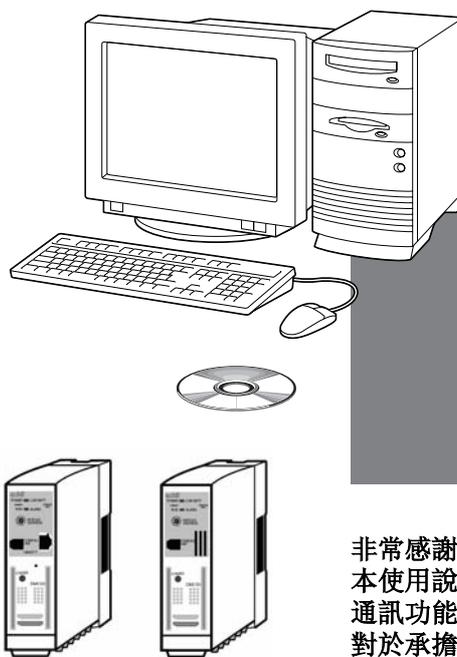


模件型調節器 DMC50 使用說明書 通訊連接篇



非常感謝您購買模件型調節器DMC50。
本使用說明書中記述了正確使用DMC50
通訊功能的必要事項。
對於承擔使用DMC50的通訊功能的操作
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，請在失效安全設計，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
- 宇宙航天設備
- 原子能設備等

請勿把本產品用在與人身安全直接相關的用途上。

要求

請確保把本使用說明書送到本產品使用者手中。

禁止擅自複印和轉載全部或部分本使用說明書的內容。
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

©2000 Yamatake Corporation ALL RIGHTS RESERVED

DIGITRONIC ©是株式會社 山武的註冊商標。
Ethernet © 是富士Xerox株式會社的註冊商標。
ISaGRAF © 是ICS Triplex ISaGRAF公司的註冊商標。
其它記載的公司名及產品是個公司的註冊商標。

安全上的注意事項

■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，防患於未然，以免給您及他人造成人體損害及財產損失，請務必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

	<p>△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。</p> <p>圖中有具體注意內容(左圖表示小心觸電)。</p>
	<p>⊘記號：為了避免危險發生，禁止某些特定行為時使用的符號。</p> <p>在圖中或在其附近注明具體禁止事項(左圖表示禁止分解)。</p>
	<p>●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。</p> <p>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)。</p>

警告

	本機在安裝、拆除及配線作業時，務必在切斷供給電源後進行。 否則有觸電的危險。
	請勿分解本機。 否則有觸電、發生故障的危險。
	FG端子切實實施接地電阻 100 Ω 以下接地後，再進行測量對象或外部控制回路的連接。 否則有觸電、發生火災的危險。
	請勿觸摸電源端子等帶電部件。 否則有觸電的危險。
	請在DMC50的外部採取安全措施，即使由於DMC50的故障或DMC50的外部原因發生異常的場合，系統全體也能安全運行。 否則由於異常動作，會造成重大安全事故。

注意

	請在規格書中記載的使用條件（溫度、濕度、電壓、振動、衝擊、安裝方向、環境等）範圍內使用本機。 否則有發生火災、故障的危險。
	請勿堵塞本機的通風孔。 否則有發生火災、故障的危險。
	請勿讓斷線頭、鐵粉、水等進入機箱內。 否則有發生火災、故障的危險。
	請按照本機連線的標準、指定電源幾施工方法，正確配線。 否則有發生火災、故障的危險。
	去本機的電流電壓輸入端子(2·4、6·8、10·12、14·16)的輸入，請控制在規格記載的電流・電壓範圍內。 否則有發生火災、故障的危險。
	請按規格書中記載的扭矩擰緊端子螺釘，端子螺釘沒有擰緊時有觸電、發生火災的危險。
	請勿把本機中未使用的端子作為中繼端子使用。 否則有觸電，發生火災、故障的危險。
	本機接線後請蓋上端子蓋。 否則有觸電的危險。

本使用說明書的定位

與DMC50相關的使用說明書共有7冊。請根據用途閱讀必要的使用說明書。
如果您手中無相關的使用說明書，請向本公司或代理店索取。

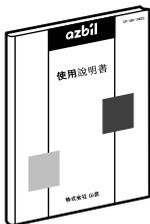


模件型調節器 DMC50

硬件篇

資料編號 CP-UM-5172C

在使用DMC50進行裝置的硬件設計或製造前請務必閱讀。
含控制模件及通訊模件兩種說明。
對本機的安裝、接線、規格、硬件的故障處理進行說明。

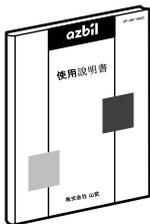


模件型調節器 DMC50

入門篇

資料編號 CP-SP-1092C

請初次使用DMC50的人員務必閱讀。
舉例對本機的動作的概要、基本使用方法進行說明。
請在使用智能編程軟件包的同時閱讀本書

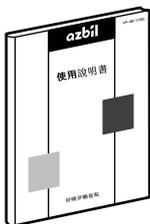


模件型調節器 DMC50

通訊連接篇

資料編號 CP-SP-1093C

本書。
請使用DMC50通訊功能的人員務必閱讀。
對本機的通訊功能進行說明。是對本機的CPL通訊、以態通訊等的說明。



模件型調節器 DMC50/AHC2001 功能塊篇

資料編號 CP-SP-1130C

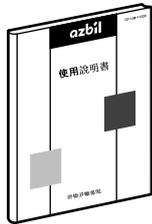
使用DMC50或AHC2001對客戶的應用實現最佳化控制的場合，請務必閱讀。
對可實現任意控制方式的ISaGRAF進行說明。



模件型調節器 DMC50/AHC2001 應用篇

資料編號 CP-SP-1134C

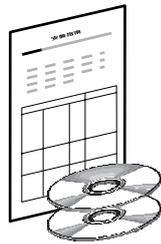
對DMC50或AHC2001的實際編程方法進行說明。
請編程人員務必閱讀。
對程序段功能塊的詳細內容、ISaGRAF的活用、實例的應用進行說明。
本書是在閱讀并理解了另冊的「入門篇」「功能塊篇」為前提，進行說明的。



**模件型調節器 DMC50/AHC2001 用 智能編程軟件包
SLP-D50/SLP-H21**

資料編號 CP-SP-1122C

對DMC50/AHC2001用軟件包SLP-D50/SLP-H21安裝到計算機的方法、計算機的操作及各功能、設定方法進行說明。



安裝指南 (SLP-D50J50)

資料編號 CP-UM-5259C

是DMC50用智能編程軟件包SLP-D50安裝到計算機的方法的說明書。

本使用說明書的構成

本使用說明書構成如下。

第1章 概要

對本機通訊的特長、使用本機的通訊端口構成通訊的概要進行說明。

第2章 連接及通訊

對本機在通訊時的連接方法及設定、本機具備的通訊方式進行說明。
另外，對各通訊連接的具體方法及智能編程編程軟件包SLP-D50/SLP-H21的操作進行說明。

第3章 以態通訊

對本機在以態通訊時的規格及例子程序進行說明。

第4章 CPL命令參考

本機用本公司的CPL通訊進行通訊時的通訊格式、通訊異常時的處理方法、各命令的詳細內容的說明。

第5章 與EST-Z系列的連接

本機與本公司的智能終端EST-Z系列連接時的方法的說明。

第6章 數據地址

CPL通訊時的數據地址的說明。

目 錄

安全上的注意事項	
請確認	
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

■ 特 長	1-1
■ 通訊端口	1-2
■ 連接端口及網絡構成	1-3
■ 通訊功能的概要	1-5

第2 章 連接及通訊

2-1 電纜的連接	2-1
2-2 通訊的設定	2-4
■ RS-485端口的通訊設定	2-4
■ 顯示器通訊端口的通訊設定	2-4
■ 編程器通訊端口的通訊設定	2-4
2-3 具體的連接方法	2-5
■ 用編程器通訊端口的編程器通訊(直接連接)	2-6
■ 用編程器通訊端口的編程器通訊(間接連接)	2-8
■ 用RS-485端口的編程器通訊	2-10
■ 用以態方式的編程器通訊	2-13
■ 用顯示器通訊端口的CPL通訊	2-17
■ 用RS-485端口的CPL通訊	2-19
■ 用以態方式的CPL通訊	2-21
2-4 時間規格	2-24
■ 命令電文、應答電文時間規格	2-24
■ RS-485驅動控制時間規格	2-24

第3 章 以態通訊

3-1 TCP/IP堆棧規格	3-1
■ IP端口與連接	3-1
3-2 客戶應用	3-2
■ 錯誤處理	3-2
■ 例子程序	3-4
3-3 用語說明	3-15

第4 章 CPL命令參考

■ 概 要	4-1
■ 基本楨格式	4-1
■ 異常處理	4-2
■ CPL地址	4-4
■ 命令詳細內容	4-4
■ 16位系命令	4-4
■ RG命令(帶數量指定的固定長連續數據讀出(32位系))	4-5
■ WG命令(帶數量指定的固定長連續數據讀出(32位系))	4-6
■ RN命令(帶重複及數量指定的固定長隨機讀出(32位系))	4-7
■ WN命令(帶重複及數量指定的固定長隨機寫入(32位系))	4-8
■ RD命令(固定長數據讀出(16位系))	4-9
■ WD命令(固定長數據寫入(16位系))	4-10
■ RU命令(帶重複的固定長隨機讀出(16位系))	4-11
■ WU命令(帶重複的固定長隨機寫入(16位系))	4-12
■ RS命令(ASCII10進制連續數據讀出(16位系))	4-13
■ WS命令(ASCII10進制連續數據寫入(16位系))	4-14

第5 章 與EST-Z系列的連接

■ 整數轉換嚮導	5-1
■ 自動運行開始時間	5-1

第6 章 數據地址

6-1 數據地址的構成	6-1
■ 網絡地址(NA)與參數地址(PA)	6-1
■ 網絡地址(NA)的構成	6-2
■ 參數地址(PA)的構成	6-2
6-2 數據地址一覽(NA區域)	6-3
6-3 數據地址一覽(PA區域)	6-4

附錄

■ ISaGRAF變量的數據型	附-1
■ 參數的數據型	附-1
■ 浮點小數格式: IEEE754格式	附-2
■ 16位系命令的動作	附-3

本使用說明書的標記

本使用說明書的標記如下。

-  **使用上的注意事項**：表示使用上敬請注意的事項。
-  **參考**：表示知道後便於理解的事項。
- ：表示參照的使用說明書及項目。
- ①②③：表示操作的步驟或圖等說明對應的部分。
- COM模件：表示通訊模件 (ME20X、MR20X)。
- CTRL模件：表示控制模件 (CS40X、CS20X、CH40X、CH20X)。
- 編程器：表示智能編程軟件包SLP-D50/SLP-H21。
- [OK]按鈕：表示計算機畫面的選擇按鈕。
- [通訊路由]：表示計算機畫面的信息及菜單。
- 》：表操作的結果、計算機或機器的顯示內容、操作後的機器的狀態。
- [F4]鍵：表示鍵盤的鍵。
- TCP/IP設定窗口：表示計算機畫面上顯示的窗口的名稱。
- 以態：表示Ethernet[®]。

第 1 章 概 要

通過使用通訊模件(以下稱為COM模件)，DMC50可由RS-485、以態方式連接，實現遠程監視。

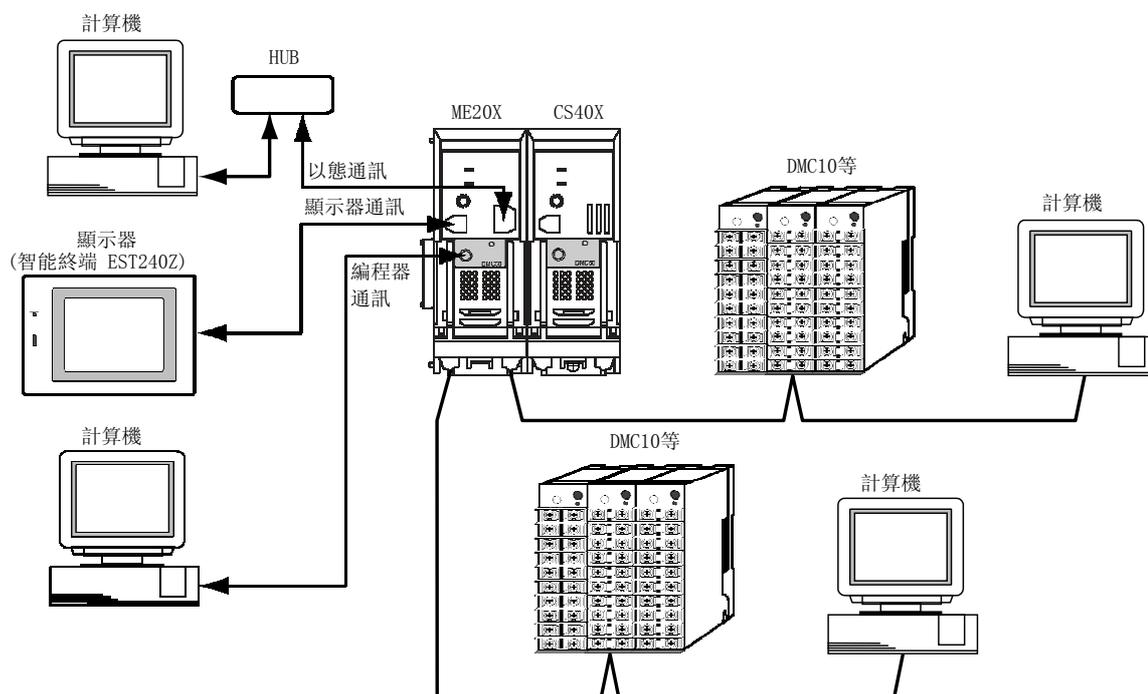
■ 特 長

DMC50的通訊功能有如下特長。

● 可採用多種連接方法

備有RS-485端口(ME20X有2個端口、MR20X有1個端口)、顯示器通訊端口、編程器通訊端口、以態端口(ME20X)。

所以可採用各種方法/路由，能與多個上位機器連接。



● 基於標準以態網絡

使用ME20X可與市售的以態機器構成系統。另外，在已有的以態網上可連接DMC50，節省了資源。

● 採用標準通訊協議TCP/IP

使用市售的開發用軟件，編制應用程序，可方便地與通用計算機或工作站進行通訊。

● 全部模件上都裝有顯示器通訊端口

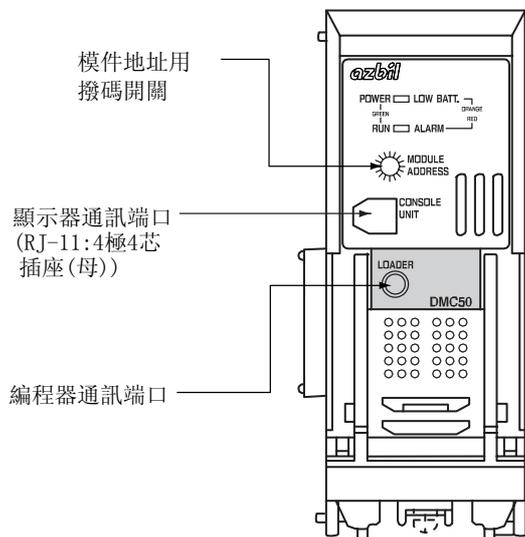
所有的控制模件(以下稱為CTRL模件)上都帶有顯示器通訊端口。所以，可不用COM模件就能與顯示器進行1對1的連接。

● 基於CPL協議

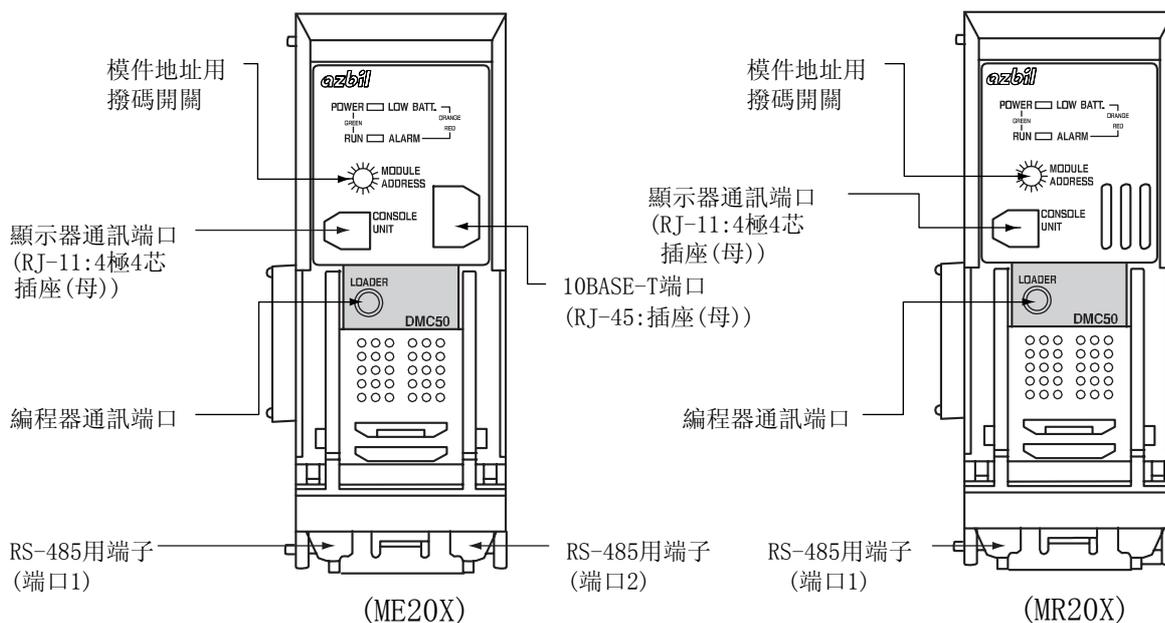
基於CPL通訊(Controller Peripheral Link:本公司上位通訊協議)標準，顯示器可方便地訪問。

