

機器接続マニュアル



機器接続マニュアルに関する注意事項

本書を正しくご使用いただくために、ご使用前に必ず「マニュアルPDFをダウンロードする前に」をお読みいただき、「はじめに(商標権などについて、対応機種一覧、マニュアルの読み方、表記のルール)」マニュアルをダウンロードしてください。ダウンロードされたマニュアルは、必ずご利用になる場所のお手元に保管し、いつでもご覧いただけるようにしておいてください。

第 1 章

メモリリンク方式

ここでは、メモリリンク方式について説明します。

またイーサネット通信に必要なソフトウェア構成についても説明します。

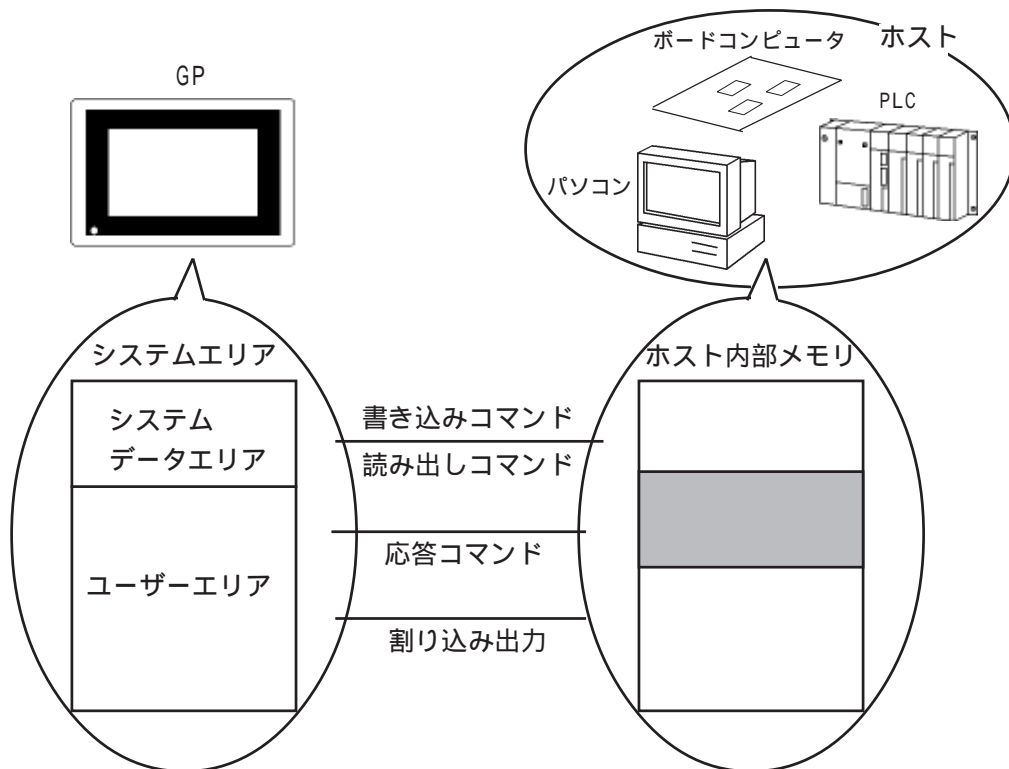
1-1

メモリリンク方式のしくみ

GPとホストとのデータのやり取りは、ホスト側のプログラムに従って行われます。

GPはホストの書き込みコマンドによって送られる表示用データにもとづいて、画面表示を行います。また、GPはホストの読み出しコマンドに従って、格納しているデータをホストに渡します。このように、GPとホストとの通信では、常にホスト側が主導権を持っています。

GPとホストとのデータのやり取りは、GP内部に設けられた記憶領域（システムエリア）を介して行われます。



1-1-1 システムエリアとは

システムエリアはGPとホストがデータのやり取りを行うための媒体となるエリアです。システムエリアはGPの内部に設けてあり、GPはシステムエリアのデータにもとづいて画面表示を行っています。

システムエリアのサイズは8192ワード(*1)です。システムエリアを構成する各エリアは次のとおりです。

0 : 19	システムデータ エリア
20 : 2032	ユーザーエリア
: 2047	特殊リレー
2048 : 2095	予約
2096 : 8191*1	ユーザーエリア
8192 : 8999	予約
9000 : 9999	拡張システムエリア*2

システムデータエリア

GPの画面制御データやエラー情報など稼働に必要なデータを書き込む領域です。各アドレスで書き込む内容が決まっています。参照 1-1-2 システムデータエリアの内容と領域

ユーザーエリア

GPとホストがデータのやり取りを行うエリアです。

ホスト側では、GPのどのアドレスにデータを書き込むかを決め、書き込むためのプログラムを作成します。GP側ではアドレスに書き込まれたデータを表示するため、別途設定(部品やタグの設定)を行います。また、Kタグ(テンキー入力)やTタグ(タッチパネル入力)によって書き込まれたデータをホストに読み込むためには、ホスト側で、GPのデータを読み出すためのプログラムを作成する必要があります。

*1 GP2000 シリーズ以外は 4096 ワード(0 から 4095)となります。

*2 GP2000 シリーズのみ対応しています。

1-1-2 システムデータエリアの内容と領域

システムデータエリアの各アドレスに書き込むデータの内容を示します。

- 重要** ・ 通常、画面表示のOFFを行う場合には、「コントロール」のバックライトOFFのビットを使用せず、「画面表示のON/OFF」をご使用ください。ダイレクトアクセス方式の場合とシステムデータエリアの内容が異なりますのでご注意ください。

アドレス	内容	機能	ビット	備考	
1	ステータス ^{*1}		0、1	予約	
			2	プリント中 ^{*2}	
			3	設定値書き込み ^{*3}	
			4~7	予約	
			8	Kタグ入力エラー ^{*4}	
			9	表示0:ON、1:OFF ^{*5}	
			10	バックライト切れ検出 ^{*6}	
			11	タッチパネル入力異常 ^{*7}	
			12~15	予約	
			3	エラーステータスGPのエラー発生時に、対応するビットがONされます。 一度ONになったビットは、電源をOFFしてから再度ONするか、オフラインモードから再度運転モードに切り替えるまで保持されます。エラーステータスの内容詳細と処理については、「1-1-4 システムデータエリアの内容と領域」の末尾をご参照ください。	
2	システムROM/RAM				
3	画面記憶メモリチェックサム				
4	SI0フレミング				
5	SI0パリティ				
6	SI0オーバーラン				
7、8	未使用				
9	内部記憶メモリの初期化が必要				
10	タイマクロック異常				
11~15	未使用				
4	時計データ 「年」	「年、月、日、時、分」 のデータがそれぞれBCD2桁 で格納されています。 <例> 1992年2月1日 17時15分	0~7	BCD2桁で西暦の下2桁のデータを格納	
5	時計データ 「月」		8~15	未使用	
6	時計データ 「日」		0~7	BCD2桁で01~12の月データを格納	
7	時計データ 「時」		8~15	未使用	
8	時計データ 「分」		0~7	BCD2桁で01~31の日付データを格納	
10	割り込み出力 ^{*15} (タッチOFF時)		8~15	未使用	
11	コントロール ^{*8}			0~7	BCD2桁で00~23の時間データを格納
				8~15	未使用
		0		バックライト ^{*9}	
		1		ブザーON	
		2		プリント開始	
		3		予約	
		4		ブザー音 ^{*10} 0:出力 1:非出力	
		5		AUX出力 0:出力 1:非出力	
		6		タッチパネルを押す事により表示OFFからONへ変更した時の割り込み出力 ^{*11} (割り込みコード:FFh) 0:割り込み出力しない 1:割り込み出力する	
		7		予約	
		8		VGA表示 ^{*12} 0:非表示 1:表示	
		9、10		予約	
11	ハードコピー出力 ^{*13} 0:表示、1:非出力キャンセル				
12~15	予約				

アドレス	内容	機能	ビット	備考
12	画面表示の ON/OFF ^{*14}	FFFFhならば画面表示が消えます。0hの場合は画面表示します。FFFFh、0h以外の値は予約		
13	割り込み出力 ^{*15}	GPのタッチタグなどのデータを使って絶対書き込みでデータを書くと、下位8ビットの内容が割り込みコードとして出力されます。(FFhは出力しません。)		
15	表示画面番号 ^{*17}	画面番号を書き込むと表示画面が切り替わります。	0～14	切り替え画面番号1～8999 (ただしBCD入力の場合は1～1999)
			15	強制画面切り替え 0:通常、1:強制画面切替え
16	ウインドウ ^{*16} コントロール		0	表示 0:OFF、1:ON
			1	ウインドウの重なり順序の入れ替え0:可、1:不可
			2～15	予約
17	ウインドウ ^{*16} 登録番号	間接指定で指定したグローバルウインドウの登録番号です。(BINまたは、BCD)		
18	ウインドウ ^{*16} 表示位置 (X座標データ)	間接指定で指定したグローバルウインドウの表示位置です。(BINまたは、BCD)		
19	ウインドウ ^{*16} 表示位置 (Y座標データ)			

*1 <ステータス>

- ・必要ビットのみをビット単位でモニタしてください。
- ・予約ビットはGPのシステムでメンテナンスなどに使用している場合がありますので、ON/OFFは不定です。

*2 <ステータス - プリント中>

プリント中にビットがONします。このビットのON中にオフラインモードへ切り替えると、プリント出力が乱れる場合があります。

*3 <ステータス - 設定値書き込み>

Kタグおよび設定値表示器による書き込みが発生するごとにビットが反転します。

*4 <ステータス - Kタグ入力エラー>

現在入力中のKタグに警報が設定されている場合、警報レンジ外の値を入力すると、ビットがONします。警報レンジ内の値を入力する、または画面が切り替わるとOFFになります。

*5 <表示 ON / OFF のステータス>

表示 ON / OFF を PLC から検出することが可能です。また、このビットは、以下の場合に変化します。

- (1) システムデータエリアの表示 ON / OFF (リンクタイプ時 LS9) に FFFF を書き込み表示を OFF した場合 (ビット 9 = 1)
- (2) スタンバイ時間が経過し、自動で表示 OFF になった場合 (ビット 9 = 1)
- (3) 表示 OFF 時から画面切替などで表示 ON になった場合 (ビット 9 = 0)
- (4) システムデータエリアのコントロールのバックライト OFF (ビット 0) ではこのビットは変化しません。

*6 <バックライト切れ検出>

バックライト切れを検出するとビットがONします。ただし、GP-377R/GP-377/GP-2000/GLC2000シリーズのみ。

*7 <タッチパネル入力異常>

タッチパネルの同一箇所に入力状態が設定時間以上続いた場合にONします。

*8 <コントロール>

予約ビットはGPのシステムでメンテナンスなどに使用している場合がありますので、必ずOFFにしてください。

*9 <コントロール - バックライト>

GP-477R/GP-470シリーズ以外の場合、ONでバックライトが消灯(LCD表示はそのまま)し、OFFで点灯します。

システムデータエリア「コントロール」のバックライトOFFのビットをONにすると、バックライトのみがOFFになっている状態で、LCD(液晶)は表示ONのままになっています。また、画面に設定されているタッチスイッチなども動作する状態となっています。通常、画面表示のOFFを行う場合は、「画面表示のON/OFF」をご使用ください。

*10 <コントロール - ブザー音>

コントロールのビット1(ブザーON)時の出力先は、以下のようになります。
ブザー音・・・コントロールのビット1がONの間、GP内部のブザーが鳴ります。
AUX出力・・・コントロールのビット1がONの間、AUXのブザー出力がONします。

*11 <タッチパネルを押す事により表示OFFからONへ変更した時の割り込み出力>

- ・タッチパネルからの表示ONの場合のみ割り込みが出力されます。
- ・GP-H70の場合は、後面オペレーションスイッチからの表示ONでは割り込みは出力されません。

*12 <コントロール -VGA 表示>

GP-570VM、GP-870VMの場合、ONで画面全体がVGA表示となります。VGA表示中に画面の任意の位置をタッチするとOFFしVGA非表示になります。

*13 <ハードコピー出力>

コントロールのビット11(ハードコピー出力)をONにすることにより、現在印字中の画面ハードコピーを中止します。

- ・ハードコピーの中止後、コントロールのビット11のOFFされませんので、ステータスのプリント中ステータスを監視するなどして、コントロールのビット11をOFFしてください。
- ・コントロールのビット11がONの間は、ハードコピーは行われません。すべて中止されることとなります。印字途中で中止を行った場合、画面1ライン分のデータを出力し、終わってから中止されます。また、すでにプリンタ側のバッファに取り込まれているデータはクリアされません。

*14 <画面表示のON/OFF>

システムデータエリア「画面表示のON/OFF」で画面表示OFFを行うと、画面表示OFF後の1回目のタッチ入力画面表示ONとしての動作となります。

*15 <割り込み出力>

アドレス10、13に、00～1Fのコントロールコードを書き込まないでください。通信ができなくなる場合があります。

*16 <ウィンドウコントロール/ウィンドウ登録番号/ウィンドウ表示位置>

ウィンドウ 参照 ウィンドウ表示<Uタグ>

*17 <表示画面番号>

ホストからの画面切り替えとタッチパネルやDスクリプトからの画面切り替えを併用した場合、希望どおりに画面が切り替わらないことがあります。これを回避するには、画面切り替えをホスト側からの画面切り替えに統一してください。タッチパネルやDスクリプトからはアドレス13番の割り込み出力を利用してホストに信号を伝え、ホスト側から画面を切り替えるようにしてください。

- 禁止**
- ・ アドレス0,2,9,14は予約領域です。データの書き込みは行わないでください。
 - ・ アドレス3,12,13,15はシステム制御で利用しているため、タグによる表示は行わないでください。
 - ・ アドレス12,13,15はワード単位で制御しているため、ビット書き込みはできません。
 - ・ アドレス12に「FFFFh」を書き込むと、表示中の画面が瞬時に消えます。GPオフラインモードの初期設定で指定したスタンバイモード時間で画面表示を消したい場合は、アドレス12には「0000h」を書き込んでください。
 - ・ アドレス10、13に、00～1Fのコントロールコードを書き込まないでください。通信ができなくなる場合があります。

1-1-3 特殊リレー

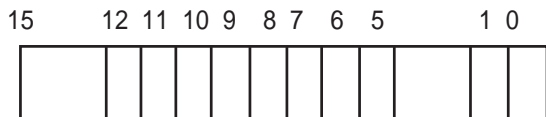
特殊リレーの構成は次のとおりです。

2032	共通リレー情報
2033	ベース画面情報
2034	予約
2035	1秒バイナリカウンタ
2036	タグのスキャンタイム
2037	予約
2038	タグのスキャンタイム
2039	予約
:	
:	
2047	

予約

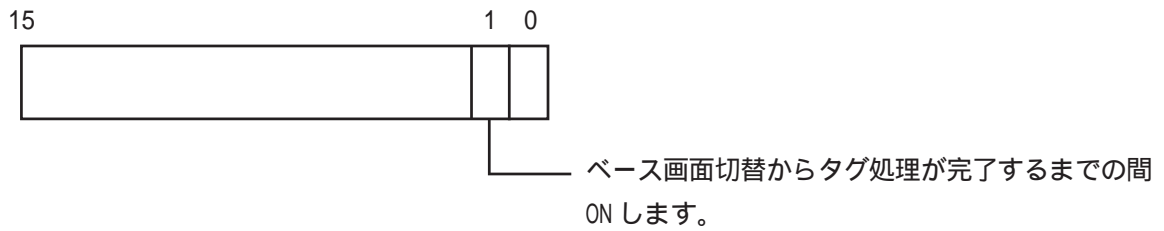
予約アドレスの値は不定です。使用しないでください。

共通リレー情報(2032)



0	予約
1	画面(ベース、ウィンドウ)切り替えからタグ処理が完了するまでの間ONになります。
2	予約
3	電源投入直後の初期画面を表示している間ONになります。
4	常時ONになっています。
5	常時OFFになっています。
6	バックアップSRAMのデータが消えたときにONします。(バックアップSRAM搭載のGPのみ)
7	Dスクリプト使用時、BCDエラーが発生するとONになります。 Dスクリプト参照 タグリファレンスマニュアル3.1 Dスクリプト
8	Dスクリプト使用時、ゼロ割算エラーが発生するとONになります。
9	ファイリングデータでバックアップSRAMに転送できなかった場合にONします。
10	ファイリングデータのコントロールワードアドレスによる転送で、PLC SRAMの転送ができなかった場合にONします。 また、ファイル項目表示器によるPLC間の転送で、転送完了ビットアドレスが有りの場合のみ、PLC エリア、PLC SRAMの転送ができなかった場合にONします。
11	ファイリングデータでファイル項目表示器によるSRAM LSエリア間の転送中の際ONになります。
12	Dスクリプト使用時、memcpy()、アドレスオフセット指定の読み出しで通信エラーが発生するとONになります。正常にデータ読み出しが終了するとOFFになります。
13-15	予約

ベース画面情報(2033)



1秒バイナリカウンタ(2035)

電源投入直後より1秒ごとにカウントアップします。データはバイナリです。

タグのスキャンタイム (2036)

表示画面に設定されているタグの一つ目の処理開始から最後のタグの処理終了までの時間です。データはバイナリで単位はmsで格納されます。データは対象タグの全処理が完了した時点で更新されます。データの初期値は0です。± 10msの誤差があります。

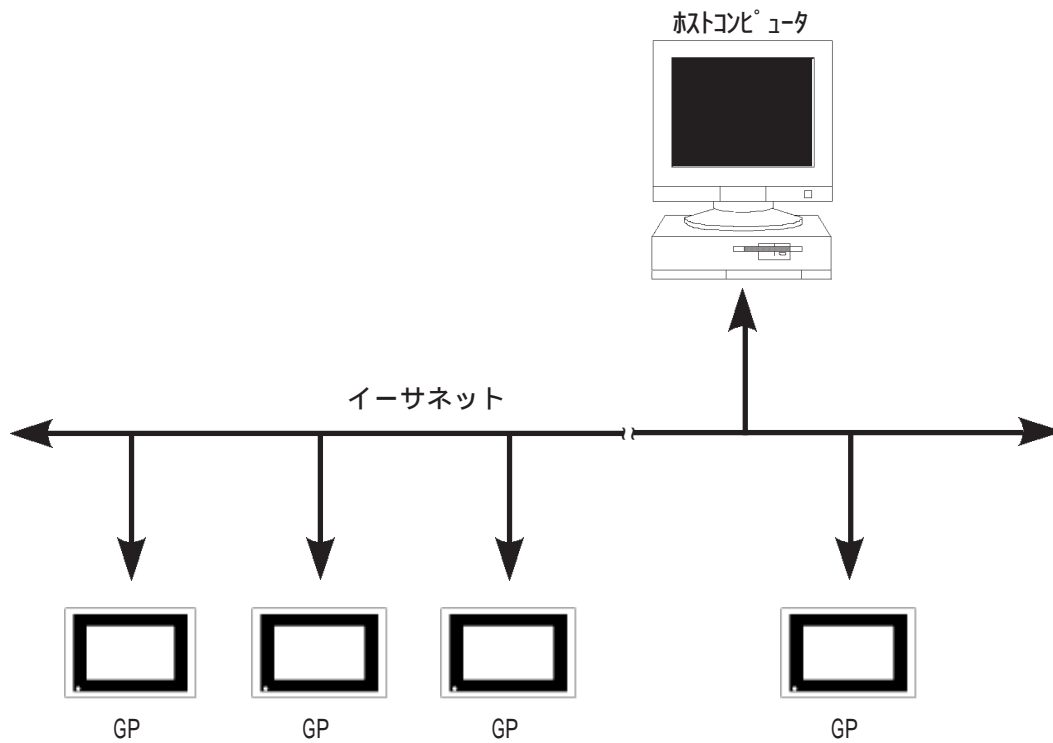
タグのスキャンカウンタ(2038)

表示画面に設定されているタグの処理がひとつとおり完了するごとにカウントアップされます。データはバイナリです。

禁止： ・ 特殊リレーはライトプロテクトされていません。タグなどで
ON/OFFしないでください。

1-2 システム構成

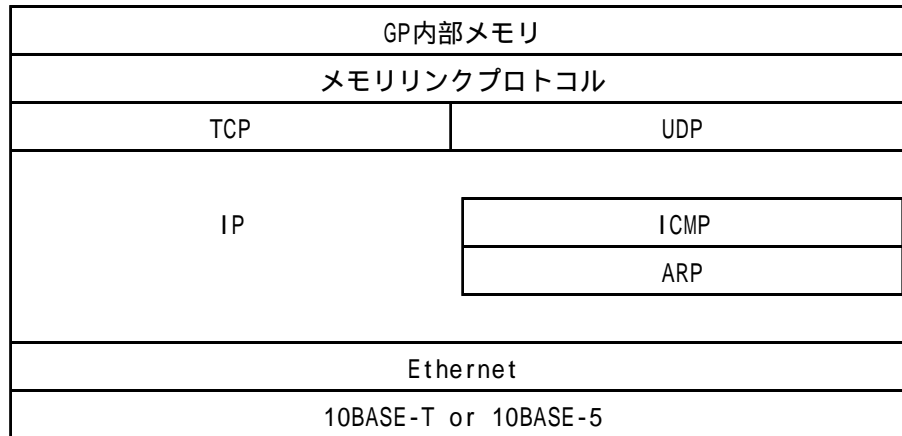
イーサネットユニットを接続することにより、GPをイーサネットに接続することができます。



- ・ 接続台数(n)はホストコンピュータの仕様によります。
- ・ GP70/GP77Rシリーズの場合はオプションのイーサネット拡張ユニットが必要です。

1-3 ソフトウェア構成

このイーサネット用プロトコルでは、TCP/IPおよびUDP/IPプロトコルをサポートしています。



TCP(Transmission Control Protocol)

TCPプロトコルはデータの信頼性、確実性を保証するプロトコルです。

- ・コネクションを確立することにより、論理接続を行います。
- ・コネクションは1つのみ確立することができます。
- ・シーケンス番号による順序制御やデータの再送機能、チェックサムによりデータの信頼性を確保します。

UDP(User Datagram Protocol)

UDPプロトコルはデータの信頼性、確実性を保証しないプロトコルです。相手局にデータが届かなかった場合でも再送信は行いません。

- ・コネクションレスでの通信が可能です。
- ・通信データの信頼性を上げるためにチェックサム付となっていますが、より信頼性が必要な場合は、TCPをご使用ください。

IP(Internet Protocol)

- ・データグラム形式での通信データの送受信を行います。

ARP(Address Resolution Protocol)

- ・ブロードキャストにより、IPアドレスからイーサネットアドレスを求めます。

ICMP(Internet Control Message Protocol)

- ・エコーリクエストに対しての応答をサポートしています。

第2章

運転までの設定と手順

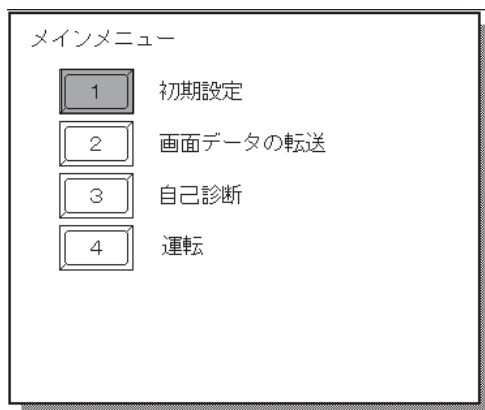
ここでは、GPをイーサネットにて通信を行うための設定とGPとホストコンピュータとの基本的な通信手順について説明します。

2-1

設定方法

通信を行うためには、イーサネット情報の設定(IPアドレス、ポート番号)を行う必要があります。

オフラインにて通信プロトコルに関する設定手順を以下に示します。

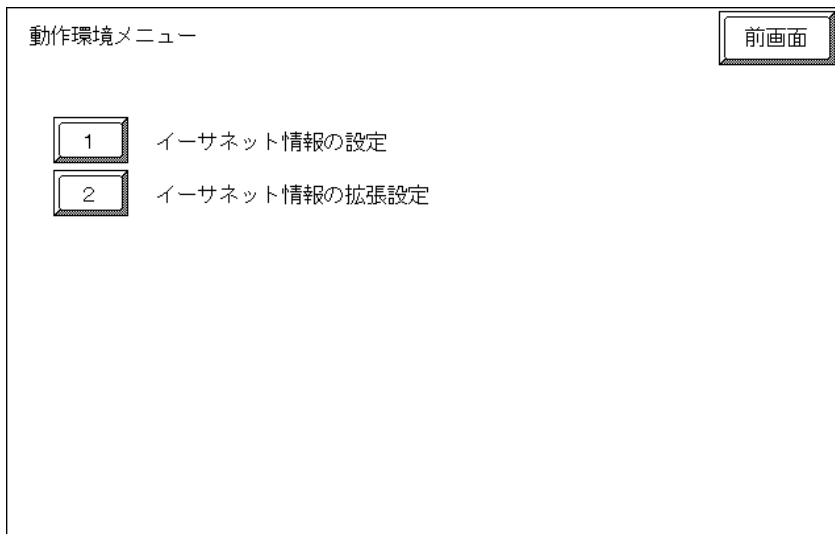


1の「初期設定」を選択します。



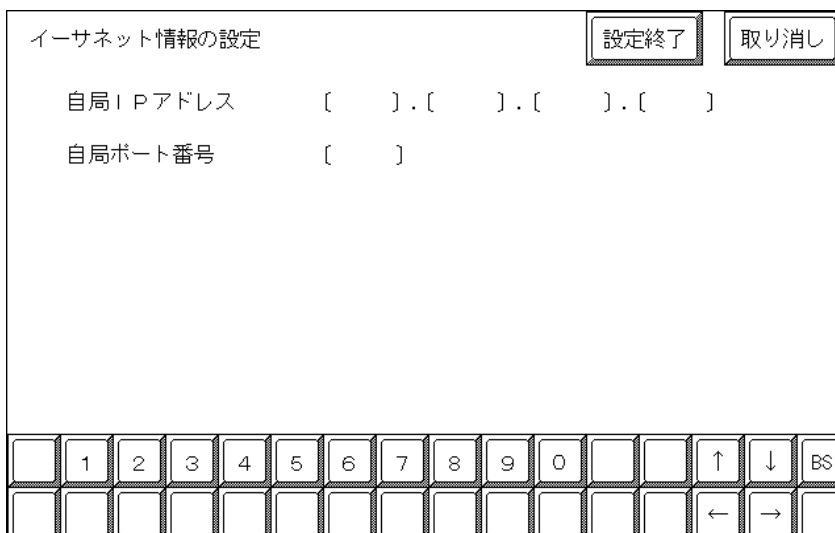
3の「動作環境の設定」を選択します。

「動作環境メニュー」画面が表示されます。



イーサネット情報の設定

1の「イーサネット情報の設定」を選択し各項目を設定します。



・自局 IP アドレス

自局 (GP) の IP アドレスを設定します。

IP アドレスは全 32 ビットを 8 ビットごとの 4 つの組に分け、それぞれをドットで区切った 10 進数で入力してください。

・自局ポート番号

自局 (GP) のポート番号を設定します。

1024 から 65535 の値を設定できます。

- 重要**
- ・自局 IP アドレス、自局ポート番号はネットワーク管理者に、ご確認ください。
 - ・自局 IP アドレスは重複しないようにしてください。自局 IP アドレスが重複しますと正しい通信ができません。
 - ・2way ドライバと同じポート番号は使用しないでください。2way ドライバのデフォルトのポート番号は 8000 ~ 8009 です。

