

# ECAT-261x EtherCAT 轉 Modbus RTU 閘道器使用手冊



繁體中文

1.5.0 版本, 2019 年 5 月

## 承諾

鄭重承諾：凡泓格科技股份有限公司產品從購買後，開始享有一年保固，除人為使用不當的因素除外。

## 責任聲明

凡使用本系列產品除產品品質所造成的損害，泓格科技股份有限公司不承擔任何的法律責任。泓格科技股份有限公司有義務提供本系列產品詳細使用資料，本使用手冊所提及的產品規格或相關資訊，泓格科技保留所有修訂之權利，本使用手冊所提及之產品規格或相關資訊有任何修改或變更時，恕不另行通知，本產品不承擔使用者非法利用資料對第三方所造成侵害構成的法律責任，未事先經由泓格科技書面允許，不得以任何形式複製、修改、轉載、傳送或出版使用手冊內容。

## 版權

版權所有 © 2019 泓格科技股份有限公司，保留所有權利。

## 商標

文件中所涉及所有公司的商標，商標名稱及產品名稱分別屬於該商標或名稱的擁有者所持有。

## 聯繫我們

如有任何問題歡迎聯繫我們，我們將會為您提供完善的諮詢服務。Email: [service@icpdas.com](mailto:service@icpdas.com)

## 支援

模組包含 ECAT-2610、ECAT-2610-DW  
及 ECAT-2611



# 目錄

配件清單.....	6
產品資訊.....	6
<b>1. 簡介 .....</b>	<b>7</b>
1.1 特色 .....	8
1.2 架構圖 .....	9
<b>2. 硬體資訊.....</b>	<b>10</b>
2.1 外觀 .....	10
2.2 規格 .....	12
2.3 腳位定義 .....	13
EtherCAT 介面.....	13
COM1 (Console Port) .....	13
COM2/COM3 (Modbus RTU) .....	13
2.4 RS-232/422/485 接線注意 .....	14
3 線 RS-232 接線 .....	14
4 線 RS-422 接線 .....	14
2 線 RS-485 接線 .....	14
2.5 INIT/NORMAL 運作模式 .....	15
2.6 機構圖 .....	16
<b>3. 啟用 ECAT-261X 模組.....</b>	<b>17</b>
3.1 出廠預設值 .....	17
3.2 連接主站與電源 .....	18
3.3 連接 POWER METER (適用 ECAT-2610-DW) .....	20
3.4 搜尋模組 .....	22
3.5 驗證模組狀態 .....	25
3.5.1 模組狀態及錯誤模式.....	27
ECAT-2610/2611 .....	28
ECAT-2610-DW .....	30
<b>4. 啟動 MODBUS RTU 設備 .....</b>	<b>32</b>
4.1 配置 MODBUS RTU 設備 .....	33
4.2 配置及上傳 .....	36
4.2.1 恢復至出廠預設值.....	44
4.3 測試 MODBUS RTU 設備 .....	47
<b>5. 啟動 MODBUS RTU MASTER .....</b>	<b>51</b>
5.1 配置及上傳 .....	51

5.2	連接 MODBUS RTU MASTER.....	56
5.3	測試 MODBUS RTU MASTER.....	60
<b>6.</b>	<b>MODBUS 資訊.....</b>	<b>65</b>
6.1	FC1(0x01) READ MULTIPLE COILS (0xxxx) FOR DO .....	68
6.2	FC2(0x02) READ MULTIPLE INPUT DISCRETE (1xxxx) FOR DI .....	69
6.3	FC3(0x03) READ MULTIPLE REGISTERS (4xxxx) FOR AO .....	70
6.4	FC4(0x04) READ MULTIPLE INPUT REGISTERS (3xxxx) FOR AI.....	71
6.5	FC5(0x05) WRITE SINGLE COIL (0xxxx) FOR DO .....	72
6.6	FC6(0x06) WRITE SINGLE REGISTER (4xxxx) FOR AO.....	73
6.7	FC15(0x0F) FORCE MULTIPLE COILS (0xxxx) FOR DO .....	74
6.8	FC16(0x10) WRITE MULTIPLE REGISTERS (4xxxx) FOR AO .....	75
6.9	FC255(0xFF) 特殊命令 .....	76
<b>7.</b>	<b>上傳配置檔操作 .....</b>	<b>77</b>
<b>8.</b>	<b>支援分散式時鐘(適用 ECAT-2610) .....</b>	<b>82</b>
8.1	MODBUS RTU 時間量測 .....	82
8.2	分散式時鐘(DC)配置與操作 .....	88
<b>9</b>	<b>OBJECT 說明及參數設定 .....</b>	<b>96</b>
9.1	標準 OBJECT .....	96
9.2	特定 OBJECT .....	97
	Input Buffer .....	97
	Output Buffer .....	97
<b>10</b>	<b>產品應用 .....</b>	<b>98</b>
10.1	泓格 ETHERCAT 系列產品 .....	98
10.2	洽詢 ODM.....	101
<b>附錄 .....</b>	<b>.....</b>	<b>104</b>
A1.	如何透過 DCON UTILITY 來取得 MODBUS RTU 指令.....	104
A2.	ECAT-2610 配置檔參考文件 .....	105
	00.Baudrate.....	106
	115200_N81_Init.txt .....	106
	9600_N81.txt .....	107
	19200_N82.txt.....	107
	38400_E81.txt.....	108
	57600_O81.txt.....	108
	01.DIO .....	109
	DIO_Addr01_1.txt .....	109
	DIO_Addr01_2.txt .....	110
	DIO_Addr01_3.txt .....	111
	DIO_Addr01_4.txt .....	112
	02.DA.....	113

DA_Addr02_1.txt.....	113
DA_Addr02_2.txt.....	114
DA_Addr02_3.txt.....	115
DA_Addr02_4.txt.....	116
DA_Addr02_5.txt.....	117
03.AD.....	118
AD_Addr03_1.txt.....	118
AD_Addr03_2.txt.....	119
04.DIO_DA_AD.....	120
DIO_DA_AD_1.txt.....	120
05.Rising_Trigger.....	121
RisingTrigger_1.txt.....	121
RisingTrigger_2.txt.....	122
RisingTrigger_3.txt.....	123
06.Initial_Value.....	124
Init_Value_1.txt.....	124
07.Swap_Byte_Word.....	126
Both_Swap_1.txt.....	126
Byte_Swap_1.txt.....	127
Word_Swap_1.txt.....	128
08.State_Change_Trigger.....	129
State_Change_1.txt.....	129
State_Change_2.txt.....	130
09.Constant_Output.....	131
Constant_1.txt.....	131
10.Bit_Command.....	132
Bit_Cmd_1.txt.....	132
11.Delay_Command.....	133
Delay_Cmd_1.txt.....	133
12.TxPdo_RxPdo_0x80_0xFF.....	134
TxPdo_RxPdo_0x80.txt.....	134
TxPdo_RxPdo_0xFF.txt.....	134
TxPdo_RxPdo_AD_0x80.txt.....	135
TxPdo_RxPdo_AD_0xFF.txt.....	135
TxPdo_RxPdo_DA_0x80_0xFF.txt.....	136
13.Commands_128_202.....	136
14.End_of_Cmd_Dealy.....	137
End_Delay_1.txt.....	137
15.TxPdo_RxPdo_Max.....	138
TxRxPdo_Max_1.txt.....	138
TxRxPdo_Max_2.txt.....	139
TxRxPdo_Max_3.txt.....	139
16.Rs485_Cycle_Time.....	140

<b>Rs485_Cycle_Time_1.txt</b> .....	140
<b>Rs485_Cycle_Time_2.txt</b> .....	141
17. <i>Ext_Sync</i> .....	142
<b>ext_sync.txt</b> .....	143
A3. ECAT-2610-DW 配置檔參考文件 .....	145
01. <b>32_bit_Read_Power_Meter</b> .....	146
02. <b>16_bit_Read_System</b> .....	147
03. <b>8_bit_DO0_DO1</b> .....	147
<b>DO0_DO1.txt</b> .....	147
<b>DO0_DO1_2.txt</b> .....	148
04. <b>16_bit_Set_Parameter</b> .....	148
05. <b>8_16_32_Full</b> .....	149
06. <b>to_meterX3</b> .....	150
07. <b>to_meterX6</b> .....	151
08. <b>TEST</b> .....	152
<b>TEST_4A.txt</b> .....	152
<b>TEST_5A_8.txt</b> .....	152
<b>TEST_5B_16.txt</b> .....	153
<b>TEST_5C_32.txt</b> .....	153
<b>TEST_5D_16.txt</b> .....	153
<b>TEST_5E_2.txt</b> .....	154
A4. 手動配置及上傳 .....	155
A4-1 設定配置檔 ( <i>Commands.txt</i> ) .....	155
修改 ECAT-2610 的配置檔 .....	155
修改 ECAT-2610-DW 的配置檔 .....	159
修改 ECAT-2611 的配置檔 .....	163
A4-2 上傳配置檔 .....	165
A5. 支援泓格 MODBUS RTU 從站產品 .....	170
A6. 手冊修訂記錄 .....	171

## 配件清單

產品包裝內應包含下列配件：



ECAT-2610(-DW)/2611 x 1



快速入門指南 x 1



CA-0915 Cable x 1



注意

如發現產品包裝內的配件有任何損壞或遺失，請保留完整包裝盒及配件，盡快聯繫我們，我們將有專人快速為您服務。

## 產品資訊



➤ 使用手冊/快速入門指南/產品型錄：

[http://ftp.icpdas.com/pub/cd/fieldbus\\_cd/ethercat/slave/ecat-2000/manual/](http://ftp.icpdas.com/pub/cd/fieldbus_cd/ethercat/slave/ecat-2000/manual/)

➤ XML 設備明說 (ESI)：

[http://ftp.icpdas.com/pub/cd/fieldbus\\_cd/ethercat/slave/ecat-2000/software/](http://ftp.icpdas.com/pub/cd/fieldbus_cd/ethercat/slave/ecat-2000/software/)

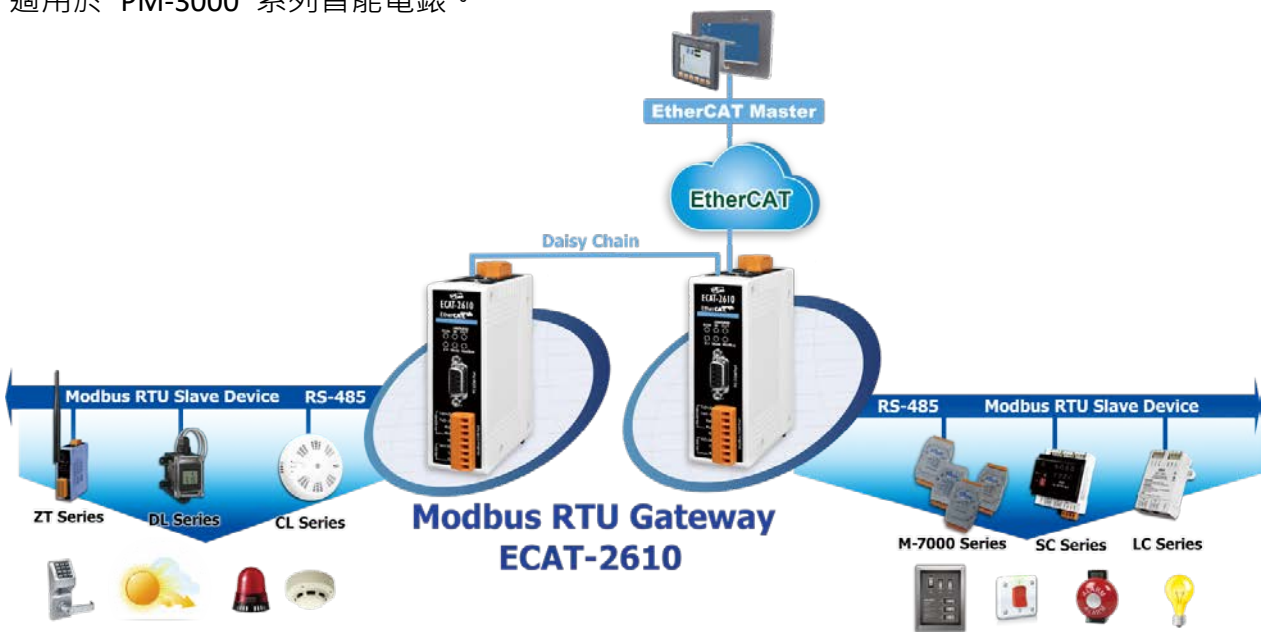
➤ 產品問答：

<http://www.icpdas.com/root/support/faq/faq.html>

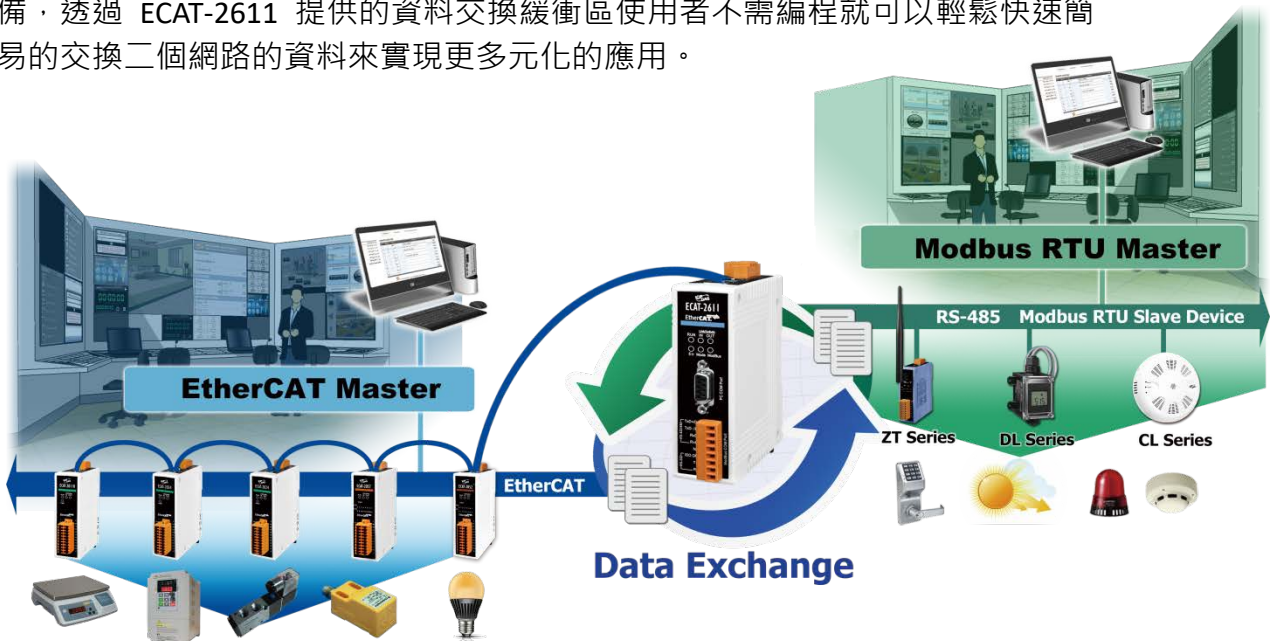
# 1. 簡介

ECAT-2610(-DW)/2611 擁有 EtherCAT 轉 Modbus RTU 的閘道器功能，能夠將 RS-232/422/485 串列設備及機器整合至 EtherCAT 控制系統中，且非常容易安裝及配置。

ECAT-2610 是經過驗證的可靠協定轉換閘道器，能將無法上網的工業設備連接至 EtherCAT 網路。ECAT-2610 閘道器執行智能協定轉換並將工業設備的串列數據轉為容易處理的 I/O 數據，傳送給主站 PLC、控制器，實現多元化的應用。ECAT-2610-DW 是 ECAT-2610 的 DWORD 版本，適用於 PM-3000 系列智能電錶。



ECAT-2611能讓您的系統可以在 EtherCAT 和 Modbus RTU 網路之間無縫地互連各自的從站設備，透過 ECAT-2611 提供的資料交換緩衝區使用者不需編程就可以輕鬆快速簡易的交換二個網路的資料來實現更多元化的應用。



## 1.1 特色

- 高性能 MCU 有效處理資料
- 2 埠 RJ-45 EtherCAT 接頭，支援菊鏈式拓撲 (Daisy Chain)
- 允許系統整合商將舊有的自動化設備改造成現代 EtherCAT 通訊結構
- 無需對已連接的設備進行硬體或軟體更改
- 能與所有 PLC 控制器相容
- 完整的串列協定轉換，無需 PLC 功能方塊

### **ECAT-2610**

- 整合 RS-232/422/485 串列設備及機器至 EtherCAT 網路
- 支援最大 256 WORD 輸入資料及 256 WORD 輸出資料
- 支援串列埠介面
  - 支援 Modbus RTU (Master) 協定
  - 支援 RS-232/422/485 埠
  - 支援最大 Baud Rate 115200 bps

### **ECAT-2610(-DW)**

- 支援最大 128 WORD 輸入資料及 128 WORD 輸出資料
- 最多可連接 6 顆 PM-3033/3133 系列智能電錶
- 提供 PM-3033/3133/3114/3112 專用設定檔範例及 ESI 檔
- 可混合使用其他的 Modbus RTU Slave
- 支援串列埠介面
  - 支援 Modbus RTU (Master) 協定
  - 支援 RS-232/422/485 埠
  - 支援最大 Baud Rate 115200 bps

### **ECAT-2611**

- 可在二個網路之間傳輸 I/O 資料
- 支援最大 256 WORD 輸入資料及 256 WORD 輸出資料
- 支援支援串列埠介面
  - 支援 Modbus RTU (Slave) 協定
  - 支援 RS-232/422/485 埠
  - 支援最大 Baud Rate 115200 bps
  - 支援 Modbus 功能碼 03、04、06、16



## 1.2 架構圖

ECAT-2610(-DW)/2611 模組架構圖如下:

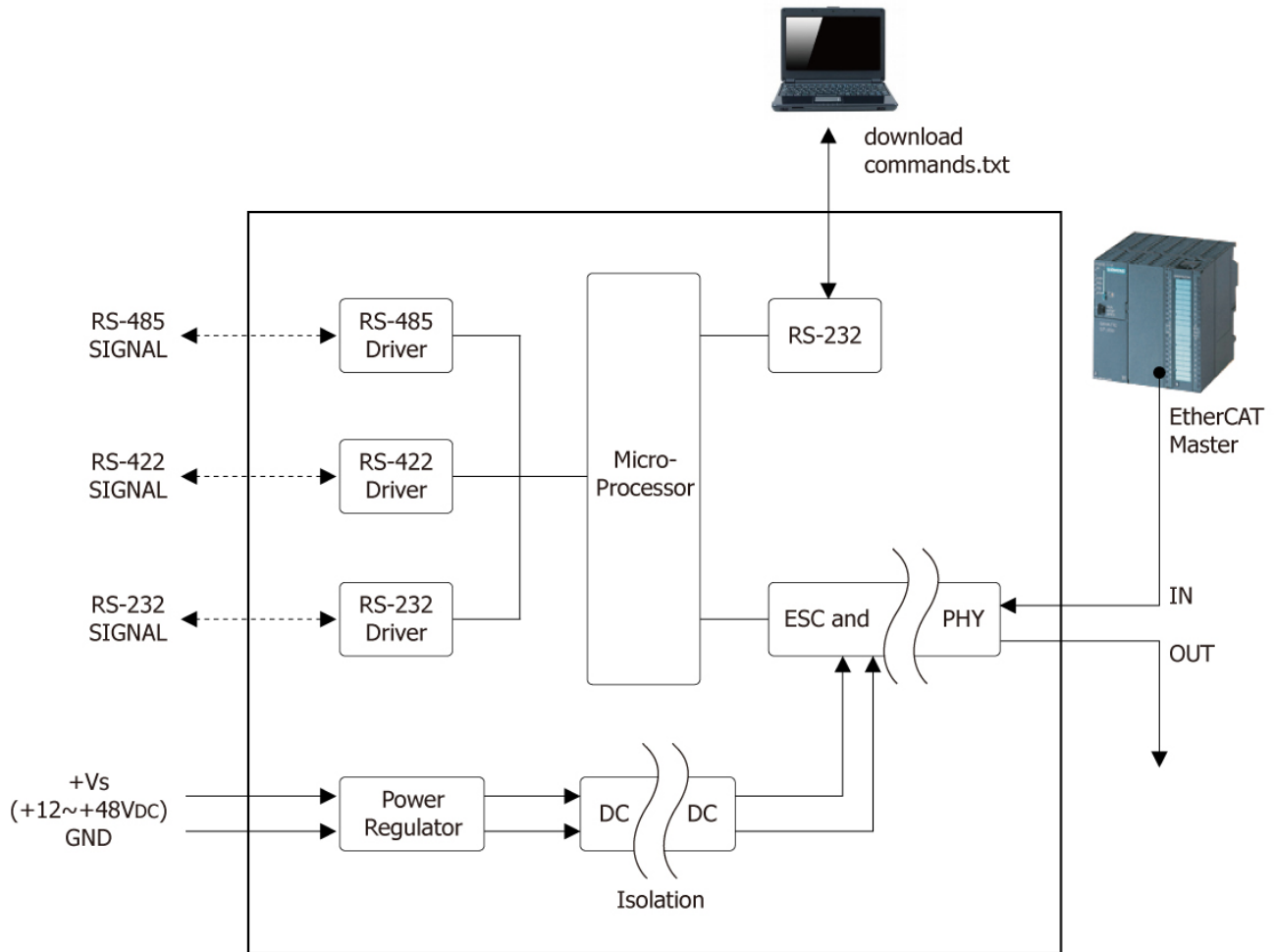
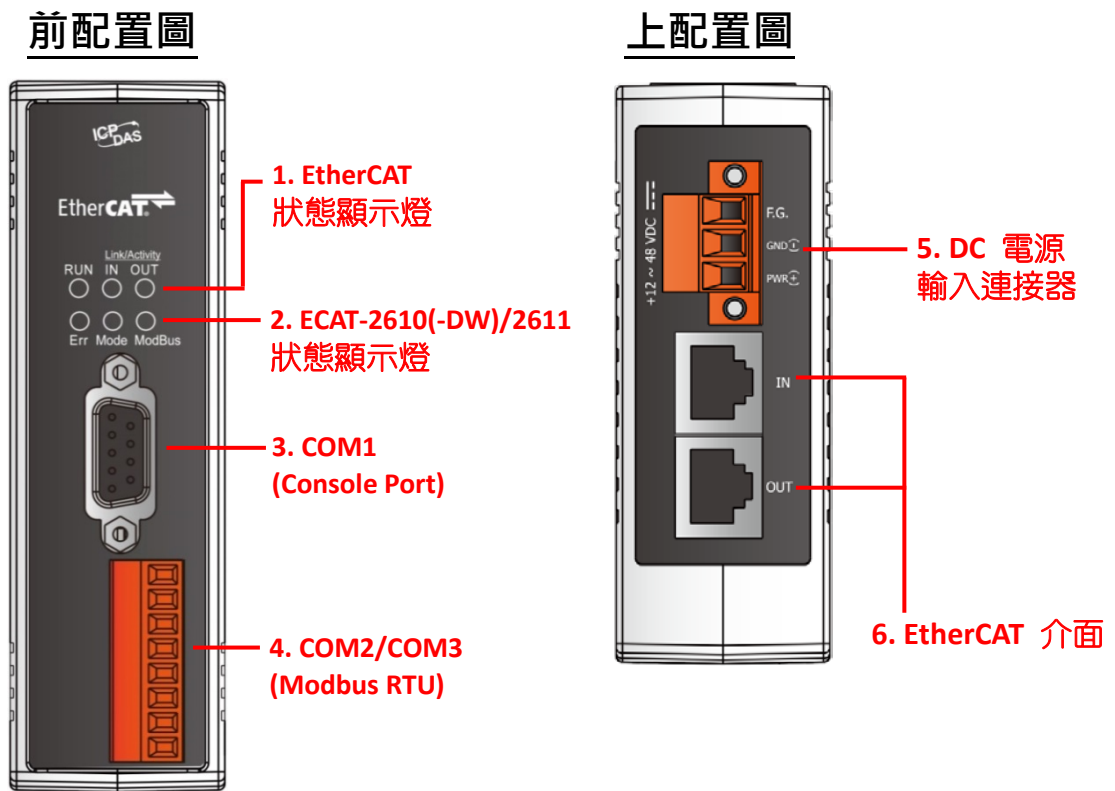


圖 1-2: ECAT-2610(-DW)/2611 架構圖

# 2. 硬體資訊

## 2.1 外觀

下面為 ECAT-2610(-DW)/2611 模組前面及上面的外觀配置圖，詳細說明如下。



### 1. EtherCAT 狀態顯示燈

項目	顏色	狀態	說明
RUN	紅色	OFF	設備處於 INIT 狀態
		閃爍	燈號顯示將以 2.5 Hz 頻率來分辨開啟或關閉·閃爍 200 ms，接著停止 200 ms
		閃一次	燈號將亮起 200 ms，接著停止 1000 ms
		ON	設備處於 OPERARIONAL 狀態
Link Activity IN/OUT	綠色	OFF	沒有偵測到網路連線
		閃爍	網路封包收送中
		ON	偵測到網路連線

## 2. ECAT-2610(-DW)/2611 狀態顯示燈

項目	顏色	狀態	說明
Err	紅色	關	正常
		閃爍	發生錯誤
Mode	綠色	閃爍 0.3 秒	啟用 DC
		閃爍 1 秒	正常
		閃爍 2 秒	沒有配置數據或配置數據錯誤
		閃爍 4 秒	配置 CRC 錯誤
Modbus	綠色	關	無 Modbus 命令
		閃爍 1 秒	正常

## 3. COM1 (Console Port · DB9-Male)

配置/診斷埠。詳細的 Console Port 腳位定義，參考至 [章節 2.3 “腳位定義”](#)。

## 4. COM2/COM3 (Modbus RTU)

可連接 Modbus RTU 設備。詳細的 Modbus RTU 腳位定義，參考至 [章節 2.3 “腳位定義”](#)。

## 5. DC 電源輸入連接器

電源接線端子連接器上的“PWR(+)”及“GND(-)”適用於 ECAT-2610(-DW)/2611 模組，用於直流供電方式開機。有效的電源輸入範圍: +12 V<sub>DC</sub> ~ +48 V<sub>DC</sub>。

### “F.G.” (Frame Ground):

在大陸性氣候區裡，電子電路不斷受到靜電 (ESD) 影響，ECAT-2610(-DW)/2611 模組設計有 Frame Ground (F.G.)，提供靜電依接地路徑釋放 (ESD)，因此能夠增強靜電 (ESD) 保護，確保模組更穩定可靠。

## 6. EtherCAT 介面




ECAT-2610(-DW)/2611 模組包含二個 RJ-45 插座為 EtherCAT 標準介面 **IN** 埠為 EtherCAT 信號輸入端用來與 EtherCAT Master 或上一個 EtherCAT Slave 的 EtherCAT 介面 OUT 埠作連結，**OUT** 埠為 EtherCAT 信號輸出端用來與下一個 EtherCAT Slave 的 EtherCAT 介面 IN 埠作連結。

## 2.2 規格

項目		ECAT-2610(-DW)	ECAT-2611
<b>Protocol</b>			
Protocol		EtherCAT	
RJ-45 Port		RJ-45 x 1 Max. distance between stations: 100 m (100BASE-TX) Data Transfer Medium: Ethernet/EtherCAT Cable (Min.CAT 5e)	
<b>Communication</b>			
Protocol		Modbus RTU (Master)	Modbus RTU (Slave)
Serial Interface	RS-232	Note that the RS-232, RS-422 and RS-485 interfaces cannot be used simultaneously <ul style="list-style-type: none"> <li>● TxD, RxD, GND</li> <li>● TxD+, TxD-, RxD+, RxD-</li> <li>● Data+, Data-</li> </ul>	
	RS-422		
	RS-485		
<b>Power Input</b>			
Redundant Input Range		+12 ~ +48 V <sub>DC</sub>	
Power Consumption		0.1 A @ 24 V <sub>DC</sub>	
Protection		Power reverse polarity protection	
Connector		3-pin Removable Terminal Block (5.08 mm)	
<b>Mechanical</b>			
Dimensions (H x W x D)		110 mm x 90 mm x 33 mm	
Installation		DIN-Rail Mounting	
<b>Environment</b>			
Operating Temperature		-25 to +75°C	
Storage Temperature		-30 to +80°C	
Relative Humidity		10 to 90% RH, Non-condensing	


## 2.3 腳位定義

### EtherCAT 介面

Pin Assignment	
F.G.	
GND(-)	
PWR(+)	
IN	
OUT	


### COM1 (Console Port)

Pin Assignment	Terminal No.	Terminal No.	Pin Assignment
-	01	06	-
RxD	02	07	-
TxD	03	08	-
-	04	09	-
GND	05		



### COM2/COM3 (Modbus RTU)

Terminal No.	Pin Assignment
COM2	TxD+/D+
	TxD-/D-
	RxD+
	RxD-
	N/A
COM3	ISO.GND
	TxD
	RxD

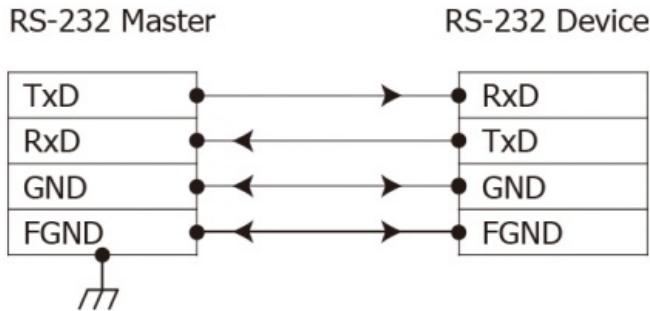


**注意**

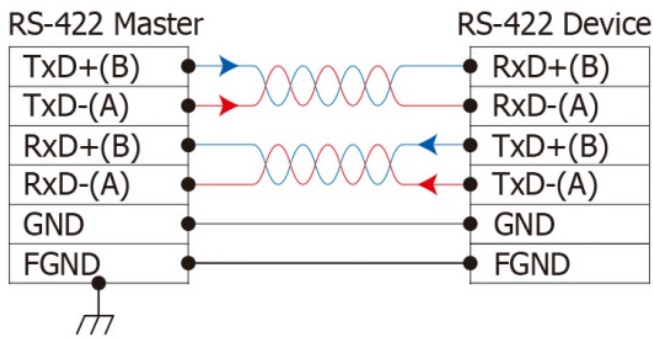
RS-232、RS-422 及 RS-485 不能同時一起使用

## 2.4 RS-232/422/485 接線注意

### 3 線 RS-232 接線



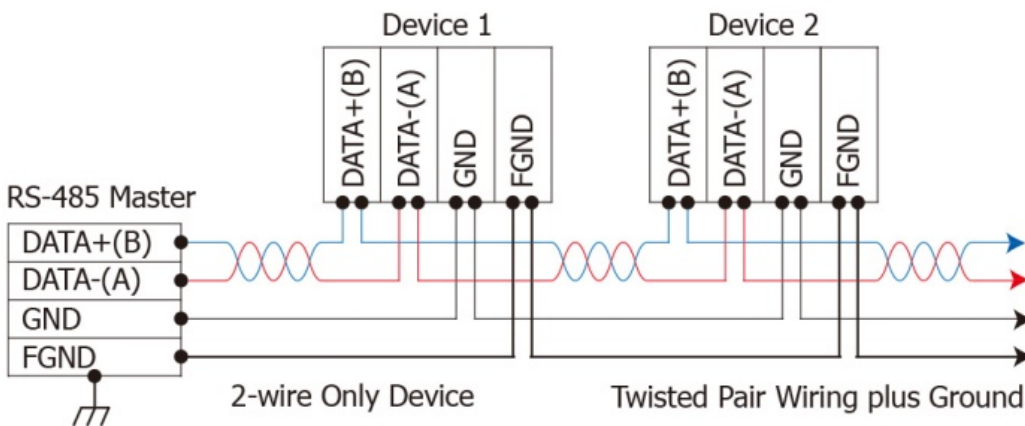
### 4 線 RS-422 接線



#### ⚠注意

1. 一般情況下，RS-422/485 Port 需將 RS-422/485 設備的所有 GND 接地。這將減少設備之間的共模電壓。
2. DATA+/- 接線必須使用雙絞線 Cable。
3. 在接線的兩端可能需要加上終端電阻 (通常使用 120 Ω)，跨接在兩線之間 (DATA+ 及 DATA-)。
4. 在 RS-422/485 接線圖中，DATA+ (B) 為正極腳位，DATA- (A) 為負極腳位。關於 B/A 腳位定義取決於您所使用的設備，請先確認。

### 2 線 RS-485 接線



## 2.5 Init/Normal 運作模式

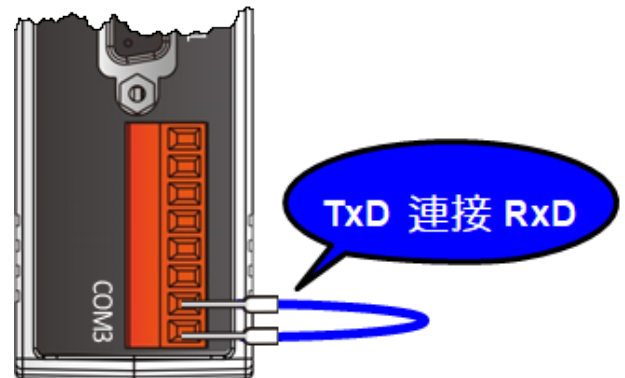
ECAT-2610(-DW)/2611 模組提供有二種運作模式可以選擇，詳細說明如下：

### ➤ Init 模式

當 ECAT-2610(-DW)/2611 模組發生問題時，可切換模組至 Init 模式來進行故障排除，步驟如下：

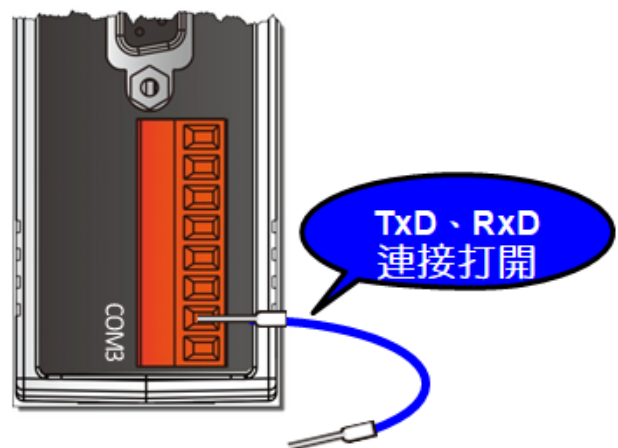
1. ECAT-2610(-DW)/2611 模組斷電關機，將模組連接至電腦主機。
2. 將模組 COM3 上的 TxD pin 連接至 RxD pin 來切換至 “Init” 模式。
3. 執行 7188ECAT.exe，然後將模組上電開機，確認模組進入 “Init” 模式。
4. 再將模組 COM3 上的 TxD pin 和 RxD pin 斷開來切換至 “Normal” 模式。
5. 清楚模組上的 EEPROM，再載入新的配置數據檔到 EEPROM。
6. 然後斷電再上電重新啟動模組，使用模組在 “Normal” 模式下運作。

上面步驟 1 到 6 詳細操作說明，參考[第 7 章 “上傳配置檔操作”](#)。



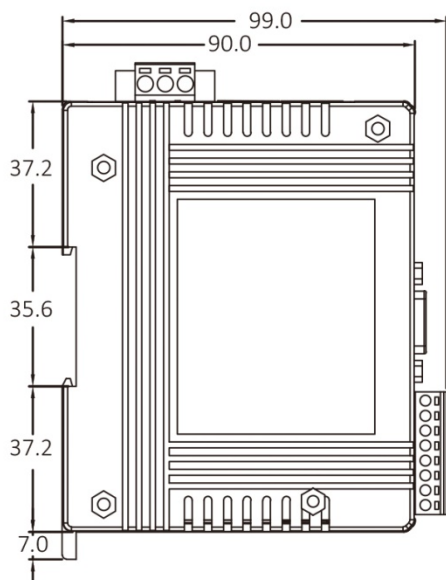
### ➤ Normal 模式

Normal 模式為出廠預設模式，是 ECAT-2610(-DW)/2611 模組大部份運作時所使用的模式。

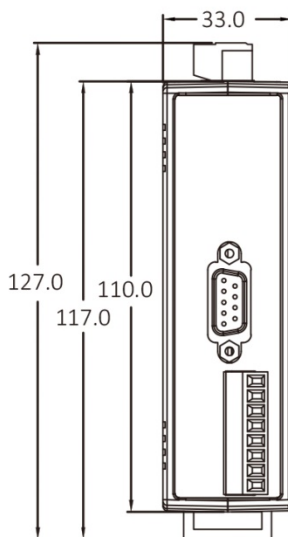


## 2.6 機構圖

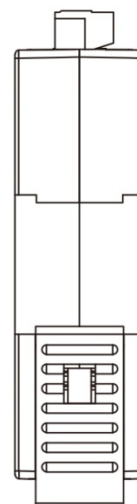
下圖為 ECAT-2610(-DW)/2611 模組的機構圖，單位為 mm (millimeters)。



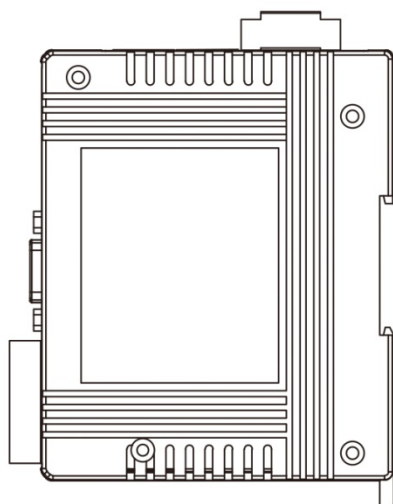
左側視圖



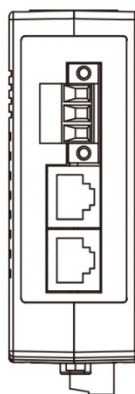
前視圖



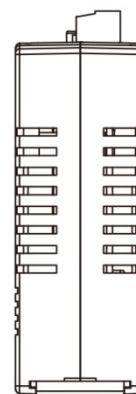
後視圖



右側視圖



上視圖



下視圖



## 3. 啟用 ECAT-261x 模組

此章節將介紹 ECAT-2610(-DW)/2611 模組如何啟動、線接、配置數據及上傳...等。

### 3.1 出廠預設值

下表為 ECAT-2610(-DW)/2611 模組原廠預設值：

項目	預設值	說明
InTxPDO	10	詳細關於 InTxPDO 及 OutRxPDO 說明，參考 <a href="#">第 9 章 “Object 說明及參數設定”</a>
OutRxPDO	10	
Run LED	紅燈	詳細關於 LED 指示燈說明，參考 <a href="#">章節 2.1 “外觀”</a>
IN LED	閃綠燈	
Mode LED	閃綠燈	
NetID (ECAT-2611)	1	詳細關於如何設定 ECAT-2610 的 Baud Rate、Data Format、指令...等說明，參考 <a href="#">章節 4.2 “配置及載入”</a>
Baud Rate	115200	
Parity	N (None)	
Stop Bit	1	詳細關於如何設定 ECAT-2611 的 NetID、Baud Rate、Data Format...等說明，參考 <a href="#">章節 5.1 “配置及載入”</a>
InTxPDO[00] = 2610SYS0	Sys_low = 0x0000	詳細關於 Sys_low 及 Sys_hi 說明，參考 <a href="#">章節 3.3.1 “模組狀態及錯誤模式”</a>
InTxPDO[01] = 2610SYS1	Sys_hi = 0xA000	

## 3.2 連接主站與電源

**步驟 1** 確認您電腦的網路設定正確且可運作，將網路線一端連至 **ECAT-2610(-DW)/2611 的 EtherCAT 介面 IN 端** 另一邊接至電腦的 **RJ-45 乙太網路埠** 上

確認您電腦的 Windows 防火牆以及 Anti-virus 防火牆都已關閉，或已正確的設定。

### 注意

建議使用獨立網卡，請勿連接至外部網路上以免造成網路風暴

- 1 從 PC 端乙太網路埠連結網路線至 **IN 埠**。
- 2 提供電源 (使用  $+12\text{ V}_{\text{DC}} \sim +48\text{ V}_{\text{DC}}$  電源) 到 ECAT-2610(-DW)/2611，請將模組上的 **PWR(+)** 接至 電源供電器正端 ( $+12 \sim +48\text{ V}_{\text{DC}}$ )，模組上 **GND(-)** 接至 電源供應器負端。

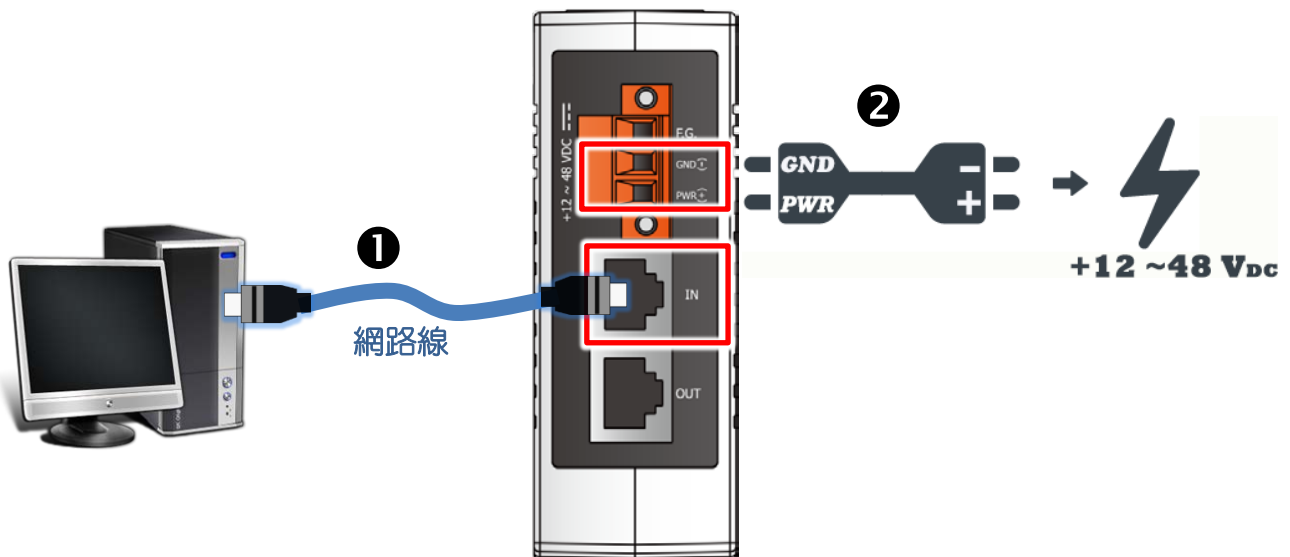


圖 3-2.1

## 步驟 2 確認 ECAT-2610(-DW)/2611 模組上的 LED 顯示燈是否正確顯示

- ❶ ECAT-2610(-DW)/2611 供電完成後，**“IN”** 及 **“Mode”** 顯示燈將顯示閃爍 (綠燈)。
- ❷ 當 ECAT-2610(-DW)/2611 連上 EtherCAT Master 時，**“RUN”** 顯示燈才會亮起 (紅燈)。

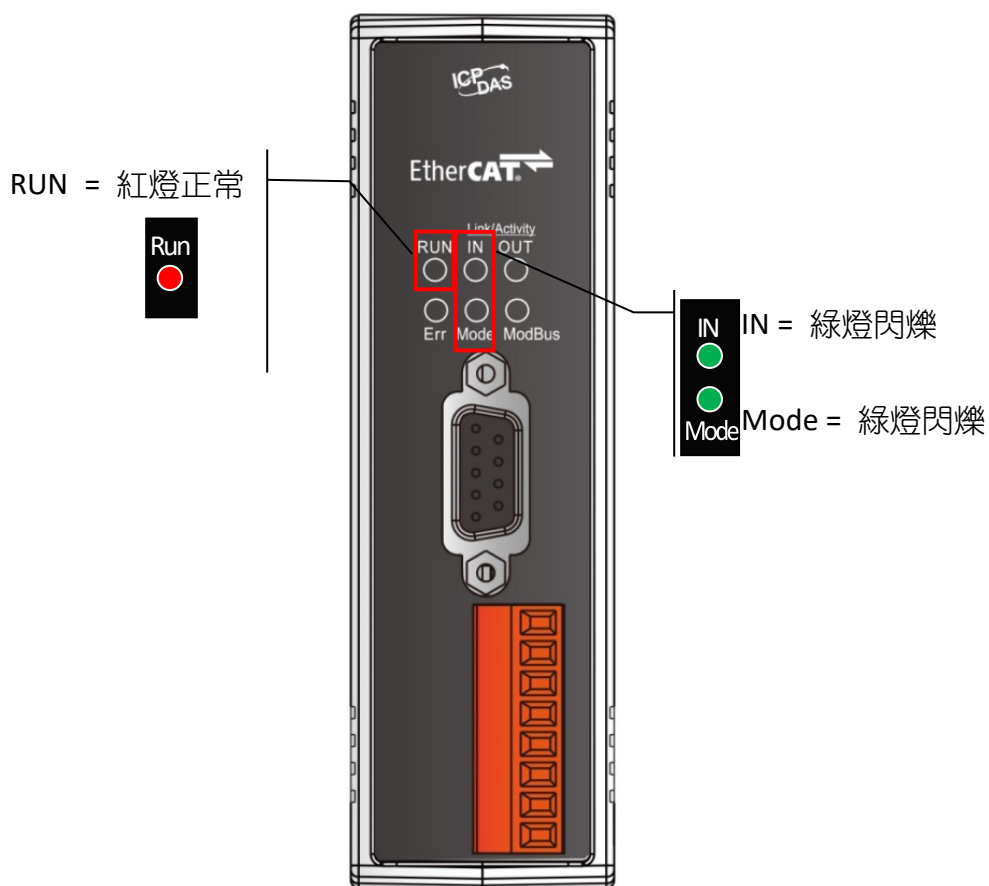


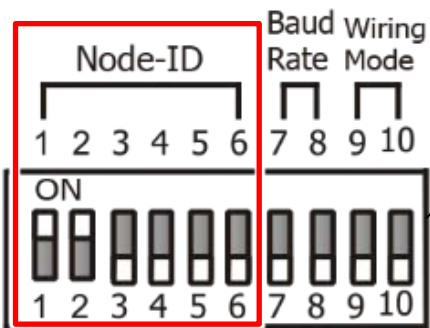
圖 3-2.2

### 3.3 連接 Power Meter (適用 ECAT-2610-DW)

此章節僅適用於 ECAT-2610-DW，其他型號請跳過此部份。請注意: ECAT-2610-DW 僅支援 PM-3033, PM-3133, PM-3114 及 PM-3112 系列智能電錶。這裡，我們以 PM-3113 為範例。

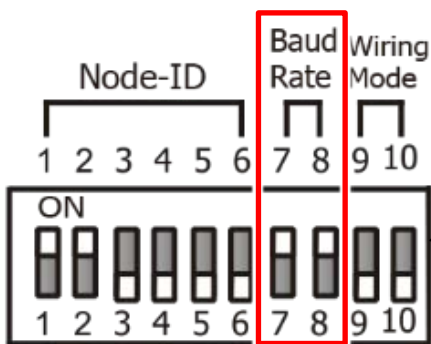
#### 步驟 1 連接 PM-3133 智能電錶到 ECAT-2610-DW

❶ 設定 PM-3133 的 Modbus RTU Address、Baud Rate 及 Wiring Mode 來符合 ECAT-2610-DW 的配置數據檔，如下：



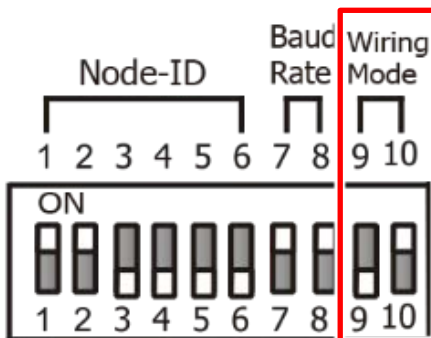
設定 DIP Switch 1 ~ 6 為 ON, ON, OFF, OFF, OFF, OFF，表示 Modbus Address 設定為 4

圖 3-3.1



設定 DIP Switch 7 ~ 8 為 ON, ON，表示 Baud Rate 設定為 115200

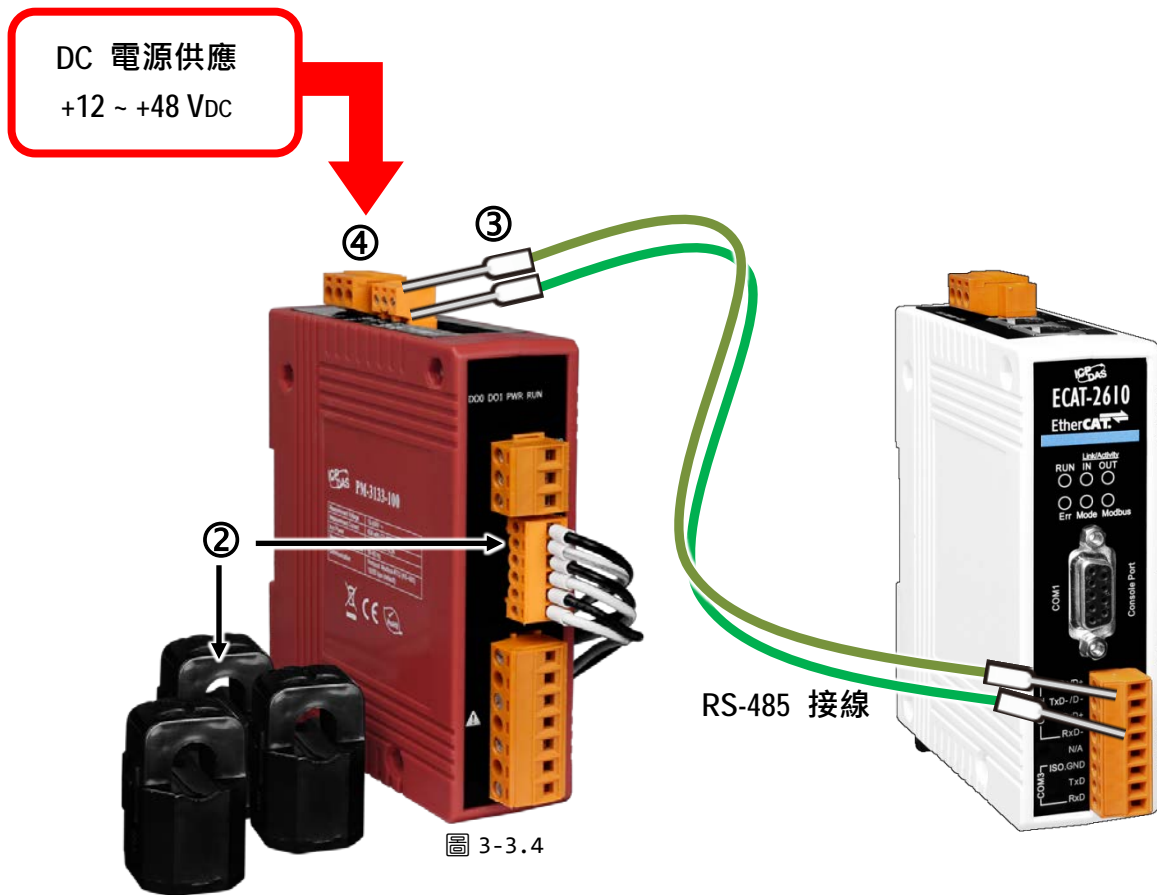
圖 3-3.2



設定 DIP Switch 9 ~ 10 為 OFF, ON，表示 Wiring Mode 設定為 3P3W-3CT

圖 3-3.3

- ② 檢查電流輸入接線端子後連接 CT，然後選擇 CT Chip。
- ③ 將 PM-3133 上的 RS-485 bus 連接到 ECAT-2610-DW 上的 COM2 (RS-485 bus)。
- ④ 提供電源到 PM-3133 (使用電源 +12 ~ +48 Vdc)。



**注意**

1. 更多更詳細關於 PM-3133 系列智能電錶的硬體配置、CT 安裝、電源供應及接線注意，請參考至 PM-3133 的快速入門指南 (<http://ftp.icpdas.com/pub/cd/powermeter/pm-3133/quickstartguide/>) 或使用手冊 (<http://ftp.icpdas.com/pub/cd/powermeter/pm-3133/user'smanual/>)。
2. 如是其他型號智能電錶 (如: PM-3030、PM-3114、PM-3112)，請參考至各自智能電錶的快速入門指南或使用手冊。

## 3.4 搜尋模組

在執行下面步驟之前，必須先安裝 EtherCAT Master 軟體 (如: Beckhoff TwinCAT)。在本例中，我們使用市面上最常使用的 **Beckhoff TwinCAT 2.X** 來配置及連結控制 ECAT-2610(-DW)/2611 模組。

### 進入 EtherCAT 網路



注意

#### 安裝最新版本的 XML 裝置描述檔案 (ESI)

請確保安裝最新版本的 XML 裝置描述檔案至 TwinCAT 上，最新版本的 XML 裝置描述檔案可以至泓格科技公司網站下載

[http://ftp.icpdas.com/pub/cd/fieldbus\\_cd/ethercat/slave/ecat-2000/software/](http://ftp.icpdas.com/pub/cd/fieldbus_cd/ethercat/slave/ecat-2000/software/)，並按照安裝說明依序進行安裝。

#### 步驟 1 安裝 ESI 檔案

- 複製 “ICPDAS ECAT-2610.xml”、 “ICPDAS ECAT-2610DW.xml” 或 “ICPDAS ECAT-2611.xml” 檔案至 EtherCAT Master 工具的資料夾，如下位置：

軟體名稱	預設路徑
Beckhoff EtherCAT Configuration	<a href="C:\EtherCAT Configurator\EtherCAT">C:\EtherCAT Configurator\EtherCAT</a>
Beckhoff TwinCAT 3.X	<a href="C:\TwinCAT\3.x\Config\Io\EtherCAT">C:\TwinCAT\3.x\Config\Io\EtherCAT</a>
Beckhoff TwinCAT 2.X	<a href="C:\TwinCAT\Io\EtherCAT">C:\TwinCAT\Io\EtherCAT</a>

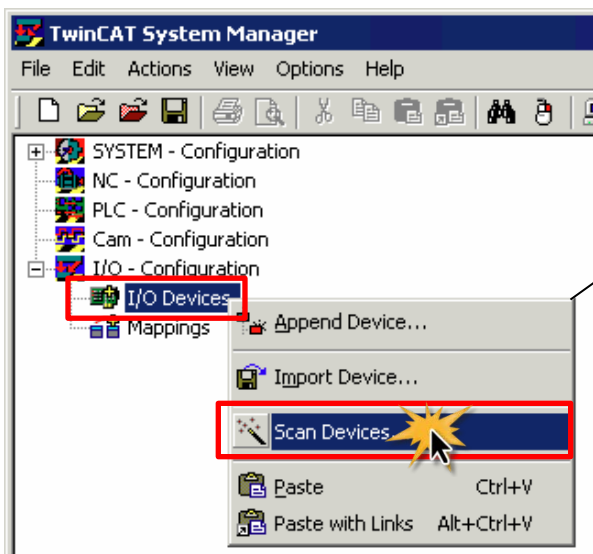
#### 步驟 2 自動掃描

- 連結 ECAT-2610(-DW)/2611 模組至 EtherCAT，連接時系統必需在安全、斷電的狀態。
- 開啟模組電源並打開 TwinCAT System Manager(Config mode)，開始掃描裝置後，之後所有的對話框皆為 “OK”，使模式設定值在 **FreeRun** 模式。



注意

在執行 TwinCAT 掃描前，請先將智能電錶連接至 ECAT-2610-DW 上，詳細參考第 3.3 節 “[連接 Power Meter \(適用 ECAT-2610-DW\)](#)”。



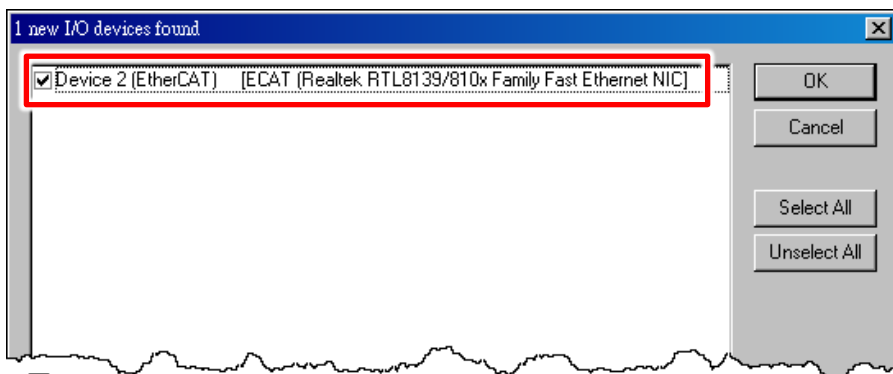
掃描設備  
(I/O Devices-> 按右鍵 -> Scan Devices...)

