Neue Funktionen in ibaLogic v5.6.0

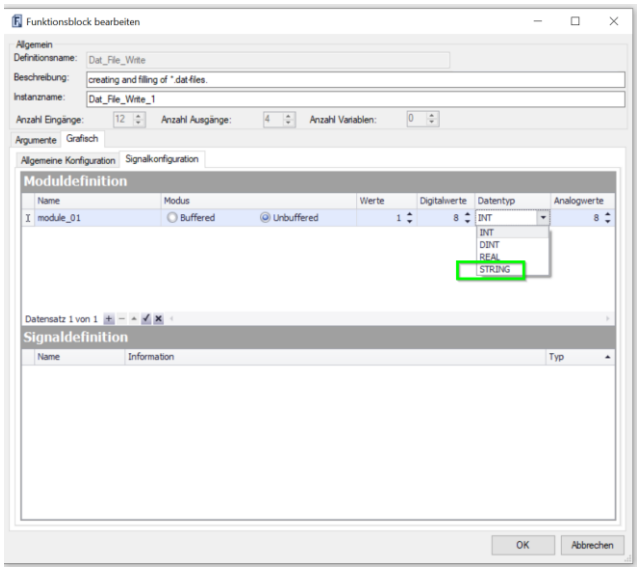
Autor: iba AG Fürth

Datum: 18.10.2021

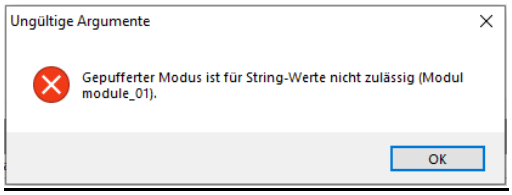
Inhaltsverzeichnis

1 DatFileWrite mit Textkanälen

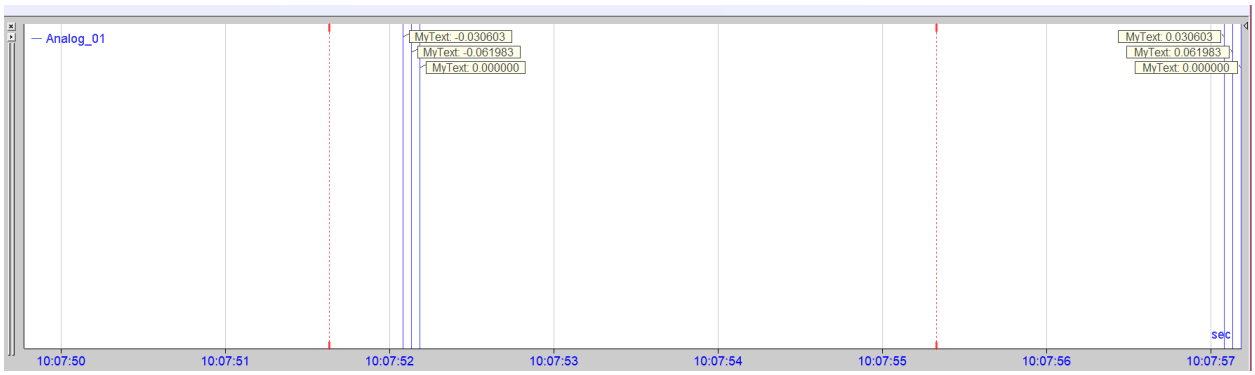
Der DatFileWrite Baustein hat jetzt auch die Möglichkeit Textkanäle zu speichern.
Dazu wurde die Auswahl an Datentypen erweitert:



Achtung: Dies ist nur für ungepufferte Werte möglich. Bei gepufferten Signalen ist dies nicht erlaubt und man erhält eine entsprechende Meldung.



Im ibaAnalyzer kann dann eine Textausgabe so aussehen. Es wird immer bei einer Änderung des Textes ein entsprechendes Signal angezeigt.



