

HITACHI

S10mini.

S10mini  
ハードウェアマニュアル

オプション  
*RS-232C*

SMJ-1-105 (C)

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。

1998年11月 (第1版) SMJ-1-105 (A) (廃版)  
2000年 4月 (第2版) SMJ-1-105 (B) (廃版)  
2008年 4月 (第3版) SMJ-1-105 (C)

- 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写することは、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

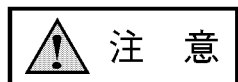
取り付け、運転、保守・点検の前に必ずこのマニュアルとその他の付属書類をすべて熟読し、正しくご使用ください。機器の知識、安全の情報そして注意事項のすべてについて熟読してご使用ください。また、このマニュアルは最終保守責任者のお手元に必ず届くようにしてください。

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**

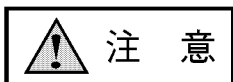
：取り扱いを誤った場合に、危険な状況が起こりえて、死亡または重傷を受ける可能性が想定される場合。



**注意**

：取り扱いを誤った場合に、危険な状況が起こりえて、中程度の傷害や軽傷を受ける可能性が想定される場合および物的損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、誤動作の原因になります。
- マニュアルに従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、誤動作の原因になります。
- 電線くずなどの異物が入らないようにしてください。  
火災、故障、誤動作の原因になります。

## 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、誤動作の恐れがあります。



### 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、感電の恐れがあります。
- ケーブルは強電機器からの配線と同一付線しないでください。強電機器の配線と同一付線した場合、誤動作の原因になります。

 注 意

- RS-232Cモジュールと外部機器の信号用接地（SG）は、必ずインタフェースケーブルで接続してください。
- インタフェースケーブルのシールド接地端子は、接地電位が同一になる場合、RS-232Cモジュール、外部機器の両側とも接地してください。耐ノイズ性が向上します。接地電位が異なる場合は、RS-232Cモジュール側だけ接地してください。
- RS-232Cモジュールのシールド用接地（SHD）端子は、CPUモジュール端子台のシールド用接地（SHD）端子を使用してください。



### 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。  
感電の恐れがあります。
- 非常停止回路、インタロック回路などはプログラマブルコントローラの外部で構成してください。  
プログラマブルコントローラの故障によって、機械の破損や事故の恐れがあります。

#### 注 意

- 運転中のプログラム変更、強制出力、RUN、STOPなどの操作は十分安全を確認して行ってください。  
操作ミスによって、機械の破損や事故の恐れがあります。
- 電源投入順序に従って投入してください。  
誤動作によって、機械の破損や事故の恐れがあります。
- このモジュールの近くでは、トランシーバ、携帯電話などを使用しないでください。近くでトランシーバ、携帯電話などを使用しますとノイズによって誤動作、モジュールダウンとなる恐れがあります。

#### 注 意

プログラムの例は、理解しやすいことを目的に書いていますので、実用のプログラムでは、送信ハンドラのリターンコードおよびシステムレジスタ (S) のエラーチェックを行ってください。

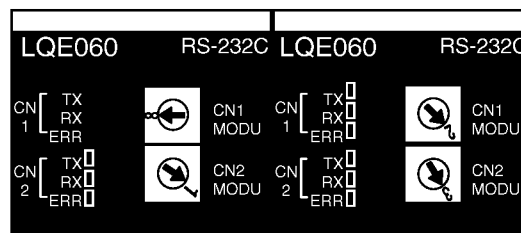
## ⊘ 禁 止


システムが誤動作します。チャンネルNo.およびプロトコル設定スイッチ（MODU NO）について、次の事項は禁止です。

- 通電中には設定しないでください。設定は、電源を落とした状態で行ってください。
- 無手順の演算ファンクションシステムとタスクシステムを混在した状態で設定しないでください。CPUユニット単位でどちらか一方だけで使用してください。
- チャンネルNo.は重複しないでください。

<設定例> チャンネルNo.が重複しないように設定します。

	MODU NO設定	通信方式	チャンネル No.
1枚目-CN1	8	H-73387°モコル	#0
1枚目-CN2	1	無手順-演算ファンクション	#1
2枚目-CN1	2	無手順-演算ファンクション	#2
2枚目-CN2	3	無手順-演算ファンクション	#3



 注 意

- RS-232Cモジュールは、アプリケーションの利用形態に合わせて以下のどれかを選択し使用してください。また、無手順一タスクシステムを選択しCPUがLQP000の場合は、オプションの拡張メモリモジュールが必要になります。
  - ・無手順一演算ファンクションシステム
  - ・無手順一タスクシステム
  - ・H-7338プロトコル
- 無手順では、接続される外部機器の仕様に合わせLGBの編集が必要になります。H-7338プロトコルのLGBの設定は固定です。H-7338で使用する場合、LGBツールなどで変更すると正常に動作しませんので注意してください。
- 受信タスク登録テーブルは、一括セーブで自動セーブされませんので、ユーザがセーブするときにアドレスを指定してください。  
LGBテーブルは、一括セーブで自動セーブされます。
- ツールからの編集後、あるいは、一括ロードによって設定したLGBテーブルと受信タスク登録テーブルは、設定後に実施するリセット前またはリセット中に停復電があると、編集や一括ロードで設定した内容ではなく、RS-232C内のフラッシュメモリに書き込まれている内容が有効になります。その場合は、再度、編集または一括ロード後にリセットを実施してください。

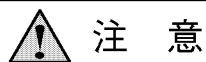


#### 4. 保守について



禁 止

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



注 意

- 静電気によってモジュールが破損する恐れがあります。作業を行う前に、人体の静電気を放電してください。
- 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、故障の原因になります。

## 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、弊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。

このマニュアルは、以下のハードウェアの説明をしたものです。

<ハードウェア>

RS-232C (LQE060)

変更内容 (SMJ-1-105(C))

追加・変更内容	ページ
補足資料 モジュールの交換、増設を追加	96

上記追加変更の他に、記述不明瞭な部分、単なる誤字・脱字などについては、お断りなく訂正しました。



## はじめに

このたびは、CPUオプションRS-232Cモジュールをご利用いただきましてありがとうございます。

この「ハードウェアマニュアル オプションRS-232C」は、RS-232Cモジュールの取扱いについて述べたものです。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。

なお、S10miniシリーズの製品には、標準仕様品と耐環境仕様品があります。

耐環境仕様品は、標準仕様品と比べ部品のメッキ厚、コーティング等が強化されています。

耐環境仕様品の型式は、標準仕様品型式の後に“-Z”が付いています。

例：標準仕様品　：LQE060

耐環境仕様品：LQE060-Z

マニュアルは、共通になっており、マニュアルに記載しているモジュール型式は、標準仕様品を代表に記してあります。

耐環境仕様品をご使用の場合も、このマニュアルに従い、正しくご使用いただくようお願いいたします。

### <商標について>

- ・ Microsoft® Windows® operating system, Microsoft® Windows® 95 operating system, Microsoft® Windows® 98 operating system, Microsoft® Windows® 2000 operating system, Microsoft® Windows® XP operating systemは、米国Microsoft Corporationの米国およびその他の国における登録商標です。
- ・ Ethernet®は米国Xerox Corp.の登録商標です。

# 目 次

1	ご使用にあたり .....	1
1.1	CPUマウントベース .....	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
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	ソフトウェア仕様 .....	7
3	各部の名称と機能、配線 .....	9
3.1	各部の名称と機能 .....	10
3.2	配 線 .....	11
4	オペレーション .....	17
4.1	システム立上げ .....	18
4.2	LGB通信制御テーブルの編集 .....	19
4.3	LGBに設定する内容 .....	20
5	プログラミング .....	35
5.1	ソフトウェア構成 .....	36
5.2	システムレジスタ .....	38
5.2.1	送信情報 .....	38
5.2.2	受信情報 .....	39
5.3	送受信ハンドラ .....	40
5.3.1	演算ファンクション .....	40
5.3.2	サブルーチン .....	43
5.4	受信データの取込み方 .....	49
5.5	ソフトウェアによるハードウェア制御 .....	50
6	プログラム例 .....	53
6.1	RS-232Cによるプリンタとの接続例 .....	54
6.1.1	概 要 .....	54

6.1.2	システムの構成	54
6.1.3	プリンタの印字フォーマット	54
6.1.4	プログラムの構成	55
6.1.5	ラダープログラムとのリンケージテーブル構成	56
6.1.6	RS-232Cモジュール	58
6.1.7	LGBの設定	59
6.1.8	C言語プログラムのフローチャート	60
6.1.9	C言語のプログラム例	61
6.1.10	C言語プログラムの作成と登録	63
6.1.11	ラダープログラム	65
6.2	パソコンによるプログラムローディング	66
6.2.1	システム構成	66
6.2.2	プログラム構成	66
6.2.3	モトローラ‘S’フォーマット(16ビット用)	67
6.2.4	LGBの設定	68
6.2.5	受信タスクの登録	68
6.2.6	受信タスク	69
6.2.7	C言語のプログラム例	70
6.2.8	プログラムローディング方法	72
7	保 守	73
7.1	保守点検	74
7.2	ユーザ設定項目のバックアップ	75
7.2.1	LGBテーブル、受信タスク登録テーブル、ユーザ演算ファンクション登録テーブル	75
7.2.2	モジュールを交換した場合	76
7.3	トラブルシューティング	77
7.3.1	CPUモジュールインディケータ表示	77
7.3.2	送信エラーコード表	78
7.3.3	受信エラーコード表	79
7.3.4	RS-232Cモジュールのメモリマップ	81
7.3.5	トレースバッファ	82
付 録		89
A.1	JIS 7単位コード表(C6220)	90
A.2	JIS 8単位コード表(C6220)	91
A.3	制御符号の説明	92

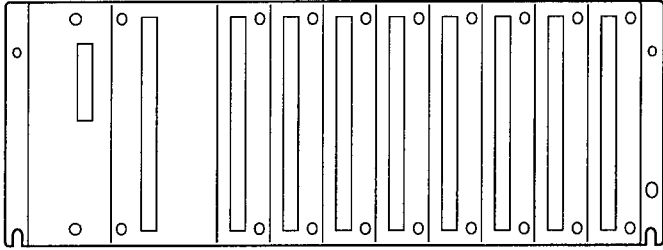
A.4	略号の説明 .....	93
A.5	トラブル調査書 .....	94
補足資料	.....	95
補足資料	モジュールの交換、増設 .....	96



# 1 ご使用にあたり

## 1 ご使用にあたり

### 1.1 CPUマウントベース

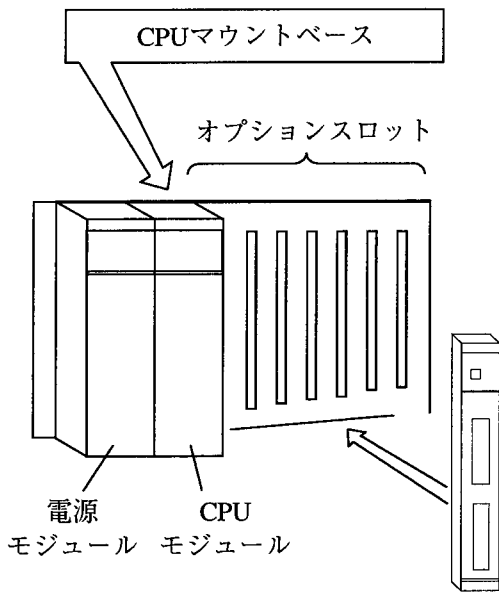


CPUマウントベースには、次の3種類があります。

- ・ 2スロットマウントベース（形式：HSC-1020）
- ・ 4スロットマウントベース（形式：HSC-1040）
- ・ 8スロットマウントベース（形式：HSC-1080）

例えば、8スロット用マウントベースの場合は、電源、CPU以外のモジュールを8モジュールまで実装することができます。

### 1.2 オプションモジュールの実装



CPUマウントベース：HSC-1080

PSスロット：電源モジュール(LQV000, LQV020, LQV100)を実装。

CPUスロット：CPUモジュール(LQP000, LQP010, LQP011, LQP120)を実装。

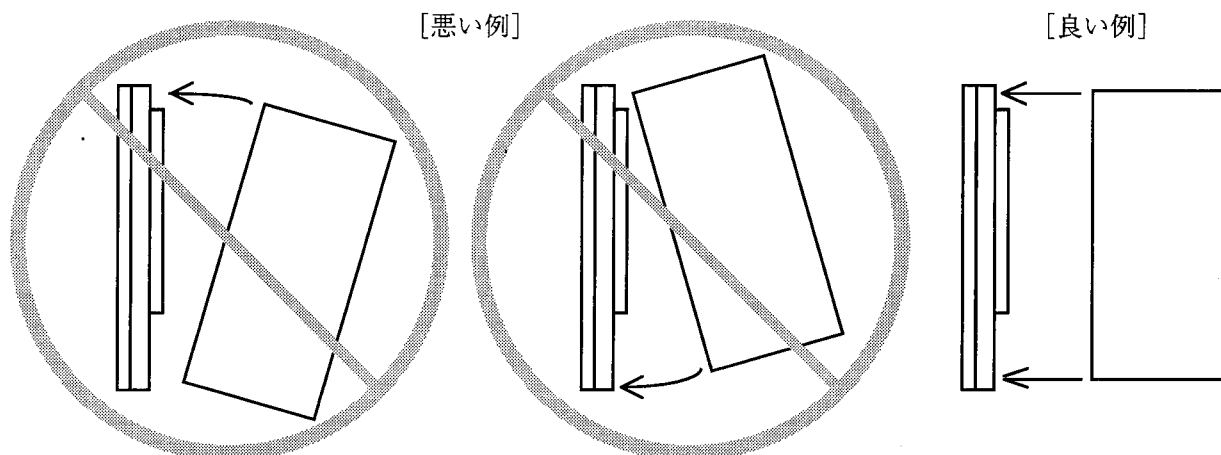
スロット0～7：CPUオプションモジュールまたは、I/Oモジュールを実装。

### ⚠ 注意

RS-232CモジュールはI/Oモジュールが間に入らないように左詰め実装してください。  
I/Oモジュールに接続される外部配線からのノイズなどにより誤動作することがあります。

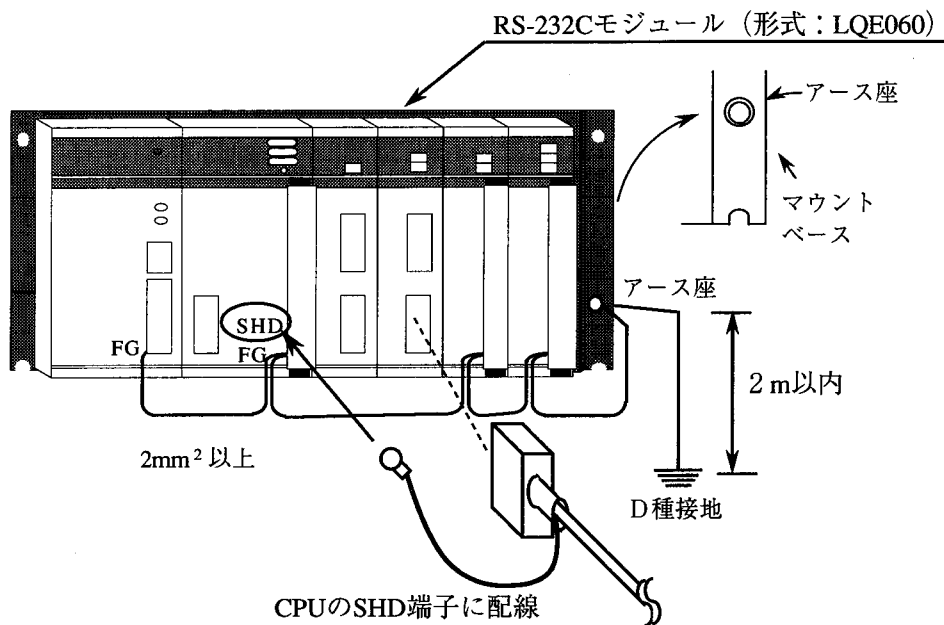
オプションモジュール実装時は、以下のことに注意してください。

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、コネクタが壊れる恐れがあります。）



## 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2 m以内としマウントベースのアース座からD種接地してください。
- アース線は、線径2 mm<sup>2</sup>以上のものを用いてください。
- RS-232Cケーブルのシールド線をCPUモジュールのSHD端子に配線してください。

# 2 仕 様

## 2 仕 様

### 2.1 用 途

RS-232Cモジュール（形式：LQE060）は、EIA RS-232C仕様に準拠し、無手順または、H-7338プロトコルによるデータ通信を行います。

### 2.2 仕 様

#### 2.2.1 システム仕様

項 目		仕 様																															
形式		LQE060																															
プロトコル		無手順/H-7338切り替え																															
RS-232Cモジュール最大実装枚数		2モジュール/CPU（左詰めで実装）																															
モジュールスロット幅		1スロット幅モジュール																															
質量		220g																															
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		EIA RS-232Cに準拠																															
伝送フレーム構成		<table border="1"> <thead> <tr> <th>ST</th> <th>DATA</th> <th>PT</th> <th>SP</th> </tr> <tr> <td>スタートビット</td> <td>データビット</td> <td>パリティビット</td> <td>ストップビット</td> </tr> </thead> <tbody> <tr> <td>スタート</td> <td>データ長</td> <td>パリティ</td> <td>ストップ</td> </tr> <tr> <td rowspan="6">無手順</td> <td rowspan="3">7ビット</td> <td>偶数</td> <td rowspan="2">2ビット</td> </tr> <tr> <td>奇数</td> </tr> <tr> <td>偶数</td> <td rowspan="2">1ビット</td> </tr> <tr> <td>奇数</td> </tr> <tr> <td rowspan="3">8ビット</td> <td>なし</td> <td>2ビット</td> </tr> <tr> <td>偶数</td> <td>1ビット</td> </tr> <tr> <td>奇数</td> <td>1ビット</td> </tr> <tr> <td>H-7338 プロトコル</td> <td>8ビット</td> <td>奇数</td> <td>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	スタート	データ長	パリティ	ストップ	無手順	7ビット	偶数	2ビット	奇数	偶数	1ビット	奇数	8ビット	なし	2ビット	偶数	1ビット	奇数	1ビット	H-7338 プロトコル	8ビット	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																												
		スタート	データ長	パリティ	ストップ																												
		無手順	7ビット	偶数	2ビット																												
				奇数																													
				偶数	1ビット																												
			奇数																														
			8ビット	なし	2ビット																												
				偶数	1ビット																												
奇数	1ビット																																
H-7338 プロトコル	8ビット	奇数	1ビット																														
伝送速度	無手順	150, 300, 600, 1200, 2400, 4800, 9600 (bps)																															
	H-7338プロトコル	19200 (bps)																															
接続ケーブル	距離	最大15m																															
	線種	シールド付きツイストペア線																															
	線径	0.08mm <sup>2</sup> 以上																															
	抵抗	229Ω/km以下（20℃）																															
	推薦品	CO-MA-VV-SB (MA) 13P×28AWG (7/0.127) 日立電線																															
接続コネクタ	種別	9ピンD-subコネクタ																															
	備考	コネクタ：HDE-CTH1（ヒロセ電機製） カバー：HDEB-9S（ヒロセ電機製）																															
ケーブル接地条件		両端接地																															

2. 2. 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		自局優先（送信中の受信要求は受けない） 他局優先（送信中でも受信要求を受けける）				
データ変換モード		テキストデータをそのまま送受信 テキストデータをASCII変換して送信およびBINARY変換して受信				
伝送ブロック構成	スタートコード	なし，1～4キャラクタ				
	テキスト	なし，1～512バイト				
	エンドコード	なし，1～4キャラクタ				
	ブロックチェックキャラクタ	なし，水平偶数パリティ，水平奇数パリティ				
送信遅延期間		<p style="text-align: center;"><math>T_0 = 0 \sim 32767 \text{ ms (1ms単位)}</math></p>				
送信中断	中断コード	なし	1キャラクタ		2キャラクタ	
再開コード	再開コード	なし	1キャラクタ	2キャラクタ	1キャラクタ	2キャラクタ
送信中断監視時間		<p style="text-align: center;"><math>T_0 = 0 \sim 3276.7 \text{ s (100ms単位)}</math></p>				
受信監視時間		<p style="text-align: center;"><math>T_0 = 0 \sim 3276.7 \text{ s (100ms単位)}</math></p>				
送信要求 (RS) 出力		送信要求出力 (RS端子はON固定となります。) 送信要求出力なし				
データ端末レディ (ER) 出力		ノットレディ出力 レディ出力 (ER端子はON固定となります。)				
データセットレディ (DR) 入力		チェックなし チェックあり				
送信バッファ容量		512バイト				
受信バッファ容量		テキスト語数256バイト以下の場合……8バッファ テキスト語数256バイト以上の場合……4バッファ				

