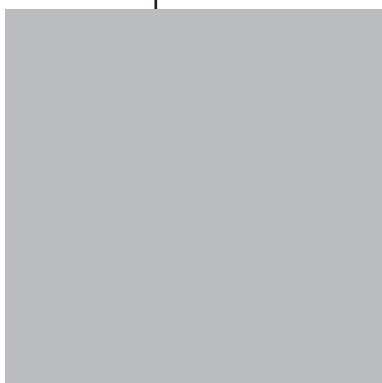




盤裝質量流量控制器 MPC系列使用說明書 通訊功能篇



非常感謝您購買盤裝質量流量控制器 MPC。本使用說明書中記述了正確安全使用 MPC 系列通訊功能的必要事項。對於承擔使用 MPC 系列通訊功能的操作盤、裝置的設計、維護的工作人員請務必在閱讀理解本書的基礎上使用。

此外，本使用說明書不只在安裝時，在維護和故障維修時也是必不可少的。請常備此手冊以供參考。

使用上的限制

本產品是在一般設備上使用前提下開發、設計和製造的。

在有下列安全性要求的場合應用時，請在事故保全設計，冗餘設計及定期維護檢查以及對系統和設備整體等考慮周全的情況下使用。

- 以人體保護為目的的安全裝置
- 輸送設備的直接控制(運行停止等)
- 航空設備
- 宇宙設備
- 原子能設備等

請不要把該產品用在與生命直接相關的用途上。

重要事項

用通訊頻繁變更MPC系列參數的場合，請對RAM的地址進行寫入。對EEPROM的地址進行寫入的場合，只能保證寫入次數在1萬次。

另外，MPC系列停電時，RAM上的數據將消失，回到EEPROM上的數據。

要求

請確保把本使用說明書送到本產品使用者手中。

禁止擅自複印全部或部分本使用說明書。禁止轉載本使用說明書。今後內容變更時恕不事先通知。

本使用說明書的內容，經過仔細審查校對，萬一有錯誤或遺漏，請向本公司提出。

對客戶應用結果，本公司有不能承擔責任的場合，請諒解。

©2003 Yamatake Corporation ALL RIGHTS RESERVED

μF[®]、Micro Flow[®]是株式會社山武的註冊商標。
MPC是株式會社山武的註冊商標。

安全注意事項

■ 圖示說明

本安全注意事項的目的：為了正確安全使用本產品，防患于未然以免給您及他人造成人生損害及財產損失，請一定遵守本安全注意事項。

本書中使用了各種圖形符號，其顯示的含義如下所示，請認真理解所述內容。



警告

當錯誤使用本產品時，可能會造成使用者死亡或重傷的危險情況。



注意

當錯誤使用本產品時，可能會造成使用者輕傷或財物損失的危險情況。

■ 圖示例

	△ 記號：在有明顯地誤操作或誤使用情況下，可能發生危險時，使用△符號表示。圖中有具體注意內容(左圖表示注意觸電)。
	⊘ 記號：為了避免危險發生，禁止某些特定行為時使用的符號。在圖中或在其附近注明具體禁止事項(左圖表示禁止分解)。
	● 記號：為避免危險發生而應盡某些特定行為的義務時使用的符號。圖中有具體指示內容(左圖是表示要把插頭從插座中拔出的意思)。

警告



請不要把本機用于醫療設備上。

注意



對本產品進行接線、安裝、拆卸時，一定要在切斷供給電源後進行。
否則有觸電的危險。



不要對本產品進行分解。
否則有發生故障的危險。



請按規定的標準、指定的電綫及施工方法正確配綫。
否則可能造成本機故障。



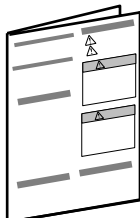
請在規格書中記載的使用條件(溫度、濕度、電壓、振動、撞擊、安裝方向、
環境等)範圍內使用本產品。否則有發生故障的危險。



請不要讓斷綫頭、鐵粉、水進入機箱內。
否則有產生誤動作或故障的危險。

本使用說明書的定位

MPC系列相關使用說明書共分3冊，根據不同的用途，請閱讀相應資料。如果您手裏無相關資料時，請向弊公司或銷售代理店索取。

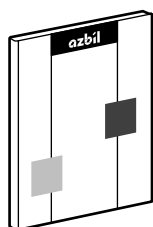


盤裝質量流量控制器 MPC系列

資料編號 CP-UM-5317C

與產品同一包裝。

用本機進行裝置設計、製作的人員，請務必閱讀。本機對使用上的安全注意事項、安裝、接綫及主要規格進行說明。有關詳細的使用方法，請參閱另冊的「詳細篇」。

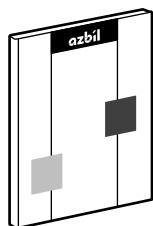


盤裝質量流量控制器 MPC系列詳細篇

資料編號 CP-SP-1153C

對本機的硬件及所有的功能進行說明。進行裝置設計、製作、維護的人員及使用該裝置進行各種操作的人員，請務必閱讀。

對安裝、接綫、本機的所有功能、設定及操作方法、故障時的處理、規格進行詳細說明。



盤裝質量流量控制器 MPC系列通訊功能篇 本書。

資料編號 CP-SP-1154C

使用本機通訊功能的人員請務必閱讀。

對通訊的概要、接綫、通訊步驟及本機的通訊數據一覽、故障時的處理及通訊規格進行說明。

本使用說明書的構成

本使用說明書由以下內容構成。

第1章 概 要

對MPC系列的通訊功能進行說明。

第2章 接 綫

對MPC系列與其他設備間進行通訊的連接方法進行說明。

第3章 設 定

對MPC系列的通訊設定進行說明。

第4章 通訊步驟

對通訊步驟、電文的構成、數據的讀出、寫入與信號的時間進行說明。

第5章 通訊數據一覽

MPC系列通訊中使用的各種數據的地址一覽。

第6章 主站用通訊程序

對MPC系列的通訊程序作成時的注意事項及程序例進行說明。

第7章 故障處理

對MPC系列的通訊在非正常動作時的檢查點進行說明。

第8章 規 格

對MPC系列的通訊規格進行說明。

附 錄

代碼表。

目 錄

安全注意事項
本使用說明書的定位
本使用說明書的構成
本使用說明書的標記

第1章	概 要	1-1
第2章	接 綫	2-1
第3章	設 定	3-1
第4章	通訊步驟	
4-1	通訊步驟與電文的概要	4-1
■	通訊步驟	4-1
■	電文的構成	4-1
■	具體例	4-2
■	數據地址的概念	4-2
4-2	數據鏈層	4-3
■	數據鏈層的說明	4-3
4-3	應用層	4-6
■	應用層的概要	4-6
4-4	數據的讀出	4-7
■	讀出命令的說明	4-7
■	讀出應答	4-8
■	10 進數值的表現(數值數據)	4-9
4-5	數據的寫入	4-10
■	寫入命令的說明	4-10
■	寫入應答	4-11
4-6	結束代碼一覽	4-13
■	正常及警告結束	4-13
■	異常結束	4-13
4-7	時間規格	4-14
■	命令電文、應答電文時間規格	4-14
■	RS-485 驅動控制時間規格	4-14
■	其他的注意事項	4-15

第5 章 通訊數據一覽

5-1	通訊數據使用的預備知識	5-1
■	通訊數據的種類及形式	5-1
■	通訊數據的存儲內存	5-1
■	數據地址	5-2
■	讀出/寫入數據數	5-2
■	數據的單位・小數點位置	5-2
5-2	通訊數據一覽	5-3
■	機種相關數據	5-3
■	動作狀態相關數據	5-4
■	瞬時流量相關數據	5-5
■	積算流量相關數據	5-5
■	功能設定相關數據	5-6
■	參數設定相關數據	5-9

