

imc Online FAMOS 取扱説明書



機械制御計測部・機械技術部

〒103-8284 東京都中央区八重洲 1-1-6

TEL.03-3279-0771 (直通 03-3245-1102) FAX.03-3246-0645

<http://www.toyo.co.jp/> E-mail:imc@toyo.co.jp

目次

1. 本書概要	3
2. システム概要	3
3. ソフトウェア編集	4
4. ソフトウェア上のチャンネル/変数	7
4.1. アナログ入力	7
4.2. 変数	8
4.2.1. ディスプレイ変数	8
4.2.2. バーチャルビット	13
4.3. バーチャルチャンネル	14
4.4. 出力	15
4.4.1. アナログ出力	15
4.4.2. デジタル出力	16
5. Online FAMOS	17
5.1. 制御コマンド	18
5.1.1. OnInitAll	19
5.1.2. OnAlways	19
5.1.3. OnTriggerStart	20
5.1.4. OnTriggerMeasure	20
5.1.5. OnTriggerEnd	21
5.1.6. OnTimer	22
5.2. Online FAMOS での関数入力	23
5.3. Online FAMOS でのチャンネルと単数値	27
5.3.1. 単数値のチャンネル化	28
5.3.2. チャンネルの単数値化	29
6. Online FAMOS のデバッグ	30
6.1. 変数名等の記述ミス	30
6.2. 初期値/最終値の定義忘れ	31
6.3. タイマーの止め忘れ	32
6.4. その他のケース	32

改訂履歴		
改訂年月日	版数	改訂内容
2021/04/12	A	初版
2021/04/13	B	関数入力の説明を追加。

1.本書概要

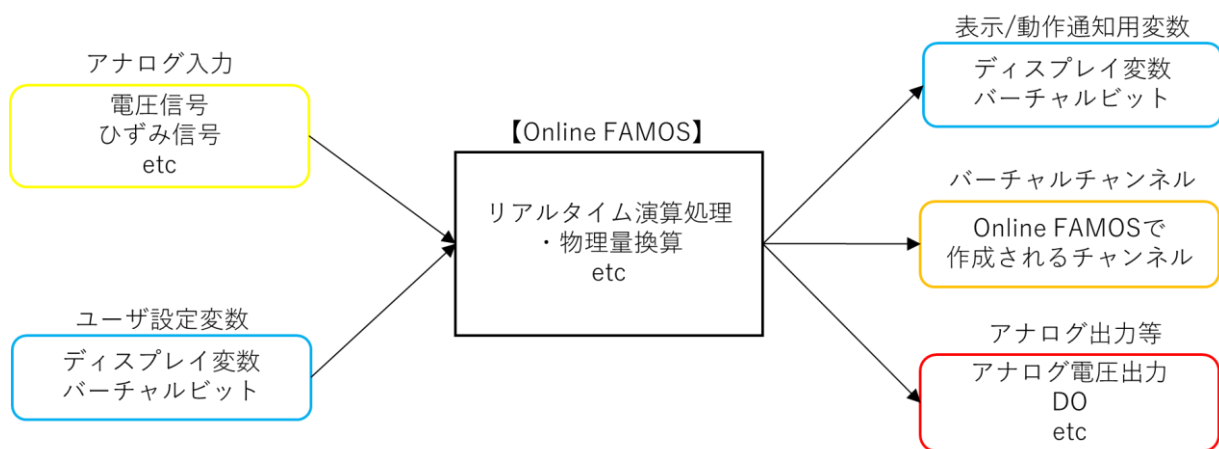
本書は、imc データロガーの Online FAMOS 機能を説明するものです。Online FAMOS 編集の参考にご利用ください。

制御ソフトとして imc STUDIO を使う前提で説明します。

2.システム概要

Online FAMOS 機能を使用する場合の、各種入出力と機能の関係を表したものが下図です。

実現する動作によっては使用しない項目も存在します。



各種入出力は、大きく分けると以下の4カテゴリに分かれます。これらの詳細は4章で説明します。

- 1) アナログ入力
- 2) 変数
- 3) バーチャルチャンネル
- 4) 出力

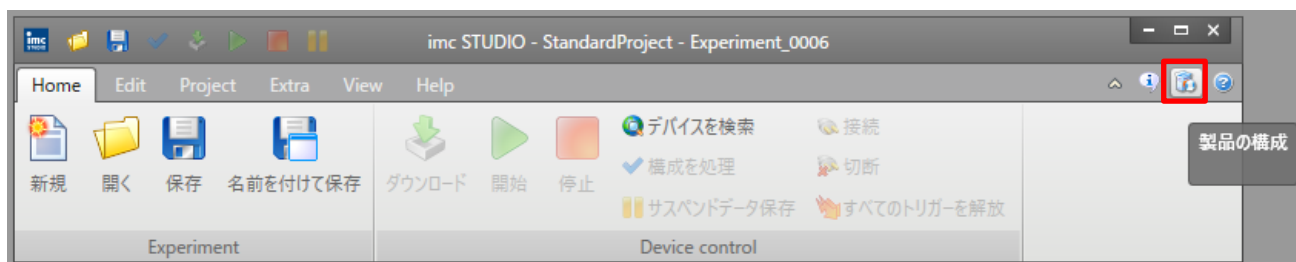
3.ソフトウェア編集

imc STUDIO では実行専用の「Runtime」というエディションが存在します。この状態はシステムの実行は可能ですが、編集機能が一部制限されています。

すべての編集機能を有効にするためには、ライセンスが有効な状態(PC にライセンスがある、dongleキーを挿入している等)で、以下の手順によりエディションを切り替えます。

Standard エディションの場合、パネルでの変数割り当てが使用できないことに注意してください。

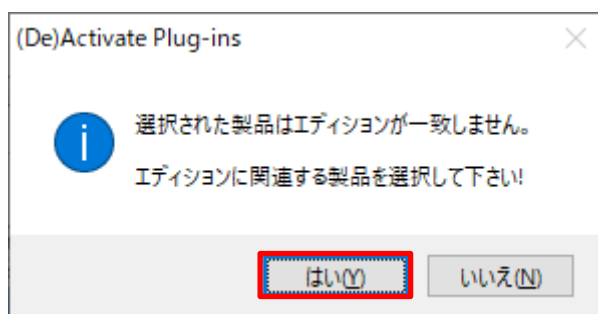
- 1) imc STUDIO を起動します。
- 2) 画面右上のアイコン[製品の構成]をクリックします。



- 3) 下図の画面で任意のエディションを選択します。



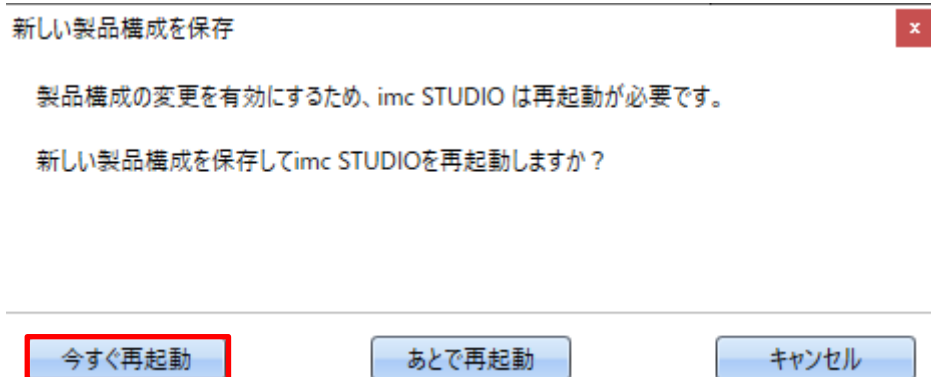
4) 下図のメッセージが表示されたら[はい]をクリックします。



5) [OK]をクリックします。



6) [今すぐ再起動]をクリックします。



7) 後は通常通り設定を読み込みます。

編集後は、必要であれば同様の手順でエディションを戻します。

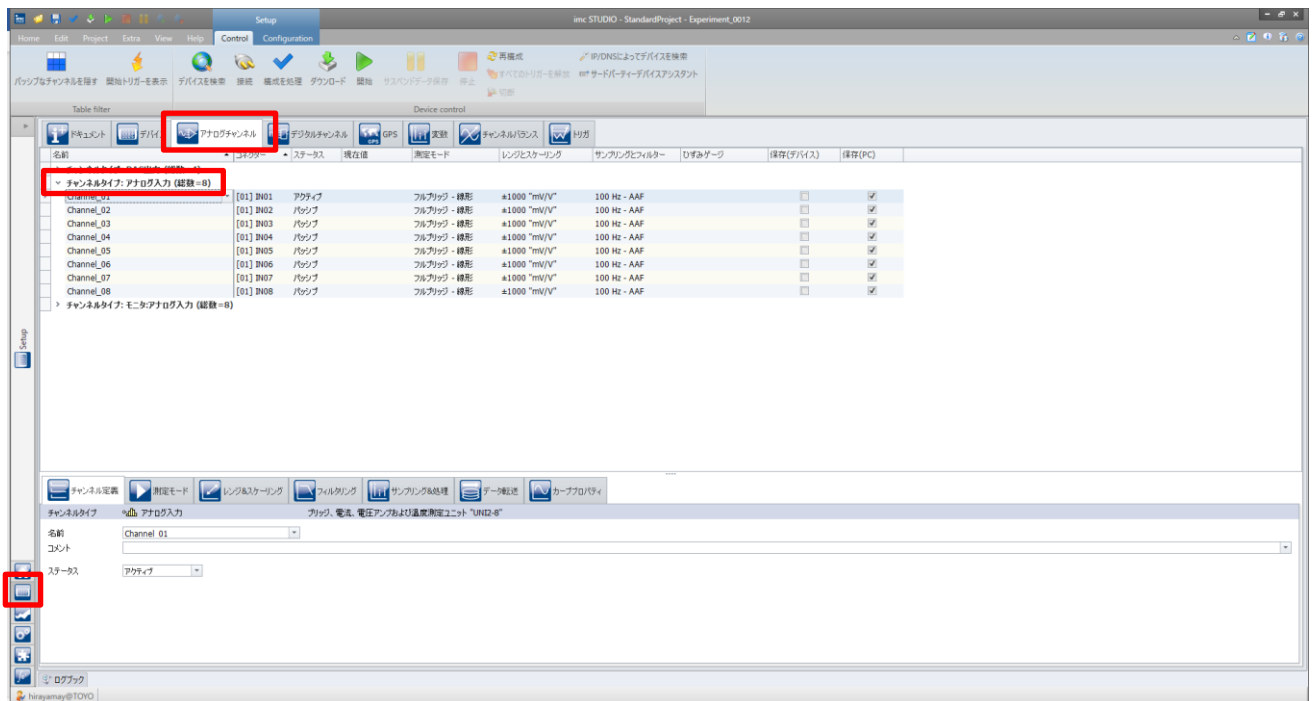
4.ソフトウェア上のチャンネル/変数

この章では本ソフトウェア上で使用されているチャンネルや変数について説明します。本ソフトウェアではこれらはすべて「名前」により管理されているため、名前の変更を行うとソフトウェアが正しく動作しなくなる可能性があります。

4.1.アナログ入力

電圧やひずみなど、計測対象となる信号です。

これらは、imc STUDIO の[Setup > アナログチャンネル > チャンネルタイプ：アナログ入力]で編集が可能です。編集項目の詳細についてはimc STUDIO のマニュアルを参照してください。



4.2.変数

Online FAMOS で基本的に利用できる変数としては 2 種類存在します。

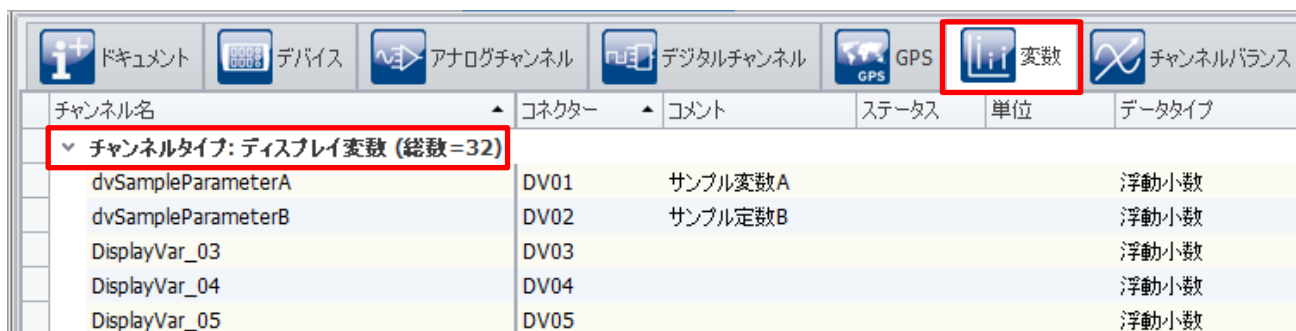
1 つは、任意の値を持つことができる「ディスプレイ変数」、もう 1 つは 0/1 の bit の値を持つ「バーチャルビット」です。

