

HITACHI

ユーザーズマニュアル

オプション

ET.NET

(LQE520)

S10mini S10V

SVJ-1-103(H)

ユーザーズマニュアル

オプション

ET.NET

(LQE520)

S10mini

SIOV

この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うえ、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い合わせください。

2002年 7月	(第1版)	SVJ-1-103 (A)	(廃版)
2002年12月	(第2版)	SVJ-1-103 (B)	(廃版)
2003年 6月	(第3版)	SVJ-1-103 (C)	(廃版)
2005年 3月	(第4版)	SVJ-1-103 (D)	(廃版)
2008年 3月	(第5版)	SVJ-1-103 (E)	(廃版)
2009年 2月	(第6版)	SVJ-1-103 (F)	(廃版)
2010年10月	(第7版)	SVJ-1-103 (G)	(廃版)
2017年 5月	(第8版)	SVJ-1-103 (H)	

- 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したりすることは、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

- 製品の据え付け、運転、保守、点検の前に、必ずこのマニュアルと関連マニュアルをすべて熟読し、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習熟してから正しく使用してください。
- このマニュアルは、製品を使用する人がいつでも見られるところに必ず保管してください。
- このマニュアルでは、安全上の注意事項のランクを潜在危険の重大度によって、「危険」、「警告」、「注意」、「通知」と区分しています。

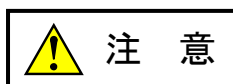
警告表示の定義



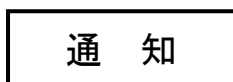
: この表示を無視して誤った取り扱いをすると、死亡または重大な傷害を引き起こす危険の存在を示す。



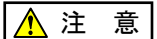
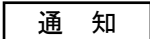
: この表示を無視して誤った取り扱いをすると、死亡または重大な傷害を引き起こすおそれのある危険の存在を示す。



: この表示を無視して誤った取り扱いをすると、軽度の傷害または中程度の傷害を引き起こすおそれのある危険の存在を示す。



: この表示を無視して誤った取り扱いをすると、人身傷害とは関係のない損害を引き起こすおそれのある危険の存在を示す。

なお、 **注意**、 **通知** 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。ど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。

「重大な傷害」、「軽度の傷害または中程度の傷害」、「人身傷害とは関係のない損害」について、具体的な内容を以下に示します。

重大な傷害

失明、けが、やけど（高温、低温）、感電傷害、骨折、中毒などで、後遺症が残るものおよび治療のために入院、長期の通院を要するもの

軽度の傷害または中程度の傷害

治療のために入院や長期の通院を必要としないけが、やけど、感電傷害など


人身傷害とは関係のない損害

周囲の財物の損傷、製品本体の故障や破損、データの損失など、人身傷害以外の損害

安全上の注意事項は、安全性を確保するための原則に基づいた、製品本体における各種対策を補完する重要なものです。製品本体やマニュアルに表示されている安全上の注意事項は、十分に検討されたものですが、それでも、予測を超えた事態が起こることが考えられます。操作するときは指示に従うだけでなく、常に自分自身でも注意するようにしてください。また、製品の安全な運転および保守のために、各種規格、基準に従って安全施策を確立してください。


このマニュアルの中に書かれている警告とその記載箇所を、以下にまとめて示します。

(3-6ページ)

 危 険

- 10BASE-5で通信する場合は、ET.NETモジュールのFG端子を、マウントベースのFG端子経由でD種接地してください。
- アース線は、線径2mm²以上のものを使用してください。

(6-16ページ)

 危 険

試験後は必ず放電してください。放電しないと感電します。

(6-21ページ)

 危 険


すべてのステーションは、D種接地以上の接地をしてください。
接地されていないステーションがある場合、感電の恐れがあります。

(3-4ページ)

 警 告

- 発煙、異臭などがあった場合は、ただちに電源を切って原因を調査してください。
- このマニュアルに記載されていない設置、配線、取り扱い、および内部の改造はしないでください。これらに起因する弊社装置と周辺機器の破損および人身災害について、弊社は一切の責任を負いません。
- 通電中は端子台やコネクタのピンに絶対に触れないでください。通電中に端子台やコネクタのピンに触れると感電する恐れがあります。

(3-2ページ)

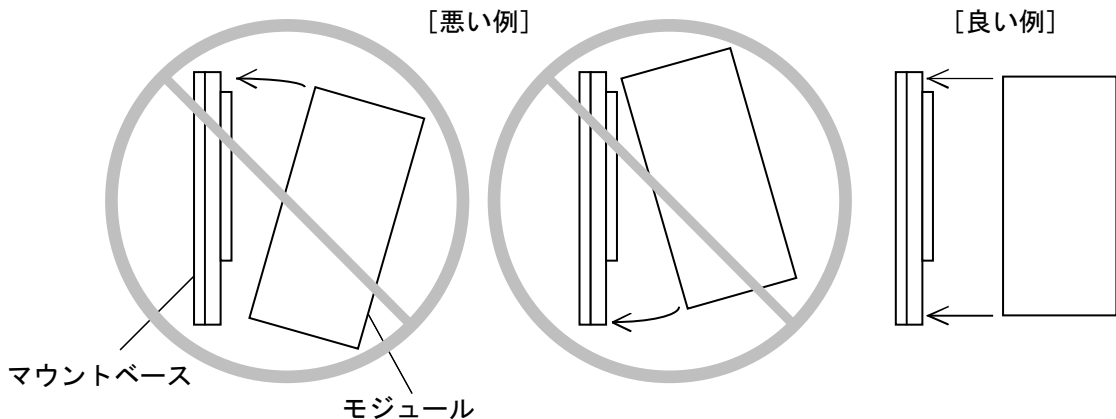
 注 意

- 故障の原因になりますので、水漏れの危険のあるところでは、防滴構造の筐体内に収納して使用してください。
- 通電中モジュールに触れますと静電気によりモジュールの誤動作、破損する恐れがあります。通電中はモジュールに触れないでください。やむを得ず触れる場合は、触れる前に人体の静電気を放電してください。また、非通電中にモジュールの各種スイッチの設定、ケーブルの取り付け／取り外し、コネクタの抜き差しなどをする前にも、人体の静電気を放電してください。

(3-4ページ)

 注 意

- コネクタにほこりなどが付着して接触不良が発生する可能性があります。装置の開梱後、ただちに設置および配線をしてください。
- モジュールが破損する恐れがあります。モジュールの取り付け／取り外しをするときは、以下の点に注意してください。
 - ・ モジュールをマウントベースのコネクタに取り付ける前に、コネクタのピンの曲がりや折れはないか、ピンが一直線上に並んでいるか、またピンにゴミなどが付着していないかを確認してください。
 - ・ モジュールは、以下に示すようにマウントベースの垂直面に沿って平行移動してください。モジュールを傾けたまま、コネクタへ取り付けまたはコネクタから取り外しすると、コネクタのピンが損傷する恐れがあります。
 - ・ 筐体の構造上、マウントベースが頭上に配置されている場合、モジュールは脚立などを使用してまっすぐに取り付けてください。斜めに取り付けるとコネクタを破損する恐れがあります。



(3-5ページ)



注 意

- マニュアルに従って取り付けをしてください。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。
- 静電気によりモジュールが破損する恐れがあります。作業する前に、人体の静電気を放電してください。
- ねじは確実に締め付けてください。締め付けが不十分な場合、誤動作や発煙、発火を引き起こす原因になります。

(7-2ページ)



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。作業する前に、人体の静電気を放電してください。
- モジュールは、必ず電源を切ってから交換してください。感電、モジュールの破損および誤動作の恐れがあります。

通 知

- 弊社提供ソフトウェアを改変して使用した場合に発生した事故や損害につきましては、弊社は責任を負いかねますのでご了承ください。
- 弊社提供以外のソフトウェアを使用した場合の信頼性については、弊社は責任を負いかねますのでご了承ください。
- ファイルのバックアップ作業を日常業務に組み入れてください。ファイル装置の障害、ファイルアクセス中の停電、誤操作、その他何らかの原因によりファイルの内容を消失することがあります。このような事態に備え、計画的にファイルのバックアップを取っておいてください。
- この製品は、産業廃棄物として専門の処理業者に廃棄を依頼してください。
- このモジュールの近くでは、トランシーバ、携帯電話等を使用しないでください。近くでトランシーバ、携帯電話等を使用しますとノイズにより誤動作、システムダウンとなる恐れがあります。
- モジュールの故障などでメモリの内容が破壊されることがあります。重要なデータは必ずバックアップを取っておいてください。
- システムの構築やプログラムの作成などは、このマニュアルの記載内容をよく読み、書かれている指示や注意を十分理解してから行ってください。誤操作により、システムが故障することがあります。
- このマニュアルは、必要なときすぐに参照できるよう、手近なところに保管してください。
- このマニュアルの記載内容について、疑問点または不明点がございましたら、販売店までお知らせください。
- お客様の誤操作に起因する事故発生や損害につきましては、弊社は責任を負いかねますのでご了承ください。
- 非常停止回路、インタロック回路などは、この製品の外部で構成してください。この製品の故障により、機械の破損や事故の恐れがあります。

通 知

モジュールNo.設定スイッチは、電源を切った状態で操作してください。動作中に操作すると誤動作の原因になります。

(3-5ページ)

通 知

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。

(3-7ページ)

通 知

- 接触不良または断線により誤動作する恐れがあります。10BASE-5コネクタを接続した場合は、ロック用ポストがリテーナによってロックされているか確認してください。
- 通電中は10BASE-5コネクタに触れないでください。静電気などによりシステムが誤動作する可能性があります。

(4-3ページ)

通 知

- S10mini用ET.NETシステムを動作させるためには、Microsoft® Internet Explorer 4.01以降が必要です。インストールされていない場合は、S10mini用ET.NETシステムが正常に動作しません。
- S10mini用ET.NETシステムをインストールする前に、すべてのWindows®上で作動するプログラムを必ず終了してください。ウイルス監視ソフトウェアなどメモリに常駐しているプログラムも必ず終了してください。終了せずにインストールすると、エラーが発生する場合があります。その場合は、「4. 2. 2 アンインストール」にて一旦アンインストールし、すべてのWindows®上で作動するプログラムを終了してから、再度S10mini用ET.NETシステムをインストールしてください。

(4-4ページ)

通 知

- S10V用ET.NETシステムを動作させるためには、S10V基本システムが必要です。インストールされていない場合は、S10V用ET.NETシステムをインストールできません。
- S10V用ET.NETシステムをインストールする前に、すべてのWindows®上で作動するプログラムを必ず終了してください。ウイルス監視ソフトウェアなどメモリに常駐しているプログラムも必ず終了してください。終了せずにインストールすると、エラーが発生する場合があります。その場合は、「4. 2. 2 アンインストール」にて一旦アンインストールし、すべてのWindows®上で作動するプログラムを終了してから、再度S10V用ET.NETシステムをインストールしてください。

(4-6ページ)

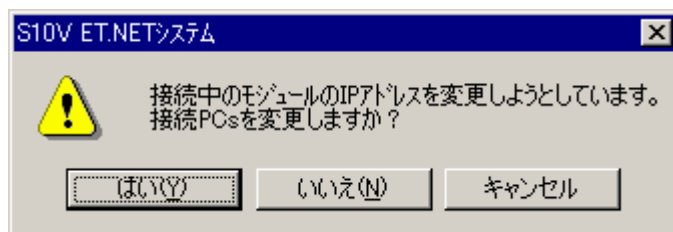
通 知

Windows®でアンインストール中に「共有ファイルを削除しますか？」の画面が表示された場合は、 ボタンをクリックして共有ファイルを削除しないでください。

(4-18ページ)

通 知

イーサネットケーブルで接続しているET.NETモジュールのIPアドレスを変更しようとした場合、以下のメッセージが表示されます。



PCsリセット後、変更したIPアドレスでPCsと接続しなおす場合は ボタンを、接続しなおさない場合は ボタンを、IPアドレスの設定を中止する場合は ボタンをクリックしてください。

(4-21ページ)

通 知

経路情報は [IPアドレス設定] 画面で ボタンをクリックしたときにPCsまたはファイルに登録されます。そのため、[IPアドレス設定] 画面で ボタンをクリックすると経路情報はPCsまたはファイルに登録されません。

通 知

- S10Vに実装する場合、以下の制限があります。
 - ・ モジュールRev.B (Ver-Rev : 0005-0001) 以前のET.NETモジュール (LQE520) は、ソケットハンドラを使用してのタスクからの通信機能を使用できません。ツールとの通信用にのみ使用できます。S10Vと組み合わせてソケットハンドラを使用される場合は、モジュールRev.F (Ver-Rev : 0006-0000) 以降のモジュールを使用してください。
 - ・ ET.NETモジュールのLQE520とLQE720を同じLPUユニット上で混在させることはできません。
- なお、上記Ver-Revは、S10V基本システムの「Module List」で表示されるET.NETモジュールのマイクロプログラムのVer-Revです。
- 1つのモジュールで、同時に使用できるソケット数は、TCPが12個、UDPが8個までです。
 - 0~9999のポート番号はシステムで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TCPが1~4096バイト、UDPが1~1472バイトです。
 - IPアドレス、サブネットマスクは、CPUまたはLPU内のOSテーブルに設定されます。CPUまたはLPUの交換やOSの再ローディングをした場合は、再設定が必要です。

—タスクの強制終了—

ソケットハンドラを利用しているタスクが強制終了されると、ソケットが登録状態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、`tcp_close()`または`udp_close()`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て、下記のいずれかの処理をしてください。

- 他のタスクまたは組み込みサブルーチンから浮いたソケットを、`tcp_close()`または`udp_close()`する。
- CPUをリセットする。
- 電源を一度遮断し、復電する。

通 知

`udp_receive()`は、パケットごとの受信をします。したがって、バッファエリアを1,472バイト確保してください。

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、その製品の故障部分をお買い上げの販売店または（株）日立パワーソリューションズにお渡しください。交換または修理を無償で行います。ただし、郵送していただく場合は、郵送料金、梱包費用はご注文主のご負担になります。

次のど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って故障した場合
- 納入品以外の事由によって故障した場合
- 納入者以外の改造または修理によって故障した場合
- リレーなどの消耗部品の寿命によって故障した場合
- 天災、災害など納入者の責任ではない事由によって故障した場合

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、逸失利益など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だけ有効であり、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
- 上記保証範囲外の事由による故障原因の調査

このページは白紙です。

このマニュアルは、以下のハードウェアおよびプログラムプロダクトの説明をしたものです。

<ハードウェア>

ET.NET (LQE520)

<プログラムプロダクト>

S-7890-29 「ET.NETシステム」 (07-01)

S-7895-29 「S10V ET.NETシステム」 (02-05)

来歴一覧表

改訂No.	来歴（改訂内容および改訂理由）	発行年月	備考
A	新規作成	2002.7	
E	モジュールの交換、増設を追加	2008.3	
F	モジュールNo.設定スイッチを「4」または「5」に設定すると、経路情報エラーとなる場合があることを追加	2009.2	
G	オフライン機能追加、安全上のご注意を変更	2010.10	
H	サポートOSにWindows® 7（32bit）、Windows® 10（32bit）を追加	2017.5	

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、S10mini, S10V用オプション ET.NET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「ユーザーズマニュアル オプション ET.NET」は、ET.NETモジュールの取り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。

S10mini, S10V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、コーティングが強化されています。

耐環境仕様品型式は、標準仕様品型式の後に「-Z」が付いています。

(例) 標準仕様品型式 : LQE520

耐環境仕様品型式 : LQE520-Z

このマニュアルは、標準仕様品と耐環境仕様品とで共通の内容となっています。このマニュアルには、標準仕様品のモジュール型式のみを記載していますが、耐環境仕様品をご使用の場合も、このマニュアルに従って、正しくご使用いただくようお願いいたします。

S10Vに実装する場合、以下の制限があります。

- ・モジュールRev.B (Ver-Rev : 0005-0001) 以前のET.NETモジュール (LQE520) は、ソケットハンドラを使用するタスクからの通信機能を使用できません。ツールとの通信用にのみ使用できます。S10Vと組み合わせてソケットハンドラを使用される場合は、モジュールRev.F (Ver-Rev : 0006-0000) 以降のモジュールを使用してください。
- ・ET.NETモジュールのLQE520とLQE720を同じLPUユニット上で混在させることはできません。

なお、上記Ver-Revは、S10V基本システムの「Module List」で表示されるET.NETモジュールのマイクロプログラムのVer-Revです。

<商標について>

- ・Microsoft®, Windows®は、米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。
- ・Ethernet®は米国Xerox Corp.の登録商標です。

<記憶容量の計算値についての注意>

- 2ⁿ計算値の場合（メモリ容量・所要量、ファイル容量・所要量など）

1KB（キロバイト）＝1,024バイトの計算値です。

1MB（メガバイト）＝1,048,576バイトの計算値です。

1GB（ギガバイト）＝1,073,741,824バイトの計算値です。

- 10ⁿ計算値の場合（ディスク容量など）

1KB（キロバイト）＝1,000バイトの計算値です。

1MB（メガバイト）＝1,000²バイトの計算値です。

1GB（ギガバイト）＝1,000³バイトの計算値です。

目 次

1	仕 様	1-1
1.1	用 途	1-2
1.2	仕 様	1-2
1.2.1	一般仕様	1-2
1.2.2	通信仕様	1-2
2	各部の名称と機能	2-1
2.1	各部の名称と機能	2-2
3	実装と配線	3-1
3.1	PCsのご利用にあたり	3-2
3.2	マウントベース	3-3
3.3	モジュールの実装	3-3
3.4	アース配線	3-6
3.5	通信ケーブルの配線	3-7
4	オペレーション	4-1
4.1	立ち上げ手順	4-2
4.2	システムインストールと立ち上げ	4-3
4.2.1	インストール	4-3
4.2.2	アンインストール	4-6
4.2.3	システム立ち上げ	4-6
4.2.4	システム終了	4-8
4.3	コマンド（S10mini用ET.NETシステム）	4-9
4.3.1	接続PCsの変更	4-9
4.3.2	IPアドレス、サブネットマスク設定	4-10
4.3.3	経路情報設定	4-11
4.4	コマンド（S10V用ET.NETシステム）	4-12
4.4.1	接続PCsの変更	4-12
4.4.2	モジュール設定	4-14
4.4.3	IPアドレス設定	4-17
4.4.4	経路情報設定	4-21
4.4.5	IPアドレス設定情報ファイルの読み込み	4-22
4.4.6	IPアドレス設定情報のファイル保存	4-24

4.4.7	IPアドレス設定情報の印刷	4-26
4.4.8	IPアドレス設定情報のCSV出力	4-28
4.4.9	IPアドレス設定情報の比較	4-30
5	プログラミング	5-1
5.1	ET.NETのソフトウェア構成	5-2
5.2	システムプログラム	5-3
5.2.1	ソケットハンドラ	5-3
5.2.2	ソケットドライバ	5-3
5.2.3	TCPプログラム	5-3
5.2.4	UDPプログラム	5-4
5.2.5	IPプログラム	5-4
5.2.6	ドライバ	5-4
5.3	ユーザプログラム	5-4
5.4	ソケットハンドラ	5-5
5.4.1	ソケットハンドラー一覧	5-6
5.5	ソケットハンドラ発行手順例	5-30
5.5.1	TCP/IPプログラム使用例	5-30
5.5.2	UDP/IPプログラム使用例	5-31
5.6	CPU間通信プログラム例	5-34
5.6.1	システム構成およびプログラム構成	5-34
5.6.2	CPU01側プログラムのフローチャートとプログラム例	5-35
5.6.3	CPU02側プログラムのフローチャートとプログラム例	5-39
5.7	CPU間連続通信プログラム例	5-42
5.7.1	システム構成およびプログラム構成	5-42
5.7.2	CPU01側プログラムのフローチャートとプログラム例	5-43
5.7.3	CPU02側プログラムのフローチャートとプログラム例	5-48
6	利用の手引き	6-1
6.1	推奨するネットワーク構成部品	6-2
6.2	10BASE-5のシステム構成	6-4
6.2.1	10BASE-5のシステム構成概要	6-4
6.2.2	10BASE-5システム構成上の注意	6-6
6.3	10BASE-Tのシステム構成	6-8
6.4	ネットワーク構成部品の設置、配線、および設定	6-10
6.4.1	同軸ケーブルの配線	6-10

6.4.2	トランシーバの設置および配線	6-11
6.4.3	同軸コネクタの取り付け	6-15
6.4.4	タップコネクタの取り付け	6-17
6.4.5	トランシーバケーブルの取り付け	6-18
6.4.6	ターミネータの取り付け	6-19
6.4.7	リピータの設置と取り付け	6-19
6.4.8	システムの接地	6-20
6.4.9	シングルポートトランシーバの設定	6-22
6.4.10	マルチポートトランシーバの設定および表示	6-23
6.5	システム定義情報	6-25
6.5.1	物理アドレス	6-25
6.5.2	IPアドレス	6-25
6.5.3	サブネットマスク	6-27
6.5.4	経路情報	6-27
6.6	ET.NETモジュールのメモリマップ	6-29
7	保 守	7-1
7.1	保守点検	7-2
7.1.1	モジュールの交換、増設	7-3
7.2	トラブルシューティング	7-5
7.2.1	手 順	7-5
7.2.2	故障かなと思ったら	7-6
7.3	エラーと対策	7-7
7.3.1	インディケータ表示メッセージ	7-7
7.3.2	エラー対処方法	7-8
7.3.3	ソケットハンドラ検出のエラーコード	7-10
7.3.4	経路情報設定エラーテーブル	7-13
7.4	トラブル調査書	7-15

目 次

図 3-1	オプションモジュールの実装	3-3
図 3-2	アース配線	3-6
図 3-3	10BASE-5通信ケーブルの配線	3-7
図 3-4	10BASE-T通信ケーブルの配線	3-8
図 4-1	[通信種類] 画面 (ET.NETシステム)	4-7
図 4-2	[[S10V] ET.NET] 画面	4-7
図 4-3	[ET.NET] 画面	4-10
図 4-4	[経路情報] 画面	4-11
図 4-5	[[S10V] ET.NET] 画面 ([RS-232C] ラジオボタンをクリック)	4-12
図 4-6	[[S10V] ET.NET] 画面 ([イーサネット] ラジオボタンをクリック)	4-12
図 4-7	モジュール設定画面	4-14
図 4-8	[[S10V] ET.NET] 画面 ([オフライン] ラジオボタンをクリック)	4-15
図 4-9	[ファイルを開く] 画面	4-15
図 4-10	「新規作成確認」ダイアログボックス	4-16
図 4-11	[[オフライン] モジュール設定] 画面	4-16
図 4-12	[[オンライン] IPアドレス設定] 画面	4-17
図 4-13	「リセット確認」メッセージ	4-18
図 4-14	「ファイル保存確認」メッセージ	4-19
図 4-15	「保存しました」メッセージ	4-19
図 4-16	「ファイルに保存しないで終了します。」メッセージ	4-20
図 4-17	[[オンライン] 経路情報] 画面	4-21
図 4-18	[[オンライン] モジュール設定] 画面 (IPアドレス設定 ボタンをクリック)	4-22
図 4-19	[[オンライン] IPアドレス設定] 画面 (読み込み ボタンをクリック)	4-22
図 4-20	[ファイルを開く] 画面	4-23
図 4-21	[[オンライン] IPアドレス設定] 画面 (選択したIPアドレス設定情報ファイルの内容表示)	4-23
図 4-22	[[オンライン] モジュール設定] 画面 (IPアドレス設定 ボタンをクリック)	4-24
図 4-23	[[オンライン] IPアドレス設定] 画面 (保存 ボタンをクリック)	4-24
図 4-24	[名前を付けて保存] 画面	4-25
図 4-25	[[オンライン] モジュール設定] 画面 (IPアドレス設定 ボタンをクリック)	4-26
図 4-26	[[オンライン] IPアドレス設定] 画面 (印刷 ボタンをクリック)	4-26
図 4-27	[印刷] ダイアログボックス	4-27
図 4-28	IPアドレス設定情報印刷例	4-27
図 4-29	[[オンライン] モジュール設定] 画面 (IPアドレス設定 ボタンをクリック)	4-28

図4-30	[[オンライン] IPアドレス設定] 画面 (CSV出力 ボタンをクリック)	4-28
図4-31	[名前を付けて保存] 画面 (CSV出力)	4-29
図4-32	[[オンライン] モジュール設定] 画面	4-30
図4-33	[ファイルを開く] 画面	4-31
図4-34	「不一致はありません。」メッセージ	4-31
図4-35	「比較不一致」メッセージ例	4-32
図5-1	ET.NETのソフトウェア構成	5-2
図5-2	ユーザプログラムからのソケットハンドラ呼び出し方法	5-5
図5-3	TCP/IPプログラム使用時のソケットハンドラ発行手順例	5-30
図5-4	UDP/IPプログラム使用時のソケットハンドラ発行手順例	5-31
図5-5	システム構成例	5-34
図5-6	CPU01側プログラムのフローチャート	5-35
図5-7	CPU02側プログラムのフローチャート	5-39
図5-8	システム構成例	5-42
図5-9	CPU01側プログラムのフローチャート	5-43
図5-10	CPU02側プログラムのフローチャート	5-48
図6-1	ネットワーク構成部品	6-3
図6-2	10BASE-5の最小構成	6-4
図6-3	10BASE-5の中規模構成	6-4
図6-4	10BASE-5の大規模構成	6-5
図6-5	10BASE-Tのシステム構成	6-8
図6-6	ハブのみによる10BASE-Tの構成例	6-8
図6-7	壁面設置例 (1)	6-12
図6-8	壁面設置例 (2)	6-13
図6-9	壁面設置例 (3)	6-13
図6-10	壁面設置例 (4)	6-13
図6-11	ボックス内設置例 (1)	6-14
図6-12	ボックス内設置例 (2)	6-14
図6-13	タップコネクタ組み立て	6-17
図6-14	コネクタとトランシーバの接続	6-18
図6-15	トランシーバケーブルの取り付け	6-18
図6-16	ターミネータの取り付け	6-19
図6-17	リピータの取り付け	6-19
図6-18	リピータの設置	6-20
図6-19	メモリマップ	6-29
図7-1	トラブルシューティング手順	7-5

図 7-2 エラーフリーズ情報	7-9
図 7-3 スタックフレームの内容	7-9

表 目 次

表 5-1	ソケットハンドラー一覧	5-6
表 6-1	ネットワーク構成部品一覧	6-2
表 6-2	10BASE-5システム構成上のパラメータ	6-5
表 6-3	SQEスイッチの設定	6-22
表 6-4	切り替えスイッチの設定	6-24
表 7-1	保守点検項目	7-2
表 7-2	S10mini CPUモジュール表示メッセージ	7-7
表 7-3	エラーメッセージ内容	7-8
表 7-4	ソケットハンドラ検出のエラーコード	7-10

このページは白紙です。

1 仕 様

1 仕 様

1. 1 用 途

ET.NETモジュール（型式：LQE520）は、S10V LPUモジュールまたはS10mini CPUモジュールと組み合わせ、IEEE802.3仕様に準拠したローカルエリアネットワーク経由で、TCP/IPまたはUDP/IPプロトコルによる通信をします。

- S10miniのCPUユニットで使用する場合、S10mini専用のET.NETモジュール（LQE020）と混在させることができます（S10VのLPUユニットでLQE020を使用することはできません）。
- S10VのLPUユニットで使用する場合、ET.NETモジュールのLQE520とLQE720を混在させることはできません。

1. 2 仕 様

1. 2. 1 一般仕様

項 目	仕 様
型 式	LQE520
マウントベース（*）への最大実装枚数	S10mini：2モジュール／CPU（左詰めで実装） S10V：2モジュール／LPU（左詰めの必要なし）
質 量	240g

（*）実装できるマウントベースの型式は、「3. 2 マウントベース」を参照してください。

1. 2. 2 通信仕様

項 目	仕 様
伝送方式	直列伝送（ビットシリアル伝送）
電氣的インタフェース	IEEE802.3準拠（CSMA/CD準拠）
符号化方式	マンチェスタ符号方式
プロトコル	TCP/IP, UDP/IP
接続台数	10BASE-5：最大100台／セグメント 10BASE-T：n台／ハブ（nはハブに依存）
ステーション台数	最大1,024台／ネットワーク
通信ケーブル	10BASE-5同軸ケーブル：最長500m／セグメント 10BASE-5トランシーバケーブル：最長50m 10BASE-Tツイストペアケーブル：最長100m／セグメント
データ転送速度	10Mbps

通 知

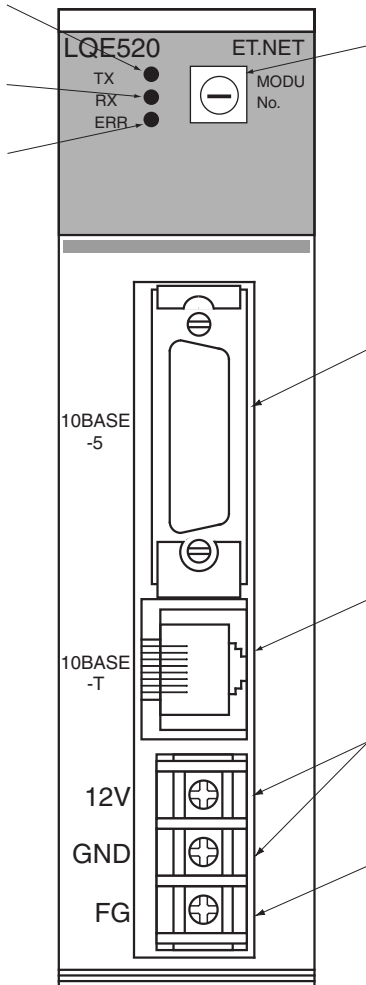
- 弊社提供ソフトウェアを改変して使用した場合に発生した事故や損害につきましては、弊社は責任を負いかねますのでご了承ください。
- 弊社提供以外のソフトウェアを使用した場合の信頼性については、弊社は責任を負いかねますのでご了承ください。
- ファイルのバックアップ作業を日常業務に組み入れてください。ファイル装置の障害、ファイルアクセス中の停電、誤操作、その他何らかの原因によりファイルの内容を消失することがあります。このような事態に備え、計画的にファイルのバックアップを取っておいてください。
- この製品は、産業廃棄物として専門の処理業者に廃棄を依頼してください。
- このモジュールの近くでは、トランシーバ、携帯電話等を使用しないでください。近くでトランシーバ、携帯電話等を使用しますとノイズにより誤動作、システムダウンとなる恐れがあります。
- モジュールの故障などでメモリの内容が破壊されることがあります。重要なデータは必ずバックアップを取っておいてください。
- システムの構築やプログラムの作成などは、このマニュアルの記載内容をよく読み、書かれている指示や注意を十分理解してから行ってください。誤操作により、システムが故障することがあります。
- このマニュアルは、必要なときすぐに参照できるよう、手近なところに保管してください。
- このマニュアルの記載内容について、疑問点または不明点がございましたら、販売店までお知らせください。
- お客様の誤操作に起因する事故発生や損害につきましては、弊社は責任を負いかねますのでご了承ください。
- 非常停止回路、インタロック回路などは、この製品の外部で構成してください。この製品の故障により、機械の破損や事故の恐れがあります。

このページは白紙です。

2 各部の名称と機能

2 各部の名称と機能

2. 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	回線上にデータが存在する場合に点灯します。																													
③	ERR LED	ハードウェア異常時に点灯します。																													
④	モジュールNo. (MODU No.) 設定スイッチ	<p>メインモジュールとサブモジュール、および通信ポートの種類を設定します。電源を切った状態で設定してください。</p> <table border="1"> <thead> <tr> <th colspan="2">モジュールNo. 設定</th> <th rowspan="2">内 容</th> </tr> <tr> <th>メイン</th> <th>サブ</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> <td>10BASE-5での通信</td> </tr> <tr> <td>2</td> <td>3</td> <td>10BASE-Tでの通信</td> </tr> <tr> <td>4</td> <td>5</td> <td>10BASE-Tでツールシステムとの通信 (*1)</td> </tr> <tr> <td>6</td> <td>7</td> <td>エラー (*2)</td> </tr> <tr> <td>8</td> <td>9</td> <td>エラー (*2)</td> </tr> <tr> <td>A</td> <td>B</td> <td>エラー (*2)</td> </tr> <tr> <td>C</td> <td>D</td> <td>保守用につき設定禁止</td> </tr> <tr> <td>E</td> <td>F</td> <td>保守用につき設定禁止</td> </tr> </tbody> </table> <p>モジュールNo.設定スイッチを4または5にした場合、IPアドレスは以下の値になります。 IPアドレス：192.192.192.001</p>	モジュールNo. 設定		内 容	メイン	サブ	0	1	10BASE-5での通信	2	3	10BASE-Tでの通信	4	5	10BASE-Tでツールシステムとの通信 (*1)	6	7	エラー (*2)	8	9	エラー (*2)	A	B	エラー (*2)	C	D	保守用につき設定禁止	E	F	保守用につき設定禁止
モジュールNo. 設定		内 容																													
メイン	サブ																														
0	1	10BASE-5での通信																													
2	3	10BASE-Tでの通信																													
4	5	10BASE-Tでツールシステムとの通信 (*1)																													
6	7	エラー (*2)																													
8	9	エラー (*2)																													
A	B	エラー (*2)																													
C	D	保守用につき設定禁止																													
E	F	保守用につき設定禁止																													
⑤	10BASE-5 コネクタ	10BASE-5で通信するためのコネクタです。																													
⑥	10BASE-T コネクタ	10BASE-Tで通信するためのコネクタです。																													
⑦	電源供給端子	10BASE-5で通信する際に接続するトランシーバの電源を供給する端子です。																													
⑧	FG端子	接地用端子です。「3. 4 アース配線」に従って接地してください。																													

(*1) モジュールNo.設定スイッチをこの設定にすると、経路設定情報エラーが発生する場合があります。詳しくは、「7. 3. 4 経路情報設定エラーテーブル」を参照してください。

(*2) エラーの確認方法は、「7. 3 エラーと対策」を参照してください。

通 知

モジュールNo.設定スイッチは、電源を切った状態で操作してください。動作中に操作すると誤動作の原因になります。

3 実装と配線

3 実装と配線

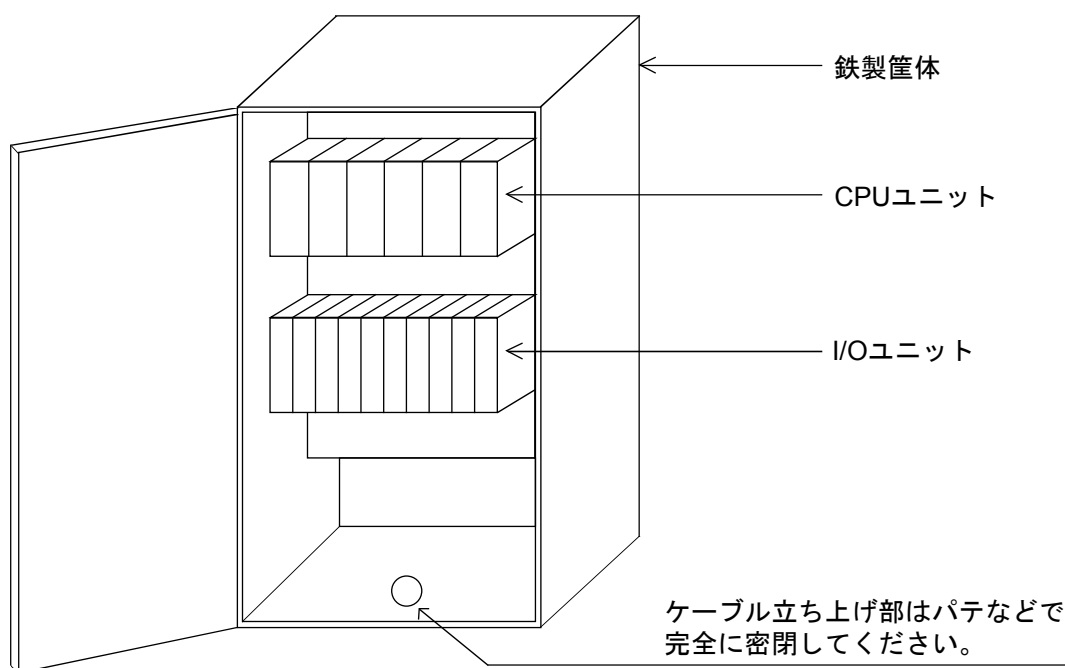
3. 1 PCsのご使用にあたり

PCs (Programmable Controllers) は電子回路、プロセッサ技術を応用した製品です。このため次のことには特に配慮してください。

- (1) システム構築に際しては、特に最大定格、動作電源電圧範囲、放熱特性、実装条件、およびその他諸条件は、このマニュアルに記載されている保証範囲内で使用してください。保証範囲を超えて使用された場合の故障および事故につきましては、弊社はその責任を負いません。