第6 章 主站用通訊程序

6-1	作成時的注意	6-1
6-2	通訊程序例	6-2
■	程序執行前	6-2
■	執行程序	6-2
■	數據讀出/寫入樣板程序	6-3

第7 章 故障處理

■	不能通訊時的確認項目	7-1
---	------------------	-----

第8章 規 格

附 錄

■	代碼表	附-1
■	與CMC10L的连接	附-2

本使用說明書的標記

本使用說明書的標記如下所示。

 使用上的注意事項：表示在使用時敬請注意的事項。

 參考：知道該項內容後助于理解。

：表示敬請參照的項目和頁碼。

①②③：表示操作步驟或者對圖進行相應說明的部分。

》：表示操作結果及操作後的狀態。

OFF：表示顯示部的7段顯示。

「OK」燈：表示顯示部的LED燈。

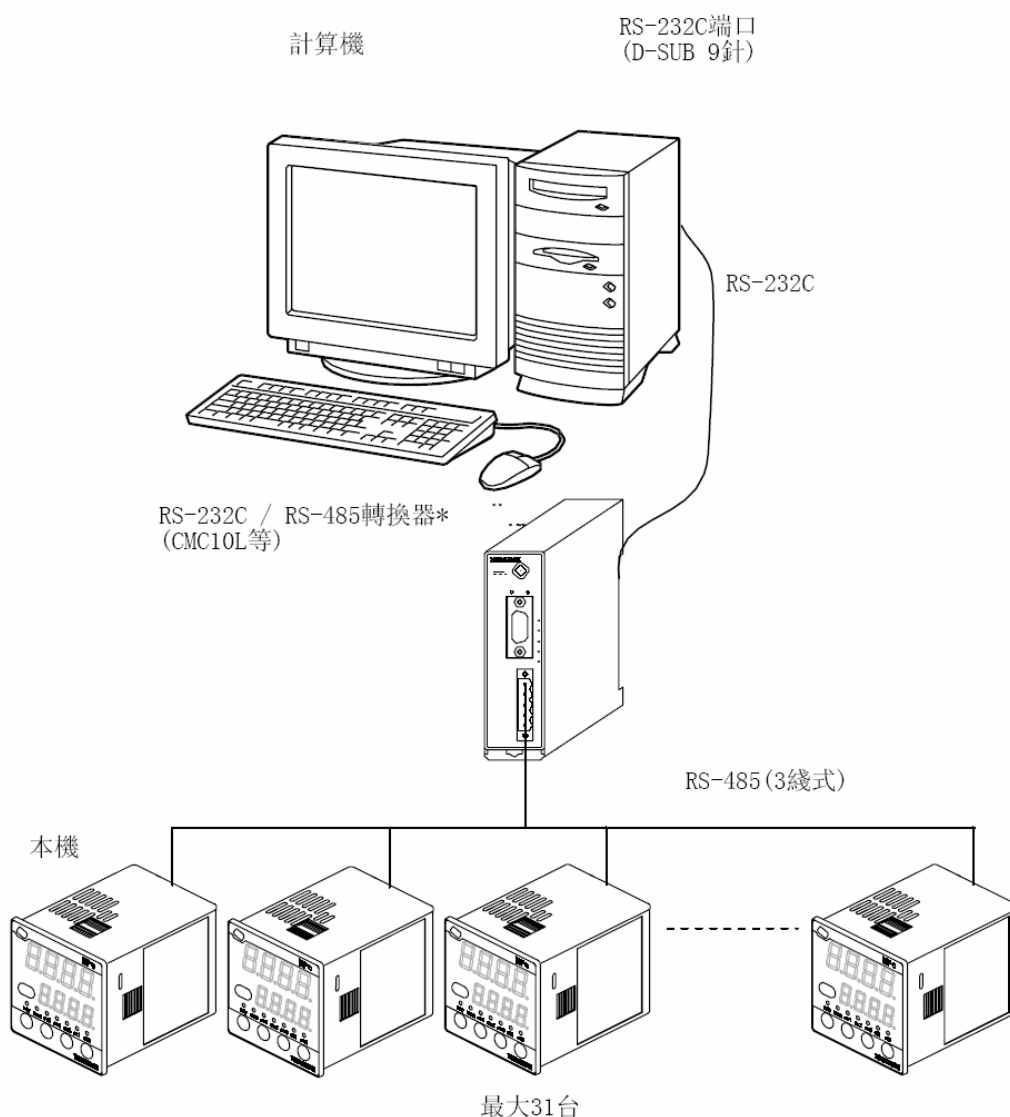
[DISP]鍵：表示顯示部的按鍵。

第 1 章 概 要

把PC或PLC等作為主計算機(也稱為主站),通過RS-232C/ RS-485 轉換器連接,實現與MPC系列(也稱為從站)進行設定值或數據等的交換。

RS-485通訊時,1台主站可最大連接31台從站。為了確定通訊的從站,需要使用“機器地址”。另外,主站對各從站進行的設定寫入或狀態讀出等,需由用戶編寫通訊程序實現。

- 下記順序成立時,可與對方設備進行各種數據的讀出及寫入。
 - ①主站(計算機)向從站(設備)發出命令電文
 - ②主站接收從站的應答電文
- 主站對從站的命令中,有“讀出”與“寫入”的2種。
- 讀寫的數據種類中,根據“數據地址”可自由選擇。



* : RS-232C / RS-485轉換器可使用本公司產品CMC10L001A000。

第 2 章 接 綫

注意



請務必切斷電源後再進行接綫作業。
否則可能產生故障。



請不要分解本機。
否則有產生故障的可能。



請按規定的標準、指定的電綫及施工方法進行正確配綫。
否則可能會產生故障。



請不要讓斷綫頭、鐵粉、水進入機箱內。
否則有產生誤動作、故障的危險。



使用上的注意事項

• 有關RS-485通訊綫以外的接綫方法, 請參閱

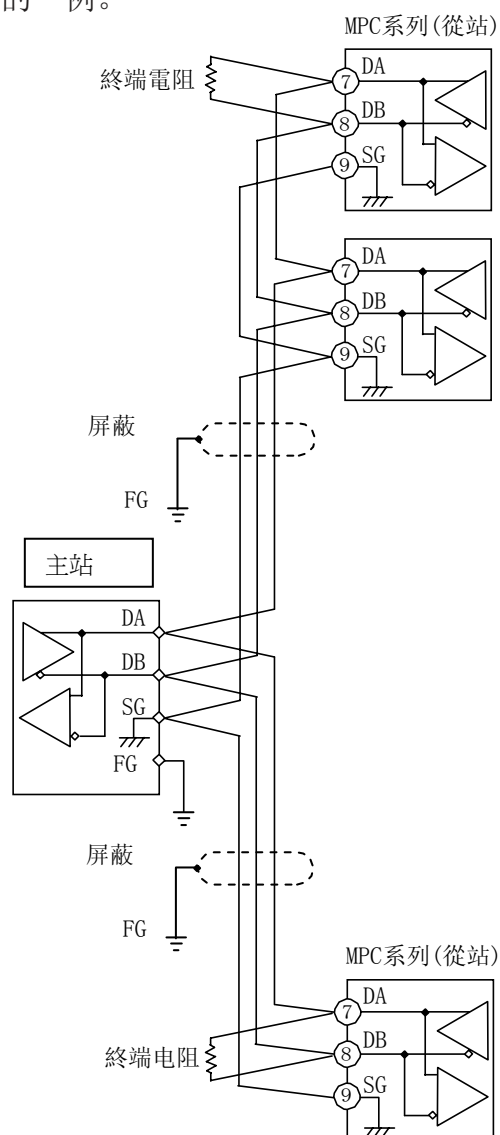


MPC系列使用說明書 CP-UM-5317C 及



MPC系列使用說明書 詳細篇 CP-SP-1153C。

以下是RS-485連接方法的一例。



通訊線路的兩端上, 請安裝 $150\ \Omega \pm 5\%$ 的 $1/2W$ 以上的終端電阻。
 屏蔽的FG接地不是在屏蔽的兩端, 而是單側1點接地。
 作為主站的轉換器, 可使用本公司的產品CMC10L001A000。

❗ 使用上的注意事項

- 請務必連接SG。如果不連接, 將不能獲得穩定的通訊。

第 3 章 設 定

爲了實現通訊功能,請事先按下記順序對本機的通訊條件及機器地址進行設定,使其與主站的設定相吻合。

● 設定方法

按下記方法進行功能設定。

- ①按[DISP]鍵,顯示積算流量。
》「L」燈亮。
- ②按[<]鍵保持3秒以上。
》顯示部1上顯示「0.00」,成爲參數設定模式。
- ③再次按[<]鍵3秒鐘以上。
》顯示部1上顯示項目編號「C-01」,成爲功能設定模式。
- ④按[∧]鍵或[∨]鍵,選擇希望的設定項目編號(下頁)。
- ⑤按[ENT]鍵。
》顯示部2上顯示的現在設定值閃爍。
- ⑥按[∧]鍵或[∨]鍵,選擇目標設定值。
- ⑦目標設定值顯示後,按[ENT]鍵。
》設定值被存儲。
- ⑧如還有其他的設定項目,回到④進行設定。
如沒有其他設定項,則進入⑨。
- ⑨按[DISP]鍵。
》從功能設定模式回到瞬時流量顯示。

❗ 使用上的注意事項

- 進入功能設定模式後1分鐘內,如果不進行任何操作,則自動回到通常顯示(瞬時流量顯示)。
- ⑥的操作中(設定閃爍中)按[DISP]鍵時,設定值不更新,保持前次值。
- 如果機器地址爲0,通訊功能將不會動作。

● 設定項目

顯示項目	項目內容	設定編號及內容	初始值	備注
C-30	機器地址設定	0: 不使用通訊功能 1~127:機器地址	0	0時通訊功能不動作 請設定成與其他從站不同的地址
C-31	傳送速度選擇	0: 38400bps 1: 19200bps 2: 9600bps 3: 4800bps 4: 2400bps	1	
C-32	通訊條件選擇	0: 8 位數據、偶數校驗、 停止位1 1: 8位數據、無奇偶校驗、 停止位2	0	

第 4 章 通訊步驟

4 - 1 通訊步驟及電文的概要

對通訊步驟的概要及電文構成的概念進行說明。

■ 通訊步驟

以下是最簡單的通訊步驟。

1. 從主站向要進行通訊的1台從站發送指定的命令電文。
2. 從站進行命令電文的處理, 進行讀出或寫入。
3. 隨後從站根據處理內容發送應答電文。
4. 主站接收應答電文并進行處理。

■ 電文的構成

一個電文由如下2層構成。主站發出的命令電文及從站發送的應答電文都是如此。

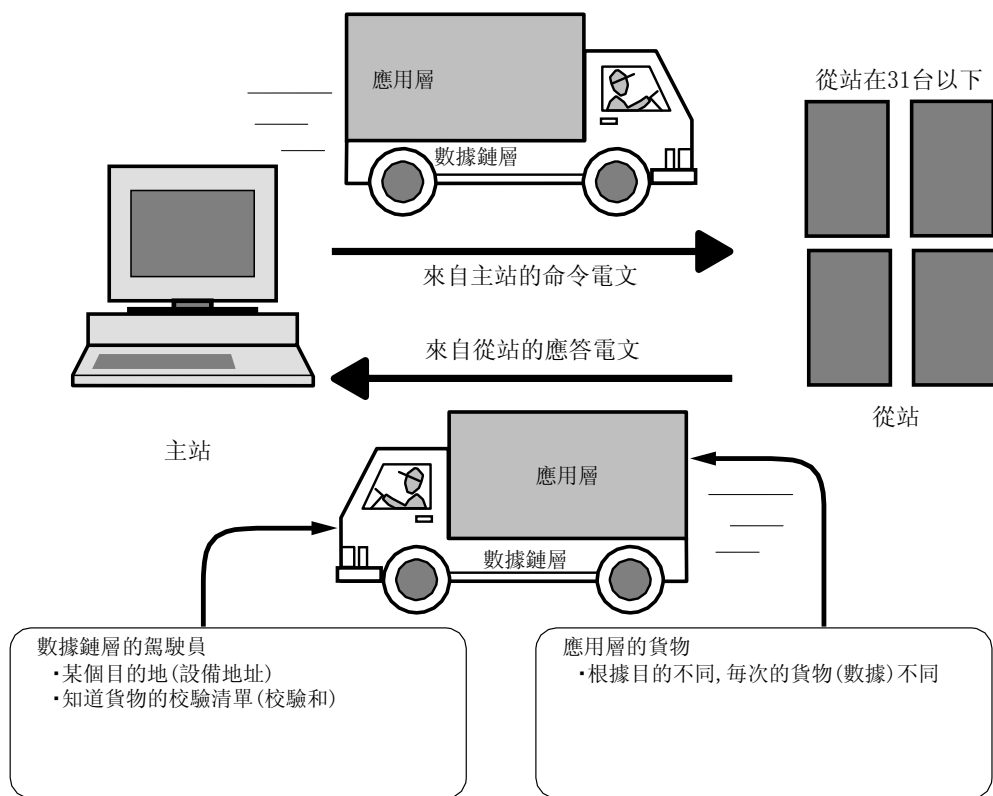
●數據鏈層

- ・含有通訊所必要的基本信息層。
- ・含有通訊電文的目的地或電文的校驗信息。

●應用層

- ・進行數據讀寫的層。
- ・內容根據目的不同而不同。

有關各層的詳細說明, 如下所示。



■ 具體例

具體的電文如下。

● 讀出命令の場合

● 命令電文



● 應答電文

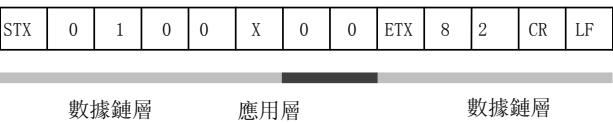


● 寫入命令の場合

● 命令電文



● 應答電文



下項對數據鏈層、應用層的詳細內容進行說明。

■ 數據地址的概念

本機採用了數據地址的概念。
這是由于各數據中與地址對應的數據的讀寫被指定由地址完成。

數據A	1001W
數據B	1002W
數據C	1003W
:	:

