

QEC ETG.7200 測試工具

使用手冊

版本 1.6 | 2026 年 4 月

昭營科技股份有限公司

修訂記錄

版本	日期	變更說明
1.6	2026 年 4 月	新增第 12 章實際操作教學五個情境 (12.1 識別 / 12.2 PDO / 12.3 DC / 12.4 PP 模式 / 12.5 CSP 模式)，附實機操作照片；圖片樣式與列表編號全面校正。
1.2	2026 年 4 月	初次對外發行。九個分頁完整記錄，附 UI 截圖說明。
1.0	2025 年	內部工程版本。

目錄

修訂記錄	2
目錄	3
1. 概覽	5
1.1 全域工具列	5
1.2 啟動流程	5
2. Information 分頁 (裝置資訊)	6
2.1 裝置識別資訊	6
2.2 Mailbox Protocol (信箱協定支援)	6
2.3 CoE Details (CoE 功能旗標)	7
2.4 Reload 按鈕	7
3. State Machine 分頁 (狀態機)	8
3.1 EtherCAT 狀態機	8
3.2 FoE 韌體下載 / 上傳	9
4. Process Data 分頁 (程序資料)	10
4.1 PDO Assignment 0x1C12 (接收 PDO 指定)	10
4.2 PDO Assignment 0x1C13 (傳送 PDO 指定)	11
4.3 PDO Configuration (PDO 映射設定)	11
5. CoE 分頁	12
5.1 SDO 上傳 / 下載	12
5.2 物件字典瀏覽器 (Object Dictionary)	13
5.3 緊急訊息紀錄 (Emergency Message)	13
6. SII EEPROM 分頁	14
6.1 直接寫入面板	14
6.2 寫入站台別名面板 (Write Alias Address)	15
6.3 EEPROM 讀取 / 顯示面板	15
7. ESC Registers 分頁 (ESC 暫存器)	16
7.1 寫入面板	16
7.2 讀取 / 顯示面板	17
8. DC 分頁 (分散式時鐘)	18
9. Online 分頁 (線上 I/O)	19
9.1 數位 / 類比輸出 (左側面板)	19
9.2 數位 / 類比輸入 (右側面板)	20
10. CiA 402 分頁 (伺服驅動器)	21
10.1 Drive ID 與 Attach (驅動器連接)	21
10.2 CiA 402 Mode (操作模式)	21
10.3 CiA 402 State (驅動器狀態)	22
10.4 Servo Control — 運動參數	22
10.5 Servo Control — 循環同步命令 (CSP / CSV / CST)	22
10.6 Profile Position 面板 (輪廓位置模式)	23
10.7 Profile Velocity 面板 (輪廓速度模式)	23
10.8 Homing 面板 (原點復歸模式)	23
10.9 Servo Status (伺服狀態)	24
11. 故障排除	25
12. 實際操作教學	26
12.1 情境 A — 首次連接與識別從站	26
12.2 情境 B — 設定 PDO 並驗證 I/O	28
12.3 情境 C — 執行 DC 同步設定	31
12.4 情境 D — PP 模式伺服試運轉	34
12.5 情境 E — CSP 模式循環同步運轉	38

12.6 情境 F — 物件字典瀏覽與 SDO 讀寫	42
13. EtherCAT API 參考	49
13.1 EtherCAT API	49
13.2 SDO / PDO / ESC / SII / FoE API.....	50
13.3 CiA 402 API	52
14. 錯誤代碼對照表.....	55
14.1 EtherCAT API 錯誤碼.....	55

1. 概覽

ETG.7200 測試工具是一套部署於 QEC 平台 (86Duino + 86HMI) 上的 EtherCAT 裝置測試應用程式。本工具提供圖形化的分頁操作介面，可依循 ETG.7200 測試規範對 EtherCAT 從站 (SubDevice) 進行調試、診斷與功能驗證。

介面上方共有九個分頁標籤：Information (裝置資訊)、State Machine (狀態機)、Process Data (程序資料)、CoE、SII EEPROM、ESC Registers (ESC 暫存器)、DC (分散式時鐘)、Online (線上 I/O) 及 CiA 402 (伺服驅動器)。

所有分頁共用一組全域工具列，確保關鍵參數在各頁面間保持同步。

1.1 全域工具列

下列控制項在每個分頁均可見。在任一頁面修改數值後，會立即反映至所有其他頁面。

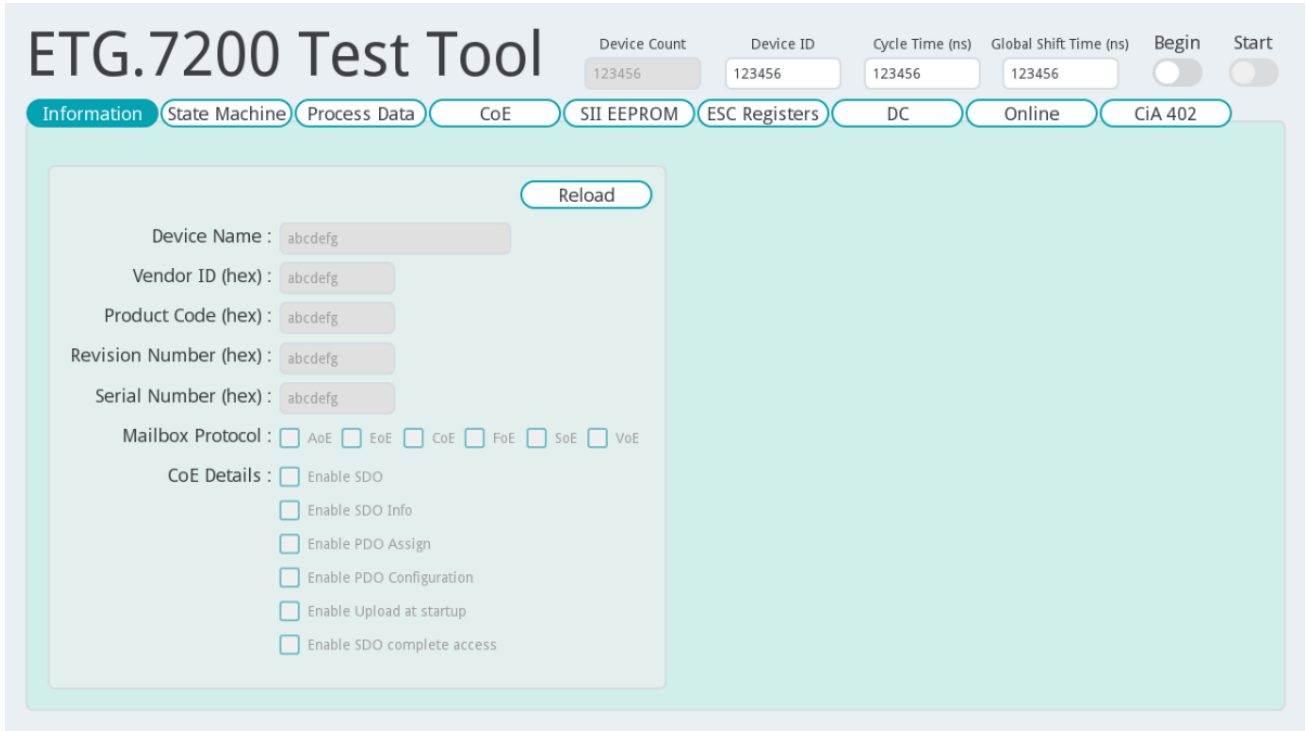
控制項	說明
Device Count (從站數量)	唯讀。Begin 啟動後所偵測到的從站數量。
Device ID (從站編號)	選擇目前作業的目標從站 (從 0 開始計數)。範圍：0 至 Device Count - 1。
Cycle Time (ns) (週期時間)	EtherCAT 主站的通訊週期，單位為奈秒。預設值：1,000,000 (即 1 ms)。Start 啟動後無法修改。
Global Shift Time (ns) (全域位移時間)	DC 位移時間偏移量，單位為奈秒。設定為 0 時，主站將在 Start 後自動計算合適的位移值。
Begin (切換開關)	撥為 ON：呼叫 <code>master.begin()</code> ，初始化 EtherCAT 網路。 撥為 OFF：呼叫 <code>master.end()</code> ，釋放網路。Start 啟動時此開關鎖定。
Start (切換開關)	撥為 ON：呼叫 <code>master.start()</code> ，開始週期性 PDO 通訊。 須先啟用 Begin。Bootstrap 模式啟用時此開關鎖定。

1.2 啟動流程

1. 開啟 QEC 硬體電源，等待 HMI 完成開機。
2. 依需求設定 Cycle Time 與 Global Shift Time。
3. 將 Begin 撥為 ON，系統將自動更新 Device Count。
4. 設定 Device ID 以選擇目標從站，起始值為 0。
5. 將 Start 撥為 ON，開始週期性通訊。
6. 切換至所需分頁，執行對應的測試操作。

注意 如果需要執行 PDO Mapping 或 DC 設定，請必須在 Begin 啟用、Start 關閉時進行。

2. Information 分頁 (裝置資訊)



[圖示：畫面截圖 — 圖 1 — Information 分頁]

Information 分頁讀取並顯示所選從站的識別資訊與功能描述。所有顯示欄位均為唯讀，點擊 Reload 按鈕後才會更新內容。

2.1 裝置識別資訊

欄位	說明
Device Name (裝置名稱)	從從站物件字典中讀取的 ASCII 名稱字串。
Vendor ID (廠商編號，十六進位)	32 位元廠商識別碼 (例如：00000002)。
Product Code (產品代碼，十六進位)	32 位元產品代碼。
Revision Number (修訂版本，十六進位)	32 位元硬體／韌體修訂版本號。
Serial Number (序號，十六進位)	32 位元序列號。

2.2 Mailbox Protocol (信箱協定支援)

六個唯讀核取方塊顯示從站在 SII EEPROM 中宣告支援的信箱協定。

核取方塊	協定說明
AoE	ADS over EtherCAT
EoE	Ethernet over EtherCAT
CoE	CANopen over EtherCAT

FoE	File Access over EtherCAT
SoE	Servo Drive Profile over EtherCAT
VoE	Vendor-specific over EtherCAT (廠商自定義)

2.3 CoE Details (CoE 功能旗標)

六個唯讀核取方塊反映 CoE Details 位元組，顯示從站支援的 CoE 服務。

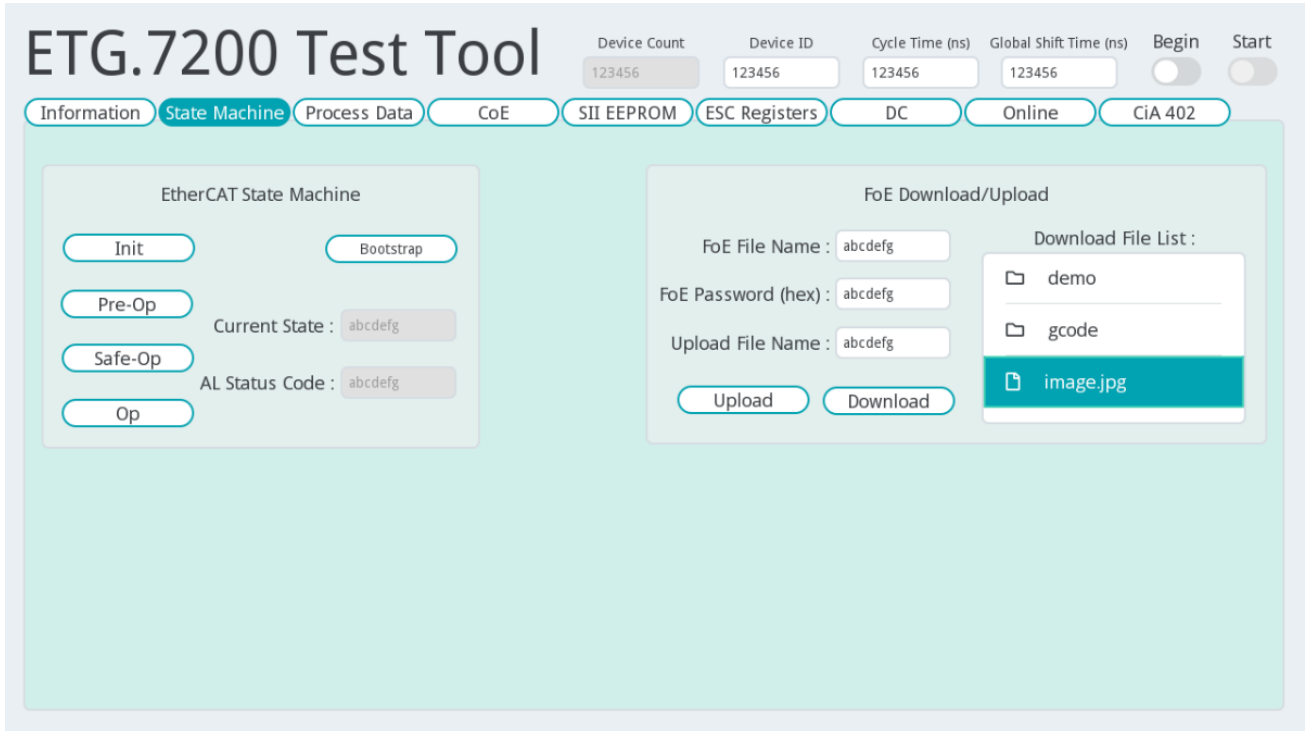
核取方塊	意義
Enable SDO	支援 SDO 通訊。
Enable SDO Info	支援 SDO Information 服務 (OD 瀏覽所需)。
Enable PDO Assign	支援透過 SDO 進行 PDO Assignment。
Enable PDO Configuration	支援透過 SDO 進行 PDO Mapping 設定。
Enable Upload at startup	裝置在啟動時自動上傳 PDO 配置。
Enable SDO complete access	支援 SDO Complete Access 傳輸。

2.4 Reload 按鈕

點擊 **Reload** 可向目前選定的從站查詢，並更新上方所有欄位的顯示內容。

操作前須先啟用 **Begin**。

3. State Machine 分頁 (狀態機)



[圖示：畫面截圖 — 圖 2 — State Machine 分頁]

State Machine 分頁包含兩個獨立功能區塊：左側為 EtherCAT 狀態機 (ESM) 控制，右側為 FoE 韌體傳輸。

3.1 EtherCAT 狀態機

此區塊可將選定從站切換至任意有效的 EtherCAT 狀態，並即時顯示目前狀態。

控制項 / 欄位	說明
Init 按鈕	命令從站切換至 Init 狀態。
Pre-Op 按鈕	命令從站切換至 Pre-Operational 狀態。
Safe-Op 按鈕	命令從站切換至 Safe-Operational 狀態。須先啟用 Start。
Op 按鈕	命令從站切換至 Operational 狀態。須先啟用 Start。
Bootstrap 按鈕	命令從站進入 Bootstrap 狀態。Bootstrap 啟用期間，Start 開關將被鎖定，防止意外啟動 PDO 通訊。
Current State (目前狀態，唯讀)	顯示從站目前的 AL 狀態：Init、Pre-Op、Safe-Op、Op 或 Bootstrap。
AL Status Code (AL 狀態碼，唯讀)	4 位十六進位的 AL 狀態碼。非零值表示從站發生錯誤。

注意 Safe-Op 與 Op 按鈕僅在 Begin 與 Start 同時啟用時才可操作。Bootstrap 模式下 Start 開關會被禁用，以防止韌體更新期間誤發 PDO 通訊。

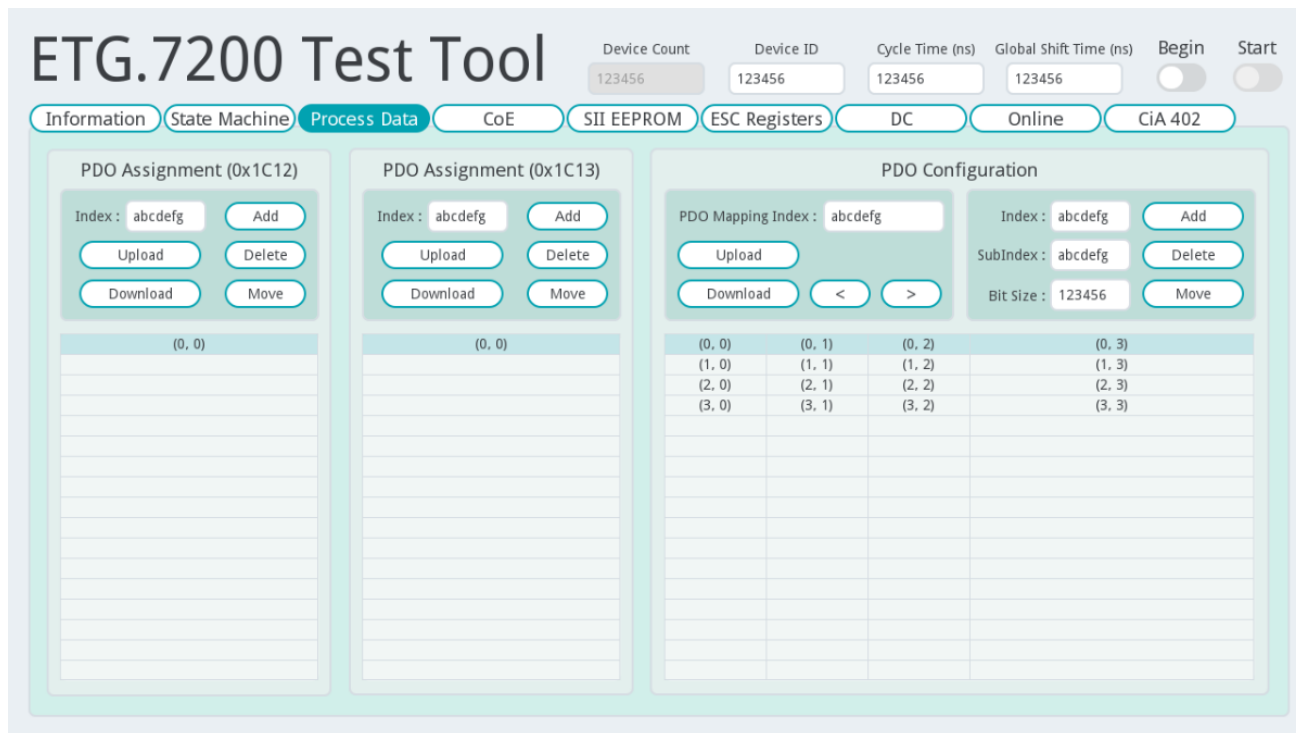
3.2 FoE 韌體下載 / 上傳

此區塊透過 File Access over EtherCAT (FoE) 協定，在 QEC USB 儲存裝置與從站之間傳輸韌體檔案。執行傳輸前，從站必須先切換至 Bootstrap 狀態。

欄位 / 控制項	說明
Download File List (下載檔案清單)	USB 磁碟的檔案瀏覽器，顯示資料夾與檔案。點選目標韌體檔案後再點擊 Download。
FoE File Name (FoE 檔案名稱)	傳送給從站 (或從從站請求) 的檔案名稱。須與從站 Bootloader 所預期的名稱相符 (例如：app.efw)。
FoE Password (FoE 密碼，十六進位)	8 位十六進位的 FoE 存取密碼。請填入從站廠商指定的值。預設值：00000000。
Upload File Name (上傳儲存檔名)	從從站取得的資料將儲存至 USB 的檔案名稱 (例如：upload.efw)。
Download 按鈕	讀取 Download File List 中所選的檔案，透過 FoE Write 傳送至從站。完成後以訊息框顯示結果。
Upload 按鈕	透過 FoE Read 從從站取得韌體，並儲存至 USB 的 Upload File Name 路徑下。最大傳輸大小：2 MB。

