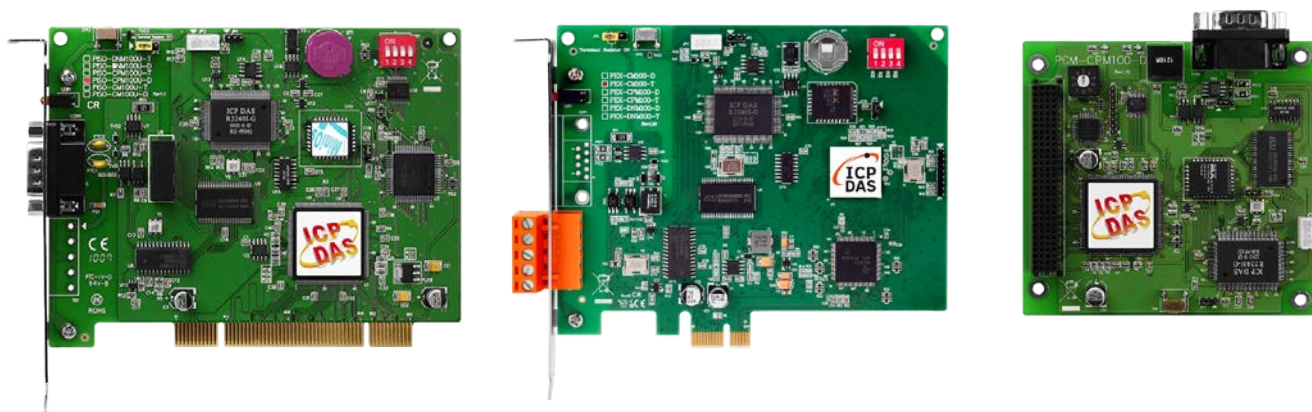


PISO-CPM100U-D/T PEX-CPM100-D/T PCM-CPM100-D

智能型 CANopen 主站板卡



重要資訊

保固說明

泓格科技股份有限公司(ICP DAS)所生產的產品，均保證原始購買者對於有瑕疵之材料，於交貨日起保有為期一年的保固。

免責聲明

泓格科技股份有限公司對於因為應用本產品所造成的損害並不負任何法律上的責任。本公司保留有任何時間未經通知即可變更與修改本文件內容之權利。本文所含資訊如有變更，恕不予另行通知。本公司盡可能地提供正確與可靠的資訊，但不保證此資訊的使用或其他團體在違反專利或權利下使用。此處包涵的技術或編輯錯誤、遺漏，概不負其法律責任。

版權所有

版權所有 2016 泓格科技股份有限公司保留所有權利。

商標識別

本文件提到的所有公司商標、商標名稱及產品名稱分別屬於該商標或名稱的擁有者所有。

連絡方式

若於使用此設定時有任何的問題，可隨時透過 mail 方式與我們聯繫。

mail : service@icpdas.com

目 錄

1. 產品資訊	7
1.1 CANopen 簡介	7
1.2 CANopen 的應用	8
1.3 CPM100 函式庫特性	9
1.4 硬體規格	11
1.5 特性	12
2. 硬體配置	13
2.1 板卡元件分佈圖	13
2.2 跳線選擇	16
2.3 腳位定義	18
2.3.1 5 針螺絲接線端子	18
2.3.2 9 針 D-sub 公頭接線端子	19
2.3.3 接線圖	20
2.4 綠色 LED 燈號	21
2.5 紅色 LED 燈號	21
2.6 硬體安裝	21
3. 軟體安裝	22
3.1 軟體架構	22
3.2 安裝驅動步驟	23
3.2.1 安裝 CPM100 CAN 板卡驅動	23
3.2.2 移除 CPM100 CAN 板卡驅動	26
4. CPM100 函式庫	27
4.1 函式列表	27
4.2 函式回傳碼	30

4.3	CANopen 主函式庫應用流程圖	32
4.4	通訊服務介紹	34
4.4.1	NMT 服務	34
4.4.2	SDO 服務	35
4.4.3	PDO 服務	35
4.4.4	SYNC 服務	36
4.4.5	EMCY 服務	36
4.5	函式描述	37
4.5.1	CPM100_GetVersion	37
4.5.2	CPM100_TotalBoard	38
4.5.3	CPM100_GetBoardSwitchNo	39
4.5.4	CPM100_GetBoardInf	40
4.5.5	CPM100_GetCANStatus	41
4.5.6	CPM100_SetFunctionTimeout	42
4.5.7	CPM100_InitMaster	43
4.5.8	CPM100_ShutdownMaster	44
4.5.9	CPM100_MasterSendBootupMsg	45
4.5.10	CPM100_SetMasterMode	46
4.5.11	CPM100_GetMasterMode	47
4.5.12	CPM100_GetFirmwareVersion	48
4.5.13	CPM100_EDS_Load	49
4.5.14	CPM100_AddNode	50
4.5.15	CPM100_RemoveNode	52
4.5.16	CPM100_RemoveAndResetNode	53
4.5.17	CPM100_DelayAndResponseTimeout	54
4.5.18	CPM100_ScanNode	55
4.5.19	CPM100_GetNodeList	56
4.5.20	CPM100_NMTChangeState	57
4.5.21	CPM100_NMTGetState	58
4.5.22	CPM100_NMTGuarding	59
4.5.23	CPM100_NMTHeartbeat	60
4.5.24	CPM100_SDOReadData	61
4.5.25	CPM100_SDOReadFile	62
4.5.26	CPM100_SDOWriteData	63
4.5.27	CPM100_SDOAbortTransmit	64
4.5.28	CPM100_PDOWrite	65
4.5.29	CPM100_PDOWrite_Fast	66
4.5.30	CPM100_PDORemote	67
4.5.31	CPM100_PDORemote_Fast	68
4.5.32	CPM100_SetPDORemotePolling	69
4.5.33	CPM100_GetPDOLastData	70
4.5.34	CPM100_GetPDOLastData_Fast	71
4.5.35	CPM100_GetMultiPDOData	72
4.5.36	CPM100_GetMultiPDOData_Fast	73
4.5.37	CPM100_GetRxPDOID	74

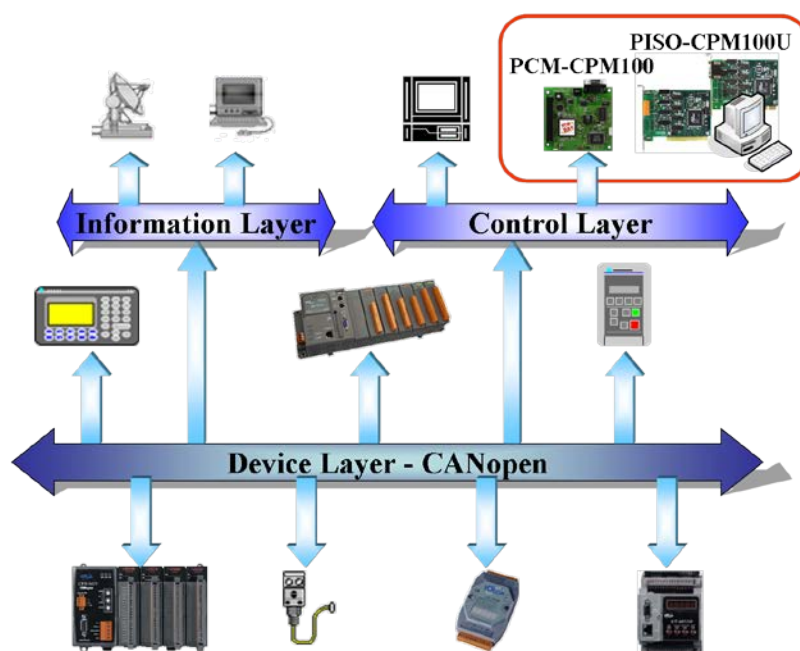
4.5.38	CPM100_GetTxPDOID	75
4.5.39	CPM100_InstallPDO	76
4.5.40	CPM100_DynamicPDO	77
4.5.41	CPM100_RemovePDO	78
4.5.42	CPM100_ChangePDOID	79
4.5.43	CPM100_GetPDOMapInfo	80
4.5.44	CPM100_InstallPDO_List	81
4.5.45	CPM100_RemovePDO_List	83
4.5.46	CPM100_PDOWUseEntry	84
4.5.47	CPM100_PDOWTxType	85
4.5.48	CPM100_PDOWEventTimer	86
4.5.49	CPM100_PDOWInhibitTime	87
4.5.50	CPM100_ChangeSYNCCID	88
4.5.51	CPM100_SetSYNC_List	89
4.5.52	CPM100_GetSYNCCID	90
4.5.53	CPM100_SendSYNCCMsg	91
4.5.54	CPM100_GetCyclicSYNCCInfo	92
4.5.55	CPM100_ChangeEMCYID	93
4.5.56	CPM100_SetEMCY_List	94
4.5.57	CPM100_GetEMCYID	95
4.5.58	CPM100_ReadLastEMCY	96
4.5.59	CPM100_GetBootUpNodeAfterAdd	97
4.5.60	CPM100_GetEMCYData	98
4.5.61	CPM100_GetNMTErr	99
4.5.62	CPM100_InstallBootUpISR	100
4.5.63	CPM100_RemoveBootUpISR	101
4.5.64	CPM100_InstallEMCYISR	102
4.5.65	CPM100_RemoveEMCYISR	103
4.5.66	CPM100_InstallNMTErrISR	104
4.5.67	CPM100_RemoveNMTErrISR	105
4.5.68	CPM100_GetMasterReadSDOWEvent	106
4.5.69	CPM100_GetMasterWriteSDOWEvent	107
4.5.70	CPM100_ResponseMasterSDOW	108
4.5.71	CPM100_InstallReadSDOWISR	109
4.5.72	CPM100_RemoveReadSDOWISR	110
4.5.73	CPM100_InstallWriteSDOWISR	111
4.5.74	CPM100_RemoveWriteSDOWISR	112
4.5.75	CPM100_GetMasterRemotePDOWEvent	113
4.5.76	CPM100_GetMasterRxPDOWEvent	114
4.5.77	CPM100_ResponseMasterPDO	115
4.5.78	CPM100_InstallRxPDOWISR	116
4.5.79	CPM100_RemoveRxPDOWISR	117
4.5.80	CPM100_InstallRemotePDOWISR	118
4.5.81	CPM100_RemoveRemotePDOWISR	119
4.5.82	CPM100_SyncGroupCycUpdata	120
4.5.83	CPM100_SyncGroupCycSend	121
4.5.84	CPM100_SyncGroupClear	122
4.5.85	CPM100_GetSYNCCGroupStopState	123
4.5.86	CPM100_InstallSYNCCGroupISR	124
4.5.87	CPM100_RemovesSYNCCGroundISR	125

5. 範例程式	-----126
5.1 範例程式簡介	-----126
5.1.1 Listen_Mode	-----127
5.1.2 NMT_Protocol	-----128
5.1.3 PDO_Parameter	-----129
5.1.4 PDO_Protocol	-----130
5.1.5 Scan_Node	-----131
5.1.6 SDO_PDO_ISR	-----132
5.1.7 SDO_Read	-----133
5.1.8 SDO_Write	-----134
5.1.9 SYNC_Protocol	-----135
5.1.10 PDO_MultiData	-----136
6. 韌體更新	-----137

1. 產品資訊

1.1 CANopen 簡介

CAN (Controller Area Network ; 控制器區域網路) 是一種串列式通訊協定，特別適合使用在主系統或子系統下提供更完整的智慧型網路設備如感應器及驅動器，它提供高安全等級和有效率的分散式即時控制，在 CAN 網路中並沒有使用傳統訊息傳送的尋址用戶和站點方式，取而代之的是訊息優先權的仲裁機制。CANopen 是一種基於 CAN 匯排流架構之上的通訊協定，它提供了低階的網路連結給簡易的工業裝置如感應器及驅動器和高階的裝置如控制器，如以下圖片所示。



CANopen 網路架構範例

CANopen是作為具備高度彈性配置能力的標準嵌入式網路而開發的，提供了多種標準化通訊物件，像適合用來傳輸即時資料的Process Data Objects (PDO)、適合用來傳輸組態資料的Service Data Objects (SDO)、可進行網路管理的Network Management Objects (NMT訊息與錯誤控制)，以及其他具有特殊功能的通訊物件 (Time Stamp、SYNC與EMCY訊息)。現今CANopen使用在各種不同的應用領域，像是醫療設備、工程車輛、航海電子、公眾傳輸與建築自動化等等。

1.2 CANopen 的應用

CANopen 是一種優化嵌入式網路的標準化網路應用層，它的規範涵蓋了裝置、介面和應用規範，標準化的應用層架構給許多不同種類的應用(例如通用的 IO 埠、馬達控制系統、航海電子等等)。

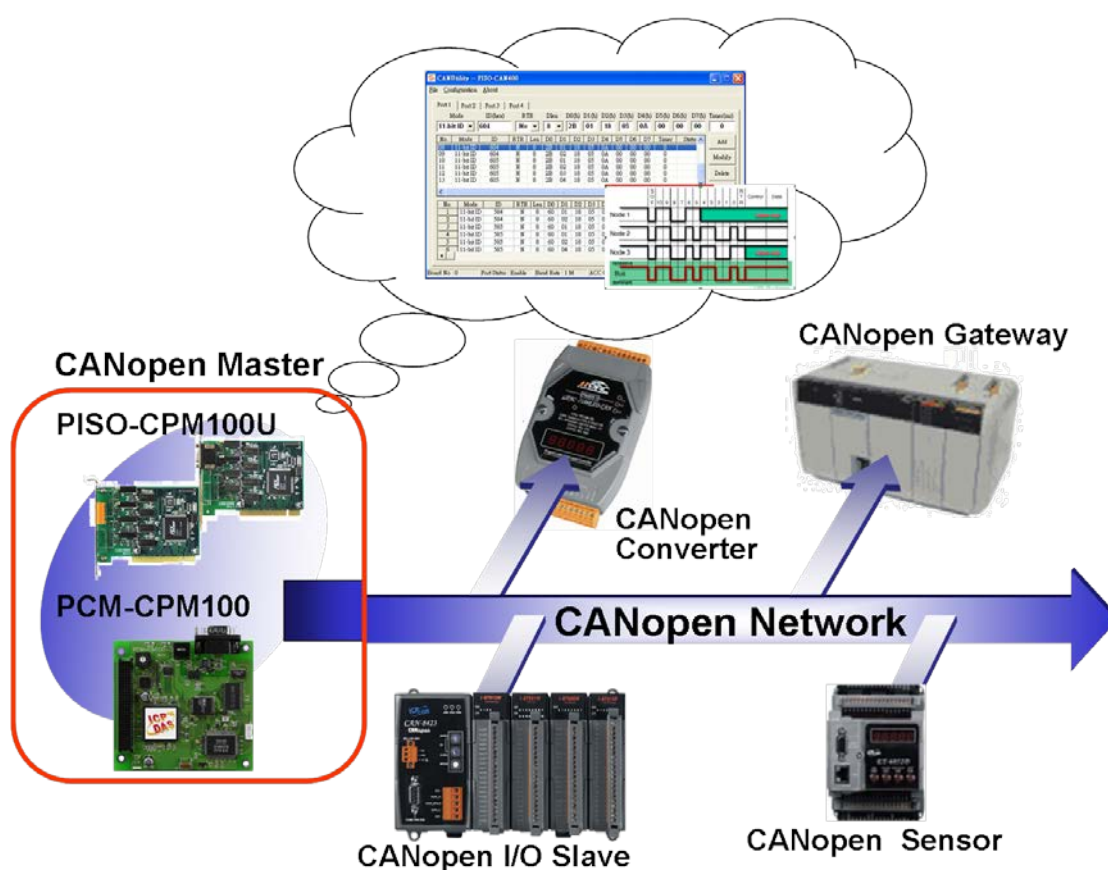
主要的 CANopen 協定和產品通常被使用在低容量和中型容量的嵌入式系統中，以下舉例一部分 CANopen 的應用領域。(如需更多資訊，請參考以下網址：<http://www.can-cia.org>)

- ◆ 卡車的控制系統。
- ◆ 越野車。
- ◆ 火車。
- ◆ 航海電子。
- ◆ 工廠自動化。
- ◆ 工具機的控制。
- ◆ 電梯和手扶梯。
- ◆ 建築自動化。
- ◆ 醫療設備和儀器。
- ◆ 非工業使用的控制系統。
- ◆ 非工業使用的設備控制。



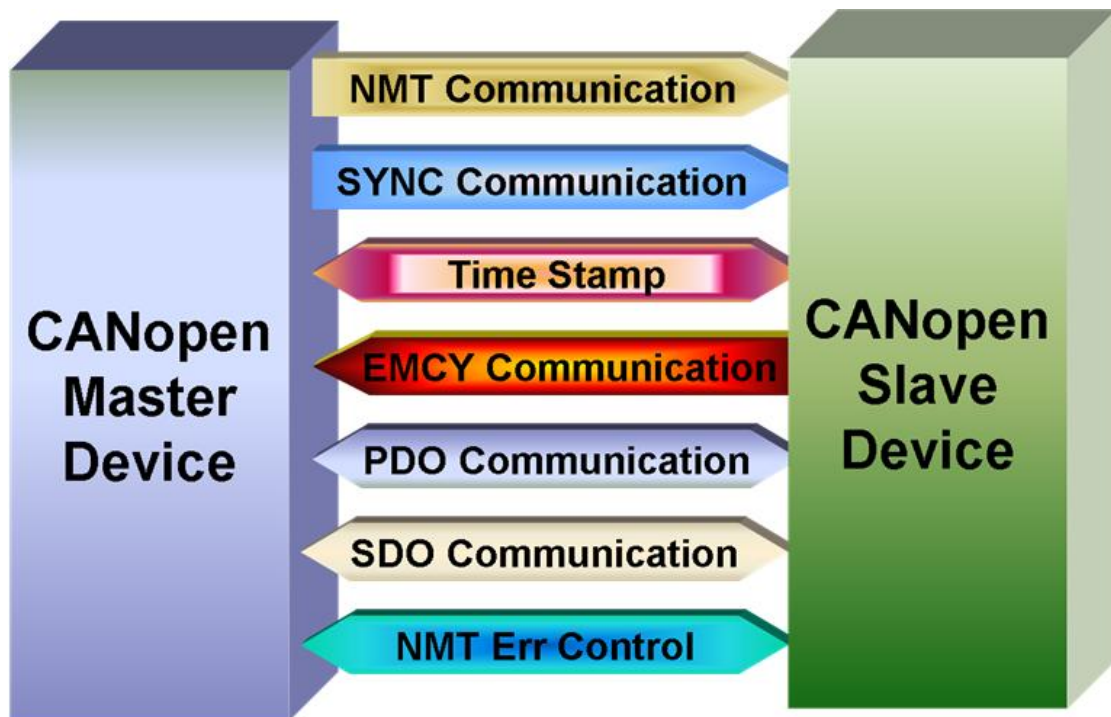
1.3 CPM100 函式庫特性

為了能方便使用 PISO-CPM100U、PEX-CPM100 和 PCM-CPM100 CAN 板卡，泓格提供了 CPM100 函式庫給 VC、VB.net 和 C# 的開發者使用，使用者可以快速的建立 CANopen 的通訊網路，大部分的通訊協定如 PDO、SDO 和 NMT 都已被函式庫自動處理。因此，它有效的降低開發 CANopen 主介面的複雜度，並且讓使用者可以忽略太細節的 CANopen 協定。這個函式庫主要支援主從架構的連結，並提供許多有用的函式去控制 CANopen 網路的僕端設備。下面架構圖描述了 CPM100 系列的一般用法。



應用架構範例

CPM100 系列遵從 CiA CANopen DS-301 V4.02 的規範並提供許多 CANopen 的功能，一般 CANopen 通訊概念如下圖所示。



CANopen 通訊概念圖

- **Node 節點管理員 (NMT Master)**
 - 改變僕端設備狀態的功能
 - 節點守衛協定與心跳協定的錯誤控制
 - 支援緊急訊息 (EMCY)
- **SDO 管理員**
 - SDO的下載與上傳的附件、區段與區塊方法
- **PDO 管理員**
 - 支援所有的傳輸種類與事件計時器
- **SYNC 管理員**
 - SYNC訊息的生產
 - 高達0.1毫秒的解析度的SYNC循環
- **EMCY 管理員**
 - EMCY訊息消費者

關於更多上面 CANopen 功能資訊，請參閱第三章與第四章的功能描述與範例程式。

1.4 硬體規格

■ PISO-CPM100U-D/T :

- ◆ 介面: Universal PCI 卡支援 5V 和 3.3V PCI 匯排流。
- ◆ 接頭: 5-pin 螺釘接線端子和 9-pin 的 D-sub 公頭接線端子。

■ PEX-CPM100-D/T :

