

WEINTEK LABS., INC.

位置決め 絶対/相対 位置移動

機能ブロック

MC_MoveAbsolute/MC_MoveRelative

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	2
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	14
7. ログインして操作する	17

1. 概要及びシステム

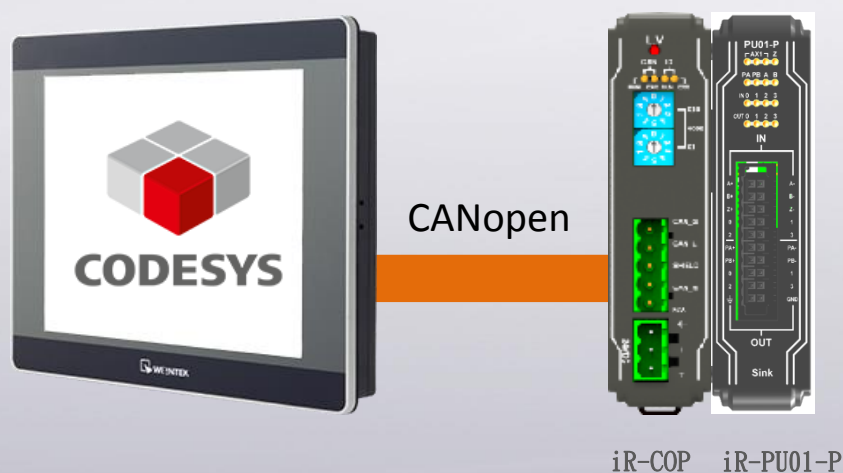
概要

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位置決め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_Positioning_Demo_20190906 を参考してください。
cMT-CTRL01 を使用する場合、iR_Application_Positioning
_Demo_CTRL_20200708 を参考してください。cMT-HMI を使用する
場合、iR_Application_Positioning_Demo_HMI_20200708 を参考し
てください)

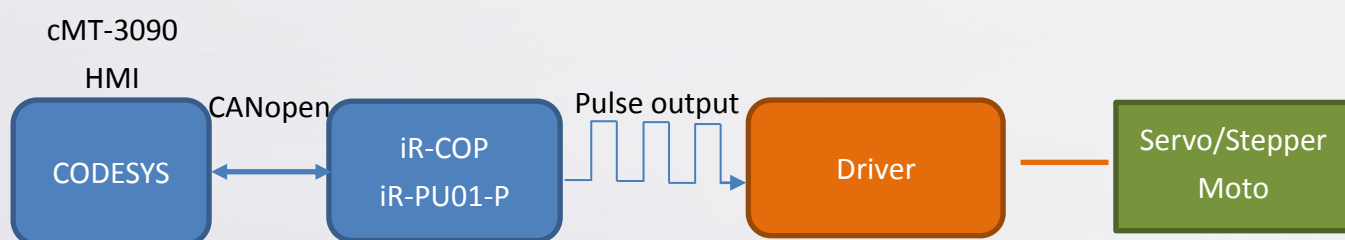
システム





cMT-CTRL01 iR-PU01-P

位置決め信号のフローチャート：



cMT-CTRL01

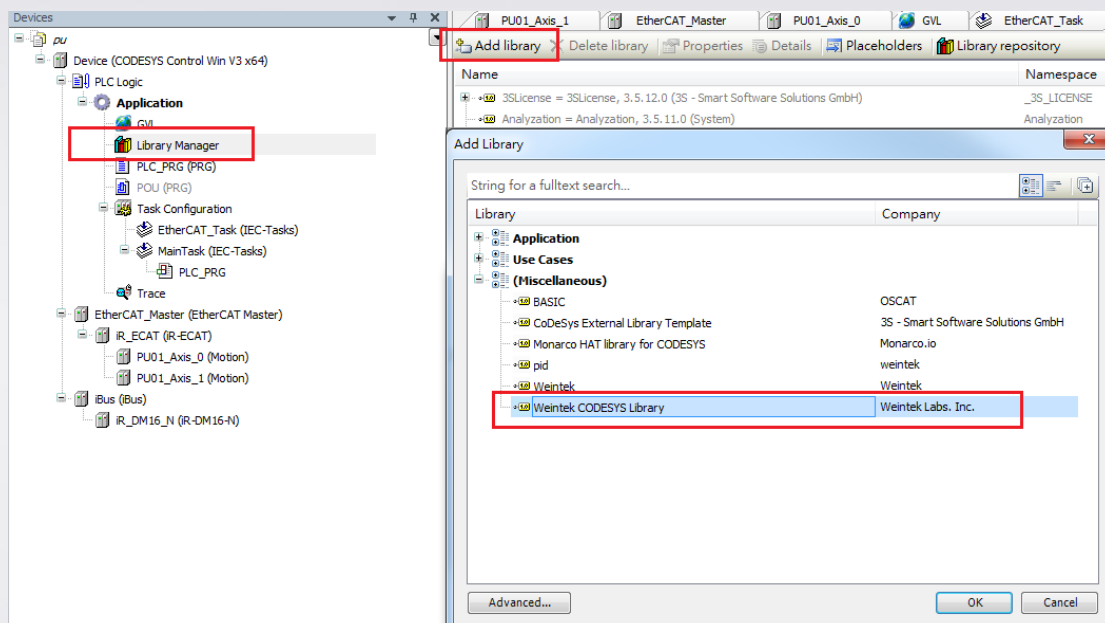


2. Weintek Library をインストールする

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)