2 Passwordgeschütze Funktionsbausteine(FBs) und Makros (MBs)

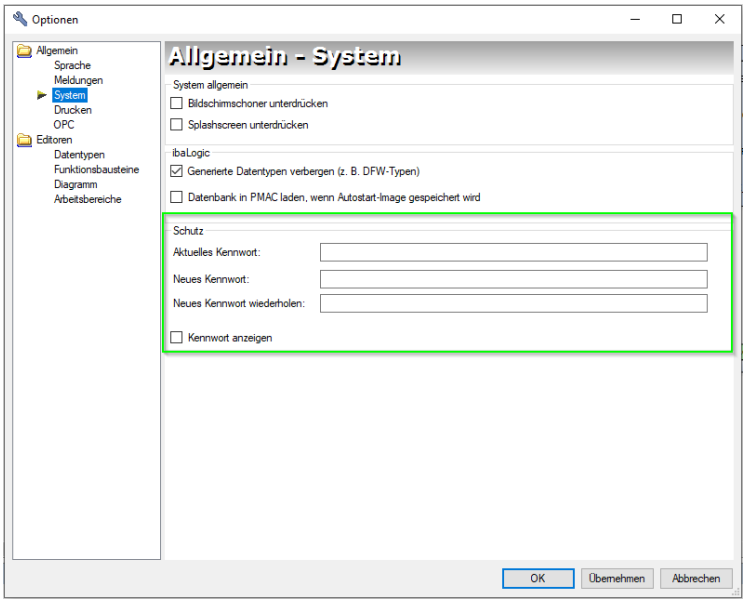
Um dem Inhalt eigener FBs und MBs zu schützen, bietet jetzt ibaLogic die Möglichkeit einen Passwortschutz anzulegen.

Dieses Passwort ist innerhalb eines Arbeitsbereiches gültig.

Bei geschützen Bausteinen ist nur der Interface-Bereich (Ein/Ausgänge) sichtbar, der Programmteil aber nicht mehr. Bei einem MB kann diesen Inhalt nicht geöffnet werden.

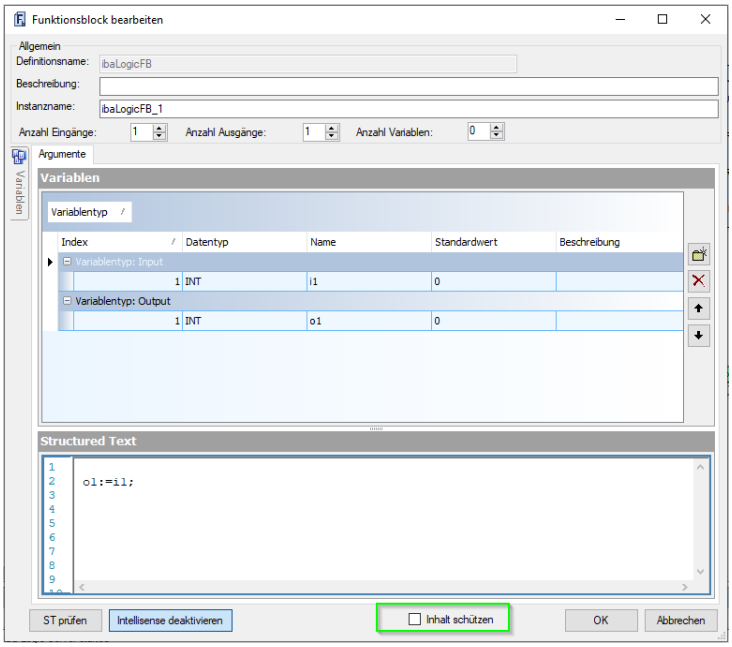
Ein geschützer Baustein (Instanz und Definition) kann aber jederzeit ohne Abfrage des Passwortes gelöscht werden.

Zum Aktivieren des Schutzes geht man zu KONFIGURATION-OPTIONEN-SYSTEM. Dort ist es möglich ein Passwort für den Schutz einzutragen.

