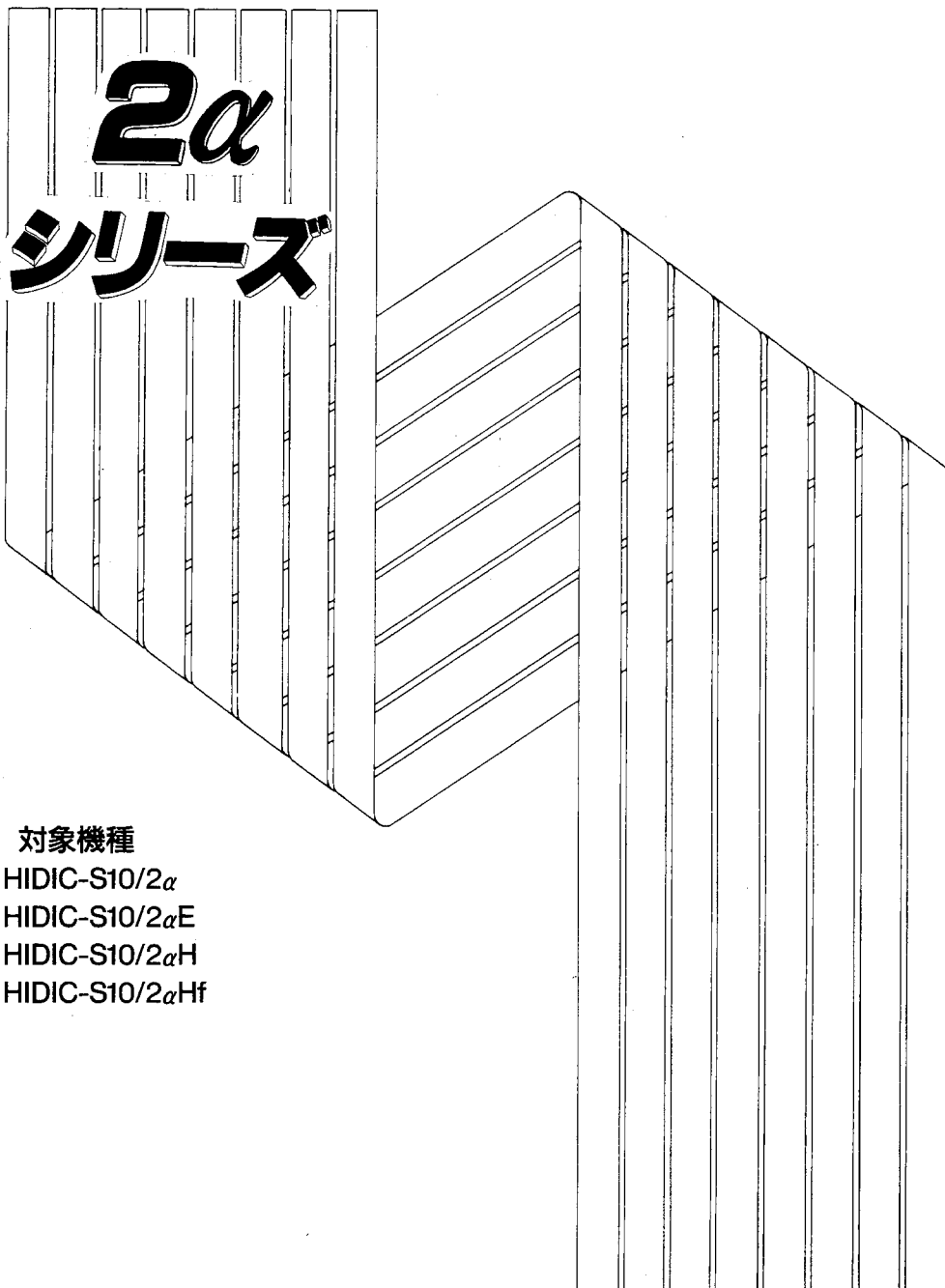


ハードウェアマニュアル  
オプション

# ETリンク V3(UDP/IPサポート)

HIDIC  
S10 $\alpha$ シリーズ



対象機種

HIDIC-S10/2 $\alpha$

HIDIC-S10/2 $\alpha$ E

HIDIC-S10/2 $\alpha$ H

HIDIC-S10/2 $\alpha$ 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。

1990年	6月	(第1版)	SP-2-021	(廃版)
1992年	11月	(第2版)	SP-2-121	(廃版)
1993年	5月	(第3版)	SP-2-221	(廃版)
1995年	3月	(第4版)	SAJ-2-110	(A) (廃版)
1997年	5月	(第5版)	SAJ-2-110	(B)

- 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。

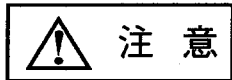
## 安全上のご注意

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。

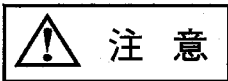
このマニュアルでは、安全注意事項のランクを「危険」「注意」として区分してあります。



：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。




：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、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S10ET LINK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「ハードウェアマニュアル オプション ETリンク」は、S10ET LINKモジュールの取扱いとS10ET LINKをLAN機構と接続する場合の配線工事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。

このマニュアルは、S10ET LINK SYSのバージョン3.0以降に対応しています。

バージョン2.0, 1.0のFDとマニュアルの対応は下記のとおりです。

FD 名称	バージョン	マニュアル番号
S10ET LINK SYS TYPE: S102A-35ETL	2.0	SP-2-222
	1.0	SP-2-122 SP-2-022

### 注 意

1. S10ET LINKモジュールは、HIDIC S10/2 $\alpha$ シリーズCPUと組合せて使用します。S10/2 $\alpha$ シリーズCPUとは、2 $\alpha$ (LWPO00), 2 $\alpha$ E(LWPO40), 2 $\alpha$ H(LWPO70), 2 $\alpha$ Hf(LWPO75)の総称です。
2. S10ET LINKモジュールを使用される場合は拡張メモリを512キロバイト以上実装してください。
3. S10ET LINK SYSTEM V2.0から2チャンネルサポート機能を追加しています。
4. MAIN, SUBにつながるランシーバは少なくとも1台は電源内蔵型ランシーバを使用してください。
5. S10ET LINK SYSTEM V3.0から下記機能を追加しています。
  - UDP/IPによるデータ通信機能
  - ソケットライブラリ関数のTERM機能
6. C言語プログラム開発環境システム(RPDP/S10)でソケットライブラリ関数を使用する場合、S10ET LINK SYSTEM V3.0 R1.0以降を使用してください。

# 目 次

<b>1</b>	<b>ご使用にあたり</b>	<b>1</b>
1.1	拡張CPUユニット	2
1.2	モジュールの実装	2
1.3	アース配線	4
1.4	2チャンネル使用時の注意	4
1.5	イーサネットモジュール交換時の注意	5
<b>2</b>	<b>仕 様</b>	<b>7</b>
2.1	用途	8
2.2	仕様	8
2.2.1	システム仕様	8
2.2.2	回線仕様	8
<b>3</b>	<b>各部の名称と機能, 配線</b>	<b>9</b>
3.1	各部の名称と機能	10
3.2	配線	11
<b>4</b>	<b>利用の手引き</b>	<b>13</b>
4.1	システム構成	14
4.2	S10/2αによるシステム構成例	19
4.3	システム定義情報	20
4.3.1	物理アドレス	20
4.3.2	IPアドレス	20
4.3.3	ホスト名	22
4.4	S10ET LINKのソフトウェア構成	23
4.5	S10ET LINKに提供されるシステムプログラム	24
4.5.1	ソケットライブラリ	24
4.5.2	TCPプログラム	24
4.5.3	UDPプログラム	24
4.5.4	IPプログラム	25
4.5.5	ドライバ	25
4.6	ユーザの作成するプログラム, テーブル	26
4.6.1	ユーザプログラム	26
4.6.2	HOSTSテーブル	26
4.7	ソケットライブラリ	28
4.7.1	ソケットライブラリー一覧	29



4. 7. 2	パラメータのデータ構造	53
4. 8	アプリケーション作成上の注意	54
4. 9	ソケットライブラリ発行手順例	55
4. 9. 1	TCP/IPプログラム使用例	55
4. 9. 2	UDP/IPプログラム使用例	56
<b>5</b>	<b>プログラム例</b>	<b>57</b>
5. 1	ソケットライブラリによるCPU間通信プログラム例	58
5. 1. 1	システム構成	58
5. 1. 2	プログラム構成	59
5. 1. 3	HOSTSテーブル例	60
5. 1. 4	ET1側プログラムのフローチャート	61
5. 1. 5	ET1側のC言語プログラム例	63
5. 1. 6	ET2側プログラムのフローチャート	64
5. 1. 7	ET2側のC言語プログラム例	65
5. 2	ソケットライブラリによるCPU間連続通信プログラム例	66
5. 2. 1	システム構成	66
5. 2. 2	プログラム構成	66
5. 2. 3	HOSTSテーブル例	68
5. 2. 4	イニシャルタスク (task1) の プログラムのフローチャート	68
5. 2. 5	イニシャルタスク (task1) のC言語プログラム例	69
5. 2. 6	ET1側 (task3) プログラムのフローチャート	70
5. 2. 7	ET1側 (task3) のC言語プログラム例	72
5. 2. 8	ET2側 (task2) プログラムのフローチャート	73
5. 2. 9	ET2側 (task2) のC言語プログラム例	75
<b>6</b>	<b>メモリマップ</b>	<b>77</b>
6. 1	CPUのメモリマップ	78
6. 2	SIOET LINKのメモリマップ	79
<b>7</b>	<b>PSE<math>\alpha</math>オペレーション</b>	<b>81</b>
7. 1	システムを立上げるにあたり	82
7. 1. 1	システム構成	82
7. 1. 2	基本オペレーション	83
7. 2	システム立上げ	84
7. 2. 1	SIOET LINKシステム立上げ手順	84
7. 2. 2	PSE $\alpha$ システム立上げ手順	85
7. 3	SIOET LINKシステムプログラム書込	87

7. 3. 1	機 能	87
7. 3. 2	オペレーション	88
7. 4	S10ET LINKシステムプログラム比較	89
7. 4. 1	機 能	89
7. 4. 2	オペレーション	90
7. 5	S10ET LINKモジュールのセットアップ	92
7. 5. 1	機 能	92
7. 5. 2	オペレーション	93

<b>8</b>	<b>パソコンプログラミング オペレーション</b>	95
8. 1	インストール	96
8. 2	システム立上げ	98
8. 2. 1	立上げ手順	98
8. 2. 2	コマンド体系	99
8. 3	S10ET LINKシステムプログラム転送	100
8. 4	S10ET LINKシステムプログラム比較	102
8. 5	S10ET LINKモジュールセットアップ	104

<b>9</b>	<b>エラーと対策</b>	107
9. 1	PSE $\alpha$ のエラーと対策	108
9. 2	パソコンプログラミングのエラーと対策	109
9. 3	CPU LED表示メッセージ	110
9. 3. 1	CPUのエラー表示と対策	110
9. 3. 2	立上げメッセージ	110
9. 4	ソケットライブラリのエラーと対策	111

<b>10</b>	<b>付 録</b>	115
10. 1	ネットワーク構成部品	116
10. 1. 1	LWE400とイーサネットとの接続の問題点	116
10. 1. 2	構成品一覧表	116
10. 1. 3	トランシーバ (タップ形) HLT-200TB	118
10. 1. 4	トランシーバ (コネクタ形) HLT-200	118
10. 1. 5	マルチポートトランシーバ H-7612-64	119
10. 1. 6	リピータ HLR-200H	119
10. 1. 7	同軸ケーブル (屋内用) HBH-CX-100	120
10. 1. 8	同軸コネクタ HBN-N-PC	121
10. 1. 9	中継コネクタ HBN-N-AJJ	122
10. 1. 10	ターミネータ (J形) HBN-T-NJ	122

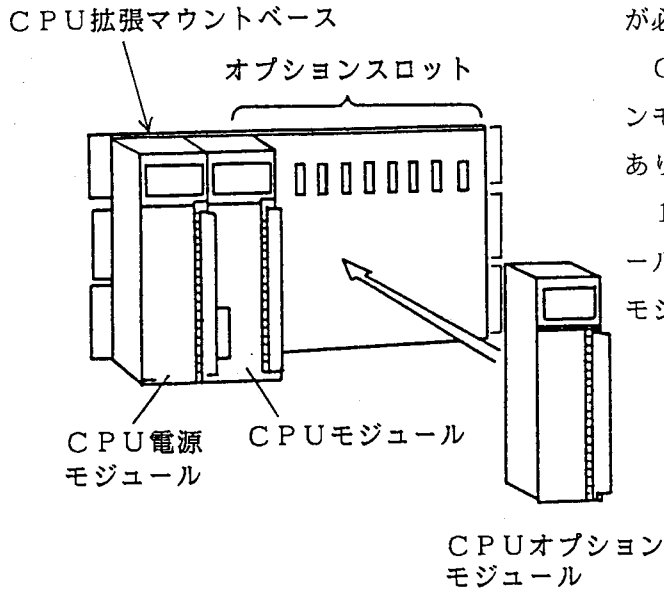
10. 1. 11	ターミネータ (P形) HBN-T-NP	122
10. 1. 12	アース端子 HBN-G-TM	123
10. 1. 13	トランシーバケーブル HBN-TC-100	123
10. 1. 14	変換器	125
10. 2	施工分担	126
10. 3	同軸ケーブルの配線	127
10. 3. 1	ケーブルセグメントの布設	127
10. 4	トランシーバ (コネクタ用) の設置・取付	128
10. 5	トランシーバ (タップ形) の設置・取付	132
10. 6	同軸コネクタの取付	132
10. 7	タップコネクタの取付	134
10. 8	トランシーバケーブルの取付	136
10. 9	ターミネータの取付	136
10. 10	リピータの設置・取付	137
10. 11	システムの接地	138
10. 12	アース端子取付方法	138
10. 13	シングルポートトランシーバの設定	139
10. 14	マルチポートトランシーバの設定および表示	140

<b>11</b>	<b>索引</b>	<b>143</b>
-----------	-----------	------------



# 1 ご使用にあたり

## 1.1 拡張CPUユニット

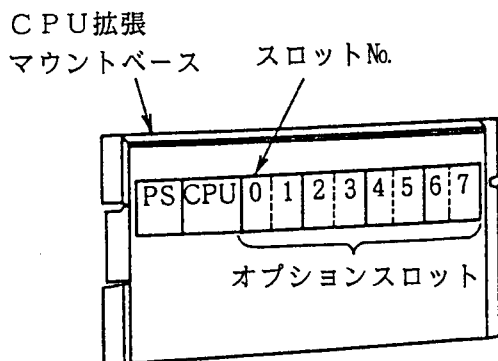


CPUオプションモジュールを使用するには、CPU拡張マウントベース（形式：HPC-1000）が必要です。

CPU拡張マウントベースには、CPUオプションモジュール用にオプションスロットが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 モジュールの実装



CPU拡張マウントベース：HPC-1000

P S スロット：CPU電源モジュール（LWV000）を実装。

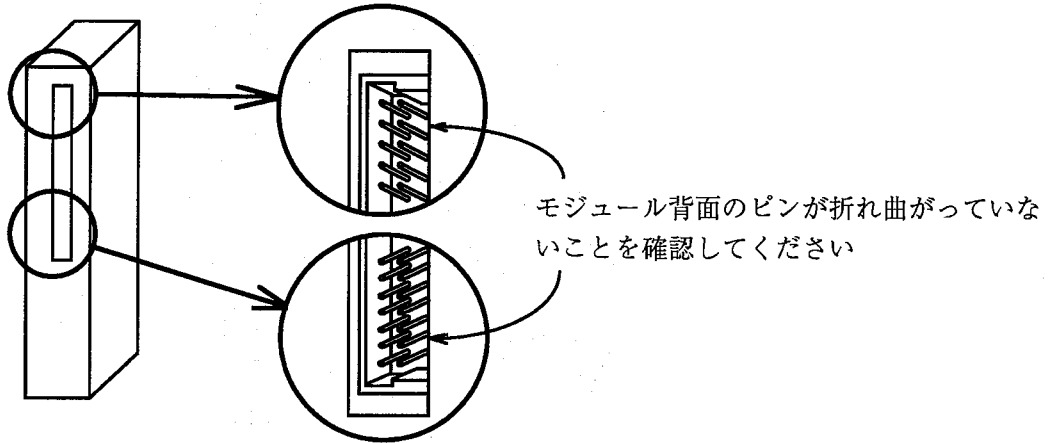
CPUスロット：CPUモジュール（LWP000）を実装。  
スロット0～7：CPUオプションモジュールを実装。

（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。）

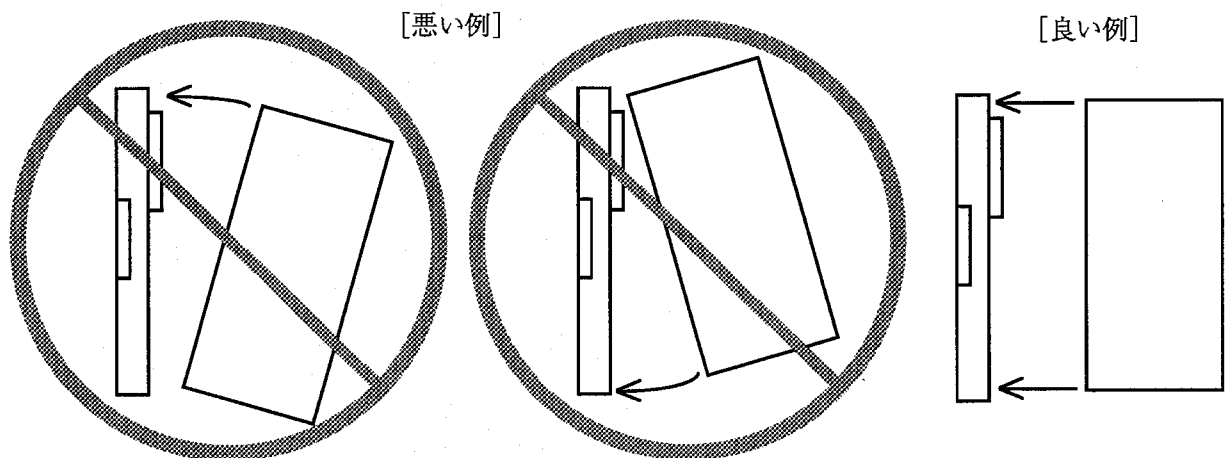
- CPUオプションモジュールは、CPUモジュールの隣りから左詰めに実装してください。

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。



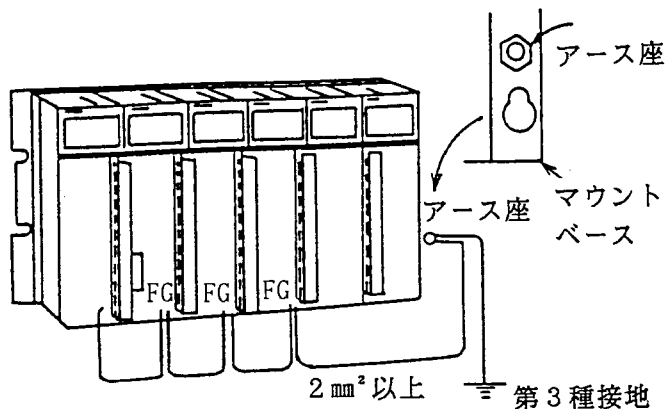
- マウントベースに対して、正面からまっすぐ実装してください。（悪い例のように斜めに実装すると、ピン曲がりが発生しオプションモジュールが誤動作することがあります。）



**⚠ 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。

### 1.3 アース配線



- ・ フレームグランド（FG）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。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。
- ・ アース線の線径は $2\text{mm}^2$ 以上を用いてください。

### 1.4 2チャンネル使用時の注意

S10ET LINKモジュールをMAIN, SUB 2台使用する場合、少なくともどちらか一方は電源内蔵型トランシーバを使用してください。



## 1.5 イーサネットモジュール交換時の注意

イーサネットモジュールの故障、修理などによりイーサネットモジュールを交換した場合、下記VER、REVのシステムF/Dでは、交換前モジュールとの回線情報を初期化するため、以下の操作が必要になりますので注意してください。

システムF/D

ツール	F/D名称	Ver, Rev	F/D型式
PSE $\alpha$	S10ET LINK SYS	VER. 3.0 REV. 1.0以前	S102A-35ETL
PC9801NS/T	S10ET LINK システム	VER. 3.3 REV. 1.0以前	H2A-ETLS-J
PS/2パソコン	S10ET LINK SYSTEM	VER. 3.2 REV. 1.0以前	H2A-ETLS-U

### (1) LWE400の通信相手を交換時

通信相手のイーサネットモジュールを交換した場合、LWE400を初期化するため、CPUのキースイッチをリセットしてください。

### (2) LWE400を交換時

LWE400を交換した場合、通信相手が記憶している回線情報を初期化してください。通信相手がLWE400の場合には、通信相手側のLWE400を初期化するため、通信相手側のCPUのキースイッチをリセットしてください。



# 2 仕 様

## 2.1 用 途

S10ET LINKモジュール（形式：LWE400）は、IEEE802.3仕様に準拠したローカルエリアネットワークに接続し、TCP/IPまたはUDP/IP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

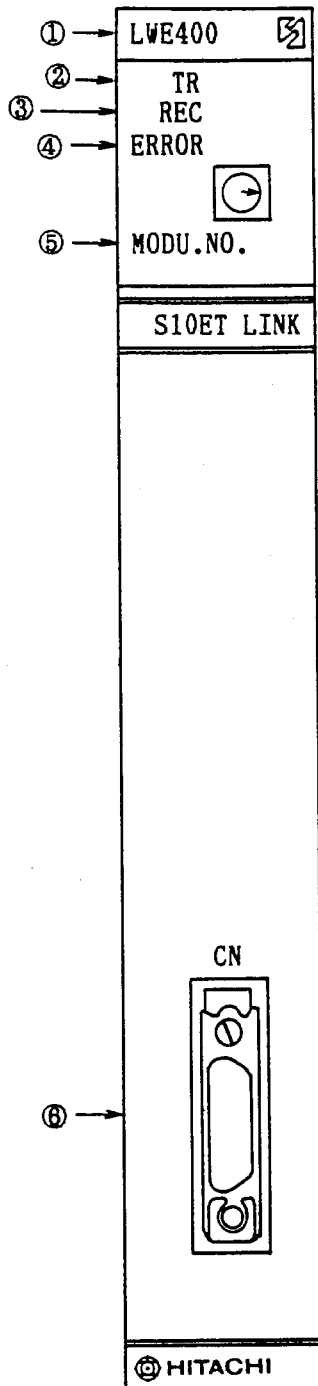
項 目		仕 様
メモリ	素 子	CMOS-RAM
	バッテリー	リチウム電池
	バックアップ期間	7年間（25℃）
S10ET LINK モジュール最大実装台数	2モジュール/CPU（2モジュール実装時は、少なくともどちらか一方は電源内蔵型トランシーバを使用してください。）	
モジュールスロット幅	1スロット幅モジュール	

### 2.2.2 回線仕様

項 目	仕 様
適 合 規 格	IEEE802.3（CSMA/CD準拠）
データ伝送速度	10Mbps
セグメント最大長（ターミネータ間距離）	500m
セグメント内 トランシーバ取付最大数	100台
ステーション間最大距離	2500m （トランシーバケーブルを除く）
システム最大ステーション数	200台
トランシーバケーブル長	標準5m，最大長15m （LANトランシーバに付属）
ステーション間経路内 リピータ最大数	2台
トランシーバ取付間隔	2.5mの整数倍

### 3 各部の名称と機能, 配線

3.1 各部の名称と機能



- ① モジュール形式  
LWE400
- ② TR (TRANSMIT)  
データ送信時、点灯します。
- ③ REC (RECEIVE)  
データ受信時、点灯します。
- ④ ERROR  
エラー発生時、点灯します。
- ⑤ モジュールNo.設定スイッチ  
MAINモジュールは“0”に設定  
します。  
SUBモジュールは“1”に設定し  
ます。  
ただし、1台のみ実装の場合は“0”  
を設定し“1”は設定しないでくだ  
さい。  
(注) 2モジュール実装時は、少なくと  
もどちらか一方は電源内蔵型トラ  
ンシーバを使用してください。
- ⑥ トランシーバケーブル用コネクタ  
トランシーバケーブルを介してトラ  
ンシーバと接続します。

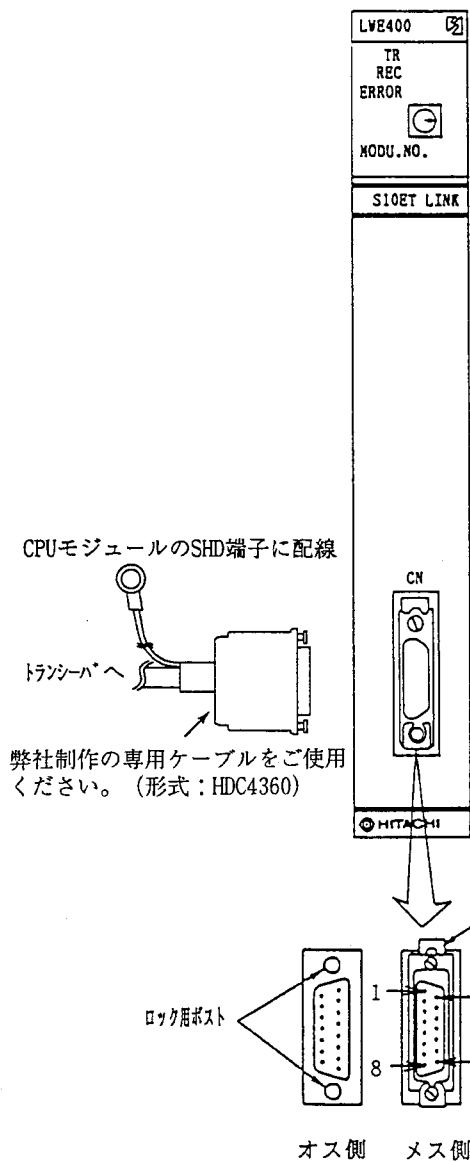
3.2 配線

**!** 強 制

- 10BASE 5のコネクタ（リテーナなど）に触れる際には、必ず人体に帯電している静電気を放電してから作業を行ってください。人体に帯電している静電気により感電、誤動作、故障の原因となります。
- トランシーバケーブルの接続は、モジュール側のロック用リテーナをスライドさせ、ケーブルのロック用ポストに完全にロックするように取付けてください。
- トランシーバケーブルは、下図のように必ずシールドアース配線のできるケーブルをご使用ください。アース用配線のないケーブルでは、ノイズなどの影響によりエラーが発生する可能性があります。

P I N No.	信 号 名	P I N No.	信 号 名
1	シールド		
2	衝突検出+	9	衝突検出-
3	送 信+	10	送 信-
4	—	11	—
5	受 信+	12	受 信-
6	電源リターン	13	電 源
7	—	14	—
8	—	15	—

コネクタ Dsub 15ピン







## 4 利用の手引き

## 4.1 システム構成

基本構成は、例1のように最大長500mの同軸ケーブルとそれに接続されるステーションからなります。ステーションは、トランシーバケーブルとトランシーバを介して同軸ケーブルに接続されます。この基本構成をセグメントといい、1セグメントのステーション数は、最大100台です。

ステーション間距離が500m以上となる場合は、例2に示すようにリピータを使用して分岐状にセグメントの数を増します。

例2は、最大ステーションの距離が1,500m以内のシステム例であり、どの2つのステーション間の経路を取っても通過するリピータの数が2台以下となるように構成します。

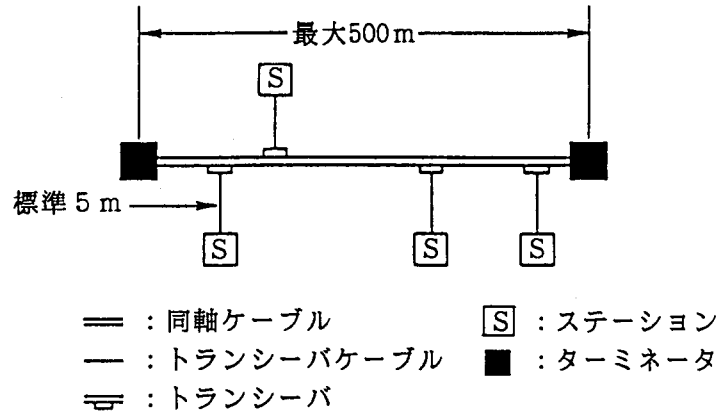
例3は、ステーション間の最大距離を2,500mとした例であり、リピータにリンクケーブル(最大500m)を付けたものを1台のリピータとして数え、リンクセグメントと呼びます。システム構成上の制限事項と留意点を以下に示します。

セグメント最大長	500m
セグメント内トランシーバ取付最大数	100台
ステーション間最大距離	2,500m以下(トランシーバケーブルを除く)
システム最大ステーション数	200台
トランシーバケーブル長	標準5m, 最大長15m
ステーション間経路内リピータ最大数	2台

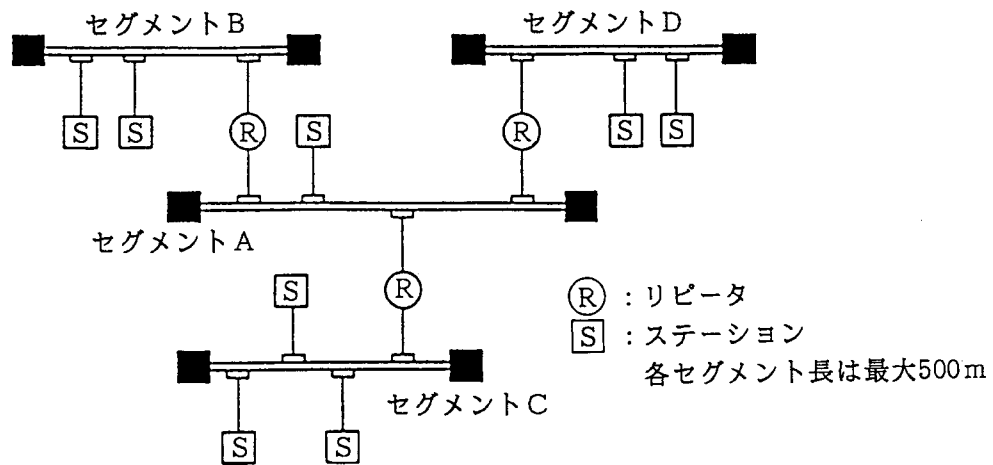
### 注 意

- リピータは、トランシーバケーブルとトランシーバを介して同軸ケーブルに接続してください。
- リピータは、同軸セグメント中のどの位置のトランシーバにも取付けられます。
- リンクケーブルに、ステーションを取付けないでください。
- トランシーバの取付間隔は、2.5mの整数倍としてください。

(例1) 最小構成 (リピータなし, セグメント長 最大500m)



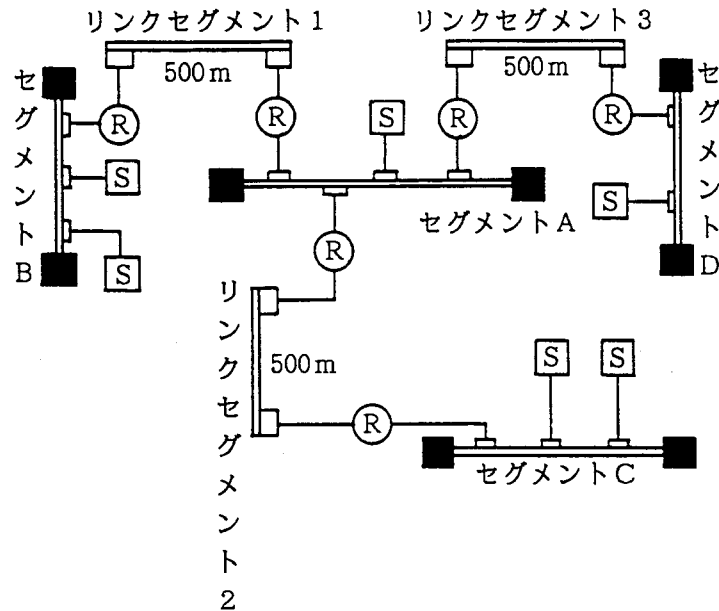
(例2) 中規模構成 (リピータ使用, トランシーバ間最大1,500m)



**注意**

- 任意のステーション間のリピータは2台以下にしてください。
- リピータを2台以上接続するセグメントは1つだけにしてください。

(例3) 大規模構成 (リピータ, リンクセグメント使用, トランシーバ間最大2,500m)



**注意**

- リンクセグメントは最大500mです。
- リンクセグメントにはステーションを取付けることはできません。
- リンクセグメントは両端のリピータを含めてリピータ1台とみなします。
- 任意のステーション間のリピータは2台以下にしてください。
- リピータを2台以上接続するセグメントは1つだけにしてください。

〈制限事項〉

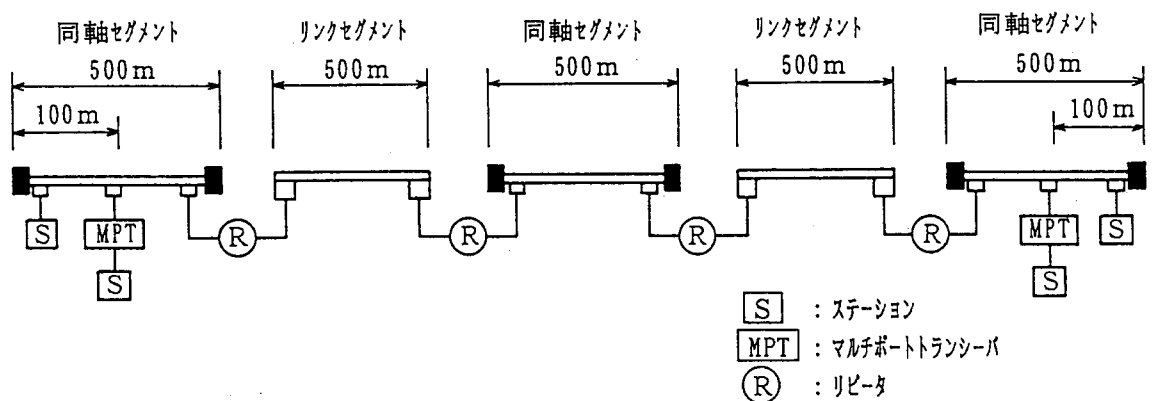
- 最大規模構成時におけるマルチポートトランシーバの設置位置の制限  
同軸ケーブル長として最大2,500m（5セグメント）で構成するシステムにおいて、最遠端の同軸ケーブルセグメント上にマルチポートトランシーバを設置する場合には、設置によりデータの遅延時間が増加するためにマルチポートトランシーバの設置位置に制限が生じます。

マルチポートトランシーバを経由したステーション間の最大距離は、マルチポートトランシーバ1台を通過することにより同軸ケーブル長に換算して100m減少します。したがって、あるステーションから他のステーションに至る経路の同軸ケーブル線長 $l$  [m]には次のような制限があります。

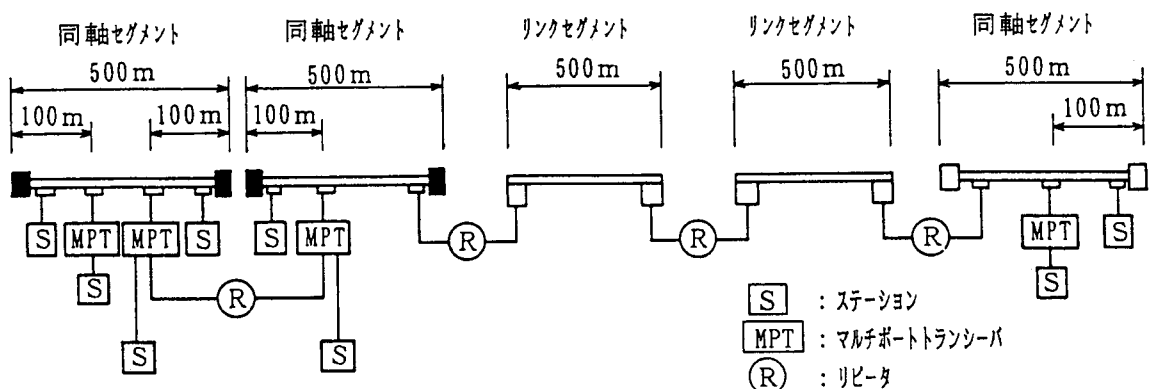
$$l \text{ [m]} \leq 2,500 \text{ [m]} - 100 \times n \text{ [m]}$$