- ◆ 介面: PCI express 匯流排。
- ◆ 接頭: 5-pin 螺釘接線端子和 9-pin 的 D-sub 公頭接線端子。

■ PISO-CAN100-D

- ◆ 介面: 支援 5V 和 3.3V PCI 匯排流的 PCI-104 卡。
- ◆ 接頭: 9-pin D-sub 公頭接線端子。

■ CPU : 80186 CPU, 80 MHz

■ CAN 控制器 : NXP SJA1000T, 16 MHz

■ CAN 收發器 : NXP 82C250

■ CAN 匯排流介面 : 符合 ISO 11898-2 規範

■ 指撥開關設定板卡號碼

■ 隔離電壓: 2500 Vrms 光隔離保護

■ 電源: 160 mA @ 3.3 V, 30 mA @ 12 V

■ 操作溫度 : 0 到 +60°C

■ 儲存溫度 : -20°C 到 70°C

■ 濕度 : 0 到 85%無結露

■ 尺寸 : 請參閱 2.1 章節

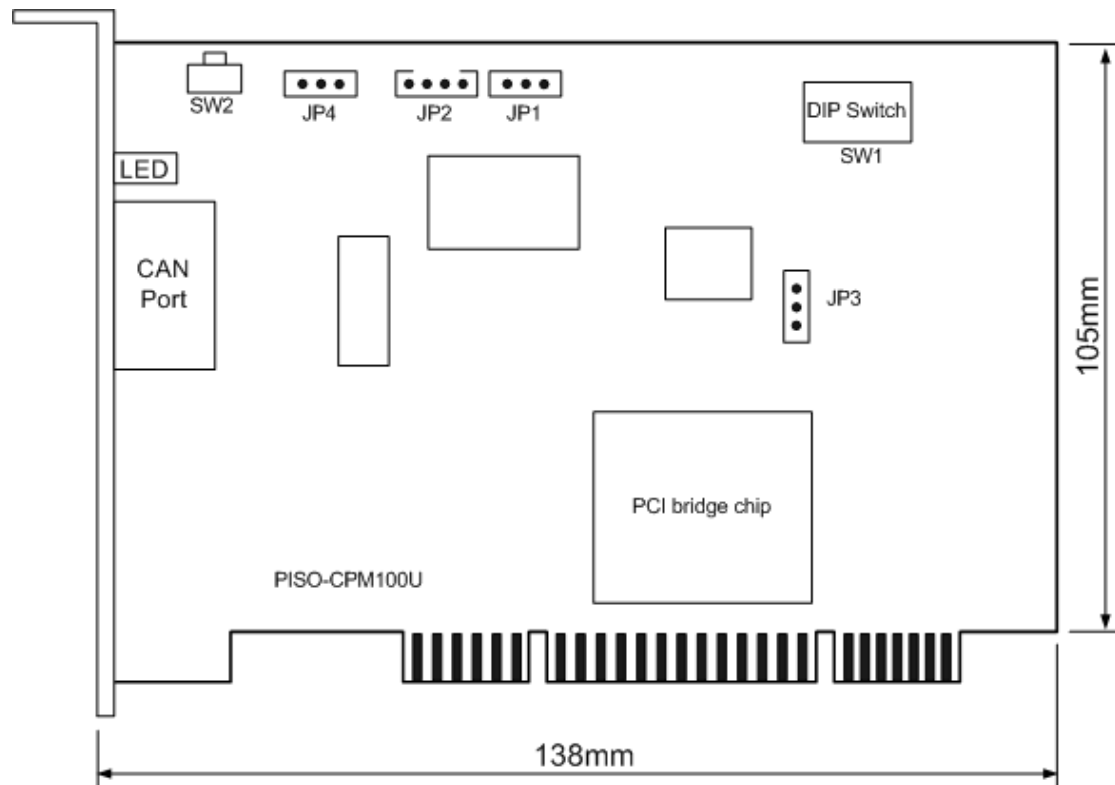
1.5 特性

- 一組 CAN 通訊埠
- 符合 CiA DS-301 V4.02
- 支援 8 種速率：10kbps、20kbps、50kbps、125kbps、250kbps、500kbps、800kbps 和 1Mbps
- 支援 1 到 127 個節點 ID
- 支援 SDO 區段上傳與下載
- 支援節點守衛協定和心跳協定
- 提供 5 組 SYNC 循環傳輸
- 支援 EDS 檔案
- 支援 EMCY 協定
- PDO 通訊支援單字節輸出、動態映射與同時讀取多個 PDO 資料
- 120Ω 終端電阻的跳線選擇
- 支援韌體更新機制
- 兩顆 LED 指示燈 (Tx/Rx 和 Err LED)
- 提供對僕端設備的監聽模式
- 阻塞(Block-function)與非阻塞(non-block-function)功能的選擇
- 提供範例程式與軟體工具
- 提供 VC++、C#.Net 和 VB.Net 的函式庫

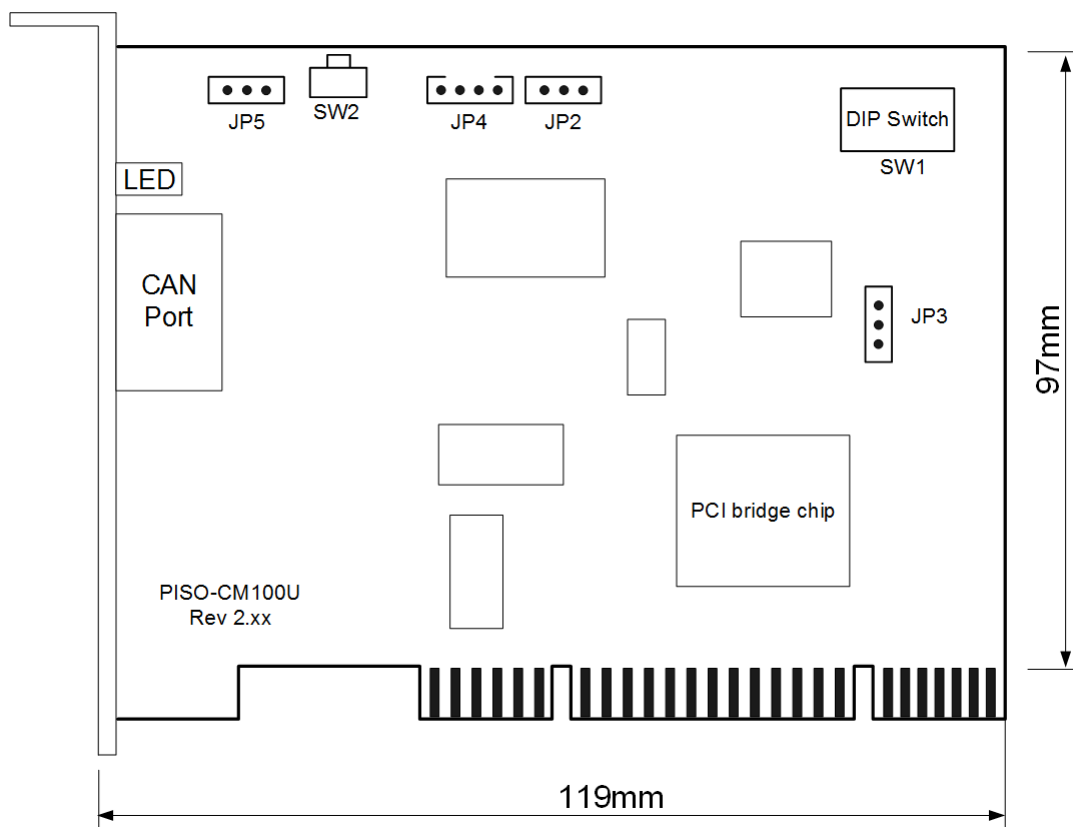
2. 硬體配置

這一節將介紹 PISO-CPM100U, PEX-CPM100 和 PCM-CPM100 的硬體設定，包含 CAN 網路的接線與與終端電阻設定等說明。

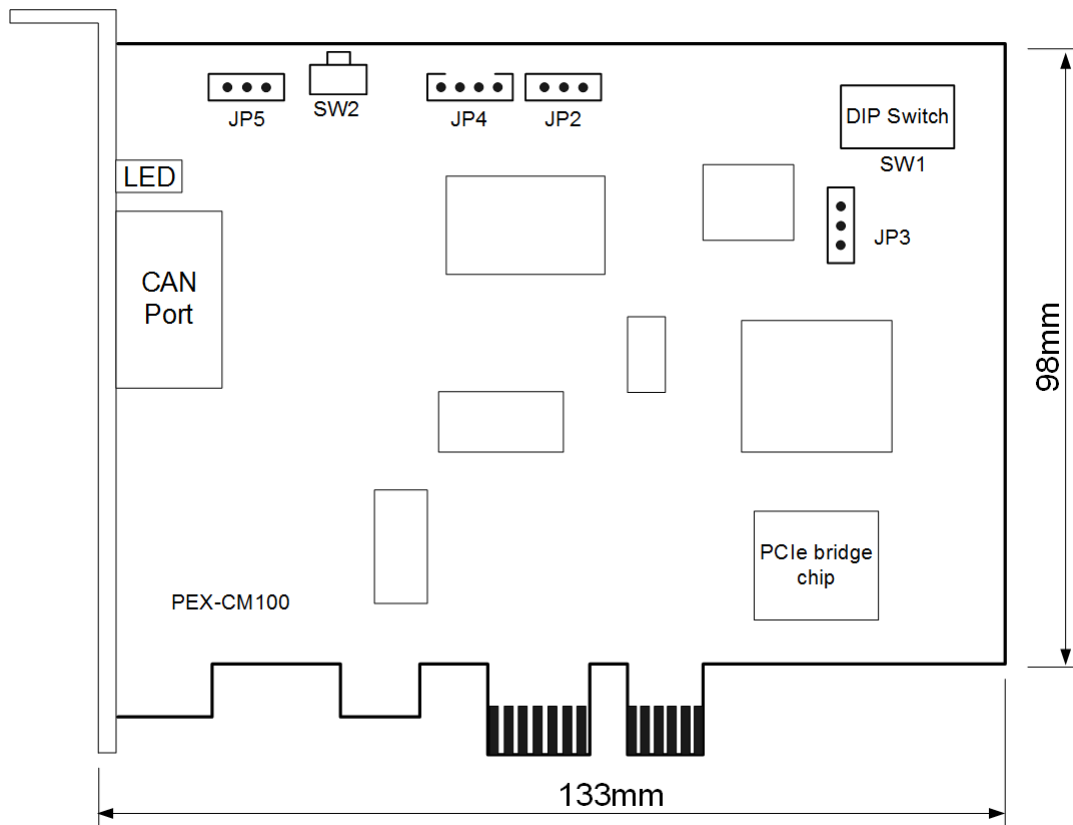
2.1 板卡元件分佈圖



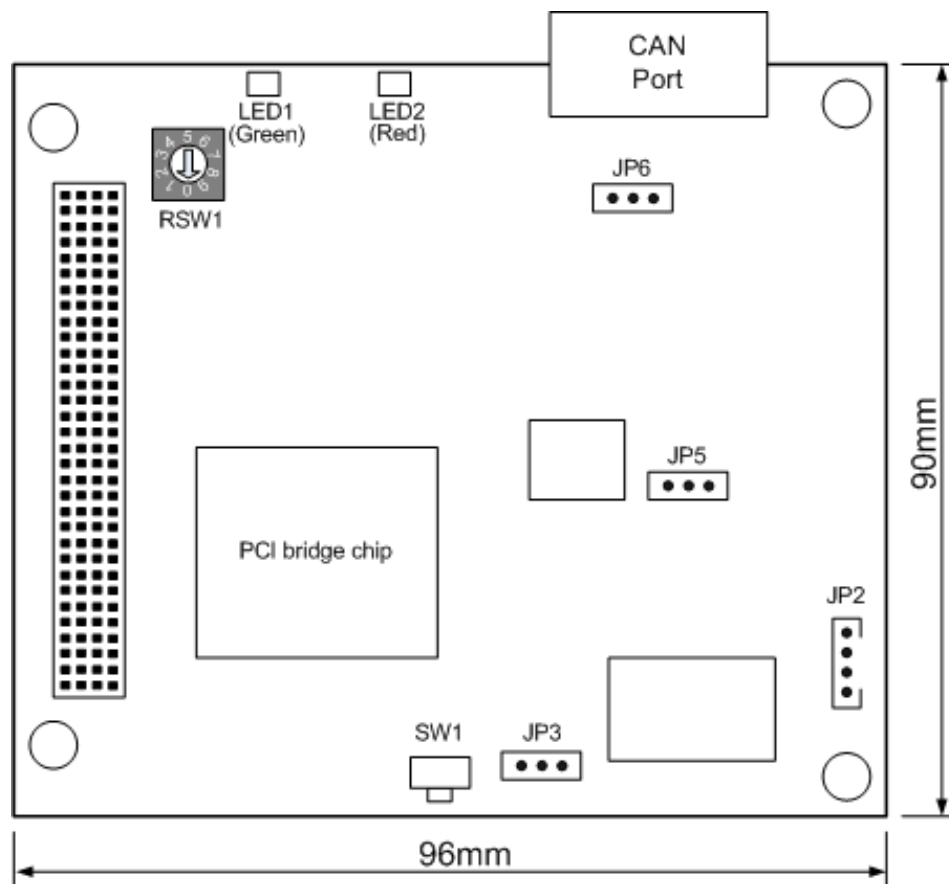
PISO-CPM100U 板卡元件分佈圖 PCB Rev 1.xx 版



PISO-CPM100U 板卡元件分佈圖 PCB Rev 2.xx 版



PEX-CPM100 板卡元件分佈圖



PCM-CPM100 板卡元件分佈圖

2.2 跳線選擇

下表列出了跳線與 DIP 開關的定義，使用者需要參考下表配置 CPM100U 板卡。請注意，PISO-CPM100U 有新舊兩種不同版本 (Rev1.xx 和 Rev2.xx)，它們的跳線有些不同，不同的跳線部分將在括號中標出。

功能	說明
Flash 保護	Flash 保護跳線。如果使用者希望保護 Flash 內的韌體，請啟用此跳線，啟用後將無法通過 Utility 更新版卡韌體。
終端電阻	CAN bus 的 120Ω 終端電阻
重置按鈕	用於進入下載模式的重置按鈕。如果使用者希望強制板卡進入下載模式，可按下此按鈕初始化板卡，使 CPM100 進入韌體下載模式。
指撥開關	DIP 開關為 CPM100 的板卡號碼。如果 bit1 為 ON，則板卡號碼為 1，板卡號碼的範圍為 0 ~ 15，請注意，PC 中每個 CM100、DNM100 和 CPM100 的號碼必須是唯一的。
旋鈕	旋轉開關是 PCM-CM100 的板卡號碼和中斷線設定。由於 PCI-104 架構只有 4 條中斷線，因此只有 0 ~ 3 的旋鈕號碼有用，若使用者將旋鈕設置為 4 ~ 9，板卡號碼會設為 4 ~ 9，但中斷線仍使用 0 ~ 3 中的一條。

功能	PISO-CPM100U v1.xx	PISO-CPM100U v2.xx	PEX-CPM100	PCM-CPM100
Flash 保護	JP3  啟用  關閉	JP3  啟用  關閉	JP3  啟用  關閉	JP5  啟用  關閉
終端電阻	JP4  啟用  關閉	JP5  啟用  關閉	JP5  啟用  關閉	JP6  啟用  關閉
重置按鈕	SW2 	SW2 	SW2 	SW1 
指撥開關 旋鈕	DIP Switch  範例為板卡號碼 1	DIP Switch  範例為板卡號碼 1	DIP Switch  範例為板卡號碼 1	 範例為板卡號碼 0

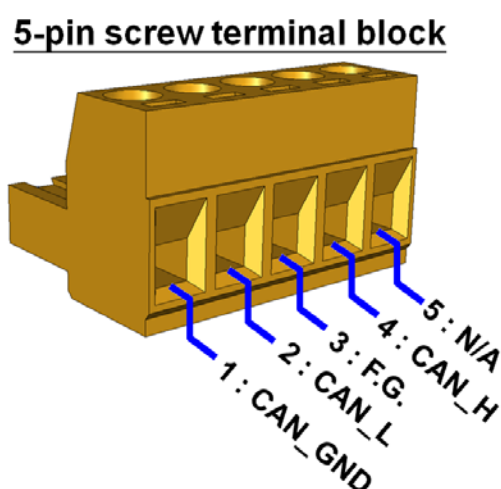
跳線與開關功能說明

2.3 腳位定義

CPM100 系列有兩種端子用於 CANopen 網路的線路連接，一種是 5 針螺絲接線端子，另一種 9 針 D-sub 公頭接線端子，而接線端子的詳細腳位定義說明如下：

2.3.1 5 針螺絲接線端子

CAN 匯排流接口的 5-pin 螺絲接線端子如下圖所示，而對應的腳位定義如下表。



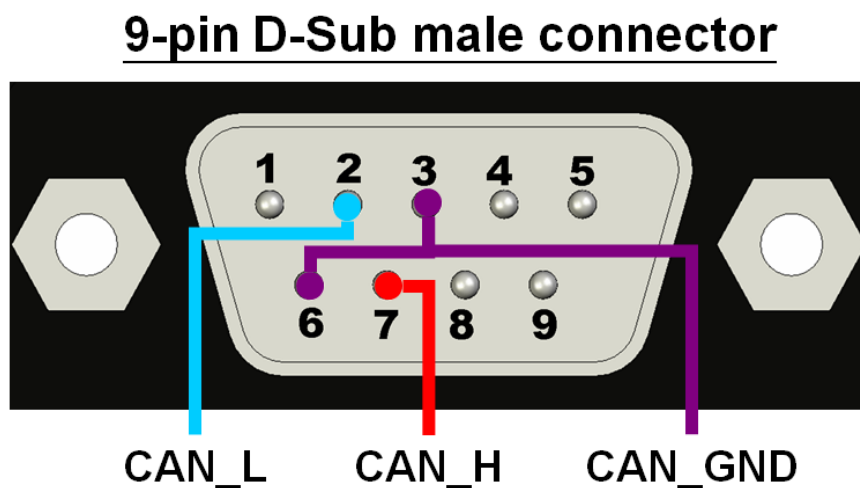
5-pin 螺絲接線端子接頭

Pin No.	腳位名稱	說明
1	CAN_GND	CAN 電源接地腳位
2	CAN_L	CAN low 訊號腳位
3	F.G	大地接線腳位
4	CAN_H	CAN high 訊號腳位
5	N/A	保留

5-pin 螺絲接線端子腳位定義

2.3.2 9 針 D-sub 公座接線端子

CAN 匯排流接口的 9-pin D-sub 公座接線端子如下圖所示，而對應的腳位定義如下表。



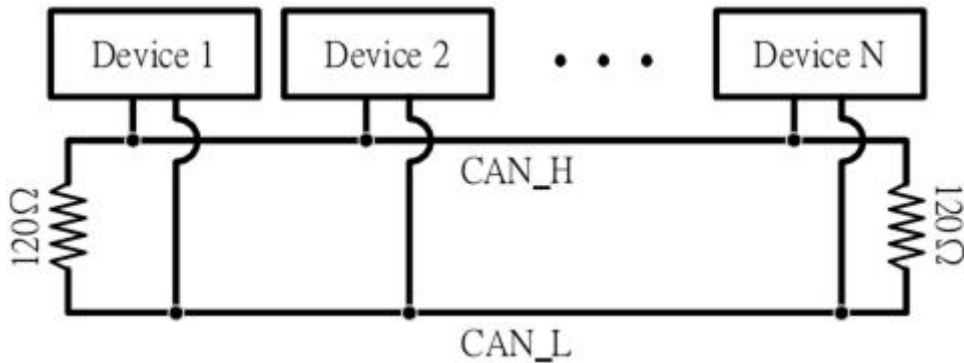
9-pin D-sub 公座接線端子

Pin No.	腳位名稱	說明
1	N/A	保留
2	CAN_L	CAN low 訊號腳位
3	CAN_GND	CAN 電源接地腳位
4	N/A	保留
5	N/A	保留
6	CAN_GND	CAN 電源接地腳位
7	CAN_H	CAN high 訊號腳位
8	N/A	保留
9	N/A	保留

9-pin D-sub 公頭接線端子腳位定義

2.3.3 接線圖

為了將 CAN 總線的波形反射影響降至最低，總線必須於端點加入終端電阻如下圖所示。根據已知 ISO 11898-2 的規範，每一個終端電阻需為 120Ω (或介於 108Ω 與 132Ω 之間)，接線本身的電阻值應小於 $70\text{m}\Omega/\text{m}$ 。在開始架設 CAN bus 之前，使用者應先檢查匯排流上的電阻值是否正常。



CANopen 網路拓譜

此外為了將長距離傳輸所引起的電壓損耗降到最低，實際使用的終端電阻值應比 ISO 11898-2 所規範的再高一點。下表的數據可作為在架設 CAN bus 時的參考。