圖 3-4.1



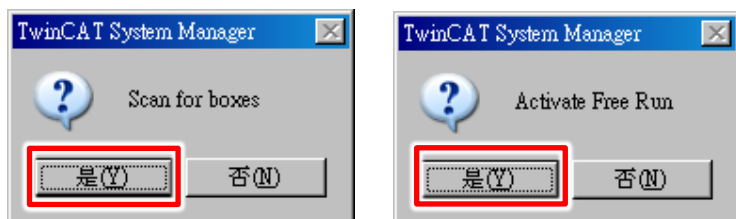
按下“確定”鍵

圖 3-4.2



選擇連接的網路卡並按下“OK”鍵

圖 3-4.3



按下“是”鍵開始搜尋  
按下“是”鍵來激活 TwinCAT System Manager 的 Free Run 模式

圖 3-4.4

在 TwinCAT System Manager 左側視窗將顯示 Box1 (模組名稱) · 如下所示。

➤ ECAT-2610/2611

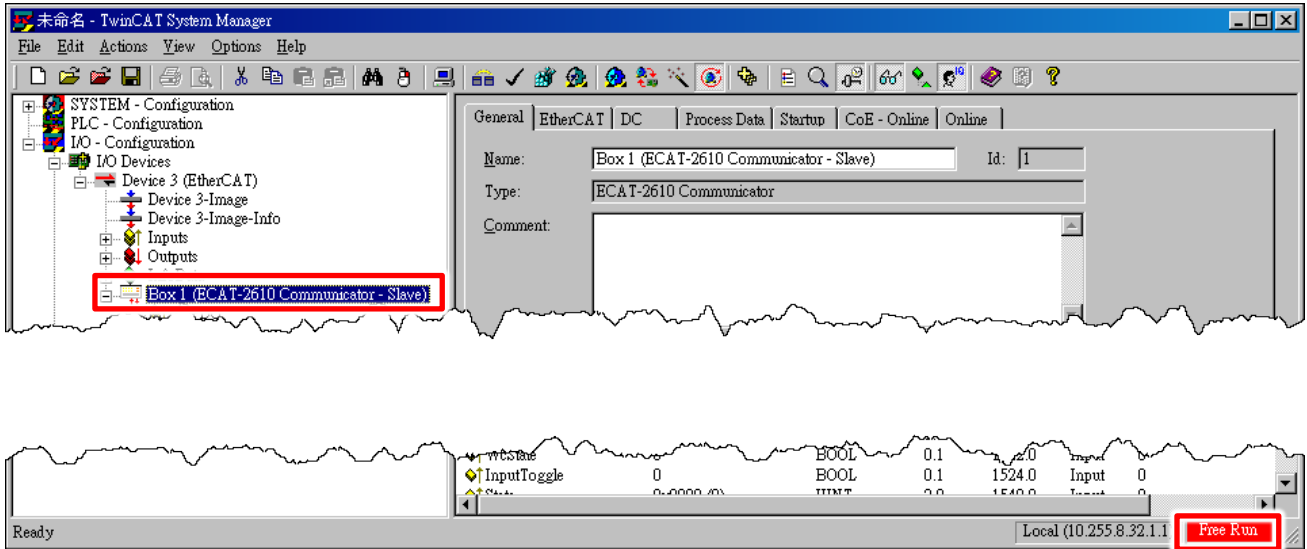


圖 3-4.5

➤ PM-3000 系列智能電錶

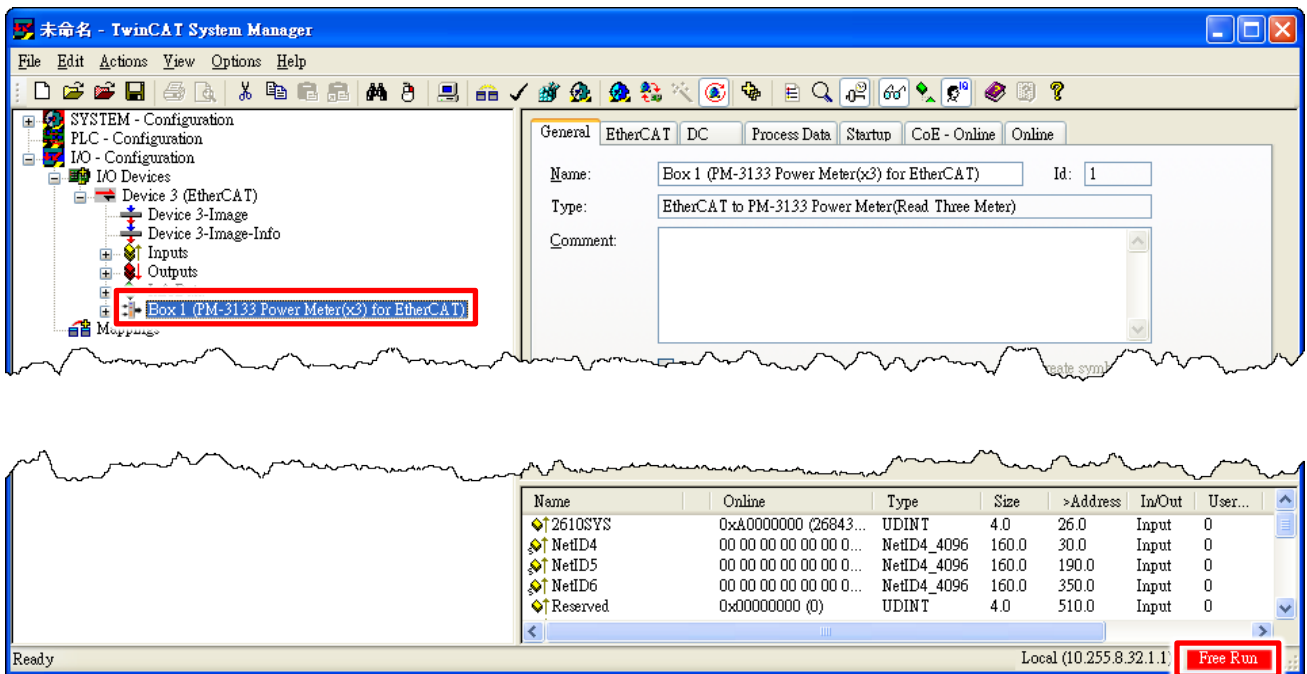


圖 3-4.6



### 3.5 驗證模組狀態

❶ 在 TwinCAT System Manager 左側欄位視窗中，單擊您想要設定的 EtherCAT BOX 後，開啟樹狀分支，如下所示。

- **ECAT-2610/2611:** 單擊左側欄位視窗中 “TxPDO 0x00-0x7F” 來查看模組狀態。

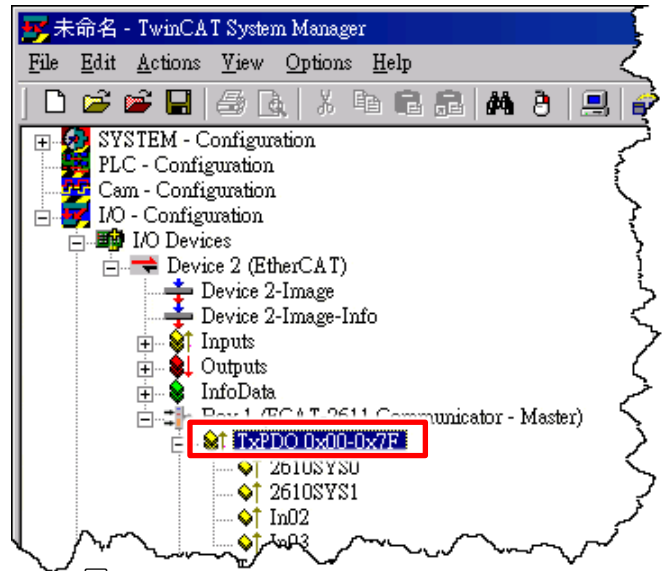


圖 3-5.1

- **PM-3000 系列智能電錶:** 單擊左側欄位視窗中 “PM Inputs Channel 1” 來查看模組狀態。

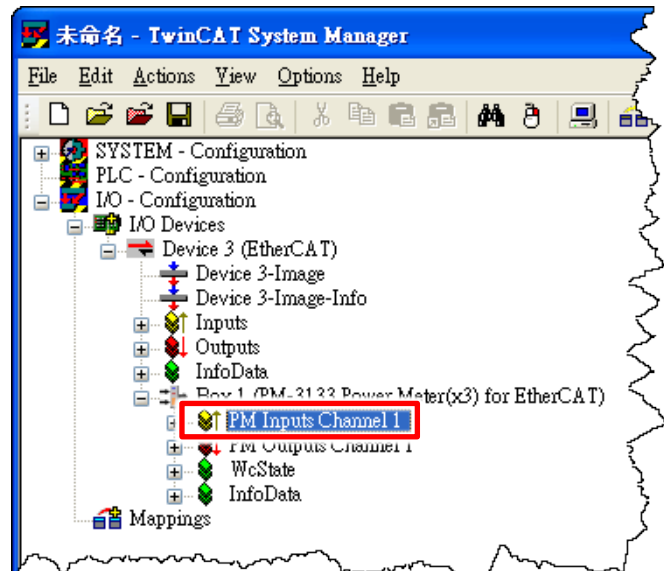


圖 3-5.2

② 在右側視窗，檢查 **Sys\_hi** 項目顯示值，如下圖所示。

➤ **ECAT-2610/2611: 2610/2611SYS1 = 0xA000 = 表示模組正常運作中。**

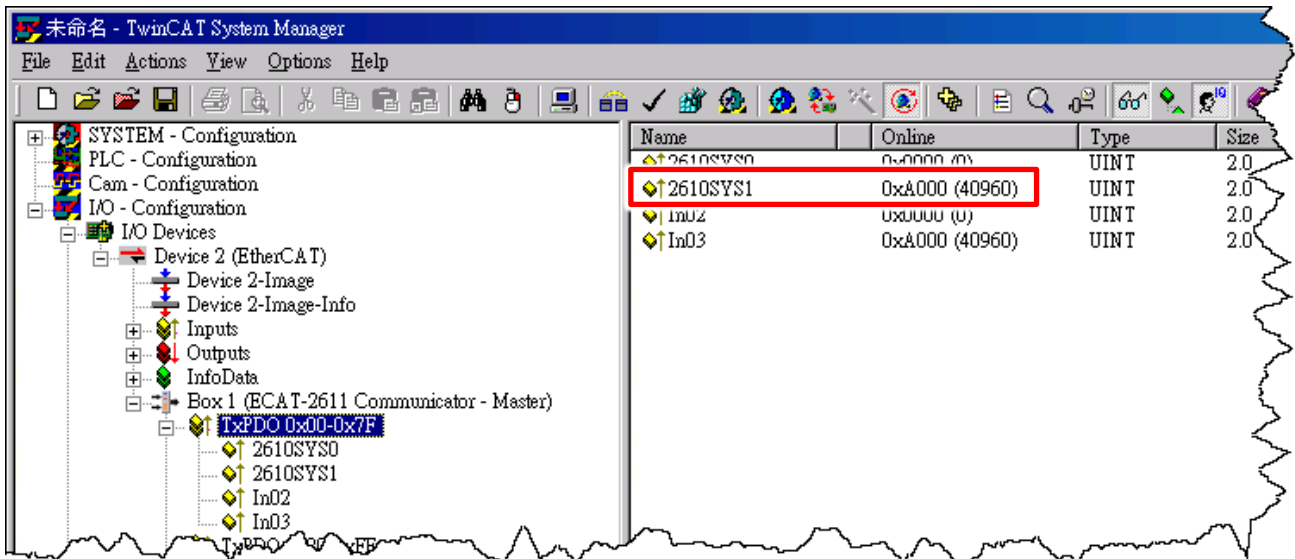


圖 3-5.3

➤ **PM-3000 系列智能電錶: 2610SYS = 0xA0000000 = 表示模組正常運作中。**

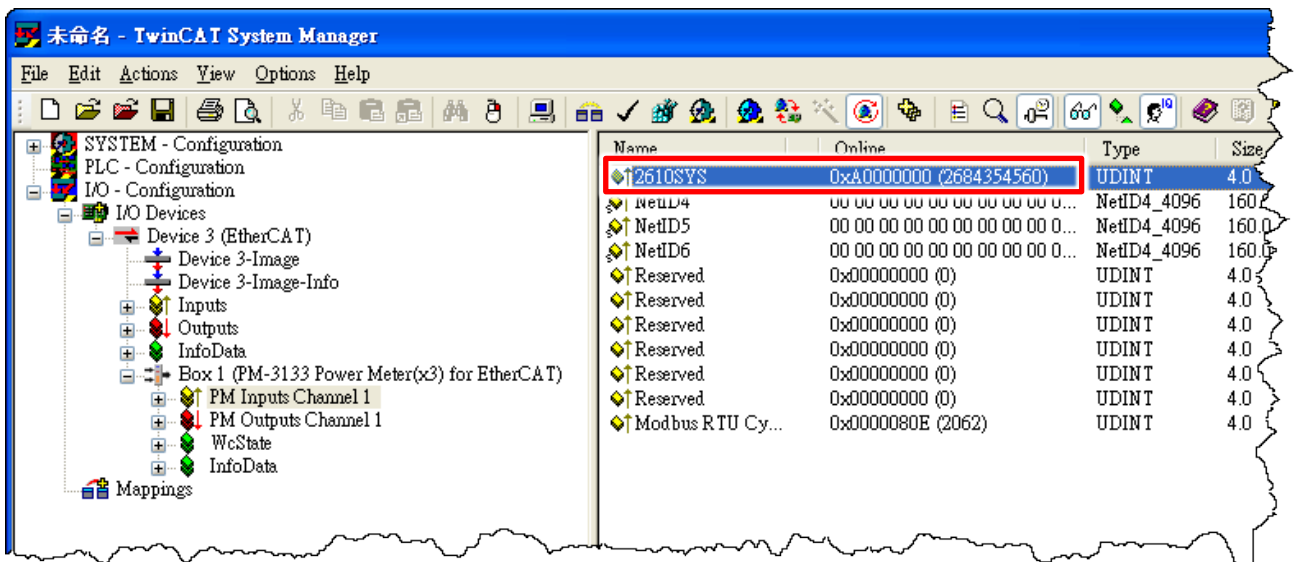


圖 3-5.4

**注意**

詳細關於 ECAT-2610(-DW)/2611 模組狀態說明，參考[章節 3.5.1 “模組狀態及錯誤模式”](#)。

### 3.5.1 模組狀態及錯誤模式

當 ECAT-2610(-DW)/2611 模組供電開機後，將從 EEPROM 讀取並檢查配置檔，如發現任何配置錯誤，ECAT-2610(-DW)/2611 模組將會進入錯誤模式，詳細說明如下。

**注意**

1. 詳細關於配置數據檔 (commands.txt) 格式及說明，請參考章節 [A4 “手動配置數據檔及上傳”](#)。
2. EEPROM 用於存儲不經常更改的數據，不適合頻繁的存取大量數據，且清除/寫入週期是有限制的，因此在測試使用時不應經常更改，否則容易造成模組損壞。

➤ 配置數據有錯誤時，Err LED 顯示燈將亮起，如下圖所示：

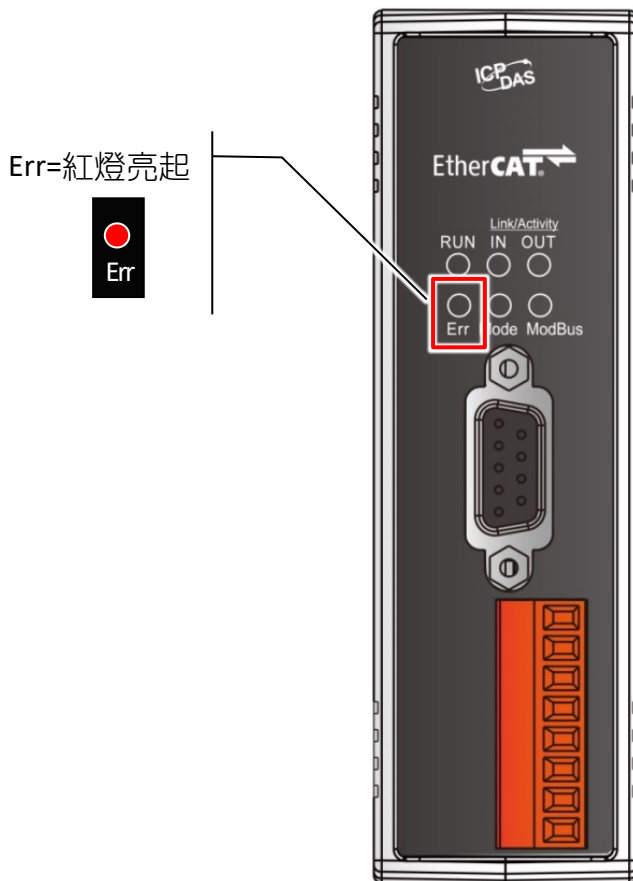


圖 3-5.5

## ECAT-2610/2611

➤ 您可查看 **InTxPDO[00]** 及 **InTxPDO [01]** 來確定錯誤來源。詳細項目如下表。

**表 3-5.1 (Read): InTxPDO[00] = 2610/2611SYS0 = Sys\_low** 定義說明，如下：

Bit	ECAT-2610	ECAT-2611
12-15	N/A	
11	InMax/OutMax Error	
10	CmdFun Error: 命令功能錯誤，詳細設定說明參考 <a href="#">第 6 章 “Modbus 資訊”</a>	
09	CmdLen Error: 命令長度錯誤，詳細設定說明參考 <a href="#">第 6 章 “Modbus 資訊”</a>	
08	CmdNum Error: 命令數量錯誤，設定範圍: 0 ~ 300 (最大)	
07	Read CRC Error	
06	Address Error	
05	Delay Value Error: 設定範圍: 0 ~ 255 ms	
04	Timeout Value Error : 設定範圍: 0 ~ 255 ms	
03	Stop Bit Error: 設定範圍: 1, 2	
02	Parity Bit Error: 設定範圍: N (None), E (EVEN), O (ODD)	
01	Baud Rate Error: 設定範圍: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200	
00	Init_pin 短接: 進入偵錯模式，詳細說明參考 <a href="#">第 7 章 “載入配置檔操作”</a>	

**表 3-5.2 (Read): InTxPDO[01] = 2610/2611SYS1 = Sys\_hi** 定義說明，如下：

Bit	ECAT-2610	ECAT-2611
15	Exec Baud Rate 3: 設定範圍: 參考下 <a href="#">表 3-5.3</a>	1
14	Exec Baud Rate 2: 設定範圍: 參考下 <a href="#">表 3-5.3</a>	N/A
13	Exec Baud Rate 1: 設定範圍: 參考下 <a href="#">表 3-5.3</a>	N/A
12	Exec Baud Rate 0: 設定範圍: 參考下 <a href="#">表 3-5.3</a>	N/A
11	Exec Even Parity: 設定範圍: 0 (不是 EVEN Parity) · 1 (是 EVEN Parity)	N/A
10	Exec Odd Parity: 設定範圍: 0 (不是 ODD Parity) · 1 (是 ODD Parity)	N/A
09	Exec Stop Bit: 設定範圍: 0 (1 Stop Bit) · 1 (2 Stop Bit)	N/A
08	Exec Default = 115200 + N81	N/A
07	N/A	N/A
06	N/A	N/A
05	Exec Ext_Sync	N/A
04	Exec CRC Error	
03	Exec return FC (Function Code) Error	
02	Exec return Net_ID Error	
01	Exec with init value	
Bit 00	Exec Modbus Timeout	

表 3-5.3: Baud Rate 設定如下:

Bit3	Bit2	Bit1	Bit0	Baud Rate
0	0	0	0	保留
0	0	0	1	保留
0	0	1	0	保留
0	0	1	1	1200
0	1	0	0	2400
0	1	0	1	4800
0	1	1	0	9600
0	1	1	1	19200
1	0	0	0	38400
1	0	0	1	57600
1	0	1	0	115200
1	0	1	1	230400
1	1	0	0	460800
1	1	0	1	921600
1	1	1	0	保留
1	1	1	1	保留

- **OutRxPDO[00]** 及 **OutRxPDO [01]** 提供模組系統控制 (如: 清除 sys\_low 及 sys\_hi...等)。詳細系統控制項目如下表。

表 3-5.4 (Write): **OutRxPDO[00] = 2610/2611CTL0** 及 **OutRxPDO[01] = 2610/2611CTL1** 定義如下:

Bit	ECAT-2610	ECAT-2611	ECAT-2610	ECAT-2611
Bit	OutRxPDO[00]		OutRxPDO[01]	
06-15	N/A		N/A	
05	當 High 時開始 Ext_Sync 運作 當 Low 時停止 Ext_Sync 運作	N/A	N/A	
04	當 High 時啟用 Ext_Sync 機制	N/A	N/A	
03	Command TimeOut No Re-send		N/A	
02	No CRC Check		N/A	
01	Clear Sys_low, Sys_hi		N/A	
00	Initial Ready		N/A	

## ECAT-2610-DW

➤ 您可查看 **InTxPDO[00]** 來確定錯誤來源。詳細項目如下表。

**表 3-5.5 (Read): InTxPDO[00] = 2610SYS** 定義說明，如下：

Bit	ECAT-2610-DW
31	Exec Baud Rate 3: 設定範圍: 參考下 <a href="#">表 3-5.6</a>
30	Exec Baud Rate 2: 設定範圍: 參考下 <a href="#">表 3-5.6</a>
29	Exec Baud Rate 1: 設定範圍: 參考下 <a href="#">表 3-5.6</a>
28	Exec Baud Rate 0: 設定範圍: 參考下 <a href="#">表 3-5.6</a>
27	Exec Even Parity: 設定範圍: 0 (不是 EVEN Parity) , 1 (是 EVEN Parity)
26	Exec Odd Parity: 設定範圍: 0 (不是 ODD Parity) , 1 (是 ODD Parity)
25	Exec Stop Bit: 設定範圍: 0 (1 Stop Bit) , 1 (2 Stop Bit)
24	Exec Default = 115200 + N81
22-23	N/A
21	Exec Ext_Sync
20	Exec CRC Error
19	Exec return FC (Function Code) Error
18	Exec return Net_ID Error
17	Exec with init value
16	Exec Modbus Timeout
12-15	N/A
11	InMax/OutMax Error
10	CmdFun Error: 命令功能錯誤，詳細設定說明參考 <a href="#">第 6 章 “Modbus 資訊”</a>
09	CmdLen Error: 命令長度錯誤，詳細設定說明參考 <a href="#">第 6 章 “Modbus 資訊”</a>
08	CmdNum Error: 命令數量錯誤，設定範圍: 0 ~ 300 (最大)
07	Read CRC Error
06	Address Error
05	Delay Value Error: 設定範圍: 0 ~ 255 ms
04	Timeout Value Error : 設定範圍: 0 ~ 255 ms
03	Stop Bit Error: 設定範圍: 1, 2
02	Parity Bit Error: 設定範圍: N (None), E (EVEN), O (ODD)
01	Baud Rate Error: 設定範圍: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
00	Init_pin 短接: 進入偵錯模式，詳細說明參考 <a href="#">第 7 章 “載入配置檔操作”</a>

表 3-5.6: Baud Rate 設定如下:

Bit3	Bit2	Bit1	Bit0	Baud Rate
0	0	0	0	保留
0	0	0	1	保留
0	0	1	0	保留
0	0	1	1	1200
0	1	0	0	2400
0	1	0	1	4800
0	1	1	0	9600
0	1	1	1	19200
1	0	0	0	38400
1	0	0	1	57600
1	0	1	0	115200
1	0	1	1	230400
1	1	0	0	460800
1	1	0	1	921600
1	1	1	0	保留
1	1	1	1	保留

- **OutRxPDO[00]** 提供模組系統控制 (如: 清除 sys\_low 及 sys\_hi...等)。詳細系統控制項目如下表。

表 3-5.7 (Write): OutRxPDO[00] = 2610CTL 定義如下:

Bit	ECAT-2610-DW
06-31	N/A
05	當 High 時開始 Ext_Sync 運作 當 Low 時停止 Ext_Sync 運作
04	當 High 時啟用 Ext_Sync 機制
03	Command TimeOut No Re-send
02	No CRC Check
01	Clear Sys_low, Sys_hi
00	Initial Ready

## 4. 啟動 Modbus RTU 設備



### 注意

在“啟動 Modbus RTU 設備”前，請先確保 ECAT-2610 模組正常運行，詳細參考 [第 3 章 “啟用 ECAT-261x 模組”](#)。

下面範例，我們將使用 M-7050D 模組來進行測試，而其它我司設備或是第三方 Modbus RTU 設備，請參考各自設備的快速入門指南或使用手冊來執行。

此章節將介紹如何配置、連接 Modbus RTU Slave 設備，包括設定 Slave ID、Baud Rate、Data Format...等，以及如何將配置檔載入到 ECAT-2610 模組來控制您的 Modbus RTU Slave 設備。

下圖為連接 Modbus RTU Slave 設備的配置快速程序參考，請依下列順序來正確啟用您的 Modbus RTU 設備：

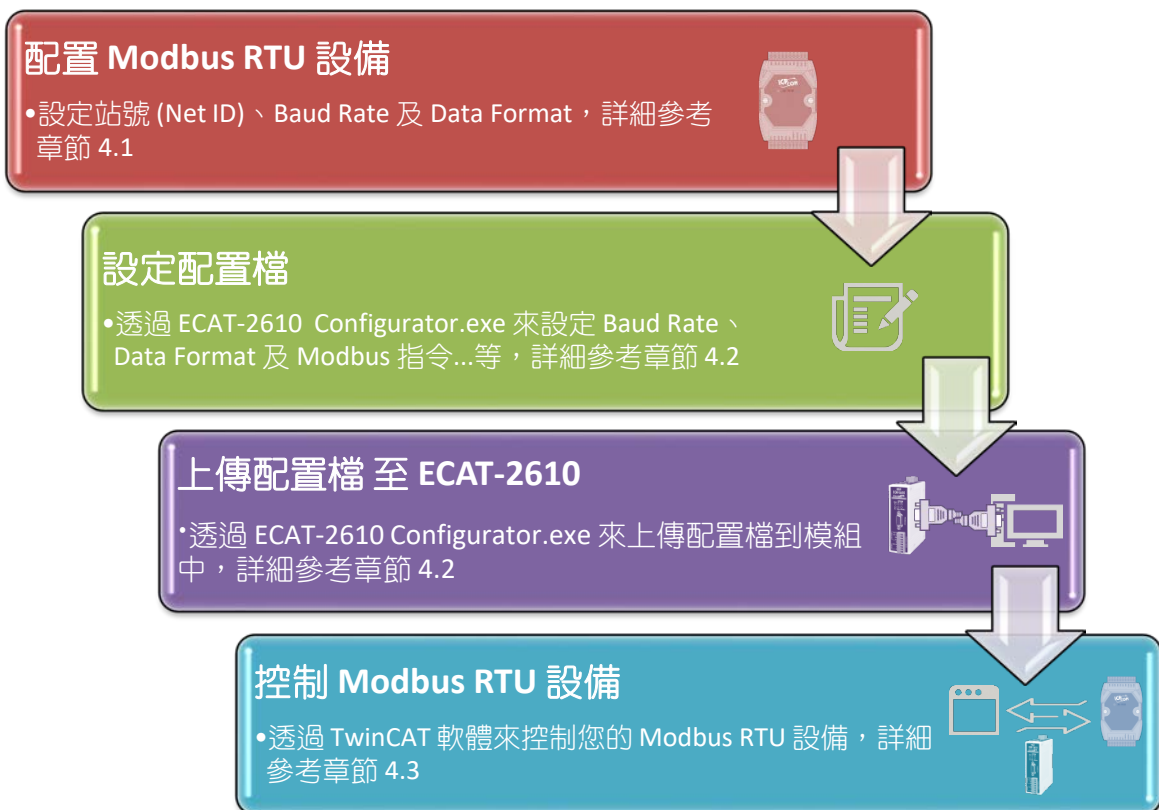


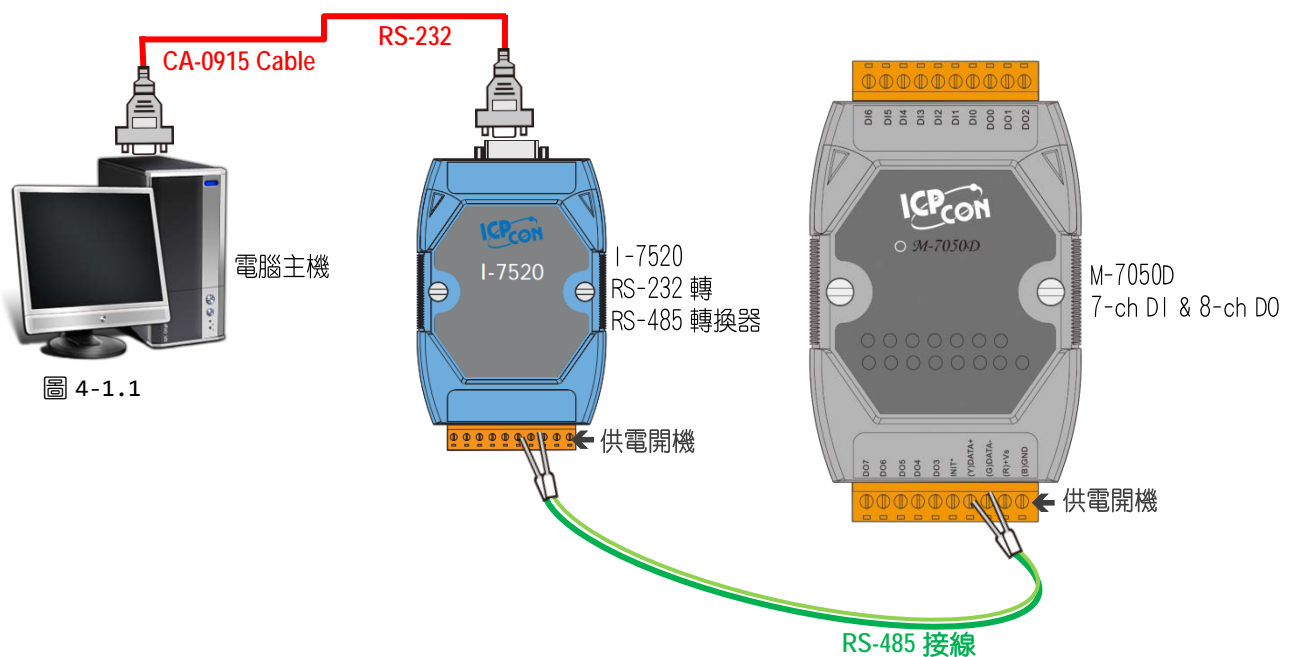
圖 4.1: 啟用 Modbus RTU Slave 設備



## 4.1 配置 Modbus RTU 設備

下面配置 Modbus RTU 設備方法只適用於我司的 Modbus RTU 設備，其它第三方 Modbus RTU 設備請參考各自設備的使用手冊來執行配置站號 (Net ID)、Baud Rate 及 Data Format...等。

### 步驟 1 將 Modbus Slave 設備 (如: M-7050D，選購品) 連接至電腦主機



### 步驟 2 下載並執行 DCON Utility Pro 軟體

如使用我司的 Modbus RTU 設備，可從泓格科技的軟體網站中免費下載 DCON Utility Pro 來執行配置 Modbus 設備，詳細下載位置如下：

[http://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon\\_utility/](http://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon_utility/)



### 步驟 3 搜尋模組

❶ 按下 “COM Port” 按鈕來開啟 “Comport Option” 配置對話框。選擇 COM Port (如: COM1) , 此 COM Port 為您電腦所連接至 M-7050D 的 COM Port , 然後按下 “OK” 按鈕。

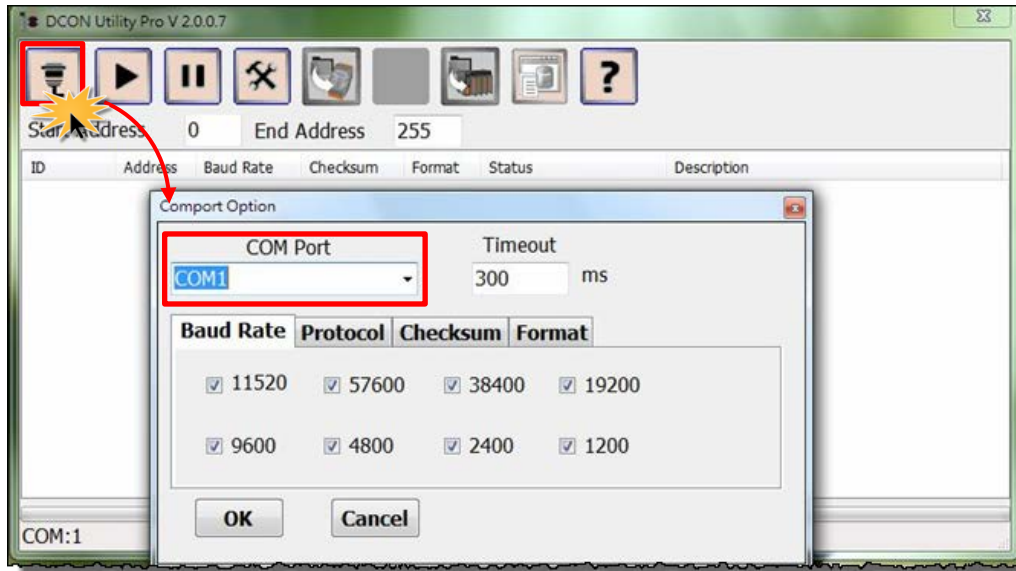


圖 4-1.2

❷ 按下 “Start Search” 按鈕來尋搜 Modbus RTU Slave 設備。



圖 4-1.3

❸ 當搜尋到 Modbus RTU Slave 設備時將被顯示在列表中，再按下 “Stop Search” 按鈕。

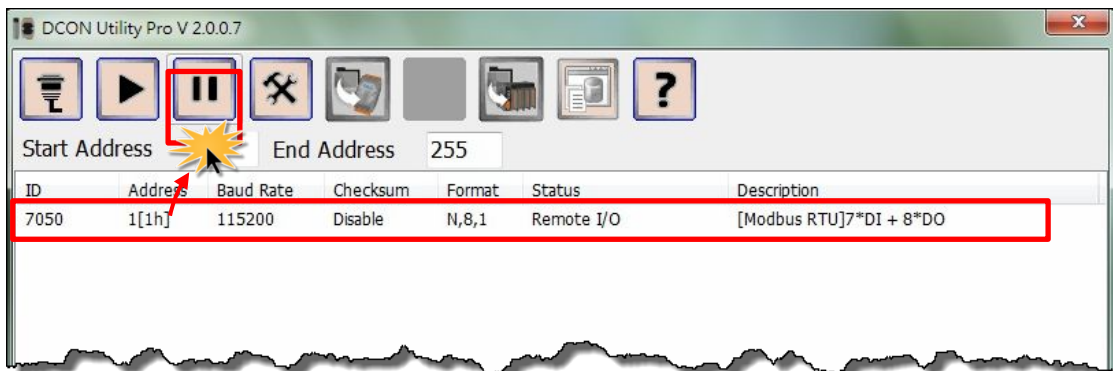


圖 4-1.4

## 步驟 4 配置 Modbus RTU 設備的 NetID、Baud Rate 及 Format

- 1 按下列表中的**模組名稱(ID 欄位)**來開啟配置對話框。
- 2 設定 Modbus RTU 設備的 **Address (Slave ID)**、**Baud Rate** 及 **Data Format**。
- 3 按下 **“Set Module Configurations”** 按鈕來儲存新的設定值。

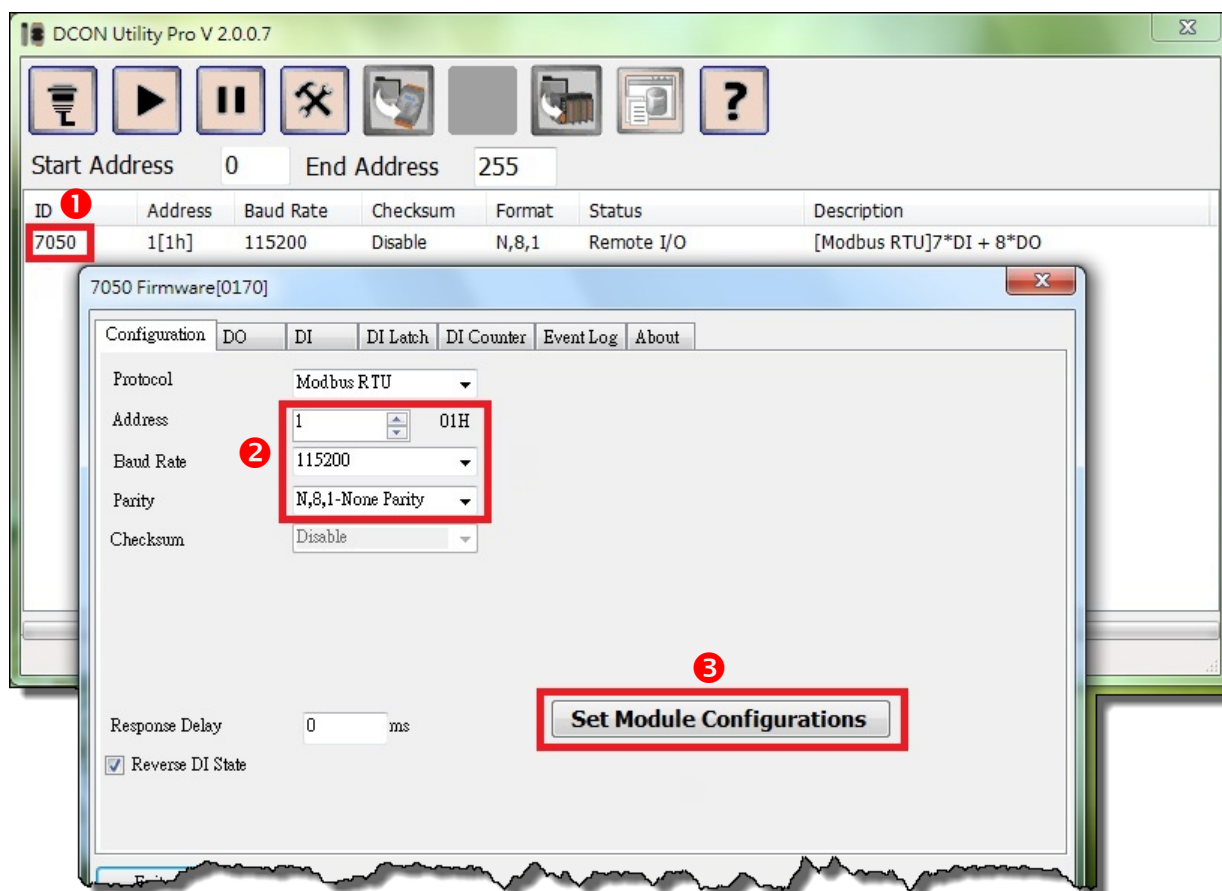


圖 4-1.5

### ⚠ 注意

當在 RS-485 網路中有多個 Modbus RTU 設備時，則每個設備都需分配指定一個唯一的站號 (Net ID)

## 4.2 配置及上傳

### **步驟 1** 將下載 Cable 連接至電腦主機及 ECAT-2610 模組

- ❶ 將 ECAT-2610 模組斷電關機。
- ❷ 將配件 CA-0915 Cable 連接至電腦主機上的 COM Port 及 ECAT-2610 模組上的 COM1 Port。

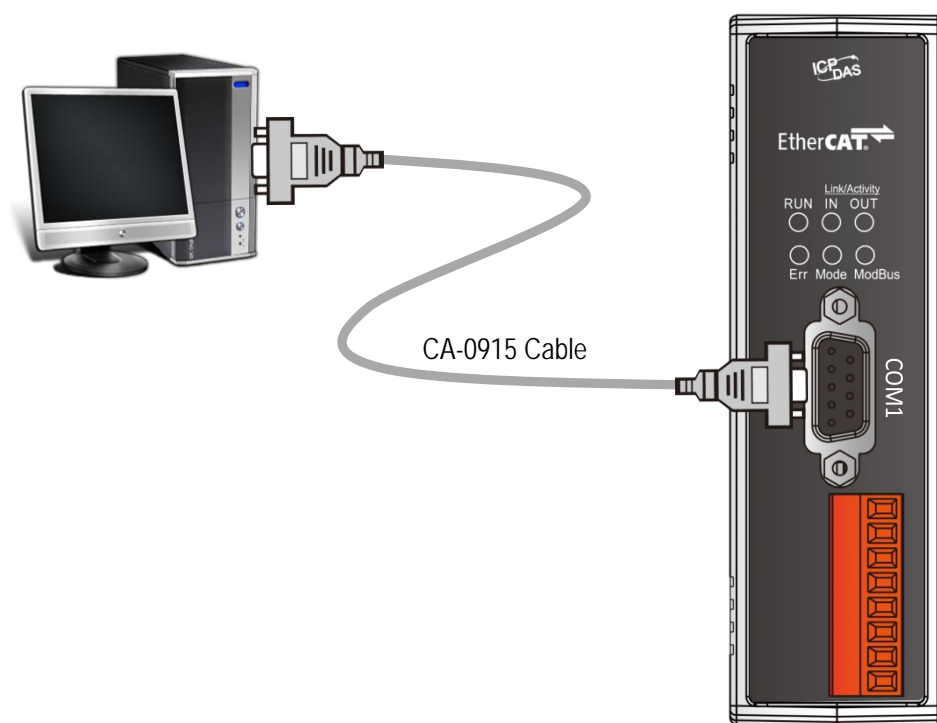


圖 4-2.1








## 步驟 2 下載 ECAT-2610\_Utl\_xxxxxxI 壓縮檔

❶ “ECAT-2610\_Utl\_xxxxxx.zip” 可以從泓格科技網站下載。下載位置如下：

 [http://ftp.icpdas.com/pub/cd/fieldbus\\_cd/ethercat/slave/ecat-2000/software/](http://ftp.icpdas.com/pub/cd/fieldbus_cd/ethercat/slave/ecat-2000/software/)

❷ 將 ECAT-2610\_Utl\_xxxxxx.zip 解壓縮後，包含 7188ECAT 資料夾是配置/偵錯工具。

❸ 將 7188ECAT 資料夾複製到電腦硬碟上 (如: E:\)，資料夾中應包含下列檔案：

 more commands_2610	配置 DI、DO、AD 及 DA 等範例參考文件 詳細說明參考 <a href="#">A2. “ECAT-2610 配置數據檔參考文件”</a>
 7188ECAT.exe	執行檔，詳細參考 <a href="#">A4-2 “上傳配置檔”</a>
 7188XW.CF4	7188ECAT.exe 控制檔
 commands.txt	配置檔，用來設定 Modbus RTU 設備的文件檔。ECAT-2610 將透過此配置檔來與 Modbus RTU 設備進行通訊
 execCOM1.bat	使用電腦主機的 COM1 來載入配置數據檔至 ECAT-2610，詳細參考 <a href="#">A4-2 “上傳配置檔”</a>
 execCOM2.bat	使用電腦主機的 COM2 來載入配置數據檔至 ECAT-2610，詳細參考 <a href="#">A4-2 “上傳配置檔”</a>
 ECAT-2610 Configurator.exe	ECAT-2610 Configurator.exe 配置工具包

## 步驟 3 執行 ECAT-2610\_Configurator.exe

雙擊 “ECAT-2610 Configurator.exe” 來開啟配置工具包。

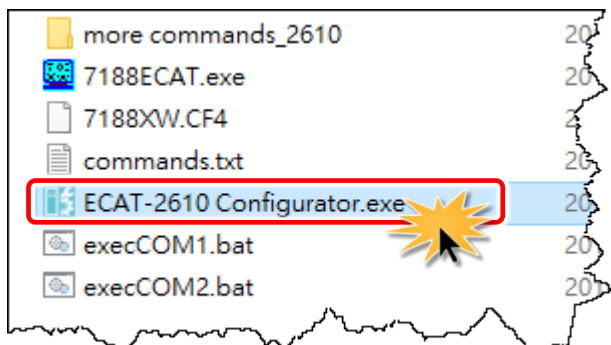


圖 4-2.2

➤ 左側視窗是用來設定配置數據並上傳至 ECAT-2610 模組上。

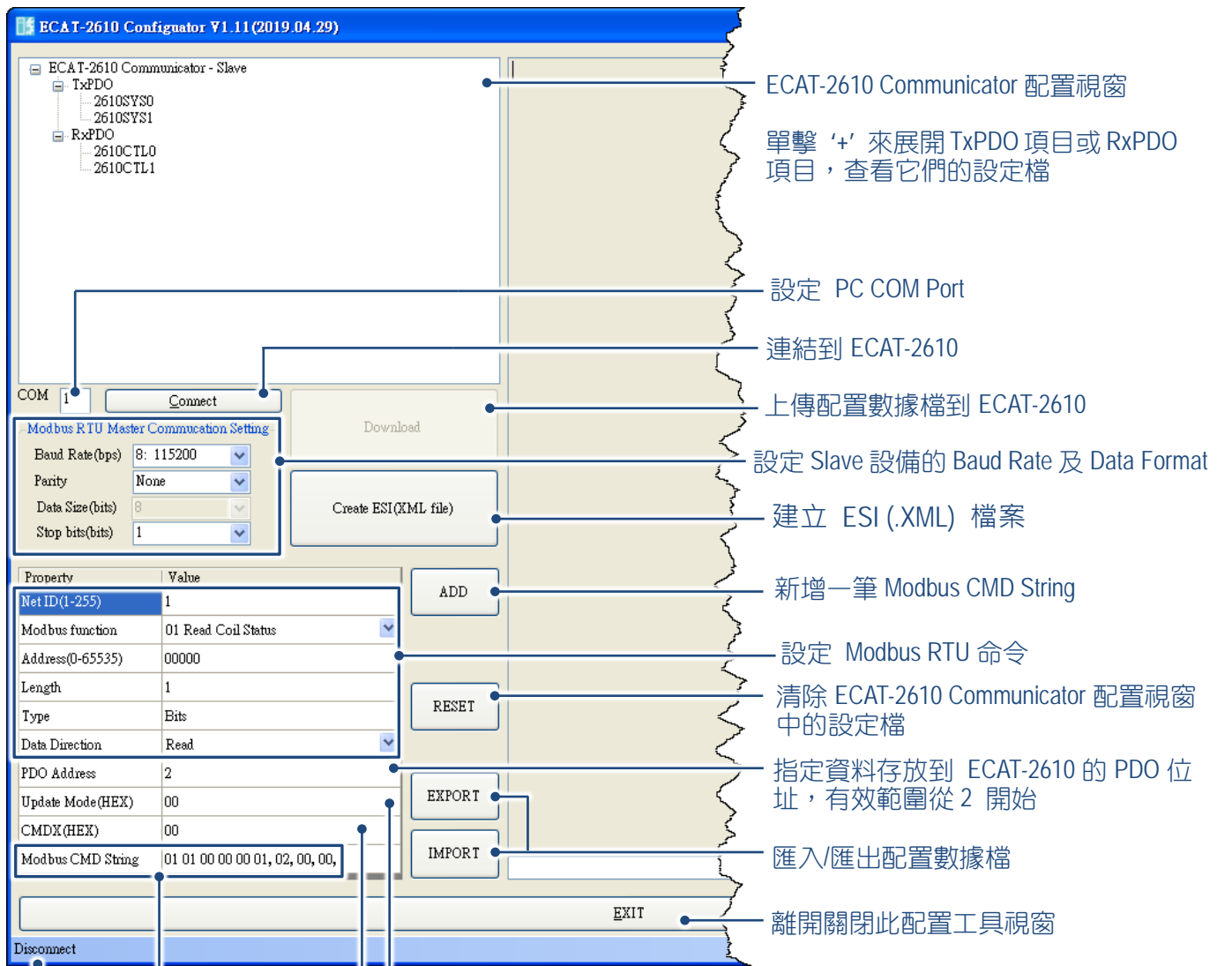
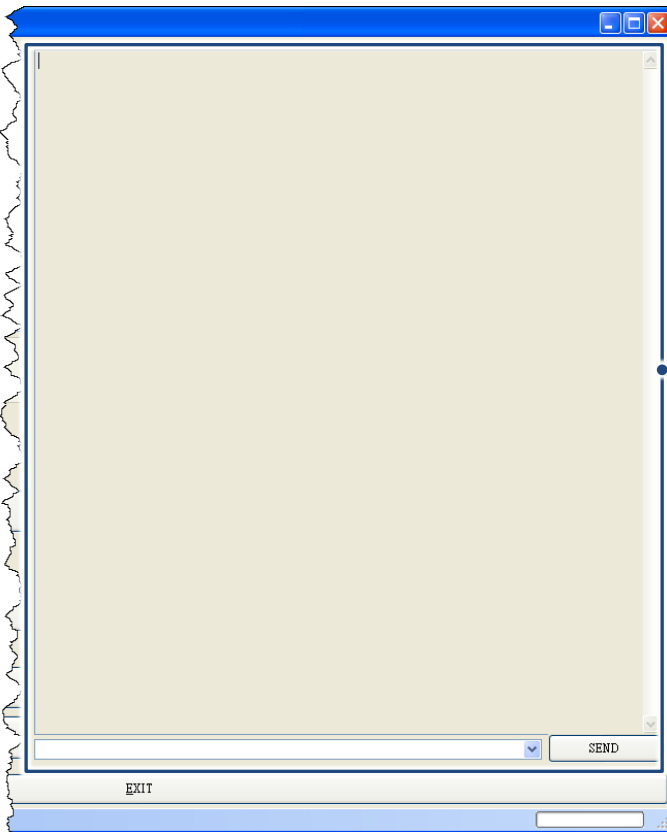


圖 4-2.3

- 狀態列
- 當 Property 參數配置完成後，此欄位將自動帶出 Modbus 命令字串
- ECAT-2610 Communicator 配置視窗
- 單擊 '+' 來展開 TxPDO 項目或 RxPDO 項目，查看它們的設定檔
- 設定 PC COM Port
- 連結到 ECAT-2610
- 上傳配置數據檔到 ECAT-2610
- 設定 Slave 設備的 Baud Rate 及 Data Format
- 建立 ESI (.XML) 檔案
- 新增一筆 Modbus CMD String
- 設定 Modbus RTU 命令
- 清除 ECAT-2610 Communicator 配置視窗中的設定檔
- 指定資料存放到 ECAT-2610 的 PDO 位址，有效範圍從 2 開始
- 匯入/匯出配置數據檔
- 離開關閉此配置工具視窗
- 設定更新模式 (HEX)
  - 00: 循環更新
  - ≠00: 當 InTxPDO[位址]上升時更新，詳細查看範例 05.Rising\_Trigger
- 設定特殊功能碼 (HEX)，預設: 00 (無)
  - 設定範圍如下:
  - 01: Power-On value
  - 02: byte-swap
  - 04: word-swap
  - 06: both-swap
  - 08: state change trigger
  - 10: constant output

➤ 左側視窗是用來原廠偵錯。



原廠偵錯 (請跳過)

圖 4-2.4

#### 步驟 4 修改 COM Port 碼、Baud Rate 及 Data Format 設定值

- 1 在 COM 欄位設定 COM Port 碼為您電腦所連接至 ECAT-2610 的 COM Port (如: PC 上的 COM Port 為 COM4，請在 COM 欄位輸入 4)。
- 2 依據您的 Modbus RTU 設備 (如: M-7050D) 來設定 Baud Rate 及 Data Format 值。

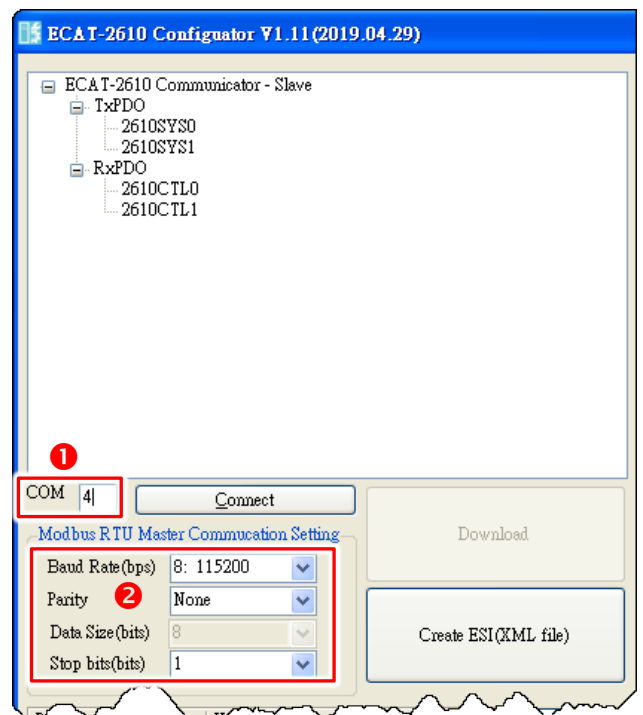


圖 4-2.5

### 步驟 5 設定 Modbus RTU 命令和相關屬性

M-7050D 模組具有 7 通道的數位輸入及 8 通道的數位輸出，詳細配置步驟如下：

設定寫 8 個數位輸出通道的 Modbus 命令(圖 4-2.6)，如下：

- ❶ 設定 Modbus 命令，請依據您的 Modbus RTU 設備(如: M-7050D) 在 “Net ID(1-255)”、“Modbus Function”、“Address (0-65535)” 及 “Length” 欄位設定適當的值。
- ❷ 在 “PDO Address” 欄位設定 RxPDO 位址。
- ❸ 在 “Update Mode(HEX)” 欄位設定更新模式。
- ❹ 在 “CMDX(HEX)” 欄位設定特殊功能碼。
- ❺ 單擊 “ADD” 按鈕來新增一筆 “OUTWORD02” 配置檔項目。

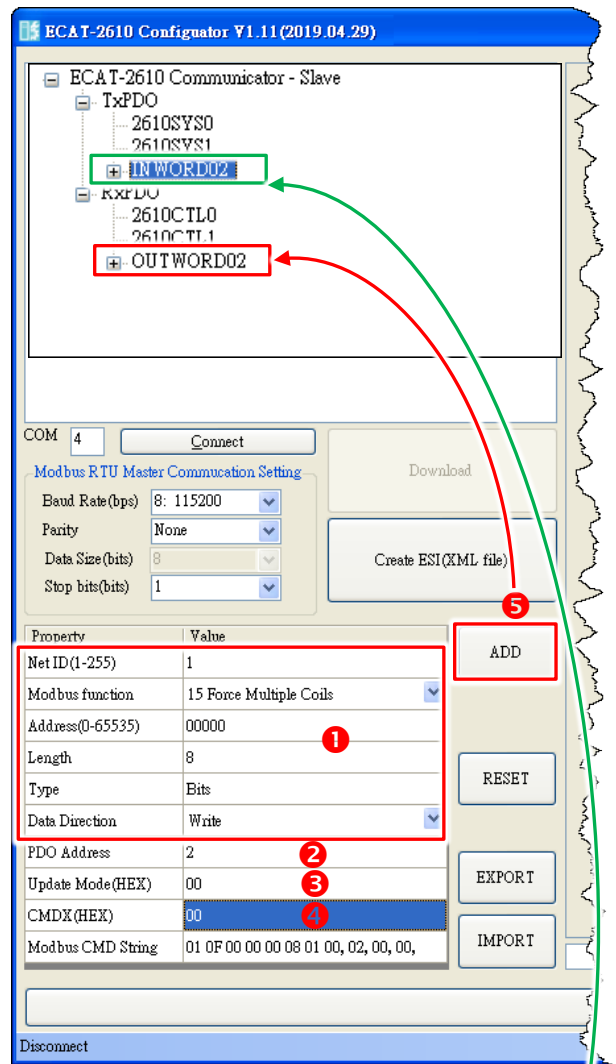


圖 4-2.6

設定讀 7 個數位輸入通道的 Modbus 命令 (圖 4-2.7)，如下：

- ❶ 設定 Modbus 命令，請依據您的 Modbus RTU 設備(如: M-7050D) 在 “Net ID(1-255)”、“Modbus Function”、“Address (0-65535)” 及 “Length” 欄位設定適當的值。
- ❷ 在 “PDO Address” 欄位設定 TxPDO 位址。
- ❸ 在 “Update Mode(HEX)” 欄位設定更新模式。
- ❹ 在 “CMDX(HEX)” 欄位設定特殊功能碼。
- ❺ 單擊 “ADD” 按鈕來新增一筆 “INWORD02” 配置檔項目。