$n$  : 経由するマルチポートトランシーバの総数

2,500mの同軸ケーブルで構成されるシステムにおいて、マルチポートトランシーバは最遠端の同軸ケーブルターミネータから100m内側（ステーション間の距離を減少させる位置）に設定してください。

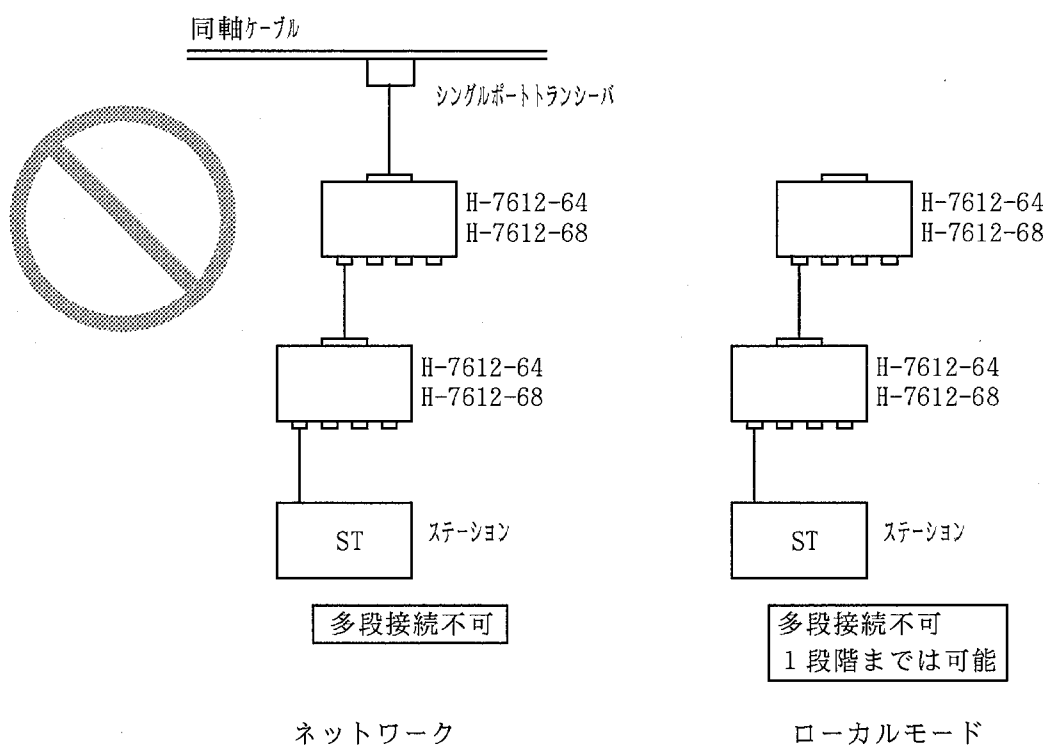


同様にマルチポートトランシーバを使用して、セグメント間のリピータを接続する場合もマルチポートトランシーバ1台を通過することにより、最遠端のステーション間の距離を100m減少させる位置にマルチポートトランシーバを設置してください。



**注意**

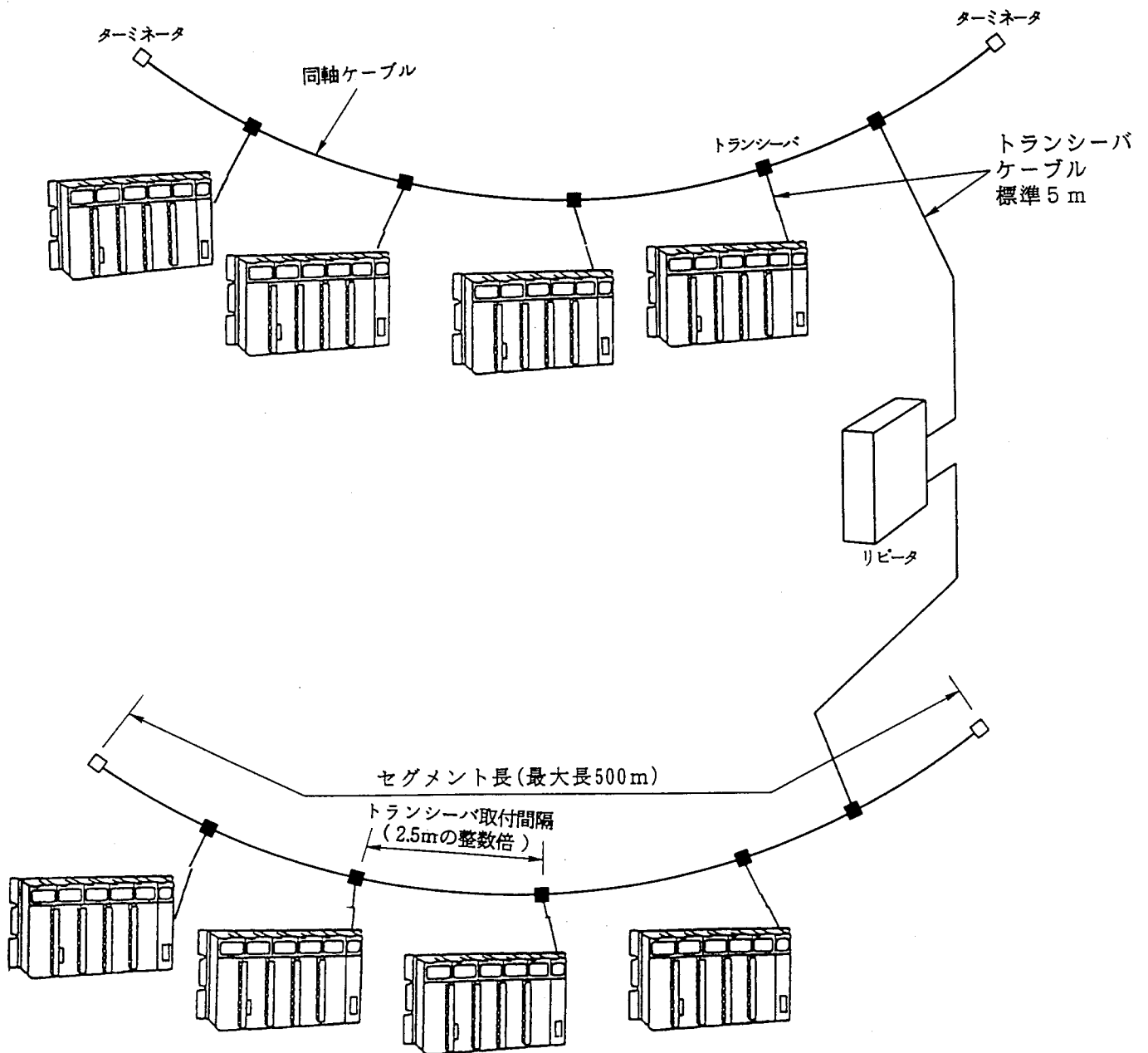
- マルチポートトランシーバ(H-7612-64/68)の多段接続の制限  
マルチポートトランシーバは、ネットワークモードで使用する場合、伝送特性上の制約から多段接続はできません。



**注意**

- シングルポートトランシーバの機種制限  
ネットワークモードにおいてはマルチポートトランシーバの上位に接続されるシングルポートトランシーバは、マルチポートトランシーバから給電されるDC12Vで動作する条件を保证するため、下記の指定機種を使用してください。
  - ・ HLT-200TB
  - ・ HLT-200

4.2 S10/2αによるシステム構成例



### 4.3 システム定義情報

各ステーションには下記情報を定義します。3つともネットワーク内で、ユニークな情報でなければなりません。

- ① 物理アドレス  
 ② IPアドレス  
 ③ ホスト名
- } S10ET LINK 1枚ごとに②, ③を定義してください。

#### 4.3.1 物理アドレス

1枚のS10ET LINKには、48ビットの物理アドレスを割り付けてあります。このアドレスは全世界に1つのユニークなアドレスであり、ROM化されています。ユーザはアドレスを変更できません。物理アドレスは16進で記述します。例を以下に示します。

08006300B001

#### 4.3.2 IPアドレス

TCP/IPとUDP/IPはIPアドレスという32ビットの論理アドレスを使用します。IPアドレスはネットワーク番号とホスト番号からなり、そのアドレスの割り付けはホストの台数によって、次の3とおりが使用できます。

- ・クラスA（ネットワーク番号の上位1ビットを0とします。）

ネットワーク番号 (8ビット)	ホスト番号 (24ビット)
--------------------	---------------

- ・クラスB（ネットワーク番号の上位2ビットを10とします。）

ネットワーク番号 (16ビット)	ホスト番号 (16ビット)
---------------------	---------------

- ・クラスC（ネットワーク番号の上位3ビットを110とします。）

ネットワーク番号 (24ビット)	ホスト番号 (8ビット)
---------------------	--------------

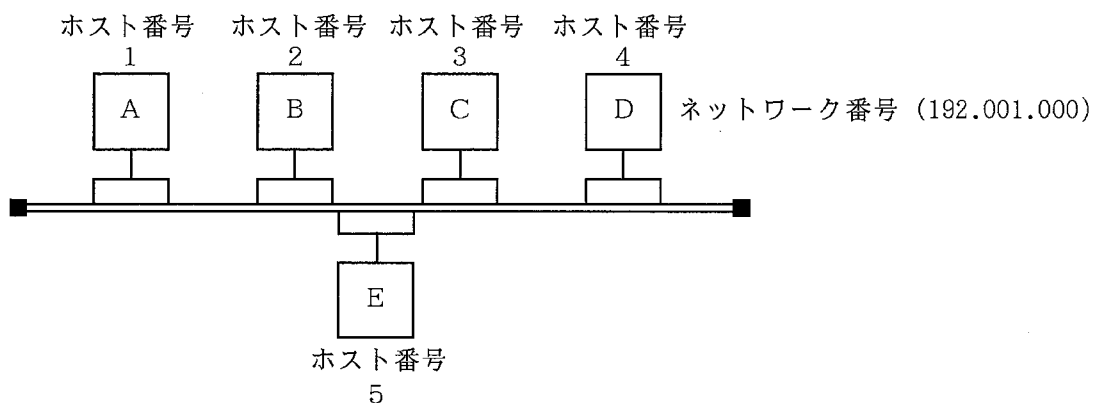
また、このアドレスは8ビットごとに“.”(ピリオド)”で区切り、10進数で表します。例えば、クラスCでは以下のように表現します。

クラスCの場合

11000000	00000001	00000000	00000001
192	.	001	.
		000	.
			001
ネットワーク番号			ホスト番号



1つのネットワークは、ネットワーク番号で決定され、ネットワーク内の各ホストにはユニークなホスト番号を定義します。したがって、同一ネットワーク内にホストが200台以内である場合は、クラスCを選択します。例えば、ネットワーク番号として(192.001.000)を選択したとします。



ステーションA, B, C, D, Eは同一ネットワークに属するため、ユニークなホスト番号として、1~5を割り当てます。したがって、それぞれのステーションのIPアドレスは以下のようになります。

ステーションA : 192.001.000.001

ステーションD : 192.001.000.004

ステーションB : 192.001.000.002

ステーションE : 192.001.000.005

ステーションC : 192.001.000.003

### 4.3.3 ホスト名

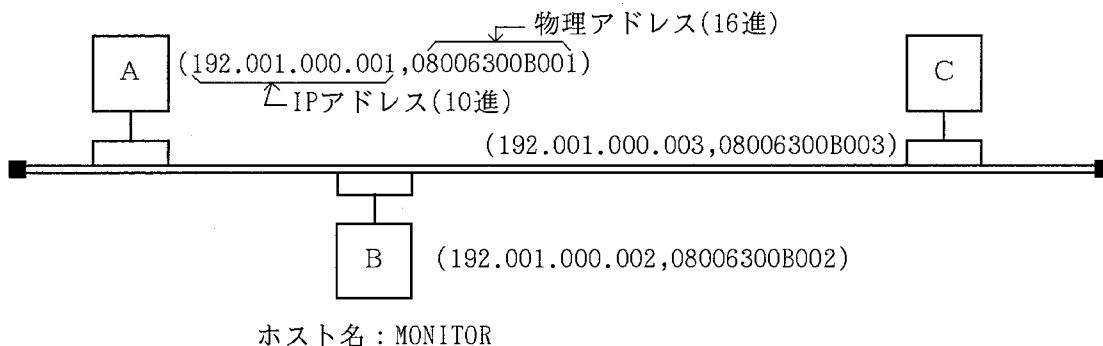
各ステーションにはホスト名と呼ばれる名前が付いています。ユーザプログラムのステーション指定には、ホスト名を使用します。ホスト名はTCP/IPプログラムまたはUDP/IPプログラムによりIPアドレス、物理アドレスに変換されます。

ホスト名はASCIIコード16桁以内で、数字、アルファベット、特殊文字を使用して定義してください。

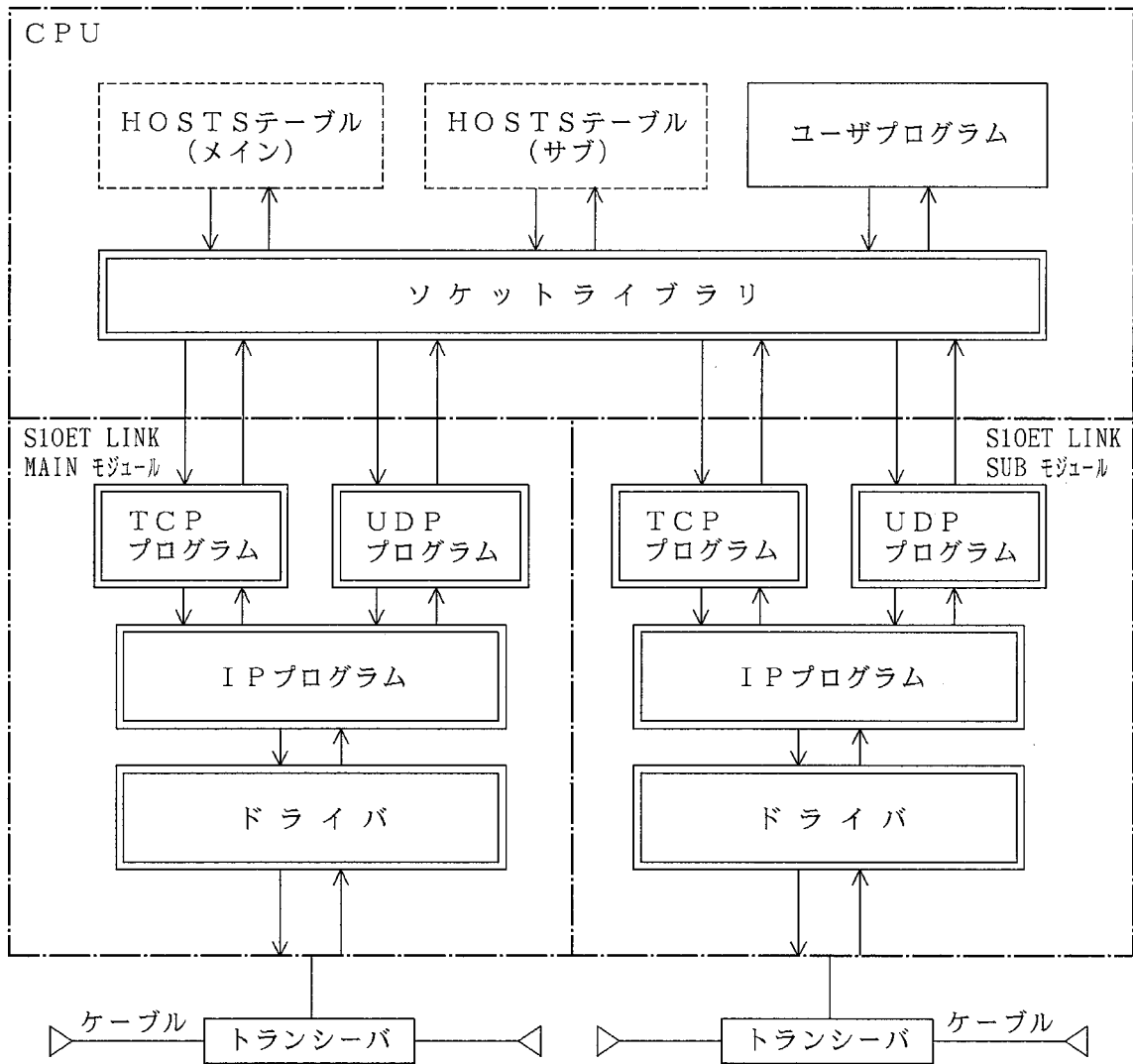
(例)

ホスト名 : ELECTRIC MOTOR

ホスト名 : BELT CONVEYOR



4.4 SIOET LINKのソフトウェア構成



- : システム提供プログラム
- : ユーザ作成テーブル
- : ユーザ作成プログラム

## 4.5 S10ET LINKに提供されるシステムプログラム

4.4節のソフトウェア構成図で示したシステム提供プログラムの説明をします。

システムプログラムは、フロッピーディスクで提供し、PSEにてローディングします。

システムプログラムは、下記5種類に大別でき、CPUまたはS10ET LINKモジュール上で動作します。

- ・ソケットライブラリ
- ・TCPプログラム
- ・UDPプログラム
- ・IPプログラム
- ・ドライバ

### 4.5.1 ソケットライブラリ

ソケットライブラリはC言語の関数として呼び出され、ユーザプログラムの代わりにS10ET LINKモジュールを制御します。ユーザはソケットライブラリを利用することにより、ハードウェア仕様ならびにプロトコルを意識することなくプログラミングできます。

### 4.5.2 TCPプログラム

上位のプロトコルとして、高信頼性のデータ送受信管理を行います。TCPプログラムの機能を以下に示します。

- 信頼性チェック
  - ・受信応答発信号 (ACK) の確認
  - ・シーケンス番号による順序チェック
  - ・データのチェックサム
- データ再送 (信頼性チェックにてエラー発生時)
- 受信可能データ量のフロー制御
- 複数プロセスの同時通信 (多重化)
- コネクションの確立による論理接続
- データのセキュリティと優先順位管理

### 4.5.3 UDPプログラム

上位のプロトコルとして、高速かつ大量のデータ送受信管理を行います。UDPプログラムの機能を以下に示します。

- コネクションレス型の通信
- 同時通信
- パケットにもとづいたデータ伝送

#### 4.5.4 IPプログラム

下位のプロトコルとして、通信路の論理的な接続を行います。IPプログラムの機能を以下に示します。

- パケットの最大長に応じたデータの分割と再組立て
- IPアドレスと物理アドレスの交換

#### 4.5.5 ドライバ

通信回路を制御し、回線（トランシーバ）へのデータ送受信を行います。ドライバの機能を以下に示します。

- 受信データのCRC（Cyclic Redundancy Check：巡回冗長検査）
- 送信時のデータ衝突検出と再送

## 4.6 ユーザの作成するプログラム、テーブル

4.5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、ユーザが作成する必要のあるソフトウェアについて説明します。

### 4.6.1 ユーザプログラム

ユーザプログラムは、ソケットライブラリを起動し、データの送信、受信を実施します。ユーザプログラムはCモードプログラムで作成し、H-S10/2αシリーズCPUにローディングします。

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（Compact Process Monitor System）と拡張メモリが必要です。

ソケットライブラリについては4.7節で説明します。

ソケットライブラリによるプログラムについては、「第5章 プログラム例」を参照してください。

### 4.6.2 HOSTSテーブル

HOSTSテーブルには、ネットワークに接続されている全ステーションの情報を設定します。Cモードプログラムで作成し、H-S10/2αシリーズCPUにローディングします。全ステーションに同一データをローディングします。

「5.1.3 HOSTSテーブル例」を参照してください。

HOSTSテーブル (メイン)

アドレス	ホスト名	IPアドレス
160000	MOTOR 1	192.001.000.001
14	MOTOR 2	192.001.000.002
28	MOTOR 3	192.001.000.003
3C	MOTOR 4	192.001.000.004
⋮	⋮	⋮
160F8C	MOTOR 200	192.001.000.200
160FA0	0———0	0———0

有効データ  
(最大200件まで)

← 終了コード

HOSTSテーブル (サブ)

アドレス	ホスト名	IPアドレス
161000	PUMP 1	192.001.001.001
14	PUMP 2	192.001.001.002
28	PUMP 3	192.001.001.003
3C	PUMP 4	192.001.001.004
⋮	⋮	⋮
161F8C	PUMP 200	192.001.001.200
161FA0	0———0	0———0

有効データ  
(最大200件まで)

← 終了コード

- ・ テーブルの基本構造を以下に示します。20バイト (キャラクタ) で構成されます。

```

struct u_host {
    char name [16] ; /* ホスト名      */
    char addr [ 4] ; /* IPアドレス */
};

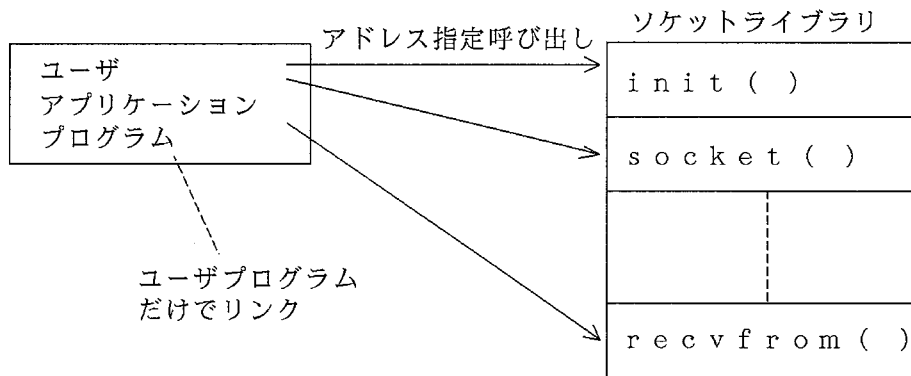
```

- ・ 設定件数は1～200件です。
- ・ 最後尾に終了コード (20バイトですべて“0”) を設定してください。

4.7 ソケットライブラリ

ソケットライブラリはC言語の関数として呼び出され、ユーザプログラムの代わりにS10ET LINKモジュールを制御し、データの送受信を実施します。ソケットライブラリは11種類の関数群で構成されます。これらはH-S10/2αシリーズCPUおよび、S10ET LINKモジュールにシステムプログラムをローディングすることにより、ユーザプログラムから呼び出せます。

ソケットライブラリは、アドレス指定で呼び出してください。ユーザプログラムは、ソケットライブラリを含めた形では作成（リンク）できません。





## 4.7.1 ソケットライブラリー一覧

以下にソケットライブラリー一覧を示します。

名 称	アドレス	機 能	対応プログラム
init ( )	164000	モジュールの初期化	TCP/IP および UDP/IP
socket ( )	16400C	ソケットを登録します。	TCP/IP または UDP/IP
gethostname ( )	164012	現在の自局情報を読出します。	TCP/IP および UDP/IP
getpeername ( )	164018	ソケットIDにより結ばれている相手局情報 を読出します。	TCP/IP
accept ( )	16401E	listen( )関数により受動状態になったソケ ットに対しコネクション要求を受付けます。	TCP/IP
connect ( )	164024	受動状態で待っている相手局に対してコネク ションを確立します。	TCP/IP
listen ( )	16402A	ソケットを受動状態にします。	TCP/IP
shutdown ( )	164030	ソケットで確立されたコネクションを終了し ます。	TCP/IP および UDP/IP
send ( )	164036	ソケットを使ってデータを送信します。	TCP/IP
sendto ( )	16403C	ソケットを使ってデータを送信します。	UDP/IP
recv ( )	164042	ソケットからデータを受信します。	TCP/IP
recvfrom ( )	164048	ソケットからデータを受信します。	UDP/IP

(注) `shutdown ( )` の終了条件は、“送受信を許可しない”のみサポートしています。

— 制限 —

- 1つのモジュールで、同時に使用できるソケット数は10個です。
- 0～9999のポート番号はシステムが占有していますので、ユーザは10000～65535を使用してください。
- データ送受信のデータ長は、1回の関数発行で1～4096バイトです。
- 相手局未指定の場合、1つの自局ポート番号に2つ以上のソケットを割り当てることができます。その場合、`listen ( )` を発行した順番に、相手の`connect ( )` を受け付けます。
- 相手局指定の場合、1つのコネクション（自局ポート番号、相手局ホスト名、相手局ポート番号で決まります。）に2つ以上のソケットを割り当てることはできません。
- TCP/IPプログラムを選択したソケットに対して`sendto ( )` , `recvfrom ( )` は発行できません。間違えて発行した場合、リターン値、`*err - p`でエラーを返します。
- UDP/IPプログラムを選択したソケットに対して`accept ( )` , `connect ( )` , `listen ( )` , `send ( )` , `recv ( )` , `getpeername ( )` は発行できません。間違えて発行した場合、リターン値、`*err - p`でエラーを返します。

— 用語 —

着信 : TCP/IPプログラムまたはUDP/IPプログラムが、データを受信したことを意味します。

モジュール : TCP/IPプログラムおよびUDP/IPプログラムがローディングされているハードウェアを意味します。

— タスクの強制終了 —

ソケットライブラリを利用しているタスクが強制終了されると、ソケットが有効のまま残ってしまいます。（そのタスクが自分で使用しているソケットを、`shutdown ( )` した後ならばこの限りではありません。）

つまり、タスクが強制終了されたときのソケットの状態が、タスクが終了したにもかかわらず残ってしまうことです。以下、そういう状態のソケットを、「浮いたソケット」と呼ぶことにします。

浮いたソケットは、他のタスクで使用できません。

---

したがって、浮いたソケットまたはモジュールに対し、下記1～3のいずれかの処理を行ってください。

1. 他のタスクまたは組込みサブルーチンから、浮いたソケットを `shutdown ( )` する。
2. CPUをリセットする。
3. 電源を一度遮断し、復電する。

init ( )

機能

この関数は、モジュールにTCP/IPプログラムおよびUDP/IPプログラムの起動要求を発行し、起動したTCP/IPプログラムおよびUDP/IPプログラムに対して、初期設定データを与えて初期処理実行を要求します。処理結果は、リターン値として返ります。

使用するモジュールに対して、最初に1回のみ実行してください。

リンク手順

C 言語

```

    {
long (*init)( );
long *err_p;
long lanpn, rtn;
    }
init=0x164000;
    {
rtn =(*init)(err_p,lanpn);
    }

```

パラメータ

〔入力パラメータ詳細〕

lanpn : モジュールの指定  
 1 MAINモジュール  
 2 SUBモジュール

〔出力パラメータ詳細〕

リターン値 : 初期化の結果  
 (0) 正常終了 (−1) 異常終了  
 \*err\_p : 異常終了時にエラーコードが返ります(正常終了時は“ゼロ”  
 クリア)。エラーコードには下記があります。  
 エラーコードの詳細は9.4節を参照してください。

```

EOPNOTSUPP , EINIT
ENOTTERM   , EILLBOARD
ESYSTEM    , EILLPARAM
ERSPTIMEOUT , ENOTFILE
EILLINIT   , EUNKWN

```

socket ( )

機能

この関数はTCP/IPプログラムまたはUDP/IPプログラムのソケットの登録を行う関数で、リターン値には登録されたソケットのIDを返します。ホスト名の文字列の長さが0の場合、相手局ポート番号は、強制的にNULLと見なされます。

TCP/IPプログラムでconnect ( )を使用する場合は、相手局指定（相手ポート番号、相手局ホスト名を設定）でソケット登録をしてください。

TCP/IPプログラムでlisten ( )と、accept ( )を使用する場合は、相手局指定、相手局未指定のどちらかでソケット登録をしてください。

UDP/IPプログラムを選択する場合、相手局指定でソケット登録をしてください。ただし、recvfrom ( )を使用する場合および、相手局IPアドレス (dst\_ip)、相手局ポート番号 (dat\_port) 設定で、sendto ( )を使用する場合は、相手局未指定でソケット登録できます。

リンク手順

C 言語

```

long    (*socket)( );
struct  sockaddr *saddr;
char    *name;
long    name_len;
long    *err_p;
long    lanpn, rtn;
    }
socket=0x16400c;
    }
rtn =(*socket)(saddr, name, name_len, err_p, lanpn);
    }

```

パラメータ

〔入力パラメータ詳細〕

\*saddr : ソケット情報へのポインタ。

saddr->sa\_family : ソケットタイプ

1 TCP/IPプログラムを  
選択

2 UDP/IPプログラムを  
選択

saddr->sa\_port : 自局ポート番号

saddr->dst\_port : 相手ポート番号

\* name : 相手局ホスト名の文字列へのポインタ。  
name\_len : ホスト名の文字列の長さ  
          0 ..... 相手局未指定  
          1 ~ 16 ..... 相手局指定  
lanpn : モジュールの指定  
      1 MAINモジュール  
      2 SUBモジュール

[出力パラメータ詳細]

リターン値 : 登録されたソケットIDが返ります。

(-1) 異常終了

\* err\_p : 異常終了時にエラーコードが返ります (正常終了時は“ゼロ”  
クリア)。エラーコードには下記があります。

エラーコードの詳細は9.4節を参照してください。

ENOBUFS , EOPNOTSUPP  
ESOCKALRDY , EHOSTS  
EILLPORT , ENOTINIT  
EILLPARAM , ENOTFILE  
EUDPALRDY

```
gethostname ( )
```

## 機能

この関数は、TCP/IPプログラムおよびUDP/IPプログラムで自局のホスト名、IPアドレス、物理アドレスを返す関数で、リターン値として処理結果を返します。

## リンク手順

## C 言語

```
long    (*gethostname)( );
struct  host_type *name;
long    *name_len;
long    *err_p;
long    lanpn, rtn;
    {
gethostname=0x164012;
    }
rtn =(*gethostname)(name, name_len, err_p, lanpn);
    }
```

## パラメータ

[入力パラメータ詳細]

lanpn : モジュールの指定  
 1 MAINモジュール  
 2 SUBモジュール

[出力パラメータ詳細]

\*name : ホストの状態情報  
    name->host\_name : ホスト名  
    name->ip\_addr : IPアドレス  
    name->e\_addr : 物理アドレス

\*name\_len : ホストの状態情報の長さ。(バイト数)

リターン値 : 処理結果  
            (0) 正常終了      (-1) 異常終了

\*err\_p : 異常終了時にエラーコードが返ります(正常終了時は  
          “ゼロ”クリア)。エラーコードには下記があります。  
          エラーコードの詳細は9.4節を参照してください。

EOPNOTSUPP , ENOTINIT  
EILLBOARD , ESYSTEM  
ENOTFILE , EUNKWN



```
getpeername ( )
```

### 機能

この関数は、TCP/IPプログラムで入力パラメータのソケットIDにより結ばれている相手局のホスト名、IPアドレス、ポート番号と、自局のソケットID、ポート番号、TCPローカルコネクション名を読み出す関数で、処理結果をリターン値として返します。

TCPローカルコネクション名とは、TCP/IPプログラムとライブラリの間で使用されるデータなので、ユーザはこのデータを変更しないでください。

### リンク手順

#### C 言語

```
long    (*getpeername)( );
long    s;
struct  socktbl *name;
long    *name_len, *err_p;
long    lanpn, rtn;
        }
getpeername=0x164018;
        }
rtn = (*getpeername)(s, name, name_len, err_p, lanpn);
        }
```

### パラメータ

[入力パラメータ詳細]

s : ソケットID

lanpn : モジュールの指定

- 1 MAINモジュール
- 2 SUBモジュール

## 〔出力パラメータ詳細〕

```
*name          : 相手局情報
  name->id      : ソケットID
  name->lcn     : TCPローカルコネクションネーム
  name->rmt_name : 相手局ホスト名
  name->local_prt : 自局ポート番号
  name->remote_prt : 相手局ポート番号
  name->remote_ip : 相手局IPアドレス
*name_len      : 相手局情報の長さ。(バイト数)
リターン値    : 処理結果
                (0) 正常終了   (-1) 異常終了
*err_p        : 異常終了時にエラーコードが返ります(正常終了時は
                “ゼロ”クリア)。エラーコードには下記があります。
                エラーコードの詳細は9.4節を参照してください。
EOPNOTSUPP    , ENOTCONN
ENOTSOCK      , ENOTINIT
EILLBOARD    , ESYSTEM
EILLPARAM    , ERSPTIMEOUT
ENOTFILE     , EUNKWN
```

accept ( )

機能

この関数は、TCP/IPプログラムでlisten ( )により受動状態になったソケットに対するコネクション要求を待ち、コネクション要求を受ける関数で、リターン値に確立後のソケットIDを返します。

パラメータで指定されたソケットIDと、リターン値で返すソケットIDは同じ値です。

確立後のソケットIDテーブル情報は、相手局指定のソケットのときには、テーブルの全情報を返します。また、相手局未指定のソケットのときには、相手局の情報は0を返します。

リンク手順

C 言語

```

long    (*accept)( );
long    s ;
struct  socktbl *addr ;
long    *addr_len, *err_p ;
long    lanpn, rtn ;
        }
accept=0x16401E ;
        }
rtn =(*accept)(s, addr, addr_len, err_p, lanpn) ;
        }

```

パラメータ

〔入力パラメータ詳細〕

s : ソケットID

lanpn : モジュールの指定

1 MAINモジュール

2 SUBモジュール

## 〔出力パラメータ詳細〕

- \* `addr` : 確立後のソケットIDテーブル  
テーブルの型は「4.7.2 パラメータのデータ構造」  
を参照してください。
- \* `addr_len` : テーブルサイズ (バイト数)
- リターン値 : コネクション確立による新しいソケットID  
(-1) 異常終了
- \* `err_p` : 異常終了時にエラーコードが返ります (正常終了時は  
“ゼロ”クリア)。エラーコードには下記があります。  
エラーコードの詳細は9.4節を参照してください。
- |                          |                           |
|--------------------------|---------------------------|
| <code>EILLFNC</code>     | , <code>EOPNOTSUPP</code> |
| <code>ECONNALRDY</code>  | , <code>ETIMEOUT</code>   |
| <code>ECONNFAIL</code>   | , <code>ENOTSOCK</code>   |
| <code>ENOTINIT</code>    | , <code>EILLBOARD</code>  |
| <code>ESYSTEM</code>     | , <code>EILLPARAM</code>  |
| <code>ERSPTIMEOUT</code> | , <code>ECHKSUM</code>    |
| <code>EICMP</code>       | , <code>EADDR</code>      |
| <code>ENOTFILE</code>    | , <code>EUNKWN</code>     |
| <code>ENETID</code>      | , <code>EBROADCAST</code> |

connect ( )

機能

この関数は、TCP/IPプログラムで受動状態で待っている相手局に対してコネクション要求を発行する関数で、リターン値で処理結果を返します。

コネクション要求を発行する前に、相手局ではアクセプト関数を実行している（受動状態となる）必要があります。相手局が未受動状態のときにコネクション要求を発行するとエラーとなります。

リンク手順

C 言語

```

long    (*connect)( );
long    s ;
struct  skidtbl *name ;
long    *name_len, *err_p ;
long    lanpn, rtn ;
        }
connect=0x164024 ;
        }
rtn =(*connect)(s, name, name_len, err_p, lanpn) ;
        }

```

パラメータ

〔入力パラメータ詳細〕

s                   : ソケットID  
lanpn               : モジュールの指定  
                    1 MAINモジュール  
                    2 SUBモジュール

## 〔出力パラメータ詳細〕

- \*name : コネクションを確立したソケット情報。  
テーブルの型は「4.7.2 パラメータのデータ構造」  
を参照してください。
- \*name\_\_len : ソケット情報の長さ。(バイト数)  
リターン値 : 終了処理結果を返します。  
(0) 正常終了 (-1) 異常終了
- \*err\_\_p : 異常終了時にエラーコードが返ります(正常終了時は  
“ゼロ”クリア)。エラーコードには下記があります。  
エラーコードの詳細は9.4節を参照してください。
- |            |               |
|------------|---------------|
| EILLFNC    | , ENOBUFS     |
| EOPNOTSUPP | , ECONNALRDY  |
| ETIMEOUT   | , ECONNFAIL   |
| ENOTSOCK   | , ENOTINIT    |
| EILLBOARD  | , ESYSTEM     |
| EILLPARAM  | , ERSPTIMEOUT |
| ECHKSUM    | , EICMP       |
| EADDR      | , ENOTFILE    |
| EUNREMOTE  | , EUNKWN      |
| ENETID     | , EBROADCAST  |

listen ( )

