

ECM-XFU ITE

使用手冊

Ver.04
2023/03/10

使用前準備

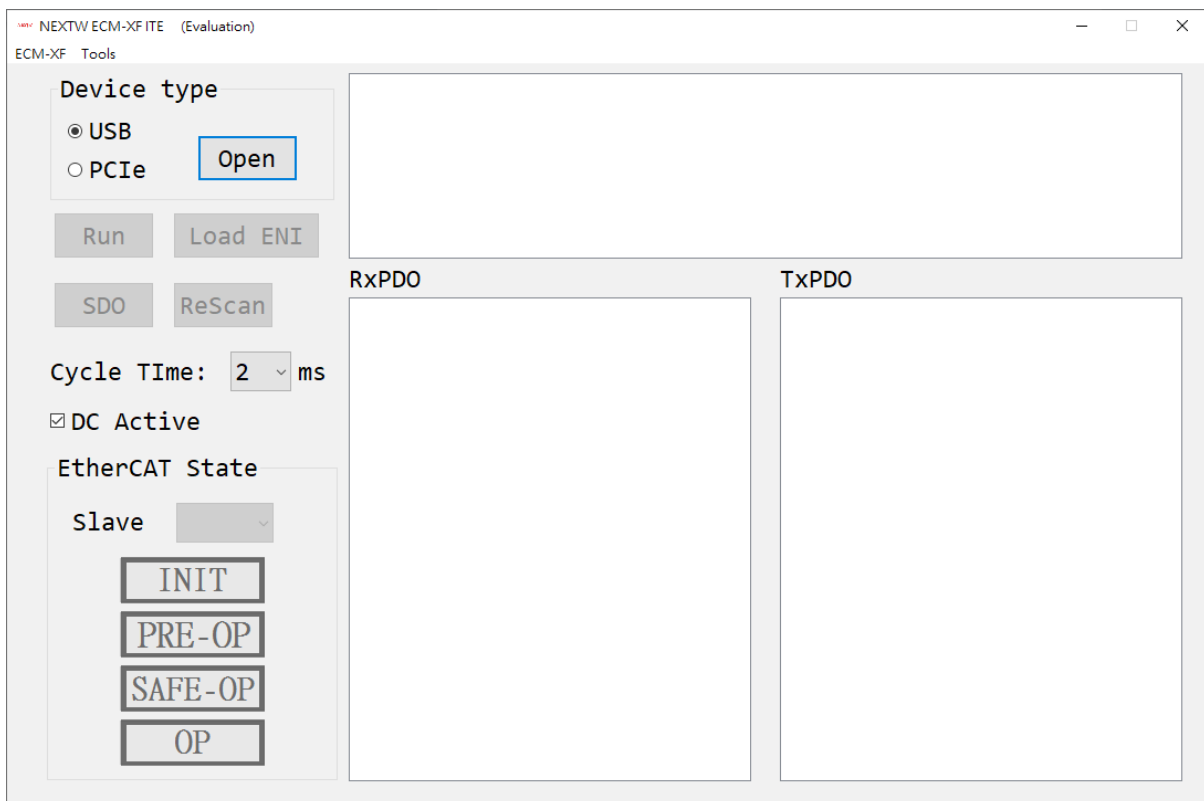
1. ECM-XFU-SK 板若為 1.1 或以前則無法支援 USB 連接
2. **ECM-XF 僅可使用 Update Firmware 功能。**
3. 將您的主板連接到從站。

版本(評估版/正式版)

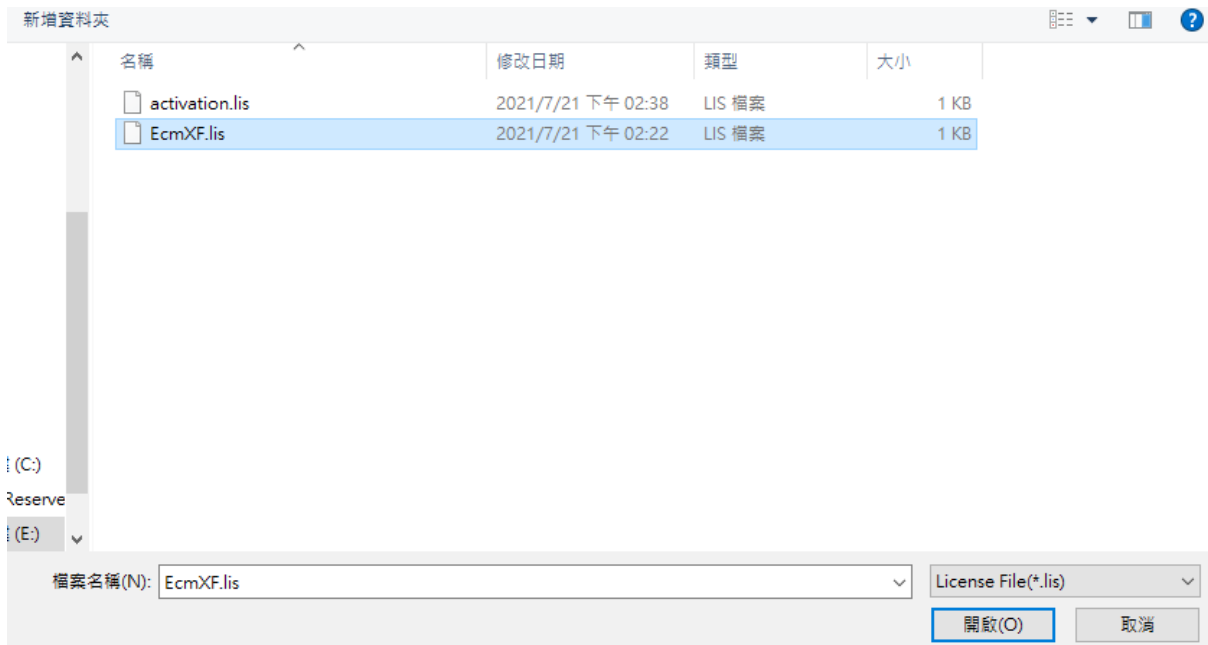
左上方視窗標題欄中顯示 Evaluation 為評估版本，評估版本可切換狀態、讀取和寫入從站 PDO、與從站進行 SDO 操作、韌體升級等功能。

