

# **NEXTW**

ECM-PCle

EtherCAT Master Control Card

**使用手冊**

2019/12/01

Ver. 1.1

## 版本紀錄

版本號	日期	版本描述
1.0	2019/04/01	第一版
1.01	2019/10/09	修正規格及驅動範例API介紹
1.1	2019/12/01	修正驅動程式安裝

# 目錄

1.簡介	3
1.1 特色	3
1.2 規格	3
1.3 軟體支援	3
1.4 介面	4
2. 安裝及試運行	5
2.1 包裝內容物	5
2.2 硬體及裝置驅動安裝	5
2.3 ECM-PCIe試運行	6
3. ECM-PCIe範例程式	6
3.1 ECM-PCIe SDK軟體開發套件	6
3.2 ECM-PCIe驅動函式庫	6
3.3 ECM-PCIe SDK使用步驟	10

## 1. 簡介

ECM-PCIe是專為PC-Base控制器開發的EtherCAT運動控制軸卡，使用NEXTW開發的高性能主站晶片EC-01M，支援最多64個從站裝置，最小週期時間500us。

### 1.1 特色

- PCI Express 2.0 X1介面
- 可連接64個從站裝置
- 最小週期時間500us
- FIFO緩存空間80筆

### 1.2 規格

系統	
介面	Compliant with PCI Express 2.0 Single-lane (X1) PCI Express
EtherCAT通訊	
通訊協議	CANOpen over EtherCAT(CoE)
週期時間(us)	500(MIN.)
支援從站裝置數量	64

### 1.3 軟體支援

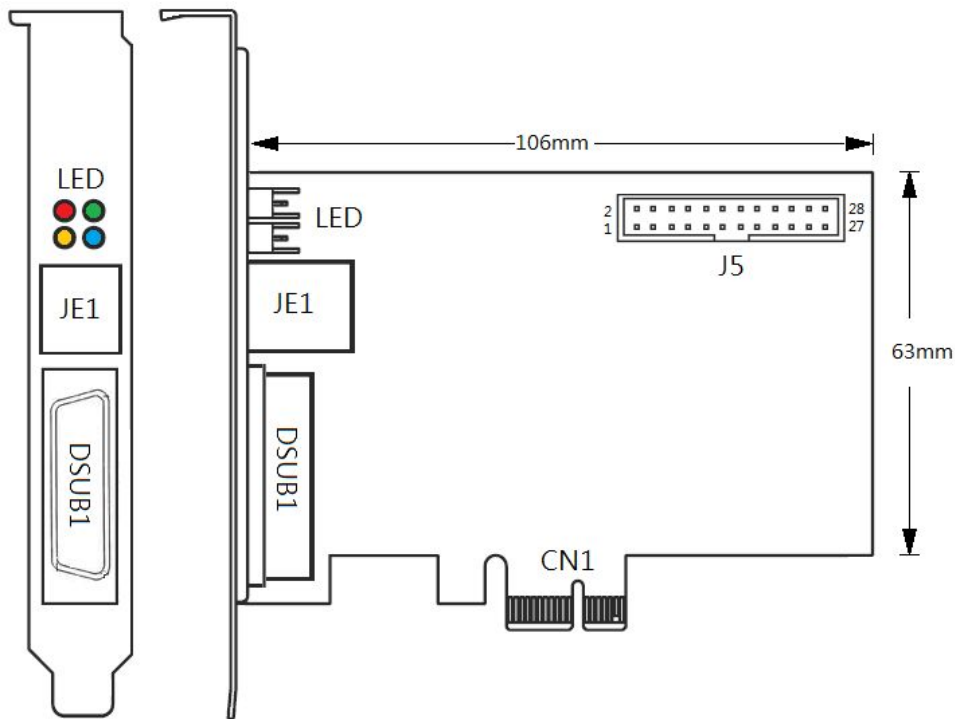
ECM-PCIe支援Windows 7/10, 64位元/32位元並提供NEXTW ECM-PCIe SDK軟體開發套件，以利使用者開發應用程式及測試ECM-PCIe裝置。

NEXTW ECM-PCIe SDK包含

- ECM-PCIe裝置驅動程式

- 驅動函式庫
- NEXTW ECM ITE整合測試程式

## 1.4 介面



- EtherCAT網路連接埠(JE1)