ディスプレイ変数は imc STUDIO 上表記ゆれがあり、表示変数、と表示される場合もあります。

4.2.1.ディスプレイ変数

ディスプレイ変数は imc STUDIO の[Setup > 変数 > チャンネルタイプ:ディスプレイ変数]から確認できます。最大で 32 個の変数を定義可能です。

下図の[チャンネル名]がソフトウェア上の名前で、[コメント]が内容を示しています。

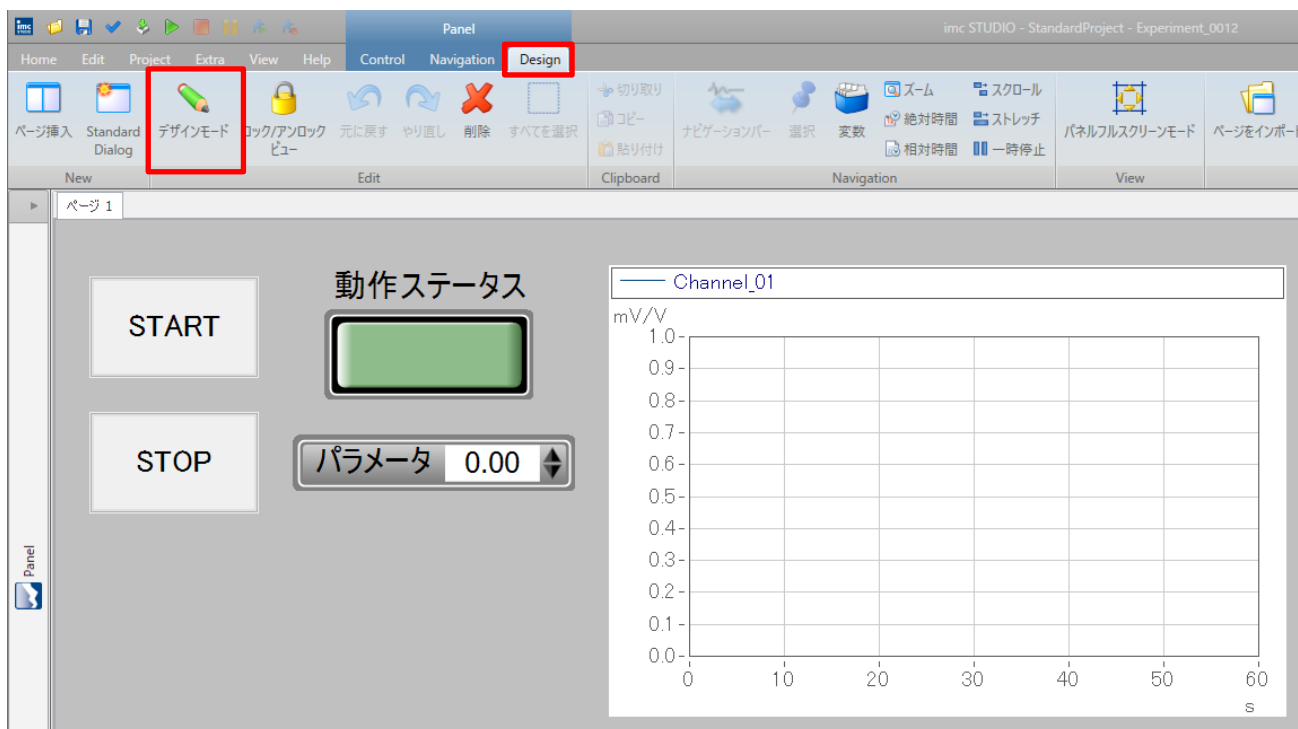


チャンネル名	コネクタ	コメント	ステータス	単位	データタイプ
▼ チャンネルタイプ: ディスプレイ変数 (総数=32)					
dvSampleParameterA	DV01	サンプル変数A			浮動小数
dvSampleParameterB	DV02	サンプル定数B			浮動小数
DisplayVar_03	DV03				浮動小数
DisplayVar_04	DV04				浮動小数
DisplayVar_05	DV05				浮動小数

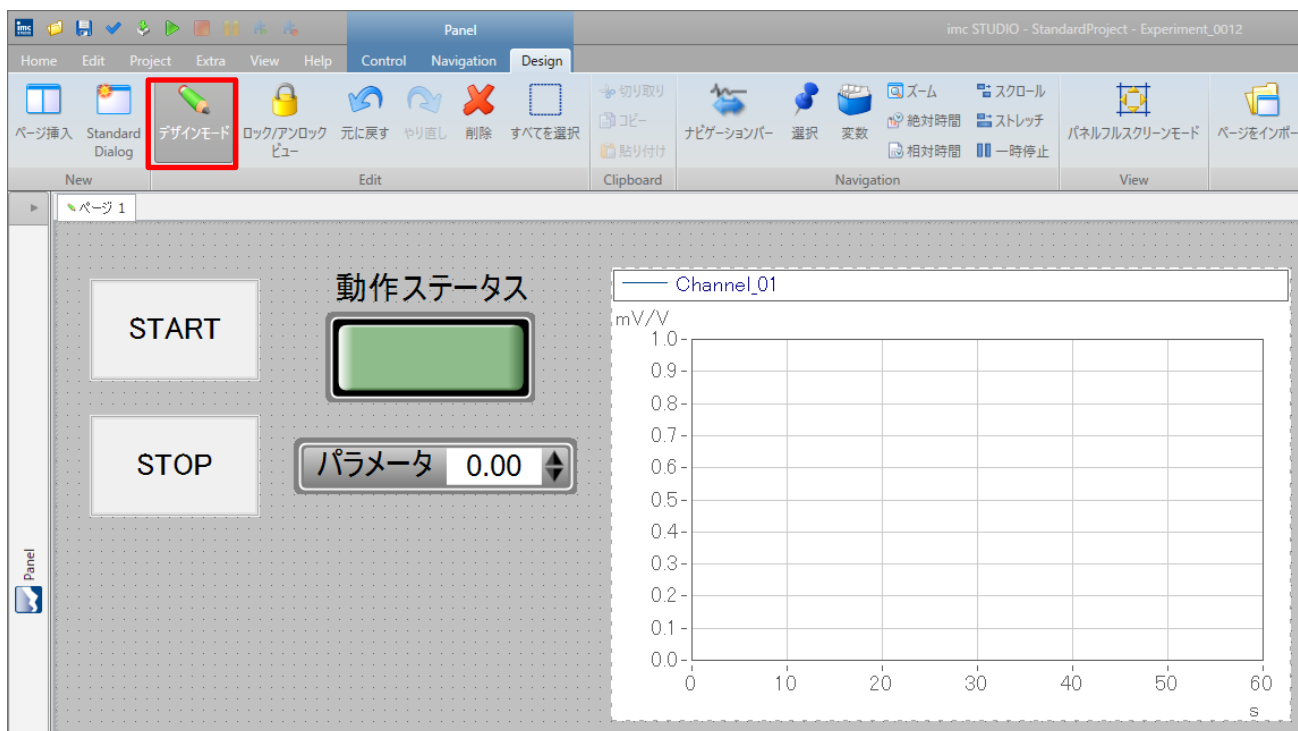
これらの値は後述する[Panel]画面経由で Online FAMOS に対して渡す、または Online FAMOS で計算した結果を表示する、という使い方ができます。

Online FAMOS の内部でのみ使用する値については、Online FAMOS それ自体で任意に作成できるため、ディスプレイ変数を使用する必要はありません。

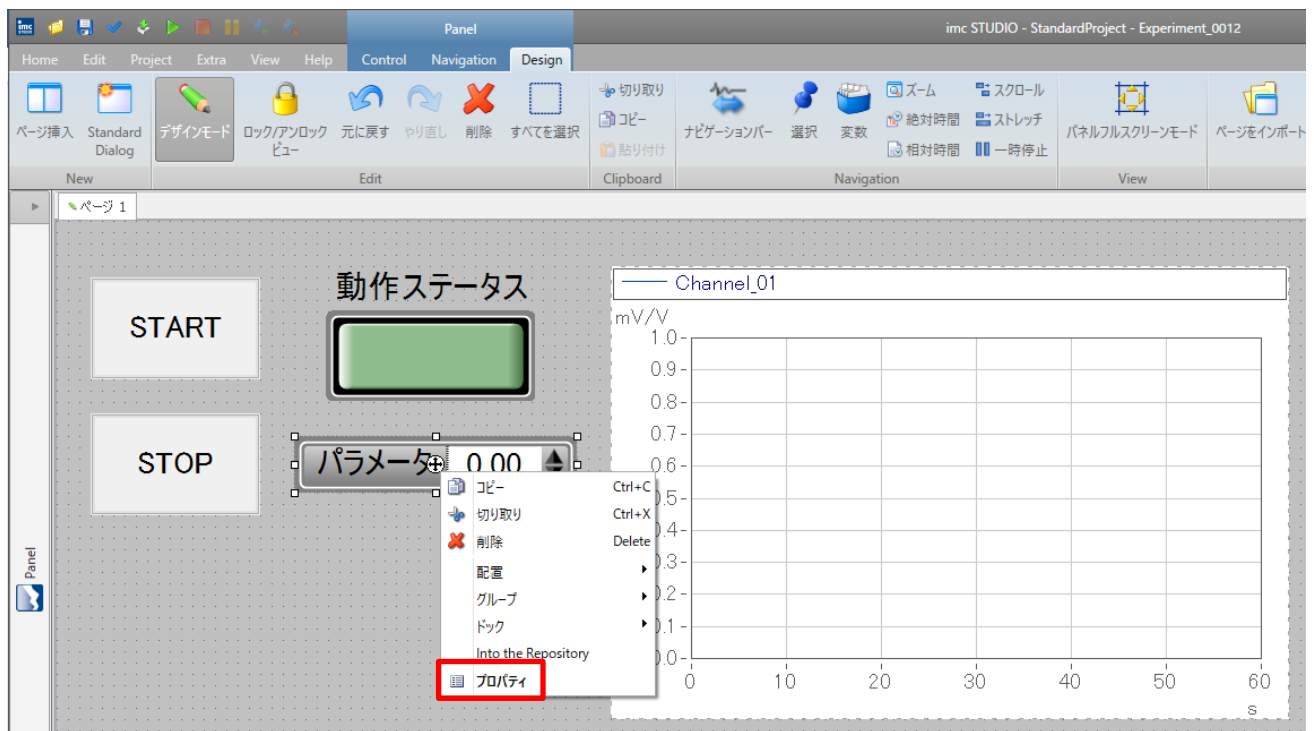
これら変数の割り当てを編集/確認するためには[Panel]画面でメニューの[Design > デザインモード]を選択してデザインモードに移行します。



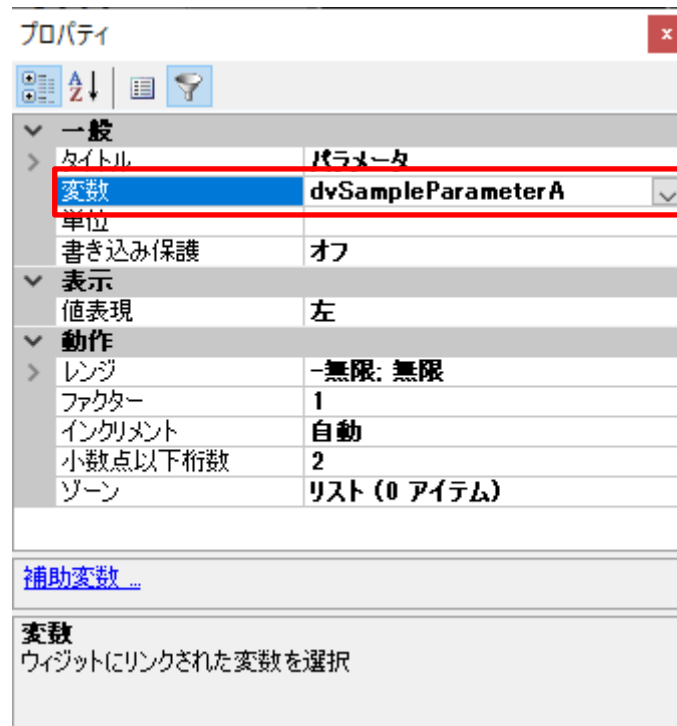
下図のようにパネル上に点線が表示され、[デザインモード]アイコンが押し込まれた状態になっていればデザインモードです。



