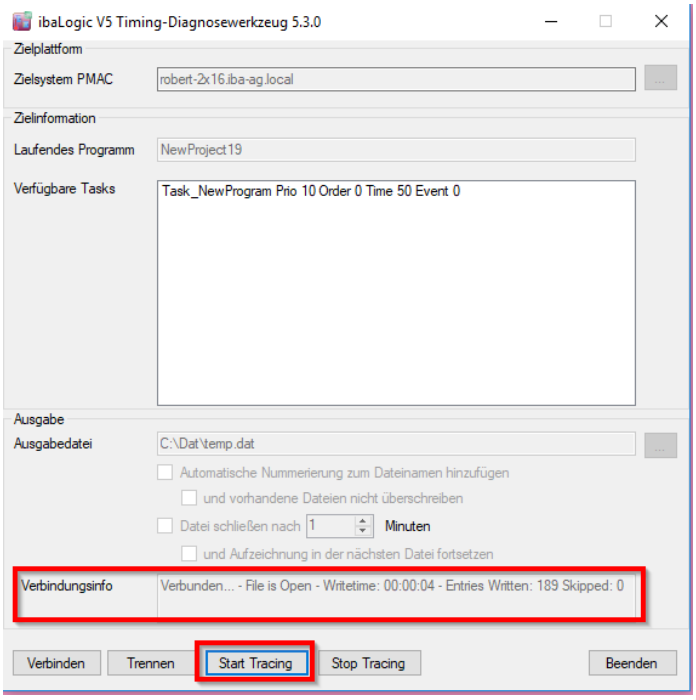




- Eingabe eines Dateinamens für die Dat-Datei oder über die Ordneranwahl Pfad und Datei auswählen.
- Über die Optionen das Ablegen der Dat-Dateien festlegen.
- Man kann nun über VERBINDEN sich zu einem laufenden PMAC verbinden. Die gefundenen Tasks werden in der Task-Übersicht angezeigt.



- TRENNEN trennt die Verbindung
- Eine Aufzeichnung wird erst mit START TRACING gestartet.
  Hinweis: Wenn noch kein PMAC verbunden ist, macht START TRACING automatisch ein VERBINDEN.



Über die VERBINDUNGINFO sieht man ob Daten aufgezeichnet werden. Der Zähler für geschriebene Daten WRITTEN wird ständig erhöht. Ebenso sieht man die aktuelle Zeitdauer der gerade geschriebenen Dat-Datei WRITETIME

- STOP TRACING beendet die Aufzeichnung und das geschriebene Dat-Datei kann mit ibaAnalyzer angesehen werden

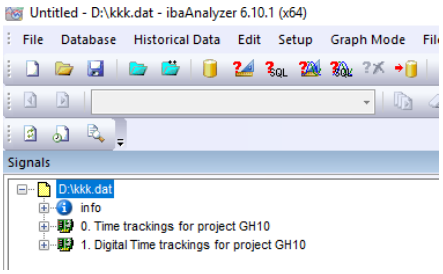
2.1.1 Task-Diagnose Dat-Datei im ibaAnalyzer ansehen

Jede Task kann drei Zustände haben:

ZUSTAND	Analogwert	Digitalwert
Ruhend	0	0
Läuft	1	1
Unterbrochen	-1	1

Diese Daten werden mit 1 µs Genauigkeit in die Dat-Datei eingetragen.
Haben alle Tasks die gleiche Priorität, so können sie nicht unterbrochen werden. Zu diesem Zweck gibt es auch die Aufzeichnung als digitalen Wert. Damit kann man leichter durch Doppelklick sich die Laufzeiten auswerten etc.

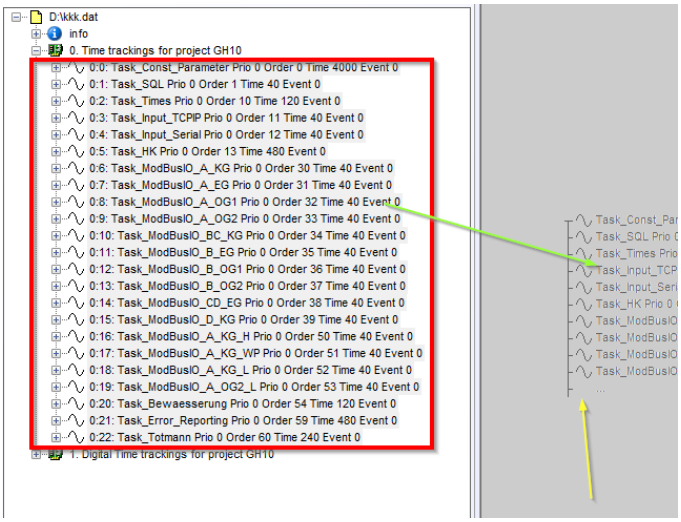
Mit dem Öffnen der Dat-Datei im ibaAnalyzer bekommt man daher folgende Ansicht:



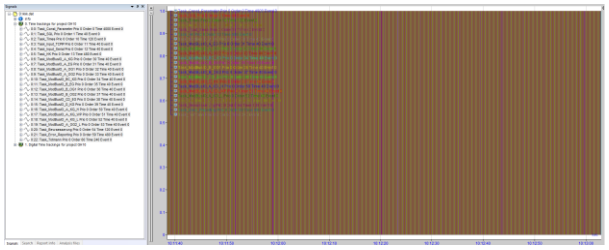
Man sieht den analogen und den digitalen Baum.

Anzeigen der analogen Werte
Hier empfiehlt sich folgendes Vorgehen:

Alle Signale markieren und per Darg&Drop (grüner Pfeil) in die Ansicht ziehen. Dabei vor dem Loslassen die SHIFT Taste drücken und gedrückt lassen. Damit werden alle in einen Graphen mit gemeinsamer Skala gepackt.

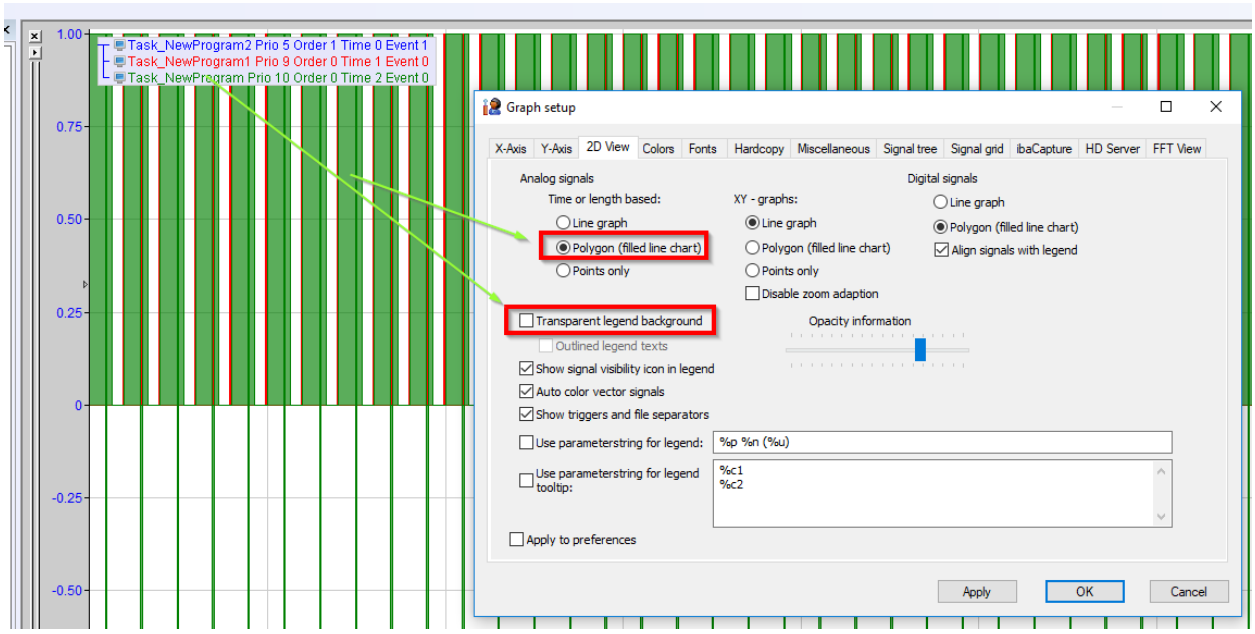


Das Ergebnis sieht so aus.