DEM19005J_20200729

2. CODESYS インタフェースに Weintek CODESYS Library を追加します。

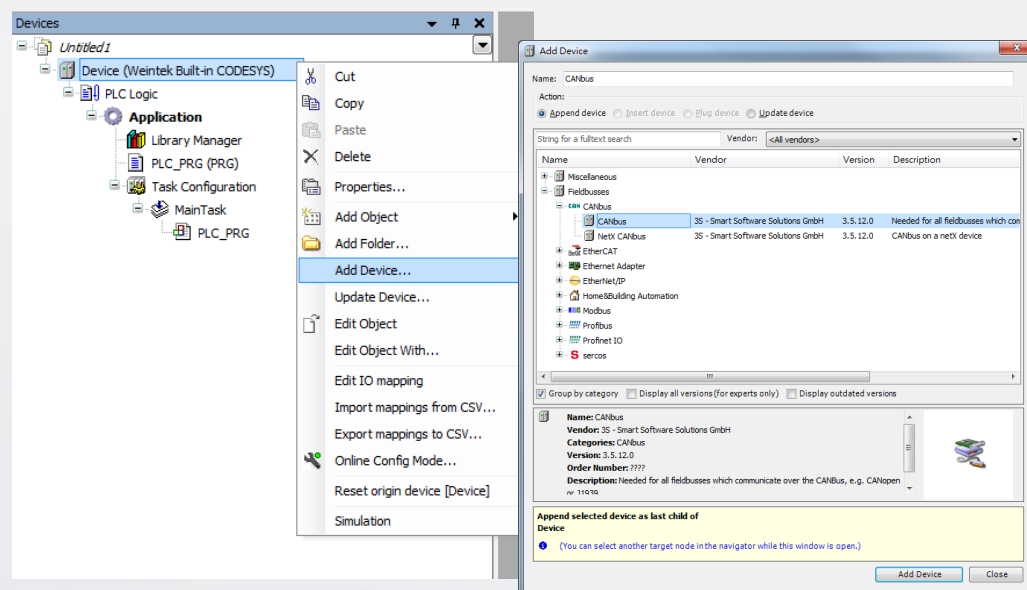


3. インストールが完成したら、Motion Function Block(モーション機能ブロック)を使用できます。

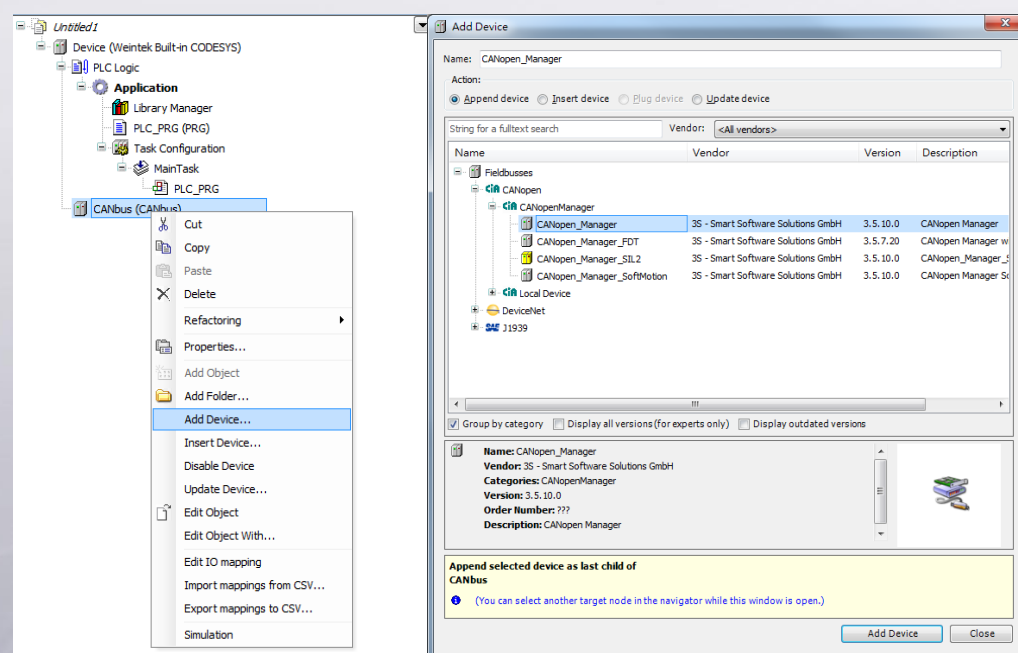
3. Codesys プロジェクトに iR-PU01-P を追加する

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]