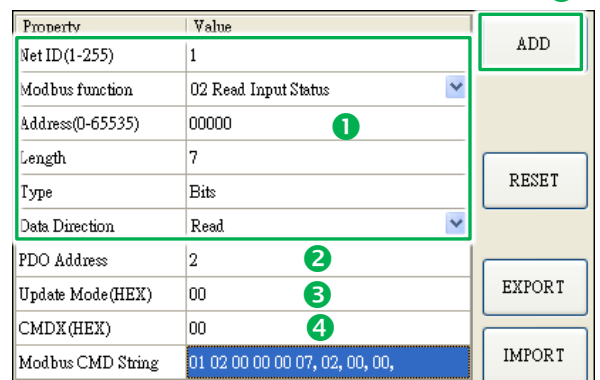
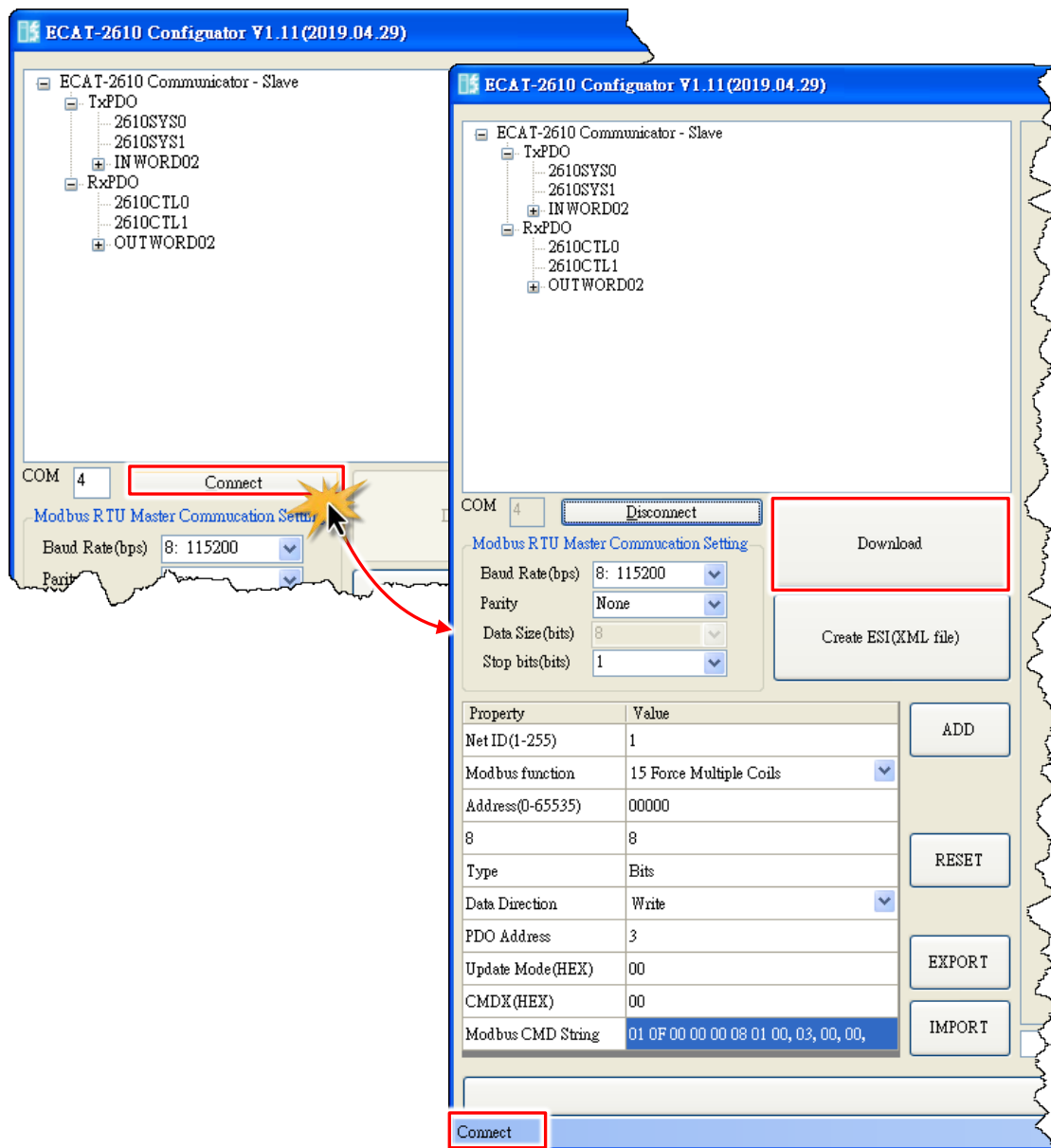


圖 4-2.7



### 步驟 6 按下 “Connect” 按鈕來連結 ECAT-2610

確認狀態列顯示 “Connect” 及 “Download” 按鈕將解鎖。



4-2.8

### 步驟 7 開始上傳

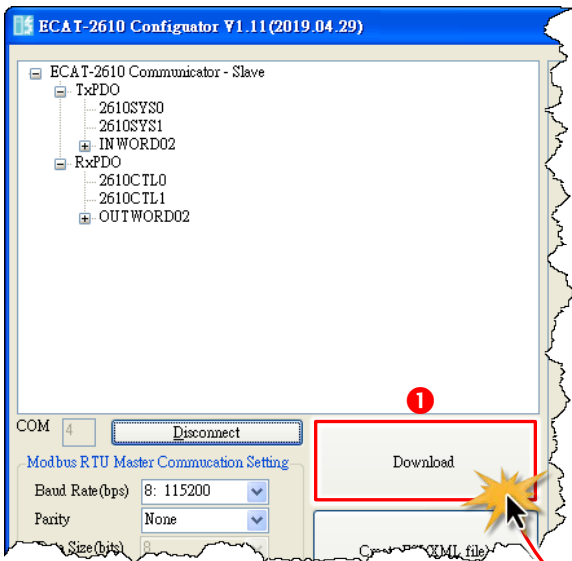
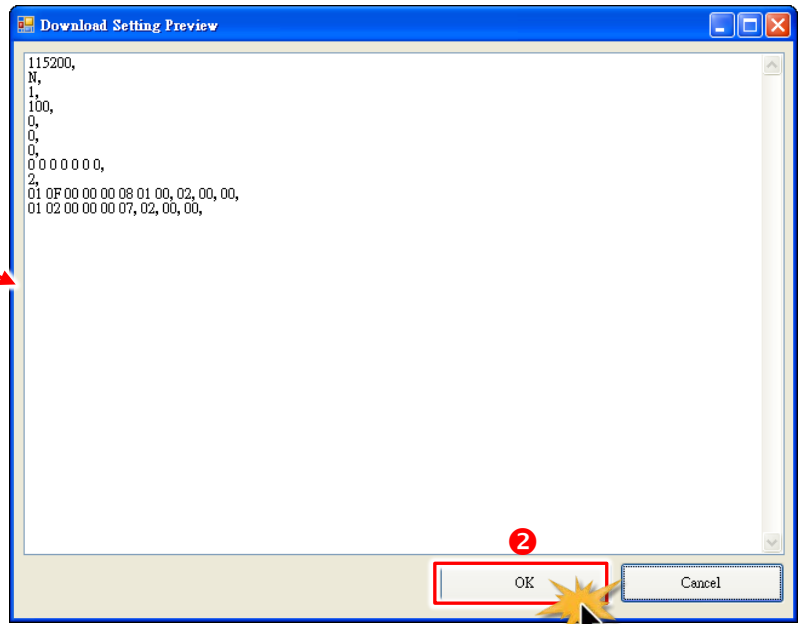


圖 4-2.9

- ❶ 按下 “Download” 按鈕將開啟 “Download Setting Preview” 視窗。
- ❷ 確認 “Download Setting Preview” 視窗中的配置數據是否正確 (詳細關於配置數據格式說明, 可參考至 [commands.txt](#)), 然後按下 “OK” 按鈕來繼續下一步。



- ❸ 彈跳出 “ECAT-2610 Configurator” 對話框, 要求您重新啟動 ECAT-2610 模組。此時, 將 ECAT-2610 模組斷電再上電, 重新啟動後按下 “確定” 按鈕來繼續上傳。

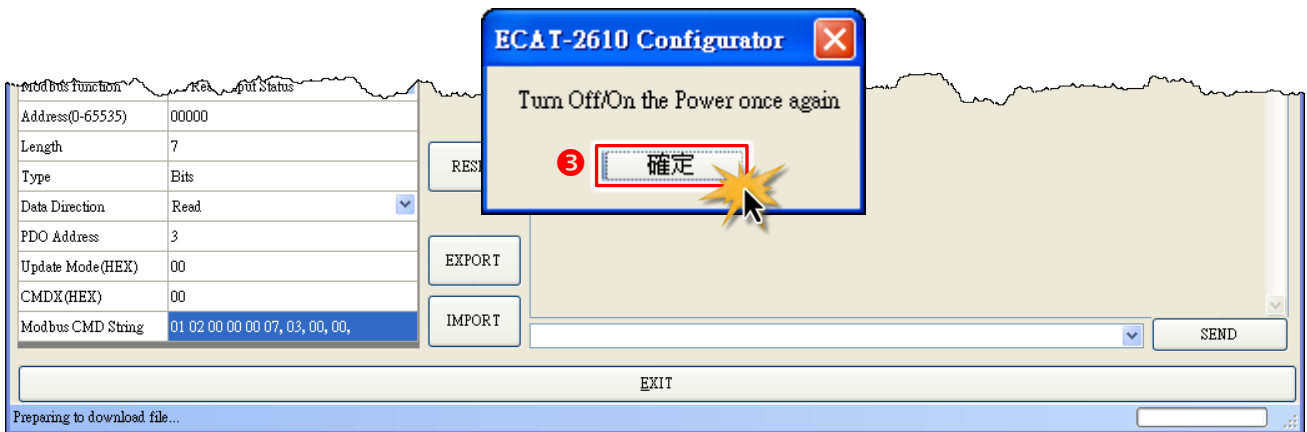


圖 4-2.10

④ 在狀態列將顯示上傳更新進度狀態。

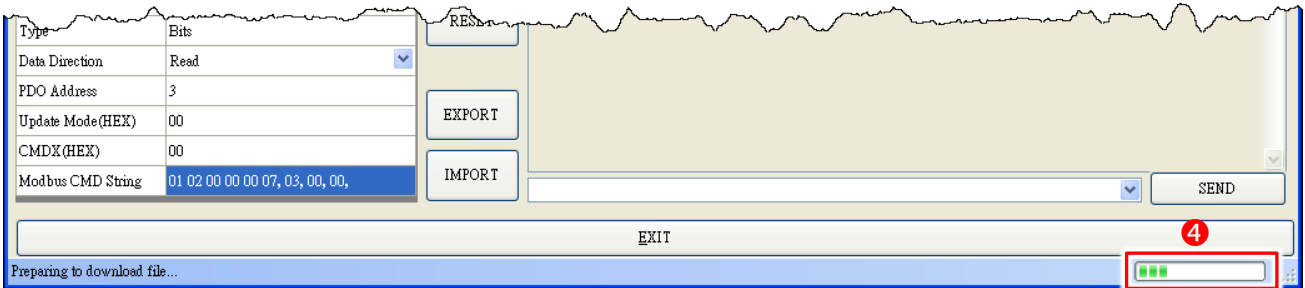


圖 4-2.11

⑤ 上傳成功後，會再次彈跳出“ECAT-2610 Configurator”對話框，要求您再次重新啟動 ECAT-2610 模組。此時，再次將 ECAT-2610 模組斷電再上電，重新啟動後按下“確定”按鈕來完成上傳。



圖 4-2.12

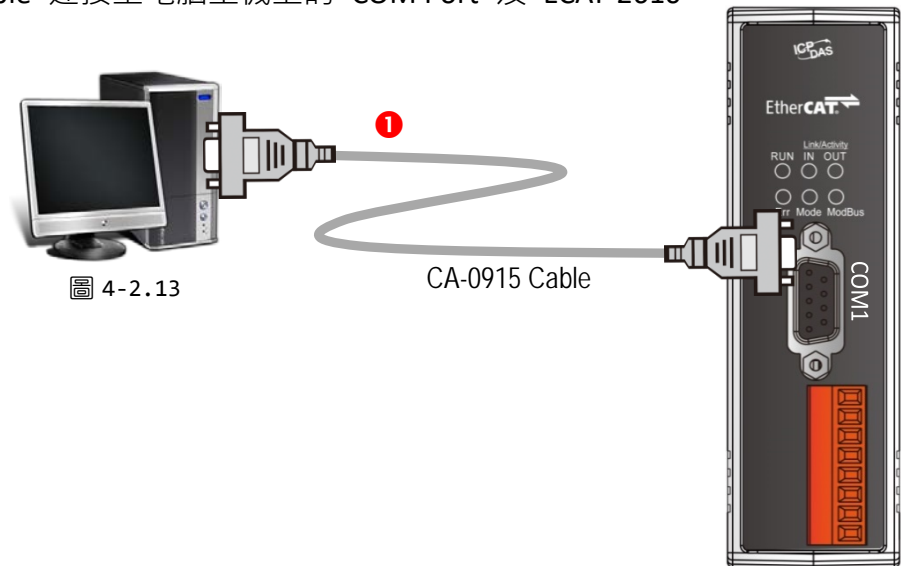
**⚠ 注意:**

如果經由 ECAT-2610 Configurator.exe 上傳配置數據檔失敗，此時模組不能正常運作，那麼請再執行一次手動配置數據檔及上傳，詳細說明參考至 [A4. “手動配置及上傳”](#)。

## 4.2.1 恢復至出廠預設值

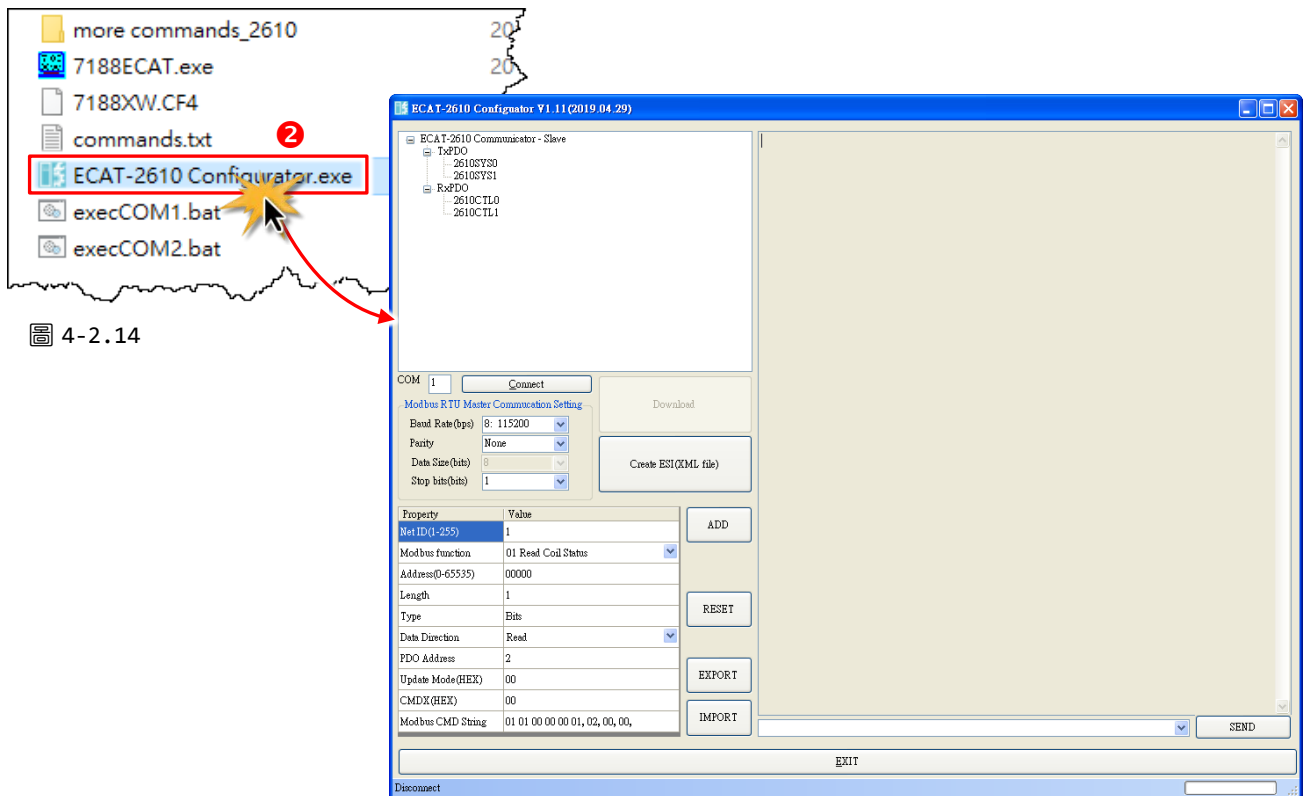
依據下列步驟來將所有參數恢復至出廠預設值：

**步驟 1:** 將配件 CA-0915 Cable 連接至電腦主機上的 COM Port 及 ECAT-2610 模組上的 COM1 Port。



**步驟 2:** 在 7188ECAT 資料夾中，雙擊 “ECAT-2610 Configurator.exe” 來開啟配置工具包。

**注意:** 如果您沒有 7188ECAT 資料夾，參考章節 4.2 “配置及上傳” 來下載。



**步驟 3:** 在 COM 欄位設定 COM Port 碼為您電腦所連接至 ECAT-2610 的 COM Port (如: PC 上的 COM Port 為 COM4，請在 COM 欄位輸入 4)。

**步驟 4:** 按下 “Connect” 按鈕來連結 ECAT-2610。

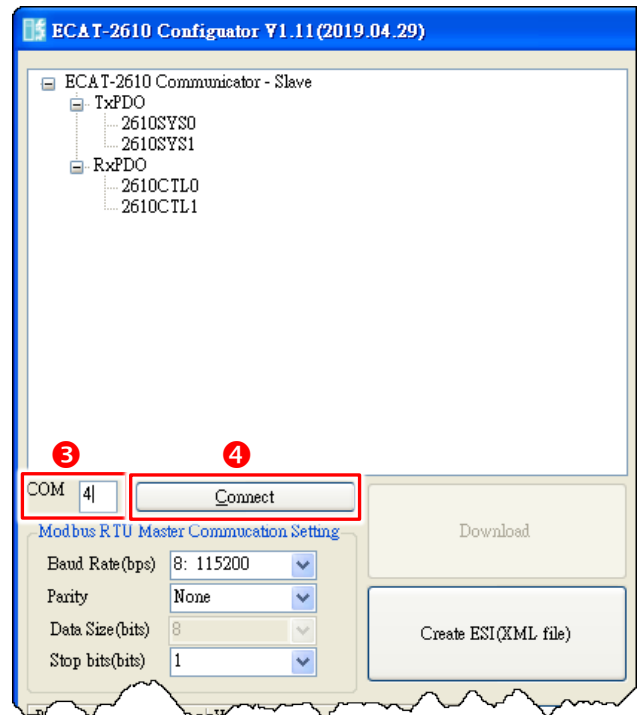


圖 4-2.15

**步驟 5:** 確認狀態列顯示 “Connect” 及 “Download” 按鈕將解鎖。

**步驟 6:** 按下 “Download” 按鈕將開啟 “ECAT-2610 Configurator” 對話框，要求您上傳出廠預設值到 ECAT-2610 模組，按下 “確定” 按鈕來繼續。

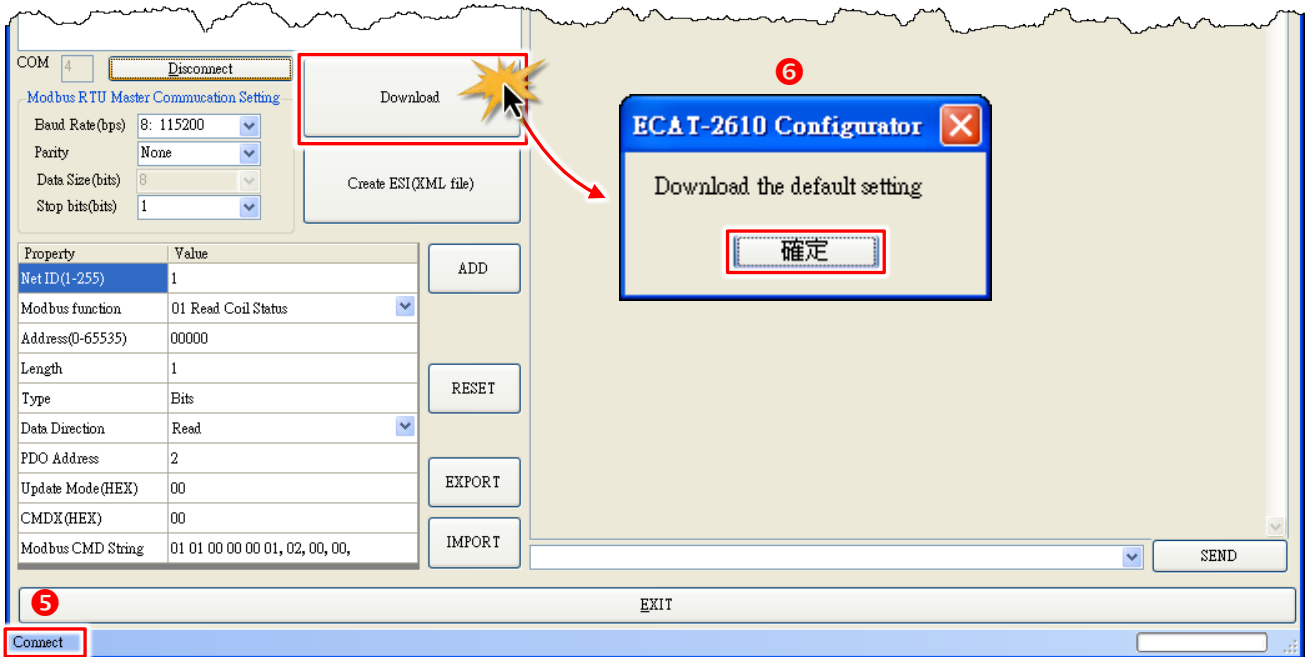
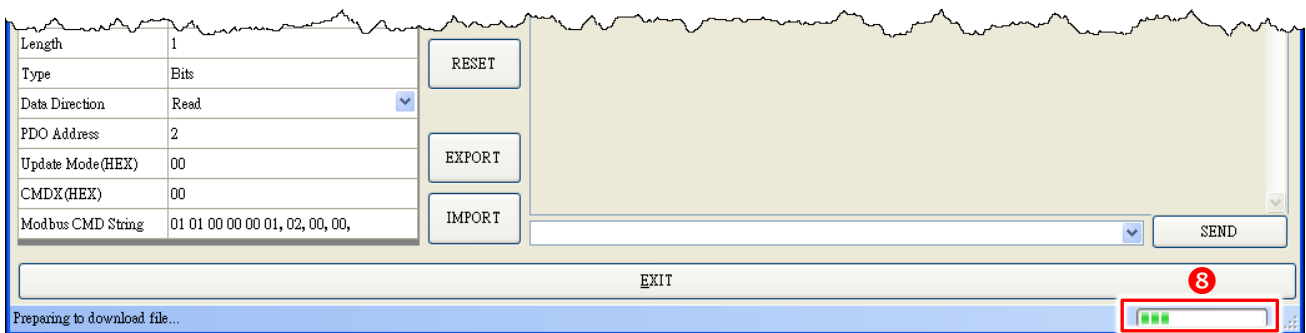


圖 4-2.16

**步驟 7:** 彈跳出 “ECAT-2610 Configurator” 對話框，要求您重新啟動 ECAT-2610 模組。此時，將 ECAT-2610 模組斷電再上電，重新啟動後按下 “確定” 按鈕來繼續上傳。



**步驟 8:** 在狀態列將顯示上傳更新進度狀態。



**步驟 9:** 上傳成功後，會再次彈跳出 “ECAT-2610 Configurator” 對話框，要求您再次重新啟動 ECAT-2610 模組。此時，再次將 ECAT-2610 模組斷電再上電，重新啟動後按下 “確定” 按鈕來完成上傳。



## 4.3 測試 Modbus RTU 設備

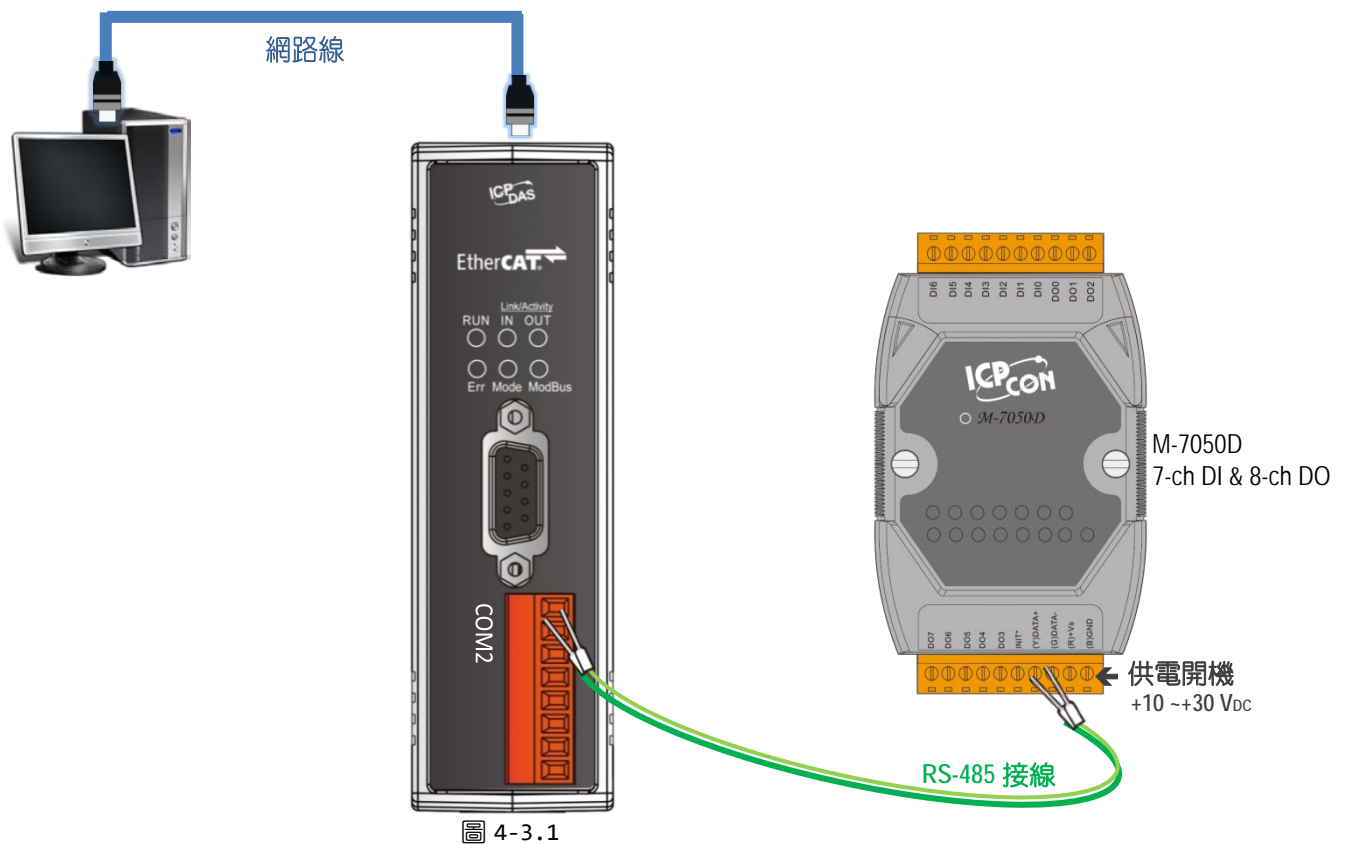
在測試 Modbus RTU 設備前請先確認配置檔設定完成並已上傳到 ECAT-2610 模組中，詳細操作參考至 [章節 4.2 “配置及上傳”](#)。

### ⚠ 注意

測試方式請依據您的 Modbus RTU 設備來執行。下面範例，我們將使用 M-7050D 模組來進行，而其它 Modbus RTU 設備，請參考各自的快速入門指南及使用手冊來執行。

### 步驟 1 將 Modbus RTU 設備連接至 ECAT-2610 模組

- ❶ ECAT-2610 模組保持在網路連線狀態，詳細接線方式參考[章節 3.2 “連接主站與電源”](#)。
- ❷ 將 Modbus RTU 設備連接至 ECAT-2610 模組上 COM2 (RS-485 bus)。
- ❸ 提供電源到 Modbus RTU 設備。



## 步驟 2 開啟 TwinCAT Master 軟體

安裝 ESI 檔案 (ICPDAS ECAT-2610.xml) 並執行 EtherCAT Master 軟體 (如: Beckhoff TwinCAT 2.X) , 詳細說明請參考 [章節 3.4 “搜尋模組”](#) 。

## 步驟 3 自動掃描

開啟模組電源並打開 TwinCAT System Manager(Config mode) , 開始掃描裝置後(如圖 4-3.2) , 之後所有的對話框皆為 “OK” , 使模式設定值在 FreeRun 模式。

### 注意

連結 ECAT-2610 模組至 EtherCAT , 連接時系統必需在安全、斷電的狀態。

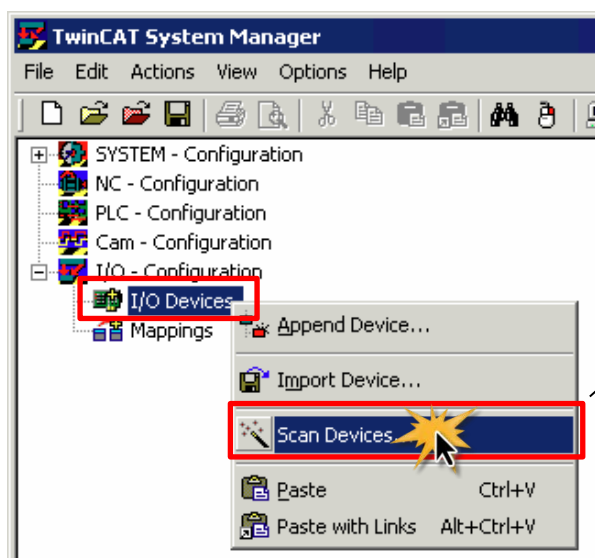


圖 4-3.2

### 掃描設備

(I/O Devices-> 按右鍵-> Scan Devices...)



## 步驟 4 透過 TwinCAT 設置每個模組

在 TwinCAT System Manager 左邊欄位的視窗，單擊您想要設定的 EtherCAT BOX 後開啟樹狀分支 (如下圖 4-3.3) 後單擊 **Inxx** 或 **Outxx** 作設定。

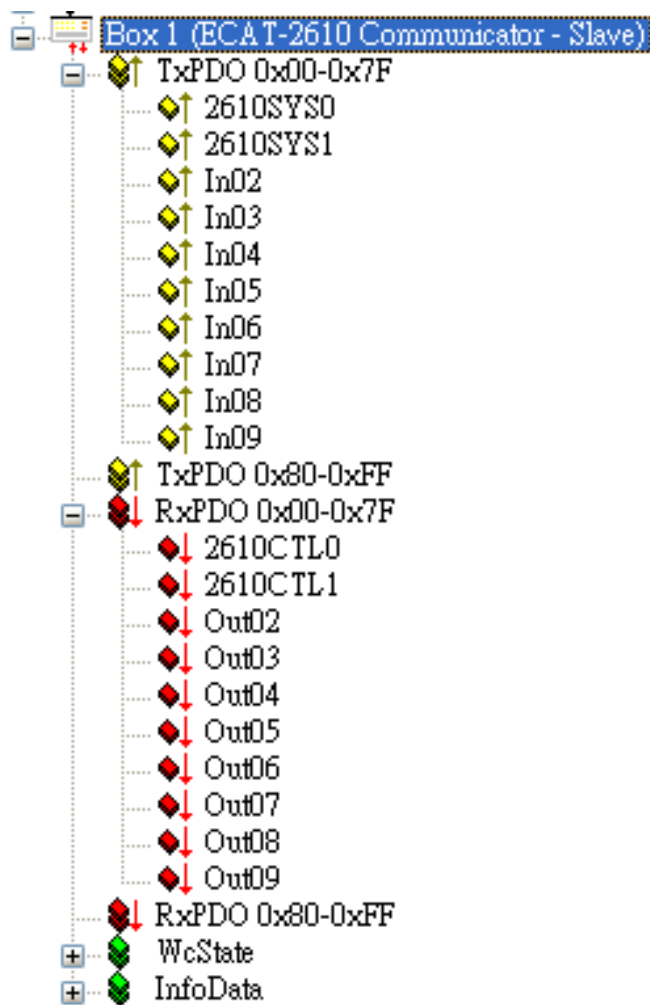


圖 4-3.3

➤ 確認 M-7050D 模組的 DO 功能測試結果，如下：

- ❶ 在左側視窗中，按下 “Out02”。
- ❷ 在右側視窗中，按下 “Online”。
- ❸ 再按下 “Write” 按鈕來開啟 “Set Value Dialog” 對話框。
- ❹ 在 “Set Value Dialog” 對話框中，在 “Hex:” 欄位輸入 0x00ff (設定所有 DO 都 ON 起)，再按下 OK 按鈕。

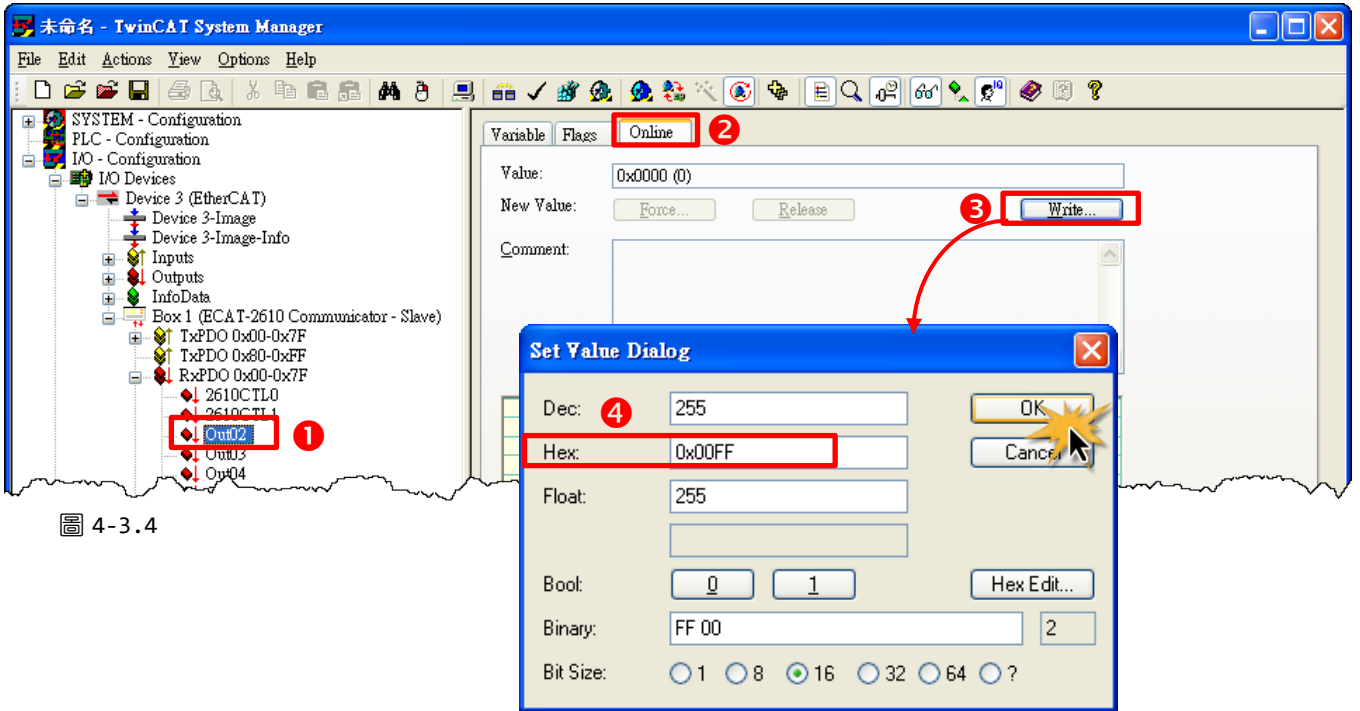


圖 4-3.4

❺ 檢查 M-7050D 模組上的數位輸出的 LED 顯示燈全部亮起。

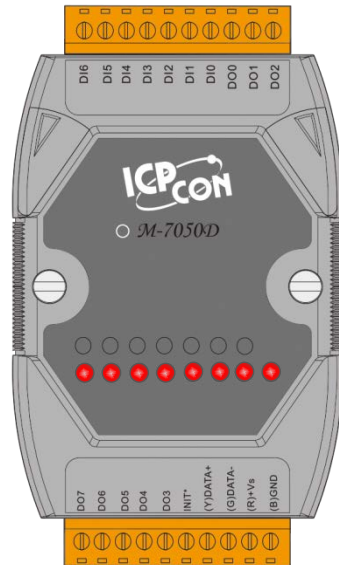


圖 4-3.5

# 5. 啟動 Modbus RTU Master



## 注意

在“啟動 Modbus RTU 從端設備”前，請先確保 ECAT-2611 模組正常運行，詳細參考第 3 章“啟用 ECAT-261x 模組”。

下面範例，我們將使用 PC 作為 Modbus RTU Master 來進行測試，而其它 Modbus RTU Master，請參考各自的快速入門指南或使用手冊來執行。

此章節將介紹如何配置 ECAT-2611 模組，包括 NetID、Baud Rate、Data Format...等，以及如何與 Modbus RTU Master 連接。

## 5.1 配置及上傳

### 步驟 1 將下載 Cable 連接至電腦主機及 ECAT-2611 模組

- ❶ ECAT-2611 模組斷電關機。
- ❷ 配件 CA-0915 Cable 連接至電腦主機上的 COM Port 及 ECAT-2611 上的 COM1 Port。
- ❸ ECAT-2611 模組供電開機。

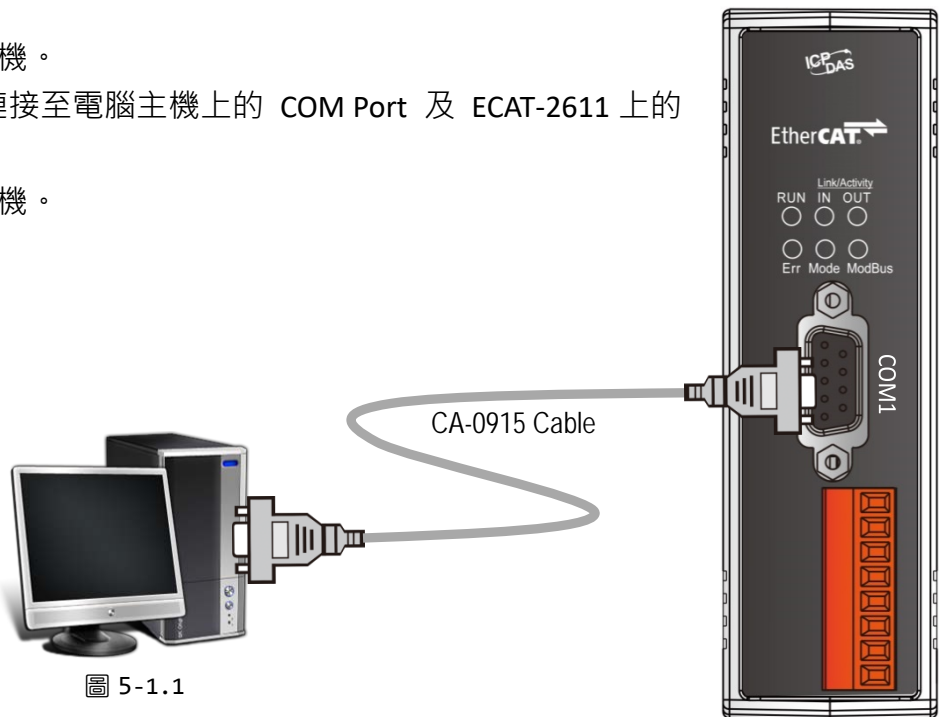


圖 5-1.1







## 步驟 2 下載 ECAT-2611 Utl xxxxxxI 壓縮檔

❶ “ECAT-2611\_Utl\_xxxxxx.zip” 可以從泓格科技網站下載。下載位置如下：

 [http://ftp.icpdas.com/pub/cd/fieldbus\\_cd/ethercat/slave/ecat-2000/software/](http://ftp.icpdas.com/pub/cd/fieldbus_cd/ethercat/slave/ecat-2000/software/)

❷ 將 ECAT-2611\_Utl\_xxxxxx.zip 解壓縮後，包含 7188ECAT 資料夾是配置/偵錯工具。

❸ 將 7188ECAT 資料夾複製到電腦硬碟上 (如: E:\)，資料夾中應包含下列檔案：

 7188ECAT.exe	執行檔，詳細參考 <a href="#">A4-2 “上傳配置檔”</a>
 7188XW.CF4	7188ECAT.exe 控制檔
 commands.txt	配置檔，用來設定 Modbus RTU 設備的文件檔。ECAT-2611 將透過此配置檔來與 Modbus RTU Master 進行通訊
 execCOM1.bat	使用電腦主機的 COM1 來載入配置數據檔至 ECAT-2611，詳細參考 <a href="#">A4-2 “上傳配置檔”</a>
 execCOM2.bat	使用電腦主機的 COM2 來載入配置數據檔至 ECAT-2611，詳細參考 <a href="#">A4-2 “上傳配置檔”</a>
 ECAT-2611 Configurator.exe	ECAT-2611 Configurator.exe 配置工具包

## 步驟 3 執行 ECAT-2611 Configurator.exe

雙擊 “ECAT-2611 Configurator.exe” 來開啟配置工具包。

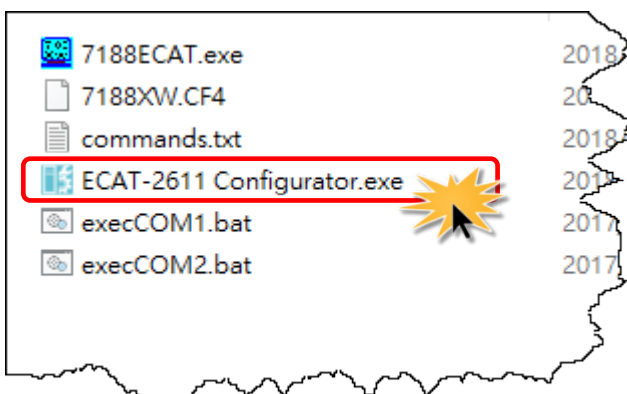
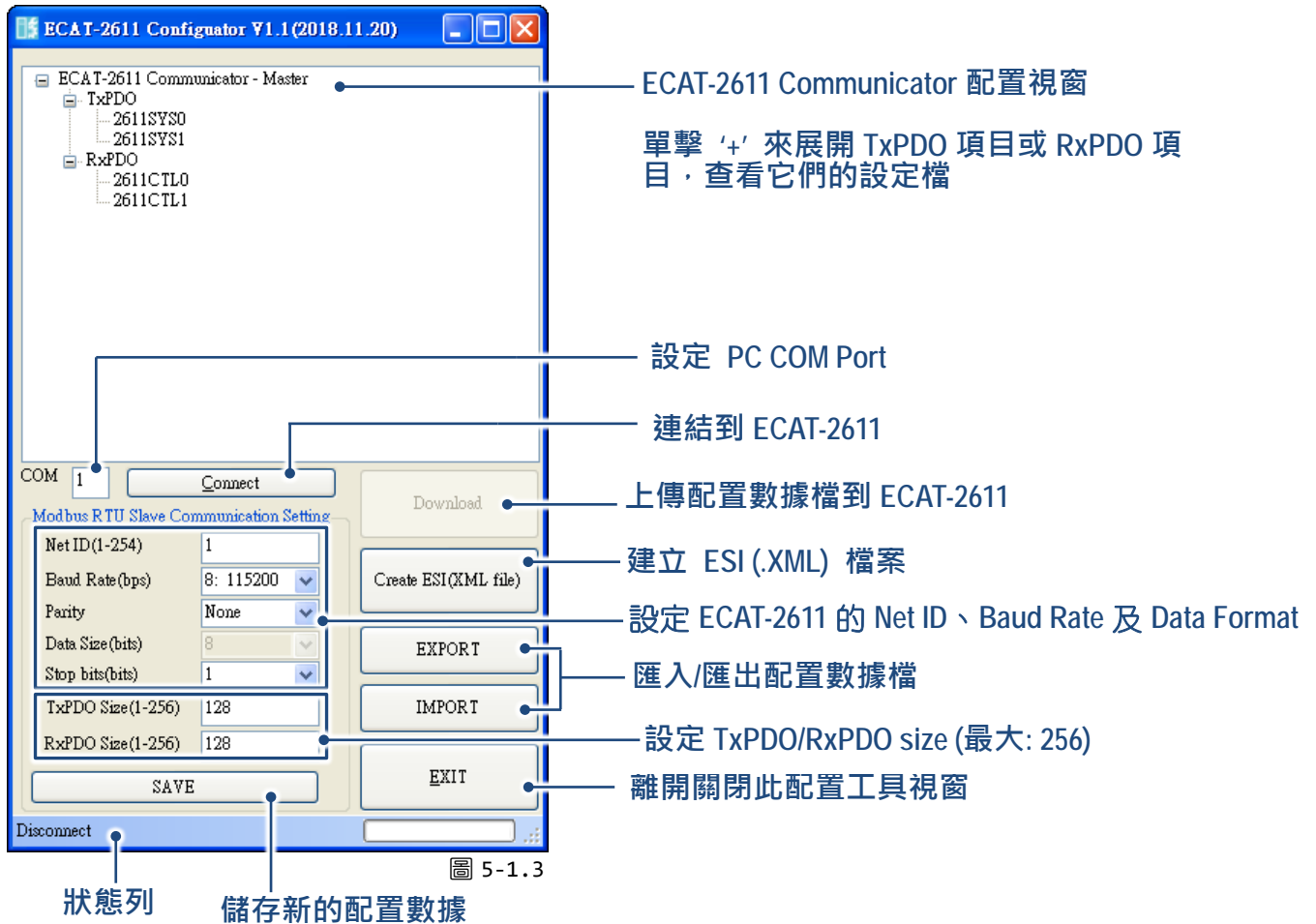


圖 5-1.2

➤ ECAT-2611 Configurator.exe 是用來設定配置數據並上傳至 ECAT-2611 模組上。



#### 步驟 4 修改 COM Port 碼、Baud Rate 及 Data Foramt 設定值

- ❶ 在 COM 欄位設定 COM Port 碼為您電腦所連接至 ECAT-2611 的 COM Port，詳細看圖 5-1.4。(範例: PC 上的 COM Port 為 COM8，請在 COM 欄位輸入 8)。
- ❷ 在相關欄位下拉式選單中設定 Net ID、Baud Rate、Data Format 及 TxPDO/RxPDO size 值，詳細看圖 5-1.4。(範例: 修改 Net ID 為 2)。

ECAT-2611 出廠預設值，如下表:

COM	1	Data Size (bits)	8
Net ID	1	Stop bits (bits)	1
Baud Rate (bps)	115200	TxPDO size	128
Parity	None	RxPDO size	128

- 3 按下 “Connect” 按鈕來連結 ECAT-2611。
- 4 確認狀態列顯示 “Connect” 及 “Download” 按鈕將解鎖。

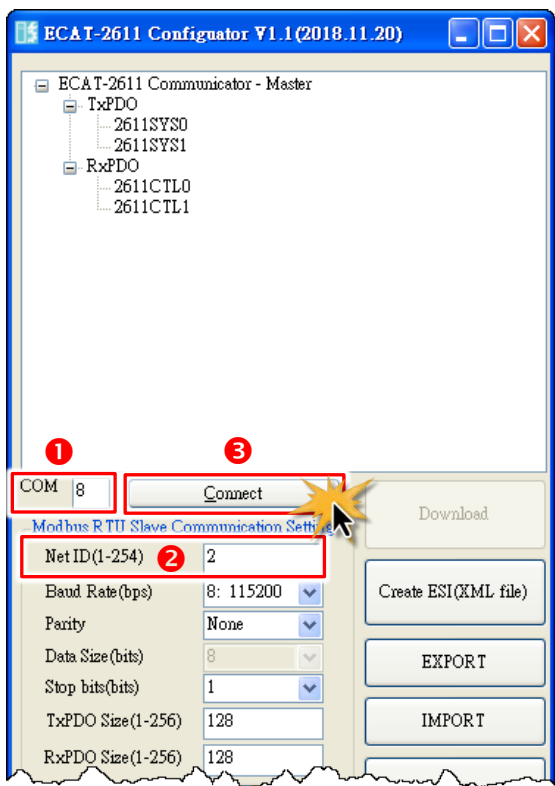


圖 5-1.4

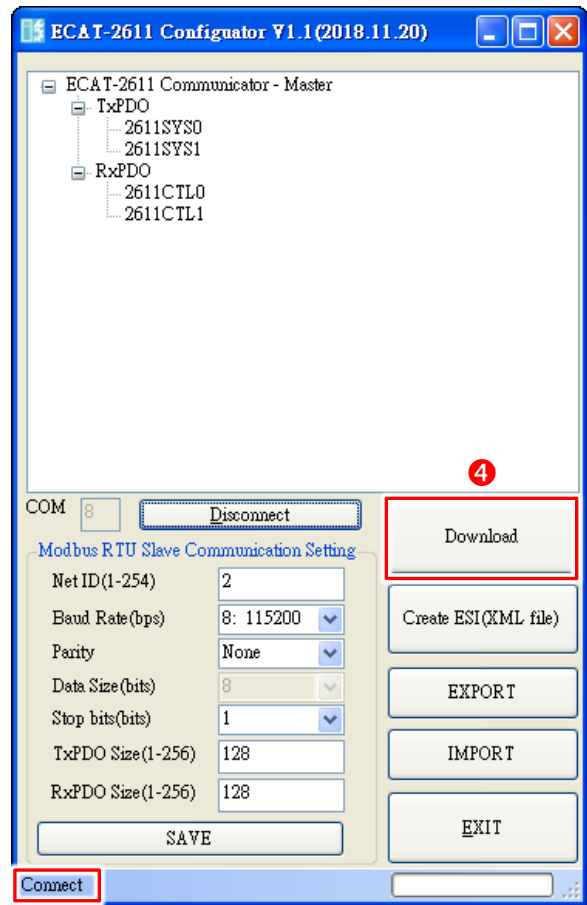


圖 5-1.5

### 步驟 7 開始上傳

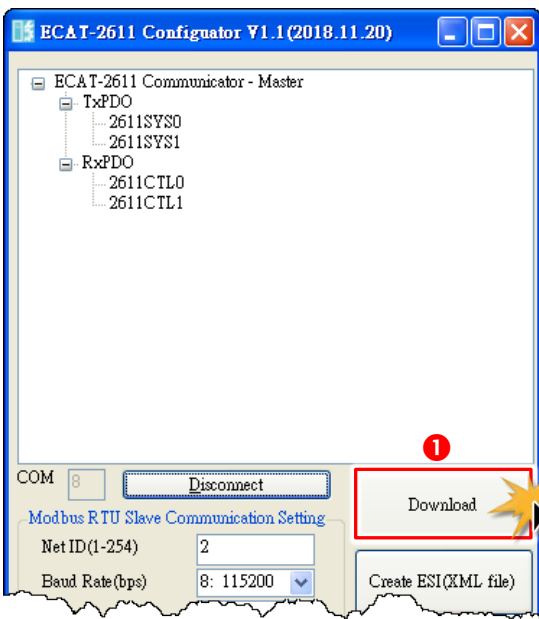


圖 5-1.6

- 1 按下 “Download” 按鈕。
- 2 彈跳出 “ECAT-2611 Configurator” 對話框，要求您重新啟動 ECAT-2611 模組。此時，將 ECAT-2611 模組斷電再上電，重新啟動後按下 “確定” 按鈕來繼續上傳。



③ 在狀態列將顯示上傳更新進度狀態。

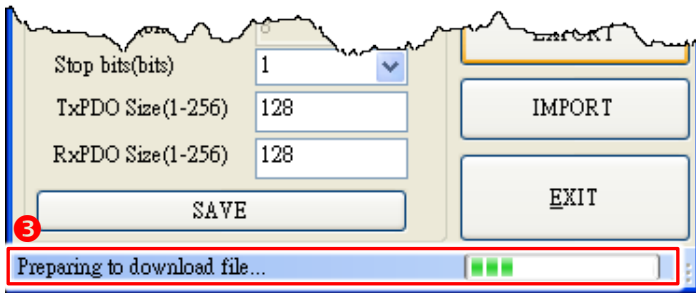


圖 5-1.7

④ 上傳成功後，會再次彈跳出“ECAT-2611 Configurator”對話框，要求您再次重新啟動 ECAT-2611 模組。此時，再次將 ECAT-2611 模組斷電再上電，重新啟動後按下“確定”按鈕來完成上傳。

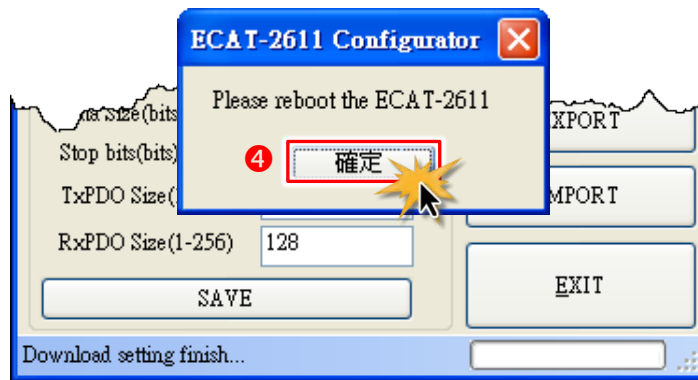


圖 5-1.8

**⚠ 注意:**

如果經由 ECAT-2611 Configurator.exe 上傳配置數據檔失敗，此時模組不能正常運作，那麼請再執行一次手動配置數據檔及上傳，詳細說明參考 [A4. “手動配置及上傳”](#)。

## 5.2 連接 Modbus RTU Master

此範例，使用 PC 作為 Modbus RTU Master，詳細說明如下：

### 步驟 1 將 Modbus RTU Master 連接至 ECAT-2611 模組

- ❶ ECAT-2611 模組保持在網路連線狀態，詳細接線方式參考章節 3.2 “連接網站與電源”。
- ❷ 將 Modbus RTU Master 連接至 ECAT-2611 模組上 COM2 (RS-485 bus)。
- ❸ 提供電源到 Modbus RTU 設備。

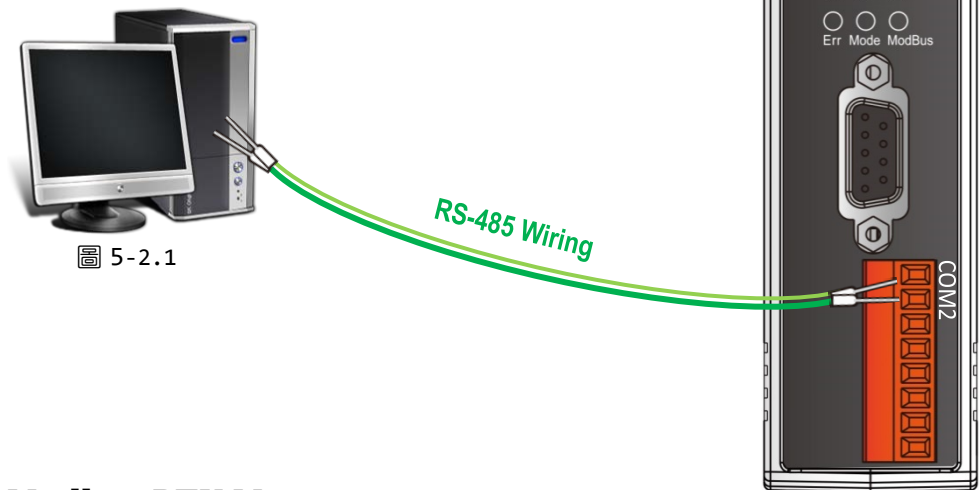


圖 5-2.1

### 步驟 2 配置電腦為 Modbus RTU Master

- ❶ 在您的電腦安裝 Modbus Master Tools (如: Modbus Poll, 是一個 Modbus Master 模擬軟體)。  
下載位置: <https://www.modbustools.com/download.html>

- ❷ 執行 **Modbus Poll.exe** 程式。  
單擊 “Setup” 功能選單中的 “Read/Write Definition F8” 項目來開啟 “Read/Write Definition” 對話框。