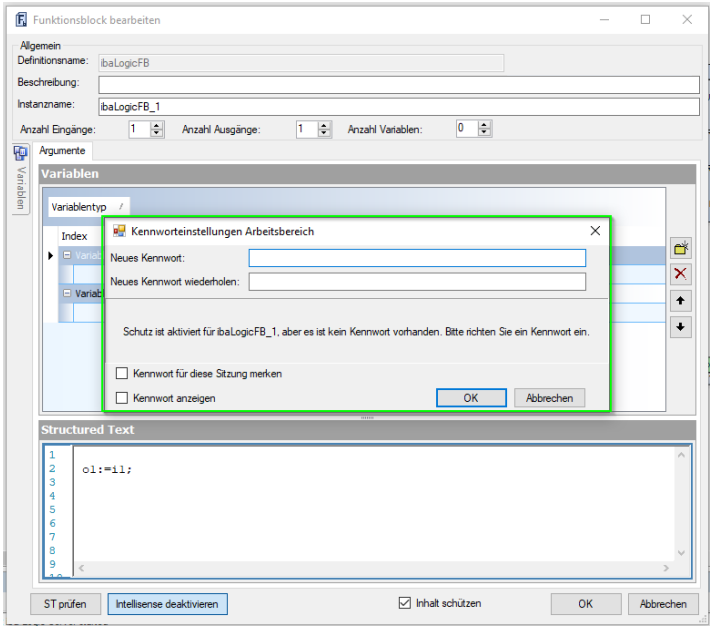


Zum Deaktivieren wird das aktuelle Kennwort eingetragen und das neue leer gelassen.

In jedem FB oder Makro gibt es die Möglichkeit diesen zu schützen. Dabei bezieht sich dann der Schutz immer auf die Definition d.h. damit sind auch alle Instanzen dieses speziellen FBs/Makros geschützt.

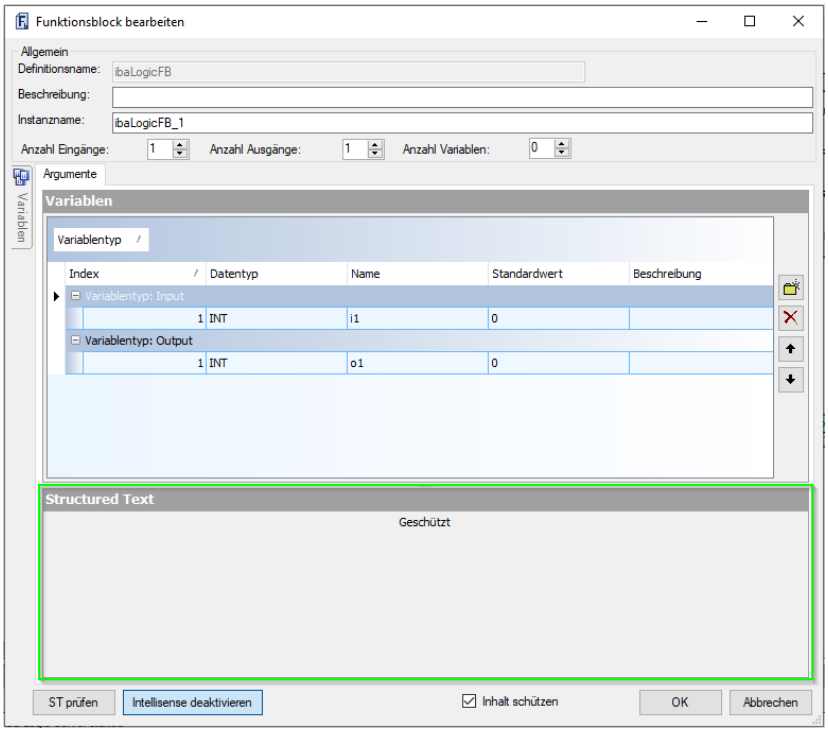


Wählt man dies an und hat aber bisher noch kein Passwort vergeben, so kommt eine Aufforderung ein Passwort anzugeben.



