

Modbus Slave Tool User Manual

V1.00, Aug. 2025



Written by Mac Cho
Edited by Sunny Chiu

目錄

1. MODBUS 通訊的資料格式差異	3
1.1. 暫存器位址基準的差異	4
1.2. 傳輸資料格式的差異	7
2. MODBUS SLAVE TOOL	8
2.1. 前置作業	9
2.2. 設定 MODBUS SLAVE TOOL	10

1. Modbus 通訊的資料格式差異

Modbus 是一種通訊協定，是 Modicon 公司於 1979 年為使用可程式化邏輯控制器 (PLC) 通訊而發表。現在已經成為工業通訊的業界標準，廣泛使用於各種自動化的設備。Modbus 通訊協定可分為串列通訊和乙太網路通訊兩大類型：串列通訊包括 Modbus RTU 及 Modbus ASCII 兩種協定，以 RS-232、RS-485 或 RS-422 為常用的通訊介面，適合短距離、多設備以及低成本考量的應用。乙太網路通訊包含 Modbus TCP，透過乙太網路傳輸資料，常用於需要網路連接的自動化設備或遠端監控系統。

雖然大部分的 Modbus 設備都遵循同樣的標準，但不同的製造商生產的設備之間仍然存在著一些差異。而這些微小的差異常常讓工程師花費很多時間與精力在測試與除錯上。Modbus Slave Tool 能協助使用者更快了解軟體和硬體之間是否使用正確且相同的資料格式進行通訊。

Modbus Master 軟體與 Slave 設備之間可能出現的差異包含暫存器位址的基準與資料傳輸的格式。

1.1. 暫存器位址基準的差異

Modbus 協定的資料分為四大類。

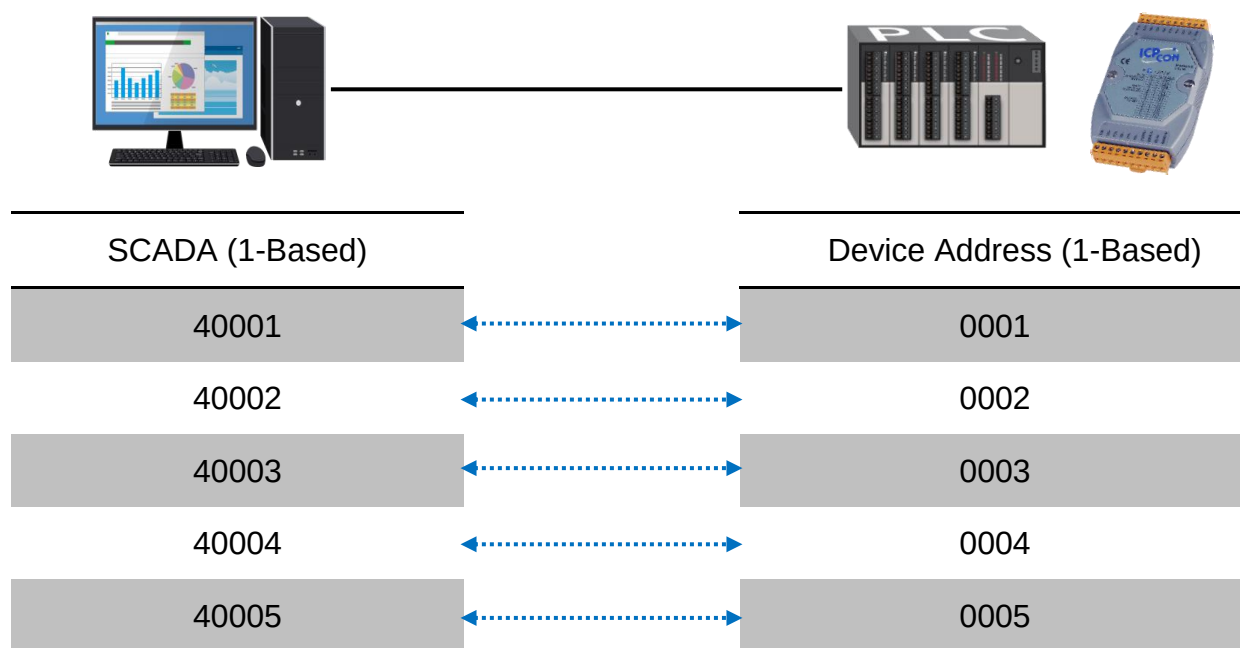
暫存器類型	PLC/SCADA	Modbus 位址	簡稱	讀寫	資料
Coils	00001 ~ 09999	0000H~FFFFH	0x	讀寫	DO
Discrete Inputs	10001 ~ 19999	0000H~FFFFH	1x	唯讀	DI
Input Registers	30001 ~ 39999	0000H~FFFFH	3x	唯讀	AI
Holding Registers	40001 ~ 49999	0000H~FFFFH	4x	讀寫	AO