また、保証範囲内の使用であっても、弊社製品について予測される故障発生率、故障モードを考慮して、弊社製品の動作が原因でシステムが人身事故、火災事故、その他の拡大損害を生じないようにフェールセーフなどのシステム上の対策を講じてください。

- (2) PCsは防火、防塵、防水構造ではありません。設置の際には下図のように鉄製の防塵、防水筐体を実装してください。



⚠ 注意

- 故障の原因になりますので、水漏れの危険のあるところでは、防滴構造の筐体内に収納して使用してください。
- 通電中モジュールに触れますと静電気によりモジュールの誤動作、破損する恐れがあります。通電中はモジュールに触れないでください。やむを得ず触れる場合は、触れる前に人体の静電気を放電してください。また、非通電中にモジュールの各種スイッチの設定、ケーブルの取り付け／取り外し、コネクタの抜き差しなどをする前にも、人体の静電気を放電してください。

3.2 マウントベース

ET.NETモジュールは、マウントベースに実装して使用します。実装できるマウントベースの種類は、以下のとおりです。

シリーズ	名 称	型 式
S10V	4スロットLPUマウントベース	HSC-1540
	8スロットLPUマウントベース	HSC-1580
S10mini	2スロットCPUマウントベース	HSC-1020
	4スロットCPUマウントベース	HSC-1040
	8スロットCPUマウントベース	HSC-1080

3.3 モジュールの実装

オプションモジュールは、以下のようにマウントベースのオプションスロット（スロットナンバ0～7）に実装してください。

- S10miniシリーズでは、オプションモジュールとCPUモジュールとの間にI/Oモジュールが入らないように左詰め、さらにオプションモジュール間に空きスロットがないように実装してください。
- S10Vシリーズでは、実装位置や空きスロットに制約はありません。
- ET.NETモジュールのLQE520とLQE720を混在させることはできません。
- ET.NETモジュールは、CMUモジュールとペアで実装してください（ツールシステムとの通信を行う場合を除く）。

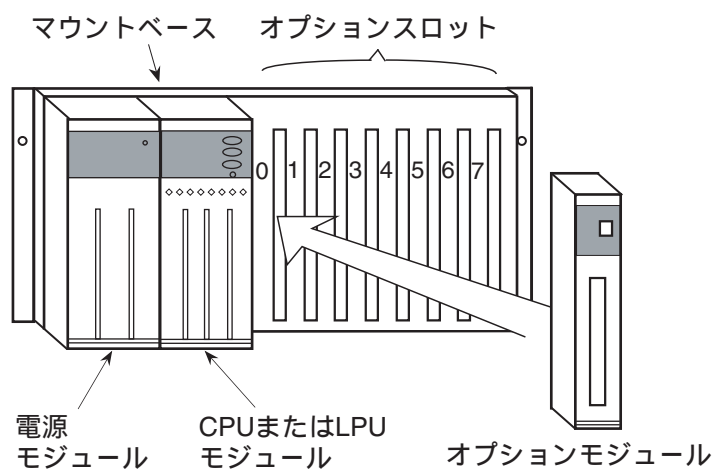


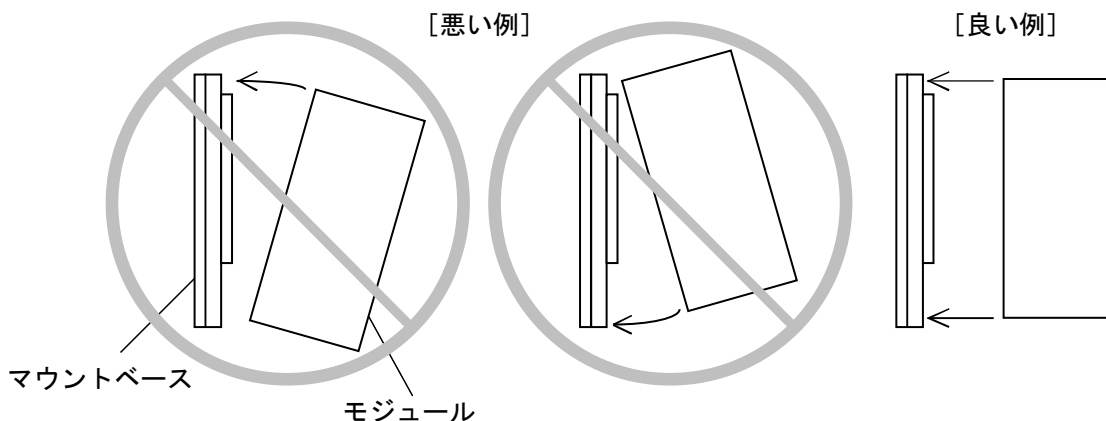
図3-1 オプションモジュールの実装

警告

- 発煙、異臭などがあった場合は、ただちに電源を切って原因を調査してください。
- このマニュアルに記載されていない設置、配線、取り扱い、および内部の改造はしないでください。これらに起因する弊社装置と周辺機器の破損および人身災害について、弊社は一切の責任を負いません。
- 通電中は端子台やコネクタのピンに絶対に触れないでください。通電中に端子台やコネクタのピンに触れると感電する恐れがあります。

注意

- コネクタにほこりなどが付着して接触不良が発生する可能性があります。装置の開梱後、ただちに設置および配線をしてください。
- モジュールが破損する恐れがあります。モジュールの取り付け／取り外しをするときは、以下の点に注意してください。
 - ・モジュールをマウントベースのコネクタに取り付ける前に、コネクタのピンの曲がりや折れはないか、ピンが一直線上に並んでいるか、またピンにゴミなどが付着していないかを確認してください。
 - ・モジュールは、以下に示すようにマウントベースの垂直面に沿って平行移動してください。モジュールを傾けたまま、コネクタへ取り付けまたはコネクタから取り外しすると、コネクタのピンが損傷する恐れがあります。
 - ・筐体の構造上、マウントベースが頭上に配置されている場合、モジュールは脚立などを使用してまっすぐに取り付けてください。斜めに取り付けるとコネクタを破損する恐れがあります。



 注 意

- マニュアルに従って取り付けをしてください。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。
- 静電気によりモジュールが破損する恐れがあります。作業する前に、人体の静電気を放電してください。
- ねじは確実に締め付けてください。締め付けが不十分な場合、誤動作や発煙、発火を引き起こす原因になります。

通 知

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。

3 実装と配線

3.4 アース配線

(1) 10BASE-5時のアース配線

ET.NETモジュール正面のFG端子を、下図のとおり接地してください。線径や線長は、CPUまたはLPUのマニュアルを参照してください。

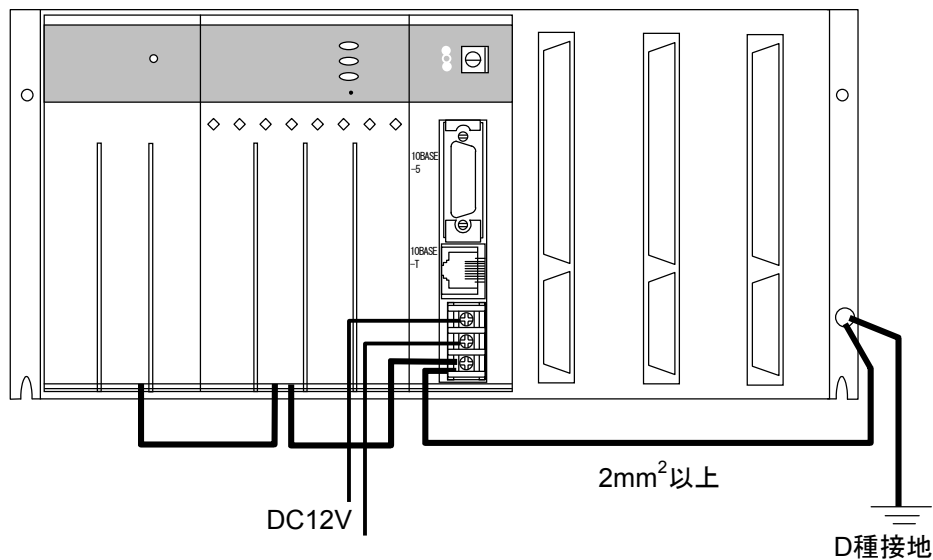


図3-2 アース配線

(2) 10BASE-T時のアース配線

ET.NETモジュールへのアース配線は不要です。

危険

- 10BASE-5で通信する場合は、ET.NETモジュールのFG端子を、マウントベースのFG端子経由でD種接地してください。
- アース線は、線径 2mm^2 以上のものを使用してください。

3.5 通信ケーブルの配線

(1) 10BASE-5通信ケーブルの配線

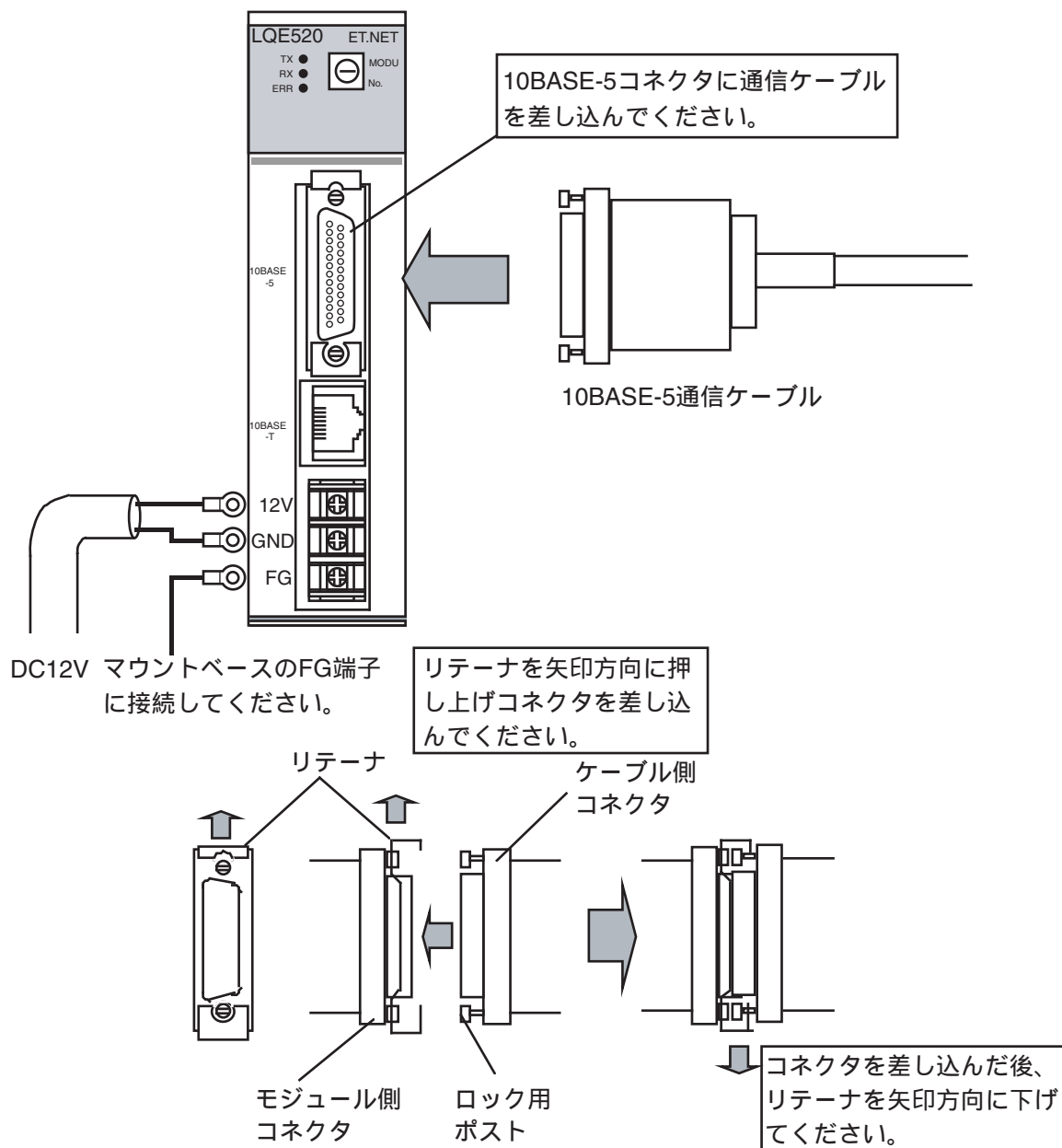


図3-3 10BASE-5通信ケーブルの配線

通 知

- 接触不良または断線により誤動作する恐れがあります。10BASE-5コネクタを接続した場合は、ロック用ポストがリテーナによってロックされているか確認してください。
- 通電中は10BASE-5コネクタに触れないでください。静電気などによりシステムが誤動作する可能性があります。

3 実装と配線

(2) 10BASE-T通信ケーブルの配線

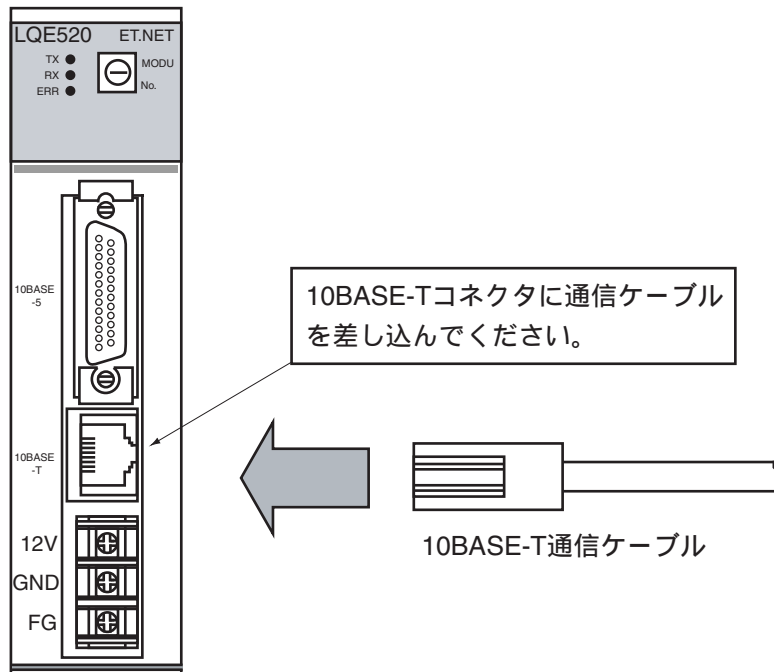


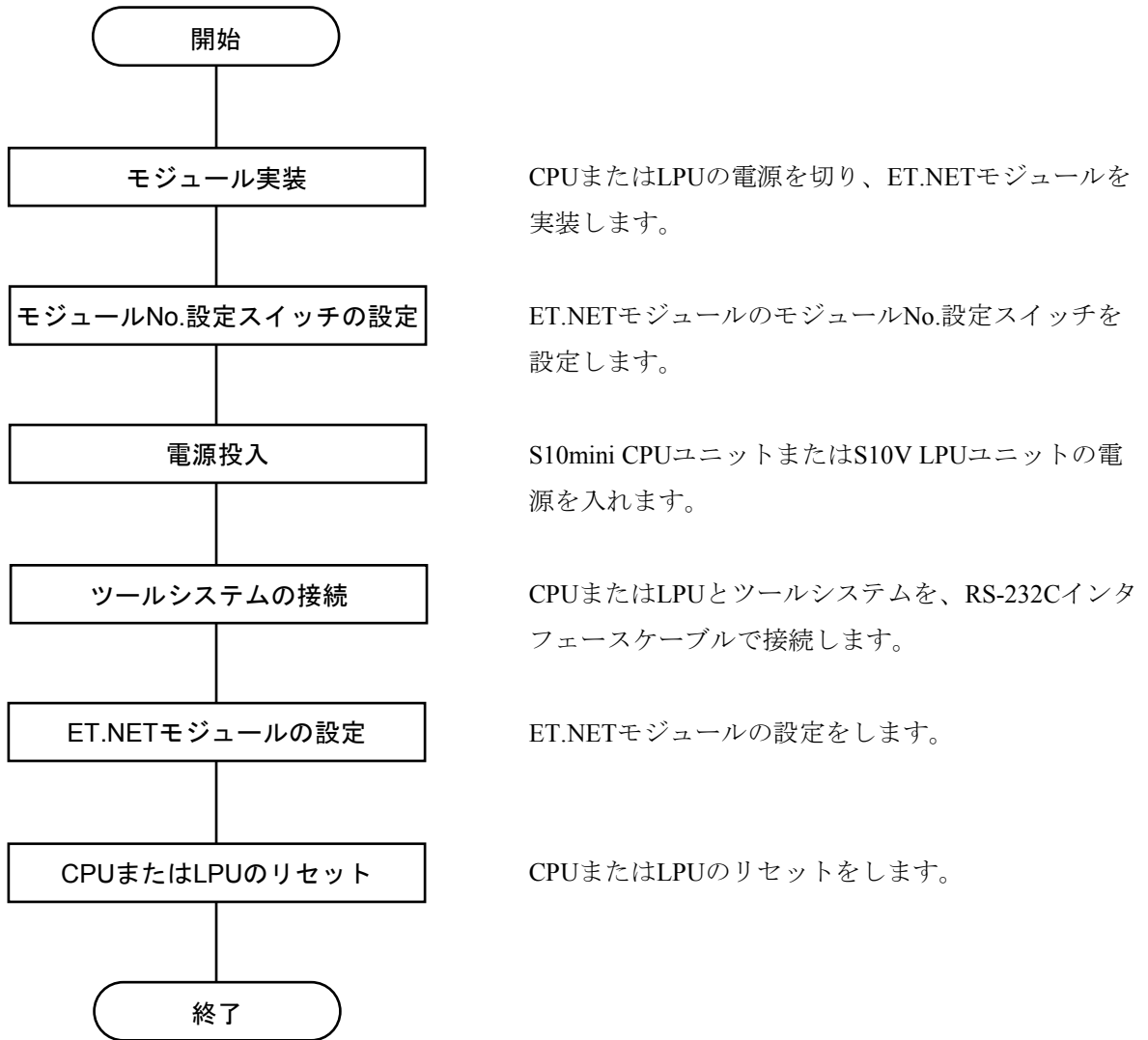
図 3 - 4 10BASE-T通信ケーブルの配線

4 オペレーション

4 オペレーション

4.1 立ち上げ手順

立ち上げは、以下の手順で行ってください。



4. 2 システムインストールと立ち上げ

4. 2. 1 インストール

(1) S10mini用ET.NETシステムインストール

まず、お手元のCDが正しいものか確認してください。

インストールするには、S10mini用ET.NETシステムCDのDISK1フォルダに格納されている“setup.exe”をダブルクリックし、セットアッププログラムを実行します。

インストール後、インストールしたプログラムの画面は表示されませんので、必要に応じてデスクトップにショートカットを貼り付けてください。

通 知

- S10mini用ET.NETシステムを動作させるためには、Microsoft® Internet Explorer 4.01以降が必要です。インストールされていない場合は、S10mini用ET.NETシステムが正常に動作しません。
- S10mini用ET.NETシステムをインストールする前に、すべてのWindows®上で作動するプログラムを必ず終了してください。ウイルス監視ソフトウェアなどメモリに常駐しているプログラムも必ず終了してください。終了せずにインストールすると、エラーが発生する場合があります。その場合は、「4. 2. 2 アンインストール」にて一旦アンインストールし、すべてのWindows®上で作動するプログラムを終了してから、再度S10mini用ET.NETシステムをインストールしてください。

(2) S10V用ET.NETシステムインストール

まず、お手元のCDが正しいものか確認してください。

インストールするには、S10V用ET.NETシステムCDのDISK1フォルダに格納されている“setup.exe”をダブルクリックし、セットアッププログラムを実行します。

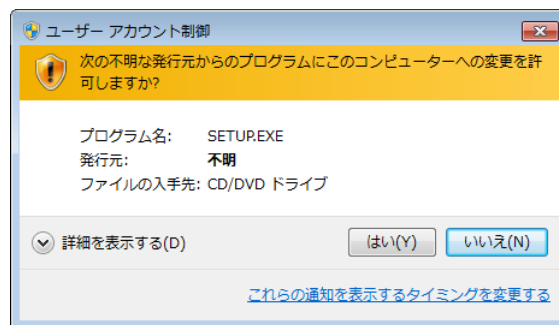
インストール後、インストールしたプログラムの画面は表示されませんので、必要に応じてデスクトップにショートカットを貼り付けてください。

通 知

- S10V用ET.NETシステムを動作させるためには、S10V基本システムが必要です。インストールされていない場合は、S10V用ET.NETシステムをインストールできません。
- S10V用ET.NETシステムをインストールする前に、すべてのWindows®上で作動するプログラムを必ず終了してください。ウイルス監視ソフトウェアなどメモリに常駐しているプログラムも必ず終了してください。終了せずにインストールすると、エラーが発生する場合があります。その場合は、「4. 2. 2 アンインストール」にて一旦アンインストールし、すべてのWindows®上で作動するプログラムを終了してから、再度S10V用ET.NETシステムをインストールしてください。

<Windows® 7 (32bit) 、Windows® 10 (32bit) でのインストール時の留意事項>

Windows® 7 (32bit) またはWindows® 10 (32bit) 搭載のパソコンへS10V用ET.NETシステムをインストールする場合は、パソコンの初期状態から最初に作成した管理者アカウントでWindows®にログオンし、S10V用ET.NETシステムCDのDISK1フォルダに格納されている“setup.exe”をダブルクリックします。“setup.exe”を起動すると、以下のダイアログボックスが表示される場合がありますので、**はい** ボタンをクリックしてセットアッププログラムを実行してください。



このシステムはユーザー別アプリケーションには対応していないため、必ず管理者アカウントでログオンしてからインストールしてください。

標準アカウントからユーザーアカウント制御(*)を使用してインストールしたり、標準アカウントからユーザーアカウント制御を使用して作成した管理者アカウントでログオンしてからは、正しくインストールされない場合があります。

パソコンの初期状態から最初に作成した管理者アカウントでログオンしてからインストールしてください。

インストールしたユーザーアカウントとは別のユーザーアカウントでログオンした際に、プログラムメニューの中にインストールしたプログラムが表示されない場合は、パソコンの初期状態から最初に作成した管理者アカウントでログオンし直し、プログラムを一度アンインストールしてから、再度インストールしてください。

また、新規にアカウントを作成する場合は、ユーザーアカウント制御を使用せずに管理者アカウントでログオンしてください。

(*) ユーザーアカウント制御は、標準アカウントに一時的に管理者権限を与えることができる機能です。

再インストールする際に、読み取り専用ファイルの検出メッセージが表示される場合は、**はい** ボタンをクリックして、上書きしてください。

4 オペレーション

4.2.2 アンインストール

ET.NETシステムのバージョンアップ時には、以下の手順でアンインストールしてください。

(1) Windows® 2000からのアンインストール

Windows®の [コントロールパネル] を開いてください。 [アプリケーションの追加と削除] をダブルクリックし、 [プログラムの変更と削除] タブで、 S10miniの場合は“ET.NETシステム”、 S10Vの場合は“S10V ET.NETシステム”を選択し、 ボタンをクリックしてください。

[ファイル削除の確認] 画面が表示されますので、 ボタンをクリックしてください。

(2) Windows® XPからのアンインストール

Windows®の [コントロールパネル] を開いてください。 [プログラムの追加と削除] をダブルクリックし、 [プログラムの変更と削除] タブで、 S10miniの場合は“ET.NETシステム”、 S10Vの場合は“S10V ET.NETシステム”を選択し、 ボタンをクリックしてください。

[ファイル削除の確認] 画面が表示されますので、 ボタンをクリックしてください。

(3) Windows® 7 (32bit) 、 Windows® 10 (32bit) からのアンインストール (S10Vのみ)

Windows®の [コントロールパネル] を開いてください。 [プログラムのアンインストール] をクリックし、“S10V ET.NETシステム”を選択し、 ボタンをクリックしてください。

[ファイル削除の確認] 画面が表示されますので、 ボタンをクリックしてください。

通 知

Windows®でアンインストール中に「共有ファイルを削除しますか？」の画面が表示された場合は、 ボタンをクリックして共有ファイルを削除しないでください。

4.2.3 システム立ち上げ

ET.NETシステムの立ち上げ方法を示します。

- ① ボタンから立ち上げる方法として、 S10miniの場合は [Hitachi S10] – [ET.NETシステム]、 S10Vの場合は [Hitachi S10V] – [S10V ET.NETシステム] を選択します。デスクトップに [ET.NETシステム] または [S10V ET.NETシステム] のショートカットを作成した場合は、ショートカットをダブルクリックしてください。
- ② [通信種類] 画面が表示されます。ET.NETシステムがS-7890-29の場合は、図4-1の [通信種類] 画面が表示されます。ET.NETシステムがS-7895-29の場合は、図4-2の [[S10V] ET.NET] 画面が表示されます。



図 4-1 「通信種類」画面 (ET.NETシステム)

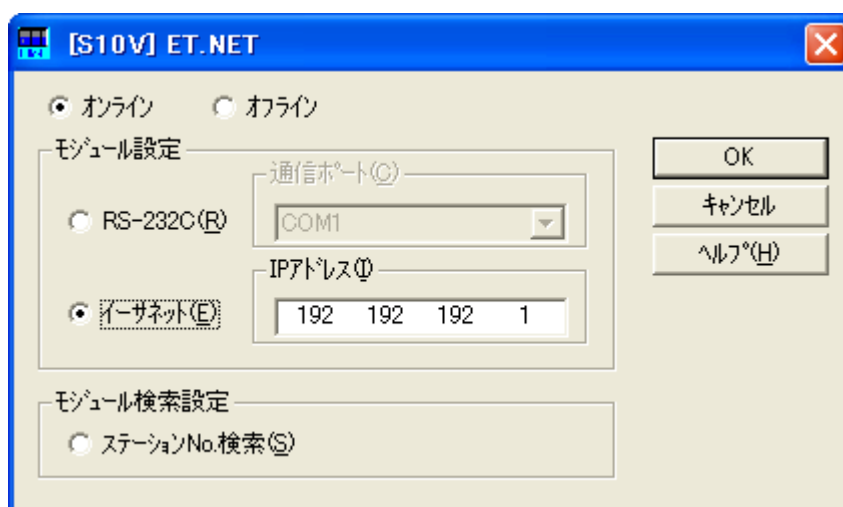
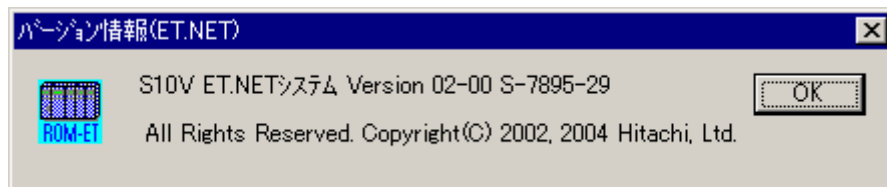


図 4-2 「[S10V] ET.NET」画面

この後の操作方法は、ET.NETシステムがS-7890-29の場合は「4.3 コマンド (S10mini用ET.NETシステム)」を参照してください。ET.NETシステムがS-7895-29の場合は「4.4 コマンド (S10V用ET.NETシステム)」を参照してください。

4 オペレーション

ET.NETシステムのバージョンは、[バージョン情報(ET.NET)]画面から確認してください。
[バージョン情報(ET.NET)]画面は、[ET.NET]画面または[[S10V] ET.NET]画面の左上のアイコンから「EtNetのバージョン情報」を選択することで表示されます。



4. 2. 4 システム終了

(1) S10mini用ET.NETシステムの場合

[ET.NET]画面(図4-3)の または ボタンをクリックしてください。

(2) S10V用ET.NETシステムの場合

[[S10V] ET.NET]画面(図4-2)の または ボタンをクリックしてください。

4. 3 コマンド (S10mini用ET.NETシステム)

4. 3. 1 接続PCsの変更

機能：PCsとパソコンの通信種類を設定します。

操作：以下に操作手順を示します。

- ① [ET.NET] 画面 (図 4-3) の ボタンをクリックするか、ET.NETシステム立ち上げ時に表示されます (図 4-1 参照)。
- ② [RS-232C] ラジオボタンをクリックし、[通信ポート] をプルダウンメニューから選択して、 ボタンをクリックしてください。
先の設定から変更をしない場合は、 ボタンをクリックしてください。

S10miniではGPIBをサポートしていません。[通信種類] 画面では「RS-232C」を選択してください。

4 オペレーション

4.3.2 IPアドレス、サブネットマスク設定

機能：メインモジュール、サブモジュールのIPアドレスおよびサブネットマスクを設定します。

操作：以下に操作手順を示します。

- ① [ET.NET] 画面を立ち上げます。「IPアドレス」および「サブネットマスク」を設定してください。



(この画面は、サブモジュール未実装時を示します。)

図 4-3 [ET.NET] 画面

- ② 設定が終了したら、**OK** ボタンをクリックしてください。設定しない場合は、**閉じる** ボタンをクリックしてください。

[ET.NET] 画面において、下表のとおり、ET.NETモジュール実装、未実装により、設定、表示方法が異なります。

実装状態	IPアドレス	サブネットマスク	経路情報	物理アドレス
実装	設定可能 (設定内容表示)	設定可能 (設定内容表示)	設定可能 (ボタン表示)	参照可能 (物理アドレス表示)
未実装	設定可能 (設定内容表示)	設定可能 (設定内容表示)	設定不可 (ボタン非表示)	参照不可 (FFFFFFFF表示)

4.3.3 経路情報設定

機能：ルーティングテーブルを設定します。

操作：以下に操作手順を示します。

- ① [ET.NET] 画面の **経路情報** ボタンをクリックしてください。
- ② [経路情報] 画面が表示されます。「相手局IPアドレス」および「ゲートウェイIPアドレス」を設定してください。

	相手局IPアドレス	ゲートウェイIPアドレス
デフォルト(F)	0 0 0 0	0 0 0 0
ユーザ1	0 0 0 0	0 0 0 0
ユーザ2	0 0 0 0	0 0 0 0
ユーザ3	0 0 0 0	0 0 0 0
ユーザ4	0 0 0 0	0 0 0 0
ユーザ5	0 0 0 0	0 0 0 0
ユーザ6	0 0 0 0	0 0 0 0
ユーザ7	0 0 0 0	0 0 0 0
ユーザ8	0 0 0 0	0 0 0 0
ユーザ9	0 0 0 0	0 0 0 0
ユーザ10(A)	0 0 0 0	0 0 0 0
ユーザ11(B)	0 0 0 0	0 0 0 0
ユーザ12(C)	0 0 0 0	0 0 0 0
ユーザ13(D)	0 0 0 0	0 0 0 0
ユーザ14(E)	0 0 0 0	0 0 0 0

図4-4 [経路情報] 画面

- ③ 設定が終了したら、**OK** ボタンをクリックしてください。設定しない場合は、**キャンセル** ボタンをクリックしてください。

4 オペレーション

4.4 コマンド (S10V用ET.NETシステム)

4.4.1 接続PCsの変更

機能：PCsとパソコンの通信種類を設定します。

操作：以下に操作手順を示します。

- ① ET.NETシステム立ち上げ画面からPCsとパソコンを接続する通信種類を選択してください。

・RS-232C接続のとき

[RS-232C] ラジオボタンをクリックし、プルダウンメニューから [通信ポート] を選択してください。“COM1”～“COM4”までの通信ポートが選択可能です。デフォルトは、“COM1”となります。



図4-5 [S10V] ET.NET] 画面 ([RS-232C] ラジオボタンをクリック)

・イーサネット接続のとき

[イーサネット] ラジオボタンをクリックし、接続するPCsのIPアドレスを入力してください。

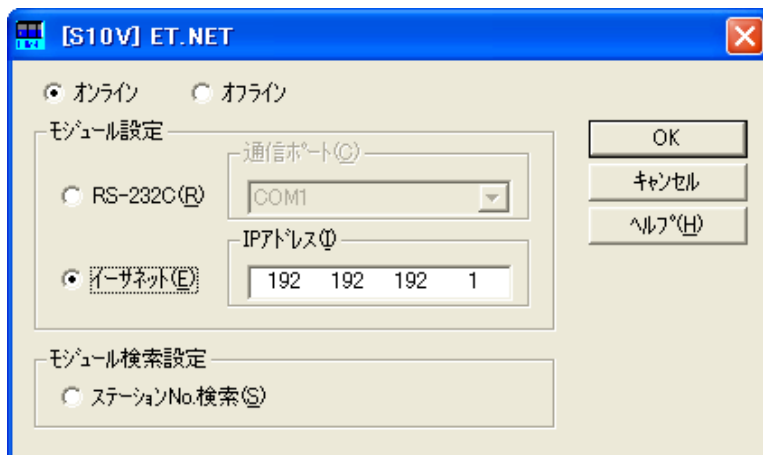


図4-6 [S10V] ET.NET] 画面 ([イーサネット] ラジオボタンをクリック)

- ② 設定が終了したら ボタンをクリックしてください。[モジュール設定] 画面が表示されます。

ステーションNo.検索はET.NET (LQE720) 専用の機能のため、ET.NET (LQE520) では使用できません。

4 オペレーション

4.4.2 モジュール設定

機能：目的の画面を表示します。オンラインモードとオフラインモードで使用できる機能が異なります。

操作：以下にオンラインモード、オフラインモード別に操作手順を示します。

(1) オンラインモード

- ① [[S10V] ET.NET] 画面の [RS-232C] ラジオボタンまたは [イーサネット] ラジオボタンをクリックし、**OK** ボタンをクリックしてください。
- ② [モジュール設定] 画面が表示されます。

