

使用說明

威綸函式庫說明

此文件介紹威綸函式庫的功能。

V1.0.2

目錄

1.介紹	1
2.加入 Weintek Library	1
3.PID 控制相關指令	3
3.1 PID(FB).....	3
3.2 PWM(FB).....	4
4.iR-COP 參數指令	6
4.1 Analog_Config(FB)	6
4.2 AO_Ch_Pa(FB).....	6
4.3 AI_Ch_Pa(FB).....	7
4.4 Analog_VI_Read(FB)	8
5.Motion Control 運動控制指令	9
5.1 MC_Power(FB)軸控功能啟動	9
5.2 MC_MoveVelocity(FB)速度模式移動	11
5.3 MC_MoveAbsolute(FB)絕對位置移動.....	14
5.4 MC_MoveRelative(FB)相對位置移動	18
5.5 MC_Home(FB)歸原點.....	21
5.6 MC_Stop(FB)停止	23
5.7 MC_Halt(FB)暫停	24
5.8 MC_Reset(FB)復歸	28
5.9 MC_Gear_Weintek(FB)手搖輪	29
附錄 A. FB Error Code	32
附錄 B. 歸原點方式.....	33

本文件中出現的其他公司名、產品名或商標均為各公司的商標或註冊商標。

本文件的資訊可能隨時變更，本公司將不另行通知。

Copyright© 2018 Weintek Labs., INC. All rights reserved.

1. 介紹

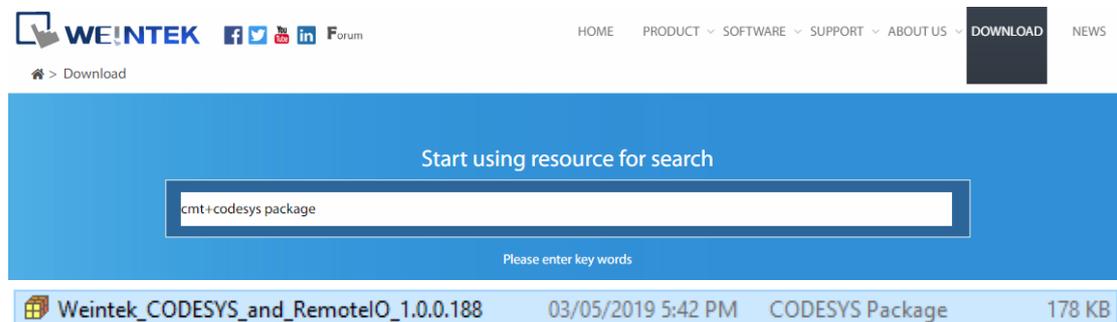
此文件介紹威綸函式庫的功能，未來將依照客戶使用需要，陸續發佈新增的功能塊。

2. 加入 Weintek Library

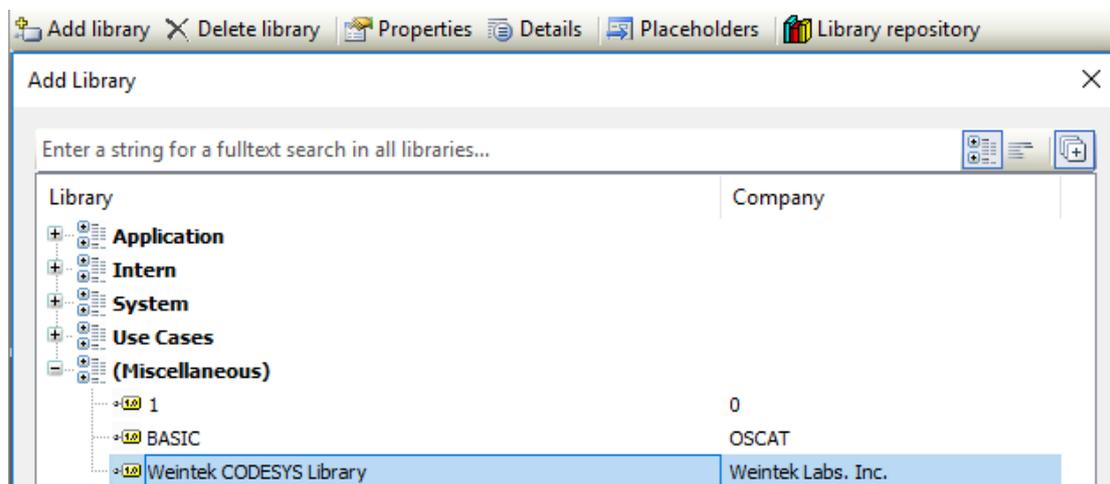
Step1. 在威綸官網下載 cMT+CODESYS Package

網址連結[<https://www.weintek.com/globalw/Download/Download.aspx>]

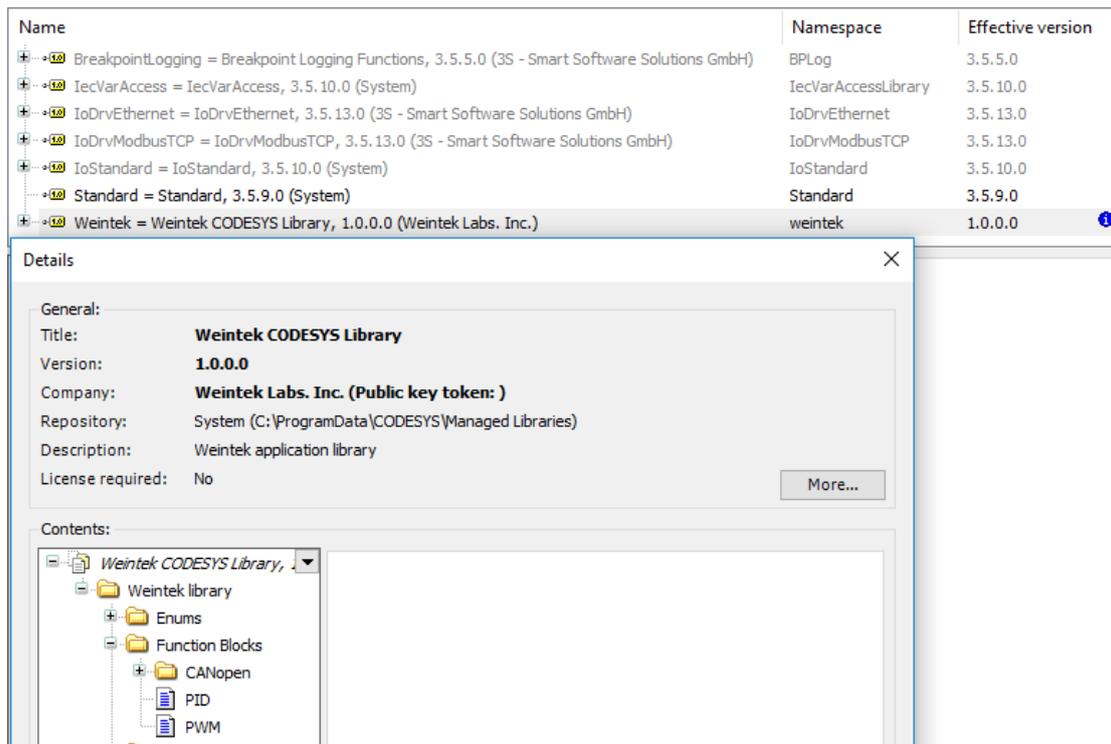
※版本號 1.0.0.188 或更新，已自動將 Weintek_CODESYS_Library 安裝在 CODESYS 內。



Step2. 在 [Add library] » (Miscellaneous) 下，加入 Weintek CODESYS Library



Step3. 開啟 [Details]，在 Function Blocks 下可看到功能塊。



Step4. 程式中宣告功能塊。

```

PROGRAM PLC_PRG
VAR
    PID : weintek.PID ;
END_VAR

PID (
    FUNCTION_BLOCK PID
    weintek codesys library, 1.0.0.0 (weintek labs. inc)

    VAR_INPUT    Manual           BOOL    Manual mode; MV := MV_Manual
    VAR_INPUT    Run              BOOL    Weintek PID FB enable
    VAR_INPUT    SV               REAL    Set Value
    VAR_INPUT    PV               REAL    Process Value
    VAR_INPUT    Dir              BOOL    False = heating ; True = cooling
    VAR_INPUT    MV_Manual        REAL    Manual mode Output Value
    VAR_INPUT    MV_Max           REAL    Output Max value
    VAR_INPUT    MV_Min           REAL    Output Min value
    VAR_INPUT    Auto_Deadband    REAL    Auto tuning dead band
    VAR_INPUT    Bias             REAL    Offset
    VAR_INPUT    Time_Base        REAL    Time Base = Second;
    VAR_INPUT    Error_Deadband   REAL    Actual MV dead band
    VAR_OUTPUT   MV               REAL    PID Auto output value
    VAR_OUTPUT   I_MV            REAL    Manual mode output value
    VAR_IN_OUT   Kp               REAL    Gain Proportional value
    VAR_IN_OUT   Ki               REAL    Gain Integral value
    VAR_IN_OUT   Kd               REAL    Gain Derivative value
    VAR_IN_OUT   Tf               REAL    Derivative-action time constant
    VAR_IN_OUT   Autotune        BOOL    Auto tuning enable
    
```

3. PID 控制相關指令

3.1 PID(FB)

➤ 功能：

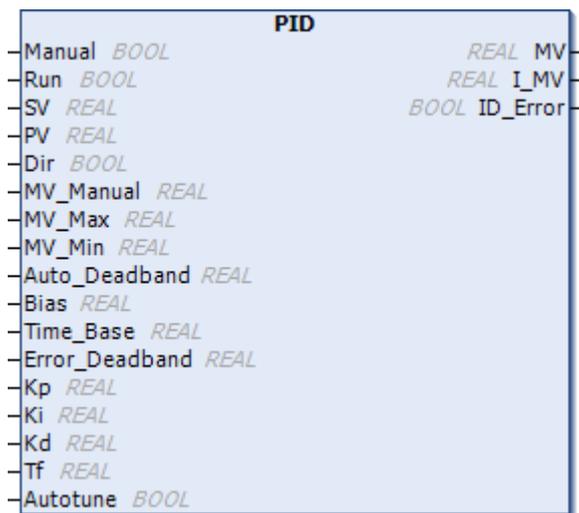
- 該指令為比例微分積分控制器。
- “Run”為 TRUE 時，開始計算 PID 功能塊輸出。
- “Run”&“AutoTune”為 TRUE 時，PID 功能塊進行自動校正功能，等待自動校正完成，“AutoTune”轉變為 FALSE。
- 當 $SV > PV$ 時，“Dir”設定為 FALSE； $SV < PV$ 時，“Dir”設定為 TRUE。
- “Time_Base”更新時間不可設定為 0。