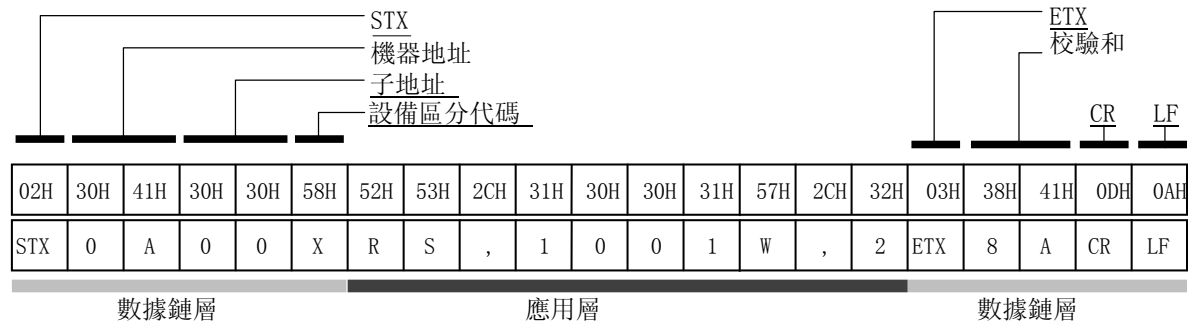
實際的數據及數據地址的對應請參閱
👉5-2 通訊數據一覽(5-3頁)

4 - 2 數據鏈層

■ 數據鏈層的說明

- 數據鏈層上含有用于發送電文的8個基本信息。
- 數據鏈層的命令電文與應答電文的構造相同。

___的文字在本機使用的場合總是一定的。



數據鏈層的各功能表示如下。

● STX(Start of TeXt)

◆作用：表示電文的起始部分。

◇說明

- 固定為02H。
- 設備接收到「STX」時, 即使處於電文中途, 都判斷為新命令電文的第1個文字。

● 機器地址

◆作用：指定發送目的地的機器。可與指定的 1 台機器進行通訊。

◇說明 ▪對指定地址為 0 的機器, 通訊功能停止。

所以, 要進行通訊的場合, 請務必設定1以上的值。

- 采用16進制的2個文字。詳細請參閱下面的例子。

□例：通訊對方的機器地址為10的場合

① 10(10 進制數)=0AH(16 進制數)

② 轉換成字符串代碼

0=30H

A=41H

③ 用②求出的“0A”(30H、41H)作為機器地址

📖 參考

設定的詳細內容請參閱

👉 第3章 設定

❗ 使用上的注意事項

與應用層的數據地址的功能完全不同, 請注意其差異。

● 子地址

◇說明：本機中是固定的。以與機器地址相同的形式, 請務必設定為“00”(30H、30H)。

● 設備區分代碼

◇說明：對本機, 僅可指定字符串代碼的“X”(58H)或“x”(78H)。

● ETX (End of Text)

◆作用：表示在此之前為應用層。

◇說明：▪ 固定為03H。

● 校驗和

◆作用：校驗電文在通訊途中是否因為某種異常(例如干擾)而未產生變化的值。

◇說明：▪ 16進制數的2個文字。

▪ 校驗和的作成方法

① 從STX到ETX的電文按1字節單位進行加算

② 加算結果取2的補碼

③ 轉換成字符串代碼

□例：按前頁的命令電文例的場合進行說明。

① 從STX到ETX的字符串代碼按每1字節加算
計算結果的下位1字節是76H

② 加算結果取2的補碼
結果為8AH

③ 轉換成字符串代碼。把它作為校驗和使用
結果為“8A”, (38H)與(41H)

關於字符串代碼的轉換, 請參閱機器地址例。

❗ 使用上的注意事項

不能省略命令電文的校驗和。

- CR與LF (Carriage Return / Line Feed)

- ◆作用：表示電文的最後。

- ◇說明
 - “CR” 是(0DH), “LF” 是(0AH)。
 - 請務必成對使用CR 與 LF。

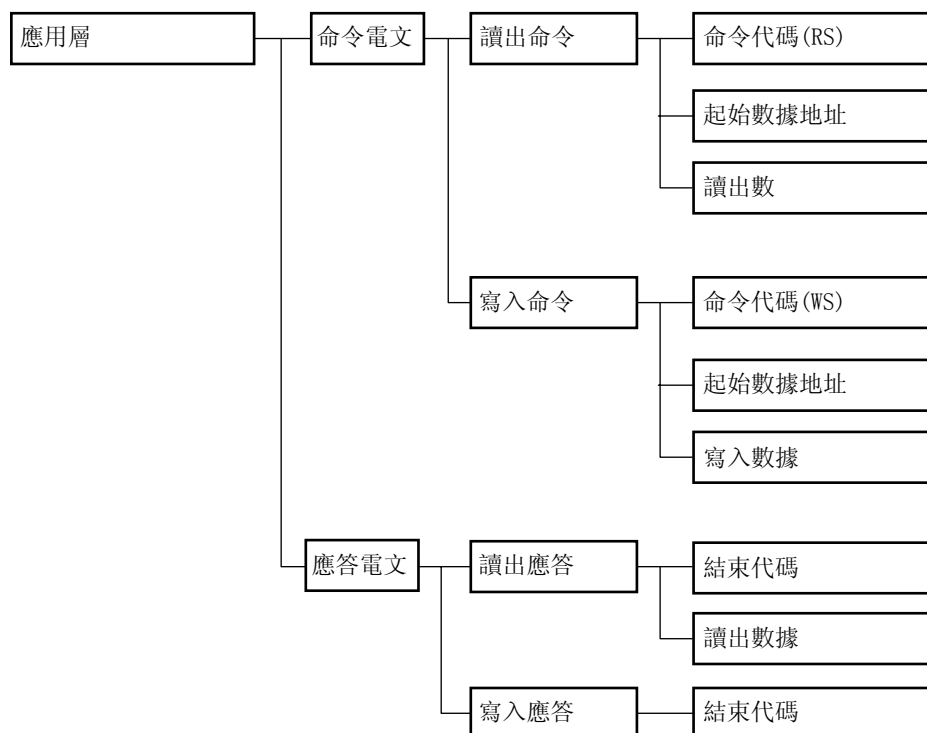
- ❗ 使用上的注意事項

- 如果數據鏈層的內容有以下異常的場合，機器將不會應答。
 - 通訊條件不符合(傳送速度不符, 奇偶校驗錯誤產生等)
 - STX、ETX、CR、LF不處於規定的位置
 - 設備區別代碼不為「X」或「x」
 - 機器地址、子地址、校驗和不是2個文字
 - 校驗和的計算與電文的校驗和不符
 - 含有規定以外的文字
 - 發送的機器地址與對象機器的機器地址不同
 - 機器地址為「00」
 - 除校驗和外, 數據鏈層的內容把與機器的命令電文相同的電文作為應答電文設定。
 - 機器地址及校驗和採用的16進制數的數值部分, 請使用A~F的大寫字母。

4 - 3 應用層

■ 應用的概要

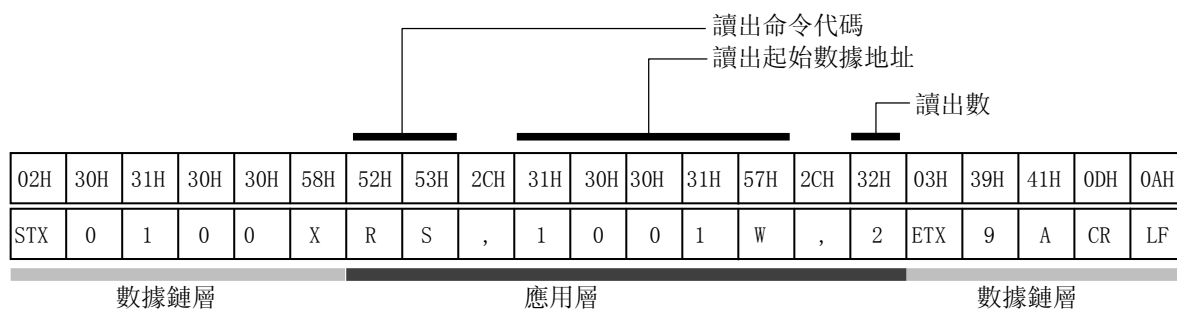
- 應用層中, 含有命令、數據、數據數及電文的判定信息(結束代碼)。
- 應用層中命令電文與應答電文的構造不同。
- 命令電文中有“讀出命令”與“寫入命令”。應答電文根據各自的命令作出對應的應答。
- 根據結束代碼, 可判斷命令電文是被如何處理的。



4 - 4 數據的讀出

■ 讀出命令的說明

- 從指定的讀出起始數據地址開始，可把連續的數據地址的內容用1個電文讀出的命令。
- 讀出命令的應用層由下記3種類的數據構成。



- 各數據之間用逗號“，”（字符串代碼的2CH）進行區分。
 - 應用層的各數值或文字全部使用大寫字符串代碼。
 - 各數值使用10進制數。
 - 各數據中，不可附加多餘的“0”或空格。
- 例： “RS, 01001W, 2” 不能用下劃綫。
 □例： “RS, 1001W, 02” 不能用下劃綫。
 □例： 上圖是從1001W開始把2個數據的信息用1個電文進行讀出的例。

