

PISO-CM200U 系列 使用手冊

版本 1.0.0, 2016 年 10 月



產品技術服務與使用資訊

PISO-CM200U-D / PISO-CM200U-T

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。

技術服務

如有任何問題，請與本公司客服聯絡，我們將盡速為您服務。

Email 信箱：service@icpdas.com

目錄

1.	簡介.....	5
1.1.	規格.....	6
1.2.	特性.....	8
1.3.	機構.....	9
1.4.	概觀.....	11
1.4.1.	版卡布置.....	11
1.4.2.	接腳定義.....	12
1.4.3.	開關設定.....	13
1.4.4.	終端電阻設定.....	14
1.4.5.	LED指示燈及操作模式.....	15
2.	入門.....	16
2.1.	驅動程式安裝.....	16
2.2.	安裝硬體裝置.....	19
2.3.	CAN總線接線.....	20
3.	Windows應用程式介面功能參考.....	21
3.1.	系統資訊應用程式介面.....	21
3.1.1.	CM200_GetDllVersion.....	22
3.1.2.	CM200_GetBoardInf.....	23
3.1.3.	CM200_TotalBoard.....	25
3.1.4.	CM200_TotalCM200Board.....	26
3.1.5.	CM200_GetCM200BoardSwitchNo.....	27
3.1.6.	CM200_GetCardPortNum.....	28
3.1.7.	CM200_ActiveBoard.....	29
3.1.8.	CM200_CloseBoard.....	30
3.1.9.	CM200_BoardIsActive.....	31
3.1.10.	CM200_CheckMCUMode.....	32

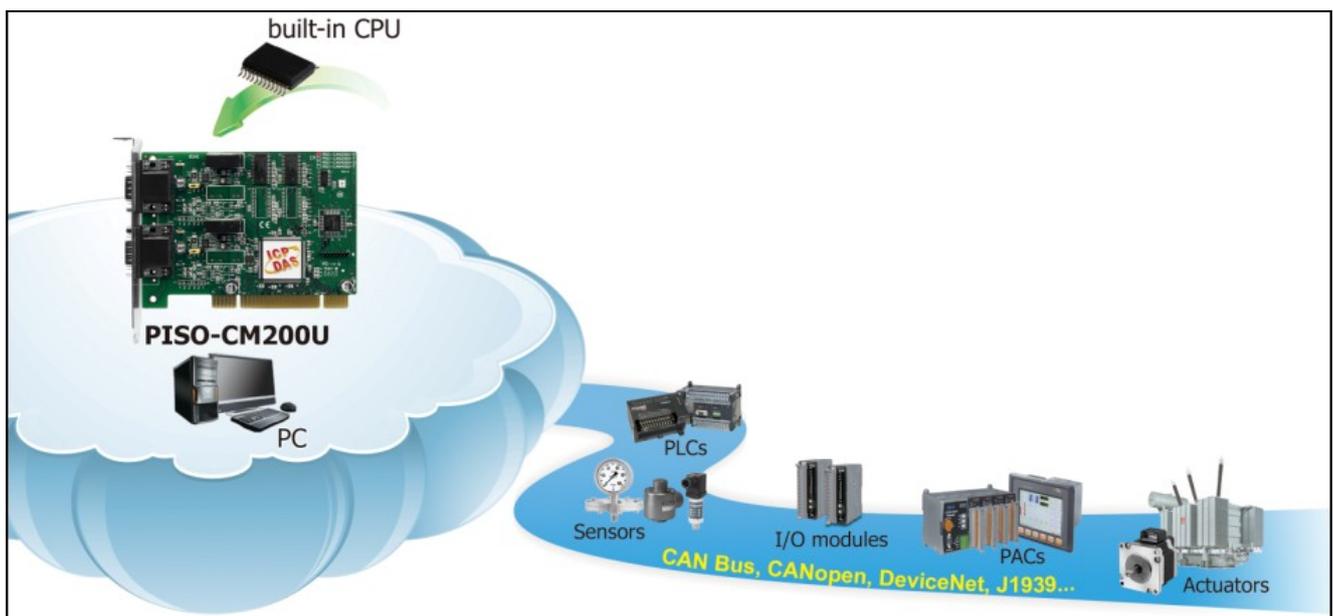
3.1.11.	CM200_HardwareReset.....	33
3.1.12.	CM200_AdjustDateTime	34
3.2.	CAN總線應用程式介面	35
3.2.1.	CM200_CANReset	36
3.2.2.	CM200_CANGetStatus	37
3.2.3.	CM200_EnableCAN.....	39
3.2.4.	CM200_DisableCAN	40
3.2.5.	CM200_SetCANConfig.....	41
3.2.6.	CM200_GetCANConfig	44
3.2.7.	CM200_AddCyclicTxMsg	46
3.2.8.	CM200_DeleteCyclicTxMsg	49
3.2.9.	CM200_EnableCyclicTxMsg.....	50
3.2.10.	CM200_DisableCyclicTxMsg	51
3.2.11.	CM200_CheckCyclicTxRestMsg	52
3.2.12.	CM200_IsTxTimeout	53
3.2.13.	CM200_CANInit	54
3.2.14.	CM200_RxMsgCount.....	55
3.2.15.	CM200_ReceiveCANMsg	56
3.2.16.	CM200_SendCANMsg.....	58
3.2.17.	CM200_ClearSoftBuffer	60
3.2.18.	CM200_ClearTxSoftBuffer	61
3.2.19.	CM200_ClearRxSoftBuffer	62
3.2.20.	CM200_ClearBufferStatus	63
3.3.	回傳碼定義.....	64
4.	韌體更新.....	66
5.	附錄.....	69
5.1.	防電磁干擾磁環	69

1. 簡介

CAN (控制區域總線) 是一種串行通信協議，並有效地支持具有非常高的安全級別的分佈式實時控制應用。它是一種特別適合於“智能”設備系統或子系統中傳感器和致動器間的網絡。且在 CAN 網絡上，有別於傳統站與站間的資料傳遞方式，高優先權的訊息會優先被傳送。

PISO-CM200U 為具有 2 個 CAN 通道的 CAN 板卡，提供非常強大和經濟的解決方案，可以覆蓋範圍廣的 CAN 應用。其具有 32 位元微控制器，可以實現許多其他功能，如訊息過濾、預處理，以及 CAN 報文存儲 (時間戳記)，以及 CAN 信息的實時傳輸。此外，PISO-CM200U 還採用了 Bosch C_CAN 控制器和 NXP TJA1042 收發器，提供總線仲裁和錯誤檢測與自動校正和重新傳輸功能。藉由此微控制器，即使在具有高的總線負載的系統上，CAN 控制器不會丟失數據。此外，搭配集成智能功能，PISO-CM200U 能夠預處理 CAN 數據流，因而可減輕個人電腦的負擔，在個人電腦上的應用程序的實時性要求將會被大大降低。另外，它可以安裝在 32 位 PCI 總線或通用 PCI 總線插槽並支持真正的“即插即用”技術。

此外，用戶可以通過使用 PISO-CM200U 函式庫開發自己的 CAN 總線應用程序。當 PISO-CM200U 是啟用時，用戶的應用程序和 CAN 總線韌體之間的數據交換是通過 PISO-CM200U 的存儲器映射方法進行。



1.1. 規格

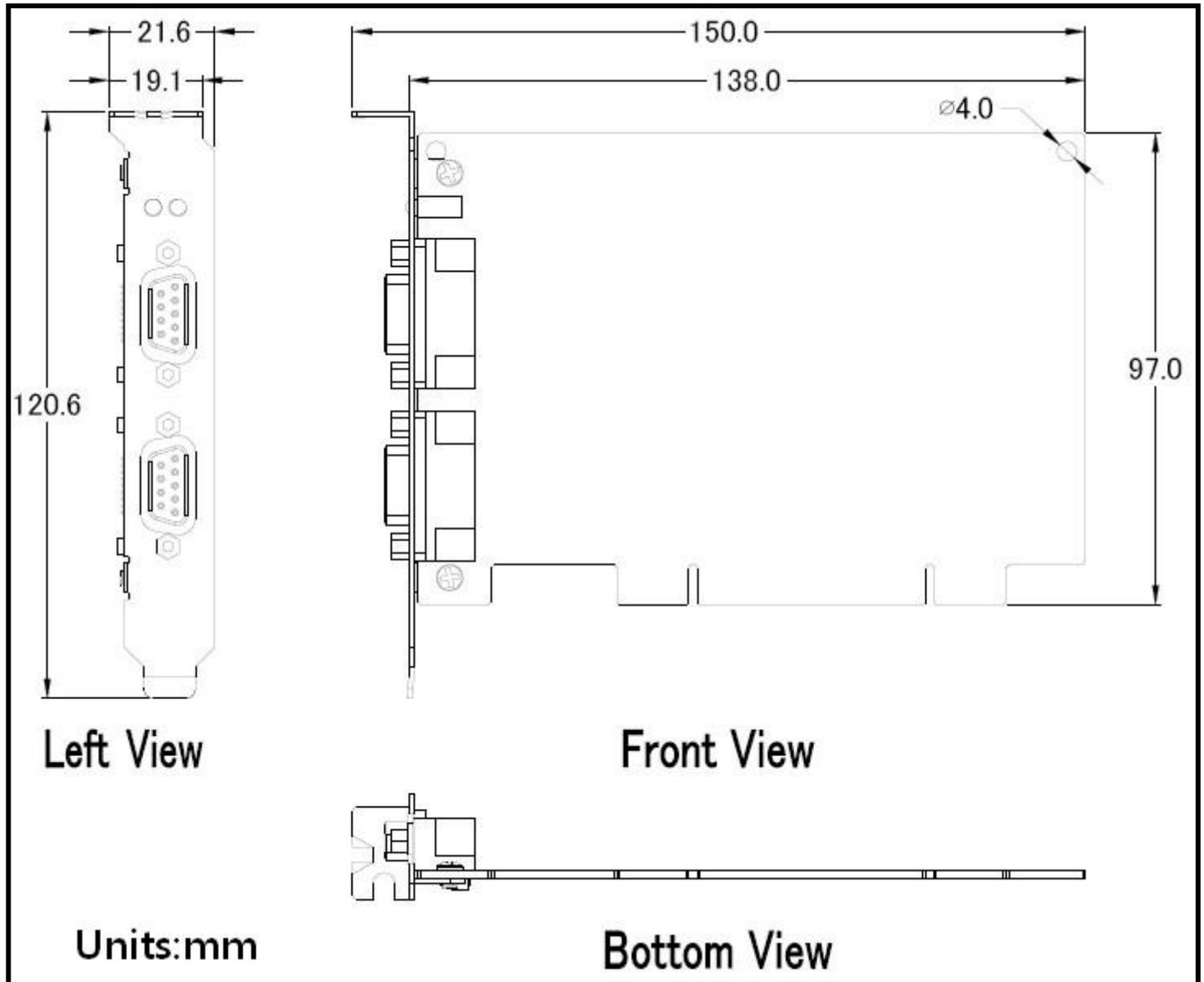
模組名稱	PISO-CM200U-D	PISO-CM200U-T
硬體		
微控制器	32-bit 微控制器	
記憶體	16 KB	
實時時鐘	支援	
匯流排介面		
類型	Universal PCI 介面, 支援 5 V 訊號, 33 MHz, 32 位元, 隨插即用	
板卡編號	指撥開關設定	
CAN 總線介面		
控制器	Bosch C_CAN	
收發器	NXP 82C250	
通道數	2	
接頭	9 針公座 D-Sub 端子	5 針螺絲端子
通訊速率(bps)	10 k, 20 k, 50 k, 125 k, 250 k, 500 k, 800 k, 1 M (允許使用者自定義鮑率)	
隔離	3000 V DC-DC 隔離, 依 UL1577 規範 3000 Vrms 持續一分鐘 (光耦合)	
終端電阻	跳線設定 120 Ω 終端電阻	
指示燈		
圓形 LED	Rx/Tx LED, ERR LED	
電源		
功耗	800 mA @ 5 V	
軟體		
驅動程式	Windows XP/7/8.1/10 (32 位元/64 位元)作業系統	
函式庫/範例	C#.Net, VB.Net, VC++.Net	
機構		
尺寸 (長 x 寬 x 深)	150 mm x 121 mm x 22 mm	
環境		
操作溫度	0 ~ 60 °C	
儲存溫度	-20 ~ 70 °C	

