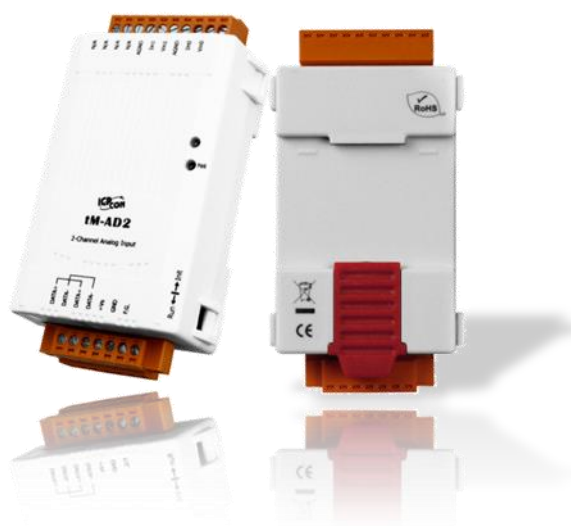


tM 系列使用手冊

V1.1, 2023/07



技術支援: service@icpdas.com

授權者: Wayne Chen, David Chen

編輯者: Janice Hong

保固說明

泓格科技股份有限公司，所生產製造的產品自交貨給原購買者起，均享有一年的保固期限。此保固僅限於產品材料與製造上的瑕疵。

免責聲明

泓格科技股份有限公司，對於因使用本系列產品所造成的任何損害並不負任何法律上的責任。本公司保留在任何時候修訂本書而不需通知的權利，並將確實地提供正確且可靠的資訊。然而，泓格科技股份有限公司無義務對任何因非法、不當使用，而導致的第三方侵權行為承擔任何法律責任。

版權宣告

版權所有 © 2018 泓格科技股份有限公司保留所有權利。

商標識別

本手冊中所提及之所有商標，均屬於其合法註冊公司所有。

技術服務

若您對產品有任何問題與建議，歡迎隨時與我們聯繫，我們將會盡快地回覆您。

Email: service@icpdas.com

改版記錄

此章節提供了本文件的歷史修改資訊。

下表顯示了歷史修改項目：

版本號	發行日期	說明
V1.1	2023/07	修改 6.4.1 節的下載連結
V1.1	2020/12	新增附錄 G – 儲存或載入 I/O 專案
V1.0	2020/07	第一版

目錄

改版記錄	3
目錄	4
第 1 章 介紹	8
1.1 產品特色.....	9
1.2 產品選型.....	10
1.3 產品規格.....	11
1.3.1 通用規格	11
1.3.2 規格 - AI.....	12
1.3.3 規格 - DI/DO	14
1.3.4 規格 - Multifunction I/O.....	17
1.4 軟體支援.....	19
1.4.1 DCON Utility Pro	19
1.4.2 EZ Data Logger	19
1.4.3 OPC Server	20
1.4.4 PACSDK	20
第 2 章 硬體設置	21
2.1 安裝.....	21
2.2 尺寸圖.....	21
2.3 接線圖.....	22
2.3.1 接線建議	22
2.3.2 接線 – tM-AD2.....	22
2.3.3 接線 – tM-AD5.....	23
2.3.4 接線 – tM-AD5C	23
2.3.5 接線 – tM-AD8.....	24
2.3.6 接線 – tM-AD8C	24
2.3.7 接線 – tM-TH8	25
2.3.8 接線 – tM-R5	25
2.3.9 接線 – tM-P3R3	26
2.3.10 接線 – tM-PD3R3.....	27
2.3.11 接線 – tM-P3POR3	28
2.3.12 接線 – tM-P4A4	29
2.3.13 接線 – tM-P4C4	30
2.3.14 接線 – tM-P8	31
2.3.15 接線 – tM-PDW8	32
2.3.16 接線 – tM-C8	33

2.3.17 接線 – tM-DA1P1R1	34
2.3.18 接線 – tM-AD2P2C2	35
2.4 硬體連接.....	36
2.5 操作模式開關 與 通訊參數.....	37
第 3 章 軟體工具 - DCON Utility Pro	38
3.1 搜尋 I/O 模組	38
3.2 設定頁面 – Configuration	40
3.2.1 設定通用項目 (適用模組: tM-P3R3, PD3R3, P3POR3, P4A4, P4C4, R5, C8, P8, PDW8) ..	41
3.2.2 設定類比項目 (適用模組: tM-AD2, AD4P2C2)	42
3.2.3 設定類比項目 (適用模組: tM-AD5, AD8, AD5C, AD8C).....	43
3.2.4 設定類比項目 (適用模組: tM-TH8, DA1P1R1)	44
3.3 設定頁面 – AI	45
3.3.1 設定頁面 01 - AI (適用模組: tM-AD2).....	45
3.3.2 設定頁面 02 - AI (適用模組: tM-AD5, AD8, AD5C, AD8C)	46
3.3.3 設定頁面 03 - AI (適用模組: tM-TH8)	47
3.4 設定其它 I/O 頁面	48
3.4.1 設定頁面 - User Defined Type.....	48
3.4.2 設定頁面 - AO	50
3.4.3 設定頁面 - AI/DO Alarm.....	52
3.4.4 設定頁面 - DO	54
3.4.5 設定頁面 - Host WDT.....	55
3.4.6 設定頁面 - DI.....	57
3.5 命令列工具 (Command Line Tool)	58
3.6 校正功能 (Calibration)	59
3.6.1 校正 AO 模組	59
第 4 章 軟體工具 – Modbus Master Tool	61
第 5 章 軟體工具 – EZ Data Logger / eLogger HMI	65
5.1 系統設定 (System Config)	67
5.2 群組設定 (Group Setup).....	68
5.2.1 安裝驅動程式	69
5.2.2 安裝裝置	70
5.2.3 修改通道設定 與 加入工作群組	71
5.2.4 設定工作群組屬性	74
5.3 開始執行 (Start) 或 進行模擬 (Simulation).....	75
5.3.1 量測計 (Gauge)	76

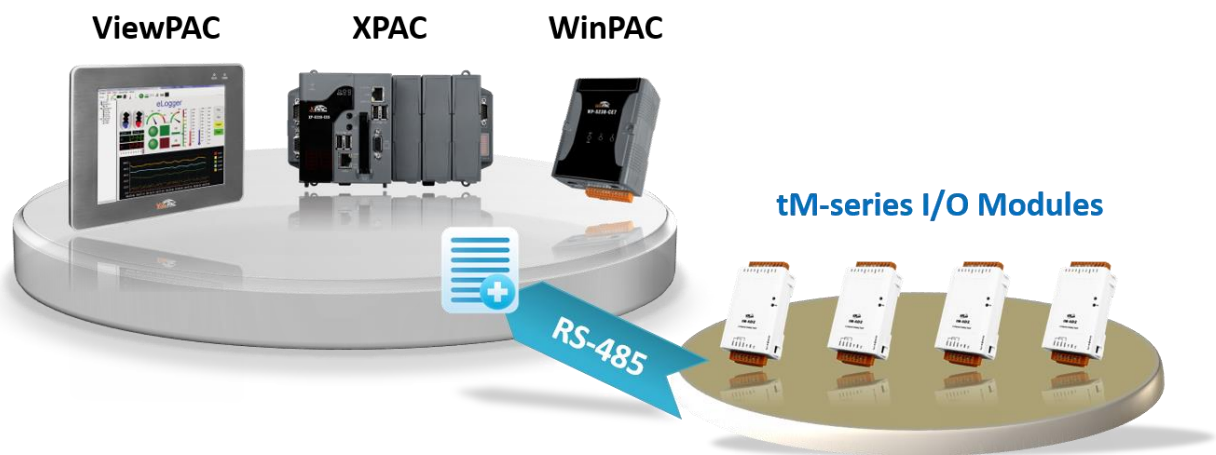
5.3.2 趨勢圖 (Trend)	77
5.3.3 平面配置圖 (Layout).....	78
5.4 開啟資料庫 (Open Database)	80
第 6 章 軟體開發工具	83
6.1 使用 DCON Utility Pro.....	83
6.2 使用 PACSDK.....	84
6.3 Modbus 開發工具 與 圖控軟體	89
6.3.1 軟體工具 – OPC DA Server 與 UA 系列產品.....	89
6.3.2 軟體工具 – nModbus.....	95
6.3.3 軟體工具 – LabView	98
6.3.4 圖控軟體 – Indusoft.....	101
6.4 Linux Modbus 開發工具.....	105
6.4.1 LinPAC 與 Modbus 應用	105
6.4.2 LinPAC 與 Python 應用	109
6.4.3 LinPAC 與 Perl 應用	111
6.5 使用 Win-GRAF 開發軟體.....	114
附錄 A 功能說明.....	118
A.1 INIT 模式.....	118
A.2 雙看門狗功能 (Dual Watchdog).....	119
A.2.1 開機值 (Power-on Value)	120
A.2.2 安全值 (Safe Value)	120
A.3 機殼接地 (Frame Ground)	121
A.4 轉換率 (Slew Rate) 控制.....	122
A.5 類比輸出回讀值 (AO Read-back)	122
A.6 進階數位輸入功能 (DI Latch、Counter).....	123
附錄 B 類比輸出/入範圍的 Type Code.....	124
B.1 tM-AD2 資料範圍	125
B.2 tM-AD5 資料範圍	126
B.3 tM-AD5C 資料範圍	126
B.4 tM-AD8 資料範圍	127
B.5 tM-AD8C 資料範圍	127
B.6 tM-AD4P2C2 資料範圍	128
B.7 tM-DA1P1R1 資料範圍	128
B.8 tM-TH8 資料範圍.....	129
B.8.1 Steinhart–Hart 係數 (User-defined Type).....	131
附錄 C Modbus 位址對應.....	132

附錄 D	Modbus RTU Protocol 指令	135
D.1	01 (0x01) Read Coil	137
D.2	02 (0x02) Read Discrete Inputs	138
D.3	03 (0x03) Read Multiple Registers	139
D.4	04 (0x04) Read Multiple Input Registers	140
D.5	05 (0x05) Force Single Coil.....	141
D.6	06 (0x06) Preset Single Register	142
D.7	15 (0x0F) Force Multiple Coils	143
D.8	16 (0x10) Set Multiple Register	144
附錄 E	DCON Protocol 指令	145
E.1	一般指令	147
E.2	I/O 指令	147
E.3	Host Watchdog、Power-on Value、Safe Value 指令	148
E.4	DI Latched、Counter 指令	149
E.5	Alarm 指令	149
E.6	模組校正指令	150
E.7	熱敏電阻 (User-defined Type) 指令	150
附錄 F	快速儲存與套用 I/O 設定檔	238
F.1	設定、寫入 與 儲存 I/O 組態	239
F.2	使用設定檔來配置模組	242
附錄 G	儲存或載入 I/O 專案.....	245
G.1	儲存專案 (Save Project).....	245
G.2	載入專案 (Load Project).....	248
附錄 H	其他資訊.....	252
H.1	如何搜尋產品網頁?	252
H.2	如何快速查詢章節?	253

第 1 章 介紹

tM 系列是一款具有數位 (Digital) 或類比 (Analog) I/O 功能的網路資料擷取與控制模組。藉由採用 DCON 或 Modbus RTU/ASCII 通訊協定，PAC 可經由 RS-485 串列匯流排來遠端控制 I/O 模組。RS-485 通訊埠的傳輸速度，最高可設為 115,200 bps。

Modbus 已經是工業界標準的通訊協定，也是當今最普遍用來連接工業電子設備的方式。同時，可讓 tM 系列模組，能更完善的與 HMI、SCADA、PLC 或 其他軟體系統進行整合。

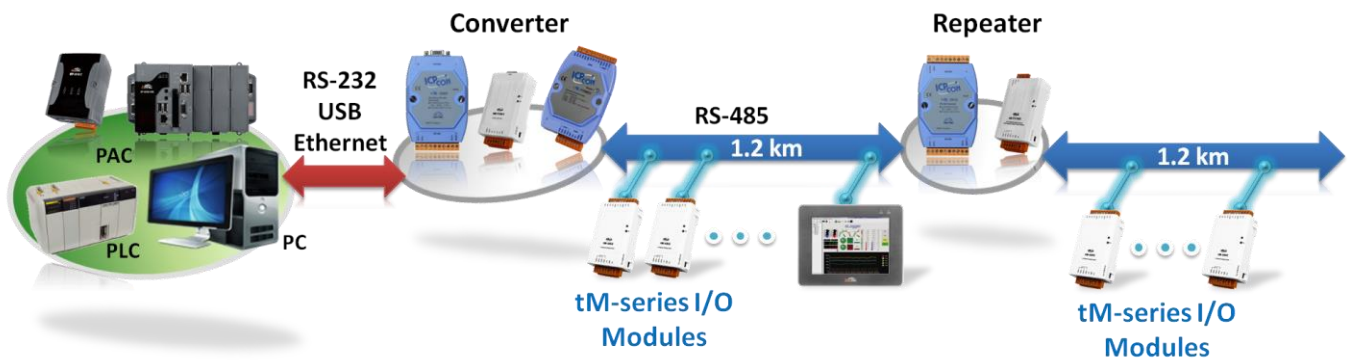


tM 系列 RS-485 小巧型 I/O 模組支援各種 I/O 類型，像是光隔離 (Photo-isolated) 數位輸入、功率繼電器 (Power Relay)、PhotoMOS 繼電器 (PhotoMOS Relay) 與 開集極 (Open-collector) 數位輸出 和 類比輸入 (電壓與電流)。相較於 M-7000 系列，tM 系列的低通道數設計更具經濟效益，並適用在分散式 I/O 應用。

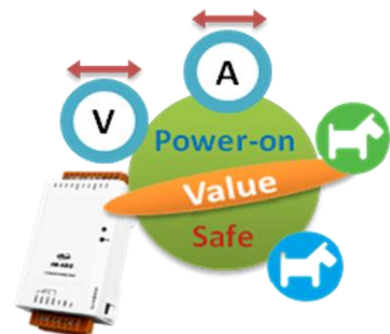
tM 系列提供雙看門狗功能，即模組看門狗 (Module Watchdog) 與 主機看門狗 (Host Watchdog)，可加強控制系統的可靠度與穩定性。此外，tM 系列具備了小巧的尺寸外型，可輕易的安裝在任何地方，甚至直接置入在機器內，可大幅地節省空間。同時，它配備有兩個可拆式的接線端子台，可方便配線。

1.1 產品特色

- 小巧外型與低功耗設計
- 通訊協定:
DCON, Modbus RTU/ASCII
- 支援 RS-485 工業多點網路 (Multi-Drop Network)
RS-485 匯流排，最高速率 (Baud Rate) 可達 115200 bps



- 可編程 I/O 類型 與 調整範圍
- 雙看門狗設計 (Dual-watchdog)
- 可編程開機值 (Power-on value) 與 安全值 (Safe value)
- 提供 DI Latch 功能
- 提供低速計數器
- 隔離型數位輸入與輸出
- 符合 RoHS 標準，不含鹵素
- 簡易的導軌式安裝



1.2 產品選型

tM 系列模組 (製表日期: 2019/06)				
匯流排	RS-485			
通訊協定	Modbus RTU, Modbus ASCII, DCON			
型號	AI	AO	DI	DO
tM-AD2	雙通道 (單端, 電壓/電流)	-	-	-
tM-AD5	5 通道 (差動, 電壓)			
tM-AD5C	5 通道 (差動, 電流)			
tM-AD8	8 通道 (單端, 電壓)			
tM-AD8C	8 通道 (單端, 電流)			
tM-TH8	8 通道 (熱敏電阻)			

型號	AI	AO	DI	DO
tM-DA1P1R1	-	1 通道		
		電壓/電流	Sink/Source	Form C 繼電器
tM-AD4P2C2	4 通道	-	2 通道	
	2 通道 (單端, 電壓)		Source	NPN, Sink
	2 通道 (單端, 電流)			
tM-P3R3	-		3 通道	
tM-PD3R3			Sink/Source	Form A 繼電器
tM-P3POR3			Source + Dry	
			Sink/Source	PhotoMos 繼電器
tM-P4A4	-		4 通道	
tM-P4C4			Sink	PNP, Source
			Source	NPN, Sink
tM-R5	-		-	5 通道
				Form A 繼電器
tM-P8	-		8 通道	-
			Sink/Source	
tM-PDW8			Sink/Source + Dry	
tM-C8			-	8 通道
			NPN, Sink	

1.3 產品規格

此章節將介紹各個 tM 模組的規格。

1.3.1 通用規格

Communication	
Interface	RS-485
Format	(N, 8, 1), (N, 8, 2), (O, 8, 1), (E, 8, 1)
Baud Rate	1200 ~ 115200 bps
Protocol	DCON, Modbus/RTU, Modbus/ASCII
Dual Watchdog	Yes, Module (2.3 seconds), Communication (Programmable)
LED Indicators	
Power	1 LED as Power Indicator
EMS Protection	
ESD (IEC 61000-4-2)	±4 kV Contact for Each Terminal ±8 kV Air for Random Point
	<u>tM-TH8:</u> ±2 kV Contact for Each Terminal ±3 kV Air for Random Point
EFT (IEC 61000-4-4)	±4 kV for Power
Isolation	
Intra-module Isolation, Field-to-Logic	Analog Input: 2500 VDC (Multi-function I/O: tM-AD4P2C2, tM-DA1P1R1) Digital Input: 3750 VDC PhotoMOS Relay Output: 2000 VDC
Mechanical	
Dimensions (W x L x H)	52 mm x 98 mm x 27 mm
Installation	DIN-Rail Mounting
Temperature Range	
Operating	-25 °C to +75 °C
Storage	-30 °C to +75 °C
Humidity	10 ~ 95% RH, Non-condensing

1.3.2 規格 - AI

Module Name		tM-AD2	tM-AD5	tM-AD5C	tM-AD8	tM-AD8C
Analog Input						
Input Channels		2 single-ended	5 differential	5 differential	8 single-ended	8 single-ended
Type		0 ~ 500 mV 0 ~ 1 V 0 ~ 2.5 V 0 ~ 5 V 0 ~ 10 V 0 ~ 20 mA 4 ~ 20 mA	+/- 1 V +/- 2.5 V +/- 5 V +/- 10 V	0 ~ 20 mA 4 ~ 20 mA +/- 20 mA	0 ~ 500 mV 0 ~ 1 V 0 ~ 2.5 V 0 ~ 5 V 0 ~ 10 V	0 ~ 20 mA 4 ~ 20 mA
Resolution	Normal Mode	14-bit				
	Fast Mode	12-bit				
Sampling Rate	Normal Mode	10 Hz total				
	Fast Mode	200 Hz total				
Accuracy	Normal Mode	± 0.1%				
	Fast Mode	± 0.5%				
Zero Drift		± 20 μV/°C				
Span Drift		± 25 ppm/°C				
Input Impedance	Voltage	10 MΩ	20 MΩ	-	10 MΩ	-
	Current	125 Ω	-	125 Ω	-	125 Ω
Over Protection	Voltage	120 VDC		-	120 VDC	-
	Current	Yes, 50 mA at 110 VDC	-			
Open Wire Detection for 4 ~ 20 mA		Yes	-			
Power Requirements						
Reverse Polarity Protection		Yes				
Powered from Terminal Block		Yes, 10 ~ 30 VDC				
Consumption		0.7 W Max.	1 W Max.		0.7 W Max.	

(請見 1.3.1 節 - 通用規格)

Module Name	tM-TH8
Analog Input	
Input Channels	8
Type	Thermistor
Thermistor Type	Precon ST-A3, Fenwell Type U, YSI L 100, YSI L 300, YSI L 1000, YSI B 2252, YSI B 3000, YSI B 5000, YSI B 6000, YSI B 10000, YSI H 10000, YSI H 30000, User-defined
Resolution	16-bit
Sampling Rate	8 Hz total
Accuracy	± 1%
Over-voltage Protection	120 VDC
Open Wire Detection	Yes
Individual Channel Configurable	Yes
Power Requirements	
Reverse Polarity Protection	Yes
Powered from Terminal Block	Yes, 10 ~ 30 VDC
Consumption	1 W Max.

(請見 1.3.1 節 - 通用規格)

1.3.3 規格 - DI/DO

Module Name		tM-P8	tM-PDW8
Digital Input/Counter			
Input Channels		8	
Type		Wet Contact (Sink, Source)	Wet Contact (Sink, Source) Dry Contact (Source)
Wet Contact	ON Voltage Level	+3.5 VDC ~ 50 VDC	
	OFF Voltage Level	+1 VDC Max.	
Dry Contact	ON Voltage Level	-	Close to GND
	OFF Voltage Level	-	Open
Counters	Max. Count	65535 (16-bit)	
	Max. Input Frequency	100 Hz	
	Min. Pulse Width	5 ms	
Input Impedance		10 KΩ, 0.5 W	
Over-voltage Protection		±70 VDC	
Power Requirements			
Reverse Polarity Protection		Yes	
Powered from Terminal Block		Yes, 10 ~ 30 VDC	
Consumption		0.2 W Max.	0.43 W Max.

(請見 1.3.1 節 - 通用規格)

Module Name		tM-P3POR3	tM-P3R3	tM-PD3R3
Digital Input/Counter				
Input Channels		3		
Type		Wet Contact (Sink, Source)		Dry Contact (Source)
Wet Contact	ON Voltage Level	+3.5 VDC ~ +50 VDC		-
	OFF Voltage Level	+1 VDC Max.		-
Dry Contact	ON Voltage Level	-		Close to GND
	OFF Voltage Level	-		Open
Counters	Max. Count	65535 (16-bit)		
	Max. Input Frequency	100 Hz		
	Min. Pulse Width	5 ms		
Input Impedance		10 KΩ, 0.5W		-
Over-voltage Protection		70 VDC		-

Module Name		tM-P3POR3	tM-P3R3	tM-PD3R3	tM-R5
Relay Output					
Output Channels		3			5
Relay Type		PhotoMOS Relay	Power Relay, Form A (SPST N.O.)		
Operating Load Voltage Range		80 V (AC peak or DC)	250 VAC or 30 VDC		
Load Current		1 A Max.	5 A Max.		
		3 A, (Peak, 1 ms, 1 shot)			
Output Off State Leakage Current		1 uA	-		
Operating Time		5 ms (Max.)	6 ms		
Release Time		0.5 ms (Max.)	3 ms		
Electrical Life (Resistive load)	VDE	-	5 A @250 VAC 30,000 ops (10 ops/minute) at 75°C		
			5 A @30 VDC 70,000 ops (10 ops/minute) at 75°C		
	UL		5 A @250 VAC/30 VDC 6,000 ops		
			3 A @250 VAC/30 VDC 100,000 ops		
Mechanical Life		-	20,000,000 ops at no load (300 ops/minute)		
Electrical Endurance		No Arcing, No Bounce, and No Switching Noise	-		
Power-on Value		Yes, Programmable			
Safe Value		Yes, Programmable			
Power Requirements					
Reverse Polarity Protection		Yes			
Powered from Terminal Block		Yes, 10 ~ 30 VDC			
Consumption		0.4 W Max.	0.8 W Max.	0.9 W Max.	1 W Max.

(請見 1.3.1 節 - 通用規格)

Module Name		tM-P4A4		tM-P4C4		tM-C8	
Digital Input/Counter							
Input Channels		4				No Digital Input	
Type		Wet Contact (Sink)		Wet Contact (Source)			
On Voltage Level		+4.0 VDC ~ +50 VDC		+3.5 VDC ~ +50 VDC			
Off Voltage Level		+1 VDC Max.					
Input Impedance		10 KΩ, 0.66 W					
Counters	Channels	4					
	Max. Count	65535 (16-bit)					
	Max. Frequency	100 Hz					
	Min. Pulse Width	5 ms					
Over-voltage Protection		70 VDC					
Digital Output							
Output Channels		4				8	
Type		Isolated Open Emitter (Source)		Isolated Open Collector (Sink)			
Max. Load Current		650 mA/channel		700 mA/channel			
Load Voltage		+10 VDC ~ +40 VDC		+3.5 ~ +50 VDC			
Over-voltage Protection		47 VDC		60 VDC			
Overload Protection		Yes				1.4 A	
Short Circuit Protection		Yes					
Power-on Value		Yes, Programmable					
Safe Value		Yes, Programmable					
Isolation							
Intra-module Isolation, Field-to-Logic		3750 VDC					
Power Requirements							
Reverse Polarity Protection		Yes					
Powered from Terminal Block		Yes, 10 ~ 30 VDC					
Consumption		0.5 W Max.				0.8 W Max.	

(見 1.3.1 節 - 通用規格)

1.3.4 規格 - Multifunction I/O

Module Name	AI	AO	DI	DO
tM-DA1P1R1	-	1	1	Relay: 1
tM-AD4P2C2	Voltage: 2 ; Current: 2	-	2	2

(請見 1.3.1 節 - 通用規格)

Module Name		tM-DA1P1R1	tM-AD4P2C2
Analog Input			
Input Channel	Voltage	No Analog Input	2 single-ended
	Current		
Type	Voltage		±1 V, ±2.5 V, ±5 V, ±10 V
	Current		±20 mA, 0 ~ 20 mA, 4 ~ 20 mA
Resolution	Normal Mode		14-bit
	Fast Mode		12-bit
Sampling Rate	Normal Mode		10 Hz total
	Fast Mode		200 Hz total
Accuracy	Normal Mode		± 0.1%
	Fast Mode		± 0.5%
Zero Drift			±20 μV/°C
Span Drift			±25 ppm/°C
Input Impedance	Voltage		10 MΩ
	Current		136 Ω
Over Protection	Voltage		120 VDC
	Current		Yes, 50 mA at 110 VDC
Analog Output			
Output Channels		1	No Analog Output
Type		0 ~ 10 V, 0 ~ 20 mA, 4 ~ 20 mA	
Resolution		12-bit	
Accuracy		+/-0.1% of FSR	
DA Output Response Time		10 ms	
Voltage Output Capability		20 mA	
Current Load Resistance		500 Ω	

Module Name		tM-DA1P1R1	tM-AD4P2C2
Digital Input/Counter			
Input Channel		1	2
Type	Dry Contact	Source	-
	Wet Contact	Sink/Source	Source
Wet Contact	ON Voltage Level	+3.5 VDC ~ +50 VDC	
	OFF Voltage Level	+1 VDC Max.	
Dry Contact	ON Voltage Level	Close to GND	-
	OFF Voltage Level	Open	
Input Impedance		10 KΩ, 0.5 W	
Counters	Channels	1	2
	Max. Count	65535 (16-bit)	
	Max. Input Frequency	100 Hz	50 Hz
	Min. Pulse Width	5 ms	10 ms
Over-voltage Protection		70 VDC	
Digital/Relay Output			
Channels		1	2
Type		Power Relay, Form C	Isolated Open Collector (Sink)
Max. Load Current		NO : 10 A @ 250 VAC NC : 6 A @ 250 VAC	700 mA/Channel
Load Voltage		-	+3.5 VDC ~ +50 VDC
Over Protection	voltage	-	60 VDC
	Current		1.4 A (with short circuit protection)
Short Circuit Protection			
Operating Load Voltage Range		250 VAC or 30 VDC	-
Operating Time		15 ms Max.	
Release Time		5 ms Max.	
Mechanical Endurance		1 X 10 ⁷ OPS	
Electrical Endurance		5 X 10 ⁴ OPS	
Power On Value		Yes, Programmable	
Safe Value		Yes, Programmable	
Power Requirements			
Reverse Polarity Protection		Yes	
Powered from Terminal Block		Yes, 10 ~ 30 VDC	
Consumption		1.8 W Max.	1 W Max.

1.4 軟體支援

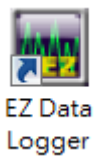
ICP DAS 提供了下列免費的軟體工具與開發套件，包含：

1.4.1 DCON Utility Pro



DCON Utility Pro 可用來透過序列埠 (RS-232/485) 搜尋、設定並測試 I/O 模組。它支援 DCON 與 Modbus 通訊協定，並可在 Window PC 或 PAC 上運行。(詳見 [第 3 章](#))

1.4.2 EZ Data Logger



EZ Data Logger 是泓格科技提供給使用者，用來在 Windows 2000/XP/Vista 上建立小型 SCADA 系統的軟體。EZ Data Logger 有簡易版與專業版兩種版本，簡易版不僅具備了完整的功能，對 ICP DAS 產品用戶是完全免費的。(詳見 [第 4 章](#))

EZ Data Logger 是一套簡易型的數據採集軟體，可應用在小型的遠程 I/O 系統。在簡單易用的介面下，使用者可簡易、快速地建立一個數據採集應用軟體，而不需要任何編程工具。

主要功能：

- 支援 DCON、Modbus RTU、Modbus ASCII、Modbus TCP 等通訊協定
- 支援多個 COM Port 與 TCP/IP 連線
- 支援虛擬通道的定義
- 支援邏輯控制 (VB Script)
- 支援警報通知 (使用手機簡訊 或 電子郵件)
- 靈活的群組設定
- 即時趨勢圖 (可放大縮小)
- Layout 嵌入背景圖
- 網路攝影機檢視
- 使用 Access 資料庫格式 (可匯出 Excel 或 CVS 檔)
- 報表列印
- 警示燈與警示聲
- 可以直接搜尋 DCON 與 Modbus 模組
- 數值轉換
- 支援權限管理
- 無需編寫程式

1.4.3 OPC Server







NAPOPC_ST DA Server 是一套免費且支援 ICP DAS 產品的 OPC DA Server ("OPC" 代表 "OLE for Process Control" 而 "DA" 代表 "Data Access")。

基於微軟公司的 OLE COM (元件物件模型) 與 DCOM (分散式元件物件模型) 技術，NAPOPC_ST DA Server 定義了一套標準的物件、介面與方法，可用在程序控制 與 製造自動化應用，以促使設備資料間的互通性。

藉由 SCADA/HMI/資料庫軟體的使用，系統能夠經由對 NAPOPC_ST DA Server 發出請求，透過 NAPOPC_ST DA Server 蒐集不同電腦間 ICP DAS 設備以及第三方設備之資料，再將結果回應給 SCADA/HMI/資料庫，以達到系統整合的目的。

ICP DAS 針對不同 PAC 的作業系統 (OS)，提供幾個專有的 DA Servers 版本：

版本	 NAPOPC_ST	 NAPOPC_XPE	 NAPOPC_CE5	 NAPOPC_CE6
平台	Desktop Windows	Windows XP Embedded	Windows CE5	Windows CE6
費用	免費/\$	免費	免費	免費

請參訪 <https://opc.icpdas.com> 取得更多資訊。

1.4.4 PACSDK

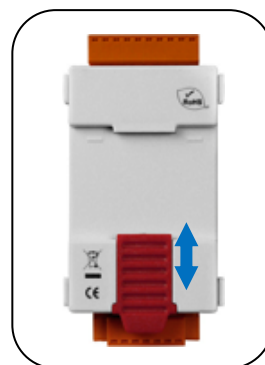


PACSDK 是一種軟體開發工具，包含了開發 XPAC、WinPAC 與 ViewPAC 應用程式所需的 Header 檔、Library 檔、文件 與 工具。(詳見 [6.2 節](#))

泓格科技發行了新的 SDK (稱為 PACSDK)，它整合並取代了原先的 XPACSDK 與 WinPACSDK，讓 User 可輕易地將開發的應用程式用在不同的 PAC 平台上。

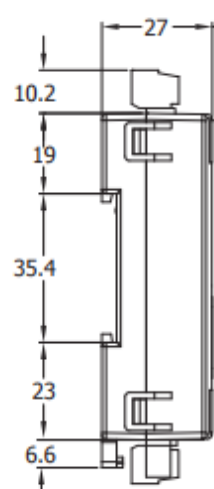
第 2 章 硬體設置

2.1 安裝

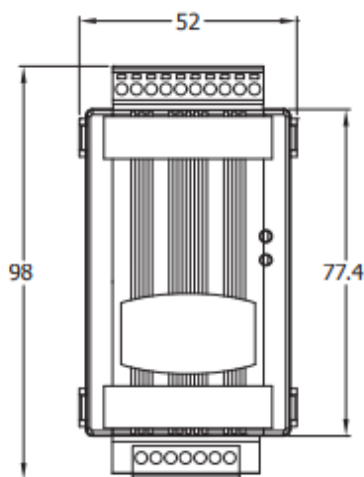


2.2 尺寸圖

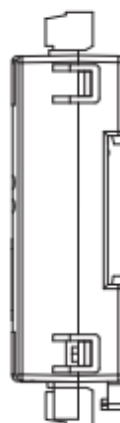
下方提供了 tM 系列模組的尺寸圖，可供使用者在自訂機櫃時參考用，尺寸單位是公厘 (mm)。



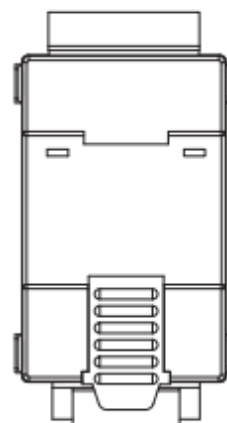
Left Side View



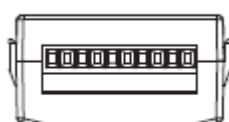
Front View



Right Side View



Rear View



Top View



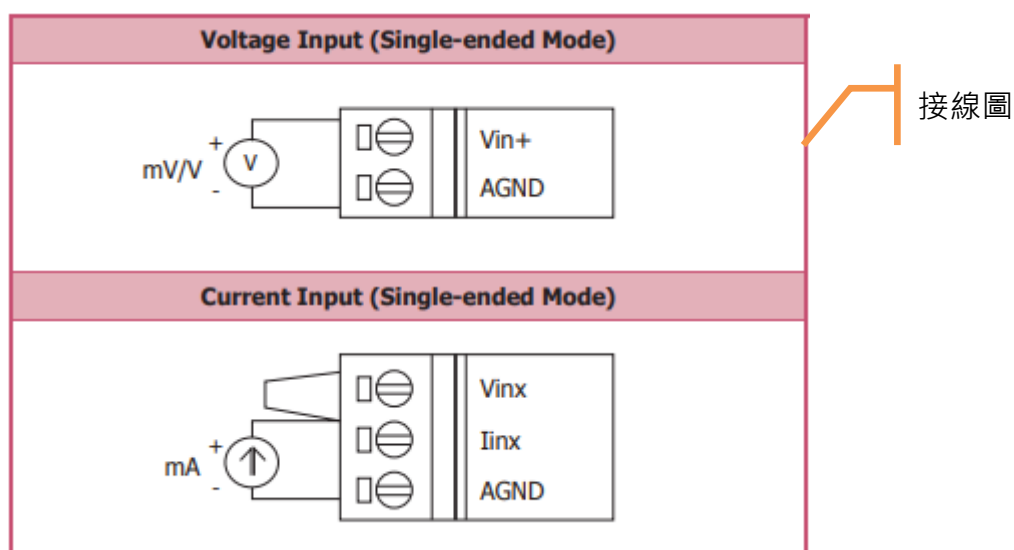
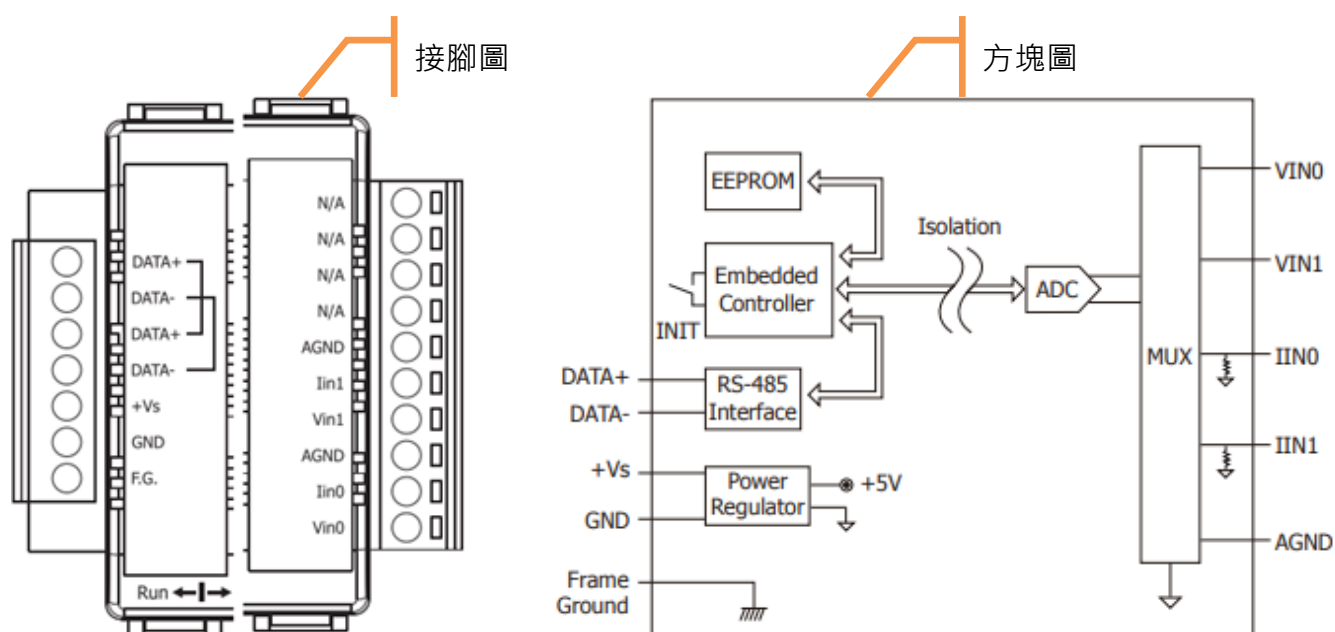
Bottom View

2.3 接線圖

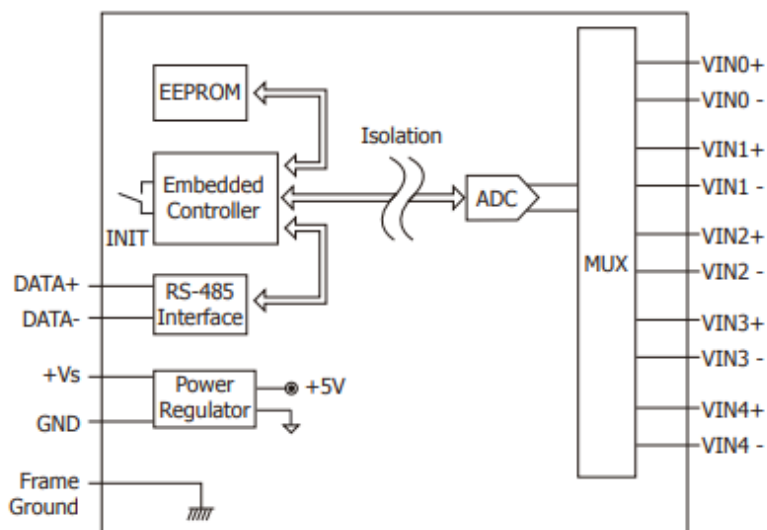
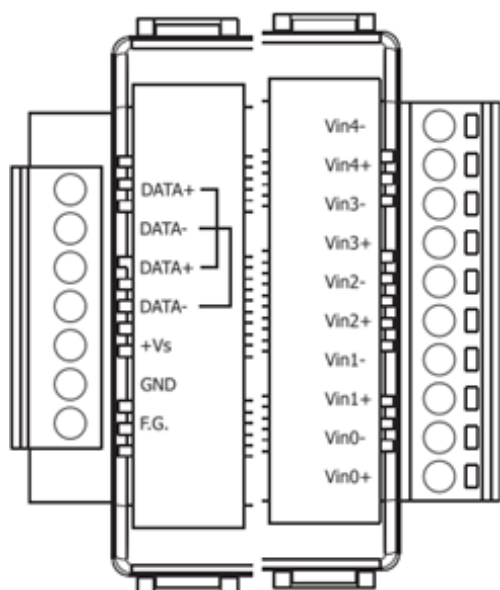
2.3.1 接線建議

- 使用 26-12 AWG 作為信號連接線
- 建議剝線長度為 $7 \pm 0.5\text{mm}$
- 使用壓接式端子台 (Crimp Terminal) 接線
- 避免和高壓電纜 或 電力設備連接
- 使用 24 AWG 絕緣雙絞線作為 RS-485 訊號傳輸線 (例如: Belden 9841 傳輸線)

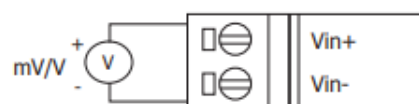
2.3.2 接線 – tM-AD2



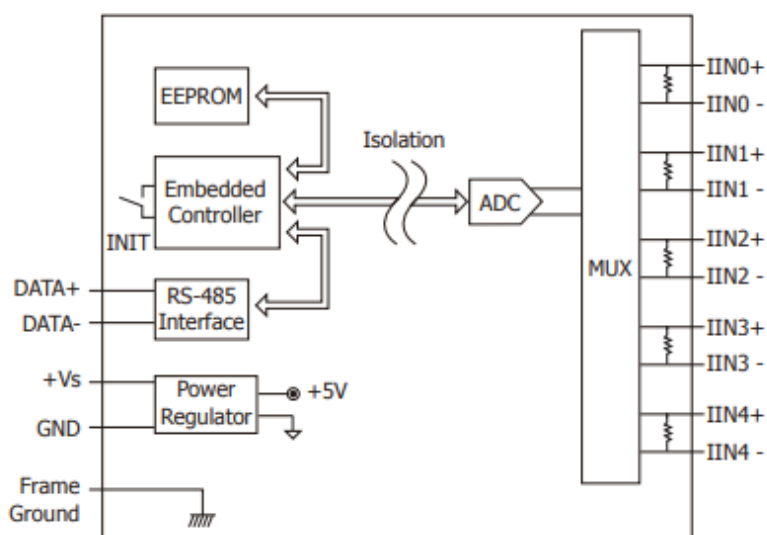
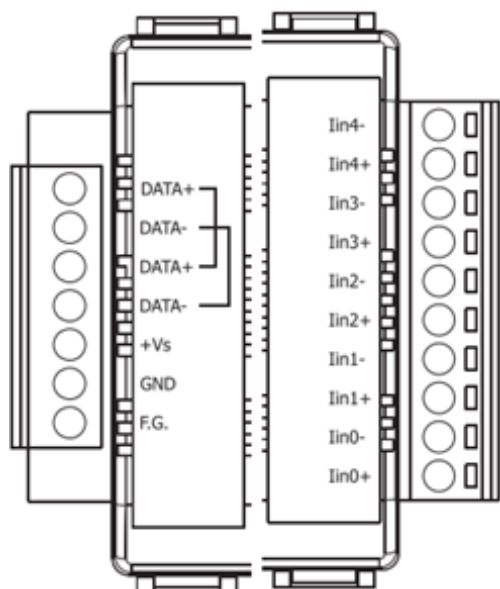
2.3.3 接線 – tM-AD5



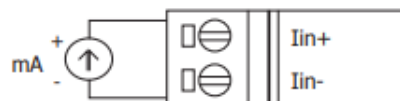
Voltage Input (Differential Mode)



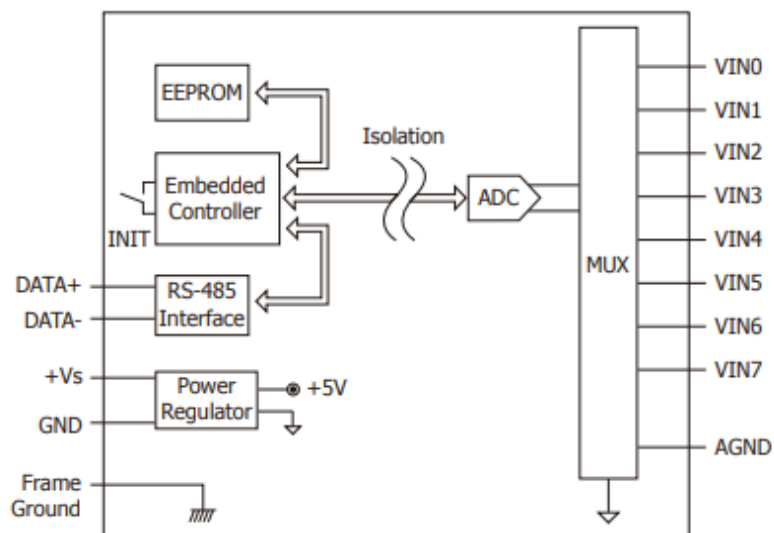
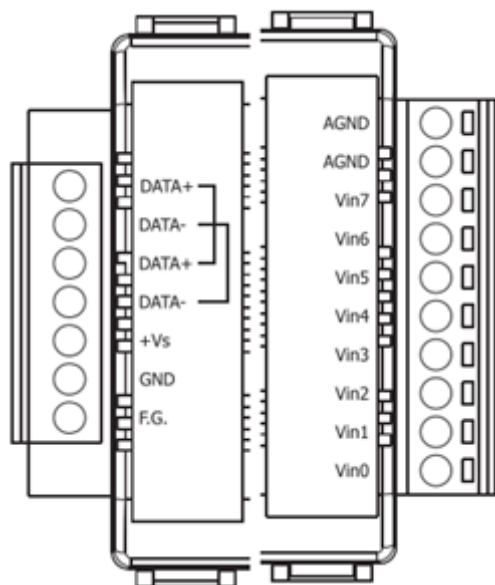
2.3.4 接線 – tM-AD5C



Current Input



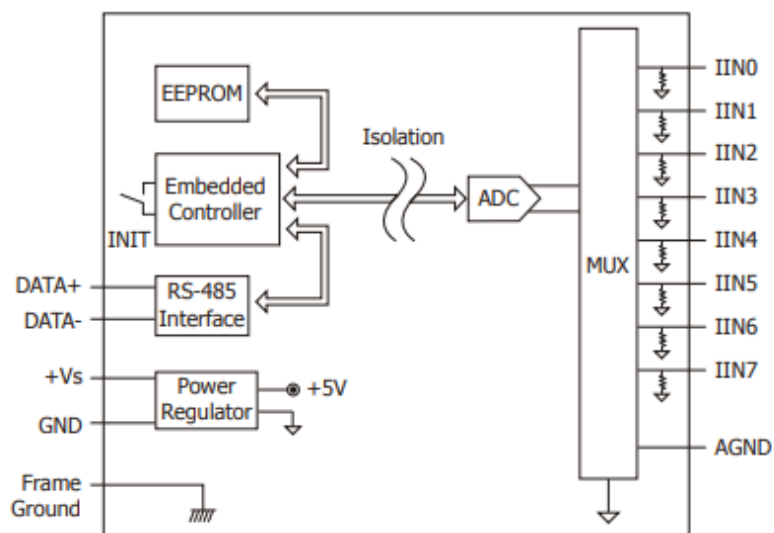
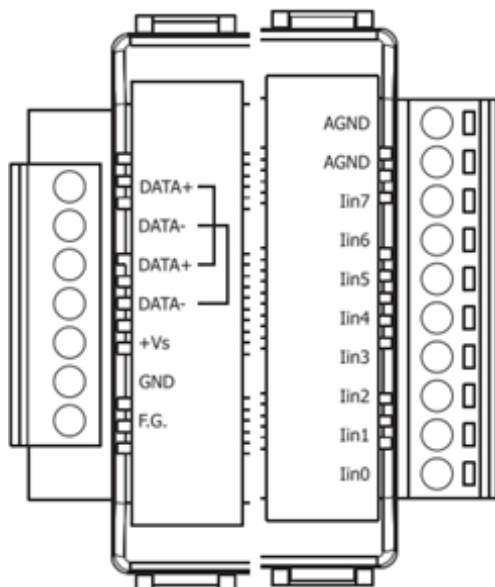
2.3.5 接線 – tM-AD8



Voltage Input (Single-ended Mode)



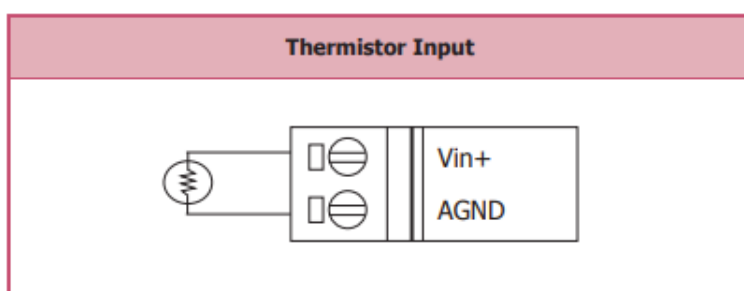
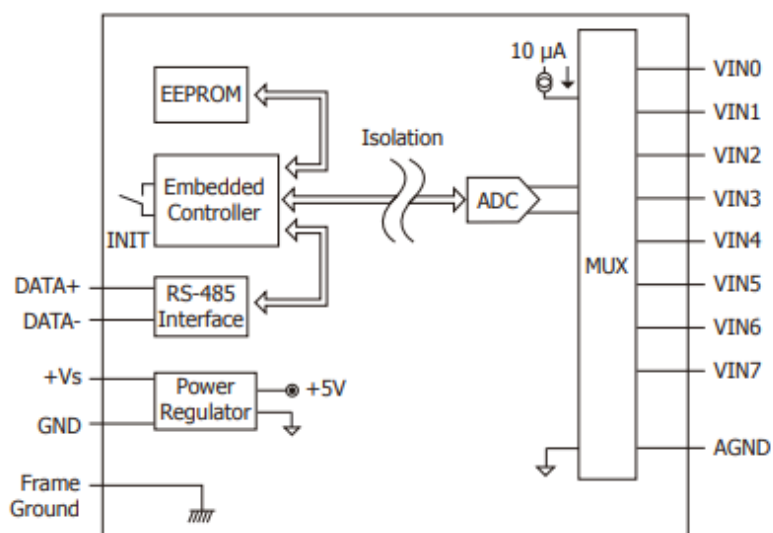
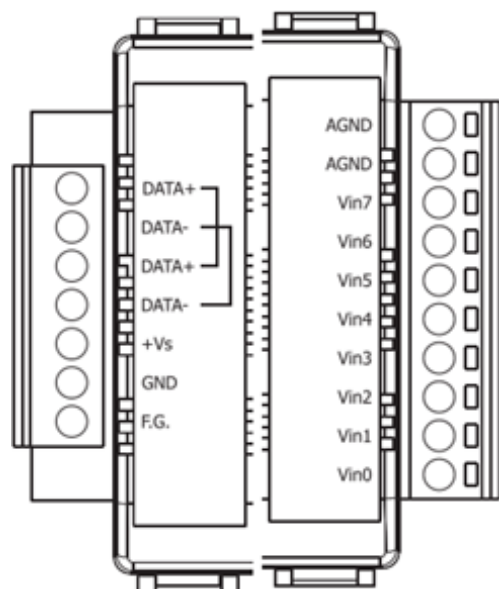
2.3.6 接線 – tM-AD8C



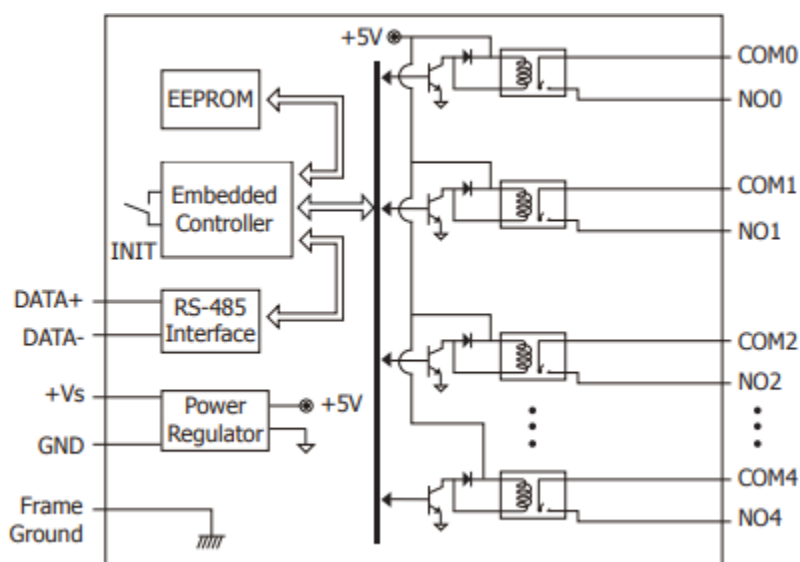
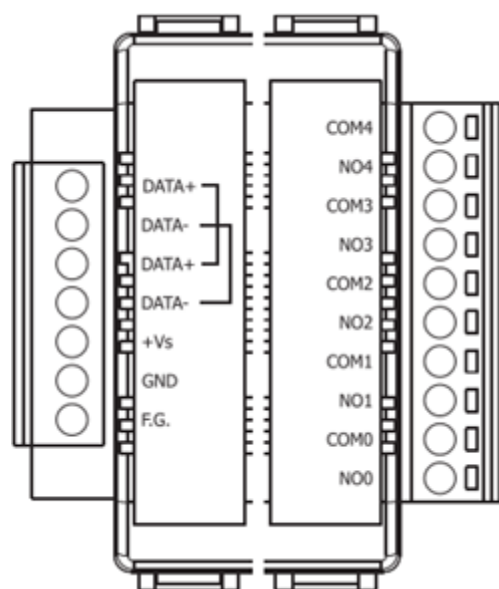
Current Input (Single-ended Mode)



2.3.7 接線 - tM-TH8

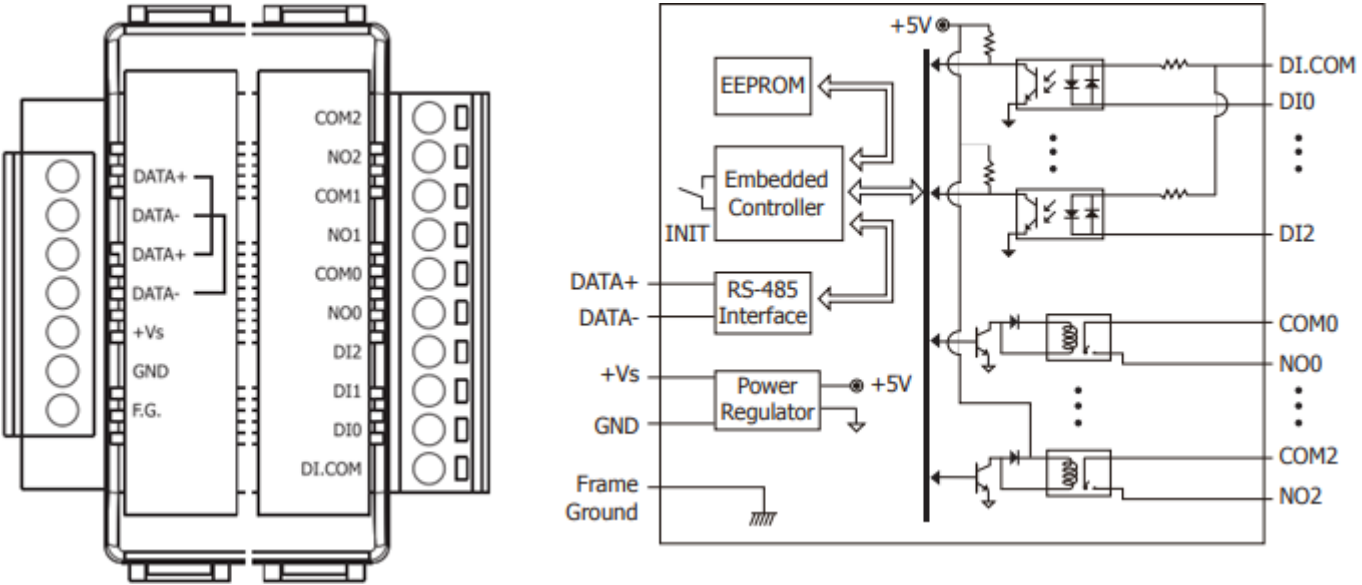


2.3.8 接線 - tM-R5



Power Relay	ON State Readback as 1	OFF State Readback as 0
Relay Output		

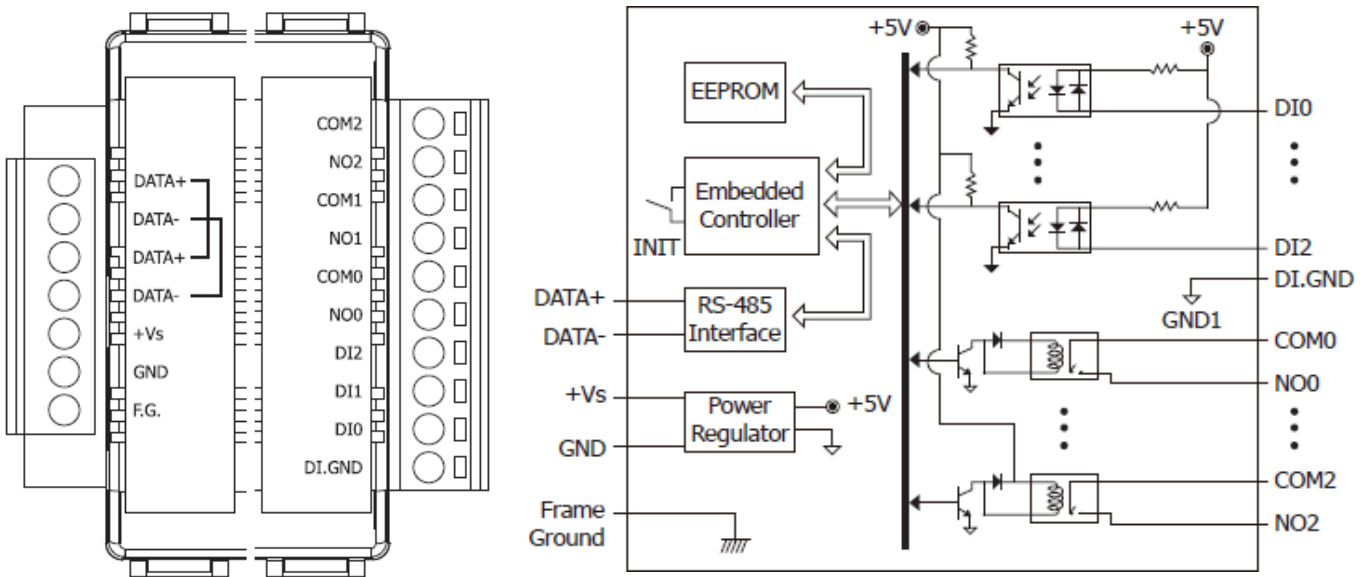
2.3.9 接線 – tM-P3R3



Input Type	ON State Readback as 1	OFF State Readback as 0
Wet Contact (Sink)	+3.5 ~ +50 Vpc	OPEN or <1 Vpc
Wet Contact (Source)	+3.5 ~ +50 VDC	OPEN or <1 VDC

Power Relay	ON State Readback as 1	OFF State Readback as 0
Relay Output	COMx	COMx

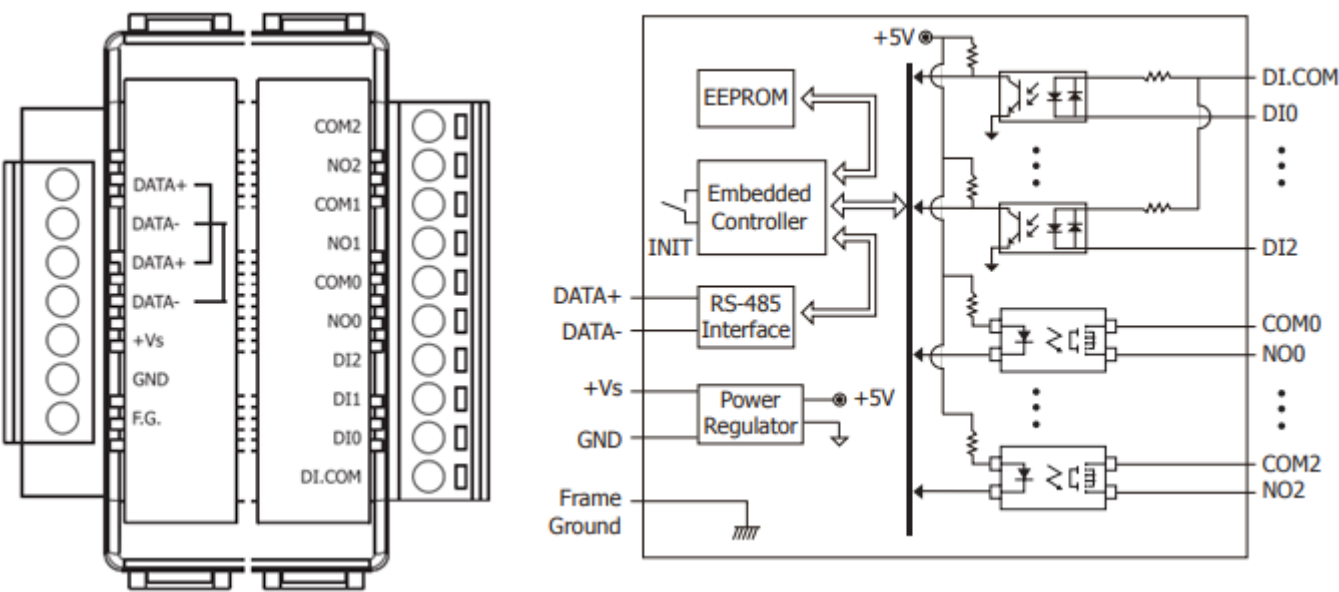
2.3.10 接線 – tM-PD3R3



Input Type	ON State Readback as 1	OFF State Readback as 0
Dry Contact	Close to GND	Open

Power Relay	ON State Readback as 1	OFF State Readback as 0
Relay Output		

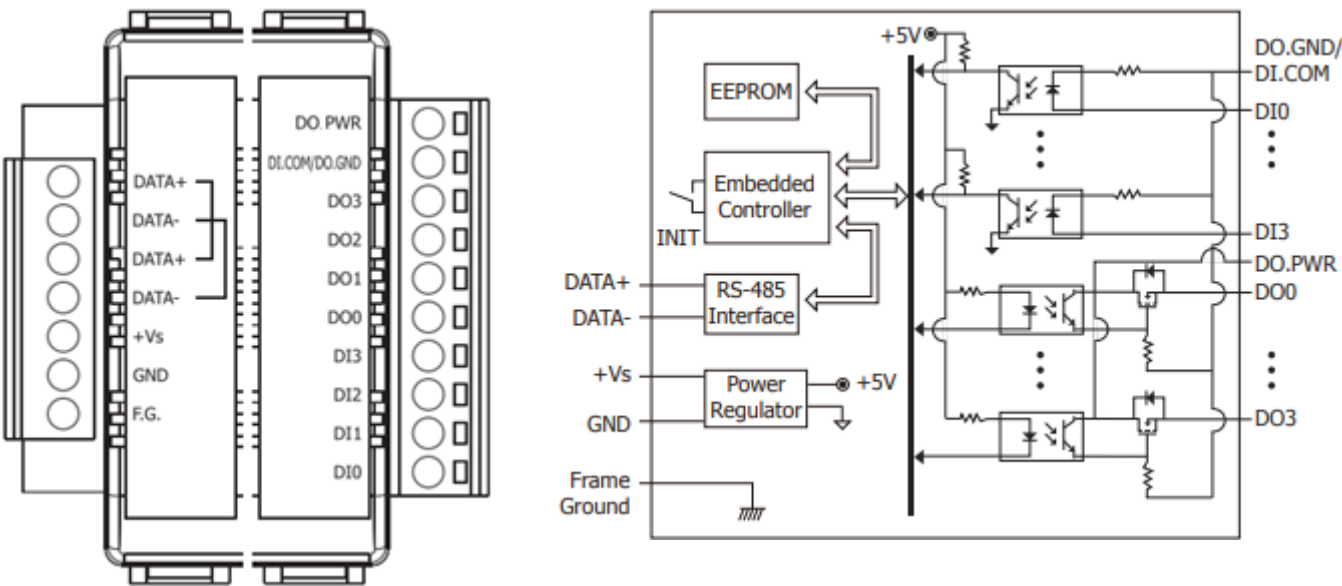
2.3.11 接線 – tM-P3POR3



Digital Input/ Counter	ON State Readback as 1	OFF State Readback as 0
Sink	+3.5 ~ +50 V _{DC}	OPEN or <1.0 V _{DC}
Source	+3.5 ~ +50 V _{DC}	OPEN or <1.0 V _{DC}

PhotoMOS Relay	ON State Readback as 1	OFF State Readback as 0
Relay Output		

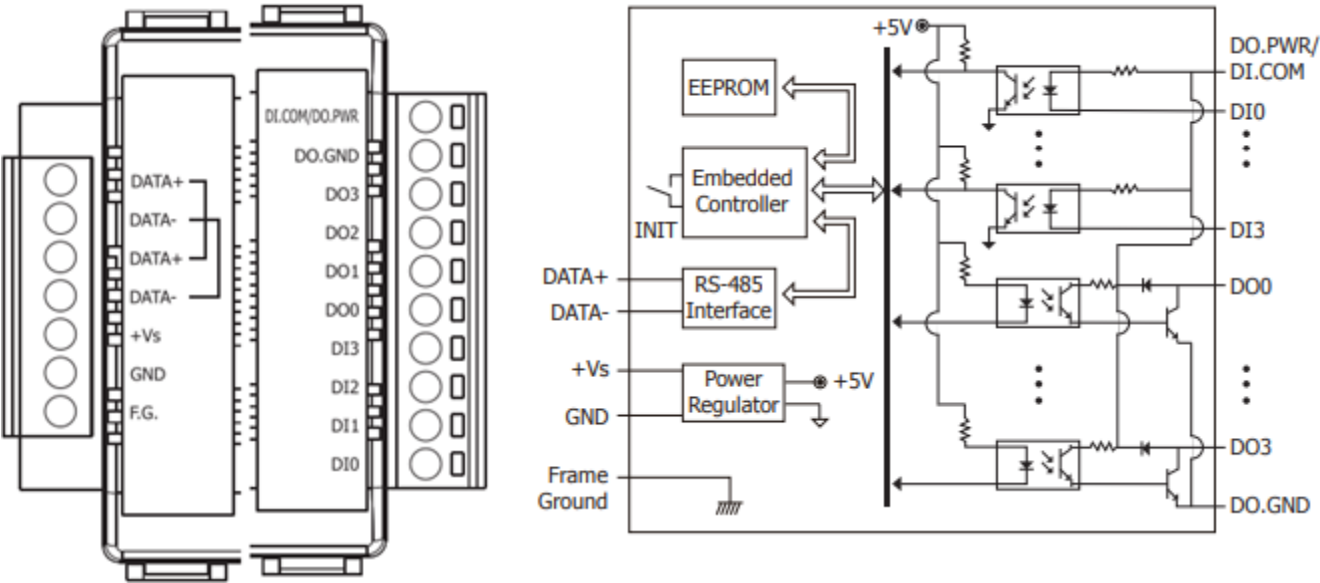
2.3.12 接線 – tM-P4A4



Digital Input/ Counter	ON State Readback as 1	OFF State Readback as 0
	+3.5 ~ +50 Vdc	OPEN or <1 Vdc
Sink		

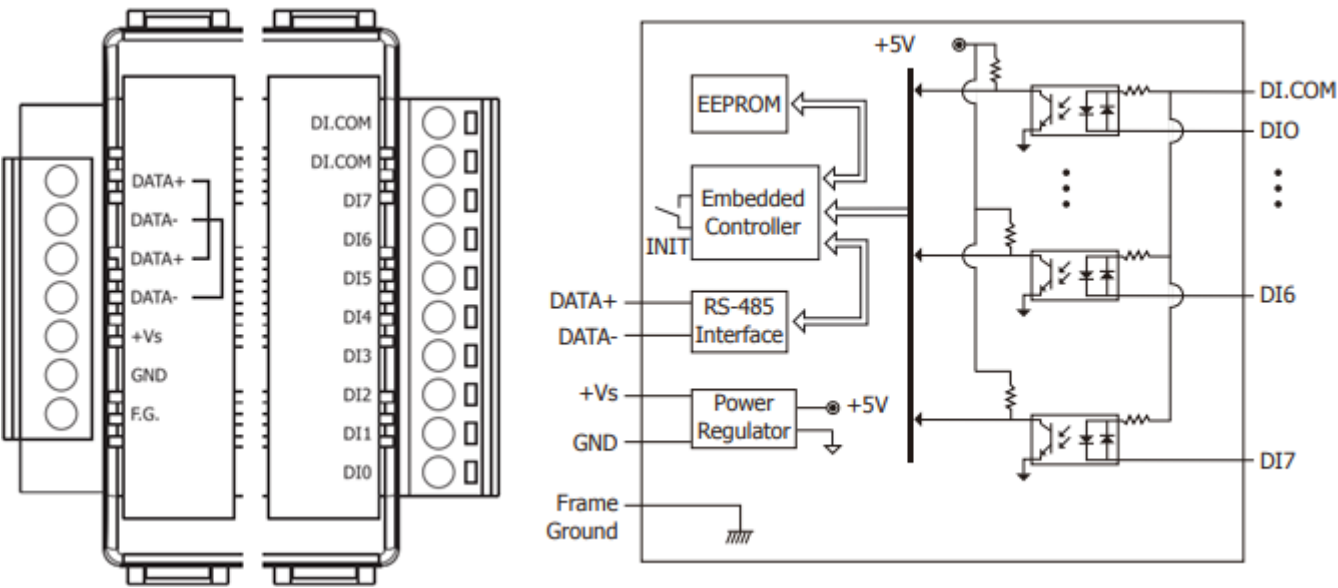
Output Type	ON State Readback as 1	OFF State Readback as 0
Drive Relay		
Resistance Load		

2.3.13 接線 – tM-P4C4



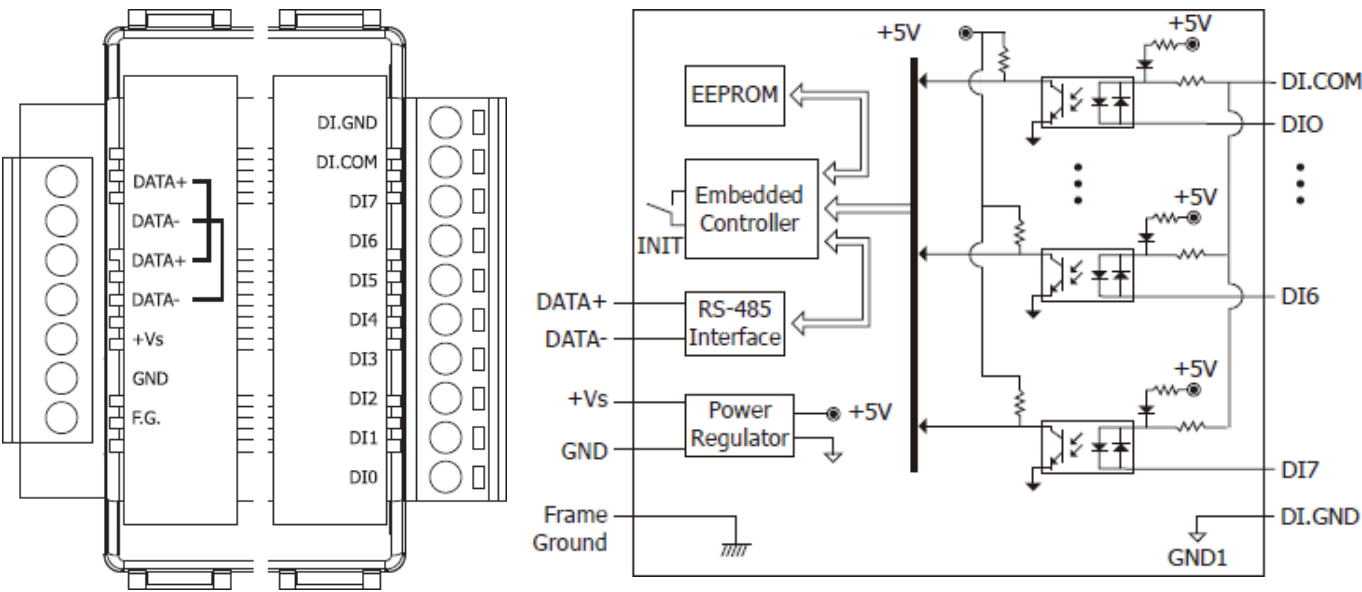
Digital Input/ Counter	ON State Readback as 1	OFF State Readback as 0
	+3.5 ~ +50 V _{DC}	OPEN or <1 V _{DC}
Source		
Output Type	ON State Readback as 1	OFF State Readback as 0
Drive Relay		
Resistance Load		

2.3.14 接線 – tM-P8



Input Type	ON State Readback as 1	OFF State Readback as 0
	+3.5 ~ +50 V _{DC}	OPEN or <1 V _{DC}
Wet Contact (Sink)		
Wet Contact (Source)		

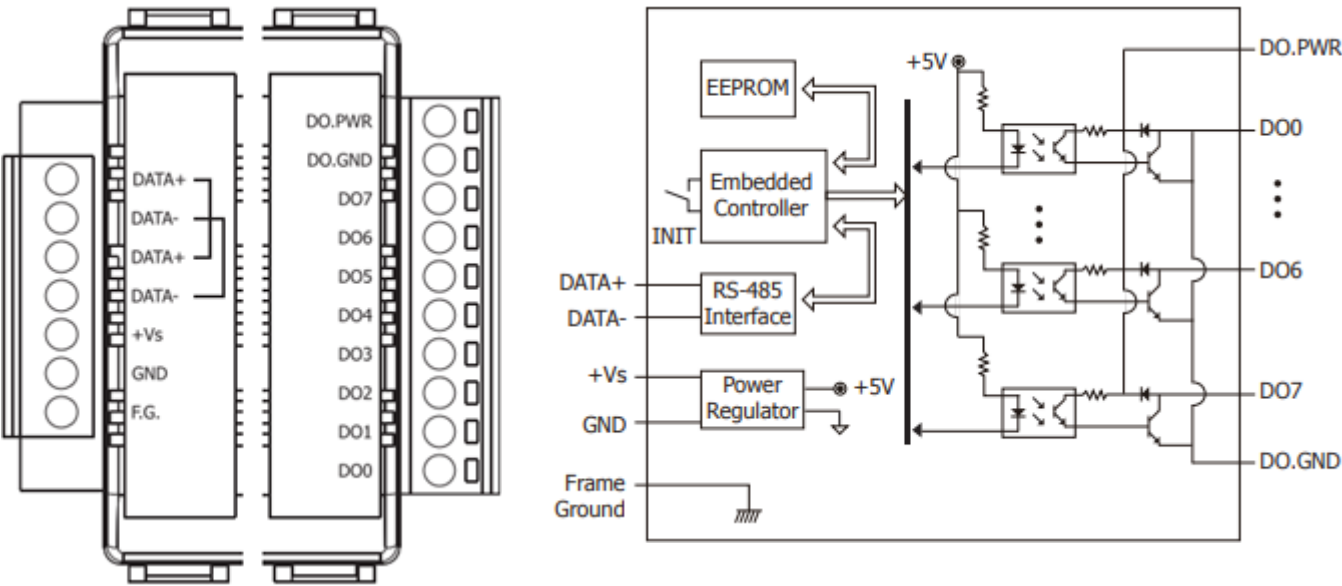
2.3.15 接線 – tM-PDW8



Input Type	ON State Readback as 1	OFF State Readback as 0
Wet Contact (Sink)	+3.5 ~ +50 Vdc	OPEN or <1 Vdc
Wet Contact (Source)	+3.5 ~ +50 Vdc	OPEN or <1 Vdc

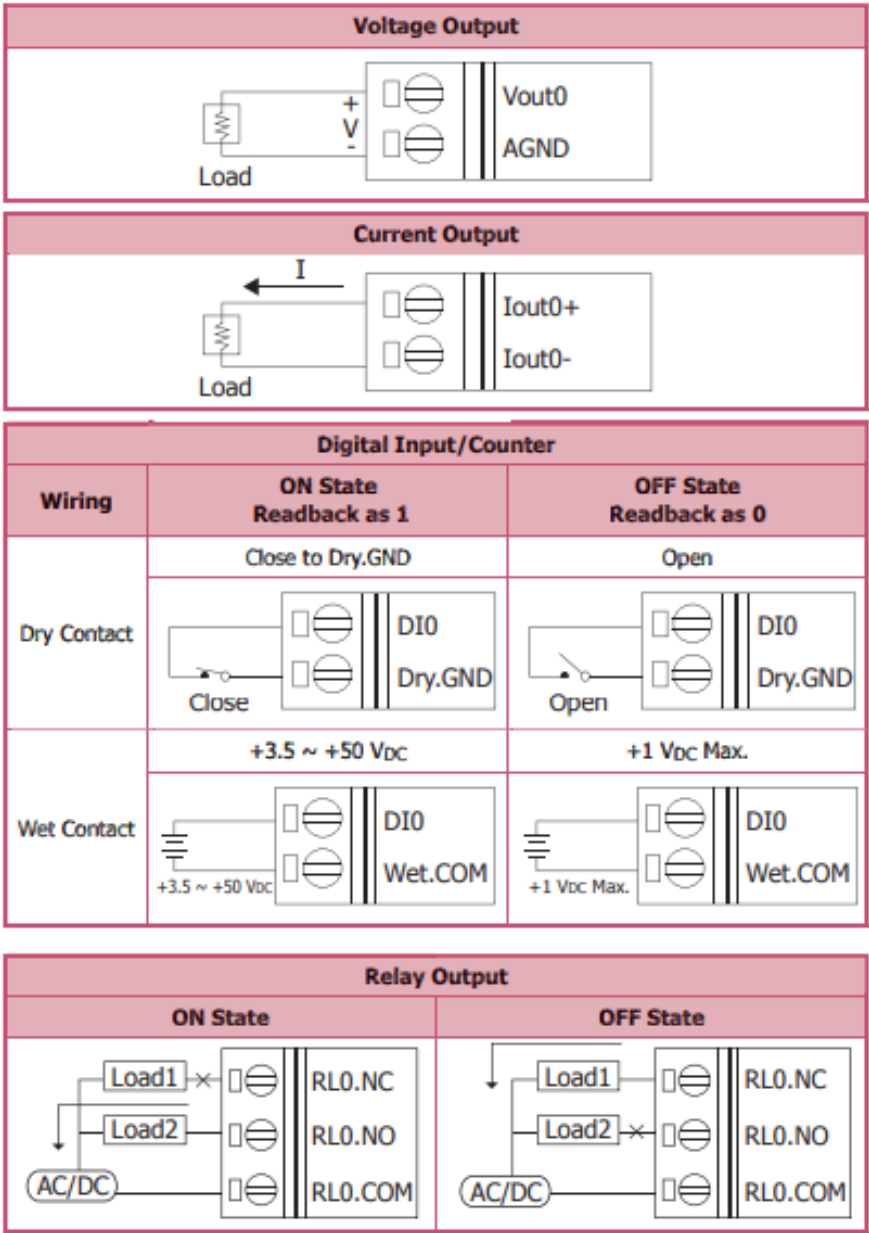
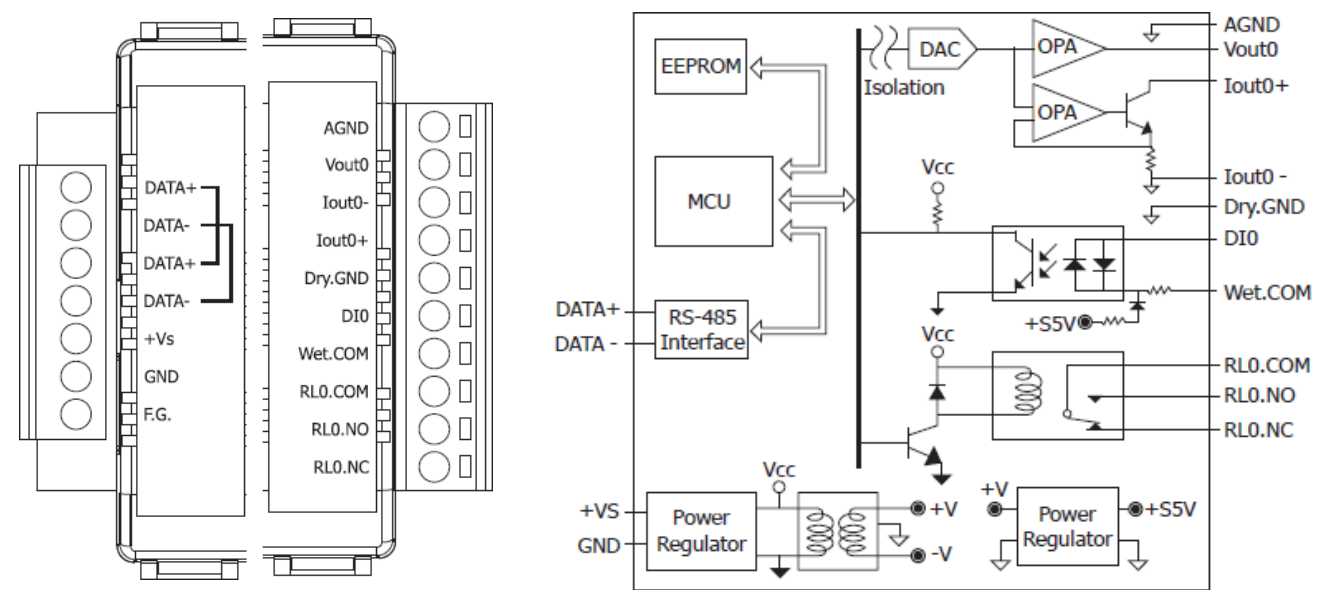
Input Type	ON State Readback as 1	OFF State Readback as 0
Dry Contact	Close to GND	Open

2.3.16 接線 – tM-C8

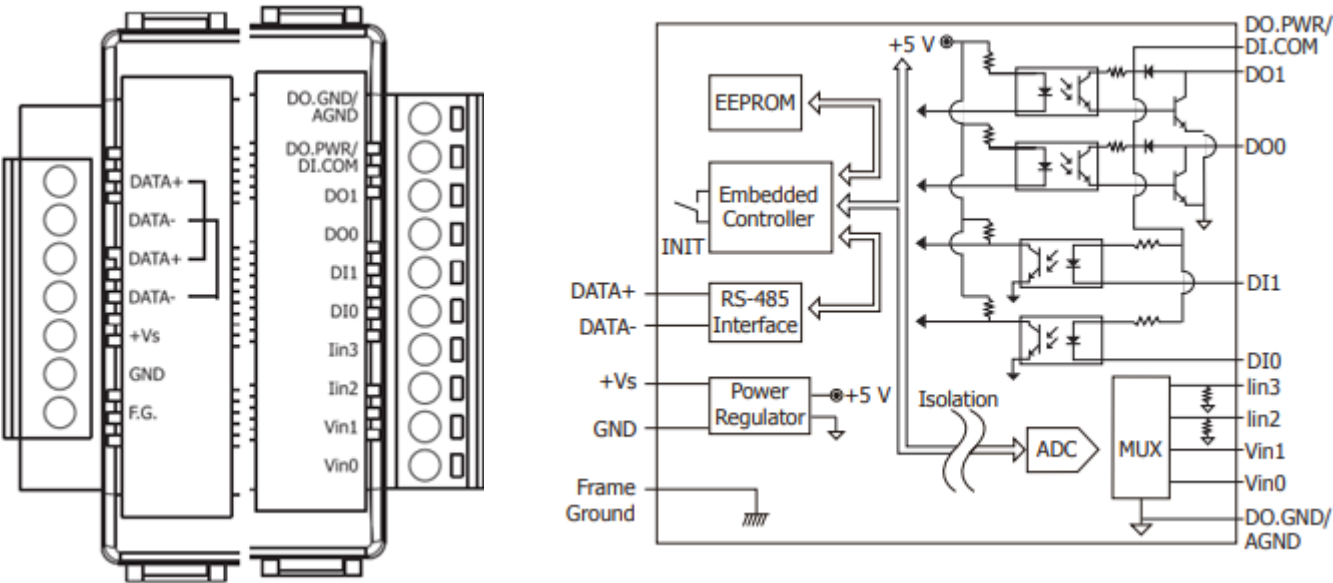


Output Type	ON State Readback as 1	OFF State Readback as 0
	Relay ON	Relay OFF
Drive Relay		
Resistance Load		

2.3.17 接線 – tM-DA1P1R1



2.3.18 接線 – tM-AD2P2C2



Voltage Input for Vin0 and Vin1	Current Input for Vin0 and Vin1
	Requires Optional External 125 Ω Resistor

Current Input for Iin2 and Iin3

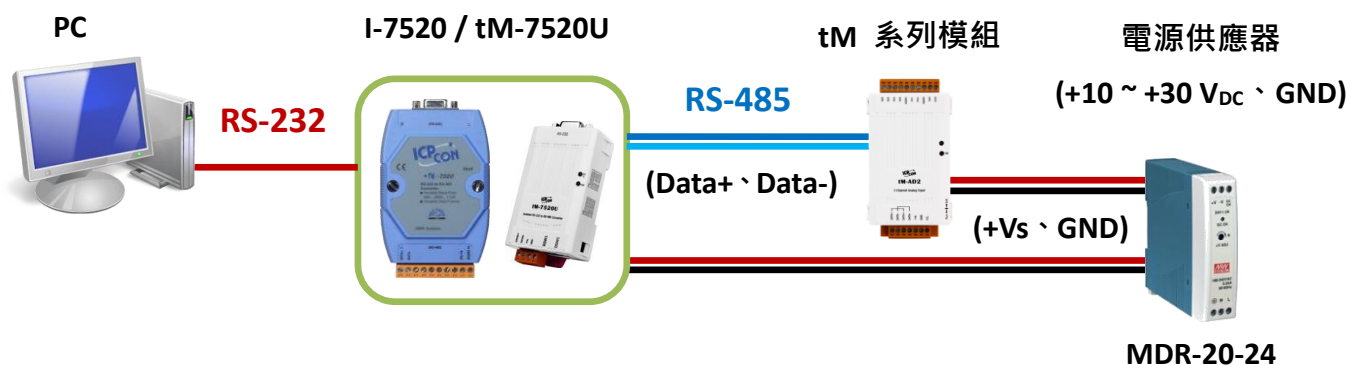
Digital Input/Counter	ON State Readback as 1	OFF State Readback as 0
	+3.5 ~ +50 V _{DC}	OPEN or <1 V _{DC}
Source		

