

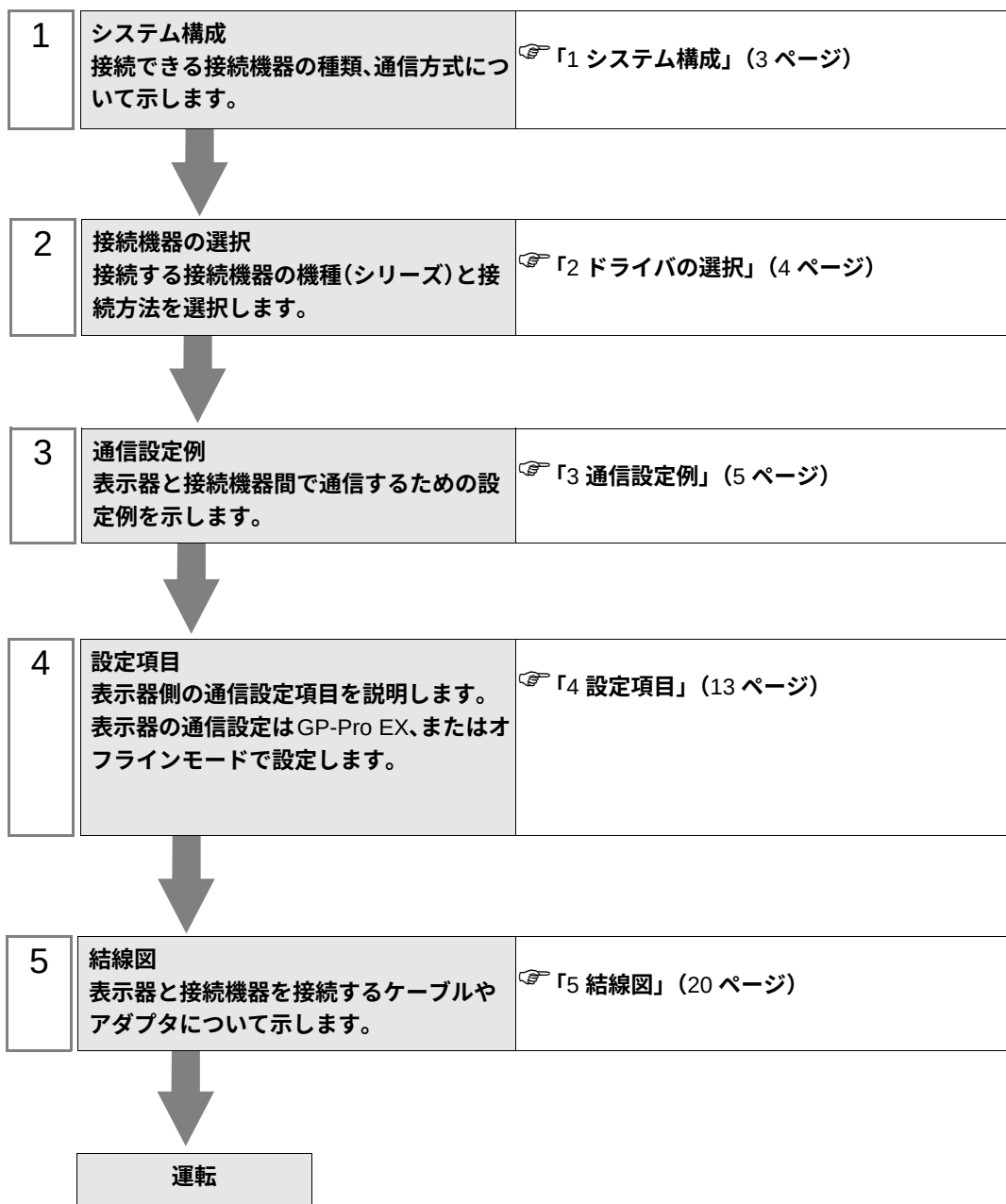
メモリリンク ドライバ

1	システム構成.....	3
2	ドライバの選択.....	4
3	通信設定例.....	5
4	設定項目.....	13
5	結線図.....	20
6	使用可能デバイス.....	40
7	デバイスコードとアドレスコード.....	41
8	エラーメッセージ.....	42
9	メモリリンクコマンド（シリアル通信）.....	43
10	サンプルプログラム（シリアル通信）.....	61
11	メモリリンクコマンド（イーサネット通信）.....	75
12	メモリリンク API（イーサネット通信）.....	89
13	サンプルプログラム（イーサネット通信）.....	121

はじめに

本書は表示器と接続機器（対象 PLC）を接続する方法について説明します。

本書では接続方法を以下の順に説明します。

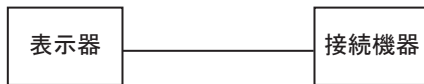


1 システム構成

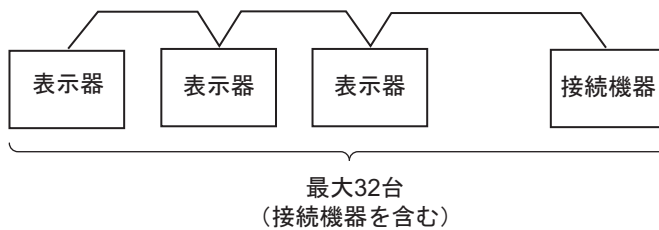
メモリリンクを使用する場合、接続構成は以下のようになります。

■ シリアル

- 1:1 接続



- 1:n 接続

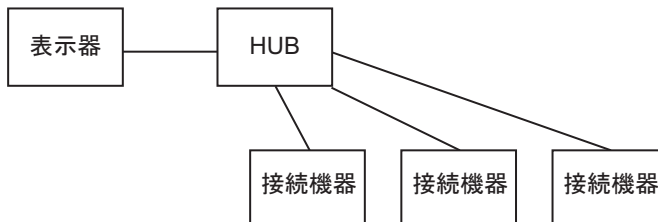


■ イーサネット

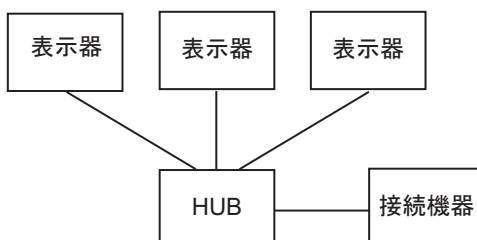
- 1:1 接続



- 1:n 接続

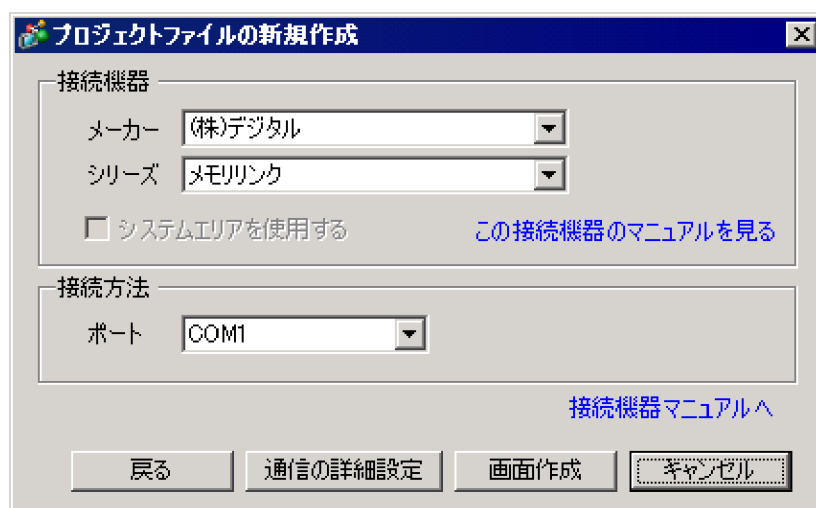


- n:1 接続 (「イーサネット (UDP)」を選択した場合のみ可能です。)



2 ドライバの選択

表示器と接続する接続機器を選択します。



設定項目	設定内容
メーカー	接続する接続機器のメーカーを選択します。「(株)デジタル」を選択します。
シリーズ	接続する接続機器の種類（シリーズ）と接続方法を選択します。「メモリリンク」を選択します。 「メモリリンク」の接続構成はシステム構成で確認してください。 👉「1 システム構成」(3 ページ)
ポート	接続機器と接続する表示器のポートを選択します。

3 通信設定例

(株) デジタルが推奨する表示器と接続機器の通信設定例を示します。

3.1 設定例 1

■ GP-Pro EX の設定 (RS232C 接続：互換モード)

◆ 通信設定

設定画面を表示するには、ワークスペースの [システム設定ウィンドウ] から [接続機器設定] を選択します。

接続機器1

概要

メーカー (株) デジタル シリーズ メモリリンク ポート COM1

文字列データモード 1 変更

通信設定

通信方式 ☒ RS232C ☐ RS422/485(2線式) ☐ RS422/485(4線式)

通信速度 9600

データ長 ☐ 7 ☒ 8

パリティ ☒ なし ☐ 偶数 ☐ 奇数

ストップビット ☒ 1 ☐ 2

フロー制御 ☐ なし ☐ ER(DTR/CTS) ☒ XON/XOFF

送信ウェイト 0 (ms)

通信プロトコル ☒ 互換モード ☐ 拡張モード

拡張モード設定

号機 No. 1

通信形式 1:1 アスキー

ターミネータ ☒ CRLF ☐ CR

☐ ETX サムチェック

☐ ACK

☐ NAK

RI / VCC ☒ RI ☐ VCC

RS232Cの場合、9番ピンをRI(入力)にするかVCC(5V電源供給)にするかを選択できます。デジタル製RS232Cアイソレーションユニットを使用する場合は、VCCを選択してください。

初期設定

■ 接続機器の設定

接続機器の仕様に依存します。

3.2 設定例 2

■ GP-Pro EX の設定 (RS232C 接続：拡張モード)

◆ 通信設定

設定画面を表示するには、ワークスペースの[システム設定ウィンドウ]から[接続機器設定]を選択します。

接続機器1

概要

メーカー シリーズ ポート [接続機器変更](#)

文字列データモード [変更](#)

通信設定

通信方式 ☒ RS232C ☐ RS422/485(2線式) ☐ RS422/485(4線式)

通信速度

データ長 ☐ 7 ☒ 8

パリティ ☒ なし ☐ 偶数 ☐ 奇数

ストップビット ☒ 1 ☐ 2

フロー制御 ☐ なし ☐ ER(DTR/CTS) ☒ XON/XOFF

送信ウェイト (ms)

通信プロトコル ☐ 互換モード ☒ 拡張モード

拡張モード設定

号機No.

通信形式

ターミネータ ☒ CRLF ☐ CR

☐ ETX サムチェック

☐ ACK

☐ NAK

RI / VCC ☒ RI ☐ VCC

RS232Cの場合、9番ピンをRI(入力)にするかVCC(5V電源供給)にするかを選択できます。デジタル製RS232Cアイソレーションユニットを使用する場合は、VCCを選択してください。

[初期値設定](#)

■ 接続機器の設定

接続機器の仕様に依存します。

3.3 設定例 3

■ GP-Pro EX の設定 (RS422/485(2 線式) 接続：互換モード)

◆ 通信設定

設定画面を表示するには、ワークスペースの[システム設定ウィンドウ]から[接続機器設定]を選択します。

■ 接続機器の設定

接続機器の仕様に依存します。

3.4 設定例 4

■ GP-Pro EX の設定 (RS422/485(2 線式) 接続：拡張モード)

◆ 通信設定

設定画面を表示するには、ワークスペースの[システム設定ウィンドウ]から[接続機器設定]を選択します。

■ 接続機器の設定

接続機器の仕様に依存します。

3.5 設定例 5

■ GP-Pro EX の設定 (RS422/485(4 線式) 接続：互換モード)

◆ 通信設定

設定画面を表示するには、ワークスペースの[システム設定ウィンドウ]から[接続機器設定]を選択します。

接続機器1

概要

メーカー (株)デジタル シリーズ メモリリンク ポート COM1

文字列データモード 1 変更

通信設定

通信方式 ☐ RS232C ☐ RS422/485(2線式) ☒ RS422/485(4線式)

通信速度 9600

データ長 ☐ 7 ☒ 8

パリティ ☒ なし ☐ 偶数 ☐ 奇数

ストップビット ☒ 1 ☐ 2

フロー制御 ☐ なし ☐ ER(DTR/CTS) ☒ XON/XOFF

送信ウェイト 0 (ms)

通信プロトコル ☒ 互換モード ☐ 拡張モード

拡張モード設定

号機No. 0

通信形式 1:1 アスキー

ターミネータ ☒ CR/LF ☐ CR

☐ ETX サムチェック

☐ ACK

☐ NAK

RI / VCC ☒ RI ☐ VCC

RS232Cの場合、9番ピンをRI(入力)にするかVCC(5V電源供給)にするかを選択できます。デジタル製RS232Cアイソレーションユニットを使用する場合は、VCCを選択してください。

初期設定

■ 接続機器の設定

接続機器の仕様に依存します。

3.6 設定例 6

■ GP-Pro EX の設定 (RS422/485(4 線式) 接続：拡張モード)

◆ 通信設定

設定画面を表示するには、ワークスペースの[システム設定ウィンドウ]から[接続機器設定]を選択します。

■ 接続機器の設定

接続機器の仕様に依存します。

3.7 設定例 7

■ GP-Pro EX の設定 (イーサネット (UDP) 接続)

◆ 通信設定

設定画面を表示するには、ワークスペースの [システム設定ウィンドウ] から [接続機器設定] を選択します。

接続機器1

概要 [接続機器変更](#)

メーカー (株)デジタル シリーズ メモリンク ポート イーサネット(UDP)

文字列データモード 1 [変更](#)

通信設定

ポート番号 1024

送信ウェイト 0 (ms)

デマンドポーリング

☒ デマンドポーリングを使用する

ポーリング周期 20 (sec)

[初期値設定](#)

■ 接続機器の設定

接続機器の仕様に依存します。

3.8 設定例 8

■ GP-Pro EX の設定 (イーサネット (TCP) 接続)

◆ 通信設定

設定画面を表示するには、ワークスペースの [システム設定ウィンドウ] から [接続機器設定] を選択します。

接続機器1

概要 [接続機器変更](#)

メーカー (株)デジタル シリーズ メモリリンク ポート イーサネット(TCP)

文字列データモード 1 [変更](#)

通信設定

ポート番号 1024

送信ウェイト 0 (ms)

デマンドポーリング

☒ デマンドポーリングを使用する

ポーリング周期 20 (sec)

[初期値設定](#)

■ 接続機器の設定

接続機器の仕様に依存します。

4 設定項目

表示器の通信設定は GP-Pro EX、または表示器のオフラインモードで設定します。

各項目の設定は接続機器の設定と一致させる必要があります。

👉 「3 通信設定例」(5 ページ)

重要

・ 表示器の IP アドレスは、表示器のオフラインモードで設定する必要があります。

参照：保守 / トラブル解決ガイド「2.5 イーサネット設定」

4.1 シリアル接続

■ GP-Pro EX での設定項目

◆ 通信設定

設定画面を表示するには、ワークスペースの [システム設定ウィンドウ] から [接続機器設定] を選択します。

接続機器1

概要

メーカー (株)デジタル シリーズ メモリリンク ポート COM1

文字列データモード 1 変更

通信設定

通信方式 ☒ RS232C ☐ RS422/485(2線式) ☐ RS422/485(4線式)

通信速度 9600

データ長 ☐ 7 ☒ 8

パリティ ☒ なし ☐ 偶数 ☐ 奇数

ストップビット ☒ 1 ☐ 2

フロー制御 ☐ なし ☐ ER(DTR/CTS) ☒ XON/XOFF

送信ウェイト 0 (ms)

通信プロトコル ☐ 互換モード ☒ 拡張モード

拡張モード設定

機種 No. 0

通信形式 1:1 アスキー

ターミネータ ☒ CRLF ☐ CR

☐ ETX サムチェック

☐ ACK

☐ NAK

RI / VCC ☒ RI ☐ VCC

RS232Cの場合、9番ピンをRI(入力)にするかVCC(5V電源供給)にするかを選択できます。デジタル製RS232Cアイソレーションユニットを使用する場合は、VCCを選択してください。

初期設定

設定項目	設定内容
通信方式	接続機器と通信する通信方式を選択します。
通信速度	接続機器と表示器間の通信速度を選択します。
データ長	データ長を選択します。
パリティ	パリティチェックの方法を選択します。
ストップビット	ストップビット長を選択します。
フロー制御	送受信データのオーバーフローを防ぐために行う通信制御方式を選択します。
送信ウェイト	表示器がパケットを受信してから、次のコマンドを送信するまでの待機時間 (ms) を「0 ～ 255」で入力します。
通信プロトコル	通信プロトコルを選択します。
号機 No.	通信を行う接続機器の局番を「0 ～ 31」で入力します。
通信形式	通信形式に「1:1 アスキー」「1:1 バイナリ」「1:N アスキー」「1:N バイナリ」のいずれかを選択します。
ターミネータ	使用するターミネータを選択します。ターミネータは通信形式が「1:1 アスキー」「1:N アスキー」の場合のみ有効です。
ETX サムチェック	データのやり取りをする場合、サムチェックコードを付加するかどうかを設定します。
ACK	ACK を使用する場合にチェックします。
NAK	NAK を使用する場合にチェックします。
RI/VCC	通信方式で RS232C を選択した場合に、9 番ピンの RI/VCC を切り替えます。

■ オフラインモードでの設定項目

MEMO

- ・ オフラインモードへの入り方や操作方法は保守 / トラブル解決ガイドを参照してください。

参照：保守 / トラブル解決ガイド「2.2 オフラインモードについて」

◆ 通信設定

設定画面を表示するには、オフラインモードの[周辺機器設定]タブから[接続機器設定]をタッチします。表示された一覧から設定したい接続機器をタッチし、[通信設定]タブをタッチします。

(1/2 ページ)

通信設定	オプション			
メモリリンク		[COM1]		Page 1/2
通信方式	RS232C			
通信速度	9600			
データ長	☐ 7 ☒ 8			
パリティ	☒ なし ☐ 偶数 ☐ 奇数			
ストップビット	☒ 1 ☐ 2			
フロー制御	ER(DTR/CTS)			
送信ウェイト(ms)	0			
通信プロトコル	互換モード			
➡				
終了		戻る		2005/09/22 14:14:07

設定項目	設定内容
通信方式	接続機器と通信する通信方式を選択します。
通信速度	接続機器と表示器間の通信速度 (bps) を選択します。
データ長	データ長を選択します。
パリティ	パリティチェックの方法を選択します。
ストップビット	ストップビット長を選択します。
フロー制御	送受信データのオーバーフローを防ぐために行う通信制御方式を選択します。
送信ウェイト	表示器がパケットを受信してから、次のコマンドを送信するまでの待機時間 (ms) を「0～255」で入力します。
通信プロトコル	通信プロトコルに「互換モード」「拡張モード」のいずれかを選択します。

(2/2 ページ)

通信設定	オプション			
メモリリンク		[COM1]		Page 2/2
拡張モード 設定				
号機No.	0			
通信形式	1:1 アスキー			
ターミネータ	☒ CR, LF ☐ CR			
ETX, サムチェック	☒ OFF ☐ ON			
ACK	☒ OFF ☐ ON			
NAK	☒ OFF ☐ ON			
				←
終了		戻る		2005/09/22 14:14:10

設定項目	設定内容
号機 No.	通信を行う接続機器の局番を「0 ～ 31」で入力します。
通信形式	通信形式に「1:1 アスキー」「1:1 バイナリ」「1:N アスキー」「1:N バイナリ」のいずれかを選択します。
ターミネータ	使用するターミネータに「CR,LF」「CR」のいずれかを選択します。ターミネータは通信形式が「1:1 アスキー」「1:N アスキー」の場合のみ有効です。
ETX サムチェック	データのやり取りをする場合、サムチェックコードを付加するかどうかを設定します。
ACK	ACK を使用する場合にチェックします。
NAK	NAK を使用する場合にチェックします。

◆ オプション

設定画面を表示するには、オフラインモードの [周辺機器設定] から [接続機器設定] をタッチします。表示された一覧から設定したい接続機器をタッチし、[オプション] をタッチします。

通信設定	オプション			

メモリリンク [COM1] Page 1/1

RI / VCC ☒ RI ☐ VCC

RS232Cの場合、9番ピンをRI(入力)にするかVCC(5V電源供給)にするかを選択できます。デジタル製RS232Cアイソレーションユニットを使用する場合は、VCCを選択してください。

	終了		戻る	2005/09/22 14:14:13
--	----	--	----	------------------------

設定項目	設定内容
RI/VCC	通信方式で RS232C を選択した場合に、9 番ピンの RI/VCC を切り替えます。

4.2 イーサネット接続

■ GP-Pro EX での設定項目

◆ 通信設定

設定画面を表示するには、ワークスペースの[システム設定ウィンドウ]から[接続機器設定]を選択します。

接続機器1

概要

メーカー (株)デジタル シリーズ メモリリンク ポート イーサネット(UDP)

文字列データモード 1 変更

通信設定

ポート番号 1024

送信ウェイト 0 (ms)

デマンドポーリング

☒ デマンドポーリングを使用する

ポーリング周期 20 (sec)

初期設定

[接続機器変更](#)

設定項目	設定内容
ポート番号	通信を行う外部機器のポート番号を「1024～65535」で入力します。
送信ウェイト	表示器がパケットを受信してから、次のコマンドを送信するまでの待機時間 (ms) を「0～255」で入力します。
デマンドポーリングを使用する	接続機器の生存を確認するためにポーリングの催促コマンドを使用する場合にチェックをつけます。
ポーリング周期	ポーリング周期 (sec) を「10～100」で入力します。

■ オフラインモードでの設定項目

MEMO

- ・ オフラインモードへの入り方や操作方法は保守 / トラブル解決ガイドを参照してください。

参照：保守 / トラブル解決ガイド「2.2 オフラインモードについて」

◆ 通信設定

設定画面を表示するには、オフラインモードの[周辺機器設定]タブから[接続機器設定]をタッチします。表示された一覧から設定したい接続機器をタッチし、[通信設定]タブをタッチします。

通信設定				
メモリリンク		[UDP]	Page 1/1	
ポート番号 1024 ▼ ▲ 送信ウェイト(ms) 0 ▼ ▲ デマンドポーリングを使用する ☐ OFF ☒ ON ポーリング周期(s) 20 ▼ ▲				
終了		戻る		2005/09/22 14:14:17

設定項目	設定内容
ポート番号	通信を行う接続機器のポート番号を「1024～65535」で入力します。
送信ウェイト	表示器がパケットを受信してから、次のコマンドを送信するまでの待機時間(ms)を「0～255」で入力します。
デマンドポーリングを使用する	接続機器の生存を確認するためにポーリングの催促コマンドを使用する場合にチェックをつけます。
ポーリング周期	ポーリング周期(sec)を「10～100」で入力します。

5 結線図

以下に示す結線図と接続機器メーカーが推奨する結線図が異なる場合がありますが、本書に示す結線図でも動作上問題ありません。

- 外部機器本体の FG 端子は D 種接地を行ってください。詳細は外部機器のマニュアルを参照してください。
- 表示器内部で SG と FG は接続されています。接続機器と SG を接続する場合は短絡ループが形成されないようにシステムを設計してください。
- ノイズなどの影響で通信が安定しない場合はアイソレーションユニットを接続してください。
- 接続機器と IPC を RS-232C で接続する場合、使用できる COM ポートはシリーズによって異なります。詳細は IPC のマニュアルを参照してください。

使用可能ポート

シリーズ	使用可能ポート
PS-2000B	COM1 ^{*1} 、COM2、COM3 ^{※1} 、COM4
PS-3650A、PS-3651A	COM1 ^{※1}
PS-3700A (Pentium®4-M)	COM1 ^{※1} 、COM2 ^{※1} 、COM3 ^{*2} 、COM4

*1 RI/5V を切替えることができます。IPC の切替えスイッチで切替えてください。

*2 通信方式をディップスイッチで設定する必要があります。

- 接続機器と PS-3700A(Pentium®4-M) の COM3 を接続する場合、COM3 の通信方式をディップスイッチで設定する必要があります。詳細は PS-3700A(Pentium®4-M) のマニュアルを参照してください。

ディップスイッチの設定：RS-232C

ディップスイッチ	設定値	設定内容
1	OFF	予約（常時 OFF）
2	OFF	COM3 の通信方式：RS-232C
3	OFF	
4	OFF	TX データの出力モード：常に出力
5	OFF	TX への終端抵抗挿入 (220Ω)：なし
6	OFF	RX への終端抵抗挿入 (220Ω)：なし
7	OFF	TXA と RXA の短絡：しない
8	OFF	TXB と RXB の短絡：しない
9	OFF	Auto Detection：無効
10	OFF	

ディップスイッチの設定：RS-422/485（4 線式）

ディップスイッチ	設定値	設定内容
1	OFF	予約（常時 OFF）
2	ON	COM3 の通信方式：RS-422/485
3	ON	
4	OFF	TX データの出力モード：常に出力
5	OFF	TX への終端抵抗挿入 (220Ω)：なし
6	OFF	RX への終端抵抗挿入 (220Ω)：なし
7	OFF	TXA と RXA の短絡：しない
8	OFF	TXB と RXB の短絡：しない
9	OFF	Auto Detection：無効
10	OFF	

ディップスイッチの設定：RS-422/485（2 線式）

ディップスイッチ	設定値	設定内容
1	OFF	予約（常時 OFF）
2	ON	COM3 の通信方式：RS-422/485
3	ON	
4	OFF	TX データの出力モード：常に出力
5	OFF	TX への終端抵抗挿入 (220Ω)：なし
6	OFF	RX への終端抵抗挿入 (220Ω)：なし
7	ON	TXA と RXA の短絡：する
8	ON	TXB と RXB の短絡：する
9	ON	Auto Detection：有効
10	ON	