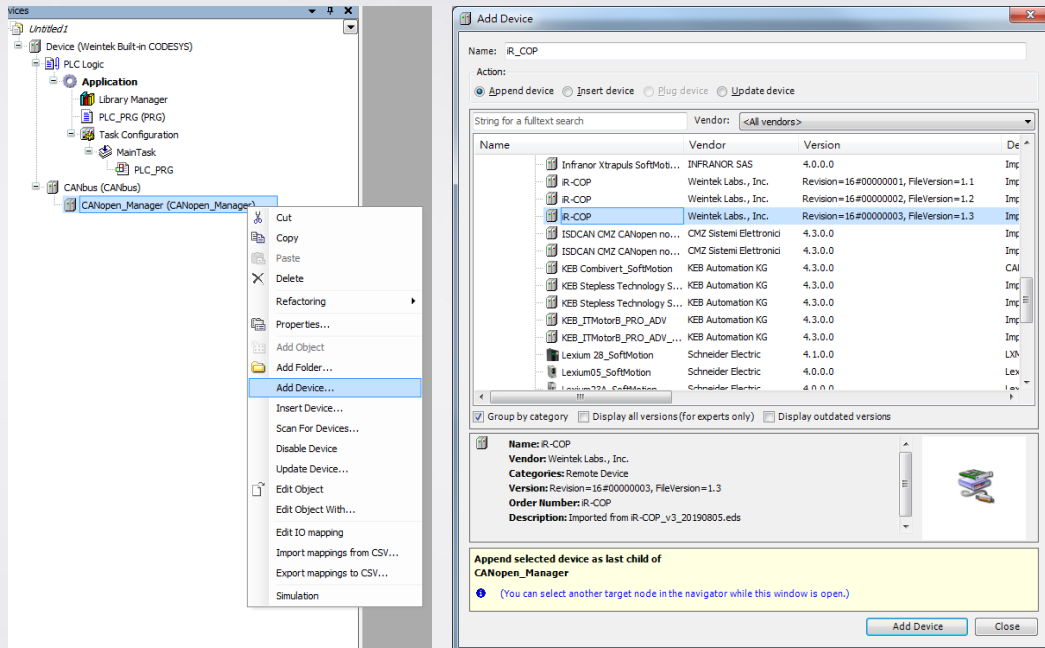
- CANopen_Manager 装置を追加します :
[CANbus]->[Add Device]->[CANopen_Manager]



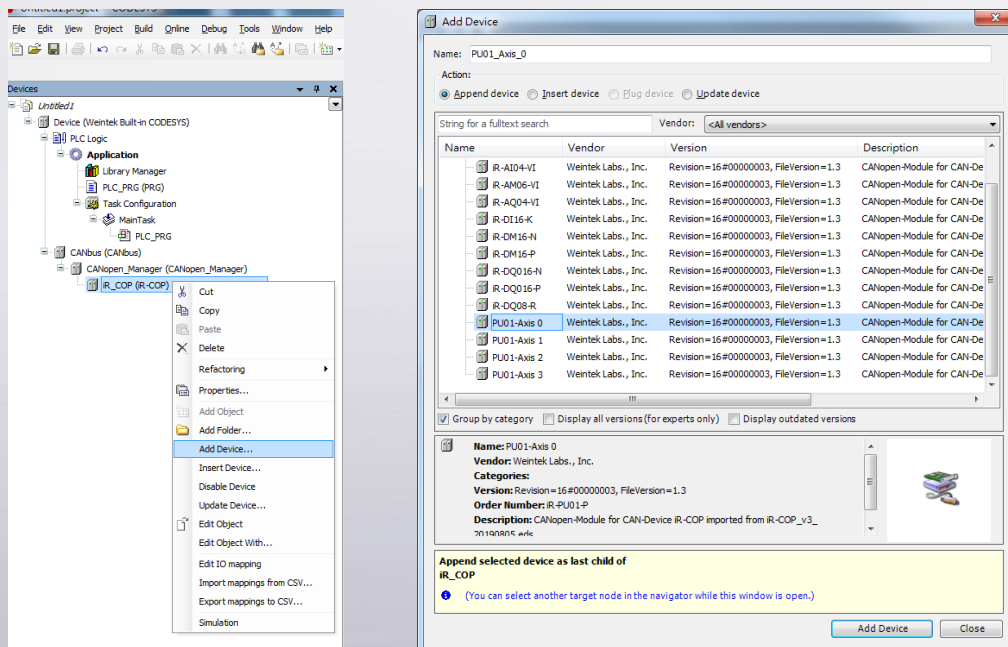
位置決め 絶対/相対位置移動

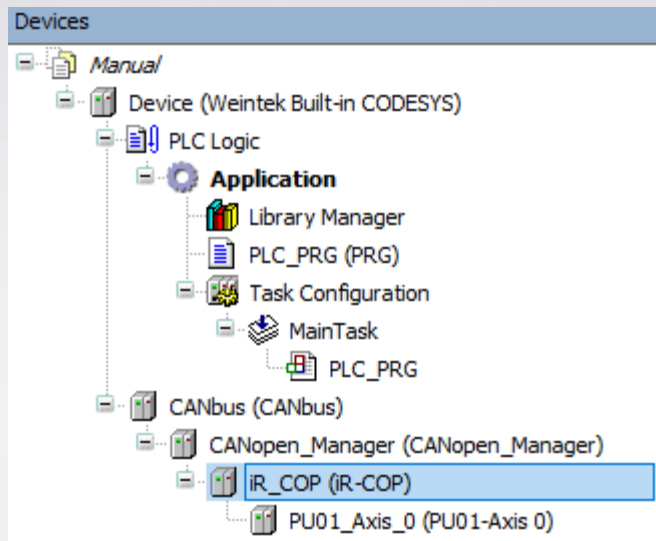


- iR-COP モジュールを追加します：
[CANopen_Manager] ->[Add Device]->[iR-COP] (バージョン V1.3)



