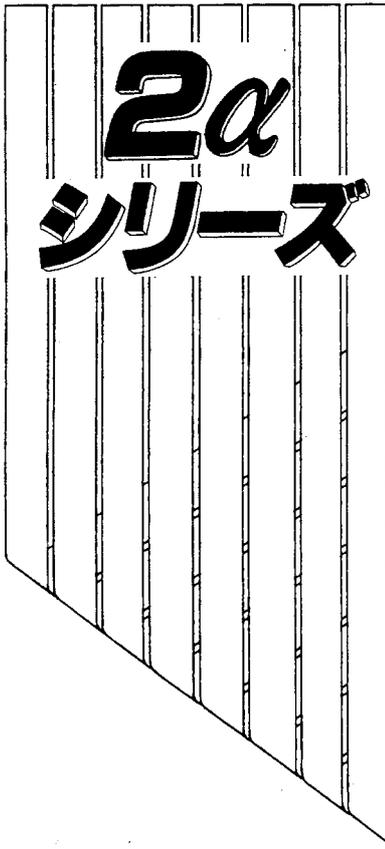


ハードウェアマニュアル  
オプション

# 外部機器リンク (無手順)



対象機種

HIDIC-S10/2 $\alpha$	NESP-S25E
HIDIC-S10/2 $\alpha$ E	NESP-2 $\alpha$ E
HIDIC-S10/2 $\alpha$ H	NESP-2 $\alpha$ H
HIDIC-S10/2 $\alpha$ Hf	NESP-2 $\alpha$ Hf

HITACHI

本製品を輸出される場合には、外国為替及び外国貿易法の規制並びに米国輸出管理規則など外国の輸出関連法規をご確認の上、必要な手続きをお取りください。  
なお、不明な場合は、弊社担当営業にお問合わせください。

1988年	3月	(第1版)	SP-2-007	(廃版)
1989年	3月	(第2版)	SP-2-107	(廃版)
1989年	6月	(第3版)	SP-2-207	(廃版)
1992年	3月	(第4版)	SP-2-307	(廃版)
1997年	5月	(第5版)	SAJ-2-105	(A)

- このマニュアルの一部、または全部を無断で転写したり複製することは、固くお断りいたします。
- このマニュアルの内容を、改良のため予告なしに変更することがあります。

## 安全上のご注意

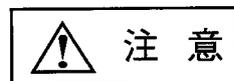
取付、運転、保守・点検の前に必ずこのマニュアルとその他の付属書類をすべて熟読し、正しくご使用ください。機器の知識、安全の情報そして注意事項のすべてについて熟読してご使用ください。また、このマニュアルは最終保守責任者のお手元に必ず届くようにしてください。

このマニュアルでは、安全注意事項のランクを「危険」「注意」として区分してあります。



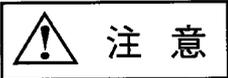
**危険**

：取り扱いを誤った場合に、危険な状況が起こりえて、死亡または重傷を受ける可能性が想定される場合。



**注意**

：取り扱いを誤った場合に、危険な状況が起こりえて、中程度の障害や軽傷を受ける可能性が想定される場合および物的障害だけの発生が想定される場合。

なお、 **注意** に記載した事項でも、状況によっては重大な結果に結びつく可能性があります。

いずれも重要な内容を記載していますので必ず守ってください。

禁止、強制の絵表示の説明を次に示します。



：禁止（してはいけないこと）を示します。例えば分解禁止の場合は  となります。



：強制（必ずしなければならないこと）を示します。例えば接地の場合は  となります。

### 1. 取付について

#### **注意**

- カタログ、マニュアルに記載の環境で使用してください。  
高温、多湿、じんあい、腐食性ガス、振動、衝撃がある環境で使用すると感電、火災、誤動作の原因となることがあります。
- マニュアルにしたがって取り付けてください。  
取り付けに不備があると落下、故障、誤動作の原因となることがあります。
- 電線くずなどの異物を入れないでください。  
火災、故障、誤動作の原因となることがあります。

## 2. 配線について

### 強制

- 必ず接地 (FG) を行ってください。  
接地しない場合は、感電、誤動作のおそれがあります。

### 注意

- 定格にあった電源を接続してください。  
定格と異なった電源を接続すると火災の原因になることがあります。
- 配線作業は、資格のある専門家が行ってください。  
配線を誤ると火災、故障、感電のおそれがあります。

## 3. 使用上の注意

### 危険

- 通電中は端子に触れないでください。  
感電のおそれがあります。
- 非常停止回路、インタロック回路等はPCの外部で構成してください。  
PCの故障により、機械の破損や事故のおそれがあります。

### 注意

- 運転中のプログラム変更、強制出力、RUN、STOP等の操作は十分安全を確認して行ってください。  
操作ミスにより、機械の破損や事故のおそれがあります。
- 電源投入順序にしたがって投入してください。  
誤動作により、機械の破損や事故のおそれがあります。

#### 4. 保守について

### 危 険

- 電池の (+) (-) の逆接続、充電、分解、加熱、火中に投入、ショートはしないでください。  
破損、発火のおそれがあります。

### 禁 止

- 分解、改造はしないでください。  
火災、故障、誤動作の原因となります。

### 注 意

- モジュール／ユニットの脱着は電源をOFFしてから行ってください。  
感電、誤動作、故障の原因となることがあります。
- ヒューズは指定品と交換してください。  
火災、故障の原因となります。

## 保証・サービス

特別な保証契約がない場合、この製品の保証は次のとおりです。

### 1. 保証期間と保証範囲

#### 【保証期間】

この製品の保証期間は、ご注文のご指定場所に納入後1年といたします。

#### 【保証範囲】

上記保証期間中に、このマニュアルに従った製品仕様範囲内の正常な使用状態で故障が生じた場合は、その機器の故障部分をお買い上げの販売店または（株）日立エンジニアリング・アンド・サービスにお渡しください。交換または修理を無償で行います。ただし、郵送いただく場合は、郵送料金、梱包費用はご注文主のご負担になります。

次のいずれかに該当する場合は、この保証の対象範囲から除外いたします。

- 製品仕様範囲外の取り扱いおよび使用により故障した場合。
- 納入品以外の事由により故障した場合。
- 納入者以外の改造または修理により故障した場合。
- リレーなどの消耗部品の寿命により故障した場合。
- 上記以外の天災、災害など、納入者側の責任ではない事由により故障した場合。

ここでいう保証とは、納入した製品単体の保証を意味します。したがって、当社ではこの製品の運用および故障を理由とする損失、逸失利益等の請求につきましては、いかなる責任も負いかねますのであらかじめご了承ください。また、この保証は日本国内でのみ有効であり、ご注文主に対して行うものです。

### 2. サービスの範囲

納入した製品の価格には技術者派遣などのサービス費用は含まれておりません。次に該当する場合は別個に費用を申し受けます。

- 取り付け調整指導および試運転立ち会い。
- 保守点検および調整。
- 技術指導、技術教育、およびトレーニングスクール。
- 保証期間後の調査および修理。
- 保証期間中においても、上記保証範囲外の事由による故障原因の調査。

## はじめに

このたびは、CPUオプション外部機器リンクモジュールをご利用いただきありがとうございます。

このオプションマニュアル外部機器リンク（無手順）編は、外部機器リンクモジュールの取扱いを述べたものです。このマニュアルをお読みいただき正しくご使用いただくようお願いいたします。

### 注 意

このマニュアルのオペレーションは、外部機器リンクプログラムフロッピー Ver2.0 Rev2.0以降のものを使用してください。

演算ファンクション使用時は、ラダーまたはコンパクトPMS Ver4.1以降をご使用ください。

また、ラダー演算ファンクション以外のFA-BASIC、C言語、アセンブラでユーザプログラムを作るときは、6.4Kバイト以上の拡張メモリを実装してください。

NESP（Nissan Electronic Sequence Processor）シリーズをご使用のユーザは下記対応表を参照のうえご使用ください。

【HIDIC-S10 $\alpha$ シリーズ】	【NESP-S25シリーズ】
HIDIC-S10/2 $\alpha$ .....	NESP-S25 E
HIDIC-S10/2 $\alpha$ E .....	NESP-2 $\alpha$ E
HIDIC-S10/2 $\alpha$ H .....	NESP-2 $\alpha$ H
HIDIC-S10/2 $\alpha$ Hf .....	NESP-2 $\alpha$ Hf

次の名称を本文では略称で説明しています。

ディスプレイ画面	: 画面
フロッピーディスク	: フロッピー
HIDIC-S10/2 $\alpha$	: CPU
FA-BASIC、C言語、アセンブラプログラム	: Cモード

記号の説明



: 注意事項

: 参考事項

# 目 次

1	ご使用にあたり	1
1.1	拡張CPUユニット	2
1.2	オプションモジュールの実装	2
1.3	アース配線	4
1.4	オプションモジュールの交換	4
2	仕 様	5
2.1	用 途	6
2.2	仕 様	6
2.2.1	システム仕様	6
2.2.2	回線仕様	6
2.3	システムの入れ換えに関する注意事項	10
3	各部の名称と機能、配線	15
3.1	各部の名称と機能	16
3.2	配 線	17
3.2.1	インタフェース信号	17
3.2.2	外部機器接続方法	21
4	利用の手引き	25
4.1	外部機器との接続方法	26
4.1.1	RS-232Cによる接続	26
4.1.2	RS-422による接続	27
4.2	外部機器リンクシステムのソフトウェア構成	28
4.3	外部機器リンクシステムに提供されるシステムプログラム	29
4.3.1	通信制御プログラム	29
4.3.2	外部機器管理プログラム	29
4.3.3	送信ハンドラ	29
4.3.4	受信ハンドラ	30
4.4	ユーザの作成するプログラム	31
4.4.1	ユーザアプリケーションプログラム	31
4.4.2	外部機器側のプログラム	31
4.5	L G B通信制御テーブル	32
4.5.1	伝送フレーム	32
4.5.2	伝送速度	32

4.5.3	優先制御	33
4.5.4	データ変換モード	34
4.5.5	テキスト語数、スタートコード数、スタートコード、エンドコード数、 エンドコード、BCCコード	36
4.5.6	送信中断／再開、中断コード、再開コード、送信中断監視時間	41
4.5.7	送信遅延時間	41
4.5.8	受信監視時間	42
4.5.9	RS-422送信ゲートコントロール	42
4.5.10	送信要求 (Request to Send : RS)	43
4.5.11	データ端末レディ (Equipment Ready : ER)	44
4.5.12	データセットレディ (Data set Ready : DR)	44
4.6	送信バッファ、受信バッファ	45
4.7	送信情報	47
4.8	受信情報	48
4.9	通信制御プログラムの受信完了時起動タスク	49
4.10	ハンドラユーザリンクページ	51
4.10.1	演算ファンクション	52
4.10.2	Cモードサブルーチン	53
4.10.3	ソフトウェアによるハードウェア制御	68
4.11	送信エラーコード表	70
4.12	受信エラーコード表	72
4.13	CPU LED表示メッセージ表	75
5	オペレーション	77
5.1	システムを立上げるにあたり	78
5.1.1	使い方	78
5.1.2	外部機器リンクモジュール	78
5.1.3	システム構成	78
5.1.4	基本オペレーション	79
5.2	システム立上げ	80
5.2.1	外部機器リンクシステム立上げ手順	80
5.2.2	PSEシステム立上げ手順	81
5.3	システムプログラムの書込み (ロード)	83
5.4	システムプログラムの照合 (コンペア)	86
5.5	LGB通信制御テーブル編集	88
5.5.1	設定内容	89
5.5.2	オペレーション	95

5.6	MCS .....	100
5.6.1	機能概要 .....	100
5.6.2	オペレーション .....	101
5.6.3	補足説明および注意事項 .....	105
5.7	UFET .....	105
5.7.1	機能概要 .....	105
5.7.2	ユーザ演算ファンクション .....	105
5.7.3	送受信演算ファンクション回路作成手順 .....	105
5.8	受信起動タスク登録 .....	108
5.8.1	機能概要 .....	108
5.8.2	オペレーション .....	108
6	プログラム例 .....	109
6.1	RS-232Cによるプリンタとの接続例 .....	110
6.1.1	概要 .....	110
6.1.2	システム構成 .....	110
6.1.3	プリンタの印字フォーマット .....	111
6.1.4	プログラム構成 .....	112
6.1.5	ラダープログラムとのリンケージテーブル構成 .....	113
6.1.6	ハードウェア構成 .....	115
6.1.7	LGBの設定 .....	115
6.1.8	Cモードプログラムのフローチャート .....	117
6.1.9	C言語のプログラム例 .....	118
6.1.10	Cモードプログラムの作成と登録 .....	121
6.2	パソコンによるプログラムローディング .....	125
6.2.1	システム構成 .....	125
6.2.2	プログラム構成 .....	125
6.2.3	モトローラ 'S' フォーマット (16ビット用) .....	126
6.2.4	LGBの設定 .....	127
6.2.5	受信タスクの登録 .....	127
6.2.6	受信タスク .....	128
6.2.7	C言語のプログラム例 .....	129
6.2.8	プログラムローディング方法 .....	131
6.3	演算ファンクションによるデータ転送 .....	132
6.3.1	システム構成 .....	132
6.3.2	ハードウェアの設定 .....	133
6.3.3	LGBの設定 .....	134

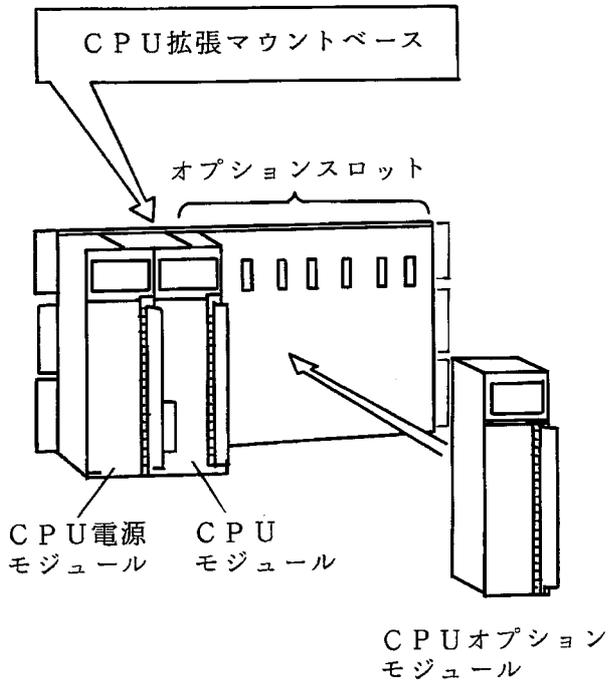
6.3.4	データテーブル構成 .....	136
6.3.5	ラダー回路によるプログラム .....	138
<b>付 録</b> .....		141
付録A	CPUのメモリマップ .....	142
付録B	外部機器リンクモジュールのメモリマップ .....	143
付録C	トレースバッファ .....	144
C.1	トレースバッファの構成 .....	144
C.2	トレースデータの種別 .....	144
C.3	状態フラグ .....	145
C.4	トレースの詳細データ .....	146
付録D	JIS 7単位コード表 (C6220) .....	150
付録E	JIS 8単位コード表 (C6220) .....	151
付録F	制御符号の説明 .....	152
付録G	略号一覧表 .....	153



# 1 ご使用にあたり

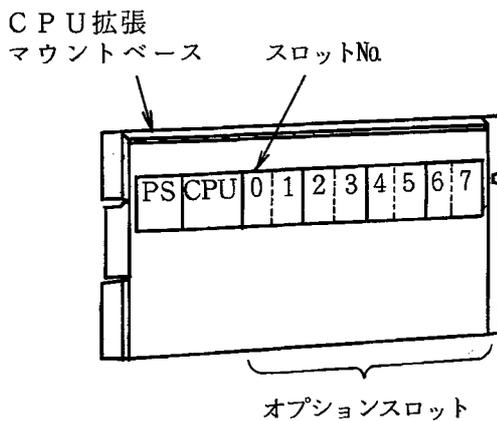
# 1 ご使用にあたり

## 1.1 拡張CPUユニット



- オプションモジュールをご使用いただくためには、CPU拡張マウントベース（形式：HPC-1000）が必要です。
- CPU拡張マウントベースには、オプションモジュール用に8スロットあります。
- 1スロットタイプのモジュールの場合には8モジュール、2スロットタイプのモジュールの場合には、4モジュールを実装できます。

## 1.2 オプションモジュールの実装



CPU拡張マウントベース：HPC-1000

PSスロット：CPU電源（LWV000）を実装。

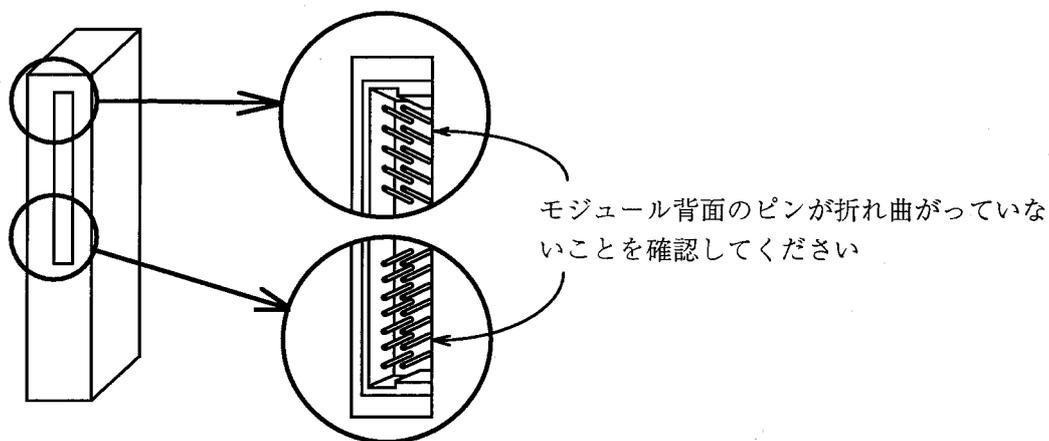
CPUスロット：CPUモジュール（LWP000）を実装。

スロット0～7：オプションモジュールを実装。

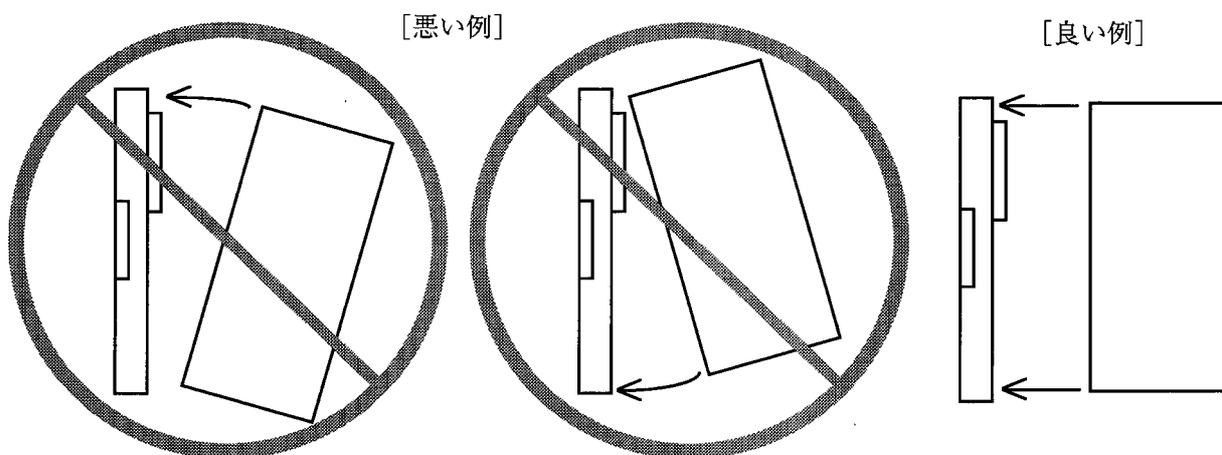
どのスロットも同じで、スロットによる違いはありません。ただし、CPU間リンクモジュールおよびサブCPU間リンクモジュールは、オプションスロットに左詰めに実装してください。

オプションモジュール実装時は、以下のことに注意してください。

- コネクタのピンが曲がっていないことを確認してください。



- マウントベースに対して、正面からまっすぐ実装してください。（悪い例のように斜めに実装すると、ピン曲がりが発生しオプションモジュールが誤動作することがあります。）

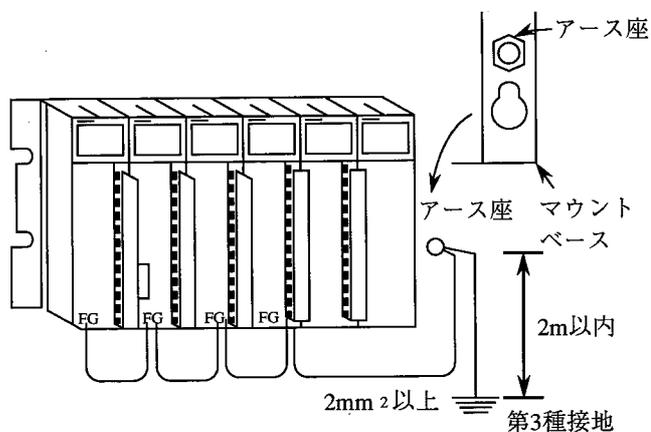


**⚠ 注意**

キャビネットの構造上、頭上にマウントベースが位置する場合、脚立などを使用して、斜めに実装することのないようにしてください。

## 1 ご使用にあたり

### 1.3 アース配線



#### 強制

- FG（フレームグラウンド）のアース配線は、外部端子のある各モジュールのFG端子を、マウントベースのアース座に接続してください。アースの配線距離は2m以内としマウントベースのアース座から第3種接地してください。
- アース線は、線径2mm<sup>2</sup>以上のものを用いてください。

### 1.4 オプションモジュールの交換

モジュールは電源OFF時に交換してください。

電源がON時に交換すると、ハードウェア、ソフトウェアの破壊につながります。

2 仕 様

## 2 仕 様

### 2.1 用 途

外部機器リンクモジュール（形式：LWE046）は、CPUとRS-232CまたはRS-422インタフェースを持つ機器（パソコン、プリンタなど）との間でデータ通信をする場合に用います。

### 2.2 仕 様

#### 2.2.1 システム仕様

項 目		仕 様
メモリ	素 子	CMOS-RAM
	バッテリー	リチウム電池
	バックアップ期間	7年間（25℃）
外部機器リンクモジュール 最大実装台数		2モジュール／CPU
モジュールスロット幅		2スロット幅モジュール
重 量		約1kg

#### 2.2.2 回線仕様

##### (1) ハードウェア仕様

項 目	仕 様						
伝送方式	直列伝送（ビットシリアル伝送）						
通信方式	半二重通信						
同期方式	調歩同期方式						
インタフェース	<table border="1"><thead><tr><th>種 別</th><th>仕 様</th></tr></thead><tbody><tr><td>RS-232C</td><td>EIA RS-232-Cに準拠</td></tr><tr><td>RS-422</td><td>EIA RS-422-Aに準拠</td></tr></tbody></table>	種 別	仕 様	RS-232C	EIA RS-232-Cに準拠	RS-422	EIA RS-422-Aに準拠
	種 別	仕 様					
	RS-232C	EIA RS-232-Cに準拠					
RS-422	EIA RS-422-Aに準拠						
(注) ロータリスイッチにて、どちらか一方を選択し使用します。							
機器接続台数	<table border="1"><thead><tr><th>RS-232C</th><th>RS-422</th></tr></thead><tbody><tr><td>1台／モジュール</td><td>マルチドロップ接続により 最大8台／モジュール</td></tr></tbody></table>	RS-232C	RS-422	1台／モジュール	マルチドロップ接続により 最大8台／モジュール		
	RS-232C	RS-422					
1台／モジュール	マルチドロップ接続により 最大8台／モジュール						



- ・送信信号は、RS-232C/RS-422の選択にかかわらず両方のコネクタから出力します。
- ・受信信号は、RS-232C/RS-422の選択により選択されたコネクタから入力します。

項 目	仕 様																																						
伝送フレーム構成	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">ST</td> <td style="text-align: center;">DATA</td> <td style="text-align: center;">PT</td> <td style="text-align: center;">SP</td> </tr> <tr> <td style="text-align: center;">スタートビット</td> <td style="text-align: center;">データビット</td> <td style="text-align: center;">パリティビット</td> <td style="text-align: center;">ストップビット</td> </tr> </table> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>No.</th> <th>スタート</th> <th>データ長</th> <th>パリティ</th> <th>ストップ</th> </tr> </thead> <tbody> <tr> <td>0</td> <td rowspan="8" style="text-align: center;">1ビット</td> <td rowspan="4" style="text-align: center;">7ビット</td> <td style="text-align: center;">偶数</td> <td rowspan="2" style="text-align: center;">2ビット</td> </tr> <tr> <td>1</td> <td style="text-align: center;">奇数</td> </tr> <tr> <td>2</td> <td style="text-align: center;">偶数</td> <td rowspan="2" style="text-align: center;">1ビット</td> </tr> <tr> <td>3</td> <td style="text-align: center;">奇数</td> </tr> <tr> <td>4</td> <td rowspan="4" style="text-align: center;">8ビット</td> <td style="text-align: center;">なし</td> <td style="text-align: center;">2ビット</td> </tr> <tr> <td>5</td> <td style="text-align: center;">なし</td> <td style="text-align: center;">1ビット</td> </tr> <tr> <td>6</td> <td style="text-align: center;">偶数</td> <td style="text-align: center;">1ビット</td> </tr> <tr> <td>7</td> <td style="text-align: center;">奇数</td> <td style="text-align: center;">1ビット</td> </tr> </tbody> </table>	ST	DATA	PT	SP	スタートビット	データビット	パリティビット	ストップビット	No.	スタート	データ長	パリティ	ストップ	0	1ビット	7ビット	偶数	2ビット	1	奇数	2	偶数	1ビット	3	奇数	4	8ビット	なし	2ビット	5	なし	1ビット	6	偶数	1ビット	7	奇数	1ビット
	ST	DATA	PT	SP																																			
スタートビット	データビット	パリティビット	ストップビット																																				
No.	スタート	データ長	パリティ	ストップ																																			
0	1ビット	7ビット	偶数	2ビット																																			
1			奇数																																				
2			偶数	1ビット																																			
3			奇数																																				
4		8ビット	なし	2ビット																																			
5			なし	1ビット																																			
6			偶数	1ビット																																			
7			奇数	1ビット																																			
伝送速度	150, 300, 600, 1,200, 2,400, 4,800, 9,600 (bps) (※1)																																						
接続ケーブル	<table border="1" style="width: 100%; text-align: center;"> <tr> <td style="width: 10%;"></td> <td style="width: 40%;">RS-232C</td> <td style="width: 40%;">RS-422</td> </tr> <tr> <td>距 離</td> <td>最大 15 m</td> <td>最大 1km</td> </tr> <tr> <td>線 種</td> <td colspan="2">シールド付ツイストペア線</td> </tr> <tr> <td>線 径</td> <td>0.08mm<sup>2</sup> 以上</td> <td>0.5mm<sup>2</sup> 以上</td> </tr> <tr> <td>抵 抗</td> <td>229 Ω /km 以下 (20℃)</td> <td>33.4 Ω /km 以下 (20℃)</td> </tr> <tr> <td>推 薦 品</td> <td>CO-MA-VV-SBAWG28-13P 日立電線製</td> <td>KPEV-SB-8P (0.5mm<sup>2</sup>) 日立電線製</td> </tr> </table>		RS-232C	RS-422	距 離	最大 15 m	最大 1km	線 種	シールド付ツイストペア線		線 径	0.08mm <sup>2</sup> 以上	0.5mm <sup>2</sup> 以上	抵 抗	229 Ω /km 以下 (20℃)	33.4 Ω /km 以下 (20℃)	推 薦 品	CO-MA-VV-SBAWG28-13P 日立電線製	KPEV-SB-8P (0.5mm <sup>2</sup> ) 日立電線製																				
	RS-232C	RS-422																																					
距 離	最大 15 m	最大 1km																																					
線 種	シールド付ツイストペア線																																						
線 径	0.08mm <sup>2</sup> 以上	0.5mm <sup>2</sup> 以上																																					
抵 抗	229 Ω /km 以下 (20℃)	33.4 Ω /km 以下 (20℃)																																					
推 薦 品	CO-MA-VV-SBAWG28-13P 日立電線製	KPEV-SB-8P (0.5mm <sup>2</sup> ) 日立電線製																																					
接続コネクタ	<table border="1" style="width: 100%; text-align: center;"> <tr> <td style="width: 10%;"></td> <td style="width: 40%;">RS-232C</td> <td style="width: 40%;">RS-422</td> </tr> <tr> <td>種 別</td> <td>25ピン D-SUB コネクタ</td> <td>40点 端子台コネクタ</td> </tr> <tr> <td>備 考</td> <td>コネクタ：DB-25P (日本航空電子製) カバー：HDB-CTH1 (ヒロセ電機製)</td> <td>ネジ径：3mm</td> </tr> </table>		RS-232C	RS-422	種 別	25ピン D-SUB コネクタ	40点 端子台コネクタ	備 考	コネクタ：DB-25P (日本航空電子製) カバー：HDB-CTH1 (ヒロセ電機製)	ネジ径：3mm																													
	RS-232C	RS-422																																					
種 別	25ピン D-SUB コネクタ	40点 端子台コネクタ																																					
備 考	コネクタ：DB-25P (日本航空電子製) カバー：HDB-CTH1 (ヒロセ電機製)	ネジ径：3mm																																					



(※1) 伝送速度は 1200～4800 (bps) の範囲内で使用することを推奨します。

## 2 仕 様

### (2) ソフトウェア仕様

項 目	仕 様																																																				
伝送制御手順	無手順																																																				
優先制御	<table border="1"> <thead> <tr> <th>No.</th> <th>優先制御内容</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>自局優先 (送信中の受信要求は受付けない。)</td> </tr> <tr> <td>1</td> <td>他局優先 (送信注でも受信要求を受付ける。)</td> </tr> </tbody> </table>	No.	優先制御内容	0	自局優先 (送信中の受信要求は受付けない。)	1	他局優先 (送信注でも受信要求を受付ける。)																																														
No.	優先制御内容																																																				
0	自局優先 (送信中の受信要求は受付けない。)																																																				
1	他局優先 (送信注でも受信要求を受付ける。)																																																				
データ変換モード	<table border="1"> <thead> <tr> <th>No.</th> <th>データ変換方式</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>テキストデータをそのまま送受信する。</td> </tr> <tr> <td>1</td> <td>テキストデータを ASCII 変換して送信する。 受信したテキストデータを ASCII から BINARY へ変換する。</td> </tr> </tbody> </table>	No.	データ変換方式	0	テキストデータをそのまま送受信する。	1	テキストデータを ASCII 変換して送信する。 受信したテキストデータを ASCII から BINARY へ変換する。																																														
No.	データ変換方式																																																				
0	テキストデータをそのまま送受信する。																																																				
1	テキストデータを ASCII 変換して送信する。 受信したテキストデータを ASCII から BINARY へ変換する。																																																				
伝送ブロック構成	<table border="1"> <thead> <tr> <th>SCD</th> <th>TEXT</th> <th>ECD</th> <th>BCC</th> </tr> </thead> <tbody> <tr> <td>スタートコード</td> <td>テキスト</td> <td>エンドコード</td> <td>ブロックチェック キャラクタ</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>No.</th> <th>スタートコード</th> <th>No.</th> <th>テキスト</th> <th>No.</th> <th>エンドコード</th> <th>No.</th> <th>BCC</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>なし</td> <td>0</td> <td>なし</td> <td>0</td> <td>なし</td> <td>0</td> <td>なし</td> </tr> <tr> <td>1</td> <td>1 キャラクタ</td> <td>1</td> <td>1 バイト</td> <td>1</td> <td>1 キャラクタ</td> <td rowspan="2">1</td> <td>水平偶数 パリティ</td> </tr> <tr> <td>2</td> <td>2 キャラクタ</td> <td>:</td> <td>:</td> <td>2</td> <td>2 キャラクタ</td> </tr> <tr> <td>3</td> <td>3 キャラクタ</td> <td>511</td> <td>511 バイト</td> <td>3</td> <td>3 キャラクタ</td> <td rowspan="2">2</td> <td>水平奇数 パリティ</td> </tr> <tr> <td>4</td> <td>4 キャラクタ</td> <td>512</td> <td>512 バイト</td> <td>4</td> <td>4 キャラクタ</td> </tr> </tbody> </table>	SCD	TEXT	ECD	BCC	スタートコード	テキスト	エンドコード	ブロックチェック キャラクタ	No.	スタートコード	No.	テキスト	No.	エンドコード	No.	BCC	0	なし	0	なし	0	なし	0	なし	1	1 キャラクタ	1	1 バイト	1	1 キャラクタ	1	水平偶数 パリティ	2	2 キャラクタ	:	:	2	2 キャラクタ	3	3 キャラクタ	511	511 バイト	3	3 キャラクタ	2	水平奇数 パリティ	4	4 キャラクタ	512	512 バイト	4	4 キャラクタ
SCD	TEXT	ECD	BCC																																																		
スタートコード	テキスト	エンドコード	ブロックチェック キャラクタ																																																		
No.	スタートコード	No.	テキスト	No.	エンドコード	No.	BCC																																														
0	なし	0	なし	0	なし	0	なし																																														
1	1 キャラクタ	1	1 バイト	1	1 キャラクタ	1	水平偶数 パリティ																																														
2	2 キャラクタ	:	:	2	2 キャラクタ																																																
3	3 キャラクタ	511	511 バイト	3	3 キャラクタ	2	水平奇数 パリティ																																														
4	4 キャラクタ	512	512 バイト	4	4 キャラクタ																																																
送信遅延時間	<p style="text-align: center;"><math>T_0=0\sim 32,767</math>ミリ秒 (1ミリ秒単位)</p>																																																				
送信中断・再開コード	<table border="1"> <thead> <tr> <th>No.</th> <th>中断コード</th> <th>再開コード</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>なし</td> <td>なし</td> </tr> <tr> <td>1</td> <td rowspan="2">1 キャラクタ</td> <td>1 キャラクタ</td> </tr> <tr> <td>2</td> <td>2 キャラクタ</td> </tr> <tr> <td>3</td> <td rowspan="2">2 キャラクタ</td> <td>1 キャラクタ</td> </tr> <tr> <td>4</td> <td>2 キャラクタ</td> </tr> </tbody> </table>	No.	中断コード	再開コード	0	なし	なし	1	1 キャラクタ	1 キャラクタ	2	2 キャラクタ	3	2 キャラクタ	1 キャラクタ	4	2 キャラクタ																																				
No.	中断コード	再開コード																																																			
0	なし	なし																																																			
1	1 キャラクタ	1 キャラクタ																																																			
2		2 キャラクタ																																																			
3	2 キャラクタ	1 キャラクタ																																																			
4		2 キャラクタ																																																			

項 目	仕 様											
送信中断監視時間	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">S C D</td> <td style="text-align: center;">T E X T 1</td> <td style="text-align: center;">T E X T 2</td> <td style="text-align: center;">—</td> <td style="text-align: center;">中 断 コ ー ド</td> <td style="text-align: center;">—</td> <td style="text-align: center;">再 開 コ ー ド</td> <td style="text-align: center;">—</td> <td style="text-align: center;">T E X T n</td> <td style="text-align: center;">E C D</td> <td style="text-align: center;">B C C</td> </tr> </table> <p style="text-align: center;"> <math>T_0</math>  <math>T_0=0\sim 3,276.7</math>秒 (100ミリ秒単位)         </p>	S C D	T E X T 1	T E X T 2	—	中 断 コ ー ド	—	再 開 コ ー ド	—	T E X T n	E C D	B C C
S C D	T E X T 1	T E X T 2	—	中 断 コ ー ド	—	再 開 コ ー ド	—	T E X T n	E C D	B C C		
受信監視時間	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">S C D</td> <td style="text-align: center;">T E X T 1</td> <td style="text-align: center;">T E X T 2</td> <td style="text-align: center;">—</td> <td style="text-align: center;">—</td> <td style="text-align: center;">T E X T n</td> <td style="text-align: center;">E C D</td> <td style="text-align: center;">B C C</td> </tr> </table> <p style="text-align: center;"> <math>T_0</math>  <math>T_0=0\sim 3,276.7</math>秒 (100ミリ秒単位)         </p>	S C D	T E X T 1	T E X T 2	—	—	T E X T n	E C D	B C C			
S C D	T E X T 1	T E X T 2	—	—	T E X T n	E C D	B C C					
RS-422 ゲートコントロール	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">No.</td> <td style="text-align: center;">ゲートコントロール内容</td> </tr> <tr> <td style="text-align: center;">0</td> <td>送信ゲートオープン</td> </tr> <tr> <td style="text-align: center;">1</td> <td>送信ゲート自動コントロール (注1)</td> </tr> </table>	No.	ゲートコントロール内容	0	送信ゲートオープン	1	送信ゲート自動コントロール (注1)					
No.	ゲートコントロール内容											
0	送信ゲートオープン											
1	送信ゲート自動コントロール (注1)											
送信要求 (RS) 出力	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">No.</td> <td style="text-align: center;">送信要求出力内容</td> </tr> <tr> <td style="text-align: center;">0</td> <td>送信要求出力 (注2)</td> </tr> <tr> <td style="text-align: center;">1</td> <td>送信要求出力しない。</td> </tr> </table>	No.	送信要求出力内容	0	送信要求出力 (注2)	1	送信要求出力しない。					
No.	送信要求出力内容											
0	送信要求出力 (注2)											
1	送信要求出力しない。											
データ端末レディ (ER) 出力	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">No.</td> <td style="text-align: center;">データ端末レディ出力内容</td> </tr> <tr> <td style="text-align: center;">0</td> <td>ノットレディ出力</td> </tr> <tr> <td style="text-align: center;">1</td> <td>レディ出力 (注3)</td> </tr> </table>	No.	データ端末レディ出力内容	0	ノットレディ出力	1	レディ出力 (注3)					
No.	データ端末レディ出力内容											
0	ノットレディ出力											
1	レディ出力 (注3)											
データセットレディ (DR) 出力	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">No.</td> <td style="text-align: center;">データセットレディチェック内容</td> </tr> <tr> <td style="text-align: center;">0</td> <td>チェックなし</td> </tr> <tr> <td style="text-align: center;">1</td> <td>チェックあり</td> </tr> </table>	No.	データセットレディチェック内容	0	チェックなし	1	チェックあり					
No.	データセットレディチェック内容											
0	チェックなし											
1	チェックあり											

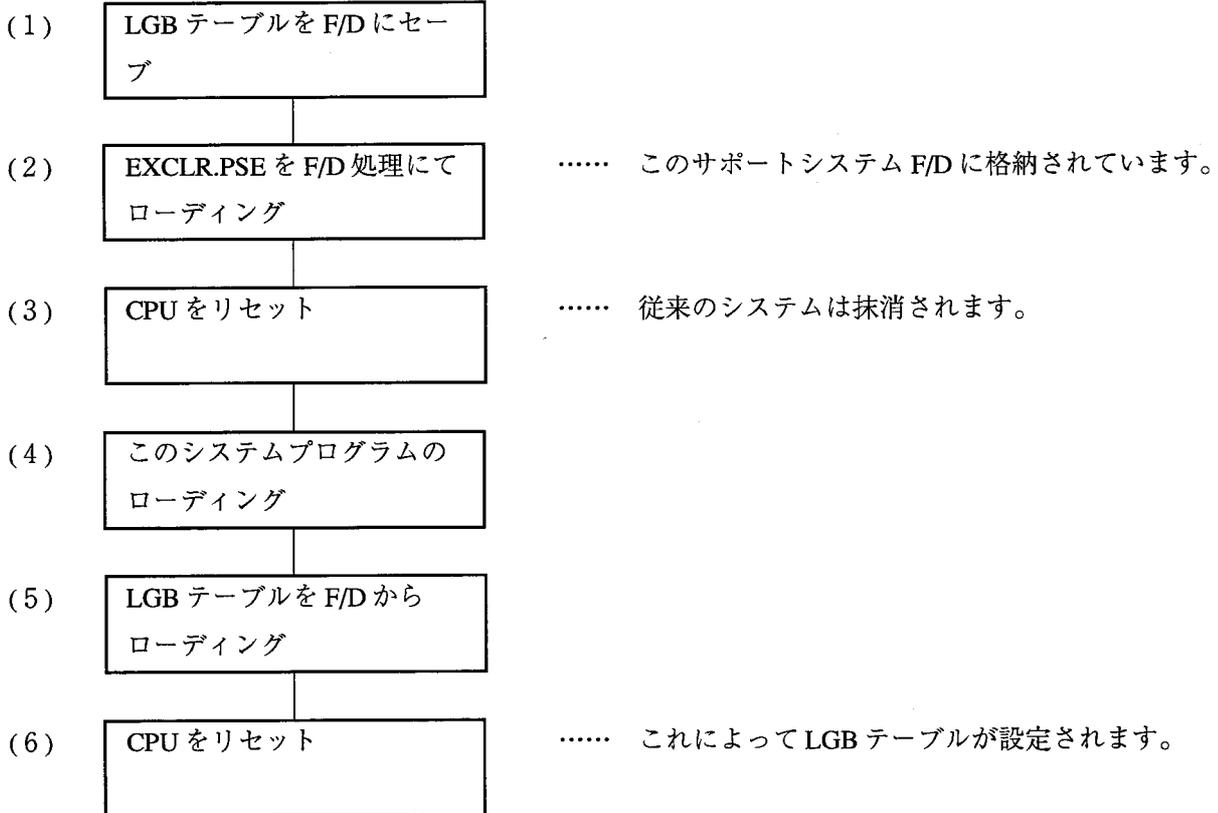


- ・送信ゲートはデータ送信時に自動的にオープンします。
- ・RS 端子は '0' (ON) 固定となります。
- ・ER 端子は '0' (ON) 固定となります。

### 2.3 システムの入れ換えに関する注意事項

- Ver1.2 Rev6.0以前の外部機器リンクのソフトウェアと混在して使用しないでください。混在して使用すると、CPUが停止しますので、絶対に混在しないでください。
- Ver1.2 Rev6.0以前のシステムを改造/増築などで、このシステムに入れ換える場合は、下記手順で行ってください。

[タスクシステム時]



[演算ファンクションシステム時]

- |     |                                |  |
|-----|--------------------------------|--|
| (1) | LGB テーブルを F/D に<br>セーブ         |  |
| (2) | EXCLR.PSE を F/D 処理にて<br>ローディング | ..... このサポートシステム F/D に格納されています。  |
| (3) | CPU をリセット                      | ..... 従来の外部機器リンクシステムは抹消されます。   |
| (4) | このシステムプログラムの<br>ローディング         |  |
| (5) | LGB テーブルを F/D から<br>ローディング     |  |
| (6) | CPU をリセット                      | ..... これによって LGB テーブルが設定されます。  |
| (7) | UFET の登録                       | ..... このサポートシステムにて登録します。   |
| (8) | ラダー回路の修正                       | ..... <div style="border: 1px solid black; padding: 5px; display: inline-block;">PSE <math>\alpha</math> システム Ver4.1 以降のシステムにて<br/>行います。</div> |

SDM → SD0	..... 左記ユーザ演算ファンクションを書き換えます。
RVM → RV0	
SDS → SD1	
RVS → RV1	

## 2 仕 様

### LGB テーブルセーブ手順

外部機器リンクシステムソフトウェア入れ換え時、LGB テーブル再設定の手間と設定ミスを削減するため、LGB テーブルを F/D に保存します。

```
FLOPPY/RAM DISK
KEYIN NO.=          [CLS]
-----
                FLOPPY MENU
-----
1 : DIRECTORY
2 : PCS → FLOPPY
3 : FLOPPY → PCS
4 : FILE ERASE
5 : COMPARE PCS <> FLOPPY
6 : FORMATTING
7 : INITIALIZE
8 : FILE COPY
9 : DRIVE SELECT
-----
```