正式版本可操作完整的功能，包含 SII Editor、Cyclic test、Code Generator 等。

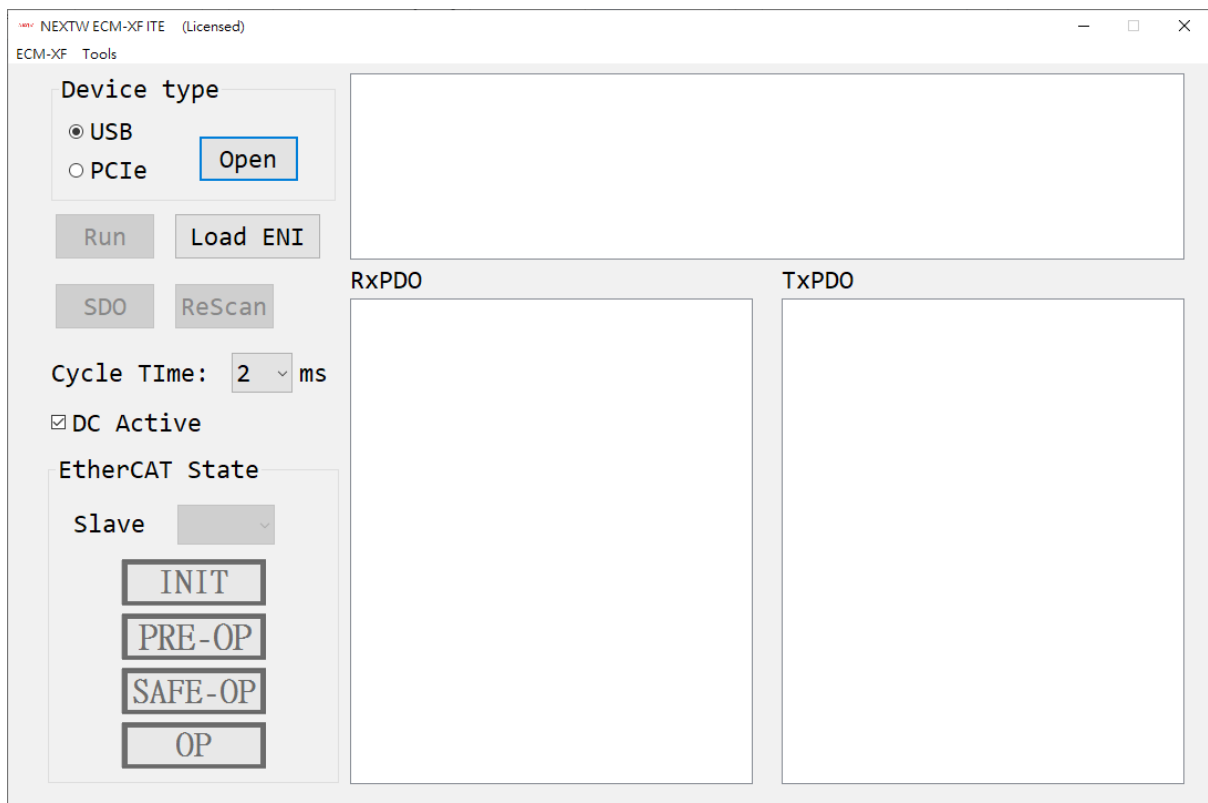
正式版本於完成付費後點擊[ECM-XF]選單內[License]，會有詢問是否已取得 License 視窗，點選[取消]，會產生“activation.lis”檔案，將此檔案及硬體序號（位於 ECM-XFU-SK 或 ECM-XF-PCle 板子背面）註明公司/使用者後寄至 service@nextw.com.tw，將可取得專屬的授權檔案。



獲得授權後，將授權文件 “EcmXF.lis” 放入 ITE 執行檔 的同一文件夾中。然後單擊工具欄上的[ECM-XF]中[License]，於詢問視窗中選擇[確定]，選擇授權文件並單擊打開。

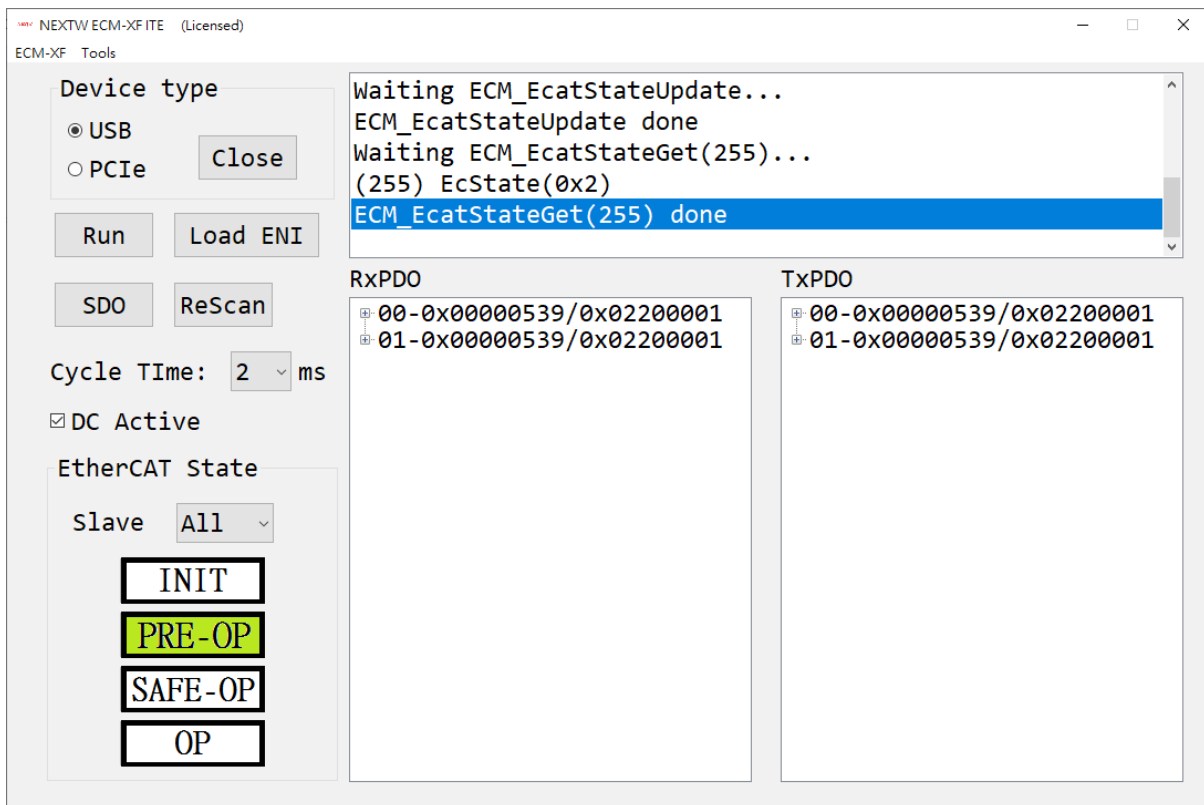


左上方的視窗將顯示 “Licensed” 表示已升級為正式版。



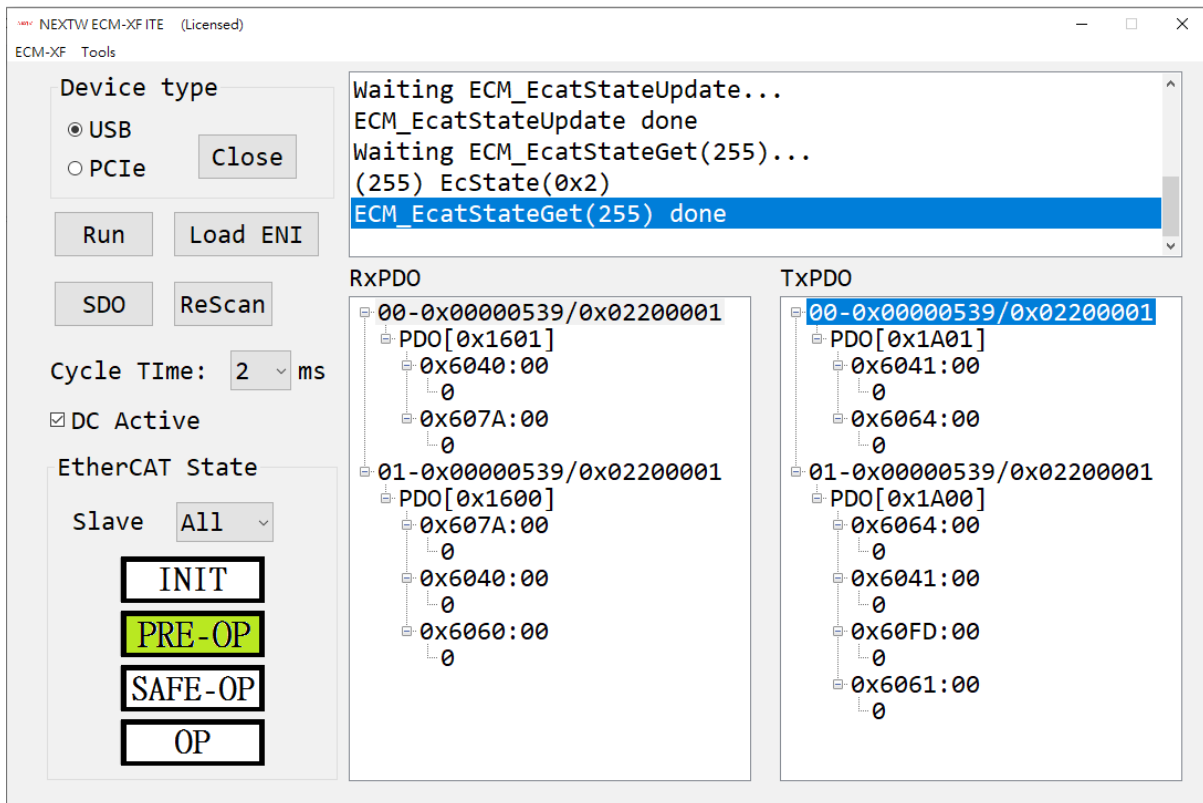
功能

此 ITE 支援 使用 USB 或 PCIe 連接至電腦的 ECM-XFU-SK/ECM-XF-PCIe 裝置。
選擇 USB 或 PCIe 介面，選擇適當的週期時間 (Cycle Time) 及 DC Active，
之後按下 “Open” 按鈕。



