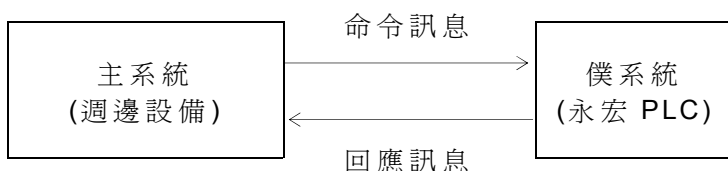


附 錄 二：FATEK 通訊協定

本通訊協定 (Protocol) 是永宏 PLC 主機上各通訊埠在標準通訊模式下均適用之通訊協定，任何對 PLC 之資料存取 (自 PLC 內部讀出或由外界寫入 PLC) 或運作、控制等，除在硬體連線及通訊參數設定必需通訊雙方一致外，在通訊訊息格式 (Message format) 方面亦必需符合本通訊協定之格式 PLC 才能正確回應。在介紹通訊協定之前首先需了解永宏 PLC 和與其通訊之週邊設備間之角色與互動關係。

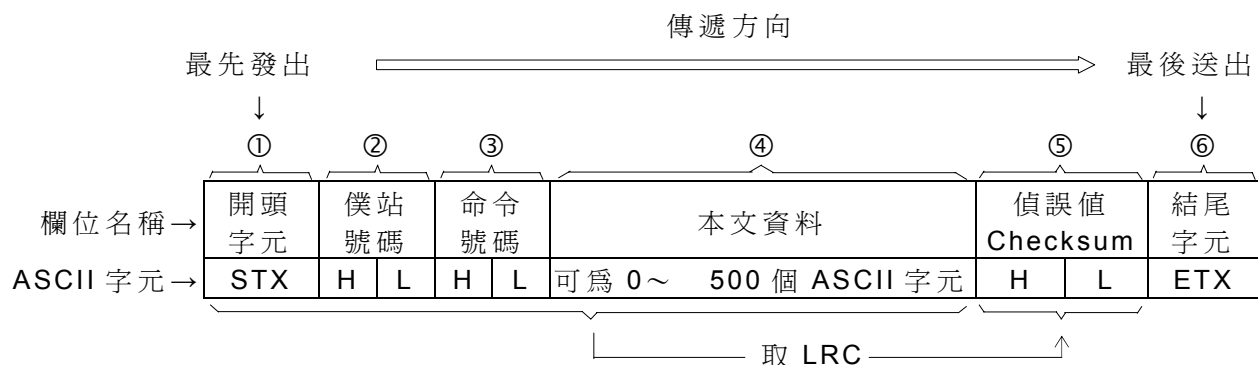
1.1 主僕定位與通訊互動關係

在永宏 PLC 之通訊架構上，永宏 PLC 是被定位為僕系統 (SLAVE) 而任何與永宏 PLC 連線之週邊設備均為主系統 (MASTER)，也就是說任何週邊設備與永宏 PLC 間之通訊均是由主系統 (週邊設備) 來主動發出命令，僕系統 (永宏 PLC) 只有在收到命令訊息後才依該命令之要求回應訊息給主系統，而不能主動發出訊息給主系統，如下之關係圖所示：



1.2 永宏 PLC 通訊訊息格式

永宏 PLC 之通訊訊息格式無論是命令訊息 (主系統發出) 或回應訊息 (僕系統發出) 均可概分為 6 個資料欄位，如下圖之範例：



①開頭字元 (STX)：ASCII 碼之開始字元 STX 之 16 進制碼數為 02H，無論命令或回應訊息之開頭字元均為 STX，接收方以此判知傳輸資料之開頭。

②僕站號碼：為兩位數之 16 進制數值，在永宏 PLC 通訊系統中之網路架構採用主僕系統在整個網路系統中只有一個主系統，但可以有 254 個僕系統，每個僕系統均有一個獨一無二之站號，分別為 1~FEH (站號 0 則當作對所有僕系統作廣播下命令)，當主系統欲對僕系統下命令時是以站號來指定由那個 PLC，或所有 PLC (廣播時) 來接收這個命令。在回應訊息中，僕系統會將自己的站號回應給主系統，以供主系統確認是它所指定的那個僕站 (PLC) 所送回之訊息。

註：PLC 之站號在出廠時均設為 1（第 1 站），站號之更改設定必須透過 FP-07C 或 WinProladder 來執行。

- ③ 命令號碼：為兩位數之 16 進制數值，所謂命令號碼係由主系統要求僕系統所執行之動作類別，例如要求讀取或寫入單點狀態、填入或讀取暫存器資料、強制設定、運轉、停止.....等，如同站號一般，在回應訊息中，僕系統亦會將其自主系統接收之命令號碼原原本本地隨同本文資料一併傳回主系統。
- ④ 本文資料：本文資料可為 0（無本文資料）～500 個 ASCII 字元，在命令訊息中此欄位資料用以指定命令所要運作或存取之對象（位址）或要寫入之數值。在回應訊息中本欄位之開頭為一個錯誤碼字元，在正常（沒有錯誤）情況下此錯誤碼必為字元 0（30H），其後跟著才是要回應給主系統之狀態或數值等本文資料。當有錯誤時，本開頭字元不再是 0，而代之以錯誤碼，同時其後不再有其他本文資料（即本文資料僅為一個字元之錯誤碼），請參閱第 3 節之說明。
- ⑤ 偵誤值(CHECKSUM)：偵誤值係將前述①～④各欄位之所有 ASCII 字元之 16 進制數值以“縱式餘數查核法” LRC（Longitudinal Redundancy Check）計算產出一個 Byte 長度（兩個 16 進制數值 00～FF）之偵誤值。當接收端收到訊息後依同樣之計算法則將①～④欄位之所有字元算出其偵誤值，兩者值相同表傳輸之資料正確，否則即有傳輸錯誤發生。本通訊協定 LRC 偵誤值之計算法是將各 ASCII 字元之 16 進制數碼值（8 位元長度）從頭至尾依序相加，但不考慮進位，因此最終結果仍為 8 位元長度之偵誤值。
- ⑥ 結尾字元（ETX）：ASCII 碼之結尾字元 ETX 之 16 進制數碼為 03H，無論命令或回應訊息之結尾字元均為 ETX，當接收方收到 ETX 字元後便知該次通訊已結束，可開始處理該命令或資料了。

1.3 永宏 PLC 之通訊錯誤碼

在通訊過程中無論是主系統命令、位址、數值範圍等之軟體或操作上的錯誤以及硬體上之問題均可能造成僕系統無法處理主系統所下達之命令，此時僕系統仍會回應訊息給主系統（但前提是僕系統收到的必須是正確的通訊，亦即無 Checksum 錯誤之命令，否則不會有任何回應），無論主系統下的命令碼或本文資料為何，在錯誤發生時僕系統所回應之訊息格式都是一樣的，除必有之開頭字元 STX 和結尾字元 ETX 及偵誤值外，並將其所收到之站號和命令碼原原本本地回應給主系統，同時僕系統會判斷該錯誤是屬何種錯誤而將該錯誤代碼回應給主系統，下圖為永宏 PLC 之通訊錯誤之回應訊息格式。

● 下表為永宏 PLC 通訊錯誤碼及其說明：

錯誤碼	說明
0	通訊正常（沒有錯誤情形發生）
2	不合法數值（如 10 進制格式中有 16 進制數字）
4	不合法之命令格式（含不合法之命令碼），或通訊命令無法執行
5	不能啟動（下 RUN 命令但 Ladder Checksum 不合）
6	不能啟動（下 RUN 命令但 PLC ID≠Ladder ID）
7	不能啟動（下 RUN 命令但程式語法錯誤）
9	不能啟動（下 RUN 命令，但 Ladder 之程式指令 PLC 無法執行）
A	不合法之位址