1.2. 特性

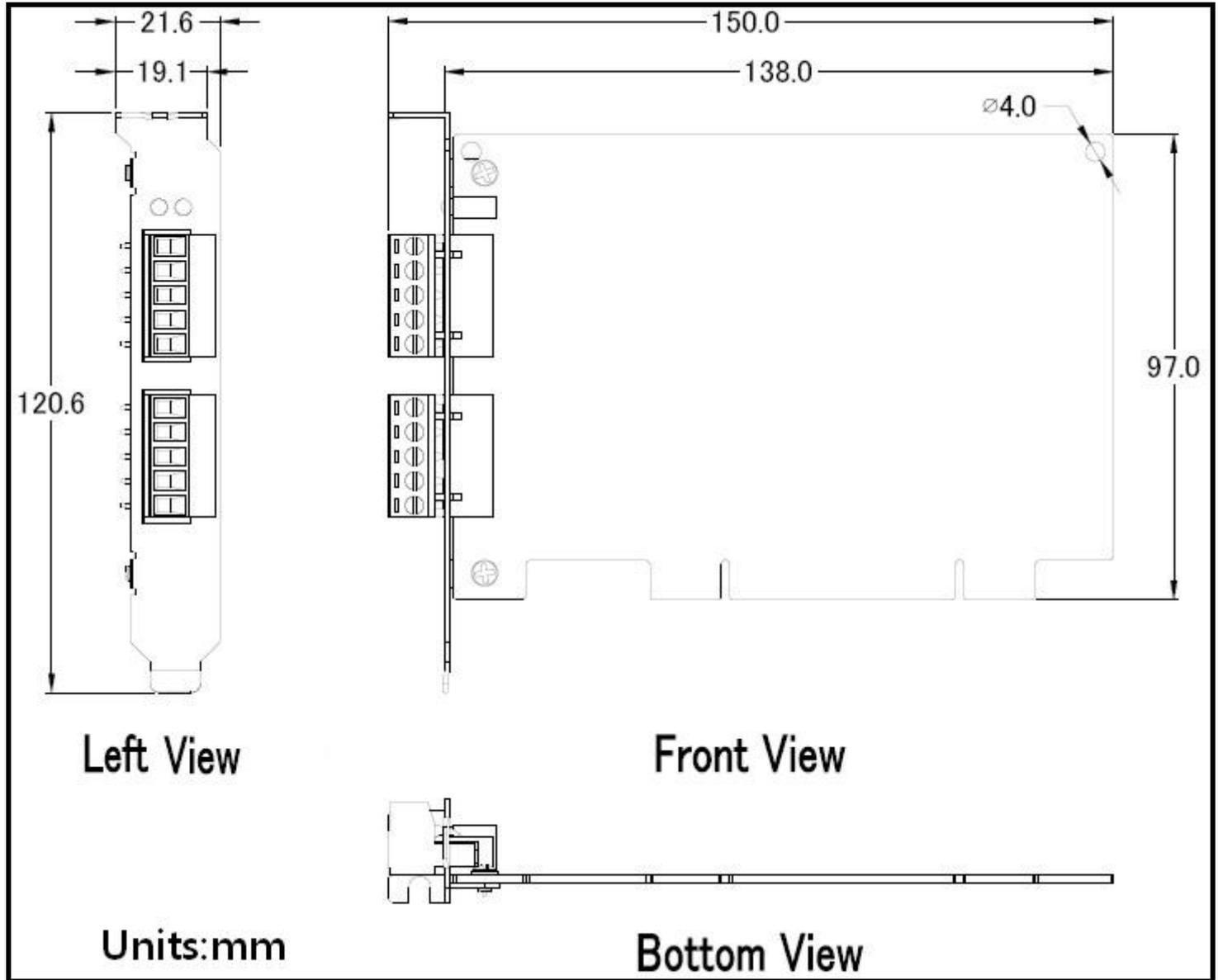
- 32 位元微處理器
- NXP 82C250 CAN 收發器
- Bosch C_CAN CAN 控制器
- 支援 CAN 2.0A 及 2.0B 規範
- CAN 訊息支援時間戳記 (精度: ± 1 毫秒)
- 透過指撥開關調整版卡編號
- 雙埠隨機存取記憶體通訊架構
- 支援實時時鐘
- 驅動程式支援 Windows XP/7/8.1/10 作業系統
- 提供 C#.Net, VB.Net and VC++.Net 範例及函式庫
- 支援 256 筆 CAN 總線傳送/接收緩衝
- 支援 5 筆循環傳送 CAN 總線訊息
- 循環傳送 CAN 總線訊息精度: ± 1 毫秒
- 易於更新韌體

1.3. 機構

PISO-CM200U-D



PISO-CM200U-T

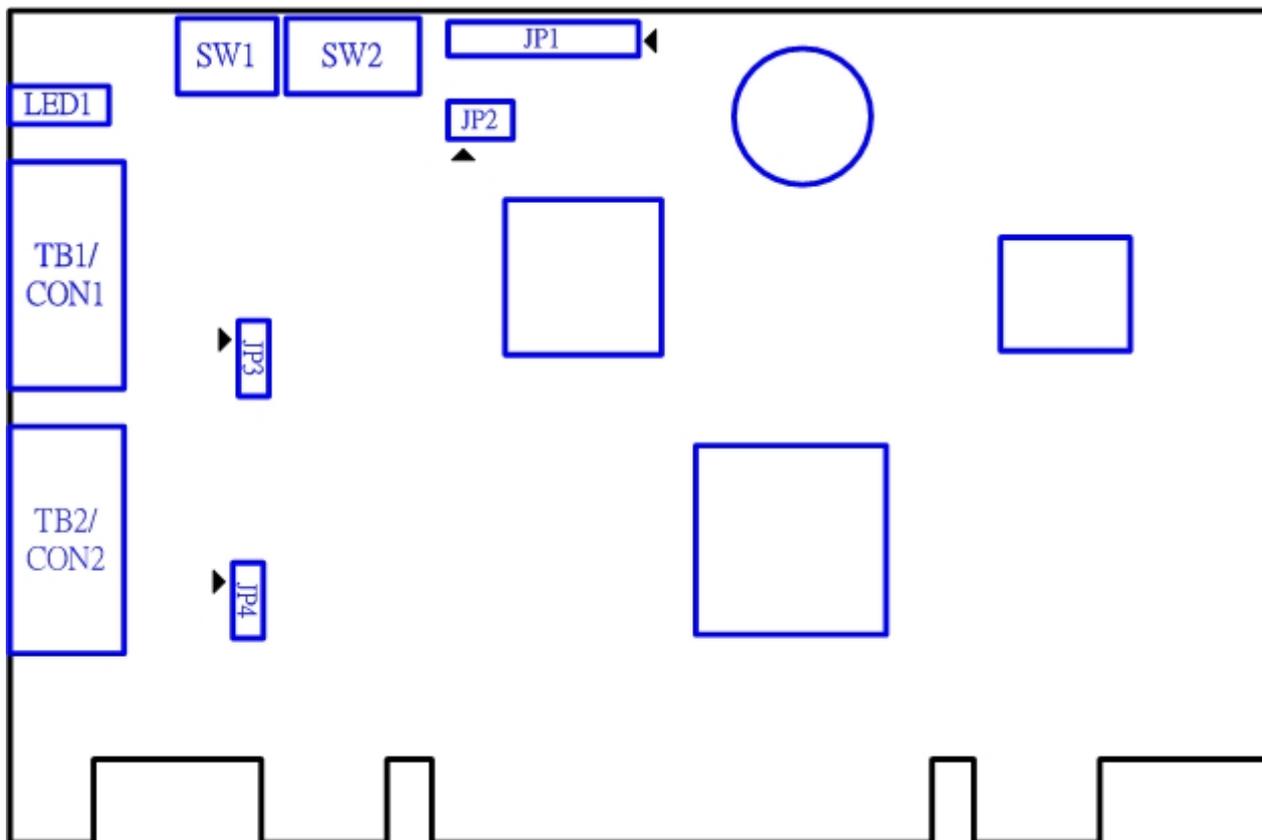


1.4. 概觀

本節介紹 PISO-CM200U 系列的硬體設置。這些信息包括板卡布置，引腳分配，跳線和開關選擇和 LED 指示燈。

1.4.1. 版卡布置

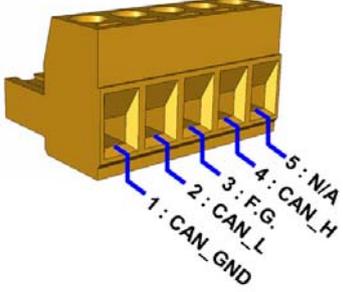
下圖就是說明 PISO-CM200U 系列功能的板卡布置。



1.4.2. 接腳定義

PISO-CM200U 系列的 5 針螺絲端子及 9 針公座 D-Sub 端子接頭詳列於下表。

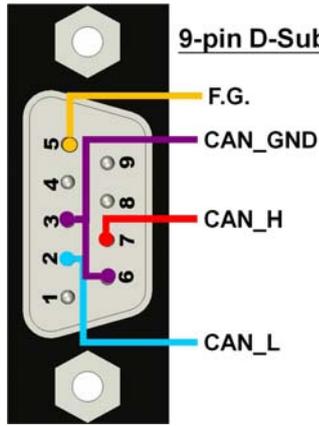
5 針螺絲端子接腳定義		
接腳	名稱	描述
1	CAN_GND	CAN 埠 CAN_Gnd 訊號
2	CAN_L	CAN 埠 CAN_Low 訊號
3	F.G.	機架接地
4	CAN_H	CAN 埠 CAN_High 訊號
5	N/A	沒有使用



5-pin screw terminal block

5: N/A
4: CAN_H
3: F.G.
2: CAN_L
1: CAN_GND

9 針公座 D-Sub 接腳定義		
接腳	名稱	描述
1	N/A	沒有使用
2	CAN_L	CAN 埠 CAN_Low 訊號
3	CAN_GND	CAN 埠 CAN_Gnd 訊號
4	N/A	沒有使用
5	F.G.	機架接地
6	CAN_GND	CAN 埠 CAN_Gnd 訊號
7	CAN_H	CAN 埠 CAN_High 訊號
8	N/A	沒有使用
9	N/A	沒有使用



9-pin D-Sub male connector

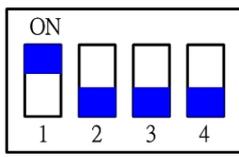
F.G.
CAN_GND
CAN_H
CAN_L

電子電路設備容易受到不同程度的靜電放電 (ESD, Electro-Static Discharge) 現象所干擾，在大陸性氣候區會變得更糟。F.G.腳位提供了一個進行防靜電的接地路徑。因此，正確地連接 F.G.可以增強的 ESD 保護能力，並提高了模組的可靠性。