● 讀出命令代碼 (RS)

- ◆作用： 表示讀出的命令。
- ◇說明 “RS”（52H、53H）的2文字。

● 讀出起始數據地址

- ◆作用： 指定執行讀出的起始的數據地址。
- ◇說明
 - 有關數據地址與讀出數據的對應，請參閱 [☞5-2 通訊數據一覽（5-3頁）](#)。
 - 數據地址的數值的後面，務必附上“W”（57H）。

● 讀出數據數

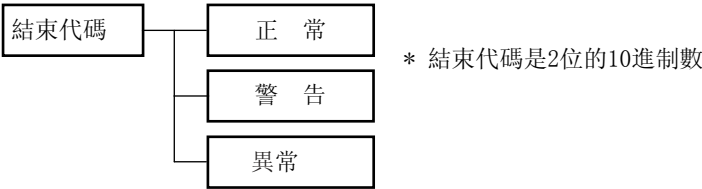
- ◆作用： 從指定的數據地址開始，指定要讀出的連續的幾個數據。
- ◇說明 1個電文能讀出的數據的個數有限制。
詳見 [☞■讀出/寫入數據數（5-2頁）](#)。

■ 讀出應答

數據鏈層的電文正確的場合, 根據命令電文的內容, 答復應答電文。應用層的數據全部用10進制的字符串代碼表示。

● 結束代碼

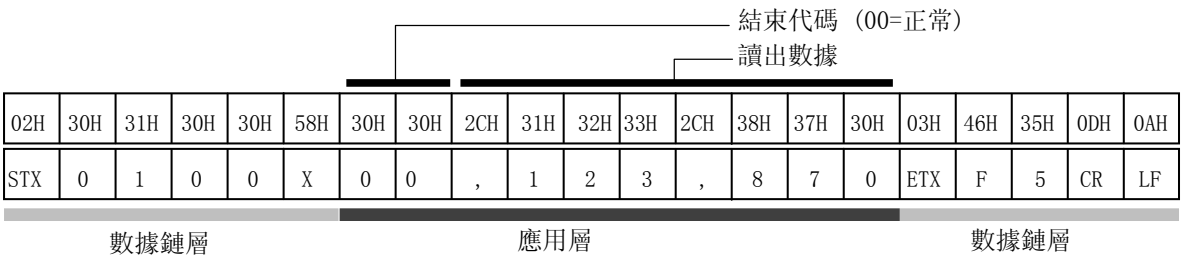
- ◆作用 : 是能知道命令電文在設備側是如何被處理的數值。處理的結果設定成各種各樣的值。
- ◇說明 應答電文中必須含“結束代碼”。
結束代碼的分類如下。



● 正常應答

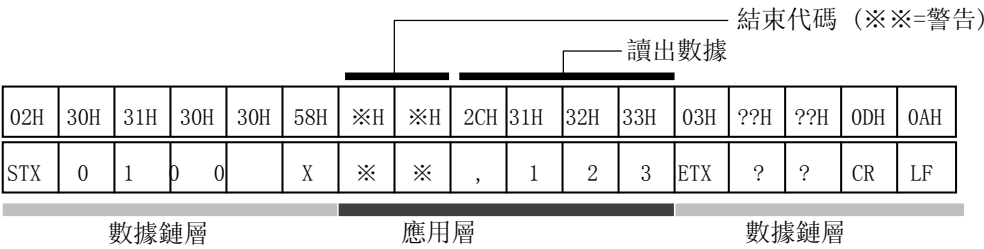
- ◆作用 : 答復指定個數的數據。
 - 數據之間用逗號“,” (2CH)進行區分
 - 各數據的範圍、位數依存于讀出的數據
 - 各數據使用除小數點以外的數值。
- 例 : 把“20.0”作為“200”。
下圖是正常應答的例子。

(讀出數據為2個, 所有的數據被正確讀出時)



● 警告應答

下圖是警告應答的例子。
(※※是異常代碼對應的數值)

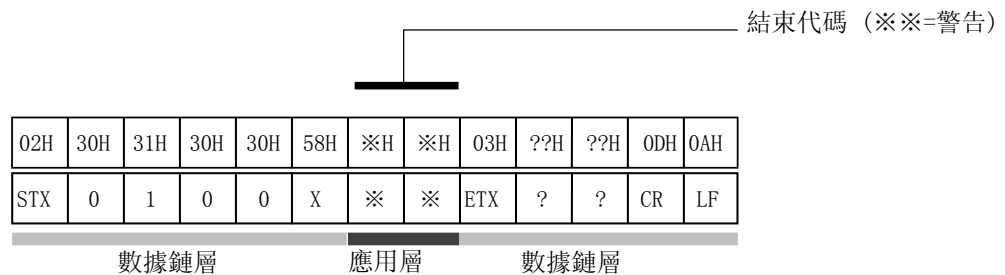


● 異常應答

命令電文中有異常, 不能正常讀出時, 答復異常應答。

下圖是異常應答的例子。

(※※是異常代碼對應的數值)



■ 10進制數值的表現(數值數據)

◆規定：數據地址的數值部、讀出數、寫入值、讀出數據全部遵循以下規定。

①數值為負的場合, 數值前附加負記號“—”(2DH)。

□例：“—123”用(2DH、31H、32H、33H)表示

②數值為0的場合, 取1個0。

□例：“0”用(30H)表示

□例：“00”不能用(30H、30H)表示

③數值為正的場合, 數值前附加正記號“+”。

□例：“+123”不能用(2BH、31H、32H、33H)表示

④數值前不可附加多餘的0或空格。

□例：“0123”不可用(30H、31H、32H、33H)表示

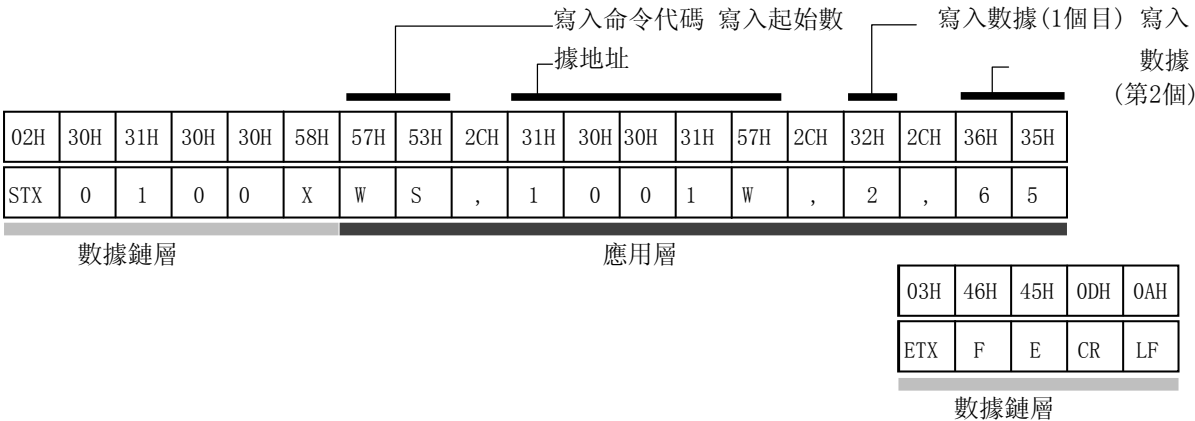
□例：“123”不可用(20H、31H、32H、33H)表示

4 - 5 數據的寫入

■ 寫入命令的說明

從指定的寫入起始數據地址用1個電文對連續的數據地址的內容同時進行複數寫入的命令。

寫入命令的應用層由如下3種類的數據構成。



各數據之間用逗號“,” (字符串代碼的2CH) 進行區分。

寫入數據的個數不需要指定。

應用層的各數值或文字, 全部使用大寫字符串代碼。

各數值採用10進制數。

各數據中不可附加多餘的“0” (30H) 或空格。

□例： “WS, 01001W, 2” 不能用下劃綫

□例： WS, 1001W, 02」不能用下劃綫

□例： 上圖是用1個電文把2及65分別寫入1001W地址及1002W地址中的例

● 寫入命令代碼 (WS)

◆作用：表示寫入命令。

◇說明 “WS” (57H、53H) 的2個文字。

● 寫入起始數據地址

◆作用：指定執行寫入的起始的數據地址。

▪ 有關數據地址與寫入數據的對應關係, 請參閱

👉 5-2 通訊數據一覽 (5-3頁)。

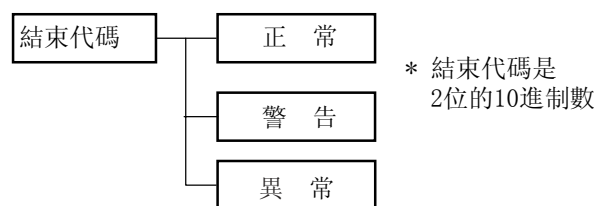
▪ 在數據地址的數值後面, 必須附加 “W” (57H)。

● 寫入數據

- ◆作用：從指定的數據地址向連續的數據地址中寫入數據。
- ◇說明
 - 根據各數據地址，能寫入的數值的範圍不同。
 - 各數據用逗號(2CH)區分。
 - 與各數據對應的能被寫入的數據地址，是從起始數據地址開始按順序各加1後得到的值。(參照前頁的例)
 - 1個電文能夠寫入數據的數量有限制。詳細請參閱  ■讀出/寫入數據數 (5-2頁)。

■ 寫入應答

- ◆作用：數據鏈層的電文正確の場合，只答復結束代碼。
- ◇說明 結束代碼的分類如下。



● 正常應答

- ◆作用：寫入正常完成的場合，答復正常應答。
- 例：正常應答的例子(所有數據正確寫入時)

												結束代碼 (00=正常)
02H	30H	31H	30H	30H	58H	30H	30H	03H	38H	32H	0DH	0AH
STX	0	1	0	0	X	0	0	ETX	8	2	CR	LF
數據鏈層						應用層		數據鏈層				