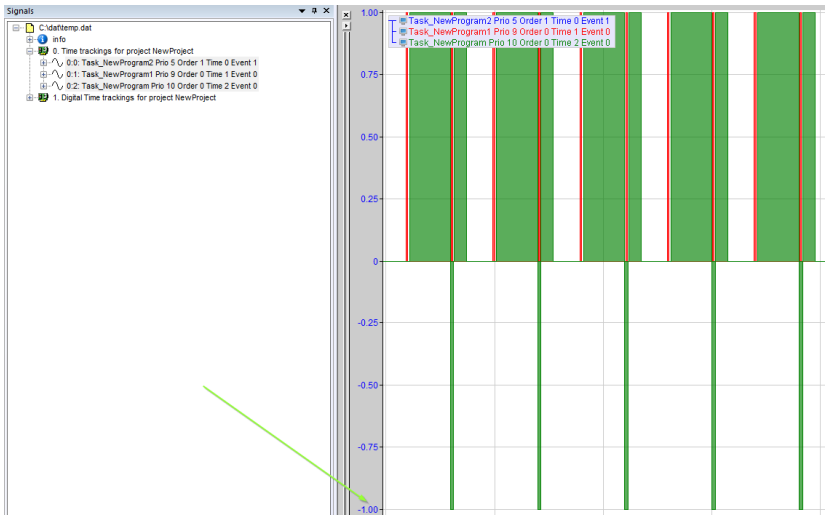


Zur besseren Übersicht sollte man jetzt im Graphen noch ein paar Settings machen:

Mit rechter Maustaste erhält man das SETUP Menü.



Wichtig ist der ausgefüllte Graph. Damit hat man dann wenn man hineinzoomt eine klare Ansicht.

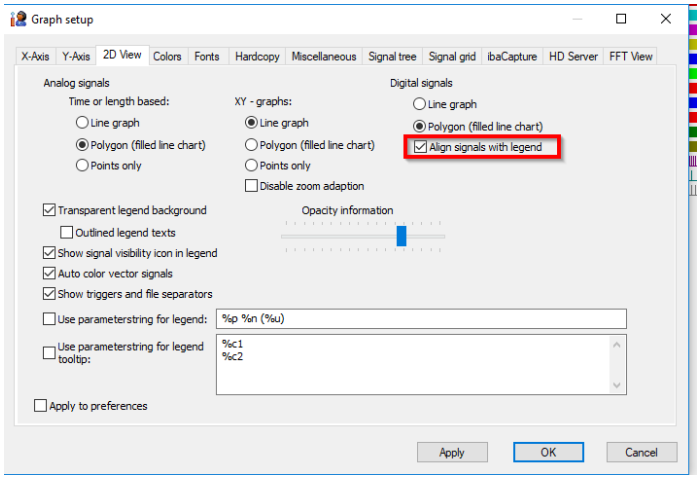


Jede Task ist nun mit ihrer Berechnungsdauer und ihrem Zeitpunkt der Berechnung im Bezug auf alle anderen Tasks zu sehen. Werte von -1 zeigen an, dass hier Tasks auf Grund von Prioritäten unterbrochen wurden. Gibt es keine Unterbrechungen ist die digitale Anzeige übersichtlicher. (siehe nächstes Kapitel)

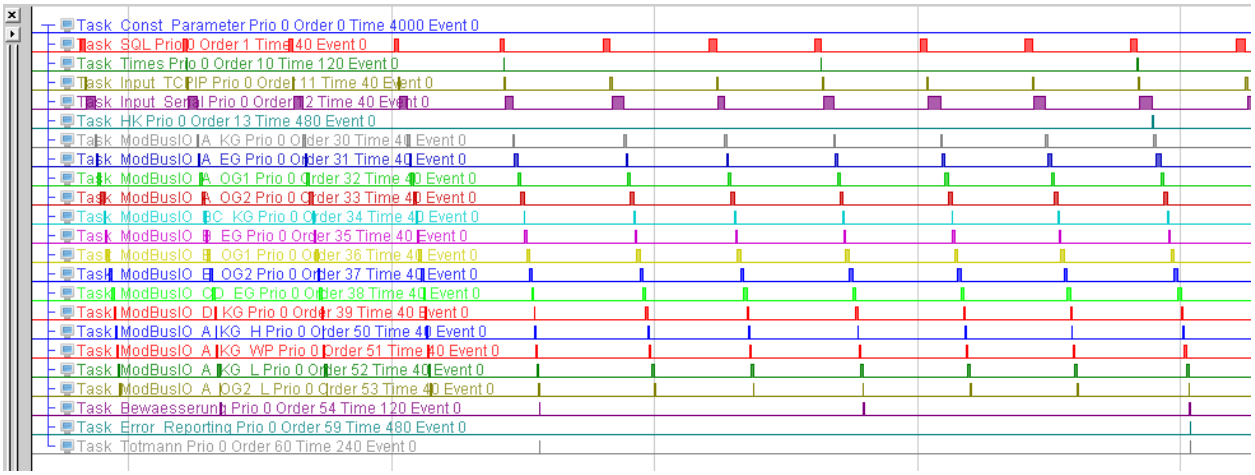
Anzeigen der digitalen Werte

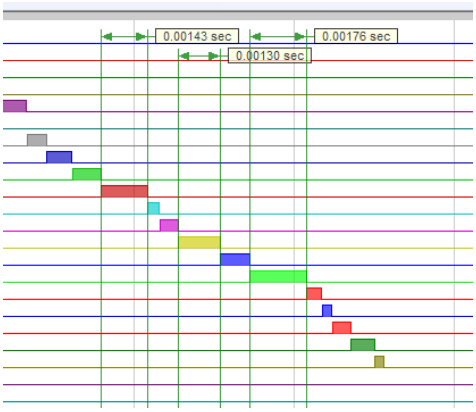
Hat man nur eine Prioriät in allen Tasks verwendet, kann man auch die digitalen Werte benutzen.

Diese genau wie bei den analogen per Drag&Drop und SHIFT Taste in die Ansicht holen.



Die Signale noch auf Legenden-Höhe setzen und man erhält diese Ansicht:

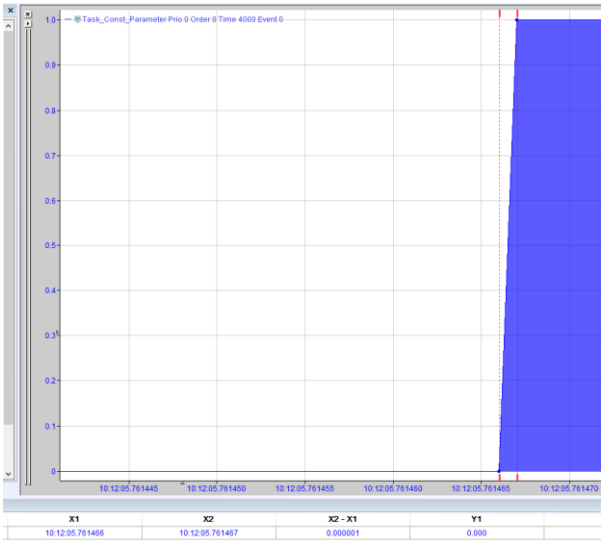




Ein Doppelklick auf den jeweiligen High-Zustand zeigt die Laufzeit direkt an. Ein Doppelklick auf Low-Zustand kann die Abstände anzeigen.