## 機能

この関数は、TCP/IPプログラムでパラメータのsのソケットを受動状態にさせる関数で、リターン値に処理結果を返します。

## リンク手順

## C 言語

```
long (*listen)( );
long s, *err_p;
long lanpn, rtn;
}
listen=0x16402A;
}
rtn =(*listen)(s, err_p, lanpn);
}
```

## パラメータ

## 〔入力パラメータ詳細〕

s : ソケットID  
lanpn : モジュールの指定  
1 MAINモジュール  
2 SUBモジュール

## 〔出力パラメータ詳細〕

リターン値 : 処理結果  
(0) 正常終了 (-1) 異常終了  
\*err\_p : 異常終了時にエラーコードが返ります(正常終了時は“ゼロ”  
クリア)。エラーコードには下記があります。

エラーコードの詳細は9.4節を参照してください。

```
EILLFNC , ENOBUFS
EOPNOTSUPP , ECONNALRDY
ENOTINIT , EILLBOARD
ESYSTEM , EILLPARAM
ENOTFILE , EUNKWN
ENETID , EBROADCAST
```

shutdown ( )

機能

この関数は、TCP/IPプログラムおよびUDP/IPプログラムで、パラメータのsのソケットで確立された接続を終了させソケットを無効にする関数で、リターン値に処理結果を返します。

リンク手順

C 言語

```
long (*shutdown)( );
long s, how, *err_p;
long lanpn, rtn;
    }
shutdown=0x164030;
    }
rtn =(*shutdown)(s, how, err_p, lanpn);
    }
```

パラメータ

[入力パラメータ詳細]

s : ソケットID  
how : 1 (固定) …………… 送受信を許可しない。  
lanpn : モジュールの指定  
1 MAINモジュール  
2 SUBモジュール

[出力パラメータ詳細]

リターン値 : 処理結果  
(0) 正常終了 (−1) 異常終了  
\*err\_p : 異常終了時にエラーコードが返ります (正常終了時は“ゼロ”  
クリア)。エラーコードには下記があります。  
エラーコードの詳細は9.4節を参照してください。  
EOPNOTSUPP , ENOTSOCK  
ENOTINIT , EILLBOARD  
ESYSTEM , EILLPARAM  
ERSPTIMEOUT , ENOTFILE



send ( )

機能

この関数は、TCP/IPプログラムでパラメータのsのソケットのコネクションを使って、msgで示すデータをlenで示すバイト数分相手局に送信します。

相手局のプログラムが、recv ( ) したかどうかにかかわらず相手局にデータが正常に着信すればリターンします。

リンク手順

C 言語

```

long (*send)( );
long s ;
long *msg ;
long len, *err_p ;
long lanpn, rtn ;
    }
send = 0x164036 ;
    }
rtn = (*send)(s, msg, len, err_p, lanpn) ;
    }

```

パラメータ

[入力パラメータ詳細]

s : ソケットID

\*msg : 送信するデータへのポインタ。

len : 送信するデータサイズ。(バイト数)

lanpn : モジュールの指定

1 MAINモジュール

2 SUBモジュール

〔出力パラメータ詳細〕

リターン値 : 実際に送信したデータサイズ  
(-1) 異常終了

\*err\_p : 異常終了時にエラーコードが返ります (正常終了時は“ゼロ”  
クリア)。エラーコードには下記があります。

エラーコードの詳細は9.4節を参照してください。

ENOBUFS	, EOPNOTSUPP
ETIMEOUT	, ENOTCONN
ENOTSOCK	, ENOTINIT
EILLBOARD	, ESYSTEM
EILLPARAM	, ERSPTIMEOUT
ECHKSUM	, EICMP
EADDR	, ENOTFILE
ETRANS	, ECONNSHUT
EUNKWN	

sendto ( )

機能

この関数は、UDP/IPプログラムでパラメータのsのソケットを使って、msgで示すデータをlenで示すバイト数分相手局に送信します。

同時通信で送信する場合、相手局IPアドレスのホスト番号をすべて1に設定してください。

リンク手順

C 言語

```
long (*sendto)( );
long s;
long *msg;
long len, *err_p;
long lanpn, rtn, dst_ip, dat_port;
}
sendto=0x16403C;
}
rtn =(*sendto)(s, msg, len, err_p, lanpn, dst_ip, dat_port);
}
```

パラメータ

〔入力パラメータ詳細〕

s	:	ソケットID	
*msg	:	送信するデータへのポインタ。	
len	:	送信するデータサイズ。(バイト数)	
lanpn	:	モジュールの指定	
		1 MAINモジュール	
		2 SUBモジュール	
dst_ip	:	相手局IPアドレス	この入力パラメータは、個々に省略 (ゼロ設定)できます。
dat_port	:	相手局ポート番号	

省略時、socket ( )で指定したパラメータが有効となります。パラメータの設定について、sendto ( )とsocket ( )の関係を示します。

sendto ( )		socket ( )	
		相手局指定	相手局未指定
dst_ip (相手局IPアドレス)	設定	sendto ( ) 有効	sendto ( ) 有効
	省略	socket ( ) 有効	入力パラメータエラー
dat_port (相手局ポート番号)	設定	sendto ( ) 有効	sendto ( ) 有効
	省略	socket ( ) 有効	入力パラメータエラー

## 〔出力パラメータ詳細〕

リターン値 : 実際に送信したデータサイズ

(-1) 異常終了

\*err\_p: 異常終了時にエラーコードを返します (正常終了時は“ゼロ”クリア)。エラーコードには下記があります。

エラーコードの詳細は9.4節を参照してください。

```

ENOBUFS          , EOPNOTSUPP
ENOTSOCK         , ENOTINIT
EILLBOARD        , ESYSTEM
EILLPARAM        , EICMP
ECHKSUM          , ENOTFILE
EADDR
ETRANS           , EILLPORT
EUNKWN          , ENETID

```

recv ( )

機能

この関数は、TCP/IPプログラムでパラメータで渡されたsのソケットからデータをbufの示す領域へlenで示されたバイト数分受信する関数です。

受信待ち時間の指定は、データが全く着信していない場合のみ有効となります。受信待ち時間の単位はミリ秒です。指令された受信待ち時間が0で、受信データがない場合はただちにリターンします。

相手局で、1回のsend ( )を使用して送信したデータも、自局では1回のrecv ( )で、受信できるとは限りません。1回で受信できない場合は、再度recv ( )を発行してください。

リンク手順

C 言語

```

long  (*recv)( );
long  s ;
char  *buf ;
long  len, *err_p ;
long  lanpn, tim, rtn ;
    }
recv = 0x164042 ;
    }
rtn = (*recv)(s, buf, len, err_p, lanpn, tim) ;
    }

```

パラメータ

〔入力パラメータ詳細〕

s : ソケットID

\* buf : 受信データを格納するバッファへのポインタ。

len : 受信バッファサイズ (バイト数)

lanpn : モジュールの指定

1 MAINモジュール

2 SUBモジュール

tim : 受信待ち時間 (ミリ秒)

[出力パラメータ詳細]

- リターン値 : 実際に受信したデータ長。  
 (0) 受信データなしまたは、クローズ中  
 (-1) 異常終了
- \*err\_p : クローズ時にはENOTCONNコードが、異常終了時にはエラーコードが返ります(正常時は“ゼロ”クリア)。エラーコードには下記があります。

エラーコードの詳細は9.4節を参照してください。

EOPNOTSUPP, ENOTCONN , ENOTSOCK  
 ENOTINIT , EILLBOARD , ESYSTEM  
 EILLPARAM , ERSPTIMEOUT, ECHKSUM  
 EICMP , EADDR , ENOTFILE  
 ECONNSHUT , EUNKWN

リターン値が0で、\*err\_pがENOTCONNの場合、相手局がshutdownを発行(クローズ中)したことを意味します。

リターン値	*err_p	意味
>0	0	正常受信
=0	0	受信データなし
=0	ENOTCONN	クローズ中
-1	エラーコード	異常終了

recvfrom ( )

#### 機能

この関数は、UDP/IPプログラムでパラメータで渡された *s* のソケットからデータを *buf* の示す領域へ受信します。

受信待ち時間の指定は、データが全く着信していない場合のみ有効です。受信待ち時間の単位はミリ秒です。指令された受信待ち時間がゼロで、受信データがない場合はただちにリターンします。

UDP/IPプログラムの送受信は、パケットごとに行いますのでデータ受信サイズの指定はありません。

#### リンク手順

C 言語	
long	(*recvfrom)( );
long	s ;
char	*buf ;
long	*err_p ;
long	lanpn, tim, rtn ;
	}
	recvfrom=0x164048 ;
	}
	rtn =(*recvfrom)(s, buf, err_p, lanpn, tim);
	}

#### パラメータ

##### [入力パラメータ詳細]

*s* : ソケット ID

\* *buf* : 受信データを格納するバッファへのポインタ。

*lanpn* : モジュールの指定

1 MAINモジュール

2 SUBモジュール

*tim* : 受信待ち時間 (ミリ秒)

[出力パラメータ詳細]

リターン値 : 実際に受信したデータ長。

(0) 受信データなし

(-1) 異常終了

\*err\_P : 異常終了時にエラーコードが返ります (正常時は“ゼロ”クリア)。

エラーコードには下記があります。

エラーコードの詳細は9.4節を参照してください。

EOPNOTSUPP, ENOTSOCK , ENOTINIT  
 EILLBOARD , ESYSTEM , EILLPARAM  
 ECHKSUM , EICMP , EADDR  
 ENOTFILE , EUNKWN , EILLPORT  
 ENOBUFS , EALOCALRDY

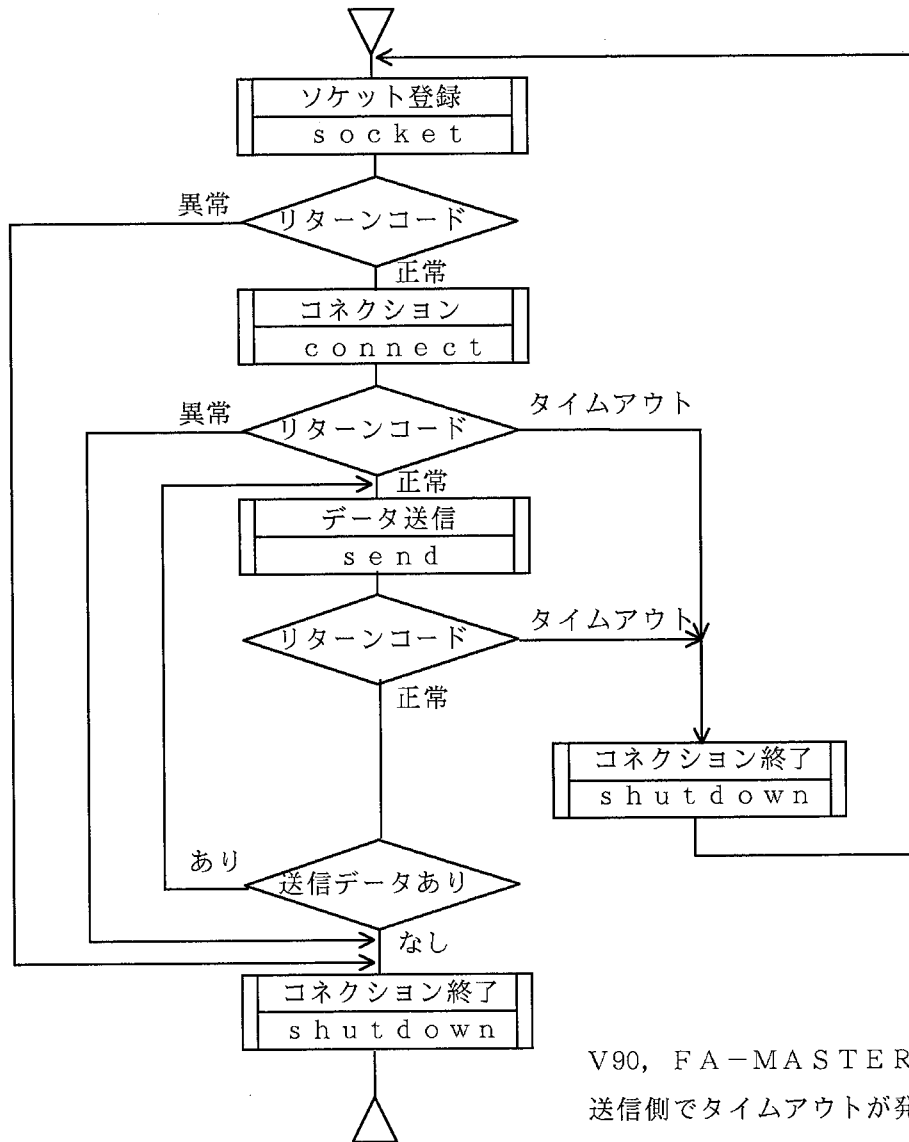
リターン値	*err_P	意 味
>0	0	正常受信
=0	0	受信データなし
-1	エラーコード	異常終了





4.8 アプリケーション作成上の注意

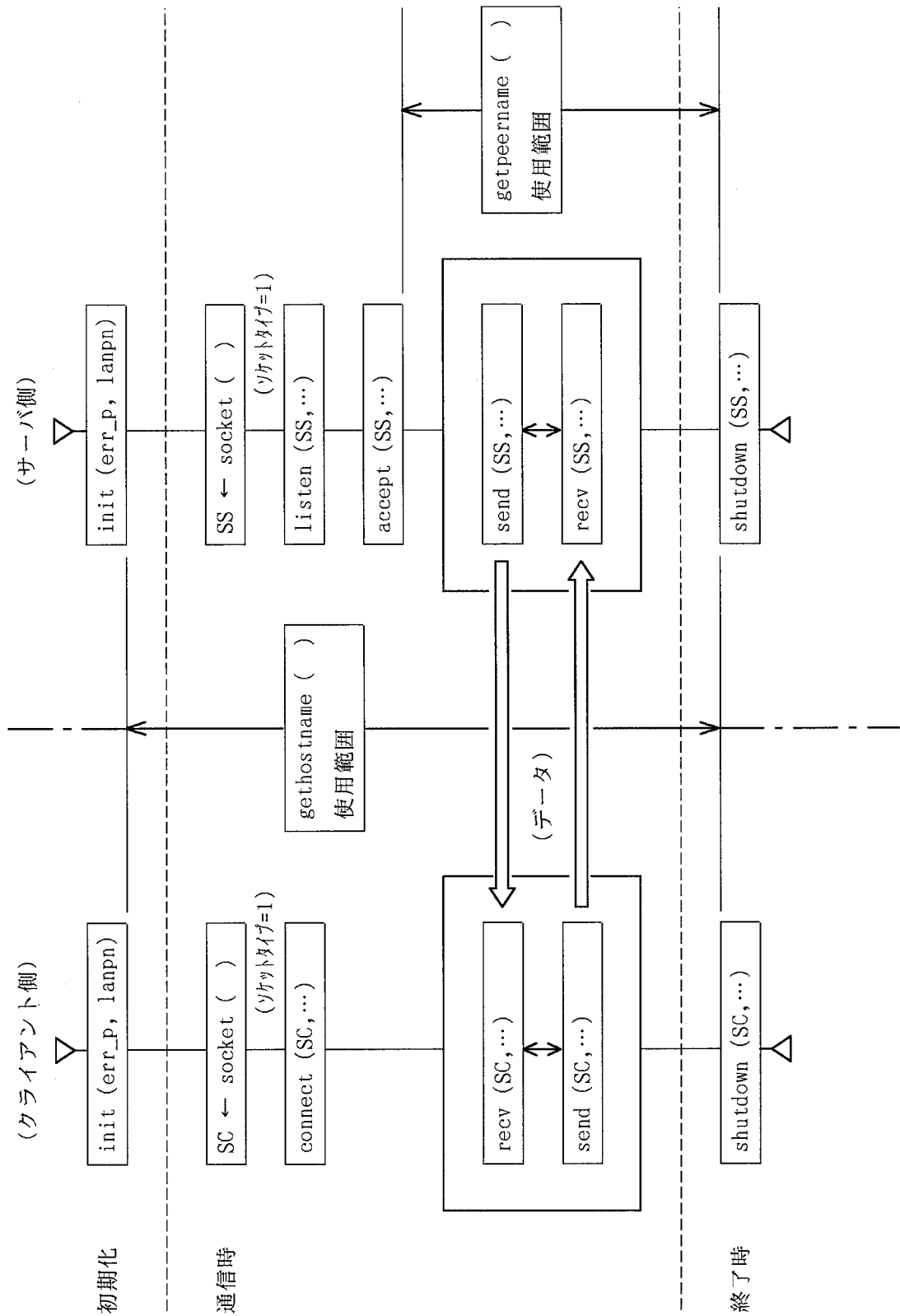
S10αとS10α(または他の装置)との通信において、送信側でタイムアウトのエラーが発生した場合には、下図のように一旦、shutdownを行い、socketから再接続してください。sendのリトライでは通信が回復しないことがあります。



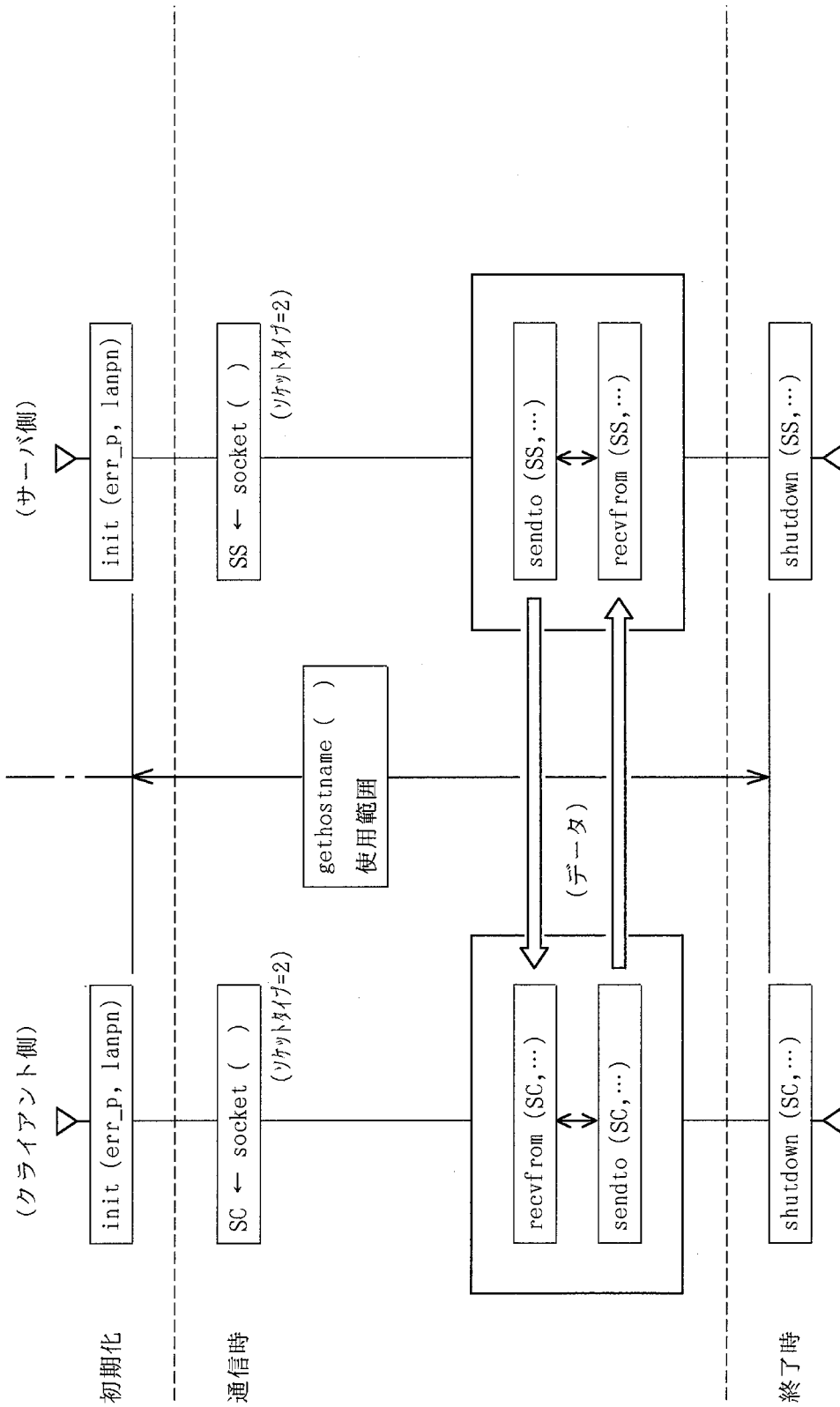
V90, FA-MASTERなど送信側でタイムアウトが発生した場合、shutdownを発行してください。

4.9 ソケットライブラリ発行手順例

4.9.1 TCP/IPプログラム使用例



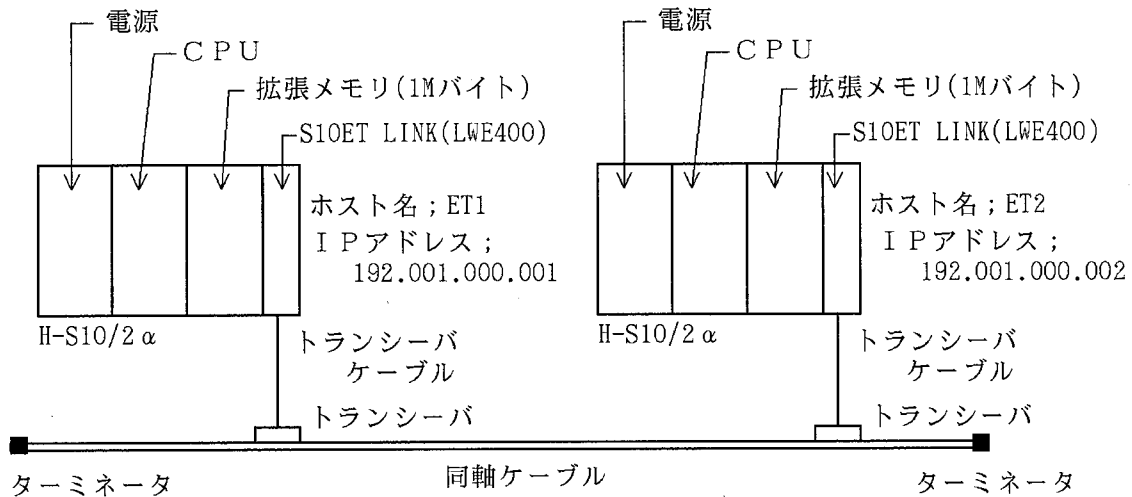
4.9.2 UDP/IPプログラム使用例



## 5 プログラム例

5.1 ソケットライブラリによるCPU間通信プログラム例

5.1.1 システム構成

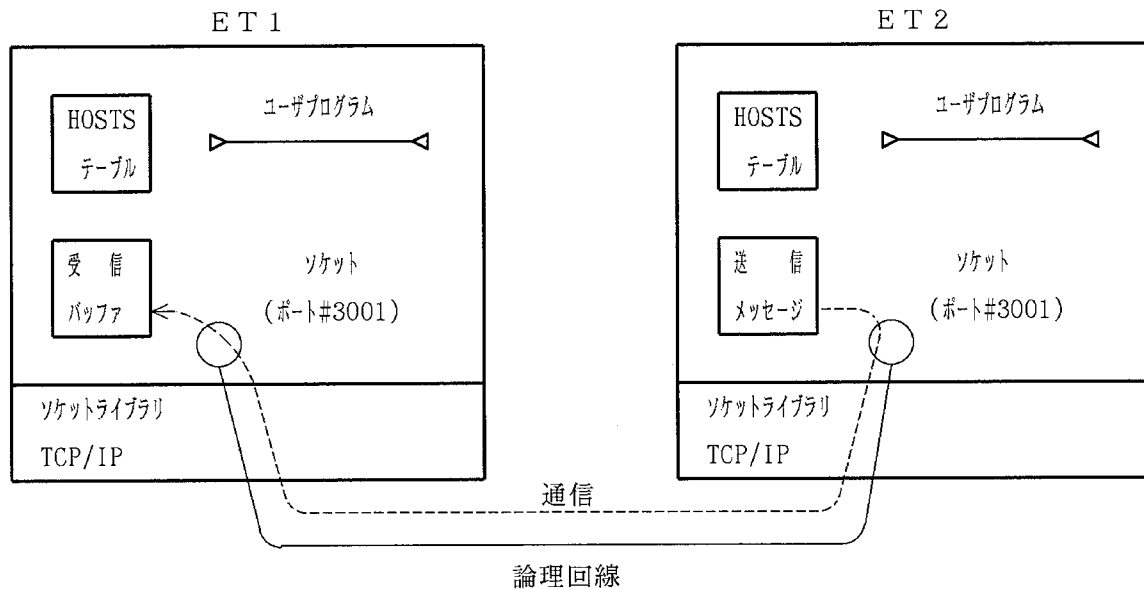


システム構成品一覧

品名	形式	数量	備考
電源	LWV000	2	
CPU	LWP000	2	
拡張メモリ	LWM004	2	
S10ET LINK	LWE400	2	
トランシーバケーブル	HBN-TC-100	2	メーカー; 日立電線(株)
トランシーバ	HLT-200TB	2	メーカー; 日立電線(株)
同軸ケーブル	HBN-CX-100	1	メーカー; 日立電線(株)
ターミネータ	HBN-T-NJ	2	メーカー; 日立電線(株)

## 5.1.2 プログラム構成

プログラムの構成を以下に示します。ET1とET2を論理回線で接続し、ET1はET2から1024バイトのデータを受信します。



項目		ホスト名	
		ET1	ET2
機能		受信	送信
送信メッセージ		—————	アドレス ; 0x1E6000 バイト数 ; 1024
受信バッファ		アドレス ; 0x1E6000 バイト数 ; 1024	—————
ポート番号		3001	3001
HOSTSテーブル先頭アドレス		0x160000	0x160000
ソケットライ ブラリの先頭 アドレス	init( )	0x164000	0x164000
	socket( )	0x16400C	0x16400C
	gethostname( )	0x164012	0x164012
	getpeername( )	0x164018	0x164018
	accept( )	0x16401E	0x16401E
	connect( )	0x164024	0x164024
	listen( )	0x16402A	0x16402A
	shutdown( )	0x164030	0x164030
	send( )	0x164036	0x164036
	recv( )	0x164042	0x164042

## 5.1.3 HOSTSテーブル例

HOSTSテーブルはCモードプログラムで作成し、全ステーションに同一データをローディングします。

## ● Cモードプログラム例

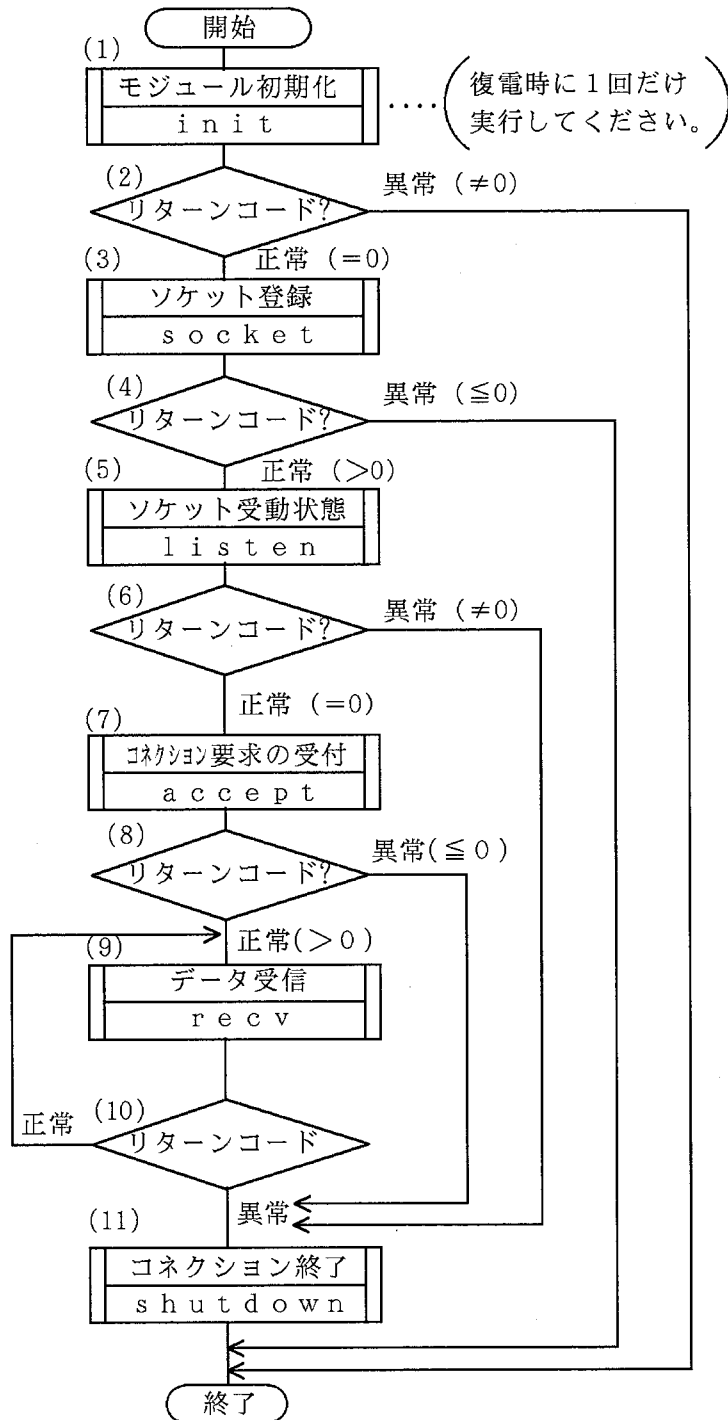
```
#define MAXSAP 3
struct u_host {
    char  name [16] ;
    char  addr [4] ;
};
struct u_host u_host1_g[MAXSAP] = {
    "ET1", 192, 1, 0, 1,
    "ET2", 192, 1, 0, 2,
    "  ",  0, 0, 0, 0,
};
```

## ● ローディング後のHOSTSテーブル

```
0x00160000 45543100 00000000 00000000 00000000 <ET1..... <
0x00160010 C0010001 45543200 00000000 00000000 <タ...ET2..... <
0x00160020 00000000 C0010002 20202000 00000000 <....タ... .. <
0x00160030 00000000 00000000 00000000 00000000 <..... <
```



## 5.1.4 ET1側プログラムのフローチャート



- (1) モジュールの初期化を行います。
- (2) `init()`を呼び出したリターンコードにより、正常か異常かを判定します。
- (3) ポート番号を3001としてソケットの登録を行います。
- (4) 登録されたソケットIDはリターンコードで返されますので、リターンコードが正  
のときは正常に登録されたものと見なします。
- (5) `listen()`を呼び出してソケットを受動状態にします。
- (6) リターンコードにより、正常か異常かを判定します。
- (7) ET2側からの接続要求に対して接続要求を受付けます。
- (8) リターンコードに返ってくるソケットIDが負であれば異常と見なします。
- (9) ET2側から送信されたデータを受信バッファに取り込みます。
- (10) データをすべて取り込んだか、またはエラーコードが返ってくるまで繰り返します。
- (11) 確立した接続を終了させます。

## 5.1.5 ET1側のC言語プログラム例

```

#define SOCK_STREAM 1          /* ソケット タイプ          */
#define MAIN        1L        /* main/sub 番号          */
#define INIT        0x164000L /* init() 先頭アドレス    */
#define SOCKET      0x16400CL /* socket() 先頭アドレス  */
#define LISTEN     0x16402AL /* listen() 先頭アドレス  */
#define ACCEPT     0x16401EL /* accept() 先頭アドレス  */
#define RECV       0x164042L /* recv() 先頭アドレス    */
#define SHUTDOWN   0x164030L /* shutdown() 先頭アドレス */

struct sockaddr {
    short sa_family; /* ソケット タイプ          */
    short sa_port;   /* 自局 ホート番号        */
    short dst_port;  /* 相手局 ホート番号      */
    short rsv0;      /* (予備)                  */
    long  dst_ip;    /* 相手局 IPアドレス      */
    char  e_addr[6]; /* 物理 アドレス          */
    short rsv1;      /* (予備)                  */
};

struct sckidtbl {
    short id; /* ソケット ID          */
    short lcn; /* TCP ローカル コネクション ネーム */
    char  rmt_name[16]; /* 相手局 ホスト名      */
    short local_prt; /* 自局 ホート番号      */
    short remote_prt; /* 相手局 ホート番号    */
    long  remote_ip; /* 相手局 IPアドレス    */
    long  rsv; /* (予備)              */
};

main()
{
    register long (*init)();
    register long (*socket)();
    register long (*listen)();
    register long (*accept)();
    register long (*recv)();
    register long (*shutdown)();
    register long (*term)();

    long s; /* ソケット番号 */
    long err_p, rtn, addr_len;
    char *buf;

    struct sockaddr saddr;
    struct sckidtbl addr;

    buf = (char *)0x1E6000L; /* 受信バッファ先頭アドレス */

    init = ( long (*)() )INIT;
    rtn = ( *init )( &err_p, MAIN );

    if ( rtn == 0 ) {
        saddr.sa_family = SOCK_STREAM;
        saddr.sa_port = 3001;

        socket = ( long (*)() )SOCKET;
        s = ( *socket )( &saddr, "", 0L, &err_p, MAIN );

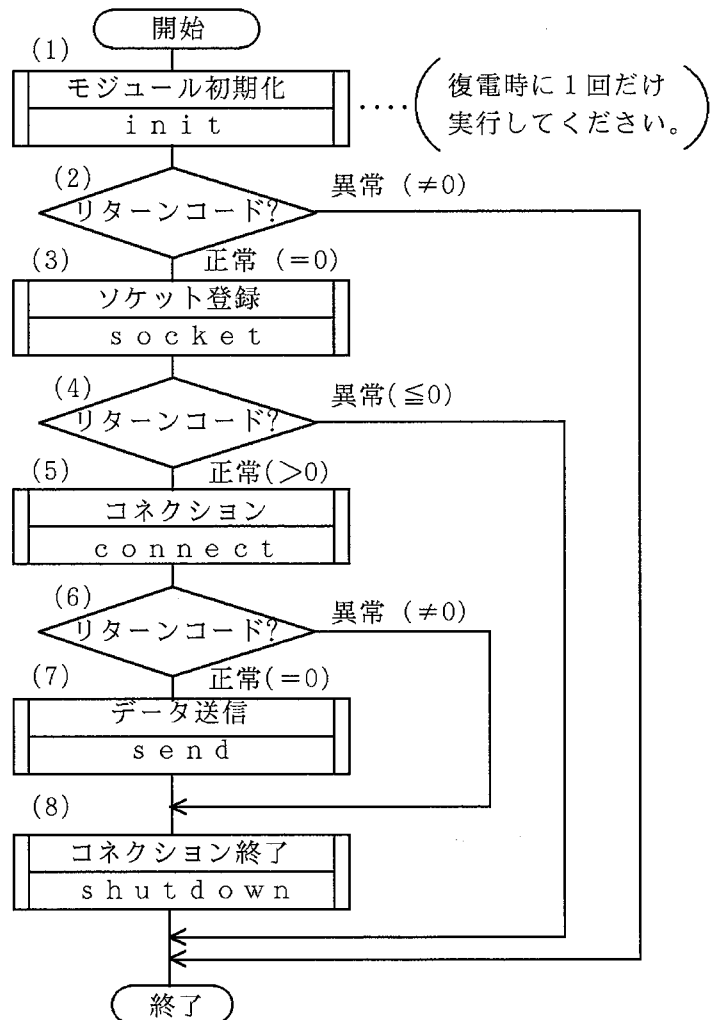
        if ( s > 0 ) {
            listen = ( long (*)() )LISTEN;
            rtn = ( *listen )( s, &err_p, MAIN );

            if ( rtn == 0 ) {
                accept = ( long (*)() )ACCEPT;
                rtn = ( *accept )( s, &saddr, &addr_len, &err_p, MAIN );

                if ( rtn != -1 ) {
                    rtn = 0;
                    do {
                        buf = buf + rtn;
                        if( buf >= (char *)0x1E6400L )
                            buf = (char *)0x1E6000L; /* 受信バッファ先頭アドレス */
                        recv = ( long (*)() )RECV;
                        rtn = ( *recv )( s, buf, 1024L, &err_p, MAIN, 1000L );
                    }while( rtn >= 0 && err_p == 0 );
                }
            }
            shutdown = ( long (*)() )SHUTDOWN;
            rtn = ( *shutdown )( s, 1L, &err_p, MAIN );
        }
    }
}