打開連接後，系統會偵測硬體，並掃描連接於主站的從站裝置，並使從站進入 Pre-OP 狀態。右上角的文字框將顯示 訊息 (如 芯片韌體版本、執行過程訊息等)。從站的預設的 PDO 設置將顯示在 RxPDO 和 TxPDO 文字框中。
前面的 00、01 代表從站順序編號，後面則代表該從站的製造商編號 (Vendor ID) 及產品編號(Product ID)，透過點擊 RxPDO 和 TxPDO 文字框中內容前面的 “+” 符號，會顯示該從站的詳細的 PDO 內容。
若無列出相關 PDO 內容，則可能需要手動配置從站的 0x1C12/0x1C13 內容。可以透過 Load ENI 來自動寫入相關配置。注意，必須是 CoE 的從站裝置才能使用 SDO 進行讀/寫或 Load ENI。

下圖為連結 2 台的 YASKAWA 驅動器的預設配置。

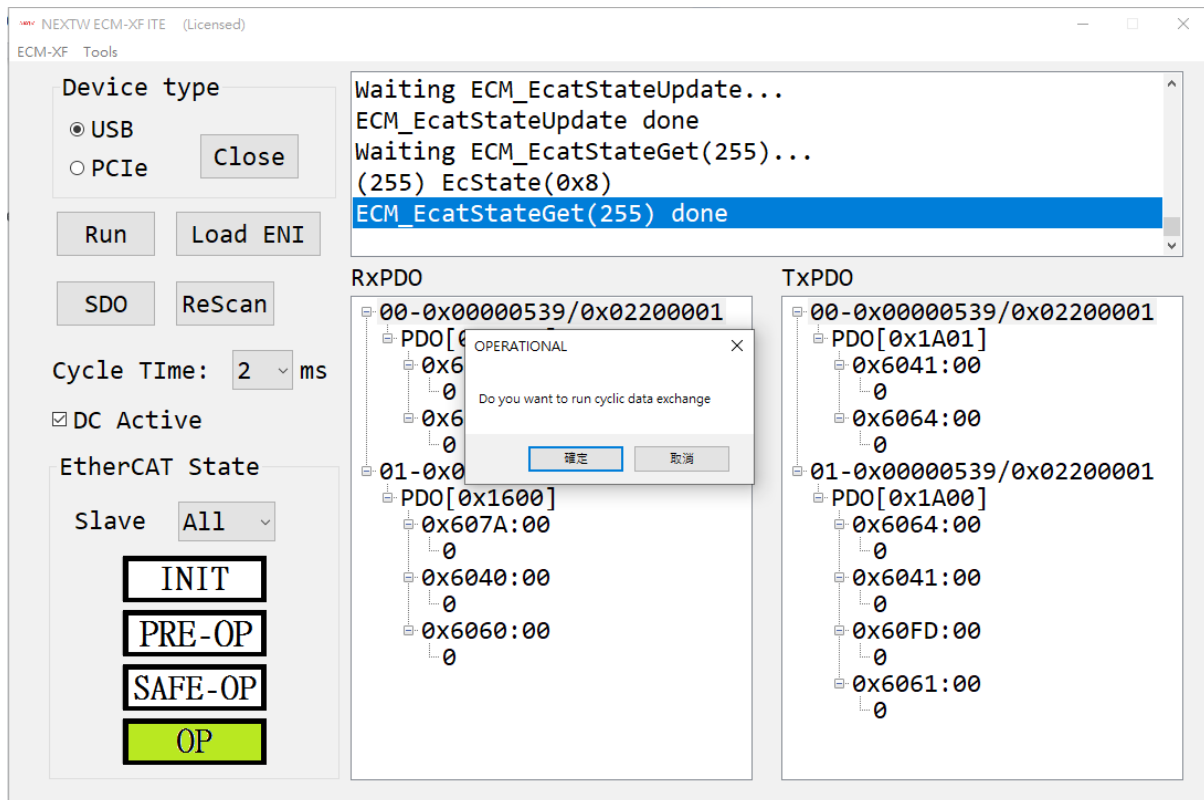


在 Pre-OP 狀態下，可以進行 SDO 讀/寫、使用 ENI 中的配置，或是使用 Tools 中的 SII editor 和 code generator。

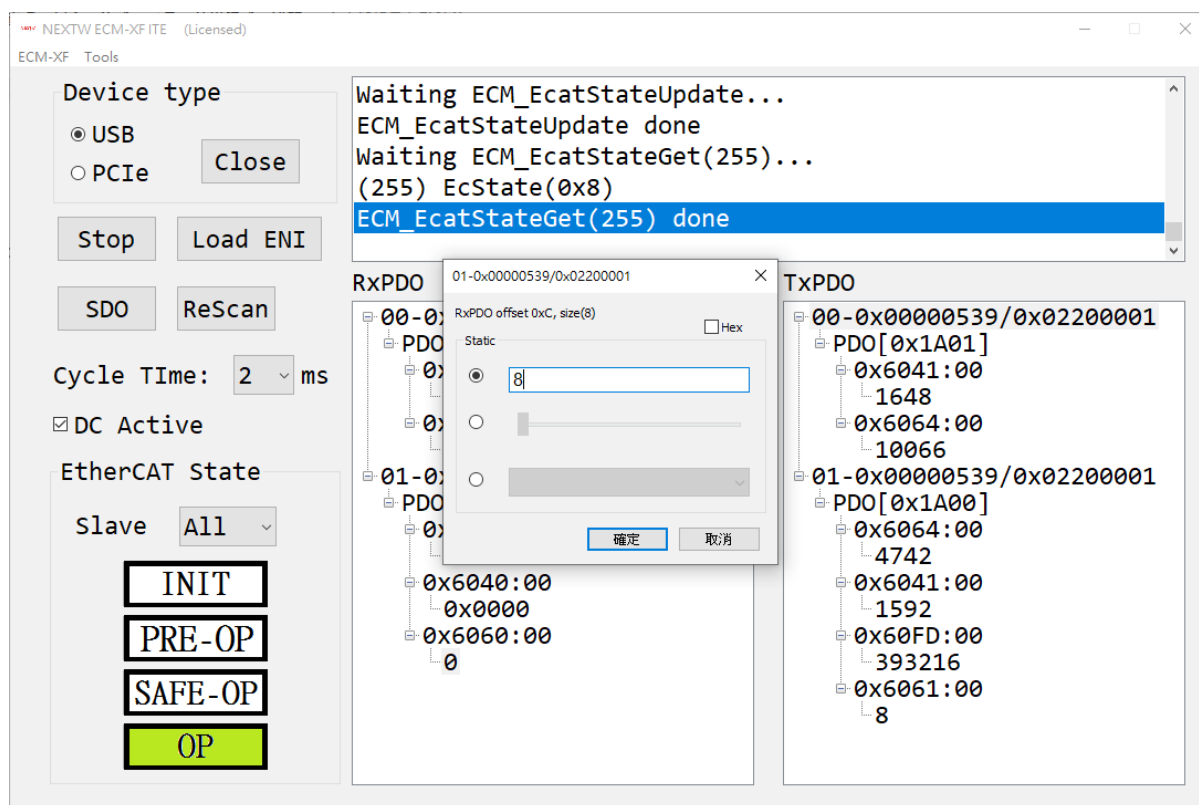
左下角可改變從站的狀態，欲切至 OP 狀態，必須依照 Init->PREOP->SafeOP->OP 依序切換，於 OP 狀態可返回之前任一狀態。需要注意狀態變更的順序（從上到下），否則切換狀態會失敗。若切換狀態失敗，可參考訊息區的 AL Status Code，以瞭解失敗原因。

