

WEINTEK LABS., INC.

速度制御 試運転

機能ブロック

MC_MoveVelocity

サンプルプロジェクト

目次

1. 概要及びシステム.....	1
2. Weintek Library をインストールする	3
3. Codesys プロジェクトに iR-PU01-P を追加する	4
3.1. Weintek Built-in CODESYS を使用して iR-PU01-P 装置を追加する....	4
3.2. cMT-CTRL を使用して iR-PU01-P 装置を追加する	6
4. iR-PU01-P パラメータの設定	7
5. 機能ブロックの紹介.....	10
6. サンプルプロジェクト.....	12
7. ログインして操作する.....	14

1. 概要及びシステム

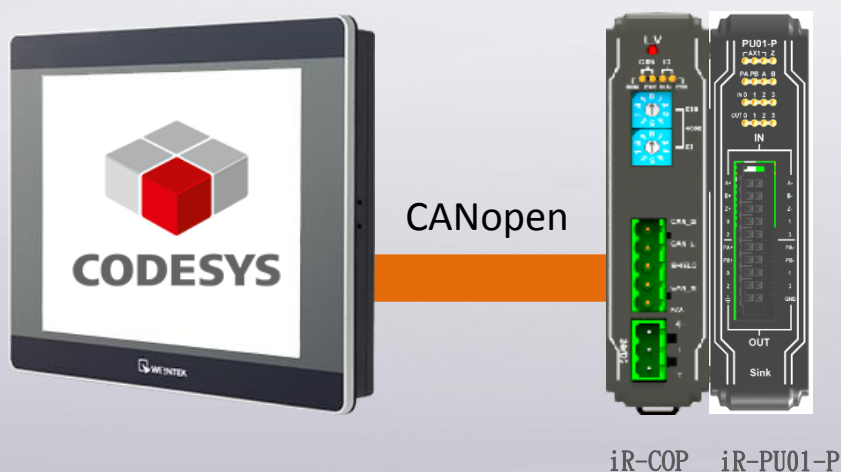
概要

以下では、Weintek Library 機能ブロックと iR-PU01-P を使用してサーボモーター、またはステップモーターを制御して試運転する方法を説明致します。

CODESYS で iR-PU01-P のパルス出力信号を制御し、サーボモーターまたはステップモーターのドライバが iR-PU01-P から送ったパルス信号に基づいてモーターの回転距離及び速度を決め、位置決め或いは速度制御を果たします。

(iR-COP V1.03 を使用する場合、
iR_Application_JOG_Demo_20190906 を参考してください。
cMT-CTRL01 を使用する場合、
iR_Application_JOG_Demo_CTRL_20200707 を参考してください。
cMT-HMI を使用する場合、
iR_Application_JOG_Demo_HMI_20200707 を参考してください)