イーサネット情報の拡張設定

2の「イーサネット情報の拡張設定」を選択し各項目を設定します。

イーサネット情報の拡張設定												設定終了	取り消し		
送信ウエイト	[]	(ms)											
TCPタイムアウト値	[]	(× 2sec)											
IPルータアドレス	[].	[].	[].	[]			
サブネットマスク	[].	[].	[].	[]			
	1	2	3	4	5	6	7	8	9	0			↑	↓	BS
													←	→	

- 送信ウエイトタイム (0 ~ 255)

GPからのコマンド送信時にウエイト時間を入れることができます。
通信回線上の通信負荷が多い場合、ご使用ください。
設定が必要ない場合は、0に設定してください。

- TCP タイムアウト値 (0 ~ 65535)

TCPのタイムアウト値です。設定した時間内に、相手局より応答がない場合タイムアウトになります。0に設定すると、デフォルト値として15秒に設定されます。

- ルータ IP アドレス

ルータのIPアドレスを設定します。(ルータの設定は一つのみです。)
ルータを使用しない場合は、全て0に設定してください。

- サブネットマスク

サブネットマスクを設定します。
使用されない場合は、全て0に設定してください。

- 重要**
- ・オフラインよりメモリの初期化をした場合、各設定値に不定値が設定が設定されることがあります。必ず設定値をご確認ください。
 - ・ルータIPアドレス、サブネットマスクはネットワーク管理者にご確認ください。

2-2 コネクション

イーサネット対応メモリリンクプロトコルでは、TCP/IP および UDP/IP の2つのコネクションをサポートしています。

個々のコネクションには以下の特徴があります。

- ・TCP/IP コネクションは1対1の交信用で一斉同報交信はできません。
- ・UDP/IP コネクションではGP からのアクション(たとえばタッチ入力<ESC I>など)は行えません。

	TCP/IPコネクション	UDP/IPコネクション
1対1のトランザクション	可能	可能
ホストからの一斉同報トランザクション	不可	可能
GPからのアクション	可能	不可

2-2-1 開設手順

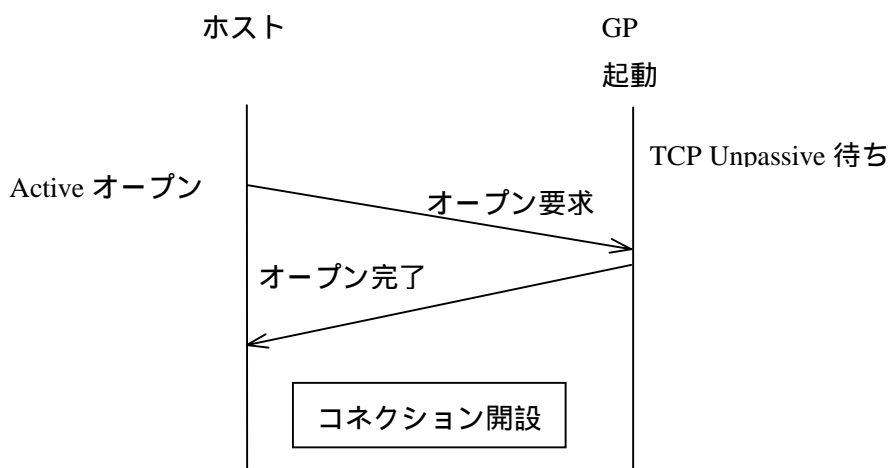
TCP コネクション

GP 側

GP は運転モードになるとコネクション開設要求(Unpassive)待ちになり、ホストからのコネクション開設要求を無限に待ちます。

ホスト側

GP に対し、開設要求(Active オープン)を行います。それに対し、GP が応えることでコネクションが開設されます。



UDP コネクション

GP 側

GP は運転モードになるとUDP 交信可能状態になります。

ホスト側

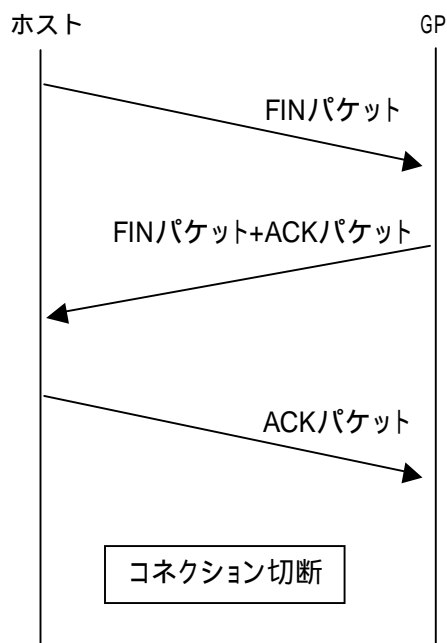
特別なアクションは必要ありません。

TCP コネクション

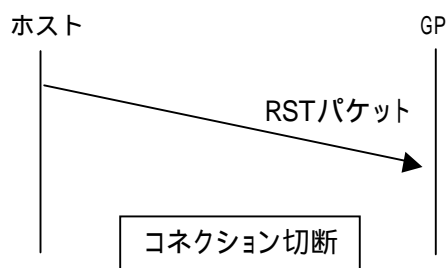
ホスト側

ホストはGPとの通信が終了した場合やTCP/IPレベルでエラーが発生した場合に、任意のタイミングで以下のようにコネクションを切断することができます。

- ・ FINパケットの場合



- ・ RSTパケットの場合



コマンド送信後、レスポンス待ちやコマンド送信中など、どのような状況でもコネクションが切断されることがあります。

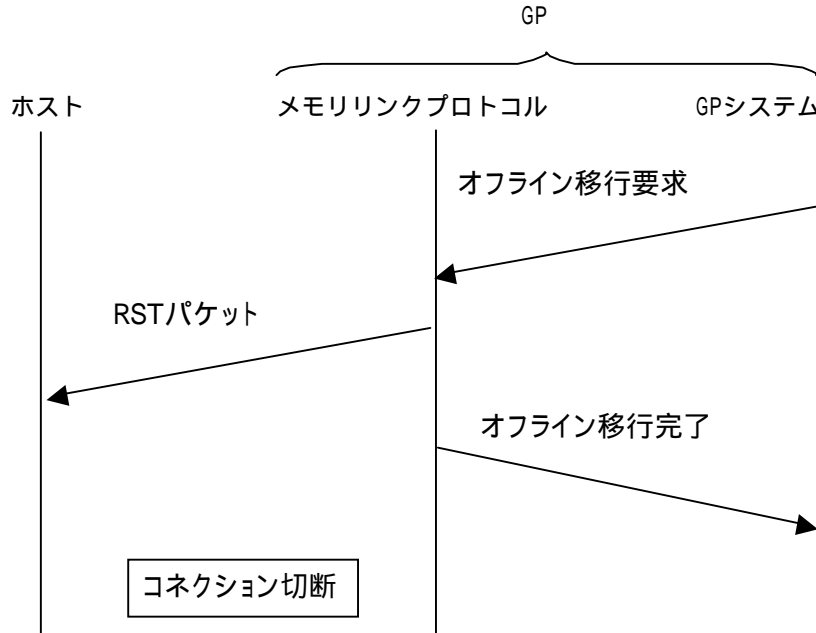
GP 側

次のような場合にGP側からコネクションを切断します。

GPはコネクションが切断されると、オフラインへ移行しない限り再接続待ちになります。

・GPがオフラインになった場合

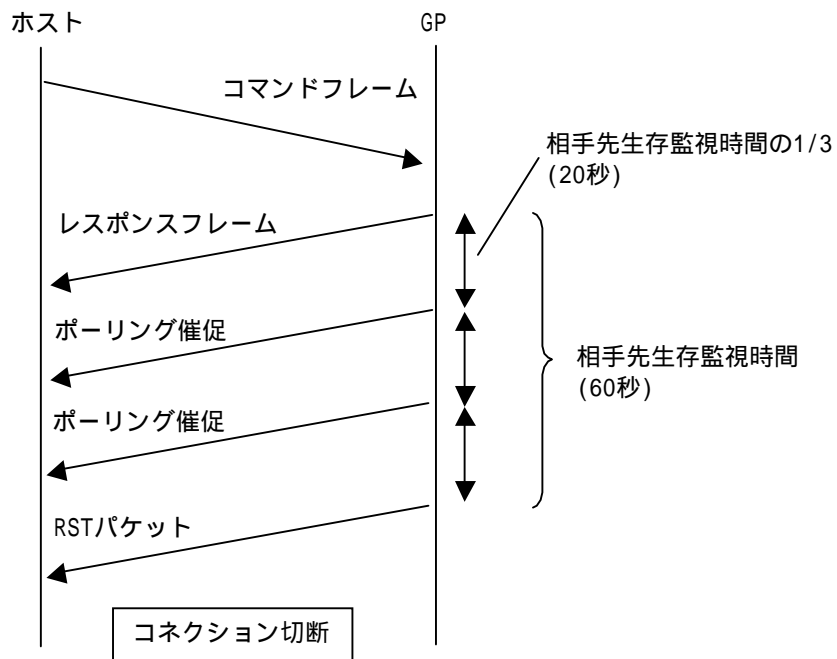
GPがオフラインに移行した場合、GP から RST パケットを送ることでコネクションを切断します。



・相手先生存監視時間以上経ってもホストからの通信がない場合

相手先生存監視時間以上経ってもホストからの通信がない場合、RST パケットを送ることでコネクションを切断します。

ただし、GP は相手先生存監視時間の 1/3 ごとにポーリングの催促コマンドをホストに送ります。ホストはポーリングの催促コマンドを受け取ると、ポーリングコマンドを GP に送るよう処理してください。このコマンドを GP が受けると相手先生存監視タイマをリセットします。



- ・TCP/IPレベルでエラーが発生した場合

TCP/IPレベルでエラーが発生した場合、RSTパケットを送り、コネクションを切断します。

ポーリングコマンドについて

一定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ります。

ホストはポーリングの催促コマンドを受け取ると、ポーリングコマンドをGPへ送るよう処理してください。ホストが送ったポーリングコマンドを受け取ることで、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はコネクションを切断します。

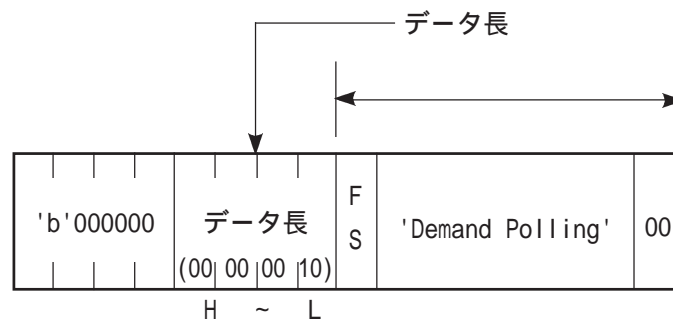
ホスト側でデジタル製のAPIを使用している場合、ポーリングの催促コマンドの応答処理はAPIで自動的に行われます。

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。

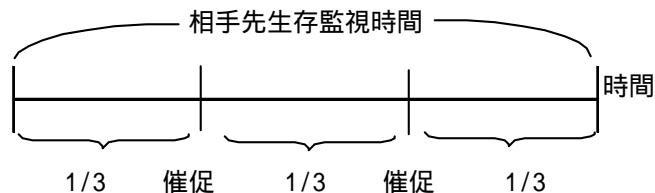
ホスト なし

GP レスポンスデータ部



< 設定内容 >

- ・データ： 'Demand Polling'



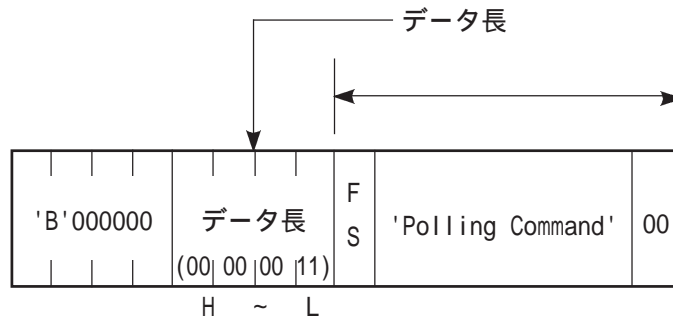
相手先生存監視時間の1/3のタイミングでポーリングの催促コマンドを送信します。

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。

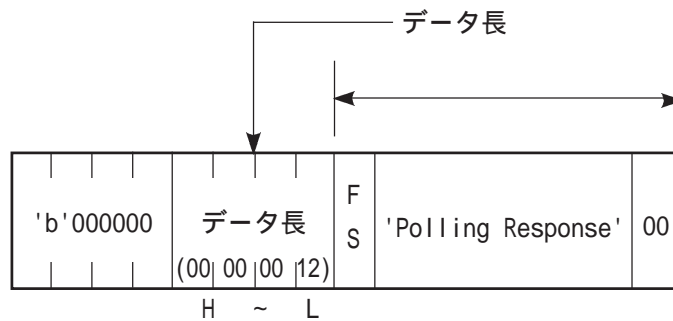
2 ポーリングコマンド(FS Polling)

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

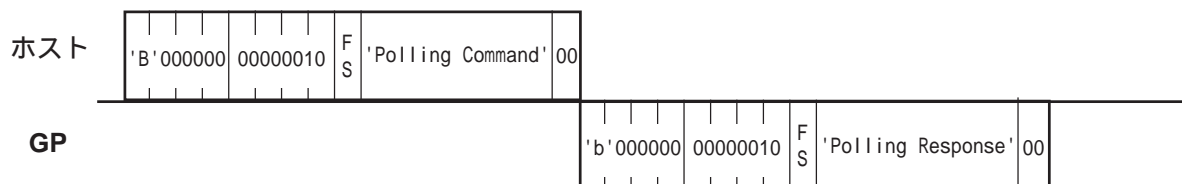


< 設定内容 >

- ・データ: 'Polling Command'

< 例題 >

ホスト側から GP 側へポーリングコマンドを送信し、ホストが正常に動作していることを確認します。



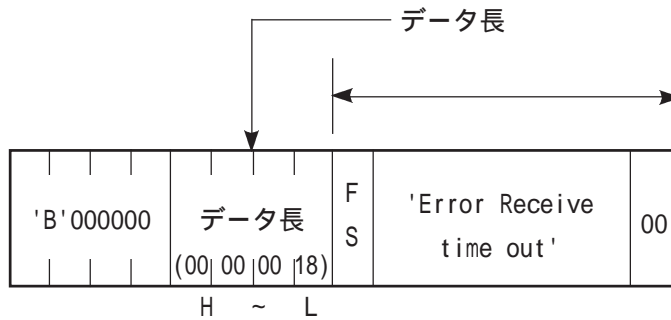
3 エラー通知(FS Error)

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

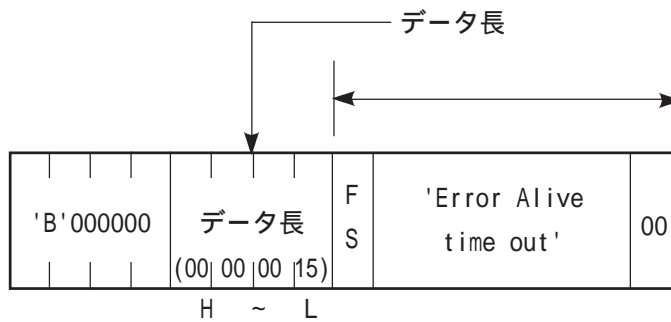
- ・キャラクター間タイムアウトエラーフレーム



<設定内容>

- ・データ： 'Error Receive time out'

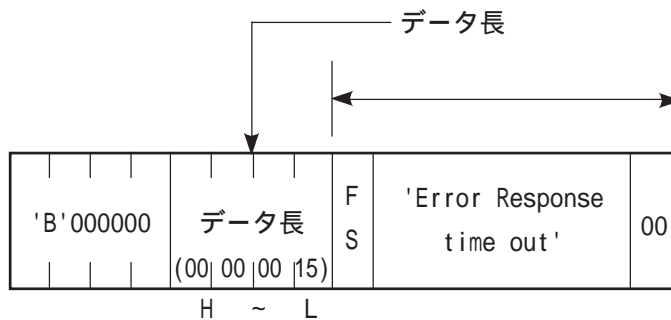
- ・相手先生存監視タイムアウトエラーフレーム



<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

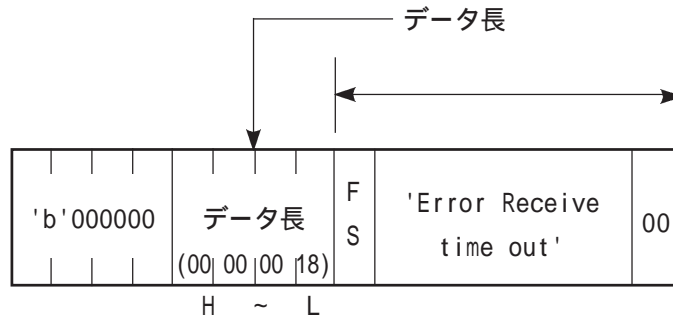


<設定内容>

- ・データ： 'Error Response time out'

GP レスポンスデータ部

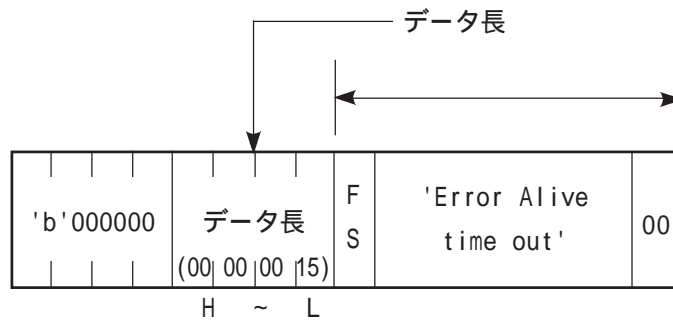
・キャラクター間タイムアウトエラーフレーム



<設定内容>

- ・データ: 'Error Receive time out'

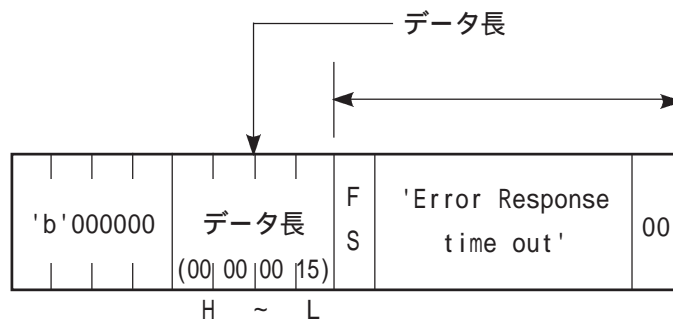
・相手先生存監視タイムアウトエラーフレーム



<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 Response time out'

2-3 通信プロトコルの制御手順

ホストとGPの通信コマンドはある一定の手順で設定され、通信しています。ここでは、基本的なコマンドとコマンドの設定について説明します。

本マニュアルで使用されるコマンド

本マニュアルで使用されるコマンドについて説明します。

コマンド

ある一定の手順で設定される通信コマンドの集合体を以下に示します。

コマンドデータ部： ホストコンピュータからの指令コマンド

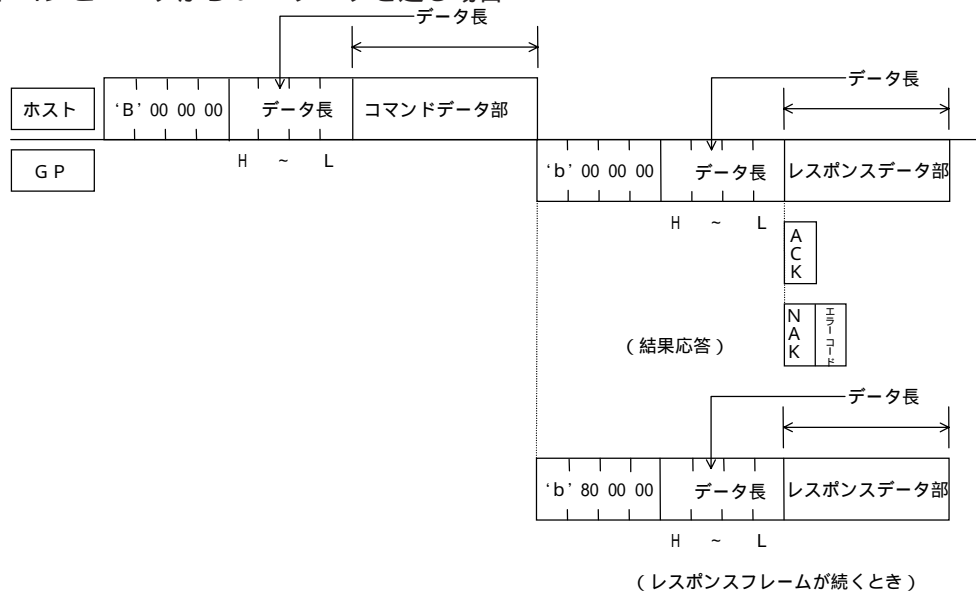
レスポンスデータ部： コマンドに対する本機の応答コマンド

割り込み出力データ部： GPのタッチ応答コマンド

2-3-1 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。

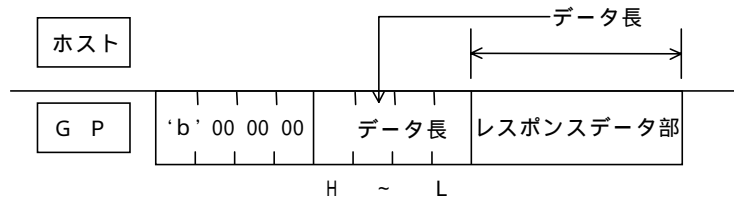
ホストコンピュータからGPへデータを送る場合



コマンドデータ部は、ホストよりGPへの転送データです。

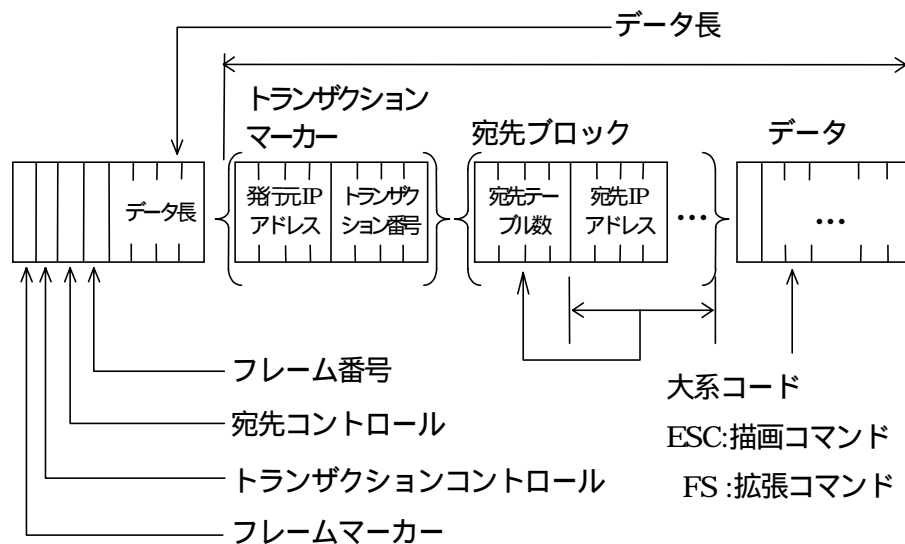
GPはコマンドデータ部を解析した結果、レスポンスデータ部、ACK、NAKまたは無応答となります。

GPからホストコンピュータへデータを送る場合



2-3-2 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様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バイトは全てのフレームで存在します。

よって、フレームチェックをする時は先頭の8バイトをチェックし、その後、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。