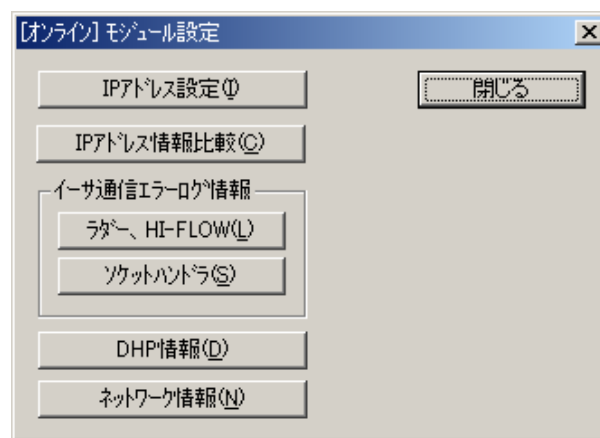


図 4-7 モジュール設定画面

- ③ ET.NETモジュールのIPアドレス情報を設定したい場合は、**IPアドレス設定** ボタンをクリックしてください。[[オンライン] IPアドレス設定] 画面が表示されます。
詳細は、「4.4.3 IPアドレス設定」を参照してください。
- ④ [モジュール設定] 画面を終了する場合は、**閉じる** ボタンをクリックしてください。
[[S10V] ET.NET] 画面に戻ります。

RS-232C接続とET.NET (LQE720) 以外とのイーサネット接続では、以下のコマンドをサポートしていませんので実行できません。

- ラダー、HI-FLOW (イーサ通信エラーログ情報)
- ソケットハンドラ (イーサ通信エラーログ情報)
- DHP情報
- ネットワーク情報

(2) オフラインモード

- ① [[S10V] ET.NET] 画面の [オフライン] ラジオボタンをクリックしてください。[OK] ボタンが [編集ファイル選択] ボタンに切り替わります。

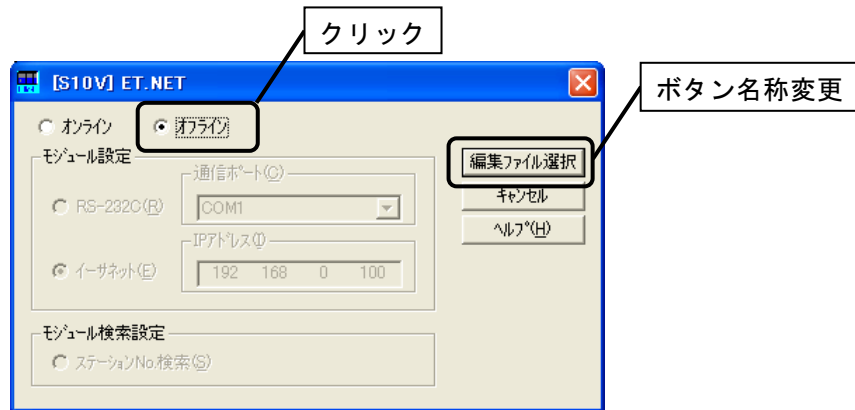


図 4-8 [[S10V] ET.NET] 画面（[オフライン] ラジオボタンをクリック）

- ② [編集ファイル選択] ボタンをクリックしてください。[ファイルを開く] 画面が表示されます。

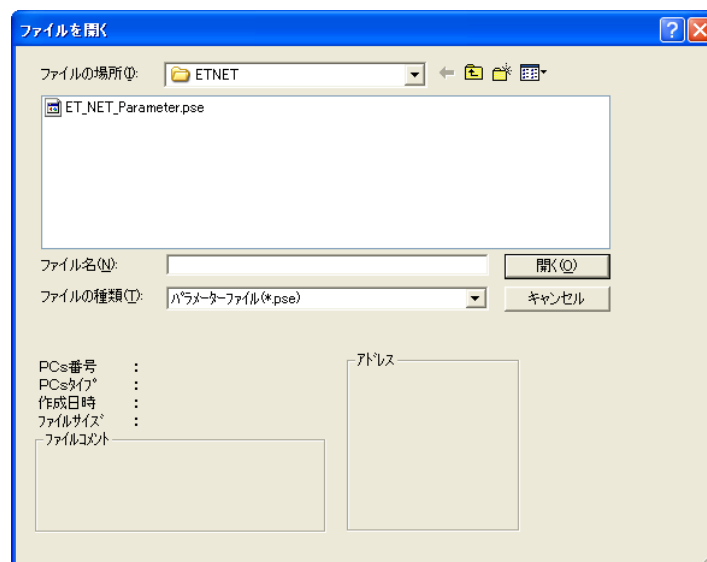


図 4-9 [ファイルを開く] 画面

4 オペレーション

既存のIPアドレス設定情報ファイルを編集する場合は、該当のファイルを選択してください。

[ファイルを開く]画面が閉じ、[[オフライン]モジュール設定]画面が表示されます。

新規に作成する場合は、そのフォルダに存在しない名称を入力してください。[ファイルを開く]画面が閉じ、「新規作成確認」ダイアログボックスが表示されます。

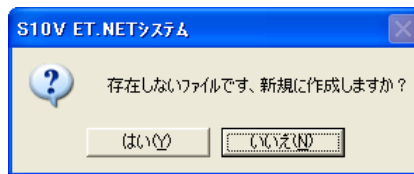


図 4-10 「新規作成確認」ダイアログボックス

はい ボタンをクリックすると、「新規作成確認」ダイアログボックスが閉じ、[[オフライン]モジュール設定]画面が表示されます。

いいえ ボタンをクリックすると、「新規作成確認」ダイアログボックスが閉じ、[ファイルを開く]画面に戻ります。

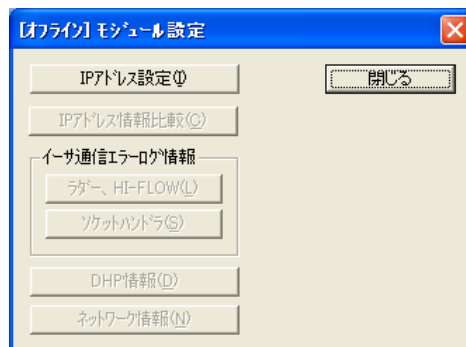


図 4-11 [[オフライン]モジュール設定]画面

- ③ ET.NETモジュールのIPアドレス情報を設定したい場合は、**IPアドレス設定** ボタンをクリックしてください。[[オフライン]IPアドレス設定]画面が表示されます。
詳細は、「4.4.3 IPアドレス設定」を参照してください。
- ④ [[オフライン]モジュール設定]画面を終了する場合は、**閉じる** ボタンをクリックしてください。[[S10V]ET.NET]画面に戻ります。

4. 4. 3 IPアドレス設定

機能：ET.NETモジュールのIPアドレス情報を設定します。オンラインモードとオフラインモードでは、**書込み** ボタンをクリックしたときの書き込み先が異なります。

オンラインモード：パソコンに接続された先のPCs

オフラインモード：[[S10V] ET.NET] 画面で選択したファイル

操作：以下に操作手順を示します。

(1) オンラインモード

- ① [[オンライン] モジュール設定] 画面の **IPアドレス設定** ボタンをクリックしてください。
[[オンライン] IPアドレス設定] 画面が表示されます。
- ② IPアドレス情報を設定してください。

図 4-12 [[オンライン] IPアドレス設定] 画面

・モジュール

設定するET.NETモジュールを選択してください。

選択項目	備考
ET.NET (メイン)	初期値
ET.NET (サブ)	

・IPアドレス/サブネットマスク

ET.NETモジュールのIPアドレスおよびサブネットマスクを設定してください。設定の詳細は「6. 5 システム定義情報」を参照してください。

・物理アドレス

ET.NETモジュールに割り付けられている48ビットのアドレスが表示されます。ET.NETモジュールが未実装の場合は、「00:00:00:00:00:00」または「FF:FF:FF:FF:FF:FF」が表示されません。

4 オペレーション

- ③ ルーティングテーブルを設定する場合は、**経路情報** ボタンをクリックしてください。[[オンライン] 経路情報] 画面が表示されますので、各経路の相手局アドレスとゲートウェイIPアドレスを設定してください。
詳細は、「4. 4. 4 経路情報設定」を参照してください。
- ④ IPアドレス情報の設定が終了したら、**書込み** ボタンをクリックしてください。設定しない場合は、**キャンセル** ボタンをクリックしてください。
- ⑤ **書込み** ボタンをクリックすると、以下の「リセット確認」メッセージが表示されます。

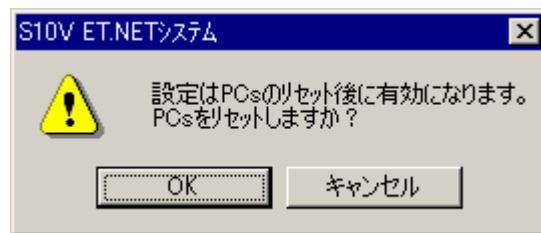
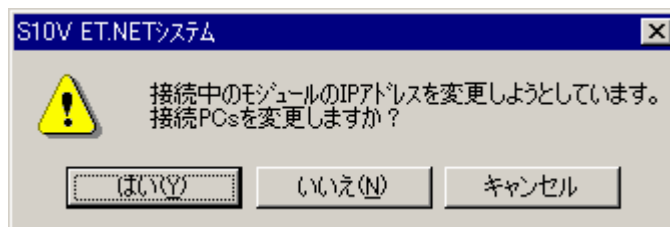


図 4-13 「リセット確認」メッセージ

OK ボタンをクリックしてPCsをリセットしてください。リセット終了後、設定したIPアドレスや経路情報が有効になります。なお、PCsに設定されるIPアドレスや経路情報は、選択されたメインまたはサブモジュールのどちらかだけです。**書込み** ボタンをクリックしたときに、選択されていない方のIPアドレスや経路情報は、PCsに設定されません。メインとサブ2つのモジュールのIPアドレスや経路情報をPCsに設定する場合、「ET.NET (メイン)」または「ET.NET (サブ)」を選択して **書込み** ボタンをクリックする操作をメインとサブモジュールごとに行ってください。

通 知

イーサネットケーブルで接続しているET.NETモジュールのIPアドレスを変更しようとした場合、以下のメッセージが表示されます。



PCsリセット後、変更したIPアドレスでPCsと接続しなさい場合は **はい** ボタンを、接続しなさい場合は **いいえ** ボタンを、IPアドレスの設定を中止する場合は **キャンセル** ボタンをクリックしてください。

(2) オフラインモード

- ① [[オフライン] モジュール設定] 画面の **IPアドレス設定** ボタンをクリックしてください。
[[オフライン] IPアドレス設定] 画面が表示されます。
- ② IPアドレス情報を設定してください（設定については、「(1) オンラインモードの②」を参照してください。ただし、物理アドレスは、「00:00:00:00:00:00」で表示されます）。
- ③ ルーティングテーブルを設定する場合は、**経路情報** ボタンをクリックしてください。[[オフライン] 経路情報] 画面が表示されますので、各経路の相手局アドレスとゲートウェイアドレスを設定してください。
- ④ IPアドレス情報の設定が終了したら、**書込み** ボタンをクリックしてください。
設定しない場合は、**キャンセル** ボタンをクリックしてください。
- ⑤ **書込み** ボタンをクリックすると、以下の「ファイル保存確認」メッセージが表示されます。

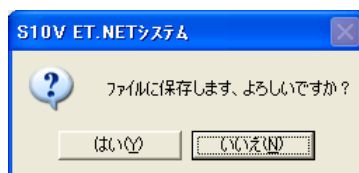


図4-14 「ファイル保存確認」メッセージ

はい ボタンをクリックしてください。「ファイル保存確認」メッセージが閉じ、IPアドレス設定情報（“メイン” および “サブ”）が、[[S10V] ET.NET] 画面で選択したファイルに書き込まれます。書き込みが完了すると、「保存しました」メッセージが表示されます。**OK** ボタンをクリックしてください。

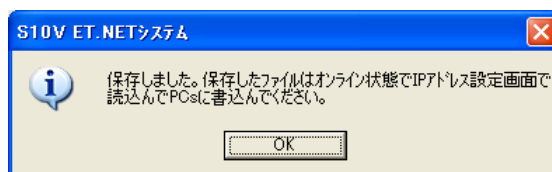


図4-15 「保存しました」メッセージ

編集した内容を保存しない場合は、「ファイル保存確認」メッセージの **いいえ** ボタンをクリックしてください。「ファイル保存確認」メッセージが閉じ、[[オフライン] IPアドレス設定] 画面に戻ります。

4 オペレーション

編集状態 (*) で [[オフライン] IPアドレス設定] 画面の **キャンセル** ボタンをクリックすると、「ファイルに保存しないで終了します。」メッセージが表示されます。

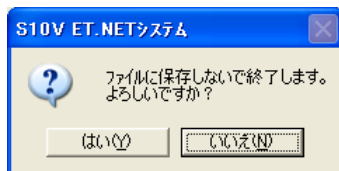


図 4-16 「ファイルに保存しないで終了します。」メッセージ

はい ボタンをクリック：編集した内容（経路情報を含みます）を破棄し、[[オフライン] モジュール設定] 画面に戻ります。

いいえ ボタンをクリック：[[オフライン] IPアドレス設定] 画面に戻ります。

(*) 編集状態の場合、[[オフライン] IPアドレス設定] 画面のタイトルの後ろに、「*」マークが付加されます。

編集状態でなければ、**キャンセル** ボタンをクリックしても、「ファイルに保存しないで終了します。」メッセージは表示されずに、[[オフライン] モジュール設定] 画面に戻ります。

4. 4. 4 経路情報設定

機能：ET.NETモジュールの経路情報を設定します。

操作：以下に操作手順を示します。

- ① [[オンライン] IPアドレス設定] 画面の **経路情報** ボタンをクリックしてください。
- ② [[オンライン] 経路情報] 画面が表示されます。経路情報を設定してください。

	相手局アドレス				ゲートウェイIPアドレス			
デフォルト(F)	0	0	0	0	0	0	0	0
経路1(1)	0	0	0	0	0	0	0	0
経路2(2)	0	0	0	0	0	0	0	0
経路3(3)	0	0	0	0	0	0	0	0
経路4(4)	0	0	0	0	0	0	0	0
経路5(5)	0	0	0	0	0	0	0	0
経路6(6)	0	0	0	0	0	0	0	0
経路7(7)	0	0	0	0	0	0	0	0
経路8(8)	0	0	0	0	0	0	0	0
経路9(9)	0	0	0	0	0	0	0	0
経路10(A)	0	0	0	0	0	0	0	0
経路11(B)	0	0	0	0	0	0	0	0
経路12(C)	0	0	0	0	0	0	0	0
経路13(D)	0	0	0	0	0	0	0	0
経路14(E)	0	0	0	0	0	0	0	0

図4-17 [[オンライン] 経路情報] 画面

- ・相手局アドレス
相手局のネットワークアドレスまたはIPアドレスを指定してください。
 - ・ゲートウェイIPアドレス（相手局アドレスを先に指定したとき、ネットワークアドレスが自動で表示されます。）
ゲートウェイのIPアドレスを指定してください。
- ③ 設定が終了したら **OK** ボタンをクリックしてください。設定しない場合は **キャンセル** ボタンをクリックしてください。

通 知

経路情報は [IPアドレス設定] 画面で **書込み** ボタンをクリックしたときにPCsまたはファイルに登録されます。そのため、[IPアドレス設定] 画面で **キャンセル** ボタンをクリックすると経路情報はPCsまたはファイルに登録されません。

4 オペレーション

4.4.5 IPアドレス設定情報ファイルの読み込み

機能：既存のIPアドレス設定情報ファイルを読み込み、その内容を [[オンライン] IPアドレス設定] 画面および [[オンライン] 経路情報] 画面に表示します (Ver-Rev番号：02-03以降でサポートしています。また、オンラインモードでのみ使用できます)。

操作：以下に操作手順を示します。

- ① PCsに接続した状態にしてください (「4.4.1 接続PCsの変更」参照)。
- ② [[オンライン] モジュール設定] 画面の **IPアドレス設定** ボタンをクリックしてください。

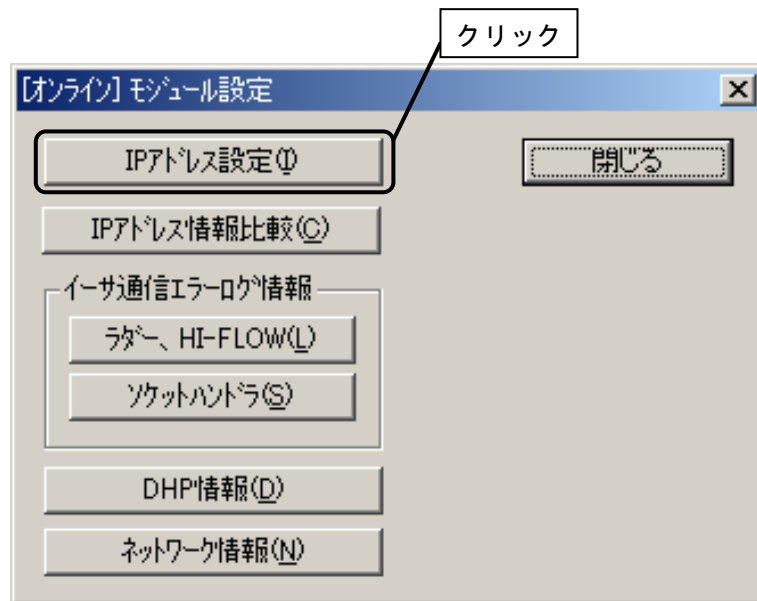


図 4-18 [[オンライン] モジュール設定] 画面 (**IPアドレス設定** ボタンをクリック)

- ③ [[オンライン] IPアドレス設定] 画面の **読み込み** ボタンをクリックしてください。

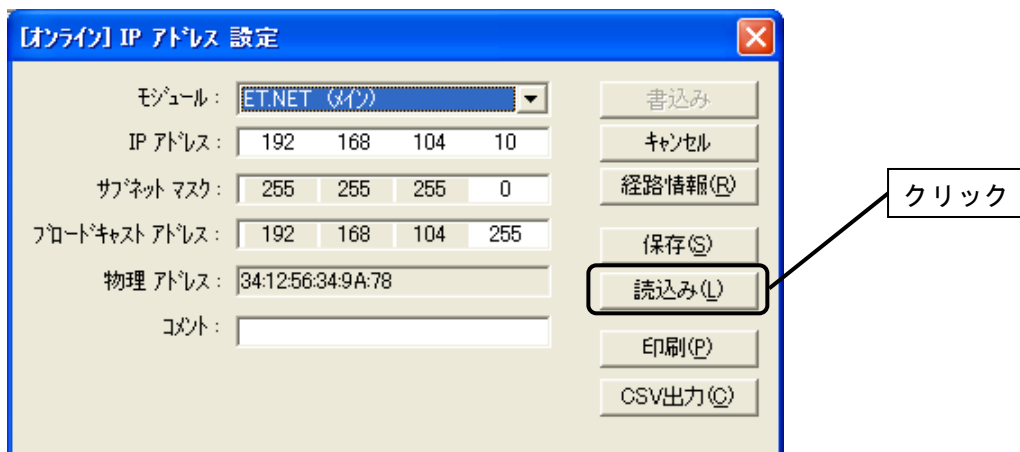


図 4-19 [[オンライン] IPアドレス設定] 画面 (**読み込み** ボタンをクリック)

- ④ [[オンライン] IPアドレス設定] 画面の **読み込み** ボタンをクリックしてください。
[ファイルを開く] 画面が表示されます。表示したいIPアドレス設定情報ファイルを選択して、**開く** ボタンをクリックしてください。



図 4-20 [[ファイルを開く] 画面

- ⑤ [ファイルを開く] 画面が閉じ、選択したIPアドレス設定情報ファイルの内容が [[オンライン] IPアドレス設定] 画面に表示されます。

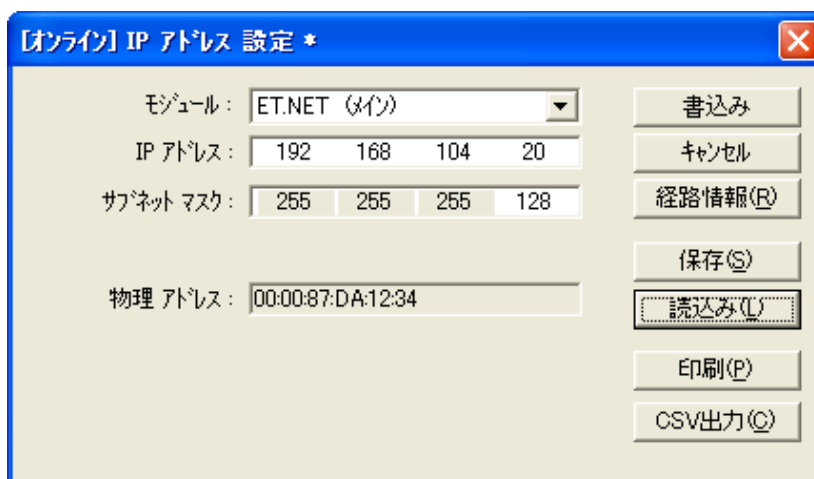


図 4-21 [[オンライン] IPアドレス設定] 画面 (選択したIPアドレス設定情報ファイルの内容表示)

[ファイルを開く] 画面でファイルを選択し、**開く** ボタンをクリックすると、[[オンライン] IPアドレス設定] 画面タイトルに「*」マークが付加されます。

4 オペレーション

4.4.6 IPアドレス設定情報のファイル保存

機能：表示中のIPアドレス設定情報（経路情報を含みます）をファイルに保存します（Ver-Rev番号：02-03以降でサポートしています。また、オンラインモードでのみ使用できます）。

操作：以下に操作手順を示します。

- ① PCsに接続した状態にしてください（「4.4.1 接続PCsの変更」参照）。
- ② [[オンライン] モジュール設定] 画面の **IPアドレス設定** ボタンをクリックしてください。

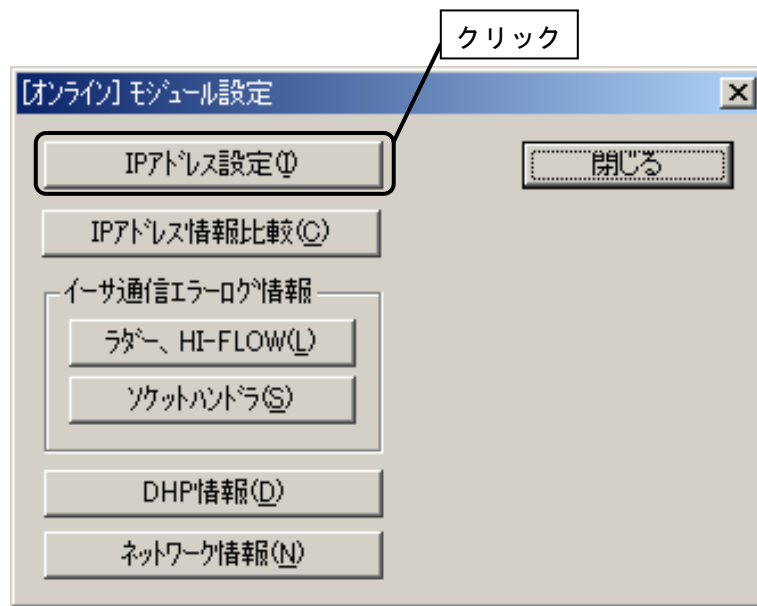


図 4-22 [[オンライン] モジュール設定] 画面（ **IPアドレス設定** ボタンをクリック）

- ③ [[オンライン] IPアドレス設定] 画面の **保存** ボタンをクリックしてください。



図 4-23 [[オンライン] IPアドレス設定] 画面（ **保存** ボタンをクリック）

- ④ [名前を付けて保存] 画面が表示されます。“保存する場所”と“ファイル名”を指定し、**保存** ボタンをクリックしてください。

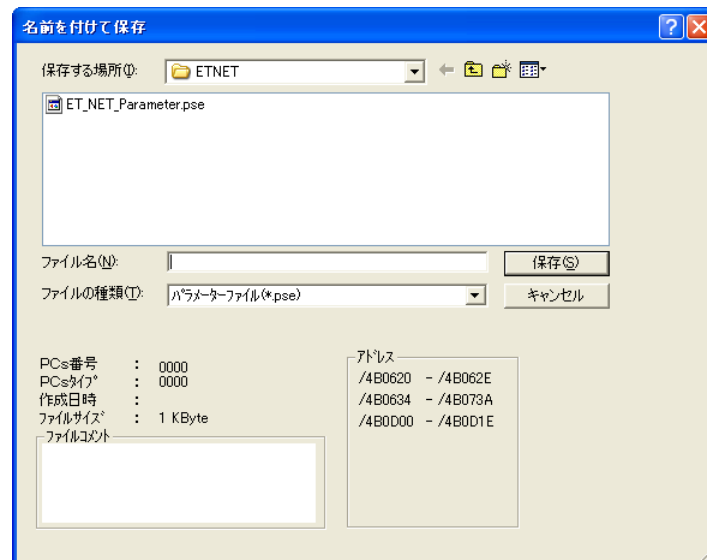


図 4-24 [名前を付けて保存] 画面

- ⑤ [名前を付けて保存] 画面が閉じ、表示中のIPアドレス設定情報（経路情報を含みます）を指定したファイルに保存します。

4 オペレーション

4.4.7 IPアドレス設定情報の印刷

機能：表示中のIPアドレス設定情報を印刷します（Ver-Rev番号：02-03以降でサポートしていません）。

操作：以下に操作手順を示します。

- ① オンライン時はPCsに接続した状態にしてください（「4.4.1 接続PCsの変更」参照）。オフライン時は編集ファイルを選択した状態にしてください。
- ② [[オンライン] モジュール設定] 画面の **IPアドレス設定** ボタンをクリックしてください。

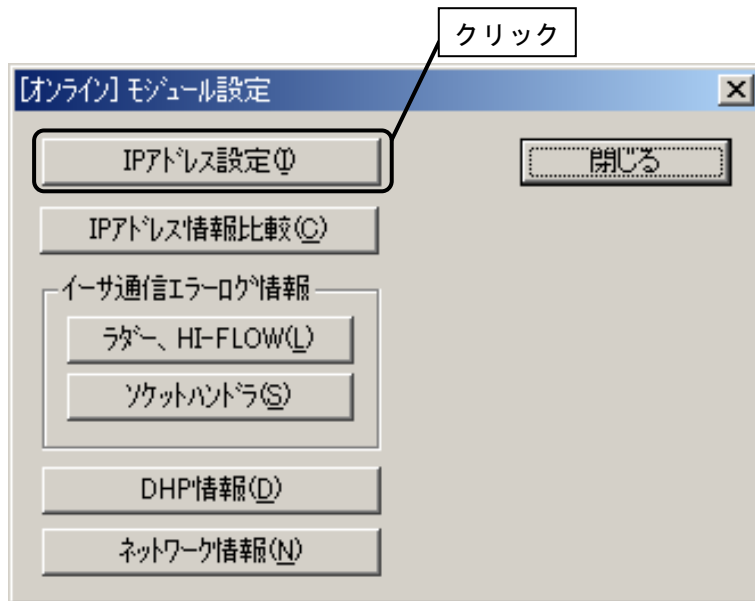


図 4-25 [[オンライン] モジュール設定] 画面（ **IPアドレス設定** ボタンをクリック）

- ③ [[オンライン] IPアドレス設定] 画面の **印刷** ボタンをクリックしてください。



図 4-26 [[オンライン] IPアドレス設定] 画面（ **印刷** ボタンをクリック）

- ④ [印刷] ダイアログボックスが表示されますので、出力先プリンタの指定やプロパティの設定などを行い、**OK** ボタンをクリックしてください。IPアドレス設定情報が印刷されます。

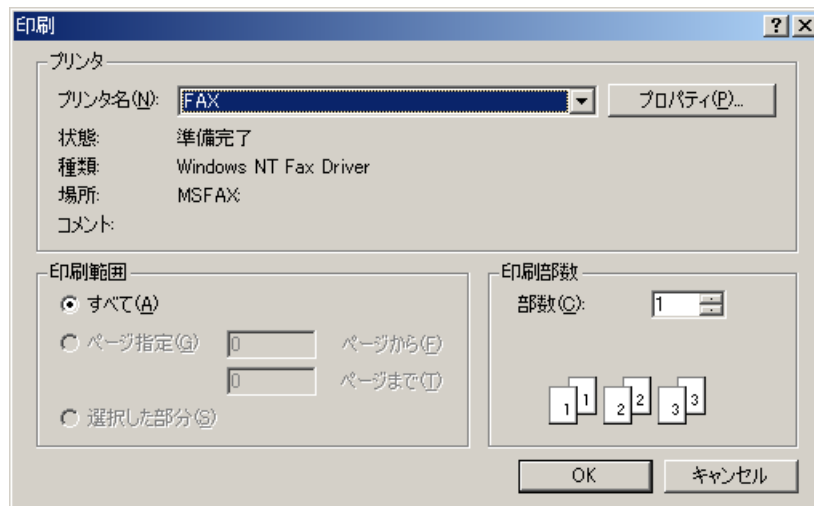


図 4-27 [印刷] ダイアログボックス

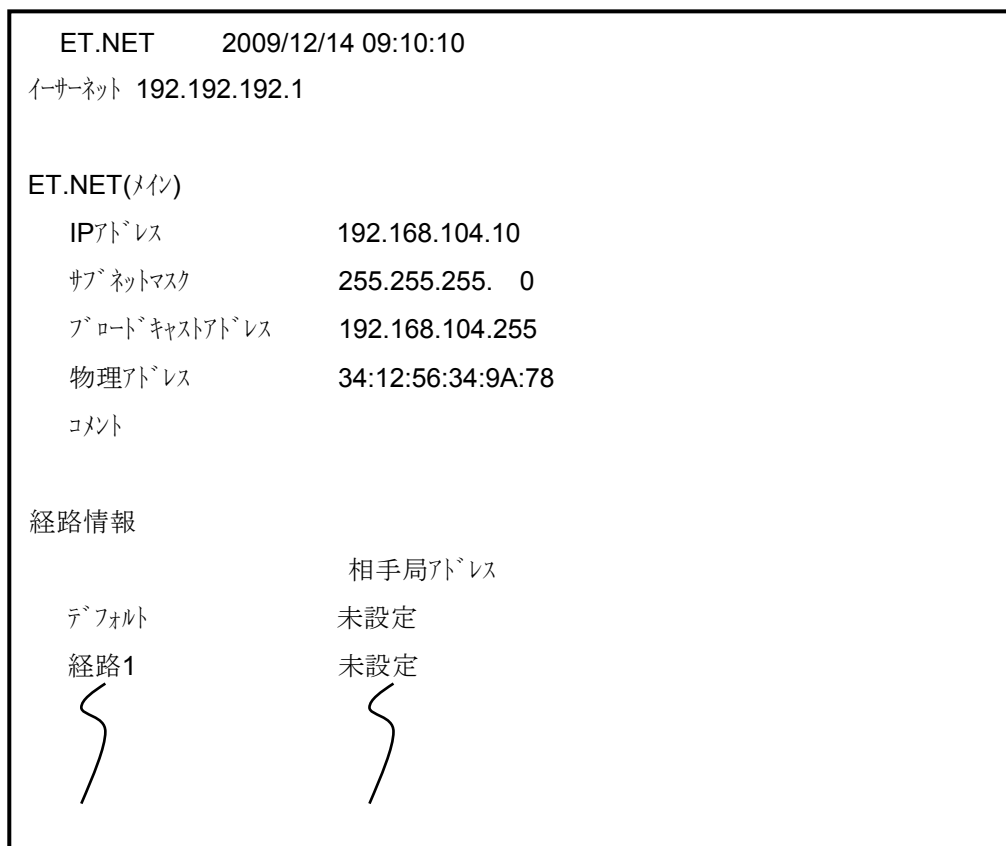


図 4-28 IPアドレス設定情報印刷例

4 オペレーション

4.4.8 IPアドレス設定情報のCSV出力

機能：表示中のIPアドレス設定情報をCSV形式でファイルに出力します（Ver-Rev番号：02-03以降でサポートしています）。

操作：以下に操作手順を示します。

- ① オンライン時はPCsに接続した状態にしてください（「4.4.1 接続PCsの変更」参照）。
オフライン時は編集ファイルを選択した状態にしてください。
- ② [[オンライン] モジュール設定] 画面の **IPアドレス設定** ボタンをクリックしてください。

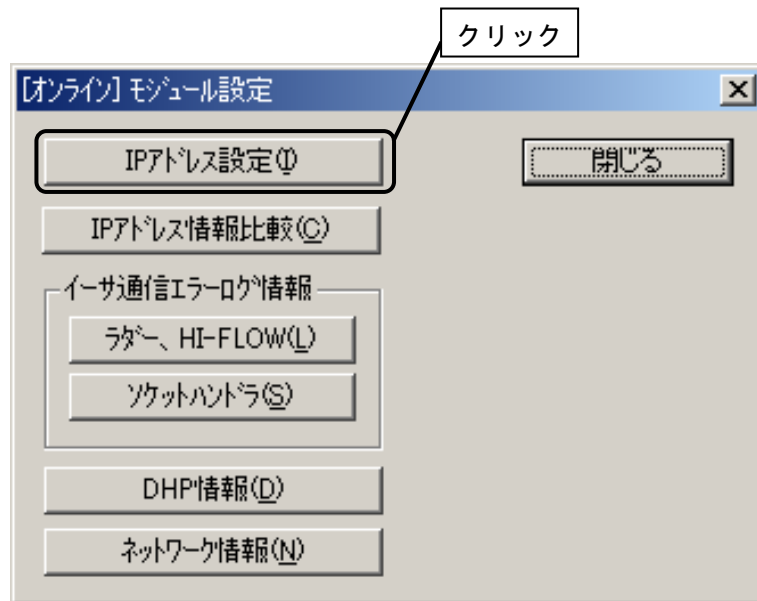


図 4-29 [[オンライン] モジュール設定] 画面（ **IPアドレス設定** ボタンをクリック）

- ③ [[オンライン] IPアドレス設定] 画面の **CSV出力** ボタンをクリックしてください。



図 4-30 [[オンライン] IPアドレス設定] 画面（ **CSV出力** ボタンをクリック）

- ④ [名前を付けて保存] 画面が表示されますので、“保存する場所”と“ファイル名”を指定して、**保存** ボタンをクリックしてください。IPアドレス設定情報がCSV形式で指定されたファイル名で保存されます。

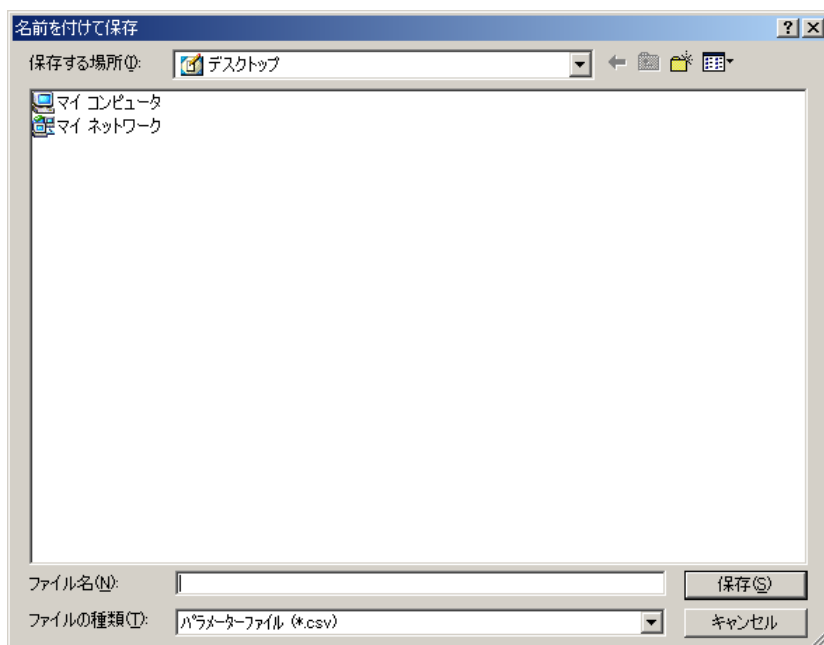


図 4-31 [名前を付けて保存] 画面 (CSV出力)