● 警告應答

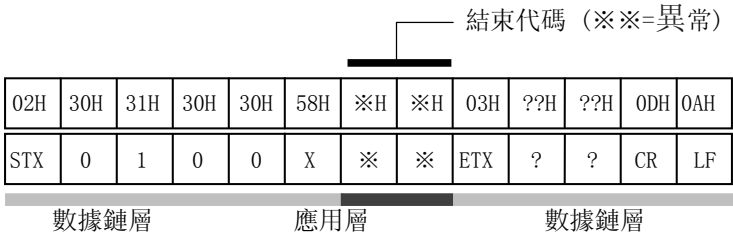
下圖是警告應答時的例子。
(※※是異常代碼對應的數值)

												結束代碼 (※※=警告)
02H	30H	31H	30H	30H	58H	※H	※H	03H	??H	??H	0DH	0AH
STX	0	1	0	0	X	※	※	ETX	?	?	CR	LF
數據鏈層						應用層		數據鏈層				

● 異常應答

◆作用：命令電文有異常, 不能寫入處理の場合, 答復異常應答。

□例：異常應答時
(※※是異常代碼對應的數值)



4 - 6 結束代碼一覽

■ 正常及警告結束

結束代碼	種 類	內容及動作
00	正常	正常結束
21	警告	由指定的外部切換輸入等進行的通訊, 向不可設定的地址進行了寫入操作。 該地址中不寫入任何內容, 處理繼續進行。
23	警告	由于訪問了範圍外的地址, 所以終止了讀出。 由于訪問了範圍外的地址, 所以終止了寫入。 但已經執行了在此之前的寫入處理。

■ 異常結束

結束代碼	種 類	內容及動作
40	異常	地址中未設定W 廢棄了所有電文
41	異常	未設定WS或RS 廢棄了所有電文
43	異常	ETX(03H) 未設定到正確的位置 未設定地址後面的“,” 廢棄了所有電文
46	異常	地址異常 廢棄了所有電文
47	異常	讀出的WORD數異常 廢棄了所有電文
48	異常	寫入數值異常 除發生異常的地址外, 執行了寫入處理
99	異常	未定義命令或其他電文異常 廢棄了所有電文

4 - 7 時間規格

■ 命令電文、應答電文時間規格

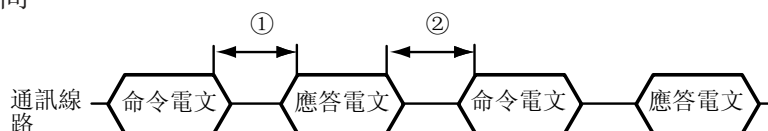
有關主站的命令電文發送與從站的應答電文發送的時間規格, 需要注意以下事項。

● 應答監視時間

主站發送命令電文完畢後到開始接收從站的應答電文為止的最長應答時間為2秒。(①的部分), 所以, 請把應答監視時間設定為2秒。通常當應答監視時間到的場合, 執行命令電文的再發送。詳見

➡第6章 主站用通訊程序。

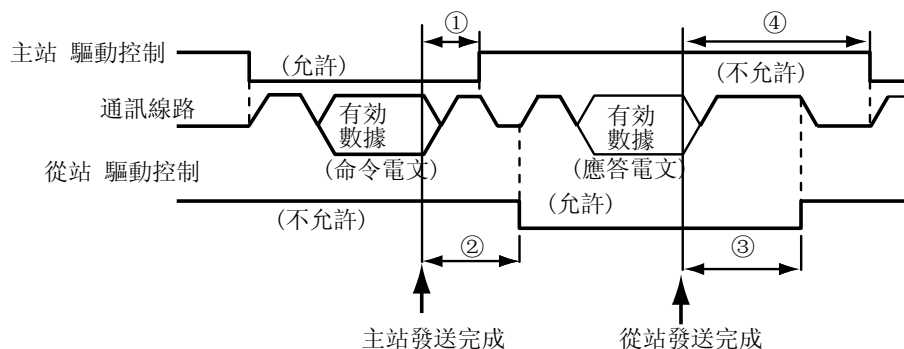
● 發送開始時間



- ① 主站發送完畢後到從站開始發送為止的時間 = 2000ms以下
(請把主站的命令電文發送完畢後的應答監視時間設定成2s)
- ② 從站完成發送後到主站發送開始為止的時間 = 10ms以上
(從主站的應答電文接收完畢後到下一個發送開始為止, 請等待10ms以上)

■ RS-485 驅動控制時間規格

由主站對RS-485的發送/接收直接控制的場合, 請注意以下時序。



- ① 主站發送完畢 - 驅動不允許時間 = 500 μ s 以下
- ② 從站接收完畢 - 驅動允許時間 = 15ms 以上
- ③ 從站發送完畢 - 驅動不允許時間 = 10ms 以下
- ④ 主站接收完畢 - 驅動允許時間 = 10ms 以上

■ 其他的注意事項

- 由1個電文讀出/寫入的數據越多, 則從主站命令電文發送完畢到從站發送應答電文的時間就越慢。
要加快從站的應答, 可減少用1個電文進行的讀出/寫入的數據的數量。
- 當1個電文的讀出/寫入數據數量為1個數據時, 從主站命令電文發送完畢到從站發送應答電文的時間約為30ms。

第 5 章 通訊數據一覽

5 - 1 使用通訊數據前的預備知識

■ 通訊數據種類及形式

● 通訊數據種類

通訊數據可分為以下幾種類型。

- 機器種類相關數據
- 動作狀態相關數據
- 瞬時流量相關數據
- 積算流量相關數據
- 功能設定相關數據
- 參數設定相關數據

● 通訊數據的形式

通訊數據的形式劃分如下。

- 數值數據：用數值顯示的數據。(PV、SP等)
- 位數據：每位都有含義的數據。
(報警等)

位數據需要在發送時合成, 接收時分解。

重要事項

對EEPROM的寫入次數小於1萬次。因為通訊需要對SP等數據進行頻繁地反復寫入的場合, 請把RAM作為存儲對象。

■ 通訊數據的存儲內存

● 內存種類

通訊數據存儲在機器的內存(記憶單元)中。本機可使用的內存有2種。

- RAM：切斷電源後數據消失, 但可反復寫入。
- EEPROM：切斷電源後數據也不消失。但記憶單元的特性對寫入次數有限制。
限制次數為1萬次。

● 通訊對象內存

在通訊中，對上述2種內存，有必要按目的及用途進行讀出/寫入。
對象內存的差異如下。

- RAM：僅對RAM進行讀出/寫入。對RAM寫入後，電源切斷後再通電時，EEPROM內的數據複製到RAM中，RAM的數據與EEPROM相同。
- EEPROM：對RAM及EEPROM兩者進行寫入。

■ 數據地址

數據地址的規定如下表所示。

通訊數據	RAM		EEPROM	
	偏移值	地址	偏移值	地址
機器種類相關數據	1000	1001～1199	4000	4001～4199
動作狀態相關數據	1200	1201～1399	4200	4201～4399
瞬時流量相關數據	1400	1401～1599	4400	4401～4599
積算流量相關數據	1600	1601～1799	4600	4601～4799
功能設定相關數據	2000	2001～2199	5000	5001～5199
參數設定相關數據	2200	2201～2399	5200	5201～5399

■ 讀出/寫入數據數

1次通訊能連續讀出/寫入的數據數量如下。

	RAM	EEPROM
讀出	1～10WORD	1～10WORD
寫入	1～10WORD	1～10WORD

■ 數據的單位・小數點位置

讀出/寫入數據中，不附加小數點。

單位或小數點位置由各數據決定。

有關各數據的單位・小數點位置，請參閱機器本體的使用說明書。

5 - 2 通訊數據一覽

對各數據的地址及可讀出/寫入(R/W)狀態的規定如下表所示。

R/W 欄記號的含義

○可能

×不可

■ 機器種類相關數據

顯 示	數據範圍	RAM			EEPROM			備 注
		地址	R	W	地址	R	W	
氣體種類	0 : 用戶設定 1 : 氮氣/空氣 3 : 氫氣 4 : 二氧化碳	1001	○	×	4001	×	×	可由功能設定 (地址5018)變更氣體的種類
全量程流量	根據流量量程	1002	○	×	4002	×	×	除去小數點後的值
瞬時流量小數點 顯示位置	0 : 無小數點 1 : xxxx. 2 : xxx. x 3 : xx. xx 4 : x. xxx	1003	○	×	4003	×	×	
積算流量小數點 顯示位置	0 : 無小數點 1 : xxxxxxxx. 2 : xxxxxxxx. x 3 : xxxxxxx. xx 4 : xxxxx. xxx	1004	○	×	4004	×	×	

■ 動作狀態相關數據

顯 示	數據範圍	RAM			EEPROM			備 注
		地址	R	W	地址	R	W	
報警狀態位	*1參照	1201	○	×	4201	×	×	狀態用10進制數表示
事件狀態位	*2參照	1202	○	×	4202	×	×	狀態用10進制數表示
控制狀態位	*3參照	1203	○	×	4203	×	×	狀態用10進制數表示
動作模式	0:閥全閉 1:閥控制 2:閥全開	1204	○	○	4204	○	○	外部輸入強制全閉/全開時, 不寫入
使用中的SP號	0:SP-0 1:SP-1 2:SP-2 3:SP-3	1205	○	○	4205	○	○	選擇了外部輸入切換SP號時, 不寫入 對選擇了比功能設定的SP數 (地址5004) 大的值, 不寫入
使用中的SP值 (設定流量)	(0~100%FS) L/min(standard)	1206	○	×	4206	×	×	全量程流量乘以括號內的比例後的流量 除去 (L/min(standard)) 小數點後的值
PV值(控制流量)	(0~100%FS) L/min(standard)	1207	○	×	4207	×	×	
閥驅動輸出	0.0~100.0%	1208	○	×	4208	×	×	除去小數點後的值

*1: 報警狀態位構成(地址 1201)

0:正常 1:異常

位No.	內 容
0	瞬時流量偏差下限報警
1	瞬時流量偏差上限報警
2	未定義(通常為0)
3	未定義(通常為0)
4	傳感器異常
5	輸入輸出調整數據異常
6	傳感器校正數據異常
7	用戶設定數據異常
8	閥過熱防止限幅動作

*2: 事件狀態位構成(地址 1202)

0:OFF 1:ON

位No.	內 容
0	事件1輸出狀態
1	事件2輸出狀態
2	未定義(通常為0)
3	外部接點1輸入狀態
4	外部接點2輸入狀態
5	未定義(通常為0)
6	未定義(通常為0)
7	未定義(通常為0)

*3: 控制狀態位構成(地址 1203)