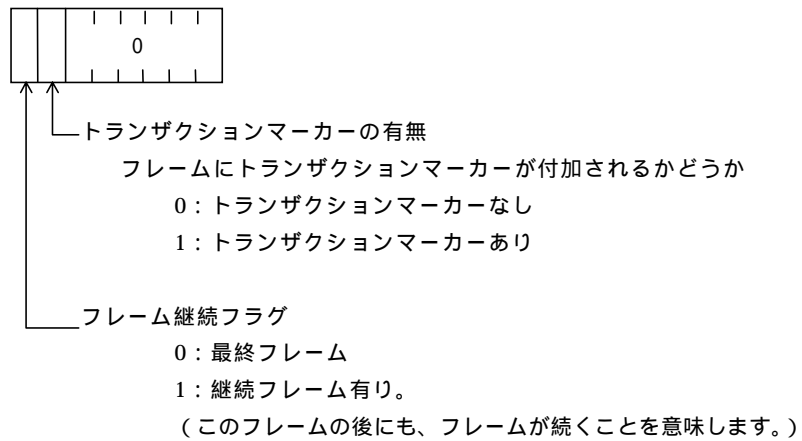
‘B’ :バイナリコマンドフレーム

‘b’ :バイナリレスポンスフレーム



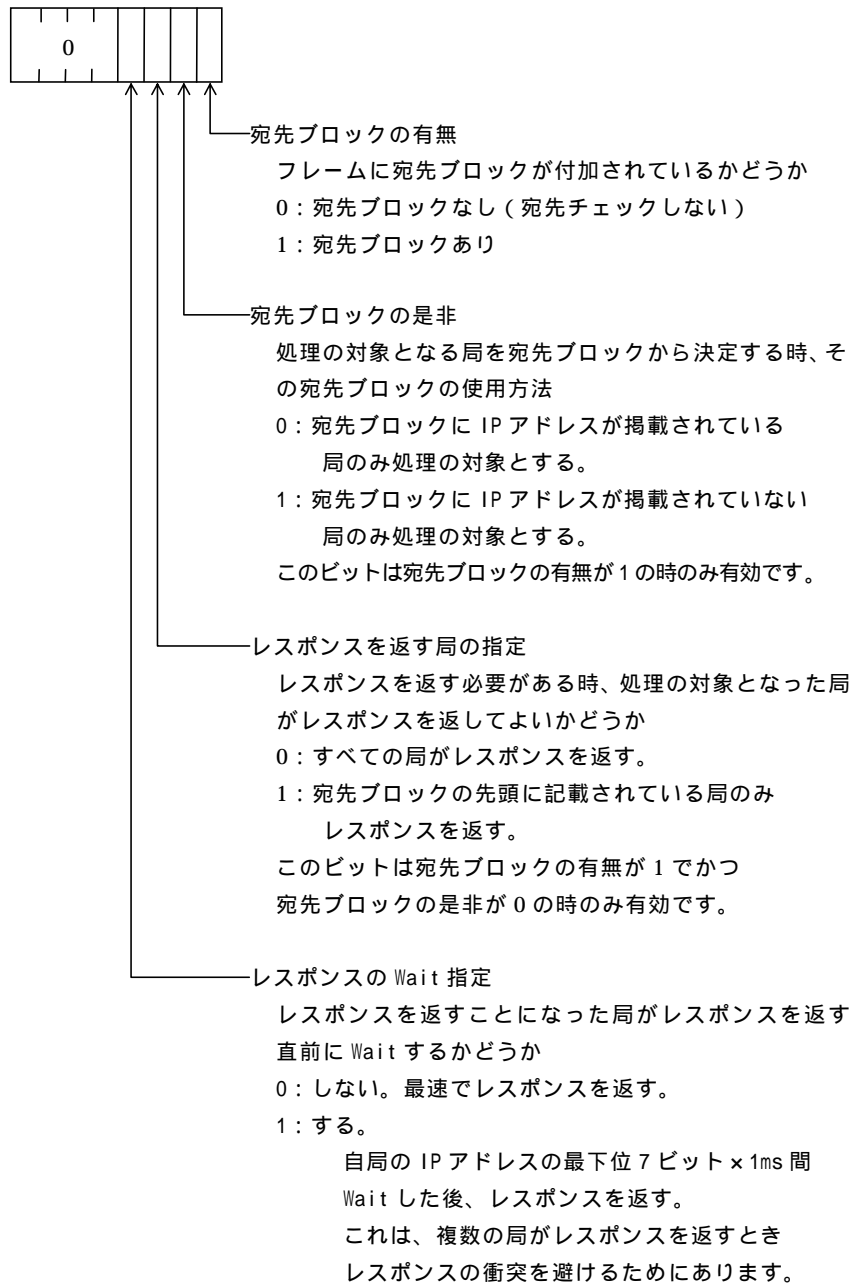
・バイナリフレームしかサポートしていません。

トランザクションコントロール(1バイト)



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。その時、分割フレームである事を示すビットで、先頭と途中のフレームには1をセットし、最終フレームには0をセットします。

宛先コントロール(1バイト)

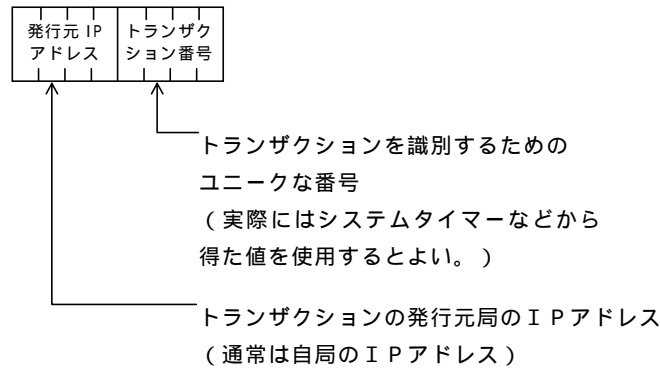


宛先コントロールの使用例

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時は 09h を指定します。

トランザクションマーカ



トランザクションマーカの使用用途

GPはトランザクションマーカ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返信します。)ここまではマーカ無しの場合と同じです。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。

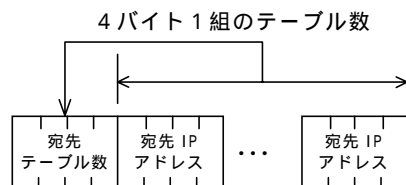
フレーム番号

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。



例 宛先 IP アドレスが 1.2.3.4 と 1.2.3.5 の 2 組の場合

0	0	0	2	1	2	3	4	1	2	3	5
---	---	---	---	---	---	---	---	---	---	---	---

第3章

画面の転送

ここではGP-PRO/PB for Windowsにて作画されたデータをイーサネット上のGPに送信、またはイーサネット上のGP内のデータをGP-PRO/PB for Windowsにて受信する方法を説明します。

3-1 画面転送の基本手順

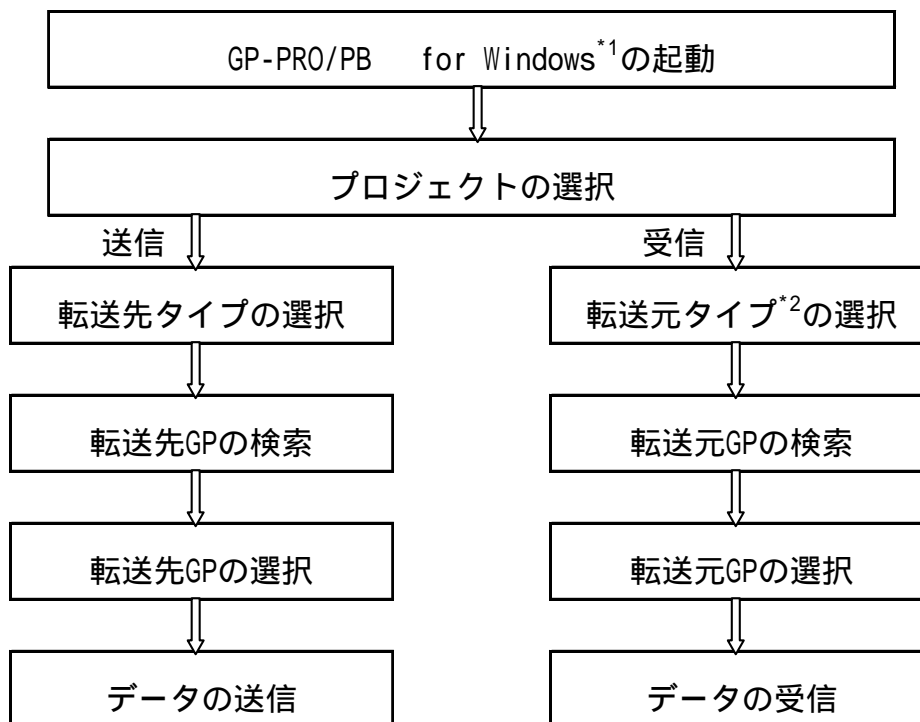
初めてイーサネットにて送信する場合は、先に各GPにイーサネット用のプロトコルを転送ケーブルにて、転送しておく必要があります。

GPの設定は [参照](#) 2-1 設定方法をご覧ください。

イーサネットの布設につきましては、[参照](#) GP70シリーズGPイーサネットI/Fユニットユーザーズマニュアル(ユニットに付属)をご覧ください。

3-1-1 転送の流れ

GP-PRO/PB for Windowsで作成したデータをイーサネット上のGPに転送する、またはGP内のデータを受信する基本操作の流れを示します。



*1 GP-PRO/PB for Windows Ver.2.0以降のバージョン

*2 実際の操作は[転送設定]の「転送先タイプ」を選択します。

画面の転送

[プロジェクト(P)]から[画面の転送(T)]を選択すると、転送のための画面が表示されます。内容についての概要を示します。

GP-PRO/PB から画面データを送信します。

画面の転送を終了します。

プロジェクト名を表示します。

GP-PRO/PB に画面データを受信します。

転送設定で現在選択されている転送の方法が表示されます。

転送設定で現在選択されているセットアップの方法が表示されます。イーサネットを選択している場合は無しになります。

転送設定で現在選択されている転送先タイプが表示されます。

転送を中断します。

転送設定ダイアログボックスが表示されます。

転送の経過を表示します。

転送の進行状況を表示します。

コメントを表示します。

転送準備の進行状況を表示します。

転送設定

[設定(S)]から[設定(S)]を選択すると、転送設定のためのダイアログボックスが表示されます。概要について説明します。

GP からデータを受信するためのアップロード情報を送信するかどうかを設定します。

転送先タイプがイーサネットの場合、シミュレーション機能は利用できません。

データを送信するとき、GP システムの設定の情報を転送するかどうかを設定します。





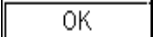
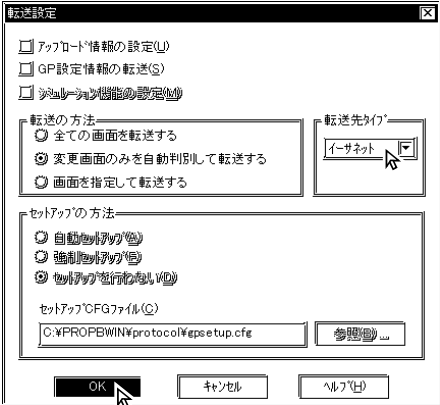


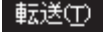

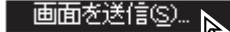
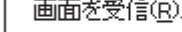

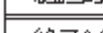
転送先タイプを選択します。

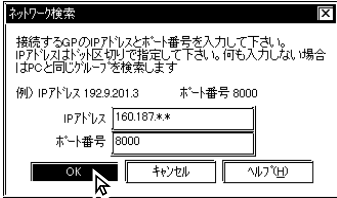
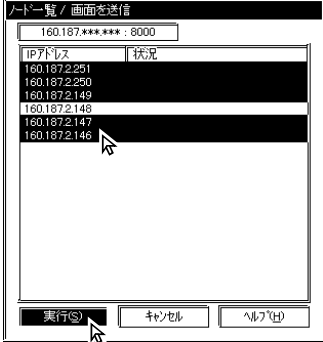
転送の方法を選択します。

セットアップの方法を選択します。転送先タイプがイーサネットの場合、セットアップは行えません。

3-1-2 画面転送の操作手順

イーサネット上の GP へ画面を送信します。GP からの受信についてはNOTE を参照ください。

OPERATION	NOTE
<p>GP-PRO/PB for Windows95 で転送するプロジェクトを選択します。</p>	<p>イーサネット上の GP からデータを受信する場合は、PLC タイプが「メモリリンクEthernetタイプ」のプロジェクトを選んでください。「転送先タイプ」にイーサネットが選択できるようになります。</p>
<p>[画面の転送] をオープンします。</p>	
<p>[設定(S)] から [設定(S)] を選択、または  をクリックします。</p>	
<p>  </p>	<p>「GP 設定情報の転送」を指定すると、ネットワーク上のすべての GP の IP アドレスが、[GP システムの設定] にて設定された IP アドレスに変更されます。「GP 設定情報の転送」を指定する場合は十分ご注意ください。</p>
<p>[転送設定] ダイアログボックスの転送先タイプで「イーサネット」を選択し、 で確定します。</p>	
	
<p>[画面の転送] で [転送(T)] から [画面を送信(S)] を選択、または  をクリックします。</p>	<p>GP からデータを受信する場合は、[転送(T)] から [画面を送信(S)] を選択、または  をクリックします。</p>
<p>     </p>	<p>GP からデータを受信するためには、GP にデータを送信する際に [アップロード情報] を送信しておく必要があります。</p>

OPERATION	NOTE
<p>[ネットワーク検索]ダイアログボックスで、転送先のGPのIPアドレスとポート番号を入力し、OKで検索を開始します。</p>  <p>検索結果が[ノード一覧 / 画面を送信]ダイアログボックスに表示されます。送信したいGPのIPアドレスを選択し、実行(S)で転送を開始します。</p> 	<p>2回目以降の転送の場合は、前回の設定値で自動的に検索します。</p> <p>「*」を用いて検索することもできます。 例) 192.9.*.*</p> <p>Ctrl キーを押しながら選択すると、複数選択することができます。 Shift キーを押しながら選択すると、連続選択することができます。</p> <p>GP からデータを受信する場合は、複数選択はできません。</p> <p>受信を開始すると、保存先を尋ねるダイアログボックスが表示されます。保存先フォルダとファイル名を指定してください。</p>

第4章

コマンドデータ

ここではホスト側からGP側へ送信するコマンドデータ部と、GP側からホストへ応答するレスポンスデータ部について説明しています。一例をあげてそれぞれのコマンドデータの内容を記述します。

4-1 表示用コマンドデータ

システムエリアへの書き込み、システムエリアからの読み出し、または図形入力などのコマンドを下記に示します。

コマンド一覧

コマンド	内 容
ESC W	システムエリア書き込み
ESC R	システムエリア読み出し
ESC T	文字列表示
ESC L	直線表示
ESC B	四角形表示
ESC S	塗込み四角形
ESC C	円表示
ESC A	円弧表示
ESC G	扇形表示
ESC P	塗込み
ESC I	割り込み出力問い合わせ
ESC I	タッチ入力

拡張コマンド一覧

コマンド	内 容	拡張内容
ESC t	文字列表示 拡張機能	回転、方向、強調
ESC l	直線表示 拡張機能	矢印
ESC b	四角形表示 拡張機能	面取り
ESC s	塗込み四角形表示 拡張機能	面取り
ESC c	塗込み円表示 拡張機能	タイリングパターン
ESC g	扇形表示 拡張機能	線種
ESC #	輝度/コントラスト調整	
ESC \$	輝度/コントラスト現在値	



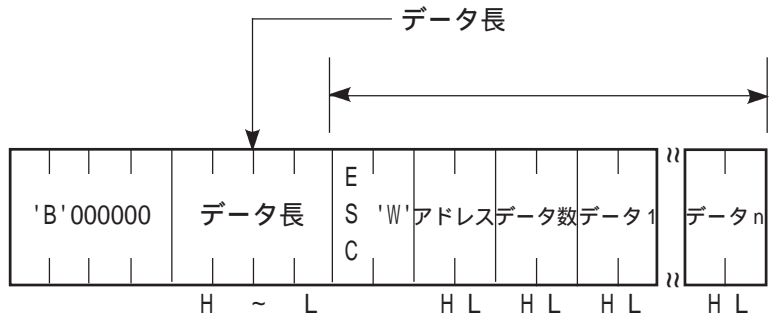
- ・ コマンド内で使用するコントロールコード
参照 2-3 通信プロトコルの制御手順
- ・ 描画関連のコマンドの最大 X、Y 座標は使用機種により異なります。

4-2 システムエリアへの書き込み <ESC W>

ホストはこの書き込みコマンドによってシステムエリアの任意のアドレスにデータを書き込みます。

書き込みコマンドの内容は次のとおりです。

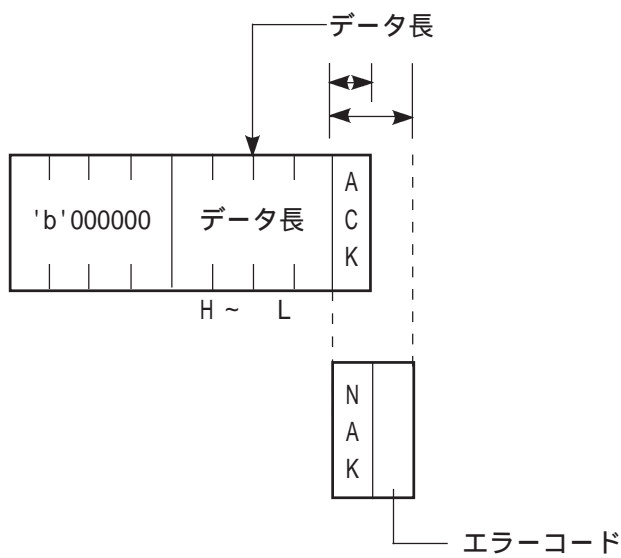
ホスト コマンドデータ部



<設定範囲>

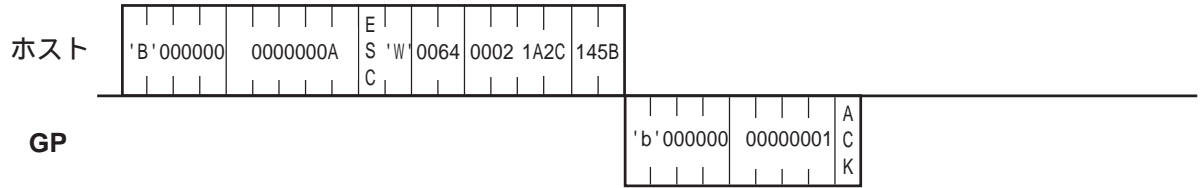
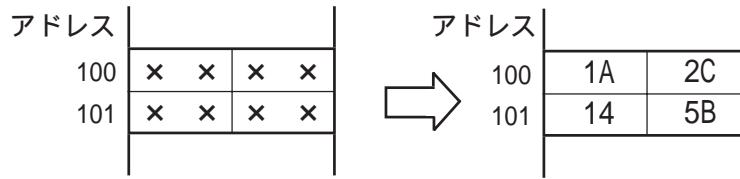
- ・アドレス : 0000h ~ 1FFFh (0 ~ 8191)
- ・データ数 : 0001h ~ 0040h (1 ~ 64)
- ・データ : 0000h ~ FFFFh

GP レスポンスデータ部
ACKまたはNAKで応答



<例題>

システムエリアのアドレス 100 に 16 進データ 1A2C と 145B を書き込みます。

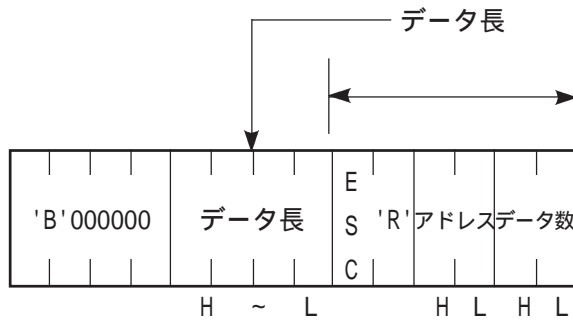


4-3 システムエリアからの読み出し <ESC R>

ホストはこの読み出しコマンドによってシステムエリアの任意のアドレスのデータを読み出します。

読み出しコマンドの内容は次のとおりです。

ホスト コマンドデータ部



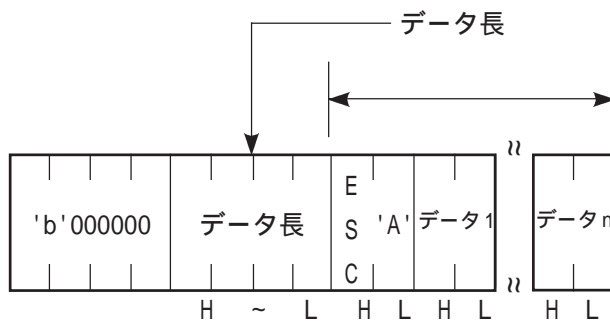
<設定範囲>

- ・アドレス : 0000h ~ 1FFFh (0 ~ 8191)
- ・データ数 : 0001h ~ 0040h (1 ~ 64)

GP レスポンスデータ部

正常時

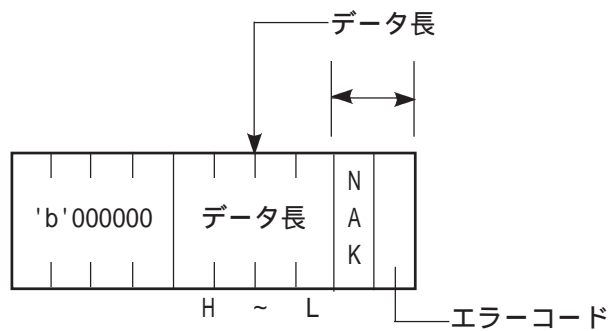
<設定範囲>



- ・データ : 0000h ~ FFFFh

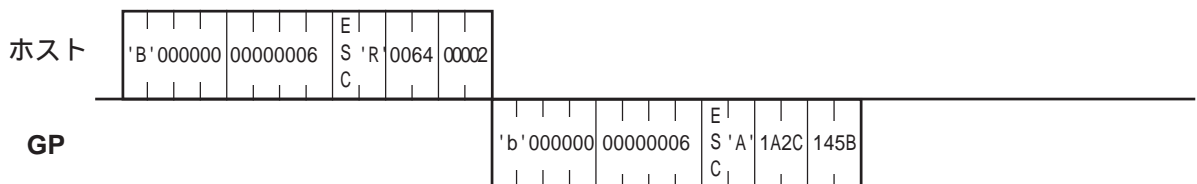
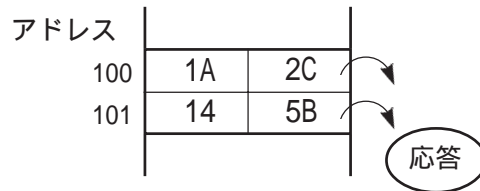
異常時

NAK で応答



<例題>

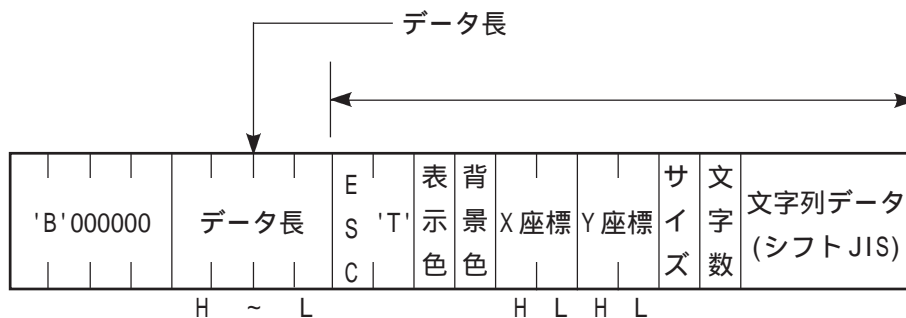
システムエリアのアドレス 100 から 16 進データ 2 ワードを読み出します。



4-4 文字列表示 <ESC T>

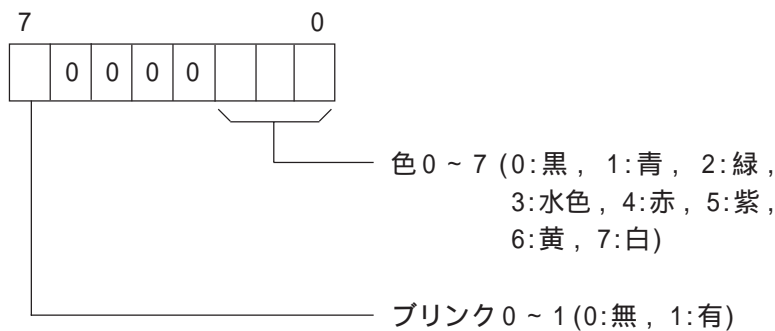
文字列を書くためのコマンドデータ部の内容は以下のとおりです。

ホスト コマンドデータ部



<設定範囲>

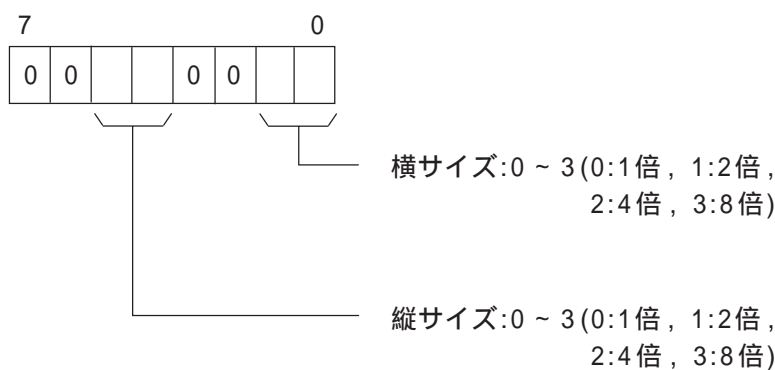
- ・表示色 / 背景色



・X座標: 0000h ~ 031Fh (0 ~ 799)

・Y座標: 0000h ~ 0257h (0 ~ 599)