Hinweis: Die Aufzeichnung erfolgt mit non-äquidistanten Werten. Es werden nur die Status-Änderungen eingetragen. Daher eignen sich die Signale nicht für Formeln im ibaAnalyzer, da dieser nur äquidistante Signalwerte bearbeiten kann.
Man kann aber durch die Funktion RESAMPLE äquidistante Signale bilden, falls dies notwendig wäre.

Hinweis: Normalerweise würde eine Zustandsänderung das analoge Signal so aussehen lassen, dass eine schräge Linie vom letzten Zustand zum neuen Zustand gezogen wird. Dies macht eine sinnvolle Darstellung nur umständlich möglich. Daher werden beim Aufzeichnen einer Status-Änderung immer zwei Werte eingetragen. Der neue Zustand und 1µs vorher der alte Zustand nochmal wiederholt. Damit ergibt sich eine Schräge die man praktisch nur beim extremen Einzoomen sieht. Aber die analogen Zustände können für die Übersicht klar dargestellt werden.

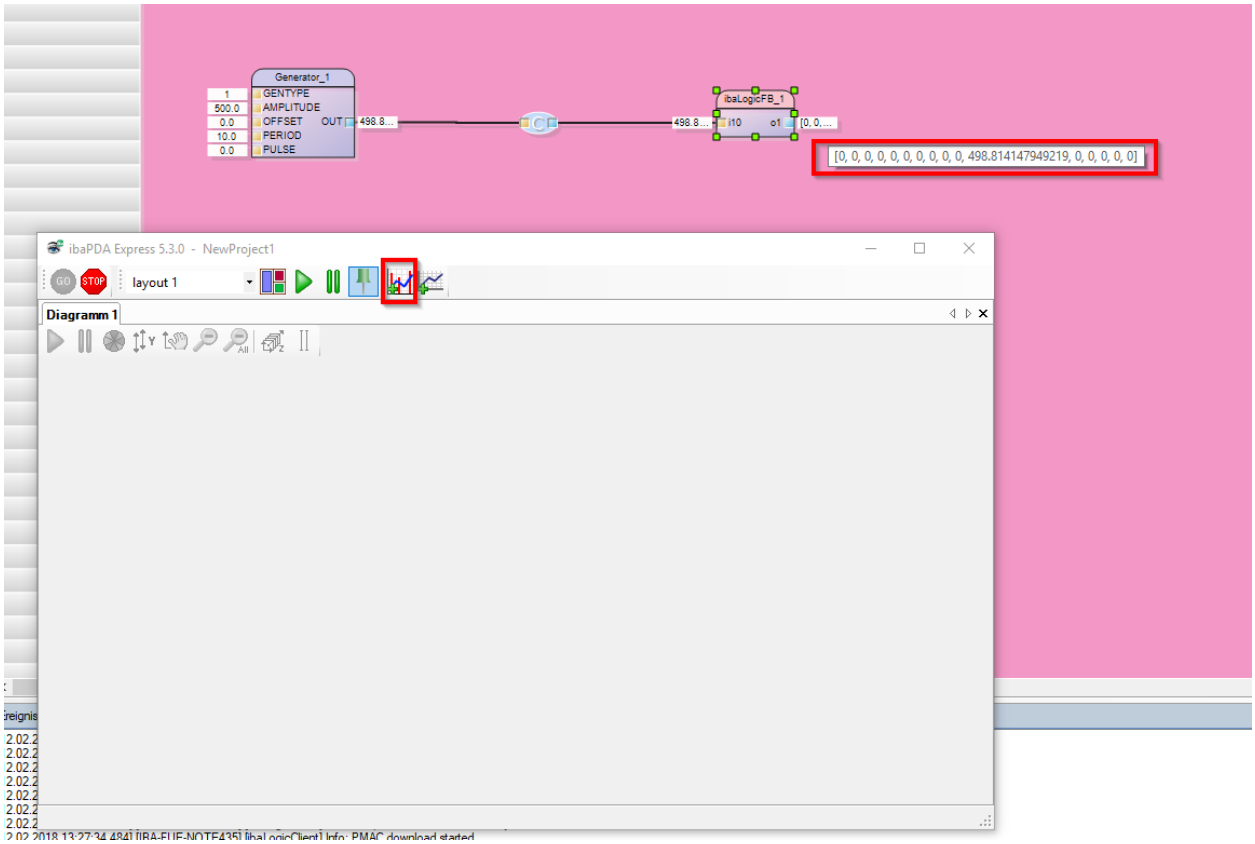


3 Anzeige von Arrays in ibaPDAExpress

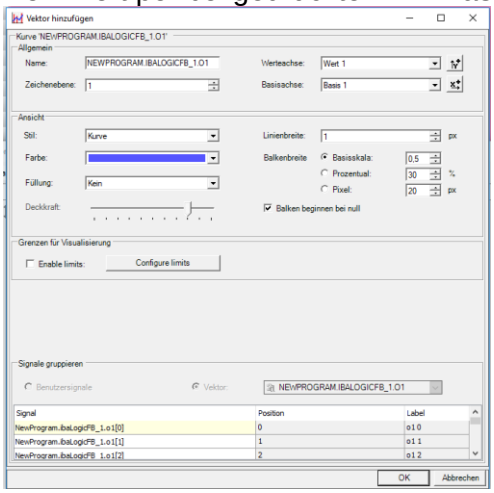
Die Anzeige von Arrays in ibaLogic ging bisher nur über einen eigenen FB, der entsprechende Bereiche ausgibt.
Damit Array-Werte einfacher zu lesen sind, kann man Arrays nun in ibaPDAExpress hineinziehen.

Der Tooltip eines Arrays zeigt auch die laufenden Werte an, aber bei größeren Arrays ist es sehr unübersichtlich.

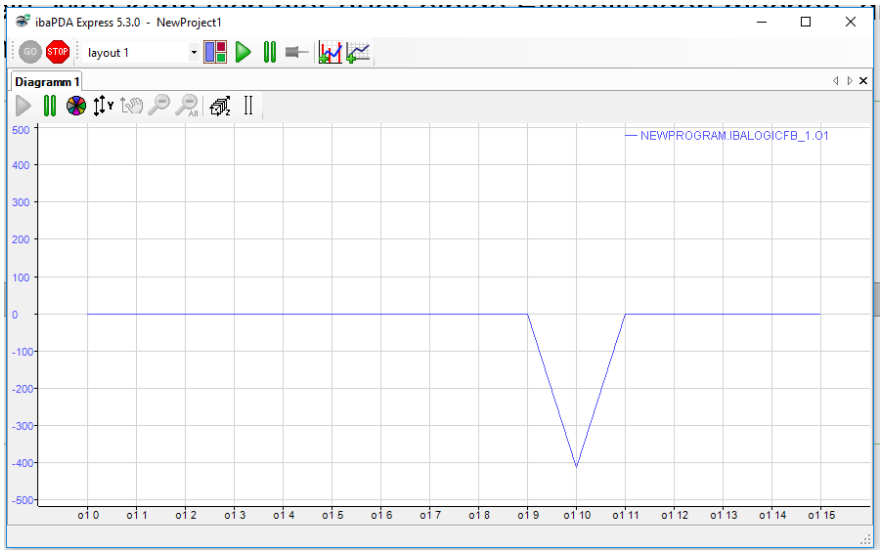
In ibaPDAExpress kann man sich nun das DIAGRAMM als Array-Anzeige aufmachen.