注意 檔案傳輸可能需要數秒。傳輸期間請勿切換 Begin 或 Start 開關。

4. Process Data 分頁 (程序資料)



[圖示：畫面截圖 — 圖3 — Process Data 分頁]

Process Data 分頁由左至右分為三個區塊：PDO Assignment 0x1C12 (接收 PDO 指定)、PDO Assignment 0x1C13 (傳送 PDO 指定)，以及 PDO Configuration (PDO 映射設定)。PDO 操作需在 Begin 啟用、Start 關閉的狀態下進行。

注意 PDO 設定須在 Pre-Operational 狀態下執行。Start 啟動後，寫入操作將被禁用。

4.1 PDO Assignment 0x1C12 (接收 PDO 指定)

設定哪些 RxPDO 映射物件 (索引範圍 0x1600–0x17FF) 被指定至從站的接收通道。

控制項	說明
Index 欄位	輸入欲新增或刪除的 RxPDO 映射索引 (十六進位，例如：1600)。
Upload 按鈕	從從站讀取目前的 0x1C12 指定清單，並顯示於下方表格。
Download 按鈕	將編輯後的指定清單寫回從站的 0x1C12 物件。須在 Start 關閉時操作。
Add 按鈕	將 Index 欄位中輸入的索引值加入指定清單末尾。
Delete 按鈕	從指定清單中移除 Index 欄位中的索引值。
Move 按鈕	將符合的項目向上移動一個位置。
指定清單表格	顯示最多 16 個已指定的 RxPDO 索引。

4.2 PDO Assignment 0x1C13 (傳送 PDO 指定)

操作方式與 4.1 節完全相同，但作用於傳送通道 (0x1C13)。有效索引範圍：0x1A00–0x1BFF。

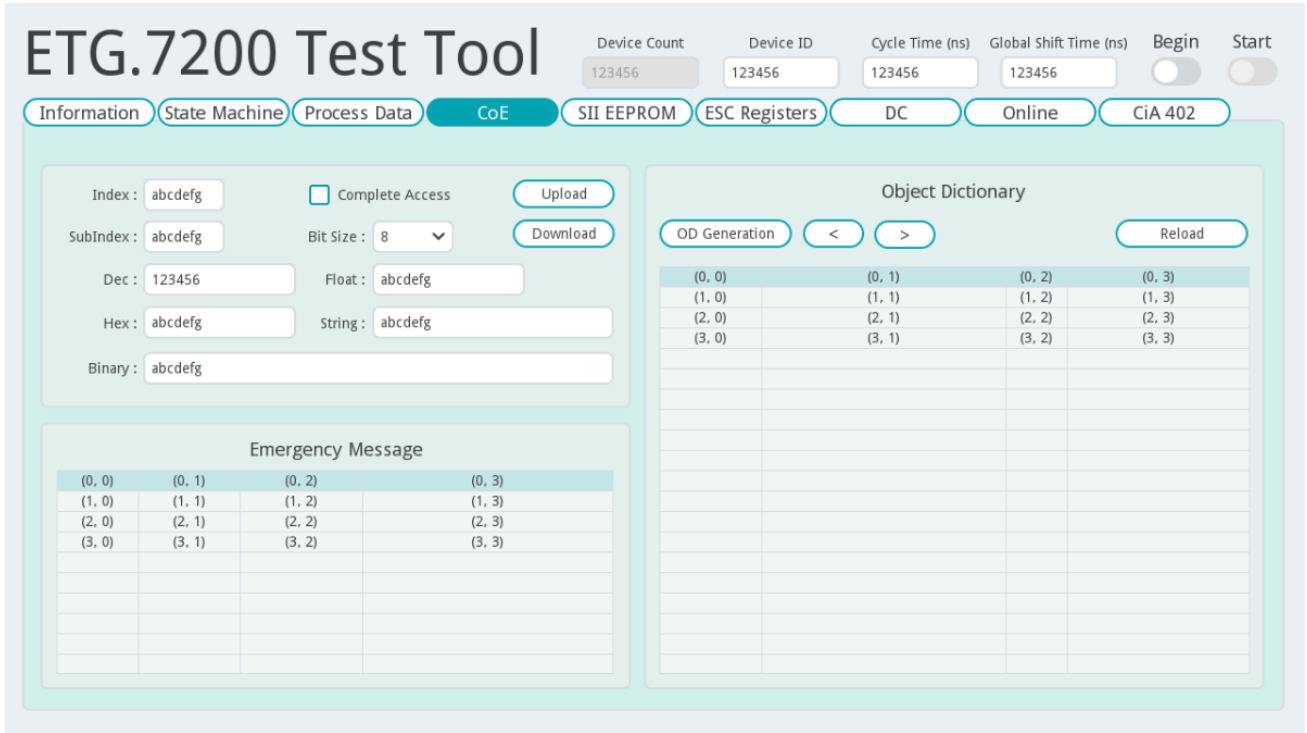
控制項	說明
Index 欄位	輸入欲新增或刪除的 TxPDO 映射索引 (十六進位，例如：1A00)。
Upload / Download	從從站讀取或寫入 0x1C13 物件。
Add / Delete / Move	操作方式與 4.1 節相同，但作用於 TxPDO 清單。
指定清單表格	顯示最多 16 個已指定的 TxPDO 索引。

4.3 PDO Configuration (PDO 映射設定)

讀取並編輯使用者所選定之 PDO 物件的映射項目。

控制項 / 欄位	說明
PDO Mapping Index (PDO 映射索引)	輸入欲設定的 PDO 映射物件索引 (例如：1600 代表第一個 RxPDO，1A00 代表第一個 TxPDO)。
Upload 按鈕	透過 SDO 從從站讀取目前的映射設定，並填入映射表格。
Download 按鈕	將映射表格的內容寫回從站。須在 Start 關閉時操作。
< 和 > 按鈕	當映射項目超過可顯示的行數時，用於前後捲動。
映射表格	以欄列形式顯示各映射項目的 Index、SubIndex 與 Bit Size。
Index 欄位	欲新增映射的物件索引 (十六進位，例如：6040)。
SubIndex 欄位	欲新增映射的物件子索引 (十六進位，例如：00)。
Bit Size 欄位	映射資料的位元寬度 (例如：16)。
Add 按鈕	將指定的 Index / SubIndex / Bit Size 項目附加至映射表格末尾。
Delete 按鈕	從映射表格中移除符合 Index / SubIndex / Bit Size 的項目。
Move 按鈕	將符合的項目向上移動一個位置。

5. CoE 分頁



[圖示：畫面截圖 — 圖 4 — CoE 分頁]

CoE 分頁包含兩個並列區塊：左側為直接 SDO 存取面板，右側為物件字典 (OD) 瀏覽器。左下方另有緊急訊息 (Emergency Message) 紀錄表。

5.1 SDO 上傳 / 下載

對從站上任意指定的物件執行單次 SDO Upload (讀取) 或 SDO Download (寫入) 。
一次只能執行一項欄位功能。

欄位 / 控制項	說明
Index (索引)	4 位十六進位的物件索引 (例如：6040) 。
SubIndex (子索引)	2 位十六進位的子索引 (例如：00) 。
Bit Size (位元大小，下拉選單)	資料寬度：8、16、32 或 64 位元。 勾選 Complete Access 時停用。
Complete Access 核取方塊	勾選後，以單次 SDO 操作傳輸整個物件 (所有子索引) 。 SubIndex 與 Bit Size 控制項將同時停用。
Dec (十進位)	資料的有號十進位表示。修改此欄位時，Hex、Float、String 及 Binary 欄位將被清除。
Hex (十六進位)	資料的十六進位表示。同時修改時優先於 Dec 欄位。
Float (浮點數)	資料的浮點數表示。
String (字串)	資料的 ASCII 字串表示。
Binary (二進位原始資料)	由 Upload 填入的原始位元組傾印內容；亦可手動編輯後作為 Download

	的寫入資料。
Upload 按鈕	從從站讀取物件，並將結果填入所有數值欄位。
Download 按鈕	將目前數值欄位的內容寫入從站。

5.2 物件字典瀏覽器 (Object Dictionary)

透過 SDO Info 服務列舉從站的所有可存取物件，並在表格中顯示 Index (索引)、Name (名稱)、Flags (存取旗標) 及 Value (數值) 四個欄位。

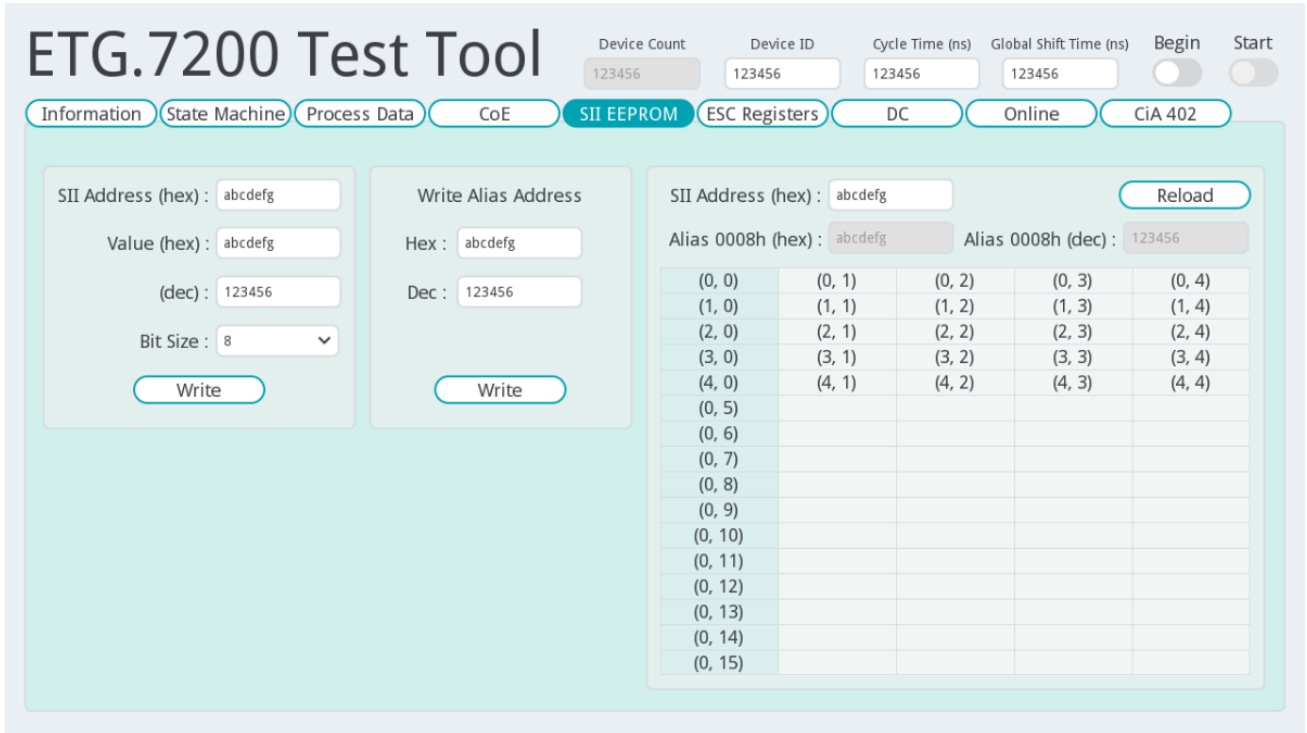
控制項 / 欄位	說明
OD Generation 按鈕	從從站查詢完整的物件字典清單，並重設畫面至第一頁。
< 和 > 按鈕	每頁顯示 19 筆，用於前後翻頁。
Reload 按鈕	對目前可見的所有可讀物件執行 SDO Upload，重新讀取數值並更新 Value 欄位。
Index 欄	物件索引與子索引，格式為 XXXX:XX (十六進位)。
Name 欄	從站回報的物件或項目名稱字串。
Flags 欄	存取權限標示：RO (唯讀)、WO (唯寫) 或 RW (可讀寫)。
Value 欄	執行 Reload 後的目前數值，依資料型別格式化 (十進位、十六進位、浮點數或字串)。

5.3 緊急訊息紀錄 (Emergency Message)

擷取通訊期間從任何從站接收到的 CoE Emergency 訊息。環形緩衝區最多儲存 9 筆；緩衝區滿時，最舊的訊息將被覆蓋。

欄位	說明
欄 0	發送緊急訊息的從站編號 (Device ID)。
欄 1	16 位元緊急錯誤碼 (Emergency Error Code)。
欄 2	8 位元錯誤暫存器值 (Error Register)。
欄 3	5 個位元組的廠商自定義錯誤資料 (十六進位)。

6. SII EEPROM 分頁



[圖示：畫面截圖 — 圖 5 — SII EEPROM 分頁]

SII EEPROM 分頁分為三個區塊：左側為直接寫入面板，中間為站台別名 (Station Alias) 寫入面板，右側為 EEPROM 讀取與顯示面板。

注意 寫入資料至 SII EEPROM 是屬於進階功能。寫入前請務必確認所有數值正確無誤。

6.1 直接寫入面板

欄位 / 控制項	說明
SII Address (SII 位址，十六進位)	寫入操作的起始字組偏移位址。
Value (數值，十六進位)	欲寫入指定位址的十六進位資料。
(dec) (十進位)	寫入數值的十進位表示，與 Hex 欄位同步。
Bit Size (位元大小，下拉選單)	寫入資料寬度：8、16、32 或 64 位元。
Write 按鈕	將指定數值寫入 SII EEPROM 的對應位址。

注意 寫入不正確的資料至 SII EEPROM 可能導致從站無法回應。寫入前請務必確認所有數值正確無誤。

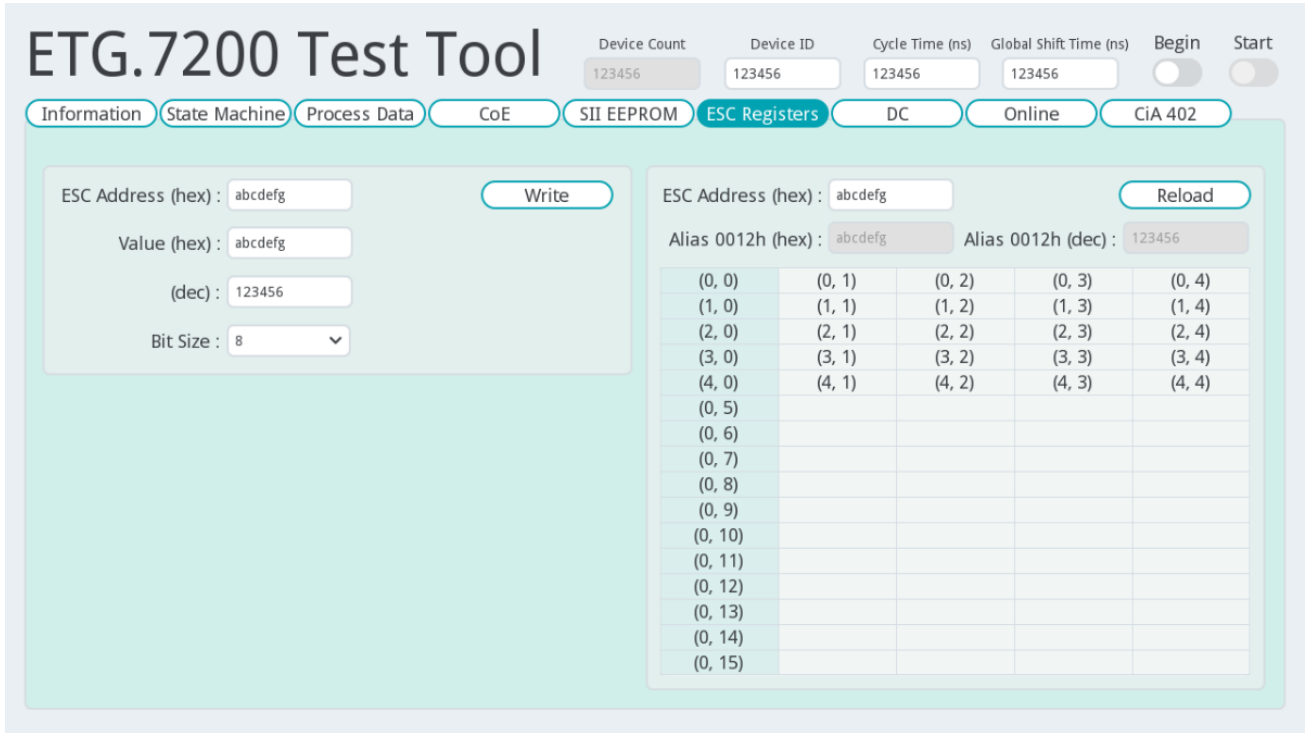
6.2 寫入站台別名面板 (Write Alias Address)

欄位 / 控制項	說明
Hex 欄位	以十六進位輸入站台別名值。
Dec 欄位	以十進位輸入站台別名值 (與 Hex 欄位同步)。
Write 按鈕	將站台別名寫入 SII EEPROM。操作前從站應處於 Init 或 Pre-Op 狀態。

6.3 EEPROM 讀取 / 顯示面板

欄位 / 控制項	說明
SII Address (SII 位址，十六進位)	讀取顯示的起始位址。
Reload 按鈕	從指定位址開始讀取 EEPROM 內容，並填入下方表格。
Alias 0008h (十六進位，唯讀)	Reload 後顯示 EEPROM 字組 0x0008 (Station Alias) 的內容。
Alias 0008h (十進位，唯讀)	別名值的十進位表示。
資料表格	以欄列格式顯示 EEPROM 原始內容。

7. ESC Registers 分頁 (ESC 暫存器)



[圖示：畫面截圖 — 圖 6 — ESC Registers 分頁]

ESC Registers 分頁提供對 EtherCAT 從站控制器 (ESC) 暫存器空間的低階讀寫存取。介面佈局與 SII EEPROM 分頁相同：左側為寫入面板，右側為讀取與顯示面板。

注意 ESC 暫存器的寫入操作立即生效且不可復原。是屬於進階功能。對保留位址或關鍵功能暫存器 (例如 AL Control 0x0120) 進行寫入，可能導致裝置狀態意外改變。