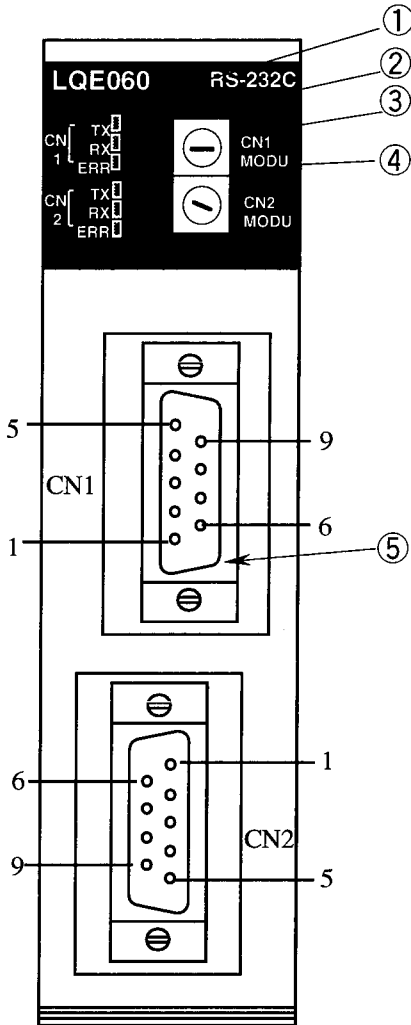
<このページは余白です>



### 3 各部の名称と機能、配線

### 3 各部の名称と機能、配線

#### 3.1 各部の名称と機能



No.	名称	機能
①	TX LED	データ送信時に点灯します。
②	RX LED	データ受信時に点灯します。
③	ERR LED	ハードウェア異常時に点灯します。
④	チャンネルNo.および プロトコル設定 スイッチ	チャンネルNo.および通信方式の設定を行います。設定はCPUリセット後有効になります。

MODU NO	通信方式	チャネル No.
0	無手順-演算ファンクションシステム	#0
1	無手順-演算ファンクションシステム	#1
2	無手順-演算ファンクションシステム	#2
3	無手順-演算ファンクションシステム	#3
4	無手順-タスクシステム	#0
5	無手順-タスクシステム	#1
6	無手順-タスクシステム	#2
7	無手順-タスクシステム	#3
8	H-7338 <sup>o</sup> プロトコル	#0
9	H-7338 <sup>o</sup> プロトコル	#1
A~F	保守用(設定するとエラーになります)	

CPUモジュールがLQP000で、タスクシステムを使用する場合は、拡張メモリモジュールが必要です。

#### 禁止

システムが誤動作します。チャンネルNo.およびプロトコル設定スイッチ (MODU NO) について、次の事項は禁止です。

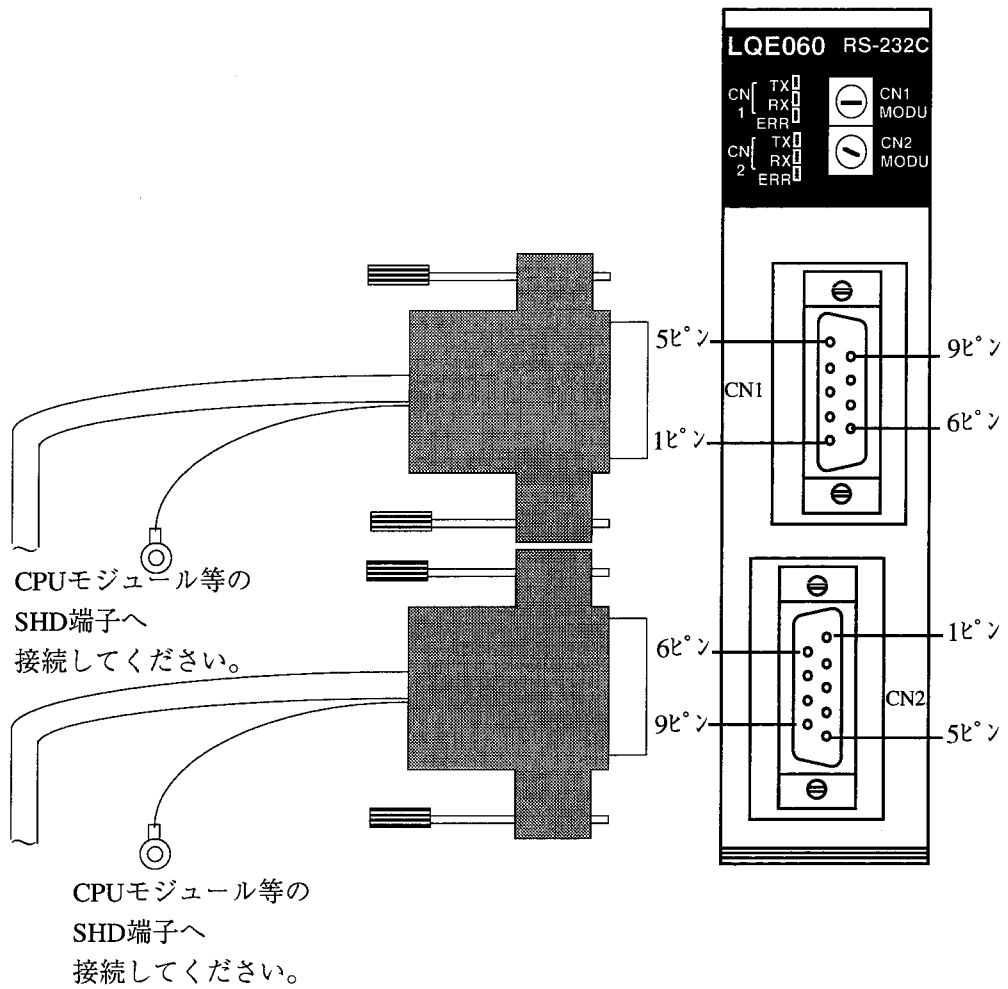
- 通電中には設定しないでください。設定は、電源を落とした状態で行ってください。
- 無手順の演算ファンクションシステムとタスクシステムを混在で設定しないでください。CPUユニット単位でどちらか一方のみで使用してください。
- チャンネルNo.は重複しないでください。

〈設定例〉 チャネルNo.が重複しないように設定します。

LQE060 RS-232C		LQE060 RS-232C			MODU NO設定	通信方式	チャネル No.
CN1	TX RX ERR	CN1	TX RX ERR	1枚目-CN1	8	H-7338 <sup>o</sup> プロトコル	#0
CN2	TX RX ERR	CN2	TX RX ERR	1枚目-CN2	1	無手順-演算ファンクション	#1
				2枚目-CN1	2	無手順-演算ファンクション	#2
				2枚目-CN2	3	無手順-演算ファンクション	#3

3.2 配線

No.	名称	機能					
⑤	RS-232C コネクタ	外部機器と接続するためのコネクタ DSUB-9ピン(オス)					
		ピン #	信号名	入出力 区分	信号の意味	解放時の状態	
		1	CD (data Carrier Detect)	データチャネル 受信キャリア	入力	1: キャリアなし 0: キャリアあり	キャリアなし
		2	RD (Receive Data)	受信データ	入力	1: マーク 0: スペース	マーク
		3	SD (Send Data)	送信データ	出力	1: マーク 0: スペース	—
		4	ER (Equipment Ready)	データ端末 レディ	出力	1: ノットレディ 0: レディ	—
		5	SG (Signal Ground)	信号接地	—	—	—
		6	DR (data set Ready)	データセット レディ	入力	1: ノットレディ 0: レディ	ノットレディ
		7	RS (Request to Send)	送信要求	出力	1: 要求なし 0: 要求あり	—
		8	CS (Clear to Send)	送信可	入力	1: 送信不可 0: 送信可能	送信不可
9	(空き)						



### 3 各部の名称と機能、配線

● RS-232Cの信号内容

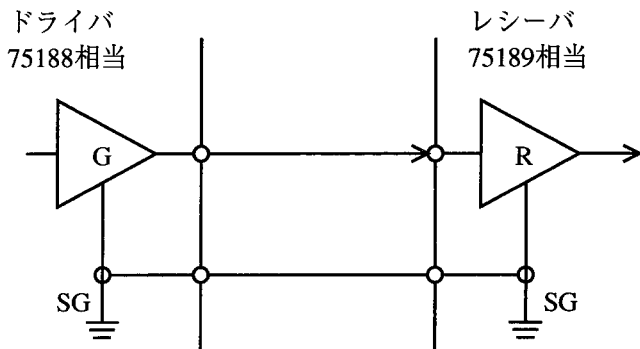
- [SD] RS-232Cモジュールから端末へのデータ線です。RS, CR, DR, ERの4つの制御線がオンのときに有効です。
- [RD] 端末からRS-232Cモジュールへのデータ線です。CDがオンのとき有効です。データを伝送していない間(CDがオフの間)は、マーク状態にしておきます。
- [RS] RS-232Cモジュールから端末へ出力するデータがあることを表す制御線です。RSがオンの間は、端末はRS-232Cモジュールからのデータの入力状態を続けます。RSをいったんオフしたならば、CSがオフになるまで再びオンにはできません。
- [CS] 端末が通信回路へのデータ送信が可能であることを表す制御線です。CSがオンのときは、端末はRS-232Cモジュールからのデータの入力が可能です。
- [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、RS-232Cモジュール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] 端末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。RS-232Cモジュールは、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] RS-232Cモジュールが端末に対して、データの入出力ができることを表す制御線です。RS-232Cモジュールは、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。(CPUのSHD端子を使用します。コネクタ内にはありません。)

● 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す。一般に、±12Vが使用されます。

● RS-232Cの基本回路



G : ドライバ  
R : レシーバ

● 外部機器との接続方法

(1) 接続例

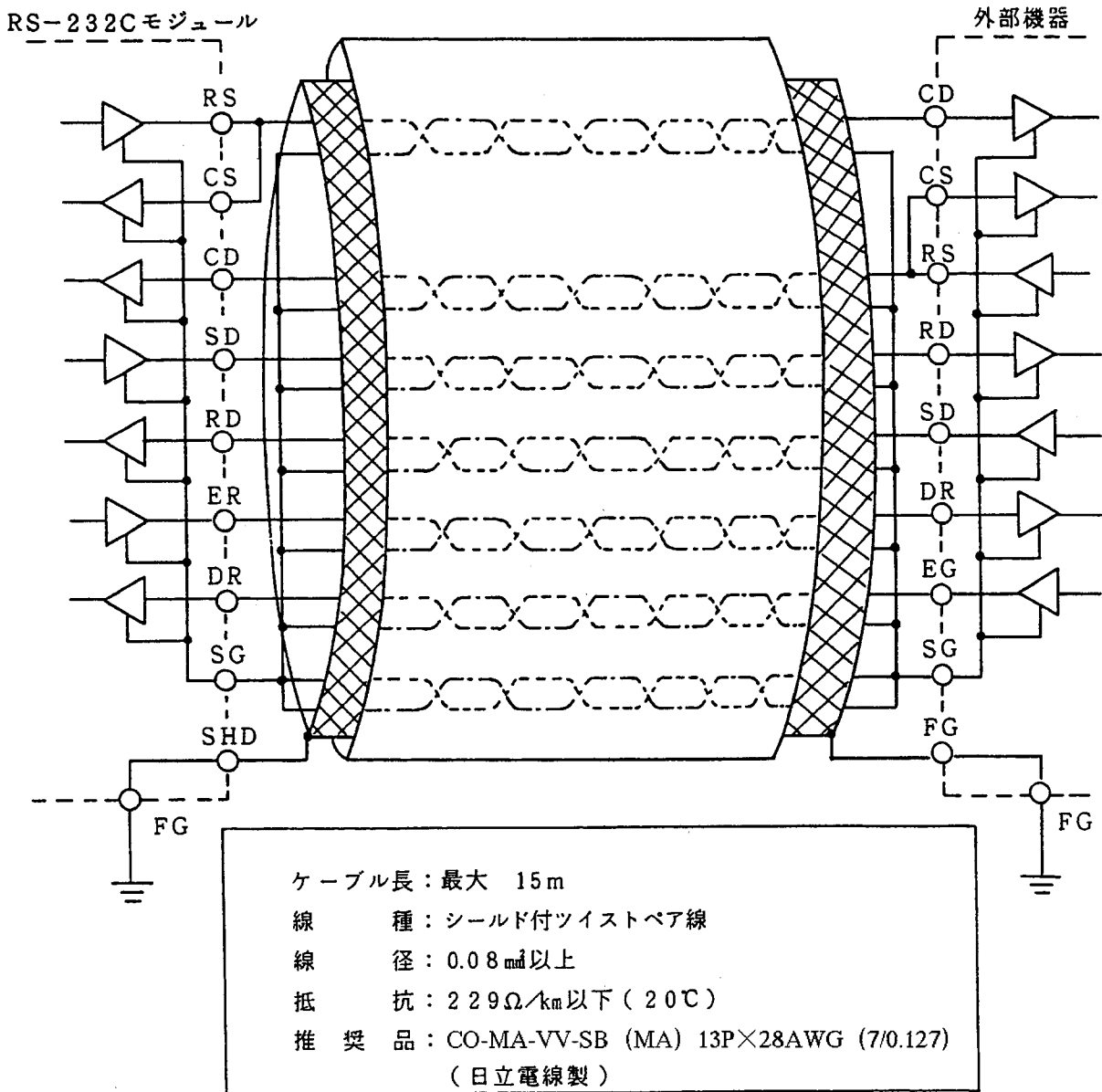
接続	名称	接続方法 (論理的接続)			外部機器例
		RS-232Cモジュール	ケーブル	外部機器	
データ イ レ ク ト 接 続	Full Modem Support (標準タイプ) (CDによる受信管理とDRによる送信管理を行いながらデータの送受信を行う。)	RS CS CD SD RD ER DR SG		RS CS CD SD RD ER DR SG	パーソナルコンピュータ
	CDによる受信管理 (外部機器からの送信要求(RS)で、外部機器リンクを受信可能状態とする。)	RS CS CD SD RD ER DR SG		RS CS CD SD RD ER DR SG	
	DRによる送信管理 (外部機器からのデータ端末レディ(ER)でRS-232Cモジュールから送信データを送信する。)	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)
	データのみ (外部機器, RS-232Cモジュールの状態をチェックせず、データを送受信する。)	RS CS CD SD RD ER DR SG		RS CS CD SD RD ER DR SG	
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	プリンタ

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

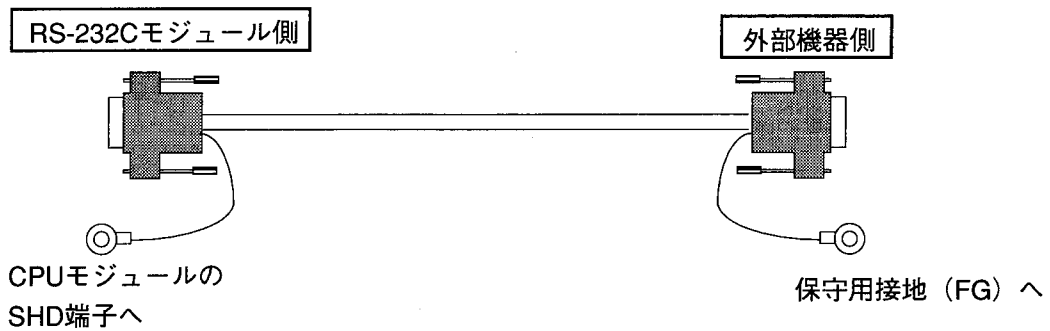
#### (2) RS-232Cインタフェースの接続方法

接続例の標準タイプを例にとり、接続方法を下記に示します。



⚠ 注 意

- RS-232Cモジュールと外部機器の信号用接地（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接地端子は、接地電位が同一になる場合、RS-232Cモジュール、外部機器の両側とも接地してください。耐ノイズ性が向上します。接地電位が異なる場合は、RS-232Cモジュール側のみ接地してください。
- RS-232Cモジュールのシールド用接地（SHD）端子は、CPUモジュール端子台のシールド用接地（SHD）端子を使用してください。



<このページは余白です>



## 4 オペレーション

## 4.1 システム立上げ

- |     |                 |  |
|-----|-----------------|--|
| [1] | モジュール実装         | [1] CPUの電源を切り、RS-232Cモジュールを実装します。  |
| [2] | ロータリスイッチ設定      | [2] RS-232CモジュールのMODU NOスイッチ（通信方式、チャンネルNo.）をコネクタごとに使用方法に合わせて設定します。   |
| [3] | 外部機器リンクシステム立上げ  | [3] CPUとWindows®パソコンを接続し、外部機器リンクシステムを立上げます。「ソフトウェアマニュアル オプション外部機器リンク For Windows® (SAJ-3-143)」を参照してください。   |
| [4] | LGBの通信制御テーブルの編集 | [4] 接続される外部機器の仕様に合わせてLGB通信制御テーブルの編集を行います。<br>「ソフトウェアマニュアル オプション外部機器リンク For Windows® (SAJ-3-143)」を参照してください。 |
| [5] | 演算ファンクションの登録    | [5] アプリケーションプログラムを演算ファンクションで行うときに登録します。「ソフトウェアマニュアル オプション外部機器リンク For Windows® (SAJ-3-143)」を参照してください。       |
| [6] | 受信起動タスクの登録      | [6] アプリケーションプログラムをCモードで作る場合に登録します。「ソフトウェアマニュアル オプション外部機器リンク For Windows® (SAJ-3-143)」を参照してください。            |

### ⚠ 注意

- RS-232Cモジュールは、アプリケーションの利用形態に合わせて以下のいずれかを選択し使用してください。また、無手順一タスクシステムを選択しCPUがLQP000の場合は、オプションの拡張メモリモジュールが必要になります。
  - ・無手順一演算ファンクションシステム
  - ・無手順一タスクシステム
  - ・H-7338プロトコル
- 無手順では、接続される外部機器の仕様に合わせLGBの編集が必要になります。H-7338プロトコルのLGBの設定は固定です。H-7338で使用する場合、LGBをツールなどで変更すると正常に動作しませんので注意してください。
- 編集したLGBはリセット後（CPUモジュールのリセットスイッチによる）有効になります。LGBを編集後、リセット前またはリセット中に停復電があった場合は、前回設定したLGBへ戻ります。その場合は再度LGBを編集しリセットを実施してください。

## 4.2 LGB通信制御テーブルの編集

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Windows®パソコン「外部機器リンクFor Windows®」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、伝送手順がくいちがって正常送受信が行え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行ってください。

### ■ 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
- ・ テキストサイズ
- ・ 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
- ・ 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
- ・ システム選択

4.3 LGBに設定する内容

LGBの内容を変更する場合の各項目内容を次に説明します。

■ 伝送フレーム (DATA FRAME)

回線上の1バイトデータのフレーム構成を決定します。

P S E α 選択項目		伝 送 フ レ ー ム 内 容	初期値
選択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	○

- 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

- 伝送速度 (BAUD RATE)  
回線の伝送速度 (bps) を設定します。(150~9,600bps)

P S E α 選択項目		伝 送 レ ー ト 内 容	初期値
選択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。	

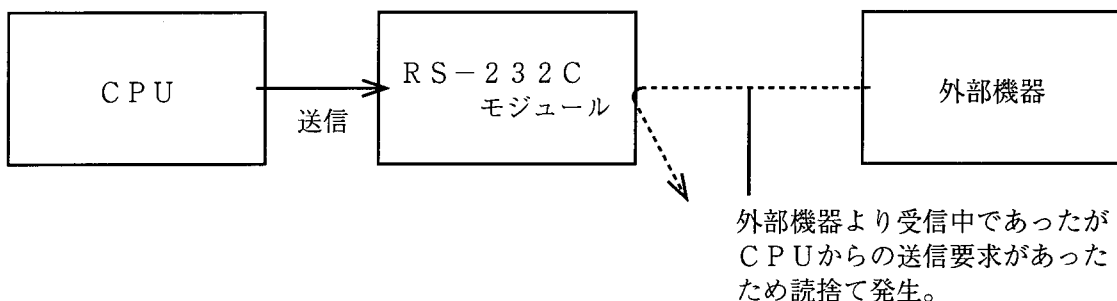
BPS:ビット/秒

※ 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。

- 優先制御 (PRIORITY LEVEL)  
自局 (RS-232C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、RS-232C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。