ラダーかコンパクト PMS を立上げ、FLOPPY MENU 画面にしてください。

(このとき F/D ドライブにワーク F/D を挿入してください。)

```
PCS → FLOPPY
KEYIN NEXT DATA    [CLS]
-----
                PSE FILE HEADER
-----
FILE NAME :
PCS NO.   : 0000
PCS TYPE  : 00F2
Y-M-D-H   : - - - -
COMMENT   :
-----
```

“2” を選択してください。

```
PCS → FLOPPY
HEADER OK? [SET/CNT/RTY/CLS]
-----
                PSE FILE HEADER
-----
FILE NAME : LGB .PSE
PCS NO.   : 9999
PCS TYPE  : 00F2
Y-M-D-H   : 91-12-12-13
COMMENT   : No.0,1 LGB
FILE SIZE : 0036( K-WORD )

ADDRESS = /060000 - /071FFE
-----
```

FILE NAME を “LGB.PSE” PCS NO. : 9999 とし、Y-M-D-H (日付) と COMMENT (コメント) を入力後、

**設定** キーを押してください。

PCS → FLOPPY  
 HEADER OK? [SET/CNT/RTY/CLS]

-----  
 PSE FILE HEADER  
 -----

FILE NAME : LGB .PSE  
 PCS NO. : 9999  
 PCS TYPE : 00F2  
 Y-M-D-H : 91-12-12-13  
 COMMENT : No.0,1 LGB  
 FILE SIZE : 0004( K-WORD )

ADDRESS = /F48100 - /F481FE  
 ADDRESS = /F58100 - /F581FE  
 -----

ここで **続行**、**設定** キーを押し、下表の各チャンネルのLGBアドレスを入力します。

(ここでは、チャンネルNo.0,1)

チャネルNo.	アドレス
0	/F48100~/F481FE
1	/F58100~/F581FE

PCS → FLOPPY  
 SUCCESS [CNT/CCS] ADDRESS=/F581FE

-----  
 PSE FILE HEADER  
 -----

FILE NAME : LGB .PSE  
 PCS NO. : 9999  
 Y-M-D-H : 91-12-12-13  
 COMMENT : No.0,1 LGB  
 FILE SIZE : 0004( K-WORD )

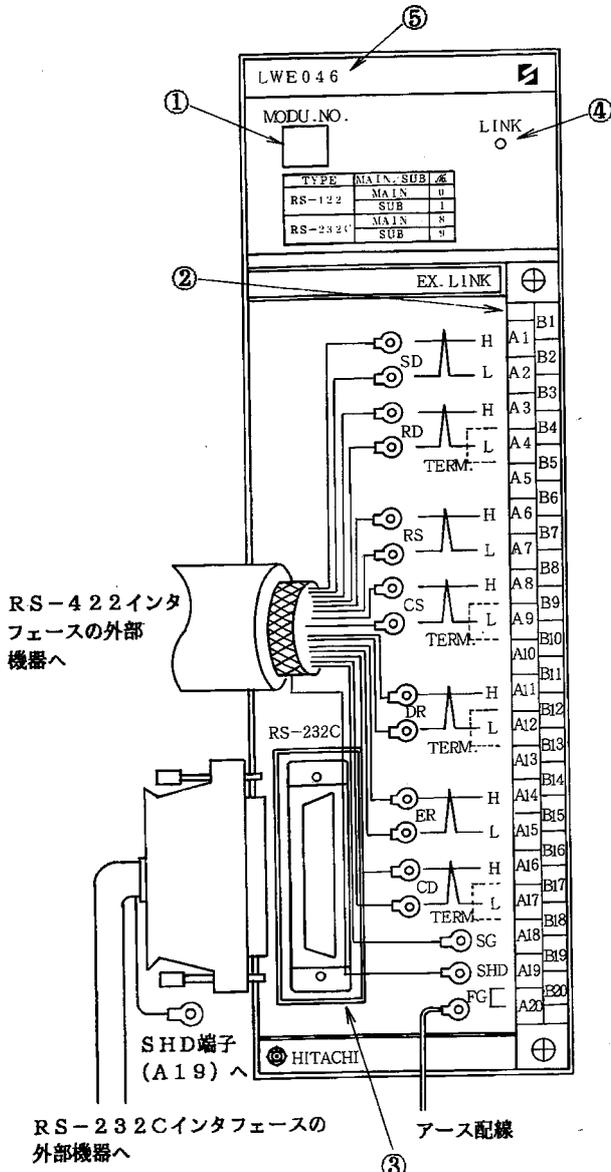
ADDRESS = /F48100 - /F481FE  
 ADDRESS = /F58100 - /F581FE  
 -----

**設定** キーを押すと、F/D に保存されます。



### 3 各部の名称と機能、配線

3.1 各部の名称と機能



① モジュール No. 設定スイッチ

1台のCPUに外部機器リンクモジュールを2枚まで実装できます。インタフェースは、RS-422とRS-232Cを選択できます。

モジュール No. 設定スイッチ下部の対応表にて、モジュール No. を選択し、設定してください。

スイッチの設定

No.	MAIN/SUB	インタフェース
0	MAIN (チャネル0)	RS-422
1	SUB (チャネル1)	
8	MAIN (チャネル0)	RS-232C
9	SUB (チャネル1)	

② RS-422 インタフェース用端子台

- SD : 送信データ線を接続します。
  - RD : 受信データ線を接続します。
  - RS : 送信要求線を接続します。
  - CS : 送信可線を接続します。
  - DR : データセットレディ線を接続します。
  - ER : データ端末レディ線を接続します。
  - CD : データチャネル受信キャリア検出線を接続します。
  - SG : 外部機器のシグナルグランドと接続します。
  - SHD : ケーブルのシールド線を接続します。  
SHDとFGは内部で接続されています。
  - FG : アース配線を接続します。
- 「1.3 アース配線」を参照してください。

TERM. : RS-422 インタフェースでは、回線の受信信号 (RD, CS, DR, CD) を終端対抗 (100 Ω) で終端してください。  
この端子を短絡すると、内部で終端抵抗 (100 Ω) が接続されます。

③ RS-232C 用コネクタ

RS-232C インタフェースの外部機器と接続します。

④ LINK LED

外部機器へデータ送信時、点滅します。

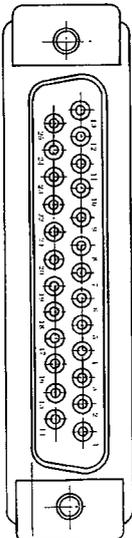
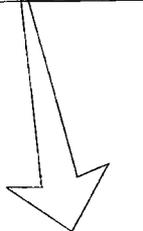
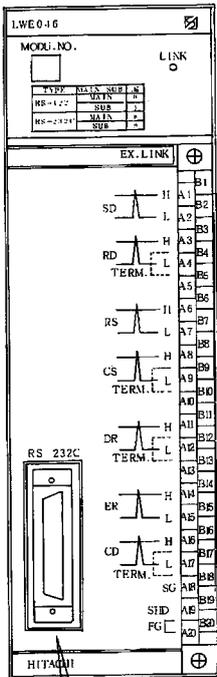
⑤ モジュール形式

LWE046

3.2 配線

3.2.1 インタフェース信号

(1) RS-232Cインタフェース信号



ピン 番号	信号名		入出力 区分	信号の意味	開放時の状態
	略称	名称			
1		(未使用)			
2	SD	送信データ (Send Data)	出力	1: マーク 0: スペース	——
3	RD	受信データ (Receive Data)	入力	1: マーク 0: スペース	マーク
4	RS	送信要求 (Request to Send)	出力	1: 要求なし 0: 要求あり	——
5	CS	送信可 (Clear to Send)	入力	1: 送信できない。 0: 送信できる。	送信できない。
6	DR	データセットレディ (Data set Ready)	入力	1: ノットレディ 0: レディ	ノットレディ
7	SG	信号用接地 (Signal Ground)	——	——	——
8	CD	データチャネル受信キャリア ア検出 (data Carrier Detect)	入力	1: キャリアなし 0: キャリアあり	キャリアなし
9	P5	+5 V出力 (最大 300mA)	出力	——	——
10		(未使用)			
11		(未使用)			
12	P5	+5 V出力 (最大 300mA)	出力	——	——
13		(未使用)			
14		(未使用)			
15		(未使用)			
16		(未使用)			
17		(未使用)			
18		(未使用)			
19		(未使用)			
20	ER	データ端末レディ (Equipment Ready)	出力	1: ノットレディ 0: レディ	——
21		(未使用)			
22		(未使用)			
23		(未使用)			
24		(未使用)			
25		(未使用)			

### 3 各部の名称と機能、配線

#### ● RS-232Cの信号内容

- [SD] 外部機器リンクから端末へのデータ線です。RS, CS, DR, ERの4つの制御線がオンのときに有効です。
- [RD] 端末から外部機器リンクへのデータ線です。CDがオンのとき有効です。  
データを伝送していない間（CDがオフの間）は、マーク状態にしておきます。
- [RS] 外部機器リンクから端末へ出力するデータがあることを表す制御先です。RSがオンの間は、端末は外部機器リンクからのデータの入力状態を続けます。RSをいったんオフしたならば、CSがオフになるまで再びオンにはできません。
- [CS] 端末が通信回路へのデータ送信が可能であることを表す制御線です。CSがオンのときは、端末は外部機器リンクからのデータを入力できます。
- [DR] 端末が動作できることを表す制御線です。具体的には、端末が回線と接続され、外部機器リンクとの制御信号のやりとりができる状態を表します。
- [SG] 信号用のアースです。すべての信号の基準電圧（0V）になります。
- [CD] 外部機器リンクが通信回線から有効な信号を受信していることを表す制御線です。CDがオンのときにRDは有効です。外部機器リンクは、CDがオンになったら、端末からデータを入力します。CDがオフの間は、RDをマーク状態にしておきます。
- [ER] 外部機器リンクが端末に対して、データの入出力ができることを表す制御線です。外部機器リンクは、ERがオンになったら回線と接続し、オフになると回線を切り離します。
- [SHD] ケーブルのシールド用アースです。SHDはコネクタ内になく端子台A19を使用します。
- [FG] 保安用のアースです。FGもコネクタ内になく端子台A20, B20を使用します。
- [P5] 将来拡張用のDC5V出力です。

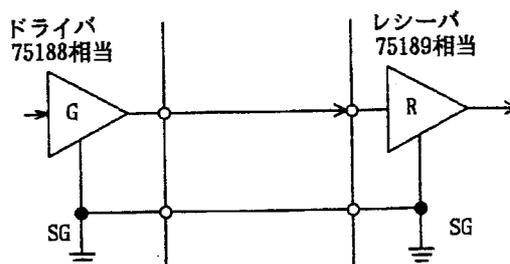
#### ● RS-232Cの電圧レベル

呼び名	マーク	スペース
解釈	1/オフ	0/オン
出力条件	-5V ~ -25V	+5V ~ +25V
入力条件	≤ -3V	≥ +3V

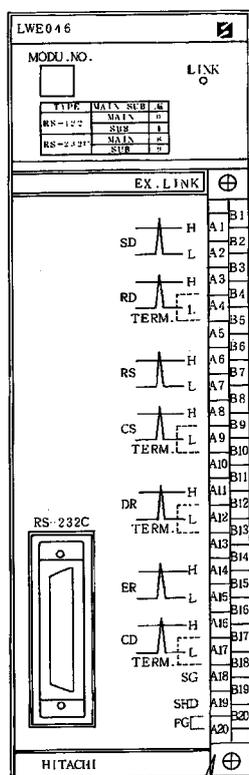
出力条件は、マークまたはスペースを送信するときの電圧値で、±12Vをよく使用します。

#### ● RS-232Cの基本回路

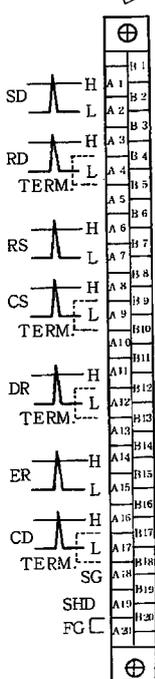
- G: ドライバ
- R: レシーバ



(2) RS-422インタフェース信号



ピン番号	信号名		入出力区分	信号の意味	開放時の状態
	略称	名称			
A1	SD	送信データ (Send Data)	出力	1:マーク 0:スペース	——
A2					
A3	RD	受信データ (Receive Data)	入力	1:マーク 0:スペース	マーク
A4					
A6	RS	送信要求 (Request to Send)	出力	1:要求なし 0:要求あり	——
A7					
A8	CS	送信可 (Clear to Send)	入力	1:送信できない。 0:送信できる。	送信できない。
A9					
A11	DR	データセットレディ (Data set Ready)	入力	1:ノットレディ 0:レディ	ノットレディ
A12					
A14	ER	データ端末レディ (Equipment Ready)	出力	1:ノットレディ 0:レディ	——
A15					
A16	CD	データチャネル受信キャリア検出 (data Carrier Detect)	入力	1:キャリアなし 0:キャリアあり	キャリアあり
A17					
A18	SG	信号用接地 (Signal Ground)	——	——	——
A19	SHD	シールド用接地 (SHield ground)	——	——	——
A20	FG	保守用接地 (Frame Ground)	——	——	——
B4	TERM.	受信データ用終端抵抗 (Terminal Register)	——	——	——
B5					
B9	TERM.	送信可応答用終端抵抗 (Terminal Register)	——	——	——
B10					
B12	TERM.	データセットレディ用終端抵抗 (Terminal Register)	——	——	——
B13					
B17	TERM.	受信キャリア検出用終端抵抗 (Terminal Register)	——	——	——
B18					
B20	FG	保守用接地 (Frame Ground)	——	——	——



- ・特記なき端子は未使用です。
- ・ピン番号 A19, A20, B20 はモジュール内部でつながっていますがマウントベースとは接続されていません。アース配線する場合は「1.3 アース配線」を参照してください。

### 3 各部の名称と機能、配線

- RS-422の信号内容

RS-422の信号内容は、RS-232Cに準じています。

- RS-422の終端抵抗

RS-422インタフェースは、回線の受信信号（RD, CS, DR, CD）を $100\Omega$ で終端してください。

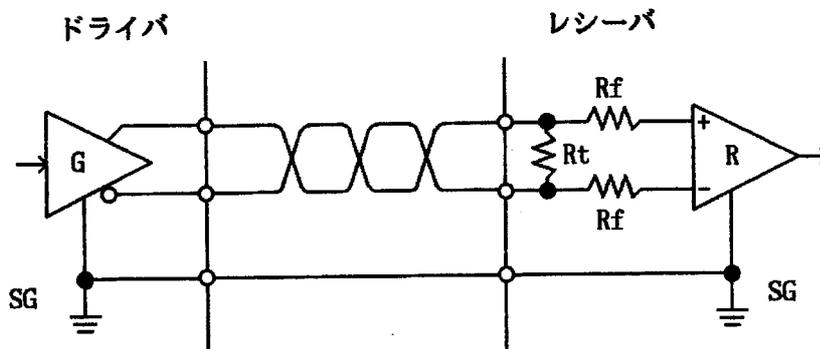
〔TERM.〕端子を短絡することにより、内部で終端抵抗 $100\Omega$ が接続されます。

- RS-422の電圧レベル

呼び名	マーク	スペース
解釈	1 / オフ	0 / オン
出力条件	-3 V ~ -6 V	+3 V ~ +6 V

出力条件は、マークまたはスペースを送信するときの電圧値で、 $\pm 5V$ をよく使用します。

- RS-422の基本回路



G : ドライバ AM26LS31 (AMD社製) 相当

R : レシーバ AM26LS32 (AMD社製) 相当

Rt : ケーブルの終端抵抗 $100\Omega$ ,  $\pm 10\%$ , 1/2W

Rf : 入力補償用抵抗 $4.7k\Omega$ ,  $\pm 10\%$ , 1/4W

3.2.2 外部機器接続方法

(1) 論理的接続例

No.	接続	名称	接続方法 (論理的接続)			外部機器例
			外部機器リンク	ケーブル	外部機器	
1		Full Modem Support (標準タイプ) ( CDによる受信管理と DRによる送信管理を しながらデータを送 受信します。 )	RS CS CD SD RD ER DR SG		RS CS CD SD RD ER DR SG	パーソナル コンピュータ
2	データ レク ト	CDによる受信管理 ( 外部機器からの送信 要求 (RS) で、外部 機器リンクを受信で きる状態とします。 )	RS CS CD SD RD ER DR SG		RS CS CD SD RD ER DR SG	
3	接続	DRによる送信管理 ( 外部機器からのデー タ端末レディ (ER) で外部機器リンクか ら送信データを送信 します。 )	RS CS CD SD RD ER DR SG		RS CS CD SD RD ER DR SG	キャラクタ ディスプレイ (HC-5111-11)
4		データのみ ( 外部機器、外部機器 リンクの状態をチェ ックせず、データを送 受信します。 )	RS CS CD SD RD ER DR SG		RS CS CD SD RD ER DR SG	プリンタ
5	モデム 接続	モデムとの接続	RS CS CD SD RD ER DR SG		RS CS CD SD RD ER DR SG	

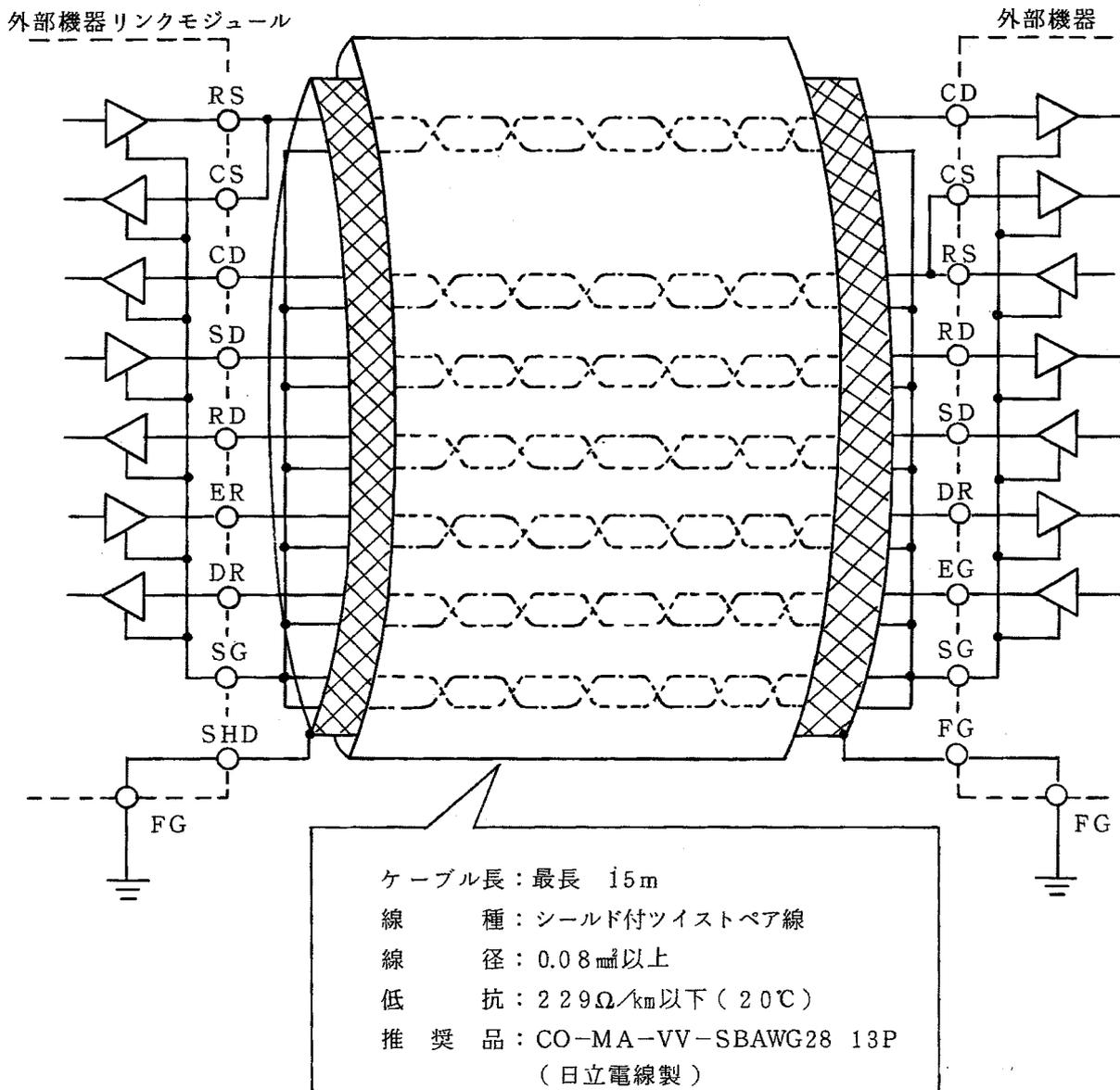


・上表中 (P) は常に送信要求 (RS) を要求ありもしくは、データ端末レディ (ER) をレディ状態にしておくことを意味します。

### 3 各部の名称と機能、配線

#### (2) RS-232Cインタフェースの接続方法

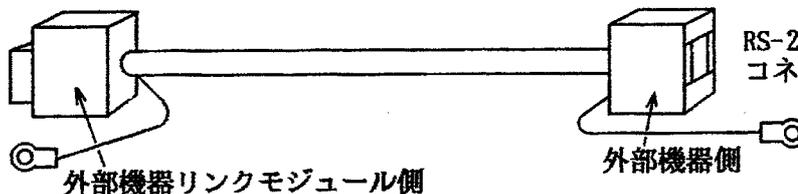
標準タイプ「3.2.2(1)論理的接続例」のNo.1の場合を例にとり、接続方法を下記に示します。



- ・お互いの信号用接地 (SG) は、必ずインタフェースケーブルにて接続してください。  
 インタフェースケーブルのシールド線は必ず両端にて保守用接地 (FG) に接続してください。
- ・外部機器リンクモジュールのシールド用接地 (SHD) 端子は、RS-422 インタフェース用端子台のシールド用接地 (SHD) 端子を使用してください。

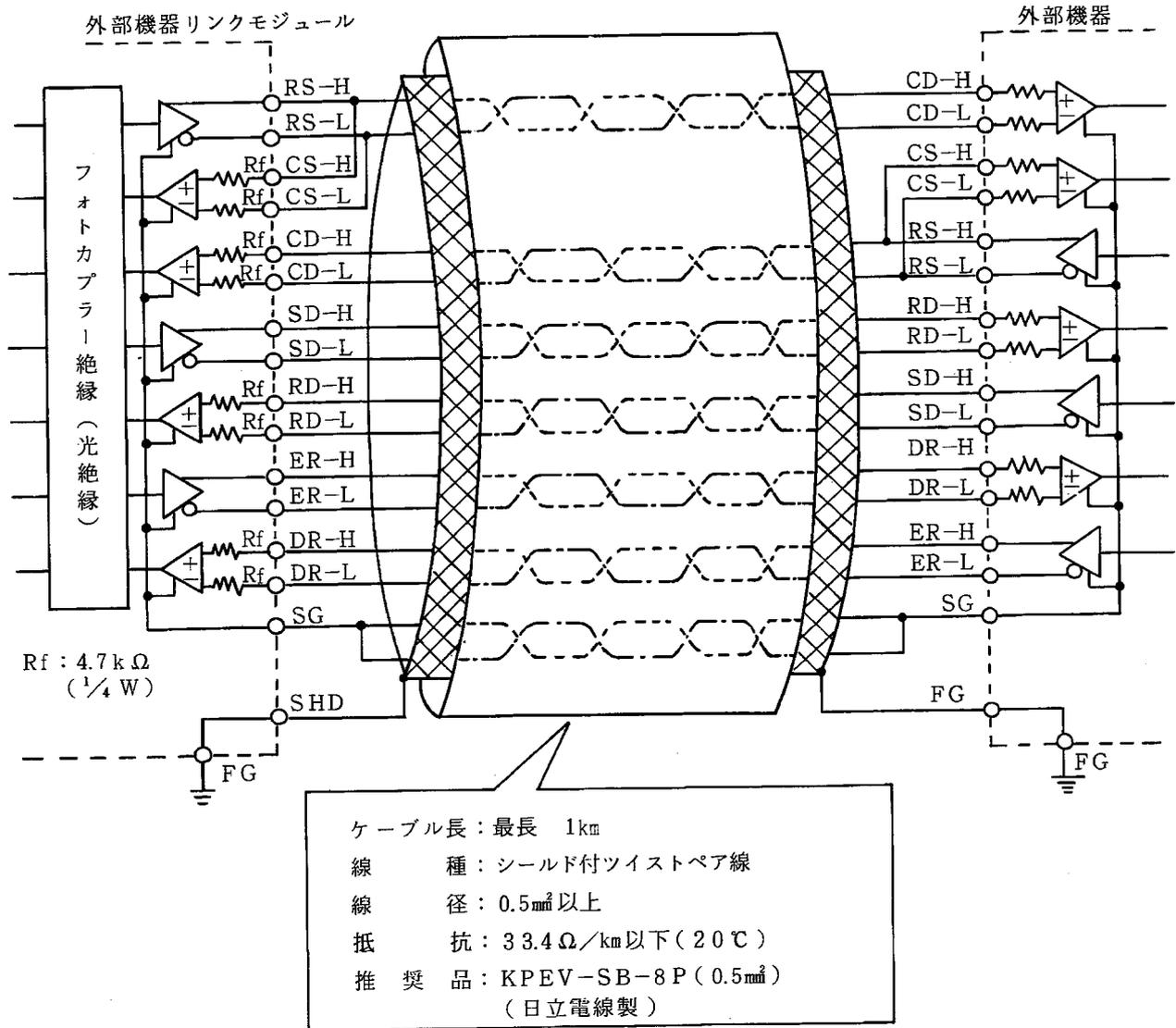
RS-232C用  
コネクタへ

RS-422用  
SHD端子へ



(3) RS-422インタフェースの接続方法

標準タイプ「3.2.2(1)論理的接続例」のNo.1の場合を例にとり、接続方法を下記に示します。



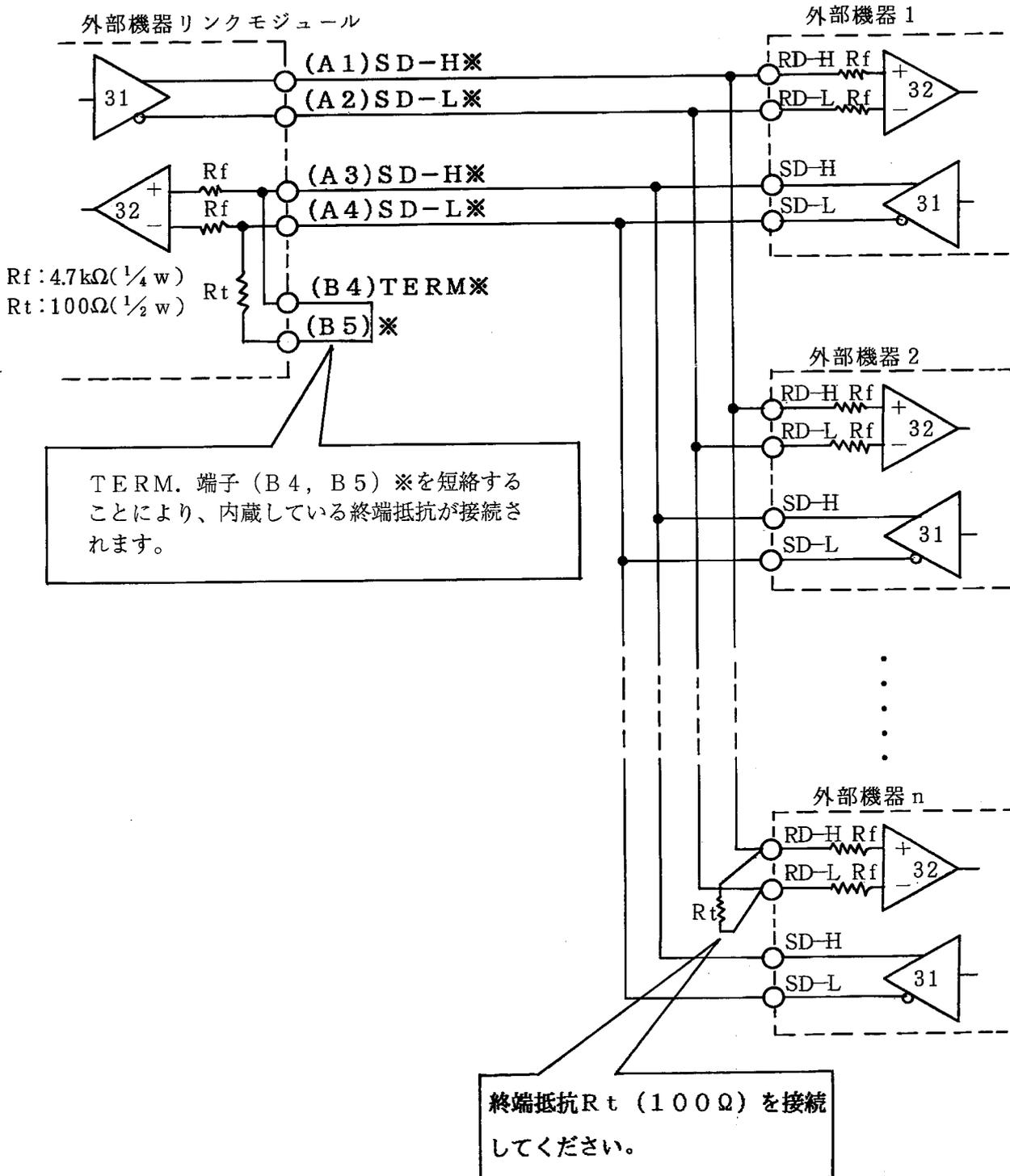
- ・お互いの信号用接地 (SG) は、必ずインタフェースケーブルにて接続してください。
- ・インタフェースケーブルのシールド線は、必ず両端にて保守用接地 (FG) に接続してください。

### 3 各部の名称と機能、配線

#### (4) 終端抵抗の接続方法

RS-422インタフェースで外部機器と接続する場合、終端機器（外部機器リンクモジュールも含む）の受信部には、終端抵抗 [100Ω] が必要となります。

終端抵抗の接続方法を送信データ (SD)、受信データ (RD) 信号を例にとり以下に示します。



※ ( ) 内はRS-422インタフェース用の端子番号を表します。

## 4 利用の手引き

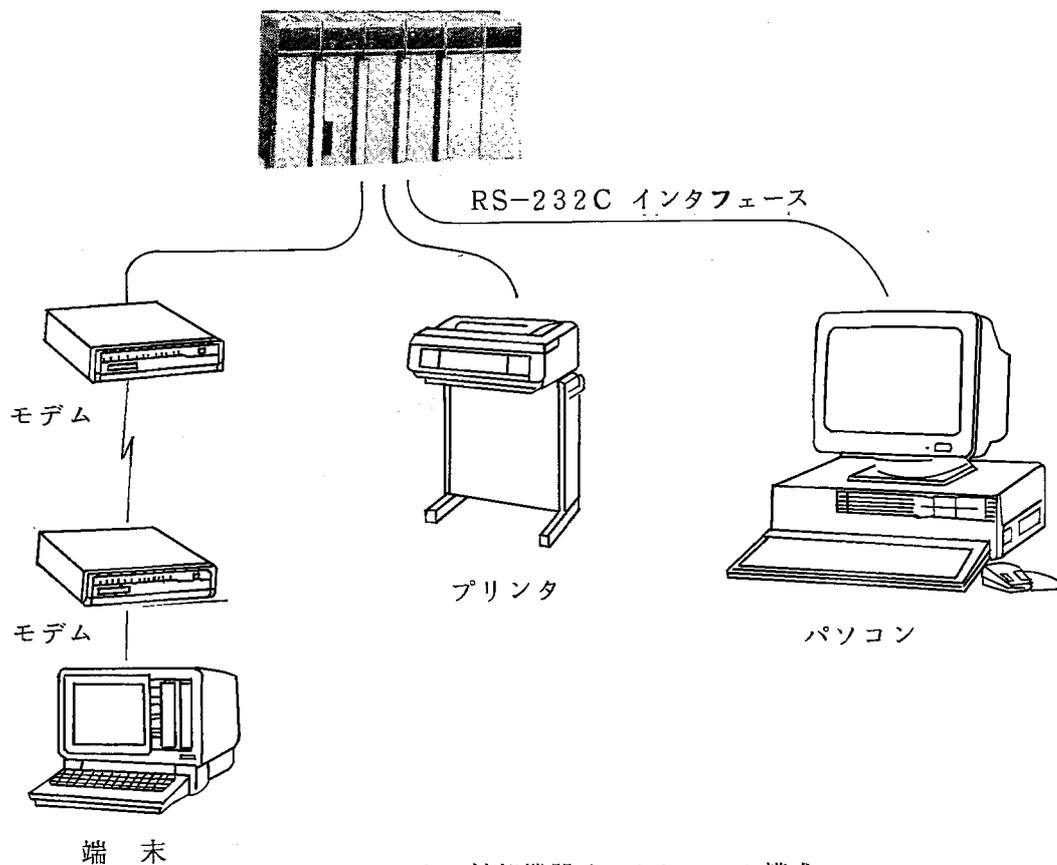
## 4 利用の手引き

この章では外部機器リンクモジュールを使用して外部機器（パソコン、プリンタなど）を接続する方法について説明します。

### 4.1 外部機器との接続方法

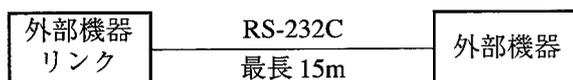
#### 4.1.1 RS-232Cによる接続

RS-232Cインタフェースによるパソコン、プリンタなどとの接続は、次のようになります。

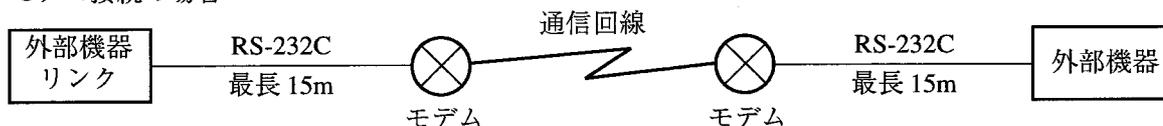


RS-232Cによる外部機器リンクシステム構成

● 直接接続の場合



● モデム接続の場合

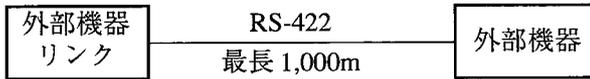


RS-232Cインタフェースによる機器接続は1対1接続で、ケーブル長は最長15mまでとなります。ケーブル長が15m以上の場合には、装置間をモデム（MODEM：変復調装置）で接続してください。

4.1.2 RS-422による接続

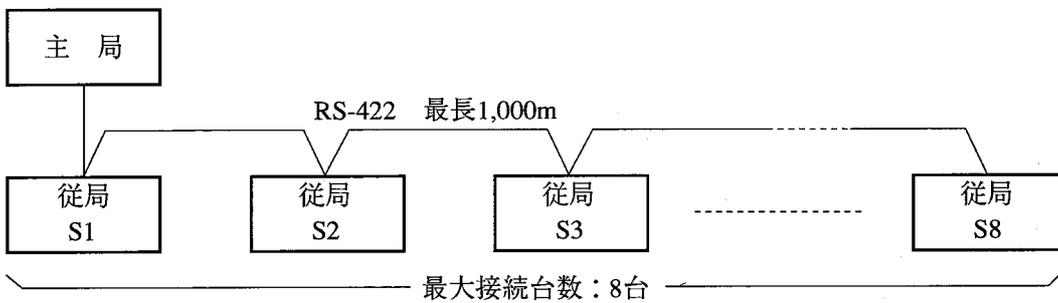
RS-422インタフェースによるパソコン、プリンタなど外部機器との接続には、1対1と1対nの2種類の接続方法があります。

(1) 1対1接続



RS-422インタフェースは、RS-232Cと比べ、ケーブル長がモデムを使用しなくても最長1,000mまで延長できます。

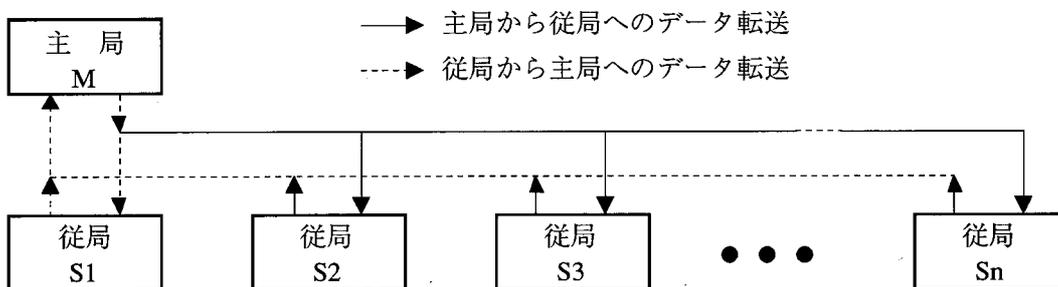
(2) 1対n接続



RS-422インタフェースは、最大8台までの外部機器をマルチドロップ方式で接続できます。

RS-422のマルチドロップ接続は、データの送信・受信に関しては、主局（または親局）と従局（または子局）の関係が必要であり、かつ通信回線を共通使用するために、通信規則（プロトコル）が必要となります。

すなわち、RS-422のマルチドロップ接続できる外部機器は、パソコンやコンピュータなどのインテリジェントな機器のみであり、プリンタやCRT端末などのノンインテリジェント機器は、接続できません。

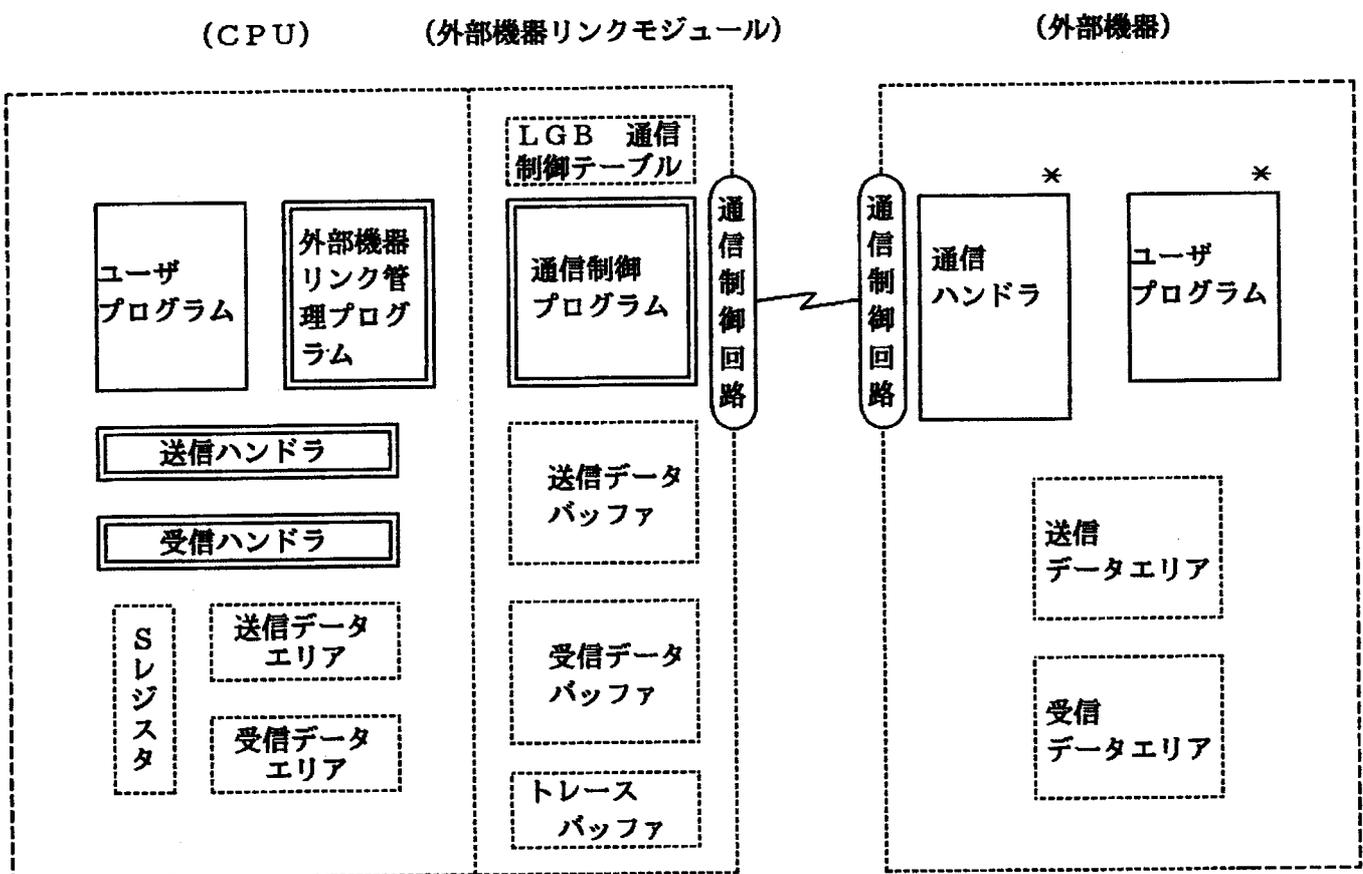


## 4.2 外部機器リンクシステムのソフトウェア構成

外部機器リンクシステムを構成するためには、まず接続したい機器のハードウェア仕様を確認して、接続可能かどうかを調べてください。

次にどう接続するか（RS-232Cによる1対1の接続か、RS-422による1対nのマルチドロップ接続かなど）を検討します。

以上を決定した後、外部機器リンクシステムのプログラムを作成してください。それぞれを個別に説明する前に提供システムプログラムを含めた外部機器リンクシステムのソフトウェア構成概要を示します。



- ・  はシステム提供、 はユーザ作成プログラムを表します。
- ・ \*は必要な場合と不必要な場合があります。

### 4.3 外部機器リンクシステムに提供されるシステムプログラム

「4.2 外部機器リンクシステムのソフトウェア構成」で示したシステム提供プログラムについて説明します。システムプログラムはCPUまたは外部機器リンクモジュール上で動作し、ポータブルPSE $\alpha$ にてローディングします。

システムプログラムは

- (1) 通信制御プログラム
- (2) 外部機器リンク管理プログラム
- (3) 送信ハンドラ
- (4) 受信ハンドラ

の4つに分類できます。また(1)～(4)まではそれぞれメインモジュール用(チャンネル0)、サブモジュール用(チャンネル1)の2種類が使用できます。

#### 4.3.1 通信制御プログラム

送信ハンドラから送信指令を受けると回線に送信データを送り出します。また、回線から受信データをすべて受けとったら、外部機器リンク管理プログラムに対して受信完了を知らせます。