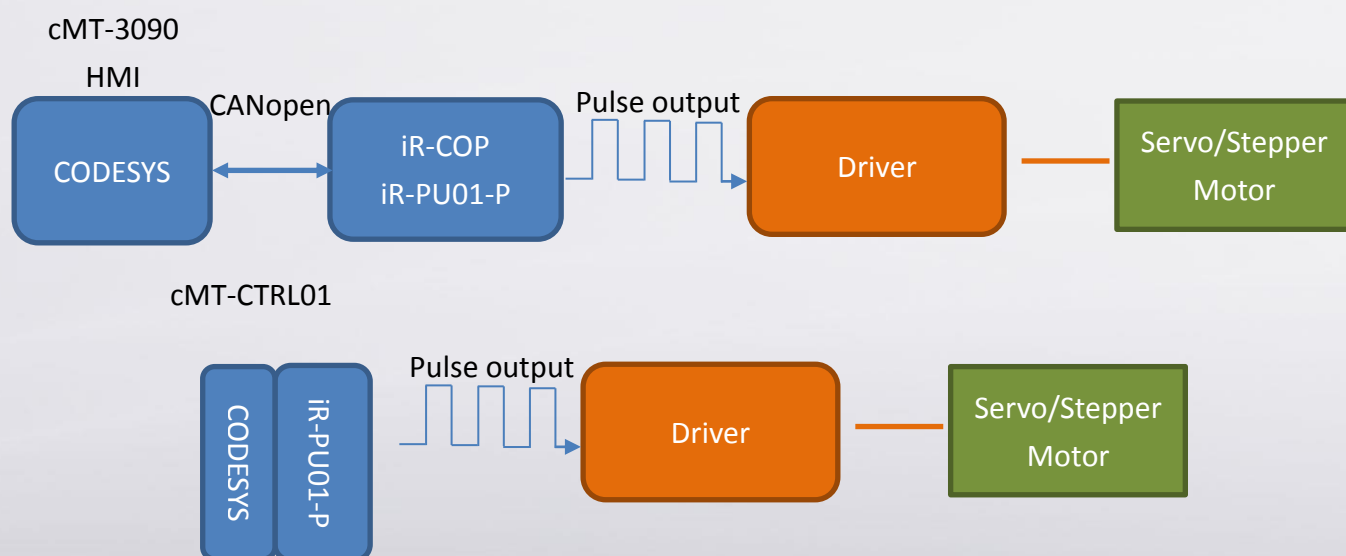
システム





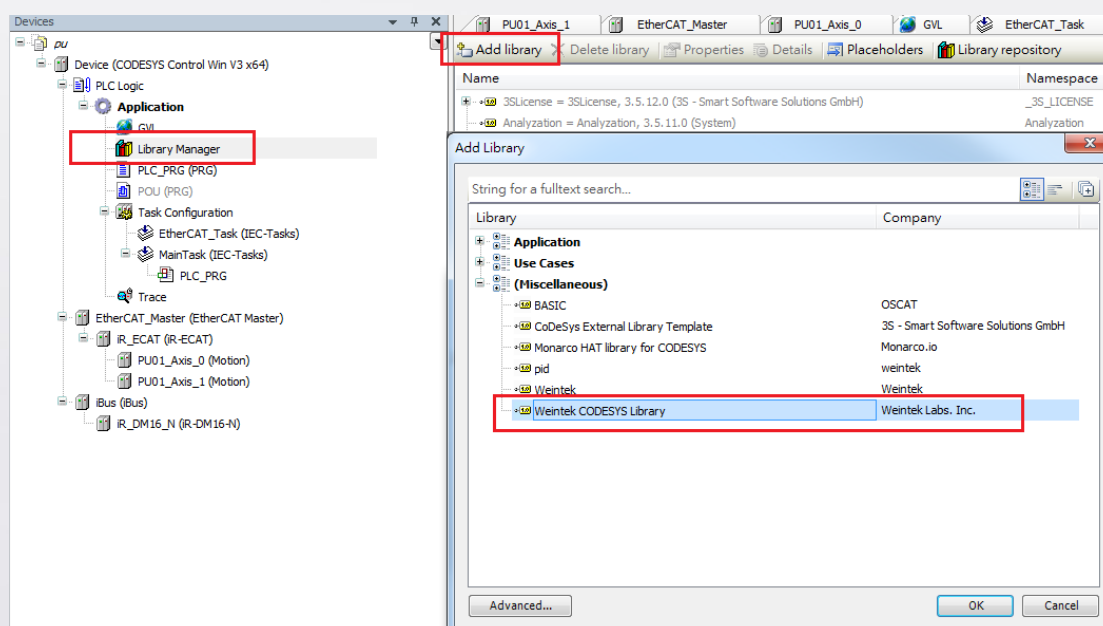
cMT-CTRL01 iR-PU01-P

試運転信号のフローチャート：



2. Weintek Library をインストールする

1. Weintek オフィシャルウェブサイトのダウンロードコーナーをオープンし、[cMT+CODESYS Package]を検索し、ダウンロードしてインストールします。<https://www.weintek.com/globalw/Download/Download.aspx> (iR-PU01-P 記述ファイルを含む)
2. CODESYS インタフェースに Weintek CODESYS Library を追加します。



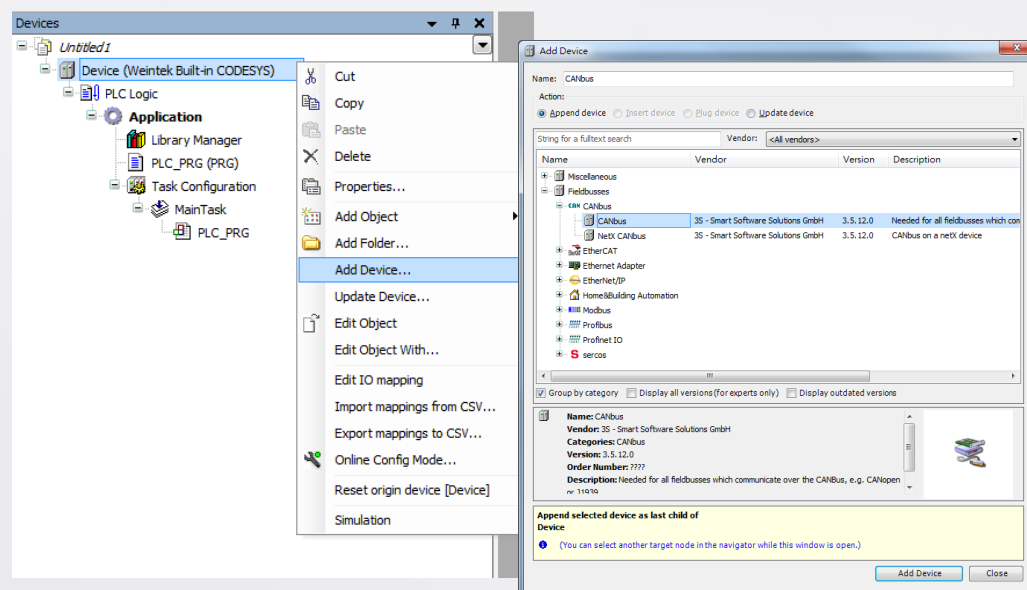
3. インストールが完成したら、Motion Function Block(モーション機能ブロック)を使用できます。