四大類資料可分別對應到設備的 I/O 動作

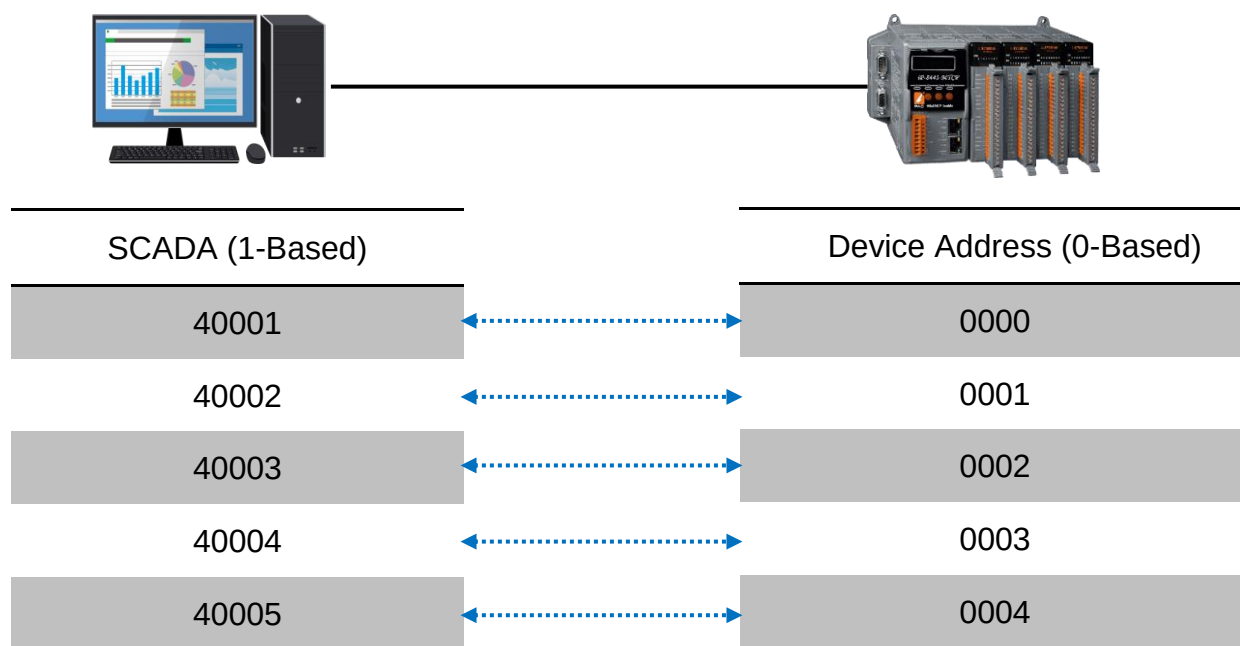
位址	說明
0xxxx	輸出或讀回設備的 DO 通道的資料
1xxxx	讀回設備的 DI 通道的資料
3xxxx	讀回設備的 AI 通道的資料
4xxxx	輸出或讀回設備的 AO 通道的資料

以 4xxxx 為範例，4 表示 Modbus 的暫存器類型，xxxx 則是通訊時使用的暫存器位址。大部分的 PLC 或 SCADA 軟體的文件中，第一個位址是從 1 開始 (1-based)，但設備內部的記憶體，則是從 0 開始 (0-based)。因此會看到使用者容易理解 1-based 與工程常用的 0-based 兩種資料位址的寫法。如果硬體的文件和軟體的位址基準都是從 1 開始，那麼軟體讀回的資料會與硬體文件的敘述相同。如果軟體是 1-Based，而硬體文件是 0-Based，那麼從這個軟體讀硬體的資料時，讀到的硬體資料為 [軟體位址-1] 的硬體文件位址的資料。

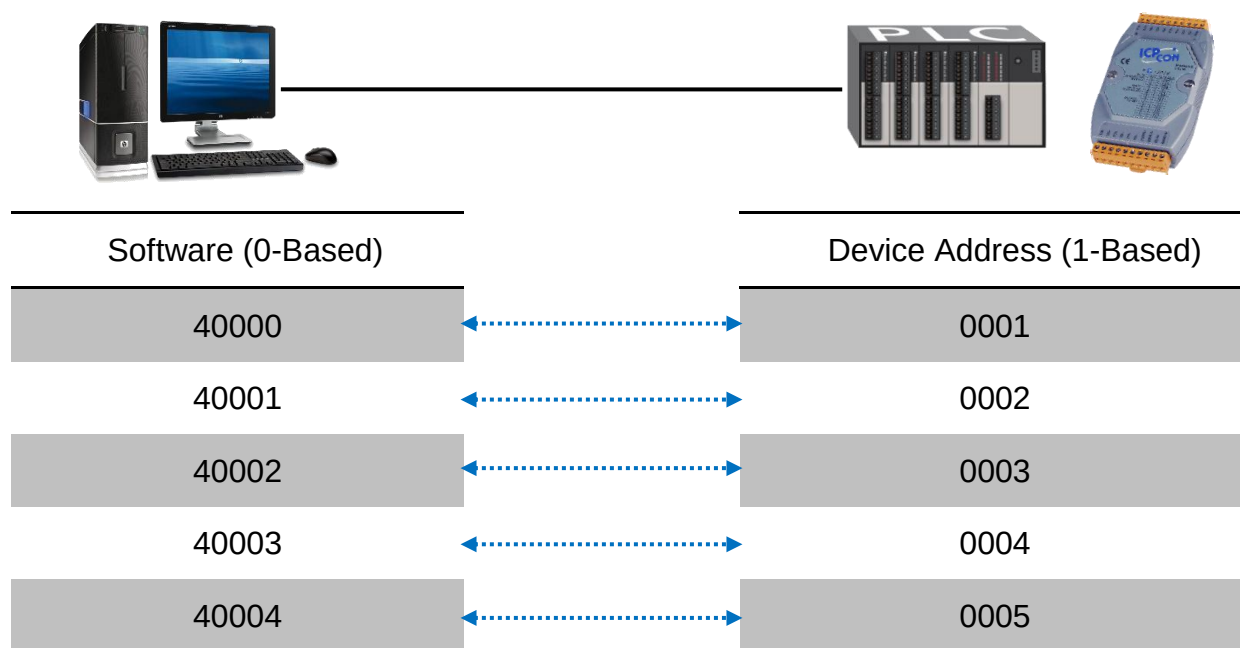
大部分的 SCADA 軟體的位址是從 1 開始 (1-Based)·如果硬體文件的位址 (如 PLC·M-7000) 也是從 1 開始，軟體讀寫資料的位址與硬體文件標記的位址相同。