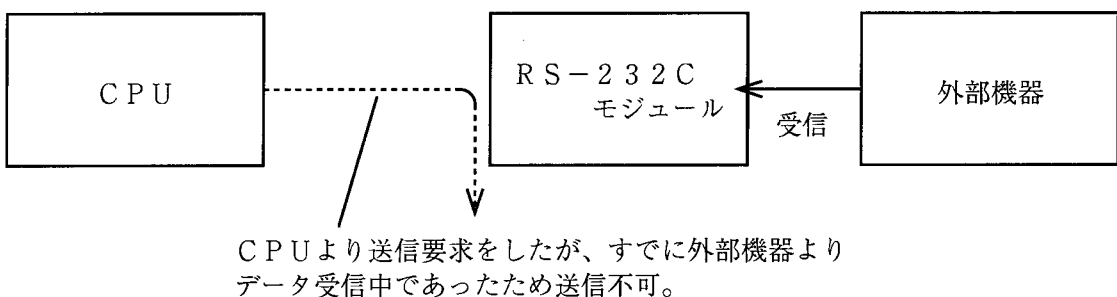
P S E α 選択項目		優 先 制 御 内 容	初期値
選択No.	表 示 内 容		
0	SELF	自局優先	○
1	OTHERS	他局優先	

● 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捨てが発生したことは認識できないので、CPU側より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は送信不可であることをSレジスタにて確認できます。

## 4 オペレーション

### ■ データ変換モード (DATA CHANGE MODE)

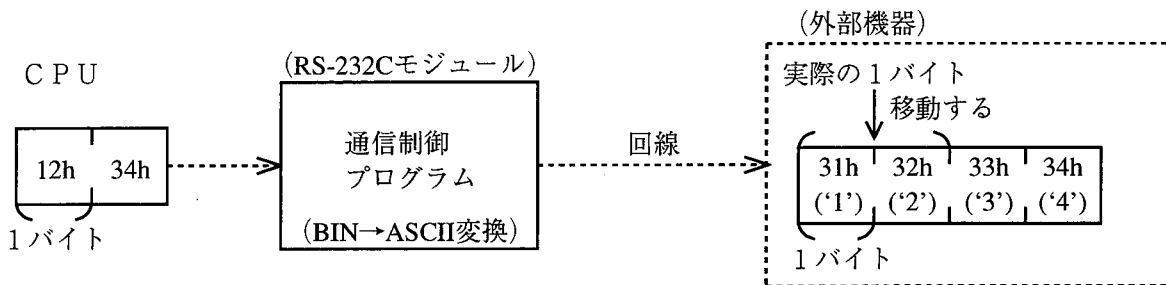
回覧上のTEXTデータをASCII (アスキー) データとして扱うか、BINARY (バイナリ) データとして扱うかの選択を行います。

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	○

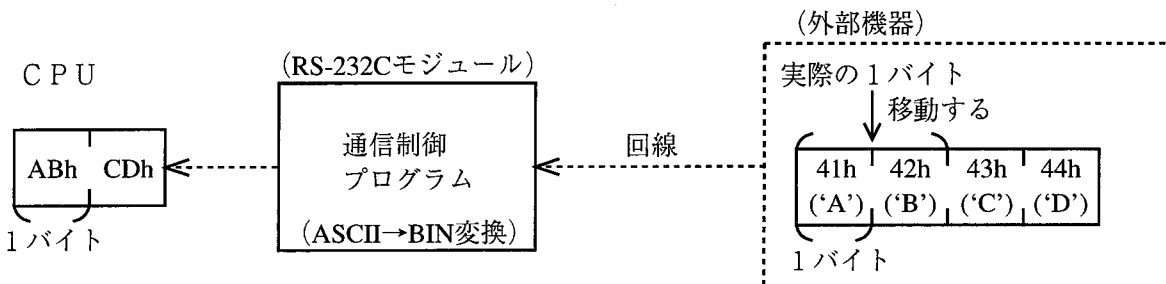
### ● ASCII 指定の場合

ASCII 指定をすると外部機器のプログラムでASCII↔BINARY変換するため回覧上のデータ量は2倍となります。

また回覧上に '0' ~ '9', 'A' ~ 'F' 以外のTEXTデータを送信するとエラーになります。



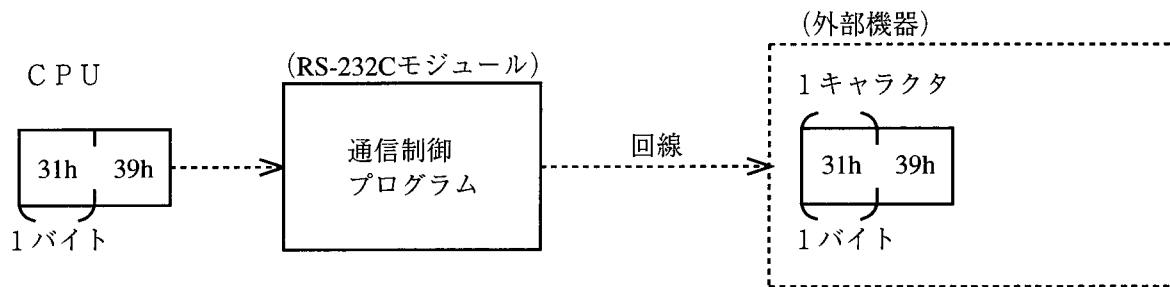
上図例の場合、CPUからデータ12h, 34hを送信すると通信制御プログラムは、BINARY→ASCII変換を行い、外部機器へデータ31h ('1'), 32h ('2'), 33h ('3'), 34h ('4')を送信しますので、それにあわせて外部機器側のプログラムを作成する必要があります。



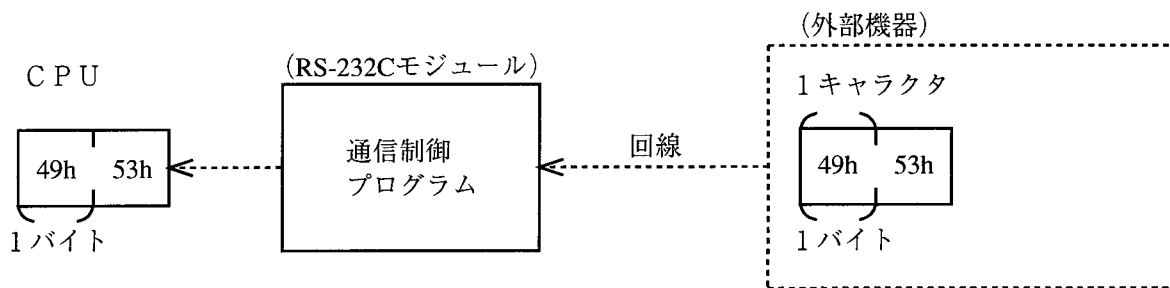
上図例の場合、外部機器からデータ41h ('A'), 42h ('B'), 43h ('C'), 44h ('D')を受信すると通信制御プログラムはASCII→BINARY変換を行いCPUへデータABh, CDhを渡しますので、それにあわせて外部機器側のプログラムを作成する必要があります。

● 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 オペレーション

### ■ テキストサイズ (TEXT SIZE)

TEXT (テキスト) 語数は0～512の範囲で指定できます。

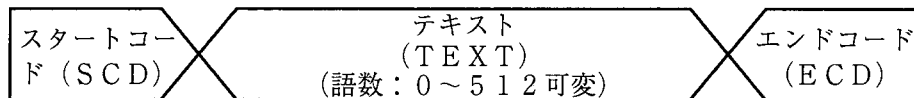
P S E 選 択 項 目		テキストサイズ内容	初期値
メニューNo.	表 示 内 容		
0	NO TEXT	テキストなし	
1～512	001～512 [BYTE]	1～512 [バイト]	256

TEXTデータのスタートは、SCDありの場合はSCD受信後の次のデータからとし、TEXTデータの終了はECDの受信または指定したTEXT語数分のデータ受信をもって、終了とします。

したがって、TEXT語数、SCD、ECD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。

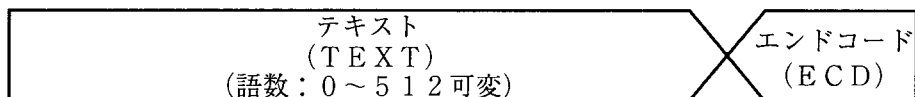
ASCII指定の場合、通信制御プログラムは、回線上データをASCII→BINARY変換します。

#### ● SCD, ECDありの場合



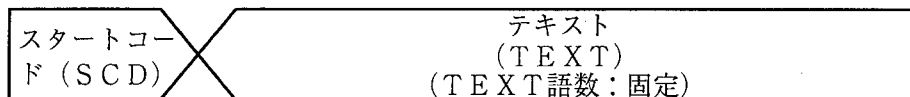
上記の場合、TEXT語数を512としてもTEXT内にECDをユーザが設定すればそこで、通信制御プログラムは送受信を終了します。またECDが存在しないとTEXT長を512とし、その前後にSCD、ECDをつけたものとして処理します。

#### ● ECDありの場合



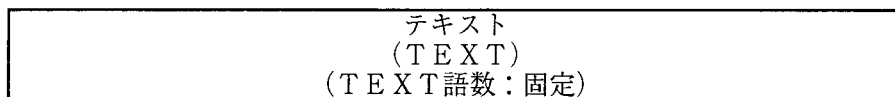
上記の場合もTEXT内にECDをユーザ設定することによりTEXT長を可変として扱うことができます。

#### ● SCDありの場合



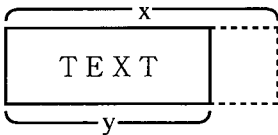
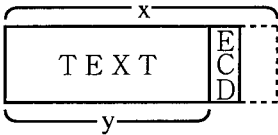
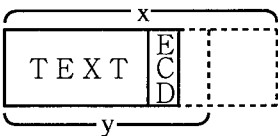
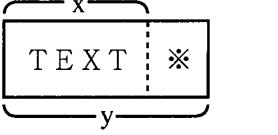
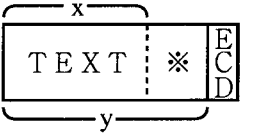
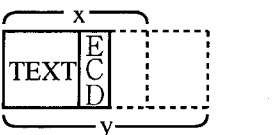
上記の場合はTEXT長はTEXT語数指定分固定となります。

#### ● TEXTのみの場合



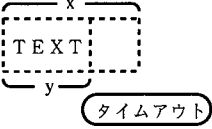
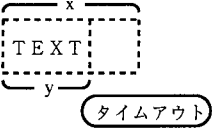
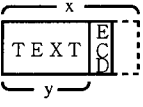
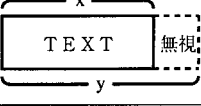
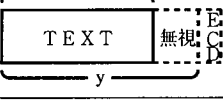
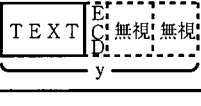


LGB指定テキスト語数と送信ハンドラの送信語数の関係を以下に示します。  
 送信ハンドラ送信語数を x バイト, LGB指定テキスト語数を y バイトとします。

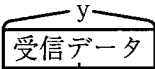
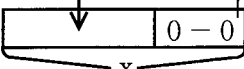
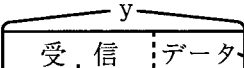

大小関係	LGB指定 エンドコード	テキスト中に エンドコード	回線に送信されるデータ	
$x \geq y$	なし	—		LGB指定テキスト語数が送信。
$x \geq y$	あり	なし		LGB指定テキスト語数+ECD (エンドコード) が送信。
$x \geq y$	あり	あり		テキストの先頭からテキスト中の ECD (エンドコード) までが送信。
$x < y$	なし	—	 ※以前の送信バッファのデータ	テキスト部+以前の送信バッファ のデータが送信。
$x < y$	あり	なし	 ※以前の送信バッファのデータ	テキスト部+以前の送信バッファ のデータ+ECD (エンドコー ド) が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

#### 4 オペレーション

LGB指定テキスト語数と受信ハンドラの取込み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	
$x > y$	なし	—		LGB指定語数まで、受信データを待ち、TEXT (テキスト) 監視時間のタイムアウトとなる。
$x > y$	あり	なし		
$x > y$	あり	あり		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。
$x \leq y$	なし	—		LGB指定語数のみ受信し、以降は無視される。
$x \leq y$	あり	なし		
$x \leq y$	あり	あり		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み語数を  $x$  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>	取り残しデータは演算ファンクションハンドラの場合は無視します。タスクハンドラの場合は5.3.2項を参照してください。

### ■ スタートコード (SCD:START CODE)

TEXT (テキスト) の開始を示すデータで有/無指定、およびありの場合のコード数 (1~4 キャラクタ)、コードデータを設定できます。

P S E 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	<input type="radio"/>	
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。

- SCDありの場合、通信制御プログラムは、SCD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、TEXTデータの直前へ指定SCDコードを付加して送信します。

SCDは、ASCII指定の場合にも、ASCII変換されません。

### ■ エンドコード (ECD:END CODE)

TEXT (テキスト) の終了を示すデータで、有/無指定、およびありの場合のコード数 (1~4 キャラクタ)、コードデータを設定できます。

P S E 選 択 項 目			エンドコード内容	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	<input type="radio"/>	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		

CD1~CD4: 00h~FFhのエンドコードを示す16進。

- ECDありの場合、通信制御プログラムは、ECD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、TEXTデータの直後へ指定ECDコード付加して送信します。

ECDは、ASCII指定の場合にも、ASCII変換されません。

### ■ ブロックチェックキャラクタ (BCC:BCC MODE)

送受信フレームの合理性チェック用データでECDありの場合はECDの次に、ECDなしの場合はTEXTの次に存在します。

P S E 選 択 項 目		B C C 内 容	初期値
メニューNo.	表 示 内 容		
0	NO BCC	BCCなし	<input type="radio"/>
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	

## 4 オペレーション

BCCチェックに関しては、有/無の指定とBCC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の指定ができます。

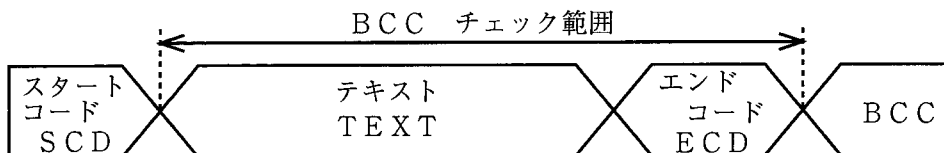
$$\text{水平偶数パリティ} \cdots (\text{BCC})_E = (00h) \text{EOR} \left( \sum_{i=0}^n \text{EOR } D_i \right)$$

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

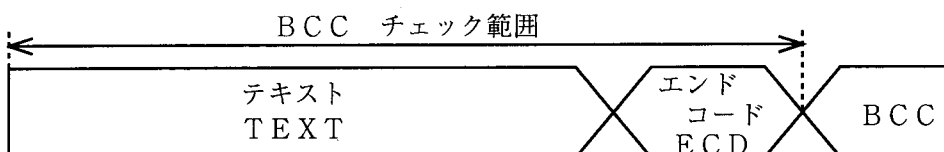
$$(\text{BCC})_{08} = (FFh) \text{EOR} \left( \sum_{i=0}^n \text{EOR } D_i \right) (\text{データビット8ビット})$$

以下に、BCCチェック範囲を示します。

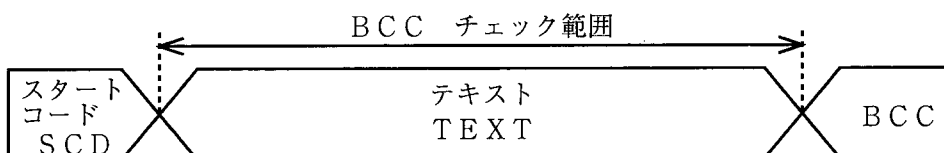
- SCD, ECDありの場合



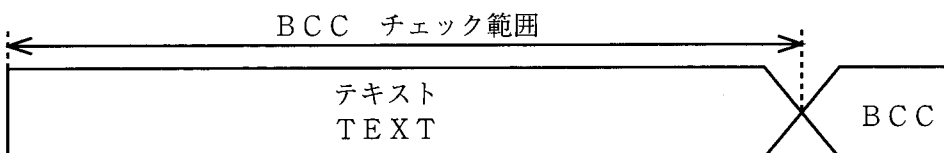
- ECDありの場合



- SCDありの場合



- TEXTのみの場合



ECDがない場合は、TEXT語数指定分の固定長としてチェックします。

BCCチェックありの場合は、上記規則にのっと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。

ASCII指定の場合、ASCII変換される前のTEXT (BINARYデータ) とECDがチェック範囲です。

■ 送信遅延時間 (SEND DELAY TIME)

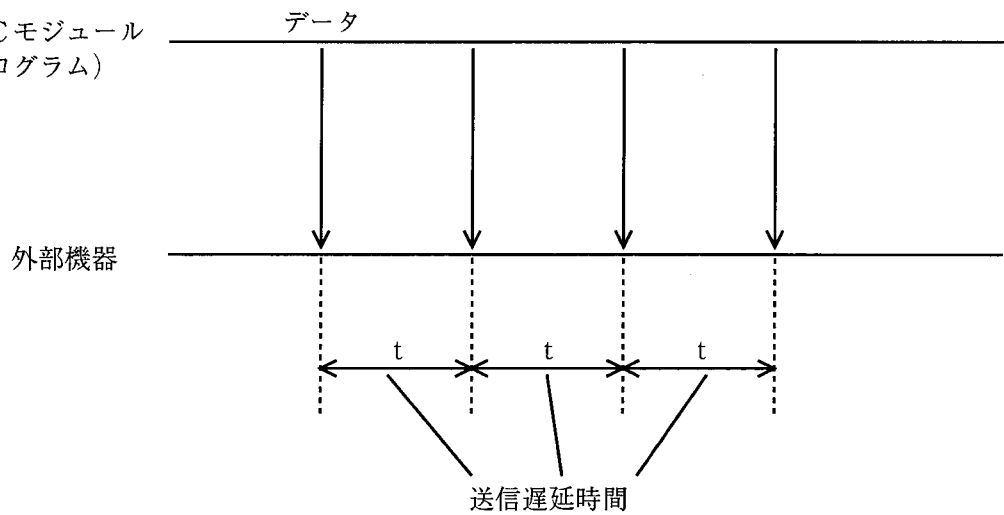
通信制御プログラムより外部機器へデータ送信時、データ送信の時間間隔を規定します。

P S E 設 定		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	0 0 0 0 1 ~ 3 2 7 6 7 [ms]	1 ~ 3 2 7 6 7 [ms]	

制限事項

伝送レート	送信遅延時間設定範囲
150 [BPS]	1 2 8 ~ 3 2,7 6 7 [ms]
300 [BPS]	6 4 ~ 3 2,7 6 7 [ms]
600 [BPS]	3 2 ~ 3 2,7 6 7 [ms]
1200 [BPS]	1 6 ~ 3 2,7 6 7 [ms]
2400 [BPS]	8 ~ 3 2,7 6 7 [ms]
4800 [BPS]	4 ~ 3 2,7 6 7 [ms]
9600 [BPS]	2 ~ 3 2,7 6 7 [ms]

RS-232Cモジュール  
(通信制御プログラム)