- 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

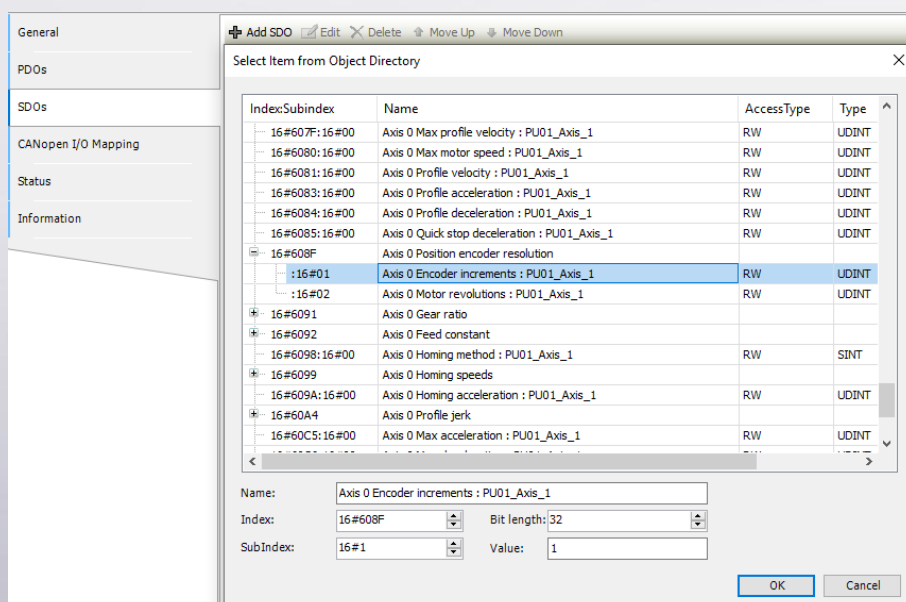
本例で使用された設定項目は以下のとおりです：

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
9	16#6098:16#00	Axis 1 Homing method : PU01_Axis_1	27	8
10	16#607C:16#00	AX1 Home offset	0	32
11	16#6099:16#01	Speed during search for switch : AX1_PU01	10000	32
12	16#6099:16#02	Speed during search for zero : AX1_PU01	2000	32
13	16#609A:16#00	AX1 Homing acceleration : AX1_PU01	10000	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、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(phase*2)	B(phase*2)	
	6	A(phase*4)	B(phase*4)	

- **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ます。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し、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。

- **原点復帰方式:6098h**

iR-PU01-P には 37 種類の原点復帰方式が提供され、本例では方式 27 を使用しました。起動した後、迅速に負極限へ移動して原点を探し、原点センサーに触れたら、鈍速で原点センサーから離れます。

- **原点オフセット:607Ch**

原点復帰が完了したら、原点オフセットで開始位置を設定します。

- **原点復帰速度:6099h**

原点復帰モーションが起動されてから、[Speed during search for switch] に設定した速度を使用します。

一番目の参考ポイントを探し出した後、[Speed during search for zero] に設定した速度を使用します。

- **原点復帰加速度:609Ah**

原点復帰モーションの加速度を設定します。

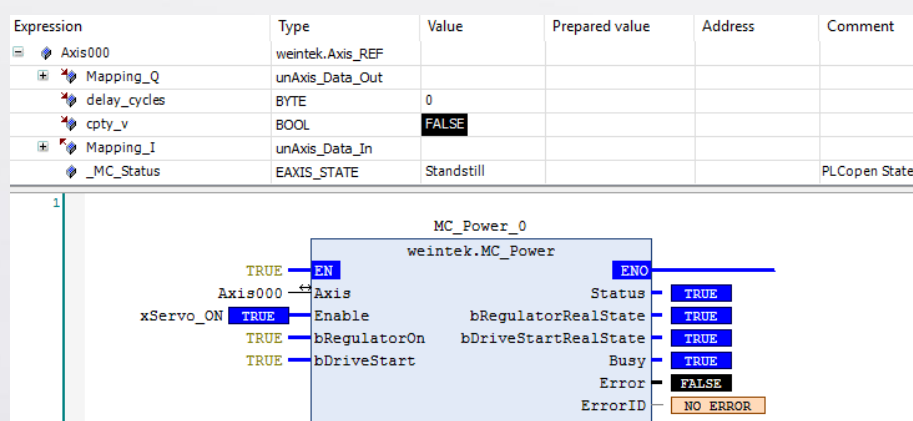
5. 機能ブロックの紹介

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。

原点復帰 MC_Home

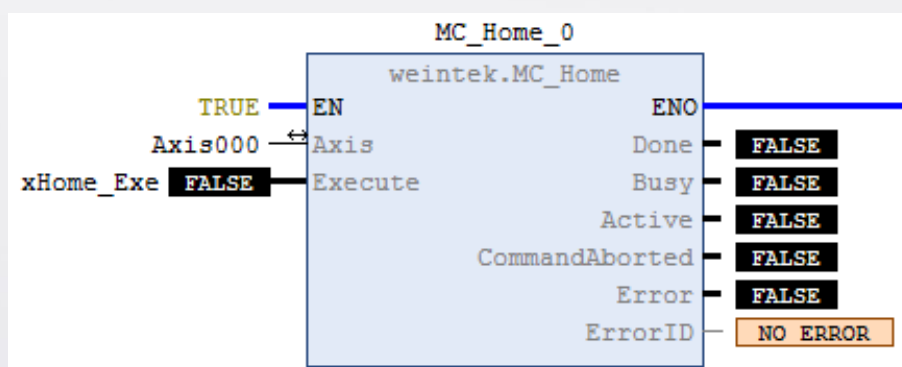
Motion Function Block には 37 種類の原点復帰方式が提供され、MC_Home 機能ブロックをトリガーすると、モジュールは設定に基づき対応する原点復帰方式を実行します。原点の設定パラメータは下記のとおりです：