Den Wert per bei gedrückter ALT-Taste in das Diagramm ziehen.



Diese Maske kann man einfach bestätigen.
Hinweis: Man kann hier auch einige Einstellungen machen, die man der DIAGRAMM Beschreibung des ibaQPanels entnehmen kann.

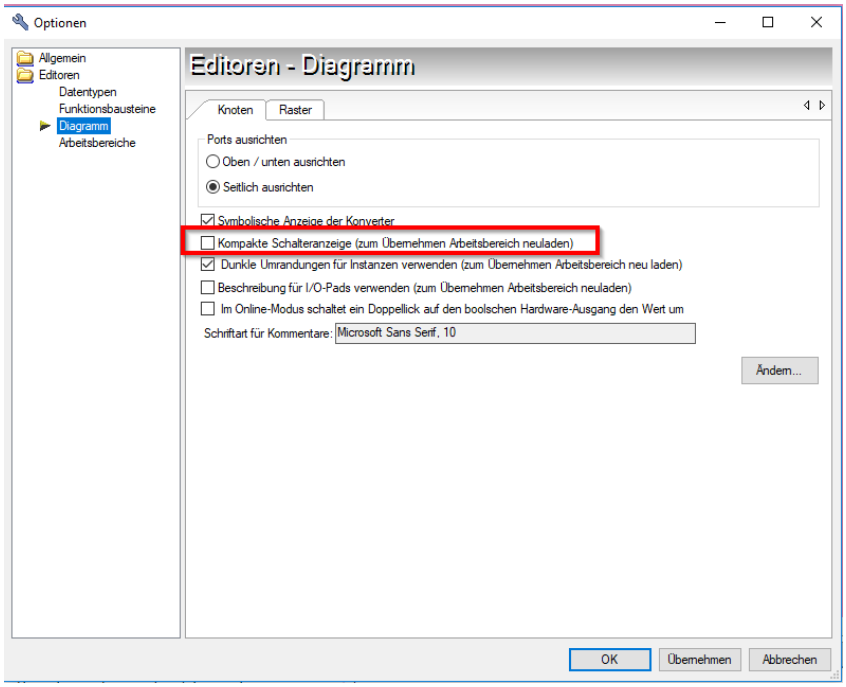


Man erhält nun eine Darstellung des Arrays. Beginnend auf der linken Seite mit dem niedrigsten Index und auf der rechten Seite mit dem höchsten Array-Index (hier: 15).
Zoomen und andere bekannte Hantierungen sind möglich.

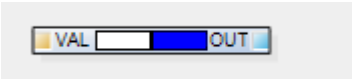
4 Schalter als kompaktes Symbol

Es ist jetzt möglich sich den Schalter auch als kompaktes Symbol anzuzeigen.

Dazu gibt es in den Optionen folgende Funktion:

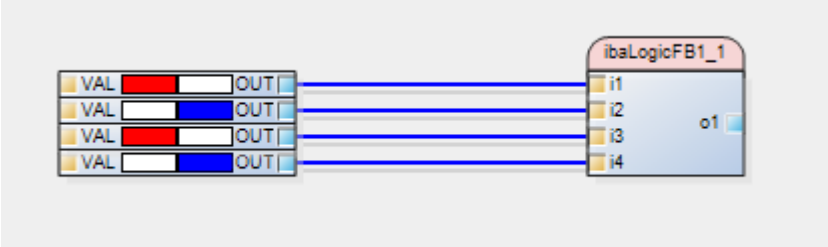


Hat man diese ausgewählt werden die Schalter nun so dargestellt:



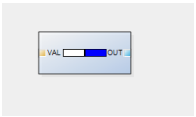
Bedenken Sie, dass der innere Bereich der Schaltbereich ist. Zum Verschieben daher den Schalter im Bereich der Beschriftung VAL/OUT den Schalter mit der Maus anpacken.

Die kompakte Darstellung ermöglicht nun ein direktes Gegenüberstellen von Schaltern zu Konnektoren an Bausteinen.



Damit bestehende Schalter kompakt dargestellt werden, muss der Arbeitsbereich nach einstellen der Option einmal neu geöffnet werden.

Die „alten Schalter“ werden aber in ihrer Größe noch unverändert dargestellt und müssen manuell korrigiert werden, d.h. in ihrer Höhe angepasst werden.



5 Erweitertes Playback

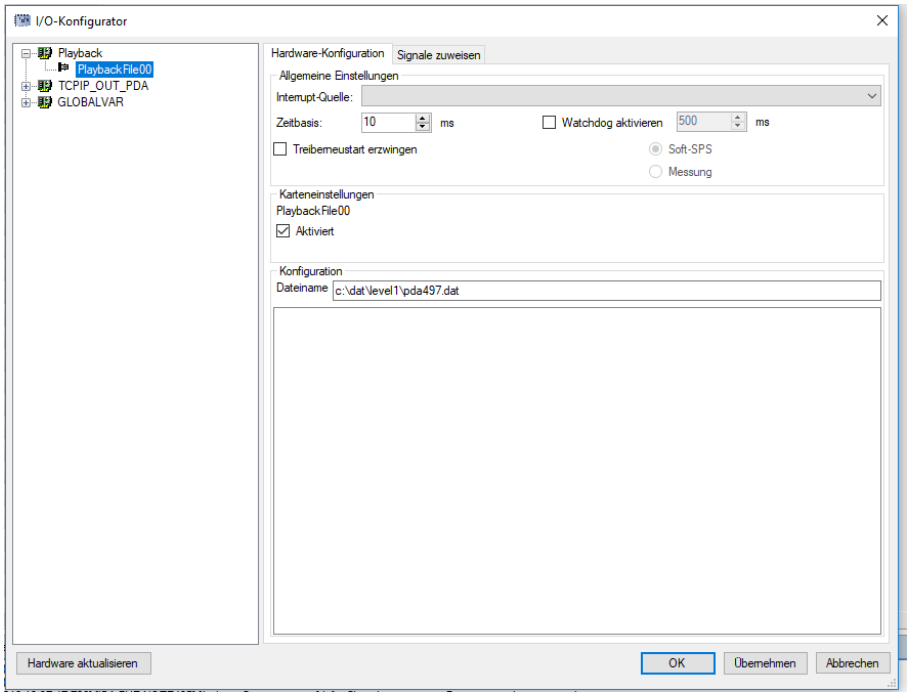
Bisher war es möglich, eine Playback Datei als Datenquelle abzuspielen.