結線図 1

表示器（接続ポート）	ケーブル		備考
GP（COM1） IPC ^{*1*2}	A	自作ケーブル（ER 制御の場合）	ケーブル長は 15m 以内にしてください。
	B	自作ケーブル（X 制御の場合）	
	C	自作ケーブル（制御方式なしの場合）	

*1 使用できるポートはシリーズによって異なります。

☞ 使用可能ポート（20 ページ）

*2 PS-3700A(Pentium®4-M) の COM3 を使用する場合、COM3 の通信方式をディップスイッチで設定する必要があります。

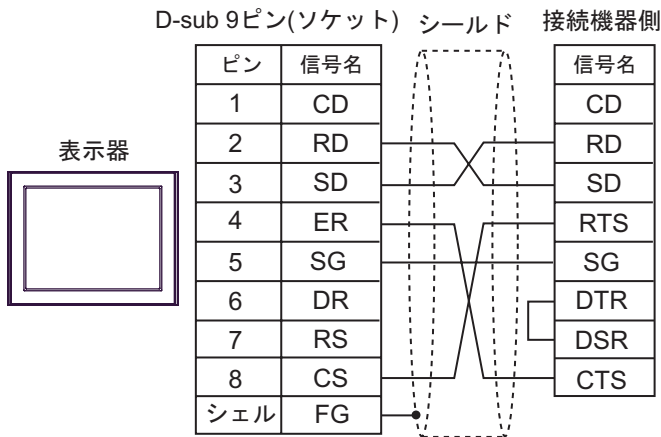
☞ ディップスイッチの設定：RS-232C（20 ページ）

重 要

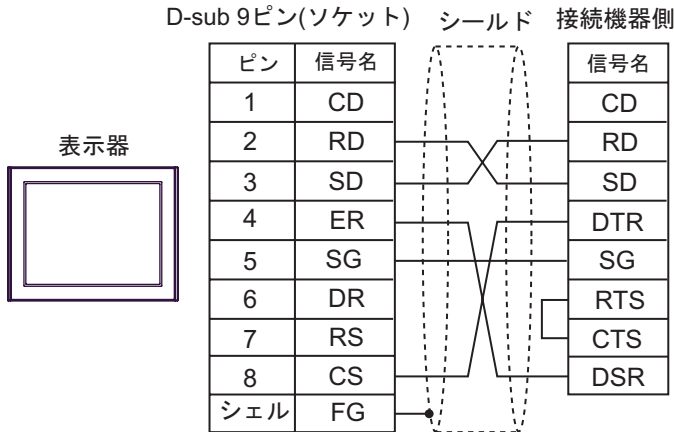
・ ホストによって、RS232C コネクタの形状やピン番号と信号名の対応が異なります。
ホストのインターフェイス仕様にしたがって、正しく接続してください。

A) 自作ケーブルを使用する場合

- ・ 接続機器が RTS/CTS 制御の場合



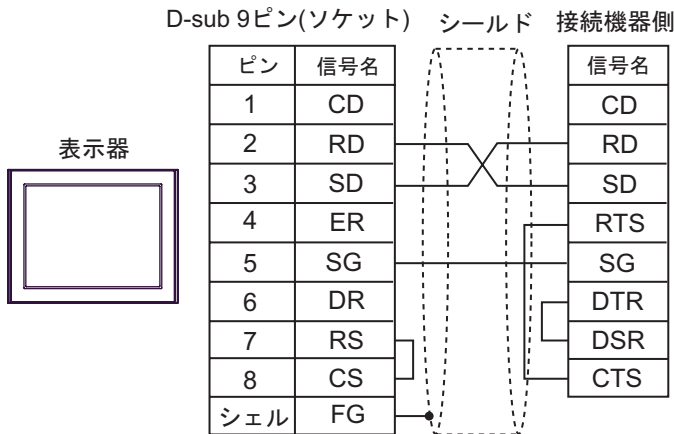
- 接続機器が DTR/DSR 制御の場合



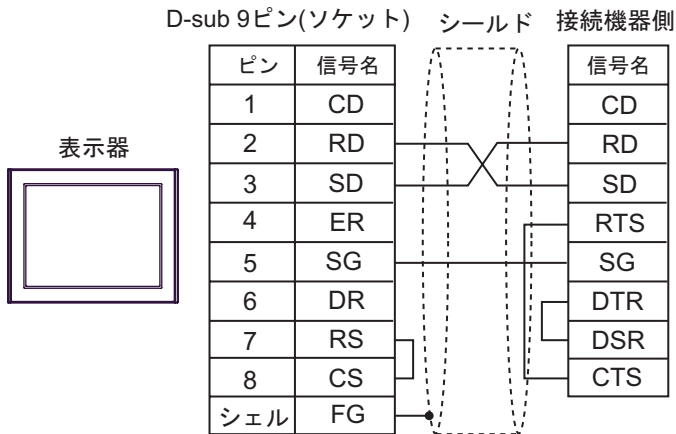
禁止：

- GP の ER が OFF のとき、ホスト側は送信しないようにしてください。

B) 自作ケーブルを使用する場合（X 制御の場合）



C) 自作ケーブルを使用する場合（制御方式なしの場合）



結線図 2

表示器 (接続ポート)	ケーブル		備考
GP ^{*1} (COM1) AGP-3302B (COM2) IPC ^{*2} (COM3)	A	(株) デジタル製 COM ポート変換アダプタ CA3-ADPCOM-01 + (株) デジタル製 コネクタ端子台変換アダプタ CA3-ADPTRM-01 + 自作ケーブル	
	B	(株) デジタル製 COM ポート変換アダプタ CA3-ADPCOM-01 + (株) デジタル製 AGP 用 422 ケーブル CA3-CBL422-01	
	C	(株) デジタル製 COM ポート変換アダプタ CA3-ADPCOM-01 + (株) デジタル製 AGP 用マルチリンクケーブル CA3-CBLMLT-01	
	D	自作ケーブル	
GP ^{*3} (COM2)	E	(株) デジタル製 オンラインアダプタ CA4-ADPONL-01 + (株) デジタル製 コネクタ端子台変換アダプタ CA3-ADPTRM-01 + 自作ケーブル	
	F	(株) デジタル製 オンラインアダプタ CA4-ADPONL-01 + (株) デジタル製 AGP 用 422 ケーブル CA3-CBL422-01	
	G	(株) デジタル製 オンラインアダプタ CA4-ADPONL-01 + (株) デジタル製 AGP 用マルチリンクケーブル CA3-CBLMLT-01	
	H	(株) デジタル製 オンラインアダプタ CA4-ADPONL-01 + 自作ケーブル	

*1 AGP-3302B 除く全 GP 機種

*2 PS-3700A(Pentium®4-M) の COM3 のみ使用できます。COM3 を使用する場合、COM3 の通信方式をディップスイッチで設定する必要があります。

 **ディップスイッチの設定：RS-422/485 (4 線式) (21 ページ)**

*3 GP-3200 シリーズおよび AGP-3302B を除く全 GP 機種

MEMO

- RS422 ケーブルを使用した場合の制御方法は XON/XOFF 制御のみです。また、XON/XOFF は ASCII の場合のみ有効です。

強制：

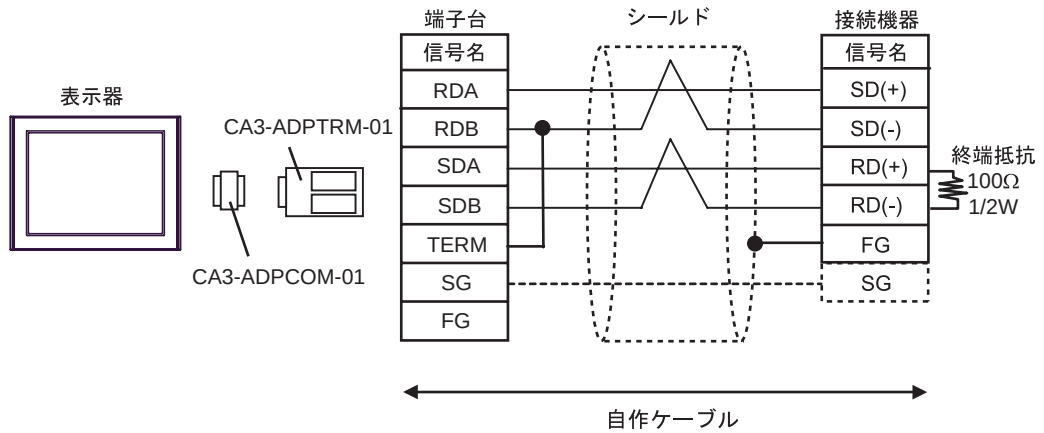
- 24AWG 線材を使用した、静電容量 50pF/m 程度、特性インピーダンス 100Ω 程度のツイストペアケーブルを使用してください。

重 要

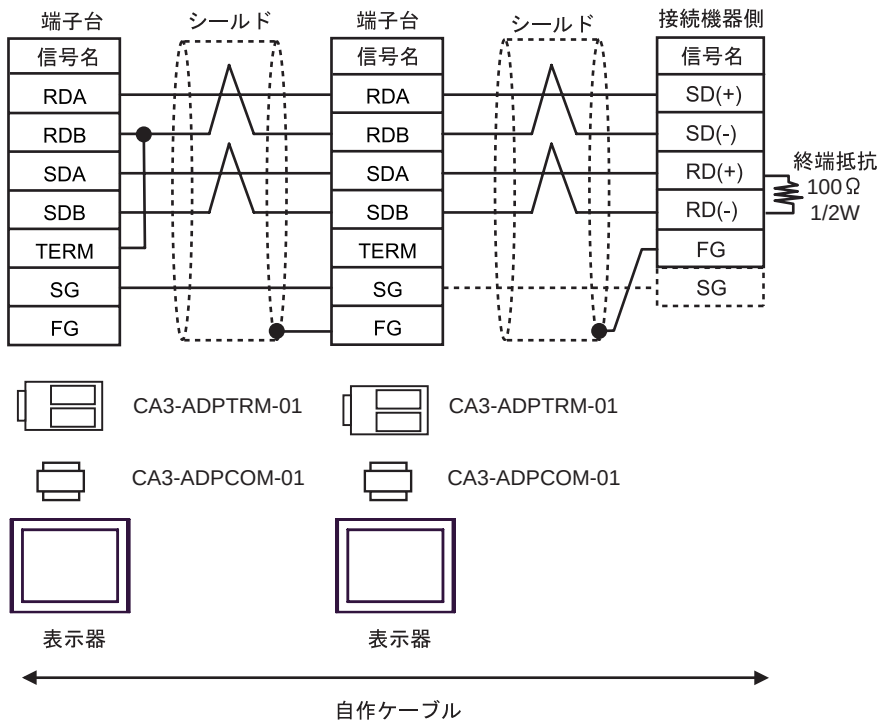
- 通常 RS422 ケーブルの最大長は 1000m ですが、接続するホストによって制限があります。接続の際には、必ず接続するホストのマニュアルをご参照ください。
- 接続するホストによって、接続のしかたや終端抵抗などが異なります。
- GP 側はアイソレーションをされていません。
- GP 同士は SG を必ず接続してください。
- 接続機器と GP の間は、接続機器がアイソレーションされている場合は、SG を接続しなくてもよい場合があります。

A) (株) デジタル製 COM ポート変換アダプタ (CA3-ADPCOM-01) および (株) デジタル製 コネクタ端子台変換アダプタ (CA3-ADPTRM-01)、自作ケーブルを使用する場合

- 1:1 接続の場合



1:n 接続の場合

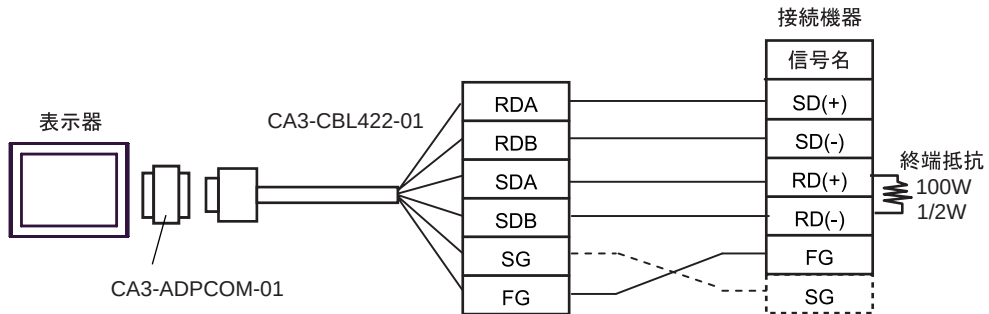


MEMO

- CA3-ADPTRM-01 の RDB と TERM を接続することにより、GP 側 RDA と RDB の間に 100Ω 1/2W の終端抵抗が挿入されます。

B) (株) デジタル製 COM ポート変換アダプタ (CA3-ADPCOM-01) および (株) デジタル製 AGP 用 422 ケーブル (CA3-CBL422-01)、自作ケーブルを使用する場合

- 1:1 接続の場合

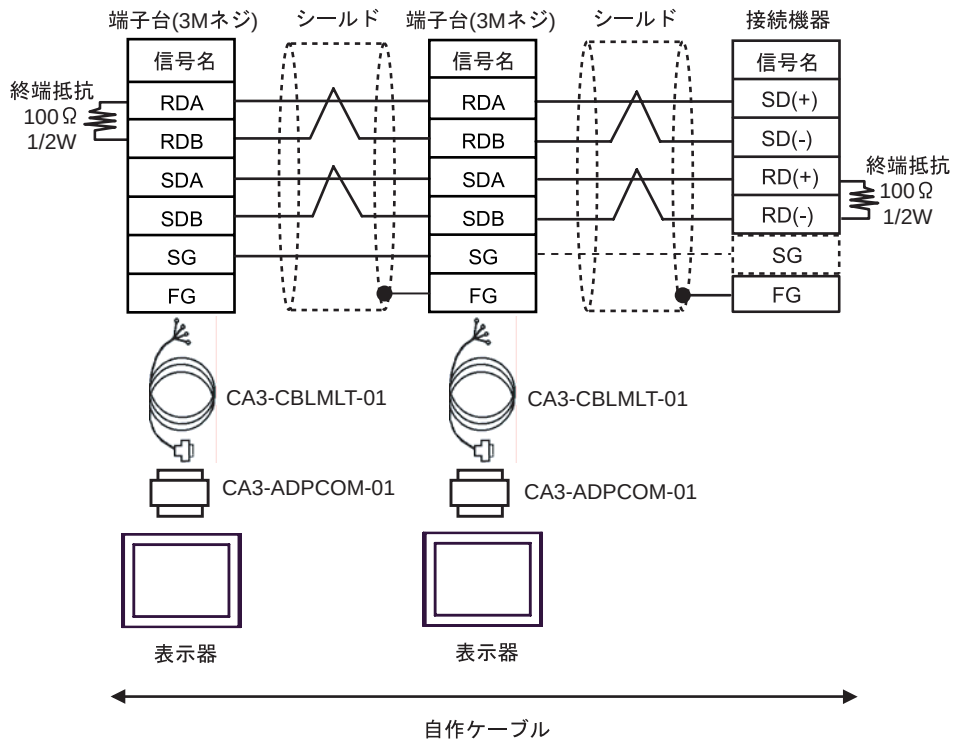


MEMO

- CA3-CBL422-01 には RDA-RDB 間に 100Ω の終端抵抗が挿入されています。

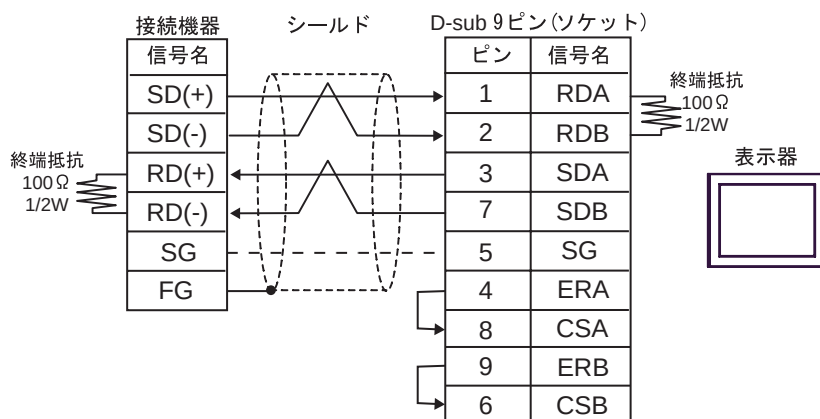
C) (株) デジタル製 COM ポート変換アダプタおよび (株) デジタル製 AGP 用マルチリンクケーブル (CA3-CBLMLT-01)、自作ケーブルを使用する場合

- 1:n 接続の場合

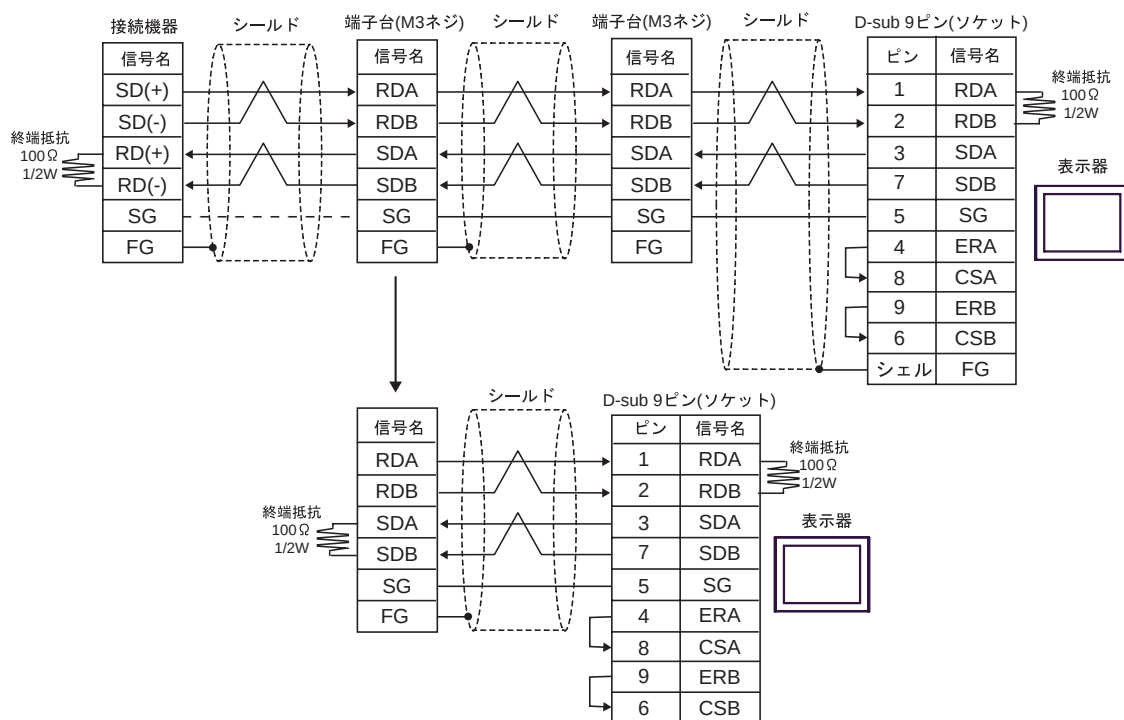


D) 自作ケーブルを使用する場合

- 1:1 接続の場合

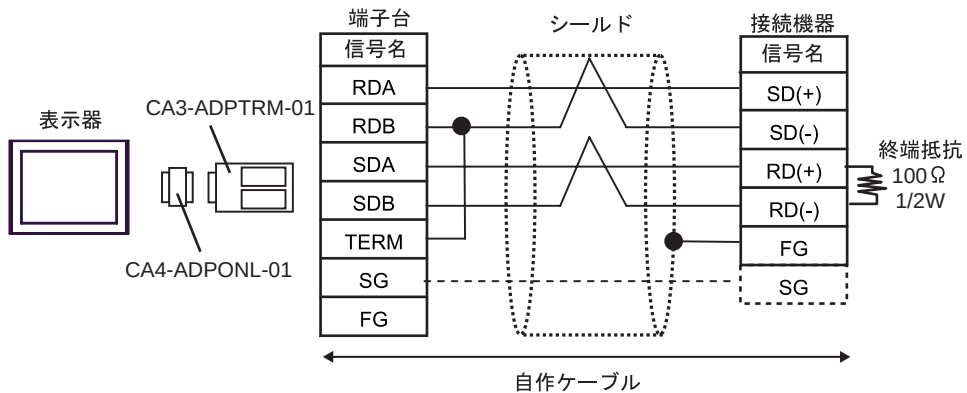


- 1:n 接続の場合

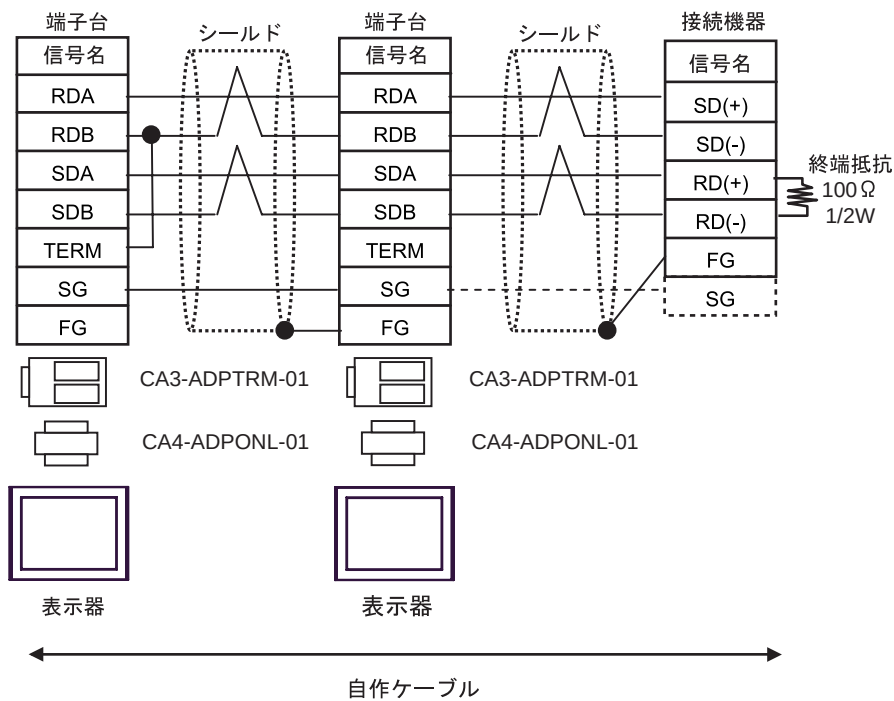


E) (株) デジタル製オンラインアダプタ (CA4-ADPONL-01) および (株) デジタル製 コネクタ端子台変換アダプタ (CA3-ADPTRM-01)、自作ケーブルを使用する場合

- 1:1 接続の場合



- 1:n 接続の場合

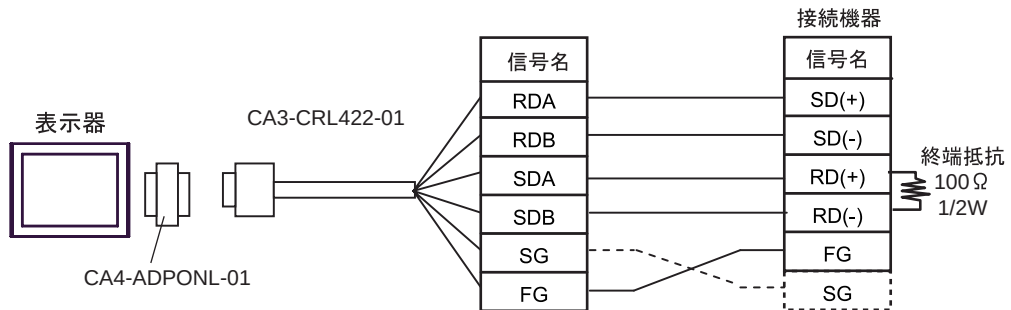


MEMO

- CA3-ADPTRM-01 の RDA と TERM を接続することにより、GP 側 RDA-RDB 間に 100Ω の終端抵抗が挿入されます。

F) (株) デジタル製オンラインアダプタ (CA4-ADPONL-01) および (株) デジタル製 AGP 用 422 ケーブル (CA3-CBL422-01) を使用する場合