- ・サイズ



・文字サイズ (バイト数): 01h ~ 50h (1 ~ 80)

・文字列データ: ANK文字は1バイト、漢字は2バイト

GP レスポンスデータ部
ACK または NAK で応答

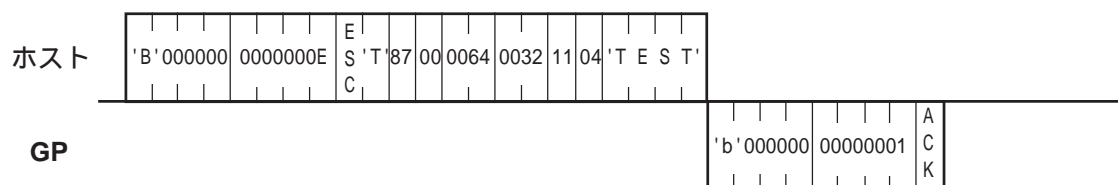
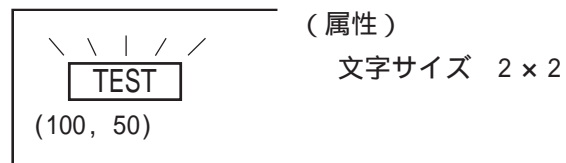
参照 4-2 システムエリアへの書き込み



- GP-675 シリーズの 64 色、3 速プリンクは使用できません。

< 例題 >

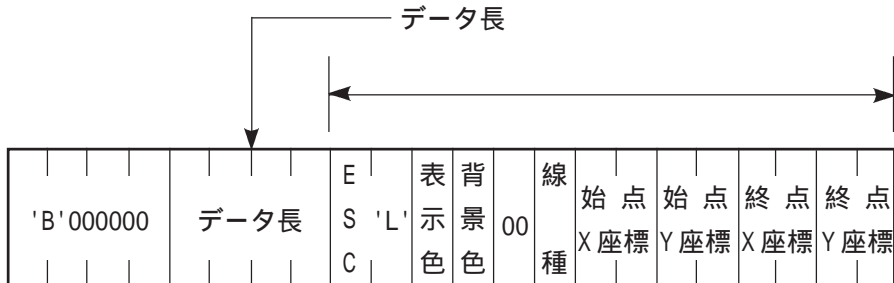
座標 (100, 50) に「TEST」とプリンク表示します。



4-5 直線表示 <ESC L>

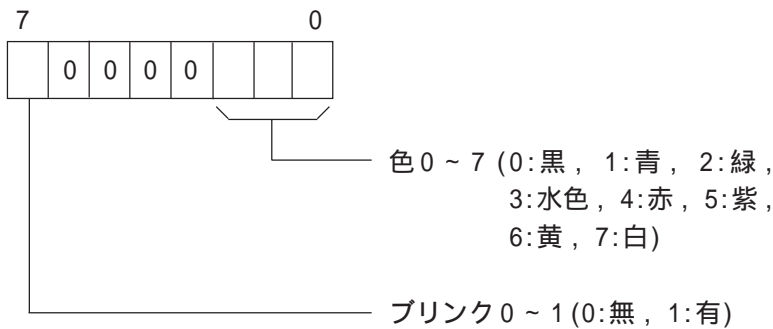
直線を書くためのコマンドデータ部の内容は以下のとおりです。

ホスト コマンドデータ部



<設定範囲>

- ・表示色 / 背景色

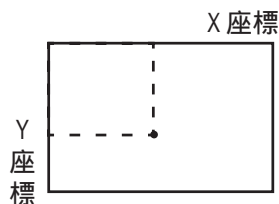


- ・線種： 00h ~ 07h (00 : ———、01 : - - - -、02 : — - - -、03 : — - - -、04 : ———、05 : - - - -、06 : — - - -、07 : — - - -)
- ・X座標： 0000h ~ 031Fh (0 ~ 799)
- ・Y座標： 0000h ~ 0257h (0 ~ 599)

GP レスポンスデータ部
ACKまたはNAKで応答
参照 4-2 システムエリアへの書き込み



- ・ GP-675シリーズの64色、3速ブリンクは使用できません。
- ・ 線種0 ~ 3は1ドット線、4 ~ 7は2ドット線です。
- ・ 点描画を行う場合は、始点X座標と終点X座標を同じ値に、始点Y座標と終点Y座標も同じ値にして、コマンド送信してください。



< 例題 >

座標 (100, 50)(400, 250) を結ぶ2ドット破線を描画します。

(100, 50)



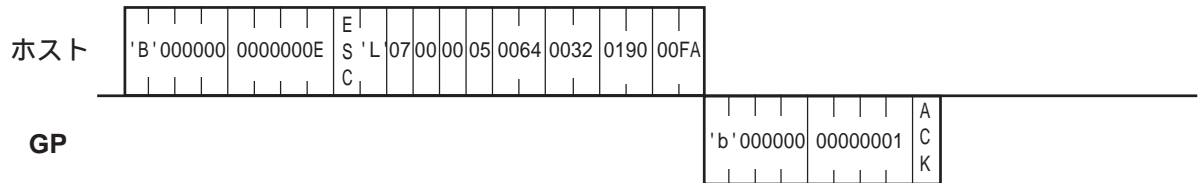
(400, 250)

(属性)

表示色 : 白

背景色 : 黒

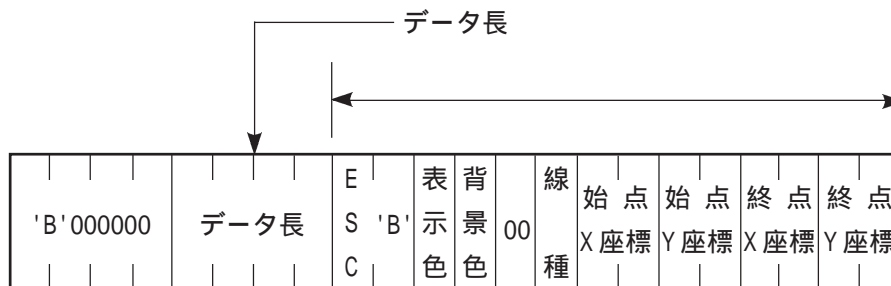
線種 : 2ドット破線



4-6 四角形表示 <ESC B>

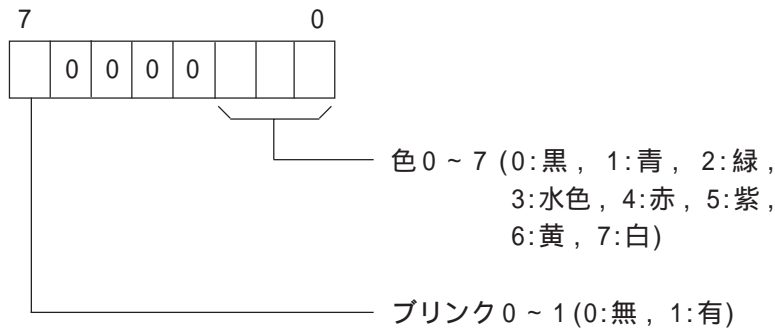
四角形を書くためのコマンドデータ部の内容は以下のとおりです。

ホスト コマンドデータ部



<設定範囲>

- ・表示色 / 背景色



- ・線種: 00h ~ 07h (00: ———、01: - - - -、02: — - - -、03: — - - -、04: ———、05: - - - -、06: — - - -、07: — - - -)
- ・X座標: 0000h ~ 031Fh (0 ~ 799)
- ・Y座標: 0000h ~ 0257h (0 ~ 599)

GP レスポンスデータ部
ACKまたはNAKで応答
参照 4-2 システムエリアへの書き込み

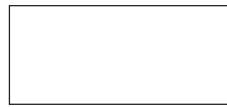


- ・線種 0 ~ 3 は 1 ドット線、4 ~ 7 は 2 ドット線です。
- ・GP-675 シリーズの 64 色、3 速ブリンクは使用できません。

< 例題 >

座標 (100, 50)(200, 100) に四角形を描画します。

(100, 50)



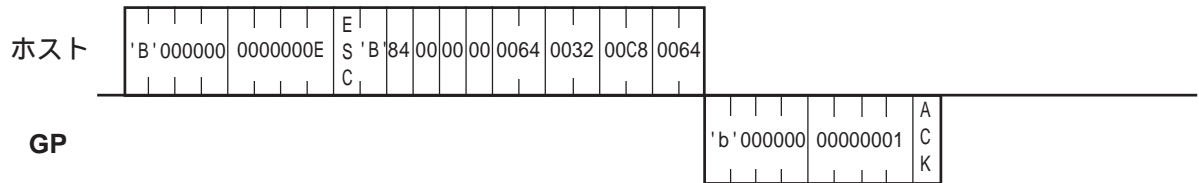
(200, 100)

(属性)

表示色 : 赤 プリンタ : 有

背景色 : 黒 プリンタ : 無

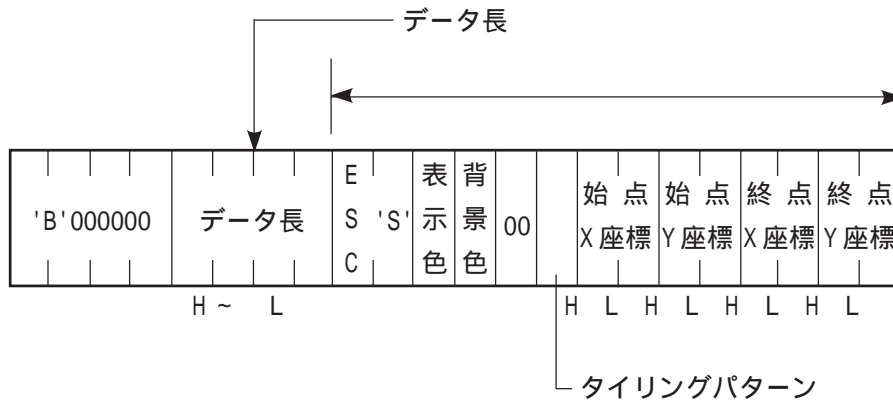
線種 : 1ドット実線



4-7 塗り四角形表示 <ESC S>

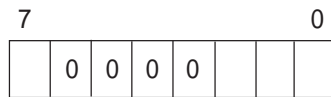
塗り四角形を描くためのコマンドデータ部の内容は以下のとおりです。

ホスト コマンドデータ部



<設定範囲>

- ・表示色 / 背景色



色0 ~ 7 (0:黒, 1:青, 2:緑, 3:水色, 4:赤, 5:紫, 6:黄, 7:白)

ブリンク0 ~ 1 (0:無, 1:有)

- ・タイリングパターン: 00h ~ 08h^{*1}
- ・X座標: 0000h ~ 031Fh (0 ~ 799)
- ・Y座標: 0000h ~ 0257h (0 ~ 599)

GP レスポンスデータ部
ACKまたはNAKで応答
参照 4-2 システムエリアへの書き込み

^{*1} タイリングパターンの種類

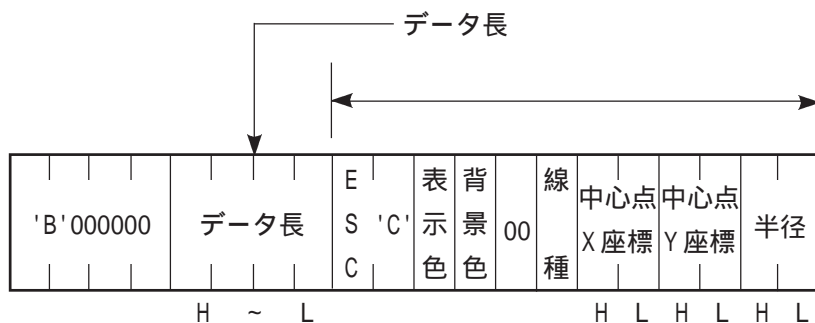
タイリングパターン番号	タイリングパターン	タイリングパターン番号	タイリングパターン	タイリングパターン番号	タイリングパターン
0		3		6	
1		4		7	
2		5		8	

8ドット

4-8 円表示 <ESC C>

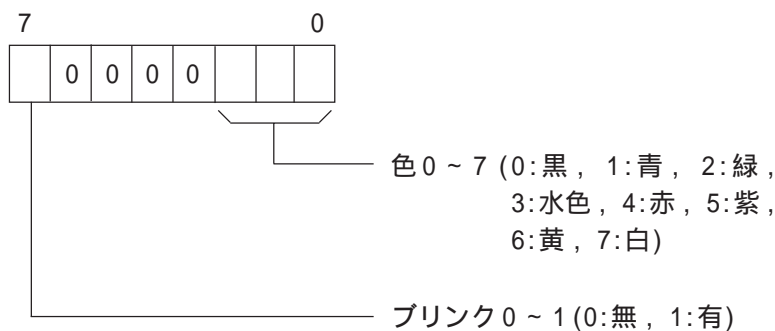
円を描くためのコマンドデータ部の内容は以下のとおりです。

ホスト コマンドデータ部



<設定範囲>

- ・表示色 / 背景色



- ・線種: 00h ~ 03h (00: ———、01: - - - -, 02: — - —、03: — - - -)
- ・X座標: 0000h ~ 031Fh (0 ~ 799)
- ・Y座標: 0000h ~ 0257h (0 ~ 599)
- ・半径: 0001h ~ 031Fh (1 ~ 799)

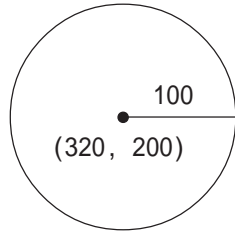
GP レスポンスデータ部
ACKまたはNAKで応答
参照 4-2 システムエリアへの書き込み



- ・ GP-675シリーズの64色、3速ブリンクは使用できません。

< 例題 >

座標 (320, 200) を中心として半径 100 の円を描画します。

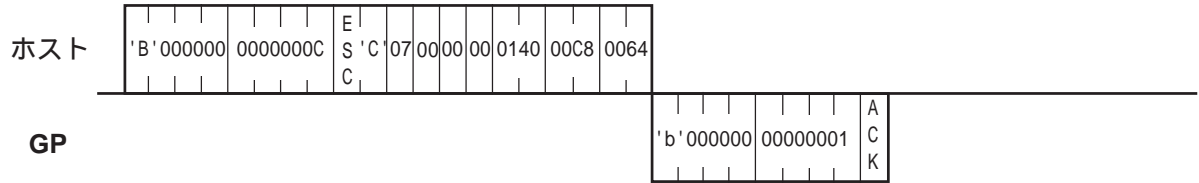


(属性)

表示色 : 白

背景色 : 黒

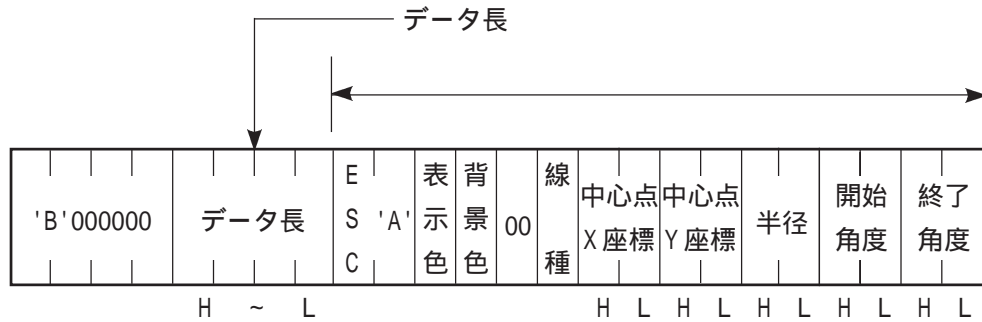
線種 : 1 ドット実線



4-9 円弧表示 <ESC A>

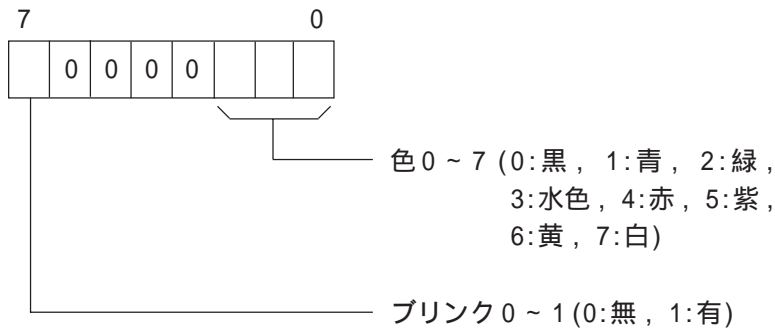
円弧を描くためのコマンドデータ部の内容は以下のとおりです。

ホスト コマンドデータ部



<設定範囲>

- ・表示色 / 背景色



- ・線種： 00h ~ 03h (00 : ———、01 : - - - -、02 : — - -、03 : — - - -)
- ・X座標： 0000h ~ 031Fh (0 ~ 799)
- ・Y座標： 0000h ~ 0257h (0 ~ 599)
- ・半径： 0001h ~ 031Fh (1 ~ 799)
- ・角度： 0000h ~ 0168h (0 ~ 360)

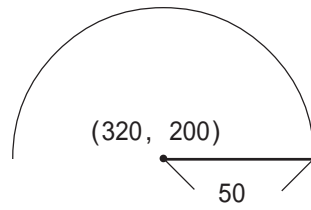
GP レスポンスデータ部
ACKまたはNAKで応答
参照 4-2 システムエリアへの書き込み



- ・ GP-675シリーズの64色、3速ブリンクは使用できません。
- ・ 描画方向は反時計回りです。
- ・ 開始角度と終了角度を同じにしないでください。

< 例題 >

座標 (320, 200) を中心として半径 50 の半円 (円弧) を描画します。



(属性)

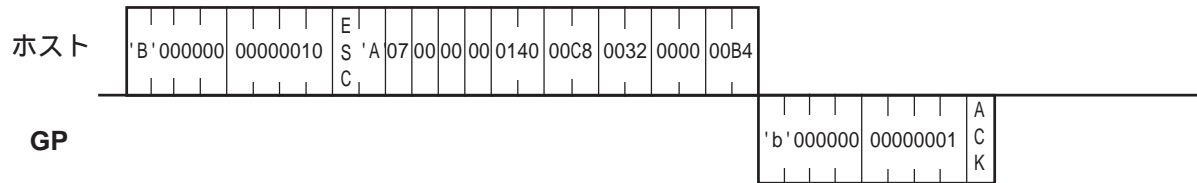
表示色 : 白

背景色 : 黒

線種 : 1 ドット実線

開始角度 : 0°

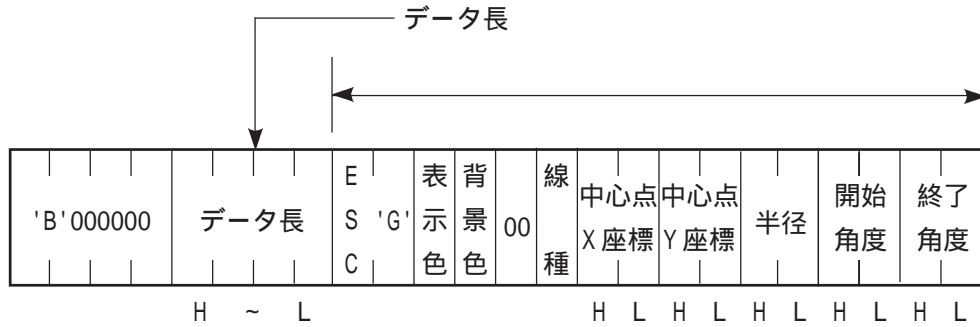
終了角度 : 180°



4-10 扇形表示 <ESC G>

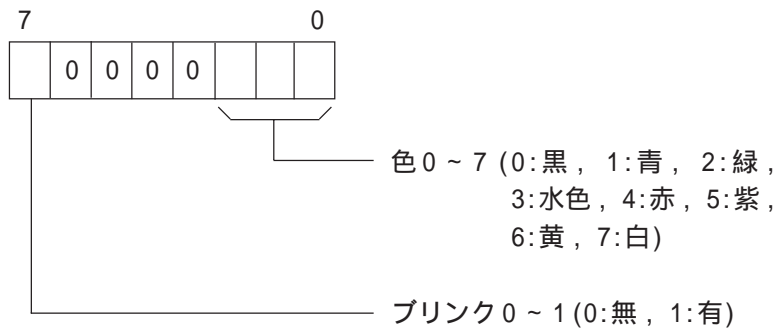
扇形を描くためのコマンドデータ部の内容は以下のとおりです。

ホスト コマンドデータ部



<設定範囲>

・表示色 / 背景色



- ・線種: 00h ~ 03h (00: ———、01: - - - -、02: — — —、03: — — — —)
- ・X座標: 0000h ~ 031Fh (0 ~ 799)
- ・Y座標: 0000h ~ 0257h (0 ~ 599)
- ・半径: 0001h ~ 031Fh (1 ~ 799)
- ・開始角度: 0000h ~ 0168h (0 ~ 360)
- ・終了角度: 0000h ~ 0168h (0 ~ 360)

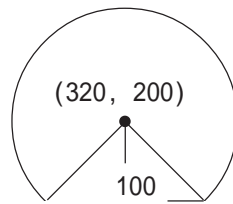
GP レスポンスデータ部
ACKまたはNAKで応答
参照 4-2 システムエリアへの書き込み



- ・ GP-675シリーズの64色、3速ブリンクは使用できません。
- ・ 描画方向は反時計回りです。
- ・ 開始角度と終了角度を同じにしないでください。

< 例題 >

座標 (320, 200) を中心として半径 100 の扇形を描画します。

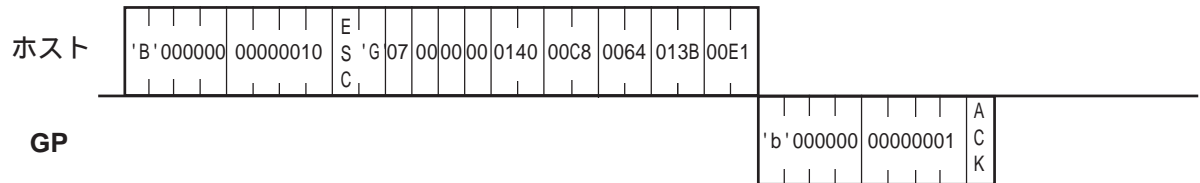


(属性)

表示色 : 白 開始角度 : 315 °

背景色 : 黒 終了角度 : 225 °

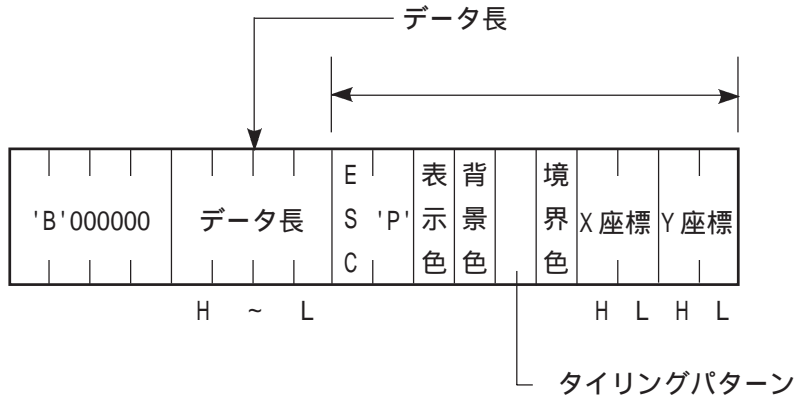
線種 : 1 ドット実線



4-11 塗り込み表示 <ESC P>

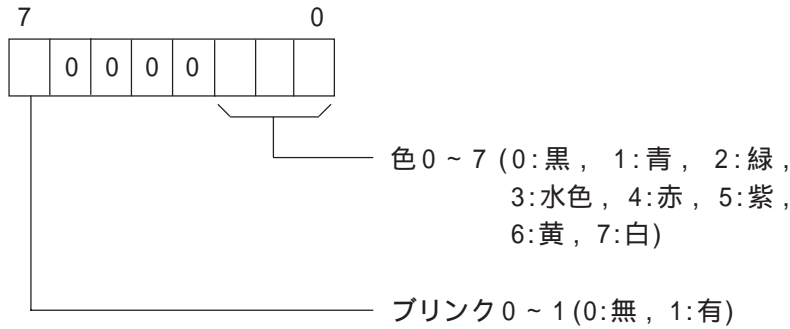
図形の塗り込みを行うためのコマンドデータ部の内容は以下のとおりです。

ホスト コマンドデータ部



<設定範囲>

- ・表示色 / 背景色 / 境界色



- ・タイリングパターン： 00h ~ 08h
参照 4-7 塗り込み四角形表示 「タイリングパターンの種類」
- ・X座標： 0000h ~ 031Fh (0 ~ 799)
- ・Y座標： 0000h ~ 0257h (0 ~ 599)

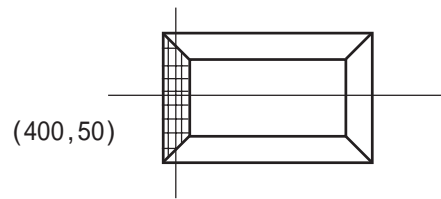
GP レスponseデータ部
ACKまたはNAKで応答
参照 4-2 システムエリアへの書き込み



- ・ GP-675シリーズの64色、3速ブリンクは使用できません。
- ・ 境界色のブリンクは「無」にしてください。

< 例題 >

座標 (400, 50) を囲む図形を塗り込みます。



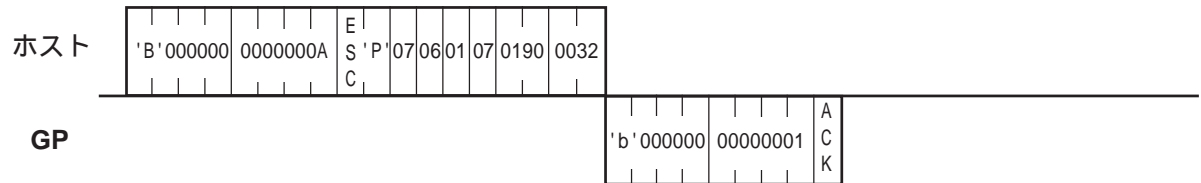
(属性)