匯排流長度 (meter)	匯排流纜線參數		終端電阻 (Ω)
	長度與電阻之關係 ($\text{m}\Omega/\text{m}$)	橫截面 (Type)	
0~40	70	0.25(23AWG)~ 0.34 mm^2 (22AWG)	124 (0.1%)
40~300	<60	0.34(22AWG)~ 0.6 mm^2 (20AWG)	127 (0.1%)
300~600	<40	0.5~ 0.6 mm^2 (20AWG)	150~300
600~1K	<20	0.75~ 0.8 mm^2 (18AWG)	150~300

電纜特性與終端電阻之間的關係

2.4 綠色 LED 燈號

當 CPM100 成功啟動後並且成功傳輸或接收一筆 CAN bus 訊號，綠色 LED 指示燈會閃爍一次。如果頻繁的收送匯排流訊息，則此綠色 LED 指示燈會顯示恆亮。

2.5 紅色 LED 燈號

當有一些錯誤發生時，紅色 LED 指示燈會恆亮直到錯誤被解決。使用者可以使用 API “CPM100_GetCANStatus”功能去得到當前匯排流的狀態來做判斷。

2.6 硬體安裝

當使用者想要開始使用 CPM100，請先遵循以下硬體安裝步驟：

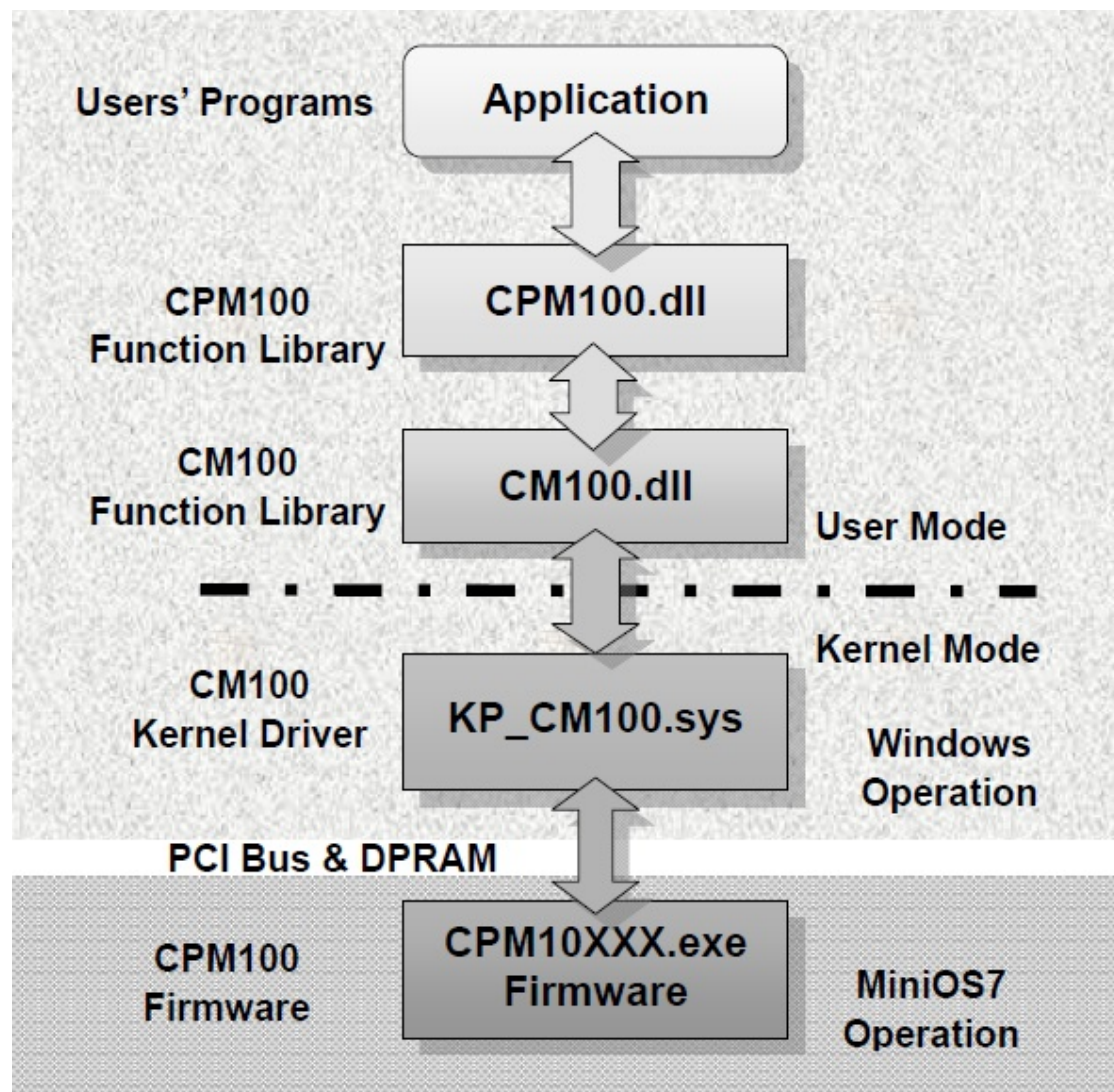
1. 關閉個人電腦電源
2. 設定終端電阻和使用 DIP 開關設定板號，詳細資訊請參閱 2.1 章節。
3. 在個人電腦主機板上尋找未使用的 PCI/PCI-104 插槽，並將 CPM100 安裝在此插槽上。
4. 連結 CAN 匯排流纜線到 5-pin 螺釘接線端子或 9-pin 公頭接線端子。

當完成上述步驟後，請開啟電腦電源。

3. 軟體安裝

3.1 軟體架構

CPM DLL 驅動是 CANopen 規範功能的總合，給 CPM100 模組板卡使用並支持 Windows 2000/XP/Vista/7 32bit 的作業環境。使用者的主端應用程式可以使用 VC++、VB.net 和 C#開發工具進行開發。應用層和驅動結構如下圖所示。



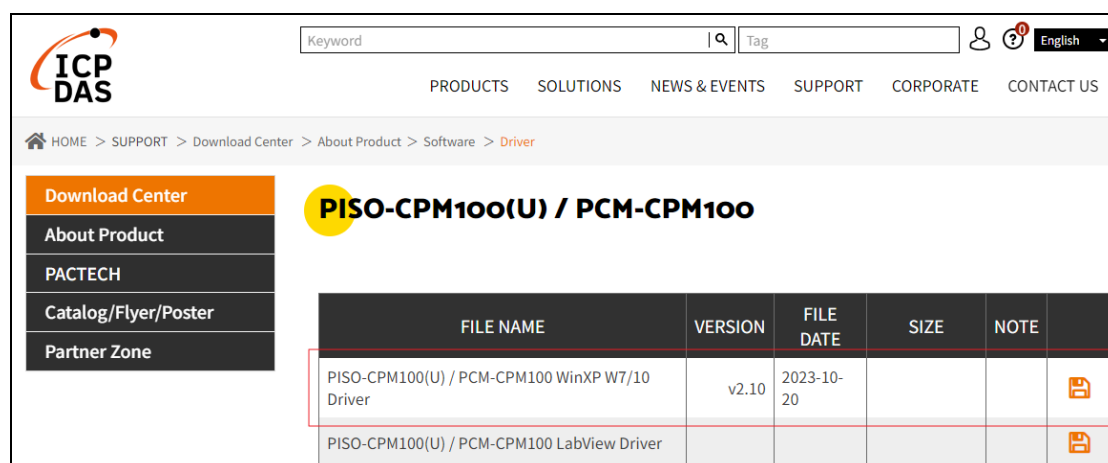
CPM100 的 driver 結構概念

3.2 安裝驅動步驟

當使用者想要使用 CPM100 CAN 板卡時，需先安裝 CPM100 驅動程式，當完成安裝程序時，CPM_Utility 軟體工具和範例程式也已自動的完成安裝。範例程式的參考有助於使用者開發 VC++、VB.net 和 C# 的 CANopen 主端應用程式介面，範例程式也提供一個簡易主/從架構的溝通介面來傳輸基本功能和 CANopen 主端程式的架構，這對開發者了解這些功能和開發 CANopen 主端程式是有幫助的，如果使用者不想開發自己的主端程式，可以使用泓格所提供的 CPM_Utility 來當作簡易 CANopen 主端程式。以下步驟說明如何一步步的安裝 CPM100 驅動：

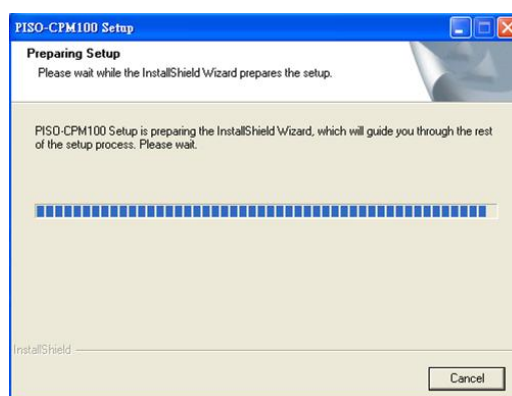
安裝 CPM100 CAN 板卡驅動程式

步驟 1: 在 PISO-CPM100U/PEX-CPM100 產品網頁上，按一下網頁右上方的 "Download Center"(下載中心)，找到 "Driver"(驅動程式)欄位後進入，即可下載驅動程式。

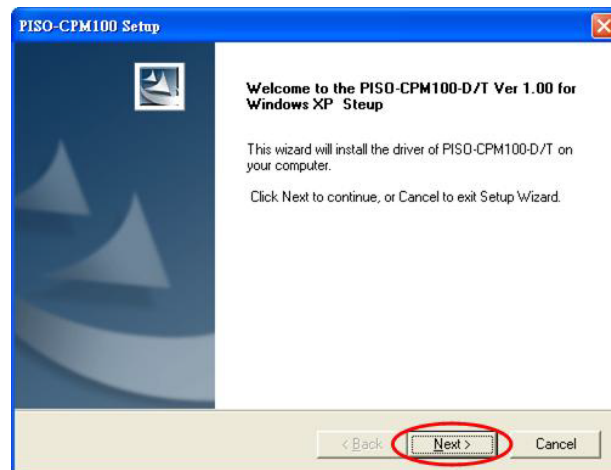


FILE NAME	VERSION	FILE DATE	SIZE	NOTE
PISO-CPM100(U) / PCM-CPM100 WinXP W7/10 Driver	v2.10	2023-10-20		
PISO-CPM100(U) / PCM-CPM100 LabView Driver				

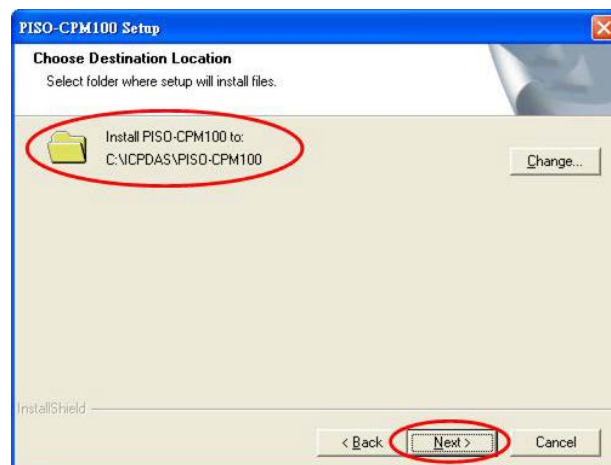
步驟 2: 雙擊驅動程式安裝檔開始安裝驅動程式，等待安裝嚮導準備就緒。



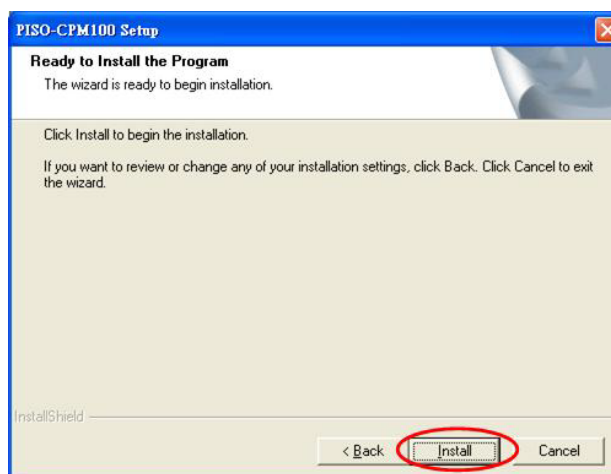
步驟 3: 點擊”Next”進行 CPM100 的安裝程序。



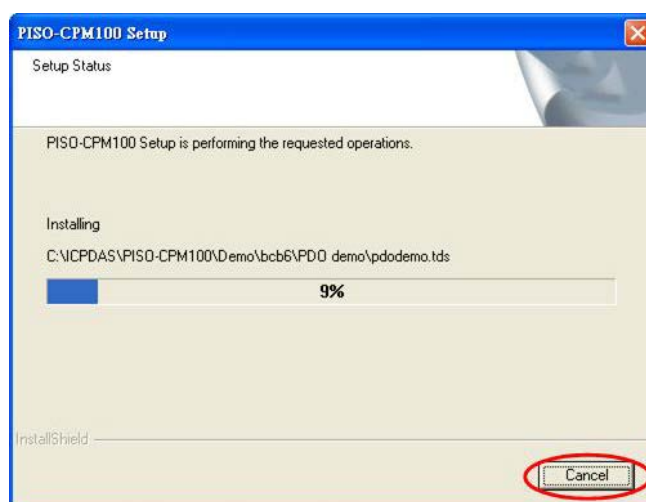
步驟 4: 選擇 CPM100 所要安裝的資料夾位置，接著按下”Next”按鍵。



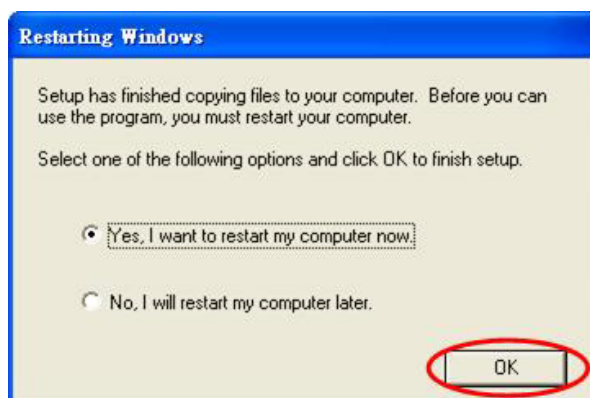
步驟 5: 按下”Install”進行安裝。



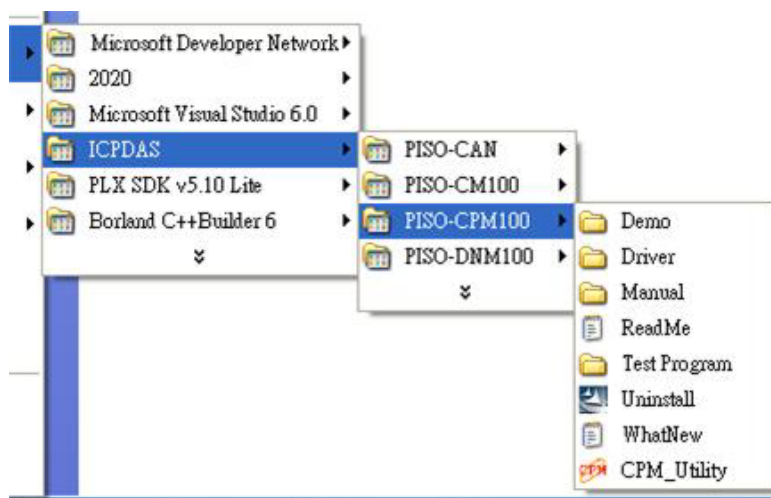
步驟 6: 等待 CPM100 安裝完成。



步驟 7: 最後，重啟電腦完成安裝程序。

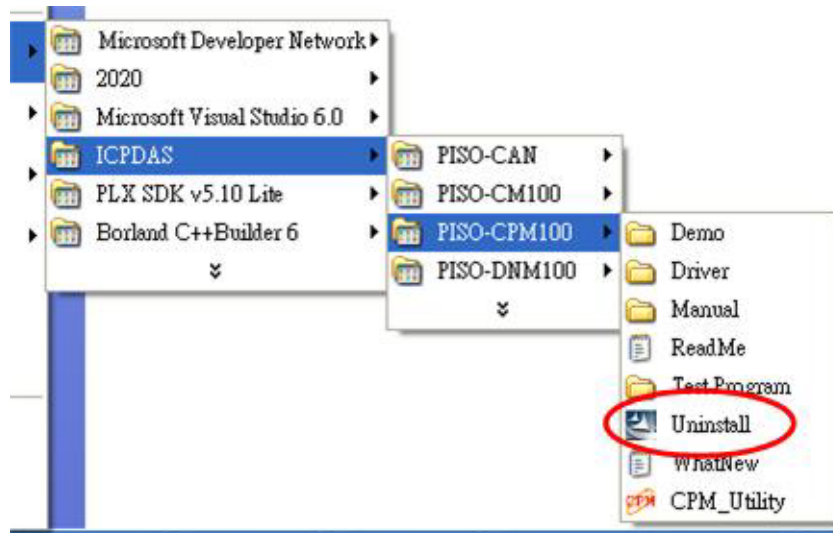


步驟 8: 當完成安裝程序時，可由 Windows 開始選單找到 CPM100 建立的資料夾。



移除 CPM100 CAN 板卡驅動

當使用者不再使用 CPM100 驅動程式時，可由 Windows 開始選單的相同路徑內找到“Uninstall”，點擊即可移除。



4. CPM100 函式庫

4.1 函式列表

為了更方便地使用 CPM100，我們在 CPM100 函式庫中提供了一些有用且易於使用的函式。這些函式庫適用於不同的編譯器，如 VC、VB 和 C#。使用者可以使用這些函數來控制 CPM100。下表列出了 CPM100 函式程式庫提供的所有函數。

監聽模式	函式名稱	描述
O	CPM100_GetVersion	取得 CPM100 函式庫的版本。
O	CPM100_TotalBoard	取得所有安裝在 PCI bus 的 CAN 板卡數量。
O	CPM100_GetBoardSwitchNo	取得從 DIP 指撥開關所設定的板號。
O	CPM100_GetBoardInf	取得 CPM100 板子相關的訊息。
O	CPM100_GetCANStatus	取得 CAN 控制器當前狀態。
O	CPM100_SetFunctionTimeout	設定一個對所有函式的逾時時間。
O	CPM100_InitMaster	啟動 CPM100 模組。
O	CPM100_ShutdownMaster	移除所有節點並停止主端程式。
X	CPM100_MasterSendBootupMsg	讓 CPM100 發出一個開機訊息。
O	CPM100_SetMasterMode	設定主端設備為正常模式或監聽模式。
O	CPM100_GetMasterMode	取得主端設備為何種操作模式。
O	CPM100_GetFirmwareVersion	取得 CPM100 模組的韌體版本。
O	CPM100_EDS_Load	從 EDS 檔案加入一個僕端節點。
O	CPM100_AddNode	建立一個僕端節點到 CPM100 主端管理員。
O	CPM100_RemoveNode	從 CPM100 主端管理員移除一個僕端節點。
X	CPM100_RemoveAndResetNode	從 CPM100 主端管理員移除一個僕端節點並且重啟僕端設備。
O	CPM100_DelayAndResponseTimeout	改變最小時間區間和 CAN 訊息回應逾時時間。
X	CPM100_ScanNode	掃描所有在 CANopen 網路上的節點。
O	CPM100_GetNodeList	取得在 CPM100 主端管理員的節點列表。
X	CPM100_NMTChangeState	設定 CANopen 節點狀態。

O	CPM100_NMTGetState	取得 CANopen 節點狀態。
O	CPM100_NMTGuarding	啟動節點守衛功能。
O	CPM100_NMTHeartbeat	啟動節點心跳功能。
X	CPM100_SDOReadData	使用上傳 SDO 協定來讀取資料。
X	CPM100_SDOReadFile	從特定的僕端節點讀取大量 SDO 資料。
X	CPM100_SDOWriteData	使用下載 SDO 協定去寫入資料。
X	CPM100_SDOAbortTransmit	傳送 SDO 中斷訊息。
X	CPM100_PDOWrite	使用 PDO 協定對 CANopen 節點寫入資料。
X	CPM100_PDOWrite_Fast	只發送 RxPDO 並不做任何檢查。
X	CPM100_PDORemote	使用 PDO 協定讀取 CANopen 節點資料。
X	CPM100_PDORemote_Fast	只傳送 Rtr 訊息並不對回應訊息做檢查。
X	CPM100_SetPDORemotePolling	設定 PDO 輪詢列表並輪詢。
O	CPM100_GetPDOLastData	取得最後一個輸入或輸出的 PDO 資料。
O	CPM100_GetPDOLastData_Fast	在快速模式底下取得最後一個輸入或輸出的 PDO 資料。
O	CPM100_GetMultiPDOData	一次取得多個輸入或輸出的 PDO 資料。
O	CPM100_GetMultiPDOData_Fast	在快速模式底下一次取得多個輸入或輸出的 PDO 資料。
O	CPM100_GetRxPDOID	取得特定僕端設備 RxPDO 的所有 COB-ID。
O	CPM100_GetTxPDOID	取得特定僕端設備 TxPDO 的所有 COB-ID。
X	CPM100_InstallPDO	安裝並開啟特定 PDO。
X	CPM100_DynamicPDO	新增或變更 PDO 對僕端設備的映射。
X	CPM100_RemovePDO	移除特定的 PDO 映射項目或物件。
X	CPM100_ChangePDOID	改變僕端設備的 PDO COB-ID。
O	CPM100_GetPDOMapInfo	取得所有 PDO 相關資訊。
O	CPM100_InstallPDO_List	手動安裝 PDO 物件並不確認是否存在。
O	CPM100_RemovePDO_List	手動移除 PDO 物件並不確認真實的狀態。
X	CPM100_PDOWUseEntry	改變 PDO 物件的有效項目數量。
X	CPM100_PDOTxType	設定特定 TxPDO 的傳輸型態。
X	CPM100_PDOWEventTimer	設定特定 TxPDO 的事件時間。