- 1:1 接続の場合

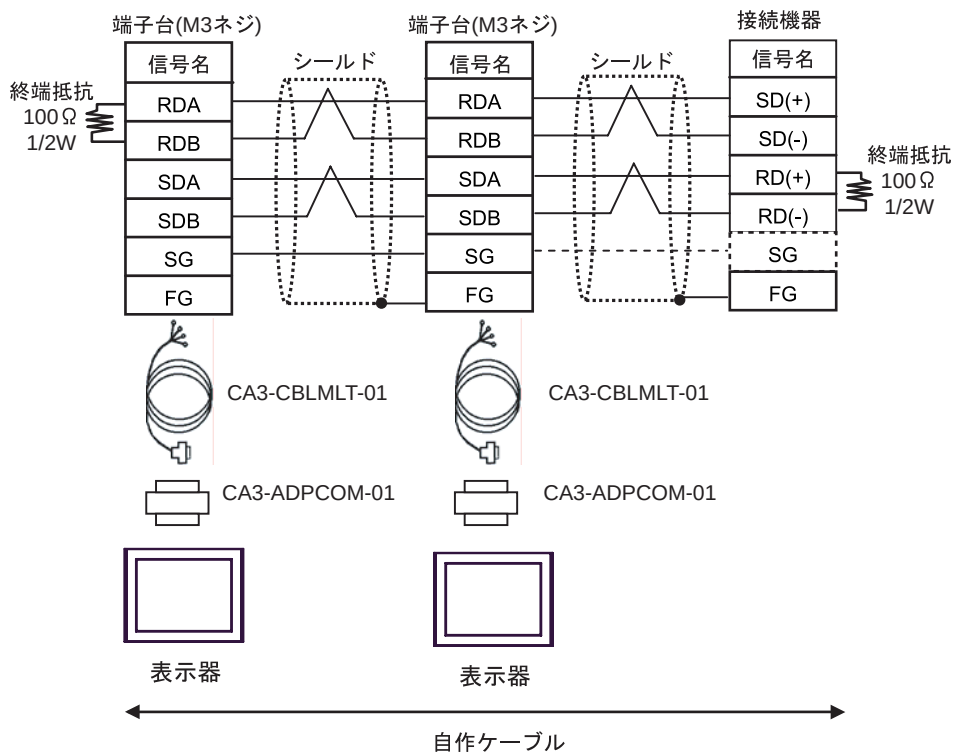


MEMO

- CA3-CBL422-01 には RDA-RDB 間に 100Ω の終端抵抗が挿入されています。

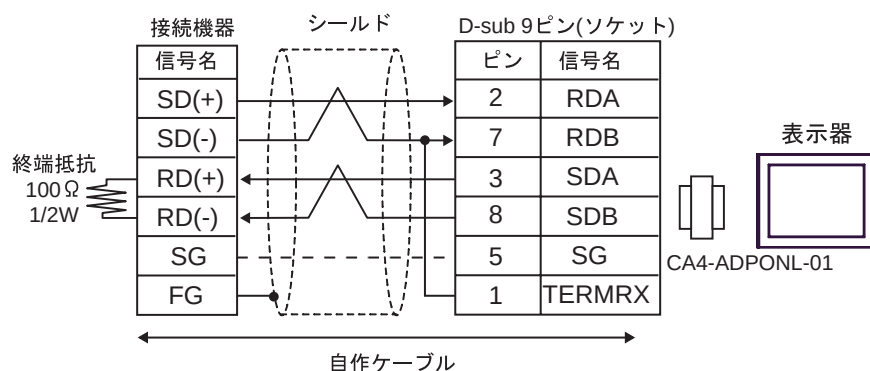
G) (株) デジタル製オンラインアダプタ (CA4-ADPONL-01) および (株) デジタル製 AGP 用マルチリンクケーブル (CA3-CBLMLT-01)、自作ケーブルを使用する場合

- 1:n 接続の場合

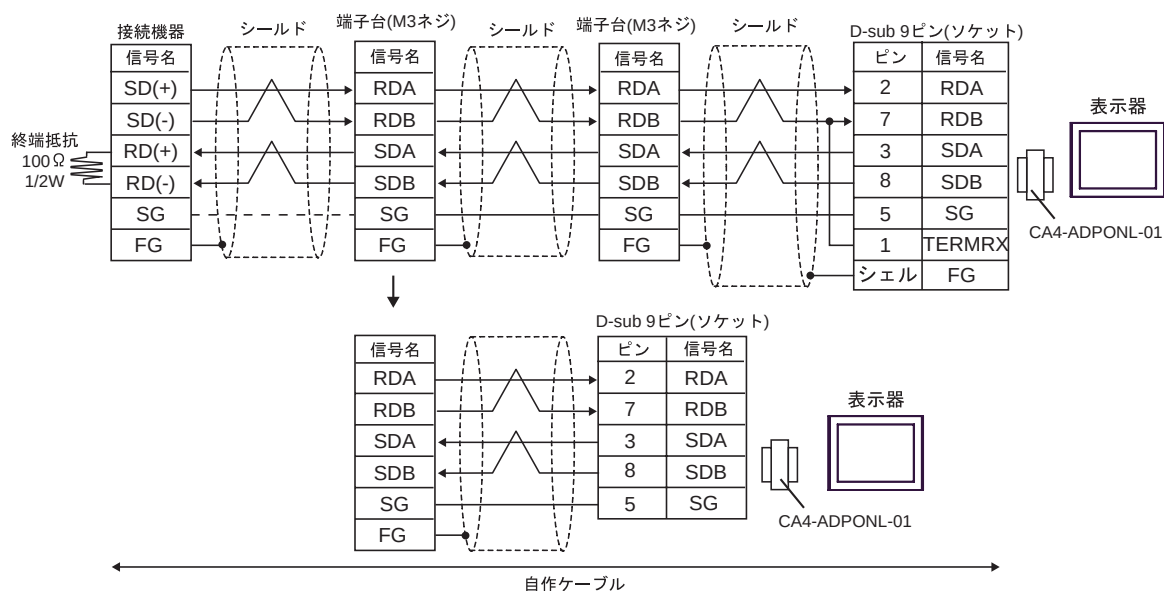


H) (株) デジタル製オンラインアダプタ (CA4-ADPONL-01) + 自作ケーブルを使用する場合

- 1:1 接続の場合



- 1:n 接続の場合



結線図 3 (RS-422 (2 線式) 接続)

表示器 (接続ポート)	ケーブル		備考
GP ^{*1} (COM1) AGP-3302B (COM2)	A	(株) デジタル製 COM ポート変換アダプタ CA3-ADPCOM-01 + (株) デジタル製 コネクタ端子台変換アダプタ CA3-ADPTRM-01 + 自作ケーブル	
	B	自作ケーブル	
GP ^{*2} (COM2)	C	(株) デジタル製オンラインアダプタ CA4-ADPONL-01 + (株) デジタル製 コネクタ端子台変換アダプタ CA3-ADPTRM-01 + 自作ケーブル	
	D	(株) デジタル製オンラインアダプタ CA4-ADPONL-01 + 自作ケーブル	
IPC ^{*3} (COM3)	E	(株) デジタル製 COM ポート変換アダプタ CA3-ADPCOM-01 + (株) デジタル製 コネクタ端子台変換アダプタ CA3-ADPTRM-01 + 自作ケーブル	
	F	自作ケーブル	

*1 AGP-3302B を除く全 GP 機種

*2 GP-3200 シリーズおよび AGP-3302B を除く全 GP 機種

*3 PS-3700A(Pentium®4-M) の COM3 のみ使用できます。COM3 を使用する場合、COM3 の通信方式をディップスイッチで設定する必要があります。

 **ディップスイッチの設定：RS-422/485 (2 線式) (21 ページ)**

MEMO

- RS422 ケーブルを使用した場合の制御方法は XON/XOFF 制御のみです。また、XON/XOFF は ASCII の場合のみです。

強制：

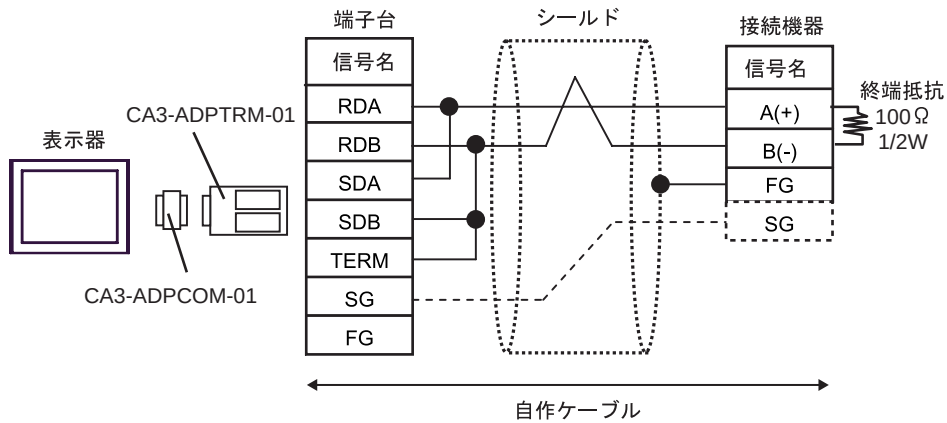
- 24AWG 線材を使用した、静電容量 50pF/m 程度、特性インピーダンス 100Ω 程度のツイストペアケーブルを使用してください。

重 要

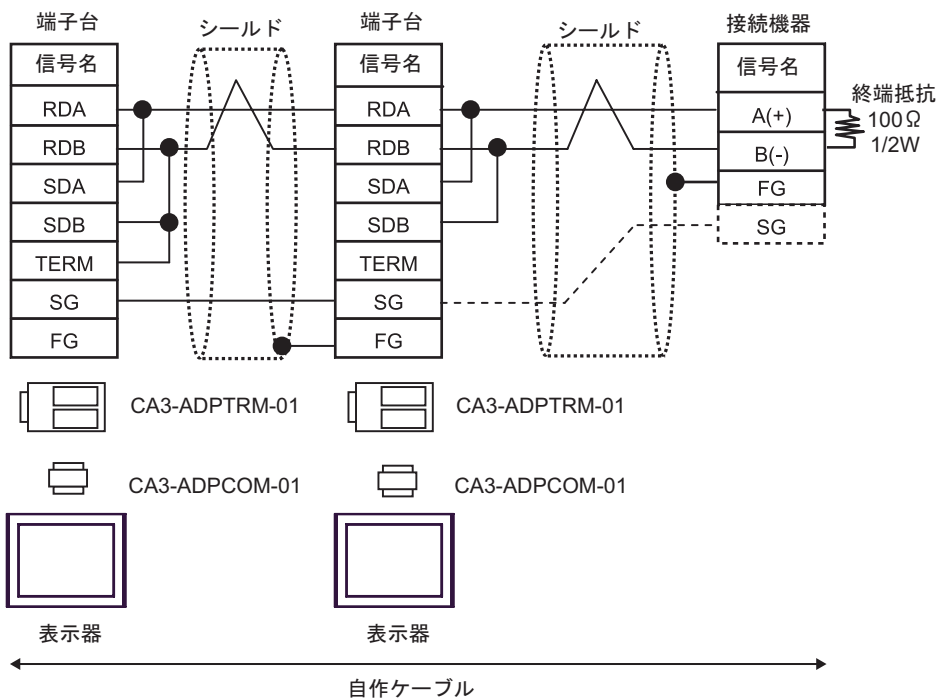
- 通常 RS422 ケーブルの最大長は 1000m ですが、接続するホストによって制限があります。接続の際には、必ず接続するホストのマニュアルをご参照ください。
- 接続するホストによって、接続のしかたや終端抵抗などが異なります。SG がある場合は接続してください。

A) (株) デジタル製 COM ポート変換アダプタ (CA3-ADPCOM-01) および (株) デジタル製 コネクタ端子台変換アダプタ (CA3-ADPTRM-01)、自作ケーブルを使用する場合

- 1:1 接続の場合



- n:1 接続の場合

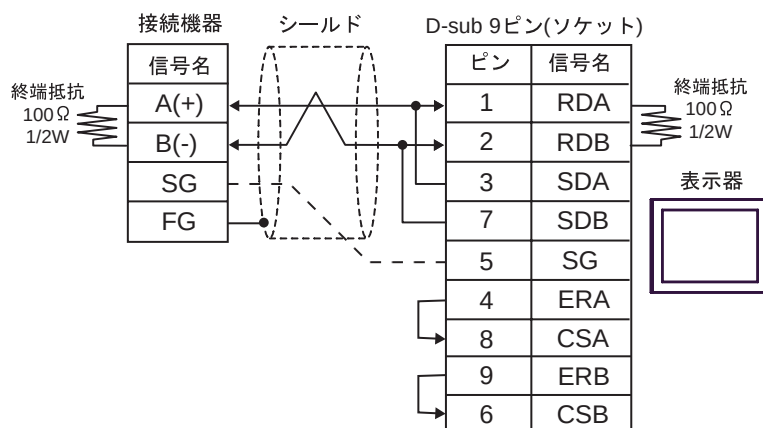


MEMO

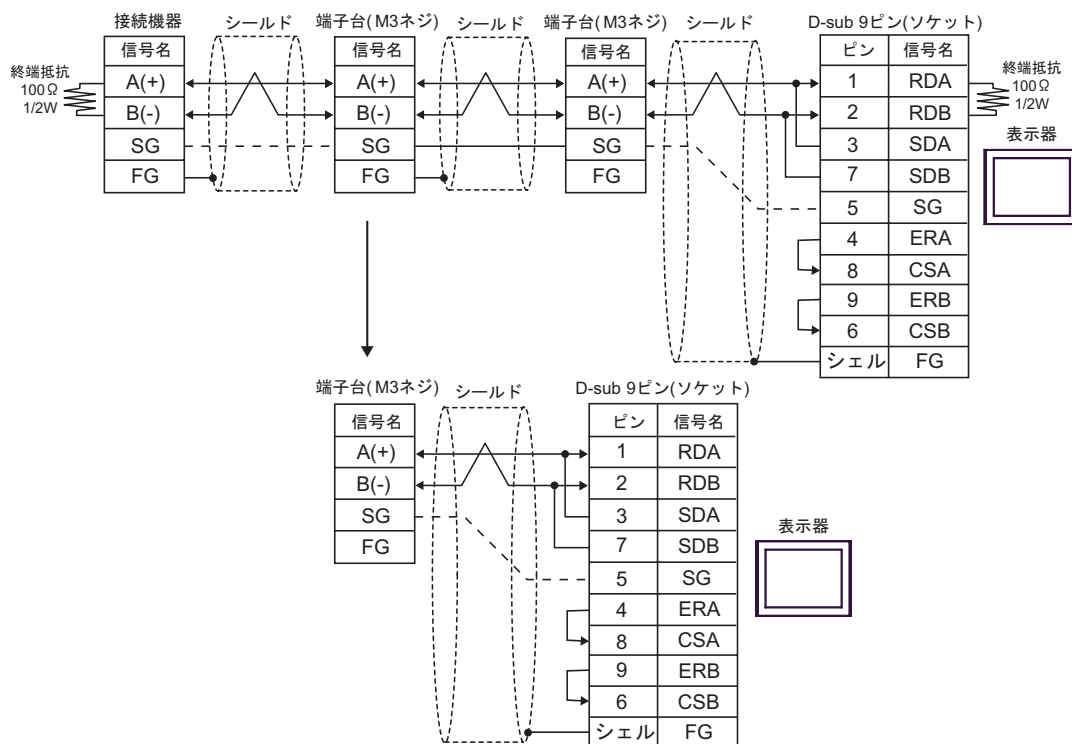
- CA3-ADPTRM-01 の RDB と TERM を接続することにより、表示器側 RDA-RDB 間に 100Ω 1/2W の終端抵抗が挿入されます。

B) 自作ケーブルを使用する場合

- 1:1 接続の場合

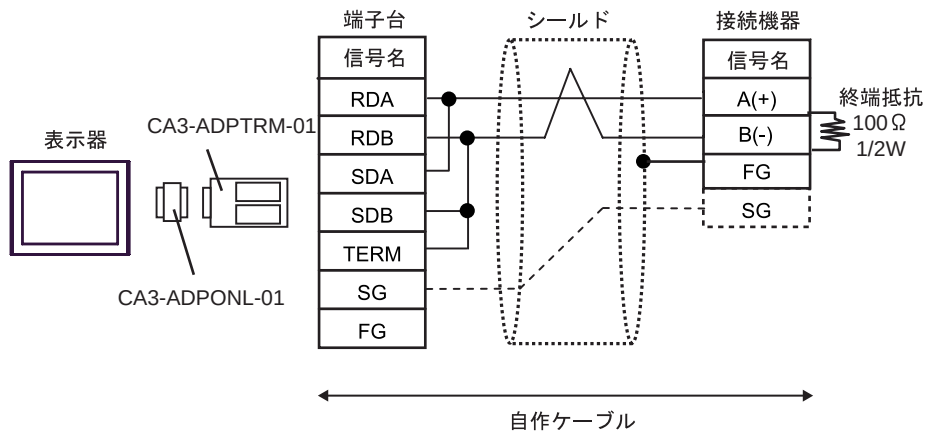


- n:1 接続の場合

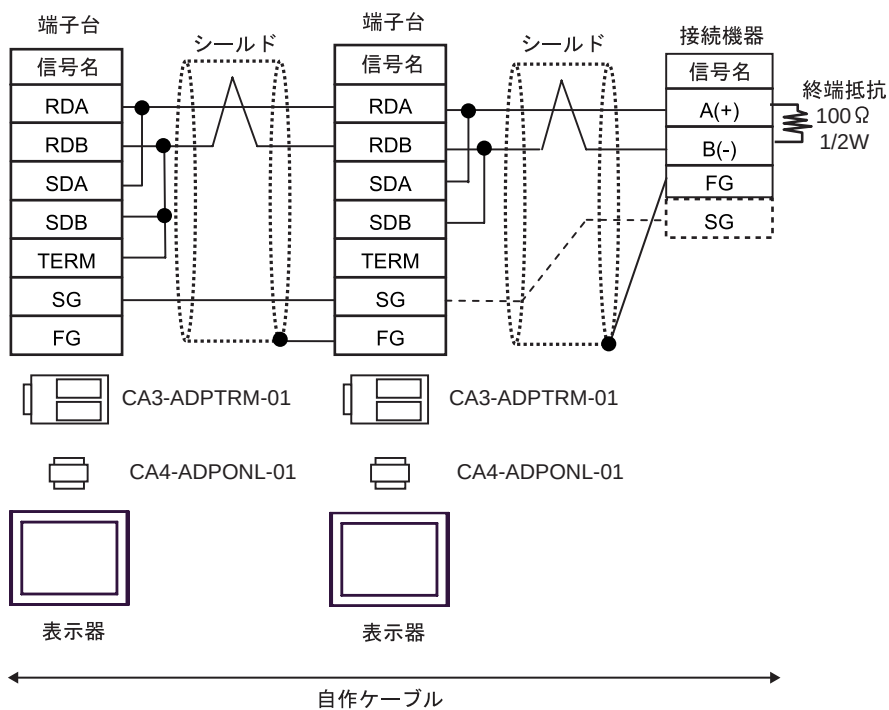


C) (株) デジタル製オンラインアダプタ (CA4-ADPONL-01) および (株) デジタル製 コネクタ端子台変換アダプタ (CA3-ADPTRM-01)、自作ケーブルを使用する場合

- 1:1 接続の場合

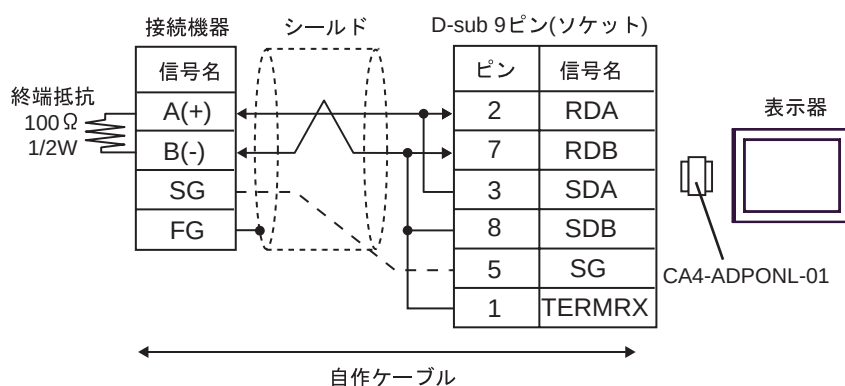


- n:1 接続の場合

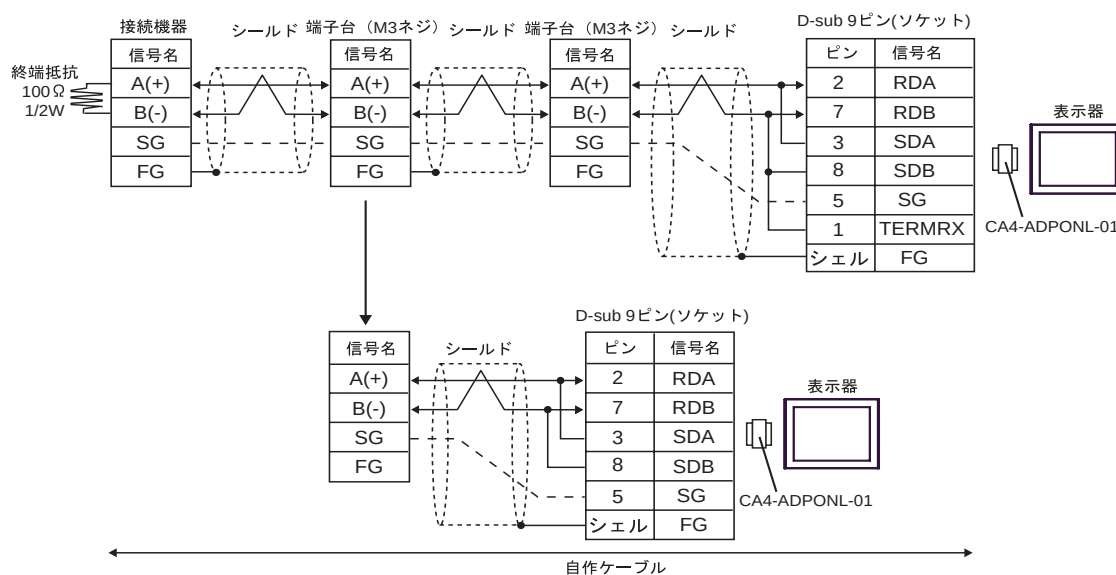


D) (株) デジタル製オンラインアダプタ (CA4-ADPONL-01) および自作ケーブルを使用する場合

- 1:1 接続の場合

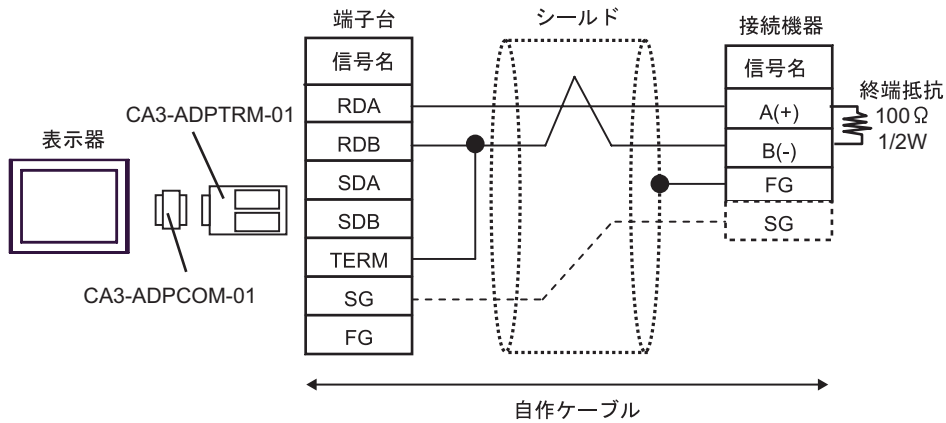


- n:1 接続の場合

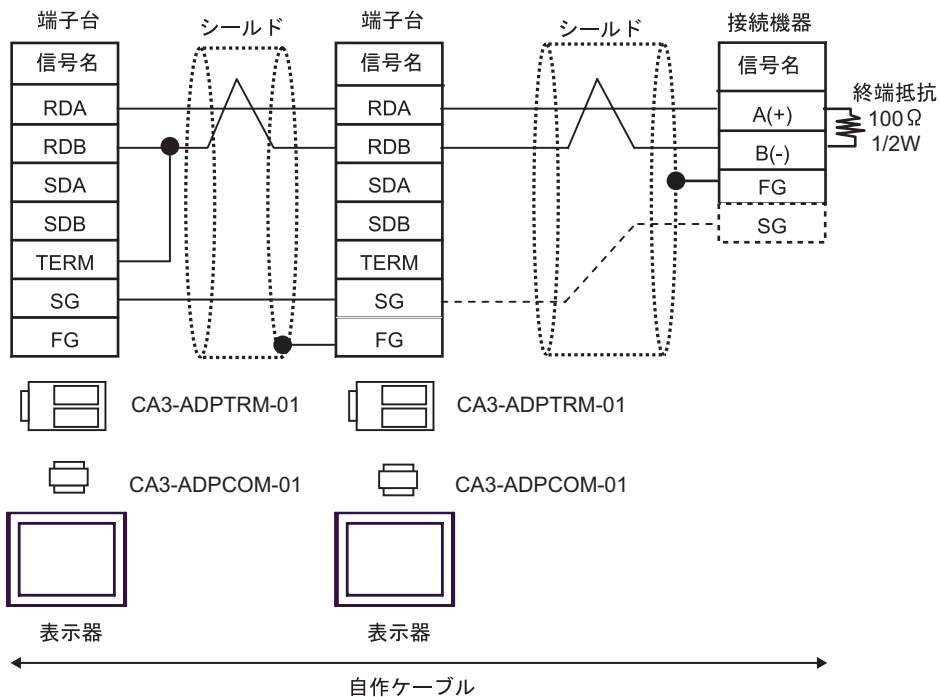


E) (株) デジタル製 COM ポート変換アダプタ (CA3-ADPCOM-01) および (株) デジタル製 コネクタ端子台変換アダプタ (CA3-ADPTRM-01)、自作ケーブルを使用する場合