➤ 方程式： $MV = K_p E + K_i \int_0^t E dt + K_d \frac{dE}{dt} + BIAS$

$$E = SV - PV, \text{ when Dir} = \text{FALSE}$$

$$E = PV - SV, \text{ when Dir} = \text{TRUE}$$

➤ 圖示：



➤ 參數說明：

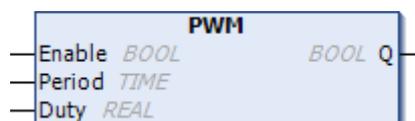
輸入參數	數據類型	定義	說明
Manual	BOOL	手動模式	TRUE=輸出手動值 Mout，FALSE=使用 PID 控制器。
Run	BOOL	啟動功能塊器	
SV	REAL	目標值	
PV	REAL	當前值	
Dir	BOOL	控制方向	FALSE=溫升/TRUE=溫降
MV_Manual	REAL	手動輸出值	
MV_Max	REAL	輸出最大值	輸出上限
MV_Min	REAL	輸出最小值	輸出下限

Auto_Deadband	REAL	自動校正靜止區間	啟用自動校正期間，目標值的 ±Auto_Deadband 為靜止區間
BIAS	REAL	前饋輸出	
Time_Base	REAL	更新時間	時間單位=秒(s)
Error_Deadband	REAL	偏差靜止區間	輸出=0 的區間
輸入輸出	數據類型	定義	說明
Tf	REAL	微分區間	微分量分次輸出 $MV_d = \frac{T_f * MV_d(\text{last cycle}) + K_d * E}{T_f + T_s}$ MV _d =微分項輸出 T _s =Time_Base
Kp	REAL	比例常數	
Ki	REAL	積分常數	
Kd	REAL	微分常數	
Autotune	BOOL	自動校正	啟動自動校正功能後，必須等待校正完畢(Tf、Kp、Ki、Kd 更新)，PID 控制器才會進行控制
輸出參數	數據類型	定義	說明
MV	REAL	PID 輸出	
I_MV	REAL	累積積分量	
ID_Error	BOOL	裝置編號錯誤	當使用非威綸 CODESYS 控制器執行 PID 功能塊，會發生錯誤。

※如何使用請參閱 iR_Application_Oven_Demo 烤箱應用範例。

3.2 PWM(FB)

- 功能：
 - “Enable”為 TRUE 時，輸出 PWM 信號。
 - “Q”為 TRUE 的時間=“Period” * “Duty”
- 圖示：



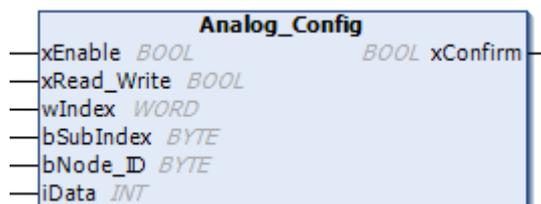
➤ 參數說明：

輸入參數	數據類型	定義	說明
Enable	BOOL	啟用	TRUE=啟動
Period	TIME	時間週期	一次 TRUE & FALSE=一個週期
Duty	REAL	運作比例	Period時間內 Q 的輸出時間比例，範圍 0~100%
輸出參數	數據類型	定義	說明
Q	BOOL	輸出	啟動後從 TRUE 開始週期

4. iR-COP 參數指令

4.1 Analog_Config(FB)

- 功能：
 - 此功能塊只能對 iR-COP 下的類比模組進行參數讀取/寫入。
 - 當"xEnable"由 FALSE 轉變為 TRUE 時，讀取/寫入一個類比模組參數。
- 圖示：

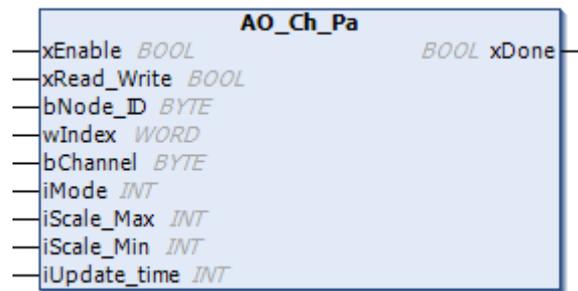


- 參數說明：

輸入參數	數據類型	定義	說明
xEnable	BOOL	啟動	給予一個上升緣即觸發
xRead_Write	BOOL	讀/寫切換	TRUE=寫入；FALSE=讀取
wIndex	WORD	索引	物件字典的索引
bSubIndex	BYTE	副索引	物件字典的副索引
bNode_ID	BYTE	站號	iR-COP 的站號
輸出參數	數據類型	定義	說明
xConfirm	BOOL	完成	讀取/寫入完成
輸入輸出	數據類型	定義	說明
iData	INT	操作資料	讀取/寫入的資料

4.2 AO_Ch_Pa(FB)

- 功能：
 - 此功能塊只能對 iR-COP 下的類比模組進行參數讀取/寫入，AQ04-VI 及 AM06-VI 可使用。
 - 以通道為單位讀取/寫入類比輸出參數。
 - 當"xEnable"由 FALSE 轉變為 TRUE 時，讀取/寫入一個通道的類比參數。
- 圖示：



➤ 參數說明：

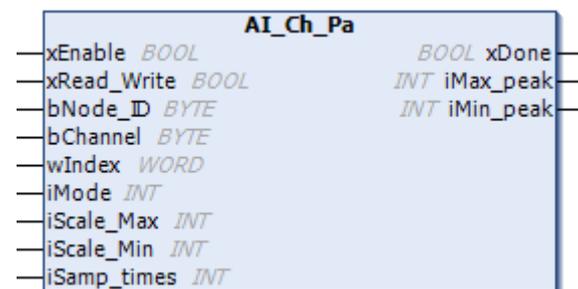
輸入參數	數據類型	定義	說明
xEnable	BOOL	啟動	給予一個上升緣即觸發
xRead_Write	BOOL	讀/寫切換	TRUE=寫入；FALSE=讀取
bNode_ID	BYTE	站號	iR-COP 的站號
wIndex	WORD	索引	物件字典的索引
bChannel	BYTE	操作通道	通道 0~3
輸出參數	數據類型	定義	說明
xDone	BOOL	完成	功能塊完成
輸入輸出	數據類型	定義	說明
iMode	INT	通道模式	
iScale_Max	INT	通道最大值	
iScale_Min	INT	通道最小值	
iUpdate_Time	INT	輸出時間	

4.3 AI_Ch_Pa(FB)

➤ 功能：

- 此功能塊只能對 iR-COP 下的類比模組進行參數讀取/寫入，AI04-VI 及 AM06-VI 可使用。
- 以通道為單位讀取/寫入類比輸入參數。
- 當“xEnable”由 FALSE 轉變為 TRUE 時，讀取/寫入一個通道的類比參數。

➤ 圖示：



➤ 參數說明：

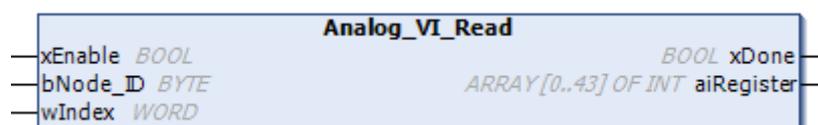
輸入參數	數據類型	定義	說明
xEnable	BOOL	啟動	給予一個上升緣即觸發
xRead_Write	BOOL	讀/寫切換	TRUE=寫入；FALSE=讀取
bNode_ID	BYTE	站號	iR-COP 的站號
wIndex	WORD	索引	物件字典的索引
bChannel	BYTE	操作通道	通道 0~3
輸出參數	數據類型	定義	說明
xDone	BOOL	完成	功能塊完成
iMax_Peak	INT	通道峰值+	
iMin_Peak	INT	通道峰值-	
輸入輸出	數據類型	定義	說明
iMode	INT	通道模式	
iScale_Max	INT	通道最大值	
iScale_Min	INT	通道最小值	
iSamp_Times	INT	濾波取樣次數	

4.4 Analog_VI_Read(FB)

➤ 功能：

- 此功能塊只能對 iR-COP 下的類比模組進行參數讀取/寫入，讀取所有類比參數，AI04-VI、AQ04-VI 及 AM06-VI 可使用。
- 當“xEnable”由 FALSE 轉變為 TRUE 時，讀取模組內所有的類比參數。

➤ 圖示：



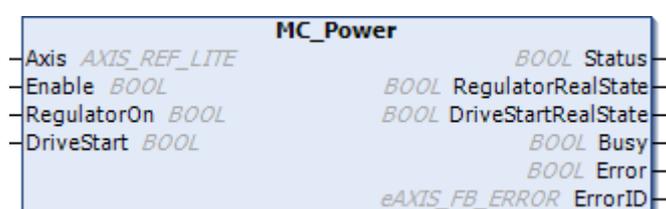
➤ 參數說明：

輸入參數	數據類型	定義	說明
xEnable	BOOL	啟動	給予一個上升緣即觸發
bNode_ID	BYTE	站號	iR-COP 的站號
wIndex	WORD	索引	物件字典的索引
輸出參數	數據類型	定義	說明
xDone	BOOL	完成	功能塊完成
aiRegister	INT[0..43]	類比模組暫存器	讀取類比模組共 44 個暫存器資料

5. Motion Control 運動控制指令

5.1 MC_Power(FB)軸控功能啟動

- 功能：
 - 當"Enable"、"RegulatorOn"、"DriveStart"皆為 TRUE 時，指定軸"Axis"進入待命狀態<Standstill>。軸在待命狀態<Standstill>下可進行運動控制。
 - 當"Enable"、"RegulatorOn"數值為 FALSE，軸狀態進入不作動狀態<Disable>。
 - 當"DriveStart"數值為 FALSE，指定軸快速停止，可以當作急停使用。
- 圖示：