## 4 オペレーション

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

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。

P S E 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	

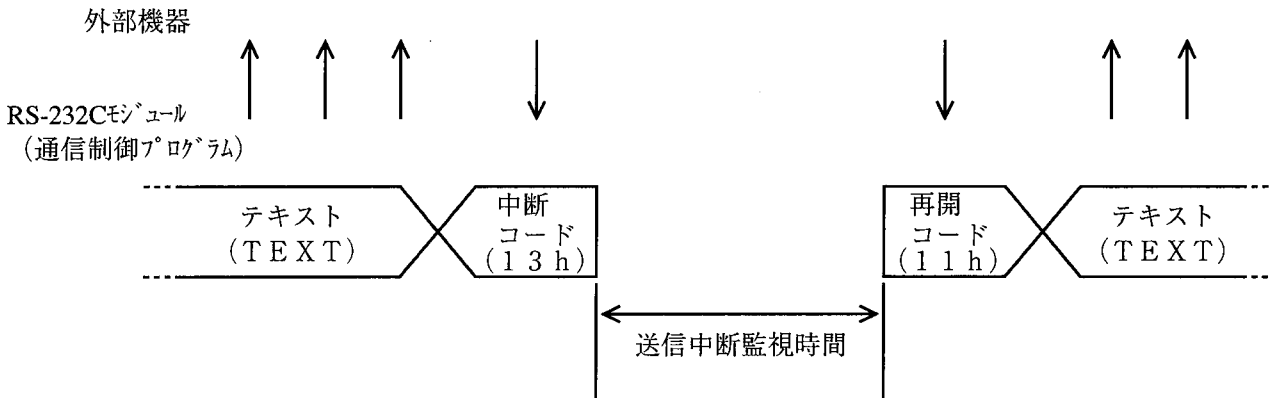
BR:中断コード。 CN:再開コード。

CD1~4:00h~FFhの送信中断,再開コードを示す16進値。

中断/再開処理の有無,中断/再開処理あり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)の指定を行います。

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可能となり、その他のコードは無視します。また中断/再開コードとも、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])

とした場合の例です。

後で説明する受信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、受信監視時間を越えた場合はエラーとします。

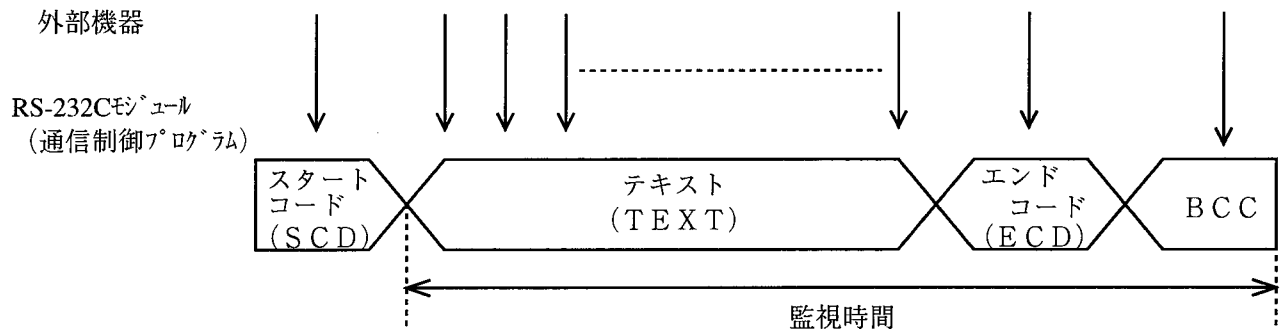
### ■ 送信中断監視時間 (SEND BREAK TIMEOUT)

P S E 設 定		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 [100ms]	0.1~3276.7 [s]	32767

### ■ 受信監視時間 (RECEIVE TIMEOUT)

通信制御プログラムのTEXT (テキスト) 受信時のTEXT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。

P S E 設定		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 [100ms]	0.1~3276.7 [s]	32767



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

P S E 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	無効	○

### ■ 送信要求 (RS:REQUEST TO SEND)

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、RS-232Cモジュールは、送信データを送信できます。

P S E 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	送信要求なし出力	

#### ・送信要求ありの場合

RS-232C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ともに、送信可能状態となります。

#### ・送信要求なしの場合

RS-232C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ともに、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(「5.2.1 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、RS-232C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(一般には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 オペレーション

### ■ データ端末レディ (ER:EQUIPMENT READY)

外部機器に対して、RS-232C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、RS-232Cモジュールと外部機器間のプロトコルによりますが、一般にはRS-232Cモジュールが受信可能な状態をレディと定義します。

P S E 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	○

#### ・レディの場合

RS-232Cモジュールは、外部機器に対しデータ端末レディ (ER) 端子よりレディ状態を出力し続けます。

#### ・ノットレディの場合

RS-232Cモジュールは、外部機器に対しデータ端末レディ (ER) 端子よりノットレディ状態を出力し続けます。

- 外部機器よりデータを受信する場合は、レディを設定してください。
- 外部機器より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、RS-232C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 (一般には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。

### ■ 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。

P S E 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	

#### ・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しますとエラーとします。

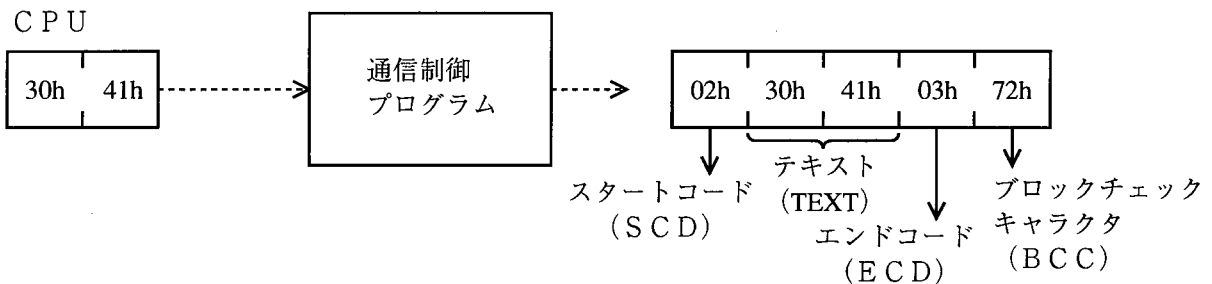
#### ・チェックなしの場合

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  
RS-232CモジュールのDR端子と外部機器の受信レディ出力端子 (一般には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。



通信制御プログラムは送るべきデータに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）

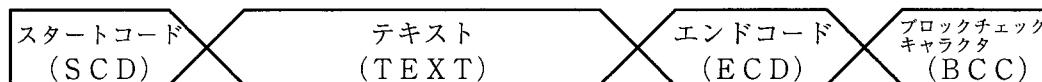
エンドコード……………1文字指定03h（ETX：テキスト終了）

ブロックチェックキャラクタ……………水平偶数パリティ

とした場合の例です。

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。

下図に一般的なブロック構成を示します。



#### ■ システム選択

チャンネルNo.設定スイッチ（MODU NO）により、演算ファンクションシステムかタスクシステムか選択されます。

演算ファンクションの登録または、受信起動タスクレベルの設定を行ってください。

##### ● 演算ファンクション

外部機器とデータの送受信を行うため演算ファンクションを登録します。

名称	機能
SD 0	チャンネルNo.0の送信演算ファンクション
SD 1	チャンネルNo.1の送信演算ファンクション
SD 2	チャンネルNo.2の送信演算ファンクション
SD 3	チャンネルNo.3の送信演算ファンクション
RV 0	チャンネルNo.0の受信演算ファンクション
RV 1	チャンネルNo.1の受信演算ファンクション
RV 2	チャンネルNo.2の受信演算ファンクション
RV 3	チャンネルNo.3の受信演算ファンクション

## 4 オペレーション

---

- タスクシステム

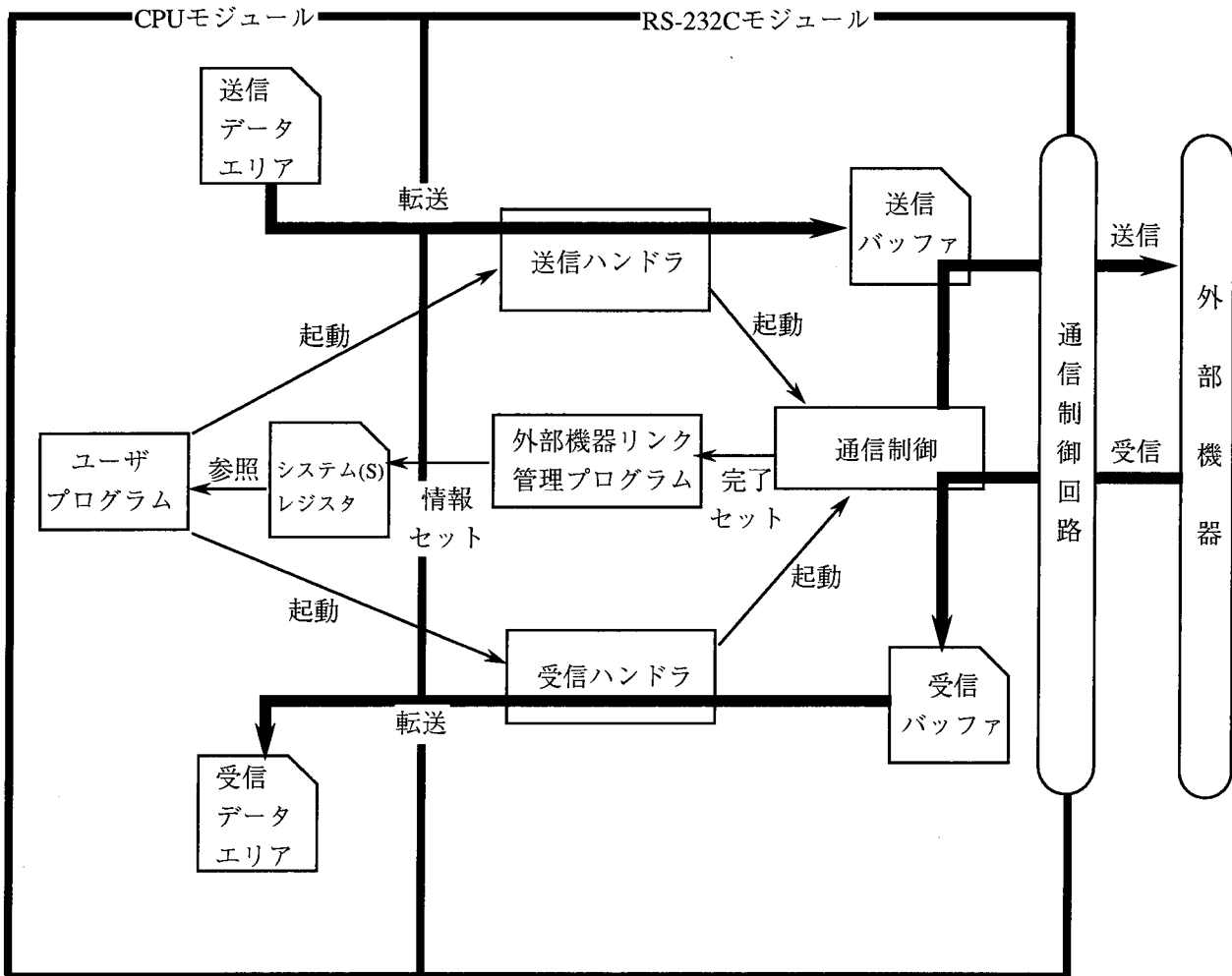
RS-232Cモジュールにデータ受信があった場合、CPU OSより起動をかけるユーザタスクを指定します。

	設定範囲	備考
起動タスク番号	2～100	整数値入力
起動要因	0～16	整数値入力

# 5 プログラミング

5.1 ソフトウェア構成

外部機器との通信は次のように行われています。



- 通信制御プログラム  
送信ハンドラからの送信指令を受けると回線に送信データを送出します。また、回線からの受信データをすべて受け取ったとき、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。
- 外部機器管理プログラム  
通信制御プログラムの送信受信に関する情報を、CPUのシステム（S）レジスタに反映しユーザプログラムに知らせます（「5.2.1 送信情報」，「5.2.2 受信情報」参照）。
- 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送信データエリアより指定語数分送信データをRS-232C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。ユーザプログラムにて、送信したいタイミングで送信ハンドラに起動をかけてください。
- 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み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受信データエリアへ指定語数分通信データを転送。
  - ・データを取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。ユーザプログラムにて受信完了を認識後に受信ハンドラを起動し、データを取込んでください（受信完了認識は「5.2.2 受信情報」参照）。
- システムレジスタ  
送信可能か受信完了かエラーはないかなどの情報が設定されています。ユーザプログラムはこの情報を参照して作成します。

5.2 システムレジスタ

5.2.1 送信情報

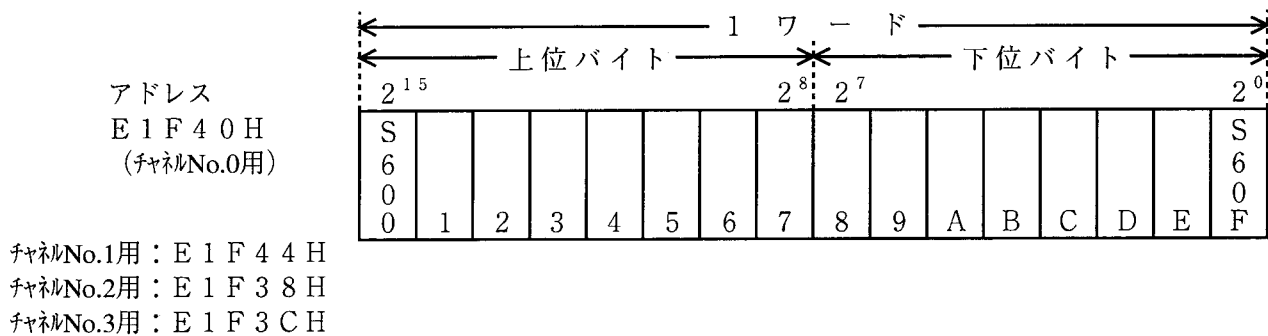
CPUから外部機器への送信は送信ハンドラによって行いますが、送信動作の情報はCPUのシステムレジスタ（Sレジスタ）に反映されます。

ユーザプログラムは、このSレジスタを参照し、送信可能/不可能、送信エラーの識別を行います。

チャネルNo.0用	チャネルNo.1用	チャネルNo.2用	チャネルNo.3用	意味	ビット内容	
					0	1
S600	S620	S5C0	S5E0	送信可フラグ	送信可能	現在送信中
1	1	1	1	ハンドラエラーフラグ	エラーなし	エラーあり
2	2	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	3	3	受信打ち切り送信	エラーなし	エラーあり
4	4	4	4	未使用	未使用	
5	5	5	5			
6	6	6	6			
7	7	7	7			
8	8	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します。 「7.3.2 送信エラーコード表」の下位バイト参照	
9	9	9	9			
A	A	A	A			
B	B	B	B			
C	C	C	C			
D	D	D	D			
E	E	E	E			
S60F	S62F	S5CF	S5EF			

システム（S）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よりワードデータとして読込むことができます。



5.2.2 受信情報

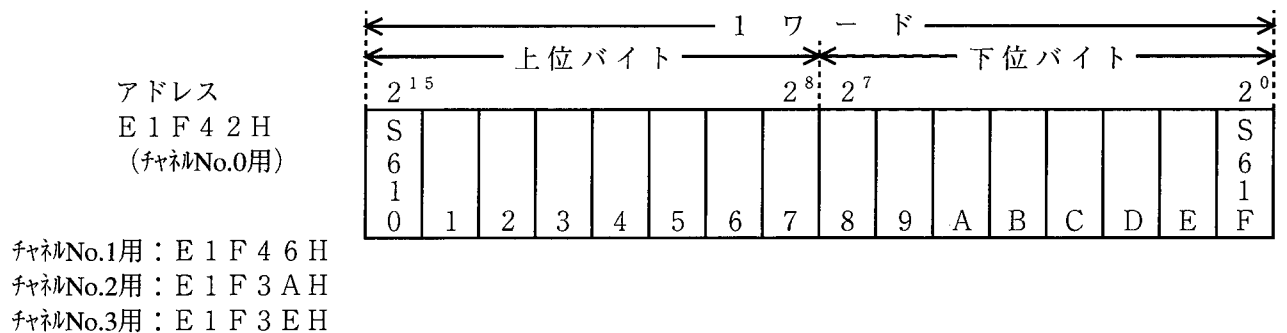
外部機器からの受信は通信制御プログラムが行い、正常受信および異常受信とその情報は、CPUのシステム（S）レジスタに反映されます。

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行います。

チャネルNo.0用	チャネルNo.1用	チャネルNo.2用	チャネルNo.3用	意味	ビット内容	
					0	1
S610	S630	S5D0	S5F0	受信完了フラグ	受信データなし	受信データあり
1	1	1	1	ハンドラエラーフラグ	エラーなし	エラーあり
2	2	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	3	3	システムエラー	エラーなし	エラーあり
4	4	4	4	ハンドラエラーコード	ハンドラエラー内容を示す。受信エラーコード表上位バイトの下1桁参照。	
5	5	5	5			
6	6	6	6			
7	7	7	7			
8	8	8	8	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。「7.3.3 受信エラーコード表」の下位バイト参照。	
9	9	9	9			
A	A	A	A			
B	B	B	B			
C	C	C	C			
D	D	D	D			
E	E	E	E			
S61F	S63F	S5DF	S5FF			

S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よりワードデータとして読込むことができます。



## 5 プログラミング

### 5.3 送受信ハンドラ

送信，受信のハンドラは、起動されるユーザプログラムがラダーかCモードかにより演算ファンクションかサブルーチンになります。

#### 禁 止

チャンネルNo.0は演算ファンクション、チャンネルNo.1はタスクシステムといった使用、また送信はタスクシステム、受信は演算ファンクションといった使用はできません。  
必ずCPUユニット単位で演算ファンクションかタスクシステムに統一してください。

#### 5.3.1 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8種類があります。

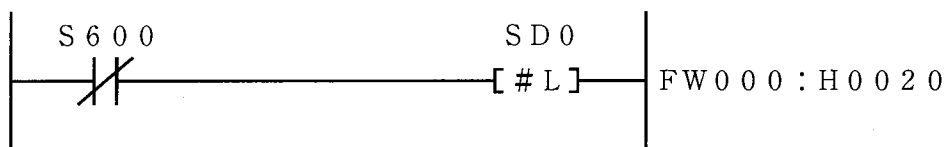
名 称	機 能
SD0	チャンネルNo.0用送信演算ファンクション
SD1	チャンネルNo.1用送信演算ファンクション
SD2	チャンネルNo.2用送信演算ファンクション
SD3	チャンネルNo.3用送信演算ファンクション
RV0	チャンネルNo.0用受信演算ファンクション
RV1	チャンネルNo.1用受信演算ファンクション
RV2	チャンネルNo.2用受信演算ファンクション
RV3	チャンネルNo.3用受信演算ファンクション

(注) 演算ファンクションを使用しラダープログラムを組む場合は、必ずLGB登録(4.2 LGB通信制御テーブルの編集)より、演算ファンクションを登録後、ラダープログラム編集画面から ビルド 受信 を選択し、一度CPU内のデータをWindows® Toolに受信してください。受信をすることにより登録された演算ファンクションを使用してラダープログラムを組むことができます。



## SD0, SD1, SD2, SD3 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より指定語数分を外部機器へ送信します。
- パラメータ** 転送アドレス：XW000, FW000などニーモニック  
転送語数 : 1~512
- リターンコード** リターンコードは、システムレジスタに格納されます。  
(「5.2.1 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No.0に接続されている外部機器へ送信します。





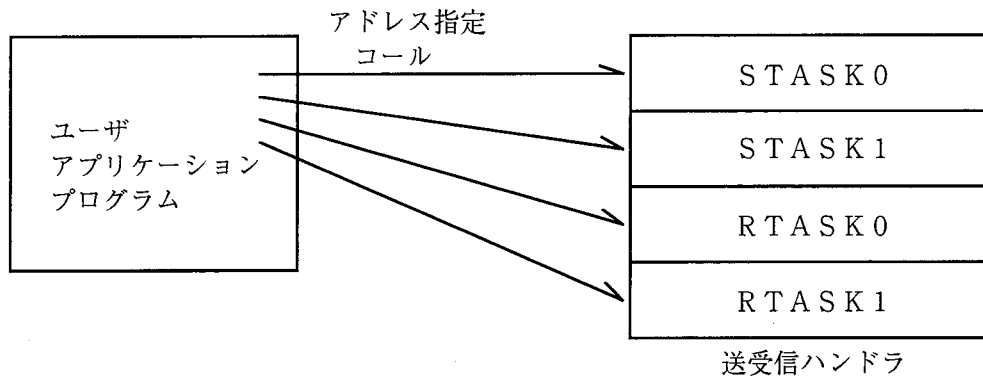
## 5.3.2 サブルーチン

Cモードプログラム用の送受信ハンドラは、以下に示す8種類があります。

名 称	アドレス	機 能
STASK0	107000H	チャンネルNo.0用送信サブルーチン
STASK1	107006H	チャンネルNo.1用送信サブルーチン
STASK2	107018H	チャンネルNo.2用送信サブルーチン
STASK3	10701EH	チャンネルNo.3用送信サブルーチン
RTASK0	10700CH	チャンネルNo.0用受信サブルーチン
RTASK1	107012H	チャンネルNo.1用受信サブルーチン
RTASK2	107024H	チャンネルNo.2用受信サブルーチン
RTASK3	10702AH	チャンネルNo.3用受信サブルーチン