Ebenso kann man entscheiden, ob das Passwort für die gesamte Session gemerkt werden soll. Dies bedeutet, dass man, solange man den Client oder den Arbeitsbereich nicht schließt, alle Passwort-geschützten Bausteine bearbeiten kann.

Ein geschützer FB sieht so aus:

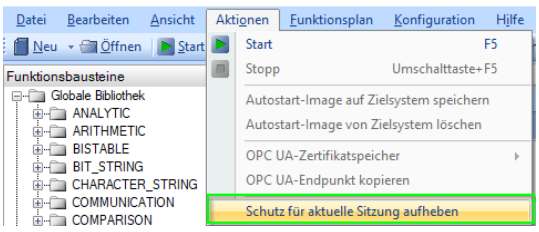


Der Inhalt ist nicht mehr zu sehen.

Bei einem geschützten Makro kommt man nicht mehr auf den internen Makro-Plan.
Schützt man einen Makro, werden auch eventuelle offene Makro-Inhalte (eigene Programm Tabs) geschlossen.

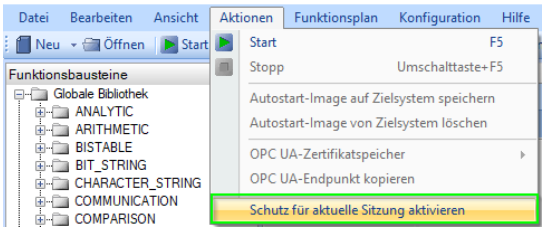
Will man den Baustein wieder bearbeiten, hat man mehrere Möglichkeiten.

- Man gibt das Passwort an und kann den Baustein bearbeiten. Beim FB schließt man den Baustein wieder, so ist der Schutz sofort wieder wirksam.
  Beim MB muss man die Lasche mit dem Makro-Inhalt schließen, damit der Makro wieder geschützt ist.
- Man nimmt den Schutz beim Baustein weg und setzt den Schutz nach der Bearbeitung wieder. Dazu ist jedesmal das Passwort einzugeben
- Eine Möglichkeit den Schutz für eine Bearbeitungs-Session generell aufzuheben, kann man über das folgende Menu erreichen:



  Es muss das Password einmalig eingegeben werden. Nun können auch alle geschützten FBs oder MBs bearbeiten werden.
  Will man diesen Schutz generell wieder einschalten, kann man wieder diesen Menü-Punkt anwählen.

Bei Anwahl ist der Schutz sofort wieder aktiviert.

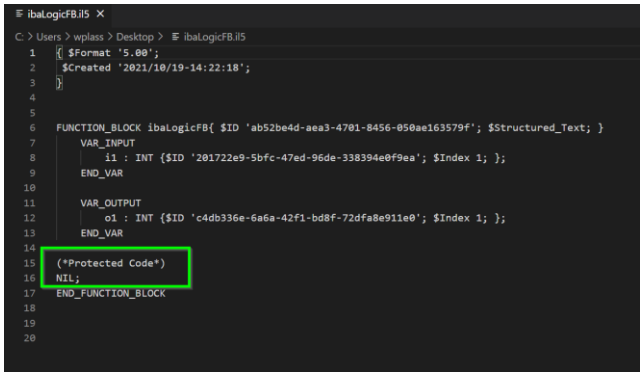


- Ebenso wird mit Schließen des Clients oder des Arbeitsbereiches die Sitzung beendet, und damit wäre beim nächsten Öffnen der Schutz wieder gegeben.

Beim Export geschützter Bausteine, bzw Programme oder Projekte, die diese Bausteine enthalten, ist zu beachten, dass diese nur komplett exportiert werden, wenn das Passwort bekannt ist.

Ein Export ohne Passwort enthält nur die offenen Inhalte. Importiert man einen solchen Export ist dieser nicht komplett lauffähig, da ihm die geschützten Inhalte fehlen.