3. Codesys プロジェクトに iR-PU01-P を追加する

3.1. Weintek Built-in CODESYS を使用して iR-PU01-P 装置を追加する

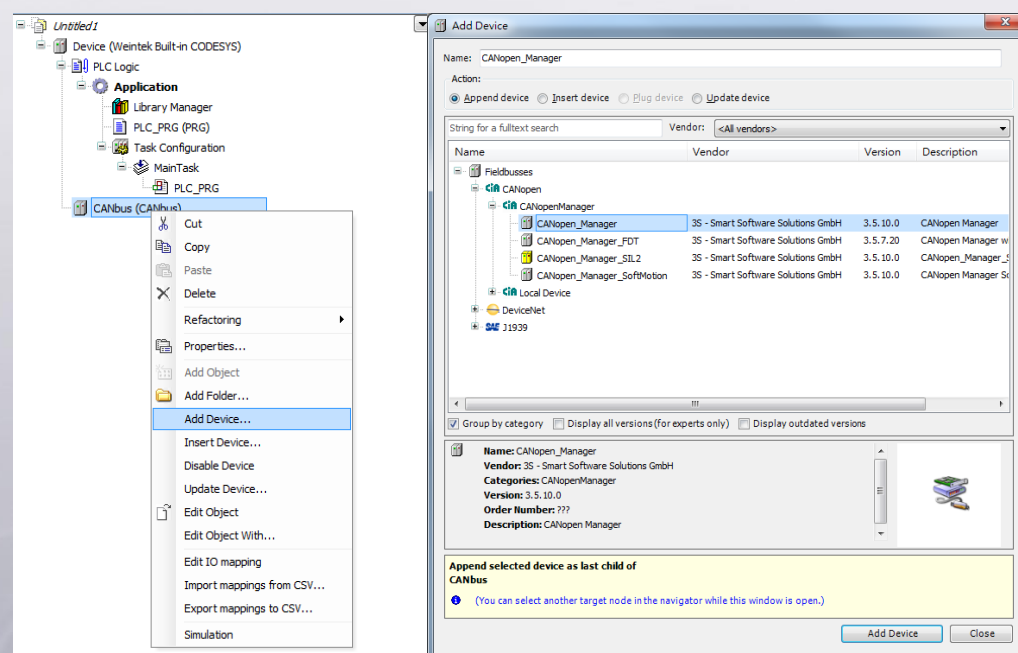
- CANbus 装置を追加します：

[Device]->[Add Device]->[Fieldbusses]->[CANbus]