#### 4.3.2 外部機器管理プログラム

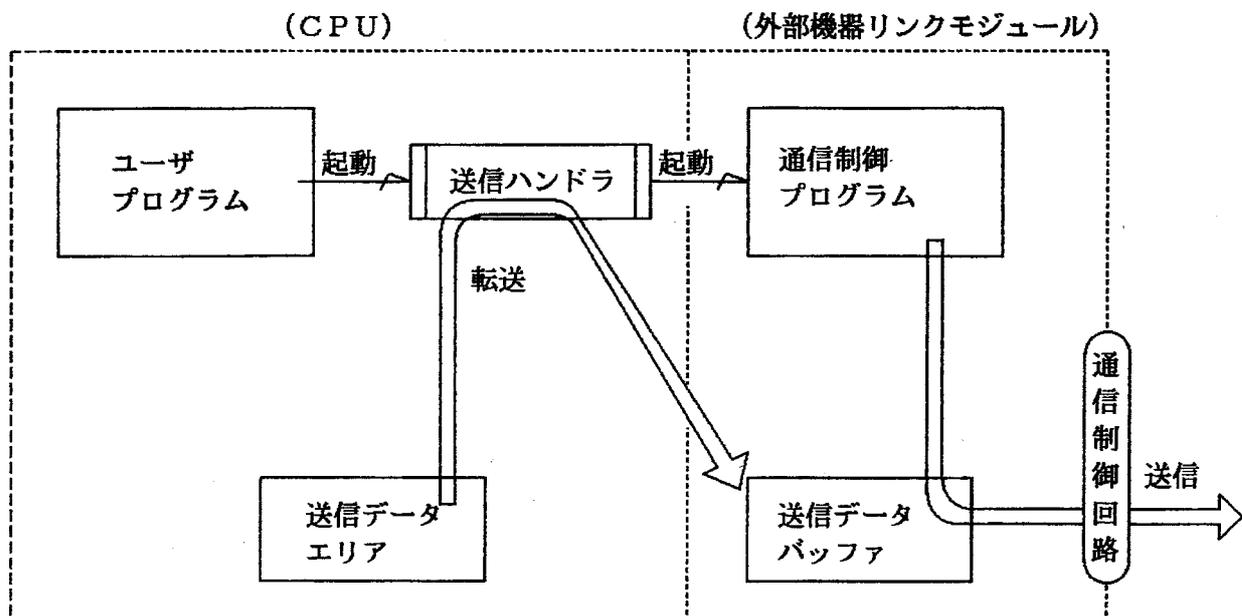
通信制御プログラムから送受信に関する情報を、CPUのシステムレジスタなどによってユーザプログラムに知らせます。(「4.7 送信情報」、「4.8 受信情報」参照)

#### 4.3.3 送信ハンドラ

送信ハンドラの機能を以下に示します。

- ・ユーザプログラムによって指定されたエリアから指定語数分送信データを外部機器リンクモジュール内の送信バッファへ転送。
- ・通信制御プログラムへ送信開始指令を発行。
- ・送信起動に対する各種エラーチェック。

ユーザプログラムは、送信したいタイミングで送信ハンドラを起動します。



## 4 利用の手引き

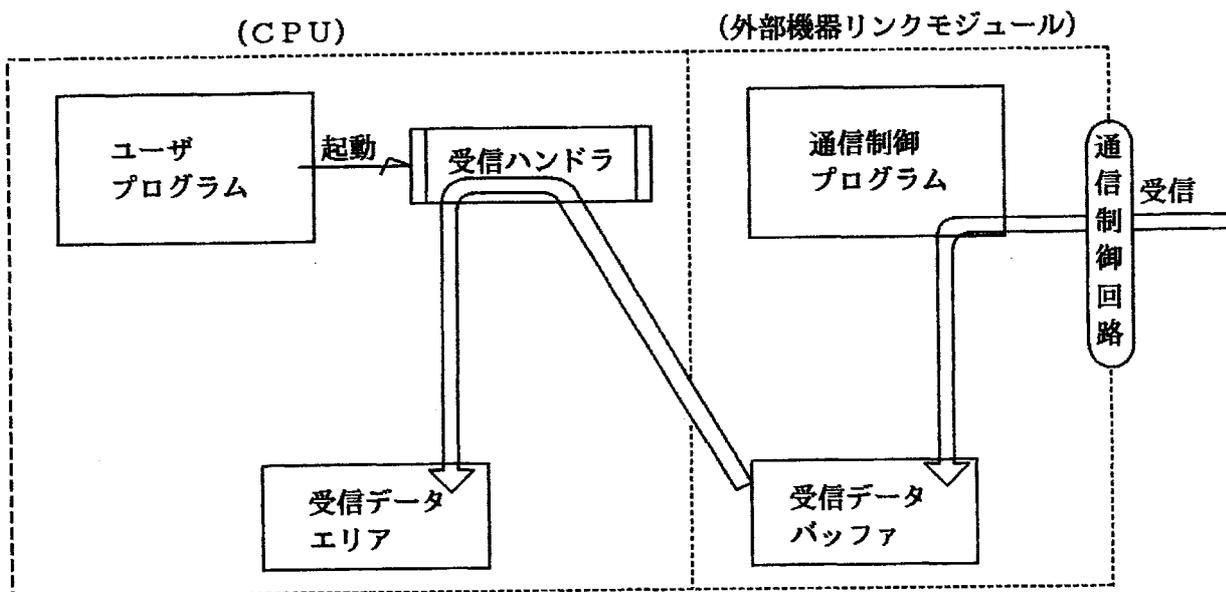
送信ハンドラは、メインモジュール用（チャンネル0）とサブモジュール用（チャンネル1）の2種類が使用できます。また、ユーザプログラムの違い（「4.4.1 ユーザアプリケーションプログラム」参照）により2種類に区別され、計4種類の送信ハンドラが使用できます。（「4.10 ハンドラユーザリンクエージ」参照）

### 4.3.4 受信ハンドラ

受信ハンドラの機能を以下に示します。

- ・受信バッファに未取込の受信データがある場合、ユーザプログラムによって指定されたエリアへ指定語数分通信データを転送。
- ・データを取り込んだ受信バッファを空バッファとして、通信制御プログラムへ知らせる。
- ・受信ハンドラ起動に対する各種エラーチェック。

ユーザプログラムは受信完了を認識すると、受信ハンドラを起動し、データを取り込みます。（受信完了認識は「4.10 ハンドラユーザリンクエージ」参照）



受信ハンドラはメインモジュール用（チャンネル0）とサブモジュール用（チャンネル1）の2種類が使用できます。また、ユーザプログラムの違い（「4.4.1 ユーザアプリケーションプログラム」参照）により2種類に区別され、計4種類の受信ハンドラが使用できます。（「4.10 ハンドラユーザリンクエージ」参照）

## 4.4 ユーザの作成するプログラム

4.3節ではシステム提供のプログラムについて説明しましたが、この節では、外部機器リンクシステムを構成するために、ユーザが作成するソフトウェアについて説明します。

### 4.4.1 ユーザアプリケーションプログラム

送信、受信ハンドラはユーザアプリケーションから起動されます。ユーザプログラムには次の2種類の形態があります。

- (1) ラダープログラム …シーケンスプログラムとも呼ばれます。A接点 (  )、B接点 (  )、出力コイル (  ) などで構成されます。
- (2) Cモードプログラム…コンピュータ言語 (C言語、アセンブラなど) で作成され、タスク、Pコイルの形で実行されます。CPMS (コンパクトPMS) と拡張メモリが必要です。

したがって、アプリケーションの種類により送受信ハンドラは2種類あります。

ラダープログラムでは演算ファンクションから送受信ハンドラを起動し、また、Cモードプログラムではサブルーチンから送受信ハンドラを起動します。

### 4.4.2 外部機器側のプログラム

送信データは送信ハンドラ、通信制御プログラムにより、回線を通して外部機器に送られます。外部機器がプリンタ、CRTなどの場合、送信データをハンドリングするプログラムは必要ありませんが、その送信されてきたデータを解析する場合には外部機器側にもプログラムが必要です。

また、外部機器リンクモジュールからデータを受信するだけでなく、外部機器リンクモジュールへデータを送信する必要があるときも同様にプログラムが必要となる場合があります。

この外部機器側のプログラムは外部機器リンクモジュールに仕様をあわせてください。

- (例) スタートコード  
 エンドコード  
 BCC  
 回線上データフォーマット (ASCIIまたはBINARY)  
 中断、再開など

したがって、設計段階でのLGB通信制御テーブル「4.5 LGB通信制御テーブル」の内容、および外部機器側プログラムの仕様については十分検討してください。

## 4 利用の手引き

### 4.5 LGB通信制御テーブル

LGB (Line Group Block) とは、通信制御プログラムが回線を通じて送受信するための伝送上の情報群を意味します。この情報はPSEによりユーザが決定します。

LGB をどのように設定するかは非常に重要であり、外部機器とハード的に接続できなかつたり、また、伝送手順が異なり正常に送受信できないことがあります。

ここではLGBの設定値と外部機器（プログラムも含む）との関係について説明します。

LGBの要素を以下に示します。

- 伝送フレーム
- 伝送速度
- 優先制御
- データ変換モード
- テキスト語数
- スタートコード数
- スタートコード
- エンドコード数
- エンドコード
- BCC
- 送信中断/再開
- 中断コード
- 再開コード
- 送信遅延時間
- 送信中断監視時間
- 受信監視時間
- RS-422ゲートコントロール
- 送信要求
- データ端末レディ
- データセットレディ

#### 4.5.1 伝送フレーム

回線上の1バイトデータのフレーム構成を決定します。

#### 4.5.2 伝送速度

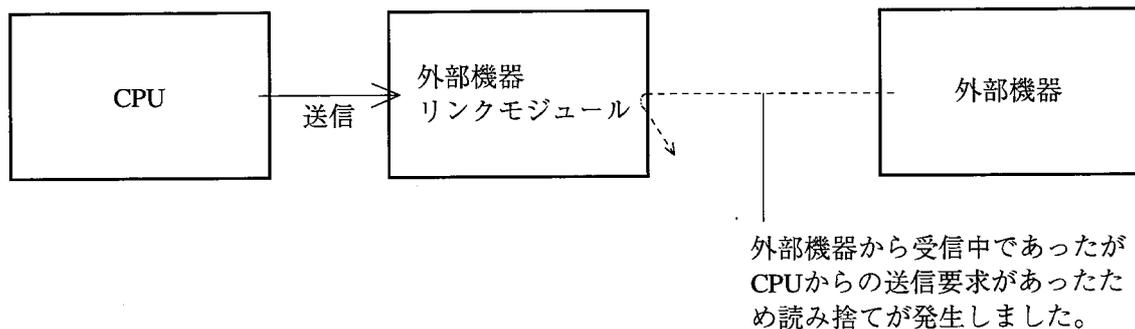
回線の伝送速度 (bps) を設定します。(150~9,600bps)

## 4.5.3 優先制御

自局（外部機器リンクモジュール）、または他局（外部機器）の優先順位を指定します。

優先順位とは、外部機器リンクモジュールに対してCPUと外部機器の双方からの働きかけがあった場合、どちらを優先するかを意味します。

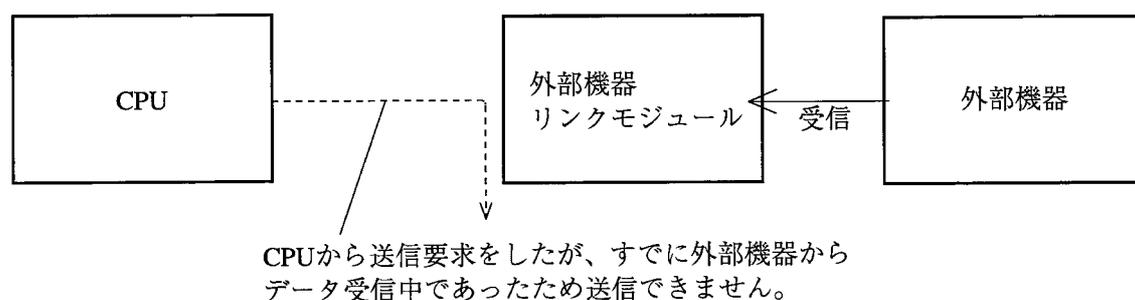
## ● 自局優先



上図例の場合、CPU上のアプリケーションプログラムは外部機器からのデータ受信を打切って送信を開始したことをSレジスタにて認識できます。

ただし、外部機器側はデータ読み捨てが発生したことは認識できないので、CPU側から外部機器側へ知らせる必要があります。

## ● 他局優先



上図例の場合、CPU上のアプリケーションプログラムが送信できないことをSレジスタにて確認できます。

## 4 利用の手引き

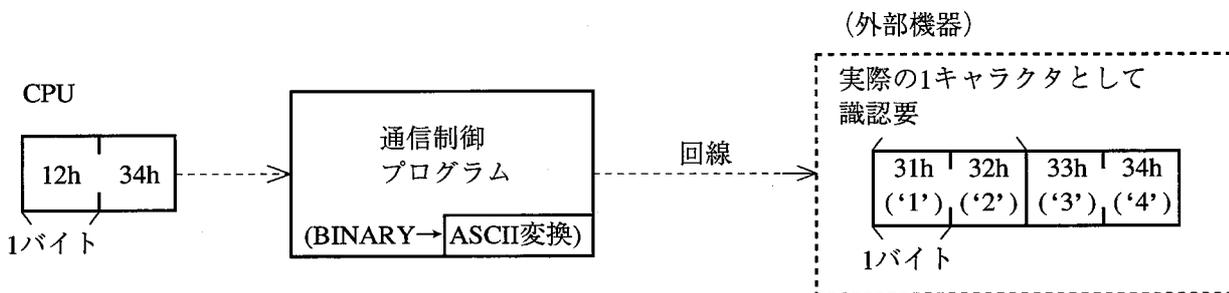
### 4.5.4 データ変換モード

回線上のテキストデータをASCII（アスキー）データとして扱うか、BINARY（バイナリ）データとして扱うかの選択を行います。

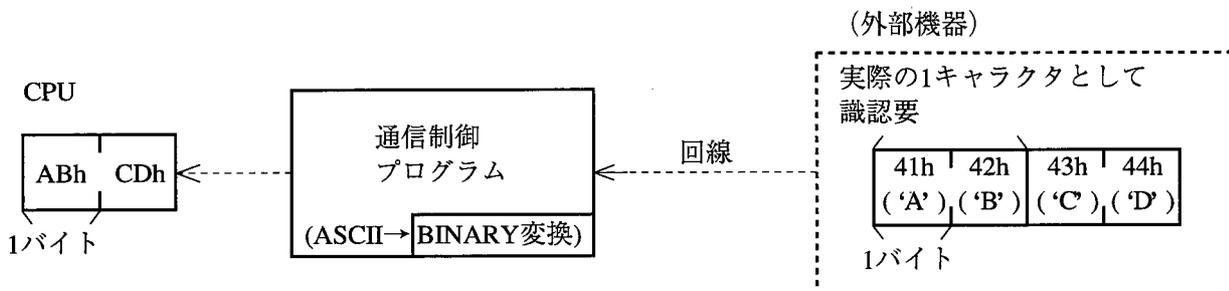
#### ● ASCII指定の場合

ASCIIを指定すると、外部機器リンクモジュールは回線上のデータを4ビット長のBINARY値（0～Fh）をASCII変換した値（30h～39h, 41h～46h）として扱います。

したがって、回線上のデータ量は2倍となり‘0’～‘9’，‘A’～‘F’以外のテキストデータを受信するとエラーとなります。



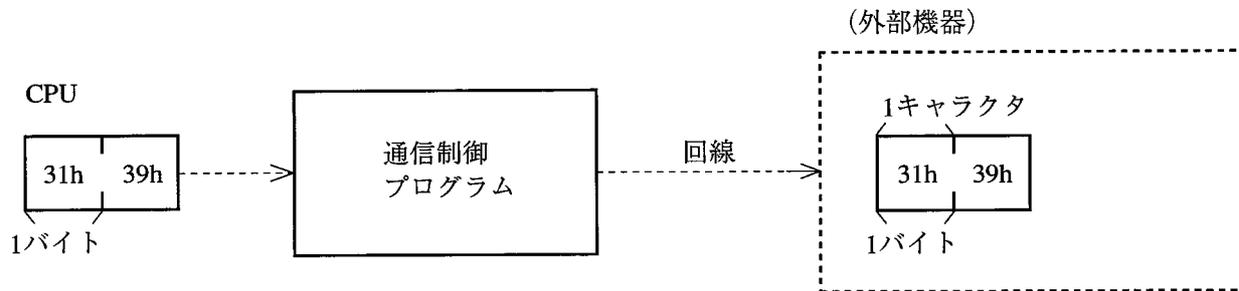
上図例の場合、CPUからデータ12h, 34hを送信すると通信制御プログラムは、BINARY→ASCII変換を行い、外部機器へデータ31h（‘1’）、32h（‘2’）、33h（‘3’）、34h（‘4’）を送信しますので、それにあわせて外部機器側のプログラムを作成する必要があります。



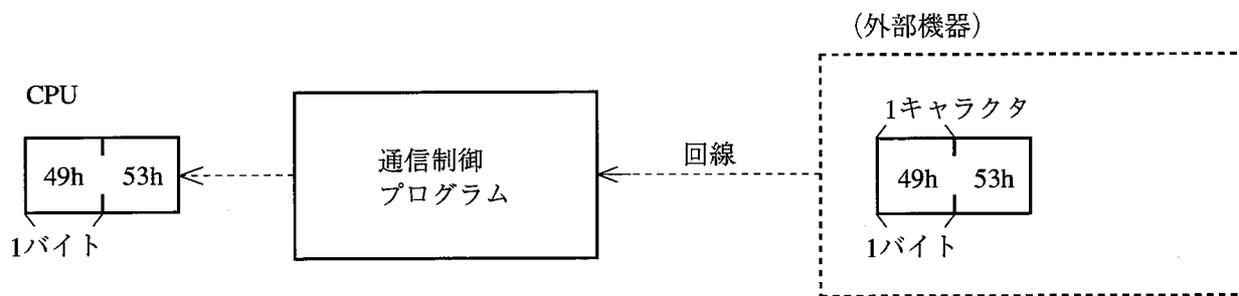
上図例の場合、外部機器からデータ41h（‘A’）、42h（‘B’）、43h（‘C’）、44h（‘D’）を受信すると通信制御プログラムはASCII→BINARY変換を行い、CPUへデータABh, CDhを渡しますので、それにあわせて外部機器側のプログラムを作成する必要があります。

## ● BINARY指定の場合

BINARYを指定すると、外部機器側のプログラムでASCII↔BINARY変換する必要はありません。



上図例の場合、CPUからデータ31h, 39hを送信すると、通信制御プログラムはそのまま外部機器へデータ31h, 39hを送信します。

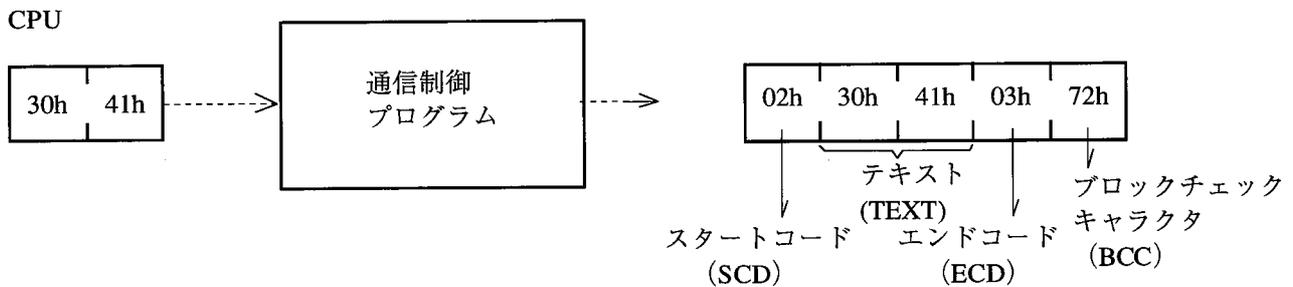


上図例の場合、外部機器からデータ49h, 53hを受信すると、通信制御プログラムはそのままCPUへデータ49h, 53hを渡します。

## 4 利用の手引き

### 4.5.5 テキスト語数、スタートコード数、スタートコード、エンドコード数、エンドコード、BCCコード

この項では送受信データを一連のブロックとし、CPUから送信する場合を例に説明します。(テキストはBINARY指定とします。)



上図でわかるように通信制御プログラムはデータにLGBの指定によりスタートコード (SCD)、エンドコード (ECD)、ブロックチェックキャラクタ (BCC) コードを付加して送信します。

上記例は スタートコード………1文字指定02h (STX:テキスト開始)

エンドコード………1文字指定03h (ETX:テキスト終了)

ブロックチェックキャラクタ………水平偶数パリティとした場合です。

したがって、このLGB (通信制御テーブル) の指定にあわせて外部機器側のプログラムを作る必要があります。

下図にブロック構成を示します。



以降各部について説明します。

#### (1) スタートコード (SCD)

テキストの開始を示すデータで、有/無指定、および有の場合のコード数 (1~4文字)、コードデータを設定できます。

スタートコード有の場合、通信制御プログラムは、スタートコード受信で初めて、外部機器からの受信と認識し、それ以前に受信したデータはすべて無視します。

また外部機器への送信の際には、テキストデータの次へ指定スタートコードを付加して送信します。

スタートコードは、ASCII指定の場合には、ASCII変換しません。

#### (2) エンドコード (ECD)

テキスト文字の終了を示すデータで、有/無指定、および有の場合のコード数 (1~4文字)、コードデータを設定できます。

エンドコード有の場合、通信制御プログラムは、エンドコード受信で外部機器からの受信終了と認識します。

また外部機器への送信の際には、テキストデータの次へ指定エンドコードを付加して送信します。

エンドコードは、ASCII変換の場合には、ASCII変換しません。

## (3) テキスト (TEXT) 語数

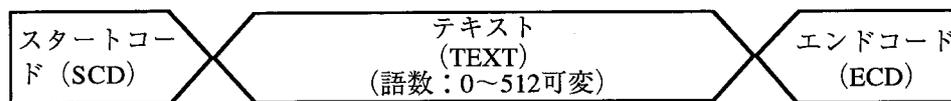
テキスト語数は0~512の範囲で指定できます。

スタートコード有の場合、テキストデータのスタートは、スタートコード受信後の次のデータからとし、テキストデータは、エンドコードの受信または指定したテキスト語数分のデータ受信で終了します。

したがって、テキスト語数、スタートコード、エンドコードの指定をうまく行くと、様々な形のブロックを送受信できます。

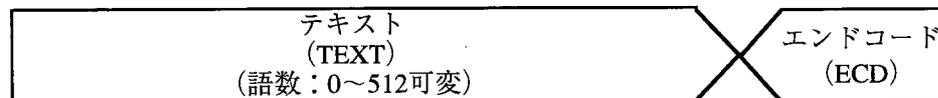
ASCII指定の場合、通信制御プログラムは、回線データをASCII→BINARY変換します。

## ● スタートコード、エンドコードありの場合



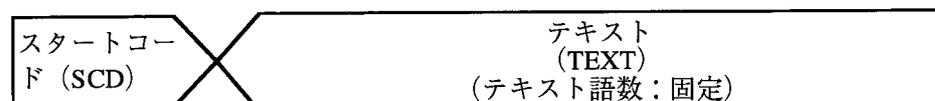
上記の場合、テキスト語数を512としてもテキスト内にエンドコードをユーザが設定すれば、通信制御プログラムは送受信を終了します (BINARY指定のみ)。また、エンドコードが存在しないとテキスト長を512とし、その前後にスタートコード、エンドコードをつけたものとして処理します。

## ● エンドコードありの場合



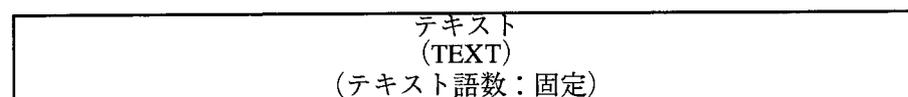
上記の場合もテキスト内にECDをユーザが設定することにより、テキスト長を可変として扱うことができます。

## ● スタートコードありの場合



上記の場合はテキスト長はテキスト語数指定分固定となります。

## ● テキストのみの場合



## 4 利用の手引き

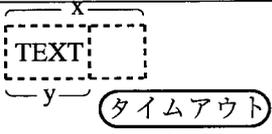
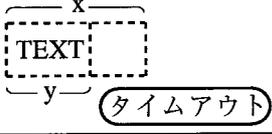
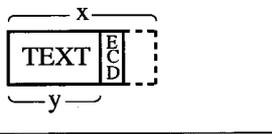
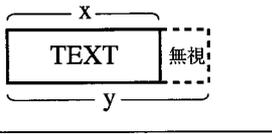
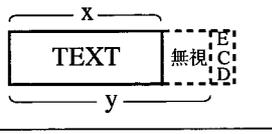


LGB 指定テキスト語数と送信ハンドラの送信語数の関係を以下に示します。  
送信ハンドラ送信語数を  $x$  バイト、LGB 指定テキスト語数を  $y$  バイトとします。

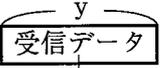
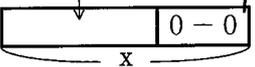
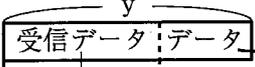
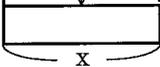
大小関係	LGB 指定 エンドコード	テキスト中に エンドコード	回線に送信されるデータ	
$x \geq y$	無	—		LGB指定テキスト語数分が 送信される。
$x \geq y$	有	無		LGB指定テキスト語数+ ECD (エンドコード) が送 信される。
$x \geq y$	有	有		テキストの先頭からテキス ト中のECD (エンドコー ド) までが送信される。
$x < y$	無	—		テキスト部+以前の送信 バッファのデータが送信さ れる。
$x < y$	有	無		テキスト部+以前の送信 バッファのデータ+ ECD (エンドコード) が送信さ れる。
$x < y$	有	有		テキストの先頭からテキス ト中のECD (エンドコー ド) までが送信される。

参

- ・LGB指定テキスト語数と受信ハンドラの取込語数の関係を以下に示します。  
LGB指定テキスト語数をxバイト、回線よりの受信データ語数をyバイトとします。

大小関係	LGB 指定 エンドコード	テキスト中に エンドコード	受信データバッファに格納されるデータ	
$x > y$	無	—		LGB指定語数まで、受信データを持ち、TEXT (テキスト) 監視時間のタイムアウトとなる。
$x > y$	有	無		
$x > y$	有	有		テキストの先頭からテキスト中のECD (エンドコード) までが受信。
$x \leq y$	無	—		LGB指定語数のみ受信し、以降は無視される。
$x \leq y$	有	無		
$x \leq y$	有	有		テキストの先頭からテキスト中のECD (エンドコード) までが受信。

- ・受信ハンドラに対する受信バッファと受信データの関係を示します。  
受信ハンドラの取込語数をxバイトとし、実際に受信バッファに格納された語数をyバイトとします。

大小関係	ユーザ指定エリアへ取り込まれるデータ	
$x \geq y$	<p>(受信バッファ) </p> <p>受信データエリア </p>	ゼロクリアします。
$x < y$	<p>(受信バッファ) </p> <p>受信データエリア </p>	演算ファンクションハンドラの場合、取り残しデータは無視します。タスクハンドラの場合は (4. 10. 2) を参照してください。

## 4 利用の手引き

### (4) ブロックチェックキャラクタ (BCC)

送受信フレームの合理性チェック用データでエンドコードありの場合はエンドコードの次に、エンドコードなしの場合は、テキストの次に存在します。

ブロックチェックキャラクタのチェックに関しては、有/無の指定とブロックチェックキャラクタチェックありの場合は水平偶数または水平奇数パリティを指定できます。

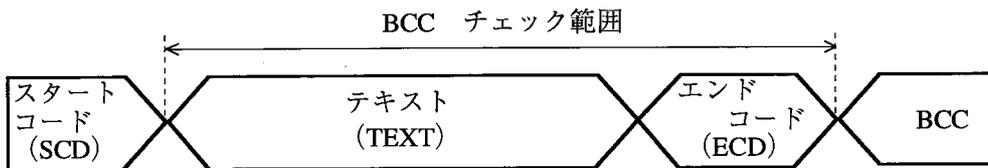
$$\text{水平偶数パリティ} \cdots \cdots (\text{BCC})_E = (00\text{h}) \text{ EOR } \left( \sum_{i=0}^n \text{EOR } D_i \right)$$

$$\text{水平奇数パリティ} \cdots \cdots (\text{BCC})_{07} = (7\text{Fh}) \text{ EOR } \left( \sum_{i=0}^n \text{EOR } D_i \right) : (\text{データビット、7ビット})$$

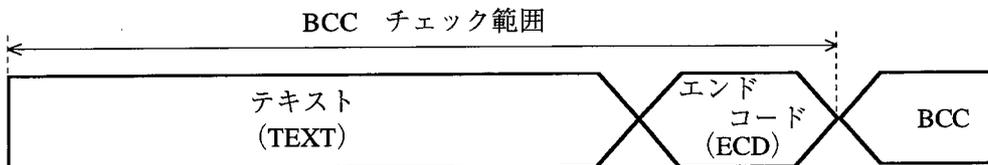
$$(\text{BCC})_{08} = (\text{FFh}) \text{ EOR } \left( \sum_{i=0}^n \text{EOR } D_i \right) : (\text{データビット、8ビット})$$

以下に、ブロックチェックキャラクタのチェック範囲を示します。

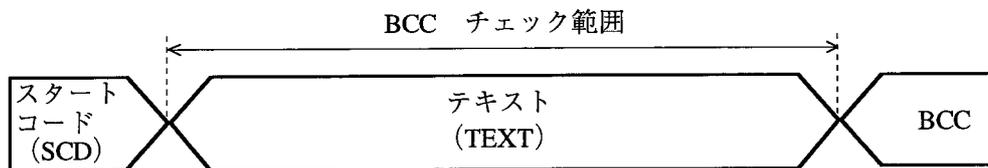
- スタートコード、エンドコードありの場合



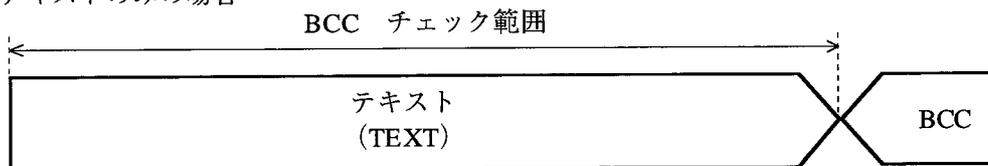
- エンドコードありの場合



- スタートコードありの場合



- テキストのみの場合



エンドコードがない場合は、テキスト語数指定分の固定長としてチェックします。

ブロックチェックキャラクタのチェックありの場合は、上記規則により外部機器側のプログラムを作成しなければなりません。

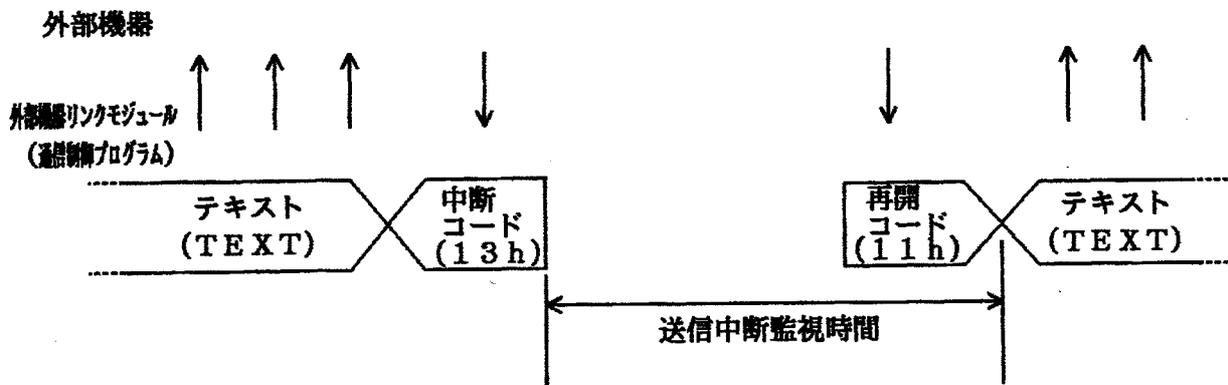
## 4.5.6 送信中断／再開、中断コード、再開コード、送信中断監視時間

外部機器側がTEXT受信中に何らかの原因（処理しきれないなど）により通信制御プログラムの送信に対しての中断／再開を要求する場合に使用します。

中断／再開処理の有無、中断／再開処理有の場合の中断コード（1～2キャラクタ）、再開コード（1～2キャラクタ）を指定します。

中断コード受信後、通信制御プログラムは、再開コードのみ受信でき、その他のコードは無視します。また中断／再開コード共、ASCII指定でもそのまま変換せず使用します。

送信中断監視時間は、通信制御プログラムが中断コードを受信してから、再開コードを受信するまでの時間を意味し、オーバーするとエラーとします。



上記例は、

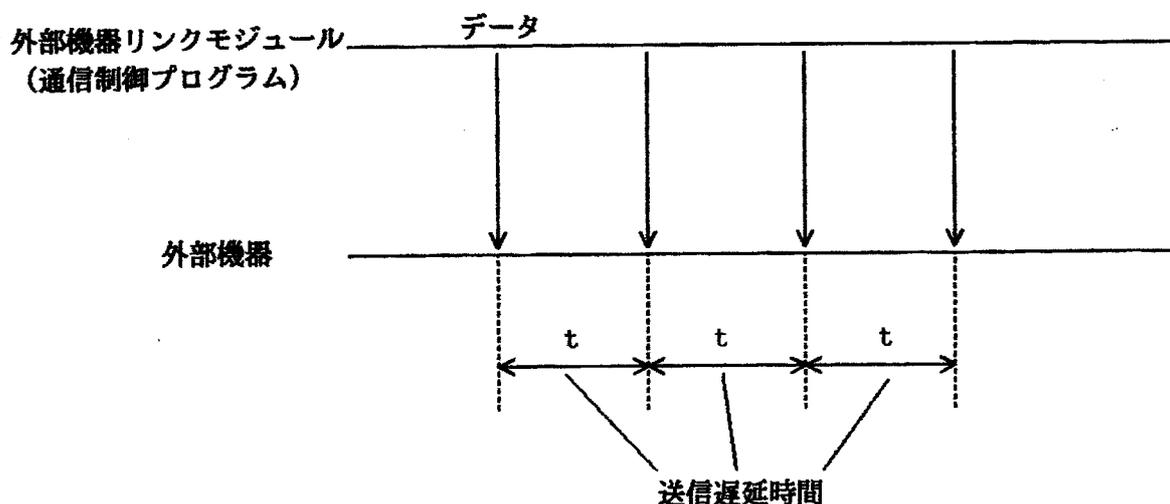
中断コード……1文字指定：13h（DC3；装置制御3〔X-OFF〕）

再開コード……1文字指定：11h（DC1；装置制御1〔X-ON〕）

後で説明する受信中断監視時間と送信中断監視時間は、それぞれ独立して通信制御プログラムが監視します。したがって、送信中断中であっても、受信中断監視時間を越えた場合はエラーとします。

## 4.5.7 送信遅延時間

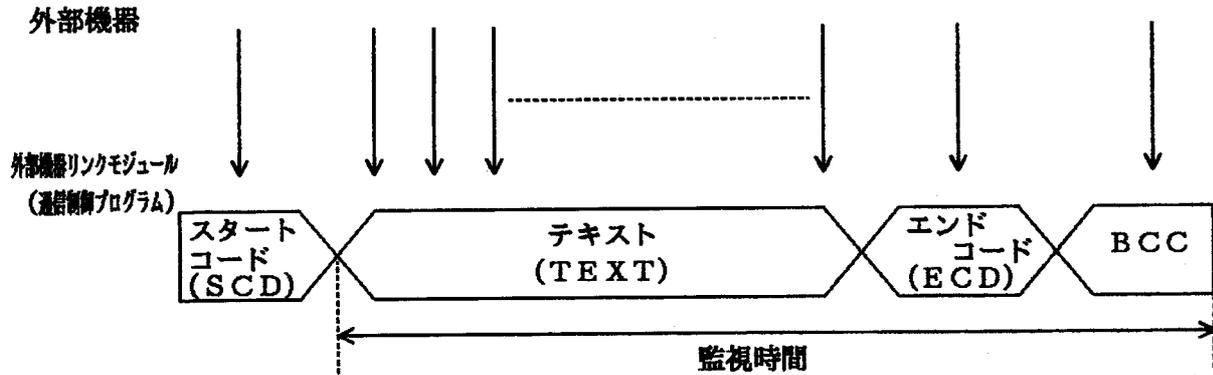
通信制御プログラムから外部機器へデータ送信時、データ送信の時間間隔を規定します。



## 4 利用の手引き

### 4.5.8 受信監視時間

通信制御プログラムのテキスト受信時のテキスト受信開始から全データ受信終了までの時間監視時間を規定します。



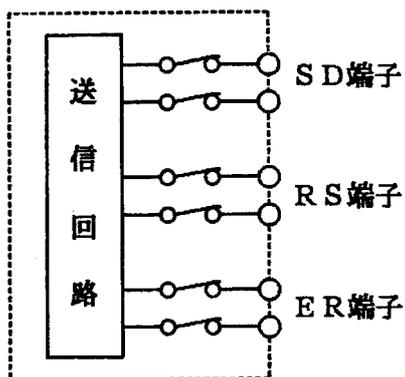
### 4.5.9 RS-422送信ゲートコントロール

RS-422の送信ゲートコントロール方法をオープン、コントロールにて指定します。

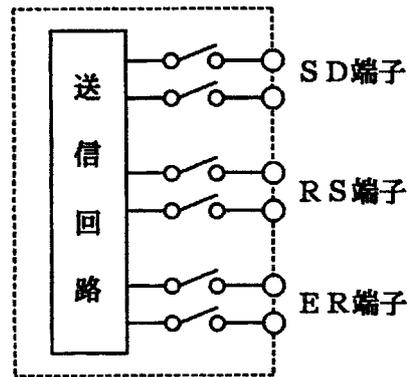
なお、送信ゲートオープンは、RS-422の出力端子（送信データ：SD、送信要求：RS、データ端末レディ：ER）が回線と電氣的に接続されることを意味します。

送信ゲートクローズは、RS-422の出力端子（送信データ：SD、送信要求：RS、データ端末レディ：ER）が回線と電氣的に切り離されていることを意味します。

ゲートオープン状態  
(外部機器リンクモジュール)



ゲートクローズ状態  
(外部機器リンクモジュール)



- オープンの場合  
送信ゲートを常にオープン状態とします。

- コントロールの場合

送信ゲートのオープン↔クローズを自動制御します。

(通常はゲートクローズ状態で送信時のみゲートをオープン状態とします。)

[1対1接続時の設定]

使用する外部機器リンクモジュールすべてをゲートオープンで設定してください。

[1対nマルチドロップ接続時の設定]

マスタとなる外部リンクモジュールは、ゲートオープンで設定してください。

スレーブとなる外部機器リンクモジュールすべてをゲートコントロールで設定してください。

(スレーブ側出力信号 (SD, RS, ER) の回線上での衝突を防ぎます。)

#### 4.5.10 送信要求 (Request to Send : RS)

外部機器に対して、送信要求の有無 (RS端子の状態) 出力を指定します。

送信要求あり指定時のみ、外部機器リンクモジュールは、送信データを送信できます。

- ・送信要求ありの場合

外部機器リンクモジュールは、外部機器に対し、常時送信要求ありを出力し続けると共に、送信可能状態となります。

- ・送信要求なしの場合

外部機器リンクモジュールは、外部機器に対し、常時送信要求なしを出力し続けると共に、送信不可能状態となります。

( 送信要求なし指定時に、送信データを送信しますと、送信データは送信されずに、CPUのシステムレジスタ (「4.7 送信情報」参照) の送信可フラグは、'現在送信中'のままとなりますので、注意してください。 )

- 外部機器へデータを送信する場合は、送信要求ありを設定してください。

- 外部機器へデータを送信しない場合は、送信要求なしを設定してください。

外部機器側に受信可能/不可能切替機能がある場合は、外部機器リンクモジュールのRS端子と外部機器の受信可能/不可能検出端子 (CD端子) を接続することにより送信データ以外の無効データ (ノイズなど) の誤受信を防ぐことができます。

## 4 利用の手引き

### 4.5.11 データ端末レディ (Equipment Ready : ER)

外部機器機に対して、外部機器リンクモジュールのレディ、ノットレディ出力を指定します。

レディ、ノットレディの定義は、外部機器リンクモジュールと外部機器間のプロトコルによりますが、外部機器リンクモジュールが受信可能をレディと定義します。

- ・レディの場合

外部機器リンクモジュールは、外部機器に対しデータ端末レディ (ER) 端子からレディ状態を出力し続けます。

- ・ノットレディの場合

外部機器リンクモジュールは、外部機器に対しデータ端末レディ (ER) 端子からノットレディ状態を出力し続けます。

- 外部機器からデータを受信する場合は、レディを設定してください。

- 外部機器からデータを受信しない場合は、ノットレディを設定してください。

外部機器側に送信可能/不可能変換機能がある場合には、外部機器リンクモジュールのER端子と外部機器の送信可能/不可能検出端子 (DRまたはCS端子) を接続し、外部機器側を送信可能/不可能状態に制御します。

### 4.5.12 データセットレディ (Data set Ready : DR)

外部機器のレディ状態 (DR端子の状態) のチェック有無を指定します。

- ・チェックありの場合

外部機器のレディ状態 (DR端子の状態) をチェックし、レディ状態のときのみ送信データを送信します。外部機器がノットレディのときに送信データを送信するとエラーとします。

- ・チェックなしの場合

外部機器のレディ状態 (DR端子の状態) をチェックせず、外部機器に対し送信データを送信します。

- 外部機器側に受信レディ出力機能がある場合

- 外部機器リンクモジュールのDR端子と外部機器の受信レディ出力端子 (ER端子) とを接続し、チェックありを設定します。

- 外部機器側に受信レディ出力機能がない場合

- チェックなしを設定します。

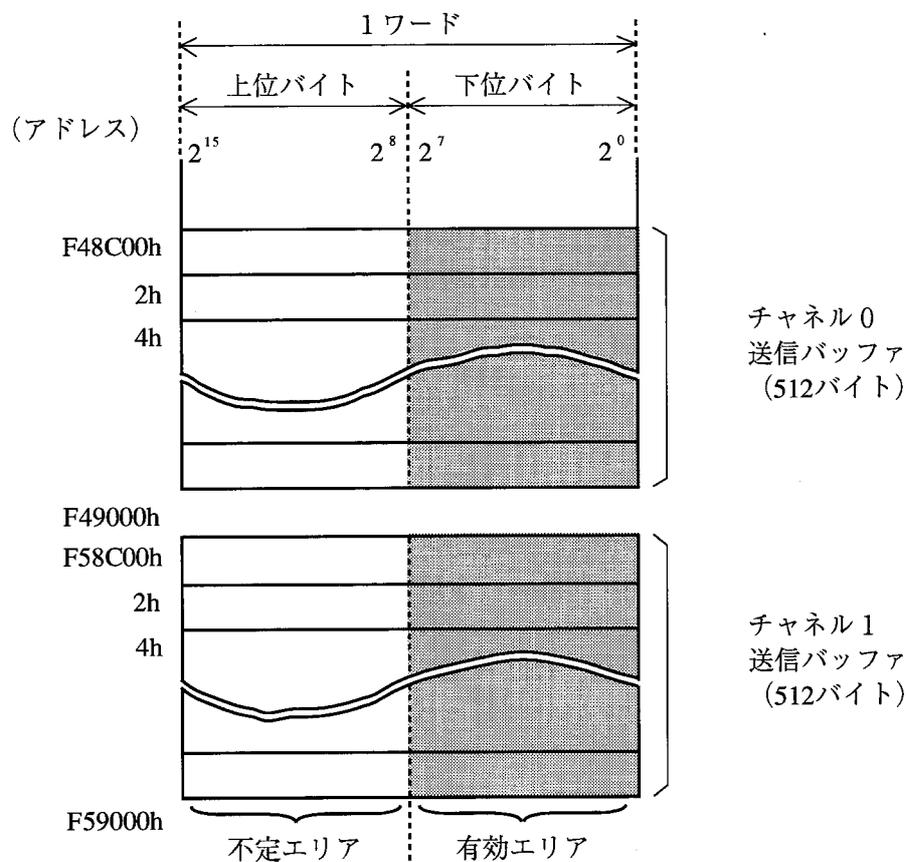
## 4.6 送信バッファ、受信バッファ

CPUから外部機器へ送信する際、送信データは、送信ハンドラによって送信データエリアから、通信制御プログラム内の送信バッファへ転送され、その後回線へ送出されます。

送信バッファは、モジュール（チャンネル0, 1）に1つだけ存在します。

したがって、前の送信処理が終了しないうちに送信起動をした場合には、二重起動（送信ビジー）としてシステム（S）レジスタへ反映します。（「4.7 送信情報」参照）

チャンネル0, 1の送信バッファ構成を示します。



(注) 外部機器リンクモジュールの送信データバッファは、下位バイト（2<sup>7</sup>～2<sup>0</sup>）のみ有効です。

## 4 利用の手引き

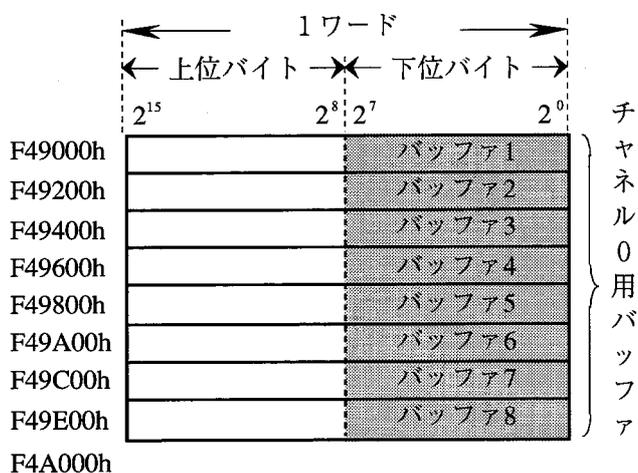
外部機器リンクからの受信データは、通信制御プログラムにより受信バッファに格納されます。ユーザプログラムは、受信ハンドラにより、受信データエリアへデータを取り込みます。