- 1:1 接続の場合



- n:1 接続の場合

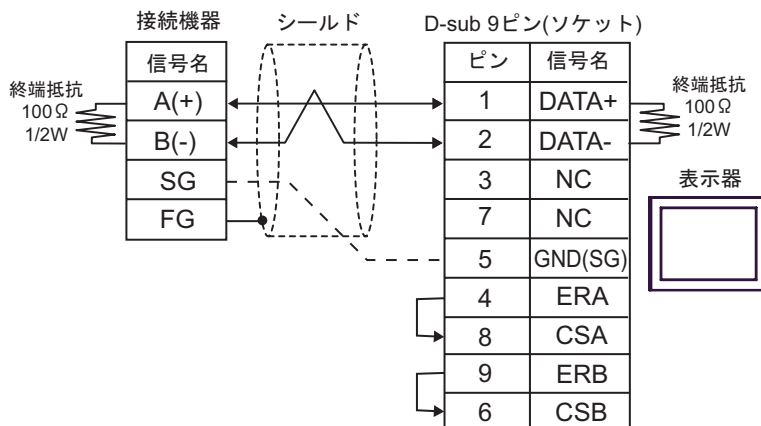


MEMO

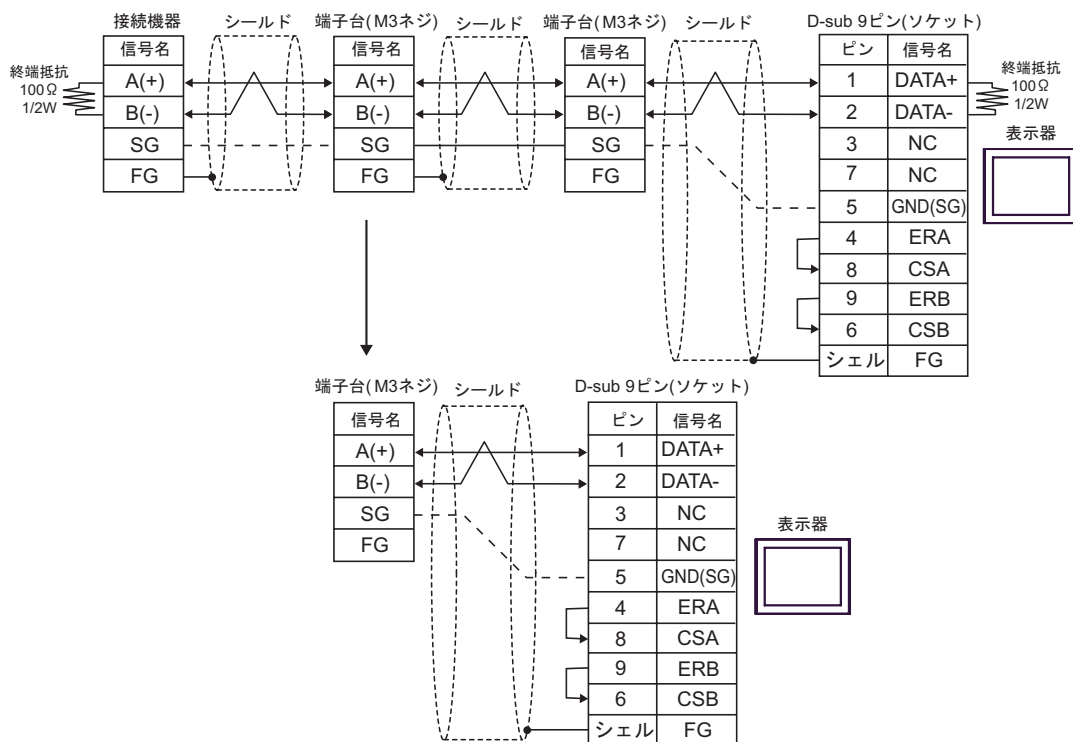
- CA3-ADPTRM-01 の RDB と TERM を接続することにより、表示器側 RDA-RDB 間に 100Ω 1/2W の終端抵抗が挿入されます。

F) 自作ケーブルを使用する場合

- 1:1 接続の場合



- n:1 接続の場合



6 使用可能デバイス

使用可能なデバイスアドレスの範囲を下表に示します。ただし、実際にサポートされるデバイスの範囲は接続機器によって異なりますので、ご使用の接続機器のマニュアルで確認してください。

デバイス	ビットアドレス	ワードアドレス	32 bits	備考
内部デバイス	000000 - 999915	0000 - 9999		

MEMO ・ 表中のアイコンについてはマニュアル表記上の注意を参照してください。

 「表記のルール」

7 デバイスコードとアドレスコード

デバイスコードとアドレスコードは、データ表示器などのアドレスタイプで「デバイスタイプ&アドレス」を設定している場合に使用します。

デバイス	デバイス名	デバイスコード (HEX)	アドレスコード
内部デバイス	-	0000	ワードアドレスの値

8 エラーメッセージ

エラーメッセージは表示器の画面上に「番号：機器名：エラーメッセージ（エラー発生箇所）」のように表示されます。それぞれの内容は以下のとおりです。

項目	内容
番号	エラー番号
機器名	エラーが発生した接続機器の名称。接続機器名は GP-Pro EX で設定する接続機器の名称です。（初期値 [PLC1]）
エラーメッセージ	発生したエラーに関するメッセージを表示します。
エラー発生箇所	エラーが発生した接続機器の IP アドレスやデバイスアドレス、接続機器から受信したエラーコードを表示します。 MEMO ・ IP アドレスは「IP アドレス (10 進数):MAC アドレス (16 進数)」のように表示されます。 ・ デバイスアドレスは「アドレス：デバイスアドレス」のように表示されます。 ・ 受信エラーコードは「10 進数 [16 進数]」のように表示されます。

エラーメッセージの表示例

「RHAA035:PLC1: 書き込み要求でエラー応答を受信しました（受信エラーコード：2[02]）」

- MEMO**
 - ・ 受信したエラーコードの詳細は、接続機器のマニュアルを参照してください。
 - ・ ドライバ共通のエラーメッセージについては「保守/トラブル解決ガイド」の「エラーが表示されたら（エラーコード一覧）」を参照してください。

■ 接続機器特有のエラーコード

エラーコード	説明
06	チェックサムコードが一致しない。
10	未定義コマンドを受信しました。
12	指定データ数と受信データ数が一致しません。
24	指定した属性コードがフォーマット外のデータです。
25	コントラスト調整できない機種にコントラスト調整コマンドを送信しました。
26	指定したコントラスト設定値が範囲外です。
27	輝度調整できない機種に輝度調整コマンドを送信しました。
28	指定した輝度設定値が範囲外です。
FA	指定したシステムエリアのアドレスが範囲外です。
FB	指定したシステムエリアの範囲を超えて書き込み/読み出しを行いました。
FC	受信したデータフォーマットに異常があります。
FF	表示器がデータ送信できない状態が 10 秒以上続いた

■ 接続機器特有のエラーメッセージ

メッセージ ID	エラーメッセージ	原因と対処方法
RHxx128	Memory Link: 受信データがチェックサム異常になりました (エラーコード :06 接続先 :%s)	送信文中のチェックサムが間違っています。送信文を見直してください。
RHxx129	Memory Link: 未定義コマンドを受信しました (エラーコード :10 接続先 :%s)	送信文中のコマンドが間違っています。送信文を見直してください。
RHxx130	Memory Link: 指定データ数と受信データ数が一致しません (エラーコード :12 接続先 :%s)	送信文中のデータ数が間違っています。送信文を見直してください。
RHxx146	Memory Link: 指定したシステムエリアのアドレスが範囲外です (エラーコード :FA 接続先 :%s)	送信文中のシステムエリア指定が間違っています。送信文を見直してください。
RHxx147	Memory Link: 指定したシステムエリアの範囲を超えて書き込み / 読み出しを行いました (エラーコード :FB 接続先 :%s)	システムエリア範囲を超えています。送信文を見直してください。
RHxx148	Memory Link: 受信したデータフォーマットに異常があります (エラーコード :FC 接続先 :%s)	送信文のフォーマットが間違っています。送信文を見直してください。
RHxx149	Memory Link: データ送信できませんでした (エラーコード :FF 接続先 :COM ポート名)	GP からコマンドを送信できない状態です。ケーブルの配線を確認してください。
RHxx150	Memory Link: 指定した属性コードがフォーマット外のデータです (エラーコード :24 接続先 :%s)	送信文中の属性指定が間違っています。送信文を見直してください。
RHxx151	Memory Link: この機種ではコントラスト調整ができません (エラーコード :25 接続先 :%s)	コントラスト調整できない機種にコントラスト調整コマンドを送信しました。GP の機種を確認してください。
RHxx152	Memory Link: 指定したコントラスト設定値が範囲外です (エラーコード :26 接続先 :%s)	送信文中のコントラスト設定値が間違っています。送信文を見直してください。
RHxx153	Memory Link: この機種では輝度調整ができません (エラーコード :27 接続先 :%s)	輝度調整できない機種に輝度調整コマンドを送信しました。GP の機種を確認してください。
RHxx154	Memory Link: 指定した輝度設定値が範囲外です (エラーコード :28 接続先 :%s)	送信文中の輝度設定値が間違っています。送信文を見直してください。

※接続先の後ろの %s には、SIO の場合：COM ポート名、TCP の場合：自局ポート番号、UDP の場合：相手先 IP アドレスが入ります。

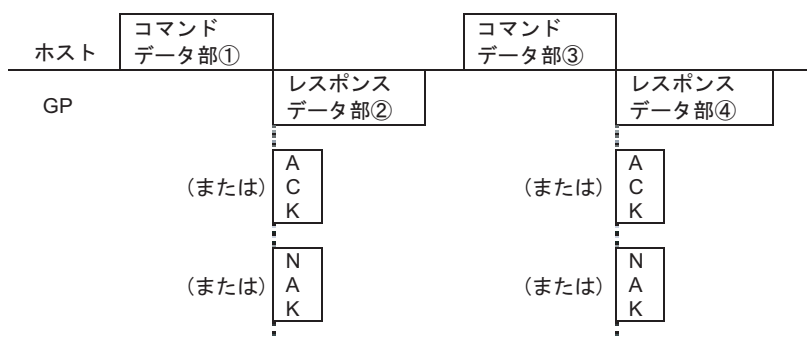
9 メモリリンクコマンド（シリアル通信）

9.1 通信プロトコルの制御の基本手順

通信プロトコルの制御がどのように行われるか基本的な手順を以下に示します。

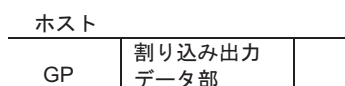
9.1.1 SIO

■ ホストコンピュータから GP ヘータを送る場合



- ①及び③ (コマンドデータ部) はホストより GP への転送データです
- GP はコマンドデータ部を解析した結果、②及び④ (レスポンスデータ部)、ACK、NAK のいずれかを送信するか又は無反応となります。
- ホストからのコマンドデータ③は GP からのレスポンスデータ②を受信した後に送信してください。

■ GP からホストコンピュータヘータを送る場合



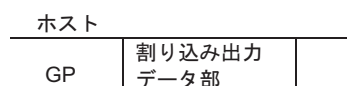
- タッチパネル入力された時に割り込み出力データ部のデータを送信します。(割り込み出力)
- 1:n または RS422/485(2 線式) の場合は割り込み出力されません。割り込み出力問い合わせを参照ください。

9.1.2 SIO 互換モードの通信

■ ホストコンピュータ→ GP



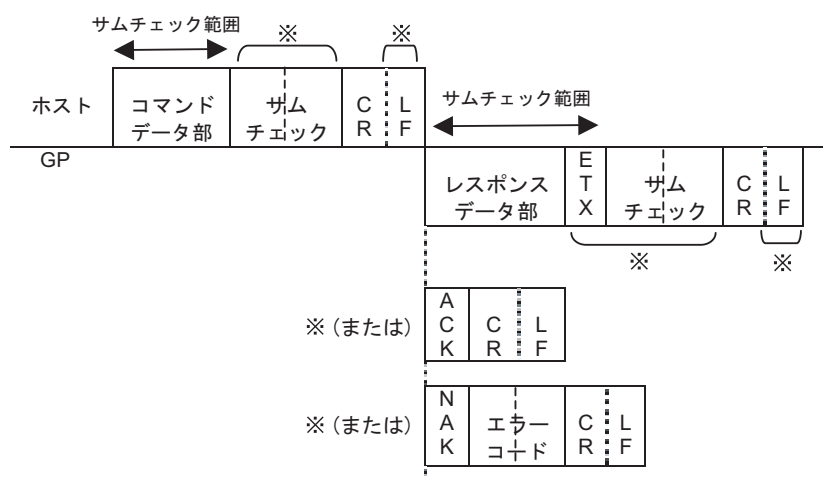
■ GP →ホストコンピュータ（割り込み出力）



- RS244（2線式）の場合は割り込み出力を使用できません。

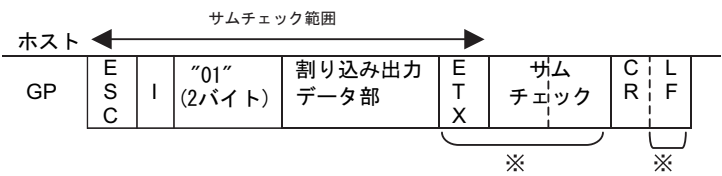
9.1.3 SIO 拡張モード（1：1 アスキー）の通信

■ ホストコンピュータ→ GP



- ※印は設定により付加されないことがあります。

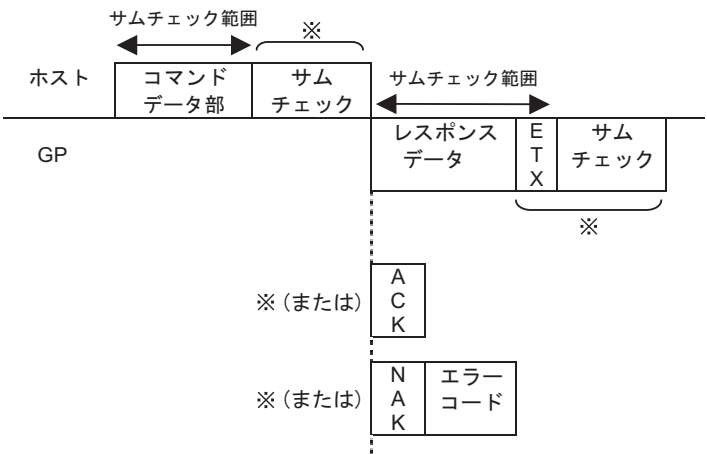
■ GP →ホストコンピュータ（割り込み出力）



- ※印は設定により付加されないことがあります。
- RS244/485（2線式）、UDP 接続の場合、割り込み出力は「割り込み出力問い合わせコマンド」を使用してください。

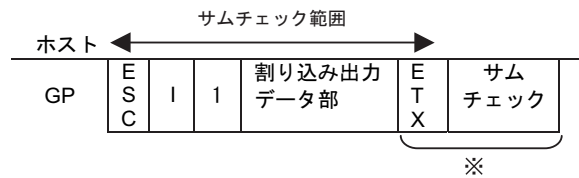
9.1.4 SIO 拡張コマンド（1：1 バイナリ）の通信

■ ホストコンピュータ→ GP



- ※印は設定により付加されないことがあります。

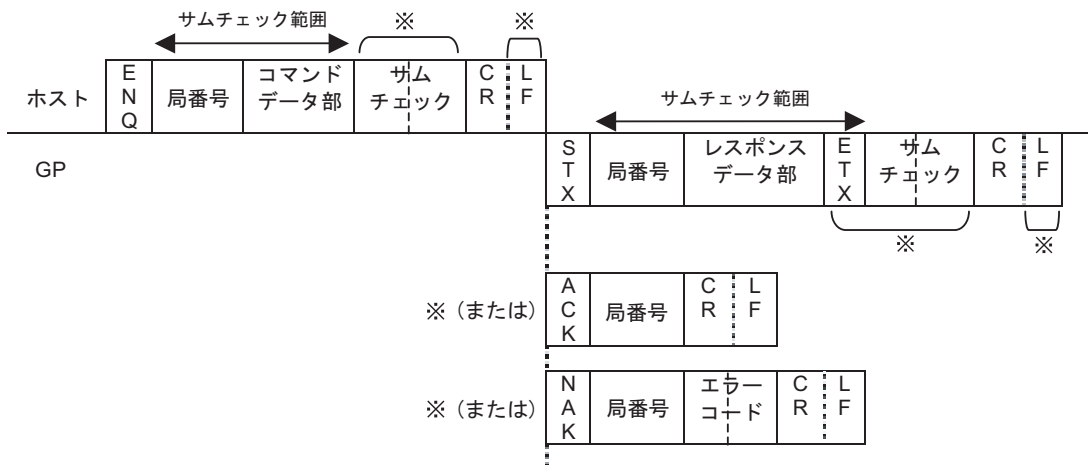
■ GP → ホストコンピュータ（割り込み出力）



- ・ ※印は設定により付加されないことがあります。
- ・ バイナリモードの場合は XON/XOFF 制御できません。ER 制御にて受信応答（ACK/NAK）有で通信してください。
- ・ RS244/485（2 線式）の場合、割り込み出力は「割り込み出力問い合わせコマンド」を使用してください。

9.1.5 SIO 拡張モード（1：n アスキー）の通信

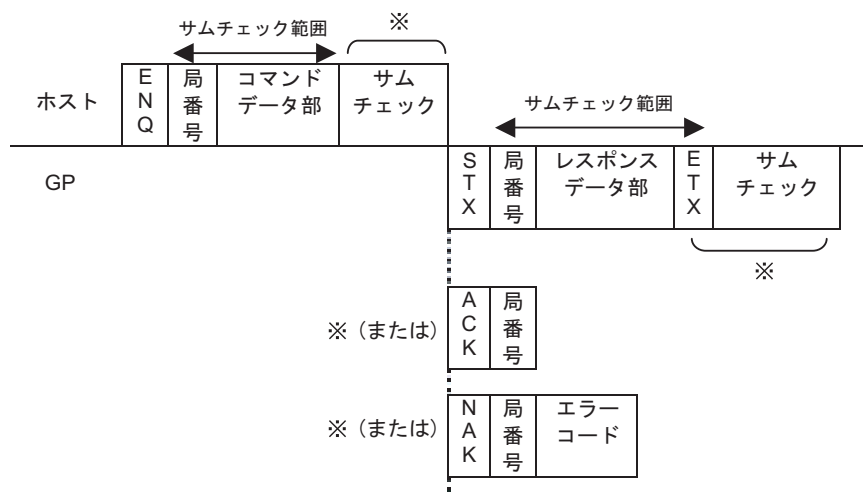
■ ホストコンピュータ → GP



- ・ ※印は設定により付加されないことがあります。
- ・ 局番号を「FF」と設定すると、共通のデータを全局同時に転送することができます。ただし、ACK、NAK 応答は行われません。この場合、一度コマンドを送信してから次のコマンドを送信するまでに 100ms 以上の間隔を空けてください。
また、レスポンスデータを必要とする「システムエリアからの読出し」（ESC R）・「輝度・コントラスト現在値」（ESC \$）コマンドは使用できません。
- ・ 1：n の場合、割り込み出力は「割り込み出力問い合わせ」コマンドを使用します。

9.1.6 SIO 拡張モード (1:n バイナリ) の通信

■ ホストコンピュータ→ GP



- ※印は設定により付加されないことがあります。
- 局番号を「FF」と設定すると、共通のデータを全局同時に転送することができます。ただし、ACK、NAK 応答は行われません。この場合、一度コマンドを送信してから次のコマンドを送信するまでに 100ms 以上の間隔を空けてください。
また、レスポンスデータを必要とする「システムエリアからの読出し」(ESC R)・「輝度・コントラスト現在値」(ESC \$) コマンドは使用できません。
- バイナリモードの場合は XON/XOFF 制御できません。ER 制御にて受信応答 (ACK/NAK) 有で通信してください。
- 1:n の場合、割り込み出力は「割り込み出力問い合わせ」コマンドを使用します。

9.1.7 サムチェックコード

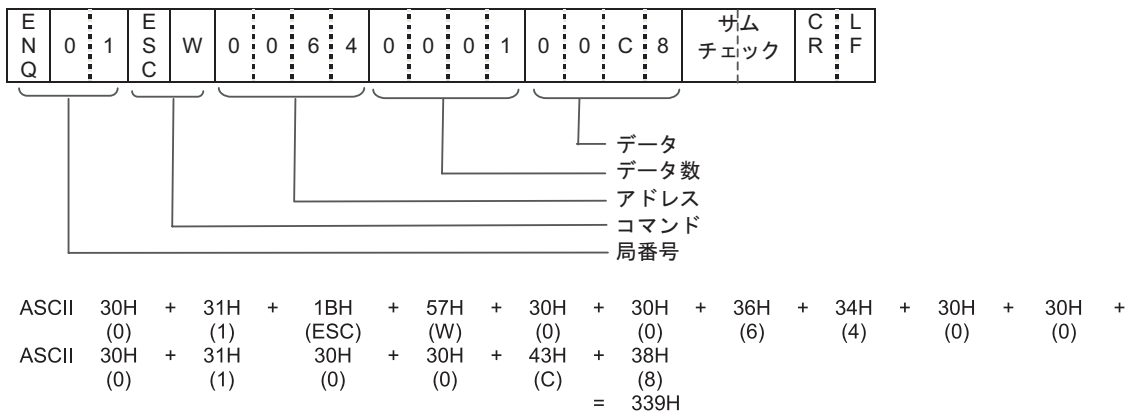
サムチェックコードとは、サムチェックの対象となるデータを加算した結果（サム）の下位1バイト（8ビット）のことをいいます。

アスキーモードの場合は、ASCII 下 2 桁 (16 進) に変換します。

バイナリモードの場合は、そのまま下位 1 バイトを使用します。

＜例＞拡張モード 1:n アスキーの場合

システムエリアの 100 番地に 200 (10 進) を書き込みます。



サムチェックコードは下 2 桁の「39」（33H、39H）となります。

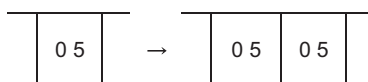
9.2 SIO 1：N バイナリ通信の注意事項

SIO で拡張モード・1：N バイナリ通信時には 2 重化処理が発生します。

9.2.1 ホストコンピュータ→GP

■ ENQ

ホストからの送信で「サムチェック範囲」もしくは「サムチェック」のデータに ENQ コードと同じ「05h」があった場合、直前に「05h」を付加して送信してください。

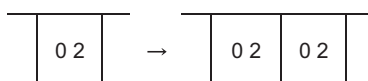


ただし、コマンドデータ部に「データ数」がある場合、付加した「05h」をデータ数に含めません。

9.2.2 GP→ホストコンピュータ

■ STX

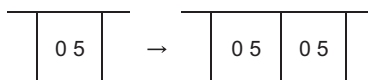
GP シリーズからの返信で「サムチェック範囲」もしくは「サムチェック」のデータに STX コードと同じ「02h」があった場合、直前に「02h」を付加して返信します。



ただし、レスポンスデータ部に「データ数」がある場合、付加した「05h」をデータ数に含めません。

■ ENQ

2 線式 1：N の場合、GP シリーズからの返信で「サムチェック範囲」もしくは「サムチェック」のデータに ENQ コードと同じ「05h」があった場合、直前に「05h」を付加して送信してください。



9.3 コマンドフォーマット

9.3.1 Read フォーマット

■ SIO 互換モード

◆ ホスト コマンドデータ部

E S C	R	アドレス (4バイト)	データ数 (4バイト)	C R
-------------	---	----------------	----------------	--------

< 設定範囲 >

アドレス：0000H ～ 270FH(0 ～ 9999)

データ数：0001H ～ 0100H(1 ～ 256)

全てアスキーコードで入力してください

◆ GP レスポンスデータ部

- ・ 正常時

E S C	A	データ1 (4バイト)	...	データN (4バイト)	C R
-------------	---	----------------	-----	----------------	--------

< 設定範囲 >

データ：0000H ～ FFFFH

- ・ 異常時

NAK で応答

■ SIO 拡張モードのアスキーの場合

◆ ホスト コマンドデータ部

E S C	R	アドレス (4バイト)	データ数 (4バイト)	サ ム チ ェ ッ ク	C R	L F
				※		※

※印は設定により付加されないことがあります。

< 設定範囲 >

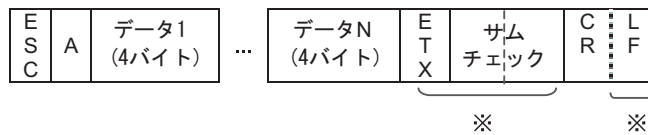
アドレス：0000H ～ 270FH(0 ～ 9999)

データ数：0001H ～ 0100H(1 ～ 256)

全てアスキーコードで入力してください

◆ GP レスponseデータ部

- 正常時



※印は設定により付加されないことがあります。

< 設定範囲 >

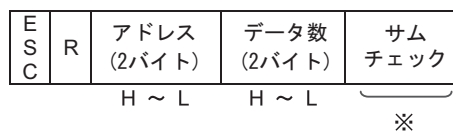
データ：0000H ～ FFFFH

- 異常時

NAK で応答

■ SIO 拡張モードのバイナリの場合

◆ ホスト コマンドデータ部



※印は設定により付加されないことがあります。

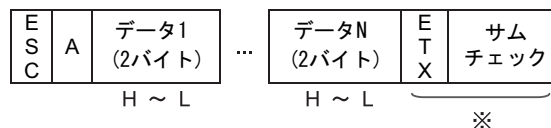
< 設定範囲 >

アドレス：0000H ～ 270FH(0 ～ 9999)

データ数：0001H ～ 0200H(1 ～ 512)

◆ GP レスponseデータ部

- 正常時



※印は設定により付加されないことがあります。

< 設定範囲 >

データ：0000H ～ FFFFH

- 異常時

NAK で応答

9.3.2 Write フォーマット

■ SIO 互換モードの場合

◆ ホスト コマンドデータ部

E S C	W	アドレス (4バイト)	データ1 (4バイト)	...	データN (4バイト)	C R
-------------	---	----------------	----------------	-----	----------------	--------

< 設定範囲 >

アドレス：0000H ～ 270FH(0 ～ 9999)

データ： 0000H ～ FFFFH

全てアスキーコードで入力してください

互換モードの場合、GP からの応答コマンドがありません。

互換モードの場合、書込みデータ数に制限はありません

■ SIO 拡張モードのアスキーの場合

◆ ホスト コマンドデータ部

E S C	W	アドレス (4バイト)	データ数 (4バイト)	データ1 (4バイト)	...	データN (4バイト)	サム チェック	C R	L F
							※		※

※印は設定により付加されないことがあります。

< 設定範囲 >

アドレス：0000H ～ 270FH(0 ～ 9999)

データ数：0001H ～ 0100H(1 ～ 256)