X	CPM100_PDOInhibitTime	設定特定 TxPDO 的抑制時間。
X	CPM100_ChangeSYNCID	改變 SYNC 的 COB-ID。
O	CPM100_SetSYNC_List	設定 SYNC COB-ID 並不確認是否存在。
O	CPM100_GetSYNCID	取得 SYNC COB-ID。
X	CPM100_SendSYNCMsg	傳送 SYNC 訊息。
X	CPM100_GetCyclicSYNCInfo	取得所有循環傳送的 SYNC 資訊。
X	CPM100_ChangeEMCYID	設定 EMCY 的 COB-ID。
O	CPM100_SetEMCY_List	設定 EMCY COB-ID 並不確認是否存在。
O	CPM100_GetEMCYID	取得 EMCY COB-ID。
O	CPM100_ReadLastEMCY	從僕端設備取得最後的 EMCY 訊息。
O	CPM100_GetBootupNodeAfterAdd	從已加入 CPM100 節點列表的節點取得開機訊息。
O	CPM100_GetEMCYData	從 EMCY 緩衝區取得 EMCY 訊息。
O	CPM100_GetNMTErr	從 NMT 事件緩衝區取得 NMT 錯誤事件。
O	CPM100_InstallBootUpISR	安裝使用者自定義的僕端設備開機程序。
O	CPM100_RemoveBootUpISR	移除僕端設備的開機程序。
O	CPM100_InstallEMCYISR	安裝使用者自定義的 EMCY 程序。
O	CPM100_RemoveEMCYISR	移除 EMCY 程序。
O	CPM100_InstallNMTErrISR	安裝使用者自定義的守衛/心跳事件程序。
O	CPM100_RemoveNMTErrISR	移除守衛/心跳事件程序。
O	CPM100_GetMasterReadSDOEvent	取得傳送 CPM100 的 SDO 讀取訊息。
O	CPM100_GetMasterWriteSDOEvent	取得傳送 CPM100 的 SDO 寫入訊息。
X	CPM100_ResponseMasterSDO	對 SDO 發送者回傳 SDO 訊息。
O	CPM100_GetMasterRemotePDOEvent	取得傳送到 CPM100 的遠端 PDO 訊息。
O	CPM100_GetMasterRxPDOEvent	取得傳送到 CPM100 的 RxPDO RTR 訊息。
X	CPM100_ResponseMasterPDO	對 RTR 發送者回傳 RxPDO RTR 訊息。
O	CPM100_InstallRxSDOISR	安裝使用者自定義的 SDO 主端程序。
O	CPM100_RemoveRxSDOISR	移除 SDO 主端程序。
O	CPM100_InstallRxPDOISR	安裝使用者自定義的 PDO 寫入主端程序。
O	CPM100_RemoveRxPDOISR	移除 PDO 寫入主端程序。

O	CPM100_InstallTxPDOISR	安裝使用者自定義的遠端 PDO 主端程序。
O	CPM100_RemoveTxPDOISR	移除遠端 PDO 主端程序。
X	CPM100_SyncGroupCycUpdata	為 SYNC 協定更新 PDO 群組資料。
X	CPM100_SyncGroupCycSend	開始傳送 SYNC 信號並符合 PDO 資料格式。
X	CPM100_SyncGroupClear	清除緩衝區裡面的 SYNC 和 PDO 群組資料。
X	CPM100_GetSYNCGroupStopState	取得 SYNC 和 PDO 群組的狀態。
X	CPM100_InstallSYNCGroupISR	安裝 SYNC 和 PDO 群組資料程序。
X	CPM100_RemoveSYNCGroupISR	移除 SYNC 和 PDO 群組資料程序。

4.2 函式回傳碼

下面列表說明 CANopen 函式庫支援的所有回傳碼。

回傳碼	錯誤名稱	描述
0	CPM_NoError	代表正常無誤。
1	CPM_DriverError	沒有開啟驅動程序。
3	CPM_BoardNumberErr	CPM100 沒有特定的板號。
5	CPM_ConfigErr	CPM100 沒有成功的設定配置。
6	CPM_MasterInitErr	CPM100 初始化失敗。
7	CPM_MasterNotInit	CPM100 沒有被初始化。
8	CPM_ListenMode	CPM100 目前在監聽模式。
9	CPM_NodeErr	設定節點編號錯誤。
10	CPM_NodeExist	這個節點已存在。
12	CPM_TxBusy	Tx 緩衝區忙碌中，請等待後重傳。
13	CPM_UnknowCmd	此韌體版本無支援此函式。
14	CPM_CmdReceErr	收到的命令有錯誤的資料長度。
15	CPM_DataEmpty	無接收到資料。
16	CPM_MemAllocErr	CPM100 沒有足夠的記憶體空間。
17	CPM_SendCycMsgErr	循環訊息傳送錯誤。
18	CPM_StatusErr	CANopen 僕端設備的 NMT 狀態發生錯誤。
20	CPM_SetGuardErr	設定守衛和生存時間的參數發生錯誤。

21	CPM_SetHbeatErr	設定心跳參數發生錯誤。
22	CPM_SegLenErr	SDO 區段收到錯誤的長度資料。
23	CPM_SegToggleErr	SDO 區段收到錯誤的交替位元。
24	CPM_SegWriteErr	SDO 寫入錯誤區段。
25	CPM_Abort	回傳訊息是一個中止訊息。
26	CPM_PDOLenErr	PDO 輸出錯誤的長度資料。
27	CPM_COBIDErr	次 COB-ID 不存在或是錯誤的。
28	CPM_PDOLnstErr	安裝 PDO 物件發生錯誤。
29	CPM_PDODynaErr	PDO 映射資料設定錯誤。
30	CPM_PDONumErr	PDO 編號與 COB-ID 不匹配。
31	CPM_PDOSetErr	PDO 參數設定錯誤。
32	CPM_PDOEntryErr	PDO 項目參數多於可用的項目。
33	CPM_SetCobIdErr	EMCY 或 SYNC 的 COB-ID 設定錯誤。
34	CPM_CycFullErr	已經存在 5 個輪詢訊息程序。
35	CPM_Timeout	回應訊息逾時。
36	CPM_DataLenErr	資料長度設定錯誤。
37	CPM_SyncIdDifferent	在僕端設備群組裡沒有符合的 SYNC ID。
40	CPM_Wait	命令沒有執行完成。(只在非阻塞模式)
41	CPM_Processing	命令正在執行。(只在非阻塞模式)
50	CPM_LoadEDSErr	載入 EDS 檔案失敗。
51	CPM_EDSFormatErr	EDS 檔案格式不正確。

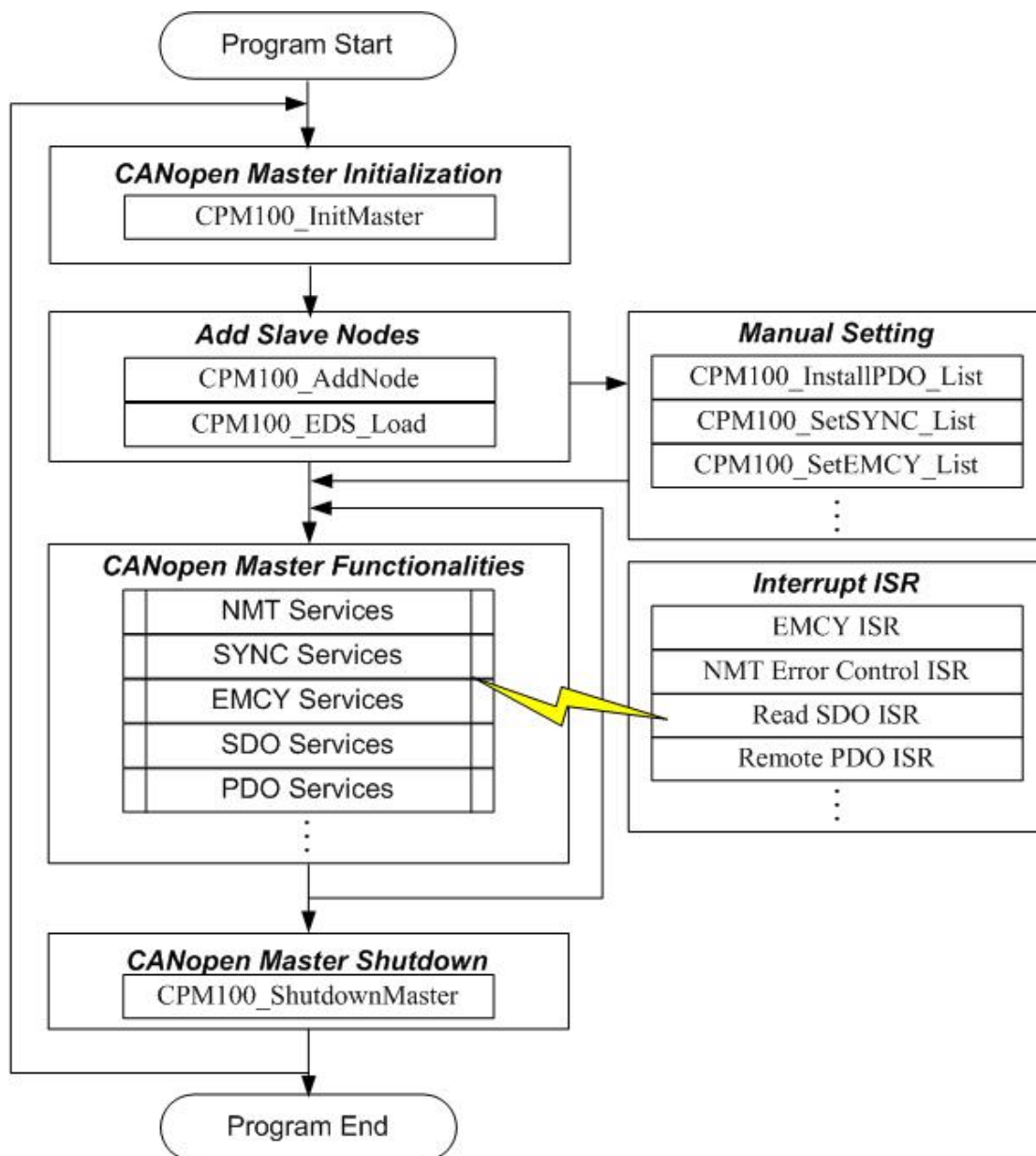
4.3 CANopen 主函式庫應用流程圖

在這個章節描述使用者如何使用 CANopen 函式庫來建立使用者的應用程式，這個資訊可以使開發者容易應用 CANopen 的函式庫。當在建立一個 CANopen 的主端程式有些操作條件必須遵守，舉例來說，如果 CANopen 的節點設備裝態在預操作模式(Pre-operation Mode)的話，則 PDO 通訊物件將不被允許使用，更多資訊請參考第五章的範例程式。

當使用者想要使用 CANopen 主端函式庫的函式時，必須先呼叫"CPM100_InitMaster"函式，這個函式是用來初始化 CPM100 模組和配置 CAN 埠。

在成功的完成 CAN 介面的初始化後，使用者需要使用函式 "CPM100_AddNode" 或 "CPM100_EDS_Load" 去加入至少一個 CANopen 裝置在節點列表裡。"CPM100_AddNode"函式可以使用自動或手動的方式來建立節點，如果使用者要用手動方式建立節點，則 POD ID、SYNC ID 和 EMCY ID 分別要使用函式 "CPM100_InstallPDO_List"、"CPM100_SetSYNC_List" 和 "CPM100_SetEMCY_List"去手動加入。

如果 "CPM100_InitMaster"、"CPM100_AddNode" 和 "CPM100_EDS_Load" 函式都已執行完成，則 NMT、SYNC、EMCY、SDO 和 PDO 通訊服務在呼叫 "CMP100_ShutdownMaster" 之前都可以在任何時間被使用，因為 "CMP100_ShutdownMaster" 會中止所有被 "CPM100_InitMaster" 函式所創造出來的程序。



主要程式設計順序圖

4.4 通訊服務介紹

4.4.1 NMT 服務

CANopen函式庫提供許多如"CPM100_AddNode"、"CPM100_EDS_Load"、"CPM100_RemoveNode"、"CPM100_NMTChangeState"、"CPM100_NMTGetState"、"CPM100_NMTHeartbeat"和"CPM100_NMTGuarding"等NMT相關服務的函式。成為主端設備的先決條件是先將從站節點設備透過"CPM100_AddNode"或"CPM100_EDS_Load"函式進行註冊，被註冊過的節點設備可以藉由"CPM100_RemoveNode"函式從節點列表內移除，透過NMT服務協定，NMT主端可以控制從站設備的狀態。下表為NMT命令的參數和功能，同時也是"CPM100_NMTChangeState"所對應的輸入參數。

命令參數	描述
1 (0x01)	進入操作狀態 (Enter Operational)
2 (0x02)	停止 (Stop)
128 (0x80)	進入預操作狀態 (Enter Pre-Operational)
129 (0x81)	節點重置 (Reset Node)
130 (0x82)	通訊重置 (Reset Communication)

使用"CPM100_NMTGetState"函式取得的節點狀態數值如下表描述。

節點狀態	描述
4 (0x04)	停止 (STOPPED)
5 (0x05)	操作狀態 (OPERATIONAL)
127 (0x7F)	預操作狀態 (PRE-OPERATIONAL)

如果從站設備已新增在節點列表內，可藉由呼叫"CPM100_NMTGuarding"和"CPM100_NMTHeartbeat"函式啟動節點守衛協定和心跳協定功能，也可以透過這兩個函式改變守衛協定和心跳協定的參數。

4.4.2 SDO 服務

當物件資料長度 ≤ 4 位元組時，將使用 "Initiate SDO download協議" 或 "Initiate SDO upload協議"，如果物件資料長度 > 4 位元組，將使用 "Download SDO segment協議" 或 "Upload SDO segment協議"。呼叫 CPM100_SDOReadData 和 CPM100_SDOWriteData 這兩個函式時，將根據物件資料長度自動選擇SDO協議。

"CPM100_SDOAbortTransmit"函式可以在任何時候中止尚未傳輸的 SDO 訊息，使用中止服務不須經過從站設備的確認。

4.4.3 PDO 服務

在使用"CPM100_AddNode"加入從站設備之後，預設TxPDOs和RxPDOs (TxPDO 1 ~ 10, RxPDO 1 ~ 10)也將被加入CPM100的控制列表下，如果除了預設值之外還有其他PDOs時，需使用"CPM100_InstallPDO"函式來安裝這些TxPDOs或RxPDOs的物件，"CPM100_DynamicPDO"函式則是可以用來加入或改變PDO的映射物件，每個PDO支援0到8個應用物件，這些物件被定義在CANopen CiA-401的規範內，而它們對應到DI/DO/AI/AO通道的相關參數。在呼叫"CPM100_InstallPDO"和"CPM100_DynamicPDO"函式之後，PDO通訊物件將被映射和啟用，當某些PDO物件已不再使用時，可以使用"CPM100_RemovePDO"函式去移除它。

當使用者想通過 PDO 輸出資料時，可呼叫CPM100_PDOWriteData函式，該函式可寫入整個 PDO 的 8-bytes 資料，也可以寫入其中某部分。使用該函式將通過 PDO 協定向相應節點發送資料，同時將輸出資料放入 PDO 緩衝區，因此你可以查看 PDO 的輸出歷史記錄，但如果 CPM100 與從站之間的連接中斷，歷史輸出資訊可能與從站上的真實狀態不一致。

在CANopen規範中，使用者可以藉由遠端傳輸要求CAN訊息的功能取得TxPDOs的資料，這裡是由"CPM100_PDORemote"函式實現此功能，使用者也可以使用"CPM100_GetPDOLastData"函式從PDO緩衝區內取得最後一筆的RxPDO和TxPDO的資料。

4.4.4 SYNC 服務

呼叫函式 CPM100_SendSYNCMsg 可啟動 SYNC 物件傳輸，該函式支援單次 SYNC 訊息傳輸和循環 SYNC 訊息傳輸。函式 CPM100_SendSYNCMsg 的參數 "Timer" 可調整主站發送 SYNC COB-ID 的週期，參數 "Times" 可以設置 SYNC 報文的發送次數，如果 "Timer" 參數設置為 0，則將停止 SYNC 物件的發送，當 "Times" 參數設置為非零時，該功能將發送 SYNC 訊息，直至計時器數到為 0 或參數 "Times" 次數倒數到 0。

4.4.5 EMCY 服務

緊急資訊是由設備內部的錯誤狀況觸發的，使用者可以呼叫函式 CPM100_ReadLastEMCY 接收 EMCY 訊息，或呼叫函數 CPM100_InstallEMCYISR 安裝用戶自訂的 EMCY 中斷程序，在這種情況下，如果有 CAN 從站發送 EMCY，這些 EMCY 報文將由用戶定義的 EMCY 中斷程序處理。

4.5 函式描述

4.5.1 CPM100_GetVersion

◆ 說明：

取得 CPM100.dll 驅動程式的版本資訊。

◆ 語法：

WORD CPM100_GetVersion(**void**)

◆ 參數：

無

◆ 回傳：

DLL 版本資訊。

4.5.2 CPM100_TotalBoard

- ◆ 說明：
取得所有安裝在 PCI bus 的 CAN 板卡數量。
- ◆ 語法：
WORD CPM100_TotalBoard (**void**)
- ◆ 參數：
無。
- ◆ 回傳：
回傳板卡數量。

4.5.3 CPM100_GetBoardSwitchNo

◆ 說明：

讀取在 DIP 開關的 CPM100 板號。

◆ 語法：

WORD CPM100_GetBoardSwitchNo (**BYTE** BoardCntNo, **BYTE**
*BoardNo)

◆ 參數：

BoardCntNo：[input] 特定 CPM100 模組的編號。例如，當第一個 CPM100 被啟用，其值為 0，如果第二個被啟用，其值為 1。

*BoardNo： [output] 存取 DIP 開關板號的變數位址。

4.5.4 CPM100_GetBoardInf

◆ 說明：

取得 CPM100 模組的資訊，包含供應商名稱、設備名稱和中斷編號。

◆ 語法：

WORD CPM100_GetBoardInf (**BYTE** BoardNo, **DWORD** *dwVID,
DWORD *dwDID, **DWORD**
*dwSVID, **DWORD** *dwSDID,
DWORD *dwSAuxID, **DWORD**
*dwIrqNo)

◆ 參數：

BoardNo : [input] CPM100 DIP 開關板卡編號(範圍：0 到 15)。
*dwVID : [output] 板卡的供應商名稱。
*dwDID : [output] 板卡的設備名稱。
*dwSVID : [output] 板卡的子供應商名稱。
*dwSDID : [output] 板卡的子設備名稱。
*dwSAuxID : [output] 板卡的子輔助名稱。
*dwIrqNo : [output] 板卡的邏輯中斷編號。

4.5.5 CPM100_GetCANStatus

◆ 說明：

取得特定 CPM100 模組的 CAN 控制器狀態。

◆ 語法：

WORD CPM100_GetCANStatus (**BYTE** BoardNo, **BYTE** *bStatus)

◆ 參數：

BoardNo： [input] CPM100 DIP 開關板卡編號(範圍：0 到 15)。

*bStatus： [output] 取得 CAN 控制器狀態的變數位址。下表為 bStatus 參數的位元定義。

位元(Bit)	名稱	數值	狀態
Bit 7	匯排流狀態	1	匯排流關閉
		0	匯排流開啟
Bit 6	錯誤狀態	1	錯誤
		0	正常
Bit 5	傳輸狀態	1	傳輸中
		0	閒置
Bit 4	接收狀態	1	接收中
		0	閒置
Bit 3	傳輸完成狀態	1	傳輸完成
		0	尚未完成
Bit 2	傳輸緩衝區狀態	1	釋放
		0	鎖住
Bit 1	資料溢位狀態	1	溢位
		0	不足
Bit 0	接收緩衝區狀態	1	有資料
		0	空的