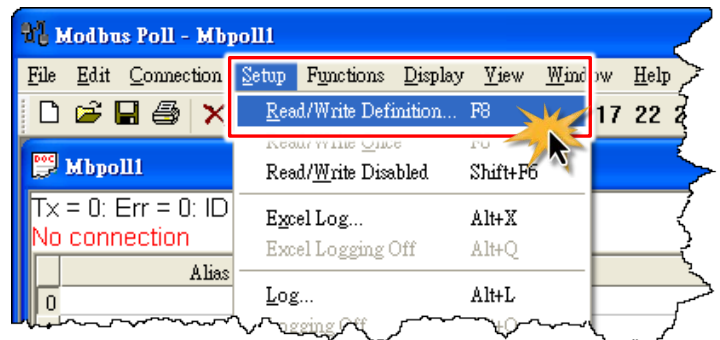


圖 5-2.2

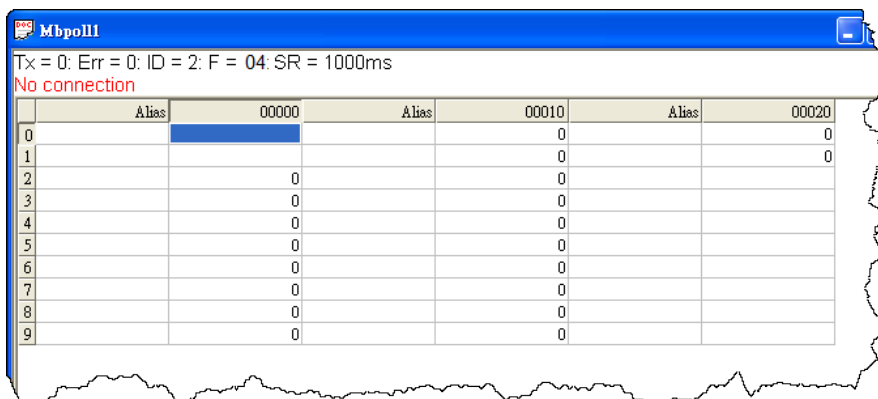
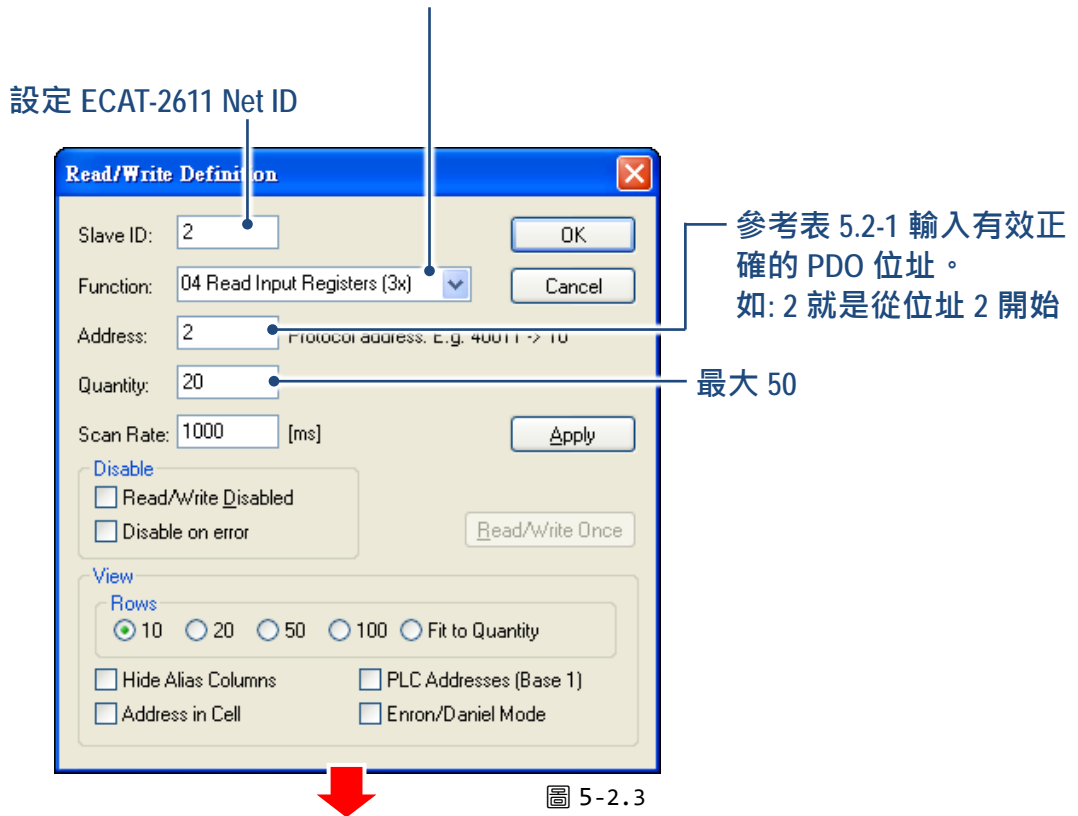


⑤ 在 “Read/Write Definition” 對話框，依據 ECAT-2611 模組來設定 Slave ID、Function code、Address 及 Quantity...等值，然後再按下 “OK” 按鈕。

範例: 使用 Function 04 每隔 1000 ms 從 Slave ID 2 的 ECAT-2611 模組從位址開始 2 取讀 20 個暫存器 (Register)。

表 5.2-1: ECAT-2611 支援功能如下

功能碼	名稱	有效讀/寫 PDO 位址	章節
03 (0x03)	Read holding registers	Readback Multiple TxPDO [02 ~ FF]	6.3
04 (0x04)	Read input registers	Read Multiple RxPDO [00 ~ FF]	6.4
06 (0x06)	Write single register	Write single TxPDO [02 ~ FF]	6.6
16 (0x10)	Write multiple registers	Write Multiple TxPDO [02 ~ FF]	6.8



④ 單擊 “**Connection**” 功能選單中的 “**Connect... F3**” 項目來開啟 “Connection Setup” 對話框。

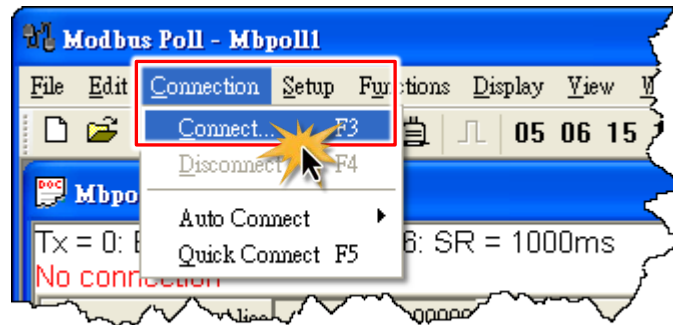


圖 5-2.4

⑤ 在 “Connection Setup” 對話框，從 “**Connection**” 下拉式選單中選擇 “**Serial Port**” 項目，並配置相關序列埠設定後，按下 “**OK**” 按鈕。

設定連接類型，依據 Modbus Master (如:PC)  
連接 ECAT-2611 的方式

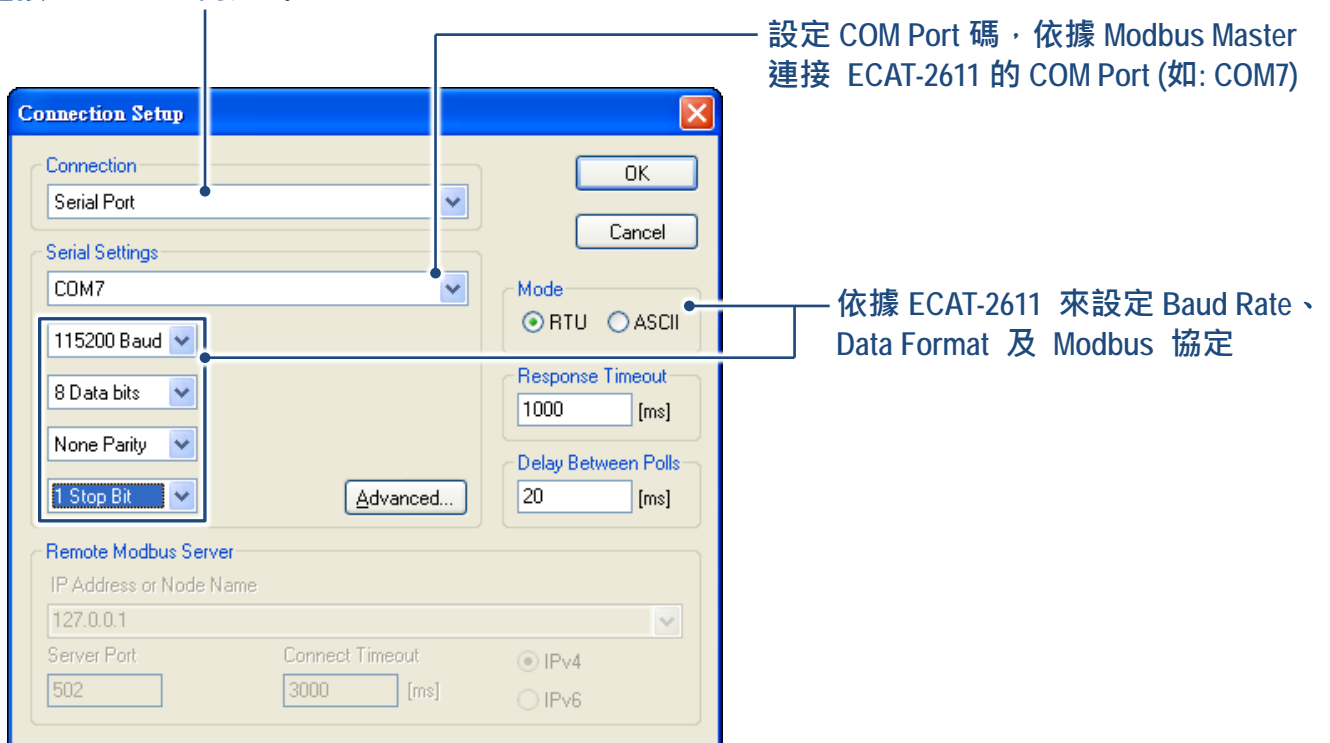


圖 5-2.5

⑥ 在 “Mbpoll1” 視窗 · 確認連線成功 (ERR = 0)。

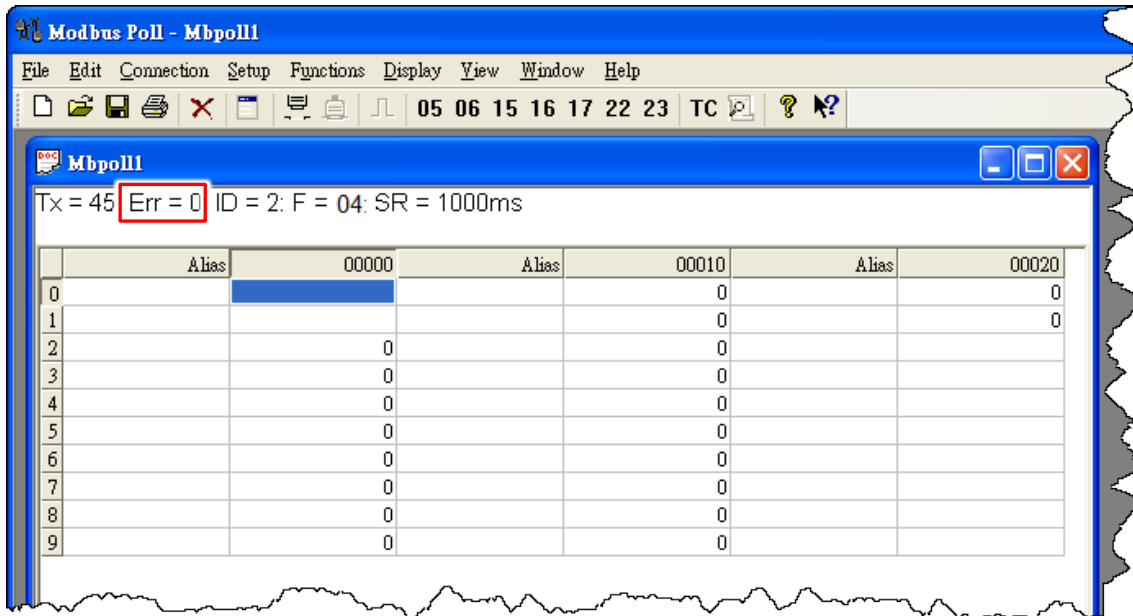


圖 5-2.6

## 5.3 測試 Modbus RTU Master

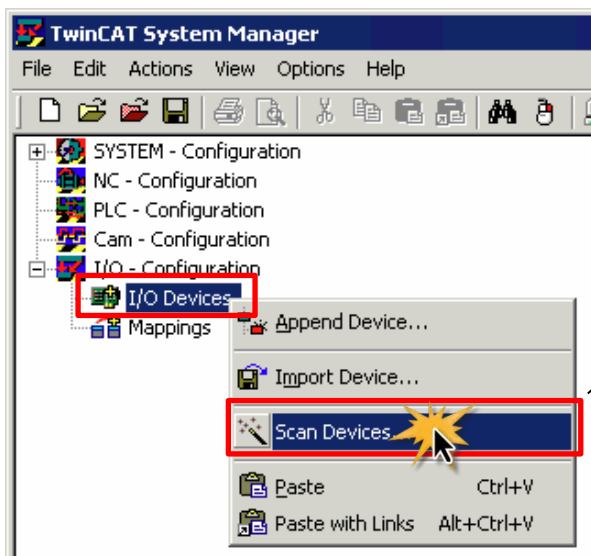
在測試 Modbus RTU Master 前請先確認配置檔設定完成並已載入到 ECAT-2611 模組中，詳細操作參考[章節 5.1 “配置及上傳”](#)。

### 步驟 1 開啟 TwinCAT Master 軟體

- 1 安裝 ESI 檔案 (ICPDAS ECAT-2611.xml)，並執行 EtherCAT Master 軟體 (如: Beckhoff TwinCAT 2.X)，詳細說明請參考[章節 3.4 “搜尋模組”](#)。
- 2 開啟模組電源並打開 TwinCAT System Manager(Config mode)，開始掃描裝置後(如圖 5-3.1)，之後所有的對話框皆為 “OK”，使模式設定值在 FreeRun 模式。

#### ⚠ 注意

連結 ECAT-2611 模組至 EtherCAT，連接時系統必需在安全、斷電的狀態。



掃描設備  
(I/O Devices-> 按右鍵-> Scan Devices...)

圖 5-3.1

## 步驟 2 透過 TwinCAT 設置每個模組

在 TwinCAT System Manager 左邊欄位的視窗，單擊您想要設定的 EtherCAT BOX 後開啟樹狀分支 (如下圖 5-3.2) 後單擊 **Inxx** 或 **Outxx** 作設定。

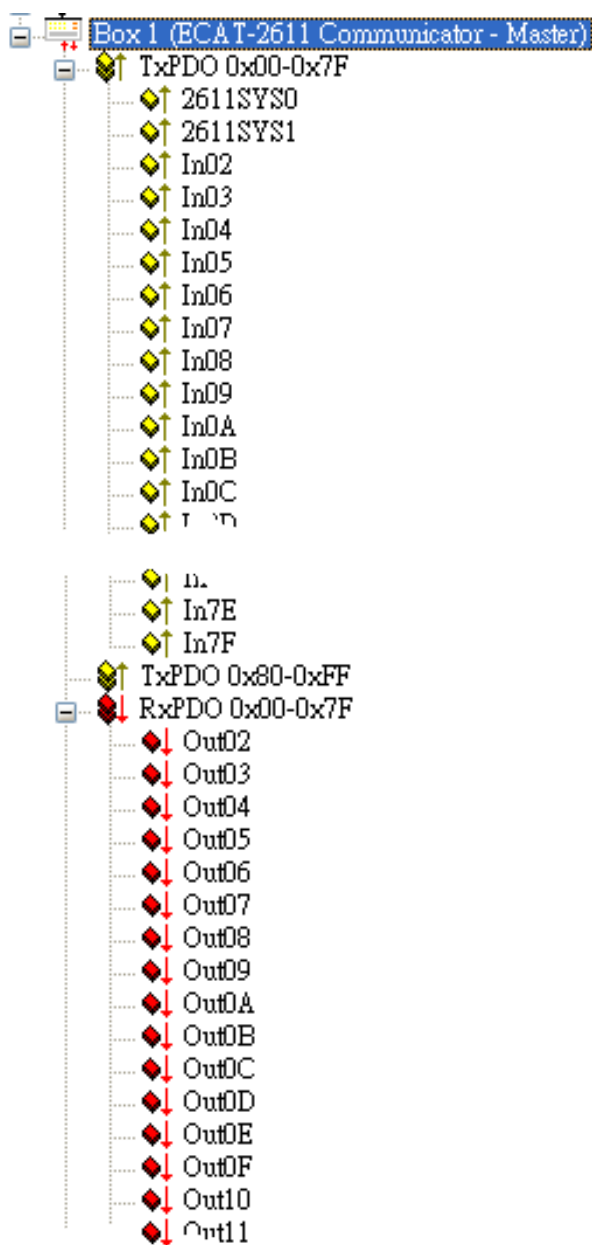


圖 5-3.2

➤ 驗證 Function 04 “Read holding registers” 的功能測試結果，如下：

- ❶ 在 “Modbus Poll” 程式，設定 Founction 04 及相關參數，詳細請參考[節章 5.2 “連接 Modbus RTU Master”](#)。
- ❷ 在 TwinCAT System Manager 左邊欄位的視窗，按下 “Out02”。
- ❸ 然後在右側視窗中，按下 “Online”。
- ❹ 在按下 “Write” 按鈕來開啟 “Set Value Dialog” 對話框。
- ❺ 在 “Set Value Dialog” 對話框中的 “Dec:” 欄位輸入 10，按下 OK 按鈕。
- ❻ 請重覆步驟 2 到 5，將 Out03 到 Out0A 的值也設定為 10。

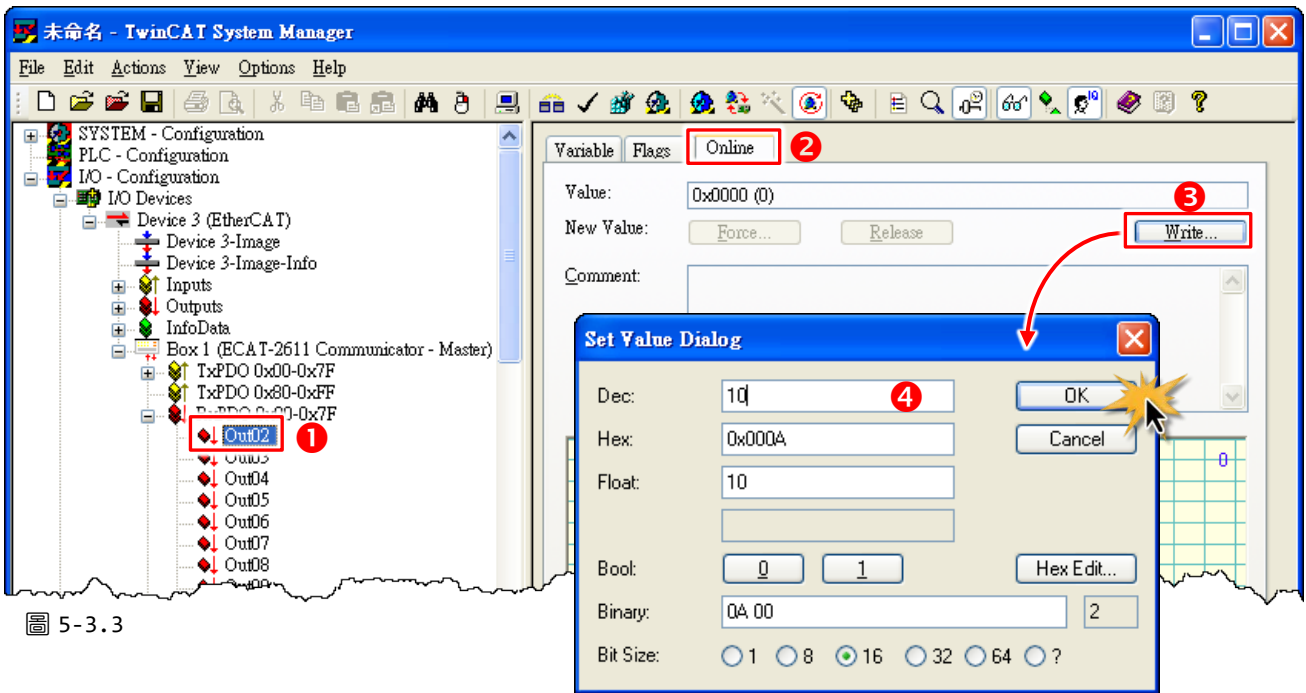


圖 5-3.3

❷ 回到 “Modbus Poll” 程式，確認位址 2 到 10 欄位值顯示 10。

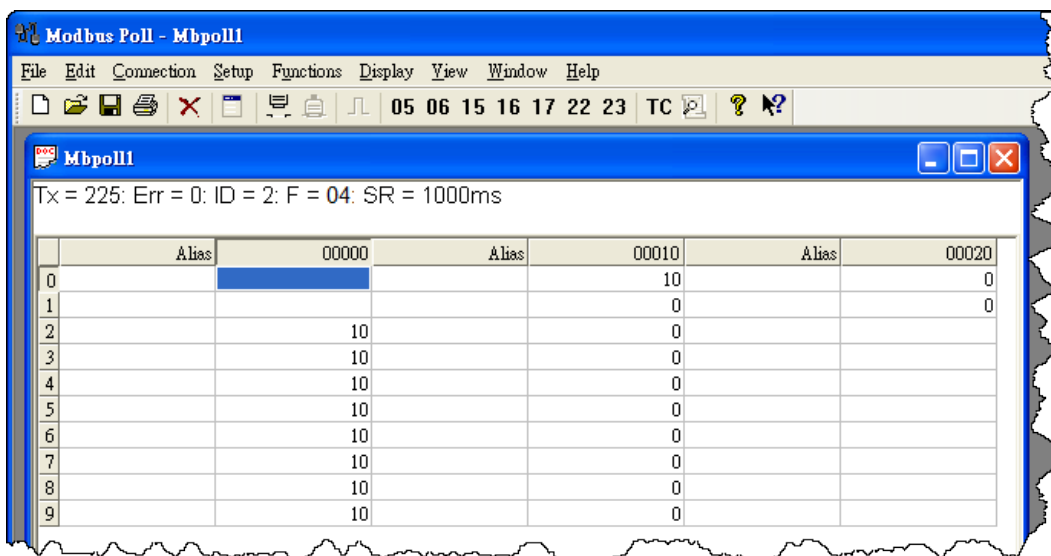


圖 5-3.4

➤ 驗證 Function 16 “Write multiple registers” 的功能測試結果，如下：

❶ 在 “Modbus Poll” 程式，設定 Function 16 及相關參數，詳細請參考[節章 5.2 “連接 Modbus RTU Master”](#)。

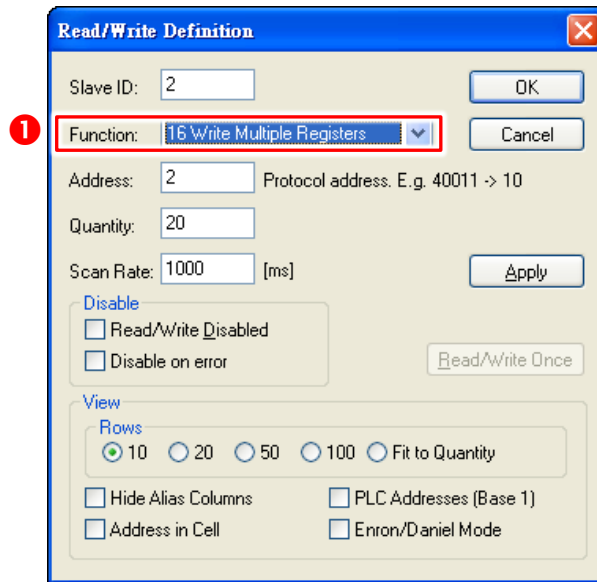


圖 5-3.5

❷ 在 “Modbus Poll” 程式，**雙擊位址 2 欄位**來開啟 “Enter signed int 16” 對話框。

❸ 在 “Enter signed int 16” 對話框，輸入 “20”，按下 **OK** 按鈕。

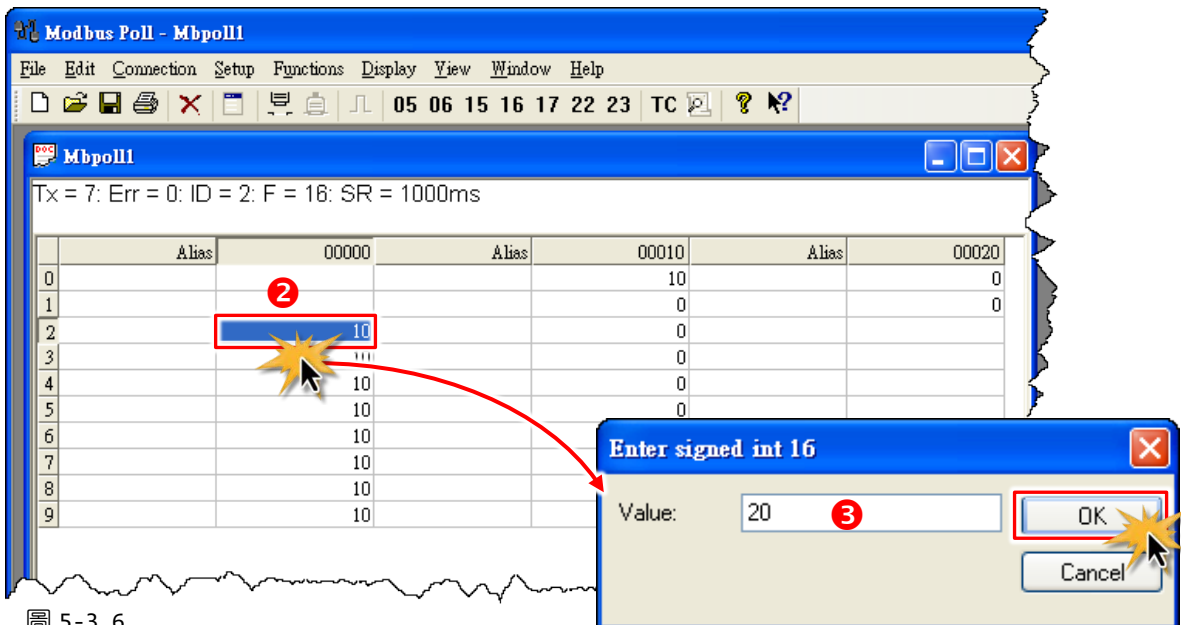


圖 5-3.6

- ④ 請重覆步驟 2 到 3，將位址 3 到 10 的值都設定為 20。

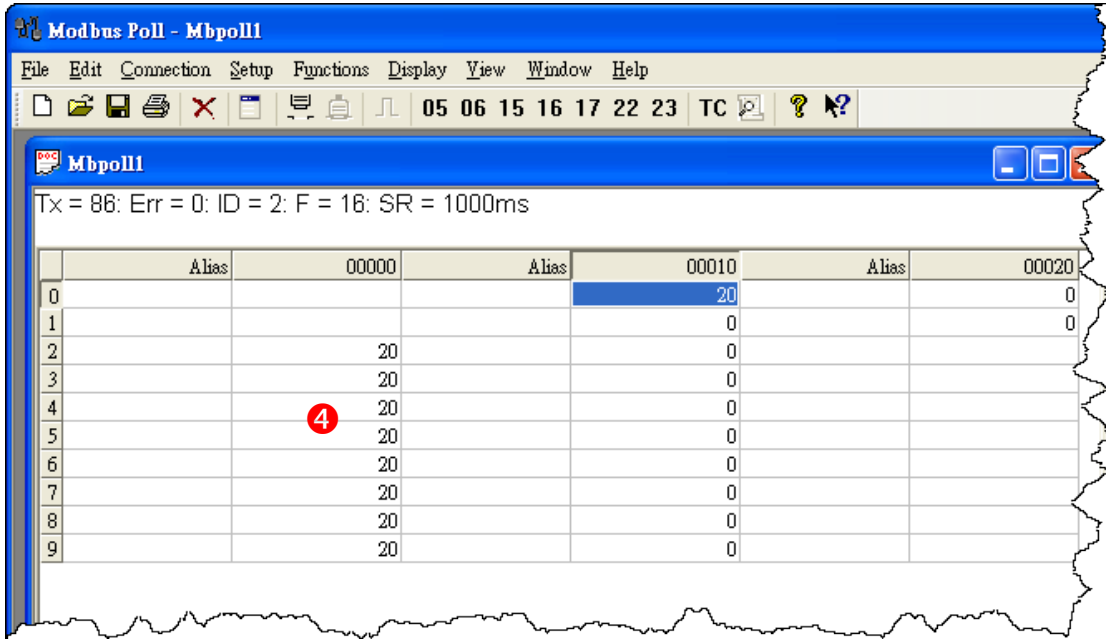


圖 5-3.7

- ⑤ 在 TwinCAT System Manager 左邊欄位的視窗，按下 TxPDO 0x00-0x7F。  
 ⑥ 然後在右側視窗中，確認 In02 到 In0A 的 Online 值是顯示 0x0014 (20)。

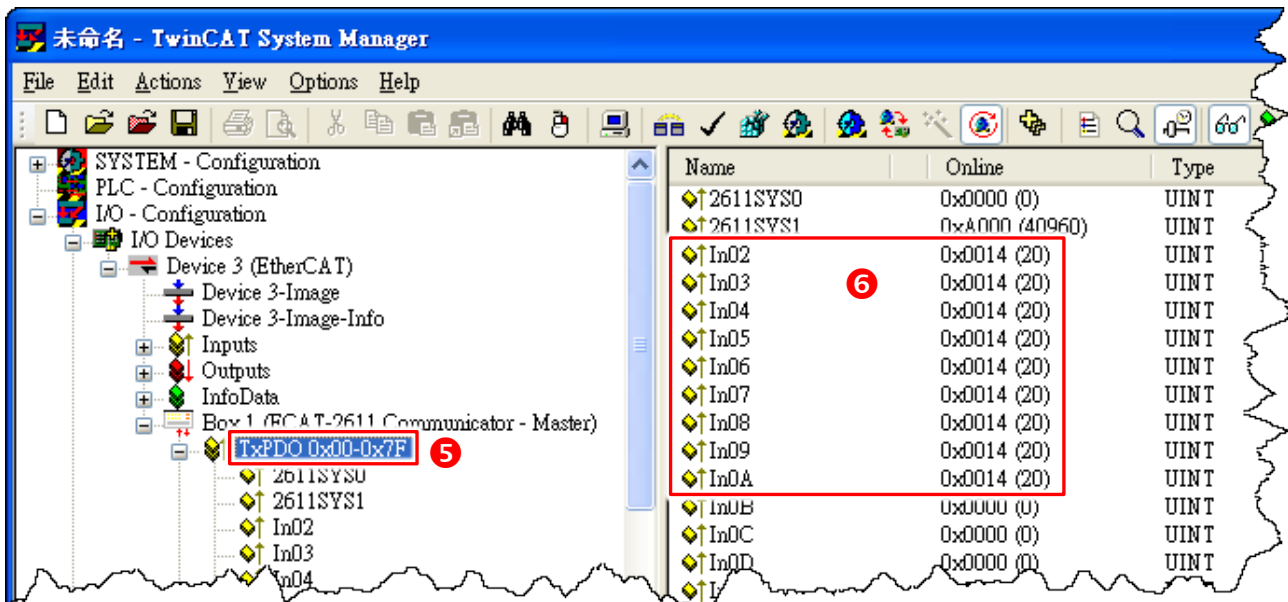


圖 5-3.8



## 6. Modbus 資訊

Modbus 是由 MODICON 公司在 1979 發展出來的一套通訊協定。它具有標準化、採開放式架構的特性，而且廣泛的被工業自動化產品所使用的通訊協定。透過 Modbus、SCADA 和 HMI 軟體可以很容易地將許多串列設備整合在一起。更多更詳細的 Modbus 資訊，可參考至 <http://www.modbus.org>。

現今 Modbus 協定版本有 Modbus RTU (如: RS-485/RS-232 序列通訊界面)、Modbus ASCII 以及 Modbus TCP。下面我們將詳細介紹 Modbus RTU 訊息結構。

### Modbus 訊資結構

Master 設備詢問訊息包括其它 Slave 設備的位址或廣播位址、功能代碼、任何所需資料以及檢查錯誤欄位。Slave 設備回應訊息包括確認功能代碼、回應資料及檢查錯誤欄位。

#### RTU Data 結構

Byte 00	Byte 01	Byte 02-03	Byte 04-05
站號 (Net ID)	功能代碼 (Function Code)	資料欄位	
		參考位址 (Address Mapping)	通道數 (Point)

- **站號(Net ID):** 指定接收地址 (Modbus/RTU slave)。
- **功能代碼 (Function Code):** 指定訊息類型。
- **資料欄位:** 資料區塊 (參考位址 + 通道數)。

## 站號 (Net ID)

在 Modbus RTU 結構中第一個 byte 是接收位址。有效的位址範圍是 0 到 247。當位址為 0 的時候，是為廣播功能，當位址為 1 到 247 的時候，分別是 Modbus 設備的 Net ID。

## 功能代碼 (Function Code)

Modbus RTU 結構中第二個 byte 是 Function Code (功能代碼)。Function Code 是要求 Slave 設備需執行的類型。有效的 Function Code 範圍是 1 到 255 之間。而 Slave 設備的回應訊息可設定相同的 Function Code，當發生錯誤時，系統將 Function Code 最高位元設定為 1，此時 Master 設備會知道該訊息是否已正確發送。

功能碼	功能敘述	參考位址
01 (0x01)	Read Multiple Coils Status for DO	0xxxx
02 (0x02)	Read Multiple Input Discrete for DI	1xxxx
03 (0x03)	Read Multiple Registers for AO	4xxxx
04 (0x04)	Read Multiple Input Registers for AI	3xxxx
05 (0x05)	Write Single Coil for DO	0xxxx
06 (0x06)	Write Single Register for AO	4xxxx
15 (0x0F)	Force Multiple Coils for DO	0xxxx
16 (0x10)	Write Multiple Registers for AO	4xxxx

**注意：**詳細關於對應位址(參考位址)請參考至您的 Slave 設備。

## 資料欄位

傳輸資料格式分別有 8 位元、16 位元及 32 位元。當資料為 16 位元暫存器傳輸是以 high-byte 優先 (例如: 0x0A0B ==> 0x0A, 0x0B)。當資料為 32 位元暫存器傳輸是二個 16 位元暫存器，且是以 Low-word 優先 (如: 0x0A0B0C0D ==> 0x0C, 0x0D, 0x0A, 0x0B)。

此資料欄位所傳送的訊息是 Master 設備及 Slave 設備之間的資訊，此資訊包含了 Master 設備採取的動作訊息或 Slave 設備任何請求資訊。如 Master 設備不需要這些資訊，此資料欄位可以為空白。

參考位址	說明
0xxxx	<b><u>Read/Write Discrete Outputs or Coils.</u></b> 0x 參考位址是用於設備輸出資料到數位輸出通道。
1xxxx	<b><u>Read Discrete Inputs.</u></b> 1x 參考位址是用於控制相對應的數位輸入通道的 ON/OFF 狀態。
3xxxx	<b><u>Read Input Registers.</u></b> 3x 參考暫存器包含一個 16-bit 位址接收外部訊息來源，如類比訊息。
4xxxx	<b><u>Read/Write Output or Holding Registers.</u></b> 4x 暫存器是用於儲存 16-bit 資料數 (二進制或十進制) 或從 CPU 傳送資料到輸出通道。

### 注意

詳細關於對應位址 (參考位址) 請參考至您的 Slave 設備。

## 6.1 FC1(0x01) Read Multiple Coils (0xxxx) for DO

這個功能代碼是用來讀取目前的 coil 狀態或 DO Readback 值。

### [Request]

Byte	說明	大小	設定值
00	站號 (Net ID)	1 Byte	1 ~ 247
01	功能代碼 (Function code)	1 Byte	0x01
02-03	D/O 起始位址	2 Bytes	詳細 Modbus 位址參考至您的 Slave 設備。 Byte 02 = high byte Byte 03 = low byte
04-05	點數 (通道數)	2 Bytes	Byte 04 = high byte Byte 05 = low byte

### [Response]

Byte	說明	大小	設定值
00	站號 (Net ID)	1 Byte	1 ~ 247
01	功能代碼 (Function code)	1 Byte	0x01
02	Byte 數	1 Byte	回應訊息的 Byte 數( $n = (\text{Points}+7)/8$ )
03	數據 (Data)	n Bytes	回應訊息 n= 1; Byte 03 = data bit 7 to 0 n= 2; Byte 04 = data bit 15 to 8 ..... n= m; Byte m+2 = data bit(8m-1)~8(m-1)

### [Error Response]

Byte	說明	大小	設定值
00	站號 (Net ID)	1 Byte	1 ~ 247
01	功能代碼 (Function code)	1 Byte	0x81
02	異常代碼 (Exception code)	1 Byte	更詳細資訊請參考至 Modbus 標準規範 (Modbus Standard Specification)

## 6.2 FC2(0x02) Read Multiple Input Discrete (1xxxx) for DI

這個功能代碼是用來讀取目前的 DI 值。

### [Request]

Byte	說明	大小	設定值
00	站號 (Net ID)	1 Byte	1 ~ 247
01	功能代碼 (Function code)	1 Byte	0x02
02-03	D/I 起始位址	2 Bytes	詳細 Modbus 位址參考至您的 Slave 設備。 Byte 02 = high byte Byte 03 = low byte
04-05	點數 (通道數)	2 Bytes	Byte 04 = high byte Byte 05 = low byte

### [Response]

Byte	說明	大小	設定值
00	站號 (Net ID)	1 Byte	1 ~ 247
01	功能代碼 (Function code)	1 Byte	0x02
02	Byte 數	1 Byte	回應訊息的 Byte 數 ( $n = (\text{Points} + 7) / 8$ )
03	數據 (Data)	n Bytes	回應訊息 n= 1; Byte 03 = data bit 7 to 0 n= 2; Byte 04 = data bit 15 to 8 ..... n= m; Byte m+2 = data bit(8m-1)~8(m-1)

### [Error Response]

Byte	說明	大小	設定值
00	站號 (Net ID)	1 Byte	1 ~ 247
01	功能代碼 (Function code)	1 Byte	0x82
02	異常代碼 (Exception code)	1 Byte	更詳細資訊請參考至 Modbus 標準規範 (Modbus Standard Specification)

## 6.3 FC3(0x03) Read Multiple Registers (4xxxx) for AO

這個功能代碼是用來 Readback 保存暫存器值或類比輸出值。

### [Request]

Byte	說明	大小	設定值
00	站號 (Net ID)	1 Byte	1 ~ 247
01	功能代碼 (Function code)	1 Byte	0x03
02-03	A/O 起始位址	2 Bytes	詳細 Modbus 位址參考至您的 Slave 設備。 Byte 02 = high byte Byte 03 = low byte
04-05	16-bit Registers 數 (通道數)	2 Bytes	Word 數 Byte 04 = high byte Byte 05 = low byte

### [Response]

Byte	說明	大小	設定值
00	站號 (Net ID)	1 Byte	1 ~ 247
01	功能代碼 (Function code)	1 Byte	0x03
02	Byte 數	1 Byte	回應訊息的 Byte 數 (n=Points x 2 Bytes)
03~	Register 值	n Bytes	Register 值: n= 2; Byte 03 = high byte Byte 04 = low byte ..... n= m; Byte 03 = high byte Byte 04 = low byte ..... Byte m+1 = high byte Byte m+2 = low byte

### [Error Response]

Byte	說明	大小	設定值
00	站號 (Net ID)	1 Byte	1 ~ 247
01	功能代碼 (Function code)	1 Byte	0x83
02	異常代碼 (Exception code)	1 Byte	更詳細資訊請參考至 Modbus 標準規範 (Modbus Standard Specification)

## 6.4 FC4(0x04) Read Multiple Input Registers (3xxxx) for AI

這個功能代碼是用來讀取輸入暫存器或電流類比輸入值。

### [Request]

Byte	說明	大小	設定值
00	站號 (Net ID)	1 Byte	1 ~ 247
01	功能代碼 (Function code)	1 Byte	0x04
02-03	AI 起始位址	2 Bytes	詳細 Modbus 位址參考至您的 Slave 設備。 Byte 02 = high byte Byte 03 = low byte
04-05	16-bit Registers 數 (通道數)	2 Bytes	Word 數 Byte 04 = high byte Byte 05 = low byte

### [Response]

Byte	說明	大小	設定值
00	站號 (Net ID)	1 Byte	1 ~ 247
01	功能代碼 (Function code)	1 Byte	0x04
02	Byte 數	1 Byte	回應訊息的 Byte 數 (n=Points x 2 Bytes)
03~	Resgister 值	n Bytes	Register 值: n= 2; Byte 03 = high byte Byte 04 = low byte ..... n= m; Byte 03 = high byte Byte 04 = low byte ..... Byte m+1 = high byte Byte m+2 = low byte

### [Error Response]

Byte	說明	大小	設定值
00	站號 (Net ID)	1 Byte	1 ~ 247
01	功能代碼 (Function code)	1 Byte	0x84
02	異常代碼 (Exception code)	1 Byte	更詳細資訊請參考至 Modbus 標準規範 (Modbus Standard Specification)

## 6.5 FC5(0x05) Write Single Coil (0xxxx) for DO

這個功能代碼是用來設定單一 coil 狀態或訊號數位輸出值。

### [Request]

Byte	說明	大小	設定值
00	站號 (Net ID)	1 Byte	1 ~ 247
01	功能代碼 (Function code)	1 Byte	0x05
02-03	D/O 起始位址	2 Bytes	詳細 Modbus 位址參考至您的 Slave 設備。 Byte 02 = high byte Byte 03 = low byte
04-05	輸出值	2 Bytes	0xFF 00 → 設定輸出為 ON 0x00 00 → 設定輸出為 OFF 如設定其它值將不被接受且不會影響到 coil。 Byte 04 = high byte Byte 05 = low byte

### [Response]

Byte	說明	大小	設定值
00	站號 (Net ID)	1 Byte	1 ~ 247
01	功能代碼 (Function code)	1 Byte	0x05
02-03	D/O 位址	2 Bytes	此值是與 Request 的 Byte 02-03 相同
04-05	輸出值	2 Bytes	此值是與 Request 的 Byte 04-05 相同

### [Error Response]

Byte	說明	大小	設定值
00	站號 (Net ID)	1 Byte	1 ~ 247
01	功能代碼 (Function code)	1 Byte	0x85
02	異常代碼 (Exception code)	1 Byte	更詳細資訊請參考至 Modbus 標準規範 (Modbus Standard Specification)



## 6.6 FC6(0x06) Write Single Register (4xxxx) for AO

這個功能代碼是用來設定一個 Holding Registers 並且能夠儲存該模組配置值。

### [Request]

Byte	說明	大小	設定值
00	站號 (Net ID)	1 Byte	1 ~ 247
01	功能代碼 (Function code)	1 Byte	0x06
02-03	A/O 位址	2 Bytes	詳細 Modbus 位址參考至您的 Slave 設備。 Byte 02 = high byte Byte 03 = low byte
04-05	Register 值	2 Bytes	Register 值 Byte 04 = high byte Byte 05 = low byte

### [Response]

Byte	說明	大小	設定值
00	站號 (Net ID)	1 Byte	1 ~ 247
01	功能代碼 (Function code)	1 Byte	0x06
02-03	A/O 位址	2 Bytes	此值是與 Request 的 Byte 02-03 相同
04-05	Register 值	2 Bytes	此值是與 Request 的 Byte 04-05 相同

### [Error Response]

Byte	說明	大小	設定值
00	站號 (Net ID)	1 Byte	1 ~ 247
01	功能代碼 (Function code)	1 Byte	0x86
02	異常代碼 (Exception code)	1 Byte	更詳細資訊請參考至 Modbus 標準規範 (Modbus Standard Specification)

## 6.7 FC15(0x0F) Force Multiple Coils (0xxxx) for DO

這個功能代碼是用來設定多個 coils 狀態或寫多個 DO 值。

### [Request]

Byte	說明	大小	設定值
00	站號 (Net ID)	1 Byte	1 ~ 247
01	功能代碼 (Function code)	1 Byte	0x0F
02-03	D/O 起始位址	2 Bytes	詳細 Modbus 位址參考至您的 Slave 設備。 Byte 02 = high byte Byte 03 = low byte
04-05	輸出通道數 (點)	2 Bytes	Byte 04 = high byte Byte 05 = low byte
06	Byte 數	1 Byte	$n = (\text{Points} + 7) / 8$
07	輸出值	n Bytes	一個bit 對應一個通道。 如: 值為 1 表示通道為 ON , 值為 0 表示為 OFF。 n= 1; Byte 07 = data bit 7 to 0 n= 2; Byte 08 = data bit 15 to 8 ..... n= m; Byte m+6 = data bit(8m-1)~8(m-1)

### [Response]

Byte	說明	大小	設定值
00	站號 (Net ID)	1 Byte	1 ~ 247
01	功能代碼 (Function code)	1 Byte	0x0F
02-03	D/O 起始位址	2 Bytes	此值是與 Request 的 Byte 02-03 相同
04-05	輸出通道數 (點數)	2 Bytes	此值是與 Request 的 Byte 04-05 相同

### [Error Response]

Byte	說明	大小	設定值
00	站號 (Net ID)	1 Byte	1~ 247
01	功能代碼 (Function code)	1 Byte	0x8F
02	異常代碼 (Exception code)	1 Byte	更詳細資訊請參考至 Modbus 標準規範 (Modbus Standard Specification)

## 6.8 FC16(0x10) Write Multiple Registers (4xxxx) For AO

這個功能代碼是用來設定多個 Holding Registers 並且能夠儲存模組配置值。

### [Request]

Byte	說明	大小	設定值
00	站號 (Net ID)	1 Byte	1 ~ 247
01	功能代碼 (Function code)	1 Byte	0x10
02-03	A/O 起始位址	2 Bytes	詳細 Modbus 位址參考至您的 Slave 設備。 Byte 02 = high byte Byte 03 = low byte
04-05	16-bit Register 數 (通道數)	2 Bytes	Word 數 Byte 04 = high byte Byte 05 = low byte
06	Byte 數	1 Byte	n = Points x 2 Bytes
07	Register 值	n Bytes	Register 值 n = 2; Byte 03 = high byte Byte 04 = low byte ..... n = m; Byte 03 = high byte Byte 04 = low byte ..... Byte m+1 = high byte Byte m+2 = low byte

### [Response]

Byte	說明	大小	設定值
00	站號 (Net ID)	1 Byte	1 ~ 247
01	功能代碼 (Function code)	1 Byte	0x10
02-03	A/O 起始位址	2 Bytes	此值是與 Request 的 Byte 02-03 相同
04-05	16-bit Register 數 (通道數)	2 Bytes	此值是與 Request 的 Byte 04-05 相同

### [Error Response]

Byte	說明	大小	設定值
00	站號 (Net ID)	1 Byte	1 ~ 247
01	功能代碼 (Function code)	1 Byte	0x90
02	異常代碼 (Exception code)	1 Byte	更詳細資訊請參考至 Modbus 標準規範 (Modbus Standard Specification)

## 6.9 FC255(0xFF) 特殊命令

此功能碼是僅適用於 ECAT-2610 模組的特殊命令。

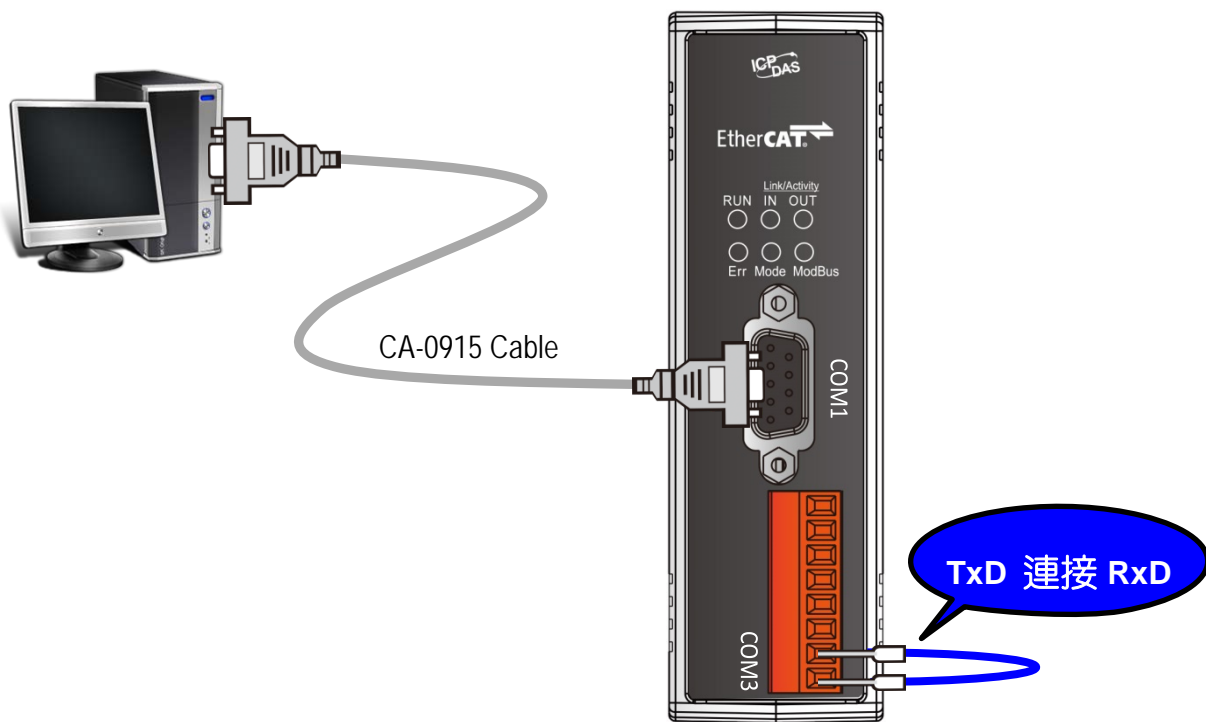
命令	說明	備註
<b>FF 03 00 00 00 02</b>	讀取模組系統狀態 (2610Sys0 + 2610Sys1)	詳細說明，參考 <a href="#">章節 3.3.1 “模組狀態及錯誤模式”</a>
<b>FF 03 00 01 00 01</b>	儲存 RS-485 週期時間 單位: 0.1 ms	詳細說明，參考 <a href="#">16. Rs485 Cycle Time</a> 。
<b>FF 06 00 00 00 nn</b>	Delay 100 ms x nn	詳細說明，參考 <a href="#">11. Delay Command</a> 。
<b>FF 06 00 01 00 nn</b>	Delay 1 ms x nn	

## 7. 上傳配置檔操作

若發現 ECAT-2610(-DW)/2611 模組無法正常運作時 (如: 模組沒有回應, LED 顯示燈沒亮...等問題), 請將模組 COM3 上的 TxD pin 連接至 RxD pin (如圖 7-1.1) 來進入 Init 模式 (偵錯模式)。當進入 Init 模式 (偵錯模式) 後, 將會跳過 EEPROM 並停止命令執行, 此時將 EEPROM 清除後再載入新的配置檔 (commands.txt) 使模組恢復正常, 詳細操作步驟如下。

### 步驟 1 將 TxD pin 連接至 RxD pin

- ❶ ECAT-2610(-DW)/2611 模組斷電關機。
- ❷ 使用 CA-0915 Cable 將 ECAT-2610(-DW)/2611 模組的 COM1 (Console Port) 連接電腦主機的 COM Port。
- ❸ 再將模組 COM3 上的 TxD pin 連接至 RxD pin。



## 步驟 2 執行 7188ECAT.exe(配置/偵錯工具)

❶ 這裡以 Windows 10 系列為範例，在 Windows 搜尋欄位輸入 “cmd”，按下 “Enter” 鍵來開啟命令提示字元視窗。

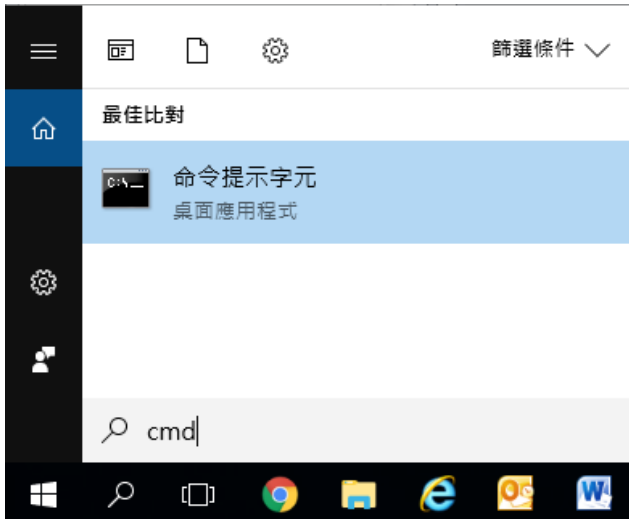


圖 7-1.2

### ⚠ 注意

開啟命令提示字元視窗的方法，請依據您使用的作業系統來執行

- ❷ 輸入 **E:** (此為 7188ECAT 資料夾所存放的磁碟槽區)，按 **Enter** 鍵。
- ❸ 輸入 **cd 7188ecat**，按 **Enter** 鍵來進入 7188ECAT 資料夾中。
- ❹ 輸入 **execcom4**，按 **Enter** 鍵來自動執行 **7188ECAT.exe** 程式。

**注意:** execcom4 是使用電腦 COM4 來上傳配置檔到 ECAT-2610(-DW)/2611，詳細參考圖 A4-17。

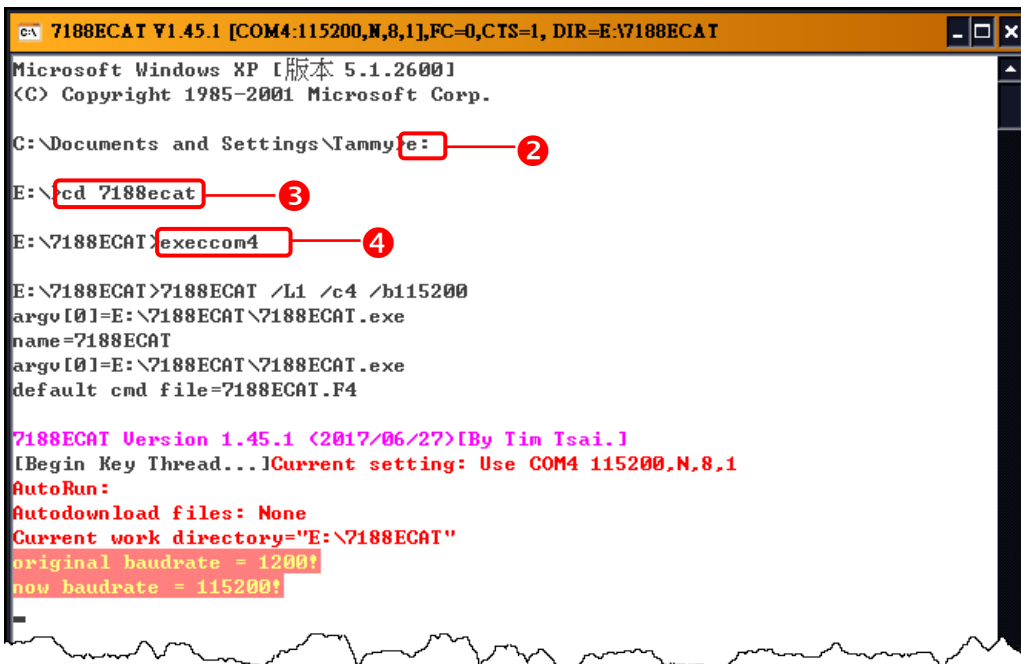


圖 7-1.3

### 步驟 3 將 ECAT-2610(-DW)/2611 模組供電開機來進入 Init 模式

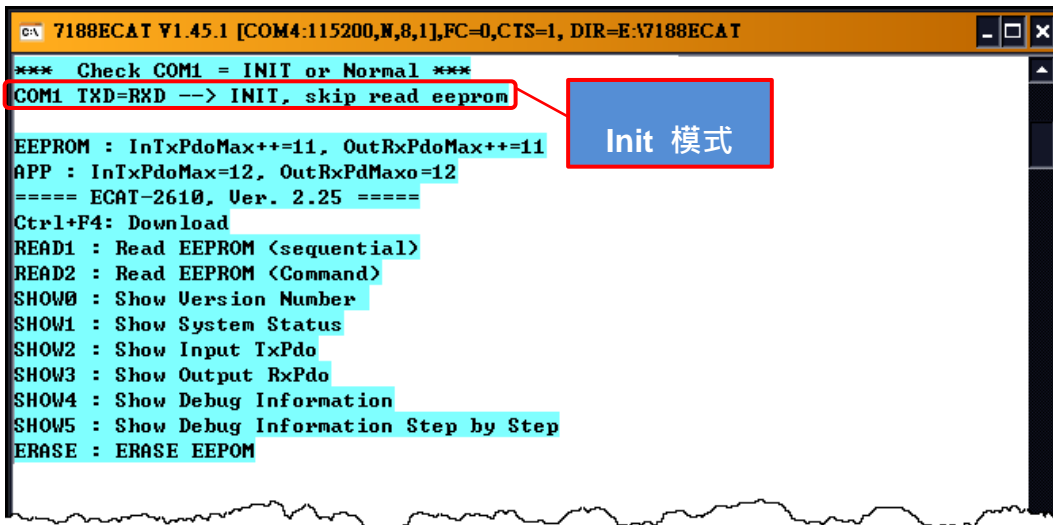


圖 7-1.4

➤ 將列出 10 個指令用來載入配置數據及原廠偵錯，詳細說明如下：

指令	說明	載入/偵錯
<b>CTRL+F4</b>	載入配置檔 (commands.txt) 到 EEPROM	<b>載入</b>
READ1	讀取 EEPROM	原廠偵錯
READ2	讀取 EEPROM	原廠偵錯
SHOW0	顯示版本號	原廠偵錯
SHOW1	顯示狀態	原廠偵錯
SHOW2	顯示輸入 InTxPDO[00] 到 InTxPDO[FF]	原廠偵錯
SHOW3	顯示輸出 OutRxPDO[00] 到 OutRxPDO[FF]	原廠偵錯
SHOW4	顯示偵錯資訊	原廠偵錯
SHOW5	慢速顯示偵錯資訊	原廠偵錯
<b>ERASE</b>	清除 EEPROM	<b>載入</b>