データ： 0000H ～ FFFFH

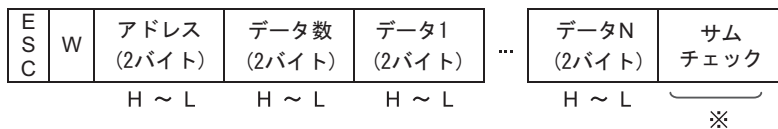
全てアスキーコードで入力してください

◆ GP レスポンスデータ部

ACK または NAK で応答

■ SIO 拡張モードのバイナリの場合

◆ ホスト コマンドデータ部



※印は設定により付加されないことがあります。

< 設定範囲 >

アドレス：0000H ~ 270FH(0 ~ 9999)

データ数：0001H ~ 0200H(1 ~ 512)

データ： 0000H ~ FFFFH

◆ GP レスポンスデータ部

ACK または NAK で応答

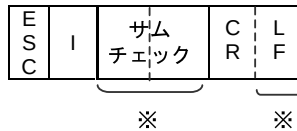
9.3.3 割り込み出力の問い合わせ

拡張モードで「1:n アスキー」「1:n バイナリ」を選択している場合、または 2 線式通信の場合に GP のシステムエリアの絶対値書込みなどで出力された割り込みコードを GP からホストへ出力するためのコマンドです。

2 線式通信の場合は 1:1 接続でも以下の手順で処理してください。

■ SIO のアスキーモードの場合

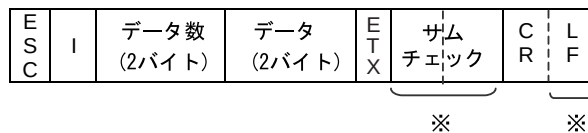
◆ ホスト コマンドデータ部



※は設定により付加されない場合があります。

◆ GP レスポンスデータ部

- 正常時



- 異常時
NAK で応答

< 設定範囲 >

データ数

ホストから問い合わせコマンドが送信された場合は、すでに発生している割り込み出力のデータ数のことをいいます。

すでに発生している割り込み出力データをすべて取得する場合は、今回送信したコマンドを含めたデータ回数のコマンドを送信する必要があります。

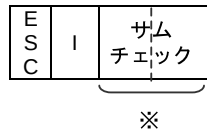
データ

00H ~ FEH を ASCII2 桁 (16 進) に変換して出力されます。

割り込み出力のデータがないときには「00」となります。

■ SIO のバイナリモードの場合

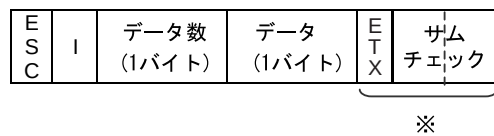
◆ ホスト コマンドデータ部



※は設定により付加されない場合があります。

◆ GP レスポンスデータ部

- ・ 正常時



※は設定により付加されない場合があります。

- ・ 異常時
NAK で応答

< 設定範囲 >

データ数

ホストから問い合わせコマンドが送信された場合は、すでに発生している割り込み出力のデータ数のことをいいます。

すでに発生している割り込み出力データをすべて取得する場合は、今回送信したコマンドを含めたデータ回数のコマンドを送信する必要があります。

データ

00H ～ FEH が出力されます。

割り込み出力のデータがないばあい「00」となります。

9.3.4 輝度・コントラスト調整

輝度・コントラストをコマンドを用いて調整するためのコマンドデータ部の内容は以下のとおりです。GP の機種によって輝度・コントラスト調整ができないものもあります。

■ SIO 互換モードの場合

◆ ホスト コマンドデータ部

E S C	#	属性 (4バイト)	設定値 (4バイト)	C R
-------------	---	--------------	---------------	--------

< 設定範囲 >

属性：0000H ～ 0001H（0：コントラスト調整、1：輝度調整）

設定値：「■ 輝度・コントラスト一覧」（59 ページ）を参照してください。

全てアスキーコードで入力してください。

◆ GP レスポンス部

レスポンスデータはありません。

■ SIO 拡張モードのアスキーの場合

◆ ホスト コマンドデータ部

E S C	#	属性 (4バイト)	設定値 (4バイト)	サム チェック	C R	L F
				※	※	

※は設定により付加されないことがあります。

< 設定範囲 >

属性：0000H ～ 0001H（0：コントラスト調整、1：輝度調整）

設定値：「■ 輝度・コントラスト一覧」（59 ページ）を参照してください。

全てアスキーコードで入力してください。

◆ GP レスポンスデータ部

ACK または NAK で応答

■ 拡張モードのバイナリモードの場合

◆ ホスト コマンドデータ部

E S C	#	属性 (2バイト)	設定値 (2バイト)	サム チェック
		H ~ L	H ~ L	※

< 設定範囲 >

属性：0000H ~ 0001H（0：コントラスト調整、1：輝度調整）

設定値：「■ 輝度・コントラスト一覧」（59 ページ）を参照してください。

全てアスキーコードで入力してください。

◆ GP レスポンスデータ部

ACK または NAK で応答

9.3.5 輝度・コントラスト現在値

輝度・コントラストの現在値をコマンドを用いて取得するためのコマンドデータ部の内容は以下のとおりです。GP の機種によって輝度・コントラストがないものもあります。

■ SIO 互換モードの場合

◆ ホスト コマンドデータ部

E		C
S	\$	R
C		

◆ GP レスポンスデータ部

E	D	コントラスト値 (4バイト)	輝度値 (4バイト)	C
S				R
C				

■ SIO 拡張モードのアスキーモードの場合

◆ ホスト コマンドデータ部

E	\$	サム	C	L
S		チェック	R	F
C				

※ ※

※は設定により付加されない場合があります。

◆ GP レスポンスデータ部

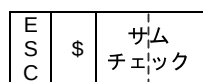
E	D	コントラスト値 (4バイト)	輝度値 (4バイト)	サム	C	L
S				チェック	R	F
C						

※ ※

※は設定により付加されない場合があります。

■ SIO 拡張モードのバイナリモードの場合

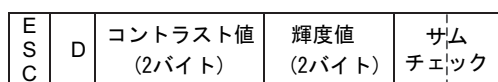
◆ ホスト コマンドデータ部



※

※は設定により付加されない場合があります。

◆ GP レスポンスデータ部



H ~ L

H ~ L

※

※は設定により付加されない場合があります。

■ 輝度・コントラスト一覧

GP	輝度設定 設定範囲	コントラスト設定 設定範囲
AGP-3302B	0 (明) ~ 7 (暗)	0 (明) ~ 7 (暗)
AGP-3301L	0 (明) ~ 7 (暗)	0 (明) ~ 7 (暗)
AGP-3301S	0 (明) ~ 7 (暗)	0 (明) ~ 7 (暗)
AGP-3300L	0 (明) ~ 7 (暗)	0 (明) ~ 7 (暗)
AGP-3300S	0 (明) ~ 7 (暗)	0 (明) ~ 7 (暗)
AGP-3300T	0 (明) ~ 7 (暗)	-
AGP-3400S	0 (明) ~ 7 (暗)	0 (明) ~ 7 (暗)
AGP-3400T	0 (明) ~ 7 (暗)	-
AGP-3500L	0 (明) ~ 7 (暗)	0 (明) ~ 7 (暗)
AGP-3500S	0 (明) ~ 7 (暗)	0 (明) ~ 7 (暗)
AGP-3500T	0 (明) ~ 7 (暗)	-
AGP-3600T	0 (明) ~ 7 (暗)	-
AGP-3450T	0 (明) ~ 7 (暗)	-
AGP-3550T	0 (明) ~ 7 (暗)	-
AGP-3650T	0 (明) ~ 7 (暗)	-
AGP-3750T	0 (明) ~ 7 (暗)	-

10 サンプルプログラム（シリアル通信）

10.1 サンプルシステム

GP とホストがデータをやり取りするために必要なホスト側のプログラム例と、GP 側で行う部品設定例を示します。また、以下の部品設定でサンプルプログラムを実行した場合の、GP の画面の変化を示します。

以下の画面を例に作成の手順を示します。

「モータ ON」「モータ OFF」「表示」「異常」スイッチを押すと、それぞれの割り込みコードがホストに出力され、以下の動作を行います。

各スイッチの動作

「モータ ON」... モータを稼動し、沈殿槽に沈殿物が 50% 入ります。

「モータ OFF」... モータを停止します。

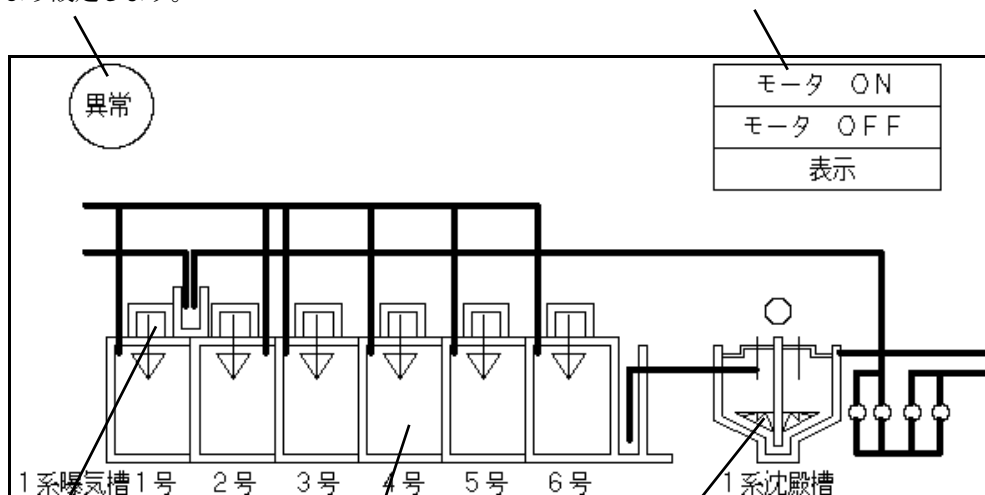
「表示」..... 沈殿槽に沈殿物が 50% 入ります。

「異常」..... 沈殿槽に沈殿物が 20% 入ります。

■ 完成イメージ

異常スイッチです。押すとランプが点灯するように設定します。

モータの稼動スイッチ、モータの停止スイッチ、沈殿槽の量表示のスイッチです。押すと、ランプが点灯するように設定します。



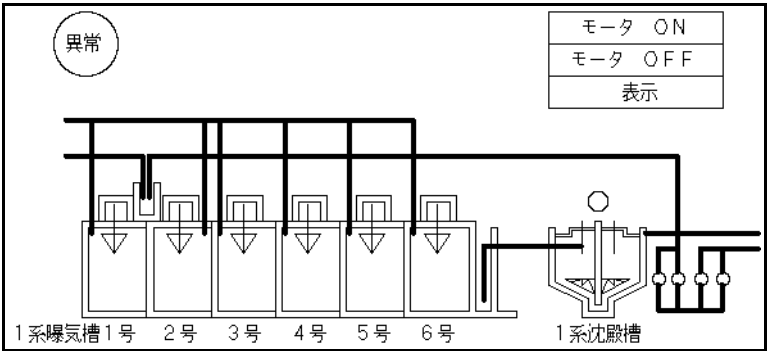
モータ ON スwitchが押されると、マークを表示するよう設定します。

プログラムを実行すると、物質が入ったイメージにします。

「モータ ON」「表示」「異常」スイッチが押されると、その時の沈殿物の量を表示するよう設定します。

■ 作成手順

(1) GP-Pro EX で画面を作成します。



GP 運転時に表示される画面です。

(2) GP-Pro EX で部品を設定します。

◆ 部品設定例

スイッチ

画面番号	部品	スイッチ	ワードアドレス	ワード動作	定数
ベース画面 1	モータ ON	ワードスイッチ	#MEMLINK	ワード書込み 16 ビット Dec	0031
	モータ OFF				0032
	表示				0033
	異常				0034

◆ アドレス使用マップ

部品設定例で示した部品は、それぞれのアドレスに次のように割り付ます。

スイッチ

アドレス 13 (割り込み出力) にデータが書き込まれると、RS232C ポートから下位 1 バイトのコードが出力されます。このため、スイッチはワード書込みを使用しています。

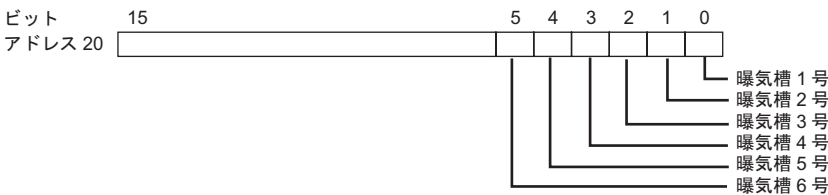
モータ ON..... アドレス 13 に 0031 をワード書込み

モータ OFF..... アドレス 13 に 0032 をワード書込み

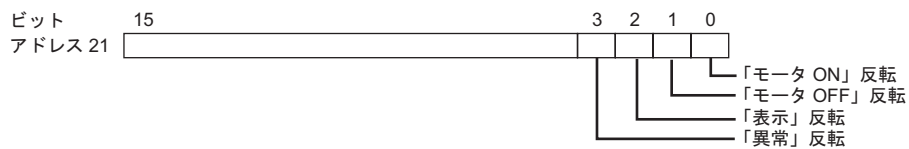
表示 アドレス 13 に 0033 をワード書込み

異常 アドレス 13 に 0034 をワード書込み

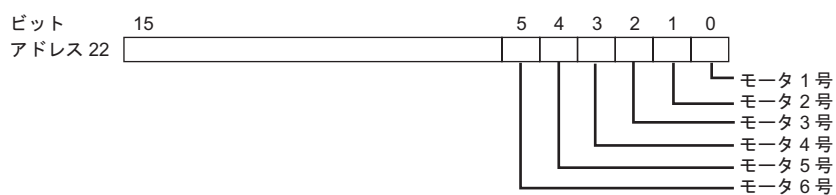
曝気槽



モータのスイッチ



モータ



沈殿槽



(3) GP とホストがデータをやり取りするためのプログラムを、ホスト側で作成します。

<例> DOS / V 機、C 言語を使用した場合

```

/*****/
/*
*/
/*GP シリーズ メモリリンク 通信サンプルプログラム */
/*
*/
/*****/

#include<stdio.h>
#include<dos.h>
#include<string.h>
#include<stdlib.h>
#include<conio.h>

#define data_size_str2 20 /*str2 のデータサイズ 20bite*/
#define data_size_wr_data 24 /*wr_data のデータサイズ 24bite*/

#define serial_port_BIOS 0x14 /*DOS/V シリアルポート BIOS*/
#define serial_port_number 0x00 /* 使用シリアルポート番号 */
#define serial_port_INT 0x00 /* シリアルポートの初期化 */
#define serial_port_parameter 0xE7 /* 9600bps,8bit,stopbit;1,parity:none*/

#define get_status 0x03 /* シリアルポート状態の取得 */
#define serial_port_write 0x01 /* シリアルポートの書き込み */
#define serial_port_read 0x02 /* シリアルポートの読み出し */

#define status_bit_6000 0x6000 /* ポートステータス bit13,14*/
#define status_bit_0020 0x0020 /* ポートステータス bit5*/

/*****/
/*
SIO の通信設定
*/
/*****/
void open_SIO(void); /*RS232C の通信設定 */

/*****/
/*
ポートステータスの取得、判別
*/
/*****/

int err_status(void); /* ポートステータスの取得 */
void write_ready(void); /* 送信バッファレジスタ、送信レジスタの状態の取得 */
int read_ready(void); /* データセット状態の確認 */
```

```

/*****/

/*          データ書き込み          */
/*****/

void write_data(char wr_data); /* レジスタへの書き込み */
void write(char *wr_data); /* GP へのデータの書き込み */

/*****/

/*          データ読み出し          */
/*****/

int read_data(void); /* GP からのデータ読み出し */
void change_screen(int interrupt_data); /* GP からの割り込み受信データの判別 */
int read(void); /* GP からの割り込み受信データの読み込み */

/*****/

/*          キー入力の確認          */
/*****/

int kbhit(void);

/*****/

/*          グローバル変数          */
/*****/

int interrupt_data, port_status;
char *str2;

void main(void)
{
int no_data;

    char *wr_data = (char*)malloc(sizeof(char)*data_size_wr_data);
/*wr_data のメモリを確保 */
    str2 = (char*)malloc(sizeof(char)*data_size_str2); /*str2 のメモリを確保 */
    open_SIO(); /*RS232C の通信設定 */
    wr_data = "\x1bW000F0001\x0d\0"; /* アドレス 15 に 0x1 を書き込み : 画面番号 1 設定 */
    write(wr_data);
    wr_data = "\x1bW0014003F\x0d\0";
/* アドレス 20 に 0x3F を書き込み : 曝気槽 1 ~ 6 号物質投入 */
    write(wr_data);

/*****/

/*          GP からのデータ受信と判別、書き込み          */
/*          キーを押すと実行終了          */
/*****/

    while(1)
    {
        no_data = read();

```

```

        if(no_data == 1) /* キー入力があれば no_data=1*/
        {
            break;
        }
        else
        {
            wr_data = str2;
            write(wr_data);
        }
    }
    getch(); /* キーのコードをキーバッファから取り除く */
    free(wr_data); /*wr_data のメモリ領域開放 */
    free(str2); /*str2 のメモリ領域開放 */
}

```

/* 送信バッファレジスタ、送信レジスタの状態の取得 */

```

void write_ready(void)
{
    int err6000;

    err6000 = 0;
    while(status_bit_6000 != err6000)
    {
        err6000 = err_status() & status_bit_6000;
    }
    return;
}

```

/* データセットの状態の確認 */

```

int read_ready(void)
{
    int no_data,err0020;

    err0020 = 0;
    while(status_bit_0020 != err0020)
    {
        err0020 = err_status() & status_bit_0020;
        if(kbhit()) /* キー入力の有無の判定 */
        {
            no_data = 1; /* キー入力があれば no_data= 1 */
            break; /* プログラム終了 */
        }
    }
}

```

```

        return(no_data);
    }
    /*GP へのデータ書き込み */
    void write(char *wr_data)
    {
        while(*wr_data != '\0') /* データが NULL になるまで書き込み */
        {
            write_ready();
            write_data(*wr_data);
            wr_data++; /* ポインタが示すアドレスをインクリメント */
        }
        return;
    }
    /******
    /*      GP からの割り込み受信データの判別      */
    /*      アドレス 20,21,22,23 にデータ書き込み      */
    /******
    void change_screen(int interrupt_data)
    {
        switch(interrupt_data)
        {
            /*interrupt_data が 1 なら、アドレス 21 に 0x1, 22 に 0x3F, 23 に 0x50 を書き込み */
            case 1: str2 = "\x1bW00150001003F0050\x0d\0";
                    break;

            /*interrupt_data が 2 なら、アドレス 21 に 0x2, 22 に 0x0, 23 に 0x0 を書き込み */
            case 2: str2 = "\x1bW0015000200000000\x0d\0";
                    break;

            /*interrupt_data が 3 なら、アドレス 21 に 0x4, 22 に 0x0, 23 に 0x50 を書き込み */
            case 3: str2 = "\x1bW0015000400000050\x0d\0";
                    break;

            /*interrupt_data が 4 なら、アドレス 21 に 0x8, 22 に 0x0, 23 に 0x20 を書き込み */
            case 4: str2 = "\x1bW0015000800000020\x0d\0";
                    break;

            /*interrupt_data が 1 ～ 4 以外なら NULL を書き込み */
            default : str2 = "\0";
                    break;
        }
        return;
    }

```

```

/*****
/*      GP からの割り込み受信データの読み込み      */
/*      interrupt_data が NULL 以外になるまで、実行      */
*****/

int read(void)
{
    int no_data;
    do
    {
        no_data = read_ready();/* データセットの状態の確認 */
        if(no_data == 1)/* キー入力があれば no_data=1*/
        {
            break;
        }
        else
        {
            read_data();/*GP からの受信データ読み出し */
            change_screen(interrupt_data); /*GP からの受信データの判別 */
        }
    }while(*str2 == '\0');
    return(no_data);
}

/*RS232C の通信設定 */
void open_SIO(void)
{
    union REGS regs;
    regs.x.dx = serial_port_number;
    regs.h.ah = serial_port_INT;
    regs.h.al = serial_port_parameter;
    int86(serial_port_BIOS,&regs,&regs);
    return;
}

/* ポートステータスの取得 */
int err_status (void)
{
    union REGS regs;
    regs.x.dx = serial_port_number;
    regs.h.ah = get_status;
    int86(serial_port_BIOS,&regs,&regs);
    port_status = regs.x.ax;
    return(port_status);
}

```

```
}

/* レジスタへの書き込み */
void write_data(char wr_data)
{
    union REGS regs;
        regs.x.dx = serial_port_number;
        regs.h.ah = serial_port_write;
        regs.h.al = wr_data;
        int86(serial_port_BIOS,&regs,&regs);
    return;
}

/* GP からのデータ読み出し */
int read_data(void)
{
    union REGS regs;
        regs.x.dx = serial_port_number;
        regs.h.ah = serial_port_read;
        int86(serial_port_BIOS,&regs,&regs);
        interrupt_data = regs.h.al;
    return(interrupt_data);
}
```

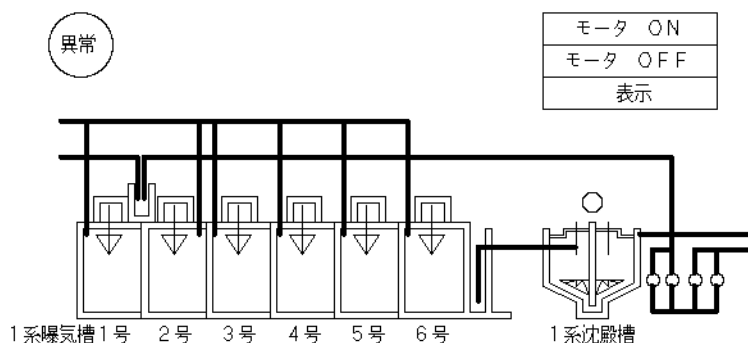
MEMO

- open_SIO(void)、err_status(void)、write_data(char wr_data)、read_data(void) は、機種依存する関数です。DOS/V 機以外でプログラムする場合は作りかえが必要です。
-

(4) GP に画面データを転送し、運転します。

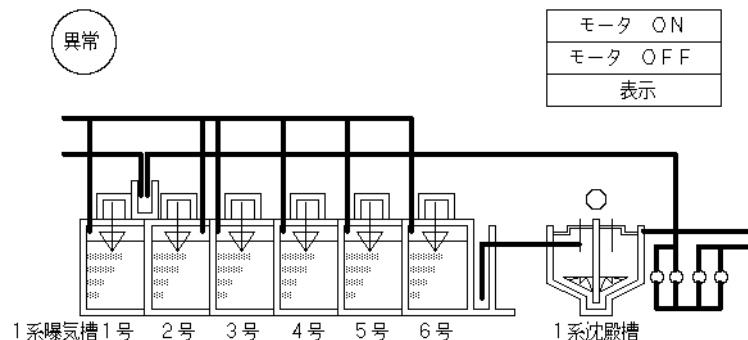
◆ GP 運転画面

プログラム実行前



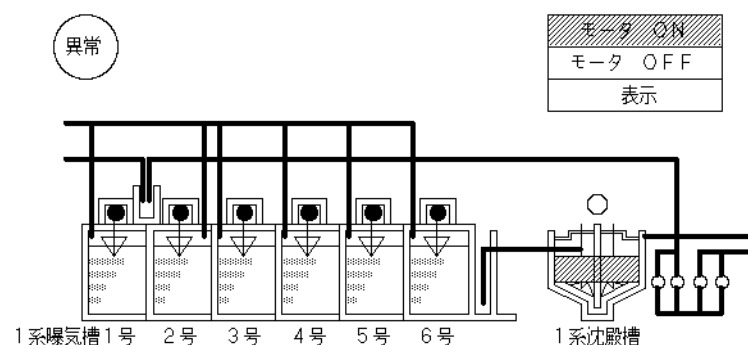
プログラム実行後

 が 6 つ入った表示になります。



モータ ON のスイッチを押します。

ホストにアスキーコード “0031(HEX)” (=データ “1”) が GP から出力され、ホストのプログラムによって画面が変わります。



10.2 複数の GP との通信時の問題点と解決策

1 台のホストで複数の GP をコントロールする場合、通信におけるホストの役割は

1. 表示用のデータ転送
2. 各号機のタッチパネル入力情報の読出し（ポーリング）

の 2 つがあります。

ただし、接続する GP の台数や表示用データの量によりホストの負担は大きくなります。

また、通信に費やす時間が大きく、GP の反応速度（表示の更新やタッチ入力の反応）も低下し、大きな影響を与えます。

したがって、マルチドロップシステムを設計するにあたっては、接続する GP の台数と表示用データの量については十分検討して設定してください。

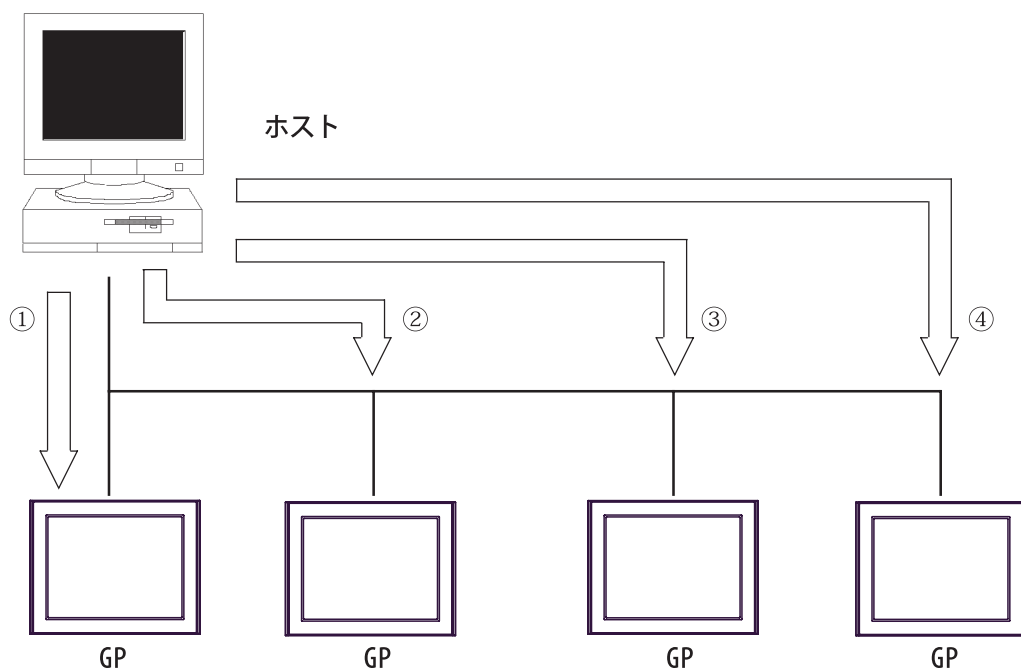
■ 表示用データの共通化（一斉同報形式）

ホストから各号機にデータを転送したのでは時間がかかります。

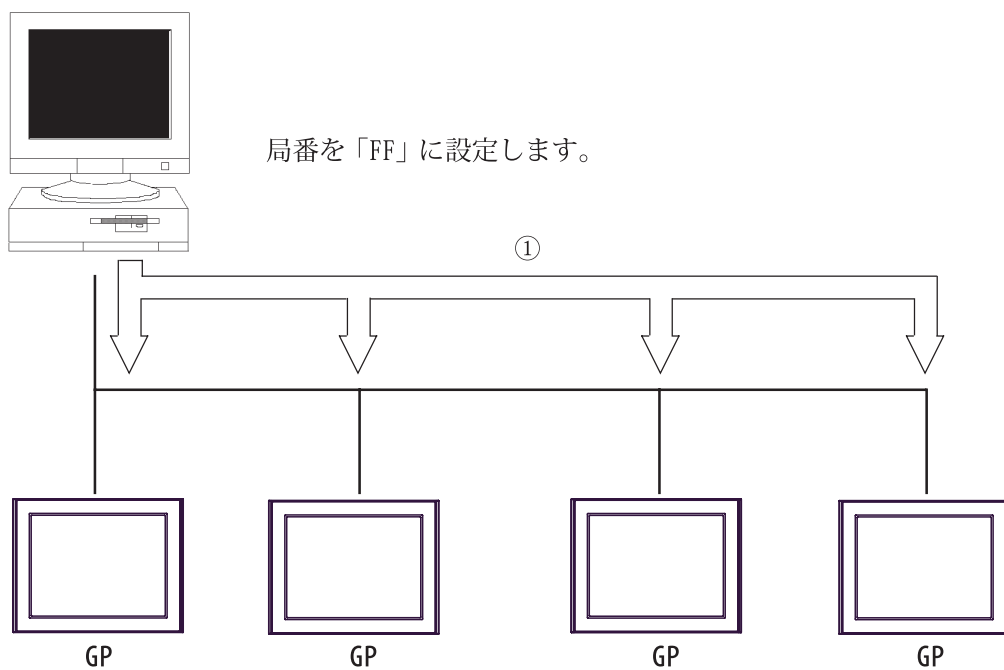
そこで全機に共通するデータを同時に転送することで通信を効率よく行うことができます。