1.4 通訊命令功能詳述

本節針對永宏 PLC 通訊協定中所提供之通訊命令碼逐一舉例說明主系統之命令訊息及僕系統之回應訊息格式（只舉通訊成功之例子，若有通訊錯誤情況請參考第 1.3 節之說明）。

1.4.1 元件類別及其指定方法

通訊功能中最主要的功能在於讀取或寫入 PLC 內部元件（單點或暫存器）之狀態或數值資料。因為首先您必須明瞭 PLC 內部到底有哪些元件可供您運用，以及如何來指定它們。下表為永宏 PLC 可供您存取之單點與暫存器及其位址指定方法（關於元件之特性大小範圍及意義請參考基礎功能篇手冊第 2.1 節之說明）。

元件類別	代號	名稱	單點位址指定 （5 個字元）	16 位元暫存器指定 （6 個字元）	32 位元暫存器指定 （7 個字元）
單點狀態	X	輸入接點	X0000～X9999	WX0000～WX9984	DWX0000～DWX9968
	Y	輸出繼電器	Y0000～Y9999	WY0000～WY9984	DWY0000～DWY9968
	M	內部繼電器	M0000～M9999	WM0000～WM9984	DWM0000～DWM9968
	S	步進繼電器	S0000～S9999	WS0000～WS9984	DWS0000～DWS9968
	T	計時器接點	T0000～T9999	WT0000～WT9984	DWT0000～DWT9968
	C	計數器接點	C0000～C9999	WC0000～WC9984	DWC0000～DWC9968
暫存器資料	TMR	計時器暫存器	-	RT0000～RT9999	DRT0000～DRT9998
	CTR	計數器暫存器	-	RC0000～RC9999	DRC0000～DRC9998
	HR	資料暫存器	-	R00000～R65535	DR00000～DR65534
	DR	資料暫存器	-	D00000～D65535	DD00000～DD65534
	FR	檔案暫存器	-	F00000～F65535	DF00000～DF65534

- 單點狀態（X，Y，M，S）可以將連續 16 個或 32 個狀態組成 16 位元或 32 位元暫存器來用，如上表之 WX△△△△或 DWX△△△△，但△△△△必須為 8 的倍數。
- 由上表可知單點位址指定需 5 個位元，16 位元暫存器位址指定均為 6 個字元，而 32 位元暫存器之位址指定必為 7 個字元。
- 上表之元件位址範圍（Boundary）為永宏 PLC 元件之最大範圍，因各系列 PLC 機種之元件多寡不一，範圍亦不盡相同，使用者需自行注意其所使用機型 PLC 之元件大小範圍（例如 PLC 之 X、Y 位址範圍為 0000～0255，S 位址為 0000～0999），又如若通訊命令位元之元件位址指定超出該 PLC 之範圍，PLC 將回應 error A（不合法位址），同時不執行該命令。

1.4.2 通訊命令說明

- 下表為永宏 PLC 通訊協定所提供之通訊命令一覽表：

命令號碼	功能敘述	一次通訊可處理之訊息長度	備註
40	PLC 概略系統狀態讀取	-	
41	PLC 之 RUN/STOP 控制	-	
42	單一個單點之運作控制	1 點	
43	連續多個單點之抑/致能狀態讀取	1～256 點	
44	連續多個單點之狀態讀取	1～256 點	
45	連續多個單點之狀態寫入	1～256 點	
46	連續多個暫存器之資料讀取	1～64Words	
47	連續多個暫存器之資料寫入	1～64Words	
48	任意單點/暫存器混合之狀態/資料讀取	1～64 點或 Words	
49	任意單點/暫存器混合之狀態/資料寫入	1～32 點或 Words	
4E	測試回傳	0～256 字元	
53	PLC 詳細系統狀態讀取	-	

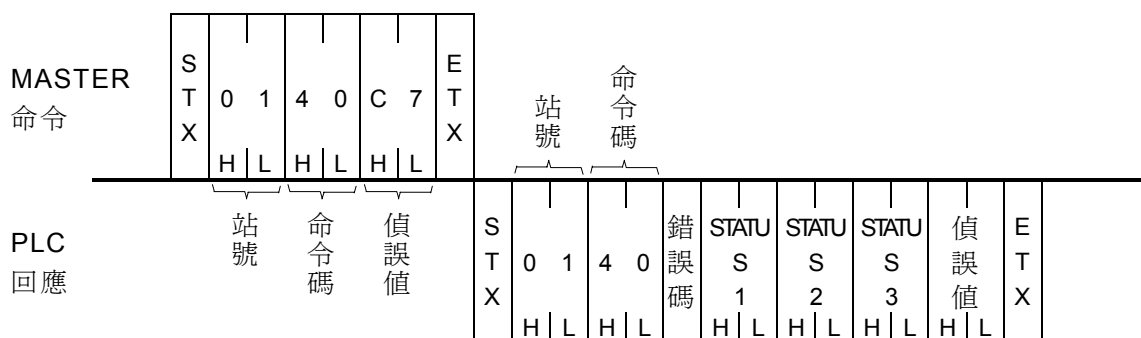
註 1：在本文訊息中單點狀態之表示是以一個字元來表示（1 表示 ON，0 表 OFF），而 16 位元暫存器資料則以 4 個字元來表示一個 Word 的數值（0000H～FFFFH）。

註 2：在 32 位元暫存器時，其資料為 DW（連續兩個 Word），故須以 8 個字元來表示，因此若元件為 32 位元暫存器，則一個元件需以 2W 計，例如命令碼 46 和 47 在 16 位元元件一次最多可處理 64 個，但若為 32 位元元件則一次最多只能處理 32 個。

- 註 3：在任意單點/暫存器混合運作之命令（48，49）中，其訊息長度為單點和 Word 數之總和，兩者總數不得超過 64W（命令 48）及 32W（命令 49），亦即多一個點，所容許之 Word 數便少一個，反之亦同，而 32 位元元件因其訊息長度是一個元件佔用 2 個 Words，故多一個 32 位元元件，則所容許之單點或 Word 數就少 2 個，例如命令 48，其訊息長度可為 1~64W，假設其讀取 20 個 32 位元元件，則其訊息已佔用 40 個 Words，只剩 24W 可供單點或 16 位元暫存器使用，因此本例在一次通訊中命令碼最多只能讀取 44 個元件（20 個 32 位元元件，24 個單點或 16 位元元件）。
- 註 4：上表中之連續單點或連續暫存器運作（讀取或寫入）係指運作之對象元件不只一個，且為連續號碼，因此在指定這些運作對象時，無需一一列舉其元件號碼，只需指定其起始號碼及元件之個數 N 即可，但其運作對象只能為單點或暫存器之一，不能混合。
- 註 5：相對於連續多點之運作，任意（Random）多個運作對象，同樣可在一次通訊中讀取或寫入多個單點和暫存器，但因其號碼可為非連續故必須一一列舉其單點或暫存器號碼，也因此容許將單點和暫存器混合運作。
- 註 6：程式之存檔（Save）或載入（Load）運作，係將 PLC 內部整個程式區之程式取出存入磁碟檔案中，或將磁碟檔案中之程式整個載入到 PLC 去。因在一次通訊中儲存或載入動作最大可傳輸 64 words，故程式之存檔或載入均需執行多次通訊才能作完。

● 命令碼 40 (PLC 概略系統狀態讀取)

格式



STATUS 1 :

H				L			
B7	B6	B5	B4	B3	B2	B1	B0

STATUS 2 :
(LADDER
程式容量)