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FA-BASICにて作成します。

Cモードプログラム用送受信ハンドラは、アドレス指定でコールするため、ユーザアプリケーションプログラムは、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。



## 5 プログラミング

### STASK0, STASK1, STASK2, STASK3 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より指定語数分を外部機器へ送信します。

#### リンク手順

(例# 0用)

C 言語	アセンブラ言語
<code>long (*stask0)( );</code>	<code>move.l #sbyte, -(A7)</code>
<code>long rtn, sadr, sbyte ;</code>	<code>move.l #sadr, -(A7)</code>
	<code>lea \$107000, A0</code>
<code>stask0 = 0x107000l ;</code>	<code>jsr (A0)</code>
	<code>addq.l #8, A7</code>
<code>rtn = (*stask0)(sadr, sbyte) ;</code>	

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

`s a d r` : 送信データ格納エリアアドレス

`s b y t e` : 送信バイト語数

`r t n` : リターンコード

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

(エラー情報をシステムレジスタに格納します。「5.2.1 送信情報」参照。)

#### Note

`l o n g ( 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。

RS-232C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、システムレジスタに格納されます。「5.2.1 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 )
{
    stask0 = 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
bne    LB1 -----> 送信不可ならばLB1へ
move.l  #100, -(A7)
move.l  #150000, -(A7) -----> 転送語数256バイト
lea     $107000, A0 -----> 送信データエリア150000h番地
jsr     (A0)
addq.l  #8, A7
tst.l   D0
bne    ERRB -----> 送信エラーならばERRBへ
:

```

## 5 プログラミング

RTASK0, RTASK1, RTASK2, RTASK3 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。  
 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
(例 # 0用)	<pre>long (*rtask0)(); long  rtn, radr, rbyte;  rtask0 = 0x10700C1;  rtn = (*rtask0)(radr, rbyte);</pre>	<pre>move.l    #rbyte, -(A7) move.l    #radr, -(A7) lea      \$10700C, A0 jsr      (A0) addq.l   #8, A7</pre>

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**

r a d r : 受信データ格納エリアアドレス  
 r b y t e : 受信バイト語数  
 r t n : リターンコード

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、まだ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れた。  
 エンドコードが現れた。  
 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 x x 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（「7.3.3 受信エラーコード表」を参照）。また、エラー情報をシステムレジスタに格納します（「5.2.2 受信情報」参照）。

= FFFFFFFFh : 受信ハンドラ起動異常  
 エラー情報をシステムレジスタに格納します（「5.2.2 受信情報」参照）。

RS-232C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、システムレジスタに格納されます。「5.2.2 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) 転送します。

```

:
register long (*rtask0)( );
register long rtn ;
:
if ( (*(short*)0xE1F421 & 0x8000) != 0 )
{
rtask0 = 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しか読込めません。) 例えば、外部機器より下記データを受信した場合の例を示します。

"1 2 3 4 5 6 7 8 9 0"

10 キャラクタ受信

受信ハンドラ 起動時の取込語数	リターンコード	取込データ
3	0 (正常)	"1 2 3"
4	0 (正常)	"4 5 6 7"
4	001A0000h (正常, ブロック終了)	"8 9 0" 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は、エンドコードをセットし、まだ余りのエリアがある場合は余りのエリアへ 0 を書込みます。

## 5 プログラミング

ここで外部機器より下記データを受信した場合の例を示します。

“12345” 5キャラクタ受信, エンドコード/030001

受信ハンドラ 起動時取込語数	取込データ	リターンコード
7	“12345”/0300	/001A0000
8	“12345”/030001	/001A0000
9	“12345”/03000100	/001A0000

### アセンブラ言語の例

- 受信システムレジスタ S610 (アドレス E1F42 h 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 150000 h 番地のデータエリアに 256 バイト (16 進: 100 h) 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)
move.l  #150000, -(A7) -----> 256バイト
lea     $10700C, A0 -----> 150000h番地より
jsr     (A0)
addq.l  #8, A7
tst.l   D0
bne     ERRB -----> 受信エラーならばERRBへ
:

```

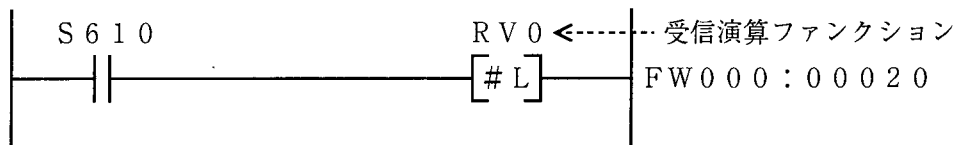


## 5.4 受信データの取込み方

通信制御プログラムが受信完了すると、その情報は該当のシステム（S）レジスタへ反映されます。これを参照して次のようにプログラムを作ります。

- ラダープログラムで作成する場合

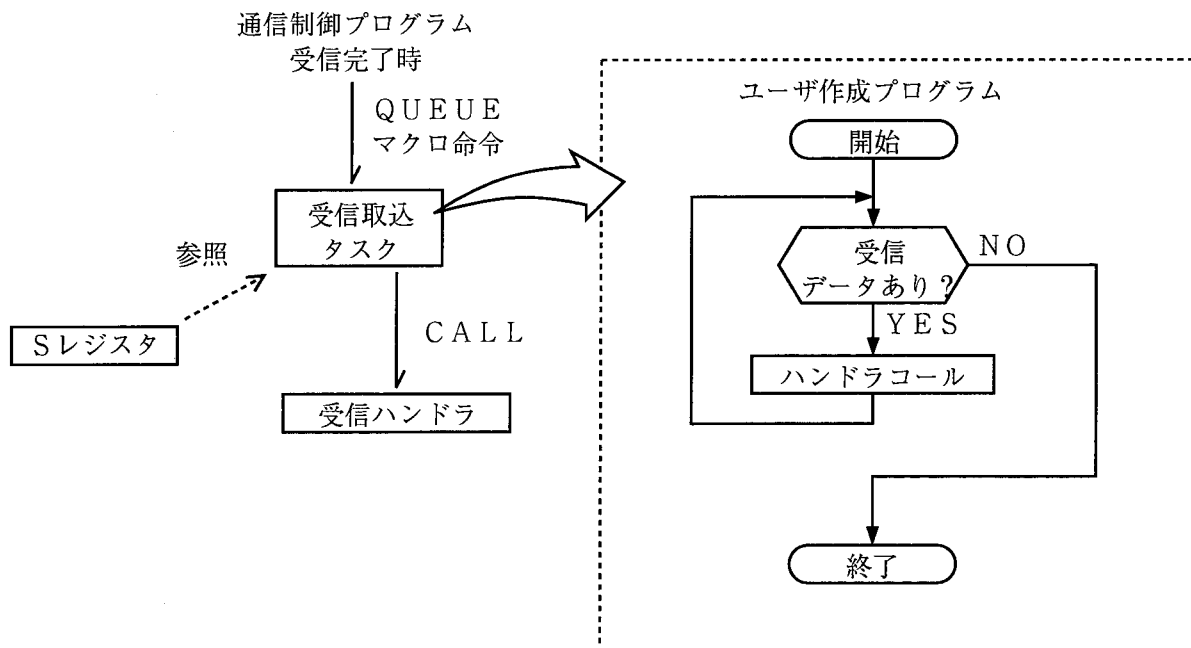
Sレジスタを条件として受信ハンドラ（演算ファンクション）を起動すると、受信データ取込み遅れはシーケンスサイクル内（標準30ms）でおさまります。



- Cモードプログラムの場合

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より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込むことが可能となります。



### 5.5 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No.)
- サブルーチンを使う場合  
送信ハンドラ名称 : TASK? (?はチャンネルNo.) (※3)

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一)	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。 (「7.3.1 CPUモジュール」インディケータ表示参照)																						
・Cモードサブルーチンのとき アドレス 62FFE H	0000H または 0001H	最新ハードウェア状態取込み要求	<ul style="list-style-type: none"> <li>・チャンネルNo.0のとき DWFFFの上位バイト (62FFE H)</li> <li>・チャンネルNo.1のとき DWFFFの下位バイト (62FFF H)</li> <li>・チャンネルNo.2のとき DWFFEの上位バイト (62FFC H)</li> <li>・チャンネルNo.3のとき DWFFEの下位バイト (62FFD H)</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>⑤</td><td>⑥</td> </tr> </table> </div> <div style="margin-top: 10px;"> <table style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">0 : ON (※1) =</td> <td rowspan="2" style="font-size: 3em; vertical-align: middle;">}</td> <td style="text-align: center;">① RS出力状態</td> </tr> <tr> <td style="text-align: center;">1 : OFF (※2) =</td> <td style="text-align: center;">② CS入力状態</td> </tr> <tr> <td style="text-align: center;">0 : OFF (※2) =</td> <td rowspan="2" style="font-size: 3em; vertical-align: middle;">}</td> <td style="text-align: center;">③ CD入力状態</td> </tr> <tr> <td style="text-align: center;">1 : ON (※1) =</td> <td style="text-align: center;">④ ER出力状態</td> </tr> <tr> <td></td> <td></td> <td style="text-align: center;">⑤ 送信ゲート状態</td> </tr> <tr> <td></td> <td></td> <td style="text-align: center;">⑥ DR入力状態</td> </tr> </table> </div>	①	②	③	④	⑤	⑥	0 : ON (※1) =	}	① RS出力状態	1 : OFF (※2) =	② CS入力状態	0 : OFF (※2) =	}	③ CD入力状態	1 : ON (※1) =	④ ER出力状態			⑤ 送信ゲート状態			⑥ DR入力状態
①	②	③	④	⑤	⑥																				
0 : ON (※1) =	}	① RS出力状態																							
1 : OFF (※2) =		② CS入力状態																							
0 : OFF (※2) =	}	③ CD入力状態																							
1 : ON (※1) =		④ ER出力状態																							
		⑤ 送信ゲート状態																							
		⑥ DR入力状態																							

- (※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10ms (最大) 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。
- (※4) ソフトウェアリセット以外の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”で使用してください。“PROT. ON”の状態で行うとプロテクトエラーとなります。

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 ・Cモードサブ ルーチンのと きアドレス 62FFE H	0100H	送信ゲートCLOSE要求	・チャンネルNo.0のとき DWFFFの上位バイト (62FFE H) ・チャンネルNo.1のとき DWFFFの下位バイト (62FFF H) ・チャンネルNo.2のとき DWFFEの上位バイト (62FFC H) ・チャンネルNo.3のとき DWFFEの下位バイト (62FFD H)
	0101H	送信ゲートOPEN要求	
	0200H	DR OFF要求 (※1)	
	0201H	DR ON要求 (※2)	/00……OFF報告 /01……ON報告
	0300H	RS OFF要求 (※1)	
	0301H	RS ON要求 (※2)	・チャンネルNo.0のとき DWFFFの上位バイト (62FFE H) ・チャンネルNo.1のとき DWFFFの下位バイト (62FFF H) ・チャンネルNo.2のとき DWFFEの上位バイト (62FFC H) ・チャンネルNo.3のとき DWFFEの下位バイト (62FFD H) /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。

<このページは余白です>

## 6 プログラム例

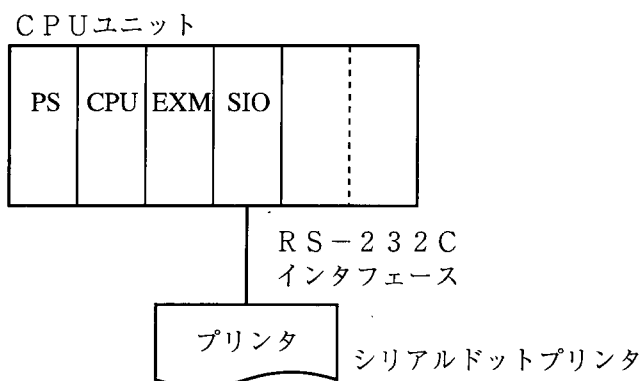
## 6 プログラム例

### 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 システムの構成

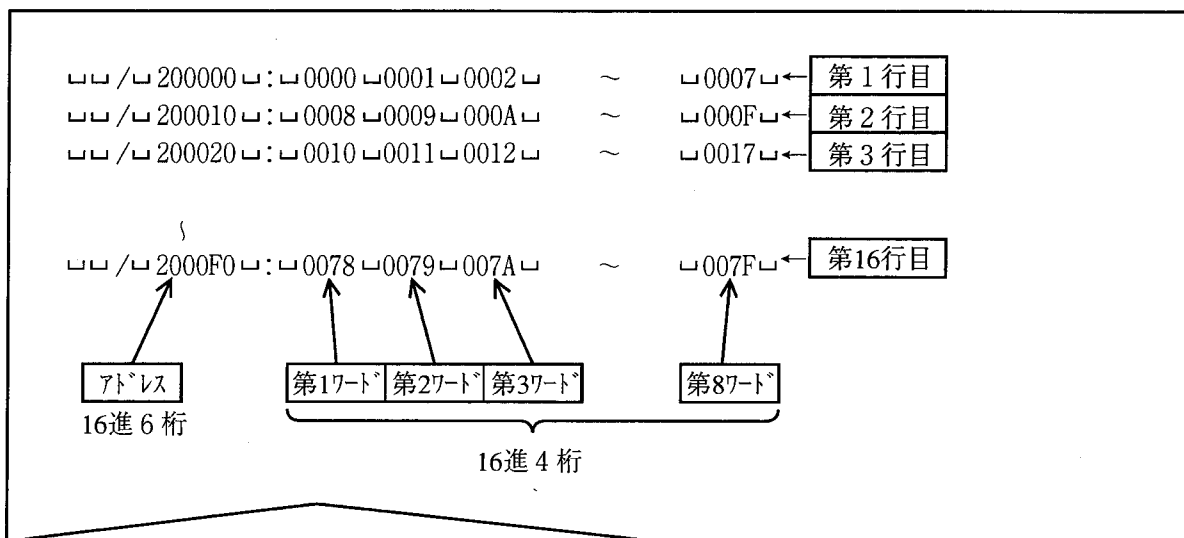


CPUユニットの機器構成

略称	名称	形式	数量
PS	電源モジュール	LQV000	1
CPU	CPUモジュール	LQP000	1
EXM	拡張メモリモジュール	LQM000	1
SIO	RS-232Cモジュール	LQE060	1
	CPUマウントベース	HSC-1080	1

#### 6.1.3 プリンタの印字フォーマット

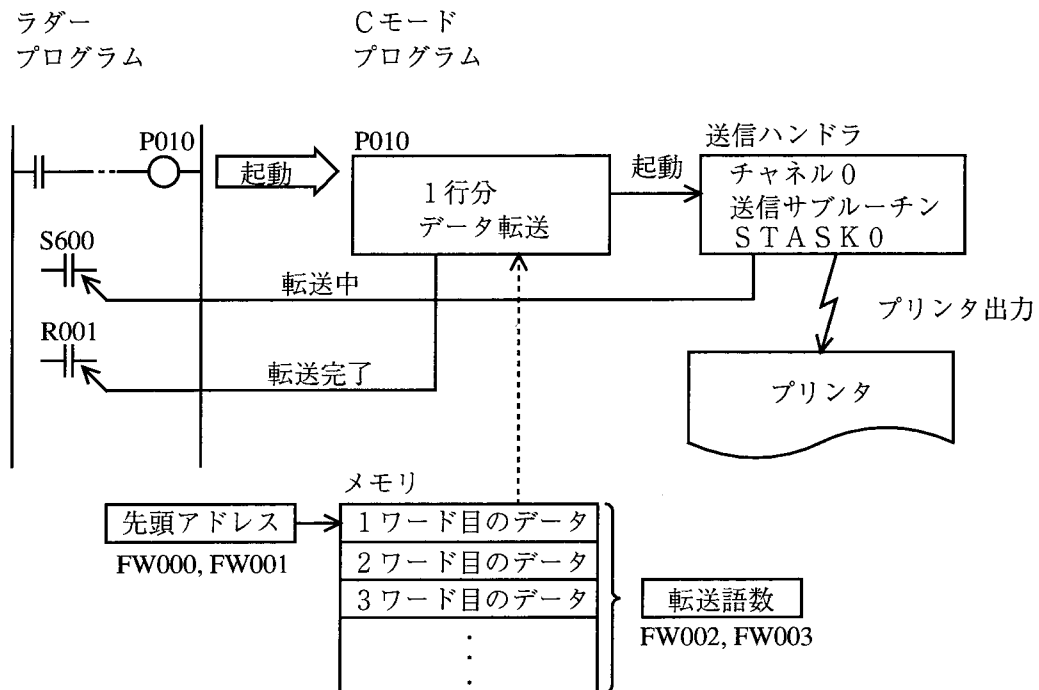
プリンタの印字フォーマットは、次のように行います。



□	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

## 6.1.4 プログラムの構成

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ています。



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すると起動します。

また、メモリのプリンタ出力のための制御情報テーブルは、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 プログラム例

### 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		} 32ビット長
FW001	0E2002h			
FW002	0E2004h	転送ワード語数		} 32ビット長
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から出力完了するとONとなり、初めて1行分の印字処理の起動を行ったとき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	<div style="border: 1px solid black; width: 100%; height: 15px; position: relative;"> <span style="position: absolute; right: 0; top: 0; bottom: 0; width: 10px; height: 10px; background: repeating-linear-gradient(45deg, transparent, transparent 2px, black 2px, black 4px);"></span> </div>	

このメモリエリアは、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

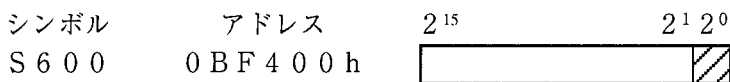
OFF時 = 0000h



(3) 転送中フラグ

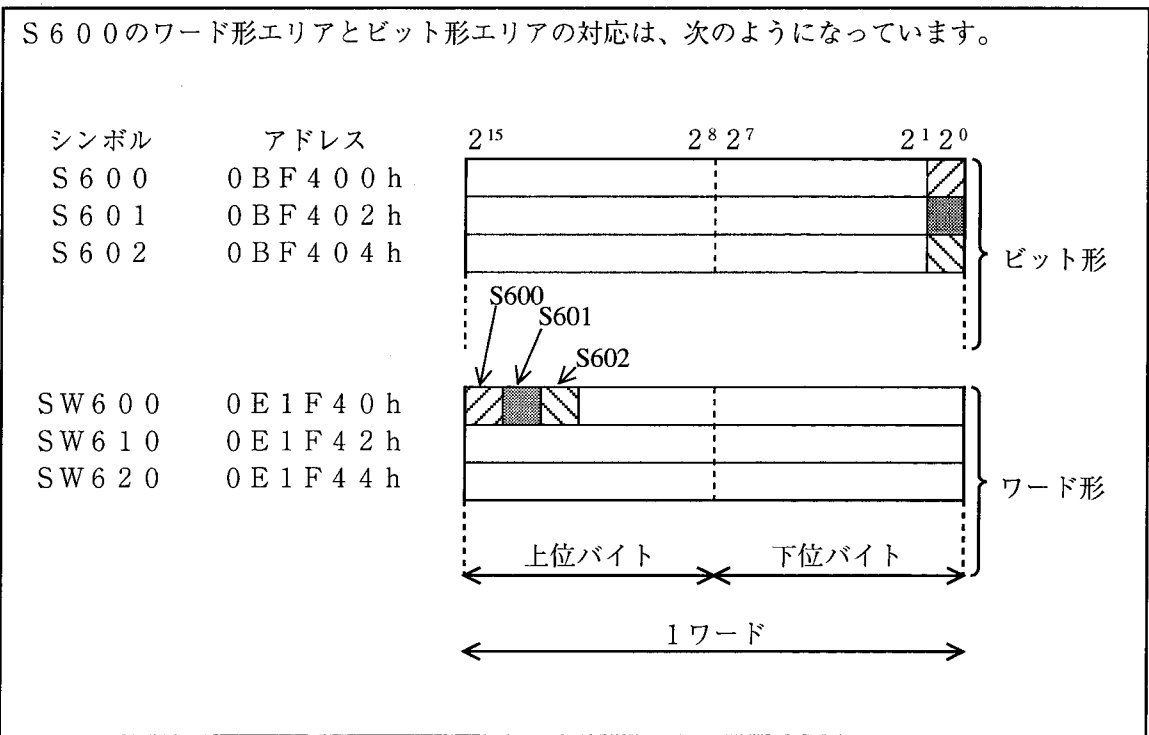
プリンタへの1行分データ出力処理が起動する送信ハンドラ (STASK0) のデータ転送中フラグのシステムレジスタS600を使用します。

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。



このメモリエリアは、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

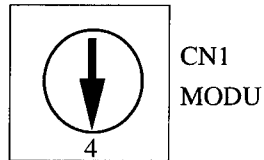


## 6 プログラム例

### 6.1.6 RS-232Cモジュール

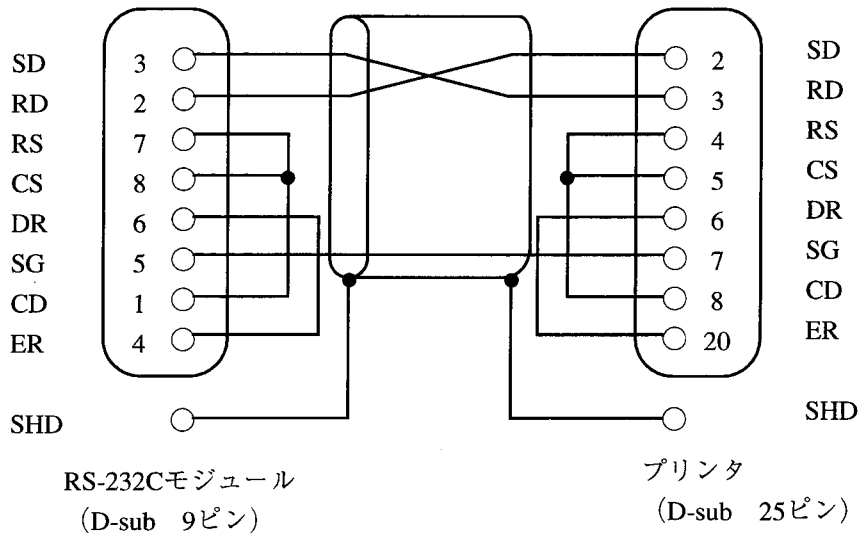
#### (1) MODU NOスイッチの設定

CN1を無手順一タスクシステムで使用するため、MODU NOスイッチは4に設定します。



#### (2) RS-232Cの信号線

RS-232Cの線号接続は、データ信号線のみで、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

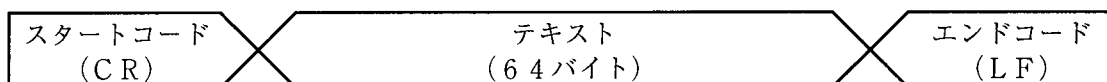
今回使用したプリンタのシリアルインタフェースの仕様は、次のものを使用しています。

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、自動的に改行を行います。

## 〈伝送ブロックの構成〉

ここでは、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、次のようにしています。



- 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〉

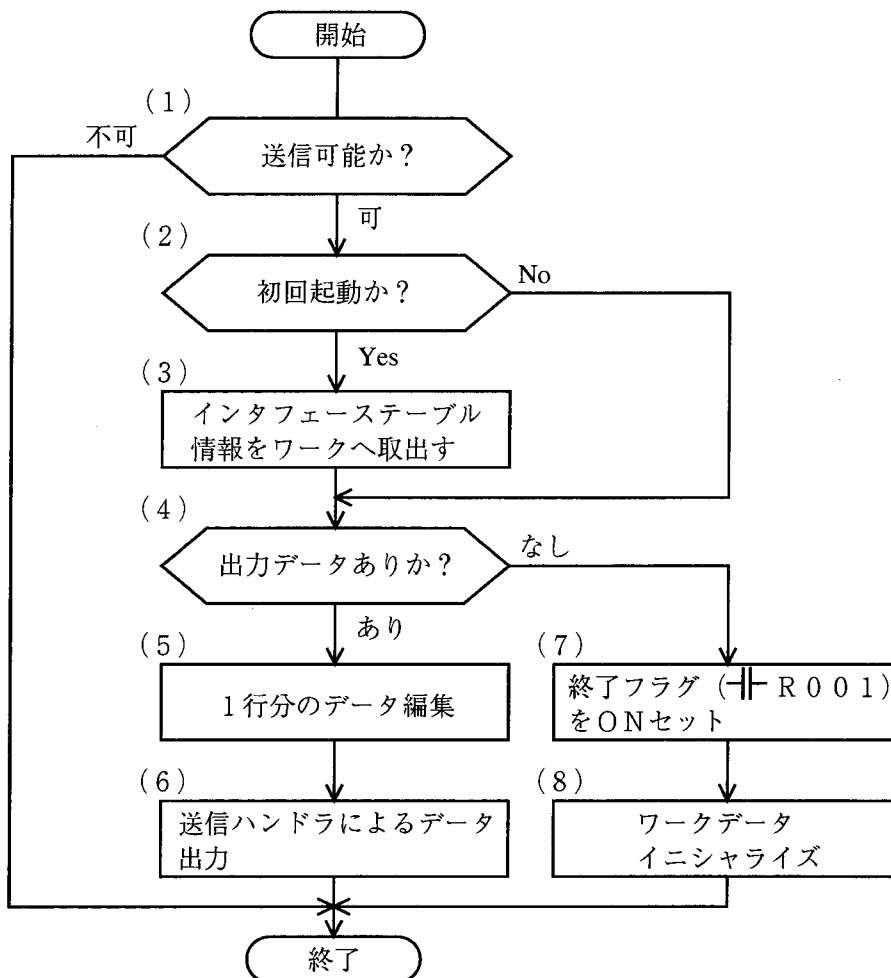
```