常見的切換狀態失敗原因為 PDO 配置錯誤、從站不支援 Cycle Time 設置或不支援 DC Active。

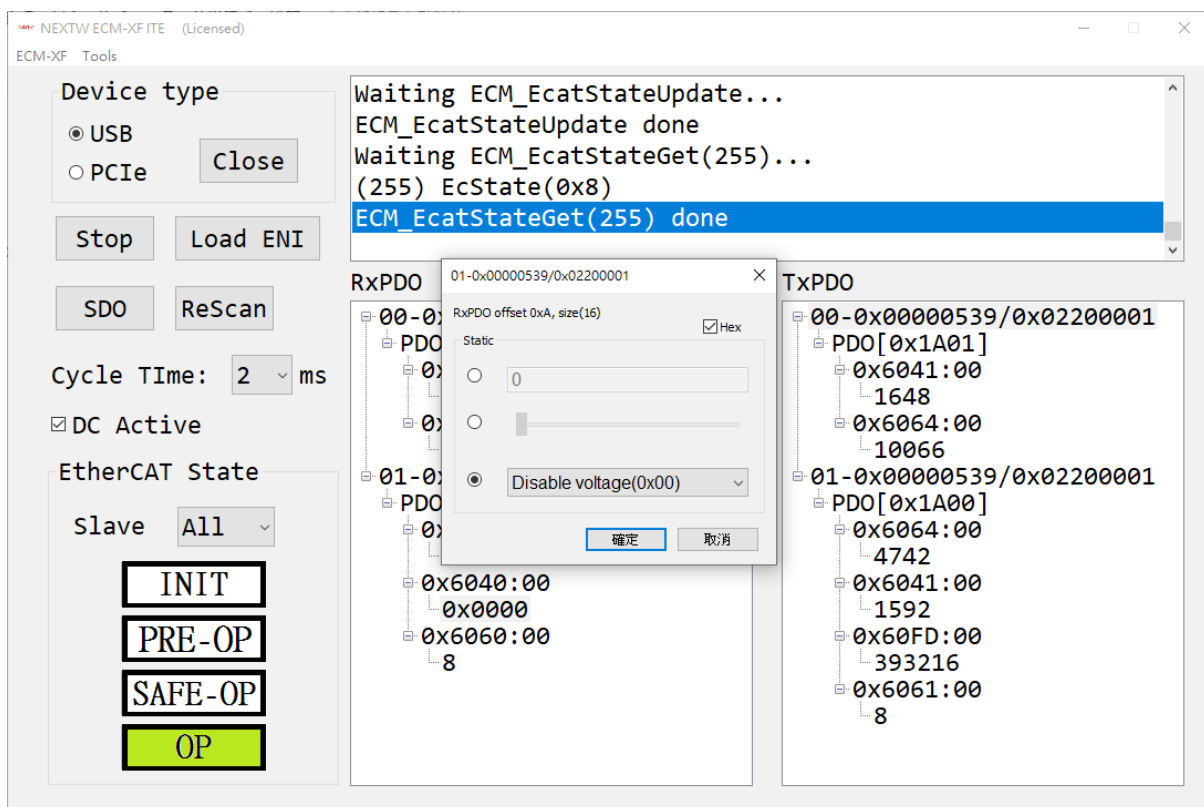
當進入 OP 狀態時，將會彈出視窗顯示是否要進行 cyclic data exchange 的詢問視窗，選擇[確定]，即開始與從站進行實際的週期性資料交換，若選擇[取消]，使用介面上的數值並不會與從站進行資料交換，按下左邊的[Run]按鈕則會開始進行週期性資料交換，再按下[Stop]則會停止將介面上的數值與從站進行週期性資料交換。



RxPDO 為主站送至從站的命令，在 OP 模式下，可變更 RxPDO 內的數值。
TxPDO 為從站送至主站的狀態，下圖顯示 2 個從站的當前狀態。如果需要，
可以右鍵單擊 Hex 旁的方框以切換為十六進制。而在 RxPDO 雙擊值可以變更
數值。

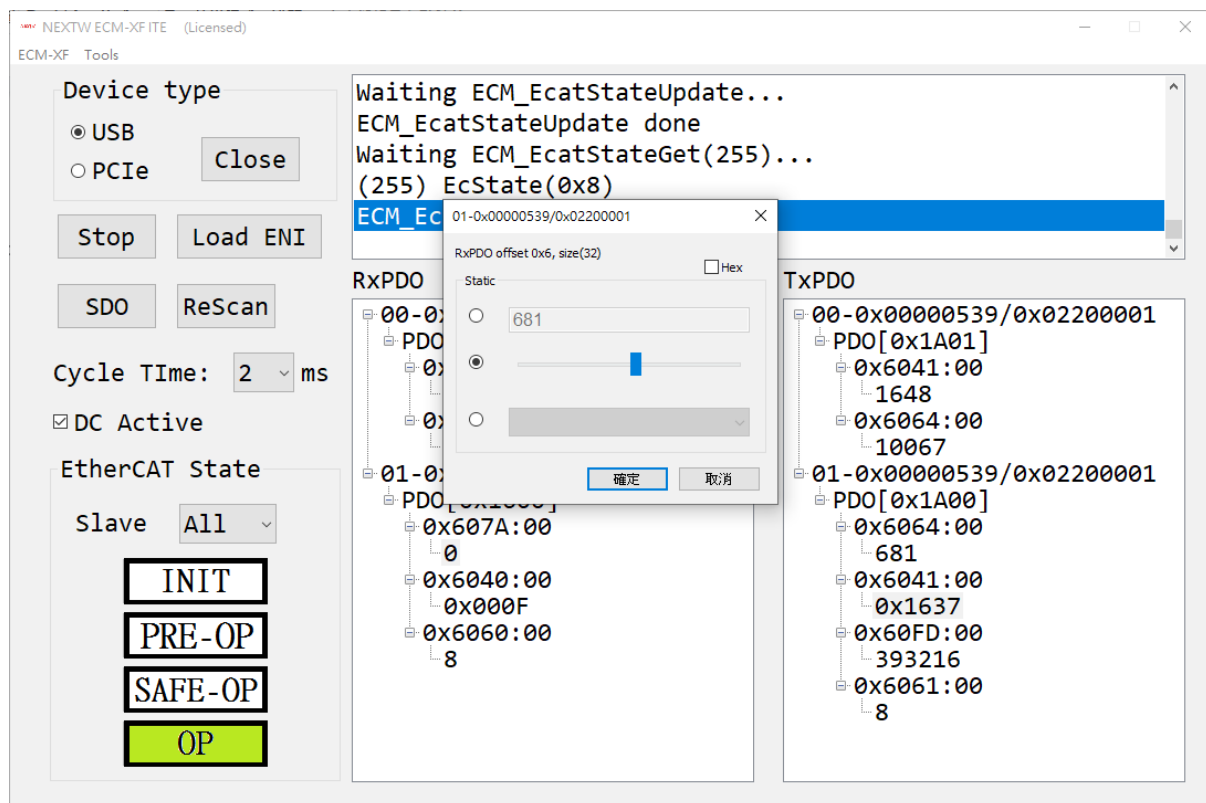


雙擊 RxPDO 中 0x6040 (Control Word) 可以改變運行模式下的 402 狀態機。
但不需要每次都按 [確定]來改變狀態。可以觀察到 TxPDO 中 0x6041 (Status Word) 發生變化。
針對伺服馬達若要激磁 (Servo on)，可透過改變 0x6040 的數值，依序選擇
“Shutdown(0x6)”、“Switch On(0x7)”、“Enable Operation(0xF)”，
切至 Enable Operation 後馬達即會激磁，詳細的說明可參考從站關於 0x6040
的說明。