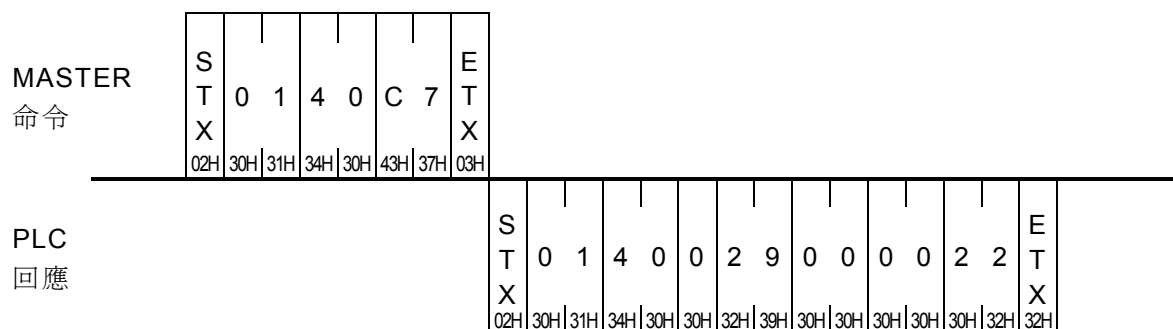
00H , 保留
FFH , FB 8K Step 程式
53H , FBE 8K Step 程式
54H , FBE 13K Step 程式
55H , FBN 8K Step 程式
56H , FBN 13K Step 程式
AAH , FM 8K Step 程式
ABH , FM 13K Step 程式

B0 : RUN/STOP
B1 : BAT LOW/正常
B2 : Ladder checksum
error/正常
B3 : 使用 ROM PACK/
未使用
B4 : WDT Timeout/正常
B5 : 設定 ID/未設 ID
B6 : 緊急停機/正常
B7 : 0 (保留供未來使用)

STATUS 3 : 為 0 (保留供未來使用)

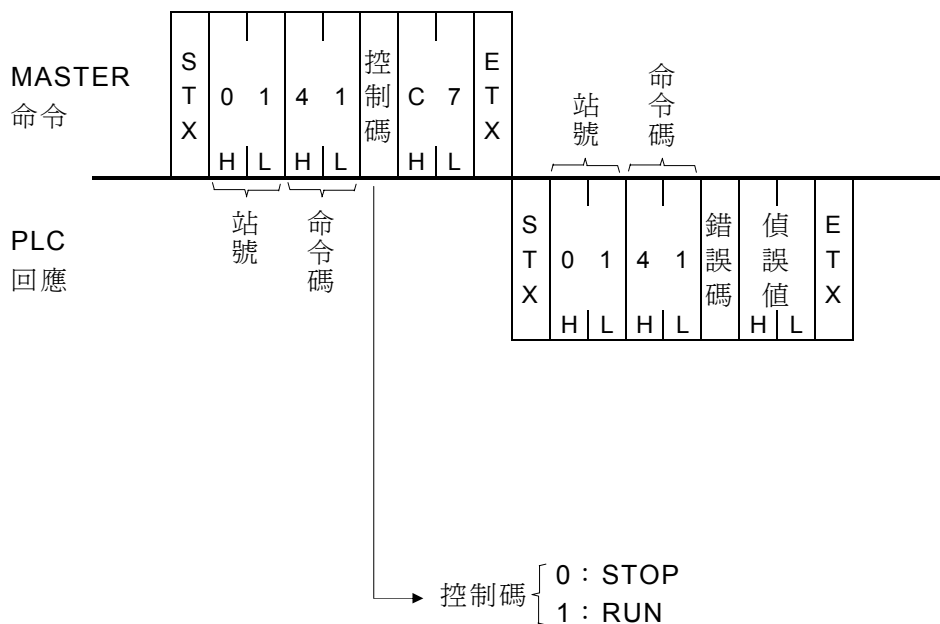
範例

假設 PLC 插有 ROM PACK，PLC 和 ROM PACK 均設 ID，其他狀態均正常，且 PLC 在 RUN 情況下，MASTER 以命令碼 40 去讀取 PLC 之系統狀態將得到如下結果 (B5，B3，B0 為 1，其餘為 0，故 STATUS 為 29H)。



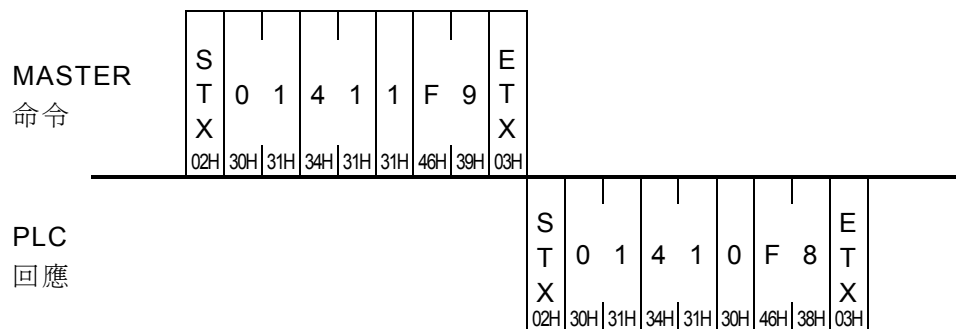
● 命令碼 41 (PLC 之 RUN/STOP 控制)

格式



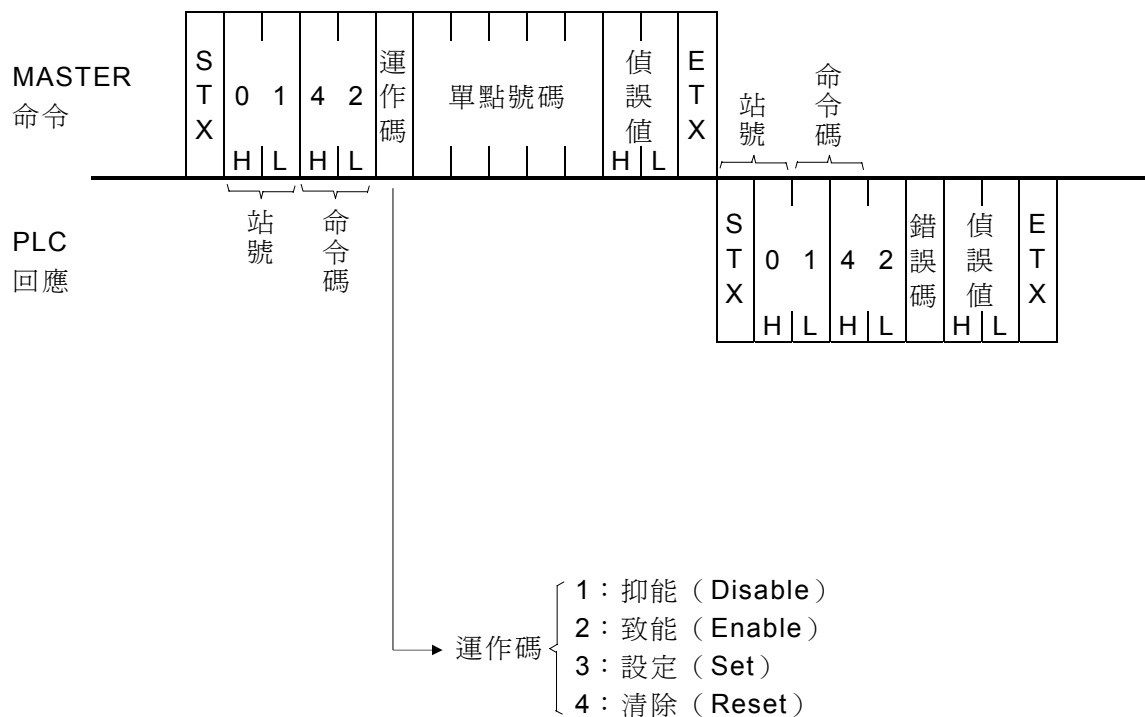
範例

將 PLC 啟動 (RUN)