表示色 : 白

背景色 : 黄

境界色 : 白

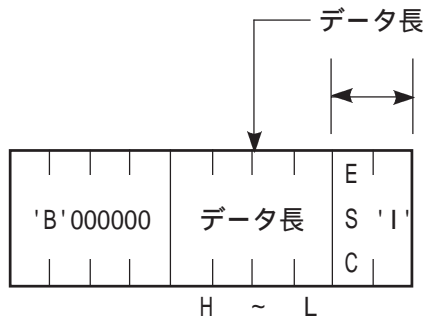
タイリングパターン : 1



4-12 割り込み出力の問い合わせ <ESC I>

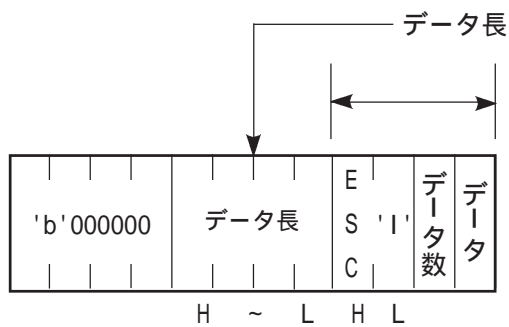
タッチパネル入力があったかどうかGPに問い合わせるコマンドの内容は以下のとおりです。

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

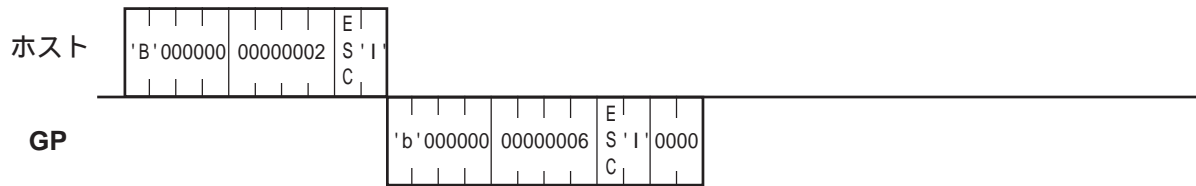
参照 4-3 システムエリアからの読み出し

<設定内容>

- データ数：今回送信するデータと前回までの割り込み出力したデータのことをいいます。割り込み出力のデータがないときには「00h」となります。
- 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ときには「00h」となります。

< 例題 >

タッチパネル入力があったかどうか GP に問い合わせます。

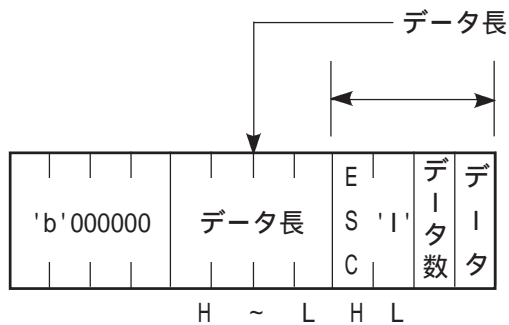


4-13 タッチ入力 <ESC I>

タッチ入力によって、GP側からホスト側へデータを送るためのコマンドの内容は以下のとおりです。

ホスト なし

GP レスponseデータ部



<設定内容>

- ・ データ数： 01h
- ・ データ： 00h ~ FFh

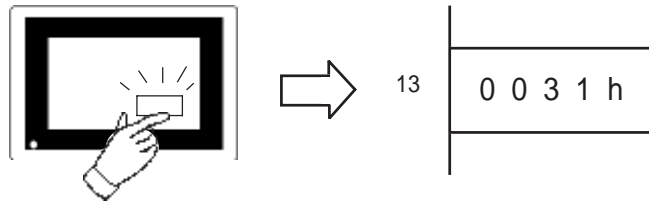
GPのTタグ、またはシステムエリアへの絶対値書き込みなどでアドレス13にデータを書くと、下位8ビットの内容が割り込みコードとして出力されます。(13h、11h、FFhは出力しません。「00」が出力されます。)

GPがアドレス13への書き込みが発生したときにレスponseデータを自動的に出力します。

本アクションはTCPコネクションのみです。UDPコネクションでは割り込み出力の問い合わせを行ってください。

< 例題 >

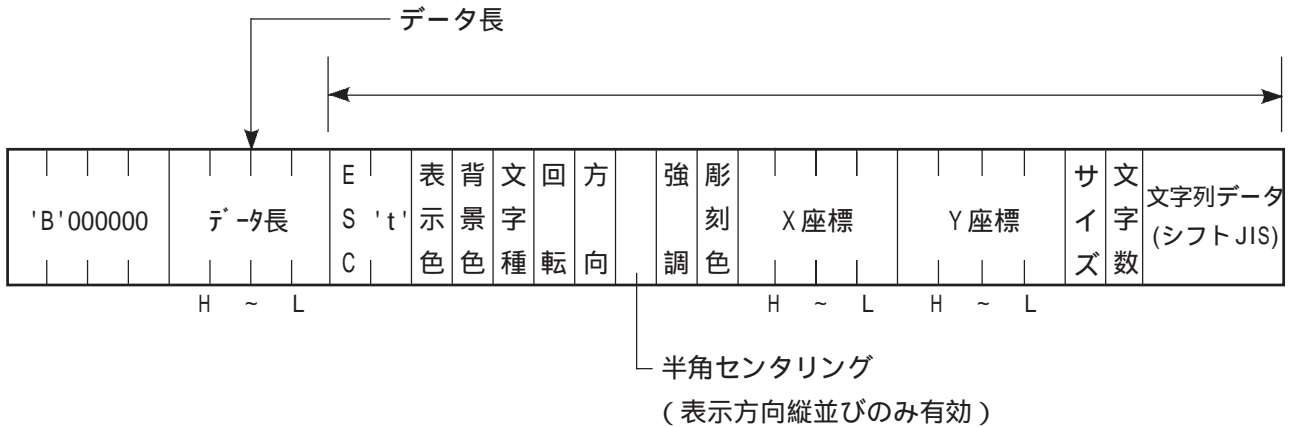
T タグによりシステムエリア 13 に 0031h を書き込みます。



4-14 文字列表示 拡張機能 <ESC t>

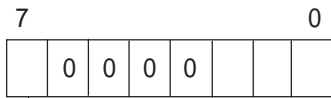
拡張機能を用いて文字列を書くためのコマンドデータ部の内容は以下のとおりです。
 拡張機能の内容は回転、方向、強調の追加です。

ホスト コマンドデータ部



<設定範囲>

・表示色 / 背景色 / 彫刻色



色 0 ~ 7 (0:黒, 1:青, 2:緑, 3:水色, 4:赤, 5:紫, 6:黄, 7:白)

ブリンク 0 ~ 1 (0:無, 1:有)

彫刻色はブリンク固定です。

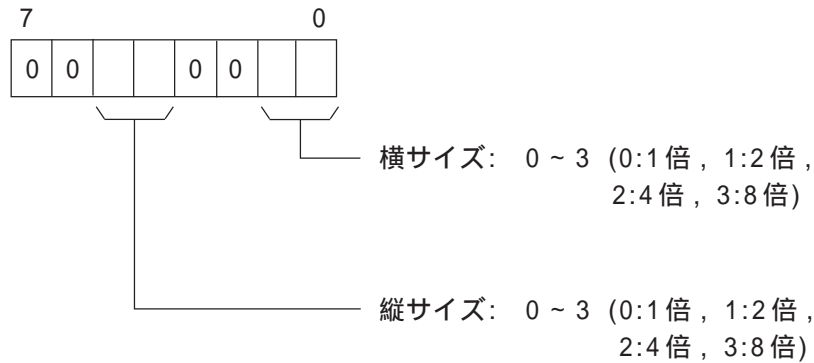
・文字種 :



文字種 0 ~ 9

- 0: 1/4角 (シフトJISコード)
- 1: 半角
- 2: 全角 (シフトJISコード) 半角表示
- 3: 5 × 7フォント
- 4: 7 × 9フォント
- 5: 11 × 16フォント
- 6: 24 × 32フォント
- 7: 7 × 9Fフォント
- 8: 11 × 16Fフォント
- 9: 1/2角 (シフトJISコード)

- ・回転： 00h ~ 03h (0:0°、1:90°、2:180°、3:270°)
- ・方向： 00h ~ 01h (0:横並び、1:縦並び)
- ・半角センタリング： 00h ~ 01h (0:なし、1:あり)
- ・強調： 00h ~ 02h (0:ノーマル、1:強調文字、2:彫刻文字)
- ・彫刻： 00h ~ 07h (0:黒、1:青、2:緑、3:水色、4:赤、5:紫、6:黄、7:白)
- ・X座標： 0000h ~ 031Fh (0 ~ 799)
- ・Y座標： 0000h ~ 0257h (0 ~ 599)
- ・サイズ

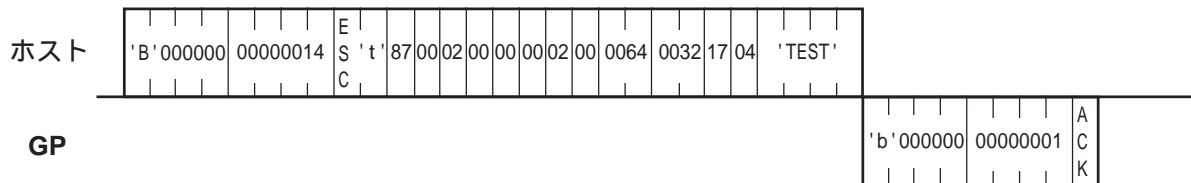
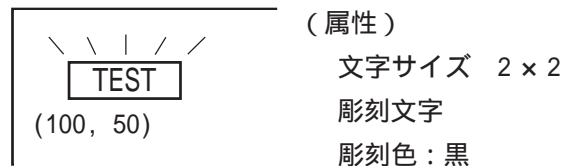


- ・文字サイズ(バイト数): 01h ~ 50h (1 ~ 80)
- ・文字列データ: ANK文字は1バイト、漢字は2バイト

GP レスポンスデータ部
 ACKまたはNAKで応答
 参照 4-2 システムエリアへの書き込み

< 例題 >

座標(100, 50)に全角で「TEST」とブリンク表示します。

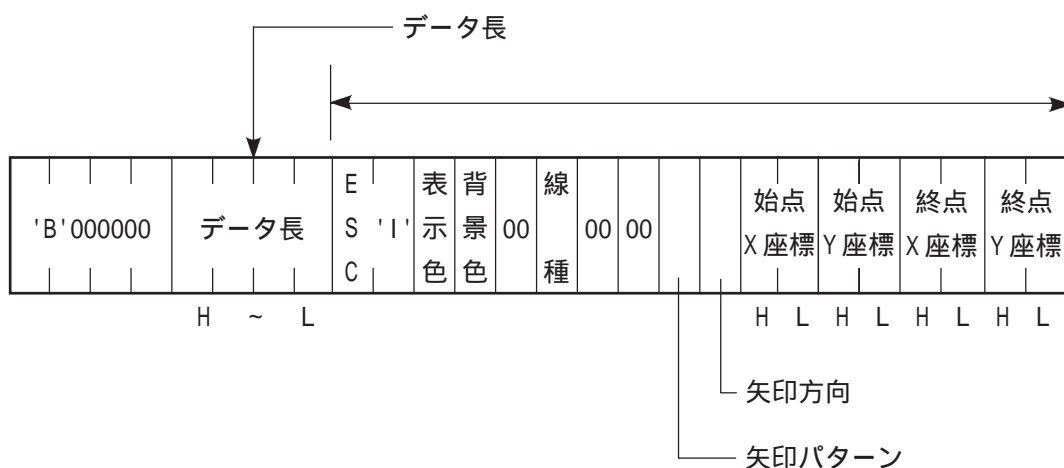


・ GP-675シリーズの64色、3速ブリンクは使用できません。

4-15 直線表示 拡張機能 <ESC I (スモールL)>

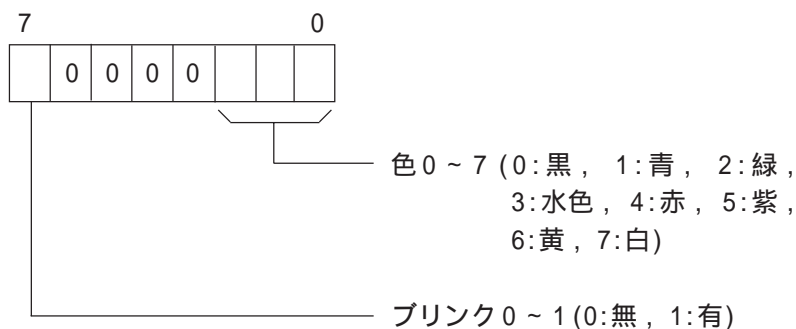
拡張機能を用いて直線を書くためのコマンドデータ部の内容は以下のとおりです。
拡張機能の内容は矢印の追加です。

ホスト コマンドデータ部



<設定範囲>

・表示色 / 背景色



- ・線種： 00h ~ 07h (00 : ———、01 : - - - -、02 : — - -、03 : — - - -、04 : ———、05 : - - - -、06 : — - -、07 : — - - -)
- ・矢印パターン： 00 ~ 03h (00:無、01:矢印、02:リザーブ、03:リザーブ)
- ・矢印方向： 00h ~ 01h (0 : 両端、1 : 終点)
- ・X座標： 0000h ~ 031Fh (0 ~ 799)
- ・Y座標： 0000h ~ 0257h (0 ~ 599)

GP レスポンスデータ部
ACKまたはNAKで応答
参照 4-2 システムエリアへの書き込み

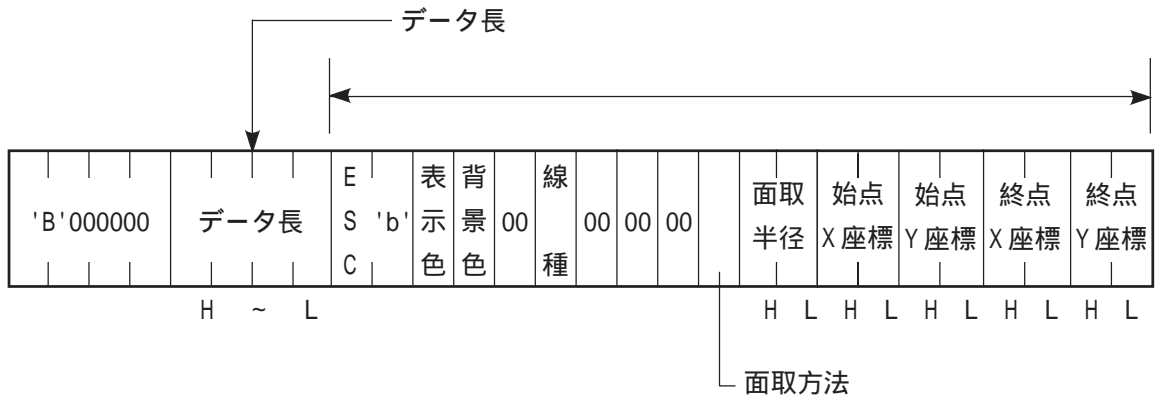


- ・ GP-675シリーズの64色、3速ブリンクは使用できません。
- ・ 線種0 ~ 3は1ドット線、4 ~ 7は2ドット線です。

4-16 四角形表示 拡張機能 <ESC b>

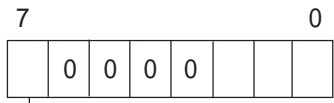
拡張機能を用いて四角形を書くためのコマンドデータ部の内容は以下のとおりです。
 拡張機能の内容は面取りの追加です。

ホスト コマンドデータ部



<設定範囲>

- ・表示色 / 背景色



色 0 ~ 7 (0:黒, 1:青, 2:緑, 3:水色, 4:赤, 5:紫, 6:黄, 7:白)

ブリンク 0 ~ 1 (0:無, 1:有)

- ・面取方法 : 00h ~ 02h (00 : なし、01 : 曲線、02 : 直線)
- ・面取半径 : 00h ~ 20h (0 ~ 32)
- ・線種 : 00h ~ 03h、08h、09h (00 : ———、01 : - - - -、02 : — - - -、03 : — - - -、08 : ———、09 : ———)
- ・X座標 : 0000h ~ 031Fh (0 ~ 799)
- ・Y座標 : 0000h ~ 0257h (0 ~ 599)

異常時

ACK または NAK で応答

参照 4-2 システムエリアへの書き込み

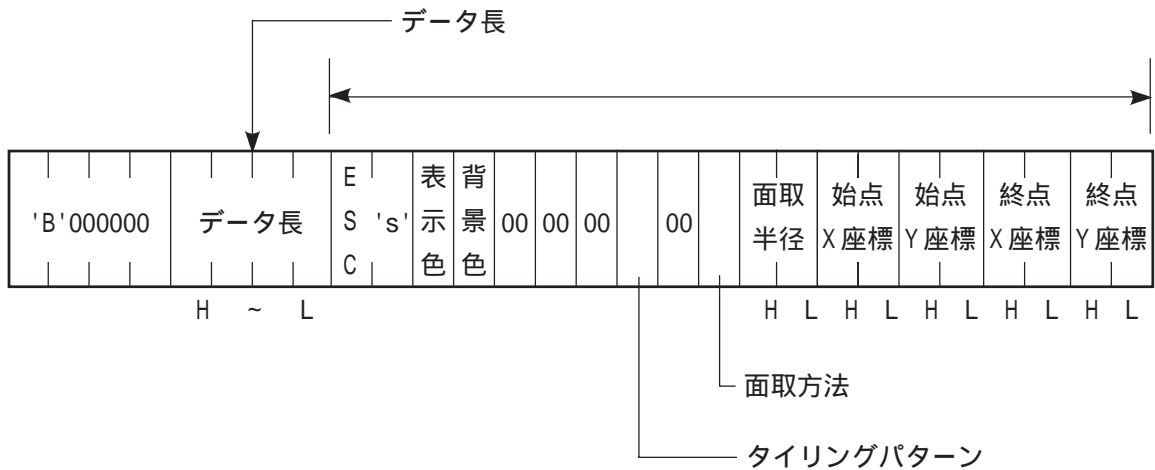


- ・ GP-675 シリーズの 64 色、3 速ブリンクは使用できません。
- ・ 線種は 0 ~ 3 が 1 ドット線、8 は 3 ドット線、9 は 5 ドット線です。

4-17 塗り四角形表示 拡張機能 <ESC s>

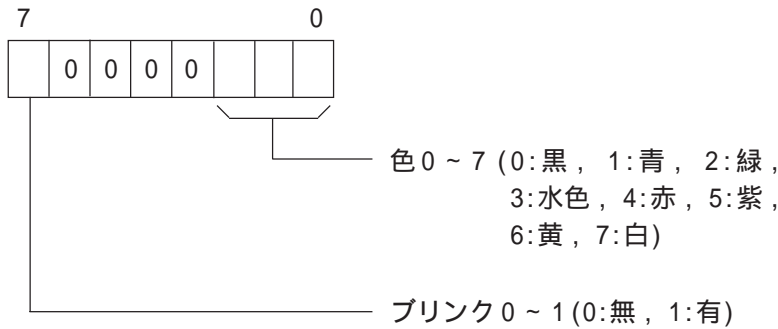
拡張機能を用いて塗り四角形を書くためのコマンドデータ部の内容は以下のとおりです。
 拡張機能の内容は面取りの追加です。

ホスト コマンドデータ部



<設定範囲>

- ・表示色 / 背景色



- ・面取方法 : 00h ~ 02h (0 : なし、1 : 曲線、2 : 直線)
- ・面取半径 : 00h ~ 20h (0 ~ 32)
- ・タイリングパターン : 00h ~ 08h
 参照 4-7 塗り四角形表示 「タイリングパターンの種類」
- ・X座標 : 0000h ~ 031Fh (0 ~ 799)
- ・Y座標 : 0000h ~ 0257h (0 ~ 599)

GP レスポンスデータ部
 ACKまたはNAKで応答
 参照 4-2 システムエリアへの書き込み

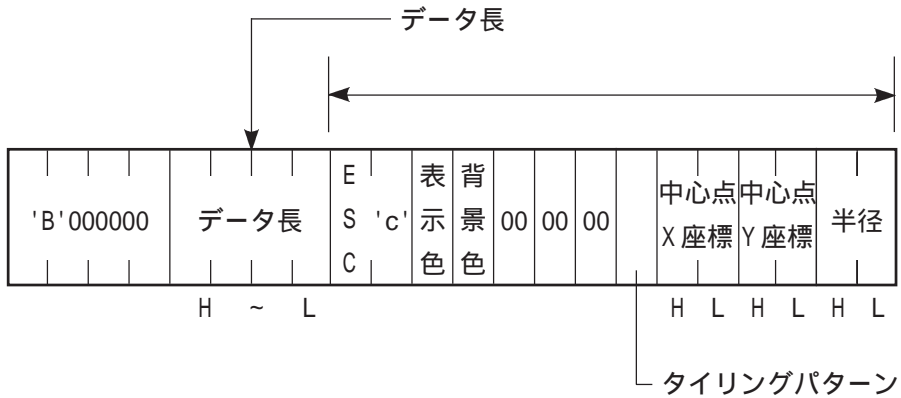


・ GP-675シリーズの64色、3速ブリンクは使用できません。

4-18 塗込み円表示 拡張機能 <ESC c>

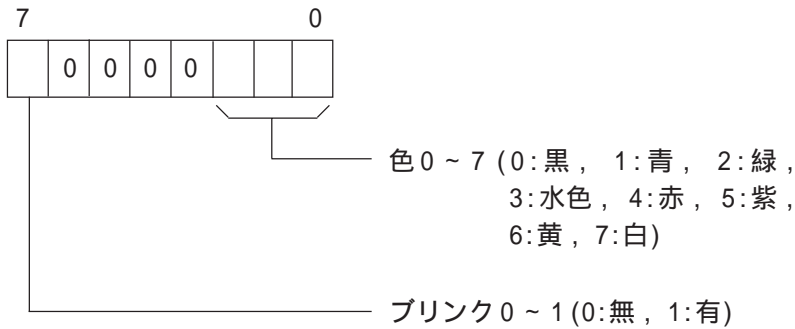
拡張機能を用いて塗込み円を書くためのコマンドデータ部の内容は以下のとおりです。
 拡張機能の内容はタイリングパターンの追加です。

ホスト コマンドデータ部



<設定範囲>

・表示色 / 背景色



・タイリングパターン： 00h ~ 08h

参照 4-7 塗込み四角形表示「タイリングパターンの種類」

・X座標： 0000h ~ 031Fh (0 ~ 799)

・Y座標： 0000h ~ 0257h (0 ~ 599)

・半径： 0001h ~ 031Fh (1 ~ 799)

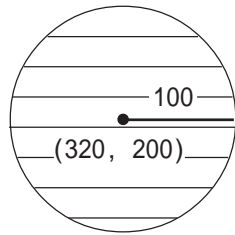
GP レスポンスデータ部
 ACKまたはNAKで応答
 参照 4-2「システムエリアへの書き込み」



・ GP-675シリーズの64色、3速ブリンクは使用できません。

< 例題 >

座標 (320, 200) を中心として半径 100 円をタイリングパターン 4 で描画します。

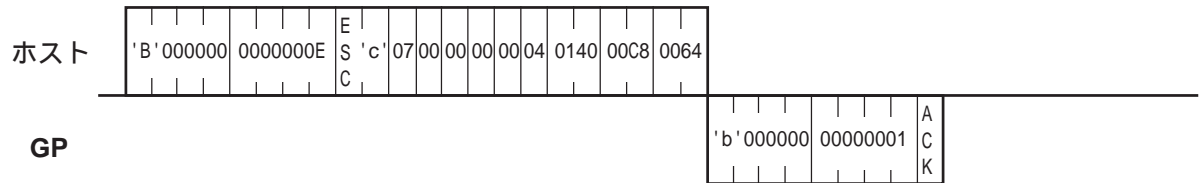


(属性)

表示色 : 白

背景色 : 黒

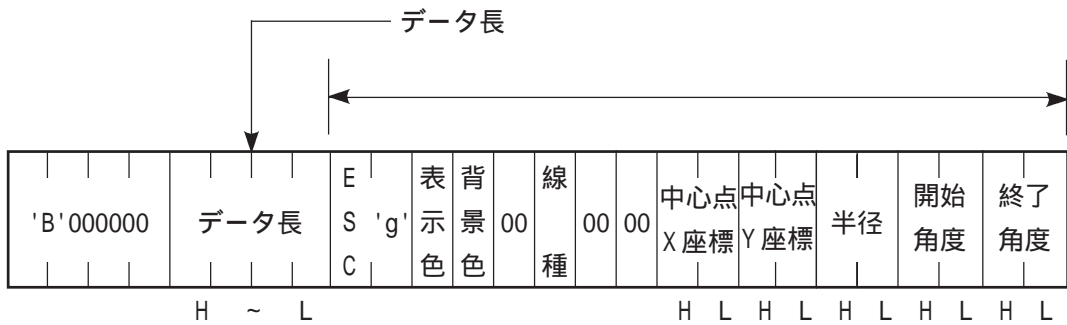
タイリングパターン : 4



4-19 扇形表示 拡張機能 <ESC g>

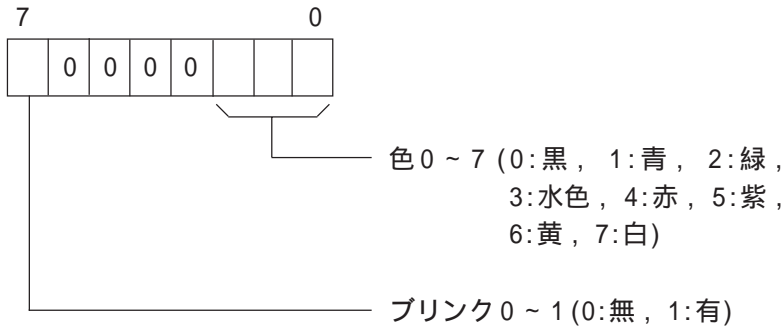
拡張機能を用いて扇形を書くためのコマンドデータ部の内容は以下のとおりです。
 拡張機能の内容は線種の追加です。