4.5.6 CPM100_SetFunctionTimeout

◆ 說明：

有些函式有時候會花費比較多的時間來完成任務，例如“CPM100_ScanNode”，而 CPM100 模組一直沒有收到函式回傳的值，當到達逾時時間則會回報“CPM_Timeout”錯誤訊息，使用者可以透過此函式來調整適合的逾時時間給所有的函式，預設逾時時間為 1 秒。

◆ 語法：

WORD CPM100_SetFunctionTimeout (**BYTE** BboardNo, **WORD** Timeout)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。

Timeout : [input] 所有函式的最大逾時時間(ms)。

4.5.7 CPM100_InitMaster

◆ 說明：

此函式用來當配置 CAN 控制器和初始化 CPM100 模組，在使用其他函式前必須先呼叫此函式一次。

◆ 語法：

WORD CPM100_InitMaster (**BYTE** BoardNo, **BYTE** Node, **BYTE** BaudRate, **BYTE** BlockMode)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

Node： [input] 主端節點 ID。當參數為 0 時，此為正常主端且其它主端皆無法控制此主端，如果參數介於 1 到 127 之間，則其他主端可透過一些 ISR 函式來控制此主端。

BaudRate： [input] CPM100 模組的鮑率。

數值	鮑率
0	10 Kbps
1	20 Kbps
2	50 Kbps
3	125 Kbps
4	250 Kbps
5	500 Kbps
6	800 Kbps
7	1 Mbps

BlockMode： [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.8 CPM100_ShutdownMaster

◆ 說明：

此函式移除所有掛載在主端的僕端設備並停止其他所有函式，在退出使用者應用程式前必須呼叫此函式。

◆ 語法：

WORD CPM100_ShutdownMaster (**BYTE** BoardNo)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

4.5.9 CPM100_MasterSendBootupMsg

◆ 說明：

此函式可讓“CPM100_InitMaster”被呼叫後，重送一個開機啟動訊息。

◆ 語法：

WORD CPM100_MasterSendBootupMsg (**BYTE** BoardNo, **BYTE** BlockMode)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

BlockMode： [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.10 CPM100_SetMasterMode

◆ 說明：

此函式可設定主端為監聽模式或正常模式。在監聽模式中，CPM100 模組無法傳送任何 CANopen 的訊息，同時有些函式也無法使用在此模式下。在呼叫完成“CPM100_InitMaster”函式後，使用者可以在任何時間選擇監聽模式或正常模式。

◆ 語法：

WORD CPM100_SetMasterMode (**BYTE** BoardNo, **BYTE** Mode, **BYTE** BlockMode)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

Mode： [input] 1 為監聽模式，其餘為正常模式 (預設為正常模式)。

BlockMode： [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.11 CPM100_GetMasterMode

◆ 說明：

使用者可以呼叫此函式來取得現在主端為何種模式。

◆ 語法：

WORD CPM100_GetMasterMode (**BYTE** BoardNo, **BYTE** *Mode,
BYTE BlockMode)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

*Mode： [output] 0 代表預設正常模式且 1 代表監聽模式。

BlockMode： [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.12 CPM100_GetFirmwareVersion

◆ 說明：

取得 CPM100 模組的韌體版本。

◆ 語法：

WORD CPM100_GetFirmwareVersion (BYTE BoardNo, WORD
*Fir_Ver, BYTE BlockMode)

◆ 語法：

WORD CPM100_GetBoardInf (BYTE BoardNo, DWORD *dwVID,
DWORD *dwDID, DWORD
*dwSVID, DWORD *dwSDID,
DWORD *dwSAuxID, DWORD
*dwIrqNo)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

*Fir_Ver： [output] CPM100 模組韌體版本資訊。

BlockMode： [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.13 CPM100_EDS_Load

◆ 說明：

使用者呼叫此函式去掛載 EDS 檔案到主端節點來加入特定的僕端設備。使用這個函式並不會傳送任何訊息去確認僕端設備是否存在。

◆ 語法：

WORD CPM100_EDS_Load (**BYTE** BoardNo, **BYTE** Node, char
*FilePath, **WORD**
DelayTime, **WORD** ResTimeout, **BYTE**
BlockMode)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。

Node : [input] 僕端設備節點 ID(範圍: 1 到 127)。

*FilePath : [input] EDS 檔案的絕對或相對路徑。

DelayTime : [input] 這個參數定義從 CPM100 傳送的兩個訊息之間的時間區間，這樣可以避免主端太過頻繁的傳遞訊息導致僕端節點的緩衝區溢位。太大的參數值會降低 CPM100 的效能，參數的時間單位為毫秒(ms)，當完成 ESD 檔案的掛載這個參數將被施行。

ResTimeout : [input] CANopen 僕端設備回應的逾時時間。當傳送一個 CANopen 訊息到僕端設備，會等待此設備回應直到逾時時間。參數的時間單位為毫秒(ms)，當完成 ESD 檔案的掛載這個參數將被施行。

BlockMode : [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.14 CPM100_AddNode

◆ 說明：

此函式可以將 CANopen 上的特定從站 ID 加入到主端節點列表，此函式有三種模式：模式一是自動加入節點，模式二是手動加入節點，模式三是允許在開機程序時，自動加入節點。在自動模式中，呼叫完此函式並且加入從站節點後，從站設備會直接進入操作模式，然後主端會掃描 TxPDO1~TxPDO10 和 RxPDO1~RxPDO10 是否有支援的物件並安裝到 CPM100 的韌體中。手動模式中，此函式會加入一個 CANopen 的從站設備到主節點列表，並且不會傳送任何訊息到從站設備來確認是否存在。此外手動模式並不會安裝 SYNC ID、EMCY ID 和任何的 PDO 物件到 CPM100 韌體中，使用者還必須呼叫“CPM100_SetSYNC_List”、“CPM100_SetEMCY_List”和“CPM100_InstallPDO_List”進行物件安裝以完成節點加入程序。從主端節點列表移除節點需要呼叫“CPM100_RemoveNode”函式。

◆ 語法：

WORD CPM100_AddNode (**BYTE** BoardNo, **BYTE** Node, **BYTE** AddMode, **WORD** DelayTime, **WORD** ResTimeout, **BYTE** BlockMode)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。

Node : [input] 僕端設備節點 ID(範圍：1 到 127)。

AddMode : [input] 1: 自動模式，2: 手動模式，3: 等待開機模式。

DelayTime : [input] 此參數是指從 CPM100 傳送兩個訊息之間的最短時間區間。它可以避免主端在短時間內去傳送太多 CANopen 訊息導致一些節點發生錯誤。但如果延遲時間太長的話會降低 CPM100 模組的效能。此參數時間單位為毫秒(ms)。

- ResTimeout :** [input] CANopen 僕端設備回應的逾時時間。當傳送一個 CANopen 訊息到僕端設備，會等待此設備回應直到逾時時間。參數的時間單位為毫秒(ms)，當完成 ESD 檔案的掛載這個參數將被施行。
- BlockMode :** [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.15 CPM100_RemoveNode

◆ 說明：

此函式從主端節點列表移除特定僕端節點 ID，此節點 ID 必須由 CPM100_AddNode 函式成功的加入過。

◆ 語法：

WORD CPM100_RemoveNode (**BYTE** BoardNo, **BYTE** Node,
BYTE BlockMode)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。

Node : [input] 僕端設備的節點 ID(範圍：1 到 127)。

BlockMode : [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.16 CPM100_RemoveAndResetNode

◆ 說明：

此函式除了從主端移除僕端節點 ID 之外還對僕端設備進行重設，此節點 ID 必須由 CPM100_AddNode 函式成功的加入過。

◆ 語法：

WORD CPM100_RemoveAndResetNode (**BYTE** BoardNo, **BYTE** Node, **BYTE** BlockMode)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

Node： [input] 僕端設備的節點 ID(範圍：1 到 127)。

BlockMode： [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.17 CPM100_DelayAndResponseTimeout

◆ 說明：

此函式可以在任何時候更改僕端節點的 CAN 訊息最小時間區間以及回應逾時時間。

◆ 語法：

WORD CPM100_DelayAndResponseTimeout (**BYTE** BoardNo,
BYTE Node, **WORD**
DelayTime, **WORD**
ResTimeout, **BYTE**
BlockMode)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。

Node : [input] 僕端設備的節點 ID(範圍：1 到 127)。

DelayTime : [input] 此參數是指從 CPM100 傳送兩個訊息之間的最短時間區間。它可以避免主端在短時間內去傳送太多 CANopen 訊息導致一些節點發生錯誤。但如果延遲時間太長的話會降低 CPM100 模組的效能。此參數時間單位為毫秒(ms)。

ResTimeout : [input] CANopen 僕端設備回應的逾時時間。當傳送一個 CANopen 訊息到僕端設備，會等待此設備回應直到逾時時間。參數的時間單位為毫秒(ms)，這個參數將被施行當完成 ESD 檔案的掛載。

BlockMode : [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.18 CPM100_ScanNode

◆ 說明：

使用者可以使用這個函式去知道目前 CANopen 網路中有多少僕端節點。

◆ 語法：

WORD CPM100_ScanNode (**BYTE** BoardNo, **BYTE** S_Node,
BYTE E_Node, **BYTE** *NodeList,
BYTE BlockMode)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。

S_Node : [input] 起始節點 ID。

E_Node : [input] 最終結點 ID，此函式會從起始節點 ID 掃描至最終 ID。如果起始節點 ID 大於等於最終結點 ID，則此函式會從結點 1 掃描至節點 127 號。

*NodeList : [output] 這是一個 16 位元組的陣列參數，每一個位元代表的是一個節點 ID。如果值為 0 代表此節點不存在，若為 1 代表此節點存在於 CANopen 網路中。舉例如下，如果 NodeList[0] 的值為 0x16 (0001 0110)，代表節點 1、2 和 4 存在此網路中而 0、3、5、6 和 7 不存在，如果 NodeList[1] 的值為 0x41 (0100 0001)，代表節點 8 和 14 存在而 9、10、11、12、13 和 15 不存在。

BlockMode : [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.19 CPM100_GetNodeList

◆ 說明：

此函式可以知道多少僕端節點設備已被加入到 CPM100 的節點列表裡。

◆ 語法：

WORD CPM100_GetNodeList (**BYTE** BoardNo, **BYTE** *NodeList,
BYTE BlockMode)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。

*NodeList : [output] 這是一個 16 位元組的陣列參數，每一個位元代表的是一個節點 ID。如果值為 0 代表此節點不存在，若為 1 代表此節點存在於 CANopen 網路中。舉例如下，如果 NodeList[0] 的值為 0x16 (0001 0110)，代表節點 1、2 和 4 存在此網路中而 0、3、5、6 和 7 不存在，如果 NodeList[1] 的值為 0x41 (0100 0001)，代表節點 8 和 14 存在而 9、10、11、12、13 和 15 不存在。

BlockMode : [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.20 CPM100_NMTChangeState

◆ 說明：

此函式用來改變僕端設備的 NMT 狀態。

◆ 語法：

WORD CPM100_NMTChangeState (**BYTE** BoardNo, **BYTE** Node,
BYTE State, **BYTE**
BlockMode)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。

Node : [input] 僕端設備的節點 ID(範圍：1 到 127)。如果參數設為 0 代表對所有的節點設備。

State : [input] NMT 命令的定義。

1：開始

2：停止

128：進入預操作模式

129：重設節點

130：重設通訊

BlockMode : [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.21 CPM100_NMTGetState

◆ 說明：

此函式可以得知僕端節點設備目前的 NMT 狀態。

◆ 語法：

WORD CPM100_NMTGetState (**BYTE** BoardNo, **BYTE** Node,
BYTE *State, **BYTE** BlockMode)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

Node： [input] 僕端設備的節點 ID(範圍：1 到 127)。

*State： [output] 僕端設備的 NMT 狀態。

4：停止

5：操作

127：預操作

BlockMode： [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.22 CPM100_NMTGuarding

◆ 說明：

此函式可以設定特定僕端節點設備的守衛時間和生命週期因素，因此，主端設備會根據此函式的守衛時間參數自動地去對僕端設備傳送守衛訊息。如果主端沒有收到來自僕端設備對守衛訊息的確認回應，則 CPM100 將會產生一個“Node_Guarding_Event”事件給使用者，使用者可以使用“CPM100_InstallNMTErrISR”函式去安裝一個使用者自定義的程序來取得守衛失敗事件和運行守衛失敗程序。假如僕端設備沒有在節點生命週期時間(生命週期時間 = “GuardTime” * “Lift Time”)內收到守衛訊息，則它將會觸發 EMCY 訊息，建議將“LiftTime”的值設定比 1 還大。

注意！根據 CANopen 規範規定，從站節點設備是不允許共同同時使用守衛協定和心跳協定這兩種錯誤控制機制。

◆ 語法：

WORD CPM100_NMTGuarding (**BYTE** BoardNo, **BYTE** Node,
WORD GuardTime, **BYTE**
LifeTime, **BYTE** BlockMode)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。

Node : [input] 僕端設備的節點 ID(範圍：1 到 127)。

GuardTime : [input] 守衛時間(範圍: 1 到 65535 ms)。

LifeTime : [input] 生存時間因子(範圍: 1 到 255)。

BlockMode : [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.23 CPM100_NMTHeartbeat

◆ 說明：

此函式可以設定特定僕端設備的心跳時間和對 CPM100 的消耗時間，然後此僕端設備會自動的根據“ProduceTime”時間參數來對主端定時發送心跳訊息。如果主端在“ConsumeTime”時間參數內沒有從僕端接收到心跳訊息，則 CPM100 模組將會產生一個“Heartbeat_Event”事件給使用者，使用者可以使用“CPM100_InstallNMTErrISR”函式去安裝一個使用者自定義的程序來取得守衛失敗事件和運行守衛失敗程序，建議將“ConsumeTime”的值設定比“ProduceTime”的值還大。

◆ 語法：

WORD CPM100_NMTHeartbeat (**BYTE** BoardNo, **BYTE** Node,
WORD ProduceTime, **WORD**
ConsumeTime, **BYTE**
BlockMode)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

Node： [input] 僕端設備的節點 ID(範圍：1 到 127)。

ProduceTime： [input] 僕端設備的產生時間(範圍：1 到 65535 ms)。

ConsumeTime： [input] 主端設備的消耗時間(範圍：1 到 65535 ms)。

BlockMode： [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.24 CPM100_SDORReadData

◆ 說明：

此函式可對特定僕端設備進行 SDO 上傳，使用者須輸入節點 ID、物件索引和次索引參數至此函式。此函式皆支援迅速模式(小於等於 4 個 byte 的資料長度)和區段模式(大於 4 個 byte 的資料長度)。

◆ 語法：

WORD CPM100_SDORReadData (**BYTE** BoardNo, **BYTE** Node,
WORD Index, **BYTE** SubIndex,
DWORD *RDLen, **BYTE** *RData,
BYTE BlockMode)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。

Node : [input] 僕端設備的節點 ID(範圍：1 到 127)。

Index : [input] 僕端設備物件字典的索引。

SubIndex : [input] 僕端設備物件字典的次索引。

*RDLen : [output] 資料總長度。

*RData : [output] 特定僕端設備的 SDO 回應資料。

BlockMode : [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.25 CPM100_SDORReadFile

◆ 說明：

此函式用於當 SDO 上傳資料超過 1024 byte 的時候。當使用者使用“CPM100_ReadData”函式去讀取 SDO 資料且此資料長度超過 1024 個 byte，則此資料將會保存在一個檔案，使用者可以透過此函式去讀取此檔案。

◆ 語法：

WORD CPM100_SDORReadFile (**BYTE** BoardNo, **BYTE** Node,
WORD Index, **BYTE** SubIndex,
DWORD Start, **DWORD** Len,
DWORD *RLen, **BYTE** *RData)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。
Node : [input] 僕端設備的節點 ID(範圍：1 到 127)。
Index : [input] 僕端設備物件字典的索引。
SubIndex : [input] 僕端設備物件字典的次索引。
Start : [input] 從檔案讀取 SDO 資料的起始位置(範圍：0 到資料最大長度)。
Len : [input] 讀取 SDO 的資料長度。
*RLen : [output] 回傳的資料長度。
*RData : [output] 特定僕端設備的 SDO 回傳資料。

4.5.26 CPM100_SDOWriteData

◆ 說明：

此函式可以傳送一個 SDO 訊息到特定的僕端設備，此程序稱為 SDO 下載協定，此函式的節點參數代表此節點設備會收到 SDO 訊息。由於每個物件的資料長度在物件字典裡都是不同的，所以使用者必須要知道寫入特定僕端設備的物件資料長度是多少，此函式皆支援迅速模式(小於等於 4 個 byte 的資料長度)和區段模式(大於 4 個 byte 的資料長度)。

◆ 語法：

WORD CPM100_SDOWriteData (**BYTE** BoardNo, **BYTE** Node,
WORD Index, **BYTE** SubIndex,
DWORD TDLen, **BYTE** *TData,
WORD *RDLen, **BYTE** *RData,
BYTE BlockMode)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。

Node : [input] 僕端設備的節點 ID(範圍：1 到 127)。

Index : [input] 僕端設備物件字典的索引。

SubIndex : [input] 僕端設備物件字典的次索引。

TDLen : [input] 寫入資料的總長度。

*TData : [input] 寫入僕端設備的 SDO 資料。

*RDLen : [output] 回傳資料的總長度。

*RData : [output] 從僕端設備回傳的 SDO 資料。

BlockMode : [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.27 CPM100_SDOAbortTransmit

◆ 說明：

此函式用來取消 SDO 傳輸。此函式的節點參數用來明確說明哪個特定僕端設備與主端之間的 SDO 通訊。

◆ 語法：

WORD CPM100_SDOAbortTransmit (**BYTE** BoardNo, **BYTE** Node,
WORD Index, **BYTE** SubIndex,
DWORD *AData, **BYTE**
BlockMode)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

Node： [input] 僕端設備的節點 ID(範圍：1 到 127)。

Index： [input] 僕端設備物件字典的索引。

SubIndex： [input] 僕端設備物件字典的次索引。

*AData： [input] 送去僕端設備的中止資料。

BlockMode： [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.28 CPM100_PDOWrite

◆ 說明：

此函式用來傳送一個 PDO 訊息到特定的僕端設備。執行此函式之前，使用者如果要使用非預設 PDO 則需先使用“CPM100_InstallPDO”函式去安裝 PDO 物件到 CPM100 底下，然後如果僕端設備不在操作模式則使用“CPM100_NMTChangeState”函式去改變僕端設備的 NMT 狀態到操作模式，使用“offset”參數去設定 PDO 資料的起始位置，然後使用“TData”和“DLen”參數去指定資料和資料長度，如此，此函式將會從特定起始位置遵循資料長度去覆蓋僕端節點的 PDO 緩衝區，並同時透過 PDO 協定傳送資料到特定的僕端設備。

◆ 語法：

WORD CPM100_PDOWrite (**BYTE** BoardNo, **WORD** Cobid, **BYTE** Offset, **BYTE** DLen, **BYTE** *TData, **BYTE** BlockMode)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。

Cobid : [input] PDO 物件使用的 COB-ID。

Offset : [input] PDO 資料的起始位元組位置(範圍：0 到 7)。

DLen : [input] 資料長度 $dlen + offset \leq 8$ (PDO 總長度)。

*TData : [output] 資料指標指向 PDO 資料。

BlockMode : [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.29 CPM100_PDOWrite_Fast

◆ 說明：

此函式與“CPM100_PDOWrite”相似但不檢查是否將 PDO 訊息確實傳送到 CAN 的網路中，因此在高速鮑率底下(大於等於 250kbps)，此函式比“CPM100_PDOWrite”還快上兩倍。

◆ 語法：

WORD CPM100_PDOWrite_Fast (**BYTE** BoardNo, **WORD** Cobid,
BYTE Offset, **BYTE** DLen,
BYTE *TData)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。
Cobid : [input] PDO 物件使用的 COB-ID。
Offset : [input] PDO 資料的起始位元組位置(範圍：0 到 7)。
DLen : [input] 資料長度 $dlen + offset \leq 8$ (PDO 總長度)。
*TData : [output] 資料指標指向 PDO 資料。

4.5.30 CPM100_PDORemote

◆ 說明：

此函式用來傳送遠端傳輸要求(RTR)PDO 訊息到僕端設備。

◆ 語法：

WORD CPM100_PDORemote (**BYTE** BoardNo, **WORD** Cobid,
BYTE *DLen, **BYTE** *TData, **BYTE**
BlockMode)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

Cobid： [input] PDO 物件使用的 COB-ID。

*DLen： [output] RTR PDO 訊息的資料長度。

*TData： [output] 僕端設備回應的 PDO 訊息。

BlockMode： [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.31 CPM100_PDORemote_Fast