原点復帰方式 6098(CiA402 規則の 37 種類原点復帰方式に基づき)

- 原点復帰速度-鈍速 6099#01
- 原点復帰速度-快速 6099#02
- 原点復帰加速度 609A
- 原点オフセット 607C

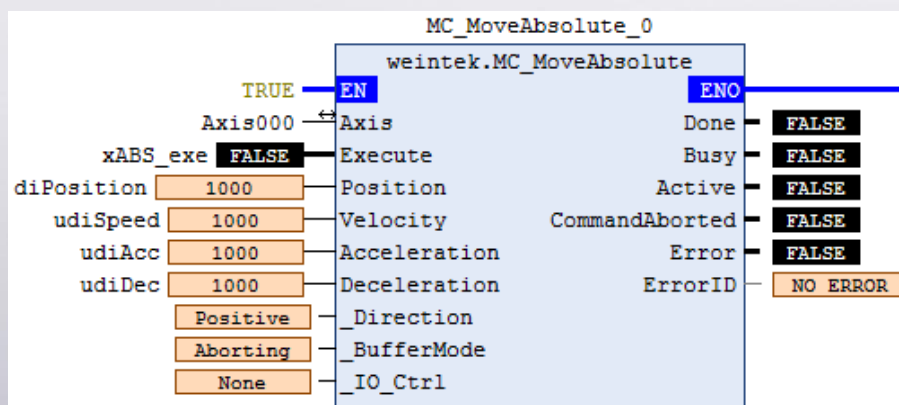
+ Add SDO Edit Delete Move Up Move Down				
Line	Index/Subindex	Name	Value	Bit length
1	16#6098:16#00	Axis 1 Homing method : PU01_Axis_1	27	8
2	16#6099:16#01	Axis 1 Speed during search for switch : PU01_Axis_1	2000	32
3	16#6099:16#02	Axis 1 Speed during search for zero : PU01_Axis_1	10000	32
4	16#609A:16#00	Axis 1 Homing acceleration : PU01_Axis_1	10000	32
5	16#607C:16#00	Axis 1 Home offset : PU01_Axis_1	1000	32

軸が〈Standstill〉状態にいる際に MC_Home.Execute をトリガーすると、モジュールは上記パラメータに基づいて原点復帰を実行します。軸状態は〈Homing〉になり、原点復帰が完了した後、また〈Standstill〉に復元します。



絶対位置移動 MC_MoveAbsolute

絶対位置を指定します。機能ブロックのパラメータに基づいて移動させます。MC_MoveAbsolute.Execute をトリガーすると、機能ブロックパラメータで目標位置に移動させます。



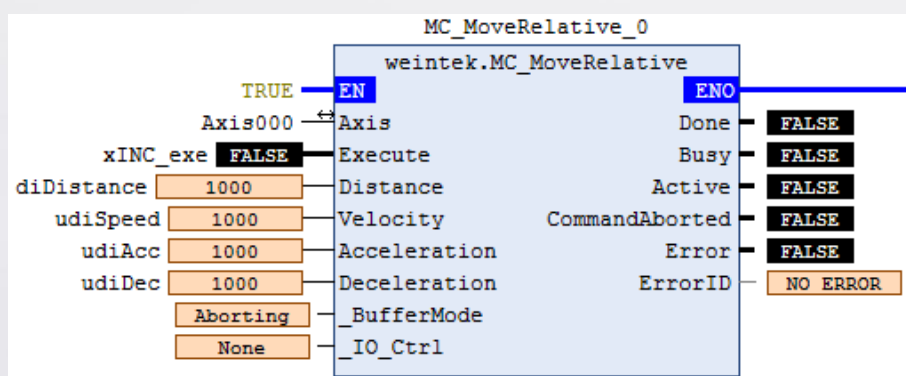
- Position : 絶対位置を入力します。目標位置 = 絶対位置。
- Velocity : 移動速度です。数値が 0 ではありません。
- Acceleration/Deceleration : 移動の加速度/減速度です。数値が 0 ではありません。

- Direction : 回転軸である場合、本パラメータで移動方向または最短ルートを指定します。
- _BufferMode : 本パラメータを設定し、1 個前の移動コマンドに続いて連続移動を果たします。
- _IO_Ctrl : デジタル入力で移動コマンドをトリガーし、動作が完成したら、デジタル信号を出力します。
- 軸状態はトリガーされた後で〈Discrete Motion〉になり、目標位置に移動したら、また〈Standstill〉になります。