Output Type	ON State Readback as 1	OFF State Readback as 0
	Relay ON	Relay OFF
Drive Relay		
Resistance Load		

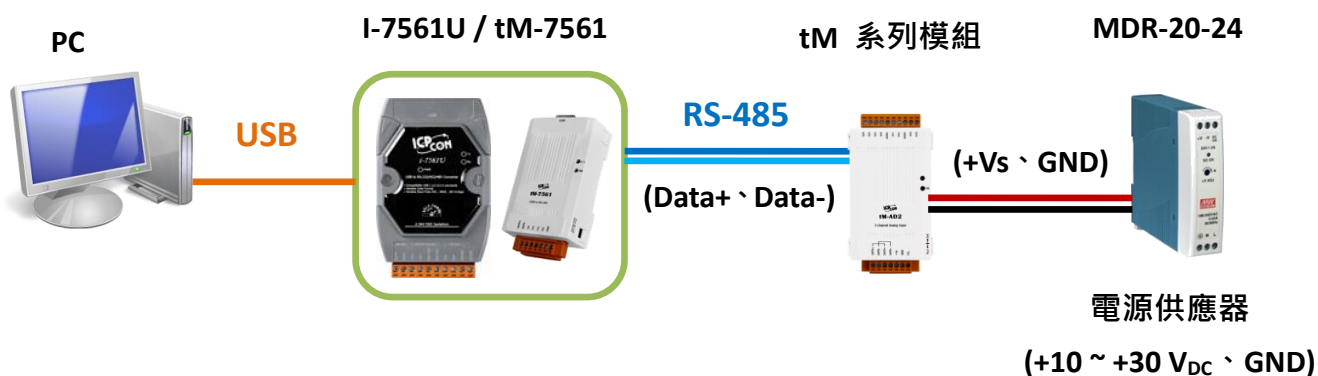
2.4 硬體連接

在使用 tM 系列模組之前，需先以 "DCON Utility Pro" 為每一個模組進行設定。tM 系列模組配備有 RS-485 通訊埠，可透過 RS-232 (或 USB) 轉 RS-485 轉換器 與 PC 進行連線。

1) 使用 RS-232 轉 RS-485 轉換器



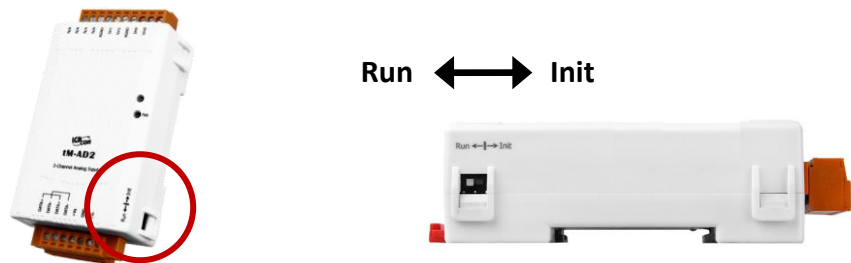
2) 使用 USB 轉 RS-485 轉換器



2.5 操作模式開關 與 通訊參數

tM 系列模組提供 2 種操作模式 (Run / Init)，可由機殼上的開關位置來決定。在模組的右側有一個操作模式選擇開關。

操作模式選擇開關

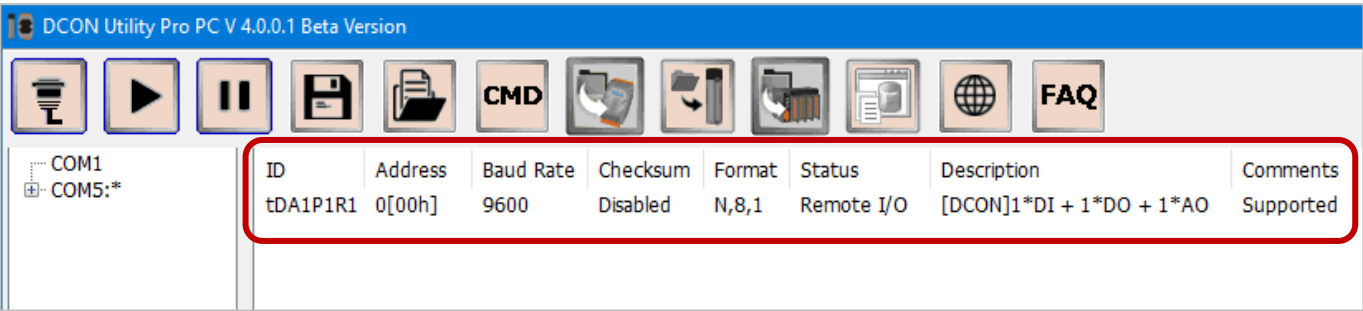


Run 模式:

tM 系列模組的操作模式開關，一般需設在 Run 的位置。

Init 模式:

若您不曉得或忘記模組的通訊參數時，可將開關撥至 “Init” 的位置再啟動電源，此時，模組的通訊參數固定為 Protocol: DCON, Address: 0, Baud Rate: 9600 (N,8,1), Checksum: Disable。



預設通訊參數:

通訊參數	出廠預設值 (Run 模式)	初始值固定 (INIT 模式)
Protocol	Modbus RTU	DCON
Address	1	0
Baud Rate	9600 bps	
Parity	n,8,1-no parity	
Checksum	Disable	

第 3 章 軟體工具 - DCON Utility Pro

在開始使用 tM 系列模組前，需先以 "DCON Utility Pro" 設定每一個模組的通訊參數 與 相關 I/O 設定 (依需求來設定) 。"DCON Utility Pro" 是一套支援 DCON 與 Modbus RTU/ASCII 通訊協定的工具軟體，可讓使用者輕易地搜尋、設定並測試 I/O 模組。

您可在網頁下載此軟體 與 使用手冊。

https://www.icpdas.com/en/product/guide+Software+Utility_Driver+DCON__Utility__Pro

(註: 此軟體可在 PC 或 PAC 上使用；此版為 3.0 版)

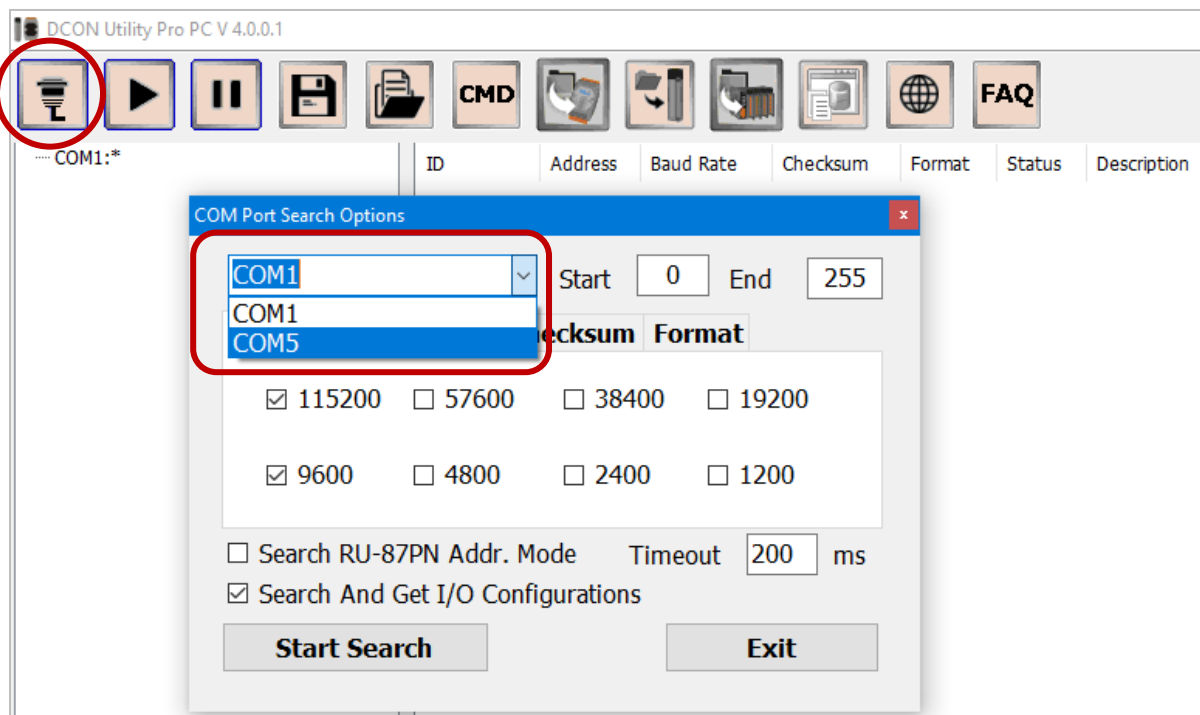


3.1 搜尋 I/O 模組

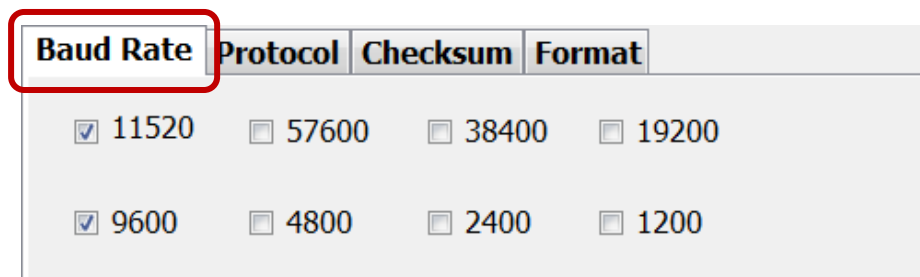
此例，使用 tM-7561 (USB to RS-485 轉換器) 來連接 PC (COM Port) 與 模組 (RS-485)。

(註: 請確認 PC 與 模組已連接，且電源供應器已開啟。)

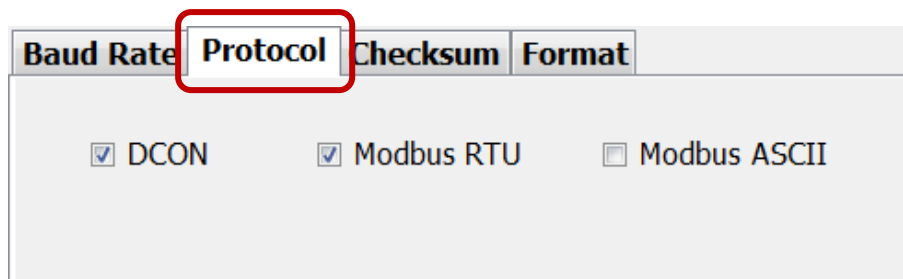
步驟 1: 選擇 PC 採用的 COM Port 與 搜尋條件 (Baud Rate, Protocol, Checksum 與 Format)。



Baud Rate: 模組的 Baud Rate 預設為 9600，您也可勾選多個選項。

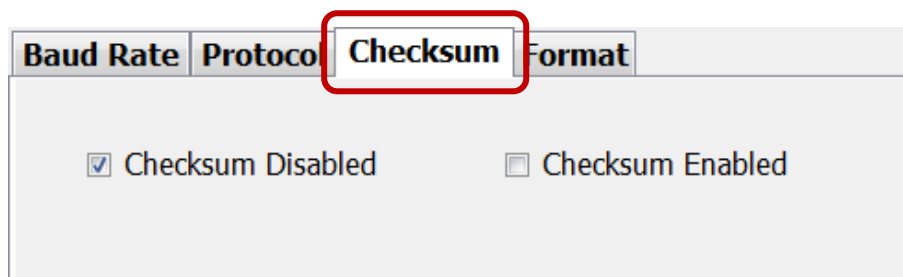


Protocol: 若以 Init 模式開機，模組的 Protocol 固定為 DCON。
 若以 Run 模式開機，模組的 Protocol 預設為 Modbus RTU。



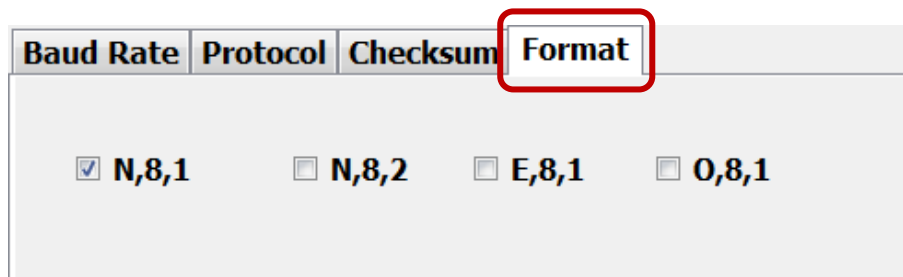
Baud Rate	Protocol	Checksum	Format
	<input checked="" type="checkbox"/> DCON	<input checked="" type="checkbox"/> Modbus RTU	<input type="checkbox"/> Modbus ASCII

Checksum: 預設為 “Disabled”。



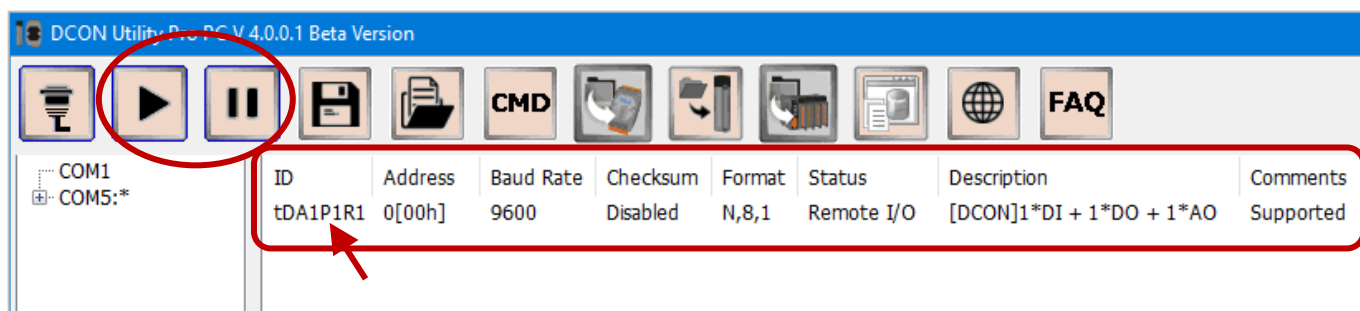
Baud Rate	Protocol	Checksum	Format
		<input checked="" type="checkbox"/> Checksum Disabled	<input type="checkbox"/> Checksum Enabled

Format: 預設為 “N,8,1”。



Baud Rate	Protocol	Checksum	Format
			<input checked="" type="checkbox"/> N,8,1
			<input type="checkbox"/> N,8,2
			<input type="checkbox"/> E,8,1
			<input type="checkbox"/> O,8,1

步驟 2: 設定完搜尋條件，請點選 “Start Search” (▶) 開始尋找模組，找到模組後點選 “Stop Search” (⏏) 停止搜尋，並點選型號 (ID) 開啟設定視窗。



ID	Address	Baud Rate	Checksum	Format	Status	Description	Comments
tDA1P1R1	0[00h]	9600	Disabled	N,8,1	Remote I/O	[DCON]1*DI + 1*DO + 1*AO	Supported

3.2 設定頁面 – Configuration

搜尋並找到模組後，接著可設定或測試 I/O 模組。

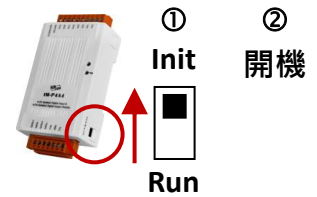
註: DCON Utility Pro V3 (以上) 可支援在 Run 模式開啟多個模組的設定視窗。

在 DCON Utility Pro 中，每個 tM 模組都有一個 “Configuration” 設定頁面，可用來設定 Protocol、Address、Baud Rate...等通訊參數。由於各模組的設定畫面，不盡相同，以下將會依據 I/O 分類來進行說明。

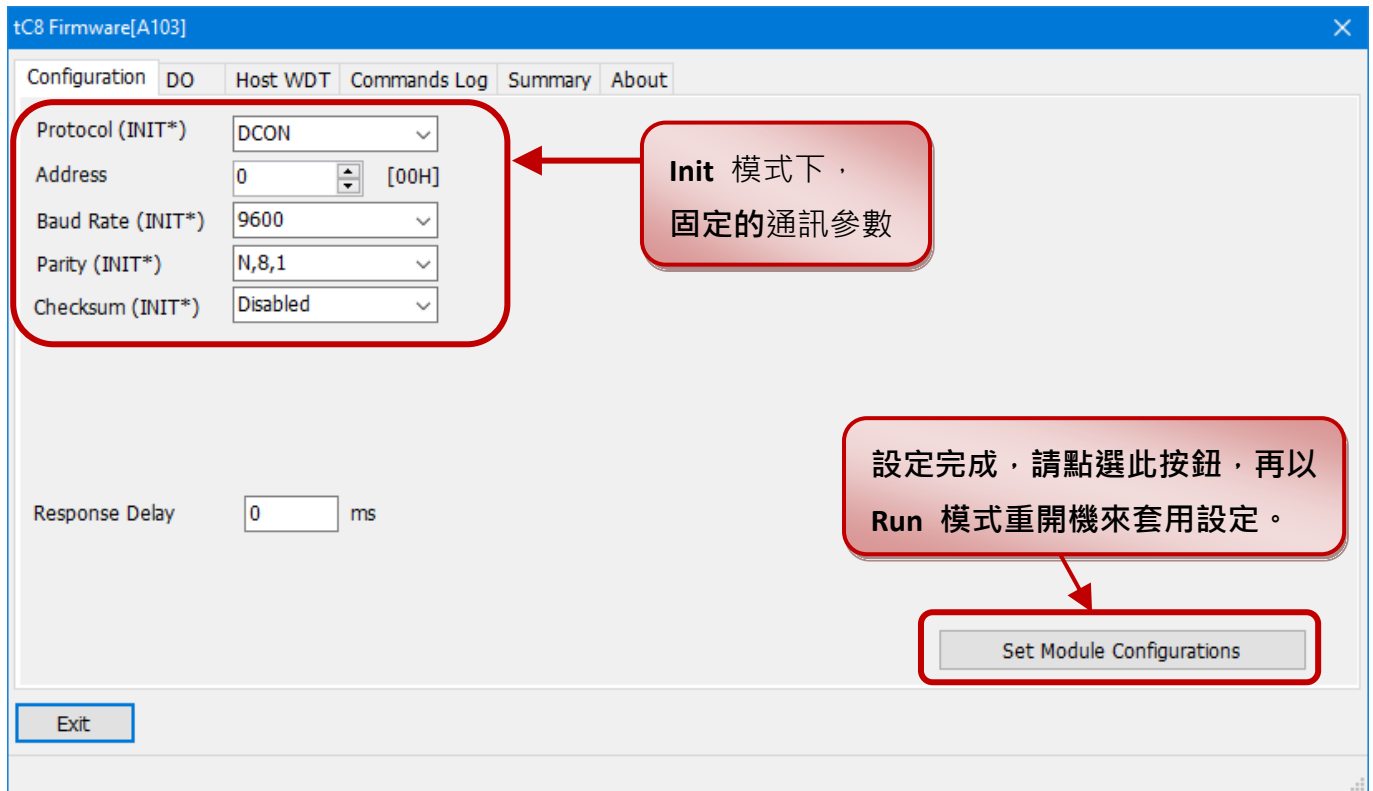
型號	說明	章節
tM-P3R3 tM-PD3R3	3 通道 數位輸入 與 3 通道 繼電器輸出模組。	3.2.1
tM-P3POR3	3 通道 數位輸入 與 3 通道 光耦合繼電器輸出模組。	
tM-P4A4	4 通道 數位輸入 與 4 通道 數位輸出模組。	
tM-P4C4	4 通道 數位輸入 與 4 通道 數位輸出模組。	
tM-R5	5 通道 繼電器輸出模組。	
tM-P8 tM-PDW8	8 通道 數位輸入模組。	
tM-C8	8 通道 數位輸出模組。	3.2.2
tM-AD2	2 通道 電壓/電流輸入模組。	
tM-AD4P2C2	4 通道 類比輸入、2 通道 數位輸入 與 2 通道 繼電器輸出模組。	3.2.3
tM-AD5	5 通道 差動電壓輸入模組。	
tM-AD8	8 通道 單端電壓輸入模組。	
tM-AD5C	5 通道 差動電流輸入模組。	
tM-AD8C	8 通道 單端電流輸入模組。	
tM-TH8	8 通道 熱敏電阻輸入模組。	3.2.4
tM-DA1P1R1	1 通道 類比輸出、1 通道 數位輸入 與 1 通道 繼電器輸出模組。	

3.2.1 設定通用項目 (適用模組: tM-P3R3, PD3R3, P3POR3, P4A4, P4C4, R5, C8, P8, PDW8)

在 Init 模式下，會採用固定的通訊參數 (如下圖) 來搜尋模組，您可以在 “Configuration” 頁面中，點選 “Set Module Configurations” 按鈕，再以 Run 模式 重開機，才會套用這些設定值。



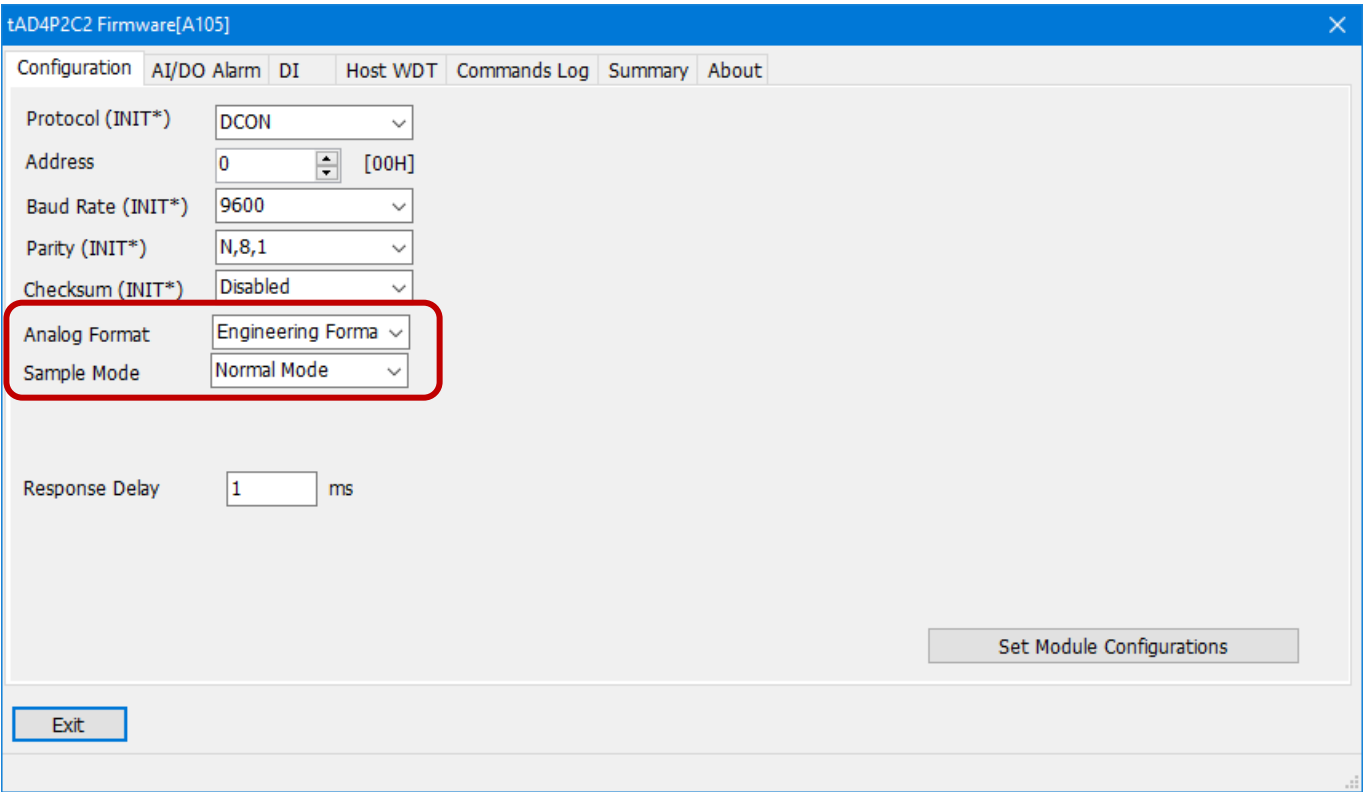
注意: 只有 **Protocol = DCON** 時，需以 Init 模式 設定標示有 (INIT*) 的項目。若已在 Run 模式，仍可直接將開關切換到 Init 再設定。完成後，需切回 Run 再重新上電。



Protocol	設定通訊協定 可選用 DCON、Modbus RTU 與 Modbus ASCII
Address	設定模組唯一的識別位址 (範圍: 0-255)
Baud Rate	設定通訊速率 可選用 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
Parity	設定同位元，可選用 N,8,1-None Parity、N,8,2-None Parity、E,8,1-Even Parity 與 O,8,1-Odd Parity
Checksum	設定檢查碼，可選用 Disabled/ Enabled 註: 使用 Modbus Protocol 時，固定設為 Disabled
Respond Delay	設定回應的延遲時間 (範圍: 0-30 ms)

3.2.2 設定類比項目 (適用模組: tM-AD2, AD4P2C2)

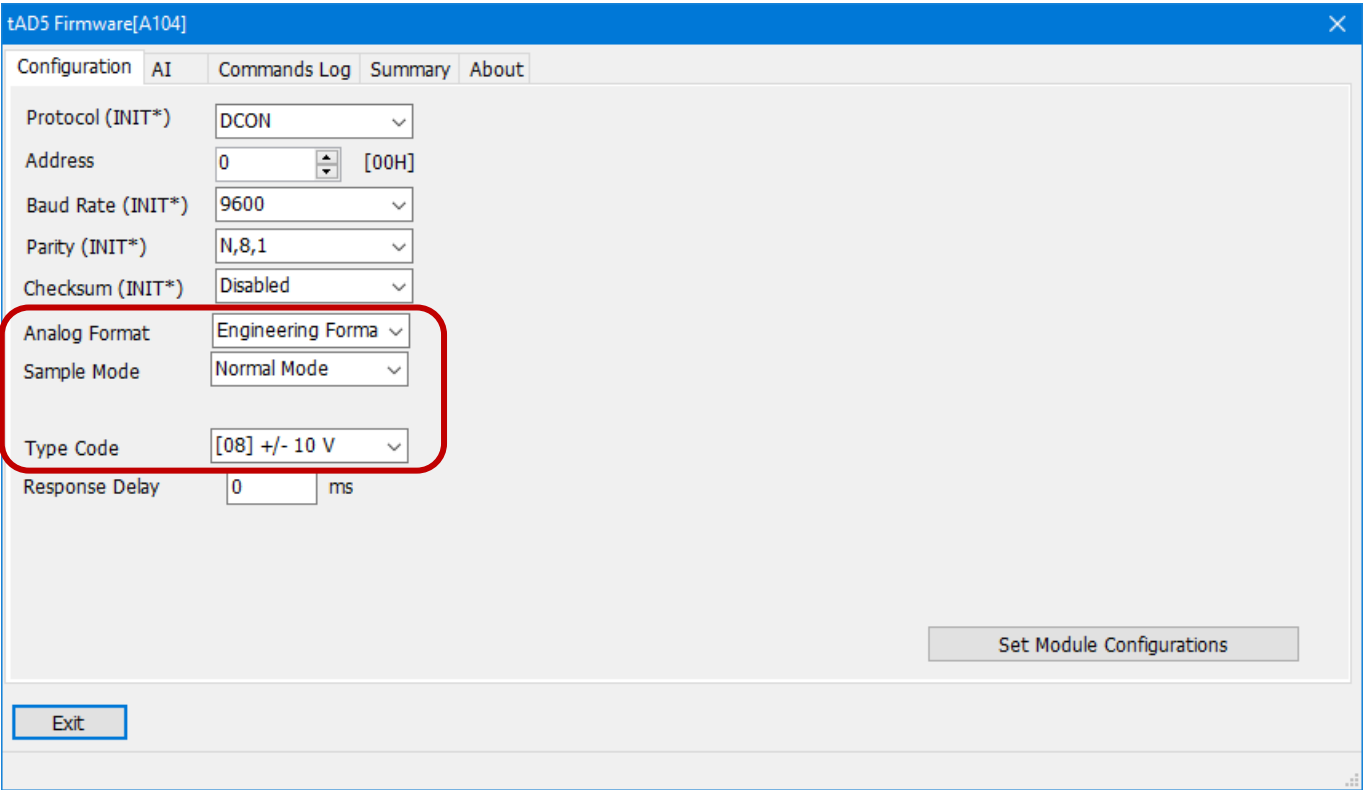
在 “Configuration” 頁面，包含通訊參數 (3.2.1 節) 與 類比資料的設定項目。



Analog Format	設定類比資料的格式 <u>DCON Protocol</u> : 可選用 Engineering Format 、Percent Format 或 2’s Complement Format <u>Modbus RTU/ASCII Protocol</u> : 可選用 Engineering Format 或 2’s Complement Format
Sample Mode	設定資料取樣的模式 可選用 Normal Mode 與 Fast Mode 。

3.2.3 設定類比項目 (適用模組: tM-AD5, AD8, AD5C, AD8C)

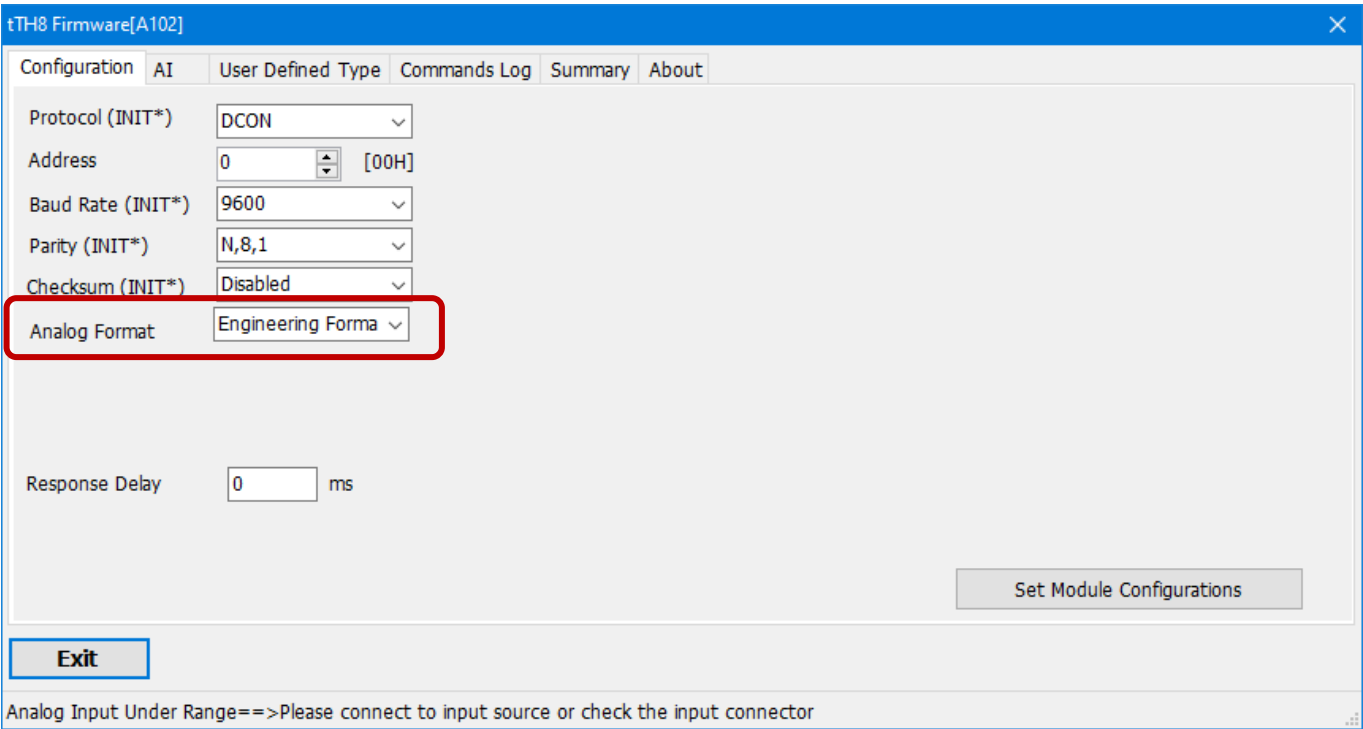
在 “Configuration” 頁面，包含通訊參數 (3.2.1 節) 與 類比資料的設定項目。



Analog Format	設定類比資料的格式 <u>DCON Protocol</u> : 可選用 Engineering Format 、Percent Format 或 2’s Complement Format <u>Modbus RTU/ASCII Protocol</u> : 可選用 Engineering Format 或 2’s Complement Format
Sample Mode	設定資料取樣的模式 可選用 Normal Mode 與 Fast Mode
Type Code	設定資料的類型與範圍 <u>AD5</u> : +/- 1 V 、 +/- 2.5 V 、 +/- 5 V 、 +/- 10 V <u>AD8</u> : 0 ~ 500 mV 、 0 ~ 1 V 、 0 ~ 2.5 V 、 0 ~ 5 V 、 0 ~ 10 V <u>AD5C</u> : +/- 20 mA 、 0 ~ 20 mA 、 4 ~ 20 mA <u>AD8C</u> : 0 ~ 20 mA 、 4 ~ 20 Ma (參考: 附錄 B.2 ~ B.5 資料範圍)

3.2.4 設定類比項目 (適用模組: tM-TH8, DA1P1R1)

在 “Configuration” 頁面，包含通訊參數 (3.2.1 節) 與 類比資料的設定項目。



Analog Format	<p>設定類比資料的格式，</p> <p><u>DCON Protocol:</u></p> <p>tM-TH8: 可選用 Engineering Format、Percent Format、2’s Complement Format 或 Ohms</p> <p>tM-DA1P1R1: 可選用 Engineering Format、Percent Format 或 2’s Complement Format</p> <p><u>Modbus RTU/ASCII Protocol:</u></p> <p>tM-TH8, tM-DA1P1R1: 可選用 Engineering Format 或 2’s Complement Format</p>
---------------	---

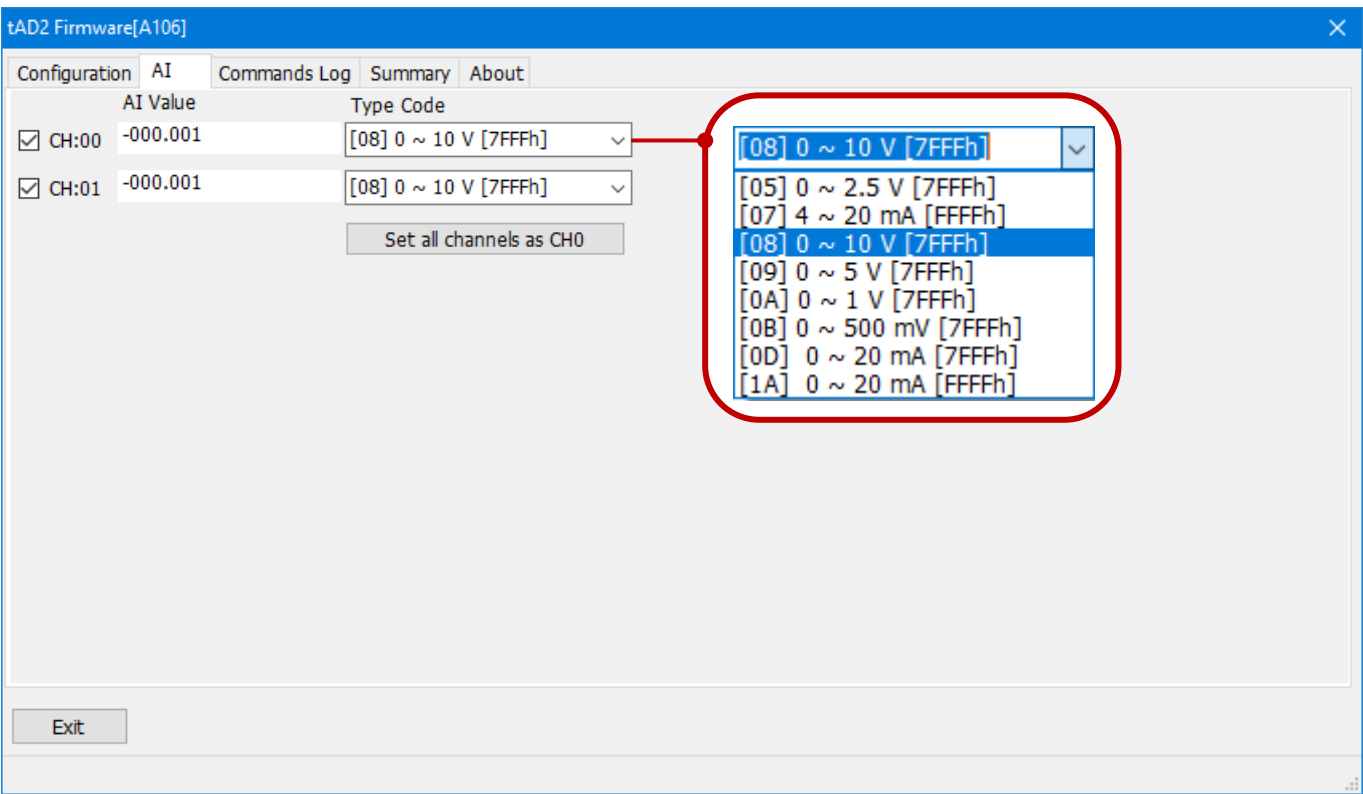
3.3 設定頁面 – AI

“AI” 頁面，可用來偵測/設定模組的類比輸入通道。

型號	說明	章節
tM-AD2	具有 2 個 電壓/電流輸入通道	3.3.1
tM-AD5	具有 5 個 電壓輸入通道	3.3.2
tM-AD8	具有 8 個 電壓輸入通道	
tM-AD5C	具有 5 個 電流輸入通道	
tM-AD8C	具有 8 個 電流輸入通道	
tM-TH8	具有 8 個 熱敏電阻輸入通道	3.3.3

3.3.1 設定頁面 01 - AI (適用模組: tM-AD2)

在 “AI” 頁面，可偵測各通道的 AI 值 並指定資料類型與範圍。

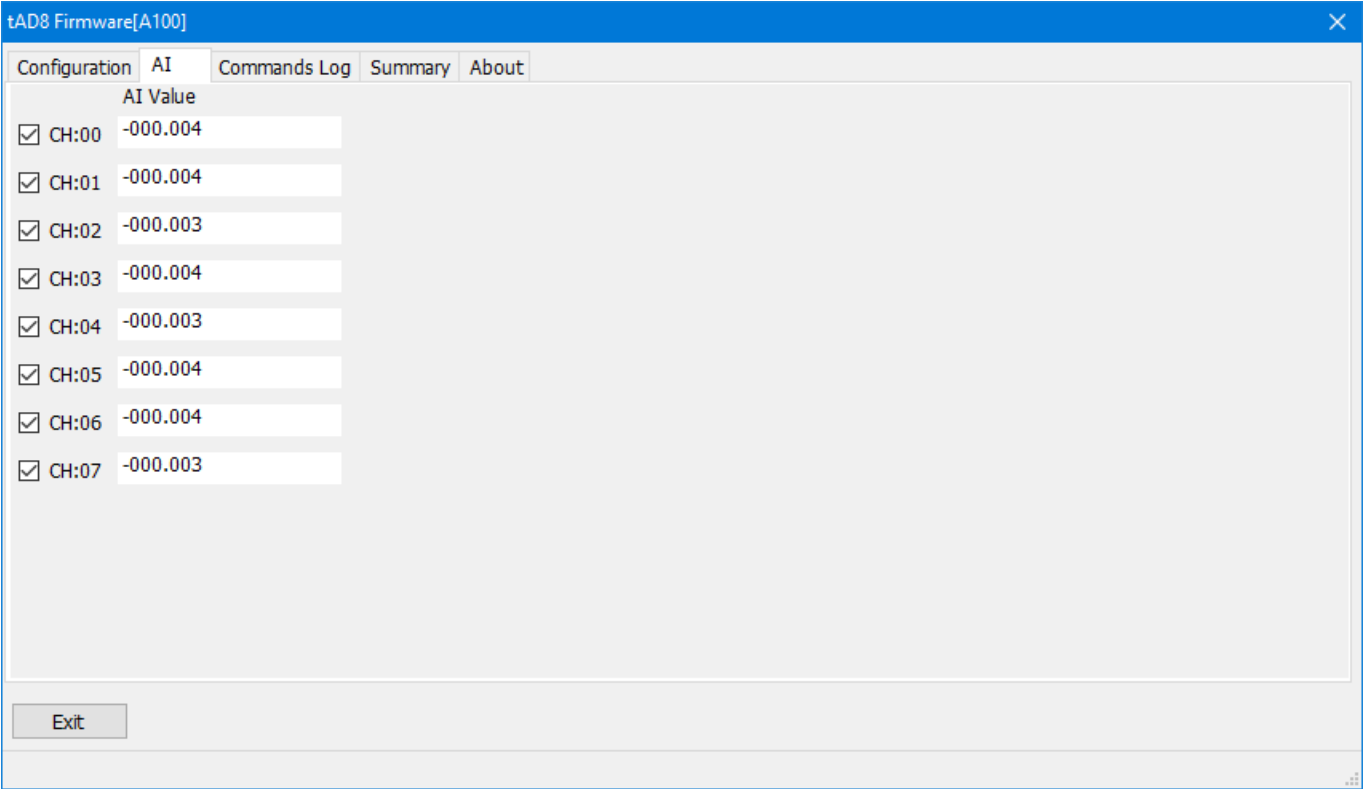


CH: xx	勾選可偵測該通道目前的數值 (即，AI Value)
Type Code	設定資料的類型與範圍 (tM-AD2: 0 ~ 500 mV、0 ~ 1 V、0 ~ 2.5 V、0 ~ 5 V、0 ~ 10 V、0 ~ 20 mA、4 ~ 20 mA，參考: 附錄 B.1 資料範圍)
Set all channels as CH0	將所有通道設為和 CH:00 一樣的 Type Code

3.3.2 設定頁面 02 - AI (適用模組: tM-AD5, AD8, AD5C, AD8C)

在 “AI” 頁面，可偵測各通道的 AI 值。

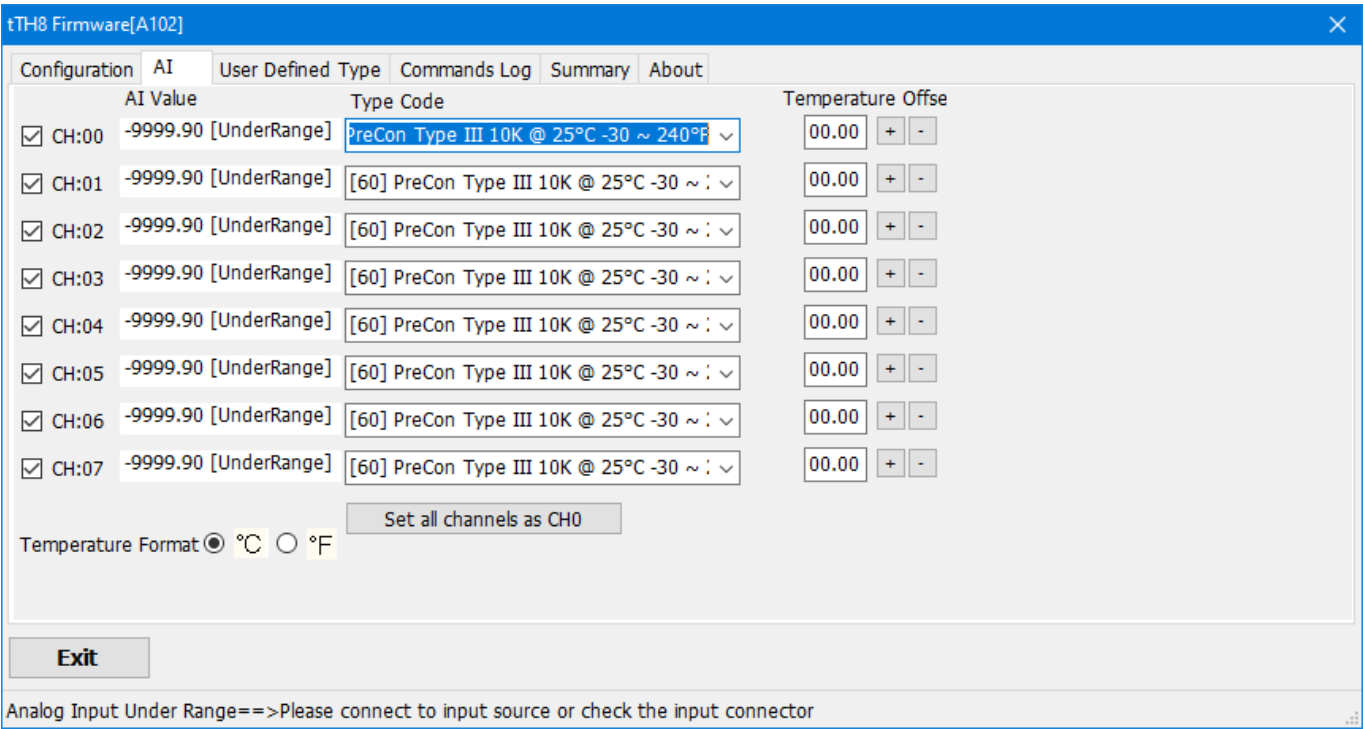
註: tM-AD5、tM-AD5C 有 5 個類比輸入通道 (即，CH:00 ~ CH:04)。
tM-AD8、tM-AD8C 有 8 個類比輸入通道 (即，CH:00 ~ CH:07)。



CH: xx	勾選可偵測該通道目前的數值 (即，AI Value)		
	DCON, Modbus RTU/ASCII		
	Protocol	Analog Format	Under Range
	DCON	Engineering	-9999.9
	Modbus		-32768
	DCON, Modbus	2's Complement	8000
	DCON	Percent	-999.99
註: 只適用於量測範圍為 0-20 mA 和 4-20 mA。			

3.3.3 設定頁面 03 - AI (適用模組: tM-TH8)

在 “AI” 頁面，可偵測各通道的 AI 值 並指定溫度相關設定。



CH: xx	<div>勾選可偵測該通道目前的數值 (即，AI Value)</div> <div>DCON, Modbus RTU/ASCII</div> <table><tr><th>Protocol</th><th>Analog Format</th><th>Under Range</th><th>Over Range</th></tr><tr><td>DCON</td><td rowspan="2">Engineering</td><td>-9999.9</td><td>+9999.9</td></tr><tr><td>Modbus</td><td>-32768</td><td>32767</td></tr><tr><td>DCON, Modbus</td><td>2's Complement</td><td>8000</td><td>7FFF</td></tr><tr><td>DCON</td><td>Percent</td><td>-999.99</td><td>+999.99</td></tr></table>	Protocol	Analog Format	Under Range	Over Range	DCON	Engineering	-9999.9	+9999.9	Modbus	-32768	32767	DCON, Modbus	2's Complement	8000	7FFF	DCON	Percent	-999.99	+999.99
Protocol	Analog Format	Under Range	Over Range																	
DCON	Engineering	-9999.9	+9999.9																	
Modbus		-32768	32767																	
DCON, Modbus	2's Complement	8000	7FFF																	
DCON	Percent	-999.99	+999.99																	
Type Code	設定熱敏電阻 (Thermistor) 的類型與溫度範圍 (參考: 附錄 B.8 資料範圍)																			
Set all channels as CH0	將所有通道設為和 CH:00 一樣的 Type Code																			
Temperature Offset	設定偏移量，來調整溫度的讀值																			
Temperature Format	設定溫度單位 可選用攝氏 (°C) 或 華氏 (°F)																			

3.4 設定其它 I/O 頁面

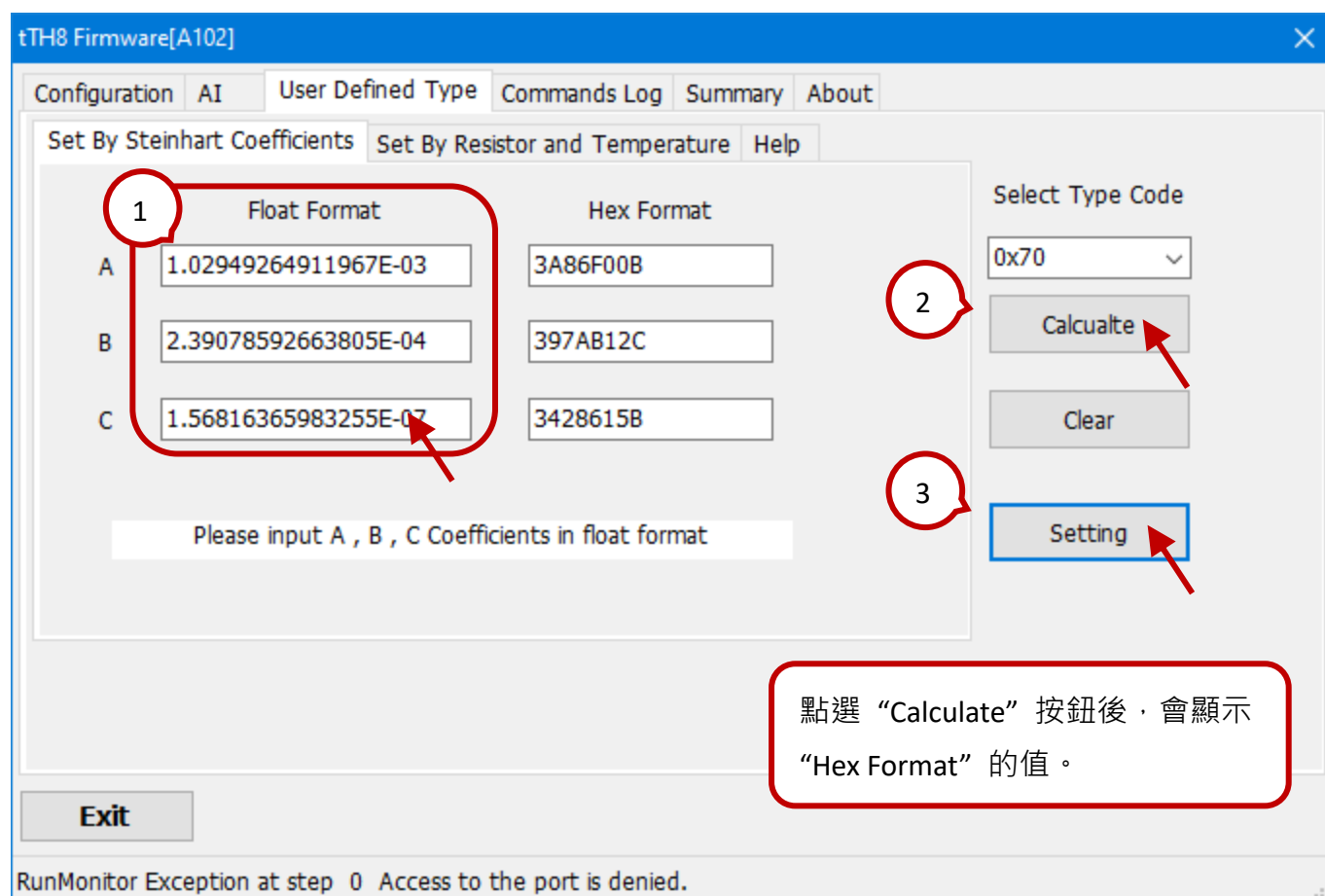
3.4.1 設定頁面 - User Defined Type

tM-TH8 是一款熱敏電阻 (Thermistor) 輸入模組，“User Defined Type” 是一個客製化頁面，以下將說明使用的方式。

1. 若已有製造商提供之熱敏電阻的 Steinhart 係數，可直接在 “Set By Steinhart Coefficients” 頁面，填入 A、B、C 三種數據 (Float Format)，點選 “Calculate” 再點選 “Setting” 按鈕。

例如：有一個熱敏電阻，元件編號為 YSI H Mix 10000，以下是 Steinhart 係數表。

Coefficient	Float Format	Hex Format
A	1.02949264911967E-03	3A86F00B
B	2.39078592663805E-04	397AB12C
C	1.56816365983255E-07	3428615B



2. 另外，您也可查詢熱敏電阻的對應表 (電阻-溫度)，在 “Set By Resistor and Temperature” 頁面填入 3 組電阻值 (R) 與 溫度值 (°C)，並點選 “Calculate” 按鈕，來算出 Steinhart 係數。

建議依照以下規則來設定，以得到精確的結果。

$$(1) -40^{\circ}\text{C} \leq T_1, T_2, T_3 \leq 150^{\circ}\text{C} \quad (2) |T_2 - T_1| \leq 50^{\circ}\text{C} \quad (3) |T_3 - T_2| \leq 50^{\circ}\text{C}$$

例如:

電阻值 (Ω)	溫度 ($^{\circ}\text{C}$)
R1: 29490	T1: 0
R2: 3893	T2: 50
R3: 816.8	T3: 100

注意：如果電阻大於 204700 歐姆，則視為低於範圍 (Under Range)。

tTH8 Firmware[A102]

Configuration AI User Defined Type Commands Log Summary About

Set By Steinhart Coefficients **Set By Resistor and Temperature** Help

	Resistor Value		Temperature $^{\circ}\text{C}$
R1(ohms)	29490	T1	0
R2(ohms)	3893	T2	50
R3(ohms)	816.8	T3	100

Please input resistor and temperature values

Select Type Code: 0x70

Calcualte Clear Setting

Exit

Analog Input Under Range==>Please connect to input source or check the input connector

3. 此時，會自動切換到 “Set By Steinhart Coefficients” 頁面，並顯示計算的結果，請點選 “Setting” 按鈕來完成設置。

Set By Steinhart Coefficients **Set By Resistor and Temperature** Help

	Float Format	Hex Format
A	0.00102949264911967	3A86F00B
B	0.000239078592663805	397AB12C
C	1.56816365983256E-07	3428615B

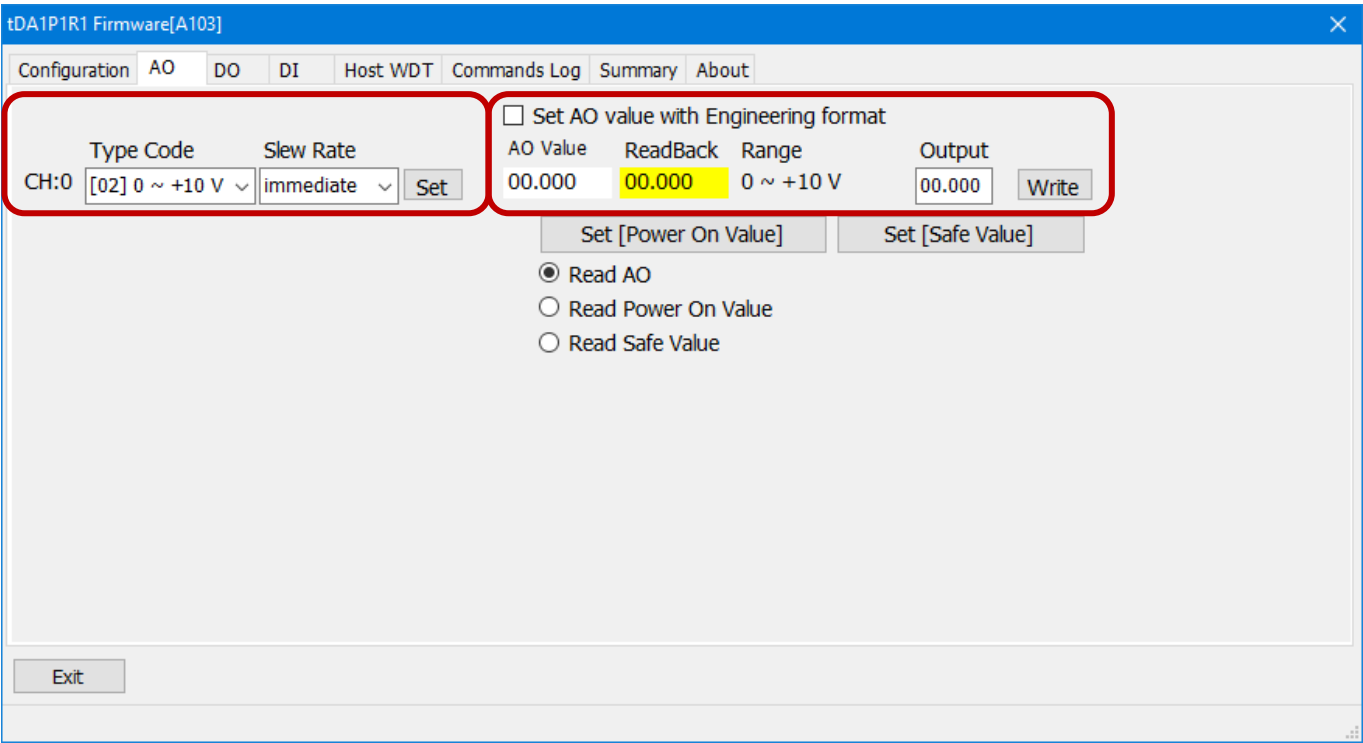
Please input A , B , C Coefficients in float format

Select Type Code: 0x70

Calcualte Clear Setting

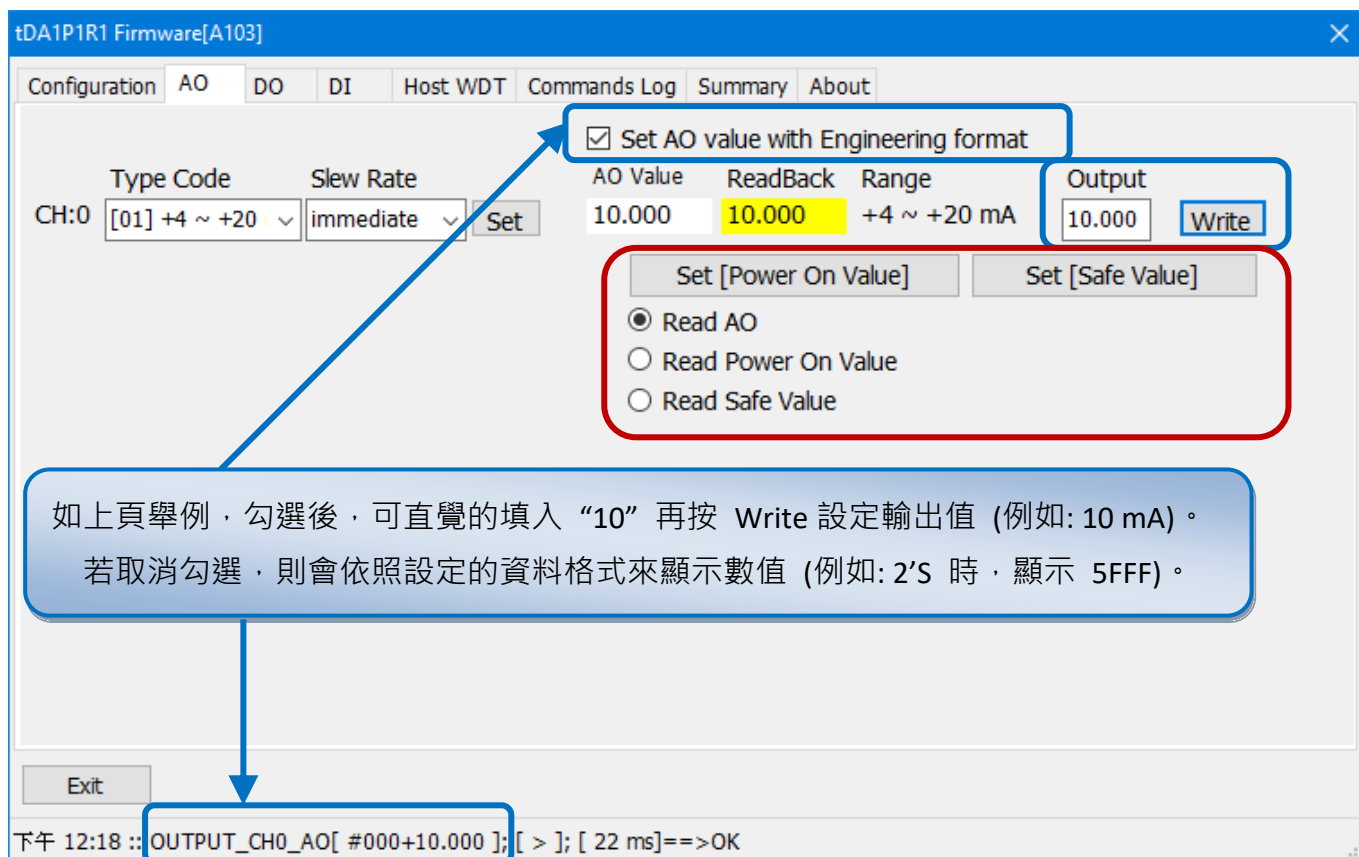
3.4.2 設定頁面 - AO

tM-DA1P1R1 具有 1 個類比輸出通道 (AO)。



Type Code	設定資料的類型與範圍 (tM-DA1P1R1: 0 ~ 10 V、0 ~ 20 mA、4 ~ 20 mA、參考: 附錄 B.7 資料範圍)
Slew Rate	設定電壓/電流轉換速率 (V/sec, mA/sec)，即每秒增加的電壓/電流 (參考: 附錄 A.4 轉換率控制)
Set	點選可套用 Type Code 或 Slew Rate 設定

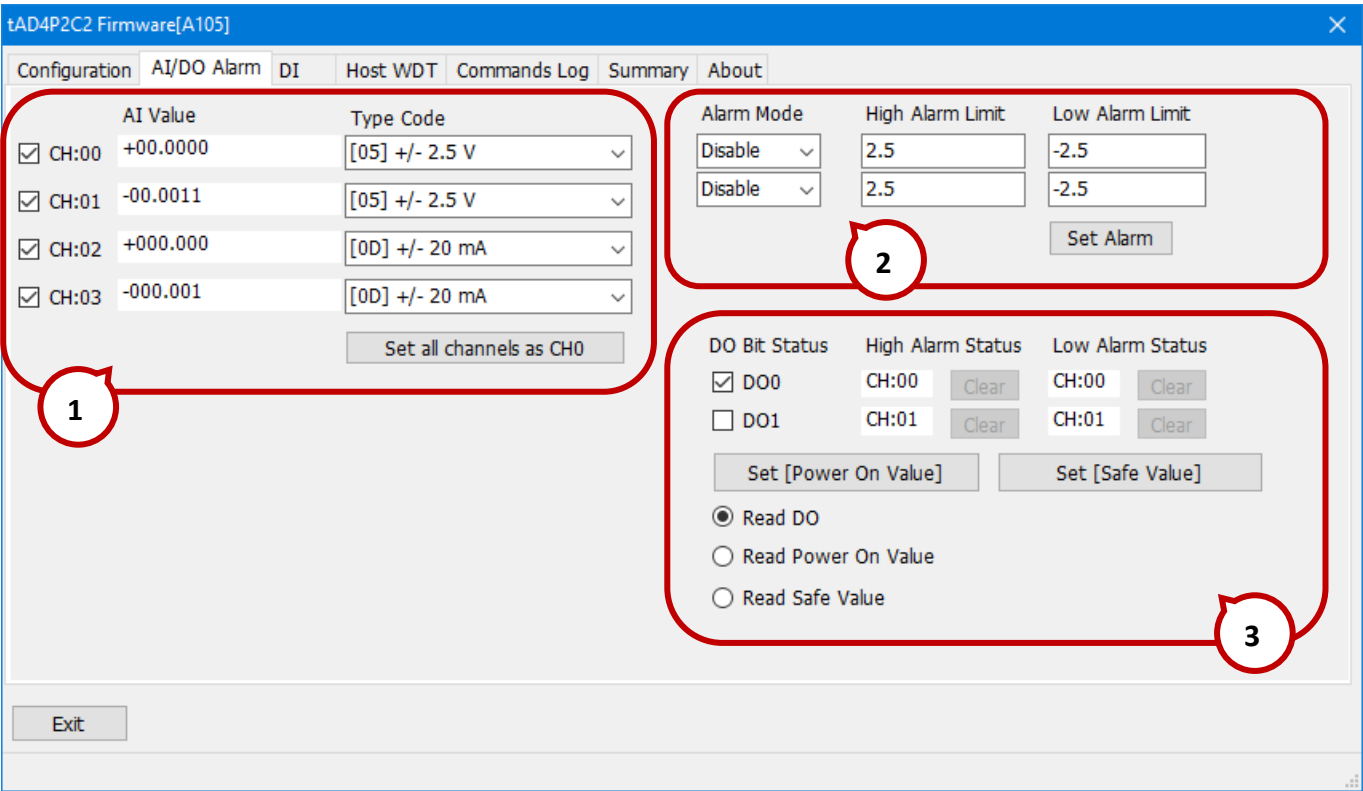
☑ Set AO value with the Engineering format: 1) <u>勾選</u> ，以工程 (Engineering) 格式，來顯示/設定 AO Value 與 ReadBack。 2) <u>未勾選</u> ，則依設定的資料格式來顯示 (參考: 3.2.4 節， “Configuration” 頁面 – Analog Format) 例如: 當 Type Code = +4 ~ +20mA, 且 Analog Format = 2’s Complement, 若要輸出 10 mA, 可勾選此選項，直接填入 10 再按 Write 按鈕；若取消勾選則會顯示 5FFF。	
AO Value	顯示設定的 AO 值、Power-on Value 或 Safe Value
ReadBack	顯示目前的 AO 值 (參考: 附錄 A.5 類比輸出回讀值) (註: 當有設定 Slew Rate 時，輸出值會變化，因此 Readback 和設定的 AO 值就可能不同。)
Range	顯示目前 Type Code 的數值範圍
Output / Write	設定輸出值，請點選 “Write” 按鈕套用設定



Set to Power-on Value	設定輸出值後，點選此按鈕套用設定 (參考: 附錄 A.2 雙看門狗功能)
Set to Safe Value	
Read AO	點選可讀取 AO 值
Read Power-on Value	點選可讀取目前的 Power-on Value 或 Safe Value (此時會無法設定 Power-on Value 與 Safe Value)
Read Safe Value	

3.4.3 設定頁面 - AI/DO Alarm

tM-AD4P2C2 具有 4 個類比輸入通道 (AI) 與 2 個繼電器輸出通道 (Relay Output) ，您可將 Relay 接上警報燈/警報器，並設定其警報模式 與 上/下限值。



CH: xx	勾選可偵測該通道目前的 AI Value
Type Code	設定資料的類型與範圍 tM-AD4P2C2: +/- 1 V 、 +/- 2.5 V 、 +/- 5 V 、 +/- 10 V 、 +/- 20 mA 、 0 ~ 20 mA 、 4 ~ 20 mA (參考: 附錄 B.6 資料範圍)
Set all channels as CH0	將所有通道設為和 CH:00 一樣的 Type Code

當 Alarm Mode = Disable 時，才可勾選 DOx 核取方塊。

DOx	用來設定 DO 值、Power-on Value 或 Safe Value
Set to Power-on Value	勾選 DOx 後，再點選此按鈕，套用設定值 (參考: 附錄 A.2 雙看門狗功能)
Set to Safe Value	
Read DO	讀取 DO 狀態值
Read Power-on Value	點選可讀取目前的 Power-on Value 或 Safe Value (此時會無法設定 Power-on Value 與 Safe Value)
Read Safe Value	

tAD4P2C2 Firmware[A105]

Configuration AI/DO Alarm DI Host WDT Commands Log Summary About

	AI Value	Type Code	Alarm Mode	High Alarm Limit	Low Alarm Limit
<input checked="" type="checkbox"/> CH:00	+000.002	[08] +/- 10 V	Latch	10	2
<input checked="" type="checkbox"/> CH:01	+000.001	[08] +/- 10 V	Momentar	10	3
<input checked="" type="checkbox"/> CH:02	-000.001	[0D] +/- 20 mA			
<input checked="" type="checkbox"/> CH:03	-000.001	[0D] +/- 20 mA			

Set all channels as CH0

DO Bit Status

☒ DO0

☒ DO1

Set [Power On Value] Set [Safe Value]

☒ Read DO

☐ Read Power On Value

☐ Read Safe Value

High Alarm Status

CH:00 Clear

CH:01 Clear

Low Alarm Status

CH:00 Clear

CH:01 Clear

Exit

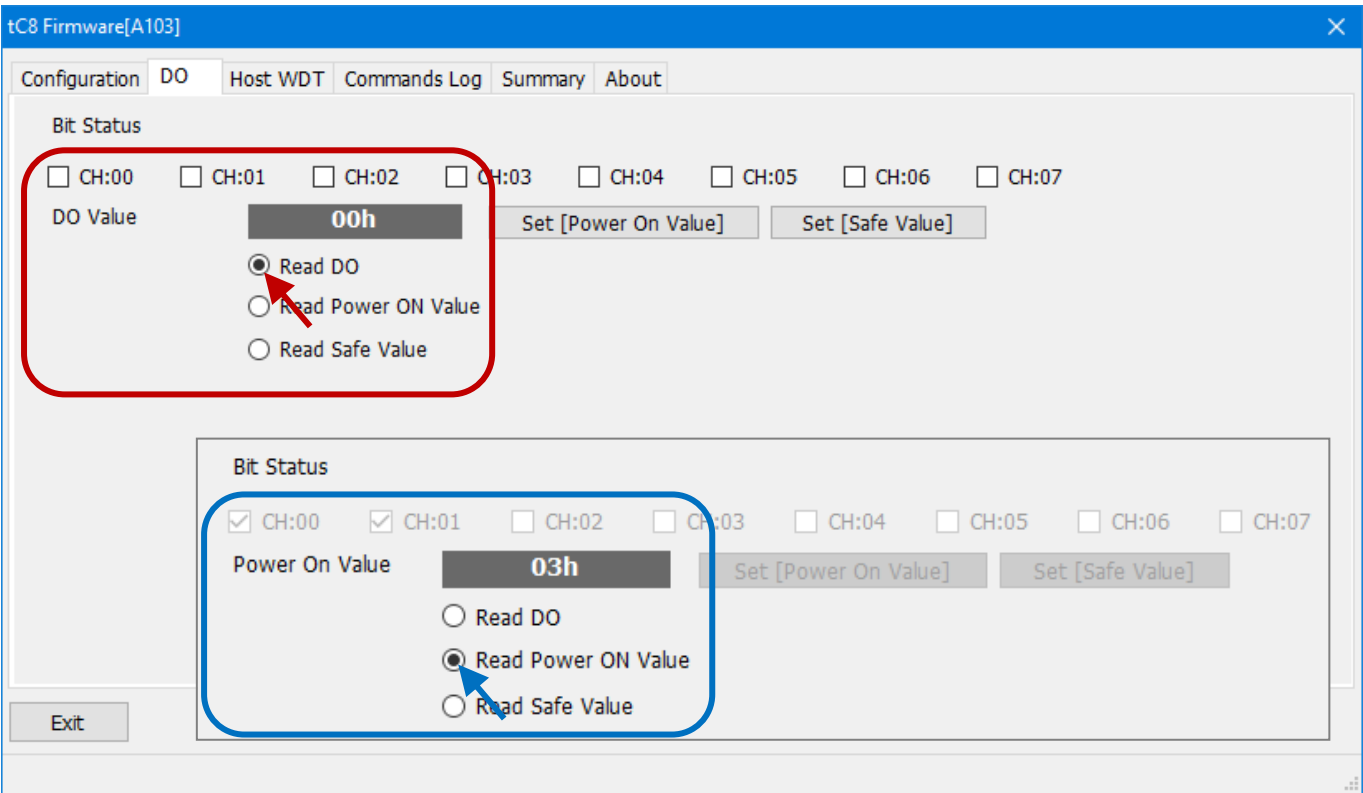
下午 01:47 ::GET_CH1_ALARM_MODE[@00RAC1]; [!021]; [15 ms]==>OK

Alarm Mode	<p>設定警報模式，可選擇</p> <p>Disable: 不啟用</p> <p>Momentary: 若超出警界值，會產生警報</p> <p>Latch: 若超出警界值，會產生警報，直到手動清除警報</p>
High Alarm Limit	設定 Type Code 後，可設定警報的上限值
Low Alarm Limit	設定 Type Code 後，可設定警報的下限值
Set Alarm	點選 “Set Alarm” 按鈕，來套用 Alarm Mode、High Alarm Limit 或 Low Alarm Limit 設定
(High/Low) Alarm Status	<p>顯示/清除警報狀態</p> <p>1) <u>CH: xx</u>，超出警報的上限/下限值時，此處會顯示紅色</p> <p>2) <u>Clear</u>，Alarm Mode 設為 Latch 時，可點選此按鈕，來清除警報狀態</p>

3.4.4 設定頁面 - DO

“DO” 頁面，可用來設定/顯示模組的 DO 狀態值、Power-on Value 與 Safe Value。

型號	說明
tM-DA1P1R1	具有 1 個 數位輸出通道
tM-P3R3、tM-PD3R3、tM-P3POR3	具有 3 個 繼電器輸出通道
tM-P4A4、tM-P4C4	具有 4 個 數位輸出通道
tM-R5	具有 5 個 繼電器輸出通道
tM-C8	具有 8 個 數位輸出通道



Bit Status (CH: xx)	設定 DO Value、Power-on Value 與 Safe Value
DO Value	勾選 CH: xx 後，會顯示數值 (16 進制)
Set to Power-on Value	勾選 CH: xx 後，再點選此按鈕，套用設定值 (參考: 附錄 A.2 雙看門狗功能)
Set to Safe Value	
Read DO	點選可讀取目前的 DO Value
Read Power-on Value	點選可讀取目前的 Power-on Value 或 Safe Value (此時會無法設定 Power-on Value 與 Safe Value)
Read Safe Value	

3.4.5 設定頁面 - Host WDT

“Host WDT” 頁面，可用來啟用此功能，並可重置 Host Watchdog Timeout 狀態。您可參考附錄 A.2，了解主機看門狗 (Host Watchdog) 的功能說明。

型號	說明
tM- <u>DA1P1R1</u>	具有 1 個 類比輸出通道 與 1 個 數位輸出
tM-AD4P2 <u>C2</u>	具有 2 個 數位輸出通道
tM-P3 <u>R3</u> 、tM-PD3 <u>R3</u> 、tM-P3 <u>POR3</u>	具有 3 個 繼電器輸出通道
tM-P4 <u>A4</u> 、tM-P4 <u>C4</u>	具有 4 個 數位輸出通道
tM-R5	具有 5 個 繼電器輸出通道
tM-C8	具有 8 個 數位輸出通道

- 1) 當 “Configuration” 頁面的 **Protocol = DCON** 時，畫面如下：
(參考: 3.2.1 節 通用設定項目)

tAD4P2C2 Firmware[AI05]

Configuration AI/DO Alarm DI Host WDT Commands Log Summary About

☐ Enable WDT ☐ Auto Send Host OK

WDT Timeout 25.50 Set Timer
(0.1 ~ 25.5 sec)

Reset Watchdog Status

Exit

Enable WDT	勾選可啟用 Watchdog 功能
Auto Send Host OK	勾選可自動傳送主機通訊正常的訊號 (即，不會 Timeout，測試用)
WDT Timeout / Set Timer	設定 Watchdog Timeout 時間，並點選 “Set Timer” 按鈕套用設定
Reset Watchdog Status	Timeout 後，需清除 Timeout 狀態，才能寫入 AO/DO 值

- 2) 當 “Configuration” 頁面的 **Protocol = Modbus RTU/ASCII** 時，畫面如下：
(參考: 3.2.1 節 通用設定項目)

The screenshot shows the 'Host WDT' configuration window. It has tabs for Configuration, AI/DO Alarm, DI, Host WDT, Commands Log, Summary, and About. The 'Host WDT' tab is active. It contains two checkboxes: 'Enable WDT' and 'Enable Output When WDT Timeout'. The 'Enable Output When WDT Timeout' checkbox is highlighted with a red box. Below these checkboxes is a 'WDT Timeout' field with the value '25.50' and a 'Set Timer' button. Below the field is the text '(0.1 ~ 25.5 sec)'. At the bottom left is a 'Reset Watchdog Status' button. At the bottom right is an 'Exit' button.