以上指令中 **CTRL+F4** 及 **ERASE** 是開放用於載入新的配置檔到 ECAT-2610(-DW)/2611 模組 EEPROM 中。其它指令則是用於我司工廠偵錯使用。

#### 注意

EEPROM 用於存儲不經常更改的數據，不適合頻繁的存取大量數據，且清除/寫入週期是有限制的，因此在測試使用時不應經常更改，否則容易造成模組損壞。

**步驟 4** 將模組 COM3 上 TxD Pin 和 RxD Pin 斷開

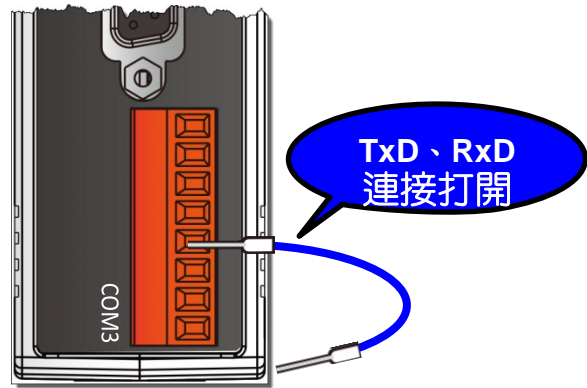


圖 7-1.5

**步驟 5** 上傳新的配置檔到 ECAT-2610(-DW)/2611 模組 EEPROM 中

❶ 輸入 **erase** ，按 **Enter** 鍵來清除 EEPROM 。



圖 7-1.6

❷ 將 ECAT-2610(-DW)/2611 斷電再上電來重啟模組 。

❸ 同時按下鍵盤上 **[Ctrl] + [F4]** 鍵來將新的配置檔上傳到模組中 。



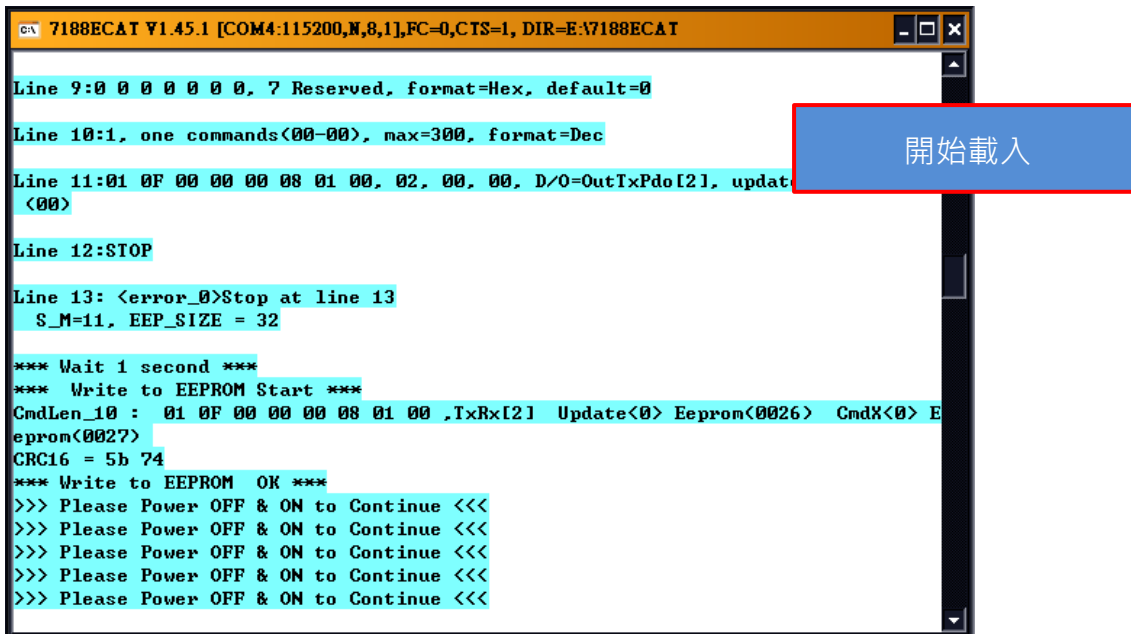


圖 7-1.7


- ④ 再將 ECAT-2610(-DW)/2611 斷電再上電來重啟模組。
- ⑤ 按下視窗右上方的  圖示來關閉視窗。



圖 7-1.8

# 8. 支援分散式時鐘(適用 ECAT-2610)

EtherCAT 提供了分散式時鐘機制 (DC · Distributed Clocks) · 透過 IEEE1588[13]標準的分散式時脈計算法實現同步，可以精確的同步主站及所有從站間的時間。更多更詳細關於 EtherCAT 及分散式時鐘資訊，可參考至 <http://www.ethercat.org/> 網站。

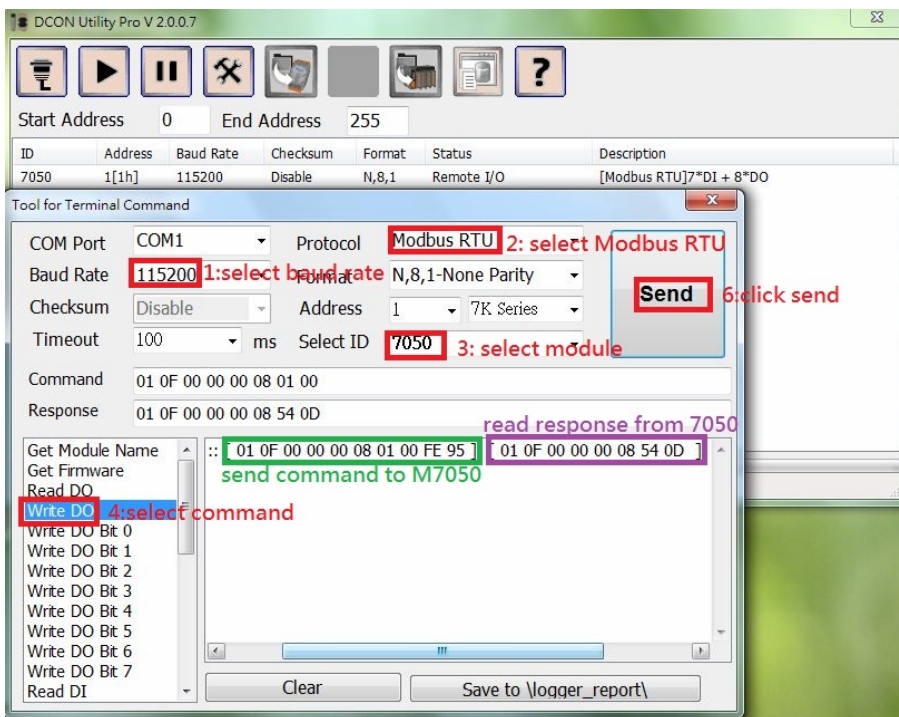
此章節將介紹如何透過 TwinCAT 3.0 軟體來配置及運作分散式時鐘功能。

## 8.1 Modbus RTU 時間量測

這裡我們使用 M-7050 模組為範例，它具有 7 通道的數位輸入及 8 通道的數位輸出。本節將介紹四個範例為量測傳送命令、讀取回應及 DC 週期的時間，詳細如下所述。

### 範例一：量測單一 Modbus 命令: Write DO 時間

❶ 使用 DCON Utility 輕鬆快速取得 Write DO 命令，詳細操作說明可參考 [A1. “如何透過 DCON Utility 來取得 Modbus RTU 指令”](#)，如下圖所示：



傳送 Wirte DO 命令字串:  
01 0F 00 00 00 08 01 00 **FE 95**

讀取回應字串:  
01 0F 00 00 00 08 54 0D

**注意: FE 95 及 54 0D 是 checksum bytes 檢查碼。**

圖 8-1.1

❷ 將 Wirte DO 配置檔 (如圖 8-1.2) 上傳到 ECAT-2610 模組中，詳細操作參考 [章節 4.2 “配置及上傳”](#) 或 [A4. “手動配置及上傳”](#)。

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
1, one commands(00-00), max=300, format=Dec
01 0F 00 00 00 08 01 00, 02, 00, 00, D/O=OutTxPdo[2], update cyclically, (00)
STOP
```

---

```
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
```

圖 8-1.2

❸ ECAT-2610 模組先傳送 Wirte DO 命令: **01 0F 00 00 00 08 01 00 FE 95** 到 M-7050 模組，然後再從 M-7050 模組讀取回應: **01 0F 00 00 00 08 54 0D**。此傳送並讀取 (send\_then\_read) 的動作將循環的重復進行。

❹ 下面將實際量測 M-7050 模組硬體的傳送命令及讀取回應時間:

- 傳送 Wirte DO 命令: **01 0F 00 00 00 08 01 00 FE 95** 時間約 1 ms。
- 讀取回應: **01 0F 00 00 00 08 54** 時間約 1 ms。
- 重複傳送並讀取 (send\_then\_read) 的時間約 5 ms。
- 第 N 個讀取 (read\_N) 到第 N+1 個傳送 (send\_N+1) 時間約 2 ms。
- 所以一個傳送+讀取的週期 (send\_read\_cycle) 時間約 5+2=7 ms。

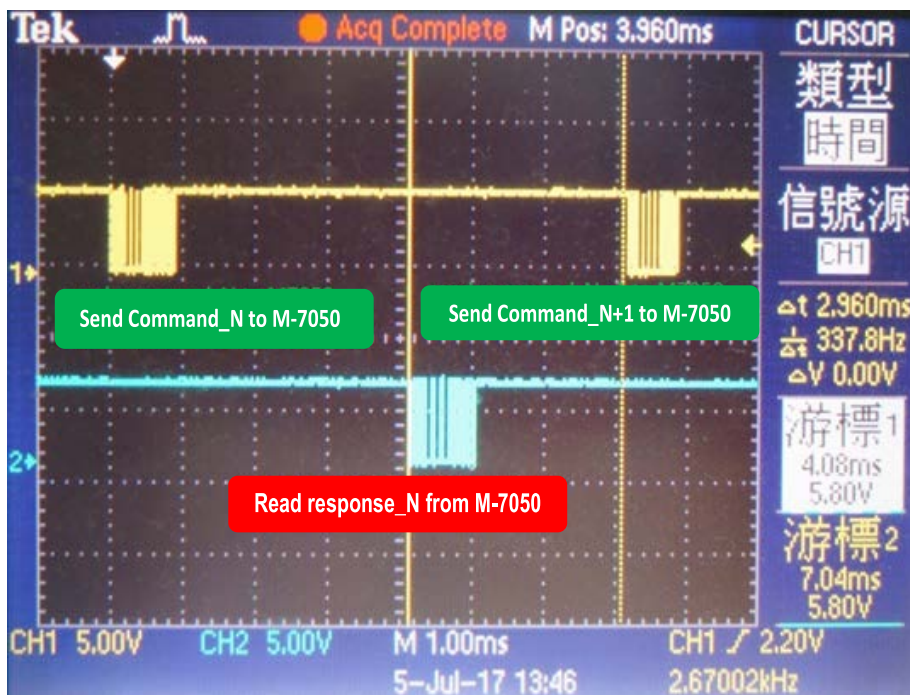
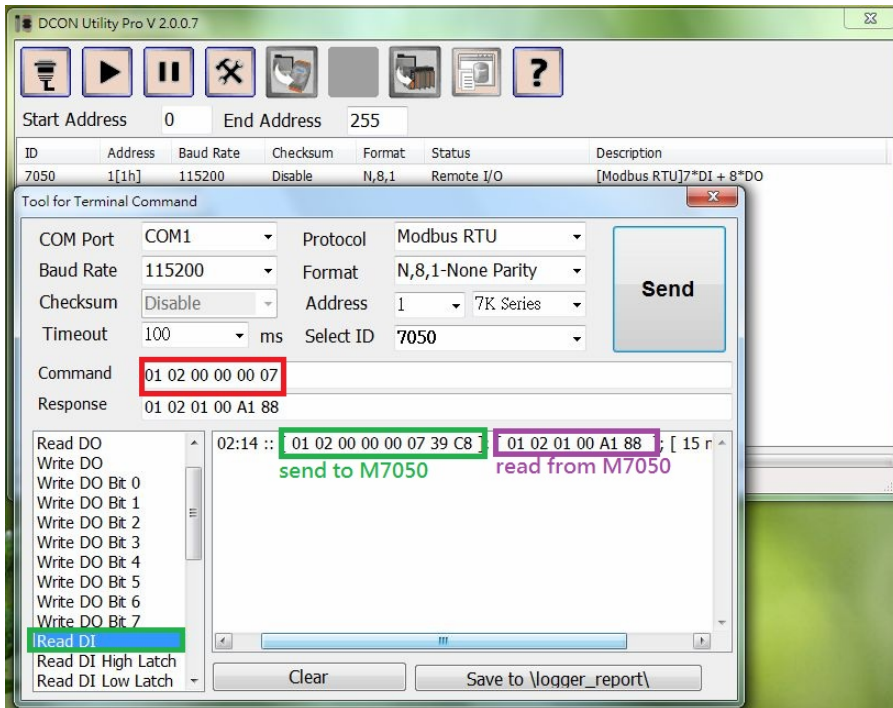


圖 8-1.3

## 範例二：量測單一 Modbus 命令:Read DI 時間

❶ 使用 DCON Utility 輕鬆快速取得 Read DI 命令，詳細操作說明可參考 [A1. “如何透過 DCON Utility 來取得 Modbus RTU 指令”](#)，如下圖所示：



傳送 Read DI 命令字串:  
01 02 00 00 00 07 **39 C8**

接收回應字串:  
01 02 01 00 **A1 88**

**注意: 39 C8 及 A1 88 是 checksum bytes 檢查碼。**

圖 8-1.4

❷ 將 Read DI 配置檔 (如圖 8-1.5) 上傳到 ECAT-2610 模組中，詳細操作參考 [章節 4.2 “配置及上傳”](#) 或 [A4. “手動配置及上傳”](#)。

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
1, one commands(00-00), max=300, format=Dec
01 02 00 00 00 07, 02, 00, 00, InTxPdo[02]=D/I, update cyclically, (00)
STOP

-----
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
```

圖 8-1.5

③ ECAT-2610 模組將先傳送 Read DI 命令: **01 02 00 00 00 07 39 C8** 到 M-7050 模組，然後再從 M-7050 模組讀取回應: **01 02 01 00 A1 88**。此傳送並讀取 (send\_then\_read) 的動作將循環的重復進行。

④ 下面將實際量測 M-7050 模組硬體的傳送命令及讀取回應時間:

- 傳送 Read DI 命令: **01 02 00 00 00 07 39 C8** 時間約 0.8 ms。
- 讀取回應: **01 02 01 00 A1 88** 時間約 0.7 ms。
- 重復傳送並讀取 (send\_then\_read) 的時間約 4 ms。
- 第 N 個讀取 (read\_N) 到第 N+1 個傳送 (send\_N+1) 時間約 2 ms。
- 所以一個傳送+讀取的週期 (send\_read\_cycle) 時間約  $4+2 = 6$  ms。

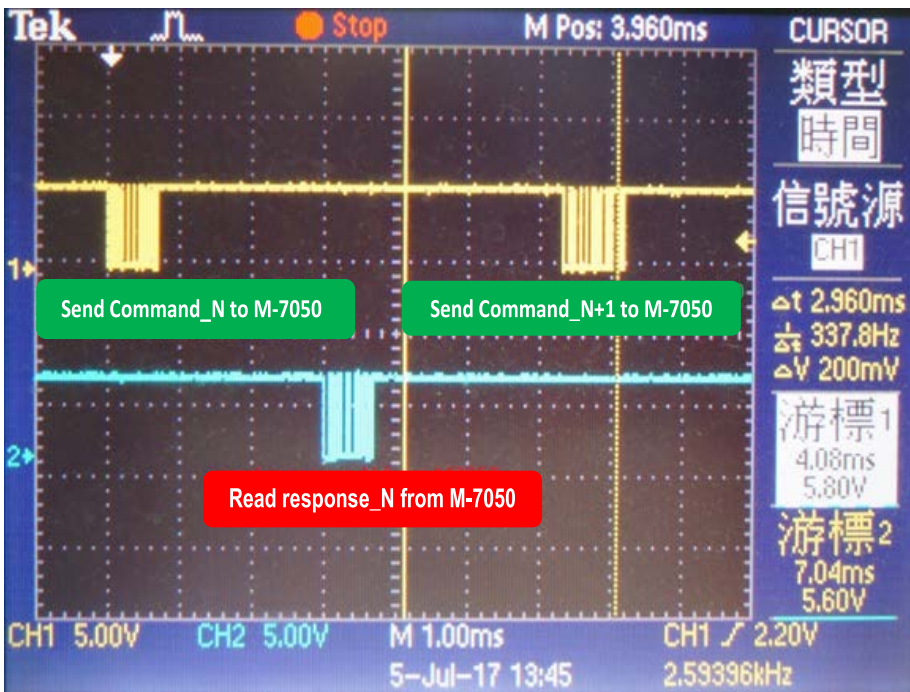


圖 8-1.6

### 範例三: 量測 Modbus 命令: Write DO + Read DI 時間

- ❶ 將 Wirte DO+Read DI 配置檔 (如圖 8-1.7) 上傳到 ECAT-2610 模組中·詳細操作參考 [章節 4.2 “配置及上傳”](#) 或 [A4. “手動配置及上傳”](#)。

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
2, two commands(00-01), max=300, format=Dec
01 0F 00 00 00 08 01 00, 02, 00, 00, D/I=OutTxPdo[2], update cyclically, (00)
01 02 00 00 00 07, 02, 00, 00, InTxPdo[02]=D/I, update cyclically, (01)
STOP
```

---

```
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
```

圖 8-1.7

- ❷ ECAT-2610 模組先傳送第一組 Wirte DO 命令: **01 0F 00 00 00 08 01 00 FE 95** 到 M-7050 模組·再從 M-7050 模組讀取回應: **01 0F 00 00 00 08 54 0D**。第二次再傳送 Read DI 命令: **01 02 00 00 00 07 39 C8** 到 M-7050 模組·再從 M-7050 模組讀取回應: **01 02 01 00 A1 88**。

第一組 Write DO Command\_01: (傳送) 01 0F 00 00 00 08 01 00 FE 95 + (讀取) 01 0F 00 00 00 08 54 0D  
 第二組 Read DI Command\_02: (傳送) 01 02 00 00 00 07 39 C8 + (讀取) 01 02 01 00 A1 88

- ❸ 下面將實際量測 M-7050 模組硬體的 Write DO + Read DI 命令時間:
  - 下圖可以看到一次的 Write DO + Read DI 命令(Command\_1 + Command\_02) 時間約 13.2 ms。

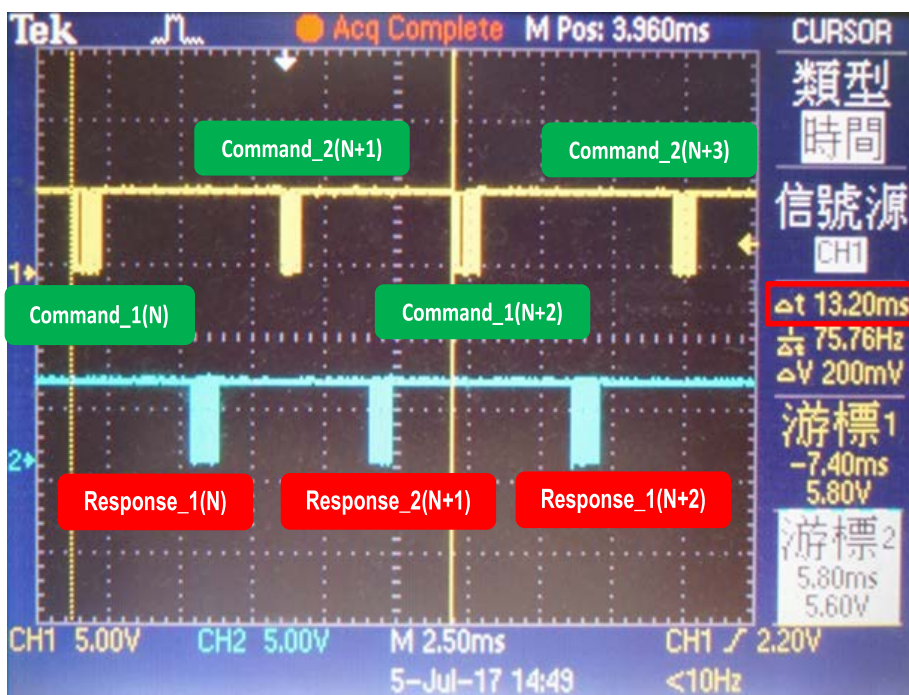


圖 8-1.8

### 範例三: 量測 DC 週期時間

ECAT-2610 模組會自動檢查並同步第一個命令到 DC Signal。在上面測量 Write DO 命令(圖 8-1.3) 得到週期時間約 7 ms，測量 Write DO + Read DI 命令(圖 8-1.8) 得到週期時間約 13.2 ms。

➤ 如果 DC 設定為 20 ms，其測量 Write DO 命令的週期時間 (cycle time) 如下圖所示:

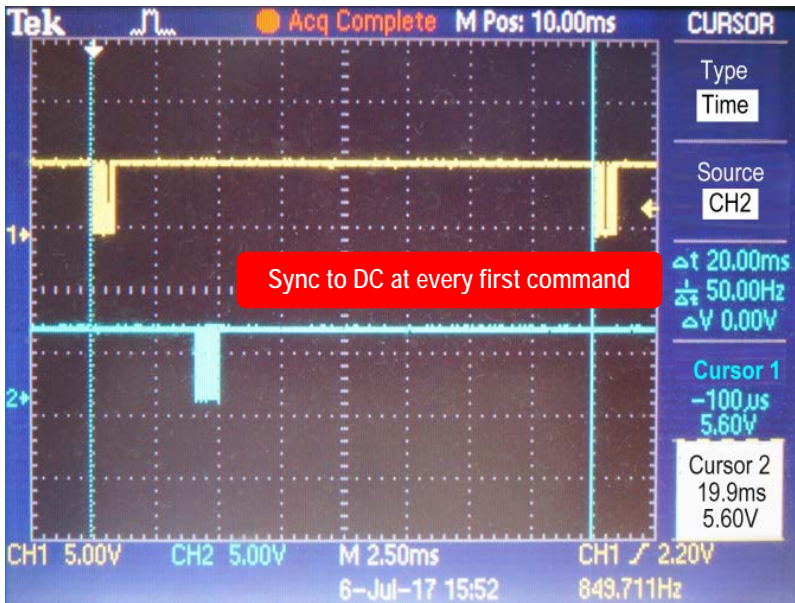


圖 8-1.9

➤ 如果 DC 設定為 20 ms，其測量 Write DO + Read DI 命令的週期時間 (cycle time) 如下圖所示:

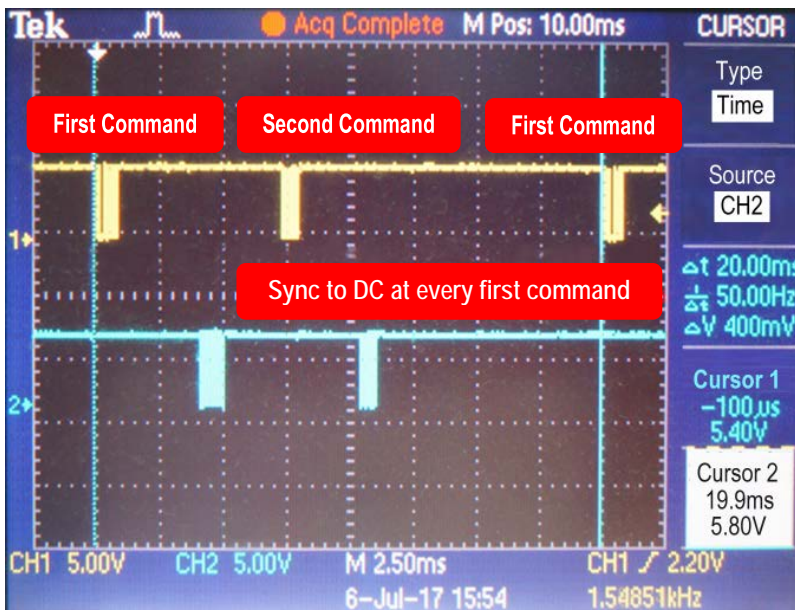


圖 8-1.10

一次的 Write DO + Read DI 命令 (Command\_1 + Command\_02) 時間約 13.2 ms (如圖 8.1-8)。

如果 DC 設定在 10 ms 小於 13.2 ms，ECAT-2610 模組自動檢查並同步第一個命令到 DC Signal。所以 DC = 10 ms 的時間量測圖與 DC = 20 ms 相同。總命令的週期時間可以大於 DC 週期時間是沒有問題的。

## 8.2 分散式時鐘(DC)配置與操作

下圖為分散式時鐘 (DC) 測試架構範例:

- DC-Synchron 功能啟用 (Active)
- DI = ECAT-2052 模組
- DO1 = M-7055 第一台模組 · DO2 = M-7055 第二台模組
- DO1 = DI · DO2 = DI

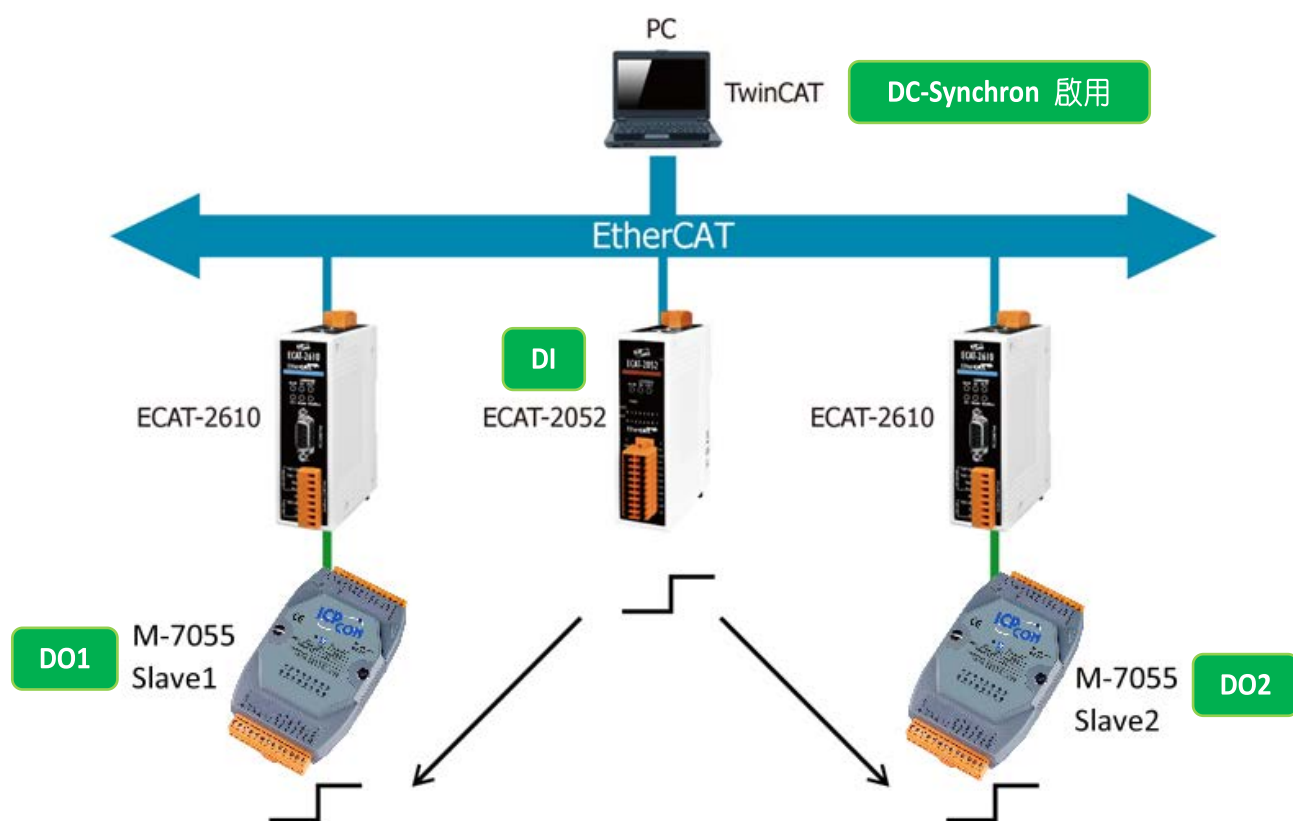
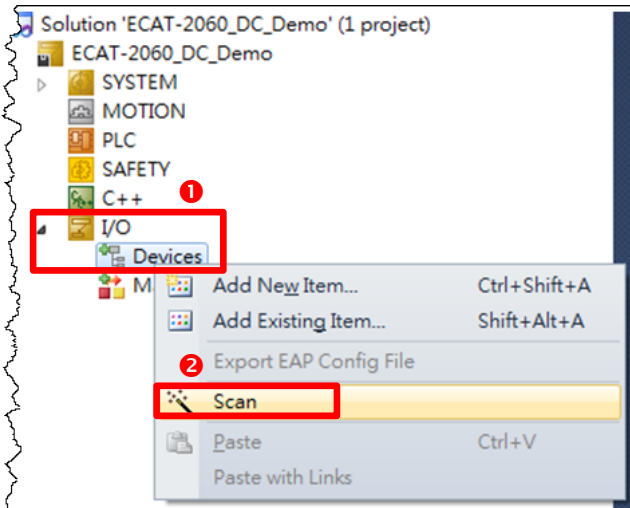


圖 8-2.1: 分散式時鐘(DC)測試架構



下面將透過 TwinCAT 3.0 軟體來設定啟用 DC-Synchron 功能，詳細步驟如下：

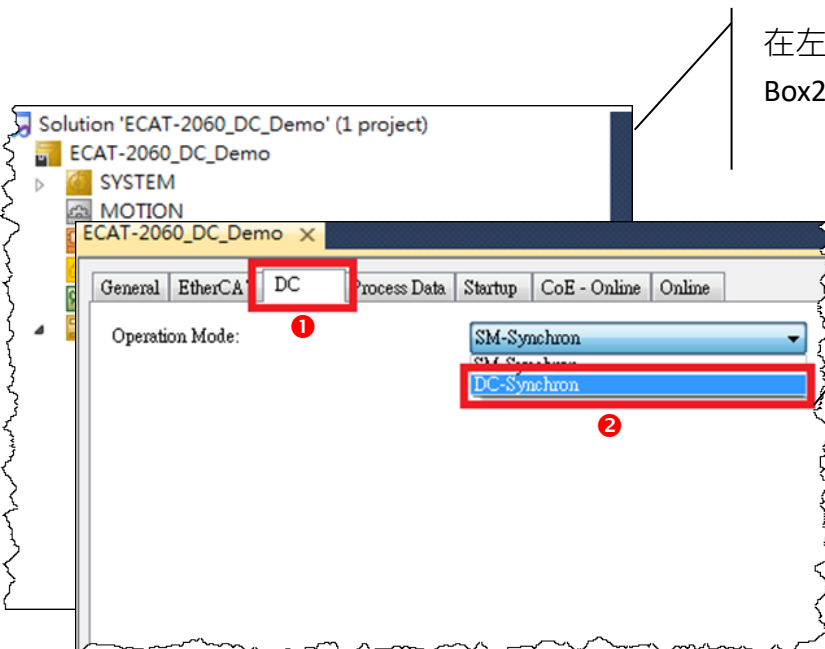
### 步驟 1 掃描設備



I/O -> Devices->按右鍵-> Scan

圖 8-2.2

### 步驟 2 ECAT-2610 設定 DC 模式

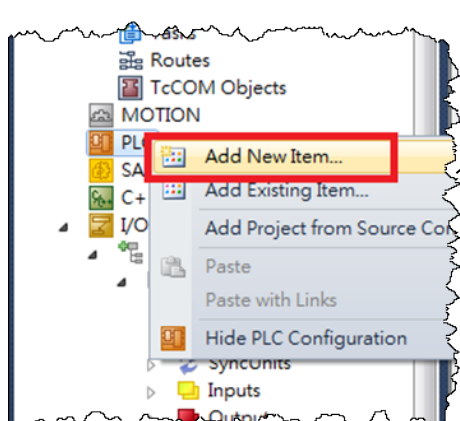


在左側視窗，單擊 EtherCAT Box1 及 Box2 [ECAT-2610]

在右側視窗，點選 DC 項目  
在 Operation Mode 下拉式  
選單，選擇 DC-Synchron

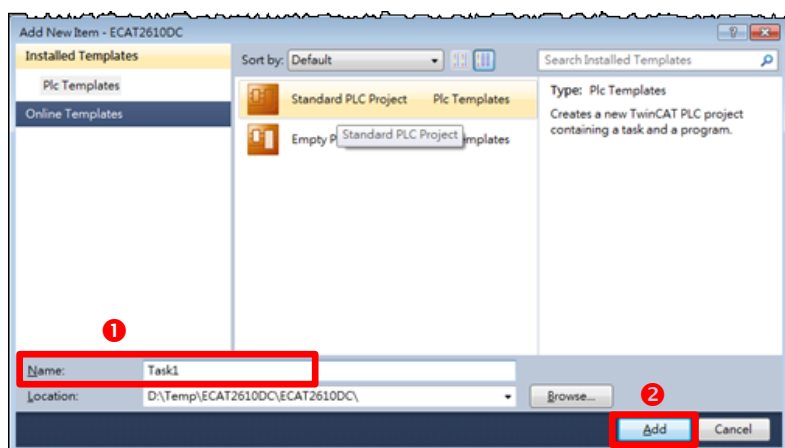
圖 8-2.3

### 步驟 3 建立啟用 PLC 專案 (Activate PLC)



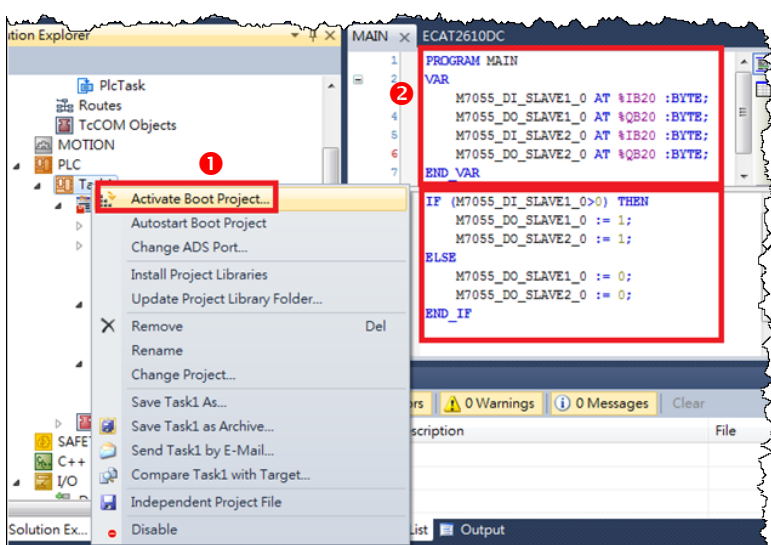
在左側視窗  
PCL ->按右鍵-> Add New Item...

圖 8-2.4



在 Name 欄位輸入專案名稱  
(如:Task1)

圖 8-2.5



在左側視窗  
Task1 ->按右鍵-> Activate Boot Project...

在右側視窗，參閱 MAIN 項目下的訊息，確認建立成功並開始寫入程式碼

圖 8-2.6

### 步驟 4 映射 M-7055 輸出點數至 ECAT-2610

雙擊 MAIN.M7055\_DO\_SLAVE1\_0  
從 Box1 [ECAT-2610] 點擇 Out000  
按下 OK

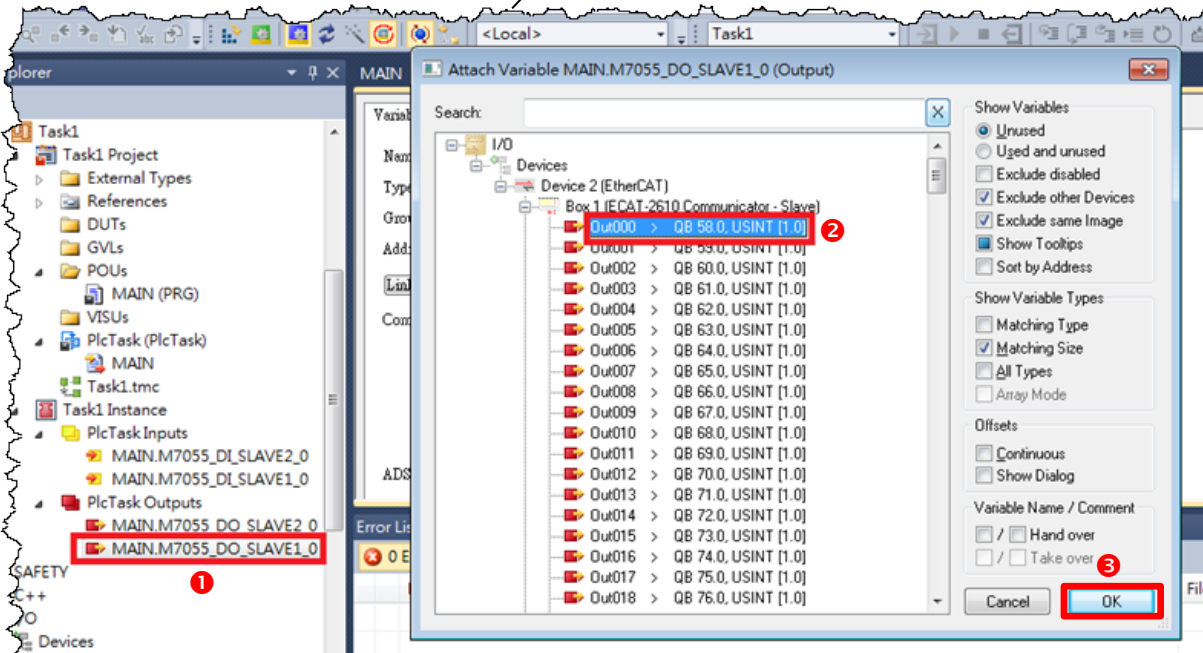
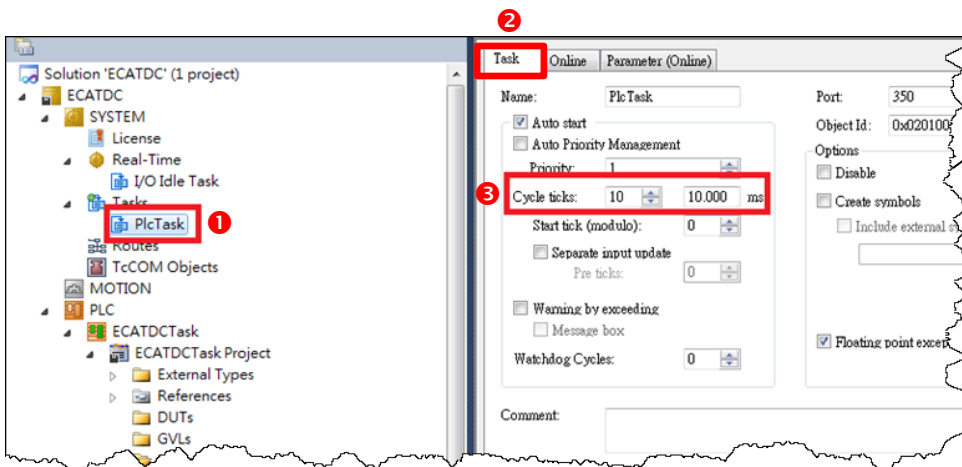


圖 8-2.7

### 步驟 5 週期時間 (Cycle Time) 設定



在左側視窗，按下  
**PlcTask**

在右側視窗，按下 **Task**  
設定 **cycle time = 10 ms**

圖 8-2.8

### 步驟 6 執行 PLC

按下 **Activate Configuration** 符號

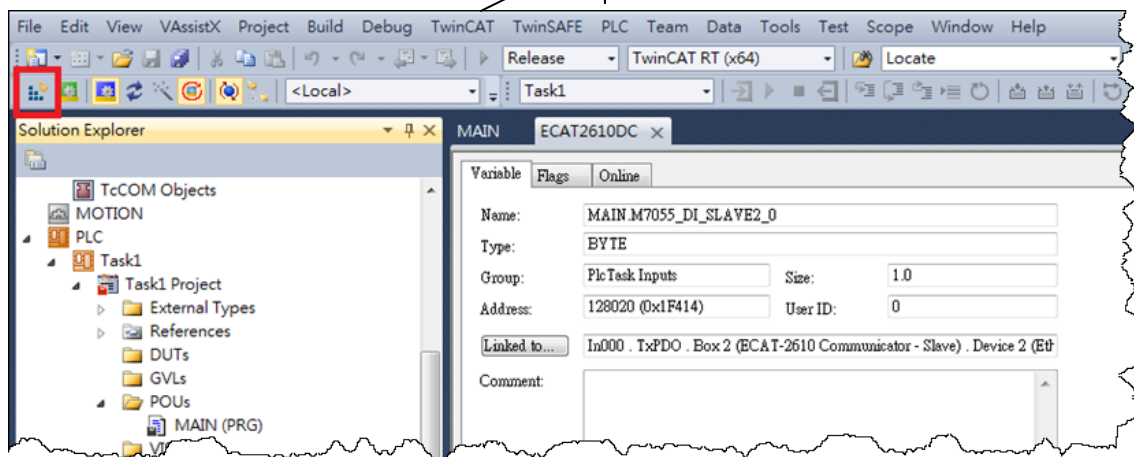
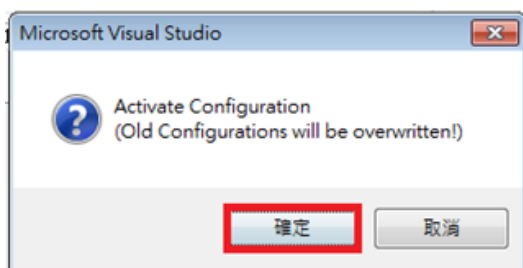
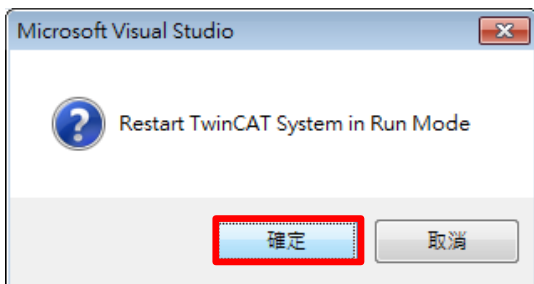


圖 8-2.9



按下 **確定** 按鈕

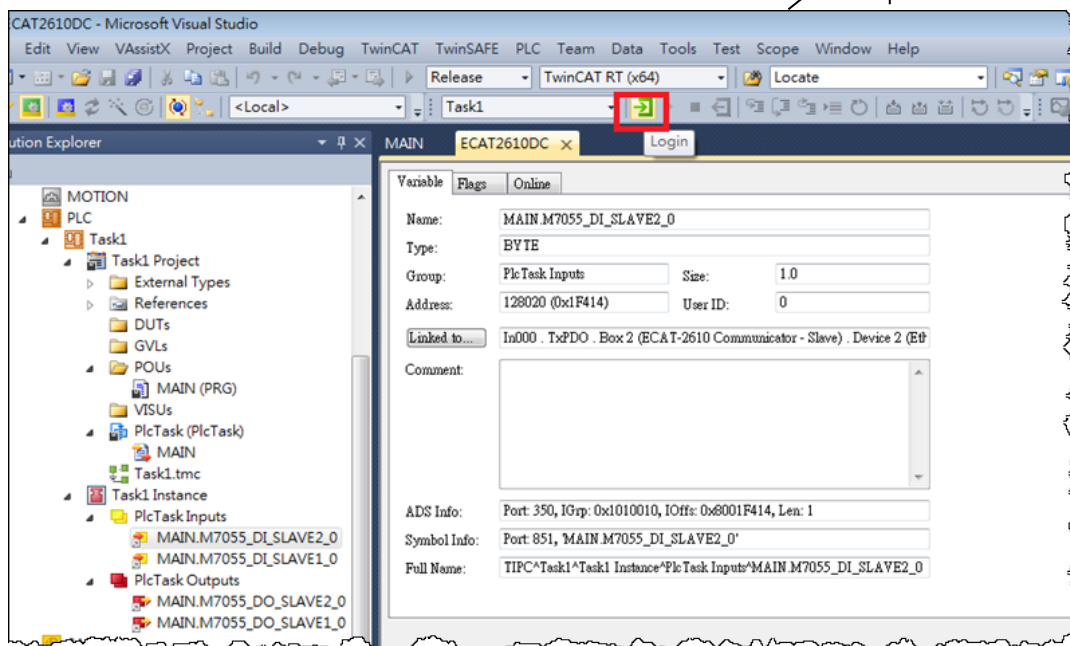
圖 8-2.10



按下 **確定** 按鈕

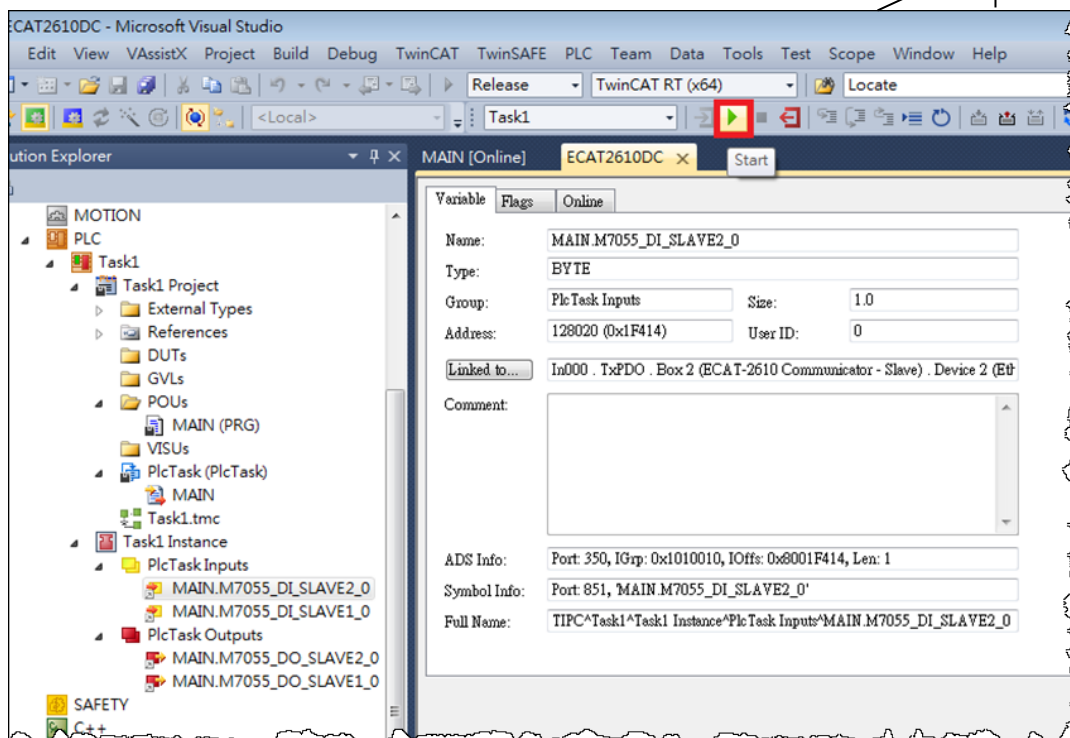
圖 8-2.11

按下 **Login** 符號



8-2.12

按下 **Start** 符號



8-2.13

如果 **DC-Synchron** 功能關閉，DO1 及 DO2 非同步是自己獨立，此時量測到 DO1 到 DO2 的時間為毫秒 (ms)，如下：

- DO1 到 DO2 時間約 2ms

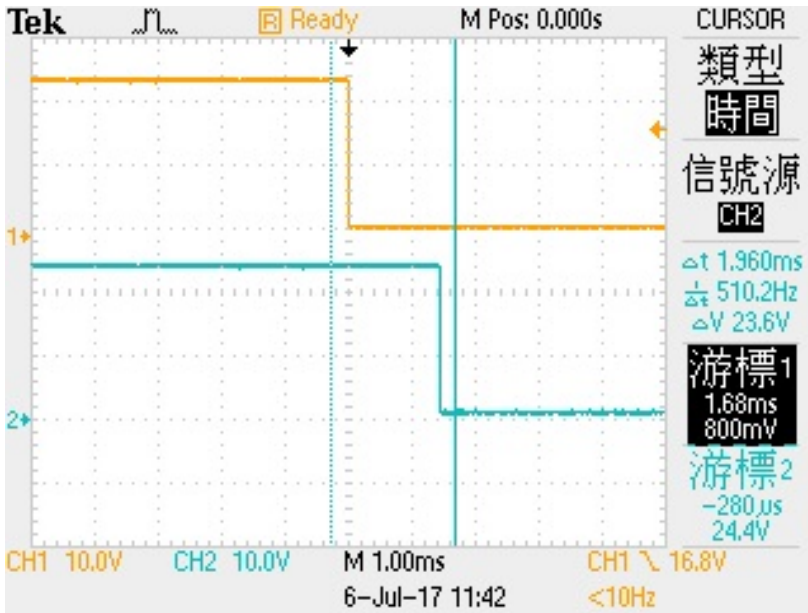


圖 8-2.14

- DO1 到 DO2 時間約 5ms

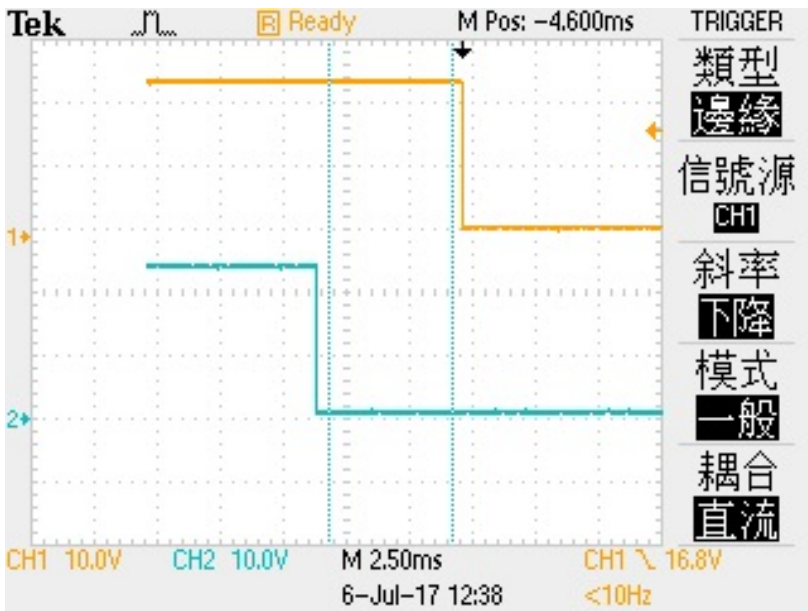


圖 8-2.15

- 量測 DO1 到 DO2 最差時間約 7 ms，請參考[章節 8.1 “Modbus RTU 時間量測”](#)。

如果 **DC-Synchron** 功能啟用，DO1 及 DO2 將同步到 DC，此時量測到 DO1 到 DO2 的時間為微秒( $\mu\text{s}$ )

➤ DO1 到 DO2 時間約  $6\ \mu\text{s}$

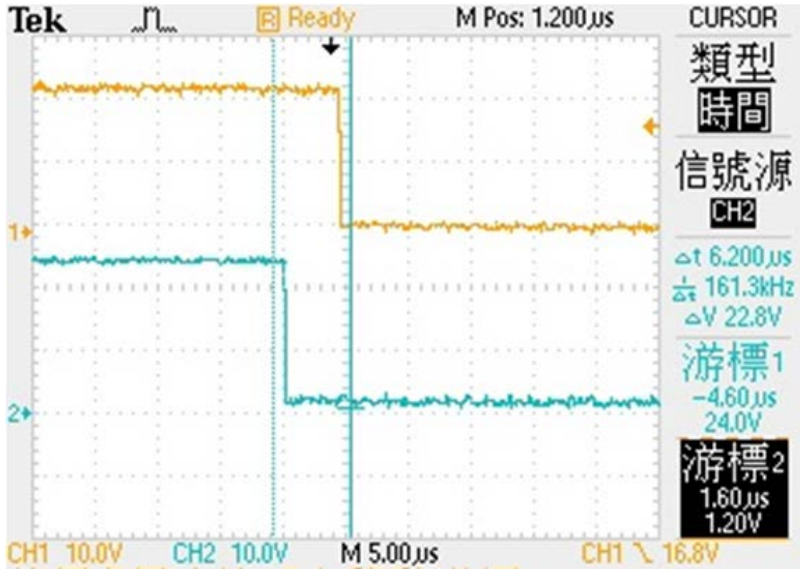


圖 8-2.16

➤ DO1 到 DO2 時間約  $100\ \mu\text{s}$

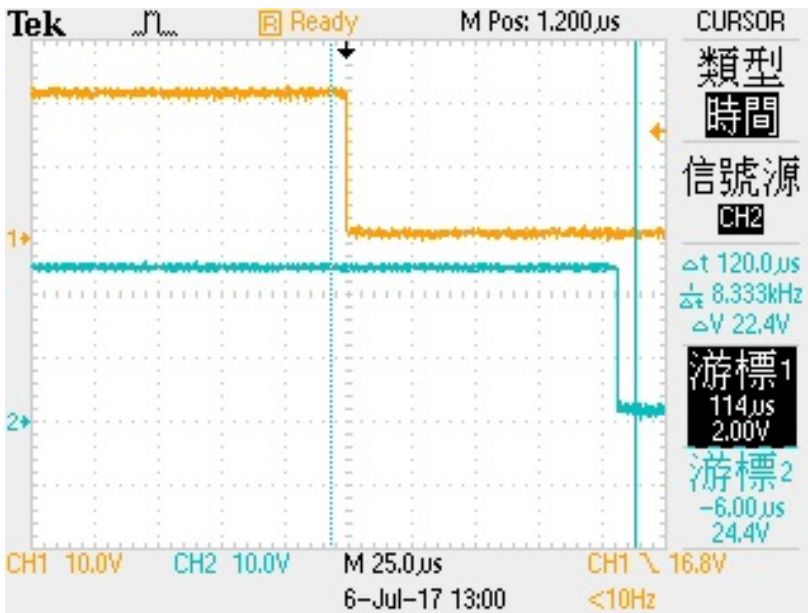


圖 8-2.17

➤ 在分散式時鐘功能，DO1 到 DO2 時間差異約幾微秒( $\mu\text{s}$ )。

# 9 Object 說明及參數設定

## 9.1 標準 Object

Index	項目	Sub-Index	說明	類型	標誌	備註
1000h	Device type	00h	Device type	UINT32	RO	0x0000 0192(No profile)
1001h	Error Register	00h	Error Register	UINT16	RO	0x00
1008h	Device name	00h	ECAT-2610/2611 Communicator	Visible string	RO	-
1009h	Hardware Version	00h	Hardware Version	UINT16	RO	-
100Ah	Software Version	00h	Software Version	UINT16	RO	-
1018h	Identity object	00h	Number of entries	UINT16	RO	04h
		01h	Vendor ID	UINT32	RO	-
		02h	Product Code	UINT32	RO	-
		03h	Revision Number	UINT32	RO	-
		04h	Serial Number	UINT32	RO	-
1600h	Receive PDO mapping	00h	RxPDO 0x00-0x7F	UINT16	RO	參考下表 8-1
1601h		RxPDO 0x80-0xFF	UINT16	RO		
1A00h	Transmit PDO mapping	00h	TxPDO 0x00-0x7F	UINT16	RO	參考下表 8-1
1A01h		TxPDO 0x80-0xFF	UINT16	RO		
1C00h	Sync Manager Communication Type	00h	Sync Manger Type	UINT16	RO	04h
		01h	Write to Mailbox	UINT16	RO	01h
		02h	Read from Mailbox	UINT16	RO	02h
		03h	Process Data Out	UINT16	RO	03h
		04h	Process Dat in	UINT16	RO	04h
1C12h	Sync Manager Rx PDO Assign	00h	SyncManager 2 Assignment	UINT16	RO	No. of assigned RxPDO(0-1)
		01h	Assigned RxPDO	UINT16	RO	Assigned to RxPDO 1600h
		02h	Assigned RxPDO	UINT16	RO	Assigned to RxPDO 1601h
1C13h	Sync Manager Tx PDO Assign	00h	SyncManager 3 Assignment	UINT16	RO	No. of assigned TxPDO(0-1)
		01h	Assigned TxPDO	UINT16	RO	Assigned to TxPDO 1A00h
		02h	Assigned TxPDO	UINT16	RO	Assigned to TxPDO 1A01h

表 8-1:定義 ECAT-2610/2611 模組 PDO 映射位址及數量:

PDO	對應特定 Object	Internal Memory
TxPDO 1A00h	Index 2000h, sub-index 1 ~ 128	Input Data, bytes 0 ~ 127
TxPDO 1A01h	Index 2010h, sub-index 1 ~ 128	Input Data, bytes 128 ~ 255
RxPDO 1600h	Index 2100h, sub-index 1 ~ 128	Output Data, bytes 0 ~ 127
RxPDO 1601h	Index 2110h, sub-index 1 ~ 128	Output Data, bytes 128 ~ 255



## 9.2 特定 Object

### Input Buffer

Index	項目	Sub-Index	說明	類型	標誌
2000h	Inputs	00h	No. of entries	UINT16	RO
		01h	Input byte 0000	UINT16	RO
		02h	Input byte 0001	UINT16	RO
		---	---	---	---
		80h	Input byte 0127	UINT16	RO
2010h	Inputs	00h	No. of entries	UINT16	RO
		01h	Input byte 0128	UINT16	RO
		02h	Input byte 0129	UINT16	RO
		---	---	---	---
		80h	Input byte 0255	UINT16	RO

#### 注意

Gateway 只建立 Objects 所需的數量來保存子網段配置

### Output Buffer

Index	項目	Sub-Index	說明	類型	標誌
2100h	Outputs	00h	No. of entries	UINT16	RO
		01h	Output byte 0000	UINT16	R(W)
		02h	Output byte 0001	UINT16	R(W)
		---	---	---	---
		80h	Output byte 0127	UINT16	R(W)
2110h	Outputs	00h	No. of entries	UINT16	RO
		01h	Output byte 0128	UINT16	R(W)
		02h	Output byte 0129	UINT16	R(W)
		---	---	---	---
		80h	Output byte 0255	UINT16	R(W)

#### 注意

- 1: 因一致性原則，所以 I/O Data 的數據資料為唯讀 (Read-only)
- 2: Gateway 只建立 Objects 所需的數量來保存子網段配置

# 10 產品應用

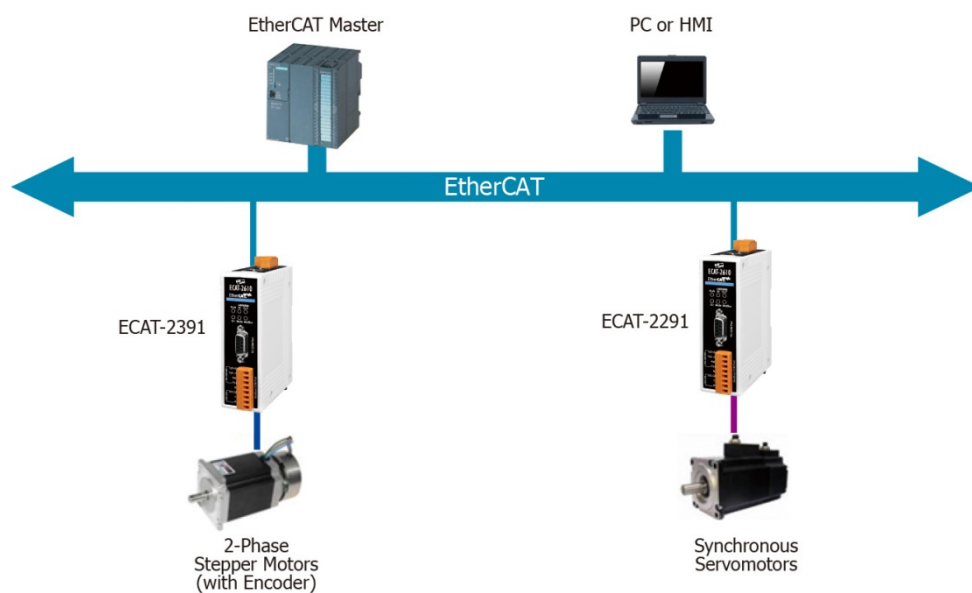
## 10.1 泓格 EtherCAT 系列產品

更多更詳細關於泓格 EtherCAT 系列產品，可參考 [EtherCAT 選型](#) 網站。



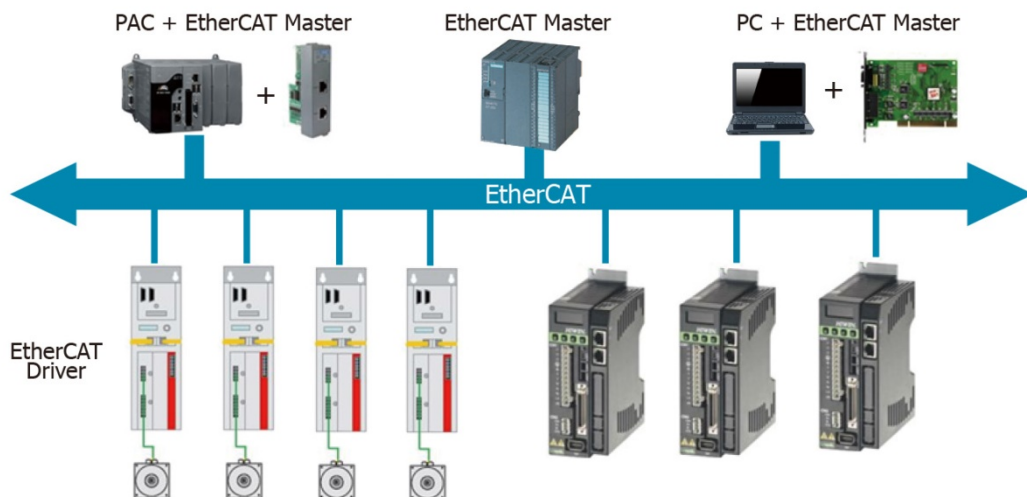
➤ EtherCAT 運動從站系統架構圖，如下：

### EtherCAT Motion Slave Solutions



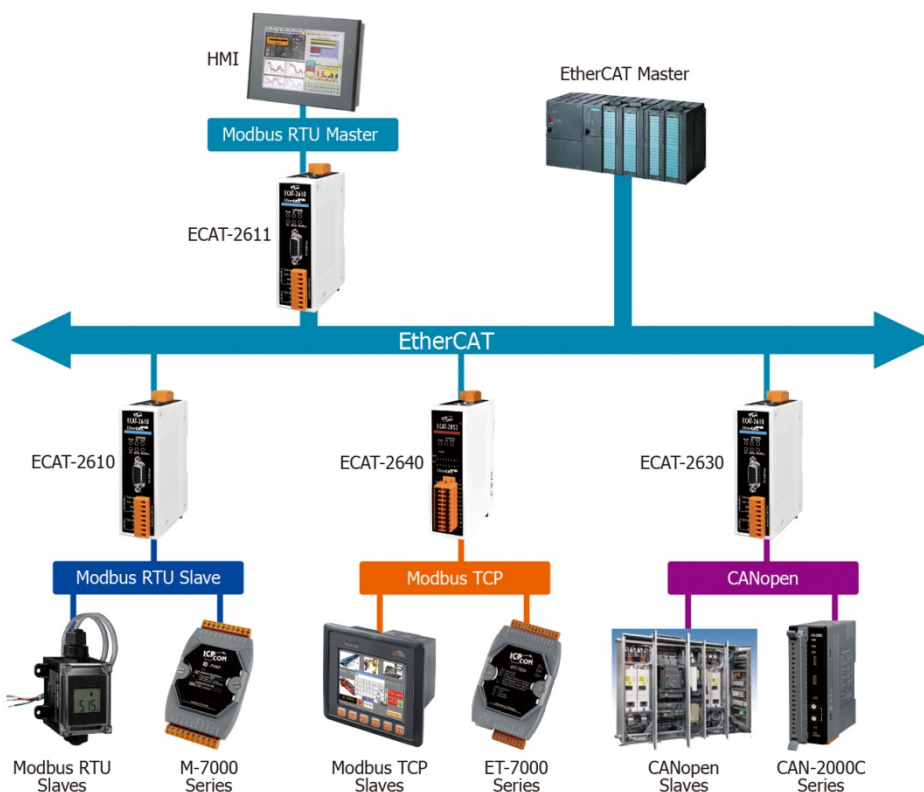
➤ EtherCAT 運動主站系統架構圖，如下：

## EtherCAT Motion Master Solutions



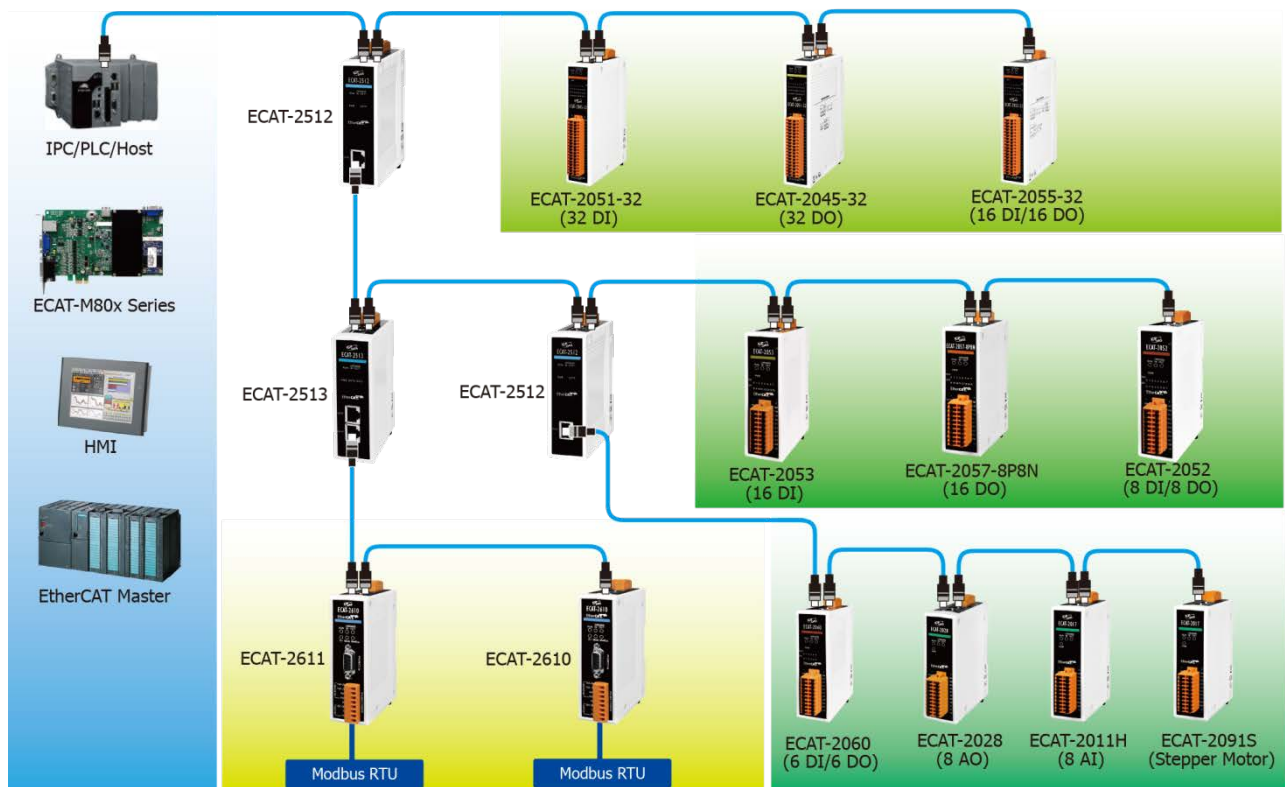
➤ EtherCAT 閘道器系統架構圖，如下：

## EtherCAT Gateway Solutions



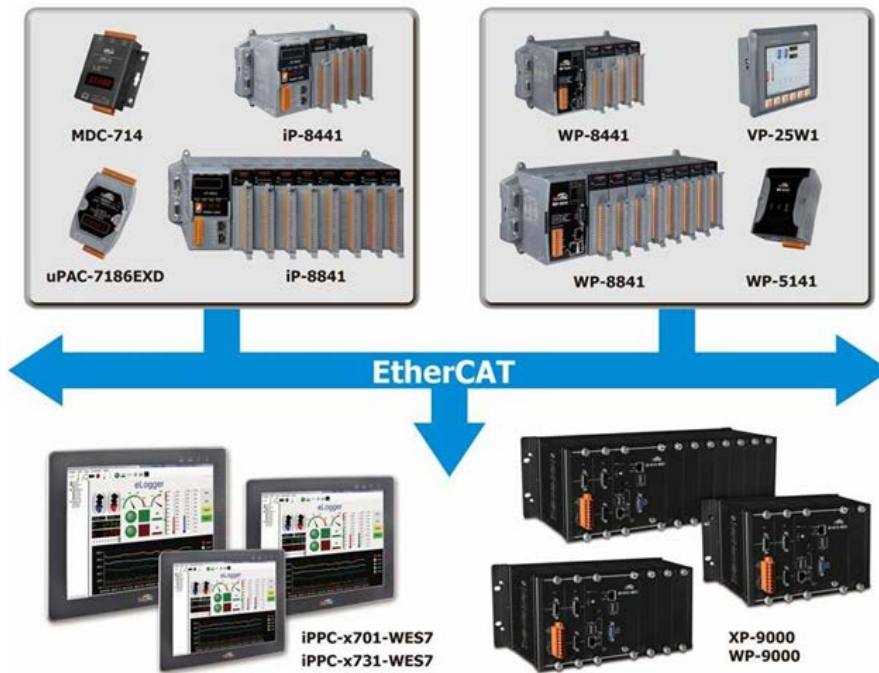
➤ EtherCAT 系列系統架構圖，如下：

### EtherCAT System Diagram (ICPDAS)

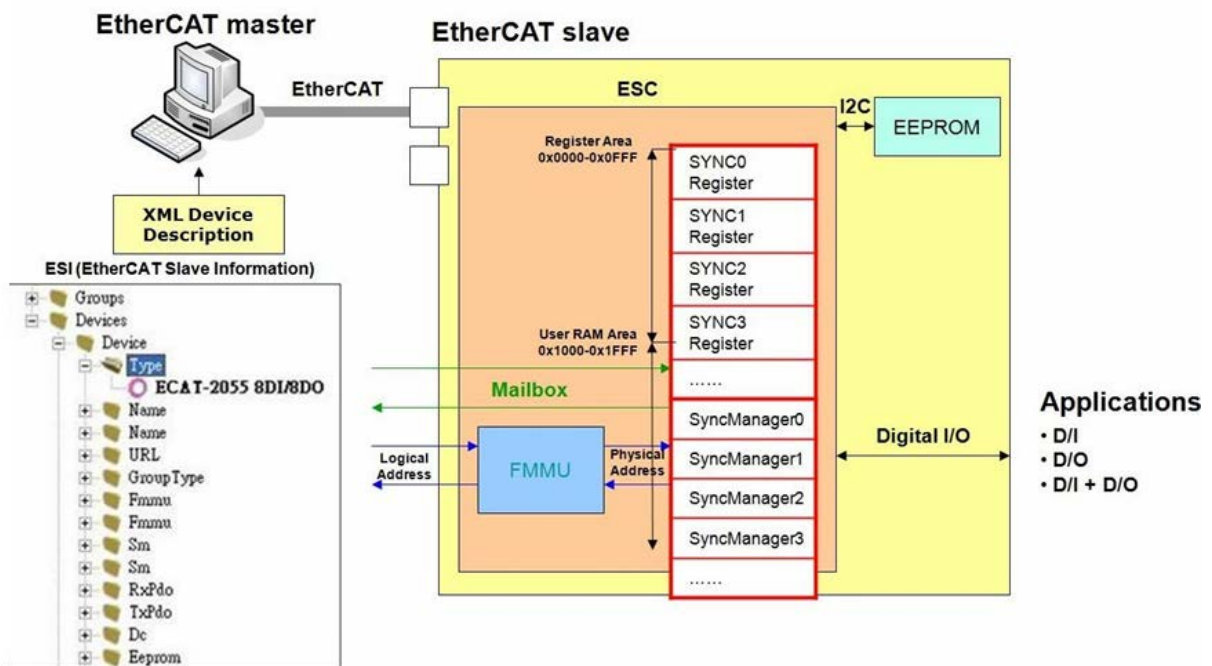


## 10.2 洽詢 ODM

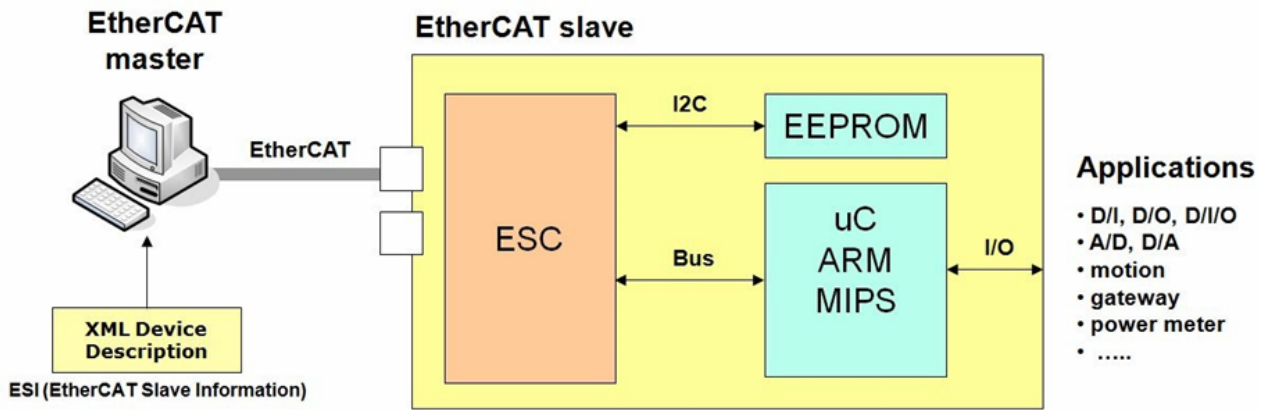
➤ 泓格提供了多樣化的 EtherCAT 主站解決方案，如下：



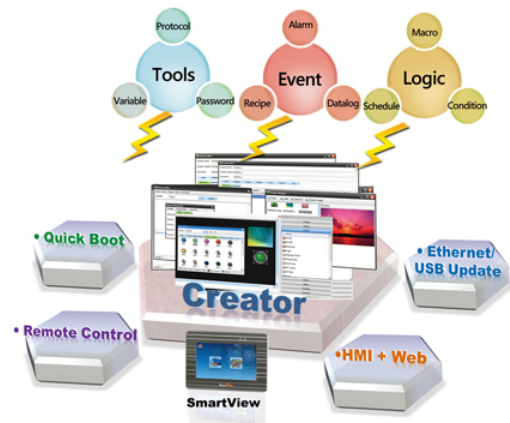
➤ 泓格提供了典型從站 Simple\_IO 解決方案，如下：



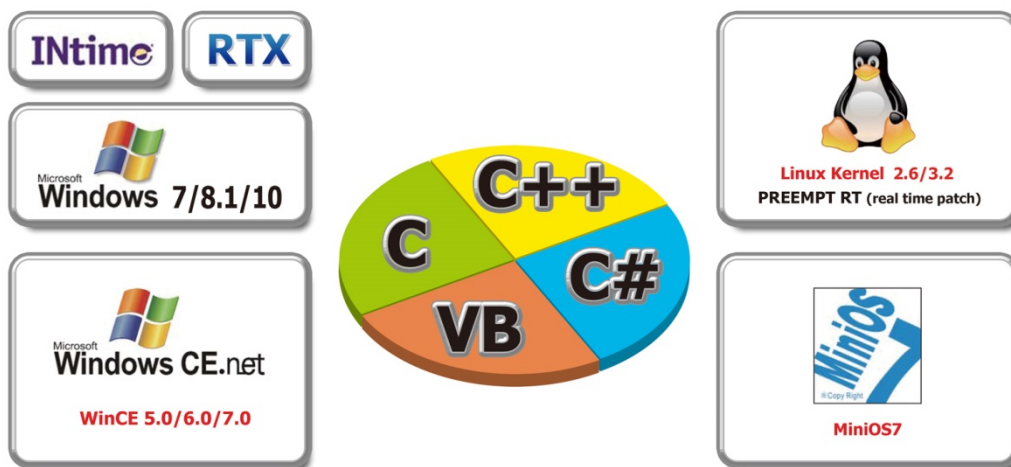
➤ 泓格提供了典型從站 Complex\_IO (ESC+uC: ARM 或 MIPS, 32-bit) 解決方案，如下：



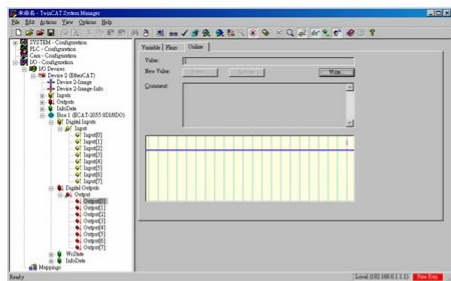
➤ 泓格提供了豐富的控制軟體系統，如下：



➤ 泓格支援多種開發軟體環境，如下：



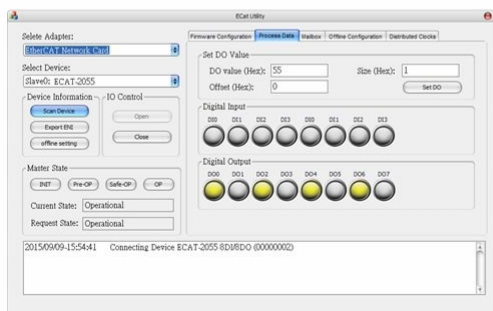
➤ 泓格提供了多樣軟體工具，如下：



Beckhoff TwinCAT2

### ECAT Utility Feature:

- Export EtherCAT Network Information file
- ICP DAS Slave I/O Module Diagnostic
- Firmware Configuration/Download



ICP DAS ECAT Utility

# 附錄

## A1. 如何透過 DCON Utility 來取得 Modbus RTU 指令

下面取得 Modbus RTU 指令方法只適用於我司的 Modbus RTU 設備，其它第三方設備請參考各自設備的使用手冊來執行。請將 Modbus RTU 設備 (如: M-7050) 連接至電腦主機並供電開機，詳細接線、設定可參考至 [章節 4.1 “配置 Modbus RTU 設備”](#)。

- ❶ 在 DCON Utility Pro 軟體 (如還沒安裝 DCON Utility Pro 軟體，請參考[章節 4.1 “配置 Modbus RTU 設備”](#))，按下 **“Command Line”** 按鈕來開啟 **“Tool for Terminal Command”** 對話框。
- ❷ 在 **COM Port**、**Baud Rate**、**Format** 及 **Address** 下拉式選單中依據您的 Modbus RTU 設備來設定。
- ❸ 從 **“Select ID”** 下拉式選單中設定您的 **Modbus RTU 設備名稱**後，將會帶出此設備所有的指令。
- ❹ 點選您所需的指令。
- ❺ 在 **Command** 欄位給出 Modbus RTU 指令碼 (不含 checksum)。可復制此指令至配置檔 (commands.txt)中使用。
- ❻ 單擊 **“Send”** 按鈕。
- ❼ 將顯示 Modbus RTU 指令+回應訊息。

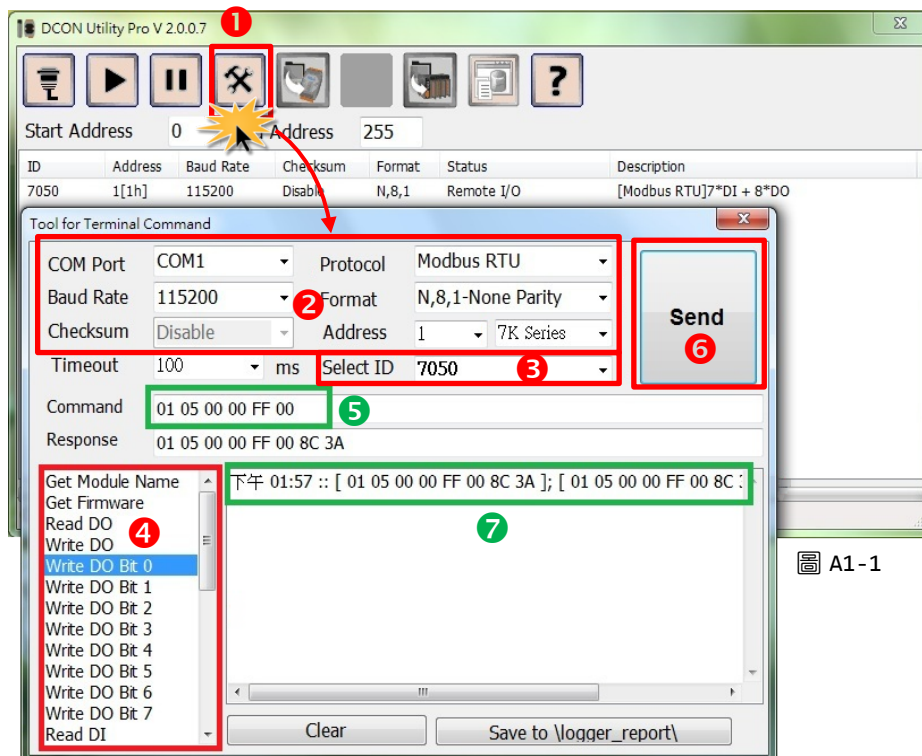


圖 A1-1



## A2. ECAT-2610 配置檔參考文件

在 7188ECAT 資料夾 (配置/偵錯工具) 中包含 **more commands** 資料夾，裡面提供了許多關於 DI、DO、AD 及 DA ...等配置檔 (commands.txt) 範例，您可參考這些範例來有效正確的設定配置檔 (commands.txt)，詳細說明如下。

### ⚠ 注意

此章節是使用我司 Modbus RTU 設備 (M-7000 系列) 作為說明範例，其它第三方設備請參考各自設備的使用手冊來設定 Modbus 命令。

➤ 開啟 **more commands** 資料夾，內容如下：

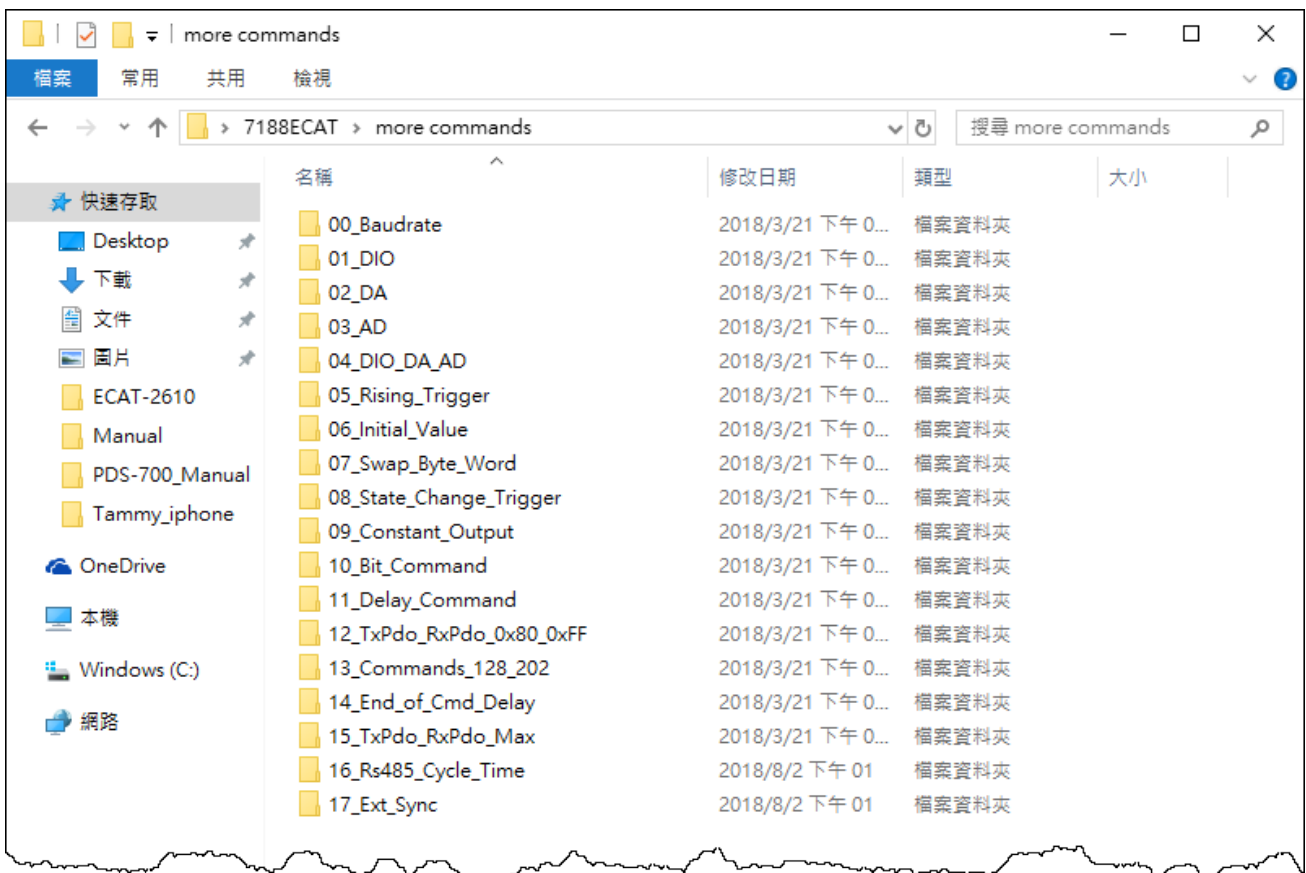


圖 A2-1

## 00.Baudrate

**00\_Baudrate** 資料夾提供設定 Baud Rate、Parity 及 Stop Bit 的範例，詳細說明如下：

**注意: ECAT-2610 僅支援 16 Data Bits。**

### 115200\_N81\_Init.txt

**115200\_N81\_Init.txt** 範例是 ECAT-2610 原廠預設的配置檔 (commands.txt)，如下：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
3, three commands(00-02), max=300, format=Dec
FF 03 00 00 00 02, 02, 00, 00, InTxPdo[2]=InTxPdo[0]=Sys_Lo, InTxPdo[3]=InTxPdo[1]=Sys_Hi (00)
FF 06 00 00 00 64, 02, 00, 00, 0x64=100, delay 100X0.01 sec = 1 sec, (01)
FF 06 00 01 00 64, 02, 00, 00, 0x64=100, delay 100X1ms = 0.1 sec, (02)
STOP
```