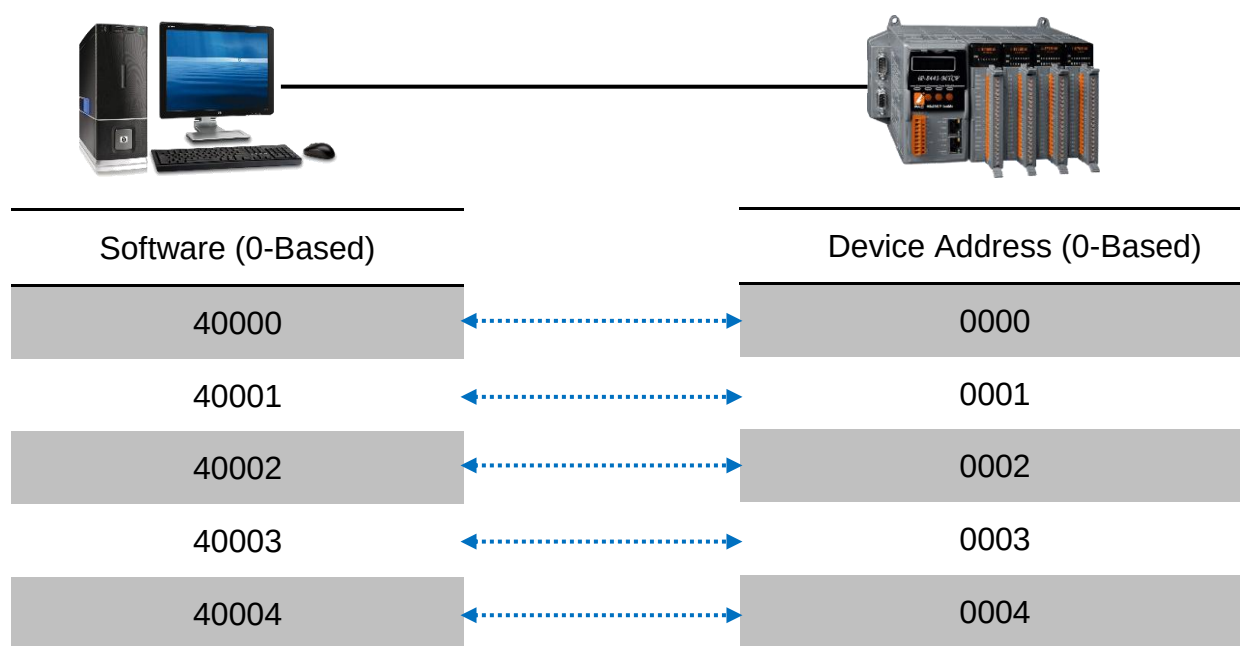
當軟體的位址從 1 開始 (1-Based·如 Win-GRAF)·而硬體文件的位址是從 0 開始 (0-Based，如 iP-8411-MRTU)，軟體讀寫資料的位址為 [硬體文件標記的位址+1]。



如果軟體的位址是從 0 開始 (0-Based, 如 LabVIEW) , 而硬體文件的位址 (如 PLC · M-7000) 是從 1 開始 , 軟體讀寫資料的位址為 [硬體文件標記的位址-1] 。



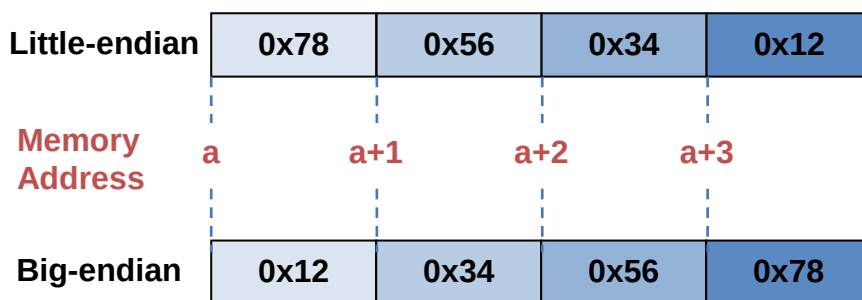
當軟體的位址從 0 開始 (0-Based, 如 Modbus Utility) , 而硬體文件的位址是從 0 開始 (0-Based , 如 iP-8411-MRTU) , 軟體讀寫資料的位址與硬體文件標記的位址相同 。