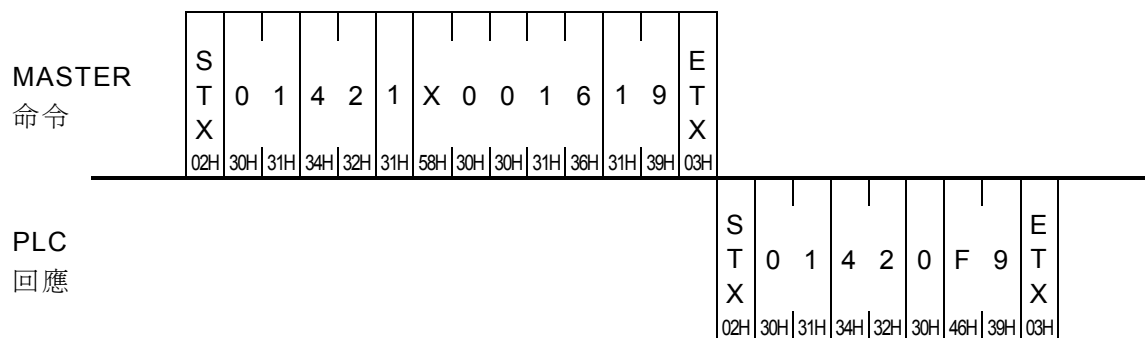


● 命令碼 42（單一個單點運作控制）

格式 本命令可對命令中所指定之單點作抑能、致能、設定、清除等四種運作。

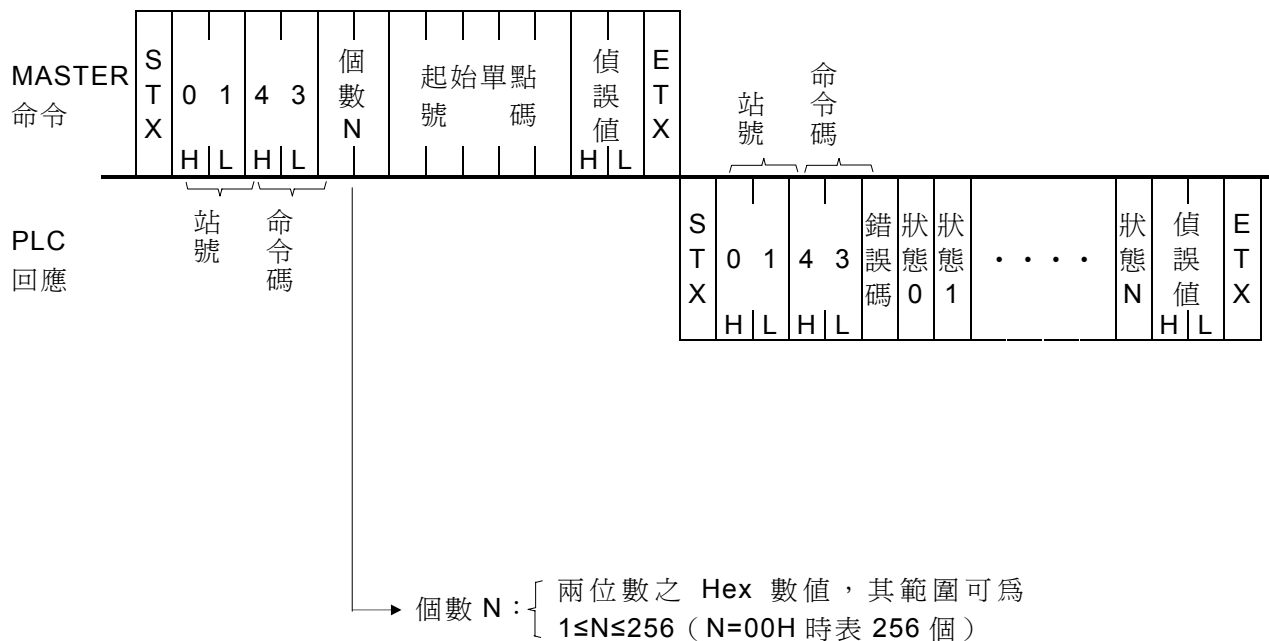


範例 下圖通訊格式為將單點 X1 6 抑能之範例。

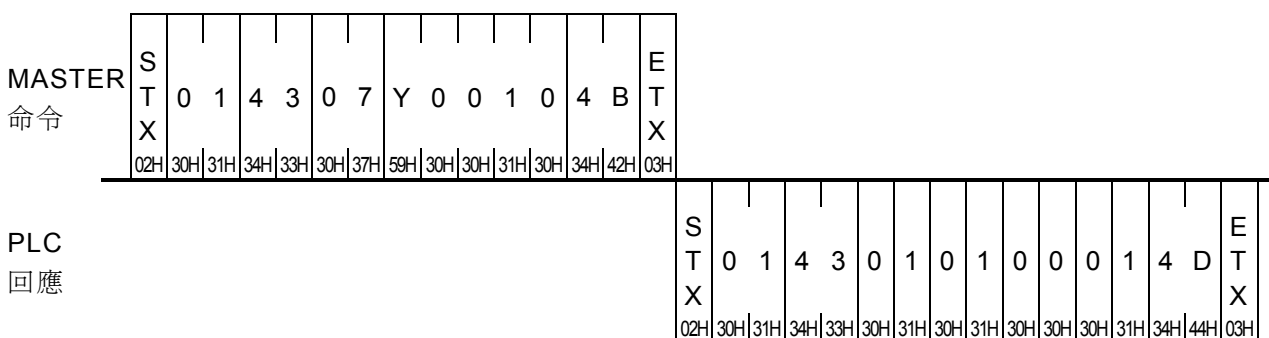


● 命令碼 43 (多個連續單點之抑/致能狀態讀取)

格式 用以讀取自命令中所指定之單點開始往高位之連續 N 個單點之抑/致能狀態。

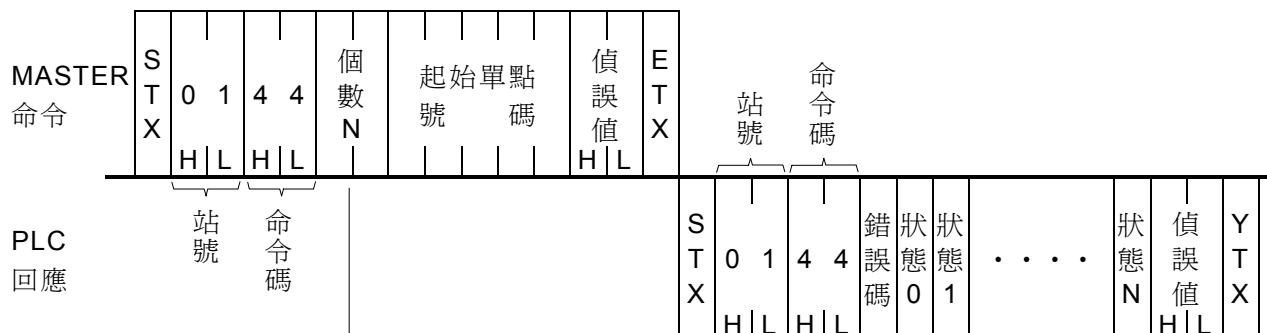


範例 設 Y10 開始至 Y16 之連續 7 個單點中，Y10，Y12，Y16 為抑能，其他均為致能情況，下圖為讀取 Y10~Y16 連續 7 個單點之抑/致能狀態之結果。