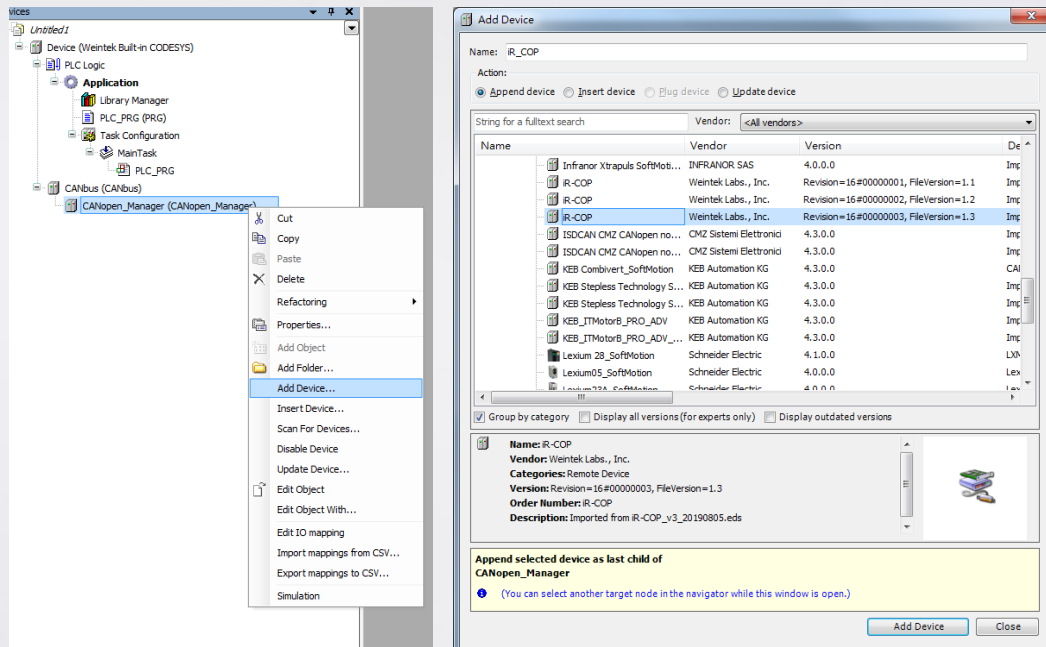


- CANopen_Manager 装置を追加します：

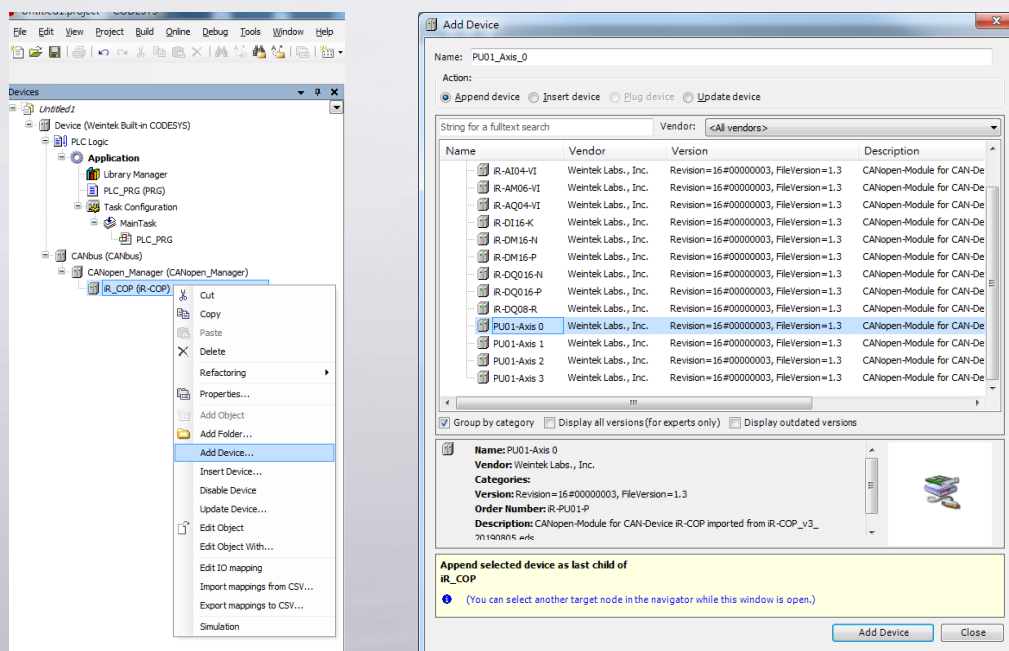
[CANbus]->[Add Device]->[CANopen_Manager]

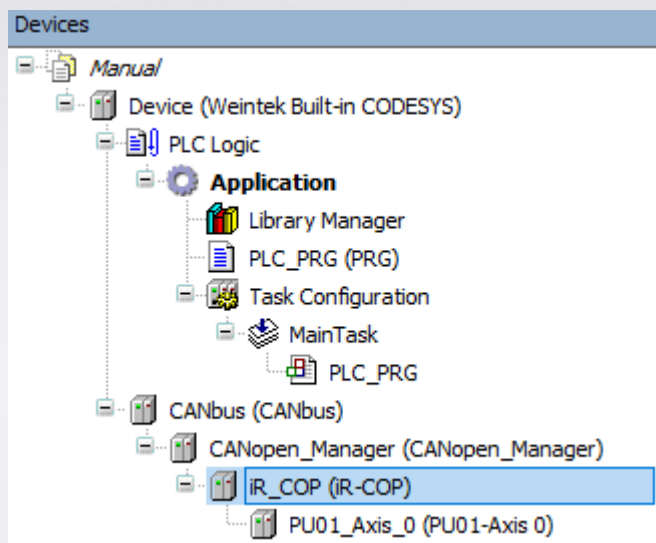


- iR-COP モジュールを追加します：
[CANopen_Manager] ->[Add Device]->[iR-COP] (バージョン V1.3)



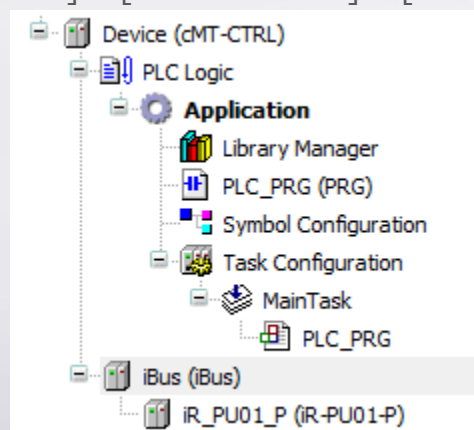
- iR-PU01-P モジュールを追加します：
[iR-COP]->[Add Device]->[PU01-Axis 0]





3.2. cMT-CTRL を使用して iR-PU01-P 装置を追加する

- iR-PU01-P 装置を追加します：
[iBus]->[Add Device]->[Miscellaneous]->[iR-PU01-P]