4 オペレーション

4.4.9 IPアドレス設定情報の比較

機能：接続先PCsに設定されているIPアドレス設定情報とユーザが選択したIPアドレス設定情報ファイルの内容を比較し、その結果を表示します（Ver-Rev番号：02-03以降でサポートしています。また、オンラインモードでのみ使用可能です）。

操作：以下に操作手順を示します。

- ① PCsに接続した状態にしてください（「4.4.1 接続PCsの変更」参照）。
- ② [[オンライン] モジュール設定] 画面の **IPアドレス情報比較** ボタンをクリックしてください。

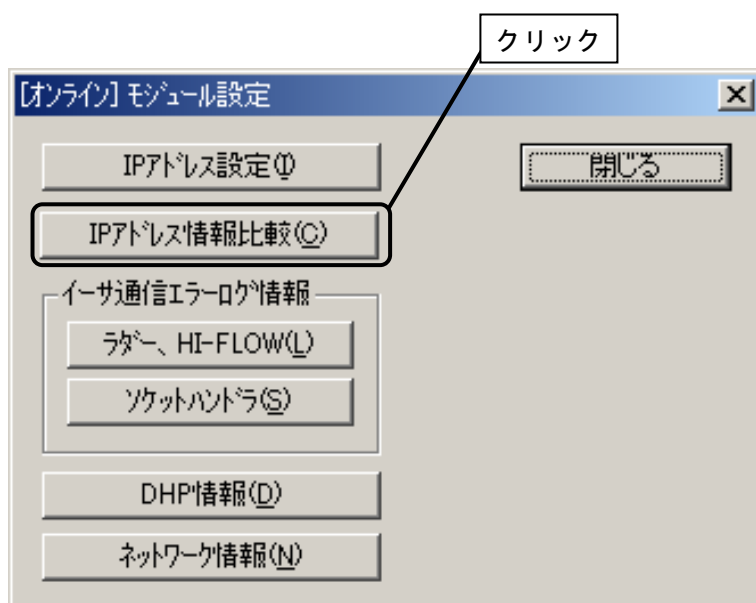


図 4-32 [[オンライン] モジュール設定] 画面

- ③ [ファイルを開く] 画面が表示されます。比較するIPアドレス設定情報ファイルを選択して、**開く** ボタンをクリックしてください。IPアドレス設定情報を比較します。

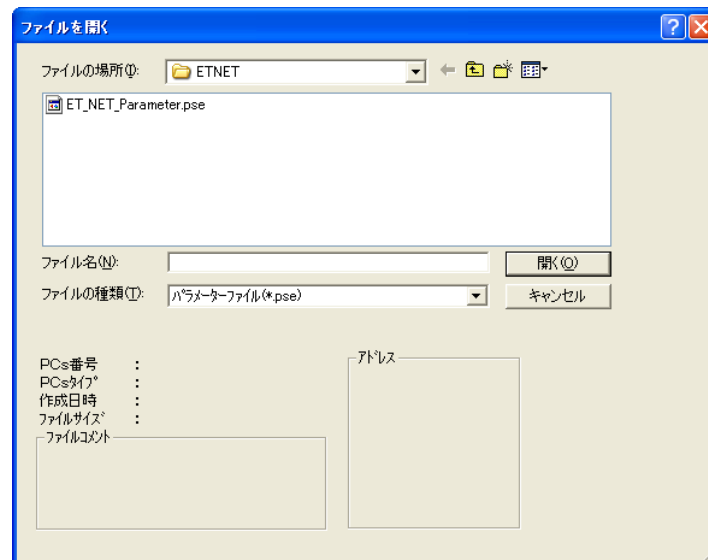


図 4-33 「ファイルを開く」画面

- ④ IPアドレス設定情報 (*) のすべての項目が一致した場合、以下の「不一致はありません。」のメッセージが表示されます。**OK** ボタンをクリックしてください。



図 4-34 「不一致はありません。」メッセージ

(*) IPアドレス設定情報の項目を以下に示します。

- IPアドレス (MAIN/SUB)
- サブネットマスク (MAIN/SUB)
- ブロードキャストアドレス (MAIN/SUB)
- コメント (MAIN/SUB)
- 経路情報 (MAIN/SUB)

4 オペレーション

- ⑤ IPアドレス設定情報 (*) の項目が1つでも不一致となった場合、その不一致項目と内容が表示されます。表示内容を確認してください。



図 4-35 「比較不一致」メッセージ例

(*) IPアドレス設定情報の項目を以下に示します。

- IPアドレス (MAIN/SUB)
- サブネットマスク (MAIN/SUB)
- ブロードキャストアドレス (MAIN/SUB)
- コメント (MAIN/SUB)
- 経路情報 (MAIN/SUB)

5 プログラミング

5. 1 ET.NETのソフトウェア構成

ET.NETのソフトウェアは、ET.NETモジュール内のシステムプログラムと、ユーザの作成するユーザプログラムから構成されます。

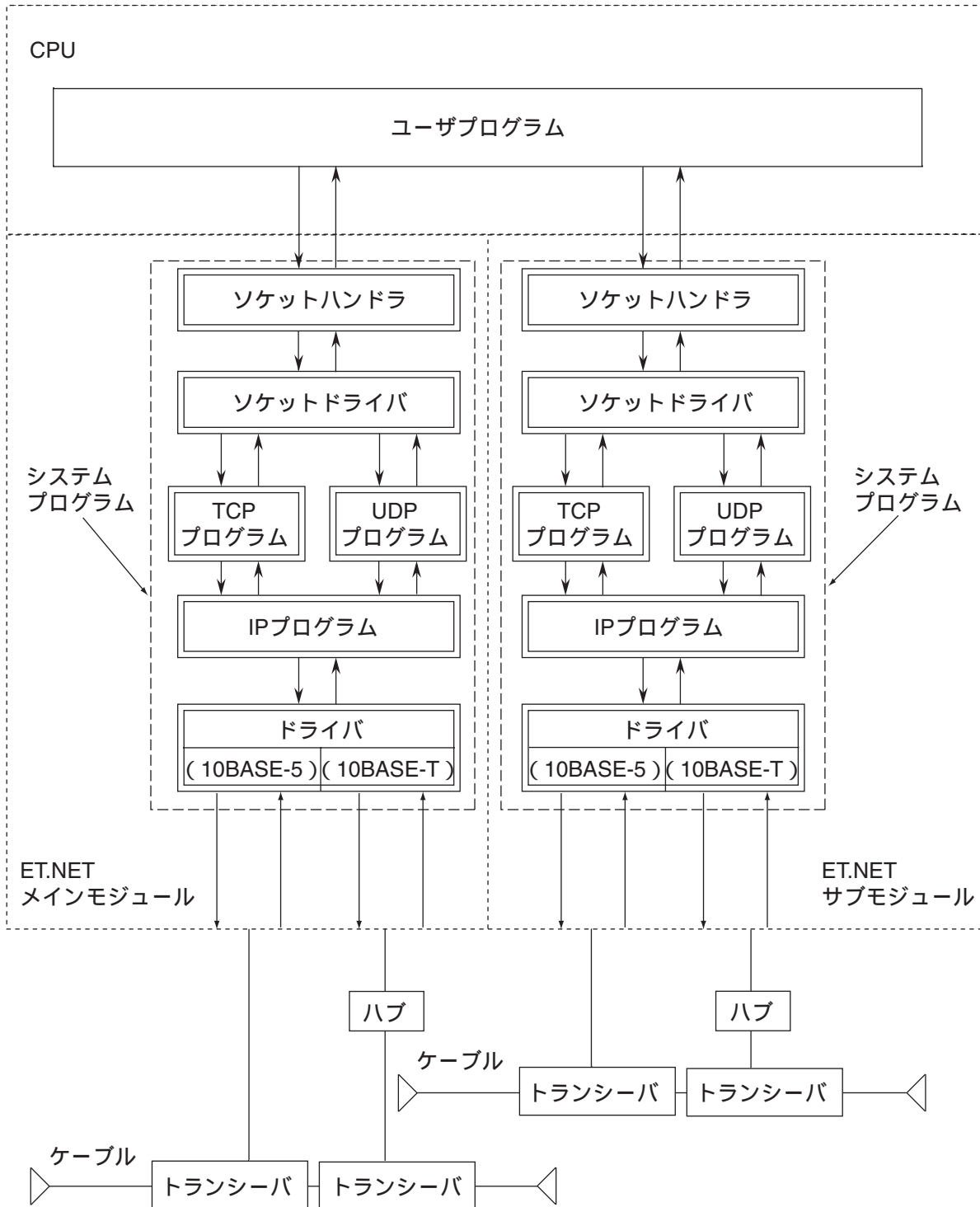


図 5 - 1 ET.NETのソフトウェア構成

5. 2 システムプログラム

システムプログラムは次の6種類に大別できます。

- ソケットハンドラ
- ソケットドライバ
- TCPプログラム
- UDPプログラム
- IPプログラム
- ドライバ

5. 2. 1 ソケットハンドラ

ソケットハンドラは、ユーザプログラムからC言語の関数として呼び出され、ET.NETモジュールを制御します。ユーザはソケットハンドラを利用することにより、ハードウェア仕様および通信プロトコルを意識することなくプログラムできます。

5. 2. 2 ソケットドライバ

ソケットドライバは、ソケットハンドラからのコマンドを、TCPプログラムまたはUDPプログラムにメモリインタフェースで受け渡し、処理します。

5. 2. 3 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します。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

5 プログラミング

5. 2. 4 UDPプログラム

上位のプロトコルとして、高速かつ大容量のデータ送受信管理をします。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
- 同報通信
- パケットに基づいたデータ転送

5. 2. 5 IPプログラム

下位のプロトコルとして、通信回線の論理的な接続をします。IPプログラムの機能を以下に示します。

- パケットの最長に応じたデータの分割と再組み立て
- IPアドレスと物理アドレスの変換

5. 2. 6 ドライバ

通信回線とのデータ送受信をします。ドライバの機能を以下に示します。

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

5. 3 ユーザプログラム

ユーザプログラムは、ソケットハンドラを起動することで、データの送受信をします。

5.4 ソケットハンドラ

ソケットハンドラは、C言語の関数としてユーザプログラムから呼び出され、ET.NETモジュールを制御し、データの送受信をします。ソケットハンドラは、20種類の関数で構成されます。

ソケットハンドラは、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、ソケットハンドラを含めた形では作成（リンク）できません。

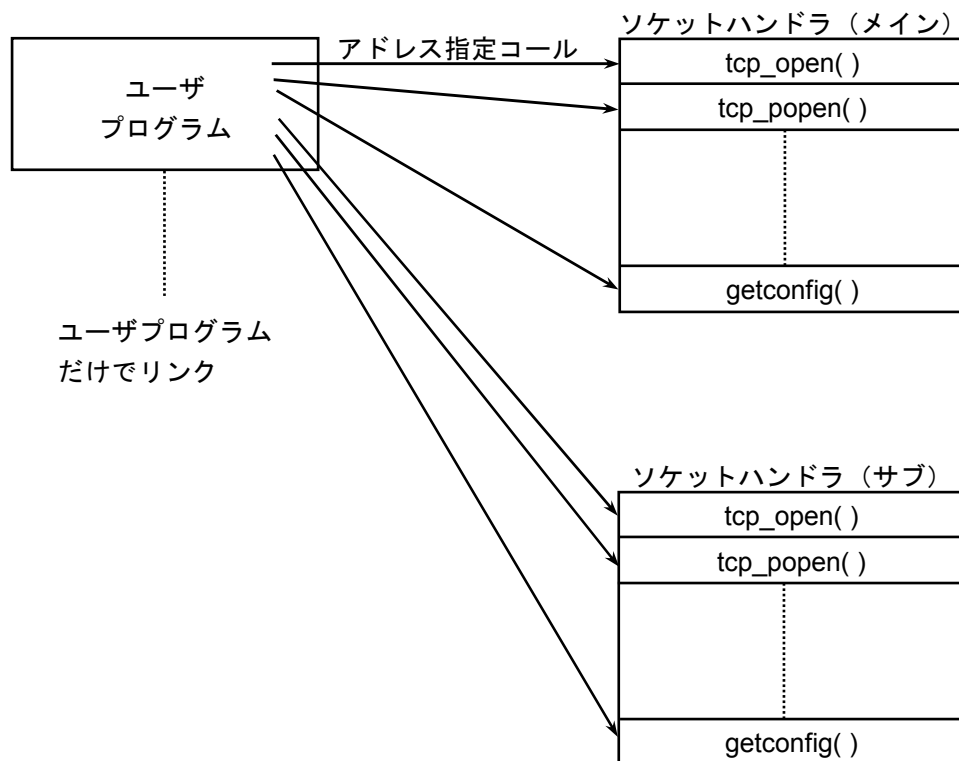


図5-2 ユーザプログラムからのソケットハンドラ呼び出し方法

5 プログラミング

5.4.1 ソケットハンドラー一覧

ソケットハンドラー一覧および各ソケットハンドラの機能を以下に示します。

各ソケットハンドラの呼び出しアドレスは、S10mini, S10Vともに同じアドレスです。

表 5-1 ソケットハンドラー一覧

名 称	呼び出しアドレス (S10mini, S10V共通)		機 能	対応プログラム
	メイン	サブ		
tcp_open()	/874100	/8F4100	TCP能動的オープン	TCP/IP
tcp_popen()	/874106	/8F4106	TCP受動的オープン	TCP/IP
tcp_accept()	/87410C	/8F410C	TCPコネクション要求 受け付け	TCP/IP
tcp_close()	/874112	/8F4112	TCPコネクション終了	TCP/IP
tcp_abort()	/87411E	/8F411E	TCPコネクション強制 終了	TCP/IP
tcp_getaddr()	/874124	/8F4124	TCPソケット情報読み 出し	TCP/IP
tcp_stat()	/87412A	/8F412A	TCPコネクション状態 読み出し	TCP/IP
tcp_send()	/874130	/8F4130	TCPデータ送信	TCP/IP
tcp_receive()	/874136	/8F4136	TCPデータ受信	TCP/IP
udp_open()	/874160	/8F4160	UDPオープン	UDP/IP
udp_close()	/874166	/8F4166	UDPクローズ	UDP/IP
udp_send()	/87416C	/8F416C	UDPデータ送信	UDP/IP
udp_receive()	/874172	/8F4172	UDPデータ受信	UDP/IP
route_list()	/874178	/8F4178	経路情報読み出し	TCP/IPおよびUDP/IP
route_del()	/87417E	/8F417E	経路情報削除	TCP/IPおよびUDP/IP
route_add()	/874184	/8F4184	経路情報登録	TCP/IPおよびUDP/IP
arp_list()	/87418A	/8F418A	ARP情報読み出し	TCP/IPおよびUDP/IP
arp_del()	/874190	/8F4190	ARP情報削除	TCP/IPおよびUDP/IP
arp_add()	/874196	/8F4196	ARP情報登録	TCP/IPおよびUDP/IP
getconfig()	/87419C	/8F419C	コンフィグレーション 情報読み出し	TCP/IPおよびUDP/IP

通 知

- S10Vに実装する場合、以下の制限があります。
 - ・ モジュールRev.B (Ver-Rev : 0005-0001) 以前のET.NETモジュール (LQE520) は、ソケットハンドラを使用してのタスクからの通信機能を使用できません。ツールとの通信用のみ使用できます。S10Vと組み合わせてソケットハンドラを使用される場合は、モジュールRev.F (Ver-Rev : 0006-0000) 以降のモジュールを使用してください。
 - ・ ET.NETモジュールのLQE520とLQE720を同じLPUユニット上で混在させることはできません。
- なお、上記Ver-Revは、S10V基本システムの「Module List」で表示されるET.NETモジュールのマイクロプログラムのVer-Revです。
- 1つのモジュールで、同時に使用できるソケット数は、TCPが12個、UDPが8個までです。
 - 0～9999のポート番号はシステムで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TCPが1～4096バイト、UDPが1～1472バイトです。
 - IPアドレス、サブネットマスクは、CPUまたはLPU内のOSテーブルに設定されます。CPUまたはLPUの交換やOSの再ローディングをした場合は、再設定が必要です。

—タスクの強制終了—

ソケットハンドラを利用しているタスクが強制終了されると、ソケットが登録状態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、`tcp_close()`または`udp_close()`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て、下記のいずれかの処理をしてください。

- 他のタスクまたは組み込みサブルーチンから浮いたソケットを、`tcp_close()`または`udp_close()`する。
- CPUをリセットする。
- 電源を一度遮断し、復電する。

5 プログラミング

tcp_open()

[機能] この関数は、TCP/IPプログラムのソケットの登録、ポートの確保、相手局に対してのコネクションの要求を発行します。リターン値には、登録されたソケットIDまたはエラーコードが返ります。この関数は、SYNを送信し、コネクションの確立（相手局からのSYN受信）を待ちます。相手局からの応答がない場合、75秒後にポート開放エラー（エラーコード：/F0FF）でリターンしますので、tcp_open()を再発行してください。

[リンク手順]

メイン設定	サブ設定
<pre> struct open_p{ long dst_ip; short dst_port; short src_port; char notuse; char ttl; }; } short (*tcp_open)(); short rtn; struct open_p *padr; } tcp_open = (short(*)())0x874100; } rtn = (*tcp_open)(padr); } </pre>	<pre> struct open_p{ long dst_ip; short dst_port; short src_port; char notuse; char ttl; }; } short (*tcp_open)(); short rtn; struct open_p *padr; } tcp_open = (short(*)())0x8F4100; } rtn = (*tcp_open)(padr); } </pre>

[パラメータ]

<入力パラメータ詳細>

padr : 入力パラメータの先頭アドレス (S10Vの場合は、必ず偶数アドレスを指定してください。)

padr -> dst_ip : 相手局のIPアドレス

padr -> dst_port : 相手局のポート番号

padr -> src_port : 自局のポート番号

padr -> notuse : 0固定 (未使用)

padr -> ttl : Time to live (0とした場合、デフォルト値は30となります。)

<出力パラメータ詳細>

リターン値 : 登録されたソケットIDまたはエラーコードが返ります。

(0~/000F) : 登録されたソケットID

(/F000~/FFFF) : エラー発生

エラーコードは、「7. 3. 3 ソケットハンドラ検出のエラーコード」を参照してください。

tcp_popen()

[機能] この関数は、TCP/IPプログラムのソケットを登録し、そのソケットを受動状態にします。リターン値には登録されたソケットまたはエラーコードが返ります。この関数は、UNIXにおけるsocket+bind+listenに相当します。dst_ip、dst_portを0に設定すると、任意の相手局からの接続要求を受け付けることができます。また、src_portを0に設定すると、1024～2047までの任意のポートが確保されます。

[リンク手順]

メイン設定	サブ設定
<pre> struct popen_p{ long dst_ip; short dst_port; short src_port; char listennum; char ttl; }; short (*tcp_popen)(); short rtn; struct popen_p *padr; tcp_popen = (short(*) ())0x874106; rtn = (*tcp_popen)(padr); </pre>	<pre> struct popen_p{ long dst_ip; short dst_port; short src_port; char listennum; char ttl; }; short (*tcp_popen)(); short rtn; struct popen_p *padr; tcp_open = (short(*) ())0x8F4106; rtn = (*tcp_popen)(padr); </pre>

[パラメータ]

<入力パラメータ詳細>

padr : 入力パラメータの先頭アドレス (S10Vの場合は、必ず偶数アドレスを指定してください。)

padr -> dst_ip : 相手局のIPアドレス (相手局未指定の場合は、0とします。)

padr -> dst_port : 相手局のポート番号 (相手局未指定の場合は、0とします。)

padr -> src_port : 自局のポート番号

padr -> listennum : 0固定

padr -> ttl : Time to live (0とした場合、デフォルト値は30となります。)

<出力パラメータ詳細>

リターン値 : 登録されたソケットIDまたはエラーコードが返ります。

(0~/000F) : 登録されたソケットID

(/F000~/FFFF) : エラー発生

エラーコードは、「7. 3. 3 ソケットハンドラ検出のエラーコード」を参照してください。

5 プログラミング

tcp_accept()

[機能] この関数は、tcp_popen()により受動状態となったソケットIDに対する接続の要求（SYNの受信）を待ち、接続の確立を受け付けます。リターン値には接続確立後の登録されたソケットIDまたはエラーコードが返ります。入力パラメータのソケットIDと接続確立後の登録されたソケットIDは同じ値となります。この関数は相手局と接続されるまで待ち続けます。

[リンク手順]

メイン設定	サブ設定
<pre> struct accept_p{ short s_id; }; } short (*tcp_accept)(); short rtn; struct accept_p *padr; } tcp_accept = (short(*) ())0x87410C; } rtn = (*tcp_accept)(padr); } </pre>	<pre> struct accept_p{ short s_id; }; } short (*tcp_accept)(); short rtn; struct accept_p *padr; } tcp_accept = (short(*) ())0x8F410C; } rtn = (*tcp_accept)(padr); } </pre>

[パラメータ]

<入力パラメータ詳細>

padr : 入力パラメータの先頭アドレス (S10Vの場合は、必ず偶数アドレスを指定してください。)

padr -> s_id : ソケットID

<出力パラメータ詳細>

リターン値 : 登録されたソケットIDまたはエラーコードが返ります。

(0~/000F) : 登録されたソケットID

(/F000~/FFFF) : エラー発生

エラーコードは、「7. 3. 3 ソケットハンドラ検出のエラーコード」を参照してください。

tcp_close()

[機能] この関数は、ソケットIDに対応したコネクションを終了させ、ソケットを削除します。リターン値には処理結果が返ります。この関数は、FINを送信し、コネクションの終了（相手局からのFIN受信）を待ちます。相手局からの応答がない場合、30秒後にソケットドライバタイムアウトエラー（エラーコード：/F012）でリターンしますので、tcp_abort()を発行してください。

[リンク手順]

メイン設定	サブ設定
<pre> struct close_p{ short s_id; }; } short (*tcp_close)(); short rtn; struct close_p *padr; } tcp_close = (short(*) ())0x874112; } rtn = (*tcp_close)(padr); } </pre>	<pre> struct close_p{ short s_id; }; } short (*tcp_close)(); short rtn; struct close_p *padr; } tcp_close = (short(*) ())0x8F4112; } rtn = (*tcp_close)(padr); } </pre>

[パラメータ]

<入力パラメータ詳細>

padr : 入力パラメータの先頭アドレス（S10Vの場合は、必ず偶数アドレスを指定してください。）

padr -> s_id : ソケットID

<出力パラメータ詳細>

リターン値 : 処理結果が返ります。

(0) : 正常終了

(/F000~/FFFF) : エラー発生

エラーコードは、「7. 3. 3 ソケットハンドラ検出のエラーコード」を参照してください。

5 プログラミング

tcp_abort()

[機能] この関数は、ソケットIDに対応したコネクションを強制終了（RSTを送信）させ、ソケットを削除します。リターン値には処理結果が返ります。

[リンク手順]

メイン設定	サブ設定
<pre> struct abort_p{ short s_id; }; { short (*tcp_abort)(); short rtn; struct abort_p *padr; { tcp_abort = (short(*) ())0x87411E; { rtn = (*tcp_abort)(padr); { </pre>	<pre> struct abort_p{ short s_id; }; { short (*tcp_abort)(); short rtn; struct abort_p *padr; { tcp_abort = (short(*) ())0x8F411E; { rtn = (*tcp_abort)(padr); { </pre>

[パラメータ]

<入力パラメータ詳細>

padr : 入力パラメータの先頭アドレス（S10Vの場合は、必ず偶数アドレスを指定してください。）

padr -> s_id : ソケットID

<出力パラメータ詳細>

リターン値 : 処理結果が返ります。

(0) : 正常終了

(/F000~/FFFF) : エラー発生

エラーコードは、「7. 3. 3 ソケットハンドラ検出のエラーコード」を参照してください。

tcp_getaddr()

[機能] この関数は、ソケットIDに対応したコネクション相手局のIPアドレス、自局ポート番号、相手局ポート番号を取得します。リターン値には処理結果が返ります。処理結果が正常終了の場合、outinfの取得情報が有効となります。

[リンク手順]

メイン設定	サブ設定
<pre> struct sid_p{ short s_id; } struct getaddr_p{ long ipaddr; short src_port; short dst_port; }; short (*tcp_getaddr)(); short rtn; struct sid_p *padr; struct getaddr_p *outinf; tcp_getaddr = (short(*) ())0x874124; rtn = (*tcp_getaddr)(padr, outinf); </pre>	<pre> struct sid_p{ short s_id; } struct getaddr_p{ long ipaddr; short src_port; short dst_port; }; short (*tcp_getaddr)(); short rtn; struct sid_p *padr; struct getaddr_p *outinf; tcp_getaddr = (short(*) ())0x8F4124; rtn = (*tcp_getaddr)(padr, outinf); </pre>

[パラメータ]

<入力パラメータ詳細>

padr : 入力パラメータの先頭アドレス (S10Vの場合は、必ず偶数アドレスを指定してください。)

padr -> s_id : ソケットID

<出力パラメータ詳細>

outinf : 出力パラメータの先頭アドレス (S10Vの場合は、必ず偶数アドレスを指定してください。)

outinf -> ipaddr : 相手局のIPアドレス

outinf -> src_port : 自局のポート番号

outinf -> dst_port : 相手局のポート番号

リターン値 : 処理結果が返ります。

(0) : 正常終了

(/F000~/FFFF) : エラー発生

エラーコードは、「7. 3. 3 ソケットハンドラ検出のエラーコード」を参照してください。

5 プログラミング

tcp_stat()

[機能] この関数は、ソケットIDに対応したコネクションのステータスを取得します。リターン値には、処理結果が返ります。処理結果が正常終了の場合、outinfの取得情報が有効となります。

[リンク手順]

メイン設定	サブ設定
<pre> struct sid_p{ short s_id; } struct stat_p{ unsigned short stat; unsigned short urg; unsigned short sendwin; unsigned short rcvwin; }; { short (*tcp_stat)(); short rtn; struct sid_p *padr; struct stat_p *outinf; { tcp_stat = (short(*)())0x87412A; } rtn = (*tcp_stat)(padr, outinf); } </pre>	<pre> struct sid_p{ short s_id; } struct stat_p{ unsigned short stat; unsigned short urg; unsigned short sendwin; unsigned short rcvwin; }; { short (*tcp_stat)(); short rtn; struct sid_p *padr; struct stat_p *outinf; { tcp_stat = (short(*)())0x8F412A; } rtn = (*tcp_stat)(padr, outinf); } </pre>

[パラメータ]

<入力パラメータ詳細>

padr : 入力パラメータの先頭アドレス (S10Vの場合は、必ず偶数アドレスを指定してください。)

padr -> s_id : ソケットID

<出力パラメータ詳細>

outinf : 出力パラメータの先頭アドレス (S10Vの場合は、必ず偶数アドレスを指定してください。)

outinf -> stat : コネクション状態

- 0 : CLOSED
- 1 : LISTEN
- 2 : SYN_SENT
- 3 : SYN_RECEIVED
- 4 : ESTABLISHED
- 5 : CLOSE_WAIT
- 6 : FIN_WAIT_1
- 7 : CLOSING
- 8 : LAST_ACK
- 9 : FIN_WAIT_2
- 10 : TIME_WAIT

outinf -> urg : urgent dataの有無

- 0 : urgent dataなし
- 0以外 : urgent data数

outinf -> sendwin : 送信ウィンドウの残量

outinf -> rcvwin : 到着済み受信データ量

リターン値 : 処理結果が返ります。

(0) : 正常終了

(/F000~/FFFF) : エラー発生

エラーコードは、「7. 3. 3 ソケットハンドラ検出のエラーコード」を参照してください。

5 プログラミング

tcp_send()

[機能] この関数は、ソケットIDに対応したコネクシオンにパラメータのbufからlen分のデータを送信します。リターン値には処理結果が返ります。リターン値が/F012の場合は、tcp_stat()でコネクシオン状態および送信ウィンドウ残量により送信リトライ中を確認してください。この関数は、送信ウィンドウにデータが格納された時点でリターンします。データの送信状態は、tcp_stat()の送信ウィンドウ残量により確認してください。

[リンク手順]

メイン設定	サブ設定
<pre> struct send_p{ short s_id; short len; char *buf; }; { short (*tcp_send)(); short rtn; struct send_p *padr; { tcp_send = (short(*) ())0x874130; } rtn = (*tcp_send)(padr); { </pre>	<pre> struct send_p{ short s_id; short len; char *buf; }; { short (*tcp_send)(); short rtn; struct send_p *padr; { tcp_send = (short(*) ())0x8F4130; } rtn = (*tcp_send)(padr); { </pre>

[パラメータ]

<入力パラメータ詳細>

padr : 入力パラメータの先頭アドレス (S10Vの場合は、必ず偶数アドレスを指定してください。)

padr -> s_id : ソケットID

padr -> len : 送信データ長 (1~4096バイト)

padr -> buf : 送信データの先頭アドレス

(S10Vの場合は、必ず偶数アドレスを指定してください。)

<出力パラメータ詳細>

リターン値 : 処理結果が返ります。

(0) : 正常終了

(/F000~/FFFF) : エラー発生

エラーコードは、「7. 3. 3 ソケットハンドラ検出のエラーコード」を参照してください。

tcp_receive()

[機能] この関数は、ソケットIDに対応したコネクションからパラメータのlen分のデータをbufに受信します。リターン値には処理結果が返ります。この関数は、パラメータのtimに受信待ち時間を指定できますが、受信待ち時間以内であってもデータを受信した時点でリターンします。

[リンク手順]

メイン設定	サブ設定
<pre> struct receive_p{ short s_id; short len; char *buf; long tim; }; { short (*tcp_receive)(); short rtn; struct receive_p *padr; { tcp_receive = (short(*) ())0x874136; { rtn = (*tcp_receive)(padr); { </pre>	<pre> struct receive_p{ short s_id; short len; char *buf; long tim; }; { short (*tcp_receive)(); short rtn; struct receive_p *padr; { tcp_receive = (short(*) ())0x8F4136; { rtn = (*tcp_receive)(padr); { </pre>

[パラメータ]

<入力パラメータ詳細>

padr : 入力パラメータの先頭アドレス (S10Vの場合は、必ず偶数アドレスを指定してください。)

padr -> s_id : ソケットID

padr -> len : 受信データ長 (1~4096バイト)

padr -> buf : 受信データの先頭アドレス

(S10Vの場合は、必ず偶数アドレスを指定してください。)

padr -> tim : 受信待ち時間 (ms : 0~86,400,000 (24時間))

<出力パラメータ詳細>

リターン値 : 処理結果が返ります。

(0) : 正常終了 (受信データなし)

(/0001~/1000) : 正常終了 (受信したバイト数)

(/F000~/FFFF) : エラー発生

エラーコードは、「7. 3. 3 ソケットハンドラ検出のエラーコード」を参照してください。

5 プログラミング

udp_open()

[機能] この関数は、UDP/IPプログラムのソケットの登録、ポートの確保をします。リターン値には登録されたソケットIDまたはエラーコードを返します。

パラメータのdst_ipに0を指定すると、任意のホストからパケットを受信できます。

パラメータのdst_portに0を指定すると、任意のポートからデータを受信できます。

パラメータのsrc_portに0を指定すると、1024～2048までの使用していないポートが確保されます。

[リンク手順]

メイン設定	サブ設定
<pre> struct uopen_p{ long dst_ip; short dst_port; short src_port; char pktmode; char ttl; }; { short (*udp_open)(); short rtn; struct uopen_p *padr; { udp_open = (short(*) ())0x874160; { rtn = (*udp_open)(padr); { </pre>	<pre> struct uopen_p{ long dst_ip; short dst_port; short src_port; char pktmode; char ttl; }; { short (*udp_open)(); short rtn; struct uopen_p *padr; { udp_open = (short(*) ())0x8F4160; { rtn = (*udp_open)(padr); { </pre>

[パラメータ]

<入力パラメータ詳細>

padr : 入力パラメータの先頭アドレス (S10Vの場合は、必ず偶数アドレスを指定してください。)

padr -> dst_ip : 相手局のIPアドレス

padr -> dst_port : 相手局のポート番号

padr -> src_port : 自局のポート番号

padr -> pktmode : パケットモード (0固定)

padr -> ttl : Time to live (0とした場合、デフォルト値は30となります。)

<出力パラメータ詳細>

リターン値 : 登録されたソケットIDまたはエラーコードが返ります。

(/0020~/0027) : 登録されたソケットID

(/F000~/FFFF) : エラー発生

エラーコードは、「7. 3. 3 ソケットハンドラ検出のエラーコード」を参照してください。

udp_close()

[機能] この関数は、ソケットIDに対応したソケットを削除する関数です。リターン値には処理結果が返ります。

[リンク手順]

メイン設定	サブ設定
<pre> struct uclose_p{ short s_id; }; { short (*udp_close)(); short rtn; struct uclose_p *padr; { udp_close = (short(*) ())0x874166; { rtn = (*udp_close)(padr); { </pre>	<pre> struct uclose_p{ short s_id; }; { short (*udp_close)(); short rtn; struct uclose_p *padr; { udp_close = (short(*) ())0x8F4166; { rtn = (*udp_close)(padr); { </pre>

[パラメータ]

<入力パラメータ詳細>

padr : 入力パラメータの先頭アドレス (S10Vの場合は、必ず偶数アドレスを指定してください。)

padr -> s_id : ソケットID

<出力パラメータ詳細>

リターン値 : 処理結果が返ります。

(0) : 正常終了

(/F000~/FFFF) : エラー発生

エラーコードは、「7. 3. 3 ソケットハンドラ検出のエラーコード」を参照してください。

5 プログラミング

udp_send()

[機能] この関数は、ソケットIDに対応したソケットにパラメータのbufからlen分のデータを送信します。リターン値には処理結果が返ります。dst_ip、dst_portの指定は、udp_open()で指定されたものが優先されます。

[リンク手順]

メイン設定	サブ設定
<pre> struct usend_p{ short s_id; short notuse; long dst_ip; short dst_port; short len; char *buf; }; short (*udp_send)(); short rtn; struct usend_p *padr; { udp_send = (short(*)())0x87416C; } rtn = (*udp_send)(padr); { </pre>	<pre> struct usend_p{ short s_id; short notuse; long dst_ip; short dst_port; short len; char *buf; }; short (*udp_send)(); short rtn; struct usend_p *padr; { udp_send = (short(*)())0x8F416C; } rtn = (*udp_send)(padr); { </pre>

[パラメータ]