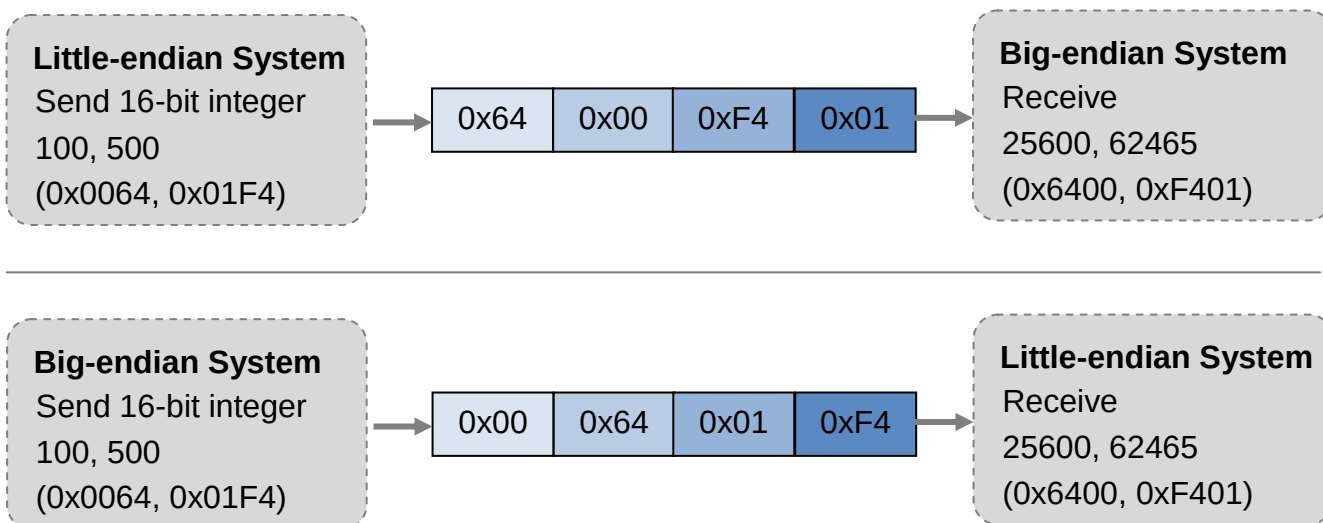
1.2. 傳輸資料格式的差異

在不同機器或是網路之間交換資料時，需要注意不同的系統可能會採用不同的放置規則。現今最常見的放置規則有兩種，分別是 **Big-endian** 與 **Little-endian**。**Big-Endian** 是指資料放進記憶體中的時候，最高位的位元組會放在最低的記憶體位址，而 **Little-Endian** 則是剛好相反，它會把最高位的位元組放在最高的記憶體位址。

以一個 32 位元 (bits) 的整數資料為例，0x12345678 在 **Big-endian** 和 **Little-endian** 的系統中，會按照下圖中的規則放置資料在記憶體中的位置：



大部份的系統都是 **Little-endian**，很少發生位元組順序導致資料被錯誤解讀。但如果已確定資料錯誤不是因為 **Modbus Master** 與 **Slave** 暫存器位址基準不同所造成，就可以考慮系統間存在位元組順序不同的狀況。下圖以實例說明收送系統因為位元組順序不同造成的錯判現象：



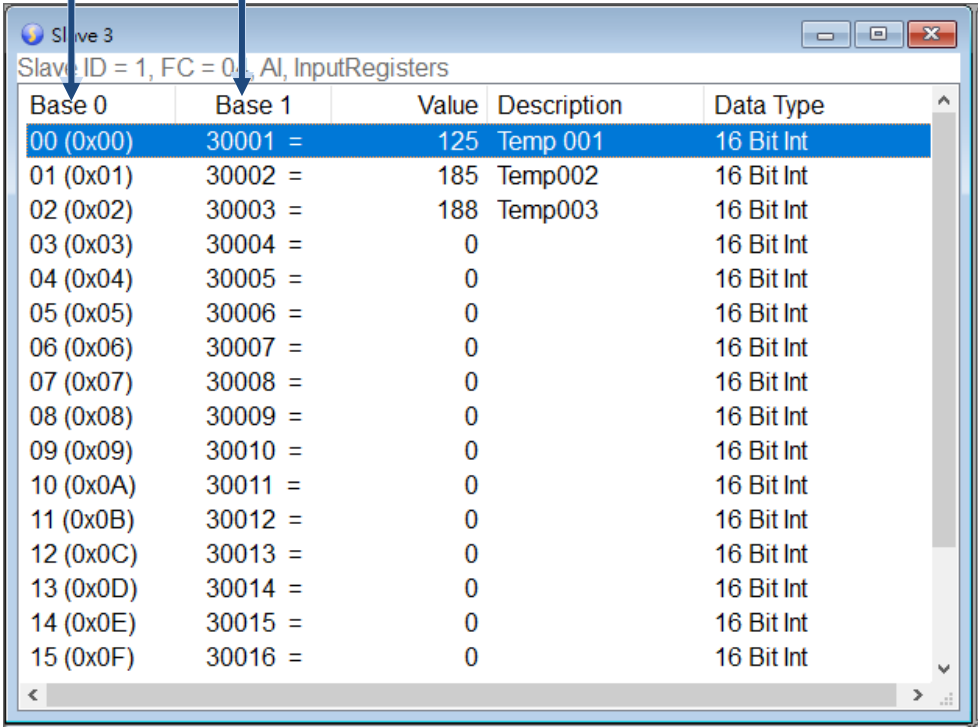
2. Modbus Slave Tool

為了能方便的地確認 Modbus Master 軟體解讀資料的方式，泓格科技設計了 Modbus Slave Tool，能夠模擬 Modbus Slave 設備與使用者的 Master 軟體進行通訊。透過提供已知的 Slave 的資料了解 Master 軟體解讀資料的格式。

Modbus Slave Tool 針對四種 Modbus 協定的資料類型，開啟四個視窗。提供每筆資料以 0-based 與 1-based 表示的位址，固定以 Little-endian 的格式傳輸資料。