Power F.G.的接線並不是必要的，根據實際應用中，用戶可以修改佈線的配置。

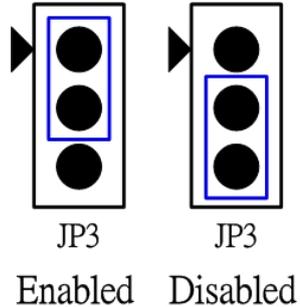
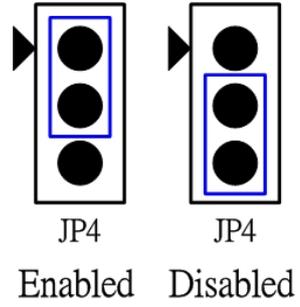
1.4.3. 開關設定

下表定義了 PISO-CM200U 系列的開關及指撥開關的設定。使用者可以參考下表來設定 PISO-CM200U 系列的硬體

開關	描述	狀態
SW1	此開關是用來讓版卡韌體進入 Boot Loader 模式。當韌體需要做更新時，使用者可以長按此開關 3 秒來重置韌體。當韌體重置後，就會進入到 Boot Loader 模式，此時就可以透過更新工具做韌體更新。	 SW1
SW2	此指撥開關是用來設定版卡的編號。此 4 個指撥開關會對應至版卡由 0 至 15 的編號。數字 1 為最低有效位，數字 4 為最高有效位。當一台 PC 上插多張版卡時，每塊板卡的編號設定需不相同。	<p>DIP Switch</p>  <p>Thus configuration indicates that the board number is set to 1.</p>

1.4.4. 終端電阻設定

下表定義了 PISO-CM200U 系列 CAN 埠的終端電阻設定。使用者可依下表來調整是否要啟用終端電阻。

跳線	描述	狀態
JP3	啟用/關閉 CAN 埠 1 的終端電阻設定。	 JP3 Enabled JP3 Disabled
JP4	啟用/關閉 CAN 埠 2 的終端電阻設定。	 JP4 Enabled JP4 Disabled

1.4.5. LED 指示燈及操作模式

當PISO-CM200U系列的韌體功能處於不同模式時，LED指示燈的顯示狀態將會不同。下面列出PISO-CM200U系列的各種模式及LED指示燈的顯示狀態。

1. Boot Loader 模式

當使用韌體更新工具時，版卡會被切換至“Boot Loader”模式。在此模式下，PISO-CM200U系列的紅、綠LED指示燈會交替每秒閃爍一次。使用者可以使用韌體更新工具來更新版卡韌體。

2. Firmware Running 模式

當版卡正常啟動時，會進入“Firmware Running”模式，在此模式下，當CAN總線無訊息在收送時，PISO-CM200U系列的紅、綠LED指示燈將會恆滅，當有再傳送/接收CAN訊息時，綠色指示燈將會閃爍。當CAN總線有出現錯誤時，紅色指示燈將會恆亮；此時使用者可以使用“CM200_CANGetStatus”函式來確認CAN總線出現何種異常。

3. Firmware Reset 模式

當使用者長按“SW1”開關超過3秒時，版卡會進入“Firmware Reset”模式。在此模式下，模組韌體將會重置，此時紅、綠LED指示燈將會恆滅約1秒鐘的時間，之後版卡將會進入“Boot Loader”模式。當模組韌體有異常不受控制時，使用者可透過“SW1”開關來重置/更新韌體。

2. 入門

本章說明PISO-CM200U系列版卡的驅動程式安裝及CAN總線的接線。

2.1. 驅動程式安裝

PISO-CM200U 系列版卡的驅動程式可以使用在 Windows 的作業系統上。使用者可以依作業系統類型來選擇相對應之驅動程式。在產品配件所附之 Fieldbus 光碟片中可以找到相對應之驅動程式。

CD:\can\pci\piso-cm200u\driver\windows.

使用者也可以由產品網頁上自行下載最新的驅動程式。產品網頁如下:

ftp://ftp.icpdas.com/pub/cd/fieldbus_cd/can/pci/piso-cm200u/driver/windows

或者

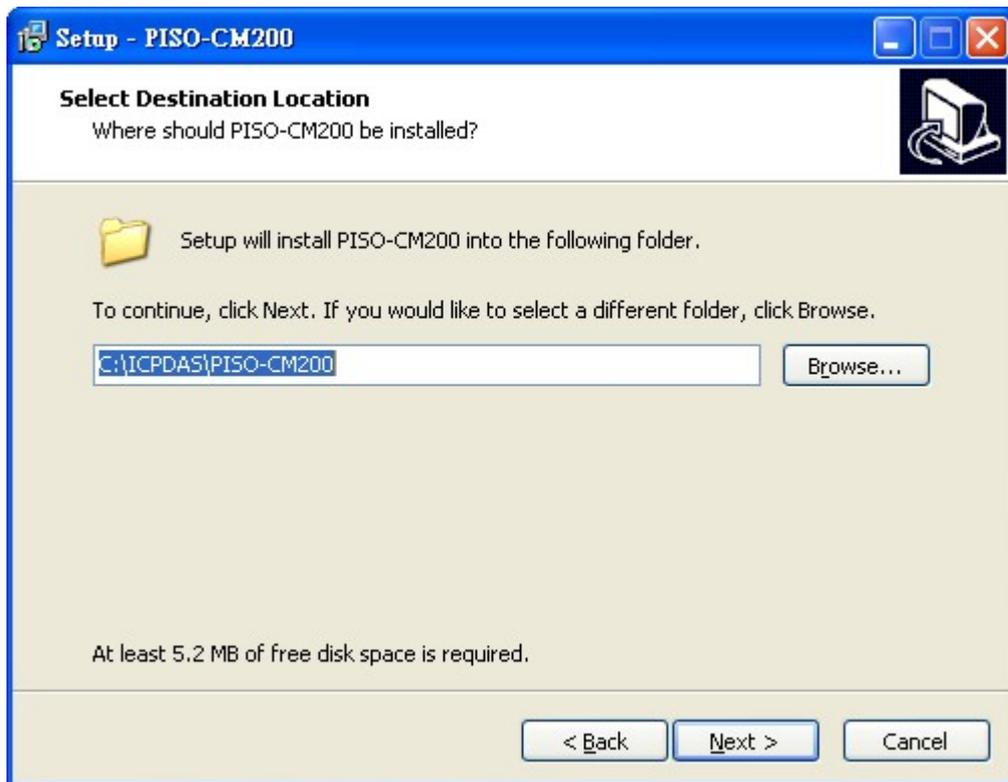
http://ftp.icpdas.com/pub/cd/fieldbus_cd/can/pci/piso-cm200u/driver/windows

下面描述為使用 Windows XP 作業系統為範例來安裝驅動程式軟體。如果使用者使用其他作業系統，安裝步驟會與 Windows XP 相類似。

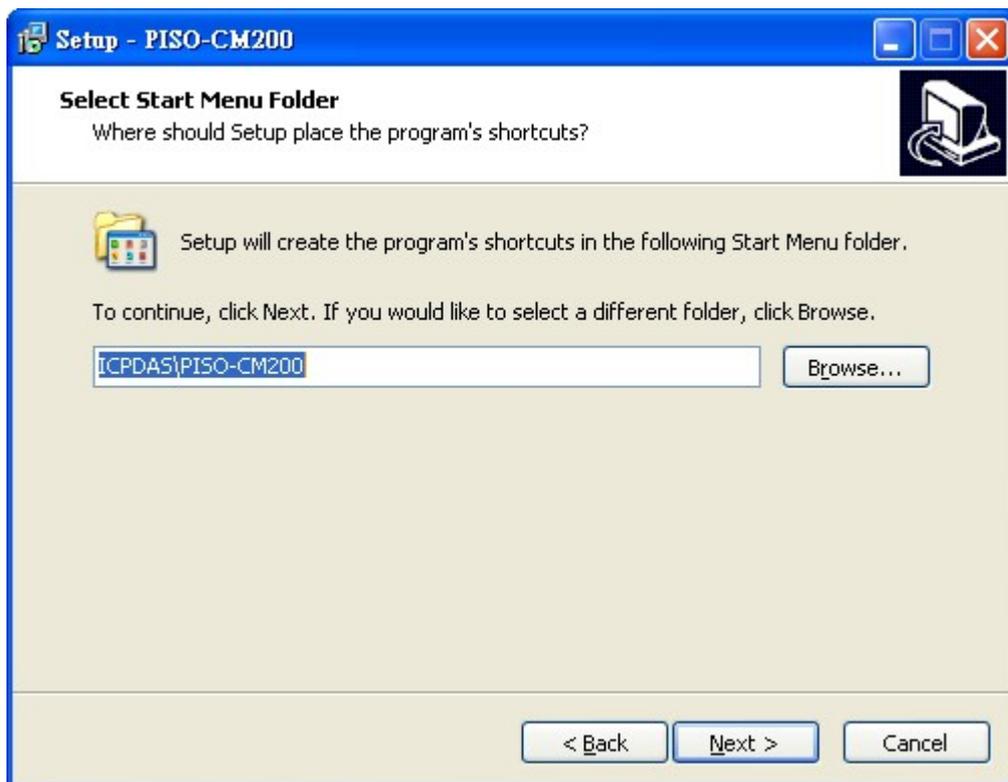
步驟 1: 執行驅動程式安裝軟體(piso-cm200_setup.exe)。使用者可以由光碟片或產品網頁上取得版卡之驅動程式。