7.1 寫入面板

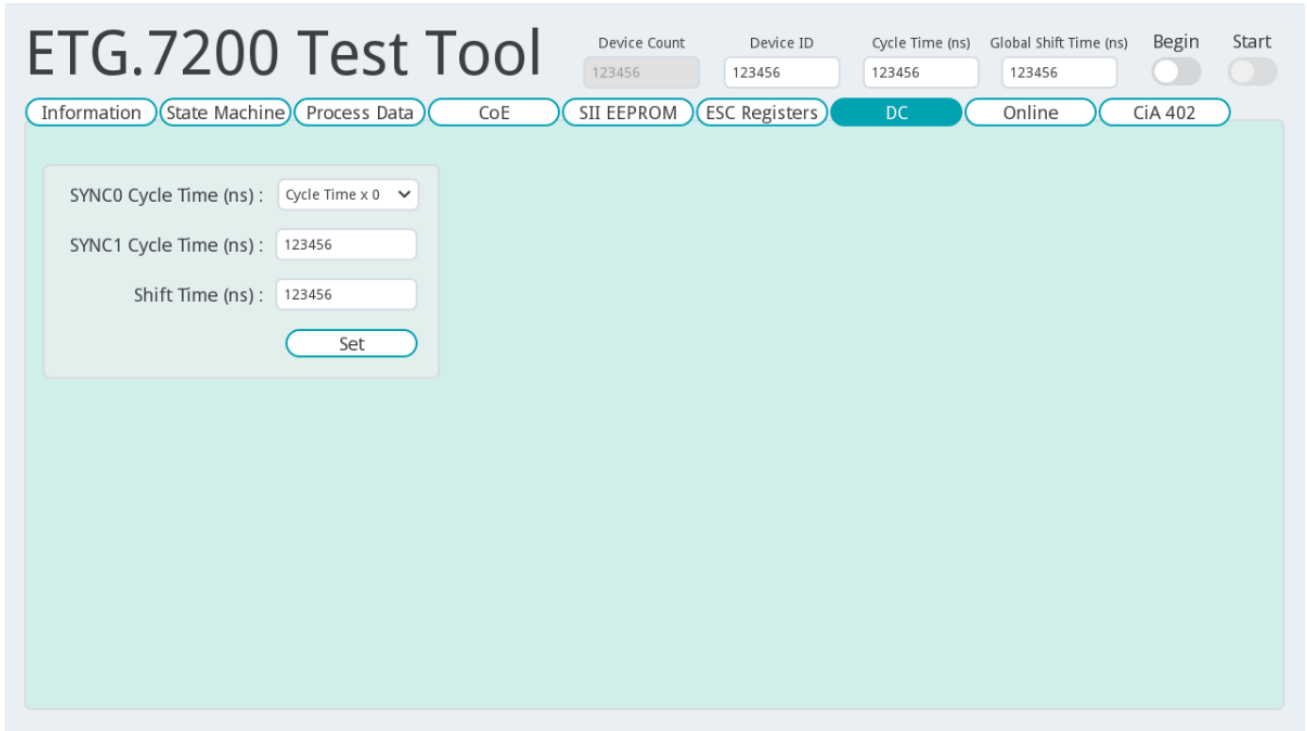
欄位 / 控制項	說明
ESC Address (ESC 位址，十六進位)	欲寫入的 4 位十六進位暫存器位址 (例如：0120 代表 AL Control)。
Value (數值，十六進位)	欲寫入的十六進位數值。
(dec) (十進位)	寫入數值的十進位表示。
Bit Size (位元大小，下拉選單)	資料寬度：8、16、32 或 64 位元。
Write 按鈕	立即將指定數值寫入 ESC 暫存器。需先啟用 Begin。

注意 ESC 暫存器的寫入操作立即生效且不可復原。對保留位址或關鍵功能暫存器 (例如 AL Control 0x0120) 進行寫入，可能導致裝置狀態意外改變。

7.2 讀取 / 顯示面板

欄位 / 控制項	說明
ESC Address (ESC 位址，十六進位)	讀取操作的起始位址。
Reload 按鈕	從指定位址讀取 256 個位元組並顯示於下方表格，同時讀取暫存器 0x0012 的站台別名值。
Alias 0012h (十六進位，唯讀)	Reload 後顯示 ESC 暫存器 0x0012 (Station Alias) 的值。
Alias 0012h (十進位，唯讀)	別名值的十進位表示。
暫存器表格	以 16 列 × 16 位元組的格式顯示 ESC 暫存器內容。每列標題顯示起始位址，其餘欄位以十六進位群組顯示資料。

8. DC 分頁 (分散式時鐘)



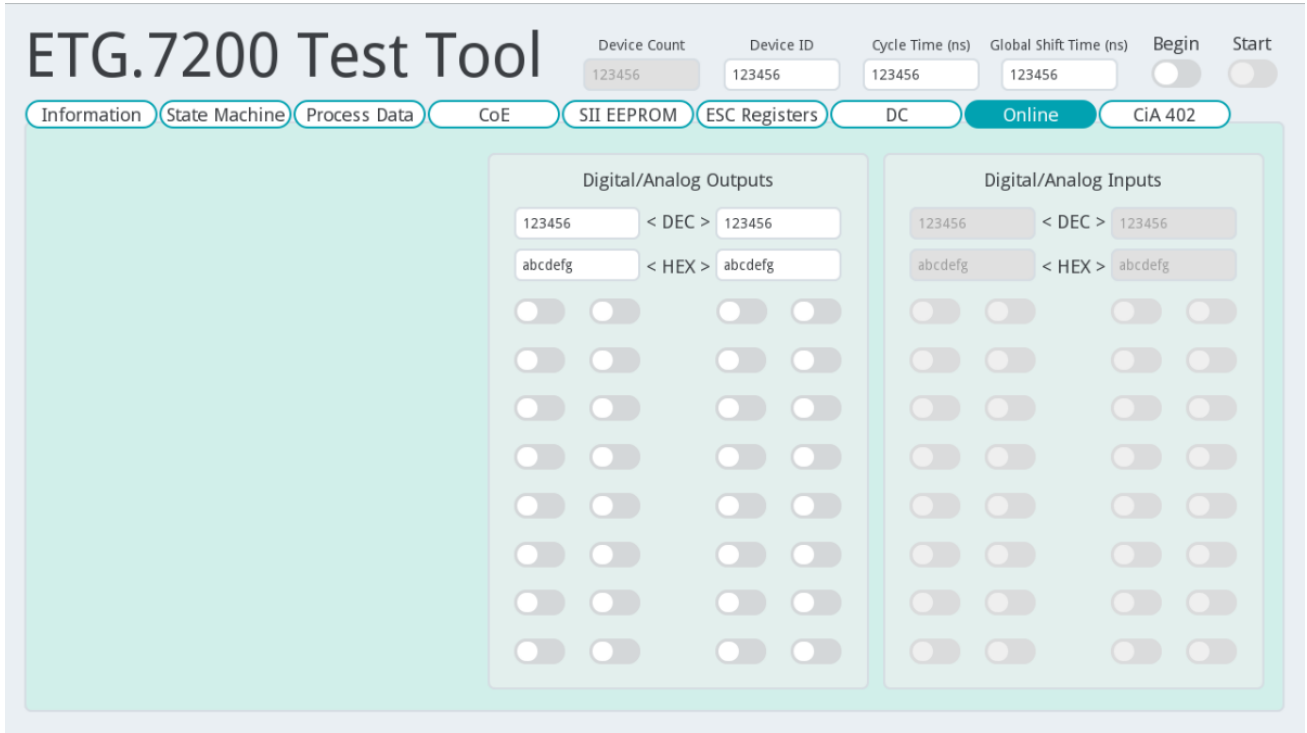
[圖示：畫面截圖 — 圖 7 — DC 分頁]

DC 分頁設定所選從站的分散式時鐘 (Distributed Clocks, DC) 同步參數。點擊 **Set** 按鈕後，系統將呼叫 `master.setDc()` 套用設定。此操作必須在 **Begin** 啟用、**Start** 關閉時進行。

欄位 / 控制項	說明
SYNC0 Cycle Time (ns) (下拉選單)	將全域 Cycle Time 乘以所選倍數，作為 SYNC0 脈衝週期。選項：Cycle Time × 0 (停用) 至 × 10。
SYNC1 Cycle Time (ns)	SYNC1 脈衝週期，單位為奈秒。設為 0 可停用 SYNC1。數值須 ≥ 0。
Shift Time (ns)	相對於網路參考時鐘的 SubDevice 本地 DC 位移時間，單位為奈秒。接受負值。
Set 按鈕	以上述三個參數呼叫 <code>master.setDc()</code> ，並以訊息框顯示執行結果。僅在 Begin 啟用且 Start 關閉時可操作。

注意 SYNC0 倍數下拉選單的絕對時間值是依據目前的 Cycle Time 計算而來。若需非預設週期，請先修改 Cycle Time 再設定 DC。

9. Online 分頁 (線上 I/O)



[圖示：畫面截圖 — 圖 8 — Online 分頁]

Online 分頁提供 PDO 數位輸入的即時監控，以及 PDO 數位輸出的即時控制。每側最多支援 32 位元資料 (兩個 16 位元字組)。此分頁需在 Start 啟用後才能正常作業。

9.1 數位 / 類比輸出 (左側面板)

輸出面板提供三種同步輸入方式控制 PDO 輸出值：十進位字組、十六進位字組及個別位元切換開關。三種表示方式保持同步。Start 啟用後，輸出資料在每個 PDO 週期寫入從站。

元件	說明
DEC 欄位 (Word 1 / Word 2)	以無號十進位整數表示的 16 位元輸出值。範圍：0–65535。Word 1 對應位元 0–15；Word 2 對應位元 16–31。
HEX 欄位 (Word 1 / Word 2)	相同數值的 4 位十六進位字串表示。修改 Hex 欄位時，Dec 欄位與位元切換開關同步更新。
位元切換開關 (共 32 個)	個別位元控制開關，排列為 8 欄 × 4 列。切換任一開關時，對應的 Dec 與 Hex 欄位立即更新。

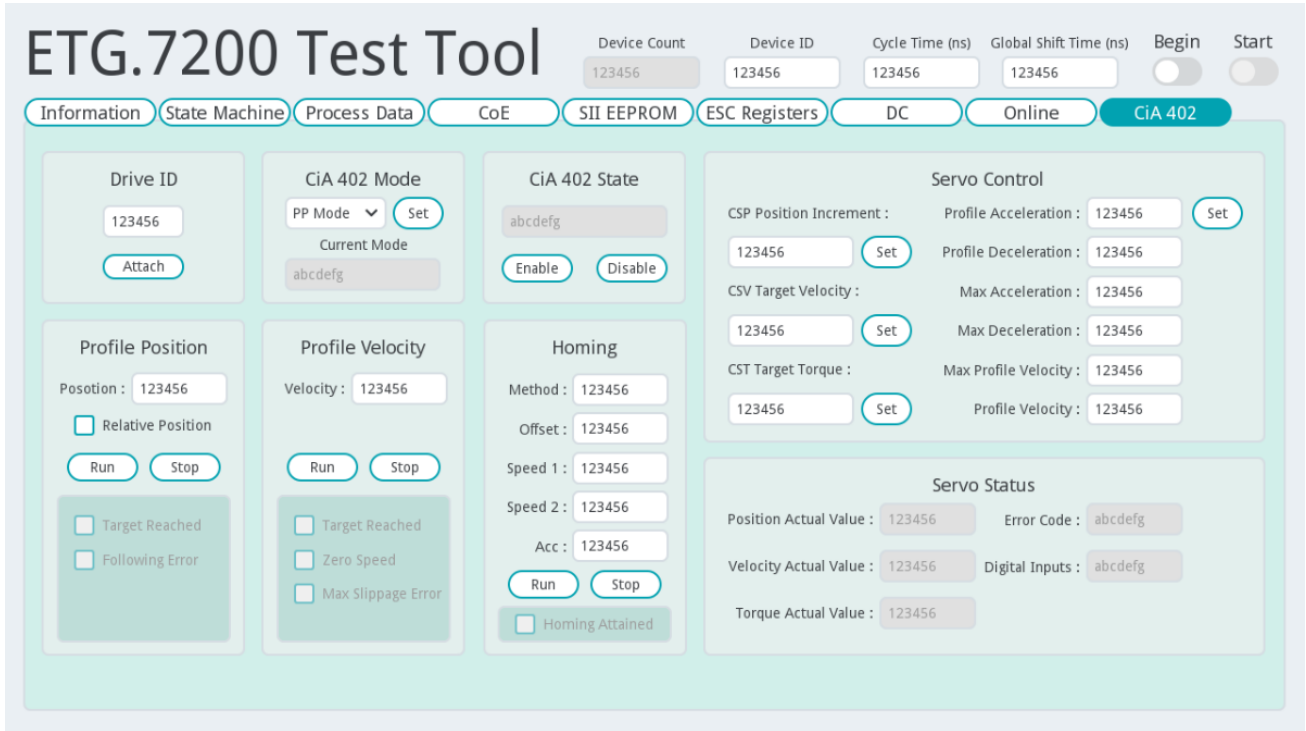
注意 超出從站所設定的 PDO 輸出寬度的位元，在寫入前會自動遮罩為 0。

9.2 數位 / 類比輸入 (右側面板)

輸入面板反映從站最新接收到的 PDO 資料。所有欄位均為唯讀，在每個輪詢週期自動更新。

元件	說明
DEC 欄位 (Word 1 / Word 2)	目前輸入值的無號十進位整數表示。
HEX 欄位 (Word 1 / Word 2)	目前輸入值的 4 位十六進位字串表示。
位元指示開關 (共 32 個)	唯讀的切換指示器，每個開關對應一個輸入位元。ON = 邏輯 1；OFF = 邏輯 0。

10. CiA 402 分頁 (伺服驅動器)



[圖示：畫面截圖 — 圖9 — CiA 402 分頁]

CiA 402 分頁實現 IEC 61800-7 / CiA 402 伺服驅動器輪廓，包含五個功能區塊：驅動器 ID 與連接、操作模式設定、CiA 402 狀態控制、伺服控制參數，以及各模式專屬的運動命令面板。

10.1 Drive ID 與 Attach (驅動器連接)

控制項	說明
Drive ID 欄位	EtherCAT 節點內的軸編號 (0–255)。須與驅動器韌體設定的軸編號一致。
Attach 按鈕	呼叫 <code>motor.attach()</code> ，將 CiA 402 驅動器物件連接至所選的 Device ID 與 Drive ID。啟用 Attach 時，系統會自動從驅動器讀取運動參數並預先填入各欄位。再次點擊可執行 Detach 斷開連接。需先啟用 Begin。

10.2 CiA 402 Mode (操作模式)

控制項	說明
模式下拉選單	選擇目標操作模式：PP (輪廓位置)、PV (輪廓速度)、HM (原點復歸)、CSP (循環同步位置)、CSV (循環同步速度)、CST (循環同步扭矩)。
Set 按鈕	呼叫 <code>motor.setCiA402Mode()</code> 設定所選模式。切換至 CSP 模式時，系統會鎖存目前實際位置作為初始目標位置，防止位置跳變。
Current Mode (目前模式, 唯讀)	顯示驅動器目前回報的操作模式。

10.3 CiA 402 State (驅動器狀態)

控制項	說明
狀態顯示欄位 (唯讀)	目前的 CiA 402 狀態機狀態：Not Ready To Switch On、Switch On Disabled、Ready To Switch On、Switch On、Operation Enabled、Fault、Fault Reaction Active 或 Quick Stop Active。
Enable 按鈕	執行 CiA 402 標準啟用程序，使驅動器進入 Operation Enabled 狀態。
Disable 按鈕	將驅動器切換回 Switch On Disabled 狀態。

10.4 Servo Control — 運動參數

以下欄位適用於 PP 與 PV 模式。執行 Attach 時自動從驅動器預載，點擊 Set 後一次寫入驅動器。

欄位	物件 / 說明
Profile Acceleration (輪廓加速度)	物件 0x6083 — 加速度，單位：使用者單位/s ² 。
Profile Deceleration (輪廓減速度)	物件 0x6084 — 減速度，單位：使用者單位/s ² 。
Max Acceleration (最大加速度)	物件 0x60C5 — 最大允許加速度。
Max Deceleration (最大減速度)	物件 0x60C6 — 最大允許減速度。
Max Profile Velocity (最大輪廓速度)	物件 0x607F — 輪廓運動的速度上限。
Profile Velocity (輪廓速度)	物件 0x6081 — PP 模式運動的目標速度。
Set 按鈕	一次將上述六個參數寫入驅動器。

10.5 Servo Control — 循環同步命令 (CSP / CSV / CST)

以下命令透過 CyclicCallback 在每個 EtherCAT 週期執行。驅動器須處於 Operation Enabled 狀態，命令才會生效。

欄位 / 按鈕	說明
CSP Position Increment (CSP 位置增量)	每個週期累加的位置增量 (例如：100 表示在 1 ms 週期下每次移動 100 counts)。負值可反向運動。
CSP Set 按鈕	套用 CSP 位置增量命令。驅動器須處於 CSP 模式且 Operation Enabled。
CSV Target Velocity (CSV 目標速度)	CSV 模式的定速命令，單位：使用者單位/s。負值可反向運動。
CSV Set 按鈕	套用 CSV 速度命令。
CST Target Torque (CST 目標扭矩)	CST 模式的定扭矩命令。範圍：-32768 至 32767 (額定扭矩的 0.1%)。
CST Set 按鈕	套用 CST 扭矩命令。

10.6 Profile Position 面板 (輪廓位置模式)

欄位 / 按鈕	說明
Position 欄位	目標位置，單位：使用者單位。
Relative Position 核取方塊	勾選時以相對位移執行；不勾選時以絕對位置執行。
Run 按鈕	執行輪廓位置運動至目標位置 (<code>motor.pp_Run()</code>)。
Stop 按鈕	發出相對位移 0 的命令，使驅動器停止並確認新設定點。
Target Reached (唯讀核取方塊)	反映狀態字 — 驅動器到達目標位置時設為 ON。
Following Error (唯讀核取方塊)	反映狀態字 — 追隨誤差超過設定限制時設為 ON。

10.7 Profile Velocity 面板 (輪廓速度模式)

欄位 / 按鈕	說明
Velocity 欄位	目標速度，單位：使用者單位/s。
Run 按鈕	命令驅動器加速至目標速度 (<code>motor.pv_Run()</code>)。
Stop 按鈕	命令驅動器減速至零速 (<code>motor.pv_Run(0)</code>)。
Target Reached (唯讀核取方塊)	驅動器速度達到目標速度時設為 ON。
Zero Speed (唯讀核取方塊)	驅動器速度低於零速閾值時設為 ON。
Max Slippage Error (唯讀核取方塊)	指令速度與實際速度的差值超過限制時設為 ON。

10.8 Homing 面板 (原點復歸模式)

欄位 / 按鈕	說明
Method (復歸方法)	整數型復歸方法代碼 (-128 至 127)。預設值：33 (索引脈衝復歸)。完整定義請參閱 CiA 402 第 13 節。
Offset (原點偏移)	復歸完成後附加至原點位置的有號 32 位元偏移量。
Speed 1 (搜尋速度)	搜尋限位開關時使用的速度。
Speed 2 (歸零速度)	搜尋原點位置或索引脈衝時使用的爬行速度。
Acc (加減速度)	復歸移動期間使用的加速度與減速度。
Run 按鈕	將所有復歸參數寫入驅動器，並啟動復歸程序 (<code>motor.hm_Run()</code>)。
Stop 按鈕	停止復歸操作 (<code>motor.hm_Stop()</code>)。
Homing Attained (唯讀核取方塊)	驅動器回報復歸程序成功完成時設為 ON。

10.9 Servo Status (伺服狀態)

以下所有欄位在 **Attach** 啟用期間於每個輪詢週期自動更新。

欄位	說明
Position Actual Value (實際位置)	物件 0x6064 的目前位置，單位：使用者單位。
Velocity Actual Value (實際速度)	物件 0x606C 的目前速度。
Torque Actual Value (實際扭矩)	物件 0x6077 的目前扭矩 (額定扭矩的 0.1%)。
Error Code (錯誤碼)	物件 0x603F 的 4 位十六進位驅動器錯誤碼。
Digital Inputs (數位輸入)	驅動器數位輸入物件 0x60FD 的 8 位十六進位數值。

11. 故障排除

現象	可能原因	處理方式
Begin 切換後顯示錯誤碼	根據錯誤代碼排查	根據顯示錯誤碼，去查錯誤碼對照表。
Start 開關呈灰色無法操作	Bootstrap 模式處於啟用狀態	在 State Machine 分頁將 Bootstrap 關閉後，再啟用 Start。
SDO Upload / Download 回傳錯誤	從站未進入 Pre-Op 或更高狀態，或根據錯誤代碼排查	確認 Begin 已啟用，且從站已至少切換至 Pre-Op 狀態。可根據顯示錯誤碼，去查錯誤碼對照表。
FoE Download 失敗	從站未處於 Bootstrap 狀態	先在 State Machine 分頁命令從站進入 Bootstrap 狀態，再執行 FoE 傳輸。
CiA 402 Attach 失敗	Drive ID 不符	確認 Drive ID 與驅動器韌體設定的軸編號一致。
CSP 模式 Attach 時發生位置跳變	初始目標位置陳舊	系統已自動防止此問題：Attach 時會將目前實際位置鎖存為 CspTargetPosition。
CoE 分頁出現緊急訊息	驅動器或 I/O 發生故障	記錄 Error Code 與 Error Register，並查閱從站廠商的錯誤代碼說明文件。
位元切換開關操作無效	Start 尚未啟用	輸出位元切換開關須在 Start 啟用後才能對從站生效。