● 命令碼 44（多個連續單點狀態讀取）

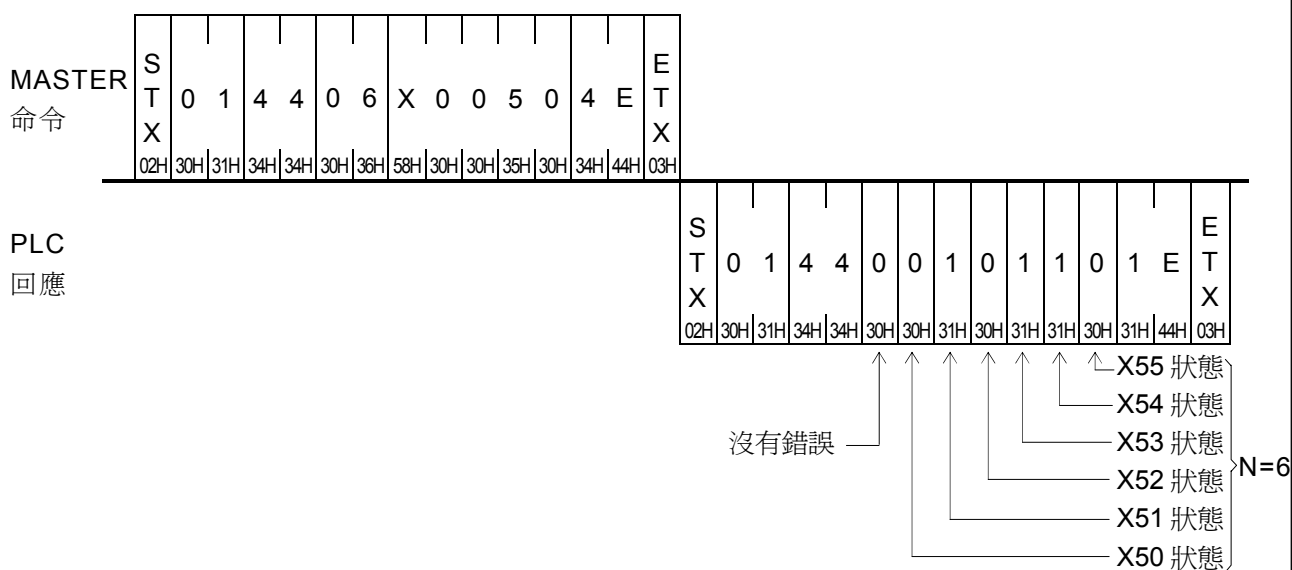
格式



→ 個數 N：{ 兩位數之 Hex 數值，其範圍可為
1≤N≤256（N=00H 時表 256 個）

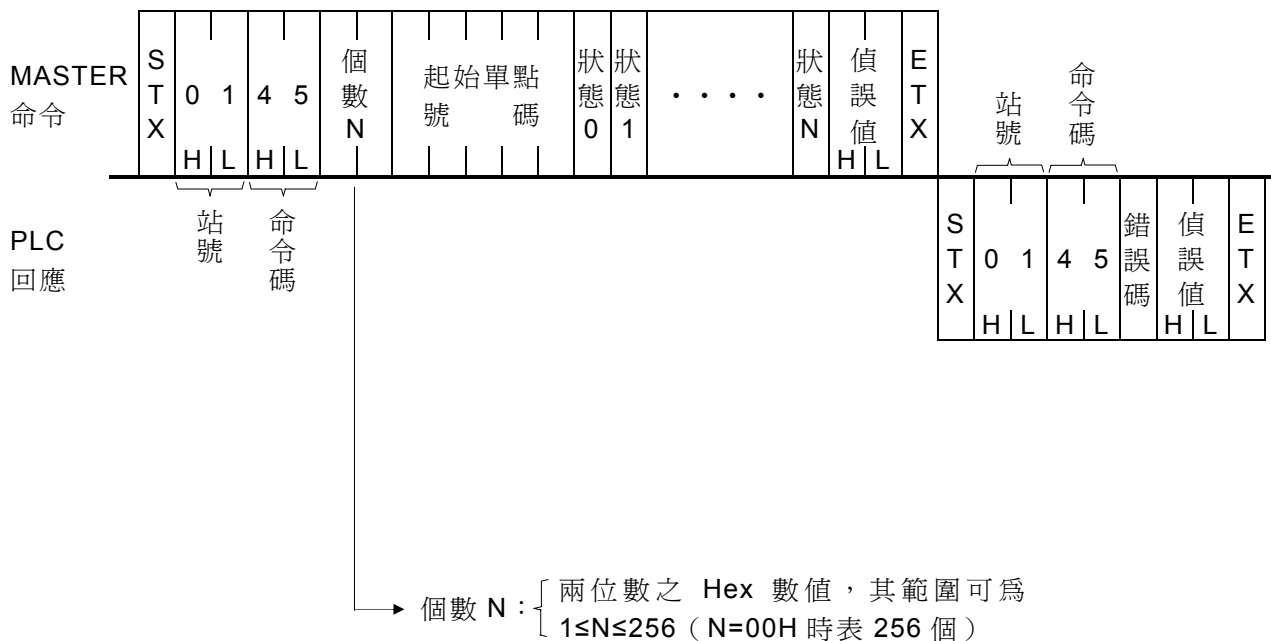
範例

讀取自 X50 開始連續 6 個輸入點（即 X50~X55）狀態，並假設 X50，X52 和 X55 之狀態為 0，而 X51，X53 和 X54 之狀態為 1，下圖為其通訊結果。