割り当てられた変数を編集/確認するには、任意の対象を右クリックして[プロパティ]を選びます。



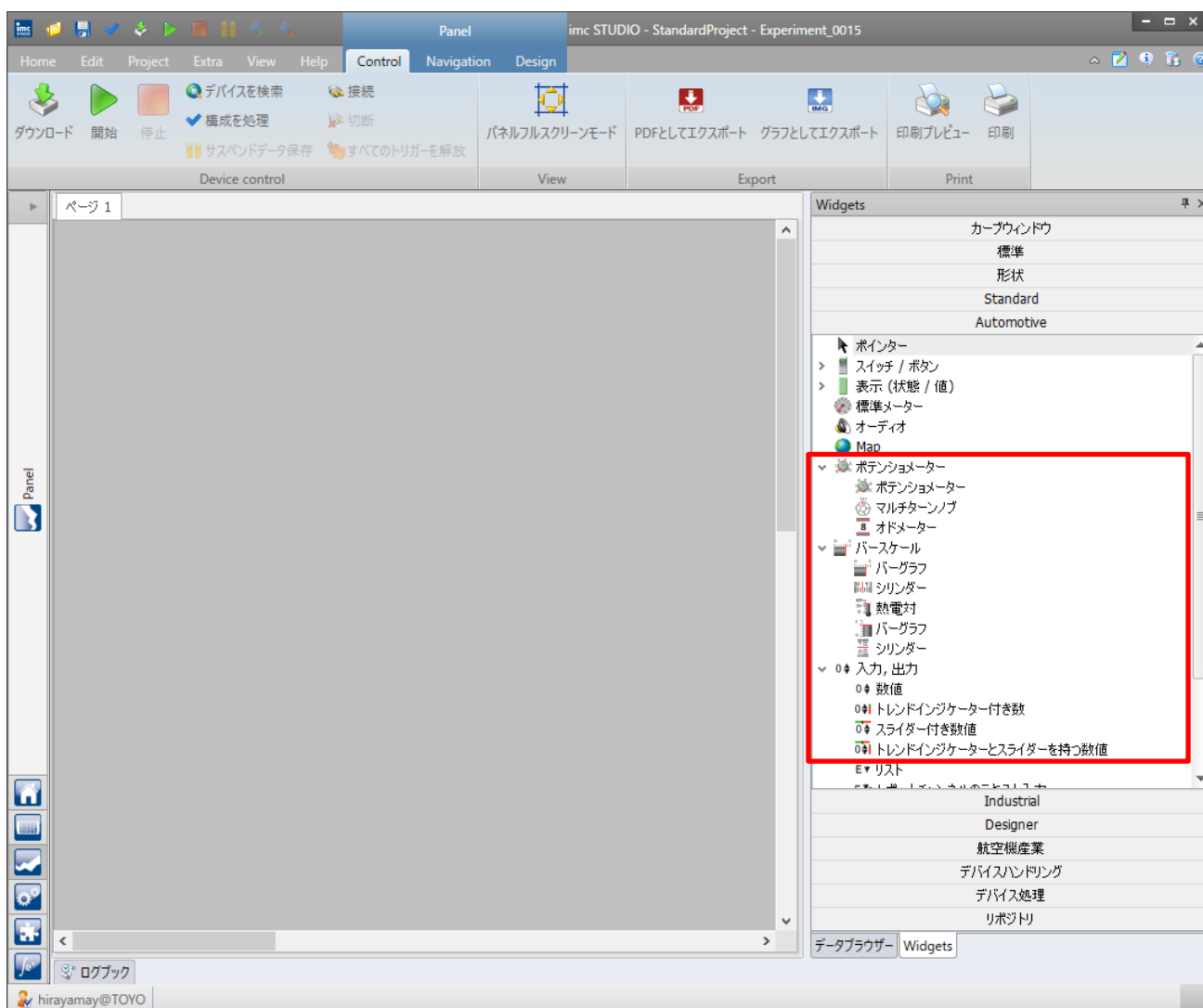
プロパティウィンドウ上の[変数]が割り当てられている変数です。右端のアイコンから変更ができます。



変更時には下図のような画面が表示されるので、任意の変数を選択します。

名前	コメント	単位
DisplayVar_19		
DisplayVar_20		
DisplayVar_21		
DisplayVar_22		
DisplayVar_23		
DisplayVar_24		
DisplayVar_25		
DisplayVar_26		
DisplayVar_27		
DisplayVar_28		
DisplayVar_29		
DisplayVar_30		
DisplayVar_31		
DisplayVar_32		
dvSampleParameterA	サンプル変数A	
dvSampleParameterB	サンプル定数B	

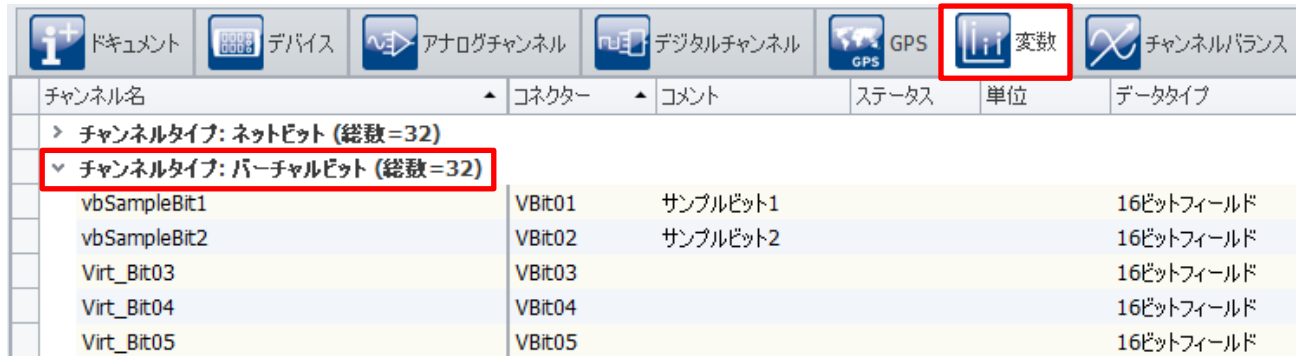
メーターや入力・出力の数値など、数値形式である様々なウィジェットに割り当てることができます。



4.2.2.バーチャルビット

バーチャルビットは imc STUDIO の[Setup > 変数 > チャンネルタイプ : バーチャルビット]から確認できます。最大で 32 個の変数が定義可能です。

下図の[チャンネル名]がソフトウェア上の名前で、[コメント]が内容を示しています

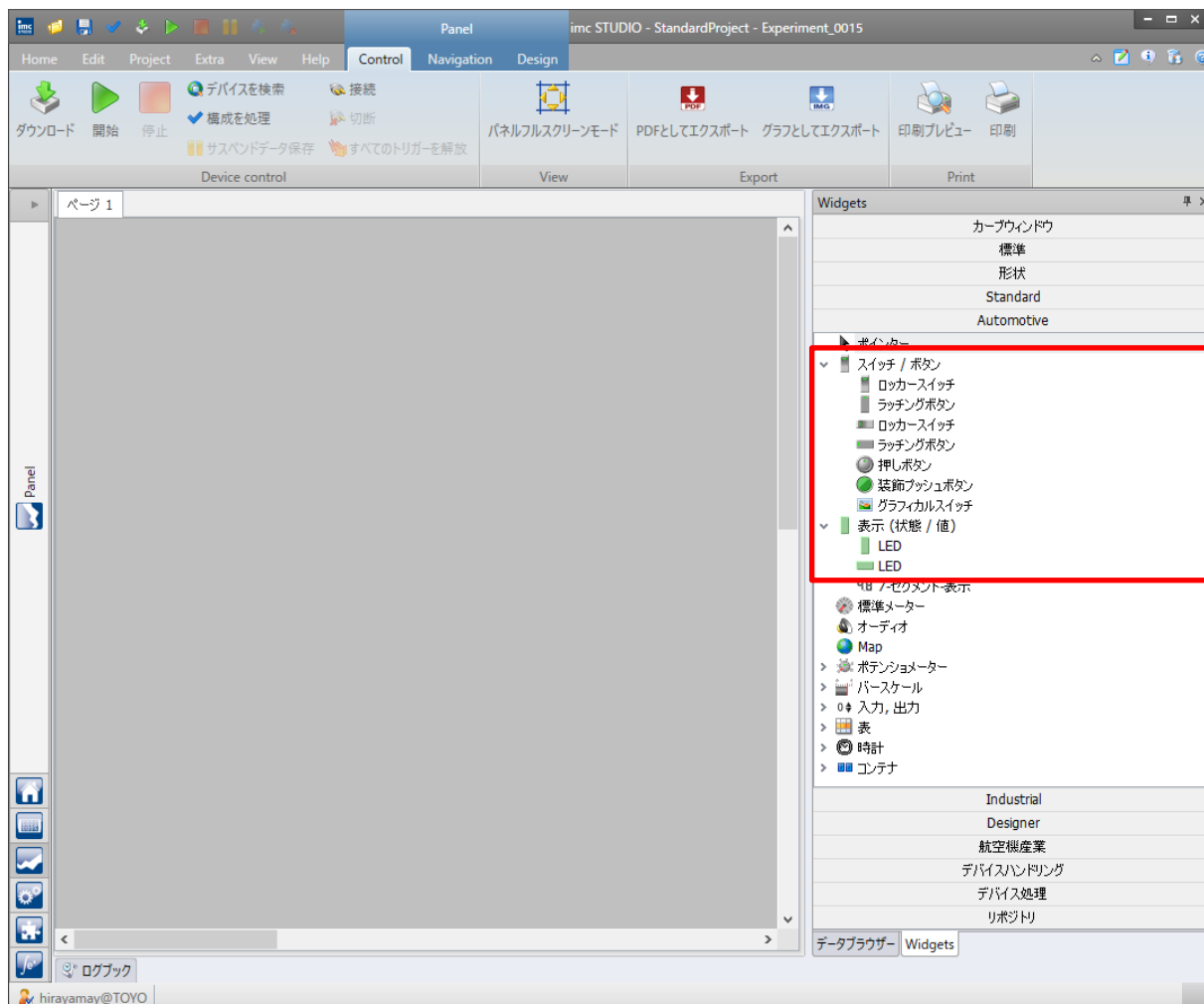


チャンネル名	コネクター	コメント	ステータス	単位	データタイプ
チャンネルタイプ: ネットビット (総数=32)					
チャンネルタイプ: バーチャルビット (総数=32)					
vbSampleBit1	VBIt01	サンプルビット1			16ビットフィールド
vbSampleBit2	VBIt02	サンプルビット2			16ビットフィールド
Virt_Bit03	VBIt03				16ビットフィールド
Virt_Bit04	VBIt04				16ビットフィールド
Virt_Bit05	VBIt05				16ビットフィールド

ディスプレイ変数と同じく、[Panel]画面経由で Online FAMOS に対して渡す、または Online FAMOS で計算した結果を表示する、という使い方ができます。

変数の割り当ての確認/編集についてもディスプレイ変数と同様です。

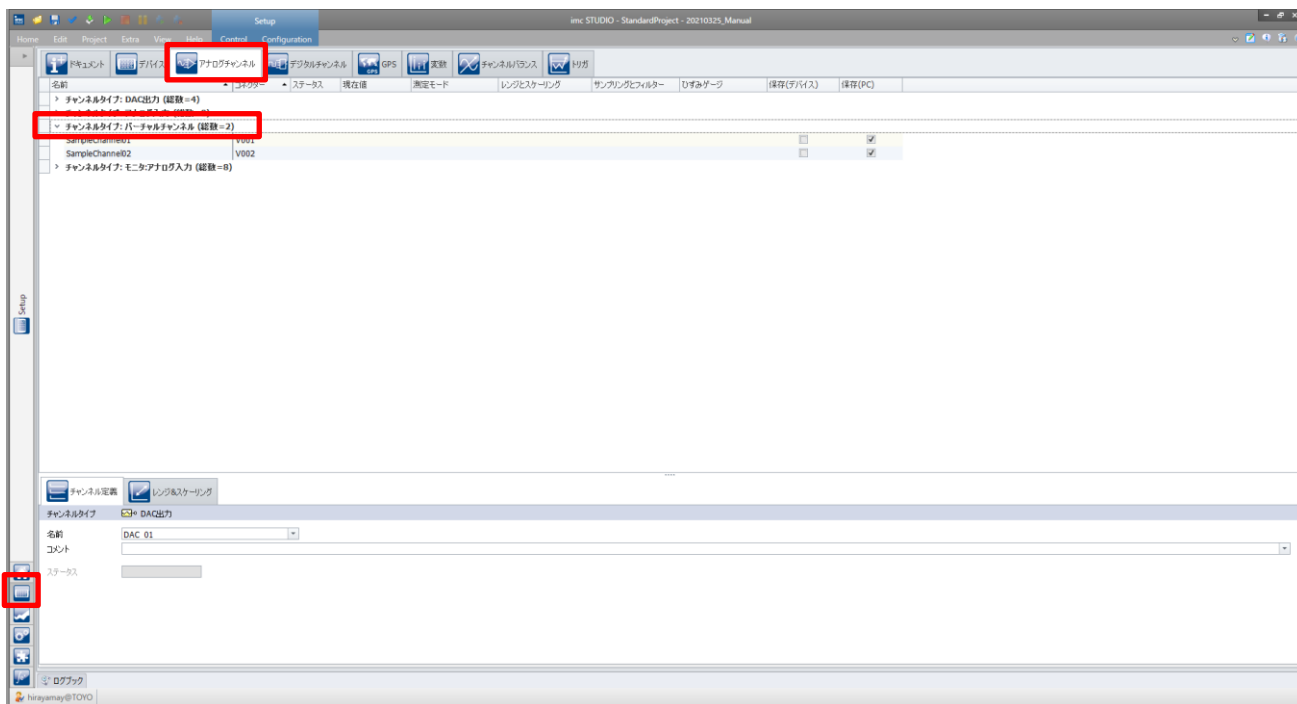
バーチャルビットは 0/1 の形式のため、主にスイッチや LED 表示などに使用します。