12. 實際操作教學

12.1 情境 A — 首次連接與識別從站

目的：初次將一台 EtherCAT 從站接上 QEC 平台後，透過 ETG.7200 工具建立主站與從站連線，並從 Information 分頁讀取該從站的識別資訊，以確認硬體、接線與工具溝通均正常。

前置條件：

- QEC 硬體 (86Duino + 86HMI) 已通電，HMI 開機完成並顯示 ETG.7200 主畫面。
- 欲測試的 EtherCAT 從站已連接至 QEC 所指定的 EtherCAT 連接埠 (連接 ETH0)，且從站已獨立供電。
- 從站規格書 (或 ESI 檔) 已備妥，可供比對 Vendor ID、Product Code 等識別資訊。

操作步驟：

1. 確認全域工具列上的 Cycle Time 維持預設值 1,000,000 ns，Global Shift Time 維持 0。

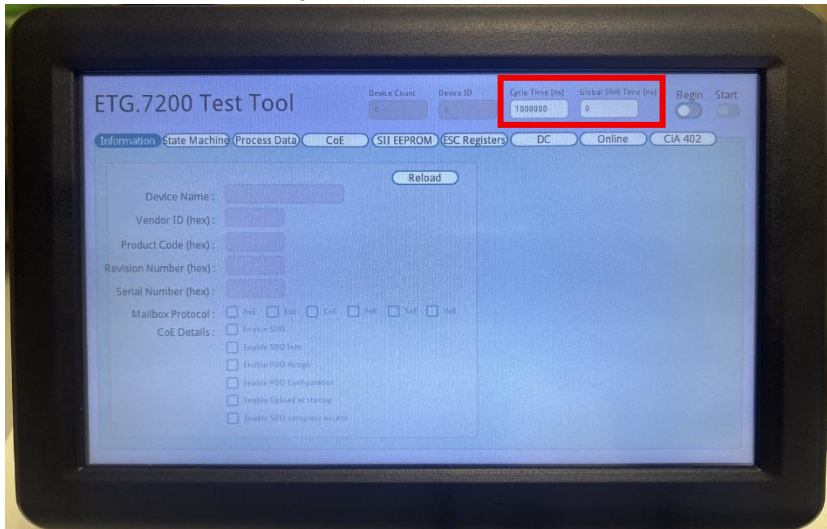


圖 12.1-1 Begin 撥 ON 前，Cycle Time 維持預設值。

2. 將 Begin 開關撥為 ON。系統將呼叫 `master.begin()` 初始化 EtherCAT 網路並掃描從站。
3. 檢視 Device Count 欄位，應自動更新為所連接的從站數量。若僅接一台，Device Count 應為 1。

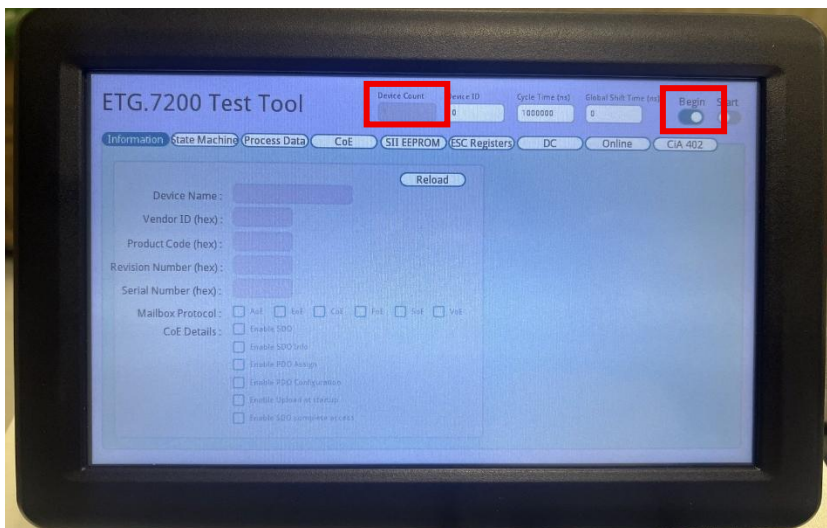


圖 12.1-2 Begin 撥 ON 後，Device Count 自動更新為偵測到的從站數量。

4. 設定 Device ID = 0，以選擇目前作業的目標從站。
5. 點擊分頁標籤切換至 Information 分頁。
6. 點擊 Reload 按鈕，工具會向從站查詢並更新所有識別欄位。

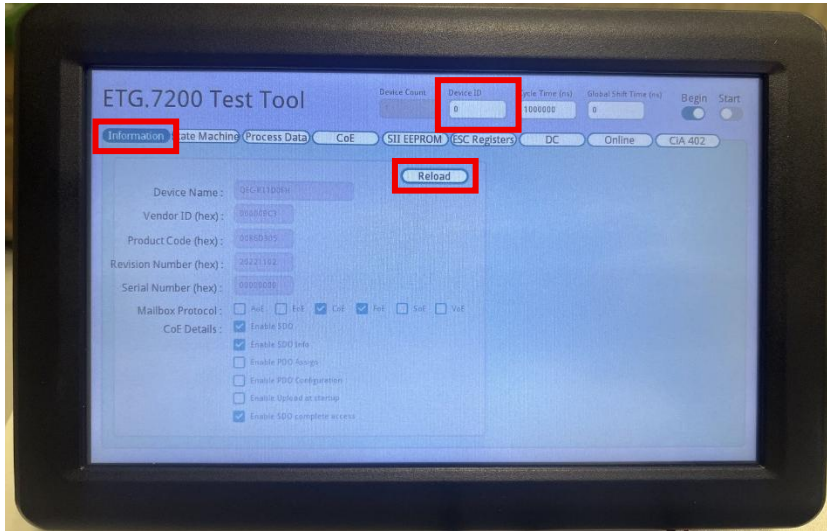


圖 12.1-3 Reload 成功後 Information 分頁所有欄位皆填入值，並勾選所支援的 Mailbox 協定。

7. 檢視 Device Name、Vendor ID、Product Code、Revision Number 與 Serial Number，並對照從站規格書或 ESI 檔加以確認。
8. 檢視 Mailbox Protocol 與 CoE Details 區塊的核取方塊，確認從站所宣告支援的通訊協定 (如 CoE、FoE) 與工具後續測試需求一致。

預期結果：

- Device Count 顯示的數值與實際接上的從站數量相符。
- Information 分頁中各識別欄位皆有有效數值，Vendor ID 與 Product Code 與規格書一致。
- Mailbox Protocol 區塊至少會勾選 CoE (絕大多數 EtherCAT 從站皆支援)。

驗證方式：

將 Information 分頁顯示的 Vendor ID (十六進位) 與從站規格書或 ESI 檔的 Vendor Id 欄位相互比對；若完全一致，代表主站與從站已可正確溝通信箱通訊。

常見問題：

- Device Count 顯示為 0：檢查 EtherCAT 線材、從站電源以及 QEC 使用的網路介面是否正確。可先拔除並重新插入 EtherCAT 線，再將 Begin 撥 OFF 後再撥 ON 重試。
- Reload 按鈕無反應或回傳錯誤：確認 Begin 已啟用；若仍失敗，先將從站切換至 Pre-Op 或更高狀態，再重新 Reload。
- 識別欄位出現異常值或亂碼：從站 SII EEPROM 內容可能損毀，可至 SII EEPROM 分頁檢查並重新燒錄。

注意 本情境結束時可保持 Begin 為 ON、Start 為 OFF，以利後續情境 B、C 對 PDO 與 DC 進行設定。

12.2 情境 B — 設定 PDO 並驗證 I/O

目的：以一台具備數位或類比 I/O 的 EtherCAT 從站為例，透過 Process Data 分頁設定 PDO Assignment 與 PDO Mapping，並在 Online 分頁實際驗證輸出與輸入訊號。

前置條件：

- 已完成情境 A，可透過 Information 分頁讀取從站識別資訊。
- Begin 為 ON、Start 為 OFF。PDO Mapping 必須在 Start 關閉狀態下進行。
- 已備妥從站的 ESI 檔或 PDO 映射說明，可確認可用的 RxPDO / TxPDO 索引與對應物件 (0x1C12 & 0x1C13)。
- 設定 Begin 為 ON、Start 為 ON。EtherCAT 進入 OP 狀態，用 PDO 執行 I/O 功能。
- 若為輸出驗證，現場有可觀測的指示燈、繼電器或實際受控裝置；若為輸入驗證，現場有可觸發的訊號源 (按鈕、感測器等)。

操作步驟：

1. 切換至 State Machine 分頁，確保從站位於 Pre-OP 狀態 (Current State 顯示 Pre-Op)。

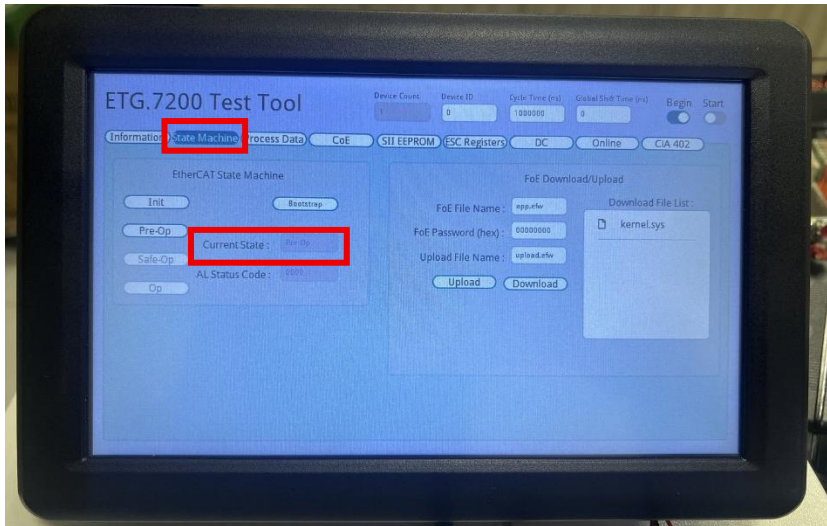


圖 12.2-1 切換至 State Machine 分頁，Current State 顯示 Pre-Op。

2. 切換至 Process Data 分頁。

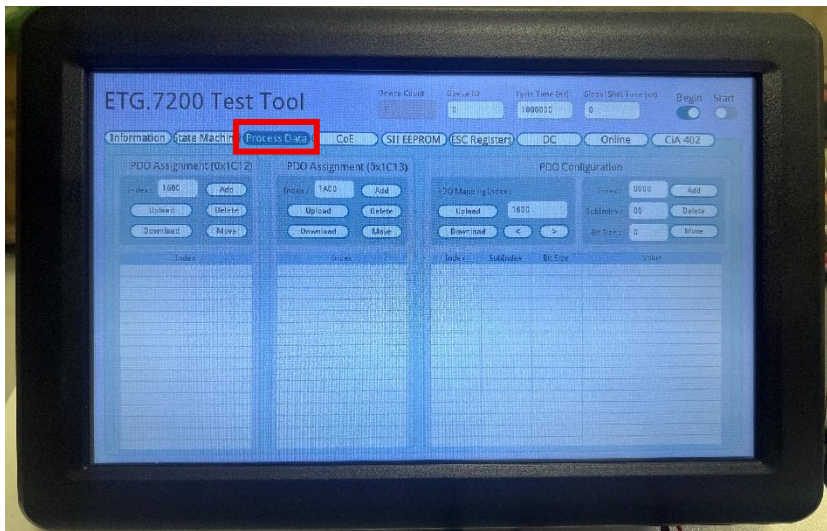


圖 12.2-2 Process Data 分頁初始畫面，PDO Assignment 與 Configuration 區塊尚未填入任何項目。

3. 在 PDO Assignment 0x1C12 (接收 PDO 指定) 區塊，這是欲啟用的 RxPDO 索引 (例如 0x1600)。可以點擊 Upload 按鈕，將會把預設的 0x1C12 物件顯示在下方表單中；點擊 Download 按鈕，則是會透過 SDO 下載 Assignment 與 Mapping 至從站寫入 PDO Mapping，可以再次點擊 Upload 按鈕檢查是否成功寫入。

- 在 PDO Assignment 0x1C13 (傳送 PDO 指定) 區塊，這是欲啟用的 TxPDO 索引 (例如 0x1A00)。做法與第 3 點相同。(因文件是展示 Digital Output 模組，無 TxPDO 索引)

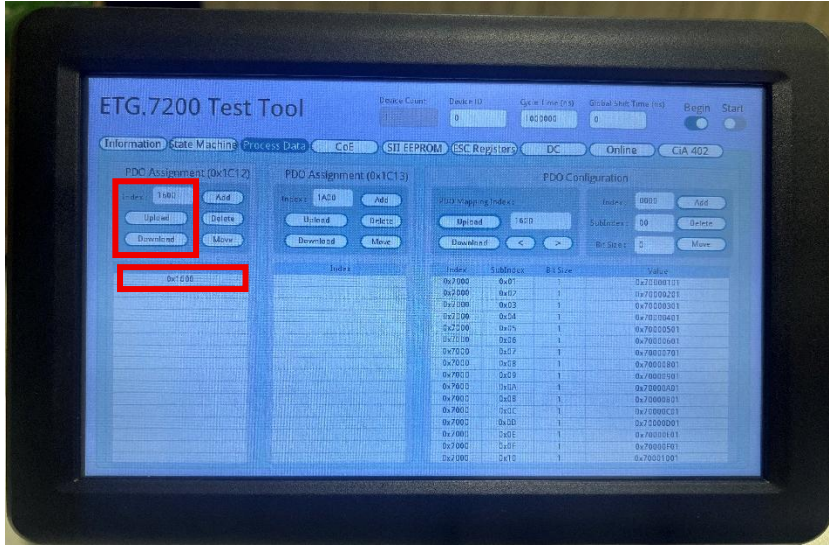


圖 12.2-3 點擊 0x1C12 的 Upload 按鈕後，左側 Index 區顯示已指派的 RxPDO (例如 0x1600)。

- 在 PDO Configuration 區塊中，針對每筆被指定的 PDO，檢視或修改映射項目 (Index、Subindex、Bit Length)。若需自訂，可點擊新增列並填入目標物件與位元長度。

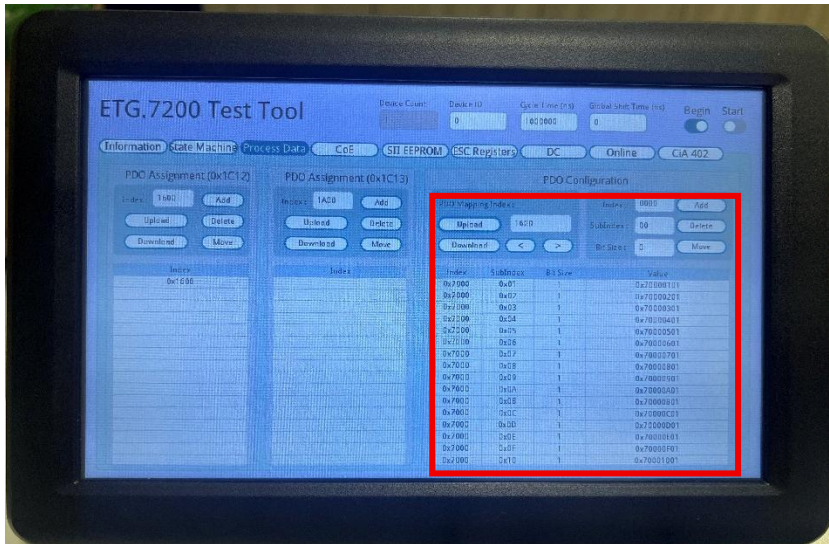


圖 12.2-4 PDO Configuration 區塊顯示 0x1600 的映射項目 (SubIndex、Bit Size、Value)。

- 完成後，則可以將 Start 開關撥為 ON，設定 EtherCAT 狀態為 OP，啟用週期性 PDO 通訊。
- 確認 Current State 顯示 Op，AL Status Code 為 0x0000。

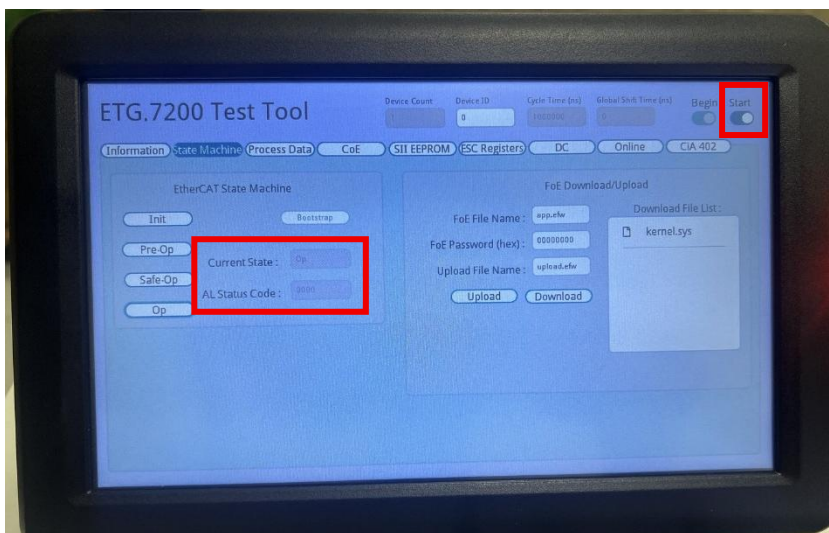


圖 12.2-5 State Machine 分頁：Start 啟用後 Current State 切換至 Op，AL Status Code 為 0x0000。

8. 切換至 Online 分頁，檢視左側輸出面板與右側輸入面板。
9. 撥動左側輸出區塊中任一開關，或於類比輸出欄位輸入數值，觀察現場對應裝置是否動作。

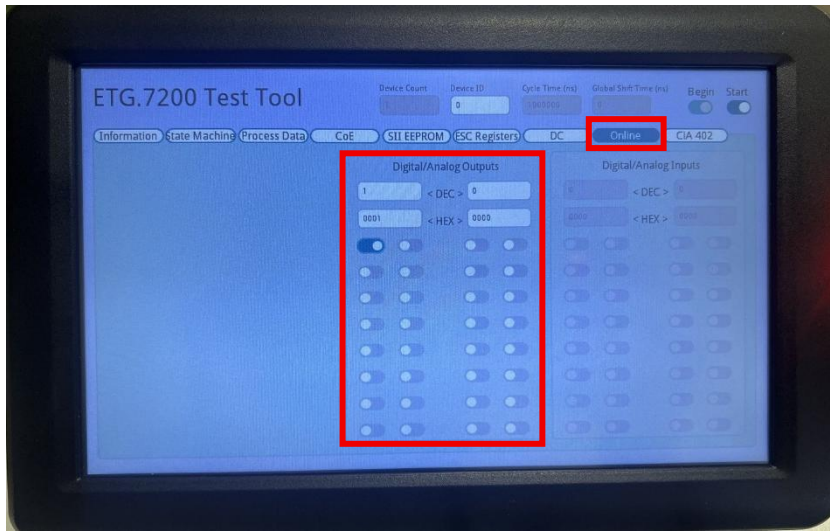


圖 12.2-6 Online 分頁：撥動左側輸出第一位元後，現場 LED 對應亮起。

預期結果：

- Start 啟用後，從站可順利進入 Init、Pre-Op、Safe-Op、Op 四狀態。
- 現場輸出動作與 Online 分頁的操作完全對應；現場輸入變化亦即時反映於畫面。
- AL Status Code 始終為 0x0000，無錯誤。