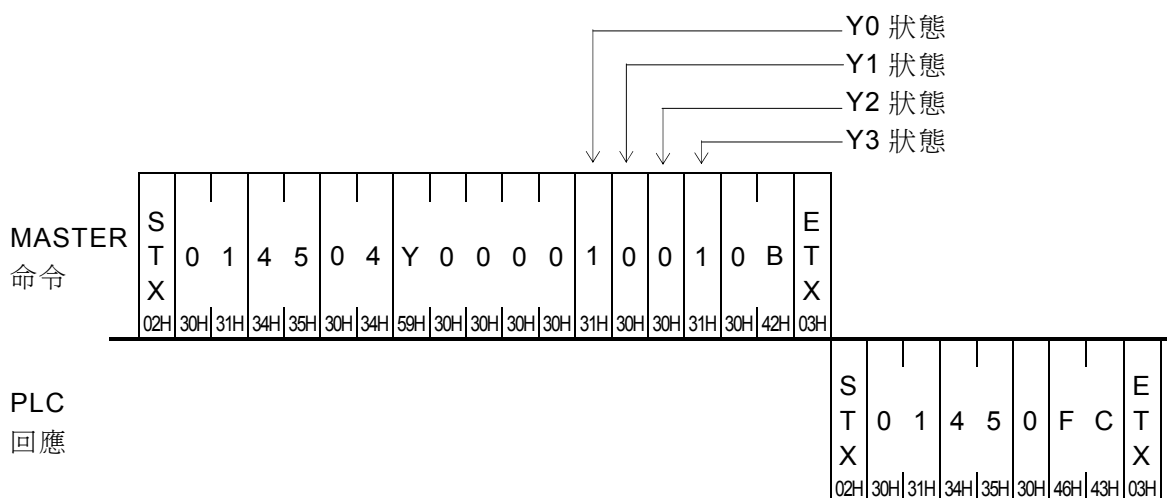
● 命令碼 45（多個連續單點狀態寫入）

格式



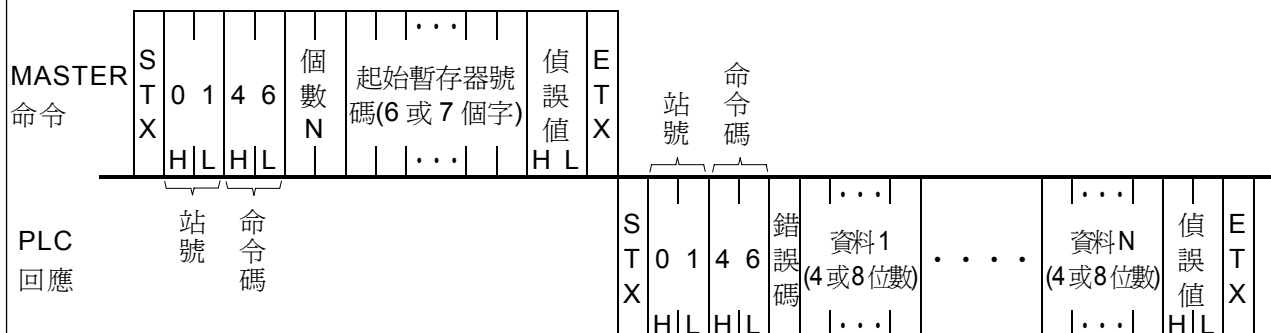
範例

自 Y0 開始連續寫入 4 個輸出點（Y0～Y3）的狀態，分別是 Y0 和 Y3 為 1，Y1 和 Y2 為 0。



● 命令碼 46（多個連續暫存器資料讀取）

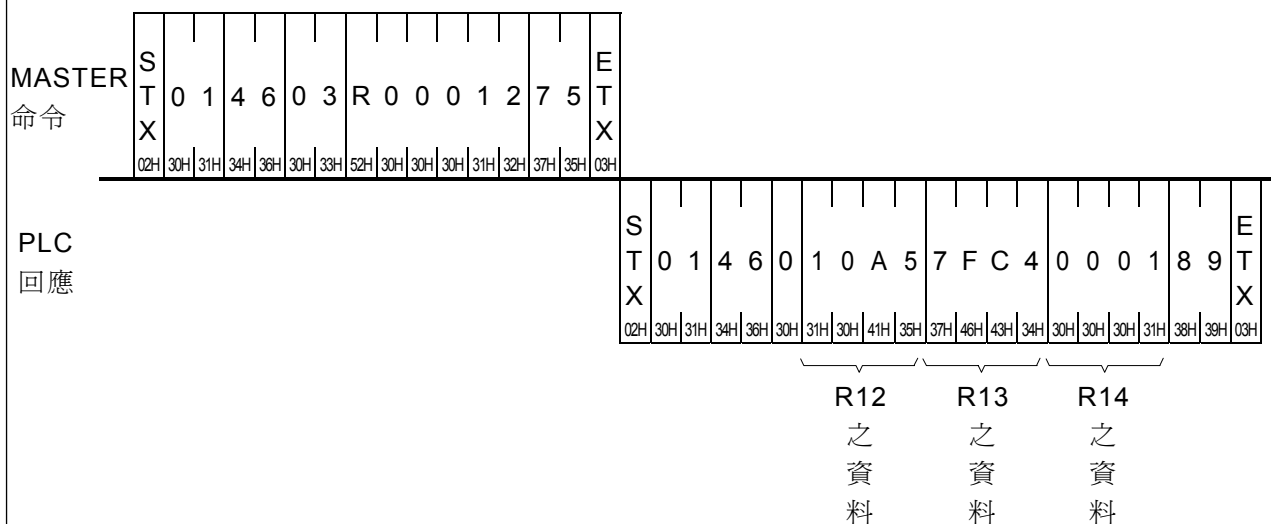
格式



- 個數 N 為兩位數之 Hex 數值，其範圍可為 01H~40H 或 20H（32 位元元件時）
- 16 位元暫存器號碼為 6 個字元，而其資料為 4 個字元之 Hex 數值（可表示 0000H~FFFFH）
- 32 位元暫存器號碼為 7 個字元，而其資料為 8 個字元之 Hex 數值（可表示 00000000H~FFFFFFFFH）

範例

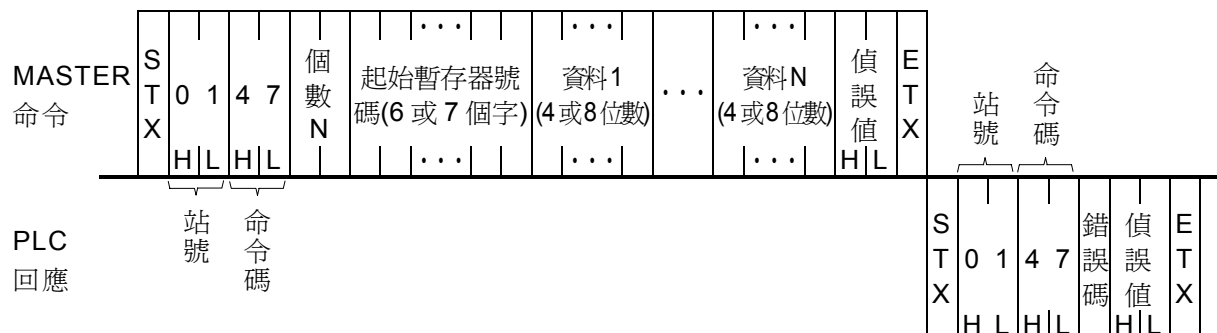
讀取 16 位元暫存器 R12 開始之連續 3 個 16 位元暫存器（即 R12，R13，R14）之資料



- 由上例 PLC 之回應可知 R12=10A5H，R13=7FC4H，R14=0001H

● 命令碼 47（多個連續暫存器資料寫入）

格式



- 個數 N 為兩位數之 Hex 數值，其範圍可為 01H~40H 或 20H（32 位元元件時）
- 16 位元暫存器號碼為 6 個字元，而其資料為 4 個字元（可表示 0000H~FFFFH）
- 32 位元暫存器號碼為 7 個字元，而其資料為 8 個字元（可表示 00000000H~FFFFFFFFH）

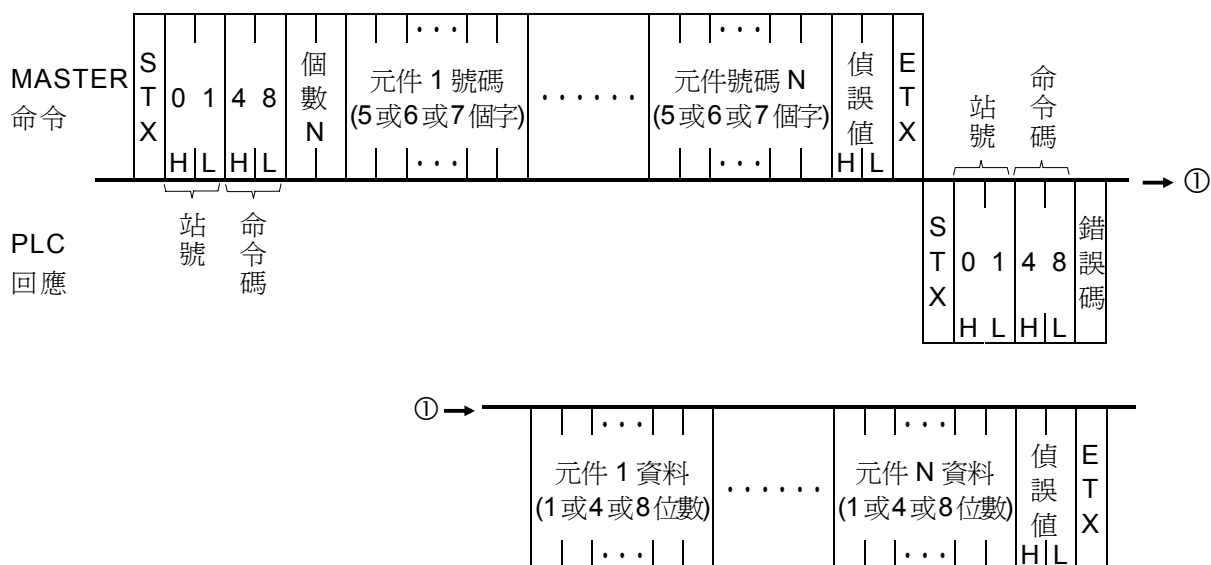
範例

將 16 位元暫存器 WY8 寫入 AAAAH，而 WY24 寫入 5555H。因 WY8 和 WY24 為連續（即自 WY8 起連續 2 個暫存器），故為多個連續暫存器寫入之格式。