4.3.バーチャルチャンネル

バーチャルチャンネルは、Online FAMOS 機能により計算された仮想的なチャンネルです。

imc STUDIO の[Setup > アナログチャンネル > チャンネルタイプ : バーチャルチャンネル]から確認ができますが、ここでは基本的に内容の編集はできません。編集自体は Online FAMOS 上で行います。データの保存設定についてはこの画面で設定可能です。

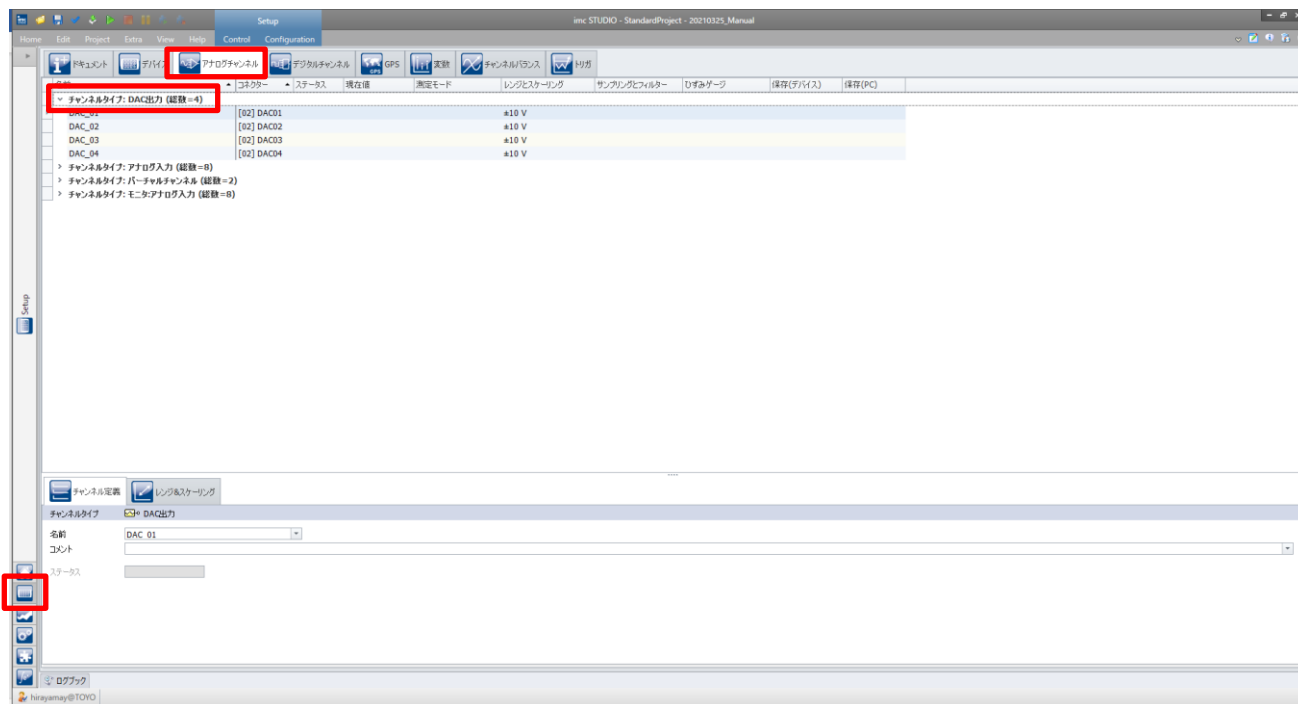


4.4.出力

4.4.1.アナログ出力

アナログ出力は、任意の電圧値を出力するチャンネルです。Online FAMOS 等から値が定義できます。

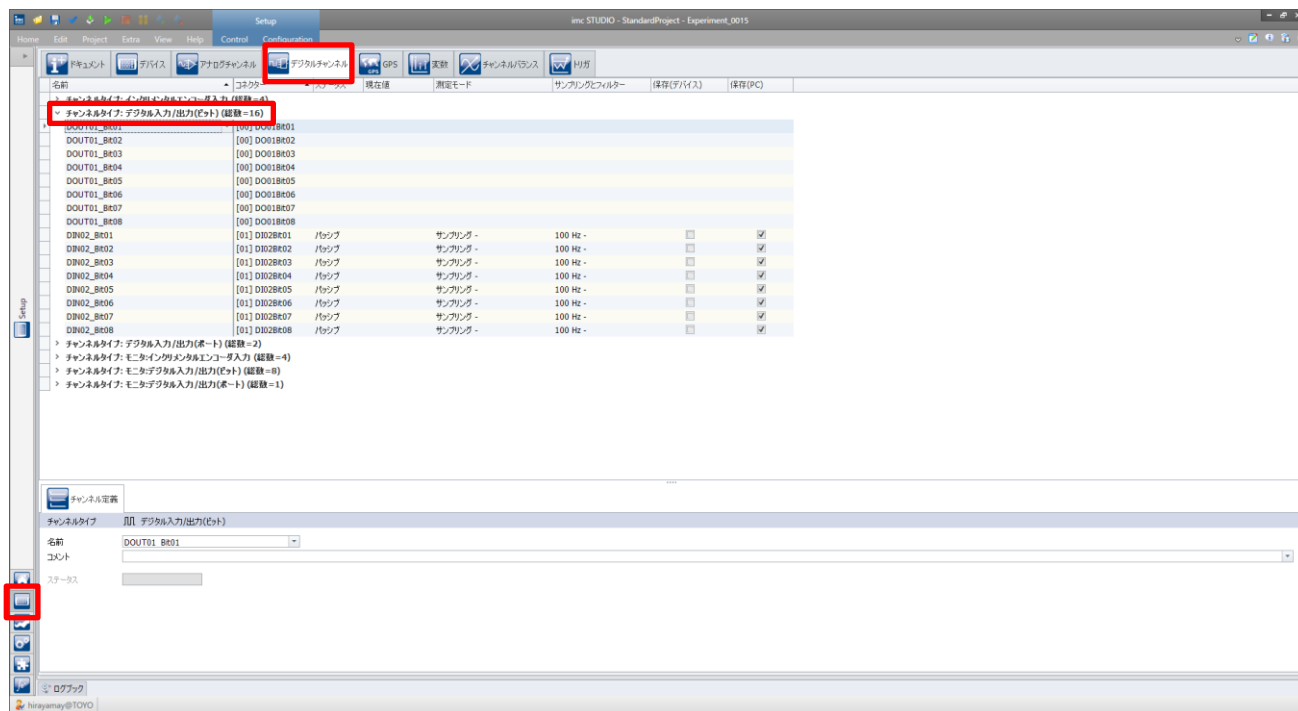
imc STUDIO の[Setup > アナログチャンネル > チャンネルタイプ : DAC 出力]から確認できます。



4.4.2.デジタル出力

デジタル出力は、0/1 の接点や規定電圧等(接続方式やモデルによる)を出力するチャンネルです。Online FAMOS 等から値が定義できます。

imc STUDIO の[Setup > アナログチャンネル > チャンネルタイプ : デジタル入力/出力(ビット)]から確認できます。



5. Online FAMOS

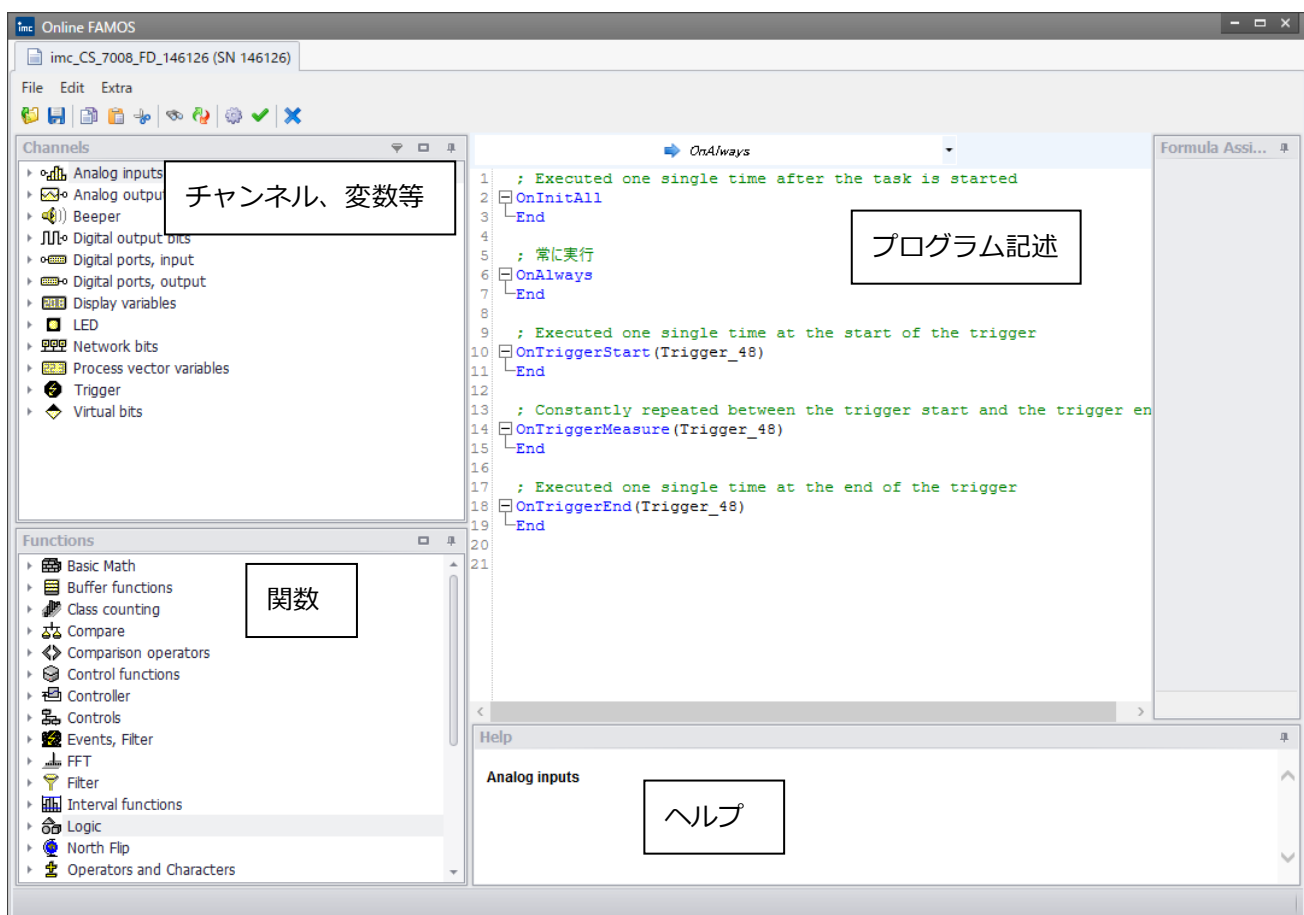
Online FAMOS は、リアルタイムの計算を実行する imc 社製データロガーの機能です。