4. iR-PU01-P パラメータの設定

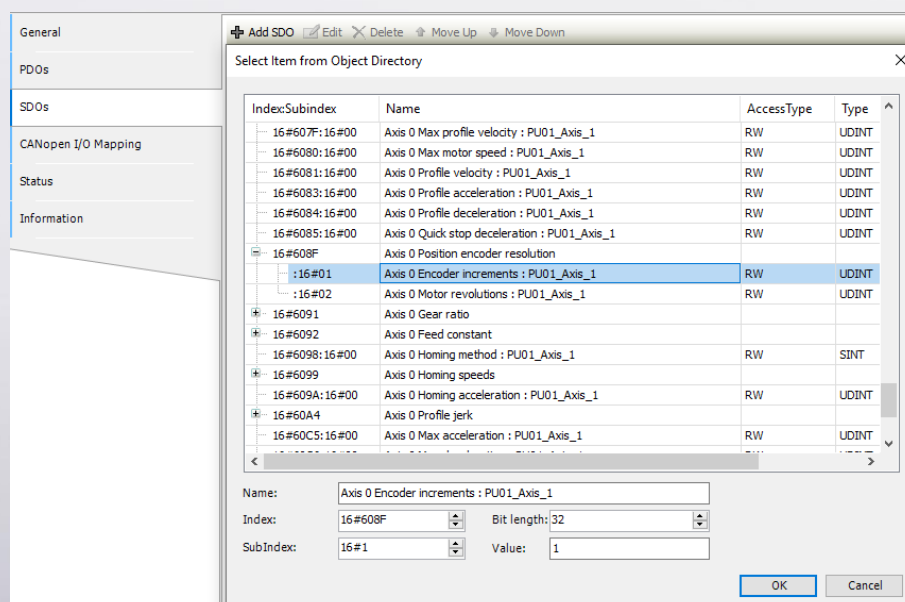
本例で使用された設定項目は以下のとおりです：

+ Add SDO Edit Delete Move Up Move Down				
Line	Index:Subindex	Name	Value	Bit length
1	16#608F:16#01	Encoder increments : AX1_PU01	16#1	32
2	16#608F:16#02	Axis 1 Motor revolutions : PU01_Axis_1	1	32
3	16#5511:16#00	Axis 1 Pulse Output Method : PU01_Axis_1	4	8
4	16#6080:16#00	Axis 1 Max motor speed : PU01_Axis_1	200000	32
5	16#607F:16#00	Axis 1 Max profile velocity : PU01_Axis_1	200000	32
6	16#60C5:16#00	Axis 1 Max acceleration : PU01_Axis_1	100000	32
7	16#60C6:16#00	Axis 1 Max deceleration : PU01_Axis_1	100000	32
8	16#6085:16#00	Axis 1 Quick stop deceleration : PU01_Axis_1	100000	32

モーションコントロールモジュールでモーターを制御する前に、まずモジュールに保護措置を取り、及び単位を設定する必要があります。もし設定無しでそのまま機能ブロックで制御を行った場合、iR-PU01-P モジュールのインジケータはエラー状態を表示します。

パラメータの設定方法は以下のとおりです：

[iR-COP]->[SDOs]->[Add SDO]



追加した SDO パラメータはログインした後、iR-PU01-P モジュール内に書き込まれます。

● モーター解像度設定:608Fh

モーター解像度とは、モーターが一周回転するのに所要するパルス数です。試運転では全部 1 に設定されます。

$$\text{モーター解像度} \text{Position encoder resolution} = \frac{\text{エンコーダーインクリメント} \text{encoder increments}}{\text{モーター公転} \text{motor revolution}}$$

● パルス出力方式:5511h

パルス出力方式はモータードライバがサポートするパルス方式に基づいて設定しなければなりません。ドライバと iR-PU01-P モジュールの設定されたパルス方式が同じの場合のみ、正確にモーターの方向と距離を制御できます。
サブインデックス 00h : パルス出力方式

Bit7- Bit 4	予約済み			
Bit3- Bit 0	設定値	PA	PB	
	0	Disable	Disable	
	1	CW	CCW	
	2	Pulse	NC	
	3	Pulse	Direction	
	4	A	B	
	5	A(2x)	B(2x)	
	6	A(4x)	B(4x)	

- **速度最大値:6080h, 607Fh, 60C5h, 60C6h**

モーター最大回転速度 : 6080h

モーター仕様に基づいて最大回転速度を記入します。普通のモーターの最大回転速度の単位は RPM(Round per minute)ですが、このパラメータに記入する単位は PPS(Pulse per second)です。記入する前に、上記両者の単位を換算する必要があります。