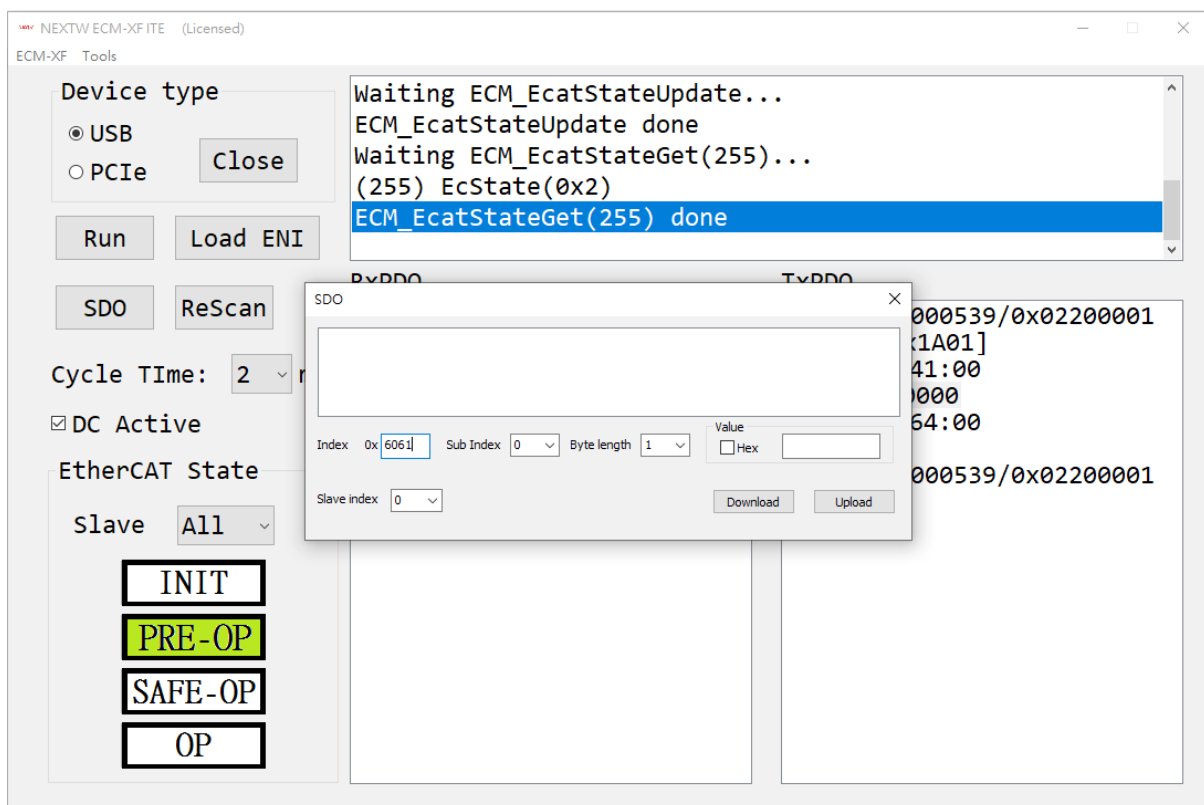
激磁後RxPDO中的0x6064 (Position actual value) 會與TxPDO中的0x607A (Target position) 值相同。接著可以雙擊 0x607A (Target position) 設置下一個週期時間的新位置。亦可使用拉bar來改變數值。

當您處於運行模式為CSP (即0x6060 : 0 Operation mode = 8) 時，驅動器會於每個週期嘗試轉動馬達至目標位置 (0x607A)，而回傳目前的實際位置 (0x6064)。更多的運行模式請參閱從站的使用手冊。

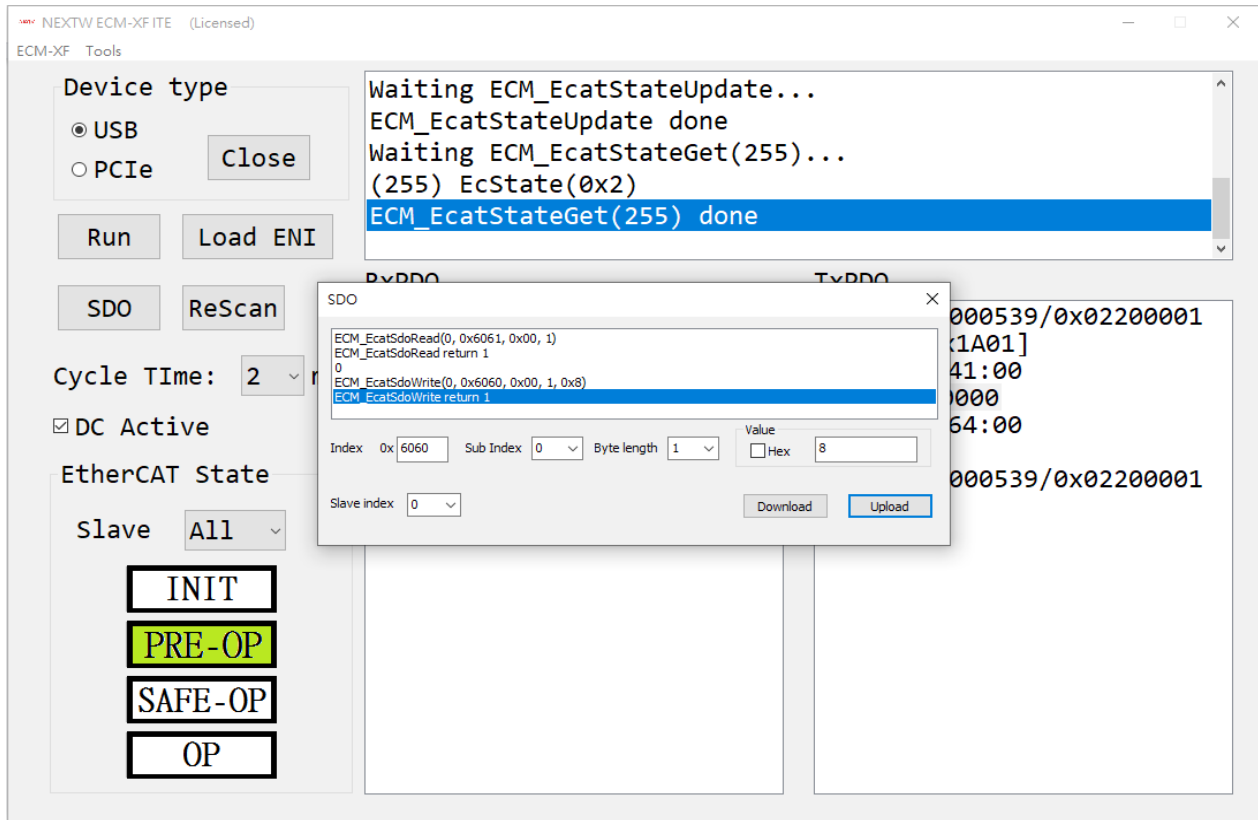


SDO

SDO 是非週期性資料交換，CoE 裝置在 PRE-OP、SAFE-OP 及 OP 階段均可進行，按下[SDO]按鈕後會出現 SDO 視窗。例如，欲讀取運行模式，可讀取 Object Index 為 0x6061，Sub Index 為 0, Byte length(Size) 為 1，然後按下 [Read]即可於右側 Value 地方讀到數值。更多關於 Object 的定義請參閱從站的使用手冊。



若欲透過 SDO 機制寫入 Object，則除了 Index、Sub Index、Byte length 均需填寫外，Value 亦需填寫。例如欲寫入運行模式，可寫入 Object Index 為 0x6060，Sub Index 為 0, Byte length(Size) 為 1，Value 則為欲寫入的值，然後按下[Write]即會寫入。