```

## 5.1.6 ET2側プログラムのフローチャート



- (1) モジュールの初期化を行います。
- (2) `init()` を呼び出したリターンコードにより、正常か異常かを判定します。
- (3) ポート番号を 3001 としてソケットの登録を行います。
- (4) 登録されたソケット ID はリターンコードで返されますので、リターンコードが正  
のときは正常に登録されたものと見なします。
- (5) ET2 から ET1 へコネクションを要求し、ET1 がコネクションの要求を受け付  
けた場合にコネクションの確立を行います。
- (6) `connect()` を呼び出したリターンコードにより、正常か異常かを判定しま  
す。
- (7) 送信メッセージを送信します。
- (8) `connect()` で確立したコネクションを終了させます。

## 5.1.7 ET2側のC言語プログラム例

```

#define SOCK_STREAM 1 /* ソケット タイプ */
#define MAIN 1L /* main/sub 番号 */
#define INIT 0x164000L /* init() 先頭アドレス */
#define SOCKET 0x16400CL /* socket() 先頭アドレス */
#define CONNECT 0x164024L /* connect() 先頭アドレス */
#define SEND 0x164036L /* send() 先頭アドレス */
#define SHUTDOWN 0x164030L /* shutdown() 先頭アドレス */

struct sockaddr {
    short sa_family; /* ソケット タイプ */
    short sa_port; /* 自局 ポート番号 */
    short dst_port; /* 相手局 ポート番号 */
    short rsv0; /* (予備) */
    long dst_ip; /* 相手局 IPアドレス */
    char e_addr[6]; /* 物理 アドレス */
    short rsv1; /* (予備) */
};

struct sckidtbl {
    short id; /* ソケット ID */
    short lcn; /* TCP ローカル コネクション ネーム */
    char rmt_name[16]; /* 相手局 ホスト名 */
    short local_prt; /* 自局 ポート番号 */
    short remote_prt; /* 相手局 ポート番号 */
    long remote_ip; /* 相手局 IPアドレス */
    long rsv; /* (予備) */
};

main()
{
    register long (*init)();
    register long (*socket)();
    register long (*connect)();
    register long (*send)();
    register long (*shutdown)();
    register long (*term)();

    long s; /* ソケット番号 */
    long err_p, rtn, name_len;
    char *msg;

    struct sockaddr saddr;
    struct sckidtbl idtbl;

    msg = (char *)0x1E6000L; /* メッセージデータ先頭アドレス */

    init = ( long (*)() )INIT;
    rtn = ( *init )( &err_p, MAIN );

    if ( rtn == 0 ) {
        saddr.sa_family = SOCK_STREAM;
        saddr.sa_port = 3001;
        saddr.dst_port = 3001;

        socket = ( long (*)() )SOCKET;
        s = ( *socket )( &saddr, "ET1", 3L, &err_p, MAIN );

        if ( s > 0 ) {
            connect = ( long (*)() )CONNECT;
            rtn = ( *connect )( s, &idtbl, &name_len, &err_p, MAIN );

            if ( rtn == 0 ) {
                send = ( long (*)() )SEND;
                rtn = ( *send )( s, msg, 1024L, &err_p, MAIN );
            }
            shutdown = ( long (*)() )SHUTDOWN;
            rtn = ( *shutdown )( s, 1L, &err_p, MAIN );
        }
    }
}

```

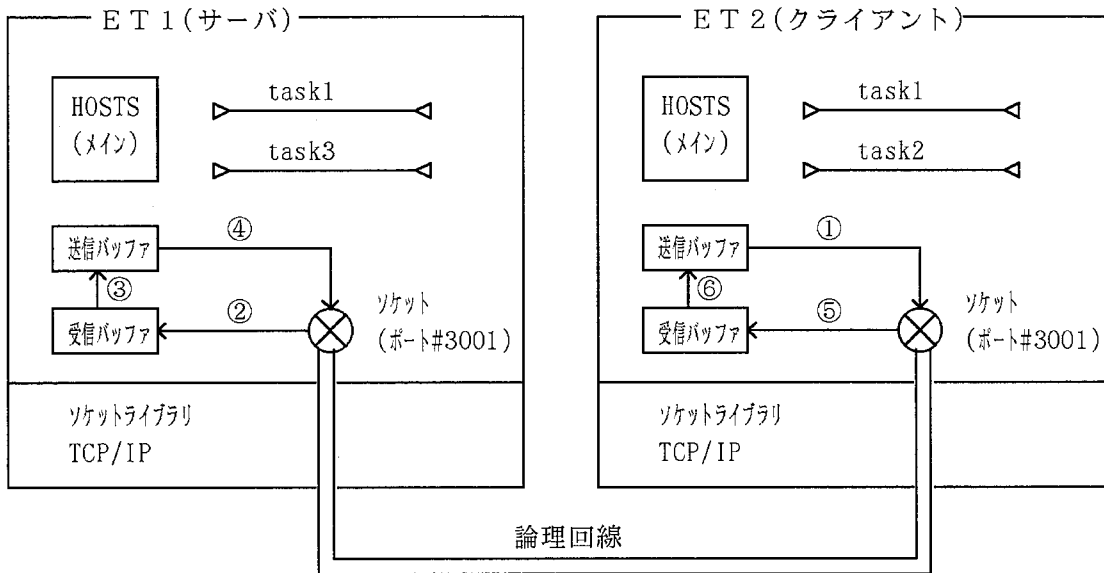
5.2 ソケットライブラリによるCPU間連続通信プログラム例

5.2.1 システム構成

5.1.1項のシステム構成と同様です。

5.2.2 プログラム構成

プログラム構成を以下に示します。ET1とET2を論理回線で接続し、ET1とET2の間で1024バイトのデータを送受信します。



- ① データ送信 (1024バイト)
- ② データ受信 (1024バイト)
- ③ データコピー (受信バッファ→送信バッファ)
- (ループ) ④ データ送信 (1024バイト)
- ⑤ データ受信 (1024バイト)
- ⑥ データ比較チェック (送信バッファ, 受信バッファ)  
データ反転コピー (受信バッファ→送信バッファ)

項 目	ホスト名	
	ET 1	ET 2
機 能	受信/送信 (サーバ)	送信/受信/比較 (クライアント)
送信バッファ	アドレス : 0x1E1000 バイト数 : 1024	アドレス : 0x1E1000 バイト数 : 1024
受信バッファ	アドレス : 0x1E2000 バイト数 : 1024	アドレス : 0x1E2000 バイト数 : 1024
ポート番号	3 0 0 1	3 0 0 1
HOSTSテーブル先頭アドレス	0x160000	0x160000
イニシャルタスク名	t a s k 1	t a s k 1
通信タスク名	t a s k 3	t a s k 2

## タスク登録

	タスク名	プログラム先頭アドレス	USP	実行レベル	タスクNo.	区 分
ET 1	task1	0x1E8000	0x1E8800	2	1	初期設定
	task3	0x1EA000	0x1EA800	3	2	サーバ
ET 2	task1	0x1E8000	0x1E8800	2	1	初期設定
	task2	0x1E9000	0x1E9800	3	2	クライアント

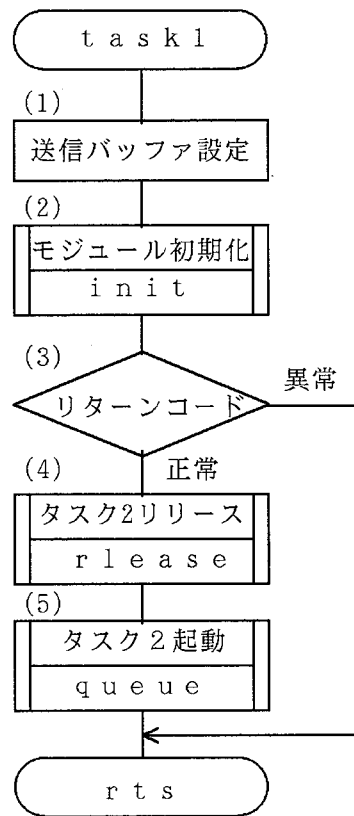
タスク登録は、『CPMSマニュアル V4 PSEα デバッガ編』(SP-3-222)マニュアルを参照してください。

5.2.3 HOSTSテーブル例

5.1.3項のHOSTSテーブル例と同様です。

5.2.4 イニシャルタスク (task1) のプログラムのフローチャート

ET1, ET2とも共通です。



- (1) 送信バッファの初期設定を行います。
- (2) S10ET LINKモジュール (メイン) の初期化を行います。
- (3) init( )のリターンコードにより正常か異常かを判定します。
- (4) タスク番号=2のプログラム (task2またはtask3) をリリースします。
- (5) タスク番号=2のプログラムを起動します。



## 5.2.5 イニシャルタスク (task1) のC言語プログラム例

```

#define SOCK_STREAM 1          /* ソケット タイプ          */
#define MAIN 1L              /* モジュール番号(main)    */
#define INIT 0x164000L      /* init() 先頭アドレス    */

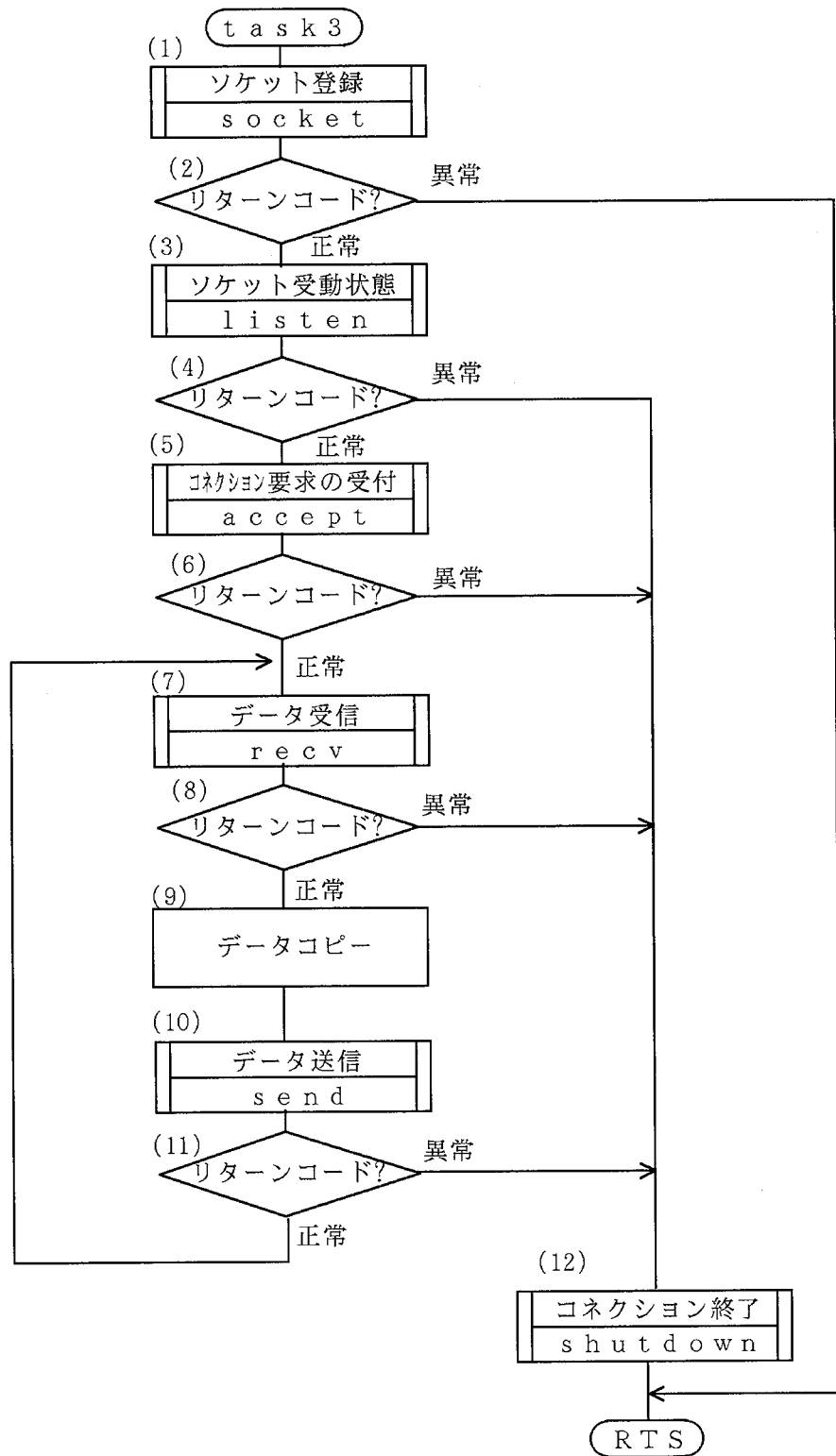
/*****
/* task1 : 初期化処理(main) */
*****/
main()
{
    register long (*init)();
    long err_p, rtn;
    long tn, fact;
    short i;
    char *buf;

    buf = (char *)0x1E1000L;          /* 送信データ設定 */
    for( i = 0 ; i < 1024 ; i++ )
        *buf++ = 0x55;

    init = ( long (*)() )INIT;
    if( ( *init )( &err_p, MAIN ) == 0 ) {
        tn = 2L;                      /* TASK2, TASK3 */
        fact = 0L;
        rleas( &tn );                 /* タスク リリース */
        queue( &tn, &fact );        /* タスク 起動 */
    }
}

```

5.2.6 ET1側 (task 3) プログラムのフローチャート



- (1) ポート番号を3001としてソケットの登録を行います。
- (2) 登録されたソケットIDはリターンコードで返されます。  
リターンコードがゼロより大きい場合、正常に登録されたものと見なします。
- (3) `listen()`関数を呼び出してソケットを受動状態にします。
- (4) リターンコードにより、正常か異常かを判定します。
- (5) ET2側からのコネクション要求に対してコネクション要求を受付けます。
- (6) リターンコードに返ってくるソケットIDが負であれば異常と見なします。
- ループ → (7) ET2側から送信されたデータを受信バッファに取り込みます。
- (8) リターンコードがエラーまたは、取り込みデータなしの場合、(12)を実行します。
- (9) 受信バッファのデータを送信バッファへコピーします。
- (10) 送信バッファのデータをET2に送信します。
- (11) リターンコードにより、正常か異常かを判定し、正常な場合(7)~(11)を繰り返します。
- (12) 確立したコネクションを終了させます。

## 5.2.7 ET1側 (task3) のC言語プログラム例

```

#define SOCK_STREAM 1 /* ソケット タイプ */
#define MAIN 1L /* モジュール番号(main) */
#define SOCKET 0x16400CL /* socket() 先頭アドレス */
#define LISTEN 0x16402AL /* listen() 先頭アドレス */
#define ACCEPT 0x16401EL /* accept() 先頭アドレス */
#define SEND 0x164036L /* send() 先頭アドレス */
#define RECV 0x164042L /* recv() 先頭アドレス */
#define SHUTDOWN 0x164030L /* shutdown() 先頭アドレス */

struct sockaddr {
    short sa_family; /* ソケット タイプ */
    short sa_port; /* 自局 ポート番号 */
    short dst_port; /* 相手局 ポート番号 */
    short rsv0; /* (予備) */
    long dst_ip; /* 相手局 IPアドレス */
    char e_addr[6]; /* 物理 アドレス */
    short rsv1; /* (予備) */
};

struct sckidtbl {
    short id; /* ソケット ID */
    short lcn; /* TCP ローカル コネクション ネーム */
    char rmt_name[16]; /* 相手局 ホスト名 */
    short local_prt; /* 自局 ポート番号 */
    short remote_prt; /* 相手局 ポート番号 */
    long remote_ip; /* 相手局 IPアドレス */
    long rsv; /* (予備) */
};

/*****
/* task3 : サーバ (ET1)
*****/
main()
{
    register long (*socket)();
    register long (*connect)();
    register long (*listen)();
    register long (*accept)();
    register long (*send)();
    register long (*recv)();
    register long (*shutdown)();
    register long (*term)();

    long s; /* ソケット番号 */
    long err_p, rtn, addr_len;
    short i;
    char *sbuf, *rbuf;

    struct sockaddr saddr;
    struct sckidtbl addr;

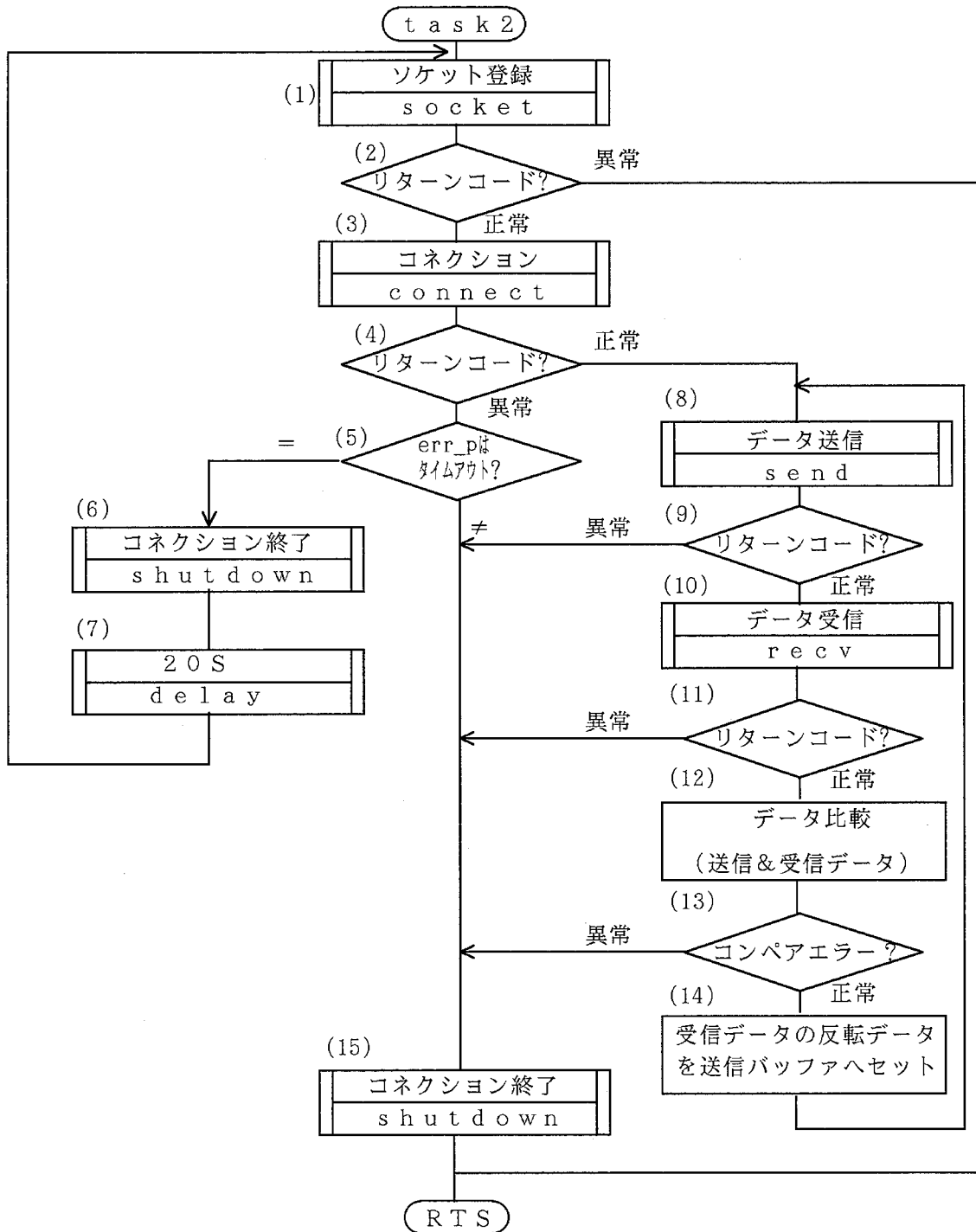
    sbuf = (char *)0x1E1000L; /* メッセージデータ先頭アドレス */
    rbuf = (char *)0x1E2000L; /* 受信バッファ先頭アドレス */

    saddr.sa_family = SOCK_STREAM;
    saddr.sa_port = 3001;

    socket = ( long (*)() )SOCKET;
    s = ( *socket )( &saddr, "", 0L, &err_p, MAIN );
    if( s > 0 ) {
        listen = ( long (*)() )LISTEN;
        if( ( *listen )( s, &err_p, MAIN ) == 0 ) {
            accept = ( long (*)() )ACCEPT;
            if( ( *accept )( s, &saddr, &addr_len, &err_p, MAIN ) != -1 ) {
                while( 1 ) {
                    recv = ( long (*)() )RECV;
                    rtn = ( *recv )( s, rbuf, 1024L, &err_p, MAIN, 1000L );
                    if( (rtn == -1) || (rtn == 0) )
                        break;
                    for(i = 0; i < 1024; i++)
                        sbuf[i] = rbuf[i];
                    send = ( long (*)() )SEND;
                    if( ( *send )( s, sbuf, 1024L, &err_p, MAIN ) == -1 )
                        break;
                }
            }
        }
        shutdown = ( long (*)() )SHUTDOWN;
        ( *shutdown )( s, 1L, &err_p, MAIN );
    }
}

```

5.2.8 ET2側 (task2) プログラムのフローチャート



- ル
- (1) ポート番号を3001としてソケットの登録を行います。
- (2) 登録されたソケットIDは、リターンコードで返されます。  
リターンコードがゼロより大きい場合、正常に登録されたものと見なします。
- ル (3) ET2からET1へコネクションを要求し、ET1がコネクションの要求を受け付けた場合にコネクションの確立を行います。
- ル (4) `connect()`を呼び出したリターンコードにより、正常か異常かを判定します。正常な場合は、(8)から実行します。
- ブ (5) `err_p` (エラーコード)の内容が、タイムアウトの場合、(6)、(7)を行い、それ以外の場合(15)を行います。
- (6) 異常の場合で、`err_p = 7, 8`のときは、`shutdown()`を呼び出します。
- (7) 20秒間のDELAYマクロを発行し、(1)から繰り返し実行します。
- ル → (8) 送信バッファのデータを送信します。
- (9) リターンコードにより、正常か異常かを判定します。
- ル (10) ET1から送信されたデータを受信バッファへ取り込みます。
- ル (11) リターンコードが正常か異常か、または取り込みデータなしかを判定します。
- ル (12) 自局の送信バッファと受信バッファのデータの比較を行います。
- ブ (13) 比較結果の判定を行います。
- (14) 受信データの反転データを送信バッファへコピーします。
- (15) 確立したコネクションを終了させます。

## 5.2.9 ET2側 (task2) のC言語プログラム例

```

#define SOCK_STREAM 1 /* ソケット タイプ */
#define MAIN 1L /* モジュール番号(main) */
#define SOCKET 0x16400CL /* socket() 先頭アドレス */
#define CONNECT 0x164024L /* connect() 先頭アドレス */
#define SEND 0x164036L /* send() 先頭アドレス */
#define RECV 0x164042L /* recv() 先頭アドレス */
#define SHUTDOWN 0x164030L /* shutdown() 先頭アドレス */

struct sockaddr {
    short sa_family; /* ソケット タイプ */
    short sa_port; /* 自局 ポート番号 */
    short dst_port; /* 相手局 ポート番号 */
    short rsv0; /* (予備) */
    long dst_ip; /* 相手局 IPアドレス */
    char e_addr[6]; /* 物理 アドレス */
    short rsv1; /* (予備) */
};

struct sckidtbl {
    short id; /* ソケット ID */
    short lcn; /* TCP ローカル コネクション ネーム */
    char rmt_name[16]; /* 相手局 ホスト名 */
    short local_prt; /* 自局 ポート番号 */
    short remote_prt; /* 相手局 ポート番号 */
    long remote_ip; /* 相手局 IPアドレス */
    long rsv; /* (予備) */
};

/*****
/* task2 : クライアント(ET2)
*****/
main()
{
    register long (*socket)();
    register long (*connect)();
    register long (*send)();
    register long (*recv)();
    register long (*shutdown)();
    register long (*term)();

    long s; /* ソケット番号 */
    long err_p, rtn, name_len;
    long t;
    short i, cerr_flag;
    char *sbuf, *rbuf;

    struct sockaddr saddr;
    struct sckidtbl idtbl;

    sbuf = (char *)0x1E1000L; /* メッセージデータ先頭アドレス */
    rbuf = (char *)0x1E2000L; /* 受信バッファ先頭アドレス */
    t = 20000L; /* デレイ時間(msec) */

    saddr.sa_family = SOCK_STREAM;
    saddr.sa_port = 3001;
    saddr.dst_port = 3001;

    while( 1 ) {
        socket = ( long (*)() )SOCKET ;
        s = (*socket)( &saddr, "ET1", 3L, &err_p, MAIN );
        if( s <= 0 ) break ;

        connect = ( long (*)() )CONNECT ;
        rtn = (*connect)( s, &idtbl, &name_len, &err_p, MAIN );
        if( rtn == -1 && ( err_p == 7 || err_p == 8 ) ) {
            shutdown = ( long (*)() )SHUTDOWN ;
            (*shutdown)( s, 1L, &err_p, MAIN );
            delay( &t );
        }
        else break ;
    }

    if( s > 0 ) {
        if( rtn == 0 ) {
            while( 1 ) {
                send = ( long (*)() )SEND ;
                if( (*send)( s, sbuf, 1024L, &err_p, MAIN ) == -1 )
                    break ;
                recv = ( long (*)() )RECV ;
                rtn = (*recv)( s, rbuf, 1024L, &err_p, MAIN, 1000L );
                if( (rtn == -1) || (rtn == 0) )
                    break ;
                cerr_flag = 0 ;
                for( i = 0; i < 1024; i++ ) {
                    if( sbuf[i] != rbuf[i] ) {
                        cerr_flag = 1 ;
                        break ;
                    }
                }
            }
        }
    }
}

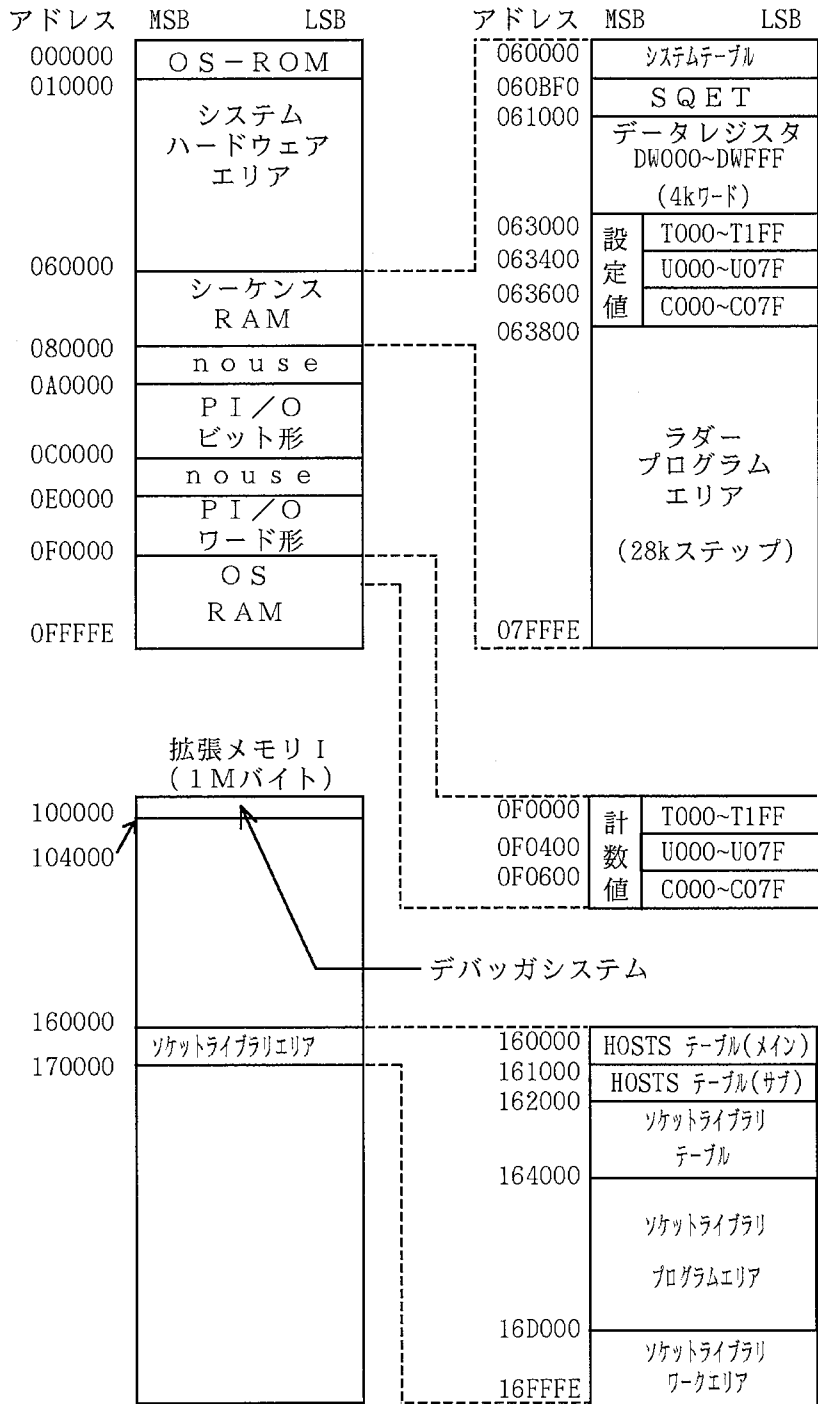
```

```
        sbuf[i] = ~rbuf[i] ;
    }
    if( cerr_flag == 1 )
        break ;
    }
}
shutdown = ( long (*)() )SHUTDOWN ;
( *shutdown )( s, 1L, &err_p, MAIN ) ;
}
}
```

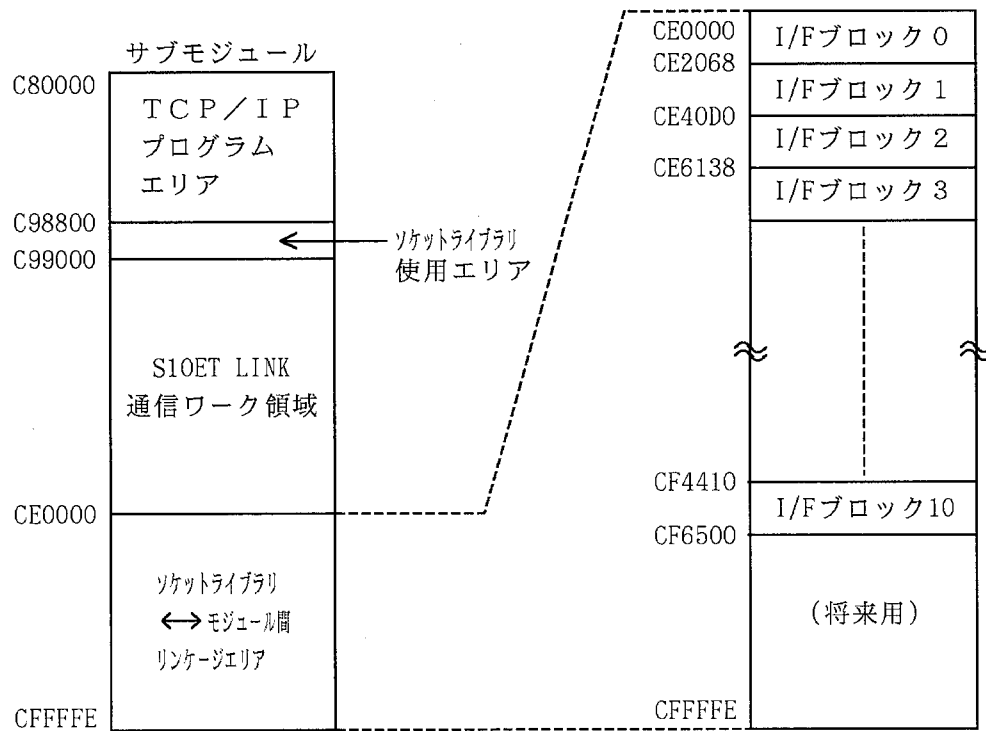
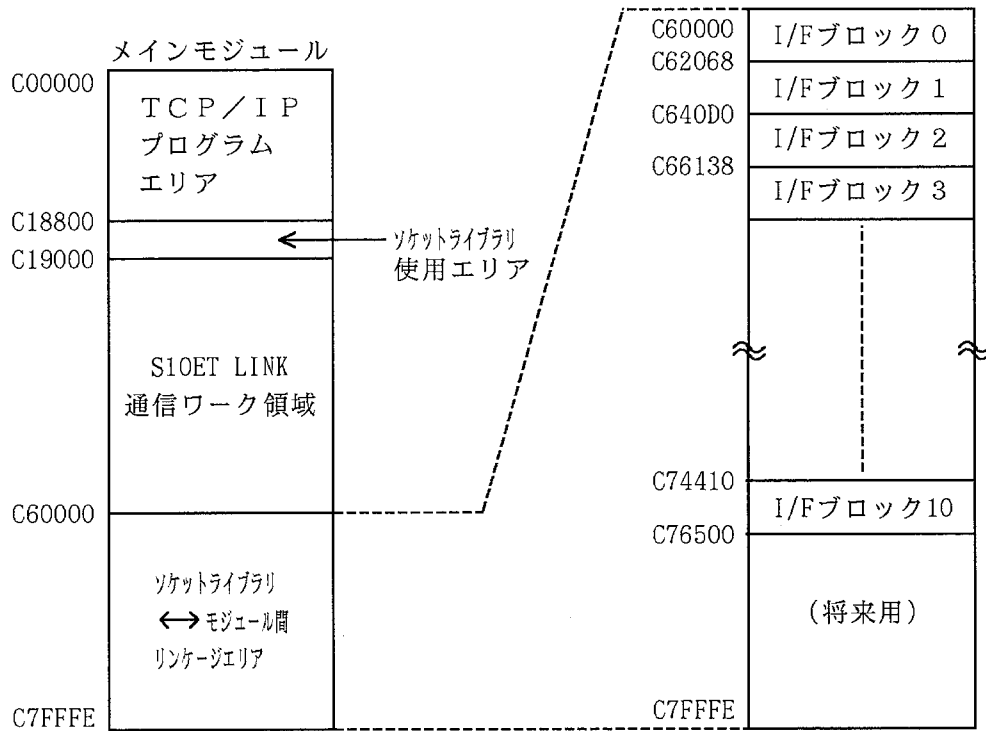


## 6 メモリマップ

6.1 CPUのメモリマップ



## 6.2 S10ET LINKのメモリマップ





7 P S E  $\alpha$

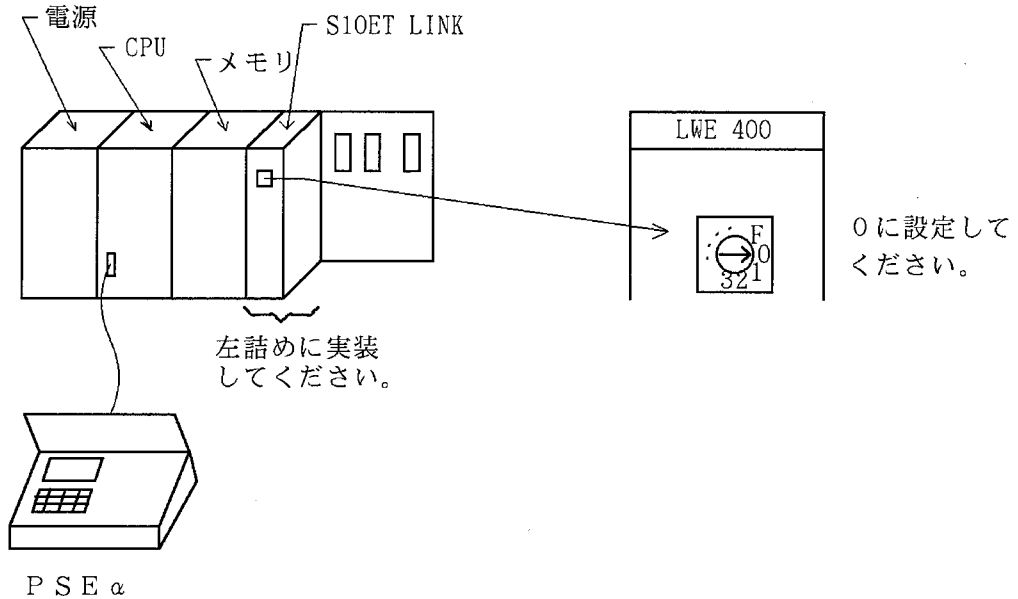
オペレーション

この章はPSEαを使用するお客様を対象としています。  
 パソコンプログラミングを使用される場合は、「8 パソコンプログラミングオペレーション」を読んでください。