Enable WDT	勾選可啟用 Watchdog 功能
Enable Output When WDT Timeout	勾選，即使 Watchdog Timeout 後，仍可寫入 AO/DO 值
WDT Timeout / Set Timer	設定 Watchdog Timeout 時間，並點選 “Set Timer” 按鈕套用設定
Reset Watchdog Status	清除 Timeout 狀態

例如:

1. 勾選 Enable WDT 並設定 “WDT Timeout” 時間為 5 秒，再點選 “Set Timer” 按鈕。5 秒後，狀態列會顯示 Watchdog Timeout 的訊息，此時，無法寫入輸出值，會設為 Safe Value。

註: 若模組為 Modbus Protocol，可勾選 “Enable Output When WDT Timeout”，表示 Timeout 後仍可寫入 AO/DO 值。

2. 點選 “Reset Watchdog Status” 按鈕，可清除 Timeout 狀態，並會在重開機後，載入 Power-on Value；若未清除此狀態，則重開機後，輸出值會是 Safe Value。

註: 若模組為 DCON Protocol，可勾選 “Auto Send Host OK”，表示主機會一直傳送通訊正常的訊息給模組，取消此勾選，再點 “Set Timer” 按鈕，即可再正常使用 Watchdog 功能。

3.4.6 設定頁面 - DI

“DI” 頁面，可用來顯示/清除 DI、DI Latch 與 DI Counter 的狀態。

型號	說明
tM-DA1 <u>P</u> 1R1	具有 1 個 數位輸入通道
tM-AD4 <u>P</u> 2C2	具有 2 個 數位輸入通道
tM- <u>P</u> 3R3、tM- <u>P</u> D3R3、tM- <u>P</u> 3POR3	具有 3 個 數位輸入通道
tM- <u>P</u> 4A4、tM- <u>P</u> 4C4	具有 4 個 數位輸入通道
tM-P8、tM-PDW8	具有 8 個 數位輸入通道

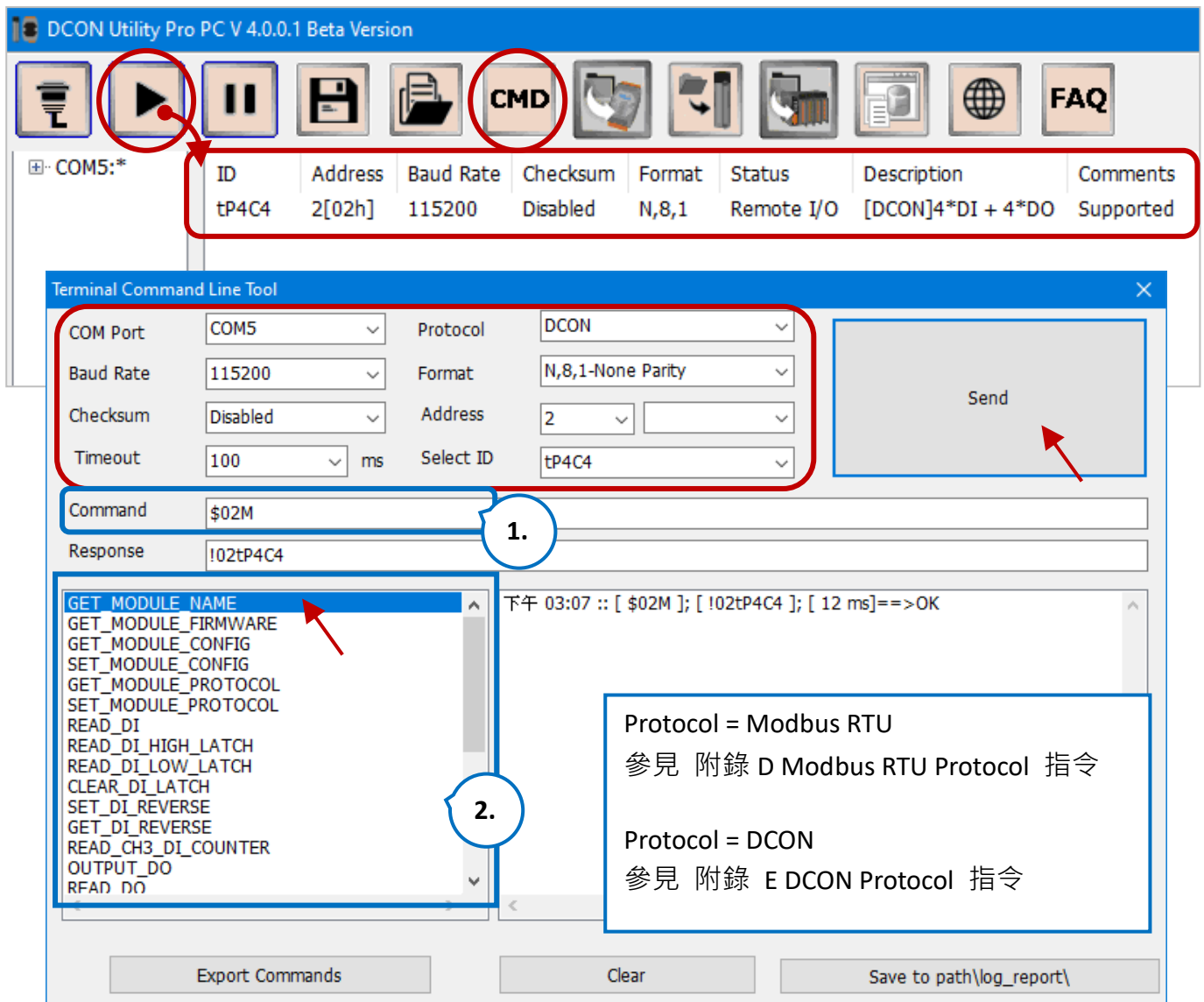
DI 頁面

Reverse DI Status		將 DI 狀態設為反相
DI Bit Status		顯示各 DI 通道的狀態 (ON/OFF)
DI Value		顯示 DI 狀態值 (16 進制)
DI Latch	High Latch	顯示各 DI 通道的高門鎖 (Latch High) 狀態
	Low Latch	顯示各 DI 通道的低門鎖 (Latch Low) 狀態
	Clear DI Latch	清除 DI 通道的門鎖狀態 (參考: 附錄 A.6)
DI Counter	Falling/Rising Edge	DI 由 High 到 Low 或 由 Low 到 High，計數值會加 1
	Clear	清除各 DI 通道的計數值

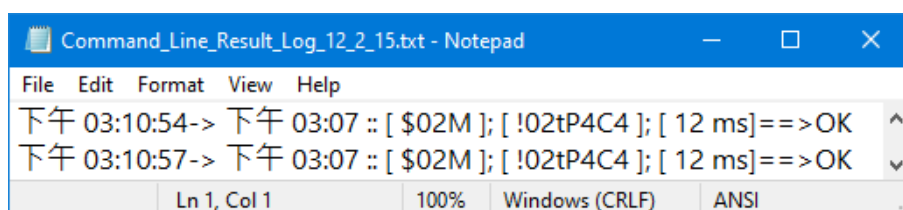
3.5 命令列工具 (Command Line Tool)

“Command Line” 功能可用來作為 tM 模組測試與除錯的工具，目前可支援 Modbus RTU Protocol 與 DCON Protocol 指令。

1. 請先搜尋 tM 模組，再點選 “Command Line” 按鈕，來開啟設定畫面。
2. 在 “Terminal Command Line Tool” 視窗，會自動填入模組的設定。您可直接在 **Command** 欄位，輸入指令 (1) 或使用預設的指令 (2)，再點選 “Send” 按鈕。



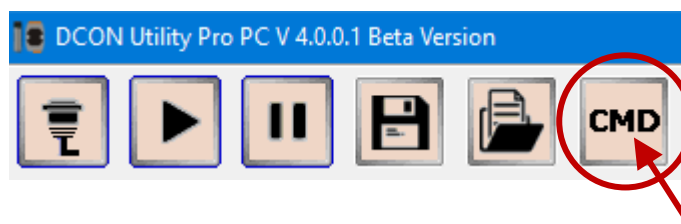
3. 點選 “Save to \logger_report\” 按鈕，可將測試結果儲存在 DCON Utility Pro 安裝路徑下的 “log_report” 資料夾，且檔名為 Command_Line_Result_Log_mm_dd_xx.txt。



3.6 校正功能 (Calibration)

一般模組在出廠之前已經完成校正，若有出現需校正的情況，您可使用 **DCON Utility Pro** 的“命令列工具 (Command Line)”輸入以下指令。

注意：請務必在完全了解設定程序後，才開始進行校正功能。



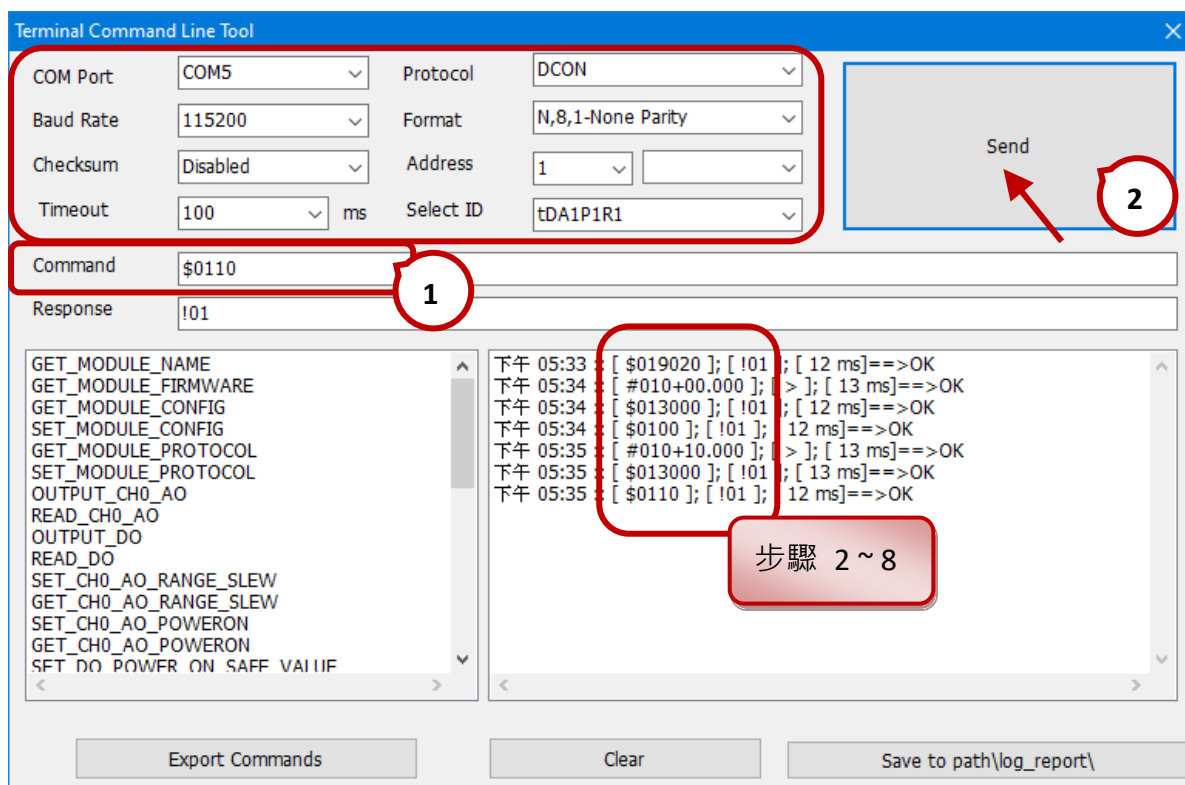
3.6.1 校正 AO 模組

tM-DA1P1R1 具有一個 AO 通道，請依照下列步驟使用 DCON Utility Pro 來執行校正功能：

1. 請將模組暖開機至少 30 分鐘。
2. 設定 ([\\$AA9NTS](#)) (或 讀取 [\\$AA2](#)) Type Code。

例如: **\$019020**，表示將 AO 模組 (位址=**01**) 通道 **0** 的 Type Code 設為 **2** (0 ~ +10V) 且 Slew Rate 為 **0** (Immediate)。

註：校正時，模組的 Protocol 需為 DCON。



tM-DA1P1R1 模組的校正電壓/電流：

Type Code (T)	0	1	2	4
Zero 輸出	0mA	4mA	0V	0V
Span 輸出	+20mA	+20mA	+10V	+5V

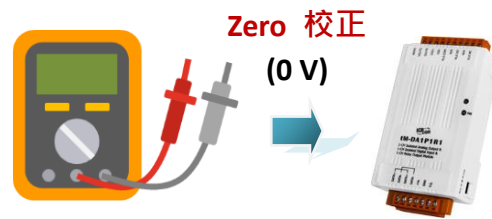
3. 輸出 **Zero** 校正電壓/電流。(此例為 **0 V**)

指令格式: **#AAN(Data) = # [AA] 模組位址 [N] 通道編號 [Data] +00.000 (V, mA)**

指令: **#010+00.000**

4. 調整校正值

量測並調整 (加/減) 電壓/電流，直到從 AO 通道所量測的電壓/電流，接近 Zero 校正電壓/電流。



指令格式: **\$AA3NVV = \$ [AA] 模組位址 [3] 設定 AO [N] 通道編號**

[VV] 00 ~ 5F: 增加 0 ~ 95 ; FF ~ A1: 減少 1 ~ 95

指令: **\$01301F** (1F 表示增加 31 * 2.44 mV 或 增加 31 * 4.88 uA)

5. 調整好後，輸入 **Zero** 校正指令。(此值會存入 EEPROM)

指令格式: **\$AA0N = \$ [AA] 模組位址 [0] Zero 校正 [N] 通道編號**

指令: **\$0100**

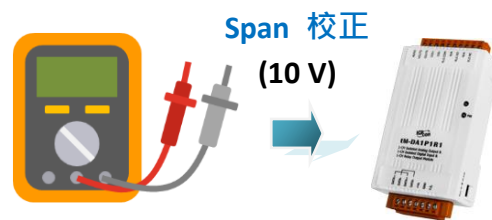
6. 輸出 **Span** 校正電壓/電流。(此例為 **10 V**)

指令格式: **#AAN(Data) = # [AA] 模組位址 [N] 通道編號 [Data] +10.000 (V, mA)**

指令: **#010+10.000**

7. 調整校正值

量測並調整 (加/減) 電壓/電流，直到從 AO 通道所量測的電壓/電流，接近 Span 校正電壓/電流。



指令格式: **\$AA3NVV = \$ [AA] 模組位址 [3] 設定 AO [N] 通道編號**

[VV] 00 ~ 5F: 增加 0 ~ 95 ; FF ~ A1: 減少 1 ~ 95

指令: **\$01301F** (1F 表示增加 31 * 2.44 mV 或 增加 31 * 4.88 uA)

8. 調整好後，輸入 **Span** 校正指令。(此值會存入 EEPROM)

指令格式: **\$AA1N = \$ [AA] 模組位址 [1] Span 校正 [N] 通道編號**

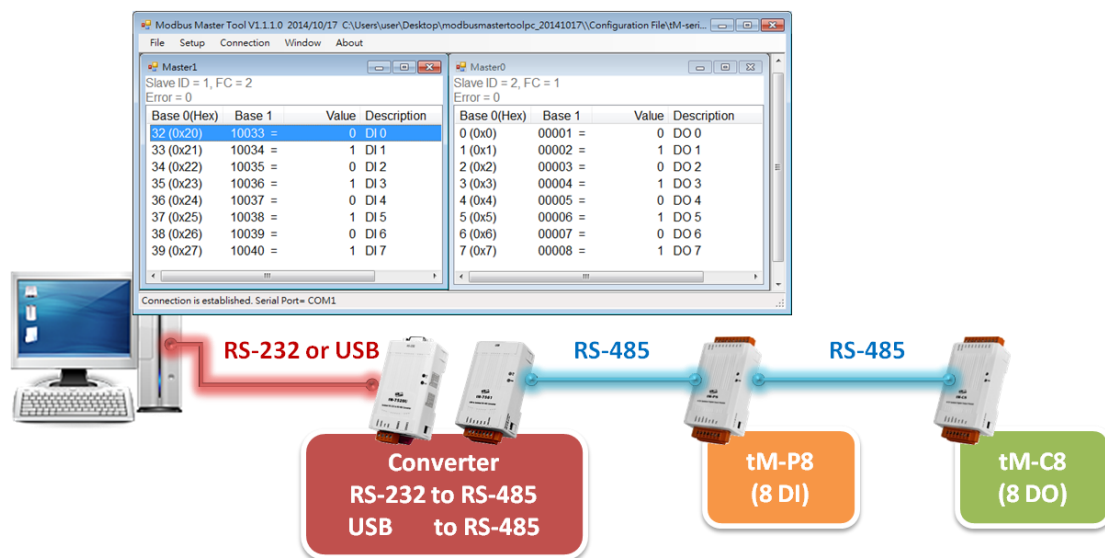
指令: **\$0110**

第 4 章 軟體工具 – Modbus Master Tool

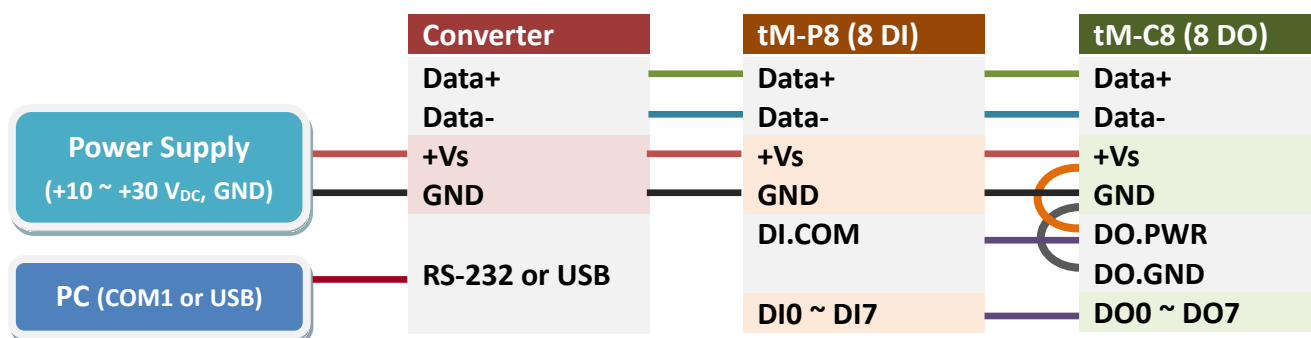
Modbus Master Tool 可支援 Modbus RTU/ASCII 通訊協定，可用來在 PC/PAC 上模擬 與 測試 I/O 模組。首先，請下載軟體與手冊：

網頁: https://www.icpdas.com/en/product/guide+Software+Development__Tools+Modbus__Tool#674

軟體: <https://www.icpdas.com/en/download/show.php?num=1026> (PC 或 WinCE)



硬體接線 (測試用)：



此範例中會使用一個 tM 系列的 DI 與 DO 模組，請依照下列步驟，來進行設定。

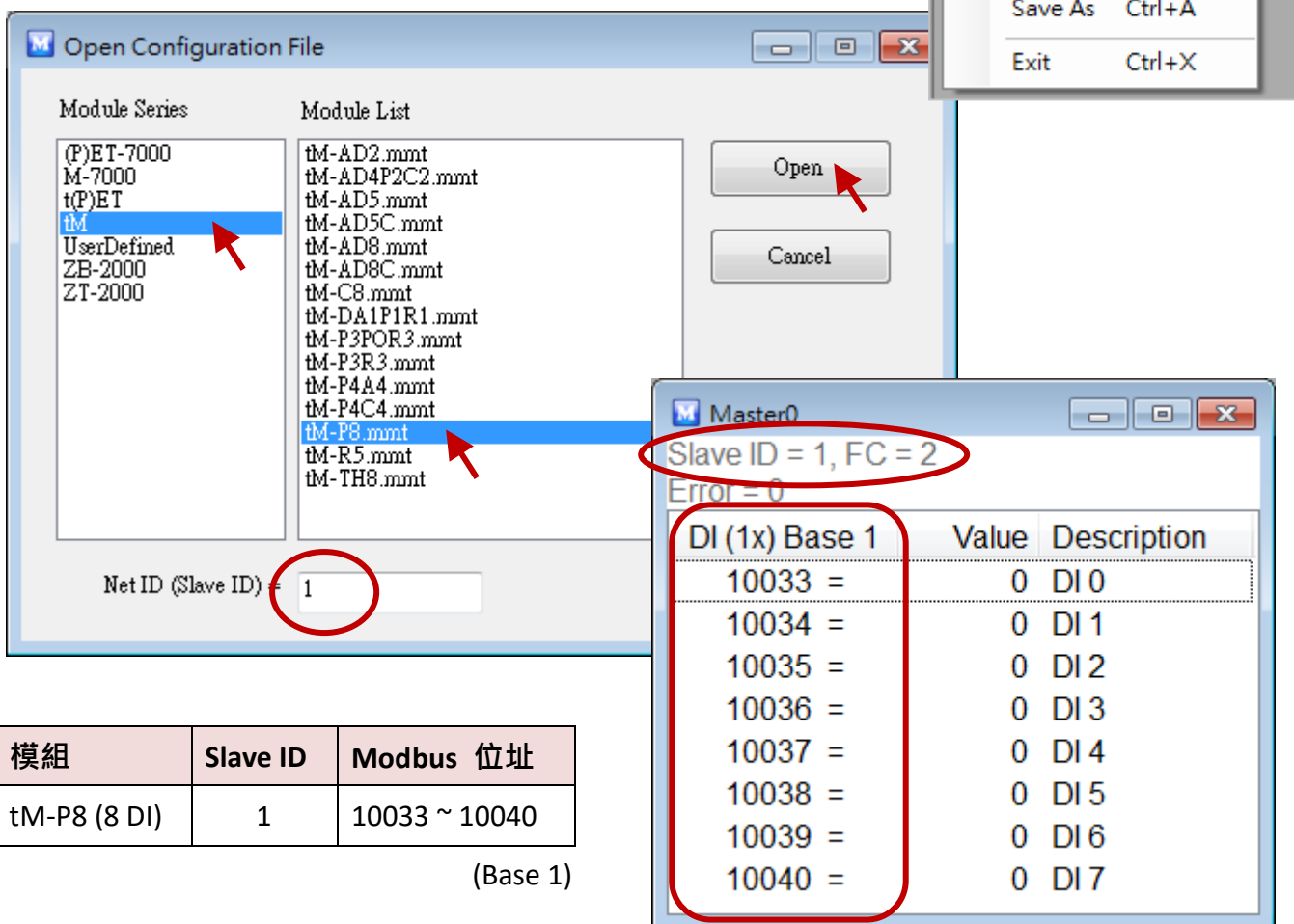
模組名稱	說明	Slave ID	Baud Rate	Protocol	Modbus 位址
tM-P8	8 DI	1	9600	Modbus RTU	10033 ~ 10040
tM-C8	8 DO	2			00001 ~ 00008

註： 參考附錄 C，查詢 tM 系列模組使用的 Modbus 位址。

步驟 1: 開啟組態檔

解壓縮下載檔案後，開啟 Modbus Master Tool 軟體。

在“File”選單中點選“Open”，接著，選取模組系列 (例如: tM)、名稱 (例如: tM-P8) 並填入 Slave ID (例如: 1)，再點選“Open”按鈕，來開啟組態檔。



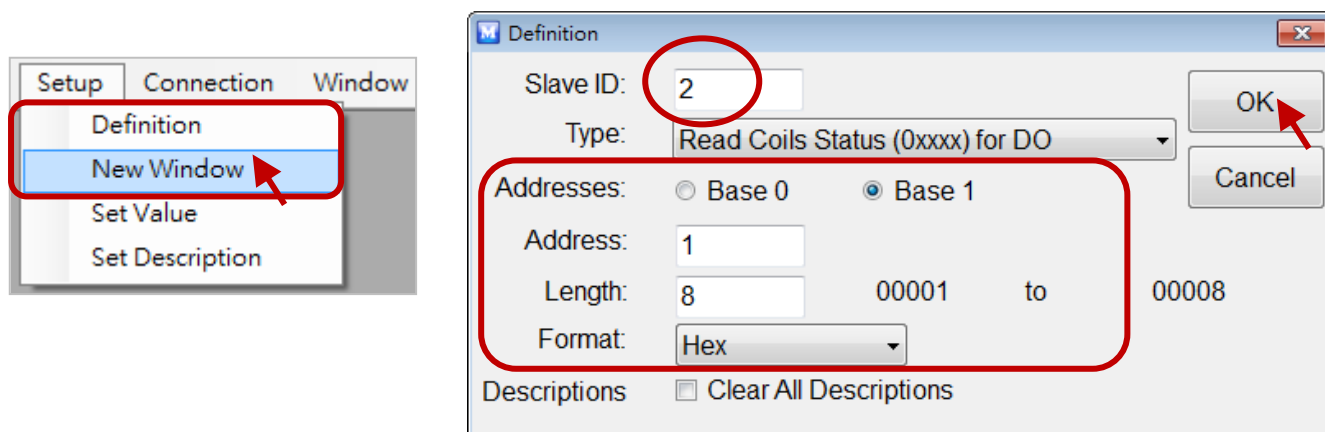
模組	Slave ID	Modbus 位址
tM-P8 (8 DI)	1	10033 ~ 10040

(Base 1)

DI (1x) Base 1	Value	Description
10033 =	0	DI 0
10034 =	0	DI 1
10035 =	0	DI 2
10036 =	0	DI 3
10037 =	0	DI 4
10038 =	0	DI 5
10039 =	0	DI 6
10040 =	0	DI 7

步驟 2: 新增視窗 與 設定 DO 配置

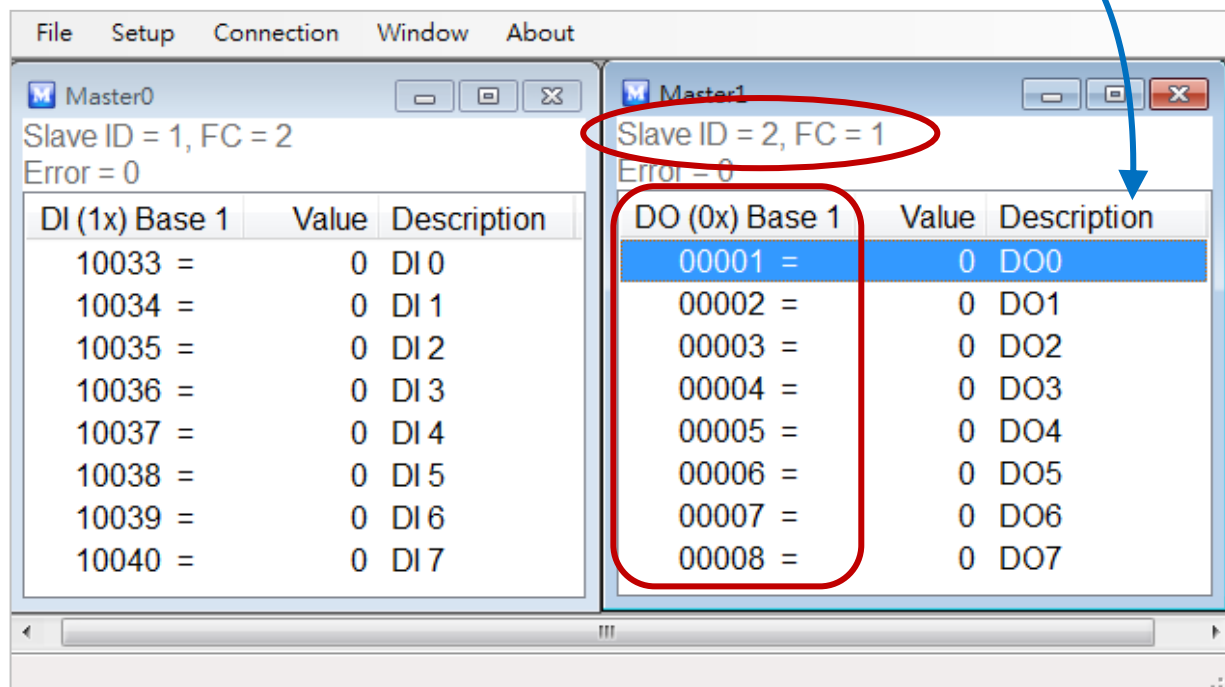
在“Setup”選單中點選“New Window”，再點選“Definition”來設定 DO 配置，再點選 OK。



模組名稱	Slave ID	Modbus 位址
tM-C8 (8 DO)	2	00001 ~ 00008

(Base 1)

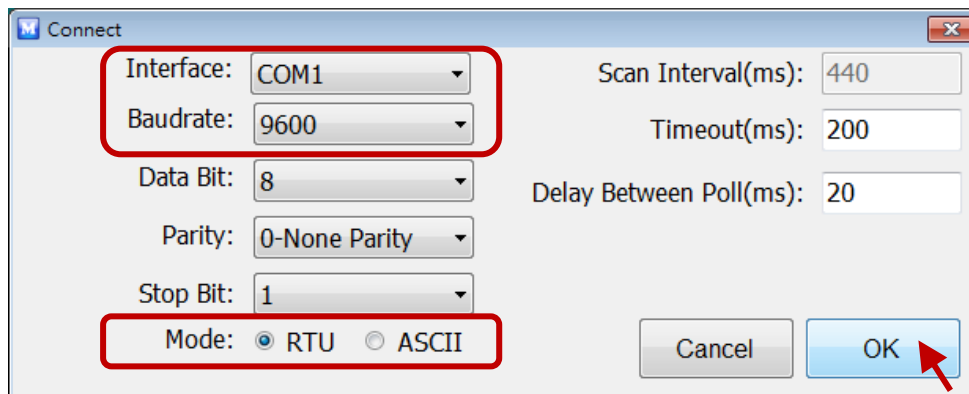
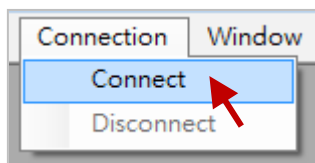
滑鼠雙擊 “Description” 欄位，
可輸入方便識別的說明。



步驟 3: 建立 Serial Port 連線

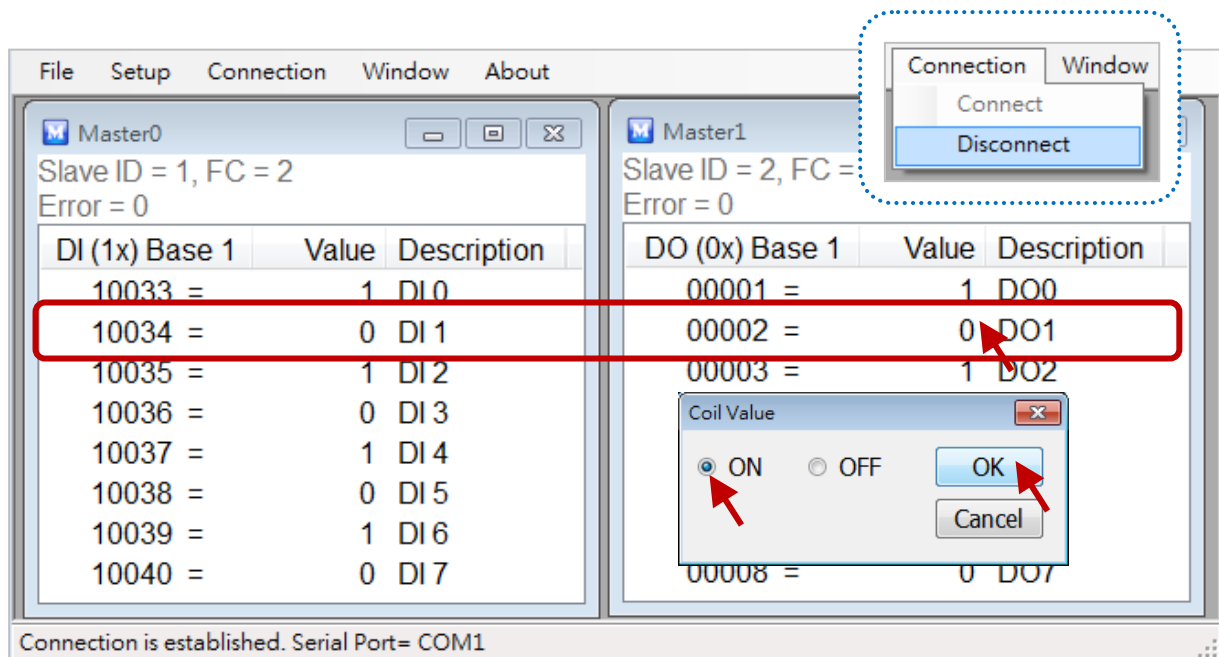
在 “Connection” 選單中點選 “Connect”，此例，tM 模組採用 Modbus RTU 通訊協定 與 PC 的 COM1 進行通訊，請設定參數如下，並點選 “OK” 連線。

註: 請先使用 DCON Utility Pro (參考第 3 章) 設定好模組的通訊參數。



步驟 4: 測試 DI/DO 模組

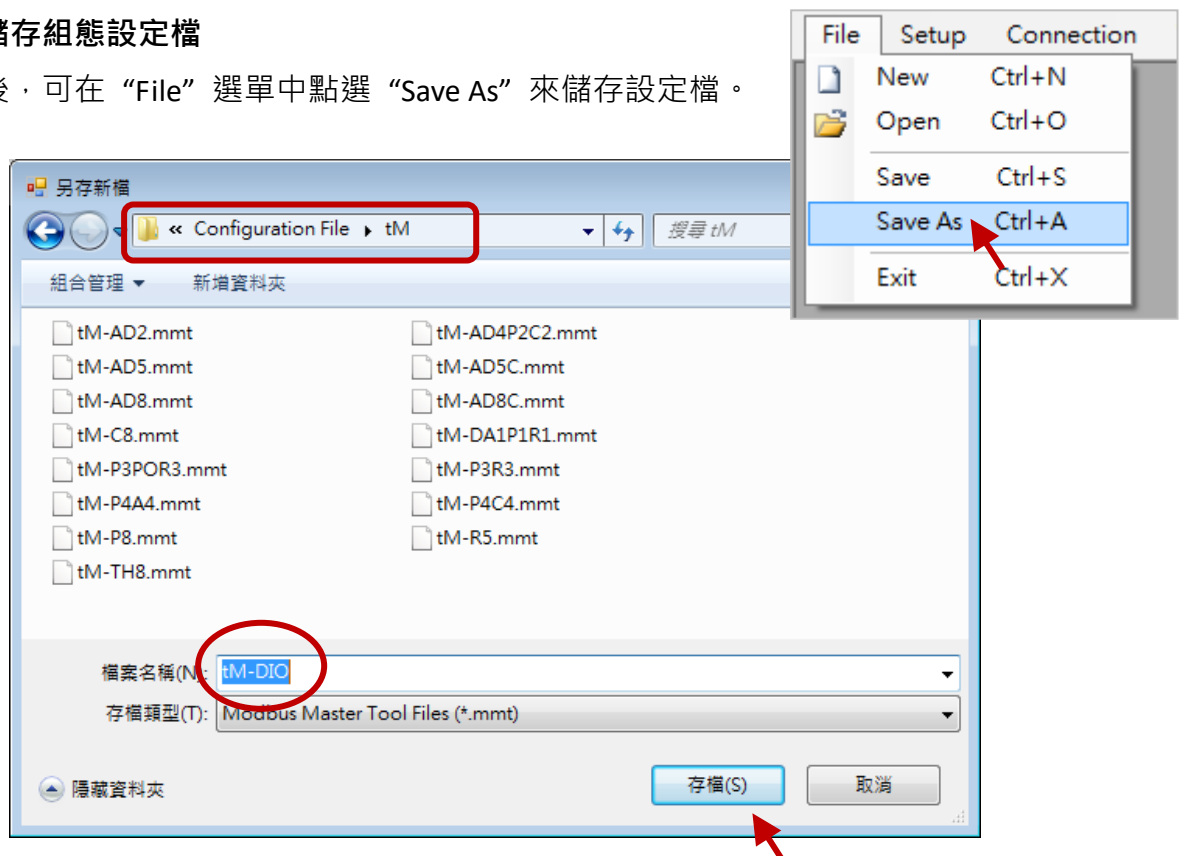
連線後，可見到 DI/DO 模組目前的狀態值。滑鼠雙擊任一 DO 通道，可變更狀態值，並查看 DI 狀態值的變化。**註：**在“Connection”選單中，點選“Disconnect”可取消連線。



DI (1x) Base 1	Value	Description	DO (0x) Base 1	Value	Description
10033 =	1	DI 0	00001 =	1	DO 0
10034 =	1	DI 1	00002 =	1	DO 1

步驟 5: 儲存組態設定檔

測試完成後，可在“File”選單中點選“Save As”來儲存設定檔。



第 5 章 軟體工具 – EZ Data Logger / eLogger HMI

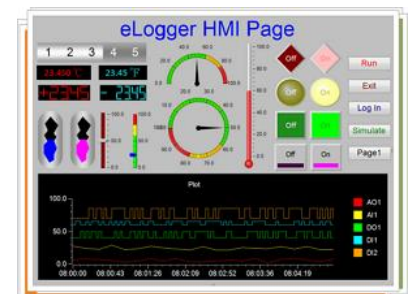
註: EZ Data Logger 於 2019, July 不再支援新軟體功能，建議選用 eLogger HMI。

支援項目	EZ Data Logger	eLogger
硬體平台	PC	PC, XPAC, WinPAC, ViewPAC, iPPC Win-GRAF 系列 PAC
作業系統	Windows XP, 7, 10	Windows XP, 7, 10, WinCE 6, 7
HMI	有	有
Web HMI	無	有
Data Logger	Access, MS SQL, MySQL	CSV, MS SQL, MySQL
通訊協定	Modbus RTU, Modbus TCP 與 DCON	Modbus RTU, Modbus TCP 與 插槽上的 I-8K/I-87K 通訊
第三方驅動擴充性	無	提供開發工具以增加第三方 Plug-in 驅動

eLogger 是泓格科技開發的一套免費且簡單易用的 HMI (Human Machine Interface) 軟體，不僅可用來設計 Local HMI 還可設計 Web Server HMI，提供 PC、手機透過網頁瀏覽的方式對 PAC 進行遠端操作。搭配 Win-GRAF 開發軟體，使用者無需再撰寫任何程式語言，即可製作出專業的監控應用。網頁: <https://www.icpdas.com/en/product/guide+Software+eLogger+eLogger>

eLogger 特色:

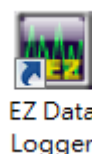
- 支援 Modbus RTU/ASCII 與 Modbus TCP 通訊協定。
- 支援即時趨勢圖:
一個趨勢圖可同時紀錄五筆資料。
- 支援 Local HMI:
多樣式元件，可切換頁面，最多可達 32 個頁面。
- 支援 Web Server HMI:
提供管理者登入功能。
- 支援權限管理 (適用 Local HMI):
區分為三種層級的操作管理：管理者、進階使用者、使用者。
- 支援遠端控制:
執行 eLogger 開發版的“遠端操作”功能，可透過網路來上傳/開始/停止專案。
- 支援資料庫
本地端資料庫: 支援 .CSV 檔案格式。
遠端資料庫: 支援 Microsoft SQL Server 2005 以上版本。



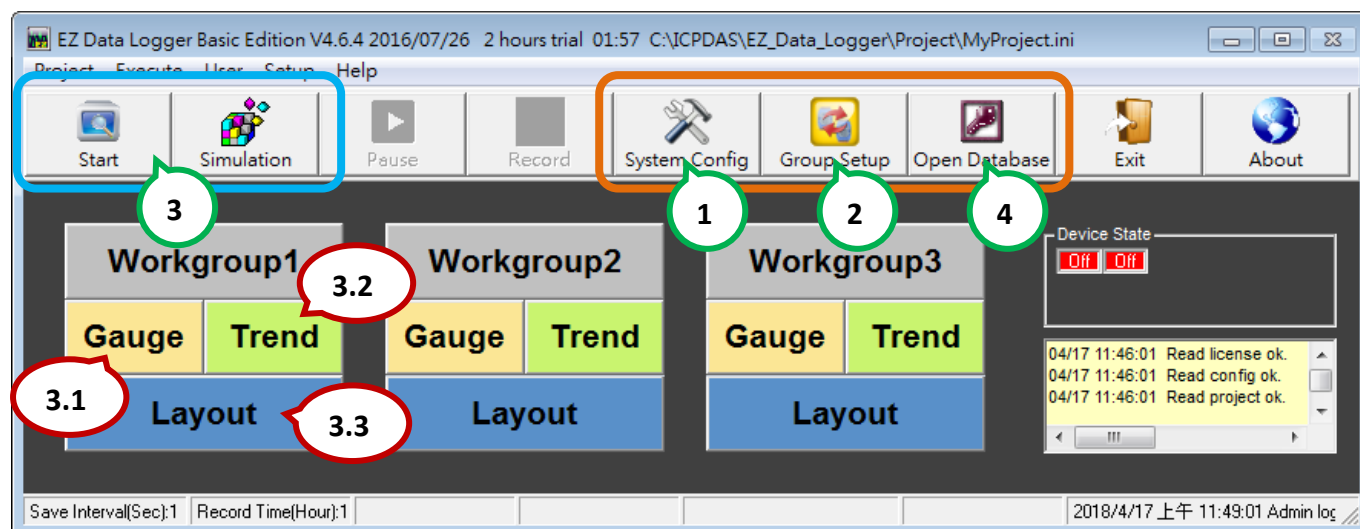
EZ Data Logger 是一個簡易型的數據採集軟體，可支援 DCON 與 Modbus RTU/ASCII 通訊協定，用來讓使用者快速、簡單的架設一個數據採集系統，省去撰寫程式的麻煩。

您可在網站下載軟體 與 使用手冊 (.chm)。

https://www.icpdas.com/en/product/guide+Software+EZ_Data_Logger+EZ_Data_Logger



安裝並開啟 EZ Data Logger 軟體。此軟體可分為 4 大操作步驟，後續將一一說明。



步驟 1: [系統設定 \(System Config\)](#)

步驟 2: [群組設定 \(Group Setup\)](#)

步驟 3: [開始執行 \(Start\) 或 進行模擬 \(Simulation\)](#)

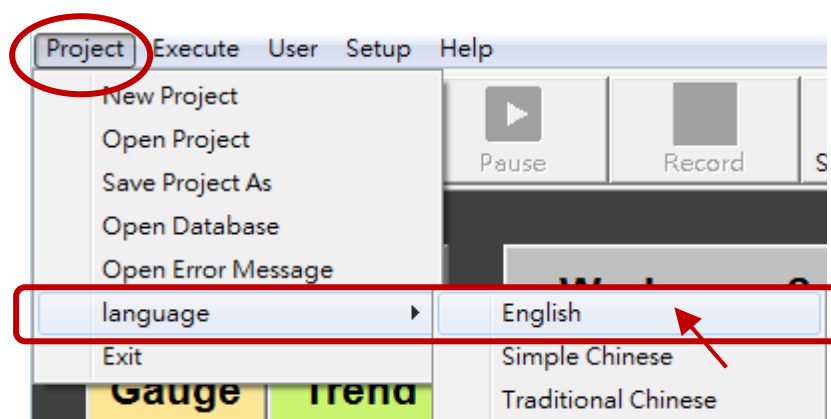
3.1 [量測計 \(Gauge\)](#)

3.2 [趨勢圖 \(Trend\)](#)

3.3 [平面配置圖 \(Layout\)](#)

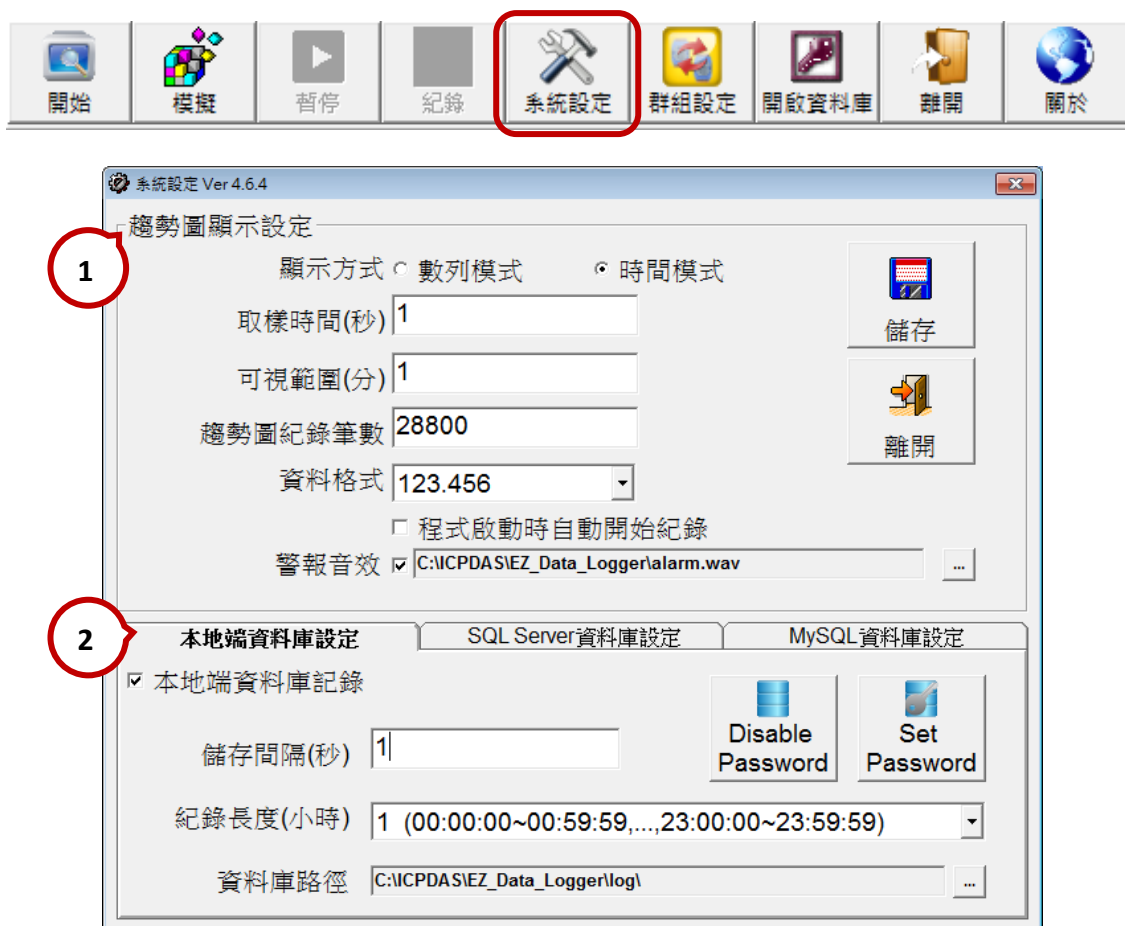
步驟 4: [開啟資料庫 \(Open Database\)](#)

首先，您可在“Project”選單中變更所需的介面語言。



5.1 系統設定 (System Config)

在 "系統設定" 視窗，可設定趨勢圖 與 資料庫。

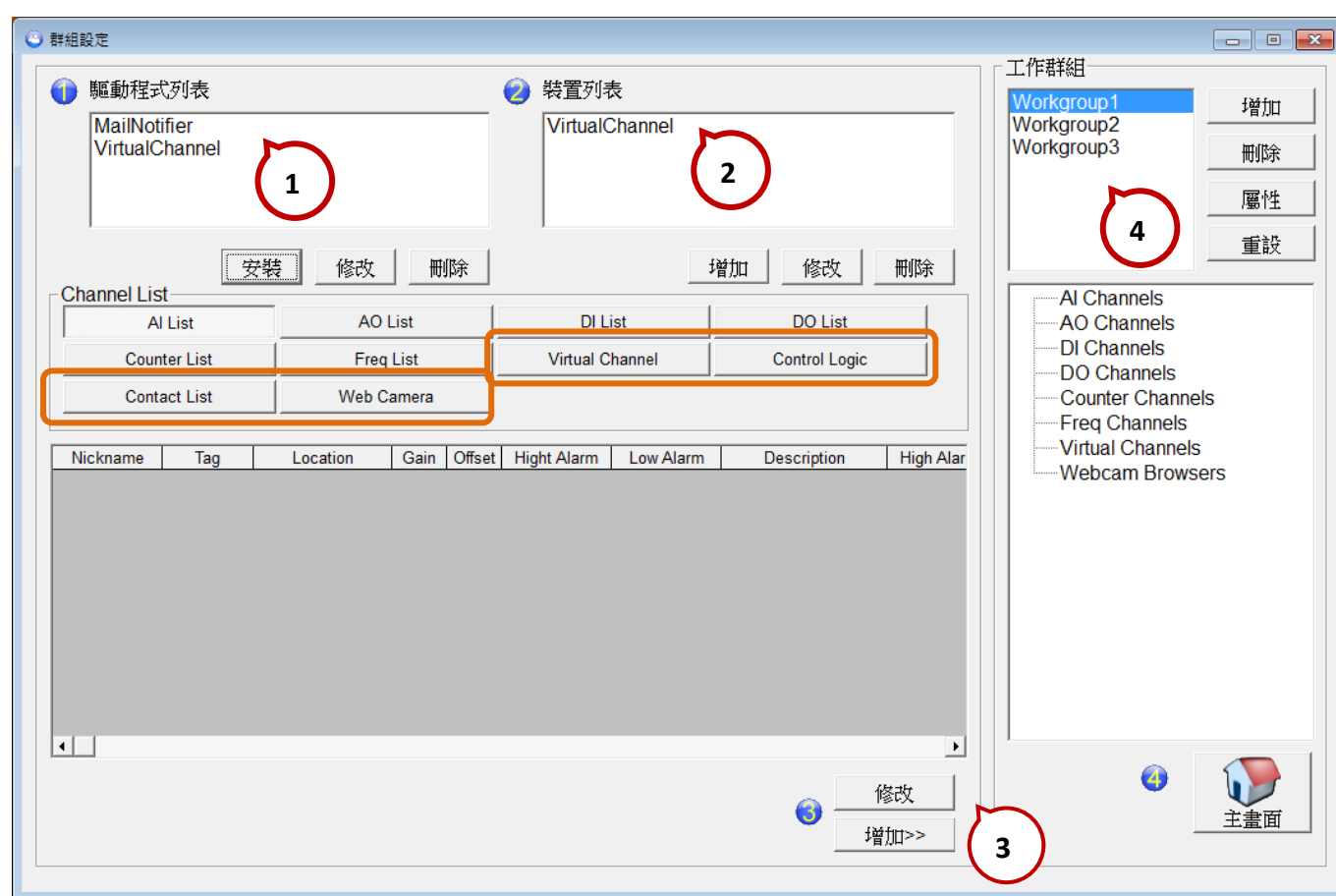
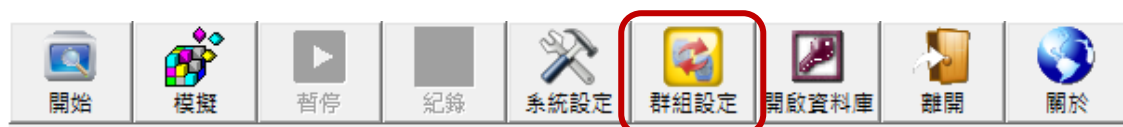


趨勢圖顯示設定	
顯示方式	數列模式: X 軸會顯示記錄筆數，最小取樣時間 0.01 秒。 時間模式: X 軸會顯示記錄時間，最小取樣時間 1 秒。
取樣時間 (秒)	掃描數值、更新資料庫、顯示數值的時間間隔。
可視範圍 (分)	趨勢圖 X 軸的可視範圍。
趨勢圖記錄筆數	每一條曲線所能紀錄的點數。
資料格式	數值顯示的資料格式。
程式啟動時自動開始紀錄	勾選可在程式執行時，自動開始紀錄資料。
本地端資料庫設定	
儲存間隔 (秒)	儲存到資料庫的時間間隔，必須是 "取樣時間" 的倍數。
紀錄長度 (小時)	若超過設定時間，EZ Data Logger 會再建立新的資料庫開始紀錄。
資料庫路徑	資料庫的檔名格式為 Datayyyymmdd_hh.mdb，會自動儲存在設定路徑內。(yyyy:西元年, mm:月, dd:日, hh:小時)

5.2 群組設定 (Group Setup)

在 "群組設定" 視窗，可分成 4 個設定步驟。

- 步驟 1: [安裝驅動程式](#)
- 步驟 2: [安裝裝置](#)
- 步驟 3: [修改通道設定](#) 與 [加入工作群組](#)
- 步驟 4: [設定工作群組屬性](#)



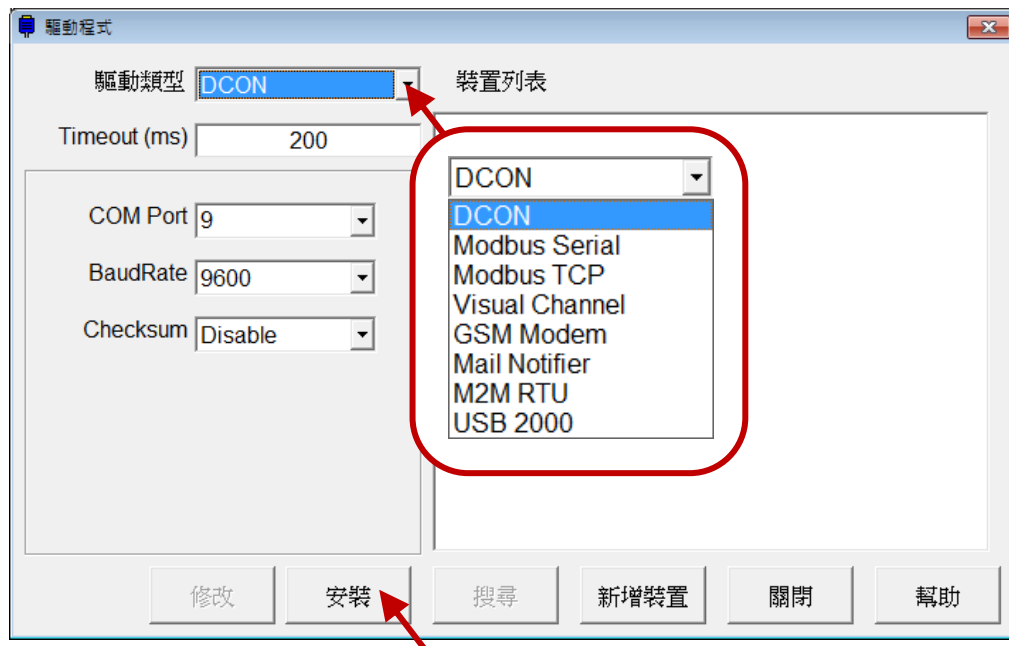
註: 關於 虛擬通道 (Virtual Channel)、控制邏輯 (Control Logic)、網路攝影機 (Web Camera)、警報通知 (Mail Notifier) 與 權限管理功能，請參考 [EZ Data Logger 使用手冊](#) (.chm)。

5.2.1 安裝驅動程式

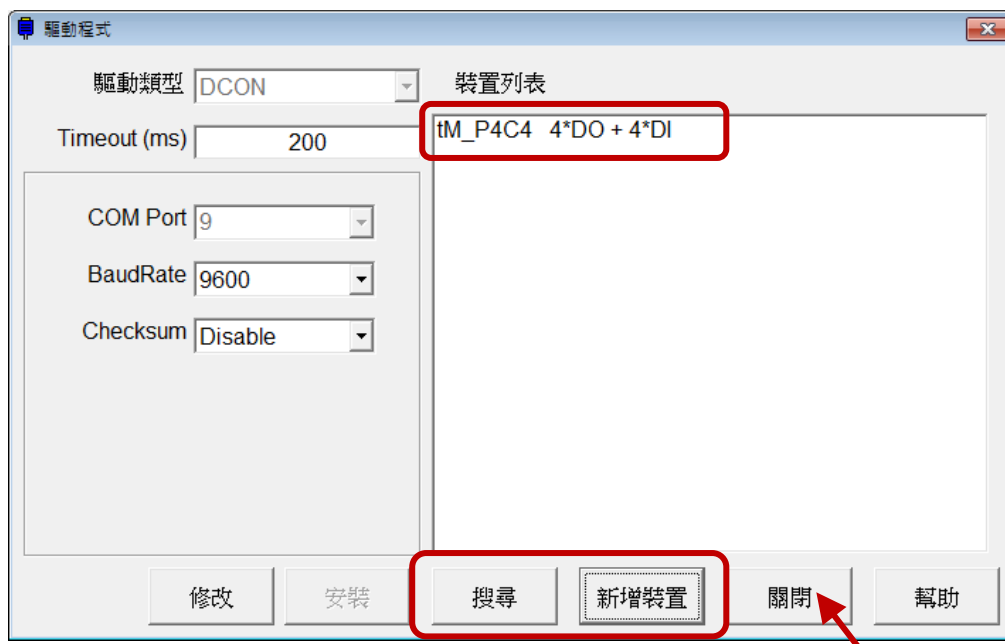
1. 在 驅動程式列表，點選“安裝”按鈕。



2. 請依據應用需求，來設定驅動類型、Timeout (ms)、COM Port、Baud Rate 與 Checksum 設定，再點選“安裝”按鈕。

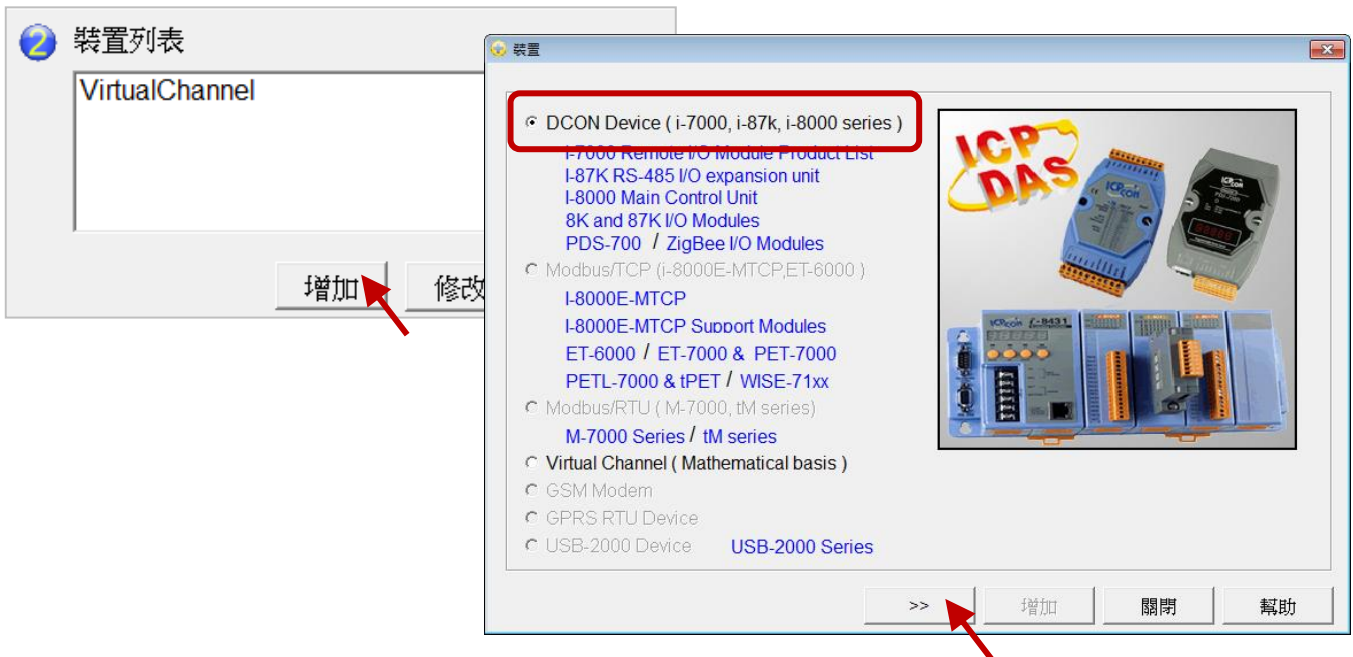


您可點選“新增裝置”按鈕，來加入需使用的設備 (參考: 5.2.2 節 – 安裝裝置)。或是，點選“搜尋”按鈕，尋找已連接的設備。點選“關閉”按鈕，可關閉此視窗。

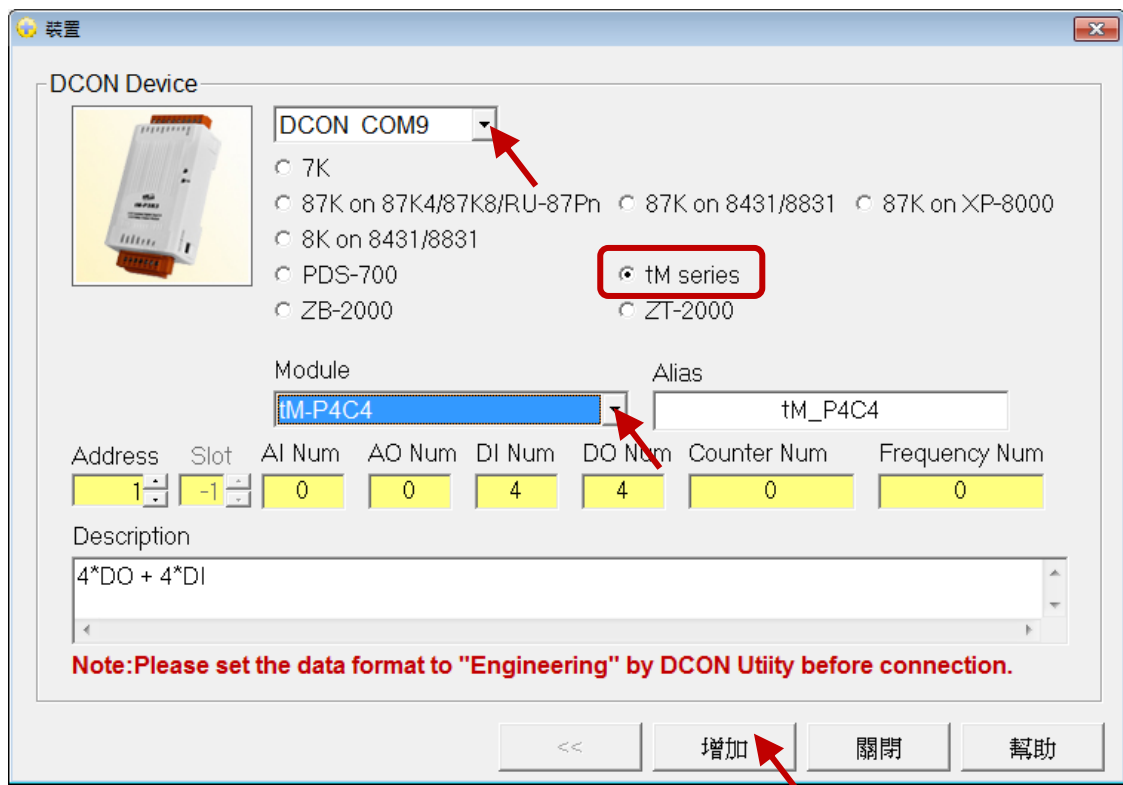


5.2.2 安裝裝置

1. 在 裝置列表 點選“增加”按鈕，再點選“裝置”畫面中的“>>”按鈕。



2. 選擇通訊設定 (例如: DCON COM9)，再選擇 tM 系列模組的型號，您可修改別名 (Alias)。最後，請點選“增加”按鈕。

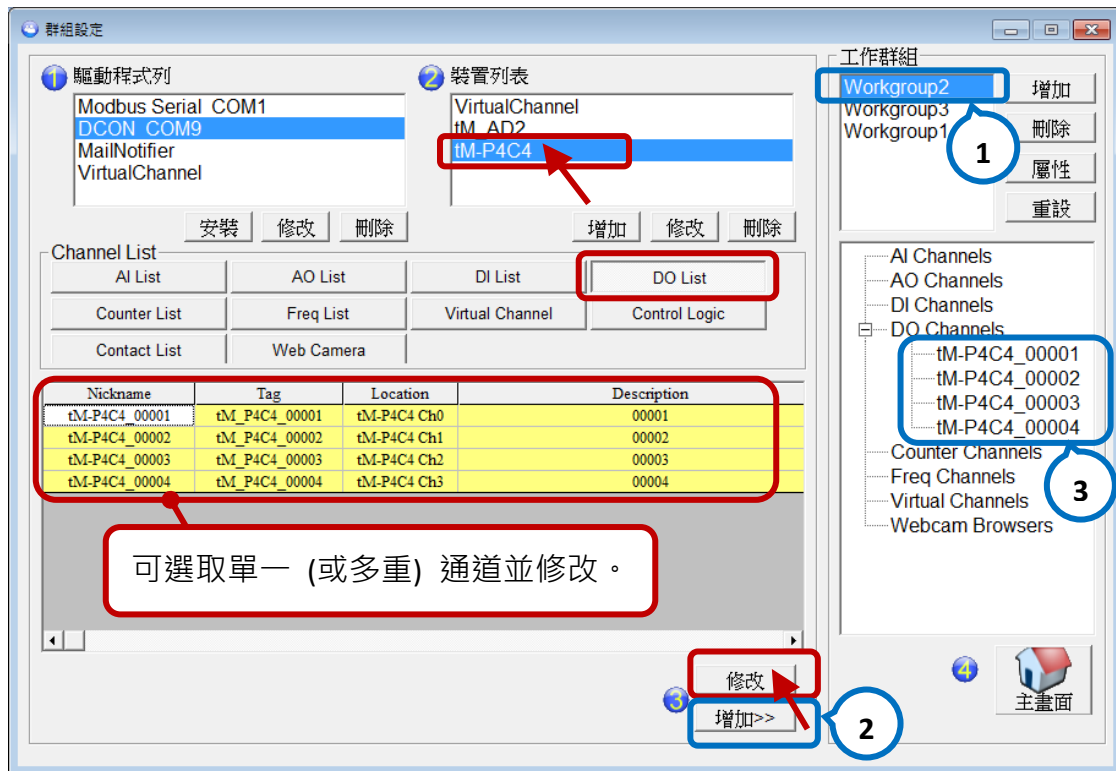


5.2.3 修改通道設定 與 加入工作群組

在 **裝置列表** 點選 **tM** 模組名稱 (例如: **tM_P4C4**)，會選取所有的 I/O T。如果想將其加入到工作群組，可先指定工作群組編號 (例如: **Workgroup2**)，再點選“增加 >”按鈕。

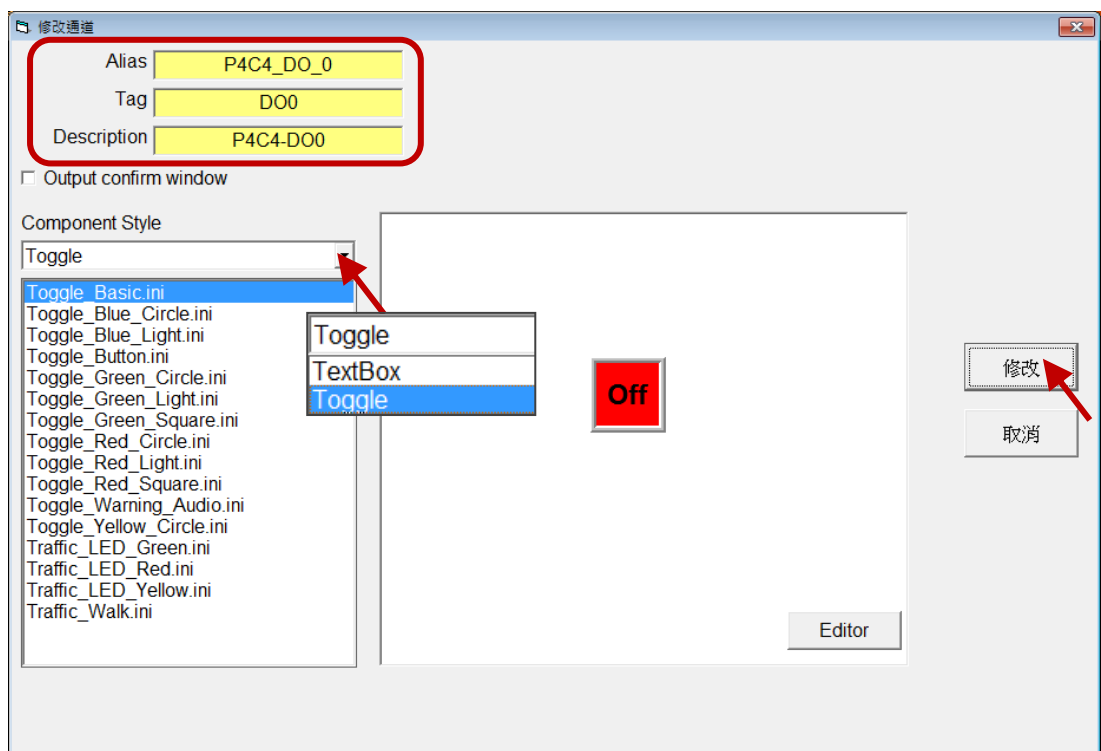
修改通道設定

1. 選取 tM 模組的 I/O 通道 (單一/多重)，並點選“修改”按鈕。

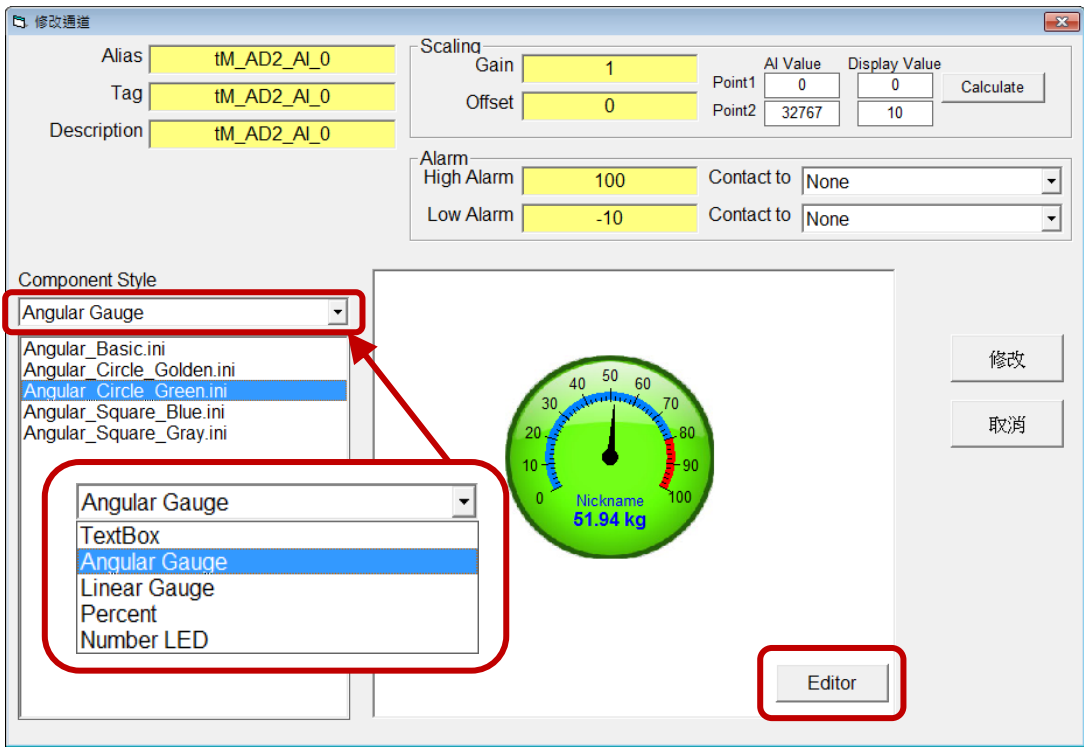


2. 您可修改通道名稱、選取物件樣式，再點選“修改”按鈕。

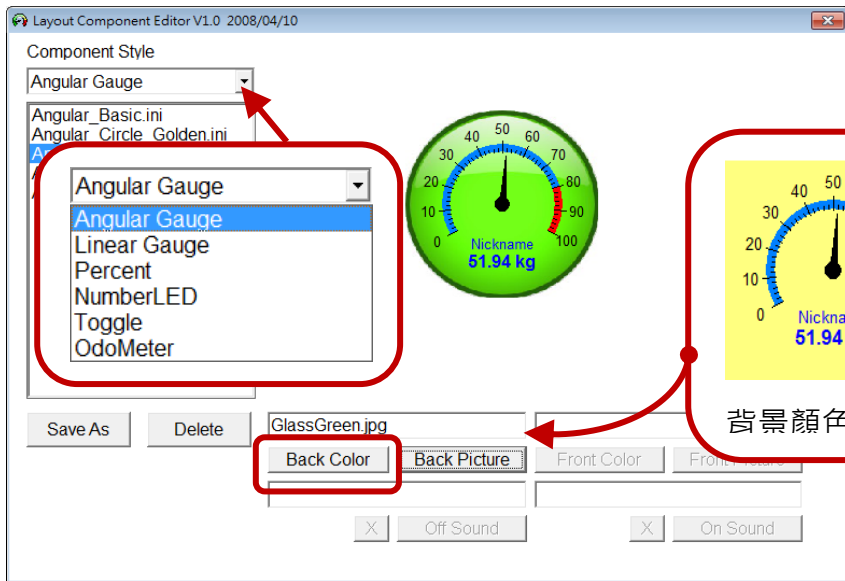
DO 通道:



AI 通道:



Alias	別名，可指定方便識別的名稱。
Tag	標籤名稱，可用在虛擬通道 (Virtual Channel) 或 邏輯控制 (Control logic) 功能上。名稱開頭需為 a ~ z 的英文字母，且第一個字母不可以使用數字。
Description	說明，可用來顯示物件的提示名稱。
Scaling	數值轉換功能，填入 AI 值 (AI Value) 與 顯示值 (Display Value) 的範圍，再點選 “Calculate” 按鈕，會自動算出 Gain/Offset 值。
Alarm	警報功能，若超過高/低警戒值 (High/Low Alarm) 時，警報燈會亮 且 會播放警報聲，直到關閉為止。(參考: 5.3.1 "量測計" 畫面)
Editor	點選此按鈕，可編輯物件樣式。

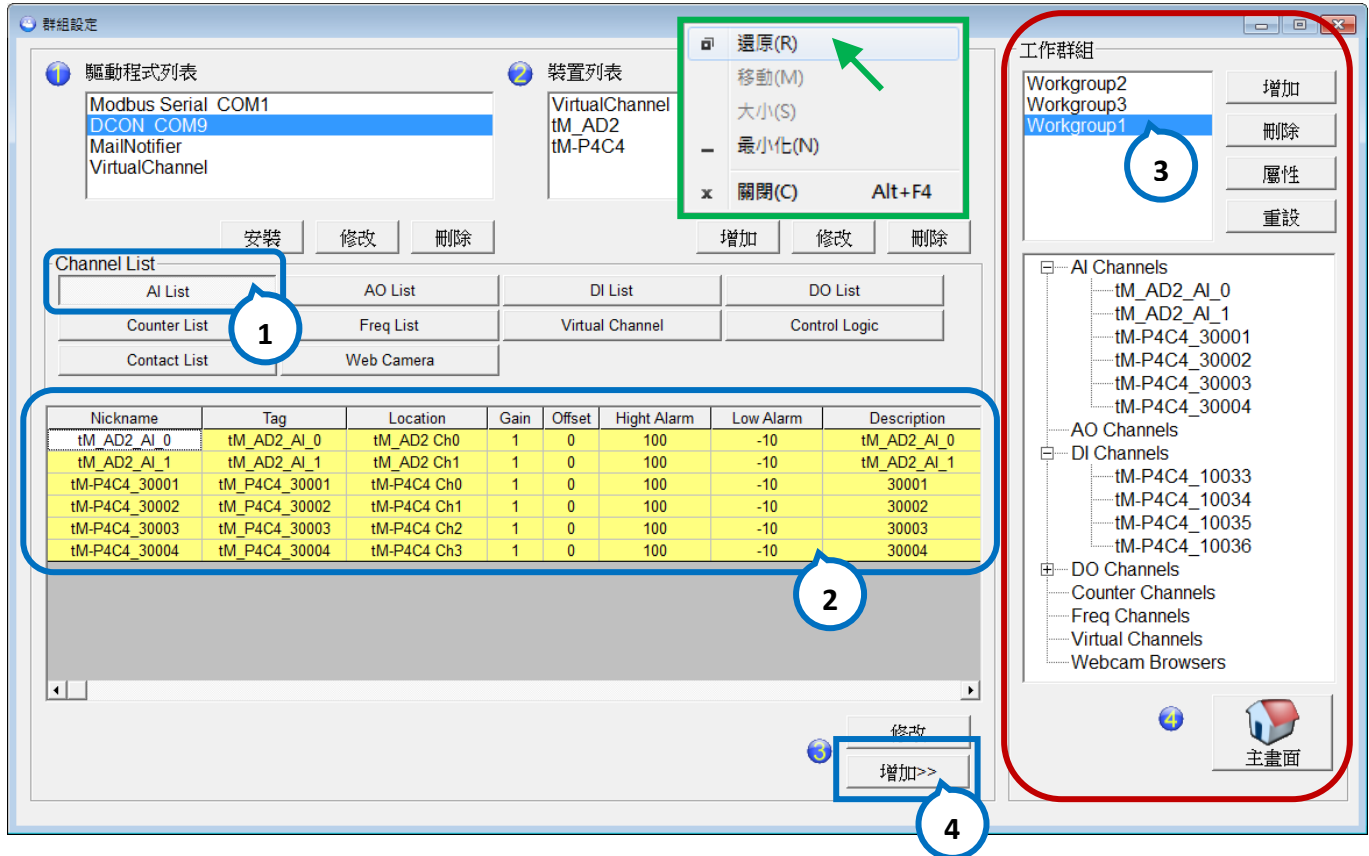


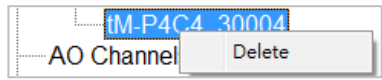
若設定背景顏色，表示不使用圖片。

5.2.4 設定工作群組屬性

如同 5.2.3 節 說明，您可以選取多個 I/O 通道設定，並將其加入到指定的工作群組中。每一個工作群組可加入 32 個通道。

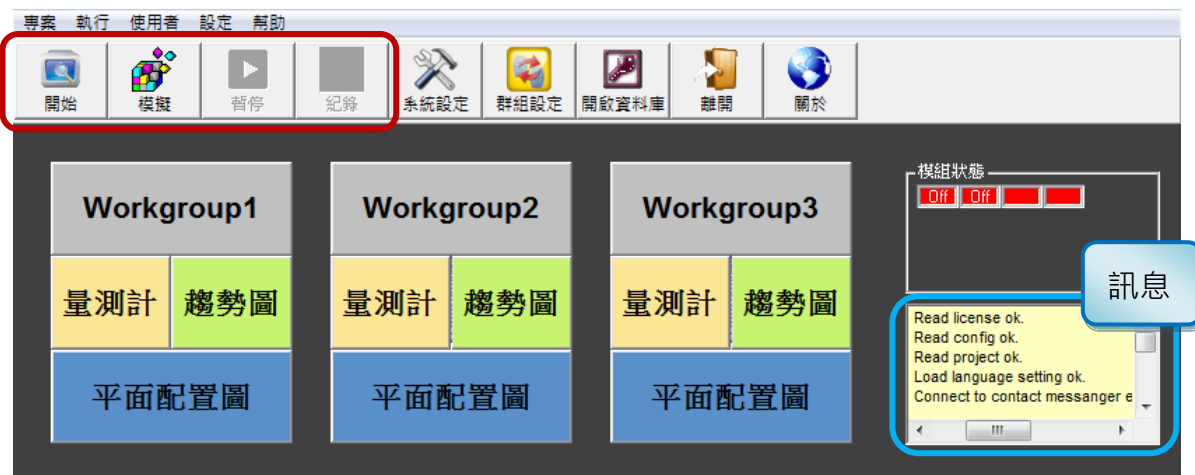
註：若您的 PC 無法縮小視窗，可在視窗上方點選滑鼠右鍵，再點選 "還原" 來縮小視窗。



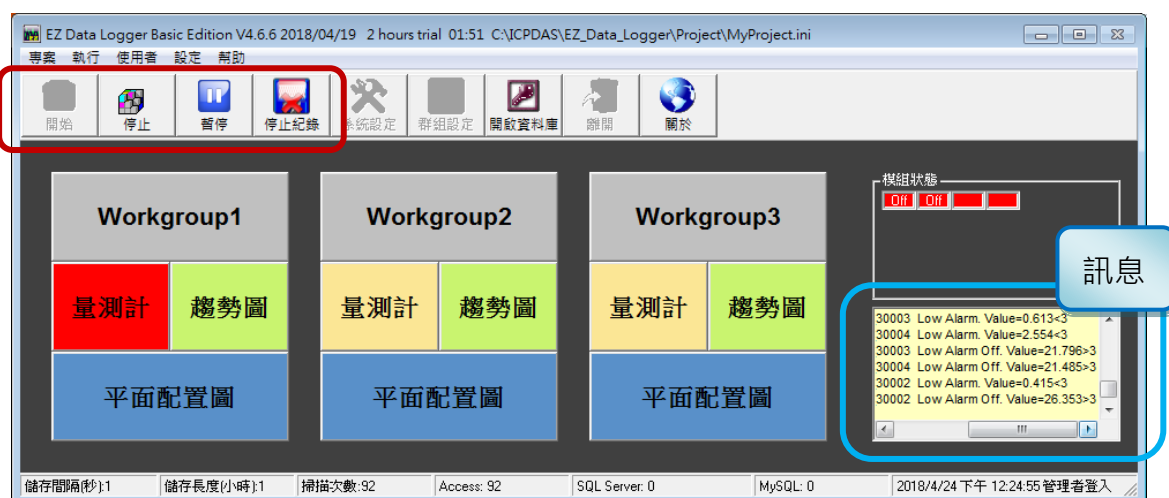
工作群組	
增加	可增加一個群組 (Workgroup)。
刪除	可刪除指定的群組。
屬性	<p><u>工作群組名稱:</u> 可指定方便識別的名稱。</p> <p><u>隨啟動開啟畫面:</u> 若已有加入通道設定，則可勾選量測計、趨勢圖、平面配置圖，讓程式啟動後，可開啟指定的畫面 (桌面左上角)。</p>
重設	可清除所有加入的通道。
通道 (Channels)	<p>可在單一通道上，點選滑鼠右鍵來刪除該通道。</p> 
主畫面	可回到主畫面。

5.3 開始執行 (Start) 或 進行模擬 (Simulation)

設定完通道與工作群組後，即可開始執行或模擬資料擷取。



開始/停止	可開始/停止擷取數據並記錄。
模擬/停止	可開始/停止模擬數據並記錄。
暫停/繼續	可暫停/繼續動態顯示趨勢圖。
記錄/停止記錄	可開始/停止將數據記錄到資料庫。
訊息	可顯示資料採集過程中的狀態訊息。



5.3.1 量測計 (Gauge)

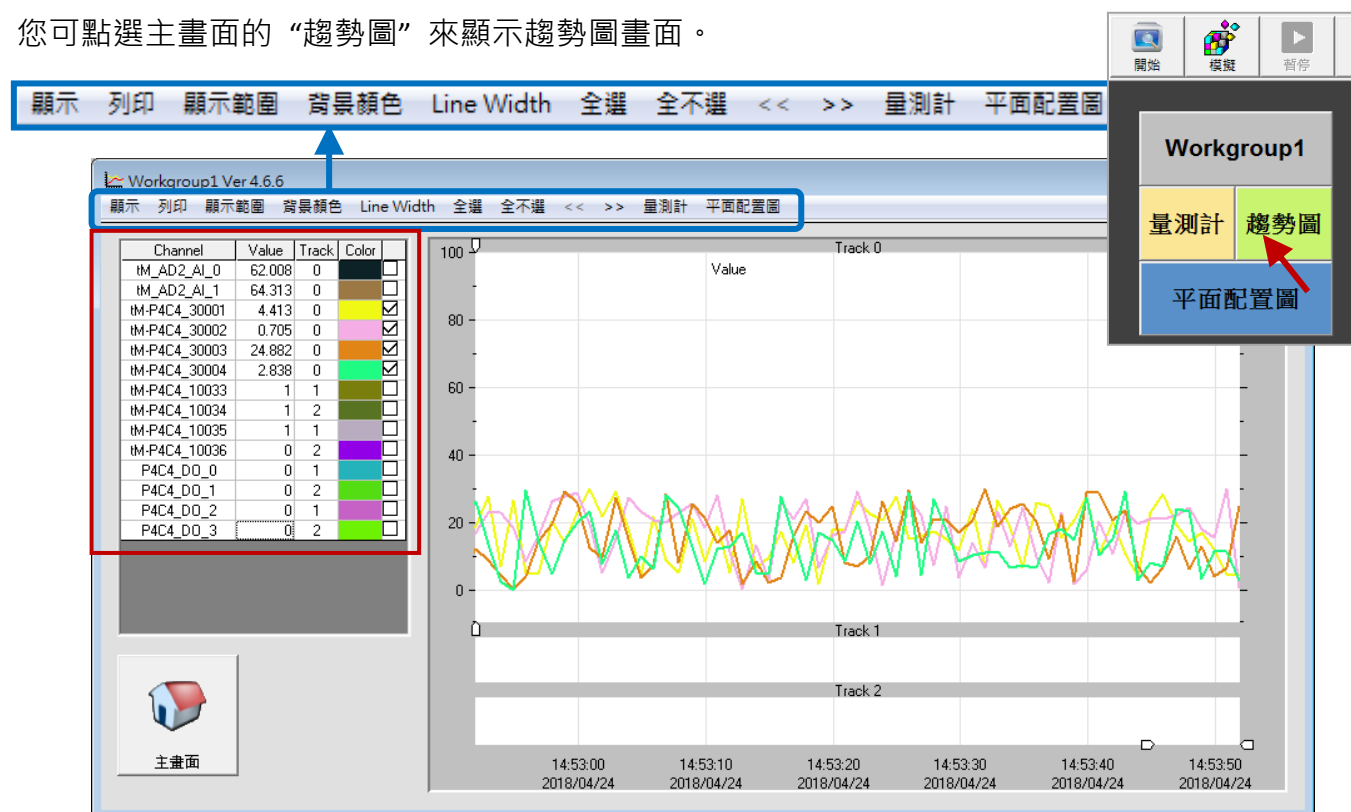
您可點選主畫面的“量測計”來顯示量測計畫面。若擷取/模擬的 AI 值超出警戒值 (High/Low Alarm) 時，警報燈會亮 且會播放警報聲，直到關閉為止。



範圍	AI 最大值 (橘色箭頭)/ AI 最小值 (紅色箭頭): 可設定量測計顯示的最大/最小值。
<<	可切換到前一個工作群組。
>>	可切換到下一個工作群組。
趨勢圖	可切換到“趨勢圖”畫面。
平面配置圖	可切換到“平面配置圖”畫面。
0 ~ 7	8 ~ 15
若通道數超過 8 個，可點選按鈕切換顯示的通道。	

5.3.2 趨勢圖 (Trend)

您可點選主畫面的“趨勢圖”來顯示趨勢圖畫面。



顯示	可暫停/繼續顯示動態趨勢圖。
列印	可列印趨勢圖 (包含工作群組、檔案日期/時間資訊)。
顯示範圍	可設定 Y 軸顯示的最大/最小值。 可設定 X 軸顯示的時間範圍 (單位: 分鐘)。
背景顏色	可設定“趨勢圖”的背景顏色。
Line Width	可設定“趨勢圖”的線寬。
全選 / 全不選	可顯示/隱藏全部的趨勢線。
<< / >>	可切換到前/下一個工作群組。
量測計	可切換到“量測計”畫面。
平面配置圖	可切換到“平面配置圖”畫面。
Channel	通道名稱。
Value	開始/模擬資料擷取時，點選清單上的 AO/DO 通道，可設定輸出數值。
Track	開始/模擬資料擷取時，點選 DI/DO 可切換資料的顯示 Track (1 ~ 2)。
Color	點選可變更趨勢線的顏色。(開始/模擬資料擷取時，無法變更顏色。)

設定輸出數值

Nickname:P4C4_DO_0

Tag:DO0

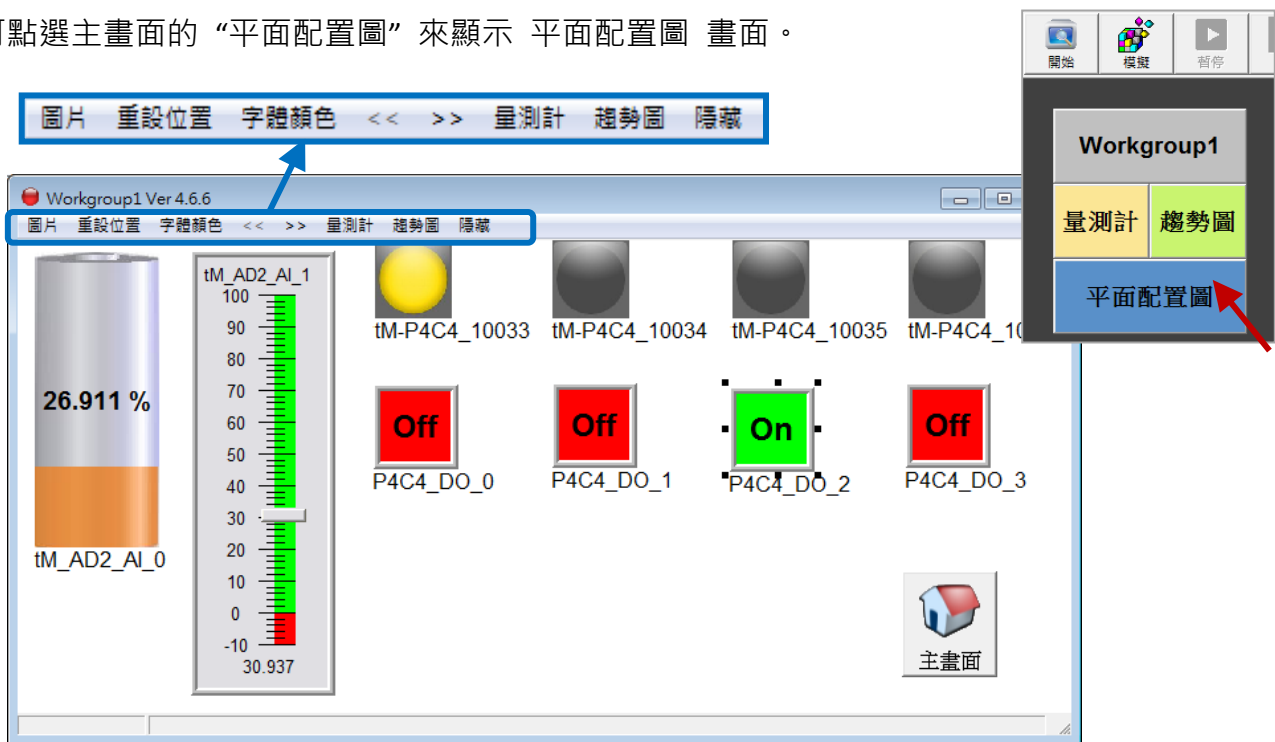
Description:P4C4-DO0


開 關 反向

手動 自動

5.3.3 平面配置圖 (Layout)

您可點選主畫面的“平面配置圖”來顯示 平面配置圖 畫面。



圖片	可設定/取消背景圖片，支援圖片格式 *.jpg, *.bmp。
重設位置	可重設所有物件的位置。
字體顏色	可設定標籤文字的顏色。 
<< / >>	可切換到前/下一個工作群組。
量測計	可切換到“量測計”畫面。
趨勢圖	可切換到“趨勢圖”畫面。
隱藏	可隱藏此畫面。

滑鼠右鍵功能表:

在“平面配置圖”畫面，物件的右鍵功能表中，可設定手動控制、自動輸出值 與 屬性。

註: “手動控制” 與 “自動輸出” 功能，用於邏輯控制。請參考 [EZ Data Logger 使用手冊 \(.chm\)](#)。
 當啟用了邏輯控制 (參考: 5.2 節，群組設定 - “Control Logic” 頁面)，EZ Data Logger 會自動對邏輯控制的 AO/DO 輸出數值，您也可視需要切換為手動輸出數值。

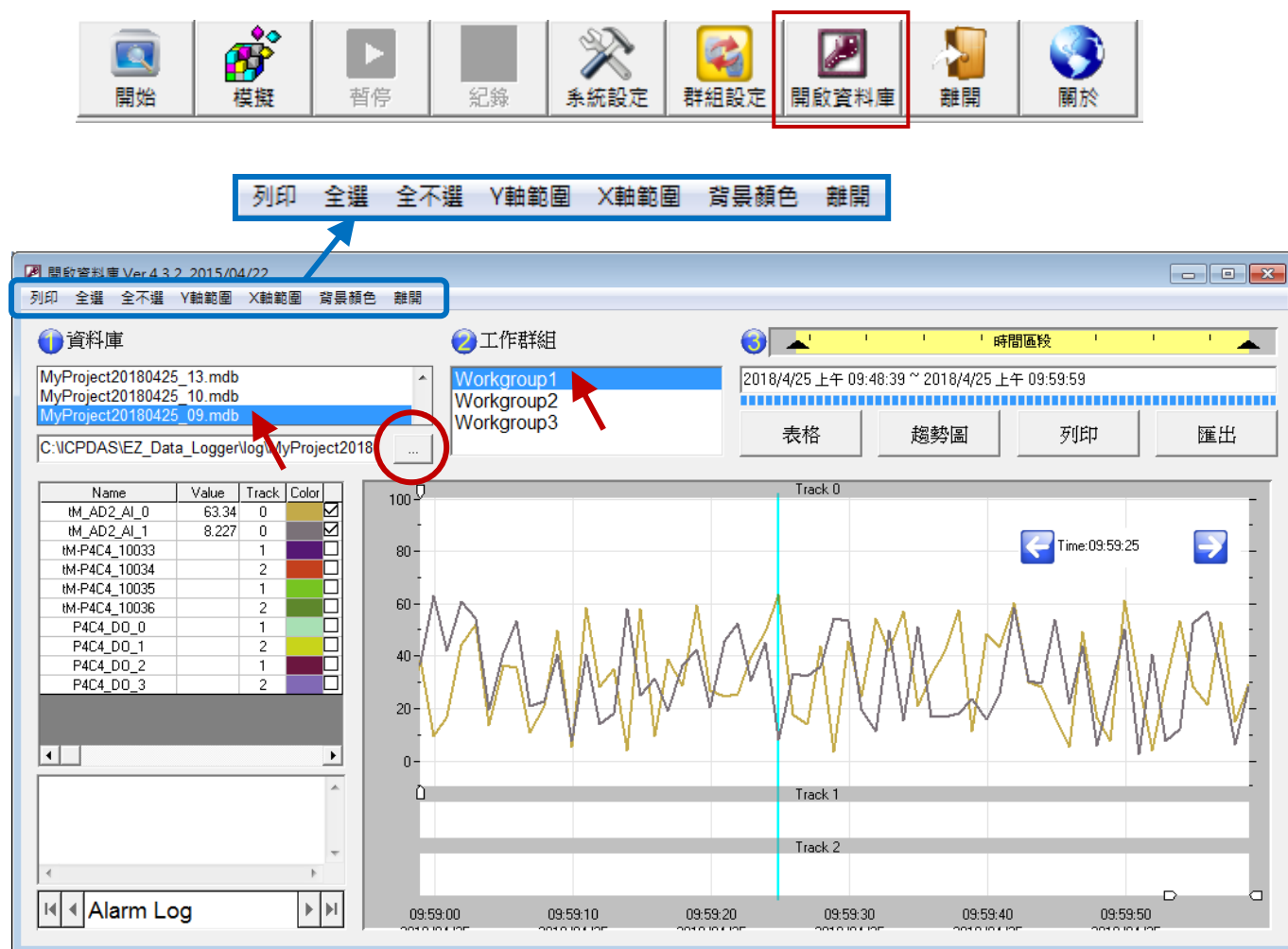
設定範例: (數值 = 123.456)