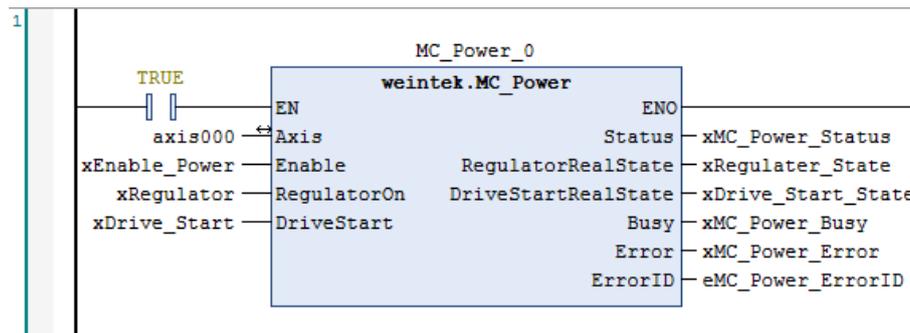
- 參數說明：

輸入參數	數據類型	定義	說明
Enable	BOOL	執行功能塊	執行功能塊時保持數值為 TRUE。
RegulatorOn	BOOL	運動控制系統開關	FALSE：關閉運動控制。 TRUE：啟動運動控制，啟動後可下任何運動指令。
DriveStart	BOOL	快速停止功能開關	FALSE：啟動快速停止功能。 TRUE：關閉快速停止功能。
輸出參數	數據類型	定義	說明
Status	BOOL	功能塊狀態	TRUE：指定軸已進入待命狀態。
RegulatorRealState	BOOL	運動控制狀態	FALSE：運動控制系統尚未啟動。 TRUE：運動控制系統已啟動，可下任何運動指令。
DriveStartRealState	BOOL	快速停止功能狀態	FALSE：快速停止功能已啟動。 TRUE：快速停止功能未啟動。

Busy	BOOL	功能塊狀態	TRUE：功能塊已被執行。
Error	BOOL	錯誤狀態	TRUE：當 Error 發生時。
ErrorID	BOOL	錯誤碼	功能塊錯誤碼, 參考"附錄 A"
輸入輸出	數據類型	定義	說明
Axis	AXIS_REF_LITE	軸參數	指定欲操作軸。

➤ 程式編輯：

■ LD：

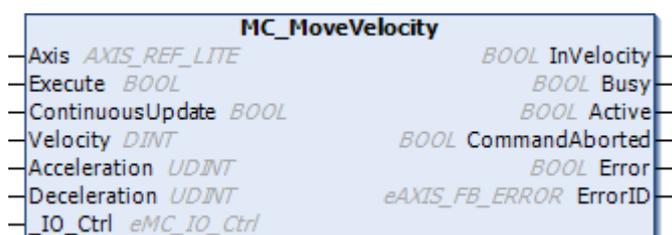


■ ST：

```
// MC_Power function block
MC_Power_1(
    Axis:= Axis000,
    Enable:= xEnable_Power,
    RegulatorOn:= xRegulator,
    DriveStart:= xDrive_Start,
    Status=> xMC_Power_Status,
    RegulatorRealState=> xRegulator_State,
    DriveStartRealState=> xDrive_Start_State,
    Busy=>xMC_Power_Busy ,
    Error=> xMC_Power_Error,
    ErrorID=> eMC_Power_ErrorID);
```

5.2 MC_MoveVelocity(FB)速度模式移動

- 功能：
 - 指定目標速度作速度控制。
 - “Execute”由 FALSE 轉變為“TRUE”觸發速度控制。
 - “Velocity” 速度參數為正=正轉；速度參數為負=負轉。
 - 試運轉 JOG 可使用 MC_MoveVelocity 功能塊，詳細可參考 Demo project “DEM19004_iR_Application_JOG_Demo_20190906”。
- 圖示：



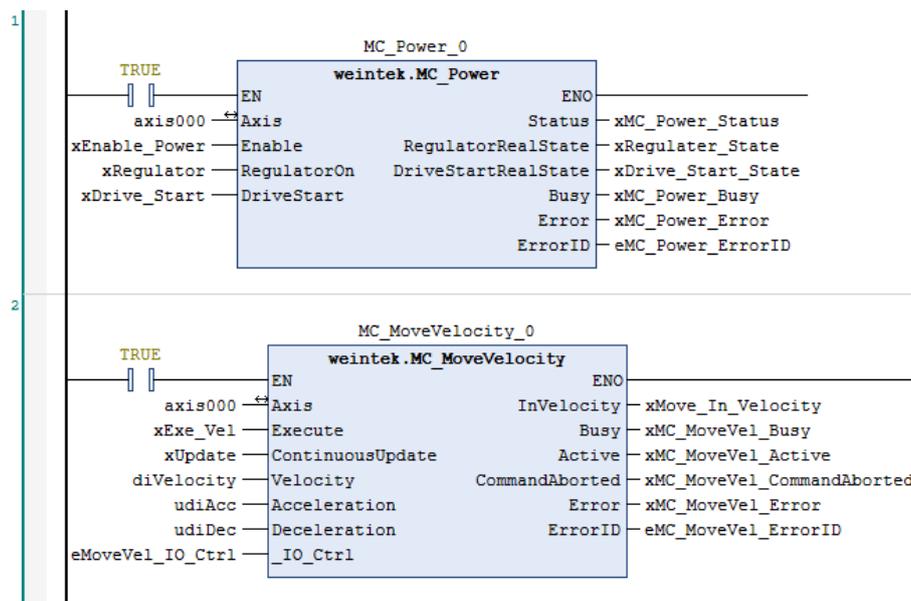
- 參數說明：

輸入參數	數據類型	定義	說明
Execute	BOOL	執行	當數值由 FALSE 轉變為 TRUE 時執行。
ContinuousUpdate	BOOL	運動中更新	TRUE：在運動中可以改變目標速度。
Velocity	DINT	運動速度	指定目標速度。單位為使用者單位/s。
Acceleration	UDINT	加速度	指定加速度。指定加速度單位為使用者單位/s ²
Deceleration	UDINT	減速度	指定減速度。指定減速度單位為使用者單位/s ²
_IO_Ctrl	eMC_IO_Ctrl	IO 控制選項	None：不使用 I0~I2：使用數位輸入觸發運動
輸出參數	數據類型	定義	說明
InVelocity	BOOL	速度到達	TRUE：到達目標速度。
Busy	BOOL	功能塊狀態	TRUE：功能塊已被執行。
Active	BOOL	運動狀態	TRUE：當前運動中功能塊。
CommandAborted	BOOL	命令中止	TRUE：功能塊被其他功能塊或事件而中止運動。

Error	BOOL	錯誤狀態	TRUE：當 Error 發生時。
ErrorID	BOOL	錯誤碼	功能塊錯誤碼，參考“附錄 A”
輸入輸出	數據類型	定義	說明
Axis	AXIS_REF_LITE	軸參數	指定欲操作軸。

➤ 程式編輯：

■ LD：



■ ST：

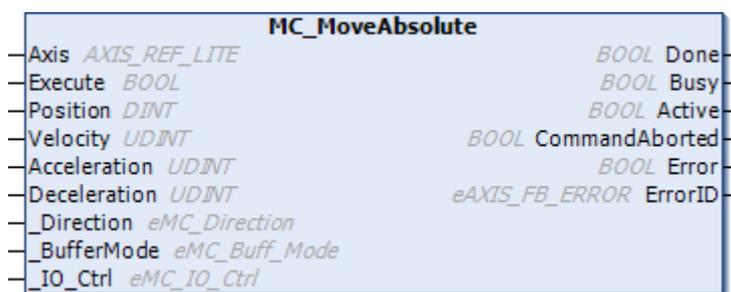
```
// MC_Power function block
MC_Power_1(
    Axis:= Axis000,
    Enable:= xEnable_Power,
    RegulatorOn:= xRegulator,
    DriveStart:= xDrive_Start,
    Status=> xMC_Power_Status,
    RegulatorRealState=> xRegulator_State,
    DriveStartRealState=> xDrive_Start_State,
    Busy=>xMC_Power_Busy ,
    Error=> xMC_Power_Error,
    ErrorID=> eMC_Power_ErrorID);

// MC_Velocity function block
MC_MoveVelocity_0(
```

```
Axis:= Axis000,  
Execute:= xExe_Vel,  
ContinuousUpdate:= xUpdate,  
Velocity:= diVelocity,  
Acceleration:= udiAcc,  
Deceleration:= udiDec,  
_IO_Ctrl:= eMoveVel_IO_Ctrl,  
InVelocity=> xMove_In_Velocity,  
Busy=> xMC_MoveVel_Busy,  
Active=> xMC_MoveVel_Active,  
CommandAborted=> xMC_MoveVel_CommandAborted,  
Error=> xMC_MoveVel_Error,  
ErrorID=> eMC_MoveVel_ErrorID);
```

5.3 MC_MoveAbsolute(FB)絕對位置移動

- 功能：
 - 指定目標絕對位置作定位控制。
 - “Execute”由 FALSE 轉變為“TRUE”觸發定位控制。
 - 就算尚未歸原點，也可以執行定位控制。
 - 可在輸入變數指定運動速度、加速度、減速度、Buffer Mode 與 IO 觸發控制。
 - 若為旋轉軸，可在“_Direction”指定旋轉方向。
- 圖示：