Export-Beispiel:



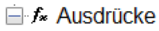
3 Playback (variable Geschwindigkeit, längenbasierte Signale)

Es wurden einige Neuerungen in das Playback-Handling eingebracht.

Seit vergangenen Versionen ist es bereits möglich, verschiedene Playback Dateien (gleicher Art) gesteuert abzuspielen. D.h. einlesen, starten, anhalten, etc

Neu hinzugekommen sind nun folgende Funktionen und Änderungen:

- Dat-Dateien, die Ausdrücke enthalten (= Signale die mit dem ibaAnalyzer ereugt wurden), stehen jetzt auch zur Verfügung und können damit abgespielt werden.

  Achtung: Dies ist für das PADU-S-IT2x16 nicht möglich

- Abspielen von längenbasierten Signalen

  Bisher waren nur Dat-Dateien zeitbasierte abspielbar. Nun kann man auch Dateien längenbasierte über ein Geschwindigkeitssignal wieder abspielen.
  Dabei ist zu beachten, dass alle längenbasierten Signale in einer Dat-Datei nur über ein einziges Geschwindigkeitssignal abspielbar sind. Sollten sich in der Datei mehrere „Längen-Messstellen“ mit jeweils ihrer eigenen Geschwindigkeit befinden, so hat man sich für eine Messstelle und deren Längesignale zu entscheiden.

- Abspielen der Playback-Dateien mit variabler Geschwindigkeit

  Damit ist es möglich z.B eine Playback Datei mit einem aktuellen Prozess zu synchronisieren, indem man das Playback langsamer oder schneller abspielt.

- Abspielen im Single-step Modus = nächster Messwert aus Dat-Datei

  Mit dieser Funktion ist es möglich, alle Messwerte nacheinander auszulesen, um Messwert-genaue Berechnungen über die Dat-Datei durchzuführen. Oder um Simulationen Messwert-genau schneller durchzuführen. Z.B. Stundenwerte / Tageswert abspielen.

Für diese Funktionen wurden die Ein-/Ausgänge des Playbacks erweitert.

Ausgänge:

Neu hinzugegekommen sind die markierten Ausgänge:

SingleStepActive: False - Dat-Datei wird normal abgespielt
True - nächster Messwert aus der Dat-Datei wird gelesen, wenn DoStep toggelt

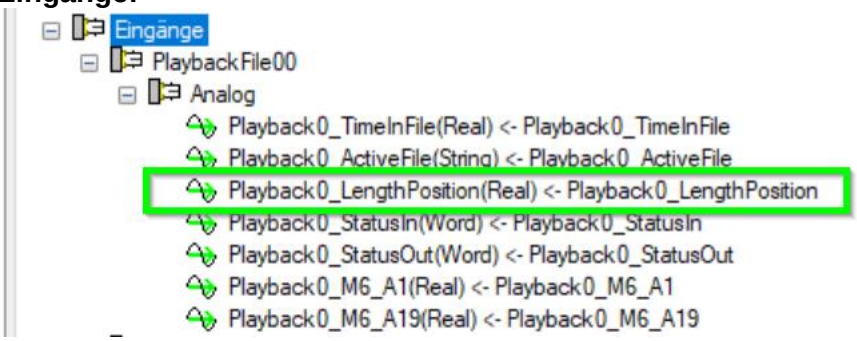
DoStep: Toggle bringt nächsten Messwert.
True - nächster Messwert aus der Dat-Datei wird gelesen, wenn DoStep toggelt. 1.Toggle nach einem Restart-Signal bringt Wert an Stelle 0 !

SpeedFactor: 1 = normales Abspielen aller zeitbasierten Signale,d.h. 1fache Geschwindigkeit. Andere Faktoren entsprechend.