プロフィール最大速度 : 607Fh

速度プロフィールの最大許容速度です。モーター最大回転速度 6080h と衝突していれば、数値が小さいほうが最大速度に選ばれます。

最大加速度/減速度 : 60C5h, 60C6h

モーター最大加速度と減速度を制限し、設定値が最大値を超えたら、最大値で実行されます。

- **急停止減速度:6085h**

iR-PU01-P モジュールがモーション中でエラーが発生、またはリミットに触れた場合、自動的に急停止で設定された減速度でモーションを静止します。

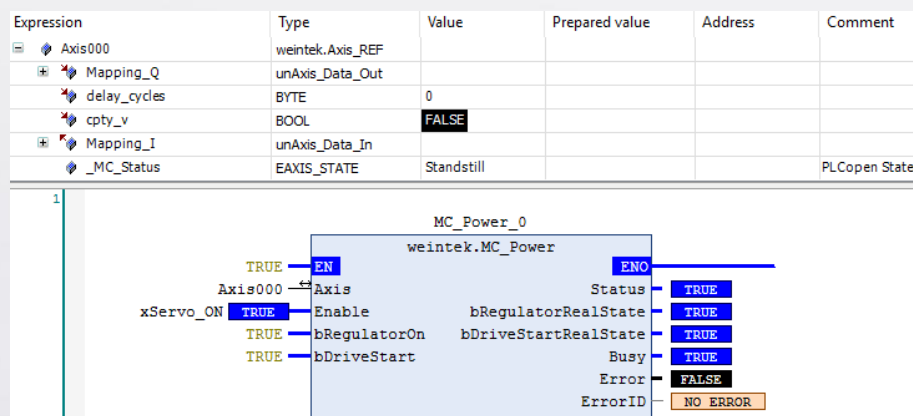
5. 機能ブロックの紹介

Weintek Library の追加方法については、ドキュメント《Weintek 関数ライブラリ》の ch 2 をご参照ください：

UM018017J_CODESYS_Weintek_Library_UserManual_20190305_jpn

軸制御機能の起動：MC_Power

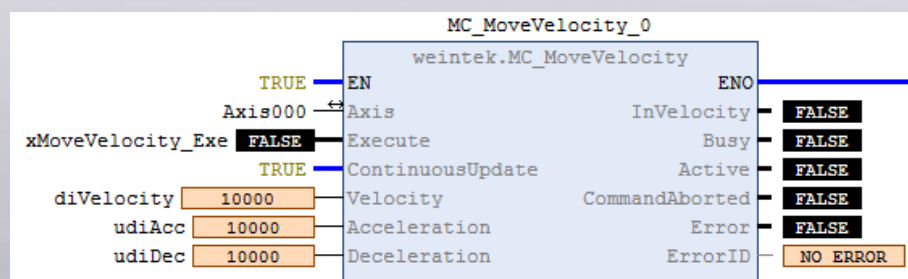
MC_Power ブロックはサーボ制御の Servo 起動機能のように、全てのモーション機能ブロックを使用する前に、先に Power 機能ブロックを起動する必要があります。Power 機能ブロックが起動された後、エラーが発生していなかったら、軸は待機状態〈Standstill〉に入ります。



上図の例のように、MC_Power.Enable をトリガーした後、MC_Status は〈Standstill〉状態にいたので、軸が待機になり随時にモーション制御の機能ブロックからのコマンドを実行できると示しています。

速度モード：MC_MoveVelocity

速度モードの機能は、モーターの回転速度を指定し、等速運動にさせることです。MC_MoveVelocity.Execute をトリガーすれば、機能ブロックのパラメータで速度を制御できます。



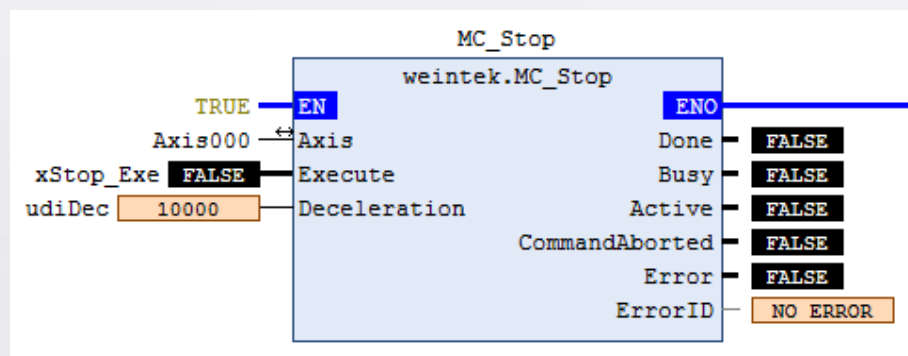
- Velocity：速度の設定です。正数は正転で、負数は逆転です。
- Acceleration：モーションの加速度で、数値は 0 ではありません。

- Deceleration : モーションの減速度で、数値は 0 ではありません。
- ContinuousUpdate : 速度を継続に更新します。TRUE=軸が動作している時に、目標速度、加速度、減速度を変更できます。軸状態はトリガーされた後に〈Continuous Motion〉になり、MC_Stop 或いは MC_Halt でモーションを停止する必要があります。