メニューの[Home > Online FAMOS]から編集画面を開きます。



Online FAMOS の画面構成は下図のようになっています。

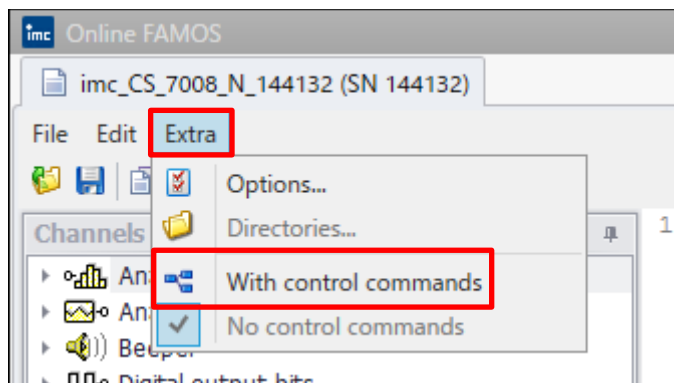
現在の imc STUDIO のバージョンでは、基本的に表示は英語です。



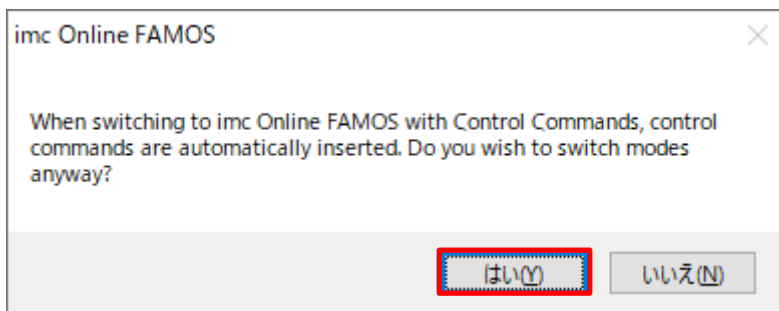
5.1.制御コマンド

Online FAMOS では、それぞれの記述がいつ実行されるのかを規定した制御コマンドが存在します。

新規に開始する場合は、メニューの[Extra > With control commands]により制御コマンドを表示します。

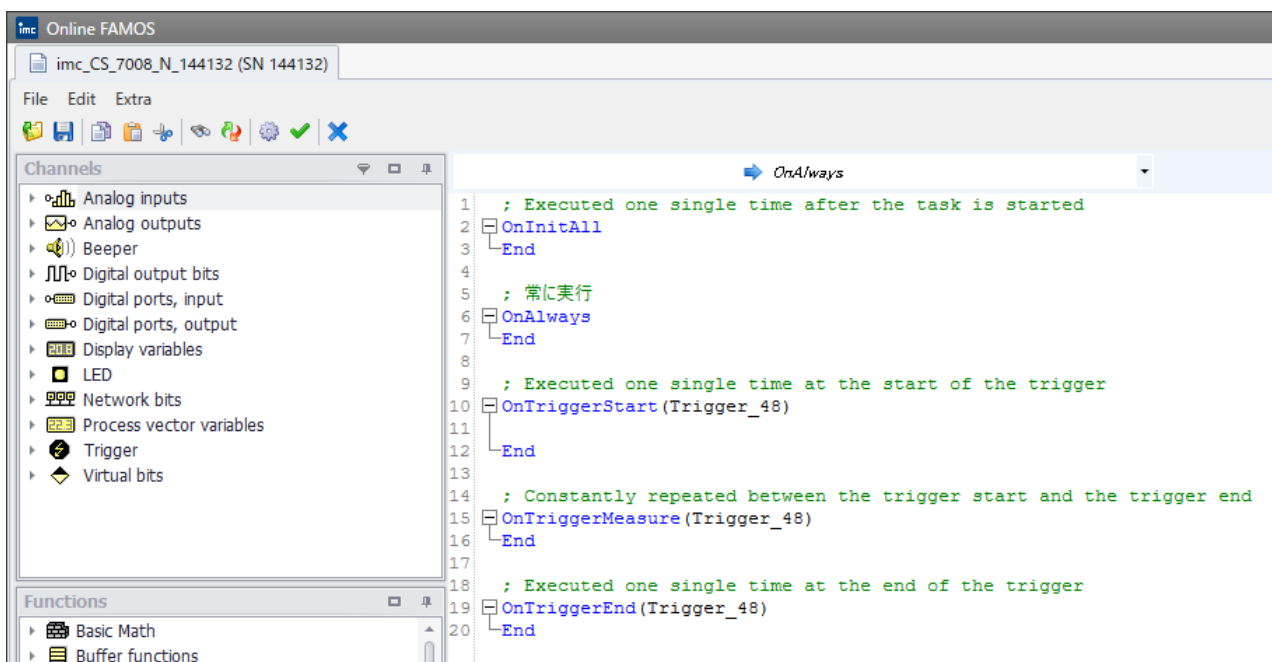


[はい]をクリックします。



各制御コマンドは、On~End という記述で区切られています。

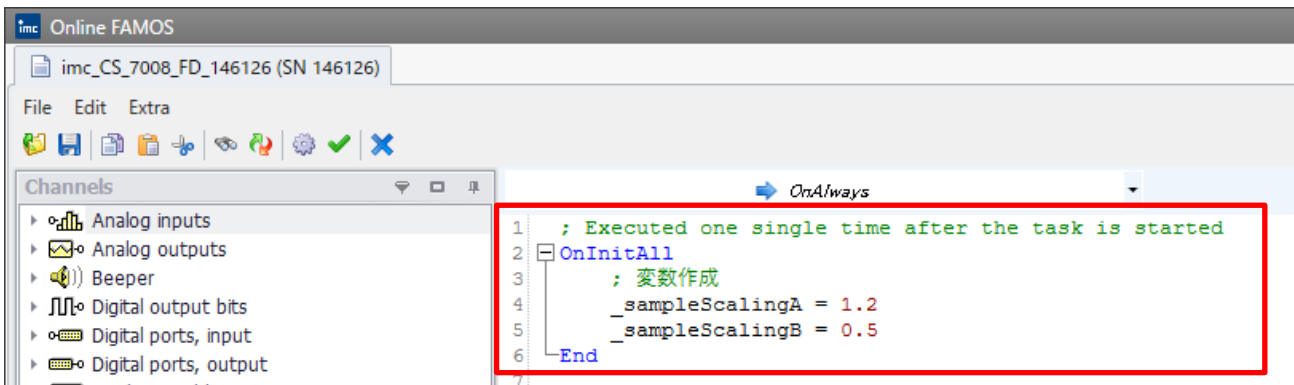
なお、;から始まる緑色の部分はコメントです。



5.1.1.OnInitAll

OnInitAll は、起動後に初めて動作開始するときに実行されるブロックです。

主に変数の初期作成を行います。



任意名称 = 任意の数値

という記述で、新しい変数を作成できます。

ここで作成されるのは、単数値のみです。チャンネルデータ(バーチャルチャンネル)を新たに作成する場合は、後述する OnTriggerMeasure ブロックを使用します。

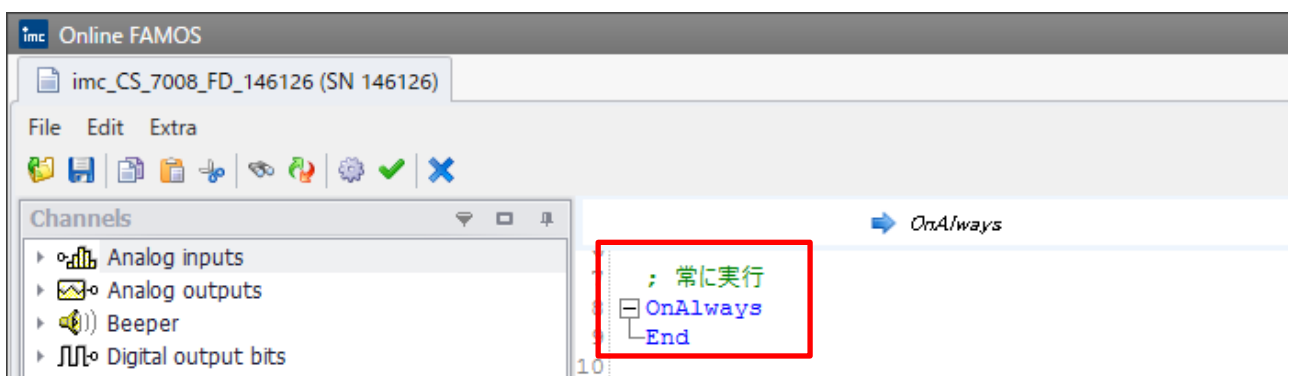
ディスプレイ変数やバーチャルビットと異なり、この値はそのままの形式では外部([Panel]画面など)に表示させることはできません。

5.1.2.OnAlways

OnAlways は、設定がデータロガーにダウンロードされた後、常に動作が実行されるブロックです。

主な使い方としては、デジタル入力に対してのデジタル出力や変数の切り替え定義などです。

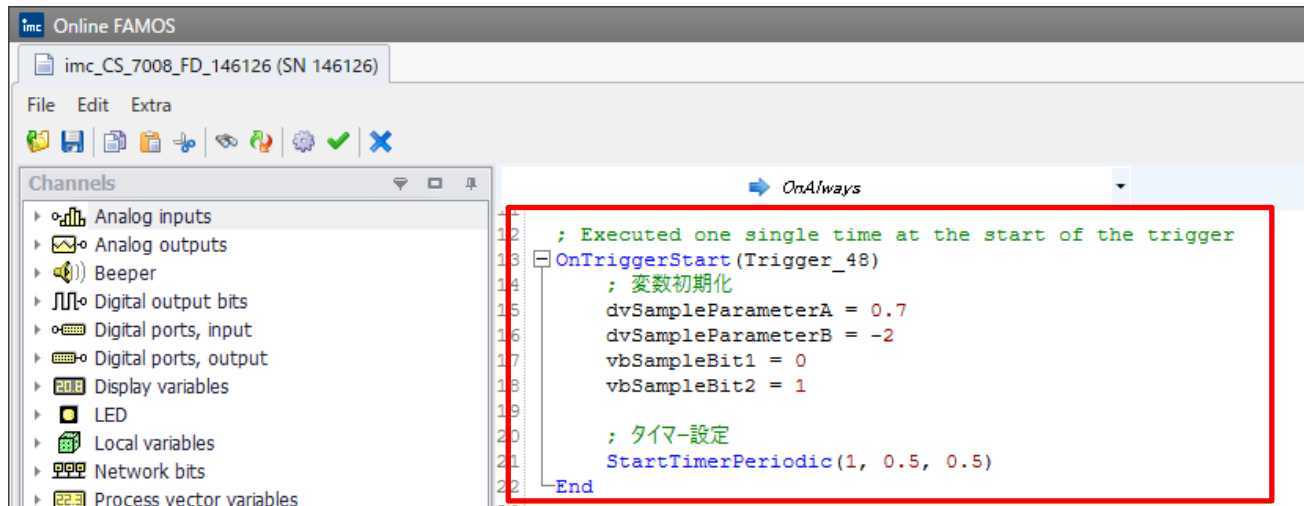
計測中の動作には別のブロックを使用します。



5.1.3.OnTriggerStart

OnTriggerStart は、計測の開始時に 1 回だけ実行されるブロックです。

主に変数の初期値の定義を行います。OnInitAll で新たに作成された変数の他、ディスプレイ変数やバーチャルビットもここで初期化できます。

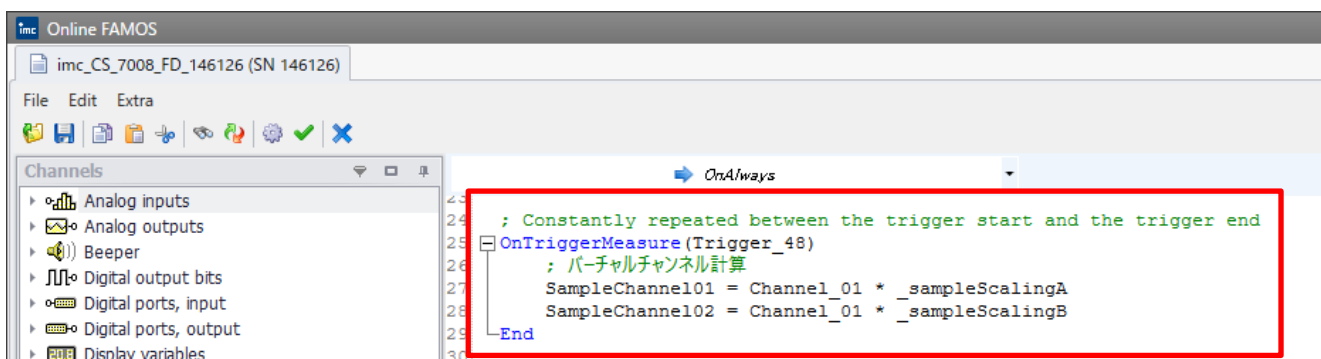


また、特殊な処理として「タイマー」の作成もここで行う場合があります。これについての動作は 5.1.6 節に後述します。

5.1.4.OnTriggerMeasure

OnTriggerMeasure は、計測実行中常に実行されるブロックです。動作周期は、使用されているチャンネルのうち最もサンプリングが早いチャンネルに依存します。

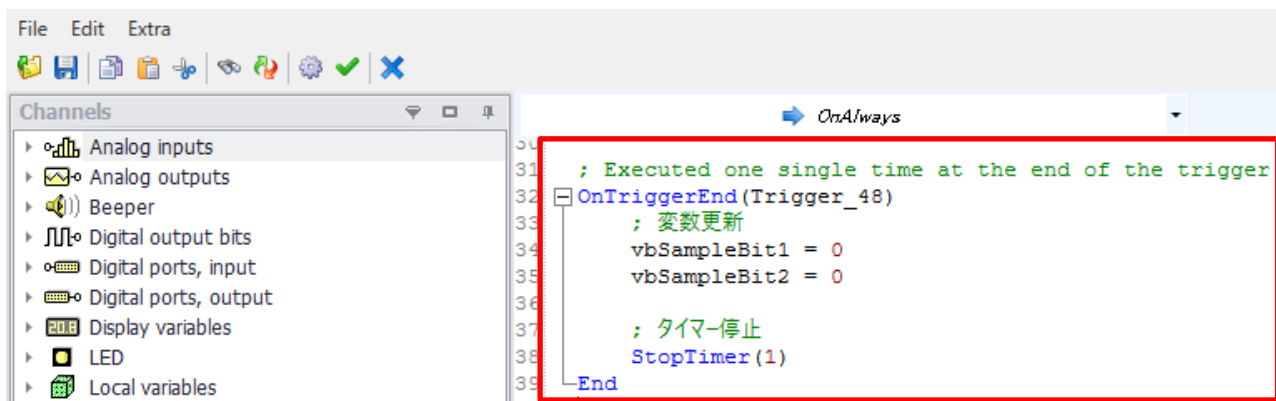
バーチャルチャンネルの作成や、アナログ入力に対しての各種計算など、ソフトウェアのメインとなる動作は主にここで記述します。