小數位數	單位	顯示
0	°C	123 °C
1	°F	123.5 °F
2	kg	123.46 kg
3	V	123.456 V

屬性	
Nickname	通道設定中的別名 (Alias) 。 (參考 5.2.3 節)
Tag name	通道設定中的標籤名稱 (Tag) 。 (參考 5.2.3 節)
Device	裝置列表中的模組名稱 。 (參考 5.2.2 節)
I/O Channel = n	可顯示 I/O 的通道編號 。
設定小數位數	可設定欲顯示的小數位數 (範圍: 0 ~ 3) 。
設定單位	可設定欲顯示的單位 (例如: °C) 。
輸入最大刻度	可設定欲顯示的最大刻度值 (例如: 100) 。
輸入最小刻度	可設定欲顯示的最小刻度值 (例如: -10) 。

5.4 開啟資料庫 (Open Database)

在 "開啟資料庫" 視窗，可開啟資料記錄檔 (.mdb) 並匯出成 Excel 或 CSV 檔案。



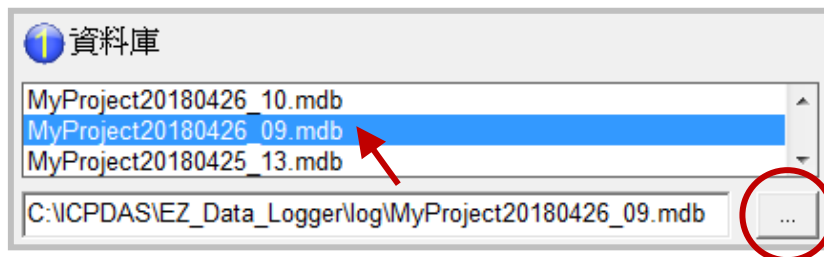
在功能表中，提供了列印、顯示通道資料 與 趨勢圖樣式的設定，後續會說明載入資料表格 與 趨勢圖的方式。

列印	可列印資料表格 或 趨勢圖。
全選 / 全不選	勾選/取消勾選所有核取方塊，以顯示/隱藏所有通道的記錄資料。
Y 軸範圍	可設定 Y 軸顯示的最大/最小值。
X 軸範圍	可設定 X 軸顯示的時間範圍 (單位: 分鐘)。
背景顏色	可設定趨勢圖的背景顏色。
離開	可離開此設定視窗。

註: 在開始/模擬資料擷取時，會無法修改 Y 軸範圍、X 軸範圍 與 背景顏色。

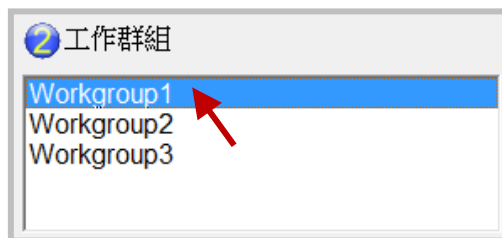
步驟 1: 選擇資料庫:

您可點選任一列表中的資料記錄檔 或 點選瀏覽按鈕 開啟其他位置的檔案。



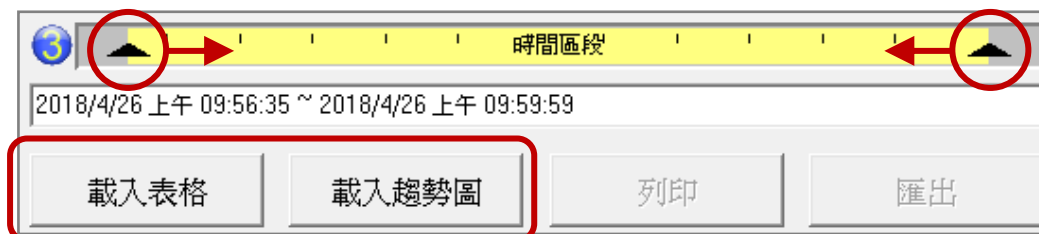
步驟 2: 選擇工作群組:

您可點選列表中，任何一個有加入通道設定工作群組。(參考: 5.2.3 節)



步驟 3: 指定時間區段 並 載入表格/趨勢圖:

您可拖動指標 (左/右) 來指定時間區段，再點選 載入表格 / 載入趨勢圖 按鈕，來載入表格資料 或 趨勢圖。



1) 載入表格: 點選“載入表格”按鈕，來載入表格資料。

註: 勾選 (或 取消勾選) 通道，可顯示 (或隱藏) 表格資料。

Name	Value	Track	Color	Check
tM_AD2_AI_0	0			<input checked="" type="checkbox"/>
AD2_AI_1	0			<input checked="" type="checkbox"/>
tM-AD8_30001	0			<input checked="" type="checkbox"/>
tM-AD8_30002	0			<input checked="" type="checkbox"/>
tM-AD8_30003	0			<input checked="" type="checkbox"/>
tM-AD8_30004	0			<input checked="" type="checkbox"/>
tM-AD8_30005	0			<input checked="" type="checkbox"/>
tM-AD8_30006	0			<input checked="" type="checkbox"/>
tM-AD8_30007	0			<input checked="" type="checkbox"/>
tM-AD8_30008	0			<input checked="" type="checkbox"/>
tM-P474_10033	1			<input checked="" type="checkbox"/>

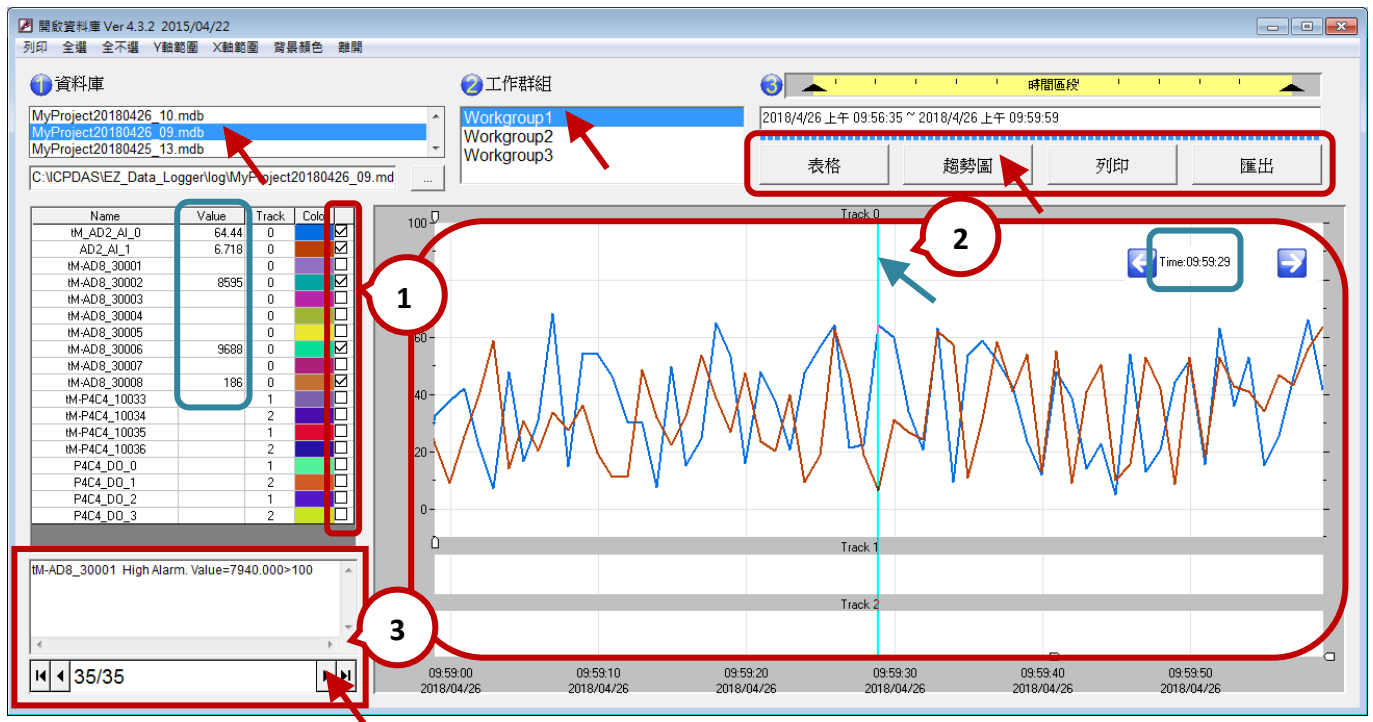
List	tM_AD2_AI_00	AD2_AI_11	tM-AD8_300012	tM-AD8_300023	SamplingTime	AlarmLog
0	32.57	65.405	3453	2275	2018/4/26 上午 09:56:35	tM-AD8_30001 High Alarm. Value=
1	24.24	52.635	4237	7194	2018/4/26 上午 09:56:36	
2	0.94	51.326	8297	5758	2018/4/26 上午 09:56:37	
3	65.04	21.602	8759	9218	2018/4/26 上午 09:56:38	
4	23.57	55.336	5556	8400	2018/4/26 上午 09:56:39	
5	28.64	30.365	3083	744	2018/4/26 上午 09:56:40	
6	65.45	12.215	353	9833	2018/4/26 上午 09:56:41	
7	67.74	33.313	3657	7423	2018/4/26 上午 09:56:42	
8	40.76	29.711	1717	7974	2018/4/26 上午 09:56:43	
9	31.76	45.304	3340	5685	2018/4/26 上午 09:56:44	
10	37.92	53.544	7581	4019	2018/4/26 上午 09:56:45	
11	41.86	6.666	6393	7741	2018/4/26 上午 09:56:46	
12	44.57	2.399	9787	7445	2018/4/26 上午 09:56:47	
13	5.9	18.194	3486	6585	2018/4/26 上午 09:56:48	
14	32.54	12.932	9082	1010	2018/4/26 上午 09:56:49	
15	13.47	46.153	6693	992	2018/4/26 上午 09:56:50	tM-AD8_30005 High Alarm off. Val
16	52.92	54.767	119	5756	2018/4/26 上午 09:56:51	tM-AD8_30005 High Alarm. Value=
17	60.87	57.277	4497	516	2018/4/26 上午 09:56:52	

2) 載入趨勢圖: 點選 “載入趨勢圖” 按鈕，來載入趨勢資料。

註: 勾選 (或 取消勾選) 通道，可顯示 (或隱藏) 趨勢線。

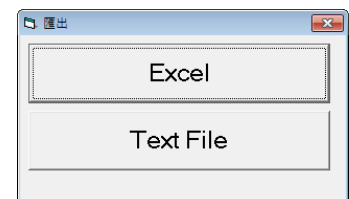
滑鼠點選在趨勢圖上，可顯示該時間點 與 通道數值。

點選切換按鈕，可顯示某筆警報訊息。



3) 列印: 點選 "列印" 按鈕，可列印趨勢圖。

4) 匯出: 點選 "匯出" 按鈕，可將資料檔存成為 Excel 或 CSV 檔。



匯出至 Excel 檔

A1		List						
	A	B	C	D	E	F	G	H
1	List	tM-AD2_AI_00	AD2_AI_11	tM-AD8_300023	tM-AD8_300067	tM-AD8_300089	SamplingTime	AlarmLog
2	0	32.57	65.41	2,275.00	2,808.00	5,854.00	09:56:35	
3	1	24.24	52.64	7,194.00	1,039.00	4,277.00	09:56:36	
4	2	0.94	51.33	5,758.00	1,038.00	7,431.00	09:56:37	
5	3	65.04	21.60	9,218.00	6,405.00	8,984.00	09:56:38	
6	4	23.57	55.34	8,400.00	6,729.00	6,476.00	09:56:39	
7	5	28.64	30.37	744.00	7,116.00	7,109.00	09:56:40	

匯出至文字檔

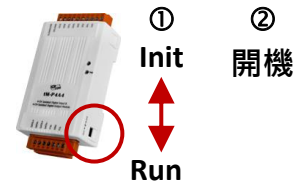
```

MyProject20180426_09_csv.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

List,AD2_AI_00,AD2_AI_11,AD8_300023,AD8_300067,AD8_300089, SamplingTime, AlarmLog,
0, 32.57, 65.405, 2275, 2808, 5854, 2018/04/26 09:56:35,tM-AD8_30001 High Alarm. Value=3
1, 24.24, 52.635, 7194, 1039, 4277, 2018/04/26 09:56:36,
2, 0.94, 51.326, 5758, 1038, 7431, 2018/04/26 09:56:37,
3, 65.04, 21.602, 9218, 6405, 8984, 2018/04/26 09:56:38,
4, 23.57, 55.336, 8400, 6729, 6476, 2018/04/26 09:56:39,
5, 28.64, 30.365, 744, 7116, 7109, 2018/04/26 09:56:40,
  
```

第 6 章 軟體開發工具

tM 系列模組可支援 DCON 與 Modbus 通訊協定，當模組採用 **DCON** 通訊協定，可使用 DCON Utility Pro 或 PACSDK 來設定/取得模組的資訊。當模組採用 **Modbus** 通訊協定，可搭配在 6.3 節介紹的軟體工具與圖控軟體 (SCADA)，來開發您的應用。



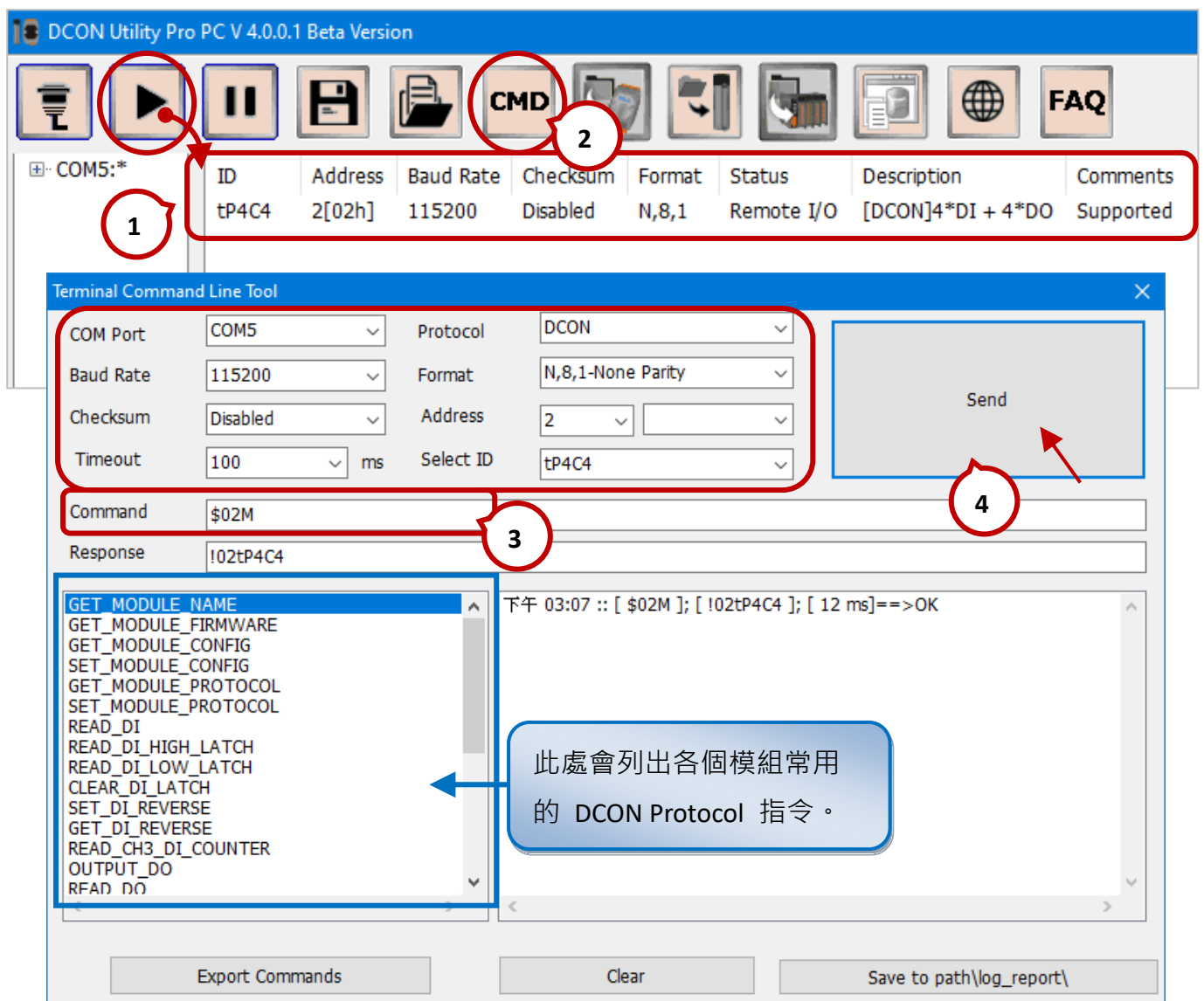
6.1 使用 DCON Utility Pro

步驟 1: 將模組設定在 Init 模式，才可設定 Protocol。

步驟 2: 搜尋 tM 模組，並將 Protocol 設為 DCON，再以 Run 模式重新啟動模組。

步驟 3: 使用“命令列工具”功能，於“Command”欄位輸入 DCON Protocol 指令。

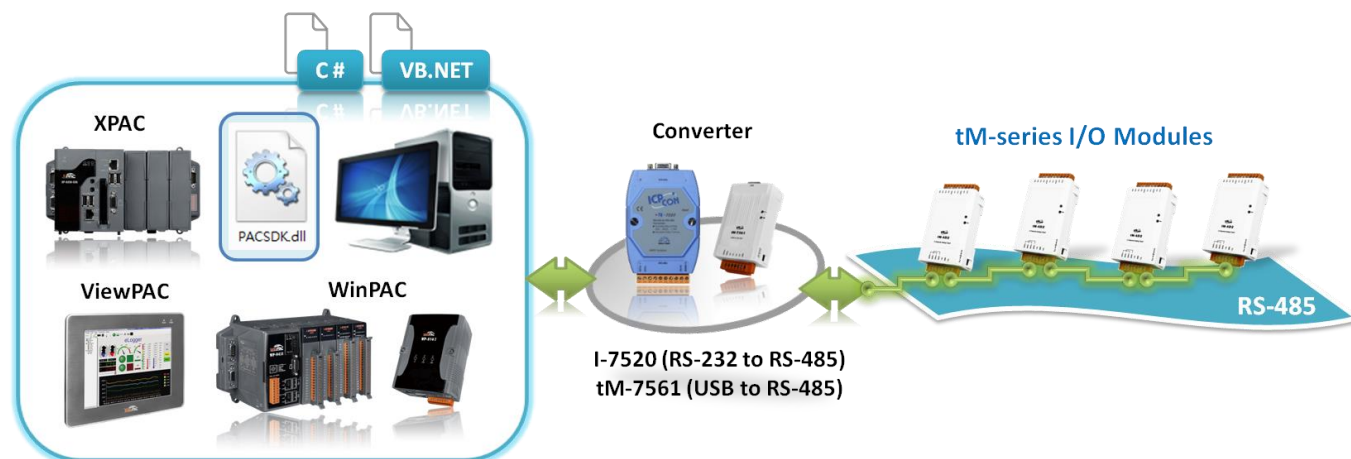
(參考: 附錄 E，DCON Protocol 指令。)



6.2 使用 PACSDK

PACSDK 是一個可在 PC 與 PAC (XPAC、WinPAC 與 ViewPAC 系列) 平台上使用的軟體開發套件，內含標頭檔 (Header file)、函式庫 (Library)、說明文件 與 相關工具等。PACSDK 可用於開發 VC/C#/VB.net 應用程式，來存取 tM 系列 I/O 模組的資料。

註: PACSDK 只支援 DCON 通訊協定。

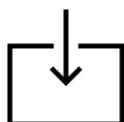


PACSDK

項目		PACSDK	說明
標頭檔 (Header file)		PACSDK.h	於 Windows PC 上，開發 VC 應用程式時，使用的函式庫
函式庫 (Library)		PACSDK.lib	
DLL 檔	C 程式使用	PACSDK.dll	於 PC/PAC 上，執行 VC 應用程式時，載入的函式庫
	.NET 程式使用 (C#、VB)	PACNET.dll	1) 開發 .NET 應用程式時，使用的 Manage 函式庫 2) 於 PC/PAC 上，執行 .NET 應用程式時，載入的函式庫

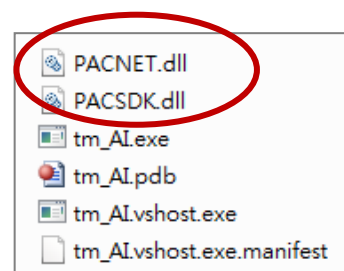
您可在下列位置下載相關檔案:

1) PACSDK:



您可依據使用的開發平台 (PC 或 PAC) 來下載 SDK。

若您是採用下列 **PC 版** 與 **XPAC (WES)** 開發平台，下載後，請將 PACSDK.dll、PACNET.dll 複製到應用程式 (exe 檔) 所在的目錄中。



PC 版本
https://www.icpdas.com/en/product/guide+Software+Development__Tools+PAC__SDK

XPAC, iPPC (WES7)
適用 XP-8000-WES7、XP-9000-WES7、iPPC (WES7) https://www.icpdas.com/en/download/show.php?num=2540

若您是採用下列 **XPAC (CE6)**、**WinPAC (CE5)**、**WinPAC/ViewPAC (CE7)** 平台，請於網頁下載並安裝相關檔案，來開發應用程式。

開發工具: Visual Studio 2008 專業版或更早的版本

XPAC (CE6)
適用 XP-8000-CE6 https://www.icpdas.com/en/download/show.php?num=2473
WinPAC (CE5)
WinCE5.0 PAC/ViewPAC (無 I/O 插槽) https://www.icpdas.com/en/download/show.php?num=2594 WinCE5.0 PAC/ViewPAC (有 I/O 插槽) https://www.icpdas.com/en/download/show.php?num=2593
WinPAC/ViewPAC (CE7)
WinCE7.0 PAC/ViewPAC (無 I/O 插槽) https://www.icpdas.com/en/download/show.php?num=2409 WinCE7.0 PAC/ViewPAC (有 I/O 插槽) https://www.icpdas.com/en/download/show.php?num=2348

2) PAC API 手冊:



您可在手冊中了解 PACSDK 的安裝方式、開發環境設置 與 **PAC_IO** API 功能。
(pac_standard_api_manual_x.x.x.pdf)

各平台的手冊路徑如下表:

PC
https://www.icpdas.com/en/download/show.php?num=1049
PAC (WES)
https://www.icpdas.com/en/download/show.php?num=2527
PAC (WinCE)
https://www.icpdas.com/en/download/show.php?num=2407

此處會列出與 I/O 模組相關的指令。

```
int iBitValue = 1;
bool ret = PACNET.IO.WriteDOBit(hPort, iSlot, iDO_TotalCh, iChannel, iBitValue);
PACNET.UART.Close(hPort);
```

[C#]

```
// If using the remote I/O such as the I-7K, M-7K, tM series module
IntPtr hPort;
hPort = PACNET.UART.Open("");
byte iAddr = 1;
int iChannel = 2;
int iDO_TotalCh = 8;
int iBitValue = 1;
bool ret = PACNET.IO.WriteDOBit(hPort, PAC_REMOTE_IO(iAddr), iDO_TotalCh, iChannel,
iBitValue);
PACNET.UART.Close(hPort);
```

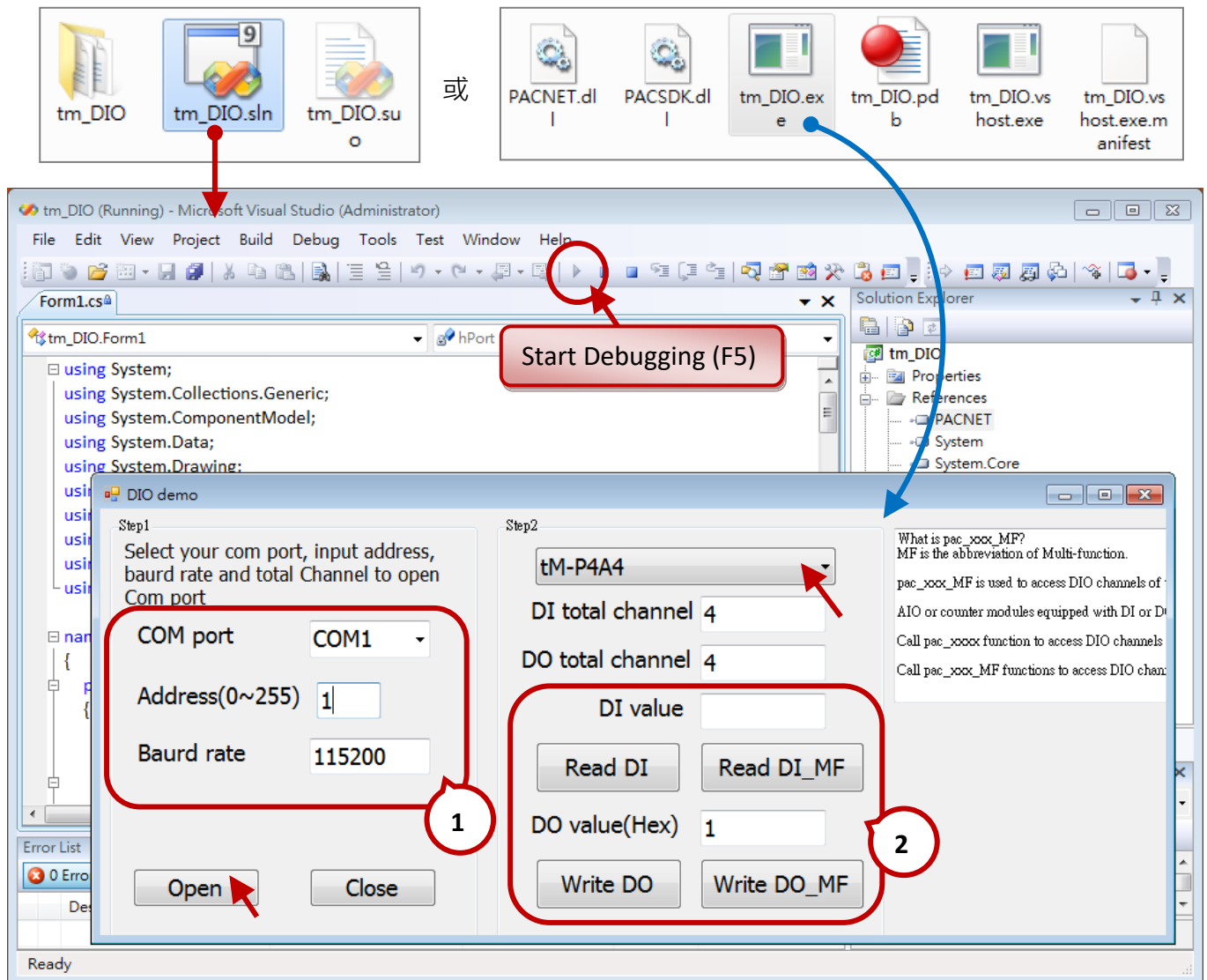
Remarks

The function can support for Local or Remote. When the module is local, the second Parameter's range is from 0 to 7. If remote, the second Parameter need use the macro, PAC_REMOTE_IO(0...255), which range is from 0 to 255.

3) C# 範例程式 (DCON):

www.icpdas.com/en/download/show.php?num=2876

以 DIO demo 為例，您可參考 2.4 節完成硬體連接，接著滑鼠雙擊 tm_DIO.sln 開啟專案，再點選 “Start Debugging (F5)” 按鈕。或者，滑鼠雙擊 ...\\tm_DIO\\bin\\Debug 內的 tm_DIO.exe。



步驟 1: 選擇使用的 COM Port，並填入 tM 模組的位址 與 Baud rate，再點選 “Open” 開啟 COM Port。此例，COM Port = COM9、Address = 1 與 Baud rate = 9600，若您不清楚模組的設定值，可使用 DCON Utility Pro 設定。**註:** 模組的 Protocol 需設為 DCON。

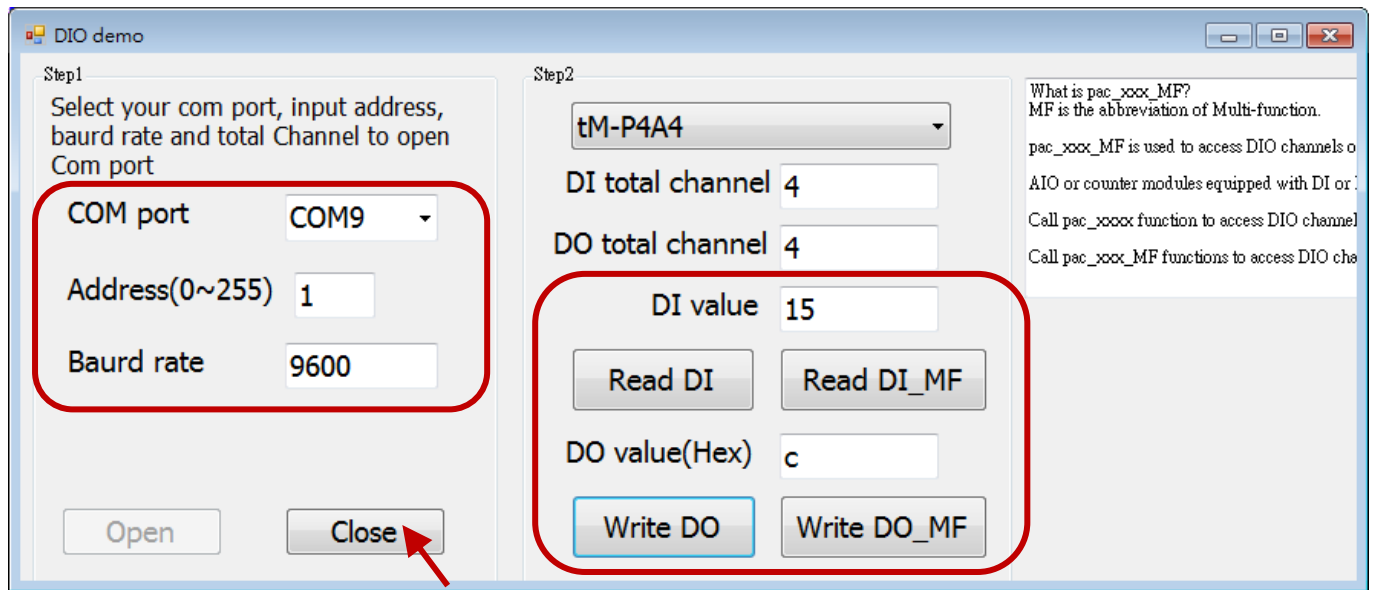
步驟 2: 選擇 tM 模組的型號後，在 “DI total channel” 與 “DO total channel” 欄位會自動顯示 DI 與 DO 通道數。此例，使用 tM-P4A4。您可點選 “Read DI” 來讀取 DI 值，或在 “DO value(Hex)” 欄位輸入數值 (16 進制，例如: "C" = 1101)，再點選 “Write DO” 寫入 DO 值。

注意: 若您使用 Multi-function I/O，意即同時支援類比與數位 IO 的模組 (例如: tM-AD4P2C2)，請點選 “Read DI_MF” 或 “Write DO_MF”。

完成上述步驟後，畫面如下:

DI value = 15 ₍₁₀₎ = 1111，表示 DI3、DI2、DI1、DI0 皆為 ON。

DO value = c ₍₁₆₎ = 1101，表示 DI3、DI2 與 DI0 皆為 ON，DI1 為 OFF。



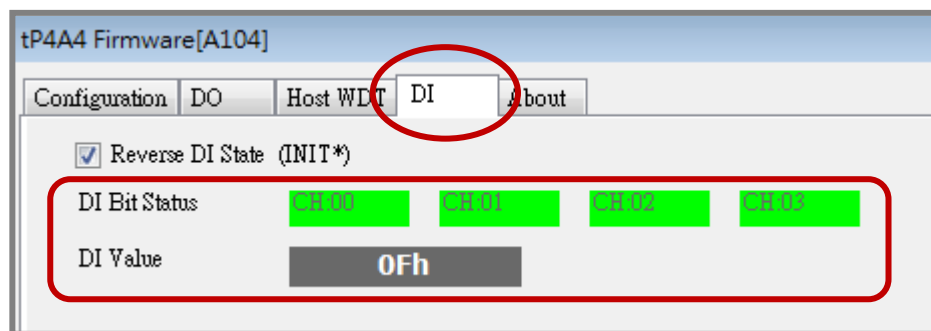
步驟 3: 點選 “Close” 釋放 COM Port。

您可看模組上燈號的變化 或 使用 DCON Utility Pro 查看 DI 與 DO 的狀態。

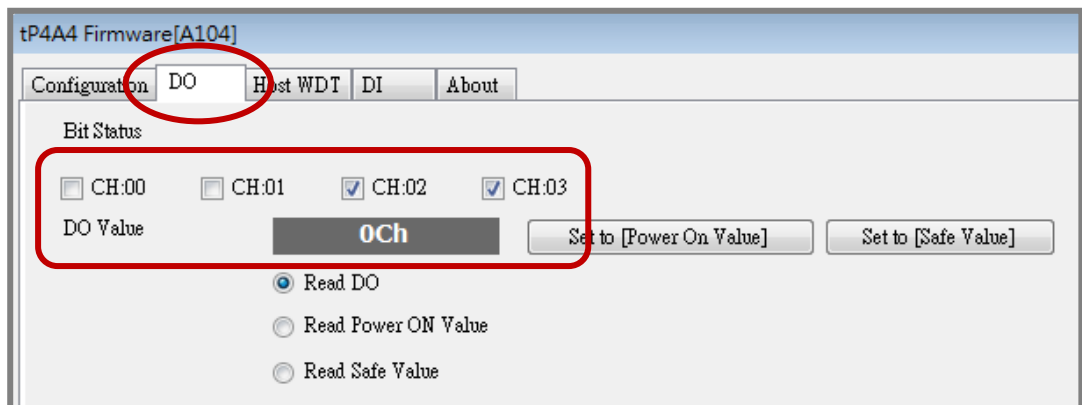
步驟 1: 搜尋 I/O 模組 (參考: 3.1 節)。**注意:** 請記得先釋放 Demo 使用的 COM Port。

步驟 2: 滑鼠雙擊搜尋到的模組型號，來開啟設定頁面。再點選 DI 或 DO 頁籤，來查看結果。

[DI 頁面]:



[DO 頁面]:



6.3 Modbus 開發工具 與 圖控軟體

泓格科技提供多種可支援 Modbus RTU/ASCII 通訊協定的軟體工具 與 圖控軟體，可輔助您開發應用系統。

6.3.1 軟體工具 – OPC DA Server 與 UA 系列產品

NAPOPC DA Server 是一套免費且支援泓格科技 (ICP DAS) 產品的

OPC DA Server 軟體。使用者可藉由 NAPOPC DA Server 來整合

ICP DAS 的遠程 I/O 模組 及 PAC 控制器板卡等設備，並讓有

支援 OPC DA Client 功能的圖控、人機介面 與 資料庫等軟體，對設備資料進行擷取或寫入，

以達成程序控制 與 工業自動化系統應用。此外，使用者還可選購 Modbus 通訊協議的授權，

讓 PC 版 (NAPOPC_ST DA Server)，也可支援第三方 Modbus 設備。

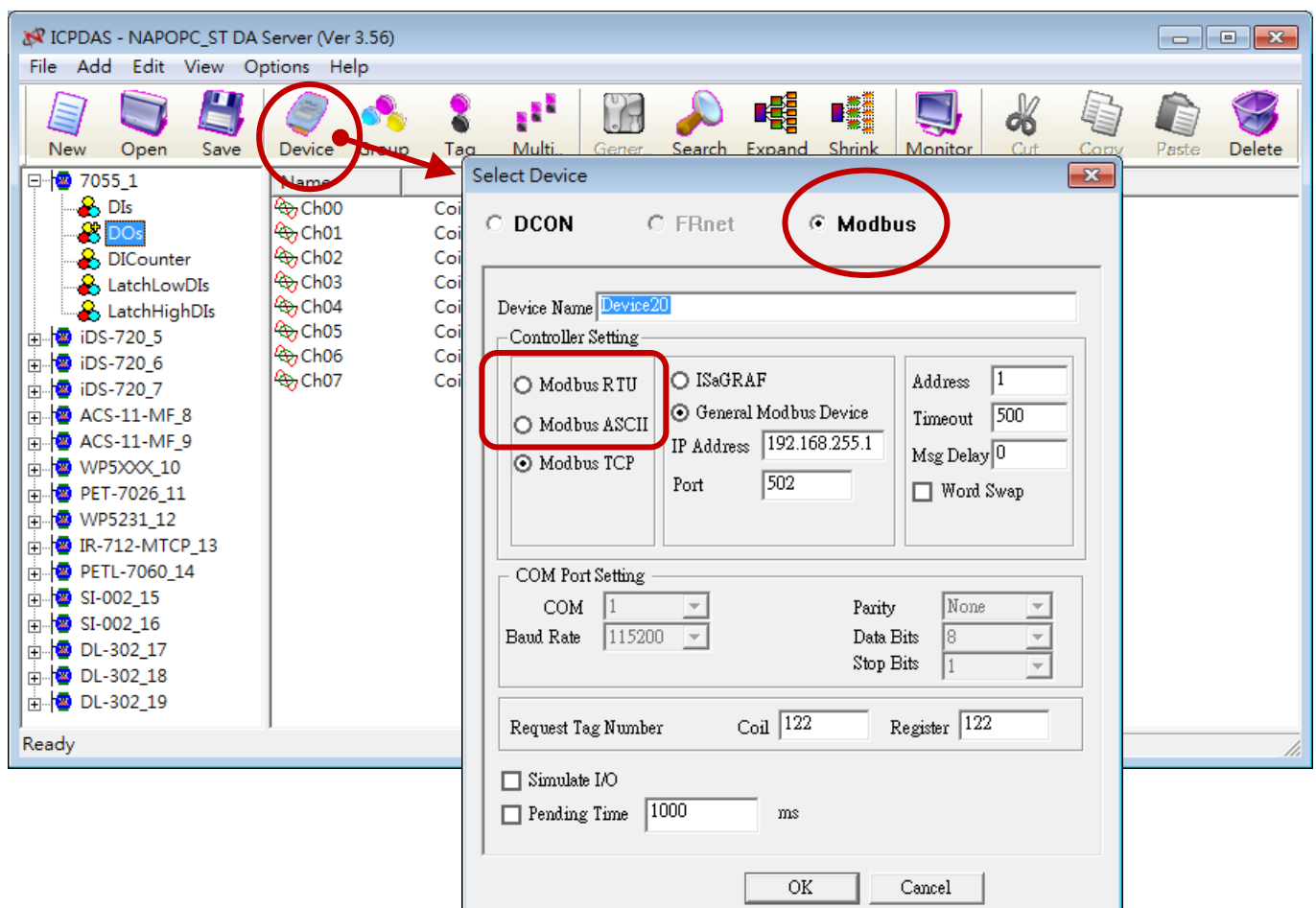


產品網頁/ Data Sheet/ DM:

https://opc.icpdas.com/opcda_c.htm

https://ftp.icpdas.com.tw/pub/cd/8000cd/napdos/napopcsvr/data_sheet/napopc_st_data_sheet_c.pdf

<https://www.icpdas.com/root/support/catalog/pdf/Brochure/NAPOPC/NAPOPC-DM-tc.pdf>



UA 系列工業物聯網通訊服務器



UA 系列 是一款 IIoT 工業物聯網通訊服務器，用來整合 IT 與 OT 系統設備。UA 系列具備 IIoT 閘道器功能，意即可透過 Modbus TCP/RTU/ASCII、MQTT 或 EtherNet/IP 通訊協議，存取/控制工廠中遠程 I/O 模組與控制器的資料，再將 I/O 資料轉換為符合 OPC UA 或 MQTT 通訊協議，以供 MES、ERP、SCADA 和雲端服務的應用需求。此外，還提供 I/O 資料寫入遠端資料庫 或 寫入本地端檔案 (.csv) 的功能，來儲存歷史資料。

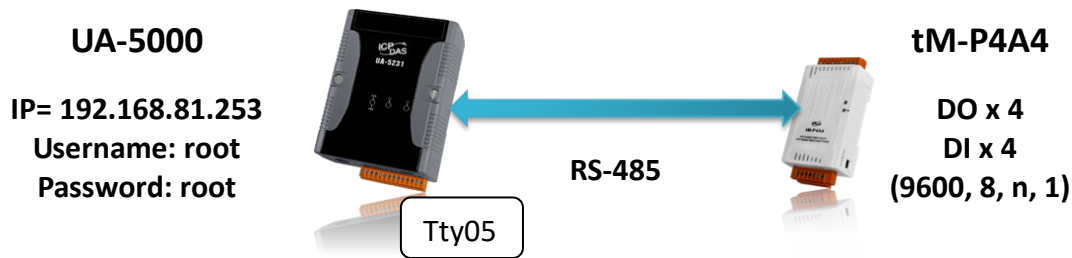
UA 系列可連線 IoT 雲端 (Cloud) 平台，並將 I/O 資料傳送至像是 Amazon AWS 與 Windows Azure 等平台。另外，也可連線至 IFTTT (If This Then That) 雲端平台上支援的網路服務 APP，當發生特殊事件時，用戶可透過手機 APP 收到訊息通知。

UA 系列產品可增進 IT 與 OT 系統的互聯網與協同操作能力，讓用戶可輕鬆設定、管理，並加速佈署工業物聯網。

請參訪泓格科技 UA 系列產品網頁，了解更多詳細資訊。

https://www.icpdas.com/en/product/guide+IIoT+Controller_Server+Communication__Server

本範例將透過 UA 系列的 RS-485 介面，來連接 tM-P4A4 模組，藉以讀/寫模組的 Modbus RTU I/O 資料，並加入 Modbus RTU 轉 OPC UA 通訊轉換功能。

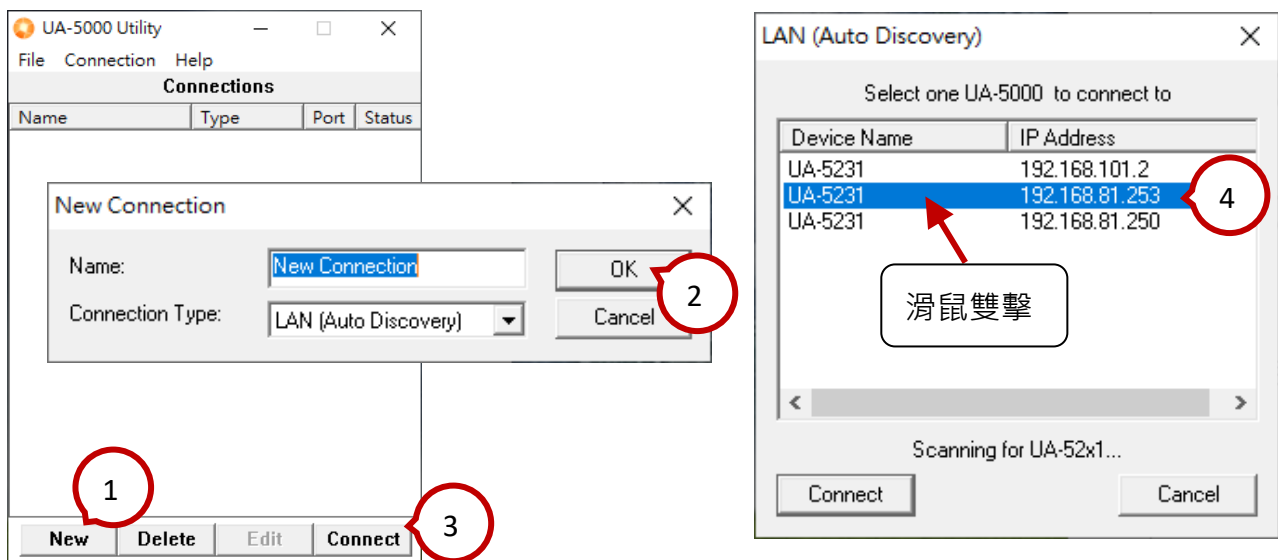


設定步驟

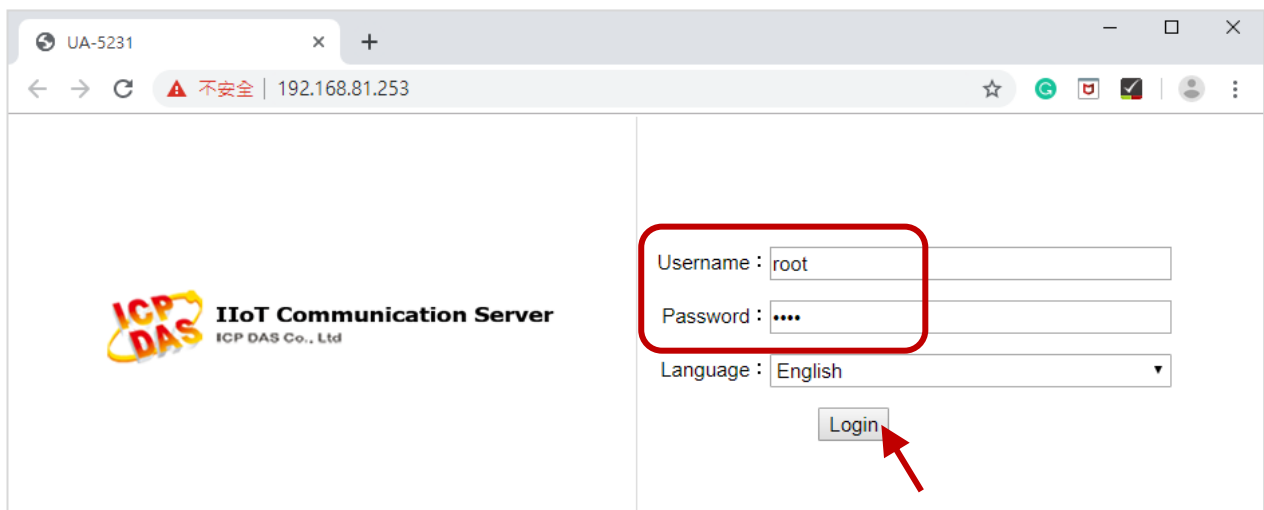
1. 登入 UA-5000 系列的設定網頁。

若您不知道 UA-5000 系列的 IP，可下載 UA-Series Utility 搜尋網路上的 UA 設備。

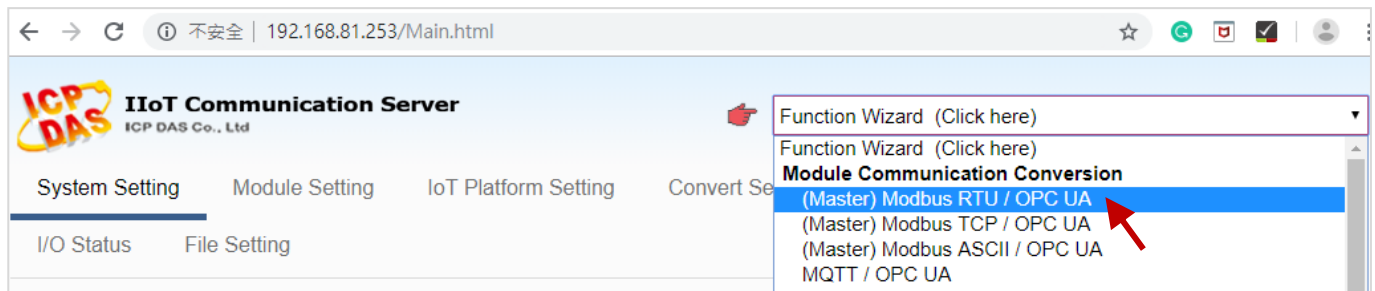
<https://www.icpdas.com/en/download/index.php?nation=US&kind1=&model=&kw=ua->



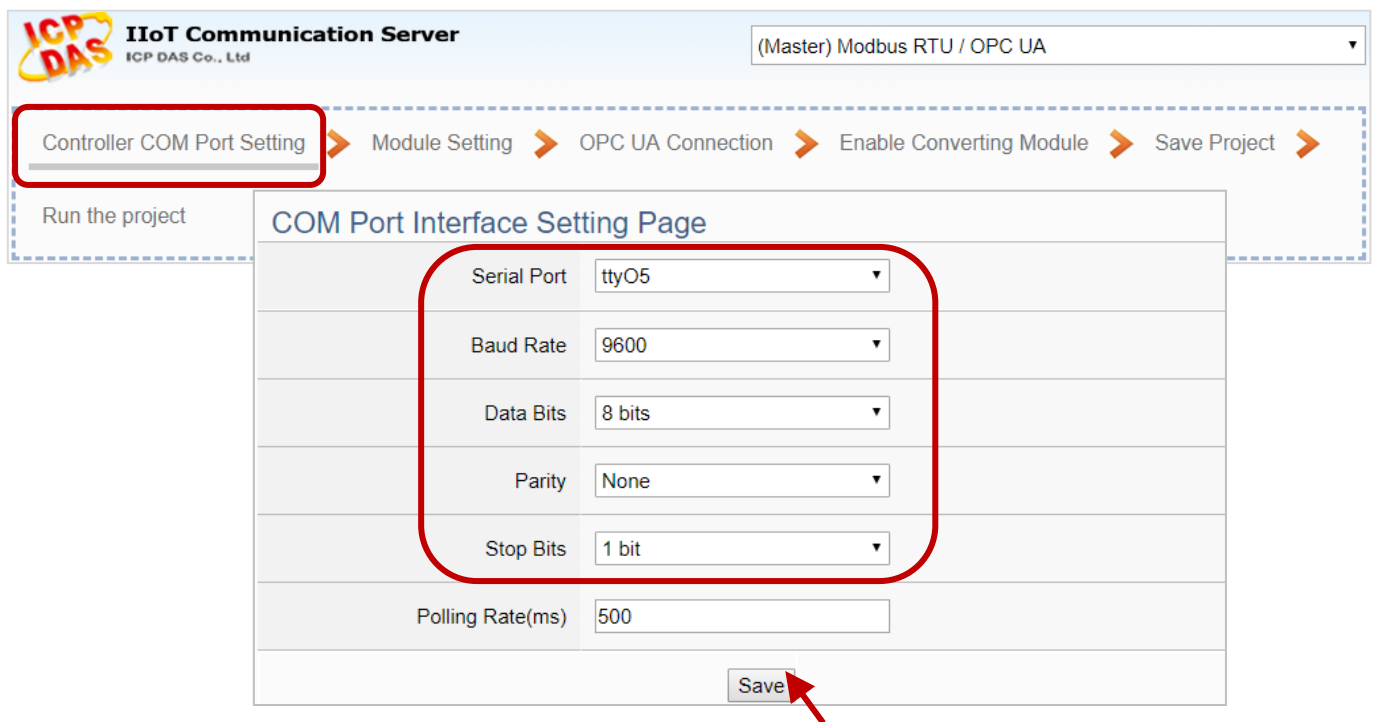
滑鼠雙擊找到的 UA 設備，會自動開啟 Web 瀏覽器，請輸入 Username 與 Password (root) 並點選 Login 按鈕，登入設定頁面。



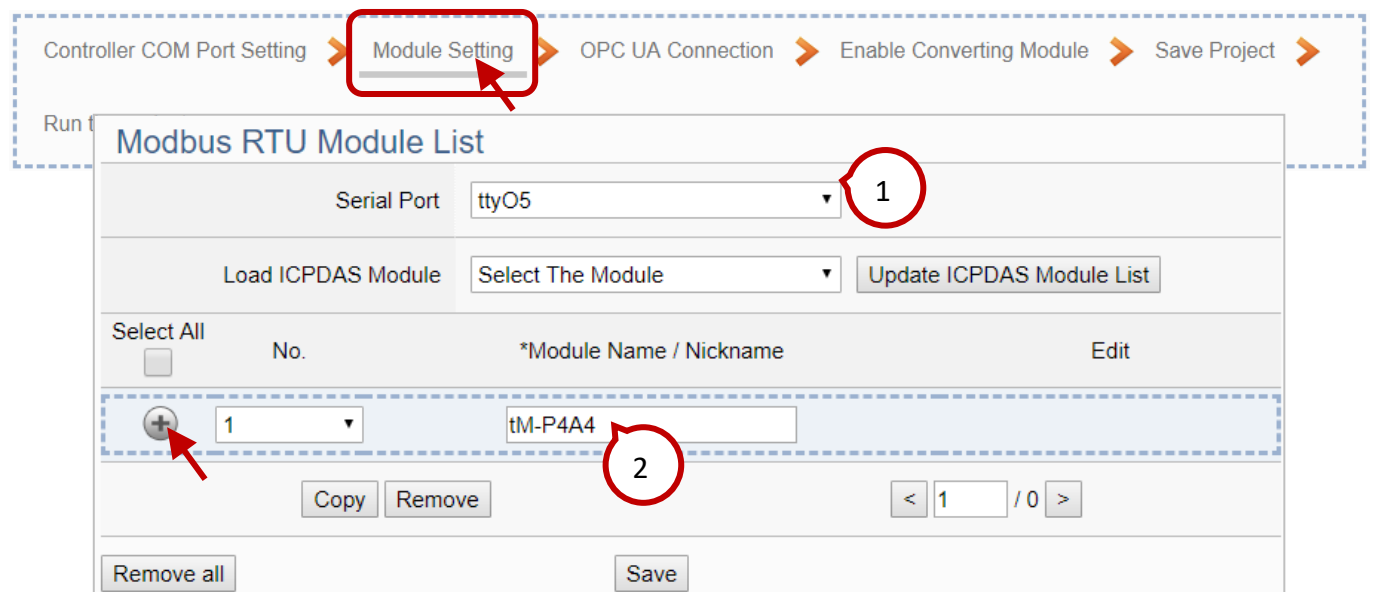
2. 選擇 (Master) Modbus RTU/OPC UA，表示需使用 Modbus RTU 與 OPC UA 的通訊轉換。



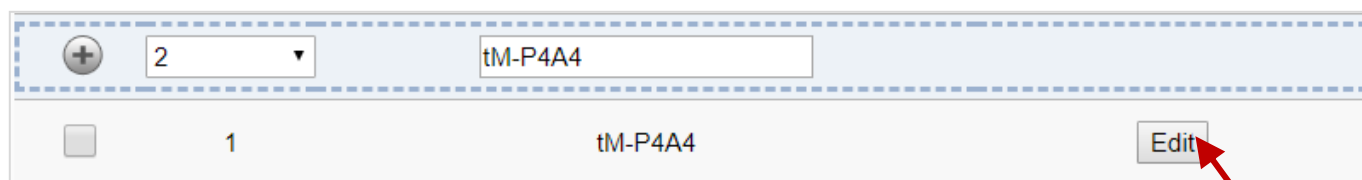
3. 首先，設定控制器的 COM Port (例如: TtyO5) 與 通訊參數 (9600, 8, n, 1)，再按 Save 儲存設定。



4. 點選 Module Setting，設定 COM Port 並輸入模組名稱，再按 “+” 按鈕新增此模組。

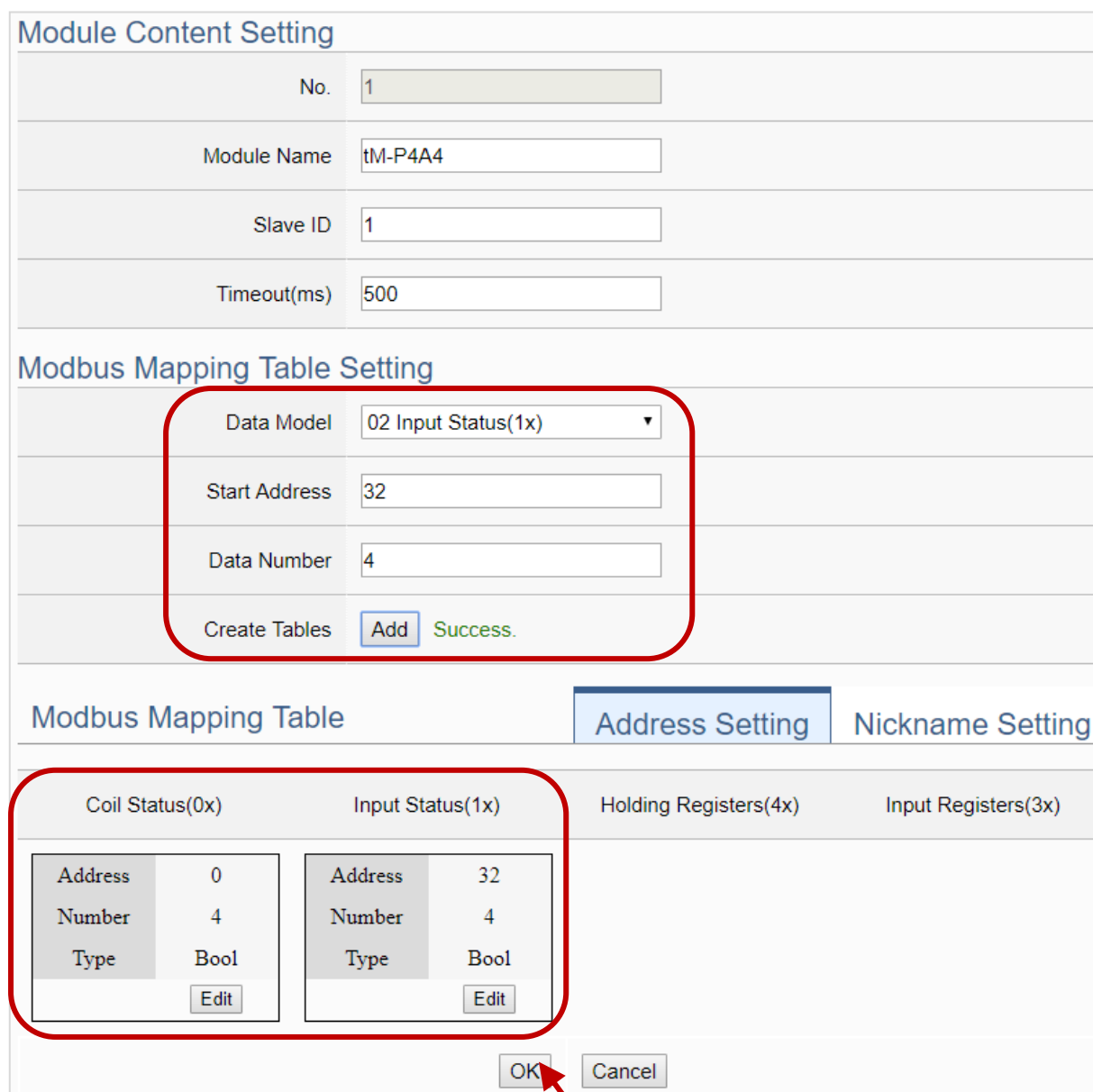


接著，點選 Edit 按鈕進入編輯畫面。



	No.	Module Name	
<input type="checkbox"/>	1	tM-P4A4	Edit

5. 此例使用 tM-P4A4 模組 (4 DI · 4DO)，請參考附錄 C 查詢 Modbus 的起始位址。設定好 Data Model、Start Address 與 Data Number 後，點選 Add 按鈕來新增 Modbus 對應表，再按 OK。



Module Content Setting

No.	1
Module Name	tM-P4A4
Slave ID	1
Timeout(ms)	500

Modbus Mapping Table Setting

Data Model	02 Input Status(1x)
Start Address	32
Data Number	4
Create Tables	Add Success.

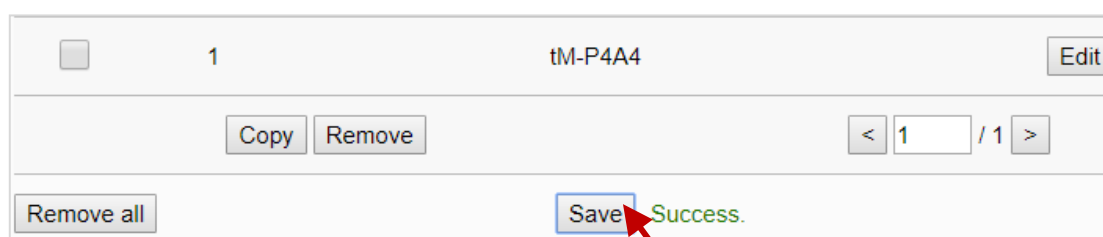
Modbus Mapping Table

Address Setting Nickname Setting

Coil Status(0x)		Input Status(1x)		Holding Registers(4x)		Input Registers(3x)	
Address	0	Address	32				
Number	4	Number	4				
Type	Bool	Type	Bool				
Edit		Edit					

OK Cancel

此時，會回到上一步驟的 Modbus RTU Master List 畫面，請記得按 Save 按鈕，儲存設定。



	No.	Module Name	
<input type="checkbox"/>	1	tM-P4A4	Edit

Copy Remove

Remove all Save Success.

6. 點選 OPC UA Connection，進入 Local Server 設定。預設啟用匿名登入，此畫面只需點選 Save 按鈕即可。

OPC UA Connection

Server

Server Name ICPDAS_OPC_UA_Server

Port 48010

Save Success.

User Identity Tokens

Anonymous Login ☒ Enabled

User Password Login ☒ Enabled

Certificate Login ☐ Enabled

Save Success.

7. 點選 Enable Converting Module，勾選要啟用通訊轉換的模組，並按 Save 按鈕。

Enable Converting Module

Modbus RTU Module List

No.	*Module Name / Nickname	Edit	All Enabled
1	tM-P4A4	Edit	<input checked="" type="checkbox"/>

< 1 / 1 >

Save Success.

8. 點選 Save Project 來儲存專案，再點選 Run the Project 更新並執行新專案。此時，UA 控制器已可處理 tM 模組的通訊轉換了。

Save Project

Run the project Success.

若想了解更多相關應用內容，請參訪：

https://www.icpdas.com/en/product/guide+IIoT+Controller_Server+Communication__Server

6.3.2 軟體工具 – nModbus



nModbus 是針對 Modbus 通訊的一個免費軟體開發工具 (SDK)，可使用 C# 來實作 Modbus 通訊協定。此軟件是由一群志願者開發、維護 並免費提供給大眾使用。

ICP DAS 基於官方發佈的版本 ([NModbus net-2.0 1.11.0.0-source.zip](http://www.icpdas.com/en/download/show.php?num=1024))，進而驗證並改善 DLL。程式開發者可在 Windows PC 或 WinCE 設備上，使用 ICP DAS 所發佈的 DLL 檔，來開發 Modbus 應用程式。

您可在下列位置，下載 DLL、nModbus API 手冊 與 範例程式。更多資訊，可見 nModbus 網頁：
https://www.icpdas.com/en/product/guide+Software+Development__Tools+Modbus__Tool

nModbus API 手冊:

<https://www.icpdas.com/en/download/show.php?num=1024>

範例程式:

<https://www.icpdas.com/en/download/show.php?num=1025>

● **PC 版:**

- DLL: log4net.dll, nmodbuspc.dll

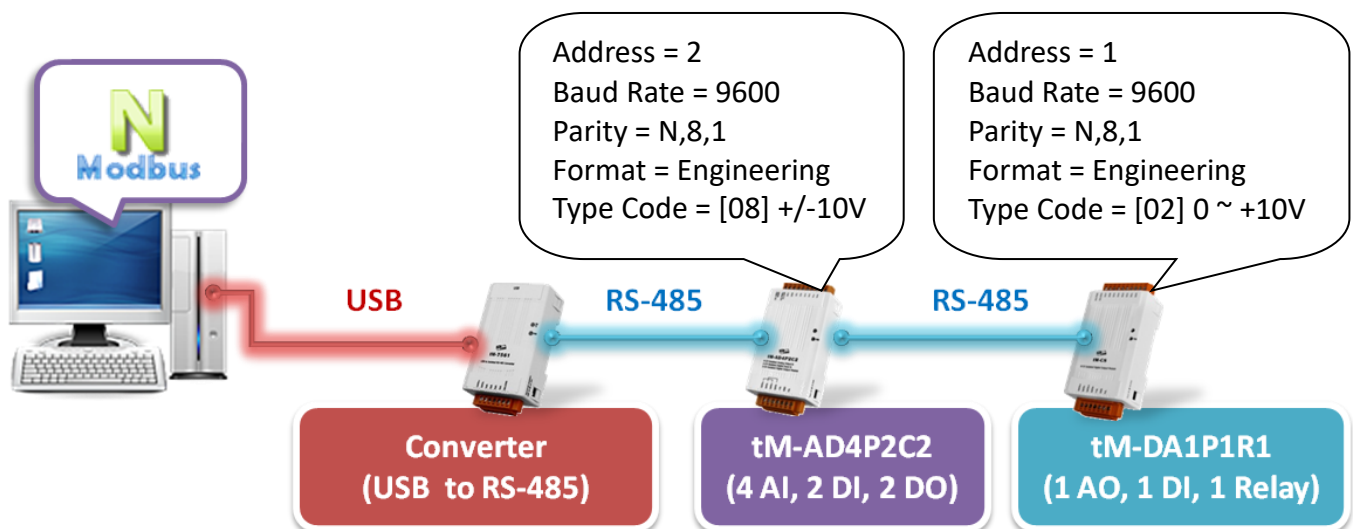
● **WinCE 版:**

- DLL: cabcdll.dll, fc19.dll, nmodbusce.dll

以下將介紹一個適用於 tM 模組的 Modbus RTU Master 範例，請先下載範例。

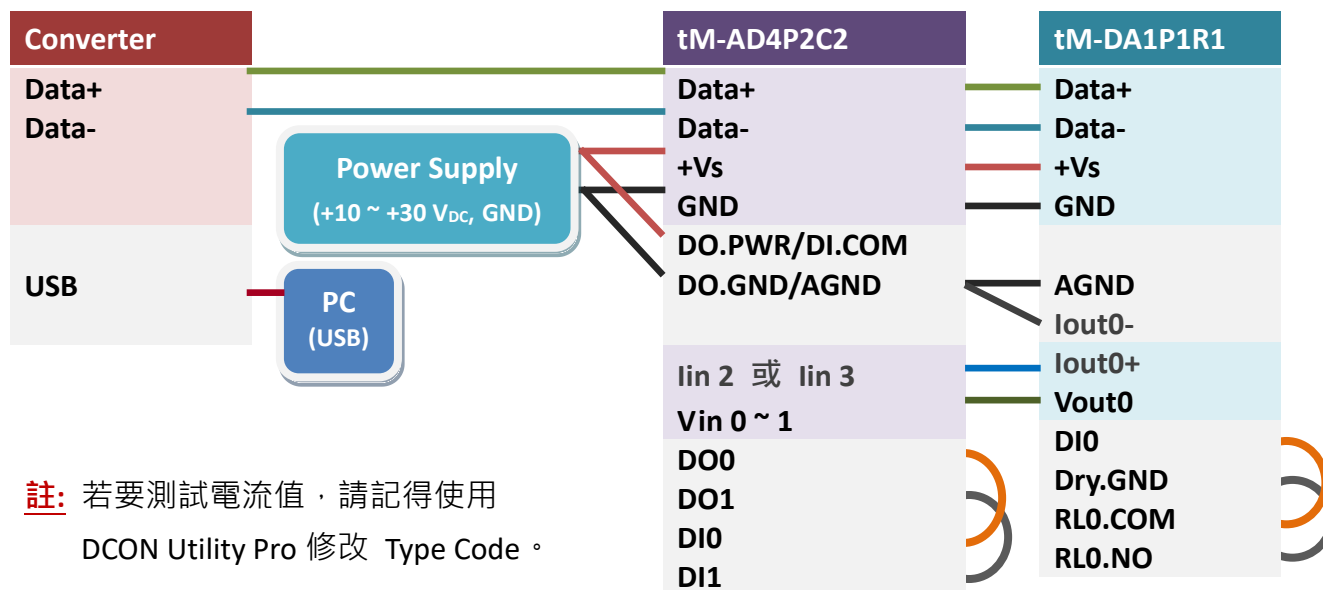
www.icpdas.com/en/download/show.php?num=2876 (C# Demo for Modbus Protocol)

使用模組前，請使用 DCON Utility Pro 進行設定。(參考: 第 3 章)



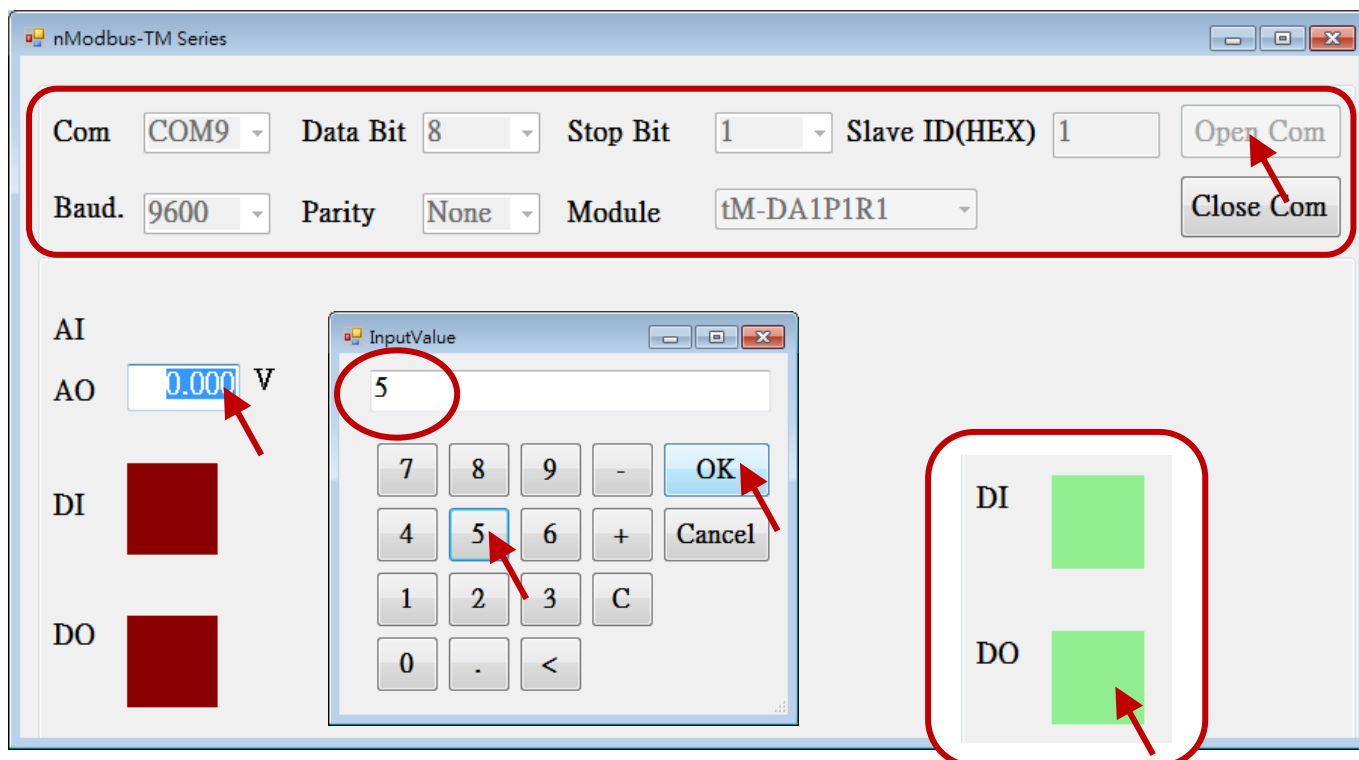
硬體接線 (測試用)：

此例，使用了一個 tM-AD4P2C2 與 tM-DA1P1R1 模組，為了測試 DI/DO/AI/AO 的狀態，請先接好設備的接線。您可在 2.3 節 查看模組的接線方式。



滑鼠雙擊 PC 上 ...\modbusrtu_master_tm\bin\Debug 路徑下的 modbusrtu_master_tm.exe。接著，填入模組的通訊參數，並點選“Open Com”按鈕來進行通訊。

您可在 tM-DA1P1R1 輸入 AO 值，再查看 tM-AD4P2C2 的 AI0, AI1 值的變化。或者，點選 DO 按鈕，並查看 DI 狀態的變化。



目前 tM-AD4P2C2 的 AI0, AI1 = 4.957 V，您可點選 DO 按鈕，並查看 DI 狀態的變化。



6.3.3 軟體工具 – LabVIEW

LabVIEW (**L**aboratory **V**irtual **I**strumentation **E**ngineering **W**orkbench) 是由美國國家儀器公司所開發的圖形化程式編譯平臺。LabVIEW 提供了一個簡單易懂的圖形化開發環境，並支援多種硬體驅動與軟體分析工具，能大幅縮短開發控制程式的時間，因此受到系統開發及研究人員的喜愛。目前廣泛的被應用在實驗室控制測試與工業自動化控制的領域。

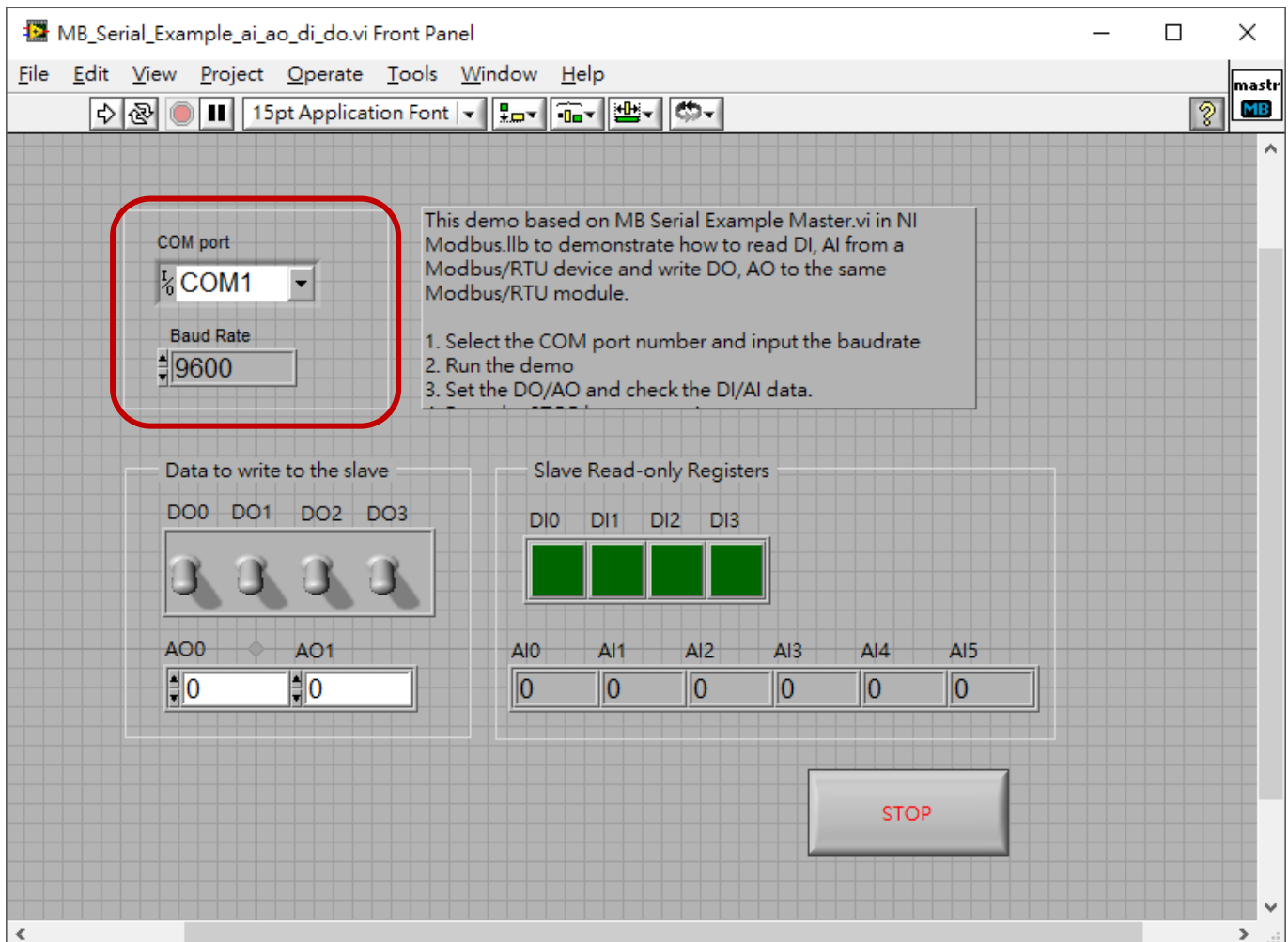
您可在以下位置，下載 Modbus RTU 範例程式 (ModbusRTU.zip)，並連接 tM 系列模組。

https://www.icpdas.com/en/product/guide+Software+Development__Tools+LabVIEW__Tools

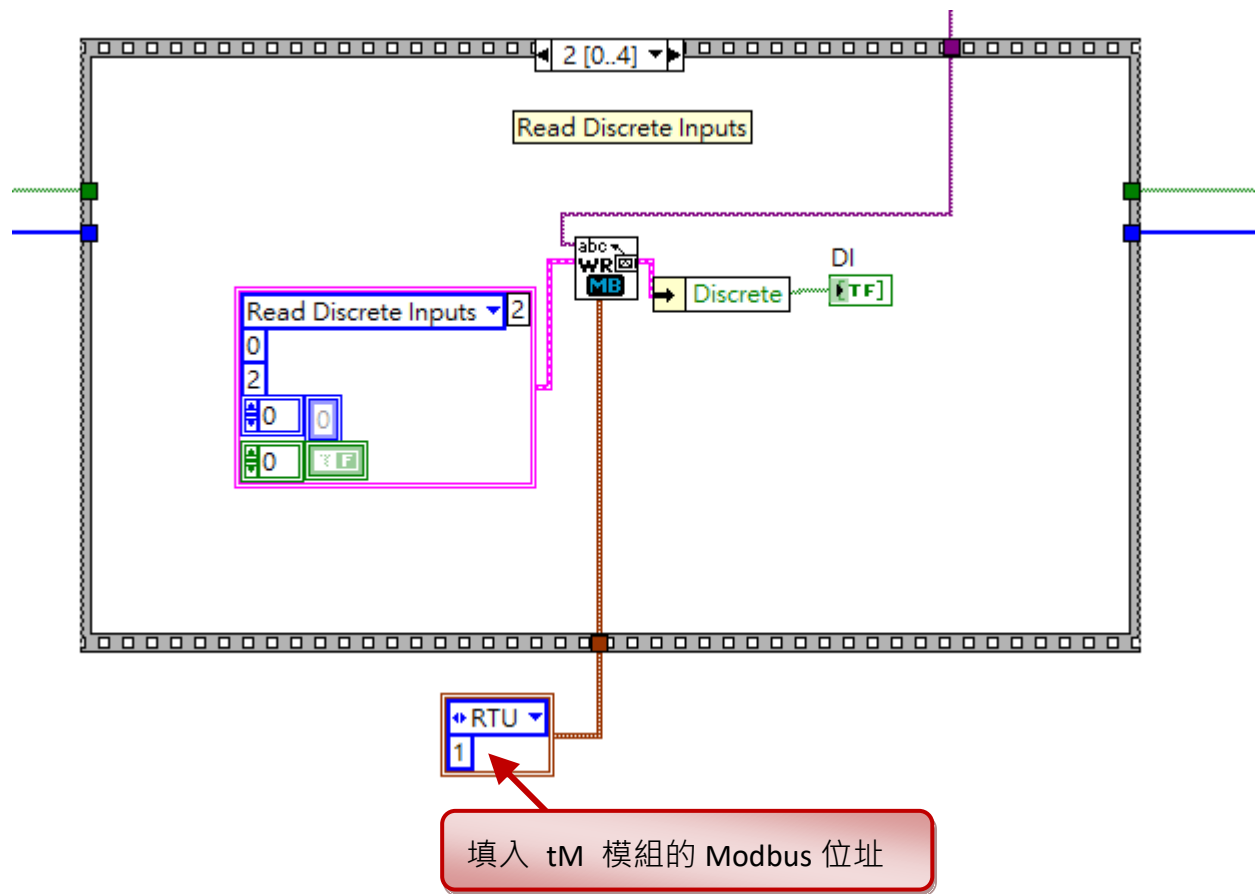
註： PC 與 tM 模組進行通訊前，請參考第三章：軟體工具 - DCON Utility Pro 的說明，設定模組組態並確認通訊埠號、模組的 Modbus 位址 與 Baud Rate 設定值。

步驟 1: 下載 ModbusRTU.zip 並解壓縮。

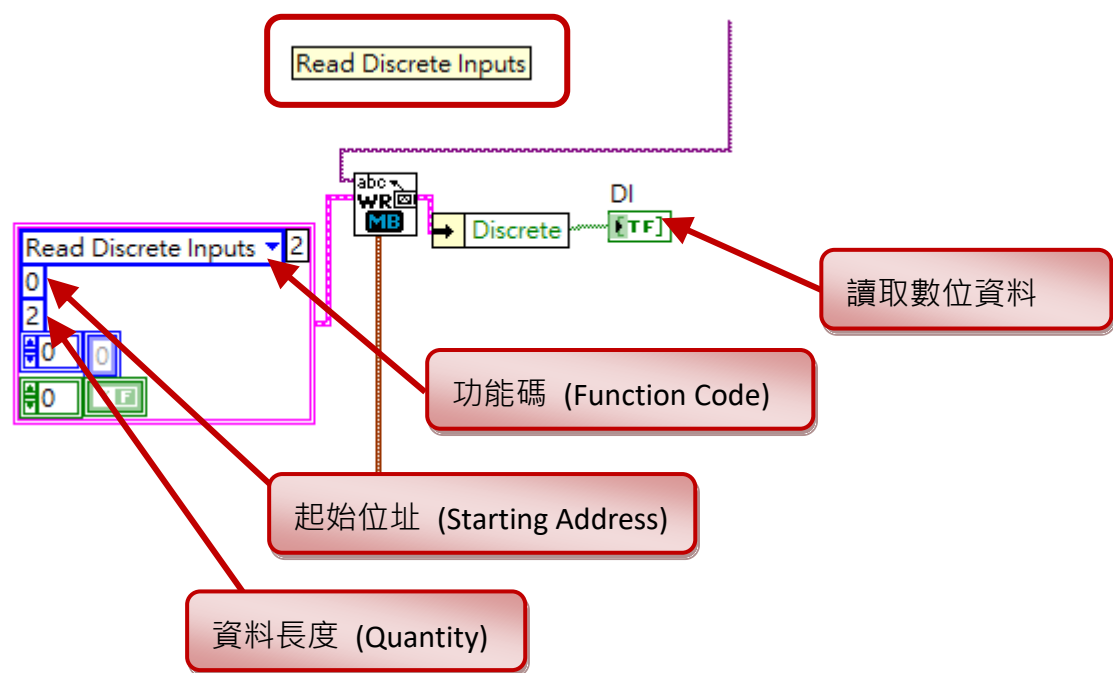
步驟 2: 選擇適合模組 I/O 類型的範例程式 (例如: MB_Serial_Example_ai_ao_di_do.vi)，並填入正確的 COM Port 編號 與 Baud Rate。