---

```
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
```

圖 A2-2

範例：

Baud rate = **115200**, Parity = **N** (None), Stop Bit = **1**, TimeOut = **100**, 命令數量 = **3**,

(00) Modbus 命令, PDO[位址], 更新模式, 特殊碼 = **FF 03 00 00 00 02, 02, 00, 00**,

此命令用來讀取 ECAT-2610 模組狀態，詳細參考 [章節 3.3.1 “模組狀態及錯誤模式”](#)。

(01) Modbus 命令, PDO[位址], 更新模式, 特殊碼 = **FF 06 00 00 00 64, 02, 00, 00**,

(02) Modbus 命令, PDO[位址], 更新模式, 特殊碼 = **FF 06 00 01 00 64, 02, 00, 00**,

此命令用來延遲 Modbus 命令，詳細參考 [11.Delay Command](#)。

## 9600\_N81.txt

**9600\_N81.txt** 和 [115200 N81 init.txt](#) 很相似，除了 Baud Rate (範例: 9600) · 如下:

```
START
9600, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
3, three commands(00-02), max=300, format=Dec
FF 03 00 00 00 02, 02, 00, 00, InTxPdo[2]=InTxPdo[0]=Sys_Lo, InTxPdo[3]=InTxPdo[1]=Sys_Hi (00)
FF 06 00 00 00 64, 02, 00, 00, 0x64=100, delay 100X0.01 sec = 1 sec, (01)
FF 06 00 01 00 64, 02, 00, 00, 0x64=100, delay 100X1ms = 0.1 sec, (02)
STOP
```

---

```
OutRxPdo[00]=261OCTL0, OutRxPdo[01]=261OCTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
```

圖 A2-3

## 19200\_N82.txt

**19200\_N82.txt** 和 [115200 N81 init.txt](#) 很相似，除了 Baud Rate 及 Stop Bit (範例: 19200 及 2) · 如下:

```
START
19200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
2, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
3, three commands(00-02), max=300, format=Dec
FF 03 00 00 00 02, 02, 00, 00, InTxPdo[2]=InTxPdo[0]=Sys_Lo, InTxPdo[3]=InTxPdo[1]=Sys_Hi (00)
FF 06 00 00 00 64, 02, 00, 00, 0x64=100, delay 100X0.01 sec = 1 sec, (01)
FF 06 00 01 00 64, 02, 00, 00, 0x64=100, delay 100X1ms = 0.1 sec, (02)
STOP
```

---

```
OutRxPdo[00]=261OCTL0, OutRxPdo[01]=261OCTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
```

圖 A2-4

## 38400\_E81.txt

**38400\_E81.txt** 和 [115200\\_N81\\_init.txt](#) 很相似，除了 Baud Rate 及 Parity (範例: 38400 及 EVEN):

```
START
38400, baud rate, from 1200,2400 ~ 57600,115200
E, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
3, three commands(00-02), max=300, format=Dec
FF 03 00 00 00 02, 02, 00, 00, InTxPdo[2]=InTxPdo[0]=Sys_Lo, InTxPdo[3]=InTxPdo[1]=Sys_Hi (00)
FF 06 00 00 00 64, 02, 00, 00, 0x64=100, delay 100X0.01 sec = 1 sec, (01)
FF 06 00 01 00 64, 02, 00, 00, 0x64=100, delay 100X1ms = 0.1 sec, (02)
STOP
```

---

```
OutRxPdo[00]=261OCTL0, OutRxPdo[01]=261OCTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=261OSYS0, InTxPdo[01]=261OSYS1, InTxPdo[02..FF]=Out[02..FF]
```

圖 A2-5

## 57600\_O81.txt

**57600\_O81.txt** 和 [115200\\_N81\\_init.txt](#) 很相似，除了 Baud Rate 及 Parity (範例: 57600 及 ODD):

```
START
57600, baud rate, from 1200,2400 ~ 57600,115200
O, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
3, three commands(00-02), max=300, format=Dec
FF 03 00 00 00 02, 02, 00, 00, InTxPdo[2]=InTxPdo[0]=Sys_Lo, InTxPdo[3]=InTxPdo[1]=Sys_Hi (00)
FF 06 00 00 00 64, 02, 00, 00, 0x64=100, delay 100X0.01 sec = 1 sec, (01)
FF 06 00 01 00 64, 02, 00, 00, 0x64=100, delay 100X1ms = 0.1 sec, (02)
STOP
```

---

```
OutRxPdo[00]=261OCTL0, OutRxPdo[01]=261OCTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=261OSYS0, InTxPdo[01]=261OSYS1, InTxPdo[02..FF]=Out[02..FF]
```

圖 A2-6

## 01.DIO

**01\_DIO** 資料夾提供 Digital Input (DI) 及 Digital Output (DO) 的相關設定範例，詳細說明如下：

### DIO\_Addr01\_1.txt

**DIO\_Addr01\_1.txt** 提供 1 個 Write DO 的 Modbus 命令範例，如下圖所示：

```

START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
1, one commands(00-00), max=300, format=Dec
01 0F 00 00 00 08 01 00, 02, 00, 00, D/O=OutTxPdo[2], update cyclically, (00)
STOP
=====
OutRxPdo[00]=261OCTL0, OutRxPdo[01]=261OCTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]

```

圖 A2-7

以 M-7050 模組為範例。

命令數量= 1,

**Modbus 命令, PDO[位址], 更新模式, 特殊碼 =**

(00) **01 0F 00 00 00 08 01 00, 02, 00, 00**, → 從 OutRxPDO[02] 讀取資料，然後將 Modbus 命令傳送到 DO 模組。

## DIO\_Addr01\_2.txt

DIO\_Addr01\_2.txt 提供 2 個 Modbus 命令範例為 Write DO 及 Read DI，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
2, two commands(00-01), max=300, format=Dec
01 0F 00 00 00 08 01 00, 02, 00, 00, D/O=OutTxPdo[2], update cyclically, (00)
01 02 00 00 00 08, 02, 00, 00, InTxPdo[02]=D/I, update cyclically, (01)
STOP
```

---

```
OutRxPdo[00]=261OCTL0, OutRxPdo[01]=261OCTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=261SYS0, InTxPdo[01]=261SYS1, InTxPdo[02..FF]=Out[02..FF]
```

圖 A2-8

以 M-7050 模組為範例。

命令數量= 2,

**Modbus 命令, PDO[位址], 更新模式, 特殊碼 =**

(00) **01 0F 00 00 00 08 01 00, 02, 00, 00**, → 從 OutRxPDO[02] 讀取資料，然後將 Modbus 命令傳送到 DO 模組。

(01) **01 02 00 00 00 08, 02, 00, 00**, → 讀取模組的 DI，然後將讀取值寫到 InTxPDO[02]。

## DIO\_Addr01\_3.txt

DIO\_Addr01\_3.txt 和 [DIO\\_Addr01\\_2.txt](#) 很相似，除了 Read DI Latch，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
3, three commands(00-02), max=300, format=Dec
01 0F 00 00 00 08 01 00, 02, 00, 00, D/I=OutTxPdo[2], update cyclically, (00)
01 02 00 00 00 08, 02, 00, 00, InTxPdo[02]=D/I, update cyclically, (01)
01 01 00 40 00 07, 03, 00, 00, InTxPdo[03]=D/I latch, update cyclically, (02)
STOP
```

---

```
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
```

圖 A2-9

以 M-7050 模組為範例，

命令數量= 3,

**Modbus 命令, PDO[位址], 更新模式, 特殊碼 =**

(00) **01 0F 00 00 00 08 01 00, 02, 00, 00** → 從 OutRxPDO[02] 讀取資料，然後將 Modbus 命令傳送到 DO 模組。

(01) **01 02 00 00 00 08, 02, 00, 00** → 讀取模組的 DI，然後將讀取值寫到 InTxPDO[02]。

(02) **01 01 00 40 00 07, 03, 00, 00** → 讀取模組的 DI Latch High 值，然後將讀取值寫到 InTxPDO[03]。

## DIO\_Addr01\_4.txt

DIO\_Addr01\_4.txt 和 [DIO\\_Addr01\\_2.txt](#) 很相似，除了 Read DI Latch 及 DO Readback，如下圖所示：

```

START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
4, four commands(00-03), max=300, format=Dec
01 0F 00 00 00 08 01 00, 02, 00, 00, D/O=OutTxPdo[2], update cyclically, (00)
01 01 00 00 00 08, 02, 00, 00, InTxPdo[02]=D/O read back, update cyclically, (01)
01 02 00 00 00 08, 03, 00, 00, InTxPdo[03]=D/I, update cyclically, (02)
01 01 00 40 00 07, 04, 00, 00, InTxPdo[04]=D/I_Latch_High, update cyclically, (03)
STOP
=====
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
=====

```

圖 A2-10

以 M-7050 模組為範例，

命令數量=4，

**Modbus 命令, PDO[位址], 更新模式, 特殊碼 =**

(00) 01 0F 00 00 00 08 01 00, 02, 00, 00, → 從 OutRxPDO[02] 讀取資料，然後將 Modbus 命令傳送到 DO 模組。

(01) 01 0F 00 00 00 08 01 00, 02, 00, 00, → 讀取模組的 DO Readback 值，然後將讀取值寫到 InTxPDO[02]。

(02) 01 02 00 00 00 08, 02, 00, 00, → 讀取模組的 DI，然後將讀取值寫到 InTxPDO[03]。

(03) 01 01 00 40 00 07, 03, 00, 00, → 讀取模組的 DI Latch High 值，然後將讀取值寫到 InTxPDO[04]。



## 02.DA

02\_DA 資料夾提供 Analog Output (DA) 的相關設定範例，詳細說明如下：

### DA\_Addr02\_1.txt

DA\_Addr02\_1.txt 提供 4 個 Modbus 命令範例為 Write AO0 ~AO3，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
4, four commands(00-03), max=300, format=Dec
02 06 00 00 00 00, 02, 00, 00, D/A_0=OutRxPdo[02], update cyclically, (00)
02 06 00 01 00 00, 03, 00, 00, D/A_1=OutRxPdo[03], update cyclically, (01)
02 06 00 02 00 00, 04, 00, 00, D/A_2=OutRxPdo[04], update cyclically, (02)
02 06 00 03 00 00, 05, 00, 00, D/A_3=OutRxPdo[05], update cyclically, (03)
STOP
```

---

```
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
```

圖 A2-11

以 M-7024 模組為範例，

命令數量= 4,

**Modbus 命令, PDO[位址], 更新模式, 特殊碼 =**

- (00) 02 06 00 00 00 00, 02, 00, 00, → 讀取 OutRxPDO[02] 的資料並傳送 Modbus 命令到 AO0。
- (01) 02 06 00 01 00 00, 03, 00, 00, → 讀取 OutRxPDO[03] 的資料並傳送 Modbus 命令到 AO1。
- (02) 02 06 00 02 00 00, 04, 00, 00, → 讀取 OutRxPDO[04] 的資料並傳送 Modbus 命令到 AO2。
- (03) 02 06 00 03 00 00, 05, 00, 00, → 讀取 OutRxPDO[05] 的資料並傳送 Modbus 命令到 AO3。

M-7024 模組支援 4 個 16-bit DA 通道。下表為相對應位址：

DA channel	PDO[位址]
0	OutRxPDO[02]
1	OutRxPDO[03]
2	OutRxPDO[04]
3	OutRxPDO[05]

## DA\_Addr02\_2.txt

DA\_Addr02\_2.txt 提供 1 個 Modbus 命令範例為 Write 全部 AO，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
1, one commands(00-00), max=300, format=Dec
02 10 00 00 00 04 08 12 34 12 34 12 34 12 34, 02, 00, 00, DA_0/1/2/3=OutRxPdo[2/3/4/5], update cyclically, (00)
STOP
```

---

```
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
```

圖 A2-12

以 M-7024 模組為範例。

命令數量= 1,

**Modbus 命令, PDO[位址], 更新模式, 特殊碼 =**

(00) 02 10 00 00 00 04 08 12 34 12 34 12 34 12 34, 02, 00, 00,

→ 一次讀取 OutRxPDO[02/03/04/05] 的資料並傳送 Modbus 命令到 AO0/1/2/3。

M-7024 模組支援 4 個 16-bit DA 通道。下表為相對應位址：

DA channel	PDO[位址]
0	OutRxPDO[02]
1	OutRxPDO[03]
2	OutRxPDO[04]
3	OutRxPDO[05]

## DA\_Addr02\_3.txt

DA\_Addr02\_3.txt 和 [DA\\_Addr02\\_1.txt](#) 很相似，除了 AO0 ~ AO3 Readback，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
8, eight commands(00-07), max=300, format=Dec
02 06 00 00 00 00, 02, 00, 00, D/A_0=OutRxPdo[02], update cyclically, (00)
02 06 00 01 00 00, 03, 00, 00, D/A_1=OutRxPdo[03], update cyclically, (01)
02 06 00 02 00 00, 04, 00, 00, D/A_2=OutRxPdo[04], update cyclically, (02)
02 06 00 03 00 00, 05, 00, 00, D/A_3=OutRxPdo[05], update cyclically, (03)
02 03 00 40 00 01, 02, 00, 00, InTxPdo[02]=D/A_0 read back, update cyclically, (04)
02 03 00 41 00 01, 03, 00, 00, InTxPdo[03]=D/A_1 read back, update cyclically, (05)
02 03 00 42 00 01, 04, 00, 00, InTxPdo[04]=D/A_2 read back, update cyclically, (06)
02 03 00 43 00 01, 05, 00, 00, InTxPdo[05]=D/A_3 read back, update cyclically, (07)
STOP
```

---

```
OutRxPdo[00]=261OCTL0, OutRxPdo[01]=261OCTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=261OSYS0, InTxPdo[01]=261OSYS1, InTxPdo[02..FF]=Out[02..FF]
```

圖 A2-13

以 M-7024 模組為範例，

命令數量= 8,

**Modbus 命令, PDO[位址], 更新模式, 特殊碼 =**

- (00) 02 06 00 00 00 00, 02, 00, 00, → 讀取 OutRxPDO[02] 的資料並傳送 Modbus 命令到 AO0。
- (01) 02 06 00 01 00 00, 03, 00, 00, → 讀取 OutRxPDO[03] 的資料並傳送 Modbus 命令到 AO1。
- (02) 02 06 00 02 00 00, 04, 00, 00, → 讀取 OutRxPDO[04] 的資料並傳送 Modbus 命令到 AO2。
- (03) 02 06 00 03 00 00, 05, 00, 00, → 讀取 OutRxPDO[05] 的資料並傳送 Modbus 命令到 AO3。
- (04) 02 06 00 00 00 00, 02, 00, 00, → 讀取模組的 AO0 Readback 值，然後將讀取值寫到 InTxPDO[02]。
- (05) 02 06 00 01 00 00, 03, 00, 00, → 讀取模組的 AO1 Readback 值，然後將讀取值寫到 InTxPDO[03]。
- (06) 02 06 00 02 00 00, 04, 00, 00, → 讀取模組的 AO2 Readback 值，然後將讀取值寫到 InTxPDO[04]。
- (07) 02 06 00 03 00 00, 05, 00, 00, → 讀取模組的 AO3 Readback 值，然後將讀取值寫到 InTxPDO[05]。

## DA\_Addr02\_4.txt

DA\_Addr02\_4.txt 和 [DA\\_Addr02\\_2.txt](#) 很相似，除了全部 AO Readback，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
2, two commands(00-01), max=300, format=Dec
02 10 00 00 00 04 08 12 34 12 34 12 34 12 34, 02, 00, 00, DA_0/1/2/3=OutRxPdo[2/3/4/5], update cyclically, (00)
02 03 00 40 00 04, 02, 00, 00, InTxPdo[2/3/4/5]=DA_0/1/2/3 read back, update cyclically, (01)
STOP

-----
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
-----
```

圖 A2-14

以 M-7024 模組為範例。

命令數量= 2,

**Modbus 命令, PDO[位址], 更新模式, 特殊碼 =**

(00) **02 10 00 00 00 04 08 12 34 12 34 12 34 12 34, 02, 00, 00,**

→ 一次讀取 OutRxPDO[02/03/04/05] 的資料並傳送 Modbus 命令到 AO0/1/2/3。

(01) **02 03 00 40 00 04, 02, 00, 00,**

→ 一次讀取模組的 AO0/1/2/3 Readback 值，然後將讀取值寫到 InTxPDO[02/03/04/05]。

## DA\_Addr02\_5.txt

DA\_Addr02\_5.txt 提供 5 個 Modbus 命令範例為 Write AO 及 AO Readback，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
5, four commands(00-04), max=300, format=Dec
02 03 00 40 00 02, 02, 00, 00, InTxPdo[2/3]=DA_0/1 read back, update cyclically, (00)
02 03 00 42 00 02, 04, 00, 00, InTxPdo[4/5]=DA_2/3 read back, update cyclically, (01)
02 10 00 00 00 02 04 12 34 12 34, 02, 00, 00, DA_0/1=OutTxPdo[2/3], update cyclically, (02)
02 06 00 02 00 00, 04, 00, 00, DA_2=OutTxPdo[4], update cyclically, (03)
02 06 00 03 00 00, 05, 00, 00, DA_3=OutTxPdo[5], update cyclically, (04)
STOP
```

---

```
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
```

圖 A2-15

命令數量= 5,

**Modbus 命令, PDO[位址], 更新模式, 特殊碼 =**

(00) 02 03 00 40 00 02, 02, 00, 00,

→ 讀取模組的 AO0/1 Readback 值，再將讀取值寫到 InTxPDO[02/03]。

(01) 02 03 00 42 00 02, 04, 00, 00,

→ 讀取模組的 AO2/3 Readback 值，再將讀取值寫到 InTxPDO[04/05]。

(02) 02 10 00 00 00 02 04 12 34 12 34, 02, 00, 00,

→ 讀取 OutRxPDO[02/03] 的資料並傳送 Modbus 命令到 AO0/1。

(03) 02 06 00 02 00 00, 04, 00, 00, → 讀取 OutRxPDO[04] 的資料並傳送 Modbus 命令到 AO2。

(04) 02 06 00 03 00 00, 05, 00, 00, → 讀取 OutRxPDO[05] 的資料並傳送 Modbus 命令到 AO3。

### 03.AD

03\_AD 資料夾提供 Analog Input (AD) 的相關設定範例，詳細說明如下：

#### AD\_Addr03\_1.txt

AD\_Addr03\_1.txt 提供 8 個 Modbus 命令範例為 Read AI0 ~ AI7，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
8, 8 commands(00-07), max=300, format=Dec
03 04 00 00 00 01, 02, 00, 00, InTxPdo[2]=A/I_0, update cyclically, (00)
03 04 00 01 00 01, 03, 00, 00, InTxPdo[3]=A/I_1, update cyclically, (01)
03 04 00 02 00 01, 04, 00, 00, InTxPdo[4]=A/I_2, update cyclically, (02)
03 04 00 03 00 01, 05, 00, 00, InTxPdo[5]=A/I_3, update cyclically, (03)
03 04 00 04 00 01, 06, 00, 00, InTxPdo[6]=A/I_4, update cyclically, (04)
03 04 00 05 00 01, 07, 00, 00, InTxPdo[7]=A/I_5, update cyclically, (05)
03 04 00 06 00 01, 08, 00, 00, InTxPdo[8]=A/I_6, update cyclically, (06)
03 04 00 07 00 01, 09, 00, 00, InTxPdo[9]=A/I_7, update cyclically, (07)
STOP
-----
OutRxPdo[00]=261OCTL0, OutRxPdo[01]=261OCTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=261SYS0, InTxPdo[01]=261SYS1, InTxPdo[02..FF]=Out[02..FF]
```

圖 A2-16

以 M-7017 模組為範例，

命令數量= 8,

**Modbus 命令, PDO[位址], 更新模式, 特殊碼=**

- (00) 03 04 00 00 00 01, 02, 00, 00, → 讀取模組的 AI0，然後將讀取值寫到 InTxPDO[02]。
- (01) 03 04 00 01 00 01, 03, 00, 00, → 讀取模組的 AI1，然後將讀取值寫到 InTxPDO[03]。
- (02) 03 04 00 02 00 01, 04, 00, 00, → 讀取模組的 AI2，然後將讀取值寫到 InTxPDO[04]。
- (03) 03 04 00 03 00 01, 05, 00, 00, → 讀取模組的 AI3，然後將讀取值寫到 InTxPDO[05]。
- (04) 03 04 00 04 00 01, 06, 00, 00, → 讀取模組的 AI4，然後將讀取值寫到 InTxPDO[06]。
- (05) 03 04 00 05 00 01, 07, 00, 00, → 讀取模組的 AI5，然後將讀取值寫到 InTxPDO[07]。
- (06) 03 04 00 06 00 01, 08, 00, 00, → 讀取模組的 AI6，然後將讀取值寫到 InTxPDO[08]。
- (07) 03 04 00 07 00 01, 09, 00, 00, → 讀取模組的 AI7，然後將讀取值寫到 InTxPDO[09]。

M-7017 模組支援 7 個 16-bit AD 通道，下表為相對應位址：

AD channel	PDO[位址]
0	InTxPDO[02]
1	InTxPDO[03]
2	InTxPDO[04]
3	InTxPDO[05]
4	InTxPDO[06]
5	InTxPDO[07]
6	InTxPDO[08]
7	InTxPDO[09]

## AD\_Addr03\_2.txt

AD\_Addr03\_2.txt 提供 1 個 Modbus 命令範例為 Read 全部 AI，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
1, 1 commands(00-00), max=300, format=Dec
03 04 00 00 00 08, 02, 00, 00, InTxPdo[2/9]=A/I_0~7, update cyclically, (00)
STOP
```

---

```
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
```

圖 A2-17

以 M-7017 模組為範例，

命令數量= 1,

Modbus 命令, PDO[位址], 更新模式, 特殊碼 =

(00) 03 04 00 00 00 08, 02, 00, 00, → 一次讀取模組的 AI0/1/2/3/4/5/6/7 值，然後將讀取值寫到 InTxPDO[02/03/04/05/06/07/08/09]。

## 04.DIO\_DA\_AD

**04\_DIO\_DA\_AD** 資料夾提供 Digital Input (DI)、Digital Output (DO)、Analog Input (AD)、Analog Output (DA) 及 Modbus 命令延遲時間...等相關設定範例，詳細說明如下：

### DIO\_DA\_AD\_1.txt

**DIO\_DA\_AD\_1.txt** 提供 5 個 Modbus 命令範例 (設定 DO、DI、DA、AD 及 Delay 2 ms)，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
5, five commands(00-04), max=300, format=Dec
01 0F 00 00 00 08 01 00, 02, 00, 00, D/O=OutTxPdo[2], update cyclically, (00)
01 02 00 00 00 08, 02, 00, 00, InTxPdo[02]=D/I, update cyclically, (01)
02 10 00 00 00 04 08 12 34 12 34 12 34 12 34, 03, 00, 00, DA_0/1/2/3=OutRxPdo[3/4/5/6], update cyclically, (02)
FF 06 00 01 00 02, 02, 00, 00, delay 1ms x 2=2ms, A/D need more delay @ 115.2K (03)
03 04 00 00 00 08, 03, 00, 00, InTxPdo[3/0A]=A/I_0~7, update cyclically, (04)
STOP

-----
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
-----

DIO Address = 0x01
D/A Address = 0x02
A/D Address = 0x03
```

圖 A2-18

命令數量= 5,

**Modbus 命令, PDO[位址], 更新模式, 特殊碼=**

(00) **01 0F 00 00 00 08 01 00, 02, 00, 00** → 讀取 OutRxPDO[02] 並傳送 Modbus 命令到 DO 模組。

(01) **01 02 00 00 00 08, 02, 00, 00** → 讀取模組的 DI，然後將讀取值寫到 InTxPDO[02]。

(02) **02 10 00 00 00 04 08 12 34 12 34 12 34 12 34, 03, 00, 00** → 一次讀取 OutRxPDO[02/03/04/05] 並傳送 Modbus 命令到 AO0/1/2/3。

(03) **FF 06 00 01 00 02, 02, 00, 00** → Delay 2 ms。

(04) **03 04 00 00 00 08, 03, 00, 00** → 一次讀取 AI0/1/2/3/4/5/6/7，然後將讀取值寫到 InTxPDO[03/04/05/06/07/08/09/0A]。



## 05.Rising\_Trigger

05\_Rising\_Trigger 資料夾提供 Digital Input (DI) 的 Read/Clear Counter 及 High/Low Latch...等相關設定範例，詳細說明如下：

### RisingTrigger\_1.txt

RisingTrigger\_1.txt 和 [DIO Addr01\\_2.txt](#) 很相似，除了 Read/Clear DIO Counter，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
4, four commands(00-03), max=300, format=Dec
01 0F 00 00 00 08 01 00, 02, 00, 00, D/O=OutRxPdo[02], update cyclically, (00)
01 02 00 00 00 07, 02, 00, 00, InTxPdo[02]=D/I, update cyclically, (01)
01 04 00 00 00 01, 03, 00, 00, InTxPdo[03]=Cnt_0, update cyclically, (02)
01 05 02 00 FF 00, 03, 01, 00, Clear Cnt_0, update OutRxPdo[03].bit0 rising, (03)
STOP
```

---

```
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
```

圖 A2-19

以 M-7050 模組為範例，

M-7050 模組 DI 可以用來作 16-bit 事件計數器，在命令(02) 將讀取 Counter0 並循環執行。命令(03) 中更新模組不等於 00，所以該命令將不會被循環執行。如果 OutRxPDO[03] 的 bit0 從 0 變為 1 (上升)，命令(03) 將被執行一次，也就是 DIO Counter 將被清除為 0。

下表為 DI、DO、Counter0 相對應位址：

DO/DI/Event Counter	PDO[位址]	清除 Counter
DO	OutRxPDO[02]	-
DI	InTxPDO[02]	-
Cnt_0	InTxPDO[03]	OutRxPDO[03].bit0

## RisingTrigger\_2.txt

RisingTrigger\_2.txt 和 [RisingTrigger\\_1.txt](#) 很相似，除了 Read/Clear DI0 ~ DI6 Counter，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
16, 16 commands(00-15), max=300, format=Dec
01 0F 00 00 00 08 01 00, 02, 00, 00, D/O=OutRxPdo[02], update cyclically, (00)
01 02 00 00 00 07, 02, 00, 00, InTxPdo[02]=D/I, update cyclically, (01)
01 04 00 00 00 01, 03, 00, 00, InTxPdo[03]=Cnt_0, update cyclically, (02)
01 04 00 01 00 01, 04, 00, 00, InTxPdo[04]=Cnt_1, update cyclically, (03)
01 04 00 02 00 01, 05, 00, 00, InTxPdo[05]=Cnt_2, update cyclically, (04)
01 04 00 03 00 01, 06, 00, 00, InTxPdo[06]=Cnt_3, update cyclically, (05)
01 04 00 03 00 01, 07, 00, 00, InTxPdo[07]=Cnt_4, update cyclically, (06)
01 04 00 03 00 01, 08, 00, 00, InTxPdo[08]=Cnt_5, update cyclically, (07)
01 04 00 03 00 01, 09, 00, 00, InTxPdo[09]=Cnt_6, update cyclically, (08)
01 05 02 00 FF 00, 03, 01, 00, Clear Cnt_0, update OutRxPdo[03].bit0 rising, (09)
01 05 02 01 FF 00, 03, 02, 00, Clear Cnt_1, update OutRxPdo[03].bit1 rising, (10)
01 05 02 02 FF 00, 03, 04, 00, Clear Cnt_2, update OutRxPdo[03].bit2 rising, (11)
01 05 02 03 FF 00, 03, 08, 00, Clear Cnt_3, update OutRxPdo[03].bit3 rising, (12)
01 05 02 04 FF 00, 03, 10, 00, Clear Cnt_4, update OutRxPdo[03].bit4 rising, (13)
01 05 02 05 FF 00, 03, 20, 00, Clear Cnt_5, update OutRxPdo[03].bit5 rising, (14)
01 05 02 06 FF 00, 03, 40, 00, Clear Cnt_6, update OutRxPdo[03].bit6 rising, (15)
STOP
=====
OutRxPdo[00]=261OCTL0, OutRxPdo[01]=261OCTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=261OSYS0, InTxPdo[01]=261OSYS1, InTxPdo[02..FF]=Out[02..FF]
=====
```