◆ 說明：

此函式傳輸 RTR PDO 訊息到僕端設備但不等待回應訊息。

◆ 語法：

WORD CPM100_PDORemote_Fast (**BYTE** BoardNo, **WORD**
Cobid)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。

Cobid : [input] PDO 物件使用的 COB-ID。

4.5.32 CPM100_SetPDORemotePolling

◆ 說明：

假如 CANopen 僕端設備不支援 TxPDOs 的事件計時器功能，則使用此函式可以配置最多 125 個 TxPDO 物件到遠端輪詢列表，接著，CPM100 模組將會輪詢已配置的 TxPDOs 物件並自動更新資料到緩衝區裡，使用者可以使用 CPM100_GetMultiPDOData 去輕易且快速地取得在緩衝區裡的 TxPDOs 資料。

◆ 語法：

WORD CPM100_SetPDORemotePolling (**BYTE** BoardNo, **BYTE** PDOCnt, **WORD** *Cobid, **BYTE** BlockMode)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。

PDOCnt : [input] *Cobid 陣列的數量。

*Cobid : [input] TxPDO 物件使用的 COB-ID 陣列。

BlockMode : [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.33 CPM100_GetPDOLastData

◆ 說明：

此函式可從 PDO 資料的緩衝區內取得最後一筆的 RxPDO 和 TxPDO 資料，最後的 PDO 資料存在 PDO 緩衝區裡，所以會與實際上的狀況不相符。

◆ 語法：

WORD CPM100_GetPDOLastData (**BYTE** BoardNo, **WORD** Cobid,
BYTE *IsNew, **BYTE** *DLen,
BYTE *RData, **BYTE** BlockMode)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。

Cobid : [input] PDO 物件使用的 COB-ID。

*IsNew : [output] 檢查資料是否已被讀取過。0 代表被讀過，1 則沒有。

*DLen : [output] PDO 訊息的資料長度。

*RData : [output] 從 PDO 緩衝區取得的 PDO 訊息。

BlockMode : [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.34 CPM100_GetPDOLastData_Fast

◆ 說明：

此函式與“CPM100_GetPDOLastData”相似且效能較好，但是不會檢查讀取的資料是否為新的。

◆ 語法：

WORD CPM100_GetPDOLastData_Fast (**BYTE** BoardNo, **WORD** Cobid, **BYTE** *DLen, **BYTE** *RData)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。
Cobid : [input] PDO 物件使用的 COB-ID。
*DLen : [output] PDO 訊息的資料長度。
* RData : [output] 從 PDO 緩衝區取得的 PDO 訊息。

4.5.35 CPM100_GetMultiPDOData

◆ 說明：

此函式與“CPM100_GetPDOLastData”一樣可取得 PDO 緩衝區資料的最後一筆 RxPDO 和 TxPDO 資料，不同點在於此函式可以同時取得最多 50 筆的 PDO 資料。

◆ 語法：

WORD CPM100_GetMultiPDOData (**BYTE** BoardNo, **BYTE** PDOCnt, **WORD** *Cobid, **BYTE** *IsNew, **BYTE** *DLen, **BYTE** *RData, **BYTE** BlockMode)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。

PDODCnt : [input] 想到取得的 PDO 總數。

*Cobid : [input] PDO 物件使用最多 50 個 COB-ID。

*IsNew : [output] 檢查資料是否已被讀取過。0 代表被讀過，1 則沒有。

*DLen : [output] 從 PDO 緩衝區取得的總資料長度。

*RData : [output] 從 PDO 緩衝區取得的 PDO 訊息。

BlockMode : [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.36 CPM100_GetMultiPDOData_Fast

◆ 說明：

此函式與“CPM100_GetMultiPDOData”函式相似且效能較好，不同點在於此函式並不會確認取得的資料是否為新的。

◆ 語法：

```
WORD CPM100_GetMultiPDOData_Fast (BYTE BoardNo,  
                                     BYTE PDOCnt, WORD  
                                     *Cobid,  BYTE *DLen,  
                                     BYTE *RData)
```

◆ 參數：

BoardNo：	[input] CPM100 板卡編號(範圍：0 到 7)。
PDOPCnt：	[input] 想到取得的 PDO 總數。
*Cobid：	[input] PDO 物件使用最多 50 個 COB-ID。
*DLen：	[output] 從 PDO 緩衝區取得的總資料長度。
*RData：	[output] 從 PDO 緩衝區取得的 PDO 訊息。

4.5.37 CPM100_GetRxPDOID

◆ 說明：

此函式可以從安裝在主端下的特定僕端設備取得所有 RxPDO COB-IDs。

◆ 語法：

WORD CPM100_GetRxPDOID (**BYTE** BoardNo, **BYTE** Node,
BYTE *PDO_Cnt, **WORD**
*ID_List, **BYTE** BlockMode)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。

Node : [input] 僕端設備的節點 ID(範圍：1 到 127)。

*PDO_Cnt : [output] 安裝 RxPDO 的總數。

*ID_List : [output] RxPDO COB-ID 列表。

BlockMode : [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.38 CPM100_GetTxPDOID

◆ 說明：

此函式可以從安裝在主端下的特定僕端設備取得所有 TxPDO COB-IDs。

◆ 語法：

WORD CPM100_GetTxPDOID (**BYTE** BoardNo, **BYTE** Node,
BYTE *PDO_Cnt, **WORD** *ID_List,
BYTE BlockMode)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

Node： [input] 僕端設備的節點 ID(範圍：1 到 127)。

*PDO_Cnt： [output] 安裝 TxPDO 的總數。

*ID_List： [output] TxPDO COB-ID 列表。

BlockMode： [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.39 CPM100_InstallPDO

◆ 說明：

呼叫此函式之後，一個 PDO COB-ID 將會被安裝在 CANopen 主端函式庫堆疊的 PDO 物件列表底下。假設僕端設備有被定義預設的 PDO 物件在 RxPDO1 到 RxPDO10 和 TxPDO1 到 TxPDO10 的情況下，則使用“CPM100_AddNode”函式的自動模式會自動的安裝這些 PDO 物件，否則需要呼叫此函式來進行手動安裝。

◆ 語法：

WORD CPM100_InstallPDO (**BYTE** BoardNo, **BYTE** Node,
WORD Cobid, **BYTE** RxTx, **WORD**
PDO_No, **BYTE** BlockMode)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。

Node : [input] 僕端設備的節點 ID(範圍：1 到 127)。

Cobid : [input] PDO 物件使用的 COB-ID。

RxTx : [input] PDO 種類。0 代表 RxPDO；1 代表 TxPDO。

PDO_No : [input] PDO 映射物件編號(範圍: 1 到 512)。

BlockMode : [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.40 CPM100_DynamicPDO

◆ 說明：

此函式可以修改在 CANopen 主端函式庫堆疊之 PDO 物件列表的 PDO 物件映射資料。注意！呼叫此函式之前請先確認此 PDO 物件是否已被安裝在 CPM100 主端底下。

◆ 語法：

WORD CPM100_DynamicPDO (**BYTE** BoardNo, **BYTE** Node,
WORD Cobid, **BYTE** RxTx, **BYTE** Entry,
DWORD EntryData, **BYTE** BlockMode)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。

Node : [input] 僕端設備的節點 ID(範圍：1 到 127)。

Cobid : [input] PDO 物件使用的 COB-ID。

RxTx : [input] PDO 種類。0 代表 RxPDO；1 代表 TxPDO。

Entry : [input] PDO 映射物件的次索引值(範圍：1 到 8)。

EntryData : [input] 一個 4 byte 的映射應用物件資訊。使用者需要查詢 CANopen 從站設備手冊物件資訊，此參數的格式如下，EntryData = 0x64110310：主索引代表 0x6411，子索引為 0x03，資料長度為 0x10 bits 也就是 2 bytes。假設函式的參數為 Cobid = 0x333、RxTx = 0、Entry = 2 和 EntryData = 0x64110310，這個範例對應到 16 bit 資料，COB-ID 為 0x333 的 RxPDO 第二個項目的 0x6411 索引和 0x03 子索引值物件。

BlockMode : [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.41 CPM100_RemovePDO

◆ 說明：

此函式可移除已透過“CPM100_InstallPDO”或“CPM100_AddNode”安裝的 TxPDO 或 RxPDO 物件，也可移除映射到 TxPDO 或 RxPDO 的單一物件。

◆ 語法：

WORD CPM100_RemovePDO (**BYTE** BoardNo, **BYTE** Node,
WORD Cobid, **BYTE** Entry,
BYTE BlockMode)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。

Node : [input] 僕端設備的節點 ID(範圍：1 到 127)。

Cobid : [input] PDO 物件使用的 COB-ID。

Entry : [input] PDO 映射物件的次索引值(範圍：0 到 8)。如果設為 0，則特定的 PDO 物件將被移除，如果設定 1 到 8 之間，則特定的 PDO 物件次索引內容將被移除。

BlockMode : [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.42 CPM100_ChangePDOID

◆ 說明：

此函式可以將僕端設備從舊的 COB-ID 改變成新的 COB-ID。

◆ 語法：

WORD CPM100_ChangePDOID (**BYTE** BoardNo, **WORD**
Old_Cobid, **WORD** New_Cobid,
BYTE BlockMode)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

Old_Cobid： [input] PDO 物件使用舊的 COB-ID。

New_Cobid： [input] PDO 物件使用新的 COB-ID。

BlockMode： [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.43 CPM100_GetPDOMapInfo

◆ 說明：

此函式可以取得 PDO 物件映射資料的資訊。

◆ 語法：

```
WORD CPM100_GetPDOMapInfo (BYTE BoardNo, WORD Cobid,  
                             BYTE *RxTx, BYTE *Tx_Type,  
                             WORD *Event_Timer, BYTE  
                             *Entry_Cnt, DWORD *Map_Data,  
                             BYTE BlockMode)
```

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

Cobid： [input] PDO 物件使用的 COB-ID。

*RxTx： [output] PDO 種類。0 代表 RxPDO；1 代表 TxPDO。

*Tx_Type： [output] 傳輸種類。

*Event_Timer： [output] PDO 事件計時器。

*Entry_Cnt： [output] PDO 物件的有效項目數量。

*Map_Data： [output] 一個 4 byte 的陣列參數。回傳 PDO 物件每個有效項目的映射資料。

BlockMode： [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.44 CPM100_InstallPDO_List

◆ 說明：

此函式與“CPM100_InstallPDO”函式相似，它可以安裝舊的或新的 PDO 物件在 CPM100 主端的 PDO 物件列表，但是它並沒有傳送任何訊息去確認這個 PDO 物件是否真實存在僕端設備上，它只有改變 CPM100 記憶體裡的列表，這也代表使用者可以使用此函式來任意的加入或改變 PDO 映射資料而不會影響到 CANopen 網路。在呼叫此函式之後，CPM100 將會運行與此函式配置相同 IDs 的僕端 PDOs 物件，當 CPM100 運行在監聽模式之下這是非常有用的。使用者可以使用 “CPM100_RemovePDO_List” 函式去移除 “CPM100_InstallPDO_List” 函式所有加入的 PDO 物件。

◆ 語法：

WORD CPM100_InstallPDO_List (**BYTE** BoardNo, **BYTE** Node, **WORD** Cobid, **BYTE** RxTx, **WORD** PDO_No, **BYTE** EntryUse, **DWORD** *EntryData, **BYTE** BlockMode)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。

Node : [input] 僕端設備的節點 ID(範圍：1 到 127)。如果 “Node” 參數是 CPM100 的節點 ID，則 “EntryUse” 和 “EntryData” 參數將是無用的，在此情況下，使用者可以使用 “CPM100_InstallRxPDOISR” 或者 “CPM100_InstallRemotePDOISR” 去安裝使用者的回呼函式去運行接收到 CPM100 的 PDOs 物件。

Cobid : [input] PDO 物件使用的 COB-ID。

RxTx : [input] PDO 種類。0 代表 RxPDO；1 代表 TxPDO。

PDO_No : [input] PDO 映射物件編號(範圍：1 到 512)。

EntryUse : [input] 將被安裝映射物件的總項目。

***EntryData :** [input] 映射應用物件的 4 byte 陣列資訊。舉例如下，如果配置成 Cobid = 0x333、RxTx = 0、PDO_No = 10、Entry = 2、EntryData[0] = 0x64110310 和 EntryData[1] = 0x62000108，它將會映射 COB-ID 0x333 的 RxPDO10 物件，且第一個項目在主索引 0x6411 和子索引 0x03 物件的 16 bit 資料，和第二個項目在主索引 0x6200 和次索引 0x01 物件的 16 byte 資料

BlockMode : [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.45 CPM100_RemovePDO_List

◆ 說明：

此函式可移除已被“CPM100_InstallPDO_List”安裝的 TxPDO 或 RxPDO 物件，此函式也可以移除映射在 TxPDO 或 RxPDO 的單一物件。

◆ 語法：

WORD CPM100_RemovePDO_List (**BYTE** BoardNo, **BYTE** Node,
WORD Cobid, **BYTE** Entry,
BYTE BlockMode)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。

Node : [input] 僕端設備的節點 ID(範圍：1 到 127)。

Cobid : [input] PDO 物件使用的 COB-ID。

Entry : [input] PDO 映射物件的次索引值(範圍：0 到 8)。如果參數設為 0，則特定的 PDO 物件將被移除，如果介於 1 到 8 之間，則特定 PDO 物件的次索引內容將被移除。

BlockMode : [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.46 CPM100_PDUseEntry

◆ 說明：

此函式可以改變 PDO 物件的有效物件映射項目。有效項目從 1 開始至“Entry”參數，因此，如果“Entry”參數為 0 代表 PDO 物件沒有有效的物件映射項目。

◆ 語法：

WORD CPM100_PDUseEntry (**BYTE** BoardNo, **WORD** Cobid,
BYTE Entry, **BYTE** BlockMode)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

Cobid： [input] PDO 物件使用的 COB-ID。

Entry： [input] PDO 映射物件的有效項目數量。

BlockMode： [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.47 CPM100_PDOTxType

◆ 說明：

此函式用來改變 TxPDO 的傳輸種類。預設傳輸型態為 255。

◆ 語法：

WORD CPM100_PDOTxType (**BYTE** BoardNo, **WORD** Cobid,
BYTE Tx_Type, **BYTE** BlockMode)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

Cobid： [input] PDO 物件使用的 COB-ID。

Tx_Type： [input] TxPDO 的傳輸種類(範圍：0 到 255)。如下表所示。

傳輸種類	PDO 傳輸				
	循環	非循環	同步	非同步	遠端傳送要求
0		X	X		
1-240	X		X		
241-251	保留				
252			X		X
253				X	
254				X	
255				X	

BlockMode： [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.48 CPM100_PDOWEventTimer

◆ 說明：

此函式用來改變 TxPDO 物件的事件計時器，預設事件計時器為 0。
當僕端設備 PDO 物件的事件計時器比 0 還大，則 PDO 訊息會根據
“Timer”參數自動的傳到主端。

◆ 語法：

WORD CPM100_PDOWEventTimer (**BYTE** BoardNo, **WORD** Cobid,
WORD Timer, **BYTE** BlockMode)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

Cobid： [input] PDO 物件使用的 COB-ID。

Timer： [input] TxPDO 的事件計時器。單位為毫秒。

BlockMode： [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數
設為 1 時，使用者的應用程序將被延遲直到此函式回
傳，如果設為 0，則此函式會直接回傳
“CPM_Processing”的訊息。如果使用者輸入相同的
“BoardNo”和“Node”的值並再次呼叫此函式，如果程
序還未完成則回傳“CPM_Wait”。

4.5.49 CPM100_PDOLInhibitTime

◆ 說明：

此函式用來設定 TxPDO 的抑制時間，此時間是 PDO 傳輸的最小時間區間。

◆ 語法：

WORD CPM100_PDOLInhibitTime (**BYTE** BoardNo, **WORD** Cobid, **WORD** Time, **BYTE** BlockMode)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

Cobid： [input] PDO 物件使用的 COB-ID。

Time： [input] TxPDO 的抑制時間。

BlockMode： [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.50 CPM100_ChangeSYNCID

◆ 說明：

此函式用來更改僕端設備的 SYNC COB-ID。

◆ 語法：

WORD CPM100_ChangeSYNCID (**BYTE** BoardNo, **BYTE** Node,
WORD Cobid, **BYTE** BlockMode)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

Node： [input] 僕端設備的節點 ID(範圍：1 到 127)。

Cobid： [input] SYNC 物件使用的 COB-ID。

BlockMode： [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.51 CPM100_SetSYNC_List

◆ 說明：

如果使用者使用“CPM100_AddNode”函式的手動模式來加入僕端節點，則當僕端設備的 SYNC ID 需要被改變或設定的時候，必須呼叫此函式。此函式只能改變在 CPM100 COB-ID 列表裡的 SYNC COB-ID，真實存在僕端設備裡的值與此函式配置的值有可能不同，使用者需自行確認。

◆ 語法：

WORD CPM100_SetSYNC_List (**BYTE** BoardNo, **BYTE** Node, **WORD** Cobid, **BYTE** BlockMode)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。

Node : [input] 僕端設備的節點 ID(範圍：1 到 127)。

Cobid : [input] SYNC 物件使用的 COB-ID。

BlockMode : [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.52 CPM100_GetSYNCID

◆ 說明：

此函式可以取得 CPM100 COB-ID 列表裡的 SYNC ID。

◆ 語法：

WORD CPM100_GetSYNCID (**BYTE** BoardNo, **BYTE** Node,
WORD *Cobid, **BYTE** BlockMode)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

Node： [input] 僕端設備的節點 ID(範圍：1 到 127)。

*Cobid： [output] 回傳 SYNC 物件使用的 COB-ID。

BlockMode： [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.53 CPM100_SendSYNCMsg

◆ 說明：

此函式可以循環的傳送一個帶有特定 COB-ID 的 SYNC 訊息。如果“Timer”參數為 0，則 SYNC 訊息將被停止，如果比 0 還來的大，此函式會根據每隔參數“Timer”毫秒的時間單位來直到完成傳送 SYNC 訊息的“Times”參數的次數。當參數“Times”設為 0 時，此函式會持續傳送 SYNC 訊息直到“Timer”為 0 時，使用者可以設定最多 5 個循環 SYNC 訊息搭配不同的 ID。

◆ 語法：

WORD CPM100_SendSYNCMsg (**BYTE** BoardNo, **WORD** Cobid,
WORD Timer, **DWORD** Times,
BYTE BlockMode)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。

Cobid : [input] SYNC 物件使用的 COB-ID。

Timer : [input] SYNC 訊息的傳輸週期。如果設為 0 則 SYNC 訊息將會停止。

Times : [input] SYNC 訊息的傳輸次數。如果設為 0，SYNC 訊息會一直傳送直到“Timer”被設為 0。

BlockMode : [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.54 CPM100_GetCyclicSYNCInfo

◆ 說明：

此函式可以取得最多 5 個被“CPM100_SendSYNCMsg”函式所配置的 SYNC 訊息資訊，使用者可以知道哪些 SYNC ID 被設置。

◆ 語法：

WORD CPM100_GetCyclicSYNCInfo (**BYTE** BoardNo, **WORD**
*Cobid, **WORD** *Timer, **DWORD**
*Times, **BYTE** BlockMode)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。

*Cobid : [input] COB-ID 陣列。最多回傳 5 個 SYNC ID。

*Timer : [input] 5 個 WORD 陣列。每一個值代表 SYNC 訊息的循環週期。

*Times : [input] 5 個 Double WORD 陣列。每一個值代表 SYNC 訊息之前設定的傳輸次數。

BlockMode : [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.55 CPM100_ChangeEMCYID

◆ 說明：

此函式用來改變特定僕端設備的 EMCY COB-ID。

◆ 語法：

WORD CPM100_ChangeEMCYID (**BYTE** BoardNo, **BYTE** Node,
WORD Cobid, **BYTE** BlockMode)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

Node： [input] 僕端設備的節點 ID(範圍：1 到 127)。

Cobid： [input] EMCY 物件使用的 COB-ID。

BlockMode： [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.56 CPM100_SetEMCY_List

◆ 說明：

如果使用者使用“CPM100_AddNode”函式的手動模式，當要更改或設定 EMCY ID 時，則此函式必須被呼叫到。此函式可以改變 CPM100 COB-ID 列表裡的 EMCY COB-ID，也就是 CPM100 可對由此函式配置特定的 EMCY COB-ID 進行 EMCY 訊息。

◆ 語法：

WORD CPM100_SetEMCY_List (**BYTE** BoardNo, **BYTE** Node,
WORD Cobid, **BYTE** BlockMode)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

Node： [input] 僕端設備的節點 ID(範圍：1 到 127)。

Cobid： [input] EMCY 物件使用的 COB-ID。

BlockMode： [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.57 CPM100_GetEMCYID

◆ 說明：

此函式可以取得 CPM100 COB-ID 列表裡的 EMCY ID。

◆ 語法：

WORD CPM100_GetEMCYID (**BYTE** BoardNo, **BYTE** Node, **WORD**
*Cobid, **BYTE** BlockMode)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