腳位	訊號	腳位	訊號
1	TX+	2	TX-
3	RX+	4	NC
5	NC	6	RX-
7	NC	8	NC

- 狀態指示燈(LED)

燈色	狀態	燈色	狀態
Red	Power	Green	PreOP
Yellow	SafeOP	Blue	OP

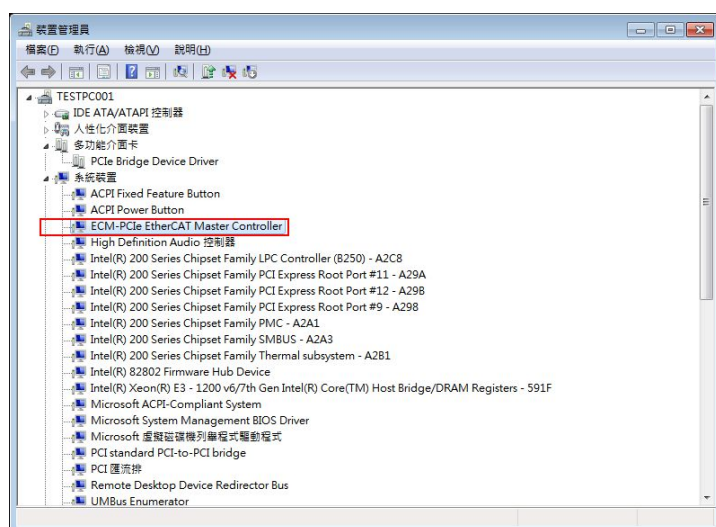
## 2. 安裝及試運行

### 2.1 包裝內容物

- ECM-PCIe卡
- 擋板
- DSUB26(DSUB1)轉接線(選購)
- 28pin(J5)轉DSUB26轉接線(選購)
- DSUB26轉接板(選購)

### 2.2 硬體及裝置驅動安裝

1. 打開包裝盒，確認內容物是否缺件。
2. 去除身體靜電後，打開靜電袋將ECM-PCIe卡取出，取下DSUB1連接頭螺絲，裝上擋板後將螺絲鎖回以固定擋板。
3. 確認電腦電源已關閉，確保採取適當的ESD(靜電放電)保護措施後打開機殼。
4. 選擇相容PCI Express 2.0的X1/X4/X8/X16 PCIe插槽，確認插槽無保護裝置及機殼無後檔片，插上ECM-PCIe卡。
5. 下載ECM-PCIe裝置驅動程式，解壓縮後執行Setup.exe，依指示完成安裝。
6. 完成驅動程式軟體安裝後，**ECM-PCIe EtherCAT Master Control Card**將出現在**系統裝置列表**下。



## 2.3 ECM-PCIe試運行

- a. 透過JE1連接EtherCAT從站裝置，建議使用具屏蔽CAT5e網線連接。
- b. 確認從站裝置配置是否正確及狀態是否正常，如:極限開關、原點開關、緊急停止開關....。
- c. 打開NEXTW ECM ITE整合測試程式。
- d. 點選Open按鍵開啟ECM-PCIe裝置。
- e. 配置EtherCAT從站裝置數量及類型。
- f. 選擇同步模式及同步週期。
- g. 設定從站驅動器模式(CSP/CSV)。
- h. 點選Connect按鍵，啟動EtherCAT通訊，ECM-PCIe會依照先前步驟設定完成配置，並將從站狀態切換至OP狀態。
- i. 從站裝置類型為IO裝置，可以直接透過控制項設定輸出狀態並觀察輸入狀態。
- j. 從站裝置類型為驅動器裝置，可以透過點選ServoOn按鍵使驅動器進入ServoOn狀態，此時驅動器及馬達一切就緒等待接收運動命令，透過視窗控制項送出適當的位置命令或速度命令，觀察驅動器及馬達狀態是否正常。

## 3. ECM-PCIe範例程式

### 3.1 ECM-PCIe SDK軟體開發套件

- NEXTWPCleLib.lib : PCIe裝置驅動函式庫
- NEXTWPCleLib.dll : PCIe裝置驅動函式庫動態連結檔
- driver.h : ECM-PCIe驅動範例標頭檔
- driver.cpp : ECM-PCIe驅動範例源碼檔
- EcmExample.cpp : ECM-PCIe驅動使用範例