Das erweiterte Playback kann nun:

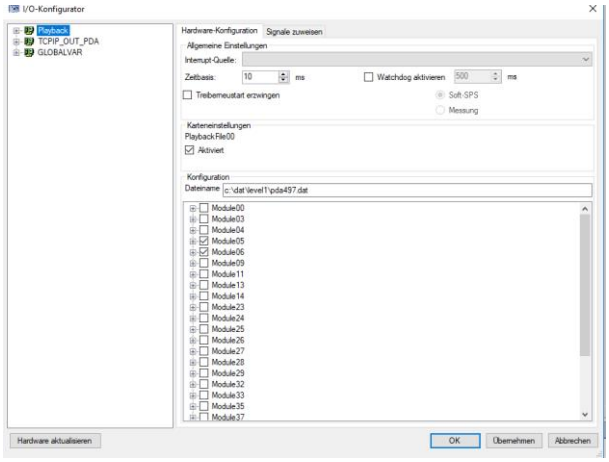
- Verschiedene Playback-Dateien gesteuert abspielen. Die Playback-Dateien sollten von der gleichen Datenquelle kommen, damit Signale immer eindeutig sind.
- Steuern des Playbacks durch
  - Anhalten
  - Replay (von Anfang wieder beginnen)

<u>Hinweis: Beim ersten Laden von alten Projekten mit aktivem Playback muss einmal der I/O Konfigurator geöffnet werden und „Hardware aktualisieren“ gedrückt werden.</u>

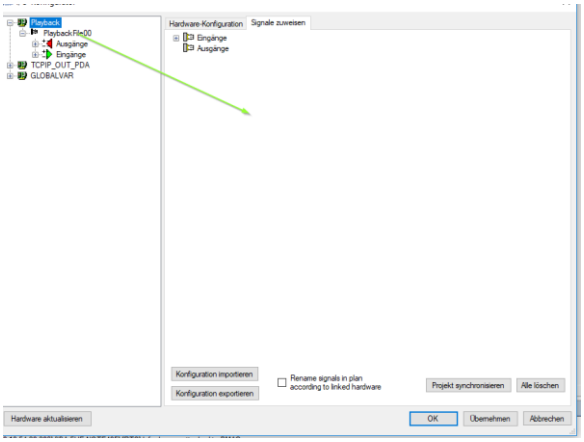
Anlegen eines ersten DatFiles wie bisher:



- Eintragen Pfad und Dateiname, nachdem man Playback aktiviert hat
- ÜBERNEHMEN



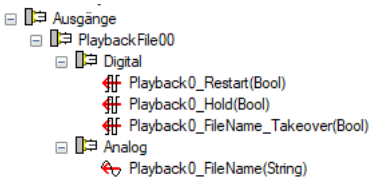
- Module und/oder Signale auswählen
- ÜBERNEHMEN



- Dann noch die Signalzuweisung per Drag&Drop vornehmen.
- Mit OK den Dialog verlassen

Damit hat man nun alle Signale zur Verfügung.

Für die **Ausgänge** gibt es:



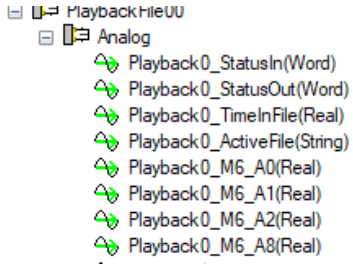
Restart eine HighFlanke an diesem Ausgang startet die Playback-Datei von vorne
Hold ein High-Signal an diesem Ausgang hält das Abspielen der Datei an

Für das Abspielen verschiedener Playback-Dateien dienen die Ausgänge TAKEOVER und FILENAME.

FileName abzuspielende Dat-Datei mit Pfad und Name
FileName_Takeover High-Flanke übernimmt die unter FileName angegeben Datei zum abspielen und fängt mit dem Playback am Anfang der Datei an.

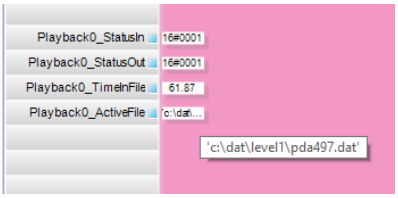
Hinweis: Diese letzten beiden Ausgänge braucht man nicht zu beschalten, wenn man nur die Dat-Datei aus der IO-Konfiguraton abspielen will.

Eingänge:



Die Eingänge geben einen Status für die Ein/Ausgänge an.

Er sollte 1 sein, alles andere zeigt einen Fehler an (z.B. Datei nicht gefunden).



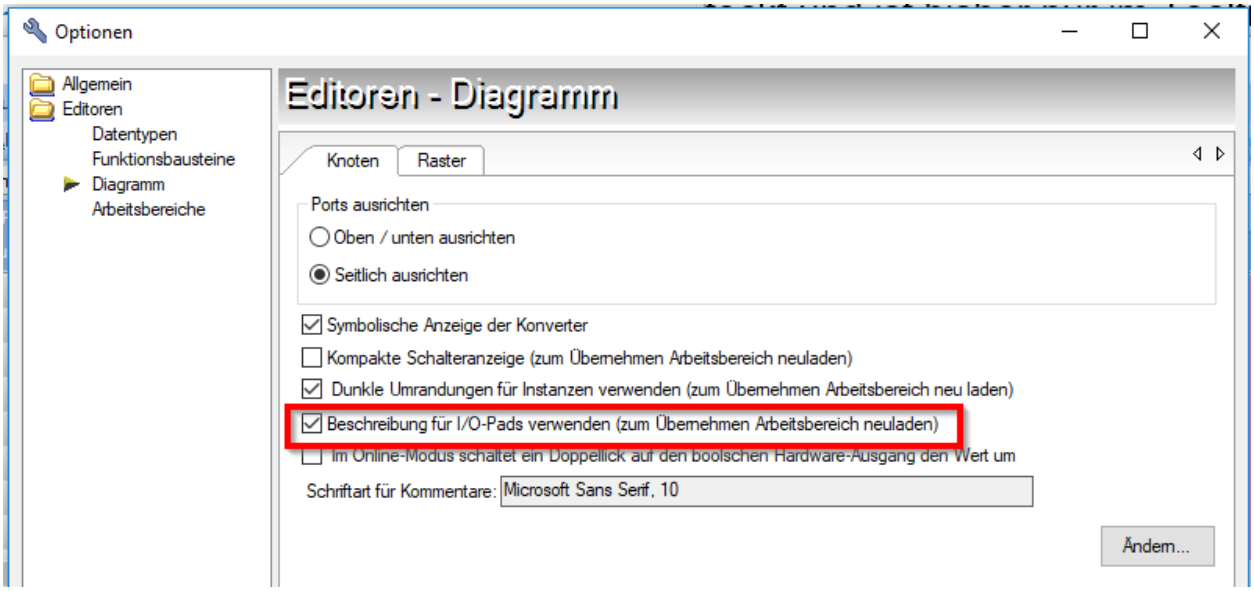
TimeInFile zeigt die aktuelle Zeit in der Playback-Datei an.
ActiveFile zeigt den Pfad und Namen der gerade abgespielten Playback-Datei als Rückmeldung.

6 Wahlweise Anzeige von Signalnamen oder Signal-Beschreibung

Bisher war im Plan es nur möglich, den Signalnamen aus der IO-Konfiguration für die IO-Signale anzuzeigen. Bei Playback-Dateien oder anderen Konfiguration wird der eigentliche Signalname in der Signalbeschreibung versteckt und war bisher nur im Tooltip zu sehen.

Der Grund war, dass Signalnamen IEC konform sein müssen und bestimmte Signal-Imports oder Playback-Signalnamen diese Norm nicht erfüllen. Daher wird der Signalname generisch erzeugt und der eigentliche Signalname in die Beschreibung verschoben.

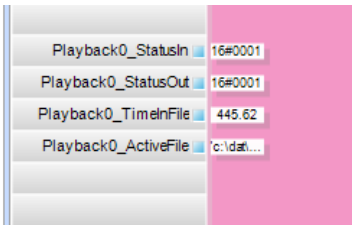
Es ist nun in den Optionen möglich, sich statt dem Signalnamen die ursprüngliche Bezeichnung anzeigen zu lassen. Das betrifft nur die Anzeige im Plan, für die Berechnung oder Anzeige der Werte in PdaExpress wird nach wie vor der IEC-Konforme Name verwendet.