■ 通訊端口

DMC50上實裝的通訊端口的位位置及名稱如下。



CTRL模件



COM模件

❗ 使用上的注意事項

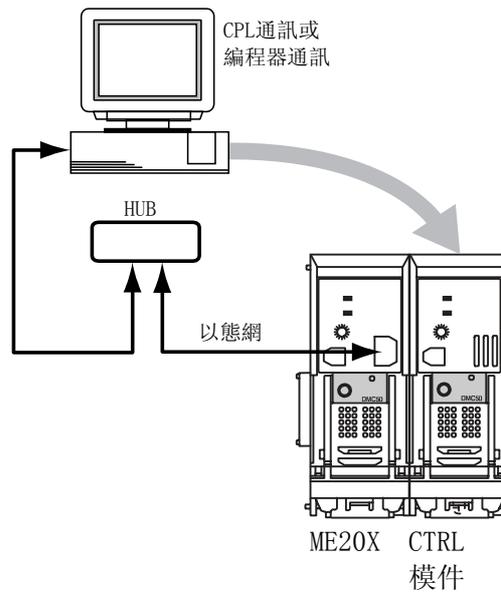
編程器通訊端口與顯示器通訊端口不可同時使用。
 編程器通訊用的插口與顯示器通訊用的接頭同時連接時，會不能正常通訊的情況。

■ 連接端口及網絡構成

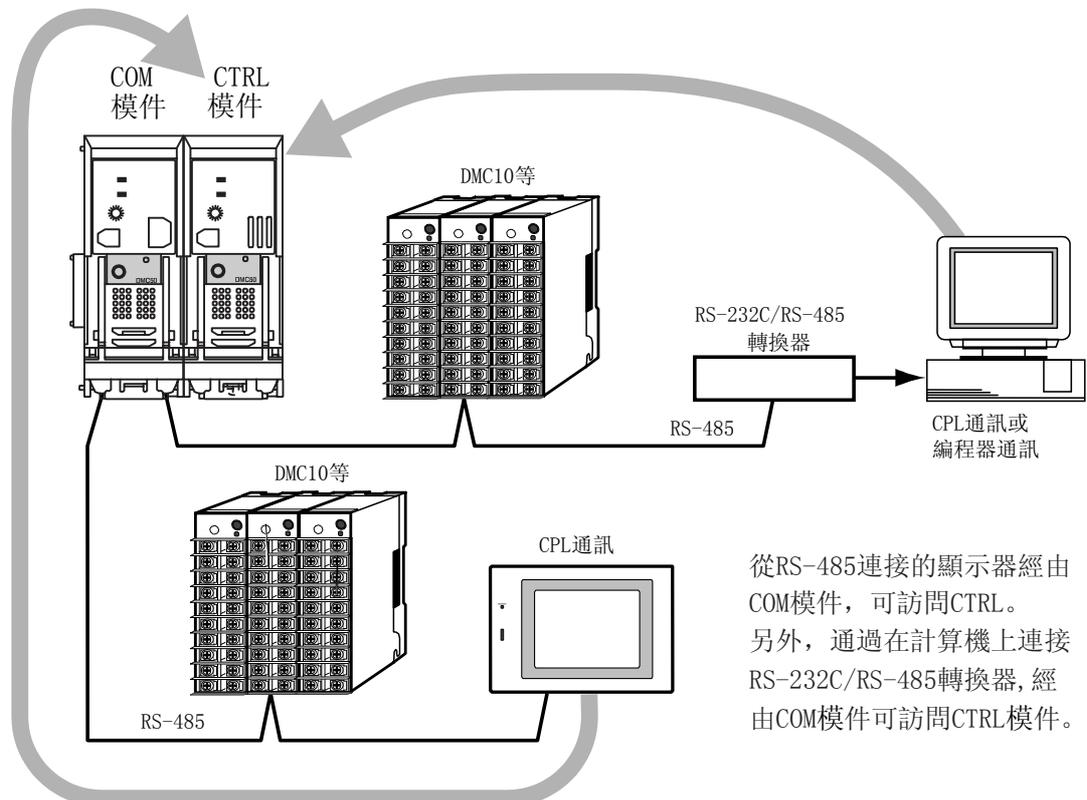
用DMC50可構成的網絡構成例如下。

● 以態連接

從以態網連接的工作站、計算機經由ME20X，可訪問CTRL模件



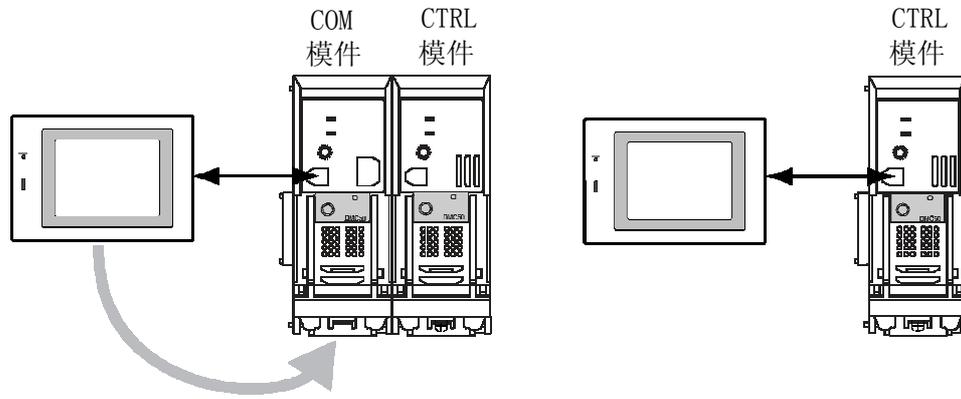
● RS-485連接



從RS-485連接的顯示器經由COM模件，可訪問CTRL。另外，通過在計算機上連接RS-232C/RS-485轉換器，經由COM模件可訪問CTRL模件。

● 顯示器通訊端口連接

從顯示器通訊端口連接的顯示器經由COM模件可訪問CTRL模件。另外，CTRL模件上備有顯示器通訊端口，可直接訪問CTRL模件。

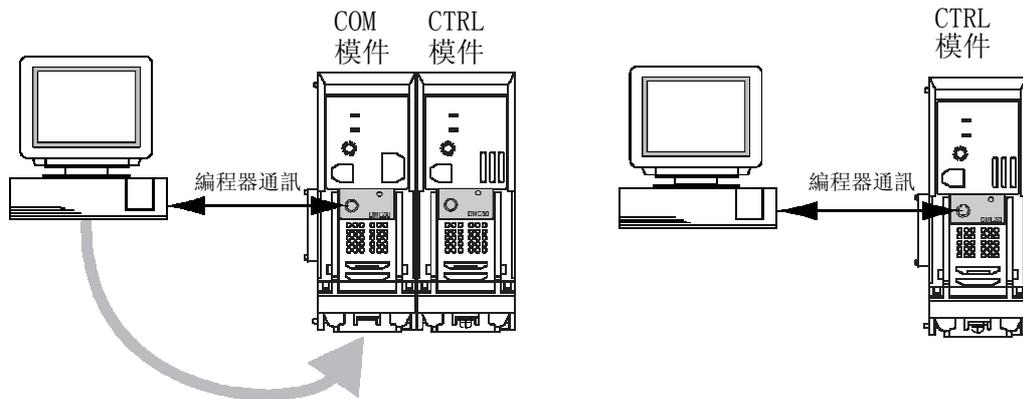


❗ 使用上的注意事項

顯示器通訊端口不對應多路分支。請勿把RJ-11接頭製作成多分支結構。否則，有不能正常通訊的可能。

● 編程器通訊端口連接

從編程器通訊端口連接的編程器經由COM模件，可訪問CTRL模件。另外，CTRL模件上備有顯示器通訊端口，可直接訪問CTRL模件。



❗ 使用上的注意事項

用間接連接進行通訊的場合，編程器通訊或CPL通訊同時多個執行時，由於內部的處理負荷的原因，可能會有超時的情況發生。在用編程器監視中發生超時的場合，請直接與儀錶的編程器通訊端口連接，停止其它通訊端口的通訊。

■ 通訊功能的概要

DMC50的通訊功能如下。

- 編程器通訊
- CPL通訊
- 時鐘同步功能

● 編程器通訊

編程器通訊採用編程器通訊專用協議，DMC50與編程器連接。編程器通訊可經由各儀錶的編程器通訊端口、COM模件的RS-485、以態接口連接。

● CPL通訊

用於計算機或CPL協議對應的顯示器與DMC50連接時使用。CPL通訊可經由各儀錶的顯示器通訊端口、COM模件的RS-485、以態接口連接。

● 時鐘同步功能

用CPL通訊或編程器通訊變更COM模件的日曆/時間設定後，底板連接的CTRL模件的時鐘可與COM模件的時鐘同步。

❗ 使用上的注意事項

用時鐘同步功能進行時鐘數據設定的場合，各模件間的時鐘數據的精度為±1秒。

● 端口與協議

通訊端口與協議的關係如下。

端口名	協議
編程器通訊端口	編程器通訊
顯示器通訊端口	CPL通訊
RS-485端口1	CPL/編程器通訊切換
RS-485端口2	CPL/編程器通訊切換
以態IP端口1	編程器通訊
以態IP端口2	編程器通訊
以態IP端口3	CPL通訊
以態IP端口4	CPL通訊

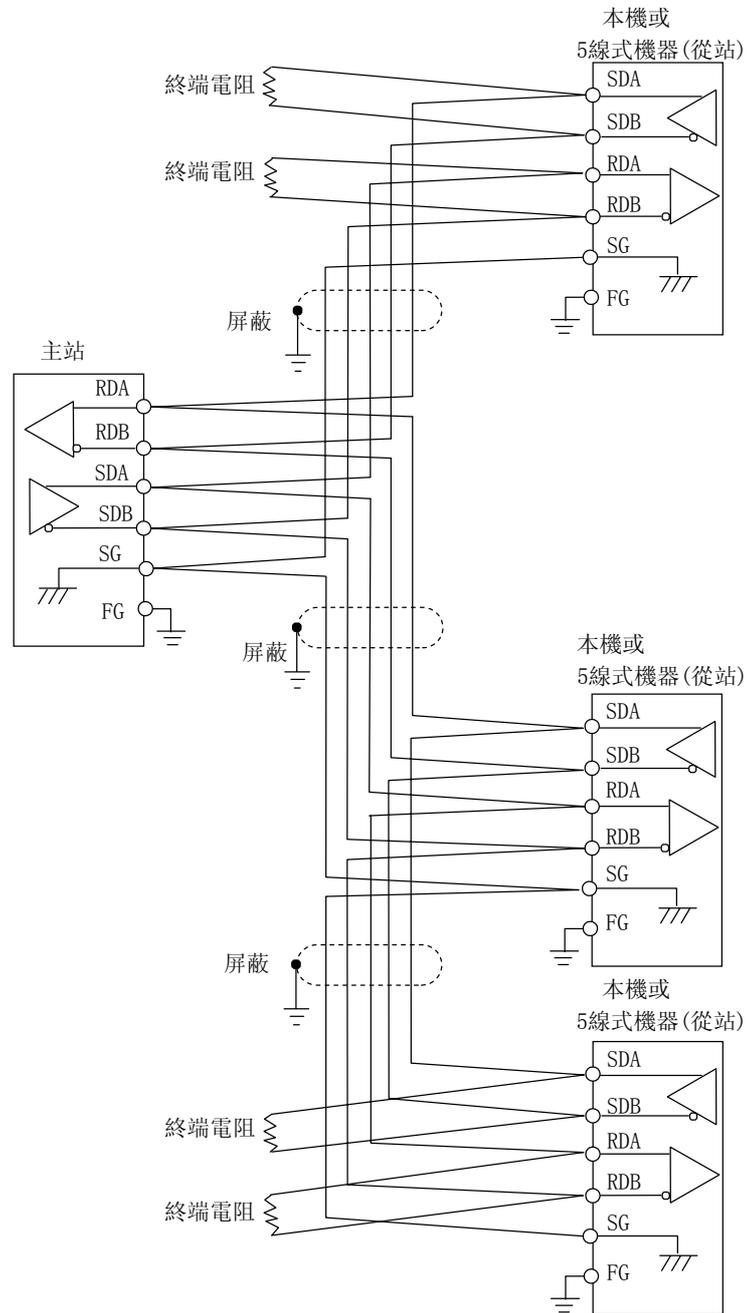
第 2 章 連接與通訊

2 - 1 電纜的連接

請按以下方法進行電纜的連接。

● 5線式RS-485連接的場合

請按下圖所示方法與5線式機器連接。



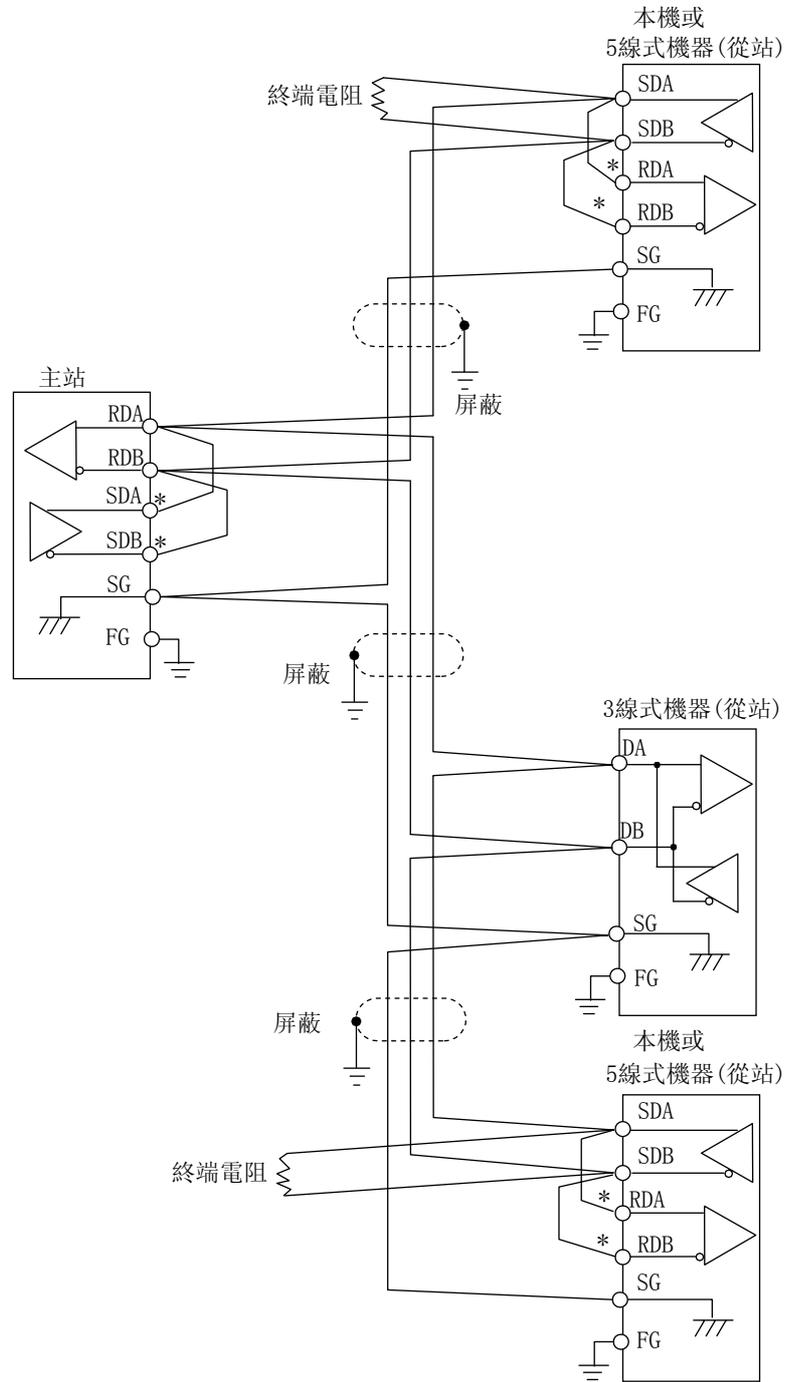
請在通訊路的兩端安裝 $150\ \Omega \pm 5\%$ 、 $1/2W$ 以上的終端電阻。
屏蔽的FG接地不是屏蔽兩端，請單側1處接地。

❗ 使用上的注意事項

請務必連接SG。如果不連接。通訊會不穩定。

● 3線式RS-485連接の場合

請按下圖所示與3線式機器連接。



請在通訊路的兩端安裝 $150\ \Omega \pm 5\%$ 、 $1/2\text{W}$ 以上的終端電阻。

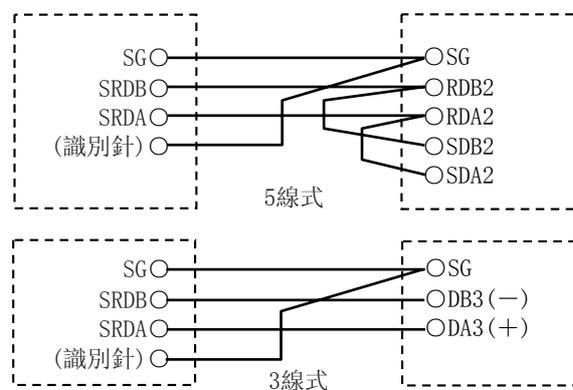
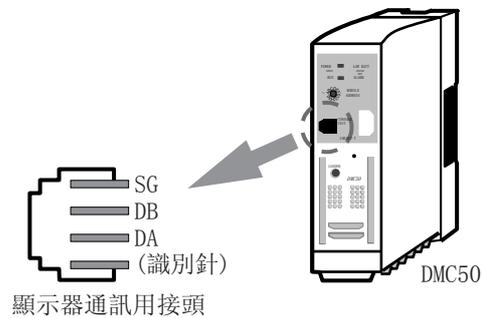
屏蔽的FG接地不是屏蔽兩端，請單側1處接地。

* 符號的接線請在外部進行。

❗ 使用上的注意事項

請務必連接SG。如果不連接。通訊會不穩定。

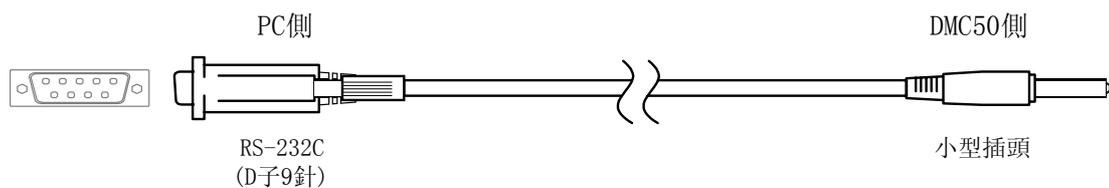
● 顯示器通訊端口的場合



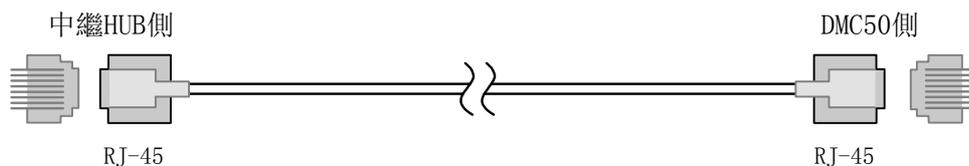
❗ 使用上的注意事項

請務必把識別針連接到SG上。

● 編程器通訊端口的場合(專用電纜: 81440793-001)