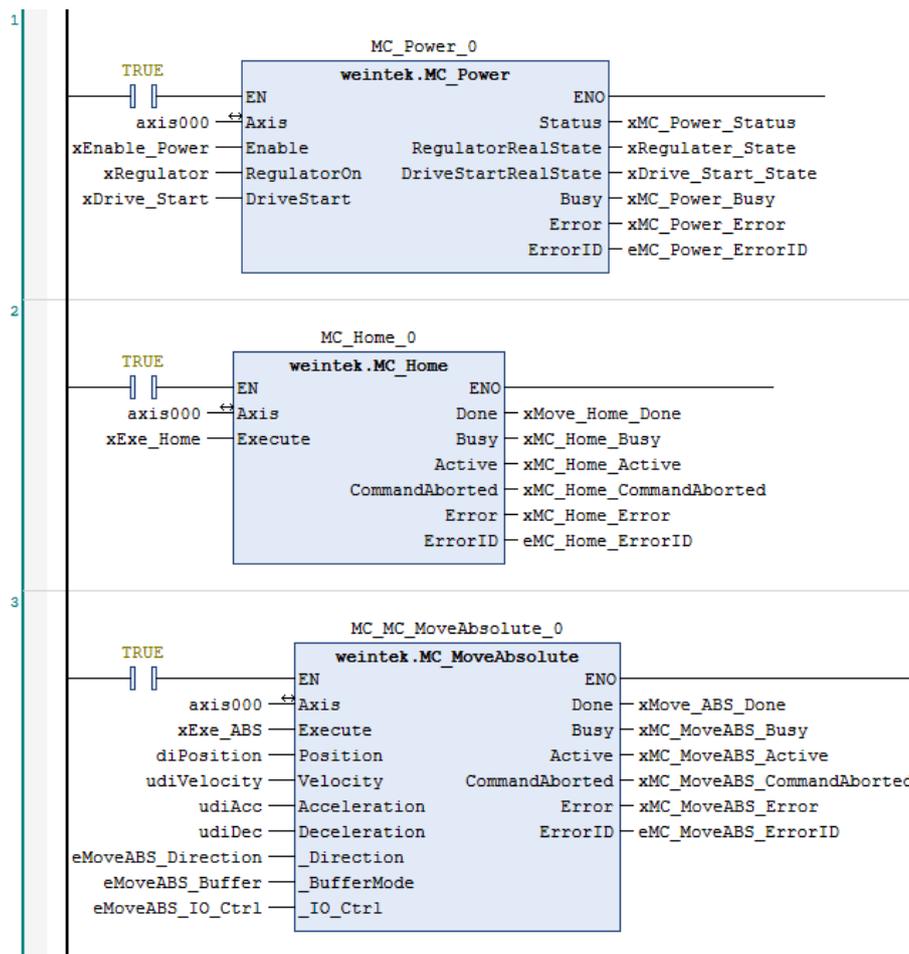
- 參數說明：

輸入參數	數據類型	定義	說明
Execute	BOOL	執行	當數值由 FALSE 轉變為 TRUE 時執行。
Position	DINT	移動位置	指定目標絕對位置，單位為使用者單位。
Velocity	UDINT	運動速度	指定目標速度。單位為使用者單位/s
Acceleration	UDINT	加速度	指定加速度。指定加速度單位為使用者單位/s ²
Deceleration	UDINT	減速度	指定減速度。指定減速度單位為使用者單位/s ²
_Direction	eMC_Direction	移動方向	Positive：正轉 ShortestWay：最短路徑 Negative：負轉 Current：上次旋轉方向 (只有旋轉軸能設定旋轉方向)
_BufferMode	eMC_Buff_Mode	連續定位	Aborting：不使用。 Buffered：定位接續前一個運動。

			BlendingPrev：速度與定位接續前一個運動
_IO_Ctrl	eMC_IO_Ctrl	IO 控制選項	None：不使用 I0~I2：使用數位輸入觸發運動 O0~O2：使用運動完成輸出
輸出參數	數據類型	定義	說明
Done	BOOL	功能塊完成	TRUE：到達目標位置
Busy	BOOL	功能塊狀態	TRUE：功能塊已被執行。
Active	BOOL	運動狀態	TRUE：當前運動中功能塊。
CommandAborted	BOOL	命令中止	TRUE：功能塊被其他功能塊或事件而中止運動。
Error	BOOL	錯誤狀態	TRUE：當 Error 發生時。
ErrorID	BOOL	錯誤碼	功能塊錯誤碼，參考“附錄 A”
輸入輸出	數據類型	定義	說明
Axis	AXIS_REF_LITE	軸參數	指定欲操作軸。

➤ 程式編輯範例：

- LD：



■ ST :

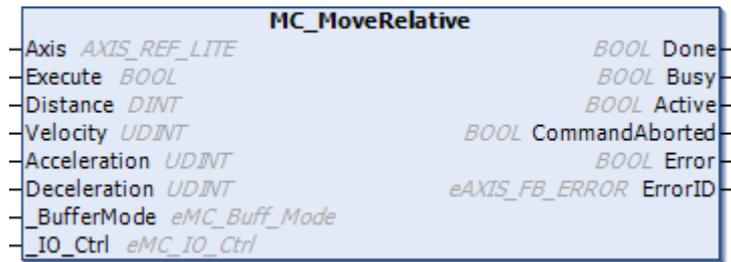
```
// MC_Power function block
MC_Power_0(
    Axis:= Axis000,
    Enable:= xEnable_Power,
    RegulatorOn:= xRegulator,
    DriveStart:= xDrive_Start,
    Status=> xMC_Power_Status,
    RegulatorRealState=> xRegulator_State,
    DriveStartRealState=> xDrive_Start_State,
    Busy=>xMC_Power_Busy ,
    Error=> xMC_Power_Error,
    ErrorID=> eMC_Power_ErrorID);

// MC_Home function block
MC_Home_0(
    Axis:= Axis000,
    Execute:= xExe_Home,
```

```
Done=> xMove_Home_Done,  
Busy=> xMC_Home_Busy,  
Active=> xMC_Home_Active,  
CommandAborted=> xMC_Home_CommandAborted,  
Error=> xMC_Home_Error,  
ErrorID=> eMC_Home_ErrorID);  
// MC_MoveAbsolute function block  
MC_MC_MoveAbsolute_0(  
  Axis:= Axis000,  
  Execute:= xExe_ABS,  
  Position:= diPosition,  
  Velocity:= udiVelocity,  
  Acceleration:= udiAcc,  
  Deceleration:= udiDec,  
  _Direction:= eMoveABS_Direction,  
  _BufferMode:= eMoveABS_Buffer,  
  _IO_Ctrl:= eMoveABS_IO_Ctrl,  
  Done=> xMove_ABS_Done,  
  Busy=> xMC_MoveABS_Busy,  
  Active=> xMC_MoveABS_Active,  
  CommandAborted=> xMC_MoveABS_CommandAborted,  
  Error=> xMC_MoveABS_Error,  
  ErrorID=> eMC_MoveABS_ErrorID);
```

5.4 MC_MoveRelative(FB)相對位置移動

- 功能：
 - 指定從收到命令時，當前位置開始計算的移動距離作定控制。
 - “Execute”由 FALSE 轉變為“TRUE”觸發定位控制。
 - 可在輸入變數指定運動速度、加速度、減速度、Buffer Mode 與 IO 觸發控制。
- 圖示：

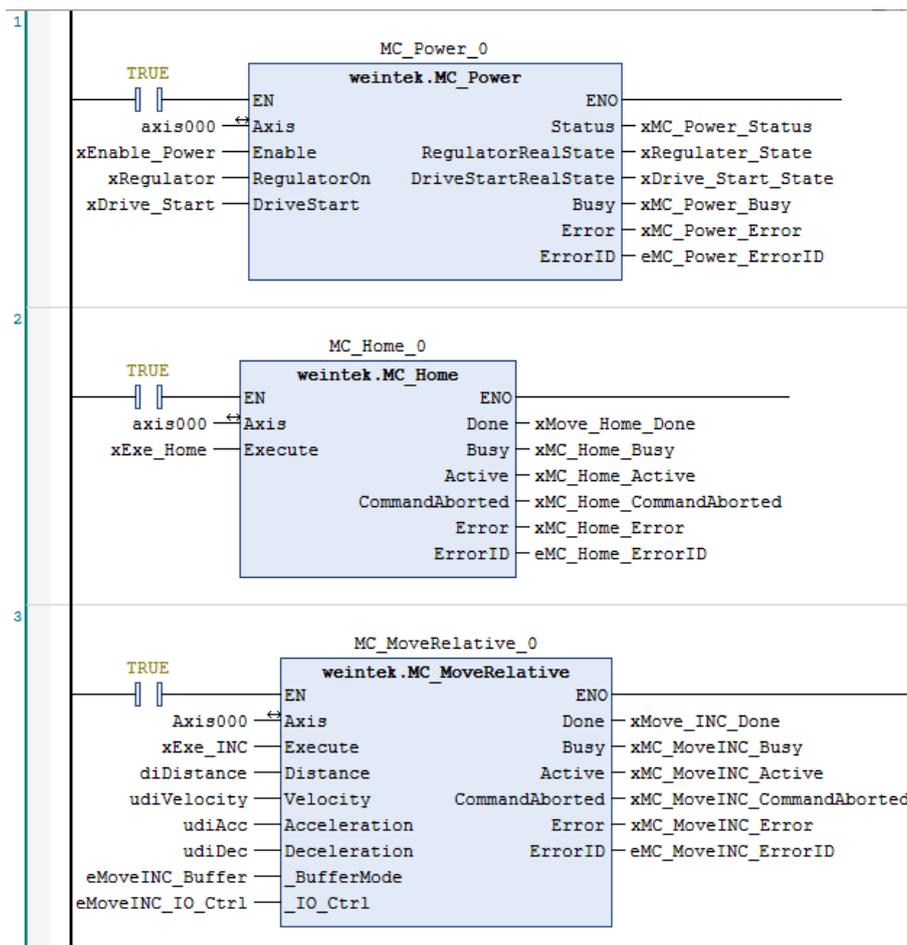


- 參數說明：

輸入參數	數據類型	定義	說明
Execute	BOOL	執行	當數值由 FALSE 轉變為 TRUE 時執行。
Distance	DINT	移動位置	指定從命令位置開始計算的移動距離。單位為使用者單位。
Velocity	UDINT	運動速度	指定目標速度。單位為使用者單位/s
Acceleration	UDINT	加速度	指定加速度。指定加速度單位為使用者單位/s ²
Deceleration	UDINT	減速度	指定減速度。指定減速度單位為使用者單位/s ²
_BufferMode	eMC_Buff_Mode	連續定位	Aborting：不使用。 Buffered：定位接續前一個運動。 BlendingPrev：速度與定位接續前一個運動
_IO_Ctrl	eMC_IO_Ctrl	IO 控制選項	None：不使用 I0~I2：使用數位輸入觸發運動 O0~O2：使用運動完成輸出
輸出參數	數據類型	定義	說明

Done	BOOL	功能塊完成	
Busy	BOOL	功能塊狀態	TRUE：功能塊已被執行。
Active	BOOL	運動狀態	TRUE：當前運動中功能塊。
CommandAborted	BOOL	命令中止	TRUE：功能塊被其他功能塊或事件而中止運動。
Error	BOOL	錯誤狀態	TRUE：當 Error 發生時。
ErrorID	BOOL	錯誤碼	功能塊錯誤碼，參考”附錄 A”
輸入輸出	數據類型	定義	說明
Axis	AXIS_REF_LITE	軸參數	指定欲操作軸。

■ LD :



■ ST :