Node： [input] 僕端設備的節點 ID(範圍：1 到 127)。

*Cobid： [output] 回傳 EMCY 物件使用的 COB-ID。

BlockMode： [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.58 CPM100_ReadLastEMCY

◆ 說明：

此函式可以檢查某一僕端設備是否有產生過 EMCY，如果有，則此函式將回傳此僕端設備最後的 EMCY 訊息。

◆ 語法：

WORD CPM100_ReadLastEMCY (**BYTE** BoardNo, **BYTE** Node,
BYTE *IsNew, **BYTE** *RData,
BYTE BlockMode)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

Node： [input] 僕端設備的節點 ID(範圍：1 到 127)。

*IsNew： [output] 檢查資料是否有被讀取過。0 代表被讀取過，1 代表是新的。

*RData： [output] 從 EMCY 緩衝區取得 8 byte EMCY 訊息。

BlockMode： [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.59 CPM100_GetBootUpNodeAfterAdd

◆ 說明：

如果使用者不知道哪一個僕端節點發生開機訊息和哪一個 CPM100 接收它，使用者可以使用此函式。此函式不但可取得僕端節點 ID 還可以取得 CPM100 插槽總數，參數“BoardNo”代表接收開機訊息的 CPM100 編號，參數“Node”代表產生開機訊息的僕端節點 ID。此函式通常與“CPM100_InstallBootUpISR”一起使用，但請注意，此函式只能取得由被“CAN100_AddNode”加入過主端的僕端節點 ID。

◆ 語法：

```
WORD CPM100_GetBootUpNodeAfterAdd (BYTE *BoardNo,  
                                     BYTE *Node)
```

◆ 參數：

BoardNo : [input] 接收開機訊息的 CPM100 板號。

Node : [input] 接收開機訊息的僕端節點 ID。

4.5.60 CPM100_GetEMCYData

◆ 說明：

如果使用者不知道哪個節點發生 EMCY 訊息和哪個 CPM100 模組接收它，則使用者可以使用此函式，此函式不只可以取得特定僕端節點 ID 的 EMCY 訊息，還可以取得 CPM100 模組的板號。參數“BoardNo”表示接收 EMCY 訊息的 CPM100 模組編號，參數“Node”是產生 EMCY 訊息的僕端節點 ID，參數“*RData*”是一個 8 byte 的 EMCY 訊息資料，此函式通常搭配“CPM100_InstallEMCYISR”使用，關於範例請參考 4.1.2 NMT_Protocol 章節。

◆ 語法：

WORD CPM100_GetEMCYData (**BYTE** *BoardNo, **BYTE** *Node,
BYTE *RData)

◆ 參數：

*BoardNo : [output] CPM100 板卡編號(範圍：0 到 7)。
*Node : [output] 接收 EMCY 訊息的僕端節點 ID。
*RData : [output] 從 EMCY 緩衝區取得 8 byte 的 EMCY 訊息。

4.5.61 CPM100_GetNMTErr

◆ 說明：

此函式用來檢查 CPM100 模組是否從任意僕端節點拿到 NMT 錯誤事件，函式參數有哪個 CPM100 模組取得 NMT 錯誤事件、哪個節點產生此事件和這個事件是什麼種類。參數“BoardNo”代表顯示 Heartbeat_Event 或 Node_Guarding_Event 的 CPM100 模組板號，參數“Node”代表回應心跳協定或守衛協定的僕端節點 ID，NMTErr 代表 NMTErr 事件模式，如果 NMTErr 事件是 Node_Guarding_Event，則 NMTErr 參數是取得 CPM_Node_Guarding_Event 或 CPM_Heartbeat_Event。此函是通常伴隨“CPM100_InstallNMTErrISR”，關於範例請參考 4.1.2 NMT_Protocol 章節。

◆ 語法：

WORD CPM100_GetNMTErr (**BYTE** *BoardNo, **BYTE** *Node,
BYTE *NMTErr)

◆ 參數：

*BoardNo： [output] CPM100 板卡編號(範圍：0 到 7)。
*Node： [output] 取得 NMT 錯誤事件的僕端節點 ID。
*NMTErr： [output] 取得 CPM_Node_Guarding_Event 的值代表節點守衛事件，取得 CPM_Heartbeat_Event 的值代表心跳事件。

4.5.62 CPM100_InstallBootUpISR

◆ 說明：

此函式允許使用者使用僕端開機 ISR(中斷服務程序，interrupt service routine)，當使用者輸入開機程序到此函式，則所有僕端設備觸發開機時都會進入開機 ISR。如果已被 CPM100 加入的僕端設備發生開機訊息，則 CPM100 將會進入開機程序中並遵守使用者開機程序去執行特定機制。

◆ 語法：

WORD CPM100_InstallBootUpISR (**BYTE** BoardNo, **void**
(*BOOTISR)())

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

(*BOOTISR)()： [input] 指標指向“void XXX()”格式的函式。“XXX”代表使用者開機程序的函式名稱，此程序通常搭配“CPM100_GetBootUpNodeAfterAdd”函式使用。

4.5.63 CPM100_RemoveBootUpISR

◆ 說明：

如果使用者不需要開機 ISR 函式，可以呼叫此函式去移除使用者 ISR。

◆ 語法：

WORD CPM100_RemoveBootUpISR (**BYTE** BoardNo)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

4.5.64 CPM100_InstallEMCYISR

◆ 說明：

此函式允許使用者使用 EMCY ISR。當使用者輸入 EMCY 程序到此函式，則所有驅動 EMCY 的僕端設備都將進入 EMCY ISR，如果僕端設備發生 EMCY，則 CPM100 會進入 EMCY 程序並遵守使用者的 EMCY 程序來執行一些安全機制。

◆ 語法：

WORD CPM100_InstallEMCYISR (**BYTE** BoardNo, **void**
(*EMCYISR)())

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

(*EMCYISR)()： [input] 指標指向“void XXX()”格式的函式。“XXX”代表使用者 EMCY 程序的函式名稱，此程序通常搭配“CPM100_GetEMCYData”函式使用。

4.5.65 CPM100_RemoveEMCYISR

◆ 說明：

如果使用者不需要 EMCY ISR 函式，可以呼叫此函式去移除使用者 ISR。

◆ 語法：

WORD CPM100_RemoveEMCYISR (**BYTE** BoardNo)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

4.5.66 CPM100_InstallNMTErrISR

◆ 說明：

此函式允許使用者使用 NMTErr ISR，當使用者輸入 NMTErr 程序到此函式，則所有被僕端觸發 Heartbeat_Event 和 Node_Guarding_Event 的設備，都將進入 ISR。如果使用者曾使用“CPM100_NMTGuarding”函式去開啟守衛協定或是“CPM100_Heartbeat”去開啟心跳協定，則當 CPM100 模組沒有收到守衛確認或是心跳訊息，CPM100 模組會進入 NMTErr ISR 去執行使用者 NMTErr 程序。

◆ 語法：

WORD CPM100_InstallNMTErrISR (**BYTE** BoardNo, **void**
(*NMTErrISR)())

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。
(*NMTErrISR)() : [input] 指標指向“void XXX()”格式的函式。“XXX”代表使用者程序的函式名稱，此程序通常搭配“CPM100_GetNMTErr”函式使用。

4.5.67 CPM100_RemoveNMTErrISR

◆ 說明：

如果使用者不需要 NMTErr ISR 函式，可以呼叫此函式去移除使用者 ISR。

◆ 語法：

WORD CPM100_RemoveNMTErrISR (**BYTE** BoardNo)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

4.5.68 CPM100_GetMasterReadSDOEvent

◆ 說明：

此函式可以取得 CPM100 模組送去特定節點 ID 的 SDO 讀取訊息。舉例說明當 CPM100 節點 ID 為 2，如果傳送一個 COB-ID 等於 0x602 的 SDO 訊息到 CPM100 模組來讀取物件，使用者可以使用此函式來獲取 SDO 訊息和回應一些資訊到 SDO 發送端。參數“BoardNo”代表接收讀取 SDO 訊息的 CPM100 編號，“Index”和“SubIndex”參數則指向物件資料。此函式通常搭配“CPM100_InstallReadSDOISR”函式，關於範例請參考 4.1.6 SDO_PDO_ISR 章節。

注意！當“CPM100_InitMaster”函式中的“Node”參數大於 0，則此函式才有效。

◆ 語法：

WORD CPM100_GetMasterReadSDOEvent (**BYTE** *BoardNo,
WORD *Index, **BYTE** *SubIndex)

◆ 參數：

BoardNo： [output] CPM100 板卡編號(範圍：0 到 7)。

*Index： [output] 取得 SDO 訊息的物件索引。

*SubIndex： [output] 取得 SDO 訊息的物件副索引。

4.5.69 CPM100_GetMasterWriteSDOEvent

◆ 說明：

此函式可以取得 CPM100 模組送去特定節點 ID 的 SDO 寫入訊息。舉例說明當 CPM100 節點 ID 為 2，如果傳送一個 COB-ID 等於 0x602 的 SDO 訊息到 CPM100 模組來寫入物件，使用者可以使用此函式來獲取 SDO 訊息。參數“BoardNo”代表接收寫入 SDO 訊息的 CPM100 編號，“Index”和“SubIndex”參數則指向物件資料，參數“WLen”是參數“*WData”的資料長度。此函式通常搭配“CPM100_InstallWriteSDOISR”函式，關於範例請參考 4.1.6 SDO_PDO_ISR 章節。

注意！當“CPM100_InitMaster”函式中的“Node”參數大於 0，則此函式才有效。

◆ 語法：

WORD CPM100_GetMasterWriteSDOEvent (**BYTE** *BoardNo,
 WORD *Index, **BYTE** *SubIndex,
 BYTE *WLen, **BYTE** *WData)

◆ 參數：

- BoardNo : [output] CPM100 板卡編號(範圍：0 到 7)。
- *Index : [output] 取得 SDO 訊息的物件索引。
- *SubIndex : [output] 取得 SDO 訊息的物件副索引。
- *WLen : [output] 寫入資料的長度。
- *WData : [output] 回傳 0 到 4 個 byte 的 SDO 寫入資料。

4.5.70 CPM100_ResponseMasterSDO

◆ 說明：

此函式可以對 SDO 發送端回覆 SDO 訊息。舉例說明當 CPM100 節點 ID 為 2，如果傳送一個 COB-ID 等於 0x602 的 SDO 訊息到 CPM100 模組來讀取或寫入物件，則 CPM100 模組需要回覆相應的 SDO 訊息，此時可以使用此函式，當執行此函式後，CPM100 模組將會傳送一個帶有 COB-ID 為 0x582 的 SDO 訊息到 CANopen 網路中。此函式通常搭配 SDO_ISR 系列函式，關於範例請參考 4.1.6 SDO_PDO_ISR 章節。

注意！當“CPM100_InitMaster”函式中的“Node”參數大於 0，則此函式才有效。

注意！當 CPM100 想要回覆 SDO 中止訊息，請改用“CPM100_SDOAbortTransmit”實作(請參考 3.5.20 章節)。

◆ 語法：

WORD CPM100_ResponseMasterSDO (**BYTE** BoardNo, **BYTE** ResType, **WORD** Index, **BYTE** SubIndex, **BYTE** Len, **BYTE** *Data)

◆ 參數：

- BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。
- ResType : [input] SDO 訊息的回覆種類。0 代表回覆 SDO 讀取訊息，1 代表 SDO 寫入訊息。
- Index : [input] 僕端設備物件字典的主索引。
- SubIndex : [input] 僕端設備物件字典的次索引。
- Len : [input] 回應資料長度。當 ResType 為 1(寫入型態)時，“Len”和“*Data”參數都是無用的。
- *Data : [input] 回傳 0 到 4 個 byte 的 SDO 回應資料。

4.5.71 CPM100_InstallReadSDOISR

◆ 說明：

此函式允許使用者使用 CPM100 模組的 ReadSDO ISR。當使用者輸入 SDO 讀取程序到此函式時，所有傳送到特定 CPM100 的 SDO 讀取訊息將會觸發 ISR。舉例如下，當 CPM100 初始化為節點 ID 為 2 時，如果傳送一個 COB-ID 為 0x602 的 SDO 訊息來讀取 CPM100 模組的物件資料，則 CPM100 將會進入 ISR。

注意！當“CPM100_InitMaster”函式中的“Node”參數大於 0，則此函式才有效。

◆ 語法：

WORD CPM100_InstallReadSDOISR (**BYTE** BoardNo, **void**
(*RSDOISR)())

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

(*RSDOISR)()： [input] 指標指向“void XXX()”格式的函式。“XXX”代表使用者程序的函式名稱，此程序通常搭配 “CPM100_GetMasterReadSDOEvent” 函式使用。

4.5.72 CPM100_RemoveReadSDOISR

◆ 說明：

當使用者不需要 ReadSDO ISR 功能，則呼叫此函式移除使用者 ISR。

◆ 語法：

WORD CPM100_RemoveReadSDOISR (**BYTE** BoardNo)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

4.5.73 CPM100_InstallWriteSDOISR

◆ 說明：

此函式允許使用者使用 CPM100 模組的 WriteSDO ISR。當使用者輸入 SDO 讀取程序到此函式時，所有傳送到特定 CPM100 的 SDO 寫入訊息將會觸發 ISR。舉例如下，當 CPM100 初始化為節點 ID 為 2 時，如果傳送一個 COB-ID 為 0x602 的 SDO 訊息來寫入 CPM100 模組的物件資料，則 CPM100 將會進入 ISR。

注意！當“CPM100_InitMaster”函式中的“Node”參數大於 0，則此函式才有效。

◆ 語法：

WORD CPM100_InstallWriteSDOISR (**BYTE** BoardNo, **void**
(*WSDOISR)())

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

(*WSDOISR)()： [input] 指標指向“void XXX()”格式的函式。“XXX”代表使用者程序的函式名稱，此程序通常搭配“CPM100_GetMasterWriteSDOEvent”函式使用。

4.5.74 CPM100_RemoveWriteSDOISR

◆ 說明：

當使用者不需要 WriteSDO ISR 功能，則呼叫此函式移除使用者 ISR。

◆ 語法：

WORD CPM100_RemoveWriteSDOISR (**BYTE** BoardNo)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

4.5.75 CPM100_GetMasterRemotePDOEvent

◆ 說明：

使用此函式來取得所有傳送到 CPM100 模組的遠端 PDO 訊息。舉例說明如下，假設 CPM100 模組已被“CPM100_InstallPDO_List”函式安裝一個 COB-ID 為 0x444 的 TxPDO 物件，如果傳送一個遠端 COB-ID 0x444 的 PDO 訊息到 CPM100，則使用者可以使用此函式來取得 PDO 訊息。參數“BoardNo”代表接收遠端 PDO 訊息的 CPM100 編號，參數“CobId”是傳送訊息到 CPM100 的 PDO COB-ID。此函式通常搭配“CPM100_InstallRemotePDOISR”使用，關於範例請參考 4.1.6 SDO_PDO_ISR 章節

注意！當“CPM100_InitMaster”函式中的“Node”參數大於 0，則此函式才有效。

◆ 語法：

WORD CPM100_GetMasterRemotePDOEvent (**BYTE** *BoardNo,
WORD *CobId)

◆ 參數：

BoardNo： [output] 接收遠端 PDO 訊息的 CPM100 板卡編號
(範圍：0 到 7)。

*CobId： [output] 回傳遠端 PDO 訊息的 COB-ID。

4.5.76 CPM100_GetMasterRxPDOEvent

◆ 說明：

此函式可以取得所有傳送到 CPM100 模組的 RxPDO 訊息。舉例說明，假設 CPM100 已被“CPM100_InstallPDO_List”函式安裝過一個 COB-ID 為 0x333 的 RxPDO 物件，如果傳送一個 COB-ID 0x333 的 RxPDO 訊息到 CPM100 模組，使用者可以使用此函式來取得這個 RxPDO 訊息。參數“BoardNo”代表接收接收 RxPDO 訊息的 CPM100 編號，“CobId”參數是 RxPDO 的 COB-ID，參數“WLen”是參數“*WData”的資料長度，而“*WData”參數是 RxPDO 訊息裡的內容。此函式通常搭配“CPM100_InstallRxPDOISR”函式，關於範例請參考 4.1.6 SDO_PDO_ISR 章節。

注意！當“CPM100_InitMaster”函式中的“Node”參數大於 0，則此函式才有效。

◆ 語法：

WORD CPM100_GetMasterRxPDOEvent (**BYTE** *BoardNo, **WORD**
*CobId, **BYTE** *WLen,
BYTE *WData)

◆ 參數：

BoardNo： [output] 接收 RxPDO 訊息的 CPM100 板卡編號(範圍：0 到 7)。

*CobId： [output] 回傳 RxPDO COB-ID。

*WLen： [output] RxPDO 的資料長度。

*WData： [output] 回傳 0 到 4 個 byte 的 RxPDO 資料。

4.5.77 CPM100_ResponseMasterPDO

◆ 說明：

此函式可以回覆遠端 PDO 訊息到發送端。舉例說明，假設 CPM100 已被“CPM100_InstallPDO_List”函式安裝過一個 COB-ID 為 0x444 的 TxPDO 物件，如果傳送一個 COB-ID 0x444 的遠端 PDO 訊息到 CPM100 模組，而 CPM100 需要回覆一個 TxPDO 訊息，使用者可以使用此函式來實作。當使用者執行這個函式時，CPM100 將會傳送一個 TxPDO 訊息到 CANopen 網路中。此函式通常搭配“CPM100_InstallRemotePDOISR”函式，關於範例請參考 4.1.6 SDO_PDO_ISR 章節。

注意！當“CPM100_InitMaster”函式中的“Node”參數大於 0，則此函式才有效。

◆ 語法：

WORD CPM100_InstallRemotePDOISR (**BYTE** BoardNo, **WORD** CobId, **BYTE** Len, **BYTE** *Data)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。
CobId : [input] 回應 PDO 訊息的 TxPDO COB-ID。
Len : [input] 回覆資料的長度。
*Data : [input] 回傳 TxPDO COB-ID 的 PDO 訊息。

4.5.78 CPM100_InstallRxPDOISR

◆ 說明：

此函式允許使用者使用 CPM100 模組的 RxPDO ISR，當使用者輸入程序到此函式中，所有 CPM100 PDO 物件的 RxPDO 訊息將會觸發 ISR。舉例說明，假設 CPM100 已被“CPM100_InstallPDO_List”函式安裝過一個 COB-ID 為 0x333 的 PDO 物件，如果傳送一個 COB-ID 0x333 的 PDO 訊息到 CPM100 模組，則 CPM100 模組將會進入 ISR。

注意！當“CPM100_InitMaster”函式中的“Node”參數大於 0，則此函式才有效。

◆ 語法：

WORD CPM100_InstallRxPDOISR (**BYTE** BoardNo,
void(*RXPDOISR)())

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

(*RXPDOISR)()： [input] 指標指向“void XXX()”格式的函式。“XXX”代表使用者程序的函式名稱，此程序通常搭配“CPM100_GetMasterRxPDODoEvent”函式使用。

4.5.79 CPM100_RemoveRxPDOISR

◆ 說明：

當使用者不需要 RxPDO ISR 功能，則呼叫此函式移除使用者 ISR。

◆ 語法：

WORD CPM100_RemoveRxPDOISR (**BYTE** BoardNo)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

4.5.80 CPM100_InstallRemotePDOISR

◆ 說明：

此函式允許使用者使用 CPM100 模組的遠端 PDO ISR，當使用者輸入程序到此函式中，所有 CPM100 PDO 物件的遠端 PDO 訊息將會觸發 ISR。舉例說明，假設 CPM100 已被“CPM100_InstallPDO_List”函式安裝過一個 COB-ID 為 0x444 的 TxPDO 物件，如果傳送一個 COB-ID 0x333 的遠端 PDO 訊息到 CPM100 模組，則 CPM100 模組將會進入 ISR。

注意！當“CPM100_InitMaster”函式中的“Node”參數大於 0，則此函式才有效。

◆ 語法：

WORD CPM100_InstallRemotePDOISR (**BYTE** BoardNo,
void(*REMOTEPDOISR)())

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。
(*REMOTEPDOISR)() : [input] 指標指向“void XXX()”格式的函式。“XXX”代表使用者程序的函式名稱，此程序通常搭配“CPM100_GetMasterRemotePDOEvent”函式使用。

4.5.81 CPM100_RemoveRemotePDOISR

◆ 說明：

當使用者不需要遠端 PDO ISR 功能，則呼叫此函式移除使用者 ISR。

◆ 語法：

WORD CPM100_RemoveRemotePDOISR (**BYTE** BoardNo)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

4.5.82 CPM100_SyncGroupCycUpdata

◆ 說明：

使用“CPM100_SyncGroupCycUpdata”函式去傳送 SYNC 和 PDO 群組資料之前，使用者需要先使用此函式填滿最多 15 個迴圈的緩衝區，如果使用者想要他們的裝置運行更多的群組迴圈次數，使用者在執行完此函式之後，仍可以使用此函式來連續填滿迴圈緩衝區。