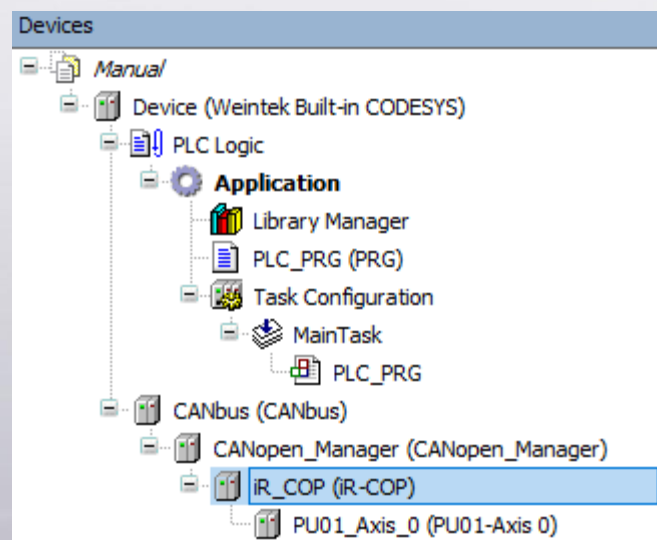
停止 : MC_Stop

停止は軸の移動を中断するコマンドで、停止コマンドを使用すれば、停止動作が完成した場合のみ、他の移動コマンドを出すことができます。

MC_Stop.Execute をトリガーすると、モーションコントロールが停止されます。



- Deceleration : 減速度です。0 ではありません。軸状態は機能ブロックの実行が完了後、〈Standstill〉になります。



6. サンプルプロジェクト

試運転機能

xEnable_Power : システムを起動します。

xServe_ON : サーバーを起動します。

xJOG_exe : TRUE=試運転を実行、FALSE=試運転を停止します。

xUpdate : TRUE=運転中で速度を変更可能、FALSE=運転中で速度を変更できません。機能ブロックを実行する前に、本パラメータを事前に設定する必要があります。

eIO_Control : iR-PU01-P モジュールに内蔵されたデジタル入力で移動をトリガーします。

diSpeed : JOG 試運転の速度を設定します。

udiAcc : JOG 試運転の加速度を設定します。

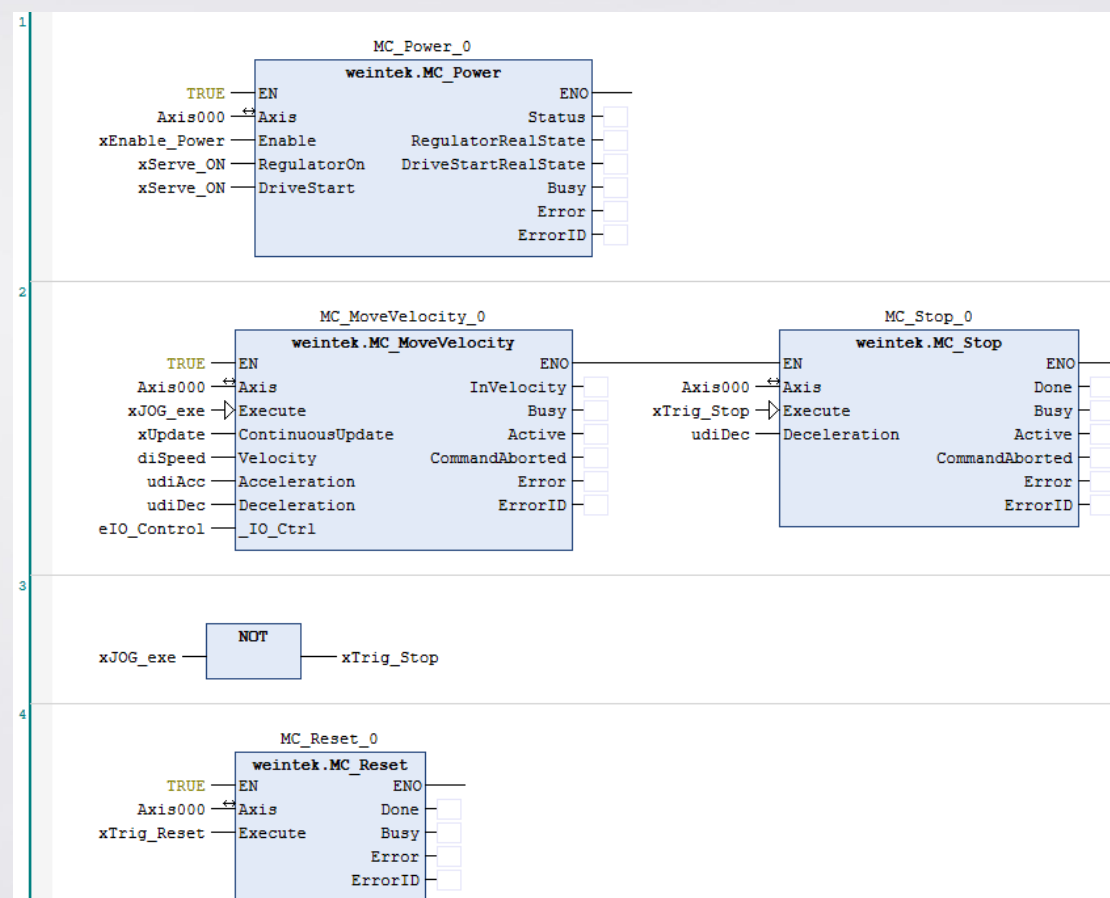
udiDec : JOG 試運転の減速度を設定します。

宣言

```
// Axis reference
Axis000 : Weintek.Axis_REF_Lite ;
// Motion Control Function Block
MC_Power_0: weintek.MC_Power ;
MC_MoveVelocity_0: weintek.MC_MoveVelocity;
MC_Stop_0: weintek.MC_Stop;
MC_Reset_0: weintek.MC_Reset;
// JOG Button
xEnable_Power, xServe_ON, xJOG_exe, xTrig_Reset, xUpdate : BOOL ;
eIO_Control : weintek.eMC_IO_CTRL ;
// JOG parameter
diSpeed : DINT := 1000 ;
udiAcc : UDINT := 1000 ;
udiDec : UDINT := 1000 ;
```