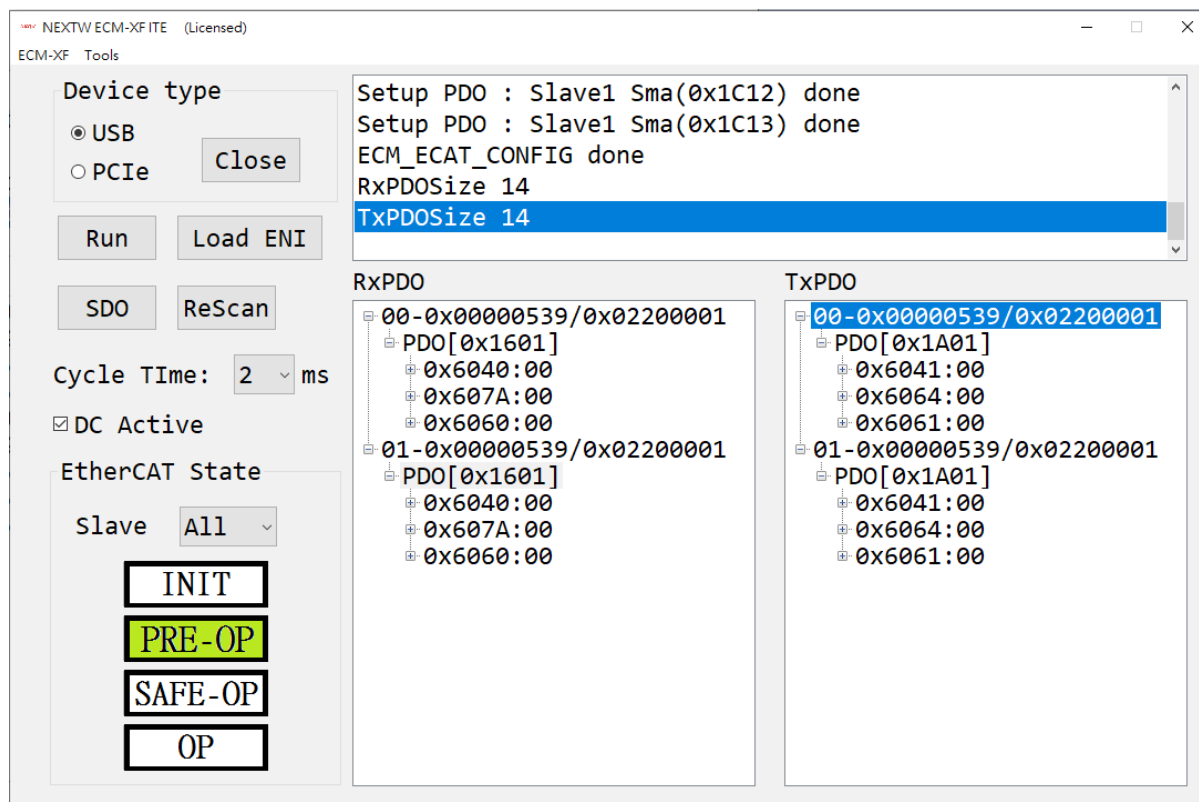
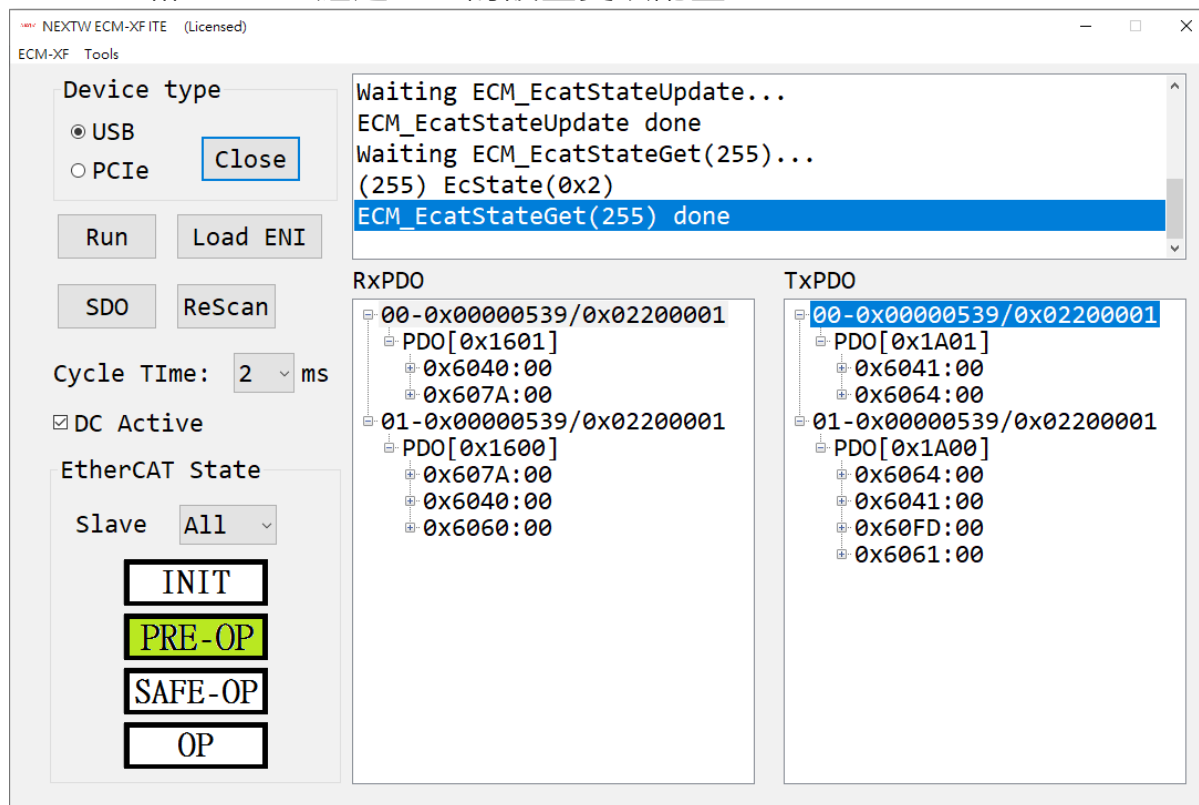


若欲確認是否正確寫入，則可使用 SDO Read 讀取該數值。

Object 的存取權由從站控制，是否能讀取/寫入從站的 Object 請參閱從站手冊。

Load ENI

ITE 支援讀取 ENI 內的 PDO 配置。按下[Open]後再按下[Load ENI]按鈕，選擇 ENI 文件，ITE 會根據所選的 ENI 文件嘗試配置從站 PDO 結構。下圖顯示 RxPDO 和 TxPDO 經過 ENI 的設置更改配置。



若無 ENI 檔案亦可在 PRE-OP 階段自行透過 SDO 寫入 0x1C12、0x1C13 的方式來配置 PDO，例如欲設定 RxPDO 的 Sync Manager PDO assign objects (0x1C12)設定其 PDO mapping objects 為 0x1601，詳細的作法及步驟如下：

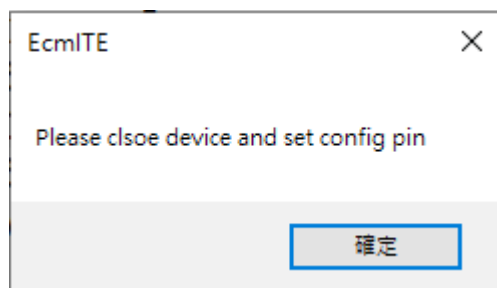
1. SDO Write 0x1C12:0 (size 1) Value = 0
2. SDO Read 0x1C12:0 (size 1) 確認為 Value = 0
3. SDO Read 0x1C12:1 (size 2) 確認預設值 (例如 0x1600)
4. SDO Write 0x1C12:1 (size 2) Value 為欲設定的 PDO 值 (例如 0x1601)
5. SDO Read 0x1C12:1 (size 2) 確認設定是否正確 (如 0x1601)
6. SDO Write 0x1C12:0 (size 1) Value = 1 (PDO mapping objects 的數量)
7. SDO Read 0x1C12:0 (size 1) 確認 Value 為 1
8. 按下[ReScan]

若欲進一步改變 PDO mapping objects 中的內容 mapping entry，則可在上述 2 和 3 中間加做以下步驟: (如欲將 0x1601 設有 2 個 mapping, 分別為 0x6040:0 size 2 bytes 及 0x607A:0 size 4 bytes)