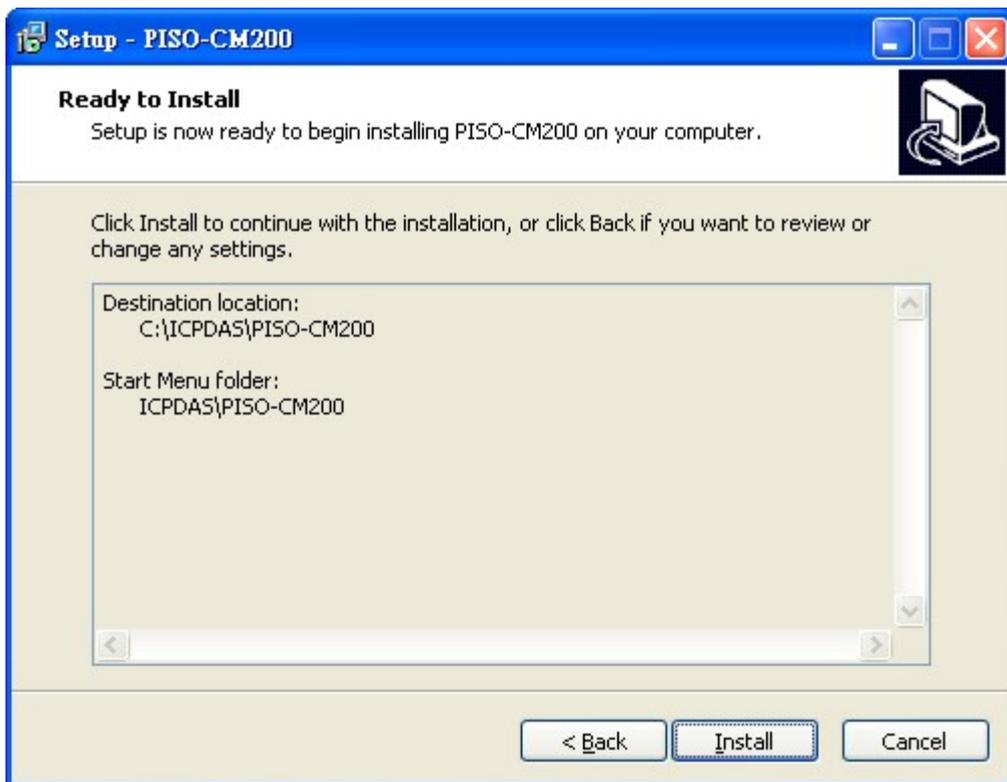
步驟 2: 選擇檔案安裝路徑。



步驟 3: 選擇程式集安裝路徑。



步驟 4: 準備安裝，請按下”Install”按鈕，開始安裝檔案。



步驟 5: 安裝完成，請重新啟動作業系統。



2.2. 安裝硬體裝置

步驟 1: 關掉電腦電源。

步驟 2: 打開電腦機殼。

步驟 3: 選擇未使用的 PCI 插槽，並移除 保護裝置。

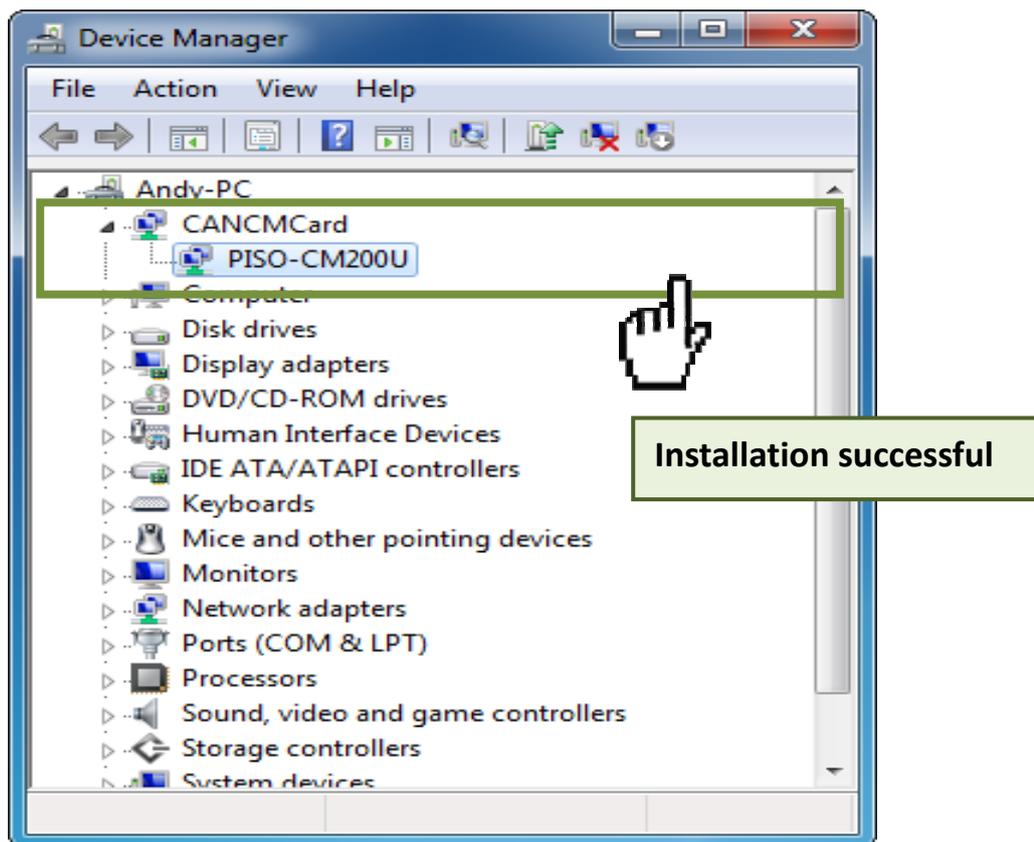
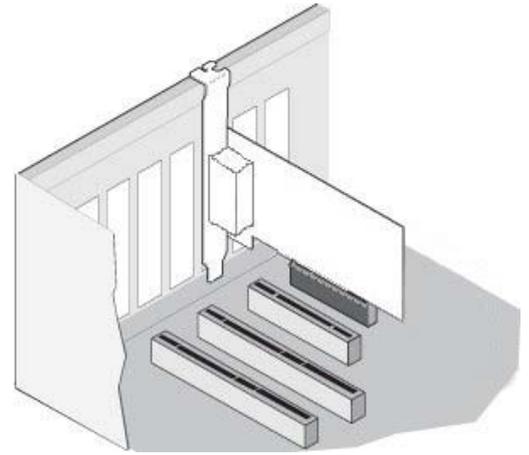
步驟 4: 小心插入 PISO-CM200U 卡至 PCI 插槽，並以螺絲固定住。

步驟 5: 裝回電腦機殼。

步驟 6: 啟動電腦電源。

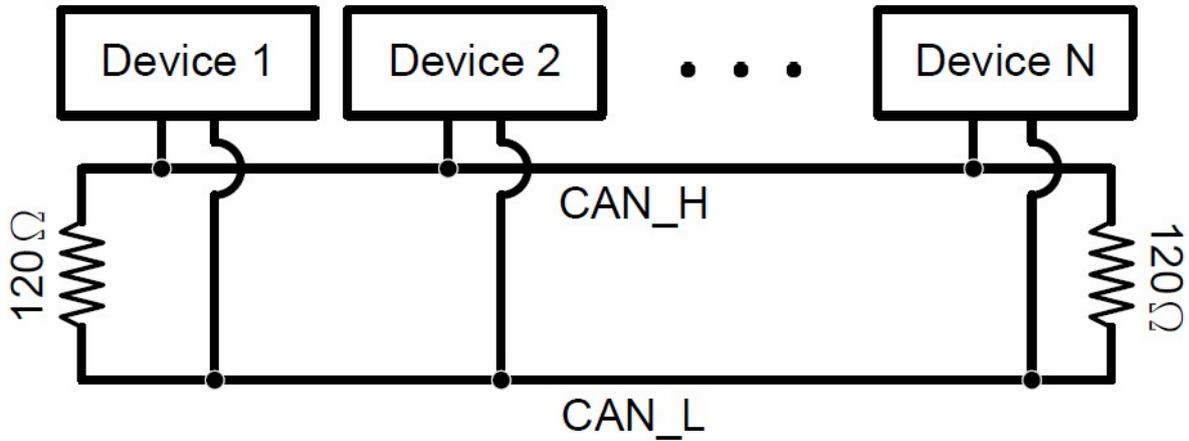
步驟 7: 進入 Windows 後，依照提示訊息完成即插即用驅動安裝。

步驟 8: 開啟“裝置管理員”來確認 PISO-CM200U 卡是否正確安裝。當 PISO-CM200U 卡被正確安裝，將如下圖所示。



2.3. CAN 總線接線

為了盡量減少 CAN 總線上的反射效果，CAN 總線具有在兩端使用兩個終端電阻將總線終止。根據 ISO11898-2 規範，每一個終端電阻為 120Ω (或 $108\Omega \sim 132\Omega$ 之間)。總線拓撲結構和這些終端電阻的位置，如下圖所示。



此外，為了最小化長距離上的電壓下降，終端電阻應大於在 ISO11898-2 中定義的值更高。下表可以作為一個很好的參考。

導線長度 (m)	導線參數		終端電阻 (Ω)
	阻值(m Ω /m)	導線截面(mm ²)	
0~40	70	0.25(23AWG)~ 0.34mm ² (22AWG)	124 (0.1%)
40~300	<60	0.34(22AWG)~ 0.6mm ² (20AWG)	127 (0.1%)
300~600	<40	0.5~0.6mm ² (20AWG)	150~300
600~1K	<20	0.75~0.8mm ² (18AWG)	150~300

3. Windows 應用程式介面功能參考

本章說明 PISO-CM200U 韌體所支援之系統資訊應用程式介面。內容包含系統資訊應用程式介面，CAN 總線應用程式介面及功能函式回傳碼。相關內容可以有效的幫助使用者開發其應用程式。

3.1. 系統資訊應用程式介面

PISO-CM200U 系列所支援之系統資訊應用程式介面條列於下表。每個應用程式介面的內容條列於下列小節。

- 系統資訊相關聯之應用程式介面總表

函式庫功能	功能描述
CM200_GetDllVersion	讀取功能函式庫的版本
CM200_GetBoardInf	讀取 PCI 版卡裝置識別碼資訊
CM200_TotalBoard	讀取系統所安裝之 PISO-CM200U 系列版卡總數量
CM200_TotalCM200Board	讀取系統所安裝之 PISO-CM200U 版卡總數量
CM200_GetCM200BoardSwitchNo	讀取 PISO-CM200U 版卡的指撥開關設定值
CM200_GetCardPortNum	讀取所選擇之板卡 CAN 埠數量
CM200_ActiveBoard	啟用所選擇之 PISO-CM200U 系列版卡
CM200_CloseBoard	關閉所選擇之 PISO-CM200U 系列版卡
CM200_BoardIsActive	檢查所選擇之 PISO-CM200U 系列版卡是否有啟用
CM200_CheckMCUMode	檢查所選擇之 PISO-CM200U 系列版卡之韌體運作模式
CM200_HardwareReset	重置所選擇之 PISO-CM200U 系列版卡
CM200_AdjustDateTime	設定所選擇之 PISO-CM200U 系列版卡的實時時鐘日期

3.1.1. CM200_GetDllVersion

取得 PISO-CM200U 系列版卡之函式庫(CM200.dll)版本。

語法

```
C#  
-----  
UInt16 CM200_GetDllVersion();
```

參數

此函式無設定參數。

回傳值

回傳函式庫(CM200.dll)之版本。例如：當回傳值為"100" (16 進制)時，表示函式庫的版本為 v1.00。

3.1.2. CM200_GetBoardInf

取得 PISO-CM200U PCI 版卡的裝置識別碼資訊，如：廠商識別碼、設備識別碼、子供應商識別碼、子設備識別碼及插斷請求碼等資訊。

語法

```
C#
-----
Int16 CM200_GetBoardInf(
    Byte BoardNo,
    out UInt32 dwVID,
    out UInt32 dwDID,
    out UInt32 dwSVID,
    out UInt32 dwSDID,
    out UInt32 dwSAuxID,
    out UInt32 dwIrqNo
);
```

參數

BoardNo

[in] 指定的板卡編號。

dwVID

[out] 要獲取之 vendor ID 的變數位址。

dwDID

[out] 要獲取之 device ID 的變數位址。

dwSVID

[out] 要獲取之 sub-vendor ID 的變數位址。

dwSDID

[out] 要獲取之 sub-device ID 的變數位址。

dwSAuxID

[out] 要獲取之 sub-auxiliary ID 的變數位址。

dwlrqNo

[out] 要獲取之插斷請求碼資訊的變數位址。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第 3.3 章節的回傳值定義。

3.1.3. CM200_TotalBoard

取得個人電腦系統上 PISO-CM200U 系列版卡的總數量。

語法

C#

```
Int16 CM200_TotalBoard (  
    out Byte BoardNo  
);
```

參數

BoardNo

[out] 要獲取之個人電腦系統上 PISO-CM200U 系列版卡的總數量的變數位址。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第 3.3 章節的回傳值定義。

3.1.4. CM200_TotalCM200Board

取得個人電腦系統上 PISO-CM200U 版卡的總數量。

語法

```
C#  
-----  
Int16 CM200_TotalCM200Board (  
    out Byte BoardNo  
);
```