相対位置移動 MC_MoveRelative

移動距離を指定します。現在位置+移動距離=目標位置です。

MC_MoveRelative.Execute をトリガーすると、機能ブロックパラメータで目標位置に移動させます。

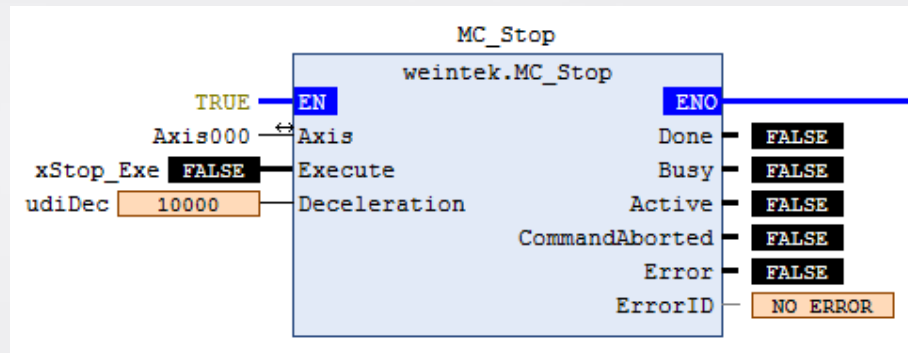


- Position : 相対位置を入力します。目標位置 = 現在位置 + 相対位置。
- Velocity : 移動速度です。数値が 0 ではいけません。
- Acceleration/Deceleration : 移動の加速度/減速度です。数値が 0 ではいけません。
- _BufferMode : 本パラメータを設定し、1 個前の移動コマンドに続いて連続移動を果たします。
- _IO_Ctrl : デジタル入力で移動コマンドをトリガーし、動作が完成したら、デジタル信号を出力します。
- 軸状態はトリガーされた後で〈Discrete Motion〉になり、目標位置に移動したら、また〈Standstill〉になります。