驗證方式：

可使用三用電錶、邏輯分析儀或現場裝置 (如 LED、繼電器) 做雙向驗證：切換輸出時量測到實際電位變化，觸發輸入時畫面位元同步改變。

常見問題：

- 失敗、回傳 ECAT_ERR_MASTER_NOT_IN_PREOP (-1102) 類錯誤：確認從站目前為 Pre-OP 狀態，且 Start 為 OFF，再重試。
- 切換至 Safe-Op 時失敗：多為 PDO Mapping 與從站實際物件字典不符，回到 Process Data 分頁檢查映射長度與物件位址。
- 位元切換開關操作無效：Start 尚未啟用，或從站未進入 Op 狀態，輸出位元不會生效。

12.3 情境 C — 執行 DC 同步設定

目的：針對支援分散式時鐘 (Distributed Clocks, DC) 的從站，完成 DC 參數設定並啟用 SYNC0/SYNC1，使從站動作與主站時鐘嚴格同步，為後續高精度運動控制 (如 CSP/CSV) 奠定基礎。

前置條件：

- 已完成情境 A 並可識別從站。
- **Begin** 為 ON、**Start** 為 OFF。DC 參數須在 **Start** 關閉狀態下設定。
- 從站支援 DC：於 **Information** 分頁確認，或使用 `isSupportDC()` API 檢查。
- 已決定系統所需的 **Cycle Time** (例如 1 ms、500 μ s、250 μ s)，並確認該值在從站允許範圍內。

操作步驟：

1. 於全域工具列設定 **Cycle Time** 為所需值 (例如 1,000,000 ns 代表 1 ms)。
2. 將 **Global Shift Time** 留為 0，由主站在 **Start** 時自動計算合適的位移量。

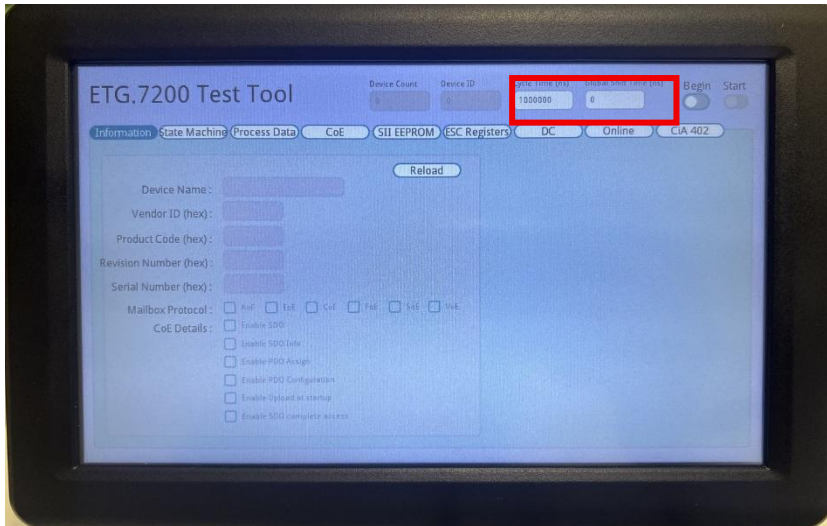


圖 12.3-1 全域工具列：Cycle Time 設為 1,000,000 ns (1 ms)，Global Shift Time 留 0。

3. 切換至 DC 分頁。
4. 於 DC 模式選單選擇 SYNC0、SYNC0+SYNC1 或 Disable，依從站需求而定。一般 CiA 402 伺服建議選 SYNC0。

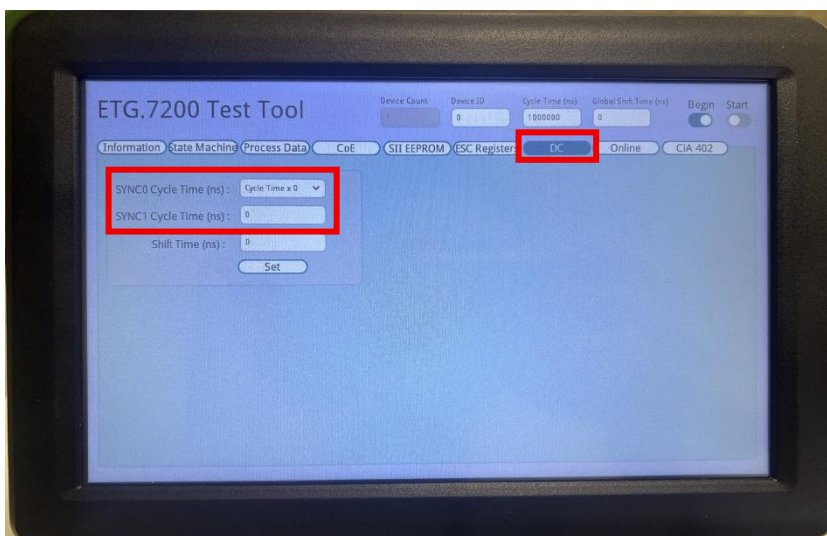


圖 12.3-2 DC 分頁初始畫面：SYNC0 Cycle Time 尚未設定倍數。

- 設定 SYNC0 Cycle Time，通常等於或為 Cycle Time 的整數倍 (例如兩者皆為 1,000,000 ns)。可以點選 SYNC0 的下拉式選單。

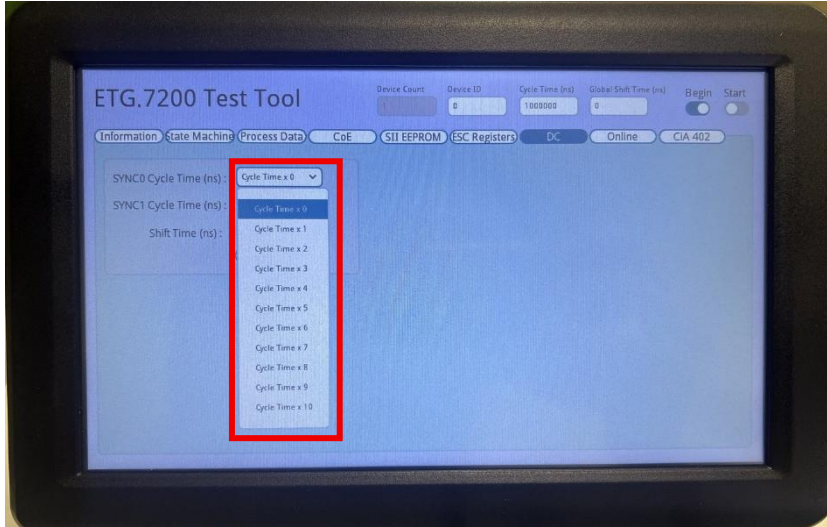


圖 12.3-3 SYNC0 Cycle Time 下拉選單展開 (可選 Cycle Time $\times 0 \sim \times 10$)。

- 點選 **Set** 按鈕，工具會呼叫 `setDc()` 將 DC 設定下達至該從站。成功後，會有 DC 設定成功的視窗出現。

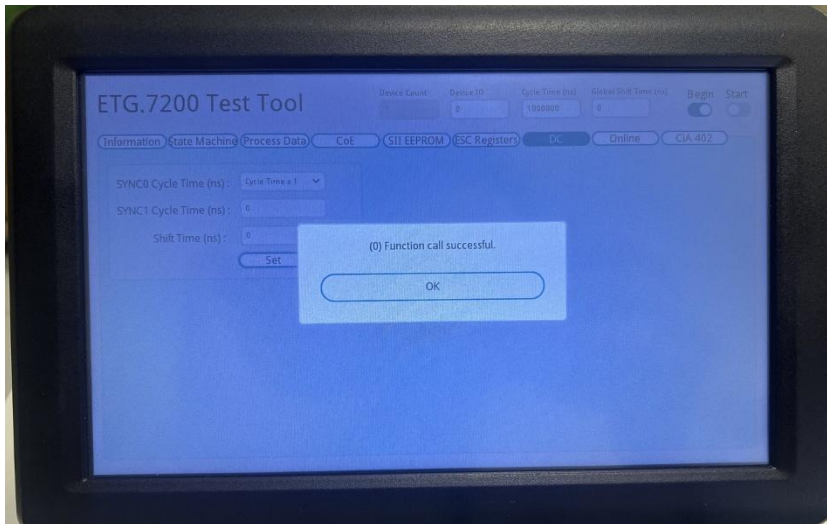


圖 12.3-4 點選 **Set** 後成功回傳 "(0) Function call successful"，代表 `setDc()` 執行完成。

- 若尚有其他支援 DC 的從站，切換 Device ID 後重複上述步驟，確保所有需 DC 同步的從站皆已配置。

8. 完成後，將 **Start** 撥為 **ON**，EtherCAT 將切換至 **Op** 狀態，且 **Global Shift Time** 已由主站在 **Start** 時自動計算合適的位移量。

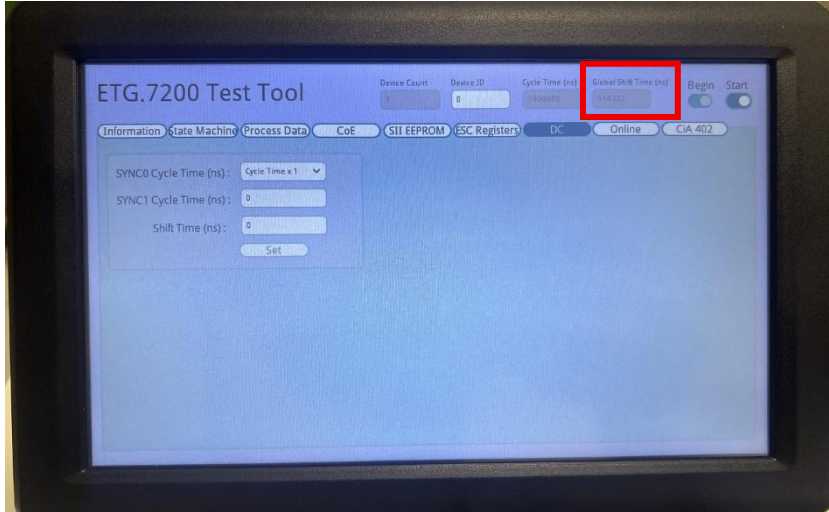


圖 12.3-5 Start 撥 ON 後，主站自動計算並填入合適的 *Global Shift Time*。

9. 回到 **State Machine** 分頁，觀察 **AL Status Code** 應保持 **0x0000**，且 **Current State** 停留於 **Op**。

預期結果：

- 所有需 DC 同步的從站均能順利進入 **Op** 狀態。
- **Working Counter (WKC)** 穩定等於 **Expected Working Counter**，無同步失敗錯誤。
- 若後續結合 **CiA 402** 伺服，運動命令 (**CSP**) 所產生的實際位置軌跡平順、無抖動。

驗證方式：

- 在主站的 **cyclic callback** 中呼叫 `getWorkingCounter()` 與 `getExpectedWorkingCounter()` 進行比較，兩者應持續相等。
- 使用示波器觀察從站 **SYNC0** 輸出腳位 (若有提供)，波形週期應精確等於設定的 **SYNC0 Cycle Time**。
- 切換至 **ESC Registers** 分頁讀取 DC 相關暫存器 (如 **0x0910** 系統時間、**0x098E** **SYNC0 Cycle Time**)，確認實際值與設定值一致。

常見問題：

- 回傳 **ECAT_ERR_DEVICE_NO_DC (-2003)**：該從站不支援 DC，應停用 DC 或改用僅需自由運行的應用。
- 切換至 **Safe-Op** 時即時出現 **AL Status Code** 非零：多為 **SYNC0 Cycle Time** 與主站 **Cycle Time** 不相容，須調整為相同或整數倍。
- 系統時間無法收斂 (**Start** 後 **Device Count** 急速跳動或錯誤回呼不斷觸發)：檢查 **Global Shift Time** 設定，或將其還原為 **0** 讓主站自動計算。

注意 DC 設定會寫入從站的 **Mailbox**，若變更後需重新生效，請先切回 **Pre-OP** 狀態再 **OP**。

12.4 情境 D — PP 模式伺服試運轉

目的：使用 CiA 402 分頁以 Profile Position (PP) 模式對伺服驅動器進行小行程點對點試運轉，驗證驅動器參數、PDO 映射與工具控制流程皆正常。

前置條件：

- 伺服驅動器已依情境 A、B、C 完成識別、PDO 設定與 DC 同步；目前處於 Op 狀態。
- 機台具備緊急停止 (E-Stop) 與軟體限位，並已確認馬達旋轉不會造成機構碰撞。
- 已知驅動器的馬達解析度 (6080h 或 60EFh) 以及一個小行程的安全目標位置 (例如 1 轉以內、換算為 counts)。
- Start 為 ON；Begin 為 ON；DC 為 Cycle time x 1。

操作步驟：

1. 切換至 CiA 402 分頁。

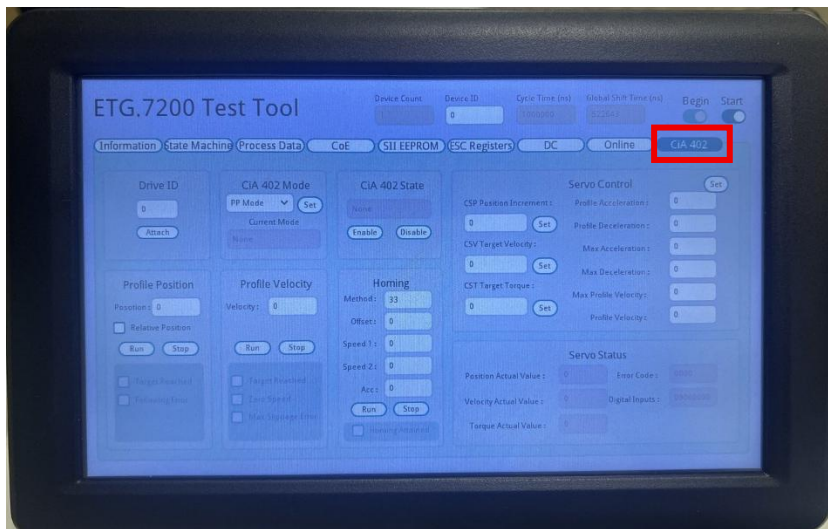


圖 12.4-1 切換至 CiA 402 分頁後的初始畫面。

2. 於 Drive ID 欄位填入對應的軸編號 (通常為 0，除非單一從站含多軸)。
3. 點擊 Attach 按鈕。成功後 按鈕呈現 toggle 狀態，並顯示 Detach。

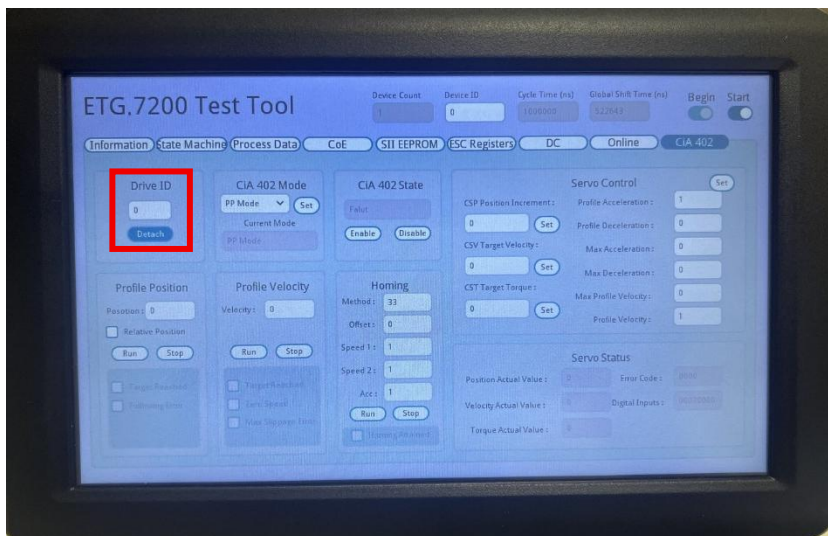


圖 12.4-2 點擊 Attach 後按鈕切換為 Detach。

- 於 CiA 402 Mode 面板選擇 Profile Position (pp)，點擊 Set。

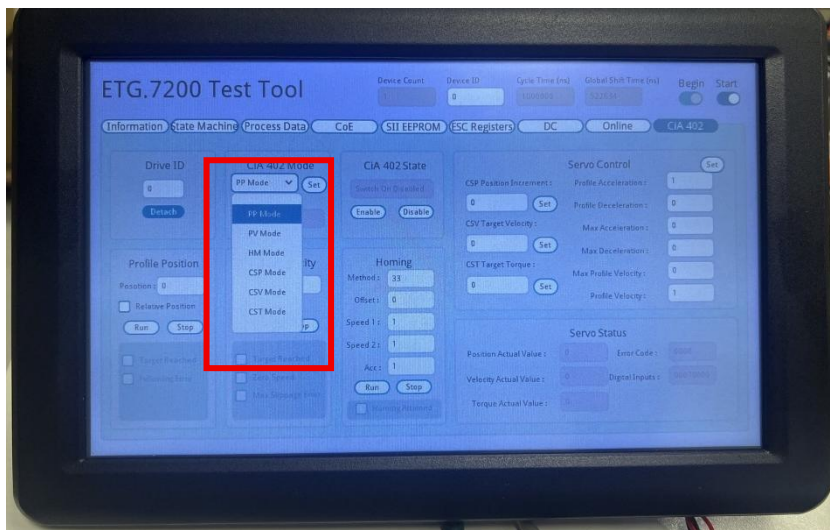


圖 12.4-3 展開 CiA 402 Mode 下拉選單，可選 PP Mode / PV Mode / HM Mode / CSP Mode / CSV Mode / CST Mode 共六種模式。

Current Mode 欄位應顯示 PP Mode。

- 於 CiA 402 State，在設定完成模式後，需點選 Enable 按鈕，將 CiA 402 State 切換至 Operation Enabled 狀態。觀察 Current State 顯示 Operation enabled，Error Code 應為 0。馬達也應該被啟動。

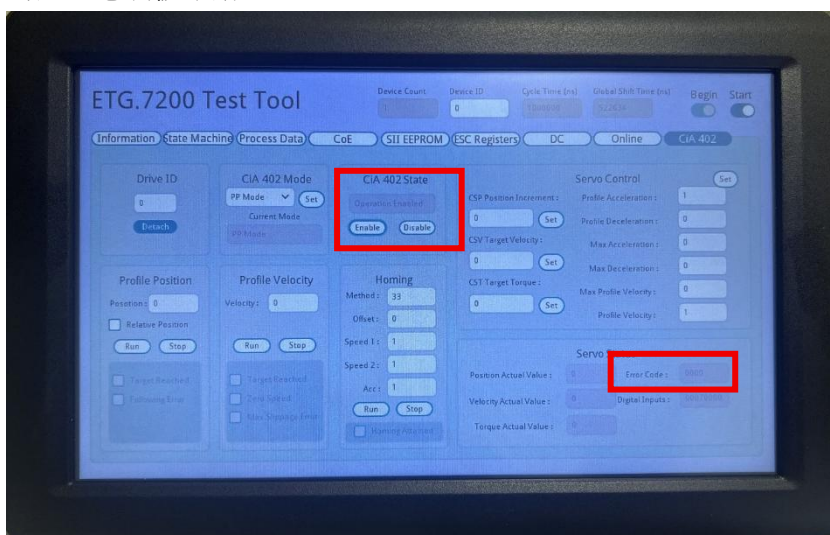


圖 12.4-4 點擊 Enable 按鈕後，CiA 402 State 切換為 Operation Enabled，馬達被啟動。

- 於 Servo Control — 運動參數面板依序設定：Profile Velocity、Profile Acceleration、Profile Deceleration，數值建議先以低速進行試運轉 (例如 profile velocity 設為馬達額定轉速的 10%)。完成後，請按 Set。