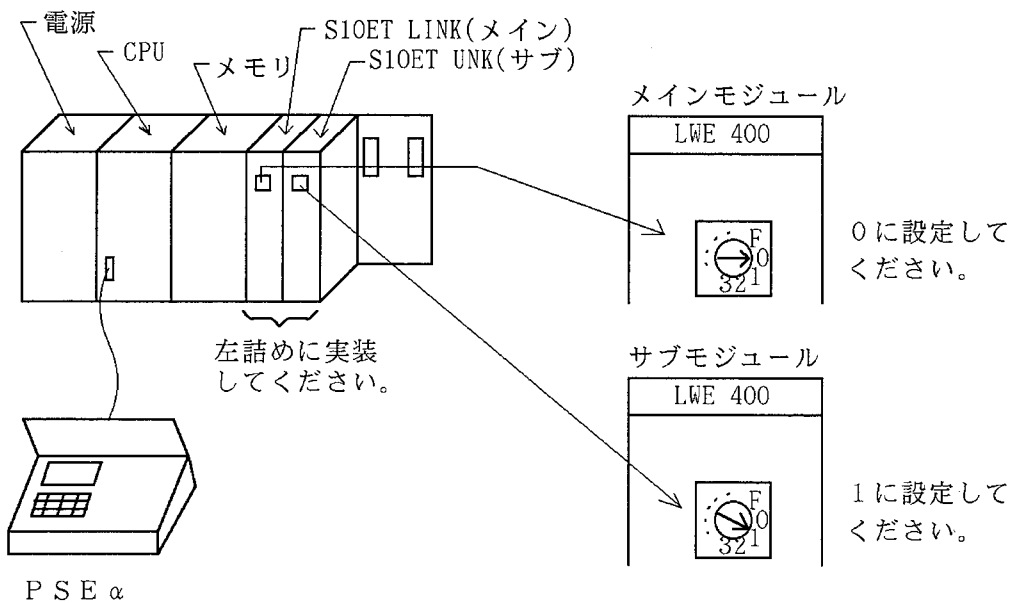
**7.1 システムを立上げるにあたり**

7.1.1 システム構成

<1チャンネル構成>



<2チャンネル構成>



## 7.1.2 基本オペレーション

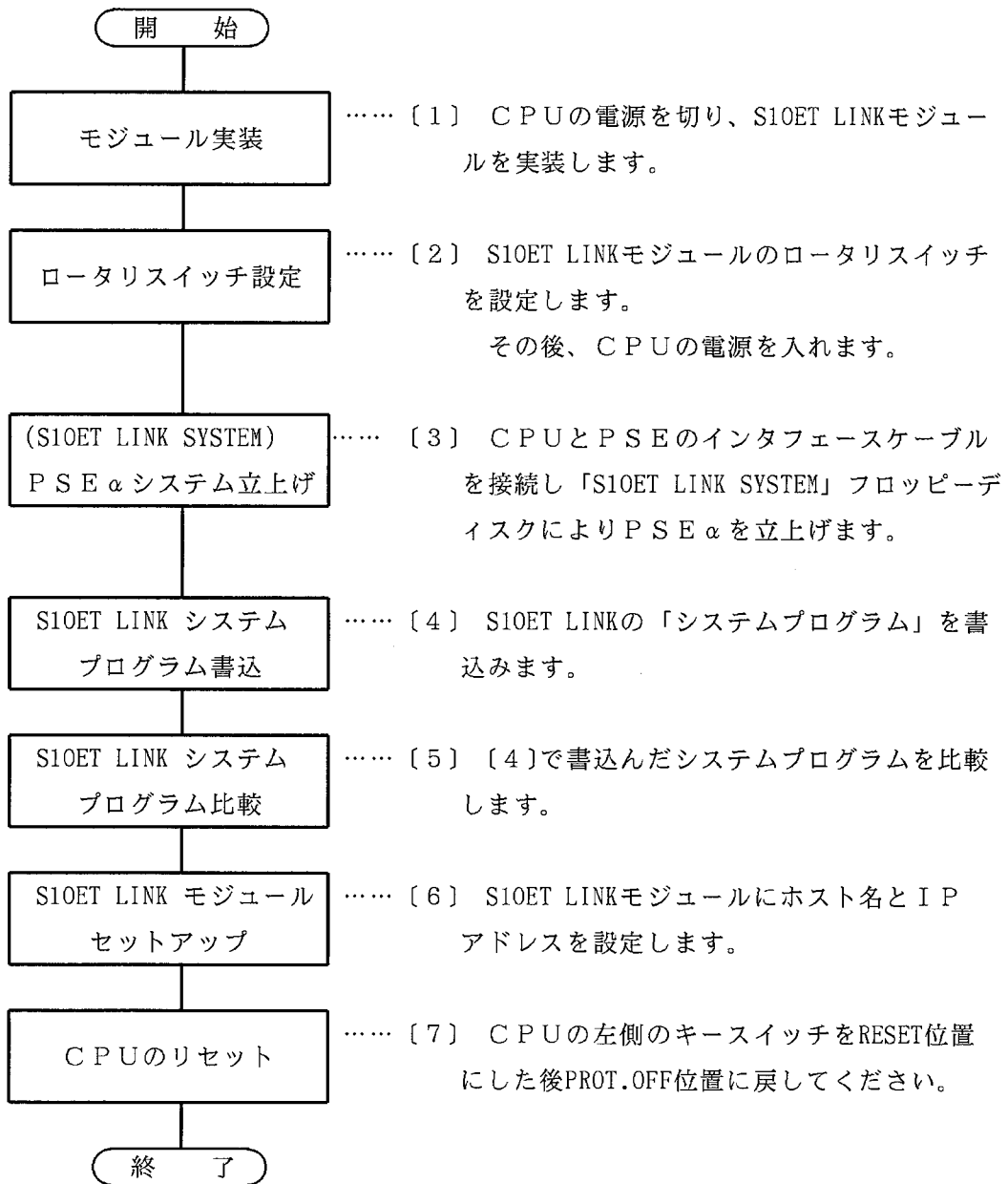
- PSEαの操作は、画面に表示されたカーソルにそって入力することにより簡単に行えます。
- 選択する操作には、次の2種類があります。
  - ・ 選択項目のナンバを入力する。
  - ・ 設定キーまたは再設定キーを選択して押す。
- 設定キーまたは再設定キーを押す場合の操作  
画面に〔SET/RTY/CLS〕のように選択キーが表示される場合、それらのキーの意味は、次のようになっています。

表示画面名称	対応するキー	意味
SET	<b>設 定</b> キー	OKの時
CLS	<b>終 了</b> キー	一つまたはそれ以上前の画面に戻す時
RTY	<b>再 設 定</b> キー	データの再設定をする時
CNT	<b>続 行</b> キー	処理を繰返し行う時
DEL	<b>削 除</b> キー	ファイルなどの削除を行う時

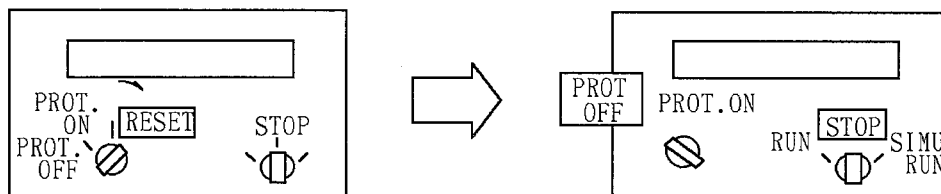
(注) CLS: CLOSE  
 RTY: RETRY  
 CNT: CONTINUE  
 DEL: DELETE

**7.2 システム立ち上げ**

7.2.1 S10ET LINKシステム立ち上げ手順

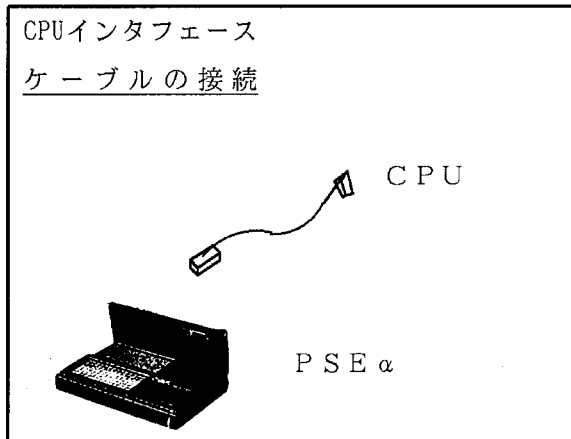


\* RESET位置のままではCPUは動作しません。

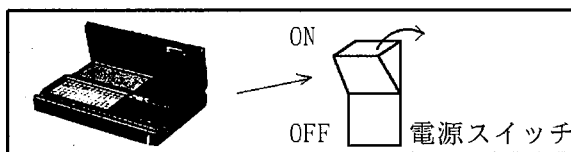




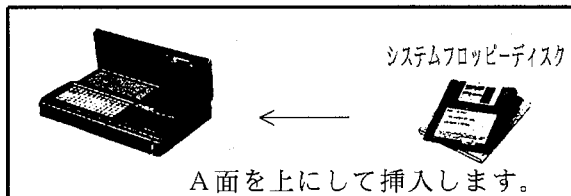
## 7.2.2 PSEαシステム立ち上げ手順



- 〔1〕 PSEαの電源がOFFの状態CPUとインタフェースケーブルを正しく接続します。この時、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) に設定してください。



- 〔2〕 PSEαの電源をONにしてください。



- 〔3〕 システムフロッピーディスク「S10ET LINK SYSTEM」をPSEαに挿入してください。

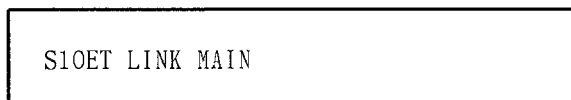


- 〔4〕 PSEαの画面上に、左図のメッセージが表示されます。任意のキーを押してください。

〔システム書込中〕



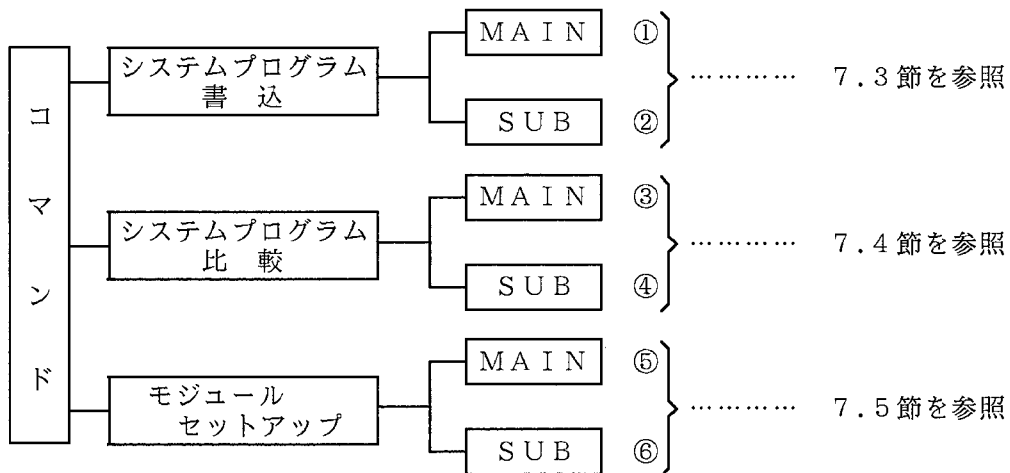
- 〔5〕 PSEαの画面上に「SYSTEM LOADING」と表示され、フロッピーディスクからシステムプログラムがPSEαのメモリへ書込まれます。



- 〔6〕 S10ET LINKサポートシステムメイン画面が表示されます。

```

S10ET LINK MAIN
*** S10ET LINK SUPPORT SYSTEM (Ver*.* Rev*.*) ***
KEYIN MENU No. = ■
-----
M E N U
-----
1 : (MAIN) SYSTEM PROGRAM LOAD
2 : (SUB)
3 : (MAIN) SYSTEM PROGRAM COMPARE
4 : (SUB)
5 : (MAIN) MODULE SETUP
6 : (SUB)
-----
    
```

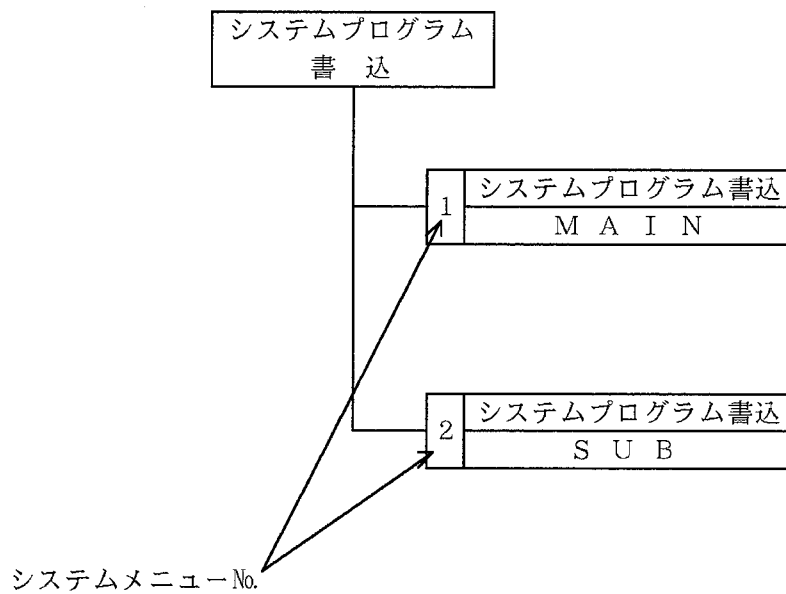


- ①, ② システムプログラム書込  
 フロッピーディスクのS10ET LINKシステムプログラムをCPUおよびS10ET LINKモジュールのメモリへ書込みます。
- ③, ④ システムプログラム比較  
 CPUおよびS10ET LINKモジュールのメモリ内容とフロッピーディスクの内容を比較します。
- ⑤, ⑥ モジュールセットアップ  
 S10ET LINKモジュールのホスト名、IPアドレスを表示、変更します。

### 7.3 S10ET LINKシステムプログラム書込

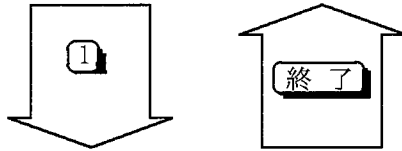
#### 7.3.1 機能

この処理は、システムフロッピーディスクに格納されている「S10ET LINKシステムプログラム」をCPU本体およびS10ET LINKモジュールへ書込みます。



7.3.2 オペレーション

SIOET LINK MAIN  
KEYIN MENU No. = ■



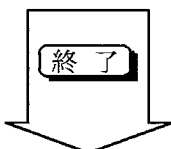
PROGRAM LOAD  
SET SYSTEM F/D & STRIKE [SET] KEY!



PROGRAM LOAD  
SYSTEM PROGRAM LOADING



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



\*\*\*\* PLEASE PCS RESET !! \*\*\*\*

[1] SIOET LINKサポートシステムメイン画面から **1** を選択します。  
・ SIOET LINKモジュールの実装を確認し、CPUのキースイッチをSTOP、プロテクトOFFに設定してください。

[2] 「SIOET LINK SYSTEM」3.5インチフロッピーディスクが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。  
・ ERR = 11が表示された場合は、システムプログラムファイルが壊れていますので、別のシステムフロッピーディスクに交換してください。

[3] システムプログラムの書込みが開始されます。

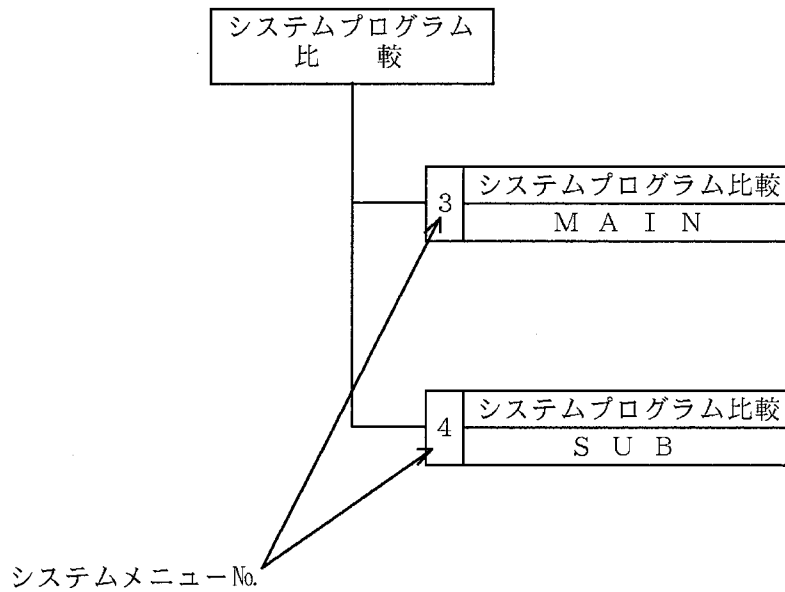
[4] 左図のメッセージが表示されたら書込みは正常終了です。  
**終了** キーを押してください。

[5] 左図のメッセージが表示されたらCPUをリセットしてください。

## 7.4 S10ET LINKシステムプログラム比較

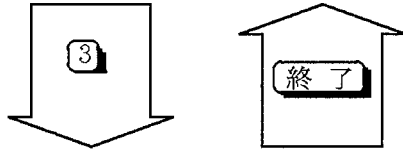
### 7.4.1 機能

この処理は、書込済の「S10ET LINKシステムプログラム」とシステムフロッピーディスクに格納されている「S10ET LINKシステムプログラム」を比較します。



7.4.2 オペレーション

S10ET LINK MAIN  
KEYIN MENU No. = ■



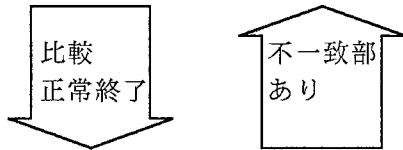
- [1] S10ET LINKサポートシステムメイン画面から [3] を選択します。
- ・ S10ET LINKモジュールの実装を確認してください。

PROGRAM COMPARE  
SET SYSTEM F/D & STRIKE [SET] KEY !



- [2] 「S10ET LINK SYSTEM」 3.5 インチフロッピーディスクがPSEα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。
- ・ ERR = 1 1 が表示された場合は、システムプログラムファイルが壊れていますので別のシステムフロッピーディスクに交換してください。

PROGRAM COMPARE  
SYSTEM PROGRAM COMPARING



- [3] システムプログラムの比較が開始されます。

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

- [4] 左図のメッセージが表示されたら比較は正常終了です。
- [終了] キーを押してください。

不一致部ありの場合

```

PROGRAM COMPARE
STRIKE ANY KEY ■ [CNT/CLS]
COMPARE ERROR ON PCS MEMORY ADDRESS=/C00000
***** PCS MEMORY DATA *****
/C00000 0050 0000 0050 0000 0050 0000 0050 0000
/C00010 0050 0000 0050 0000 0050 0000 0050 0000
/C00020 0050 0000 0050 0000 0050 0000 0050 0000
/C00030 0050 0000 0050 0000 0050 0000 0150 0000
/C00040 0000 0000 0000 0000 0000 0000 0000 0000
/C00050 0000 0000 0000 0000 0000 0000 0000 0000
/C00060 0000 0000 0000 0000 0000 0000 0000 0000
/C00070 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S10ET LINK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** … 次のアドレスから比較を再開する場合。

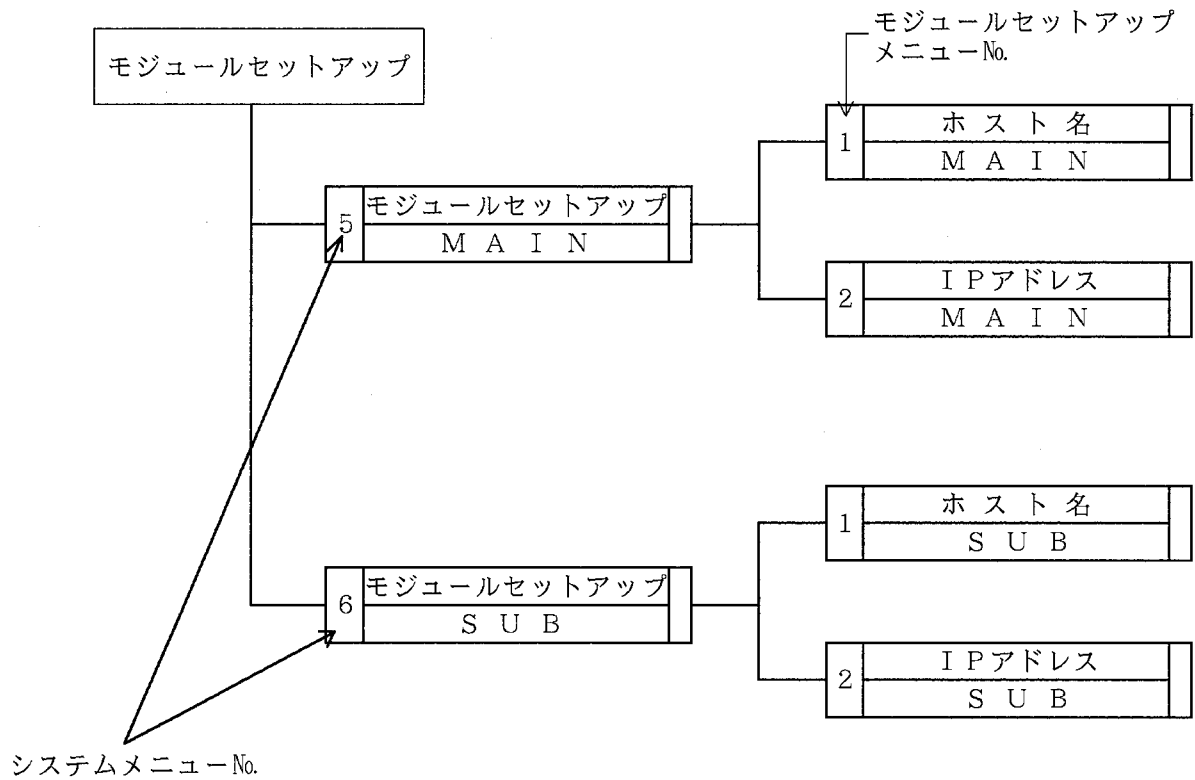
**終 了** … 処理を終了する場合。(S10ET LINKサポートシステムメイン画面へ)

**その他のキー** … 画面に表示する表示内容を切り替える場合。  
 (CPUおよびS10ET LINKモジュールの内容 ⇄ フロッピーディスクの内容)

**7.5 S10ET LINKモジュールのセットアップ**

7.5.1 機能

この処理は、通信に必要な情報をS10ET LINKモジュールへ設定します。  
 情報にはホスト名、IPアドレスがありますので、両方とも設定してください。





## 7.5.2 オペレーション

```

SIOET LINK MAIN
*** SIOET LINK SUPPORT SYSTEM
KEYIN MENU No. = ■

```

〔1〕 SIOET LINKサポートシステム  
メイン画面から **5** を選択しま  
す。

- ・ SIOET LINKモジュールの実  
装を確認し、CPUのキースイ  
ッチをSTOP、プロテクトOFF  
に設定してください。



```

MODULE SETUP MENU
KEYIN No. = ■
*** SIOET LINK TABLE (MAIN) ***
1 :: HOST NAME EDIT
2 :: IP ADDRESS EDIT

```

〔2〕 モジュールセットアップメ  
ニュー画面から設定する項目の番  
号 **1** または **2** を選択し  
ます。