### 3.2 ECM-PCIe驅動函式庫

int ECM_OpenDevice()	
功能描述	開啟ECM-PCIe裝置

參數	無
返回值	0: 成功 ; 其他 : 失敗

void ECM_CloseDevice()	
功能描述	關閉ECM-PCIe裝置
參數	無
返回值	無

int ECM_Initialize(int nCycTime, char* bNetworkMap, bool isDCEnable, unsigned short* slavefoundOut)	
功能描述	初始化EtherCAT通訊
參數	int nCycTime : EtherCAT DC時間(單位: us) char* bNetworkMap : EtherCAT網路裝置拓樸 0x08 : CiA402驅動器類型裝置(CSP模式) 0x09 : CiA402驅動器類型裝置(CSV模式) 0x0A : CiA402驅動器類型裝置(CST模式) 0x10 : EtherCAT IO類型裝置 bool isDCEnable: 啟用DC同步模式 unsigned short* slavefoundOut : 實際從站數量返回值
返回值	0:成功 ; 其他: 失敗

void SetServo(char* bServoMap)	
功能描述	將從站切換至操作狀態(Operation State), 驅動器從站將會ServoON。
參數	char* bServoMap : 一維陣列, 各從站是否切換至操作狀態。 1: SERVO-ON 0: SERVO-OFF



返回值	無
-----	---

unsigned short CSDataExchange(char* pbMaskMap, int* pnTXD, int* pnRXD);	
功能描述	周期同步資料交換
參數	char* pbMaskMap: 周期命令遮罩 0 : 不更新對應從站周期命令 1 : 更新對應從站周期命令 int* pnTXD: 周期循環資料輸出值 int* pnRXD: 周期循環資料輸入值
返回值	0xFFFF : CRC檢查錯誤 Others : FIFO剩餘數量

int ReadValue(int* pnRXD)	
功能描述	讀取周期同步資料。
參數	int *nValueOut : 周期循環資料輸入值
返回值	0: 讀取成功 Others : 讀取失敗

int SDORead(unsigned short slave, unsigned short index, unsigned short subIndex, int* nValue)	
功能描述	執行SDO讀取
參數	unsigned short slave : 從站站號 1~64 unsigned short index : 物件索引號 unsigned short subIndex : 物件副索引號 int* nValue : 物件內容值
返回值	0:成功 Others: 失敗

int SDOWrite(unsigned short slave,unsigned short index,unsigned short subIndex,char objectSize, int value)	
功能描述	執行SDO寫入
參數	unsigned short slave : 從站站號 1~64 unsigned short index : 物件索引號 unsigned short subIndex : 物件副索引號 char objectSize : 物件位元組數 int value : 物件寫入值
返回值	0:成功 Others: 失敗

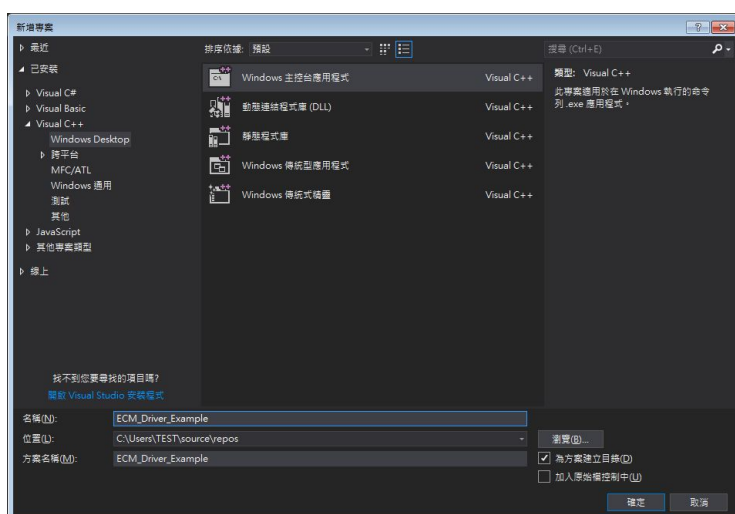
void ECM_ResetMotionCard()	
功能描述	重置ECM-PCIe卡於初始狀態
參數	無
返回值	無

void ECM_GoHome(char* bMaskMap, char* bHomeModeMap, int* nHomeOffset)	
功能描述	從站類型為驅動器，執行回零運動。
參數	char* bMaskMap : 0 : 不執行回零運動 1 : 執行回零運動 char* bHomeModeMap : 一維陣列，設置回零模式。 int* nHomeOffset : 一維陣列，設置回零完成後，位置偏移值。
返回值	無