● 以態端口的場合(10BASE-T)



❗ 使用上的注意事項

如果經由100BASE-T/10BASE-T的自適應HUB連接，有產生連接故障的情況。請使用中繼HUB，不使用100BASE-T/10BASE-T自適應型的HUB。請勿用10BASE-T交叉電纜直接進行機器間的連接。有可能不能識別。

2 - 2 通訊的設定

■ RS-485端口的通訊設定

RS-485端口1、2的通訊設定如下。

傳送速度	9600bps、19200bps、38400bps
校驗	偶數校驗
停止位長	1停止位
數據長	8位
流量控制	XON控制/無S參數

■ 顯示器通訊端口的通訊設定

顯示器通訊端口的通訊設定如下。

傳送速度	9600bps、19200bps、38400bps
校驗	偶數校驗
停止位長	1停止位
數據長	8位
流量控制	XON控制/無S參數

■ 編程器通訊端口的通訊設定

編程器通訊端口的通訊設定如下。編程器通訊端口的設定由編程器側自動進行。

傳送速度	9600bps
校驗	偶數校驗
停止位長	1停止位
數據長	8位
流量控制	XON控制/無S參數

2 - 3 具體的連接方法

DMC50的連接有如下方式。

- 用編程器通訊端口的編程器通訊
- 用RS-485端口的編程器通訊
- 用以態端口的編程器通訊
- 用顯示器通訊端口的CPL通訊
- 用RS-485端口的CPL通訊
- 用以態端口的CPL通訊

連接方法有直接連接與間接連接。

間接連接是指，經由COM模件的通訊端口與已連接的CTRL模件連接。CTRL模件單體也支持直接連接。

參考

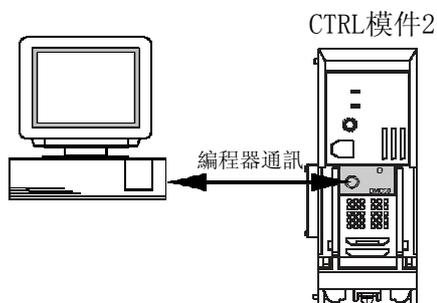
變更各通訊端口的設定時，到恢復原來狀態前有不能連接的情況。

但由於編程器通訊端口沒有設定項目，只要模件地址符合就能連接。不可連接時，請用編程器通訊端口連接。

■ 用編程器通訊端口進行編程器通訊(直接連接)

對從要連接的儀錶的編程器通訊端口連接的方法進行說明。

例：連接模件地址為2的CTRL模件。

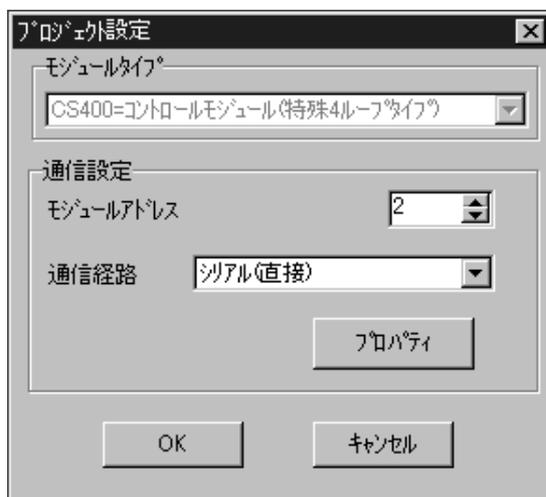


● 設置

用專用的編程器電纜把計算機的串口與CTRL模件的編程器通訊端口連接(型號:81440793-001)。

● 儀錶的連接

- ① 啟動編程器。
- ② 打開要連接的CTRL模件用的項目。
項目為製作的場合，請進行[項目的新規製作]。
- ③ 從菜單的[選項]打開[項目設定]。
》項目設定窗口打開。



- ④ 請把項目設定窗口的[模件地址]設定成與要連接的CTRL模件用的撥碼開關相同的值。
- ⑤ 把[通訊路由]設為[串行(直接)]。

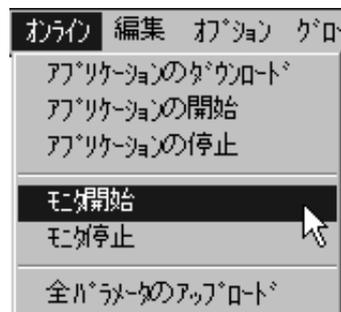
- ⑥ 按[屬性]按鈕。
 》 串行通訊設定窗口打開。



参考

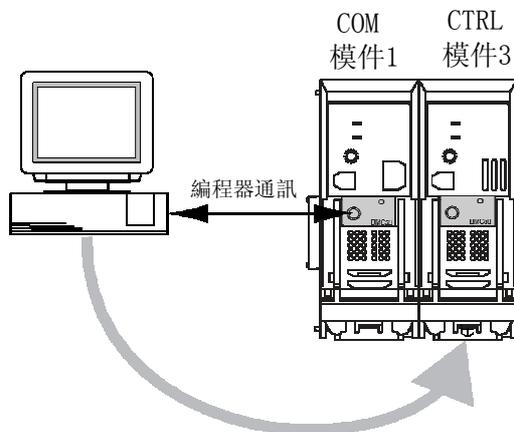
編程器通訊端口連接の場合、傳送速度固定為9600bps。

- ⑦ 請把[COM端口]的名稱設為與用編程器電纜連接的計算機的串口相同的名稱，按[OK]按鈕。
 》 串行通訊設定窗口關閉。
- ⑧ 連接「模件地址2」的CTRL模件 (CS400)。
- ⑨ 請按項目設定窗口的[OK]按鈕。
- ⑩ 讀出CTRL模件的信息時，可按[選項目] → [監視開始]選擇。



■ 用編程器通訊端口進行編程器通訊(間接連接)

對經由COM模件的編程器通訊端口與CTRL模件連接的方法進行說明。
例：經由COM模件(模件地址1)，與模件地址3的CTRL模件連接。

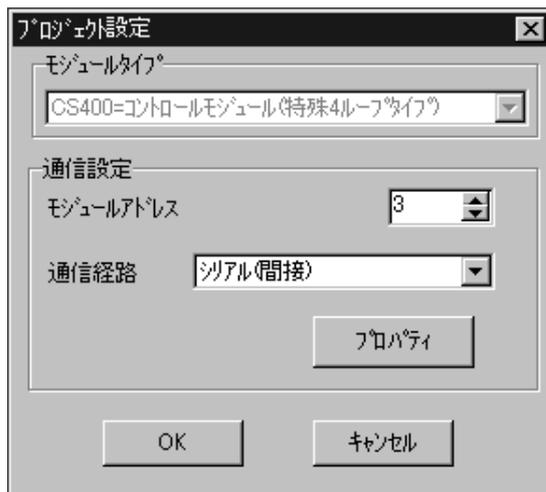


● 設置

- ① 用連接用接頭連接COM模件與CTRL模件。
- ② 用編程器電纜連接計算機的串口與COM模件的編程器通訊端口。

● CTRL模件的連接

- ① 打開要連接的CTRL模件用的項目。
未製作項目的場合，請執行[項目的新規製作]。
- ② 從菜單的[選項]打開[項目設定]。
》項目設定窗口打開。



- ③ 請把項目設定窗口的[模件地址]設定成與要連接的CTRL模件的撥碼開關相同的值。
- ④ 請把[通訊路由]設為[串行(間接)]。

- ⑤ 請按[屬性]按鈕。
》串行通訊設定窗口打開。

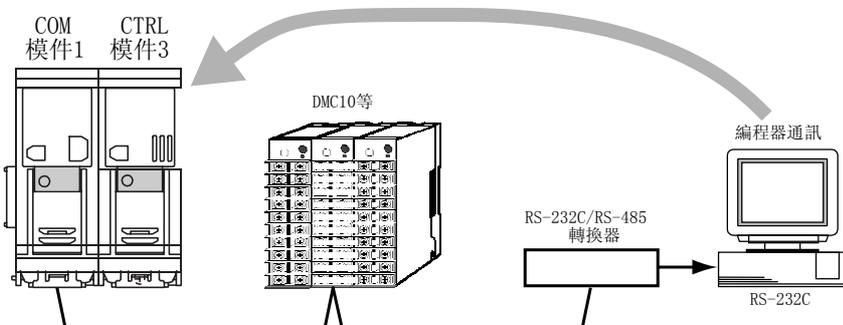


- ⑥ 請把[通訊模件地址]設為與要經由的COM模件的撥碼開關相同的值。
- ⑦ 請把[COM端口]設為與用編程器電纜連接的計算機串口相同的名稱，按[OK]按鈕。
》串行通訊設定窗口打開。
- ⑧ 請按項目設定窗口的[OK]按鈕。
- ⑨ 讀出CTRL模件的信息時，請按[在線] → [監視開始]選擇。

■ 用RS-485端口進行編程器通訊

對經由COM模件的RS-485端口與CTRL模件連接的方法進行說明。

例：經由COM模件(模件地址1)連接到模件地址為3的CTRL模件(CS40X)上。



● 設置

- ① COM模件與CTRL模件由底座單元接口連接。
- ② 計算機的串口與RS-232C/RS-485轉換器由RS-232C交叉電纜連接。
- ③ RS-232C/RS-485轉換器(本公司產CMC10L001A000或相當品)上連接RS-232C交叉電纜的另一端。
- ④ 設定RS-232C/RS-485轉換器，用RS-485側的電纜連接COM模件的RS-485端口。轉換器側的通訊設定如下。

傳送速度	9600bps
校驗	偶數校驗
停止位長	1停止位
數據長	8位
流量控制	XON控制/無S參數

● 經由COM模件 RS-485端口的設定

- ① 啟動編程器。
- ② 打開COM模件用的項目。
未製作項目的場合，請執行[項目的新規作成]。
- ③ 按菜單的[選項] → 打開[項目設定]。
》項目設定窗口打開。



- ④ 請把項目設定窗口的[模件地址]設為與要連接的COM模件用撥碼開關相同的值。
- ⑤ 請把[通訊路由]設為[串行(直接)]。
- ⑥ 請按[屬性]按鈕。
》串行通訊設定窗口打開。



- ⑦ 請把[COM端口]設為與用編程器電纜連接的計算機的串口相同的名稱。
》串行通訊設定窗口關閉。
- ⑧ 請按項目設定窗口的[OK]按鈕。
- ⑨ 從[在線] → [監視開始]置為在線狀態，變更連接的RS-485的設定。請變更使用的端口的協議。

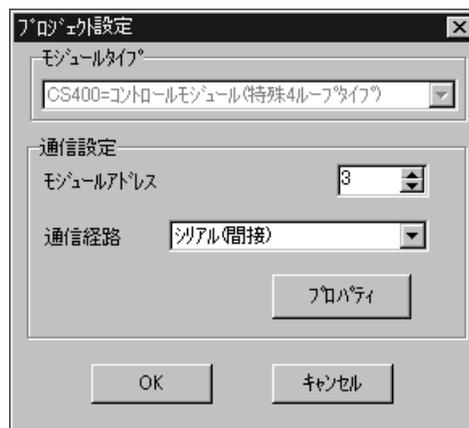
下圖是變更RS-485端口1。

インスタンスヘッダ		インスタンスボディ	
			1
1	伝送速度(RS-485ポート1)	bps	9600
2	プロトコル(RS-485ポート1)	--	2=ロータリ通信
3	伝送速度(RS-485ポート2)	bps	0=なし 1=CPL
4	プロトコル(RS-485ポート2)	--	2=ロータリ通信
5	IPアドレス	IPA	192.168.1.1
6	サブネットマスク	IPA	255.255.255.0
7	デフォルトルーター	IPA	0.0.0.0
8	KeepAliveTime	--	7200
9	IPポート1	--	1250
10	プロトコル(IPポート1)	--	2
11	IPポート2	--	1251
12	プロトコル(IPポート2)	--	2
13	IPポート3	--	1252
14	プロトコル(IPポート3)	--	1
15	IPポート4	--	1253
16	プロトコル(IPポート4)	--	1

- ⑩ 要使設定生效，請把COM模件的電源重新投入。

● 與CTRL模件的連接

- ① 打開要連接的CTRL模件用的項目。
未製作項目的場合，請執行[項目的新規作成]。
- ② 按菜單的[選項] → 打開[項目設定]。
》項目設定窗口打開。



- ③ 請把項目設定窗口的[模件地址]設為與要連接的CTRL模件用撥碼開關相同的值。
- ④ 請把[通訊路由]設為[串行(間接)]。
- ⑤ 請按[屬性]按鈕。
》串行通訊設定窗口打開。

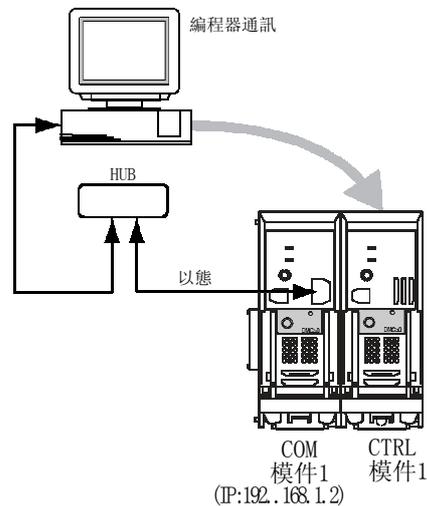


- ⑥ 請把[通訊模件地址]設為與要經由的COM模件撥碼開關相同的值。
請把[COM端口]設為與用編程器電纜連接的計算機串口相同的名稱，按[OK]按鈕。
》串行通訊設定窗口關閉。
- ⑦ 請按項目設定窗口的[OK]按鈕。
- ⑧ 讀出CTRL模件信息時，請按[在線] → [監視開始]選擇。

■ 用以態方式進行編程器通訊

對經由COM模件的以態端口與CTRL模件連接的方法進行說明。

例：經由COM模件(模件地址1、IP地址 192.168.1.2)連接模件地址為1的CTRL模件。



● 設置

- ① 請設定計算機的TCP/IP。根據環境，需要設定缺省的路由。詳細情況請與使用環境的網絡管理者詢問。
- ② 10BASE-T電纜經由中繼HUB與計算機及COM模件連接。
- ❗ 使用上的注意事項
用100BASE-T/10BASE-T自適應HUB連接時，有產生連接故障的情況。請採用中繼HUB，而不用HUB100BASE-T/10BASE-T自適應HUB。
- ③ 計算機的串口與COM模件的編程器通訊端口用編程器電纜連接。

● 路由COM模件的設定

- ① 啟動編程器。
- ② 打開ME20X用的項目。
未製作項目的場合，請執行[項目的新規作成]。
- ③ 按菜單的[選項] → 打開[項目設定]。
》項目設定窗口打開。



- ④ 請把項目設定窗口的[模件地址]設為與要連接的COM模件用撥碼開關相同的值。
- ⑤ 請把[通訊路由]設為[串行(直接)]。
- ⑥ 請按[屬性]按鈕。
》串行通訊設定窗口打開。



- ⑦ 請把[COM端口]設為與用編程器電纜連接的計算機的串口相同的名稱，按[OK]按鈕。
》串行通訊設定窗口關閉。
- ⑧ 請按項目設定窗口的[OK]按鈕。
- ⑨ 從[在線] → [監視開始]置為在線狀態，變更COM模件的以態關連的通訊設定。
請設定COM模件的IP地址、子網掩碼、缺省路由、超時檢查。設定內容根據使用的環境決定。請向使用環境的網絡管理員者詢問。

下圖是IP地址192.168.1.2、子網掩碼255.255.255.0、缺省路由192.168.1.37的設定例。

インスタンスヘッダ		インスタンスボディ	
			1
1	伝送速度(RS-485ポート1)	bps	9600
2	プロトコル(RS-485ポート1)	--	2
3	伝送速度(RS-485ポート2)	bps	9600
4	プロトコル(RS-485ポート2)	--	1
5	IPアドレス	IPA	192.168.1.2
6	サブネットマスク	IPA	255.255.255.0
7	デフォルトルータ	IPA	192.168.1.37
8	KeepAliveTime	--	7200
9	IPポート1	--	1250
10	プロトコル(IPポート1)	--	2
11	IPポート2	--	1251
12	プロトコル(IPポート2)	--	2
13	IPポート3	--	1252
14	プロトコル(IPポート3)	--	1
15	IPポート4	--	1253
16	プロトコル(IPポート4)	--	1

如果沒有特別需要，不需變更KeepAliveTime，IP端口1~4的各設定。

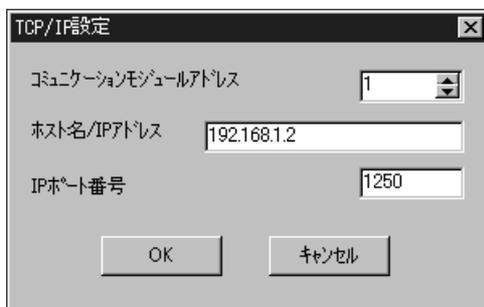
- ⑩ 要使設定生效，請把COM模件重新通電。

● CTRL模件的連接

- ① 打開要連接的CTRL模件用的項目。
未製作的場合，請執行[項目的新規製作]。
- ② 按菜單的[選項] → 打開[項目設定]。
》項目設定窗口打開。



- ③ 請把項目設定窗口的[模件地址]設為與要連接的CTRL模件的撥碼開關相同的值。
- ④ 請把[通訊路由]設為[TCP/IP(間接)]。
- ⑤ 請按[屬性]。
》TCP/IP設定窗口打開。



- ⑥ 請把[通訊模件地址]設為與要經由的COM模件的撥碼開關相同的值。
- ⑦ 請把[主站名/IP地址]設為與COM模件的IP地址相同的值(上記例為192.168.1.2)，按[OK]按鈕。
》串行通訊設定窗口關閉。

! 使用上的注意事項

用編程器打開多個項目、同時經由以態網連接の場合，請勿同時使用IP端口1、IP端口2，一側的IP端口請作為備用。
在計算機進行備份の場合，當編程器不能執行恰當的通訊切斷時，請重新通電，在由Keep Alive Time設定的時間內，有不能連接的可能。