位No.	內 容
0	OK燈(瞬時PV控制狀態) 0:燈滅 1:燈亮(瞬時PV OK)
1	慢啓動動作 0:通常動作 1:慢啓動動作
2	數字設定/模擬設定 0:數字設定 1:模擬設定
3	積算計數狀態 0:積算流量<積算事件設定 1:積算流量≥積算事件設定
4	未定義(通常爲0)
5	未定義(通常爲0)
6	未定義(通常爲0)
7	未定義(通常爲0)

■ 瞬時流量相關數據

顯 示	數據範圍	RAM			EEPROM			備 注
		地址	R	W	地址	R	W	
設定流量 SP-0	(0~100%FS) L/min(standard)	1401	○	○	4401	○	○	數據範圍爲全量程流量乘以括號內的 比例後的流量 除去(L/min(standard))小數點 後的值
設定流量 SP-1	(0~100%FS) L/min(standard)	1402	○	○	4402	○	○	
設定流量 SP-2	(0~100%FS) L/min(standard)	1403	○	○	4403	○	○	
設定流量 SP-3	(0~100%FS) L/min(standard)	1404	○	○	4404	○	○	

■ 積算流量相關數據

顯 示	數據範圍	RAM			EEPROM			備 注
		地址	R	W	地址	R	W	
積算事件設定 流量下4位	0~9999	1601	○	○	4601	○	○	參數設定的地址2217及5217與該 地址相同
積算事件設定 流量上4位	0~9999	1602	○	○	4602	○	○	參數設定的地址2218及5218與該 地址相同
積算流量下4位	0~9999	1603	○	○	4603	○	○	積算流量復位時 請在上位及下位兩者中寫入0
積算流量上4位	0~9999	1604	○	○	4604	○	○	

■ 功能設定相關數據

顯 示	數據範圍	RAM			EEPROM			備 注
		地址	R	W	地址	R	W	
設定鎖定	0:無鎖定 1:鎖定流量設定以 外的設定 2:鎖定全部設定	2001	○	○	5001	○	○	
動作模式切換 選擇(由鍵操作的 切換)	0:鍵操作不執行動 作模式切換 1:鍵操作執行動作 模式切換	2002	○	○	5002	○	○	
流量設定方法選擇 (SP設定方法選擇)	0:數字設定(由鍵操 作或通訊進行設定) 1:模擬設定(由外部 模擬輸入電壓設定)	2003	○	×	5003	○	×	*1
流量設定數選擇 (SP數選擇)	0:SP數=1(僅SP-0) 1:SP數=2 (SP-0、SP-1) 2:SP數=3 (SP-0~SP-2) 3:SP數=4 (SP-0~SP-3)	2004	○	○	5004	○	○	
模擬設定時、 輸入電壓量程選 擇(模擬SP輸入 量程選擇)	0:0~5V輸入 1:1~5V輸入	2005	○	×	5005	○	×	*1
流量模擬輸出 電壓量程選擇 (PV模擬輸出 量程選擇)	0:0~5V輸出 1:1~5V輸出	2006	○	×	5006	○	×	*1
事件1輸出種類 指定	0:不使用 (通常為OFF)	2007	○	○	5007	○	○	
事件2輸出種類 指定	1:報警發生時ON 2:積算脈衝輸出 3:流量OK時ON 4:動作模式= 控制中ON 5:動作模式= 全開中ON 6:動作模式= 控制中或全 開中ON 7:動作模式= 全閉中ON 8:流量上限事件 9:流量下限事件① 10:流量下限事件② 11:積算流量事件 —1~—11:上述 1~11的反轉輸出 (通常:ON、事件 發生時:OFF)	2008	○	○	5008	○	○	

顯 示	數據範圍	RAM			EEPROM			備 注
		地址	R	W	地址	R	W	
未定義	0	2009	○	×	5009	○	×	*1
外部開關輸入1 功能指定	0:不使用 1:積算復位	2010	○	○	5010	○	○	
外部開關輸入2 功能指定	2:積算計數動作停止 3:SP號切換 4:流量設定方法切換 5:動作模式強制全閉 6:動作模式強制全開 7:慢啓動動作切換 8:動作模式切換 (接點ON時控制、 OFF時強制全閉)	2011	○	○	5011	○	○	
未定義	0	2012	○	×	5012	○	×	*1
積算事件發生時 的閥自動切斷功能	0:無功能 1:有功能	2013	○	○	5013	○	○	
控制開始時的積算 復位功能切換	0:無功能 1:有功能	2014	○	○	5014	○	○	
流量報警設定 種類	0:不使用 1:僅使用上限報警 2:僅使用下限報警 3: 使用上下限報警	2015	○	○	5015	○	○	
報警發生時的 動作選擇	0:控制繼續 (忽略報警) 1:強制全閉 2:強制全開	2016	○	○	5016	○	○	
緩慢啓動設定	0:無緩慢啓動 1~8:有緩慢啓動 (整定時間約相當 于1~6s)	2017	○	○	5017	○	○	
氣體種類設定	0:各氣體種類的 補償係數(C.F.), 由用戶設定 1:空氣、氮氣 3:氫氣 4:二氧化碳(CO ₂)	2018	○	○	5018	○	○	
流量顯示單位選擇	0:20℃ 1個大氣壓標準 1: 0℃ 1個大氣壓標準 2:25℃ 1個大氣壓標準 3:35℃ 1個大氣壓標準	2019	○	○	5019	○	○	
入口壓設定	0:0~0.1MPa 1:0.05~0.15MPa 2:0.15~0.25MPa 3:0.25~0.35MPa 4:0.35~0.45MPa 5:0.45~0.5MPa	2020	○	○	5020	○	○	
直接設定功能	0:無功能 1:有功能	2021	○	○	5021	○	○	
未定義	0	2022	○	×	5022	○	×	*1

顯 示	數據範圍	RAM			EEPROM			備 注
		地址	R	W	地址	R	W	
PV濾波 (平均化)	0: 無PV濾波 1: 採樣2次的移動 平均 2: 採樣4次的移動 平均 3: 採樣8次的移動 平均	2023	○	○	5023	○	○	
未定義	0	2024	○	×	5024	○	×	*1
未定義	0	2025	○	×	5025	○	×	*1
未定義	0	2026	○	×	5026	○	×	*1
未定義	0	2027	○	×	5027	○	×	*1
模擬任意 量程功能	0: 無功能 1: 有功能	2028	○	×	5028	○	×	*1
PV強制置零功能	0: 無功能 1: 有功能	2029	○	○	5029	○	○	
機器地址設定	0: 不使用通訊功能 1~127: 機器地址	2030	○	×	5030	○	×	*1
傳送速度選擇	0: 38400bps 1: 19200bps 2: 9600bps 3: 4800bps 4: 2400bps	2031	○	×	5031	○	×	*1
通訊條件選擇	0: 8位數據、 偶數校驗、 停止位1 1: 8位數據、無校驗、 停止位2位	2032	○	×	5032	○	×	*1

*1 : 執行寫入也能答復正常應答，但不進行數據寫入。

■ 參數設定相關數據

顯 示	數據範圍	RAM			EEPROM			備 注
		地址	R	W	地址	R	W	
瞬時流量OK判定範圍	(0.5~100% FS) L/min(standard)	2201	○	○	5201	○	○	數據範圍為全量程流量乘 以括號內的比例後的流量 (L/min(standard)) (根據機器種類, 數據範圍不瞬 同) *2
瞬時流量OK判定 回差	(0.5~100% FS) L/min(standard)	2202	○	○	5202	○	○	
時流量偏差上限 報警	(0.5~100% FS) L/min(standard)	2203	○	○	5203	○	○	
瞬時流量偏差上限 報警回差	(0.5~100% FS) L/min(standard)	2204	○	○	5204	○	○	
瞬時流量偏差下限 報警	(0.5~100% FS) L/min(standard)	2205	○	○	5205	○	○	
瞬時流量偏差下限 報警回差	(0.5~100% FS) L/min(standard)	2206	○	○	5206	○	○	
瞬時流量偏差 報警判定 延時時間	1.0~999.9s	2207	○	○	5207	○	○	*2
事件輸出1 延時時間	0.0~999.9s	2208	○	○	5208	○	○	積算脈衝輸出時, 即使設定了 延時時間, 也無効。 *2
事件輸出2 延時時間	0.0~999.9s	2209	○	○	5209	○	○	
用戶設定 轉換係數(C.F.)	0.100~9.999	2210	○	○	5210	○	○	僅當功能設定的氣體種類選擇 (地址5018), 選擇了“用戶設 定“時, 設定值才有效。 *2
未定義	0	2211	○	×	5211	○	×	*1
未定義	0	2212	○	×	5212	○	×	*1
事件輸出1 上下限流量設定	(0~100% FS) L/min(standard)	2213	○	○	5213	○	○	*2
事件輸出2 上下限流量設定	(0~100% FS) L/min(standard)	2214	○	○	5214	○	○	*2
未定義	0	2215	○	×	5215	○	×	*1、*2
未定義	0	2216	○	×	5216	○	×	*1、*2
模擬任意量程	(10~100% FS) L/min(standard)	2217	○	×	5217	○	×	*1、*2
積算事件設定 流量下4位	0~9999	2218	○	○	5218	○	○	地址1601及4601與該地址相同 *2
積算事件設定 流量上4位	0~9999	2219	○	○	5219	○	○	地址1602及4602與該地址相同
PV強制置零功能 延時時間	0.0~999.9s	2220	○	○	5220	○	○	*2

*1 : 執行寫入也能答復正常應答, 但不進行數據寫入。

*2 : 所有數據都是舍去小數點後的值。

第 6 章 主站用通訊程序

6 - 1 製作時的注意事項

- 機器的最長應答時間為2秒。所以，請把應答監視時間設為2秒。
- 2 秒以內沒有應答的場合，請再次發送同樣的電文。
- 通訊中由于干擾等的影響，電文不能正確傳遞時，請務必進行上述的再次發送。

參 考

命令電文的設備區分代碼中，如果使用“X”，則應答電文的設備區分代碼也為“X”，同樣，命令電文中使用“x”的話，則應答電文也使用“x”。主站再次發送電文時，通過交替使用設備區分代碼“X”與“x”，能方便的判斷出收到的應答電文是上次的電文還是本次的電文。