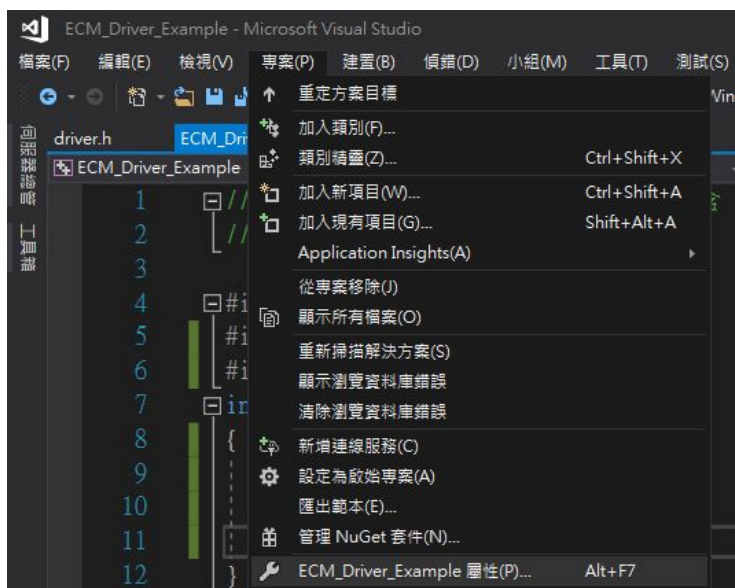
### 3.3 ECM-PCIe SDK使用步驟

以Visual C++為例

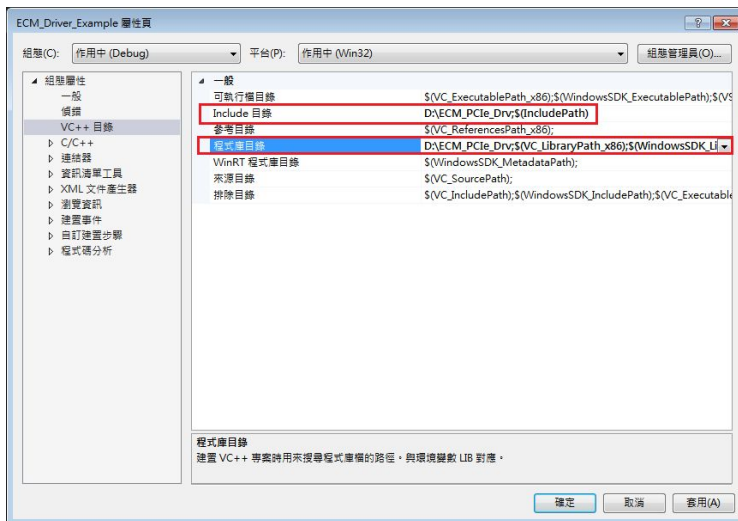
1. 開啟Visual Studio，選擇檔案->新增->專案，選擇Visual C++ -> Windows Desktop，選擇Windows主控台應用程式，輸入專案名稱後，按確定完成新增，回到Visual Studio C++開發環境。