// MC_Power function block

```
MC_Power_0(  
    Axis:= Axis000,  
    Enable:= xEnable_Power,  
    RegulatorOn:= xRegulator,  
    DriveStart:= xDrive_Start,  
    Status=> xMC_Power_Status,  
    RegulatorRealState=> xRegulator_State,  
    DriveStartRealState=> xDrive_Start_State,  
    Busy=>xMC_Power_Busy ,  
    Error=> xMC_Power_Error,  
    ErrorID=> eMC_Power_ErrorID);  
// MC_Power function block  
MC_Home_0(  
    Axis:= Axis000,  
    Execute:= xExe_Home,  
    Done=> xMove_Home_Done,  
    Busy=> xMC_Home_Busy,  
    Active=> xMC_Home_Active,  
    CommandAborted=> xMC_Home_CommandAborted,  
    Error=> xMC_Home_Error,  
    ErrorID=> eMC_Home_ErrorID);  
// MC_Relative function block  
MC_MoveRelative_0(  
    Axis:= Axis000,  
    Execute:= xExe_INC,  
    Distance:= diDistance,  
    Velocity:= udiVelocity,  
    Acceleration:= udiAcc,  
    Deceleration:= udiDec,  
    _BufferMode:= eMoveINC_Buffer,  
    _IO_Ctrl:= eMoveINC_IO_Ctrl,  
    Done=> xMove_INC_Done,  
    Busy=> xMC_MoveINC_Busy,  
    Active=> xMC_MoveINC_Active,  
    CommandAborted=> xMC_MoveINC_CommandAborted,  
    Error=> xMC_MoveINC_Error,  
    ErrorID=> eMC_MoveINC_ErrorID );
```

5.5 MC_Home(FB)歸原點

➤ 功能：

- 當“Execute”由 FALSE 轉變為 TRUE 時對指定軸“Axis”執行歸原點運動。
- 參考物件字典 6098 歸原點方式執行歸原點動作。
- 可設定 1~37 種歸原點方式，可在 CODESYS 使用[Add SDOs]寫入歸原點方式。
- 詳細歸原點方式可參考“附錄 B”。

➤ 圖示：

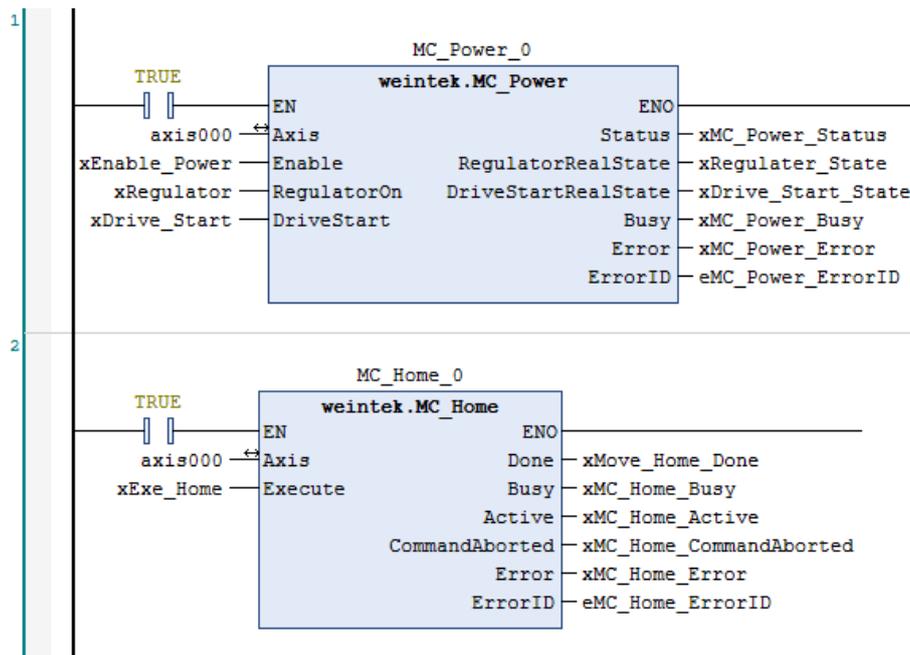


➤ 參數說明：

輸入參數	數據類型	定義	說明
Execute	BOOL	執行	當數值由 FALSE 轉變為 TRUE 時執行。
輸出參數	數據類型	定義	說明
Done	BOOL	功能塊完成	TRUE：歸原點運動完成。
Busy	BOOL	功能塊狀態	TRUE：功能塊已被執行。
Active	BOOL	運動狀態	TRUE：當前運動中功能塊。
CommandAborted	BOOL	命令中止	TRUE：功能塊被其他功能塊或事件而中止運動。
Error	BOOL	錯誤狀態	TRUE：當 Error 發生時。
ErrorID	BOOL	錯誤碼	功能塊錯誤碼，參考“附錄 A”
輸入輸出	數據類型	定義	說明
Axis	AXIS_REF_LITE	軸參數	指定欲操作軸。

➤ 程式編輯：

- LD：



■ ST :

```
// MC_Power function block
MC_Power_0(
    Axis:= Axis000,
    Enable:= xEnable_Power,
    RegulatorOn:= xRegulator,
    DriveStart:= xDrive_Start,
    Status=> xMC_Power_Status,
    RegulatorRealState=> xRegulator_State,
    DriveStartRealState=> xDrive_Start_State,
    Busy=>xMC_Power_Busy ,
    Error=> xMC_Power_Error,
    ErrorID=> eMC_Power_ErrorID);

// MC_Power function block
MC_Home_0(
    Axis:= Axis000,
    Execute:= xExe_Home,
    Done=> xMove_Home_Done,
    Busy=> xMC_Home_Busy,
    Active=> xMC_Home_Active,
    CommandAborted=> xMC_Home_CommandAborted,
    Error=> xMC_Home_Error,
    ErrorID=> eMC_Home_ErrorID);
```

5.6 MC_Stop(FB)停止

- 功能：
 - 停止運動，由軸當前速度減速到速度為 0。
 - “Execute”由 FALSE 轉變為“TRUE”觸發減速停止運動。
 - MC_Stop 功能塊觸發後到功能塊完成前，無法對軸下其他運動命令。
- 圖示：

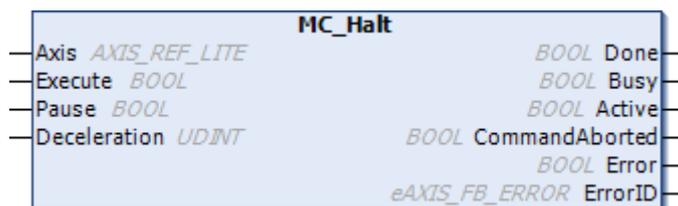


- 參數說明：

輸入參數	數據類型	定義	說明
Execute	BOOL	執行	當數值由 FALSE 轉變為 TRUE 時執行。
Decleration	UDINT	減速度	指定減速度。指定減速度單位為使用者單位/s ²
輸出參數	數據類型	定義	說明
Done	BOOL	功能塊完成	TRUE：速度=0 且 Execute=FALSE。
Busy	BOOL	功能塊狀態	TRUE：功能塊已被執行。
Active	BOOL	運動狀態	TRUE：當前運動中功能塊。
CommandAborted	BOOL	命令中止	TRUE：功能塊被其他功能塊或事件而中止運動。
Error	BOOL	錯誤狀態	TRUE：當 Error 發生時。
ErrorID	BOOL	錯誤碼	功能塊錯誤碼，參考“附錄 A”
輸入輸出	數據類型	定義	說明
Axis	AXIS_REF_LITE	軸參數	指定欲操作軸。

5.7 MC_Halt(FB)暫停

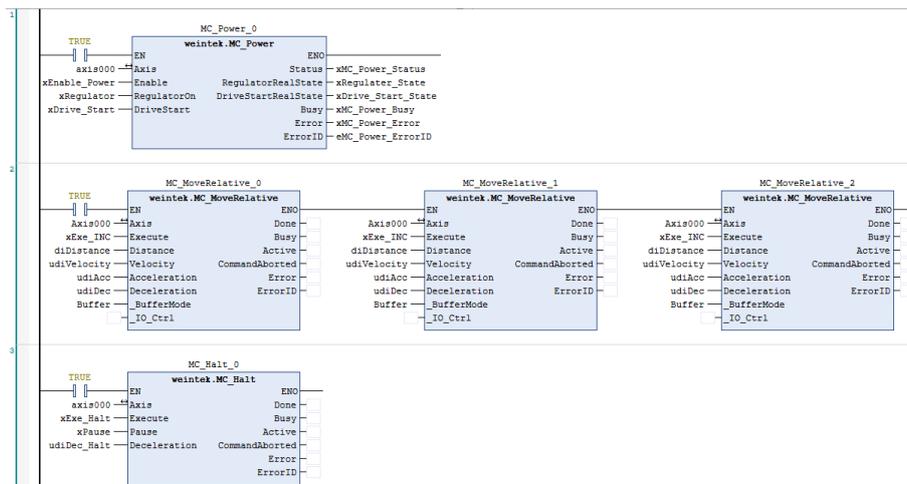
- 功能：
 - 暫停當前運動，減速到 0 完成，減速中可啟動其他運動功能塊。
 - “Execute”由 FALSE 轉變為“TRUE”觸發減速暫停運動。
 - “Pause”為 TRUE 時執行“Execute”暫停運動，當“Execute”恢復為 FALSE 可繼續原先執行的運動控制。
- 圖示：



- 參數說明：

輸入參數	數據類型	定義	說明
Execute	BOOL	執行	當數值由 FALSE 轉變為 TRUE 時執行。
Pause	BOOL	暫停	不中斷 Buffer Mode 連續定位。
Deceleration	UDINT	減速度	指定減速度。指定減速度單位為使用者單位/ s^2
輸出參數	數據類型	定義	說明
Done	BOOL	功能塊完成	TRUE：速度=0 且 Execute=FALSE。
Busy	BOOL	功能塊狀態	TRUE：功能塊已被執行。
Active	BOOL	運動狀態	TRUE：當前運動中功能塊。
CommandAborted	BOOL	命令中止	TRUE：功能塊被其他功能塊或事件而中止運動。
Error	BOOL	錯誤狀態	TRUE：當 Error 發生時。
ErrorID	BOOL	錯誤碼	功能塊錯誤碼，參考“附錄 A”
輸入輸出	數據類型	定義	說明
Axis	AXIS_REF_LITE	軸參數	指定欲操作軸。