6 - 2 通訊程序例

本節所示程序例是用Borland 公司的Windows95/98/NT/2000版C++ Builder 5.0 或Borland C ++ Compiler 5.5 編制的。該程序僅作為客戶編制程序時的參考，並不保證所有操作。Borland C ++ Compiler 5.5可從Borland公司網站下載。

■ 程序執行前

請確認機器的通訊條件、機器地址。

■ 執行程序

該程序執行數據的讀出及數據的寫入。執行後，顯示通訊的命令電文、應答電文的應用層。

```
命令 RS, 1000W, 2 執  
行結果 00, 0, 0  
命令 WS, 1000W, 2  
執行結果 00
```

執行結果顯示例

● 執行進行通訊的設定

調出Open ()，初始化RS-232C 。

● 執行命令

Command 中設定要執行的文字列，調出AppCPL()。

■ 數據讀出/寫入樣板程序

❗ 使用上的注意事項

使用本程序例所產生的不良後果，本公司不負任何責任。

```
//-----
// C++ Builder 5
// Borland C++
// bcc32 cpl.cpp
//
// cygwin + gcc
// gcc cpl.cpp
//
#include <stdio.h>
#include <windows.h>
#pragma hdrstop

#define COMRESENDNUM 2
#define BUFFERSIZE 4096
#define TIMEOUT 2000

HANDLE handle;
unsigned long ErrorCode;
bool CheckSum;

bool AppCPL( char* tosend, char * received );
int Open( void );
int Close( void );
bool Write( unsigned char *Mesg, unsigned long Size );
bool Read( unsigned char *Buffer, unsigned long SizeToRead,
           unsigned long *ReadSize, unsigned long *ErrFlag );
void CPLSum( unsigned char *str, int len, unsigned char *buf );

//-----
int main(int argc, char* argv[])
{
    char command[255];
    char recieve[255];

    handle = (void *)0xffffffff;
    ErrorCode = 0;
    CheckSum = true;

    if(Open()==0){
        strcpy(command,"RS,1000W,2");
        AppCPL(command,recieve);
        printf("命令 :%s\n",command);
        printf("執行結果:%s\n",recieve);

        strcpy(command,"WS,1000W,2");
        AppCPL(command,recieve);
        printf("命令 :%s\n",command);
        printf("執行結果:%s\n",recieve);
        Close();
        getchar();
    }
    return 0;
}

//-----
int Open(void)
{
    COMMTIMEOUTS Timeouts;
    _DCB DCB;

    handle = CreateFile( "\\\\.\\COM1", GENERIC_READ|GENERIC_WRITE,
                        0, 0, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, 0 );
    if( handle==(void *)0xffffffff ) return 3;

    if( !SetupComm( handle, BUFFERSIZE, BUFFERSIZE ) ){
        CloseHandle( handle );
        handle = (void*)0xffffffff;
        return 4;
    }
}
```

```

if( !PurgeComm( handle, PURGE_TXABORT|PURGE_RXABORT|
    PURGE_TXCLEAR|PURGE_RXCLEAR ) ){
    CloseHandle( handle );
    handle = (void*)0xffffffff;
    return 5;
}

Timeouts.ReadIntervalTimeout      = 0xFFFFFFFF;
Timeouts.ReadTotalTimeoutMultiplier = 0;
Timeouts.ReadTotalTimeoutConstant = 0;
Timeouts.WriteTotalTimeoutMultiplier = 0;
Timeouts.WriteTotalTimeoutConstant = 0;

if( !SetCommTimeouts( handle, &Timeouts ) ){
    CloseHandle( handle );
    handle = (void*)0xffffffff;
    return 6;
}

if( !GetCommState( handle, &DCB ) ){
    CloseHandle( handle );
    handle = (void*)0xffffffff;
    return 7;
}

DCB.BaudRate=CBR_19200;
DCB.fBinary=1;
DCB.fParity= 1;
DCB.ByteSize=0x08;
DCB.Parity=EVENPARITY;
DCB.StopBits=ONESTOPBIT;

if( !SetCommState( handle, &DCB ) ){
    CloseHandle( handle );
    handle = (void*)0xffffffff;
    return 8;
}

return 0;
}

int Close( void )
{
    if( handle!=(void *)0xffffffff ){
        if( !SetCommMask( handle, 0 ) ){
            CloseHandle( handle );
            handle = (void*)0xffffffff;
            return 2;
        }
        if( !EscapeCommFunction( handle, CLRDR ) ){
            CloseHandle( handle );
            handle = (void*)0xffffffff;
            return 3;
        }
        if( !CloseHandle( handle ) ) return 4;
        handle = (void*)(0xffffffff);
    }else{
        return 1;
    }

    return 0;
}

//-----
bool Write( unsigned char *Mesg, unsigned long Size )
{
    unsigned long Error;
    unsigned long Errors;
    unsigned long SizeWritten;
    COMSTAT Stat;

    if( handle==(void *)0xffffffff ){
        return false;
    }
    if( Size>BUFFERSIZE ){
        printf( " 發送數據太長      " );
        return false;
    }
}

```

```

do{
    ClearCommError( handle, &Errors, &Stat );
}while( BUFFERSIZE < Stat.cbOutQue + Size );

if( WriteFile( handle, Mesg, Size, &SizeWritten, 0 )==false ){
    if( GetLastError()==ERROR_IO_PENDING ){
        while( GetOverlappedResult( handle, 0, &SizeWritten, true ) ){
            Error = GetLastError();
            if( Error==ERROR_IO_INCOMPLETE ){
                continue;
            }else{
                ClearCommError( handle, &Errors, &Stat );
                break;
            }
        }
    }else{
        ClearCommError( handle, &Errors, &Stat );
    }
}

if( Size==SizeWritten ){
    return true;
}else{
    return false;
}
}

//-----
bool Read( unsigned char *Buffer, unsigned long SizeToRead,
           unsigned long *ReadSize, unsigned long *ErrFlag )
{
    BOOL          rt;
    unsigned char  RecvMesg[BUFFERSIZE];
    unsigned char  *cptr;
    unsigned long  i;
    unsigned long  stime;
    unsigned long  dtime;
    unsigned long  Error=0;
    unsigned long  Size;
    COMSTAT        Stat;

    *ReadSize = *ErrFlag = dtime = 0;

    if( handle==(void *)0xffffffff ){
        return false;
    }

    stime = GetTickCount();
    do{
        ClearCommError( handle, &Error, &Stat );
        if( Stat.cbInQue>0 ){
            ZeroMemory( RecvMesg, BUFFERSIZE );
            rt = ReadFile( handle, RecvMesg, Stat.cbInQue, &Size, 0 );
            RecvMesg[Stat.cbInQue] = 0x00;
            if( !rt ){
                // 清除錯誤標誌
                ClearCommError( handle, &Error, &Stat );
                // 設置引數錯誤標誌
                *ErrFlag = Error;
                return false;
            }
        }

        for( cptr=&RecvMesg[0],i=0 ; cptr<&RecvMesg[Stat.cbInQue] ; cptr++,i++){
            if( SizeToRead==0xffffffff ){
                if( *cptr==0x02 ){
                    *ReadSize = 0;
                    Buffer[(*ReadSize)++] = *cptr;
                }else{
                    if( (*cptr=='\n') ){
                        Buffer[(*ReadSize)++] = cptr[0];
                        if( Buffer[(*ReadSize)-2]=='\r' ){
                            Buffer[(*ReadSize)] = 0x00;
                            goto OutOfWhile;
                        }
                    }
                    Buffer[(*ReadSize)++] = *cptr;
                }
            }
        }
    }
}

```

```

        }else{
            Buffer[(*ReadSize)++] = *cptr;
            if( (*ReadSize)>=SizeToRead ){
                goto OutOfWhile;
            }
        }
    }
    Sleep(1);
    dtime = GetTickCount() - stime;
    }while( (SizeToRead>*ReadSize) && (dtime<TIMEOUT) );

OutOfWhile:

    if( ((SizeToRead>*ReadSize)&&(SizeToRead!=0xFFFFFFFF))||(*ReadSize==0) ){
        *ErrFlag = 0x00010000;
        return false;
    }
    return true;
}
//-----
void CPLSum( unsigned char *str, int len, unsigned char *buf )
{
    int          num=0;
    unsigned char *ch;

    for( ch=&str[0] ; ch<&str[len] ; ch++ ) num += *ch;
    num = ~(num&0x000000FF)&0x000000FF);
    sprintf( (char*)buf, "%02X", num );
}
//-----

bool AppCPL( char* tosend, char * received )
{
    unsigned char theMsg[BUFFERSIZE];
    unsigned char theApp[BUFFERSIZE];
    unsigned char theHdr[16];
    unsigned long SzToSnd;
    unsigned long RdSz;
    unsigned long ErrFlg;
    bool          rt;
    int           Cnt=0;
    unsigned char Sum[4];
    char          *etx;

    if( handle==(void*)(0xFFFFFFFF) ){
        return false;
    }

    ZeroMemory( theMsg, BUFFERSIZE );
    sprintf( (char*)theHdr, "%x02%x30%x31%x30%x30%x58" );
    sprintf( (char*)theMsg, "%s%s%x03", theHdr, tosend );

    if( CheckSum ){
        SzToSnd = strlen( (char*)theMsg );
        CPLSum( theMsg, SzToSnd, &theMsg[SzToSnd] );
    }
    strcat( (char*)theMsg, "%r%n" );
    SzToSnd = strlen( (char*)theMsg );

resend:
    Write( theMsg, SzToSnd );
    rt = Read( theMsg, 0xFFFFFFFF, &RdSz, &ErrFlg );

    if( !rt ){
        if( Cnt++ < COMRESENDNUM ){
            goto resend;
        }
        if( ErrFlg ){
            ErrorCode = ErrFlg;
        }else{
            ErrorCode = 0x000f0000;
        }
        return false;
    }

```

```

    }else{
        if( strcmp( (char*)theMsg, (char*)theHdr, 6 )!=0 ){
            ErrorCode = 0x00020000;
            if( Cnt++<COMRESENDNUM ){
                goto resend;
            }
            return false;
        }
        if( CheckSum ){
            CPLSum( theMsg, RdSz-4, Sum );
            if( (theMsg[RdSz-4]!=Sum[0])||(theMsg[RdSz-3]!=Sum[1]) ){
                ErrorCode = 0x00040000;
                if( Cnt++<COMRESENDNUM ){
                    goto resend;
                }
                return false;
            }
        }
        ZeroMemory( theApp,BUFFERSIZE );
        CopyMemory( theApp, &theMsg[6], RdSz-6 );
        etx = strstr( (char*)theApp, "%x03" );

        if( etx==NULL ){
            ErrorCode = 0x00200000;
            if( Cnt++<COMRESENDNUM ){
                goto resend;
            }
            return false;
        }else{
            *etx = 0x00;
        }
        strcpy( received, (char*)&theApp[0] );
    }
    return true;
}
//-----