-----
                CHANNEL No.0 LGB DATA
-----
0 : DATA FRAME ----- ST+8DT+EP+1SP                ---※
1 : BAUD RATE ----- 4800 [BPS]                      4,800 bps      ---※
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 MODE ----- 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]       3276.7s
B : RECIEVE TIMEOUT ----- 32767 [100 mSEC]          3276.7s
C : RS-422 GATE CONTROL ----- OPEN                  操作しない
D : REQUEST TO SEND (RS) ----- LOW
E : EQUIPMENT READY (ER) ----- HIGH
F : DATA SET READY (DR) ----- NO CHECK
G : SYSTEM MODE (TASK) ----- TU : ** FT : **
-----

```

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

6.1.8 C言語プログラムのフローチャート



(1) システムレジスタ (S600) の状態より、送信可能かを調べます。

(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の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

(5)~(6) : データありの場合、データ1行分を編集した後に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。

## 6.1.9 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 ] ;      プリンタ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 > 0l ) && ( ct < 50 ) )

```

## 6 プログラム例

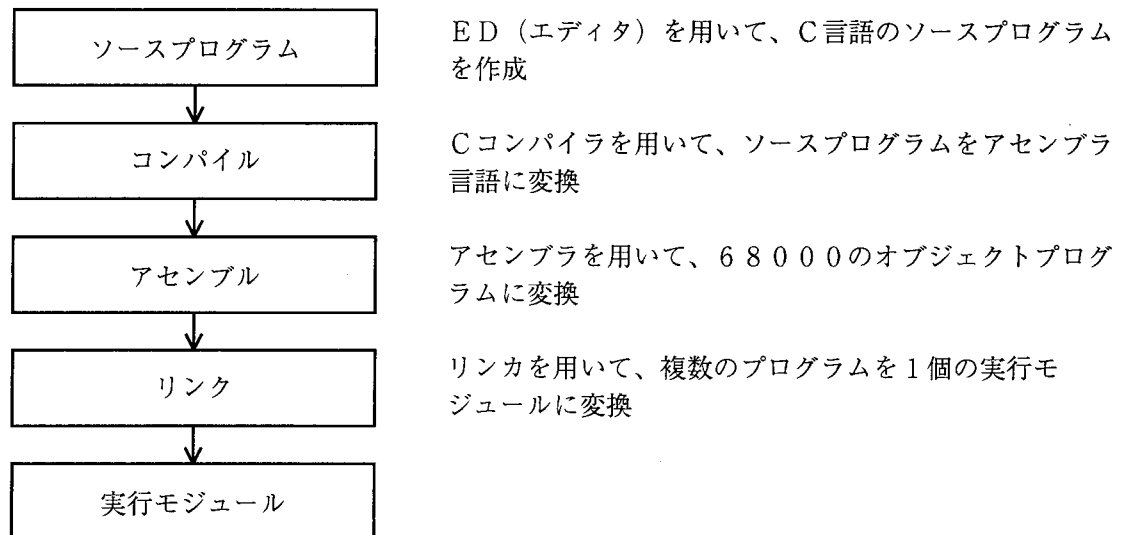
### (2) バイナリ→ASCII (アスキー) 変換サブルーチン

```
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 ;                      キャラクタのセットポインタ
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言語プログラムの作成と登録

## (1) C言語プログラムの作成

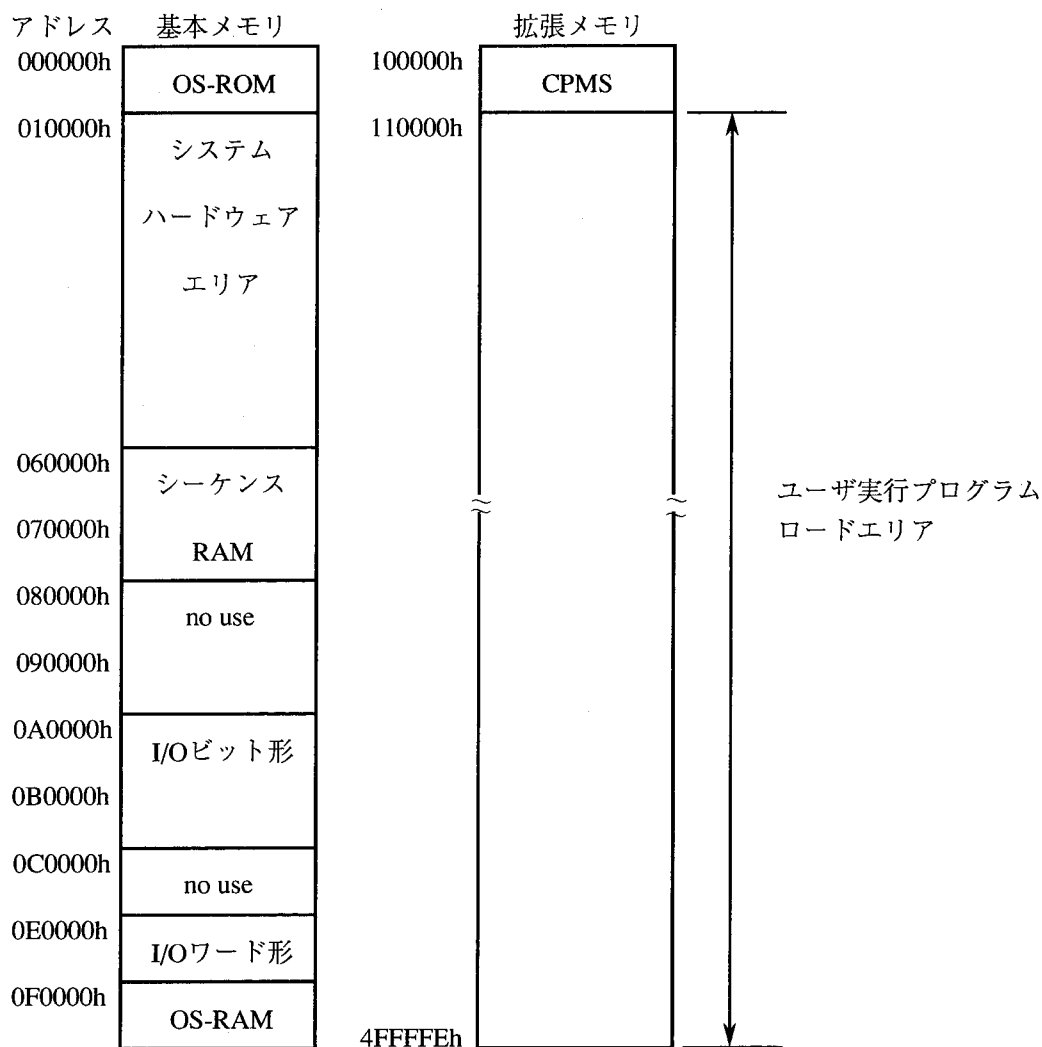
CP/M68kシステムを用いPSE $\alpha$ で次の手順に従いプログラムを作成します。



## 6 プログラム例

### (2) C言語プログラムのロード（書込み）

実行可能なC言語プログラムは、拡張メモリ（アドレス110000h以降）にロードします。



LQP000には、拡張メモリは内蔵されていません。

LQP010、LQP011は1Mバイトの拡張メモリを内蔵しています。

LQP120は2Mバイトの拡張メモリを内蔵しています。

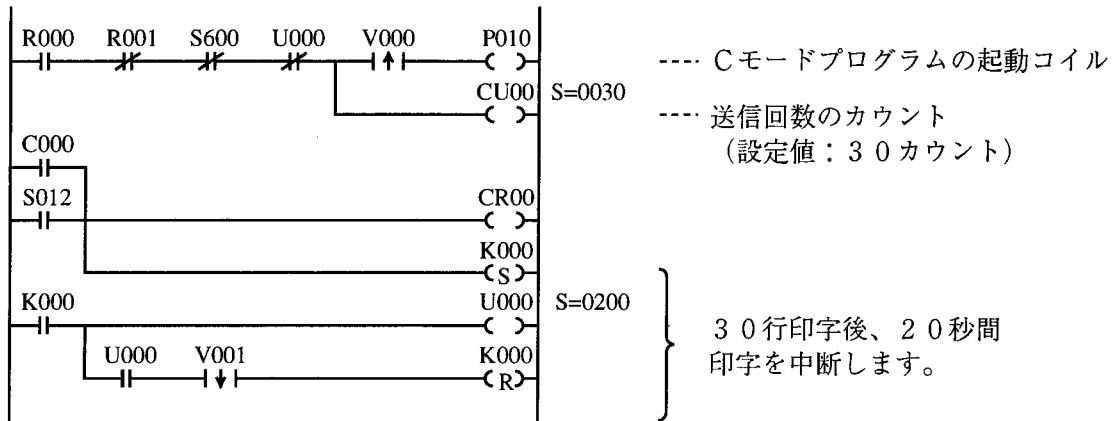
拡張メモリはCPU内蔵と拡張メモリモジュール（1Mバイト／モジュール）を合計して最大4Mバイトまで実装できます。



## 6. 1. 11 ラダープログラム


プリンタに出力するためには、P 0 1 0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を次に示します。



R 0 0 0 ……印字指令  
 R 0 0 1 ……印字完了  
 S 6 0 0 ……外部機器リンクの転送中  
 C 0 0 0 ……30行印字カウンタ  
 U 0 0 0 ……30行印字後、中断タイマ  
 S 0 1 2 ……STOP→RUN信号  
 K 0 0 0 ……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。

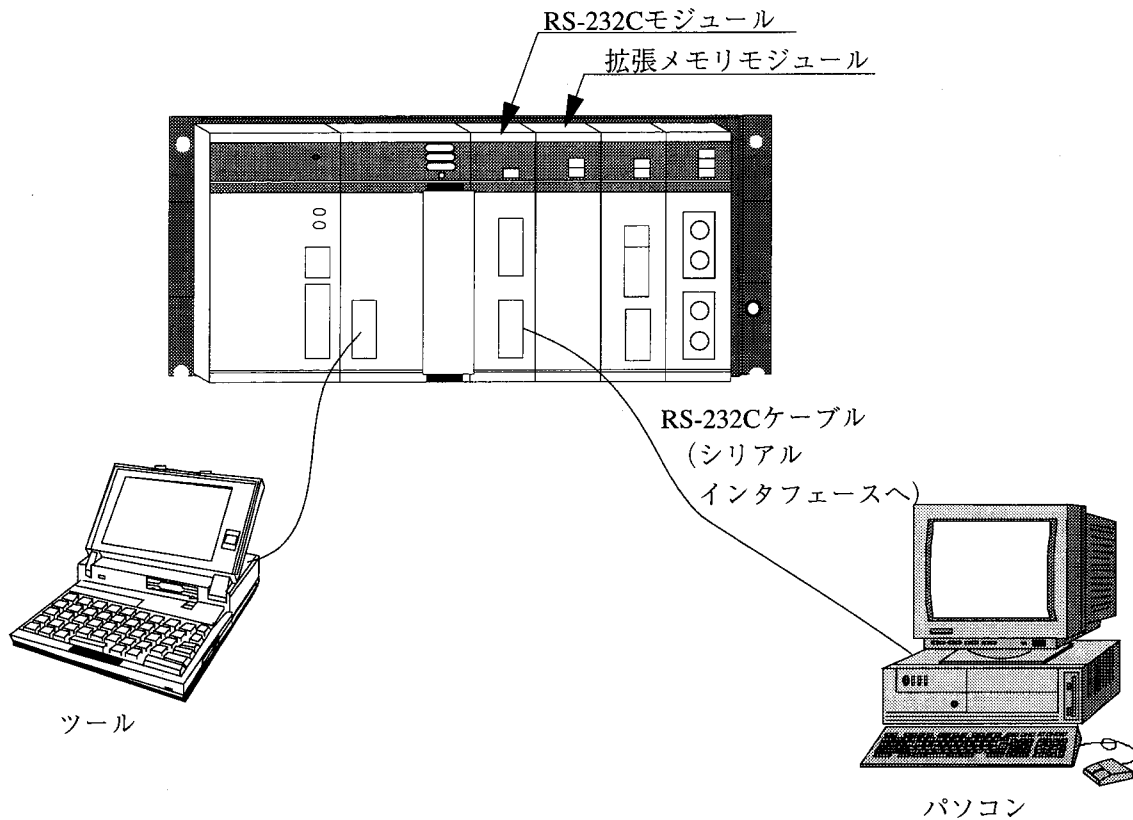
 注 意

プログラムの例は、理解しやすいことを目的に書いていますので、実用のプログラムでは、送信ハンドラのリターンコードおよびシステムレジスタ (S) のエラーチェックを行ってください。

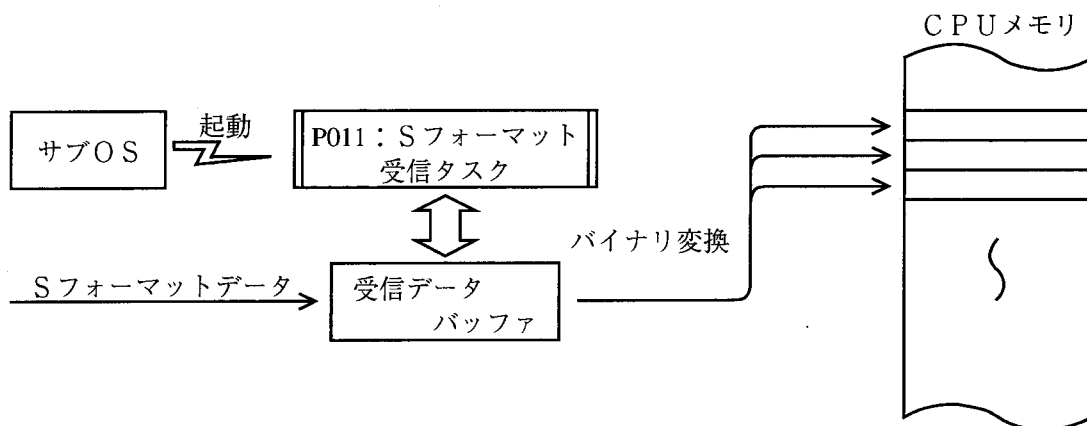
6.2 パソコンによるプログラムローディング

6.2.1 システム構成

パソコンをRS-232Cインタフェースを用いて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。



6.2.2 プログラム構成



パソコンより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。

## 6.2.3 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。

データフォーマット

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' (53 h) 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' (32 h) : データレコード  
'9' (39 h) : 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。

(補足) 今回使用したパソコンでは各レコードのサムデータの後にCR (キャリッジリターン: 0D h), LF (ラインフィード: 0A h) が追加されて転送されました。そこでこのこと  
を利用し、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' (=53 h) … 1文字

エンドコード (ECD) = CR (0D h) + LF (0A h) … 2文字

## 6 プログラム例

### 6.2.4 LGBの設定

LGBの設定例を示します。