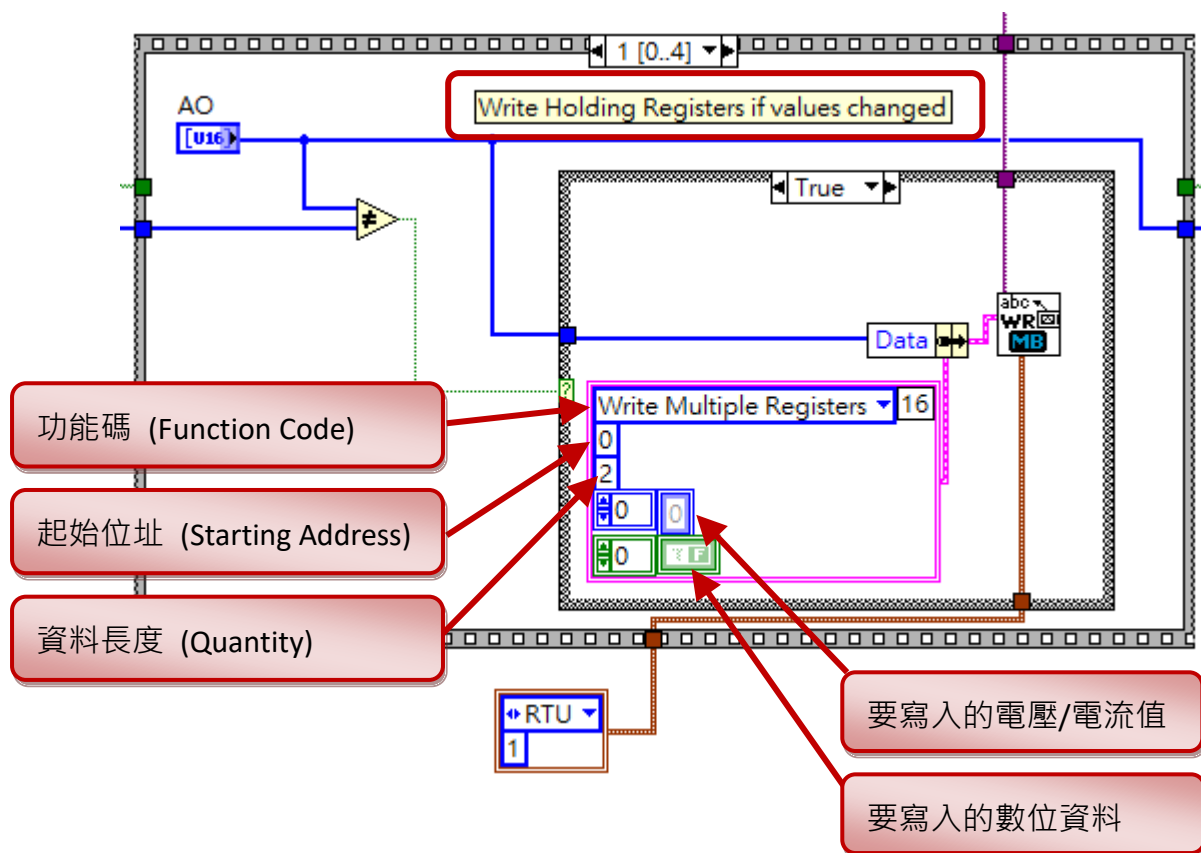
步驟 3: 填入模組的 Modbus 位址。



步驟 4: 欲讀取模組資料，需設定功能碼 (Function Code)、起始位址 (Starting Address) 與資料長度。



步驟 5: 欲寫入模組資料, 需設定功能碼 (Function Code)、起始位址 (Starting Address)、資料長度與資料內容。



Modbus Function Code 說明

功能碼	說明	I/O 位址
01	Read Coils (讀取數位輸出狀態)	0xxxx
02	Read Discrete Inputs (讀取數位輸入狀態)	1xxxx
03	Read Holding Registers (讀取類比輸出資料)	4xxxx
04	Read Input Registers (讀取類比輸入資料)	3xxxx
05	Write Single Coil (寫入單通道數位輸出)	0xxxx
06	Write Single Register (寫入單通道類比輸出)	4xxxx
15	Write Multiple Coils (寫入多通道數位輸出)	0xxxx
16	Write Multiple Registers (寫入多通道類比輸出)	4xxxx

6.3.4 圖控軟體 – InduSoft

InduSoft Web Studio 是一套功能強大的自動化整合開發工具，包含所有開發全方位 SCADA (Supervisory Control And Data Acquisition)、HMI (Human-Machine Interface)、嵌入式控制系統所需的基本組件。InduSoft Web Studio 支援 Windows 作業系統，含括 Windows 10 / 8 / 7 (包含 32-bit 與 64-bit)、Windows XP / Vista 等，還有 Windows Server 版本，且內建支援本機 (Local) 與遠端 (Web) 視覺化瀏覽功能。除此之外，InduSoft 亦遵照工業標準提供 Microsoft .NET、OPC、DDE、ODBC、XML 及 ActiveX 等使用介面。

注意:

泓格科技正式發佈 InduSoft 硬體鎖 (Hardkey) 全面以 v8.1 版授權，請參訪以下網頁了解新增的特色。

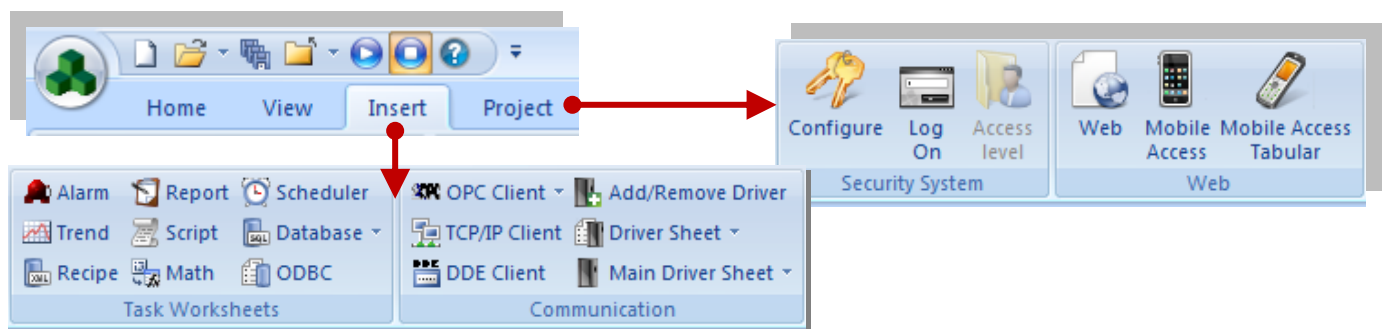


InduSoft 網頁:

<https://www.icpdas.com/en/product/guide+Software+InduSoft+InduSoft>

軟體特色:

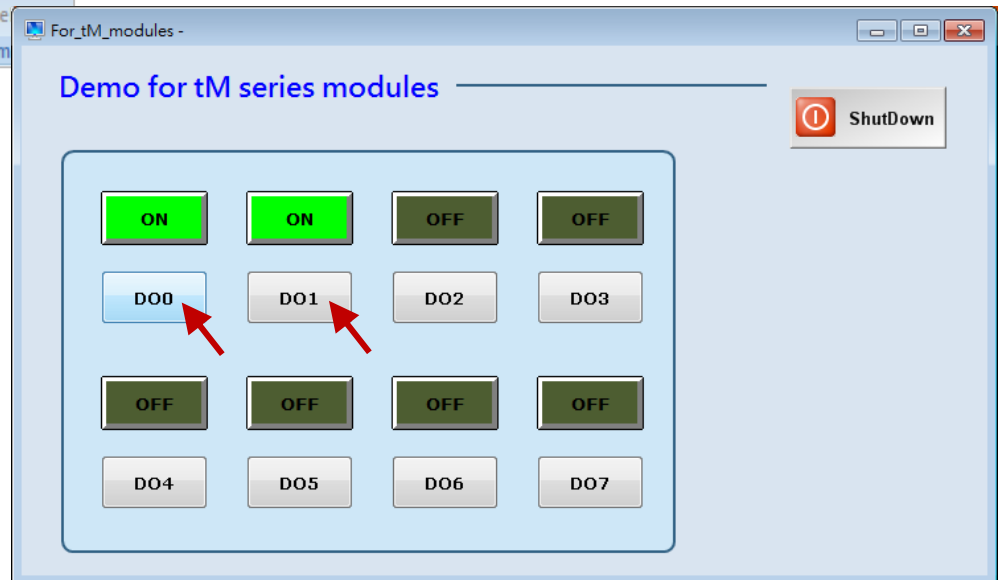
- ✧ 支援圖形和動畫設計工具。
- ✧ 連接任何 SQL 資料庫(MS SQL、MySQL、Sybase、Oracle)、MS Access、Excel 或 ERP/MES 系統。
- ✧ 警報、事件、趨勢圖、配方和報表管理工具。
- ✧ 提供超過 240 種通訊驅動程式。
- ✧ 支援網頁伺服器、資料庫和全面系統備援。
- ✧ 支援 InduSoft 內建函數 及 標準 VBScript 兩種功能強大的語言腳本。
- ✧ 於遠端使用 IE 瀏覽器 或 InduSoft Secure Viewer，以網頁方式瀏覽監控畫面。
- ✧ 支援多重開發者比對與合併變更的設定檔。
- ✧ 在任何支援 HTML5 瀏覽器上 (例如：iOS Safari、Google Chrome 等)，監看/變動系統數值。



範例說明:



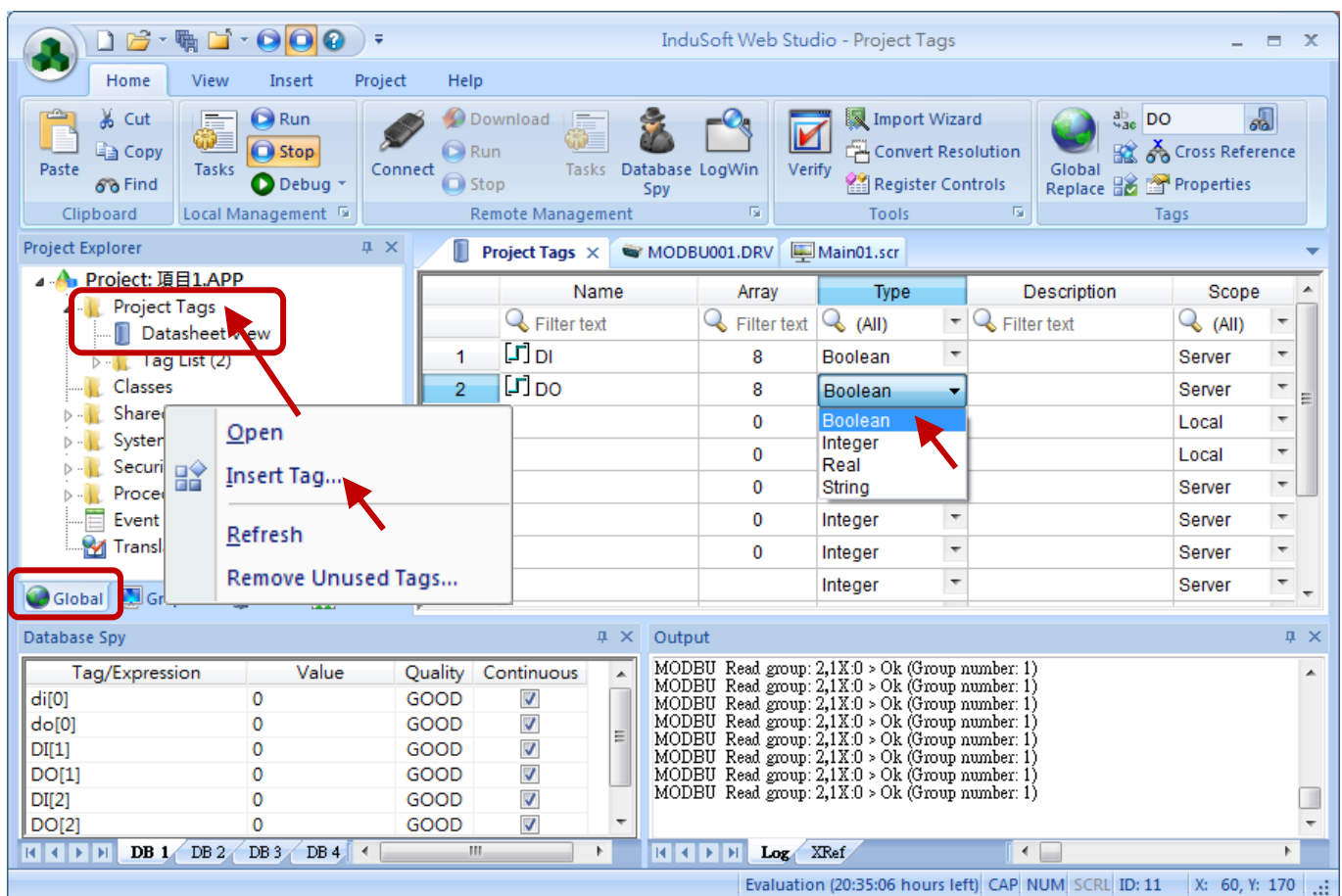
此範例是透過 Modbus RTU 的方式，來讀取/寫入 tM-P8 (8 DI) 與 tM-C8 (8 DO) 模組，您可參考 InduSoft 手冊，了解詳細的設定方式。於 "Home" 功能表，點選 Run (F5) 可執行專案。



設定步驟:

1. 建立變數。

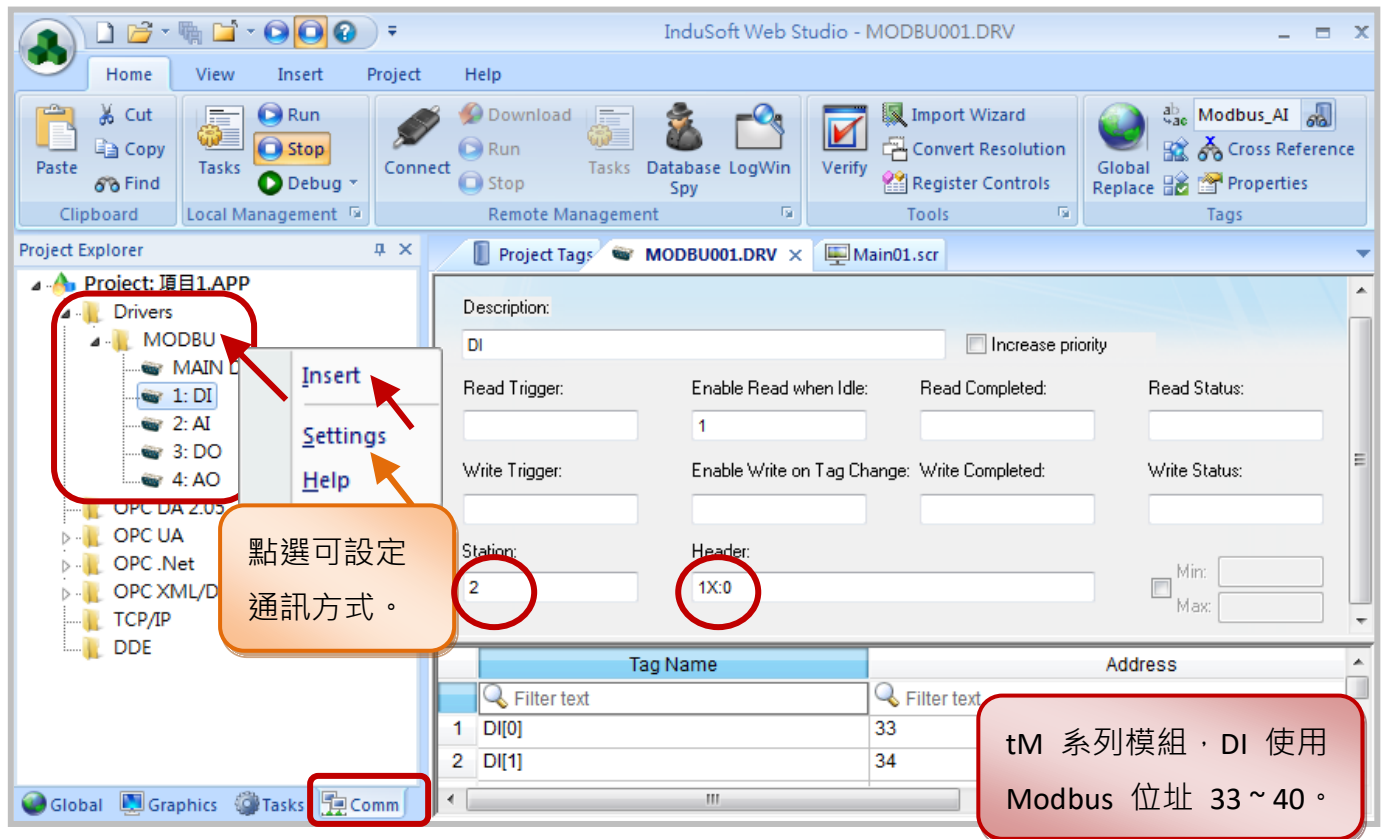
在 "Global" 頁籤，滑鼠右鍵點選 "Project Tags" 再點選 "Insert Tag" 建立變數。



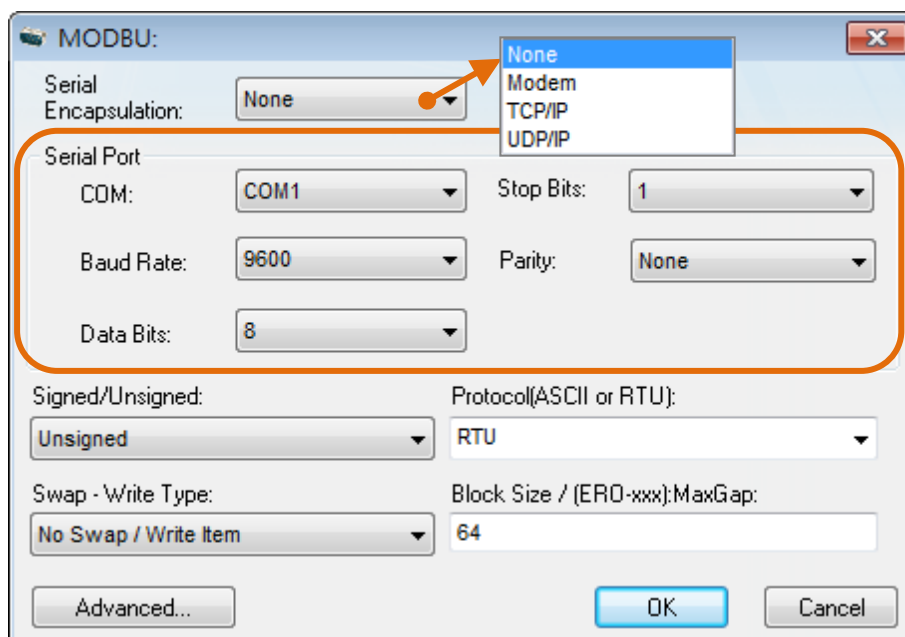
2. 設定變數 與 通訊方式。

在 "Comm" 頁籤，滑鼠右鍵點選 "MODBU" 再點選 "Insert" 設定變數。此例，tM-P8 (DI) 的 Net-ID=2、Header = 1x:0，而 tM-C8 (DO) 的 Net-ID=3、Header = 0x:0。

註：使用 tM 模組前，請使用 DCON Utility Pro 設定好通訊參數 (此例，Protocol = Modbus RTU、Address = 2 (tM-P8) 或 3 (tM-C8)、Baud Rate = 9600、Parity = N,8,1 與 Checksum = Disable)。

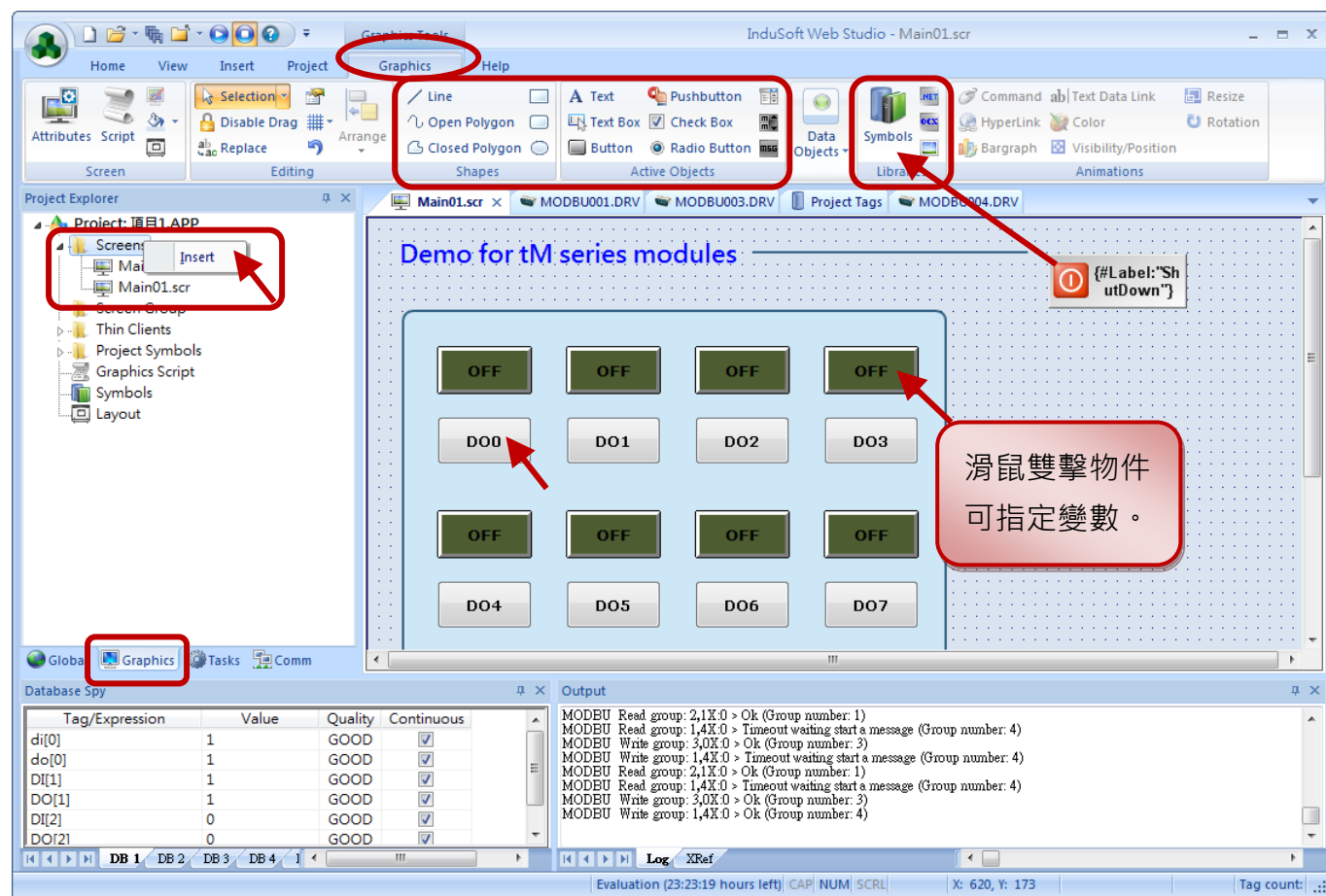


若點選 "Settings" 可設定 InduSoft 與 tM 模組通訊的方式，此例使用 COM1。

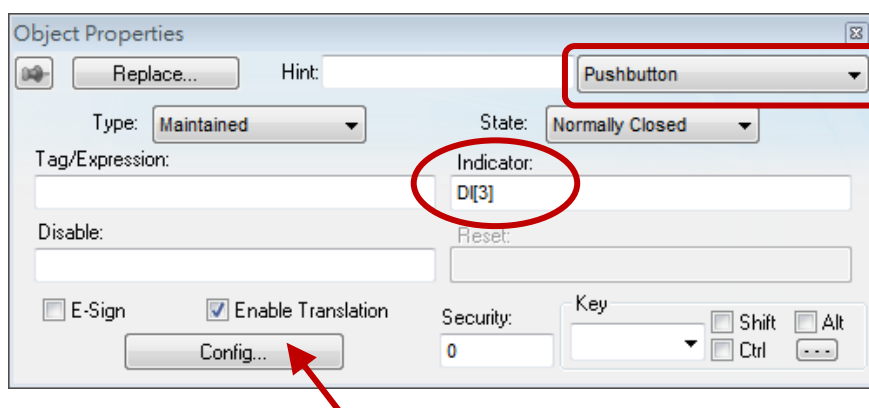


3. 設計畫面 與 指定變數。

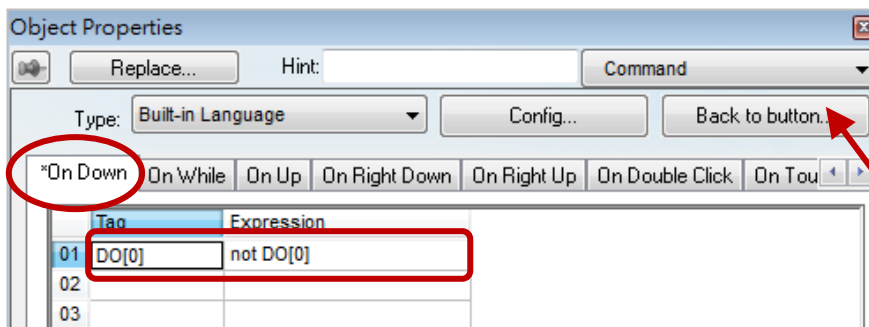
在 "Graphics" 頁籤，滑鼠右鍵點選 "Screen" 再點選 "Insert" 來新增畫面，並使用 "Graphics" 功能表的物件來設計畫面，滑鼠雙擊物件可指定變數。



Pushbutton – DI



Button – DO



6.4 Linux Modbus 開發工具

泓格科技 LinPAC 系列產品採用具有高彈性、開放性特色的 Linux 作業系統，您可以使用 DCON 或 Modbus 通訊協定來控制 tM 系列模組。本章節將以 Modbus 通訊協定為例，介紹三種軟體工具應用，提供您多元化的開發。

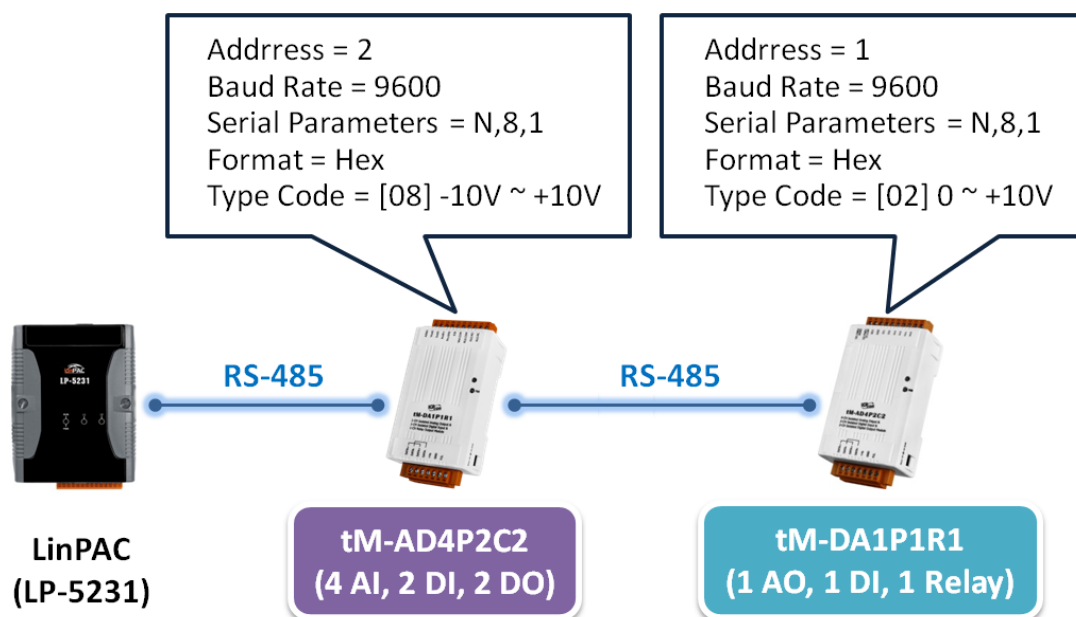
6.4.1 LinPAC 與 Modbus 應用

LinPAC 系列產品支援 Modbus 與 DCON 通訊協定，用戶可參訪網頁資訊下載 LinPAC SDK。
https://www.icpdas.com/tw/product/guide+Software+Development__Tools+Modbus__Tool#2845

下表為 AM335x SDK 的 Modbus 測試工具檔案路徑:

作業系統	SDK 路徑
Windows	C:\cygwin\LinPAC_AM335x_SDK\examples\xvboard 或 C:\cygwin\LinPAC_AM335x_SDK\examples\modbus
Linux	root@LinuxPC-ICPDAS:/icpdas/linpac_am335x_sdk/i8k/examples/xvboard 或 root@LinuxPC-ICPDAS:/icpdas/linpac_am335x_sdk/i8k/examples/modbus

本例使用 LP-5231 控制器搭配 tM-DA1P1R1 與 tM-AD4P2C2 模組來做測試，以下為模組的設置參數：



用戶必須設置從站設備的設備站號 (Net ID) ，其設置步驟如下：

```
root@LP-5231:~# getmodbus 2 9600 1 4 484 1 100
1
root@LP-5231:~# setmodbus 2 9600 1 16 484 1 2 100
root@LP-5231:~# getmodbus 2 9600 2 4 484 1 100
2
```

步驟 1： 使用以下指令**查詢** tM-AD4P2C2 模組的設備站號 (Net ID) 。

指令：

getModbus <comport> <baudrate> <NetID> <command> <address> <count> <timeout(ms)>

```
# getModbus 2 9600 1 4 484 1 100
```

步驟 2： 使用以下指令**修改** tM-AD4P2C2 模組的設備站號 (Net ID) 。

指令：

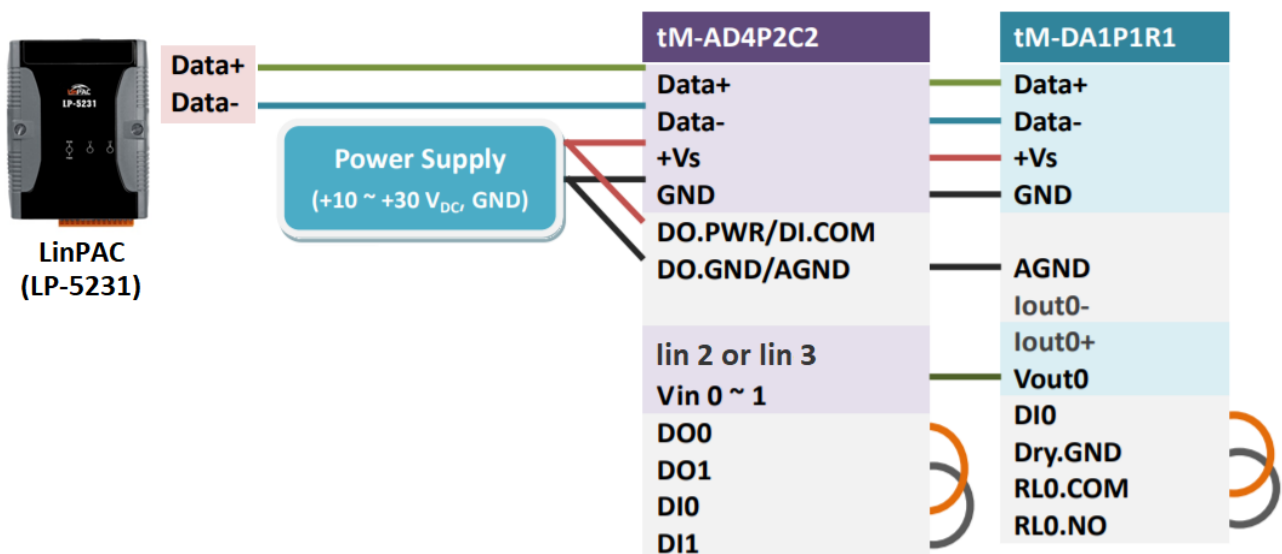
setModbus <comport> <baudrate> <NetID> <command> <address> <count> <value> <timeout(ms)>

```
# setModbus 2 9600 1 16 484 1 2 100 // Set the NetID as 2
```

注意：

1. 用戶可參考 附錄 C - Modbus 位址對應 (Base 1) ，來設置設備位址 (Net ID) 。
2. tM 系列模組的基底位址 (base address) 是由 0 開始 (Base 0) 。例如: Modbus 位址 400485 是用來讀取/寫入模組位址 (即，Net ID) ，此例，需使用位址 484 來取得/設定 Net ID 。

硬體接線 (測試用)：



□ 驗證 AI/AO 功能

連接 tM-DA1P1R1 模組的 AO 通道與 tM-AD4P2C2 模組的 AI 通道，使用範例程式 setModbus.c 與 getModbus.c 測試 AI/AO 功能。測試步驟如下：

```
root@LP-5231:~# setmodbus 2 9600 1 16 32 1 65535 100
root@LP-5231:~# getmodbus 2 9600 1 3 32 1 100
65535
root@LP-5231:~# getmodbus 2 9600 2 4 0 1 100
32767
```

步驟 1：使用以下指令，設定 tM-DA1P1R1 模組的 AO 值。

```
# setModbus 2 9600 1 16 32 1 65535 100 //輸出 10V 電壓
```

步驟 2：使用以下指令，讀回 tM-DA1P1R1 模組的 AO 值。

```
# getModbus 2 9600 1 3 32 1 100
```

步驟 3：使用以下指令，讀取 tM-AD4P2C2 模組的 AI 值。

```
# getModbus 2 9600 2 4 0 1 100
```

□ 驗證 DI/DO 功能

連接 tM-DA1P1R1 模組的 DO 通道與 DI 通道，使用範例程式 setModbus.c 與 getModbus.c 測試 DI/DO 功能。測試步驟如下：

```
root@LP-5231:~# setmodbus 2 9600 1 15 0 1 1 100
wCount=1 iCount=8 iIndex=0
root@LP-5231:~# getmodbus 2 9600 1 1 0 1 100
1
root@LP-5231:~# getmodbus 2 9600 1 2 32 1 100
1
```

步驟 1：使用以下指令，設定 tM-DA1P1R1 模組的 DO 狀態。

```
# setModbus 2 9600 1 15 0 1 1 100 //設置 DO 通道狀態為 ON
```

步驟 2：使用以下指令，讀回 tM-DA1P1R1 模組的 DO 狀態。

```
# getModbus 2 9600 1 1 0 1 100
```

步驟 3：使用以下指令，讀取 tM-DA1P1R1 模組的 DI 狀態。

```
# getModbus 2 9600 1 2 32 1 100
```


連接 tM-AD4P2C2 模組的 DO 通道與 DI 通道，使用範例程式 setModbus.c 與 getModbus.c 測試 DI/DO 功能。測試步驟如下：

```
root@LP-5231:~# setmodbus 2 9600 2 15 0 1 1 100
wCount=1 iCount=8 iIndex=0
root@LP-5231:~# getmodbus 2 9600 2 1 0 1 100
1
root@LP-5231:~# getmodbus 2 9600 2 2 32 1 100
1
```

步驟 1：使用以下指令，設定 tM-AD4P2C2 模組的 DO 狀態。

```
# setModbus 2 9600 2 15 0 1 1 100 //設置 DO 通道狀態為 ON
```

步驟 2：使用以下指令，讀回 tM-AD4P2C2 模組的 DO 狀態。

```
# getModbus 2 9600 2 1 0 1 100
```

步驟 3：使用以下指令，讀取 tM-AD4P2C2 模組的 DI 狀態。

```
# getModbus 2 9600 2 2 32 1 100
```


6.4.2 LinPAC 與 Python 應用

LinPAC 系列產品支援 Python 語言的開發環境，用戶可在 LinPAC 中安裝 Python 官網提供的 Modbus 工具，來測試 tM 系列模組。本例使用 LP-5231 控制器搭配 tM-DA1P1R1 模組，應用 modbus-tk 工具來存取資料，關於 modbus-tk 軟體的更多資訊，請參考以下連結：

<https://github.com/ljean/modbus-tk>，其安裝與測試步驟如下：

步驟 1：使用以下指令，檢查 Python 的版本號，並確認 Python 為 2.5 以上版本。

```
# python --version
```

```
root@LP-5231:~# python --version
Python 2.7.3
root@LP-5231:~#
```

步驟 2：使用以下指令，安裝 pyserial 模組。

```
# pip install pyserial
```

步驟 3：使用以下指令，下載 modbus-tk 套件。

```
# wget https://github.com/ljean/modbus-tk/archive/master.zip
```

步驟 4：使用以下指令，解壓縮 modbus-tk 套件。

```
# unzip master.zip
```

步驟 5：使用以下指令，安裝 modbus-tk 工具。

```
# cd modbus-tk-master
# python setup.py build
# python setup.py install
```

步驟 6：使用以下指令，檢查 pyserial 與 modbus-tk 套件是否安裝成功。

```
# pip list
```

```
root@LP-5231:~/modbus-tk-master# pip list
Package      Version
-----
distribute   0.6.24dev-r0
modbus-tk    0.5.8
pip          18.1
pyserial     3.4
setuptools   0.6rc11
```

步驟 7：於 modbus-tk 套件所提供的範例中，找到範例程式 rtumaster_example.py。

```
root@LP-5231:~# cd modbus-tk-master/examples/
root@LP-5231:~/modbus-tk-master/examples# ls
modbus_system_monitor.py  rtumaster_example.py  tcpmaster_example.py
mysimu.py                rtuslave_example.py  tcpslave_example.py
root@LP-5231:~/modbus-tk-master/examples#
```

步驟 8：修改範例程式 rtumaster_example.py 的參數設定。

```
import modbus_tk
import modbus_tk.defines as cst
from modbus_tk import modbus_rtu
#PORT = 1
PORT = '/dev/ttyO2' 通訊埠
def main():
    """main"""
    logger = modbus_tk.utils.create_logger("console")
    try:
        #Connect to the slave
        master = modbus_rtu.RtuMaster(
            serial.Serial(port=PORT, baudrate=9600, bytesize=8, parity='N', stopbits=1, xonxoff=0)
        )
        master.set_timeout(5.0)
        master.set_verbose(True)
        logger.info("connected")
        logger.info(master.execute(1, cst.READ_HOLDING_REGISTERS, 32, 1))
        #
        #
        #
        #
    except Exception as e:
        logger.error(e)
```

通訊埠參數設定

Modbus 命令
master.execute(NetID, function code, address, count)

注意：在 LinPAC 系列設置 Modbus 時，Modbus 位址的起始位置由 0 開始。

步驟 9：執行範例程式，來讀取 tM-DA1P1R1 模組的 AO 值，下圖為執行結果。

```
root@LP-5231:~/modbus-tk-master/examples# python rtumaster_example.py
2018-12-10 17:47:25,575 INFO modbus_rtu.__init__ MainThread RtuMaster /dev/ttyO2 is opened
2018-12-10 17:47:25,578 INFO rtumaster_example.main MainThread connected
2018-12-10 17:47:25,580 DEBUG modbus.execute MainThread -> 1-3-0-32-0-1-133-192
2018-12-10 17:47:25,606 DEBUG modbus.execute MainThread <- 1-3-2-255-255-185-244
2018-12-10 17:47:25,607 INFO rtumaster_example.main MainThread (65535,)
```

6.4.3 LinPAC 與 Perl 應用

LinPAC 系列產品支援 Perl 語言的開發環境，用戶可在 LinPAC 中安裝 Perl 官網提供的 Modbus 工具，來測試 tM 系列模組。本例使用 LP-5231 控制器搭配 tM-DA1P1R1 模組，應用 Device-Modbus-RTU 工具來存取資料，其安裝與測試步驟如下：

步驟 1：下載並解壓縮 Device-Modbus-RTU 套件（檔名: Device-Modbus-RTU-0.022.tar.gz）。

網址: <https://metacpan.org/release/Device-Modbus-RTU>

步驟 2：使用以下指令，安裝 Device-Modbus-RTU 的相依模組。

```
# sudo cpan Role::Tiny Try::Tiny Device::SerialPort Device::Modbus
```

步驟 3：使用以下指令，安裝 Device-Modbus-RTU 軟體。

```
# cd Device-Modbus-RTU-0.022
# perl Makefile.PL
# make
# make test
# make install
```

步驟 4：使用 'instModsh' 指令，檢查 perl 模組是否安裝成功。

```
root@LP-5231:~# instmodsh
Available commands are:
  l                - List all installed modules
  m <module>       - Select a module
  q                - Quit the program
cmd? l
```

```
Installed modules are:
Device::Modbus
Device::Modbus::RTU
Device::SerialPort
Perl
Role::Tiny
Try::Tiny
cmd? q
```

步驟 5：於 Device-Modbus-RTU 套件所提供的範例中，找到範例程式 write_new_addr.pl 與 simple_client_rtu.pl。

```
root@LP-5231:~# cd Device-Modbus-RTU-master/examples/
root@LP-5231:~/Device-Modbus-RTU-master/examples# ls
arduino_client.ino  server_rtu.pl  simple_client_rtu.pl  write_new_addr.pl
```

步驟 6：修改範例程式的參數設定。

❑ 修改 write_new_addr.pl 與 simple_client_rtu.pl 文本中的連接埠設定。

```
my $client = Device::Modbus::RTU::Client->new(
    port      => '/dev/ttyO2',    // 通訊埠
    baudrate => 9600,             // 每秒傳輸位元
    parity    => 'none',          // 同位檢查
);
```

❑ 修改 write_new_addr.pl 的 Modbus 命令設定，並設置 AO 通道的值。

```
my $req = $client->write_single_register(    // Modbus 功能碼
    unit      => 1,                  // 從站設備的站號 (NetID)
    address   => 32,                  // 通道位址
    value     => 65535               // 設定值
);
```

❑ 修改 simple_client_rtu.pl 的 Modbus 命令設定，並讀取 AO 通道的值。

```
my $req = $client->read_holding_registers(    // Modbus 功能碼
    unit      => 1,                  // 從站設備的站號 (NetID)
    address   => 32,                  // 通道位址
    quantity  => 1,                  // 所要操作的暫存器長度
);
```

範例程式 write_new_addr.pl

```
#!/usr/bin/env perl
use Device::Modbus;
use Device::Modbus::RTU::Client;
use Data::Dumper;
use strict;
use warnings;
use v5.10;

my $client = Device::Modbus::RTU::Client->new(
    port      => '/dev/ttyO2',
    baudrate  => 9600,
    parity    => 'none',
);

my $req = $client->write_single_register(
    unit      => 1,
    address   => 32,
    value     => 65535
);

say "->" . Dumper $req;
$client->send_request($req);
my $resp = $client->receive_response;
say "<-" . Dumper $resp;
```

通訊埠參數設定

Modbus 命令設定

注意： 在 LinPAC 系列設置 Modbus 時，Modbus 位址的起始位置由 0 開始。

步驟 7：執行範例程式來控制 tM 系列模組。

(1) 執行 write_new_addr.pl 的結果:

```
root@LP-5231:~/Device-Modbus-RTU-master/examples# perl write_new_addr.pl
->$VAR1 = bless( {
    'unit' => 1,
    'function' => 'Write Single Register',
    'value' => 65535,
    'address' => 32,
    'code' => 6
  }, 'Device::Modbus::Request' );
<-$VAR1 = bless( {
    'unit' => 1,
    'crc' => 45193,
    'message' => bless( {
        'function' => 'Write Single Register',
        'value' => 65535,
        'address' => 32,
        'code' => 6
      }, 'Device::Modbus::Response' )
  }, 'Device::Modbus::RTU::ADU' );
```

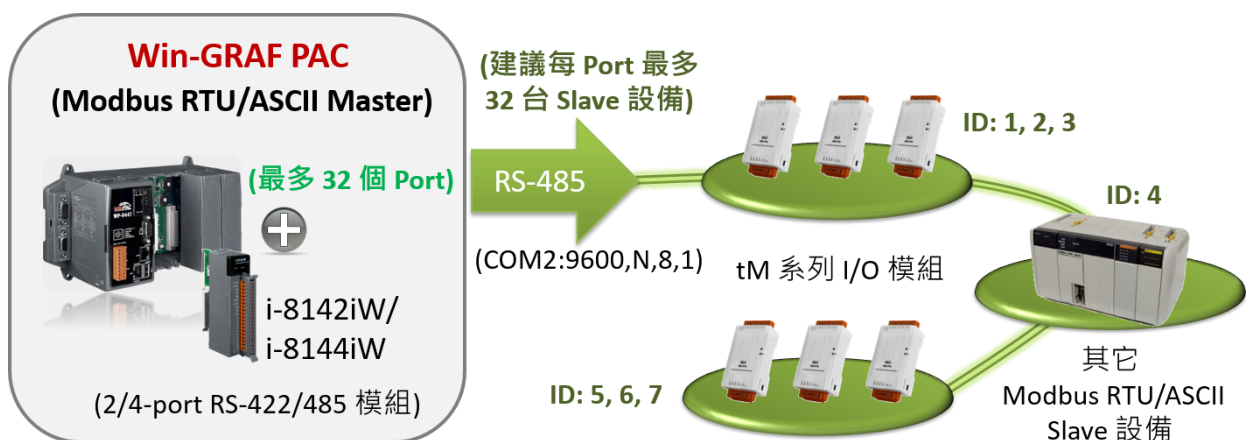
(2) 執行 simple_client_rtu.pl 的結果:

```
root@LP-5231:~/Device-Modbus-RTU-master/examples# perl simple_client_rtu.pl
->$VAR1 = bless( {
    'unit' => 1,
    'function' => 'Read Holding Registers',
    'quantity' => 1,
    'address' => 32,
    'code' => 3
  }, 'Device::Modbus::Request' );
<-$VAR1 = bless( {
    'unit' => 1,
    'crc' => 62649,
    'message' => bless( {
        'bytes' => 2,
        'function' => 'Read Holding Registers',
        'values' => [
            65535
        ],
        'code' => 3
      }, 'Device::Modbus::Response' )
  }, 'Device::Modbus::RTU::ADU' );
```

6.5 使用 Win-GRAF 開發軟體

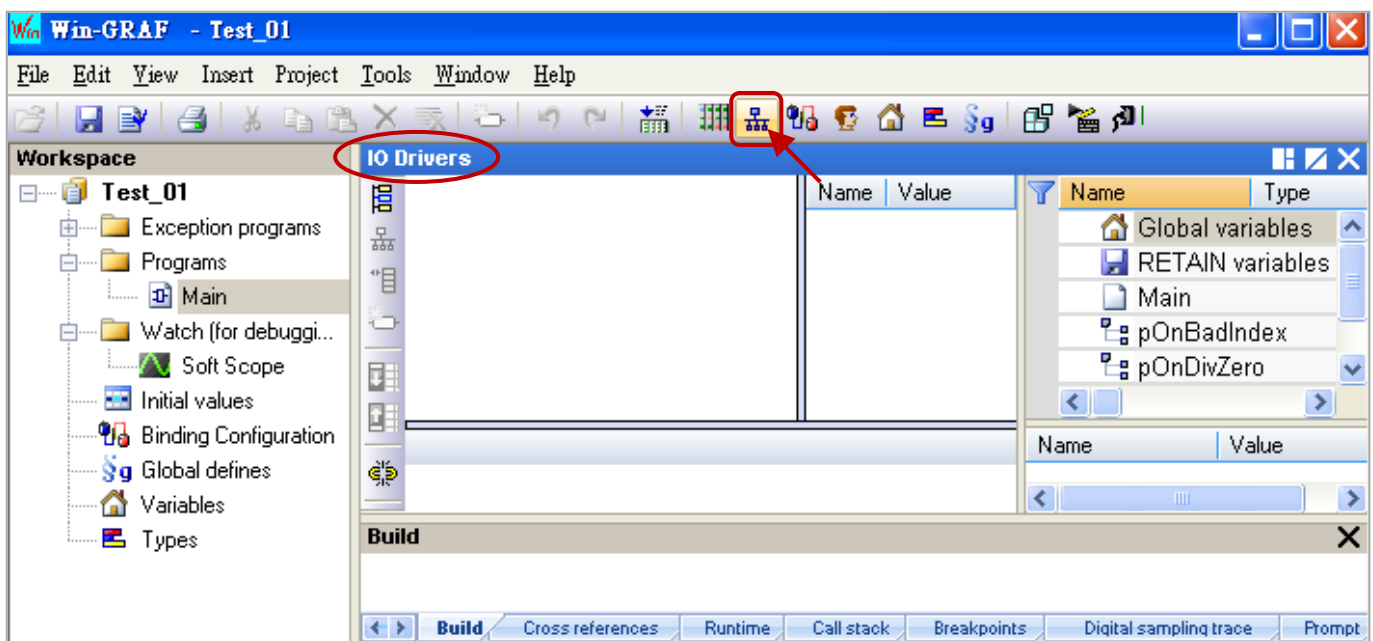
Win-GRAF 為一套 PLC-like 軟邏輯開發軟體，符合 IEC 61131-3 標準 PLC 語法，可安裝於 Windows 7/8 (以上) 電腦上。使用 Win-GRAF 開發的控制程式，可下載到 Win-GRAF PAC 內去執行，例如: WinPAC 系列 (WP-5238-CE7、WP-8xx8、WP-8x28-CE7)，或是帶有觸控螢幕的 ViewPAC 系列 (VP-x208-CE7 與 VP-x238-CE7) 或 CPU 等級較高階的 XPAC 系列 (XP-8x38-CE6)，業者可輕鬆快速的在各種應用領域中，開發出符合工業等級的監控系統，並進行資料採集 與 監控設備資訊。
網頁: https://www.icpdas.com/en/product/guide+Software+Development__Tools+Win-GRAF

Win-GRAF PAC 可支援 Modbus RTU/ASCII/TCP/UDP Master/Slave 與 DCON 通訊協定，以下範例將介紹如何啟用 Win-GRAF PAC 為 Modbus RTU/ASCII Master 來連接 tM 系列模組：

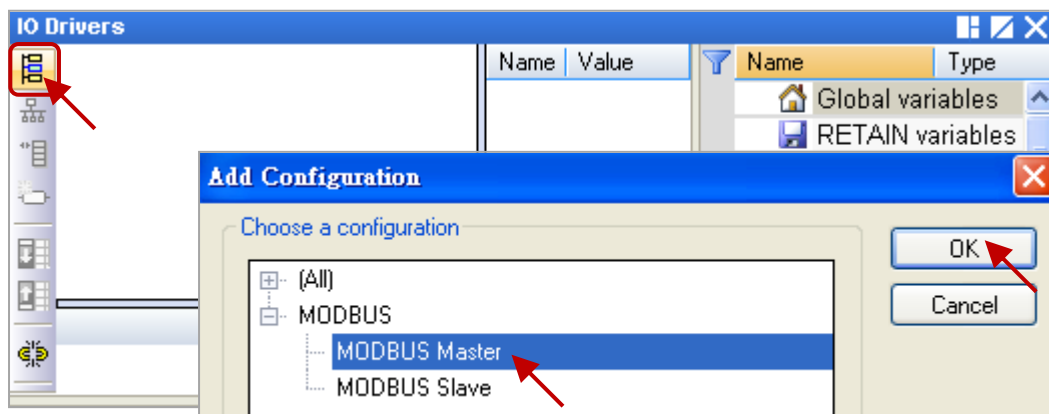


Win-GRAF 操作步驟:

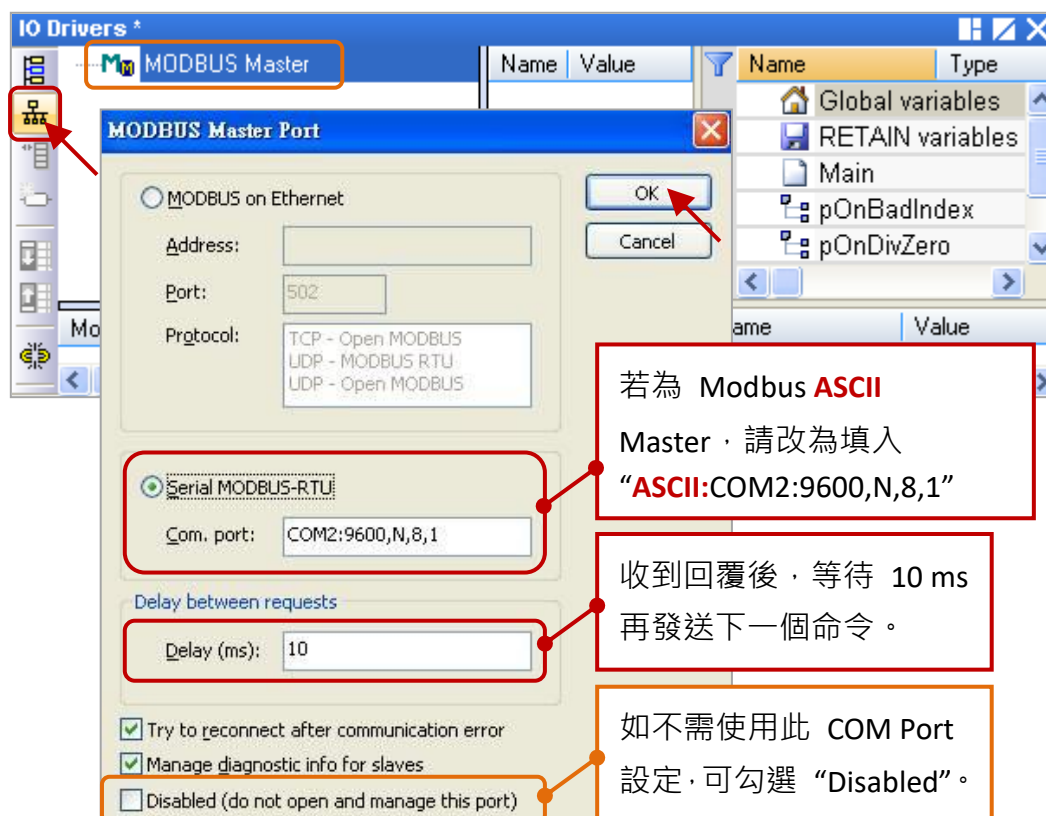
1. 滑鼠點選工具列上的“Open Fieldbus Configuration”按鈕來開啟“I/O Drivers”視窗。



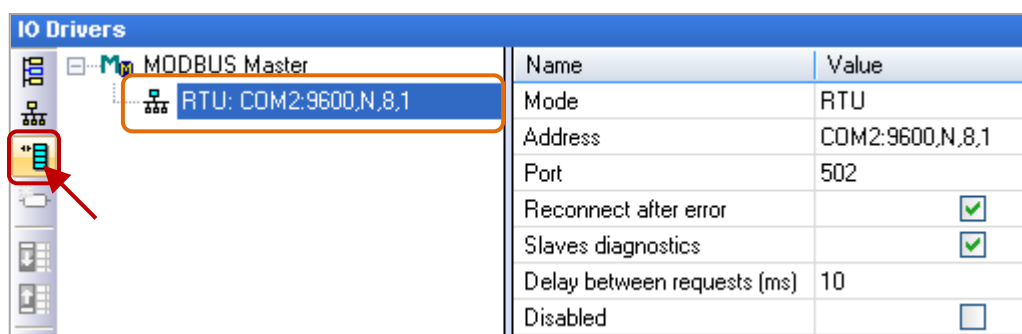
- 點選 “I/O Drivers” 視窗左側的 “Insert Configuration” 按鈕，再點選 “MODBUS Master” 並點選 “OK” 來啟用一個 Modbus Master 設定。



- 點選左側的 “Insert Master/Port” 按鈕，開啟設定視窗。點選 “Serial MODBUS-RTU” 並設定 COM Port (例如: “COM2:9600,N,8,1”) 與 Delay (建議值: 10 ms，可設為 0 ~ 10000)，再點選 “OK”。



- 點選左側的 “Insert Slave/Data Block” 按鈕，來建立一個 Data Block。



5. 每一個 Data Block 都代表一個 Modbus Master Request，可依應用需求來設定。於“MODBUS Master Request”設定視窗中，完成以下設定並按“OK”。

The image shows the 'MODBUS Master Request' configuration window. It has several sections: 'Request' with 'Description' and 'Slave/Unit' fields; 'MODBUS Request' with a list of request types; 'Data block' with 'Base address' and 'Nb items' fields; 'Activation' with radio buttons for 'Periodic', 'On call', and 'On change'; and 'Misc.' with 'Timeout' and 'Nb trials' fields. Annotations are as follows: 'a' points to the 'Slave/Unit' field with the value '1'; 'b' points to the '<2> Read Input Bits' option in the 'MODBUS Request' list; 'c' points to the 'Base address' field with the value '1'; 'd' points to the 'Periodic' radio button; 'e' points to the 'Timeout' field with the value '250'. A red box with text '此選項不適用在“讀取”指令。' (This option is not applicable for 'read' commands) points to the 'On change' radio button.

a. Slave/Unit:

填入 Slave 設備的站號 (Net-ID，此例為“1”)。

b. MODBUS Request:

此例，用來讀取 DI 資料，選擇“<2> Read Input Bits”選項。

c. Base address:

預設值由 1 開始。

Nb items:

讀取 DI 的數量 (此例為 8)。

d. Periodic:

表示週期性的發送，此例為每 2 秒發送一次。“on error”表示每當發生異常時，下一次的發送時間 (此例為 15 秒)。

e. Timeout: 設定多久未回應，即表示異常。

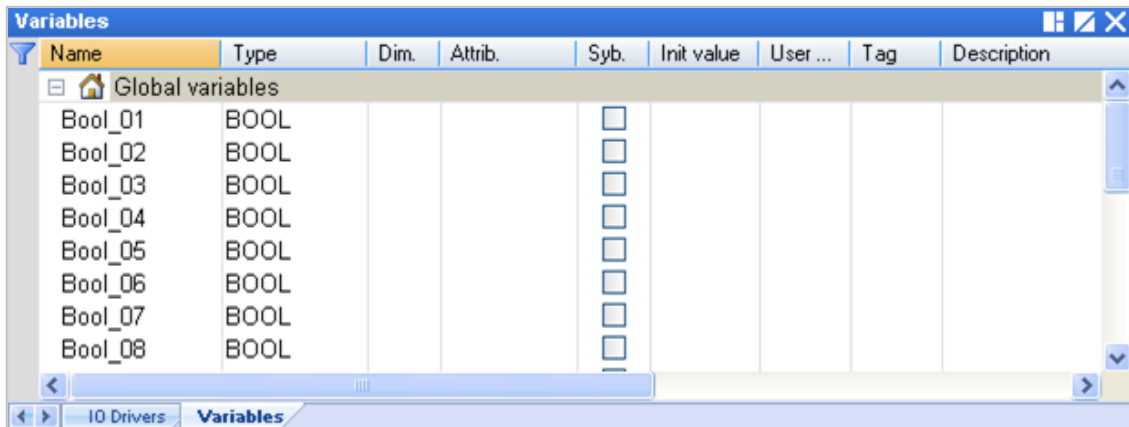
(對於 Modbus RTU/ASCII 建議值: 200 ~ 1000 ms；此例為 250 ms)

6. 於“IO Drivers”視窗，已建立一個讀取 DI 的 Data Block。請開啟“Variables”視窗，新增需使用的變數。

The image shows the 'IO Drivers' window. On the left, there is a tree view with 'Test_01' selected. Below it, a red box with text '滑鼠雙擊，開啟該視窗。' (Double-click the mouse to open this window) points to the 'Variables' folder. In the center, a table shows the configuration for a 'MODBUS Master' driver. A red box highlights the '<2> Read Input Bits [1] [1..16]' entry. Below this, a blue box with text '使用小技巧: 按“F1”鍵，可查看詳細的設定說明。(MODBUS Master Configuration)' (Useful tip: Press the 'F1' key to view detailed configuration instructions. (MODBUS Master Configuration)) points to the highlighted entry. On the right, a table shows the configuration details for the selected entry.

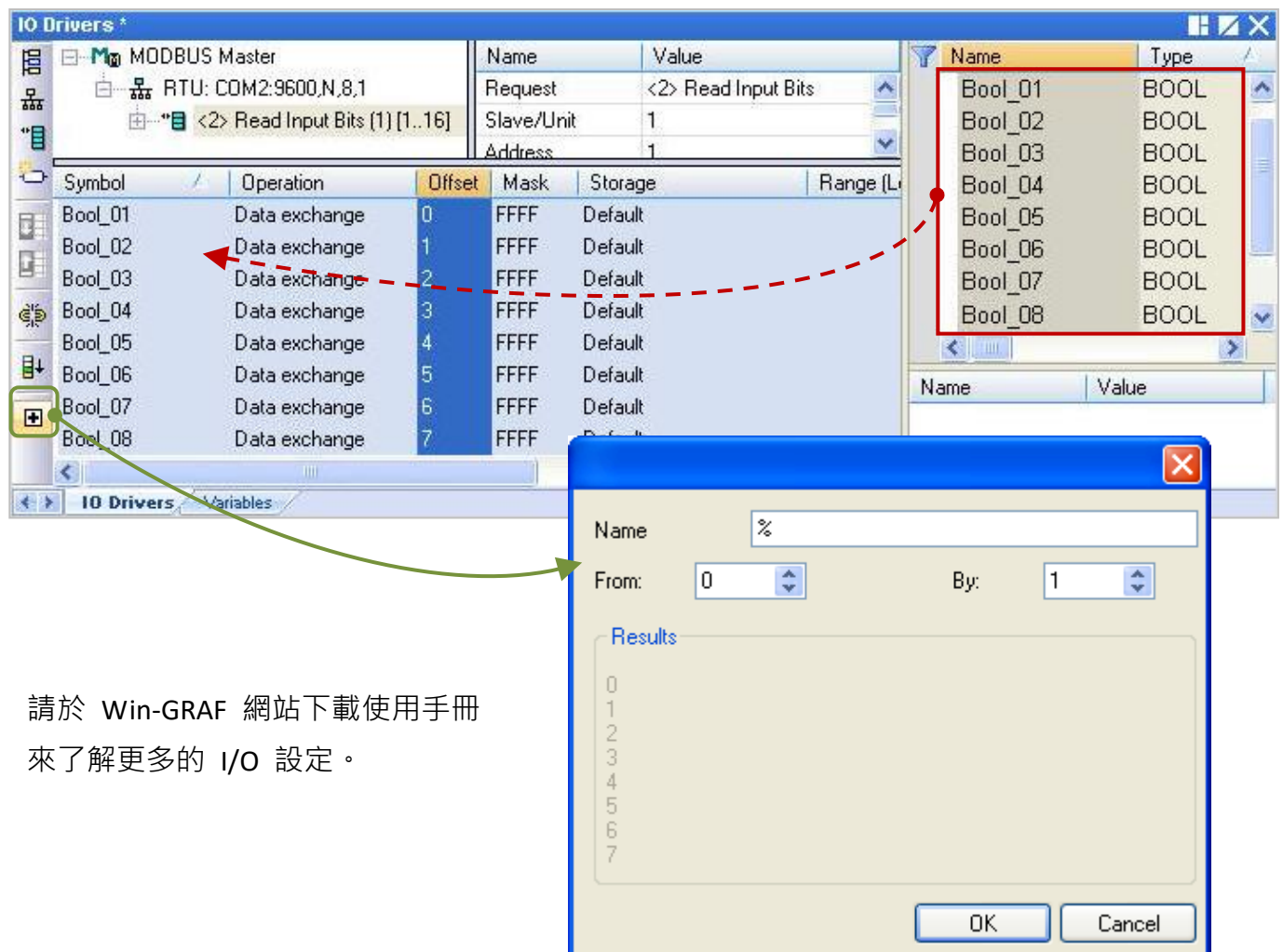
Name	Value
Request	<2> Read Input Bits
Slave/Unit	1
Address	1
Nb Item	16
Activation	Periodic
Period (ms)	2000
Period on error	15000
Timeout (ms)	250
Number of trials	1
Description	

此處新增了 8 個變數 (Name: “Boo_01 ~ Boo_08”; Type: BOOL) , 可用來讀取 8 個 DI 資料。



7. 請將變數區中的變數 (“Boo_01” ~ “Boo_08”) 拖曳到 Data Block 的 “Symbol” 區域。接著，選取所有的 “Offset” 欄位，並點選左側的 “Iterate Property” 按鈕，設定 “Offset” 值。

注意：“Offset” 的值是由 “0” 開始，而 “Offset” 的值加 1 (Base address) 才是該變數的 Modbus 位址。若選用的是 32-bit 或以上的資料型態 (例如: DINT)，則需佔用 2 個 Modbus 位址。

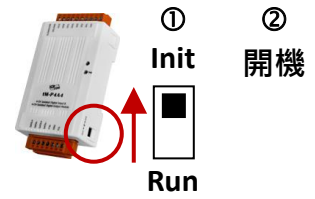


請於 Win-GRAF 網站下載使用手冊
來了解更多的 I/O 設定。

附錄 A 功能說明

A.1 INIT 模式

tM 系列模組內建有一個 EEPROM，可用來儲存模組位址、類型代碼 (Type Code) 與 通訊速率 (Baud Rate) 等等的設定資訊。有時，您可能忘了模組的通訊參數，並且沒有可查詢的資訊，無法與模組進行通訊。為解決此問題，tM 系列模組提供了一個好用的“INIT”模式。

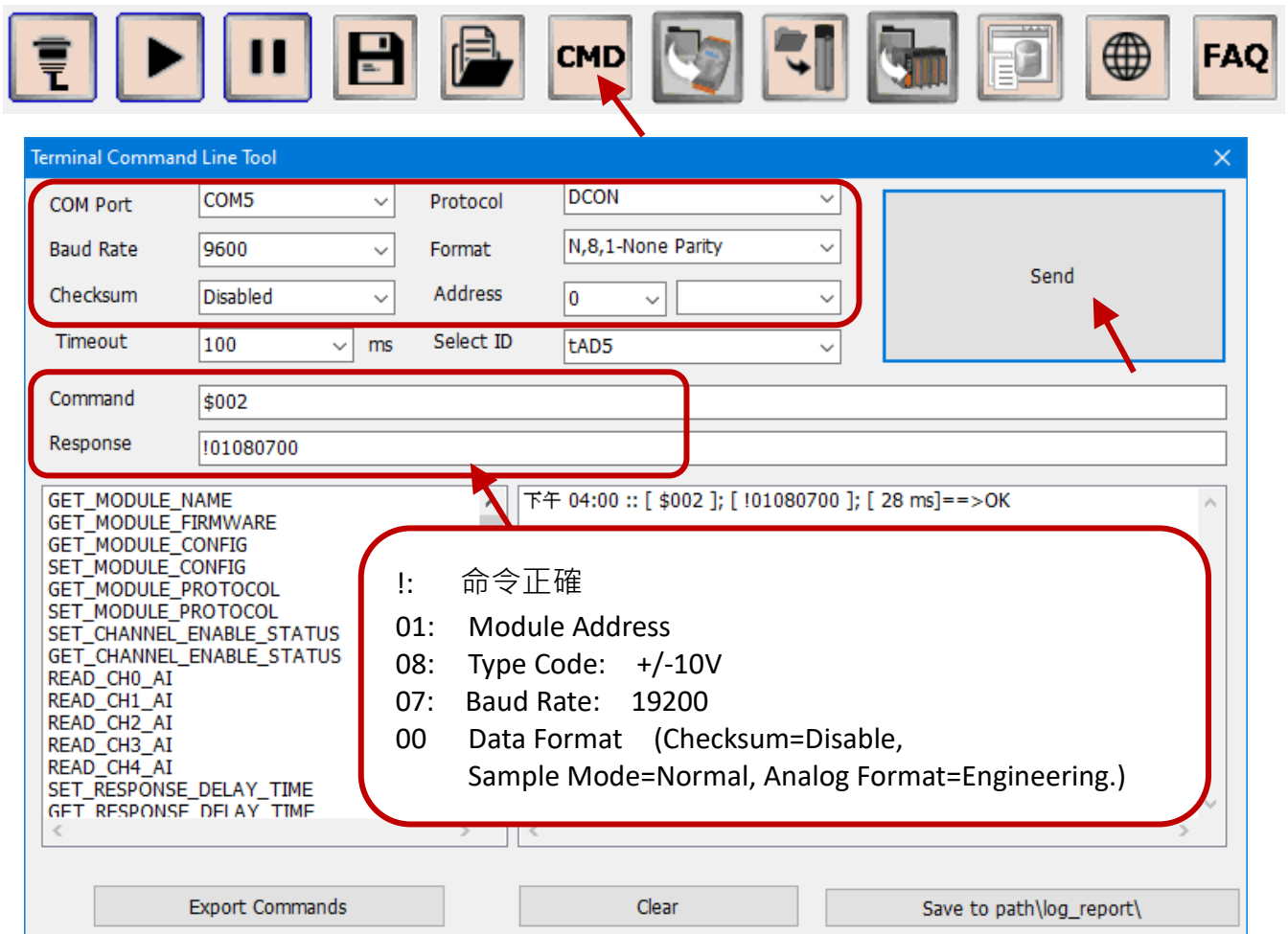


當您以“INIT”模式開機時，模組的通訊參數如下，如此可立即搜尋到模組。

1. 通訊協定 (Protocol): **DCON**
2. 位址 (Address): **00**
3. 通訊速率 (Baud Rate): **9600 bps**
4. 檢查碼 (Checksum): **Disable**

此時，您可在“INIT”模式下，使用 DCON Utility Pro 的“Command Line”功能，輸入 \$002 指令來讀取原先儲存在 EEPROM 內的模組設定值，並查看附錄 D 的 \$AA2 指令，來了解回應值。

註： DCON Utility Pro 會在指令後自動加上 (CR)，所以只需輸入 \$002。



A.2 雙看門狗功能 (Dual Watchdog)

無論是在智慧建築、重要醫療設備、高架道路的車流量監控、工廠生產流程等應用中，系統可能因為天然災害 (例如：雷擊)、惡劣環境 (例如：雜訊、電磁干擾) 或 種種不可預期的狀況，而導致當機情形。此時，若設備是安裝在不易人工隨時手動重開機的地方，或是因為設備停擺會導致危險災害 亦或是 產能與成本的巨大損失，就需要使用**看門狗功能**來自動幫您的設備重開機，並載入安全的設定值，以維護系統的安全並確保正常運作。

tM 系列 I/O 模組，具有雙看門狗保護機制。

雙看門狗 (Dual Watchdog) = 模組看門狗 (Module Watchdog) + 主機看門狗 (Host Watchdog)

模組看門狗 (Module Watchdog):

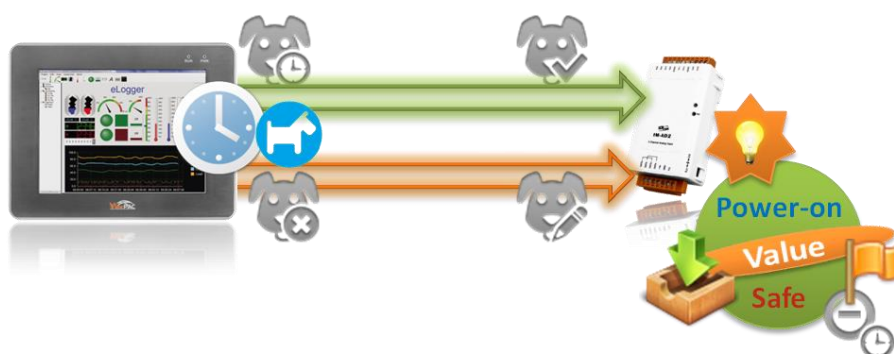


是一種硬體重置電路，可用來監控模組的運作狀態。硬體看門狗使用一個計時器，當模組正常工作時，會在超時前將 Timer 時間重置 (Reset) 為 0，再繼續計數。若發生超時 (Timeout) 狀況，模組會自動重開機並載入預設的開機值 (Power-on Value)。

主機看門狗 (Host Watchdog):



是一種軟體功能，可用來監控主機與模組之間的通訊狀態。軟體看門狗使用控制主機 (CPU) 的內部計時器，當系統正常運作時，會傳送主機正常的訊號給模組，若發生超時狀況，模組會自動載入安全值 (Safe Value)。如有清除超時狀態，則會在重開機後載入開機值 (Power-on Value)。



您可使用 DCON Utility Pro 來測試**主機看門狗功能** (參考 [3.4.5 節](#)，設定頁面 - Host WDT)，在 “Host WDT” 頁籤中，可設定 WDT 時間 (Watchdog Timeout)，也可清除超時狀態。

A.2.1 開機值 (Power-on Value)

每次上電開機、軟體命令重開機、或 Module Watchdog 超時導致的重開機，輸出通道會載入預設的 Power-on Value。

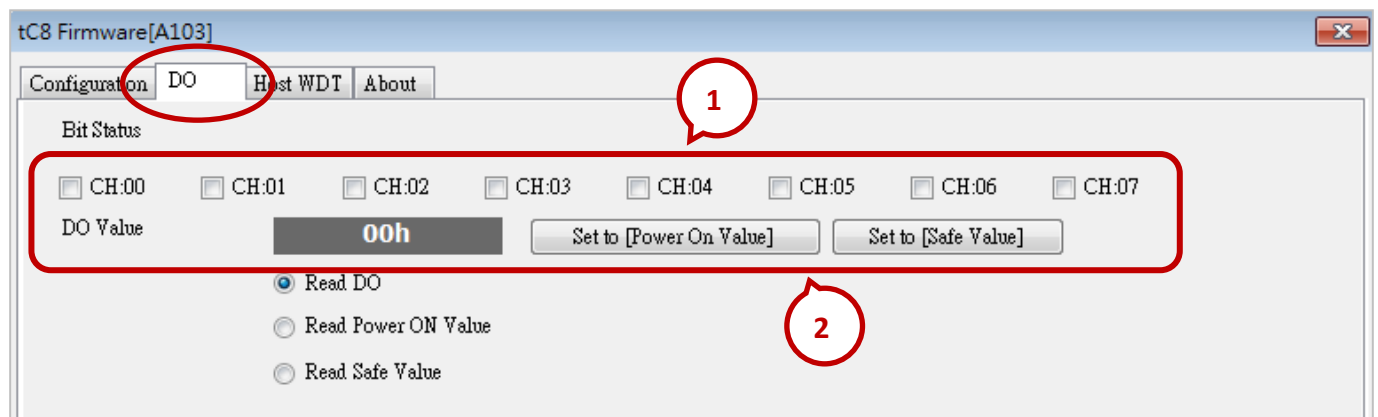
A.2.2 安全值 (Safe Value)

當啟用了 Host Watchdog 功能，且發生了超時狀況 (註: tM 模組上的 PWR 燈號會閃，超時狀態會儲存在 EEPROM，即使模組重開機也不會改變。)，輸出通道會先載入安全值 (Safe Value)。如有清除超時狀態，則會在下次重開機後，載入 “Power-on Value”。

可查閱附錄 E.3 Host Watchdog 相關指令。

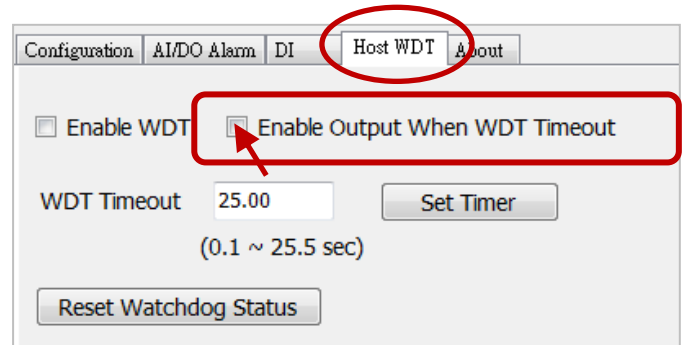
指令	說明
~AA1	重置模組的 Host Watchdog 狀態。
~AA5N	將模組 AO 值，設為 Safe Value。
~AA5V	將模組 DO 值，設為 Power-on 或 Safe Value。
\$AA4N	將模組 AO 值，設為 Power-on Value。

您可使用 DCON Utility Pro 來設定 Power-on Value 或 Safe Value，並啟用 Host Watchdog 功能 (參考 3.4.2 ~ 3.4.5 節)。在 “AO/DO” 頁籤中，可為 tM 模組的每個輸出通道，設定個別的值。



注意:

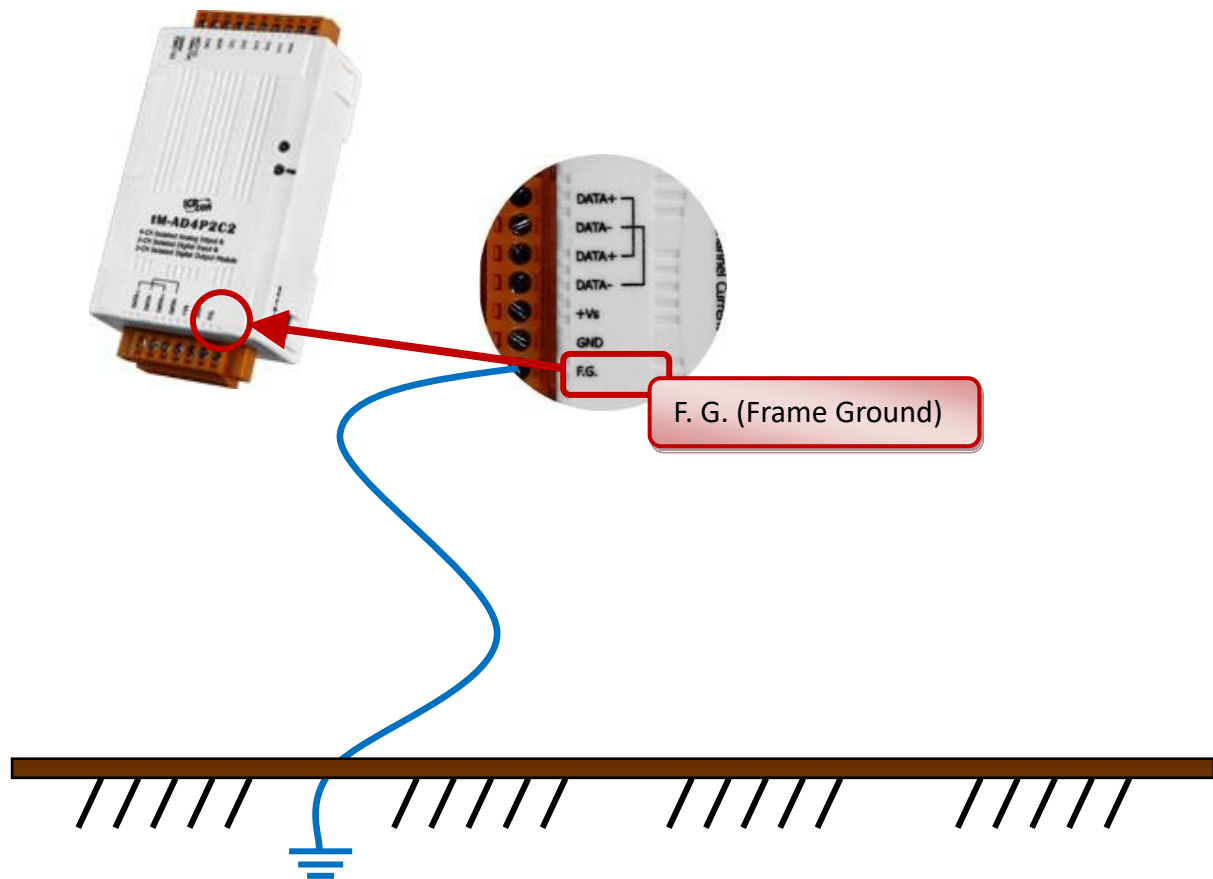
當 Protocol = DCON 時，Host Watchdog Timeout 後，輸出值會立即設為 Safe Value，此時，無法對 AO/DO 下指令，來變更輸出值。但是，當 Protocol = Modbus 時，DCON Utility Pro 有提供 Timeout 後仍可對 AO/DO 下指令的功能。



A.3 機殼接地 (Frame Ground)

電子電路常容易受到靜電放電 (ESD) 的影響，由其是在大陸性氣候區域更為嚴重。tM 系列模組提供了新的機殼接地 (Frame Ground) 設計，可旁路 ESD 路徑，從而增強了靜電保護 (ESD) 能力，並確保模組的可靠度。

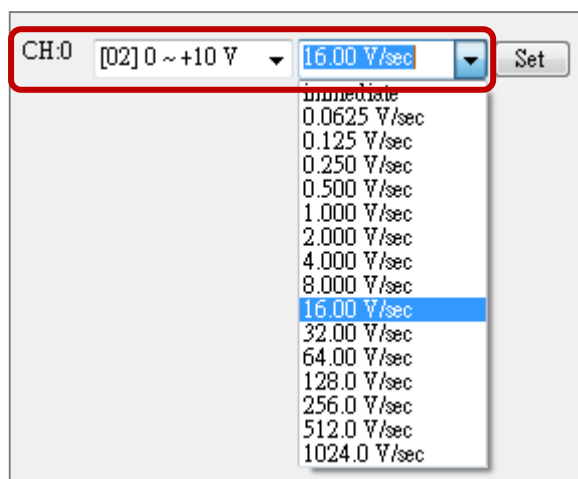
將模組的接地端子 (F. G.) 連接到導線，並將導線連接到接地端，可為模組提供更加完善的保護。



A.4 轉換率 (Slew Rate) 控制

此功能用於類比輸出模組 (例如: tM-DA1P1R1) , 一般而言, 使用者可立即地設置模組的輸出值, 而轉換率 (Slew Rate), 則可設定每秒改變多少電壓/電流值。

註: tM-DA1P1R1 模組是每 10 ms 就會更新一個通道。

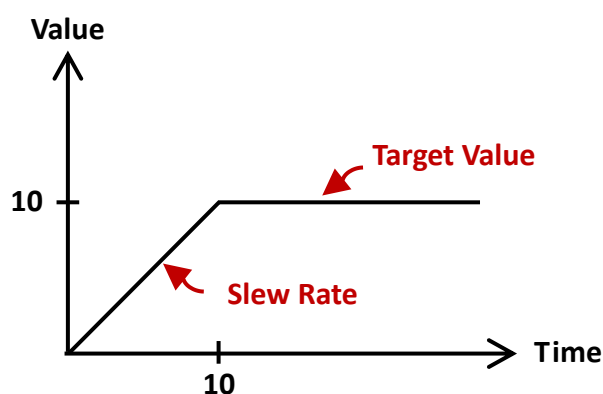


(DCON Utility Pro – AO 畫面)

Slew Rate (S) 設定表

S	V/sec	mA/sec
0	Immediate	Immediate
1	0.0625	0.125
2	0.125	0.25
3	0.25	0.5
4	0.5	1.0
5	1.0	2.0
6	2.0	4.0
7	4.0	8.0
8	8.0	16.0
9	16.0	32.0
A	32.0	64.0
B	64.0	128.0
C	128.0	256.0
D	256.0	512.0
E	512.0	1024.0

例如: 模組目前的輸出值是 0V, 想要改變為 10V 時, 您可立刻設定為 10V, 也可設定每 1 秒增加 1V 的方式, 讓輸出值逐漸地增加到 10V。



A.5 類比輸出回讀值 (AO Read-back)

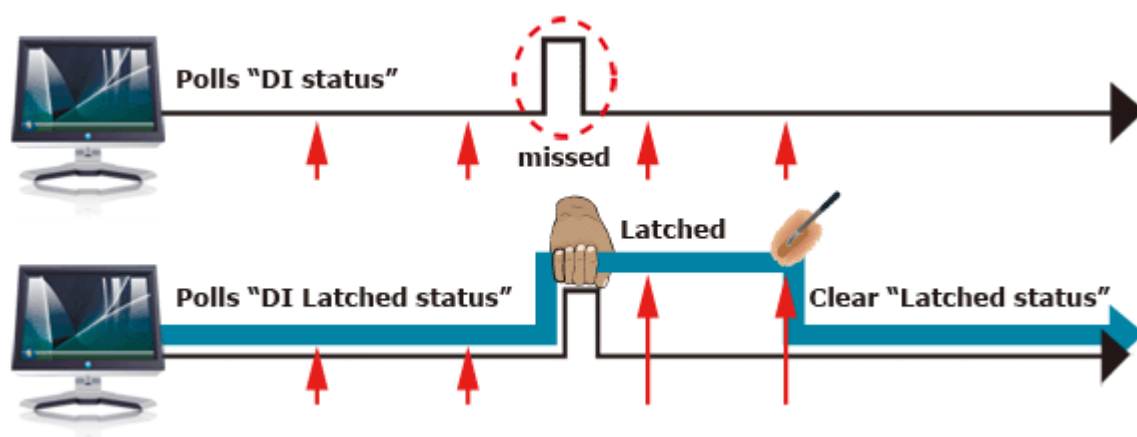
有些模組可使用硬體電路去量測實際的輸出值, 此量測值就稱為 Readback。然而, tM-DA1P1R1 模組並沒有此電路, Readback 即是設定的 AO 值。此外, 當有設定 Slew Rate 時, 輸出值會變化, 因此 Readback 和設定的 AO 值就可能不同。

A.6 進階數位輸入功能 (DI Latch、Counter)

模組中的數位輸入 (DI) 通道不僅是作讀取數位輸入狀態使用，同時更能提供多種進階功能。

DI Latch

tM 系列模組的 DI 通道皆有提供門鎖 (Latch) 的功能，用以保存模組內建暫存器的高/低準位事件。通常主控制器會定時地分別輪詢模組，來取得所有的數位輸入狀態，但有時可能會有突發事件，還未詢問到的 DI 通道，突然產生一個極短暫的脈衝信號 (Pulse Signal)，以致漏掉該信號。使用 DI Latch 功能將不會再遺漏短時間內 ($\geq 5\text{ ms}$) 的信號。



tM 模組有提供讀取 DI 的高門鎖 (High Latch) 或 低門鎖 (Low Latch) 狀態的指令，關於 DI 門鎖 (Latch) 指令的詳細說明，請參考 附錄 E.4 - DI Latched 指令。

低速計數器

tM 系列的 DI 模組除了能偵測數位輸入的高低準位，也會對數位輸入進行計數，低於 100 Hz 的信號都能成功被檢測與計數。



最大計數值：
65535 (16-bit)

附錄 B 類比輸出/入範圍的 Type Code

以下列出 tM 系列模組的類比輸出/入所使用的 Type Code。同一個 Type Code 可以有 2 種資料的格式可以設定，分別為

■ Engineering (工程數值):

實際的類比值 與 軟體讀到的值，通常是容易計算的 1000 或 10000 倍的數值。例如：

Type Code 05: 軟體讀到 15000 代表實際上是 1.5 V；軟體讀到 -23000 代表 -2.3 V。

Type Code 08: 軟體讀到 2500 代表實際上是 2.5 V；軟體讀到 -6500 代表 -6.5 V。

Type Code	Range	Data Format	Min.	Max.
05	-2.5 V ~ +2.5 V	Engineering	-25000	+25000
		2's Complement	8000h	7FFFh
06	-20 mA ~ +20 mA	Engineering	-20000	+20000
		2's Complement	8000h	7FFFh
07	+4 mA ~ +20 mA	Engineering	+4000	+20000
		2's Complement	0000h	FFFFh
08	-10 V ~ +10 V	Engineering	-10000	+10000
		2's Complement	8000h	7FFFh