LengthSpeed: zum zeitrichtigen Abspielen von längenbasierten Signalen wird hier das passende Geschwindigkeitssignal (oder eine statische Geschwindigkeit) angegeben

Anmerkung: beim SpeedFactor und beim LenghtSpeed gilt, dass eine 0 beim Start intern auf den Faktor 1 gesetzt wird. Wird im laufenden Programm eine 0 angelegt, bleibt der letzte Wert intern bestehen.

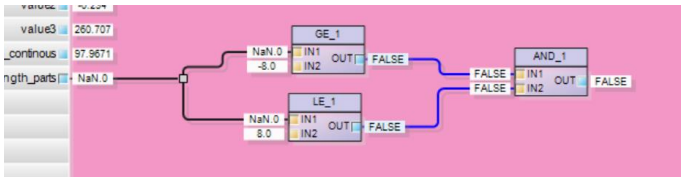
Eingänge:



LengthPosition: aktuelle Längenpositon beim Abspielen längenbasierter Signale

Die Parametrierung für die einzelnen Fälle:

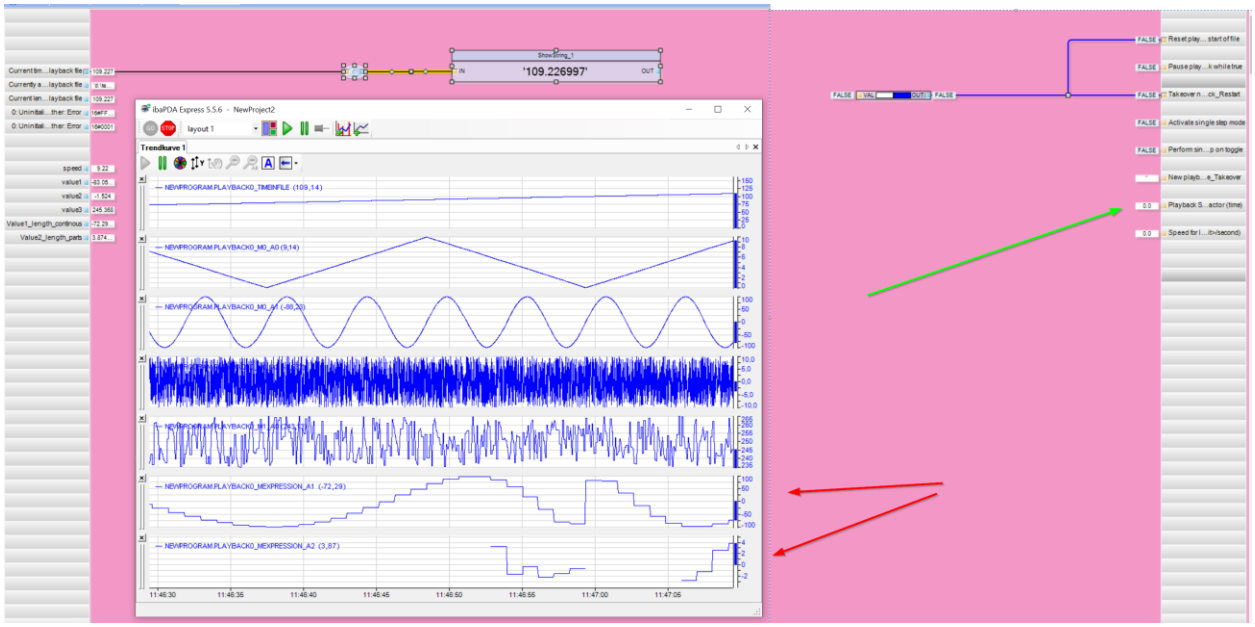
Für das Abspielen von zeitbasierten oder längenbasierten Signalen innerhalb einer Dat-Datei ist zu beachten, dass diese jeweils ihre eigene unabhängige Parametrierung habe.
Dabei kann es sein, dass z.B. die zeitbasierten Signale bereits abgespielt sind, während die längenbasierten Signale dies noch nicht sind. In diesem Fall würde bei den zeitbasierten Signale kein Messwert zurückgegeben (NaN = not a number). Ebenso wenn Lücken im Signal vorhanden sind.
Will man diese Lücken abfragen, so geht dies nur über eine Abfrage, ob das Signal im zu erwartenden Bereich liegt.