- ⑧ 請把項目設定窗口的[模件地址]設為與要連接的CTRL模件用的撥碼開關相同的值。

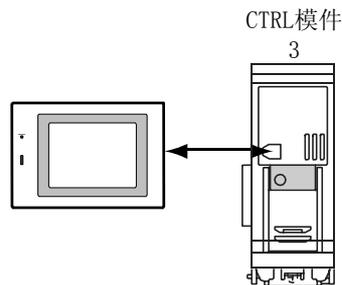
- ⑨ 請按項目設定窗口的[OK]按鈕。
- ⑩ 讀出CTRL模件的信息時，請按[在線] → [監視開始]選擇。

■ 由顯示器通訊端口進行CPL通訊

計算機進行CPL通訊の場合，請編制客戶專用程序。這裏對與本公司產顯示器連接時的情況進行說明。

對在要連接的CTRL模件的顯示器通訊端口上直接連接顯示器的方法進行說明。

例：模件地址為3的CTRL模件上連接顯示器



計算機的串口與CTRL模件的編程器通訊端口用編程器電纜連接。

● 顯示器通訊端口的設定

- ① 啟動編程器。
- ② 打開CTRL模件用的項目。
未製作項目的場合，請執行[項目的新規作成]。
- ③ 按菜單的[選項] → 打開[項目設定]。
》項目設定窗口打開。



- ④ 請把項目設定窗口的[模件地址]設為與要連接的CTRL模件的撥碼開關相同的值。
- ⑤ 請把[通訊路由]設為[串行(直接)]。

- ⑥ 請按[屬性]按鈕。
 》 串行通訊設定窗口打開。



- ⑦ 請把[COM端口]設為與用編程器電纜連接的計算機的串口相同的名稱，按[OK]按鈕。
 》 串行通訊設定窗口關閉。
- ⑧ 請按項目設定窗口的[OK]按鈕。
- ⑨ 按[在線] → [監視開始]置為在線狀態，請變更CTRL模件的通訊設定(前面端口用)。
 本例中，由於用傳送速度19200bps連接，所以要變更設定。



- ⑩ 要使設定生效，請把CTRL模件重新通電。

● 設置

- ① 按[在線] → [監視停止]置為在線狀態，請從CTRL模件上取下編程器電纜的插頭。
- ② CTRL模件的顯示器通訊端口與顯示器的RS-485端口用顯示器通訊用電纜連接。
- ③ 顯示器側的通訊設定如下。

傳送速度	設定值(本例為19200bps)
校驗	偶數校驗
停止位長	1停止位
數據長	數據 8位
流量控制	XON控制/無S參數

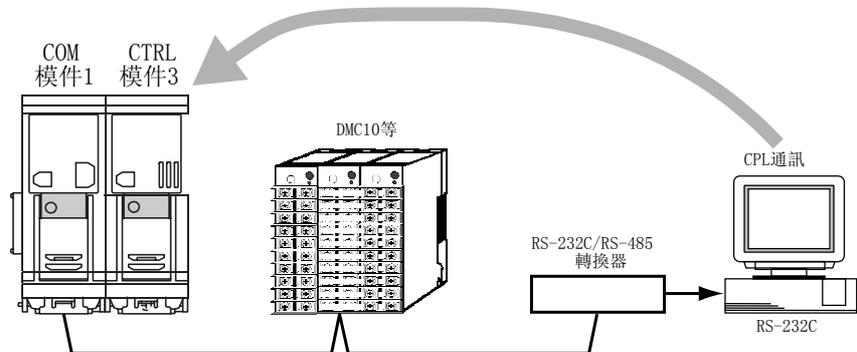
❗ 使用上的注意事項

編程器通訊端口及顯示器通訊端口不能同時使用。編程器通訊用的插口與顯示器通訊用的接口同時連接時，有不能正確通訊的情況。

■ 用RS-485端口進行CPL通訊

用計算機進行CPL通訊の場合，請編制用戶專用的程序。
對經由COM模件的RS-485端口與CTRL模件連接的方法進行說明。

例：經由COM模件(模件地址1)與模件地址為3的CTRL模件連接。



● 設置

- ① COM模件與CTRL模件用連接用接頭連接。
- ② COM模件的RS-485端口與轉換器的RS-485端口連接。
- ③ 計算機(或上位計算機)的通訊設定如下。

傳送速度	設定值(本例為19200bps)
校驗	偶數校驗
停止位長	1停止位
數據長	8位

● COM模件 RS-485端口的設定

- ① 啟動編程器。
- ② 打開COM模件用的項目。
未作成的場合，請執行[項目的新規作成]。
- ③ 按菜單的[選項] → 打開[項目設定]。
》項目設定窗口打開。



- ④ 請把項目設定窗口的[模件地址]設為與要連接的COM模件用的撥碼開關相同的值。
- ⑤ 請把[通訊路由]設為[串行(直接)]。
- ⑥ 請按[屬性]按鈕。
 》串行通訊設定窗口打開。



- ⑦ 請把[COM端口]設為與用編程器電纜連接的計算機的串口相同的名稱，按[OK]按鈕。
 》串行通訊設定窗口關閉。
- ⑧ 請按項目設定窗口的[OK]按鈕。
- ⑨ 請按[在線] → [監視開始]置為在線狀態，變更要連接的RS-485的設定。請變更要使用的端口的傳送速度。
 下圖是變更為19200bps的例。

インスタンスヘッダ		インスタンスボディ	
			1
1	伝送速度(RS-485ポート1)	bps	9600=9600bps
2	プロトコル(RS-485ポート1)	--	9600=9600bps
3	伝送速度(RS-485ポート2)	bps	38400=38400bps
4	プロトコル(RS-485ポート2)	--	1
5	IPアドレス	IPA	192.168.1.2
6	サブネットマスク	IPA	255.255.255.0

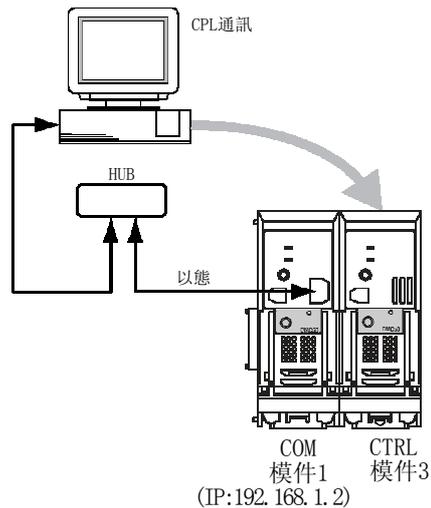
- ⑩ 把協議設為CRL。
 要使設定生效，請把COM模件重新通電源。

■ 用以態網進行CPL通訊

經由以態網進行CPL通訊の場合，請編制客戶專用的程序。編制程序的詳細內容在下章進行說明。

此處，對DMC50的連接進行說明。

例：經由COM模件(模件地址1、IP地址192.168.1.2)與模件地址為3的CTRL模件連接。



● 設置

- ① 設定計算機(或上位計算機)的TCP/IP。
根據環境，需要設定缺省的路由。詳細情況請與使用環境的網絡管理者詢問。
- ② 10BASE-T電纜通過中繼HUB連接計算機與COM模件。
- ❗ 使用上的注意事項
通過100BASE-T/10BASE-T自適應HUB連接時，有產生連接故障的情況。
請使用中繼HUB，而不用100BASE-T/10BASE-T自適應HUB。
- ③ 計算機的串口與COM模件的編程器通訊端口上用編程器電纜連接。

● COM模件 以態端口的設定

- ① 啟動編程器。
- ② 打開COM模件用的項目。
未製作項目的場合，請執行[項目的新規作成]。
- ③ 按菜單的[選項] → 打開[項目設定]。
》項目設定窗口打開。



- ④ 請把項目設定窗口的[模件地址]設為與要連接的COM模件用的撥碼開關相同的值。
- ⑤ 請按[屬性]按鈕。
》串行通訊設定窗口打開。



- ⑥ 請把[COM端口]設為與用編程器電纜連接的計算機的串口相同的名稱，請按[OK]按鈕。
》串行通訊設定窗口關閉。
- ⑦ 請按項目設定窗口的[OK]按鈕。
- ⑧ 按[在線] → [監視開始]置為在線狀態，變更COM模件的以態關連的通訊設定。請設定COM模件的IP地址、子網掩碼、缺省路由、超時檢查。設定內容根據使用環境有差異。請向使用環境的網絡管理員詢問。

下圖是IP地址192.168.1.2、子網掩碼255.255.255.0、缺省路由192.168.1.37的設定例。

インスタンスヘッダ		インスタンスボディ	
			1
1	伝送速度(RS-485ポート1)	bps	9600
2	ポート番号(RS-485ポート1)	--	2
3	伝送速度(RS-485ポート2)	bps	9600
4	ポート番号(RS-485ポート2)	--	1
5	IPアドレス	IPA	192.168.1.2
6	サブネットマスク	IPA	255.255.255.0
7	デフォルトゲート	IPA	192.168.1.37
8	KeepAliveTime	--	7200
9	IPポート1	--	1250
10	ポート番号(IPポート1)	--	2
11	IPポート2	--	1251
12	ポート番号(IPポート2)	--	2
13	IPポート3	--	1252
14	ポート番号(IPポート3)	--	1
15	IPポート4	--	1253
16	ポート番号(IPポート4)	--	1

如沒有特別需要，不需變更KeepAliveTime、IP端口1~4的各設定。

- ⑨ 要使設定生效，請把COM模件重新通電。

2 - 4 時間規格

■命令電文、應答電文時間規格

主站與從站連接の場合，有關主站的命令電文發送及從站的應答電文發送時間，請注意以下事項。

● 應答監視時間

主站發送命令電文結束後，到開始接收來自從站的應答電文的最長應答時間為3秒。(①的部分)

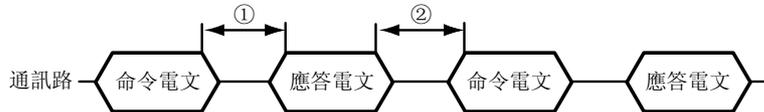
因此，請設定應答監視時間為3秒。

在應答監視時間到達的情況下，請再次發送命令電文。

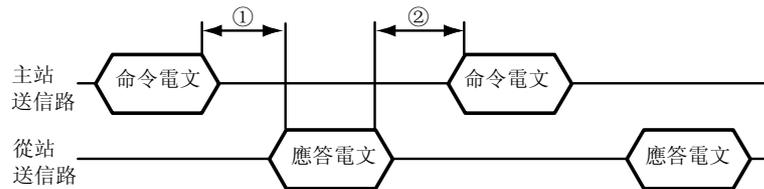
● 送信開始時間

主站接收完應答電文後，到開始發送下一個命令電文需要等待(發送到同一從站的場合和發送到不同從站的場合) 10ms以上。(②的部分)

- RS-485 3線式



- RS-485 5線式

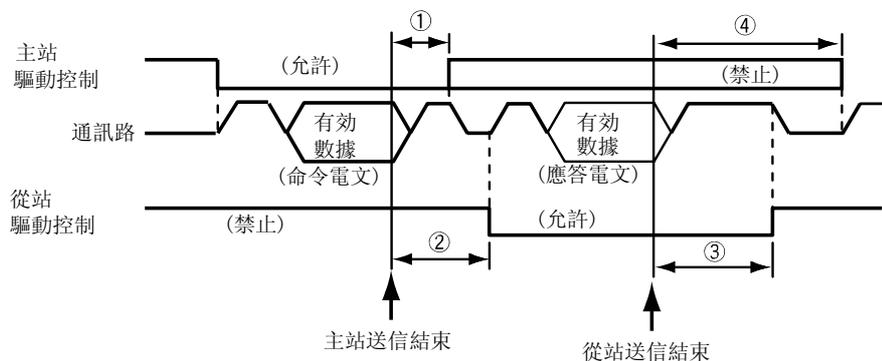


① 主站送信結束 - 從站送信開始時間 = 3000ms以下

② 從站送信結束 - 主站送信開始時間 = 10ms以上

■ RS-485驅動控制時間規格

由主站直接控制RS-485 3線式發送/接收の場合，請注意以下時間。



① 主站發送結束 - 驅動禁止時間 = 500 μs 以下

② 從站收信結束 - 驅動允許時間 = 1ms 以上

③ 從站發送結束 - 驅動禁止時間 = 10ms 以下

④ 主站收信結束 - 驅動允許時間 = 10ms 以上

第 3 章 以態通訊

3 - 1 TCP/IP堆棧規格

COM模件的TCP/IP的相關事項的說明。

■ IP端口與連接

● IP端口

ME20X有四個IP端口。出廠時設定的各IP端口編號及協議的關係如下。編程器通訊、CPL通訊可使用的端口各有二個，但當超時發生時作為備用的原因，請勿同時使用2個端口。

IP端口No.	端口編號	協議
1	1250	編程器通訊服務器
2	1251	編程器通訊服務器
3	1252	CPL通訊服務器
4	1253	CPL通訊服務器

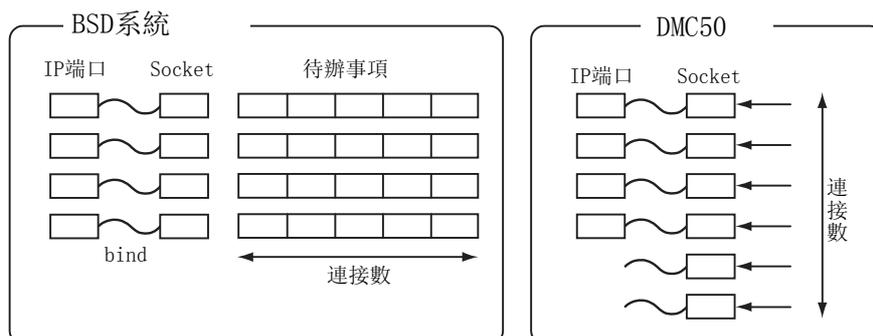
📖 參考

作為registeredport，根據IANA(Internet Assigned Numbers Authority)登錄後，端口編號1024~49151的範圍可按一覽表形式表示。出廠時設定值的端口編號1250~1253是registered port，但2000年10月（現在）未登錄。

● 連接

全IP端口合計的ME20X的TCP/IP堆棧的連接數為6個。如BSD(Berkeley Software Distribution)系統那樣，各IP端口沒有待辦事項(用於具有連接的客戶端的隊列的數量)，由於用「Socket描述符的數量」進行管理，所以全部的IP端口合計為6個。

Socket描述符是一般由BSD等系統使用的方式，是供內部/外部的過程及通訊使用的。把它當作文件描述符的擴展就更容易理解。



3 - 2 客戶端應用

經由以態網進行CPL通訊の場合，請由客戶編制專用程序。在此對程序編制時的詳細內容進行說明。

■ 錯誤處理

● CPL錯誤

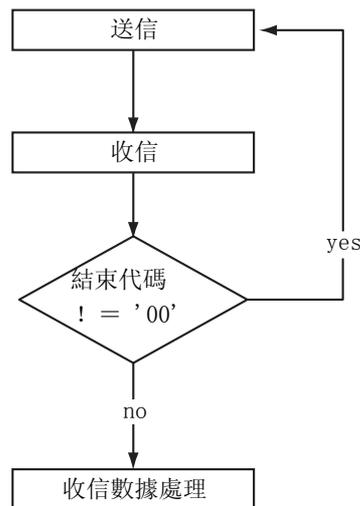
CPL通訊產生如下的結束代碼。

結束代碼	結果
00	正常結束
13	執行時異常
80	CPL Ack 響應

- 代碼「13」、代碼「80」の場合，請執行再試處理。
- 上表以外的代碼の場合，CPL楨有不正確的可能。在應用層面上請勿執行重試等異常處理，而是消除發生的原因。

❗ 使用上的注意事項

多次執行重試處理(例如20次以上)也得不到正常結束代碼の場合，有可能是DMC50側發生了異常。



● TCP級別的錯誤

發生超時的場合，請再次確認超時時間的設定。超時時間按CPL的規定，為3秒，請加上TCP網絡上的RTT(Round Trip Time)。

$$tmout=3(sec)+RTT$$

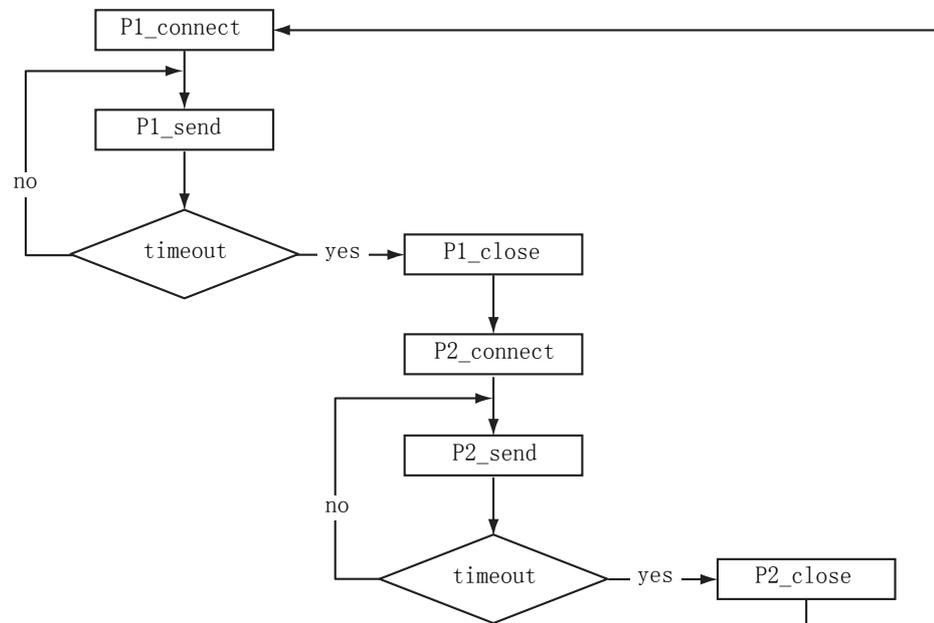
實際上，把以態網作為LAN使用的場合，3秒沒有問題。

📖 參考

支持4.4BSD Socket層的系統中，有監視描述符集合的select()函數。BSD的TCP/IP程序的超時在調用該函數時指定。

● TCP級別的超時發生時的錯誤處理

發生超時的場合，請按以下方法切換二個CPL端口。下圖中，把二個端口分別作為P1、P2。



■ 例子程序

❗ 使用上的注意事項

有關使用該例子程序給裝置造成的損害，本公司概不負責。

```

/*****
 *
 * DMC50 TCP/IP communication test client program
 *
 *****/
#ifdef HAVE_CONFIG_H
# include "config.h"
#endif
#include <ctype.h>
#include <sys/types.h> /* basic system data types */
#include <sys/socket.h> /* basic socket definitions */
#include <sys/time.h> /* timeval{} for select() */
#include <time.h> /* timespec{} for pselect() */
#include <netinet/in.h> /* sockaddr_in{} and other Internet defns */
#include <arpa/inet.h> /* inet(3) functions */
#include <errno.h>
#include <fcntl.h> /* for nonblocking */
#include <netdb.h>
#include <signal.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/stat.h> /* for S_xxx file mode constants */
#include <sys/uio.h> /* for iovec{} and readv/writev */
#include <unistd.h>
#include <sys/wait.h>
#include <sys/un.h> /* for Unix domain sockets */

#ifdef HAVE_SYS_SELECT_H
# include <sys/select.h> /* for convenience */
#endif

#ifdef HAVE_POLL_H
# include <poll.h> /* for convenience */
#endif

#ifdef HAVE_STRINGS_H
# include <strings.h> /* for convenience */
#endif

#include <stdarg.h> /* ANSI C header file */
#include <syslog.h> /* for syslog() */

/* Three headers are normally needed for socket/file ioctl's:
 * <sys/ioctl.h>, <sys/filio.h>, and <sys/sockio.h>.
 */
#ifdef HAVE_SYS_IOCTL_H
# include <sys/ioctl.h>
#endif

#ifdef HAVE_SYS_FILIO_H
# include <sys/filio.h>
#endif
#ifdef HAVE_SYS_SOCKIO_H
# include <sys/sockio.h>
#endif

#ifdef HAVE_PTHREAD_H