```

第 7 章 故障處理

■ 不能通訊時的確認項目

- ① 裝置電源是否合上。
- ② 接綫有無錯誤。
- ③ 主計算機側與本機側的通訊條件設定是否一致。

下記設定項目只要有一個不同, 就不能通訊。

下劃綫部分表示在本機側可設定。

傳送速度	:	<u>38400</u> 、 <u>19200</u> 、 <u>9600</u> 、 <u>4800</u> 、 <u>2400</u> bps
數據長	:	<u>8</u> 、7位
校驗	:	<u>無奇偶校驗</u> 、 <u>偶數校驗</u> 、奇數校驗
停止位	:	<u>1</u> 、 <u>2</u> 停止位

- ④ 主計算機發送的命令楨的機器地址與本機設定的地址是否一致。

出廠時, 本機的地址設為0, 即使命令楨的機器地址設定為00(30H、30H), 本機也不應答。

- ⑤ 多分支連接的機器, 是否設定了不同的機器地址。
- ⑥ 通訊時刻與

 4-7 命令電文、應答電文時間規格 (4-14頁)
是否符合。

- ⑦ 設備區分代碼(本機の場合, “X” 或 “x”)以外的字符串代碼, 請使用大寫字母。

第 8 章 規 格

名 稱	備 注
傳送形態	平衡形
傳送路	3綫式
傳送速度 (bps)	38400、19200、9600、4800、2400
傳送距離	500m以下 (但與MA500DIM、CMC410連接時為300m)
通訊方式	半雙工
同期方式	調步同期
數據形式	數據8位 1停止位 偶數校驗 數據8位 2停止位 無奇偶校驗
錯誤檢測	校驗檢查、校驗和
機器地址	0~127 (0時, 通訊功能無效)
連接形式	1:N (31台以下)
其他	基于RS-485

附錄

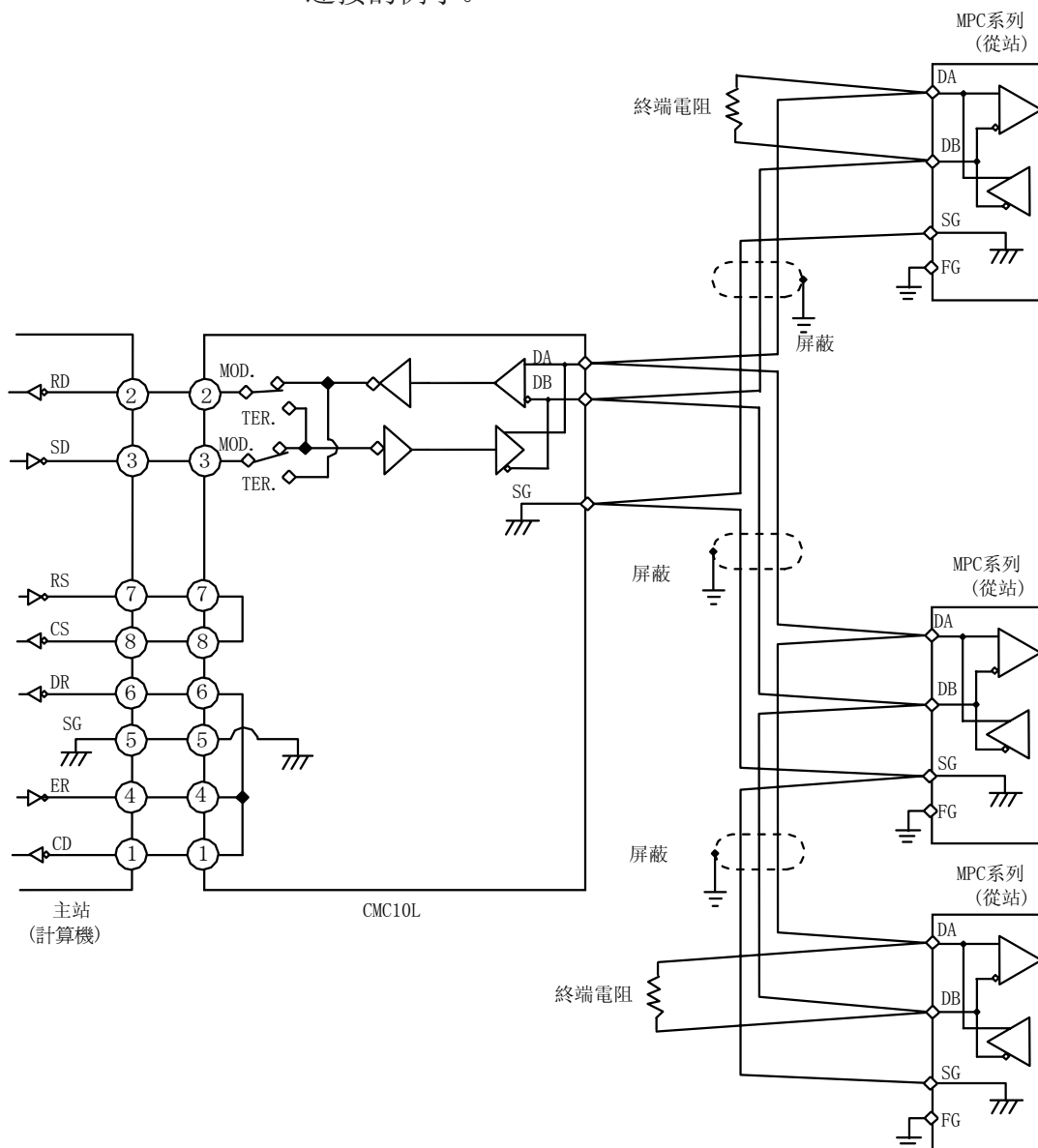
■ 代碼表

上位 下位	0	1	2	3	4	5	6	7
0			空格	0	@	P	、	p
1			!	1	A	Q	a	q
2	STX		"	2	B	R	b	r
3	ETX		#	3	C	S	c	s
4			\$	4	D	T	d	t
5			%	5	E	U	e	u
6			&	6	F	V	f	v
7			'	7	G	W	g	w
8			(	8	H	X	h	x
9			)	9	I	Y	i	y
A	LF		*	:	J	Z	j	z
B			+	;	K	[	k	{
C			,	<	L	¥	l	!
D	CR		—	=	M	]	m	}
E			.	>	N	^	n	~
F			/	?	O	_	o	

該通訊系統中不使用 部分
(根據機器, 使用的代碼不同)

■ 與CMC10L的连接

下圖是使用直通電纜D-SUB (9針) 與終端模式的主站 (計算機) 連接的例子。



通訊線路的兩端處，請安裝 $150\ \Omega \pm 5\%$ 的1/2W以上的終端電阻。
屏蔽的FG接地，不是在屏蔽層的两端，而是單側的1點接地。
主站的變換器，可使用本公司生產的CMC10L001A000。

⚠ 使用上的注意事項

請務必連接SG。如果不連接，將不能獲得穩定的通訊。

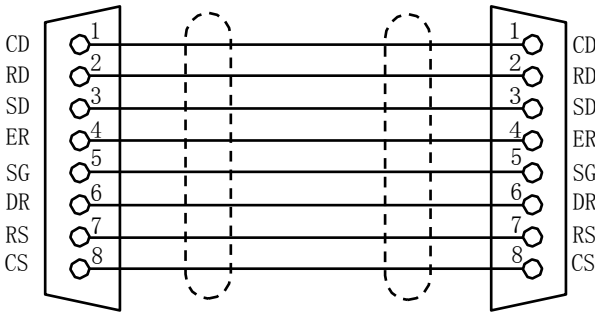
請把主站的SD與從站的RD、主站的RD與從站的SD連接。

為此, 主計算機(主站)側 RS-232C插口的針排列(調制解調器/終端)要與使用的電纜種類(交叉/直通)匹配, 并按下表所示, 對CMC10L內部的MODE開關進行設定。

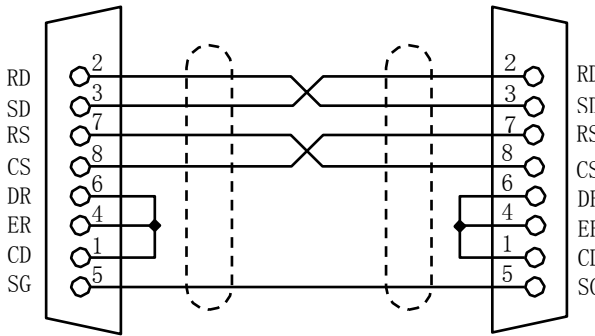
RS-232C	電纜種類	MODE開關
TERMINAL	直通	MODEM
TERMINAL	交叉	TERMINAL
MODEM	直通	TERMINAL
MODEM	交叉	MODEM

● RS-232C電纜

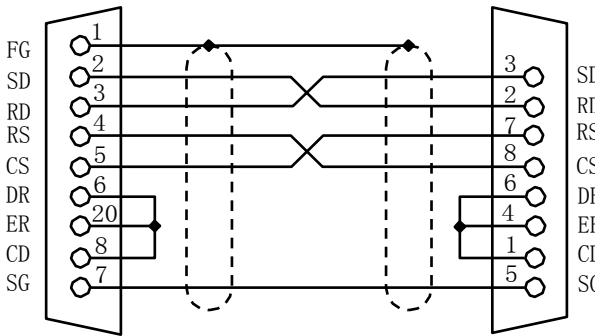
直通:如②-②、③-③那樣,把相同編號的針連接的兩側帶D-SUB (9針) 插口的RS-232C電纜。



交叉:如②-③、③-②那樣,各信號與不同的針號連接的兩側帶D-SUB (9針) 插口的RS-232C電纜。



D-SUB25針-9針轉換電纜:如圖所示,把D-SUB (25針) 轉換成D-SUB (9針) 的RS-232C電纜。





本資料所記内容如有變更恕不另行通知