受信バッファは、LGBのテキスト語数が256バイト以下の場合、8ケースとして扱い、257~512バイトの場合は、4ケースとして扱う、サイクリック（リング）バッファ構成です。

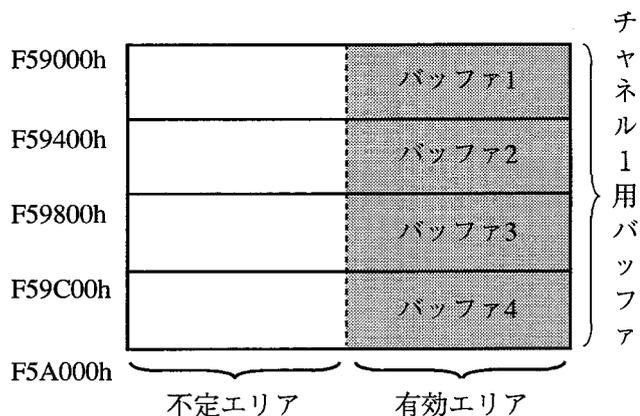
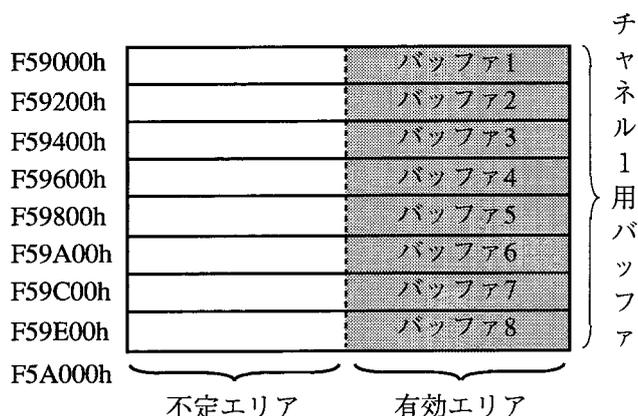
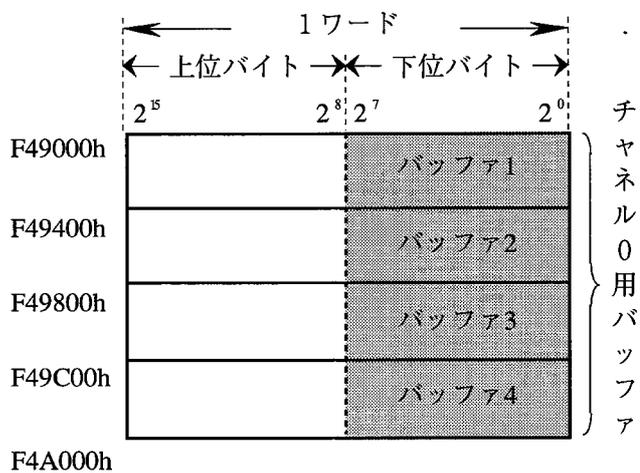
受信ハンドラは、受信バッファケースの中でもっとも古いデータを取り込みます。したがって、全受信バッファに未取込み受信データがある状態でデータを外部機器から受信した場合には、最も古いデータを格納しているバッファに上書きします。

ユーザプログラムからは意識する必要はありませんが、チャンネル0, 1の受信バッファ構成を示します。

テキスト語数が256バイト以下の場合



テキスト語数が257バイト以上の場合



・外部リンクモジュールの受信データバッファは、下位バイト（ $2^7 \sim 2^0$ ）のみ有効です。

4.7 送信情報

CPUから外部機器に対する送信は、システム提供の送信ハンドラによって行いますが、送信動作に対する情報は、CPUのSレジスタ（システムレジスタ）に反映されます。

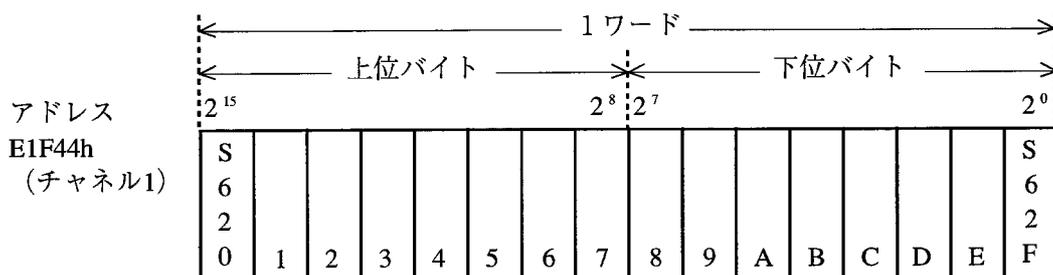
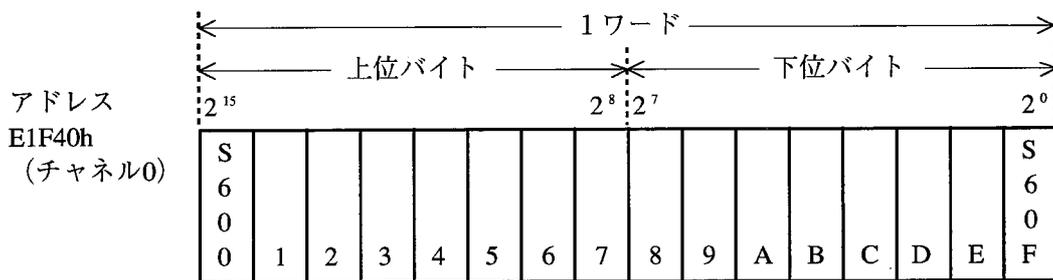
ユーザプログラムは、このSレジスタを参照し、送信可能/不可能、送信エラーを識別します。

チャンネル0の送信情報は、S600～S60Fに割付けられています。チャンネル1の送信情報は、S620～S62Fに割付けられています。

チャンネル0	チャンネル1	意味	ビット内容	
			0	1
S600	S620	送信可フラグ	送信可能	現在送信中
1	1	ハンドラエラーフラグ	エラーなし	エラーあり
2	2	通信制御プログラムエラー	エラーなし	エラーあり
3	3	受信打ち切り送信	エラーなし	エラーあり
4	4	未使用	未使用	未使用
5	5			
6	6			
7	7			
8	8	エラー詳細コード	ハンドラエラー、通信制御プログラムのエラーの内容をコードで示す（「4.11 送信エラーコード表」下位バイト参照）。	ハンドラエラー、通信制御プログラムのエラーの内容をコードで示す（「4.11 送信エラーコード表」下位バイト参照）。
9	9			
A	A			
B	B			
C	C			
D	D			
E	E			
S60F	S62F			

レジスタは、CPUのリセットにより0に初期化されます。

また、Sレジスタは、CPUからワードデータとして読込むことができます。チャンネル0用Sレジスタは、アドレスE1F40hに割付けられています。チャンネル1用Sレジスタは、アドレスE1F44hに割付けられています。



## 4 利用の手引き

### 4.8 受信情報

外部機器からのデータ受信は、通信制御プログラムが行い、正常および異常受信ともその情報は、CPUのシステム(S)レジスタに反映されます。

ユーザプログラムは、このSレジスタを参照することにより、受信データの有無、受信エラーの識別をします。

チャンネル0の受信情報は、S610~S61Fに割付けられています。チャンネル1の受信情報は、S630~S63Fに割付けられています。

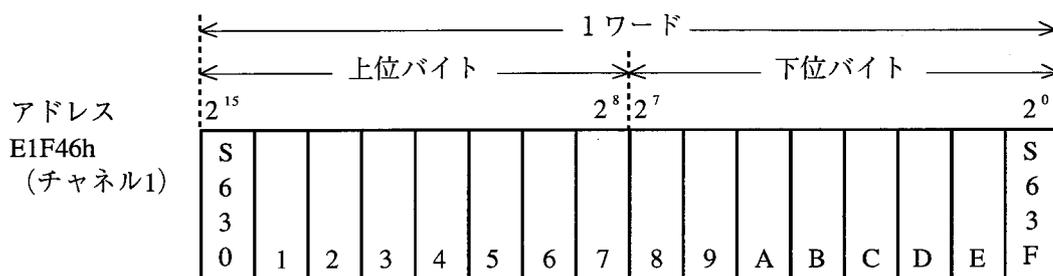
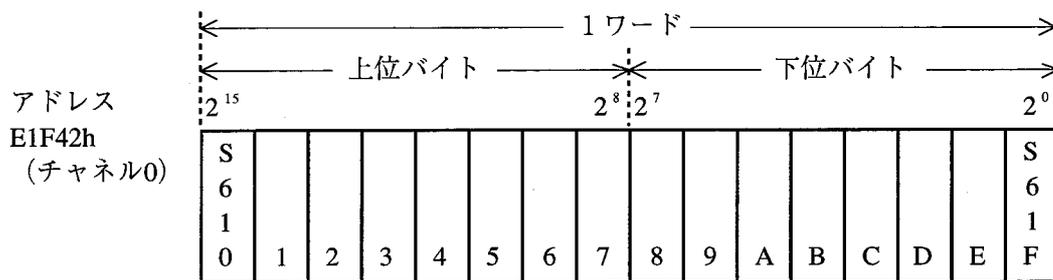
チャンネル0	チャンネル1	意味	ビット内容	
			0	1
S610	S630	受信完了フラグ	受信データなし	受信データあり
1	1	ハンドラエラーフラグ	エラーなし	エラーあり
2	2	通信制御プログラムエラー	エラーなし	エラーあり
3	3	システムエラー	エラーなし	エラーあり
4	4	ハンドラエラーコード	ハンドラエラー内容を示します。受信エラーコード表上位バイトの下1桁参照。	
5	5			
6	6			
7	7			
8	8			
9	9			
A	A			
B	B	エラー詳細コード	通信制御プログラムエラー、システムエラーの内容を示します(「4.12 受信エラーコード表」下位バイト参照)。	
C	C			
D	D			
E	E			
S61F	S63F			

SレジスタはCPUのリセットにより0にイニシャライズされます。

また、SレジスタはCPUからワードデータとして読込むことができます。

チャンネル0用SレジスタはアドレスE1F42hに割付けられています。

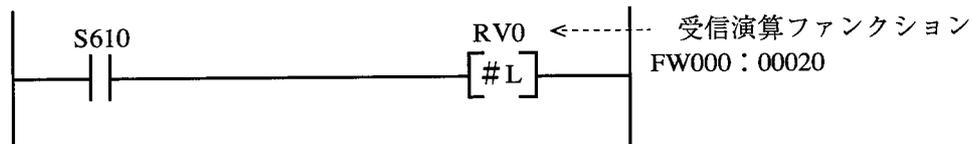
チャンネル1用SレジスタはアドレスE1F46hに割付けられています。



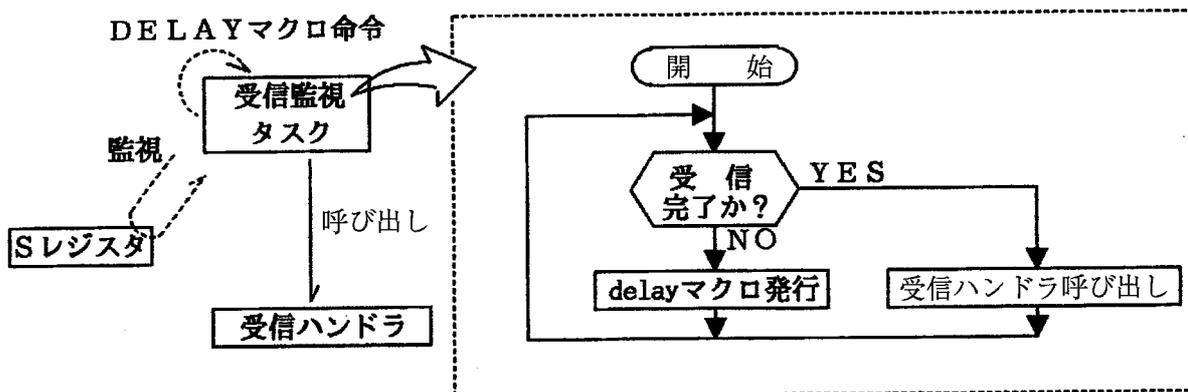
## 4.9 通信制御プログラムの受信完了時起動タスク

通信制御プログラムが受信完了すると、その情報は該当のレジスタへ反映されます。

ユーザプログラムがデータプログラムの場合は、Sレジスタを条件として受信ハンドラ（演算ファンクション）を起動すると、受信データ取込み遅れはシーケンスサイクル内（標準 30ms）でおさまります。



ユーザプログラムがCモードプログラムの場合は、Sレジスタを監視し、受信完了発生するとき、受信ハンドラ（サブルーチン）を起動する必要があります。

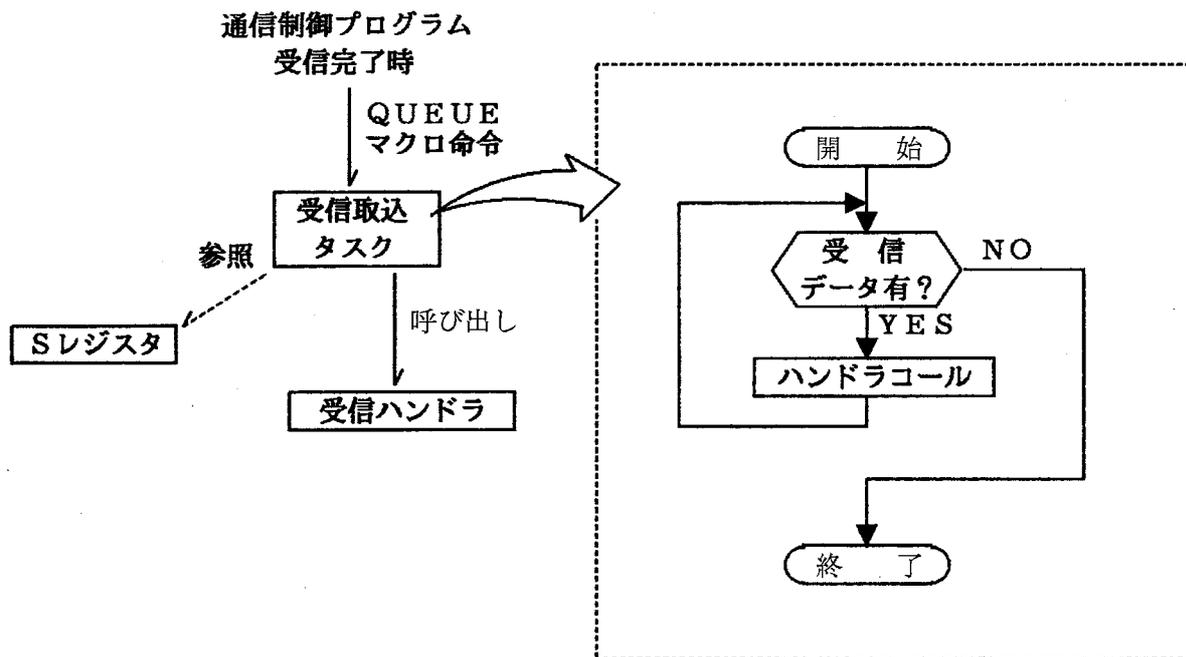


上図方式の場合、受信完了監視間隔は、CPMS（コンパクトPMS）のDELAYマクロ（処理遅延）で決まります。よってDEALYマクロによるOS負荷の増大、受信監視タスクの処理レベルが低い（処理レベルを低く設定します。）ための処理遅れにより受信データ取込み遅れが考えられます。

## 4 利用の手引き

ユーザプログラムがCモードプログラムの場合は、通信制御プログラムが受信完了時に起動するユーザタスクを作成し登録します。

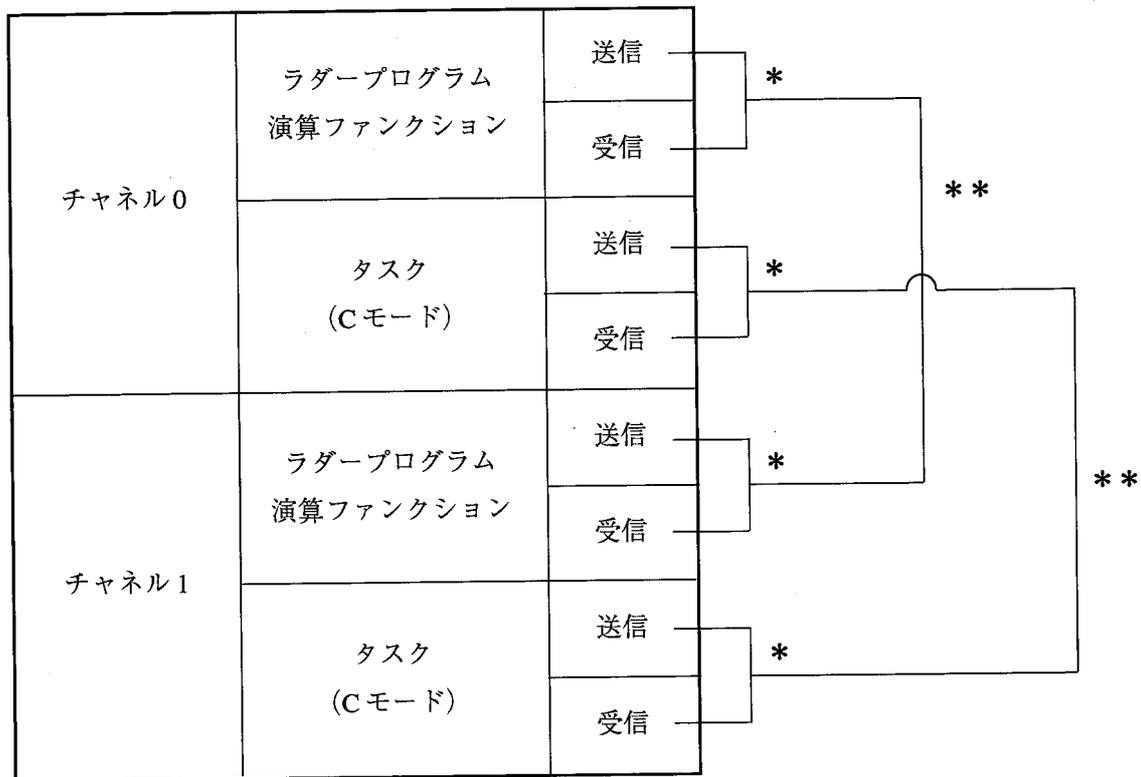
これによりユーザプログラムで受信完了を監視する必要がなくなり、通信制御プログラムから起動されたタスク内で、受信ハンドラをサブルーチンコールすれば、受信データを取り込むことができるようになります。起動タスク No. はPSE（「5.8 受信起動タスク登録」参照）、タスクは、コンパクトPMSのPSEデバッグ機能（「コンパクトPMSデバッグ PSE  $\alpha$  V5 (SAJ-3-010)」参照）で登録します。



## 4.10 ハンドラユーザーリンクage

送信、受信ハンドラには、起動されるユーザプログラムにより演算ファンクション、サブルーチンの2つの種類があります。さらにチャンネル0用、チャンネル1用の区別があります。

CPU1台にチャンネル0, 1の2台の外部機器モジュールが実装できますが、チャンネル0は演算ファンクション、チャンネル1はタスクといった仕様や、送信はタスクで受信は演算ファンクションといった使用は許されません。



\* …チャンネル0, 1のどちらかを使用のとき、可能な組み合わせ

\*\*…チャンネル0, 1の両方使用のとき、可能な組み合わせ

## 4 利用の手引き

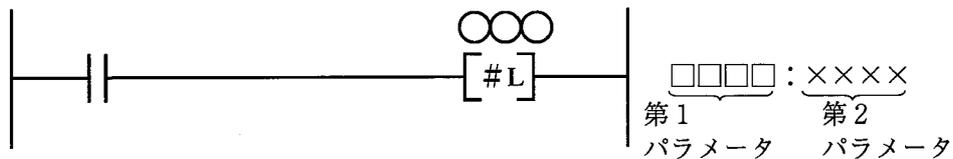
### 4.10.1 演算ファンクション

送受信の演算ファンクションは、以下に示す4種類が使用できます。

名 称	機 能
SD0	チャンネル0送信演算ファンクション
SD1	チャンネル1送信演算ファンクション
RV0	チャンネル0受信演算ファンクション
RV1	チャンネル1受信演算ファンクション

(注1) CPUおよび外部機器リンクモジュールにシステムプログラム（外部機器リンク管理プログラム、送受信用ハンドラ、通信制御制御プログラム）をロード（「5.3 システムプログラムの書込み」参照）した後に、CPUに上記の演算ファンクションをユーザ演算ファンクションとして登録（「5.7 UFET」参照）を行う必要があります。

(基本形式)



○○○ : 演算ファンクション名称

□□□□ : 転送アドレス (送受信データエリアのアドレス)

×××× : 転送語数

●転送アドレスは、ニーモニック入力 (XW000, FW000, DW000 など) できます。ただし、転送アドレス部に対しては、数値データを直接入力できません。

●転送語数は、1～512の範囲です。

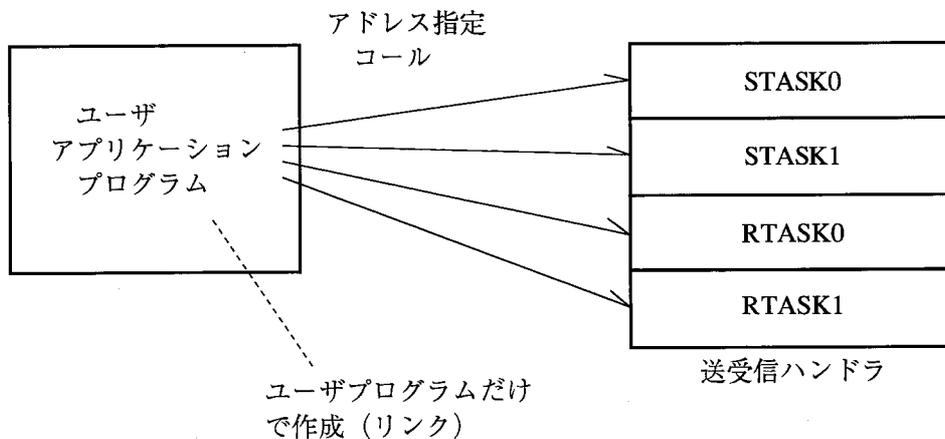
## 4.10.2 Cモードサブルーチン

Cモードプログラム用の送受信ハンドラ（サブルーチン）は、以下に示す4種類が使用できます。これらはCPUおよび、外部機器リンクモジュールにシステムプログラム（外部機器リンク管理プログラム、送受信ハンドラ、通信制御プログラム）をロードすることによりユーザアプリケーションからコールできるようにします。

名 称	アドレス	機 能
STASK0	107000h	チャンネル0送信サブルーチン
STASK1	107006h	チャンネル1送信サブルーチン
RTASK0	10700Ch	チャンネル0送信サブルーチン
RTASK1	107012h	チャンネル1送信サブルーチン

ユーザ作成のCモードアプリケーションプログラムは、C言語または68000アセンブラ言語にて作成します。

Cモードサブルーチン（Cモードプログラム用送信ハンドラ）は、アドレス指定で呼び出すため、ユーザアプリケーションプログラムは、Cモードサブルーチン（Cモードプログラム用送受信ハンドラを含めた形では作成（リンク）できません。



## 4 利用の手引き

### SD0 チャンネル0送信演算ファンクション

〈機能〉

パラメータによって指定されたエリアから指定語数分を外部機器へ送信します。(外部機器リンクチャンネル0)

〈キー入力手順〉

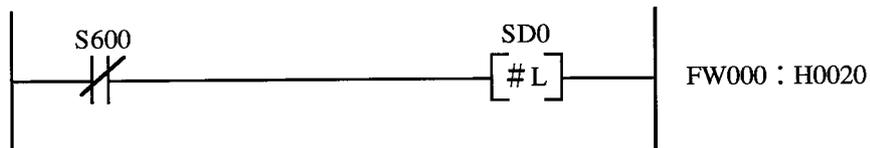
〈リターンコード〉

リターンコード (エラー情報) は、システムレジスタS600～S60Fに格納します。

(「4.7 送信情報」参照)

〈プログラム例〉

入力条件S600がOFFのとき、FW000から32バイト (16進: 20h) をチャンネル0に接続されている外部機器へ送信します。



## SD1 チャンネル1送信演算ファンクション

〈機能〉

パラメータによって指定されたエリアにより指定語数分を外部機器へ送信します。(外部機器リンクチャンネル1)

〈キー入力手順〉

[FUNC] [S] [D] [1] [シフト] [#] [転送アドレス] [転送語数]

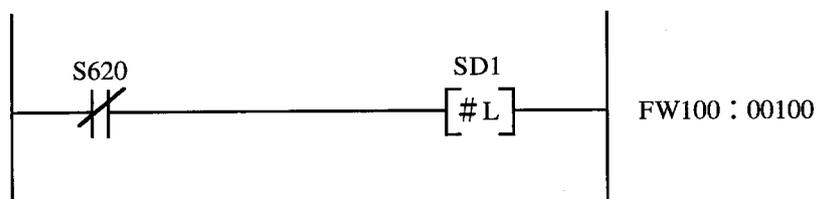
〈リターンコード〉

リターンコード(エラー情報)は、システムレジスタS620~S62Fに格納します。

(「4.7 送信情報」参照)

〈プログラム例〉

入力条件S620がOFFのとき、FW100から100バイトをチャンネル1に接続されている外部機器へ送信します。



## 4 利用の手引き

### RVO チャンネル0 受信演算ファンクション

#### 〈機能〉

パラメータによって指定されたエリアへ受信データを指定語数分転送します。受信データなしの場合は、何もしません。(外部機器リンクチャンネル0)

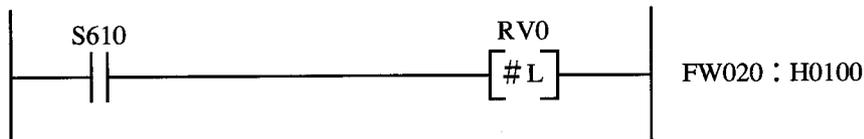
〈キー入力手順〉 FUNC R V 0 シフト # 転送アドレス 転送語数

#### 〈リターンコード〉

リターンコード (エラー情報) は、システムレジスタ S610~S61F に格納します。  
(「4.8 受信情報」参照)

#### 〈プログラム例〉

入力条件 S610 が ON (チャンネル0 受信データあり) のとき、受信データを FW020 から 256 バイト (16進: 100h) 転送します。



#### 〈備考〉

- 受信ハンドラは、最も古い受信データを取り込みます。「4.6 送信バッファ、受信バッファ」参照) データ取込み後に未取込みの受信データがある場合、システムレジスタ S610 は OFF せず、未取込みデータがなくなったらシステムレジスタ S610 を OFF します。
- 受信演算ファンクションは、1 ブロックの受信データ (1 つの受信バッファ内データ) を一括して読み込みます。  
例えば、外部機器から下記データを受信した場合の例を示します。

▼A B C D E F 0 1 2 3▼

 10文字受信

受信ハンドラ 起動時の取込み語数	取込みデータ
7	▼A B C D E F 0▼

7文字を取込んだ後、再び受信ハンドラを起動すると、未取込みの“123”は無視し、次のブロックの受信データ (次の受信バッファ内データ) を取込みます。

また、取込み語数が受信語数より大きい場合、余りのエリアに0を書き込みます。

## RV1 チャンネル1受信演算ファンクション

## 〈機能〉

パラメータによって指定されたエリアへ受信データを指定語数分転送します。受信データなしの場合は、何もしません。(外部機器リンクチャンネル1)

## 〈キー入力手順〉

FUNC R V 1 シフト # 転送アドレス 転送語数

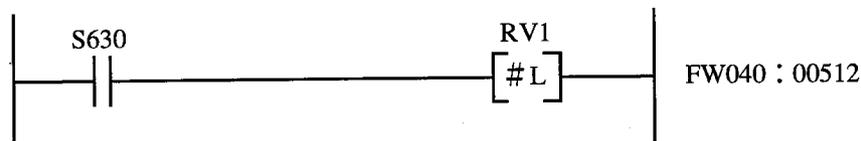
## 〈リターンコード〉

リターンコード (エラー情報) は、システムレジスタS630～S63Fに格納します。

(「4.8 受信情報」参照)

## 〈プログラム例〉

入力条件S630がON (チャンネル1受信データあり) のとき、受信データをFW040から512バイト (16進: 100h) 転送します。



## 〈備考〉

- 受信ハンドラは、最も古い受信データを取り込みます。「4.6 送信バッファ、受信バッファ」参照) データ取込み後に未取込みの受信データがある場合、システムレジスタS630はOFFせず、未取込みデータがなくなったらシステムレジスタS630をOFFします。
- 受信演算ファンクションは、1ブロックの受信データ (1つの受信バッファ内データ) を一括して読み込みます。

例えば、外部機器から下記データを受信した場合の例を示します。

▼A B C D E F 0 1 2 3▼ 10文字受信

受信ハンドラ 起動時の取込み語数	取込みデータ
7	▼A B C D E F 0▼

7文字を取込んだ後、再び受信ハンドラを起動すると、未取込みの“123”は無視し、次のブロックの受信データ (次の受信バッファ内データ) を取込みます。

また、取込み語数が受信語数より大きい場合、余りのエリアに0を書き込みます。

## 4 利用の手引き

### STASK0 チャンネル0送信サブルーチン

〈機能〉

パラメータによって指定されたエリアから指定語数分を外部機器へ送信します。(外部機器リンクチャンネル0)

〈リンク手順〉

C言語	アセンブラ言語
⋮	⋮
long (*stask0)( );	move.l #sbyte, -(A7)
long rtn, sadr, sbyte ;	move.l #sadr, -(A7)
⋮	lea \$107000, A0
staskm = 0x1070001 ;	jsr (A0)
⋮	addq.l #8, A7
rtn = (*stask0)(sadr, sbyte);	⋮
⋮	⋮

(注) アセンブラ言語の場合、D0レジスタ (リターンコード格納) 以外のレジスタの内容は、保証します。(C言語の場合は、特にレジスタを意識する必要はありません。)

〈パラメータ〉

sadr : 送信データ格納エリアアドレス

sbyte : 送信バイト語数

rtn : リターンコード

(注) アセンブラ言語の場合、リターンコードは、D0レジスタに格納します。

〈リターンコード〉

=0 : 正常終了

=FFFFFFFFh : 送信ハンドラ起動異常

( エラー情報は、S600~S60FとE1F40h番地にワードデータで格納します。  
(「4.7 送信情報」参照)

(注) long (\*f) ( ) ; 倍精度整数へのポインタの関数値として返す関数fの宣言。



・外部機器リンクモジュール未実装時、上記リターンコードは、不定です。

ただし、エラー情報は S600~S60F と E1F40h 番地にワードデータで格納します。(「4.7 送信情報」参照)

## 〈C言語の例〉

- ・ 送信用システムレジスタS600（アドレス E1F40h番地）の最上位ビット（送信可フラグ）をチェックし、送信可能ならば、アドレス140000h番地の送信データエリアの32バイト（16進：20h）を送信します。

```

      :
      register long (*stask0)( );
      register long rtn ;
      :
      if ( (*(short*)0xE1F401 & 0x8000) == 0 )
      {
          staskm = 0x1070001 ;
          rtn = (*stask0)(0x1400001, 0x201) ;
          if ( rtn != 0 )
              goto errb ;
      }
      else
      {
          :
      }

```

## 〈アセンブラ言語の例〉

- ・ 送信用システムレジスタS600（アドレス E1F40h番地）の最上位ビット（送信可フラグ）をチェックし、送信可能ならば、アドレス150000h番地の送信データエリアの256バイト（16進：100h）を送信します。

```

      :
      btst    #7, $E1F40 -----> 送信不可ならばLB1へ
      bne     LB1
      move.l  #$100, -(A7) -----> 転送語数256バイト
      move.l  #$150000, -(A7) -----> 送信データエリア150000h番地
      lea    $107000, A0
      jsr    (A0)
      addq.l #8, A7
      tst.l  D0
      bne    ERRB -----> 送信エラーならばERRBへ
      :

```

## 4 利用の手引き

### STASK1 チャンネル1送信サブルーチン

#### 〈機能〉

パラメータによって指定されたエリアから指定語数分を外部機器へ送信します。(外部機器リンクチャンネル0)

#### 〈リンク手順〉

C言語	アセンブラ言語
⋮	⋮
long (*stask1)( );	move.l #sbyte, -(A7)
long rtn, sadr, sbyte ;	move.l #sadr, -(A7)
⋮	lea \$107006, A0
stasks = 0x1070061 ;	jsr (A0)
⋮	addq.l #8, A7
rtn = (*stask1)(sadr, sbyte);	⋮
⋮	⋮

(注) アセンブラ言語の場合、D0レジスタ(リターンコード格納)以外のレジスタの内容は、保証します。(C言語の場合は、特にレジスタを意識する必要はありません。)

#### 〈パラメータ〉

sadr : 送信データ格納エリアアドレス

sbyte : 送信バイト語数

rtn : リターンコード

(注) アセンブラ言語の場合、リターンコードは、D0レジスタに格納します。

#### 〈リターンコード〉

=0 : 正常終了

=FFFFFFFFh : 送信ハンドラ起動異常

( エラー情報は、S620~S62FとE1F44h番地にワードデータで格納します。  
(「4.7 送信情報」参照)



・外部機器リンクモジュール未実装時、上記リターンコードは、不定です。

ただし、エラー情報はS620~S62FとE1F44h番地にワードデータで格納します。(「4.7 送信情報」参照)

## 〈C言語の例〉

- ・ 送信用システムレジスタS620（アドレス E1F44h番地）の最上位ビット（送信可フラグ）をチェックし、送信可能ならば、アドレス148000h番地の送信データエリアの16バイト（16進：10h）を送信します。

```

      :
      register long (*stask1)( );
      register long rtn ;
      :
      if ( (*(short*)0xE1F441 & 0x8000) == 0 )
      {
          stasks = 0x1070061 ;
          rtn = (*stask1)(0x1480001, 0x101);
          if ( rtn != 0)
              goto errb ;
      }
      else
      {
          :
      }

```

## 〈アセンブラ言語の例〉

- ・ 送信用システムレジスタS620（アドレス E1F44h番地）の最上位ビット（送信可フラグ）をチェックし、送信可能ならば、アドレス158000h番地の送信データエリアの40バイトを送信します。

```

      :
      btst    #7, $E1F44
      bne     LB1 -----> 送信不可ならばLB1へ
      move.l  #$40, -(A7) -----> 40バイト
      move.l  #$158000, -(A7) -----> 158000h番地
      lea    $107006, A0
      jsr    (A0)
      addq.l #8, A7
      tst.l  D0
      bne    ERRB -----> 送信エラーならばERRBへ
      :

```

## 4 利用の手引き

### RTASK0 チャネル0受信サブルーチン

〈機能〉

パラメータによって指定されたエリアへ受信データを語数分転送します。受信データなしの場合は何もしません。(外部機器リンクチャンネル0)

受信ハンドラ(サブルーチン)は最も古い受信データを取込みます。

〈リンク手順〉

C言語	アセンブラ言語
↓	↓
long (*rtask0)( );	move.l #rbyte, -(A7)
long rtn, radr, rbyte;	move.l #radr, -(A7)
↓	lea \$10700C, A0
staskm = 0x10700C1;	jsr (A0)
↓	addq.l #8, A7
rtn = (*rtask0)(radr, rbyte);	↓
↓	↓

(注) アセンブラ言語の場合、D0レジスタ(リターンコード格納)以外のレジスタの内容は、保証します。(C言語の場合は、特にレジスタを意識する必要はありません。)

〈パラメータ〉

radr : 受信データ格納エリアアドレス

rbyte : 受信バイト語数

rtn : リターンコード

(注) アセンブラ言語の場合、リターンコードは、D0レジスタに格納します。

〈リターンコード〉

=0 : 正常終了

(データ取込み後、未取込みのデータがあるとき、システムレジスタの受信データありビット(S610, E1F42h番地内の2<sup>15</sup>ビット)は、受信データありのままです。)

=1 : 受信バッファ内に受信データなし

=001A0000h : 受信データ取込み中バッファ内に、テキストの最終データが現われた。

(エンドコードが現われた。  
または、LGBのテキスト語数分取込んだ。)

=001A00xxh : 受信データ取込み中バッファ内に、受信エラー発生データが現われた。エラー発生データ以降、受信バイト数まで、受信データ格納エリアをゼロクリアします。リターンコード中のxxは、受信エラーコード下位バイトのエラーコードを表します。

(「4.12 受信エラーコード表」参照) また、エラー情報をS610~S61FとE1F42h番地にワードデータで格納します。(「4.8 受信情報」参照)

=FFFFFFFFh : 受信ハンドラ起動異常

(エラー情報は、S610~S61FとE1F42h番地にワードデータで格納します。  
。「4.8 受信情報」参照)



・外部機器リンク未実装時、上記リターンコードは、不定です。

ただし、エラー情報はS610~S61FとE1F42h番地にワードデータで格納します。(「4.8 受信情報」参照)

## 〈C言語の例〉

- 受信システムレジスタS610（アドレス E1F42h番地）の最上位ビット（受信完了フラグ）をチェックし、受信データありならば、受信データバッファの内容をアドレス140000h番地の受信データエリアに20バイト（16進：14h）を転送します。

```

:
register long (*rtask0)( );
register long rtn;
:
if ((*(short*)0xE1F421 & 0x8000) != 0)
{
    rtaskm = 0x10700C1;
    rtn = (*rtask0)(0x1400001, 0x141);
    if (rtn != 01)
        goto errb;
}
else
{
    :
}

```

- 受信Cモードサブルーチンの場合、1ブロックの受信データ（1つの受信、バッファ内データ）をアプリケーションプログラムに応じて、分割して読み込むことができます。（受信演算ファンクションは、1ブロックを一括して読み込みます。）外部機器から下記データを受信した場合の例を以下に示します。

"1234567890"
--------------

10キャラクタ受信

受信ハンドラ 起動時の取込み語数	リターンコード	取込みデータ
3	0 (正常)	"123"
4	0 (正常)	"4567"
4	001A0000h (正常、ブロック終了)	"890" 0

リターンコード（ロング長）の上位ワードにEOFコード（001Ah）が存在するとそのブロックの終了を意味します。またそのとき、下位ワードが0ならばそのブロックは正常受信を意味し、≠0の場合は異常受信を意味します。（そこまでのデータを正常受信したことを意味します。）またその異常受信のコードは、受信エラーコードの下位バイトと同じです。

ただし、実際のブロックより大きく取込み語数を設定した場合、余りのエリアに0を書込みます。

## 4 利用の手引き

---

### RTASK0 チャンネル0受信サブルーチン

〈アセンブラ言語の例〉

- ・ 受信用システムレジスタS610（アドレス E1F42h番地）の最上位ビット（受信完了フラグ）をチェックし、受信データありならば、受信データバッファの内容をアドレス150000h番地の受信データエリアに256バイト（16進：100h）を転送します。

⋮

```
btst    #7, $E1F42
beq     LB1 -----> 受信データなしならばLB1へ
move.l  #$100, -(A7) -----> 256バイト
move.l  #$150000, -(A7) -----> 150000h番地から
lea     $10700C, A0
jsr     (A0)
addq.l  #8, A7
tst.l   D0
bne     ERRB -----> 受信エラーならばERRBへ
```

⋮

## RTASK1 チャネル1受信サブルーチン

## 〈機能〉

パラメータによって指定されたエリアへ受信データを語数分転送します。受信データなしの場合は何もしません。(外部機器リンクチャンネル1)

受信ハンドラ(サブルーチン)は最も古い受信データを取込みます。

## 〈リンク手順〉

C言語	アセンブラ言語
⋮	⋮
long (*rtask1)( );	move.l #rbyte, -(A7)
long rtn, radr, rbyte ;	move.l #radr, -(A7)
⋮	lea \$107012, A0
stasks = 0x1070121 ;	jsr (A0)
⋮	addq.l #8, A7
rtn = (*rtask1)(radr, rbyte);	⋮
⋮	⋮

(注) アセンブラ言語の場合、D0レジスタ(リターンコード格納)以外のレジスタの内容は、保証します。(C言語の場合は、特にレジスタを意識する必要はありません。)

## 〈パラメータ〉

radr : 受信データ格納エリアアドレス

rbyte : 受信バイト語数

rtn : リターンコード

(注) アセンブラ言語の場合、リターンコードは、D0レジスタに格納します。

## 〈リターンコード〉

=0 : 正常終了

(データ取込み後、未取込みのデータがあるとき、システムレジスタの受信データありビット(S630, E1F46h番地内の2<sup>15</sup>ビット)は、受信データありのままです。)

=1 : 受信バッファ内に受信データなし

=001A0000h : 受信データ取込み中バッファ内に、テキストの最終データが現われた。

(エンドコードが現われた。  
または、LGBのテキスト語数分取込んだ。)

=001A00xxh : 受信データ取込み中バッファ内に、受信エラー発生データが現われた。エラー発生データ以降、受信バイト数まで、受信データ格納エリアをゼロクリアします。リターンコード中のxxは、受信エラーコード下位バイトのエラーコードを表します。

(「4.12 受信エラーコード表」参照) また、エラー情報をS630~S63FとE1F46h番地にワードデータで格納します。(「4.8 受信情報」参照)

=FFFFFFFFh : 受信ハンドラ起動異常

(エラー情報は、S630~S63FとE1F46h番地にワードデータで格納します。  
す。  
(「4.8 受信情報」参照)



・外部機器リンクモジュール未実装時、上記リターンコードは、不定です。  
ただし、エラー情報はS630~S63FとE1F46h番地にワードデータで格納します。  
(「4.8 受信情報」参照)