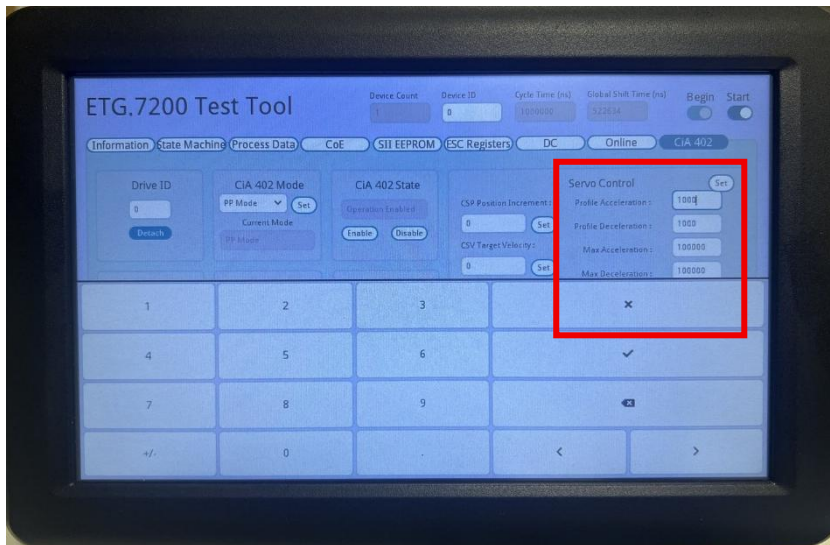


圖 12.4-5 於 Servo Control 輸入運動參數時，HMI 跳出數字鍵盤協助輸入。

- 於 Profile Position 面板輸入 Target Position；建議第一次試運轉以馬達 1 轉以內的 counts 作為目標。
- 點擊 Run 按鈕啟動運動。

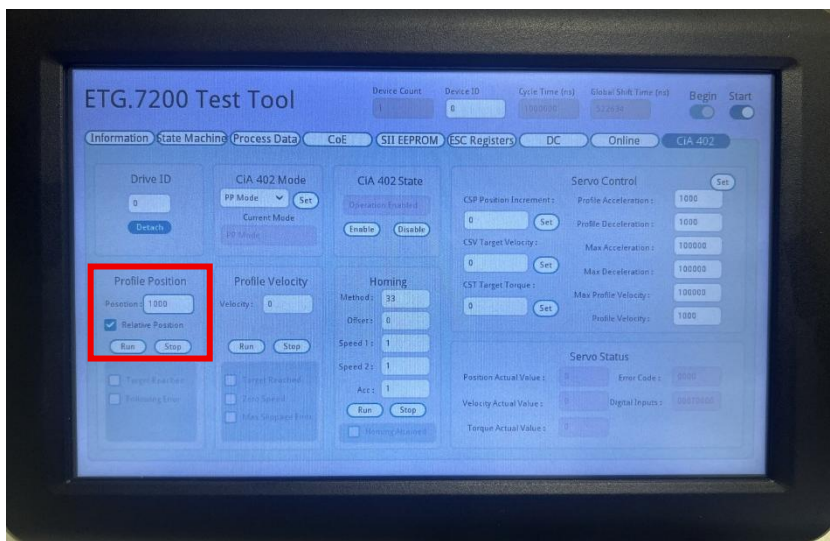


圖 12.4-6 Profile Position 與運動參數設定完成，準備按 Run 啟動運動。

9. 於 Servo Status 區塊觀察 Position Actual Value 持續增加並逼近 Target Position；運動結束時 Target Reached 旗標被設為 1。

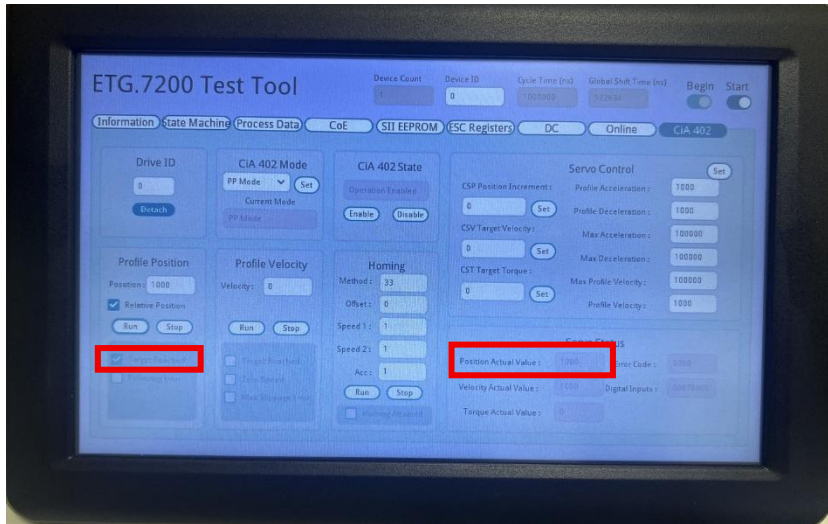


圖 12.4-7 Run 執行完成，Target Reached 勾選，Position Actual Value = 1000。

預期結果：

- 馬達能依設定的加減速曲線平順到達目標位置，且 Target Reached 於 Position Window 誤差範圍內設為 1。
- 運動過程中 Error Code 維持 0、AL Status Code 維持 0x0000。
- Disable 後馬達釋放輸出，可手動轉動。

驗證方式：

- 以第 13 章 `getPositionActualValue()` 與 `pp_IsTargetReached()` 在 cyclic callback 內紀錄實際軌跡，確認到達目標。
- 以外部編碼器或位置感測器比對實際位移與 Target Position 差異。

常見問題：

- 失敗並回傳 ECAT_ERR_DEVICE_CIA402_ADD_FAIL (-2501)：Drive ID 與驅動器韌體內軸編號不符，請核對驅動器設定。
- Enable 後 Current State 停在 Switched On 或出現 Fault：讀取 Error Code (603Fh)，依驅動器錯誤表處理 (常見為激磁異常、限位觸發等)。
- Run 後位置未動、Target Reached 始終為 0：確認 Profile Velocity 與 Profile Acceleration 非 0，以及 Target Position 與目前位置差距足夠大。
- 位置跳變或失速：檢查 PDO 映射是否包含 6040h/607Ah/6041h/6064h，或改以 CSP 模式搭配 Shift Time (情境 C) 驗證。

注意 首次試運轉建議將目標位置限制在一轉以內，並準備好隨時按下 E-Stop 或點擊 Disable，以防參數設定錯誤造成意外。

12.5 情境 E — CSP 模式循環同步運轉

目的：使用 CiA 402 分頁的循環同步位置 (Cyclic Synchronous Position, CSP) 模式，由主站每個 EtherCAT 週期向驅動器寫入新的目標位置。本情境驗證 CSP 通訊鏈、CspTargetPosition 行為以及 CSP Position Increment 控制。

前置條件：

- 伺服驅動器已依情境 A、B 完成識別與 PDO 設定；PDO 映射至少包含 6040h、6041h、6064h、607Ah。
- 已依情境 C 完成 DC 同步；CSP 模式對 DC 抖動敏感，沒有 DC 通常會造成位置不平順。
- 驅動器支援 CSP：可用 `getSupportedDriveModes()` 確認，或於 CiA 402 Mode 下拉選單確認有 CSP 選項。
- 機台具備緊急停止 (E-Stop)；CSP 模式啟用後，驅動器會即時跟隨目標位置，動作可能比 PP 更快。
- Begin 為 ON、Start 為 ON、DC 為 Cycle time x 1，從站處於 Op 狀態。

操作步驟：

1. 切換至 CiA 402 分頁。
2. 於 Drive ID 欄位填入軸編號 (通常為 0)，點擊 Attach。成功後按鈕切換為 Detach。
3. 於 CiA 402 Mode 面板選擇 CSP Mode，點擊 Set。Current Mode 應更新為 CSP Mode。

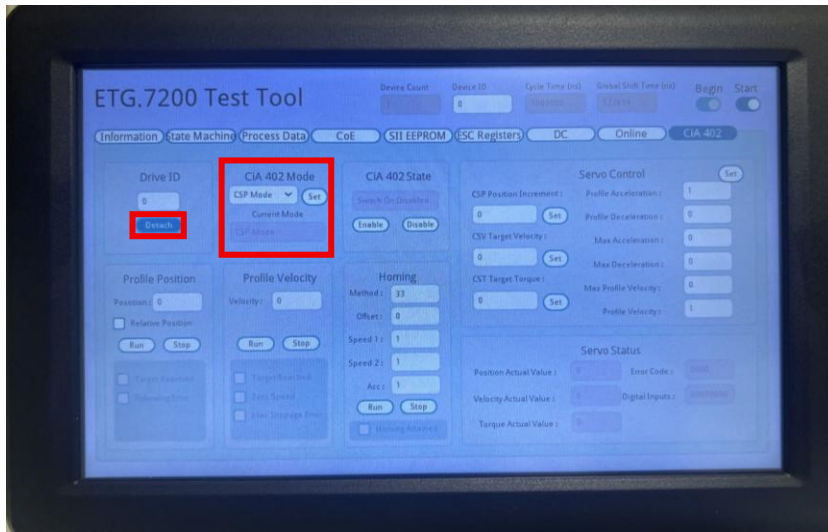


圖 12.5-2 CSP Mode 已 Attach，CiA 402 State 顯示 Switch On Disabled，等待 Enable。

- 於 CiA 402 State 面板點擊 Enable，將驅動器切換至 Operation Enabled。此時 CSP Position Increment 為 0，馬達尚未動作但已啟動。CSP 主由主站逐週期提供位置，Increment 為根據當前位置 + 或 - 數值。

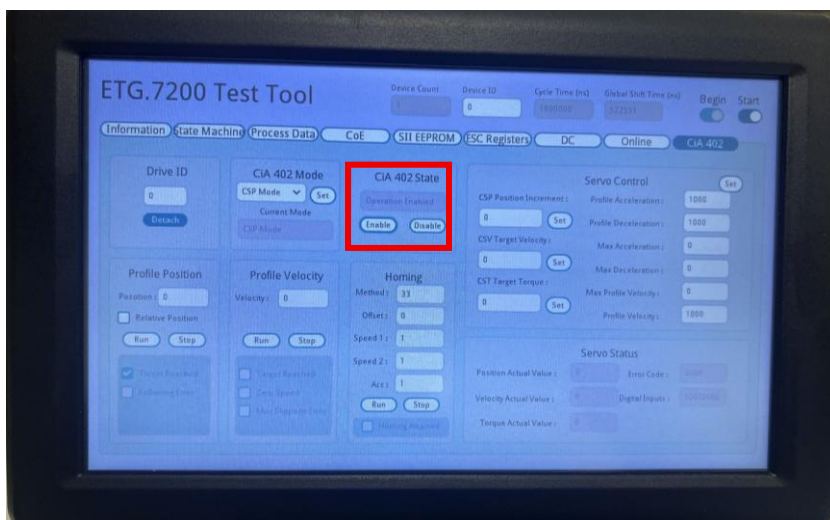


圖 12.5-4 點擊 Enable 後 CiA 402 State 顯示 Operation Enabled；Increment 仍為 0，馬達靜止待命。

- 點選 CSP Position Increment 欄位，HMI 跳出數字鍵盤；輸入每個週期的位置增量 (正值前進、負值後退；本範例為 1 count/cycle)，按 ✓ 確認。

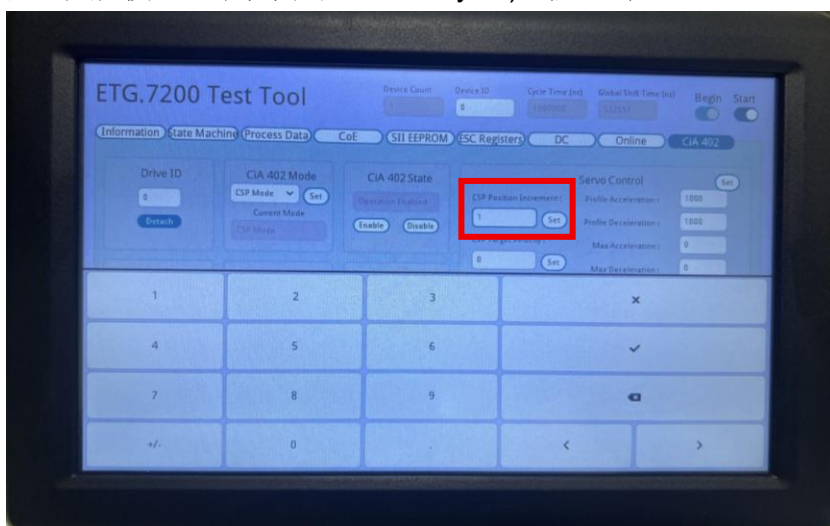


圖 12.5-5 點選 CSP Position Increment 欄位後跳出的數字鍵盤畫面。

6. 按下 Increment 欄位旁的 Set，主站每週期會將目標位置往前累加 Increment counts。馬達開始連續運轉，速率約等於 $\text{Increment} \times (1 / \text{Cycle Time})$ 。

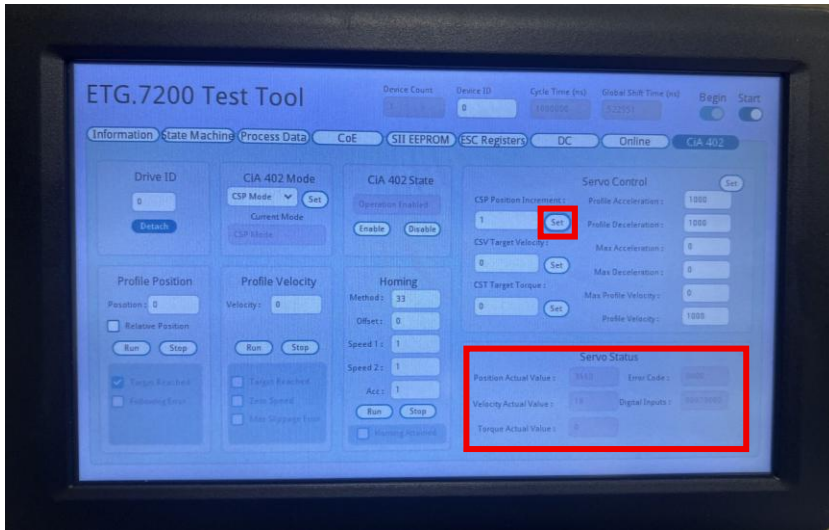


圖 12.5-6 CSP 運轉中：Increment = 1，Position Actual Value 累加至 3553。

7. 若需停止，將 CSP Position Increment 改為 0 並按 Set，目標位置會凍結在最後寫入值；或直接點擊 Disable 進入 Switch On Disabled，馬達依驅動器設定的 Disable Operation Option 動作（通常為自然停止）。

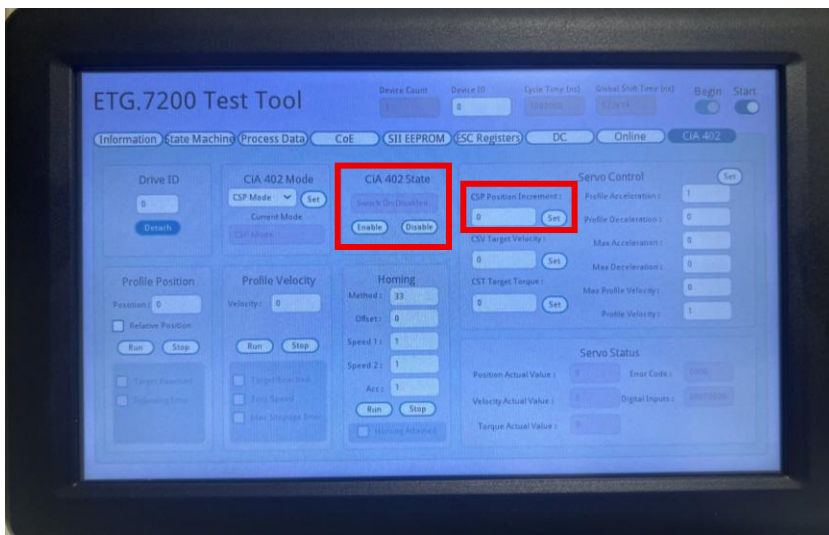


圖 12.5-7 若需停止，將 CSP Position Increment 改為 0 並按 Set；或直接點擊 Disable 進入 Switch On Disabled

預期結果：

- 馬達依 CSP Position Increment 等速連續運轉。
- Position Actual Value 與 Position Demand Value 之差 (即 Following Error) 維持在驅動器設定的 Following Error Window 內。
- Error Code 維持 0、AL Status Code 維持 0x0000，Working Counter 等於 Expected Working Counter。
- Disable 後馬達依設定停止；再次 Enable 不應跳變 (因為 Attach 時 CspTargetPosition 已鎖存為當下實際位置)。

驗證方式：

- 以示波器或外部編碼器觀察馬達實際位移，驗證速率穩定 ($= \text{Increment} \div \text{Cycle Time}$)。
- 改變 CSP Position Increment 數值或正負號，觀察 Position Actual Value 即時跟隨變化。
- 使用 `getFollowingErrorActualValue()` 確認 Following Error 在合理範圍。

常見問題：

- **Enable** 後馬達瞬間跳變或產生過大 **Following Error**：通常是 **CspTargetPosition** 與目前實際位置不一致；本工具於 **Attach** 時會自動將 **Position Actual Value** 鎖存為 **CspTargetPosition**，若仍跳變請先 **Disable** 後重新 **Attach**。
- **Position Actual Value** 不動：確認 **CSP Position Increment** 已按 **Set** 寫入、**State** 為 **Operation Enabled**、且驅動器有實際激磁；亦可檢查 **Following Error Window** 是否過小導致驅動器自動切回 **Fault**。
- 速率與計算值不符：**Cycle Time** 設定可能不正確，或 **PDO** 並未每週期更新；改用 **ESC Registers 0x0910** 系統時間驗證主站週期。
- **Enable** 後出現 **Fault (Error Code 非 0)**：常見原因為超過軟體限位、**Following Error** 過大、或缺少 **DC** 同步造成插補失敗。讀取 **Error Code (603Fh)** 對照驅動器錯誤表，並確認情境 **C** 的 **DC** 同步已完成。

注意 **CSP** 模式為連續運轉，啟用後馬達即開始動作，請務必確認機台行程與 **E-Stop** 安全。建議先以較小的 **Increment** (例如 1-20 counts) 試運轉，再逐步增加。

12.6 情境 F — 物件字典瀏覽與 SDO 讀寫

目的：透過 CoE 分頁進行 SDO Upload (讀取) 與 SDO Download (寫入) 來存取從站的物件字典；以 OD Generation 一次匯入完整物件字典結構供瀏覽；觀察 Emergency Message 排查故障。本情境涵蓋驅動器調機、PDO 預設、錯誤排查時最常用的非週期性資料存取方法。

前置條件：

- 已完成情境 A，可透過 Information 分頁讀取從站識別資訊。
- 從站處於 Pre-Op 或更高狀態 — SDO 通訊需要 Mailbox 啟用，Init 狀態無法存取。
- 於 Information 分頁確認 Mailbox Protocol 勾選 CoE；若需 OD Generation，CoE Details 須勾選 Enable SDO Info。
- 已知欲存取的 Index / SubIndex；可由廠商規格書、ESI 檔取得，或先用 OD Generation 自動產生。