參數

BoardNo

[out] 要獲取之個人電腦系統上 PISO-CM200U 版卡的總數量的變數位址。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第 3.3 章節的回傳值定義。

3.1.5. CM200_GetCM200BoardSwitchNo

取得所指定之 PISO-CM200U 系列版卡上的指撥開關數值。

語法

```
C#
-----
Int16 CM200_GetCM200BoardSwitchNo(
    Byte BoardNo,
    out Byte BoardSwitchNo
);
```

參數

BoardNo

[in] 指定的板卡編號。

BoardSwitchNo

[out] 要獲取之 PISO-CM200U 系列版卡上的指撥開關數值的變數位址。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第 3.3 章節的回傳值定義。

3.1.6. CM200_GetCardPortNum

取得所指定之PISO-CM200U系列版卡上的CAN埠數量。

語法

C#

```
Int16 CM200_GetCardPortNum(  
    Byte BoardNo,  
    out Byte PortNum  
);
```

參數

BoardNo

[in] 指定的板卡編號。

PortNum

[out] 要獲取之 PISO-CM200U 系列版卡上的 CAN 埠數量的變數位址。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第 3.3 章節的回傳值定義。

3.1.7. CM200_ActiveBoard

啟用指定的 PISO-CM200U 系列版卡。當使用其他應用程式介面函式前，此函式必須被呼叫過一次。

語法

```
C#
-----
Int16 CM200_ActiveBoard(
    Byte BoardNo
);
```

參數

BoardNo

[in] 指定的板卡編號。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第 3.3 章節的回傳值定義。

3.1.8. CM200_CloseBoard

關閉指定的 PISO-CM200U 系列版卡。當使用者結束其應用程式前，此函式必須被呼叫過一次。

語法

C#

```
-----  
Int16 CM200_CloseBoard(  
    Byte BoardNo  
);
```

參數

BoardNo

[in] 指定的板卡編號。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第 3.3 章節的回傳值定義。

3.1.9. CM200_BoardIsActive

取得所指定之 PISO-CM200U 系列板卡是否被啟用之狀態。

語法

```
C#  
-----  
Int16 CM200_BoardIsActive(  
    Byte BoardNo,  
    out Byte IsActive  
);
```

參數

BoardNo

[in] 指定的板卡編號。

IsActive

[out] 要獲取之 PISO-CM200U 系列板卡是否被啟用之狀態的變數位址。

數值 0: 所指定之板卡尚未被啟用。

數值 1: 所指定之板卡已經被啟用。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第 3.3 章節的回傳值定義。

3.1.10. CM200_CheckMCUMode

取得所指定之PISO-CM200U系列板卡微控制器中的韌體的運行模式。

語法

```
C#
-----
Int16 CM200_CheckMCUMode (
    Byte BoardNo,
    out Byte Mode
);
```

參數

BoardNo

[in] 指定的板卡編號。

Mode

[out] 要獲取之 PISO-CM200U 系列版卡微控制器中，韌體運行模式的變數位址。
數值 0: 表示韌體目前處於“Boot Loader”模式，在此模式下 CAN 總線的通訊功能將會被關閉，使用者僅可以更新微控制器的韌體。

數值 1: 表示韌體目前處於“Firmware Running”模式，在此模式下，使用者可以透過 CAN 總線相關連之應用程式介面函式來收送 CAN 總線訊息。

使用者可以透過 CM200_HardwareReset()來將韌體由“Boot Loader”模式切換至“Firmware Running”模式。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第 3.3 章節的回傳值定義。

3.1.11. CM200_HardwareReset

重置 PISO-CM200U 系列的板卡之所有硬體功能包含 CAN 總線通訊、DRPAM 功能。重置成功後，微控制器中的韌體將會進入於“Firmware Running”模式。

語法

```
C#
-----
Int16 CM200_HardwareReset (
    Byte BoardNo
);
```

參數

BoardNo

[in] 指定的板卡編號。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第 3.3 章節的回傳值定義。

3.1.12. CM200_AdjustDateTime

使用當下之個人電腦上的系統時間來設定 PISO-CM200U 系列的板卡中的實時時鐘時間。

語法

```
C#  
-----  
Int16 CM200_AdjustDateTime (  
    Byte BoardNo  
);
```

參數

BoardNo

[in] 指定的板卡編號。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第 3.3 章節的回傳值定義。

3.2. CAN 總線應用程式介面

PISO-CM200U 系列所支援之 CAN 總線應用程式介面條列於下表。每個應用程式介面的內容條列於下列小節。

- CAN 總線應用程式介面總表

函式庫功能	功能描述
CM200_CANReset	重置所指定之板卡 CAN 控制器
CM200_CANGetStatus	讀取所指定之板卡 CAN 埠狀態
CM200_EnableCAN	啟用所指定之板卡 CAN 埠功能
CM200_DisableCAN	關閉所指定之板卡 CAN 埠功能
CM200_SetCANConfig	設定所指定之板卡 CAN 埠設定參數
CM200_GetCANConfig	讀取所指定之板卡 CAN 埠設定參數
CM200_AddCyclicTxMsg	由指定之板卡新增一筆循環傳送 CAN 訊息功能
CM200_DeleteCyclicTxMsg	由指定之板卡刪除一筆循環傳送 CAN 訊息功能
CM200_EnableCyclicTxMsg	由指定之板卡啟用一筆循環傳送 CAN 訊息功能
CM200_DisableCyclicTxMsg	由指定之板卡關閉一筆循環傳送 CAN 訊息功能
CM200_CheckCyclicTxRestMsg	由指定之板卡檢查循環傳送 CAN 訊息之剩餘傳送數量
CM200_IsTxTimeout	檢查所傳送的 CAN 訊息是否成功送出
CM200_CANInit	初始化所指定之板卡 CAN 埠功能
CM200_RxMsgCount	讀取所指定之板卡 CAN 埠的 CAN 訊息接收筆數
CM200_ReceiveCANMsg	由指定之板卡 CAN 埠接收一筆 CAN 訊息
CM200_SendCANMsg	由指定之板卡 CAN 埠傳送一筆 CAN 訊息
CM200_ClearSoftBuffer	清除指定之板卡 CAN 埠的傳送、接收緩衝區資料
CM200_ClearTxSoftBuffer	清除指定之板卡 CAN 埠的傳送緩衝區資料
CM200_ClearRxSoftBuffer	清除指定之板卡 CAN 埠的接收緩衝區資料
CM200_ClearBufferStatus	清除指定之板卡 CAN 埠的接收緩衝區狀態

3.2.1. CM200_CANReset

重置所指定之 PISO-CM200U 系列版卡的 CAN 埠硬體功能。

語法

```
C#  
-----  
Int16 CM200_CANReset (  
    Byte BoardNo,  
    Byte Port  
);
```

參數

BoardNo

[in] 指定的板卡編號。

Port

[in] 指定的 CAN 埠。

數值 1:表示 CAN 埠 1。

數值 2:表示 CAN 埠 2。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第 3.3 章節的回傳值定義。

3.2.2. CM200_CANGetStatus

取得所指定之 PISO-CM200U 系列版卡，CAN 總線的狀態。

語法

```
C#
-----
Int16 CM200_CANGetStatus (
    Byte BoardNo,
    Byte Port,
    out Byte Status
);
```

參數

BoardNo

[in] 指定的板卡編號。

Port

[in] 指定的 CAN 埠。

數值 1:表示 CAN 埠 1。

數值 2:表示 CAN 埠 2。

Status

[out] 要獲取之版卡CAN總線狀態的變數位址。

Bit	Symbol	Value	Description
2:0	LEC		Last error code Type of the last error to occur on the CAN bus.The LEC field holds a code which indicates the type of the last error to occur on the CAN bus.
		0x0	No error.
		0x1	Stuff error: More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed.
		0x2	Form error: A fixed format part of a received frame has the wrong format.
		0x3	AckError: The message this CAN core transmitted was not acknowledged.
0x4	Bit1Error: During the transmission of a message (with the exception of the arbitration field), the device wanted to send a HIGH/recessive level (bit of logical value '1'), but the monitored bus		

			value was LOW/dominant.
		0x5	Bit0Error: During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device wanted to send a LOW/dominant level (data or identifier bit logical value '0'), but the monitored Bus value was HIGH/recessive.
		0x6	CRCErrror: The CRC checksum was incorrect in the message received.
		0x7	Unused: No CAN bus event was detected
3	TXOK		Transmitted a message successfully.
		0	No message has been successfully transmitted.
		1	A message has been successfully transmitted.
4	RXOK		Received a message successfully
		0	No message has been successfully received
		1	A message has been successfully received independent of the result of acceptance filtering.
5	EPASS		Error passive
		0	The CAN controller is in the error active state.
		1	The CAN controller is in the error passive state as defined in the CAN 2.0 specification.
6	EWARN		Warning status
		0	Both error counters are below the error warning limit of 96.
		1	At least one of the error counters in the Error Counter Register has reached the error warning limit of 96.
7	BOFF		Busoff status
		0	The CAN module is not in busoff state.
		1	The CAN controller is in busoff state.

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第 3.3 章節的回傳值定義。

3.2.3. CM200_EnableCAN

啟用所指定之 PISO-CM200U 系列，CAN 總線控制器的傳送/接收功能。

語法

```
C#  
-----  
Int16 CM200_EnableCAN (  
    Byte BoardNo,  
    Byte Port  
);
```

參數

BoardNo

[in] 指定的板卡編號。

Port

[in] 指定的 CAN 埠。

數值 1:表示 CAN 埠 1。

數值 2:表示 CAN 埠 2。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第 3.3 章節的回傳值定義。

3.2.4. CM200_DisableCAN

關閉所指定之 PISO-CM200U 系列，CAN 總線控制器的傳送/接收功能。

語法

C#

```
Int16 CM200_DisableCAN (  
    Byte BoardNo,  
    Byte Port  
);
```

參數

BoardNo

[in] 指定的板卡編號。

Port

[in] 指定的 CAN 埠。

數值 1:表示 CAN 埠 1。

數值 2:表示 CAN 埠 2。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第 3.3 章節的回傳值定義。

3.2.5. CM200_SetCANConfig

設定所指定之 PISO-CM200U 系列版卡的 CAN 總線控制器設定參數，如：通訊鮑率、位元取樣點、訊息過濾...等設定值。在設定成功後，所有的設定值將會寫入板卡的 EEPROM 中給版卡韌體使用。

語法

C#

```
Int16 CM200_SetCANConfig(  
    Byte BoardNo,  
    Byte Port,  
    UInt32 BaudRate,  
    UInt32 SamplePoint,  
    Byte FilterMode,  
    UInt32 FilterArbitBit,  
    UInt32 FilterMaskBit  
);
```

參數

BoardNo

[in] 指定的板卡編號。

Port

[in] 指定的 CAN 埠。

數值 1:表示 CAN 埠 1。

數值 2:表示 CAN 埠 2。

BaudRate

[in] CAN總線通訊用的鮑率，單位為bps，數值範圍由5,000至1,000,000。

SamplePoint

[in] CAN總線鮑率的位元取樣點，以百分比來表示。數值單位為0.01%。數值範圍由0 (表示0.00%)至10,000 (表示100.00%)。

FilterMode

[in] 訊息過濾模式。

數值0: 表示使用CAN2.0A規範11位元CAN ID的訊息過濾方式。

數值1: 表示使用CAN2.0B規範29位元CAN ID的訊息過濾方式。

FilterArbitBit

[in] FilterArbitBit是要用來決定要被CAN控制器所接收之CAN ID訊息樣式。

FilterMode	FilterArbitBit (16進制) 數值範圍
11-bit CAN ID	000 ~ 7FF
29-bit CAN ID	00000000 ~ 1FFFFFFF