```

```

#include <pthread.h>
#endif

static void err_doit(int errnoflag, int level, const char *fmt, va_list ap) ;
void err_quit(const char *fmt, ...) ;
void err_msg(const char *fmt, ...) ;
void err_sys(const char *fmt, ...) ;
void usage(const char *msg) ;
void dump_buf( char *ptr, int nrcv ) ;
int recv_data( int so, char *vptr, int len, int pause ) ;

int daemon_proc=0 ;

#define TIMEOUT 3 /* cpl default timeout sec */
/* default test argumets */
#define DEFAULTTARG1 "0101xRN00LL24300202243002040E6001020E6001040E600106"
#define DEFAULTTARG2 "0100XRGLL00100101000A"
#define LOOPS_DEFAULT 1
#define PAUSE_DEFAULT 0
#define NON_SUM 0
#define WITH_SUM 1
#define WITH_SUM_ILL 2
#define CMD_SEND_DEFAULT 1
#define CMD_SEND_USER 0
#define WITH_LINT_BEFORE 1
#define WITH_LINT_AFTER 2

char cplComAppBuf[512];

#ifdef notdef /* delete if <stdlib.h> doesn't define these for
getopt() */
extern char *optarg;
extern int optind, opterr, optopt;
#endif

#define MAXLINE 4096 /* max line length */
char *host;
char *port;

#define STX 0x02
#define ETX 0x03
#define CR 0x0a
#define LF 0x0d
#define CPLENGTH 512

unsigned char cplFrame[1024]; /* buffer for cpl command frame */
unsigned char rb[1024]; /* buffer for cpl response frame */

/* set CPL frame */
/*
char *cplp buffer for CPL command to be stored
char *seq CPL command string
int mode checksum mode
int vmode display mode
int lint frame with lint flag
*/

int Set_cplFrame(char *cplp, char *seq, int mode, int vmode, int lint )
{
int len; /* length of CPL command string */
int len2; /* length of CPL command string +2 */
int st=0; /* frame start pointer */
char sum; /* temp var to calculate checksum */
char chksum; /* checksum value */
unsigned char *checkptr; /* pointer to checksum value */
char chksumStr[10]; /* checksum string */
int i;

```

```

len = strlen(seq);
cplp[st++] = STX;
len2 = 1 ;

if( vmode == 0 ) {
    /* printf("COMMAND := %s\n", seq); */
}

strncpy(&cplp[st], seq, len);
cplp[st + len] = ETX;
len2 = len + st + 1 ;

if( mode == WITH_SUM ) {
    /* generate checksum of this frame */
    sum = 0;
    for (i=st-1;i < len2; i++) {
        sum += cplp[i];
    }

    chksum = -sum;
    checkptr = (unsigned char *)&chksum;
    sprintf(chksumStr, "%02X", *checkptr);
    strncpy(&cplp[len2], &chksumStr[strlen(chksumStr)-2], 2);
    len2 += 2 ;
    cplp[len2] = 0x0d;
    cplp[len2+1] = 0x0a;
    len2 += 2 ;
}
else {
    cplp[len2] = 0x0d ;
    cplp[len2+1] = 0x0a ;
    len2+=2 ;
}

/* return the length of the CPL command frame */
return (len2);
}

/* check CPL response frame */
/*
char *cplrp    buffer for CPL command response
int rlen      length of CPL command response
int smode     sum mode
int vmode     verbose mode flag
*/
int Check_cplResponse(char *cplrp, int rlen, int smode, int vmode)
{
    char pbuf[1024];                /* buffer for print */
    int content;                   /* content flag for CPL response */
    int pi;                        /* counter for print buffer */
    int exitfor;                   /* flag to escape for loop */
    int ri;                        /* counter for response buffer */
    char sum;                      /* temp var to calculate checksum */
    char chksum;                   /* checksum value */
    char rchksum;                  /* checksum value in response packet */

    if ( rlen == 0 )
        return rlen ;
    /* reset counters and flags */
    pi = 0;
    content = 0;
    exitfor = 0;

    for (ri = 0; ri < rlen; ri++) {

```

```

switch(cplrp[ri]) {
case STX:
    content = 1;
    sum = STX;
    continue;
case ETX:
    exitfor= 1;
    sum += cplrp[ri];
    break;
default:
    if (content == 1) {
        pbuf[pi++] = cplrp[ri];
        sum += cplrp[ri];
    }
}
if (exitfor)
    break;
}
if (exitfor == 0) {
    /* if there is no ETX */
    printf("ERROR : broken response");
    exit(1);
}

/* check response checksum */
chksum = -sum;
ri++;
cplrp[ri+2] = 0;
rchksum = strtol(&cplrp[ri], NULL, 16);
if( smode ) {
    if (chksum != rchksum) {
        /* if checksums do NOT match */
        pbuf[pi] = 0;
        printf("checksum ERROR %02X, %02X, %s\n",chksum,rchksum, pbuf);
        exit(1);
    }
}

/* print the contents of the response */
pbuf[pi]= 0;
if (pbuf[5] == '8' && pbuf[6] == '0') {
    printf("ACK RESPONSE:= %s\n\n", pbuf);
    /* exit(1); */
} else if (pbuf[5] != '0' || pbuf[6] != '0') {
    printf("ERROR RESPONSE:= %s\n\n", pbuf);
    /* exit(1); */
} else {
    /* printf("RESPONSE:= %s\n\n", pbuf);*/
}
return rlen ;
}

/* send data in the specified buffer to the specified socket */
int S_write(int so, unsigned char *buf, int len, int verbose)
{
    int wpsiz, wsiz;
    static int counter = 1;
    time_t tm;
    char *tbuf ;
    tbuf = buf ;
    for (wpsiz = len; wpsiz > 0; wpsiz -= wsiz, buf += wsiz) {
        if ((wsiz = send(so, buf, wpsiz,0)) < 0) {
            printf("nsend ERROR");
            exit(1);
        }
    }
}

```

```

    }
    tm = time(NULL);
    printf( "\nSend[%d] %.24s %d Bytes\n", counter++ , ctime(&tm), wsiz );
    if( verbose != 0 )
        dump_buf( tbuf, wsiz );
    return(len - wpsiz);
}

/* recieve data in the specified buffer from the specified socket */

int recv_data( int so, char *vptr, int len, int pause )
{
    int n, rc ;
    char *ptr ;
    int flg = 0 ;

    ptr = vptr ;
    n = 0 ;
    while(1) {
        if( ioctl(so, FIONREAD, &rc) == -1 ) { /* number of bytes in socket */
            err_quit( "ioctl failed!!\n" );
        }
        else {
            if( rc == 0 ) { /* socket empty */
                if( ( flg==0 ) && ( pause <= 0 ) ) {
                    n=0 ;
                    break ;
                }
                else if( flg ) {
                    break ;
                }
                else {
                    sleep(1) ;
                    pause-- ;
                }
            }
            if ( rc > len )
                rc = len ;

            if( ( rc = recv( so, (void *)vptr, rc, NULL ) ) == EOF ) {
                printf( "recv EOF\n" );
                break ;
            }
            else if ( rc != -1 ) {
                flg = 1 ;
                ptr += rc ;
                ptr++ ;
                *ptr = 0 ;
                n = n+rc ;
            }
            else {
                err_msg( "Error: recv err!" );
            }
        }
    }

    return n ;
}

/* Entry point of this demo program */

int main(int argc, char *argv[])
{
    struct sockaddr_in name; /* structure for socket */
    int so; /* socket for connection to dmc50 */

    int cx; /* CPL command index */
    int cplen; /* length of CPL command frame */

```

```

int rsiz; /* received byte from the socket */
int ret; /* return value of select() */
struct timeval timeout, *ptimeout; /* timeout for select */

fd_set readmask; /* read mask backup for select() */
fd_set readfds; /* array of file descriptors */

char ch; /* chracter received from the socket */
char lch; /* last chracter received from the socket */
int ri; /* cpl response buffer counter */
int reslen; /* length of a cpl response frame */

int loops=LOOPS_DEFAULT ; /* number of loops issueing the commands */
int pause=PAUSE_DEFAULT ; /* interval between commandes to send */

int summode = NON_SUM ; /* cpl frame default sum mode is non-sum*/
int nmulti = 0 ;

int cmddef = CMD_SEND_USER ; /* default command send mode */
int count = 0 ; /* How many loops executed */
int rfrmcnt = 1 ; /* recev frame count */
int tmout = TIMEOUT ; /* cpl timeout */
int verbose = 0;
int lint = 0 ; /* lint data flag */

int i ;
int c ;
unsigned long inaddr;
struct servent *sp;
struct hostent *hp;
char *protocol ;
short pnum,pnum1,pnum2 ;

setbuf(stdout, NULL); /* stop buffering if stdout is connected to pipe */
if (argc < 2) {
    usage("");
}

while ( (c = getopt(argc, argv, "hi:p:o:vSsLLmc")) != EOF) {
    switch (c) {
        case 'h':
            usage("");
            break ;
        case 'i':
            if( (loops = atoi(optarg)) <= 0 )
                usage("");
            break ;
        case 'p':
            if( (pause = atoi(optarg)) <= 0 )
                pause = PAUSE_DEFAULT ;
            break ;
        case 'o':
            if( (tmout = atoi(optarg)) <= 0 )
                tmout = TIMEOUT ;
            break ;
        case 'v':
            verbose = 1 ;
            break ;
        case 'S':
            summode = WITH_SUM ;
            break ;
        case 'm':
            nmulti = 1 ;
            break ;
        case '?':
            printf( "unrecognized option" );
    }
}

```

```

        break ;
    }
}

if( optind > (argc-2) ) {
    usage("missing serverip, portnum and count\n") ;
}
else if( optind == (argc-2) ) { /* no frame input by user */
    cmddef = CMD_SEND_USER ;
    argc += 2 ;
    argv[optind+2] = DEFAULTARG1 ;
    argv[optind+3] = DEFAULTARG2 ;
}
else if( (optind == (argc-3)) && ( nmulti ) ) {
    argc ++ ;
    argv[optind+3] = DEFAULTARG2 ;
}
host = argv[optind] ;
port = argv[optind+1] ;

if( nmulti ) {
    tmout = 0 ; /* select timeout is forever */
}

/* initialize socket address structure */
bzero( (char *)&name, sizeof(name) ) ;

/*
 * First try to convert the host name as a dotted-decimal number.
 * Only if that fails do we call gethostbyname().
 */
if ( (inaddr = inet_addr(host)) != INADDR_NONE) { /* it's dotted-decimal */
    bcopy((char *) &inaddr, (char *) &name.sin_addr, sizeof(inaddr));
}
else {
    if ( (hp = gethostbyname(host)) == NULL) {
        printf("gethostbyname() error for: %s", host);
        exit(1) ;
    }
    bcopy(hp->h_addr, (char *) &name.sin_addr, hp->h_length);
}

/* see if "port" is a service name or number */
protocol = "tcp";
if ( (i = atoi(port)) == 0) {
    if ( (sp = getservbyname(port, protocol)) == NULL) {
        printf("getservbyname() error for: %s/%s", port, protocol);
        exit(1) ;
    }
    pnum1 = sp->s_port;
    pnum2 = pnum1+1;
}
else{
    pnum1 = htons(i) ;
    pnum2 = pnum1+1;
}

printf( "port1=%d\n", pnum1 ) ;
printf( "port2=%d\n", pnum2 ) ;
pnum = 0;

/* set sockaddr_in structure */
/* create socket */
connectagain:

if (pnum == pnum1) /* change service port to another port */
    pnum = pnum2;

```

```

else
    pnum = pnum1;

if ((so = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
    printf("ncan't get socket");
    exit(1);
}
printf("socket opened %d\n", so);

name.sin_family = AF_INET;
/* name.sin_len = (u_char)sizeof(name); */
name.sin_port = pnum;
/* connect to the server */
if (connect(so, (struct sockaddr *)&name, sizeof(name)) < 0) {
    printf("connection ERROR = %d\n", errno);
    exit(1);
}
printf("connected on %s:%d\n", inet_ntoa(name.sin_addr), ntohs(name.sin_port));
while(1) {
    for(cx = (optind+2); cx < argc; cx++) {
        /* Issue a CPL command supplied with a command argument */

        /* set CPL command frame to cplFrame */
        if( multi ) { /* set -m option :multi frame contain one TCP segment */
            int scl;
            cpllen = Set_cplFrame(cplFrame, argv[cx], summode, verbose, lint);
            scl = cpllen;
            cx++;
            cpllen = Set_cplFrame(&cplFrame[cpllen], argv[cx], summode, verbose, lint);
            cpllen = scl + cpllen;
        }
        else
            cpllen = Set_cplFrame(cplFrame, argv[cx], summode, verbose, lint);
        /* send cplFrame to TCP/IP */
        S_write(so, cplFrame, cpllen, verbose);

        ri = 0;
        ch = lch = 0;
        /* socket for cpl comm is set for readmask */
        FD_ZERO(&readmask);
        FD_SET(so, &readmask);

        while(1) {
            /* timeout for select() is set to tmout */
            timeout.tv_sec = tmout;
            timeout.tv_usec = 0;
            if( tmout )
                ptimeout = &timeout;
            else
                ptimeout = NULL;

            readfds = readmask;

            /* waiting for inputs */
            if ( (ret = select(so+1, (fd_set *)&readfds, NULL, NULL, ptimeout)) < 0 )
                err_sys("ERROR: select failed\n");
            else if (ret == 0) {
                printf("WARNING : CPL TIMEOUT\n");
                close(so);
                sleep(1);
                goto connectagain;
            } else if (FD_ISSET(so, &readfds)) { /* There is data at socket for cpl */
                /* just receive one byte from the socket */
                rsiz = recv_data(so, rb, sizeof(rb)-1, 5);
                if (rsiz < 0) {
                    printf("ERROR : recv error no reply error %d\n", errno);
                }
            }
        }
    }
}

```

```

        reslen = 0 ;
        close(so) ;
        sleep(1) ;
        goto connectagain ;
    }
    else if (rsiz == 0 ) {
        /* closed connection by peer */
        printf("Warning : connection closed by peer\n" ) ;
        close(so) ; /* socket close */
        goto connectagain ; /* and connect again to peer */
    }
    else {
        time_t tm;
        tm = time(NULL) ;
        printf( "Recv[%d] %.24s %d Bytes\n", rfrmcnt++, ctime(&tm), rsiz ) ;
        if( verbose ) {
            dump_buf( rb, rsiz ) ;
        }
        break ;
    }
} /* end of else if */
} /* end of while(1) */
Check_cplResponse(rb, reslen, summode, verbose) ;
if( reslen != 0 )
    sleep( pause ) ;
} /* end of for */
count++;
if (count >= loops)
    break;
}
exit(0); // close socket and sucessfully exit from this program
}

void err_sys(const char *fmt, ...)
{
    va_list ap;

    va_start(ap, fmt);
    err_doit(1, LOG_ERR, fmt, ap);
    va_end(ap);
    exit(1);
}

void err_msg(const char *fmt, ...)
{
    va_list ap;

    va_start(ap, fmt);
    err_doit(0, LOG_INFO, fmt, ap);
    va_end(ap);
    return;
}

void err_quit(const char *fmt, ...)
{
    va_list ap;

    va_start(ap, fmt);
    err_doit(0, LOG_ERR, fmt, ap);
    va_end(ap);
    exit(1);
}

static void err_doit(int errnoflag, int level, const char *fmt, va_list ap)
{
    int     errno_save, n;
    char    buf[MAXLINE];

```

```

    errno_save = errno;                /* value caller might want printed */

    vsnprintf(buf, sizeof(buf), fmt, ap);
    n = strlen(buf);
    if (errnoflag)
        snprintf(buf+n, sizeof(buf)-n, ": %s", strerror(errno_save));
    strcat(buf, "\n");

    if (daemon_proc) {
        syslog(level, buf);
    } else {
        fflush(stdout);                /* in case stdout and stderr are the same */
        fputs(buf, stderr);
        fflush(stderr);
    }
    return;
}

/* dump output ptr */
void dump_buf( char *ptr, int nrcv )
{
    int j, ch ;
    u_char cbuf[17], d ;

    for( j = 0; j < 16; j++ ) {
        printf( "%02X", j ) ;
    }
    printf( " ASCII %n" ) ;
    while(nrcv>0) {
        for(j = 0; j < ( nrcv < 16 ) ? nrcv :16 ; j++ ) {
            d = (u_char)*ptr ;
            printf( " %02x", d ) ;
            ch = (int)d ;
            if( isprint( ch ) )
                cbuf[j] = d ;
            else
                cbuf[j] = ' ' ;

            cbuf[j+1] = 0 ;
            ptr++ ;
        }
        if( nrcv < 16 ) {
            for( j = 0; j < (16-nrcv); j++ ) {
                printf( " " ) ;
            }
        }
        nrcv = nrcv - 16 ;
        printf( " %s\n", cbuf ) ;
    }
}

void usage(const char *msg)
{
    printf( "DMC50 TCP/IP commnication test client program\n\n" ) ;
    err_msg(
        "usage: cpltest [options] <host> <port> [CPL command 1] .. [CPL command n]\n"
        "options: -h      this help\n"
        "          -i n    continuous counts %n"
        "          -p n    seconds to pause after each access %n"
        "          -o n    CPL timeout seconds\n"
        "          -v      verbose\n"
        "          -S      with check sum\n"
        "          -m n    num of frame include over one TCP segment\n"
        "          -c      connect again after connection closed by peer\n"
        "%n\nnote: <port> must be 1st service port number\n"
        "          (factory setting of dmc50 is 1252) %n"
    ) ;
}