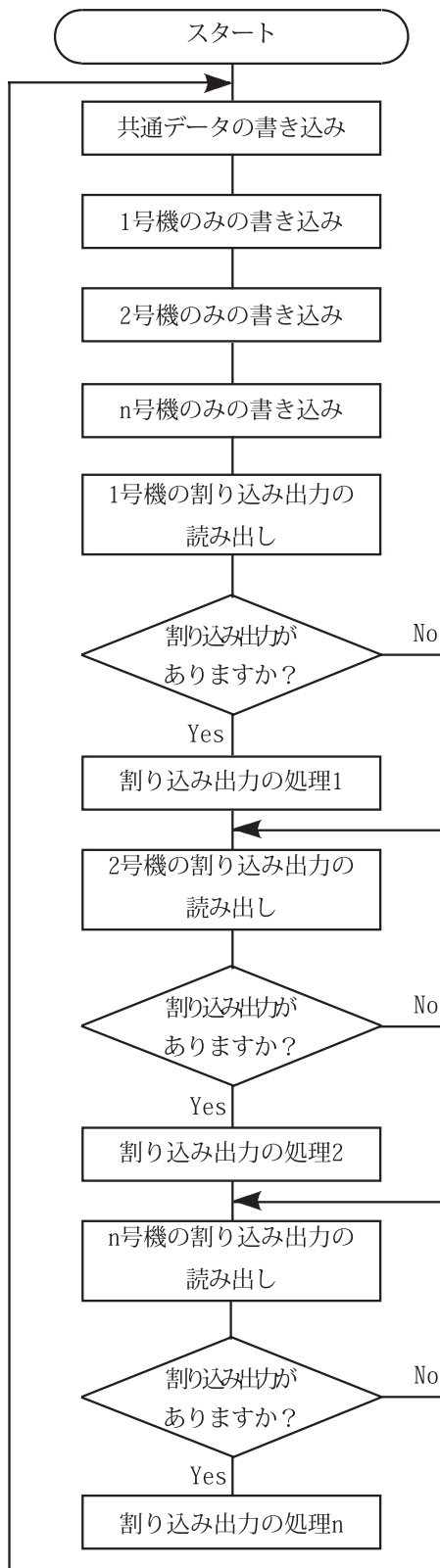
（コマンドの局番号を "FF" にすることで共通データを同時に転送できます。）

- ◆ 1つつデータを送る場合
4回分の通信時間がかかります。



- ◆ 共通データを同時に送る場合（一斉同報）
1回分の通信時間で転送できます。





(1)共通データの同時書き込み

全機に共通する表示用データをシステムエリアに書き込む

コマンド：ESC W

(号機 No.=FF)

(2)個別の書き込み

各号機に特有な表示用データをシステムエリアに書き込む

コマンド：ESC W

(3)ポーリング処理

各号機にタッチ入力の有無を確認し、必要に応じて処理を行う。

コマンド：ESC I

MEMO

- タッチパネル入力の反応を良くする場合は、「(2) 個別書き込み」を行う際、各書き込み処理の間にポーリング処理を行ってください。
- 書込むデータ量は最小限にしてください
例：変化したデータのみを書込むなど

Memo

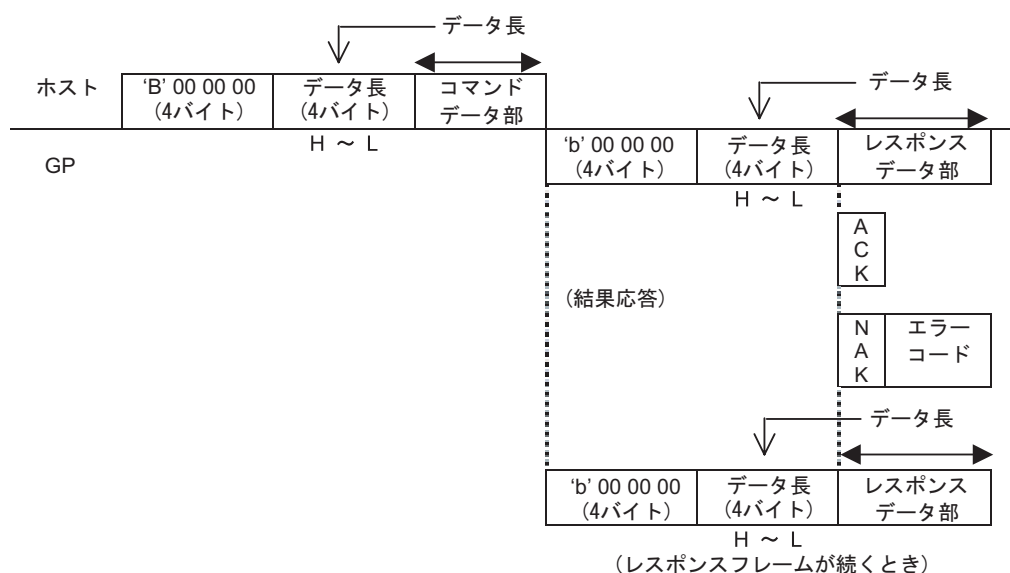
11 メモリリンクコマンド（イーサネット通信）

11.1 通信プロトコルの制御の基本手順

通信プロトコルの制御がどのように行われるか基本的手順を以下に示します。

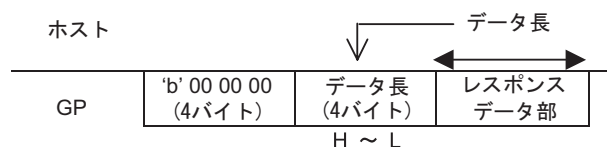
11.1.1 LAN

■ ホストコンピュータから GP ヘータを送る場合



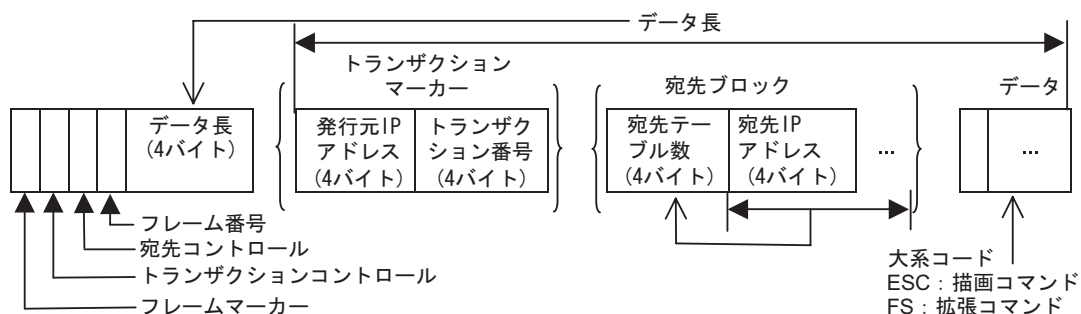
- コマンドデータ部はホストより GP への転送データです。
- GP はコマンドデータ部を解析した結果、レスポンスデータ部、ACK、NAK または無反応となります。

■ GP からホストコンピュータヘータを送る場合



■ フレームフォーマットの詳細

メモリリンク LAN の回線フレームは以下のような構造をしています。



メモリリンク LAN のフレームはフレームマーカー～データ長までの 8 バイトはすべてフレームで存在します。

よって、フレームチェックをする際は先頭の 8 バイトをチェックし、その 8 バイトの中のデータ長で以降のデータをチェックするようにしています。

◆ フレームマーカー (1 バイト)

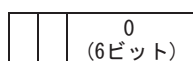
フレームの種類を識別するのに使用します。

‘ B ’ バイナリコマンドフレーム

‘ b ’ バイナリレスポンスフレーム

・バイナリフレームしかサポートしていません。

◆ トランザクションコントロール (1 バイト)



トランザクションマーカーの有無

フレームにトランザクションマーカーが付加されるかどうか

0: トランザクションマーカーなし

1: トランザクションマーカーあり

フレーム継続フラグ

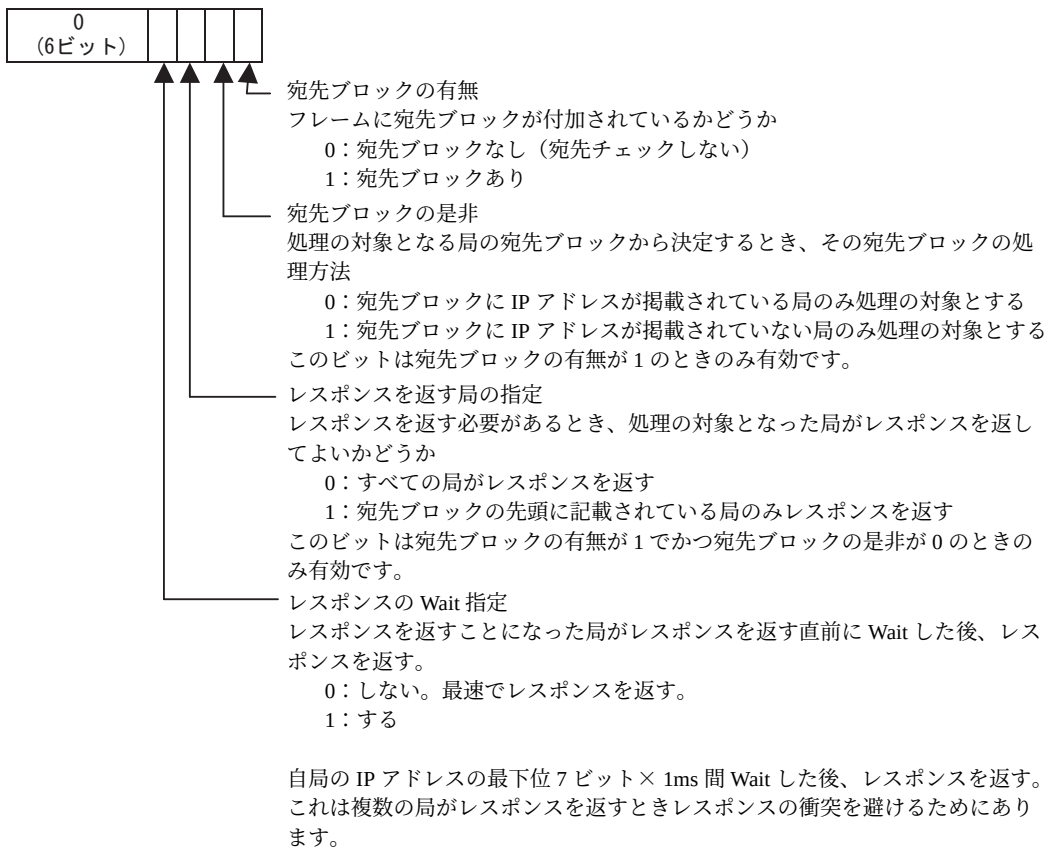
0: 最終フレーム

1: 継続フレームあり

(このフレームの後にも、フレームが続くことを意味します)

巨大なデータを交信する場合、データを複数のフレームに分割して送受信します。そのとき、分割フレームであることを示すビットで、先頭と途中のフレームには 1 をセットし、最終フレームは 0 をセットします。

◆宛先コントロール (1 バイト)

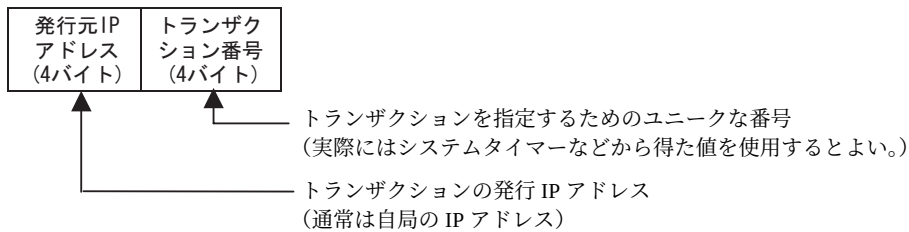


◆宛先コントロールの使用例

通常 1:1 の相手局に対して交信する場合は、このコントロールは 00h を指定します。

1:n の場合は n の内 1 局のみレスポンスを指定するのであれば 05h を指定し、すべての局にレスポンスを期待するときは 09h を指定します。

◆ トランザクションマーカー



トランザクションマーカーの使用用途

GP はトランザクションマーカーありのコマンドフレームを受信するとそのコマンドの処理を実行します。(必要があればレスポンスも送信します。) ここまではマーカー無しの処理と同じです。その後、今行った処理の結果を GP の内部にプールします。

そして次にトランザクション結果問い合わせコマンドを受けるとそのプールした内容をレスポンスとして返します。

GP は最大 10 件のトランザクションをプールします。10 件以上の登録は、古いトランザクションから順に消されその上に登録されます。

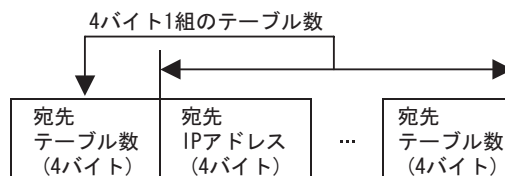
◆ フレームマーカー

1 つの意味をコマンドまたはレスポンスで複数のフレームに分かれる場合、そのフレームの順序を示す番号として 0 からの連番で指定します。最大は 255 です。

フレームは 1K バイト以下で分割してください。

◆ 宛先ブロック

宛先ブロックは、宛先ブロックの有無が 1 のとき付加されます。0 の時には付加されません。



11.2 デマンドポーリング

TCP 接続の時、通信設定で設定した時間ホストからの要求がない場合、GP はホストの生存を確認するために、ポーリングの催促コマンドを送信します。

ホストはポーリングの催促コマンドを受信すると、ポーリングコマンドを GP へ送信するように処理してください。ホストが送信したポーリングコマンドを受信することで、GP はホストの生存を確認します。

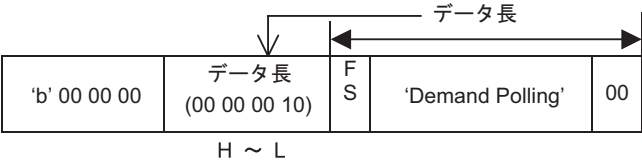
ホストからの応答がない場合、GP は接続を切断します。

ホスト側でデジタル製の API を使用している場合、ポーリングの最速コマンドの応答処理は API で自動的に行われます。

11.2.1 ポーリングの催促（FS Demand）

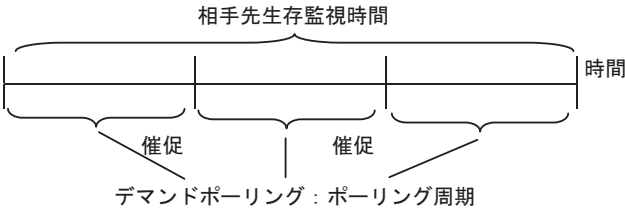
GP からホストへ送信するポーリングの催促コマンドのデータ部の内容は次のとおりです。

ホスト：なし
GP：レスポンスデータ部



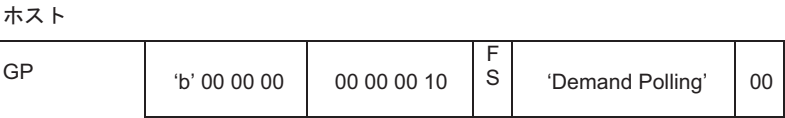
<設定内容>

- データ：' Demand Polling '



デマンドポーリング：ポーリング周期で設定した時間が経過したタイミングでポーリングの催促コマンドを送信します。

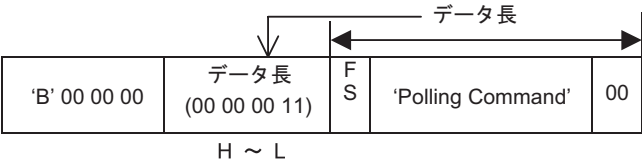
ポーリングの催促コマンドはホストのポーリングコマンド発行を促すものです。ポーリングの催促コマンドを送信することで GP がポーリング待ち状態になることはありません。



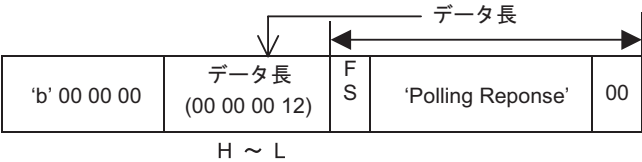
11.2.2 ポーリングコマンド（FS Polling）

ホストから GP へ送信されるポーリングコマンドの内容は次のとおりです。

ホスト：コマンドデータ



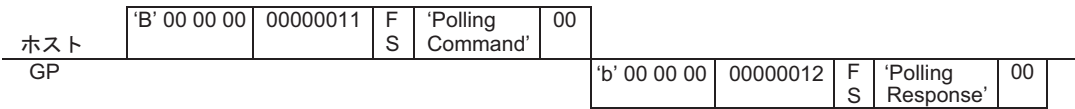
GP：レスポンスデータ部



<設定内容>

- データ：' Polling Command '

<例>



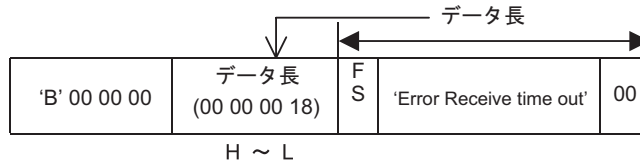
11.2.3 エラー通知 (FS Error)

プロトコル上にエラーが発生した場合、GP またはホストから相手局に対してエラー通知を行うためのコマンドの内容は次のとおりです。

このフレームに対するレスポンスは必要ありません。

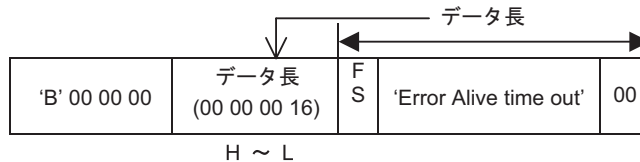
ホスト：コマンドデータ

- キャラクター間タイムアウトエラーフレーム



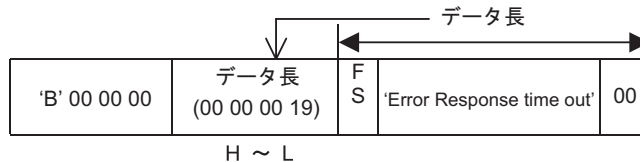
<設定内容>

- データ：' Error Receive time out '
- 相手先生存監視タイムアウトエラーフレーム



<設定内容>

- データ：' Error Alive time out '
- プロトコル間タイムアウトエラーフレーム

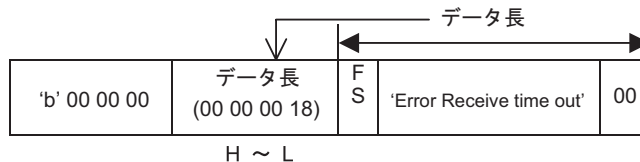


<設定内容>

- データ：' Error Response time out '

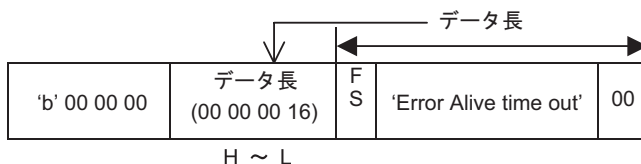
GP：レスポンスデータ部

- キャラクター間タイムアウトエラーフレーム



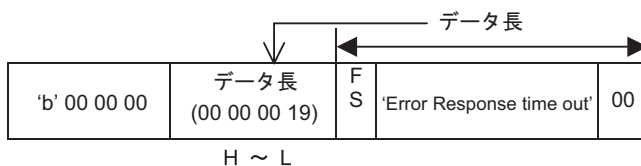
＜設定内容＞

- データ：‘ Error Receive time out ’
- 相手先生存監視タイムアウトエラーフレーム



＜設定内容＞

- データ：‘ Error Alive time out ’
- プロトコル間タイムアウトエラーフレーム



＜設定内容＞

- データ：‘ Error Receive time out ’

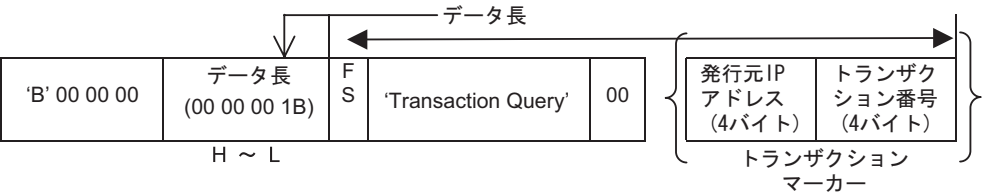
11.3 トランザクション結果問い合わせコマンド

イーサネット使用時のみ有効です。

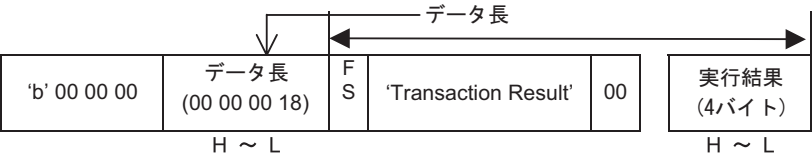
11.3.1 トランザクション結果問い合わせ

ホストから GP へ送信されるトランザクション結果問い合わせの内容は次のとおりです。

ホスト：コマンドデータ



GP：レスポンスデータ部



<設定内容>

- データ：' Transaction Query '

<実行結果>

- データ：' Transaction Result '

値意味

0x00000000 処理は正常に終了しました。

0x00000001 エラー

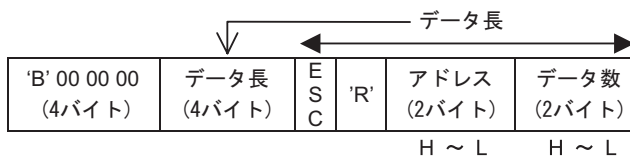
0x00000002 指定のトランザクションマーカーは GP にプールされていません。

11.4 コマンドフォーマット

11.4.1 Read フォーマット

■ LAN の場合

◆ ホスト コマンドデータ部



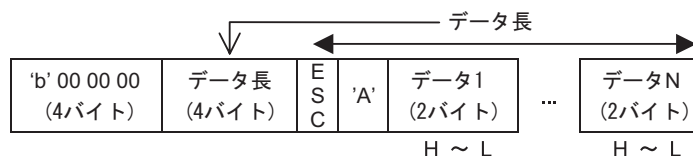
< 設定範囲 >

アドレス：0000H ~ 270FH(0 ~ 9999)

データ数：0001H ~ 0200H(1 ~ 512)