FilterMaskBit

[in] FilterMaskBit是用來決定哪些與FilterArbitBit對應之相同位元的CAN ID訊息，需要被檢查，或需要被遮罩掉不被檢查。當FilterMaskBit之位元數值為1時，表示相對應於FilterArbitBit的位元數值需要被檢查，而當FilterMaskBit之位元數值為0時，表示相對應於FilterArbitBit的位元數值則不需要被檢查。

FilterMode	FilterMaskBit (16進制) 數值範圍
11-bit CAN ID	000 ~ 7FF
29-bit CAN ID	00000000 ~ 1FFFFFFF

當接收到之CAN訊息ID欄位與需要被檢查之FilterArbitBit不相同時，該筆CAN訊息將會被過濾掉，不被CAN控制器所接受。

29-bit CAN ID範例				
	(最高位元)			(最低位元)
CAN ID位元	31 ~ 24位元	23 ~ 16位元	15 ~ 8位元	7 ~ 0位元
FilterArbitBit (16進制)	00	00	00	A0
FilterMaskBit (16進制)	00	00	00	E0
可接收之CAN ID (x: 可忽略的位元數值)	XXXX XXXX	XXXX XXXX	XXXX XXXX	101x xxxx

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第 3.3 章節的回傳值定義。

3.2.6. CM200_GetCANConfig

讀取所指定之 PISO-CM200U 系列版卡的 CAN 總線控制器設定參數，如：通訊鮑率、位元取樣點、訊息過濾...等設定值。

語法

```
C#
-----
Int16 CM200_SetCANConfig(
    Byte BoardNo,
    Byte Port,
    out UInt32 BaudRate,
    out UInt32 SamplePoint,
    out Byte FilterMode,
    out UInt32 FilterArbitBit,
    out UInt32 FilterMaskBit
);
```

參數

BoardNo

[in] 指定的板卡編號。

Port

[in] 指定的 CAN 埠。

數值 1:表示 CAN 埠 1。

數值 2:表示 CAN 埠 2。

BaudRate

[out] 要獲取之CAN總線通訊用的鮑率變數位址，單位為bps，數值範圍由5,000至1,000,000。

SamplePoint

[out] 要獲取之CAN總線鮑率的位元取樣點變數位址，以百分比來表示。數值單位為0.01%。數值範圍由0 (表示0.00%)至10,000 (表示100.00%)。

FilterMode

[out] 要獲取之訊息過濾模式的變數位址。

數值0: 表示使用CAN2.0A規範之11位元CAN ID的訊息過濾方式。

數值1: 表示使用CAN2.0B規範之29位元CAN ID的訊息過濾方式。

FilterArbitBit

[out] 要獲取之FilterArbitBit的變數位址。

FilterArbitBit是要用來決定要被CAN控制器所接收之CAN ID訊息樣式。

FilterMode	FilterArbitBit (16進制) 數值範圍
11-bit CAN ID	000 ~ 7FF
29-bit CAN ID	00000000 ~ 1FFFFFFF

FilterMaskBit

[out] 要獲取之FilterMaskBit的變數位址。

FilterMaskBit是用來決定哪些與FilterArbitBit對應之相同位元的CAN ID訊息，需要被檢查，或需要被遮罩掉不被檢查。當FilterMaskBit之位元數值為1時，表示相對應於FilterArbitBit的位元數值需要被檢查，而當FilterMaskBit之位元數值為0時，表示相對應於FilterArbitBit的位元數值則不需要被檢查。

FilterMode	FilterMaskBit (16進制) 數值範圍
11-bit CAN ID	000 ~ 7FF
29-bit CAN ID	00000000 ~ 1FFFFFFF

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第 3.3 章節的回傳值定義。

3.2.7. CM200_AddCyclicTxMsg

新增一筆循環傳送 CAN 訊息功能至所指定之 PISO-CM200U 系列版卡的韌體中。然後，使用者可以透過 CM200_EnableCyclicTxMsg()、CM200_DisableCyclicTxMsg()及 CM200_DelectCyclicTxMsg()等應用程式介面函式來啟用、關閉、刪除這些循環傳送 CAN 訊息功能。最大可新增 5 筆循環傳送 CAN 訊息功能。當成功新增一筆循環傳送 CAN 訊息功能時，會回傳一代碼(handle)，用來表示目前所新增之循環傳送 CAN 訊息功能；回傳之代碼(handle)數值越小時，表示此筆循環傳送 CAN 訊息功能的優先執行權越高。當有兩筆循環傳送 CAN 訊息功能需要同時被執行時，優先執行權越高的訊息將會先被執行。

語法

```
C#
-----
Int16 CM200_AddCyclicTxMsg(
    Byte BoardNo,
    Byte Port,
    Byte Mode,
    UInt32 MsgID,
    Byte RTR,
    Byte DataLen,
    Byte[] Data,
    UInt32 TimePeriod,
    UInt32 TransmitTimes,
    out Byte Handle
);
```

參數

BoardNo

[in] 指定的板卡編號。

Port

[in] 指定的 CAN 埠。

數值 1:表示 CAN 埠 1。

數值 2:表示 CAN 埠 2。

Mode

[in] 要傳送之CAN總線訊息ID格式。

數值0: 表示使用CAN2.0A規範之11位元CAN ID的訊息格式。

數值1: 表示使用CAN2.0B規範之29位元CAN ID的訊息格式。

MsgID

[in] CAN總線訊息ID。

RTR

[in] 要傳送之CAN總線訊息資料格式。

數值0: 表示使用CAN總線資料通訊格式。

數值1: 表示使用CAN總線遠端通訊格式。

DataLen

[in] 要傳送之CAN總線訊息資料的長度，最大值為8。

Data

[in] 要傳送之CAN總線訊息資料陣列的初始位址，使用之資料陣列長度的最大值為8。

TimePeriod

[in] 要使用之循環傳送CAN訊息功能的傳送週期。此參數數值單位為0.1毫秒。此參數數值最小值為5。

TransmitTimes

[in] 要使用之循環傳送CAN訊息功能的傳送筆數。當循環傳送CAN訊息功能啟用時，會依此參數的設定來傳送相對應之訊息筆數。當此參數的設定值為0時，表示要一直傳送，直到關閉此筆循環傳送CAN訊息功能。

Handle

[out] 用來取得此次所新增之循環傳送CAN訊息功能的代碼(handle)變數位址。使用者需要透過此代碼(handle)來啟用，關閉，或者刪除所設定之循環傳送CAN訊息功能。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第3.3章節的回傳值定義。

3.2.8. CM200_DeleteCyclicTxMsg

由所指定之 PISO-CM200U 系列版卡的韌體中，刪除一筆由 CM200_AddCyclicTxMsg()應用程式介面函式回傳的代碼(handle)所代表之循環傳送 CAN 訊息功能。

語法

```
C#
-----
Int16 CM200_DeleteCyclicTxMsg (
    Byte BoardNo,
    Byte Port,
    Byte Handle
);
```

參數

BoardNo

[in] 指定的板卡編號。

Port

[in] 指定的 CAN 埠。

數值 1:表示 CAN 埠 1。

數值 2:表示 CAN 埠 2。

Handle

[in] CM200_AddCyclicTxMsg()應用程式介面函式所回傳之循環傳送 CAN 訊息功能的代碼(handle)。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第3.3章節的回傳值定義。

3.2.9. CM200_EnableCyclicTxMsg

由所指定之PISO-CM200U系列版卡的韌體中，啟用一筆由CM200_AddCyclicTxMsg()應用程式介面函式回傳的代碼(handle)所代表之循環傳送CAN訊息功能。當此應用程式介面函式成功執行時，該筆循環傳送CAN訊息功能會開始被執行。

語法

```
C#
-----
Int16 CM200_EnableCyclicTxMsg (
    Byte BoardNo,
    Byte Port,
    Byte Handle
);
```

參數

BoardNo

[in] 指定的板卡編號。

Port

[in] 指定的 CAN 埠。

數值 1:表示 CAN 埠 1。

數值 2:表示 CAN 埠 2。

Handle

[in] CM200_AddCyclicTxMsg()應用程式介面函式所回傳之循環傳送 CAN 訊息功能的代碼(handle)。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第3.3章節的回傳值定義。

3.2.10. CM200_DisableCyclicTxMsg

由所指定之PISO-CM200U系列版卡的韌體中，停用一筆由CM200_AddCyclicTxMsg()應用程式介面函式回傳的代碼(handle)所代表之循環傳送CAN訊息功能。當此應用程式介面函式成功執行時，該筆循環傳送CAN訊息功能會停止被執行。

語法

```
C#
-----
Int16 CM200_DisableCyclicTxMsg (
    Byte BoardNo,
    Byte Port,
    Byte Handle
);
```

參數

BoardNo

[in] 指定的板卡編號。

Port

[in] 指定的 CAN 埠。

數值 1:表示 CAN 埠 1。

數值 2:表示 CAN 埠 2。

Handle

[in] CM200_AddCyclicTxMsg()應用程式介面函式所回傳之循環傳送 CAN 訊息功能的代碼(handle)。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第3.3章節的回傳值定義。

3.2.11. CM200_CheckCyclicTxRestMsg

由所指定之 PISO-CM200U 系列版卡的韌體中，取得由 CM200_AddCyclicTxMsg() 應用程式介面函式回傳的代碼(handle)所代表之循環傳送 CAN 訊息功能之剩餘傳送次數。

語法

```
C#
-----
Int16 CM200_CheckCyclicTxRestMsg (
    Byte BoardNo,
    Byte Port,
    Byte Handle,
    out UInt32 RestCount
);
```

參數

BoardNo

[in] 指定的板卡編號。

Port

[in] 指定的 CAN 埠。

數值 1:表示 CAN 埠 1。

數值 2:表示 CAN 埠 2。

Handle

[in] CM200_AddCyclicTxMsg()應用程式介面函式所回傳之循環傳送 CAN 訊息功能的代碼(handle)。

RestCount

[out] 用來取得循環傳送 CAN 訊息功能所傳送之剩餘次數的變數位址。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第3.3章節的回傳值定義。

3.2.12. CM200_IsTxTimeout

取得 CAN 控制器是否成功將 CAN 訊息傳送至 CAN 網絡上之傳送狀態。

語法

```
C#
-----
Int16 CM200_IsTxTimeout (
    Byte BoardNo,
    Byte Port,
    out Byte Status
);
```

參數

BoardNo

[in] 指定的板卡編號。

Port

[in] 指定的 CAN 埠。

數值 1:表示 CAN 埠 1。

數值 2:表示 CAN 埠 2。

Status

[out] 傳送狀態

數值 0: 表示無錯誤。

數值 1: 表示 CAN 控制器無法成功將 CAN 訊息傳送至 CAN 網絡上。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第3.3章節的回傳值定義。

3.2.13. CM200_CANInit

初始化所指定之 PISO-CM200U 系列版卡的 CAN 埠功能。

語法

```
C#  
-----  
Int16 CM200_CANInit (  
    Byte BoardNo,  
    Byte Port  
);
```

參數

BoardNo

[in] 指定的板卡編號。

Port

[in] 指定的 CAN 埠。

數值 1:表示 CAN 埠 1。

數值 2:表示 CAN 埠 2。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第 3.3 章節的回傳值定義。