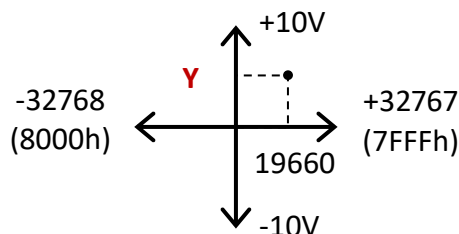
■ 2's Complement (二補數):

在文件上或是軟體的數值，會以 16 進制來表示，例如 “1230h” 或是 “ABCDh” (最後的 “h” 代表是 16 進制)，軟體的值與實際的值，要透過公式做比例換算。例如：

Type Code 08: 軟體讀到 19660 (十進制) 或是 4CCCh (十六進制)，代表實際是 6.0 V。

$$\text{數值轉換: } \frac{Y}{10} = \frac{19660}{32767}$$

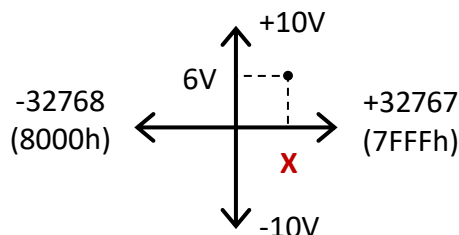
$$Y = 6.0$$



Type Code 08: 要輸出 6 V，軟體的數值必需是 19660 (十進制) 或是 4CCCh (十六進制)。

$$\text{數值轉換: } \frac{6}{10} = \frac{X}{32767}$$

$$X = 19660.2$$



B.1 tM-AD2 資料範圍

由於 tM-AD2 只支援正電壓/電流，因此 Type Code 會列出正電壓/電流的範圍 與 數值範圍。

例如：Type Code = **05**，資料範圍為 0 ~ +2.5 V，數值範圍為 0 (0000h) ~ +32767 (**7FFFh**)。

Type Code = 1A，資料範圍為 0 ~ 20 mA，數值範圍為 0 (0000h) ~ +65535 (**FFFFh**)。

Type Code	Range	Data Format	Min.	Max.
05	0 V ~ +2.5 V	Engineering	0	+25000
		2's Complement	0000h	7FFFh
07	4 mA ~ 20 mA	Engineering	4000	20000
		2's Complement	0000h	FFFFh
08	0 V ~ +10 V	Engineering	0	+10000
		2's Complement	0000h	7FFFh
09	0 V ~ +5 V	Engineering	0	+5000
		2's Complement	0000h	7FFFh
0A	0 V ~ +1 V	Engineering	0	+10000
		2's Complement	0000h	7FFFh
0B	0 V ~ +500 mV	Engineering	0	+5000
		2's Complement	0000h	7FFFh
0D	0 mA ~ +20 mA	Engineering	0	+20000
		2's Complement	0000h	7FFFh
1A	0 mA ~ 20 mA	Engineering	0	20000
		2's Complement	0000h	FFFFh

註：資料格式為 %FSR，所有 Type Code 的資料範圍是 +000.00 ~ +100.00。

B.2 tM-AD5 資料範圍

tM-AD5 支援下列正、負電壓 與 數值範圍 (-32768 ~ 32767)。

Type Code	Range	Data Format	Min.	Max.
05	-2.5 V ~ +2.5 V	Engineering	-25000	+25000
		2's Complement	8000h	7FFFh
08	-10 V ~ +10 V	Engineering	-10000	+10000
		2's Complement	8000h	7FFFh
09	-5 V ~ +5 V	Engineering	-5000	+5000
		2's Complement	8000h	7FFFh
0A	-1 V ~ +1 V	Engineering	-10000	+10000
		2's Complement	8000h	7FFFh

註: 資料格式為 %FSR，所有 Type Code 的資料範圍是 -100.00 ~ +100.00。

B.3 tM-AD5C 資料範圍

tM-AD5C 支援下列正、負電流 與 數值範圍 (-32768 ~ 32767 或 0 ~ 65535)。

Type Code	Range	Data Format	Min.	Max.
07	4 mA ~ 20 mA	Engineering	4000	20000
		2's Complement	0000h	FFFFh
0D	-20 mA ~ +20 mA	Engineering	-20000	+20000
		2's Complement	8000h	7FFFh
1A	0 mA ~ 20 mA	Engineering	0	20000
		2's Complement	0000h	FFFFh

註: 資料格式為 %FSR，所有 Type Code 的資料範圍是 -100.00 ~ +100.00。

B.4 tM-AD8 資料範圍

tM-AD5 支援下列正電壓 與 數值範圍 (0 ~ 32767)。

例如: Type Code = **0B**，資料範圍為 **0 ~ +500 mV**，數值範圍為 0 (0000h) ~ +32767 (**7FFFh**)。

Type Code	Range	Data Format	-F.S	+F.S
05	0 V ~ +2.5 V	Engineering	0	+25000
		2's Complement	0000h	7FFFh
08	0 V ~ +10 V	Engineering	0	+10000
		2's Complement	0000h	7FFFh
09	0 V ~ +5 V	Engineering	0	+5000
		2's Complement	0000h	7FFFh
0A	0 V ~ +1 V	Engineering	0	+10000
		2's Complement	0000h	7FFFh
0B	0 mV ~ +500 mV	Engineering	0	+5000
		2's Complement	0000h	7FFFh

註: 資料格式為 %FSR，所有 Type Code 的資料範圍是 +000.00 ~ +100.00。

B.5 tM-AD8C 資料範圍

tM-AD5C 只支援正電流 (0 ~ 20 mA 與 4 ~ 20 mA)。

例如: Type Code = **0D**，資料範圍為 **0 ~ +20 mA**，數值範圍為 0 (0000h) ~ +32767 (**7FFFh**)。

Type Code	Range	Data Format	Min.	Max.
07	4 mA ~ 20 mA	Engineering	4000	20000
		2's Complement	0000h	FFFFh
0D	0 mA ~ +20 mA	Engineering	0	20000
		2's Complement	0000h	7FFFh
1A	0 mA ~ 20 mA	Engineering	0	20000
		2's Complement	0000h	FFFFh

註: 資料格式為 %FSR，所有 Type Code 的資料範圍是 +000.00 ~ +100.00。

B.6 tM-AD4P2C2 資料範圍

tM-AD4P2C2 支援下列正、負電壓/電流 與 數值範圍 (-32768 ~ 32767 或 0 ~ 65535)。

Type Code	Range	Data Format	Min.	Max.
05	-2.5 V ~ +2.5 V	Engineering	-25000	+25000
		2's Complement	8000h	7FFFh
06	-20 mA ~ +20 mA	Engineering	-20000	+20000
		2's Complement	8000h	7FFFh
07	4 mA ~ 20 mA	Engineering	4000	20000
		2's Complement	0000h	FFFFh
08	-10 V ~ +10 V	Engineering	-10000	+10000
		2's Complement	8000h	7FFFh
09	-5 V ~ +5 V	Engineering	-5000	+5000
		2's Complement	8000h	7FFFh
0A	-1 V ~ +1 V	Engineering	-10000	+10000
		2's Complement	8000h	7FFFh
0D	-20 mA ~ +20 mA	Engineering	-20000	+20000
		2's Complement	8000h	7FFFh
1A	0 mA ~ 20 mA	Engineering	0	20000
		2's Complement	0000h	FFFFh

B.7 tM-DA1P1R1 資料範圍

tM-DA1P1R1 支援下列正電壓/電流 與 數值範圍。

Type Code	Range	Data Format	Min.	Max.
0	0 mA ~ 20 mA	Engineering	0	+20000
1	4 mA ~ 20 mA		4000	20000
2	0 V ~ 10 V		0	+10000
4	0 V ~ 5 V		0	+5000

註: 資料格式為 2's Complement，所有 Type Code 的資料範圍是 0000h ~ FFFFh (0 ~ 65535)。

B.8 tM-TH8 資料範圍

tM-TH8 支援下列熱敏電阻的類型 與 數值範圍。

Type Code	Thermistor Type (Range)	Data Format	Min.	Max.
60	PreCon Type III 10K @ 25°C (-30 °F ~ 240 °F)	Engineering	-3000	24000
		2's Complement	F000	7FFF
		% FSR	-012.50	+100.00
		Ohms	+000539.4	+173600.0
61	Fenwell Type U 2K @ 25°C (-50 °C ~ 150 °C)	Engineering	-5000	15000
		2's Complement	D556	7FFF
		% FSR	-033.33	+100.00
		Ohms	+000037.2	+134020.0
62	Fenwell Type U 2K @ 25°C (0 °C ~ 150 °C)	Engineering	0	15000
		2's Complement	0000	7FFF
		% FSR	+000.00	+100.00
		Ohms	+000037.2	+006530.0
63	YSI L Mix 100 @ 25°C (-80 °C ~ 100 °C)	Engineering	-8000	10000
		2's Complement	999A	7FFF
		% FSR	-080.00	+100.00
		Ohms	+000014.3	+014470.0
64	YSI L Mix 300 @ 25°C (-80 °C ~ 100 °C)	Engineering	-8000	10000
		2's Complement	999A	7FFF
		% FSR	-080.00	+100.00
		Ohms	+000035.8	+067660.0
65	YSI L Mix 1000 @ 25°C (-70 °C ~ 100 °C)	Engineering	-7000	10000
		2's Complement	A667	7FFF
		% FSR	-070.00	+100.00
		Ohms	+000106.4	+132600.0
66	YSI B Mix 2252 @ 25°C (-50 °C ~ 150 °C)	Engineering	-5000	15000
		2's Complement	D556	7FFF
		% FSR	-033.33	+100.00
		Ohms	+000041.8	+151000.0
67	YSI B Mix 3000 @ 25°C (-40 °C ~ 150 °C)	Engineering	-4000	15000
		2's Complement	DDDE	7FFF
		% FSR	-026.67	+100.00
		Ohms	+000055.6	+101000.0

Type Code	Thermistor Type (Range)	Data Format	Min.	Max.
68	YSI B Mix 5000 @ 25°C (-40 °C ~ 150 °C)	Engineering	-4000	15000
		2's Complement	DDDE	7FFF
		% FSR	-026.67	+100.00
		Ohms	+000092.7	+168300.0
69	YSI B Mix 6000 @ 25°C (-30 °C ~ 150 °C)	Engineering	-3000	15000
		2's Complement	E667	7FFF
		% FSR	-020.00	+100.00
		Ohms	+000111.5	+106200.0
6A	YSI B Mix 10000 @ 25°C (-30 °C ~ 150 °C)	Engineering	-3000	15000
		2's Complement	E667	7FFF
		% FSR	-020.00	+100.00
		Ohms	+000185.9	+177000.0
6B	YSI H Mix 10000 @ 25°C (-30 °C ~ 150 °C)	Engineering	-3000	15000
		2's Complement	E667	7FFF
		% FSR	-020.00	+100.00
		Ohms	+000237.0	+135200.0
6C	YSI H Mix 30000 @ 25°C (-10 °C ~ 200 °C)	Engineering	-1000	20000
		2's Complement	F99A	7FFF
		% FSR	-005.00	+100.00
		Ohms	+000186.7	+158000.0
70 ~ 77	User-defined (-50 °C ~ 150 °C)	Engineering	-5000	15000
		2's Complement	D556	7FFF
		% FSR	-033.33	+100.00
		Ohms	+000000.0	+000000.0

註: 針對 “User-defined” Type，若電阻值大於 204700 Ohms，將視為 Under Range。

Analog Format	Under Range	Over Range	Protocol
Engineering	-9999.9	+9999.9	DCON
	-32768	32767	Modbus RTU/ASCII
2's Complement	8000	7FFF	DCON Modbus RTU/ASCII
Percent (% FSR)	-999.99	+999.99	DCON

B.8.1 Steinhart–Hart 係數 (User-defined Type)

以下方法可獲得熱敏電阻的 Steinhart–Hart 係數：

1. 詢問製造商。一般熱敏電阻的製造商會發佈其產品的 Steinhart–Hart 係數。
2. 在熱敏電阻製造商公佈的阻抗與溫度的對照表中，使用三組已知的資料 (R1, T1), (R2, T2) 和 (R3, T3)，可求解係數。

DCON Utility Pro 可算出 Steinhart–Hart 係數，並轉換為 IEEE-754 格式，請參考 [3.4.1 設定頁面 - User Defined Type](#)。

針對典型的熱敏電阻，電阻與溫度之間的關係可用 Steinhart–Hart 方程式來表示：

$$1/T = A + B \ln R_T + C (\ln R_T)^3$$

R_T 表示在溫度 T (單位 $^{\circ}\text{K} = ^{\circ}\text{C} + 273.15$) 的電阻值 (Ω)； A 、 B 與 C 稱為 Steinhart–Hart 係數。方程式的誤差在 100°C 跨距內，小於 $\pm 0.01^{\circ}\text{C}$ 。

tM-TH8 可使用 [@AASxTttC\(data\)](#) 指令來設定 User-defined Type 的 Steinhart–Hart 係數，並以 32-bit 十六進制的 IEEE-754 標準格式傳送資料：

Bit	說明
31 (符號)	0: 表示正號 ; 1: 表示負號
30-23 (指數)	指數基底為 2，實際的指數值為儲存值減去 127。
22-00 (尾數)	尾數以 1.f 表示，f 為儲存在此欄位的小數部份。

例如：

16 進制值 **C3694000h** = **1100 0011 0110 1001 0100 0000 0000 0000** ⁽²⁾

C	3	6	9	4	0	0	0
1100	0011	0110	1001	0100	0000	0000	0000

Bit 31 為 **1**，表示為負數值。

Bit 30 ~ 23 為二進制 **10000110** 或 十進制 134，而指數值為 $134 - 127 = 7$ 。

Bit 22 ~ 00 為二進制 **1.1101001010000000000000**，以指數值 (7) 來調整尾數，調整後尾數為二進制 **11101001.0100000000000000**，即十進制 233.25。

因此，C3694000h 的浮點數值為 -233.25。

附錄 C Modbus 位址對應

適用 AI、AO、DI、DO 模組

位址 (Base 1)	讀/寫	說明		
00257	R/W	通訊協定	0: DCON ； 1: Modbus RTU	
00258		通訊協定	0: 依位址 00257 而定 ； 1: Modbus ASCII	
00273	R	模組重置 (Reset) 狀態	0: 目前未重開機 1: 目前已重開機 (或 首次開機)	
40481		韌體版本 (Low Word)		
40482		韌體版本 (High Word)		
40483		模組名稱 (Low Word)		
40484		模組名稱 (High Word)		
40485	R/W	Module 位址	有效範圍: 1 ~ 247	
40486		Bits 5 : 0 Baud Rate : 0x03 ~ 0x0A Bits 7 : 6 00: No Parity · 1 Stop Bit 01: No Parity · 2 Stop Bit 10: Even Parity · 1 Stop Bit 11: Odd Parity · 1 Stop Bit	Code	Baud Rate
			0x03	1200
			0x04	2400
			0x05	4800
			0x06	9600
			0x07	19200
			0x08	38400
			0x09	57600
			0x0A	115200
			40488	Modbus Respond Delay (ms)
00261		Host Watchdog	0: 關閉 ； 1: 啟用	
00270		Host Watchdog 的 Timeout 狀態	寫入 1 來清除此狀態	
40489		Host Watchdog 的 Timeout 值	有效範圍: 0 ~ 255 (0.1s)	
40492		Host Watchdog 的 Timeout 計數值	寫入 0 來清除此值	

適用 AI/AO 模組

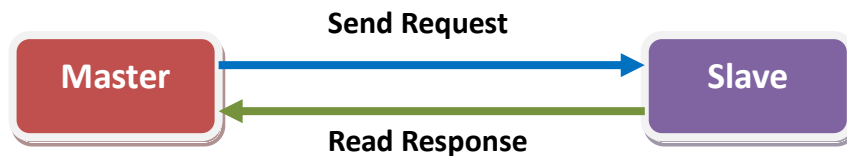
位址 (Base 1)	讀/寫	說明	
30001 ~ 30008 40001 ~ 40008	R	AI 的通道值	
40490	R/W	啟用/關閉 AI 通道 (n)	有效範圍: 00h ~ (2 ⁿ -1)h
00269		Modbus 資料格式	0: 十六進制 · 1: 工程值
00271		Fast Mode (不適用: tM-TH8/DA1P1R1)	0: 關閉 · 1: 啟用
00129 ~ 00136 10129 ~ 10136	R	AI 通道超出/低於範圍 4 ~ 20 mA 或 0 ~ 20 mA 的狀態	(適用: tM-AD2/AD5C/AD8C)
00225 ~ 00228 10225 ~ 10228			(適用: tM-AD4P2C2)
40257 ~ 40264	R/W	AI 通道的資料類型 (Type Code)	(適用: tM-AD2/TH8/AD4P2C2)
40487			(適用: tM-AD5/AD8/AD5C/AD8C)
適用: tM-AD4P2C2			
40225 ~ 40226	R/W	AI 通道 0 ~ 1 的警報高限值	
40233 ~ 40234		AI 通道 0 ~ 1 的警報低限值	
適用: tM-AD2			
40494	R/W	4 mA 至 20mA 的低限值 (Threshold)	有效範圍: 0 ~ 40 (0.1mA)
適用: tM-TH8			
40385 ~ 40392	R/W	AI 通道 0 ~ 7 的電阻偏移量 (0.1 歐姆)	有效範圍: 0 ~ 255
40449 ~ 40456		AI 通道 0 ~ 7 的溫度偏移量 (0.1 °C)	有效範圍: -128 ~ 127
40769 ~ 40784		資料類型 (Type Code) 為 70 ~ 77 的 Steinhart 係數 A	
40801 ~ 40816		資料類型 (Type Code) 為 70 ~ 77 的 Steinhart 係數 B	
40833 ~ 40848		資料類型 (Type Code) 為 70 ~ 77 的 Steinhart 係數 C	
00267		溫度單位	0: 華氏 · 1: 攝氏
00272	W	寫入 1 · 來載入出廠校正參數	
適用: tM-DA1P1R1			
40033	R/W	AO 通道 0 的值	
30065, 40065	R	AO 通道的回讀值 (Readback)	
40097	R/W	AO 通道的安全值 (Safe Value)	
40193		AO 通道的開機值 (Power-on Value)	
40289		AO 通道的轉換率 (Slew Rate)	
40417		AO 通道的資料類型 (Type Code)	

適用 DI/DO 模組

位址 (Base 1)	讀/寫	說明	
00033 ~ 00040 10033 ~ 10040	R	DI 通道值	
00001 ~ 00008	R/W	DO 通道值	
00265		DI 通道的現行狀態值	0: 一般 (ON/OFF 讀回 1/0)
00266		DO 通道的現行狀態值	1: 反相 (ON/OFF 讀回 0/1)
00065 ~ 00072 10065 ~ 10072	R	DI 通道的高門鎖 (High Latch) 值	
00097 ~ 00104 10097 ~ 10104		DI 通道的低門鎖 (Low Latch) 值	
00073 ~ 00080 10073 ~ 10080		DO 通道的高門鎖 (High Latch) 值	
00105 ~ 00112 10105 ~ 10112		DO 通道的低門鎖 (Low Latch) 值	
00264	W	寫入 1 來清除 DIO 門鎖值	
30001 ~ 30008	R		(適用: 一般 DI)
30129 ~ 30130 40129 ~ 40130		DI 通道的計數值 (16-bit · 0 ~ 65535)	(適用: tM-DA1P1R1/ AD4P2C2)
00193 ~ 00200	R/W	更新邊緣觸發計數器 (Counter) 0: 當 DI 通道出現下降邊緣訊號時, 會更新計數器 1: 當 DI 通道出現上升邊緣訊號時, 會更新計數器	
00513 ~ 00520	W	寫入 1 來清除計數值	
00260	R/W	Host Watchdog 模式 0: Timeout 後, 需清除 Timeout 狀態, 才可對 AO/DO 下指令 1: Timeout 後, 可對 AO/DO 下指令並會自動清除 Timeout 狀態	
00129 ~ 00136		DO 通道的安全值 (Safe Value)	
00161 ~ 00168		DO 通道的開機值 (Power-on Value)	
適用: tM-AD4P2C2			
00289 ~ 00290	R/W	通道 0 ~ 1 的低限警報狀態 寫入 1 可清除低限門鎖 (Latch) 警報	
00305 ~ 00306		通道 0 ~ 1 的高限警報狀態 寫入 1 可清除高限門鎖 (Latch) 警報	
00321 ~ 00322		啟用/關閉通道 0 ~ 1 的警報	
00337 ~ 00338		通道 0 ~ 1 的警報類型 (Momentary 或 Latch)	

附錄 D Modbus RTU Protocol 指令

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。可至 <https://www.modbus.org> 了解更多關於 Modbus 的詳細資訊。



Modbus RTU 資料結構:

Byte 00	Byte 01	Byte 02-03	Byte 04-05	CRC
Slave 設備的位址 (Net ID · 1 ~ 247)	功能碼 (Function Code)	資料		檢查碼 (2 Byte)
		起始位址	通道數	

- 註:** 1) 以下說明將省略 CRC 內容。
2) 請參考附錄 C - Modbus 位址對應，來查詢起始位址。

以下是 tM 系列模組支援的 Modbus RTU 功能：

功能碼	Bit	功能說明		參考位址
01 (0x01)	1	Read Coil	讀取 DO 狀態	0xxxx
02 (0x02)	1	Read Discrete Input	讀取 DI 狀態	1xxxx
03 (0x03)	16	Read Multiple Registers	讀取 AO 值	4xxxx
04 (0x04)	16	Read Multiple Input Registers	讀取 AI 值	3xxxx
05 (0x05)	1	Force Single Coil	寫入一個 DO 狀態	0xxxx
06 (0x06)	1	Preset Single Register	寫入一個 AO 值	4xxxx
15 (0x0F)	16	Force Multiple Coils	寫入多個 DO 狀態	0xxxx
16 (0x10)	16	Preset Multiple Registers	寫入多個 AO 值	4xxxx

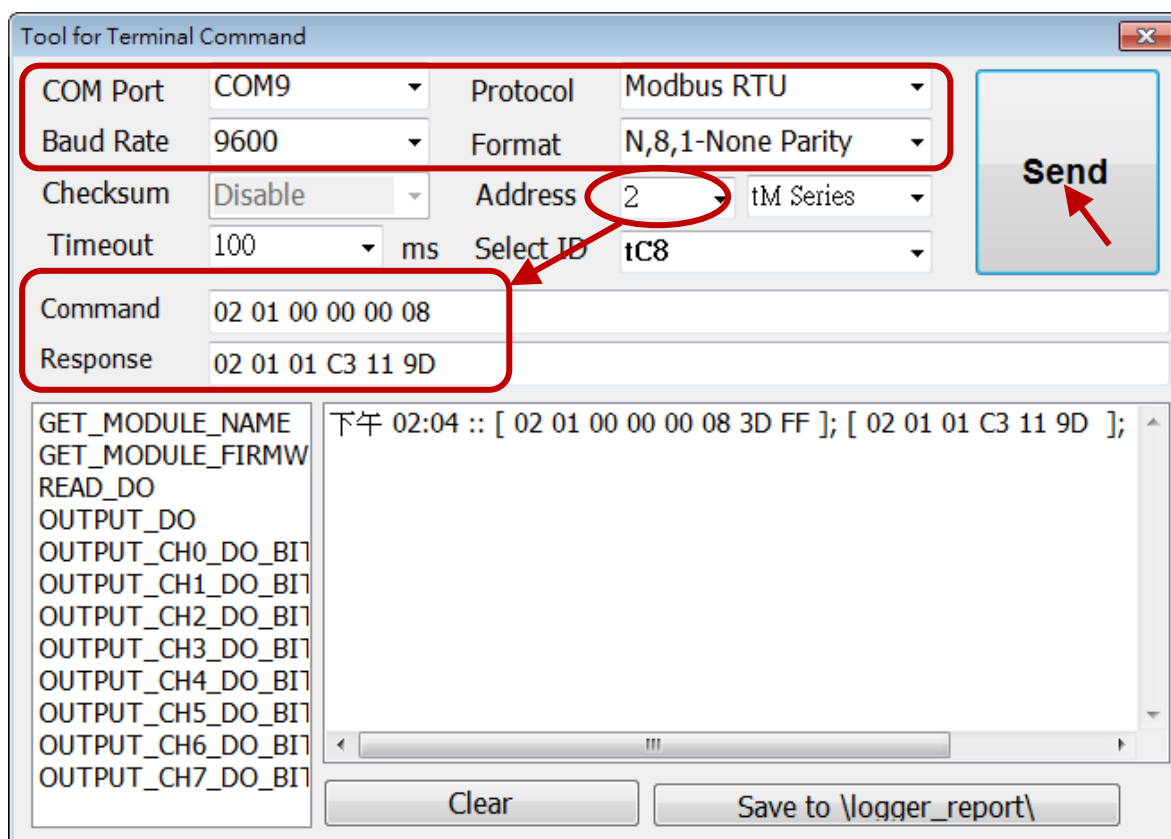
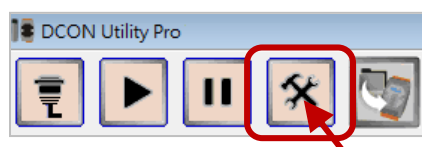
註:

- Modbus 位址 0xxxx 可用來讀取/寫入 DO 通道的狀態。
Modbus 位址 1xxxx 可用來讀取 DI 通道的狀態。
Modbus 位址 3xxxx 可用來讀取 AI 通道的數值。
Modbus 位址 4xxxx 可用來讀取/寫入 AO 通道的數值。

異常代碼 (Exception Code)

異常碼 (Hex.)	名稱	說明
01	Illegal Function	不支援的功能
02	Illegal Data Address	不合法的位址
03	Illegal Data Value	不合法的資料值
04	Slave Device Failure	Slave 設備失效
05	Acknowledge	告知 (命令仍處理中)
06	Slave Device Busy	Slave 設備忙碌

您可使用 DCON Utility Pro 的 **Command Line** 功能，來測試以下指令。



Command: 02 01 00 00 00 08

02	01	00	00	00	08
模組 ID = 02	讀取 DO 狀態	起始位址 = 0000 (Base 0)		讀取 8 個通道	

Response: 02 01 01 C3 11 9D

02	01	01	C3	11	9D
模組 ID = 02	讀取 DO 狀態	資料為 1 Byte	1100 0011 表示 DO0, 1, 6, 7 = ON	CRC	

D.1 01 (0x01) Read Coil

此功能碼用來讀取 DO 或 線圈 (Coil) 的狀態值。

[Request]

Slave 位址	功能碼	起始位址 (High · Low)	通道數量 (High · Low)
1 Byte	1 Byte	2 Byte	2 Byte
1 ~ 247	0x01	見附錄 C Modbus 位址對應	tM 模組: Max. 8 通道

[Response]

Slave 位址	功能碼	資料數 (Byte)	資料值
1 Byte	1 Byte	1 Byte	n Byte
1 ~ 247	0x01	n = (通道數+7)/8	n = 1; Byte 03 = bit 7 ~ 0 n = 2; Byte 04 = bit 15 ~ 8 n = m; Byte m+2 = bit (8m-1) ~ 8(m-1)

[Error Response]

Slave 位址	功能碼	異常碼
1 Byte	1 Byte	1 Byte
1 ~ 247	0x81 (80+功能碼)	見 " 異常代碼 (Exception Code)"

範例:

請求命令 (Master): **01 01 00 00 00 02** (Hex.)

Slave 位址	功能碼	起始位址		通道數量	
Byte 00	Byte 01	Byte 02 (H)	Byte 03 (L)	Byte 04 (H)	Byte 05 (L)
01	01	00	00	00	02
模組 ID = 01	讀取 DO 狀態	位址 = 00000 (Base 0)		讀取 2 個通道	

註: tM 模組的 Base 位址為 0 (即 · Base 0) · 因此起始位址需設為 Modbus 位址減 1 (見附錄 C)。

回應 (Slave): **01 01 01 03** (Hex.)

Slave 位址	功能碼	資料數 (Byte)	資料值
Byte 00	Byte 01	Byte 02	Byte 03
01	01	01	03
模組 ID = 01	讀取 DO 狀態	資料為 1 Byte	0000 0011 表示 DO0, 1 = ON

D.2 02 (0x02) Read Discrete Inputs

此功能碼用來讀取 DI 狀態值。

[Request]

Slave 位址	功能碼	起始位址 (High · Low)	通道數量 (High · Low)
1 Byte	1 Byte	2 Byte	2 Byte
1 ~ 247	0x02	見附錄 C Modbus 位址對應	tM 模組: Max. 8 通道

[Response]

Slave 位址	功能碼	資料數 (Byte)	資料值
1 Byte	1 Byte	1 Byte	n Byte
1 ~ 247	0x02	$n = (\text{通道數} + 7) / 8$	$n = 1$; Byte 03 = bit 7 ~ 0 $n = 2$; Byte 04 = bit 15 ~ 8 $n = m$; Byte $m+2$ = bit $(8m-1) \sim 8(m-1)$

[Error Response]

Slave 位址	功能碼	異常碼
1 Byte	1 Byte	1 Byte
1 ~ 247	0x82 (80+ 功能碼)	見 "異常代碼" (Exception Code)"

範例:

請求命令 (Master): **01 02 00 32 00 02** (Hex.)

Slave 位址	功能碼	起始位址		通道數量	
Byte 00	Byte 01	Byte 02 (H)	Byte 03 (L)	Byte 04 (H)	Byte 05 (L)
01	02	00	32	00	02
模組 ID = 01	讀取 DI 狀態	位址 = 10032 (Base 0)		讀取 2 個通道	

註: tM 模組的 Base 位址為 0 (即 · Base 0) · 因此起始位址需設為 Modbus 位址減 1 (見附錄 C)。

回應 (Slave): **01 02 01 03** (Hex.)

Slave 位址	功能碼	資料數 (Byte)	資料值
Byte 00	Byte 01	Byte 02	Byte 03
01	02	01	03
模組 ID = 01	讀取 DI 狀態	資料為 1 Byte	0000 0011 表示 DO0, 1 = ON

D.3 03 (0x03) Read Multiple Registers

此功能碼用來讀取多個 AO 或 保存暫存器 (Holding Registers) 的值。

註: tM-DA1P1R1 支援一個 AO 通道。

[Request]

Slave 位址	功能碼	起始位址 (High · Low)	通道數量 (High · Low)
1 Byte	1 Byte	2 Byte	2 Byte
1 ~ 247	0x03	見附錄 C Modbus 位址對應	Word 數 (16-bit Register)

[Response]

Slave 位址	功能碼	資料數 (Byte)	資料值
1 Byte	1 Byte	1 Byte	n Byte
1 ~ 247	0x03	n = 通道數 * 2 Byte	n = m; Byte 03 = High Byte Byte 04 = Low Byte ... Byte m+1 = High Byte Byte m+2 = High Byte

[Error Response]

Slave 位址	功能碼	異常碼
1 Byte	1 Byte	1 Byte
1 ~ 247	0x83 (80+ 功能碼)	見 " 異常代碼 (Exception Code)"

範例:

請求命令 (Master): **01 03 01 ED 00 01** (Hex.)

Slave 位址	功能碼	起始位址		通道數量	
Byte 00	Byte 01	Byte 02 (H)	Byte 03 (L)	Byte 04 (H)	Byte 05 (L)
01	03	01	ED	00	01
模組 ID = 01	讀取 Holding Registers	位址 = 40493 (Base 0)		讀取 1 個通道	

註: tM 模組的 Base 位址為 0 (即 · Base 0) · 因此起始位址需設為 Modbus 位址減 1 (見附錄 C)。

回應 (Slave): **01 03 02 00 1E** (Hex.)

Slave 位址	功能碼	資料數 (Byte)	資料值	
Byte 00	Byte 01	Byte 02	Byte 03 (H)	Byte 04 (L)
01	03	02	00	1E
模組 ID = 01	讀取 Holding Registers	資料為 2 Byte	電流值小於 3mA · 會視為 Under Range (斷線)	

D.4 04 (0x04) Read Multiple Input Registers

此功能碼用來讀取多個 AI 或 輸入暫存器 (Input Registers) 或 計數器 (Counter) 的值。

[Request]

Slave 位址	功能碼	起始位址 (High · Low)	通道數量 (High · Low)
1 Byte	1 Byte	2 Byte	2 Byte
1 ~ 247	0x04	見附錄 C Modbus 位址對應	Word 數 (16-bit Register)

[Response]

Slave 位址	功能碼	資料數 (Byte)	資料值
1 Byte	1 Byte	1 Byte	n Byte
1 ~ 247	0x04	n = 通道數 * 2 Byte	n = m; Byte 03 = High Byte Byte 04 = Low Byte Byte m+1 = High Byte Byte m+2 = Low Byte

[Error Response]

Slave 位址	功能碼	異常碼
1 Byte	1 Byte	1 Byte
1 ~ 247	0x84 (80+功能碼)	見 " 異常代碼 (Exception Code)"

範例:

請求命令 (Master): **01 04 00 07 00 01** (Hex.)

Slave 位址	功能碼	起始位址		通道數量	
Byte 00	Byte 01	Byte 02 (H)	Byte 03 (L)	Byte 04 (H)	Byte 05 (L)
01	04	00	07	00	01
模組 ID = 01	讀取 Input Registers	位址 = 30007 (Base 0)		讀取 1 個通道	

註: tM 模組的 Base 位址為 0 (即 · Base 0) · 因此起始位址需設為 Modbus 位址減 1 (見附錄 C)。

回應 (Slave): **01 04 02 00 05** (Hex.)

Slave 位址	功能碼	資料數 (Byte)	資料值	
Byte 00	Byte 01	Byte 02	Byte 03 (H)	Byte 04 (L)
01	04	02	00	05
模組 ID = 01	讀取 Input Registers	資料為 2 Byte	DI7 的計數值 = 5	

D.5 05 (0x05) Force Single Coil

此功能碼用來寫入單個 DO 或 線圈 (Coil) 的狀態值。

[Request]

Slave 位址	功能碼	起始位址 (High · Low)	資料值 (High · Low)
1 Byte	1 Byte	2 Byte	2 Byte
1 ~ 247	0x05	見附錄 C Modbus 位址對應	0xFF00 表示設定為 ON 0x0000 表示設定為 OFF (其他值皆不合法，不會套用)

[Response]

Slave 位址	功能碼	起始位址	資料值
1 Byte	1 Byte	2 Byte	2 Byte
1 ~ 247	0x05	同 Request 的 Byte 02-03	同 Request 的 Byte 04-05

[Error Response]

Slave 位址	功能碼	異常碼
1 Byte	1 Byte	1 Byte
1 ~ 247	0x85 (80+ 功能碼)	見 " 異常代碼 (Exception Code)"

範例:

請求命令 (Master): **02 05 00 03 FF 00** (Hex.)

Slave 位址	功能碼	起始位址		資料值	
Byte 00	Byte 01	Byte 02 (H)	Byte 03 (L)	Byte 04 (H)	Byte 05 (L)
02	05	00	03	FF	00
模組 ID = 02	寫入 DO 值	位址 = 00003 (Base 0)		設定為 ON	

註: tM 模組的 Base 位址為 0 (即 · Base 0) · 因此起始位址需設為 Modbus 位址減 1 (見附錄 C)。

回應 (Slave): **02 05 00 03 FF 00** (Hex.)

Slave 位址	功能碼	起始位址		資料值	
Byte 00	Byte 01	Byte 02 (H)	Byte 03 (L)	Byte 04 (H)	Byte 05 (L)
02	05	00	03	FF	00
模組 ID = 02	寫入 DO 值	00003 · 表示 DO3		已設定為 ON	

D.6 06 (0x06) Preset Single Register

此功能碼用來寫入單個 AO 或 保存暫存器 (Holding Registers) 的值。

[Request]

Slave 位址	功能碼	起始位址 (High · Low)	資料值 (High · Low)
1 Byte	1 Byte	2 Byte	2 Byte
1 ~ 247	0x06	見附錄 C Modbus 位址對應	設定輸出值

[Response]

Slave 位址	功能碼	起始位址	資料值
1 Byte	1 Byte	2 Byte	2 Byte
1 ~ 247	0x06	同 Request 的 Byte 02-03	同 Request 的 Byte 04-05

[Error Response]

Slave 位址	功能碼	異常碼
1 Byte	1 Byte	1 Byte
1 ~ 247	0x86 (80+功能碼)	見 " 異常代碼 (Exception Code)"

範例:

請求命令 (Master): **01 06 01 E7 00 0A** (Hex.)

Slave 位址	功能碼	起始位址		資料值	
Byte 00	Byte 01	Byte 02 (H)	Byte 03 (L)	Byte 04 (H)	Byte 05 (L)
01	06	01	E7	00	0A
模組 ID = 01	寫入 Holding Register	位址 = 40487 (Base 0)		設定為 10	

註: tM 模組的 Base 位址為 0 (即 · Base 0) · 因此起始位址需設為 Modbus 位址減 1 (見附錄 C)。

回應 (Slave): **01 06 01 E7 00 0A** (Hex.)

Slave 位址	功能碼	起始位址		資料值	
Byte 00	Byte 01	Byte 02 (H)	Byte 03 (L)	Byte 04 (H)	Byte 05 (L)
01	06	01	E7	00	0A
模組 ID = 01	寫入 Holding Register	表示 Modbus Respond Delay 已設定為 10 (ms)			

D.7 15 (0x0F) Force Multiple Coils

此功能碼用來寫入多個 DO 或 線圈 (Coil) 的狀態值。

[Request]

Slave 位址	功能碼	起始位址	通道數	資料數	資料值
1 Byte	1 Byte	2 Byte	2 Byte	1 Byte	n Byte
1 ~ 247	0x0F	(見附錄 C) Byte 02 = High Byte Byte 03 = Low Byte	Byte 04 = High Byte Byte 05 = Low Byte	n = (通道數+7)/8	1 Bit 對應 1 個通道。 1: 設為 ON ; 0: 設為 OFF。 n = 1; Byte 07 = bit 7 ~ 0 n = 2; Byte 08 = bit 15 ~ 8 ... n = m; Byte m+6 = bit (8m-1) ~ 8(m-1)

[Response]

Slave 位址	功能碼	起始位址	通道數
1 Byte	1 Byte	2 Byte	2 Byte
1 ~ 247	0x0F	同 Request 的 Byte 02-03	同 Request 的 Byte 04-05

[Error Response]

Slave 位址	功能碼	異常碼
1 Byte	1 Byte	1 Byte
1 ~ 247	0x8F (80+功能碼)	見 " 異常代碼 (Exception Code)"

範例:

請求命令 (Master): **02 0F 00 03 00 05 01 1F** (Hex.)

Slave 位址	功能碼	起始位址		通道數		資料數	資料值
Byte 00	Byte 01	Byte 02	Byte 03	Byte 04	Byte 05	Byte 06	Byte 07
02	0F	00	03	00	05	01	1F

模組 ID = 02，寫入多個 DO 值。從位址 = 00003 (DO3) 開始，設定 5 個 Bit 為 ON。

註: tM 模組的 Base 位址為 0 (即，Base 0)，因此起始位址需設為 Modbus 位址減 1 (見附錄 C)。

回應 (Slave): **02 0F 00 03 00 05** (Hex.)

Slave 位址	功能碼	起始位址		通道數	
Byte 00	Byte 01	Byte 02	Byte 03	Byte 04	Byte 05
02	0F	00	03	00	05
模組 ID = 02	寫入 DO 值	於 DO3 開始		已設定 5 個 Bit	

D.8 16 (0x10) Set Multiple Register

此功能碼用來寫入多個 AO 或 保存暫存器 (Holding Registers) 的值。

[Request]

Slave 位址	功能碼	起始位址	通道數	資料數	資料值
1 Byte	1 Byte	2 Byte	2 Byte	1 Byte	n Byte
1 ~ 247	0x10	(見附錄 C) Byte 02 = High Byte Byte 03 = Low Byte	Byte 04 = High Byte Byte 05 = Low Byte	(Word 數) n = 通道數 * 2 Byte	n = m; Byte 07 = High Byte Byte 08 = Low Byte ... Byte m+5 = High Byte Byte m+6 = Low Byte

[Response]

Slave 位址	功能碼	起始位址	通道數
1 Byte	1 Byte	2 Byte	2 Byte
1 ~ 247	0x10	同 Request 的 Byte 02-03	同 Request 的 Byte 04-05

[Error Response]

Slave 位址	功能碼	異常碼
1 Byte	1 Byte	1 Byte
1 ~ 247	0x90 (80+功能碼)	見 " 異常代碼 " (Exception Code)"

範例:

請求命令 (Master): **03 10 01 C0 00 02 04 00 0A 00 0A** (Hex.)

Slave 位址	功能碼	起始位址		通道數		資料數	資料值
Byte 00	Byte 01	Byte 02	Byte 03	Byte 04	Byte 05	Byte 06	Byte 07 ~ 10
03	10	01	C0	00	02	04	000A 000A

模組 ID = 03，寫入多個 Holding Registers。

從位址 = 40448 (Base 0) 開始，設定 2 個通道的溫度偏移值為 10 (1°C)。

註: tM 模組的 Base 位址為 0 (即，Base 0)，因此起始位址需設為 Modbus 位址減 1 (見附錄 C)。

回應 (Slave): **03 10 01 C0 00 02** (Hex.)

Slave 位址	功能碼	起始位址		通道數	
Byte 00	Byte 01	Byte 02 (H)	Byte 03 (L)	Byte 04 (H)	Byte 05 (L)
03	10	01	C0	00	02

模組 ID = 03 寫入 Holding Registers 表示 AI0, 1 (tM-TH8) 的溫度偏移值已設為 1 (°C)

附錄 E DCON Protocol 指令

所有 tM 系列模組的通訊，都是經由主機發出指令 與 模組傳送回應所構成。每個模組皆有一個用來定址 並 儲存在非揮發記憶體中的唯一 ID 編號。此 ID 預設為 “01”，可使用指令來變數位址。所有模組相關的指令都包含有 ID 位址，表示只有該位址的模組會回應。但唯有 ~** 和 #** 指令例外，它們會傳送到所有的模組，在此情況下，模組並不會回覆指令。

指令格式:

前導字元	模組位址	指令	[CHKSUM]	CR
------	------	----	----------	----

回應格式:

前導字元	模組位址	回應資料	[CHKSUM]	CR
------	------	------	----------	----

前導字元	像是 ~、\$、#、@、%、!、? 或 > 等符號。
位址	模組的指定位址，可使用 DCON Utility Pro 設定 (參考第 3 章)。
指令/回應	指令參數 或 回應資料。
CHKSUM	2 字元的檢查碼，若 Checksum 設定為 “Disable” 則無此值。 (參考: E.1 節 %AANNTTCCFF 指令)
CR	指令的結尾符號 (Carriage Return, \r, 0x0D)

Checksum 算法:

1. 除了 “CR” 字元之外，將指令/回應字串中的所有字元，以 ASCII 編碼進行加總。
2. 再將總和 與 0FFh 進行遮罩，或可說是只取最後兩碼為檢查碼。

例如:

指令字串: \$012(CR)

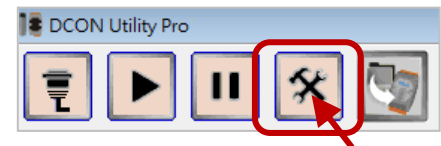
1. 字串總和 = “\$” + “0” + “1” + “2” = 24h + 30h + 31h + 32h = B7h
2. 檢查碼為 B7h，即 [CHKSUM] = “B7”
3. 指令字串與檢查碼 = \$012B7(CR)

回應字串: !01200600(CR)

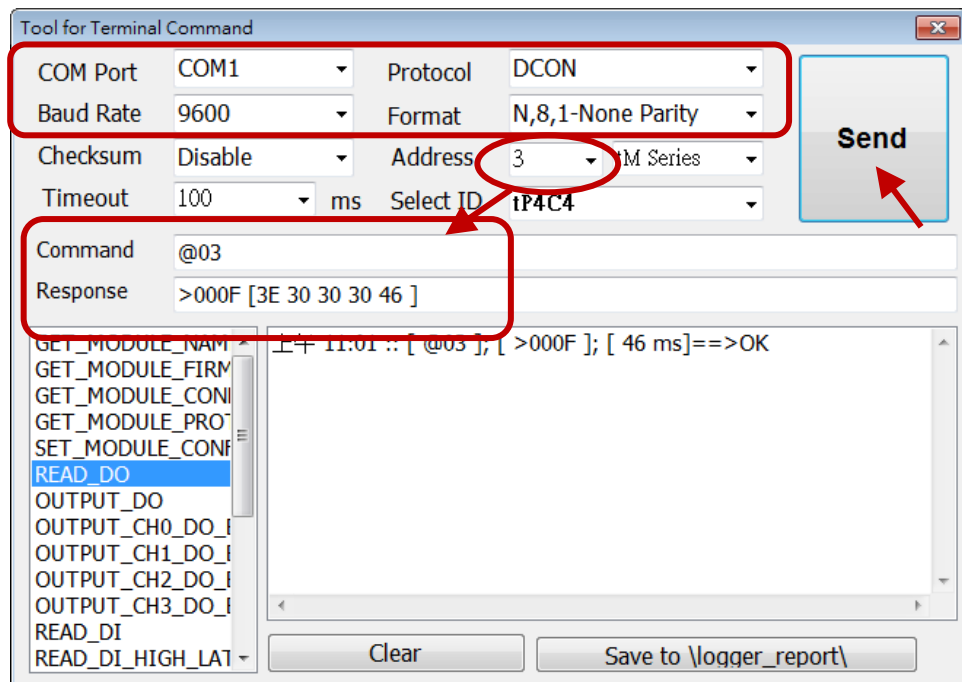
1. 字串總和 = “!” + “0” + “1” + “2” + “0” + “0” + “6” + “0” + “0”
= 21h + 30h + 31h + 32h + 30h + 30h + 36h + 30h + 30h = 1AAh
2. 檢查碼為 AAh，即 [CHKSUM] = “AA”
3. 回應字串與檢查碼 = !01200600AA(CR)

註: 所有的字元必需是大寫。

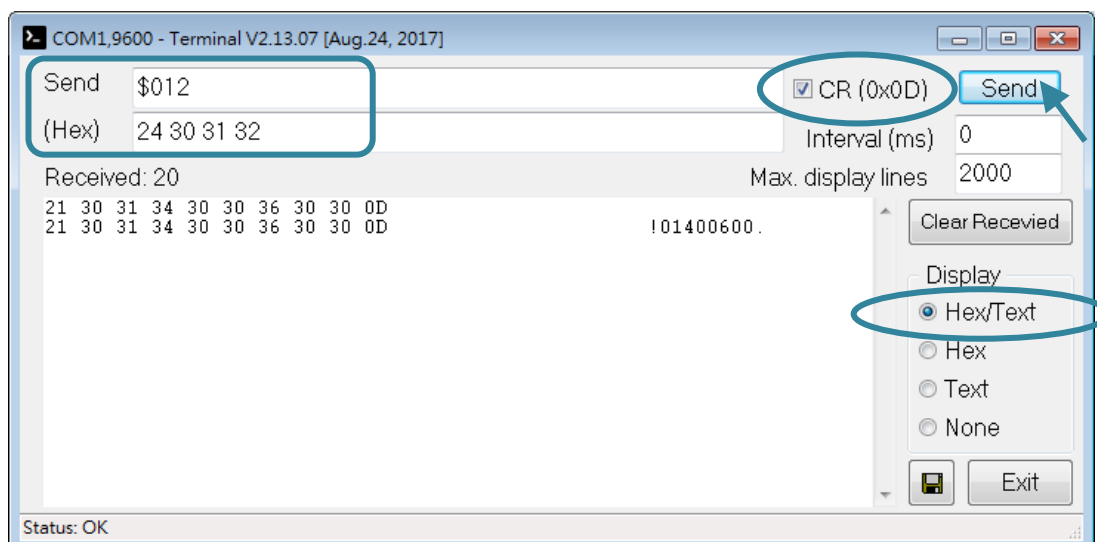
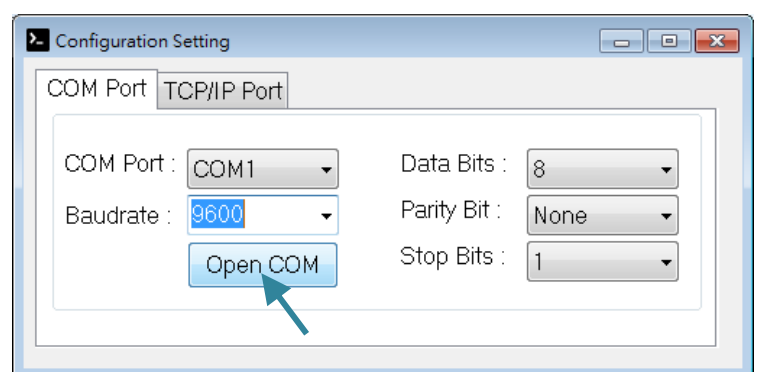
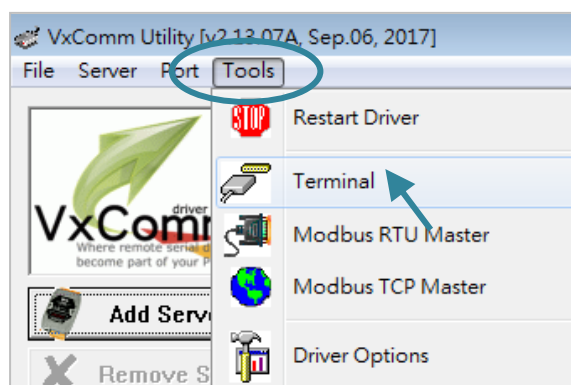
您可使用 DCON Utility Pro 的 Command Line 功能 或 VxComm Utility 的 Terminal 功能，來測試以下指令。



DCON Utility Pro



VxComm Utility



E.1 一般指令

指令	回應	說明
\$AA2	!AATCCFF	讀取模組的通訊配置
%AANNITCCFF	!AA	設定模組的通訊配置
\$AA8Ci	!AACiRrr	讀取通道的 Type Code (AI)
\$AA7CiRrr	!AA	設定通道的 Type Code (AI)
\$AA9N	!AATS	讀取通道的 Type Code 、 Slew Rate (AO)
\$AA9NTS	!AA	設定通道的 Type Code 、 Slew Rate (AO)
\$AA5	!AAS	讀取模組的重置狀態
\$AAF	!AA(Data)	讀取模組的韌體版本
\$AAM	!AA(Data)	讀取模組名稱
\$AAP	!AASC	讀取模組的通訊協定
\$AAPN	!AA	設定模組的通訊協定
\$AAI	!AAS	讀取模組的 Init 狀態
~AARD	!AAVV	讀取模組的回應延遲時間
~AARDVV	!AA	設定模組的回應延遲時間
只適用 tM-TH8 模組		
~AAI	!AA	軟體的 Init 功能
~AATnn	!AA	設定軟體 Init Timeout 時間

E.2 I/O 指令

指令	回應	說明
\$AA6	!AAVV	讀取模組的通道狀態 (AI 、 DIO) (可在此節查詢 DIO 資料回應格式)
\$AA5VV	!AA	設定模組的通道狀態 (AI)
#**	No Response	通知所有模組同步取樣 DIO 狀態
\$AA4	!S(Data)	讀取同步取樣的 DIO 狀態
@AA	>(Data)	讀取模組的通道狀態 (DIO)
@AADI	!AA00OII	讀取模組的通道狀態 (Multi-function-DIO)
@AA(Data)	>	設定模組的通道狀態 (DO)
@AADODD	!AA	設定模組的通道狀態 (Multi-function-DO)

指令	回應	說明
#AA00(Data)	>	設定模組的通道狀態 (多個, DO 0 ~ 7)
#AA0A(Data)	>	設定模組的通道狀態 (多個, DO 0 ~ 7)
#AA0B(Data)	>	設定模組的通道狀態 (多個, DO 8 ~ 15)
#AA1cDD	>	設定模組的通道狀態 (單個, DO 0 ~ 7)
#AAAcDD	>	設定模組的通道狀態 (單個, DO 0 ~ 7)
#AABcDD	>	設定模組的通道狀態 (單個, DO 8 ~ 15)
#AA	>(Data)	讀取全部的通道 (AI)
#AAN	>(Data)	讀取指定的通道 (AI)
\$AAA	>(Data)	讀取全部的通道 (AI, 16 進制值)
\$AA8N	!AA(Data)	讀取模組的 AO 值
\$AA6N	!AA(Data)	讀取模組上次寫入的 AO 值
#AAN(Data)	>	設定模組的 AO 值

E.3 Host Watchdog、Power-on Value、Safe Value 指令

指令	回應	說明
~**	No Response	通知所有模組 Host 正常
~AA0	!AASS	讀取模組的 Host Watchdog 狀態
~AA1	!AA	重置模組的 Host Watchdog 狀態
~AA2	!AAETT	讀取模組的 Host Watchdog Timeout 值
~AA3EVV	!AA	設定模組的 Host Watchdog 功能 與 Timeout 值
~AA4V	!AA(data)	讀取模組的 Power-on 或 Safe Value (DO)
~AA5V	!AA	將模組 DO 值, 設為 Power-on 或 Safe Value
~AA4	!AAPSS	讀取模組的 Power-on 與 Safe Value (Multi-function-DO)
~AA5PPSS	!AA	設定模組的 Power-on 與 Safe Value (Multi-function-DO)
\$AA7N	!AA	讀取模組的 Power-on Value (AO)
\$AA4N	!AA	將模組 AO 值, 設為 Power-on Value
~AA4N	!AA(data)	讀取模組的 Safe Value (AO)
~AA5N	!AA	將模組 AO 值, 設為 Safe Value

E.4 DI Latched、Counter 指令

指令	回應	說明
\$AALS	!AA(data)	讀取 DI Latch 狀態
\$AAC	!AA	清除 DI Latch 狀態
#AAN	!AA(data)	讀取 DI Counter
\$AACN	!AA	清除 DI Counter
適用 Multi-function 模組		
@AARECi	!AA(data)	讀取 DI Counter (適用 tM-AD4P2C2 模組)
@AAECi	!AA	清除 DI Counter (適用 tM-AD4P2C2 模組)
@AARECN	!AA(data)	讀取 DI Counter (適用 tM-DA1P1R1 模組)
@AAECN	!AA	清除 DI Counter (適用 tM-DA1P1R1 模組)

E.5 Alarm 指令

註: 下列指令只支援 tM-AD4P2C2 模組。

指令	回應	說明
@AAEATCi	!AA	啟用 Alarm 功能
@AADACi	!AA	關閉 Alarm 功能
@AARACi	!AAS	讀取 Alarm 類型設定
@AARAO	!AAHLL	讀取 High/Low Alarm 狀態
@AARHCi	!AA(data)	讀取 High Alarm Limit 值
@AARLCi	!AA(data)	讀取 Low Alarm Limit 值
@AAHI(data)Ci	!AA	設定 High Alarm Limit 值
@AALO(data)Ci	!AA	設定 Low Alarm Limit 值
@AACHCi	!AA	清除 High Latched Alarm 狀態
@AACLCi	!AA	清除 Low Latched Alarm 狀態

E.6 模組校正指令

指令	回應	說明
~AAEV	!AA	啟用/關閉校正功能 (適用所有 AI 與 tM-AD4P2C2 模組)
\$AA1	!AA	執行 Zero 校正 (V, A) (適用所有 AI 與 tM-AD4P2C2 模組)
\$AA1Ci	!AA	執行 Zero 校正 (適用 tM-AD2 模組)
\$AAO	!AA	執行 Span 校正 (V, Ω) (適用 tM-AD5, AD8, TH8, AD4P2C2 模組)
\$AAOCi	!AA	執行 Span 校正 (A) (適用 tM-AD2, AD5C, AD8C, AD4P2C2 模組)
只適用 tM-DA1P1R1 模組		
\$AAON	!AA	執行 Zero 校正
\$AA1N	!AA	執行 Span 校正
\$AA3NVV	!AA	調整 AO 通道的校正值
只適用 tM-TH8 模組		
\$AAS1	!AA	載入出廠預設的校正參數

E.7 熱敏電阻 (User-defined Type) 指令

註: 下列指令只支援 tM-TH8 模組。

指令	回應	說明
@AAA3Ci	!AA(data)	讀取通道的溫度偏移量
@AAA2CiToo	!AA	設定通道的溫度偏移量
@AAA7Ci	!AA(data)	讀取通道的電阻偏移量
@AAA6CiRrr	!AA	設定通道的電阻偏移量
~AAD	!AAT	讀取溫度單位 (°C/°F)
~AADT	!AA	設定溫度單位 (°C/°F)
@AAGxTtt	!AA(data)	讀取 Steinhart-Hart 係數 (User-defined Type)
@AASxTttC(data)	!AA	設定 Steinhart-Hart 係數 (User-defined Type)
@AARTTttR(Data)	!AA(data)	讀取電阻值的對應溫度 (User-defined Type)

\$AA2 讀取模組配置

語法:

\$AA2[CHKSUM](CR)

\$ 定義符號字元 (Delimiter Character)。

AA 模組的位址 (16 進制, 00 ~ FF)。

2 讀取模組配置的指令。

回應:

有效指令: **!AATTCCFF** [CHKSUM] (CR)

無效指令: **?AA** [CHKSUM] (CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制, 00 ~ FF)。

TT 讀取模組的 Type Code (見 [附錄 B](#))。

註: tM-AD2、tM-TH8、tM-DA1P1R1 與 tM-AD4P2C2 模組需個別設定 Type Code，
TT 會回傳 “00”；DIO 模組則會回傳 “40”。

CC 模組的 Parity 與 Baud Rate。

FF 類比模組: 模組的 Checksum、Sample Mode 與 Analog Format 設定。
數位模組: 模組的 Checksum 與 Counter 更新方向設定。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例:

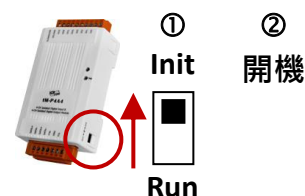
指令: \$012	讀取模組 (01) 的配置。
回應: !01<u>400</u>600	數位模組 (01) 的 Parity= N,8,1-no parity、Baud Rate=9600、 Checksum=Disable、Sample Mode=Normal、Analog Format=Engineering。
指令: \$022	讀取模組 (02) 的配置。
回應: !02<u>000</u>602	模組 (02) 的 Parity= N,8,1-no parity、Baud Rate=9600、 Checksum=Disable、Sample Mode=Normal、Analog Format= 2's Complement

相關指令: [%AANNITCCFF](#)

相關章節: 3.2 設定頁面 - Configuration

%AANNTTCCFF 設定模組的通訊配置

註: 需在 Init 模式下變更 Parity、Baud Rate 或 Checksum，並以 Run 模式重開機，套用設定。



語法:

%AANNTTCCFF[CHKSUM](CR)

% 定義符號字元 (Delimiter Character)。

AA 模組的位址 (16 進制，00 ~ FF)。

NN 設定新的模組位址 (16 進制，00 ~ FF)。

TT 設定 Type Code (見 [附錄 B](#))。

註: 若使用 tM-AD2、tM-TH8 與 tM-AD4P2C2 模組，需以 [\\$AA7CiRrr](#) 指令設定各通道的 Type Code；tM-DA1P1R1 則需使用 [\\$AA9NTS](#) 指令。此情況下，"TT" 請填 "00"。
若使用 DIO 模組，"TT" = "40"。

CC 設定 Parity 與 Baud Rate。

Bit	7	6	5	4	3	2	1	0
[CC]	Parity		Baud Rate					
Parity (需在 Init 模式)	0		1		2		3	
	N,8,1-no parity		N,8,2-no parity		E,8,1-even parity		O,8,1-odd parity	
Baud Rate (需在 Init 模式)	3	4	5	6	7	8	9	A
	1200	2400	4800	9600	19200	38400	57600	115200

FF **類比模組:** 設定 Checksum、取樣模式 (Sample Mode) 與 資料格式 (Analog Format)。

Bit	7	6	5	4	3	2	1	0
[FF]	R	C	M	(Reserved, 0)			F	
<u>C</u> hecksum (需在 Init 模式)	0: 不啟用 (Disable) 1: 啟用 (Enable)							
Sa <u>M</u> ple <u>M</u> ode	0: Normal Mode (14 bits) 1: Fast Mode (12 bits)							
Analog <u>F</u> ormat	0: Engineering 1: Percent (% of FSR)				2: 2's Complement 3: Ohms			

註: 若使用 tM-TH8、tM-DA1P1R1 時，Bit 5 為保留位元 (0)。

數位模組：設定 Counter 更新方向 與 Checksum 。

Bit	7	6	5	4	3	2	1	0
[FF]	CU	CS	保留用 (Reserved, 0)			CD		
<u>C</u> ounter <u>U</u> pdate	0: 輸入訊號為下降邊緣時，更新計數器 1: 輸入訊號為上升邊緣時，更新計數器							
<u>C</u> heck <u>s</u> um (需在 Init 模式)	0: 不啟用 (Disable) 1: 啟用 (Enable)							
<u>C</u> ode	tM-P4C4: 1 (唯讀) 其他模組可變更 Code 值，且預設為 "0"							

回應：

有效指令: **!AA** [CHKSUM] (CR)

無效指令: **?AA** [CHKSUM] (CR)

! 有效指令的定義符號。

? 無效指令的定義符號。若沒有在 Init 模式下變更 **Baud Rate** 或 **Checksum** 設定，模組將會回傳無效的指令。

AA 模組位址 (16 進制，00 ~ FF)。

註：如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例：

指令: %0102<u>40</u>0600	將數位模組的位址由 01 設為 02， Parity= N,8,1-no parity、Baud Rate=9600、 Checksum=Disable、Sample Mode=Normal、Analog Format=Engineering。
回應: !02	模組 02 回傳有效指令。
指令: %0101000<u>A</u>00	將模組 (01) 的 Baud Rate 設為 115200 bps。
回應: ?01	模組回傳無效指令，因為未在 Init 模式下設定。
指令: %0101000<u>A</u>00	(Init 模式) 設定模組 (01) 的 Baud Rate 為 115200 bps。
回應: !01	模組 01 回傳有效指令。

相關指令: [\\$AA2](#)、[\\$AA7CiRrr](#)、[\\$AA9NTS](#)

相關章節: 3.2 設定頁面 - Configuration

\$AA8Ci 讀取通道的 Type Code (AI)

註: 此指令只支援 tM-AD2、tM-TH8 與 tM-AD4P2C2 模組，可個別讀取通道。

語法:

\$AA8Ci[CHKSUM](CR)

\$ 定義符號字元 (Delimiter Character)。

AA 模組的位址 (16 進制，00 ~ FF)。

8 讀取通道 Type Code 的指令。

Ci i，表示欲讀取的 AI 通道 (0 ~ 7)。

回應:

有效指令: **!AACiRrr**[CHKSUM] (CR)

無效指令: **?AA** [CHKSUM] (CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制，00 ~ FF)。

Ci i，表示欲讀取的 AI 通道 (0 ~ 7)。

Rrr rr，表示指定通道的 Type Code。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例:

指令: \$018C0	讀取模組 (01，例如: tM-AD2) 通道 0 的 Type Code。
回應: !01C0R08	通道 0 的 Type Code = 08 (0 ~ 10 V)。
指令: \$028C3	讀取模組 (02，例如: tM-AD4P2C2) 通道 3 的 Type Code。
回應: !02C3R0D	通道 3 的 Type Code = 0D (-20 ~ +20 mA)。
指令: \$038C5	讀取模組 (03，例如: tM-TH8) 通道 5 的 Type Code。
回應: !03C5R6C	通道 5 的 Type Code = 6C (YSI H Mix 30000 @ 25°C -10°C ~ 200°C)。

相關指令: [\\$AA7CiRrr](#)、[%AANNTTCFFE](#)

相關章節:

3.3.1 設定頁面 (tM-AD2)、3.3.3 設定頁面 (tM-TH8)、3.4.3 設定頁面 (tM-AD4P2C2)

\$AA7CiRrr 設定通道的 Type Code (AI)

註: 此指令只支援 tM-AD2、tM-TH8 與 tM-AD4P2C2 模組，可個別設定通道。

語法:

\$AA7CiRrr[CHKSUM](CR)

\$ 定義符號字元 (Delimiter Character)。

AA 模組的位址 (16 進制，00 ~ FF)。

7 設定通道 Type Code 的指令。

Ci i，表示欲指定的 AI 通道 (0 ~ 7)。

Rrr rr，表示欲設定的通道 Type Code。

回應:

有效指令: **!AA** [CHKSUM] (CR)

無效指令: **?AA** [CHKSUM] (CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制，00 ~ FF)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例:

指令: \$017C0R08	設定模組 (01，例如: tM-AD2) 通道 0 的 Type Code = 08 (0 ~ 10 V)。
回應: !01	回傳有效指令。

指令: \$027C3R0D	設定模組 (02，例如: tM-AD4P2C2) 通道 3 的 Type Code = 0D (-20 ~ +20 mA)。
回應: !02	回傳有效指令。

指令: \$037C5R6C	設定模組 (03，例如: tM-TH8) 通道 5 的 Type Code = 6C (YSI H Mix 30000 @ 25°C -10°C ~ 200°C)。
回應: !03	回傳有效指令。

指令: \$037C1R30	設定模組 (03) 通道 1 的 Type Code = 30。
回應: ?03	回傳無效指令。

相關指令: [\\$AA8Ci](#)、[%AANNTTCCFF](#)

相關章節: 3.3.1 設定頁面 (tM-AD2)、3.3.3 設定頁面 (tM-TH8)、3.4.3 設定頁面 (tM-AD4P2C2)

\$AA9N 讀取通道的 Type Code 、 Slew Rate (AO)

註: 此指令只適用於 tM-DA1P1R1 。

語法:

\$AA9N[CHKSUM](CR)

~ 定義符號字元 (Delimiter Character) 。

AA 模組位址 (16 進制 , 00 ~ FF) 。

9 讀取 AO 配置的指令 。

N 欲讀取的通道 , tM-DA1P1R1 為 0 。

回應:

有效指令: !AATS[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

! 有效指令的定義符號 。

? 無效指令的定義符號 。

AA 模組位址 (16 進制 , 00 ~ FF) 。

T AO 通道的 Type Code 。(可參考附錄 B.7 tM-DA1P1R1 資料範圍)

S AO 通道的 Slew Rate 。(可參考附錄 A.4 轉換率控制)

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組 , 都會沒有回應 。

範例:

指令: \$0190	讀取模組 (01) 通道 0 的 AO 配置 。
回應: !0110	回傳 Type Code = 1 (4 ~ 20 mA) ; Slew Rate = 0 (Immediate) 。

相關指令:

[\\$AA9NTS](#)

相關章節:

3.4.2 設定頁面 - AO 、附錄 A.4 轉換率 (Slew Rate) 控制 、附錄 B.7 tM-DA1P1R1 資料範圍

\$AA9NTS 設定通道的 Type Code 、 Slew Rate (AO)

註: 此指令只適用於 **tM-DA1P1R1** 。

語法:

\$AA9NTS[CHKSUM](CR)

- ~ 定義符號字元 (Delimiter Character) 。
- AA** 模組位址 (16 進制 , 00 ~ FF) 。
- 9** 設定 AO 配置的指令 。
- N** 欲設定的通道 , tM-DA1P1R1 為 0 。
- T** AO 通道的 Type Code 。(可參考附錄 B.7 tM-DA1P1R1 資料範圍)
- S** AO 通道的 Slew Rate 。(可參考附錄 A.4 轉換率控制)

回應:

有效指令: **!AA**[CHKSUM](CR)

無效指令: **?AA**[CHKSUM](CR)

- !** 有效指令的定義符號 。
- ?** 無效指令的定義符號 。
- AA** 模組位址 (16 進制 , 00 ~ FF) 。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組 , 都會沒有回應 。

範例:

指令: \$019021	設定 AO 模組 (01) 通道 0 的 Type Code = 2 (0 ~ 10 V) ; Slew Rate = 1 (0.625 V/sec) 。
回應: !01	回傳有效指令 。

相關指令:

[\\$AA9N](#)

相關章節:

3.4.2 設定頁面 - AO 、附錄 A.4 轉換率 (Slew Rate) 控制 、附錄 B.7 tM-DA1P1R1 資料範圍

\$AA5 讀取模組的重置狀態

語法:

\$AA5[CHKSUM](CR)

\$ 定義符號字元 (Delimiter Character)。

AA 模組的位址 (16 進制, 00 ~ FF)。

5 讀取模組重置狀態的指令。

回應:

有效指令: **!AAS** [CHKSUM] (CR)

無效指令: **?AA** [CHKSUM] (CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制, 00 ~ FF)。

S 用來判斷模組是否曾重開機?

開機後, 首次執行指令會讀到 1, 之後會讀到 0, 若再次讀到 1, 表示模組有重開機。

1: 模組已重開機 (或剛開機)。

0: 模組開機後, 尚未重開機。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

範例:

指令: \$015	讀取模組 (01) 的重置狀態。
回應: !011	此回應表示模組剛開機 或 重開機了。

指令: \$015	讀取模組 (01) 的重置狀態。
回應: !010	此回應表示模組開機後, 尚未重開機。

\$AAF 讀取模組的韌體版本

語法:

\$AAF[CHKSUM](CR)

- \$ 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制，00 ~ FF)。
- F 讀取韌體版本的指令。

回應:

有效指令: !AA(Data)[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

- ! 有效指令的定義符號。
- ? 無效指令的定義符號。
- AA 模組位址 (16 進制，00 ~ FF)。
- (Data) 表示模組韌體版本的字串。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例:

指令: \$01F	讀取模組 (01) 的韌體版本。
回應: !01A2.0	表示 A2.0 版。

\$AAM 讀取模組名稱

語法:

\$AAM[CHKSUM](CR)

\$ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制, 00 ~ FF)。

M 讀取模組名稱的指令。

回應:

有效指令: **!AA(Name)**[CHKSUM](CR)

無效指令: **?AA**[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制, 00 ~ FF)。

(Name) 表示模組名稱。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

範例:

指令: \$03M	讀取模組 (03) 的名稱。
回應: !03tTH8	表示 "tM-TH8" 模組。

指令: \$02M	讀取模組 (02) 的名稱。
回應: !02tAD4P2C2	表示 "tM-AD4P2C2" 模組。

註: 使用 **tM** 系列模組時, 請勿變更模組名稱。

\$AAP 讀取模組的通訊協定

語法:

\$AAP[CHKSUM](CR)

- \$** 定義符號字元 (Delimiter Character)。
- AA** 模組位址 (16 進制, 00 ~ FF)。
- P** 讀取通訊協定的指令。

回應:

有效指令: **!AASC**[CHKSUM](CR)

無效指令: **?AA**[CHKSUM](CR)

- !** 有效指令的定義符號。
- ?** 無效指令的定義符號。
- AA** 模組位址 (16 進制, 00 ~ FF)。
- S** 模組支援的指令。
 - 0: 只支援 DCON 通訊協定
 - 1: 支援 DCON 與 Modbus RTU 通訊協定
 - 3: 支援 DCON 與 Modbus RTU/ASCII 通訊協定
- C** 模組的 Protocol 設定值 (儲存在 EEPROM 中)。
 - 0: 表示儲存在 EEPROM 中的設定值為 DCON
 - 1: 表示儲存在 EEPROM 中的設定值為 Modbus RTU
 - 3: 表示儲存在 EEPROM 中的設定值為 Modbus ASCII

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

範例:

指令: \$01P	讀取模組 (01) 的通訊協定。
回應: !0130	表示模組 (01) 支援 DCON 與 Modbus RTU/ASCII 通訊協定, 且 Protocol = DCON。

指令: \$00P	在 Init 模式下 (AA = 00, Protocol = DCON), 讀取模組的通訊協定。
回應: !0233	模組 02, 支援 DCON 與 Modbus RTU/ASCII 通訊協定, 且 Protocol = Modbus ASCII (儲存在 EEPROM 中的設定值, 請以 Run 模式重開機, 套用設定)。

相關指令: [\\$AAPN](#)

相關章節: 3.2 設定頁面 - Configuration

\$AAPN 設定模組的通訊協定

註： 需在 Init 模式下設定 Protocol。

語法：

\$AAPN[CHKSUM](CR)

\$ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制，00 ~ FF)。

P 設定通訊協定的指令。

N 0: 表示 DCON 通訊協定

1: 表示 Modbus RTU 通訊協定

3: 表示 Modbus ASCII 通訊協定

回應：

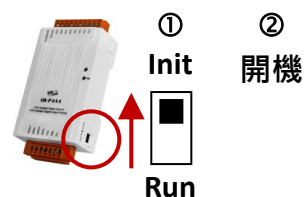
有效指令: **!AA**[CHKSUM](CR)

無效指令: **?AA**[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制，00 ~ FF)。



註： 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

註： 在 Init 模式下 (**AA = 00**，Protocol = DCON) 設定 Protocol 後，設定值會儲存在 EEPROM 中，需以 Run 模式重開機，才會套用設定。

範例：

指令: \$00P1	在 Init 模式下，設定模組的 Protocol = Modbus RTU。
回應: ?01	回傳無效指令，因為模組未在 Init 模式下設定。
指令: \$00P1	在 Init 模式下，設定模組的 Protocol = Modbus RTU。
回應: !02	模組 02 的指令有效。
指令: \$00P	讀取模組的通訊協定。
回應: !0231	模組 02，支援 DCON 與 Modbus RTU/ASCII 通訊協定，且 Protocol = Modbus RTU (儲存在 EEPROM 中的設定值，請以 Run 模式重開機，套用設定)。

相關指令: [\\$AAP](#)

相關章節: 3.2 設定頁面 - Configuration

\$AAI 讀取模組的 Init 狀態

語法:

\$AAI[CHKSUM](CR)

\$ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制 · 00 ~ FF)。

I 讀取模組 Init 狀態的指令。

回應:

有效指令: **!AAS**[CHKSUM](CR)

無效指令: **?AA**[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制 · 00 ~ FF)。

S 模組的 Init 狀態。

0: 指撥開關在 “Init” 的位置

1: 指撥開關在 “Run” 的位置

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例:

指令: \$01I	讀取模組 (01) 的 Init 狀態。
回應: !010	模組右邊的指撥開關在 “Init” 的位置。

~AARD 讀取模組的回應延遲時間

語法:

~AARD[CHKSUM](CR)

~ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制, 00 ~ FF)。

RD 讀取模組回應延遲時間的指令。

回應:

有效指令: !AAVV[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制, 00 ~ FF)。

VV 用 2 位數 16 進制值表示回應延遲時間 (單位: ms)。

例如: 01 表示 1 ms 與 1E 表示 30 ms。最大 30 ms。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

範例:

指令: ~01RD	讀取模組 (01) 的回應延遲時間。
回應: !0102	回傳 02, 表示回應延遲時間為 2 ms。

相關指令:

[~AARDVV](#)

相關章節:

3.2.2 ~ 3.2.4 設定類比項目

~AARDVV 設定模組的回應延遲時間

語法:

~AARDVV[CHKSUM](CR)

~ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制, 00 ~ FF)。

RD 設定模組回應延遲時間的指令。

VV 用 2 位數 16 進制值表示回應延遲時間 (單位: ms)。

例如: 01 表示 1 ms 與 1E 表示 30 ms。最大 30 ms。

回應:

有效指令: !AA[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制, 00 ~ FF)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

範例:

指令: ~01RD06	設定模組 (01) 的回應延遲時間為 6 ms。
回應: !01	模組回傳有效指令。

指令: ~01RD	讀取模組 (01) 的回應延遲時間。
回應: !0106	模組回傳 06, 表示回應延遲時間為 6 ms。

相關指令:

[~AARD](#)

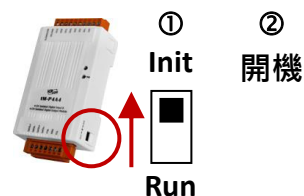
相關章節:

3.2.2 ~ 3.2.4 設定類比項目

~AAI 軟體 Init 功能

說明:

一般需在 Init 模式下，設定模組的 Parity、Baud Rate 或 Checksum。
但您仍可使用軟體 Init 的方式進行設定。



註: 此指令只適用於 tM-TH8。請先以 [~AATnn](#) 指令設定好 Timeout 時間。

語法:

~AAI[CHKSUM](CR)

~ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制，00 ~ FF)。

I 設定軟體 Init 的指令。

回應:

有效指令: **!AA[CHKSUM](CR)**

無效指令: **?AA[CHKSUM](CR)**

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制，00 ~ FF)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例:

指令: ~01T10	設定模組 (01) 狀態為軟體 Init Timeout = 16 秒。
回應: !01	模組回傳有效指令。

指令: ~01I	設定模組 (01) 狀態為軟體 Init。
回應: !01	模組回傳有效指令。

指令: %0101000 <u>7</u> 00	設定模組 (01) Parity = N,8,1-no parity、Baud Rate = 19200、Checksum = 0、Analog Format = Engineering。
回應: !01	模組回傳有效指令。

相關指令: [~AATnn](#)、[%AANNTTCCFF](#)

相關章節: 2.5 操作模式開關 與 通訊參數

~AATnn 設定軟體 Init Timeout 時間

註: 此指令只適用於 tM-TH8。

語法:

~AATnn[CHKSUM](CR)

~ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制, 00 ~ FF)。

T 設定軟體 Init Timeout 的指令。

nn 用 2 位數 16 進制值, 表示 Timeout 時間 (單位: 秒, 最多為 60 秒)。

註: 傳送 ~AAI (軟體 Init) 指令後, 需在 Init Timeout (值=0) 前使用 %AANNTTCFF 指令, 否則 Parity、Baud Rate 或 Checksum 的設定會無效。此外, 模組重開機後, Init Timeout = 0。

回應:

有效指令: !AA[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制, 00 ~ FF)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

範例:

指令: ~01T10	設定模組 (01) 狀態為軟體 Init Timeout = 16 秒。
回應: !01	模組回傳有效指令。
指令: ~01I	設定模組 (01) 狀態為軟體 Init。
回應: !01	模組回傳有效指令。
指令: %0101000 <u>7</u> 00	設定模組 (01) Parity = N, 8, 1-no parity、Baud Rate = 19200、Checksum = 0、Analog Format = Engineering。
	註: 完成此設定後, 建議將 Init Timeout 設為 0, 取消 Soft INIT 狀態。
回應: !01	模組回傳有效指令。

相關指令: [~AAI](#)、[%AANNTTCFF](#)

相關章節: 2.5 操作模式開關 與 通訊參數

\$AA6 讀取模組的通道狀態 (AI、DIO)

說明:

- 讀取每個通道的 啟用/關閉 狀態。[適用 AI 模組、tM-AD4P2C2]
- 讀取每個通道的 ON/OFF 狀態。[適用 DIO 模組、tM-DA1P1R1]

語法:

\$AA6[CHKSUM](CR)

\$ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制，00 ~ FF)。

6 讀取通道狀態的指令。

回應:

有效指令: [適用 AI 模組] !AAVV[CHKSUM](CR)

[適用 DIO 模組] !(Data)[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制，00 ~ FF)。

VV [適用 AI 模組] 用 2 位數 16 進制值，Bit 0 對應到通道 0，Bit 1 對應到通道 1...等等。
當 Bit 為 1，表示啟用該通道；Bit 為 0，表示關閉該通道。

(Data) [適用 DIO 模組] 用 4 位數 16 進制值，再加上 00。

DIO 資料回應格式:

1) \$AA4、\$AA6 與 \$AALS 指令: [第 1 組資料][第 2 組資料]00

2) @AA 指令: [第 1 組資料][第 2 組資料]

模組名稱	DIO	第 1 組資料	第 2 組資料
tM-DA1P1R1	1 DO ; 1 DI	DO 0 (00 ~ 01)	DI0 (00 ~ 01)
tM-P3R3	3 DO ; 3 DI	DO 0 ~ 2 (00 ~ 07)	DI0 ~ 2 (00 ~ 07)
tM-P3POR3			
tM-P4C4	4 DO ; 4 DI	DO 0 ~ 3 (00 ~ 0F)	DI0 ~ 3 (00 ~ 0F)
tM-P4A4			
tM-R5	5 DO	DO 0 ~ 4 (00 ~ 1F)	00
tM-C8	8 DO	DO 0 ~ 7 (00 ~ FF)	
tM-P8	8 DI	DI 0 ~ 7 (00 ~ FF)	

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例:

[適用 AI 模組、tM-AD4P2C2]

指令： \$016	讀取 AI 模組 (01，例如: tM-AD8) 的啟用/關閉狀態。																								
回應： !013F	AI0 ~ AI5 是啟用的；AI6 ~ AI7 是關閉的。																								
	<table><tr><td>AI7</td><td>AI6</td><td>AI5</td><td>AI4</td><td>AI3</td><td>AI2</td><td>AI1</td><td>AI0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td colspan="4">3</td><td colspan="4">F</td></tr></table>	AI7	AI6	AI5	AI4	AI3	AI2	AI1	AI0	0	0	1	1	1	1	1	1	3				F			
	AI7	AI6	AI5	AI4	AI3	AI2	AI1	AI0																	
	0	0	1	1	1	1	1	1																	
3				F																					

指令: \$026	讀取 AI 模組 (02，例如: tM-AD4P2C2) 的啟用/關閉狀態。												
回應: !020A	<p>AI1 與 AI3 是啟用的；AI0 與 AI2 是關閉的。</p> <table><tr><th>AI3</th><th>AI2</th><th>AI1</th><th>AI0</th></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td colspan="4">0A</td></tr></table>	AI3	AI2	AI1	AI0	1	0	1	0	0A			
AI3	AI2	AI1	AI0										
1	0	1	0										
0A													

[適用 DIO 模組、tM-DA1P1R1]

指令: \$016	讀取 DI/DO 模組 (01，例如: tM-DA1P1R1) 的 ON/OFF 狀態。				
回應: !010100	<p>DO0、DI0 為 ON。</p> <table><tr><th>DO0</th><th>DI0</th></tr><tr><td>1</td><td>1</td></tr></table>	DO0	DI0	1	1
DO0	DI0				
1	1				

指令: \$026	讀取 DI/DO 模組 (02，例如: tM-C8) 的 ON/OFF 狀態。																								
回應: !D50000	DO 0、2、4、6 為 ON；DO 1、3、5、7 為 OFF。																								
	<table><tr><td>DO7</td><td>DO6</td><td>DO5</td><td>DO4</td><td>DO3</td><td>DO2</td><td>DO1</td><td>DO0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td colspan="4">D</td><td colspan="4">5</td></tr></table>	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0	1	1	0	1	0	1	0	1	D				5			
	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0																	
	1	1	0	1	0	1	0	1																	
D				5																					

相關指令: [@AA](#)、[\\$AA5VV](#)

相關章節:

3.3 設定頁面 – AI

3.4.3 設定頁面 – AI/DO Alarm

3.4.4 設定頁面 – DO

3.4.6 設定頁面 – DI

\$AA5VV 設定模組的通道狀態 (AI)

說明:

啟用指定的通道。

註: 此指令只支援 AI 與 tM-AD4P2C2 模組，建議只啟用 (Enable) 需使用的通道。

語法:

\$AA5VV[CHKSUM](CR)

\$ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制，00 ~ FF)。

5 設定通道狀態的指令。

VV 用 2 位數 16 進制值，Bit 0 對應到通道 0，Bit 1 對應到通道 1...等等。
當 Bit 為 1，表示啟用該通道；Bit 為 0，表示關閉該通道。

回應:

有效指令: !AA[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制，00 ~ FF)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

指令: \$0151 <u>A</u>	啟用模組 (01 例如: tM-AD5) 的 AI1、AI3 與 AI4 通道。 <table><tr><th>AI4</th><th>AI3</th><th>AI2</th><th>AI1</th><th>AI0</th></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr></table>	AI4	AI3	AI2	AI1	AI0	1	1	0	1	0
AI4	AI3	AI2	AI1	AI0							
1	1	0	1	0							
回應: !01	回傳有效指令。										

指令: \$016	讀取模組 (01) 的啟用/關閉狀態。
回應: !011A	AI1、AI3 與 AI4 是啟用的。

相關指令: [\\$AA6](#)

相關章節:

3.3 設定頁面 - AI、3.4.3 設定頁面 – AI/DO Alarm

*** 通知所有模組同步取樣 DIO 狀態

說明:

在同一 RS-485 Bus 上，有同樣通訊設定的 DIO 模組都會收到此指令，並且會記錄當時的 DIO 狀態。

- 註:**
1. 此指令支援 DIO 與 Multi-function (tM-DA1P1R1、tM-AD4P2C2) 模組。
 2. 需搭配 [\\$AA4](#) 指令使用，可使用 VxComm Utility 測試指令。

語法:

***[CHKSUM](CR)

定義符號字元 (Delimiter Character)。

** 同步取樣的指令。

回應:

無回應。

範例:

指令: ***	通知所有模組同步讀取並儲存 DIO 資料。
回應:	無回應。

指令: \$014	讀取同步取樣的模組 (01，例如: tM-P8) 資料。																
回應: !<u>1C30000</u>	<table><tr><td>DI7</td><td>DI6</td><td>DI5</td><td>DI4</td><td>DI3</td><td>DI2</td><td>DI1</td><td>DI0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr></table>	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0	1	1	0	0	0	0	1	1
DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0										
1	1	0	0	0	0	1	1										

指令: \$024	讀取模組 (02，例如: tM-AD4P2C2) 的同步取樣資料。								
回應: !<u>1020000</u>	<table><tr><td>DO1</td><td>DO0</td><td>DI1</td><td>DI0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td></tr></table>	DO1	DO0	DI1	DI0	1	0	0	0
DO1	DO0	DI1	DI0						
1	0	0	0						
指令: \$024	讀取同步取樣的模組 (02，例如: tM-AD4P2C2) 資料。								
回應: !<u>0020000</u>	0，表示非第一次讀取取樣資料。 即，非最近的取樣資料，可再執行 *** 進行最新的取樣。								