圖 A2-20

以 M-7050 模組為範例，

M-7050 模組支援 7 個事件計數器(Event Counter)。命令(02) 到命令(15) 為循環讀取這 7 個事件計數器 (Event Counter)。如果 OutRxPDO[03] 的 bit0 ~ bit6 是從 0 變為 1 (上升)，bit0 ~ bit6 將會被清除為 0。

下表為 7 個事件計數器的相對應位址：

Event Counter	InTxPDO[位址]	清除 Counter
0	InTxPDO[03]	OutRxPDO[03].bit0
1	InTxPDO[04]	OutRxPDO[03].bit1
2	InTxPDO[05]	OutRxPDO[03].bit2
3	InTxPDO[06]	OutRxPDO[03].bit3
4	InTxPDO[07]	OutRxPDO[03].bit4
5	InTxPDO[08]	OutRxPDO[03].bit5
6	InTxPDO[09]	OutRxPDO[03].bit6

## RisingTrigger\_3.txt

RisingTrigger\_3.txt 和 [RisingTrigger\\_2.txt](#) 很相似，除了 Read/Clear DI Latch，如下圖所示：

```

START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
19, 19 commands(00-18), max=300, format=Dec
01 0F 00 00 00 08 01 00, 02, 00, 00, D/I=OutRxPdo[02], update cyclically, (00)
01 02 00 00 00 07, 02, 00, 00, InTxPdo[02]=D/I, update cyclically, (01)
01 04 00 00 00 01, 03, 00, 00, InTxPdo[03]=Cnt_0, update cyclically, (02)
01 04 00 01 00 01, 04, 00, 00, InTxPdo[04]=Cnt_1, update cyclically, (03)
01 04 00 02 00 01, 05, 00, 00, InTxPdo[05]=Cnt_2, update cyclically, (04)
01 04 00 03 00 01, 06, 00, 00, InTxPdo[06]=Cnt_3, update cyclically, (05)
01 04 00 03 00 01, 07, 00, 00, InTxPdo[07]=Cnt_4, update cyclically, (06)
01 04 00 03 00 01, 08, 00, 00, InTxPdo[08]=Cnt_5, update cyclically, (07)
01 04 00 03 00 01, 09, 00, 00, InTxPdo[09]=Cnt_6, update cyclically, (08)
01 05 02 00 FF 00, 03, 01, 00, Clear Cnt_0, update OutRxPdo[03].bit0 rising, (09)
01 05 02 01 FF 00, 03, 02, 00, Clear Cnt_1, update OutRxPdo[03].bit1 rising, (10)
01 05 02 02 FF 00, 03, 04, 00, Clear Cnt_2, update OutRxPdo[03].bit2 rising, (11)
01 05 02 03 FF 00, 03, 08, 00, Clear Cnt_3, update OutRxPdo[03].bit3 rising, (12)
01 05 02 04 FF 00, 03, 10, 00, Clear Cnt_4, update OutRxPdo[03].bit4 rising, (13)
01 05 02 05 FF 00, 03, 20, 00, Clear Cnt_5, update OutRxPdo[03].bit5 rising, (14)
01 05 02 06 FF 00, 03, 40, 00, Clear Cnt_6, update OutRxPdo[03].bit6 rising, (15)
01 01 00 40 00 07, 10, 00, 00, InTxPdo[10]=D/I_Latch_High,update cyclically, (16)
01 01 00 60 00 07, 11, 00, 00, InTxPdo[11]=D/I_Latch_Low,update cyclically, (17)
01 05 01 00 FF 00, 04, 01, 00, Clear D/I Latch, update OutRxPdo[04].bit0 rising, (18)
STOP
=====
OutRxPdo[00]=261OCTLO, OutRxPdo[01]=261OCTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=261OSYS0, InTxPdo[01]=261OSYS1, InTxPdo[02..FF]=Out[02..FF]
=====
    
```

圖 A2-21

以 M-7050 模組為範例，

M-7050 模組 DI 支援 Latch High\_pulse 及 Low\_pulse，也可清除 Latch High/Low\_pulse 為 0。如果 OutRxPDO[04] 的 bit0 從 0 變為 1 (上升)，bit0 將會被清除為 0。

下表為 High\_Latch 及 Low\_Latch 的相對應位址：

DI Latch	PDO[位址]	清除 Latch
High_Latch	InTxPDO[10]	OutRxPDO[04].bit0
Low_Latch	InTxPDO[11]	

## 06.Initial\_Value

06\_Initial\_Value 資料夾提供 Power-on value 設定範例，詳細說明如下：

### Init\_Value\_1.txt

Init\_Value\_1.txt 提供 8 個 Modbus 命令範例，是使用特殊功能碼 “01” 來設定 Power-on value 功能，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
8, eight commands(00-07), max=300, format=Dec
02 06 00 00 01 23, 02, 00, 01, D/A_0=OutRxPdo[02]+initial=0x0123, (00)
02 06 00 01 02 34, 03, 00, 01, D/A_1=OutRxPdo[03]+initial=0x0234, (01)
02 06 00 02 03 45, 04, 00, 01, D/A_2=OutRxPdo[04]+initial=0x0325, (02)
02 06 00 03 04 56, 05, 00, 01, D/A_3=OutRxPdo[05]+initial=0x0456, (03)
02 03 00 40 00 01, 02, 00, 00, InTxPdo[02]=D/A_0 read back, update cyclically, (04)
02 03 00 41 00 01, 03, 00, 00, InTxPdo[03]=D/A_1 read back, update cyclically, (05)
02 03 00 42 00 01, 04, 00, 00, InTxPdo[04]=D/A_2 read back, update cyclically, (06)
02 03 00 43 00 01, 05, 00, 00, InTxPdo[05]=D/A_3 read back, update cyclically, (07)
STOP
=====
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
=====
if (OutRxPdo[0].bit0=0) then D/A=initial value
if (OutRxPdo[0].bit0=1) DA_0/1/2/3=OutRxPdo[02/03/04/05]
```

圖 A2-22

如果 (OutRxPDO[00].bit0 = 0) · 則 DA0/1/2/3 = initial value

如果 (OutRxPDO[00].bit0=1) · 則 DA0/1/2/3 = OutRxPDO[02/03/04/05]

下表為初始值 (initial value) 及 Readback 的相對應位址：

DA channel	OutRxPDO[00].bit0 = 0	OutRxPDO[00].bit0 = 1	Readback
	Initial Value	OutRxPDO[位址]	InTxPDO[位址]
0	0x0123	OutRxPDO[02]	InTxPDO[02]
1	0x0234	OutRxPDO[03]	InTxPDO[03]
2	0x0345	OutRxPDO[04]	nTxPDO[04]
3	0x0456	OutRxPDO[05]	nTxPDO[05]

當 ECAT-2610 上電開機時，**OutRxPDO[00]=2610CTRL0.bit0** 被重置為 **0**。

所以 DA0/1/2/3 輸出是設定的初始值 (initial value)，如下：

DA0 = 0x0123

DA1 = 0x0234

DA2 = 0x0345

DA3 = 0x0456

再將 OutRxPDO[02/03/04/05] 設定新的值。

然後 **OutRxPDO[00]=2610CTRL0.bit0** 設定為 **1** 將新設定值更新到 DA0/1/2/3，如下：

DA0 = OutRxPDO[02]

DA1 = OutRxPDO[03]

DA2 = OutRxPDO[04]

DA3 = OutRxPDO[05]

如果 **OutRxPDO[00]=2610CTRL0.bit0** 設為 **0**，則 DA0/1/2/3 將再次被設定為初始值 (initial value)。

 **注意**

---

詳細關於 **OutRxPDO[00].bit0 (2610CTRL0)** 資訊，請參考 [章節 3.5.1 “模組狀態及錯誤模式”](#)

---

## 07.Swap\_Byte\_Word

07\_Swap\_Byte\_Word 資料夾提供交換 (Swap) 功能設定範例，詳細說明如下：

### Both\_Swap\_1.txt

Both\_Swap\_1.txt 提供 2 個 Modbus 命令範例，是使用特殊功能碼 “06” 來設定 Both Swap 功能，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
2, two commands(00-01), max=300, format=Dec
02 10 00 00 00 04 08 12 34 12 34 12 34 12 34, 02, 00, 06, DA_0/1/2/3=OutRxPdo[2/3/4/5], both swap, (00)
02 03 00 40 00 04, 02, 00, 06, InTxPdo[2/3/4/5]=DA_0/1/2/3 read back, both swap , (01)
STOP
```

---

```
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
```

---

```
input=ABCD, both swap=DCBA
```

圖 A2-23

命令數量= 2,

Modbus 命令, PDO[位址], 更新模式, 特殊碼 =

(00) 02 10 00 00 00 04 08 12 34 12 34 12 34 12 34, 02, 00, 06, →設定 DA0/1/2/3 = OutRxPDO[02/03/04/05] 並做 Both Swap。

(01) 02 03 00 40 00 04, 02, 00, 06, → 讀取 DA0/1/2/3 值，然後將讀取值寫到 InTxPDO[02/03/04/05] 並做 Both Swap。

當輸入 **ABCD** 這 4 bytes 的字，在執行 both swap 之後，這 4 bytes 的字將變為 **DCBA**。

當 **OutRxPDO[02]=0xAB** 及 **OutRxPDO[03]=0xCD**，在執行 Both swap 之後，

結果為 **DA0 = 0xDC** 及 **DA1 = 0xBA**。

## Byte\_Swap\_1.txt

The **Byte\_Swap\_1.txt** 提供 8 個 Modbus 命令範例，是使用特殊功能碼 “02” 來設定 Byte Swap 功能，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
8, eight commands(00-07), max=300, format=Dec
02 06 00 00 00 00, 02, 00, 02, D/A_0=OutRxPdo[02], byte swap, (00)
02 06 00 01 00 00, 03, 00, 02, D/A_1=OutRxPdo[03], byte swap, (01)
02 06 00 02 00 00, 04, 00, 02, D/A_2=OutRxPdo[04], byte swap, (02)
02 06 00 03 00 00, 05, 00, 02, D/A_3=OutRxPdo[05], byte swap, (03)
02 03 00 40 00 01, 02, 00, 02, InTxPdo[02]=D/A_0 read back, byte swap, (04)
02 03 00 41 00 01, 03, 00, 02, InTxPdo[03]=D/A_1 read back, byte swap, (05)
02 03 00 42 00 01, 04, 00, 02, InTxPdo[04]=D/A_2 read back, byte swap, (06)
02 03 00 43 00 01, 05, 00, 02, InTxPdo[05]=D/A_3 read back, byte swap, (07)
STOP
-----
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
-----
input=ABCD, byte swap=BADC
```

圖 A2-24

命令數量= 8,

**Modbus 命令, PDO[位址], 更新模式, 特殊碼 =**

- (00) **02 06 00 00 00 00, 02, 00, 02** → 設定 DA0= OutRxPDO[02] 並做 Byte Swap。
- (01) **02 06 00 01 00 00, 03, 00, 02** → 設定 DA1= OutRxPDO[03] 並做 Byte Swap。
- (02) **02 06 00 02 00 00, 04, 00, 02** → 設定 DA2= OutRxPDO[04] 並做 Byte Swap。
- (03) **02 06 00 03 00 00, 05, 00, 02** → 設定 DA3= OutRxPDO[05] 並做 Byte Swap。
- (04) **02 03 00 40 00 01, 02, 00, 02** → 讀取 DA0 值，然後將讀取值寫到 InTxPDO[02] 並做 Byte Swap。
- (05) **02 03 00 41 00 01, 03, 00, 02** → 讀取 DA1 值，然後將讀取值寫到 InTxPDO[03] 並做 Byte Swap。
- (06) **02 03 00 42 00 01, 04, 00, 02** → 讀取 DA2 值，然後將讀取值寫到 InTxPDO[04] 並做 Byte Swap。
- (07) **02 03 00 43 00 01, 05, 00, 02** → 讀取 DA3 值，然後將讀取值寫到 InTxPDO[05] 並做 Byte Swap。

當輸入 **ABCD** 這 4 bytes 的字，在執行 byte swap，這 4 bytes 的字將變為 **BADC**。

當 **OutRxPDO[02]=0xAB** 及 **OutRxPDO[03]=0xCD**，在執行 byte swap 之後，

結果為 **DA0 = 0xBA** 及 **DA1 = 0xDC**。

## Word\_Swap\_1.txt

**Word\_Swap\_1.txt** 提供 2 個 Modbus 命令範例，是使用特殊功能碼 “04” 來設定 Word Swap 功能，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
2, two commands(00-01), max=300, format=Dec
02 10 00 00 00 04 08 12 34 12 34 12 34 12 34, 02, 00, 04, DA_0/1/2/3=OutRxPdo[2/3/4/5], word swap, (00)
02 03 00 40 00 04, 02, 00, 04, InTxPdo[2/3/4/5]=DA_0/1/2/3 read back, word swap, (01)
STOP
```

---

```
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
```

---

```
input=ABCD, word swap=CDAB
```

圖 A2-25

命令數量= 2,

**Modbus 命令, PDO[位址], 更新模式, 特殊碼 =**

(00) **02 10 00 00 00 04 08 12 34 12 34 12 34 12 34, 02, 00, 04** → 設定 DA0/1/2/3 = OutRxPDO[02/03/04/05] 並做 Word Swap。

(01) **02 03 00 40 00 04, 02, 00, 04** → 讀取 DA0/1/2/3 值，然後將讀取值寫到 InTxPDO[02/03/04/05] 並做 Word Swap。

當輸入 **ABCD** 這 4 bytes 的字，在執行 word swap，這 4 bytes 的字將變為 **CDAB**。

當 **OutRxPDO[02]=0xAB** 及 **OutRxPDO[03]=0xCD**，在執行 byte swap 之後，結果為 **DA0 = 0xCD** 及 **DA1 = 0xAB**。



## 08.State\_Change\_Trigger

**08\_State\_Change\_Trigger** 資料夾提供狀態改變觸發 (State Change Trigger) 功能設定範例·詳細說明如下:

### State\_Change\_1.txt

**State\_Change\_1.txt** 提供 4 個 Modbus 命令範例·是使用特殊功能碼 **“08”** 來設定 AO0 ~ AO3 狀態改變觸發 (State Change Trigger) 功能·如下圖所示:

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
4, four commands(00-03), max=300, format=Dec
02 06 00 00 00 00, 02, 00, 08, D/A_0=OutRxPdo[02]+state change update, (00)
02 06 00 01 00 00, 03, 00, 08, D/A_1=OutRxPdo[03]+state change update, (01)
02 06 00 02 00 00, 04, 00, 08, D/A_2=OutRxPdo[04]+state change update, (02)
02 06 00 03 00 00, 05, 00, 08, D/A_3=OutRxPdo[05]+state change update, (03)
STOP

-----
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
-----
D/A_0/1/2/3 will update when OutRxPdo[02/03/04/05] change
```

圖 A2-26

以 DA0 = OutRxPDO[02] 為範例·  
 當 OutRxPDO[02] 有變改·則 **“02 06 00 00 00 00”** 命令將會傳送給模組。  
 當 OutRxPDO[02] 相同沒變·則 **“02 06 00 00 00 00”** 命令將被 Bypass。

下表為 DA 相對應位址:

DA channel	PDO[位址]
0	OutRxPDO[02]
1	OutRxPDO[03]
2	OutRxPDO[04]
3	OutRxPDO[05]

## State\_Change\_2.txt

State\_Change\_2.txt 提供 1 個 Modbus 命令範例，是使用特殊功能碼 “08” 來設定全部 AO 狀態改變觸發 (State Change Trigger) 功能，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
1, eight commands(00-01), max=300, format=Dec
02 10 00 00 00 04 08 12 34 12 34 12 34 12 34, 02, 00, 08, DA_0/1/2/3=OutRxPdo[2/3/4/5]+state change update, (00)
STOP

-----
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
-----
D/A_0/1/2/3 will update when OutRxPdo[02/03/04/05] change
```

圖 A2-27

當 OutRxPDO[02/03/04/05] 有變改，則 “02 10 00 00 00 04 08 12 34 12 34 12 34 12 34” 命令將會傳送給模組。

否則 “02 10 00 00 00 04 08 12 34 12 34 12 34 12 34” 命令將被 Bypass。

下表為 DA 相對應位址：

DA channel	PDO[位址]
0	OutRxPDO[02]
1	OutRxPDO[03]
2	OutRxPDO[04]
3	OutRxPDO[05]

## 09.Constant\_Output

09\_Constant\_Output 資料夾提供常數輸出功能設定範例，詳細說明如下：

### Constant\_1.txt

Constant\_1.txt 提供 4 個 Modbus 命令範例，是使用特殊功能碼 “10” 來設定 AO0 ~ AO3 常數輸出功能，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
4, four commands(00-03), max=300, format=Dec
02 06 00 00 01 23, 02, 00, 10, D/A_0=constant 0x0123, update cyclically, (00)
02 06 00 01 02 34, 03, 00, 10, D/A_1=constant 0x0234, update cyclically, (01)
02 06 00 02 03 45, 04, 00, 10, D/A_2=constant 0x0345, update cyclically, (02)
02 06 00 03 04 56, 05, 00, 10, D/A_3=constant 0x0456, update cyclically, (03)
STOP
```

---

```
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
```

圖 A2-28

下表為 DA 相對應位址：

DA channel	PDO[位址]	常數輸出
0	OutRxPDO[02]	0x0123
1	OutRxPDO[03]	0x0234
2	OutRxPDO[04]	0x0345
3	OutRxPDO[05]	0x0456

## 10.Bit\_Command

10\_Bit\_Command 資料夾提供設定 Bit 命令範例，詳細說明如下：

### Bit\_Cmd\_1.txt

Bit\_Cmd\_1.txt 提供 8 個 Modbus 命令範例為 Write DO Bit0 ~ Bit7，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
8, 8 commands(00-07), max=300, format=Dec
01 05 00 00 FF 00, 02, 00, 00, D/O.bit0=OutRxPdo[02].bit0, update cyclically, (00)
01 05 00 01 FF 00, 02, 00, 00, D/O.bit1=OutRxPdo[02].bit1, update cyclically, (01)
01 05 00 02 FF 00, 02, 00, 00, D/O.bit2=OutRxPdo[02].bit2, update cyclically, (02)
01 05 00 03 FF 00, 02, 00, 00, D/O.bit3=OutRxPdo[02].bit3, update cyclically, (03)
01 05 00 04 FF 00, 02, 00, 00, D/O.bit4=OutRxPdo[02].bit4, update cyclically, (04)
01 05 00 05 FF 00, 02, 00, 00, D/O.bit5=OutRxPdo[02].bit5, update cyclically, (05)
01 05 00 06 FF 00, 02, 00, 00, D/O.bit6=OutRxPdo[02].bit6, update cyclically, (06)
01 05 00 07 FF 00, 02, 00, 00, D/O.bit7=OutRxPdo[02].bit7, update cyclically, (07)
STOP
-----
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
```

圖 A2-29

以 M-7050 模組為範例，

M-7050 模組支援 8 個 Digital Output 通道，下表為相對應位址：

DO	PDO[位址]
Bit0	OutRxPDO[02].bit0
Bit1	OutRxPDO[02].bit1
Bit2	OutRxPDO[02].bit2
Bit3	OutRxPDO[02].bit3
Bit4	OutRxPDO[02].bit4
Bit5	OutRxPDO[02].bit5
Bit6	OutRxPDO[02].bit6
Bit7	OutRxPDO[02].bit7

## 11.Delay\_Command

11\_Dealy\_Command 資料夾提供設定 Delay Time 範例，詳細說明如下：

### Delay\_Cmd\_1.txt

Delay\_Cmd\_1.txt 和 [DIO\\_DA\\_AD\\_1.txt](#) 很相似，除了 Delay Time，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
6, six commands(00-05), max=300, format=Dec
01 0F 00 00 00 08 01 00, 02, 00, 00, D/O=OutTxPdo[2], update cyclically, (00)
01 02 00 00 00 08, 02, 00, 00, InTxPdo[02]=D/I, update cyclically, (01)
FF 06 00 01 00 01, 02, 00, 00, delay 1ms x 1=1ms, D/A need more delay @ 115.2K (02)
02 10 00 00 00 04 08 12 34 12 34 12 34 12 34, 03, 00, 00, DA_0/1/2/3=OutRxPdo[3/4/5/6], update cyclically, (03)
FF 06 00 01 00 02, 02, 00, 00, delay 1ms x 2=2ms, A/D need more delay @ 115.2K (04)
03 04 00 00 00 08, 03, 00, 00, InTxPdo[3/0A]=A/I_0~7, update cyclically, (05)
STOP
```

---

```
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
```

---

```
DIO Address = 0x01
D/A Address = 0x02
A/D Address = 0x03
```

圖 A2-30

命令(00)、(01)、(03)、(05) 詳細說明請參考 [DIO\\_DA\\_AD\\_1.txt](#)

**Modbus 命令, PDO[位址], 更新模式, 特殊碼 =**

(02) **FF 06 00 01 00 01, 02, 00, 00** → 設定 Delay 1 ms = 1 ms x 1

(04) **FF 06 00 01 00 02, 02, 00, 00** → 設定 Delay 2 ms = 1 ms x 2

如果模組需要設定延遲時間，可參考下面的 Delay 命令來設定：

**FF 06 00 00 00 XX**: 單位 = 0.01 sec = 10 ms，最大 = 255 x 0.01 sec = 2.55 sec

**FF 06 00 01 00 XX**: 單位 = 1 ms = 0.001 sec，最大 = 255 x 0.001 sec = 0.255 sec

## 12.TxPdo\_RxPdo\_0x80\_0xFF

**12\_TxPdo\_RxPdo\_0x80\_0xFF** 資料夾提供設定 InTxPDO/OutRxPDO 位址為 0x80 ~ 0xFF 的範例，詳細說明如下：

### TxPdo\_RxPdo\_0x80.txt

**TxPdo\_RxPdo\_0x80.txt** 提供 2 個 Modbus 命令範例，設定 DI 及 DO 的 InTxPDO/OutRxPDO 位址為 0x80，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
2, two commands(00-01), max=300, format=Dec
01 0F 00 00 00 08 01 00, 80, 00, 00, D/O=OutTxPdo[80], update cyclically, (00)
01 02 00 00 00 08, 80, 00, 00, InTxPdo[80]=D/I, update cyclically, (01)
STOP
```

---

```
OutRxPdo[00]=261OCTL0, OutRxPdo[01]=261OCTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=261SYS0, InTxPdo[01]=261SYS1, InTxPdo[02..FF]=Out[02..FF]
```

圖 A2-31

### TxPdo\_RxPdo\_0xFF.txt

**TxPdo\_RxPdo\_0xFF.txt** 提供 2 個 Modbus 命令範例，設定 DI 及 DO 的 InTxPDO/OutRxPDO 位址為 0xFF，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
2, two commands(00-01), max=300, format=Dec
01 0F 00 00 00 08 01 00, FF, 00, 00, D/O=OutTxPdo[FF], update cyclically, (00)
01 02 00 00 00 08, FF, 00, 00, InTxPdo[FF]=D/I, update cyclically, (01)
STOP
```

---

```
OutRxPdo[00]=261OCTL0, OutRxPdo[01]=261OCTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=261SYS0, InTxPdo[01]=261SYS1, InTxPdo[02..FF]=Out[02..FF]
```

圖 A2-32

## TxPdo\_RxPdo\_AD\_0x80.txt

**TxPdo\_RxPdo\_AD\_0x80.txt** 提供 8 個 Modbus 命令範例，設定 AI0 ~ AI7 的 InTxPDO 位址為 0x80 ~ 0x87，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
8, 8 commands(00-07), max=300, format=Dec
03 04 00 00 00 01, 80, 00, 00, InTxPdo[80]=A/I_0, update cyclically, (00)
03 04 00 01 00 01, 81, 00, 00, InTxPdo[81]=A/I_1, update cyclically, (01)
03 04 00 02 00 01, 82, 00, 00, InTxPdo[82]=A/I_2, update cyclically, (02)
03 04 00 03 00 01, 83, 00, 00, InTxPdo[83]=A/I_3, update cyclically, (03)
03 04 00 04 00 01, 84, 00, 00, InTxPdo[84]=A/I_4, update cyclically, (04)
03 04 00 05 00 01, 85, 00, 00, InTxPdo[85]=A/I_5, update cyclically, (05)
03 04 00 06 00 01, 86, 00, 00, InTxPdo[86]=A/I_6, update cyclically, (06)
03 04 00 07 00 01, 87, 00, 00, InTxPdo[87]=A/I_7, update cyclically, (07)
STOP
```

---

```
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
```

圖 A2-33

## TxPdo\_RxPdo\_AD\_0xFF.txt

**TxPdo\_RxPdo\_AD\_0xFF.txt** 提供 1 個 Modbus 命令範例，設定 AI0 ~ AI7 的 InTxPDO 位址為 0xF8 ~ 0xFF，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
1, 1 commands(00-00), max=300, format=Dec
03 04 00 00 00 08, F8, 00, 00, InTxPdo[F8/FF]=A/I_0~7, update cyclically, (00)
STOP
```

---

```
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
```

圖 A2-34

## TxPdo\_RxPdo\_DA\_0x80\_0xFF.txt

**TxPdo\_RxPdo\_AD\_0xFF.txt** 提供 5 個 Modbus 命令範例，設定 AO0 ~ AO3 的 InTxPDO/OutTxPDO 位址為 0x80 ~ 0xFF，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
5, four commands(00-04), max=300, format=Dec
02 03 00 40 00 02, 80, 00, 00, InTxPdo[80/81]=DA_0/1 read back, update cyclically, (00)
02 03 00 42 00 02, FE, 00, 00, InTxPdo[FE/FF]=DA_2/3 read back, update cyclically, (01)
02 10 00 00 00 02 04 12 34 12 34, 80, 00, 00, DA_0/1=OutTxPdo[80/81], update cyclically, (02)
02 06 00 02 00 00, FE, 00, 00, DA_2=OutTxPdo[FE], update cyclically, (03)
02 06 00 03 00 00, FF, 00, 00, DA_3=OutTxPdo[FF], update cyclically, (04)
STOP
```

---

```
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
```

圖 A2-35

## 13.Commands\_128\_202

**Commands\_128\_202** 資料夾提供設定 128 及 202 行命令列範例 (commands\_128.txt 及 commands\_202.txt)。

### ⚠ 注意

配置檔中最大的命令數量是 300，但模組 EEPROM 最大是 2047。

在配置範例 commands\_202.txt 裡，最大命令數量是 202，如果增加第 203 行命令，則 EEPROM 將會 FULL。



## 14.End\_of\_Cmd\_Dealy

**14\_End\_of\_Cmd\_Dealy** 資料夾提供設定 Modbus 命令結束時的 Delay Time 範例，詳細說明如下：

### End\_Delay\_1.txt

**End\_Dealy\_1.txt** 提供設定 end\_delay 2 秒，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
200, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, delay=200x0.01=2 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
2, two commands(00-01), max=300, format=Dec
01 0F 00 00 00 08 01 00, 02, 00, 00, D/O=OutTxPdo[2], update cyclically, (00)
01 02 00 00 00 08, 02, 00, 00, InTxPdo[02]=D/I, update cyclically, (01)
STOP
```

---

```
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
```

圖 A2-36

設定 end\_delay 為 200，單位 = 0.01 sec，所以 200 x 0.01 sec = 2 sec。

ECAT-2610 將在每個 Modbus 命令結束時延遲 2 秒，這將使 Modbus 命令掃描速度變慢方便進行偵錯。對於一般正常應用，建議將 end\_delay 設為 0。

## 15.TxPdo\_RxPdo\_Max

15\_TxPdo\_RxPdo\_Max 資料夾提供設定 InTxPdoMax/2 及 OutRxPdoMax/2 範例，詳細說明如下：

### ⚠ 注意

InTxPdoMax/2 及 OutRxPdoMax/2 是用於偵錯。

建議將 InTxPdoMax/2 及 OutRxPdoMax/2 設為 0 (自動設定)。

### TxRxPdo\_Max\_1.txt

TxRxPdo\_Max\_1.txt 提供設定 InTxPdoMax/2 = 8 及 OutRxPdoMax/2 = 10，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
8, InTxPdoMax/2, format=Dec, InTxPdoMax=8X2=16=Sys0 ~ In0F
10, OutRxPdoMax/2, format=Dec, OutRxPdoMax=10X2=20=Ctrl0 ~ Out13
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
2, two commands(00-01), max=300, format=Dec
01 0F 00 00 00 08 01 00, 02, 00, 00, D/I=OutTxPdo[2], update cyclically, (00)
01 02 00 00 00 08, 02, 00, 00, InTxPdo[02]=D/I, update cyclically, (01)
STOP
=====
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
=====
min InTxPdoMax = 10
min OutRxPdoMax = 10
```

圖 A2-37

設定 InTxPdoMax/2 為 8，所以 InTxPdoMax = 8 x 2 = 16，表示 InTxPDO[00] ~ InTxPDO[0F]。

設定 OutRxPdoMax/2 為 10，所以 OutRxPdoMax = 10 x 2 = 20，表示 OutRxPDO[00] ~ OutRxPDO[13]。

## TxRxPdo\_Max\_2.txt

**TxRxPdo\_Max\_2.txt** 提供設定  $\text{InTxPdoMax}/2 = 64$  及  $\text{OutRxPdoMax}/2 = 64$  · 如下圖所示:

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
64, InTxPdoMax/2, format=Dec, InTxPdoMax=64x2=128=Sys0 ~ In7F
64, OutRxPdoMax/2, format=Dec, OutRxPdoMax=64x2=128=Ctrl0 ~ Out7F
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
2, two commands(00-01), max=300, format=Dec
01 0F 00 00 00 08 01 00, 02, 00, 00, D/O=OutTxPdo[2], update cyclically, (00)
01 02 00 00 00 08, 02, 00, 00, InTxPdo[02]=D/I, update cyclically, (01)
STOP
```

---

```
OutRxPdo[00]=261OCTL0, OutRxPdo[01]=261OCTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=261OSYS0, InTxPdo[01]=261OSYS1, InTxPdo[02..FF]=Out[02..FF]
```

---

```
min InTxPdoMax = 10
min OutRxPdoMax = 10
```

圖 A2-38

設定  $\text{InTxPdoMax}/2$  為 64 · 所以  $\text{InTxPdoMax} = 64 \times 2 = 128$  · 表示  $\text{InTxPDO}[00] \sim \text{InTxPDO}[7F]$  ·  
 設定  $\text{OutRxPdoMax}/2$  為 64 · 所以  $\text{OutRxPdoMax} = 64 \times 2 = 128$  · 表示  $\text{OutRxPDO}[00] \sim \text{OutRxPDO}[7F]$  ·

## TxRxPdo\_Max\_3.txt

**TxRxPdo\_Max\_3.txt** 提供設定  $\text{InTxPdoMax}/2 = 128$  及  $\text{OutRxPdoMax}/2 = 128$  · 如下圖所示:

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
128, InTxPdoMax/2, format=Dec, InTxPdoMax=128x2=256=Sys0 ~ InFF
128, OutRxPdoMax/2, format=Dec, OutRxPdoMax=128x2=256=Ctrl0 ~ OutFF
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
2, two commands(00-01), max=300, format=Dec
01 0F 00 00 00 08 01 00, 02, 00, 00, D/O=OutTxPdo[2], update cyclically, (00)
01 02 00 00 00 08, 02, 00, 00, InTxPdo[02]=D/I, update cyclically, (01)
STOP
```

---

```
OutRxPdo[00]=261OCTL0, OutRxPdo[01]=261OCTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=261OSYS0, InTxPdo[01]=261OSYS1, InTxPdo[02..FF]=Out[02..FF]
```

---

```
min InTxPdoMax = 10
min OutRxPdoMax = 10
```

圖 A2-39

設定  $\text{InTxPdoMax}/2$  為 128 · 所以  $\text{InTxPdoMax} = 128 \times 2 = 256$  · 表示  $\text{InTxPDO}[00] \sim \text{InTxPDO}[FF]$  ·  
 設定  $\text{OutRxPdoMax}/2$  為 128 · 所以  $\text{OutRxPdoMax} = 128 \times 2 = 256$  · 表示  $\text{OutRxPDO}[00] \sim \text{OutRxPDO}[FF]$  ·

## 16.Rs485\_Cycle\_Time

**16\_Rs485\_Cycle\_Time** 資料夾提供設定特殊命令 “**FF 03 00 01 00 01**” 來存取 RS-485 Modbus RTU 設備的週期時間 (Cycle Time) 範例，詳細說明如下：

### Rs485\_Cycle\_Time\_1.txt

**Rs485\_Cycle\_Time\_1.txt** 和 [DIO Addr01\\_1.txt](#) 很相似，除了 RS-485 cycle time，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, delay=200x0.01=2 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
2, two commands(00-01), max=300, format=Dec
01 0F 00 00 00 08 01 00, 02, 00, 00, D/O=OutTxPdo[2], update cyclically, (00)
FF 03 00 01 00 01, 02, 00, 00, InTxPdo[2]=Rs485_Cycle_Time, unit=0.1ms (01)
STOP
```

---

```
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=In[02..FF]
```

圖 A2-40

命令數量= 2,

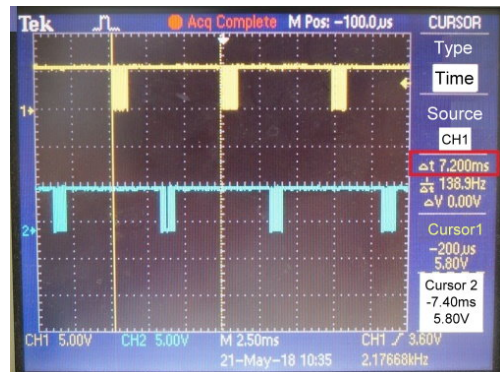
**Modbus 命令, PDO[位址], 更新模式, 特殊碼=**

(00) **01 0F 00 00 00 08 01 00, 02, 00, 00**, → 讀取 OutRxPDO[02] 並傳送 Modbus 命令到 DO 模組。

(01) **FF 03 00 01 00 01, 02, 00, 00**, → 存取 RS-485 週期時間，單位: 0.1 ms

下圖為使用 TwinCAT 及示波器實際量測的比對圖示。

Name	Online	Type	Size	>Addr...	In/Out	User...
2610SYS0	0x0000	UINT	2.0	26.0	Input	0
2610SYS1	0xa000	UINT	2.0	28.0	Input	0
In02	72	UINT	2.0	30.0	Input	0
In03	0	UINT	2.0	32.0	Input	0
In04	0	UINT	2.0	34.0	Input	0
In05	0	UINT	2.0	36.0	Input	0
In06	0	UINT	2.0	38.0	Input	0
In07	0	UINT	2.0	40.0	Input	0
In08	0	UINT	2.0	42.0	Input	0
In09	0	UINT	2.0	44.0	Input	0



## Rs485\_Cycle\_Time\_2.txt

Rs485\_Cycle\_Time\_2.txt 和 [DIO Addr01\\_2.txt](#) 很相似，除了 RS-485 cycle time，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, delay=200x0.01=2 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
3, three commands(00-02), max=300, format=Dec
01 0F 00 00 00 08 01 00, 02, 00, 00, D/O=OutTxPdo[02], update cyclically, (00)
01 02 00 00 00 08, 02, 00, 00, InTxPdo[02]=D/I, update cyclically, (01)
FF 03 00 01 00 01, 03, 00, 00, InTxPdo[03]=Rs485_Cycle_Time, unit=0.1ms (02)
STOP
```

---

```
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=In[02..FF]
```

圖 A2-41

命令數量= 3,

**Modbus 命令, PDO[位址], 更新模式, 特殊碼=**

- (00) 01 0F 00 00 00 08 01 00, 02, 00, 00, → 讀取 OutRxPDO[02] 並傳送 Modbus 命令到 DO 模組。
- (01) 01 02 00 00 00 08, 02, 00, 00, → 讀取模組的 DI，然後將讀取值寫到 InTxPDO[02]。
- (02) FF 03 00 01 00 01, 02, 00, 00, → 存取 RS-485 週期時間，單位: 0.1 ms

下圖為使用 TwinCAT 及示波器實際量測的比對圖示。

Name	Online	Type	Size	>Addr...	In/Out	User ...
2610SYS0	0x0000	UINT	2.0	26.0	Input	0
2610SYS1	0xa000	UINT	2.0	28.0	Input	0
In02	0	UINT	2.0	30.0	Input	0
In03	122	UINT	2.0	32.0	Input	0
In04	0	UINT	2.0	34.0	Input	0
In05	0	UINT	2.0	36.0	Input	0
In06	0	UINT	2.0	38.0	Input	0
In07	0	UINT	2.0	40.0	Input	0
In08	0	UINT	2.0	42.0	Input	0
In09	0	UINT	2.0	44.0	Input	0



## 17.Ext\_Sync

17\_Ext\_Sync 資料夾提供如何使用 Ext\_Sync 機制範例，詳細說明如下：

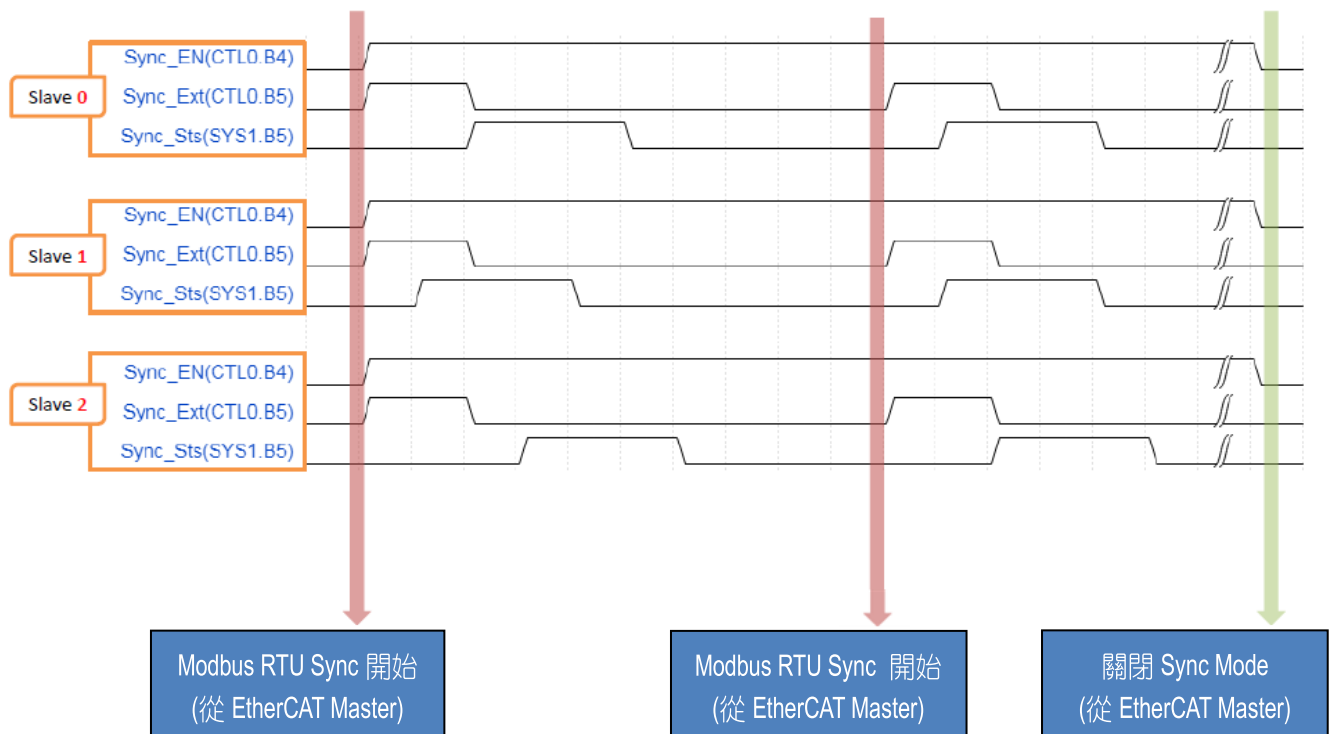
Ext\_Sync 機制可用來同步多個 ECAT-2610 模組，運作定義如下：

- 主機使用 **2610CTL0.Bit4 = High** 表示啟用 Ext\_Sync 機制
- 主機使用 **2601CTL0.Bit5 = High** 表示 Ext\_Sync 就緒
- ECTA-2610 使用 **2610SYS1.Bit5 = High** 表示執行 **Ext\_Sync\_commands**
- ECAT-2610 使用 **2610SYS1.Bit5 = Low** 表示結束 **Ext\_Sync\_commands**
- 用戶設定 **CtrlX[0]** 來表示開始的 **Ext\_Sync\_commands**

### ⚠ 注意

詳細關於 **2610CTL0.Bit4**、**2610CTL0.Bit5** 及 **2610SYS1.Bit5**，請參考 [第 3.3.1 節 “模組狀態及錯誤模式”](#)。

下圖範例顯示三個 slave 設備的 Ext\_Sync 運作：



## ext\_sync.txt

**Ext\_Sync.txt** 提供 5 個 Modbus 命令範例，在此範例中，**CtrlX[0] = 3** 表示從命令 (03) 開始為 **Ext\_Sync\_commands**，因此，命令(00) ~ (02) 是 **Normal\_commands**，它們將始終依序被執行，而命令(03) ~ (04) 是 **Ext\_Sync\_commands**，當 **Ext\_Sync (2610CTL0.Bit5)** 為 High 時，命令(3)、(4) 才會被執行，當 **Ext\_Sync (2610CTL0.Bit5)** 為 Low 時，命令(3)、(4) 則被跳過不行執，如下圖所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
3 0 0 0 0 0, 7 Reserved, format=Hex, default=0, CtrlX[0]=03=Ext_Sync_Start
5, five commands(00-04), max=300, format=Dec
01 01 00 00 00 08, 02, 00, 00, InTxPdo[02]=D/I read back, update cyclically, (00)
01 02 00 00 00 08, 03, 00, 00, InTxPdo[03]=D/I, update cyclically, (01)
01 01 00 40 00 07, 04, 00, 00, InTxPdo[04]=D/I_Latch_High, update cyclically, (02)
01 0F 00 00 00 08 01 00, 02, 00, 00, D/I=OutTxPdo[2], update cyclically, (03)
01 01 00 00 00 08, 05, 00, 00, InTxPdo[05]=D/I read back, update cyclically, (04)
STOP
```

圖 A2-42

當 **2610CTL0.Bit5 = Low** 執行順序如下：

命令(00) → (01) → (02) → (00) → (01) → (02) → ..... → (00) → (01) → (02) → .....

當 **2610CTL0.Bit5 = High** 執行順序如下：

命令(00) → (01) → (02) → (03) → (04) → ..... → (00) → (01) → (02) → (03) → (04) → .....

➤ 配置 **Ext\_Sync** 機制步驟如下：

- 步驟 1: 將所有 **Normal\_commands** 寫在 Modbus 命令配置區最前面
- 步驟 2: 將所有 **Ext\_Sync\_commands** 寫在 Modbus 命令配置區最後面
- 步驟 3: 設定 **CtrlX[0]** 來指定 **Ext\_Sync\_commands** 的起始值
- 步驟 4: 上傳配置檔 (commands.txt) 到 ECAT-2610 模組。

➤ 控制 **Ext\_Sync** 運作的步驟如下：

- 步驟 1: 主機設定 **2610CTL0.Bit4 = High** 來啟用 **Ext\_Sync** 機制
- 步驟 2: 主機設定 **2610CTL0.Bit5 = High** 來開始 **Ext\_Sync** 運作

- 步驟 3:** 主機等待 2610SYS1.Bit5 = High  
(ECAT-2610 設定 2610SYS1.Bit5 = High 表示執行 Ext\_Sync\_commands)
- 步驟 4:** 主機設定 2610CTL0.Bit5 = Low 來停止 Ext\_Sync 運作  
(ECAT-2610 設定 2610SYS1.Bit5 = Low 表示停止 Ext\_Sync\_commands)
- 步驟 5:** 主機等待 2610SYS1.Bit5 = Low
- 步驟 6:** .....
- 步驟 7:** 返回 Step 2 進行下一個 Ext\_Sync 運作
- 步驟 8:** 主機設定 2610CTL0.Bit4 = Low 來關閉 Ext\_Sync 機制
- 步驟 9:** 結束

此外，在 **Ext\_Sync.txt** 配置檔下面包含了 2610CTL0.Bit4、2610CTL0.Bit5、2610SYS1.Bit5 及 CtrlX[0] 的定義說明，如下圖所示：

```

OutRxPdo[00]=2610CTL0
Host set 2610CTL0.bit4=High to Enable the Ext_Sync mechanism
Host set 2610CTL0.bit5=High to Start one Ext_Sync operation
Host set 2610CTL0.bit5=Low to Stop this Ext_Sync operation

InTxPdo[01]=2610SYS1
2610 set 2610SYS1.bit5=High to indicate Ext_Sync_commands are executed
2610 set 2610SYS1.bit5=Low to indicate Ext_Sync_commands are end

===== assume CtrlX[0]=Sync_Start=03 =====
CtrlX[0]=Sync_Start=3 --> normal scan = 00,01,02,00,01,02,....,00,01,02,...
CtrlX[0]=Sync_Start=3 --> Ext_Sync scan = 00,01,02,03,04,00,01,02,03,04,....,00,01,02,03,04...
[00,01,02] = normal_commands = always scan
[03,04] = Ext_Sync_commands = scan when 2610CTRL0.bit5 is High
when [03,04] are executed, 2610 will set 2610SYS1.bit5 to High
when [03,04] are end, 2610 will set 2610SYS1.bit5 to Low

===== assume CtrlX[0]=Sync_Start=04 =====
CtrlX[0]=Sync_Start=4--> normal scan = 00,01,02,03,00,01,02,03,....,00,01,02,03,...
CtrlX[0]=Sync_Start=4--> Ext_Sync scan = 00,01,02,03,04,00,01,02,03,04,....,00,01,02,03,04...
[00,01,02,03] = normal_commands = always scan
[04] = Ext_Sync_command = scan when 2610CTRL0.bit5 is High
when [04] are executed, 2610 will set 2610SYS1.bit5 to High
when [04] are end, 2610 will set 2610SYS1.bit5 to Low
    
```



## A3. ECAT-2610-DW 配置檔參考文件

在 7188ECAT 資料夾 (配置/偵錯工具) 中包含 **more commands** 資料夾，裡面提供了許多關於讀取 Power Meter、讀取系統、設定參數...等配置檔 (commands.txt) 範例，您可參考這些範例來有效正確的設定配置檔 (commands.txt)，詳細說明如下。

### ⚠ 注意

此配置參考文件僅適用於 PM-3033、PM-3133、PM-3114、PM-3112 系列智能電錶。

➤ 開啟 **more commands** 資料夾，內容如下：

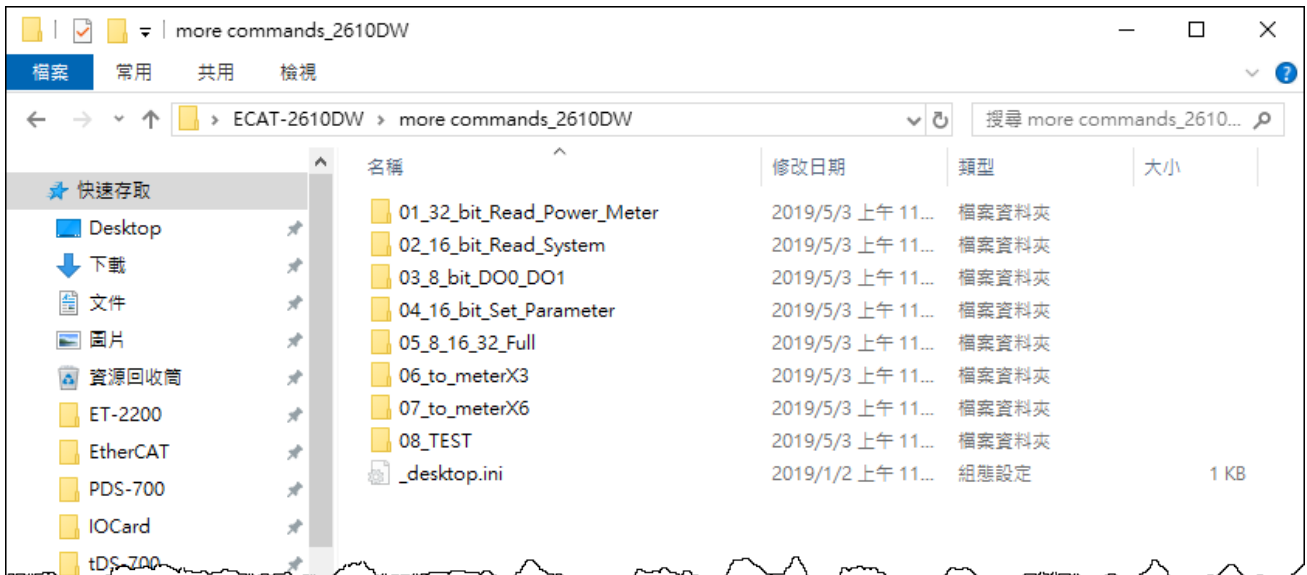


圖 A3-1

## 01.32\_bit\_Read\_Power\_Meter

可使用 Factory Utility 來配置 PM-3133 系列智能電錶。 **01\_32\_bit\_bit\_Read\_Power\_Meter.txt** 是讀取所有三相電源資料到 EtherCAT 主機。

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 2 0 0 0 0, 7 Reserved, format=Hex, default=0, CtrlX[1]=2:display_value=IEEE 754 * 100.0
8, 8 commands(00-07), max=300, format=Dec
04 04 11 00 00 12, 01, 00, 20, InTxPdo[01..09]=V_a to KVAh_a, IEEE 754, update cyclically, (00)
04 04 11 48 00 02, 0A, 00, 20, InTxPdo[0A]=Freq_a, IEEE 754, update cyclically, (01)
04 04 11 12 00 12, 0B, 00, 20, InTxPdo[0B..13]=V_b to KVAh_b, IEEE 754, update cyclically, (02)
04 04 11 4A 00 02, 14, 00, 20, InTxPdo[14]=Freq_b, IEEE 754, update cyclically, (03)
04 04 11 24 00 12, 15, 00, 20, InTxPdo[15..1D]=V_c to KVAh_c IEEE 754, update cyclically, (04)
04 04 11 4C 00 02, 1E, 00, 20, InTxPdo[1E]=Freq_c, IEEE 754, update cyclically, (05)
04 04 11 36 00 12, 1F, 00, 20, InTxPdo[1F..27]=V_avg to KVAh_tot, IEEE 754, update cyclically, (06)
04 04 11 4E 00 02, 28, 00, 20, InTxPdo[28]=Freq_max, IEEE 754, update cyclically, (07)
STOP
-----
OutRxPdo[00]=CTL, OutRxPdo[01..7F]=Out[01..7F]
InTxPdo[00]=SYS, InTxPdo[01..7F]=In[01..7F]
-----
32_bit commands

if (CtrlX[1]==1) display_value = IEEE 754 * 10.0
if (CtrlX[1]==2) display_value = IEEE 754 * 100.0
if (CtrlX[1]==3) display_value = IEEE 754 * 1000.0
else display_value = IEEE 754

address of power_meter = 04
```

PM-3133 系列的單位資料格式是 IEEE 754。

**CmdX.bit5 = High** 是用來啟用轉換 IEEE 754。

**CtrlX[1]** 是用來調整讀取數值，如下：

如 (CtrlX[1]==1) display\_value = IEEE 754 \* 10.0

如 (CtrlX[1]==2) display\_value = IEEE 754 \* 100.0

如 (CtrlX[1]==3) display\_value = IEEE 754 \* 1000.0

否則 display\_value = IEEE 754

## 02.16\_bit\_Read\_System

**02\_16\_bit\_read\_system.txt** 命令檔包含 5 個 Modbus 命令，為讀取系統資訊，如下所示：

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
5, 5 commands(00-04), max=300, format=Dec
04 04 02 00 00 01, 29, 00, 00, InTxPdo[29]=wiring, update cyclically, (00)
04 04 02 01 00 01, 2A, 00, 00, InTxPdo[2A]=phase, update cyclically, (01)
04 04 02 02 00 01, 2B, 00, 00, InTxPdo[2B]=name, update cyclically, (02)
04 04 02 03 00 01, 2C, 00, 00, InTxPdo[2C]=type, update cyclically, (03)
04 04 02 04 00 01, 2D, 00, 00, InTxPdo[2D]=ver, update cyclically, (04)
STOP
```

---

```
OutRxPdo[00]=CTL, OutRxPdo[01..7F]=Out[01..7F]
InTxPdo[00]=SYS, InTxPdo[01..7F]=In[01..7F]
```

---

```
16_bit commands

address of power_meter = 04
```

## 03.8\_bit\_D00\_D01

**D00\_D01.txt** 及 **D00\_D01\_2.txt** 命令檔包含 3/2 個 Modbus 命令，為 Write DO Bit0 ~ Bit1，如下所示。

### D00\_D01.txt

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
3, three commands(00-02), max=300, format=Dec
04 05 10 00 00 00, 1, 00, 00, D/O_0=OutTxPdo[1].bit0, update cyclically, (00)
04 05 10 01 00 00, 1, 00, 00, D/O_1=OutTxPdo[1].bit1, update cyclically, (01)
04 01 10 00 00 02, 2E, 00, 00, InRxPdo[2E].bit0/bit1=D/O_0/1, update cyclically, (02)
STOP
```

---

```
OutRxPdo[00]=CTL, OutRxPdo[01..7F]=Out[01..7F]
InTxPdo[00]=SYS, InTxPdo[01..7F]=In[01..7F]
```

---

```
8_bit commands

address of power_meter = 04
```

## D00\_D01\_2.txt

```

START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
2, three commands(00-01), max=300, format=Dec
04 0F 10 00 00 02 01 00, 1, 00, 00, D/O_0/1=OutTxPdo[1].bit0/bit1, update cyclically, (00)
04 01 10 00 00 02, 2E, 00, 00, InRxPdo[2E].bit0/bit1=D/O_0/1, update cyclically, (01)
STOP
=====
OutRxPdo[00]=CTL, OutRxPdo[01..7F]=Out[01..7F]
InTxPdo[00]=SYS, InTxPdo[01..7F]=In[01..7F]
=====
8_bit commands
address of power_meter = 04

```

## 04.16\_bit\_Set\_Parameter

可使用 Factory Utility 來配置 PM-3133 系列智能電錶。04\_16\_bit\_set\_parameter.txt 是更新 PM-3133 系列智能電錶的 EEPROM，因此建議使用狀態改變更新模式 (State Change Update Mode)。

```

START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
4, 4 commands(00-03), max=300, format=Dec
04 03 10 03 00 01, 2F, 00, 00, InTxPdo[2F]=PT_ratio, update cyclically, (00)
04 03 10 04 00 01, 30, 00, 00, InTxPdo[30]=CT_ratio, update cyclically, (01)
04 06 10 03 00 00, 02, 00, 08, PT_ratio=OutRxPdo[2], state change update(02)
04 06 10 04 00 00, 03, 00, 08, CT_ratio=OutRxPdo[3], state change update(03)
STOP
=====
OutRxPdo[00]=CTL, OutRxPdo[01..7F]=Out[01..7F]
InTxPdo[00]=SYS, InTxPdo[01..7F]=In[01..7F]
=====
16_bit commands
set PT_ratio will save to EEPROM --> can not use update cyclically --> must use state change update
set CT_ratio will save to EEPROM --> can not use update cyclically --> must use state change update
address of power_meter = 04