3.2.14. CM200_RxMsgCount

取得所指定之 PISO-CM200U 系列版卡 CAN 埠所接收之 CAN 訊息數量。

語法

```
C#  
-----  
Int16 CM200_RxMsgCount(  
    Byte BoardNo,  
    Byte Port,  
    out UInt16 MsgCount  
);
```

參數

BoardNo

[in] 指定的板卡編號。

Port

[in] 指定的 CAN 埠。

數值 1:表示 CAN 埠 1。

數值 2:表示 CAN 埠 2。

MsgCount

[out] CAN埠所接收之CAN訊息數量。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第 3.3 章節的回傳值定義。

3.2.15. CM200_ReceiveCANMsg

由所指定之 PISO-CM200U 系列版卡韌體的 CAN 埠接收緩衝區內取得一筆 CAN 訊息。

語法

C#

```
Int16 CM200_ReceiveCANMsg (  
    Byte BoardNo,  
    Byte Port,  
    out Byte Mode,  
    out UInt32 MsgID,  
    out Byte RTR,  
    out Byte DataLen,  
    Byte[] Data,  
    out UInt32 UpperTime,  
    out UInt32 LowerTime  
);
```

參數

BoardNo

[in] 指定的板卡編號。

Port

[in] 指定的 CAN 埠。

數值 1:表示 CAN 埠 1。

數值 2:表示 CAN 埠 2。

Mode

[out] 用來取得接收到之CAN總線訊息ID格式的變數位址。

數值0: 表示接收到CAN2.0A規範之11位元CAN ID的訊息格式。

數值1: 表示接收到CAN2.0B規範之29位元CAN ID的訊息格式。

MsgID

[out] 用來取得接收到之CAN總線訊息ID的變數位址。

RTR

[out] 用來取得接收到之CAN總線訊息資料格式的變數位址。

數值0: 表示使用CAN總線資料通訊格式。

數值1: 表示使用CAN總線遠端通訊格式。

DataLen

[out] 用來取得接收到之CAN總線訊息資料的長度的變數位址，最大值為8。

Data

[out] 用來取得接收到之CAN總線訊息資料的陣列變數位址，使用之資料陣列長度的最大值為8。

UpperTime

[out] 用來取得接收到之CAN總線訊息的高32位元的時間的變數位址。當LowerTime參數溢位時，此數值會依序遞增。

LowerTime

[out] 用來取得接收到之CAN總線訊息的低32位元的時間的變數位址。時間單位為0.1毫秒，每0.1毫秒此LowerTime參數會依序遞增。數值範圍為0 ~ 4,294,967,295。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第 3.3 章節的回傳值定義。

3.2.16. CM200_SendCANMsg

由所指定之 PISO-CM200U 系列版卡韌體的 CAN 埠，傳送一筆 CAN 訊息至傳送緩衝區內。當 CAN 總線是空間時，傳送緩衝區內的資料將會被傳送至 CAN 網絡上。當 CAN 總線有異常時，傳送緩衝區內的資料將不會被傳送。使用者可以透過 CM200_CANGetStatus()應用程式介面函式來取得當下的 CAN 網絡狀態。

語法

```
C#
-----
Int16 CM200_SendCANMsg (
    Byte BoardNo,
    Byte Port,
    Byte Mode,
    UInt32 MsgID,
    Byte RTR,
    Byte DataLen,
    Byte[] Data,
);
```

參數

BoardNo

[in] 指定的板卡編號。

Port

[in] 指定的 CAN 埠。

數值 1:表示 CAN 埠 1。

數值 2:表示 CAN 埠 2。

Mode

[in] 預傳送之CAN總線訊息ID格式。

數值0: 表示接收到CAN2.0A規範之11位元CAN ID的訊息格式。

數值1: 表示接收到CAN2.0B規範之29位元CAN ID的訊息格式。

MsgID

[in] 預傳送之之CAN總線訊息ID。

RTR

[in] 預傳送之CAN總線訊息資料格式。

數值0: 表示使用CAN總線資料通訊格式。

數值1: 表示使用CAN總線遠端通訊格式。

DataLen

[in] 預傳送之CAN總線訊息資料的長度，最大值為8。

Data

[in] 預傳送之CAN總線訊息資料陣列，使用之資料陣列長度的最大值為8。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第 3.3 章節的回傳值定義。

3.2.17. CM200_ClearSoftBuffer

清除所指定之 PISO-CM200U 系列版卡 CAN 埠的傳送及接收資料緩衝區內的訊息。

語法

```
C#  
-----  
Int16 CM200_ClearSoftBuffer (  
    Byte BoardNo,  
    Byte Port,  
);
```

參數

BoardNo

[in] 指定的板卡編號。

Port

[in] 指定的 CAN 埠。

數值 1:表示 CAN 埠 1。

數值 2:表示 CAN 埠 2。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第 3.3 章節的回傳值定義。

3.2.18. CM200_ClearTxSoftBuffer

清除所指定之 PISO-CM200U 系列版卡 CAN 埠的傳送資料緩衝區內的訊息。

語法

```
C#  
-----  
Int16 CM200_ClearTxSoftBuffer (  
    Byte BoardNo,  
    Byte Port,  
);
```

參數

BoardNo

[in] 指定的板卡編號。

Port

[in] 指定的 CAN 埠。

數值 1:表示 CAN 埠 1。

數值 2:表示 CAN 埠 2。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第 3.3 章節的回傳值定義。

3.2.19. CM200_ClearRxSoftBuffer

清除所指定之 PISO-CM200U 系列版卡 CAN 埠的接收資料緩衝區內的訊息。

語法

```
C#  
-----  
Int16 CM200_ClearRxSoftBuffer (  
    Byte BoardNo,  
    Byte Port,  
);
```

參數

BoardNo

[in] 指定的板卡編號。

Port

[in] 指定的 CAN 埠。

數值 1:表示 CAN 埠 1。

數值 2:表示 CAN 埠 2。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第 3.3 章節的回傳值定義。

3.2.20. CM200_ClearBufferStatus

清除所指定之 PISO-CM200U 系列版卡 CAN 埠的傳送及接收資料緩衝區的狀態及資料。當使用 `CM200_ReceiveCANMsg()`、`CM200_SendCANMsg()` 等應用程式介面函式時，使用者也許會收到緩衝區異常之回傳碼，此時就可以使用此函式來清除緩衝區的內容及重置緩衝區的狀態。

語法

```
C#
-----
Int16 CM200_ClearBufferStatus (
    Byte BoardNo,
    Byte Port,
);
```

參數

BoardNo

[in] 指定的板卡編號。

Port

[in] 指定的 CAN 埠。

數值 1:表示 CAN 埠 1。

數值 2:表示 CAN 埠 2。

回傳值

當此函式成功執行時，回傳值為 0。

當此函式執行失敗時，請參考第 3.3 章節的回傳值定義。

3.3. 回傳碼定義

PISO-CM200U 系列所支援之應用程式介面函式回傳碼及內容描述條列於下表。

回傳碼 (16 禁制)	代碼名稱	錯誤訊息描述
0x000	ERR_NO_ERR	無錯誤
0x001	ERR_INIT_DRIVER_ERROR	驅動程式初始化失敗
0x002	ERR_COMM_DRIVER_ERROR	無法與驅動程式通訊
0x003	ERR_DRIVER_UNINIT_ERROR	驅動程式沒有初始化
0x004	ERR_ACTIVE_BOARD_ERROR	版卡啟用錯誤
0x005	ERR_BOARD_ALREADY_ACTIVE_ERROR	版卡已經啟用
0x006	ERR_BOARD_UNACTIVE_ERROR	版卡沒有啟用
0x007	ERR_BOARD_NUMBER_ERROR	超過支援的板卡數量上限
0x008	ERR_PORT_NUMBER_ERROR	超過支援的 CAN 埠上限
0x009	ERR_DPRAM_ADDR_OVER_RANGE	超過所支援之 DPRAM 位址上限
0x00A	ERR_FIRMWARE_MODE_ERROR	韌體運行的模式錯誤
0x00B	ERR_SIZE_OF_NAME_OVER_RANGE	超過支援的檔名上限
0x00C	ERR_SIZE_OF_DATA_OVER_RANGE	超過支援的檔案大小上限
0x00D	ERR_ALLOC_DATA_BUFFER_ERROR	配置資料記憶體空間錯誤
0x101	ERR_SYS_COMM_ACK_TIMEOUT	與韌體通訊逾時
0x102	ERR_SYS_BOARD_ALREADY_ACTIVE	版卡已經啟用
0x103	ERR_SYS_SET_DATETIME_ERROR	設定韌體時間錯誤
0x104	ERR_SYS_FIRMWARE_MODE_ERROR	韌體運行的模式錯誤
0x105	ERR_SYS_INSTALL_IRQ_ERROR	安裝系統中斷錯誤
0x106	ERR_SYS_ENABLE_PLX_INT_ERROR	啟用系統中斷錯誤
0x107	ERR_SYS_DISABLE_PLX_INT_ERROR	關閉系統中斷錯誤
0x108	ERR_SYS_PORT_NUMBER_ERROR	超過支援的 CAN 埠上限
0x109	ERR_SYS_CAN_CONFIG_ERROR	設定 CAN 埠參數錯誤
0x10A	ERR_SYS_SET_CYCLIC_MSG_ERROR	設定循環傳送 CAN 訊息功能錯誤
0x10B	ERR_SYS_SET_DEBUG_MSG_ERROR	設定除錯訊息功能錯誤
0x10C	ERR_SYS_SOFTBUFF_IS_EMPTY	CAN 端軟體緩衝區無資料
0x10D	ERR_SYS_SOFTBUFF_IS_FULL	CAN 端軟體緩衝區溢位
0x10E	ERR_SYS_USERISR_ALREADY_INSTALL	已安裝使用者自定義之中斷功能
0x10F	ERR_SYS_NO_DPRAM_COMMDAND	沒有回傳任何 DPRAM 的命令

0x110	ERR_SYS_DOWNLOAD_DATA_ERROR	資料下載至韌體錯誤
0x111	ERR_SYS_GET_FW_INFO_ERROR	讀取韌體資訊錯誤
0x112	ERR_SYS_SET_FW_INFO_ERROR	寫入韌體資訊錯誤
0x113	ERR_SYS_NO_FW_EXIST_ERROR	無韌體存在
0x114	ERR_SYS_SIZE_OF_DATA_OVER_RANGE	寫入之檔案資料大小溢位

4. 韌體更新

請參考下列步驟來更新 PISO-CM200U 系列版卡韌體。

步驟 1: 取得”FW_Update_Tool” 韌體更新工具及韌體檔案。

”FW_Update_Tool” 韌體更新工具可以由下列光碟路徑取得:

CD:\can\pci\piso-cm200u\software\tools

也可由下列 ICP DAS 網站取得

ftp://ftp.icpdas.com/pub/cd/fieldbus_cd/can/pci/piso-cm200u/software/tools

or

http://ftp.icpdas.com/pub/cd/fieldbus_cd/can/pci/piso-cm200u/software/tools

韌體檔案可以由下列光碟路徑取得:

CD:\can\pci\piso-cm200u\firmware

也可由下列 ICP DAS 網站取得

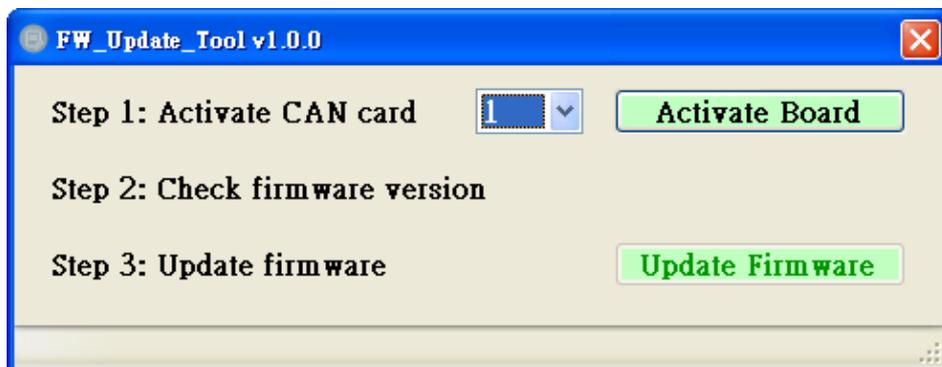
ftp://ftp.icpdas.com/pub/cd/fieldbus_cd/can/pci/piso-cm200u/firmware

or

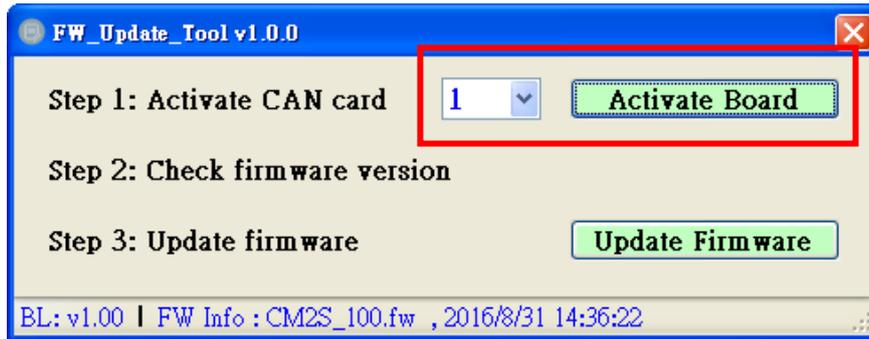
http://ftp.icpdas.com/pub/cd/fieldbus_cd/can/pci/piso-cm200u/firmware

步驟 2: 安裝 [.NET Framework 3.5](#) 套件，執行”FW_Update_Tool” 韌體更新工具。

”FW_Update_Tool” 韌體更新工具需要使用到 .NET Framework 3.5 開發套件。如使用者的系統環境無安裝過此開發套件的話，在執行”FW_Update_Tool” 韌體更新工具時，會要求使用者先下載並安裝此開發套件。

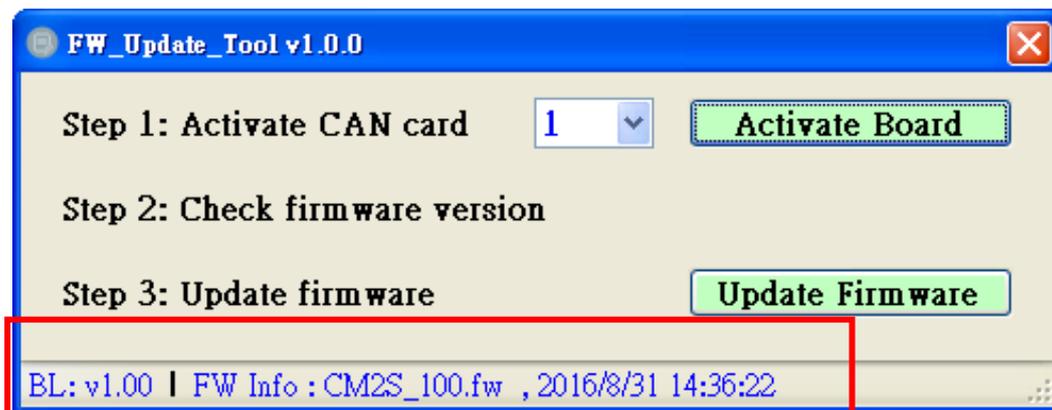


步驟 3: 選擇所要更新韌體之板卡編號後，按下“Activate Board” 按鈕，啟用所選擇之板卡。

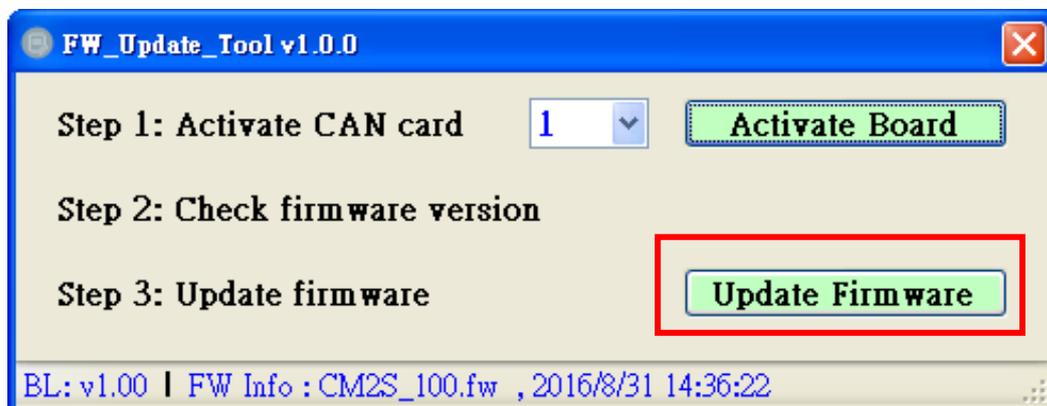


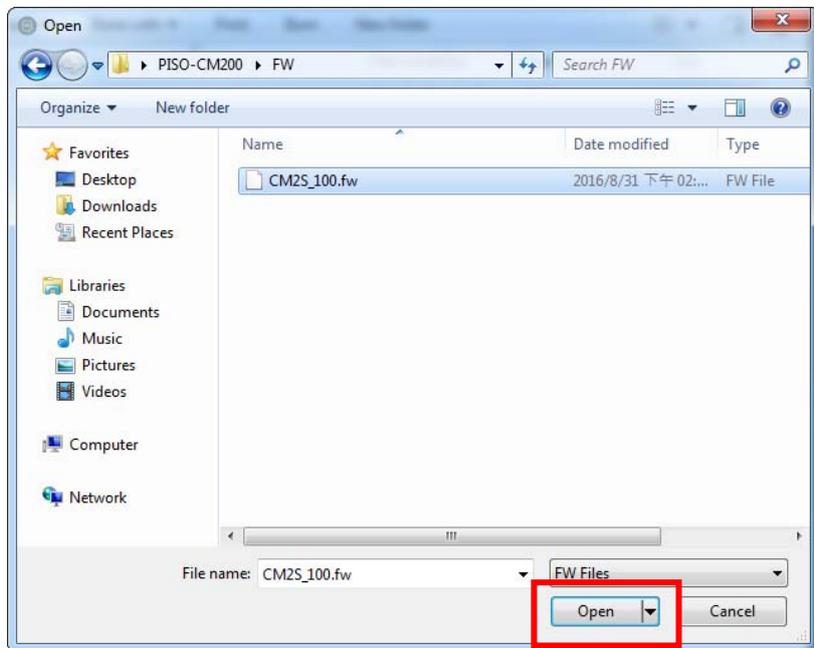
步驟 4: 由狀態欄位檢查目前板卡之韌體版本(FW Info 欄位)。

“CS2S_xxx.fw” 表示目前板卡之韌體檔案名稱，xxx 表示韌體版本。後方之日期則為檔案修改日期。

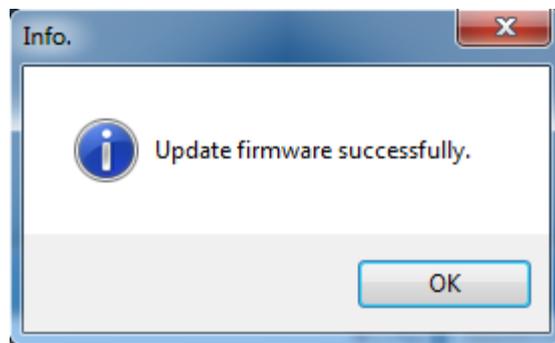


步驟 5: 如果韌體需要更新，按下“Update Firmware”按鈕，選擇要下載之韌體檔案後，將韌體更新製版卡內。





步驟 6: 更新完韌體後，會顯示安裝成功訊息。請關閉訊息視窗及程式，開始使用版卡。



5. 附錄

5.1. 防電磁干擾磁環



EMI Ferrite Split/Snap-On Core

Features

- Aimed to suppress low frequency noise generated by engine control units, inverters, and motors
- Split type
- Operation Temperature: -25°C ~ 75°C



Introduction

The split ferrite cable cores are designed to significantly reduce EMI/RFI on round cables.

The hinged plastic case surrounding the split core is designed to clamp onto the cable to provide a secure fixture of the ferrite onto the cable. The cores can be retrofitted onto existing installations or used in post-assembly operations on the data and power cables of electronic equipment. Ferrite cores are important for ensuring strong electronic signals through cables in environments where EMI or RFI can be an issue.

Applications

RS-232, RS-422, RS-485, CAN bus, FRnet, PROFIBUS, Ethernet, USB, AC/DC Power line..etc

Inverter



Noise



Inverter



Core



Noise is reduced



RS-232



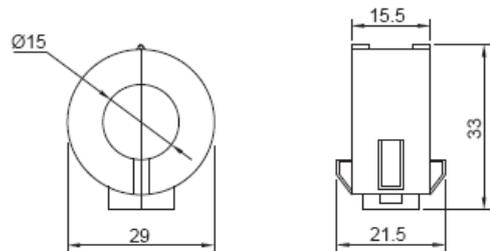
PROFIBUS

Specifications

Mechanical

Max. Cable Diameter	Ø15 mm
Material Type	Board Band Material
Additional Description	Plastic Case
Case Color	Black

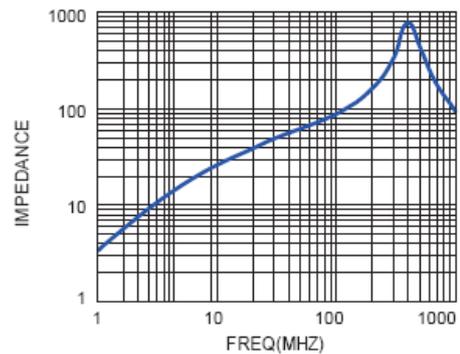
Dimensions (Units: mm)



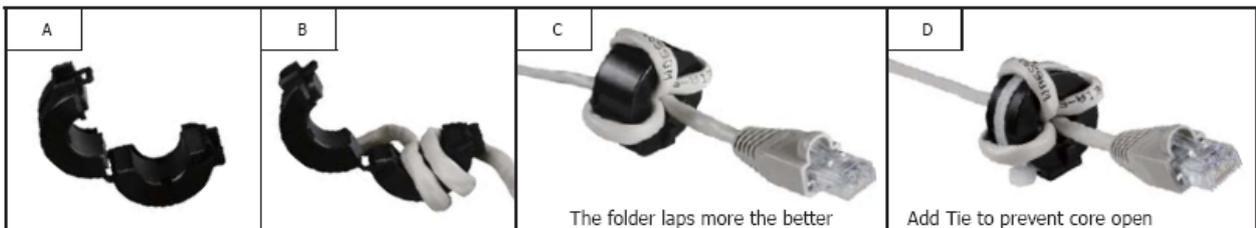
Front View

Left Side View

Characteristic



Installation



Clip-on Ferrite Core Installation