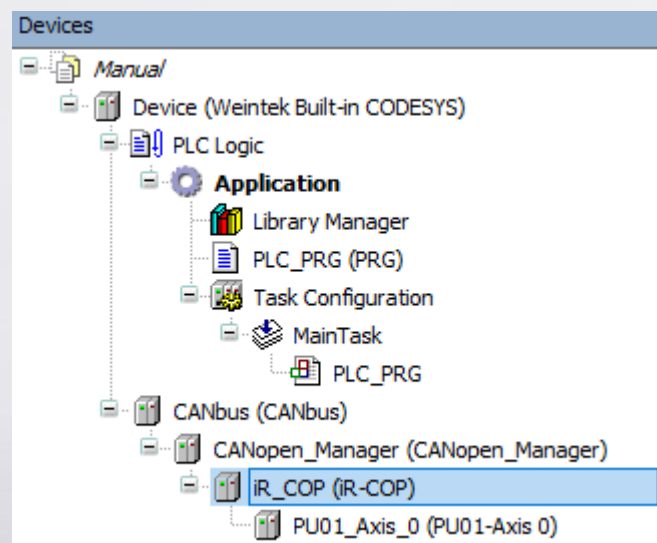
停止 MC_Stop

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

絶対/相対位置移動機能

xEnable_Power : システムを起動します。

xServe_ON : サーバーを起動します。

xHome_exe : TRUE=原点復帰を実行します。

xABS_exe : TRUE=絶対位置移動を実行します。

xINC_exe : TRUE=相対位置移動を実行します。

xTrig_Stop : TRUE=モーションを停止します。

xTrig_Reset : TRUE=iR-PU01-P のエラー状態を復帰します。

diPosition : 移動する絶対位置です。

diDistance : 移動する距離です。

udiSpeed : 移動の速度を設定します。

udiAcc : 移動の加速度を設定します。

udiDec : 移動の減速度を設定します。

eBuffer_Mode : 次の移動コマンドを Buffer に保存し、モーションの実行が完了したら、続けて実行します。

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。

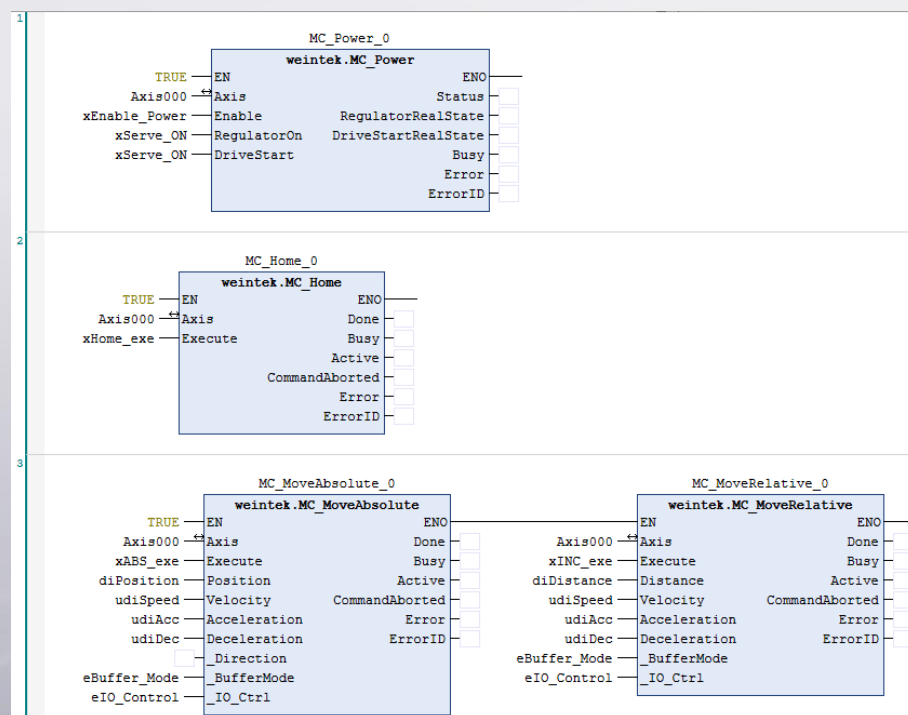
宣言

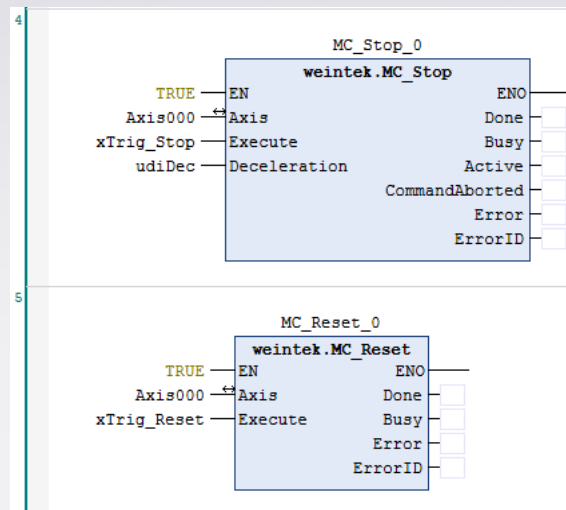
```

VAR
    // Axis reference
    Axis000 : Weintek.Axis_REF_Lite ;
    // Motion Control Function Block
    MC_Power_0: weintek.MC_Power ;
    MC_MoveAbsolute_0 : weintek.MC_MoveAbsolute ;
    MC_MoveRelative_0 : weintek.MC_MoveRelative ;
    MC_Home_0 : weintek.MC_Home ;
    MC_Stop_0: weintek.MC_Stop;
    MC_Reset_0: weintek.MC_Reset;
    // Positioning control Button
    xEnable_Power, xServe_ON, xABS_exe, xINC_exe, xHome_exe : BOOL ;
    xTrig_Stop, xTrig_Reset : BOOL;
    // Positioning parameter
    diPosition : DINT := 10000 ;
    diDistance : DINT := 10000 ;
    udiSpeed : UDINT := 1000 ;
    udiAcc : UDINT := 1000 ;
    udiDec : UDINT := 1000 ;
    eIO_Control : weintek.eMC_IO_CTRL ;
    eBuffer_Mode : weintek.eMC_BUFF_MODE ;
    // Variable
END_VAR
    
```

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 : 原点復帰機能ブロックです。
- 3 : 絶対位置移動及び相対位置移動機能ブロックです。
- 4 : 停止機能ブロックです。モーション中で xTrig_Stop をトリガーすると、モーションを停止できます。
- 5 : 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。

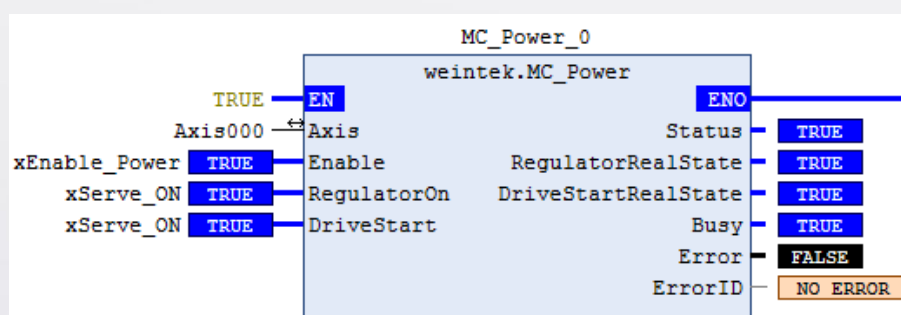
7. ログインして操作する

ここで原点復帰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。



原点復帰モーション

xHome_exe をトリガーして原点復帰を起動します。原点復帰モーションは自動的に下図の配置を参照して実行します。

9	16#6098:16#00	Axis 1 Homing method : PU01_Axis_1	27	8
10	16#607C:16#00	AX1 Home offset	0	32
11	16#6099:16#01	Speed during search for switch : AX1_PU01	10000	32
12	16#6099:16#02	Speed during search for zero : AX1_PU01	2000	32
13	16#609A:16#00	AX1 Homing aceleration : AX1_PU01	10000	32

[Speed during search for switch]に設定された速度で負方向へ原点センサーを探し、原点センサーが見つかったら、[Speed during search for zero]に設定された速度で正方向へ原点センサーを離れます。離れた後、Home offset 数値をロードし、原点復帰モーションが完了しました。

位置決め 絶対/相対位置移動



絶対位置移動

xABS_exe をトリガーして絶対位置移動を起動します。パルス = 10000 の位置に移動します。

Expression	Type	Value	Prepared value	A
Axis000	Weintek.Axis_REF_Lite	0		
_Delay_Cycles	BYTE	0		
_CMPT_PV	BOOL	FALSE		
Mapping_Q	unAxis_VAR_OUT			
Mapping_I	unAxis_VAR_IN			
Obj	stAxis_Mapping_In			
DI_B0	USINT	0		
Statusword	USINT	1591		
PositionActual	DINT	0		
ModeOpDisp	SINT	1		
VelocityActual	DINT	0		
PositionDemandInternal	DINT	10000		
DO_Status_B0	USINT	0		