- 程式編輯：
 - LD：



■ ST :

```
// MC_Power function block
MC_Power_0(
    Axis:= Axis000,
    Enable:= xEnable_Power,
    RegulatorOn:= xRegulator,
    DriveStart:= xDrive_Start,
    Status=> xMC_Power_Status,
    RegulatorRealState=> xRegulator_State,
    DriveStartRealState=> xDrive_Start_State,
    Busy=>xMC_Power_Busy ,
    Error=> xMC_Power_Error,
    ErrorID=> eMC_Power_ErrorID);

// Buffer mode continuous positioning function block
MC_MoveRelative_0(
    Axis:= Axis000,
    Execute:= xExe_INC,
    Distance:= diDistance,
    Velocity:= udiVelocity,
    Acceleration:= udiAcc,
    Deceleration:= udiDec,
    _BufferMode:= Buffer,
    _IO_Ctrl:= ,
    Done=> ,
    Busy=> ,
    Active=> ,
```

```
    CommandAborted=> ,
    Error=> ,
    ErrorID=> );
MC_MoveRelative_1(
    Axis:= Axis000,
    Execute:= xExe_INC,
    Distance:= diDistance,
    Velocity:= udiVelocity,
    Acceleration:= udiAcc,
    Deceleration:= udiDec,
    _BufferMode:= Buffer,
    _IO_Ctrl:= ,
    Done=> ,
    Busy=> ,
    Active=> ,
    CommandAborted=> ,
    Error=> ,
    ErrorID=> );
MC_MoveRelative_2(
    Axis:= Axis000,
    Execute:= xExe_INC,
    Distance:= diDistance,
    Velocity:= udiVelocity,
    Acceleration:= udiAcc,
    Deceleration:= udiDec,
    _BufferMode:= Buffer,
    _IO_Ctrl:= ,
    Done=> ,
    Busy=> ,
    Active=> ,
    CommandAborted=> ,
    Error=> ,
    ErrorID=> );
// Buffer mode continuous positioning function block
// MC_Halt function block
MC_Halt_0(
    Axis:= Axis000,
```

```
Execute:= xExe_Halt,  
Pause:= xPause ,  
Deceleration:= udiDec_Halt,  
Done=> ,  
Busy=> ,  
Active=> ,  
CommandAborted=> ,  
Error=> ,  
ErrorID=> );
```

5.8 MC_Reset(FB)復歸

➤ 功能：

- MC_Reset 指令對指定軸開始錯誤清除流程，復歸錯誤狀態。
- “Execute”由 FALSE 轉變為“TRUE”觸發復歸流程。
- 軸發生錯誤時才能開始錯誤清除流程。
- 如果驅動器發生錯誤，必須先清除驅動器錯誤再啟動 MC_Reset。

➤ 圖示：



➤ 參數說明：

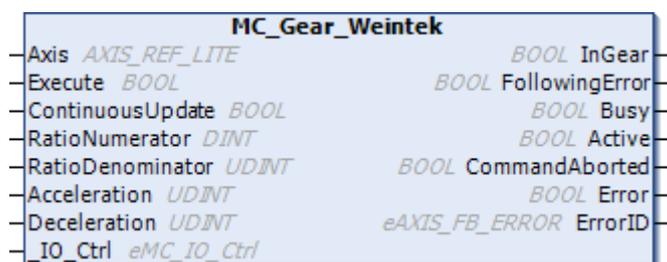
輸入參數	數據類型	定義	說明
Execute	BOOL	執行	當數值由 FALSE 轉變為 TRUE 時執行。
輸出參數	數據類型	定義	說明
Done	BOOL	功能塊完成	TRUE：復歸完成。
Busy	BOOL	功能塊狀態	TRUE：功能塊已被執行。
Error	BOOL	錯誤狀態	TRUE：當 Error 發生時。
ErrorID	BOOL	錯誤碼	功能塊錯誤碼，參考“附錄 A”
輸入輸出	數據類型	定義	說明
Axis	AXIS_REF_LITE	軸參數	指定欲操作軸。

5.9 MC_Gear_Weintek(FB)手搖輪

➤ 功能：

- 手搖輪功能將手搖輪輸入脈波乘上比例轉化為輸出脈波。
- “Execute”由 FALSE 轉變為 TRUE 觸發手搖輪功能。
- “ContinuousUpdate”為 TRUE 時啟動手搖輪功能，可在運動中改變脈波縮放比例。
- 使用手搖輪功能時脈波輸入方式 5501 的 bit-4 要設定為 1，輸入輸出非同軸。

➤ 圖示：



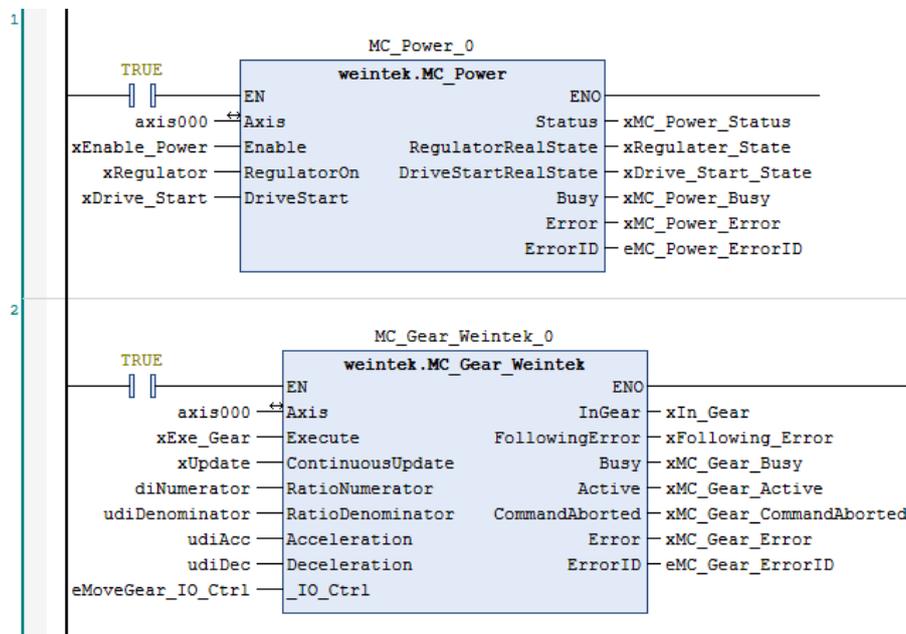
➤ 參數說明：

輸入參數	數據類型	定義	說明
Execute	BOOL	執行	當數值由 FALSE 轉變為 TRUE 時執行。
ContinuousUpdate	BOOL	運動中更新	TRUE：在運動中可以改變目標速度。
RatioNumerator	REAL	比例分子	輸出脈波 = 輸入脈波 $* \frac{\text{RatioNumerator}}{\text{RatioDenominator}}$
RatioDenominator	REAL	比例分母	
Acceleration	UDINT	加速度	指定加速度。指定加速度單位為使用者單位/s ²
Deceleration	UDINT	減速度	指定減速度。指定減速度單位為使用者單位/s ²
_IO_Ctrl	eMC_IO_Ctrl	IO 控制選項	None：不使用 I0~I2：使用數位輸入觸發運動
輸出參數	數據類型	定義	說明
InGear	BOOL	到位狀態	TRUE：輸出以跟上輸入脈波到達目標位置。
FollowingError	BOOL	跟隨誤差	TRUE：輸出與輸入差距大於設定時間。

Busy	BOOL	功能塊狀態	TRUE：功能塊已被執行。
Active	BOOL	運動狀態	TRUE：當前運動中功能塊。
CommandAborted	BOOL	命令中止	TRUE：功能塊被其他功能塊或事件而中止運動。
Error	BOOL	錯誤狀態	TRUE：當 Error 發生時。
ErrorID	BOOL	錯誤碼	功能塊錯誤碼，參考“附錄 A”
輸入輸出	數據類型	定義	說明
Axis	AXIS_REF_LITE	軸參數	指定欲操作軸。

➤ 程式編輯：

■ LD：



■ ST：

```
// MC_Power function block
MC_Power_0(
    Axis:= Axis000,
    Enable:= xEnable_Power,
    RegulatorOn:= xRegulator,
    DriveStart:= xDrive_Start,
    Status=> xMC_Power_Status,
    RegulatorRealState=> xRegulator_State,
    DriveStartRealState=> xDrive_Start_State,
    Busy=>xMC_Power_Busy ,
```

```
Error=> xMC_Power_Error,  
ErrorID=> eMC_Power_ErrorID);  
// MC_Gear function block  
MC_Gear_Weintek_0(  
Axis:= Axis000,  
Execute:= xExe_Gear,  
ContinuousUpdate:= xUpdate,  
RatioNumerator:= diNumerator,  
RatioDenominator:= udiDenominator,  
Acceleration:= udiAcc,  
Deceleration:= udiDec,  
_IO_Ctrl:= eMoveGear_IO_Ctrl,  
InGear=> xIn_Gear,  
FollowingError=> xFollowing_Error,  
Busy=> xMC_Gear_Busy,  
Active=> xMC_Gear_Active,  
CommandAborted=> xMC_Gear_CommandAborted,  
Error=> xMC_Gear_Error,  
ErrorID=> eMC_Gear_ErrorID);
```

附錄A. FB Error Code

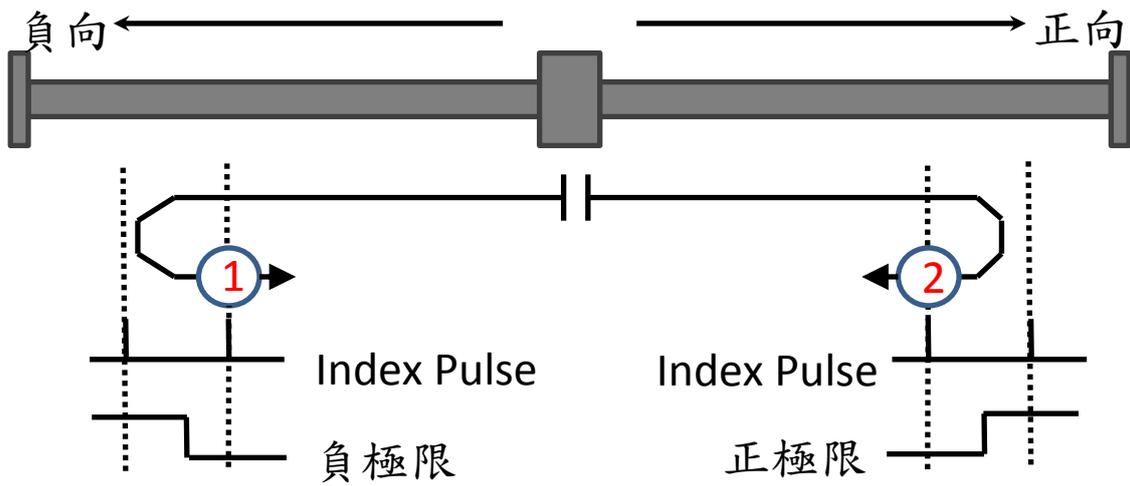
錯誤編號	FB 錯誤名稱	說明	故障排除
0	NO_ERROR	無錯誤發生	
1	AXIS_NOT_READY	運動時軸未就緒	排除其他錯誤後開啟 MC_Power 到 Status 為 True 的狀態重新啟動運動
2	AXIS_BUFFER_FULL	定位運動 Buffer 已滿	請修改程式避免將過多的定位運動排入 Buffer，並使用 MC_Reset 清除錯誤
3	AXIS_MOTION_ERROR	運動錯誤	請參考“iR-PU01-P 使用說明”的 4.3 Error 故障排除
4	AXIS_HOMING_ERROR	回原點錯誤	請檢查回原點的參數與環境設定
5	AXIS_TRANSITION_ERROR	錯誤的運動模式切換	請修改程式避免 Homing 與其他模式的運動之間的連接或是定位 Buffer 與非定位運動相接，並使用 MC_Reset 清除錯誤
6	FB_RUNTIME_ERROR	功能塊執行錯誤	您的功能塊與 CODESYS 裝置不匹配，請使用威綸 CODESYS 控制器。