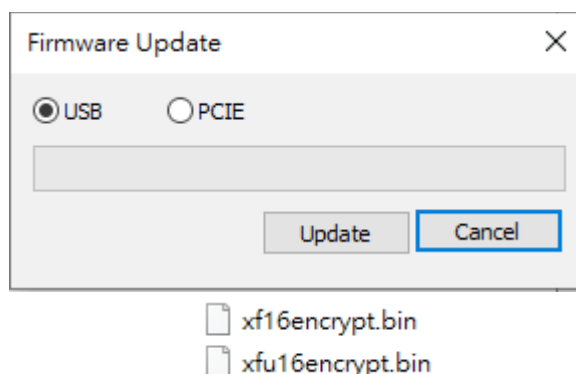
- 2.1 SDO Write 0x1601:0 (size 1) Value = 0
- 2.2 SDO Read 0x1601:0 (size 1) 確認為 Value = 0
- 2.3 SDO Write 0x1601:1 (size 4) Value = 0x60400010，其中 bit0-7 為 mapped object 的長度(單位 bit)，bit8-15 為 mapped object 的 sub Index，bit16-31 為 mapped object 的 Index。
- 2.4 SDO Read 0x1601:1 (size 4) 確認 Value = 0x60400010。
- 2.5 SDO Write 0x1601:2 (size 4) Value = 0x607A0020，其中 bit0-7 為 mapped object 的長度(單位 bit)，bit8-15 為 mapped object 的 sub Index，bit16-31 為 mapped object 的 Index。
- 2.6 SDO Read 0x1601:2 (size 4) 確認 Value = 0x607A0020。
- 2.7 SDO Write 0x1601:0 (size 1) Value = 2(mapping entry 的數量)
- 2.8 SDO Read 0x1601:0 (size 1) 確認為 Value = 2

更新韌體(Update firmware)

ITE 支援了 ECM-XF 的韌體更新。請先將硬體模式切換到 update mode 並單擊工具欄 “ECM-XF”，然後選擇 “FW update”。update mode 詳情請參見硬體板的手冊，以 ECM-XF-SK V1.2 為例，須使用跳線帽將 J3 中間針腳與 L 針腳短路即為 updata mode。如果您的開發板未處於 update mode，會出現如下圖所示的視窗。

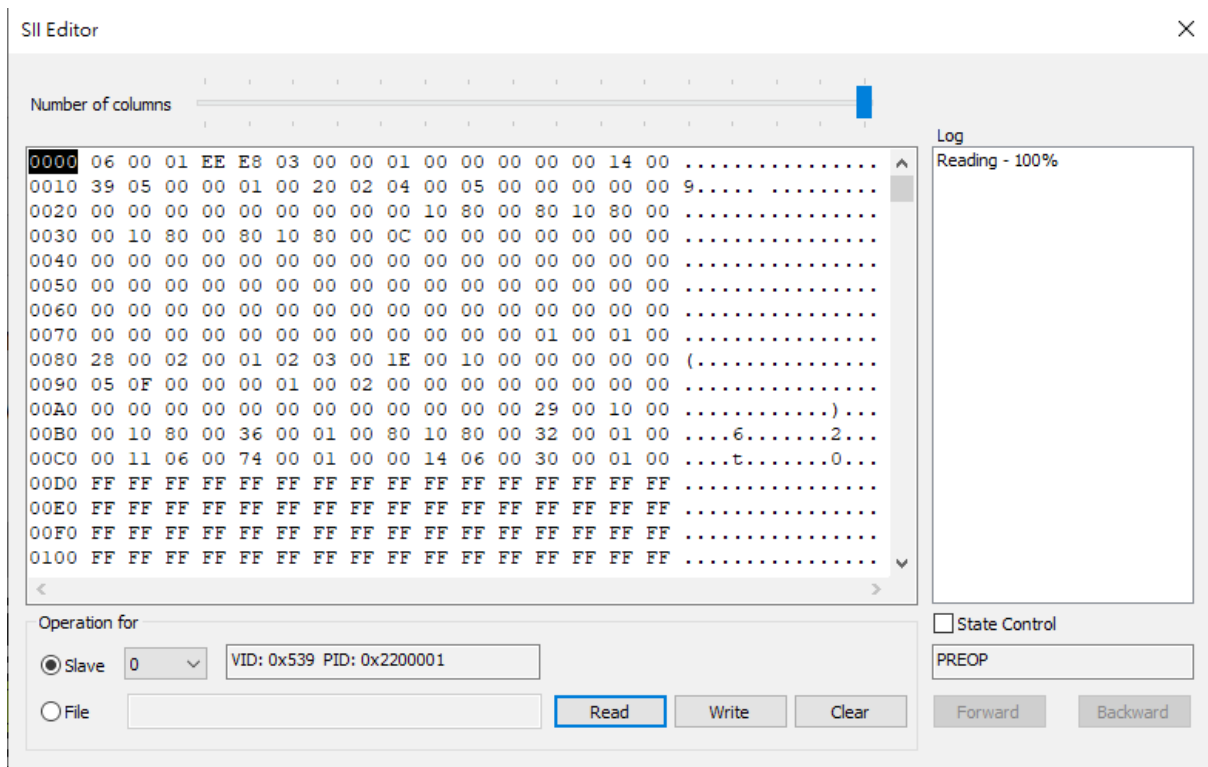
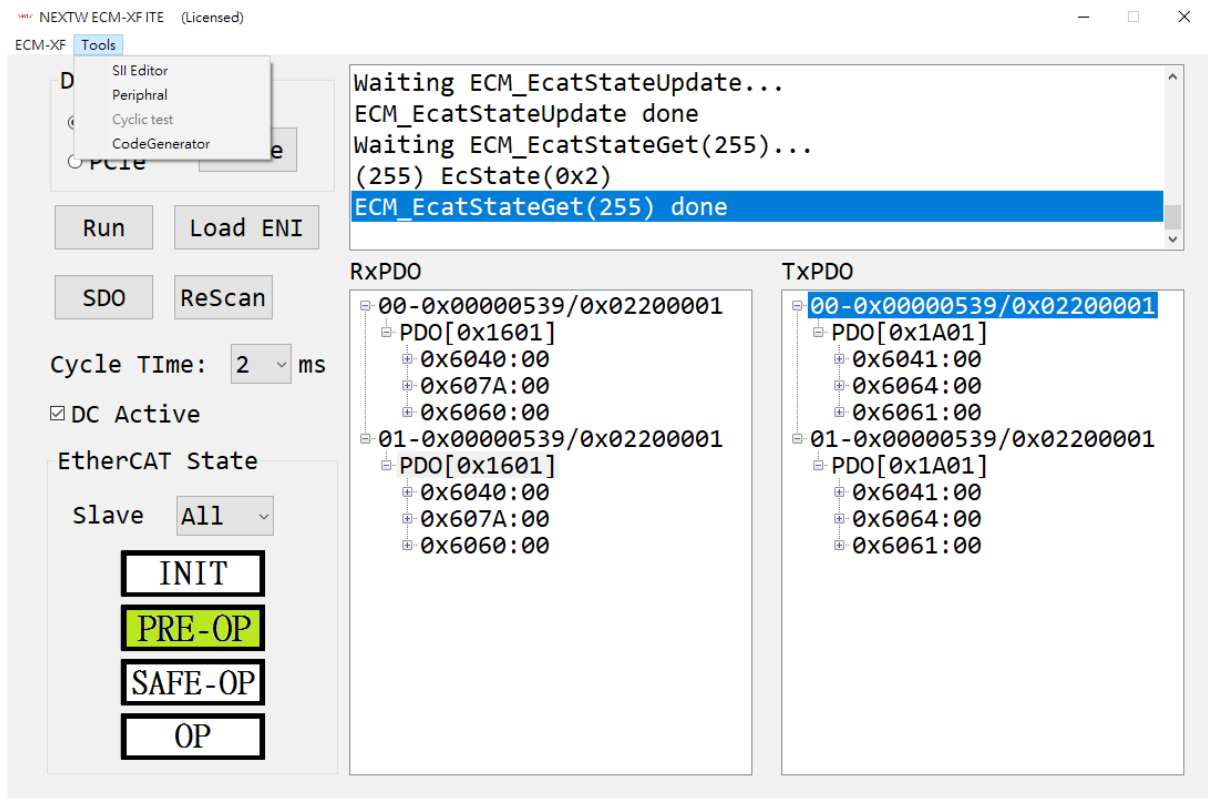