● 命令碼 48 (多個任意單點狀態或暫存器資料混合讀取)

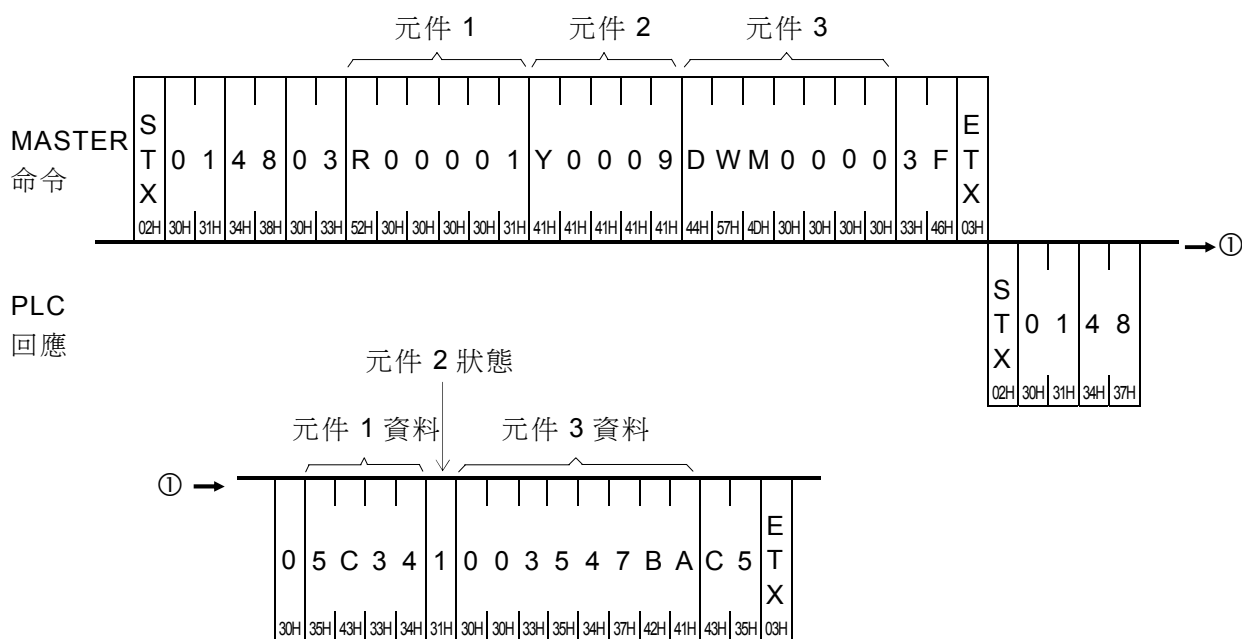
格式



- 個數 N 為兩位數之 Hex 數值，用以表示讀取元件之總數，其範圍可為 01H~40H（參閱註 3）
- 元件為單點時，其元件號碼為 5 個字元，而回應狀態則僅一個位數（1 或 0）
- 元件為 16 位元暫存器時，其元件號碼為 6 個字元，而回應資料為 4 個字元之 Hex 數值
- 元件為 32 位元暫存器時，其元件號碼為 7 個字元，而回應資料為 8 個字元之 Hex 數值

範例

讀取 R1，Y9 和 DWM0（即 M31~M0）三個元件之狀態或資料



- 由上 PLC 回應例可知 R1=5C34H，而 Y9 狀態為 1（“ON”），DWM0=3547BAH

● 命令碼 49 (多個任意單點狀態或暫存器資料混合寫入)

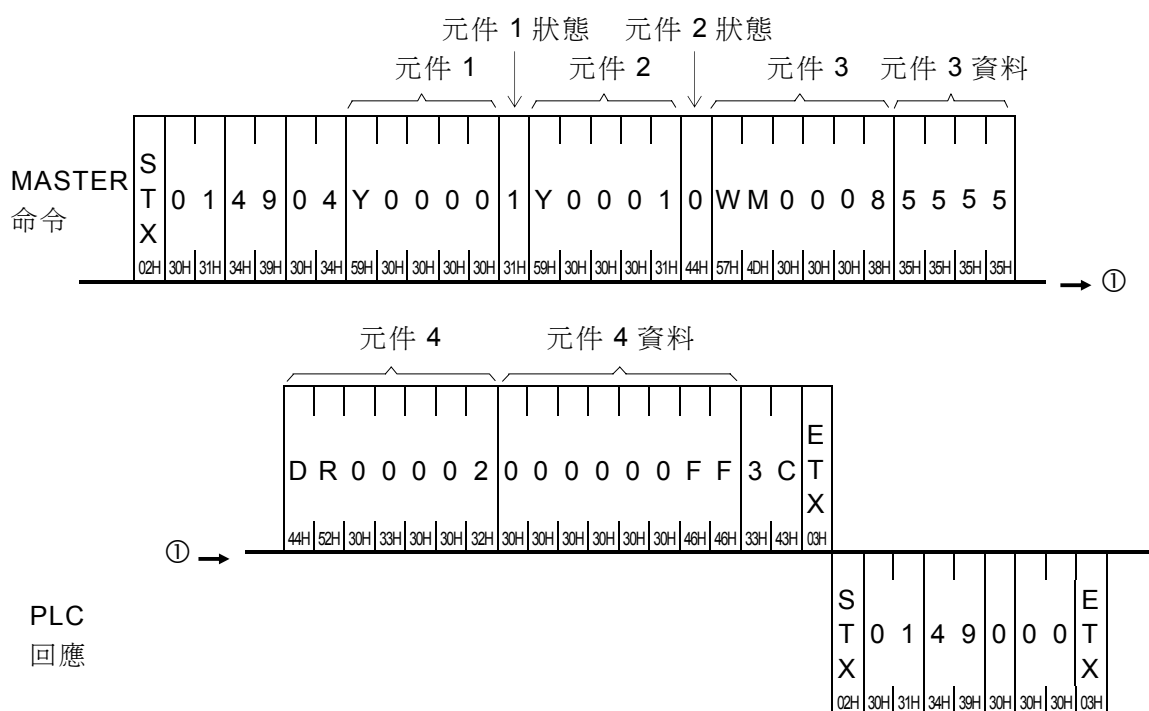
格式

PLC
回應

- 個數 N 為兩位數之 Hex 數值，用以表示寫入元件之總數，其範圍可為 01H~40H（參閱註 3）
- 元件為單點時，其元件號碼為 5 個字元，而其狀態則只佔一個位數（0 或 1）
- 元件為 16 位元暫存器時，其元件號碼為 6 個字元，而其資料為 4 位數之 Hex 數值
- 元件為 32 位元暫存器時，其元件號碼為 7 個字元，而其資料為 8 位數之 Hex 數值

範例

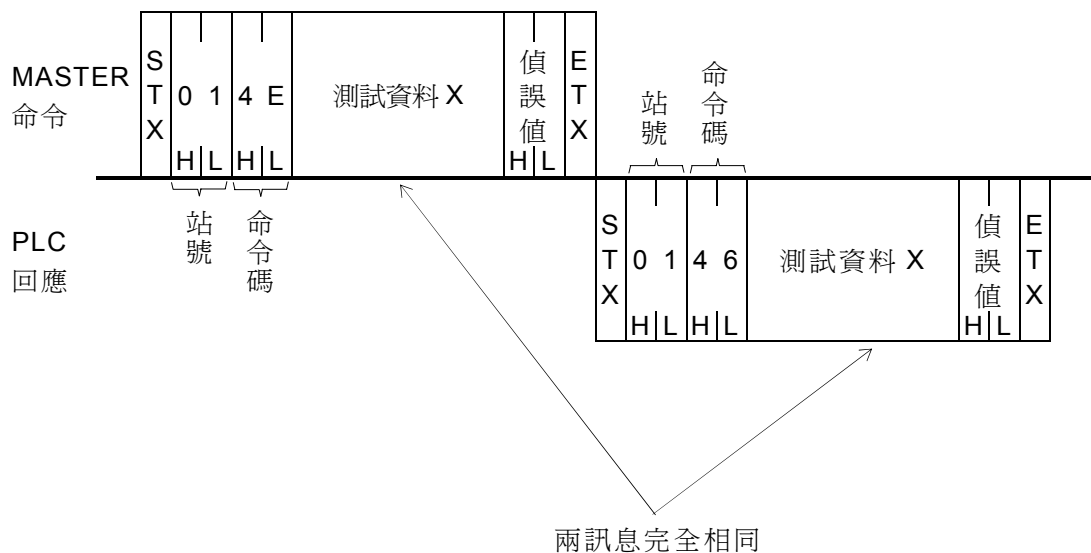
將單點 Y0 設為 1，Y1 設為 0，16 位元暫存器 WM8 設為 5555H，32 位元暫存器 DR2 設為 FFH。