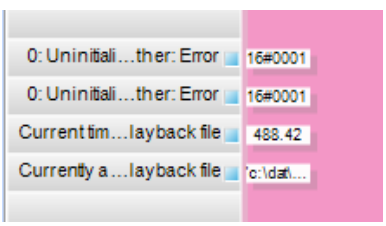
Eventuell den Arbeitsbereich nochmal laden, damit die Option nach Umstellung wirksam wird.

Beispiel:

Option nicht angewählt:



Option angewählt:



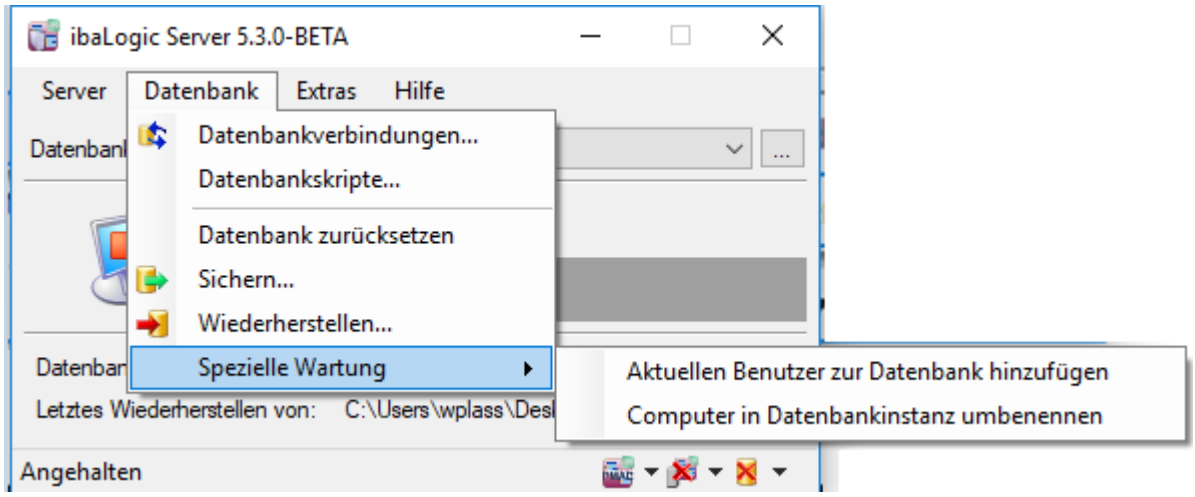
Um auch längere Bezeichnungen darzustellen, wurde eine für Microsoft typische Darstellung gewählt. Es werden die ersten und letzten Zeichen dargestellt, der Rest wird durch … unterdrückt. Auch hier zeigt aber der Tooltip den vollständigen Namen an.

Diese Darstellung wurde gewählt, da oft am Anfang einer Bezeichnung der große Zusammenhang und am anderen Ende das Detail zu finden ist. Dazwischen sind oft gleichlautende Bezeichnungen, die unterdrückt werden können.

7 Hilfsfunktionen für Änderung an Rechnername oder Datenbank

Ändert man an seinem ibaLogic-Rechner den Rechnernamen, oder ist mit einem anderen Benutzer eingeloggt als der mit dem ibaLogic installiert wurde, so kann man unter Umständen ibaLogic nicht mehr aufrufen und starten etc.

Dafür gibt es nun beim ibaLogic Server zwei neue Funktionen:



Akutellen Benutzer zur Datenbank hinzufügen: Es wird der Datenbank der aktuelle Benutzer hinzugefügt, so dass dieser auch Zugriff hat.

Computer in Datenbankinstanz umbenennen: Hat man den Rechnernamen geändert, oder die Arbeitsgruppe oder Domäne geändert, kann man damit die Datenbank wieder an den Rechnernamen anpassen

8 Positionieren von Elemente

Das Positionieren von Elementen kann nun auch über die Pfeiltasten erfolgen.

Dabei gilt folgendes:

- Wenn nichts markiert ist, dann wird mit dem Pfeiltasten der Plan bewegt
- Wenn ein oder mehrere Elemente markiert sind, dann wird werden die ausgewählten Elemente mit den Pfeiltaste verschoben.
  Wenn man dann noch zusätzlich die STRG Taste gedrückt hält, erfolgt eine Verschiebung in feiner Auflösung.

Dies erleichtert und beschleunigt eine genaue Positionierung der Bauteile z.B um gerade Verbindungslinien zu bekommen.
Auch Verbindungsknoten können so verschoben werden.

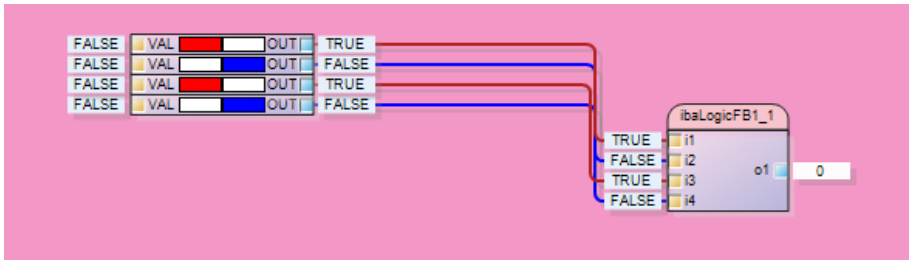
9 Verbessertes Routing

Eine unschöne Eigenschaft des Routings war bisher, dass von Hand korrigierte Verbindungen sofort wieder neu geroutet wurden, wenn man einen Baustein nur eine minimale Verschiebung macht.

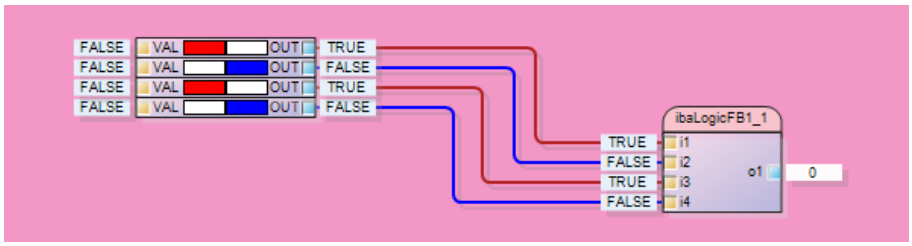
Es wurde nun eine Verbesserung eingebaut:
Solange man beim Verschieben über den eigentlichen Routing-Knick-Punkt nicht verschiebt, bleiben die Linien wie sie gezogen wurden.

Beispiel:

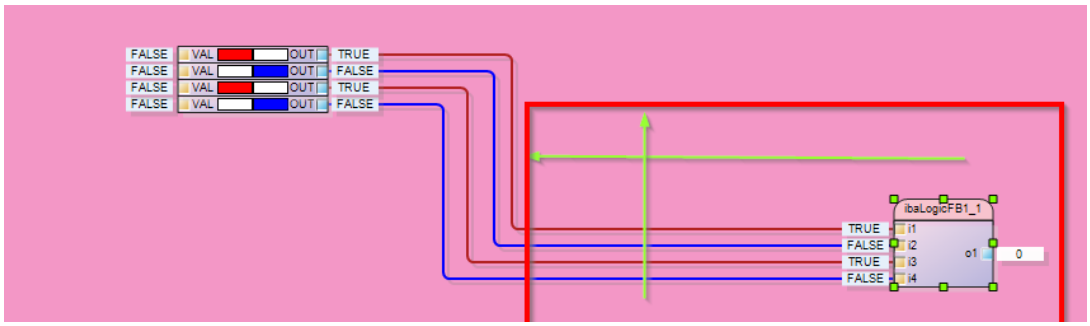
Dieses Routing ist durch Auto-Routing gegeben:



Jede Linie wird nun manuell korrigiert:



Jetzt kann man den Baustein innerhalb des gekennzeichneten Rahmens bewegen, ohne dass die eingerichteten Knicke neu geroutet werden.