操作步驟：

(A) 瀏覽完整物件字典

1. CoE 分頁初次切入時，Object Dictionary 表格與 Emergency Message 區塊均為空白。

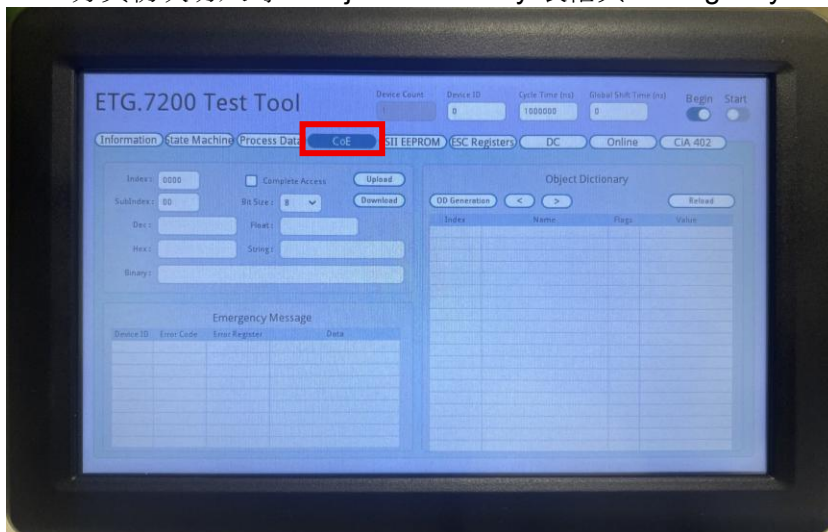


圖 12.6-1 CoE 分頁初始畫面：Object Dictionary 與 Emergency Message 均為空。

2. 點擊右上 OD Generation 按鈕。工具會依序呼叫 `getODlist()`、`getObjectDescription()`、`getEntryDescription()` 讀取從站完整字典結構。視物件數量，可能費時 1–10 秒。

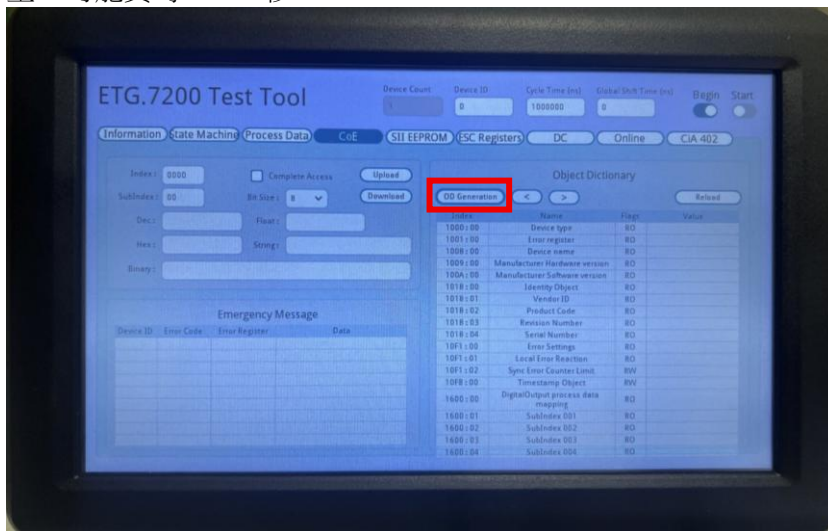


圖 12.6-2 OD Generation 完成後，Object Dictionary 列出全部 Index / Name / Flags，但 Value 欄仍空白。

3. 點擊 **Reload** 按鈕。工具會逐筆對每個物件發送 SDO Upload 並回填 Value 欄。
4. 滾動或用 </> 按鈕翻頁，瀏覽 Index、Name、Flags (RO/RW)、Value 四欄；常見物件如 1000h 裝置型別、1018h Identity Object、6064h Position Actual Value 等都可在查看。

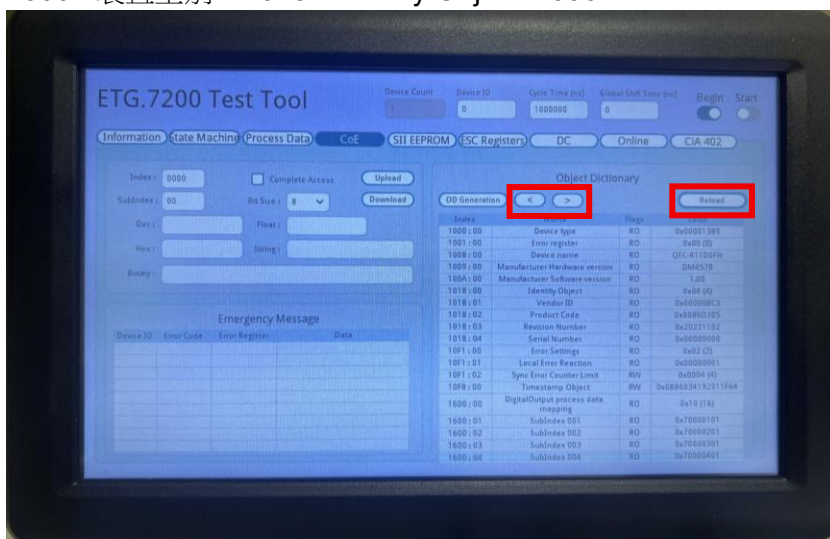


圖 12.6-3 Reload 完成後 Object Dictionary 完整顯示各物件目前值，可直接瀏覽 RO/RW 旗標與內容。

(B) 讀取單筆物件 (SDO Upload)

1. 於左上 SDO 操作面板，填入 Index (例如 6041h，CiA 402 Statusword)、SubIndex。

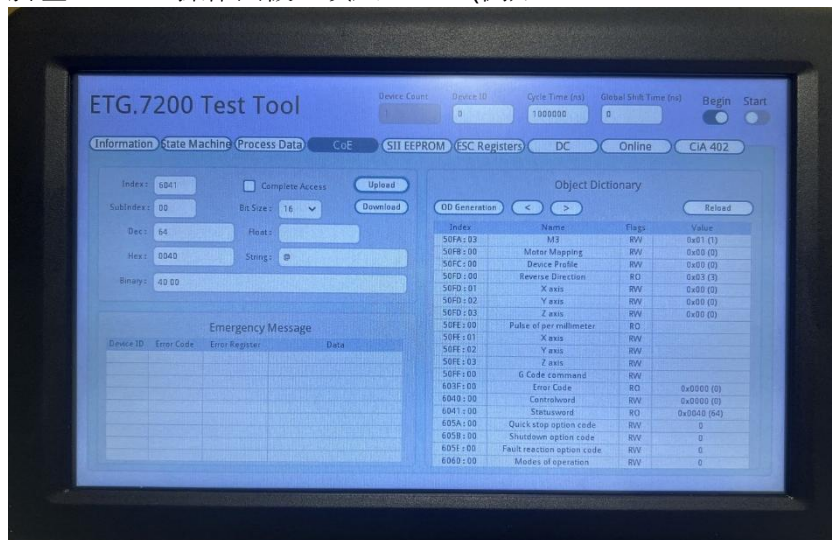


圖 12.6-4 SDO Upload 範例：輸入讀取物件。

2. 設定 Bit Size — 從下拉選單選擇 8 / 16 / 32 / 64 (此例 16)。若需一次讀取整個複合物件，可勾選 Complete Access。
3. 點擊 Upload 按鈕。工具呼叫 `sdoUpload()` 取回該物件目前值。

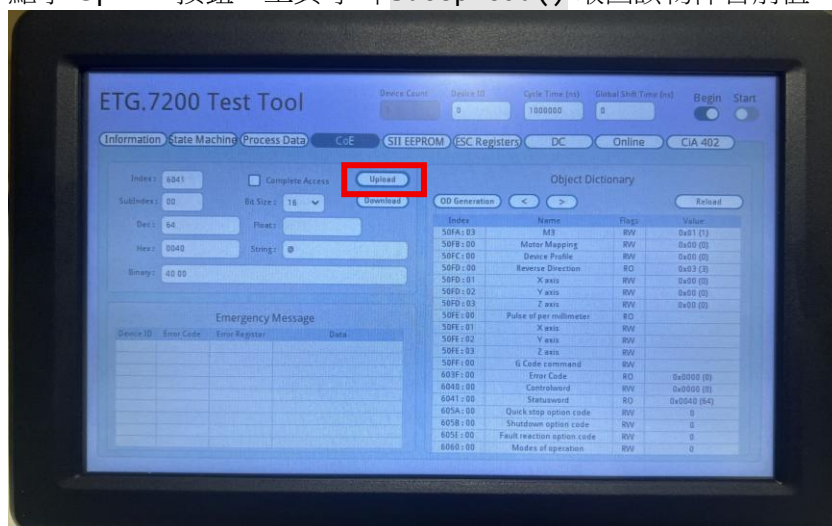


圖 12.6-5 SDO Upload 範例：讀取 6041h Statusword，Dec=64、Hex=0040、Binary=40 00；右側 Object Dictionary 同步顯示 6041:00 為 0x0040 (64)。

4. 結果會同時顯示為 Dec、Hex、Float、String、Binary 五種格式，可依需求對照。

(C) 寫入單筆物件 (SDO Download)

1. 同樣填入 Index / SubIndex / Bit Size。常見可寫物件如 6065h Following Error Window、6080h Max Motor Speed、605Ah Quick Stop Option Code 等。

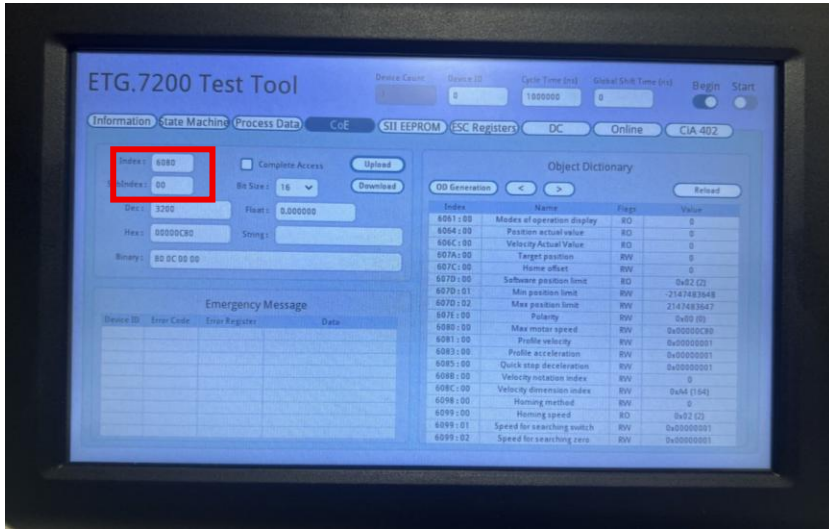


圖 12.6-6 寫入前 Upload 6080h 取得舊值：Hex=00000C80、Dec=3200。

2. 點選 Dec 或 Hex 欄位，HMI 跳出數字鍵盤輸入新值。

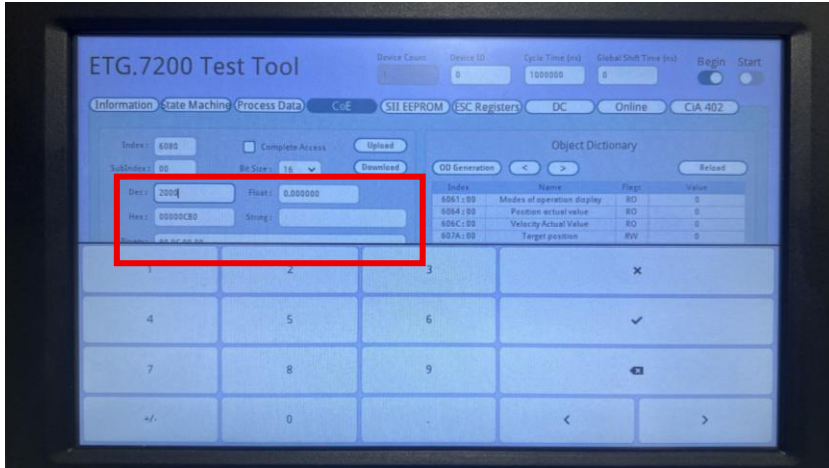


圖 12.6-7 點選 Dec 欄位後跳出鍵盤，正在輸入新值 2000

3. 點擊 Download 按鈕，工具呼叫 sdoDownload() 寫入；跳出狀態視窗。

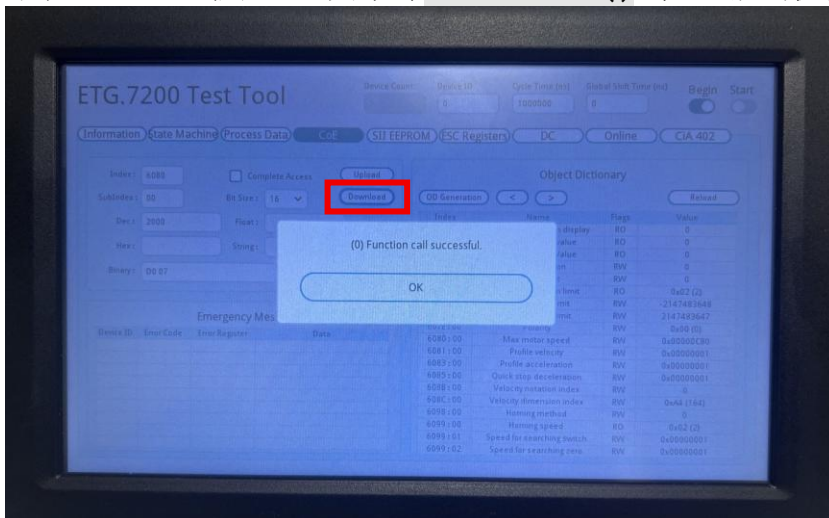


圖 12.6-8 Download 成功的對話框「(0) Function call successful」

4. 立即再點 Upload 驗證寫入，Hex 欄位現在應顯示 00007D0 (= 2000)，同時 Object Dictionary 內 6080:00 的值也更新為 0x00007D0。

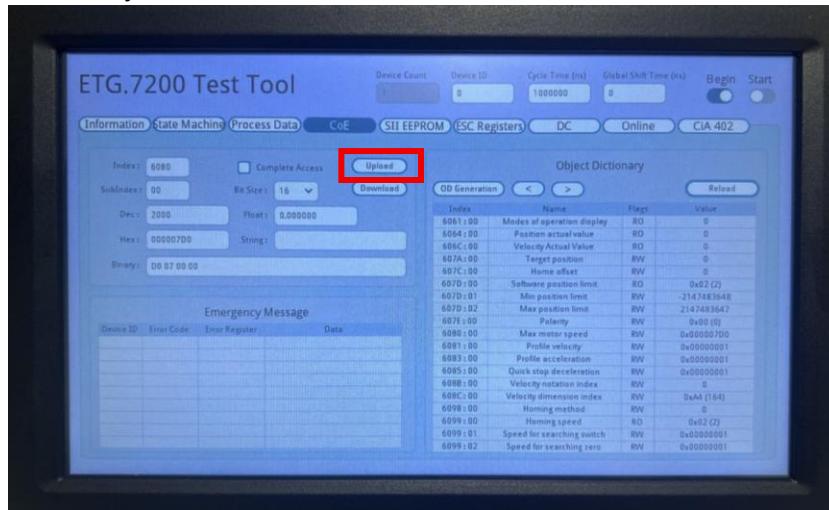


圖 12.6-9 寫入後再次 Upload，Hex = 00007D0 = 2000；右側 Object Dictionary 中 6080:00 同步更新。

- 寫入後 Upload 顯示舊值：寫入需要存入 NVM 才能跨重啟保留 — 透過 SDO Download 寫 1010h:01h = "save" (0x65766173) 觸發儲存；某些物件需在特定狀態 (例如 Pre-Op、非 Operation Enabled) 才允許寫入。
- Reload 卡住或耗時極久：物件數量過多 + 從站回應慢的組合。可改為手動 Upload 你關心的物件，或縮小單次 Bit Size。

注意 寫入物件字典等於即時改變從站行為，誤改 PDO 映射 (1600h/1A00h)、Cycle Time、Following Error Window 等關鍵物件可能讓馬達失控或失去通訊。建議先 Upload 備份原值，並避免在馬達 Operation Enabled 時做 Download。

13. EtherCAT API 參考

13.1 EtherCAT API

Functions:

Function Name	Description	Callback Available
Initialization-related functions		
begin()	Initialize the EtherCAT MDevice.	
end()	Deinitialize the EtherCAT MDevice.	
isRedundancy()	Check if the EtherCAT MDevice has cable redundancy enabled.	O
libraryVersion()	Get the EtherCAT MDevice library version.	O
firmwareVersion()	Get the EtherCAT firmware version.	O
readSettings()	Read the current EtherCAT MDevice settings.	
saveSettings()	Save the EtherCAT MDevice settings.	
Control-related functions		
start()	Start the EtherCAT MDevice.	
stop()	Stop the EtherCAT MDevice.	
update()	Update process data and handle acyclic commands.	O
setShiftTime()	Set the Global Shift Time for DC-Synchronous mode.	
getShiftTime()	Get the Global Shift Time for DC-Synchronous mode.	O
getSystemTime()	Get the system time of the current cycle.	O
getWorkingCounter()	Get the working counter for the current cycle.	O
getExpectedWorkingCounter()	Get the expected working counter.	O
Callback-related functions		
attachCyclicCallback()	Register a cyclic callback.	
detachCyclicCallback()	Unregister cyclic callback.	
attachErrorCallback()	Register an error callback.	
detachErrorCallback()	Unregister error callback.	
attachEventCallback()	Register an event callback.	
detachEventCallback()	Unregister event callback.	
errGetCableBrokenLocation1()	Get the cable broken location 1 in error callback.	O ¹
errGetCableBrokenLocation2()	Get the cable broken location 2 in error callback.	O ¹
evtGetMasterState()	Get the EtherCAT MDevice state in event callback.	O ²
SubDevice information related functions		
getSlaveCount()	Get the number of SubDevices on the network.	O
getVendorID()	Get the vendor ID of the specified device.	O
getProductCode()	Get the product code of the specified device.	O
getRevisionNumber()	Get the revision number of the specified device.	O
getSerialNumber()	Get the serial number of the specified device.	O
getAliasAddress()	Get the alias address of the specified device.	O
getSlaveNo()	Find the sequence number of the matching EtherCAT SubDevice on the network.	O

- **Note 1:** This function can only be called in error callback.
- **Note 2:** This function can only be called in event callback.