Anm: Bei INTEGER Werten wird derzeit aus internen Gründen eine 0 stat NaN zurückgegeben.

Hier Beispiele für die neuen Funktionen

<u>Beispiel1: Einfaches Abspielen der Signale (zeitbasierte und längenbasierte Signale vorhanden)</u>



Es gibt keinen Speedfaktor (grüner Pfeil), daher wird die Dat-Datei in ihrem originalen Zeitverhalten abgespielt. Interner Speedfaktor ist dann 1. Jeder anderer Faktor beeinflusst die Abspielgeschwindigkeit z.B. 2 = 2fache Geschwindigkeit, 0.5 = halbe Geschwindigkeit etc.

Die längenbasierten Signale werden ebenfalls mit einem internen Längenfaktor von 1 abgespielt. (rote Pfeile). Dabei werden auch die lückenhaften Längensignale richtig abgespielt.

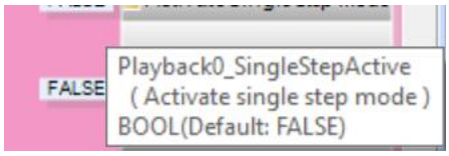
In den Lücken hat das untere Längensignal den Wert 

Will man das längenbasierte Signal wieder zeitrichtig abspielen, so muss dazu das zeitbasierte Geschwindigkeitssignal, welches der Längenumrechnung zu Grunde lag, im Dat-File enthalten sein. Dann kann man dieses Signal direkt auf den LengthSpeed Ausgang geben. Über einen multiplizierten Faktor kann man es dann auch schneller und langsamer laufen lassen.

Will man den Speedfaktor zusammen mit längenbasierten Signalen nutzen, um das Abspielen zu beschleunigen oder zu verlangsamen, muss der LengthSpeed Faktor mit dem Speedfaktor multipliziert werden, damit das Verhältnis beibehalten wird.

Beispiel2: SingleStep für zeitbasierte Signale

Für den SingleStep Modus muss der Ausgang SingleStepActive auf TRUE gesetzt sein.

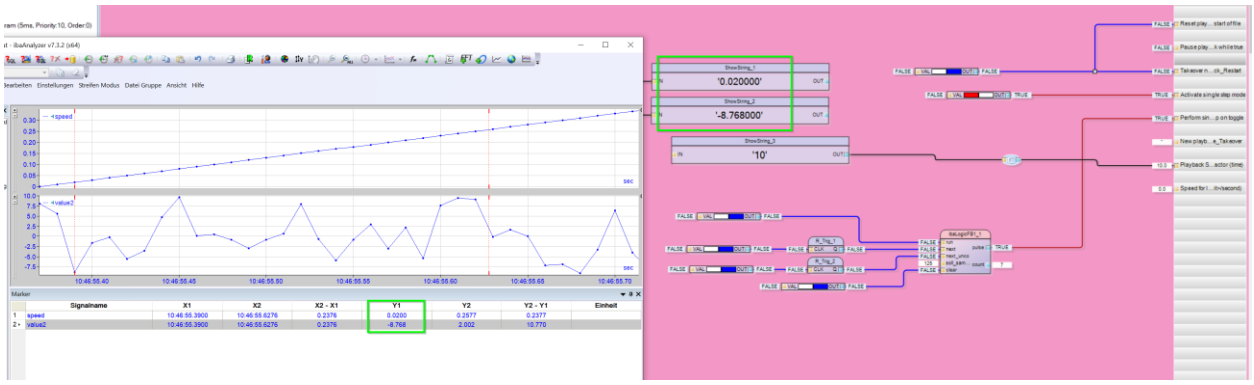


In diesem Fall bewirkt eine Wert-Änderung am Ausgang DoStep, dass der nächste Messwert angefordert wird als Signaleingang zur Verfügung steht.
Dabei muss ebenso der Ausgang Speedfaktor den gewünschten Abstand enthalten.
Der Speedfaktor berechnet sich aus dem ibaLogic Zeitbasis in der I/O-Konfiguration und der Abtastrate des Signals.