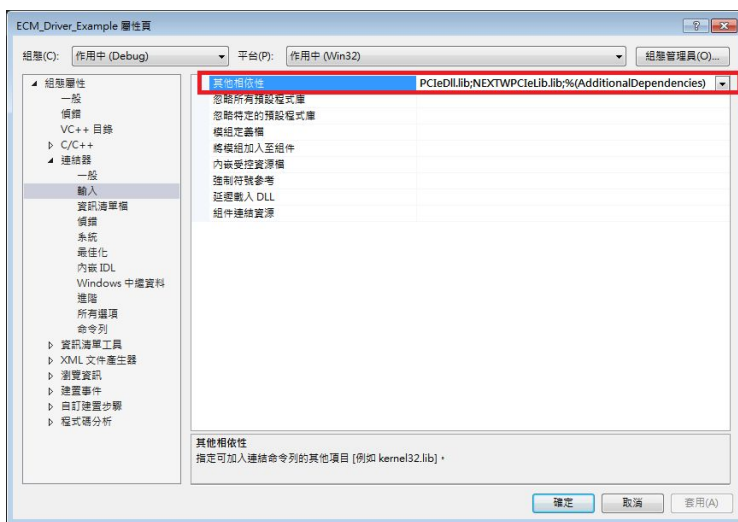
2. 設定ECM-PCIe SDK相關檔案所在路徑，點選專案->專案名稱屬性，或使用組合鍵Alt+F7。



3. 增加ECM-PCIe SDK相關檔案參考目錄：  
屬性頁左邊選取VC++目錄，右邊Include目錄及程式庫目錄裡，增加ECM-PCIe SDK路徑並用分號(;)隔開原設定值。



4. 增加ECM-PCIe 驅動函式庫：  
**屬性頁左邊選取連結器->輸入**，右邊**其他相依性**的欄位中增加**PCIeDLL.lib;NEXTWPCleLib.lib;**，完成配置後點選**確定**回到 Visual Studio開發環境。



5. 打開專案名稱.cpp，加入標頭檔driver.h，在main裡面呼叫**ECM\_OpenDevice()**，並顯示返回值。

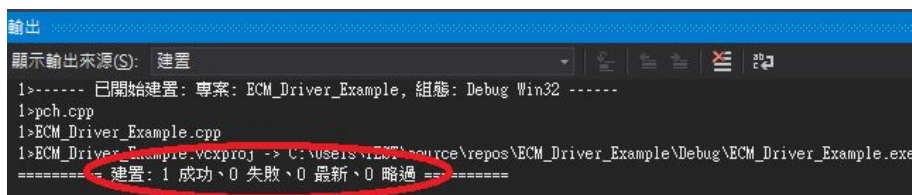
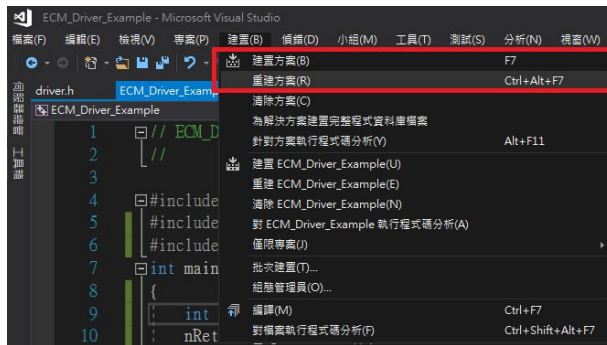
```

ECM_Driver_Example - Microsoft Visual Studio
driver.h  ECM_Driver_Example.cpp
ECM_Driver_Example

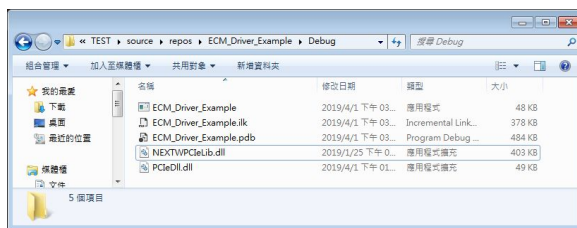
1  // ECM_Driver_Example.cpp : 此檔案包含 'main' 函式。
2  //
3
4  #include "pch.h"
5  #include <iostream>
6  #include <driver.h>
7  int main()
8  {
9      int nRet = 0;
10     nRet=ECM_OpenDevice();
11     std::cout << "ECM_OpenDevice return " << nRet;
12     ECM_CloseDevice();
13 }

```

6. 選擇**建置->建置方案**，確認建置成功，如果有錯誤產生確認前面步驟是否配置正確。



7. 複製**動態連結檔(DLL)**至專案執行檔所在目錄:  
**執行檔運行時會搜尋相依的動態連結檔**，如果當前目錄下沒有找到相依的動態連結檔會再去**系統目錄**底下搜尋。



8. 執行專案，並檢視執行結果。