## 4 利用の手引き

### RTASK1 チャネル1受信サブルーチン

〈C言語リンケージ例〉

- 受信システムレジスタS630（アドレス E1F46h番地）の最上位ビット（受信完了フラグ）をチェックし、受信データありならば、受信データバッファの内容をアドレス 160000h番地の受信データエリアに60バイト（16進：3Ch）を転送します。

```

        :
register long  (*rtask1)( );
register long  rtn;
        :
if ( (*(short*)0xE1F461 & 0x8000) != 0 )
{
    rtaskm = 0x1070121;
    rtn = (*rtask0)(0x1600001, 0x3C1);
    if ( rtn != 0 )
        goto errb;
}
else
{
    :
}

```

- 受信Cモードサブルーチンの場合、1ブロックの受信データ（1つの受信、バッファ内データ）をアプリケーションプログラムに応じて、分割して読み込むことができます。（受信演算ファンクションは、1ブロックを一括して読み込みます。）外部機器から下記データを受信した場合の例を以下に示します。

"1234567890"

10文字受信

受信ハンドラ 起動時の取込み語数	リターンコード	取込みデータ
3	0 (正常)	"123"
4	0 (正常)	"4567"
4	001A0000h (正常、ブロック終了)	"890" 0

リターンコード（ロング長）の上位ワードにEOFコード（001Ah）が存在するとそのブロックの終了を意味します。またそのとき、下位ワードが0ならばそのブロックは正常受信を意味し、≠0の場合は異常受信を意味します。（そこまでのデータを正常受信したことを意味します。）またその異常受信のコードは、受信エラーコードの下位バイトと同一です。

ただし、実際のブロックより大きく取込み語数を設定した場合、余りのエリアに0を書込みます。

## 〈アセンブラリンクージ例〉

- ・ 受信用システムレジスタS630（アドレス E1F46h番地）の最上位ビット（受信完了フラグ）をチェックし、受信データありならば、受信データバッファの内容をアドレス170000h番地の受信データエリアに512バイト（16進：200h）転送します。

```

      ⋮
btst   #7, $E1F46
beq    LB1 -----> 受信データなしのときLB1へ
move.l #$200, -(A7) -----> 512バイト
move.l #$170000, -(A7) -----> 170000h番地から
lea    $107012, A0
jsr    (A0)
addq.l #8, A7
tst.l  D0
bne    ERRB -----> 受信エラーならばERRBへ
      ⋮
```

## 4 利用の手引き

### 4.10.3 ソフトウェアによるハードウェア制御

送信ハンドラのパラメータに以下に示すデータを設定することで、RS-232Cモジュールのハードウェアを制御できます。

- 演算ファンクションを使う場合  
送信ハンドラ名称：SD? (?はチャンネルナンバ)
- サブルーチンを使う場合  
送信ハンドラ名称：STASK? (?はチャンネルナンバ) (※3)

ただし、ソフトウェアによるハードウェア制御を行う場合は、CPUのキースイッチを“PROT. OFF”の状態で使用してください。“PROT. ON”の状態ですoftwareリセット以外の制御を行うとプロテクトエラーが発生します。

転送アドレス	転送語数	内 容	リターン内容								
・演算ファンクションのとき データレジスタDWFFF	8080H	ソフトウェアリセット  CPUのリセットスイッチによるRS-232Cモジュールのリセットと同じ	正常終了後は、CPUのLEDに正常動作中のメッセージを出力します。 (「4.13 CPU LED表示メッセージ表」参照)								
・Cモードサブルーチンのときアドレス 62FFEH	0000H または 0001H	最新ハードウェア状態取込み要求    0 : ON (※1) = 1 : OFF (※2) =  0 : OFF (※2) = 1 : ON (※1) =	<ul style="list-style-type: none"> <li>・チャンネルナンバ#0のとき DWFFFの上位バイト (62FFE<sup>H</sup>)</li> <li>・チャンネルナンバ#1のとき DWFFFの下位バイト (62FFF<sup>H</sup>)</li> <li>・チャンネルナンバ#2のとき DWFFEの上位バイト (62FFC<sup>H</sup>)</li> <li>・チャンネルナンバ#3のとき DWFFFの下位バイト (62FFD<sup>H</sup>)</li> </ul> <div style="text-align: center;"> <math>2^7</math> <span style="float: right;"><math>2^0</math></span> <table border="1" style="margin: auto;"> <tr> <td>①</td><td>②</td><td>③</td><td>④</td><td></td><td>⑤</td><td></td><td>⑥</td> </tr> </table> </div> <ul style="list-style-type: none"> <li>① RS出力状態</li> <li>② CS入力状態</li> <li>③ CD入力状態</li> <li>④ ER出力状態</li> <li>⑤ 送信ゲート状態</li> <li>⑥ DR入力状態</li> </ul>	①	②	③	④		⑤		⑥
①	②	③	④		⑤		⑥				



- (※1) 回線 High レベル、または送信ゲート OPEN を表します。  
 (※2) 回線 Low レベル、または送信ゲート CLOSE を表します。  
 (※3) タスクシステムの場合要求発行から各処理が起動されるまで最大 10 ミリ秒の遅れが生じるのでマクロ命令 delay などを使用し、十分余裕を持ってリターン内容を参照してください。

転送アドレス	転送語数	内 容	リターン内容
・演算ファンク ションのとき データレジス タDWFFF ・Cモードサブ ルーチンのと きアドレス 62FFEH	0100H	送信ゲートCLOSE要求	・チャンネルナンバ#0のとき DWFFFの上位バイト (62FFEH) ・チャンネルナンバ#1のとき DWFFFの下位バイト (62FFFH)
	0101H	送信ゲートOPEN要求	・チャンネルナンバ#2のとき DWFFEの上位バイト (62FFCH) ・チャンネルナンバ#3のとき DWFFFの下位バイト (62FFDH)
	0200H	DR OFF要求 (※1)	/00……OFF報告 /01……ON報告
	0201H	DR ON要求 (※2)	
	0300H	RS OFF要求 (※1)	・チャンネルナンバ#0のとき DWFFFの上位バイト (62FFEH) ・チャンネルナンバ#1のとき DWFFFの下位バイト (62FFFH) ・チャンネルナンバ#2のとき DWFFEの上位バイト (62FFCH)
	0301H	RS ON要求 (※2)	・チャンネルナンバ#3のとき DWFFEの下位バイト (62FFDH) /00……ON報告 /01……OFF報告

なお、送信パラメータアドレスをDWFFF、/62FFEにした場合、パラメータ語数が上記表示以外の場合  
は、リターン内容を/FFにしてリターンします。



(※1) 回線 (端子) は High レベル (OFF) となります。

(※2) 回線 (端子) は Low レベル (ON) となります。

## 4 利用の手引き

### 4.11 送信エラーコード表

送信に関するエラーコードを示します。エラーコード（上位バイト、下位バイト）はSレジスタと同じです。またエラー発生時、ハンドラエラー以外はCPUのコンソールLEDに表示します。

モジュール	LED表示 上位4文字	上位 バイト	下位 バイト	内 容	
チャンネル0	LED表示しません。	40h	01h	送信起動すべきモジュールが未実装。	
			02h	送信中断中に送信起動を行った。	
			03h	送信中に送信起動を行った。	
			04h	送信ハンドラパラメータエラー。	
			05h	送信起動すべきモジュールがダウン中。	
	'GEOS'	20h	80h	他局優先中送信不可能。 他局優先指定で現在データ受信中のため、送信不可能。	
			81h	送信中送信不可能。 先に送信要求のあった送信データを現在送信中（送信中断中も含む）のため、送信不可能。	
			82h	送信要求なし中送信不可能。 CS（送信要求）入力が出送信要求なし、またはDR（データセットレディ）チェック指定でDR入力がノットレディのため、送信不可能。	
			83h	送信中断タイムアウト。 送信中に送信中断コードにより送信中断され、送信中断監視時間内に送信再開コードにより送信再開されなかった。	
			84h	送信要求タイムアウト。 送信中にCS（送信要求）入力が出送信要求なしとなり送信が中断され、送信中断監視時間内にCS入力が送信要求ありにならず、送信再開されなかった。	
			85h	データセットレディタイムアウト。 DR（データセットレディ）チェック指定で、送信中にDR入力がノットレディとなり送信が中断され、送信中断監視時間内にDR入力がレディにならず、送信再開されなかった。	
			10h	×× (不定)	受信打切り。 自局優先指定で、データ受信中に送信起動されたため、データ受信を打切りデータ送信を開始した。

モジュール	LED表示 上位4文字	上位 バイト	下位 バイト	内 容	
チャンネル1	LED表示しません。	40h	01h	送信起動すべきモジュールが未実装。	
			02h	送信中断中に送信起動を行った。	
			03h	送信中に送信起動を行った。	
			04h	送信ハンドラパラメータエラー。	
			05h	送信起動すべきモジュールがダウン中。	
	'GEIS'	20h	80h	他局優先中送信不可能。 他局優先指定で現在データ受信中のため、送信不可能。	
			81h	送信中送信不可能。 先に送信要求のあった送信データを現在送信中（送信中 中断も含む）のため、送信不可能。	
			82h	送信要求なし中送信不可能。 CS（送信要求）入力が送信要求なし、またはDR（デー タセットレディ）チェック指定でDR入力がノットレディ のため、送信不可能。	
			83h	送信中断タイムアウト。 送信中に送信中断コードにより送信中断され、送信中断 監視時間内に送信再開コードにより送信再開されなかつ た。	
			84h	送信要求タイムアウト。 送信中にCS（送信要求）入力が送信要求なしとなり送信 が中断され、送信中断監視時間内にCS入力が送信要求あ りにならず、送信再開されなかった。	
			85h	データセットレディタイムアウト。 DR（データセットレディ）チェック指定で、送信中に DR入力がノットレディとなり送信が中断され、送信中断 監視時間内にDR入力がレディにならず、送信再開されな かった。	
			10h	×× (不定)	受信打ち切り。 自局優先指定で、データ受信中に送信起動されたため、 データ受信を打ち切りデータ送信を開始した。

## 4.12 受信エラーコード表

受信に関するエラーコードを示します。エラーコード（上位バイト、下位バイト）はSレジスタと同じです。またエラー発生時、ハンドラエラー以外や、CPUのコンソールLEDに表示します。

なお、システムエラー上位バイトで10hのエラーが発生するとその外部機器リンクモジュールはダウン状態となり、送信起動、外部機器から受信は行われません。CPUのリセットスイッチにてリセットし、再度立上げてください。

モジュール	LED表示 上位4文字	上位 バイト	下位 バイト	内 容	
チャンネル0	LED表示し ません。	41h	×× (不定)	受信ハンドラ起動時、モジュール未実装。	
		42h	×× (不定)	受信ハンドラパラメータエラー。	
		43h	×× (不定)	受信バッファ異常。	
		44h	×× (不定)	モジュールダウン時受信ハンドラ起動システムエラー（上位バイトが10hのエラー）発生時に、受信ハンドラを起動した。	
	'GEOR'	20h	80h		パリティエラー。受信データでパリティエラー発生。
			81h		オーバーランエラー。受信データでオーバーランエラー発生。
			82h		フレーミングエラー。受信データでフレーミングエラー発生。
			83h		受信タイムアウト。指定受信監視時間内で全データ受信できなかった。
			84h		ASCII変換エラー。ASCII変換指定時 '0' ~ '9' および 'A' ~ 'F' 以外のデータを受信した。
			85h		エンドコードエラー。 (注) ASCII変換指定時、'0' ~ '9' および 'A' ~ 'F' 以外データまたはエンドコード以外のデータを受信した。
			86h		BCCエラー。BCC受信時、BCCが不一致。
			87h		受信キャリア検出タイムアウト。 CD（受信キャリア検出）入力がキャリアなしとなり、受信中断となったが、受信監視時間内に、CD入力がキャリアありとならず受信再開されなかった。
			02h		受信打ち切り発生。 自局優先指定でデータ受信中に送信起動されたため、途中までしかデータ受信していないバッファありを示す。
			00h		受信バッファ異常。
	'GEOO'	10h	F0h		ROMチェックサムエラー。GR時にROMチェックサムエラー発生。
			F1h		RAMクリア。ROMアイドル中にRAMがすべてクリアされた。
			F2h		RAM未ローディング。RAMプログラムがなし。
			F3h		RAMチェックサムエラー。GR時にRAMチェックサムエラー発生。
			F4h		RAMダウン。RAMプログラム破壊などによりRAMプログラムが停止した。
			F5h		RAMパリティエラー。RAMエリアでパリティエラー発生。
	'GODN'	10h	FFh	WDTエラー。ウォッチドックタイマタイムアウトエラー発生。	

モジュール	LED表示 上位4文字	上位 バイト	下位 バイト	内 容	
チャンネル1	LED表示し ません。	41h	×× (不定)	受信ハンドラ起動時、モジュール未実装。	
		42h	×× (不定)	受信ハンドラパラメータエラー。	
		43h	×× (不定)	受信バッファ異常。	
		44h	×× (不定)	モジュールダウン時、受信ハンドラ起動システムエラー（上位バイトが10hのエラー）発生し、受信ハンドラを起動した。	
	'GE1S'	20h	80h		パリティエラー。受信データでパリティエラー発生。
			81h		オーバーランエラー。受信データでオーバーランエラー発生。
			82h		フレーミングエラー。受信データでフレーミングエラー発生。
			83h		受信タイムアウト。指定受信監視時間内で全データを受信できなかった。
			84h		ASCII変換エラー。ASCII変換指定時 '0' ~ '9' および 'A' ~ 'F' 以外のデータを受信した。
			85h		エンドコードエラー。（注） ASCII変換指定時、'0' ~ '9' および 'A' ~ 'F' 以外データまたはエンドコード以外のデータを受信した。
			86h		BCCエラー。BCC受信時、BCCが不一致。
			87h		受信キャリア検出タイムアウト。 CD（受信キャリア検出）入力がキャリアなしとなり、受信中断となったが、受信監視時間内に、CD入力がキャリアありとならず受信再開されなかった。
			02h		受信打ち切り発生。 自局優先指定でデータ受信中に送信起動されたため、途中までしかデータ受信していないバッファありを示す。
			00h		受信バッファ異常。
	'GE1O'	10h	F0h		ROMチェックサムエラー。GR時にROMチェックサムエラー発生。
			F1h		RAMクリア。ROMアイドル中にRAMがすべてクリアされた。
			F2h		RAM未ローディング。RAMプログラムがなし。
			F3h		RAMチェックサムエラー。GR時にRAMチェックサムエラー発生。
			F4h		RAMダウン。RAMプログラム破壊などによりRAMプログラムが停止した。
			F5h		RAMパリティエラー。RAMエリアでパリティエラー発生。
'G1DN'	10h	FFh	WDTエラー。ウォッチドックタイマタイムアウトエラー発生。		

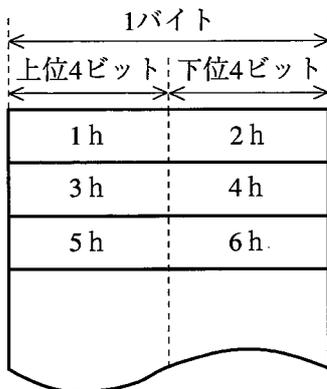
## 4 利用の手引き



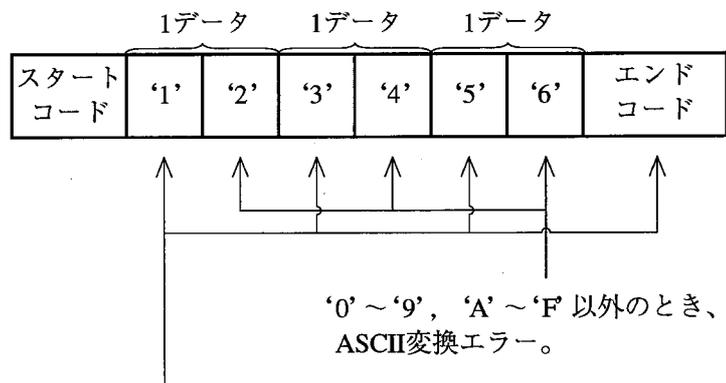
- 外部機器からデータ 12h, 34h, 56h を外部機器リンクに送信したとき、各データの下位ビットデータが '0' ~ '9', 'A' ~ 'F' 以外のとき（回線上のデータ化は、バイナリ→ASCII 変換ミスなどにより）ASCII 変換エラーとなります。また、上位4ビットデータが '0' ~ '9', 'A' ~ 'F' 以外またはエンドコード以外のとき（回線上のデータ化は、バイナリ→ASCII 変換ミス、エンドコードのアンマッチなどにより）エンドコードエラーとなります。

外部機器の送信データ

格納エリア



回線上のデータ



'0' ~ '9', 'A' ~ 'F' 以外、またはエンドコード以外のときエンドコードエラー。





# 5 オペレーション

## 5 オペレーション

### 5.1 システムを立上げるにあたり

#### 5.1.1 使い方

このシステムは、オプションモジュールの1つである「外部機器リンクモジュール」を容意に立上げるためのマンマシンツールです。

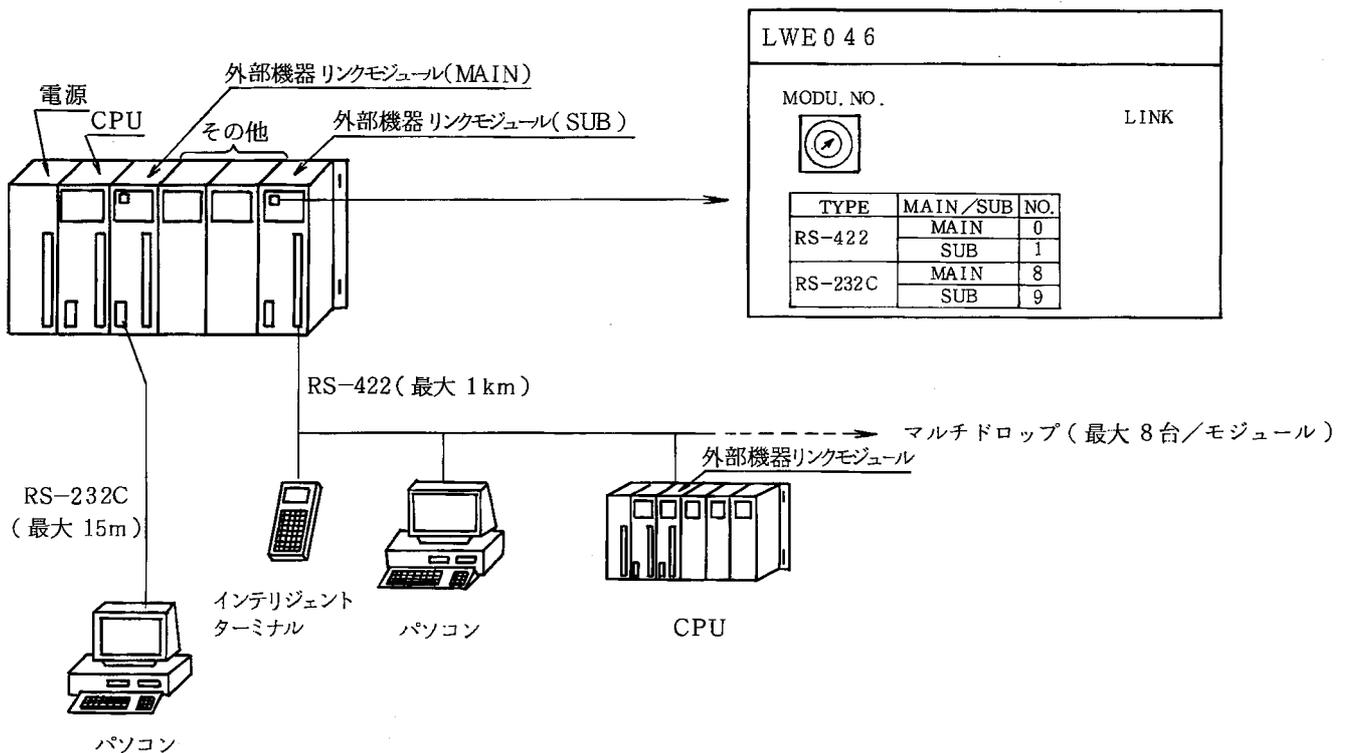
#### 5.1.2 外部機器リンクモジュール

外部機器リンクモジュールは、CPUとプリンタやパソコンなどの外部機器を接続するための機能を持ちます。

このモジュールは、CPU1台に最大2モジュールまで実装できます。

また外部機器リンクモジュールは、モジュールのロータリースイッチの設定によりRS-232Cまたは、RS-422のいずれかを選択できます。

#### 5.1.3 システム構成



外部機器リンクモジュールの設定によりMAIN (チャンネル0) とSUB (チャンネル1) のいずれかを選択できます。モジュールを2個使用する場合は、必ずMAINとSUBに区別してください。

## 5.1.4 基本オペレーション

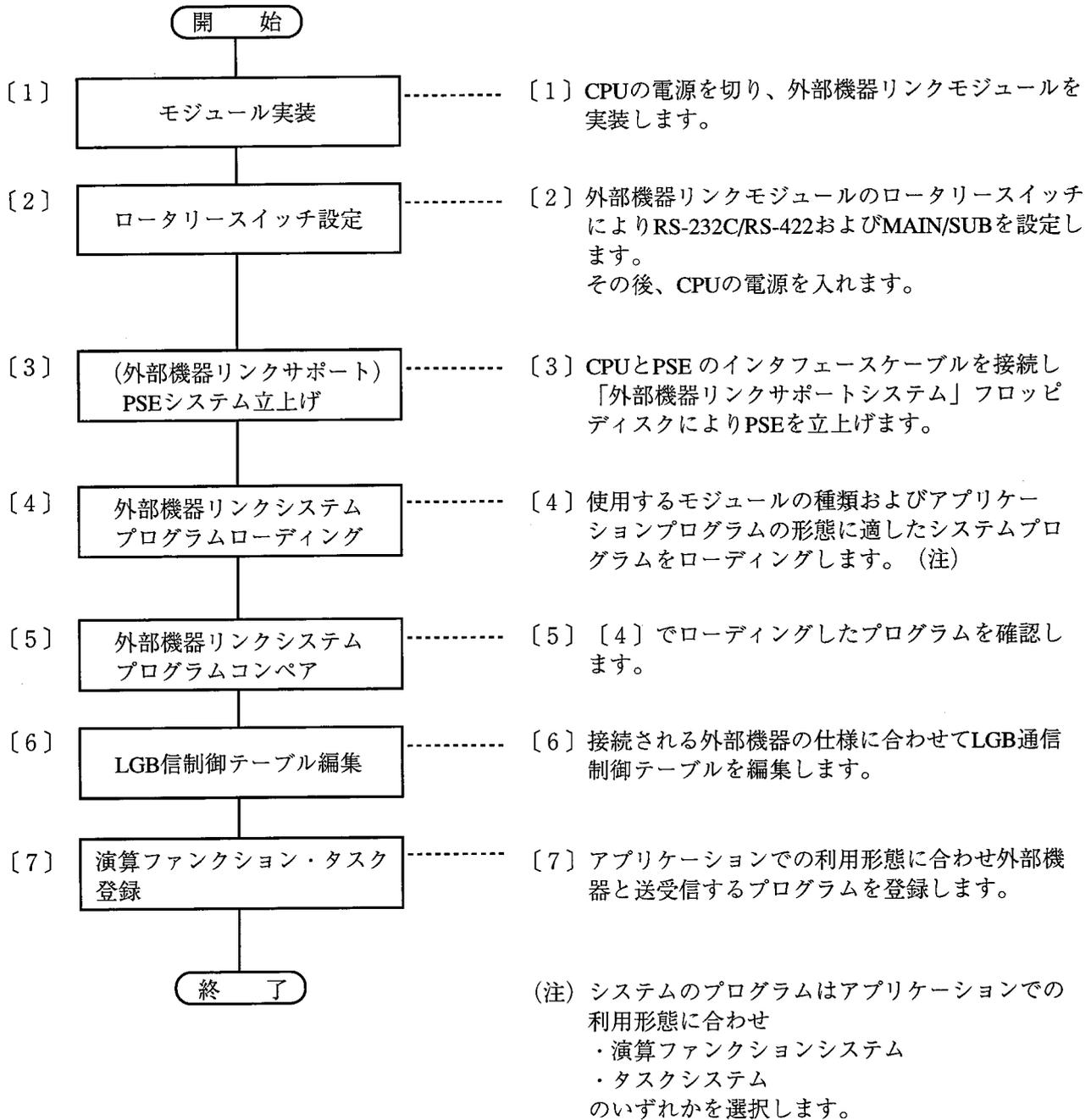
- 画面に表示されたカーソルにそって入力することにより、簡単に操作できます。
- 選択するオペレーションには、大きく分けて次の2種類があります。
  - ・ 選択項目のナンバを入力する。
  - ・ 設定キーまたは修正キーを選択して押す。
- 設定キーまたは修正キーを押す場合の操作  
画面に〔SET/RTY/CLS〕のように選択キーが表示される場合、それらのキーの意味は、次のようになっています。

表示画面名称	対応するキー	意味
SET	<input type="text" value="設定"/> キー	OKのとき
CLS	<input type="text" value="終了"/> キー	1つまたはそれ以上前の画面に戻す
RTY	<input type="text" value="再設定"/> キー	データを再設定するとき
CNT	<input type="text" value="続行"/> キー	処理を繰返すとき
DEL	<input type="text" value="削除"/> キー	ファイルなどを削除するとき

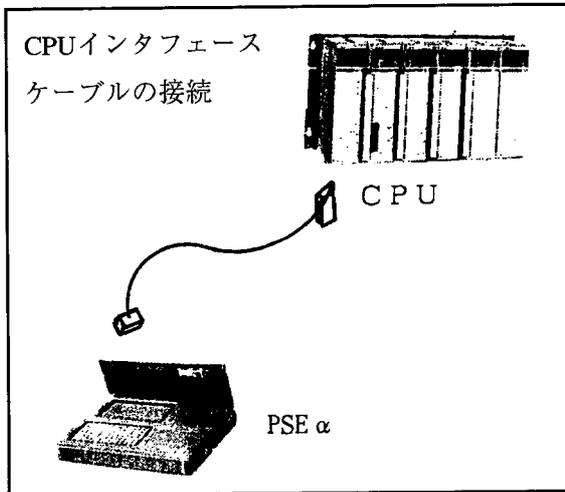
- (注) CLS : CLOSE  
 RTY : RETRY  
 CNT : CONTINUE  
 DEL : DELETE

## 5.2 システム立上げ

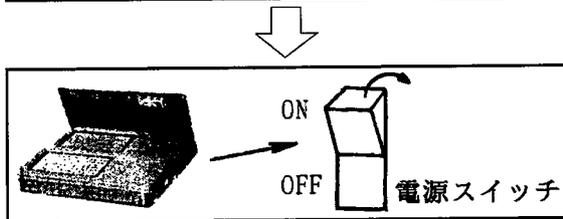
### 5.2.1 外部機器リンクシステム立上げ手順



## 5.2.2 PSEシステム立上げ手順



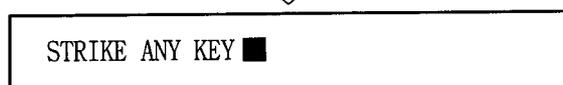
- ▶ PSE  $\alpha$  の電源がOFFの状態ではCPUと正しくインタフェースケーブルを接続します。このとき、CPUのコンソールスイッチは、ストップ (STOP) とし、メモリプロテクトスイッチは、プロテクトOFF (PROT.OFF) にセットします。



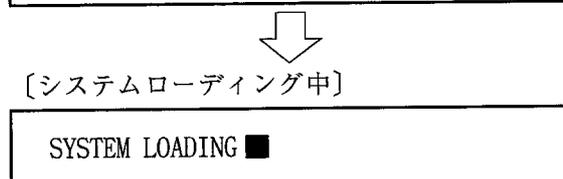
- ▶ PSE  $\alpha$  の電源をONしてください。



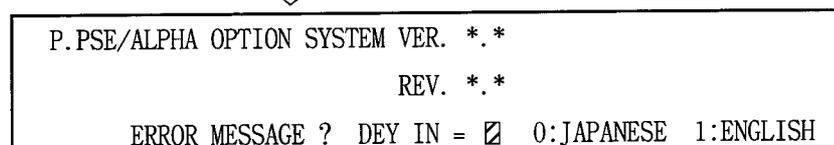
- ▶ システムフロッピーディスク「外部機器リンクプログラム」をPSE  $\alpha$  に挿入してください。



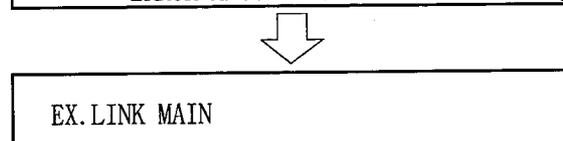
- ▶ PSE  $\alpha$  の画面上に、図のメッセージを表示します。任意のキーを入力してください。



- ▶ PSE  $\alpha$  は「SYSTEM LOADING」と表示し、フロッピーディスクから、システムプログラムをPSE  $\alpha$  メモリへローディング中です。



- ▶ エラーメッセージ表示について、和文  0 または英文  1 を選択してください。

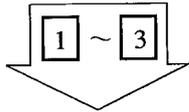


- ▶ 外部機器リンクサポートシステムメイン画面を表示します。

## 5 オペレーション

```

EX. LINK MAIN
*** EX. LINK SUPPORT SYSTEM (Ver*. Rev*.) ***
KEYIN CHANNEL No. = 
-----
SELECTION NUMBER
0 1 2 3
-----
    
```



▶ 外部機器リンクサポートシステムメイン画面

実装されている外部機器リンクのチャンネル番号を入力します。

0 ~  3

(例)  0 を入力

```

EX. LINK No. 0
*** EX. LINK SUPPORT SYSTEM (Ver*. Rev*.) ***
KEYIN MENU No. =  [CLS]
-----
CHANNEL No. 0 MENU
-----
1 : SYSTEM PROGRAM LOADING (FUNC)
2 :                               (TASK)
3 : EX. LINK LGB TABLE EDIT
4 : SYSTEM PROGRAM COMPARE (FUNC)
5 :                               (TASK)
6 : EX. LINK MCS
-----
    
```

▶ メニュー画面 (チャンネル番号0)

メニュー画面の内容

- |    |                            |
|----|----------------------------|
| 1  | : 演算ファンクションシステムをローディングします。 |
| 2  | : タスクシステムをローディングします。       |
| 3  | : LGBテーブルを編集します。           |
| 4  | : 演算ファンクションシステムをコンペアします。   |
| 5  | : タスクシステムをコンペアします。         |
| 6  | : モジュールの内容を読み出し/書込みします。    |
| 終了 | : メイン画面へ戻ります。              |

### 5.3 システムプログラムの書込み（ロード）

外部機器リンクプログラムフロッピーに格納されている「システムプログラム」をCPU本体および外部機器リンクモジュールへ書込み（ローディング）ます。



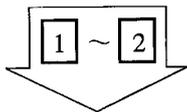
- ・タスクシステムのプログラムは、拡張メモリにもローディングされますので、タスクシステムを使用する場合（ユーザプログラムをCモードでつくる時）は、拡張メモリが実装されているか、もう一度確認してください。
- ・タスクシステムのプログラムと演算ファンクションシステムのプログラムは混在させないでください。（CPU停止となります。）ローディングするシステムプログラムはCPUごとに統一してください。
- ・タスクシステムから演算ファンクションシステム（またはその逆）へシステムプログラムを入換える場合、外部機器リンクプログラムフロッピーに格納されている‘EXCLR.PSE’をF/D処理でローディングした後、システムプログラムをローディングしてください。

## 5 オペレーション

### オペレーション

[1]

```
EX. LINK No. 0
*** EX. LINK SUPPORT SYSTEM (Ver*. * Rev*. *) ***
KEYIN MENU No. =  [CLS]
```



▶ メニュー画面から **1** または **2** を選択します。

**1** : 演算ファンクションシステム

**2** : タスクシステム

・ CPUをSTOP、プロテクトをOFFの状態にします。

[2]

```
EX. LINK OS LOAD
SET SYSTEM F/D & STRIKE [SET] KEY !!
EX. LINK NUMBER 0 ??? SYSTEM PROGRAM : Ver*. * Rev*. *
```



???? : 演算ファンクションシステム時 「FUNC」

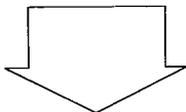
タスクシステム時

「TASK」

▶ 外部機器リンクサポートシステムフロッピーがPSE aに挿入されていることを確認後、**設定**キーを押してください。

[3]

```
EX. LINK OS LOAD
ADDRESS=/*****
SYSTEM PROGRAM LOADING
```



▶ システムプログラムのロードが開始されます。

[4]

```
*** SYSTEM PROGRAM LOAD OK !! ***
*** STRIKE [CLS] KEY !! ***
```



▶ 左図が表示されたらロード正常終了です。**終了**キーを押してください。

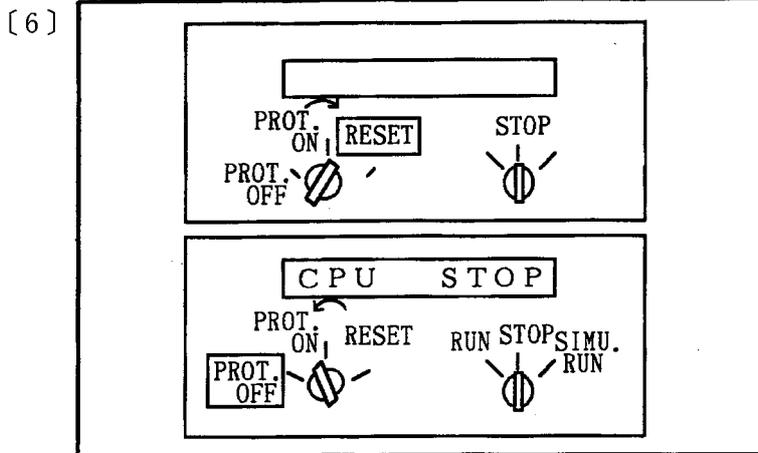
[5]

```
**** PLEASE PCS RESET !! ****
```

▶ 左図を表示し、CPUのリセット待ちになります。



・ システムプログラムがローディングされるとLGBは初期化されます。LGBを変更する場合は、LGBを編集してください。



- ▶ CPUコンソールスイッチをリセット (RESET) →プロテクトOFFとします。

メニュー画面 [1] へ戻ります。

[CPU LED表示について]

システムプログラムがローディングされるとCPU LEDに下記のメッセージを表示します。

RS☆☆?□0\*

- ・ ☆☆：バージョン、レビジョン
- ・ ? : 演算ファンクションシステム時 'F'  
タスクシステム時 'T'
- ・ \* :  $2^3$   $2^2$   $2^1$   $2^0$

I/O	I/O	I/O	I/O
-----	-----	-----	-----

- チャンネルナンバ0ローディング済/未済
- チャンネルナンバ1ローディング済/未済
- チャンネルナンバ2ローディング済/未済
- チャンネルナンバ3ローディング済/未済

- (例) チャンネルナンバ0と1をローディングしたとき '3'  
 チャンネルナンバ2と3をローディングしたとき 'C'  
 すべてローディングしたとき 'F'

5.4 システムプログラムの照合 (コンペア)

外部機器リンクプログラムフロッピーに格納されている「システムプログラム」とCPUおよび外部機器リンクモジュール各チャンネルのメモリ内容を照合 (コンペア) します。

システムローディング後や外部機器リンクモジュールの誤動作発生時に行ってください。

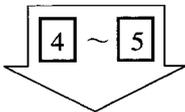
オペレーション

[1]

```
EX. LINK No. 0
*** EX. LINK SUPPORT SYSTEM (Ver*. * Rev*. *) ***
KEYIN MENU No. = ■ [CLS]
```

▶ メニュー画面から **4** または **5** を選択します。

- 4** : 演算ファンクションシステム
- 5** : タスクシステム



[2]

```
EX. LINK OS COMP
SET SYSTEM F/D & STRIKE [SET] KEY !!
EX. LINK NUMBER 0 ??? SYSTEM PROGRAM : Ver*. * Rev*. *
```

▶ 外部機器リンクサポートシステムフロッピーがPSE αに挿入されていることを確認後、**設定** キーを押してください。



??? : 演算ファンクションシステム時「FUNC」  
タスクシステム時「TASK」

[3]

```
EX. LINK OS COMP
COMPARE START ADDRESS=/*****
SYSTEM PROGRAM COMPARING
```

▶ システムプログラムのコンペアが開始されます。



[4]

```
*** SYSTEM PROGRAM COMPARE OK !! ***
*** STRIKE [CLS] KEY !! ***
```

▶ 左図が表示されたらコンペア正常終了です。**終了** キーを押してください。

不一致ありの場合の終了は左図の“OK”が“END”となります。



メニュー画面 [1] へ戻ります。

## 〔5〕 不一致ありの場合

```

STRIKE ANY KEY ■ [CNT/CLS]
COMPARE ERROR ON PCS MEMORY ADDRESS=/OFD000
***** PCS MEMORY DATA *****
/OFD000 0050 0000 0050 0000 0050 0000 0050 0000
/OFD010 0050 0000 0050 0000 0050 0000 0050 0000
/OFD020 0050 0000 0050 0000 0050 0000 0050 0000
/OFD030 0050 0000 0050 0000 0050 0000 0150 0000
/OFD040 0000 0000 0000 0000 0000 0000 0000 0000
/OFD050 0000 0000 0000 0000 0000 0000 0000 0000
/OFD060 0000 0000 0000 0000 0000 0000 0000 0000
/OFD070 0000 0000 0000 0000 0000 0000 0000 0000

```

① … 不一致となった最初のアドレスを示します。

② … 現在CPUおよび外部機器リンクモジュールのメモリ内容を表示していることを示します。

続	行
終	了
その他のキー	

… 次のアドレスから照合を再開する場合。

… 処理を終了する場合。(メニュー画面〔1〕へ戻ります。)

… 画面に表示する表示内容を切替える場合。

(CPUおよび外部機器リンクモジュールとフロッピーディスク (F/D) の内容)

### 5.5 LGB通信制御テーブル編集

LGB (Line Group Block) とは、通信制御プログラムが回線を通じて送受信を行うための伝送上の情報群を意味します。この情報はPSE $\alpha$ によりユーザが決定します。

LGBをどのように設定するかは非常に重要であり、外部機器とハードウェア的に接続できなかつたり、また、伝送手順がくいちがって正常に送受信できなかつたりすることがあります。

チャンネルナンバごとに、接続される外部機器の仕様に合った設定をしてください。

#### ● LGBに設定する内容

- ・ 伝送フレーム
- ・ 伝送速度
- ・ 優先制御
- ・ データ変換モード
- ・ テキスト語数
- ・ スタートコード
- ・ エンドコード
- ・ ブロックチェックキャラクタ
- ・ 送信遅延時間
- ・ 送信中断/再開コード
- ・ 送信中断監視時間
- ・ 受信監視時間
- ・ 送信要求
- ・ データ端末レディ
- ・ データセットレディ

#### ● システムモードに設定する内容

- ・ 演算ファンクションシステム時  
送/受信演算ファンクション (「5.7 UFET」参照)
- ・ タスクシステム時  
受信起動タスク (「5.8 受信起動タスク登録」参照)

## 5.5.1 設定内容

## (1) 伝送フレーム (DATA FRAME)

PSE選択項目		伝送フレーム内容	初期値
メニューNo.	表示内容		
0	ST+7DT+EP+2SP	ST 2 <sup>0</sup> _____ 2 <sup>6</sup> EP SP SP	
1	ST+7DT+OP+2SP	ST 2 <sup>0</sup> _____ 2 <sup>6</sup> OP SP SP	
2	ST+7DT+EP+1SP	ST 2 <sup>0</sup> _____ 2 <sup>6</sup> EP SP	
3	ST+7DT+OP+1SP	ST 2 <sup>0</sup> _____ 2 <sup>6</sup> OP SP	
4	ST+8DT+2SP	ST 2 <sup>0</sup> _____ 2 <sup>7</sup> SP SP	
5	ST+8DT+1SP	ST 2 <sup>0</sup> _____ 2 <sup>7</sup> SP	
6	ST+8DT+EP+1SP	ST 2 <sup>0</sup> _____ 2 <sup>7</sup> EP SP	
7	ST+8DT+OP+1SP	ST 2 <sup>0</sup> _____ 2 <sup>7</sup> OP SP	○



・表の記号は下記を示します。

ST : スタートビット

DT : データビット

EP : 偶数パリティビット

OP : 奇数パリティビット

SP : ストップビット

・パリティおよびストップビットは、次のように設定されることが多い。

パリティ : 偶数パリティ

ストップビット : 1ビット

## 5 オペレーション

### (2) 伝送速度 (BAUD RATE)

PSE選択項目		伝送レート内容	初期値
メニューNo.	表示内容		
0	150 [BPS]	150 [bps]	
1	300 [BPS]	300 [bps]	
2	600 [BPS]	600 [bps]	
3	1200 [BPS]	1,200 [bps]	
4	2400 [BPS]	2,400 [bps]	
5	4800 [BPS]	4,800 [bps]	○
6	9600 [BPS]	9,600 [bps]	
7	19200 [BPS]	使用しないでください。	



- ・ BPS：ビット/秒
- ・ 伝送速度は 1,200～9,600bps で使用することが多い。
- ・ 19200 [bps] を使用すると、通信時オーバーランエラーなどが発生します。ただし、発生する確率は低くリトライにより回避できます。

### (3) 優先制御 (PRIORITY LEVEL)

PSE選択項目		優先制御内容	初期値
メニューNo.	表示内容		
0	SELF	自局優先	○
1	OTHERS	他局優先	

### (4) データ変換モード (DATA CHANGE MODE)

PSE選択項目		伝送データ内容	初期値
メニューNo.	表示内容		
0	ASCII	テキストデータをASCII処理	
1	BINARY	テキストデータをBINARY処理	○

(注) データ変換モードは、BINARY (バイナリ) 指定を使用する場合が多い。

## (5) テキストサイズ (TEXT SIZE)

PSE選択項目		テキストサイズ	初期値
メニューNo.	表示内容		
0	NO TEXT	テキストなし	
1~512	001~512 [BYTE]	1~512 [バイト]	256

## (6) スタートコード (SCD : START CODE)

PSE選択項目			スタートコード内容	初期値	
メニューNo.	メニュー表示	スタートコード表示			
0	NO START CODE	NO START CODE	スタートコードなし	○	
1	1 START CODE	CD1	1スタートコード		CD=02h(STX)
2	2 START CODE	CD1+CD2	2スタートコード		
3	3 START CODE	CD1+CD2+CD3	3スタートコード		
4	4 START CODE	CD1+CD2+CD3+CD4	4スタートコード		



- ・ CD1~4 : 00h~FFh のスタートコードを示す 16 進。
- ・ スタートコード (SCD) は、STX (02h ; テキスト開始) を使用することが多い。

## (7) エンドコード (ECD : END CODE)

PSE選択項目			スタートコード内容	初期値	
メニューNo.	メニュー表示	エンドコード表示			
0	NO END CODE	NO END CODE	エンドコードなし		
1	1 END CODE	CD1	エンドコード	○	CD=03h(ETX)
2	2 END CODE	CD1+CD2	2エンドコード		
3	3 END CODE	CD1+CD2+CD3	3エンドコード		
4	4 END CODE	CD1+CD2+CD3+CD4	4エンドコード		

(注1) CD1~CD4 : 00h~FFhのエンドコードを示す16進。

(注2) エンドコード (ECD) は、ETX (03h ; テキスト終了) を使用することが多い。

## 5 オペレーション

### (8) ブロックチェックキャラクタ (BCC: BCC MODE)

PSE選択項目		BCC内容	初期値
メニューNo.	表示内容		
0	NO BCC	BCCなし	○
1	EVEN PARITY	水平偶数パリティチェック	
2	ODD PARITY	水平奇数パリティチェック	