ホスト コマンドデータ部



<設定範囲>

・表示色 / 背景色



・線種: 00h ~ 03h, 08h, 09h (00: ———、01: - - - -、02: - - - -、03: - - - -、08: ———、09: ———)

- ・X座標: 0000h ~ 031Fh (0 ~ 799)
- ・Y座標: 0000h ~ 0257h (0 ~ 599)
- ・半径: 0001h ~ 031Fh (1 ~ 799)
- ・角度: 0000h ~ 0168h (0 ~ 360)

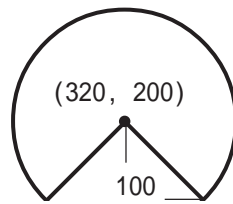
GP レスポンスデータ部
 ACKまたはNAKで応答
 参照 4-2 システムエリアへの書き込み



- ・ GP-675シリーズの64色、3速ブリンクは使用できません。
- ・ 線種は0 ~ 3が1ドット線、8は3ドット線、9は5ドット線です。
- ・ 描画方向は反時計回りです。
- ・ 開始角度と終了角度を同じにしないでください。

< 例題 >

座標 (320, 200) を中心として半径 100 の 3 ドット実線の扇形を描画します。

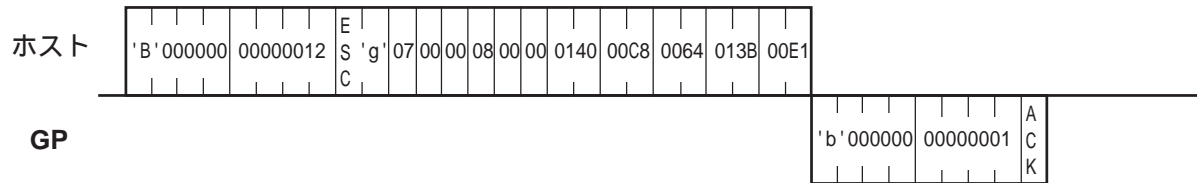


(属性)

表示色 : 白 開始角度 : 315 °

背景色 : 黒 終了角度 : 225 °

線種 : 3 ドット実線

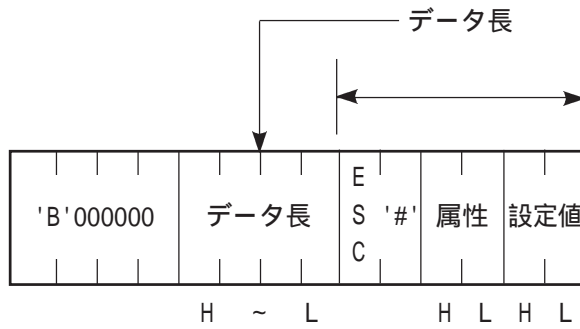


4-20 輝度・コントラスト調整 <ESC #>

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通りです。

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があります。(4-40ページの「輝度・コントラスト一覧」を参照してください。)

ホスト コマンドデータ部



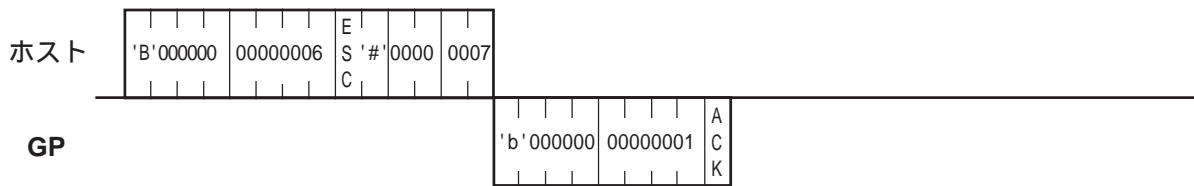
< 設定範囲 >

- ・ 属性： 00h ~ 01h (0 : コントラスト調整、1 : 輝度調整)
- ・ 設定値： 4-40 ページの「輝度・コントラスト一覧」を参照してください。

GP レスponseデータ部
ACKで応答
参照 4-2 システムエリアへの書き込み

< 例題 >

コントラスト調整の設定値7をセットします。

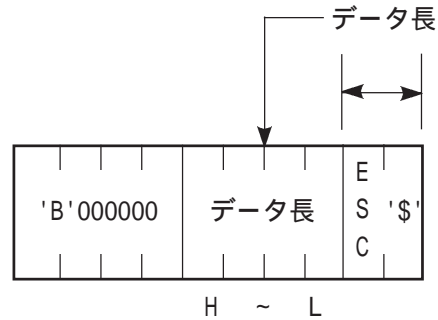


4-21 輝度・コントラスト現在値 <ESC \$>

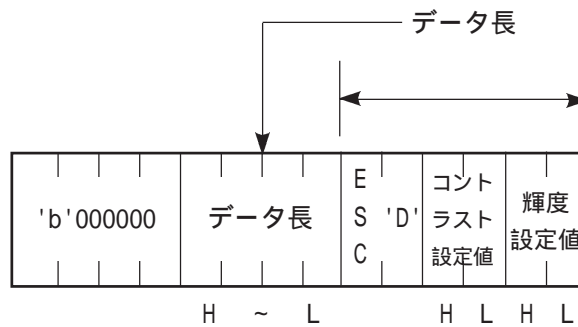
輝度・コントラストの現在値をコマンドを用いて取得するためのコマンドデータ部の内容は以下の通りです。

GPの機種によって輝度・コントラストがないものがあります。(4-40ページの「輝度・コントラスト一覧」を参照してください。)

ホスト コマンドデータ部



GP レスポンスデータ部



< 設定範囲 >

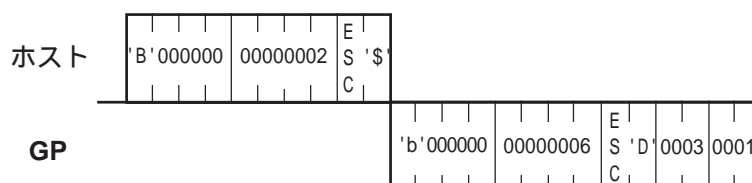
- ・ 設定値： 4-40ページの「輝度・コントラスト一覧」を参照してください。



- ・ コントラスト、輝度が設定できない機種では、データはFFFFとなります。

< 例題 >

コントラストの設定値3・輝度設定値1を取得します。



輝度・コントラスト一覧

GP機種	輝度設定 設定範囲	コントラスト設定 設定範囲
GP470E	0(暗)～1(明)	
GP570S		0(明)～7(暗)
GP570T		
GP570VM		
GP571T		
GP57JS		0(明)～7(暗)
GP675S		0(明)～7(暗)
GP675T		
GP870VM		

 設定不可

GP/GLC機種	輝度設定 設定範囲	コントラスト設定 設定範囲
GP477RE	0(暗)～1(明)	
GP577RS	0(明)～3(暗)	0(明)～7(暗)
GP577RT	0(明)～3(暗)	
GP377RT	0(明)～3(暗)	
GP2600T	0(明)～3(暗)	
GP2601T	0(明)～3(暗)	
GP2500T	0(明)～3(暗)	
GP2501T	0(明)～3(暗)	
GP2501S	0(明)～3(暗)	0(明)～7(暗)
GP2400T	0(明)～3(暗)	
GP2300T	0(明)～3(暗)	
GP2300L	0(明)～3(暗)	0(明)～7(暗)
GLC2600T	0(明)～3(暗)	
GLC2500T	0(明)～3(暗)	
GLC2400T	0(明)～3(暗)	
GLC2300T	0(明)～3(暗)	
GLC2300L	0(明)～3(暗)	0(明)～7(暗)

第5章

メモリリンク API

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 Windows 32 ビット API です。

5-1

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

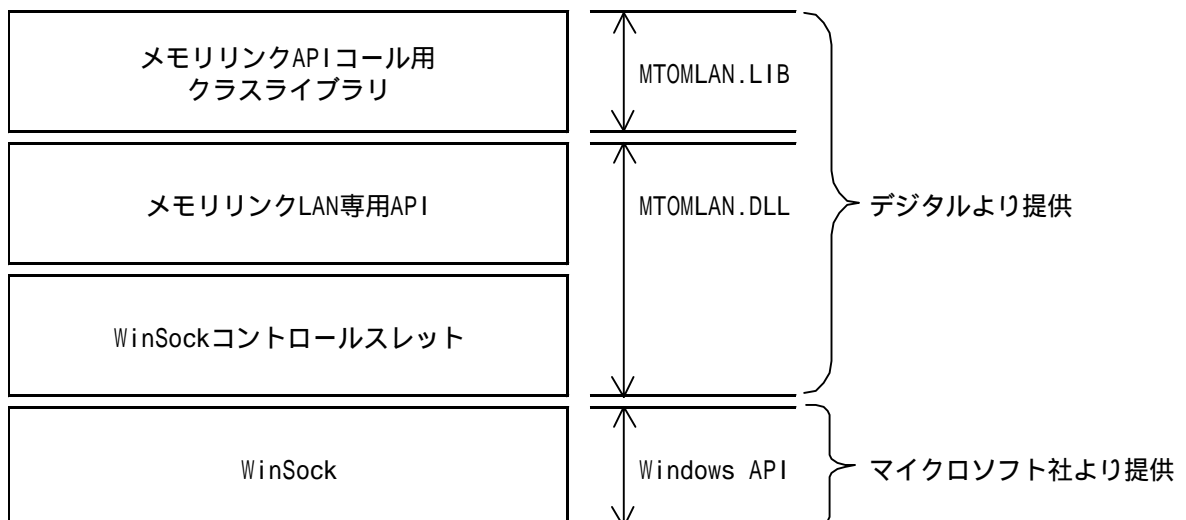
コンパイラ : Microsoft Visual C++™ Ver4.1

OS : Microsoft Windows® 95 以上

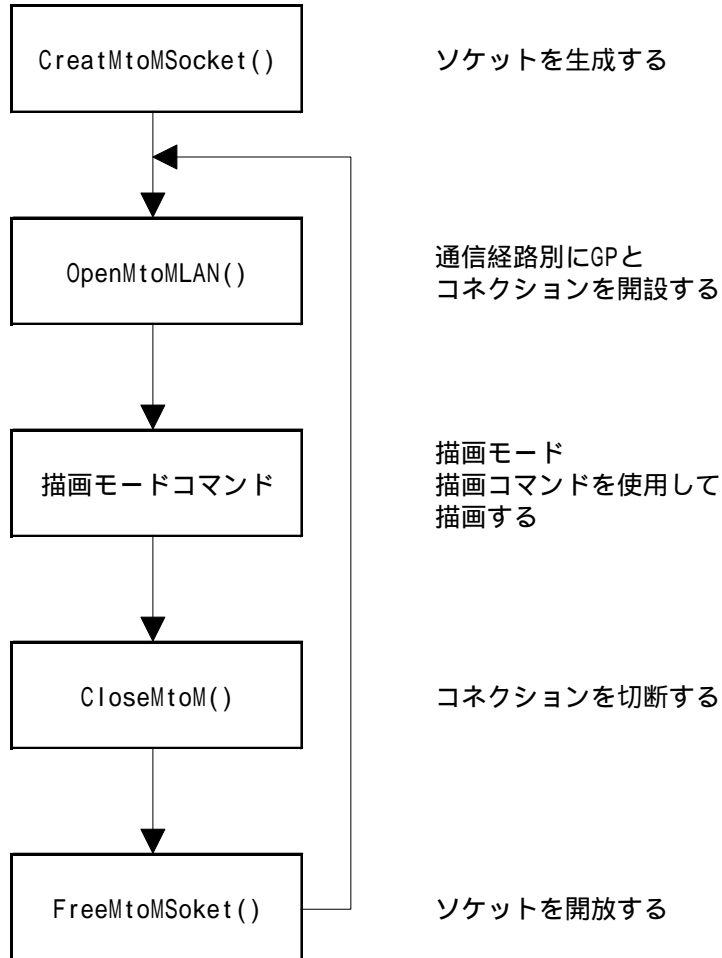
その他 : 以下のファイルは GP-PRO/PBIII for Windows の CD-ROM に収録しています。CD-ROM 内の「MTOMLAN」フォルダにある圧縮ファイル(MTOMLAN.EXE)を解凍してください。

- 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



5-1-1 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。

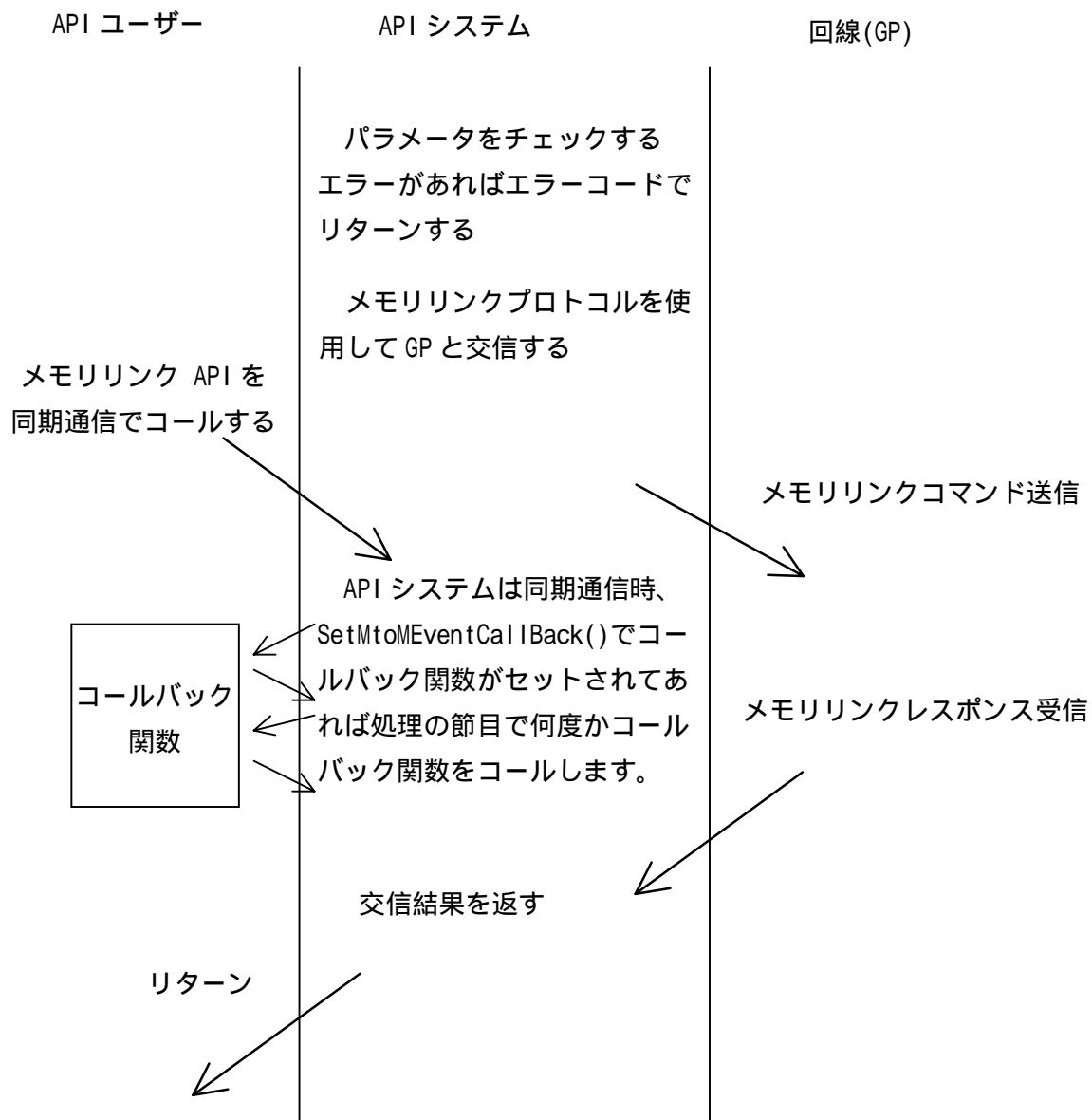
メモリリンク API は、同期 / 非同期両方の通信方法をサポートしています。

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。

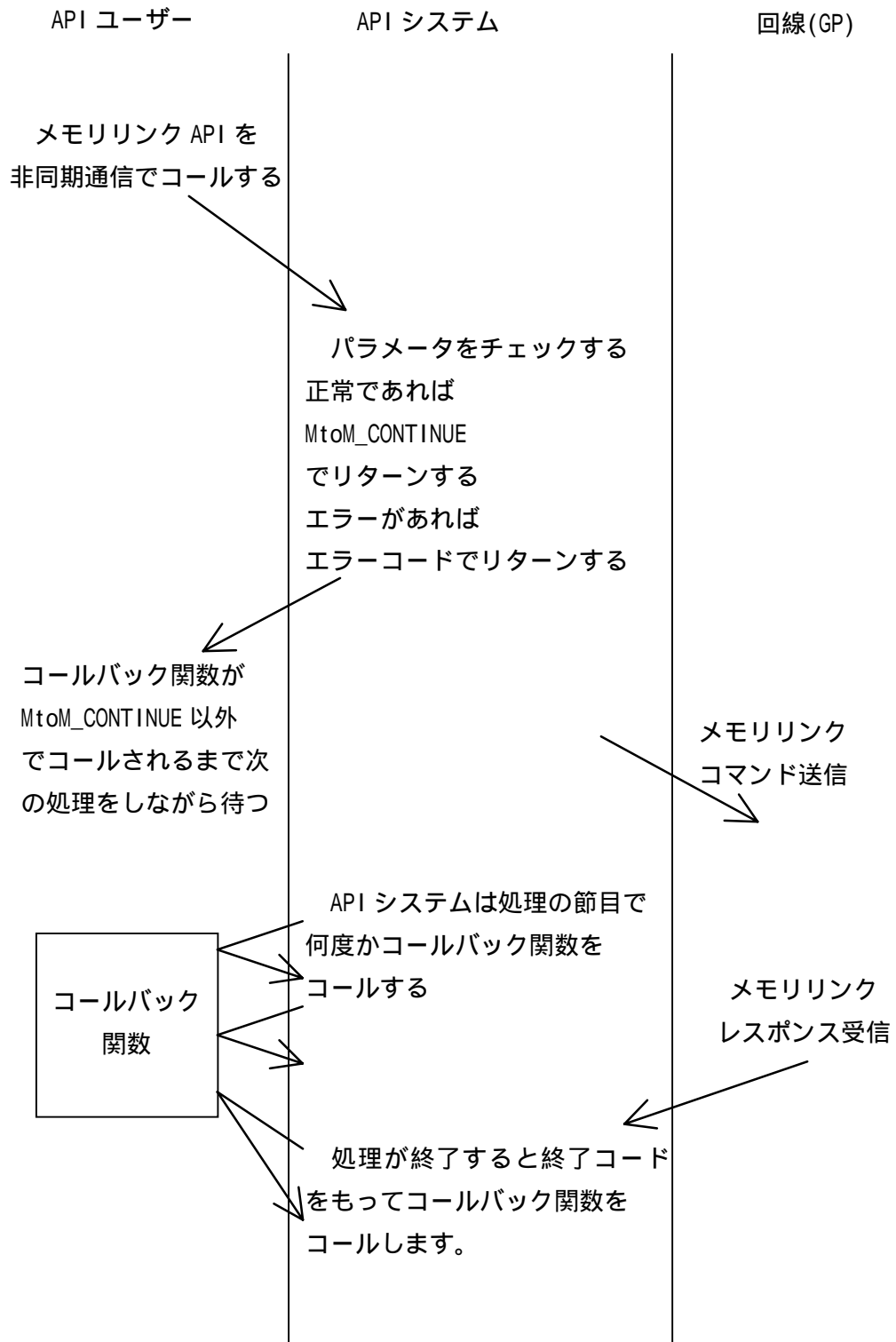


- ・第 2 パラメータが MTOMCALLBACK 型以外のものはすべて同期通信です。
- ・第 2 パラメータの MTOMCALLBACK 型の引数に NULL が指定されると同期通信になります。
- ・第 2 パラメータの MTOMCALLBACK 型の引数に NULL 以外が指定されるとシステムは非同期通信で、かつ、そのときのコールバック関数と判断し処理します。

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 FinishMtoM(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。



- ・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。

5-1-2 ソケットの dwUser1 と dwUser2 メンバーについて

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を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

クラスのコンストラクタでクラスの this ポインタを dwUser1 にセットする。

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。

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。

EventFuncJump() の中で dwUser1 からクラスポインタを取り出し、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。

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){};
};

// イベント発生時のコールバック関数

void CALLBACK EventFuncJump
(LPMtoMSOCK pMSock,int iCode,DWORD dwParam1,DWORD dwpara)
{
    CMSock* pCMSock ;

    pCMSock = (CMSock*)pMSock->swUser1 ;

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

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


5-1-3 交信方法(トランザクションタイプ)

このイーサネット用プロトコルでは4種類の交信方法(トランザクションタイプ)をサポートしています。

1対1交信

相手局(GP)に対し、1対1で交信し、かつ交信結果を保証する交信方法。内部的にはTCP/IPを利用します。

基本的な使用法は、

1. CreateMtoMSocket()でソケットを作成する(ソケットを作成するとデフォルトでこの交信方法になります。)
2. OpenMtoMLAN()でコネクションを開設する。
3. MtoMESC_*()などを利用して交信を行う。
4. CloseMtoM()でコネクションを切断する。
5. FreeMtoMSocket()でコネクションを開放する。

不特定多数交信

不特定多数の局に対して交信する方法で、レスポンスチェックは行いません。よって、交信結果は保証されません。この方法は相手局の処理スピードをいっさい考慮していないので、連続して交信すると相手局が処理しきれない場合があります。

UDP/IPの一斉同報を利用します。一斉同報先のネットIDはネットワークの情報の一斉同報時の対象ネットID(dwNetID)が使用されます。

基本的な使用法は

1. CreateMtoMSocket()でソケットを作成する。
2. SetTransitionType()に不特定多数交信(B_dwTransitionType_BroadCast)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。このときの相手局IPアドレスにはNULLを指定します。
4. MtoMESC_*()などを利用して交信を行う。
5. CloseMtoM()でコネクションを切断する。
6. FreeMtoMSocket()でソケットを開放する。

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描画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。

基本的な使用方法は

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. 対象となるネットワークのネットIDをネットワーク情報(pGPNetworkData)の一斉同報時の対象ネットID(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。



- ・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()で接続を切断する。
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. 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 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。



・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。

5-1-4 座標系

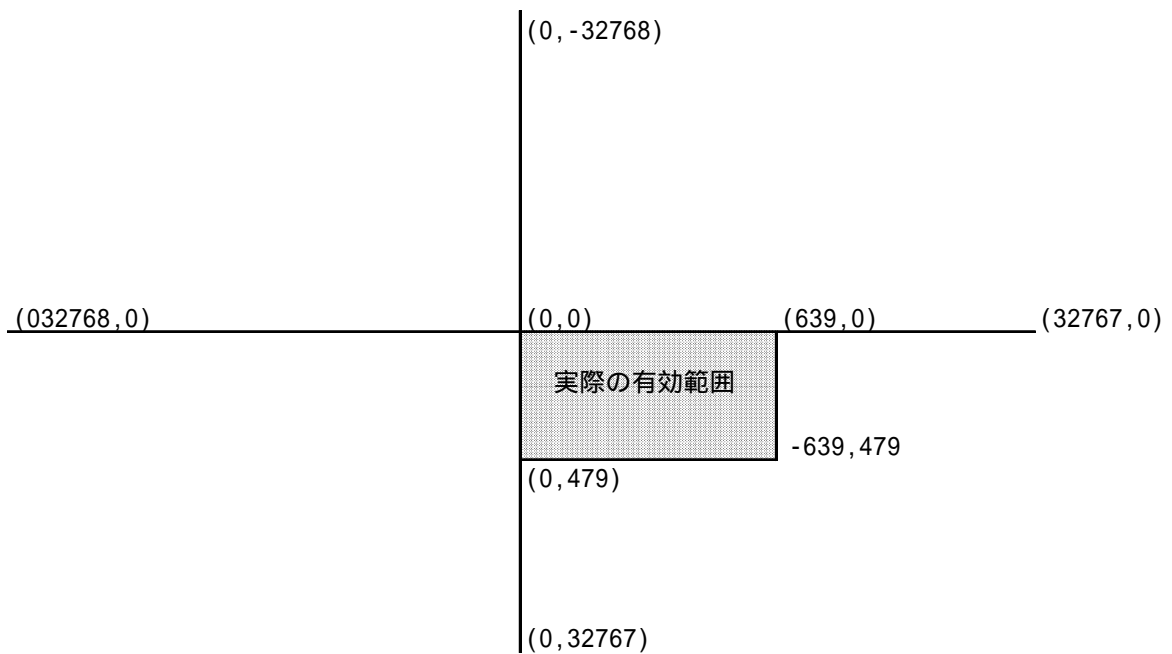
メモリリンク API では GP の画面の左上を $(0,0)$ をする第 3 象限で座標系を表します。
1 ピクセルは 1 ドットです。

座標系の最大範囲は $-32768 \sim 32767$ (16 ビット 2 の補数表現範囲) ですが、実際の表示範囲は使用する GP の解像度で決まります。(解像度を超える値を指定した場合は、自動的にクリッピングされます。)