<入力パラメータ詳細>

padr : 入力パラメータの先頭アドレス (S10Vの場合は、必ず偶数アドレスを指定してください。)

padr -> s_id : ソケットID

padr -> notuse : 0固定 (未使用)

padr -> dst_ip : 相手局のIPアドレス

padr -> dst_port : 相手局のポート番号

padr -> len : 送信するデータ長 (バイト数: 1~1472)

padr -> buf : 送信するデータの先頭アドレス

(S10Vの場合は、必ず偶数アドレスを指定してください。)

udp_open()で0以外を指定した場合、udp_open()のdst_ip、dst_portを使用します。

<出力パラメータ詳細>

リターン値: 処理結果が返ります。

(0) : 正常終了

(/F000~/FFFF) : エラー発生

エラーコードは、「7. 3. 3 ソケットハンドラ検出のエラーコード」を参照してください。

■ dst_ip、dst_portの指定について

- `udp_open()`で0以外を指定した場合、`udp_open()`で指定したパラメータを使用します。
- `udp_open()`で0を指定した場合、`udp_send()`で指定したパラメータを使用します。
- `udp_open()`で0を指定し、`udp_send()`でも0を指定した場合、アドレス不正エラー（エラーコード：`/FFF0`）でリターンします。

5 プログラミング

udp_receive()

[機能] この関数は、ソケットIDに対応したソケットからパラメータのbufにデータを受信します。リターン値には処理結果が返ります。この関数は、パラメータのtimに受信待ち時間を指定できます。ただし、受信待ち時間以内であっても、データを受信した時点でリターンします。

[リンク手順]

メイン設定	サブ設定
<pre> struct ureceive_p{ short s_id; short notuse; char *buf; long tim; }; short (*udp_receive)(); short rtn; struct ureceive_p *padr; udp_receive = (short(*)())0x874172; rtn = (*udp_receive)(padr); </pre>	<pre> struct ureceive_p{ short s_id; short notuse; char *buf; long tim; }; short (*udp_receive)(); short rtn; struct ureceive_p *padr; udp_receive = (short(*)())0x8F4172; rtn = (*udp_receive)(padr); </pre>

[パラメータ]

<入力パラメータ詳細>

padr : 入力パラメータの先頭アドレス (S10Vの場合は、必ず偶数アドレスを指定してください。)

padr -> s_id : ソケットID

padr -> notuse : 0固定 (未使用)

padr -> buf : 受信バッファの先頭アドレス

(S10Vの場合は、必ず偶数アドレスを指定してください。)

padr -> tim : 受信待ち時間 (ms : 0~86,400,000 (24時間))

<出力パラメータ詳細>

リターン値 : 処理結果が返ります。

(0) : 正常終了 (受信データなし)

(/0001~/05C0) : 正常終了 (受信したバイト数)

(/F000~/FFFF) : エラー発生

エラーコードは、「7. 3. 3 ソケットハンドラ検出のエラーコード」を参照してください。

通 知

udp_receive()は、パケットごとの受信をします。したがって、バッファエリアを1,472バイト確保してください。

route_list ()

[機能] この関数は、経路情報（経路情報テーブルサイズは最大16）を取得します。リターン値には取得したエントリ数が返ります。パラメータのlenに0を指定すると、取得エントリ数のみ返します。lenは16バイトの倍数を指定してください。

[リンク手順]

メイン設定	サブ設定
<pre> struct lstrt_p{ short len; short notuse; void *buf; }; short (*route_list)(); short rtn; struct lstrt_p *padr; route_list = (short(*))0x874178; rtn = (*route_list)(padr); </pre>	<pre> struct lstrt_p{ short len; short notuse; void *buf; }; short (*route_list)(); short rtn; struct lstrt_p *padr; route_list = (short(*))0x8F4178; rtn = (*route_list)(padr); </pre>

[パラメータ]

<入力パラメータ詳細>

padr : 入力パラメータの先頭アドレス (S10Vの場合は、必ず偶数アドレスを指定してください。)

padr -> len : データ長 (バイト数: 16の倍数)

padr -> notuse : 0固定 (未使用)

padr -> buf : データの先頭アドレス (S10Vの場合は、必ず偶数アドレスを指定してください。)

<出力パラメータ詳細>

リターン値: 取得したエントリ数が返ります。

(0) : エントリなし

(/0001~/0010) : 取得エントリ数

取得データ構造 (bufの内容)

```

typedef struct{
  unsigned long  dstaddr   : 相手局のIPアドレス
  unsigned long  gtwayaddr : ゲートウェイのIPアドレス
  unsigned short metric    : メトリック (ゲートウェイの経由数)
  unsigned short rt_types  : タイプ
  unsigned short refcnt    : 参照カウンタ
  unsigned short notuse    : (未使用)
}routeentry

```

5 プログラミング

route_del()

[機能] この関数は、経路情報テーブルから経路情報を削除します。リターン値には処理結果が返ります。

[リンク手順]

メイン設定	サブ設定
<pre> struct delrt_p{ long dstaddr; long gtwayaddr; }; { short (*route_del)(); short rtn; struct delrt_p *padr; { route_del = (short(*))0x87417E; { rtn = (*route_del)(padr); { </pre>	<pre> struct delrt_p{ long dstaddr; long gtwayaddr; }; { short (*route_del)(); short rtn; struct delrt_p *padr; { route_del = (short(*))0x8F417E; { rtn = (*route_del)(padr); { </pre>

[パラメータ]

<入力パラメータ詳細>

padr : 入力パラメータの先頭アドレス (S10Vの場合は、必ず偶数アドレスを指定してください。)

padr -> dstaddr : 相手局のIPアドレス

padr -> gtwayaddr : ゲートウェイIPアドレス

<出力パラメータ詳細>

リターン値 : 処理結果が返ります。

(0) : 正常終了

(/F000~/FFFF) : エラー発生

エラーコードは、「7. 3. 3 ソケットハンドラ検出のエラーコード」を参照してください。

route_add()

[機能] この関数は、経路情報テーブルに経路情報を登録します。リターン値には処理結果が返ります。

[リンク手順]

メイン設定	サブ設定
<pre> struct addrt_p{ long dstaddr; long gatewayaddr; short metric; }; { short (*route_add)(); short rtn; struct addrt_p *padr; { route_add = (short(*) ())0x874184; { rtn = (*route_add)(padr); { </pre>	<pre> struct addrt_p{ long dstaddr; long gatewayaddr; short metric; }; { short (*route_add)(); short rtn; struct addrt_p *padr; { route_add = (short(*) ())0x8F4184; { rtn = (*route_add)(padr); { </pre>

[パラメータ]

<入力パラメータ詳細>

padr : 入力パラメータの先頭アドレス (S10Vの場合は、必ず偶数アドレスを指定してください。)

padr -> dstaddr : 相手局のIPアドレス

padr -> gatewayaddr : ゲートウェイIPアドレス

padr -> metric : メトリック (ゲートウェイの経由数)

<出力パラメータ詳細>

リターン値 : 処理結果が返ります。

(0) : 正常終了

(F000~/FFFF) : エラー発生

エラーコードは、「7. 3. 3 ソケットハンドラ検出のエラーコード」を参照してください。

5 プログラミング

arp_list()

[機能] この関数は、ARP情報（ARP情報テーブルサイズは最大32）を取得します。リターン値には取得したエントリ数が返ります。パラメータのlenに0を指定すると、取得エントリ数のみ返します。lenは12バイトの倍数を指定してください。

[リンク手順]

メイン設定	サブ設定
<pre> struct lstartp_p{ short len; short notuse; void *buf; }; { short (*arp_list)(); short rtn; struct lstartp_p *padr; { arp_list = (short(*)())0x87418A; { rtn = (*route_arp)(padr); { </pre>	<pre> struct lstartp_p{ short len; short notuse; void *buf; }; { short (*arp_list)(); short rtn; struct lstartp_p *padr; { arp_list = (short(*)())0x8F418A; { rtn = (*route_arp)(padr); { </pre>

[パラメータ]

<入力パラメータ詳細>

padr : 入力パラメータの先頭アドレス（S10Vの場合は、必ず偶数アドレスを指定してください。）

padr -> len : データ長（バイト数：12の倍数）

padr -> notuse : 0固定（未使用）

padr -> buf : データの先頭アドレス（S10Vの場合は、必ず偶数アドレスを指定してください。）

<出力パラメータ詳細>

リターン値：取得したエントリ数が返ります。

(0) : エントリなし

(/0001~/0020) : 取得エントリ数

取得データ構造（bufの内容）

```

typedef struct{
  unsigned long  dstaddr   : 相手局のIPアドレス
  unsigned char  et_addr[6] : 相手局の物理アドレス
  unsigned char  ar_timer  : タイマ
  unsigned char  ar_flags  : フラグ
}arpt-t

```

arp_del()

[機能] この関数は、ARP情報テーブルからARP情報を削除します。リターン値には処理結果が返ります。

[リンク手順]

メイン設定	サブ設定
<pre> struct delarp{ unsigned long ipaddr; unsigned char etaddr[6]; }; } short (*arp_del)(); short rtn; struct delarp_p *padr; } arp_del = (short(*) ())0x874190; } rtn = (*arp_del)(padr); } </pre>	<pre> struct delarp{ unsigned long ipaddr; unsigned char etaddr[6]; }; } short (*arp_del)(); short rtn; struct delarp_p *padr; } arp_del = (short(*) ())0x8F4190; } rtn = (*arp_del)(padr); } </pre>

[パラメータ]

<入力パラメータ詳細>

padr : 入力パラメータの先頭アドレス (S10Vの場合は、必ず偶数アドレスを指定してください。)

padr -> ipaddr : 相手局のIPアドレス

padr -> etaddr[6] : 相手局の物理アドレス

<出力パラメータ詳細>

リターン値 : 処理結果が返ります。

(0) : 正常終了

(/F000~/FFFF) : エラー発生

エラーコードは、「7. 3. 3 ソケットハンドラ検出のエラーコード」を参照してください。

5 プログラミング

arp_add()

[機能] この関数は、ARP情報テーブルにARP情報を登録します。リターン値には処理結果が返ります。

[リンク手順]

メイン設定	サブ設定
<pre>struct addarp_p{ long ipaddr; char etaddr[6]; short flag; }; short (*arp_add)(); short rtn; struct addarp_p *padr; { arp_add = (short(*) ())0x874196; { rtn = (*arp_add)(padr); {</pre>	<pre>struct addarp_p{ long ipaddr; char etaddr[6]; short flag; }; short (*arp_add)(); short rtn; struct addarp_p *padr; { arp_add = (short(*) ())0x8F4196; { rtn = (*arp_add)(padr); {</pre>

[パラメータ]

<入力パラメータ詳細>

padr : 入力パラメータの先頭アドレス (S10Vの場合は、必ず偶数アドレスを指定してください。)

padr -> ipaddr : 相手局のIPアドレス

padr -> etaddr[6] : 相手局の物理アドレス

padr -> flag : フラグ (0固定)

<出力パラメータ詳細>

リターン値 : 処理結果が返ります。

(0) : 正常終了

(/F000~/FFFF) : エラー発生

エラーコードは、「7. 3. 3 ソケットハンドラ検出のエラーコード」を参照してください。

getconfig()

[機能] この関数は、コンフィグレーションブロックを取得します。リターン値には処理結果が返ります。

[リンク手順]

メイン設定	サブ設定
<pre> struct config_p{ void *config_ptr; }; { short (*getconfig)(); short rtn; struct config_p *padr; { getconfig = (short(*) ())0x87419C; } rtn = (*getconfig)(padr); } </pre>	<pre> struct config_p{ void *config_ptr; }; { short (*getconfig)(); short rtn; struct config_p *padr; { getconfig = (short(*) ())0x8F419C; } rtn = (*getconfig)(padr); } </pre>

[パラメータ]

<入力パラメータ詳細>

padr : 入力パラメータの先頭アドレス (S10Vの場合は、必ず偶数アドレスを指定してください。)

padr -> config_ptr : コンフィグレーションブロックの先頭アドレス

<出力パラメータ詳細>

リターン値 : 処理結果が返ります。

(0) : 正常終了

<コンフィグレーションブロックの詳細>

コンフィグレーションブロックは、下記のデータ構造になります。

```

struct config_ptr{
    long    ip_addr   : 自局のIPアドレス (ネットワークオーダ) (任意)
    long    netmask   : サブネットマスク (任意)
    long    broadcast : ブロードキャストアドレス (任意)
    char    tcp_num    : 最大TCPソケット数 (16)
    char    udp_num    : 最大UDPソケット数 (8)
    char    rt_num     : 経路情報テーブルサイズ (16)
    char    arp_num    : ARP情報テーブルサイズ (32)
    short   tcp_win    : TCPの送受信ウィンドウサイズ (1024)
};

```

5.5 ソケットハンドラ発行手順例

5.5.1 TCP/IPプログラム使用例

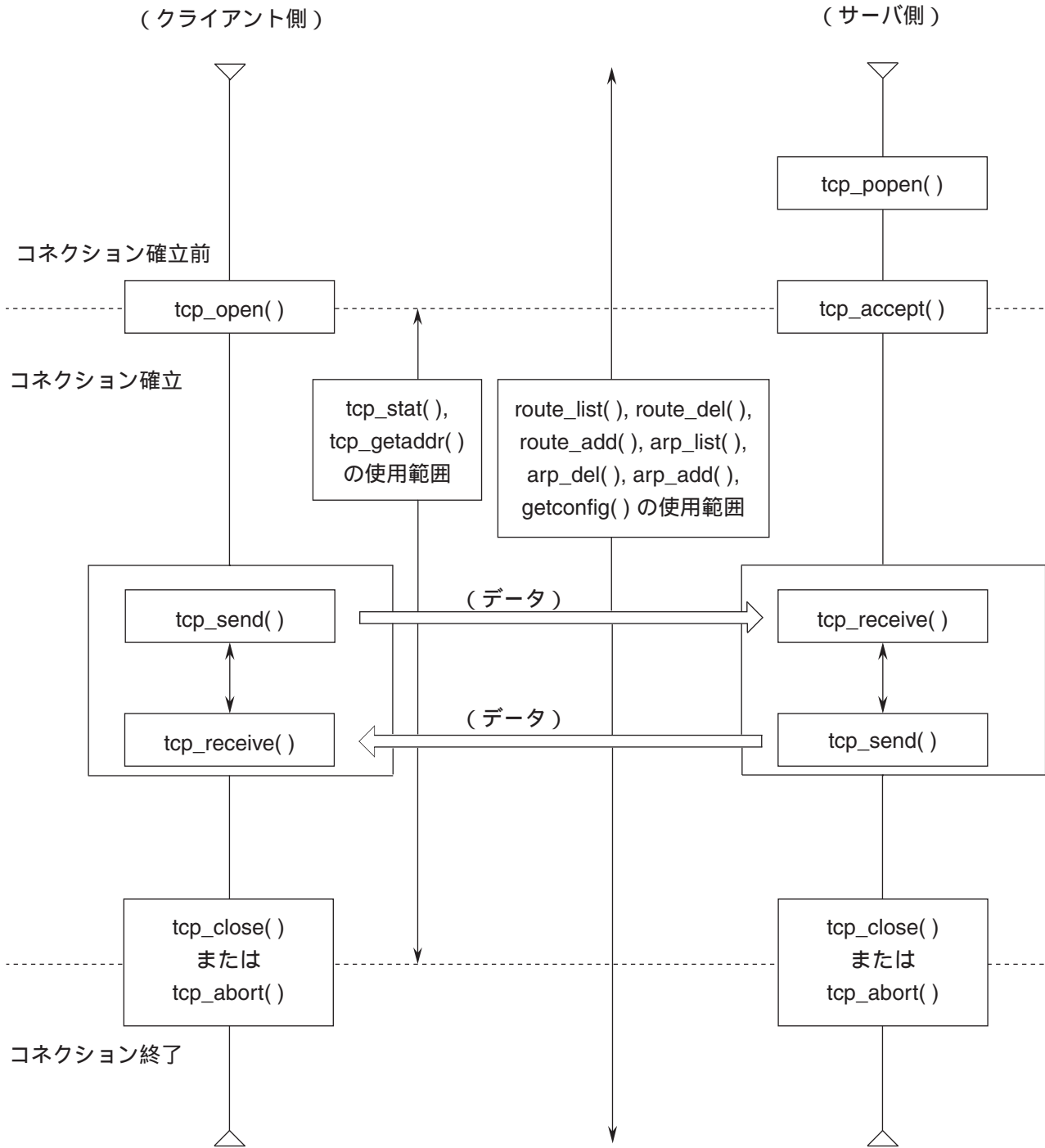


図5-3 TCP/IPプログラム使用時のソケットハンドラ発行手順例

5. 5. 2 UDP/IPプログラム使用例

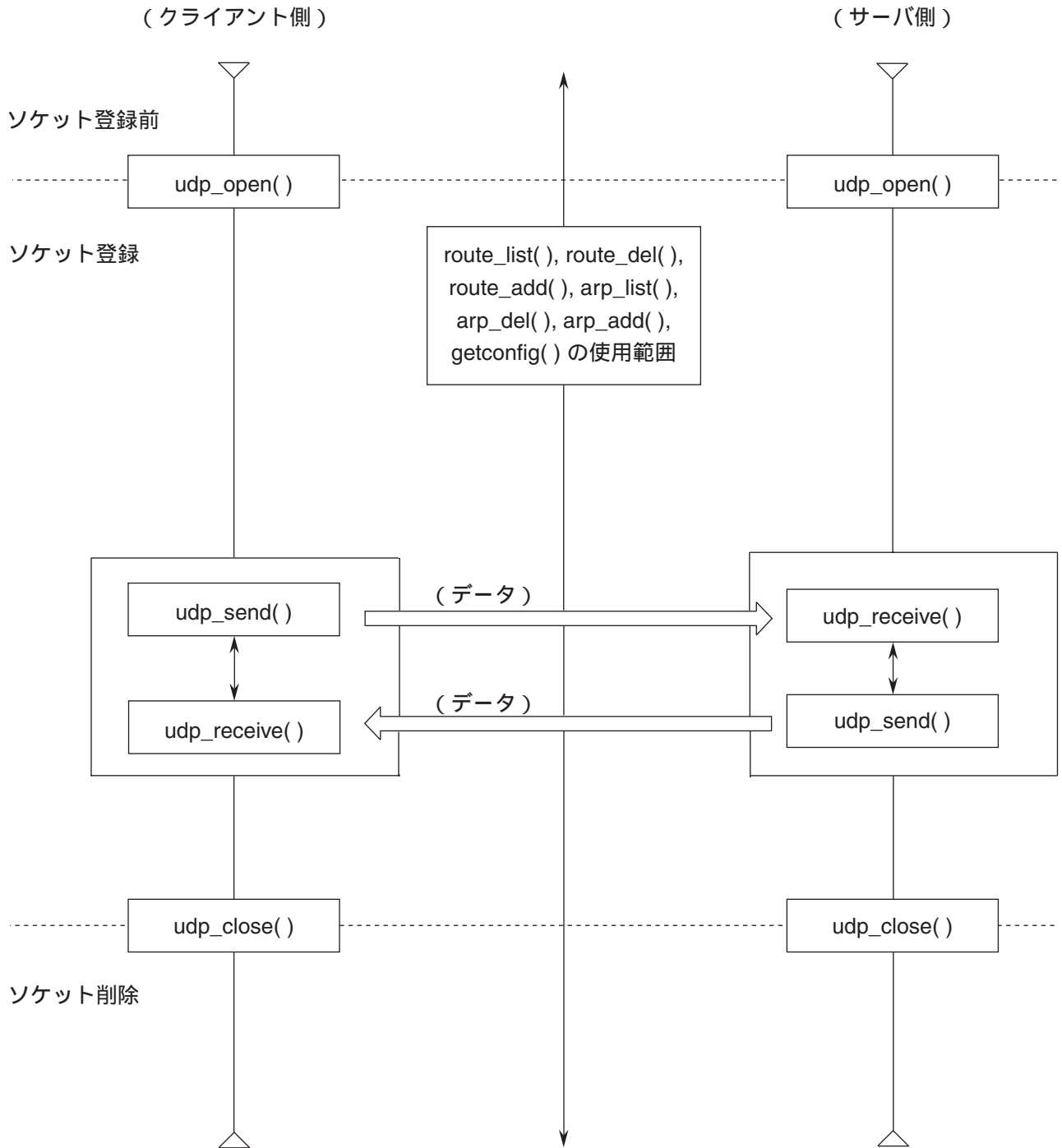
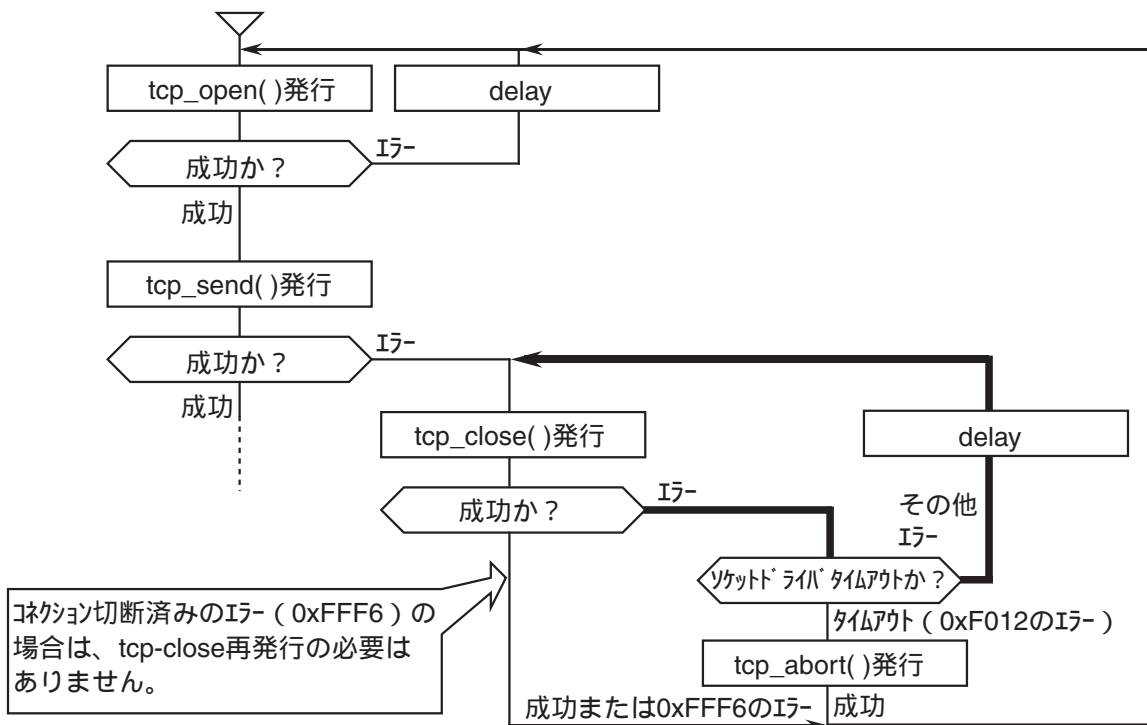


図 5-4 UDP/IPプログラム使用時のソケットハンドラ発行手順例

・ tcp_close()のエラー処理

ソケットハンドラがエラー発生を示すリターン値を返してきたためにtcp_close()を発行する場合、tcp_close()のリターンコードもチェックし、エラーの場合には「ソケットハンドラ検出のエラーコード表」に従って再発行してください。エラーのままにしてtcp_close()を再発行しないと、再コネクションできなかつたり、浮いたソケットが発生する可能性があります。下図にソケットハンドラの発行例を示します。



(注) 上記はudp_closeのエラー処理のときも同じです。

・ 同じソケットに対する非同期アクセスの禁止

1つのソケットに対し、非同期に複数のソケットライブラリ関数を発行すると、関数の実行結果がエラーとなる場合があります。複数のタスクで同じソケットに対してソケットライブラリ関数を発行する場合にこの現象が発生しやすいので、1つのソケットに対して1つのタスクで処理してください。

・送信タイムアウト検出時間

ソケットライブラリ関数を発行し、通信異常や相手装置のダウンなどによりACKパケットのタイムアウトが発生した場合、タイムアウト検出時間は下表となります。したがって、ソケットハンドラのタイムアウトを検出し、再発行または再コネクションしても、下表の時間がかかります。システム設計時には、必ず通信エラーが発生することを前提として、下表のタイムアウト時間に問題がないか確認してください。

項 目		検出時間	内 容
tcp_open()タイムアウト検出時間 (SYNのリトライ間隔)		75秒	相手装置からの応答がない場合、下記間隔でSYNのリトライをします。 6秒, 12秒, 24秒, 33秒
tcp_send()タイムアウト検出時間 (SENDのリトライ間隔)		30秒	相手装置からの応答がない場合、下記間隔で送信リトライをします。 1秒, 2秒, 4秒, 8秒, 16秒 ただし、tcp_send()発行から30秒でソケットドライバタイムアウト (リターン値 (/F012)) を検出します。
tcp_close()タイムアウト検出時間 (FINのリトライ間隔)		30秒	相手装置からFINを受信し、正常にコネクションが切断された場合は、すぐに終了します。 LQE520からFINを送信してコネクションを切断する場合も、すぐに終了します。 相手装置からの応答がない場合、下記間隔でFINのリトライをします。 1秒, 2秒, 4秒, 8秒, 16秒 ただし、tcp_close()発行から30秒でソケットドライバタイムアウト (リターン値 (/F012)) を検出するので、tcp_abort()を発行してコネクションを切断してください。
レスポンス タイムアウト 検出時間	tcp_close(), tcp_send(), udp_close()	30秒	ソケットハンドラがマイクロプログラムに対してコマンド発行後、無応答を検出する時間です。
	tcp_abort(), route_list(), route_del(), route_add(), arp_list(), arp_del(), arp_add(), getconfig(), udp_send(), tcp_getaddr(), tcp_stat()	10秒	

5.6 CPU間通信プログラム例

5.6.1 システム構成およびプログラム構成

図5-5にシステム構成を示します。CPU01のET.NETモジュールとCPU02のET.NETモジュールとを論理回線で接続し、CPU02側から1,024バイトのデータを送信し、CPU01側で受信するプログラムです。このプログラムを動作させる場合、必ずCPU01からユーザプログラムを起動してください。

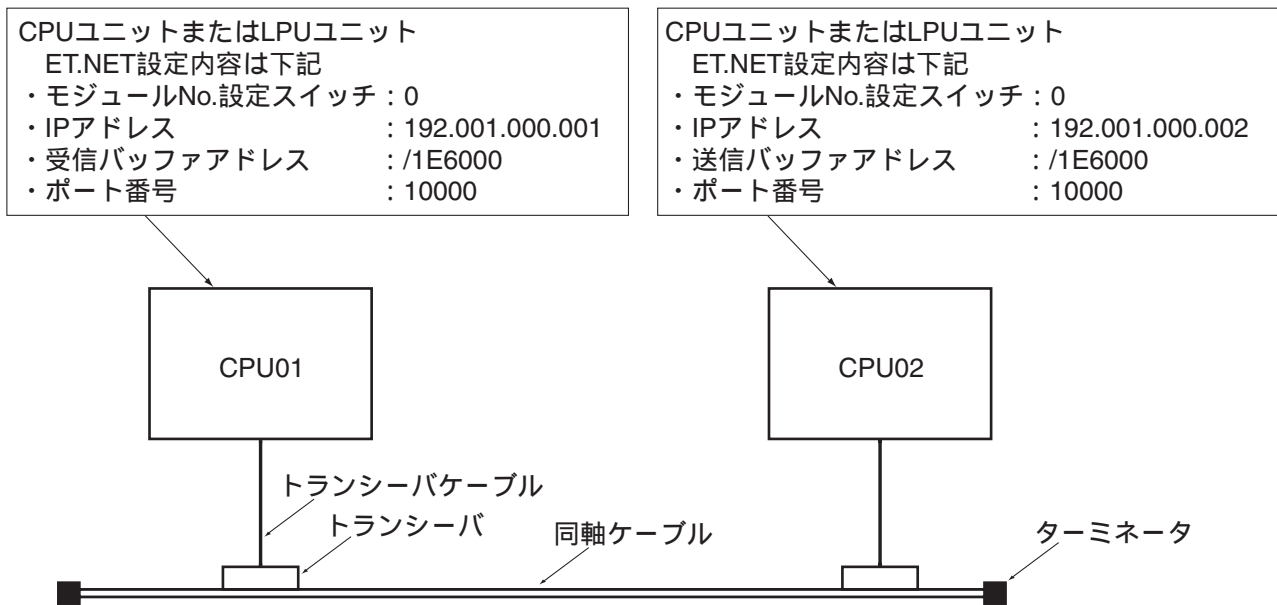


図5-5 システム構成例

5. 6. 2 CPU01側プログラムのフローチャートとプログラム例

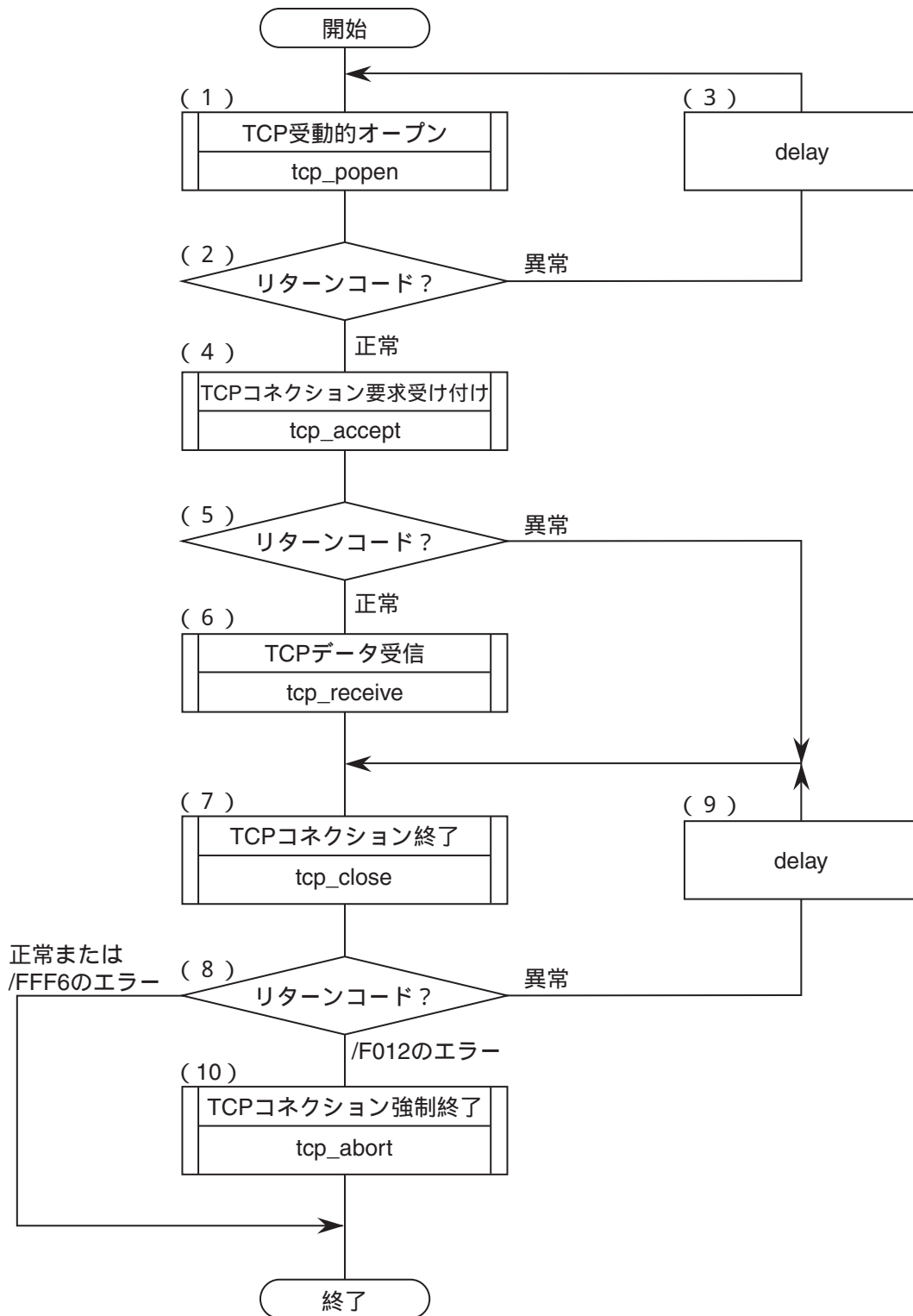


図 5-6 CPU01側プログラムのフローチャート

5 プログラミング

■ フローチャートの動作説明

- (1) ポート番号を10000としてソケットを登録し、そのソケットを受動状態にします。
- (2) 登録されたソケットIDはリターン値で返されますので、リターン値が正であれば正常に登録されたものとみなします。
- (3) (2) でリターン値がエラー発生を示している場合は、**delay**マクロを発行し、(1)、(2)を繰り返します。
- (4) CPU02側からの接続要求に対して、接続要求を受け付けます。
- (5) リターン値により、正常か異常かを判定します。
- (6) CPU02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。
- (7) 接続を終了します。
- (8) リターン値により、正常に接続が終了したかどうかを判定します。エラーコードが/FFF6である場合、その接続はクローズ済みであるので、正常に接続が終了したとみなします。
- (9) (8) でリターン値がエラー発生を示している場合は、**delay**マクロを発行し、(7)、(8)を繰り返します。
- (10) (8) でソケットドライバタイムアウトエラー (エラーコード:/F012) の場合は、接続を強制終了します。

■ プログラム例

```

#define TCP_POPEN      0x874106L    /* tcp_popen() 先頭アドレス (メイン) */
#define TCP_ACCEPT     0x87410CL    /* tcp_accept() 先頭アドレス (メイン) */
#define TCP_CLOSE      0x874112L    /* tcp_close() 先頭アドレス (メイン) */
#define TCP_RECEIVE    0x874136L    /* tcp_receive() 先頭アドレス (メイン) */
#define TCP_ABORT      0x87411EL    /* tcp_abort() 先頭アドレス      */
#define IPADDR         0xC0010002L  /* 相手局IPアドレス      */
#define RBUFADDR       0x1E6000L    /* 受信バッファ先頭アドレス */
#define PARADDR        0x1E5000L    /* パラメータ先頭アドレス  */

struct popen_p {
    long    dst_ip;                /* 相手局のIPアドレス      */
    short   dst_port;              /* 相手局のポート番号      */
    short   src_port;              /* 自局のポート番号        */
    char    listenum;              /* 0固定                    */
    char    ttl;                   /* Time to live            */
};

struct accept_p{
    short   s_id;                  /* ソケットID              */
};

struct receive_p{
    short   s_id;                  /* ソケットID              */
    short   len;                   /* バッファ長              */
    char    *buf;                  /* バッファ先頭アドレス    */
    long    tim;                   /* 受信待ち時間 (ms)      */
};

struct close_p{
    short   s_id;                  /* ソケットID              */
};

struct abort_p{
    short   s_id;                  /* ソケットID              */
};
/*****/
/* task2:サーハ (CPU01) */
/*****/
main()
{
    register short (*tcp_popen)();
    register short (*tcp_accept)();
    register short (*tcp_receive)();
    register short (*tcp_close)();
    register short (*tcp_abort)();
    long    time;
    short   rtn;
    char    *rbuf;
    struct  popen_p    *popen;
    struct  accept_p   *acct;
    struct  receive_p  *recv;
    struct  close_p    *close;