### (9) 送信遅延時間 (SEND DELAY TIME)

PSE選択項目		送信遅延時間内容	初期値
設定値	表示内容		
0	NO DELAY	データ送信遅延なし	○
1~32767	00001~32767 [mSEC]	1~32767 (ミリ秒)	

#### (制限事項)

伝送レート	送信遅延時間設定範囲
150 [BPS]	128~32,767 (ミリ秒)
300 [BPS]	64~32,767 (ミリ秒)
600 [BPS]	32~32,767 (ミリ秒)
1200 [BPS]	16~32,767 (ミリ秒)
2400 [BPS]	8~32,767 (ミリ秒)
4800 [BPS]	4~32,767 (ミリ秒)
9600 [BPS]	2~32,767 (ミリ秒)

### (10) 送信中断/再開コード (SEND BREAK/CONTINUE)

PSE選択項目			中断/再開コード内容	初期値
メニューNo.	メニュー表示	中断/再開コード表示		
0	NO BREAK/CONT.	NO BREAK/CONTINUE	中断/再開コードなし	○
1	1BR+1CN	BR:CD1 CN:CD2	1中断, 1再開コード	
2	1BR+2CN	BR:CD1 CN:CD2+CD3	1中断, 2再開コード	
3	2BR+1CN	BR:CD1+CD2 CN:CD3	2中断, 1再開コード	
4	2BR+2CN	BR:CD1+CD2 CN:CD3+CD4	2中断, 2再開コード	



- ・BR: 中断コード。 CN: 再開コード。
- ・CD1~4: 00h~FFhの送信中断、再開コードを示す16進値。
- ・中断コードおよび再開コードについては、次のような設定が多い。  
 中断コード: 13h (DC3; 装置制御3 [X-OFF])  
 再開コード: 11h (DC1; 装置制御1 [X-ON])

## (11) 送信中断監視時間 (SEND BREAK TIMEOUT)

PSE選択項目		送信中断監視時間内容	初期値
設定値	表示内容		
0	NO TIMEOUT	テキスト送信中断監視なし	
1~32767	00001~32767 [100mSEC]	0.1~3276.7 (秒)	32767

## (12) 受信監視時間 (RECEIVE TIMEOUT)

PSE選択項目		受信監視時間内容	初期値
設定値	表示内容		
0	NO TIMEOUT	テキスト受信監視なし	
1~32767	00001~32767 [100mSEC]	0.1~3276.7 (秒)	32767

## (13) RS-422ゲートコントロール (RS-422 GATE CONTROL)

PSE選択項目		ゲートコントロール内容	初期値
メニューNo.	表示内容		
0	OPEN	送信ゲートOPEN	○
1	CONTROL	送信ゲートOPEN↔CLOSE	

## (14) 送信要求 (RS : REQUEST TO SEND)

PSE選択項目		送信要求内容	初期値
メニューNo.	表示内容		
0	LOW	送信要求出力	○
1	HIGH	送信要求無出力	

## (15) データ端末レディ (ER : EQUIPMENT READY)

PSE選択項目		データ端末レディ内容	初期値
メニューNo.	表示内容		
0	LOW	NOT READY出力	
1	HIGH	READY出力	○

## 5 オペレーション

### (16) データセットレディ (DR: DATA SET READY)

PSE選択項目		データセットレディ内容	初期値
メニューNo.	表示内容		
0	NO CHECK	チェックなし	○
1	CHECK	チェックあり	



- ・LGB に不正なデータが登録されている場合、PSE は内容を “\*\* BAD DATA \*\*” と表示します。このときは該当する項目を再設定してください。
- ・外部機器リンクモジュールは、LGB の編集だけでは動作しません。オペレーションに従い編集後必ず CPU をリセットしてください。

## 5.5.2 オペレーション

LGB編集処理のオペレーションは、編集内容により3種類に大別されます。

- 選択型 …… データメニューの中から設定内容を選択します。
- 設定型 …… データメニューに提示された範囲内で数値を設定します。
- 混合型 …… データメニューの中から設定パターンを選択し、パターンに従い数値を設定します。

LGB MENU No.	編集項目	オペレーション型		
		選択型	設定型	混合型
0	伝送フレーム	○		
1	伝送レート	○		
2	優先制御	○		
3	データ変換モード	○		
4	テキストサイズ		○	
5	スタートコード			○
6	エンドコード			○
7	BCC	○		
8	送信遅延時間		○	
9	送信中断/再開コード			○
A	送信中断監視時間		○	
B	受信監視時間		○	
C	RS-422ゲートコントロール	○		
D	送信要求	○		
E	データ端末レディ	○		
F	データセットレディ	○		

EX. LINK No. 0

KEYIN MENU NO. = ■

3

▶ メニュー画面から 3 を選択します。

LGB TABLE EDIT

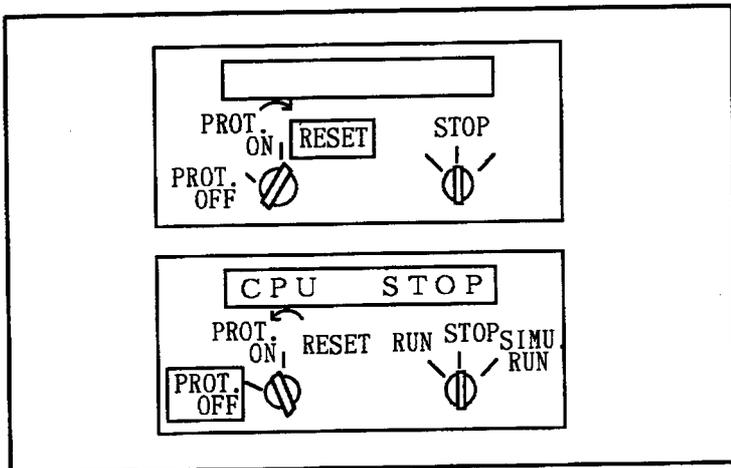
KEYIN MENU NO. = [CLS]

-----  
 CHANNEL No. 0 LGB DATA  
 -----

0 : DATA FRAME -----	ST+8DT+OP+1SP	
1 : BAUD RATE -----	4800 [BPS]	
2 : PRIORITY LEVEL -----	SELF	
3 : DATA CHANGE MODE -----	BINALY	
4 : TEXT SIZE -----	256 [BYTE]	
5 : START CODE -----	02	
6 : END CODE -----	03	
7 : BCC MODE -----	NO BCC	
8 : SEND DELAY TIME -----	NO DELAY	
9 : SEND BREAK/CONTINUE ---	NO BREAK/CONT.	
A : SEND BREAK TIMEOUT ----	32768 [100 mSEC]	
B : RECEIVE TIMEOUT -----	32768 [100 mSEC]	
C : RS-422 GATE CONTROL --	OPEN	
D : REQUEST TO SEND (RS) --	LOW	
E : EQUIPMENT READY (ER) --	HIGH	
F : DATA SET READY (DR) --	NO CHECK	
G : SYSTEM MODE (FUNC) ----	SDO:** RVO:**	… 演算ファンクション システム時
-----		
G : SYSTEM MODE (TASK) ----	TN :** FT :**	… タスクシステム時
-----		

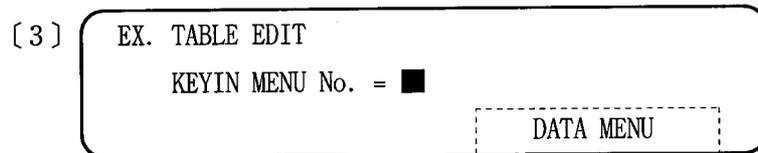
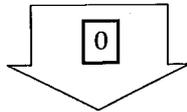
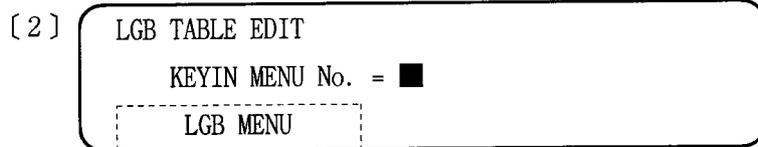
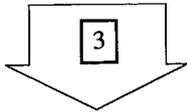
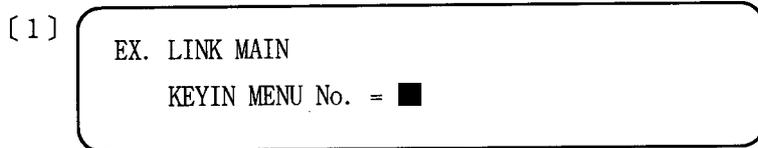
LGBを設定します。

- ① 画面の内容で良い場合  
 終了 キーを押してください。
- ② 変更する場合 (0~F項)  
 変更したいNoを入力して、内容を変更します。  
 変更内容は「5.6 MCS」を参照してください。



▶ CPUのコンソールスイッチをRESET (リセット) →PROT.OFF (プロテクトオフ) にします。

## (1) 選択型オペレーション



[1] メニュー画面から **3** を選択します。

外部機器リンクモジュールの実装を確認しCPUをSTOP、プロテクトOFFの状態にします。

[2] LGBメニュー画面から設定する項目を選択します。

( **0** ~ **4** , **7** , **C** , **F** )  
のいずれかを選択します。

(例) **1** を選択します。

[3] データメニューから設定する項目を選択します。

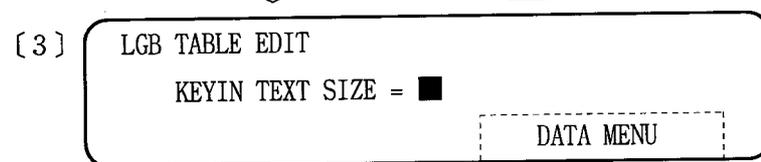
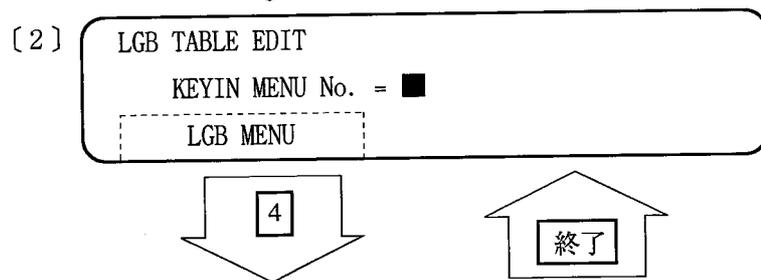
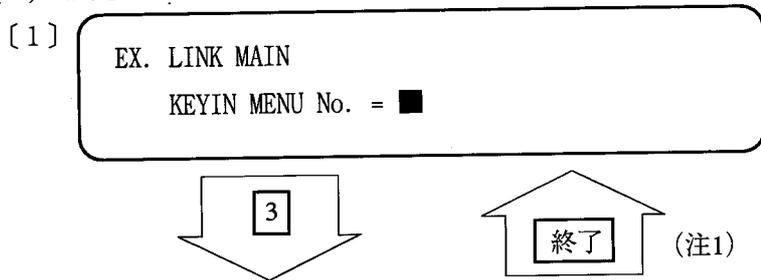
(例) **0** を選択します。

項目を選択するとデータが直ちにLGBに設定され、[2]へ処理は戻ります。



[3] でデータ設定後、[2] で **終了** キーを押すとPSEはCPUのリセット処理待ちとなります。CPUモジュールを必ずリセットしてください。

(2) 設定型オペレーション



[1] メニュー画面から **3** を選択します。

外部機器リンクモジュールの実装を確認しCPUをSTOP、プロテクトOFFの状態にします。

[2] LGBメニュー画面から設定する項目を選択します。

( **4** , **8** , **A** , **B** のいずれかを選択します。 )

(例) **4** を選択した場合。

[3] データメニューに提示された数値の範囲内で設定をします。

(例) 64バイトとする場合。

**6****4****設定** キーを押します。

・ **設定** キーを押した後、もしくは最大桁数入力後直ちにLGBに設定され [2] へ処理は戻ります。

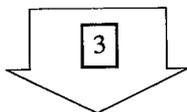


[3] でデータ設定後、[2] で **終了** キーを押すとPSEはCPUのリセット処理待ちとなります。CPUジュールを必ずリセットしてください。

## (3) 混合型オペレーション

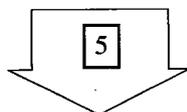
〔1〕

EX. LINK MAIN  
KEYIN MENU No. = ■



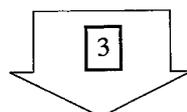
〔2〕

LGB TABLE EDIT  
KEYIN MENU No. = ■  
LGB MENU



〔3〕

LGB TABLE EDIT  
KEYIN DATA No. = ■  
DATA MENU



LGB TABLE EDIT  
KEYIN DATA No. = 3  
KEYIN CODE DATA = \* \* + \* \* + \* \*

〔1〕 メニュー画面から **3** を選択します。

外部機器リンクモジュールの実装を確認しCPUをSTOP、プロテクトOFFの状態にします。

〔2〕 LGB メニュー画面から設定する項目を選択します。

( **5** , **6** , **9** のいずれかを ) 選択します。

(例) **5** を選択した場合。

〔3〕 データMENUから設定するコードパターンを選択します。

**0** ~ **4** のいずれかを選択します。

(例) **3** を選択した場合。

- ・ **0** を選択するとデータが直ちに設定され〔2〕へ処理は戻ります。

〔4〕 コードデータを00h~FFhの範囲内で16進値で設定します。

(例) 01h, 02h, 03hとする場合。

**0** **1** **0** **2** **0** **3** を選択します。

- ・ 選択パターン数分入力後直ちにデータをLGBに設定し〔2〕へ処理は戻ります。



〔4〕でデータ設定後、〔2〕で **終了** キーを押すとPSEはCPUのリセット処理待ちとなります。CPUモジュールを必ずリセットしてください。

5.6 MCS

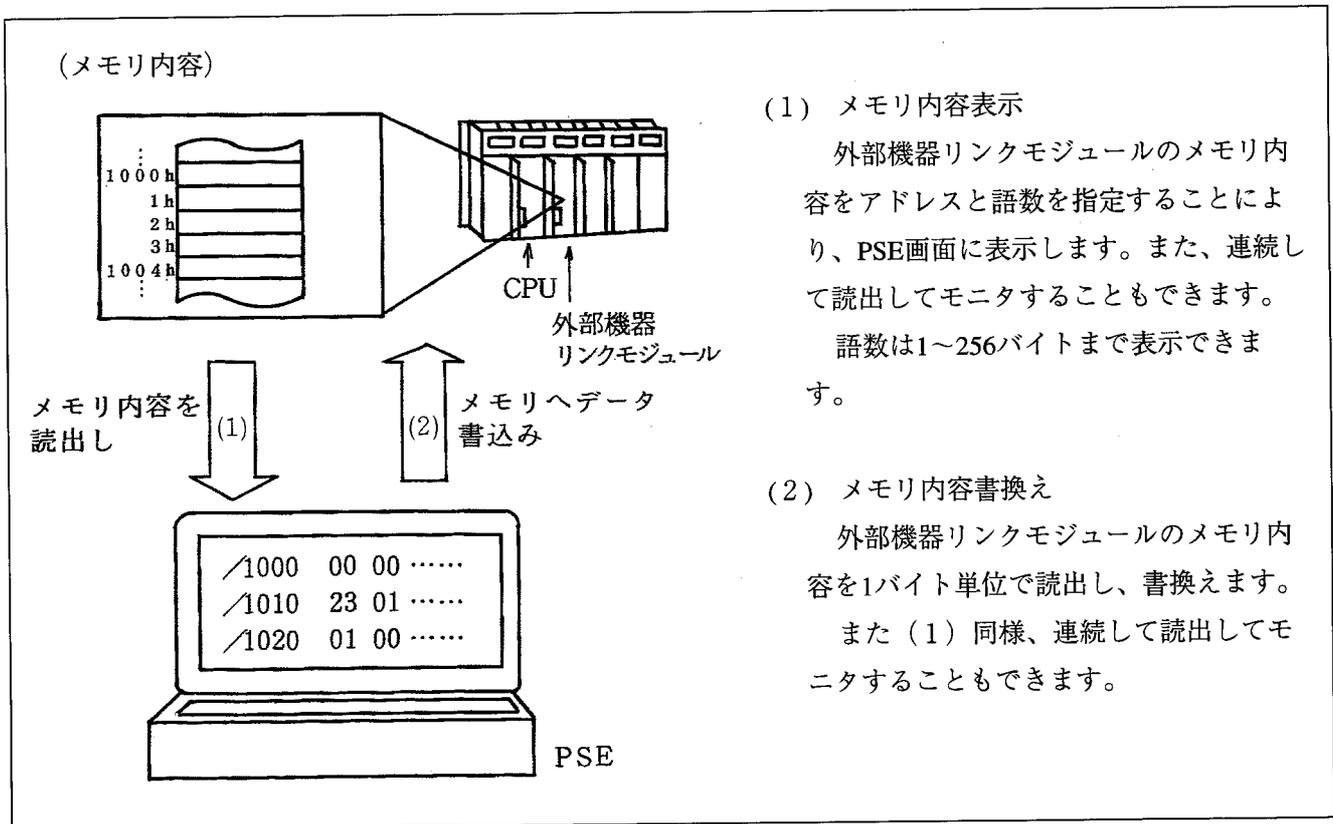
5.6.1 機能概要

この処理は、外部機器リンクモジュール内のメモリ（「付録B 外部機器リンクモジュールのメモリマップ」参照）を機械語で読出し／書込みを行います。

外部機器リンクモジュールと外部機器とのトレースデータ（「付録C トレースバッファ」参照）を参照する場合などに使用します。

なお、このMCSでは外部機器リンクモジュール以外のメモリ（CPU、拡張メモリなど）を読出し／書込みできません。

CPU、拡張メモリなどのメモリを読出し／書込みたい場合は、ラダーシステムのMCSを使用してください。

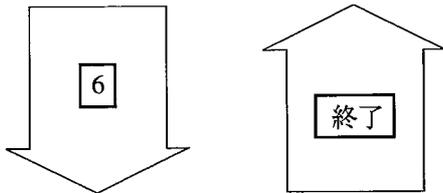


5.6.2 オペレーション

(1) MCSメニュー画面処理

EX. LINK NO. 0  
KEYIN MENU NO. = ■ [CLS]

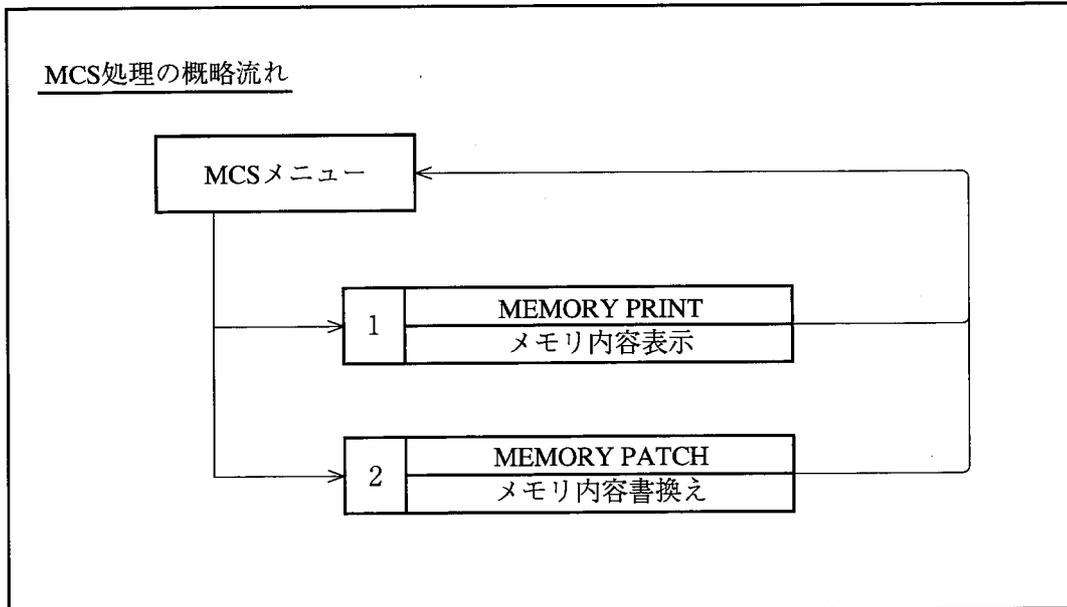
▶ メニュー画面から **6** を選択します。



【MCSメニュー画面】

MCS MENU  
KEYIN NO. = [CLS]  
\*\*\* EX. LINK CHANNEL No. 0 \*\*\*  
1 : : MEMORY PRINT  
2 : : MEMORY PATCH

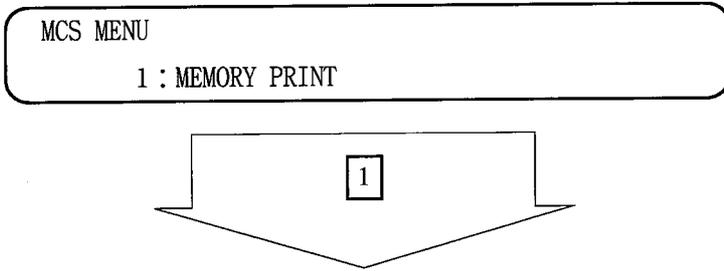
▶ MCSメニュー画面から各処理に対応したナンバキーを選択します。



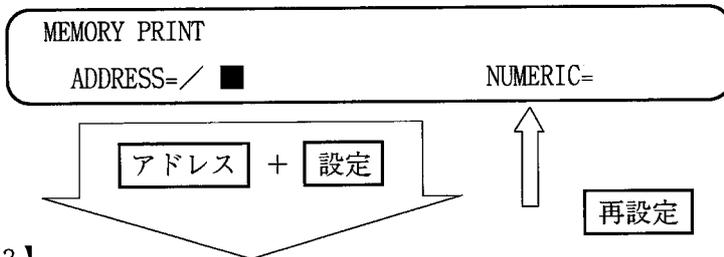
## 5 オペレーション

### (2) メモリ内容表示 (MEMORY PRINT)

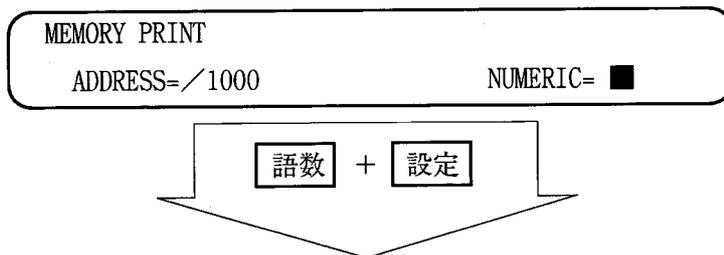
#### 【1】 MCSメニュー画面



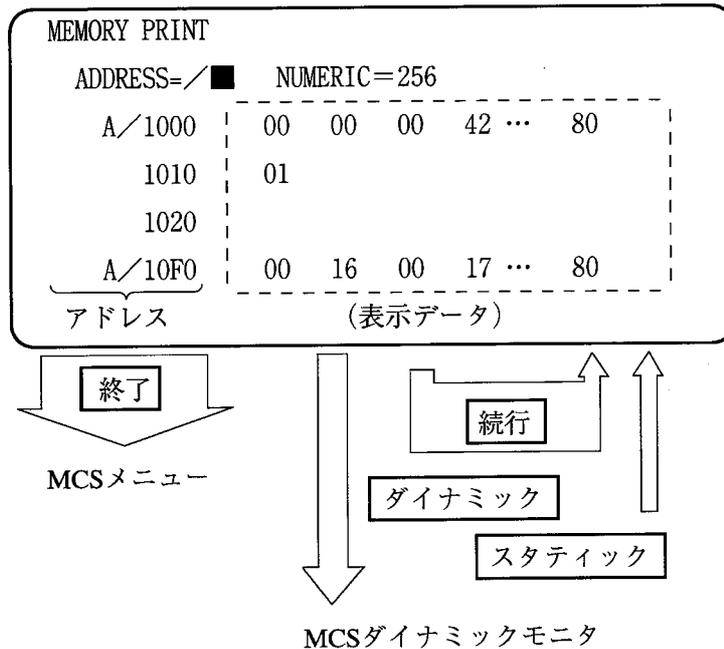
#### 【2】



#### 【3】



#### 【4】



【1】 MCSメニュー画面から“MEMORY PRINT”を選択します。  
[1] を選択します。

【2】 先頭アドレスを指定します。  
1000h番地の場合  
[1][0][0][0] と入力し、[設定] キーを押します。  
(16進アドレス)

【3】 表示語数を指定します。  
256バイトの場合  
[2][5][6] と入力し、[設定] キーを押します。  
(表示語数)

[再設定] … アドレス入力を誤った場合。

【4】 メモリ内容を表示します。  
図のようにメモリ内容が表示されます。また画面が一杯になると、上方へスクロールして表示します。

・【2】～【3】と同様に操作で別のメモリ内容を見ることができます。

[続行] … 次のアドレスからメモリ内を表示します。

[終了] … メモリプリント処理を終了します。

[ダイナミック] … 先に指定したアドレス内容を連続して読出します。ただし、この場合は、[スタティック] キー以外は受け付けません。

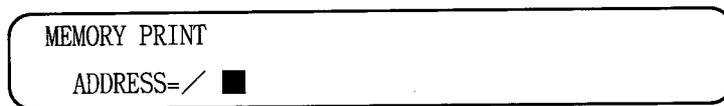
(3) メモリ内容書換え (MEMORY PATCH)

【1】 MCSメニュー画面



2

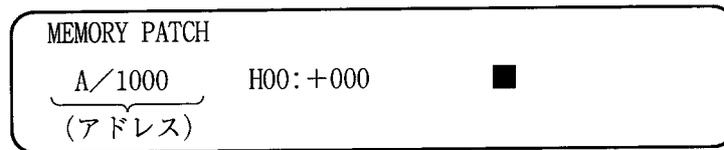
【2】



アドレス + 設定

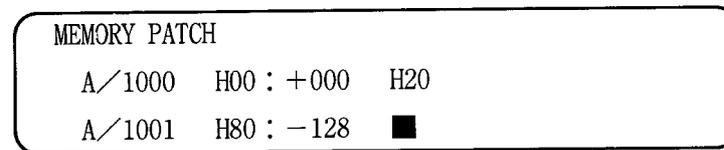
再設定

【3】



データ入力

【4】



MCSメニュー

MCSダイナミックモニター

終了

再設定

【2】 アドレス  
入力へ

続行

ダイナミック

スタティック

【1】 MCSメニューから“MEMORY PATCH”を選択します。

2 を選択します。

【2】 アドレスを入力します。

1000h番地の場合

1 0 0 0

設定

キーを押します。

(16進アドレス)

【3】 書込むデータを入力します。

アドレスを指定すると、図のように現在のメモリ内容を表示 (16進 (H), 10進(+, -) し、キー入力待ちになります。

20h (16進) = 32 (10進) を書込む場合

(1) 16進で設定する場合

H 2 0

設定 キー

を押します。

16進データ

16進データを入力する場合には必ず先頭に入力します。

(2) 10進で設定する場合

3 2

設定 キーを押

します。

(10進データ)

【4】 アドレス移動

以上データの入力終了すると、次のアドレスへ処理が移動します。

続行 ... データはそのままにして次のアドレスへ処理が移動します。

再設定 ... アドレスを再度設定する場合。

ダイナミック ... 現在カーソルが位置するアドレスデータを連続して読出します。この場合は、

スタティック 以外は受け付けません。

終了 ... 処理を終了する場合です。

## 5 オペレーション

### (4) オペレーションの補足説明

メモリ内容表示とメモリ内容書換えの入力および表示方法について、以下に例をあげて説明します。

#### [1] アドレス入力方法 [ADDRESS= / ■]

(メモリ内容表示、メモリ内容書換え共通オペレーション)

- ① 1 0 0 0 設定 ... 5.6.2の(2)および(3)の例で示したようにアドレスを直接入力する  
 合です。  
 この場合は1バイト単位に表示されます。
- ② 1 0 0 0 ! ... これもアドレスを直接入力する方法ですが、この場合データは2バイト単  
 位に表示されます。

#### [2] 語数入力方法 [NUMERIC= ■]

(メモリ内容表示オペレーション)

- ① 1 0 設定 ... [1] のアドレス入力で設定した単位 (1バイトまたは2バイト) でデータを16進  
 数で表示します。
- ② 1 0 ! ... ①と同様ですが、この場合データは10進数 (+, -) で表示します。

- アドレス入力範囲は0000h~7FFFhとなります。  
 (「付録B 外部機器リンクモジュールのメモリマップ」参照)

- 語数入力範囲

	16進	10進
1バイト単位	1~256	1~128
2バイト単位	1~128	1~128

語数入力時に最大値以上、または“0”を入力した場合は、最大値と見なして処理されます。

- ゼロ抑止

MCSのオペレーションで数値を入力する場合は、ゼロ抑止となっています。このため次のような  
 場合、0 キーを入力する必要はありません。

0 0 1 設定 ↔ 1 設定

- 書換えデータ入力範囲

	16進	10進
1バイト単位	H00~HFF	-128~127
2バイト単位	H0000~HFFFF	-32768~32767

正の10進値入力では“+”は  
 不要です。

### 5.6.3 補足説明および注意事項

#### (1) メモリ内容書換えとCPU

メモリ内容の書換えは、必ずCPUをSTOP、プロテクトOFFの状態にしてから行ってください。

また、メモリ内容を書換える場合は、書換えるアドレスに十分注意してください。不用意に内容を書換えるとプログラムの破壊、または外部機器リンクモジュール停止の原因となります。

## 5.7 UFET

### 5.7.1 機能概要

この処理は、外部機器リンク演算ファンクションシステムにおいて、外部機器とデータを送受信するためのユーザ演算ファンクションをUFET（ユーザファンクションエディションテーブル）に登録、削除します。

### 5.7.2 ユーザ演算ファンクション

ユーザ演算ファンクションとは、ユーザが登録できる演算ファンクションです。

外部機器リンクシステムで提供されるユーザファンクションには次に示す4種類があります。

名 称	機 能
SD0	チャンネル0モジュール用送信演算ファンクション
SD1	チャンネル1モジュール用送信演算ファンクション
RV0	チャンネル0モジュール用受信演算ファンクション
RV1	チャンネル1モジュール用受信演算ファンクション

### 5.7.3 送受信演算ファンクション回路作成手順

この処理によりUFET登録後、送受信演算ファンクションを回路で使用できます。シンボル入力手順は次のようになります。

(1) SD0 :	Func	S	D	0	シフト	#	転送アドレス	転送語数
(2) SD1 :	Func	S	D	1	シフト	#	転送アドレス	転送語数
(3) RV0 :	Func	R	V	0	シフト	#	転送アドレス	転送語数
(4) RV1 :	Func	R	V	1	シフト	#	転送アドレス	転送語数

転送語数はワード（16ビット）長です。

## 5 オペレーション

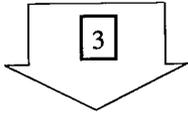
[1]

```

EX. LINK NO.0

KEYIN MENU No. = ■ [CLS]
    
```

▶ メニュー画面から **3** を入力します。



[2]

```

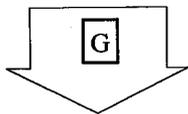
LGB TABLE EDIT

KEYIN MENU No. = ■ [CLS]

G:SYSTEM MODE(FUNC)...SDO:** RVO:**
    
```

▶ LGB編集画面から **G** を入力します。

- ・現在の登録状況がG項に表示されています。
- ・表示内容が‘\*’のときは未登録を示しています。



[3]

```

CHANNEL 0 UFET

KEYIN UFET NO.= ■ [0-F]

NO NAME PRGADDR
-----
0 **** /*****
1 **** /*****
2 **** /*****
3 **** /*****
4 **** /*****
5 **** /*****
6 **** /*****
7 **** /*****
8 **** /*****
9 **** /*****
A **** /*****
B **** /*****
C **** /*****
D **** /*****
E **** /*****
F **** /*****
-----
    
```

▶ 登録するUFETNo.を入力します。

登録済のNo.を入力した場合、削除処理となります。( [6] へ)

(例) No.=0に登録する場合 **0** と入力します。



[4] KEYIN NAME CODE= [S/R] S: SDO (SEND ) -- /\*\*\*\*\*  
 R: RVO (RECEIVE) -- /\*\*\*\*\*

プログラムアドレス

S

- ▶ ユーザファンクションの種類を画面右の表示から選びます。
- S : 送信演算ファンクション
  - R : 受信演算ファンクション
  - 終了 : 入力を中止するとき
- (例) SDOの場合
- S

[5]

No.	NAME	PRGADDR
0	SDO	/ <span style="border: 1px solid black; padding: 2px;">アドレス</span>
1	****	/ ****

- ▲ ユーザファンクションを選択すると、直ちにUFETを登録し、左図のような入力データを表示します。
- ・処理は [3] へ戻ります。
  - ・表示内容が ‘\*’ のときは未登録を示しています。

[6] KEYIN DEL KEY = ■ [DEL]

- ▲ 登録済のユーザファンクションを取消します。
- 削除 : 削除の実行 [7] へ
  - 終了 または 再設定 : 削除の中止 [3] へ

[7] REALLY ? ■ [YES ==0]

- ▲ 削除の確認をします。
- 0 : 削除を実行し [3] へ戻ります。
  - 以外 : 削除を中止し [3] へ戻ります。



・ [4] の画面でプログラムアドレスが /????? と表示されているときは、システムプログラムが壊れています。再ローディングを行ってください。

・ SD? : 送信演算ファンクション ‘?’ はチャンネルナンバ

RV? : 受信演算ファンクション

## 5.8 受信起動タスク登録

### 5.8.1 機能概要

この処理は、外部機器リンクタスクシステムにおいて、外部機器リンクモジュールにデータの受信があった場合、CPUのOSから起動をかけるユーザタスクをモジュールごとに指定します。

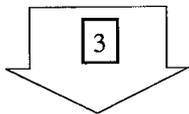
### 5.8.2 オペレーション

[1]

```

EX. LINK NO.0

KEYIN MENU No. = ■ [CLS]
    
```



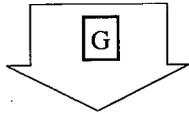
▶ メニュー画面に **3** を入力します。

[2]

```

LGB TABLE EDIT

KEYIN MENU No. = ■ [CLS]
    
```



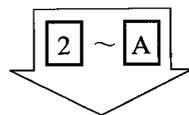
▶ LGB編集画面に **G** を入力します。

[3]

```

ATTENTION TASK

KEYIN TNO. = / ■ [01-7F/DEL]
    
```



▶ データ受信時に起動をかけるタスクのNoを16進2桁で設定します。

(例) タスクNo=2Ah (16進) とする場合

**2** **A** と入力します。

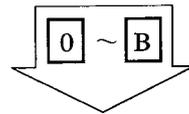
・登録済のデータを削除する場合

**削除** キーを押します。[2]へ戻ります。

[4]

```

KEYIN FACT = / ■ [00-10]
    
```



▶ データ受信時に起動をかけるタスクの要因を16進2桁で設定します。

(例) 要因を0Bh (16進) とする場合

**0** **B** と入力します。

[5]

```

G:SYSTEM MODE (TASK) ... TN:2A FT:0B
    
```

▶ 要因を設定すると直ちに受信起動タスクを登録し、左図のように入力データを表示します。

・処理は[2]へ戻ります。

・表示内容が“\*\*”のときは未登録を示しています。

## 6 プログラム例

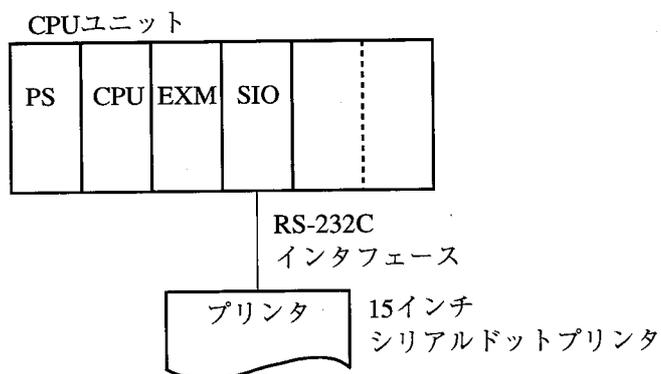
## 6 プログラム例

### 6.1 RS-232Cによるプリンタとの接続例

#### 6.1.1 概要

CPUとシリアルドットプリンタをRS-232Cインタフェースによって接続し、CPUのメモリ内容を指定されたアドレスから語数だけプリンタ出力します。

#### 6.1.2 システム構成

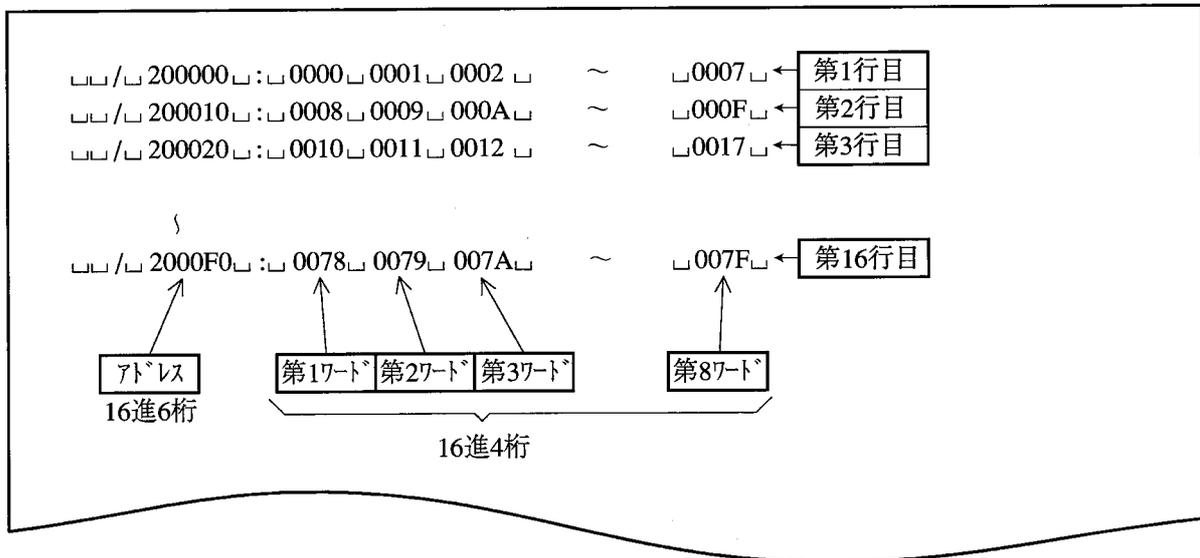


CPUの機器構成

No.	略称	名称	形式	数量
1	PS	電源モジュール	LWV000	1
2	CPU	CPUモジュール	LWP000	1
3	EXM	256Kバイト拡張メモリモジュール	LWM002	1
4	SIO	外部機器リンクモジュール	LWE046	1
5		拡張CPUマウントベース	HPC-1000	1

## 6.1.3 プリンタの印字フォーマット

プリンタの印字フォーマットを以下に示します。

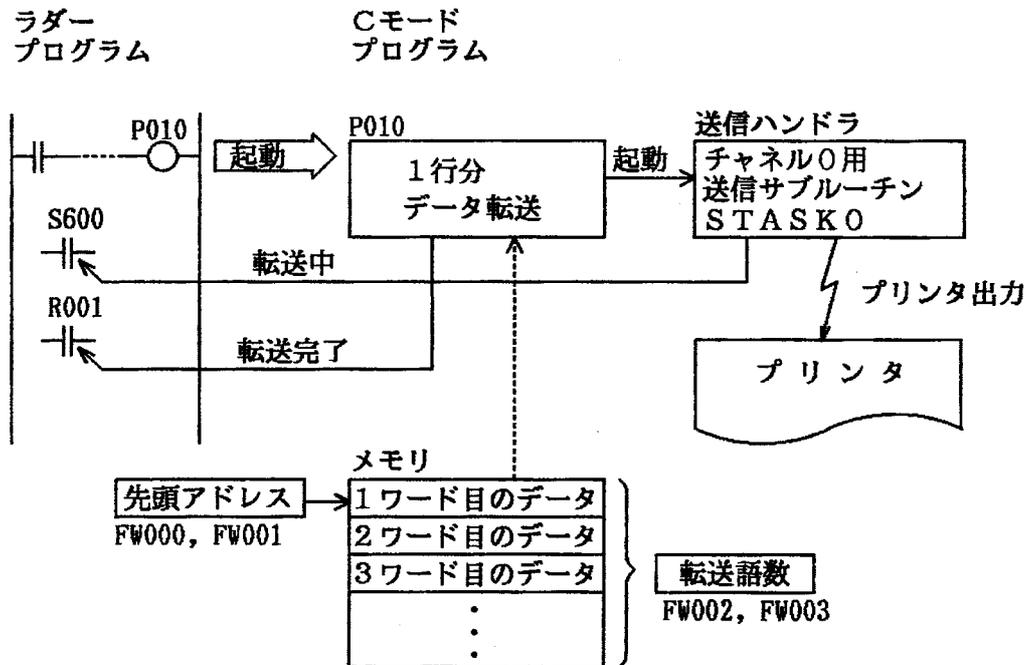


□	スペース (空白)	[20h]	
/	スラッシュ	[2Fh]	アドレスのマーク
:	コロン	[3Ah]	情報の区切り

## 6 プログラム例

### 6.1.4 プログラム構成

プリンタ出力の制御プログラムは、C言語を用いて作成しコンピュータモードプログラムとして動作させます。



Cモードプログラムは、CPUのメモリ内容をプリンタ1行分単位でプリンタ出力処理するサブルーチンとし、CモードプログラムのナンバをP010に割付けます。

このCモードプログラムは、ラダープログラム上で、コイルP010をONすることにより起動できます。

また、メモリのプリンタ出力は、制御情報テーブルにデータレジスタFW000とFW002に先頭アドレスと転送ワード語数を設定することにより行えます。

## 6.1.5 ラダープログラムとのリンケージテーブル構成

## (1) プリンタ出力の制御情報テーブル

シンボル	アドレス	2 <sup>15</sup>	2 <sup>0</sup>
FW000	0E2000h	先頭アドレス	
FW001	0E2002h		
FW002	0E2004h	転送ワード語数	
FW003	0E2006h		

(バイナリデータ)

設定例は、次のようになります。

先頭アドレス=120000h番地

転送ワード語数=16 (10h) ワードの場合、

シンボル	アドレス	データ
FW000	0E2000h	0012h
FW001	0E2002h	0000h
FW002	0E2004h	0000h
FW003	0E2006h	0010h

## (2) 印字完了フラグ

指定された転送ワード語数分をプリンタ出力完了時、ONとなり、初めて一行分の印字処理を起動したとき、OFFします。

シンボル	アドレス	2 <sup>15</sup>	2 <sup>1</sup> 2 <sup>0</sup>
R001	0AC002h		



メモリエリアは、LSB (2<sup>0</sup>ビット：最下位ビット) のみ有効です。

したがって、ON/OFFのデータは、次のようになります。

ON時=0001h

OFF時=0000h

## 6 プログラム例

### (3) 転送中フラグ

プリンタへの1行分データ出力処理が起動する送信ハンドラ (STASKM) のデータ転送中フラグのシステムレジスタS600を使用します。

ここでは、ソフト処理を容易にするためにビット形エリアをアクセスするようにしています。



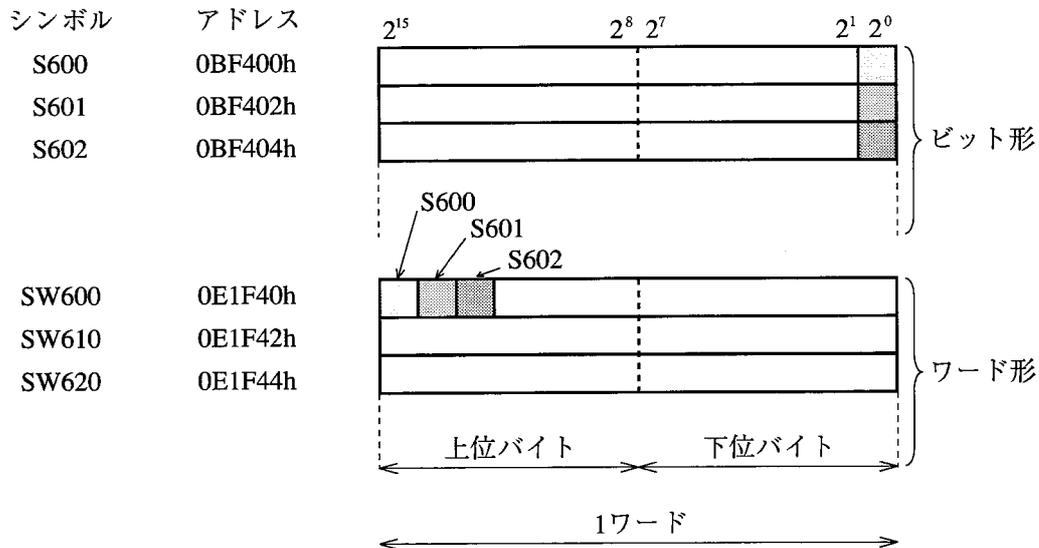
・メモリエリアは、LSB (2<sup>0</sup>ビット：最下位ビット) のみ有効です。