```

MODULE SETUP MENU
KEYIN HOST NAME = ■

```

```

MODULE SETUP MENU
KEYIN IP ADDRESS = ■ . . .

```

〔3〕 モジュールセットアップメニュー  
画面でホスト名を設定します。

ホスト名はASCIIコード（数字、  
アルファベット、特殊文字）16桁  
以内です。

（例）ホスト名を“EWS1”とするとき  
は **E** **W** **S** **1** を入力し、  
**設定** キーを押してください。

〔4〕 モジュールセットアップメニュー画面で  
IPアドレスを設定します。

（例）IPアドレスを“192.001.000.001”と

するときは **1** **9** **2** . **0** **0**

**1** . **0** **0** **0** . **0** **0** **1**

を入力し、**設定** キーを押してくださ  
い。

〔5〕 モジュールセットアップ終了後、必ずCPUをリセットしてください。

(注1) [4]のIPアドレスは、10進数を3桁ごとに“.(ピリオド)”で区切って表現します。

MMM. NNN. XXX. YYY  
( MMM, NNN, XXXは0～255の10進数 )  
( YYYは1～254の10進数 )

既設のネットワーク上に、このマシンを追加登録する場合は、そのネットワークに合ったIPアドレスとしてください。

新設の場合は下記をおすすめします。

MMM=192 ..... クラスCを選択  
NNN=001 } ..... ネットワーク番号  
XXX=000 }  
YYY=001～200 ..... ステーションごとの一貫No.

# 8 パソコンプログラミング オペレーション

この章は、パソコンプログラミングを使用するお客様を対象としています。

PSE $\alpha$ を使用される場合は「7 PSE $\alpha$ オペレーション」を読んでください。

## 8.1 インストール

●機能：システムフロッピーディスク（S10ET LINKシステム）のシステムプログラムをパソコンのハードディスクにインストールします。

●確認項目：この例では、HDDをAドライブ、FDDをBドライブとしています。

以降名称は略称で表記します。 ハードディスク：HD      ハードディスクドライブ：HDD

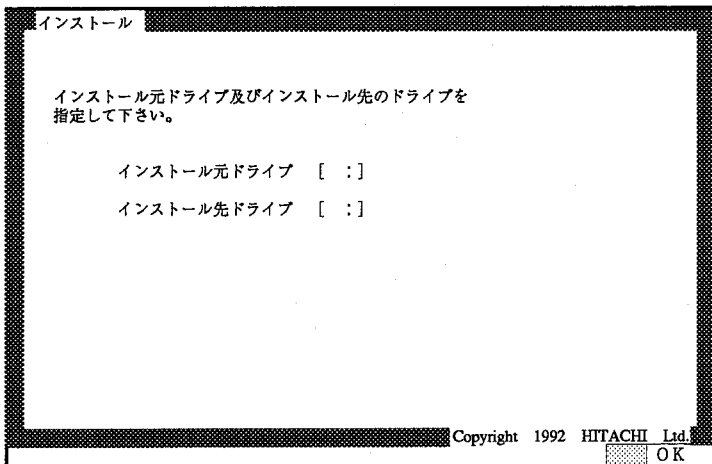
フロッピーディスク：FD      フロッピーディスクドライブ：FDD

●オペレーション：(1) システムFDをパソコンのFDDに挿入してください。

(2) B：と入力後、[RETURN]キーを押してください。カレントドライブがFDDに変更します。

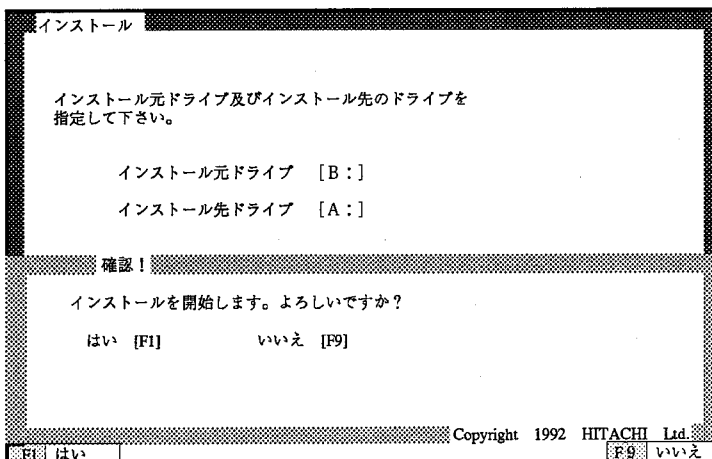
(3) INSTALLと入力後、[RETURN]キーを押してください。

インストールプログラムが立ち上がります。



●システムをインストールしたい

- ・インストール元ドライブにFDDドライブ名称を入力してください。
- ・インストール先ドライブにHDDドライブ名称を入力してください。

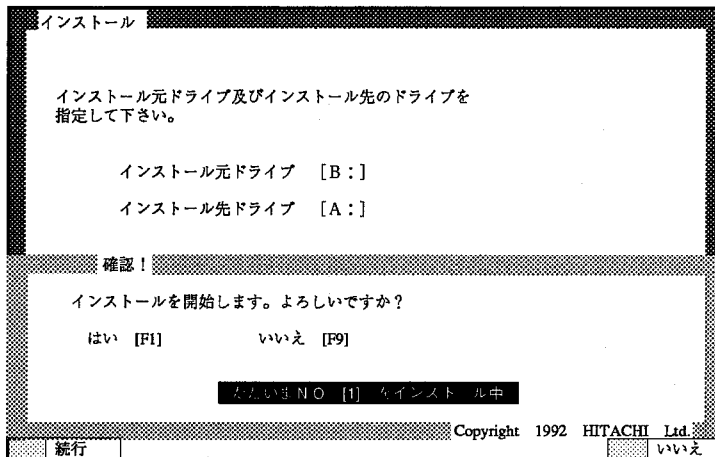


●システムインストールを開始する

- ・ **[F1]** (はい) キーを押してください。(次画面へ遷移)

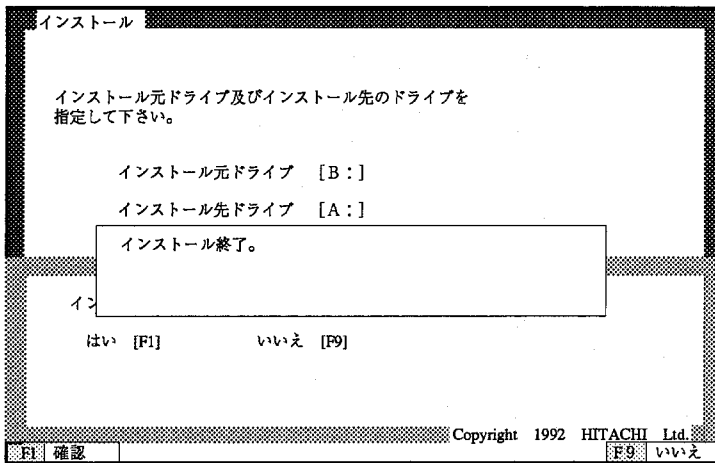
●システムインストールを中止する

- ・ **[F9]** (いいえ) キーを押してください。(前画面に戻る)



◆システムインストール中

- ・インストール中はこの画面が表示されます。
- ・インストールが正常終了すると次画面に遷移します。



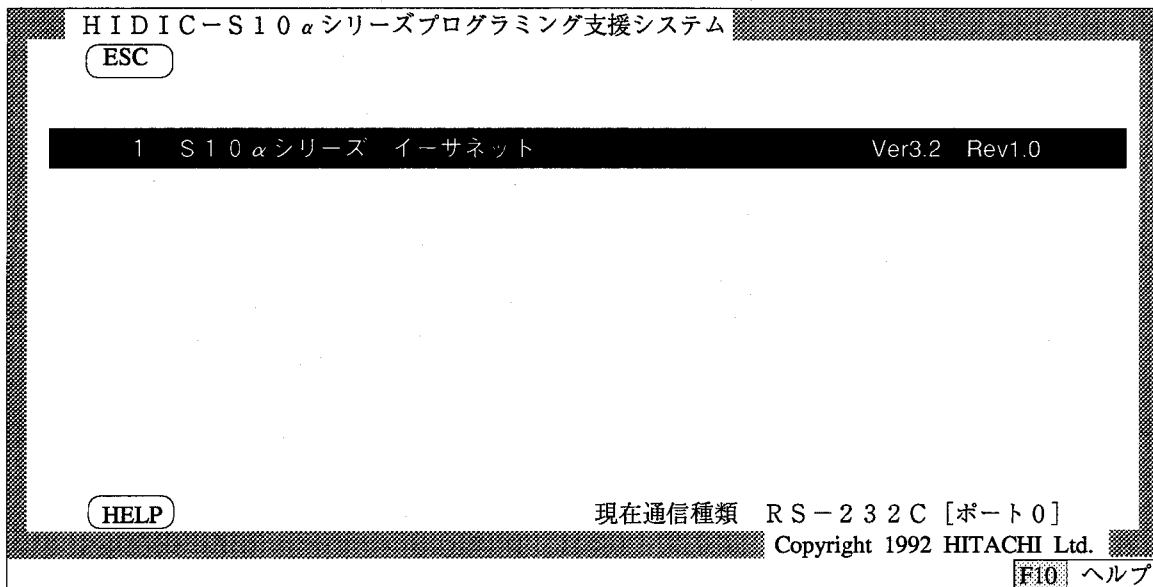
◆システムインストール終了




- ・ **F1** (確認) キーを押すと、インストールプログラムが終了します。

## 8.2 システム立上げ

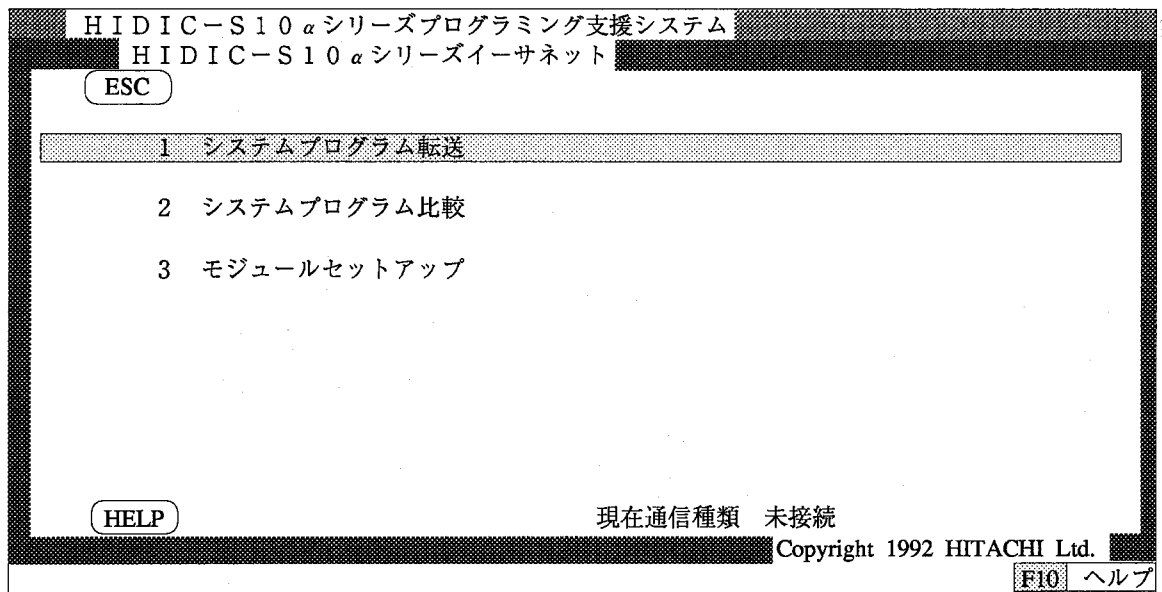
### 8.2.1 立上げ手順

- (1) パソコンを起動します。
- (2) MS-DOSが立上がった状態（コマンドプロンプトが表示された状態）になります。
- (3) CD ¥HITACHI¥S10と入力後、[RETURN]キーを押してください。
- (4) S10MENUと入力後、[RETURN]キーを押してください。
- (5) プログラミング支援システムが起動し、以下の初期画面が表示されます。



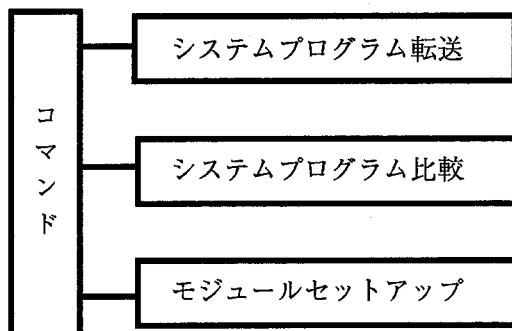
イーサネットシステムの立上げは、プログラミング支援システム初期画面でマウスカーソルをS10 α シリーズイーサネットへ移動後、マウスの左ボタンを押すか、キーボードの 、 キーを使ってカーソルをS10 α シリーズイーサネットへ移動後、 キーを押してください。

プログラミング支援システムにより立上げられたイーサネットシステムは次のような初期画面になります。この状態でイーサネットシステムは立上っています。この後は、目的のコマンドを選択してください。



### 8.2.2 コマンド体系

イーサネットシステムにおけるコマンドの体系を以下に示します。



コマンド体系に示したコマンドの概要と、コマンドを実行する際の確認事項および、コマンドの詳細な操作ガイドを8.3節～8.5節に示します。

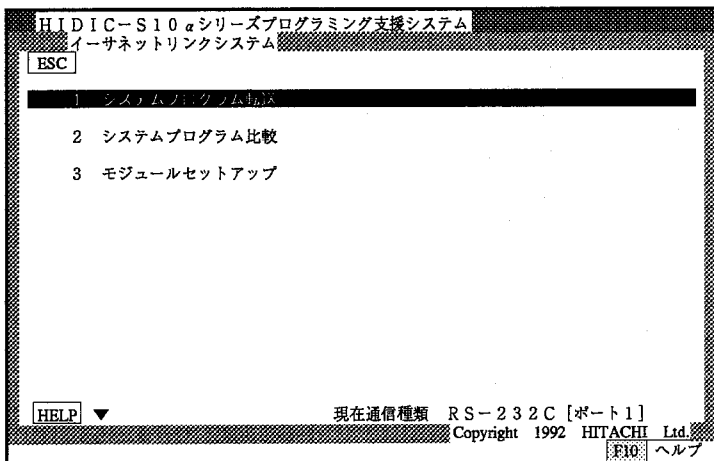
## 8.3 S10ET LINKシステムプログラム転送

●機能：「S10ET LINKシステムプログラム」をCPU (PCs) 本体とS10ET LINKモジュールへ転送します。

●確認項目：システムプログラム転送を実行する前に、必ず以下の項目を確認してください。

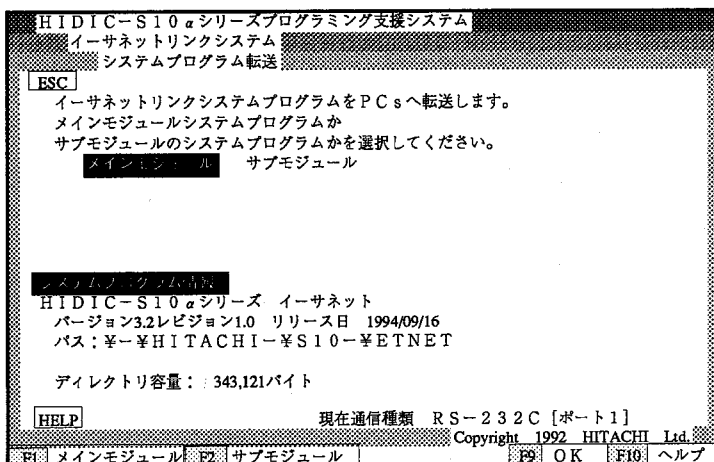
- ・ S10ET LINKモジュールが正しく実装されているか（「1 ご使用にあたり」を参照）。
- ・ CPUの状態が“STOP”でありプロテクトが“OFF”になっているか（CPUモジュールのキースイッチを該当する位置へ合わせる）。
- ・ ツールとCPU (PCs) が回線で接続されているか（画面に接続状態が表示してあります。未接続ならば、「2α、4αシリーズ ソフトウェアマニュアル オペレーションプログラミング支援システム For PC 98」（SAJ-3-014）のユーティリティ／通信接続を参照してください）。

●オペレーション：



●システムプログラムを転送したい

- ・ **↑**、**↓** キーかマウスを使ってメニューカーソルをシステムプログラム転送へ移動してください。
- ・ **改行** キーを押すか、マウスの左ボタンを押してください（次画面へ遷移）。



●システムを転送するモジュールを選択する

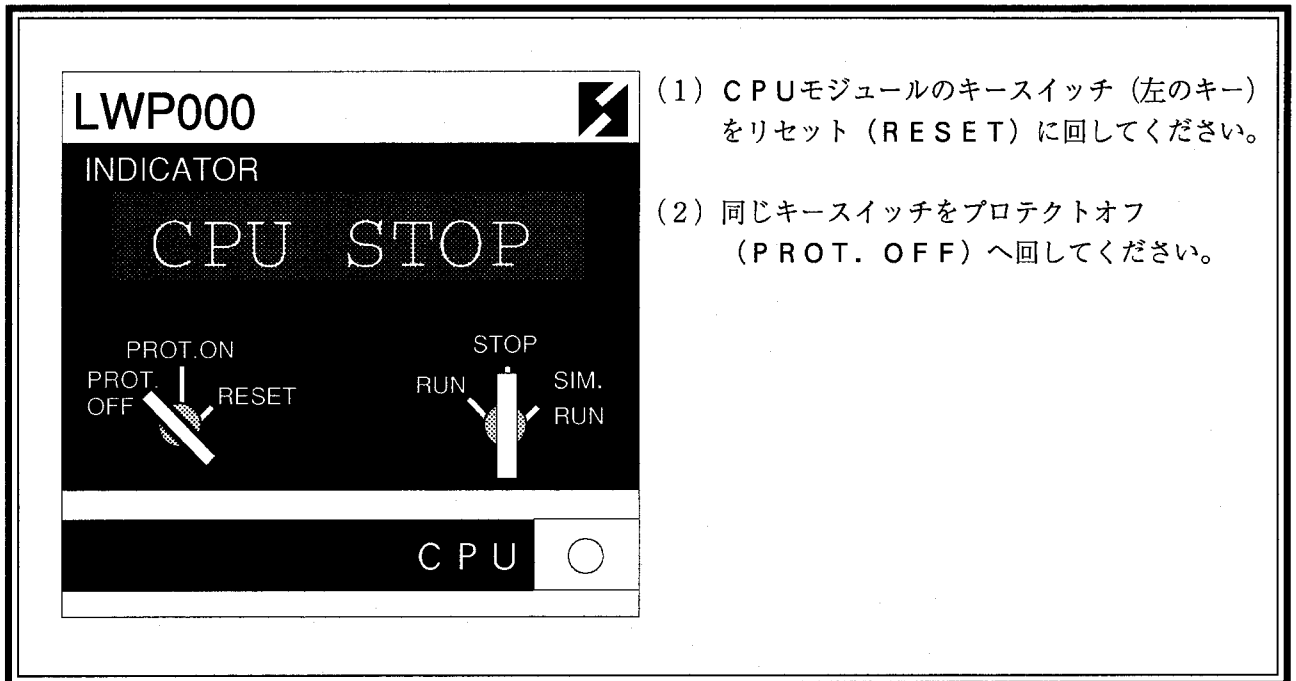
- ・ **F1** (メインモジュール)、または **F2** (サブモジュール) を押してください。

●システムプログラム転送を開始する

- ・ **F9** (OK) キーを押してください。



- CPUリセット処理：イーサネットシステムでは、システムプログラムを転送するだけでは、システムを実行させることはできません。CPUリセット処理が必要です。リセット処理は、次の手順で行ってください。



- 正常終了時のメッセージ：  
転送終了しました。
- 操作中にエラーが出たとき：9.2節の「パソコンプログラミングエラーコード表」参照

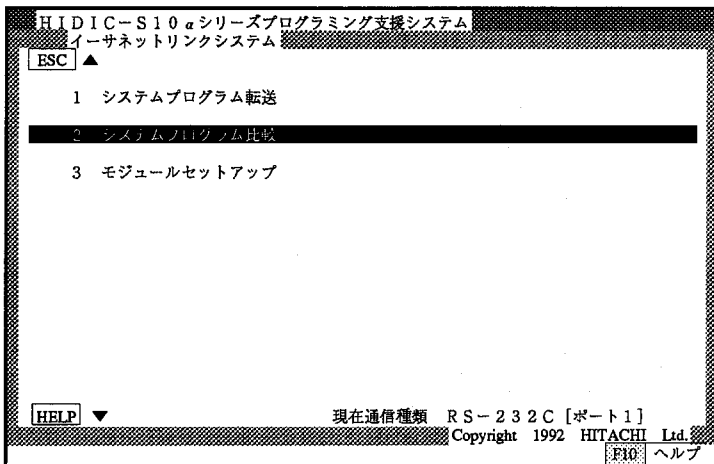
## 8.4 S10ET LINKシステムプログラム比較

●機能：「S10ET LINKシステムプログラム」とCPU (PCs) 本体とS10ET LINKモジュール内のシステムプログラムを比較します。比較した結果、2つのプログラムが違う場合、プログラムの内容を表示します。

●確認項目：システムプログラム比較を実行する前に必ず以下の項目を確認してください。

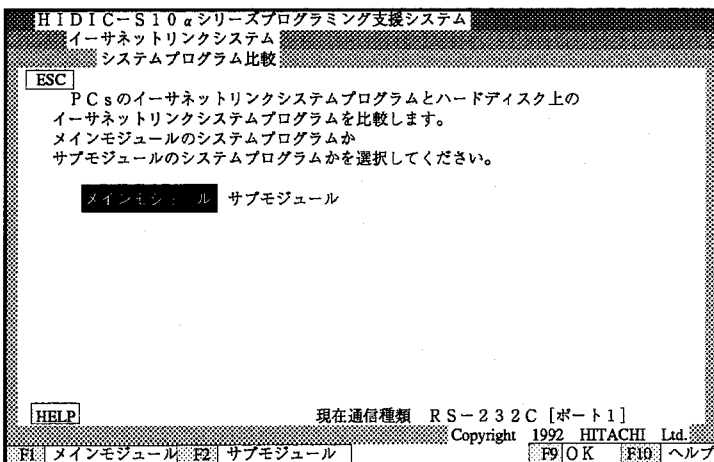
- ・S10ET LINKモジュールが正しく実装されているか（「1 ご使用にあたり」を参照）。
- ・ツールとCPU (PCs) が回線で接続されているか（画面に接続状態が表示してあります。未接続ならば、「2 a、4 aシリーズ ソフトウェアマニュアル オペレーションプログラミング支援システム For PC 98」（SAJ-3-014）のユーティリティ／通信接続を参照してください）。

●オペレーション：



●システムプログラムを比較したい

- ・**↑**、**↓** キーかマウスを使ってメニューカーソルをシステムプログラム比較へ移動してください。
- ・**改行** キーを押すか、マウスの左ボタンを押してください（次画面へ遷移）。



●システムを比較するモジュールを選択する

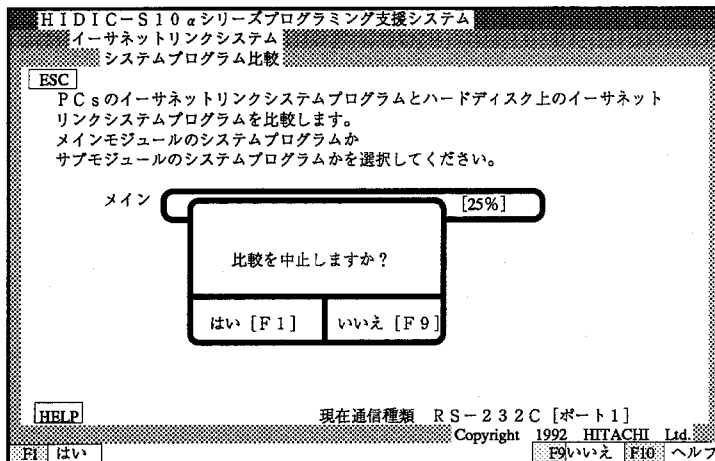
- ・**F1**（メインモジュール）、または**F2**（サブモジュール）を押してください。

●システムプログラムの比較を開始する

- ・**F9**（OK）キーを押してください。

●比較中に比較を中断する

- ・**ESC** キーを押してください。（次画面へ遷移）

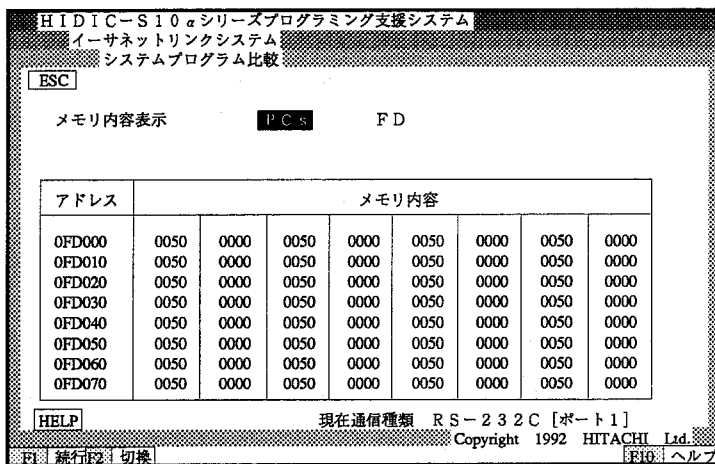


## ● 比較を中止する

- ・ **F1** (はい) キーを押してください (前画面に戻る)。

## ● 比較を中止しない

- ・ **F2** (いいえ) キーを押してください (次画面へ遷移)。



## ● 比較を続行したい

- ・ **F1** (続行) キーを押してください。

## ● 対象マシンをPCsからソール、ソールからPCsへ切り換えたい

- ・ **F2** (切換) キーを押してください。

(注)

比較内容が異なる場合、無条件でこの画面が表示されます。

## ● 正常終了時のメッセージ:

比較を中止した場合: 比較を中止しました。

比較結果が同じ場合: 比較内容は同じです。

比較結果が違う場合: 比較内容は違います。

## ● 操作中にエラーが出たとき: 9.2節の「パソコンプログラミングエラーコード表」参照

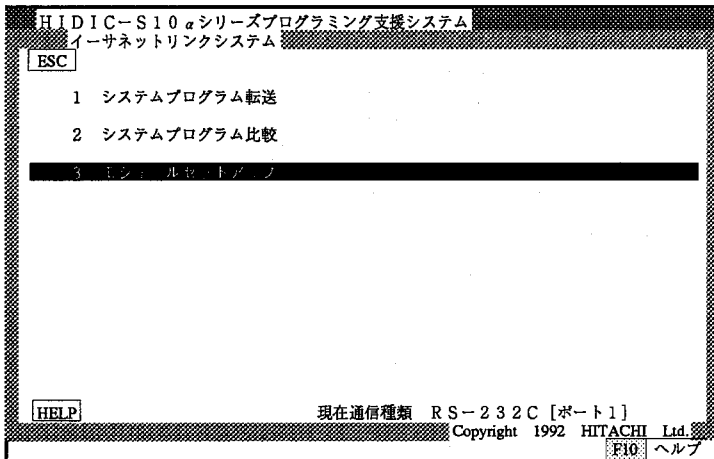
## 8.5 S10ET LINKモジュールセットアップ

●機能：このコマンドは、S10ET LINKを使って行う通信に必要な情報をS10ET LINKモジュールへ設定します。情報には、ホスト名、IPアドレスがありますので、両方とも設定してください。

●確認項目：ステーション情報テーブル編集を実行する前に必ず以下の項目を確認してください。

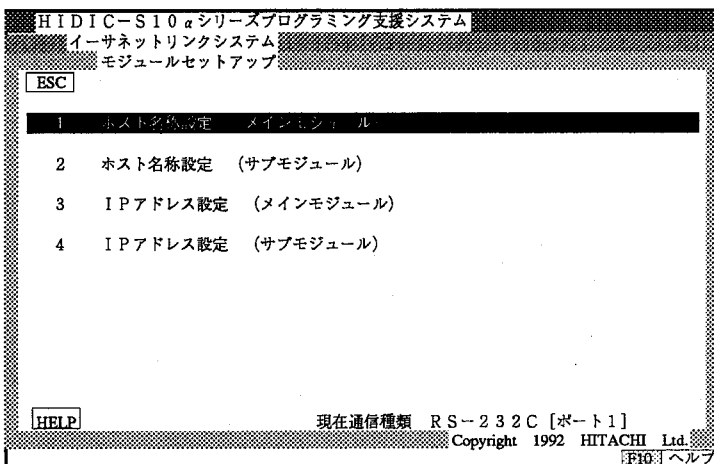
- ・S10ET LINKモジュールが正しく実装されているか（「1 ご使用にあたり」を参照）。
- ・CPUの状態が“STOP”でありプロテクトが“OFF”になっているか（CPUモジュールのキースイッチを該当する位置へ合わせる）。
- ・ツールとCPU（PCs）が回線で接続されているか（画面に接続状態が表示してあります。未接続ならば、「2α、4αシリーズ ソフトウェアマニュアル オペレーションプログラミング支援システム For PC 98」（SAJ-3-014）のユーティリティ／通信接続を参照してください）。

●オペレーション：



●ステーション情報テーブルを編集したい

- ・、 キーかマウスを使ってメニューカーソルをモジュールセットアップへ移動してください。
- ・ キーを押すか、マウスの左ボタンを押してください（次画面へ遷移）。

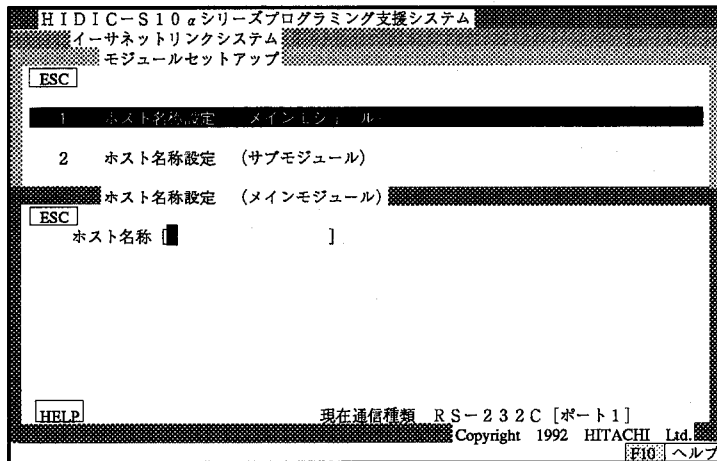


●ホスト名称を設定する

- ・、 キーかマウスを使ってメニューカーソルをホスト名称設定へ移動してください。
- ・ キーを押すか、マウスの左ボタンを押してください（次画面へ遷移）。

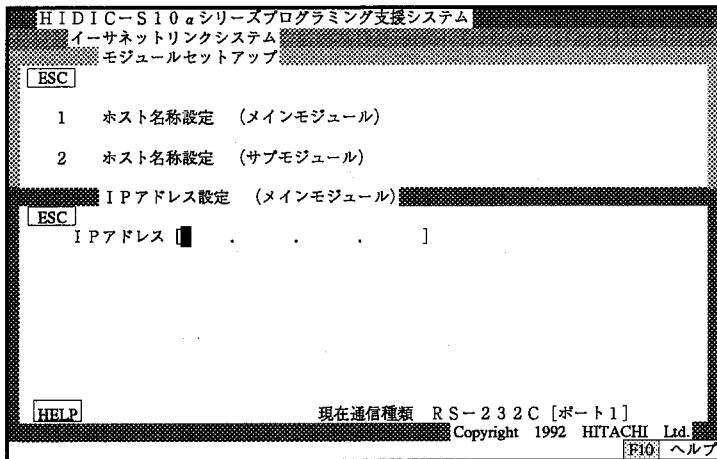
●IPアドレスを設定する

- ・、 キーかマウスを使ってメニューカーソルをIPアドレス設定へ移動してください。
- ・ キーを押すか、マウスの左ボタンを押してください（次ページの2番目の画面へ遷移）。



●ホスト名称を設定する

- ・ホスト名称を入力してください。
- ・ **改行** キーを押してください。



●IPアドレスを設定する

- ・IPアドレスを入力してください。
- ・ **改行** キーを押してください。

●IPアドレスは、10進数を3桁ごとに“.”(ピリオド)”で区切って表現します。

MMM. NNN. XXX. YYY

(MMM, NNN, XXXは0~255の10進数)  
(YYYは1~254の10進数)

既設のネットワーク上に、このマシンを追加登録する場合は、そのネットワークに合ったIPアドレスとしてください。

新設の場合は下記をおすすめします。

MMM=192 ..... クラスCを選択  
 NNN=001 } ..... ネットワーク番号  
 XXX=000 }  
 YYY=001~200 ..... ステーションごとの一貫No.

●操作中にエラーが出たとき：9.2節の「パソコンプログラミングエラーコード表」参照



## 9 エラーと対策

**9.1 PSE $\alpha$ のエラーと対策**

S10ET LINKサポートシステムにおいて追加されたエラーコードを示します。

PSEエラーコード表

エラーコード (ERR)	内容および原因	対 策
15	オプションモジュールが実装されていません。	オプションモジュールを実装してください。
16	オプションモジュール用システムプログラムがローディングされていません。	オプションモジュール用システムプログラムをローディングしてください。
17	指定された機能は、該当するオプションモジュール用システムプログラムにはありません。	入力した内容を確認してください。



## 9.2 パソコンプログラミングのエラーと対策

パソコンプログラミング ETリンクシステムにおけるエラーメッセージの一覧と対策を示します。

パソコンプログラミングエラーコード表

エラーコード	エラーメッセージ	回避策
#001	ファイルを読み込めません。	プログラミングツール上にあるファイルが壊れている可能性があります。確認後、再実行してください。
#004	ファイルが存在しません。	プログラミングツール上にあるファイルが失われている可能性があります。確認後、再実行してください。
#700	回線エラー	ツールとPCs間の回線を確認後、再実行してください。
#900	PCsタイプ不一致エラー	PCsタイプ(S10/2α、S10/2αE...)を確認後、再実行してください。
#901	オプションモジュールが実装されていません。	PCs側のオプションモジュールの実装とモジュールの設定を確認後、再実行してください。
#903	PCs No.不一致エラー	PCsのインジケータに表示されるNo.と一致するファイルを実行してください。ただし、プログラミングツール上にあるファイルが壊れている可能性もあります。確認してください。
#904	PCs RUN中エラー	PCsのキースイッチをSTOPにして再実行してください。
#908	PCs PROT. ON中メモリの書換はできません。	PCsのキースイッチをPROT. OFFにして再実行してください。

### 9.3 CPU LED表示メッセージ

#### 9.3.1 CPUのエラー表示と対策

S10ET LINKモジュールが異常時に表示するエラーメッセージと対策を示します。

モジュール	表示内容	内 容	対 策
メイン	ETLM□WDT	ウォッチドッグタイマタイムアウトエラーが発生した。	CPUキースイッチを1度RESETにし、元に戻してください。
サブ	ETLS□WDT		
メイン	ETLM□PTY	RAMにパリティエラーが発生した。	それでも再び表示される場合は、7.2節により再立上げしてください。
サブ	ETLS□PTY		
メイン	ETLM□PRG	システムプログラムに異常が発生した。	それでも再び表示される場合は、S10ET LINKを交換してください。
サブ	ETLS□PRG		
メイン	ETLMLOAD	システムプログラムがクリアされた。	S10ET LINKにシステムプログラムをローディングしてください。
サブ	ETLSLOAD		
メイン	EXD0□CEL	バッテリー異常が発生した。	S10ET LINKを交換してください。
サブ	EXD1□CEL		
メイン	EXD0□PTY	CPU（またはPSE）からRAMを読出した時、パリティエラーが発生した。	CPUキースイッチを1度RESETにし、元に戻してください。それでも再び表示される場合は、7.2節により再立上げしてください。
サブ	EXD1□PTY		

□：スペースを示します。

#### 9.3.2 立上げメッセージ

S10ET LINKは正常時に下記のメッセージを表示します。

モジュール	表示内容	内 容
メイン	ETL1□X.Y	正常時の立上げメッセージ
メイン&サブ	ETL3□X.Y	

□：スペースを示します。

X：バージョンNo.を示します。

Y：レビジョンNo.を示します。

## 9.4 ソケットライブラリのエラーと対策

ソケットライブラリのエラーコードと対策について、以下に示します。