MC_MoveAbsolute_0				MC_MoveRelative_0			
weintek.MC_MoveAbsolute_0	weintek.MC_MoveRelative_0	weintek.MC_MoveAbsolute_0	weintek.MC_MoveRelative_0	weintek.MC_MoveAbsolute_0	weintek.MC_MoveRelative_0	weintek.MC_MoveAbsolute_0	weintek.MC_MoveRelative_0
Axis000	Axis000	Axis000	Axis000	Axis000	Axis000	Axis000	Axis000
xABS_exe	xINC_exe	xABS_exe	xINC_exe	xABS_exe	xINC_exe	xABS_exe	xINC_exe
diPosition	diDistance	diPosition	diDistance	diPosition	diDistance	diPosition	diDistance
udiSpeed	udiSpeed	udiSpeed	udiSpeed	udiSpeed	udiSpeed	udiSpeed	udiSpeed
udiAcc	udiAcc	udiAcc	udiAcc	udiAcc	udiAcc	udiAcc	udiAcc
udiDec	udiDec	udiDec	udiDec	udiDec	udiDec	udiDec	udiDec
eBuffer_Mode	eBuffer_Mode	eBuffer_Mode	eBuffer_Mode	eBuffer_Mode	eBuffer_Mode	eBuffer_Mode	eBuffer_Mode
eIO_Control	eIO_Control	eIO_Control	eIO_Control	eIO_Control	eIO_Control	eIO_Control	eIO_Control
Done	Done	Done	Done	Done	Done	Done	Done
Busy	Busy	Busy	Busy	Busy	Busy	Busy	Busy
Active	Active	Active	Active	Active	Active	Active	Active
CommandAborted	CommandAborted	CommandAborted	CommandAborted	CommandAborted	CommandAborted	CommandAborted	CommandAborted
Error	Error	Error	Error	Error	Error	Error	Error
ErrorID	ErrorID	ErrorID	ErrorID	ErrorID	ErrorID	ErrorID	ErrorID
NO ERROR	NO ERROR	NO ERROR	NO ERROR	NO ERROR	NO ERROR	NO ERROR	NO ERROR

モーションが完了した後(MC_MoveAbsolute_0.Done=TRUE)、モーターはパルス = 10000 の位置に位置決めされます。軸パラメータレジスタの Axis000.Mapping_I » Obj » PositionDemandInternal で現在位置を確認できます。

相対位置移動

xINC_exe をトリガーして相対位置移動を起動します。現在位置(10000)に diDistance(10000)を足した位置に移動します。

Expression	Type	Value	Prepared value	A
Axis000	Weintek.Axis_REF_Lite	0		
_Delay_Cycles	BYTE	0		
_CMPT_PV	BOOL	FALSE		
Mapping_Q	unAxis_VAR_OUT			
Mapping_I	unAxis_VAR_IN			
Obj	stAxis_Mapping_In			
DI_B0	USINT	0		
Statusword	USINT	1591		
PositionActual	DINT	0		
ModeOpDisp	SINT	1		
VelocityActual	DINT	0		
PositionDemandInternal	DINT	20000		
DO_Status_B0	USINT	0		

MC_MoveAbsolute_0				MC_MoveRelative_0			
weintek.MC_MoveAbsolute_0	weintek.MC_MoveRelative_0	weintek.MC_MoveAbsolute_0	weintek.MC_MoveRelative_0	weintek.MC_MoveAbsolute_0	weintek.MC_MoveRelative_0	weintek.MC_MoveAbsolute_0	weintek.MC_MoveRelative_0
Axis000	Axis000	Axis000	Axis000	Axis000	Axis000	Axis000	Axis000
xABS_exe	xINC_exe	xABS_exe	xINC_exe	xABS_exe	xINC_exe	xABS_exe	xINC_exe
diPosition	diDistance	diPosition	diDistance	diPosition	diDistance	diPosition	diDistance
udiSpeed	udiSpeed	udiSpeed	udiSpeed	udiSpeed	udiSpeed	udiSpeed	udiSpeed
udiAcc	udiAcc	udiAcc	udiAcc	udiAcc	udiAcc	udiAcc	udiAcc
udiDec	udiDec	udiDec	udiDec	udiDec	udiDec	udiDec	udiDec
eBuffer_Mode	eBuffer_Mode	eBuffer_Mode	eBuffer_Mode	eBuffer_Mode	eBuffer_Mode	eBuffer_Mode	eBuffer_Mode
eIO_Control	eIO_Control	eIO_Control	eIO_Control	eIO_Control	eIO_Control	eIO_Control	eIO_Control
Done	Done	Done	Done	Done	Done	Done	Done
Busy	Busy	Busy	Busy	Busy	Busy	Busy	Busy
Active	Active	Active	Active	Active	Active	Active	Active
CommandAborted	CommandAborted	CommandAborted	CommandAborted	CommandAborted	CommandAborted	CommandAborted	CommandAborted
Error	Error	Error	Error	Error	Error	Error	Error
ErrorID	ErrorID	ErrorID	ErrorID	ErrorID	ErrorID	ErrorID	ErrorID
NO ERROR	NO ERROR	NO ERROR	NO ERROR	NO ERROR	NO ERROR	NO ERROR	NO ERROR

モーション完了した後(MC_MoveRelative_0.Done=TRUE)、モーターはパルス = 20000 の位置に位置決めされます。軸パラメータレジスタの Axis000.Mapping_I » Obj » PositionDemandInternal で現在位置を確認できます。

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.

DEM19005J_20200729