◆ 語法：

WORD CPM100_SyncGroupCycUpdata (**BYTE** BoardNo, **BYTE** RecordNum, **BYTE** CycNum, **WORD** *PDOld, **BYTE** *GroupData, **BYTE** BlockMode)

◆ 參數：

BoardNo : [input] CPM100 板卡編號(範圍：0 到 7)。

RecordNum : [input] SYNC 群組的 PDO 數量。每次更新，群組資料的 PDO 數量都是一樣的，此參數範圍小於 10。

CycNum : [input] 此次更新的迴圈次數，此參數範圍小於 15。

*PDOld : [input] 群組資料的 PDO ID 陣列，陣列大小根據“RecordNum”參數。

*GroupData : [input] PDO 群組資料緩衝區。每一個 PDO 物件保留 8 byte 的緩衝區，陣列大小為“CycNum” * “RecordNum” * 8。

BlockMode : [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.83 CPM100_SyncGroupCycSend

◆ 說明：

使用此函式啟動傳送由“CPM100_SyncGroupCycUpdate”更新的群組迴圈緩衝區的 SYNC 和 PDO 群組資料。

◆ 語法：

WORD CPM100_SyncGroupCycSend (**BYTE** BoardNo, **DWORD** TotalCycle, **DWORD** CycInterval, **BYTE** BlockMode)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

TotalCycle： [input] SYNC 群組任務的總發送次數。0: 無限次數。

CycInterval： [input] 迴圈計時器。單位時間為 0.1 毫秒，最小值為 5 (代表 0.5ms)。

BlockMode： [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.84 CPM100_SyncGroupClear

◆ 說明：

使用此函式去清除所有由“CPM100_SyncGroupCycUpdate”更新的群組迴圈緩衝區的 SYNC 和 PDO 群組資料。如果最後 SYNC 群組被某些原因停止了，在開始下一個新的 SYNC 群組資料之前，建議先使用此函式去清除緩衝區裡的群組資料。

◆ 語法：

WORD CPM100_SyncGroupClear (**BYTE** BoardNo, **BYTE** BlockMode)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

BlockMode： [input] 0 代表非阻塞函式且 1 代表阻塞函式。當參數設為 1 時，使用者的應用程序將被延遲直到此函式回傳，如果設為 0，則此函式會直接回傳“CPM_Processing”的訊息。如果使用者輸入相同的“BoardNo”和“Node”的值並再次呼叫此函式，如果程序還未完成則回傳“CPM_Wait”。

4.5.85 CPM100_GetSYNCGroupStopState

◆ 說明：

使用此函式還取得 SYNC 和 PDO 群組資料的停止狀態，此函式可以與使用者自定義 SYNC ISR 功能一起使用(請參閱 4.5.86 CPM100_InstallSYNCGroupISR)。

◆ 語法：

WORD CPM100_GetSYNCGroupStopState (**BYTE** *BoardNo,
BYTE *State)

◆ 參數：

BoardNo：[output] CPM100 板卡編號(範圍：0 到 7)。

*State： [output] 1 表示 SYNC 和 PDO 群組資料傳送完成，2 表示 SYNC 和 PDO 群組資料傳輸失敗而停止。

4.5.86 CPM100_InstallSYNCGroupISR

◆ 說明：

此函式允許使用者使用 CPM100 模組的 SYNC 群組 ISR。當使用者輸入程序到此函式中，如果 SYNC 和 PDO 群組資料被停止或完成時，一個 SYNC 群組停止狀態事件將會觸發 ISR。

◆ 語法：

WORD CPM100_InstallSYNCGroupISR (**BYTE** BoardNo,
void(*SYNCISR)())

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

(*SYNCISR)()： [input] 指標指向“void XXX()”格式的函式。“XXX”代表使用者程序的函式名稱，此程序通常搭配“CPM100_GetSyncGroupStopState”函式使用。

4.5.87 CPM100_RemovesSYNCGroundISR

◆ 說明：

如果使用者不需要 SYNC 群組 ISR 功能，可以呼叫此函式去移除使用者 ISR。

◆ 語法：

WORD CPM100_RemovesSYNCGroundISR (**Byte** BoardNo)

◆ 參數：

BoardNo： [input] CPM100 板卡編號(範圍：0 到 7)。

5. 範例程式

CPM100 產品提供了 10 種不同的範例程式包括演示 NMT 協定、SDO 協定、PDO 協定、NMT Error ISR 等，這些範例程式都支援 VC++、VB.NET 2005 和 C# 2005 開發環境。泓格還提供了 CPMUtility 工具可以輕易的控制和監控掛在 CPM100 模組底下的 CANopen 從站設備。使用者可從下方泓格官方網站下載取得。

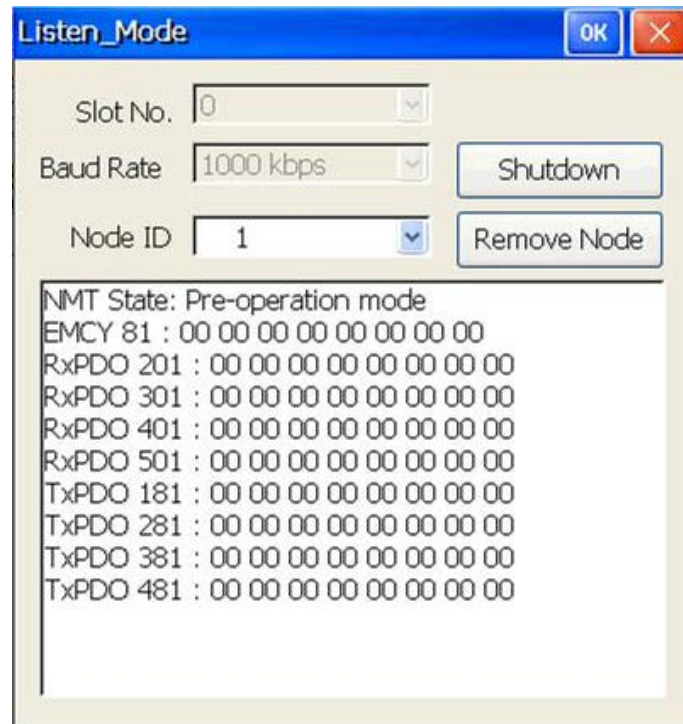
<https://www.icpdas.com/tw/download/index.php?model=PEX-CPM100-D>

5.1 範例程式簡介

這些範例程式是用來展示如何使用 CANopen 的函式庫來進行 CANopen 的通訊協定，提供了 SDO、PDO、NMT 和 SYNC 通訊的應用，每一個範例程式都已使用 CANopen 函式庫裡的一些函式。而 CANopen 函式庫的函式和範例程式的之間的關係將在下面章節描述。

5.1.1 Listen_Mode

使用“listen mode”去初始化 CPM100 模組，然後使用“manual mode”或 EDS 檔案來加入僕端節點，此時，CPM100 模組只會監聽 CANopen 的訊息並不傳輸任何訊息到 CANopen 網路上。在此範例程式中，CPM100 將會監聽 NMT 狀態，包含 4 個 TxPDO 訊息(COB ID 0x180, 0x280, 0x380 和 0x480 + Node ID)、4 個 RxPDO 訊息(COB ID 0x200, 0x300, 0x400 和 0x500 + Node ID)以及 EMCY 訊息。



相關函式列表:

CPM_InitMaster	CPM_Shutdown	CPM_AddNode
CPM_RemoveNode	CPM_EDS_Load	CPM_SetMasterMode
CPM_NMTGetState	CPM_InstallPDO_List	CPM_GetPDOLastData
CPM_GetRxPDOID	CPM_GetTxPDOID	CPM_GetTxPDOID
CPM_SetEMCY_List	CPM_GetEMCYID	CPM_ReadLastEMCY

5.1.2 NMT_Protocol

下面是 NMT 網路控制範例程式，此程式不只展示如何控制特定僕端節點的 NMT 狀態，還可以透過守衛和心跳函式去保護僕端設備。

相關函式列表:

CPM_InitMaster	CPM_Shutdown	CPM_AddNode
CPM_RemoveNode	CPM_NMTChangeState	CPM_NMTGuarding
CPM_NMTHeartbeat	CPM_GetEMCYData	CPM_GetNMTErrors
CPM_InstallNMTErrISR	CPM_RemoveNMTErrISR	CPM_InstallEMCYISR
CPM_RemoveEMCYISR		

5.1.3 PDO_Parameter

預設的 PDO 配置有時候無法滿足使用者需求，使用者需要去改變 PDO 相關的參數如傳輸種類、PDO ID、事件計時器、動態 PDO(dynamic PDO) 等等。下面這個範例程式將演示如何改變 PDO 的參數和顯示最後配置結果。

相關函式列表:

CPM_InitMaster	CPM_Shutdown	CPM_AddNode
CPM_RemoveNode	CPM_InstallPDO	CPM_RemovePDO
CPM_DynamicPDO	CPM_ChangePDOID	CPM_PDOTxType
CPM_PDOWUseEntry	CPM_PDOWEventTimer	CPM_GetTxPDOID
CPM_GetRxPDOID	CPM_GetPDOWMapInfo	

5.1.4 PDO_Protocol

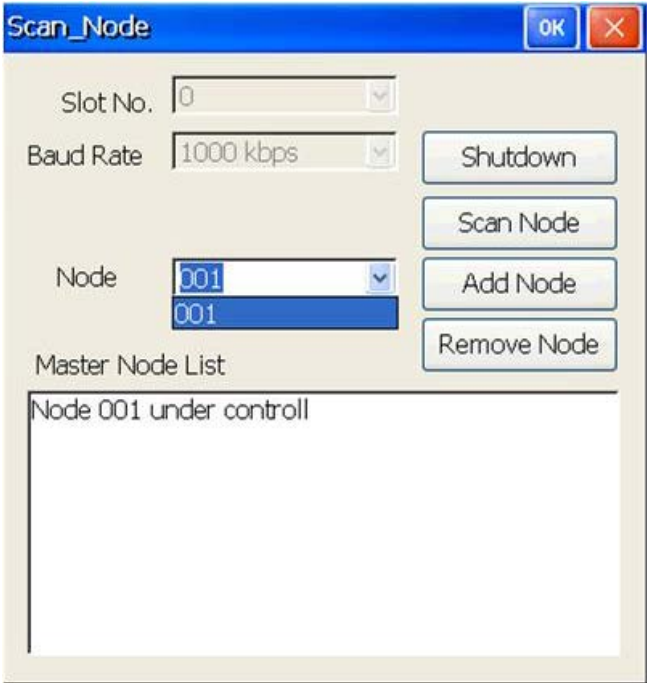
PDO 協定是在 CANopen 網路中去控制特定僕端設備 IO 的主要協定，下面這個範例程式會展示如何透過 PDO 函式去讀寫僕端設備的資料。

相關函式列表:

CPM_InitMaster	CPM_Shutdown	CPM_AddNode
CPM_RemoveNode	CPM_PDOWrite	CPM_PDORemote
CPM_GetPDOLastData		

5.1.5 Scan_Node

以下範例程式將會幫助使用者如果想知道有多少僕端設備存在 CPM100 模組的控制之下。

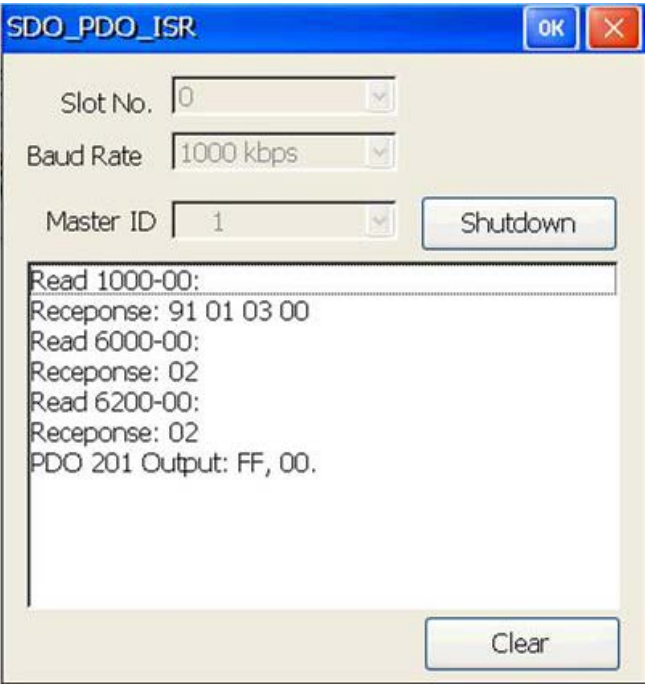


相關函式列表:

CPM_InitMaster	CPM_Shutdown	CPM_AddNode
CPM_RemoveNode	CPM_SetFunctionTimeout	CPM_ScanNode
CPM_GetNodeList		

5.1.6 SDO_PDO_ISR

以下範例程式允許使用者將 CPM100 模組設定成僕端設備，使用者可以使用另一個 CANopen 主端藉由 SDO 設備協定去讀寫 CPM100 模組使用者自定義的物件字典，或者當 CPM100 模組、I-8053W DI 模組和 I-8057W DO 模組都掛載在相同的 MCU 時，藉由 PDO 去取得或設定 DIO 狀態。如果使用者有使用這樣的應用，則此範例程式將會是很好的參考。



相關函式列表:

CPM_InitMaster	CPM_Shutdown	CPM_GetMasterReadSDOEvent
CPM_GetMasterWriteSDOEvent	CPM_GetMasterRemotePDOEvent	CPM_GetMasterRxPDOEvent
CPM_ResponseMasterSDO	CPM_ResponseMasterPDO	CPM_InstallPDO_List
CPM_InstallReadSDOISR	CPM_InstallWriteSDOISR	CPM_InstallRxPDOISR
CPM_InstallRemotePDOISR		

5.1.7 SDO_Read

SDO 協定是一種用來讀寫 CANopen 物件字典的通訊函式，使用者可以透過位址(index 和 sub-index)去讀取任何的物件資料。以下範例程式將展示這個功能。

相關函式列表:

CPM_InitMaster	CPM_Shutdown	CPM_AddNode
CPM_RemoveNode	CPM_SDOReadData	

5.1.8 SDO_Write

SDO 協定是一種用來讀寫 CANopen 物件字典的通訊函式，使用者可以透過位址(index 和 sub-index)去寫入特定物件字典中的物件資料。以下範例程式將展示這個功能。

相關函式列表:

CPM_InitMaster	CPM_Shutdown	CPM_AddNode
CPM_RemoveNode	CPM_SDOWriteData	

5.1.9 SYNC_Protocol

SYNC 協定是 PDO 通訊的同步函式且一直使用於 PDO 通訊的傳輸種類。

下面範例程式將幫助使用者知道如何使用相關的 SYNC 函式。

相關函式列表:

CPM_InitMaster	CPM_Shutdown	CPM_AddNode
CPM_RemoveNode	CPM_ChangeSYNCID	CPM_GetSYNCID
CPM_SendSYNCMsg	CPM_GetCyclicSYNCInfo	

5.1.10 PDO_MultiData

使用者有時候為了增加效能想要同時輪詢數個 PDO 物件，但是藉由傳送遠端 PDO 去一一輪詢每一個 PDO 物件資料是比較慢的。所以使用者可以對這些 PDO 物件設定時間計時器或是遠端列表，當物件資料被 CPM100 模組輪詢的時候或者被僕端設備想要自動回覆時，就可以使用“CPM100_GetMultiPDOData”函式去同時獲取在緩衝區的 POD 物件資料。

相關函式列表:

CPM_InitMaster	CPM_Shutdown	CPM_AddNode
CPM_RemoveNode	CPM_GetTxPDOID	CPM_SetPDORemotePolling
CPM_PDOWEventTimer	CPM_GetMultiPDOData	

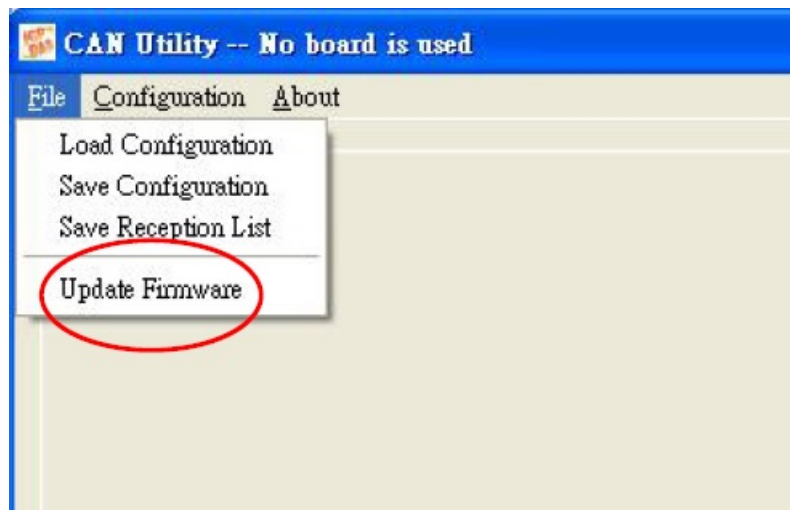
6. 韌體更新

CPM100 模組中的韌體可以透過 ICP DAS 的 CANUtility 工具來進行更新。
可由下面泓格官方網站下載取得。

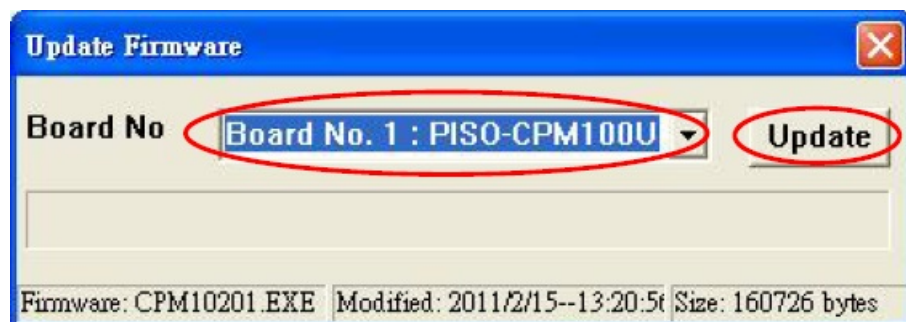
<https://www.icpdas.com/tw/download/show.php?num=3862&model=PISO-CAN200U-D>

下方將描述使用 ICP DAS 韌體更新工具更新 CPM100 韌體的程序。

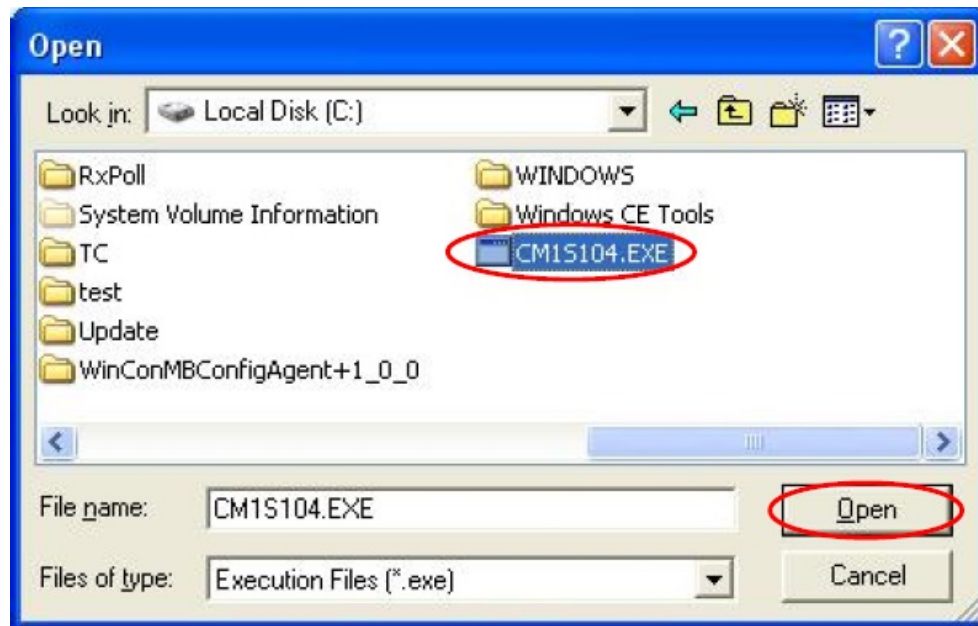
步驟 1：執行 CANUtility 韌體更新工具，在工具列裡點選“File→Update Firmware”如下圖所示。



步驟 2：務必選取相對應的模組型號，然後點選“Update”。



步驟 3：選取更新的韌體檔案（ CPM100 韌體檔案格式為“.exe”）。



步驟 4：當完成更新程序，會出現“Download OK”的視窗，如下圖所示，
點擊“OK”即完成。