0-based 位址

1-based 位址



Base 0	Base 1	Value	Description	Data Type
00 (0x00)	30001 =	125	Temp 001	16 Bit Int
01 (0x01)	30002 =	185	Temp002	16 Bit Int
02 (0x02)	30003 =	188	Temp003	16 Bit Int
03 (0x03)	30004 =	0		16 Bit Int
04 (0x04)	30005 =	0		16 Bit Int
05 (0x05)	30006 =	0		16 Bit Int
06 (0x06)	30007 =	0		16 Bit Int
07 (0x07)	30008 =	0		16 Bit Int
08 (0x08)	30009 =	0		16 Bit Int
09 (0x09)	30010 =	0		16 Bit Int
10 (0x0A)	30011 =	0		16 Bit Int
11 (0x0B)	30012 =	0		16 Bit Int
12 (0x0C)	30013 =	0		16 Bit Int
13 (0x0D)	30014 =	0		16 Bit Int
14 (0x0E)	30015 =	0		16 Bit Int
15 (0x0F)	30016 =	0		16 Bit Int

2.1. 前置作業

1. 下載 Modbus Slave Tool

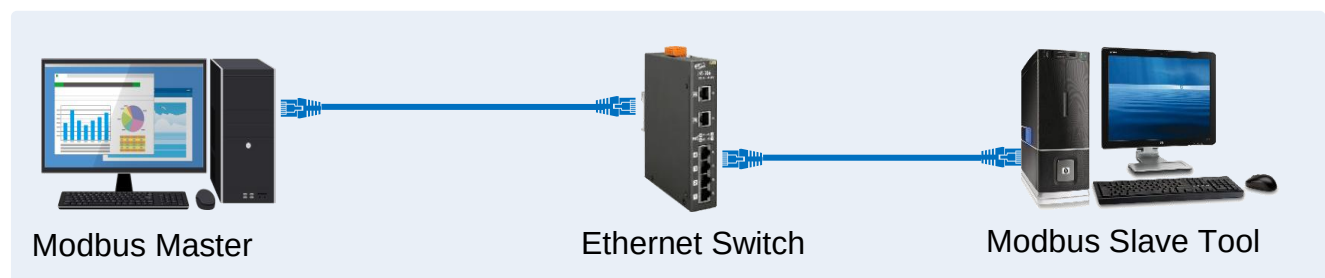
<https://www.icpdas.com/tw/download/index.php?nation=TW&kw=modbus+slave+tool>

2. 與 Modbus Master 連線

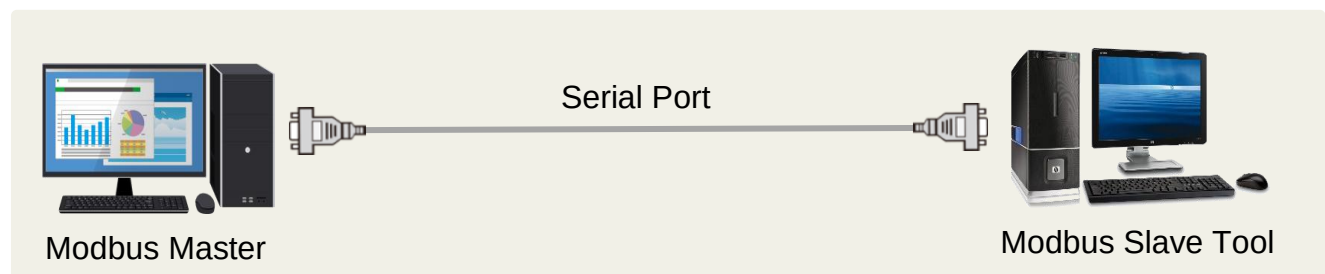
Modbus Slave Tool 可以模擬 TCP/IP 與 Serial Port 兩種介面的設備與 Master 進行通訊。

執行 Modbus Slave Tool 之前，請先依據實際使用的環境，將執行 Modbus Slave Tool 的電腦與 Master 主機透過 Serial Port 連線或連接到同一個網路。