◆ GP レスポンスデータ部

- 正常時



< 設定範囲 >

データ：0000H ~ FFFFH

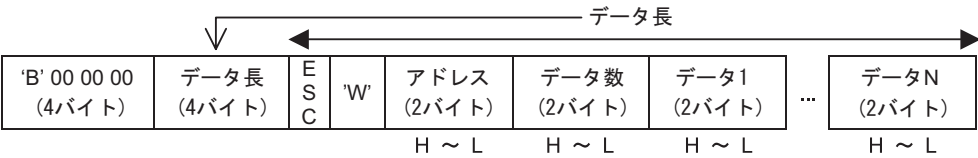
- 異常時

NAK で応答

11.4.2 Write フォーマット

■ LAN の場合

◆ ホスト コマンドデータ部



< 設定範囲 >

アドレス：0000H ~ 270FH(0 ~ 9999)

データ数：0001H ~ 0200H(1 ~ 512)

データ： 0000H ~ FFFFH

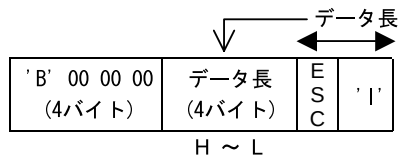
◆ GP レスポンスデータ部

ACK または NAK で応答

11.4.3 割り込み出力の問い合わせ

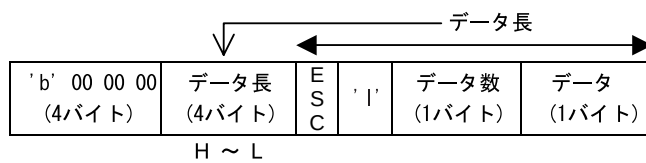
■ LAN

◆ ホスト コマンドデータ部



◆ GP レスポンスデータ部

- ・ 正常時



- ・ 異常時
NAK で応答

< 設定範囲 >

データ数

ホストから問い合わせコマンドが送信された場合はすでに発生している割り込み出力のデータ数のことをいいます。

データ

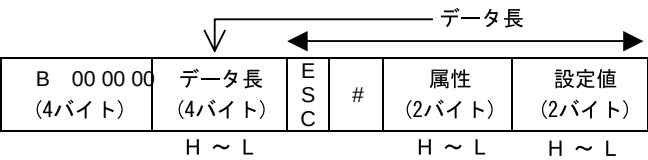
00H ~ FEH が出力されます。割り込み出力のデータがない場合は「00」となります。

11.4.4 輝度・コントラスト調整

輝度・コントラストをコマンドを用いて調整するためのコマンドデータ部の内容は以下のとおりです。表示器の機種によって輝度・コントラスト調整ができないものもあります。

■ LAN

◆ ホスト コマンドデータ部



< 設定範囲 >

属性：0000H ~ 0001H（0：コントラスト調整、1：輝度調整）

設定値：「■ 輝度・コントラスト一覧」（88 ページ）を参照してください。

全てアスキーコードで入力してください。

◆ GP レスポンスデータ部

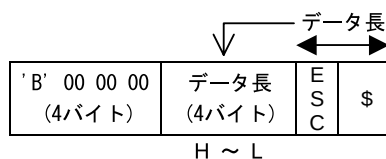
ACK または NAK で応答

11.4.5 輝度・コントラスト現在値

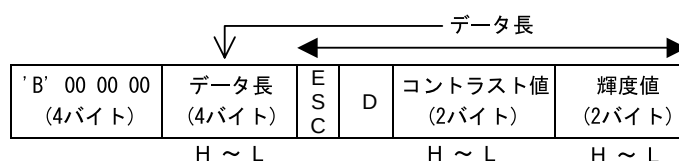
輝度・コントラストの現在値をコマンドを用いて取得するためのコマンドデータ部の内容は以下のとおりです。GP の機種によって輝度・コントラストがないものもあります。

■ LAN

◆ ホスト コマンドデータ部



◆ GP レスポンスデータ部



■ 輝度・コントラスト一覧

GP	輝度設定 設定範囲	コントラスト設定 設定範囲
AGP-3302B	0 (明) ~ 7 (暗)	0 (明) ~ 7 (暗)
AGP-3301L	0 (明) ~ 7 (暗)	0 (明) ~ 7 (暗)
AGP-3301S	0 (明) ~ 7 (暗)	0 (明) ~ 7 (暗)
AGP-3300L	0 (明) ~ 7 (暗)	0 (明) ~ 7 (暗)
AGP-3300S	0 (明) ~ 7 (暗)	0 (明) ~ 7 (暗)
AGP-3300T	0 (明) ~ 7 (暗)	-
AGP-3400S	0 (明) ~ 7 (暗)	0 (明) ~ 7 (暗)
AGP-3400T	0 (明) ~ 7 (暗)	-
AGP-3500L	0 (明) ~ 7 (暗)	0 (明) ~ 7 (暗)
AGP-3500S	0 (明) ~ 7 (暗)	0 (明) ~ 7 (暗)
AGP-3500T	0 (明) ~ 7 (暗)	-
AGP-3600T	0 (明) ~ 7 (暗)	-
AGP-3450T	0 (明) ~ 7 (暗)	-
AGP-3550T	0 (明) ~ 7 (暗)	-
AGP-3650T	0 (明) ~ 7 (暗)	-
AGP-3750T	0 (明) ~ 7 (暗)	-

12 メモリリンク API (イーサネット通信)

メモリリンク API は、アプリケーションからメモリリンクプロトコルを使用して GP にアクセスするとき、メモリリンクの詳細を知らなくても簡単にアクセスできるようにした Windows32 ビット API です。

12.1 メモリリンク API の使用方法

メモリリンク API ユーザーズはソケットを生成し、GP との通信経路を確保します。(通信経路を確保することを、コネクションを開設すると言います。)

GP との必要な通信を行い通信が終了すれば、GP との通信経路 (コネクション) を切断します。もし、そのソケットを使用して今後 GP との通信を行わないのであれば、ソケットを破棄します。通信をするのであれば、再度コネクションを開設します。(ソケットは再利用できます。)

■ 開発環境

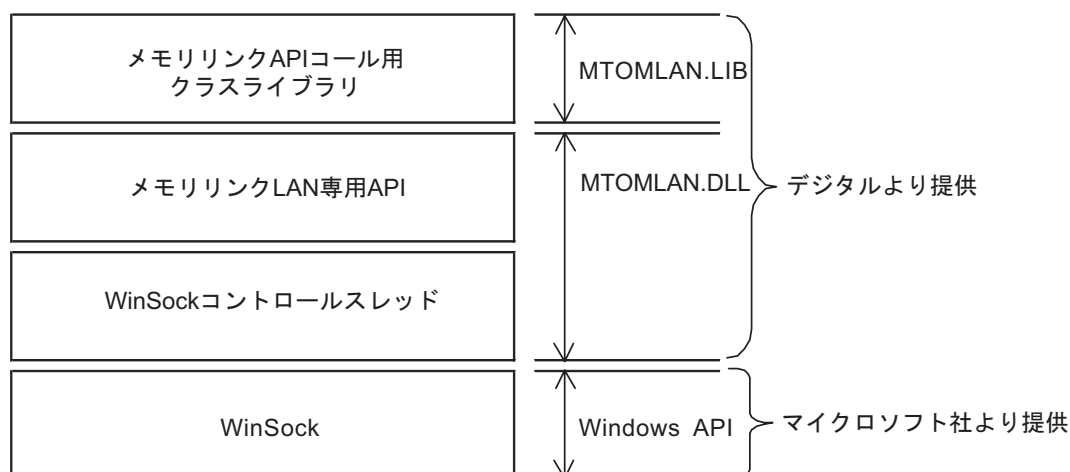
コンパイラ :Microsoft visual C++ Ver6.0

OS :Miceosoft Windows 98 以上

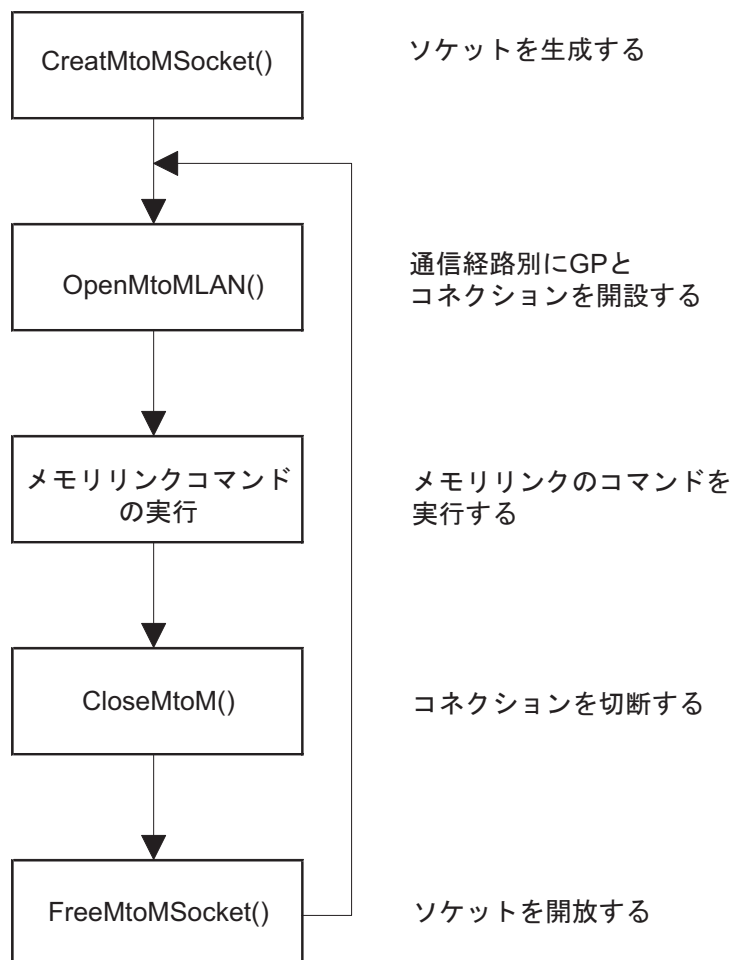
その他 :以下のファイルは GP-Pro EX の CD-ROM に収録しています。CD-ROM 内の「MTOMLAN」フォルダにある圧縮ファイル (MTOMLAN.ZIP) を解凍してください。

- ・ MTOMAPI.H
- ・ MTOMLAN.LIB
- ・ MTOMLAN.DLL

■ メモリリンク API のソフトウェア構成図



■ メモリリンク API の手順概要図



12.1.1 同期通信と非同期通信

同期通信とは、ある API をコールしたときその API から処理が正常 / 異常に関わらず終了するまで復帰しない通信方法を言います。

非同期通信は、API が終了する前に復帰し、次の処理ができる通信方法です。

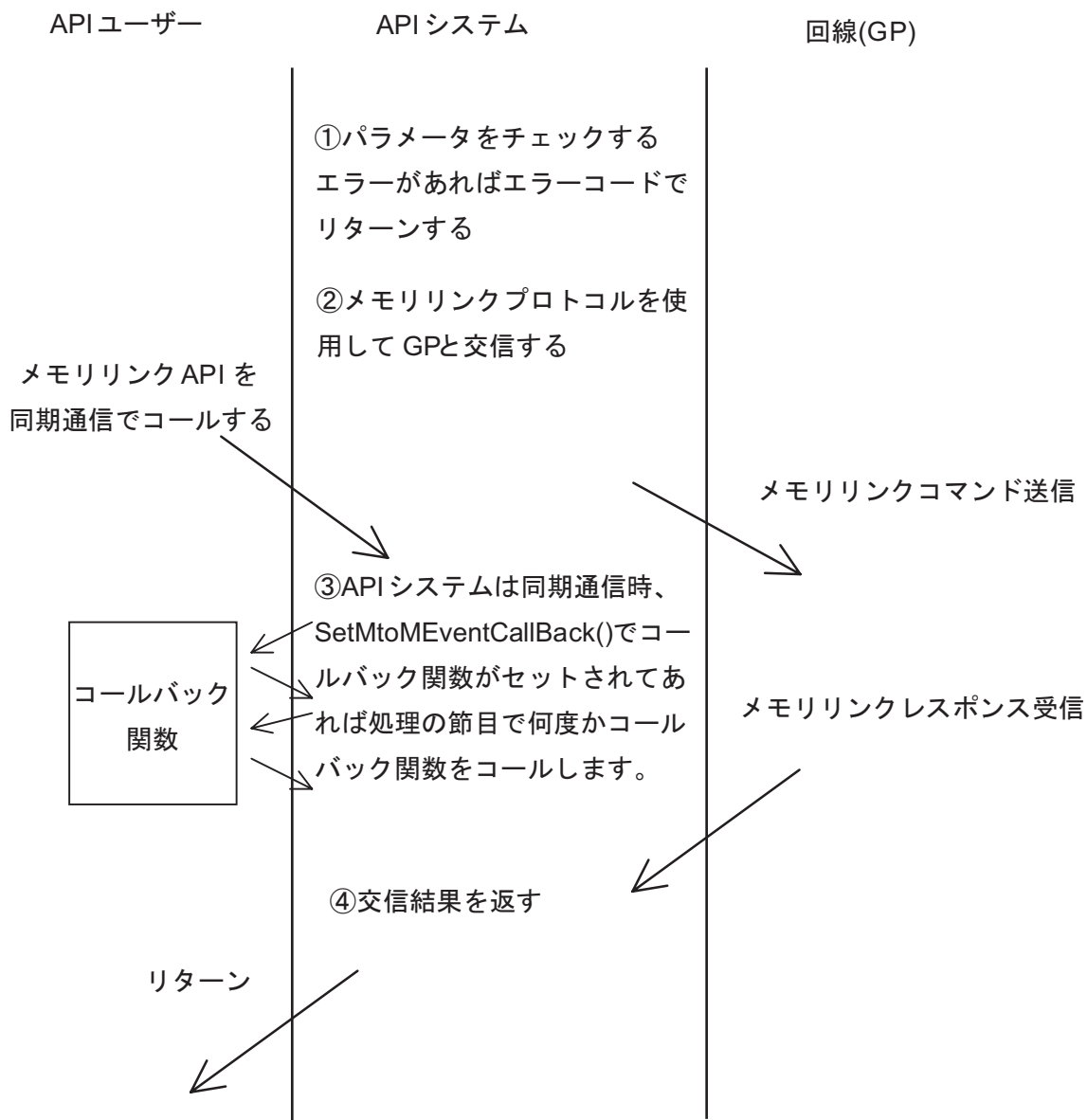
メモリリンク API は、同期 / 非同期両方の通信方法をサポートしています。

同期 / 非同期通信どちらを使用するかは、第 2 パラメータで指定します。

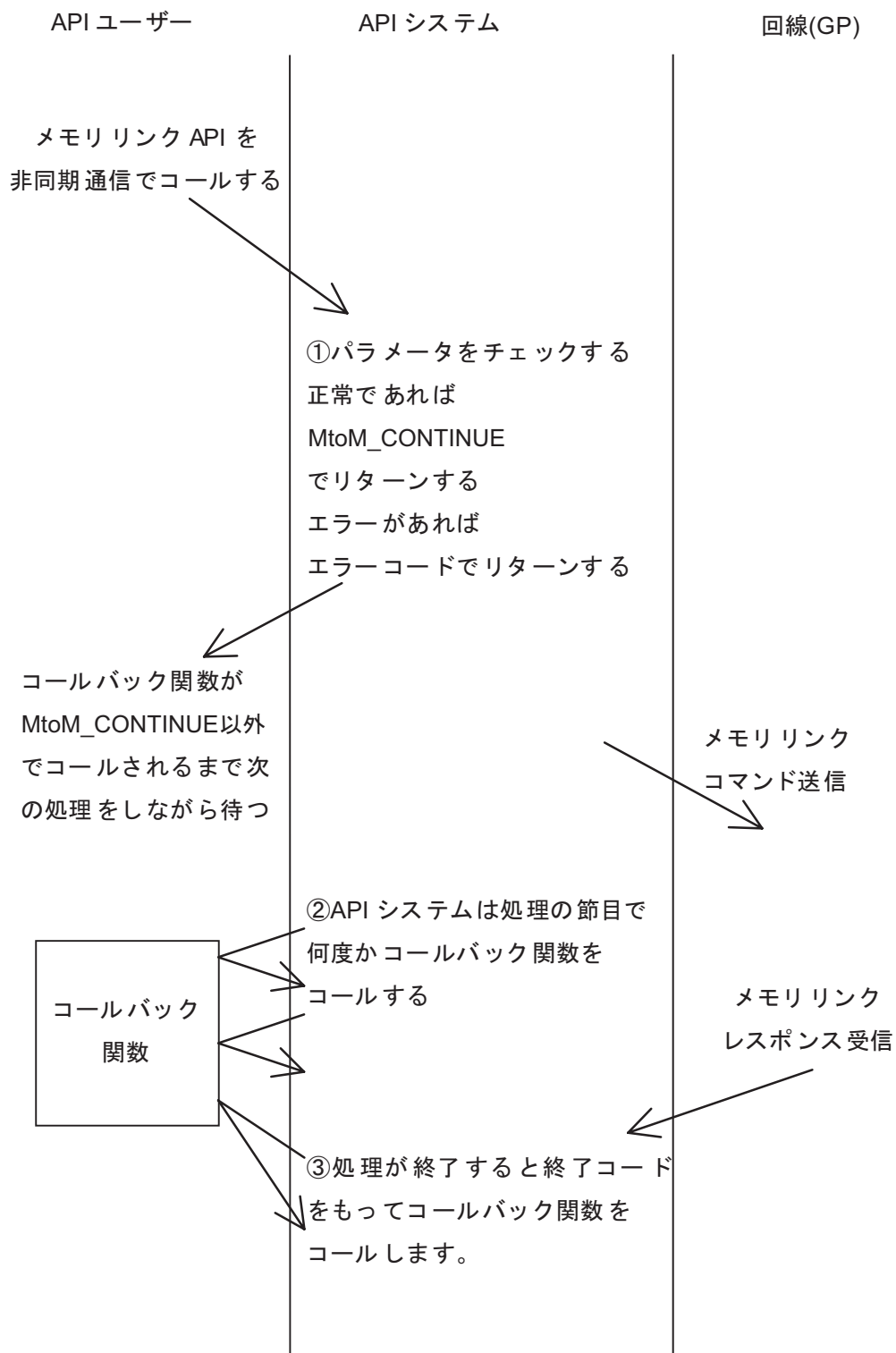
MEMO

- 第 2 パラメータが MTOMCALLBACK 型以外のものはすべて同期通信です。
 - 第 2 パラメータの MTOMCLLBACK 型の引数に NULL が指定されると同期通信になります。
 - 第 2 パラメータの MTOMCALLBACK 型の引数に NULL 以外が指定されるとシステムは非同期通信で、かつ、そのときのコールバック関数と判断し処理します。
-

■ 同期通信の手順



■ 非同期通信の手順



■ 非同期通信のキャンセル

非同期通信処理中に API の処理をキャンセルするには、2 通りの方法があります。

◆ FALSE を返す

メモリリンク API は、処理の節目でコールバック関数をコールします。そのとき、コールバック関数が FALSE を返すと、メモリリンク API は以降の処理を安全な状態で中断します。

◆ CancelMtoM() をコールする

メモリリンク API は処理を中断した後、MtoM_CANCEL コードでコールバック関数をコールします。このときソケットは不安定な状態になります。そのため API ユーザーは、CancelMtoM() 後必ず FreeMtoMSocket() をコールし、ソケットを開放してください。交信を続ける場合は、別のソケットで交信してください。
通信アプリケーションの強制終了のときなどに使用します。

■ 非同期通信のコールバック関数

非同期通信を行う場合、API ユーザーは非同期通信の処理の終了を知るために、コールバック関数を用意しなければなりません。

コールバック関数の型を下記に示します。

◆ 構文

MTOMCALLBACK FinisheMtoM(LPMtoMSOCK pMSock,int iMtoMCode)

◆ 引数

LPMtoMSOCK pMSock	処理の対象となったソケットハンドル
int iMtoMCode	処理結果 MTOM_OK: 処理は正常に終了しました。 MTOM_CONTINUE: 処理は継続中 Other: 何らかのエラーが発生し、処理は中断しました。

MEMO

- システムは処理中、処理の節目で iMtoMCode に MTOM_CONTINUE をセットし、コールバックします。

dwUser1 と dwUser2 はシステムが書き換えることはありません。

API ユーザーが自由に使用することができます。

通常、ソケットを識別する識別子使用するのが妥当です。

使用例

メモリリンクソケットをサポートした C++ のクラスを設計した場合クラスのコンストラクタで CreateMtoMSocket() をコールして、ソケットを作成し、そのソケットの dwUser1 にそのクラスのポインターをセットしておけばコールバック関数でクラスを使用することができます。

例題の動作説明

- (4) クラスのコンストラクタでクラスの this ポインターを dwUser1 にセットする。
- (5) SetMtoMEventCallBack() で API から何らかのイベント通知があったとき、最初にコールバックされる関数 (グローバルで得かつスタティックな関数) を登録する。
- (6) 実際に、何らかのイベントが発生すると SetMtoMEventCallBack() で登録した関数 (例題なら EventFuncJump()) がコールバックされる。
- (7) EventFuncJump() の中で dwUser1 からクラスポインターを取り出し、OnEventFunc() があたかも API からコールバックされたようになります。
- (8) 通常、OnEventFunc() を virtual 関数として宣言しオーバーライドすればさらに、使い勝手が良いです。

```
class CMtoMSock {
public:
    LPMtoMSOCK m_pMSock ;

    CMtoMSock();
    ~CMtoMSock();

    //API からのイベント通知が必要なときは、このメンバーをオーバーライドします。
    virtual void OnEventFunc(int iCode,DWORD dwParam1,DWORD dwParam2){}; // ⑤
};

// イベント発生時のコールバック関数
③
void CALLBACK EventFuncJump
(LPMtoMSOCK pMSock,int iCode,DWORD dwParam1,DWORD dwpara)
{
    CMSock* pCMSock ;

    pCMSock = (CMSock*)pMSock->swUser1 ;
```

```
        pCMSock->OnEventFunc(iCode,dwParam1,dwParam2) ;// ④
    }

CMSock::CMSock(DWORD dwProtocolType)
{
    if( m_pMSock = ::CreateMtoMSocket(dwProtocolType) ){
        m_pMSock->dwUser1 = (DWORD )this ; // ①
        ::SetMtoMEventCallBack(m_pMSock,EventFuncJump) ;// ②
    }
}
```


■ 特定局交信

特定局（ネットワークの情報エリアで処理対象に選ばれている局）に対して交信します。

レスポンスはネットワークの情報エリアで最初に処理対象となっている局からのレスポンスだけを有効として扱われます。つまり、先頭の局が、全局の代表として使用されます。本交信方法は複数局に対しての表示用コマンドなどに使用します。

ネットワークの情報エリアの局数が1局なら、通常の UDP/IP（一斉同報ではない）を使用し、複数なら UDP/IP の一斉同報を利用します。

基本的な使用方法是

- (1) CreateMtoMSocket() でソケットを作成する。
- (2) SetTransitionType() に不特定多数交信 (B_dwTransitionType_Specifict) を指定して交信方法を設定する。
- (3) 対象となるネットワークのネット I D をネットワーク情報 (pGPNetWorkData) の一斉同報時の対象ネット I D(dwNetID) にセットする。
- (4) 対象局をネットワーク情報にセットする。

このとき、対象局が明確的に分かっている場合は、MtoM_ResizeGPNetWorkData() をコールしてネットワーク情報のサイズを変更し、ネットワーク情報の局レコードに対象局の I P アドレス、とそのレコードが有効を示す為に、dwNodeStatus に B_dwNodeStatus_Find をセットする。不明確な場合は MtoMFS_FindNode() をコールし、ネットワークに参加している局を自動検索する。検索結果はネットワーク情報に反映されます。

MEMO	・ ネットワーク情報の先頭の局レコードは、交信中はネットワークを代表する局として扱われます。
-------------	--

- (5) MtoMESC_*() などを利用して交信を行う。
- (6) CloseMtoM() でコネクションを切断する。
- (7) FreeMtoMSocket() でソケットを開放する。

■ 厳密チェック付き特定局交信

特定局（ネットワークの情報エリアで処理対象に選ばれている局）に対して交信します。

特定局交信との違いは処理コマンド送信後、レスポンスはまず、ネットワーク情報エリアで最初に処理対象となっている局からのレスポンスだけを有効として扱い、その後、残りの局に対して、1局ずつ、今処理を行ったコマンドの正否を確認します。

ネットワークの情報エリアの局数が1局なら、通常の UDP/IP(一斉同報ではない)を使用し、複数なら UDP/IP の一斉同報を利用します。

- (1) CreateMtoMSocket() でソケットを作成する。
- (2) SetTransitionType() に不特定多数交信 (B_dwTransitionType_Specific) を指定して交信方法を設定する。
- (3) OpenMtoMLAN() でコネクションを開設する。
このとき相手局 IP アドレスには NULL を指定します。
- (4) 4. 対象局をネットワーク情報にセットする。

このとき、対象局が明確に分かっている場合は、MtoM_ResizeGPNetWorkData() をコールしてネットワーク情報のサイズを変更し、ネットワーク情報の局レコードに対象局の IP アドレス、とそのレコードが有効を示す為に dwNodeStatus に B_dwNodeStatus_Find をセットする。不明確な場合は MtoMFS_FindNode() をコールし、ネットワークに参加している局を自動検索する。検索結果はネットワーク情報に反映されます。

MEMO

- ・ ネットワーク情報の先頭の局レコードは、交信中はネットワークを代表する局として扱われます。

- (5) ネットワーク情報の各局レコードの処理の対象にするかどうかを示す (dwCheckButton) に TRUE をセットする。
- (6) MtoMESC_*() などを利用して交信を行う。
- (7) 個々の局が正常に処理を完了したかチェックするためネットワーク情報の各局レコードの局の状態 (dwNodeStatus) をチェックする。