```

```
    );  
  
    if (msg[0] != 0)  
        err_quit("%s", msg);  
    exit(1);  
}  
  
/**  
 * $Log: cplsample.c,v $  
 * Revision 1.1 2000/12/05 08:00:09  
 */
```

3 - 3 用語說明

● 以態網(Ethernet)

以態網是世界上廣泛使用的局域網。

計算機或工作站、打印機等針對以態網的設備大量出現，可構建包含這些設備的多功能網絡。

在通用計算機或工作站上編制CPL通訊程序作為TCP/IP應用，ME20X可方便地與DMC50進行通訊。

● TCP(Transmission Control Protocol)

支持連接型的通訊服務。

TCP具有給二個通訊設備間提供可靠服務的必要功能。

例如、

- 確認應答
- 流量控制
- 定時器
- 連接管理功能等。

● IP(Internet Protocol)

是TCP/IP最基本的協議。主要的功能有

- IP地址的處理(路由功能)
- 數據包的分割·再構成

IP地址的處理是，根據送信源的主站上或到達最終發送目標途中的網關上交接的包(或數據包)的IP地址，決定到下一個網關或主站的路由，通過其網絡發送該數據包。

數據包的分割·再構成是指，當收到的數據包的長度比下一網絡允許的最大長度長時，把數據包分割成多個數據包，使用原數據包附帶的信息頭的內容構成各被分割數據包的信息頭。

由最終發送目標根據信息頭的信息，把被分割的多個數據包重新構成原來的數據包。IP中有幾個輔助用的協議，其中ME20X可支持的有ARP、ICMP的二種。

● ARP(Address Resolution Protocol)

是把MAC地址與IP地址結合的協議。

當分配了送信目標的IP地址時，如果自局的內部沒有存儲送信目標的MAC地址的場合，把送信目標IP地址作為信息進行廣播。具有送信目標IP地址的機器收到廣播時，把MAC地址信息返回給發出廣播的機器。

● **ICMP(Internet Control Message Protocol)**

提供網絡診斷相關的手段。

確認IP數據包的錯誤通知及網絡上的狀態等。

例如，

- 不能到達送信目標的IP地址或端口編號等的場合，向送信源設備發出不能送達的通知
- 發出調查對方數據包狀態的要求，進行回答

ICMP自動生成TCP/IP堆棧并解釋。

● **CSMA/CD(Carrier Sense Multiple Access with Collision Detection)**

是以態網的規格，具有衝突檢測功能、允許載波偵聽的多個站可同時訪問的方式。

通訊路上如果沒有送信中的數據，則任何機器都可開始送信。送信中產生衝突(Collision)時，信息有被破壞的可能。

● **MAC地址(以態地址)**

是所有的以態機器上附加的6字節長的地址。

MAC地址是各機器固有的地址，不可變更。另外，世界上不存在有重複MAC地址的機器。

出廠時DMC50 COM模件也設定了具有固有的MAC地址。

客戶程序等使用IP地址。所以，可不考慮MAC地址即可通訊。

● **IP地址**

IP地址是由用戶設定・變更的地址，用戶使用IP地址進行通訊。

對安裝有IP協議的機器設定IP地址(現在為4字節長)。最終由使用場所(部署)的管理者進行分配。

以態經由網橋或調制解調器可構建大規模的網絡。所以，請慎重管理以免IP地址重複。

專用地址(本地地址)由RFC1918規定如下：

級別A: 10. 0. 0. 0~10. 255. 255. 255

級別B: 172. 16. 0. 0~172. 31. 255. 255

級別C: 192. 168. 0. 0~192. 168. 255. 255

● 級別A

0[7bit]. [8bit]. [8bit]. [8bit]
 (從起始到8位是IP網絡地址部、剩餘的24位是IP主地址部)
 IP網絡地址部是被預約為0及127構成的地址。
 IP主地址部的所有位被預約為0或1。

● 級別B

10[6bit]. [8bit]. [8bit]. [8bit]
 (從起始到16位是IP網絡地址部、剩餘的16位是IP主地址部)
 IP網絡地址部被預約為128. 0及191. 255。
 IP主地址部的所有位被預約為0或1。

● 級別C

110[5bit]. [8bit]. [8bit]. [8bit]
 (從起始到24位是IP網絡地址部、剩餘的8位是IP主地址部)
 IP網絡地址部被預約為192. 0. 0及223. 255. 255。
 IP主地址部的所有位被預約為0或1。

● 子網掩碼

是把1個的IP網絡地址分成多個子IP網絡的設定。
 表示子IP網絡的上位的位為1、下位的位全部為0。
 子網絡之間、物理上連接、從外部看是一個IP網絡。
 子網絡部分位的值的設定不可全部為0或1。

📖 參考

- 級別C的IP網絡地址192. 168. 1. 0中，定義1位的子網掩碼的場合。
 (從起始到25位是IP網絡部的場合)
 此時，子網掩碼為
 11111111. 11111111. 11111111. 10000000
 (=255. 255. 255. 128)
- 上記例中，192. 168. 1. 1與192. 168. 1. 129屬於同一IP網絡地址，由子網掩碼進行分割。

● 端口編號

TCP上區別服務種類的編號。
 IP地址是指定機器或站，而端口編號是指定機器或站的服務的種類。不同端口編號之間，各數據分別處理。

● Well-known端口

IP端口編號1~1023稱為well-know端口。

這是由IANA(Internet Assigned Numbes Authority)機構統一管理，公開分配的。

例如，ftp服務的21、telnet服務的23等。

● Registerd端口

IP端口編號1024~49151不是由IANA進行管理的，但作為registeredport，是由IANA登錄及一覽公開的。對ME20X，出廠時設定的1250~1253現在（2000年10月）未登錄。

● KeepAliveTime

用TCP/IP連接在ME20X上的機器與ME20X的通訊正常而無切斷、掉電、重新啟動、進程停止的場合，ME20X不容易確認處於非激活的TCP/IP的連接的周期時間。

出廠時設定為7200秒(=2小時)。所以，2小時內的連接切斷狀態不會通知，連接保持激活狀態。

第 4 章 CPL命令參考

■ 概 要

本公司既有的CPL通訊協議的應用層命令是用16位數據及可對16位的地址空間進行訪問的命令。

DMC50中有32位數據及32位空間，要訪問全部的數據，需要使用32位擴展CPL命令。

❗ 使用上的注意事項

與既有的CPL機器連接時，請注意互換性。

■ 基本楨格式

[XX]的XX是表示命令(響應)中的一個令牌。

(XX)的XX用字節數表示。 例：(2)

'X'的X是表示該文字表示的ASCII代碼。 例：'X'

命令：

[STX][sta(2)][sub(2)]['X'] [cmd(2)] {[args]..} [ETX][sum(2)][CR][LF]

響應：

[STX][sta(2)][sub(2)]['X'] [rc(2)] {[data]..} [ETX][sum(2)][CR][LF]

令牌表示	名 稱	內 容
[STX]	啟動代碼	02h
[sta(2)]	CPL機器地址	通訊電纜連接的模件的撥碼開關*1 (01h..0Fh)
[sub(2)]	CPL子地址	COM模件經由的CTRL模件的撥碼開關 (01h..0Fh) COM模件經由的送信目標不是CTRL模件的場合 為00h
['X']	CPL機器識別代碼	DMC50固定為X
[cmd(2)]~	CPL應用層	命令[args]是各命令的參數
[ETX]	結束代碼	03h
[sum(2)]	檢查和	[STX]~[ETX]的字節加算後取2的補碼 BCX
[CR]	楨結束代碼1	0Dh
[LF]	楨結束代碼2	0Ah
[rc(2)]	結束代碼*2	
[arg]	參數	ASCII代碼(詳見各命令)
[data]	應答數據	ASCII代碼(詳見各命令)

*1 : BCX(Binary Coded Hexadecimal) 01h..FFh

*2 : BCD(Binary Coded Decimal) 01..99

❗ 使用上的注意事項

- BCX數據的'A'~'F'請使用大文字。
- {[XX]..}表示[XX]的反復。

■ 異常處理

DMC50的CPL通訊異常處理分類如下。

- 通訊設定的異常
- 通訊轉送方向的異常
- CPL通訊槓的異常
- CPL應用層的異常

● 通訊設定的異常

通訊設定有異常の場合，通訊不成立，DMC50無應答。

通訊設定有如下的項目。

- DMC50 COM模件撥碼開關
- DMC50 CTRL模件撥碼開關
- 傳送速度

完全不能通訊の場合，首先請確認通訊設定。

● 通訊轉送方向的異常

CPL通訊是半雙工的通訊。命令槓多重送信の場合，除最後發送的槓之外，其它全部槓被破棄。

響應槓的返回慢の場合，在送信側進行超時處理，發生超時後，請再次發送同一命令槓(或不同命令槓)。

超時時間為3秒。

DMC50的内部處理經過3秒以上，造成響應超時の場合，當在命令發送側執行再次發送(同一槓的再發送)後，COM模件立即返回Ack響應的應答。這並不是把前次的槓破棄掉，而是表示是由於要花費時間才發生了超時。前次的槓(與本次同一槓)的處理繼續，處理完成後立即返回通常的響應。

● CPL通訊槓的異常

CPL槓有缺陷、CPL槓不能分離の場合或送信目標地址不一致，DMC50判斷為不是發送給自己的場合，將破棄該槓。這種場合就沒有響應。

超時引起的多次重複處理の場合，請確認通訊槓有無異常。

● CPL應用層的異常

應用層有異常的場合，根據各命令的異常處理，其對策不同。基本上當有表示異常的應答時，請確認應答進行相應的處理。

同時發生多個CPL異常的場合，請按下記的優先順序發送結束代碼。（不反映發生優先順序第2個以後的錯誤代碼）

根據槓解析處理判斷錯誤發生的地方，['99'] ~ ['13'] 的發生順序會變化。（槓解析處理按起始的順序逐1字節進行處理）

寫入

(['99'] > ['10'] > ['13']) ≅ ['40'] > ['21'] > ['23'] > ['22'] > ['43']

讀出

(['99'] > ['10'] > ['13']) ≅ ['40'] > ['21'] > ['23'] > ['22']

另外，命令中的一部份的地址或數據範圍等有異常的場合，不正確的部分被忽略，處理繼續進行。讀出的場合，不正確的部分返回數據'00000000'。寫入的場合，當結束代碼不是'00'時，并不表示全部數據已正確寫入。（但當寫入的場合，不能確定不正確的場所）

❗ 使用上的注意事項

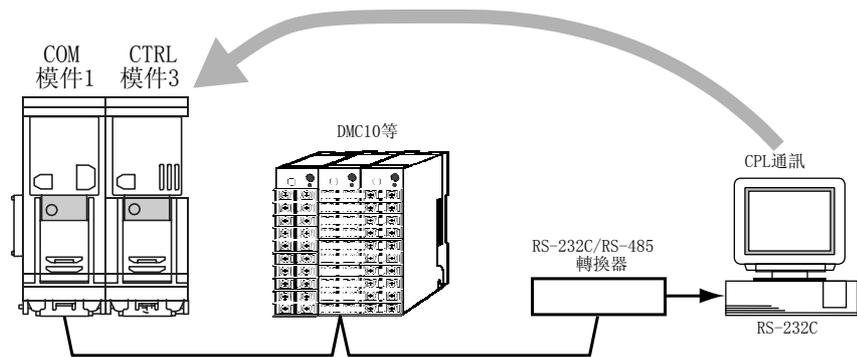
正式運行時，請在先消除通訊的不正確處、使所有的通訊槓的結束代碼為'00'的狀態下，再進行通訊。

■ CPL地址

CPL的棧中有「sta(2)」、「sub(2)」的2種地址。
指定通訊目標機器時使用。

• 經由COM模件時

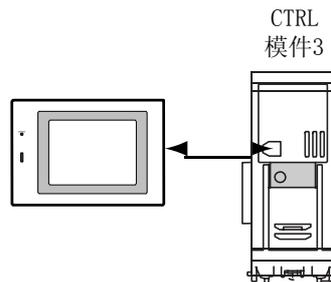
令牌表現	名稱	內容
sta(2)	CPL機器地址	<ul style="list-style-type: none"> 用於識別COM模件的CPL地址 各機器地址稱為「站」 設為與各COM模件的撥碼開關相同的值(01..0F) 00的指定無效
sub(2)	CPL子地址	<ul style="list-style-type: none"> 一個站的DMC50上，用於區別CTRL模件的CPL地址 設為與各CTRL模件的撥碼開關相同的值(01..0F) 指定COM模件時，把子地址設為00



例：與模件地址為1的COM模件連接的模件地址為3的CTRL模件
 sta='01' sub='03'
 模件地址為2的COM模件
 sta='0A' sub='00'

- 經由COM模件時

令牌表現	名稱	內容
sta(2)	CPL機器地址	<ul style="list-style-type: none"> 用於識別CTRL模件的CPL地址 設為與CTRL模件的撥碼開關相同的值(01..0F) 00的指定無效
sub(2)	CPL子地址	<ul style="list-style-type: none"> 由於不使用者子地址，指定為00(即使指定00以外的值，也被忽略)



例: 模件地址為3的CTRL模件

sta='03' sub='00'

命令詳細內容

DMC50支持如下命令。

由[rc(2)], 在應用層的結束代碼中包含了各命令的響應。

結束代碼由各命令決定。

對各命令的項進行說明。

命令	內容
RG	帶數量指定的固定長連續數據讀出
WG	帶數量指定的固定長連續數據寫入
RN	帶重複及數量指定的固定長隨機讀出
WN	帶重複及數量指定的固定長隨機寫入
RD	固定長連續數據讀出(16位訪問)
WD	固定長連續數據寫入(16位訪問)
RS	連續讀出(ASCII 10進制、16位訪問)
WS	連續寫入(ASCII 10進制、16位訪問)
RU	帶重複的固定長隨機讀出(16位訪問)
WU	帶重複的固定長隨機寫入(16位訪問)

16位系命令

使用16位訪問系命令の場合，請在ISaGRAF應用上複製參數到16位區域中。這是由於16位空間是ISaGRAF應用使用的空間。

❗ 使用上的注意事項

由客戶新規編制通訊用程序の場合，請使用32位訪問系的命令。16位訪問系的命令在與本公司產機器連接時使用。

RG

■ RG命令(帶數量指定固定長連續數據讀出(32位系))

從指定起始地址開始讀出指定數量的32位參數。
最大讀出數據個數為50個。

命令:

```
[STX][sta(2)][sub(2)]['X'] ['R'] ['G'] ['LL'] [adr(8)] [len(4)] [ETX]
[sum(2)][CR][LF]
```

響應:

```
[STX][sta(2)][sub(2)]['X'] [rc(2)] {[data(8)]..} [ETX][sum(2)][CR][LF]
```

[adr(8)] → 起始數據地址、1個數據地址(BCX)
[len(4)] → 數據個數(最大50)(BCX)
[data(8)] → 1個數據地址的數據(BCX)
['LL'] → 數量指定子(LL為固定)

RG命令有以下的結束代碼。

代碼	名稱	內容
['10']	參數異常	數量指定子[ss(2)]中指定了['LL']以外的值 32位數據/地址的位長不正確(字節數錯誤) 含有數字('0'~'9'、'A'~'F')以外的數據
['13']	命令執行異常	內部傳送發生超時 ISaGRAF的執行周期設定過短 向不存在的CTRL模件送信
['21']	地址異常	ISaGRAF運行停止狀態下訪問了變量 (10000h以前的地址)
	訪問數據型異常	訪問了本命令(10000h以前的地址)不支持的文字列型
['23']	運算參數異常	訪問了處於運算參數禁止訪問狀態(更新中等)時的地址20100001h~
['40']	讀出字數異常	讀出個數過多或為0
['80']	CPL Ack響應	當前次的信息仍處理中時, 再次發送了與前次相同的信息, 用「現在處理中」表示
['99']	未定義命令	發送了不支持的命令 (要發送'RG', 但書寫錯誤)

❗ 使用上的注意事項

DMC50內部的REAL型數據是IEEE754格式的浮點小數數據。

另外, DINT型數據是負數取2的補補數的固定小數點(無小數點)數據。

使用32位系命令時, 請注意數據訪問時數據的類型。

■ WG命令(帶數量指定的固定長連續數據寫入(32位系))

指定起始地址中寫入指定數量的32位參數。最大寫入數據個數為50。

命令：

```
[STX][sta(2)][sub(2)]['X'] ['W'] ['G'] ['LL'] [adr(8)] {[arg(8)]..}
[ETX][sum(2)][CR][LF]
```

響應：

```
[STX][sta(2)][sub(2)]['X'] [rc(2)] [ETX][sum(2)][CR][LF]
[adr(8)] → 起始數據地址、1個數據地址 (BCX)
[arg(8)] → 1個數據地址中存儲的數據(最大50個) (BCX)
['LL'] → 數量指定子(LL固定)
```

WG命令有以下的結束代碼。

代碼	名稱	內容
['10']	參數異常	數量指定子[ss(2)]中指定了['LL']以外的值 32位數據/地址的位長不正確(字節數錯誤) 含有數字('0' ~ '9'、'A' ~ 'F')以外的數據
['13']	命令執行異常	內部傳送發生超時 ISaGRAF的執行周期設定過短 向不存在的CTRL模件送信
['21']	地址異常	ISaGRAF運行停止狀態下訪問了變量 (10000h以前的地址)
	訪問數據型異常	訪問了本命令(10000h以前的地址)不支持的文字列型
['22']	寫入數據範圍外	向指定地址寫入的數據的類型不符 日曆參數中寫入格式錯誤 指定地址中寫入的數據超過了上下限的範圍
['23']	數據寫入不可	向禁止寫入參數中寫入了數據 在運行中向訪問禁止的參數寫入了數據
	運算參數異常	處於訪問禁止狀態(更新中等)時訪問了運算參數
['40']	寫入字數異常	讀出個數過多或為0
['80']	CPL Ack 響應	當前次的信息仍處理處理中時，再次發送了與前次相同的信息，用「現在處理中」表示
['99']	未定義命令	發送了不支持的命令 (要發送'WG'，但書寫錯誤)