將執行 Modbus Slave Tool 的電腦與 Master 主機連接網路



將執行 Modbus Slave Tool 的電腦與 Master 主機以 Serial Port 連線

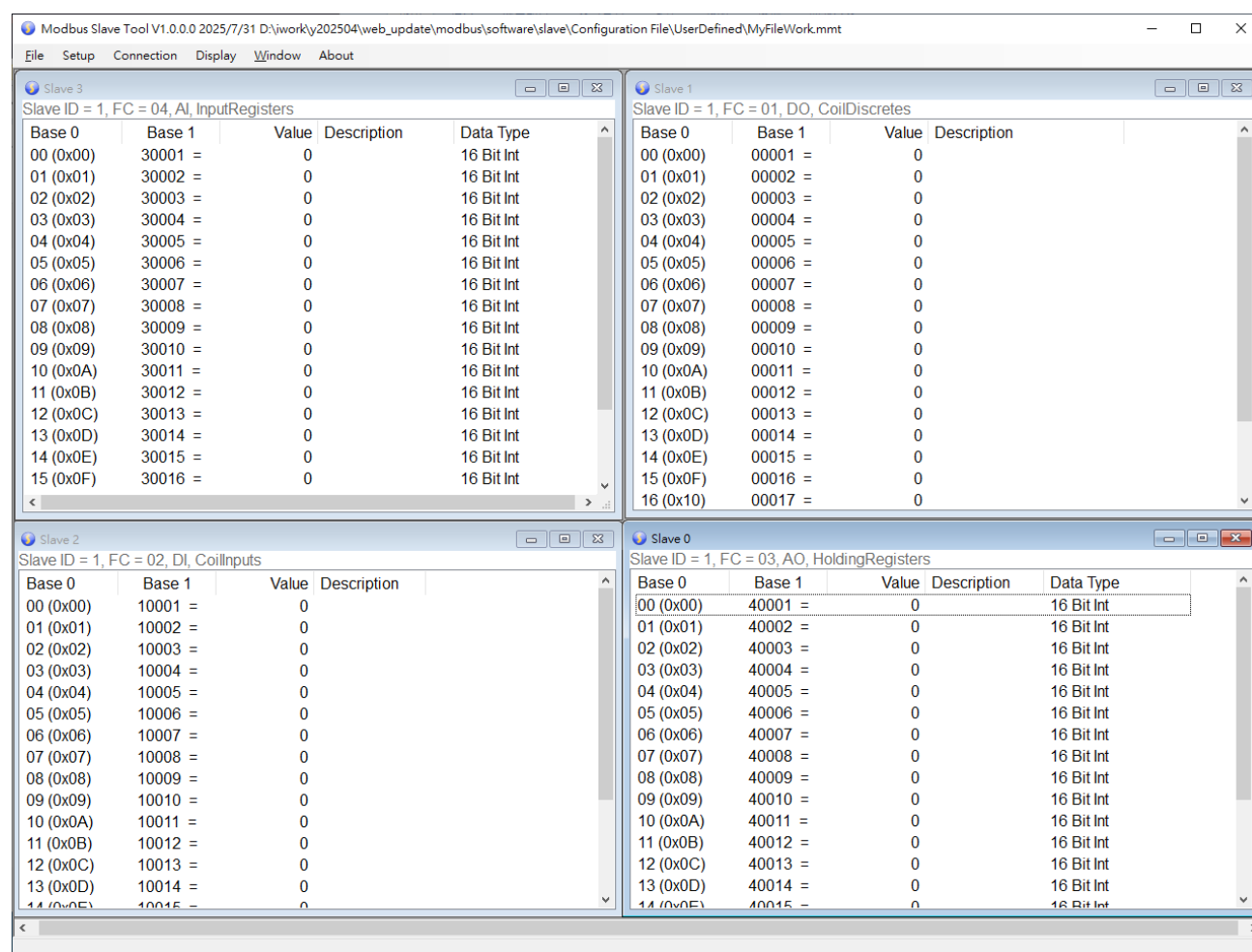


2.2. 設定 Modbus Slave Tool

Modbus 協定的資料類型

暫存器類型	PLC/SCADA	Modbus 位址	簡稱	讀寫	資料
Coils	00001 ~ 09999	0000H~FFFFH	0x	讀寫	DO
Discrete Inputs	10001 ~ 19999	0000H~FFFFH	1x	唯讀	DI
Input Registers	30001 ~ 39999	0000H~FFFFH	3x	唯讀	AI
Holding Registers	40001 ~ 49999	0000H~FFFFH	4x	讀寫	AO

當 ModbusSlaveToolPC.exe 開啟時，依照 Modbus 協定的資料類型，開啟四個視窗，分別對應 Modbus 功能代碼 01, 02, 03, 04 的資料。

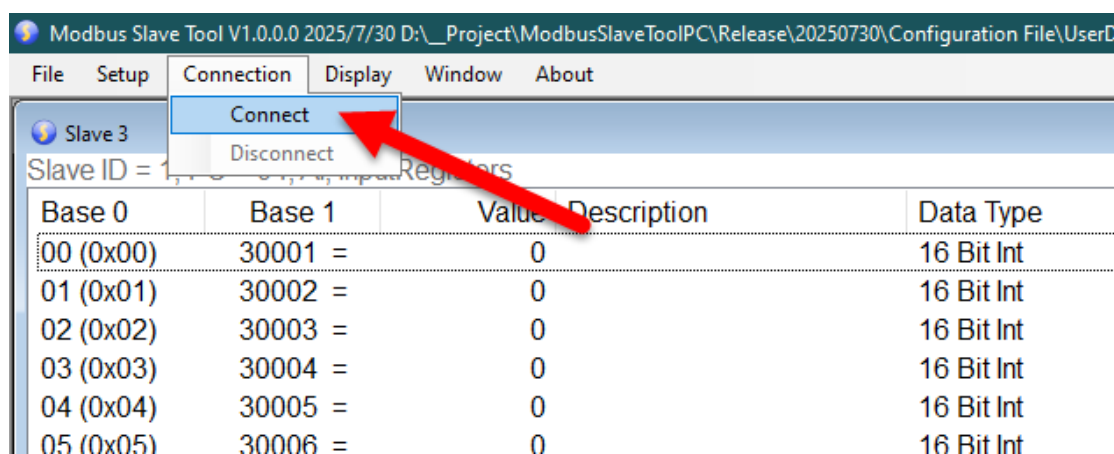


Modbus 通訊根據功能代碼 (Function Code) 存取不同類型的資料。

Function Code	Description	Address
01 (0x01)	Read the Status of the Coils (Readback DOs)	0xxxx
02 (0x02)	Read the Status of the Input (Reads DIs)	1xxxx
03 (0x03)	Read the Holding Registers (Readback AOs)	4xxxx
04 (0x04)	Read the Input Registers (Reads AIs)	3xxxx
05 (0x05)	Force a Single Coil (Writes DO)	0xxxx
06 (0x06)	Preset a Single Register (Writes AO)	4xxxx
15 (0x0F)	Force Multiple Coils (Writes DOs)	0xxxx
16 (0x10)	Preset Multiple Registers (Writes AOs)	4xxxx