● 命令碼 4E（測試回傳 loop back）

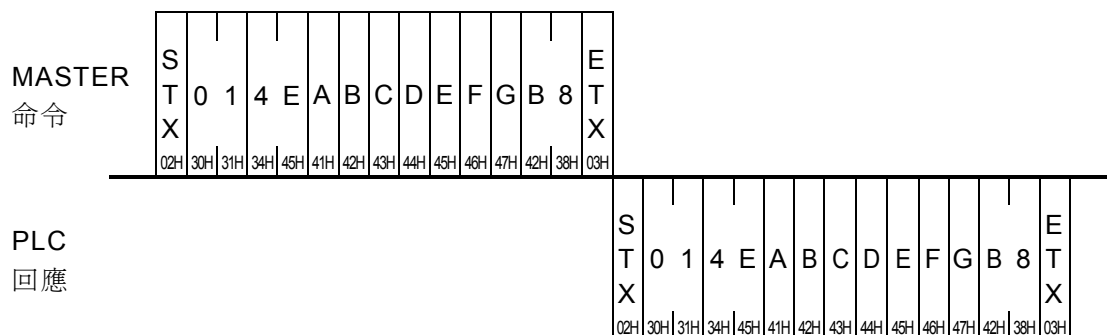
格式

本命令碼 PLC 會將所收到之 MASTER 命令原原本本地回應回去。主要功能是用於測試通訊回路之用，對 PLC 之運作無任何影響。



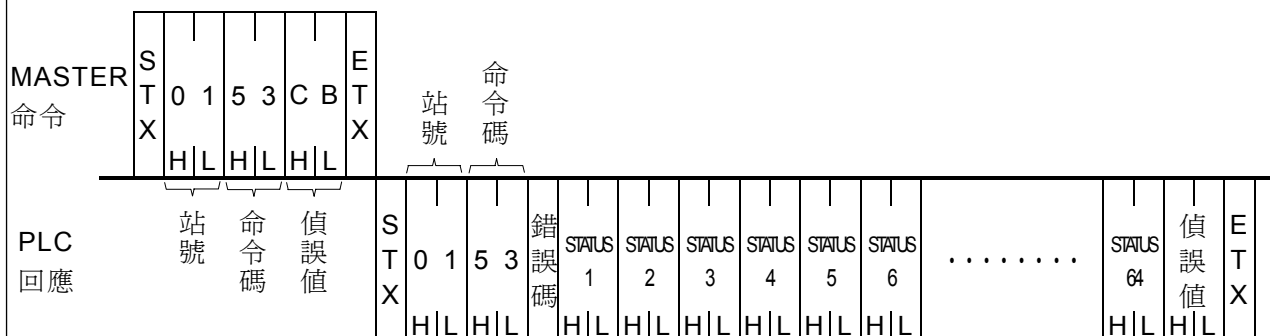
範例

將測試資料“ABCDEFGG”以命令碼 4E 使 PLC 作測試回傳之情形。



● 命令碼 53（PLC 詳細系統狀態讀取）

格式



STATUS 1	B0 : RUN/STOP B1 : 保留 B2 : Ladder checksum error/正常 B3 : 使用 ROM PACK/未使用 B4 : WDT Time out/正常 B5 : 設定 ID/未設 ID B6 : 緊急停機/正常 B7 : 保留			
	STATUS 2	主機 TYPE 00H : MA 01H : MC 其他值 : 預留		
		STATUS 3	主機之 I/O 點數 00H : 10 點 01H : 14 點 02H : 20 點 :	
			STATUS 4	OS Version of PLC 40H : V4.0X 41H : V4.1X :
				STATUS 5
	STATUS 6			Ladder Size Lo-Byte
	STATUS 7	Discrete input Hi-Byte		
	STATUS 8	Discrete input Lo-Byte		
STATUS 9	Discrete output Hi-Byte			
STATUS 10	Discrete output Lo-Byte			
STATUS 11	Register input Hi-Byte			
STATUS 12	Register input Lo-Byte			
STATUS 13	Register output Hi-Byte			
STATUS 14	Register output Lo-Byte			

↓

STATUS 15	M Relay Hi-Byte	
STATUS 16	M Relay Lo-Byte	
STATUS 17	S Relay Hi-Byte	
STATUS 18	S Relay Lo-Byte	
STATUS 19	L Relay Hi-Byte	
STATUS 20	L Relay Lo-Byte	
STATUS 21	R Register Hi-Byte	
STATUS 22	R Register Lo-Byte	
STATUS 23	D Register Hi-Byte	
STATUS 24	D Register Lo-Byte	
STATUS 25	Timer Hi-byte	
STATUS 26	Timer Lo-byte	
STATUS 27	Counter Hi-Byte	
STATUS 28	Counter Lo-Byte	
STATUS 29	<div style="display: flex; justify-content: space-between; align-items: center;">..</div> <div style="display: flex; justify-content: space-between; align-items: center;">..</div> <div style="display: flex; justify-content: space-between; align-items: center;">..</div> <div style="display: flex; justify-content: space-between; align-items: center;">..</div> <div style="display: flex; justify-content: space-between; align-items: center;">..</div> <div style="display: flex; justify-content: space-between; align-items: center;">..</div> <div style="display: flex; justify-content: space-between; align-items: center;">..</div> <div style="display: flex; justify-content: space-between; align-items: center;">..</div> <div style="display: flex; justify-content: space-between; align-items: center;">..</div> <div style="display: flex; justify-content: space-between; align-items: center;">..</div> <div style="display: flex; justify-content: space-between; align-items: center;">..</div> <div style="display: flex; justify-content: space-between; align-items: center;">..</div> <div style="display: flex; justify-content: space-between; align-items: center;">..</div> <div style="display: flex; justify-content: space-between; align-items: center;">..</div> <div style="display: flex; justify-content: space-between; align-items: center;">..</div>	
STATUS 30		
STATUS 31		
STATUS 32		
STATUS 33		
STATUS 34		
STATUS 35		
STATUS 36		
STATUS 37		
STATUS 38		
STATUS 39		
STATUS 40		
STATUS 41		
STATUS 42		
STATUS 43		
STATUS 44		
STATUS 45		
STATUS 46		
STATUS 47		
STATUS 48		
STATUS 49		
STATUS 50		
STATUS 51		
STATUS 52		
STATUS 53		
STATUS 54		
STATUS 55		
STATUS 56		
STATUS 57		
STATUS 58		
STATUS 59		
STATUS 60		
STATUS 61		
STATUS 62		
STATUS 63		
STATUS 64		

● 命令碼 53 (PLC 詳細系統狀態讀取)

範例

假設 PLC 為 FBs-20MC，OS 版本為 V4.0X，程式容量 32K words，未插 ROM PACK，有設定 ID，CPU 狀態均正常且為 RUN 模式，則 Master 去讀取 PLC 之詳細系統狀態會得到如下之結果：