```
-----  
CHANNEL No.0 LGB DATA  
-----  
0 : DATA FRAME ----- ST+8DT+OP+1SP  
1 : BAUD RATE ----- 1200 [BPS]  
2 : PRIORITY LEVEL ----- SELF ----- 自局優先  
3 : DATA CHANGE MODE ----- BINALY ----- 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 MODE ----- NO BCC  
8 : SEND DELAY TIME ----- NO DELAY  
9 : SEND BREAK/CONTINUE ----- NO BREAK/CONT.  
A : SEND BREAK TIMEOUT ----- 32767 [100 mSEC]  
B : RECIEVE 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 : 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、ここでは遅めに設定しました。

TEXT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。

TEXT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。

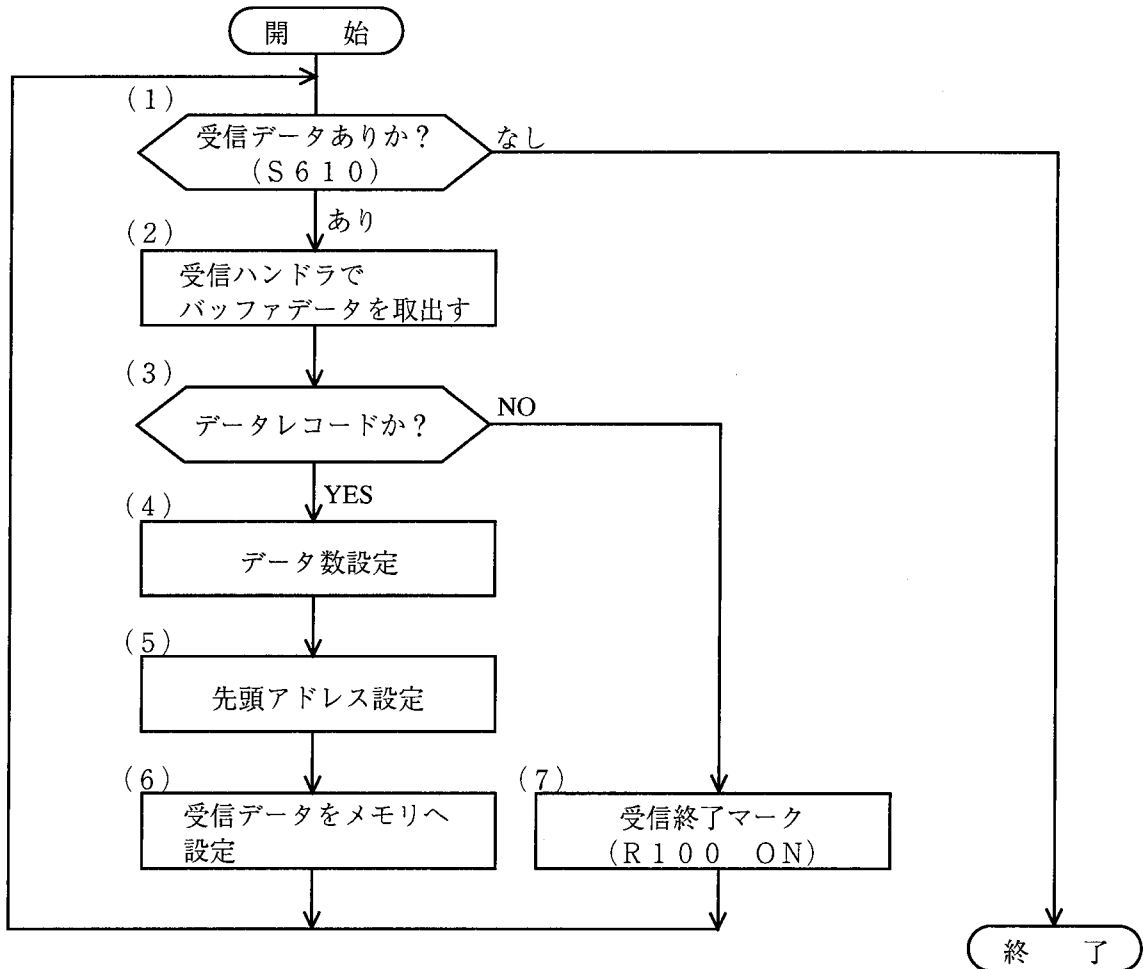
### 6.2.5 受信タスクの登録

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

```
-----  
CHANNEL No.0 LGB DATA  
-----  
⋮  
G : SYSTEM MODE (TASK) ----- TN:11 FT:01  
-----
```

## 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) ここで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。
- (7) 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 (HTR100) 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 プログラム例