相關指令: [\\$AA4](#)

相關章節: [\\$AA6 指令 – DIO 資料回應格式表](#)

\$AA4 讀取同步取樣的 DIO 狀態

- 註:**
1. 此指令支援 DIO 與 Multi-function (tM-DA1P1R1、tM-AD4P2C2) 模組。
 2. 需搭配 [#**](#) 指令使用，可使用 VxComm Utility (Terminal) 測試指令。

語法:

\$AA4[CHKSUM](CR)

- \$ 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制，00 ~ FF)。
- 4 讀取同步取樣資料的指令。

回應:

有效指令: **!S(Data)[CHKSUM](CR)**

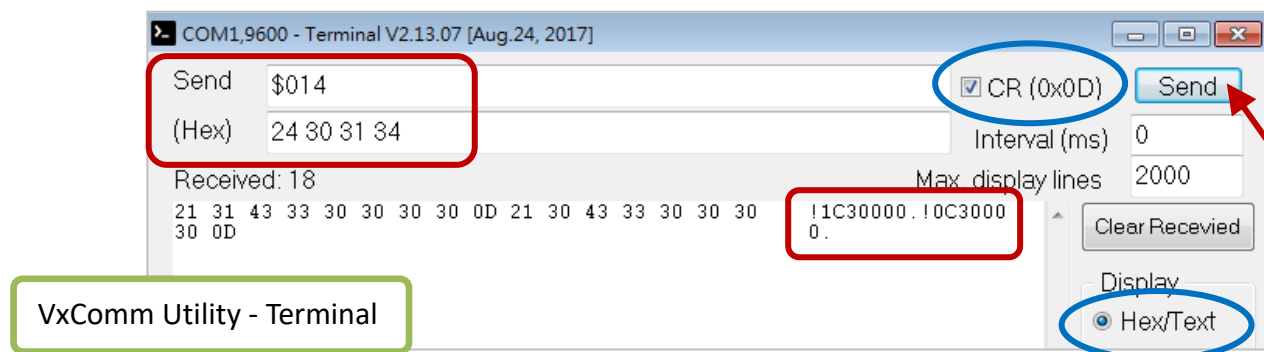
無效指令: **?AA[CHKSUM](CR)**

- ! 有效指令的定義符號。
- ? 無效指令的定義符號。
- AA 模組位址 (16 進制，00 ~ FF)。
- S 同步資料的狀態。
- 1: 取樣後，第一次讀取模組資料。
- 0: 非第一次讀取資料，即非最近取樣資料，可再執行 [#**](#) 進行取樣。

範例:

指令: #**	通知所有模組讀取 DIO 狀態並記錄。							
回應:	無回應。							

指令: \$014	讀取同步取樣的模組 (01，例如: tM-P8) 資料。							
回應: ! <u>1C30000</u>	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
	1	1	0	0	0	0	1	1



相關指令: [#**](#)

相關章節: [\\$AA6 指令 – DIO 資料回應格式表](#)

@AA 讀取模組的通道狀態 (DIO)

說明:

讀取每個通道的 ON/OFF 狀態。

註: 此指令支援 DIO 與 Multi-function (tM-DA1P1R1、tM-AD4P2C2) 模組。

語法:

@AA[CHKSUM](CR)

@ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制, 00 ~ FF)。

回應:

有效指令: >(Data)[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

> 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制, 00 ~ FF)。

(Data) 用 4 位數 16 進制值, 表示 DIO 狀態。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

指令: @01	讀取模組 (01 , 例如: tM-DA1P1R1) 的通道狀態。																		
回應: >0101	DO0、DI0 為 ON。 <table><tr><td>DO0</td><td>DI0</td></tr><tr><td>1</td><td>1</td></tr></table>	DO0	DI0	1	1														
DO0	DI0																		
1	1																		
指令: @02	讀取模組 (02 , 例如: tM-AD4P2C2) 的通道狀態。																		
回應: >0203	DO1、DI0 與 DI1 為 ON ; DO0 為 OFF。 <table><tr><td>DO1</td><td>DO0</td><td>DI1</td><td>DI0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td></tr></table>	DO1	DO0	DI1	DI0	1	0	1	1										
DO1	DO0	DI1	DI0																
1	0	1	1																
指令: @03	讀取模組 (03 , 例如: tM-P3R3) 的通道狀態。																		
回應: >0207	DO1、DI0 ~ DI2 為 ON ; DO0、DO2 為 OFF。 <table><tr><td>DO2</td><td>DO1</td><td>DO0</td><td>DI2</td><td>DI1</td><td>DI0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td colspan="3">02</td><td colspan="3">07</td></tr></table>	DO2	DO1	DO0	DI2	DI1	DI0	0	1	0	1	1	1	02			07		
DO2	DO1	DO0	DI2	DI1	DI0														
0	1	0	1	1	1														
02			07																

相關指令: [\\$AA6](#)

相關章節: 3.4.4 設定頁面 - DO、3.4.6 設定頁面 - DI

@AADI 讀取模組的通道狀態 (Multi-function-DIO)

說明:

讀取模組的 DI 與 DO 狀態。

註: 此指令只支援 tM-DA1P1R1 與 tM-AD4P2C2 模組。

語法:

@AADI[CHKSUM](CR)

- @ 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制，00 ~ FF)。
- DI 讀取 DI 與 DO 狀態的指令。

回應:

有效指令: !AA000II[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

- ! 有效指令的定義符號。
- ? 無效指令的定義符號。
- AA 模組位址 (16 進制，00 ~ FF)。
- OO 用 2 位數 16 進制值，表示 DO 狀態。
- II 用 2 位數 16 進制值，表示 DI 狀態。
- Bit 0 對應到 DI/DO 0，Bit 1 對應到 DI/DO 1...等等。
- 當 Bit 為 1，表示 DI/DO 通道為 ON；Bit 為 0，表示 DI/DO 通道為 OFF。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

指令: @01DI	讀取模組 (01，例如: tM-DA1P1R1) 的通道狀態。								
回應: !0100 <u>101</u>	DO0、DI0 為 ON。 <table><tr><td>DO0</td><td>DI0</td></tr><tr><td>1</td><td>1</td></tr></table>	DO0	DI0	1	1				
DO0	DI0								
1	1								
指令: @02DI	讀取模組 (02，例如: tM-AD4P2C2) 的通道狀態。								
回應: !0200 <u>203</u>	DO1、DI0 與 DI1 為 ON；DO0 為 OFF。 <table><tr><td>DO1</td><td>DO0</td><td>DI1</td><td>DI0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td></tr></table>	DO1	DO0	DI1	DI0	1	0	1	1
DO1	DO0	DI1	DI0						
1	0	1	1						

相關指令: @AADODD

@AA(Data) 設定模組的通道狀態 (DO)

註：此指令支援 DO 與 Multi-function (tM-DA1P1R1、tM-AD4P2C2) 模組。

語法：

@AA(Data)[CHKSUM](CR)

@ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制，00 ~ FF)。

(Data) 欲寫入 DO 模組的資料。

DIO 模組 (例如: tM-P4C4) 使用 1 位數 16 進制值，表示 DO 狀態。

DO 模組 (例如: tM-C8) 使用 2 位數 16 進制值，表示 DO 狀態。

Bit 0 對應到 DO0，Bit 1 對應到 DO1...等等。

當 Bit 為 1，表示 DO 通道為 ON；Bit 為 0，表示 DO 通道為 OFF。

回應：

有效指令: >[CHKSUM](CR)

無效指令: ?[CHKSUM](CR)

略過指令: ![CHKSUM](CR)

> 有效指令的定義符號。

? 無效指令的定義符號。

! 忽略指令的定義符號。

當 Host Watchdog Timeout 時，DO 會設為 Safe Value，並忽略 DO 收到的值。

註：如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

指令: @01 <u>7</u>	將模組 (01，例如: tM-P4C4) 的 DO0 ~ DO2 設為 ON，DO3 設為 OFF。 <table><tr><th>DO3</th><th>DO2</th><th>DO1</th><th>DO0</th></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td></tr></table>	DO3	DO2	DO1	DO0	0	1	1	1
DO3	DO2	DO1	DO0						
0	1	1	1						
回應: >	回傳有效指令。								

指令: @03 <u>07</u>	讀取模組 (03，例如: tM-C8) 的 DO0 ~ DO2 設為 ON，其他設為 OFF。							
	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0
	0	0	0	0	0	1	1	1
回應: >	回傳有效指令。							

相關指令: [@AA](#)

相關章節: 3.4.4 設定頁面 - DO

@AADODD 設定模組的通道狀態 (Multi-function-DO)

說明:

設定模組的 DO 狀態。

註: 此指令只支援 tM-DA1P1R1 與 tM-AD4P2C2 模組。

語法:

@AADODD[CHKSUM](CR)

- @ 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制, 00 ~ FF)。
- DO 設定 DO 狀態的指令。
- DD 用 2 位數 16 進制值, Bit 0 對應到 DO0, Bit 1 對應到 DO1...等等。
當 Bit 為 1, 表示 DO 通道為 ON; Bit 為 0, 表示 DO 通道為 OFF。

回應:

有效指令: !AA[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

- ! 有效指令的定義符號。
- ? 無效指令的定義符號。
- AA 模組位址 (16 進制, 00 ~ FF)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

指令: @01DO01	將模組 (01，例如: tM-DA1P1R1) 的 DO0 設為 ON。 <table><tr><td>DO0</td></tr><tr><td>1</td></tr></table>	DO0	1
DO0			
1			
回應: !01	回傳有效指令。		

指令: @02DO03	將模組 (02, 例如: tM-AD4P2C2) DO0、DO1 設為 ON。 <table><tr><th>DO1</th><th>DO0</th></tr><tr><td>1</td><td>1</td></tr></table>	DO1	DO0	1	1
DO1	DO0				
1	1				
回應: !02	回傳有效指令。				

相關指令: [@AADI](#)

相關章節: 3.4.3 設定頁面 – AI/DO Alarm

#AA00(Data) 設定模組的通道狀態 (多個, DO 0 ~ 7)

註: 此指令只支援 DO 模組, 且與 #AA0A(Data) 指令一樣。

語法:

#AA00(Data)[CHKSUM](CR)

定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制, 00 ~ FF)。

00 若模組通道數大於 8, 只會設定低位通道的 DO0 ~ DO7。

(Data) 用 2 位數 16 進制值, Bit 0 對應到 DO0, Bit 1 對應到 DO1...等等。

當 Bit 為 1, 表示 DO 通道為 ON; Bit 為 0, 表示 DO 通道為 OFF。

回應:

有效指令: >[CHKSUM](CR)

無效指令: ?[CHKSUM](CR)

略過指令: ![CHKSUM](CR)

> 有效指令的定義符號。

? 無效指令的定義符號。

! 忽略指令的定義符號。

當 Host Watchdog Timeout 時, DO 會設為 Safe Value, 並忽略 DO 收到的值。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

指令: #010005	將模組 (01, 例如: tM-P4A4) 的 DO0、2 設為 ON, DO1、3 設為 OFF。			
	DO3	DO2	DO1	DO0
	0	1	0	1
回應: >	回傳有效指令。			

指令: #030033	將模組 (03, 例如: tM-C8) 的 DO0、1、4、5 設為 ON, DO2、3、6、7 設為 OFF。							
	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0
	0	0	1	1	0	0	1	1
回應: >	回傳有效指令。							

相關指令: [#AA0A\(Data\)](#)、[#AA0B\(data\)](#)、[#AAAcDD](#)、[#AABcDD](#)、[\\$AA6](#)、[@AA](#)

#AA0A(Data) 設定模組的通道狀態 (多個, DO 0 ~ 7)

註: 此指令只支援 DO 模組, 且與 #AA00(Data) 指令一樣。

語法:

#AA0A(Data)[CHKSUM](CR)

定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制, 00 ~ FF)。

0A 若模組通道數大於 8, 只會設定低位通道的 DO0 ~ DO7。

(Data) 用 2 位數 16 進制值, Bit 0 對應到 DO0, Bit 1 對應到 DO1...等等。

當 Bit 為 1, 表示 DO 通道為 ON; Bit 為 0, 表示 DO 通道為 OFF。

回應:

有效指令: >[CHKSUM](CR)

無效指令: ?[CHKSUM](CR)

略過指令: ![CHKSUM](CR)

> 有效指令的定義符號。

? 無效指令的定義符號。

! 忽略指令的定義符號。

當 Host Watchdog Timeout 時, DO 會設為 Safe Value, 並忽略 DO 收到的值。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

指令: #010A06	將模組 (01, 例如: tM-P3POR3) 的 DO1、2 設為 ON, DO0 設為 OFF。		
	DO2	DO1	DO0
	1	1	0
回應: >	回傳有效指令。		

指令: #030A33	將模組 (03, 例如: tM-C8) 的 DO0、1、4、5 設為 ON, DO2、3、6、7 設為 OFF。							
	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0
	0	0	1	1	0	0	1	1
回應: >	回傳有效指令。							

相關指令: [#AA00\(Data\)](#)、[#AA0B\(data\)](#)、[#AA1cDD](#)、[#AAAcDD](#)、[#AABcDD](#)、[\\$AA6](#)、[@AA](#)

#AA0B(Data) 設定模組的通道狀態 (多個, DO 8 ~ 15)

註: 此指令只支援 DO 模組。tM 系列模組, 目前最多支援 8 個 DO 通道 (0 ~ 7)。

語法:

#AA0B(Data)[CHKSUM](CR)

定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制, 00 ~ FF)。

0B 若模組通道數大於 8, 只會設定高位通道的 DO8 ~ DO15。

(Data) 用 2 位數 16 進制值, Bit 0 對應到 DO8, Bit 1 對應到 DO9...等等。

當 Bit 為 1, 表示 DO 通道為 ON; Bit 為 0, 表示 DO 通道為 OFF。

回應:

有效指令: >[CHKSUM](CR)

無效指令: ?[CHKSUM](CR)

略過指令: ![CHKSUM](CR)

> 有效指令的定義符號。

? 無效指令的定義符號。

! 忽略指令的定義符號。

當 Host Watchdog Timeout 時, DO 會設為 Safe Value, 並忽略 DO 收到的值。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

指令: #020B33	將模組 (02) 的 DO8、9、12、13 設為 ON, DO10、11、14、15 設為 OFF。							
	DO15	DO14	DO13	DO12	DO11	DO10	DO9	DO8
	0	0	1	1	0	0	1	1
回應: >	回傳有效指令。							

相關指令:

[#AA00\(Data\)](#)、[#AA0A\(Data\)](#)、[#AA1cDD](#)、[#AAAcDD](#)、[#AABcDD](#)、[\\$AA6](#)、[@AA](#)

#AA1cDD 設定模組的通道狀態 (單個，DO 0 ~ 7)

註: 此指令只支援 DO 模組，且與 #AAAcDD 指令一樣

語法:

#AA1cDD[CHKSUM](CR)

- # 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制，00 ~ FF)。
- 1 設定單個低位通道 DO0 ~ DO7 的指令。
若模組通道數大於 8，只會設定 DO0 ~ DO7。
- c 指定 1 個通道 (0 ~ 7)。
- DD 00: 設定 DO 通道為 OFF。
01: 設定 DO 通道為 ON。

回應:

有效指令: >[CHKSUM](CR)

無效指令: ?[CHKSUM](CR)

略過指令: ![CHKSUM](CR)

- > 有效指令的定義符號。
- ? 無效指令的定義符號。
- ! 忽略指令的定義符號。

當 Host Watchdog Timeout 時，DO 會設為 Safe Value，並忽略 DO 收到的值。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

指令: #011 <u>2</u> 01	將模組 (01) 的 DO2 設為 ON。							
	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0
	0	0	0	0	0	1	0	0
回應: >	回傳有效指令。							

指令: #011 <u>4</u> 01	將模組 (01) 的 DO4 設為 ON。							
	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0
	0	0	0	1	0	1	0	0
回應: >	回傳有效指令。							

相關指令: [#AA00\(Data\)](#)、[#AA0A\(Data\)](#)、[#AA0B\(data\)](#)、[#AAAcDD](#)、[#AABcDD](#)、[\\$AA6](#)、[@AA](#)

#AAAcDD 設定模組的通道狀態 (單個，DO 0 ~ 7)

註: 此指令只支援 DO 模組，且與 #AA1cDD 指令一樣。

語法:

#AAAcDD[CHKSUM](CR)

- # 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制，00 ~ FF)。
- A 設定單個低位通道 DO0 ~ DO7 的指令。
若模組通道數大於 8，只會設定 DO0 ~ DO7。
- c 指定 1 個通道 (0 ~ 7)。
- DD 00: 設定 DO 通道為 OFF。
01: 設定 DO 通道為 ON。

回應:

有效指令: >[CHKSUM](CR)

無效指令: ?[CHKSUM](CR)

略過指令: ![CHKSUM](CR)

- > 有效指令的定義符號。
- ? 無效指令的定義符號。
- ! 忽略指令的定義符號。

當 Host Watchdog Timeout 時，DO 會設為 Safe Value，並忽略 DO 收到的值。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

指令: #01A <u>2</u> 01	將模組 (01) 的 DO2 設為 ON。 <table><tr><td>DO7</td><td>DO6</td><td>DO5</td><td>DO4</td><td>DO3</td><td>DO2</td><td>DO1</td><td>DO0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr></table>	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0	0	0	0	0	0	1	0	0
DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0										
0	0	0	0	0	1	0	0										
回應: >	回傳有效指令。																

指令: #01A <u>4</u> 01	將模組 (01) 的 DO4 設為 ON。 <table><tr><td>DO7</td><td>DO6</td><td>DO5</td><td>DO4</td><td>DO3</td><td>DO2</td><td>DO1</td><td>DO0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td></tr></table>	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0	0	0	0	1	0	1	0	0
DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0										
0	0	0	1	0	1	0	0										
回應: >	回傳有效指令。																

相關指令: [#AA00\(Data\)](#)、[#AA0A\(Data\)](#)、[#AA0B\(data\)](#)、[#AA1cDD](#)、[#AABcDD](#)、[\\$AA6](#)、[@AA](#)

#AABcDD 設定模組的通道狀態 (單個，DO 8 ~ 15)

註：此指令只支援 DO 模組。tM 系列模組，目前最多支援 8 個 DO 通道 (0 ~ 7)。

語法：

#AA0B(Data)[CHKSUM](CR)

- # 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制，00 ~ FF)。
- B 設定單個高位通道 DO8 ~ DO15 的指令。
- c 指定 1 個通道 (8 ~ 15)。
- DD 00: 設定 DO 通道為 OFF。
01: 設定 DO 通道為 ON。

回應：

有效指令: >[CHKSUM](CR)

無效指令: ?[CHKSUM](CR)

略過指令: ![CHKSUM](CR)

- > 有效指令的定義符號。
- ? 無效指令的定義符號。
- ! 忽略指令的定義符號。

當 Host Watchdog Timeout 時，DO 會設為 Safe Value，並忽略 DO 收到的值。

註：如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

指令: #020B33	將模組 (02) 的 DO8、9、12、13 設為 ON，DO10、11、14、15 設為 OFF。							
	DO15	DO14	DO13	DO12	DO11	DO10	DO9	DO8
	0	0	1	1	0	0	1	1
回應: >	回傳有效指令。							

相關指令：

[#AA00\(Data\)](#)、[#AA0A\(Data\)](#)、[#AA0B\(data\)](#)、[#AA1cDD](#)、[#AAAacDD](#)、[\\$AA6](#)、[@AA](#)

#AA 讀取全部的通道 (AI)

註: 此指令支援 AI 與 tM-AD4P2C2 模組。

語法:

#AA[CHKSUM](CR)

定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制, 00 ~ FF)。

回應:

有效指令: >(Data)[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

> 有效指令的定義符號。

? 無效指令的定義符號。

(Data) 所有 AI 通道的資料。

AA 模組位址 (16 進制, 00 ~ FF)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

範例:

指令: #01	讀取模組 (01, 例如: tM-AD2) 所有的 AI 通道。
回應: >+00.001+00.007 , 收到 Engineering 格式的資料: Ch00 = +00.001、Ch01 = +00.007。	
指令: #02	讀取模組 (02, 例如: tM-AD4P2C2) 所有的 AI 通道。
回應: >+07.389+07.389+00.002+00.002 , 收到 Engineering 格式的資料: Ch00 = +07.389、Ch01 = +07.389、Ch02 = +00.002、Ch03 = +00.002。	
指令: #03	讀取模組 (03, 例如: tM-TH8) 所有的 AI 通道。
回應: >-9999.9-9999.9-9999.9-9999.9-9999.9-9999.9-9999.9 收到 Engineering 格式的資料: Ch00 ~ 07 皆為 -9999.9 (Under Range)。	

相關指令: [#AAN](#)、[#AAA](#)

相關章節:

3.3 設定頁面 - AI、3.4.3 設定頁面 - AI/DO Alarm

#AAN 讀取指定的通道 (AI)

註: 此指令支援 AI 與 tM-AD4P2C2 模組。

語法:

#AAN[CHKSUM](CR)

- # 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制，00 ~ FF)。
- N 欲讀取的通道 (由 0 開始)。

回應:

有效指令: >(Data)[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

- > 有效指令的定義符號。
- ? 無效指令的定義符號。
- (Data) 該指定通道的 AI 資料。
- AA 模組位址 (16 進制，00 ~ FF)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例:

指令: #010	讀取模組 (01，例如: tM-AD5) 的通道 0。
回應: >+06.251	表示 Ch00 = +06.250
指令: #013	讀取模組 (01，例如: tM-AD2) 的通道 3。
回應: ?02	回傳無效指令，因為無通道 3。
指令: #023	讀取模組 (02，例如: tM-AD4P2C2) 的通道 3。
回應: >+00.002	表示 Ch03 = +00.002。

相關指令: [#AA](#)、[#AAA](#)

相關章節:

3.3 設定頁面 - AI、3.4.3 設定頁面 - AI/DO Alarm

\$AAA 讀取全部的通道 (AI，16 進制值)

註：此指令支援 AI 與 tM-AD4P2C2 模組。

語法：

\$AAA[CHKSUM](CR)

\$ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制，00 ~ FF)。

A 讀取全部通道的指令。

回應：

有效指令: >(Data)[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

> 有效指令的定義符號。

? 無效指令的定義符號。

(Data) 所有 AI 通道的 16 進制資料。

AA 模組位址 (16 進制，00 ~ FF)。

註：如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例：

指令: \$01A	讀取模組 (01，例如: tM-AD2) 全部的通道。
回應: >7FFA7FF7	Ch00 = +09.998、Ch01 = +09.998 (V)

指令: \$02A	讀取模組 (02，例如: tM-AD4P2C2) 全部的通道。
回應: >5E945E9400090008	表示 Ch00 = +07.389、Ch01 = +07.389、 Ch02 = +00.002、Ch03 = +00.002。

相關指令: [#AA](#)、[#AAN](#)

相關章節：

3.3 設定頁面 - AI、3.4.3 設定頁面 – AI/DO Alarm

\$AA8N 讀取模組的 AO 值

註: 此指令只適用於 **tM-DA1P1R1**。

語法:

\$AA8N[CHKSUM](CR)

~ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制, 00 ~ FF)。

8 讀取 AO Value 的指令。

N 欲讀取的通道, tM-DA1P1R1 為 0。

回應:

有效指令: **!AA(Data)**[CHKSUM](CR)

無效指令: **?AA**[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制, 00 ~ FF)。

(Data) AO Value。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

範例:

指令: #010+04.000	設定模組 (01) 通道 0 的 AO Value = 4.0。
回應: >	回傳有效指令。

指令: ~0180	讀取模組 (01) 通道 0 的 AO Value。
回應: !01+04.000	回傳 AO Value = 4.0。(可參考 B.7 tM-DA1P1R1 資料範圍)

相關指令:

[#AAN\(Data\)](#)、[\\$AA6N](#)

相關章節:

3.4.2 設定頁面 - AO、附錄 B.7 tM-DA1P1R1 資料範圍

\$AA6N 讀取模組上次寫入的 AO 值

註: 此指令只適用於 tM-DA1P1R1。

語法:

\$AA6N[CHKSUM](CR)

~ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制, 00 ~ FF)。

6 讀取上次寫入的 AO Value 指令。

N 欲讀取的通道, tM-DA1P1R1 為 0。

回應:

有效指令: !AA(Data)[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制, 00 ~ FF)。

(Data) 上次寫入的 AO Value。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

範例:

指令: #010+10.000	設定模組 (01) 通道 0 的 AO Value = 10.0。
回應: >	回傳有效指令。

指令: \$0160	讀取模組 (01) 通道 0 上次寫入的 AO Value。
回應: !01+10.000	回傳 10.0。

相關指令:

[#AAN\(Data\)](#)、[\\$AA8N](#)

相關章節:

3.4.2 設定頁面 - AO、附錄 B.7 tM-DA1P1R1 資料範圍

#AAN(Data) 設定模組的 AO 值

註: 此指令只適用於 **tM-DA1P1R1**。

語法:

#AAN(Data)[CHKSUM](CR)

- # 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制, 00 ~ FF)。
- N 欲設定的通道, tM-DA1P1R1 為 0。
- (Data) 欲寫入的 AO Value。(可參考附錄 B.7 tM-DA1P1R1 資料範圍)

回應:

有效指令: >[CHKSUM](CR)

超出範圍: ?[CHKSUM](CR)

略過指令: ![CHKSUM](CR)

- > 有效指令的定義符號。
- ? 資料超出範圍的定義符號。
若高於範圍, 會設為最大值; 若低於範圍, 會設為最小值。
- ! 忽略指令的定義符號。當 Host Watchdog Timeout 時, AO 會設為 Safe Value。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

範例:

指令: \$0190	讀取模組 (01) 通道 0 的 AO 配置。
回應: !0120	回傳 Type Code = 2 (0 ~ 10 V); Slew Rate = 0 (Immediate)。
指令: #010+05.000	設定模組 (01) 通道 0 的 AO Value = 5.0。
回應: >	回傳有效指令。
指令: #010+15.000	設定模組 (01) 通道 0 的 AO Value = 15.0。
回應: ?	模組回應超出範圍, 並會設為 10 V。

相關指令: [\\$AA9N](#)、[%AANNTTCFF](#)

相關章節: 3.4.2 設定頁面 - AO、附錄 B.7 tM-DA1P1R1 資料範圍

~** 通知所有模組 Host 正常

說明:

通知所有模組 Host 目前運作正常。

註: 傳送此指令後，在下一個指令傳送前會有 2ms 延遲時間。

語法:

~**[CHKSUM](CR)

~ 定義符號字元 (Delimiter Character)。

** Host 正常的指令。

回應:

無回應。

範例:

指令: ~**	傳送 "Host OK" 的指令給所有的模組。
回應: 無回應	

相關指令:

[~AA0](#)、[~AA1](#)、[~AA2](#)、[~AA3EVV](#)

相關章節:

3.4.5 設定頁面 – Host WDT、A.2 雙看門狗功能 (Dual Watchdog)

~AA0 讀取模組的 Host Watchdog 狀態

語法:

~AA0[CHKSUM](CR)

- ~ 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制, 00 ~ FF)。
- 0 讀取模組 Host Watchdog 狀態的指令。

回應:

有效指令: !AASS[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

- ! 有效指令的定義符號。
- ? 無效指令的定義符號。
- AA 模組位址 (16 進制, 00 ~ FF)。
- SS 用 2 位數 16 進制值, 來表示 Host Watchdog 狀態。
 - Bit 7: 0, 表示 Host Watchdog 是關閉的 (Disable)。
 - 1, 表示 Host Watchdog 是啟用的 (Enable)。
 - Bit 2: 0, 表示 Host Watchdog 未 Timeout。
 - 1, 表示 Host Watchdog 已 Timeout。

註: Host Watchdog 狀態儲存在 EEPROM, 且只能用 ~AA1 指令重置。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

範例:

指令： ~010	讀取模組 (01) 的 Host Watchdog 狀態。																								
回應： !0100	回傳 00，表示未使用 Host Watchdog 功能，也沒有發生 Timeout。																								
指令： ~020	讀取模組 (02) 的 Host Watchdog 狀態。																								
回應： !0204	回傳 04，表示 Host Watchdog 已 Timeout。																								
	<table><tr><td>Bit 7</td><td>Bit 6</td><td>Bit 5</td><td>Bit 4</td><td>Bit 3</td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td colspan="4">0</td><td colspan="4">4</td></tr></table>	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	0	0	0	0	0	1	0	0	0				4			
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0																	
	0	0	0	0	0	1	0	0																	
0				4																					

相關指令: [~**](#)、[~AA0](#)、[~AA1](#)、[~AA2](#)、[~AA3EVV](#)

相關章節: 3.4.5 設定頁面 – Host WDT、A.2 雙看門狗功能 (Dual Watchdog)

~AA1 重置模組的 Host Watchdog 狀態

語法:

~AA1[CHKSUM](CR)

- ~ 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制, 00 ~ FF)。
- 1 重置模組 Host Watchdog 狀態的指令。

回應:

有效指令: !AASS[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

- ! 有效指令的定義符號。
- ? 無效指令的定義符號。
- AA 模組位址 (16 進制, 00 ~ FF)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

範例:

指令: ~010	讀取模組 (01) 的 Host Watchdog 狀態。
回應: !0104	回傳 04, 表示 Host Watchdog 已 Timeout。
指令: ~011	重置模組 (01) 的 Host Watchdog 狀態。
回應: !01	回傳有效指令。
指令: ~010	讀取模組 (01) 的 Host Watchdog 狀態。
回應: !0100	回傳 00, 表示未使用 Host Watchdog 功能, 也沒有發生 Timeout。

相關指令:

[~**](#)、[~AA0](#)、[~AA2](#)、[~AA3EVV](#)

相關章節:

3.4.5 設定頁面 – Host WDT、A.2 雙看門狗功能 (Dual Watchdog)

~AA2 讀取模組的 Host Watchdog Timeout 值

語法:

~AA2[CHKSUM](CR)

~ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制, 00 ~ FF)。

2 讀取 Host Watchdog Timeout 值的指令。

回應:

有效指令: !AAEVV[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制, 00 ~ FF)。

E 0, 表示 Host Watchdog 是關閉的 (Disable)。

1, 表示 Host Watchdog 是啟用的 (Enable)。

VV 用 2 位數 16 進制值, 以 10^{-1} 秒來表示 Timeout 值。

例如: 01, 表示 0.1 秒; FF, 表示 25.5 秒。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

範例:

指令: ~012	讀取模組 (01) 的 Host Watchdog Timeout。
回應: !011FF	回傳 1FF, 表示 Host Watchdog 啟用, 且 Timeout = 25.5 秒。

相關指令:

[~**](#)、[~AA0](#)、[~AA1](#)、[~AA3EVV](#)

相關章節:

3.4.5 設定頁面 – Host WDT、A.2 雙看門狗功能 (Dual Watchdog)

~AA3EVV 設定模組的 Host Watchdog 功能 與 Timeout 值

說明:

啟用/關閉 模組的 Host Watchdog 功能，並設定 Timeout 值。

語法:

~AA3EVV[CHKSUM](CR)

- ~ 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制，00 ~ FF)。
- 3 設定 Host Watchdog 的指令。
- E 0，表示關閉 Host Watchdog。
1，表示啟用 Host Watchdog。
- VV 用 2 位數 16 進制值，以 10^{-1} 秒來表示 Timeout 值。
例如: 01，表示 0.1 秒；FF，表示 25.5 秒。

回應:

有效指令: !AA[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

- ! 有效指令的定義符號。
- ? 無效指令的定義符號。
- AA 模組位址 (16 進制，00 ~ FF)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例:

指令: ~013164	啟用模組 (01) 的 Host Watchdog，並設定 Timeout = 10.0 秒。 $64_{(16)} = 100_{(10)}$
回應: !01	回傳有效指令。

指令: ~012	讀取模組 (01) 的 Host Watchdog Timeout。
回應: !01164	回傳 164，表示 Host Watchdog 啟用，且 Timeout = 10.0 秒。

相關指令:

[~**](#)、[~AA0](#)、[~AA1](#)、[~AA2](#)

相關章節:

3.4.5 設定頁面 – Host WDT、A.2 雙看門狗功能 (Dual Watchdog)

~AA4V 讀取模組的 Power-on 或 Safe Value (DO)

註: tM-AD4P2C2 需使用 [~AA4](#) 指令。

語法:

~AA4V[CHKSUM](CR)

- ~ 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制, 00 ~ FF)。
- 4 讀取 Power-on Value 與 Safe Value 的指令。
- V P: 讀取 Power-on Value。
S: 讀取 Safe Value。

回應:

有效指令: !AA(Data)[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

- ! 有效指令的定義符號。
- ? 無效指令的定義符號。
- AA 模組位址 (16 進制, 00 ~ FF)。
- (Data) 用 2 位數 16 進制值, 加上 00, 來表示 Power-on Value 或 Safe Value。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

範例:

指令: ~014S	讀取模組 (01 · 例如: tM-DA1P1R1) 的 Safe Value 。		
回應: !01 <u>01</u> 00	回傳 Safe Value = 01 。		
	<table><tr><td>DO0</td></tr><tr><td>ON</td></tr></table>	DO0	ON
DO0			
ON			

指令: ~024P	讀取模組 (02 · 例如: tM-C8) 的 Power-on Value 。																
回應: !02 <u>C3</u> 00	回傳 Power-on Value = C3 。																
	<table><tr><td>DO7</td><td>DO6</td><td>DO5</td><td>DO4</td><td>DO3</td><td>DO2</td><td>DO1</td><td>DO0</td></tr><tr><td>ON</td><td>ON</td><td>OFF</td><td>OFF</td><td>OFF</td><td>OFF</td><td>ON</td><td>ON</td></tr></table>	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0	ON	ON	OFF	OFF	OFF	OFF	ON	ON
DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0										
ON	ON	OFF	OFF	OFF	OFF	ON	ON										

相關指令: [~AA5V](#)

相關章節: 3.4.4 設定頁面 - DO、A.2 雙看門狗功能 (Dual Watchdog)

~AA5V 將模組 DO 值，設為 Power-on 或 Safe Value

註: tM-AD4P2C2 需使用 [~AA5PPSS](#) 指令。

語法:

~AA5V[CHKSUM](CR)

- ~ 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制，00 ~ FF)。
- 5 設定 Power-on Value 與 Safe Value 的指令。
- V P: 設定 Power-on Value。
S: 設定 Safe Value。

回應:

有效指令: !AA[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

- ! 有效指令的定義符號。
- ? 無效指令的定義符號。
- AA 模組位址 (16 進制，00 ~ FF)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例:

指令: ~015S	將模組 (01，例如: tM-P4A4) 目前的 DO 值，設為 Safe Value。
回應: !01	回傳有效指令。

指令: ~014S	讀取模組 (01) 的 Safe Value。								
回應: !010C00	<div>回傳 Safe Value = 0C。</div> <table><tr><td>DO3</td><td>DO2</td><td>DO1</td><td>DO0</td></tr><tr><td>ON</td><td>ON</td><td>OFF</td><td>OFF</td></tr></table>	DO3	DO2	DO1	DO0	ON	ON	OFF	OFF
DO3	DO2	DO1	DO0						
ON	ON	OFF	OFF						

相關指令: [~AA4V](#)

相關章節: 3.4.3 設定頁面 – AI/DO Alarm、3.4.4 設定頁面 - DO、
A.2 雙看門狗功能 (Dual Watchdog)

~AA4 讀取模組的 Power-on 與 Safe Value (Multi-function-DO)

註：此指令只適用於 tM-DA1P1R1 與 tM-AD4P2C2。

語法：

~AA4[CHKSUM](CR)

- ~ 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制，00 ~ FF)。
- 4 讀取 Power-on Value 與 Safe Value 的指令。

回應：

有效指令：!AAPPSS[CHKSUM](CR)

無效指令：?AA[CHKSUM](CR)

- ! 有效指令的定義符號。
- ? 無效指令的定義符號。
- AA 模組位址 (16 進制，00 ~ FF)。
- PP 用 2 位數 16 進制值，表示 Power-on Value。
- SS 用 2 位數 16 進制值，表示 Safe Value。

註：如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例：

指令： ~014	讀取模組 (01) 的 Power-on Value 與 Safe Value。			
回應： !010000	回傳 Power-on Value =00；Safe Value=00。			
指令： ~024	讀取模組 (02) 的 Power-on Value 與 Safe Value。			
回應： !020203	回傳 Power-on Value =02；Safe Value=03。			
	Power-on Value		Safe Value	
	DO1	DO0	DO1	DO0
	ON	OFF	ON	ON

相關指令： [~AA5PPSS](#)

相關章節： 3.4.3 設定頁面 – AI/DO Alarm、3.4.4 設定頁面 - DO、
A.2 雙看門狗功能 (Dual Watchdog)

~AA5PPSS 設定模組的 Power-on 與 Safe Value (Multi-function-DO)

註: 此指令只適用於 tM-DA1P1R1 與 tM-AD4P2C2。

若 tM-AD4P2C2 有啟用 Alarm 功能，此設定將無效。

語法:

~AA5PPSS[CHKSUM](CR)

- ~ 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制，00 ~ FF)。
- 5 設定 Power-on Value 與 Safe Value 的指令。
- PP 用 2 位數 16 進制值，表示 Power-on Value。
- SS 用 2 位數 16 進制值，表示 Safe Value。

回應:

有效指令: !AA [CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

- ! 有效指令的定義符號。
- ? 無效指令的定義符號。
- AA 模組位址 (16 進制，00 ~ FF)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例:

指令: ~0150000	設定模組 (01) 的 Power-on Value = 00 與 Safe Value = 00。
回應: !01	回傳有效指令。

指令： ~014	讀取模組 (01) 的 Power-on Value 與 Safe Value 。			
回應： !010000	回傳 Power-on Value =00 ； Safe Value=00 。			
	Power-on Value		Safe Value	
	DO1	DO0	DO1	DO0
	OFF	OFF	OFF	OFF

相關指令: [~AA4](#)

相關章節: 3.4.3 設定頁面 – AI/DO Alarm、3.4.4 設定頁面 - DO、
A.2 雙看門狗功能 (Dual Watchdog)

\$AA7N 讀取模組的 Power-on Value (AO)

說明:

讀取指定通道的 Power-on Value (AO)。

註: 此指令只適用於 tM-DA1P1R1。

語法:

\$AA7N[CHKSUM](CR)

\$ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制, 00 ~ FF)。

7 讀取 Power-on AO Value 的指令。

N 欲讀取的通道, tM-DA1P1R1 為 0。

回應:

有效指令: !AA(Data)[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制, 00 ~ FF)。

(Data) Power-on Value (AO)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

範例:

指令: \$0170	讀取模組 (01) 通道 0 的 Power-on AO Value。
回應: !01+03.000	回傳 Power-on Value = 3.0。

相關指令: [\\$AA4N](#)

相關章節: 3.4.2 設定頁面 - AO、A.2 雙看門狗功能 (Dual Watchdog)

\$AA4N 將模組 AO 值，設為 Power-on Value

說明:

將指定通道的 AO 值設為 Power-on Value。

註: 此指令只適用於 tM-DA1P1R1。

語法:

\$AA4N[CHKSUM](CR)

\$ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制，00 ~ FF)。

4 設定 Power-on Value 的指令。

N 欲設定的通道，tM-DA1P1R1 為 0。

回應:

有效指令: !AA[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制，00 ~ FF)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例:

指令: #010+05.000	設定模組 (01) 通道 0 的 AO Value = 5.0。
回應: >	回傳有效指令。

指令: \$0140	將模組 (01) 通道 0 目前的 AO 值，設為 Power-on Value。
回應: !01	回傳有效指令。

指令: \$0170	讀取模組 (01) 通道 0 目前的 Power-on Value。
回應: !01+05.000	回傳 Power-on Value = 5.0。

相關指令: [#AAN\(Data\)](#)、[\\$AA7N](#)

相關章節: 3.4.2 設定頁面 - AO、A.2 雙看門狗功能 (Dual Watchdog)

~AA4N 讀取模組的 Safe Value (AO)

說明:

讀取指定通道的 Safe Value (AO)。

註: 此指令只適用於 tM-DA1P1R1。

語法:

~AA4N[CHKSUM](CR)

~ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制, 00 ~ FF)。

4 讀取 Safe AO Value 的指令。

N 欲讀取的通道, tM-DA1P1R1 為 0。

回應:

有效指令: !AA(Data)[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制, 00 ~ FF)。

(Data) Safe Value (AO)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

範例:

指令: ~0140	讀取模組 (01) AO 通道 0 的 Safe Value。
回應: !01+02.500	回傳 Safe Value = 2.5。

相關指令: [~AA5N](#)

相關章節: 3.4.2 設定頁面 - AO、A.2 雙看門狗功能 (Dual Watchdog)

~AA5N 將模組 AO 值，設為 Safe Value

說明:

將指定通道的 AO 值設為 Safe Value。

註: 此指令只適用於 tM-DA1P1R1。

語法:

~AA5N[CHKSUM](CR)

~ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制，00 ~ FF)。

5 指定 Safe Value 的指令。

N 欲設定的通道，tM-DA1P1R1 為 0。

回應:

有效指令: !AA[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制，00 ~ FF)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例:

指令: #010+03.500	設定模組 (01) 通道 0 的 AO Value = 3.5。
回應: >	回傳有效指令。
指令: ~0150	將模組 (01) 通道 0 目前的 AO 值，設為 Safe Value。
回應: !01	回傳有效指令。
指令: ~0140	讀取模組 (01) 通道 0 的 Safe Value。
回應: !01+03.500	回傳 Safe Value = 3.5。

相關指令: [#AAN\(Data\)](#)、[~AA4N](#)

相關章節: 3.4.2 設定頁面 - AO、A.2 雙看門狗功能 (Dual Watchdog)

\$AALS 讀取 DIO Latch 狀態

語法:

\$AALS[CHKSUM](CR)

\$ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制, 00 ~ FF)。

L 讀取 DI Latch 的指令。

S 0: 讀取 Low Latch 狀態

1: 讀取 High Latch 狀態

回應:

有效指令: **!AA(Data)**[CHKSUM](CR)

無效指令: **?AA**[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制, 00 ~ FF)。

(Data) 用 4 位數 16 進制值, 再加上 00, 表示 DIO Latch 的狀態。

請參考, [\\$AA6 指令 – DIO 資料回應格式表](#)。

DI/DO 模組: (DO 或 DI Latch) (00) 00

DIO 模組: (DO Latch) (DI Latch) 00

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

範例:

指令: \$01L0	讀取模組 (01・例如: tM-P8) 的 Low Latch 狀態。																
回應: ! <u>3C</u> 0000	<div>DI 模組: (DI Latch) (00) 00</div> <table><tr><td>DI7</td><td>DI6</td><td>DI5</td><td>DI4</td><td>DI3</td><td>DI2</td><td>DI1</td><td>DI0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr></table>	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0	0	0	1	1	1	1	0	0
DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0										
0	0	1	1	1	1	0	0										

指令: \$02L0	讀取模組 (02・例如: tM-P3POR3) 的 Low Latch 狀態。												
回應: ! <u>0707</u> 00	<div>DIO 模組:(DO Latch) (DI Latch) 00</div> <table><tr><td>DO2</td><td>DO1</td><td>DO0</td><td>DI2</td><td>DI1</td><td>DI0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>	DO2	DO1	DO0	DI2	DI1	DI0	1	1	1	1	1	1
DO2	DO1	DO0	DI2	DI1	DI0								
1	1	1	1	1	1								

相關指令: [\\$AAC](#)

相關章節: 3.4.6 設定頁面 – DI、A.6 進階數位輸入功能 (DI Latch、Counter)

\$AAC 清除 DIO Latch 狀態

說明:

清除模組的 High 與 Low Latch 狀態。

語法:

\$AAC[CHKSUM](CR)

\$ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制, 00 ~ FF)。

C 清除 DIO Latch 的指令。

回應:

有效指令: **!AA**[CHKSUM](CR)

無效指令: **?AA**[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制, 00 ~ FF)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

範例:

指令: \$02C	清除模組 (02) 的 DIO Latch 狀態。
回應: !02	回傳有效指令。

測試指令:

當 DI 由 OFF 變成 ON, 會產生 High Latch; 而由 ON 變成 OFF, 會產生 Low Latch。

若目前在 Low Latch 執行 **\$02C** 指令, 可刪除先前的 High Latch 狀態。

參考 [第 4 章的測試接線](#), 來連接 tM-P8 (8 DI) 與 tM-C8 (8 DO) 模組, 並在 PC 上使用 DCON Utility Pro 來測試指令。

1. 於 DO 設定頁面 (見 3.4.4 節), 勾選 DO0 ~ DO7 設定為 ON。
此時, DI0 ~ DI7 會由 OFF 變成 ON, 產生 High Latch。
2. 再取消勾選 DO0 ~ DO7 設定為 OFF。
此時, DI0 ~ DI7 會由 ON 變成 OFF, 會產生 Low Latch。
3. 執行 **\$02C** 指令, 可刪除先前的 High Latch 狀態。

相關指令: [\\$AALS](#)

相關章節: 3.4.4 設定頁面 - DO、3.4.6 設定頁面 - DI、A.6 進階數位輸入功能 (DI Latch、Counter)

#AAN 讀取 DI Counter

語法:

#AAN[CHKSUM](CR)

定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制 · 00 ~ FF)。

N 欲讀取的通道 (0 ~ F)。

回應:

有效指令: **!(Data)**[CHKSUM](CR)

無效指令: **?AA**[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

(Data) 使用 5 位數，用來表示指定通道的 DI Counter 值 (00000 ~ 65535)

AA 模組位址 (16 進制 · 00 ~ FF)。

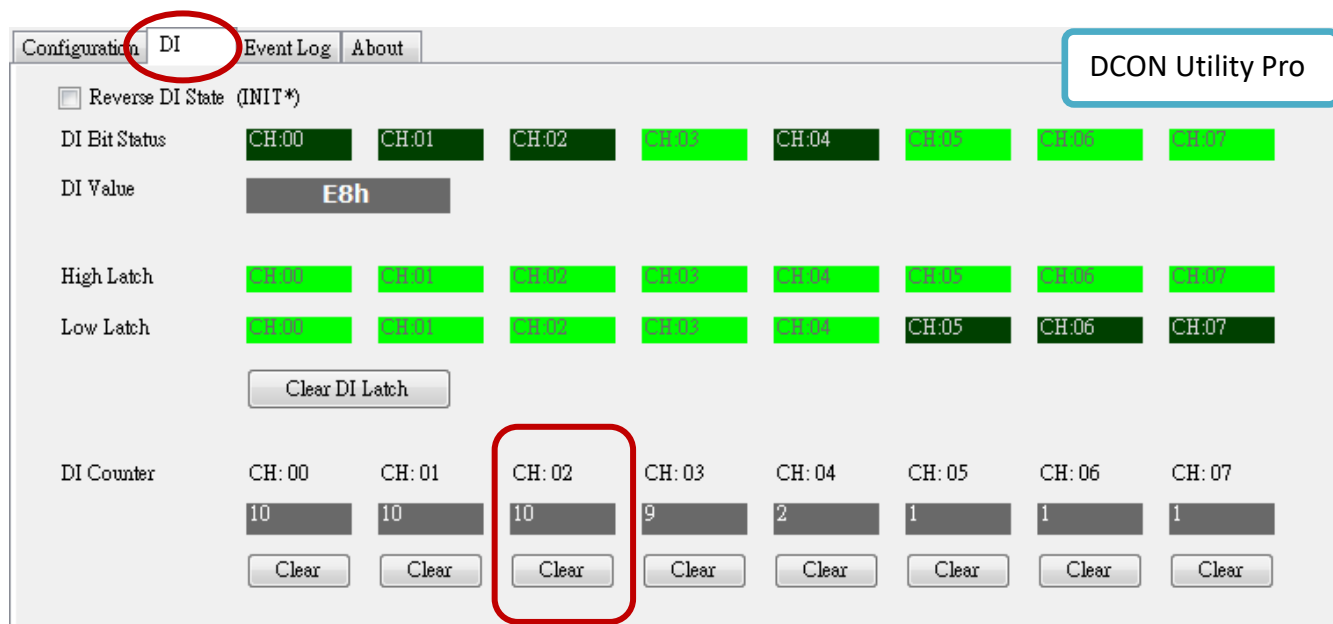
註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例:

指令: #012	讀取模組 (01，例如: tM-P8) 通道 2 的計數值。
回應: !0100010	回傳計數值為 10。

指令: #019	讀取模組 (01) 通道 9 的計數值。
回應: ?01	回傳無效指令，因為無通道 9。

相關指令: [SAACN](#) 相關章節: 3.4.6 設定頁面 – DI、A.6 進階數位輸入功能 (DI Latch、Counter)



\$AACN 清除 DI Counter

語法:

\$AACN[CHKSUM](CR)

\$ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制 · 00 ~ FF)。

C 清除 DI Counter 值的指令。

N 欲清除的通道 (0 ~ F)。

回應:

有效指令: !AA[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制 · 00 ~ FF)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例:

指令: #012	讀取模組 (01，例如: tM-P8) 通道 02 的計數值。
回應: !0100010	回傳計數值為 10。

指令: \$01C2	清除模組 (01，例如: tM-P8) 通道 02 的計數值。
回應: !01	回傳有效指令。

相關指令: [#AAN](#) 相關章節: 3.4.6 設定頁面 – DI、A.6 進階數位輸入功能 (DI Latch、Counter)

Configuration | **DI** | Event Log | About

DCON Utility Pro

☐ Reverse DI State (INIT*)

DI Bit Status: CH:00 CH:01 CH:02 CH:03 CH:04 CH:05 CH:06 CH:07

DI Value: E8h

High Latch: CH:00 CH:01 CH:02 CH:03 CH:04 CH:05 CH:06 CH:07

Low Latch: CH:00 CH:01 CH:02 CH:03 CH:04 CH:05 CH:06 CH:07

Clear DI Latch

DI Counter:

CH:00	CH:01	CH:02	CH:03	CH:04	CH:05	CH:06	CH:07
10	10	0	9	2	1	1	1
Clear	Clear	Clear	Clear	Clear	Clear	Clear	Clear

@AARECi 讀取 DI Counter (適用 tM-AD4P2C2 模組)

語法:

@AARECi[CHKSUM](CR)

@ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制 · 00 ~ FF)。

RE 讀取 DI Counter 的指令。

Ci i，表示欲讀取的通道 (0 ~ 1)。

回應:

有效指令: !AA(Data)[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制 · 00 ~ FF)。

(Data) 使用 5 位數，用來表示指定通道的 DI Counter (00000 ~ 65535)

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

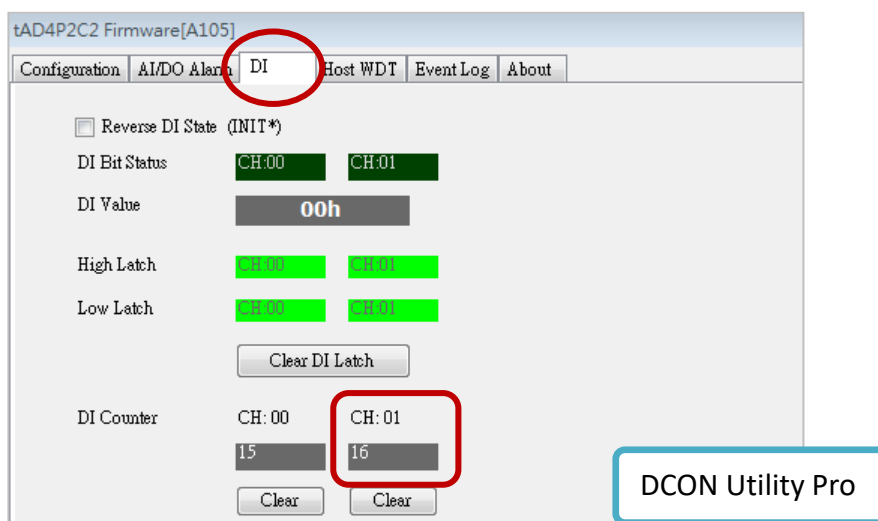
範例:

指令: @02REC1	讀取模組 (02) 通道 1 的計數值。
回應: !0200016	回傳計數值為 16。

指令: @02REC9	讀取模組 (02) 通道 9 的計數值。
回應: ?02	回傳無效指令，因為無通道 9。

相關指令: [@AAECi](#)

相關章節: 3.4.6 設定頁面 – DI、A.6 進階數位輸入功能 (DI Latch、Counter)



@AACECi 清除 DI Counter (適用 tM-AD4P2C2 模組)

語法:

@AACECi[CHKSUM](CR)

@ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制 · 00 ~ FF)。

CE 清除 DI Counter 的指令。

Ci i, 表示欲清除的通道 (0 ~ 1)。

回應:

有效指令: !AA[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制 · 00 ~ FF)。

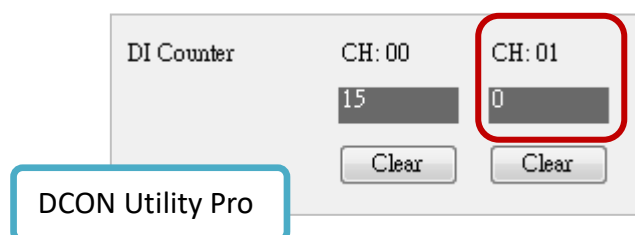
註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

範例:

指令: @02REC1	讀取模組 (02) 通道 1 的計數值。
回應: !02 <u>00016</u>	回傳計數值為 16。
指令: @02CEC1	清除模組 (02) 通道 1 的計數值。
回應: !02	回傳有效指令。
指令: @02REC1	讀取模組 (02) 通道 1 的計數值。
回應: !02 <u>00000</u>	回傳計數值為 0。

相關指令: [@AARECi](#)

相關章節: 3.4.6 設定頁面 – DI、A.6 進階數位輸入功能 (DI Latch、Counter)



@AARECN 讀取 DI Counter (適用 tM-DA1P1R1 模組)

語法:

@AARECN[CHKSUM](CR)

- @ 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制 · 00 ~ FF)。
- RE 讀取 DI Counter 的指令。
- CN N，表示欲讀取的通道，tM-DA1P1R1 為 0。

回應:

有效指令: !AA(Data)[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

- ! 有效指令的定義符號。
 - ? 無效指令的定義符號。
 - AA 模組位址 (16 進制 · 00 ~ FF)。
 - (Data) 使用 5 位數，用來表示指定通道的 DI Counter (00000 ~ 65535)
- 註:** 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

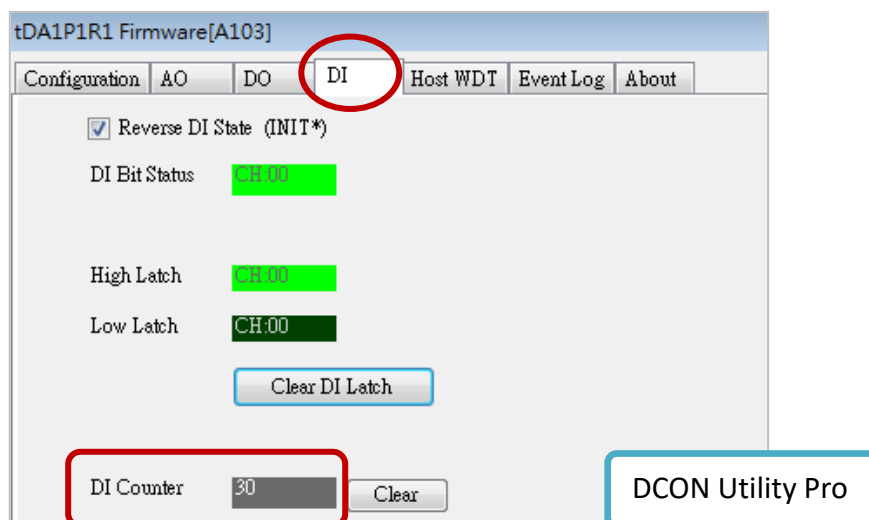
範例:

指令: @01RECO	讀取模組 (01) 通道 0 的計數值。
回應: !0100030	回傳計數值為 30。

指令: @01REC9	讀取模組 (01) 通道 9 的計數值。
回應: ?01	回傳無效指令，因為無通道 9。

相關指令: [@AACECN](#)

相關章節: 3.4.6 設定頁面 – DI、A.6 進階數位輸入功能 (DI Latch、Counter)



@AACECN 清除 DI Counter (適用 tM-DA1P1R1 模組)

語法:

@AACECN[CHKSUM](CR)

@ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制 · 00 ~ FF)。

CE 清除 DI Counter 的指令。

CN N，表示欲清除的通道 (0)。

回應:

有效指令: !AA[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制 · 00 ~ FF)。

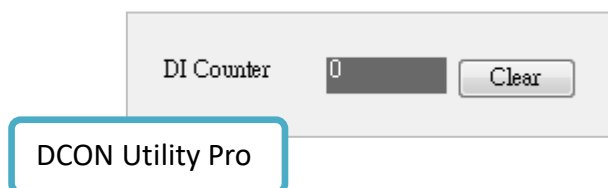
註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例:

指令: @01RECO	讀取模組 (01) 通道 0 的計數值。
回應: !0100030	回傳計數值為 30。
指令: @01CECO	清除模組 (01) 通道 0 的計數值。
回應: !01	回傳有效指令。
指令: @01RECO	讀取模組 (01) 通道 0 的計數值。
回應: !0100000	回傳計數值為 0。

相關指令: [@AARECN](#)

相關章節: 3.4.6 設定頁面 – DI、A.6 進階數位輸入功能 (DI Latch、Counter)



@AAEATCi 啟用 AI Alarm 功能

註：此指令只適用於 **tM-AD4P2C2**。

語法：

@AAEATCi[CHKSUM](CR)

@ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制，00 ~ FF)。

T Alarm 類型。

M: Momentary ; L: Latched

ci i，表示欲啟用 Alarm 的通道。

回應：

有效指令: **!AA**[CHKSUM](CR)

無效指令: **?AA**[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制，00 ~ FF)。

註：如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例：

指令: @01EAMC0	啟用模組 (01) 通道 0 的 Alarm 功能，且類型為 “Momentary”。
回應: !01	回傳有效指令。

指令: @02EALC1	啟用模組 (02) 通道 1 的 Alarm 功能，且類型為 “Latch”。
回應: !01	回傳有效指令。

相關指令: [@AADACi](#)、[@AARACi](#)

相關章節: 3.4.3 設定頁面 - AI/DO Alarm

@AADACi 關閉 AI Alarm 功能

註：此指令只適用於 **tM-AD4P2C2**。

語法：

@AADACi[CHKSUM](CR)

- @ 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制，00 ~ FF)。
- DA 關閉 Alarm 的指令。
- Ci i，表示欲關閉的 Alarm 的通道。

回應：

有效指令：!AA[CHKSUM](CR)

無效指令：?AA[CHKSUM](CR)

- ! 有效指令的定義符號。
- ? 無效指令的定義符號。
- AA 模組位址 (16 進制，00 ~ FF)。

註：如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例：

指令： @02DAC0	關閉模組 (02) 通道 0 的 Alarm 功能。
回應： !01	回傳有效指令。

相關指令： [@AAEATCi](#)

相關章節： 3.4.3 設定頁面 - AI/DO Alarm

@AARACi 讀取 Alarm 類型設定

註：此指令只適用於 **tM-AD4P2C2**。

語法：

@AARACi[CHKSUM](CR)

- @ 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制，00 ~ FF)。
- RA 讀取 Alarm 設定的指令。
- Ci i，表示欲讀取 Alarm 設定的通道。

回應：

有效指令: !AAS[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

- ! 有效指令的定義符號。
- ? 無效指令的定義符號。
- AA 模組位址 (16 進制，00 ~ FF)。
- S Alarm 類型。
0: Disable ; 1: Momentary ; 2: Latch

註：如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例：

指令: @01RAC0	讀取模組 (01) 通道 0 的 Alarm 設定。
回應: !011	回傳 Alarm 類型為 “Momentary”。

指令: @01RAC1	讀取模組 (01) 通道 1 的 Alarm 設定。
回應: !012	回傳 Alarm 類型為 “Latch”。

相關指令: [@AARAO](#)、[@AARHCi](#)、[@AARLCi](#)、[@AAHI\(data\)Ci](#)、[@AALO\(data\)Ci](#)、[@AACHCi](#)、[@AACLCi](#)

相關章節: 3.4.3 設定頁面 - AI/DO Alarm

@AARAO 讀取 High/Low Alarm 狀態

註：此指令只適用於 **tM-AD4P2C2**。

語法：

@AARAO[CHKSUM](CR)

@ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制，00 ~ FF)。

RAO 讀取 Alarm 設定的指令。

回應：

有效指令：**!AAHLL**[CHKSUM](CR)

無效指令：**?AA**[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制，00 ~ FF)。

HH 用 2 位數 16 進制值，表示目前觸發的 High Alarm。

LL 用 2 位數 16 進制值，表示目前觸發的 Low Alarm。

Bit 0 對應到通道 0，Bit 1 對應到通道 1...等等。

當 Bit 為 1，表示該通道有觸發 High/ Low Alarm；

Bit 為 0，表示該通道未觸發 High/ Low Alarm。

註：如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例：

指令：@01RAO	讀取模組 (01) 目前的 Alarm 觸發狀態。
回應：!010003	表示通道 0、1 皆觸發 Low Alarm。

指令：@02RAO	讀取模組 (02) 目前的 Alarm 觸發狀態。
回應：!020200	表示通道 1 觸發 High Alarm。

相關指令：[@AARACi](#)、[@AARHCi](#)、[@AARLCi](#)、[@AAHI\(data\)Ci](#)、[@AALO\(data\)Ci](#)、[@AACHCi](#)、[@AACLCi](#)

相關章節：3.4.3 設定頁面 - AI/DO Alarm

@AARHCi 讀取 High Alarm Limit 值

註：此指令只適用於 **tM-AD4P2C2**。

語法：

@AARHCi[CHKSUM](CR)

- @ 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制，00 ~ FF)。
- RH 讀取 High Alarm Limit 值的指令。
- Ci i，表示欲讀取 High Alarm Limit 值的通道。

回應：

有效指令: **!AA(Data)[CHKSUM](CR)**

無效指令: **?AA[CHKSUM](CR)**

- ! 有效指令的定義符號。
- ? 無效指令的定義符號。
- AA 模組位址 (16 進制，00 ~ FF)。
- (Data) 以工程值表示 High Alarm Limit 值。

註：如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例：

指令: @02RHC0	讀取模組 (02) 通道 0 的 High Alarm Limit 值。
回應: !02+09.000	回傳 High Alarm Limit = + 9.0。

指令: @02RHC1	讀取模組 (02) 通道 1 的 High Alarm Limit 值。
回應: !02+08.000	回傳 A High Alarm Limit = +8.0。

相關指令: [@AARAO](#)、[@AARLCi](#)、[@AAHI\(data\)Ci](#)、[@AALO\(data\)Ci](#)、[@AACHCi](#)、[@AACLCi](#)

相關章節: 3.4.3 設定頁面 - AI/DO Alarm

@AARLCi 讀取 Low Alarm Limit 值

註：此指令只適用於 tM-AD4P2C2。

語法：

@AARLCi[CHKSUM](CR)

- @ 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制，00 ~ FF)。
- RL 讀取 Low Alarm Limit 值的指令。
- Ci i，表示欲讀取 Low Alarm Limit 值的通道。

回應：

有效指令：!AA(Data)[CHKSUM](CR)

無效指令：?AA[CHKSUM](CR)

- ! 有效指令的定義符號。
- ? 無效指令的定義符號。
- AA 模組位址 (16 進制，00 ~ FF)。
- (Data) 以工程值表示 Low Alarm Limit 值。

註：如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例：

指令： @02RLC0	讀取模組 (02) 通道 0 的 Low Alarm Limit 值。
回應： !02-09.000	回傳 Low Alarm Limit = -9.0。

指令： @02RLC1	讀取模組 (02) 通道 1 的 Low Alarm Limit 值。
回應： !02-08.000	回傳 Low Alarm Limit = -8.0。

相關指令： [@AARAO](#)、[@AARHCi](#)、[@AAHI\(data\)Ci](#)、[@AALO\(data\)Ci](#)、[@AACHCi](#)、[@AACLCi](#)

相關章節： 3.4.3 設定頁面 - AI/DO Alarm

@AAHI(data)Ci 設定 High Alarm Limit 值

註：此指令只適用於 **tM-AD4P2C2**。

語法：

@AAHiCi[CHKSUM](CR)

- @ 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制，00 ~ FF)。
- HI 設定 High Alarm Limit 值的指令。
- (data) 以工程值表示 High Alarm Limit 值。
- Ci i，表示欲設定 High Alarm Limit 值的通道。

回應：

有效指令：!AA[CHKSUM](CR)

無效指令：?AA[CHKSUM](CR)

- ! 有效指令的定義符號。
- ? 無效指令的定義符號。
- AA 模組位址 (16 進制，00 ~ FF)。

註：如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例：

指令： @01HI+10.000C0	設定模組 (01) 通道 0 的 High Alarm Limit = + 10.0。
回應： !01	回傳有效指令。

指令： @01HI+09.000C1	設定模組 (01) 通道 1 的 High Alarm Limit = + 9.0。
回應： !01	回傳有效指令。

相關指令： [@AARAO](#)、[@AARHCi](#)、[@AARLCi](#)、[@AALO\(data\)Ci](#)、[@AACHCi](#)、[@AACLCi](#)

相關章節： 3.4.3 設定頁面 - AI/DO Alarm

@AALO(data)Ci 設定 Low Alarm Limit 值

註：此指令只適用於 **tM-AD4P2C2**。

語法：

@AALOCi[CHKSUM](CR)

- @ 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制，00 ~ FF)。
- LO 設定 Low Alarm Limit 值的指令。
- (data) 以工程值表示 Low Alarm Limit 值。
- Ci i，表示欲設定 Low Alarm Limit 值的通道。

回應：

有效指令：!AA[CHKSUM](CR)

無效指令：?AA[CHKSUM](CR)

- ! 有效指令的定義符號。
- ? 無效指令的定義符號。
- AA 模組位址 (16 進制，00 ~ FF)。

註：如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例：

指令： @01LO-05.000C0	設定模組 (01) 通道 0 的 Low Alarm Limit = -5.0。
回應： !01	回傳有效指令。

指令： @01HI+01.000C1	設定模組 (01) 通道 1 的 Low Alarm Limit = + 1.0。
回應： !01	回傳有效指令。

相關指令： [@AARAO](#)、[@AARHCi](#)、[@AARLCi](#)、[@AAHI\(data\)Ci](#)、[@AACHCi](#)、[@AACLCi](#)

相關章節： 3.4.3 設定頁面 - AI/DO Alarm

@AACHCi 清除 High Latch Alarm 狀態

註: 此指令只適用於 **tM-AD4P2C2**。

語法:

@AACHCi[CHKSUM](CR)

- @ 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制, 00 ~ FF)。
- CH 清除 High Latch Alarm 的指令。
- Ci i, 表示需清除 High Latch Alarm 的通道。

回應:

有效指令: !AA [CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

- ! 有效指令的定義符號。
- ? 無效指令的定義符號。
- AA 模組位址 (16 進制, 00 ~ FF)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

範例:

指令: @01CHC1	清除模組 (01) 通道 1 的 High Latch Alarm 狀態。
回應: !01	回傳有效指令。

相關指令: [@AARAO](#)、[@AARHCi](#)、[@AARLCi](#)、[@AAHI\(data\)Ci](#)、[@AALO\(data\)Ci](#)、[@AACLCi](#)

相關章節: 3.4.3 設定頁面 - AI/DO Alarm

@AACLCi 清除 Low Latch Alarm 狀態

註：此指令只適用於 **tM-AD4P2C2**。

語法：

@AACLCi[CHKSUM](CR)

- @ 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制，00 ~ FF)。
- CL 清除 Low Latch Alarm 的指令。
- Ci i，表示需清除 Low Latch Alarm 的通道。

回應：

有效指令：!AA [CHKSUM](CR)

無效指令：?AA[CHKSUM](CR)

- ! 有效指令的定義符號。
- ? 無效指令的定義符號。
- AA 模組位址 (16 進制，00 ~ FF)。

註：如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例：

指令： @01CLC1	清除模組 (01) 通道 1 的 Low Latch Alarm 狀態。
回應： !01	回傳有效指令。

相關指令： [@AARAO](#)、[@AARHCi](#)、[@AARLCi](#)、[@AAHI\(data\)Ci](#)、[@AALO\(data\)Ci](#)、[@AACHCi](#)

相關章節： 3.4.3 設定頁面 - AI/DO Alarm

~AAEV 啟用/關閉校正功能

註: 此指令適用於 AI 與 tM-AD4P2C2 模組。

語法:

~AAEV[CHKSUM](CR)

~ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制, 00 ~ FF)。

E 啟用/關閉校正的指令。

V 1: 啟用校正 ; 0: 關閉校正。

回應:

有效指令: !AA [CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制, 00 ~ FF)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

範例:

指令: \$010	模組 (01) 執行 Span 校正。
回應: ?01	回傳無效指令, 因為未先啟用校正功能。

指令: ~01E1	模組 (01) 啟用校正功能。
回應: !01	回傳有效指令。

指令: \$010	模組 (01) 執行 Span 校正。
回應: !01	回傳有效指令。

相關指令: [\\$AA0](#)、[\\$AA1](#)、[\\$AA0Ci](#)、
[\\$AA1Ci](#) (只適用 tM-AD2)

相關章節: 3.6 校正功能 (Calibration)

\$AA1 執行 Zero 校正 (V, A)

註: 此指令適用於 AI 與 tM-AD4P2C2 模組，使用前請先執行 ~AAEV 指令。

語法:

\$AA1[CHKSUM](CR)

\$ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制，00 ~ FF)。

1 Zero 校正的指令。

回應:

有效指令: !AA [CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制，00 ~ FF)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例:

指令: \$011	模組 (01) 執行 Zero 校正。
回應: ?01	回傳無效指令，因為未先啟用校正功能。

指令: ~01E1	模組 (01) 啟用校正功能。
回應: !01	回傳有效指令。

指令: \$011	模組 (01) 執行 Zero 校正。
回應: !01	回傳有效指令。

相關指令: [~AAEV](#)、[\\$AA0](#)、[\\$AA0Ci](#)

相關章節: 3.6.2 校正 AI 模組

\$AA1Ci 執行 Zero 校正

註: 此指令只適用於 tM-AD2 模組，使用前請先執行 ~AAEV 指令。

語法:

\$AA1Ci[CHKSUM](CR)

\$ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制，00 ~ FF)。

1 Zero 校正的指令。

Ci i，表示欲校正的通道。

回應:

有效指令: !AA [CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制，00 ~ FF)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例:

指令: \$011C0	模組 (01) 的通道 0，執行 Zero 校正。
回應: ?01	回傳無效指令，因為未先啟用校正功能。

指令: ~01E1	模組 (01) 啟用校正功能。
回應: !01	回傳有效指令。

指令: \$011C0	模組 (01) 的通道 0，執行 Zero 校正。
回應: !01	回傳有效指令。

相關指令: [~AAEV](#)、[\\$AA0Ci](#)

相關章節: 3.6.2 校正 AI 模組

\$AA0 執行 Span 校正 (V, Ω)

註: 此指令適用於 tM-AD5、AD8、TH8、AD4P2C2 模組，使用前請先執行 ~AAEV 指令。

語法:

\$AA0[CHKSUM](CR)

\$ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制，00 ~ FF)。

0 Span 校正的指令。

回應:

有效指令: **!AA** [CHKSUM](CR)

無效指令: **?AA**[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制，00 ~ FF)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例:

指令: \$010	模組 (01) 執行 Span 校正。
回應: ?01	回傳無效指令，因為未先啟用校正功能。

指令: ~01E1	模組 (01) 啟用校正功能。
回應: !01	回傳有效指令。

指令: \$010	模組 (01) 執行 Span 校正。
回應: !01	回傳有效指令。

相關指令: [~AAEV](#)、[\\$AA1](#)

相關章節: 3.6.2 校正 AI 模組

\$AA0Ci 執行 Span 校正 (A)

註: 此指令適用於 tM-AD2、AD5C、AD8C、AD4P2C2 模組，使用前請先執行 ~AAEV 指令。

語法:

\$AA0Ci[CHKSUM](CR)

\$ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制，00 ~ FF)。

0 Span 校正的指令。

Ci i，表示欲校正的通道。

回應:

有效指令: !AA [CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制，00 ~ FF)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例:

指令: ~01E1	模組 (01) 啟用校正功能。
回應: !01	回傳有效指令。

指令: \$010C0	模組 (01) 的通道 0 執行 Span 校正。
回應: !01	回傳有效指令。

相關指令: [~AAEV](#)、[\\$AA1](#)、
[\\$AA1Ci](#) (只適用 tM-AD2)

相關章節: 3.6.2 校正 AI 模組

\$AA0N 執行 Zero 校正

註: 此指令適用於 **tM-AD1P1R1** 模組。

語法:

\$AA0[CHKSUM](CR)

\$ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制, 00 ~ FF)。

0 Zero 校正的指令。

N 欲校正的通道, tM-AD1P1R1 為 0。

回應:

有效指令: **!AA** [CHKSUM](CR)

無效指令: **?AA**[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制, 00 ~ FF)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

範例:

指令: \$0100	模組 (01) 通道 0 執行 Zero 校正。
回應: !01	回傳有效指令。

相關指令: [\\$AA1N](#)、[\\$AA3NVV](#)

相關章節: 3.6.1 校正 AO 模組

\$AA1N 執行 Span 校正

註: 此指令適用於 **tM-AD1P1R1** 模組。

語法:

\$AA1[CHKSUM](CR)

- \$ 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制, 00 ~ FF)。
- 1 Span 校正的指令。
- N 欲校正的通道, tM-AD1P1R1 為 0。

回應:

有效指令: !AA [CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

- ! 有效指令的定義符號。
- ? 無效指令的定義符號。
- AA 模組位址 (16 進制, 00 ~ FF)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

範例:

指令: \$0110	模組 (01) 通道 0 執行 Span 校正。
回應: !01	回傳有效指令。

相關指令: [\\$AA0N](#)、[\\$AA3NVV](#)

相關章節: 3.6.1 校正 AO 模組

\$AA3NVV 調整 AO 通道的校正值

註：此指令適用於 **tM-AD1P1R1** 模組。

語法：

\$AA3NVV[CHKSUM](CR)

\$ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制，00 ~ FF)。

3 調整 AO 值的指令。

N 欲調整數值的通道，tM-AD1P1R1 為 0。

VV 用 2 位數 16 進制值，00 ~ 5F，表示增加 0 ~ 95 計數；FF ~ A1，表示減少 1 ~ 95 計數。
每個計數大約是 4.88 μ A 或 2.44 mV。

回應：

有效指令：**!AA** [CHKSUM](CR)

無效指令：**?AA** [CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制，00 ~ FF)。

註：如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例：

指令： \$01301F	模組 (01) 的通道 0 增加 31 計數。 (大約 +151.28 μ A 或 +75.64 mV)
回應： !01	回傳有效指令。

相關指令： [\\$AA0N](#)、[\\$AA1N](#)

相關章節： 3.6.1 校正 AO 模組

\$AAS1 載入出廠預設的校正參數

註：此指令適用於 **tM-TH8** 模組。

語法：

\$AAS1[CHKSUM](CR)

\$ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制，00 ~ FF)。

S1 載入出廠預設的校正參數。

回應：

有效指令：**!AA** [CHKSUM](CR)

無效指令：**?AA**[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制，00 ~ FF)。

註：如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例：

指令： \$01S1	模組 (01) 載入出廠預設的校正參數。
回應： !01	回傳有效指令。

相關指令：[~AAEV](#)

相關章節：3.6.2 校正 AI 模組

@AAA3Ci 讀取通道的溫度偏移量

註: 此指令適用於 **tM-TH8** 模組。

語法:

@AAA3Ci[CHKSUM](CR)

- @ 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制, 00 ~ FF)。
- A3 讀取溫度偏移量的指令。
- Ci i, 表示欲讀取的通道 (0 ~ 7)。

回應:

有效指令: !AA(Data)[CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

- ! 有效指令的定義符號。
- ? 無效指令的定義符號。
- AA 模組位址 (16 進制, 00 ~ FF)。
- (Data) 用 2 位數 16 進制值, 表示溫度偏移量。
 - 01: 表示 0.1, 02: 表示 0.2, FF: 表示 -0.1, FE: 表示 -0.2。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

範例:

指令: @03A3C2	讀取模組 (03) 通道 02 的溫度偏移量。
回應: !030A	回傳數值 1.0。

指令: @03A3C5	讀取模組 (03) 通道 05 的溫度偏移量。
回應: !03F0	回傳數值 -1.6。

相關指令: [@AAA2CiToo](#)

相關章節: 3.3.3 設定頁面 03 - AI (適用模組: tM-TH8)

@AAA2CiToo 設定通道的溫度偏移量

註: 此指令適用於 **tM-TH8** 模組。

語法:

@AAA2CiToo[CHKSUM](CR)

@ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制, 00 ~ FF)。

A2 設定溫度偏移量的指令。

Ci i, 表示欲設定的通道 (0 ~ 7)。

Too oo, 表示溫度偏移量在 0.1°C/°F, 範圍: -12.8 ~ 12.7。

使用 16 進制值, **01**: 表示 0.1, **02**: 表示 0.2, **FF**: 表示 -0.1, **FE**: 表示 -0.2。

回應:

有效指令: **!AA** [CHKSUM](CR)

無效指令: **?AA** [CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制, 00 ~ FF)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

範例:

指令: @03A2C2T0A	設定模組 (03) 通道 02 的溫度偏移量為 1.0。
回應: !03	回傳有效指令。

指令: @03A2C5TF0	設定模組 (03) 通道 05 的溫度偏移量為 -1.6。
回應: !03	回傳有效指令。

相關指令: [@AAA3Ci](#)

相關章節: 3.3.3 設定頁面 03 - AI (適用模組: tM-TH8)

@AAA7Ci 讀取通道的電阻偏移量

註：此指令適用於 **tM-TH8** 模組。

語法：

@AAA7Ci[CHKSUM](CR)

@ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制，00 ~ FF)。

A7 讀取電阻偏移量的指令。

Ci i，表示欲讀取的通道 (0 ~ 7)。

回應：

有效指令: **!AA(Data)[CHKSUM](CR)**

無效指令: **?AA[CHKSUM](CR)**

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制，00 ~ FF)。

(Data) 用 2 位數 16 進制值，表示電阻偏移量。

01: 表示 0.1，**02:** 表示 0.2，**FF:** 表示 25.5，**FE:** 表示 25.4。

註：如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例：

指令: @03A7C2	讀取模組 (03) 通道 02 的電阻偏移量。
回應: !030A	回傳數值 1.0。

指令: @03A7C5	讀取模組 (03) 通道 05 的電阻偏移量。
回應: !03F0	回傳數值 24.0。

相關指令: [@AAA6CiRrr](#)

相關章節: 3.4.1 設定頁面 - User Defined Type

@AAA6CiRrr 設定通道的電阻偏移量

註：此指令適用於 **tM-TH8** 模組。

語法：

@AAA6CiRrr[CHKSUM](CR)

@ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制，00 ~ FF)。

A6 設定電阻偏移量的指令。

Ci i，表示欲設定的通道 (0 ~ 7)。

Rrr rr，表示電阻偏移量 (0.0 ~ 25.5 歐姆)。

01: 表示 0.1，**02:** 表示 0.2，**FF:** 表示 25.5，**FE:** 表示 25.4。

回應：

有效指令: **!AA** [CHKSUM](CR)

無效指令: **?AA** [CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制，00 ~ FF)。

註：如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例：

指令: @03A6C2T0A	設定模組 (03) 通道 02 的電阻偏移量為 1.0。
回應: !03	回傳有效指令。

指令: @03A6C5TF0	設定模組 (03) 通道 05 的電阻偏移量為 24.0。
回應: !03	回傳有效指令。

相關指令: [@AAA7Ci](#)

相關章節: 3.4.1 設定頁面 - User Defined Type

~AAD 讀取溫度單位 (°C/°F)

註: 此指令適用於 **tM-TH8** 模組。

語法:

~AAD[CHKSUM](CR)

~ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制, 00 ~ FF)。

D 讀取溫度單位的指令。

回應:

有效指令: !AAT [CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制, 00 ~ FF)。

T 溫度單位。0: 攝氏 (預設); 1: 華氏。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

範例:

指令: ~03D	讀取模組 (03) 的溫度單位。
回應: !010	回傳 "攝氏"。

指令: ~03D	讀取模組 (03) 的溫度單位。
回應: !011	回傳 "華氏"。

相關指令: [~AADI](#)

~AADT 設定溫度單位 (°C/°F)

註: 此指令適用於 **tM-TH8** 模組。

語法:

~AADT[CHKSUM](CR)

~ 定義符號字元 (Delimiter Character)。

AA 模組位址 (16 進制, 00 ~ FF)。

D 設定溫度單位的指令。

T C: 攝氏 (預設); F: 華氏。

回應:

有效指令: !AA [CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

! 有效指令的定義符號。

? 無效指令的定義符號。

AA 模組位址 (16 進制, 00 ~ FF)。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組, 都會沒有回應。

範例:

指令: ~03DC	設定模組 (03) 的溫度單位為 "攝氏"。
回應: !01	回傳有效指令。

指令: ~03DF	設定模組 (03) 的溫度單位為 "華氏"。
回應: !01	回傳有效指令。

相關指令: [~AAD](#)

@AAGxTtt 讀取 Steinhart-Hart 係數 (User-defined Type)

註：此指令適用於 **tM-TH8** 模組。

語法：

@AAGxTtt[CHKSUM](CR)

- @** 定義符號字元 (Delimiter Character)。
- AA** 模組位址 (16 進制，00 ~ FF)。
- G** 讀取 Steinhart-Hart 係數的指令。
- x** **A:** 讀取 Steinhart-Hart 係數 A。
B: 讀取 Steinhart-Hart 係數 B。
C: 讀取 Steinhart-Hart 係數 C。
- Ttt** tt，表示需讀取 Steinhart-Hart 係數的 User-defined Type (70 ~ 77)。

回應：

有效指令: **!AA(Data)[CHKSUM](CR)**

無效指令: **?AA[CHKSUM](CR)**

- !** 有效指令的定義符號。
- ?** 無效指令的定義符號。
- AA** 模組位址 (16 進制，00 ~ FF)。
- (Data)** 用 8 位數 16 進制值，以 IEEE-754 格式表示 Steinhart-Hart 係數。

註：如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例：

指令: @03GAT70	讀取模組 (03) Type Code = 70 的 Steinhart-Hart 係數 A。
回應: !033A94030A	回傳數值 3A94030A，相當於 1.129241×10^{-3} 。
指令: @03GBT70	讀取模組 (03) Type Code = 70 的 Steinhart-Hart 係數 B。
回應: !0339757ACF	回傳數值 39757ACF，相當於 2.341077×10^{-4} 。
指令: @03GCT70	讀取模組 (03) Type Code = 70 的 Steinhart-Hart 係數 C。
回應: !0333BC73A5	回傳數值 33BC73A5，相當於 8.775468×10^{-8} 。

相關指令: [@AASxTttC\(data\)](#)、[@AARTTttR\(Data\)](#)

相關章節: 3.4.1 設定頁面 - User Defined Type、B.8.1 Steinhart-Hart 係數 (User-defined Type)

@AASxTttC(data) 設定 Steinhart-Hart 係數 (User-defined Type)

註：此指令適用於 **tM-TH8** 模組。

語法：

@AASxTttC(data)[CHKSUM](CR)

- @ 定義符號字元 (Delimiter Character)。
- AA 模組位址 (16 進制，00 ~ FF)。
- S 設定 Steinhart-Hart 係數的指令。
- x A: 設定 Steinhart-Hart 係數 A。
B: 設定 Steinhart-Hart 係數 B。
C: 設定 Steinhart-Hart 係數 C。
- Ttt tt，表示欲設定 Steinhart-Hart 係數的 User-defined Type (70 ~ 77)。
- C(data) (data) 為 8 位數 16 進制值，以 IEEE-754 格式表示 Steinhart-Hart 係數。

回應：

有效指令: !AA [CHKSUM](CR)

無效指令: ?AA[CHKSUM](CR)

- ! 有效指令的定義符號。
- ? 無效指令的定義符號。
- AA 模組位址 (16 進制，00 ~ FF)。

註：如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例：

指令: @03SAT70C3A94030A	設定模組 (03) Type Code = 70 的 Steinhart-Hart 係數 A 為 3A94030A，相當於 1.129241×10^{-3} 。
回應: !03	回傳有效指令。
指令: @03SBT70C39757ACF	設定模組 (03) Type Code = 70 的 Steinhart-Hart 係數 B 為 39757ACF，相當於 2.341077×10^{-4} 。
回應: !03	回傳有效指令。
指令: @03SCT70C33BC73A5	設定模組 (03) Type Code = 70 的 Steinhart-Hart 係數 C 為 33BC73A5，相當於 8.775468×10^{-8} 。
回應: !03	回傳有效指令。

相關指令: [@AAGxTtt](#)、[@AARTTttR\(Data\)](#)

相關章節: 3.4.1 設定頁面 - User Defined Type、B.8.1 Steinhart-Hart 係數 (User-defined Type)

@AARTTttR(Data) 讀取電阻值的對應溫度 (User-defined Type)

說明:

使用 [@AASxTttC\(data\)](#) 指令設定好 Steinhart-Hart 係數後，即可使用此指令讀取電阻值所對應到的溫度值。**註:** 此指令適用於 **tM-TH8** 模組。

語法:

@AARTTttR(Data)[CHKSUM](CR)