- ・API の各種パラメータは 32 ビットですが API は GP へ送るとき、32 ビットのデータを 16 ビットへ切り捨てて送信します。

例) 解像度 640 × 480 の場合



5-1-5 ポート番号の設定

メモリリンクAPIでポート番号を変更する場合は下記のようにプログラムで変更します。ポート番号を変更しない場合は初期値として8000番が使用されます。

参照ファイル

- ・ MTOMAPI.H
- ・ MSOCK.CPP

それぞれの内容は以下のとおりです。

```

*****      MTOMAPI.H      *****

// M t o M用ソケットの型定義
struct tagMtoMSOCK {
    DWORD dwStackSize ;           // ソケットの有効データサイズ ( sizeof(MtoMSOCK) )

    DWORD dwProtocolType ; // プロトコルの種類
#define B_ProtocolType_SIO      1    // プロトコルの種類はM t o M S I O
#define B_ProtocolType_LAN      2    // プロトコルの種類はM t o M L A N

    DWORD dwMtoM_Mode ; // 現在のソケットの動作モード
#define B_MtoM_Mode_ESC        0    // 描画モード (ソケット作成時のデフォルト)
#define B_MtoM_Mode_DLE        1    // データ転送モード(MtoMDLE_Communication() ~
MtoMDLE_Quit()まで)

    :

    DWORD dwIPAddress ;           // 通信相手ノードのIPアドレス
    DWORD dwPortNo ;              // 通信相手ノードのポート番号 ←
    DWORD dwLocalIPAddress ;      // 自局ノードのIPアドレス
    DWORD dwLocalPortNo ;         // 自局ノードのポート番号 ←

    void *pDLLWork;
    DWORD dwErrorCode;           // Details of error
};

```

CreateMtoMSocket()関数(サンプルソースのMSOCK.CPPを参照)をコールした後、この値を変更することで、任意のポート番号が使用可能になります。(GPで設定したポート番号を設定してください。)
デフォルト(変更しない場合)は、相手局、自局とも8000となります。

```
***** MSOCK.CPP *****
```

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

        // デフォルトは同期モード
        SyncModeEnable();
    }
}
```

m_pMSockの戻り値がソケットのポインタとなります。

5-2 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
FreeMtoMsoket	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
MtoMFS_FindNode	ネットワークに参加している局を探す
MtoM_ResizeGPNetworkData	ソケット内のネットワーク情報のサイズを変更します
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

5-2-1 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

NULL: 正常終了

Other: 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

5-2-2 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



- ・ IPアドレスは、2通りの方法で指定することが出来ます
IPアドレスをドットで区切って指定する。
例) `szipaddress="11.22.33.44";`
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 ディレクトリー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。
例) C:¥Windows¥HOSTSの内容

11.22.33.44	GP1
-------------	-----

5-2-3 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	<p>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)</p> <p>処理が完了するとその処理結果コードで復帰します。</p> <p>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)</p> <p>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。</p>

5-2-4 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

5-2-5 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

5-2-6 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



- ・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。

5-2-7 ネットワークに参加している局を探す

ネットワークに参加している局を探すコマンドは、以下のとおりです。
探し出したネットワーク情報は LPMtoMSOCK pMSock のネットワーク情報に登録されます。

構文

```
int WINAPI MtoMFS_FindNode(LPMtoMSOCK pMSock, MTOMCALLBACK pfFinish, LPCTSTR szNetID)
```

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	<p>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。</p> <p>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。</p>
LPCTSTR szNetID	ネットワークのネットID

5-2-8 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

5-2-9 交信方法（トランザクションタイプ）を設定する

交信方法（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

設定変更する前の交信方法（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 設定変更する交信方法

- B_dwTransitionType_Only1: 1局指定、コネクションを開設した局に対して交信を行います。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交信する(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、連続して交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)交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。本交信方法は複数局に対しての描画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)交信する。
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。本交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)

5-2-10 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

5-2-11 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。

次のページに、エラーコードの一覧を示します。

GP 関係のエラーに関するエラーコード

参照 第7章エラーメッセージ

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
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	WSAVERNOTSUPPROTED
10050	WSAENETDOWN	10093	WSANOTINITIALISED
10051	WSAENETUNREACH	10101	WSAEDISCON
10052	WSAENETRESET		

5-3 描画モードコマンド

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

描画コマンド一覧

コマンド	アクション
MtoMESC_W	システムエリアヘデータを書き込む
MtoMESC_R	システムエリアヘデータを読み出す
MtoMESC_T	文字列を書く
MtoMESC_L	直線を書く
MtoMESC_B	四角形を書く
MtoMESC_S	塗り込み四角形を書く
MtoMESC_C	円を描く
MtoMESC_A	円弧を描く
MtoMESC_G	扇形を描く
MtoMESC_P	図形の塗り込みを行う
MtoMESC_I	タッチパネルが押されたかどうか問い合わせる
MtoMESC_t	拡張指定で文字列を書く
MtoMESC_l	拡張指定で直線を描く
MtoMESC_b	拡張指定で四角形を描く
MtoMESC_s	拡張指定で塗り込み四角形を描く
MtoMESC_c	拡張指定で円を描く
MtoMESC_g	拡張指定で扇形を描く
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

5-3-1 システムエリアヘデータを書き込む

システムエリアヘデ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 ~ 1FFFh
INT iDataCount	書き込みデータ数を設定します。 0001h ~ 0040h (1 ~ 64)
WORD* pwData	書き込むデータ

5-3-2 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 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。
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 ~ 1FFFh
INT iDataCount	読み出すデータ数を設定します。 0001h ~ 0040h (1 ~ 64)
WORD pwoData	読み出したデータの格納先



- ・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。

5-3-3 文字列を書く

文字列を書くコマンドは、以下のとおりです。

構文

```
int WINAPI MtoMESC_T
(LPMtoMSOCK pMSock, MTOMCALLBACK pfFinish, GPCOLOR cDisplayColor, GPCOLOR cBackColor,
INT iX1, INT iY1, GPFONTSIZE cFontSize, CHAR* szString)
```

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

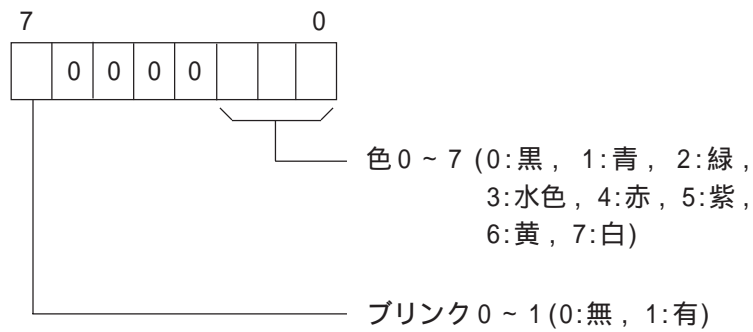
MTOM_CONTINUE : 処理依頼は正常にシステムに通知されました。処理の完了は pfFinish がコールバックされることで通知されます。
Other : エラーコード

引数

LPMtoMSOCK pMSock	メモリリンクのソケットハンドル
MTOMCALLBACK pfFinish	<p>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。</p> <p>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。</p>

GPCOLOR cDisplayColor 表示色を以下の範囲で設定します。

GPCOLOR cBackColor 背景色を以下の範囲で設定します。



- ・ GP-675シリーズの64色、3速ブリンクは使用できません。

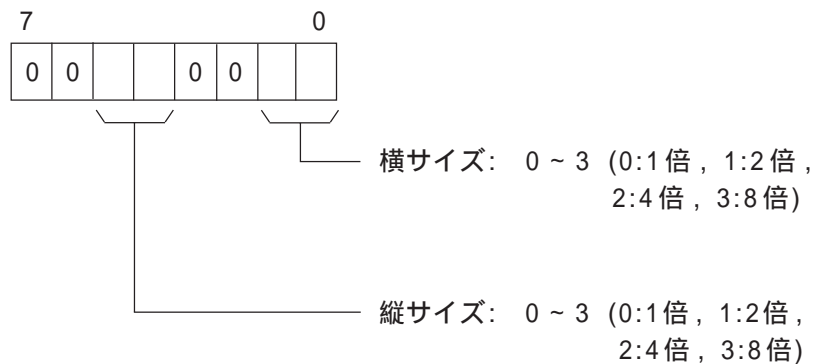
INT iX1 X座標を以下の範囲で設定します。

INT iY1 Y座標を以下の範囲で設定します。

X座標: 0000h ~ (0 ~)

Y座標: 0000h ~ (0 ~)

GPFONTSIZE cFontSize フォントサイズを以下の範囲で設定します。



CHAR* szString

シフト JIS コード

ANK文字は1バイト、漢字は2バイトで設定します。

5-3-4 直線を描く

直線を描くコマンドは、以下のとおりです。

構文

```
int WINAPI MtoMESC_L
(LPMtoMSOCK pMSock, MTOMCALLBACK pfFinish, GPCOLOR cDisplayColor, GPCOLOR
cBackColor, GPLINESTYLE cLineStyle, INT iX1, INT iY1, INT iX2, INT iY2)
```

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

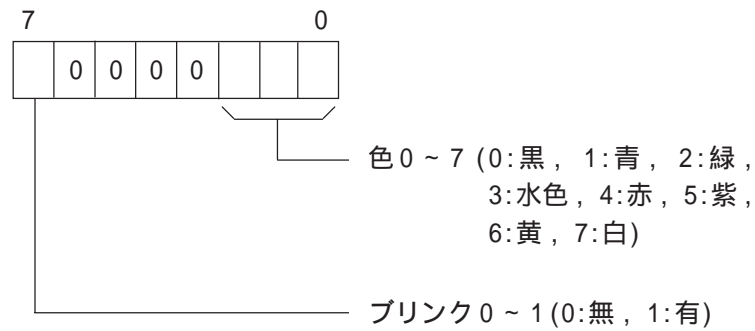
MTOM_CONTINUE : 処理依頼は正常にシステムに通知されました。処理の完了は pfFinish がコールバックされることで通知されます。
Other : エラーコード

引数

LPMtoMSOCK pMSock	メモリリンクのソケットハンドル
MTOMCALLBACK pfFinish	<p>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)</p> <p>処理が完了するとその処理結果コードで復帰します。</p> <p>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)</p> <p>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。</p>

GPCOLOR cDisplayColor 表示色を以下の範囲で設定します。

GPCOLOR cBackColor 背景色を以下の範囲で設定します。



・ GP-675シリーズの64色、3速ブリンクは使用できません。

GPLINESTYLE cLineStyle 線種を以下の範囲で設定します。

00h ~ 07h

(00 : ———、01 : - - - -、02 : — - —、03 : — - - -)

04 : ———、05 : - - - -、06 : — - —、07 : — - - -)

INT iX1 始点X座標を以下の範囲で設定します。

INT iY1 始点Y座標を以下の範囲で設定します。

INT iX2 終点X座標を以下の範囲で設定します。

INT iY2 終点Y座標を以下の範囲で設定します。

X座標 : 0000h ~ (0 ~)

Y座標 : 0000h ~ (0 ~)

5-3-5 四角形を描く

四角形を描くコマンドは、以下のとおりです。

構文

```
int WINAPI MtoMESC_B
(LPMtoMSOCK pMSock, MTOMCALLBACK pfFinish, GPCOLOR cDisplayColor, GPCOLOR
cBackColor, GPLINESTYLE cLineStyle, INT iX1, INT iY1, INT iX2, INT iY2)
```

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

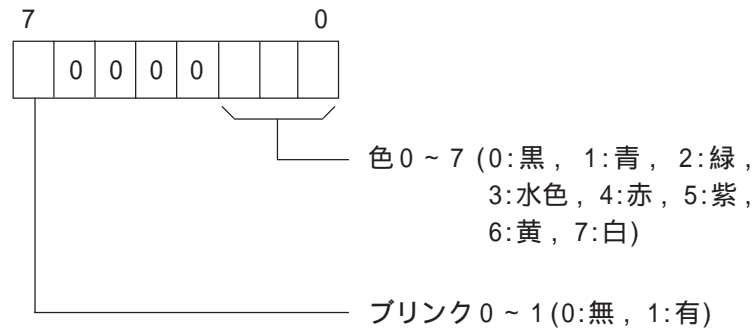
MTOM_CONTINUE : 処理依頼は正常にシステムに通知されました。処理の完了は pfFinish がコールバックされることで通知されます。
Other : エラーコード

引数

LPMtoMSOCK pMSock	メモリリンクのソケットハンドル
MTOMCALLBACK pfFinish	<p>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。</p> <p>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。</p>

GPCOLOR cDisplayColor 表示色を以下の範囲で設定します。

GPCOLOR cBackColor 背景色を以下の範囲で設定します。



- ・ GP-675シリーズの64色、3速ブリンクは使用できません。

GPLINESTYLE cLineStyle 線種を以下の範囲で設定します。

00h ~ 07h

(00 : ———、01 : - - - -、02 : — - —、03 : — - - -)

04 : ———、05 : - - - -、06 : — - —、07 : — - - -)

INT iX1 始点X座標を以下の範囲で設定します。

INT iY1 始点Y座標を以下の範囲で設定します。

INT iX2 終点X座標を以下の範囲で設定します。

INT iY2 終点Y座標を以下の範囲で設定します。

X座標 : 0000h ~ (0 ~)

Y座標 : 0000h ~ (0 ~)

5-3-6 塗り込み四角形を描く

塗り込み四角形を描くコマンドは、以下のとおりです。

構文

```
int WINAPI MtoMESC_S
(LPMtoMSOCK pMSock, MTOMCALLBACK pfFinish, GPCOLOR cDisplayColor, GPCOLOR cBackColor,
INT iX1, INT iY1, INT iX2, INT iY2, GPTILE cTile)
```

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

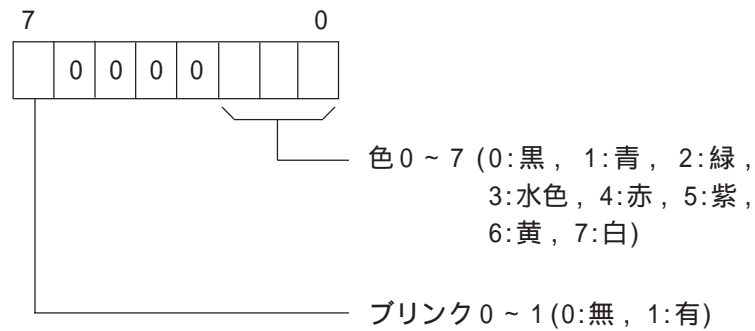
MTOM_CONTINUE : 処理依頼は正常にシステムに通知されました。処理の完了は pfFinish がコールバックされることで通知されます。
Other : エラーコード

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。

GPCOLOR cDisplayColor 表示色を以下の範囲で設定します。

GPCOLOR cBackColor 背景色を以下の範囲で設定します。



- ・ GP-675シリーズの64色、3速ブリンクは使用できません。

INT iX1 始点X座標を以下の範囲で設定します。

INT iY1 始点Y座標を以下の範囲で設定します。

INT iX2 終点X座標を以下の範囲で設定します。

INT iY2 終点Y座標を以下の範囲で設定します。

X座標 : 0000h ~ (0 ~)

Y座標 : 0000h ~ (0 ~)

GPTILE cTile タイリングパターン^{*1}を00h ~ 08hの範囲で設定します。

*1 タイリングパターンの種類

タイリングパターン番号	タイリングパターン	タイリングパターン番号	タイリングパターン	タイリングパターン番号	タイリングパターン
0		3		6	
1		4		7	
2		5		8	

5-3-7 円を描く

円を描くコマンドは、以下のとおりです。

構文

```
int WINAPI MtoMESC_C
(LPMtoMSOCK pMSock, MTOMCALLBACK pfFinish, GPCOLOR cDisplayColor, GPCOLOR
cBackColor, GPLINESTYLE cLineStyle, INT iX1, INT iY1, INT iRadius)
```

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

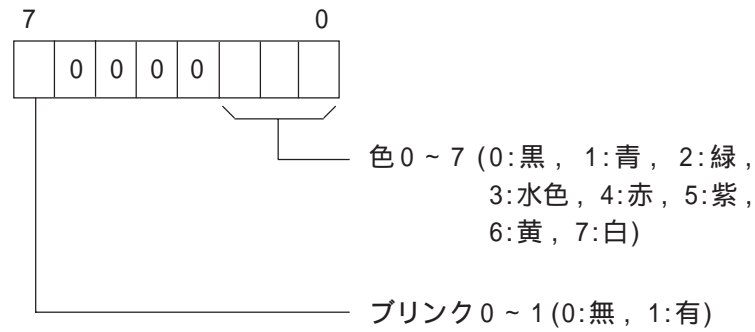
MTOM_CONTINUE : 処理依頼は正常にシステムに通知されました。処理の完了は pfFinish がコールバックされることで通知されます。
Other : エラーコード

引数

LPMtoMSOCK pMSock	メモリリンクのソケットハンドル
MTOMCALLBACK pfFinish	<p>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。</p> <p>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。</p>

GPCOLOR cDisplayColor 表示色を以下の範囲で設定します。

GPCOLOR cBackColor 背景色を以下の範囲で設定します。



- ・ GP-675 シリーズの 64 色、3 速ブリンクは使用できません。

GPLINESTYLE cLineStyle 線種を以下の範囲で設定します。

00h ~ 03h

(00 : ———、01 : - - - -、02 : — - —、03 : — - - -)

INT iX1 中心点 X 座標を以下の範囲で設定します。

INT iY1 中心点 Y 座標を以下の範囲で設定します。

X 座標 : 0000h ~ (0 ~)

Y 座標 : 0000h ~ (0 ~)

INT iRadius 半径を 0001h ~ (1 ~) の範囲で設定します。

5-3-8 円弧を描く

円弧を描くコマンドは、以下のとおりです。

構文

```
int WINAPI MtoMESC_A
(LPMtoMSOCK pMSock, MTOMCALLBACK pfFinish, GPCOLOR cDisplayColor, GPCOLOR
cBackColor, GPLINESTYLE cLineStyle, INT iX1, INT iY1, INT iRadius, INT iStartAngle, INT
iEndAngle)
```

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

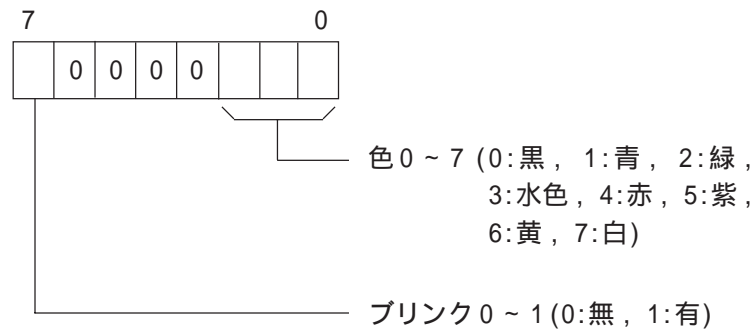
MTOM_CONTINUE : 処理依頼は正常にシステムに通知されました。処理の完了は pfFinish がコールバックされることで通知されます。
Other : エラーコード

引数

LPMtoMSOCK pMSock	メモリリンクのソケットハンドル
MTOMCALLBACK pfFinish	<p>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。</p> <p>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。</p>

GPCOLOR cDisplayColor 表示色を以下の範囲で設定します。

GPCOLOR cBackColor 背景色を以下の範囲で設定します。



- ・ GP-675シリーズの64色、3速ブリンクは使用できません。

GPLINESTYLE cLineStyle 線種を以下の範囲で設定します。

00h ~ 03h

(00 : ———、01 : - - - -、02 : — - —、03 : — - - -)

INT iX1 中心点X座標を以下の範囲で設定します。

INT iY1 中心点Y座標を以下の範囲で設定します。

X座標 : 0000h ~ (0 ~)

Y座標 : 0000h ~ (0 ~)

INT iRadius 半径を0001h ~ (1 ~)の範囲で設定します。

INT iStartAngle 開始角度を以下の範囲で設定します。

INT iEndAngle 終了角度を以下の範囲で設定します。

角度 : 0000h ~ 0168h (0 ~ 360)



- ・ 開始角度と終了角度を同じにしないでください。
- ・ 描画方向は反時計回りです。

5-3-9 扇形を描く

扇形を描くコマンドは、以下のとおりです。

構文

```
int WINAPI MtoMESC_G
(LPMtoMSOCK pMSock, MTOMCALLBACK pfFinish, GPCOLOR cDisplayColor, GPCOLOR
cBackColor, GPLINESTYLE cLineStyle, INT iX1, INT iY1, INT iRadius, INT iStartAngle, INT
iEndAngle)
```

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

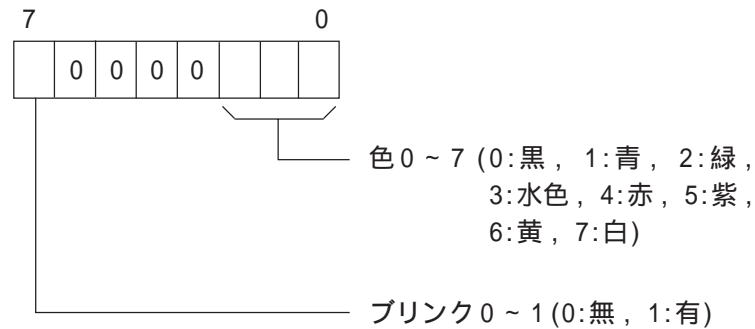
MTOM_CONTINUE : 処理依頼は正常にシステムに通知されました。処理の完了は pfFinish がコールバックされることで通知されます。
Other : エラーコード

引数

LPMtoMSOCK pMSock	メモリリンクのソケットハンドル
MTOMCALLBACK pfFinish	<p>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。</p> <p>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。</p>

GPCOLOR cDisplayColor 表示色を以下の範囲で設定します。

GPCOLOR cBackColor 背景色を以下の範囲で設定します。



- ・ GP-675 シリーズの 64 色、3 速ブリンクは使用できません。

GPLINESTYLE cLineStyle 線種を以下の範囲で設定します。

00h ~ 03h

(00 : ———、01 : - - - -、02 : — - —、03 : — - - -)

INT iX1 中心点 X 座標を以下の範囲で設定します。

INT iY1 中心点 Y 座標を以下の範囲で設定します。

X 座標 : 0000h ~ (0 ~)

Y 座標 : 0000h ~ (0 ~)

INT iRadius 半径を 0001h ~ (1 ~) の範囲で設定します。

INT iStartAngle 開始角度を以下の範囲で設定します。

INT iEndAngle 終了角度を以下の範囲で設定します。

角度 : 0000h ~ 0168h (0 ~ 360)



- ・ 開始角度と終了角度を同じにしないでください。
- ・ 描画方向は反時計回りです。

5-3-10 図形の塗り込みを行う

図形の塗り込みを行うコマンドは、以下のとおりです。

構文

```
int WINAPI MtoMESC_P
(LPMtoMSOCK pMSock, MTOMCALLBACK pfFinish, GPCOLOR cDisplayColor, GPCOLOR cBackColor,
GPTILE cTile, GPCOLOR cLimitColor, INT iX1, INT iY1)
```

戻り値

pfFinish が NULL のとき

00 : 正常終了
Other : エラーコード

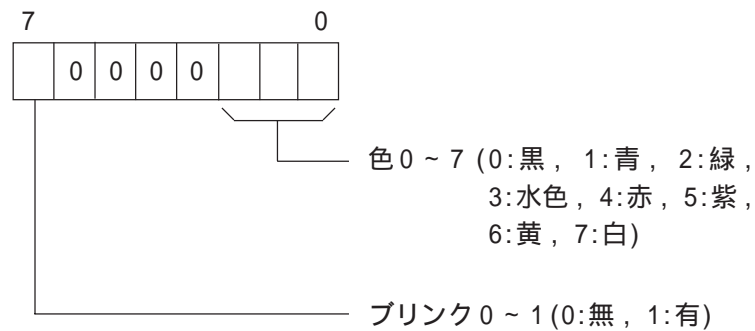
pfFinish が NULL 以外のとき

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	<p>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。</p> <p>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。</p>

GPCOLOR cDisplayColor 表示色を以下の範囲で設定します。
 GPCOLOR cBackColor 背景色を以下の範囲で設定します。
 GPCOLOR cLimitColor 境界色を以下の範囲で設定します。



- ・ GP-675シリーズの64色、3速ブリンクは使用できません。
- ・ 境界色のブリンクは「無」にしてください。

GPTILE cTile タイリングパターンを00h ~ 08hの範囲で設定します。
 参照 5-3-6「塗り込み四角形を描く」タイリングパターンの種類

INT iX1 始点X座標を以下の範囲で設定します。

INT iY1 始点Y座標を以下の範囲で設定します。

X座標 : 0000h ~ (0 ~)

Y座標 : 0000h ~ (0 ~)

5-3-11 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

5-3-12 拡張指定で文字列を書く

拡張指定で文字列を書くコマンドは、以下のとおりです。

拡張の内容は回転、方向、拡張強調の追加です。

構文