13.2 SDO / PDO / ESC / SII / FoE API

Functions:

Function Name	Description	Callback Available
SubDevice information related functions		
getVendorID()	Get the vendor ID.	O
getProductCode()	Get the product code.	O
getRevisionNumber()	Get the revision number.	O
getSerialNumber()	Get the serial number.	O
getAliasAddress()	Get the alias address.	O
getSlaveNo()	Get the sequence ID on the EtherCAT network.	O
getDeviceName()	Get the device name.	O
getMailboxProtocol()	Get the supported mailbox protocol types.	O
getCoEDetails()	Get the details about CoE supported.	O
getFoEDetails()	Get the details about FoE supported.	O
getEoEDetails()	Get the details about EoE supported.	O
getSoEChannels()	Get the number of SoE channels supported.	O
isSupportDC()	Check if the EtherCAT SubDevice has DC supported.	O
PDO access functions		
pdoBitWrite()	Write 1-bit output process data.	O
pdoBitRead()	Read 1-bit input process data.	O
pdoGetOutputBuffer()	Get the memory pointer of output process data.	O
pdoGetInputBuffer()	Get the memory pointer of input process data.	O
pdoWrite()	Write multiple bytes of output process data.	O
pdoWrite8()	Write 8-bit output process data.	O
pdoWrite16()	Write 16-bit output process data.	O
pdoWrite32()	Write 32-bit output process data.	O
pdoWrite64()	Write 64-bit output process data.	O
pdoRead()	Read multiple bytes of input process data.	O
pdoRead8()	Read 8-bit input process data.	O
pdoRead16()	Read 16-bit input process data.	O
pdoRead32()	Read 32-bit input process data.	O
pdoRead64()	Read 64-bit input process data.	O
CoE communication functions		
sdoDownload()	Write multiple bytes of data to the object.	
sdoDownload8()	Write 8-bit value to the object.	
sdoDownload16()	Write 16-bit value to the object.	
sdoDownload32()	Write 32-bit value to the object.	
sdoDownload64()	Write 64-bit value to the object.	
sdoUpload()	Read multiple bytes of data from the object.	
sdoUpload8()	Read 8-bit value from the object.	
sdoUpload16()	Read 16-bit value from the object.	
sdoUpload32()	Read 32-bit value from the object.	
sdoUpload64()	Read 64-bit value from the object.	
getODlist()	Get a list of objects existing in the object dictionary.	
getObjectDescription()	Get the object description of the object.	
getEntryDescription()	Get the entry description of the object.	
FoE communication functions		
readFoE()	Read a file from the EtherCAT SubDevice.	
writeFoE()	Write a file to the EtherCAT SubDevice.	
DC configuration functions		
setDc()	Configure DC parameters.	
SII EEPROM access functions		
writeSII()	Write multiple bytes of data to the SII EEPROM.	
writeSII8()	Write 8-bit value to the SII EEPROM.	
writeSII16()	Write 16-bit value to the SII EEPROM.	
writeSII32()	Write 32-bit value to the SII EEPROM.	
readSII()	Read multiple bytes of data from the SII EEPROM.	

readSII8()	Read 8-bit value from the SII EEPROM.	
readSII16()	Read 16-bit value from the SII EEPROM.	
readSII32()	Read 32-bit value from the SII EEPROM.	

13.3 CiA 402 API

Functions:

Function Name	Description	Callback Available
Initialization-related functions		
attach()	Initialize the object of this EtherCAT SubDevice class.	
detach()	Deinitialize the object of this EtherCAT SubDevice class.	
isStepper()	Check if the EtherCAT SubDevice is a stepper motor drive.	O
Control-related functions		
getCiA402Mode()	Get the current mode of operation. (6061 _h)	O ¹
setCiA402Mode()	Switch the mode of operation. (6060 _h , 6061 _h , 6502 _h)	O ^{1,2}
getCiA402State()	Get the current CiA 402 state. (6041 _h)	O
setCiA402State()	Switch the CiA 402 state. (6040 _h , 6041 _h)	O ²
enable()	Enable the drive function and power on the motor. (6040 _h , 6041 _h)	O ²
disable()	Disable the drive function and power off the motor. (6040 _h , 6041 _h)	O ²
Operation-related functions		
setTargetPosition()	Set the target position. (607A _h)	O ¹
setTargetVelocity()	Set the target velocity. (60FF _h)	O ¹
setTargetTorque()	Set the target torque. (6071 _h)	O ¹
setProfileAcceleration()	Set the profile acceleration. (6083 _h)	O ¹
setProfileDeceleration()	Set the profile deceleration. (6084 _h)	O ¹
setMaxAcceleration()	Set the max acceleration. (60C5 _h)	O ¹
setMaxDeceleration()	Set the max deceleration. (60C6 _h)	O ¹
setMaxProfileVelocity()	Set the max profile velocity. (607F _h)	O ¹
setMotionProfileType()	Set the motion profile type. (6086 _h)	O ¹
setPositionWindow()	Set the position window. (6067 _h)	O ¹
setPositionWindowTime()	Set the position window time. (6068 _h)	O ¹
setPositionOffset()	Set the position offset. (60B0 _h)	O ¹
setSoftwarePositionLimit()	Set the software position limit. (607D _h)	O ¹
setFollowingErrorWindow()	Set the following error window. (6065 _h)	O ¹
setPositionPolarity()	Set the position polarity. (607E _h)	O ¹
setVelocityWindow()	Set the velocity window. (606D _h)	O ¹
setVelocityWindowTime()	Set the velocity window time. (606E _h)	O ¹
setVelocityThreshold()	Set the velocity threshold. (606F _h)	O ¹
setVelocityOffset()	Set the velocity offset. (60B1 _h)	O ¹
setMaxMotorSpeed()	Set the max motor speed. (6080 _h)	O ¹
setVelocityPolarity()	Set the velocity polarity. (607E _h)	O ¹
setTorqueOffset()	Set the torque offset. (60B2 _h)	O ¹
setMaxTorque()	Set the max torque. (6072 _h)	O ¹
setPositiveTorqueLimit()	Set the positive torque limit. (60E0 _h)	O ¹
setNegativeTorqueLimit()	Set the negative torque limit. (60E1 _h)	O ¹
setQuickStopDeceleration()	Set the quick stop deceleration. (6085 _h)	O ¹
setQuickStopOptionCode()	Set the quick stop option code. (605A _h)	
setShutdownOptionCode()	Set the shutdown option code. (605B _h)	
setDisableOperationOptionCode()	Set the disable operation option code. (605C _h)	
setHaltOptionCode()	Set the halt option code. (605D _h)	
setFaultReactionOptionCode()	Set the fault reaction option code. (605E _h)	
getErrorCode()	Get the error code. (603F _h)	O ¹
getSupportedDriveModes()	Get the supported drive modes. (6502 _h)	O ¹
getMotorResolution()	Get the motor resolution. (60EF _h)	
getPositionActualValue()	Get the position actual value. (6064 _h)	O ¹
getVelocityActualValue()	Get the velocity actual value. (606C _h)	O ¹

getTorqueActualValue()	Get the torque actual value. (6077 _h)	O ¹
getCurrentActualValue()	Get the current actual value. (6078 _h)	O ¹
getPositionDemandValue()	Get the position demand value. (6062 _h)	O ¹
getPositionDemandInternalValue()	Get the position demand internal value. (60FC _h)	O ¹
getPositionActualInternalValue()	Get the position actual internal value. (6063 _h)	O ¹
getAdditionalPositionActualValue()	Get the additional position actual value. (60E4 _h)	O ¹
getFollowingErrorActualValue()	Get the following error actual value. (60F4 _h)	O ¹
getVelocityDemandValue()	Get the velocity demand value. (606B _h)	O ¹
getTorqueDemandValue()	Get the torque demand value. (6074 _h)	O ¹
getDigitalInputs()	Get the digital inputs. (60FD _h)	O ¹
Profile Position mode (pp) related functions		
pp_SetVelocity()	Set the profile velocity. (6081 _h)	
pp_SetAcceleration()	Set the profile acceleration. (6083 _h)	
pp_SetDeceleration()	Set the profile deceleration. (6084 _h)	
pp_SetMotionProfileType()	Set the motion profile type. (6086 _h)	
pp_Run()	Move to the target position. (6040 _h , 6041 _h , 607A _h)	
pp_IsTargetReached()	Check if the target position has been reached. (6041 _h)	O
pp_CheckFollowingError()	Check if the following error occurs. (6041 _h)	O
pp_Halt()	Pause the current operation. (6040 _h , 6041 _h)	
pp_Resume()	Resume the paused operation. (6040 _h , 6041 _h)	
Profile Velocity mode (pv) related functions		
pv_SetAcceleration()	Set the profile acceleration. (6083 _h)	
pv_SetDeceleration()	Set the profile deceleration. (6084 _h)	
pv_SetMotionProfileType()	Set the motion profile type. (6086 _h)	
pv_Run()	Move at a target velocity continuously. (6041 _h , 60FF _h)	
pv_IsTargetReached()	Check if the target velocity has been reached. (6041 _h)	O
pv_CheckZeroSpeed()	Check if the speed is zero. (6041 _h)	O
pv_CheckMaxSlippageError()	Check if the maximum slippage error occurs. (6041 _h)	O
pv_Halt()	Pause the current operation. (6040 _h , 6041 _h)	
pv_Resume()	Resume the paused operation. (6040 _h , 6041 _h)	
Profile Torque mode (tq) related functions		
tq_SetTorqueSlope()	Set the torque slope. (6087 _h)	
tq_SetTorqueProfileType()	Set the torque profile type. (6088 _h)	
tq_SetMotorRatedCurrent()	Set the motor rated current. (6075 _h)	
tq_SetMotorRatedTorque()	Set the motor rated torque. (6076 _h)	
tq_Run()	Drive continuously at the target torque. (6041 _h , 6071 _h)	
tq_IsTargetReached()	Check if the target torque has been reached. (6041 _h)	O
tq_Halt()	Pause the current operation. (6040 _h , 6041 _h)	
tq_Resume()	Resume the paused operation. (6040 _h , 6041 _h)	
Homing mode (hm) related functions		
hm_SetHomeOffset()	Set the home offset. (607C _h)	
hm_SetHomingMethod()	Set the homing method. (6098 _h)	
hm_SetHomingSpeeds()	Set the homing speeds. (6099 _h)	
hm_SetHomingAcceleration()	Set the homing acceleration. (609A _h)	
hm_Run()	Initiate a homing operation. (6040 _h , 6041 _h)	
hm_IsAttained()	Check the status of the homing operation. (6041 _h)	O
hm_Stop()	Stop the homing operation. (6040 _h , 6041 _h)	
Function Group "Touch Probe" related functions		
enableTouchProbe1()	Enable the touch probe 1. (60B8 _h , 60B9 _h , 60D0 _h)	
enableTouchProbe2()	Enable the touch probe 2. (60B8 _h , 60B9 _h , 60D0 _h)	
disableTouchProbe1()	Disable the touch probe 1. (60B8 _h , 60B9 _h)	
disableTouchProbe2()	Disable the touch probe 2. (60B8 _h , 60B9 _h)	
isTouchProbe1ValueReady()	Check if a positive or negative edge has occurred on the touch probe 1 signal. (60B9 _h)	O ¹

isTouchProbe2ValueReady()	Check if a positive or negative edge has occurred on the touch probe 2 signal. (60B9 _h)	O ¹
readTouchProbe1Value()	Read the touch probe position 1. (60BA _h , 60BB _h)	O ¹
readTouchProbe2Value()	Read the touch probe position 2. (60BC _h , 60BD _h)	O ¹
Low-level functions for mode-specific flow control		
setHaltBit()	Set the halt bit in the controlword. (6040 _h)	O
isTargetReached()	Check if the target has reached. (6041 _h)	O
setModeSpecificBit4()	Set the bit 4 in the controlword. (6040 _h)	O
setModeSpecificBit5()	Set the bit 5 in the controlword. (6040 _h)	O
setModeSpecificBit6()	Set the bit 6 in the controlword. (6040 _h)	O
checkModeSpecificBit12()	Check the value of bit 12 in the statusword. (6041 _h)	O
checkModeSpecificBit13()	Check the value of bit 13 in the statusword. (6041 _h)	O

- **Note 1:** This function can be used in callback functions if the related object is mapped to PDO.
- **Note 2:** This function will ignore the timeout parameter and will not wait for the actual value to match the set value when used in a callback.

14. 錯誤代碼對照表

14.1 EtherCAT API 錯誤碼

對於大多數函數而言，傳回值小於零表示發生錯誤，該值代表一個錯誤代碼。

如果存在錯誤代碼，您可以在下方找到錯誤原因和相應的糾正措施。

Definition	Code
ECAT_SUCCESS	0
ECAT_ERR_MODULE_INIT_FAIL	-100
ECAT_ERR_MODULE_GET_VERSION_FAIL	-101
ECAT_ERR_MODULE_VERSION_MISMATCH	-102
ECAT_ERR_MODULE_GENERIC_TRANSFER_INIT_FAIL	-103
ECAT_ERR_MASTER_DOWNLOAD_SETTINGS_FAIL	-200
ECAT_ERR_MASTER_SET_DEVICE_SETTINGS_FAIL	-201
ECAT_ERR_MASTER_GET_GROUP_INFO_FAIL	-202
ECAT_ERR_MASTER_GET_MASTER_INFO_FAIL	-203
ECAT_ERR_MASTER_GET_DEVICE_INFO_FAIL	-204
ECAT_ERR_MASTER_SET_GROUP_SETTINGS_FAIL	-205
ECAT_ERR_MASTER_MAPPING_INIT_FAIL	-206
ECAT_ERR_MASTER_INTERRUPT_INIT_FAIL	-207
ECAT_ERR_MASTER_ACTIVE_FAIL	-208
ECAT_ERR_MASTER_ENI_INITCMDS_FAIL	-209
ECAT_ERR_MASTER_NO_DEVICE	-210
ECAT_ERR_MASTER_ACYCLIC_INIT_FAIL	-300
ECAT_ERR_MASTER_ACYCLIC_REQUEST_FAIL	-301
ECAT_ERR_MASTER_ACYCLIC_BUSY	-302
ECAT_ERR_MASTER_ACYCLIC_TIMEOUT	-303
ECAT_ERR_MASTER_ACYCLIC_ERROR	-304
ECAT_ERR_MASTER_ACYCLIC_WRONG_STATUS	-405
ECAT_ERR_MASTER_GENERIC_SEND_FAIL	-400
ECAT_ERR_MASTER_GENERIC_RECV_FAIL	-401
ECAT_ERR_MASTER_NOT_BEGIN	-1000
ECAT_ERR_MASTER_WRONG_BUFFER_SIZE	-1001
ECAT_ERR_MASTER_REDUNDANCY_NO_DC	-1002
ECAT_ERR_MASTER_MEMORY_ALLOCATION_FAIL	-1003
ECAT_ERR_MASTER_OSLAYER_INIT_FAIL	-1004
ECAT_ERR_MASTER_NIC_INIT_FAIL	-1005
ECAT_ERR_MASTER_BASE_INIT_FAIL	-1006
ECAT_ERR_MASTER_CIA402_INIT_FAIL	-1007
ECAT_ERR_MASTER_SETUP_PDO_FAIL	-1008
ECAT_ERR_MASTER_SCAN_NETWORK_FAIL	-1009
ECAT_ERR_MASTER_START_MASTER_FAIL	-1010
ECAT_ERR_MASTER_CYCLETIME_TOO_SMALL	-1011
ECAT_ERR_MASTER_DUMP_OUTPUT_PDO_FAIL	-1012
ECAT_ERR_MASTER_CONFIG_DEVICE_FAIL	-1013
ECAT_ERR_MASTER_CONFIG_MAPPING_FAIL	-1014
ECAT_ERR_MASTER_WAIT_BUS_SYNC_TIMEOUT	-1015
ECAT_ERR_MASTER_WAIT_MASTER_SYNC_TIMEOUT	-1016
ECAT_ERR_MASTER_CYCLIC_START_FAIL	-1017
ECAT_ERR_MASTER_WRONG_BUFFER_POINTER	-1018
ECAT_ERR_MASTER_ENI_INIT_FAIL	-1050
ECAT_ERR_MASTER_ENI_MISMATCH	-1051
ECAT_ERR_MASTER_STOPPED	-1100
ECAT_ERR_MASTER_STARTED	-1101
ECAT_ERR_MASTER_NOT_IN_PREOP	-1102
ECAT_ERR_MASTER_NOT_IN_SAFEOP	-1103

ECAT_ERR_MASTER_NOT_IN_OP	-1104
ECAT_ERR_MASTER_II_TRANSITION_FAIL	-1200
ECAT_ERR_MASTER_IP_TRANSITION_FAIL	-1201
ECAT_ERR_MASTER_PS_TRANSITION_FAIL	-1202
ECAT_ERR_MASTER_SO_TRANSITION_FAIL	-1203
ECAT_ERR_DEVICE_NOT_EXIST	-2000
ECAT_ERR_DEVICE_NOT_ATTACH	-2001
ECAT_ERR_DEVICE_NO_MAILBOX	-2002
ECAT_ERR_DEVICE_NO_DC	-2003
ECAT_ERR_DEVICE_WRONG_INPUT	-2004
ECAT_ERR_DEVICE_MEMORY_ALLOCATION_FAIL	-2005
ECAT_ERR_DEVICE_VENDOR_ID_MISMATCH	-2006
ECAT_ERR_DEVICE_PRODUCT_CODE_MISMATCH	-2007
ECAT_ERR_DEVICE_NO_SUCH_FUNCTION	-2008
ECAT_ERR_DEVICE_FUNCTION_NOT_INIT	-2009
ECAT_ERR_DEVICE_BUSY	-2010
ECAT_ERR_DEVICE_TIMEOUT	-2011
ECAT_ERR_DEVICE_NO_DATA	-2012
ECAT_ERR_DEVICE_SII_READ_FAIL	-2100
ECAT_ERR_DEVICE_SII_WRITE_FAIL	-2101
ECAT_ERR_DEVICE_PDO_NOT_EXIST	-2200
ECAT_ERR_DEVICE_PDO_OUT_OF_RANGE	-2201
ECAT_ERR_DEVICE_FOE_NOT_SUPPORT	-2300
ECAT_ERR_DEVICE_FOE_REQUEST_FAIL	-2310
ECAT_ERR_DEVICE_FOE_TIMEOUT	-2311
ECAT_ERR_DEVICE_FOE_ERROR	-2312
ECAT_ERR_DEVICE_FOE_BUFFER_TOO_SMALL	-2313
ECAT_ERR_DEVICE_FOE_READ_FAIL	-2314
ECAT_ERR_DEVICE_FOE_WRITE_FAIL	-2315
ECAT_ERR_DEVICE_COE_SDO_NOT_SUPPORT	-2400
ECAT_ERR_DEVICE_COE_SDO_INFO_NOT_SUPPORT	-2401
ECAT_ERR_DEVICE_COE_BUSY	-2410
ECAT_ERR_DEVICE_COE_REQUEST_FAIL	-2411
ECAT_ERR_DEVICE_COE_TIMEOUT	-2412
ECAT_ERR_DEVICE_COE_ERROR	-2413
ECAT_ERR_DEVICE_CIA402_NOT_EXIST	-2500
ECAT_ERR_DEVICE_CIA402_ADD_FAIL	-2501
ECAT_ERR_DEVICE_CIA402_TYPE_MISMATCH	-2502
ECAT_ERR_DEVICE_CIA402_NO_MODE_SUPPORT	-2503
ECAT_ERR_DEVICE_CIA402_WRONG_MODE	-2504
ECAT_ERR_DEVICE_CIA402_MODE_NOT_SUPPORT	-2505
ECAT_ERR_DEVICE_CIA402_CHANGE_WRONG_STATE	-2506
ECAT_ERR_DEVICE_CIA402_WRITE_OBJECT_FAIL	-2507
ECAT_ERR_DEVICE_CIA402_NO_SUCH_TOUCH_PROBE	-2580
ECAT_ERR_DEVICE_CIA402_NO_SUCH_TOUCH_PROBE_SOURCE	-2581
ECAT_ERR_DEVICE_EOE_NOT_SUPPORT	-2600
ECAT_ERR_DEVICE_EOE_NO_SUCH_PORT	-2601
ECAT_ERR_DEVICE_EOE_TOO_MUCH_CONTENT	-2602
ECAT_ERR_DEVICE_EOE_BUSY	-2610
ECAT_ERR_DEVICE_EOE_REQUEST_FAIL	-2611
ECAT_ERR_DEVICE_EOE_TIMEOUT	-2612
ECAT_ERR_GROUP_WRONG_INPUT	-3000
ECAT_ERR_GROUP_NOT_ATTACH	-3001