5.1.5.OnTriggerEnd

OnTriggerEnd は、計測の停止時に 1 回だけ実行されるブロックです。

主に変数の最終値の定義を行います。OnInitAll で新たに作成された変数の他、ディスプレイ変数やバーチャルビットの値も変更できます。



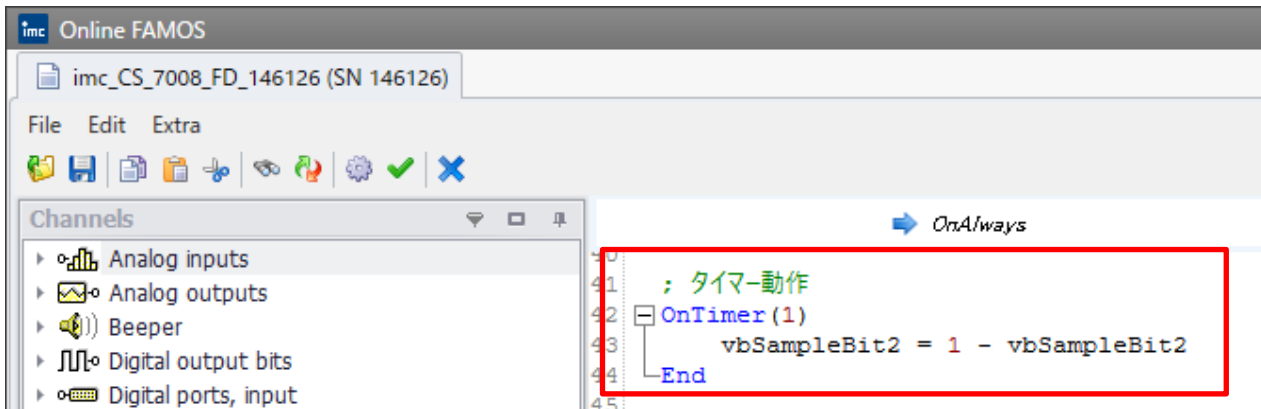
```
File Edit Extra
Channels
Analog inputs
Analog outputs
Beeper
Digital output bits
Digital ports, input
Digital ports, output
Display variables
LED
Local variables
On4ways
31 ; Executed one single time at the end of the trigger
32 OnTriggerEnd(Trigger_48)
33 ; 変数更新
34 vbSampleBit1 = 0
35 vbSampleBit2 = 0
36
37 ; タイマー停止
38 StopTimer(1)
39 End
```

「タイマー」が存在する場合は主にここで停止させます。

また、特にアナログ出力等を使用する場合は、ここで最終値(0V に戻すなど)の定義を忘れないようにします。

5.1.6.OnTimer

OnTimer は初期状態では存在しない動作ブロックで、指定のタイミング・周期で動作を実行します。



このタイマーを開始するための関数は以下の2つです。

StartTimerPeriodic(p1, p2, p3)

StartTimerSingle(p1, p2)

StartTimerPeriodic の場合、タイマーID が p1 のタイマーを p2 秒後に開始し、p3 秒毎に周期的に動作させます。

StartTimerSingle の場合、タイマーID が p1 のタイマーを p2 秒後に 1 回だけ動作させます。

ここで、タイマーID とは OnTimer の()の中の数字で、1~8 を設定できます。

OnTriggerStart ブロック以外でも任意にタイマーを作成することができます。

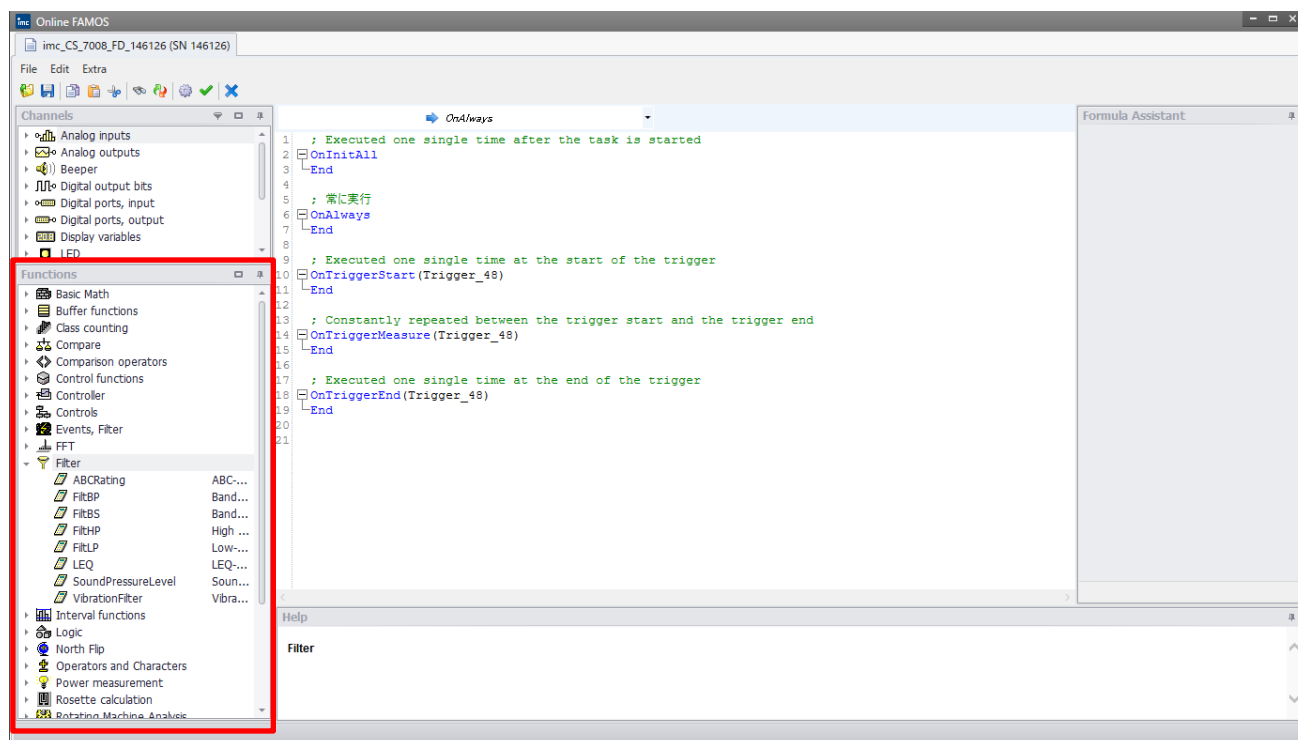
StartTimerPeriodic は停止させない限りは、測定を停止した後も動き続けるため、基本的には OnTriggerEnd ブロックで StopTimer 関数により停止させます。StopTimer の()の中の数字もタイマーID です。

なお、開始・停止はどのブロックからでも行うことができ、タイマーの中で新しくタイマーを動かす、といった操作も可能です。

5.2. Online FAMOS での関数入力

Online FAMOS では、単純な四則演算などの他、用意された様々な関数を利用できます。

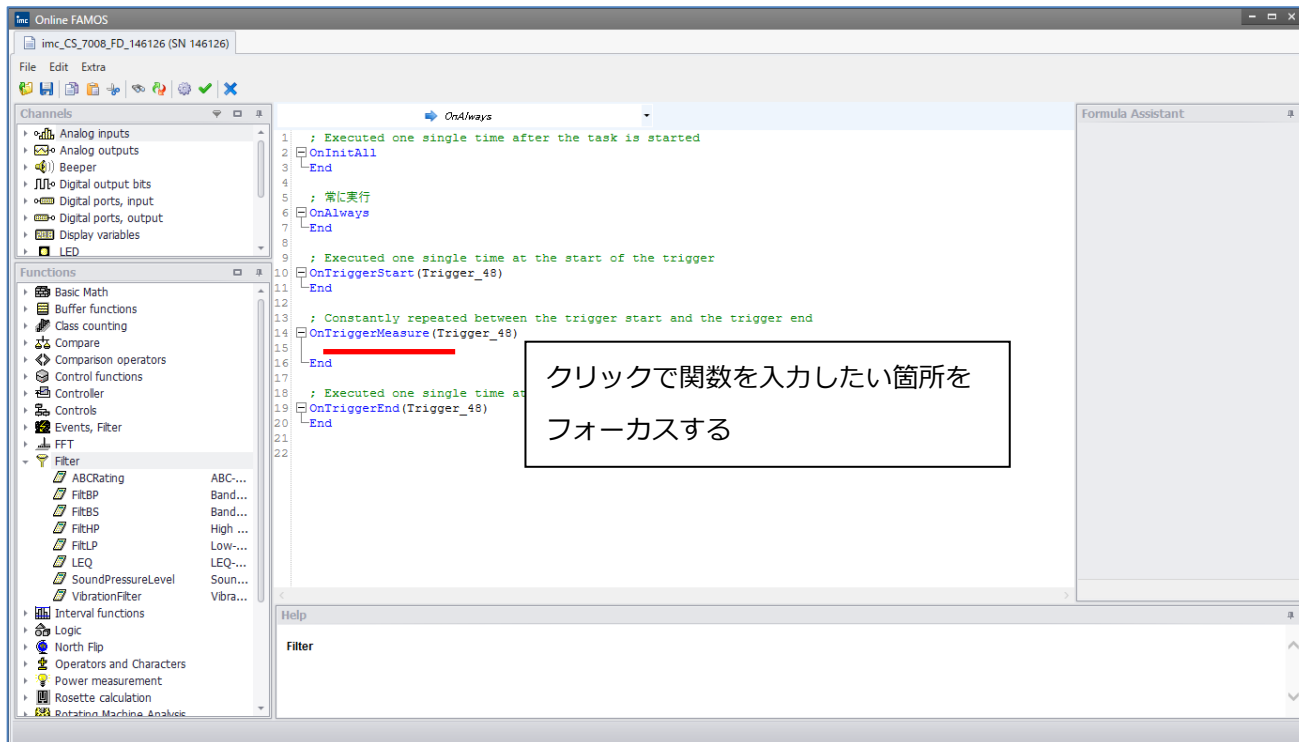
関数は画面左下のウィンドウに、カテゴリごとにツリー表示されています。



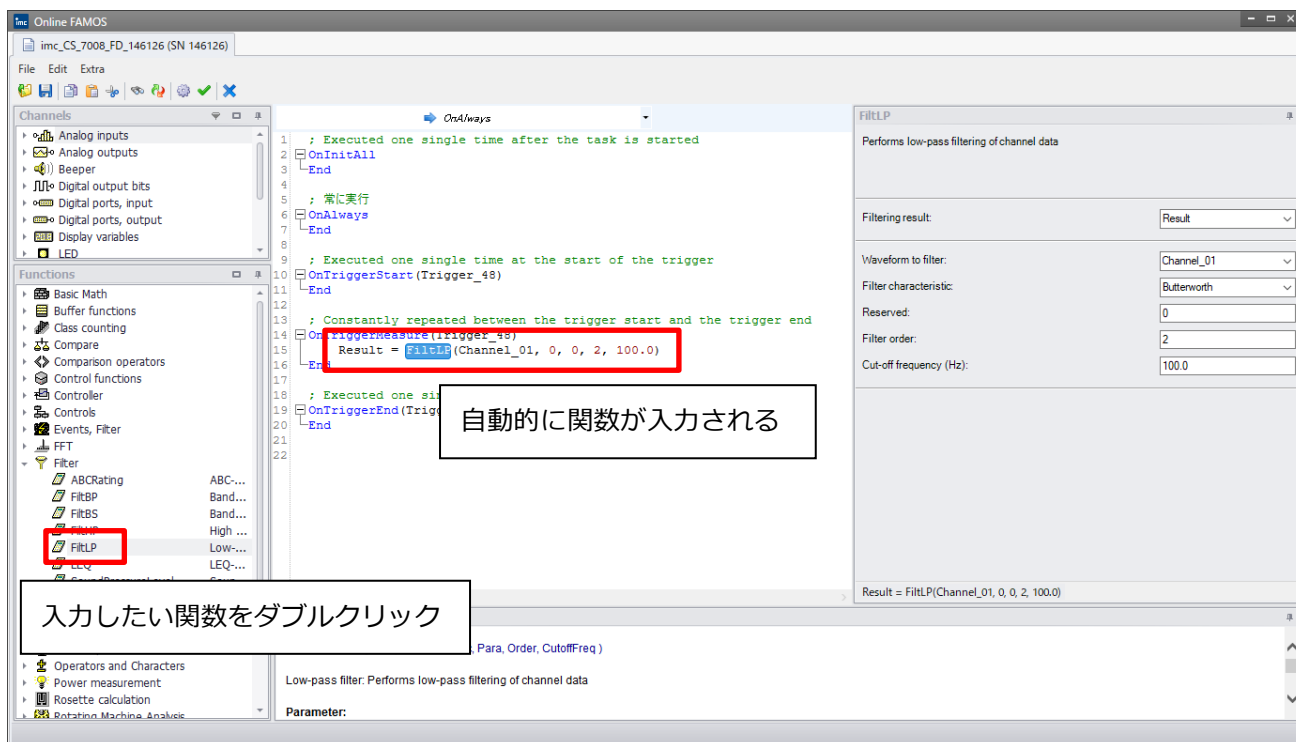
関数は手入力もできますが、アシスタントを利用する入力が簡単です。一例として、ローパスフィルタを適用する関数 `FiltLP` を入力する手順を説明します。