必須の変数を宣言し、初期値を設定します。

FBD プログラム



1 : 軸制御システムを起動し、xEnable_Power & xServe_ON は両方とも TRUE でなければなりません。間違いがなかったら、モーションコントロールを実行し始められます。

2 : JOG 試運転機能ブロックです。

3 : xJOG_exe を押して(TRUE)JOG 試運転を開始します。xJOG_exe を手を離して(FALSE)JOG 試運転を停止します。

4 : モーション中でエラーが発生したら、xTrig_Reset をトリガーして iR-PU01-P を復帰します。

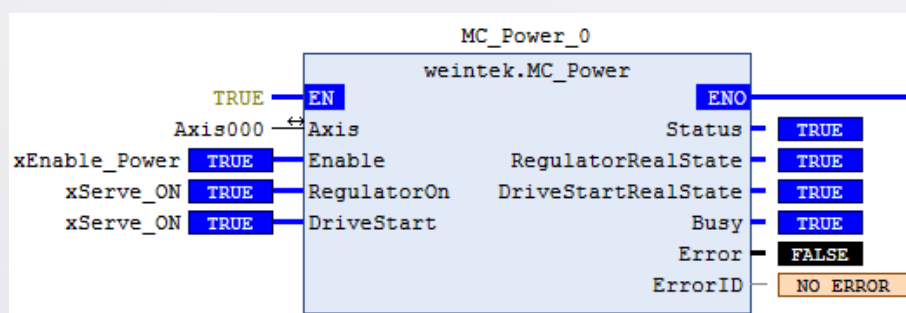
7. ログインして操作する

ここで試運転機能の操作手順を実演します。

モーションコントロールシステムを起動する

xEnable_Power & xServe_ON ボタンを押し(TRUE)、軸制御システムを起動します。

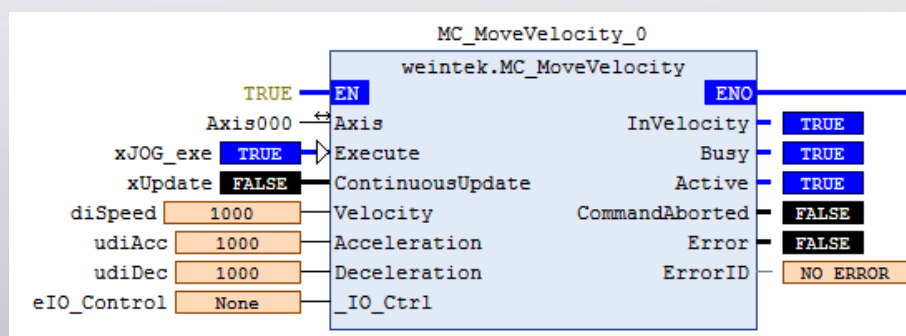
エラーが発生していなかったら、Status、RegulatorRealState、DriveStartRealState、Busy は TRUE であり、エラーが発生した場合、Error = TRUE です。



正転試運転

サンプルプロジェクトのデフォルトは正転で、xJOG_exe(=TRUE)を押せば、iR-PU01-P モジュールは正転パルスを出力します。

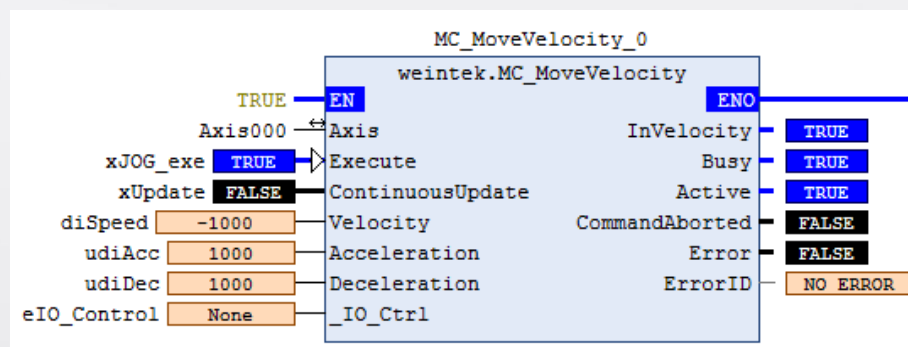
xJOG_exe(=FALSE)に手を離すと、パルスの出力を停止します。



逆転試運転

diSpeed を負数に変更してから xJOG_exe(=TRUE)ボタンを押すと、iR-PU01-P は逆転パルスを出力します。

xJOG_exe(=FALSE)に手を離すと、パルスの出力を停止します。



CODESYS® is a trademark of 3S-Smart Software Solutions GmbH.

本ドキュメントに記載されている各社名、製品名は、一般に各社の登録商標あるいは商標です。
本ドキュメントの記載内容は、予告なく変更する場合があります。

Copyright© 2020 Weintek Lab., Inc. All rights reserved.

DEM19004J_20200729