コード	リテラル	内 容	説 明	対 策
0x0001	EILLFNC	イリーガル機能を使用しました。	ライブラリの発行手順が違っています。	ライブラリ発行手順に従って発行してください。
0x0002	ENOBUFS	バッファが不足しています。 コマンドに対してリソースが不足しています	オプションモジュール1台で同時に11個以上のソケットを使用しようとした。	使用していないソケットをshutdown( )してください。
0x0003	EOPNOTSUPP	パラメータが異常です。	lanpnパラメータが1, 2以外か、sa_familyパラメータが1, 2以外になっています。	パラメータを確認してください。
0x0004	ECONNALRDY	TCPコネクションは既に確立されています。	listen( )とaccept( )または、connect( )のいずれかを、既に発行しています。	listen( ), accept( ), connect( )のいずれかのうち2回以上発行した関数をチェックしてください。
0x0005	ETIMEOUT	TCPのタイムアウトエラーが起きました。	相手局から応答がきません(ネットワーク上のハードウェア故障)。	相手ステーション、トランシーバケーブルコネクタなどをチェックしてください。
0x0007	ENOTCONN	TCPコネクションが確立されていません。	listen( )とaccept( )またはconnect( )でTCPコネクションを確立していません。	listen( )とaccept( )またはconnect( )を発行してください。
0x0008	ECONNFAIL	TCPコネクション確立に失敗しました。	accept( )かconnect( )の発行でタイムアウトになりました。	accept( )かconnect( )のいずれかを再発行してください。
0x000B	ENOTSOCK	対応するソケットが存在しません。	sa_familyパラメータが1, 2以外になっています。	パラメータをチェックしてください。

コード	リテラル	内 容	説 明	対 策
0x000C	ESOCKALRDY	TCP/IPおよびUDP/IPのソケットは既に登録済です。	TCP/IPおよびUDP/IPで自局ポート番号、相手局ポート番号、相手局ホスト名の同じものが既に登録されています。	パラメータのポート番号、相手局ホスト名をチェックしてください。
0x000D	EHOSTS	対応するホスト名が設定されていません。	socket( )のパラメータで渡されたホスト名がHOSTSテーブルに設定されていません。	パラメータで指定されたホスト名をチェックしてください。
0x000F	EILLPORT	ポート番号が異常値です。	無効なポート番号を指定しました。	パラメータのポート番号をチェックしてください。
0x0010	EINIT	S10ET LINKモジュール初期化を失敗しました。	初期化情報が破壊されています。	「S10ET LINKシステムプログラム」を再ローディングしてください。
0x0011	ENOTTERM	S10ET LINKはまだ終了処理が実行されていません。	term( )が発行されていません。	term( )を発行してください。
0x0012	ETERM	S10ET LINKモジュールの終了処理を失敗しました。	システムプログラムが破壊されています。	S10ET LINKモジュールを7.2節により再立上げしてください。
0x0013	ENOTINIT	S10ET LINKモジュールを初期化していません。	init( )が発行されていません。	init( )を発行してください。
0x0014	EILLBOARD	S10ET LINKモジュールに異常が発生しました。	S10ET LINKモジュールが故障しました。	S10ET LINKモジュールを交換してください。
0x0015	ESYSTEM	S10ET LINKにシステムエラーが発生しました。	システムプログラムが破壊されています。	S10ET LINKモジュールを7.2節により再立上げしてください。
0x0017	EILLPARAM	パラメータが異常です。	ソケットライブラリ関数のパラメータが異常です。	発行した関数のパラメータをチェックしてください。

コード	リテラル	内 容	説 明	対 策
0x0018	ERSPTIMEOUT	レスポンス待ちタイムアウトが起きました。	S10ET LINKモジュールの応答待ちでタイムアウトが起きました (相手ステーションの送受信プログラムが起動されていないか、またはネットワーク上のハードウェアが故障しています。)	相手ステーションの送受信プログラムを起動してください。 相手ステーション、自ステーション、トランシーバ、ケーブル、コネクタなどをチェックしてください。
0x0019	ECHKSUM	チェックサムエラーが起きました。	システムプログラムが破壊されました。	7.2節により再立上げしてください。
0x001A	EICMP	プロトコルエラーが起きました。	システムプログラムが破壊されました。	7.2節により再立上げしてください。
0x001B	EADDR	相手局とコネクションできません。	当該IPアドレスのステーションが存在していません。	HOSTSテーブルのIPアドレスをチェックしてください。
0x001C	ENOTFILE	TCP/IPまたはUDP/IPのテーブルが異常です。	TCP/IPまたはUDP/IPのテーブルのデータが破壊されています。	「S10ET LINKシステムプログラム」を再ローディングしてください。
0x001E	EILLINIT	モジュールのセットアップ情報が異常です。	モジュールがセットアップされていません。	7.5節によりモジュールをセットアップしてください。
0x0021	EUNREMOTE	相手局が不明です。	相手局未指定のソケットに対し、connect( )を発行しました。	socket( )で相手局を指定してください。
0x0022	ETRANS	送信エラーが起きました。	S10ET LINKモジュールトランシーバケーブル、トランシーバの異常です。	S10ET LINKモジュールトランシーバケーブル、トランシーバをチェックしてください。
0x0023	ECONNSHUT	TCPコネクションが切断されました。	TCPコネクションが強制終了されました。	shutdown( )を発行し、socket( )から再発行してください。

コード	リテラル	内 容	説 明	対 策
0x0024	EALOCALRDY	UDP/IPで受信要求中に再度受信要求を行いました。	システムプログラムが破壊されました。(UDP/IPの送受信の要求は1対1であり、システムプログラムが要求管理を行っています。)	S10ET LINKシステムプログラムを再ローディングしてください。
0x0025	EUDPALRDY	UDP/IPのソケットは既に登録済です。	UDP/IPに自局ポート番号と同じものが既に登録されています。	socket( )のパラメータのポート番号をチェックしてください。
0x0026	ENETID	ネットワーク番号が不一致です。	相手局IPアドレスのクラス、ネットワーク番号が自局IPアドレスと合いません。	相手局IPアドレスと自局IPアドレスをチェックしてください。
0x0027	EBROADCAST	相手局IPアドレスがブロードキャストアドレスです。	TCP/IPで相手局IPアドレスをブロードキャストアドレスに指定しています。	socket( )のパラメータのソケットタイプ、相手局IPアドレスをチェックしてください。
0x7FFF	EUNKWN	未定義コードが返ってきました。	S10ET LINKモジュールから未定義コードが返ってきました。	shutdown( )を発行し、socket( )から再発行してください。

# 10 付 録

## 10.1 ネットワーク構成部品

### 10.1.1 LWE400とイーサネット<sup>\*</sup>との接続の問題点

LWE400は、国際標準であるIEEE802.3規格に準拠している標準仕様品です。

しかし同一規格に準拠した異社間のトランシーバおよびリピータなどを組合せた場合、相性によってうまく動作しない場合があります。

したがって、LWE400では、トランシーバ、リピータ、同軸ケーブル、コネクタおよびターミネータはすべて当社の推奨する機器を使用してください。

イーサネットの使用にはIEEE802.3規格とオリジナルイーサネット仕様とがあります。LWE400にオリジナルイーサネット仕様の機器を接続することはできませんので注意してください。

\*イーサネットはXerox社の登録商標です。

### 10.1.2 構成品一覧表

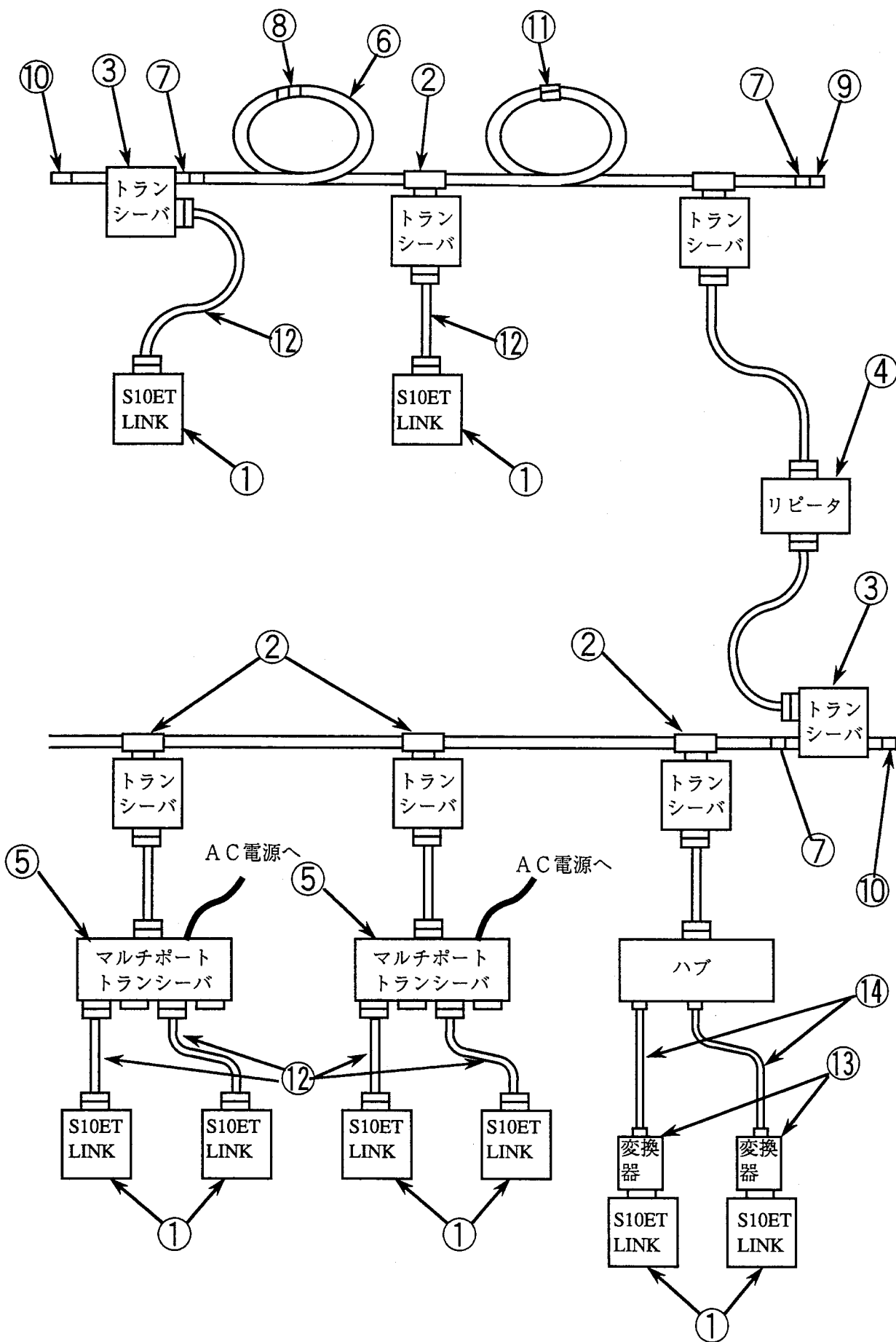
	品 名	メ ー カ	形 式	備 考
①	S10ET LINK	(株)日立製作所	LWE400	H-S10/2αに実装されるIEEE802.3準拠LAN制御装置
②	トランシーバ	日立電線(株)	HLT-200TB	タップ形トランシーバ
③	トランシーバ	日立電線(株)	HLT-200	コネクタ形トランシーバ
④	リピータ	日立電線(株)	HLR-200H	同軸ケーブルの伝送距離延長用リピータ装置
⑤	マルチポートトランシーバ	日立電線(株)	H-7612-64/ -68	4ポート/8ポートトランシーバ。 AC電源内蔵
⑥	同軸ケーブル	日立電線(株)	HBN-CX-100	屋内用、ケーブル長指定(最大長 500m)
⑦	同軸コネクタ	日立電線(株)	HBN-N-PC	同軸ケーブル用コネクタ
⑧	中継コネクタ	日立電線(株)	HBN-N-AJJ	同軸ケーブル用中継コネクタ
⑨	ターミネータ	日立電線(株)	HBN-T-NJ	J形
⑩	ターミネータ	日立電線(株)	HBN-T-NP	P形
⑪	アース端子	日立電線(株)	HBN-G-TM	同軸ケーブル用アース端子
⑫	トランシーバケーブル	日立製作所	HDC4360	オス、メスDsub 15Pinコネクタ付 最大長15m
⑬	変換器	日立電線(株)	HSN-9010	10BASE 5/T変換器
⑭	ツイストペアケーブル	日立電線(株)	HUTP-CAT5 4P	ツイストペアケーブル



### 強 告

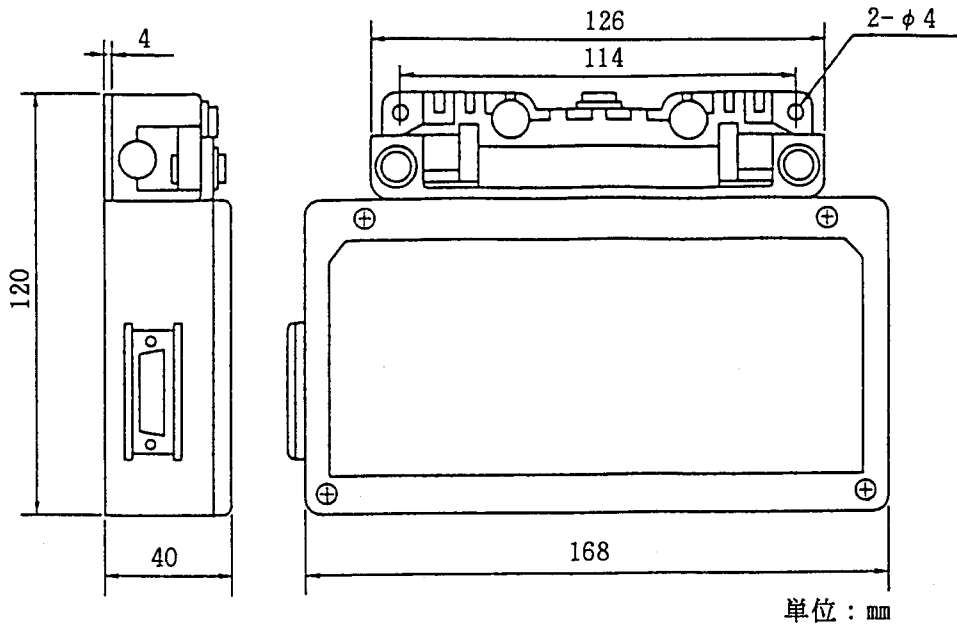
10BASE 5/T変換器は、必ず弊社指定の変換器(形式:HSN-9010、メーカー:日立電線製)を使用してください。他の変換器をご使用になるとノイズなどの影響により極端に性能がおちる可能性があります。





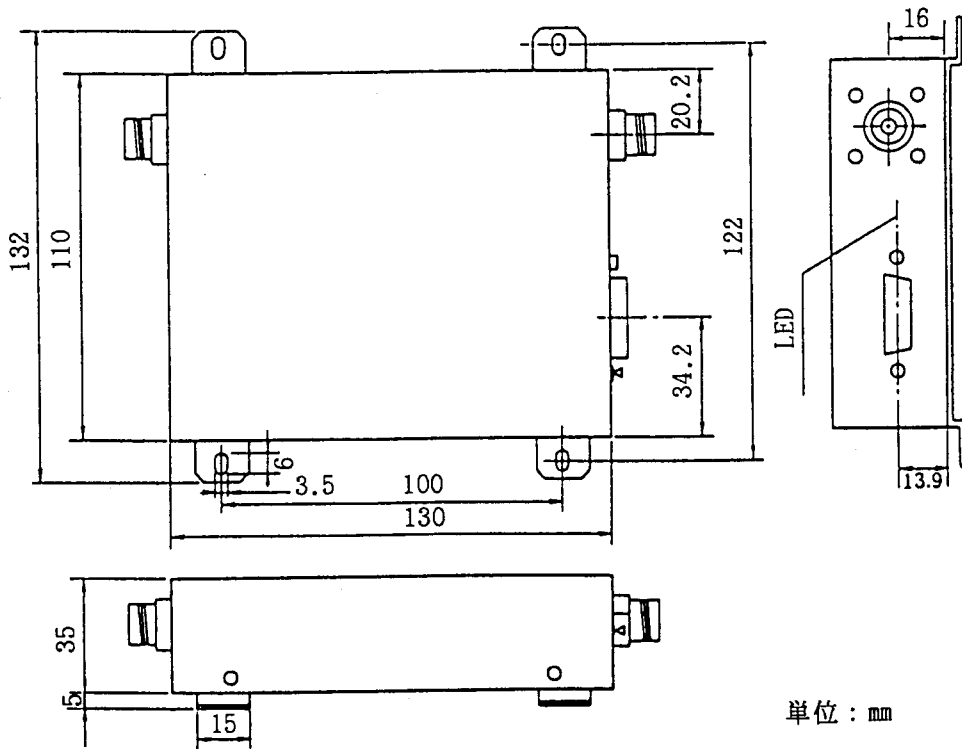
10.1.3 トランシーバ (タップ形) HLT-200TB

トランシーバ (タップ形) の外観を示します。



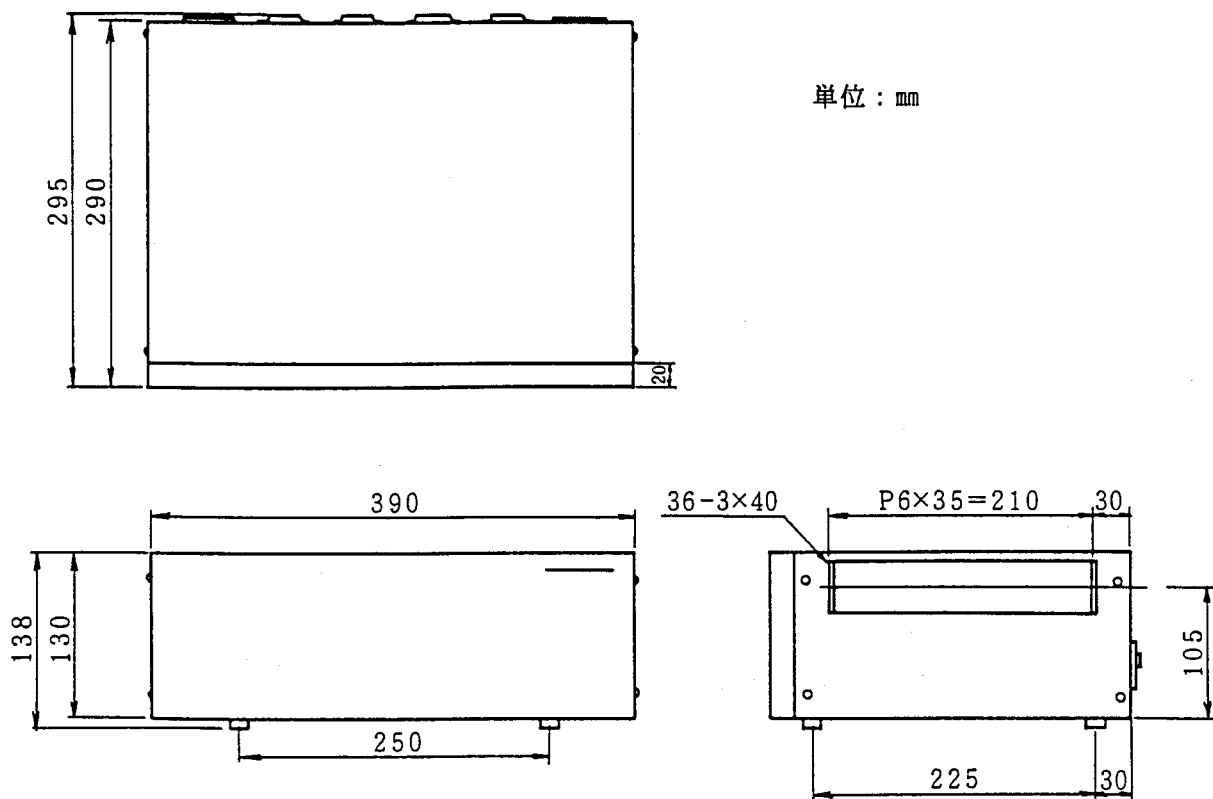
10.1.4 トランシーバ (コネクタ形) HLT-200

トランシーバ (コネクタ形) の外観を示します。



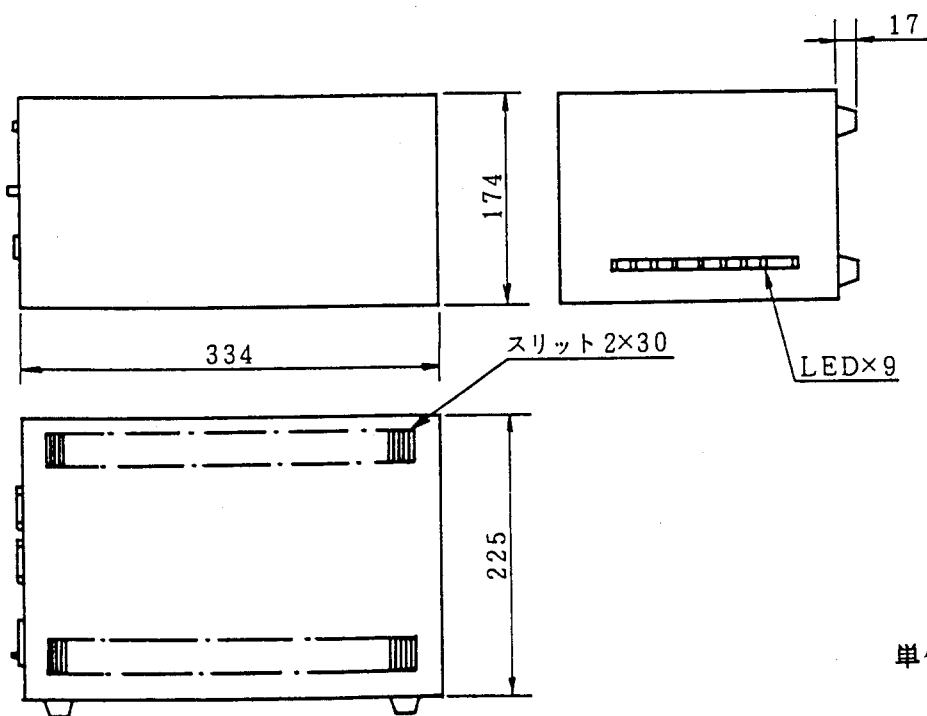
10.1.5 マルチポートトランシーバ H-7612-64

マルチポートトランシーバの外観を示します。



10.1.6 リピータ HLR-200H

リピータの外観を示します。



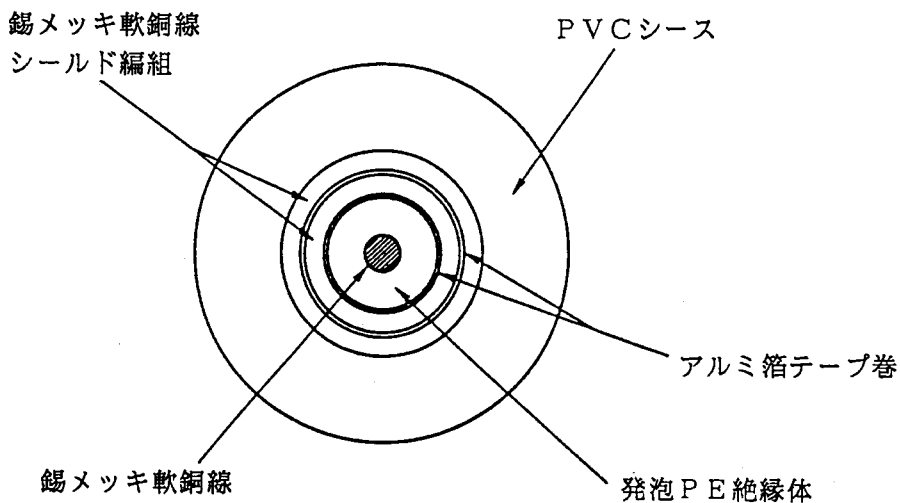
10.1.7 同軸ケーブル（屋内用） HBH-CX-100

同軸セグメント用ケーブルの構造および特性を以下に示します。

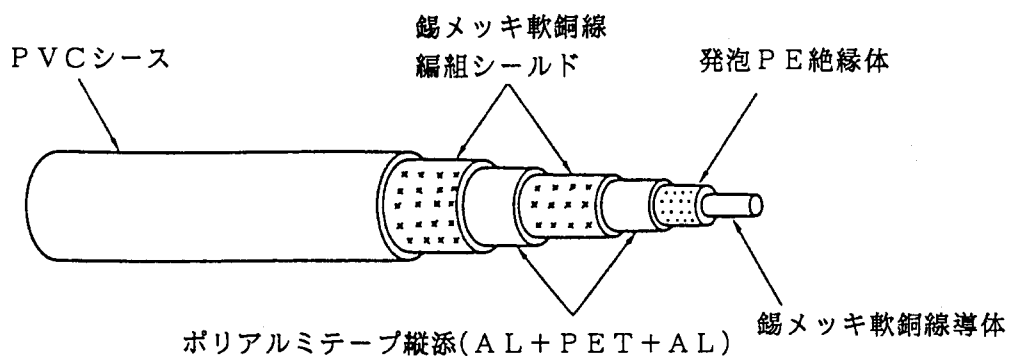
同 軸 ケ ー ブ ル の 構 造

線 心 種 類		50Ω高周波発泡PE同軸ケーブル
錫メッキ軟銅線導体	断面積 mm <sup>2</sup>	3.7(12AWG)
	構 成 本/mm	1/2.17
	外 径 mm	2.17
絶 縁 体	材 質	発泡PE
	厚 さ mm	2.015
	外 径 mm	6.2
	色 別	自然色
アルミ箔テープ巻	外 径 mm	6.35 ※1
シ ー ル ド	構 成	0.18Tmm×7持×24打
	外 径 mm	7.25
アルミ箔テープ巻	外 径 mm	7.4 ※1
シ ー ル ド	構 成	0.18Tmm×7持×24打
	外 径 mm	8.3
P V C シ ー ス	厚 さ mm	1.0
	外 径 mm	10.0~10.8
	色 別	黄

※1 アルミ箔テープは、縦添とします。 T:スズメッキ軟銅線



同軸ケーブルの構造



同軸ケーブルの構造外観

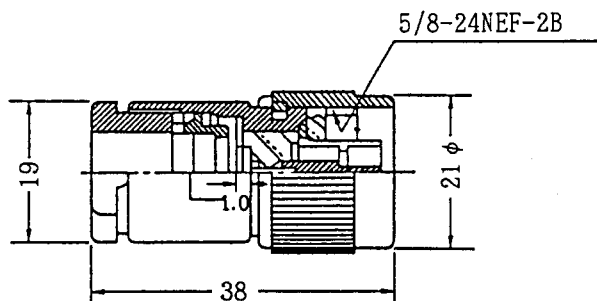
電 気 的 特 性

項 目		特 性 値
特性インピーダンス		50 ± 2 Ω
伝 播 速 度		0.77C* 以上
伝播インピーダンス		IEEE 802.3仕様に適合
減衰量	5 MHz	6.0 dB / 500 m 以下
	10 MHz	8.5 dB / 500 m 以下

\* Cは光速を示します。

10.1.8 同軸コネクタ HBN-N-PC

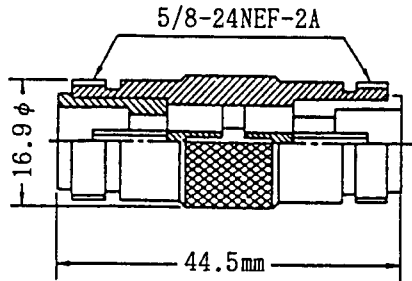
同軸ケーブル用のコネクタの外観を示します。



単位：mm

10.1.9 中継コネクタ HBN-N-AJJ

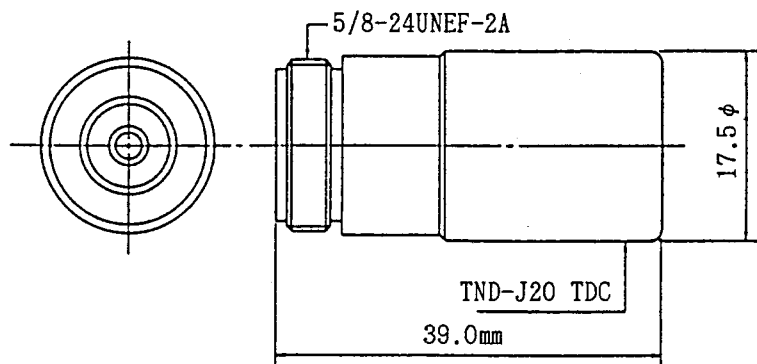
同軸ケーブル用の中継コネクタの外観を示します。



10.1.10 ターミネータ (J形) HBN-T-NJ

同軸セグメントの端末を終端するためのターミネータ (J形) の外観を示します。

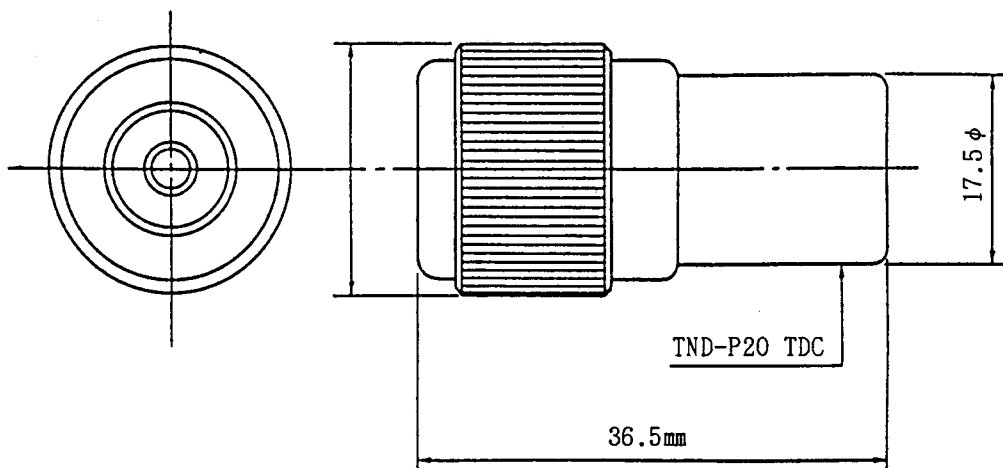
(同軸ケーブルの終端に使用します。)



10.1.11 ターミネータ (P形) HBN-T-NP

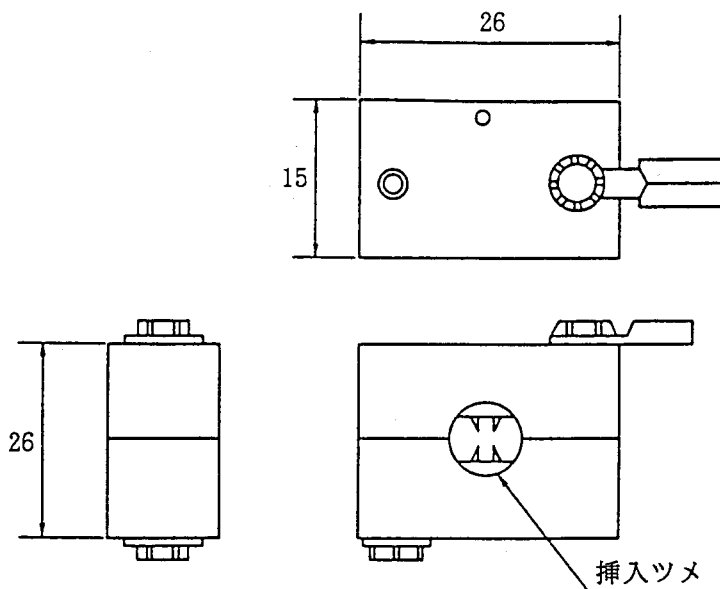
同軸セグメントの端末を終端するためのターミネータ (P形) の外観を示します。

(トランシーバ コネクタ形の終端に使用します。)



10.1.12 アース端子 HBN-G-TM

同軸セグメントを接地するためのアース端子の外観を示します。



単位：mm

HBN-TC-100

10.1.13 トランシーバケーブル

トランシーバとコントローラおよびトランシーバとリピータを接続するトランシーバケーブルの外観およびピン配置を示します。必ずシールドアース配線のできるケーブルをご使用ください。（弊社製作の専用ケーブル 型式：HDC4360を推奨いたします）アース用配線のないケーブルでは、ノイズなどの影響によりエラーが発生する可能性があります。

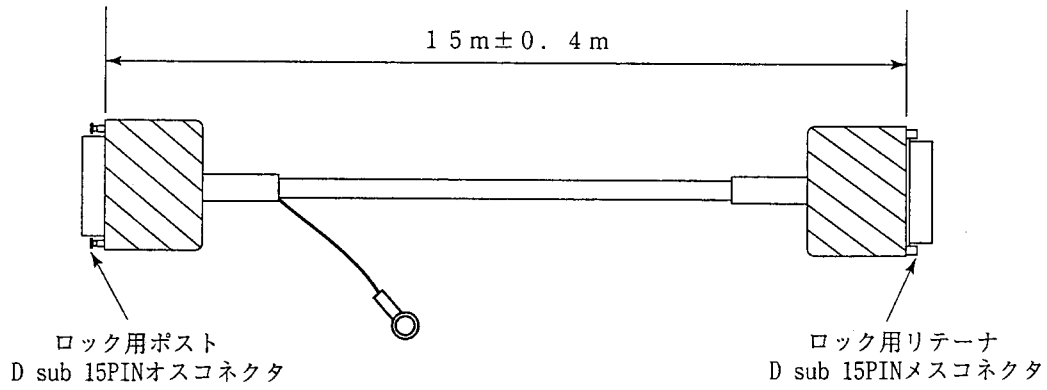
トランシーバケーブルの構造

線 心 種 類		信 号 線	電 源 線
線 心 番 号		1～6	7～8
錫メッキ軟銅線導体	断面積 mm <sup>2</sup>	22AWG	20AWG
	構成本/mm	7/0.26	7/0.32
	外径 mm	0.78	0.96
絶 縁 体	材質	架橋発泡 PE	半硬質 PVC
	厚さ mm	0.41	0.27
	外径 mm	1.6	1.5
燃 り	外径 mm	3.2	3.0
シ ー ル ド	構成	ALテープ巻	ALテープ巻
	外径 mm	3.4	3.2
燃 り 合 せ	外径	6.6	
錫メッキ軟銅線シールド	構成	0.14Tmm×9持×24打	
	外径 mm	7.3	
P V C シ ー ス	厚さ mm	1.0	
	外径 mm	9.3±0.5	
	色 別	灰	

T：スズメッキ軟銅線

〈コネクタ〉

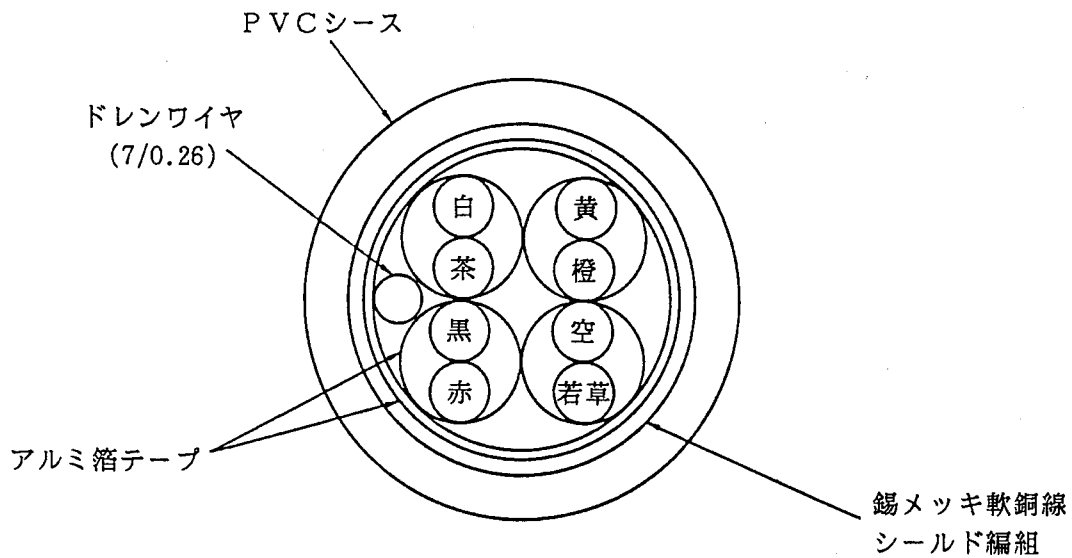
標準長トランシーバケーブルのコネクタ仕様を下図に示します。



トランシーバケーブルのコネクタ仕様

〈ケーブル断面図〉

ケーブル断面図を下図に示します。



ケーブル断面図



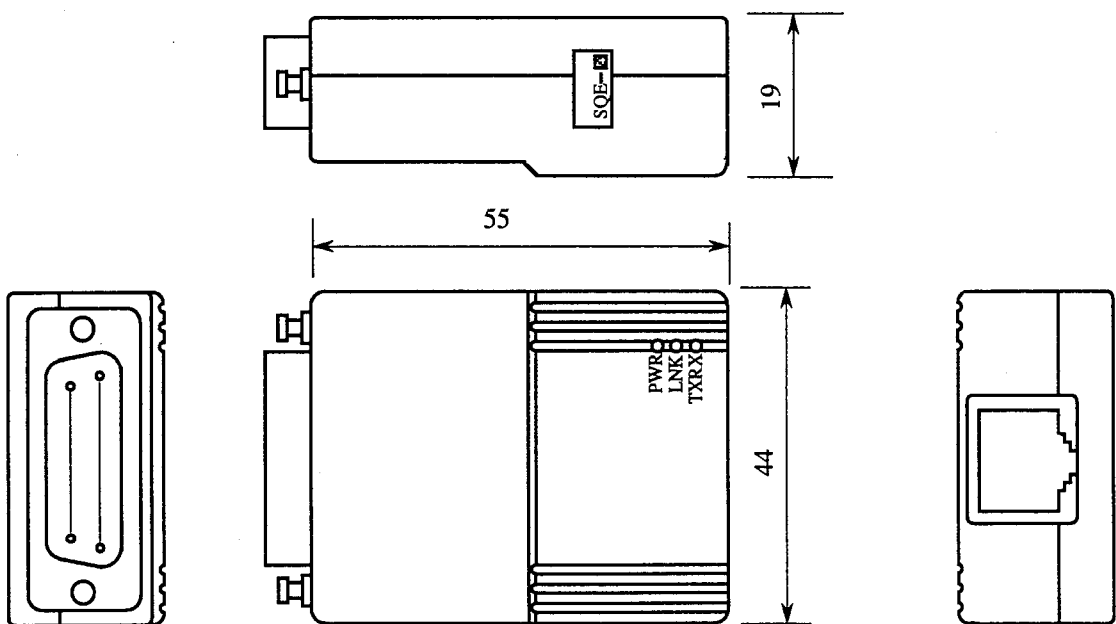
コネクタのピン配置を以下に示します。

トランシーバケーブルのピン配置

P I N No.	信 号 名	心 線 名	P I N No.	信 号 名	心 線 名
1	シールド	(ドレンワイヤ)			
2	衝突検出+	茶	9	衝突検出-	白
3	送 信+	空	10	送 信-	若草
4	—	—	11	—	—
5	受 信+	橙	12	受 信-	黄
6	電源リターン	赤	13	電 源	黒
7	—	—	14	—	—
8	—	—	15	—	—

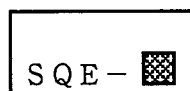
10.1.14 変換器 HSN-9010

変換器の外観図を示します。



**強 制**

ET. NETに取り付けの際は、SQEスイッチの設定は”ON”にしてください。

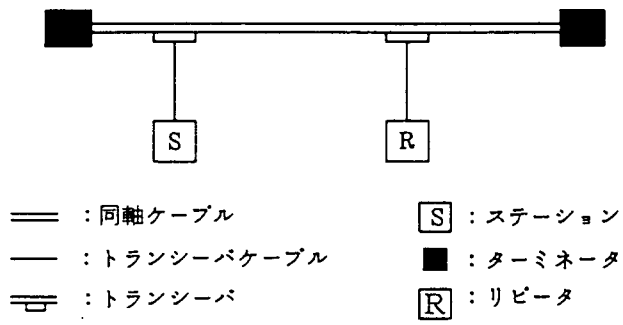


(OFF)  
(ON)



## 10.2 施工分担

下図の施工は、すべてユーザが実施してください。



### 10.3 同軸ケーブルの配線

同軸ケーブルは、屋内の配線ダクトに布設、配線を行い、100V以上の高圧配線とは区別してください。また、ケーブル布設前には必ず短絡・断線がないかどうかチェックしてください。

#### 10.3.1 ケーブルセグメントの布設

(1) ケーブルの布設配線方法は、配線される場所によりいろいろなケーブルの取付方法が考えられます。その主なものは以下のとおりです。

- ・ 天井内コロガン配線
- ・ ケーブルラック内配線
- ・ 壁面露出配線
- ・ フリーアクセス、床ビット内配線
- ・ 電線管内配線