```

5 プログラミング

```

struct abort_p *abort;
popen = (struct popen_p *) PARADDR; /* 入力パラメータ先頭アドレス */
acctp = (struct accept_p *) (popen + 1);
recv = (struct receive_p*) (acctp + 1);
close = (struct close_p *) (recv + 1);
abort = (struct abort_p *) (close + 1);

while( 1 ) {
    popen->dst_ip = IPADDR; /* 相手局のIPアドレス */
    popen->dst_port = 10000; /* 相手局のポート番号 */
    popen->src_port = 10000; /* 自局のポート番号 */
    popen->listenum = 0; /* 0固定 */

    popen->ttl = 0; /* Time to live */
    tcp_popen = ( short (*) ( ) ) TCP_POPEN;
    rtn = (tcp_popen) (popen); /* リターン値正常? */
    if( rtn > 0 ) {
        break;
    }
    time = 100; /* 100ms Delay発行 */
    delay( &time);
}
acctp->s_id = rtn; /* ソケットID */
tcp_accept = ( short (*) ( ) ) TCP_ACCEPT;
rtn = (tcp_accept) (acctp); /* TCPコネクション要求受け付け */
recv->s_id = rtn /* ソケットID */
if( rtn > 0 ) { /* リターン値正常? */
    recv->len = 1024; /* 受信バッファバイト長 */
    recv->buf = ( char *)RBUFADDR; /* 受信バッファ先頭アドレス */
    recv->tim = 60000; /* 受信待ち時間 (ms) */
    tcp_receive = (short (*) ( ) )TCP_RECEIVE;
    rtn = (tcp_receive) (recv); /* TCP受信 */
    close->s_id = recv->s_id; /* ソケットID */
} else {
    close->s_id = acctp->s_id; /* ソケットID */
}
while( 1 ) {
    tcp_close = ( short (*) ( ) ) TCP_CLOSE
    rtn = (tcp_close) (close); /* TCPコネクション終了 */
    if( rtn == 0 || rtn == ( short )0xFFFF6 ) {
        break;
    } else if ( rtn == ( short )0xF012 ) {
        tcp_abort = ( short (*) ( ) ) TCP_ABORT;
        rtn = (tcp_abort) (abort); /* TCPコネクション強制終了 */
        break;
    }
    time = 100; /* 100ms Delay発行 */
    delay( &time);
}
return;
}

```

5. 6. 3 CPU02側プログラムのフローチャートとプログラム例

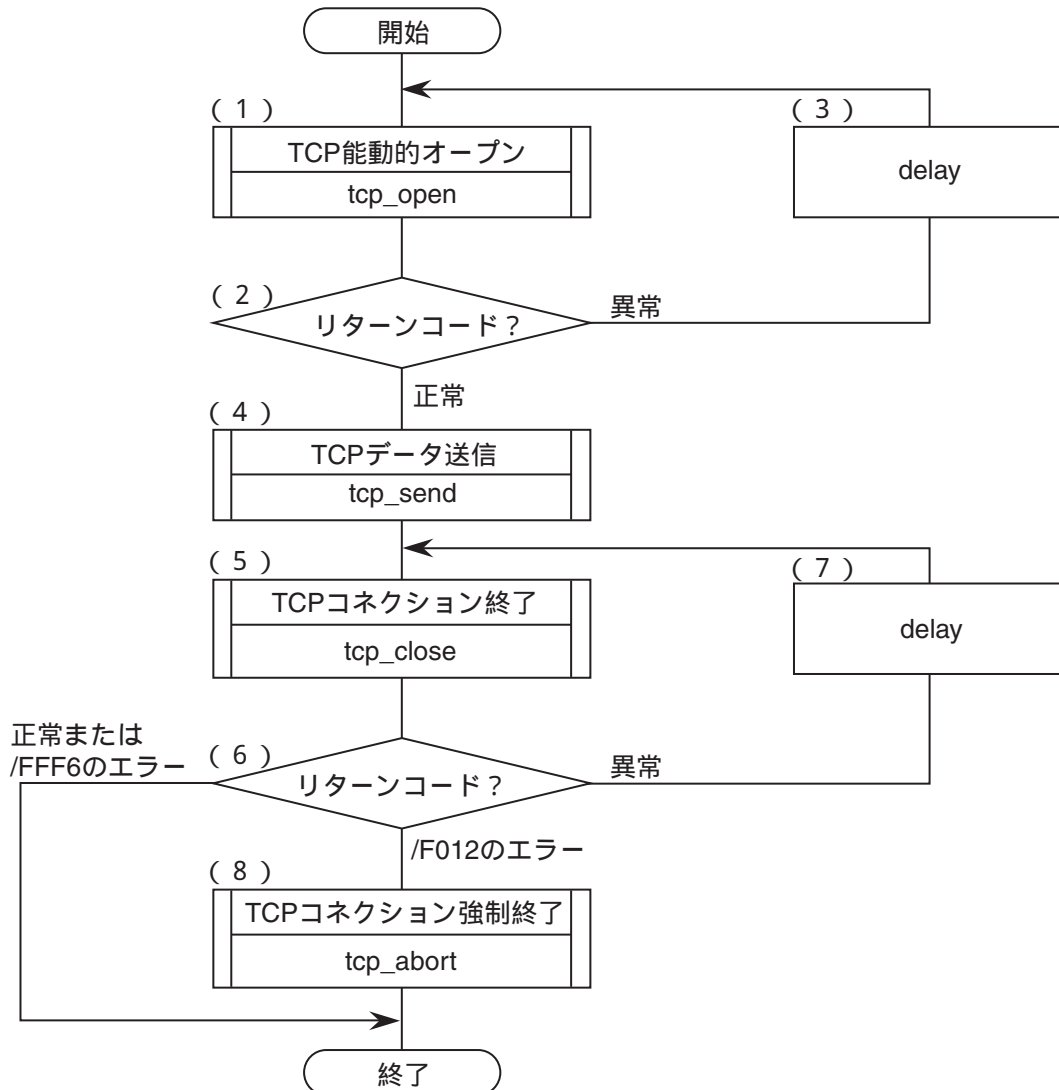


図 5-7 CPU02側プログラムのフローチャート

■ フローチャートの説明

- (1) ポート番号を10000としてソケットを登録し、そのソケットを能動状態にします。
- (2) 登録されたソケットIDはリターン値で返されますので、リターン値が正であれば正常に登録されたものとみなします。
- (3) (2) でリターン値がエラー発生を示している場合は、`delay`マクロを発行し、(1)、(2)を繰り返します。
- (4) 送信バッファのデータをCPU01に送信します。
- (5) コネクションを終了します。
- (6) リターン値により、正常にコネクションが終了したかどうかを判定します。エラーコードが/`FFF6`である場合、そのコネクションはクローズ済みであるので、正常にコネクションが終了したとみなします。
- (7) (6) でリターン値がエラー発生を示している場合は、`delay`マクロを発行し、(5)、(6)を繰り返します。
- (8) 相手局からの応答が返らないので、コネクションを強制終了します。

5 プログラミング

■ プログラム例

```
#define TCP_OPEN      0x874100L    /* tcp_open() 先頭アドレス (メイン) */
#define TCP_CLOSE     0x874112L    /* tcp_close() 先頭アドレス (メイン) */
#define TCP_SEND      0x874130L    /* tcp_send() 先頭アドレス (メイン) */
#define TCP_ABORT     0x87411EL    /* tcp_abort() 先頭アドレス */
#define IPADDR        0xC0010001L  /* 相手局IPアドレス */
#define RBUFADDR      0x1E6000L    /* 受信バッファ先頭アドレス */
#define PARADDR       0x1E5000L    /* パラメータ先頭アドレス */

struct open_p {
    long    dst_ip;                /* 相手局のIPアドレス */
    short   dst_port;              /* 相手局のポート番号 */
    short   src_port;              /* 自局のポート番号 */
    char    notuse;                /* 未使用 (0) */
    char    ttl;                   /* Time to live */
};

struct send_p {
    short   s_id;                  /* ソケットID */
    short   len;                   /* 送信データバイト長 */
    char    *buf;                  /* 送信データ先頭アドレス */
};

struct close_p {
    short   s_id;                  /* ソケットID */
};

struct abort_p {
    short   s_id;                  /* ソケットID */
};

/*****
/* task3:クライアント(CPU02) */
*****/
main()
{
    register short (*tcp_open)();
    register short (*tcp_send)();
    register short (*tcp_close)();
    register short (*tcp_abort)();
    long    time;
    short   rtn;
    struct  open_p    *open;
    struct  send_p    *send;
    struct  close_p   *close;
    struct  abort_p   *abort;
    open = (struct open_p *) PARADDR; /* 入力パラメータ先頭アドレス */
    send = (struct send_p *) (open + 1);
    close = (struct close_p *) (send + 1);
    abort = (struct abort_p *) (close + 1);

    while(1) {
        open->dst_ip = IPADDR; /* 相手局のIPアドレス */
        open->dst_port = 10000; /* 相手局のポート番号 */
    }
}
```

```

    open->src_port    = 10000;          /* 自局のポート番号 */
    open->notuse      = 0;              /* 未使用 */
    open->tll         = 0;              /* Time to live */
    tcp_open         = (short (*)()) TCP_OPEN;
    rtn              = (tcp_open) (open); /* TCP能動的オープン */
    if( rtn > 0 ) { /* リターン値正常? */
        break;
    }
    time = 100; /* 100ms Delay発行 */
    delay( &time);
}
send->s_id          = rtn; /* ソケットID */
send->len           = 1024; /* 送信データバイト長 */
send->buf           = (char *) SBUFADDR; /* 送信データ先頭アドレス */
tcp_send           = (short (*)()) TCP_SEND;
rtn                = (tcp_send) (send); /* TCPデータ送信 */
close->s_id         = send->s_id; /* ソケットID */
while( 1 ) {
    tcp_close = (short (*)()) TCP_CLOSE
    rtn = (tcp_close) (close); /* TCPコネクション終了 */
    if( rtn == 0 || rtn == (short)0xFFF6 ) {
        break;
    } else if ( rtn == (short)0xF012 ) {
        tcp_abort = (short (*)()) TCP_ABORT;
        rtn = (tcp_abort) (abort); /* TCPコネクション強制終了 */
        break;
    }
    time = 100; /* 100ms Delay発行 */
    delay( &time);
}
return;
}

```


5.7 CPU間連続通信プログラム例

5.7.1 システム構成およびプログラム構成

図5-8にシステム構成を示します。CPU01のET.NETモジュールとCPU02のET.NETモジュールとを論理回線で接続し、CPU02とCPU01で1,024バイトのデータを送受信するプログラムです。このプログラムを動作させる場合、必ずCPU01からユーザプログラムを起動してください。

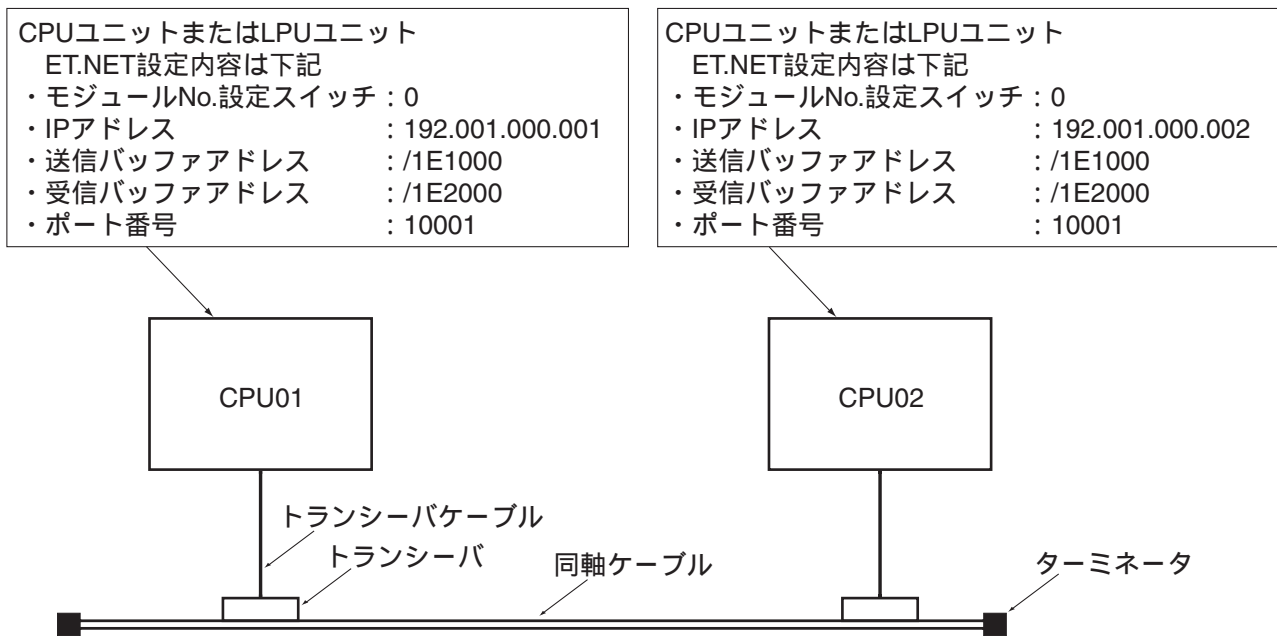


図5-8 システム構成例

5.7.2 CPU01側プログラムのフローチャートとプログラム例

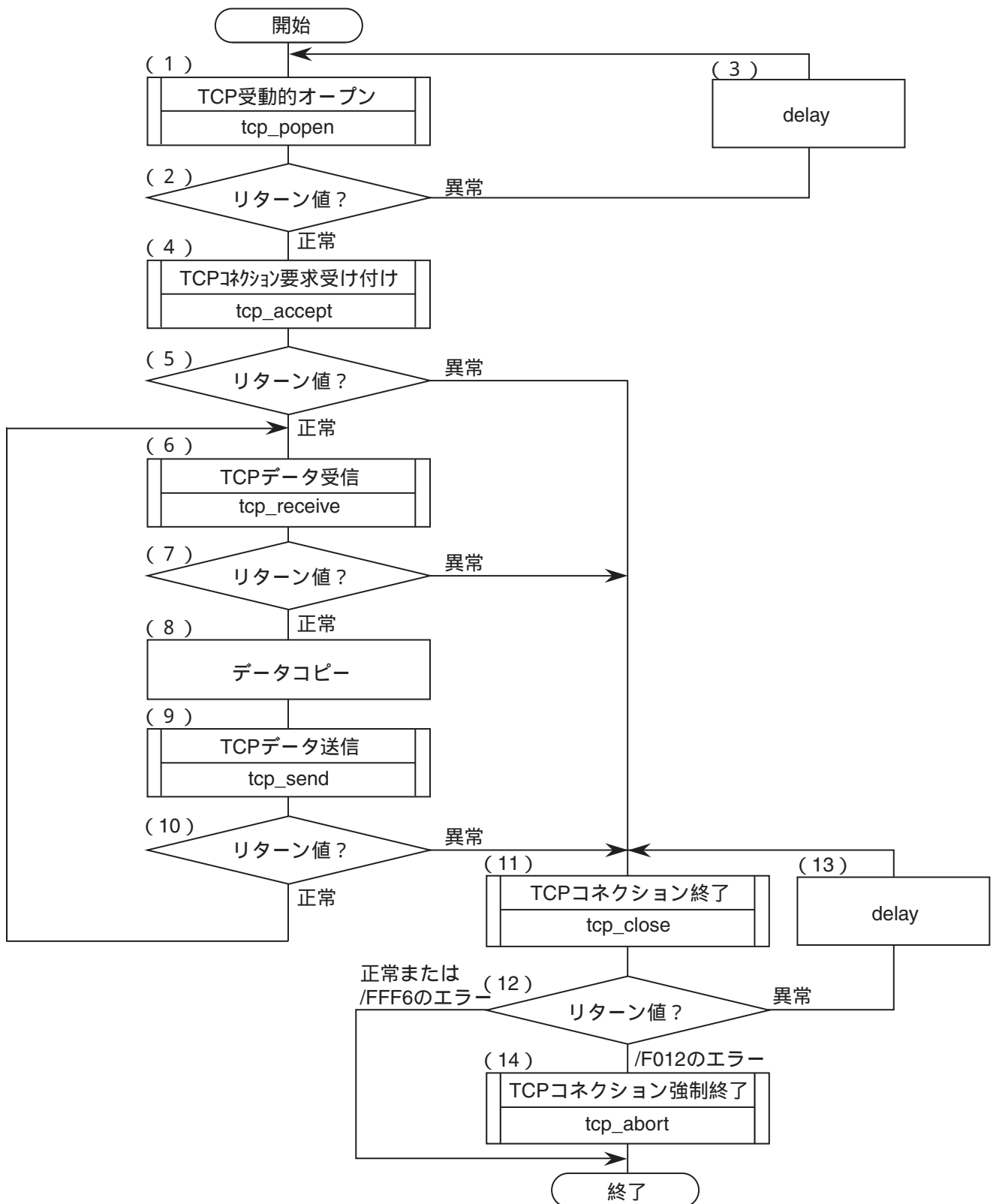


図5-9 CPU01側プログラムのフローチャート

5 プログラミング

■ フローチャートの動作説明

- (1) ポート番号を10001としてソケットを登録し、そのソケットを受動状態にします。
- (2) 登録されたソケットIDはリターン値で返されますので、リターン値が正であれば正常に登録されたものとみなします。
- (3) (2) でリターン値がエラー発生を示している場合は、**delay**マクロを発行し、(1)、(2)を繰り返します。
- (4) CPU02側からの接続要求に対して、接続要求を受け付けます。
- (5) リターン値により、正常か異常かを判定します。
- (6) CPU02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。
- (7) リターン値がエラー、または取り込みデータなしの場合は、(11) を実行します。
- (8) 受信バッファのデータを送信バッファへコピーします。
- (9) 送信バッファのデータをCPU02に送信します。
- (10) リターン値により正常か異常かを判定し、正常である場合は(6)～(10)を繰り返します。
- (11) コネクションを終了します。
- (12) リターン値により、正常にコネクションが終了したかどうかを判定します。エラーコードが/FFF6である場合、そのコネクションはクローズ済みであるので、正常にコネクションが終了したとみなします。
- (13) (12) でリターン値がエラー発生を示している場合は、**delay**マクロを発行し、(11)、(12)を繰り返します。
- (14) 相手局からの応答が返らないので、コネクションを強制終了します。

■ プログラム例

```

#define TCP_POPEN      0x874106L    /* tcp_popen() 先頭アドレス (メイン) */
#define TCP_ACCEPT     0x87410CL    /* tcp_accept() 先頭アドレス (メイン) */
#define TCP_RECEIVE    0x874136L    /* tcp_receive() 先頭アドレス (メイン) */
#define TCP_SEND       0x874130L    /* tcp_send() 先頭アドレス (メイン) */
#define TCP_CLOSE      0x874112L    /* tcp_close() 先頭アドレス (メイン) */
#define TCP_ABORT      0x87411EL    /* tcp_abort() 先頭アドレス */
#define IPADDR         0xC0010002L  /* 相手局IPアドレス */
#define SBUFADDR       0x1E1000L    /* 送信バッファ先頭アドレス */
#define RBUFADDR       0x1E2000L    /* 受信バッファ先頭アドレス */
#define PARADDR        0x1E5000L    /* パラメータ先頭アドレス */

struct popen_p {
    long    dst_ip;                /* 相手局のIPアドレス */
    short   dst_port;              /* 相手局のポート番号 */
    short   src_port;              /* 自局のポート番号 */
    char    listenum;              /* 0固定 */
    char    ttl;                   /* Time to live */
};

struct accept_p {
    short   s_id;                  /* ソケットID */
};

struct receive_p {
    short   s_id;                  /* ソケットID */
    short   len;                   /* 受信バッファ長 */
    char    *buf;                  /* 受信バッファ先頭アドレス */
    long    tim;                   /* 受信待ち時間 */
};

struct send_p {
    short   s_id;                  /* ソケットID */
    short   len;                   /* 送信データバイト数 */
    char    *buf;                  /* 送信データ先頭アドレス */
};

struct close_p {
    short   s_id;                  /* ソケットID */
};

struct abort_p {
    short   s_id;                  /* ソケットID */
};
/*****
/* task2:カーネル (CPU01) */
*****/
main()
{
    register short (*tcp_popen)();
    register short (*tcp_accept)();
    register short (*tcp_receive)();
    register short (*tcp_send)();
    register short (*tcp_close)();

```

5 プログラミング

```

register    short ( *tcp_abort ) ( );
long       time;
short      rtn, i;
char       *sbuf, *rbuf;
struct     popen_p      *popen;
struct     accept_p     *acct;
struct     receive_p    *recv;
struct     send_p       *send;
struct     close_p      *close;
struct     abort_p      *abort;
popen      = (struct popen_p *) PARADDR; /* 入力パラメータ先頭アドレス */
acct       = (struct accept_p *) (popen + 1);
recv       = (struct receive_p *) (acct + 1);
send       = (struct send_p *) (recv + 1);
close      = (struct close_p *) (send + 1);
abort      = (struct abort_p *) (close + 1);

while( 1 ) {
    popen->dst_ip      = IPADDR; /* 相手局のIPアドレス */
    popen->dst_port    = 10001; /* 相手局のポート番号 */
    popen->src_port    = 10001; /* 自局のポート番号 */
    popen->listenum    = 0; /* 0固定 */

    popen->ttl = 0; /* Time to live */
    tcp_popen = ( short (*) ( ) ) TCP_POPEN;
    rtn       = (tcp_popen) (popen); /* リターン値正常? */
    if( rtn > 0 ) {
        break;
    }
    time = 100; /* 100ms Delay発行 */
    delay( &time);
}
acct->s_id    = rtn; /* ソケットID */
tcp_accept   = ( short (*) ( ) ) TCP_ACCEPT;
rtn          = (tcp_accept) (acct); /* TCPコネクション要求受け付け */
if( rtn > 0 ) { /* リターン値正常? */
    recv->s_id    = rtn /* ソケットID */
    while( 1 ) {
        recv->len = 1024; /* 受信バッファバイト長 */
        recv->buf  = ( char *)RBUFADDR; /* 受信バッファ先頭アドレス */
        recv->tim = 60000; /* 受信待ち時間 (ms) */
        tcp_receive = (short (*) ( ) )TCP_RECEIVE;
        rtn         = (tcp_receive) (recv); /* TCP受信 */
        if( rtn > 0 ) { /* リターン値正常? */
            break;
        }
        sbuf = ( char *)SBUFADDR;
        rbuf = ( char *)RBUFADDR;
        for( i = 0 ; i < 1024 ; i++ ) {
            sbuf[ i ] = rbuf[ i ];
        }
        send->s_id = recv->s_id; /* ソケットID */
        send->len  = 1024; /* 送信データバイト長 */
        send->buf  = ( char *)SBUFADDR; /* 送信データ先頭アドレス */
        tcp_send  = ( short (*) ( ) ) TCP_SEND;

```

```
    rtn          = (*tcp_send) (send);      /* TCPデータ送信 */
    if ( rtn < 0 ) {                       /* リターン値正常? */
        break;
    }
}
close->s_id     = recv->s_id;              /* ソケットID */
} else {
close->s_id = accept->s_id;                /* ソケットID */
}
while( 1 ) {
    tcp_close = ( short (*) ( ) ) TCP_CLOSE
    rtn = (tcp_close) (close);             /* TCPコネクション終了 */
    if( rtn == 0 || rtn == ( short )0xFFF6 ) {
        break;
    } else if ( rtn == ( short )0xF012 ) {
        tcp_abort = ( short (*) ( ) ) TCP_ABORT;
        rtn = (tcp_abort) (abort);        /* TCPコネクション強制終了 */
        break;
    }
}
time = 100;                               /* 100ms Delay発行 */
delay( &time);
}
return;
}
```

5 プログラミング

5.7.3 CPU02側プログラムのフローチャートとプログラム例

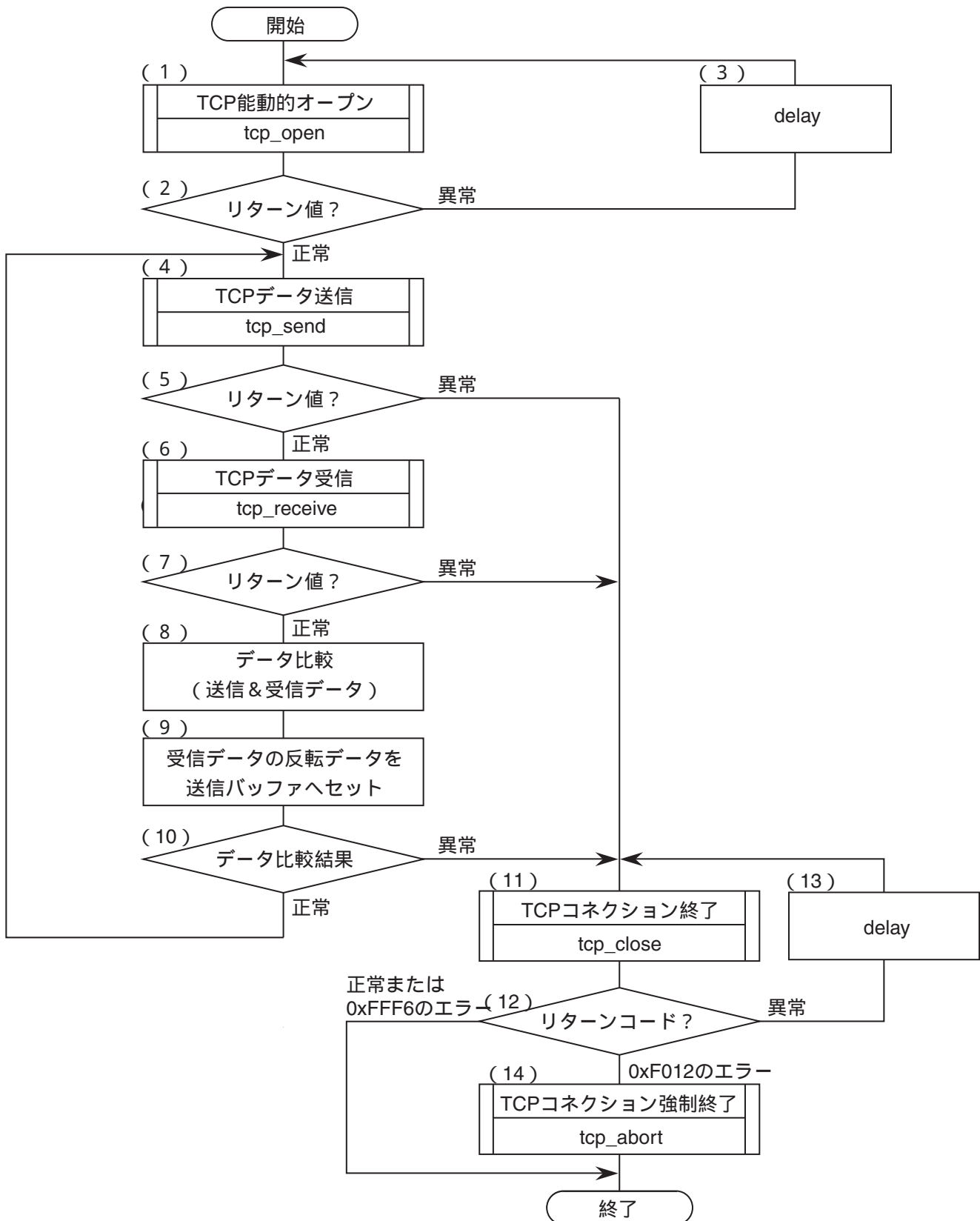


図 5-10 CPU02側プログラムのフローチャート

■ フローチャートの動作説明

- (1) ポート番号を10001としてソケットを登録し、そのソケットを能動状態にします。
- (2) 登録されたソケットIDはリターン値で返されますので、リターン値が正であれば正常に登録されたものとみなします。
- (3) (2) でリターン値がエラー発生を示している場合は、**delay**マクロを発行し、(1)、(2)を繰り返します。
- (4) 送信バッファのデータをCPU01側に送信します。
- (5) リターン値により、正常か異常かを判定します。
- (6) CPU01から送信されたデータを受信バッファへ取り込みます。
- (7) リターン値により、正常か異常かを判定します。
- (8) 自局の送信バッファと受信バッファのデータを比較します。
- (9) 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。
- (10) 比較結果を判定し、正常な場合は(4)～(10)を繰り返します。
- (11) コネクションを終了します。
- (12) リターン値により、正常にコネクションが終了したかどうかを判定します。エラーコードが/FFF6である場合、そのコネクションはクローズ済みであるので、正常にコネクションが終了したとみなします。
- (13) (12) でリターン値がエラー発生を示している場合は、**delay**マクロを発行し、(11)、(12)を繰り返します。
- (14) 相手局からの応答が返らないので、コネクションを強制終了します。

5 プログラミング

■ CPU02側のプログラム例

```
#define TCP_OPEN          0x874100L    /* tcp_open() 先頭アドレス (メイン) */
#define TCP_CLOSE        0x874112L    /* tcp_close() 先頭アドレス (メイン) */
#define TCP_SEND         0x874130L    /* tcp_send() 先頭アドレス (メイン) */
#define TCP_RECEIVE     0x874136L    /* tcp_receive() 先頭アドレス (メイン) */
#define TCP_ABORT       0x87411EL    /* tcp_abort() 先頭アドレス */
#define IPADDR          0xC0010001L   /* 相手局IPアドレス */
#define SBUFADDR        0x1E1000L    /* 受信バッファ先頭アドレス */
#define RBUFADDR        0x1E2000L    /* 受信バッファ先頭アドレス */
#define PARADDR         0x1E5000L    /* パラメータ先頭アドレス */

struct open_p {
    long    dst_ip;          /* 相手局のIPアドレス */
    short   dst_port;       /* 相手局のポート番号 */
    short   src_port;       /* 自局のポート番号 */
    char    notuse;         /* 未使用 (0) */
    char    ttl;            /* Time to live */
};

struct send_p{
    short   s_id;           /* ソケットID */
    short   len;            /* 送信データバイト長 */
    char    *buf;           /* 送信データ先頭アドレス */
};

struct receive_p{
    short   s_id;           /* ソケットID */
    short   len;            /* 送信データバイト長 */
    char    *buf;           /* 送信データ先頭アドレス */
    long    tim             /* 受信待ち時間 (ms) */
};

struct close_p{
    short   s_id;           /* ソケットID */
};

struct abort_p{
    short   s_id;           /* ソケットID */
};
/*****
/* task3:クライアント(CPU02) */
*****/
main()
{
    register short (*tcp_open)();
    register short (*tcp_send)();
    register short (*tcp_receive)();
    register short (*tcp_close)();
    register short (*tcp_abort)();
    long    time;
    short   rtn, i, cerr_flg;
    struct  open_p    *open;
    struct  send_p    *send;
    struct  receive_p *receive;
```

```

struct      close_p      *close;
struct      abort_p      *abort;

open = (struct open_p      *) PARADDR;      /* 入力パラメータ先頭アドレス */
send = (struct send_p      *) (open + 1);
recv = (struct receive_p   *) (send + 1);
close = (struct close_p    *) (recv + 1);
abort = (struct abort_p    *) (close + 1);

sbuf = ( char *) SBUFADDR;
for( i = 0 ; i < 1024 ; i++ ) {
    sbuf[ i ] = 0x55;
}

while( 1 ) {
    open->dst_ip = IPADDR;      /* 相手局のIPアドレス */
    open->dst_port = 10001;      /* 相手局のポート番号 */
    open->src_port = 10001;      /* 自局のポート番号 */
    open->notuse = 0;           /* 未使用 */
    open->ttl = 0;             /* Time to live */
    tcp_open = ( short (*) ( ) ) TCP_OPEN;
    rtn = (tcp_open) (open);    /* TCP能動的オープン */
    if( rtn > 0 ) {           /* リターン値正常? */
        break;
    }
    time = 100;               /* 100ms Delay発行 */
    delay( &time);
}
send->s_id = rtn;             /* ソケットID */
recv->s_id = rtn;             /* ソケットID
while( 1 ) {
    send->len = 1024 ;        /* 送信データバイト長 */
    send->buf = ( char *) SBUFADDR ; /* 送信データ先頭アドレス */
    tcp_send = ( short (*) ( ) ) TCP_SEND;
    rtn = (tcp_send) (send); /* TCPデータ受信 */
    if( rtn < 0 ) {           /* リターンコード異常? */
        break
    }
    recv->len = 1024;         /* 受信バッファバイト長 */
    recv->buf = ( char *) RBUFADDR ; /* 受信バッファ先頭アドレス */
    recv->tim = 60000 ;      /* 受信待ち時間 (ms) */
    rcp_receive = ( short (*) ( ) ) TCP_RECEIVE;
    rtn = (tcp_receive) ( recv ); /* TCPデータ受信 */
    if( rtn < 0 ) {           /* リターンコード異常? */
        break;
    }
    cerr_flg = 0;           /* コンパイルエラーフラグクリア */
    sbuf = ( char *)SBUFADDR; /* 送信バッファ先頭アドレス */
    rbuf = ( char *)RBUFADDR; /* 受信バッファ先頭アドレス */
    for( i = 0 ; i < 1024 ; i++ ) {
        if( sbuf[ i ] != rbuf[ i ] ) {
            cerr_flg = 1; /* コンパイルエラーフラグセット */
            break;
        }
    }
    sbuf[ i ] = ~rbuf[ i ]; /* 反転データセット */
}

```

5 プログラミング

```
    }
    if( cerr_flg == 1 ) {
        break;
    }
}
close -> s_id = send -> s_id;
while( 1 ) {
    tcp_close = ( short (*) ( ) ) TCP_CLOSE
    rtn = (tcp_close) (close);
    if( rtn == 0 || rtn == ( short )0xFFFF6 ) {
        break;
    } else if ( rtn == ( short )0xF012 ) {
        tcp_abort = ( short (*) ( ) ) TCP_ABORT;
        rtn = (tcp_abort) (abort);
        break;
    }
    time = 100;
    delay( &time);
}
return;
}
```

/* コンパイルエラー? */

/* ソケットID */

/* TCPコネクション終了 */

/* TCPコネクション強制終了 */

/* 100ms Delay発行 */

6 利用の手引き

6 利用の手引き

6.1 推奨するネットワーク構成部品

LQE520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、同じ規格に準拠した異社間のネットワーク構成部品と組み合わせた場合、相性によって正常に動作しない場合があります。したがって、LQE520と接続するネットワーク構成部品は、すべて弊社の推奨するものを使用してください。

推奨するネットワーク構成部品を表6-1および図6-1に示します。

なお、Ethernet®の仕様には、IEEE802.3規格と、オリジナルEthernet®仕様とがあります。LQE520には、オリジナルEthernet®仕様の機器を接続できません。

表6-1 ネットワーク構成部品一覧

No.	品名	メーカー	型式	備考
①	ET.NET	(株)日立製作所	LQE520	
②	トランシーバ	日立金属(株)	HLT-200TB HBN200TZ HBN200TD	タップ形
③	トランシーバ	日立金属(株)	HLT-200	コネクタ形
④	リピータ	日立金属(株)	HLR-200H	同軸ケーブルの伝送距離延長装置
⑤	マルチポート トランシーバ	(株)日立製作所	H-7612-64 H-7612-68	4ポート/8ポート AC電源内蔵
⑥	同軸ケーブル	日立金属(株)	HBN-CX-100	屋内用、最長500m
⑦	同軸コネクタ	日立金属(株)	HBN-N-PC	同軸ケーブル用
⑧	中継コネクタ	日立金属(株)	HBN-N-AJJ	同軸ケーブル用
⑨	ターミネータ	日立金属(株)	HBN-T-NJ	J形
⑩	ターミネータ	日立金属(株)	HBN-T-NP	P形
⑪	アース端子	日立金属(株)	HBN-G-TM	同軸ケーブル用
⑫	トランシーバ ケーブル	日立金属(株)	HBN-TC-100	オス、メスD-sub15ピンコネクタ付、 最長50m
⑬	ツイストペア ケーブル	日立金属(株)	HUTP-CAT5-4P	
⑭	マルチポート トランシーバ	日立金属(株)	HBM-400TZ	4ポート