如果您處於 update mode，則在按下[FW update]後可選擇連結介面(USB 或 PCIe)，之後再選擇更新檔案即可。請注意，請務必確認所選的更新檔案是正確的，錯誤的更新檔可能導致無法正常運作，且無法復原。



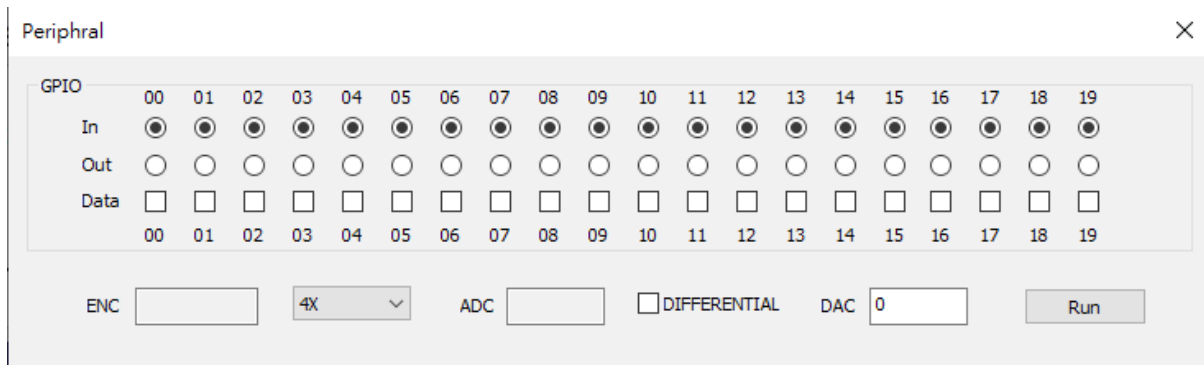
SII editor

Tools 中的[SII Editor]，可讀取從站 SII 文件和並存成檔案，或讀取檔案並寫入從站。請注意必須 Pre-OP 狀態下使用此 editor。



XFU 週邊IO (Peripheral)

Peripheral 的視窗可控制或顯示 ECM-XFU / ECM-XF-PCIe GPIO/QEI/ADC 的功能。



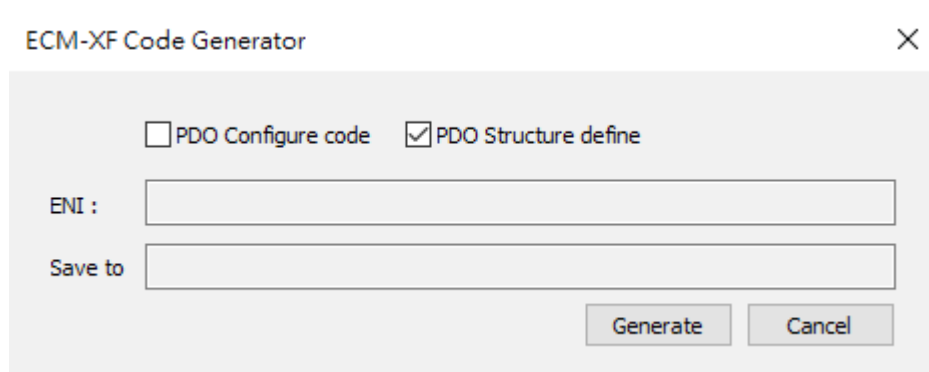
The Peripheral configuration window displays GPIO settings for pins 00 to 19. It includes sections for In (input), Out (output), and Data (direction) configuration. Below the pin settings, there are fields for ENC, a resolution dropdown (set to 4X), ADC, a DIFFERENTIAL checkbox, DAC (set to 0), and a Run button.

GPIO	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
In	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>
Out	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Data	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

ENC 4X ADC ☐ DIFFERENTIAL DAC

代碼產生工具(Code Generator)

代碼產生工具可讀取 ENI 文件(EtherCAT Network Information File)，並生成配置 c 文件(PDO Configure code)，此文件可作為撰寫程式時的配置參考。代碼產生工具亦可將當前的 PDO 結構定義(PDO Structure define)另存為 .h 文件，此 PDO 結構定義以當下實際的 PDO 定義為主，您可先使用[Load ENI]自動配置 ENI 內的定義，或是使用 SDO 讀寫 0x1C12、0x1C13 後再生成 .h 文件，此文件可作為撰寫程式時的結構定義參考。



The ECM-XF Code Generator window allows users to generate code from an ENI file. It features checkboxes for 'PDO Configure code' and 'PDO Structure define' (which is checked). Below these are input fields for 'ENI :' and 'Save to :'. At the bottom right are 'Generate' and 'Cancel' buttons.

ECM-XF Code Generator

☐ PDO Configure code ☒ PDO Structure define

ENI :

Save to

配置參考程式 (.c 文件) 和 PDO 結構定義的參考文件 (.h 文件) 可以放入您自行撰寫的應用程式中。

以下為簡單的的導入說明將 c 文件函式 ECM_ConfigPDO_CodeGen(void) 複製到 main.c 中，如下圖所示。

```
#define ECM_TBL_CNT_CC 4

int ECM_ConfigPDO_CodeCen(void)
{
    int nDone = 0;
    int nRet = 0;
    PDO_CONFIG_HEAD stEcmPdoTbl[ECM_TBL_CNT_CC] = {
        {0x539,0x2200001,0,1,0x1C12,{0x1601,0x0,0x0,0x0},{3,0,0,0},{16,0,0x6040},{32,0,0x607A},{8,0,0x6060}}},
        {0x539,0x2200001,0,1,0x1C13,{0x1A01,0x0,0x0,0x0},{3,0,0,0},{16,0,0x6041},{32,0,0x6064},{8,0,0x6061}}},
        {0x539,0x2200001,1,1,0x1C12,{0x1601,0x0,0x0,0x0},{3,0,0,0},{16,0,0x6040},{32,0,0x607A},{8,0,0x6060}}},
        {0x539,0x2200001,1,1,0x1C13,{0x1A01,0x0,0x0,0x0},{3,0,0,0},{16,0,0x6041},{32,0,0x6064},{8,0,0x6061}} }
    };
    for (int i = 0; i < ECM_TBL_CNT_CC; i++) {
        if (stEcmPdoTbl[i].PDOCnt) {
            nRet = ECM_EcatSdoSetPdoConfig(&stEcmPdoTbl[i]);
            if (nRet == 1) {
                nDone++;
            }
        }
    }
    return nDone;
}

int main()
{
    uint16_t u16SpiDataSize = 992; //fixed in USB
    uint8_t u8Version;

    PDO_CONFIG_HEAD RxPDOConfig[2];
    PDO_CONFIG_HEAD TxPDOConfig[2];
    uint8_t RxData[TEST_SPI_DATA_SIZE] = { 0 };
    uint8_t TxData[TEST_SPI_DATA_SIZE] = { 0 };
}
```

在配置部分只需使用函數 "ECM_ConfigPDO_CodeGen()"

```
nret = ECM_StateCheck(0xFF, EC_STATE_PRE_OP, 10000); // Set mode must be at PRE-OP state

if (nret == 0) {
    return -1;
}

std::cout << "All Slaves are in PRE-OP state\n";
std::cout << "Assign and configure PDO\n";

ECM_ConfigPDO_CodeCen();

// Call ECM_EcatReconfig() after setting PDO configures
nret = ECM_EcatReconfig();
if (nret < 0) {
    std::cout << "ECM_ECACONFIG error : " << nret << "\n";
    return -1;
}
```

EcmPdoDefine.h 是與 SPI 版本中的 PdoDefine.h 相同的功能文件。在 main.c 中使用 #include EcmPdoDefine.h 即可。

```
#include <EcmUsrDriver.h>
#include <platform.h>
#include <EcmPdoDefine.h>
```