### 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] ) ;  
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 l= wkl ;
70: return( wkh ) ;
71: }
72: /*****/
```





# 7 保 守

## 7.1 保守点検

RS-232Cモジュールを最適な状態で使用するには、次のような点検を行ってください。点検は、日常あるいは定期的（2回／年以上）に行ってください。

## 保守点検項目

項 目	チェックポイント
モジュールの外観	モジュールケースにひび、割れなどが点検してください。ケース類に異常があると内部回路に破損が生じている場合があり、システムの誤動作原因となります。
LED	RS-232CモジュールのERR LEDが点灯していないか点検してください。
取付けネジのゆるみ	モジュール取付け、およびRS-232Cケーブルの取付けネジなどにゆるみがないか点検してください。ゆるみがある場合には、増し締めを行ってください。ネジにゆるみがあるとシステムの誤動作、さらには加熱による焼損の原因となります。
ケーブルの被覆の状態	ケーブルの被覆に異常がないか点検してください。被覆が剥がれているとシステムの誤動作、感電、さらにはショートによる焼損の原因となります。
ほこり類の付着状態	モジュールにほこり類が付着していないか点検してください。付着しているときは、電気掃除機などで吸い取ってください。ほこりが付着すると内部回路がショートし、焼損の原因となります。
モジュールの交換	活線時の交換は、ハードウェア、ソフトウェアの破壊につながりますので、必ず電源OFFの状態で行ってください。
コネクタの状態	コネクタのコンタクト部にほこりやゴミが付着するとコネクタの特性が劣化し故障の原因となります。未使用のコネクタには、必ず付属の保護キャップをかぶせてください。

 注 意

- 静電気によりモジュールが破損する恐れがあります。作業を行う前に、人体の静電気を放電してください。
- モジュールを交換する作業は、必ず電源を切ってから行ってください。感電の恐れがあります。また、モジュールの破損、誤動作の恐れがあります。

## 7.2 ユーザ設定項目のバックアップ

### 7.2.1 LGBテーブル、受信タスク登録テーブル、ユーザ演算ファンクション登録テーブル

#### (1) RS-232Cモジュール内のフラッシュメモリによるバックアップ

LGBテーブル、受信タスク登録テーブル、ユーザ演算ファンクション登録テーブルは、ツールから編集した後のCPUモジュールのリセットスイッチによるリセット操作でRS-232Cモジュール内のフラッシュメモリへ書込まれバックアップされます。復電時は、RS-232Cモジュール内のフラッシュメモリに書込まれているLGBテーブル、受信タスク登録テーブル、ユーザ演算ファンクション登録テーブルを読み取り、動作を開始します。

#### (2) 一括セーブによるバックアップ

LGBテーブル、受信タスク登録テーブル、ユーザ演算ファンクション登録テーブルの一括セーブでバックアップについて以下の表に示します。

名称	チャンネルNo.	アドレス	一括セーブ
LGBテーブル	No.1	/F48100~/F481FE	○
	No.2	/F58100~/F481FE	○
	No.3	/F68100~/F481FE	○
	No.4	/F78100~/F481FE	○
受信タスク登録テーブル	No.1	/1070CA~/1070D0	× (ユーザ指定により可)
	No.2	/10714A~/107150	× (ユーザ指定により可)
	No.3	/1071CA~/1071D0	× (ユーザ指定により可)
	No.4	/10724A~/107250	× (ユーザ指定により可)
ユーザ演算ファンクション登録テーブル	No.1~No.4	/FAB40~/FAD3E	○

### ⚠ 注 意

- 受信タスク登録テーブルは、一括セーブで自動セーブされませんので、ユーザがセーブするときにアドレスを指定してください。  
LGBテーブルとユーザ演算ファンクション登録テーブルは、一括セーブで自動セーブされます。
- ツールからの編集後、あるいは、一括ロード後に実施するリセット前、または、リセット中に停復電があると、編集や一括ロードで設定した内容ではなく、RS-232C内のフラッシュメモリに書込まれている内容が有効になります。その場合は、再度、編集、または、一括ロード後にリセットを実施してください。

### 7.2.2 モジュールを交換した場合

#### (1) RS-232Cモジュールを交換した場合

RS-232Cモジュールを故障などにより交換した場合、LGBの設定には次の2つの方法があります。

- ・ ツールからLGBテーブル、受信タスク登録テーブルを設定しCPUモジュールのリセットスイッチでリセットを実施してください。
- ・ 一括セーブでバックアップされている場合は、ツールからローディング後にリセット操作により、一括セーブでセーブされていた内容がRS-232Cモジュール内のフラッシュメモリへ書込まれ、以降有効になります。

#### (2) CPUモジュールを交換した場合

CPUモジュールの故障などによりCPUモジュールを交換した場合には、LGBテーブルと受信タスク登録テーブルの再設定は不要です（RS-232Cモジュール内フラッシュメモリに書込まれているため）。ただし、一括ロード後にリセットされると、一括ロードされていた内容が有効になります。

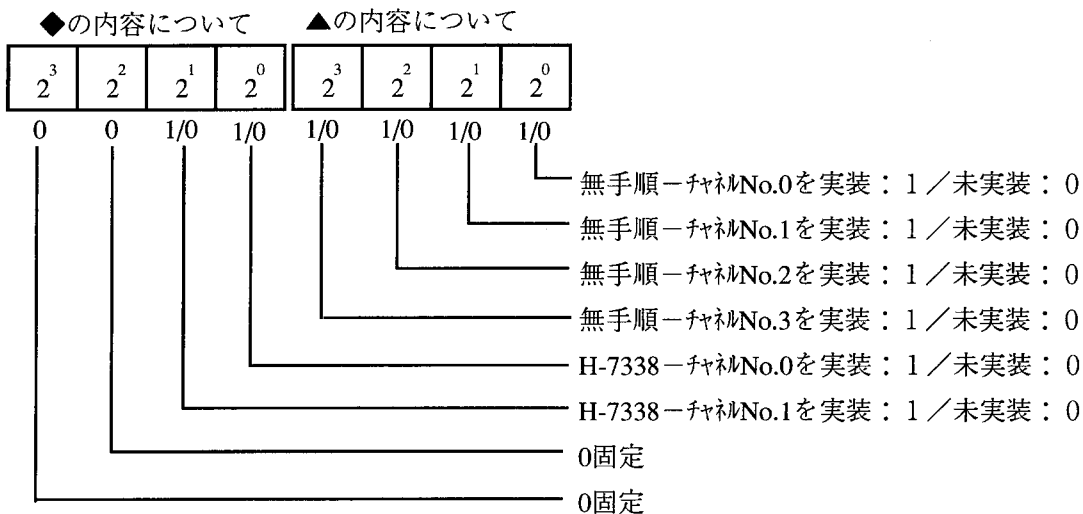
## 7.3 トラブルシューティング

### 7.3.1 CPUモジュールインディケータ表示

RS-232CモジュールがCPUモジュールのインディケータに表示するメッセージを示します。

CPUモジュール インディケータ表示	内容
'RS☆☆ F ◆▲'	演算ファンクションシステムが正常動作
'RS☆☆ T ◆▲'	タスクシステムが正常動作
'H-73 38◆▲'	H-7338サポートシステムが正常動作
'GE?S △△△△'	送信エラー発生
'GE?R □□□□'	受信エラー発生
'GE?O □□□□'	ROMエラー発生
'G?DN □□□□'	WDTエラー発生
'EXA? PTY'	RAMパリティエラー発生

- (1) ▲：無手順で使用するチャンネルNo.を表示します。  
 (2) ◆：H-7338プロトコルで使用するチャンネルNo.を示します。



CPUモジュールのインディケータ表示は、無手順または、H-7338プロトコルのどちらか一方のみの表示になります。チャンネルごとに無手順か、H-7338プロトコルのどちらで使用するかは、上記のように表示の右側の数値で分かります。

- (3) ?：チャンネルNo.を示します。  
 (4) ☆☆：バージョン、レビジョンを示します。  
 (5) △△△△：「7.3.2 送信エラーコード表」を参照してください。  
 (6) □□□□：「7.3.3 受信エラーコード表」を参照してください。

7.3.2 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レジスタのビット配列と同一です。また、エラー発生時、ハンドラエラー以外はCPUモジュールのインディケータに表示を行います。

インディケータ 表示上位4文字	上位 バイト	下位 バイト	内 容	無手順	H-7338 プロトコル
インディケータ 表示しません	40h	01h	送信起動すべきモジュールが未実装、またはMODU NOスイッチ設定誤り。	○	—
		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	送信起動すべきチャンネルがダウン。	○	—
'GE?S'	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。	○	—
			送信リトライオーバ報告、コマンド、パラメータ返 送時、または、データ返送時に送信異常発生。	—	○
		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 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、送信再開されなかつ 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 信を再開した。	○	—	

? : チャンネルNo.

## 7.3.3 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レジスタのビット配列と同一です。

インディケータ表示上位4文字	上位バイト	下位バイト	内 容	無手順	H-7338プロトコル
インディケータ表示しません	41h	×× (不定)	受信ハンドラ起動時、モジュール未実装、またはMODU NOスイッチ設定誤り。	○	—
	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)	チャンネ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。	○	—
' GR ? R'	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	オーバランエラー。受信データでオーバランエラー発生。	○	○
		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。ASCII変換指定時、'0'~'9'および、'A'~'F'以外のデータまたは、エンドコード以外のデータを受信した。	○	—
			コードエラー。規定以外のコマンド受信。規定位置以外でのスペースデータ受信。パラメータエラー発生。	—	○
		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）入力がかリヤなしとなり受信中断され、受信監視時間内にCD入力がキャリアありにならず、受信再開されなかった。	○	—
02h	受信打ち切り発生。時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。	○	—		
' GE ? O'	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。	○	○
		F5h	RAMパリティエラー。RAMエリアでパリティエラー発生。	○	○
		F6h	MODU NOスイッチ設定誤り。設定が'A'~'F'になっている。	○	○
		F7h	LGBバックアップデータ異常。	○	—
' G ? DN'	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。	○	○

? : チャンネルNo.

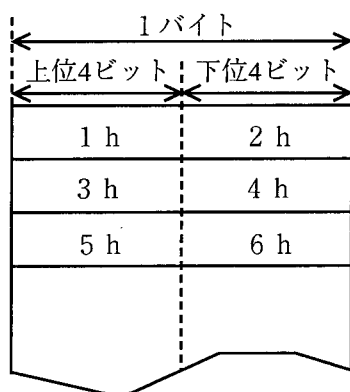
' 10F0'、' 10F5'、' 10FF' の場合、CPUのリセットスイッチにてリセットを行ってください。再度エラーが発生する場合は、モジュールを交換してください。

' 10F7' の場合、LGBを再設定後、CPUのリセットスイッチにてリセットを行ってください。同じエラーが発生する場合は、モジュールを交換してください。

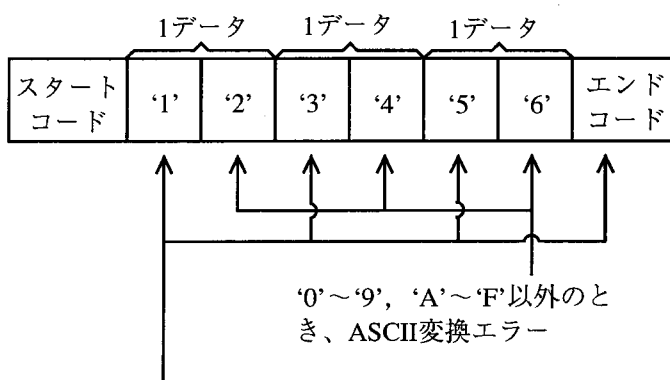
## 7 保 守

外部機器よりデータ 1 2 h, 3 4 h, 5 6 h を RS-232C モジュールに送信したとき各データの低位ビットデータが '0' ~ '9', 'A' ~ 'F' 以外するとき (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するとき (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。

外部機能の送信データ  
格納エリア



回線上のデータ

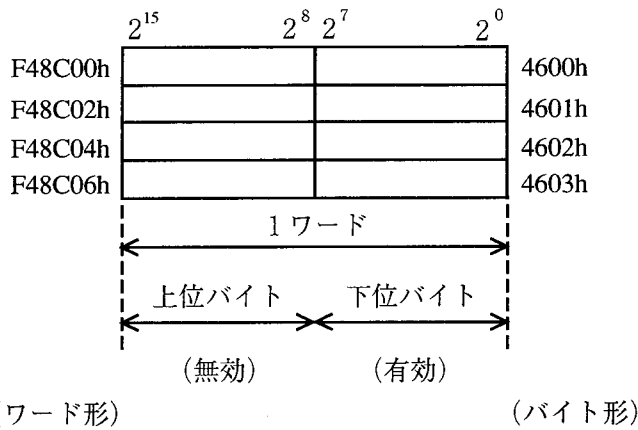


'0' ~ '9', 'A' ~ 'F' 以外または、エンドコード以外するとき  
エンドコードエラー。



7.3.4 RS-232Cモジュールのメモリマップ

システムバスのアドレス				外部機器リンク	
チャネルNo.0用	チャネルNo.1用	チャネルNo.2用	チャネルNo.3用	MCSアドレス	
F48000H	F58000H	F68000H	F78000H	システム テーブル	4000H
F48100H	F58100H	F68100H	F78100H	LGB テーブル	4080H
F48200H	F58200H	F68200H	F78200H	システム ワーク	4100H
F48C00H	F58C00H	F68C00H	F78C00H	送信 データバッファ	4600H
F49000H	F59000H	F69000H	F79000H	受信 データバッファ	4800H
F4A000H	F5A000H	F6A000H	F7A000H	送受信 トレースバッファ	5000H
F4BFFEh	F5BFFEh	F6BFFEh	F7BFFEh		5FFFH



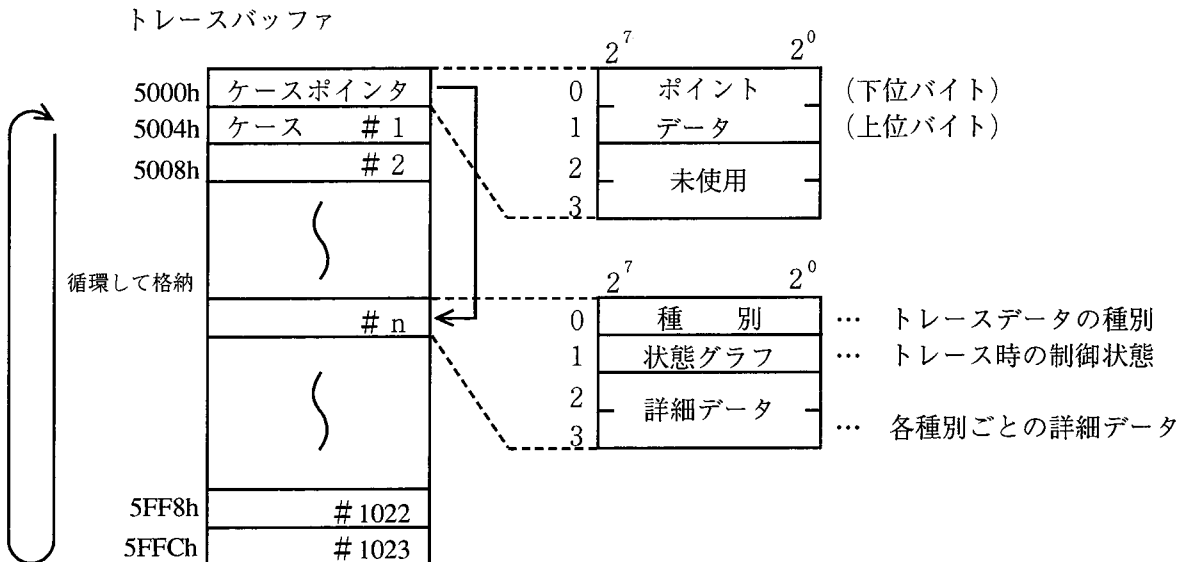
- ・RS-232Cモジュールには、通信用マイクロプロセッサとして8085（8ビット）をチャンネルナンバごとに各1個搭載されており、CPUの68000のシステムバスとは、デュアルポートRAMメモリを用いてデータの交信を行っています。
- ・CPU側のメモリは68000のワード（16ビット）形となっており、実際には、下位バイト（ $2^7 \sim 2^0$ ビット）のみ有効です。
- ・このCPUアドレス方式ではデータの読み書きが実施しづらいため「外部機器リンクサポートシステム」のMCS機能を準備しました。
- ・この「外部機器リンクサポートMCS」は、RS-232Cモジュールのチャンネルナンバ単位で、8ビットマイクロプロセッサのメモリを読み出し、書換えすることができます。
- ・タスクなどから、RS-232C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。

## 7 保 守

### 7.3.5 トレースバッファ

#### (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より構成され、ポインタにより循環して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した場合、次のデータはケース # 1 へ設定されます。

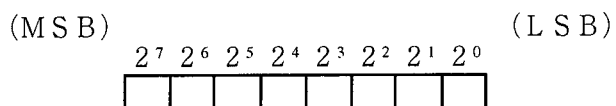
#### (2) トレースデータの種別

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、Windows®パソコンの「外部機器リンクFor Windows®」のMCS機能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などの要求を示します。
'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。

## (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。



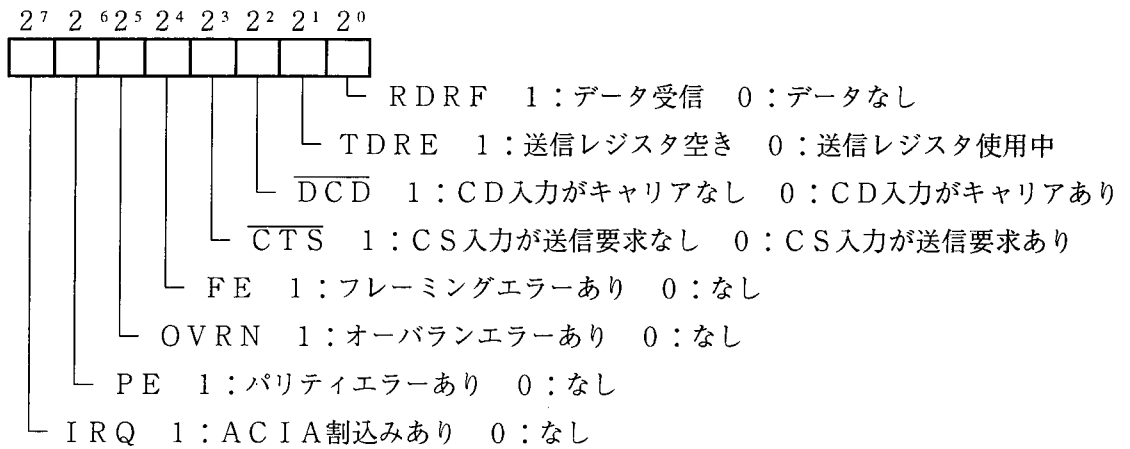
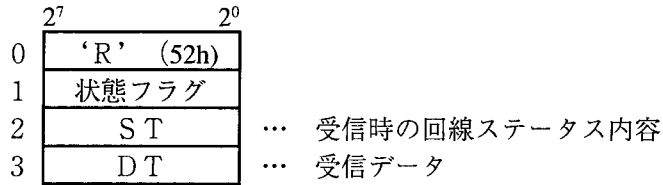
ビット	項 目	ビ ッ ト 状 態	
		‘1’	‘0’
0	将来拡張用	/	/
1			
2			
3			
4	受信の可/否	現在、CD入力がかリヤなしのため、受信不可能である。	CD入力がかリヤありで受信可能。
5	送信中断	中断コード受信、CS入力がか送信要求なしまたは、DR入力がかノットレディのため送信を中断している。	中断していない。
6	TEXT受信中	現在、TEXTデータ受信中である。	受信中ではない。
7	データ送信中	現在データ送信中である。	送信中ではない。

## 7 保 守

### (4) トレースの詳細データ

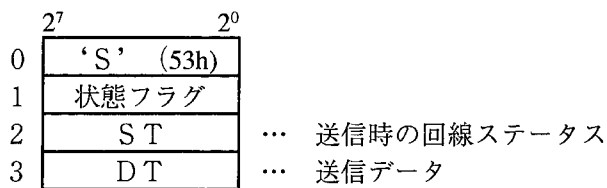
#### (4-1) 'R' (52h) データ受信

データを受信したことを示します。



#### (4-2) '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。



(4-3) '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	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)のセット要求	○			あり
6	D		40	03	RTS(RS)のセット要求	○			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	(※2)	01	83	中断タイムアウト報告に対する応答 (I-2)			○	
10	D		01	84	CS入力が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入力が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)			○	

(※3)

(※1)

- ※1 ( )内の数値は'I'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 (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

## 7 保 守

(4-4) '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入力が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入力が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	(※2)							
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		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 not ready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	I		40	03	RSの送信要求有/無セット要求に 対する応答 (D-6)			○	

(※3)

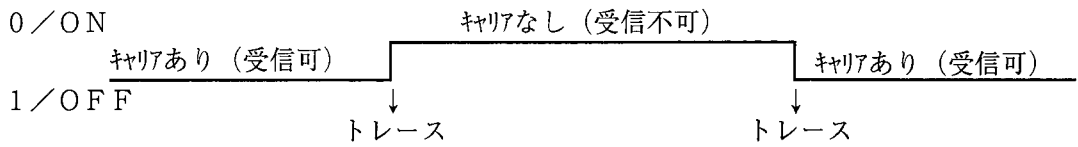
(※1)

- ※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 (3) 状態フラグ参照
- ※3 I = 49h : Interrupt status word、インター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

(4-5) 'E' (45h) CD入力の変化による受信可否状態

	2 <sup>7</sup>	2 <sup>0</sup>	
0	'E' (45h)		
1	状態フラグ		
2	ST		… CD入力変化時の回線ステータス
3	DT		… CD入力変化時の受信データレジスタ内容

CD入力

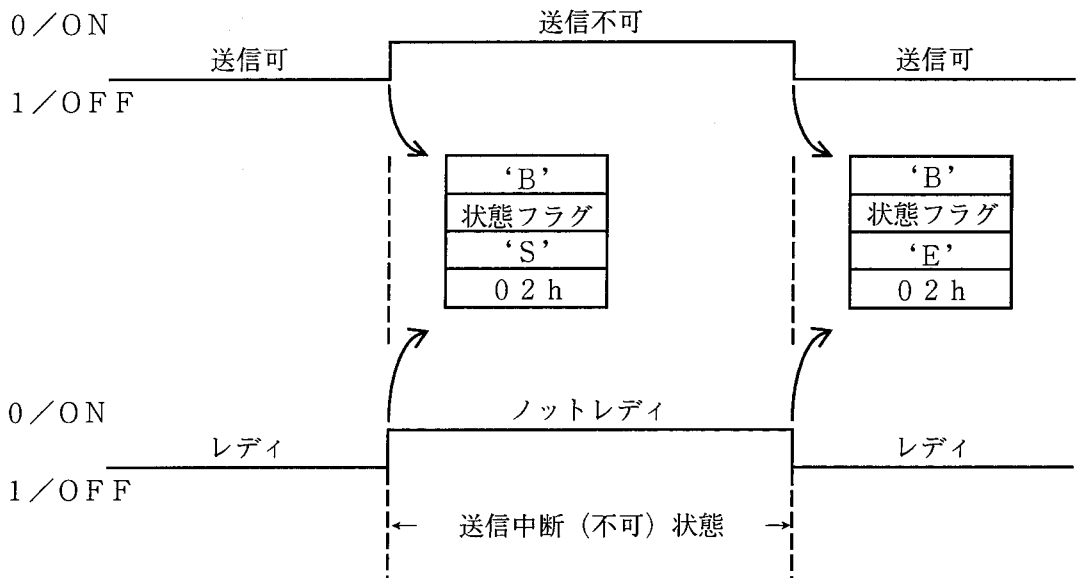


CD入力の変化時にトレースされます。

(4-6) 'B' (42h) CSまたはDRの変化による送信中断/再開状態

	2 <sup>7</sup>	2 <sup>0</sup>	
0	'B' (42h)		
1	状態フラグ		
2	ST		… { 'S' (=53h) 中断スタート 'E' (=45h) 中断終了
3	02h		

CS入力



<このページは余白です>



# 付 録

A. 1 J I S 7 単 位 コー ド 表 (C6220)

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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	0	1	1	0	0	1	1	0	1	1
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

- (1) 引用符
- (2) アポストロフ
- (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

A. 2 J I S 8 単 位 コー ド 表 (C6220)

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	8	9	10	11	12	13	14	15
0 0 0 0	0	0	0	0	0	0	0	NUL	(TC <sub>7</sub> )DLE	SP	0	@	P	,	p	↑	↑		(12)	タ	ミ	↑	↑
0 0 0 1	1	(TC <sub>1</sub> )SOH	DC <sub>1</sub>	!	1	A	Q	a	q								◦(9)	ア	チ	ム			
0 0 1 0	2	(TC <sub>2</sub> )STX	DC <sub>2</sub>	" (1)	2	B	R	b	r									「	イ	ツ	メ		
0 0 1 1	3	(TC <sub>3</sub> )ETX	DC <sub>3</sub>	#	3	C	S	c	s									」	ウ	テ	モ		
0 1 0 0	4	(TC <sub>4</sub> )EOT	DC <sub>4</sub>	\$	4	D	T	d	t									、(10)	エ	ト	ヤ		
0 1 0 1	5	(TC <sub>5</sub> )ENQ	(TC <sub>8</sub> )NAK	%	5	E	U	e	u									、(11)	オ	ナ	ユ		
0 1 1 0	6	(TC <sub>6</sub> )ACK	(TC <sub>9</sub> )SYN	&	6	F	V	f	v										カ	ニ	ヨ		
0 1 1 1	7	BEL	(TC <sub>10</sub> )ETB	' (2)	7	G	W	g	w										ア	キ	ヌ	ラ	
1 0 0 0	8	FE <sub>0</sub> (BS)	CAN	(	8	H	X	h	x										イ	ク	ネ	リ	
1 0 0 1	9	FE <sub>1</sub> (HT)	EM	)	9	I	Y	i	y										ウ	ケ	ノ	ル	
1 0 1 0	10	FE <sub>2</sub> (LF)	SUB	* ;(6)	J	Z	j	z											エ	コ	ハ	レ	
1 0 1 1	11	FE <sub>3</sub> (VT)	ESC	+ ;(7)	K	[	k	{											オ	サ	ヒ	ロ	
1 1 0 0	12	FE <sub>4</sub> (FF)	IS <sub>4</sub> (FS)	, (3)	<	L	¥	l											ヤ	シ	フ	ワ	
1 1 0 1	13	FE <sub>5</sub> (CR)	IS <sub>3</sub> (GS)	-(4)	=	M	]	m											ユ	ス	ヘ	ン	
1 1 1 0	14	SO	IS <sub>2</sub> (RS)	. (5)	>	N	^	n	—										ヨ	セ	ホ	°(13)	
1 1 1 1	15	SI	IS <sub>1</sub> (US)	/ ?	0	_ (8)	o	DEL											ッ	ソ	マ	°(14)	

- (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
- (9) 句 点
- (10) 読 点
- (11) 中 点
- (12) 長 音
- (13) 濁 点
- (14) 半濁点

E<sub>0</sub>, N<sub>0</sub>: 未定義特殊符号

## A. 3 制御符号の説明

制御符号	コード	制御付号名	意味
NUL	00	Null	空き
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 Tabulation	水平タブ
LF/NL	0A	Line Feed/New Line	改行/復改（復帰・改行）
VT	0B	Vertical Tabulation	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	
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している）
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

## A. 4 略号の説明

略 号	意 味
A C I A	Asynchronous Communications Interface Adapter
A S C I I	American Standard Code for Information Interchange
B C C	Block Check Character
B P S	Bits Per Second
C D	data Carrier Detect
C P M S	Compact Process Monitor System
C P U	Central Processing Unit
C R T	Cathode Ray Tube
C S	Clear to Send
D R	Data set Ready
E C D	End Code
E I A	Electronic Industries Association
E O R	Exclusive OR
E R	Equipment Ready
F E	Framing Error
F G	Frame Ground
G R	General Reset
I R Q	Interrupt Request
L E D	Light Emitting Diode
L G B	Line Group Block
M C S	Man-machine Communication System
O V R N	Overrun error
P C s	Programable Controlers
P E	Parity Error
R D	Receive Data
R S	Request to Send
S C D	Start Code
S D	Send Data
S G	Signal Ground
S H D	Shield
T E R M	Termination Registance
U F E T	User Function Edition Table
W D T	Watch Dog Timer

A. 5 トラブル調査書

貴会社名		担当者		発生日時	月	日	時	分
ご連絡先	ご住所							
	T E L							
	F A X							
不具合モジュール形式			CPU形式					
OS Ver. Rev.		プログラム名：				Ver.	Rev.	
サポートプログラム		プログラム名：				Ver.	Rev.	
不具合現象								
接続負荷	種 類							
	形 式							
	配線状態							
システム構成およびスイッチ設定								
通 信 欄								

補足資料

補足資料 モジュールの交換、増設

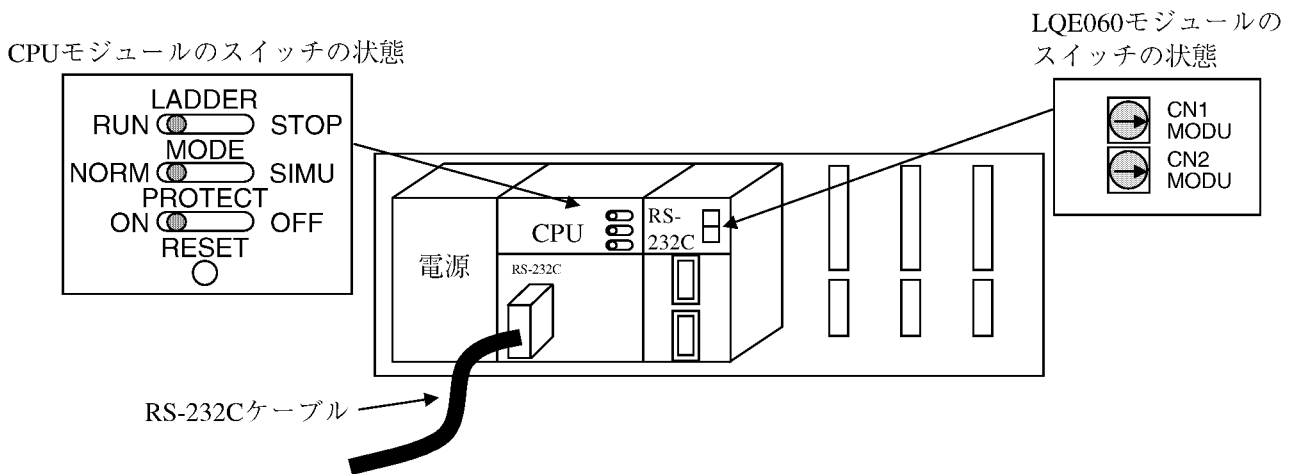
● 交換前準備品

- ① パソコン (Hitachi S10外部機器リンクシステム組み込み済み)
- ② RS-232Cケーブル (ET.NET使用の場合、10BASE-T)
- ③ RS-232Cモジュール (LQE060)
- ④ 交換対象モジュールのパラメータ値 (パラメータが読み出せない場合に使用します。)
- ⑤ オプションモジュールにET.NETが実装されている場合は、通信種類をET.NETにすることができます。

「ユーザズマニュアル オプション ET.NET (LQE520) (マニュアル番号 SVJ-1-103)」の「2.1 各部の名称と機能」、「3.2 モジュールの実装」を参照してください。

● 交換手順

- ① 実装されているRS-232Cモジュール前面のロータリスイッチの設定を記録します (CN1 MODU, CN2 MODU)。
- ② CPUモジュール前面のスイッチの状態を記録します (LADDER, MODE, PROTECT)。
- ③ パソコンとCPUモジュールをRS-232Cケーブルで接続します。



- ④ Hitachi S10外部機器リンクシステムを立ち上げ、設定されているパラメータを記録します。
  - ・各チャンネルごとにLGB登録の画面を表示して、設定パラメータを記録してください。
  - (注) Hitachi S10外部機器リンクシステムの「システムプログラム転送」、「システムプログラム比較」、「システムプログラム全チャンネル削除」機能は2αシステム専用の機能なので使用しないでください。
  - ・設定パラメータが読み出せない場合は、交換前準備品④のパラメータ値を使用してください。
- ⑤ CPUモジュール前面のLADDERスイッチをSTOPにし、ユニットの電源をOFFにします。
- ⑥ RS-232Cモジュールに接続されているケーブルを取り外します。



- ⑦ 新しいモジュールと交換し、ロータリスイッチを①で記録した状態に設定します。
- ⑧ ユニットの電源をONにし、Hitachi S10外部機器リンクシステムから④で記録したパラメータを設定します。
- ⑨ ⑧設定後、④で記録したパラメータと設定内容が一致しているかを再確認します。
- ⑩ CPUモジュール前面のRESETスイッチを押し、リセットをかけます。
- ⑪ ユニットの電源をOFFにします。
- ⑫ ③で接続したRS-232Cケーブルを取り外します。
- ⑬ ⑥で取り外したケーブルを元に戻します。
- ⑭ CPUモジュール前面のスイッチを②で記録した状態に設定します。
- ⑮ ユニットの電源をONにし、正常に動作していることを確認してください。

● 増設手順

- ① CPUモジュール前面のスイッチの設定状態を記録します。
- ② システムの停止を確認後、CPUモジュールのLADDERスイッチをSTOPにし、ユニットの電源をOFFにします。
- ③ 「1. 2 オプションモジュールの実装」を参照のうえ、増設するRS-232Cモジュールを実装します。
- ④ 「3. 1 各部の名称と機能」を参照のうえ、RS-232CモジュールのCN1, CN2 MODU NOスイッチを設定します。
- ⑤ パソコンとCPUモジュールをRS-232Cケーブルで接続し、ユニットの電源をONにした後、Hitachi S10外部機器リンクシステムから増設したRS-232Cモジュールにパラメータを設定します。  
(注) Hitachi S10外部機器リンクシステムの「システムプログラム転送」、「システムプログラム比較」、「システムプログラム全チャンネル削除」機能は2 $\alpha$ システム専用の機能なので使用しないでください。
- ⑥ CPUモジュール前面のRESETスイッチを押し、リセットをかけます。
- ⑦ ユニットの電源をOFFにし、増設したRS-232Cモジュールにケーブルを接続します。
- ⑧ CPUモジュール前面のスイッチを①で記録した状態に設定します。
- ⑨ ユニットの電源をONにし、正常に動作していることを確認してください。

<このページは余白です>

ご利用者各位

〒101-8010

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

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、弊社営業担当または弊社所員に、お渡しくださいますようお願い申  
しあげます。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	_____
製 品 名	
ご意見欄	_____ _____