❗ 使用上的注意事項

DMC50內部的REAL型數據是IEEE754格式的浮點小數數據。

另外，DINT型數據是負數取2的補的固定小數點(無小數點)數據。

用32位系命令進行數據訪問時，請注意數據的類型。

RV

■ RN命令(帶重複及數量指定的固定長隨機讀出(32位系))

從指定地址讀出32位參數。地址按1個參數指定時，可不連續的讀出參數。最大讀出數據個數為50個。

命令：

```
[STX][sta(2)][sub(2)]['X'] ['RN'] ['00'] ['LL'] {[adr(8)]...} [ETX]
[sum(2)][CR][LF]
```

響應：

```
[STX][sta(2)][sub(2)]['X'] [rc(2)] {[data(8)]..} [ETX][sum(2)][CR][LF]
```

- ['00'] → 重複登錄('00' 固定)
- [adr(8)] → 起始數據地址、1個數據地址(最大50個)(BCX)
- [data(8)] → 1個數據地址的數據(BCX)
- ['LL'] → 數量指定子(LL為固定)

RN命令有以下的結束代碼。

代碼	名稱	內容
['10']	參數異常	數量指定子[ss(2)]中指定了['LL']以外的值 32位數據/地址的位長不正確(字節數錯誤) 含有數字('0' ~ '9'、'A' ~ 'F')以外的數據
['13']	命令執行異常	內部傳送發生超時 ISaGRA的執行周期設定過短 向不存在的CTRL模件送信
['21']	地址異常	ISaGRAF運行停止狀態下訪問了變量 (10000h以前的地址)
	訪問數據型異常	訪問了本命令(10000h以前的地址)不支持的文字列型
['23']	運算參數異常	處於訪問禁止狀態(更新中等)時訪問了運算參數
['40']	讀出字數異常	讀出個數過多或為0
['80']	CPL Ack 響應	當前次的信息仍處理處理中時，再次發送了與前次相同的信息，用「現在處理中」表示
['99']	未定義命令	發送了不支持的命令 (要發送'RN'，但書寫錯誤)

❗ 使用上的注意事項

DMC50內部的REAL型數據是IEEE754隔閡司的浮點小數數據。

另外，DINT型數據是負數取2的補的固定小數點(無小數點)數據。

用32位系命令進行數據訪問時，請注意數據的類型。

■ WN命令(帶重複及數量指定的固定長隨機寫入(32位系))

向指定地址中寫入32位參數。地址按1個參數指定可寫入不連續的參數。最大寫入數據個數為25個。

命令:

```
[STX][sta(2)][sub(2)]['X'] ['WN'] ['00'] ['LL'] {[adr(8)][arg(8)]...}
[ETX][sum(2)][CR][LF]
```

響應:

```
[STX][sta(2)][sub(2)]['X'] [rc(2)] [ETX][sum(2)][CR][LF]
['00'] → 重複登錄('00'固定)
[adr(8)] → 起始數據地址、1個數據地址(最大25個)(BCX)
[arg(8)] → 1個數據地址中存儲的數據(最大25個)(BCX)
['LL'] → 數量指定子(LL為固定)
```

WN命令有以下的結束代碼。

代碼	名稱	內容
['10']	參數異常	數量指定子[ss(2)]中指定了['LL']以外的值 32位數據/地址的位長不正確(字節數錯誤) 含有數字('0'~'9'、'A'~'F')以外的數據
['13']	命令執行異常	內部傳送發生超時 ISaGRA的執行周期設定過短 向不存在的CTRL模件送信
['21']	地址異常	ISaGRAF運行停止狀態下訪問了變量 (10000h以前的地址)
	訪問數據型異常	訪問了本命令(10000h以前的地址)不支持的文字列型
['22']	寫入數據範圍外	向指定地址寫入的數據的類型不符 日曆參數中寫入格式錯誤 指定地址中寫入的數據超過了上下限的範圍
['23']	數據寫入不可	向禁止寫入參數中寫入了數據 在運行中向訪問禁止的參數寫入了數據
	運算參數異常	處於訪問禁止狀態(更新中等)時訪問了運算參數
['40']	寫入字數異常	寫入個數過多或為0
['80']	CPL Ack 響應	當前次的信息仍處理處理中時, 再次發送了與前次相同的信息, 用「現在處理中」表示
['99']	未定義命令	發送了不支持的命令 (要發送'WN', 但書寫錯誤)

❗ 使用上的注意事項

DMC50內部的REAL型數據是IEEE754隔閡司的浮點小數數據。
另外, DINT型數據是負數取2的補的固定小數點(無小數點)數據。
用32位系命令進行數據訪問時, 請注意數據的類型。

RD

■ RD命令(固定長連續數據讀出(16位系))

從指定起始地址讀出指定數量的16位參數。最大讀出數據個數為50個。

命令:

```
[STX][sta(2)][sub(2)]['X']['R']['D'][adr(4)][len(4)][ETX][sum(2)]
[CR][LF]
```

響應:

```
[STX][sta(2)][sub(2)]['X'][rc(2)]{[data(4)]..}[ETX][sum(2)][CR][LF]
[adr(4)]      →  起始數據地址、1 數據地址 (BCX)
[len(4)]      →  數據個數 (最大 50) (BCX)
[data(4)]     →  1 數據地址的數據 (BCX)
```

RD命令有以下的結束代碼。

代碼	名稱	內容
['10']	參數異常	16位數據/地址的位長不正確(字節數錯誤) 含有數字('0'~'9'、'A'~'F')以外的數據
['13']	命令執行異常	內部傳送中發生超時 ISaGRA的執行周期設定過短 向不存在的CTRL模件送信
['21']	地址異常	ISaGRAF運行停止狀態下訪問了變量 (10000h以前的地址)
	訪問數據型異常	訪問了本命令(10000h以前的地址)不支持的文字列型
['22']	讀出數據範圍外	讀出數據超過了-32768~+32767的範圍 該數據轉換成最大值(32767:7FFFh)或最小值(-32768:8000h)後讀出
['23']	運算參數異常	處於訪問禁止狀態(更新中等)時訪問了運算參數
['40']	讀出字數異常	讀出個數過多或為0
['80']	CPL Ack 響應	當前次的信息仍處理處理中時，再次發送了與前次相同的信息，用「現在處理中」表示
['99']	未定義命令	發送了不支持的命令 (要發送'RD'，但書寫錯誤)

❗ 使用上的注意事項

- 在與本公司產機器連接時，請使用16位系命令。
- 使用16位系命令時，請務必用「整數轉換嚮導」，製作16位系命令可訪問的數據。

■ WD命令(固定長連續數據寫入(16位系))

從指定起始地址寫入指定數量的32位參數。最大寫入數據個數為50個。

命令:

```
[STX][sta(2)][sub(2)]['X'] ['W'] ['D'] [adr(4)] {[dat(4)]...} [ETX]
[sum(2)][CR][LF]
```

響應:

```
[STX][sta(2)][sub(2)]['X'] [rc(2)] [ETX] [sum(2)] [CR] [LF]
[adr(4)] → 起始數據地址、1個數據地址(BCX)
[arg(4)] → 1個數據地址中存儲的數據(最大25個)(BCX)
```

WD命令有以下的結束代碼。

代碼	名稱	內容
['10']	參數異常	16位數據/地址的位長不正確(字節數錯誤) 含有數字('0' ~ '9'、'A' ~ 'F')以外的數據
['13']	命令執行異常	內部傳送發生超時 ISaGRA的執行周期設定過短 向不存在的CTRL模件送信
['21']	地址異常	ISaGRAF運行停止狀態下訪問了變量 (10000h以前的地址)
	訪問數據型異常	訪問了本命令(10000h以前的地址)不支持的文字列型
['23']	運算參數異常	處於訪問禁止狀態(更新中等)時訪問了運算參數
['40']	寫入字數異常	寫入個數過多或為0
['80']	CPL Ack 響應	當前次的信息仍處理處理中時，再次發送了與前次相同的信息，用「現在處理中」表示
['99']	未定義命令	發送了不支持的命令 (要發送'WD'，但書寫錯誤)

❗ 使用上的注意事項

- 在與本公司產機器連接時，請使用16位系命令。
- 使用16位系命令時，請務必用「整數轉換嚮導」，製作16位系命令可訪問的數據。

RU

■ RU命令(帶重複的固定長隨機讀出(16位系))

從指定地址讀出16位參數。地址按1個參數指定可讀出不連續的參數。最大讀出數據個數為50個。

命令:

```
[STX][sta(2)][sub(2)]['X'] ['RU'] ['00'] {[adr(4)]...} [ETX][sum(2)]
[CR][LF]
```

響應:

```
[STX][sta(2)][sub(2)]['X'] [rc(2)] {[data(4)]...} [ETX][sum(2)][CR][LF]
```

['00'] → 重複登錄('00' 固定)

[adr(4)] → 起始數據地址、1個數據地址(最大50個)(BCX)

[arg(4)] → 1個數據地址中存儲的數據(BCX)

RU命令中有以下的結束代碼。

代碼	名稱	內容
['10']	參數異常	重複登錄中指定了'00'以外的值 含有數字('0'~'9'、'A'~'F')以外的數據
['13']	命令執行異常	內部傳送發生超時 ISaGRA的執行周期設定過短 向不存在的CTRL模件送信
['21']	地址異常	ISaGRAF運行停止狀態下訪問了變量 (10000h以前的地址)
	訪問數據型異常	ISaGRAF運行停止狀態下訪問了變量 (10000h以前的地址) 訪問了參數不存在的地址
['22']	讀出數據範圍外	讀出數據超過了-32768~+32767的範圍 該數據轉換成最大值(32767:7FFFh)或最小值 (-32768:8000h)後讀出
['23']	運算參數異常	處於訪問禁止狀態(更新中等)時訪問了運算參數
['40']	讀出字數異常	讀出個數過多或為0
['80']	CPL Ack 響應	當前次的信息仍處理中時, 再次發送了 與前次相同的信息, 用「現在處理中」表示
['99']	未定義命令	發送了不支持的命令 (要發送'RU', 但書寫錯誤)

❗ 使用上的注意事項

- 在與本公司產機器連接時, 請使用16位系命令。
- 使用16位系命令時, 請務必用「整數轉換嚮導」, 製作16位系命令可訪問的數據。

■ WU命令(帶重複的固定長隨機寫入(16位系))

向指定地址中寫入16位參數。地址按1個參數指定可寫入不連續的參數。最大寫入數據個數為25個。

命令：

```
[STX][sta(2)][sub(2)]['X'] ['WU'] ['00'] {[adr(4)][arg(4)]...} [ETX]
[sum(2)][CR][LF]
```

響應：

```
[STX][sta(2)][sub(2)]['X'] [rc(2)] [ETX] [sum(2)] [CR] [LF]
['00'] → 重複登錄('00' 固定)
[adr(4)] → 起始數據地址、1個數據地址(BCX)
[arg(4)] → 1個數據地址中存儲的數據(最大25)(BCX)
```

WU命令有以下的結束代碼。

代碼	名稱	內容
['10']	參數異常	重複登錄中指定了'00'以外的值 含有數字('0' ~ '9'、'A' ~ 'F')以外的數據
['13']	命令執行異常	內部傳送發生超時 ISaGRA的執行周期設定過短 向不存在的CTRL模件送信
['21']	地址異常	ISaGRAF運行停止狀態下訪問了變量 (1000h以前的地址) 訪問了參數不存在的地址
	訪問數據型異常	訪問了本命令(1000h以前的地址)不支持的文字列型
['22']	寫入數據範圍外	向指定地址寫入的數據的類型不符 日曆參數中寫入格式錯誤 指定地址中寫入的數據超過了上下限的範圍
['23']	數據寫入不可	向禁止寫入參數中寫入了數據 在運行中向訪問禁止的參數寫入了數據
	運算參數異常	處於訪問禁止狀態(更新中等)時訪問了運算參數
['40']	寫入字數異常	寫入個數過多或為0
['80']	CPL Ack 響應	當前次的信息仍處理處理中時，再次發送了與前次相同的信息，用「現在處理中」表示
['99']	未定義命令	發送了不支持的命令 (要發送'WU'，但書寫錯誤)

❗ 使用上的注意事項

- 在與本公司產機器連接時，請使用16位系命令。
- 使用16位系命令時，請務必用「整數轉換嚮導」，製作16位系命令可訪問的數據。

RS

■ RS命令(ASCII10進制連續數據讀出(16位系))

從指定起始地址讀出指定數量的16位參數。最大讀出數據個數為50個。

命令:

```
[STX][sta(2)][sub(2)]['X']['R']['S'][' ',' '][adr(a)]['W'][' ',' '][len(b)]
[ETX][sum(2)][CR][LF]
```

響應:

```
[STX][sta(2)][sub(2)]['X'][rc(2)][' ',' '][data(c)][' ',' '][ETX]
[sum(2)][CR][LF]
```

[adr(a)] → 起始數據地址(32位)0~4294967295(BCD)
 [len(b)] → 數據個數(最大50)(BCD)
 [data(c)] → 1個數據地址的數據 帶符號16位(BCD)

RS命令中有以下的結束代碼。

代碼	名稱	內容
['10']	參數異常	含數字('0'~'9'、'A'~'F')以外的數據
['13']	命令執行異常	內部傳送發生超時 ISaGRA的執行周期設定過短 向不存在的CTRL模件送信
['21']	地址異常	ISaGRAF運行停止狀態下訪問了變量 (10000h以前的地址) 訪問了參數不存在的地址
	訪問數據型異常	訪問了本命令(10000h以前的地址)不支持的文字列型
['22']	讀出數據範圍外	讀出數據超過了-32768~+32767的範圍 該數據轉換成最大值(32767:7FFFh)或最小值(-32768:8000h)後讀出
['23']	運算參數異常	處於訪問禁止狀態(更新中等)時訪問了運算參數
['40']	讀出字數異常	讀出個數過多或為0
['80']	CPL Ack 響應	當前次的信息仍處理處理中時,再次發送了與前次相同的信息,用「現在處理中」表示
['99']	未定義命令	發送了不支持的命令 (要發送'RS',但書寫錯誤)

❗ 使用上的注意事項

- 在與本公司產機器連接時,請使用16位系命令。
- 使用16位系命令時,請務必用「整數轉換嚮導」,製作16位系命令可訪問的數據。

■ WS命令(ASCII10進制連續數據寫入(16位系))

向指定起始地址寫入指定數量的16位參數。

最大寫入數據個數為50個。

命令：

```
[STX][sta(2)][sub(2)]['X'] ['W'] ['S'] [' , ' ] [adr(a)] ['W'] [' , ' ]
{[data(b)] [' , ' ] ..} [ETX][sum(2)][CR][LF]
```

響應：

```
[STX][sta(2)][sub(2)]['X'] [rc(2)] [ETX][sum(2)][CR][LF]
```

[adr(a)] → 起始數據地址(32位)0~4294967295(BCD)

[arg(b)] → 寫入數據(最大50個)。帶符號16位(BCD)

WS命令中有以下的結束代碼。

代碼	名稱	內容
['10']	參數異常	含數字('0'~'9'、'A'~'F')以外的數據
['13']	命令執行異常	內部傳送發生超時 ISaGRA的執行周期設定過短 向不存在的CTRL模件送信
['21']	地址異常	ISaGRAF運行停止狀態下訪問了變量 (10000h以前的地址) 訪問了參數不存在的地址
	訪問數據型異常	訪問了本命令(10000h以前的地址)不支持的文字列型
['22']	寫入數據範圍外	向指定地址寫入的數據的類型不符 日曆參數中寫入格式錯誤 指定地址中寫入的數據超過了上下限的範圍
['23']	數據寫入不可	向禁止寫入參數中寫入了數據 在運行中向訪問禁止的參數寫入了數據
	運算參數異常	處於訪問禁止狀態(更新中等)時訪問了運算參數
['40']	寫入字數異常	寫入個數過多或為0
['80']	CPL Ack 響應	當前次的信息仍處理處理中時，再次發送了與前次相同的信息，用「現在處理中」表示
['99']	未定義命令	發送了不支持的命令 (要發送'WS'，但書寫錯誤)

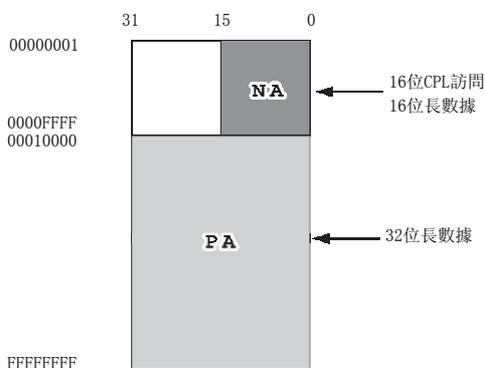
❗ 使用上的注意事項

- 在與本公司產機器連接時，請使用16位系命令。
- 使用16位系命令時，請務必用「整數轉換嚮導」，製作16位系命令可訪問的數據。

第 5 章 與EST-Z系列的連接

■ 整數轉換嚮導

本公司產顯示器智能終端EST-Z系列不支持32位系CPL命令。與EST-Z系列連接時，自動使用16位系CPL命令。



對DMC50，請在從EST-Z系列可讀出的區域(NA區域)上，準備EST訪問用的數據。

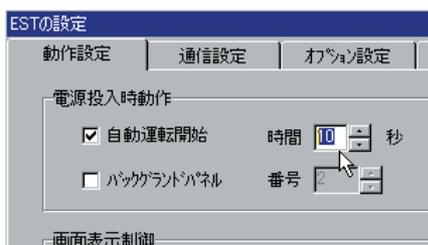
通過使用智能編程軟件包SLP-D50的「整數轉換嚮導」，可省略用於數據製作的ISaGRAF程序。

詳見

➔ 模件型數字調節器 DMC50/AHC2001 用智能編程軟件包 SLP-D50/SLP-H21 CP-SP-1122C。

■ 自動運行開始時間

由於DMC50DE的啟動時間比EST慢，請用WinAPE設定自動運行開始時間。通常要設為10~15秒左右。按[設定信息] → [基本設定] → [動作設定]的[電源投入時動作]設定。