Erst wenn man horizontal weiter nach links verschiebt als die Knickpunkte sind oder vertikal höher als die Knickpunkte, wird ein neues Routing der gesamten Linien gemacht.

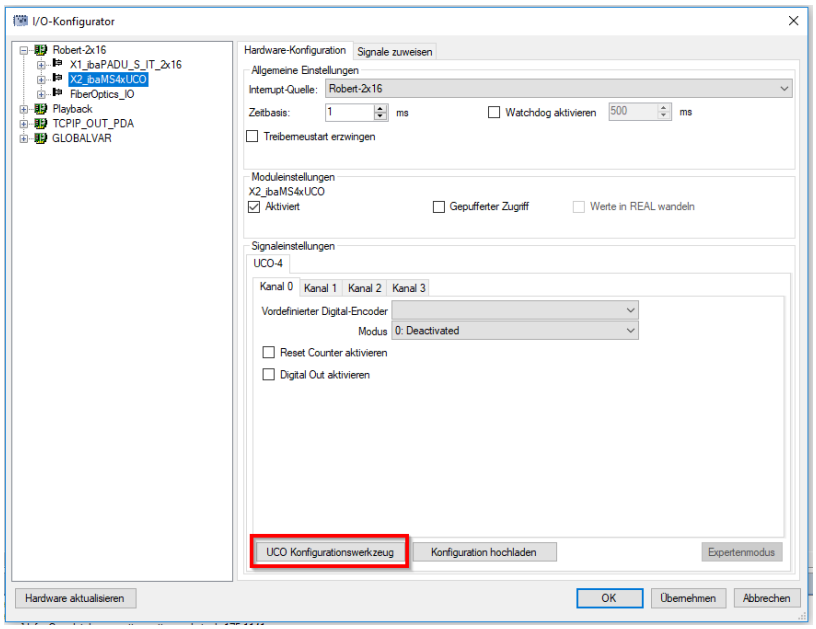
Dies erleichtert das Fixieren von Linien, wenn es auch noch nicht das Optimum ist.

10 Hinweis auf Bedienung mit ATL / STRG etc

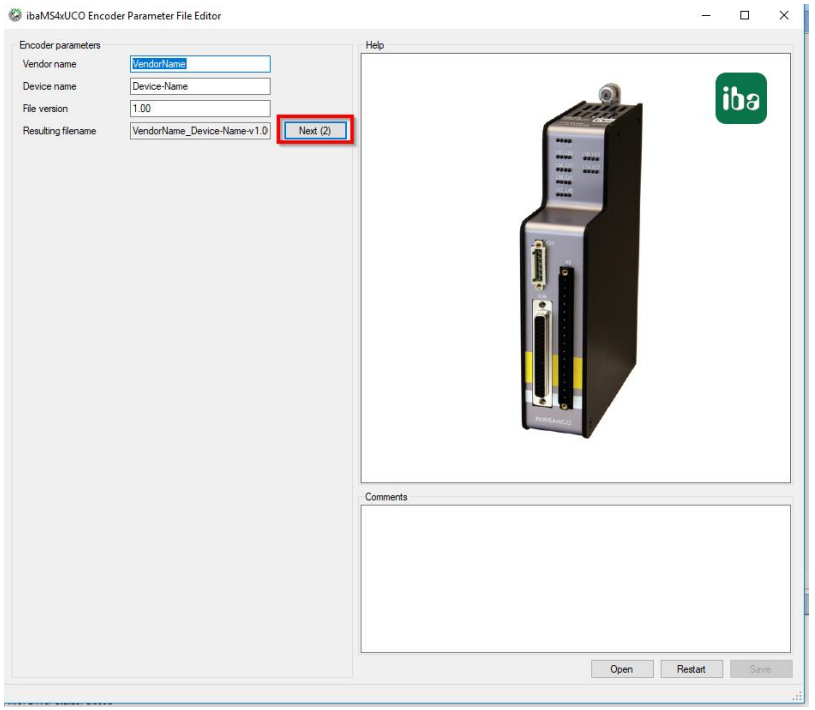
Drückt man die STRG / ALT / SHIFT Taste so erscheint in der unteren Zeile ein Hinweis welche Möglichkeiten diese Tasten bieten.

11 UCO Tool

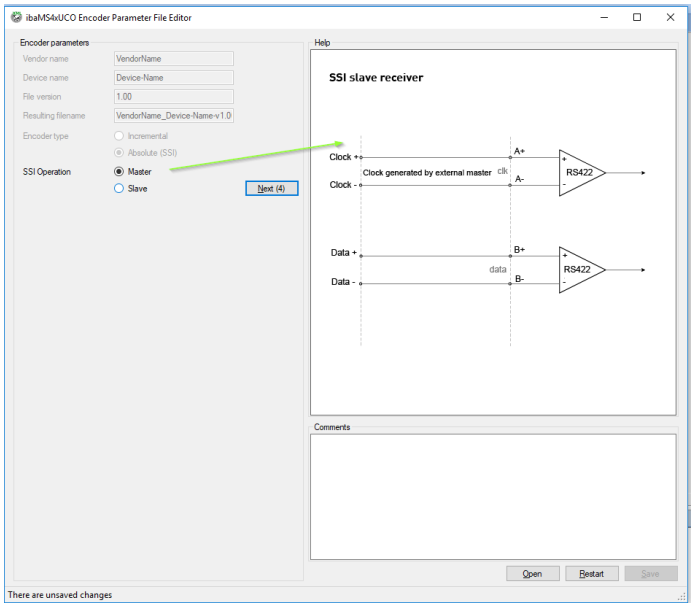
Für die Baugruppe ibaMS4xUCO wurde das aus ibaPDA bekannte Konfigurationstool integriert.



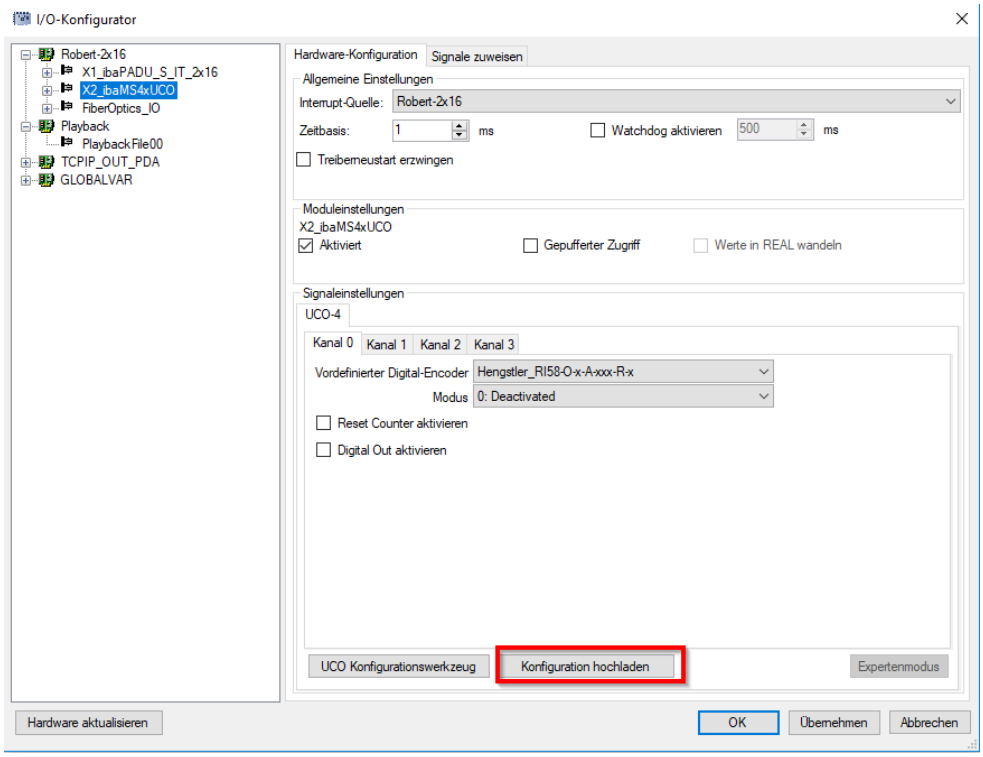
Man hat damit einen Assistenten zur Einstellung. Mit NEXT geht man durch die Konfiguration.