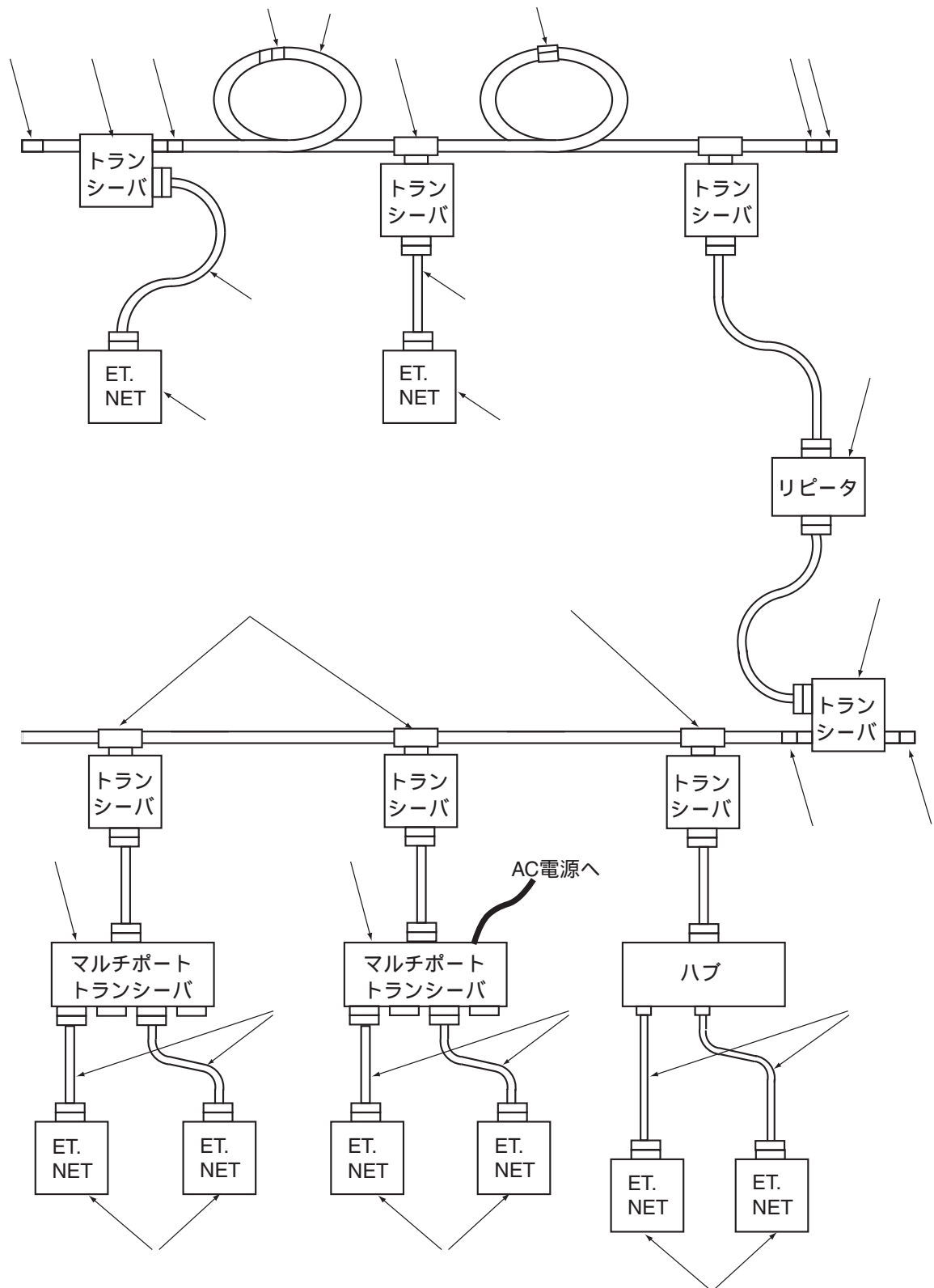


図 6-1 ネットワーク構成部品

6.2 10BASE-5のシステム構成

6.2.1 10BASE-5のシステム構成概要

10BASE-5の基本構成は、図6-2に示すとおり、最長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（ステーションとは、LQE520を含むイーサネット機器を示します）。この基本構成をセグメントといいます。1つのセグメントに接続できるステーションは、最大100台です。

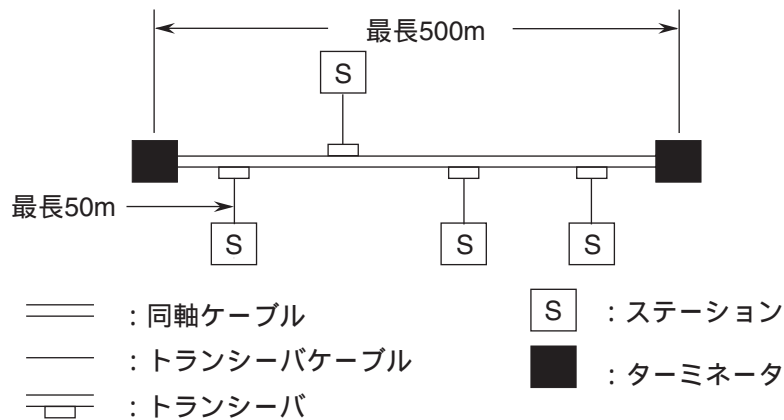


図6-2 10BASE-5の最小構成

ステーション間の距離が500m以上となる場合は、リピータを接続して分岐状にセグメントの数を増やすことになります。図6-3は、ステーション間最大距離が1,500mのシステム例です。任意の2つのステーション間は、どの経路を通過してもリピータの数が2台以下となるように構成してください。

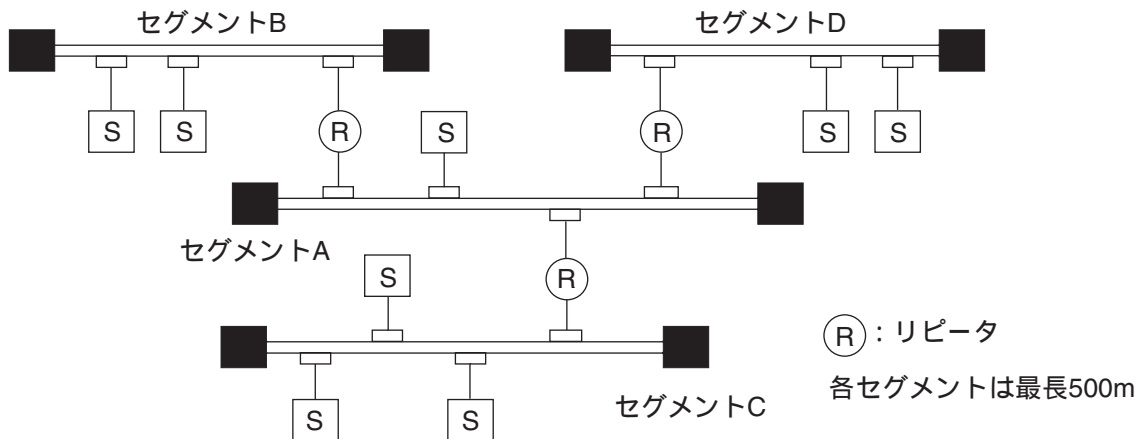


図6-3 10BASE-5の中規模構成

図6-4は、ステーション間の最大距離を2,500mとした例です。伝送距離を長くするため、リピータを両端に設置したリンクケーブル（同軸ケーブルの場合、最長500m）を用いて、これをリンクセグメントと呼びます。リンクセグメントにはステーションを接続せず、その代わり両端のリピータを含めて点線で囲った部分を1台のリピータとして数えることができ、任意ステーション間のリピータ合計台数の制限を軽減できます。

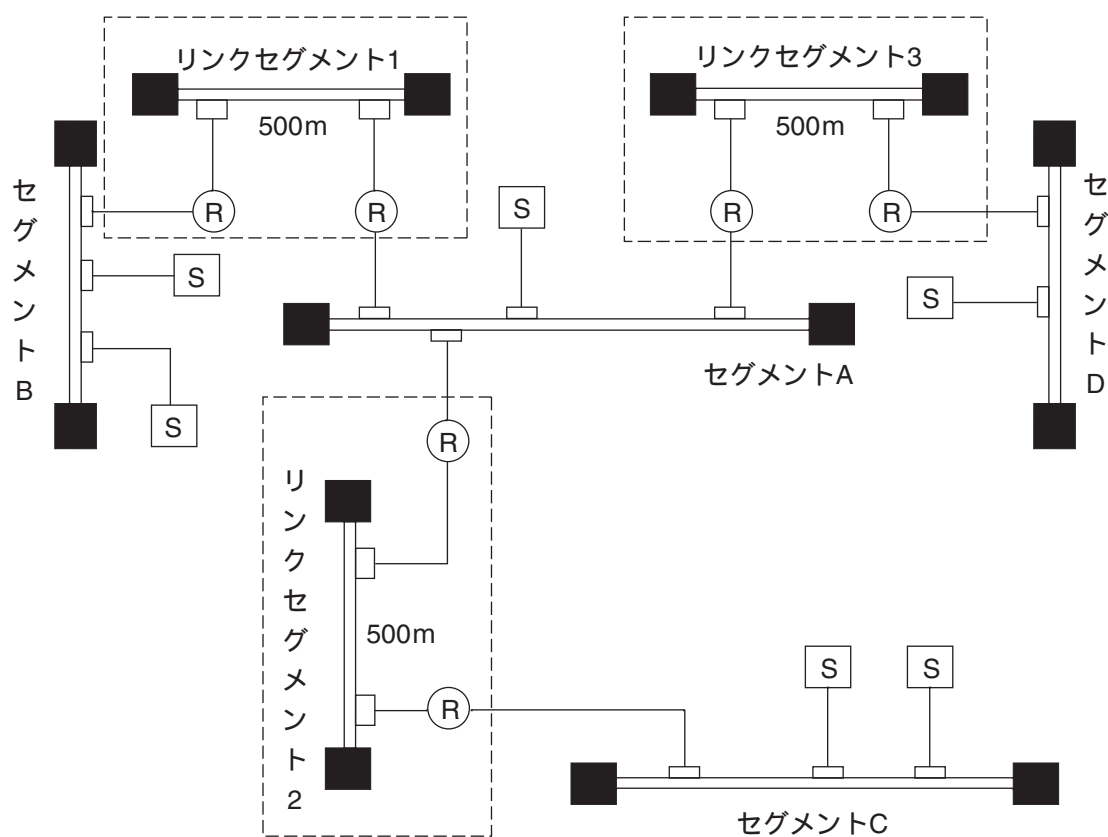


図6-4 10BASE-5の大規模構成

システム構成上のパラメータを以下に示します。

表6-2 10BASE-5システム構成上のパラメータ

項目	仕様
最大セグメント長	500m
セグメント内トランシーバ取り付け最大数	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除く）
最大ステーション数	1,024台
最大トランシーバケーブル長	50m
ステーション間経路内リピータ最大数	2台（ただし、リンクセグメントは両端のリピータを含めて、全体を1台のリピータとみなします。）

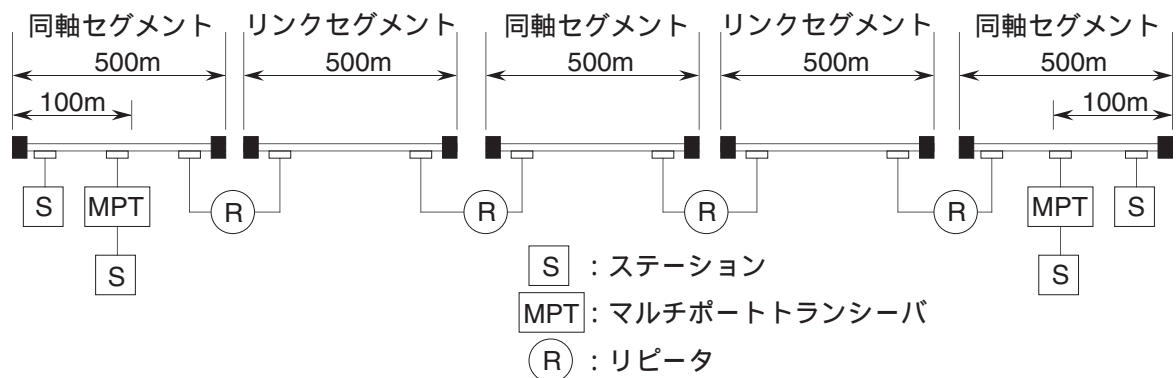
6 利用の手引き

6. 2. 2 10BASE-5システム構成上の注意

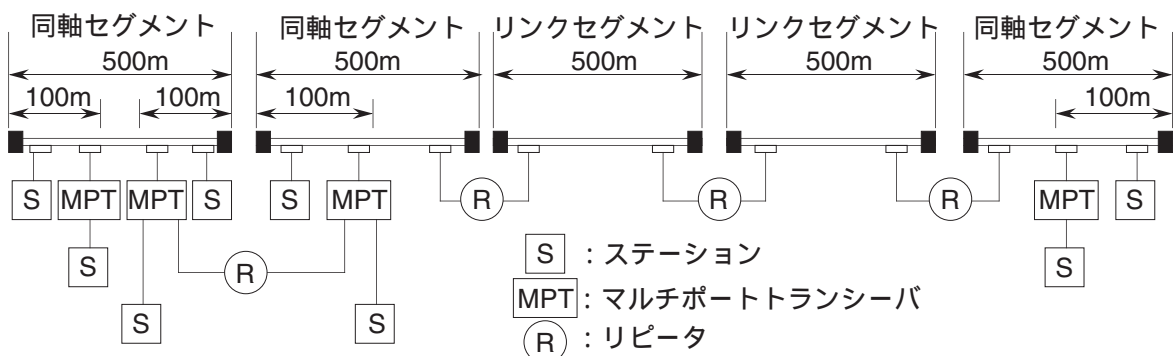
- ・リピータは、トランシーバケーブルとトランシーバを介して同軸ケーブルに接続してください。
- ・トランシーバの取り付け間隔は、2.5mの整数倍としてください。
- ・リンクケーブルには、ステーションを取り付けないでください。
- ・リピータは、どの位置のトランシーバにも取り付けられます。
- ・任意のステーション間のリピータは、2個以下にしてください。
- ・リピータが2つ以上接続されるセグメントは、1つのみとしてください。
- ・ツールシステムと接続し、MCSなどの画面を開く場合は、4画面までしか開きません。
- ・マルチポートトランシーバを経由したステーション間の最大距離は、マルチポートトランシーバ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。すなわち、2つのステーション間の経路の同軸ケーブル線長をL、経由するマルチポートトランシーバの総数をNとすると、LとNは下式の関係になります。

$$L[m] \leq 2,500[m] - 100 \times N[m]$$

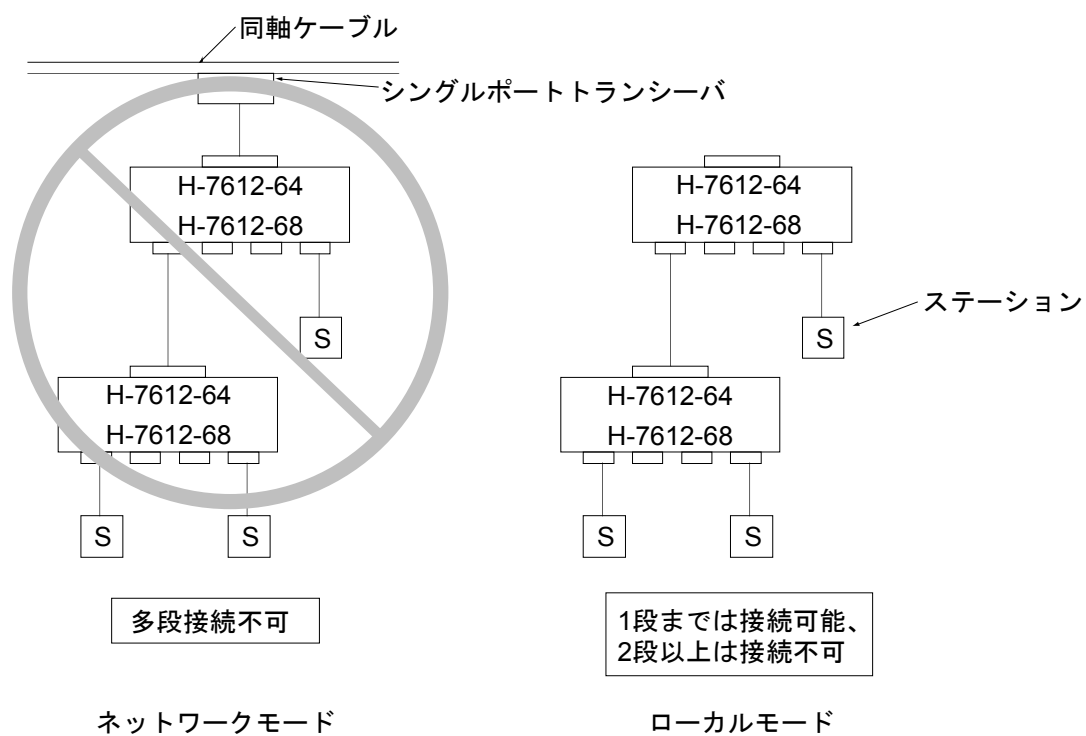
(例1) 2,500mの同軸ケーブルで構成されるシステムでは、マルチポートトランシーバを最遠端のターミネータから100m以上内側（ステーション間の距離を減少させる位置）に接続してください。



(例2) マルチポートトランシーバを経由してリピータを接続する場合も、マルチポートトランシーバを1台通過することにより、最遠端のステーション間距離を100m減少させる位置にマルチポートトランシーバを接続してください。



- マルチポートトランシーバ（H-7612-64/68）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。



- 同軸ケーブル、トランシーバケーブル、トランシーバなどのネットワーク構成機器は、表6-1および図6-1に示すものを用いてください。

6.3 10BASE-Tのシステム構成

図6-5のように、トランシーバに、トランシーバケーブル（AUIケーブル）を経由して、ハブ（マルチポートリピータ）を接続することにより、ハブに複数のステーションを接続できます。ハブにステーションを接続するには、ツイストペアケーブルを使用します。

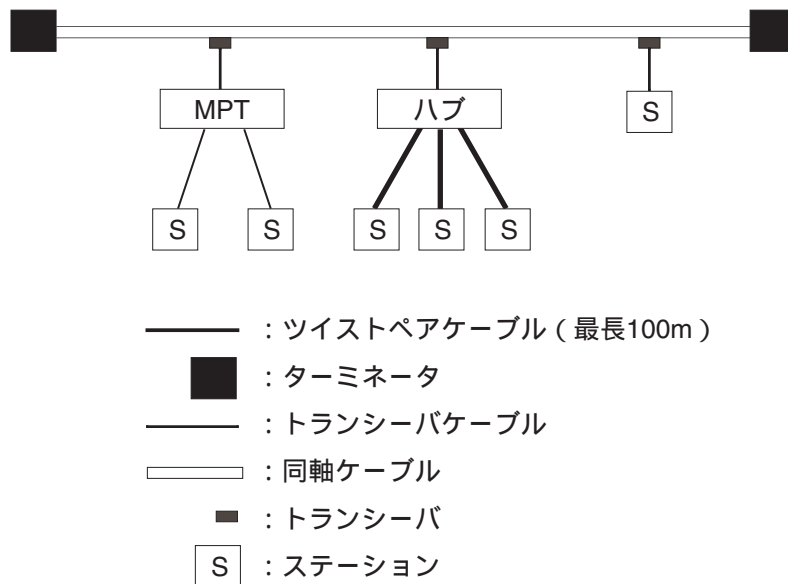


図6-5 10BASE-Tのシステム構成

また、ステーション間の距離が短い場合は、図6-6のように、同軸ケーブルやトランシーバなしで、ハブにツイストペアケーブルを介してステーションを接続できます。

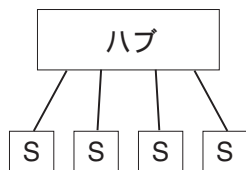
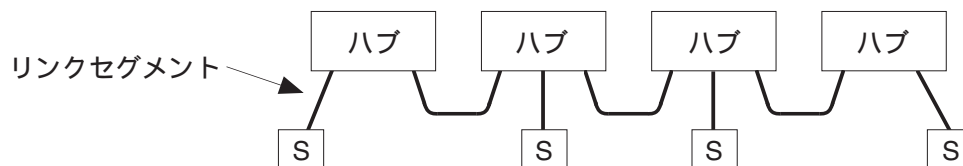


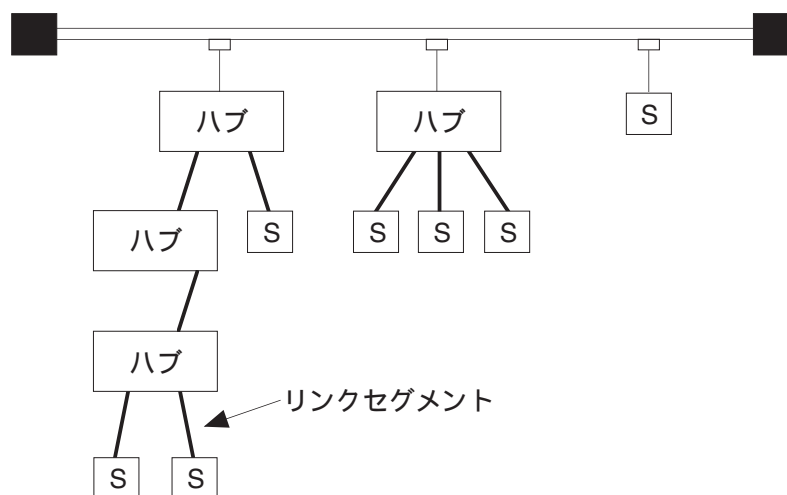
図6-6 ハブのみによる10BASE-Tの構成例

■ ハブを多段接続する際の制約

- ハブを多段接続して使用する場合は、任意のステーション間の経路において、通過するハブは最大4段、かつリンクセグメントは最大5本となるように構成してください。



- 同軸ケーブルに接続して使用する場合も、任意のステーション間の経路において、通過するハブは最大4段、かつセグメントは最大5本（同軸セグメントは3本まで）となるように構成してください。



6.4 ネットワーク構成部品の設置、配線、および設定

6.4.1 同軸ケーブルの配線

- (1) 同軸ケーブルは、屋内の配線ダクトに布設、配線し、100V以上の配線とは区別してください。また、ケーブル布設前には、必ず短絡や断線がないかどうかチェックしてください。
- (2) 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り付け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
- (3) 布設配線工事上の留意事項は、以下のとおりです。
 - ・ケーブルは、原則として屋内に布設、配線してください。
 - ・ケーブルの質量は、約1.9kg/10mです。
 - ・ケーブルの布設中、ケーブルに245N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、布設時および最終固定時ともに250mm（やむを得ないときは150mm）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を1mとしてください。その際、サドルの締め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、損失を受けやすいので、結びなどにより保護してください。
 - ・ケーブルの外部導体は接地してください。接地する場合は、1セグメントの1点で、D種接地以上の接地をしてください。接地点以外のケーブルの金属露出部分が、大地や他の金属部分に接触しないよう、コネクタやターミネータは付属のブーツをかぶせるか、絶縁テープを巻いて絶縁してください。

6. 4. 2 トランシーバの設置および配線

- (1) トランシーバの設置場所および配線方法は、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
 - ・ ステーションのそば
- (2) トランシーバを設置する上での留意事項は以下のとおりです。
 - ・ 取り付け金具を介して、木ねじなどで固定してください。
 - ・ 設置間隔は、2.5m以上としてください。
- (3) トランシーバは、ケーブルに無理な力が加わらないよう、4箇所のおねじ穴で固定してください。
- (4) トランシーバへの同軸ケーブル接続は、同軸コネクタを用います。同軸コネクタの外部導体は大地電位から浮いています。したがって、同軸コネクタは他の金属に触れないよう、ゴムブーツやビニールテープなどで絶縁してください。また、トランシーバ本体のケースはトランシーバケーブルを接続することで大地電位となります。したがって、トランシーバ本体のケースも他の金属に触れないように絶縁してください。
- (5) 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。
 - ・ コネクタおよびターミネータの緩みを確認できること
 - ・ 同軸コネクタの緩みを確認できること
 - ・ 付帯のLEDを確認できること

6 利用の手引き

トランシーバ、トランシーバケーブル設置例

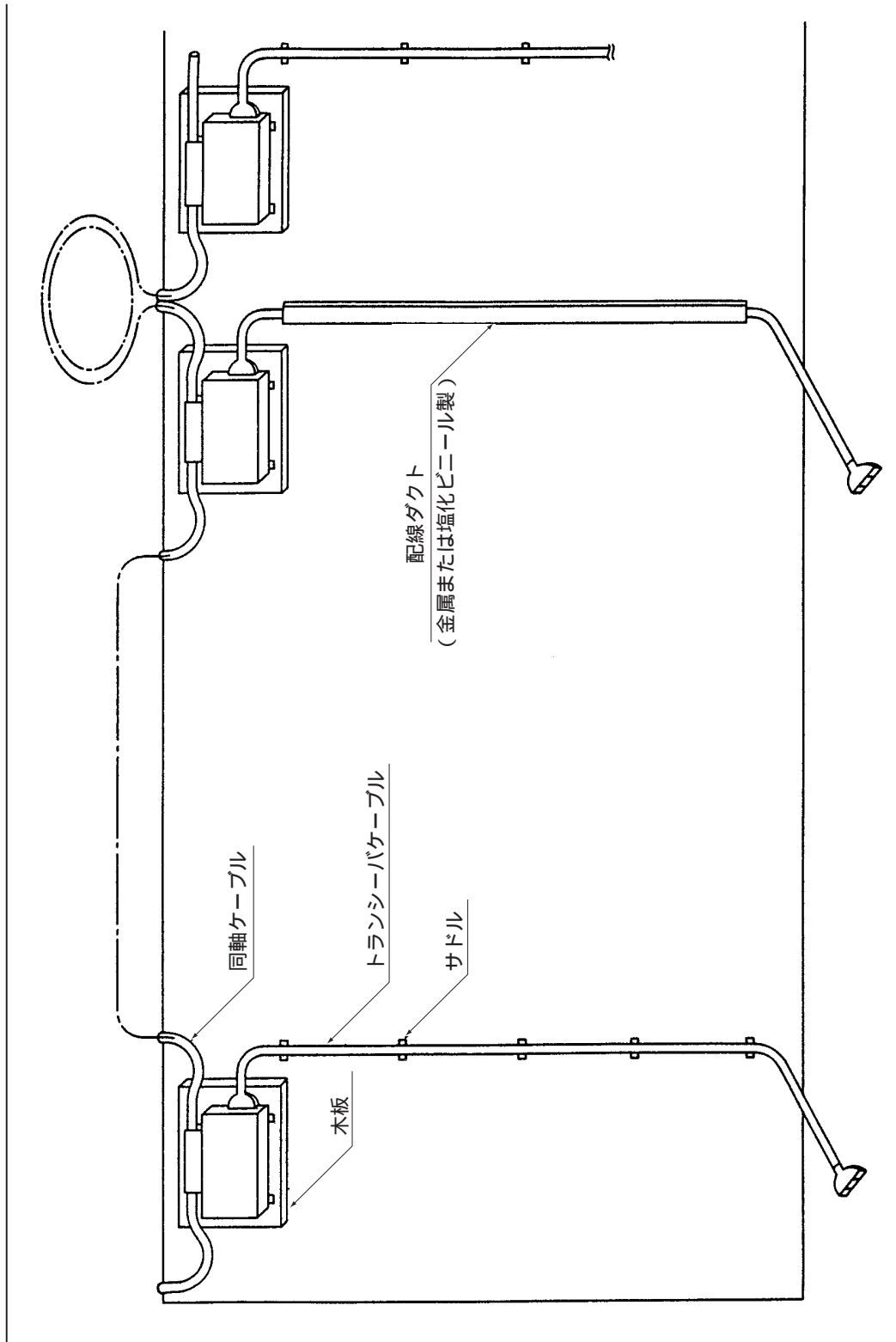


図6-7 壁面設置例 (1)

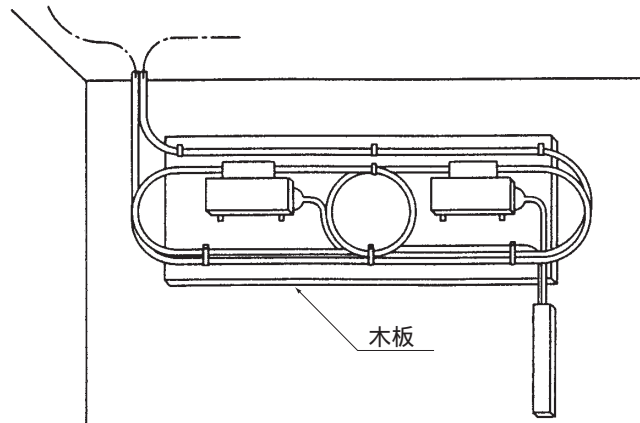


図 6 - 8 壁面設置例 (2)

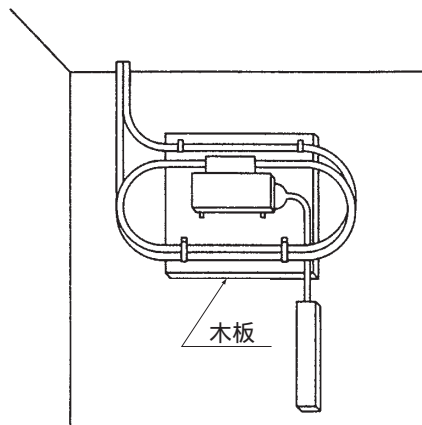


図 6 - 9 壁面設置例 (3)

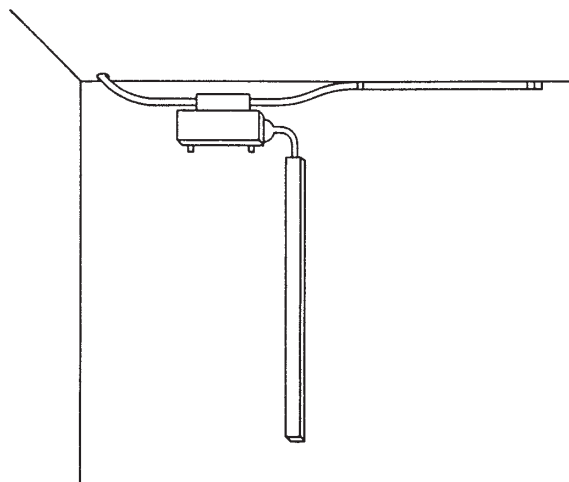


図 6 - 10 壁面設置例 (4)

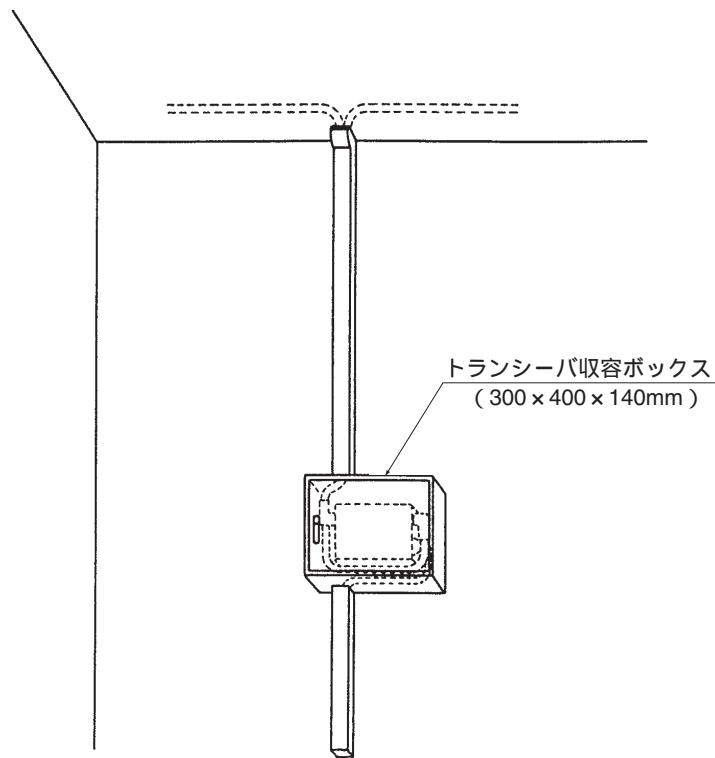


図6-11 ボックス内設置例(1)

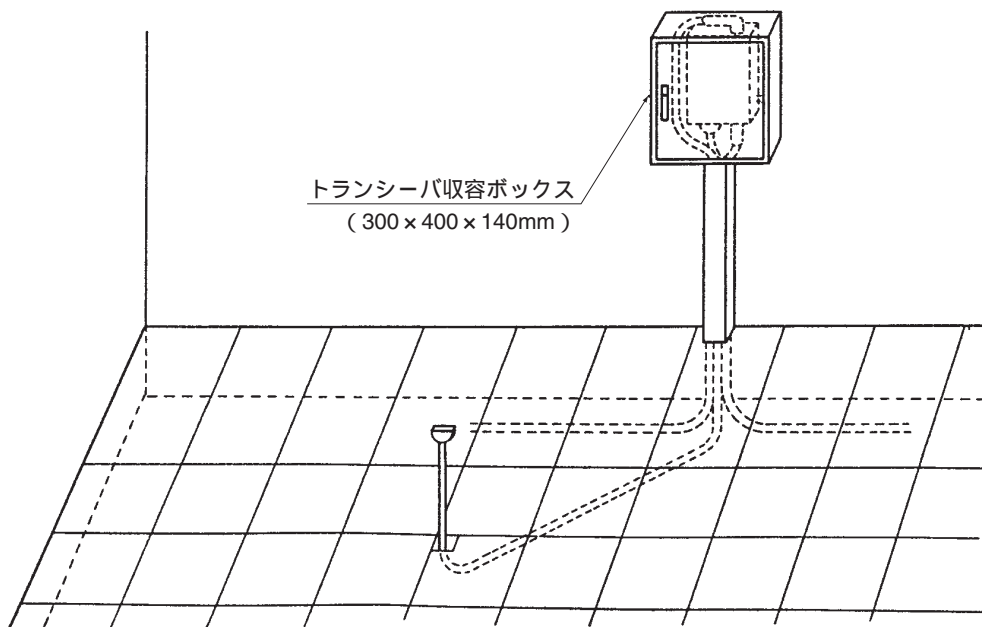


図6-12 ボックス内設置例(2)