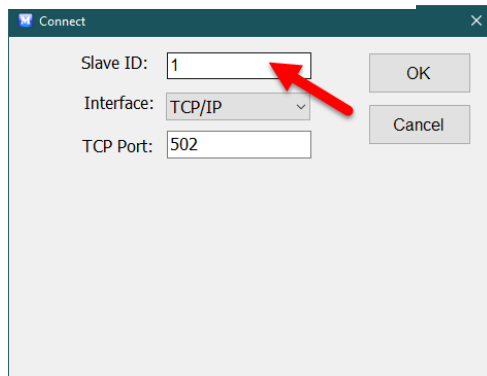
設定連線

1. 點擊工具列的 Connection > Connect。

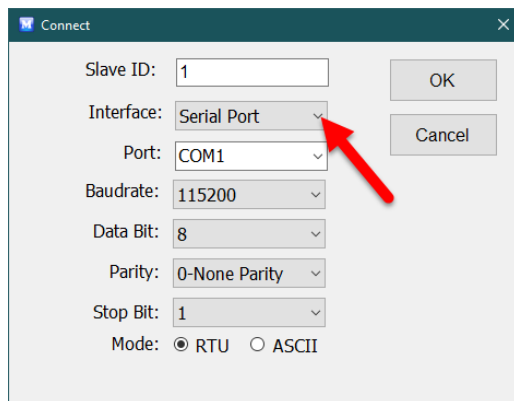


2. 選擇與 Master 連線的介面。

■ 選擇 TCP/IP，輸入站號



■ 選擇 Serial Port，設定 COM Port 號碼與站號

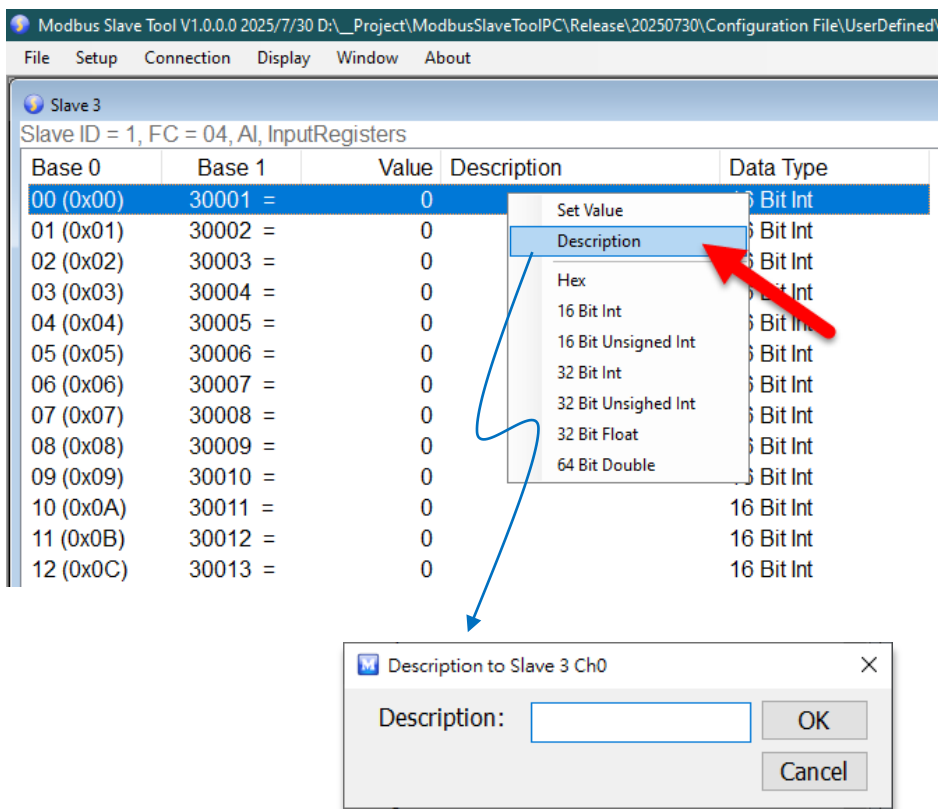


3. 建立連線後，ModbusSlaveTool 約每秒更新一次資料。

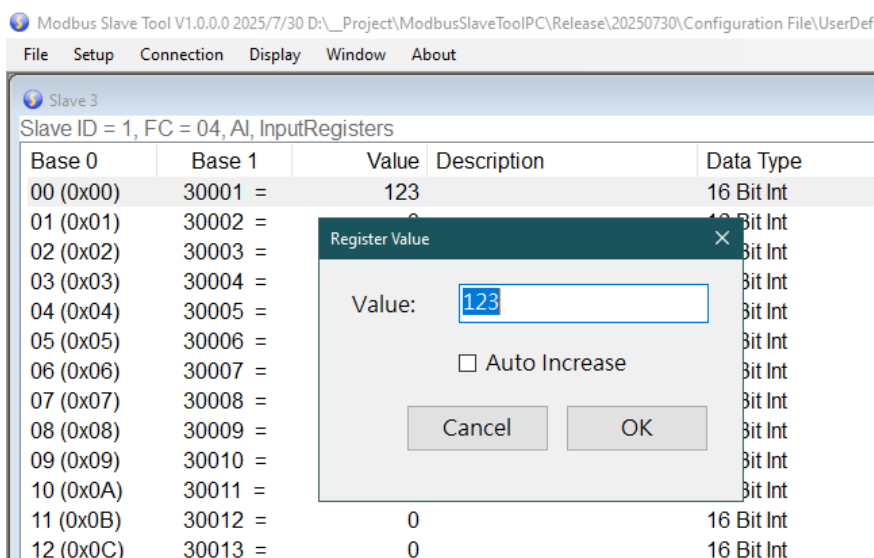
設定資料

1. 開啟每筆資料的右鍵選單，可選擇要建立資料說明或設定資料。