```
int WINAPI MtoMESC_t
(LPMtoMSOCK pMSock, MTOMCALLBACK pfFinish, GPCOLOR cDisplayColor, GPCOLORBackColor,
GPFONT cFont, INT iTurn, INT iDirection, INT iHalfcentering, INT iEmphasis, INT iSculpture,
INT iX1, INT iY1, GPFONTSIZE cFontSize, CHAR* szString)
```

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

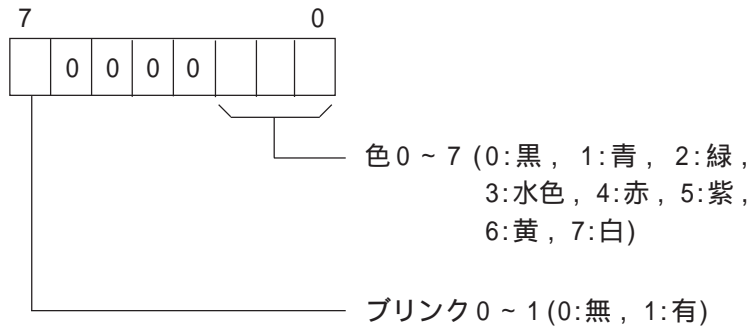
MTOM_CONTINUE : 処理依頼は正常にシステムに通知されました。処理の完了は pfFinish がコールバックされることで通知されます。
Other : エラーコード

引数

LPMtoMSOCK pMSock	メモリリンクのソケットハンドル
MTOMCALLBACK pfFinish	<p>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。</p> <p>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。</p>

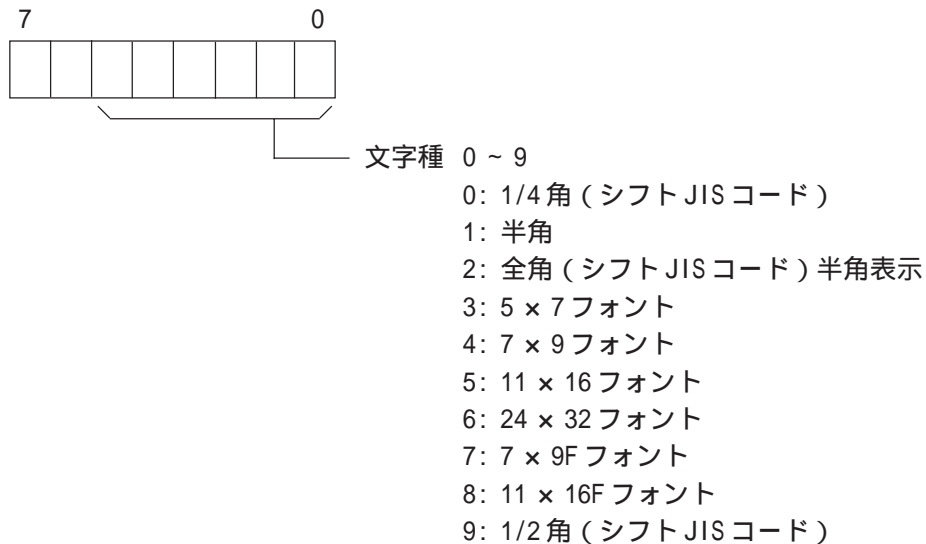
GPCOLOR cDisplayColor 表示色を以下の範囲で設定します。

GPCOLOR cBackColor 背景色を以下の範囲で設定します。



- ・ GP-675 シリーズの 64 色、3 速ブリンクは使用できません。

PFONT cFont 文字種を以下の範囲で設定します。



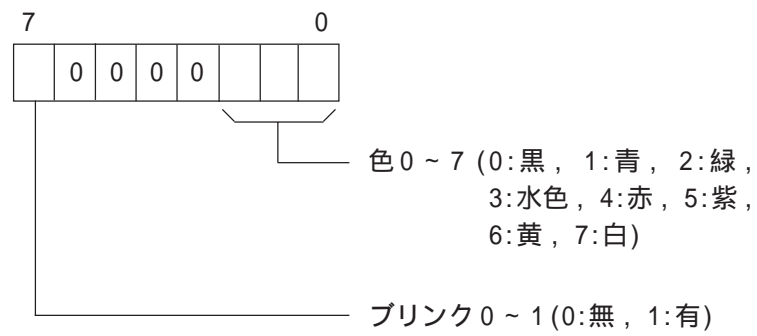
INT iTurn 文字の角度を以下の範囲で設定します。
00 ~ 03h (00:0 度、01:90 度、02:180 度、03:270 度)

INT iDirection 文字の方向を以下の範囲で設定します。
00 ~ 01h (00:横並び、01:縦並び)

INT iHalfcentering 文字の半角センタリングを以下の範囲で設定します。
00 ~ 01h (00:無し、01:有り・表示方向縦並びのみ有効)

INT iEmphasis 文字の強調を以下の範囲で設定します。
00 ~ 02h (00:ノーマル、01:強調文字、02:彫刻文字)

INT iSculpture 彫刻色を以下の範囲で設定します。



- ・ GP-675シリーズの64色、3速ブリンクは使用できません。

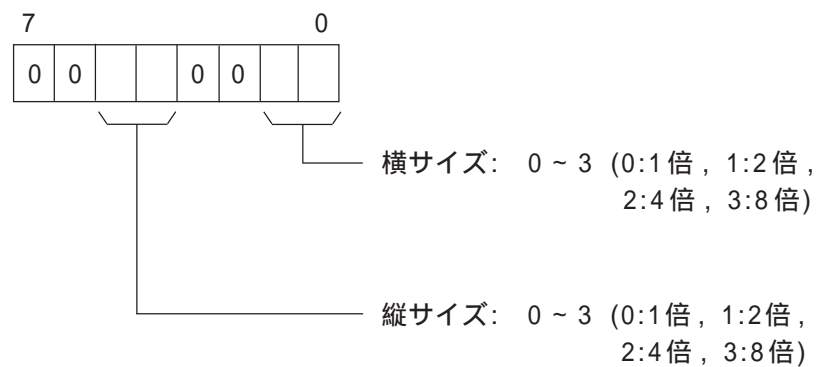
INT iX1 X座標を以下の範囲で設定します。

INT iY1 Y座標を以下の範囲で設定します。

X座標: 0000h ~ (0 ~)

Y座標: 0000h ~ (0 ~)

GPFONTSIZE cFontSize フォントサイズを以下の範囲で設定します。



CHAR* szString

シフトJISコード

ANK文字は1バイト、漢字は2バイトで設定します。

5-3-13 拡張指定で直線を描く

拡張指定で直線を描くコマンドは、以下のとおりです。
拡張内容は矢印の追加です。

構文

```
int WINAPI MtoMESC_1
(LPMtoMSOCK pMSock, MTOMCALLBACK pfFinish, GPCOLOR cDisplayColor,
GPCOLORcBackColor, GPLINESTYLE cLineStyle, INT iArrowPattern, INT ArrowDirection,
INT iX1, INT iY1, INT iX2, INT iY2)
```

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

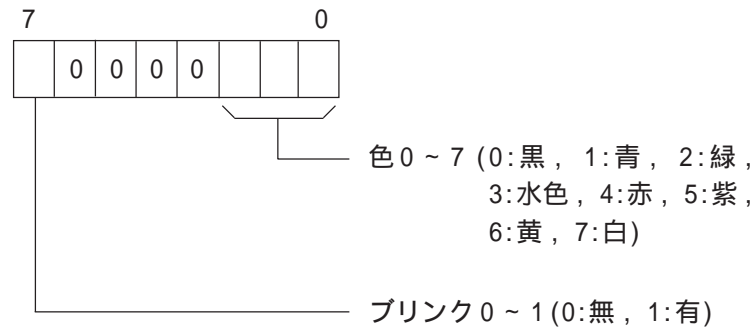
MTOM_CONTINUE : 処理依頼は正常にシステムに通知されました。処理の完了は pfFinish がコールバックされることで通知されます。
Other : エラーコード

引数

LPMtoMSOCK pMSock	メモリリンクのソケットハンドル
MTOMCALLBACK pfFinish	<p>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。</p> <p>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。</p>

GPCOLOR cDisplayColor 表示色を以下の範囲で設定します。

GPCOLOR cBackColor 背景色を以下の範囲で設定します。



- ・ GP-675 シリーズの 64 色、3 速ブリンクは使用できません。

GPLINESTYLE cLineStyle 線種を以下の範囲で設定します。

00h ~ 07h

(00 : ———、01 : - - - -、02 : — - —、03 : — - - -
04 : ———、05 : - - - -、06 : — - —、07 : — - - -)

INT iArrowPattern 矢印パターンを以下の範囲で設定します

00 ~ 03h

(00 : 無し、01 : 矢印 1、02 : リザーブ、03 : リザーブ)

INT iCornerRadius 矢印方向を以下の範囲で設定します。

00 ~ 01 h (00:両端、01:終点)

INT iX1 始点 X 座標を以下の範囲で設定します。

INT iY1 始点 Y 座標を以下の範囲で設定します。

INT iX2 終点 X 座標を以下の範囲で設定します。

INT iY2 終点 Y 座標を以下の範囲で設定します。

X 座標 : 0000h ~ (0 ~)

Y 座標 : 0000h ~ (0 ~)

5-3-14 拡張指定で四角形を描く

拡張指定で四角形を描くコマンドは、以下のとおりです。

拡張内容は面取りの追加です。

構文

```
int WINAPI MtoMESC_b
(LPMtoMSOCK pMSock, MTOMCALLBACK pfFinish, GPCOLOR cDisplayColor, GPCOLOR cBackColor,
GPLINestyle cLineStyle, INT iCornerDirection, INT iCornerRadius,
INT iX1, INT iY1, INT iX2, INT iY2)
```

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

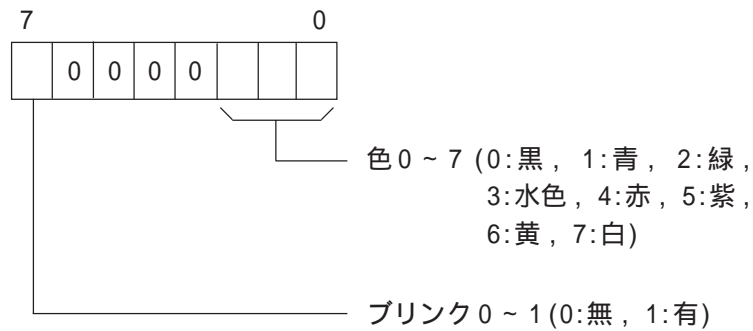
MTOM_CONTINUE : 処理依頼は正常にシステムに通知されました。処理の完了は pfFinish がコールバックされることで通知されます。
Other : エラーコード

引数

LPMtoMSOCK pMSock	メモリリンクのソケットハンドル
MTOMCALLBACK pfFinish	<p>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。</p> <p>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。</p>

GPCOLOR cDisplayColor 表示色を以下の範囲で設定します。

GPCOLOR cBackColor 背景色を以下の範囲で設定します。



- ・ GP-675シリーズの64色、3速ブリンクは使用できません。

GPLINESTYLE cLineStyle 線種を以下の範囲で設定します。

00h ~ 03h、08h、09h

(00 : ———、01 : - - - -、02 : — - - -、03 : — - - -、
08 : ———、09 : ———)



- ・ 線種は0 ~ 3が1ドット線、8は3ドット線、9は5ドット線です。

INT iCornerDirection 面取方法を以下の範囲で設定します。

00 ~ 02h (00 : 無し、01 : 曲線、02 : 直線)

INT iCornerRadius 面取半径を以下の範囲で設定します。

00 ~ 32h

INT iX1 始点X座標を以下の範囲で設定します。

INT iY1 始点Y座標を以下の範囲で設定します。

INT iX2 終点X座標を以下の範囲で設定します。

INT iY2 終点Y座標を以下の範囲で設定します。

X座標 : 0000h ~ (0 ~)

Y座標 : 0000h ~ (0 ~)

5-3-15 拡張指定で塗り込み四角形を描く

拡張指定で塗り込み四角形を描くコマンドは、以下のとおりです。
拡張内容は面取りの追加です。

構文

```
int WINAPI MtoMESC_s
(LPMtoMSOCK pMSock, MTOMCALLBACK pfFinish, GPCOLOR cDisplayColor, GPCOLOR cBackColor,
GPTILE cTile, INT iCornerDirection, INT iCornerRadius, INT iX1, INT iY1, INT iX2, INT iY2,)
```

戻り値

pfFinish が NULL のとき

00 : 正常終了
Other : エラーコード

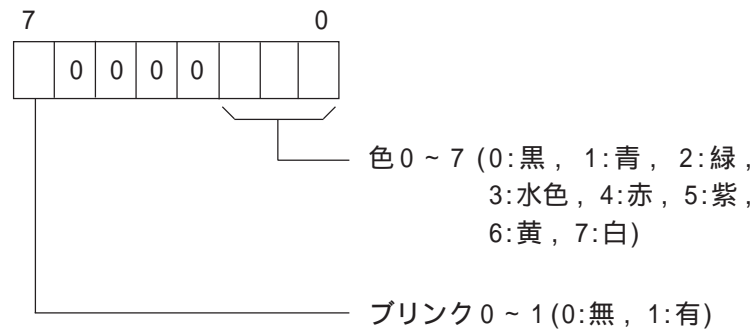
pfFinish が NULL 以外のとき

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	<p>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。</p> <p>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。</p>

GPCOLOR cDisplayColor 表示色を以下の範囲で設定します。
 GPCOLOR cBackColor 背景色を以下の範囲で設定します。



- ・ GP-675シリーズの64色、3速ブリンクは使用できません。

GPTILE cTile タイリングパターンを00h ~ 08hの範囲で設定します。
 参照 5-3-6「塗り込み四角形を描く」タイリングパターンの種類

INT iCornerDirection 面取方法を以下の範囲で設定します。
 00 ~ 02h (00:無し、01:曲線、02:直線)

INT iCornerRadius 面取半径を以下の範囲で設定します。
 00 ~ 32h

INT iX1 始点X座標を以下の範囲で設定します。

INT iY1 始点Y座標を以下の範囲で設定します。

INT iX2 終点X座標を以下の範囲で設定します。

INT iY2 終点Y座標を以下の範囲で設定します。

X座標: 0000h ~ (0 ~)
 Y座標: 0000h ~ (0 ~)

5-3-16 拡張指定で円を描く

拡張指定で円を描くコマンドは、以下のとおりです。
拡張内容はタイリングパターンの追加です。

構文

```
int WINAPI MtoMESC_c
(LPMtoMSOCK pMSock, MTOMCALLBACK pfFinish, GPCOLOR cDisplayColor,
GPCOLOR cBackColor, GPTILE cTile, INT iX1, INT iY1, INT iRadius)
```

戻り値

pfFinish が NULL のとき

00 : 正常終了
Other : エラーコード

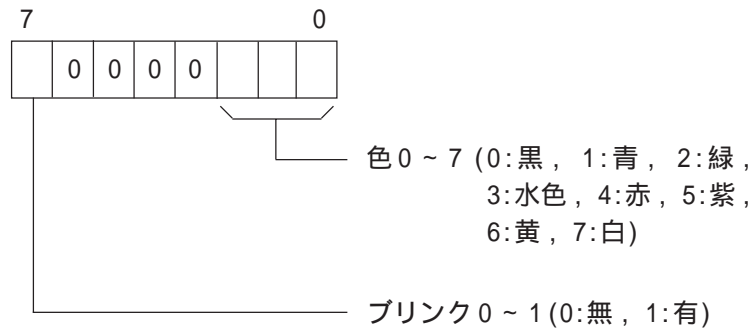
pfFinish が NULL 以外のとき

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	<p>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。</p> <p>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。</p>

GPCOLOR cDisplayColor 表示色を以下の範囲で設定します。
 GPCOLOR cBackColor 背景色を以下の範囲で設定します。



・ GP-675シリーズの64色、3速ブリンクは使用できません。

GPTILE cTile タイリングパターンを00h ~ 08hの範囲で設定します。
 参照 5-3-6「塗り込み四角形を描く」タイリングパターンの種類

INT iX1 中心点X座標を以下の範囲で設定します。
 INT iY1 中心点Y座標を以下の範囲で設定します。
 X座標 : 0000h ~ (0 ~)
 Y座標 : 0000h ~ (0 ~)

INT iRadius 半径を0001h ~ (1 ~)の範囲で設定します。

5-3-17 拡張指定で扇形を描く

拡張指定で扇形を描くコマンドは、以下のとおりです。

拡張内容は線種の追加です。

構文

```
int WINAPI MtoMESC_g
(LPMtoMSOCK pMSock, MTOMCALLBACK pfFinish, GPCOLOR cDisplayColor, GPCOLOR cBackColor,
GPLINESTYLE cLineStyle, INT iX1, INT iY1, INT iRadius, INT iStartAngle, INT iEndAngle)
```

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

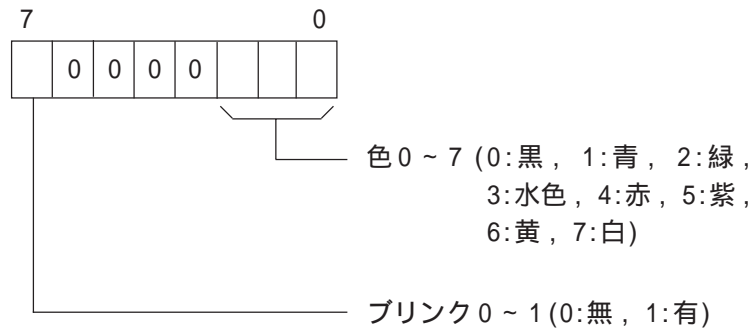
MTOM_CONTINUE : 処理依頼は正常にシステムに通知されました。処理の完了は pfFinish がコールバックされることで通知されます。
Other : エラーコード

引数

LPMtoMSOCK pMSock	メモリリンクのソケットハンドル
MTOMCALLBACK pfFinish	<p>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。</p> <p>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。</p>

GPCOLOR cDisplayColor 表示色を以下の範囲で設定します。

GPCOLOR cBackColor 背景色を以下の範囲で設定します。



- ・ GP-675シリーズの64色、3速ブリンクは使用できません。

GPLINESTYLE cLineStyle 線種を以下の範囲で設定します。

00h ~ 03h、08h、09h

(00 : ———、01 : - - - -、02 : — - -、03 : — - - -、

08 : ———、09 : ———)



- ・ 線種は0 ~ 3が1ドット線、8は3ドット線、9は5ドット線です。

INT iX1 中心点X座標を以下の範囲で設定します。

INT iY1 中心点Y座標を以下の範囲で設定します。

X座標 : 0000h ~ (0 ~)

Y座標 : 0000h ~ (0 ~)

INT iRadius 半径を0001h ~ (1 ~)の範囲で設定します。

INT iStartAngle 開始角度を以下の範囲で設定します。

INT iEndAngle 終了角度を以下の範囲で設定します。

角度 : 0000h ~ 0168h (0 ~ 360)



- ・ 開始角度と終了角度を同じにしないでください。
- ・ 描画方向は反時計回りです。

5-3-18 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	<p>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。</p> <p>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。</p>
dwContrast	<p>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)</p>
dwLight	<p>輝度調整 (0000h ~ 0001h) 0:明 ~ 1:暗 FFFFFFFFhで設定値無し(設定がない機種)</p>

5-3-19 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	<p>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)</p> <p>処理が完了するとその処理結果コードで復帰します。</p> <p>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)</p> <p>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。</p>
dwContrast	<p>コントラストの現在値</p> <p>(0000h ~ 0007h) 0:明 ~ 7:暗</p> <p>FFFFFFFFh で設定値無し(設定がない機種)</p>
dwLight	<p>輝度の現在値</p> <p>(0000h ~ 0001h) 0:明 ~ 1:暗</p> <p>FFFFFFFFh で設定値無し(設定がない機種)</p>

5-4 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_USERS_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のパラメータエラー、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



- ・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。

第6章

サンプルプログラム

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 (GpM.EXE) について説明します。GpM.EXE は Windows と GP を 1 対 1、もしくは 1 対 n で接続し、Windows 上で作画した図形をリアルタイムに GP 上で表示することのできるサンプルプログラムです。

6-1

メモリリンク LAN API サンプルプログラム

起動環境

GpM.EXE は Windows®95 上で動作します。

GpM.EXE は MtoMLAN.DLL を使用するので MtoMLAN.DLL を Windows デレクトリーにコピーして下さい。

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下さい。



- GpM.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、設定を見直して下さい。

開発環境

GpM.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/PBIII for Windows 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 (GpM.EXE) が生成されます。

コンパイラ : Microsoft® Visual C++™ Ver.4.1

OS : Microsoft® Windows® 95

基本構成

GpM.EXE は MDI (マルチドキュメントインターフェース) を利用します。1 対 1 モードのときは 1 ドキュメント・1 ビュー が 1GP をコントロールしています。

ただし、1 対 n モードの時は 1 ドキュメント・1 ビューが一斉同報機能を利用する事によりネットワーク上の全ての GP をコントロールします。

本ソフトを理解する為には Microsoft® の MFC の MDI のフレームワークがドキュメントとビューをどのように管理し、関係しているかを知る必要があります。

メモリリンク API アクセス方法

GpM.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せん。

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

GpM.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 GpM のアプリケーションクラスです。CGpMDoc は GpM 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。

描画方法

GpM.EXE は 1 つの描画方法 (例えば、文字列を表示したり、線を表示したりする方法) ごとに 2 つのオブジェクトがあります。

描画ペンオブジェクト

これはユーザーに図形指定させるオブジェクトです。(ソース上では表示指示とコメントしてあります) さらに、このオブジェクトは大きく 2 つの部分に別れています。

一つは 図形のプロパティをユーザーに指定してもらるプロパティダイアログ。

もう一つは、マウスをつかって、図形の位置などをユーザーに指定してもらう部分です。

CGpPenEsc* から始まるクラスがこれで CGpPenEsc から派生しています。

コンテンツオブジェクト

描画ペンオブジェクトでユーザーが指定した図形 (データ) を CMSock を使用し GP と通信したり、ウィンドウ上に表示したり、シリアライズ (データの保存 / 読み出し) を行うオブジェクトです。

CContentsEsc* から始まるクラスがこれで CContents から派生しています。

クラス CGpMDOC

GpM.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

関係づけられている 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。

クラス CGpList

ネットワークに参加して GP を検索するクラスです。ソケットとしては CGpMApp を使用しています。

GpWin.Cpp

これは単一のクラスではありませんが、GP の作図を Windows® 上のウィンドウでシュミレートするためのライブラリーです。

但し、本ライブラリーは 100% 完全に GP の作図をシュミレートしているわけではありません。Windows® の GDI を単純に利用してシュミレートできる範囲までしかサポートしていません。



- ・ Windows®95 では GDI がサポートされていないので太い点線は、描画されません。

MtoMAPI.H と MtoMLAM.LIB

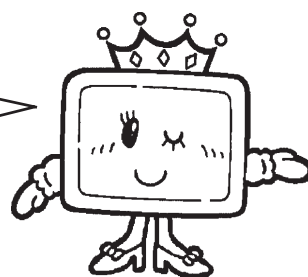
GpM.EXE は MtoMAPI.H を外部からインクルードしています。

MtoMAPI.H は付属されている FD の MtoMLAN ディレクトリーに入っています。これを適当なディレクトリーにコピーし、そのコピーした位置を defsf ile.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。

MEMO

このページは、空白です。
ご自由にお使いください。



第7章

エラーメッセージ

メモリリンク通信時に発生するエラーメッセージについて説明します。エラーメッセージの内容を確認して適切な処置を行ってください。

7-1

エラーメッセージ一覧

GPとホストの通信設定が一致していない場合や、ホストから送られてくるデータに異常がありGPと通信できない場合に表示されます。

「上位通信エラー」というメッセージに続いて次のようなメッセージが表示されます。

上位通信エラー (02:**)

** : エラーコード

エラーコード	内容	処置方法
10	未定義コマンドを受信した。	<ul style="list-style-type: none"> ・ホスト側で送信しているデータの異常が考えられます。送信データを確認してください。正しいデータを送信してください。 ・SIOエラーが発生していることが考えられます。通信設定を確認してください。接続環境（ノイズ等）を確認してください。
15	指定した表示属性がフォーマット外のデータである。	
16	指定した文字サイズがフォーマット外のデータである。	
17	指定した座標データがフォーマット外のデータである。	
18	指定した線種コードがフォーマット外のデータである。	
19	指定したタイリングパターンがフォーマット外のデータである。	
1A	指定した半径が表示域を越えている。	
1B	指定した開始角度 / 終了角度がフォーマット外のデータである。	
1C	指定した文字種コードがフォーマット外のデータである。	
1D	指定した回転コードがフォーマット外のデータである。	
1E	指定した方向コードがフォーマット外のデータである。	
1F	指定した強調コードがフォーマット外のデータである。	

エラーコード	内容	処置方法
20	指定した矢印パターンがフォーマット外のデータである。	・ホスト側で送信しているデータの異常が考えられます。送信データを確認してください。正しいデータを送信してください。
21	指定した矢印方向コードがフォーマット外のデータである。	
22	指定した面取り方法データがフォーマット外のデータである。	・SIOエラーが発生していることが考えられます。通信設定を確認してください。接続環境（ノイズ等）を確認してください。
23	指定したセンターリングコードがフォーマット外のデータである。	
FA	指定したシステムエリアのアドレスが範囲外です。	
FB	システムエリアの範囲を越えて書き込み/読み込みを行った。	
FC	GPが受信したデータフォーマットに異常があった。	
FF	GPがデータ送信できない状態が10秒以上続いた。	・通信制御が正しく行われていないことが考えられます。通信ケーブルを確認してください。

プロトコルスタックのエラーコード

プロトコルスタックエラーからのエラーコードの表示は、以下のように表示されます。

下記のエラーが発生したときは、GPのイーサネット情報およびホスト側の設定を確認してください。

上位通信エラー（02:FE:**）

**：エラーコード

エラーコード	内容
00	初期化で自局IPアドレスの設定エラー
05	初期化に失敗しました
06	通信中止処理に失敗しました
07	初期化が正常に終了しない状態で、開設しようとしてしました
08	自局ポート番号エラー
09	相手局ポート番号エラー
0A	相手局IPアドレスエラー
0B	UDPにて既に同じポート番号で開設しています
0C	TCPにて既に同じ相手と同じポート番号でコネクションを開設しています
0D	プロトコルスタックが開設を拒否しました
0E	プロトコルスタックが開設失敗を返してきました
0F	コネクションが切断されました
10	すべてのコネクションが使用中で、空きコネクションはありません