※iR-PU01-P 模組發生錯誤時，錯誤碼在軸參數的 ErrorCode 內，詳細錯誤碼可參考“iR-PU01-P 使用說明”的第 4 章故障排除。

附錄B. 歸原點方式

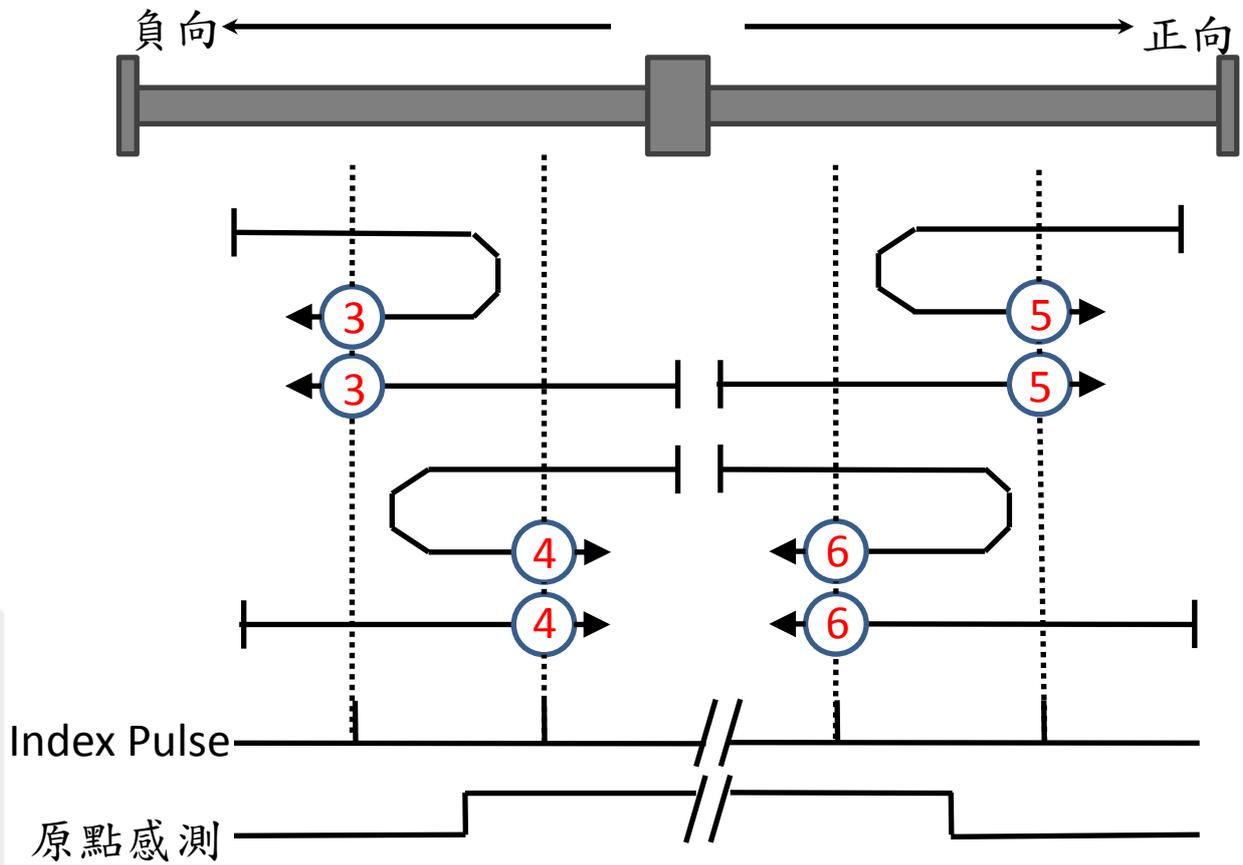
方法 1 & 2：使用極限及 Index 脈波歸原點

啟動歸原點運動後尋找極限訊號，再反向尋找第一個 Index 脈波。

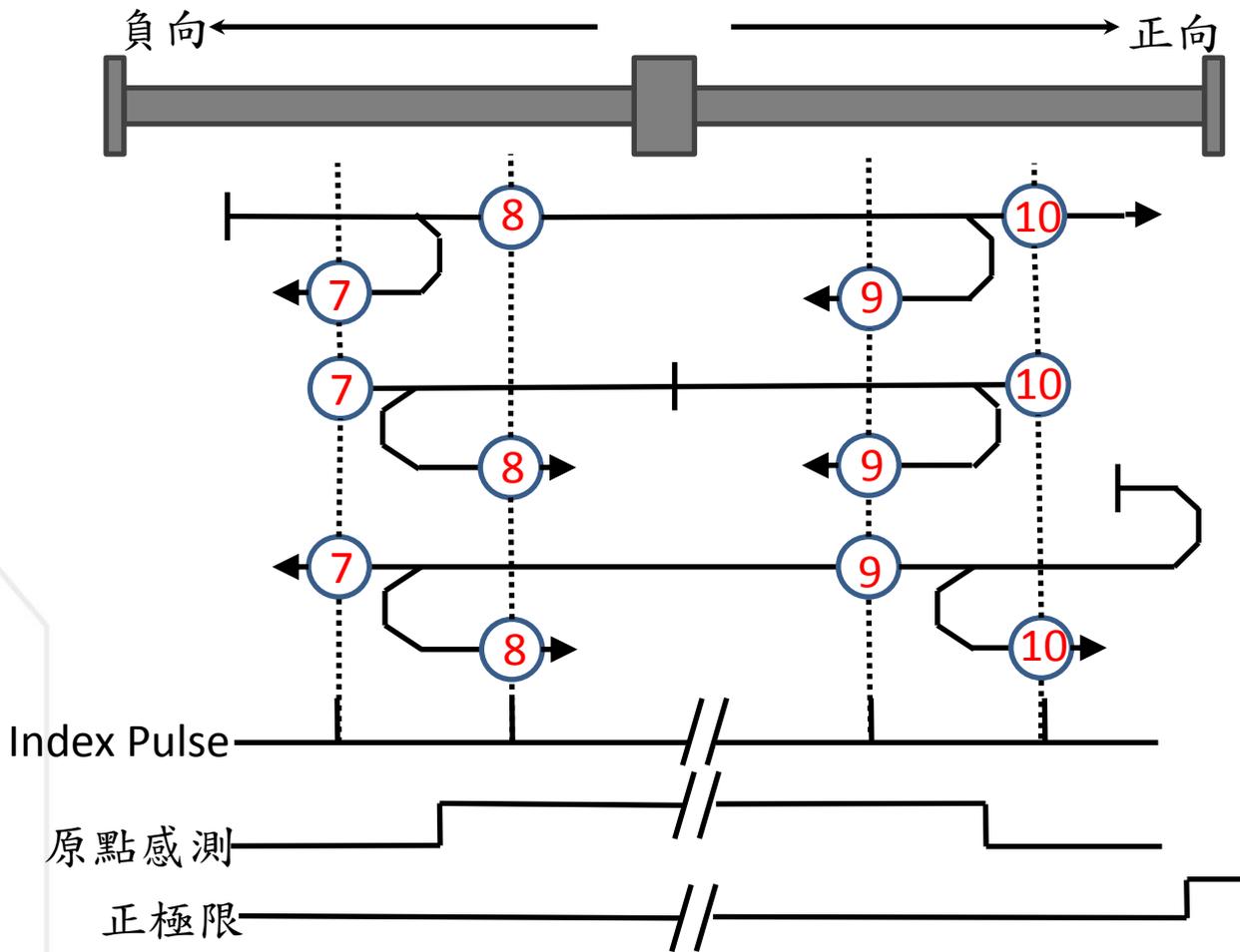


方法 3 ~ 6：使用原點感測及 Index 脈波歸原點

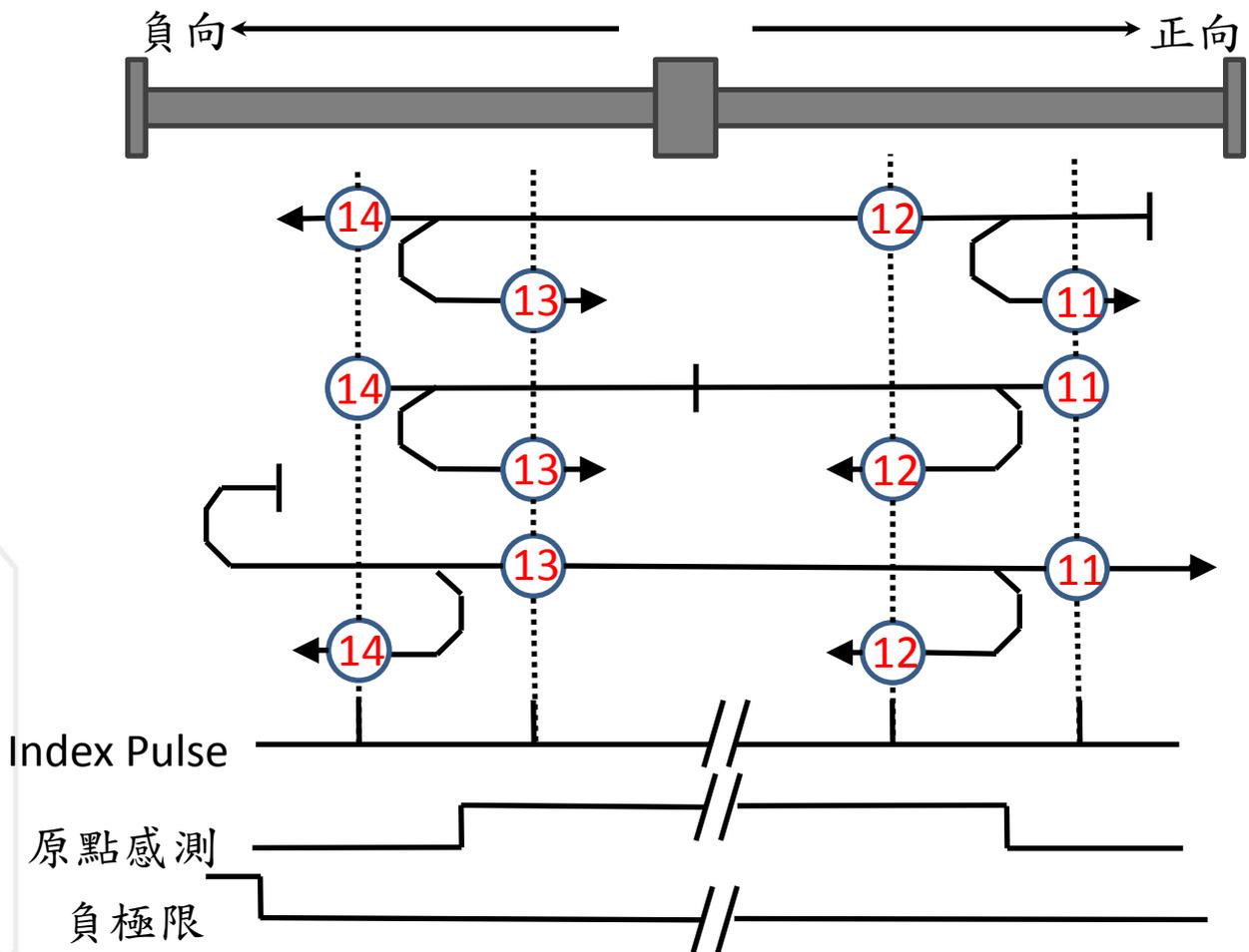
啟動歸原點運動依照原點感測位置決定運動方向尋找原點感測訊號，碰觸原點感測後再往正向/負向尋找第一個 Index 脈波。



方法 7 ~ 10：使用原點感測、正極限及 Index 脈波歸原點 - 初始移動往正向
 啟動歸原點運動往正方向尋找原點感測訊號，若碰觸正極限則反向尋找原點感測，
 找到原點感測後再往正向/負向尋找第一個 Index 脈波。

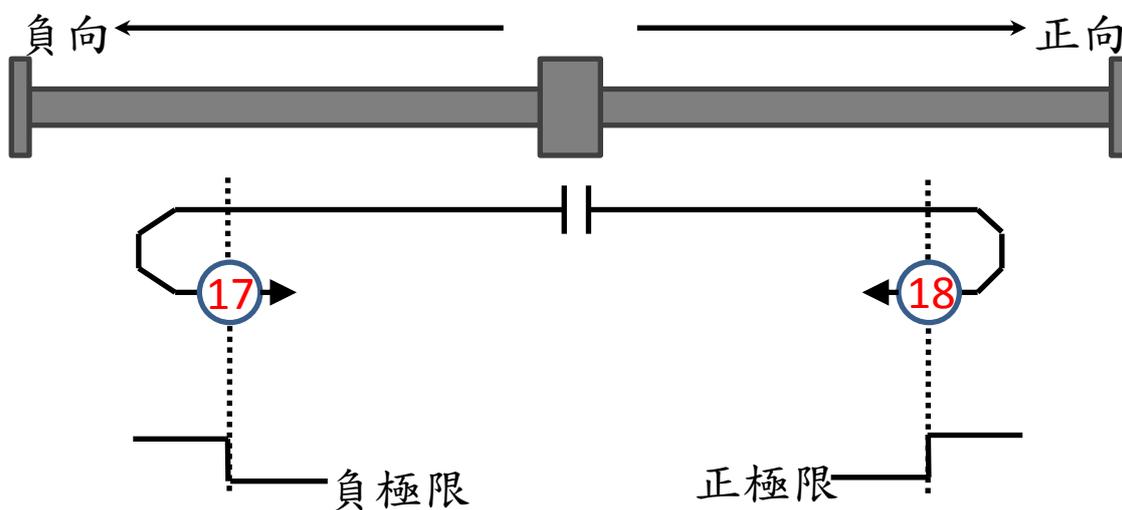


方法 11 ~ 14：使用原點感測、負極限及 Index 脈波歸原點 – 初始移動往負向