まず、入力する箇所をクリックしてフォーカスします。

ローパスフィルタの場合、計測データに対し適用する処理なので、計測中に実行するブロックである `OnTriggerMeasure` の中にフォーカスします。



画面左下の関数ウィンドウから、[Filter]カテゴリの[FiltLP]を選択してダブルクリックします。
記述欄でフォーカスしていた箇所に関数が入力されます。

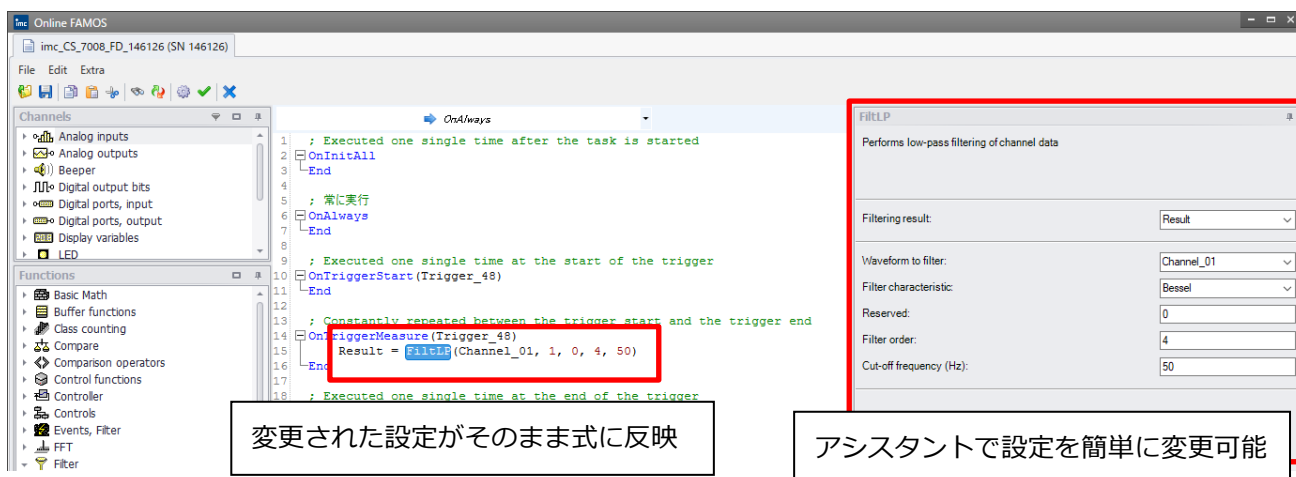


画面右側にアシスタントウィンドウが自動的に表示されるので、そこで任意の項目を編集できます。

下図では例として、フィルタ特性(Filter characteristic)をBesselとし、次数(Filter order)を4次、カットオフ周波数(Cut-off frequency)を50Hz としました。

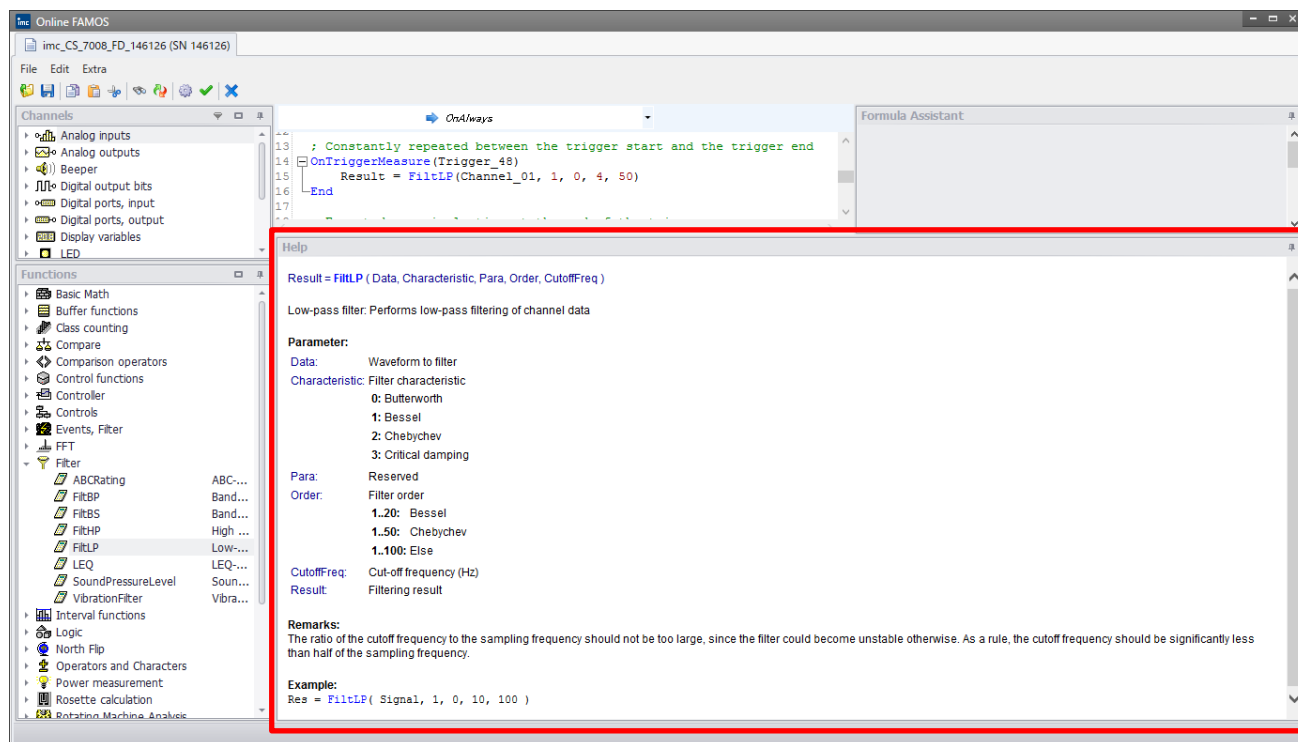
対象となるチャンネルやフィルタ特性など、いくつかの候補から選ぶような項目はドロップリスト式での選択が可能です。

このウィンドウで変更した値は、そのまま式の記述にも反映されます。



また、画面下側のヘルプウィンドウには現在選択中の関数の説明が表示されます。

この関数がどんな動作をするか、各パラメータがどのような意味を持つか、設定値はどのような範囲が選択可能か、などの各種情報が確認できます。



5.3. Online FAMOS でのチャンネルと単数値

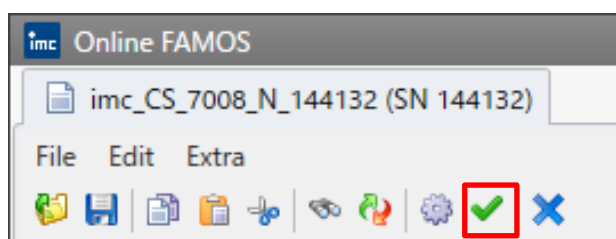
この説は、Online FAMOS で記述を行う場合の TIPS です。

Online FAMOS では、そのデータが「チャンネル」(連続値)であるのか「単数値」であるのかを明確に区別します。

チャンネルであるデータ例：アナログ入力チャンネル、アナログ入力チャンネルに加減乗除を行ったデータ

単数値であるデータ例：ディスプレイ変数、バーチャルビット、OnInitAll で作成した変数

データ形式が合わないものを入力すると、Online FAMOS はエラーを出力します。エラーチェック用のアイコンは下図です。



下図は実際にデータ形式が合わないものを入力したときのエラー例です。画面最下部のバーに表示されます。

不正なパラメータのデータ型です!単一値を指定してください! Error number: -20305
不正なパラメータのデータ型です!チャンネルを指定してください! Error number: -20066

ありがちな間違いは、OnInitAll で作成した「単数値」に対して、OnTriggerMeasure で「チャンネル」を代入しようとするケースです。

単数値とチャンネルは関数を使用すると相互に変換することが可能です。

5.3.1.単数値のチャンネル化

単数値をチャンネル化するために使用する関数は、CreateVChannel または CreateVChannelInt です。
以下が記述例です。

```
SignalSampleValue = CreateVChannel(Channel_01, _value1)
```

この記述は、単数値_value1 を、チャンネル Channel_01 と同等の特性を持つ、チャンネル SignalSampleValue に変換する、ということの意味します。
チャンネル Channel_01 と同等の特性を持つ、とは、サンプリングや単位といった特性値が Channel_01 から引き継がれることを意味します。単位の変更については後述します。

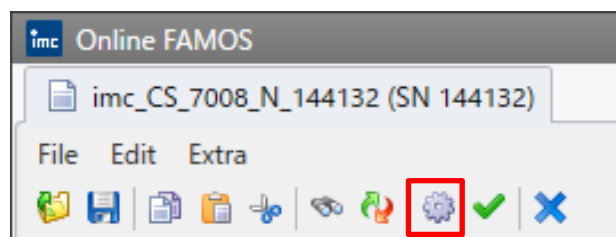
CreateVChannelInt も使い方は同じです。

CreateVChannel の場合は値の表現が実数、CreateVChannelInt の場合は値の表現が符号付整数になります。
明示的に整数値のみを取る場合は CreateVChannelInt、通常は CreateVChannel を使います。

チャンネル名(左辺)には既存のものと重複しない限り任意の文字列が使用できます。

特殊な名前として、_(アンダーバー)で始まるチャンネルは、Online FAMOS 内部限定のチャンネルとして扱われます。このチャンネルは 4.3 節で説明したバーチャルチャンネルの欄には現れず、表示や保存を行うことが無いチャンネルとなります。