Dabei werden einem immer auf der rechten Seite zum jeweiligen Punkt Erläuterungen geliefert.



Ist man mit der Konfiguration durch, sichert man diese in einer XML Datei (SAVE)



Diese XML Datei wird anschließend automatisch an das PaduS-IT-2x16 hochgeladen. Möchte man sie an weitere Zielgeräte verteilen, kann die Datei mit KONFIGURATION HOCHLADEN ausgewählt und im UCO Modul damit abgelegt werden.

12 DAQ-S / DAQ-C / DAQ

Diese ibaLogic Version kann auf den Geräten ibaDAQ-S / ibaDAQ-C und ibaDAQ installiert und gestartet werden. Die Hardware Schnittstellen der Kopfstation und die Module sind aber von ibaLogic NICHT ansprechbar. Sie stehen ausschließlich ibaPDA zur Verfügung. ibaLogic kann wie üblich mit ibaPDA kommunizieren oder auch die TCP/IP Verbindungen der Geräte nutzen.

13 IL5 name

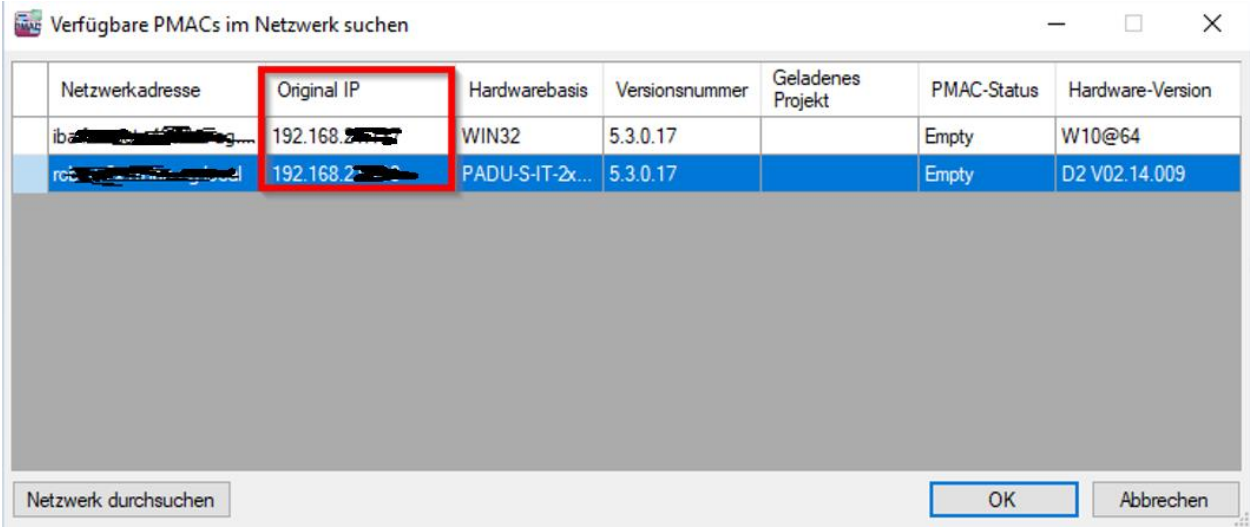
Export-Dateinen haben jetzt die Endung .IL5 und sind damit an die ibaLogic Version angepasst. Ältere .IL4 Dateien werden trotzdem erkannt und können auch importiert werden.

14 Verbessertes Disconnect/Reconnect und Erkennung von Änderungen

Beim Verlassen des Clients bzw beim Trennen des Clients vom PMAC und ein anschließendes Öffnen und Wiederverbinden war dies manchmal nur mit Neustart des Projekts möglich. Es wurden nun weitere Verbesserungen vorgenommen, so dass das Wiederverbinden erheblich verbessert wurde.

15 PMAC Suchfenster mit IP-Adresse

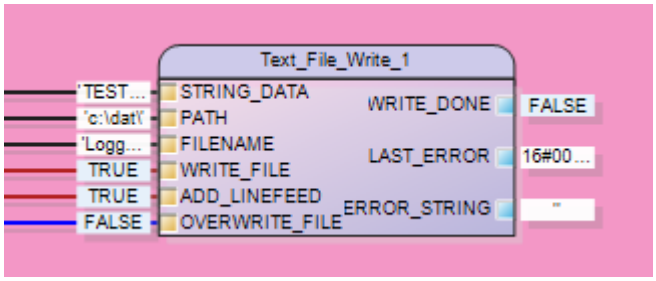
Das Suchfenster für PMACs, welches man bei der Zielsystemkonfiguraton oder beim Timing Diagnose Tool vorfindet, wurde um die IP Adresse ergänzt. Damit kann man in Netzwerken, in denen die Namensauflösung nicht korrekt funktioniert, manuell die IP-Adresse des Zielsystems anstatt des Namens verwenden, um eine Verbindung herstellen zu können.



16 Text_File_Write Baustein

In der Vergangenheit wurde manchmal eine LogFileWrite-DLL eingesetzt, um Text-Dateien für Log-ausgaben zu schreiben.

Als Ersatz für diese DLL dient nun der **Text_File_Write** Baustein. Hinweis: Das Schreiben geschieht asynchron zur Berechnung, der Ausgang „WRITE_DONE“ gibt an, ob der letzte Schreibbefehl erfolgreich abgeschlossen wurde.



STRING_DATA: Ascii-Daten die in die Datei geschrieben werden sollen

PATH: Pfadangabe der Ausgabe-Datei

FILENAME: Name der Ausgabe-Datei

WRITE_FILE: TRUE = schreiben, d.h. bei anstehendem TRUE wird in jedem Takt geschrieben, sofern der letzte Schreibbefehl abgeschlossen wurde.

ADD_LINEFEED: An den STRING_DATA Text wird automatisch ein Zeilenumbruch angehängt. Damit wird jeder Eintrag eine neue Zeile.

OVERWRITE_FILE: FALSE = Alle Daten werden in der Text-Datei angehängt an den bestehenden Inhalt. TRUE = Es wird nur der letzte Text in die Datei geschrieben, da sie komplett überschrieben wird.

WRITE_DONE: TRUE wenn Datei geschrieben

LAST_ERROR: Fehlemeldungsnummer in Hex

ERROR_STRING: Fehlermeldung als Klartext