これが B_dwNodeStatus_Nothing なら、この局は空の局レコードなので一切の処理を無視してください。

B_dwNodeStatus_Find : 処理は正常に終了しました。
B_dwNodeStatus_NotFind : 処理は異常終了しました。
B_dwNodeStatus_NonAction : この局は処理の対象外なので処理されなかった。これは、5. の dwCheckButton に TRUE をセットされなかった事を意味します。

- (8) dwNodeStatus を見て、リトライする場合は、リトライする局のみ dwCheckButton に TRUE をセットし、リトライしない局には FALSE をセットして再度 5. からやり直します。

(9) CloseMtoM() でコネクションを切断する。

(10) FreeMtoMSocket() でソケットを開放する。

12.2 基本コマンド

ここでは、メモリリンク API で使用される基本コマンドについて説明します。

■ 基本コマンド一覧

コマンド	アクション
CreateMtoMSocket	指定したプロトコルタイプのメモリリンク用ソケットを作成する
OpenMtoMLAN	メモリリンク LAN で指定した相手局とのコネクションを開設する
CloseMtoM	相手局とのコネクションを切断する
FreeMtoMSocket	ソケットを開放する
SetMtoMEventCallBack	メモリリンク API で何らかのイベント発生時、そのイベントを受けるための関数を登録する
CancelMtoM	現在処理中の非同期通信をキャンセルする
MtoM_ResizeGPNetWorkData	ソケット内のネットワーク情報のサイズを変更する
SetTransitionType	交信方法（トランザクションタイプ）の設定
GetTransitionType	現在設定されている交信方法（トランザクションタイプ）を取得する
MtoMGetLastError	エラー発生時、エラーの詳細を取得する

12.2.1 指定したプロトコルタイプのメモリリンク用ソケットを作成する

指定したプロトコルタイプのメモリリンク用ソケットを作成するコマンドは、以下のとおりです。
システムは本 API 内で、ソケットの資源を確保します。

■ 構文

LPMtoMSOCK WINAPI CreateMtoMSocket(DWORD dwProtocolType)

■ 戻り値

Other : 作成したソケットへのハンドル
NULL: 作成に失敗しました

■ 引数

DWORD dwProtocolType	使用するプロトコルの種類
B_ProtocolType_SIO:	メモリリンク SIO
B_ProtocolType_LAN:	メモリリンク LAN

12.2.2 メモリリンク LAN で指定した相手局とコネクションを開設する

メモリリンク LAN で指定した相手局とコネクションを開設するコマンドは、以下のとおりです。

■ 構文

```
int WINAPI OpenMtoMLAN(LPMtoMSOCK pMSock, MTOMCALLBACK pfFinish,  
LPCSTR szIPAddress)
```

■ 戻り値

pfFinish が NULL の時

00: 正常終了

Other: エラーコード

pfFinish が NULL 以外の時

MTOM_CONTINUE: 処理依頼は正常にシステムに通知されました。処理の完了は pfFinish が
コールバックされることで通知されます。

Other: エラーコード

■ 引数

LPMtoMSOCK pMSock	メモリリンクのソケットハンドル
MTOMCALLBACK pfFinish	NULL : 指定すると処理が完了するまで本 API は完了しません。 (同期型になります) 処理が完了するとその処理結果コードで復帰します。 NULL 以外 : 処理終了時にコールバックする関数へのポインタ。こ のパラメータが指定されると、本 API は処理をシステ ムに依頼するとすぐに MTOM_CONTINUE で復帰しま す。(非同期型になります) システムは処理が完了すると該当するソケットハンド ルと処理結果コードをもって、指定されたコールバッ ク関数をコールします。
LPCSTR szIPAddress	相手局 (GP) の IP アドレス 1 対 n 通信の場合は NULL を指定

MEMO

・ IP アドレスは、2 通りの方法で指定することができます。

(1) IP アドレスをドットで区切って指定する。

例 szipaddress="11.22.33.44";

(2) IP アドレスを局名で指定する。

例 szipaddress="GP1";

ただしこの場合は Windows フォルダに指定した局名にたいした IP アドレスが記述
された HOSTS ファイルが必要です。

C:\Windows\HOSTS の内容

例 11.22.33.44 GP1

12.2.3 相手局との TCP コネクションを切断する

相手局との TCP コネクションを切断するコマンドは、以下のとおりです。

■ 構文

```
int WINAPI CloseMtoM(LPMtoMSOCK pMSock, MTOMCALLBACK pfFinish)
```

■ 戻り値

Other : 作成したソケットへのハンドル
NULL: 作成に失敗しました

■ 引数

LPMtoMSOCK pMSock	メモリリンクのソケットハンドル
MTOMCALLBACK pfFinish	NULL : 指定すると処理が完了するまで本 API は完了しません。 (同期型になります) 処理が完了するとその処理結果コードで復帰します。 NULL 以外 : 処理終了時にコールバックする関数へのポインタ。このパラメータが指定されると、本 API は処理をシステムに依頼するとすぐに MTOM_CONTINUE で復帰します。(非同期型になります) システムは処理が完了すると該当するソケットハンドルと処理結果コードをもって、指定されたコールバック関数をコールします。

12.2.4 ソケットを開放する

ソケットを開放するコマンドは、以下のとおりです。

■ 構文

```
int WINAPI FreeMtoMSocket(LPMtoMSOCK pMSock)
```

■ 戻り値

Other : 作成したソケットへのハンドル
NULL: 作成に失敗しました

■ 引数

LPMtoMSOCK pMSock メモリリンクのソケットハンドル

12.2.5 イベントを受けるための関数を登録する

メモリリンク API で何らかのイベント発生時、そのイベントを受けるための関数を登録するコマンドは、以下のとおりです。

■ 構文

```
int WINAPI SetMtoMEventCallBack(LPMtoMSOCK pMSock, MTOMEVENTBACK pfEventFunc)
```

■ 戻り値

Other : 作成したソケットへのハンドル
NULL: 作成に失敗しました

■ 引数

LPMtoMSOCK pMSock メモリリンクのソケットハンドル
MTOMEVENTBACK pfEventFunc イベント発生時、コールバックされる関数。
 NULL を指定すると、コールバックされなくなります。

システムはイベント発生時、ソケット及びイベントコードと、情報（最大 32 ビットデータ 2 個）を持って指定された pfEventFunc をコールバックします。コールバックされる関数は以下の形をしていなければなりません。

システムは同期通信時、処理の節目にここで登録されたコールバック関数をコールします。しかし、非同期通信時は、ここで登録されたコールバック関数ではなく、処理の依頼をしたとき指定したコールバック関数をコールします。

MTOMEVENTBACK EventFunc

(LPMtoMSOCK pMtoMSOCK, int iMtoMCode, DWORD dwParam1, DWORD dwParam2);

LPMtoMSOCK pMtoMSOCK	ソケットハンドル
int iMtoMCode	イベントコード
DWORD dwParam1	第 1 情報
DWORD dwParam2	第 2 情報

コールバックされるイベントには以下のものがあります。

イベントコード	第 1 情報	第 2 情報	イベント内容
MTOM_EVENT_TOUCH	T タグコード	無意味	タッチパネルが押された
MTOM_EVENT_CLOSED	無意味	無意味	コネクションが切断された
MTOM_CONTINUE	無意味	無意味	同期通信中

12.2.6 現在処理中の非同期通信をキャンセルする

現在処理中の非同期通信をキャンセルするコマンドは、以下のとおりです。

■ 構文

int WINAPI CancelMtoM(LPMtoMSOCK pMSock)

■ 戻り値

Other : 作成したソケットへのハンドル
NULL: 作成に失敗しました

■ 引数

LPMtoMSOCK pMSock メモリリンクのソケットハンドル

MEMO

- この API コール後はソケットが不安定な状態になります。FreeMtoMSocket() をコールし、ソケットを解放してください。
-

12.2.7 ネットワーク情報のサイズを変更する

ネットワーク情報のサイズを変更するコマンドは、以下のとおりです。
管理している局レコード数が少ない場合、この API を利用して増減することができます。この API を
コールすると pMSock のネットワーク情報のエリアを示す pGPNetWorkData の値が変更されます。

■ 構文

```
int WINAPI MtoM_ResizeGPNetWorkData(LPMtoMSOCK pMSock,DWORD dwNodeCounter)
```

■ 戻り値

0: 正常にネットワーク情報のサイズを変更しました
Other: メモリ不足でサイズを変更できません

■ 引数

LPMtoMSOCK pMSock	メモリリンクのソケットハンドル
DWORD dwNodeCounter	希望する局レコードの数

12.2.8 **通信方法（トランザクションタイプ）を設定する**

通信方法（トランザクションタイプ）を設定するコマンドは、以下のとおりです。

■ 構文

DWORD WINAPI SetTransitionType(LPMtoMSOCK pMSock,DWORD dwTransitionType)

■ 戻り値

設定変更する前の通信方法（トランザクションタイプ）の設定値

■ 引数

LPMtoMSOCK pMSock メモリリンクのソケットハンドル

DWORD dwTransitionType 設定変更する通信方法

- B_dwTransitionType_Only1: 1局指定、コネクションを開設した局に対して通信を行います。デフォルトです。
(TCP/IP を利用します)
- B_dwTransitionType_BroadCast: 不特定多数の局に対して通信する（レスポンスチェックはしない）。この方法は相手局の処理スピードを一切考慮していないので、連続して通信すると相手局が処理仕切れない場合があります。
(UDP/IP の一斉同報を利用します)
- B_dwTransitionType_Specific: 特定局（ネットワークの情報エリアで処理対象に選ばれている局に対して）通信するレスポンスはネットワークの情報エリアで最初に処理対象となっている局からのレスポンスだけを有効として扱われます。つまり、先頭の局が、全局の代表として使用されます。
本通信方法は複数局に対しての表示用コマンドなどに使用します。
(UDP/IP の一斉同報を利用します)
- B_dwTransitionType_SpecificCheck: 特定局（ネットワークの情報エリアで処理対象に選ばれている局に対して）通信する。
B_dwTransitionType_Specific との違いは、処理コマンド送信後、レスポンスはまず、ネットワークの情報エリアで最初に処理対象となっている、局のレスポンスだけを有効として扱い、その後、残りの局に対して、1局ずつ処理の正否を確認します。
本通信方法は複数局に対しての厳密な処理を期待する場合、例えばファイル転送などに使用します。
(UDP/IP の一斉同報を利用します)

12.2.9 現在設定されている交信方法を取得する

現在設定されている交信方法（トランザクションタイプ）を取得するコマンドは、以下のとおりです。

■ 構文

DWORD WINAPI GetTransitionType(LPMtoMSOCK pMSock)

■ 戻り値

現在設定されている交信方法（トランザクションタイプ）の設定値
詳しくは SetTransitionType() の dwTransitionType パラメータを参照してください。

■ 引数

LPMtoMSOCK pMSock メモリリンクのソケットハンドル

12.2.10 エラー発生時、エラーの詳細を取得する

エラー発生時、エラーの詳細を取得するコマンドは、以下のとおりです。

■ 構文

DWORD WINAPI MtoMGetLastError(LPMtoMSOCK pMSock)

■ 戻り値

メモリリンク API 使用時、API がエラーを返したとき、そのエラーの詳細を返します。

■ 引数

LPMtoMSOCK pMSock メモリリンクのソケットハンドル

■ 解説

エラーの詳細には大別して 2 種類あります。1 つ目は GP との通信で GP が何らかのエラーを返してきた場合と、回線上のトラブルによるエラーに分かれます。

前者は、9999 以下の値を返します。ただし、0 は正常終了したことを示します。

後者は、10000 以上の値を返します。具体的には、メモリリンク API は内部的に Microsoft Visual C++ の Winsock を使用して、そのエラーコードがそのまま復帰値になります。

◆ GP 関係のエラーに関するエラーコード

参照 : GP-Pro EX リファレンスマニュアル

◆ Winsock 関係のエラーに関するエラーコード

コード	エラー	コード	エラー
10004	WSAEINTR	10053	WSAECONNABORTED
10009	WSAEBADF	10054	WSAECONNRESET
10013	WSAEACCES	10055	WSAENOBUFS
10014	WSAEFAULT	10056	WSAEISCONN
10022	WSAEINVAL	10057	WSAENOTCONN
10024	WSAEMFILE	10058	WSAESHUTDOWN
10035	WSAEWOULDBLOCK	10059	WSAETOOMANYREFS
10036	WSAEINPROGRESS	10060	WSAETIMEDOUT
10037	WSAEALREADY	10061	WSAECONNREFUSED
10038	WSAENOTSOCK	10062	WSAELOOP
10039	WSAEDESTADDRREQ	10063	WSAENAMETOOLONG

コード	エラー	コード	エラー
10040	WSAEMSGSIZE	10064	WSAEHOSTDOWN
10041	WSAEPROTOTYPE	10065	WSAEHOSTUNREACH
10042	WSAENOPROTYPE	10066	WSAENOTEMPTY
10043	WSAEPROTONOSUPPORT	10067	WSAEPROCLIM
10044	WSAESOCKTNOSUPPORT	10068	WSAEUSERS
10045	WSAEOPNOTSUPP	10069	WSAEDQUOT
10046	WSAEPFNOSUPPORT	10070	WSAESTALE
10047	WSAEAFNOSUPPORT	10071	WSAEREMOTE
10048	WSAEADDRINUSE	10091	WSASYSNOTREADY
10049	WSAEADDRNOTAVAIL	10092	WSAVERNOTSUPPORTED
10050	WSAENETDOWN	10093	WSANOTINITIALISED
10051	WSAENETUNREACH	10101	WSAEDISCON
10052	WSAENETRESET	-	

12.3 表示用コマンド

ここでは、メモリリンク API で使用される表示用コマンドについて説明します。

■表示用コマンド一覧

コマンド	アクション
MtoMESC_W	システムエリアヘータを書き込む
MtoMESC_R	システムエリアヘータを読み出す
MtoMESC_I	タッチパネルが押されたかどうか問い合わせる
MtoMESC_SetContrast	輝度 / コントラストの設定
MtoMESC_GetContrast	輝度 / コントラストの取得

12.3.1 システムエリアヘータを書き込む

システムエリアヘータを書き込むコマンドは、以下のとおりです。

■ 構文

```
int WINAPI MtoMESC_W  
(LPMtoMSOCK pMSock, MTOMCALLBACK pfFinish, WORD wAddress, INT iDataCount, WORD* pwData)
```

■ 戻り値

pfFinish が NULL のとき

00: 正常終了
Other: エラーコード

pfFinish が NULL 以外のとき

MTOM_CONTINUE: 処理依頼は正常にシステムに通知されました。
処理の完了は pfFinish がコールバックされることで通知されます。
Other: エラーコード

■ 引数

LPMtoMSOCK pMSock メモリリンクのソケットハンドル

MTOMCALLBACK pfFinish NULL: 指定すると処理が完了するまで本 API は完了しせん。
(同期型になります)
処理が完了するとその処理結果コードで復帰します。

NULL 以外: 処理終了時にコールバックする関数へのポインタ。このパラメータが指定されると、本 API は処理をシステムに依頼するとすぐに MTOM_CONTINUE で復帰します。(非同期型になります)

システムは処理が完了すると該当するソケットハンドルと処理結果コードをもって、指定されたコールバック関数をコールします。

WORD wAddress	システムエリアの書き込み先のアドレスを設定します。 0000h ~ 0FFFh
INT iDataCount	書き込みデータ数を設定します。 0001h ~ 0200h (1 ~ 512)
WORD* pwData	書き込むデータ

12.3.2 システムエリアからデータを読み出す

システムエリアからデータを読み出すコマンドは、以下のとおりです。

■ 構文

```
int WINAPI MtoMESC_R  
(LPMtoMSOCK pMSock, MTOMCALLBACK pfFinish, WORD wAddress, INT iDataCount, WORD pwoData)
```

■ 戻り値

pfFinish が NULL のとき

00: 正常終了

Other: エラーコード

pfFinish が NULL 以外るとき

MTOM_CONTINUE: 処理依頼は正常にシステムに通知されました。処理の完了は pfFinish がコールバックされることで通知されます。

Other: エラーコード

■ 引数

LPMtoMSOCK pMSock	メモリリンクのソケットハンドル
MTOMCALLBACK pfFinish	NULL: 指定すると処理が完了するまで本 API は完了しません。 (同期型になります) 処理が完了するとその処理結果コードで復帰します。 NULL 以外: 処理終了時にコールバックする関数へのポインタ。このパラメータが指定されると、本 API は処理をシステムに依頼するとすぐに MTOM_CONTINUE で復帰します。(非同期型になります) システムは処理が完了すると該当するソケットハンドルと処理結果コードをもって、指定されたコールバック関数をコールします。
WORD wAddress	システムエリアの読み出し先のアドレスを設定します。 0000h ~ 0FFFh
INT iDataCount	読み出すデータ数を設定します。 0001h ~ 0200h (1 ~ 512)
WORD pwoData	読み出したデータの格納先

MEMO

- システムは pwoData が示すバッファのバッファサイズはチェックしませんので十分なバッファサイズを API ユーザー側で用意してください。

12.3.3 タッチパネルが押されたかどうか問い合わせる

タッチパネルが押されたかどうか問い合わせるコマンドは、以下のとおりです。

本 API が正常完了した後、pbHave をチェックし、TURE であれば pdwCode を参照してください。

■ 構文

```
int WINAPI MtoMESC_I  
(LPMtoMSOCK pMSock, MTOMCALLBACK pfFinish, BOOL* pbHave, DWORD *pdwCode)
```

■ 戻り値

pfFinish が NULL のとき

00: 正常終了

Other: エラーコード

pfFinish が NULL 以外のとき

MTOM_CONTINUE: 処理依頼は正常にシステムに通知されました。処理の完了は pfFinish がコールバックされることで通知されます。

Other: エラーコード

■ 引数

LPMtoMSOCK pMSock	メモリリンクのソケットハンドル
MTOMCALLBACK pfFinish	NULL: 指定すると処理が完了するまで本 API は完了しません。 (同期型になります) 処理が完了するとその処理結果コードで復帰します。 NULL 以外: 処理終了時にコールバックする関数へのポインタ。このパラメータが指定されると、本 API は処理をシステムに依頼するとすぐに MTOM_CONTINUE で復帰します。(非同期型になります) システムは処理が完了すると該当するソケットハンドルと処理結果コードをもって、指定されたコールバック関数をコールします。
BOOL* pbHave	タッチパネルが押されたかどうかを返すエリア 本 API 完了後、このエリアが TRUE ならタッチパネルが押された事を意味し、pdwCode にそのコードがセットされます。
DWORD *pdwCode	タッチパネルが押されたとき、そのコードをセットするエリア

12.3.4 輝度 / コントラストの設定

輝度／コントラストの設定コマンドは、以下のとおりです。

■ 構文

DWORD WINAPI SetContrast

(LPMtoMSOCK pMSock, MTOMCALLBACK pfFinish, DWORD dwContrast, DWORD dwLight)

■ 戻り値

設定変更する前の輝度／コントラストの設定値

■ 引数

LPMtoMSOCK pMSock	メモリリンクのソケットハンドル
MTOMCALLBACK pfFinish	NULL： 指定すると処理が完了するまで本 API は完了しません。 (同型になります) 処理が完了するとその処理結果コードで復帰します。 NULL 以外： 処理終了時にコールバックする関数へのポインタ。このパラメータが指定されると、本 API は処理をシステムに依頼するとすぐに MTOM_CONTINUE で復帰します。(非同期型になります。) システムは処理が完了すると該当するソケットハンドルと処理結果コードをもって、指定されたコールバック関数をコールします。
dwContrast	コントラスト調整 (0000h ～ 0007h) 0: 明～ 7: 暗 FFFFFFFFh で設定値無し (設定がない機種)
dwLight	輝度調整 (0000h ～ 0007h) 0: 明～ 7: 暗 FFFFFFFFh で設定値無し (設定がない機種)

MEMO

・ 輝度 / コントラストの設定範囲については一覧をご覧ください。

👉 「■ 輝度・コントラスト一覧」(88 ページ)

12.3.5

輝度 / コントラストの取得

輝度／コントラストの取得コマンドは、以下のとおりです。

■ 構文

DWORD WINAPI GetContrast

(LPMtoMSOCK pMSock, MTOMCALLBACK pfFinish, DWORD*dwContrast, DWORD *dwLight)

■ 戻り値

現在設定されている輝度／コントラストの設定値

■ 引数

LPMtoMSOCK pMSock メモリリンクのソケットハンドル

MTOMCALLBACK pfFinish	NULL :	指定すると処理が完了するまで本 API は完了しません。 (同型になります)
-----------------------	--------	---

処理が完了するとその処理結果コードで復帰します。

NULL 以外： 処理終了時にコールバックする関数へのポインタ。このパラメータが指定されると、本 API は処理をシステムに依頼するとすぐに MTOM_CONTINUE で復帰します。（非同期型になります）

システムは処理が完了すると該当するソケットハンドルと処理結果コードをもって、指定されたコールバック関数をコールします。

dwContrast コントラストの現在値

(0000h ~ 0007h) 0: 明 ~ 7: 暗

FFFFFFFFh で設定値無し (設定がない機種)

dwLight 輝度の現在値

(0000h ~ 0007h) 0: 明 ~ 7: 暗

FFFFFFFFh で設定値無し (設定がない機種)

MEMO

- 輝度/コントラストの設定範囲については一覧をご覧ください。

👉「■輝度・コントラスト一覧」(88 ページ)

12.4 API 戻り値のエラーコード一覧

	コード	内容
MTOM_OK	00	処理は正常に終了しました
MTOM_CONTINUE	01	処理は継続中です
MTOM_USER_STOPED	03	ユーザーアプリケーションによって処理が中断されました。 (MtoMStop() がコールされたか、コールバック関数で FALSE が返されました)
MTOM_EVENT_TOUCH	40	タッチパネルが押されました (API の復帰値ではありませんが SetMtoMEventCallBack() で登録されたイベント発生時、コールバックされる関数には通知されます)
MTOM_EVENT_CLOSED	41	コネクションが切断されました
MTOM_ERROR	80	GP からエラーレスポンスを受けました
MTOM_ERROR_INVALID	81	API パラメータエラー、もしくは API が不正にコールされました
MTOM_ERROR_LAN	82	回線上でエラーが発生しました (Winsock がエラーを返しました)
MTOM_ERROR_TOUT_RES	83	レスポンスタイムアウトです
MTOM_ERROR_TOUT_CHAR	84	キャラクタ間タイムアウト GP からのフレームが途中で途絶えた
MTOM_ERROR_NAK	85	GP から NAK が返信されました

MEMO

- エラー応答時のエラーの詳細の取得には MtoMGetLastError() をコールしてください。

13 サンプルプログラム（イーサネット通信）

ここでは、付属のメモリリンク LAN API を使用したサンプルプログラム (AGPM.EXE) について説明します。AGPM.EXE は Windows のホストと GP を 1 対 1、もしくは 1 対 n で接続し、ホストからリアルタイムに GP 上のメモリをアクセスできるサンプルプログラムです。

13.1 メモリリンク LAN API サンプルプログラム

■ 起動環境

- (1) AGPM.EXE は Windows 98 上で動作します。
- (2) AGPM.EXE は MtoMLAN.DLL を使用するので MtoMLAN.DLL を Windows フォルダにコピーしてください。
- (3) メモリリンク LAN は TCP/IP を使用します。TCP/IP をインストールして下さい。具体的にはスタートのコントロールパネルのネットワークでプロトコルに Microsoft 社の TCP/IP をインストールしてください。

MEMO

- AGPM.EXE を起動した時、“DLL: LAN initialize error” と表示され起動できない場合は、TCP/IP の設定が正しく行われていない可能性があります。もう一度、設定を見直してください。

■ 開発環境

AGPM.EXE は以下の環境で開発されています。

サンプルプログラムのソースコードは GP-Pro EX の CD-ROM に収録しています。CD-ROM 内の「MTOMLAN」フォルダにあるソースコードを以下の環境でコンパイルすると実行ファイル (AGPM.EXE) が生成されます。

コンパイラ : Microsoft Visual C++ Ver.6.0

O S : Microsoft Windows 98

■ メモリリンク API アクセス方法

AGPM.EXE はメモリリンク API アクセスする為に、クラス CMSock を定義し、使用しています。クラス CMSock はメモリリンク API を 1 ソケット 1 オブジェクトの単位で完全に内包しています。また、メモリリンク API からのコールバックはクラス CMSock のメソッドをオーバーライドする事により実装しています。

■ CMSock の派生

AGPM.EXE はクラス CMSock から 2 つのクラスを派生して使用しています。

一つは 1 対 n モード時およびノード検索用にクラス CGpMApp が継承しています。

もう一つは 1 対 1 用のクラス CGpMDoc が継承しています。

CGpMApp は AGPM のアプリケーションクラスです。CGpMDoc は AGPM のドキュメントクラスです。

つまり、1 対 n 関係の操作はアプリケーションクラスが担当し、1 対 1 関係はドキュメントクラスが担当しています。

■ クラス CGpMDOC

AGPM.EXE の中核となっているクラスです。ドキュメント データを管理しています。コンテンツオブジェクトをオブジェクトの配列として内部に持ちます。また、1 対 1 モードでの GP とコネクションも管理しています。

■ クラス CGpMView

関係付けられている CGpMDoc のコンテンツオブジェクトをウィンドウ上に表示します。

■ MtoMAPI.H と MtoMLAM.LIB

AGPM.EXE は MtoMAPI.H を外部からインクルードしています。

MtoMAPI.H は [MtoMLAN] フォルダに入っています。これを適当なフォルダにコピーし、そのコピーした位置を defsfile.h の #include 文を変更する事で指定して下さい。

また、MtoMLAN.DLL を呼び出すために MtoMLAM.LIB をインクルードしています。これも同様に適当なディレクトリーにコピーし、その位置を [設定] の [リンカー] の [オブジェクト / ライブラリーモジュール] にて指定して下さい。