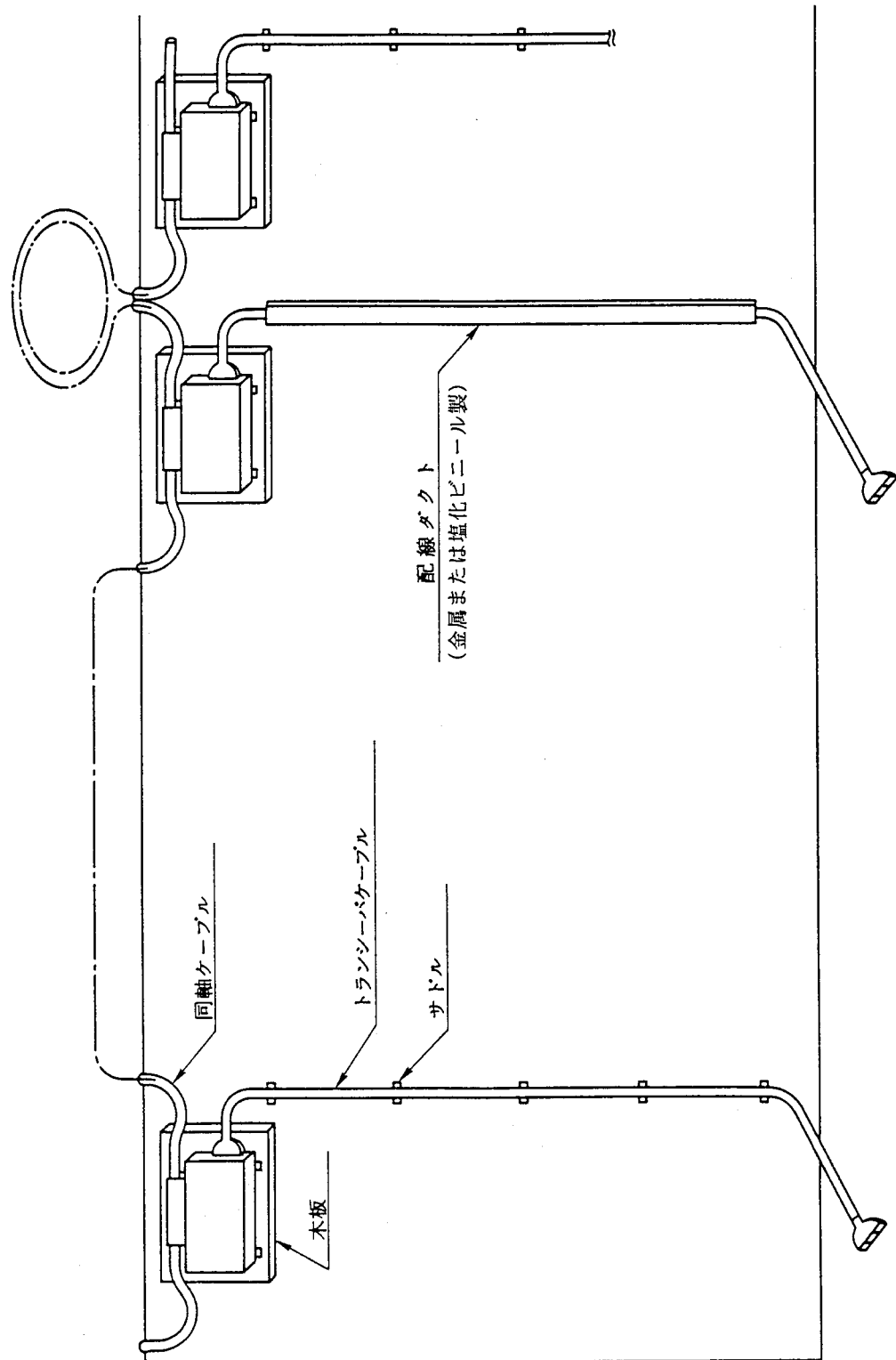
(2) 布設配線工事上の留意事項は以下のとおりです。

- ・ このケーブルは、原則として屋内に布設、配線してください。
- ・ ケーブルの重量は、約1.9kg/10mです。
- ・ ケーブルの布設中、ケーブル本体に25kg・f以上の張力を加えないでください。
- ・ ケーブルの曲げ半径は布設時、最終固定時共に250R（止むを得ないとき150R）以上としてください。
- ・ 壁面、天井などへの固定はサドルを用いて行い、特殊な場合を除き固定間隔は1mを標準とします。その際、サドルの締付けなどによりケーブルが変形しないようにしてください。
- ・ ケーブルラックにケーブルを固定する場合の固定間隔は2mを標準とします。
- ・ 管路内配線の際に使用する電線管は、防火壁貫通部に使用される場合などを除き、通常の配管の場合は、内径22mm以上の管路を使用してください。
- ・ 使用する電線管の曲げ半径は、300mm以上としてください。
- ・ 床上または床際にケーブルを配線する場合は、歩行または器物によりケーブルに変形、損失を受け易いので結びなどにより保護を行ってください。
- ・ ケーブルの外部導体は保安上、接地を行ってください。接地を行う場合は、1セグメントの1点で接地を行い第3種接地以上としてください。接地点以外のケーブルの金属の露出部分が大地や他の金属部分に接触しないようコネクタ、ターミネータは付属のブーツを被せるか、絶縁テープを巻き絶縁してください。

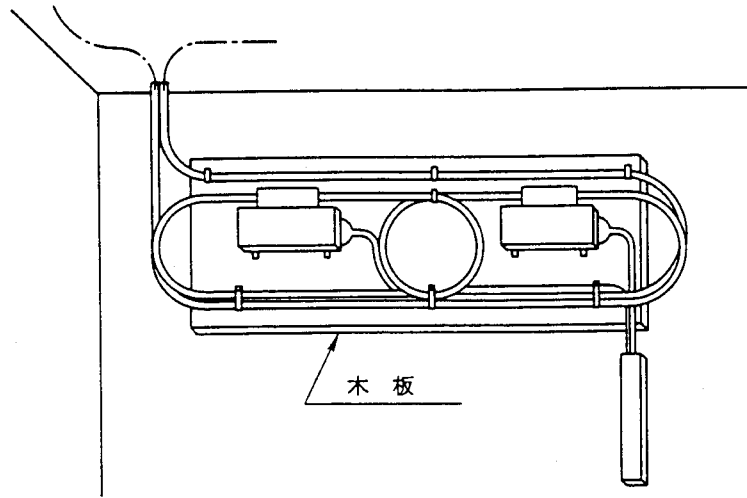
**10.4 トランシーバ（コネクタ用）の設置・取付**

- (1) トランシーバの設置場所および取付方法は、現場の状況によりいろいろ考えられます。主な設置場所は次のような所が考えられます。
- ・ 壁面に設置
  - ・ ステーションの傍に設置
- 図10-1から図10-3に設置例を示します。
- (2) トランシーバを取付ける上での留意事項は以下のとおりです。
- ・ トランシーバを取付金具を介して木ネジなどで固定してください。
  - ・ トランシーバの取付間隔は、2.5m以上としてください。
- (3) トランシーバの取付け
- 同軸ケーブル側のコネクタは、10.1.7項に示したものを使用します。トランシーバはケーブルに無理な力がかからないよう4箇所のネジ穴で固定してください。トランシーバ寸法図は10.1.4項を参照してください。同軸ケーブルの外部導体はアース電位から浮いていますので、同軸コネクタは他の金属に触れないようゴムブーツまたはビニールテープなどで絶縁してください。（トランシーバ本体のケースはトランシーバケーブルの接続によってアース電位に保たれていますが、多点アースとならないようトランシーバ本体のケースは取付時に絶縁してください。）
- また、コネクタの同軸ケーブルへの取付けは10.6節を参照してください。
- (4) 設置場所を選択する際には、下記事項を厳守してください。
- ・ コネクタ・ターミネータのゆるみを確認できる。
  - ・ トランシーバケーブルコネクタのゆるみを確認できる。
  - ・ 付帯のLEDを確認できる。

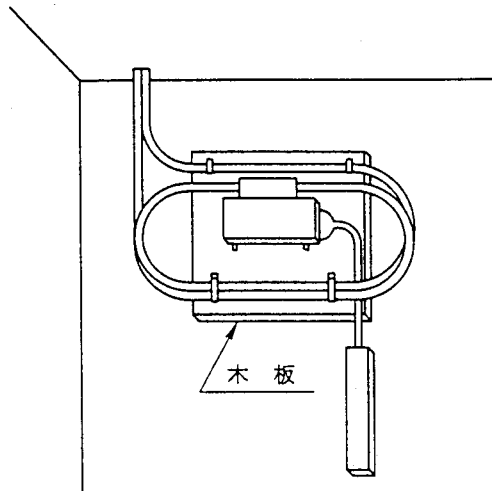
トランシーバ、トランシーバケーブル設置例



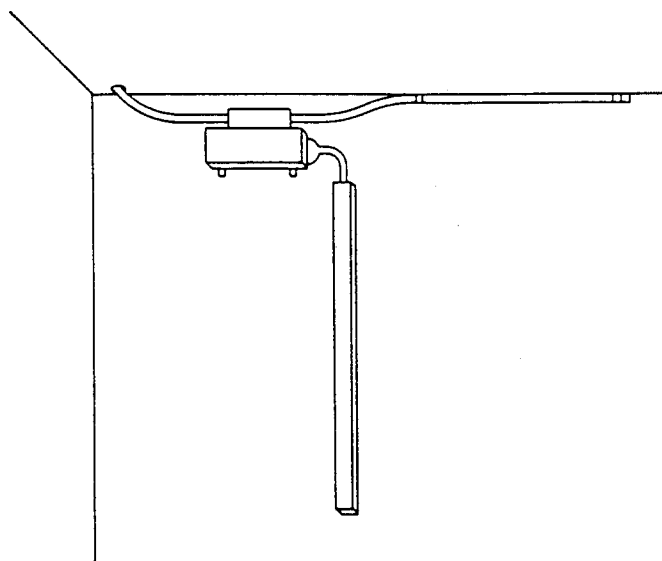
壁面設置例 (1)



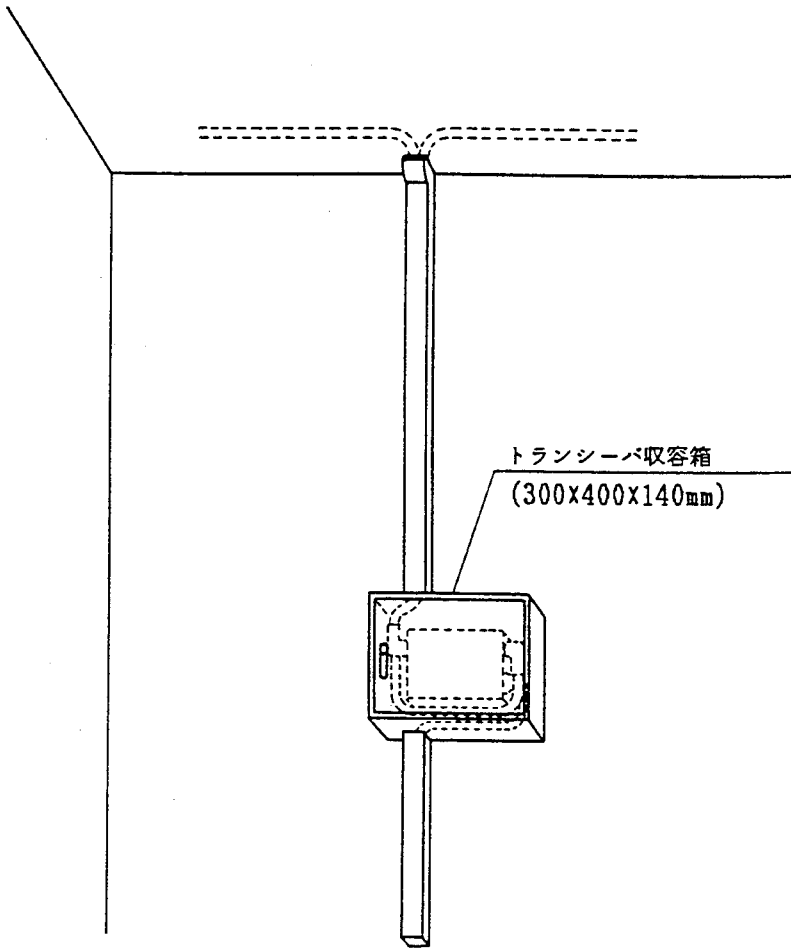
壁面設置例（2）



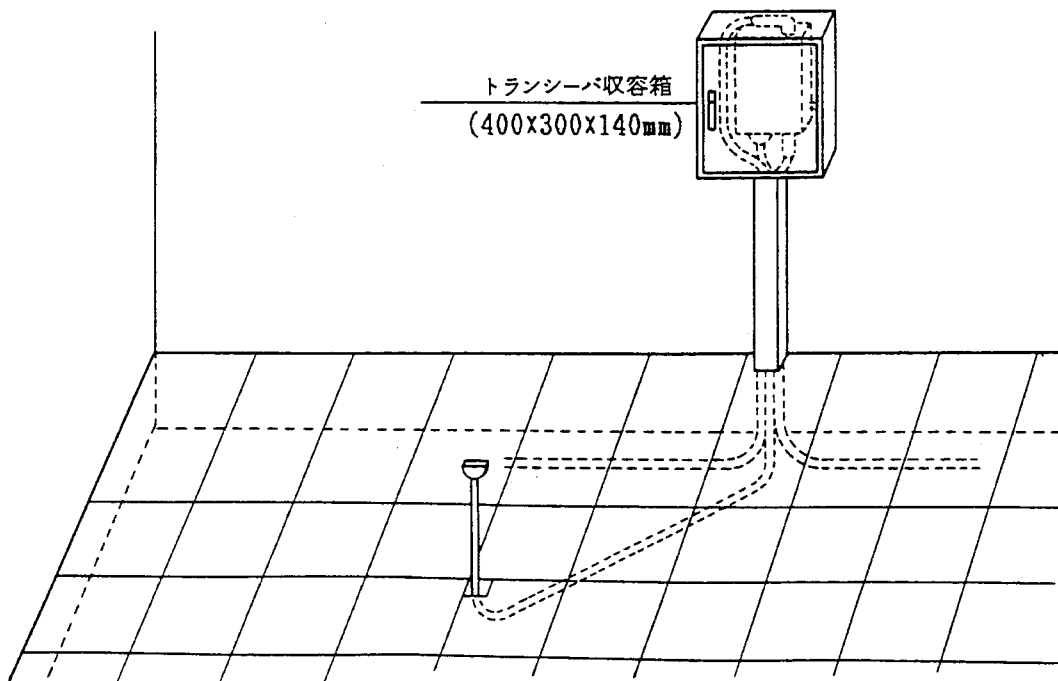
壁面設置例（3）



壁面設置例（4）



BOX内設置例 (1)



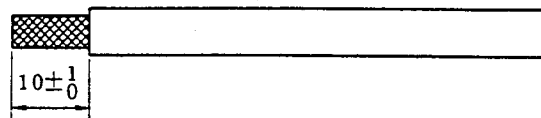
BOX内設置例 (2)

## 10.5 トランシーバ（タップ形）の設置・取付

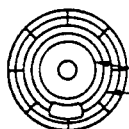
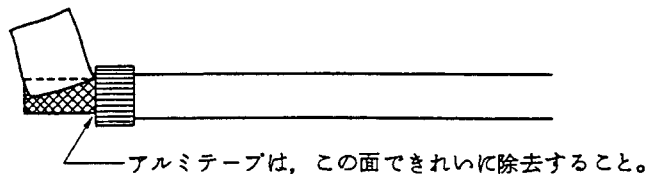
トランシーバの設置場所、取付方法および留意事項は10.4節と同様です。  
 なお、タップコネクタの同軸ケーブルの取付は10.8節を参考にして行ってください。  
 タップ形トランシーバの寸法図は10.1.3項を参照してください。

## 10.6 同軸コネクタの取付

- (1) コネクタの取付手順  
 同軸コネクタの取付作業手順を以下に示します。
- ① PVCシースムキ

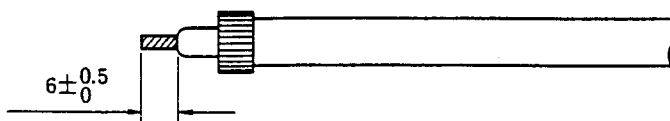


- ② アルミテープ除去

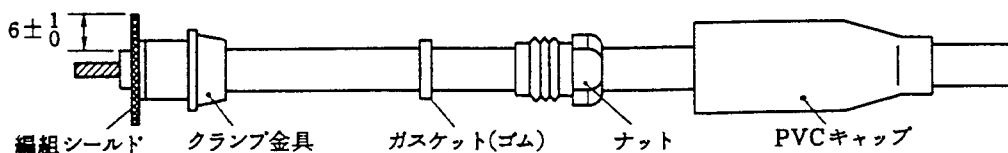


アルミテープは上図のとおりきれいに除去すること。

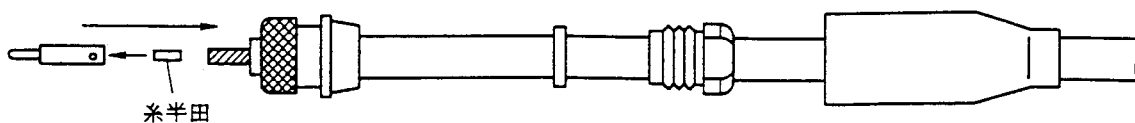
- ③ 絶縁体ムキ



- ④ 部品組み込みおよびシールド処理

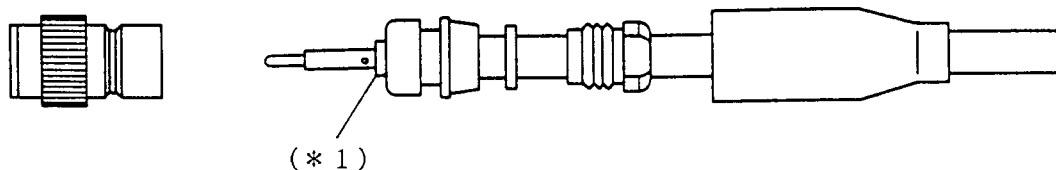


- ⑤ シールド処理およびピンコンタクト半田付け





## ⑥ 組 立

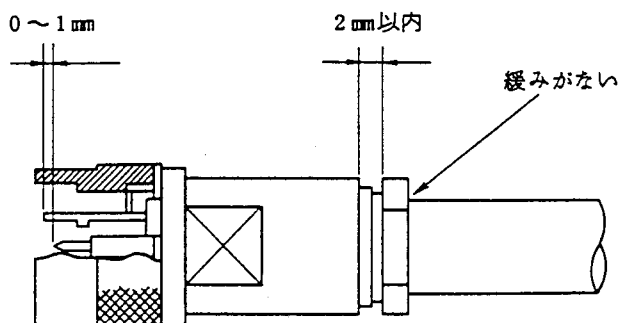


(\*1) ピンコンタクト絶縁体に1mm以上の隙間および絶縁体内に食いこみのないこと。

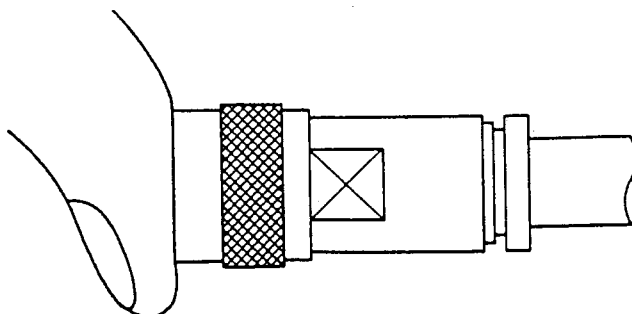
## (2) コネクタ取付後のチェック方法

## (a) コネクタ開口部の寸法

- ・コネクタ先端の外部導体と内部のコンタクトの差が0～1mm以内。異常な内部コンタクトのツギ出し、または引き込みがないこと。



- ・一般にコネクタ開口部に親指を当ててその腹に内部コンタクトの先端が軽く触れる程度。



- ・目視により中心導体に異常な偏心がないこと。

## (b) 緩みの確認

- ・取付後、コネクタのボディと同軸ケーブルを手でつかんでひねり、緩みがないこと。  
締付け後、締付けナットと本体のすき間は約2mm以内であること。

## (c) 絶縁抵抗（ターミネータをはずすこと）

- ・トランシーバが付いていないとき  
内-外導体間 1000M $\Omega$ /km以上（メガ-500VDC）

（注）試験後、必ず放電してください。

- ・トランシーバが付いているとき  
一般の回線テスターで外部導体側を内部電池の+極にして測定し、 $\infty$ 表示であること。

## 10.7 タップコネクタの取付

タップ形トランシーバのタップコネクタと同軸ケーブルの接続は次のとおりです。

- (1) 同軸ケーブル①を、タップコネクタのペース③の溝に挿入し、更に上部からカバー②を取付けることによって、同軸ケーブル①を固定します。
- (2) 六角ボルト⑥を、ボックスドライバを使用して規定されたトルクに従ってネジ締付をして、同軸ケーブル①の外部導体と接続させます。

六角ボルトネジ⑥ネジ締付トルク：30～40 [kg・cm]

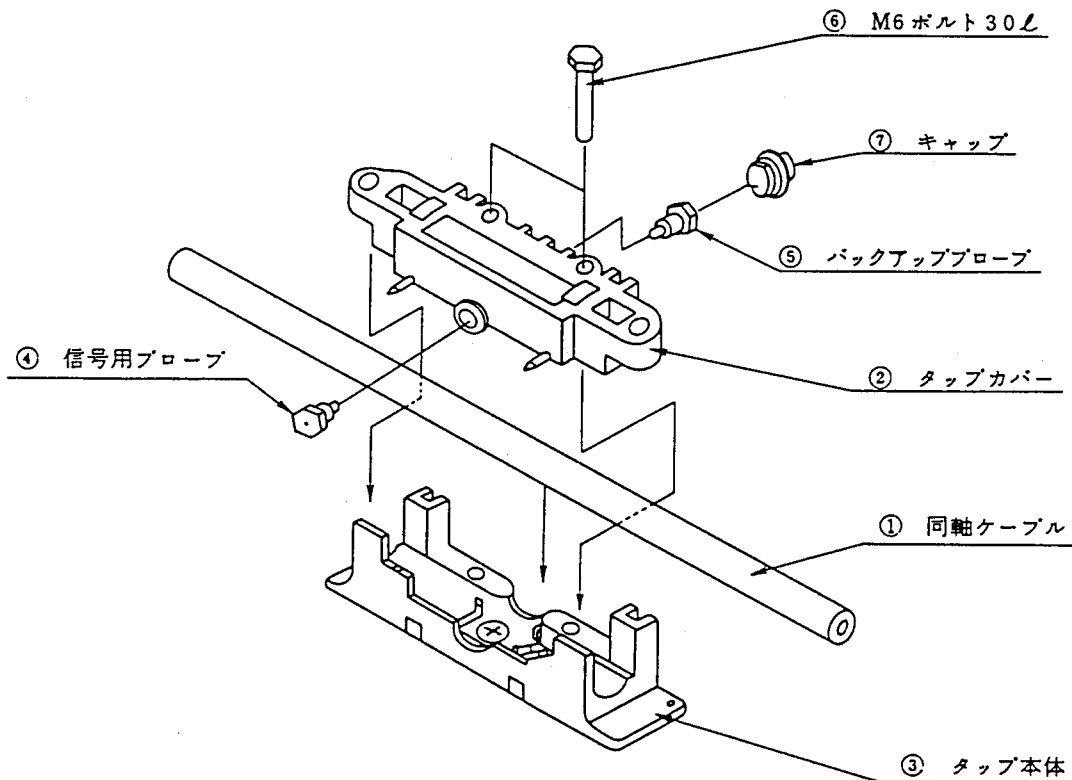
- (3) バックアッププローブ⑤、信号用プローブ④の順に、両側から同時にボックスドライバを使い、規定されたトルクでゆっくりネジ締付をして、同軸ケーブル①の中心導体と接続させます。

信号用プローブ④  
バックアッププローブ⑤ } ネジ締付トルク：20～30 [kg・cm]

- (4) バックアッププローブ⑤の上に、添付されているキャップ⑦を取付けます。

以上によってタップコネクタと同軸ケーブルの接続を完了します。

なお、信号用プローブ④およびバックアッププローブ⑤の先端とネジ山はこわれやすいので取扱いには十分注意してください。



タップコネクタ組立図

(注)

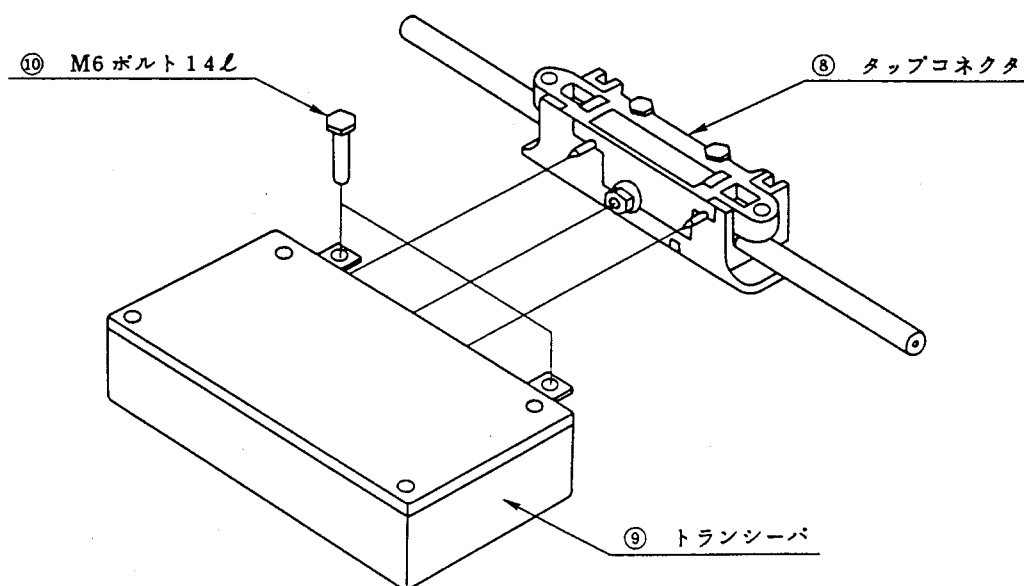
- ・同軸ケーブルへの接続手順は、上記順序で行ってください。  
プローブ④、⑤を取付後、ケーブルを取付けますと、プローブ④、⑤が破壊されます。それを防ぐためにプローブ④、⑤を完全に外した状態で、ケーブルを装着してください。
- ・プローブ④、⑤締付の作業後、ボルト⑥の増し締めは行わないでください。プローブに力が加わり破壊の原因になります。

タップコネクタとトランシーバの接続は、次の手順に従って実施します。

- (1) タップコネクタ⑧をトランシーバ⑨の側面に接着することによって、タップコネクタ⑧のプローブおよびグランド端子がトランシーバ⑨の取付穴に挿入されて、接続されます。
- (2) 六角ボルト⑩をボックスドライバを使用して、規定されたトルクに従ってネジを締付することによって、トランシーバ⑨とタップコネクタ⑧が完全に固定されます。

六角ボルト⑩ネジ締付トルク：30～40[kg・cm]

以上によってタップコネクタとトランシーバの接続を完了します。



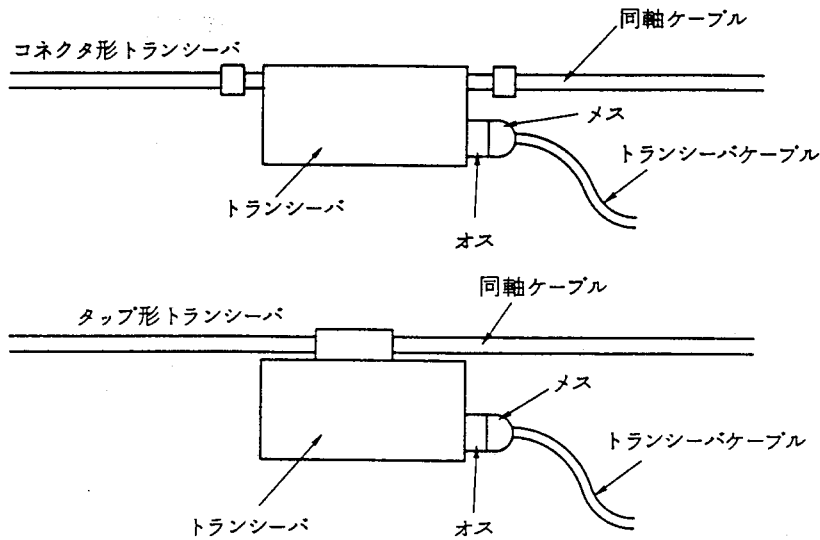
コネクタ，トランシーバ接続図

## 10.8 トランシーバケーブルの取付

トランシーバケーブルのケーブル長は標準 5 m, 最大 15 m とします。

### 〈トランシーバケーブルの取付〉

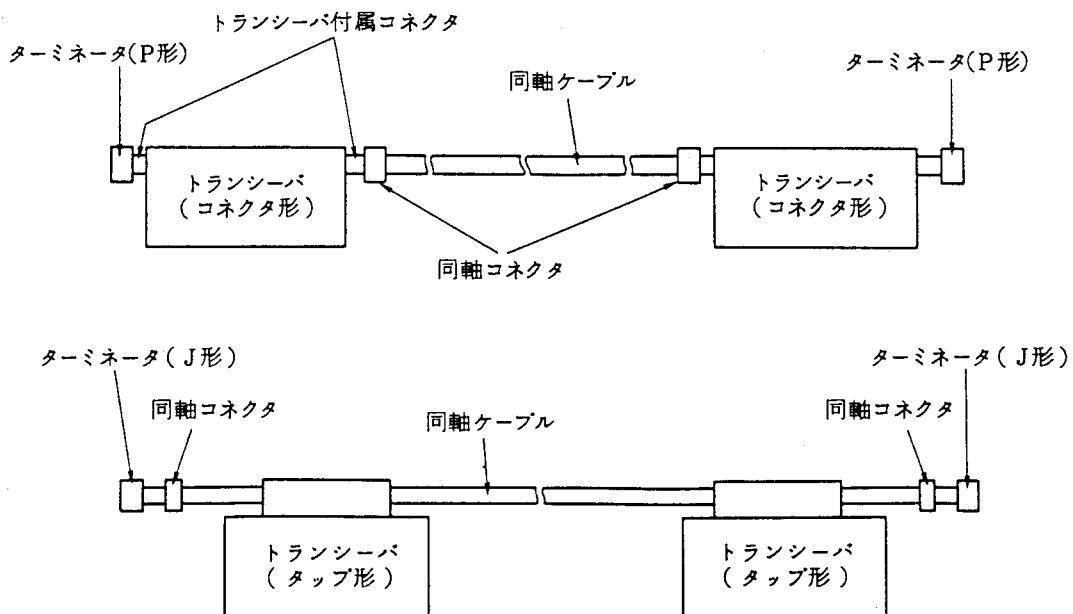
トランシーバ本体へのトランシーバケーブルの接続は、ケーブルのロック用リテーナをスライドさせ、トランシーバ本体のロック用ポストに完全にロックするように取付けてください。トランシーバ本体がオス、トランシーバケーブルがメスとなります。



## 10.9 ターミネータの取付

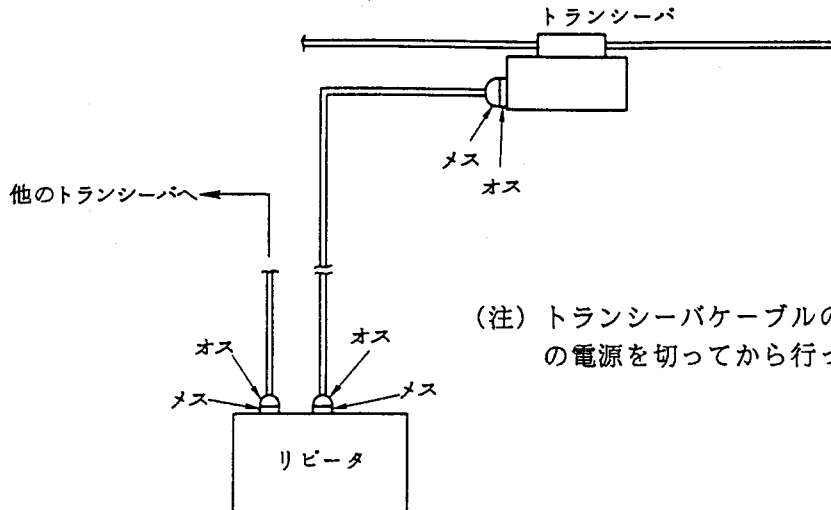
### 〈ターミネータの取付〉

ターミネータは同軸セグメントの最端部（両端）に必ず接続してください。



10.10 リピータの設置・取付

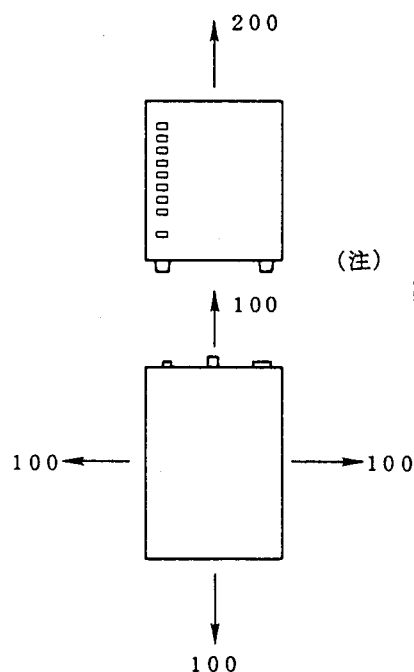
(1) 接続方法



(注) トランシーバケーブルの着脱は必ずリピータの電源を切ってから行ってください。

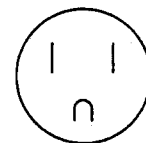
(2) 設置場所とスペースの確保

- リピータを設置する場所は、ワークステーション（サーバ）付近で、容易に保守できる場所（一般事務室内で天井裏、地下などは不可）を選び、前後、左右、上方に少なくとも以下のスペースを確保してください。なお、リピータはAC電源を必要としますので、接地付コンセントを準備してください。



定格 AC100V ±10% 平常時：0.07kVA  
突入時：10A

(注) フロントパネル面は開放できるだけのスペースを確保してください。



2極接地極付プラグ形状  
15A125V  
(JIS C8303)

単位：mm

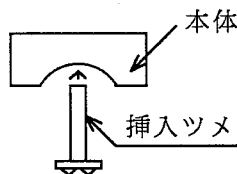
- チリやホコリの多いところでは使用しないでください。
- 底面に空気の入力口、上面に吹き出し口がありますのでふさがらないでください。
- 保守を考慮し、リピータの設置場所付近には電話を取付けることを推奨します。
- 誤って電源を切ることのないよう、独立した電源を使用してください。リピータの電源が切れると伝送機能が停止します。

### 10.11 システムの接地

- リピータの接地  
リピータは、必ず3線式電源を使用するか、または接地端子で接地を行ってください。
- 各ステーションの接地  
LAN制御機構に接続されているすべての装置は第3種以上の接地を行ってください。  
システム内に接地されていない装置がある場合、接地されている装置との間で感電のおそれがあります。また、データエラー（CRCエラー）の原因にもなります。
- 同軸ケーブルの接地  
各セグメントごとに同軸ケーブルの1点接地を行ってください。  
これは保安上の目的と同時に、不完全な大地との接触による雑音の発生を防止するためです。  
接地にはアース端子を使用してください。

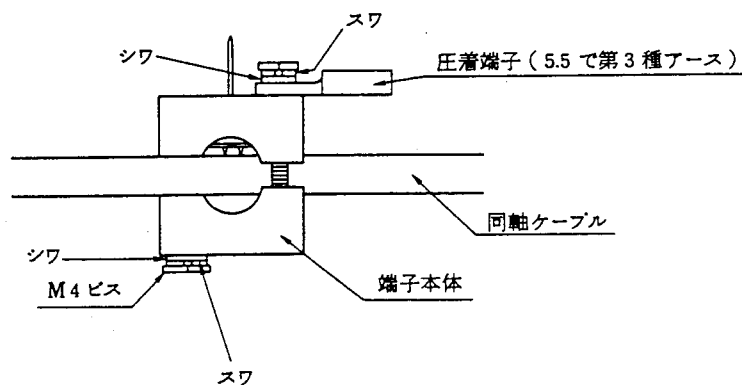
### 10.12 アース端子取付方法

- (1) 挿入ツメを本体に挿入します。

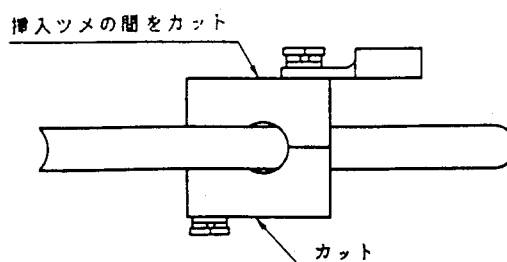


- (2) 同軸に取り付けて、M4のビスを交互に締め付けます。このとき圧着端子はどちらかのビスに取り付けます。

同軸セグメント上の位置はアースが取り付けやすい任意の1箇所のみとしてください。



- (3) 締め付け後、挿入ツメの間をカットします。



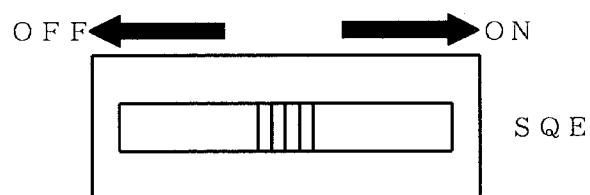
### 10.13 シングルポートトランシーバの設定

#### ●シングルポートトランシーバのSQEスイッチの設定

シングルポートトランシーバのSQEスイッチは、接続先により下表の設定変更が必要となりますので注意してください。

接続先 SQEスイッチ	SIOET LINK コントローラ	マルチポート トランシーバ	リピータ
設 定	ON	OFF	OFF

なお、シングルポートトランシーバHLT-200, -200TBのSQEスイッチはケース内部にあります。設定を変更する時はケースを開いて実施してください。（ボード上のシルク印刷“SQE”側にスイッチを倒すとONとなります。）



10.14 マルチポートトランシーバの設定および表示

(1) 動作モードの設定

マルチポートトランシーバは、ネットワークモードとローカルモードの2種類の動作モードで使用できます。モードの設定は、裏面パネル上の切替スイッチの操作により行います。

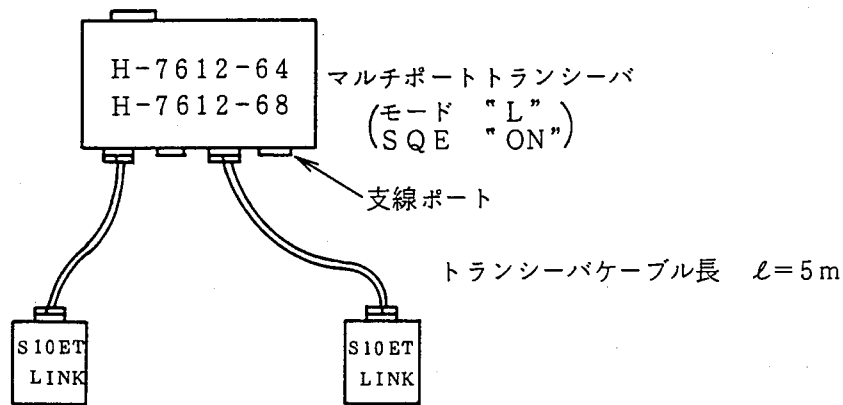
● ローカルモード

同軸ケーブルから切り離し、単独で使用するモードです。

中継ポートへのトランシーバケーブル接続は行わないでください。

モード切替スイッチを 'L' (ローカルモード) に設定して使用します。

また、このとき支線ポートのSQEスイッチは 'ON' に設定します。

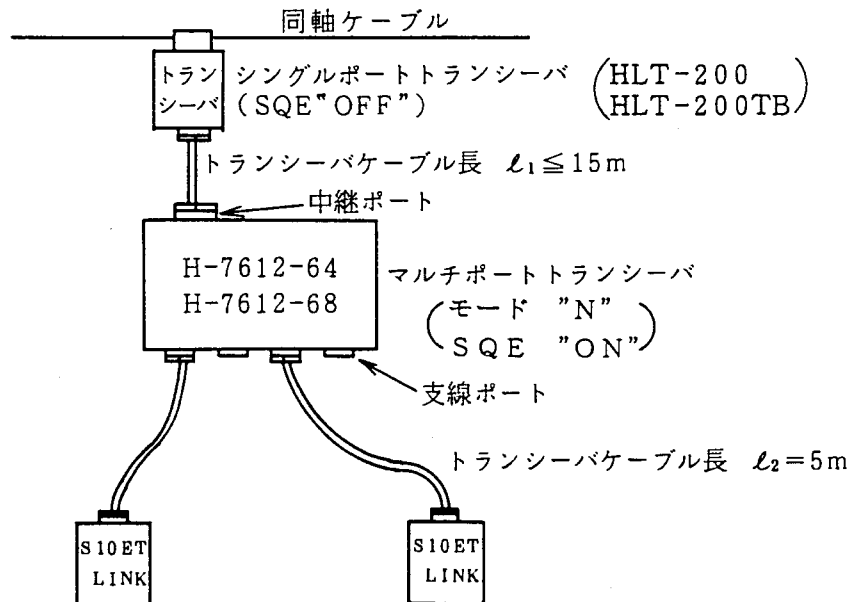


● ネットワークモード

下図のように同軸ケーブルと接続して使用するモードです。

モード切替スイッチを 'N' (ネットワークモード) に設定して使用します。

また、このとき中継ポートに接続されたシングルポートトランシーバのSQEスイッチは 'OFF' に設定します。





## (2) 切替スイッチの設定

マルチポートトランシーバには、2つの切替スイッチがあります。それぞれの機能を下表に示します。

切替スイッチの設定

スイッチの種類	スイッチの位置	機 能	製品出荷時の設定
SQE切替スイッチ	裏面パネル	SQE機能のON/OFF	‘ON’
動作モード切替スイッチ	裏面パネル	動作モードの切替	‘N’(ネットワークモード)

## (3) リピータ接続時のSQEスイッチの設定

リピータをマルチポートトランシーバに接続する場合、マルチポートトランシーバの当該支線ポートのSQEスイッチを‘OFF’に設定してください。

## (4) 電源スイッチ

裏面パネルのスイッチを‘I’側に倒すとマルチポートトランシーバの電源が‘ON’されます。

## (5) LEDの表示

筐体の正面パネル上には、“POWER”LEDおよび各支線ポートごとに“LINK”LEDがあります。

“POWER”LED：電源スイッチが‘ON’の時に点灯します。

“LINK”LED：情報ステーションがマルチポートトランシーバの支線ポートに接続されているとき（情報ステーションよりDC12Vが給電されているとき）に点灯します。



# 11 索 引

〈ア行〉		〈英数字〉	
浮いたソケット .....	30	accept ( ) .....	39
〈サ行〉		Cモードプログラム .....	26
シングルポートトランシーバ .....	18	connect ( ) .....	41
ソケットライブラリ .....	24, 28	gethostname ( ) .....	35
セグメント .....	14	getpeername ( ) .....	37
〈タ行〉		HOSTSテーブル .....	26
タスクの強制終了 .....	30	host_type .....	35, 53
着信 .....	30	init ( ) .....	32
電源内蔵型トランシーバ .....	8, 10	IPアドレス .....	20
ドライバ .....	25	IPプログラム .....	25
トランシーバケーブルの接続 .....	11	listen ( ) .....	43
〈ナ行〉		recv ( ) .....	49
ネットワーク番号 .....	20	recvfrom ( ) .....	51
ネットワークモード .....	140	sckidtbl .....	37, 39, 41, 53
〈ハ行〉		shutdown ( ) .....	44
物理アドレス .....	20	send ( ) .....	45
ホスト番号 .....	20	sendto ( ) .....	47
ホスト名 .....	22	sockaddr .....	33, 53
〈マ行〉		socket ( ) .....	33
マルチポートトランシーバ .....	17	SQEスイッチ .....	139, 140
モジュール .....	30	TCPプログラム .....	24
〈ラ行〉		TCPローカルコネクションネーム .....	37
リピータ .....	14	UDPプログラム .....	24
リンクセグメント .....	14		
ローカルモード .....	140		