したがって、ON/OFFのデータは、次のようになります。

ON時=0001h

OFF時=0000h

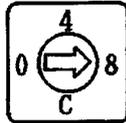
・S600のワード形エリアとビット形エリアの対応は、次のようになっています。



## 6.1.6 ハードウェア構成

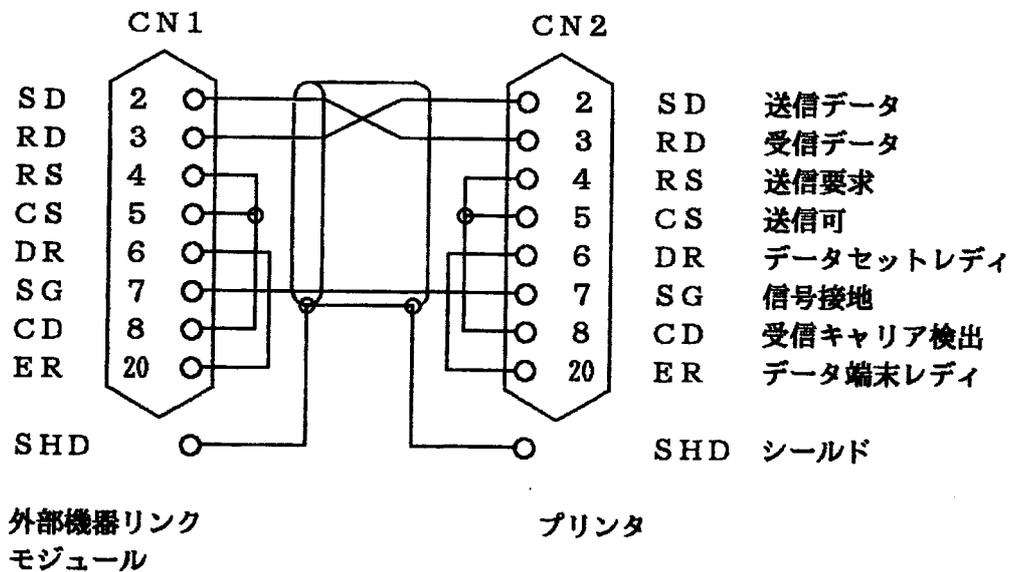
## (1) デイップスイッチのセット

RS-232Cを使用するためデイップスイッチは8にセットします。



## (2) RS-232Cの信号接続

RS-232Cの信号接続はデータ信号線のみ、他の制御線は未使用とし、次のように接続します。



## 6.1.7 LGBの設定

プリンタのシリアルインタフェース仕様を以下に示します。

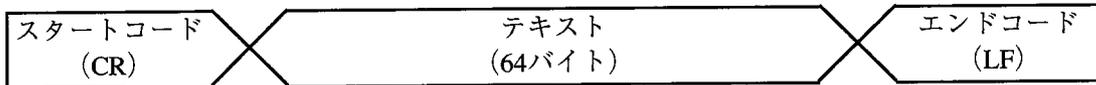
項目	内容	
伝送フレーム	スタートビット	1ビット
	データビット	8ビット
	パリティビット	あり、偶数パリティ
	ストップビット	1ビット
伝送速度	4,800bps	
印字制御	受信データバッファ (1Kバイト) を持ち、ラインフィード (改行: LF, 0Ah) を受信すると、受信データバッファの内容を印字し、自動的に改行します。	

## 6 プログラム例

〈伝送ブロックの構成〉

伝送ブロックのテキスト長は、64バイトとします。

さらに、伝送ブロックのスタートコード (SCD) とエンドコード (ECD) を用いています。



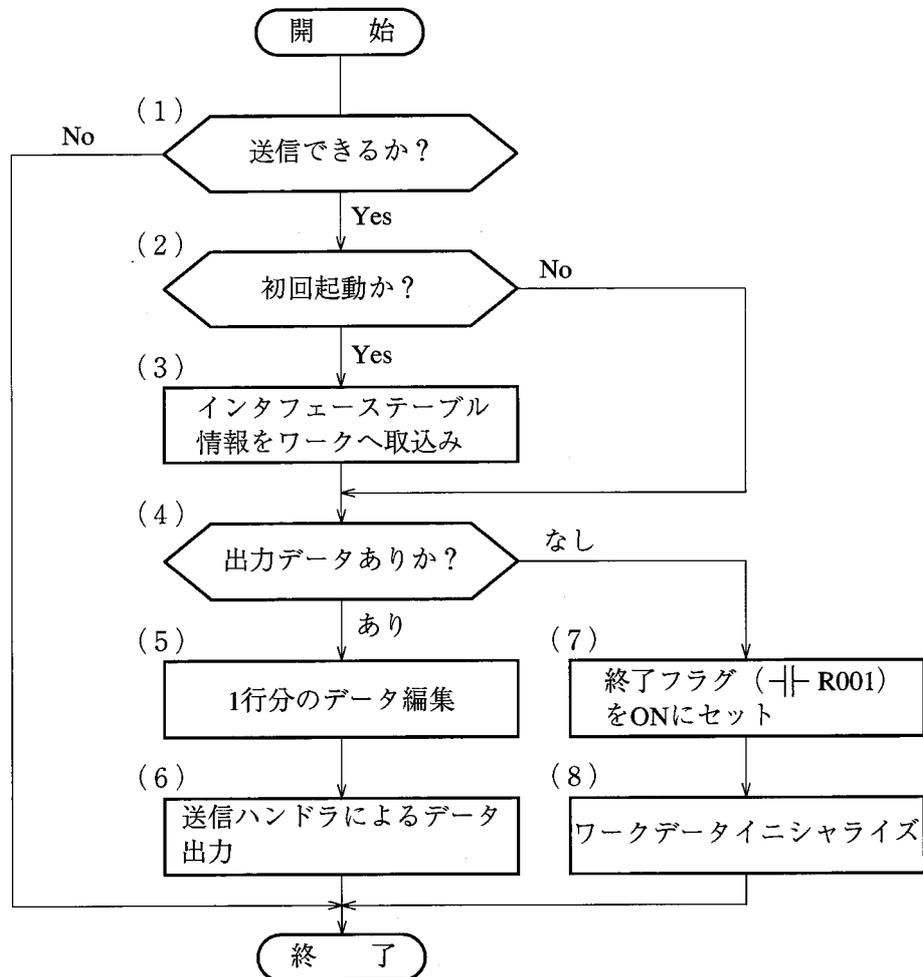
- ・CR：キャリッジリターン (0Dh)
- ・LF：ラインフィード (0Ah)
- ・ブロックチェックキャラクタ (BCC) はなしとします。

### LGBの設定内容

----- MAIN LGB MENU -----			
0 : DATA FRAME	----- ST+8DT+EP+1SP		---※
1 : BAUD RATE	----- 4800 [BPS]	4,800bps	---※
2 : PRIORITY LEVEL	----- SELF	自局優先	
3 : DATA CHANGE MODE	----- BINARY	バイナリ	
4 : TEXT SIZE	----- 064 [BYTE]	64バイト	
5 : START CODE	----- 0D	CR(キャリッジリターン)	---※
6 : END CODE	----- 0A	LF(ラインフィード)	---※
7 : BCC CODE	----- NO BCC	BCCチェックなし	
8 : SEND DELAY TIME	----- NO DELAY	なし	
9 : SEND BREAK/CONTINUE	----- NO BREAK/CONT.	なし	
A : SEND BREAK TIMEOUT	----- 32767 [100 mSEC]	3,276.7秒	
B : RECEIVE TIMEOUT	----- 32767 [100 mSEC]	3,276.7秒	
C : RS-422 GATE CONTROL	----- OPEN	オープン	
D : REQUEST TO SEND (RS)	----- LOW		
E : EQUIPMENT READY (ER)	----- HIGH		
F : DATA SET READY (DR)	----- NO CHECK		
G : SYSTEM MODE	----- "FUNK or TASK"		

他のプリンタを使用する場合には、※印の項目の内容を接続するプリンタの仕様に合わせて変更してください。

## 6.1.8 Cモードプログラムのフローチャート



- (1) システムレジスタ (S600) の状態から送信できるかを調べます。
- (2),(3) 今回が初回の起動かを調べ、初回起動の場合インタフェーステーブル情報を自タスクのワークへ取込みます。
- (4) 次タスクワークの出力残り語数を調べます。
- (5),(6) : データありの場合、データ1行分を編集した後、送信ハンドラにてプリンタへデータを出力します。
- (7),(8) : 全出力が終了した場合、終了フラグ (R001) をONとし、初回起動フラグをOFFとします。

## 6 プログラム例

### 6.1.9 C言語のプログラム例

#### (1) プログラム本体

```
1: /*****/
2: /*      Sample No.1 :: Memory dump task      */
3: /*****/
4:
5: #define TXSUB0  0x1070001          送信ハンドラのアドレス
6: #define IFTB   0xE20001          F000制御情報テーブルのアドレス
7: #define R001   0xAC0021          R001印字完了フラグのアドレス
8: #define S600   0xBF4001          S600転送中フラグのアドレス
9: #define MASK   0x0001           マスクデータ “1”
10:
11: static struct WORK {   short flag ;      処理中フラグ
12:                        long  addr ;      処理中の転送アドレス
13:                        long  word ;      処理中の転送語数
14:                        } work ;
15:
16: static char linebf[ 64 ] ;           プリント出力用の一行分バッファ
17:
18:
19: p010( )
20: {
21: register long (*txsub)( ) ;
22: register long *lpt ;
23: register char *cpt ;
24: register short wk ;
25: register short ct ;
26: register long retncd ;
27:
28: if( ( *(short *)S600 & MASK ) == 0 )   送信可チェック
29:     {
30:         if( work.flag == 0 )           制御情報テーブルの取込み
31:             {
32:                 lpt = (long *)IFTB ;
33:                 work.addr = *lpt++ ;
34:                 work.word = *lpt ;
```

```
35:     work.flag = 1 ;
36:     }
37:     if( work.word > 0 )           プリンタ出力処理
38:     {                             行バッファのイニシャライズ
39:         ct = 64 ;
40:         cpt = &linebf[0] ;
41:         while( --ct >= 0 )
42:             *cpt++ = ' ' ;
43:
44:         (long)cpt = &(work.addr) ;   アドレスデータの設定
45:         btoas( &linebf[3] , cpt[1] ) ;
46:         btoas( &linebf[5] , cpt[2] ) ;
47:         btoas( &linebf[7] , cpt[3] ) ;
48:
49:         (long)cpt = work.addr ;      メモリデータの設定
50:         ct = 12 ;
51:         While( ( work.word > 01 ) && ( ct < 50 ) )
```

## 6 プログラム例

### (2) バイナリ→ASCII (アスキー) 変換サブルーチン

```
52:      {
53:          btoas( &linebf[ct] , *cpt++ ) ; 上位バイトデータ
54:          btoas( &linebf[ct+2] , *cpt++ ) ; 下位バイトデータ
55:          ct += 5 ;                      SP(スペース)の設定
56:          work.word -= 11 ;
57:      }
58:      work.addr += 0x0000101 ;
59:
60:      linebf[2] = '/' ;                  アドレスマーク"/"
61:      linebf[10] = ':' ;                 データの区切り":"
62:
63:      (long)txsub = TXSUB0 ;             データ転送
64:      retncd = (*txsub)( &linebf[0] , 641 ) ;
65:      }
66:  else{
67:      work.flag = 0 ;
68:      *(short *)R001 = 1 ;              印字完了フラグの設定
69:      }
70:  }
71: return ;
72: }
73:
74: /*****
75: /*      Binary --> Ascii function ( byte size )          */
76: /*****
77: btoas( stp , data )
78: register char *stp ;                      キャラクタの設定ポインタ
79: register char data ;                      バイナリデータ
80: {
81: register char wk ;                        ワークレジスタ
82:
83: wk = data ;                              上位桁の設定
84: wk >>= 4 ;
85: wk &= (char)0x0F ;
86: if ( wk <= (char)0x09 )
```

```

87:   wk += (char)0x30 ;
88: else wk += (char)0x37 ;
89: *stp++ = wk ;
90:
91: data &= (char)0x0F ;           下位桁の設定
92: if( data <= (char)0x09 )
93:   data += (char)0x30 ;
94: else data += (char)0x37 ;
95: *stp = data ;
96:
97: return ;
98: }
99: /*****/

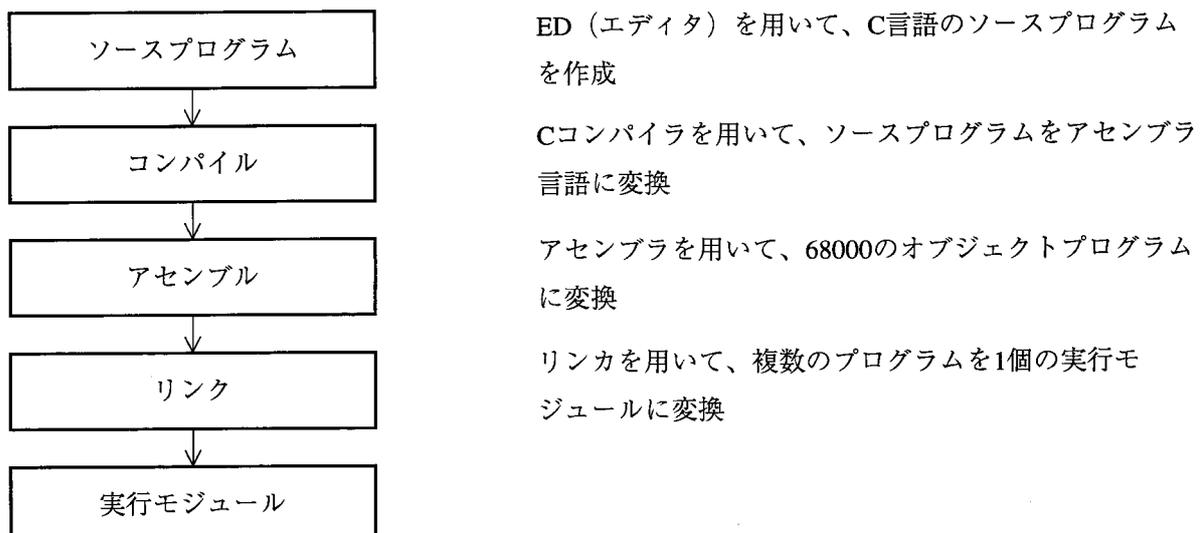
```

### 6.1.10 Cモードプログラムの作成と登録

#### (1) C言語プログラムの作成

C言語で作成されたCモードプログラムは、拡張形ポータブルPSE  $\alpha$  (HPC-6000-20) を使用して、C言語開発システムにより、C言語で作成されたソースプログラムをCPUで実行できるロードモジュールにします。

作成手順を以下に示します。



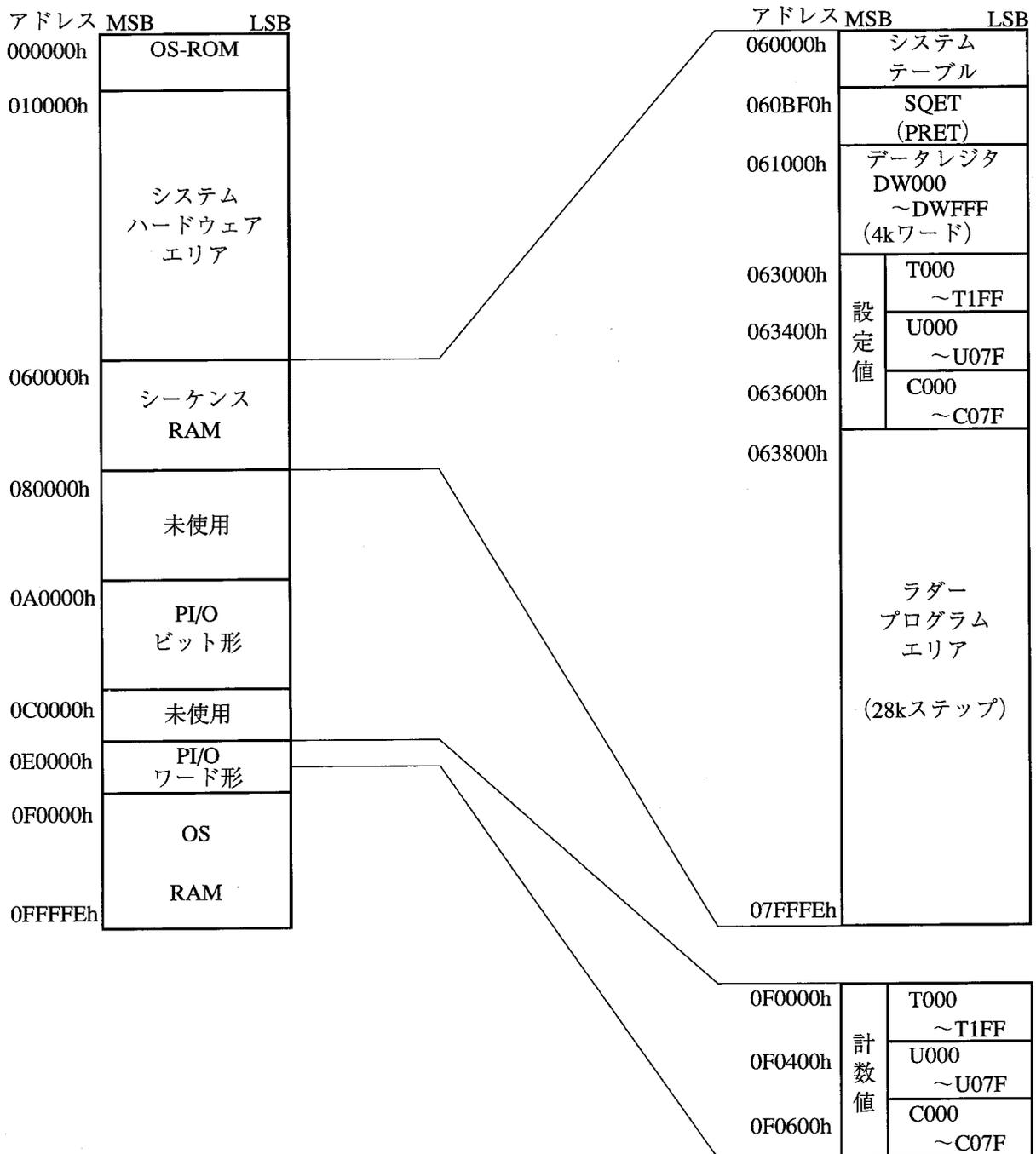
## 6 プログラム例

### (2) Cモードプログラムのロード (書込み)

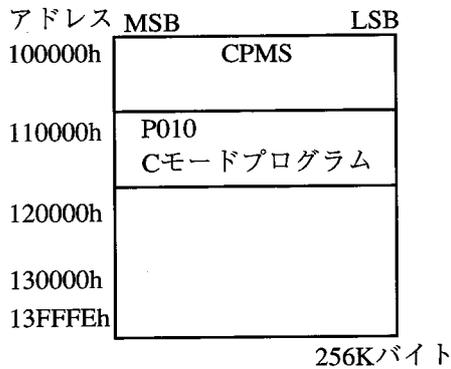
実行できるCモードプログラムは、3.5インチフロッピーディスクからポータブルPSE $\alpha$ を用いて、CPUの256Kバイトの拡張メモリにロードします。

CPUのメモリ構成を以下に示します。

〈基本メモリ〉



〈拡張メモリ〉



実行プログラムを拡張メモリのアドレス110000h番地からロード（書込み）します。

### (3) Cモードプログラムの登録

Cモードプログラムは、「PRET (C-MODE) SET機能」を用いて、登録します。

PRET機能の設定項目と、意味を以下に示します。

Pno.	Pコイルナンバ
LEVEL	プログラムレベル (1~4)
ADDRESS	プログラムの先頭アドレス
U.S.P	プログラムで使用するローカル変数エリア (スタック領域) のアドレス

設定例は、次のようになります。

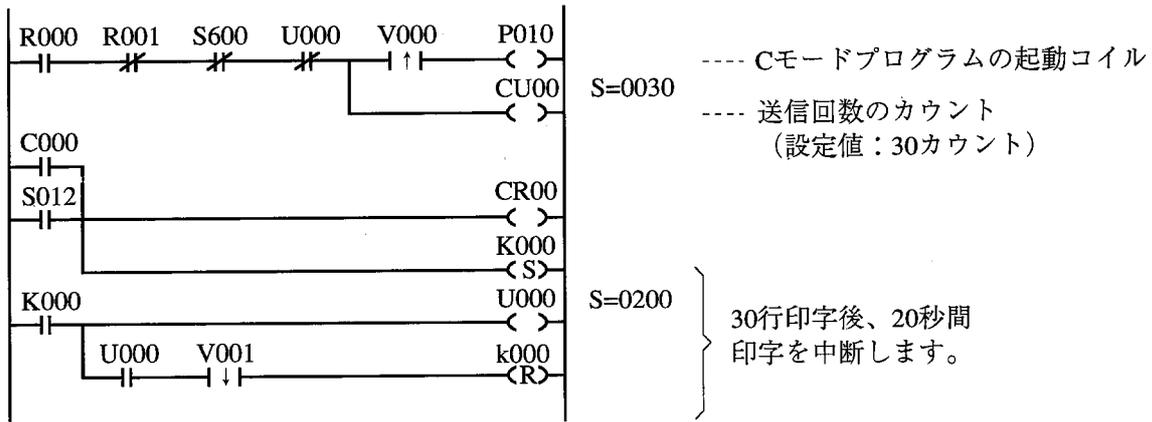
Pno.	=010
LEVEL	=4
ADDRESS	=110000
U.S.P	=120000

## 6 プログラム例

### (4) Cモードプログラムを起動するラダープログラムの作成

プリンタに出力するためには、P010に登録されたCモードプログラムを起動するラダープログラムが必要です。

ラダープログラムの例は、次に示します。



R000	印字指令
R001	印字完了
S600	外部機器リンクの転送中
C000	30行印字カウンタ
U000	30行印字後、中断タイマ
S012	STOP→RUN信号
K000	30行印字カウンタの停電記憶



・今回使用したプリンタは、印字速度がデータ転送の速度に比べかなり遅く、印字データを連続して送信した場合には、プリンタのデータ受信バッファがオーバーフローとなり誤動作となりました。  
 このため、ラダープログラムにて、30行印字後、20秒間送信を中断するようにしています。

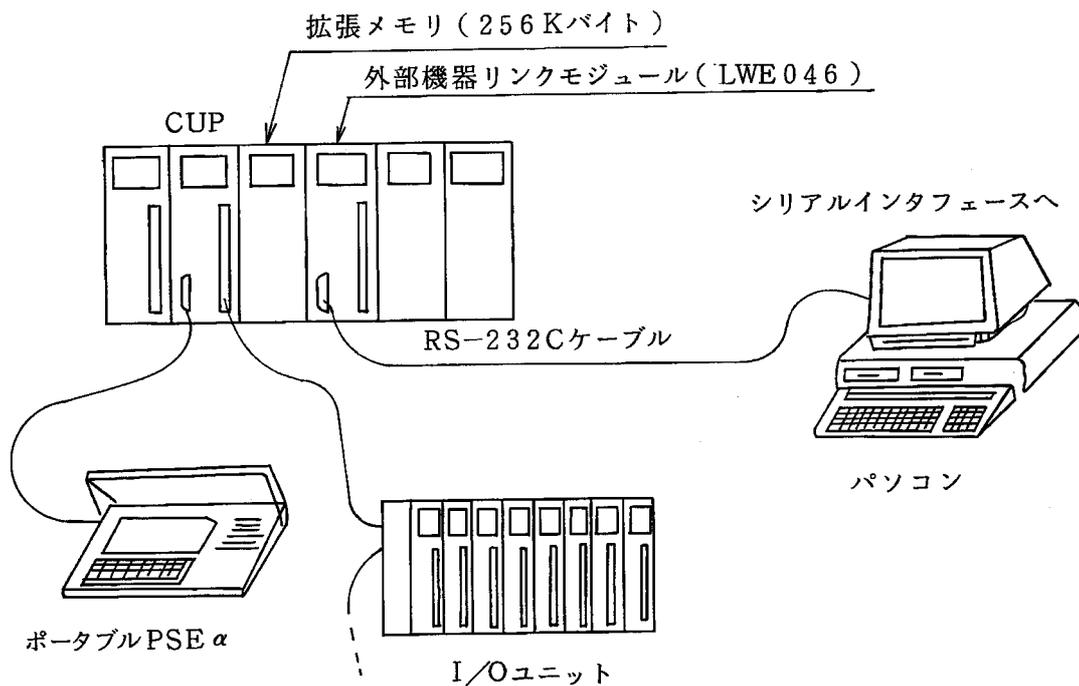


・このプログラムは、理解しやすいことを目的に書かれていますので、実用的に良いプログラムとは言えません。  
 実用プログラムでは、送信ハンドラのリターンコードおよびシステム (S) レジスタのエラーチェックを行う必要があります。

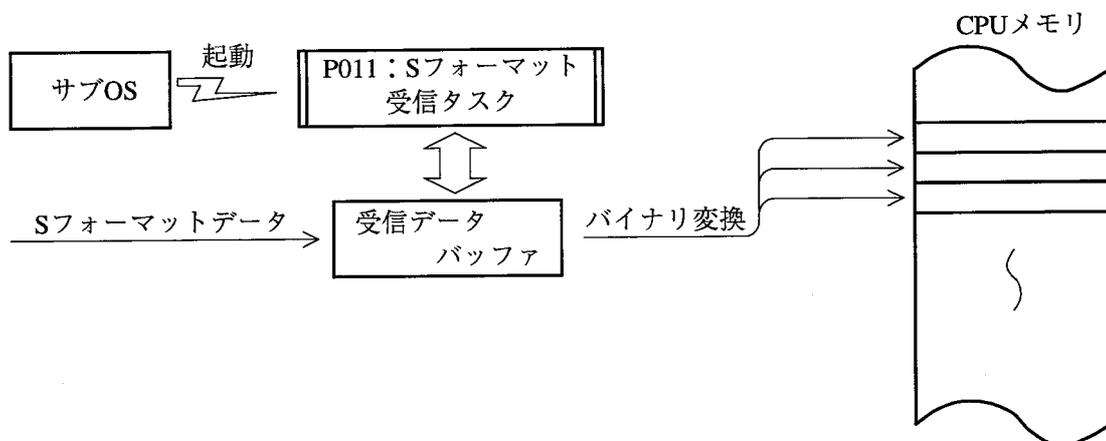
## 6.2 パソコンによるプログラムローディング

パソコンをRS-232Cインターフェースを用いてCPUに接続し、C言語などで作成したコンピュータモードプログラムを直接CPUメモリにロードします。

### 6.2.1 システム構成



### 6.2.2 プログラム構成



パソコンからSフォーマットデータを受信すると、Sフォーマット受信タスクに起動がかり、このタスクにより受信されたデータを指定のメモリアドレスへ設定します。

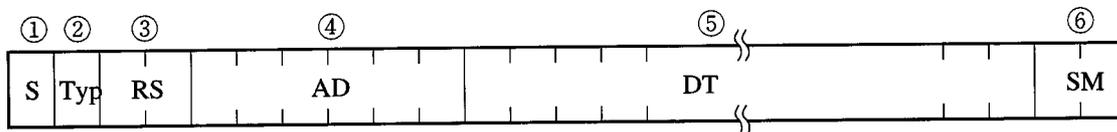
## 6 プログラム例

### 6.2.3 モトローラ 'S' フォーマット (16ビット用)

このフォーマットは68000などのオブジェクトを回線を使って送受信するためのデータフォーマットです。

① ② ③ ④ ⑤ ⑥

```
S22411000048E7FFFE207C00100000227C00110000247C00120000267000130000287C001424
S22411002000002A7C001200002C7C00160000203CD0D0D0D0223CD1D1D1D1243CD2D2D2D24B
S224110040263CD3D3D3D3286CD4D4D4D42A3CD5D5D5D52C3CD6D6D6D62E3CD7D7D7D74CDFBD
S2241100607FFF4E75000000000000000000000000000000000000000000000000000000000029
S9030000FC
```



- ① … レコードの最初を示すマークで必ず 'S' (53h) となります。
- ② … レコードのタイプを表し、次のように分けられます。  
 '2' (32h) : データレコード  
 '9' (39h) : 最終レコード
- ③ … レコードのバイトサイズを2文字で表しています。(④~⑥のバイト数)
- ④ … データの対応する先頭アドレスを6文字で表しています。
- ⑤ … メモリデータを1バイト分を2文字で表しています。
- ⑥ … ③~⑤のデータの総和の1の補数を、チェックサムデータとし2文字で表しています。



・今回使用したパソコンでは各レコードのチェックサムデータの後にCR (キャリッジリターン: 0Dh)、LF (ラインフィード: 0Ah) が追加されて転送されました。そこで、スタートコード、エンドコードを次のように決定しました。(「6.2.4 LGBの設定」参照)

スタートコード (SCD) = 'S' (=53h) … 1文字

エンドコード (ECD) = CR (0Dh) + LF (0Ah) … 2文字

### 6.2.4 LGBの設定

LGBの設定例を示します。

```

-----
                MAIN LGB MENU
-----
0 : DATA FRAME ----- ST+8DT+0P+1SP
1 : BAUD RATE ----- 1200 [BPS]
2 : PRIORITY LEVEL ----- SELF ----- 自局優先
3 : DATA CHANGE MODE ----- BINARY ----- バイナリ
4 : TEXT SIZE ----- 256 [BYTE] ----- 256バイト
5 : START CODE ----- 53 ----- 'S'
6 : END CODE ----- OD+0A ----- CR+LF
7 : BCC CODE ----- NO BCC
8 : SEND DELAY TIME ----- NO DELAY
9 : SEND BREAK/CONTINUE ----- NO BREAK/CONT.
A : SEND BREAK TIMEOUT ----- 32767 [100 mSEC]
B : RECEIVE TIMEOUT ----- 32767 [100 mSEC]
C : RS-422 GATE CONTROL ----- OPEN
D : REQUEST TO SEND (RS) ----- LOW
E : EQUIPMENT READY (ER) ----- HIGH
F : DATA SET READY (DR) ----- NO CHECK
-----

```

ボーレート : ここでは速度が速すぎるとCPU負荷の増大につながり、データが正常に受信できない場合があるため、遅めに設定しました。

テキストデータ : ここではバイナリデータとし受信タスクでバイナリ変換します。

テキストサイズ : 標準サイズの256バイトとしました。

スタートコード : Sフォーマットは 'S' からレコードが始まるためこれをスタートコードとしました。

エンドコード : Sフォーマットのチェックサムデータの次にCR, LFが送信されてくるため、これをエンドコードとしました。

その他 : パソコンに合わせて設定してください。

### 6.2.5 受信タスクの登録

ここではコンピュータモードプログラムを受信タスクとしてP011 (タスク番号11h) に登録します。さらにそのファクタ (FACT) は01hとします。

〈受信タスク登録例〉

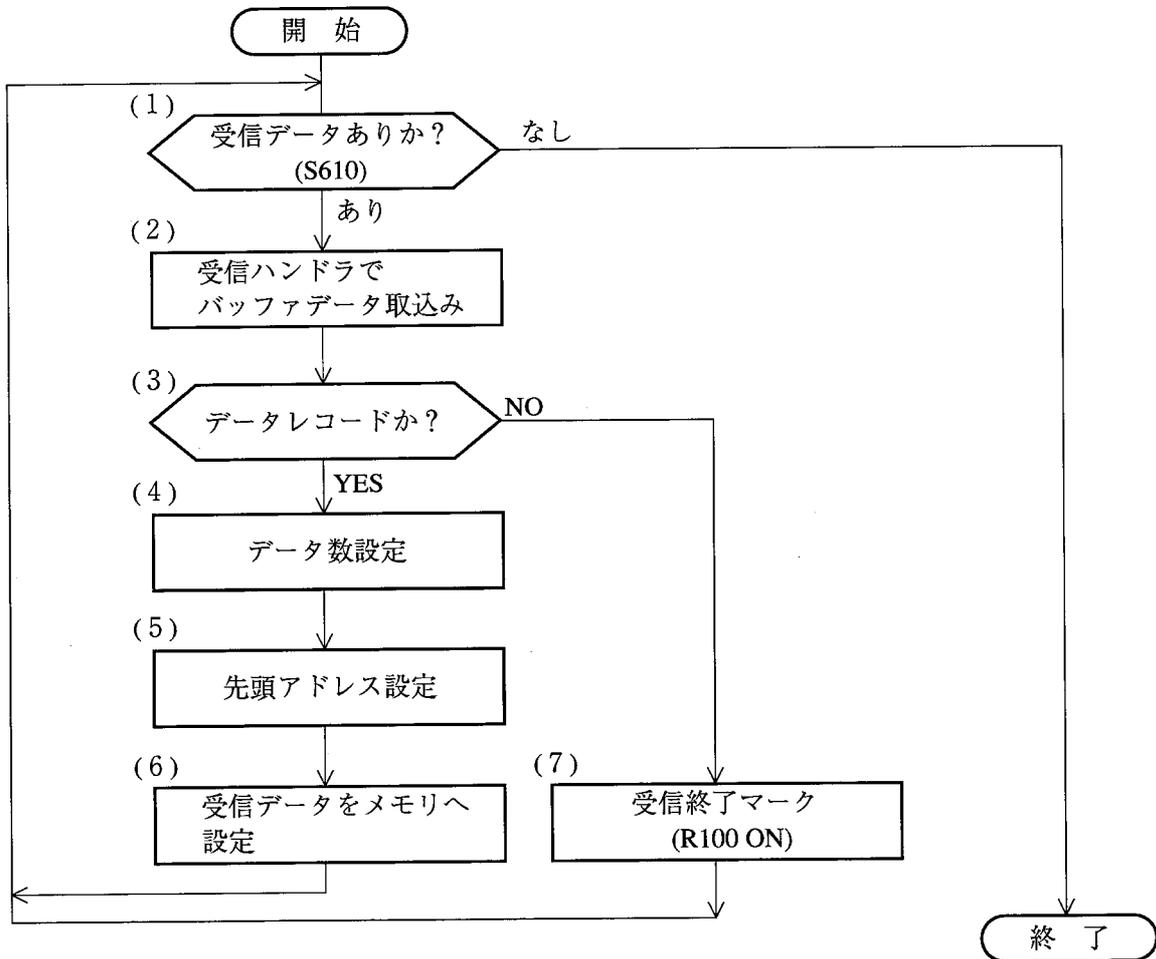
```

-----
                RECEIVE TASK MENU
-----
1 : MAIN MODULE -- TNO=/11 FACT=/01
2 : SUB MODULE -- TNO=/** FACT=/**
-----

```

## 6 プログラム例

### 6.2.6 受信タスク



- (1) 受信フラグ (S610) がONしていることを確認します。
- (2) 受信ハンドラにより現在受信したデータを取込みます。
- (3) レコードタイプが '2' (32h) であることを確認します。
- (4)~(6) データレコードだった場合は、データ数、先頭アドレスを取込み、その情報に従いメモリへデータを設定します。

また、データレコードでない場合は、最終レコードとみなし、終了マーク (R100) をONとします。

ただし、以上のデータの読み込みは受信フラグがOFFになるまで続けます。

## 6.2.7 C言語のプログラム例

```

1: /*****
2: /*      Sample No.2 :: Program Loading task      */
3: /*****
4:
5: #define RXSUB0  0x10700C1          受信ハンドラのアドレス
6: #define S610   0xBF4201          S610転送中フラグのアドレス
7: #define R100   0xAc2001          受信完了フラグのアドレス
8: #define MASK   0x0001          マスクデータ "1"
9:
10: static char buff[512] ;          受信データのバッファ(512バイト)
11:
12: p011( )
13: {
14: extern char atob( ) ;
15: register long (*sub)( ) ;          受信ハンドラ
16: register char *addr ;             アドレスのポインタ
17: register short *dpt ;            データのポインタ
18:
19: register long retncd ;            リターンコード
20: register short ct ;              ループカウンタ
21:
22: union { long lad ;
23:         char cad[4] ;
24:         } adwk ;
25:
26: (long)sub = RXSUB0 ;              受信データの取込み
27: while( ( *(short *)S610 & MASK ) != 0 )
28:     {
29:         retncd = (*sub)( &buff[0] , 801 ) ;
30:         if( buff[0] == '2' )
31:             {                      データNo.の設定
32:                 ct = (short)atob( &buff[1] ) ;
33:                 ct &= 0x003F ;      アドレスNo.の設定
34:                 adwk.cad[1] = atob( &buff[3] ) ;
35:                 adwk.cad[2] = atob( &buff[5] ) ;

```

## 6 プログラム例

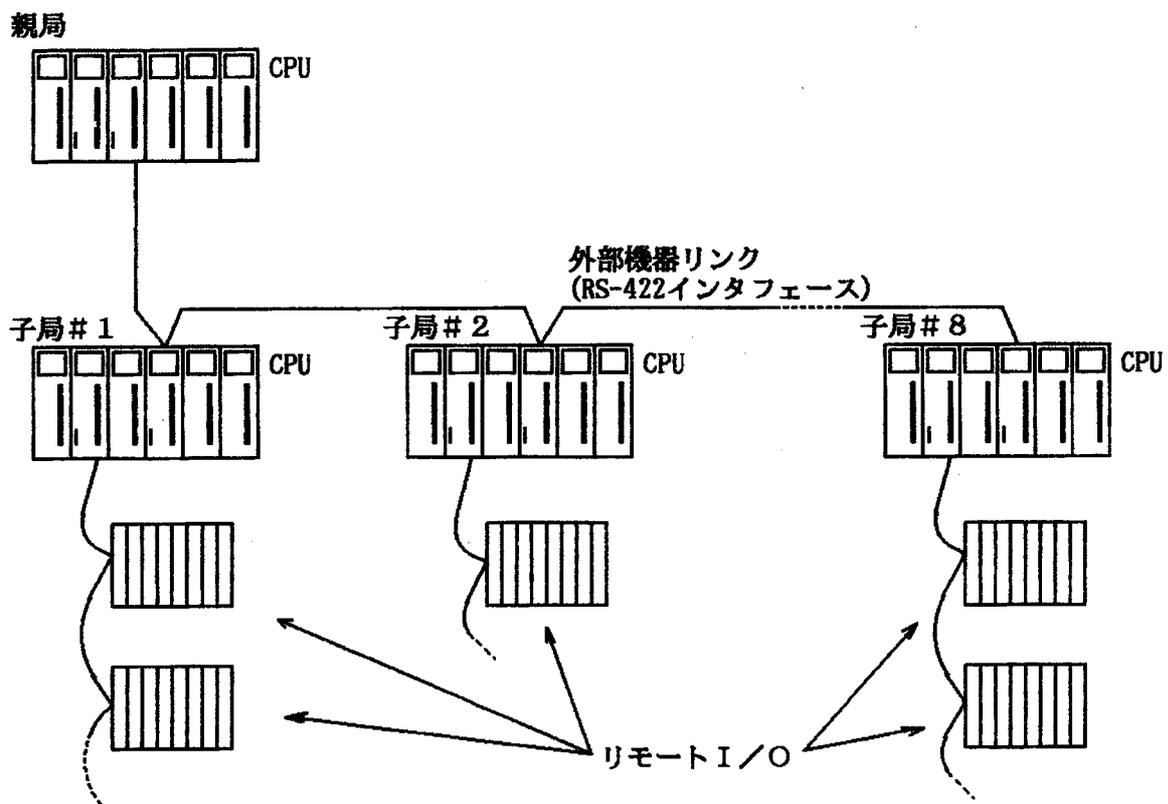
```
36:      adwk.cad[3] = atob( &buff[7] ) ;
37:      adwk.cad[0] = (char)0 ;
38:      (long)addr = adwk.lad ;           データのメモリへの書込み
39:      (char *)dpt = &buff[9] ;
40:      ct -= 3 ;
41:      while( --ct > 0 )
42:          *addr++ = atob( dpt++ ) ;
43:      }
44:      else *(short *)R100 = 1 ;         受信完了フラグの設定
45:      }
46: return ;
47: }
48: /*****
49: /*      ASCII --> BINARY function      */
50: /*****
51: char atob( pt )
52: register char *pt ;
53: {
54: register char wkh , wkl ;
55:
56: wkh = *pt++ ;
57: wkh -= '0' ;
58: if( wkh > (char)9 )
59:     wkh -= 7 ;
60: wkh <<= 4 ;
61: wkh &= (char)0xF0 ;
62:
63: wkl = *pt ;
64: wkl -= (char)0x30 ;
65: if( wkl > (char)9 )
66:     wkl -= 7 ;
67: wkl &= (char)0x0F ;
68:
69: wkh |= wkl ;
70: return( wkh ) ;
71: }
72: /*****/
```



### 6.3 演算ファンクションによるデータ転送

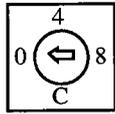
外部機器リンクモジュールを使用し、RS-422インタフェースによる最大8台のCPUでI/Oデータを送受信します。

#### 6.3.1 システム構成



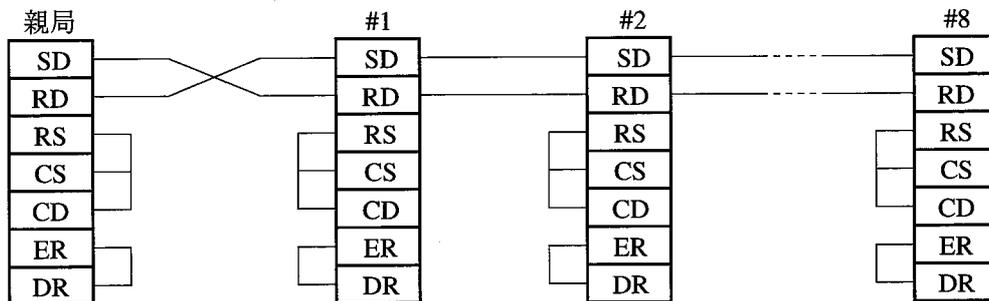
## 6.3.2 ハードウェアの設定

すべてMAINモジュールのRS-422として使用するため、親局、子局ともすべてディップスイッチを0にセットしてください。



0 : RS-422のMAINにセット

ケーブルは親局から各子局に対し次のように接続してください。



## 6 プログラム例

### 6.3.3 LGBの設定

LGBの設定例を以下に示します。

RS-422マルチドロップで接続する場合、特に子局側は、RS-422の送信ゲートコントロールを行ってください。

〈親局側設定例〉

----- MAIN LGB MENU -----			
0:DATA FRAME	-----	ST+8DT+OP+1SP	
1:BAUD RATE	-----	9600 [BPS]	----- 9,600 bps
2:PRIORITY LEVEL	-----	SELF	----- 自局優先
3:DATA CHANGE MODE	-----	BINARY	----- バイナリ
4:TEXT SIZE	-----	256 [BYTE]	----- 256バイト
5:START CODE	-----	02	----- STX(02h)
6:END CODE	-----	FF+FF+FF+FF	----- (注4)
7:BCC MODE	-----	NO BCC	
8:SEND DELAY TIME	-----	NO DELAY	
9:SEND BREAK/CONTINUE	-----	NO BREAK/CONT.	
A:SEND BREAK TIMEOUT	-----	32767 [100 mSEC]	
B:RECEIVE TIMEOUT	-----	32767 [100 mSEC]	
C:RS-422 GATE CONTROL	-----	OPEN	----- オープン (注1)
D:REQUEST TO SEND (RS)	-----	LOW	} (注5)
E:EQUIPMENT READY (ER)	-----	HIGH	
F:DATA SET READY (DR)	-----	NO CHECK	