 啟動歸原點運動往負方向尋找原點感測訊號，若碰觸負極限則反向尋找原點感測，
 找到原點感測後再往正向/負向尋找第一個 Index 脈波。



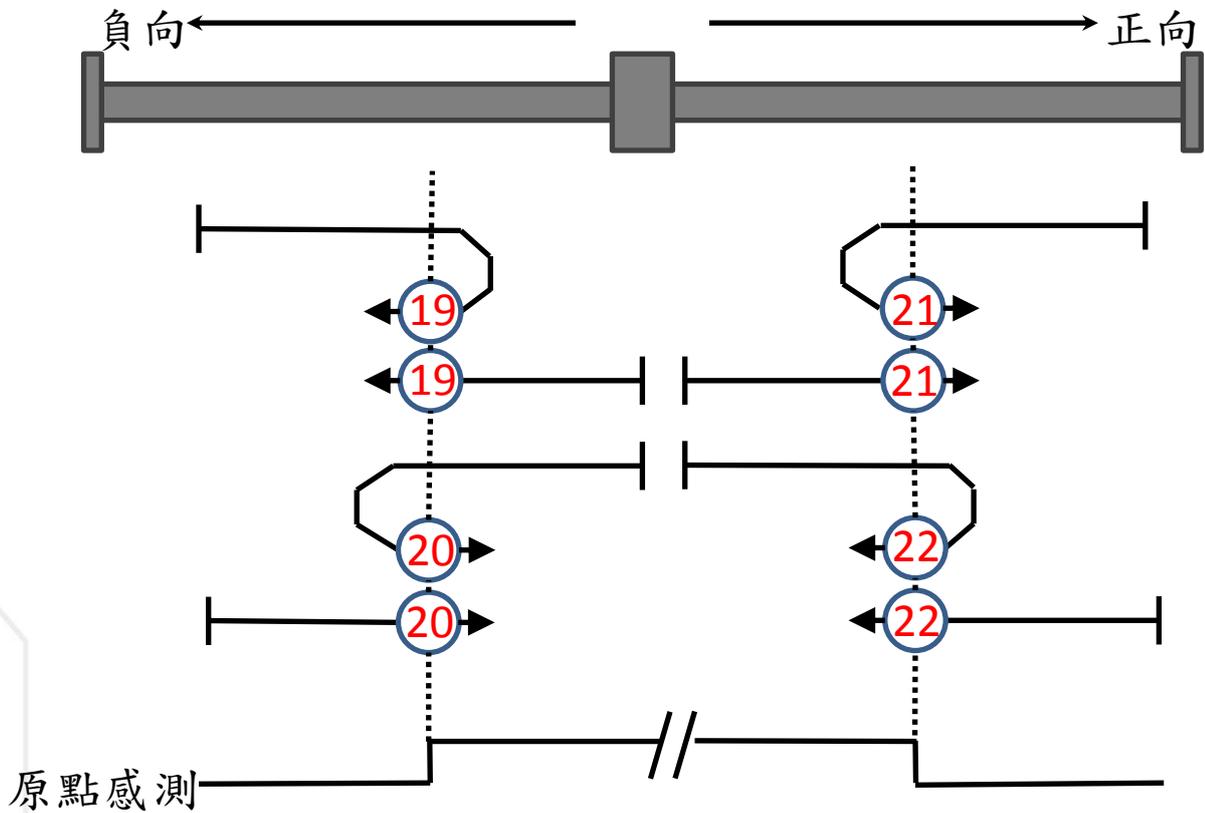
方法 17 & 18：使用極限歸原點

方法 17 & 18 與方法 1 & 2 類似但不找尋 Index 脈波，原點完成在極限位置。



方法 19 ~ 22：使用原點感測歸原點

方法 19 ~ 22 與方法 3 ~ 6 類似但不找尋 Index 脈波，所以原點完成在原點感測位置的兩側。



方法 23 ~ 26：使用原點感測、正極限歸原點

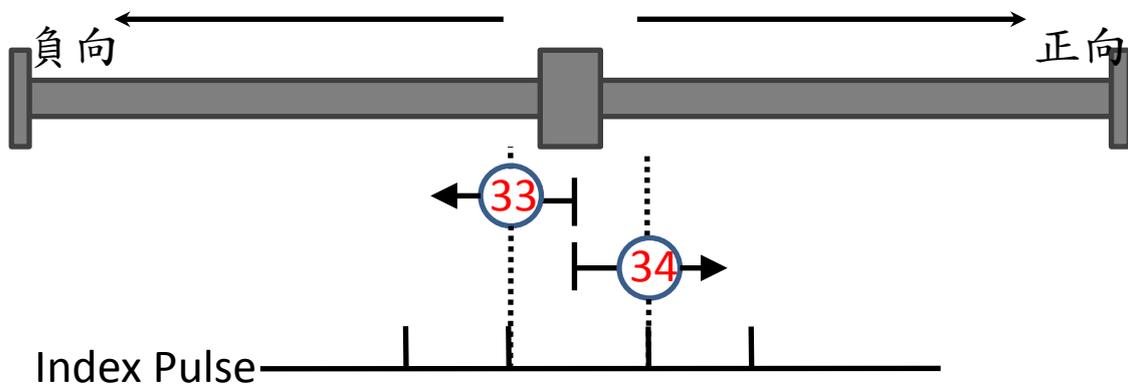
方法 23 ~ 26 與方法 7 ~ 10 類似但不找尋 Index 脈波，所以原點完成在原點感測位置的兩側。

方法 27 ~ 30：使用原點感測、負極限歸原點

方法 27 ~ 30 與方法 11 ~ 14 類似但不找尋 Index 脈波，所以原點完成在原點感測位置的兩側。

方法 33 ~ 34：使用 Index 歸原點

方法 33 ~ 34 只找尋第一個 Index 脈波。



方法 37(預設)：

當前位置為原點完成位置。

Position actual value = Home offset