```

## 05.8\_16\_32\_Full

**05\_8\_16\_32\_Full.txt** 命令檔包含 01\_32\_bit\_Read\_Power\_Meter、02\_16\_bit\_Read\_System、03\_8\_bit\_DO0\_DO1 及 04\_16\_bit\_Set\_Parameter 配置命令檔，如下所示。

```

START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 2 0 0 0 0, 7 Reserved, format=Hex, default=0, Ctrl[1]=2:display_value=read_value * 100.0
19, 19 commands(00-18), max=300, format=Dec
04 04 11 00 00 12, 01, 00, 20, InTxPdo[01..09]=V_a to KVAh_a, IEEE 754, update cyclically, (00)
04 04 11 48 00 02, 0A, 00, 20, InTxPdo[0A]=Freq_a, IEEE 754, update cyclically, (01)
04 04 11 12 00 12, 0B, 00, 20, InTxPdo[0B..13]=V_b to KVAh_b, IEEE 754, update cyclically, (02)
04 04 11 4A 00 02, 14, 00, 20, InTxPdo[14]=Freq_b, IEEE 754, update cyclically, (03)
04 04 11 24 00 12, 15, 00, 20, InTxPdo[15..1D]=V_c to KVAh_c IEEE 754, update cyclically, (04)
04 04 11 4C 00 02, 1E, 00, 20, InTxPdo[1E]=Freq_c, IEEE 754, update cyclically, (05)
04 04 11 36 00 12, 1F, 00, 20, InTxPdo[1F..27]=V_avg to KVAh_tot, IEEE 754, update cyclically, (06)
04 04 11 4E 00 02, 28, 00, 20, InTxPdo[28]=Freq_max, IEEE 754, update cyclically, (07)

04 04 02 00 00 01, 29, 00, 00, InTxPdo[29]=wiring, update cyclically, (08)
04 04 02 01 00 01, 2A, 00, 00, InTxPdo[2A]=phase, update cyclically, (09)
04 04 02 02 00 01, 2B, 00, 00, InTxPdo[2B]=name, update cyclically, (10)
04 04 02 03 00 01, 2C, 00, 00, InTxPdo[2C]=type, update cyclically, (11)
04 04 02 04 00 01, 2D, 00, 00, InTxPdo[2D]=ver, update cyclically, (12)

04 0F 10 00 00 02 01 00, 1, 00, 00, D/O_0/1=OutTxPdo[1].bit0/bit1, update cyclically, (13)
04 01 10 00 00 02, 2E, 00, 00, InRxPdo[2E].bit0/bit1=D/O_0/1, update cyclically, (14)

04 03 10 03 00 01, 2F, 00, 00, InTxPdo[2F]=PT_ratio, update cyclically, (15)
04 03 10 04 00 01, 30, 00, 00, InTxPdo[30]=CT_ratio, update cyclically, (16)
04 06 10 03 00 00, 02, 00, 08, PT_ratio=OutRxPdo[2], state change update(17)
04 06 10 04 00 00, 03, 00, 08, CT_ratio=OutRxPdo[3], state change update(18)
STOP
=====
OutRxPdo[00]=CTL, OutRxPdo[01..7F]=Out[01..7F]
InTxPdo[00]=SYS, InTxPdo[01..7F]=In[01..7F]
=====
8_bit + 16_bit + 32_bit
(00) - (07) : read_power_meter
(08) - (12) : read_system
(13) - (14) : set_DO0_DO1
(15) - (18) : set_parameter

address of power_meter = 04

```

## 06.to\_meterX3

**06\_to\_MeterX3.txt** 命令檔是出廠預設的命令檔，包含讀取 3 顆電錶到 EtherCAT 主機，如下所示。

```

START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 2 0 0 0 0 0, 7 Reserved, format=Hex, default=0, CtrlX[1]=2:display_value=IEEE 754 * 100.0
25, 25 commands(00-24), max=300, format=Dec
04 04 11 00 00 12, 01, 00, 20, InTxPdo[01..09]=V_a to KVAh_a, IEEE 754, update cyclically, (00)
04 04 11 48 00 02, 0A, 00, 20, InTxPdo[0A]=Freq_a, IEEE 754, update cyclically, (01)
04 04 11 12 00 12, 0B, 00, 20, InTxPdo[0B..13]=V_b to KVAh_b, IEEE 754, update cyclically, (02)
04 04 11 4A 00 02, 14, 00, 20, InTxPdo[14]=Freq_b, IEEE 754, update cyclically, (03)
04 04 11 24 00 12, 15, 00, 20, InTxPdo[15..1D]=V_c to KVAh_c IEEE 754, update cyclically, (04)
04 04 11 4C 00 02, 1E, 00, 20, InTxPdo[1E]=Freq_c, IEEE 754, update cyclically, (05)
04 04 11 36 00 12, 1F, 00, 20, InTxPdo[1F..27]=V_avg to KVAh_tot, IEEE 754, update cyclically, (06)
04 04 11 4E 00 02, 28, 00, 20, InTxPdo[28]=Freq_max, IEEE 754, update cyclically, (07)

05 04 11 00 00 12, 29, 00, 20, InTxPdo[29..31]=V_a to KVAh_a, IEEE 754, update cyclically, (08)
05 04 11 48 00 02, 32, 00, 20, InTxPdo[32]=Freq_a, IEEE 754, update cyclically, (09)
05 04 11 12 00 12, 33, 00, 20, InTxPdo[33..3B]=V_b to KVAh_b, IEEE 754, update cyclically, (10)
05 04 11 4A 00 02, 3C, 00, 20, InTxPdo[3C]=Freq_b, IEEE 754, update cyclically, (11)
05 04 11 24 00 12, 3D, 00, 20, InTxPdo[3D..45]=V_c to KVAh_c IEEE 754, update cyclically, (12)
05 04 11 4C 00 02, 46, 00, 20, InTxPdo[46]=Freq_c, IEEE 754, update cyclically, (13)
05 04 11 36 00 12, 47, 00, 20, InTxPdo[47..49]=V_avg to KVAh_tot, IEEE 754, update cyclically, (14)
05 04 11 4E 00 02, 50, 00, 20, InTxPdo[50]=Freq_max, IEEE 754, update cyclically, (15)

06 04 11 00 00 12, 51, 00, 20, InTxPdo[51..59]=V_a to KVAh_a, IEEE 754, update cyclically, (16)
06 04 11 48 00 02, 5A, 00, 20, InTxPdo[5A]=Freq_a, IEEE 754, update cyclically, (17)
06 04 11 12 00 12, 5B, 00, 20, InTxPdo[5B..63]=V_b to KVAh_b, IEEE 754, update cyclically, (18)
06 04 11 4A 00 02, 64, 00, 20, InTxPdo[64]=Freq_b, IEEE 754, update cyclically, (19)
06 04 11 24 00 12, 65, 00, 20, InTxPdo[65..6D]=V_c to KVAh_c IEEE 754, update cyclically, (20)
06 04 11 4C 00 02, 6E, 00, 20, InTxPdo[6E]=Freq_c, IEEE 754, update cyclically, (21)
06 04 11 36 00 12, 6F, 00, 20, InTxPdo[6F..77]=V_avg to KVAh_tot, IEEE 754, update cyclically, (22)
06 04 11 4E 00 02, 78, 00, 20, InTxPdo[78]=Freq_max, IEEE 754, update cyclically, (23)

FF 03 00 01 00 01, 7F, 00, 00, InTxPdo[7F]=Rs485_Cycle_Time, unit=0.lms (24)
STOP
=====
OutRxPdo[00]=CTL, OutRxPdo[01..7F]=Out[01..7F]
InTxPdo[00]=SYS, InTxPdo[01..7F]=In[01..7F]
=====
32_bit commands

if (CtrlX[1]==1) display_value = IEEE 754 * 10.0
if (CtrlX[1]==2) display_value = IEEE 754 * 100.0
if (CtrlX[1]==3) display_value = IEEE 754 * 1000.0
else display_value = IEEE 754

address of power_meter_1 = 04
address of power_meter_2 = 05
address of power_meter_3 = 06

```

## 07.to\_meterX6

**07\_to\_MeterX6.txt** 命令檔包含讀取 6 顆電錶到 EtherCAT 主機，如下所示。

```

START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 2 0 0 0 0 0, 7 Reserved, format=Hex, default=0, CtrlX[1]=2:display_value=IEEE 754 * 100.0
73, 73 commands(00-72), max=300, format=Dec
04 04 11 00 00 02, 01, 00, 20, InTxPdo[01]=V_a, IEEE 754, update cyclically, (00)
04 04 11 0C 00 06, 02, 00, 20, InTxPdo[02..04]=kWh_a ~ kVAh_a, IEEE 754, update cyclically, (01)
04 04 11 48 00 02, 05, 00, 20, InTxPdo[05]=Freq_a, IEEE 754, update cyclically, (02)
04 04 11 12 00 02, 06, 00, 20, InTxPdo[06]=V_b, IEEE 754, update cyclically, (03)
04 04 11 1E 00 06, 07, 00, 20, InTxPdo[07..09]=kWh_b ~ kVAh_b, IEEE 754, update cyclically, (04)
04 04 11 4A 00 02, 0A, 00, 20, InTxPdo[0A]=Freq_b, IEEE 754, update cyclically, (05)
04 04 11 24 00 02, 0B, 00, 20, InTxPdo[0B]=V_c, IEEE 754, update cyclically, (06)
04 04 11 30 00 06, 0C, 00, 20, InTxPdo[0C..0E]=kWh_c ~ kVAh_c, IEEE 754, update cyclically, (07)
04 04 11 4C 00 02, 0F, 00, 20, InTxPdo[0F]=Freq_c, IEEE 754, update cyclically, (08)
04 04 11 36 00 02, 10, 00, 20, InTxPdo[10]=V_c, IEEE 754, update cyclically, (09)
04 04 11 42 00 06, 11, 00, 20, InTxPdo[11..13]=kWh_c ~ kVAh_c, IEEE 754, update cyclically, (10)
04 04 11 4E 00 02, 14, 00, 20, InTxPdo[14]=Freq_c, IEEE 754, update cyclically, (11)

05 04 11 00 00 02, 15, 00, 20, InTxPdo[15]=V_a, IEEE 754, update cyclically, (12)
05 04 11 0C 00 06, 16, 00, 20, InTxPdo[16..18]=kWh_a ~ kVAh_a, IEEE 754, update cyclically, (13)
05 04 11 48 00 02, 19, 00, 20, InTxPdo[19]=Freq_a, IEEE 754, update cyclically, (14)
05 04 11 12 00 02, 1A, 00, 20, InTxPdo[1A]=V_b, IEEE 754, update cyclically, (15)
05 04 11 1E 00 06, 1B, 00, 20, InTxPdo[1B..1D]=kWh_b ~ kVAh_b, IEEE 754, update cyclically, (16)
05 04 11 4A 00 02, 1E, 00, 20, InTxPdo[1E]=Freq_b, IEEE 754, update cyclically, (17)
05 04 11 24 00 02, 1F, 00, 20, InTxPdo[1F]=V_c, IEEE 754, update cyclically, (18)
05 04 11 30 00 06, 20, 00, 20, InTxPdo[20..22]=kWh_c ~ kVAh_c, IEEE 754, update cyclically, (19)
05 04 11 4C 00 02, 23, 00, 20, InTxPdo[23]=Freq_c, IEEE 754, update cyclically, (20)
05 04 11 36 00 02, 24, 00, 20, InTxPdo[24]=V_c, IEEE 754, update cyclically, (21)
05 04 11 42 00 06, 25, 00, 20, InTxPdo[25..27]=kWh_c ~ kVAh_c, IEEE 754, update cyclically, (22)

FF 03 00 01 00 01, 7F, 00, 00, InTxPdo[7F]=Rs485_Cycle_Time, unit=0.1ms (72)
STOP
=====
OutRxPdo[00]=CTL, OutRxPdo[01..7F]=Out[01..7F]
InTxPdo[00]=SYS, InTxPdo[01..7F]=In[01..7F]
=====
32_bit commands

if (CtrlX[1]==1) display_value = IEEE 754 * 10.0
if (CtrlX[1]==2) display_value = IEEE 754 * 100.0
if (CtrlX[1]==3) display_value = IEEE 754 * 1000.0
else display_value = IEEE 754

address of power_meter_1 = 04
address of power_meter_2 = 05
address of power_meter_3 = 06
address of power_meter_4 = 07
address of power_meter_5 = 08
address of power_meter_6 = 09

```

## 08.TEST

**08\_TEST.txt** 命令檔包含讀取 V\_a、I\_a、kW\_a、kvar\_a、kVA\_a、PF\_a、kWh\_a、kvarh\_a、kVAh\_a 及 Freq\_a 資料，以及測試 8-bit、16-bit 及 32-bit，如下所示。

### TEST\_4A.txt

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
10, ten commands(00-09), max=300, format=Dec
04 04 11 00 00 02, 01, 00, 20, InTxPdo[1]=V_a, update cyclically, (00)
04 04 11 02 00 02, 02, 00, 20, InTxPdo[2]=I_a, update cyclically, (01)
04 04 11 04 00 02, 03, 00, 20, InTxPdo[3]=kW_a, update cyclically, (02)
04 04 11 06 00 02, 04, 00, 20, InTxPdo[4]=kvar_a, update cyclically, (03)
04 04 11 08 00 02, 05, 00, 20, InTxPdo[5]=kVA_a, update cyclically, (04)
04 04 11 0A 00 02, 06, 00, 20, InTxPdo[6]=PF_a, update cyclically, (05)
04 04 11 0C 00 02, 07, 00, 20, InTxPdo[7]=kWh_a, update cyclically, (06)
04 04 11 0E 00 02, 08, 00, 20, InTxPdo[8]=kvarh_a, update cyclically, (07)
04 04 11 10 00 02, 09, 00, 20, InTxPdo[9]=kVAh_a, update cyclically, (08)
04 04 11 48 00 02, 0A, 00, 20, InTxPdo[A]=Freq_a, update cyclically, (09)
STOP
```

---

```
OutRxPdo[00]=CTL, OutRxPdo[01..7F]=Out[01..7F]
InTxPdo[00]=SYS, InTxPdo[01..7F]=In[01..7F]
```

---

```
32_bit commands
```

### TEST\_5A\_8.txt

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
1, three commands(00-00), max=300, format=Dec
05 0F 10 00 00 08 01 00, 01, 00, 00, TEST 8_bit, update cyclically, (00)
STOP
```

---

```
OutRxPdo[00]=CTL, OutRxPdo[01..7F]=Out[01..7F]
InTxPdo[00]=SYS, InTxPdo[01..7F]=In[01..7F]
```

---



**TEST\_5B\_16.txt**

```

START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
1, three commands(00-00), max=300, format=Dec
05 OF 10 00 00 10 02 00 00, 01, 00, 00, TEST 16_bit, update cyclically, (00)
STOP
=====
OutRxPdo[00]=CTL, OutRxPdo[01..7F]=Out[01..7F]
InTxPdo[00]=SYS, InTxPdo[01..7F]=In[01..7F]
=====

```

**TEST\_5C\_32.txt**

```

START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
1, three commands(00-00), max=300, format=Dec
05 OF 10 00 00 20 04 00 00 00 00, 01, 00, 00, TEST_32_bit, update cyclically, (00)
STOP
=====
OutRxPdo[00]=CTL, OutRxPdo[01..7F]=Out[01..7F]
InTxPdo[00]=SYS, InTxPdo[01..7F]=In[01..7F]
=====

```

**TEST\_5D\_16.txt**

```

START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
1, three commands(00-00), max=300, format=Dec
05 10 10 00 00 01 02 00 00, 01, 00, 00, TEST_16_bit, update cyclically, (00)
STOP
=====
OutRxPdo[00]=CTL, OutRxPdo[01..7F]=Out[01..7F]
InTxPdo[00]=SYS, InTxPdo[01..7F]=In[01..7F]
=====

```

## TEST\_5E\_2.txt

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
1, three commands(00-00), max=300, format=Dec
05 10 10 00 00 02 04 00 00 00 00, 01, 00, 00, TEST_32_bit, update cyclically, (00)
STOP
```

---

```
OutRxPdo[00]=CTL, OutRxPdo[01..7F]=Out[01..7F]
InTxPdo[00]=SYS, InTxPdo[01..7F]=In[01..7F]
```

---

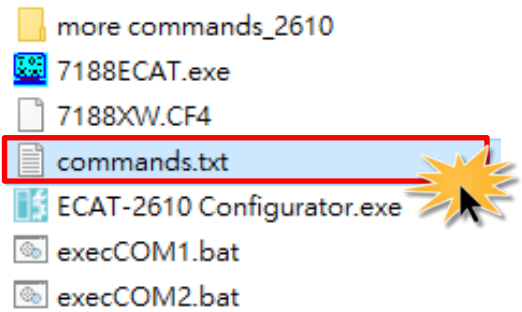
## A4.手動配置及上傳

本章節介紹出廠預設配置檔 (commands.txt) 格式，並可依照下面步驟來手動修改配置檔 (commands.txt)，然後使用配置/偵錯工具 (7188ECAT.exe) 將新的配置檔上傳至 ECAT-2610(-DW)/2611 模組中。

### A4-1 設定配置檔 (Commands.txt)

#### 修改 ECAT-2610 的配置檔

❶ 雙擊 “commands.txt” 來開啟配置數據檔。



commands.txt 配置檔是一個簡單控制 Modbus RTU 設備的指令集。語法規則由左到右，由冒號 “:” 分隔。詳細配置格式說明如下。

➤ ECAT-2610 模組出廠預設的配置數據檔如下：

- 紅色：命令設定區。
- 綠色：命令說明區。

```
commands.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, Timeout for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
3, three commands(00-02), max=300, format=Dec
FF 03 00 00 00 02, 02, 00, 00, InTxPdo[2]=InTxPdo[0]=Sys_Lo, InTxPdo[3]=InTxPdo[1]=Sys_Hi (00)
FF 06 00 00 00 64, 02, 00, 00, 0x64=100, delay 100X0.01 sec = 1 sec, (01)
FF 06 00 01 00 64, 02, 00, 00, 0x64=100, delay 100X1ms = 0.1 sec, (02)
STOP

OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
```

圖 A4-1

② 依據您的 Modbus RTU 設備來設定 Baud Rate、Parity 及 Stop Bit。

```

commands.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
    
```

圖 A4-2

參數	說明
Baud Rate	有效範圍: 1200/2400/4800/9600/19200/38400/57600/115200 bps。 (預設: 115200)
Parity	有效範圍: N (None), E (EVEN), O (ODD)。(預設: N)
Stop Bits	有效範圍: 1, 2。(預設: 1)

③ 設定 Timeout、Delay 時間及 InTxPDO/OutRxPDO 位址範圍。

```

commands.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
3, three commands(00-02), max=300, format=Dec
FF 03 00 00 00 02, 02, 00, 00, InTxPdo[2]=InTxPdo[0]=Sys_Lo, InTxPdo[3]=InTxPdo[1]=Sys_Hi (00)
00 00 00 00 02, 00, 00, 00, 0x64=100, 100, 01 sec
    
```

圖 A4-3

參數	說明
Timeout	此參數用來設定 Timeout 時間。設定範圍: 0 ~ 255。 單位= 0.01 秒 · 0.01 x 100 = 1 秒 · 最大= 2.55 秒。
End_delay Time	此參數用來設定 Delay 時間。設定範圍: 0 ~ 255。單位= 0.01 秒 · 最大= 2.55 秒。 詳細說明參考至 <a href="#">End_Delay_1.txt</a> 。
InTxPdoMax/2	InTxPdoMax 用於偵錯(Debug)。建議設定為 0 (自動設定) 詳細說明參考至 <a href="#">15.TxPdo_RxPdo_Max</a> 。
OutRxPdoMax/2	OutRxPdoMax 用於偵錯(Debug)。建議設定為 0 (自動設定) 詳細說明參考至 <a href="#">15.TxPdo_RxPdo_Max</a> 。
7 Reserved	第一個參數 (CtrlX[0]=Ext_Sync_Start) 用於設定 Ext_Sync_commands 的起始值。 其它六個參數是保留 (無作用)。詳細說明參考 <a href="#">17.Ext_Sync</a> 。

4 設定 Modbus 命令數量。

```

commands.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
3, three commands(00-02), max=300, format=Dec
FF 03 00 00 00 02, 02, 00, 00, InTxPdo[2]=InTxPdo[0]=Sys_Lo, InTxPdo[3]=InTxPdo[1]=Sys_Hi (00)
FF 00 00 00 02, 02, 00, 00, 0x64=100, delay 100x0.01 sec = 1 sec

```

圖 A4-5

參數	說明
3	此參數用來設定 Modbus 命令數量。設定範圍: 0 ~ 300 (最大)

5 依據您的 Modbus 設備來設定 Modbus RTU 命令。

```

commands.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
3, three commands(00-02), max=300, format=Dec
FF 03 00 00 00 02, 02, 00, 00, InTxPdo[2]=InTxPdo[0]=Sys_Lo, InTxPdo[3]=InTxPdo[1]=Sys_Hi (00)
FF 06 00 00 00 64, 02, 00, 00, 0x64=100, delay 100x0.01 sec = 1 sec, (01)
FF 06 00 01 00 64, 02, 00, 00, 0x64=100, delay 100x1ms = 0.1 sec, (02)
STOP

```

---

```

OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]

```

圖 A4-6

No.	Modbus 命令 (HEX)	說明
1	FF 03 00 00 00 02	此命令為出廠預設命令用來讀取 ECAT-2610 模組狀態，詳細參考 <a href="#">章節 3.3.1 “模組狀態及錯誤模式”</a> 。
	FF 06 00 00 00 64	此命令為出廠預設命令用來延遲 Modbus 命令。
	FF 06 00 01 00 64	此命令將減慢 Modbus 命令適用於偵錯，詳細參考 <a href="#">11. Delay Command</a> 。
		單位 = 0.01 sec · 延遲 0x64 (HEX) = 100 (DEC) · 100 (DEC) x 0.01 sec = 1 sec。 單位 = 1 ms = 0.001 sec · 延遲 0x64 (HEX) = 100 (DEC) · 100 (DEC) x 0.001 sec = 0.1 sec。

	<b>PDO [位址] (HEX)</b>	<b>說明</b>
②	02	此參數用來設定位址，將映射到 InTxPDO[位址] (輸入位址) 或 OutRxPDO[位址] (輸出位址)。有效設定範圍: 0x02 ~ 0xFF。 <b>注意:</b> OutRxPDO[00]、OutRxPDO[01]、InTxPDO[00]、InTxPDO[01] 是 ECAT-2610 模組系統所使用。
	<b>更新模式 (HEX)</b>	<b>說明</b>
③	00	此參數用來設定更新模式。更新模式是 8-bit 控制，詳細可參考至 <a href="#">05.Rising Trigger</a> 。 <b>00:</b> 命令將循環更新。 <b>≠00:</b> 命令將在 InTxPDO[位址]上升 (Rising Edge) 時更新。
	<b>特殊功能代 (HEX)</b>	<b>說明</b>
④	00	此參數用來設定特殊功能，包含 Power-on value、Swap、狀態改變觸發及常數輸出...等功能，詳細範例參考 <a href="#">06 Initial Value</a> 、 <a href="#">07 Swap Byte Word</a> 、 <a href="#">08 State Change Trigger</a> 及 <a href="#">09 Constant Output</a> 。  設定範圍: <b>00</b> (無，預設)、 <b>01</b> (Power-On value)、 <b>02</b> (byte-swap)、 <b>04</b> (word-swap)、 <b>06</b> (both-swap)、 <b>08</b> (狀態改變觸發)、 <b>10</b> (常數輸出)

➤ 這裡使用 M-7050 模組為範例，修改 Modbus 命令為 Write 8 個 DO 通道並儲存新設定檔，如下圖所示。

```
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
1, one commands(00-00), max=300, format=Dec
01 0F 00 00 00 08 01 00, 02, 00, 00, D/O=OutTxPdo[2], update cyclically, (00)
STOP
```

---

```
OutRxPdo[00]=2610CTL0, OutRxPdo[01]=2610CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2610SYS0, InTxPdo[01]=2610SYS1, InTxPdo[02..FF]=Out[02..FF]
```

圖 A4-7

<b>參數</b>	<b>說明</b>			
Modbus 命令數量	有效設定範圍: 0 ~ 300 (預設:1)			
<b>Modbus 命令 (HEX)</b>	<b>PDO [位址] (HEX)</b>	<b>更新模式 (HEX)</b>	<b>特殊功能代碼 (HEX)</b>	<b>說明</b>
01 0F 00 00 00 08 01 00	02	00	00	詳細關於 Modbus 命令，參考第 6 章 “Modbus 資訊”。




## 修改 ECAT-2610-DW 的配置檔

❶ “ECAT-2610DW\_Utl\_xxxxxx.zip” 可以從泓格科技網站下載。下載位置如下：

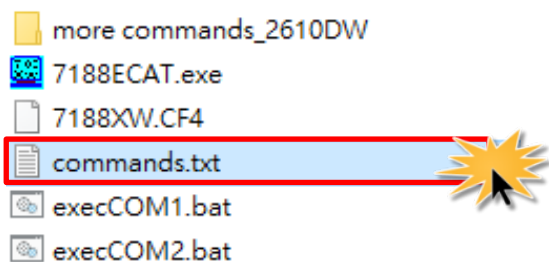
 [http://ftp.icpdas.com/pub/cd/fieldbus\\_cd/ethercat/slave/ecat-2000/software/](http://ftp.icpdas.com/pub/cd/fieldbus_cd/ethercat/slave/ecat-2000/software/)

❷ 將 ECAT-2610DW\_Utl\_xxxxxx.zip 解壓縮後，包含 7188ECAT 資料夾是配置/偵錯工具。

❸ 將 7188ECAT 資料夾複製到電腦硬碟上 (如: E:\)，資料夾中應包含下列檔案：

 more commands_2610DW	配置讀取電錶資料...等範例參考文件 詳細說明參考 <a href="#">A3. “ECAT-2610-DW 配置數據檔參考文件”</a>
 7188ECAT.exe	執行檔，詳細參考 <a href="#">A4-2 “上傳配置檔”</a>
 7188XW.CF4	7188ECAT.exe 控制檔
 commands.txt	配置檔，用來設定 Modbus RTU 設備的文件檔。ECAT-2610 將透過此配置檔來與 Modbus RTU 設備進行通訊
 execCOM1.bat	使用電腦主機的 COM1 來載入配置數據檔至 ECAT-2610，詳細參考 <a href="#">A4-2 “上傳配置檔”</a>
 execCOM2.bat	使用電腦主機的 COM2 來載入配置數據檔至 ECAT-2610，詳細參考 <a href="#">A4-2 “上傳配置檔”</a>

❹ 雙擊 “commands.txt” 來開啟配置數據檔。



commands.txt 配置檔是一個簡單控制 Modbus RTU 設備的指令集。語法規則由左到右，由冒號 “:” 分隔。詳細配置格式說明如下。

➤ ECAT-2610-DW 模組出廠預設的配置數據檔如下：

- 紅色: 命令設定區。
- 綠色: 命令說明區。

```

START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 2 0 0 0 0, 7 Reserved, format=Hex, default=0, CtrlX[1]=2:display_value=IEEE 754 * 100.0
25, 25 commands(00-24), max=300, format=Dec
04 04 11 00 00 12, 01, 00, 20, InTxPdo[01..09]=V_a to KVAh_a, IEEE 754, update cyclically, (00)
04 04 11 48 00 02, 0A, 00, 20, InTxPdo[0A]=Freq_a, IEEE 754, update cyclically, (01)
04 04 11 12 00 12, 0B, 00, 20, InTxPdo[0B..13]=V_b to KVAh_b, IEEE 754, update cyclically, (02)
04 04 11 4A 00 02, 14, 00, 20, InTxPdo[14]=Freq_b, IEEE 754, update cyclically, (03)
04 04 11 24 00 12, 15, 00, 20, InTxPdo[15..1D]=V_c to KVAh_c IEEE 754, update cyclically, (04)
04 04 11 4C 00 02, 1E, 00, 20, InTxPdo[1E]=Freq_c, IEEE 754, update cyclically, (05)
04 04 11 36 00 12, 1F, 00, 20, InTxPdo[1F..27]=V_avg to KVAh_tot, IEEE 754, update cyclically, (06)
04 04 11 4E 00 02, 28, 00, 20, InTxPdo[28]=Freq_max, IEEE 754, update cyclically, (07)
05 04 11 00 00 12, 29, 00, 20, InTxPdo[29..31]=V_a to KVAh_a, IEEE 754, update cyclically, (08)
05 04 11 48 00 02, 32, 00, 20, InTxPdo[32]=Freq_a, IEEE 754, update cyclically, (09)
05 04 11 12 00 12, 33, 00, 20, InTxPdo[33..3B]=V_b to KVAh_b, IEEE 754, update cyclically, (10)
    
```

圖 A4-8

② 依據您的 Power Meter 來設定 Baud Rate、Parity 及 Stop Bit。

```

START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
    
```

圖 A4-9

參數	說明
Baud Rate	有效範圍: 1200/2400/4800/9600/19200/38400/57600/115200 bps。 (預設: 115200)
Parity	有效範圍: N (None), E (EVEN), O (ODD)。(預設: N)
Stop Bits	有效範圍: 1, 2。(預設: 1)



③ 設定 Timeout、Delay 時間及 InTxPDO/OutRxPDO 位址範圍。

```

commands.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 2 0 0 0 0 0, 7 Reserved, format=Hex, default=0, CtrlX[1]=2:display_value=IEEE 754 * 100.0
25, 25 commands(00-24), max=300, format=Dec
04 04 11 00 00 12, 01, 00, 20, InTxPdo[01..09]=k a to KVAh_a, IEEE 754, update cyclically, (00)
    
```

圖 A4-10

參數	說明
Timeout	此參數用來設定 Timeout 時間。設定範圍: 0 ~ 255。 單位= 0.01 秒 · 0.01 x 100 = 1 秒 · 最大= 2.55 秒。
End_delay Time	此參數用來設定 Delay 時間。設定範圍: 0 ~ 255。單位= 0.01 秒 · 最大= 2.55 秒。 詳細說明參考至 <a href="#">End_Delay_1.txt</a> 。
InTxPdoMax/2	InTxPdoMax 用於偵錯(Debug)。建議設定為 0 (自動設定) 詳細說明參考至 <a href="#">15.TxPdo RxPdo Max</a> 。
OutRxPdoMax/2	OutRxPdoMax 用於偵錯(Debug)。建議設定為 0 (自動設定) 詳細說明參考至 <a href="#">15.TxPdo RxPdo Max</a> 。
7 Reserved	第二個參數 (CtrlX[1]=2) 是用來顯示 "IEEE 754 * 100.0" 值。PM-3133 系列的單位資料格式是 IEEE 754。

④ 設定 Modbus 命令數量。

```

commands.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 2 0 0 0 0 0, 7 Reserved, format=Hex, default=0, CtrlX[1]=2:display_value=IEEE 754 * 100.0
25, 25 commands(00-24), max=300, format=Dec
04 04 11 00 00 12, 01, 00, 20, InTxPdo[01..09]=k a to KVAh_a, IEEE 754, update cyclically, (00)
    
```

圖 A4-11

參數	說明
25	此參數用來設定 Modbus 命令數量。設定範圍: 0 ~ 300 (最大)

5 依據您的 Power Meter 來設定 Modbus RTU 命令。

```

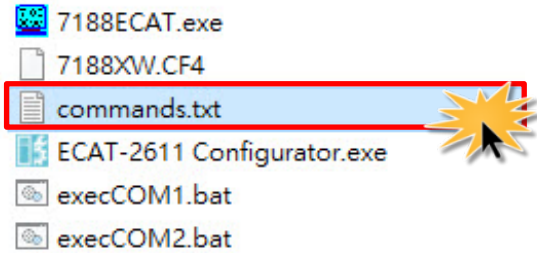
commands.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
0, delay in the end of Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, max=2.55 sec
0, InTxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0, OutRxPdoMax/2, format=Dec, valid=[0 ~ 128], default=0=Auto
0 2 0 0 0 0, 7 Reserved, format=Hex, default=0, CtrlX[1]=2:display_value=IEEE 754 * 100.0
25, 25 commands(00-24), max=300, format=Dec
04 04 11 00 00 12, 01, 00, 20, InTxPdo[01..09]=V_a to KVAh_a, IEEE 754, update cyclically, (00)
04 04 11 48 00 02, 0A, 00, 20, InTxPdo[0A]=Freq_a, IEEE 754, update cyclically, (01)
04 04 11 12 00 12, 0B, 00, 20, InTxPdo[0B..13]=V_b to KVAh_b, IEEE 754, update cyclically, (02)
04 04 11 4A 00 02, 14, 00, 20, InTxPdo[14]=Freq_b, IEEE 754, update cyclically, (03)
04 04 11 24 00 12, 15, 00, 20, InTxPdo[15..1D]=V_c to KVAh_c IEEE 754, update cyclically, (04)
04 04 11 4C 00 02, 1E, 00, 20, InTxPdo[1E]=Freq_c, IEEE 754, update cyclically, (05)
04 04 11 36 00 12, 1F, 00, 20, InTxPdo[1F..27]=V_avg to KVAh_tot, IEEE 754, update cyclically, (06)
04 04 11 4E 00 02, 28, 00, 20, InTxPdo[28]=Freq_max, IEEE 754, update cyclically, (07)
    
```

圖 A4-12

No.	Modbus 命令 (HEX)	說明
1	04 04 11 00 00 12 04 04 11 48 00 02 04 04 11 12 00 12 . .	詳細關於 Modbus 命令，參考第 6 章 “Modbus 資訊”。 詳細關於 PM-3000 系列配置通信或功能位址定義，請參考 PM-3000 系列使用手冊。
	PDO [位址] (HEX)	說明
2	01 0A 0B . .	此參數用來設定位址，將映射到 InTxPDO[位址] (輸入位址) 或 OutRxPDO[位址] (輸出位址)。有效設定範圍: 0x01 ~ 0x7F。 <b>注意:</b> OutRxPDO[00]、InTxPDO[00] 是 ECAT-2610-DW 模組系統所使用。
	更新模式 (HEX)	說明
3	00	此參數用來設定更新模式。更新模式是 8-bit 控制，詳細可參考至 <a href="#">05.Rising Trigger</a> 。 00: 命令將循環更新。 #00: 命令將在 InTxPDO[位址]上升 (Rising Edge) 時更新。
	特殊功能代 (HEX)	說明
4	20	此參數用來設定特殊功能。設定範圍: 00 (無)、08 (狀態改變觸發)、20 (IEEE 754，預設)

## 修改 ECAT-2611 的配置檔

❶ 雙擊 “commands.txt” 來開啟配置數據檔。



commands.txt 配置檔是一個簡單設定 ECAT-2611 模組 Net ID、Baud Rate 及 Data Format...等的指令集。語法規則由左到右，由冒號 “,” 分隔。詳細配置格式說明如下。

➤ ECAT-2611 模組出廠預設的配置數據檔如下：

■ 紅色：命令設定區。

■ 綠色：命令說明區。

```

commands.txt - Notepad
File Edit Format View Help
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
01, reserved, format=Dec, valid=[0 ~ 255]
128, InTxPdoMax/2, format=Dec, InTxPdoMax=8X2=16=Sys0 ~ In0F
128, OutRxPdoMax/2, format=Dec, OutRxPdoMax=10X2=20=Ctrl0 ~ Out13
0 0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
STOP

-----

OutRxPdo[00]=2611CTL0, OutRxPdo[01]=2611CTL1, OutRxPdo[02..FF]=Out[02..FF]
InTxPdo[00]=2611SYS0, InTxPdo[01]=2611SYS1, InTxPdo[02..FF]=Out[02..FF]

OutRxPdo[00] & OutRxPdo[01] are reserved for 2611 control
InTxPdo[00] & InTxPdo[01] are reserved for 2611 ststus

Use OutRxPdo[02..FF] & InTxPdo[02..FF] for normal operation

Cmd_06 & Cmd_10 can only write to InTxPdo[02..FF]

Cmd_03 can read OutRxPdo[00..FF] (OutRxPdo[00]=2611CTL0,OutRxPdo[01]=2611CTL1)
    
```

圖 A4-13

② 設定 **Baud Rate**、**Parity** 及 **Stop Bit** 值。

參數	說明
Baud Rate	有效範圍: 1200/2400/4800/9600/19200/38400/57600/115200 bps。 (預設: 115200)
Parity	有效範圍: N (None), E (EVEN), O (ODD)。(預設: N)
Stop Bits	有效範圍: 1, 2。(預設: 1)

```

commands.txt - Notepad
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
01, reserved, format=Dec, valid=[0 ~ 255]
    
```

圖 A4-14

③ 設定 **Timeout**、**Net ID**、**InTxPdoMax/2** 及 **OutRxPdoMax/2** 值。

參數	說明
100	此參數用來設定 Timeout 時間。設定範圍: 0 ~ 255。 單位= 0.01 秒 · 0.01 x 100 = 1 秒 · 最大= 2.55 秒。
Net ID	設定 Net ID (預設[DEC]: 1)。設定範圍 [DEC]: 0 ~ 255。
InTxPdoMax/2	InTxPdoMax = 8 X 2 = 16 = Sys0 ~ In0F (預設[DEC]: 128)。
OutRxPdoMax/2	OutRxPdoMax = 10 X 2 = 20 = Ctrl0 ~ Out13 (預設[DEC]: 128)。
7 Reserved	此參數是保留 (無作用) (預設[Hex]: 0 0 0 0 0 0)

```

commands.txt - Notepad
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
START
115200, baud rate, from 1200,2400 ~ 57600,115200
N, N=No Parity, E=EVEN, O=ODD
1, 1=one STOP bit, 2=two STOP bits
100, TimeOut for Modbus command, format=Dec, valid=[0 ~ 255], unit=0.01 sec, 0.01x100=1 sec, max=2.55 sec
01, reserved, format=Dec, valid=[0 ~ 255]
128, InTxPdoMax/2, format=Dec, InTxPdoMax=8X2=16=Sys0 ~ In0F
128, OutRxPdoMax/2, format=Dec, OutRxPdoMax=10X2=20=Ctrl0 ~ Out13
0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
STOP
    
```

圖 A4-15

## A4-2 上傳配置檔

### 步驟 1 將下載 Cable 連接至電腦主機及 ECAT-2610(-DW)/2611 模組

- 1 將 ECAT-2610(-DW)/2611 模組斷電關機。
- 2 將配件 CA-0915 Cable 連接至電腦主機上的 COM Port 及 ECAT-2610(-DW)/2611 模組上的 COM1 Port。

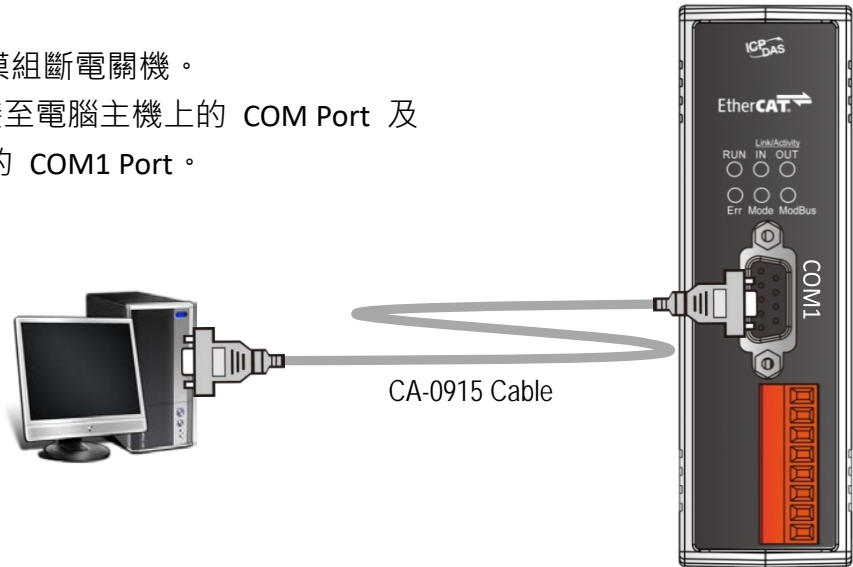


圖 A4-16

### 步驟 2 修改 execCOM1.bat 檔中的 COM Port 碼

此 COM Port 碼為您電腦所連接至 ECAT-2610(-DW)/2611 的 COM Port (如: 您 PC 上的 COM Port 為 COM4, 請修改 C1 為 C4)

#### ⚠ 注意

如果您電腦有 COM1 或 COM2 時, 可直接使用 execCOM1.bat 或 execCOM2.bat 檔案, 請跳過此步驟。



### 步驟 3 上傳配置檔(commands.txt) 到 ECAT-2610(-DW)/2611 模組

❶ 這裡以 Windows 10 系列為範例，在 Windows 搜尋欄位輸入“cmd”，按下“Enter”鍵來開啟命令提示字元視窗。

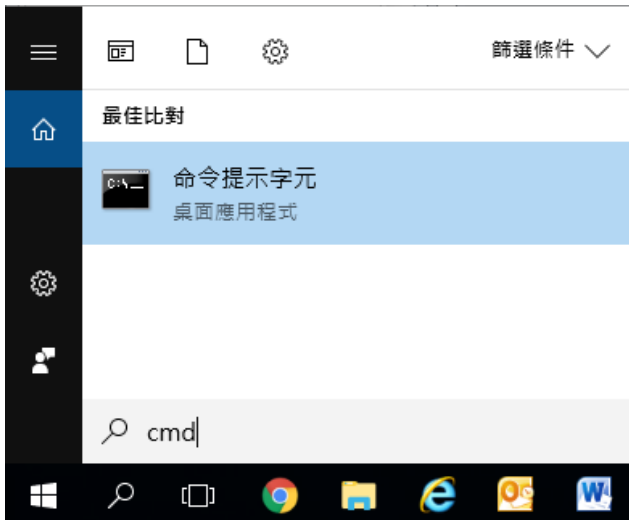


圖 A4-18

#### ⚠注意

開啟命令提示字元視窗的方法，請依據您使用的作業系統來執行

- ❷ 輸入 **E:** (此為 7188ECAT 資料夾所存放的磁碟槽區)，按 **Enter** 鍵。
- ❸ 輸入 **cd 7188ecat**，按 **Enter** 鍵來進入 7188ECAT 資料夾中。
- ❹ 輸入 **execcom4**，按 **Enter** 鍵來自動執行 **7188ECAT.exe** 程式。

#### ⚠注意

execCOM1.bat/execCOM2 是使用電腦 COM1/COM2 來載入配置檔。如果您電腦沒有 COM1 或 COM2，請參考圖 A4-17 來修改 COM Port 碼。

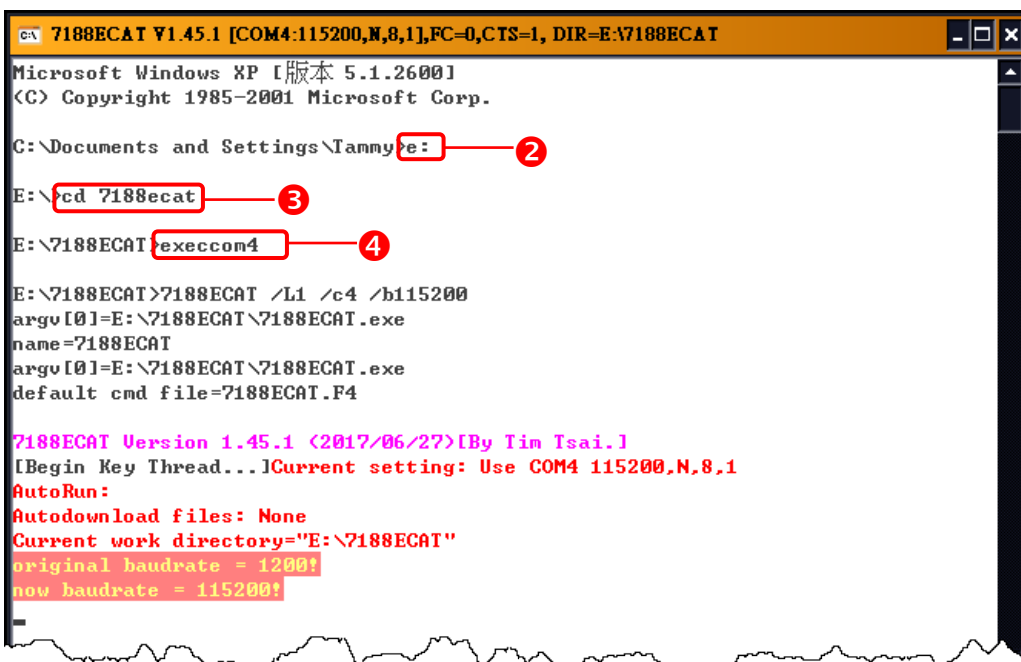


圖 A4-19

5 供電開機啟動 ECAT-2610(-DW)/2611 模組。

```
GA 7188ECAT V1.45.1 [COM4:115200,N,8,1],FC=0,CTS=1, DIR=E:\7188ECAT
COM1 TXD!=RXD --> Normal
*** Start Download from EEPROM ***
>> baudrate=10:115200, Parity=N, Stop=1, TimeOut=100, Delay=0, InMax=0, OutMax=0
, CmdNum=3
[000] : 008, FF 03 00 00 02 ,<02>, TxRx[02], Update=00, CmdX=00
[001] : 008, FF 06 00 00 64 ,<02>, TxRx[02], Update=00, CmdX=00
[002] : 008, FF 06 00 01 64 ,<02>, TxRx[02], Update=00, CmdX=00
CRC16, [Compute = eb A0], <EEP=eb a0> --> CRC16 OK
>> Load Configuration From EEPROM OK, 3 commands
EEPROM : InTxPdoMax++=3, OutRxPdoMax++=3
APP : InTxPdoMax=10, OutRxPdMaxo=10
===== ECAT-2610, Ver. 2.25 =====
Ctrl+F4: Download
READ1 : Read EEPROM <sequential>
READ2 : Read EEPROM <Command>
SHOW0 : Show Version Number
SHOW1 : Show System Status
SHOW2 : Show Input TxPdo
SHOW3 : Show Output RxPdo
SHOW4 : Show Debug Information
SHOW5 : Show Debug Information Step by Step
ERASE : ERASE EEPROM
```

在此會顯示出模組 EEPROM 中出廠預設的一組 Modbus 指令

圖 A4-20

6 輸入 erase ，再按 Enter 鍵來清除 EEPROM。

```
GA 7188ECAT V1.45.1 [COM4:115200,N,8,1],FC=0,CTS=1, DIR=E:\7188ECAT
CRC16, [Compute = eb A0], <EEP=eb a0> --> CRC16 OK
>> Load Configuration From EEPROM OK, 3 commands
EEPROM : InTxPdoMax++=3, OutRxPdoMax++=3
APP : InTxPdoMax=10, OutRxPdMaxo=10
===== ECAT-2610, Ver. 2.25 =====
Ctrl+F4: Download
READ1 : Read EEPROM <sequential>
READ2 : Read EEPROM <Command>
SHOW0 : Show Version Number
SHOW1 : Show System Status
SHOW2 : Show Input TxPdo
SHOW3 : Show Output RxPdo
SHOW4 : Show Debug Information
SHOW5 : Show Debug Information Step by Step
ERASE : ERASE EEPROM
erase
*** ERASE EEPROM Start ***
*** ERASE EEPROM OK ***
>>> Please Power OFF & ON to Continue <<<
>>> Please Power OFF & ON to Continue <<<
>>> Please Power OFF & ON to Continue <<<
>>> Please Power OFF & ON to Continue <<<
>>> Please Power OFF & ON to Continue <<<
```

輸入 erase 再按 Enter 鍵來清除 EEPROM

圖 A4-21

7 斷電再上電來重啟 ECAT-2610(-DW)/2611 模組。

⑧ 同時按下 [Ctrl] + [F4] 鍵來將配置檔 (commands.txt) 載入到 ECAT-2610(-DW)/2611 模組中。

```

7188ECAT V1.45.1 [COM4:115200,N,8,1],FC=0,CTS=1, DIR=E:\7188ECAT
Line 9:0 0 0 0 0 0, 7 Reserved, format=Hex, default=0
Line 10:1, one commands<00-00>, max=300, format=Dec
Line 11:01 0F 00 00 00 08 01 00, 02, 00, 00, D/O=OutTxPdo[2], update
<00>
Line 12:STOP
Line 13: <error_0>Stop at line 13
      S_M=11, EEP_SIZE = 32
*** Wait 1 second ***
*** Write to EEPROM Start ***
CmdLen_10 : 01 0F 00 00 00 08 01 00 ,TxRx[2] Update<0> Eeprom<0026> CmdX<0> E
eprom<0027>
CRC16 = 5b 74
*** Write to EEPROM OK ***
>>> Please Power OFF & ON to Continue <<<
>>> Please Power OFF & ON to Continue <<<
>>> Please Power OFF & ON to Continue <<<
>>> Please Power OFF & ON to Continue <<<
>>> Please Power OFF & ON to Continue <<<
    
```


圖 A4-22

⑨ 斷電再上電來重啟 ECAT-2610(-DW)/2611 模組。

```

7188ECAT V1.45.1 [COM4:115200,N,8,1],FC=0,CTS=1, DIR=E:\7188ECAT
*** Check COM1 = INIT or Normal ***
COM1 TXD!=RXD --> Normal
*** Start Download from EEPROM ***
>> baudrate=10:115200, Parity=N, Stop=1, TimeOut=100, Delay=0, InMax=0, OutMax=0
, CmdNum=1
[000] : 010, 01 0F 00 00 00 08 01 00 ,<02>, TxRx[02], Update=00, CmdX=00
CRC16, [Compute = 5b 74], <EEP=5b 74> --> CRC16 OK
>> Load Configuration From EEPROM OK, 1 commands
EEPROM : InTxPdoMax++=1, OutRxPdoMax++=2
APP : InTxPdoMax=10, OutRxPdoMax=10
===== ECAT-2610, Ver. 2.25 =====
Ctrl+F4: Download
READ1 : Read EEPROM <sequential>
READ2 : Read EEPROM <Command>
SHOW0 : Show Version Number
SHOW1 : Show System Status
SHOW2 : Show Input TxPdo
SHOW3 : Show Output RxPdo
SHOW4 : Show Debug Information
SHOW5 : Show Debug Information Step by Step
ERASE : ERASE EEPOM
    
```

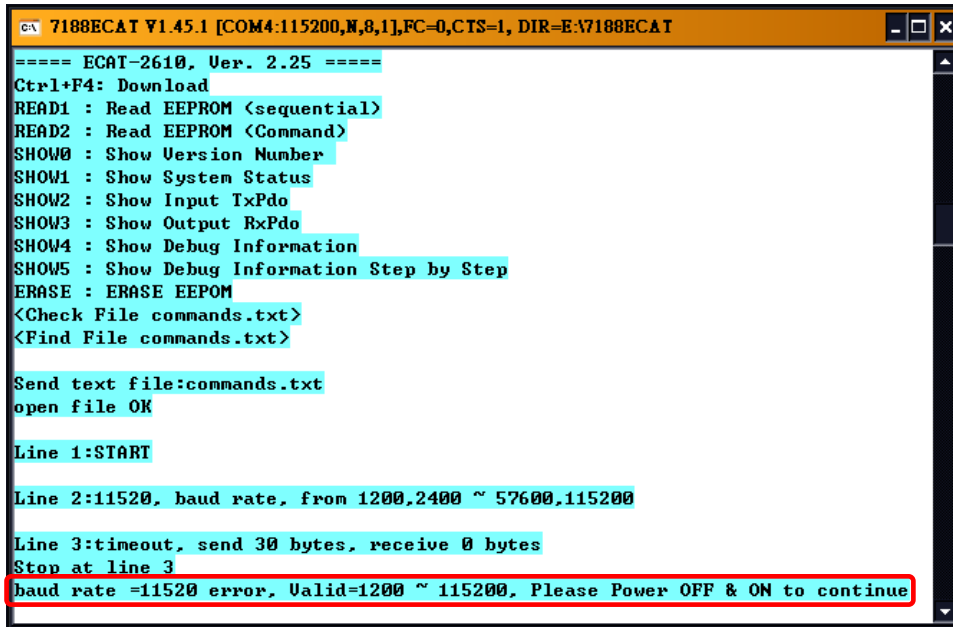
圖 A4-23

⑩ 載入完成後，按下視窗右上的  圖示來關閉視窗。



**注意**

1. 如果配置檔 (commands.txt) 中有任何設定錯誤，ECAT-2610(-DW)/2611 模組在載入過程中將會停止並顯示錯誤的地方，然後等待 ECAT-2610(-DW)/2611 模組被重新啟動後再才會繼續運作，如圖 A4-24 所示。



```
==== ECAT-2610, Ver. 2.25 ====
Ctrl+F4: Download
READ1 : Read EEPROM (sequential)
READ2 : Read EEPROM (Command)
SHOW0 : Show Version Number
SHOW1 : Show System Status
SHOW2 : Show Input TxPdo
SHOW3 : Show Output RxPdo
SHOW4 : Show Debug Information
SHOW5 : Show Debug Information Step by Step
ERASE : ERASE EEPROM
<Check File commands.txt>
<Find File commands.txt>

Send text file:commands.txt
open file OK

Line 1:START
Line 2:11520, baud rate, from 1200,2400 ~ 57600,115200
Line 3:timeout, send 30 bytes, receive 0 bytes
Stop at line 3
baud rate =11520 error, Valid=1200 ~ 115200, Please Power OFF & ON to continue
```

圖 A4-24

2. EEPROM 用於存儲不經常更改的數據，不適合頻繁的存取大量數據，且清除/寫入週期是有限制的，因此在測試使用時不應經常更改，否則容易造成模組損壞。

## A5. 支援泓格 Modbus RTU 從站產品

下表為 ECAT-2610 模組所支援泓格 Modbus RTU 從站設備:

型號	說明
 M-7000 系列	RS-485 遠端 I/O 模組 WebSite: <a href="http://www.icpdas.com/root/product/solutions/remote_io/rs-485/i-7000_m-7000/i-7000_m-7000_introduction.html">http://www.icpdas.com/root/product/solutions/remote_io/rs-485/i-7000_m-7000/i-7000_m-7000_introduction.html</a>
 M-2000 系列	RS-485 遠端 I/O 模組 WebSite: <a href="http://www.icpdas.com/root/product/solutions/remote_io/rs-485/m-2000/m-2000_selection.html">http://www.icpdas.com/root/product/solutions/remote_io/rs-485/m-2000/m-2000_selection.html</a>
 tM 系列	RS-485 遠端 I/O 模組 WebSite: <a href="http://www.icpdas.com/root/product/solutions/remote_io/rs-485/tm-series/tm-series_selection.html">http://www.icpdas.com/root/product/solutions/remote_io/rs-485/tm-series/tm-series_selection.html</a>
 LC 系列	燈光控制器 WebSite: <a href="http://www.icpdas.com/root/product/solutions/remote_io/rs-485/lighting_control/lighting_control_selection.html">http://www.icpdas.com/root/product/solutions/remote_io/rs-485/lighting_control/lighting_control_selection.html</a>
 SC 系列	燈光控制器 WebSite: <a href="http://www.icpdas.com/root/product/solutions/remote_io/rs-485/smart_control/smart_control_selection.html">http://www.icpdas.com/root/product/solutions/remote_io/rs-485/smart_control/smart_control_selection.html</a>
 DL 系列	溫濕度資料記錄器 WebSite: <a href="http://www.icpdas.com/root/product/solutions/remote_io/rs-485/dl_series/dl_selection.html">http://www.icpdas.com/root/product/solutions/remote_io/rs-485/dl_series/dl_selection.html</a>
 CL 系列	PM2.5/CO/CO2/溫度/濕度/Dew Point 記錄器 WebSite: <a href="http://www.icpdas.com/root/product/solutions/remote_io/rs-485/cl_series/cl_selection.html">http://www.icpdas.com/root/product/solutions/remote_io/rs-485/cl_series/cl_selection.html</a>
 ZT 系列	ZigBee I/O 模組 WebSite: <a href="http://www.icpdas.com/root/product/solutions/industrial_wireless_communication/wireless_solutions/wireless_selection.html#e">http://www.icpdas.com/root/product/solutions/industrial_wireless_communication/wireless_solutions/wireless_selection.html#e</a>

## A6. 手冊修訂記錄

本章提供此使用手冊的修訂記錄。

下表提供此文件每次修訂的日期與說明。

版本	發行日	說明
B1.0	2017 年 10 月	第一版: 8-bit 版本
B1.0	2018 年 3 月	第二版: 16-bit 版本
1.1	2018 年 5 月	首次發行
1.2	2018 年 8 月	<ol style="list-style-type: none"> <li>1. 修改 第 4.2 節 使用 ECAT-2610 Configuator.exe 設定配置檔並上傳到 ECAT-2610 模組。</li> <li>2. 新增 第 4.2.1 節 恢復至出廠預設值。</li> <li>3. 新增 16.Rs485_Cycle_Time、17.Ext_Sync 範例說明。</li> <li>4. 新增 A3.手動配置及上傳。</li> </ol>
1.3	2018 年 11 月	修正 第 3.3 節 配置運作模式中的 2610SYS1 檢查碼
1.4	2019 年 1 月	新增 ECAT-2611 產品資訊
1.4.1	2019 年 4 月	修改 ECAT-2611 支援功能碼表
1.5	2019 年 5 月	新增 ECAT-2610-DW 產品資訊