Speedfaktor = Abtastrate in ms / Zeitbasis in ms

Beispiele:
Abtastrate 10ms Zeitbasis 1ms → Speedfaktor = 10
Abtastrate 10ms Zeitbasis 5ms → Speedfaktor = 2

Beispiel: In der Dat-Datei sind 50ms und 100ms Signale. Man kann nun entweder die Messwerte der 50ms Signale nacheinander abfragen oder die der 100ms Signale.
Will man die 50ms Signale legt man am Ausgang Speedfaktor den Wert 0.05 an.
Mit jedem Toggle bekommt man den nächsten Wert. Die aktuelle Zeit im DatFile bekommt man auch im Eingang TimeInFile zurückgemeldet.
(Die 100ms Signale haben in diesem Fall den letzten 100ms Wert)



Hier sind es 10ms Werte bei einem Grundtakt von 1ms, die abgefragt werden. Pro Toggle wird in der Dat-Datei 10ms vorgegangen und der passende Wert gefunden und ausgegeben.
(Anm: der Baustein im Layout dient nur dazu pro Tastendruck einen Toggle zu machen bzw bis automatisch „durchzutoggeln“ bis man bei einem eingestellten Takt ist)

Beispiel3: Singelstep für längenbasierte Signale

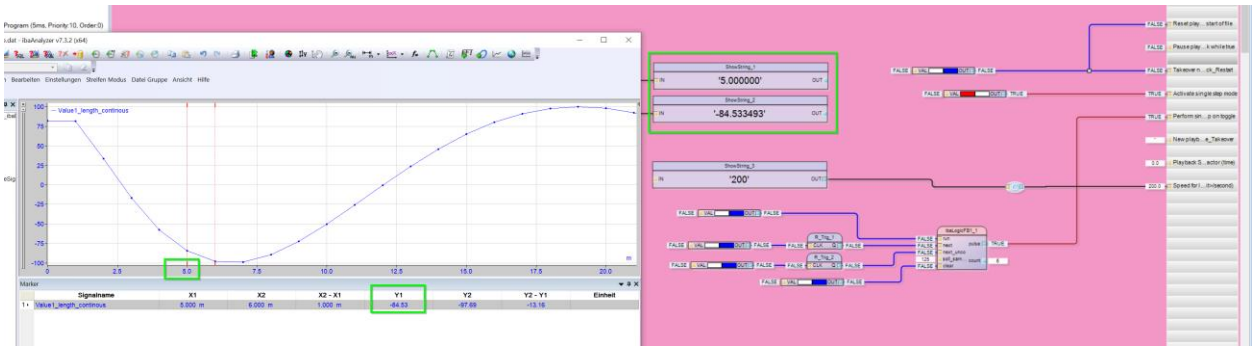
Hier gilt erstmal das gleiche wie im Beispiel 2.

Aber anstelle des Speedfaktors muss der Ausgang LenghtSpeed gesetzt sein.

Dieser Faktor berechnet sich aus dem ibaLogic Zeitbasis in der I/O-Konfiguration und der Längenauflösung des Signals in mm.

LengthSpeed = Längenauflösung in mm/ Zeitbasis in ms

Beispiele:
Längenauflösung 1m Zeitbasis 1ms → Speedfaktor = 1000
Längenauflösung 1m Zeitbasis 5ms → Speedfaktor = 200



Hier sind es 1m Werte bei einer Zeitbasis von 5ms, die abgefragt werden. Pro Toggle wird in der Dat-Datei 1m vorgegangen und der passende Wert gefunden und ausgegeben. Die aktuelle Meterangabe wird über den Eingang LenghtPosition angezeigt.