作成したチャンネルの単位を変更したり、コメントを定義したりといった操作は、以下のアイコンから実行します。

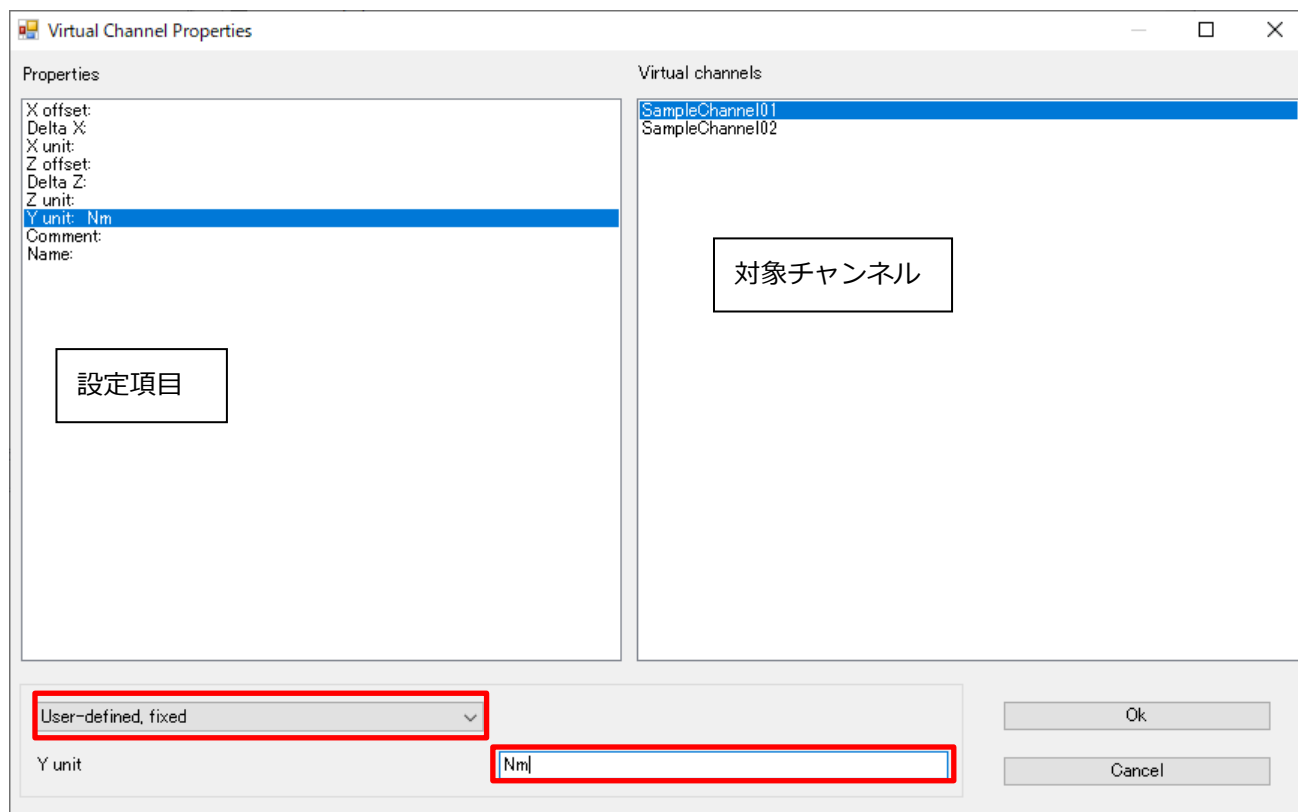


下図のような画面が表示されるので、右側で対象とするチャンネル、左側で設定する項目を選択します。

実用上、設定する項目は単位 Y unit と、コメント Comment 程度です。

選択後、下側で[User-defined, fixed]を選択し、任意の設定値を入力します。

設定後は、[OK]でこの画面を終了します。



5.3.2.チャンネルの単数値化

単数値をチャンネル化するために使用する関数は、CurrentValue です。

以下が記述例です。

```
_speedValue = CurrentValue(SignalSpeed, 0, 0)
```

この記述は、チャンネル SignalSpeed を、単数値_speedValue に変換することを意味します。

チャンネル以外のパラメータは基本的に両方とも 0 としておいて問題ありません。必要であれば詳細はヘルプを参照してください。

6. Online FAMOS のデバッグ

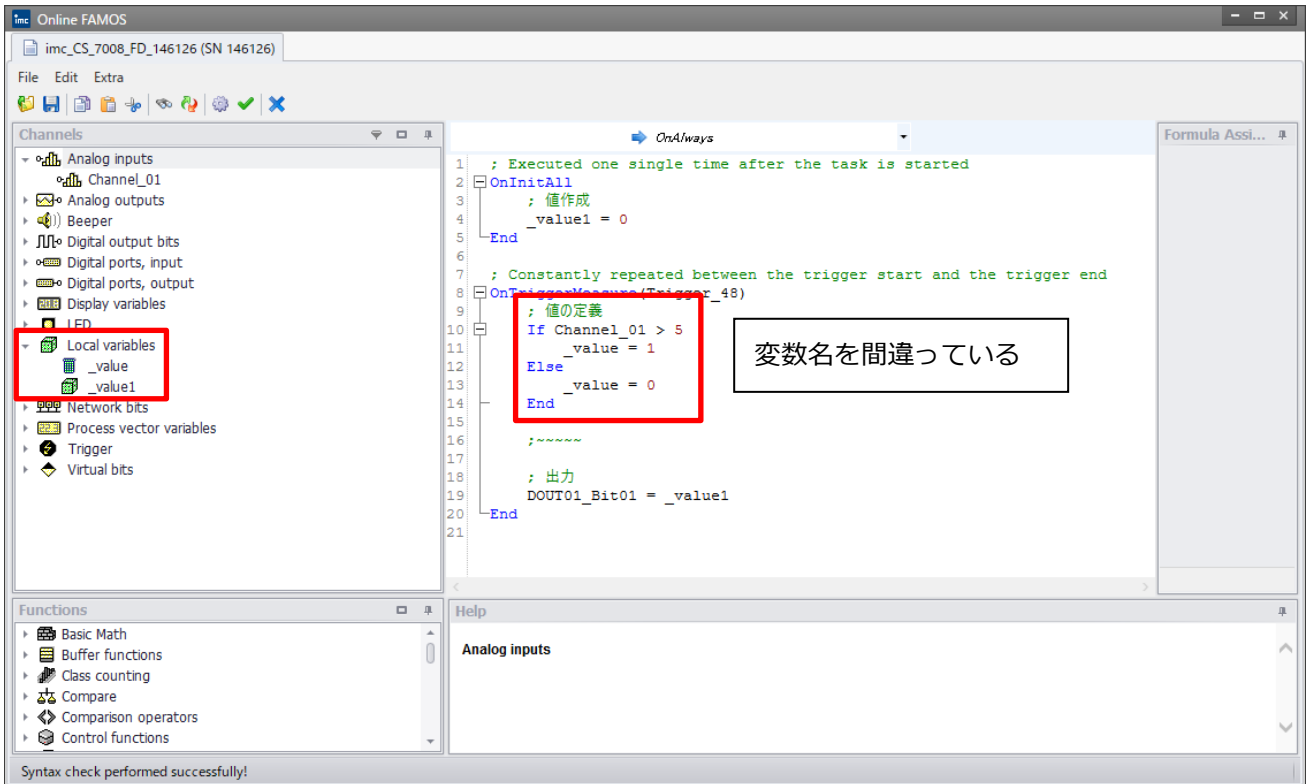
ここでは、Online FAMOS で典型的なミスやその対策等を紹介します。

作成した Online FAMOS が意図した動作とならない時の参考としてください。

6.1. 変数名等の記述ミス

例として下図の記述は、_value1 という変数を作成してそれをデジタル出力に割り当てています。

アナログ入力に応じてこの_value1 を分岐させるつもりで、記述ミスで_value という名前にしてしまっています。



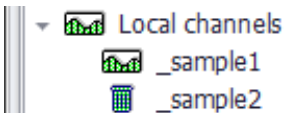
このようなミスが無い確認するには、変数リストの[Local variables]に注目してみましょう。

_value はゴミ箱のようなアイコン、_value1 はサイコロのようなアイコンになっています。

ゴミ箱のようなアイコンで表示される変数は、**Online FAMOS 上には存在するが使われていない(最終的にチャンネル化やディスプレイ変数、出力への割り当てなどがされていない)変数**であることを意味します。

通常、Online FAMOS で定義された変数はすべて何かしらの役割を持っているはずなので、このようなアイコン表示になっている変数はどこかで記述ミスをしている可能性があります。

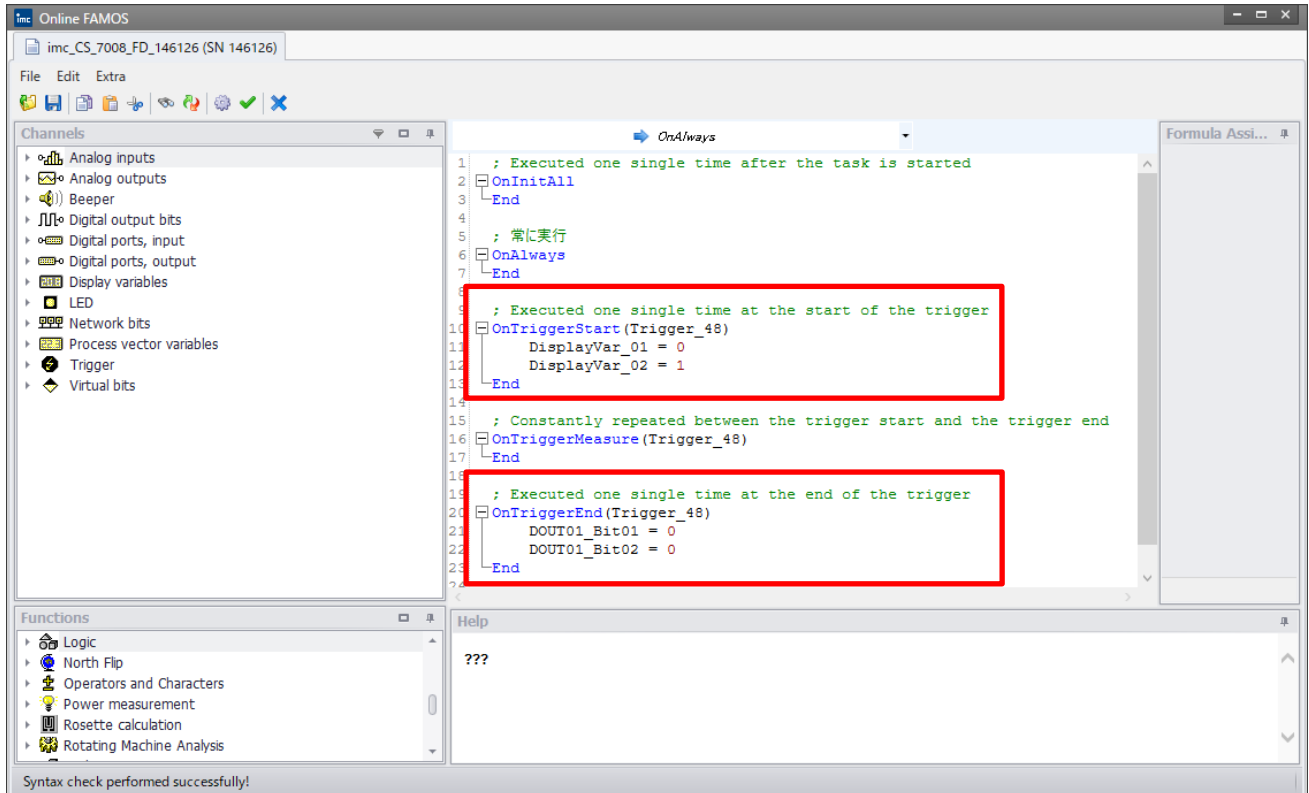
なお、変数ではなくチャンネル(_から始まるもの)の場合も同様に、ゴミ箱アイコン表示となります。



6.2.初期値/最終値の定義忘れ

最初の1回は正しく動作するが、2回目以降の動きがおかしいというケースがあります。

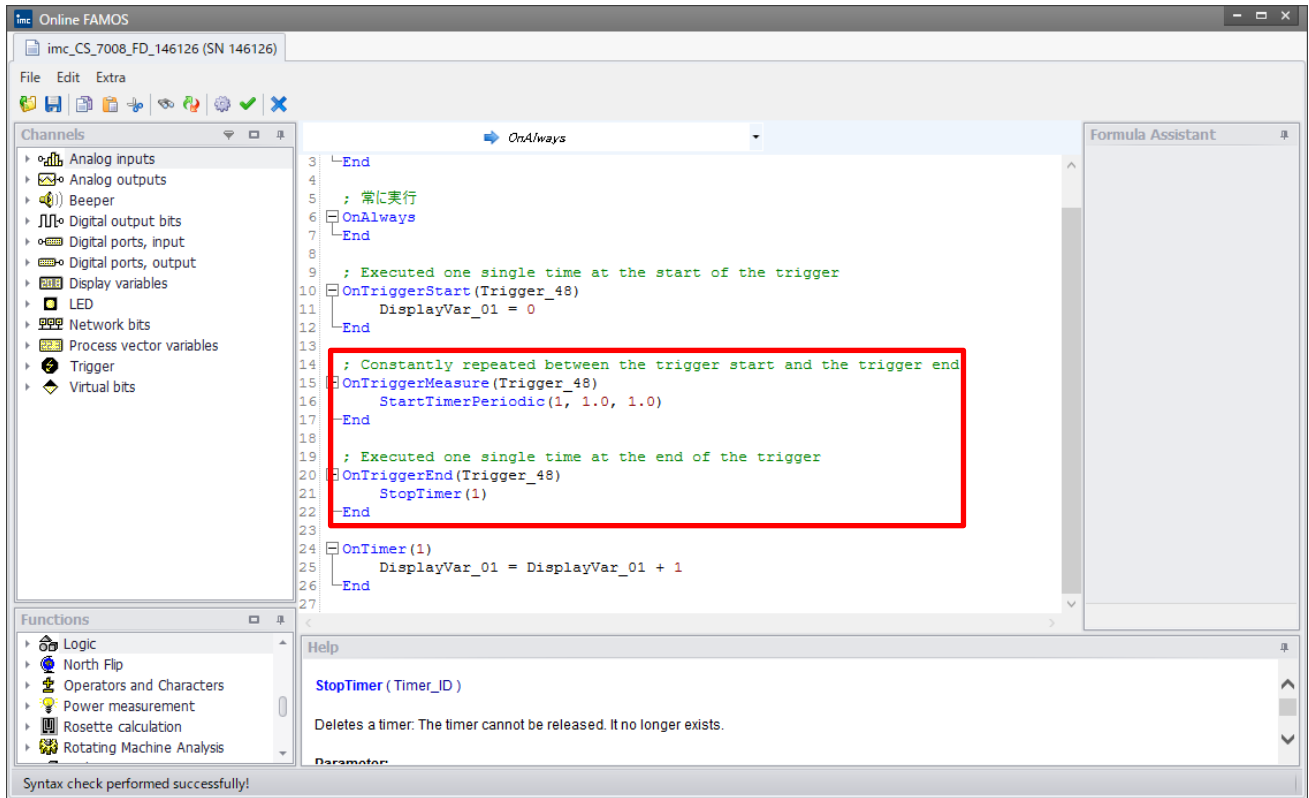
このような場合、OnTriggerStart で初期値を定義する、OnTriggerEnd で最終値を定義する、という動作が抜けている可能性があります。



例えば動作中ずっとカウントアップしている変数があり、その変数の初期値を定義していないと、2回目では増えた後の値から引き継いでいくような動作となります。

6.3.タイマーの止め忘れ

タイマー、特に StartTimerPeriodic で作成された継続タイマーは、停止させない限りずっと動作を続けます。この場合も 6.2 節と同じく、最初の 1 回は正しく動作するが、2 回目以降の動きがおかしいという状態になるケースが多いです。



基本的には `OnTriggerEnd` でタイマーを停止させるようにします。

6.4.その他のケース

原因が特定できない時は、Online FAMOS 上で使用している変数を `CreateVChannel` でチャンネル化して、実際にその変数がどのように変化しているかをモニターしてみましょう。

意図しない変化を見つけ、その変化のもととなっている記述を修正してください。