- @** 定義符號字元 (Delimiter Character)。
- AA** 模組位址 (16 進制，00 ~ FF)。
- RT** 讀取輸入電阻之對應溫度的指令。
- Ttt** tt，表示 User-defined Type (70 ~ 77，[見 附錄 B.8 tM-TH8 資料範圍](#))。
- R(Data)** (Data) 包含 7 位數 10 進制值 或 5 個數字 與小數點附加 1 位數，表示輸入電阻。

回應:

有效指令: **!AA(Data) [CHKSUM](CR)**

無效指令: **?AA[CHKSUM](CR)**

- !** 有效指令的定義符號。
- ?** 無效指令的定義符號。
- AA** 模組位址 (16 進制，00 ~ FF)。
- (Data)** 包含 +/- 符號、3 位數 10 進制值 與 小數點附加 2 位數，表示電阻對應的溫度。

註: 如果指令的語法錯誤 或 通訊錯誤 或 沒有該位址的模組，都會沒有回應。

範例:

指令: @03RTT70R0104500	讀取模組 (03) Type Code = 70，電阻為 104500 歐姆的對應溫度。
回應: !03-032.64	回傳 -32.64 度。

指令: @03RTT70R00801.2	讀取模組 (03) Type Code = 70，電阻為 801.2 歐姆的對應溫度。
回應: !03+072.62	回傳 72.62 度。

相關指令: [@AAGxTtt](#)、[@AASxTttC\(data\)](#)

相關章節: 3.4.1 設定頁面 - User Defined Type、B.8.1 Steinhart-Hart 係數 (User-defined Type)

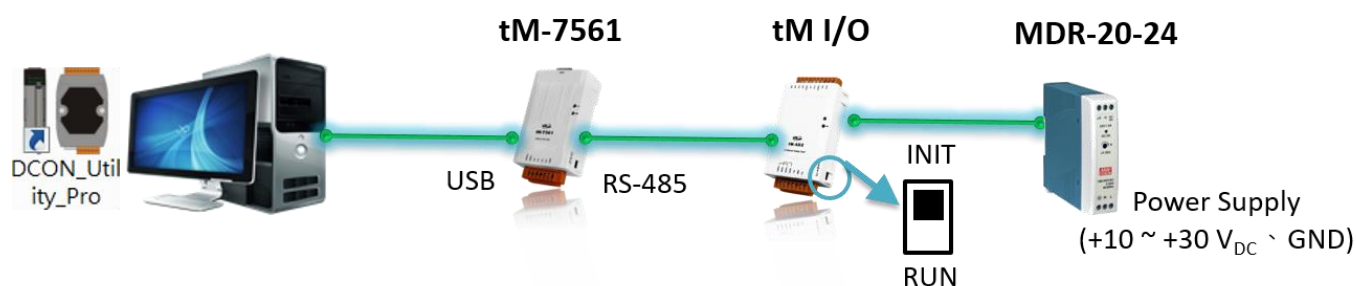
註: 可使用 [~AADT](#) 指令，設定溫度單位；使用 [~AAD](#) 指令讀取該設定。

附錄 F 快速儲存與套用 I/O 設定檔

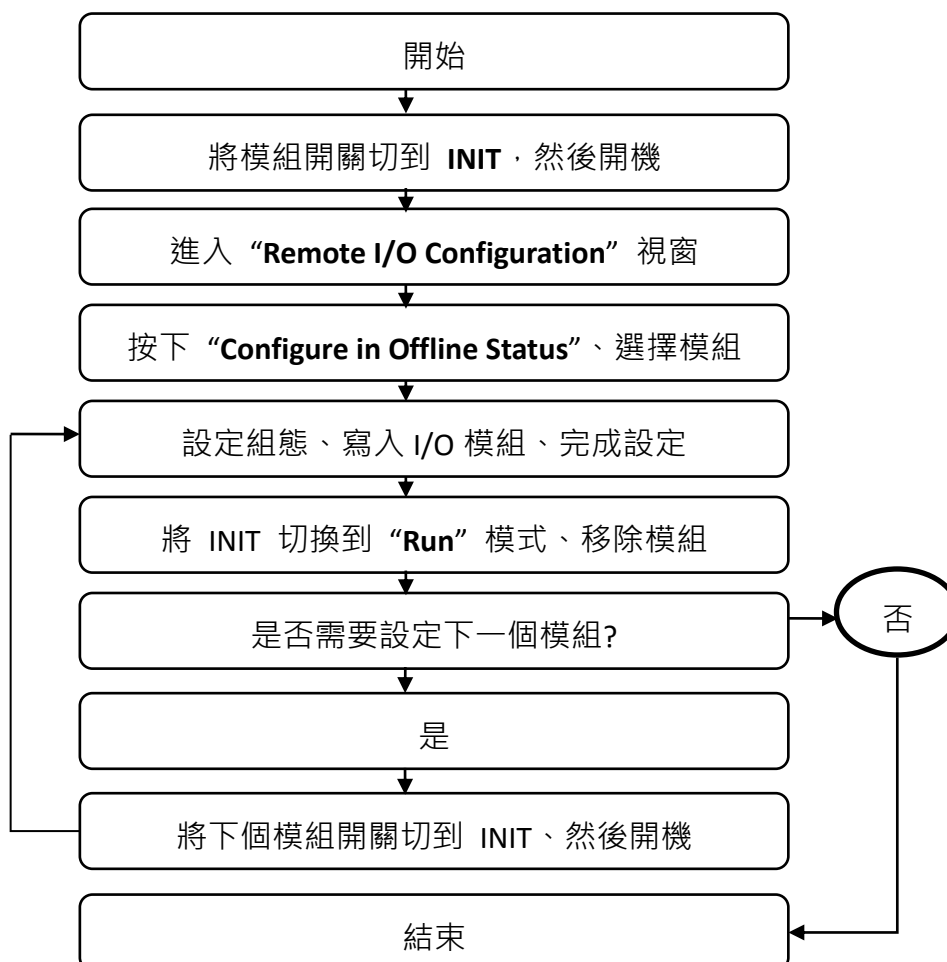
此章節將介紹 DCON Utility Pro 的“Remote I/O Configuration”功能，若一個案場需設定 250 個相同型號的模組，使用者可先設定好一個，再進行些微的修改，並一個個的寫入到模組。您也可將設定檔存起來，以便下次可快速地將設定寫入到新的模組。如此，以減輕使用者的設定負擔。

注意:

- 模組需在 Init 模式開機，才可寫入設定，且一次設定一個模組。
- 此功能可離線配置，不需連接模組也可設定並儲存檔案。
- 若使用 USB 轉 RS-485 轉換器，請確實安裝好驅動程式。

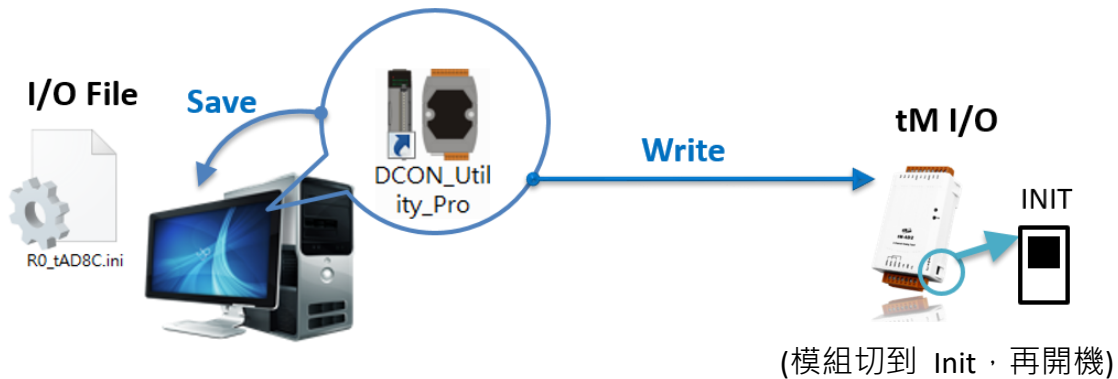


“Remote I/O INIT* Quick Configuration” 的設定流程:



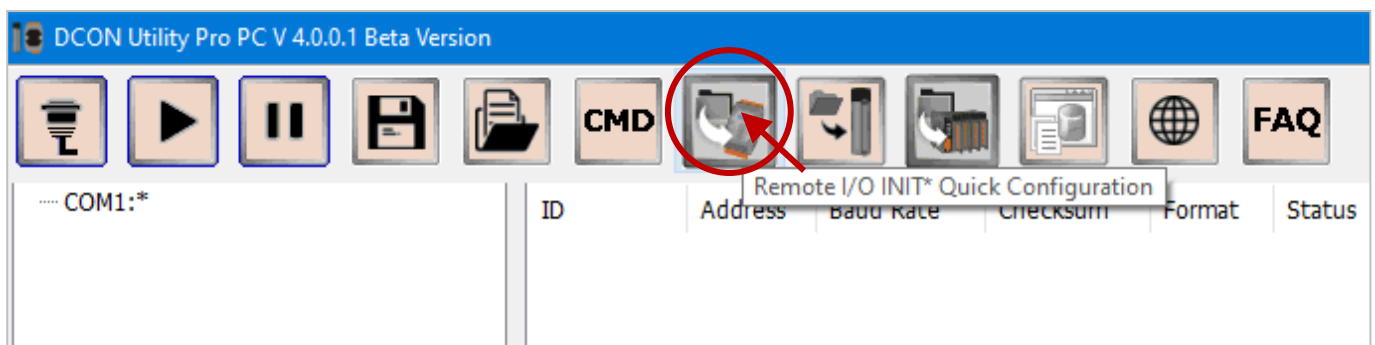
F.1 設定、寫入 與 儲存 I/O 組態

請依照以下步驟來設定並寫入模組組態，最後再存成設定檔。

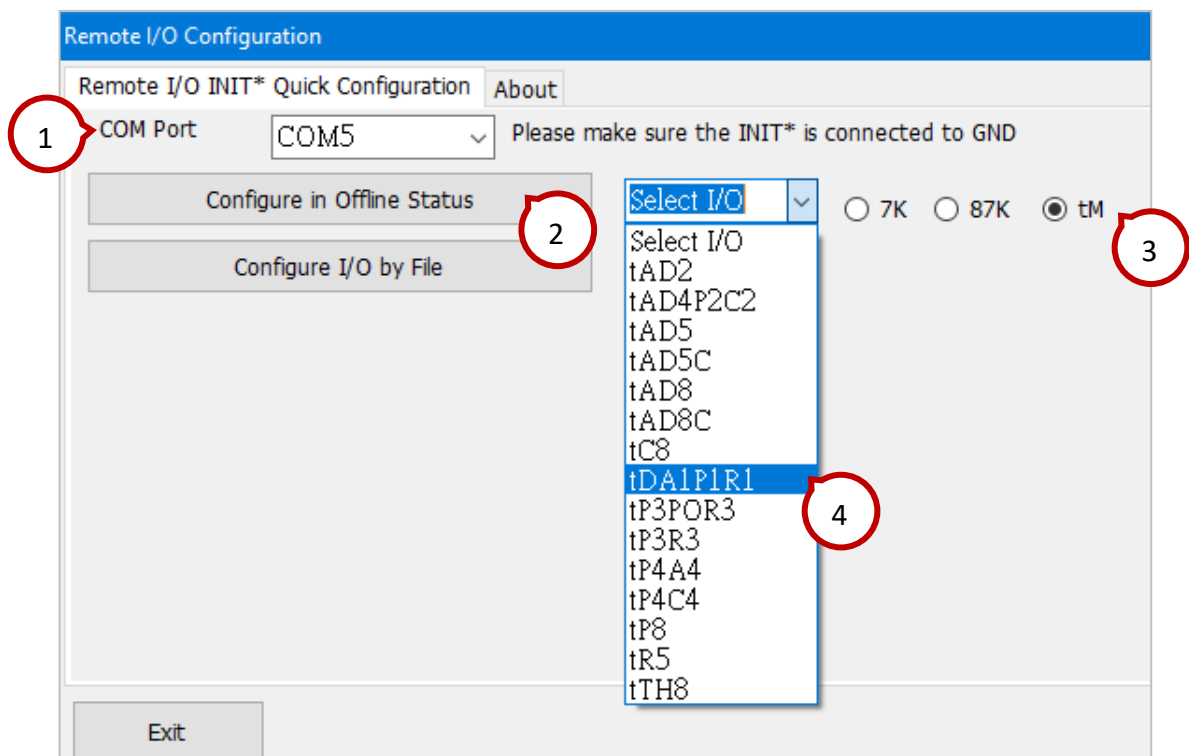


設定 I/O 組態

1. PC 上開啟 "DCON Utility Pro"，並點選 "Remote I/O INIT* Quick Configuration" 按鈕。



2. 請選取 COM Port (預設為 COM1)，點選 "Configure in Offline Status" 按鈕，並點選 "tM" 再選擇模組名稱。



3. 在設定視窗，會顯示模組 (例如: tM-DA1P1R1) 的初始設定，您可變更任何項目。

tDA1P1R1 Firmware[0000] [Offline Configuration]

Configuration AO DO DI Host WDT Commands Log Summary About

Protocol (INIT*) DCON

Address 0 [00H]

Baud Rate (INIT*) 9600

Parity (INIT*) N,8,1

Checksum (INIT*) Disabled

Analog Format Engineering Forma

Response Delay 0 ms

Exit Write Configurations to I/O Module Save configurations to the file

註: 在 Offline 模式，硬體不會隨著 IO 設定值而變化。

Configuration AO DO DI Host WDT Commands Log Summary About

Bit Status

☒ CH:00

DO Value 01h

Set [Power On Value] Set [Safe Value]

☒ Read DO

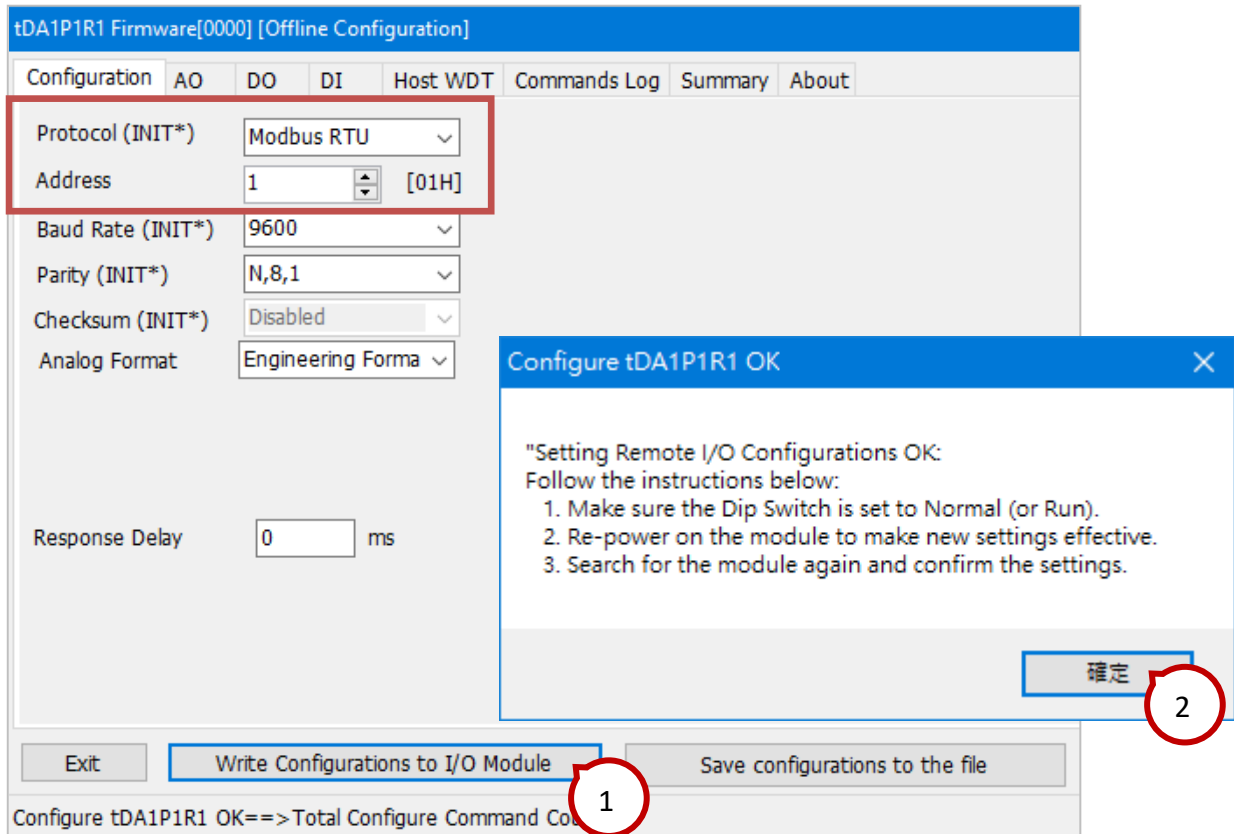
☐ Read Power ON Value

☐ Read Safe Value

寫入 I/O 組態

設定完成後，請確認 PC 已連接模組 (INIT 模式)，且設定好 COM Port (步驟 2)，然後繼續下一步驟。

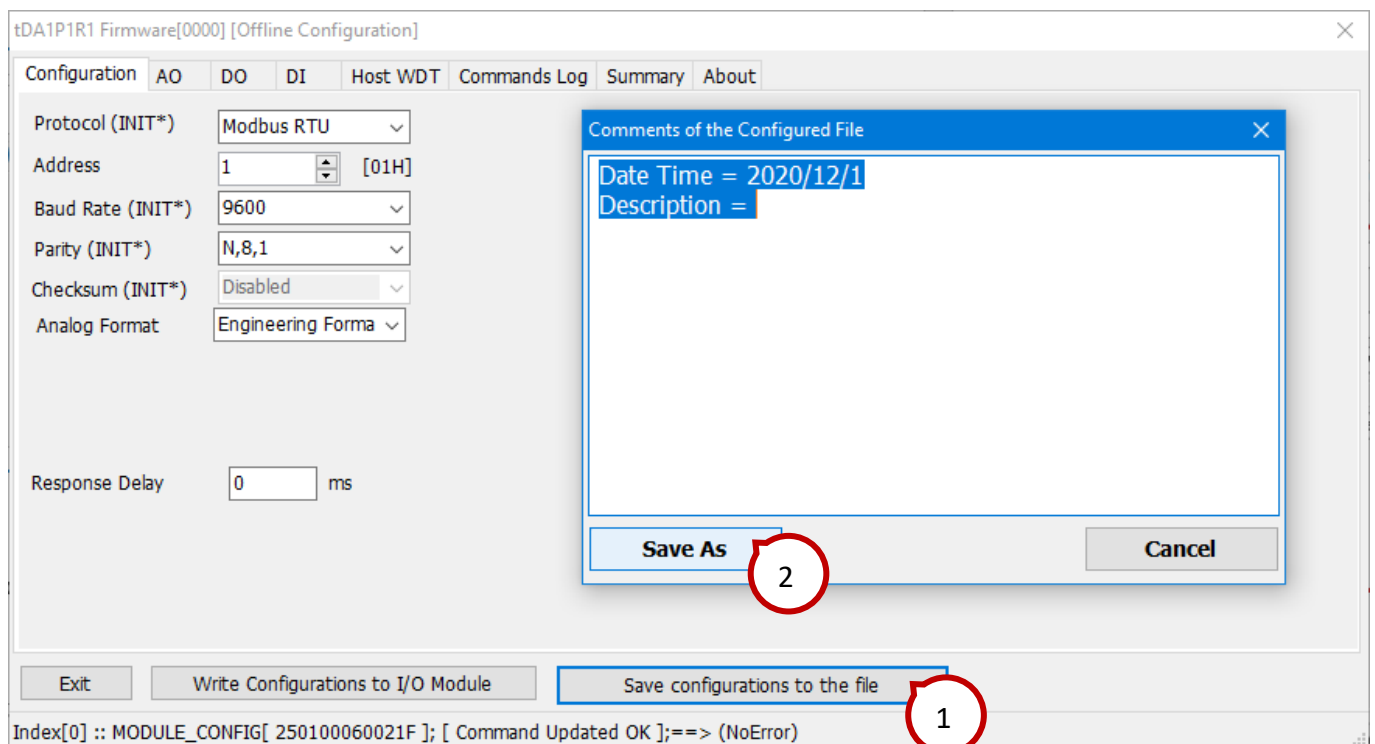
4. 點選 “Write Configurations to I/O Module” 按鈕，將設定值寫入到 I/O 模組，並點選 “確定” 按鈕。



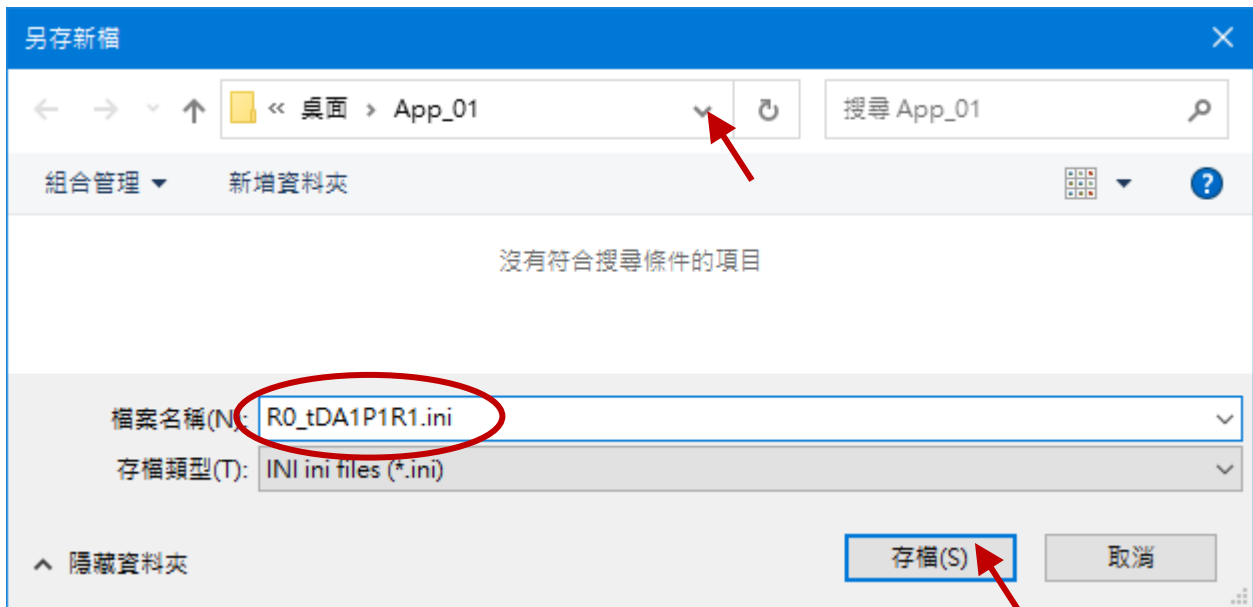
5. 將模組開關切回 **Run** 的位置，並移除移除模組。如需設定下一個，不需關閉設定視窗。
6. 裝上同型號的模組 (DIP Switch = Init)，接著重複步驟 3 ~ 6 來進行設定 (例如: Address=2)。

儲存 I/O 組態

7. 完成所有模組設定後，點選 **“Save Configuration to the File”** 按鈕，再點選 **“Save As”** 按鈕。

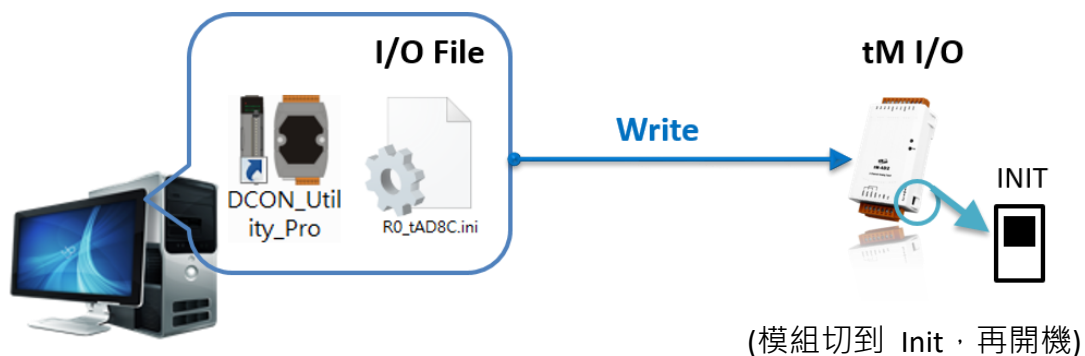


可以根據專案名稱建立資料夾 (例如: App_01) , 並將其存成設定檔供下次訂單使用。



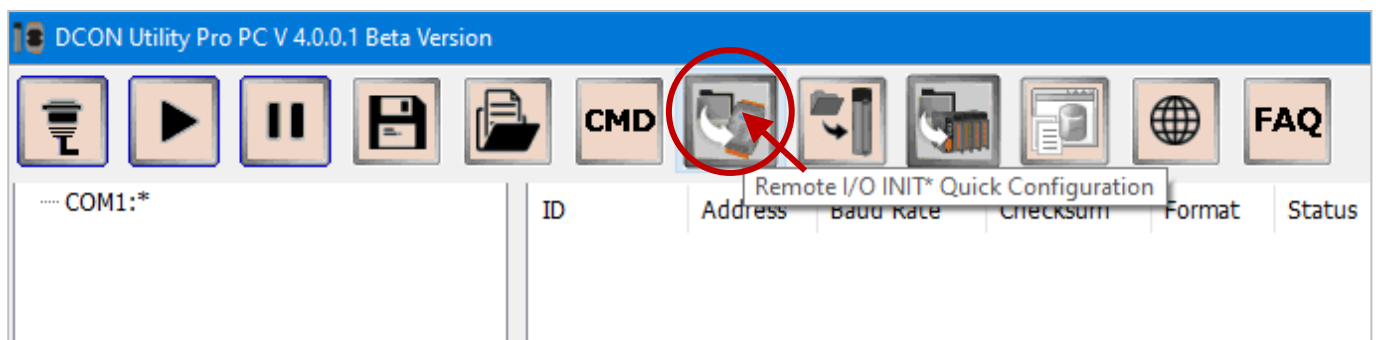
F.2 使用設定檔來配置模組

接下來將說明，如何開啟之前存好的設定檔，並寫入到模組中。

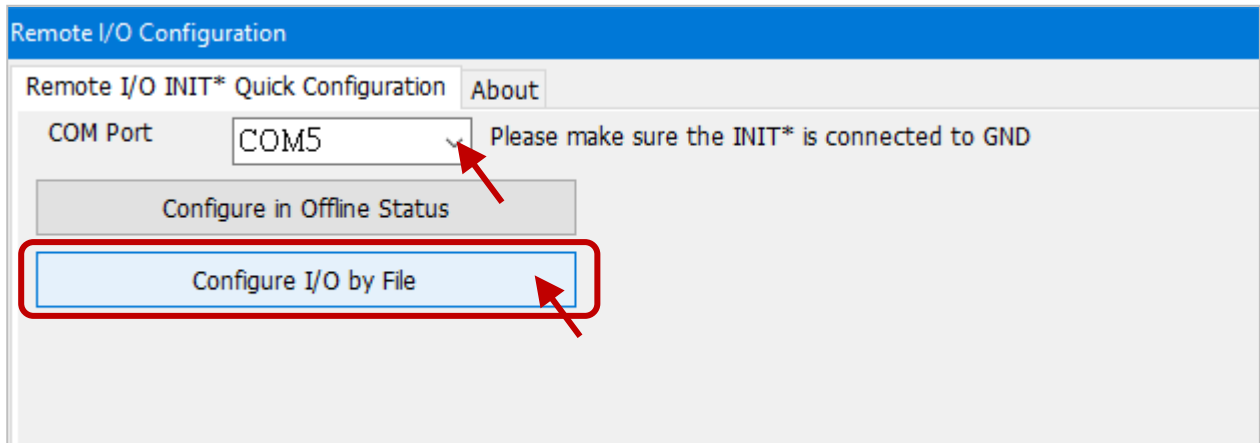


注意: 請在 Init 模式下 進行此設定，且一次設定一個模組。

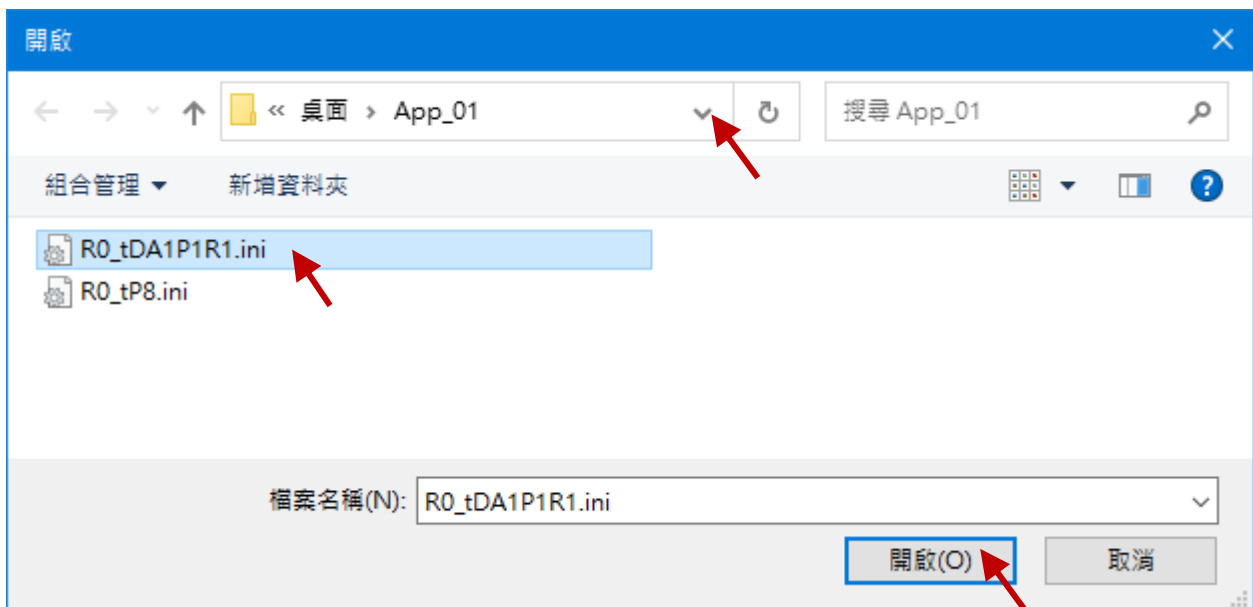
1. PC 上開啟 "DCON Utility Pro", 並點選 "**Remote I/O INIT* Quick Configuration**" 按鈕。



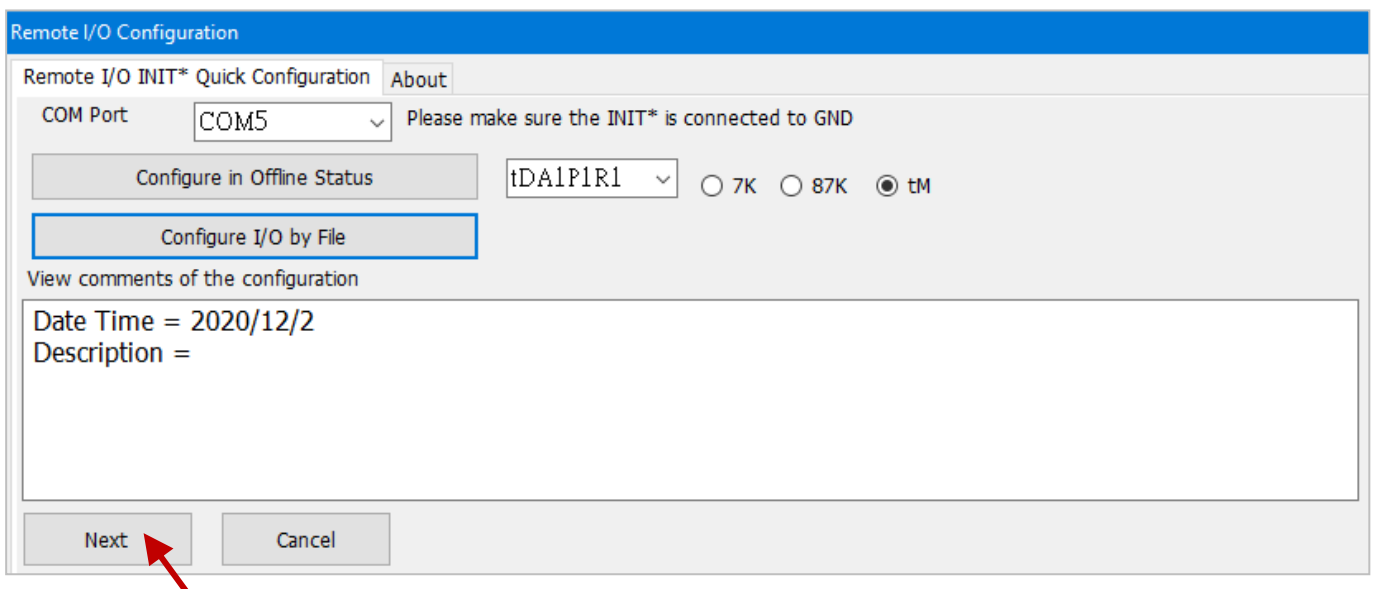
2. 在 “Remote I/O Configuration” 視窗，選擇 COM Port，並點選 “Configure I/O by File” 按鈕。



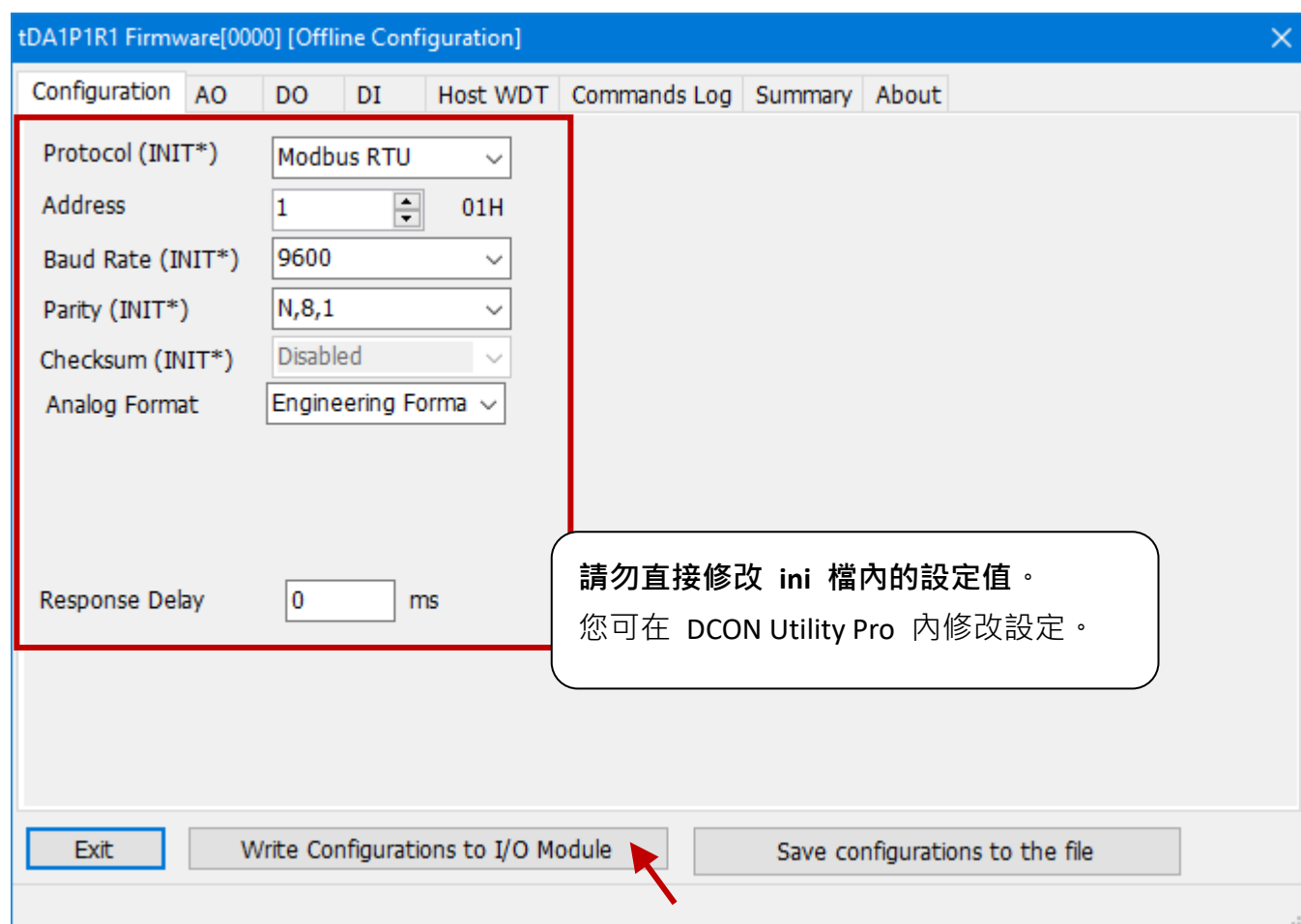
3. 選擇設定檔 (.ini) 並點選 “開啟” 按鈕。



在擴展的文字框下，點選 “Next” 按鈕。



4. 確認/修改設定值，點選 “**Write Configurations to I/O Module**” 按鈕，將設定值寫入到 I/O 模組。
完成後，將指撥開關切換到 **Run** 的位置，再重新開機來套用設定。



附錄 G 儲存或載入 I/O 專案

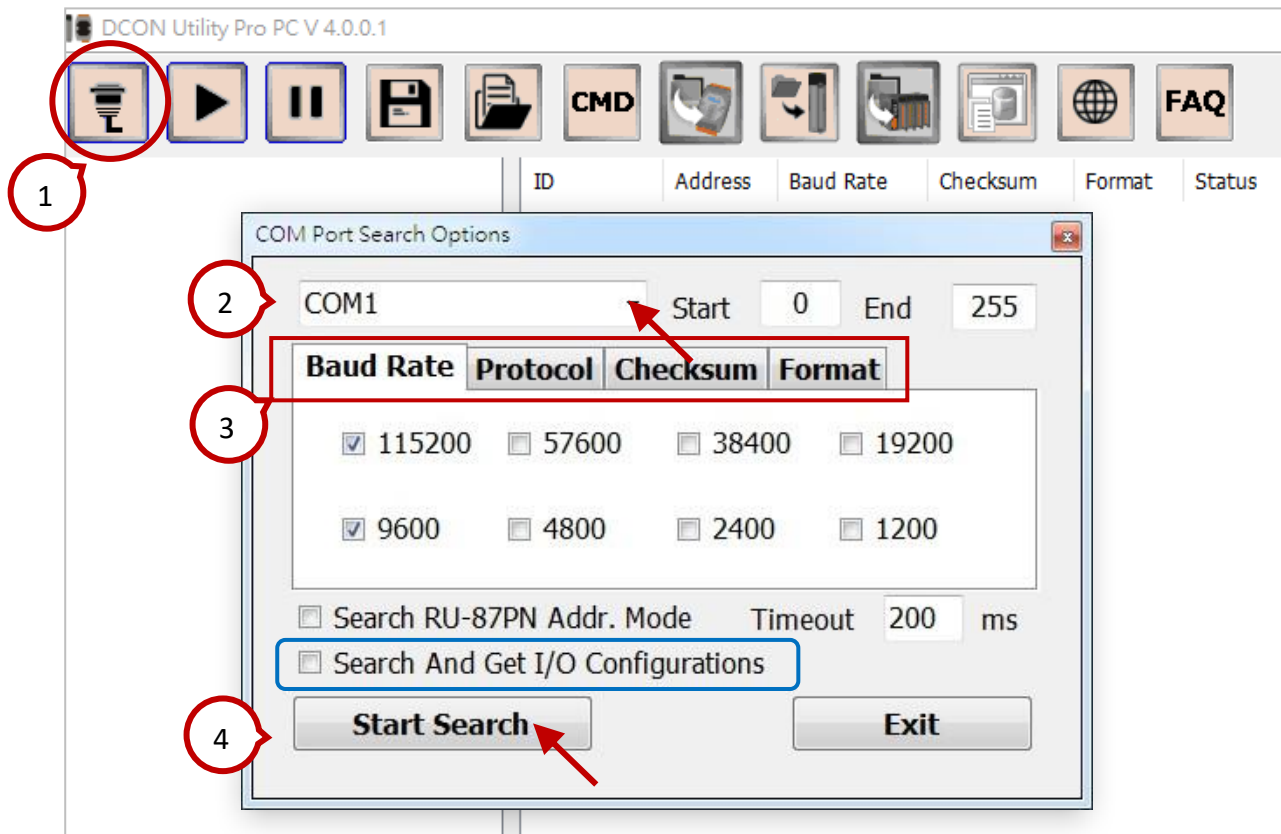
DCON Utility Pro v4.0 新增了**儲存/載入專案**的功能，當客戶的案子用到很多模組時，您可透過不同的 COM Port 來搜尋已設定好的模組，並將其存成一個專案，也可使用 ZIP 工具將專案加密。之後，如需更換模組 或 建立相同的系統，您可載入專案檔，來自動比對新模組的參數設定、型號與 使用的 COM Port，是否符合原系統正常運作的需求。

G.1 儲存專案 (Save Project)

請依照下列步驟來，儲存專案。

1. 搜尋 I/O 模組。


點選 **COM Port** 按鈕，選擇 COM Port 與 搜尋條件，再點選 **Start Search** 按鈕。



註:

- 勾選 “Search and Get I/O Configurations” 可搜尋並載入模組的設定值。若未勾選，也可開啟模組設定畫面，來載入設定值。
- 如果還有其它 COM Port 有連接模組，可用相同方式再搜尋一次。

如果上一步驟未勾選 “Search and Get I/O Configurations” 設定，儲存專案前，請開啟指定模組的設定畫面，來載入設定值。



ID	Address	Baud Rate	Checksum	Format	Status	Description	Comments
tR5	1[01h]	115200	Disabled	N,8,1	Remote I/O	[Modbus RTU]5*DO (Relay DO)	Supported
tAD8C	2[02h]	115200	Disabled	N,8,1	Remote I/O	[Modbus RTU]8*AI	Supported
tTH8	4[04h]	115200	Disabled	N,8,1	Remote I/O	[Modbus RTU]8*AI (Universal Thermistor)	Supported

tR5 Firmware[A106]

Configuration DO Host WDT Commands Log Summary About

Protocol: Modbus RTU

Address: 1 01H

Baud Rate: 115200

Parity: N,8,1

Checksum: Disabled


Response Delay: 0 ms

Set Module Configurations

Exit

2. 儲存 I/O 專案。

請點選 “Save Project” 按鈕、輸入檔案名稱，再點選 “存檔” 按鈕，來儲存專案內使用的模組與設定參數。



ID	Address	Baud Rate	Checksum	Format	Status	Description	Comments
tR5	1[01h]	115200	Disabled	N,8,1	Remote I/O	[Modbus RTU]5*DO (Relay DO)	Supported
tAD8C	2[02h]	115200	Disabled	N,8,1	Remote I/O	[Modbus RTU]8*AI	Supported
tTH8	4[04h]	115200	Disabled	N,8,1	Remote I/O	[Modbus RTU]8*AI (Universal Thermistor)	Supported

若搜尋多個
COM Port，
會列於此處。

另存新檔

<< DCON_Utility_Pro_PC > search > project

組合管理 新增資料夾

名稱 修改日期 類型 大小

沒有符合搜尋條件的項目

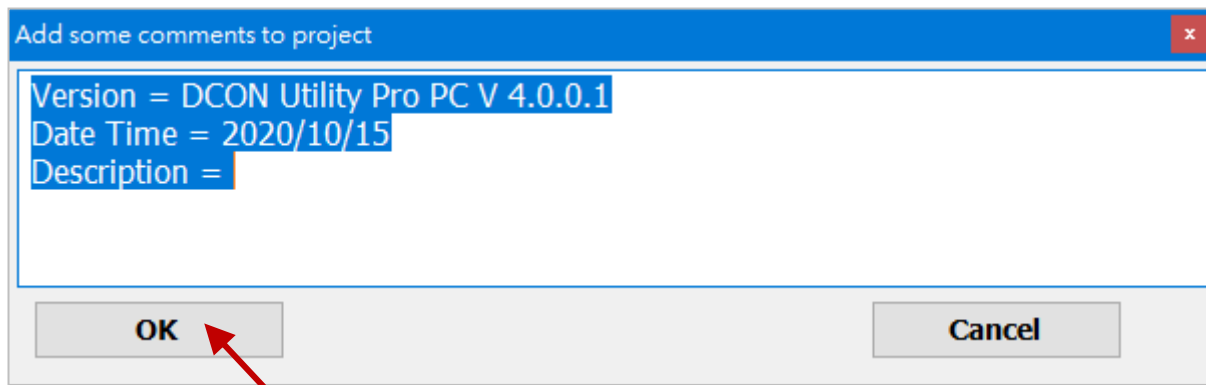
檔案名稱(N): Case01

存檔類型(T):

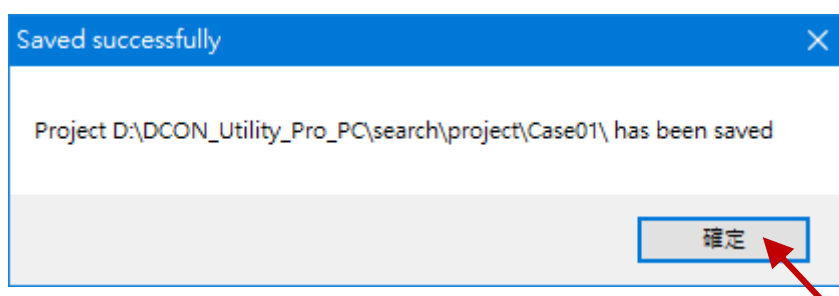
隱藏資料夾

存檔(S) 取消

接著，會出現專案的註解視窗，您可為此專案輸入註解並點選 OK。

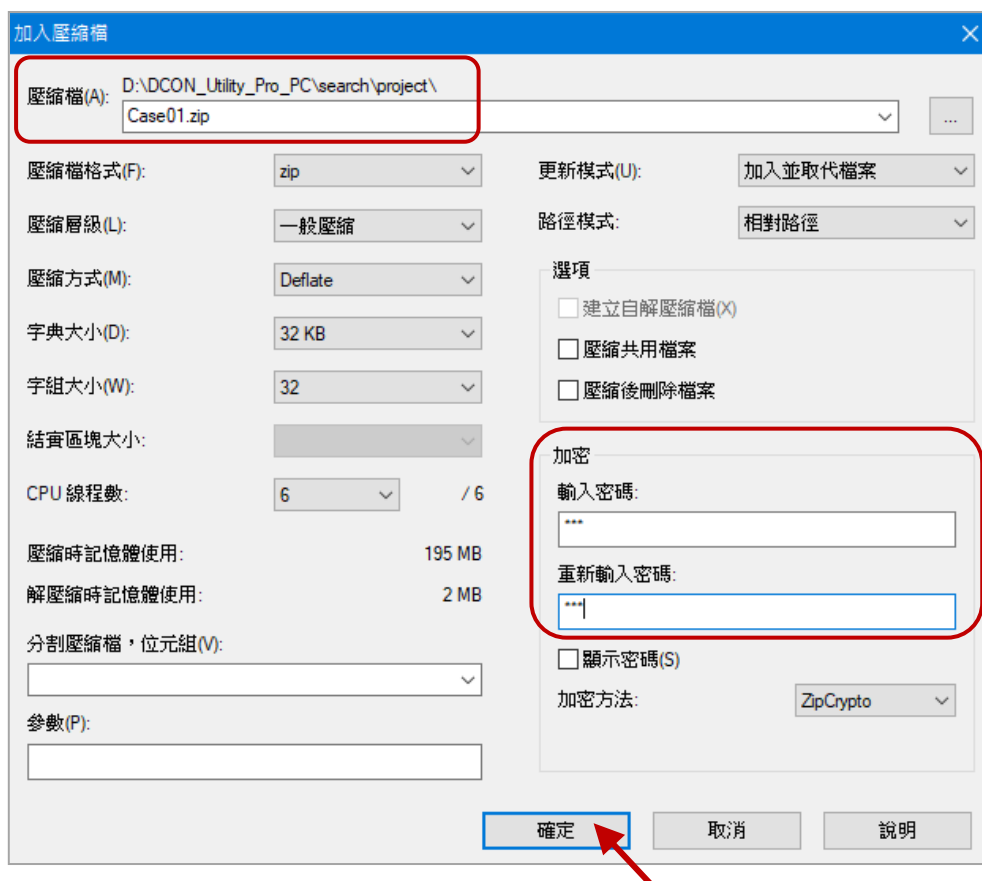


儲存成功，會出現此訊息視窗。



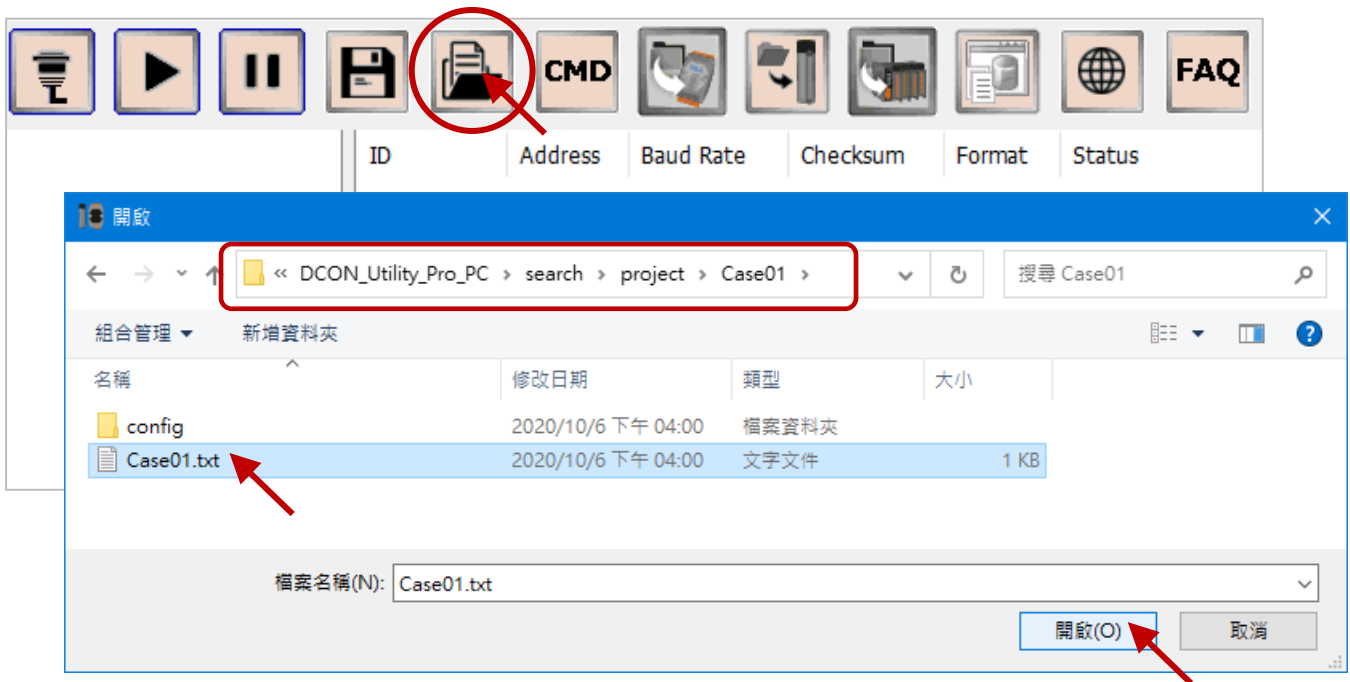
注意:

I/O 專案會儲存在 ..\DCON_Utility_Pro_PC\search\project，建議備份一份專案在 PC 上，以避免更換 DCON Utility Pro 版本時，移除了舊有的目錄。此外，您也可使用 ZIP 工具將專案檔加密。



G.2 載入專案 (Load Project)

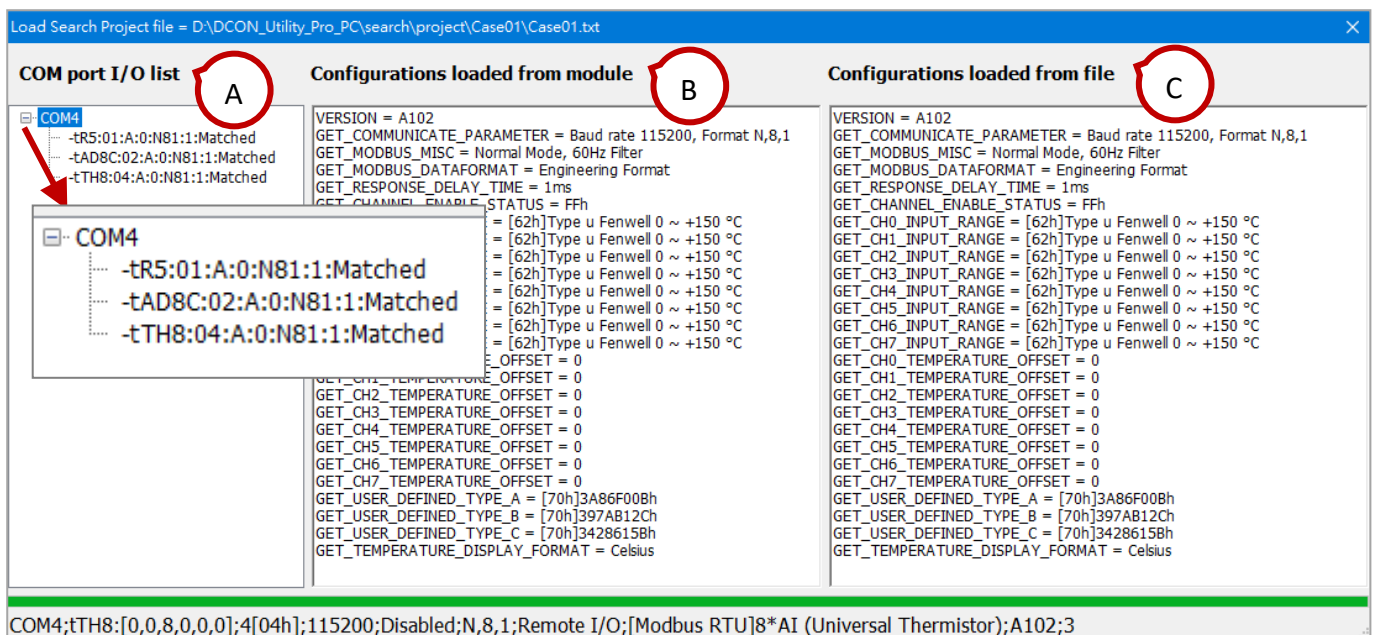
點選 “Load Project” 按鈕，並開啟專案目錄下 (例如: Case01) 的專案檔 (.txt)。



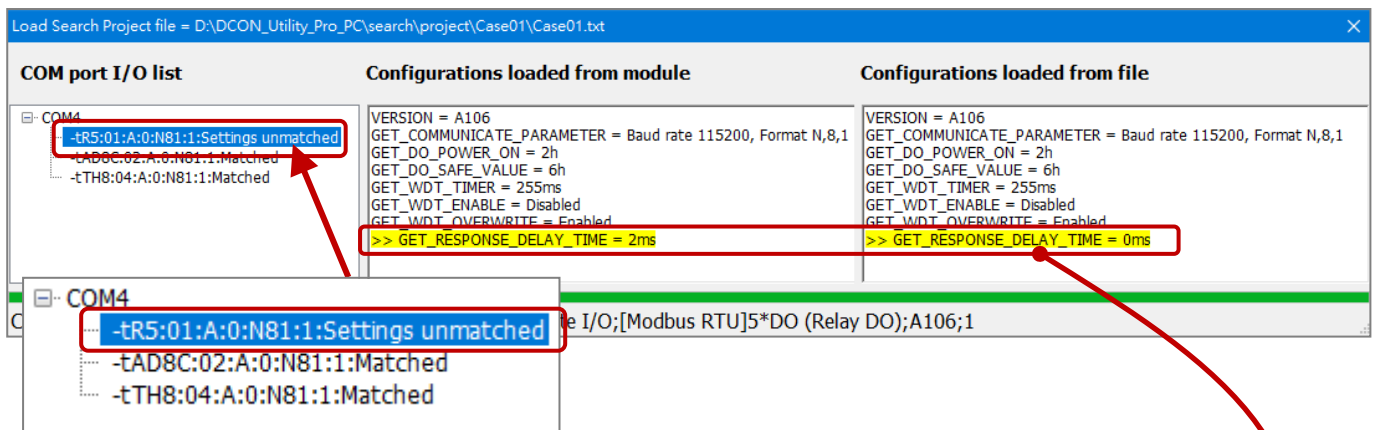
註：點選開啟按鈕後，會顯示專案的註解視窗 (參考上一頁)。

若您的 PC 有連接模組，且設定無誤，載入專案後會顯示以下視窗：

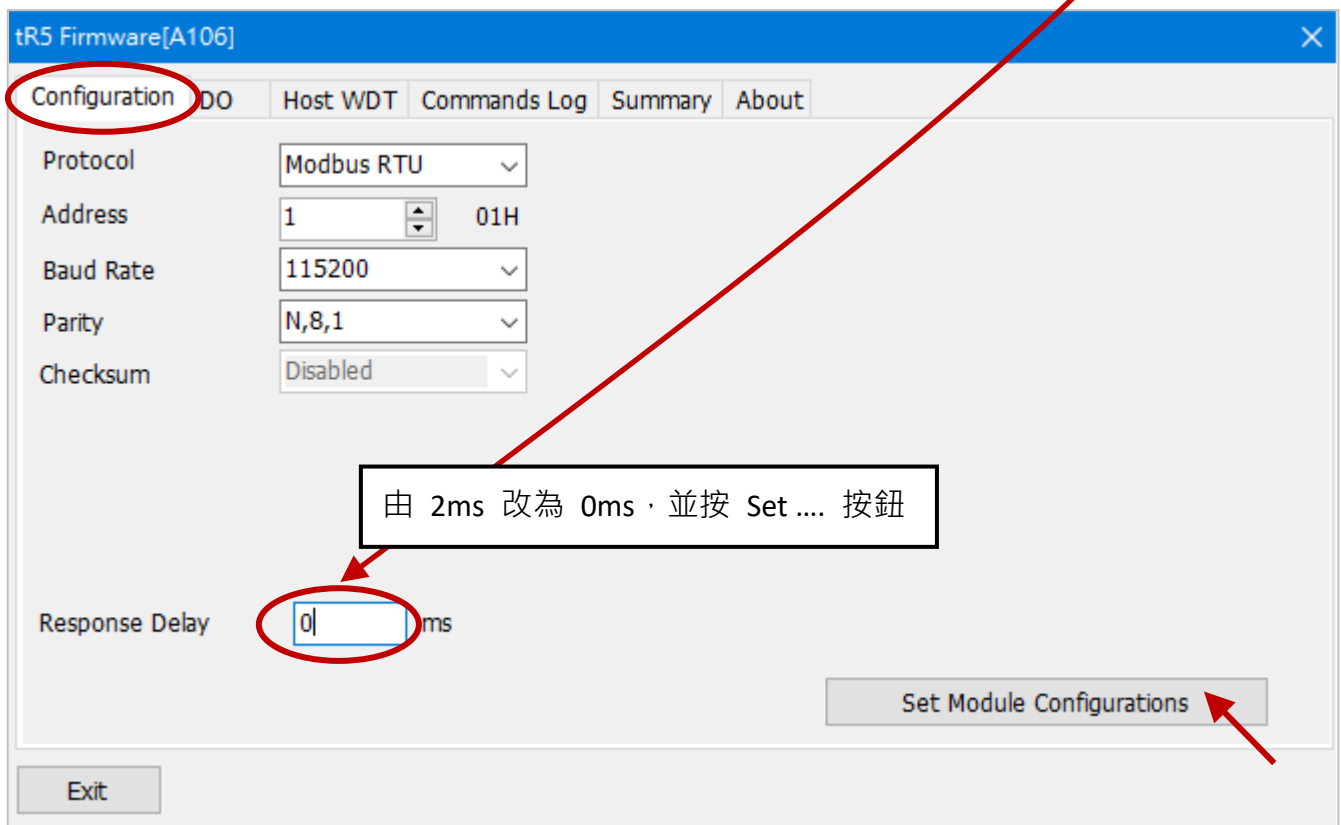
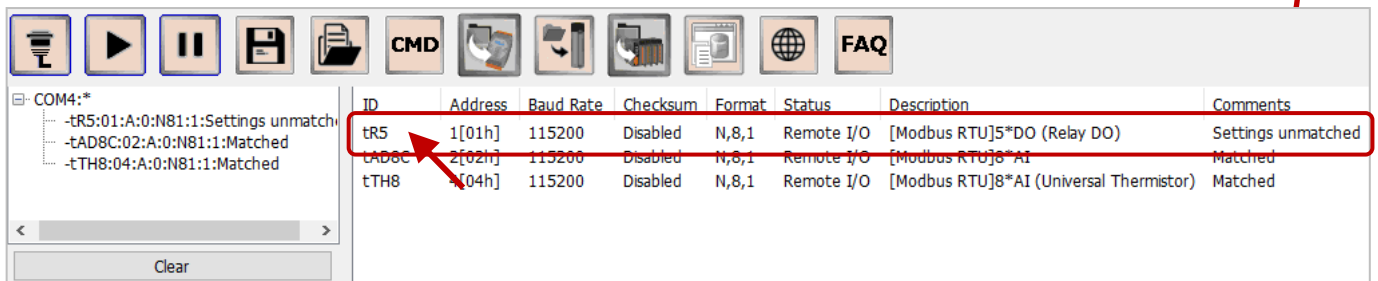
- A. **COM Port I/O list:** 顯示由專案檔載入的 COM port 編號、模組清單 與 模組的比對狀態。您可點選每一筆來查看設定。
- B. **Configurations Loaded from module:** 顯示目前的模組設定。
- C. **Configurations Loaded from file:** 顯示由專案檔載入的模組設定。



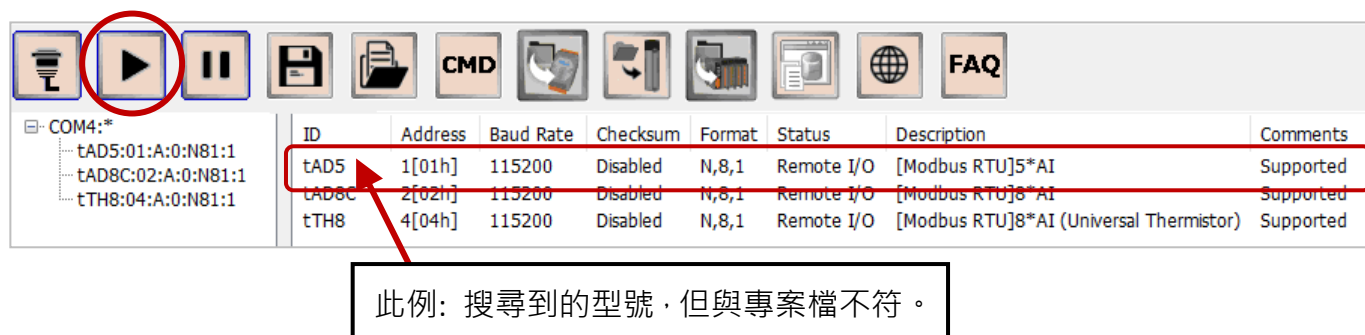
若 PC 有連接模組，但設定有誤，載入專案後，該型號旁會顯示 “Settings unmatched” 並標示出錯誤的地方。您可關閉此視窗，再去修改模組設定。



請修改為和專案檔內相同的設定值，此例，將 Response Delay 改為 0 ms。

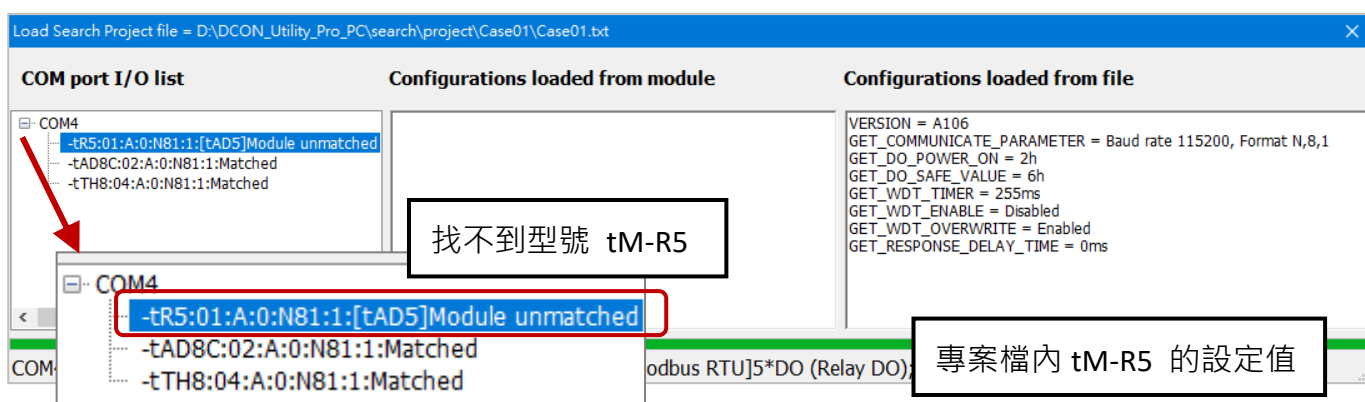


若 PC 有連接模組，但通訊參數 或 型號有誤，載入專案後，該型號旁會顯示 “Module not found” 或 “Module unmatched” 且只能查看載入的模組設定值。



ID	Address	Baud Rate	Checksum	Format	Status	Description	Comments
tAD5	1[01h]	115200	Disabled	N,8,1	Remote I/O	[Modbus RTU]5*AI	Supported
tAD8C	2[02h]	115200	Disabled	N,8,1	Remote I/O	[Modbus RTU]8*AI	Supported
tTH8	4[04h]	115200	Disabled	N,8,1	Remote I/O	[Modbus RTU]8*AI (Universal Thermistor)	Supported

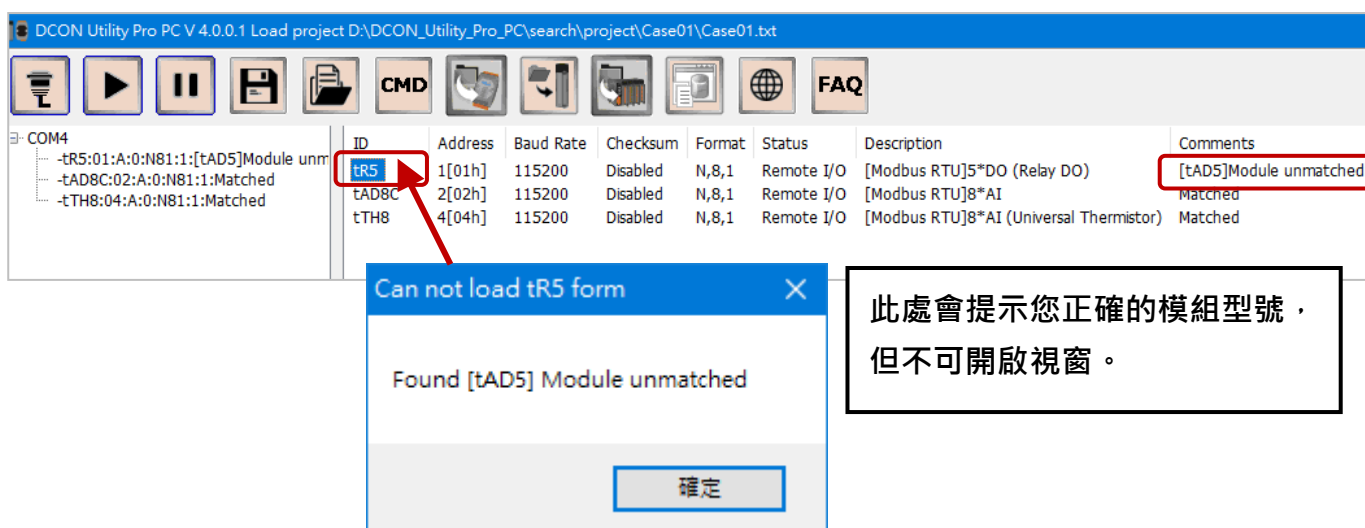
此例：搜尋到的型號，但與專案檔不符。



找不到型號 tM-R5

專案檔內 tM-R5 的設定值

關閉 Load... 視窗後，會顯示正確的模組型號，此時，您將無法開啟該設定畫面，請更換對的模組並設定正確的通訊參數，再執行 Load Project 進行比對。



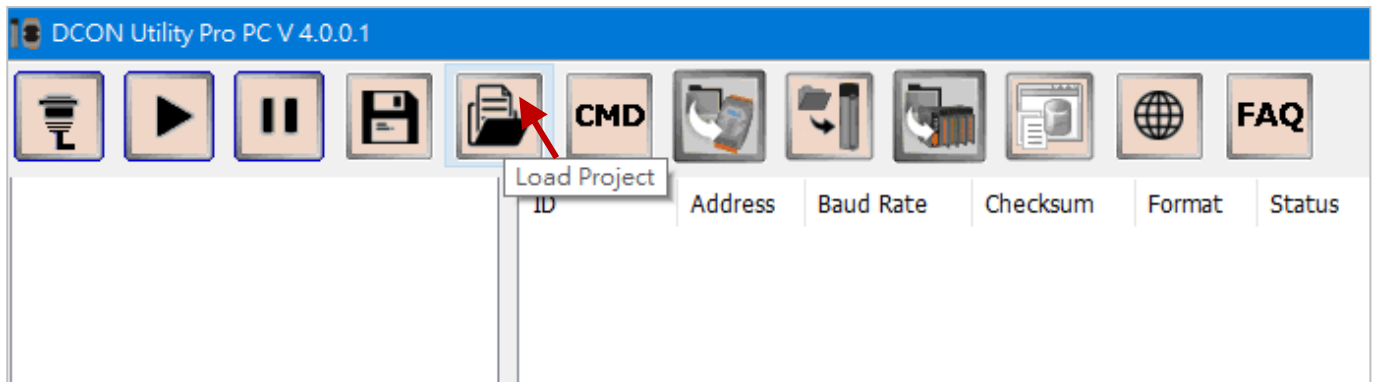
Can not load tR5 form

Found [tAD5] Module unmatched

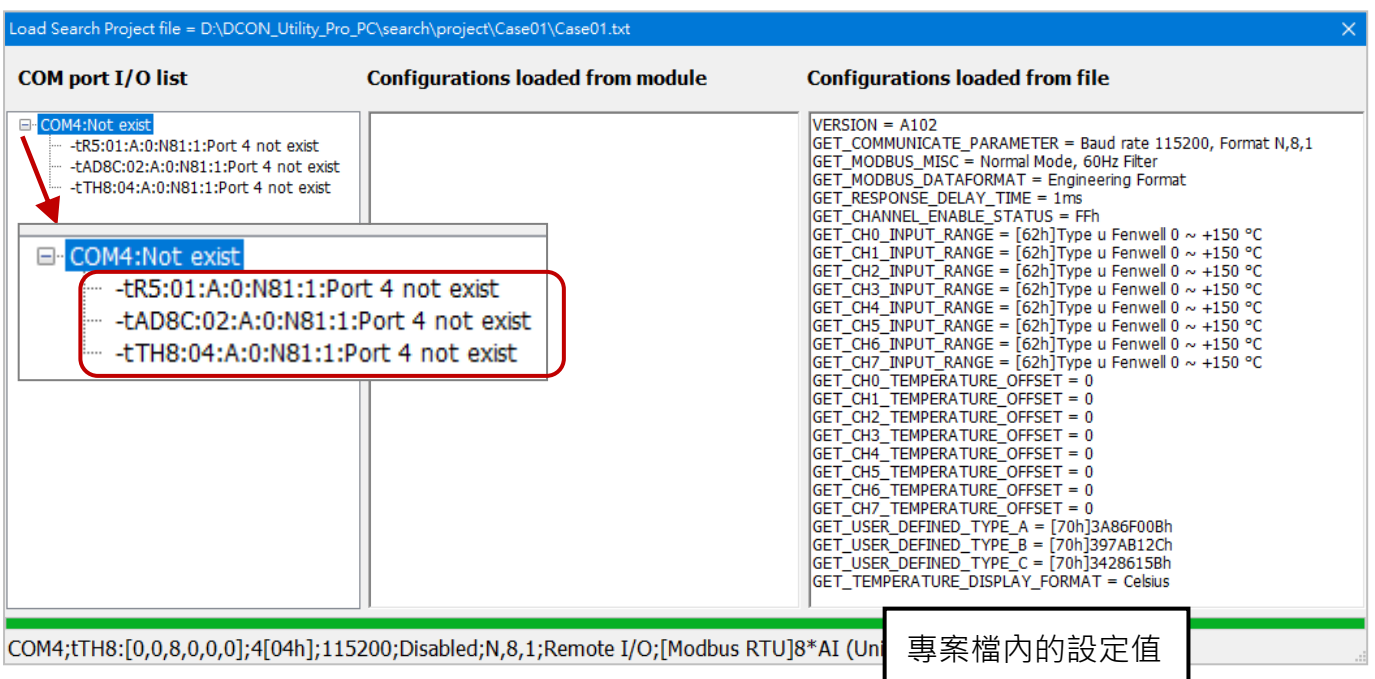
確定

此處會提示您正確的模組型號，但不可開啟視窗。

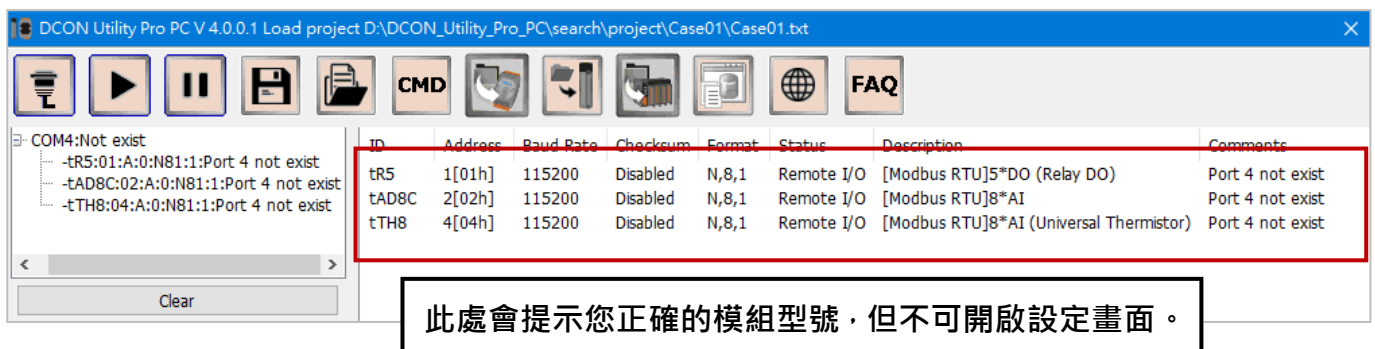
若 PC 未連接 I/O，載入專案後，該型號旁會顯示 “Not exist” 且只能查看專案檔內的設定值。



可點選每一個型號，來查看設定值。



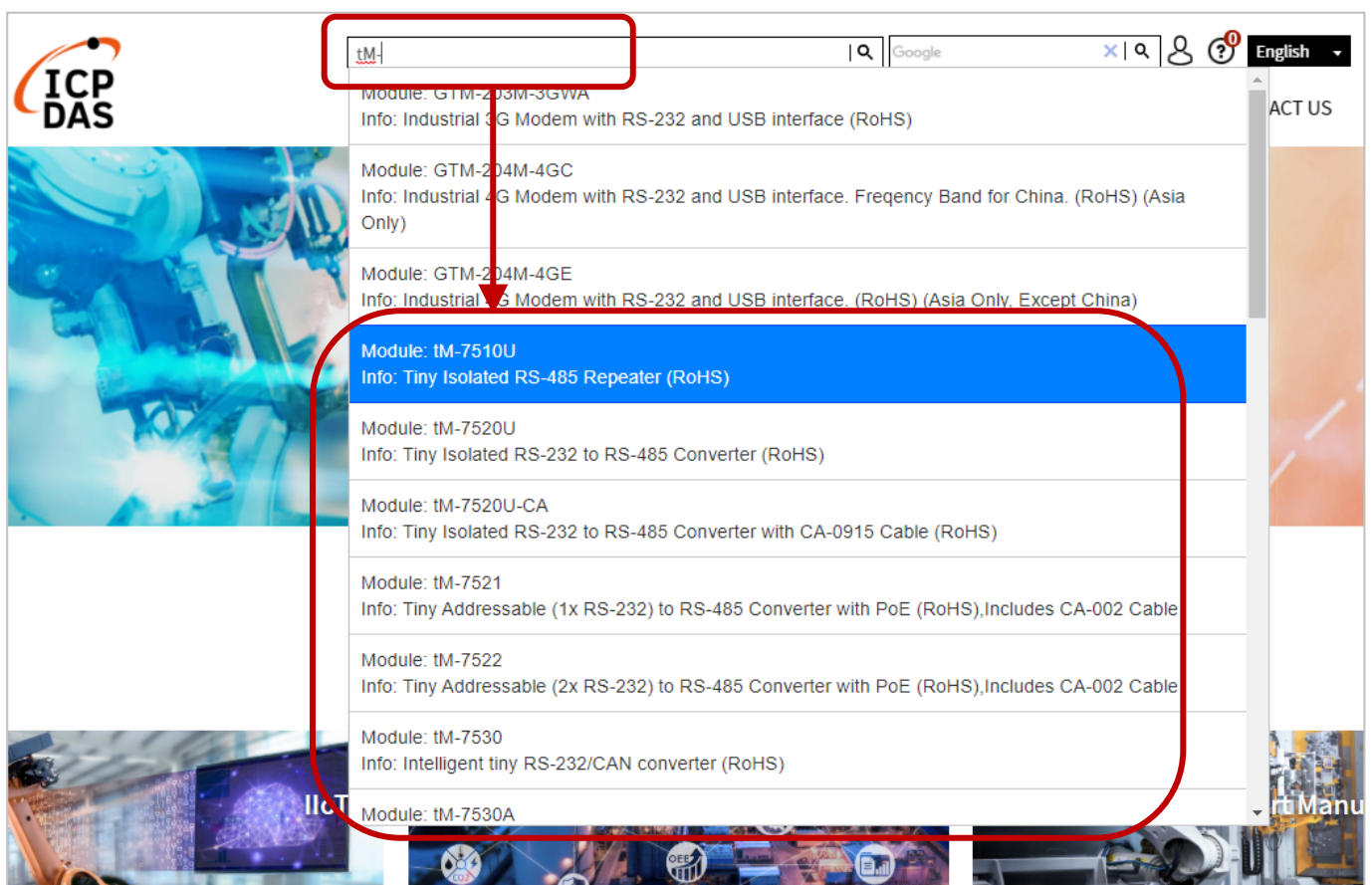
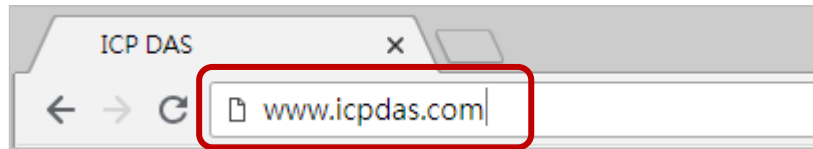
關閉 Load... 視窗後，會顯示正確的模組型號，請連接模組並設定正確的通訊參數，再執行 Load Project 進行比對。



附錄 H 其他資訊

H.1 如何搜尋產品網頁?

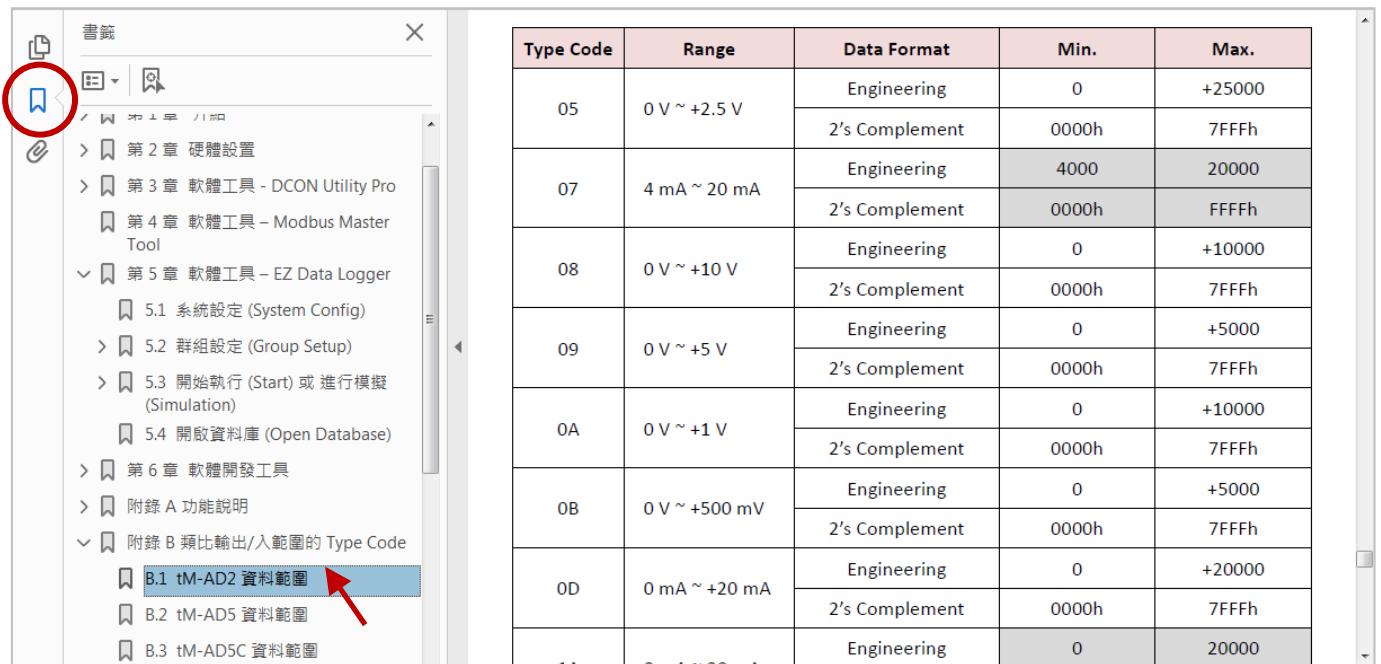
請在泓格科技的首頁 (www.icpdas.com)，於搜尋框中輸入關鍵字 (例如: tM)，即可找到相關的產品。



H.2 如何快速查詢章節？

1) 在 PC 上查看 PDF

請點選左側的“書籤”按鈕，再點選需查看的章節編號。

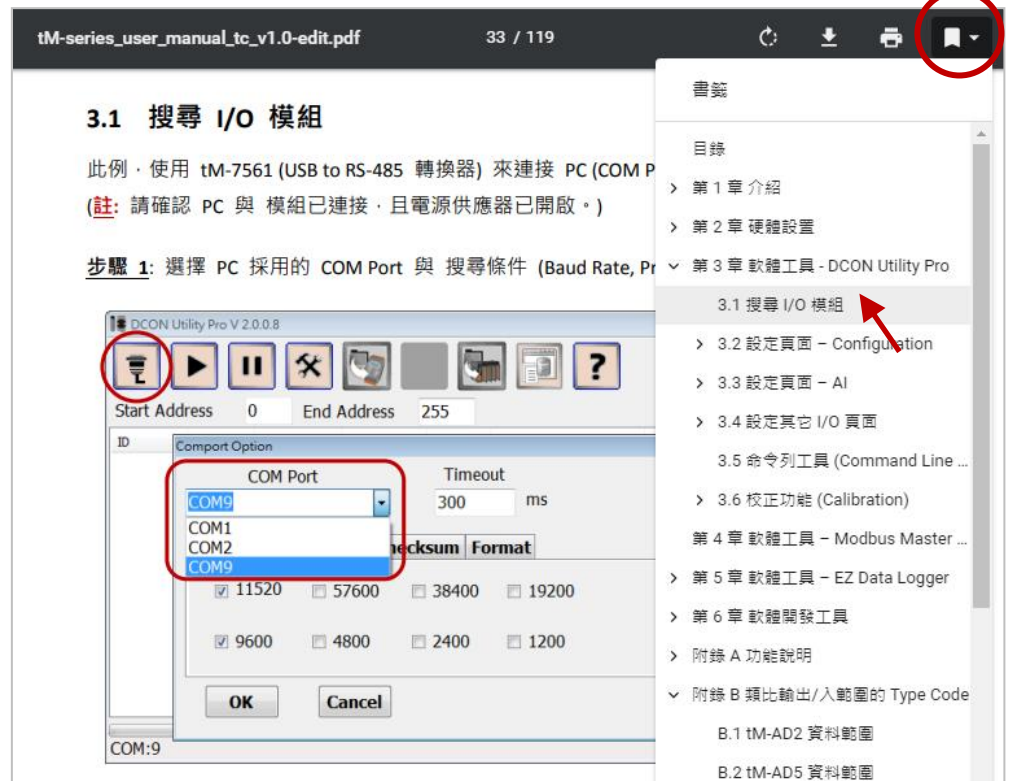


2) 在瀏覽器上查看 PDF

依據不同的瀏覽器，您可在上方 或 點選一下頁面，來找到“書籤”按鈕。



Firefox



Chrome



IE