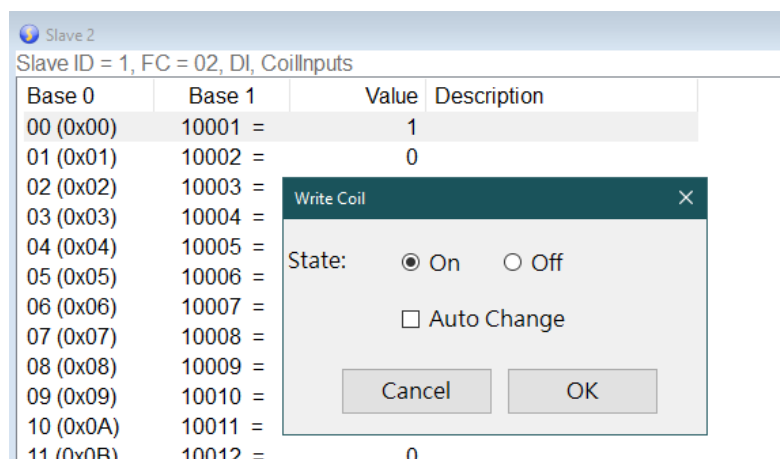
■ 設定說明



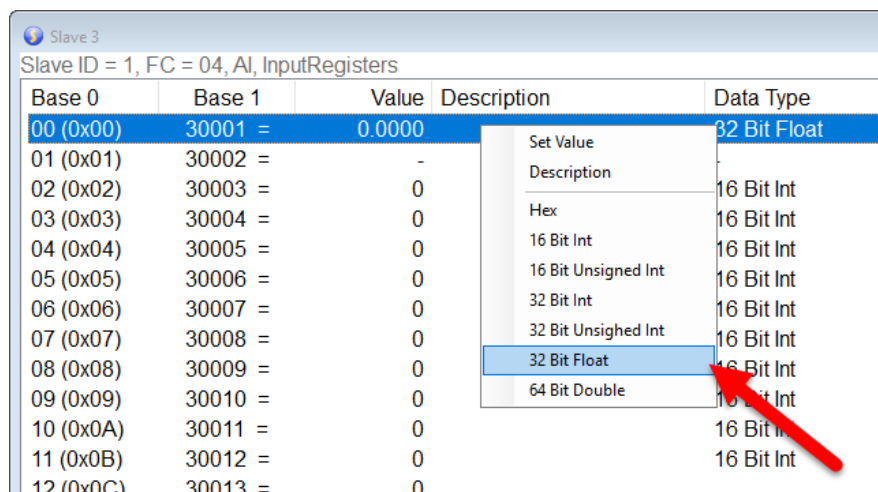
■ 設定 input register 或 holding register 資料



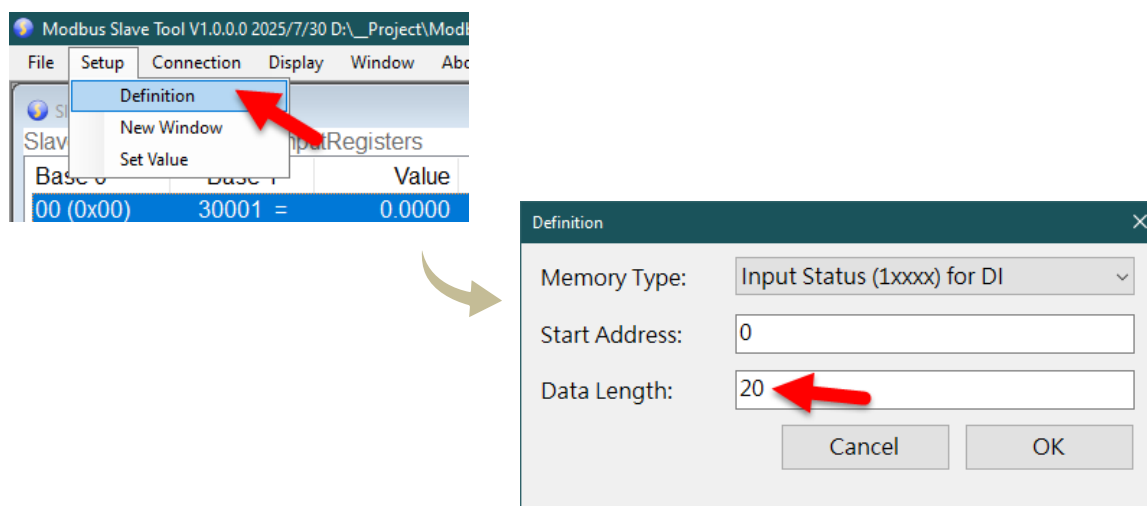
■ 設定 coil 或 discrete input 資料



■ 設定資料顯示格式



■ 設定視窗中要顯示的資料筆數



儲存專案檔

點擊工具列的 File > Save，將設定儲存成專案檔。專案檔存放在 Modbus Slave Tool 的 Configuration File > UserDefined 檔案夾中。之後需要測試時可以再次開啟使用。