⚠ 使用上的注意事項

與EST自動運行開始時間的設定時間比較，當DMC50的啟動時間慢の場合，EST側不等待DMC50的啟動就開始通訊。

該DMC50由於不能返回應答，EST反復重試，最終，通訊使用的命令被限制在RS命令/WS命令上。

由於這些命令的應答時間變遲，有發通訊超時的情況。

請無比設定自動運行開始時間。

第 6 章 數據地址

6 - 1 數據地址的構成

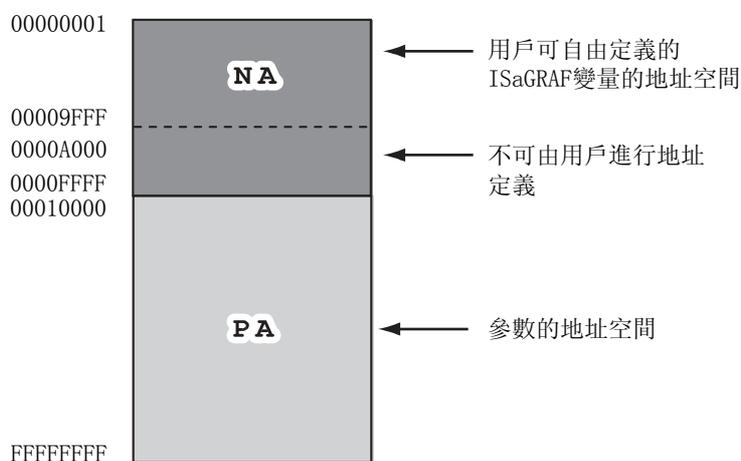
CPL通訊使用的數據地址是32位空間的地址，用16進制數標記。數據地址空間由「網絡地址(NA)」與「參數地址(PA)」的二個區域構成。

CPL通訊(Controllor Peripheral Link)是本公司上位通訊協議。

■ 網絡地址(NA)與參數地址(PA)

地址表示的數據有ISaGRAF的「變量」及DMC50專用的數據的「參數」。這些數據地址空間的區別如下。

數據地址空間	對象數據	地址名稱
00000001~0000FFFFh	變量	網絡地址(NA)
00010000~FFFFFFFh	參數	參數地址(PA)



📖 參考

- 參數有
 - 系統參數
 - 系統監視數據
 - 運算參數
 - 運算監視數據
 - 段參數
 - 用戶定義參數
- CTRL模件可由用戶編制應用程序，使用應用程序編制工具「ISaGRAF工作臺」編制。應用程序中，變量可自由定義數據。

另外，對該變量，可自由定義地址(00000001~0000FFFFh)以便通訊能訪問。把這種地址稱為「網絡地址(NA)」。

- 對COM模件，數據地址空間僅有PA區域。

■ 網絡地址(NA)的構成

網絡地址(NA)有構造如下。

$$\begin{array}{ccc} 0 & 0 & 0 & 0 & \square & \square & \square & \square & \text{h} \\ \hline \textcircled{1} & & & & \textcircled{2} & & & & \end{array}$$

① 0固定

上位16位總為0。

② 在NA應用程序上表示用戶定義的網絡地址(NA)。由16位構成，用16進制數表示。

❗ 使用上的注意事項

請勿使用A000h以後的地址，由SLP-D50/SLP-H21的自動程序功能等優先使用。

■ 參數地址(PA)的構成

參數地址(PA)構成如下。

$$\begin{array}{ccc} \square & \square & \square & \square & \square & \square & \text{h} \\ \hline \textcircled{1} & & \textcircled{2} & & \textcircled{3} & & \end{array}$$

① 參數類型ID

參數類型ID是用於識別AI設定、PID_A設定等的固有的編號。由12位構成，用16進制數表示。

② 組ID

組ID表示通道編號或PID組編號等。由12位構成，用16進制數表示。

③ 項目ID

項目ID表示項目的編號。

由8位構成，用16進制數表表示。

● 數據地址例

系統狀態(參數類型ID=0C1h)的系統報警(項目ID=1)的PA為「0C100101h」。

📖 參考

從主站側看的場合，變量及參數同樣作為數據使用。兩者的區別是僅由數據地址空間執行。

6 - 2 數據地址一覽(NA區域)

NA區域的數據地址空間為「1~0000FFFFh」。

實際使用的數據地址空間依存於由用戶編制的應用程序的變量中分配的網絡地址(NA)。

! 使用上的注意事項

- 在運行模式時才可訪問NA區域。停止模式時不能訪問。
- CPL通訊訪問の場合，請使用對象變量分配的網絡地址。網絡地址的分配由ISaGRAF工作台的字典編輯器進行。
- 請勿使用A000h以後的地址，由SLP-D50/SLP-H21的自動程序功能等優先使用。
- NA區域僅存在於CTRL模件，COM模件無。

6 - 3 數據地址一覽 (PA區域)

數據地址的一覽表如下。

- 順序按參數類型ID順序。
- 參數類型ID、組ID用16進制數表示。
- 有關項目ID，請參閱
 模件型調節器 DMC50「硬件篇」CP-UM-5172C及模件型調節器 DMC50/AHC2001「功能塊篇」CP-SP-1130C。
- 「讀/寫」欄是CPL通訊的訪問條件。含義如下：
 ○：可能
 △：應用停止模式中可能
 ×：不可

● 系統參數/系統監視數據(CTRL模件)

數據地址空間	參數類型ID	組ID	類型標籤	讀	寫	備注
00100101~0010010D	001h	001h	H/W信息	○	×	
00200101~00200102	002h	001h	日曆時間設定	○	○	
02100101~0210040D	021h	001~004h	AI設定(高分辨率型:標準輸入用)	○	○	
02200101~0220040B	022h	001~004h	AI設定(特殊型)	○	○	
02300101~02300409	023h	001~004h	AI設定(高分辨率型:選項輸入用)	○	○	
04100101~04100206	041h	001~002h	AUX-IN設定	○	○	
04500101~04500406	045h	001~004h	A0設定	○	○	
06100101~06101003	061h	001~010h	D0設定	○	○	
07100101~07101005	071h	001~010h	TP設定	○	○	
07400101~07400104	074h	001h	齊納安全柵調整值	○	○	
0A300101~0A300101	0A3h	001h	通訊設定(前面端口)	○	×	
0C100101~0C100103	0C1h	001h	系統狀態	○	×	
0C300101~0C300107	0C3h	001h	日曆時間顯示	○	×	
0C400101~0C400131	0C4h	001h	記錄(系統報警)	○	○	
0C500101~0C500131	0C5h	001h	記錄(AI報警)	○	○	
0C600101~0C600131	0C6h	001h	記錄(AUX-IN報警)	○	○	
0E100101~0E100108	0E1h	001h	AI狀態	○	×	
0E200101~0E200102	0E2h	001h	AUX-IN狀態	○	×	
0E300101~0E300104	0E3h	001h	A0狀態	○	△	
0E500101~0E50010C	0E5h	001h	DI狀態	○	×	
0E600101~0E600110	0E6h	001h	D0狀態	○	△	
0E700101~0E700110	0E7h	001h	TP狀態	○	△	
0E800101~0E800104	0E8h	001h	齊納安全柵調整計數	○	×	
0F300101~0F300101	0F3h	001h	使用中通訊設定(前面端口)	○	×	
10300101~10300116	103h	001h	內存容量監視	○	×	

● 系統參數/系統監視數據 (COM模件)

數據 地址空間	參數 類型ID	組ID	類型標籤	讀	寫	備注
00100101~0010010D	001h	001h	H/W信息	○	×	
00200101~00200102	002h	001h	日曆時間設定	○	○	
0A100101~0A100110	0A1h	001h	通訊設定 (ME20X用)	○	×	
0A200101~0A200102	0A2h	001h	通訊設定 (MR20X用)	○	×	
0A300101~0A300101	0A3h	001h	通訊設定 (前面端口)	○	×	
0C100101~0C100103	0C1h	001h	系統狀態	○	×	
0C300101~0C300107	0C3h	001h	日曆時間顯示	○	×	
0C400101~0C400131	0C4h	001h	記錄 (系統報警)	○	○	
0F100101~0F100110	0F1h	001h	使用中通訊設定 (ME20X用)	○	×	
0F200101~0F200102	0F2h	001h	使用中通訊設定 (MR20X用)	○	×	
0F300101~0F300101	0F3h	001h	使用中通訊設定 (前面端口)	○	×	

! 使用上的注意事項

- ・ CPL通訊不能變更通訊設定 (ME20X用、MR20X用、前面端口)。設定變更請由SLP-D50/SLP-H21進行。

● 運算參數/運算監視數據

數據 地址空間	參數 類型ID	組ID	類型標籤	讀	寫	備注
20100101~201FFF0C	201h	001~FFFh	PID_A設定	○	○	
20200101~202FFF0C	202h		PID_A常數	○	○	
20300101~203FFF0B	203h		PID_A監視	○	×	
21100101~211FFF0C	211h		PID_CAS設定	○	○	
21200101~212FFF0C	212h		PID_CAS常數(主站側)	○	○	
21300101~213FFF0C	213h		PID_CAS常數(從站側)	○	○	
21400101~214FFF0B	214h		PID_CAS監視	○	×	
23400101~234FFF17	234h		Ra_PID設定	○	○	
23500101~235FFF0C	235h		Ra_PID常數	○	○	
23600101~236FFF04	236h		Ra_PID監視	○	×	
24100101~241FFF0A	241h		UP_PID設定	○	○	
24200101~242FFF0B	242h		UP_PID常數	○	○	
24300101~243FFF0C	243h		UP_PID監視	○	×	
30100101~301FFF28	301h		TBL/TBR設定	○	○	

● 段參數

數據 地址空間	參數 類型ID	組ID	類型標籤	讀	寫	備注
C0000101~C0006325	C00h	001~063h	段設定	○	○	
C0100101~C6306334	C01~C63h	001~063h	段設定	○	○	
CF100101~CF1FFF3A	CF1	001~FFFh	段FB監視	○	×	

● 用戶定義參數

數據 地址空間	參數 類型ID	組ID	類型標籤	讀	寫	備注
80100101~9FFFFFF3A	801~9FFh	001~FFFh	用戶定義的類型標籤	○	○	

❗ 使用上的注意事項

- ・ 運算參數、運算監視數據、段參數、用戶定義參數已由SLP-D50/SLP-H21下裝の場合才能訪問。未下裝の場合，數據不存在不可訪問。
- ・ COM模件中無運算參數、運算監視數據、段參數、用戶定義參數。

附 錄

■ ISaGRAF變量的數據型

ISaGRAF的變量的數據型如下。

數據型	內容	範圍
BOOL型	布爾值	0~1 (0:FALSE、1:TRUE)
DINT型	倍精度整數	-2147483647~+2147483647
REAL型	實數	大約 10^{-38} ~ 10^{38} (10進制數最大7為的精度)
TIME型	時間型數據	0~T#23h59m59s999ms
STRING型*	字符串	可變長 最大255 半角文字

* CPL通訊不能訪問。

📖 參考

- ISaGRAF變量僅是CTRL模件的數據。COM模件中無。
- 各變量的數據型由編制應用程序時在ISaGRAF工作臺的辭書編輯器上定義并確定。
- BOOL型在ISaGRAF上稱為布爾型。
- DINT型是在ISaGRAF上稱為整數型的32位帶符號的整數數據。範圍從「-2147483647」起，敬請注意。
- REAL型是在ISaGRAF上稱為實數型的IEEE754規格的32位浮點小數值(單精度)數據。
- TIME型是在ISaGRAF上稱為時間型的32位無符號整數數據(單位為ms)。
- STRING型是在ISaGRAF上稱為可變長度文字列型(或文字列型、MESSAGE型)。CPL通訊不能訪問。

■ 參數的數據型

DMC50專用數據的「參數」的數據型如下。

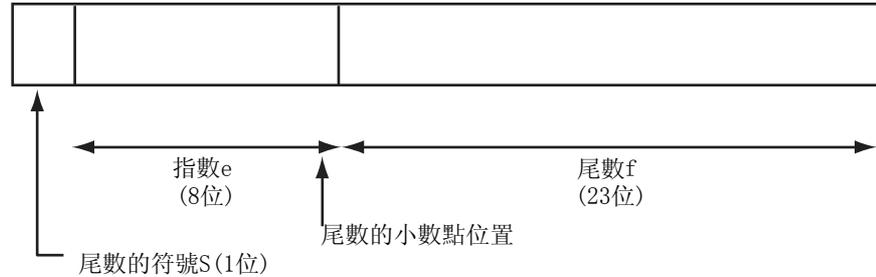
數據型	內容	範圍
BOOL型	布爾值	0~1 (0:FALSE、1:TRUE)
DINT型	倍精度整數	-2147483647~+2147483647
DWORD型	32位二進制	0~FFFFFFFFh
REAL型	實數	大約 10^{-38} ~ 10^{38} (10進制數最大7為的精度)

📖 參考

- 參數是指系統參數、運算參數、系統監視數據、運算監視數據、用戶定義參數。
- 各項目的數據型請參閱參數表。
- DINT型是32位帶符號的整數數據。範圍從「-2147483647」開始，敬請注意。
- DWORD型是32位數據。主要用於位數據等按二進制處理時使用。
- REAL型是IEEE754規格的32位浮點小數值(單精度)數據。

■ 浮點小數格式：IEEE754格式

是IEEE754(美國電氣電子技術者協會)的規格，有64位及32位。DMC50採用32位，數值用指數形式 $f \times (2^e)$ 表示，存放在各自分配的位中。(f：尾數、e：指數)
各位的含義如下。



指數是e的值加上127後的值、尾數是去除f最上位的位的1、把尾數的小數點位置向左靠後的值。(e=0時保留最上位的位)

例：10進制數的100

$$\begin{aligned}
 &100_{(10進制)} \\
 &= 64_{(16進制)} \\
 &= 1100100_{(2進制)} \\
 &= 1.1001_{(2進制)} \times (2_{(10進制)})^6_{(10進制)}
 \end{aligned}$$

指數為

$$\begin{aligned}
 &6_{(10進制)} + 127_{(10進制)} \\
 &= 133_{(10進制)} \\
 &= 10000101_{(2進制)}
 \end{aligned}$$

尾數為

$$10010000000000000000_{(2進制)}$$

$$\text{全體為 } 01000010110010000000000000000000_{(2進制)}$$

$$= 42C80000h_{(16進制)}$$

● 非數

通常，運算結果不可能為非數，外在原因(由通訊的寫入等)造成數據破損的場合，結果可能為非數。非數發生條件如下e=255、f≠0時。

- e=255, f≠0: 非數(NAN:Not A Number)
- e=255, f=0: (-s) · ∞(無限大)
- 0<e<255 : (-s) · (1.f) · 2^{e+127}(正規化數)
- e=255, f≠0: (-s) · (0.f) · 2^{e-126}(非正規化數)
- e=255, f=0: (-s) · 0(零)

■ 16位系命令的動作

命令	項 目		內 容	
RD、RU	數據地址範圍		僅NA: 0001h~FFFFh	
	動作	整數型數據的讀出	把指定地址的數據轉換成16位 超過-32768~+32767的範圍的數據時，為錯誤22	
		實數型數據的讀出 (使用禁止)	不能讀出實數數據 讀出實數數據後(使用禁止)值為不定，不能正確讀出	
		例(整數型時)	DMC50內部 → 通訊槓上 FFFF8000h(-32768) → 8000h 00007000h(28672) → 7000h 00008000h(32768) → 7FFFh(由於超過範圍，錯誤22) FFFF7000h(-36864) → 8000h(由於超過範圍，錯誤22)	
WD、WU	數據地址範圍		僅NA: 0001h~FFFFh	
	動作	整數型數據的寫入	寫入數據超過-32768~+32767的範圍時，為錯誤10，不能寫入	
		實數型數據的寫入 (使用禁止)	不能寫入實數數據 DMC50本體的數據為實數時，寫入了不定值，請絕對避免這種情況	
		例(整數型時)	通訊槓上 → DMC50 內部 8000h(-32768) → FFFF8000h 7000h(28672) → 00007000h	
RS	數據地址範圍		NA, PA: 1W~4294967295W(000000001h~FFFFFFFFh)	
	動作	NA區域 讀出 (00000001h~ 0000FFFFh)時	整數型數據的 讀出	指定地址的數據轉換成16位 超過-32768~+32767範圍的數據，為錯誤22
			實數型數據的 讀出(使用禁止)	不能讀出實數數據。讀出實數數據後，值為不定值，不能正確讀出
			例(整數型時)	DMC50內部 → 通訊槓上 FFFF8000h(-32768) → -32768 00007000h (28672) → 28672 00008000h (32768) → 32767(由於超過範圍，錯誤22) FFFF7000h(-36864) → -32768(由於超過範圍，錯誤22)
		PA區域 讀出 (00010000h~ FFFFFFFFh)時	整數型數據的 讀出	指定地址的數據進行整數型/實數型判斷，轉換成16位 超過-32768~+32767的範圍，為數據錯誤22
			實數型數據的 讀出	指定地址的數據進行整數型/實數型判斷，轉換成16位 超過-32768~+32767的範圍，為數據錯誤22 實數型的場合，按固定小數點位置下1位讀出
			例(實數型時)	DMC50內部 → 通訊槓上 C4FA0000h(-2000.0000) → -20000 44FA0000h(2000.0000) → 20000

命令	項 目		內 容	
WS	數據地址範圍		NA, PA: 1W~4294967295W (000000001h~FFFFFFFh)	
	動作	NA區域 寫入 (00000001h~ 0000FFFFh)時	整數型數據的 寫入	指定地址的數據轉換成16位 超過-32768~+32767的範圍，為數據錯誤10
			實數型數據的 寫入 (使用禁止)	不能寫入實數數據 DMC50本體的數據為實數時，寫入了不定值 請絕對避免
			例(整數型時)	通訊槓上→DMC50內部 -32768→FFFF8000h 28672→00007000h
	動作	PA區域 寫入 (00010000h~ FFFFFFFh)	整數型數據的 寫入	指定地址的數據進行整數型/實數型判斷，轉換成16位 超過-32768~+32767的範圍，為數據錯誤10
			實數型數據的 寫入	指定地址的數據進行整數型/實數型判斷，轉換成16位 超過-32768~+32767的範圍，為數據錯誤10 實數型的場合小數點位置按下1位固定寫入
			例(實數型時)	通訊槓上 → DMC50內部 10000 → 447A0000h (1000.0000) - 10000 → C47A0000h (-1000.0000)

azbil

本資料所記內容如有變更恕不另行通知

2008年3月 中文第2版(07) 日文第7版