## 〈子局側設定例〉

MAIN LGB MENU			
0:DATA FRAME	-----	ST+8DT+OP+1SP	
1:BAUD RATE	-----	9600 [BPS]	----- 9,600 bps
2:PRIORITY LEVEL	-----	SELF	----- 自局優先
3:DATA CHANGE MODE	-----	BINARY	----- バイナリ
4:TEXT SIZE	-----	256 [BYTE]	----- 256バイト
5:START CODE	-----	02	----- STX(02h)
6:END CODE	-----	FF+FF+FF+FF	----- (注4)
7:BCC MODE	-----	NO BCC	
8:SEND DELAY TIME	-----	NO DELAY	
9:SEND BREAK/CONTINUE	-----	NO BREAK/CONT.	
A:SEND BREAK TIMEOUT	-----	32767 [100 mSEC]	
B:RECEIVE TIMEOUT	-----	32767 [100 mSEC]	
C:RS-422 GATE CONTROL	---	CONTROL	----- コントロール (注2)
D:REQUEST TO SEND (RS)	---	LOW	} (注5)
E:EQUIPMENT READY (ER)	---	HIGH	
F:DATA SET READY (DR)	---	NO CHECK	

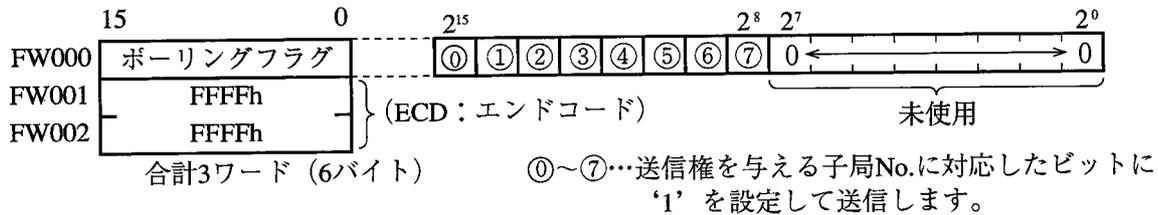
- (注1) 親局側がゲートコントロールする必要はありません。
- (注2) 子局側は必ずゲートコントロールモードに設定してください。
- (注3) 転送速度、伝送フレームなどは親側、子側で必ず一致させてください。
- (注4) この例で取扱うデータの中は、“FFFFFFFFhと‘1’のビットが32個続くデータはなし”という規則に従い、EBCD (エンドコード) を (FFh + FFh + FFh + FFh) の4文字としました。
- (注5) MENU中0~Fの項目は必ず上記例のように設定してください。

## 6 プログラム例

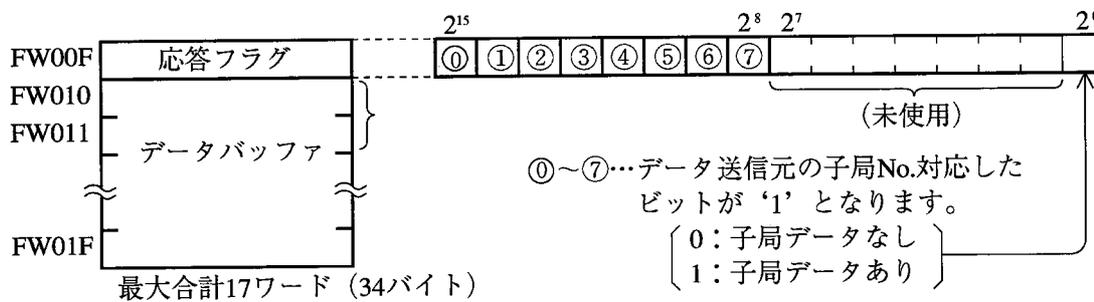
### 6.3.4 データテーブル構成

・親局側

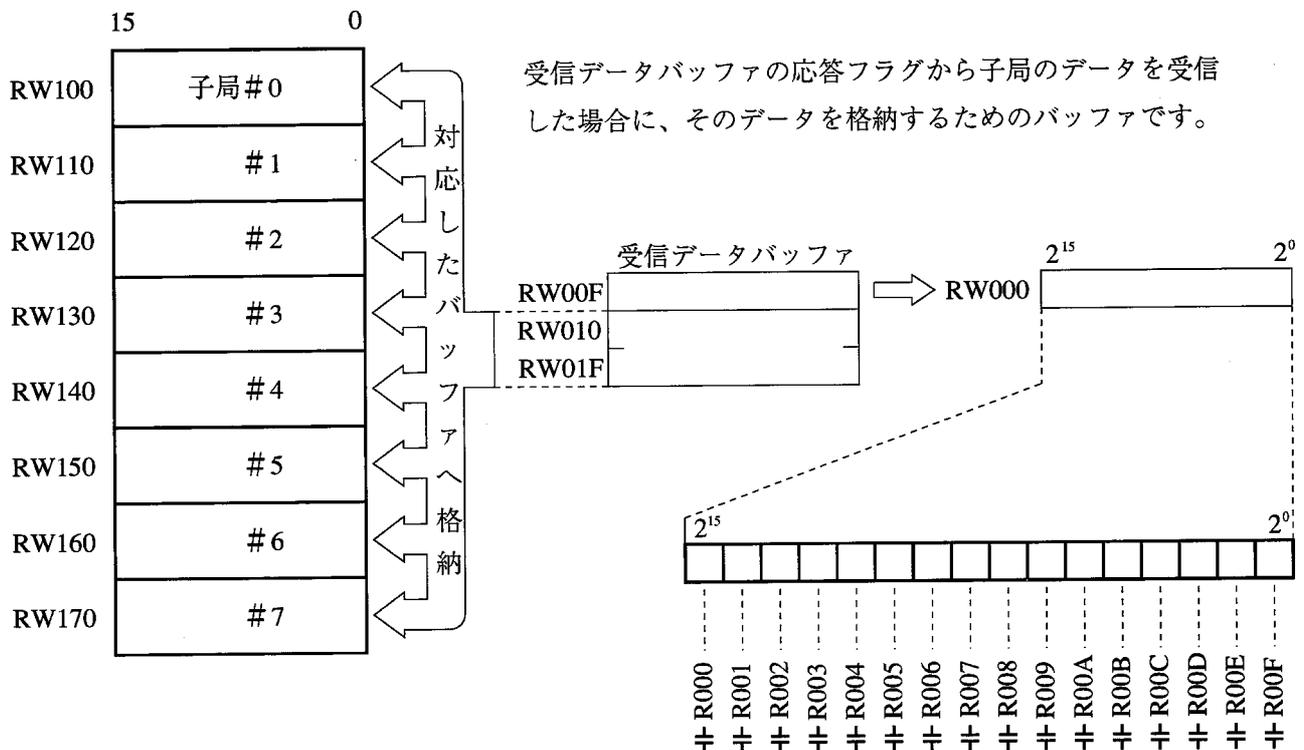
(1) 送信データバッファ (子局へ送信データを設定します。)



(2) 受信データバッファ (子局データ受信バッファ)

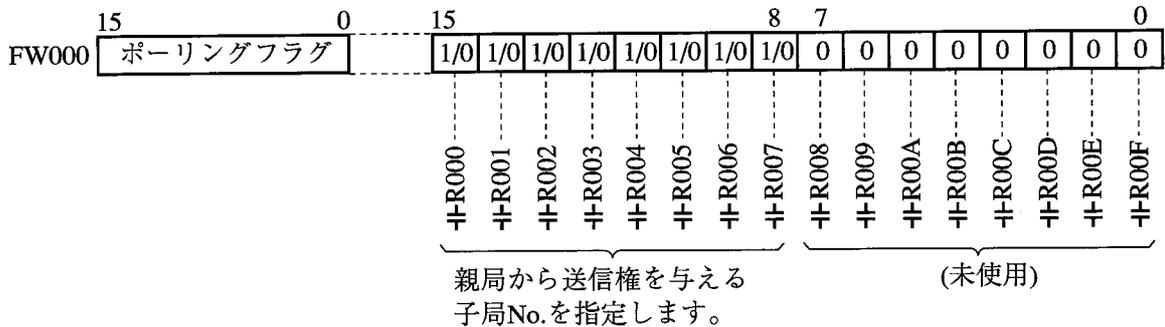


(3) 子局データ受信バッファ

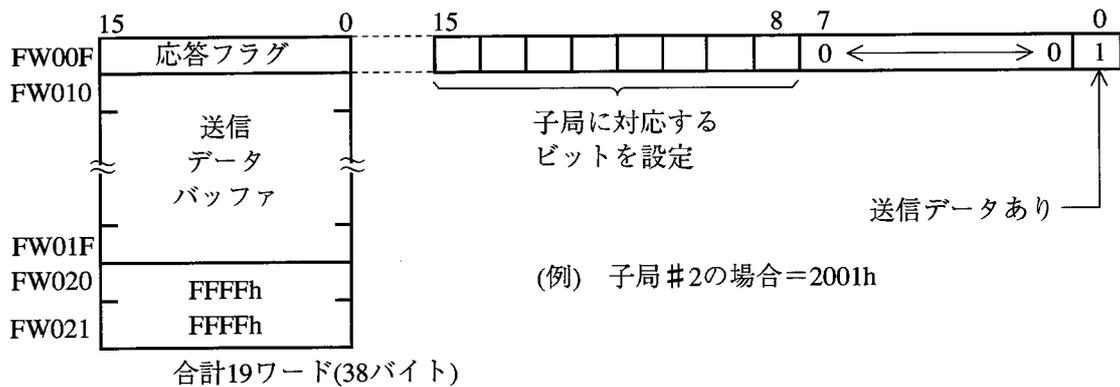


・子局側

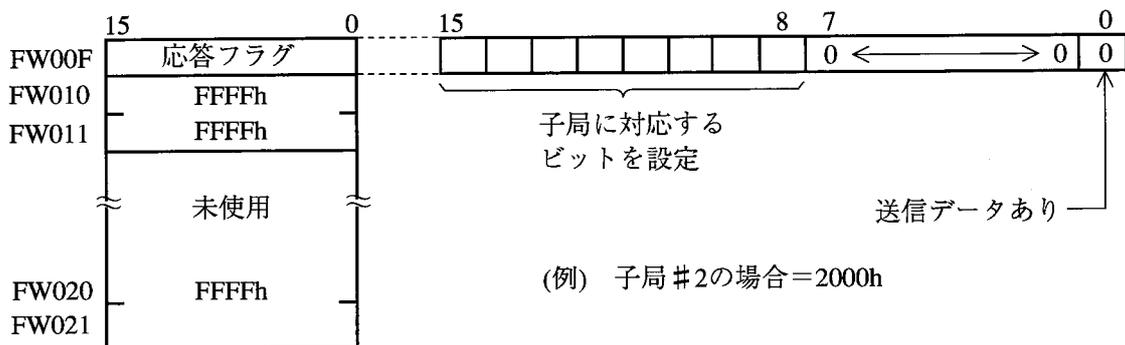
(1) ポーリングデータ受信バッファ



(2) データ送信バッファ



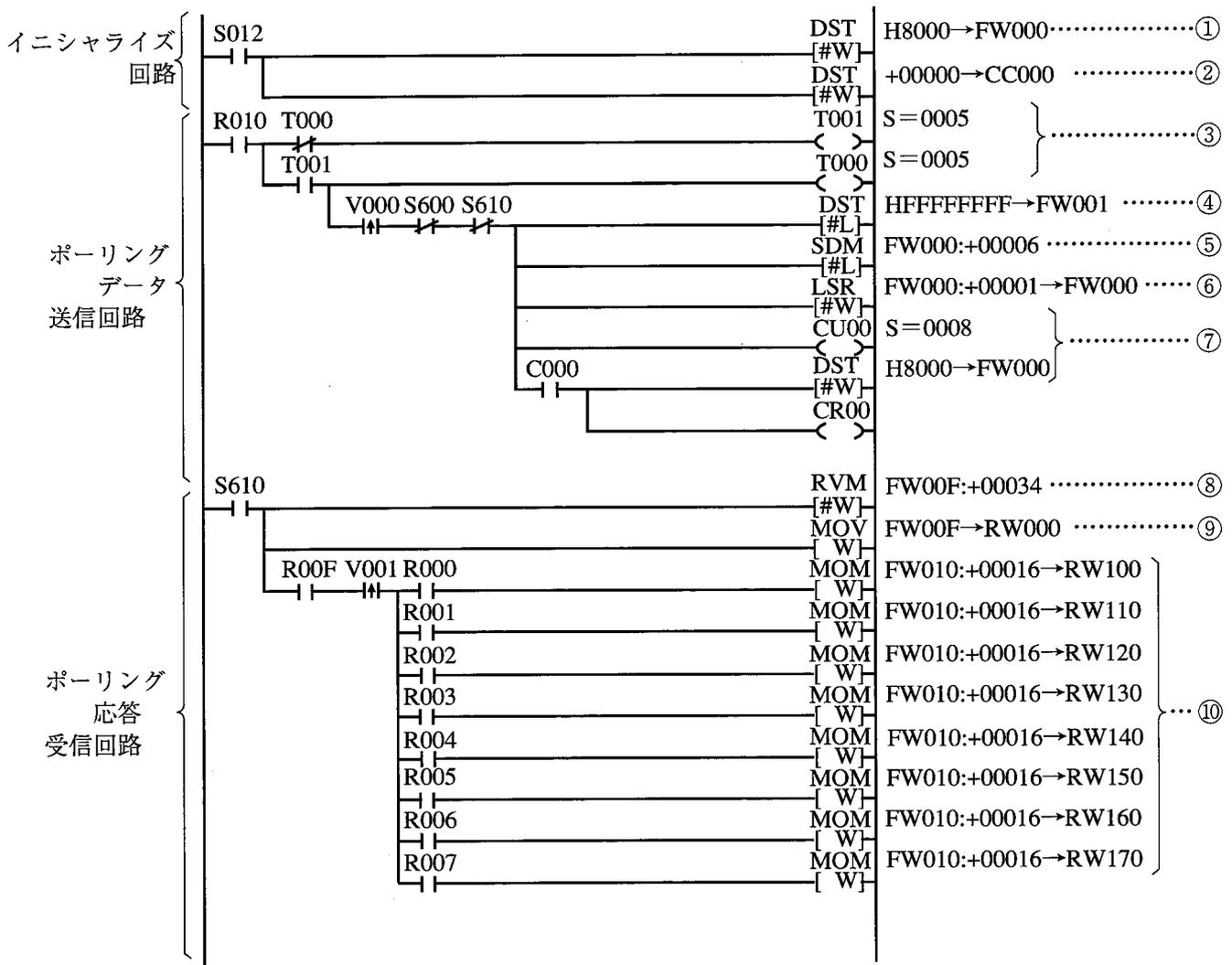
(注) データなしの場合は次のようになります。



## 6 プログラム例

### 6.3.5 ラダー回路によるプログラム

#### (1) 親局側のラダー回路



・イニシャライズ回路：ポリング用ワークの初期化は、S012を用いてSTOP→RUN時に行います。

①…ポリングデータを8000hに設定します。

②…⑦で使用了したカウンタ (CU00) のカウンタ値をクリアします。



・ S012 は STOP → RUN 時にパルスを出力します。

・ポーリングデータ出力回路： $\downarrow$  R010がONのとき、約1秒ごとにポーリングデータを送信します。

- ③… $\downarrow$  R010がONのとき、1秒周期のパルスを発生します。
- ④…送信バッファにECDを4文字 (FFh, FFh, FFh, FFh) を設定します。
- ⑤…送信バッファ (FW000~FW002) の6バイトを送信します。
- ⑥…次回のポーリングデータを設定します。
- ⑦…ポーリング相手は8台なので最終ポートへ送信した場合  
 ( $\downarrow$  C000がONの場合) ポーリングデータおよびカウンタをイニシャライズし、先頭のポートからポーリングを繰り返します。

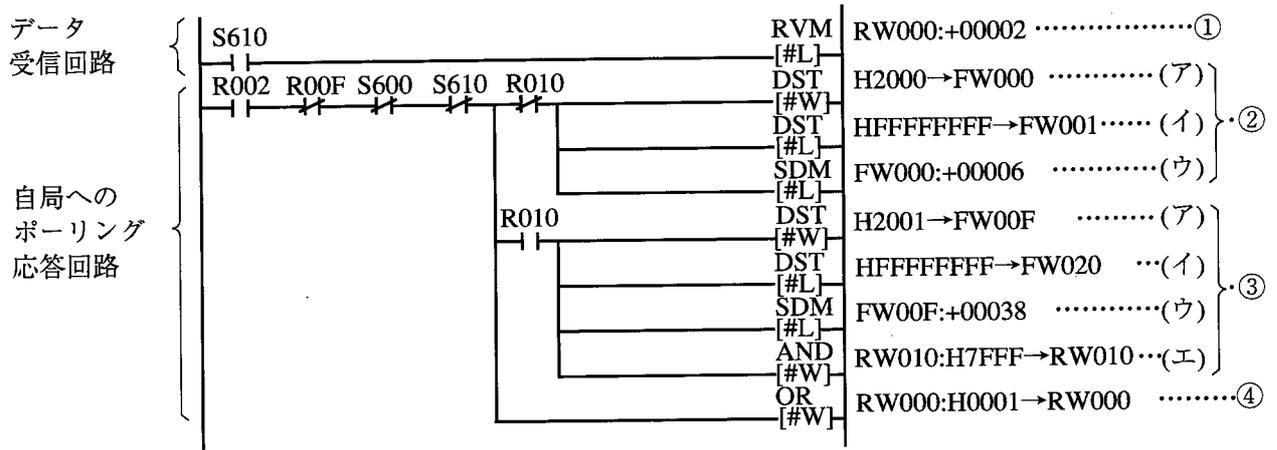


・ $\downarrow$  S600は送信中フラグ、 $\downarrow$  S610は受信中フラグで送信可かつ受信中ではない場合に処理を行います。

・ポーリング応答受信回路：子局からのポーリング応答を受信し、データありの場合対応したバッファへ受信データを設定します。

- ⑧…受信フラグ ( $\downarrow$  S610) がONのとき、受信データ (最大34バイト) を受信バッファ (FW00F~FW01F) へ取込みます。
  - ⑨…受信データの先頭1ワードのフラグデータをRW000 ( $\downarrow$  R000~R00F) へ設定します。
  - ⑩…フラグデータから子局番号 ( $\downarrow$  R000~R007に対応)、データ有無 ( $\downarrow$  R00F) を判定しデータありの場合、受信データを子局番号に対応したバッファ (R100~R17F) へ設定します。
- (注)  $\downarrow$  S610は受信フラグでONのとき、受信データありとなります。

(2) 子局側のラダー回路 (子局ナンバ2)



- ①…親局からのポーリングデータを受信します。
- ②…自局送信データなしの場合、ポーリング応答のみ送信します。  
 (ア) 自局No. (2000h) 送信バッファ先頭に設定します。  
 (イ) ECD (エンドコード) を4文字分設定します。  
 (ウ) 自局No.とECDの6バイトを送信します。

③…自局送信データありの場合のポーリング応答

- (ア) 自局No.をデータありとして設定します。
- (イ) ECDを送信データ (FW010~FW01F) の次に設定します。
- (ウ) フラグデータ+送信データ+ECDの計38バイトを送信します。
- (エ) データ送信後送信データありフラグ (|R010) をOFFします。

④…送信済フラグ (|R00F) をONします。

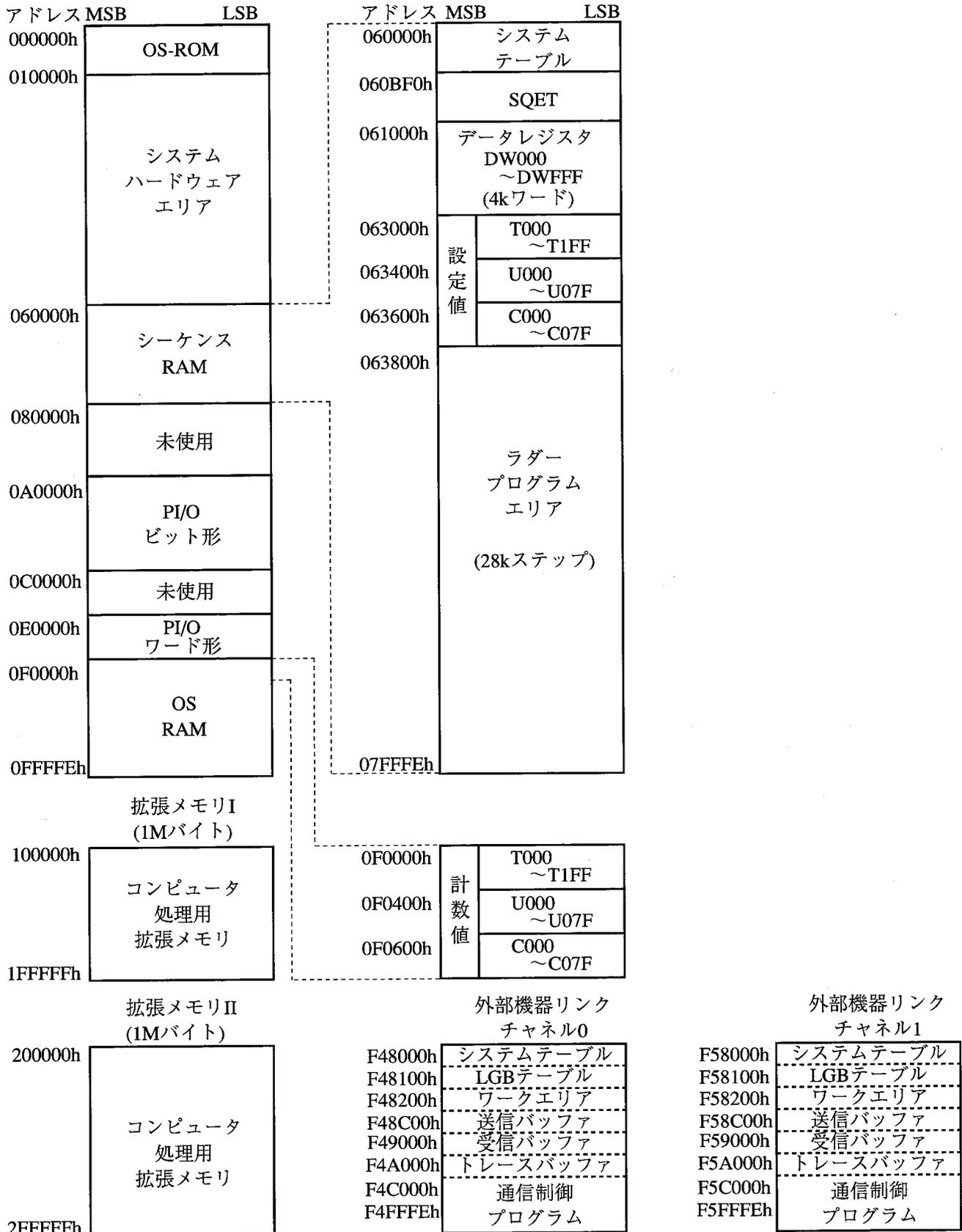
(注) ここで、このプログラム例を子局ナンバ2以外の子局で使用する場合は、上記回路中のR002およびH2000 (2000h)、h2001 (2001h) を子局No.に対応して変更する必要があります。



・このプログラムは、理解しやすいことを目的に書かれていますので、実用的に良いプログラムとは言えません。  
実用プログラムでは、送受信のシステムレジスタ (S) のエラーチェックをする必要があります。

# 付 録

付録A CPUのメモリマップ



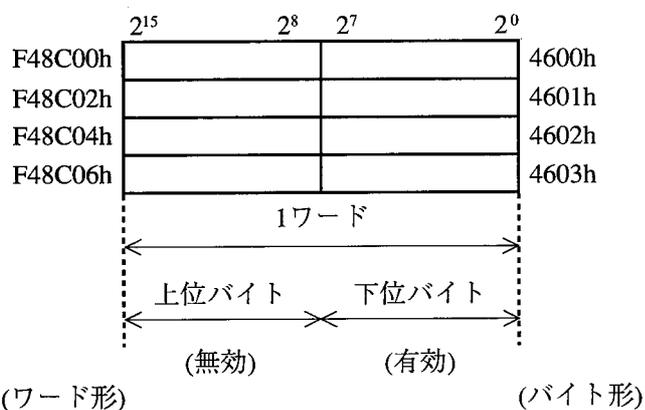
付録B 外部機器リンクモジュールのメモリマップ

外部機器リンク  
メイン(MAIN)モジュール

CPU アドレス	外部機器リンク MCSアドレス (チャンネル0)
$2^{15}$	$2^0$
F48000h	4000h
F48100h	4080h
F48200h	4100h
F48C00h	4600h
F49000h	4800h
F4A000h	5000h
F4C000h	6000h
F4FFFEh	7FFFh

外部機器リンク  
サブ(SUB)モジュール

CPU アドレス	外部機器リンク MCSアドレス (チャンネル1)
$2^{15}$	$2^0$
F58000h	4000h
F58100h	4080h
F58200h	4100h
F58C00h	4600h
F59000h	4800h
F5A000h	5000h
F5C000h	6000h
F5FFFEh	7FFFh

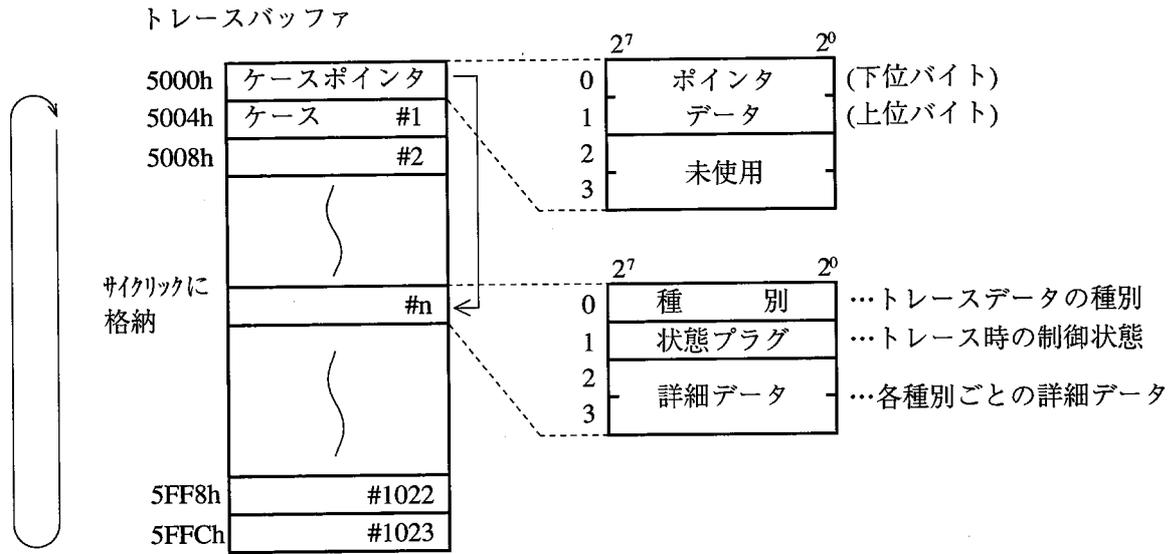


- 外部機器リンクモジュールは、通信用マクロプロセッサとして、8085 (8ビット) を搭載していて、CPUの68000のシステムバスとは、デュアルポートRAMメモリを用いて、データを交信しています。
- CPU側のメモリは、68000のワード (16ビット) 形で、下位バイト ( $2^7 \sim 2^0$ ビット) のみ有効です。
- CPUアドレス方式で、データを読出し/書込みずらいたため「外部機器リンクサポートMCS」を準備しました。
- この「外部機器リンクサポートMCS」は、外部機器リンクモジュール単位 (チャンネル0, 1) で、ビットマイクプロセッサのメモリを読出し、書込みができます。
- タスクなどから外部機器リンクモジュールのメモリをアクセスするときは、バイトアクセスしないでください。

付録C トレースバッファ

C. 1 トレースバッファの構成

トレースバッファは、4バイト/ケースのトレースデータエリアで構成され、ポインタによりサイクリックにトレースデータを格納します。



トレースデータは、ケース#1から順番に格納され、最終ケースに格納されると、次のデータはケース#1へ格納されます。

C. 2 トレースデータの種別

種別はトレースデータの大分類を示しています。このデータはASCIIコードで格納されているため、PSEの「外部機器リンクサポートMCS」で読出すことができます。

種別		内 容
ASCII	HEX	
'R'	52	データ受信を示します。
'S'	53	データ送信を示します。
'D'	44	CPUから通信制御プログラムへの送信等の要求を示します。
'I'	49	通信制御プログラムからCPUへの受信終了などの報告を示します。
'E'	45	CD入力の変化による受信可否状態を示します。
'B'	42	CSまたはDRによる送信の中断/再開の状態を示します。

C.3 状態フラグ

データトレース時の通信制御プログラムの状態をビット対応で設定します。

(MSB) 2<sup>7</sup> 2<sup>6</sup> 2<sup>5</sup> 2<sup>4</sup> 2<sup>3</sup> 2<sup>2</sup> 2<sup>1</sup> 2<sup>0</sup> (LSB)

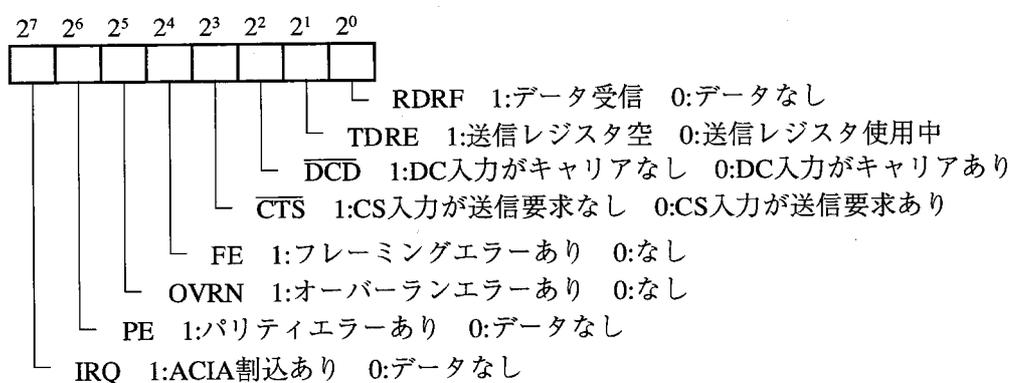


ビット	項 目	ビット状態	
		'1'	'0'
0	将来拡張用	/	/
1			
2			
3			
4	受信の可/否	現在、CD入力がかリヤなしのため、受信できません。	CD入力がかリヤありのため受信できます。
5	送信中断	中断コード受信、CS入力がか送信要求なし、またはDR入力がノットレディのため送信を中断しています。	中断していません。
6	テキスト受信	現在、テキストデータ受信中です。	受信中ではありません。
7	データ送信	現在、データ送信中です。	送信中ではありません。

### C.4 トレースの詳細データ

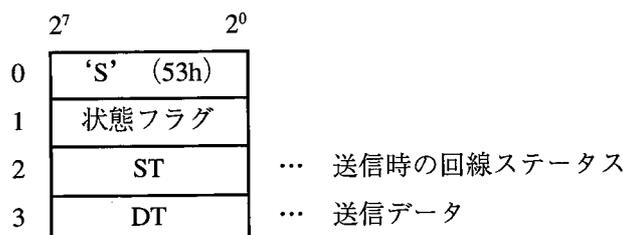
#### (1) 'R' (52h) データ受信

データを受信したことを示します。



#### (2) 'S' (53h) データ受信

データを外部機器へ送信したことを示します。



## (3) 'D' (44h) CPUから通信制御プログラムへの送信などの要求

CPU側 (サブOSおよびアプリケーションプログラム) からの通信制御プログラムに対する送信要求および受信報告に対する応答を示します。

No.	トレスデータ (HEX)				説 明	データ種別(※4)			応答 有/無
	1	2	3	4		要求	報告	応答	
1	D		80	80	通信制御プログラムのソフトウェアリセット要求	○			
2	D		01	00	データ送信要求	○			あり
3	D		40	00	ハードウェア状態の読出要求	○			あり
4	D		40	01	RS-422送信ゲートコントロールの要求	○			あり
5	D		40	02	DTR (ER) の設定要求	○			あり
6	D		40	03	RTS (RS) の設定要求	○			あり
7	D		80	00	通信制御プログラムからのリセット報告に対する応答 (I-14)			○	
8	D		01	01	送信終了報告に対する応答 (I-1)			○	
9	D		01	83	中断タイムアウト報告に対する応答 (I-2)			○	
10	D	(※2)	01	84	CS入力送信不可能による送信 中断タイムアウト報告に対する応答 (I-3)			○	
11	D		01	85	DR入力ノットレディによる送信 中断タイムアウト報告に対する応答 (I-4)			○	
12	D		02	00	データ受信報告に対する応答 (I-5)			○	
13	D		02	80	受信パリティエラー報告に対する応答 (I-6)			○	
14	D		02	81	受信オーバーランエラー報告に対する応答 (I-7)			○	
15	D		02	82	受信フレーミングエラー報告に対する応答 (I-8)			○	
16	D		02	83	受信タイムアウトエラー報告に対する応答 (I-9)			○	
17	D		02	84	ASCII変換エラー報告に対する応答 (I-10)			○	
18	D		02	85	ECDエラー報告に対する応答 (I-11)			○	
19	D		02	86	BCCチェックエラー報告に対する応答 (I-12)			○	
20									

(※3)

(※1)



※1 ( ) 内の数値は 'I' の項に NO. を示し、その応答がどの方向に対するものか示しています。

※2 「C. 3 状態フラグ」参照

※3 D = 44h : Device status word、デバイスステータスワード

※4 要求とはサブOS → 通信制御プログラムに対する要求  
報告とは通信制御プログラム → サブOS に対する報告

付 録

(4) 'I' (49h) 通信制御プログラムからCPUへの受信終了などの報告  
 通信制御プログラム側からCPU (サブOSおよびアプリケーションプログラム) への受信終了報告、エラー報告などを示します。

No.	トレースデータ (HEX)				説 明	データ種別(※4)			応答 有/無
	1	2	3	4		要求	報告	応答	
1	I		01	01	送信正常終了		○		あり
2	I		01	83	送信中断コードタイムアウトエラー		○		あり
3	I		01	84	CS入力 that 送信不可能による送信 中断タイムアウトエラー		○		あり
4	I		01	85	DR入力 that ノットレディによる送信 中断タイムアウトエラー		○		あり
5	I		02	00	データ正常受信		○		あり
6	I		02	80	受信データパリティエラー		○		あり
7	I		02	81	受信データオーバランエラー		○		あり
8	I		02	82	受信データフレーミングエラー		○		あり
9	I		02	83	受信タイムアウトエラー		○		あり
10	I		02	84	受信データASCII変換エラー		○		あり
11	I		02	85	受信データECDエラー		○		あり
12	I		02	86	受信データBCCエラー		○		あり
13	I								
14	I		80	00	GRイニシャライズ		○		あり
15	I		80	F0	ROMチェックサムエラー		○		なし
16	I	(※2)	80	F1	通信制御プログラムオールクリア		○		なし
17	I		80	F2	通信制御プログラム未ローディング		○		なし
18	I		80	F3	通信制御プログラムチェックサムエラー		○		なし
19	I		80	F4	通信制御プログラムダウン		○		なし
20	I		80	F5	通信制御プログラムメモリパリティエラー		○		なし
21	I		01	00	送信要求に対する送信開始応答 (D-2)			○	
22	I		01	80	送信要求に対し、現在受信中のため、 送信不可能応答 (D-2)			○	
23	I		01	81	送信要求に対し、現在送信中のため、 送信不可能応答 (D-2)			○	
24	I		01	82	送信要求に対し、ハードウェア ノットレディのため、送信不可能応答 (D-2)			○	
25	I		40	00	ハードウェア状態読出要求に対する応答 (D-3)			○	
26	I		40	01	RS-422送信ゲートコントロール 要求に対する応答 (D-4)			○	
27	I		40	02	ERのレディ/ノットレディセット 要求に対する応答 (D-5)			○	
28			40	03	RSの送信要求有/無設定要求に対する応答 (D-6)			○	

(※3)

(※1)



※1 ( ) 内の数値は 'D' の項のNo. を示し、その応答がどの報告に対するものかを示しています。

※2 「C.3 状態フラグ」参照

※3 I = 49h : Interrupt status word、インタラプトステータスワード

※4 要求とはサブOS → 通信制御プログラムに対する要求  
 報告とは通信制御プログラム → サブOS に対する報告



付録D JIS 7単位コード表 (C6220)

								b <sub>8</sub>	0	0	0	0	0	0	0	0
								b <sub>7</sub>	0	0	0	0	1	1	1	1
								b <sub>6</sub>	0	0	1	1	0	0	1	1
								b <sub>5</sub>	0	1	0	1	0	1	0	1
b <sub>8</sub>	b <sub>7</sub>	b <sub>6</sub>	b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>		0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	NUL	(TC <sub>7</sub> )DLE	SP	0	@	P	,	p	
0	0	0	1	1	1	1	1	(TC <sub>1</sub> )SOH	DC <sub>1</sub>	!	1	A	Q	a	q	
0	0	1	0	0	0	0	0	(TC <sub>2</sub> )STX	DC <sub>2</sub>	"(1)	2	B	R	b	r	
0	0	1	1	1	1	1	1	(TC <sub>3</sub> )ETX	DC <sub>3</sub>	#	3	C	S	c	s	
0	1	0	0	0	0	0	0	(TC <sub>4</sub> )EOT	DC <sub>4</sub>	\$	4	D	T	d	t	
0	1	0	1	1	1	1	1	(TC <sub>5</sub> )ENQ	(TC <sub>8</sub> )NAK	%	5	E	U	e	u	
0	1	1	0	0	0	0	0	(TC <sub>6</sub> )ACK	(TC <sub>9</sub> )SYN	&	6	F	V	f	v	
0	1	1	1	1	1	1	1	BEL	(TC <sub>10</sub> )ETB	'(2)	7	G	W	g	w	
1	0	0	0	0	0	0	0	FE <sub>0</sub> (BS)	CAN	(	8	H	X	h	x	
1	0	0	1	1	1	1	1	FE <sub>1</sub> (HT)	EM	)	9	I	Y	i	y	
1	0	1	0	0	0	0	0	FE <sub>2</sub> (LF)	SUB	* ;(6)	J	Z	j	z		
1	0	1	1	1	1	1	1	FE <sub>3</sub> (VT)	ESC	+ ;(7)	K	[	k	{		
1	1	0	0	0	0	0	0	FE <sub>4</sub> (FF)	IS <sub>4</sub> (FS)	,(3) <	L	¥	l			
1	1	0	1	1	1	1	1	FE <sub>5</sub> (CR)	IS <sub>3</sub> (GS)	-(4) =	M	]	m	}		
1	1	1	0	0	0	0	0	SO	IS <sub>2</sub> (RS)	.(5) >	N	^	n	-		
1	1	1	1	1	1	1	1	SI	IS <sub>1</sub> (US)	/ ?	0	_(8)	o	DEL		

- (1) 引用符
- (2) アポストロフイ
- (3) コンマ
- (4) マイナス
- (5) ピリオド
- (6) コロン
- (7) セミコロン
- (8) アンダーライン



## 付録 F 制御符号の説明

制御符号	コード	制御符号名	意 味
NUL	00	Null	空
SOH	01	Start of Heading	ヘディング開始
STX	02	Start of Text	テキスト開始
ETX	03	End of Text	テキスト終了
EOT	04	End of Transmission	伝送終了
ENQ	05	Enquiry	問い合わせ
ACK	06	Acknowledge	肯定応答
BEL	07	Bell	ベル
BS	08	Backspace	バックスペース (1文字後退する。)
HT	09	Horizontal Tablation	水平タブ
LF/NL	0A	Line Feed/New Line	改行/復改 (復帰・改行)
VT	0B	Vertical Tablation	垂直タブ
FF	0C	Form Feed	改頁
CR	0D	Carriage Return	復帰
SO	0E	Shift Out	シフトアウト
SI	0F	Shift In	シフトイン
DLE	10	Data Link Escape	データリンクでの拡張
DC1	11	Device Control 1 (X-ON)	装置制御1 (送信を開始する要求に使用)
DC2	12	Device Control 2	装置制御2
DC3	13	Device Control 3 (X-OFF)	装置制御3 (送信を止める要求に使用)
DC4	14	Device Control 4	装置制御4
NAC	15	Negative Acknowledge	否定応答
SYN	16	Synchronous Idle	同期文字
ETB	17	End of Transmission Block	伝送ブロック終了
CAN	18	Cancel	取り消し
EM	19	End of Medium	媒体終端
SUB	1A	Substitute Character	置換え文字
ESC	1B	Escape	拡張 (画面やグラフィックなどの制御コードの 拡張に使用する。)
FS	1C	File Separator	ファイルセパレイタ
GS	1D	Group Separator	グループセパレイタ
RS	1E	Record Separator	レコードセパレイタ
US	1F	Unit Separator	ユニットセパレイタ
SP	20	Space	空白、ブランク、スペース
DEL	7F	Delete	抹消

## 付録 G 略号一覧表

略 号	意 味
ACIA	Asynchronous Communications Interface Adapter
ASCII	American Standard Code for Information
BCC	Block Check Character
BPS	Bits Per Second
CD	data Carrier Detect
CPMS	Compact Process Monitor System
CPU	Central Processing Unit
CRT	Cathode Ray Tube
CS	Clear to Send
DR	Data set Ready
ECD	End Code
EIA	Electronic Industries Association
EOR	Exclusive OR
ER	Equipment Ready
FE	Framing Error
FG	Frame Ground
GR	General Reset
IRQ	Interrupt Request
LED	Light Emitting Diode
LGB	Line Group Block
MCS	Man-machine Communication System
OVRN	Overflow error
PCs	Programable controllers
PE	Parity Error
RD	Receive Data
RS	Request to Send
SCD	Start Code
SD	Send data
SG	Signal Ground
SHD	Shield
TERM	Termination Resistance
UFET	User Function Edition Table
WDT	Watch Dog Timer



ご利用者各位

〒101-10

東京都千代田区神田駿河台4丁目6番地

株式会社 日立製作所

産業機器事業部 産業システム部 制御システムグループ

電話 (03) 3258-1111 (大代表)

お 願 い

各位にはますますご清栄のことと存じます。

さて、この資料をより良くするために、お気付きの点はどんなことでも結構ですので、下欄にご記入の上、当社営業担当または当社所員に、お渡しくださいますようお願い申し上げます。なお、製品開発、サービス、その他についてもご意見を併記して頂ければ幸甚に存じます。

ご住所 〒 _____
貴会社名 (団体名) _____
芳 名 _____
ご意見欄 _____