6.4.3 同軸コネクタの取り付け

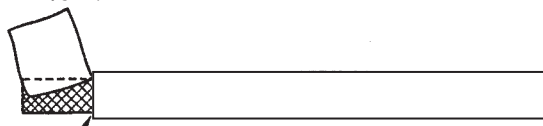
(1) 同軸コネクタの取り付け手順

同軸コネクタは、以下の手順で取り付けてください。

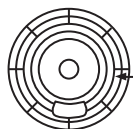
PVCシース剥ぎ



アルミテープ除去

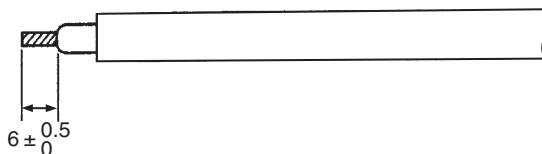


アルミテープは、この面できれいに除去してください。

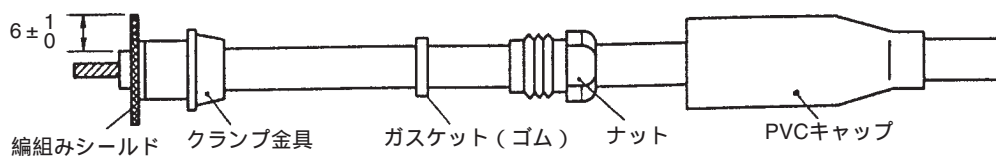


アルミテープは上図のとおりきれいに除去してください。

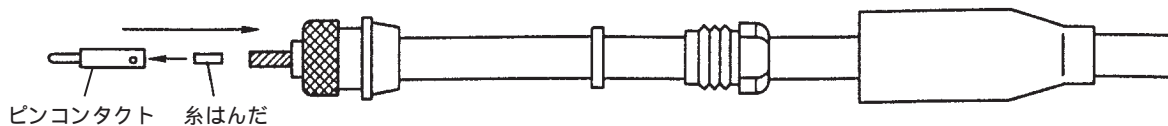
絶縁体剥ぎ



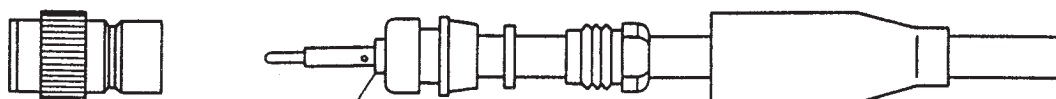
部品組み込みおよびシールド処理



ピンコンタクトはんだ付け



コネクタ取り付け



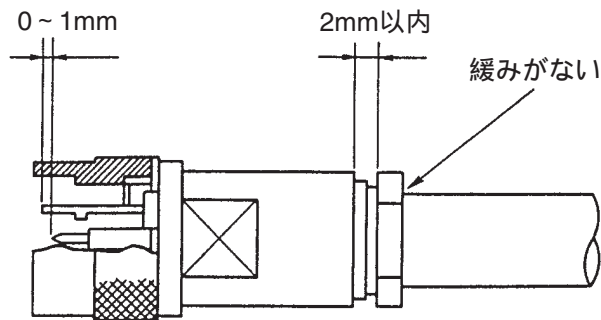
ピンコンタクトと絶縁体の間には、1mm以上の隙間ができないようにしてください。
また、ピンコンタクトが絶縁体内に食い込まないように注意してください。

6 利用の手引き

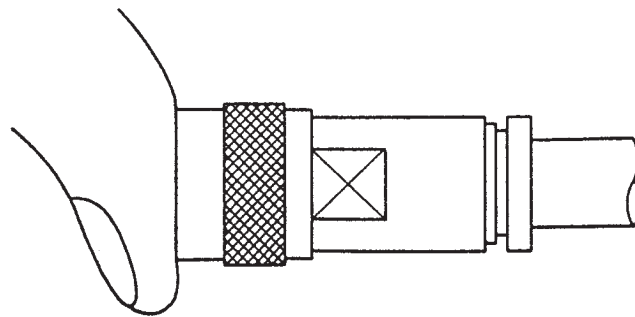
(2) 同軸コネクタ取り付け後のチェック内容

● 同軸コネクタ開口部の確認

- ・同軸コネクタ先端の外部導体とピンコンタクトの差は、0～1mm以内であること



- ・同軸コネクタ開口部に親指を当てると、ピンコンタクトの先端が親指の腹に軽く触れる程度であること



- ・目視により、ピンコンタクトに異常な偏心がないこと

● 緩みの確認

- ・同軸コネクタ取り付け後、同軸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
- ・締め付け後、締め付けナットと本体の隙間は約2mm以内であること

● 絶縁抵抗測定

絶縁抵抗は、ターミネータを外した状態で測定してください。

- ・トランシーバが付いていない場合

ピンコンタクトと外部導体の間が1000M Ω /km以上 (DC500V) であること

- ・トランシーバが付いている場合

外部導体を⊕極、ピンコンタクトを⊖極にしてテストで抵抗値を測定し、測定結果が無限大であること



試験後は必ず放電してください。放電しないと感電します。

6. 4. 4 タップコネクタの取り付け

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とは、以下の手順で接続してください。

- (1) 同軸ケーブル①を、タップコネクタ本体③の溝に挿入し、さらに上部からタップカバー②を取り付けることによって、同軸ケーブル①を固定します。
- (2) M6ボルト⑥を、ボックスドライバを使用して、規定トルク（3～4 [N・m]）で締め付けることにより、同軸ケーブル①の外部導体と接続させます。
- (3) バックアッププローブ⑤、信号用プローブ④の順に、ボックスドライバを使用して両側から同時に規定トルク（2～3 [N・m]）で締め付けることにより、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とバックアッププローブ⑤の先端とねじ山は壊れやすいので、取り扱いに十分注意してください。また、信号用プローブ④とバックアッププローブ⑤の取り付け後に、M6ボルト⑥の増し締めはしないでください。プローブに外力が加わり、破壊の原因になります。
- (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を取り付けます。

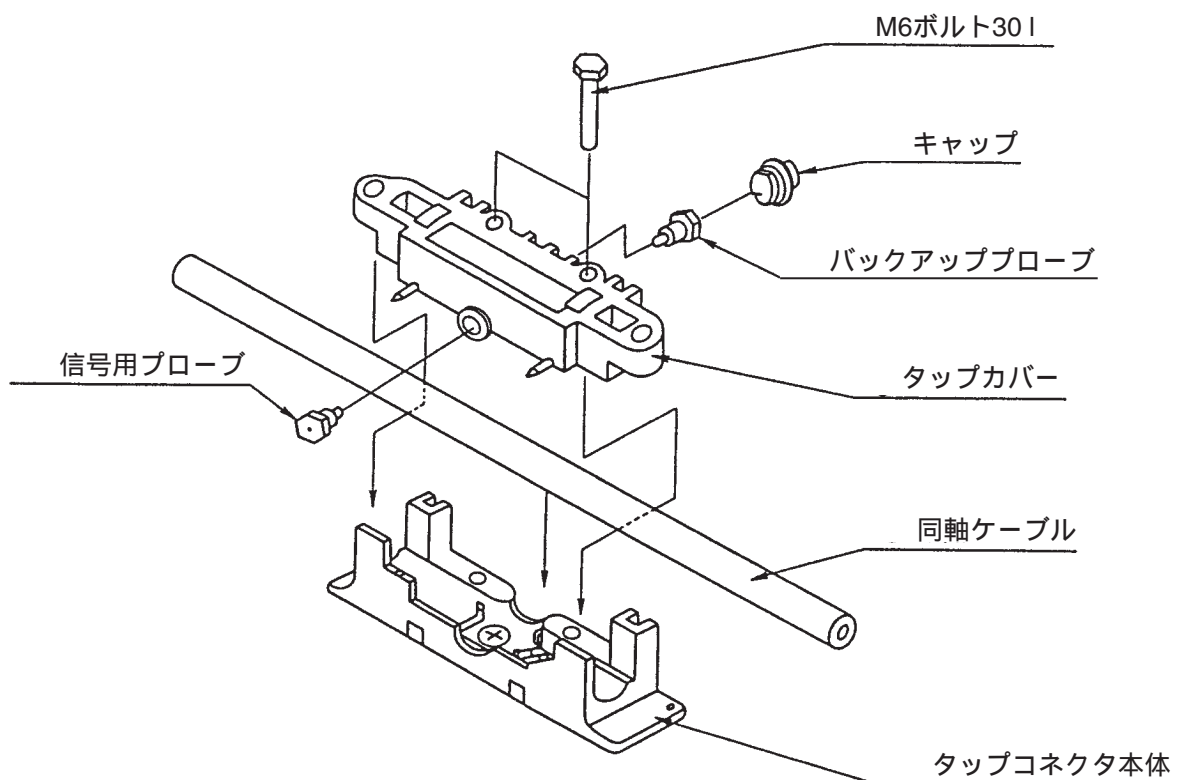


図6-13 タップコネクタ組み立て

6 利用の手引き

タップコネクタとトランシーバは、以下の手順で接続してください。

- (1) タップコネクタ①をトランシーバ②の側面に装着することによって、タップコネクタ①のプロープおよびグラウンド端子がトランシーバ②の取り付け穴に挿入されて、接続されます。
- (2) M6ボルト③を、ボックスドライバを使用して規定トルク (3~4 [N・m]) で締め付けることにより、トランシーバ②とタップコネクタ③を固定します。

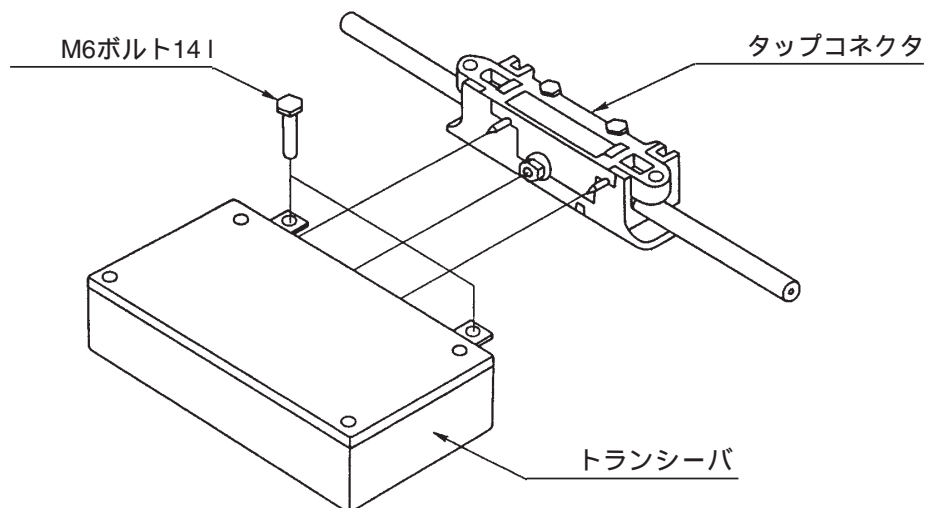


図6-14 コネクタとトランシーバの接続

6.4.5 トランシーバケーブルの取り付け

トランシーバケーブルには、トランシーバと接続するためのコネクタが取り付けられています。コネクタにロック用リテーナが付いていて、このリテーナが、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り付けてください。

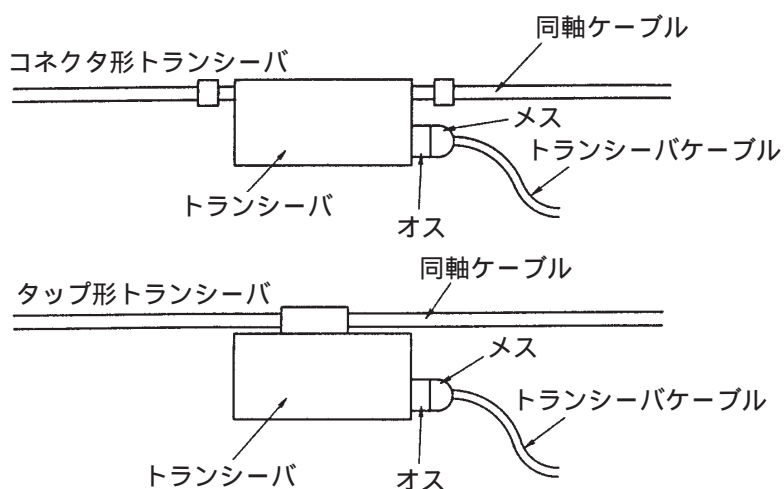


図6-15 トランシーバケーブルの取り付け

6. 4. 6 ターミネータの取り付け

ターミネータは、同軸セグメントの両端に必ず接続してください。

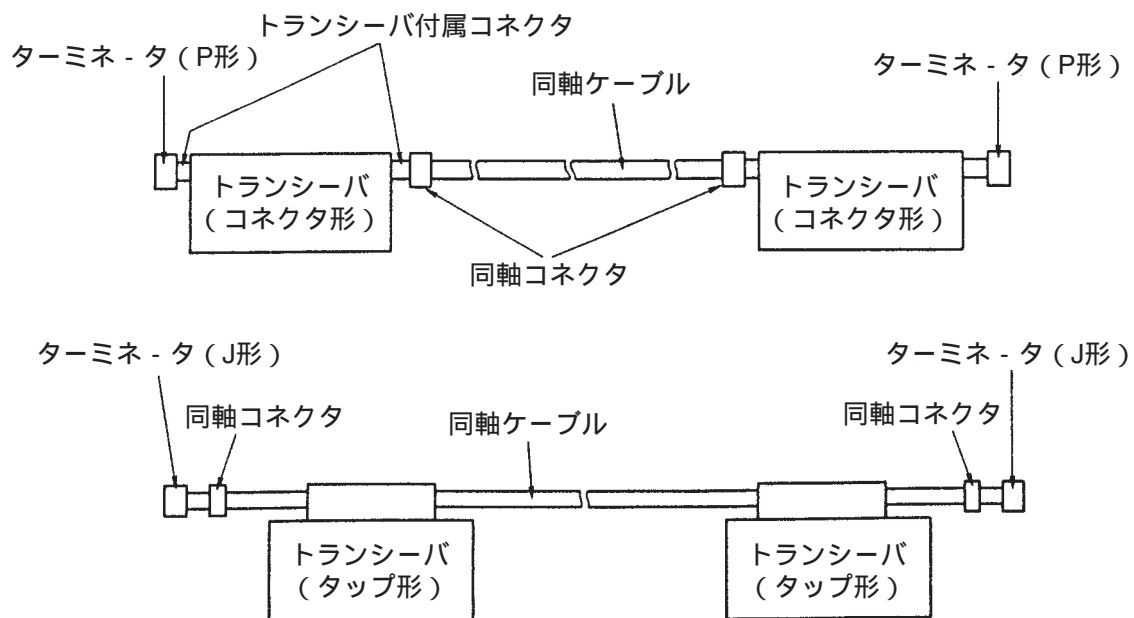


図6-16 ターミネータの取り付け

6. 4. 7 リピータの設置と取り付け

(1) 接続方法

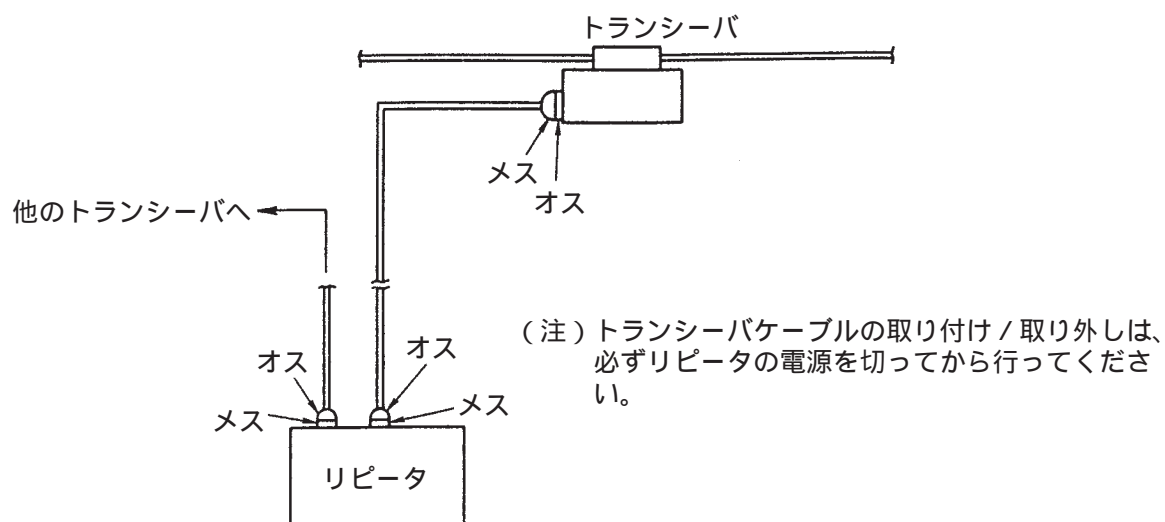


図6-17 リピータの取り付け

6 利用の手引き

(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、一般事務室内など容易に保守できる場所を選び、リピータの周囲に少なくとも図6-18に示すスペースを確保してください。
- ・リピータの電源ケーブルは、接地付きコンセントに接続してください。
- ・ちりやほこりの多いところでは使用しないでください。
- ・底面の空気取り入れ口、上面の吹き出し口は、ふさがないようにしてください。
- ・リピータの設置場所付近には、保守を考慮して電話を設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にしてください。リピータの電源が切れると、伝送機能が停止します。

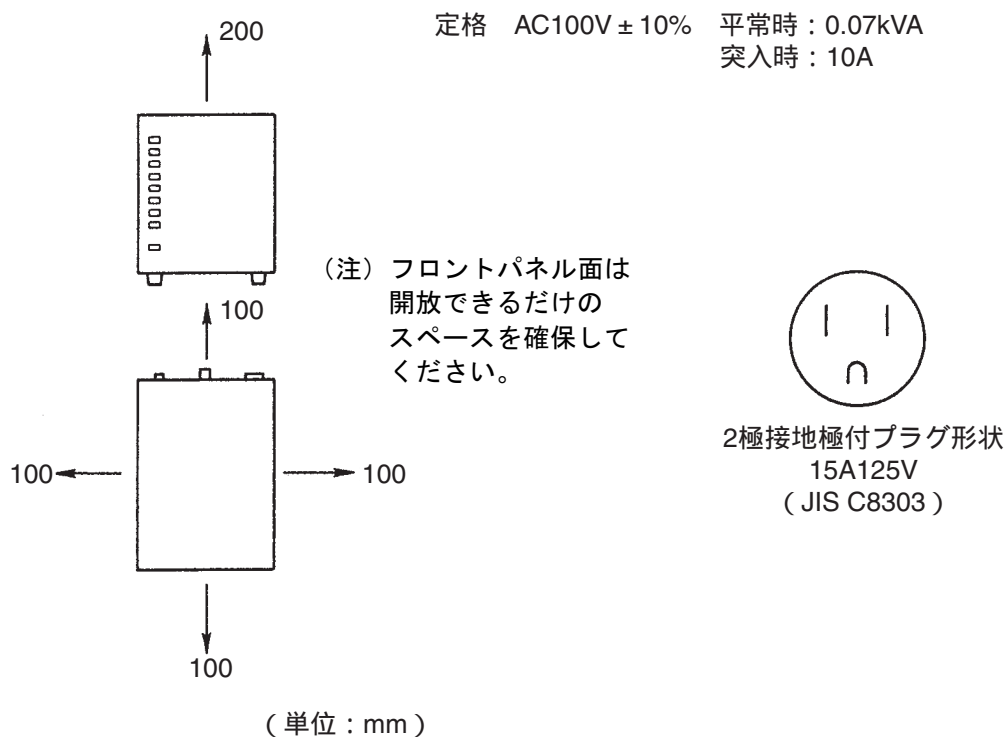


図6-18 リピータの設置

6.4.8 システムの接地

(1) リピータの接地

リピータは、接地型プラグ付き3線式電源コードを使用するか、または接地端子で接地してください。

(2) ステーションの接地

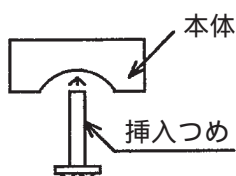
すべてのステーションは、D種接地以上の接地をしてください。

接地されていないステーションがある場合、感電の恐れがあります。また、データエラー（CRCエラー）の原因にもなります。

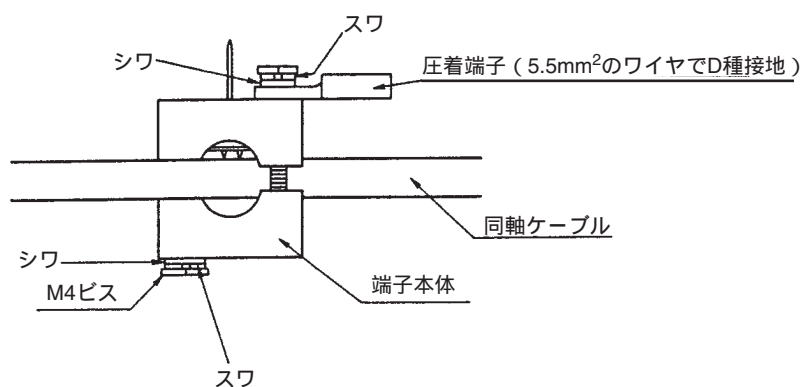
(3) 同軸ケーブルの接地

同軸ケーブルは、各セグメントごとに1点接地をしてください。接地は、D種接地以上の接地としてください。接地にはアース端子を使用してください。アース端子は以下の手順で取り付けてください。

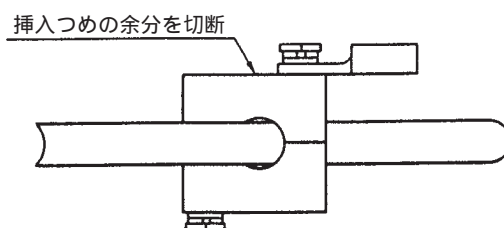
- ① 挿入つめを本体に挿入します。



- ② ①を同軸ケーブルに取り付けて、M4ビスを交互に締め付けます。この際、圧着端子をどちらかのビスに取り付けます。



- ③ M4ビスを締め付けた後、挿入つめの余分を切断します。



⚠ 危険

すべてのステーションは、D種接地以上の接地をしてください。
接地されていないステーションがある場合、感電の恐れがあります。

6 利用の手引き

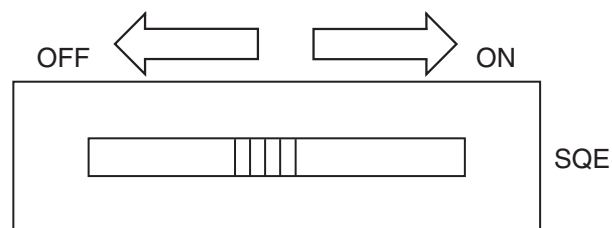
6.4.9 シングルポートトランシーバの設定

シングルポートトランシーバには、SQEスイッチがあります。SQEスイッチは、接続先により、表6-3に示す設定をしてください。

表6-3 SQEスイッチの設定

接続先	ET.NET コントローラ	マルチポート トランシーバ	リピータ
SQEスイッチ設定	ON	OFF	OFF

なお、シングルポートトランシーバHLT-200, HLT-200TBのSQEスイッチは、ケース内部にあります。設定を変更する際は、ケースを開いて設定してください。プリント基板上に「SQE」と表示してある側にスイッチを倒すと、ONになります。



6. 4. 10 マルチポートトランシーバの設定および表示

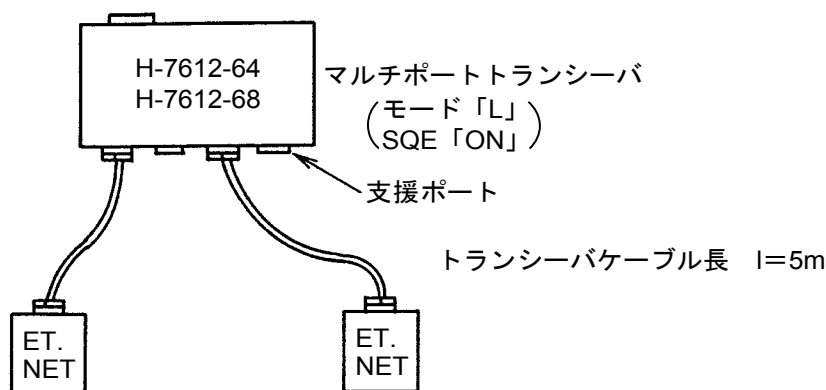
(1) 動作モードの設定

マルチポートトランシーバには、ローカルモードとネットワークモードの2種類の動作モードがあります。動作モードは、裏面パネル上の切り替えスイッチにより設定してください。

● ローカルモード

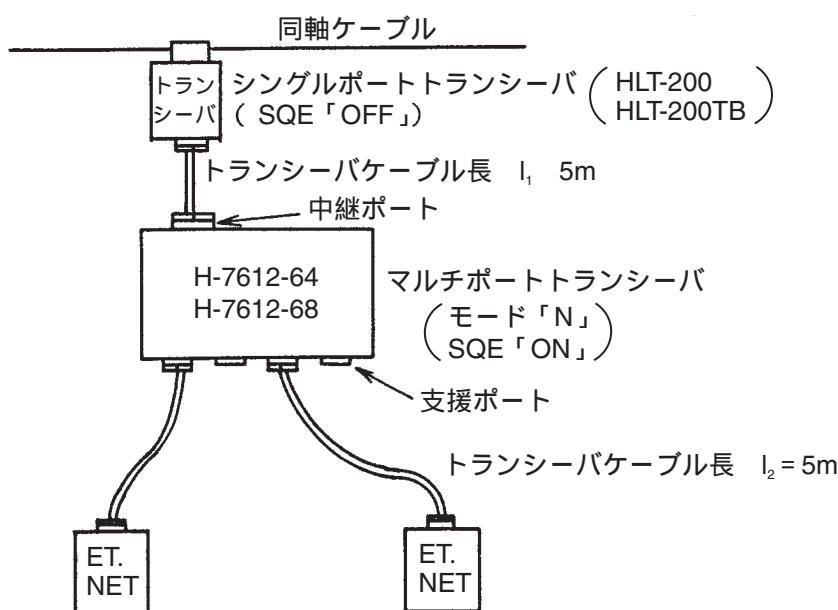
同軸ケーブルを接続せずに、単独で使用する動作モードです。中継ポートには、トランシーバケーブルを接続しないでください。

この動作モードでは、モード切り替えスイッチを「L」（ローカルモード）に、SQEスイッチは「ON」に設定します。



● ネットワークモード

同軸ケーブルと接続して使用する動作モードです。モード切り替えスイッチを「N」（ネットワークモード）に、SQEスイッチは「OFF」に設定します。



6 利用の手引き

(2) 切り替えスイッチの設定

マルチポートトランシーバには、2つの切り替えスイッチがあります。それぞれの機能を表6-4に示します。

表6-4 切り替えスイッチの設定

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切り替えスイッチ	裏面パネル	SQE機能のON/OFF	ON
動作モード切り替えスイッチ	裏面パネル	動作モードの切り替え	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

マルチポートトランシーバにリピータを接続する場合、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」側に設定すると、マルチポートトランシーバの電源が入ります。

(5) LEDの表示

正面パネルには、「POWER」LEDおよび各支線ポートごとに「LINK」LEDがあります。

「POWER」LED：マルチポートトランシーバの電源が入っている際に点灯します。

「LINK」LED：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際に点灯します。

6.5 システム定義情報

ET.NETには必ず下記②、③の情報を設定してください。また、ルータを経由して他ネットワークと接続する場合には④の情報も設定してください。②は他のステーションと重複して設定しないでください。③は同じサブネット内で同じ値としてください。

- ① 物理アドレス : ET.NET 1台ごとにユニークなアドレスが割り付けてあります。
- ② IPアドレス : ET.NET 1台ごとにET.NETシステムツールにより設定してください。
- ③ サブネットマスク : ET.NET 1台ごとにET.NETシステムツールにより設定してください。
- ④ 経路情報 : ルータを経由して他ネットワークと接続する際に設定してください。ET.NETシステムツールおよびプログラムによる設定ができます。

6.5.1 物理アドレス

ET.NET 1台ごとに、48ビットのアドレスが割り付けてあります。物理アドレスは、全世界に1つのユニークなアドレスです。ユーザは物理アドレスを変更できません。

6.5.2 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、以下の3とおりが使用できます。

- (1) 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)
--------------------	---------------

- (2) 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。)

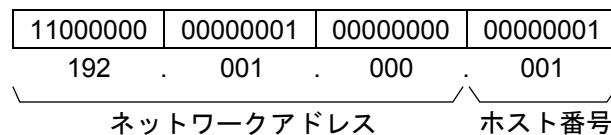
ネットワーク番号 (16ビット)	ホスト番号 (16ビット)
---------------------	---------------

- (3) 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)
---------------------	--------------

6 利用の手引き

また、このアドレスは8ビットごとに「.」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。



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、同じネットワーク内にホストが200台以内である場合には、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）を設定し、ネットワークには5台のホストを接続する場合、各ステーションのIPアドレスは以下のように設定します。

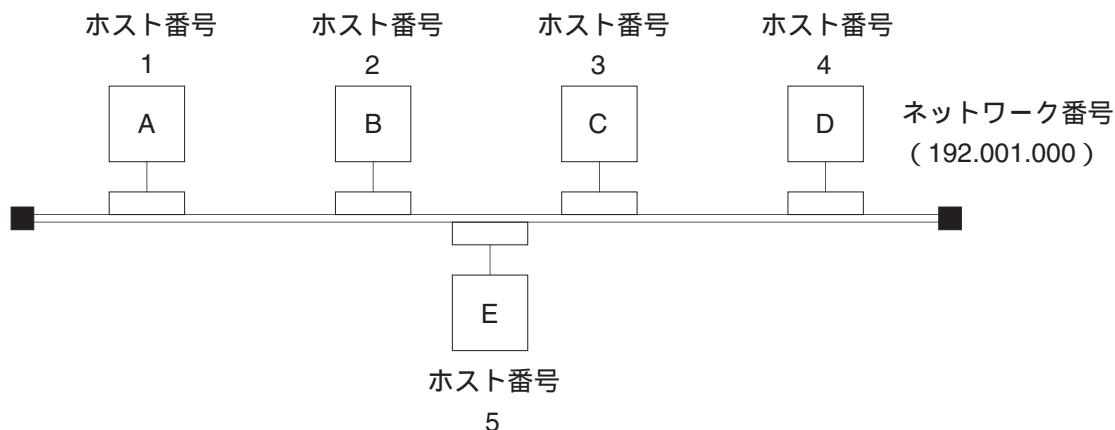
ステーションA : 192.001.000.001

ステーションB : 192.001.000.002

ステーションC : 192.001.000.003

ステーションD : 192.001.000.004

ステーションE : 192.001.000.005



IPアドレスには、特別なアドレスが2つあります。1つは、ホスト番号のビットをすべて「0」とし、ネットワーク全体を表すアドレスです。もう一方は、ホスト番号のビットをすべて「1」とするブロードキャストアドレスです。ブロードキャストアドレスは、そのネットワークに属するすべてのステーションに対してデータを送信する際に使用します（UDP/IP通信により行ってください）。

6. 5. 3 サブネットマスク

IPアドレスをサブネットに分割する場合、サブネットワーク番号とローカルホスト番号の境界を、サブネットマスクによって定義します。サブネットマスクをデフォルト値以外で使用する場合、以下の例のようなブロードキャストアドレスになることを前提にサブネットマスクを使用してください。

(例) クラスBの場合

IPアドレス	サブネットマスク	ブロードキャストアドレス
128.123.000.001	255.255.000.000	128.123.255.255
128.123.001.001	255.255.255.000	128.123.001.255

6. 5. 4 経路情報

ルータを経由して他ネットワークに接続する場合、経路情報を設定してください。

経路情報は、通信相手のIPアドレスとルータのIPアドレスをペアで登録します。

(1) 通信相手IPアドレス

通信相手ごとにIPアドレスを登録します。同じネットワークの通信相手が複数ある場合は、統合してネットワークアドレスを設定できます。ネットワークアドレスは、IPアドレスのホスト番号をすべて「0」としたものです。

(2) ルータのIPアドレス

ET.NETモジュールと同じネットワーク側のルータのIPアドレスを登録します。通信相手との通信経路に複数段ルータがある場合にも、ET.NETと同じネットワーク側のルータのみ登録してください。

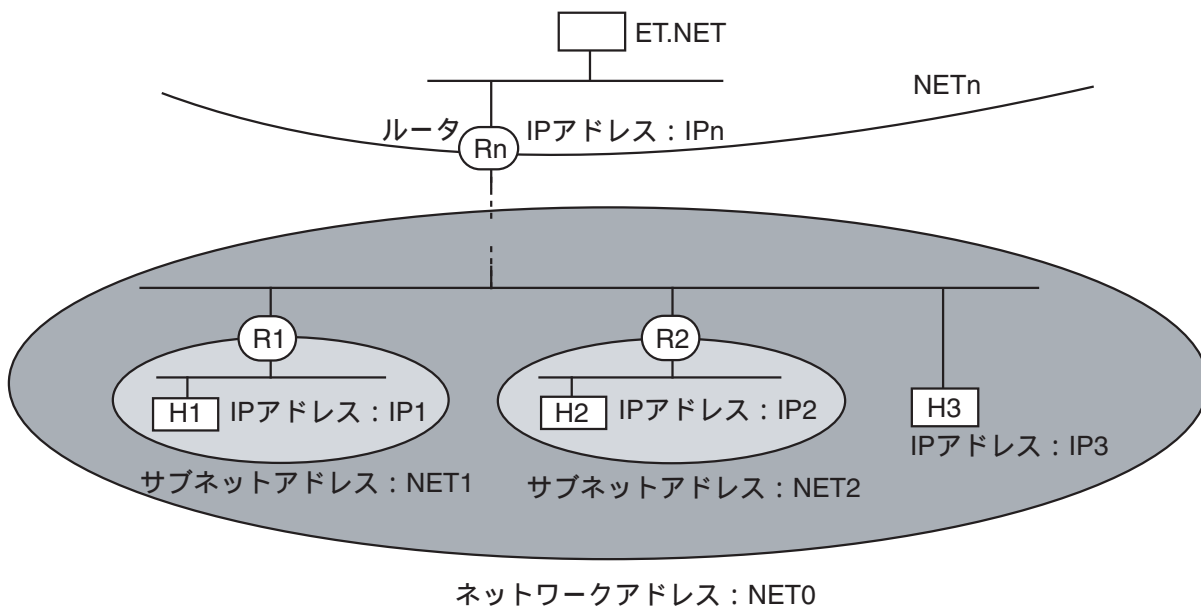
経路情報の設定方法には、次の2つがあります。

- C言語プログラムでソケットハンドラroute_add()にて設定する方法
「5. 4. 1 ソケットハンドラ一覧」を参照してください。
- ET.NETシステムまたはS10V ET.NETシステムを使用して設定する方法
「4. 3. 3 経路情報設定」を参照してください。

6 利用の手引き

経路情報の登録例を示します。

- ホストH1と通信する場合に登録する経路情報
 - ・ ルータRnのIPアドレス IPn
 - ・ ホストH1のIPアドレス IP1
- ホストH3と通信する場合に登録する経路情報
 - ・ ルータRnのIPアドレス IPn
 - ・ ホストH3のIPアドレス IP3、またはネットワークアドレス NET0



制約事項

- ・ 経路情報として登録できるのは、`route_add()`とツール設定を併せて15個までです。
- ・ `route_add()`とツールで同じ設定をした場合、ツール設定が優先され、`route_add()`の設定は無効となり、リターン値にはエラーコードが返ります。
- ・ 登録できるアドレスはIPアドレスとネットワークアドレスで、サブネットアドレスは登録できません。これは、ET.NETモジュールが、経路情報をIPアドレスまたはネットワークアドレスとして認識し、サブネットアドレスとして認識しないためです。仮にサブネットアドレスを登録したとしてもIPアドレスとして認識するため、通信できません。

6. 6 ET.NETモジュールのメモリマップ

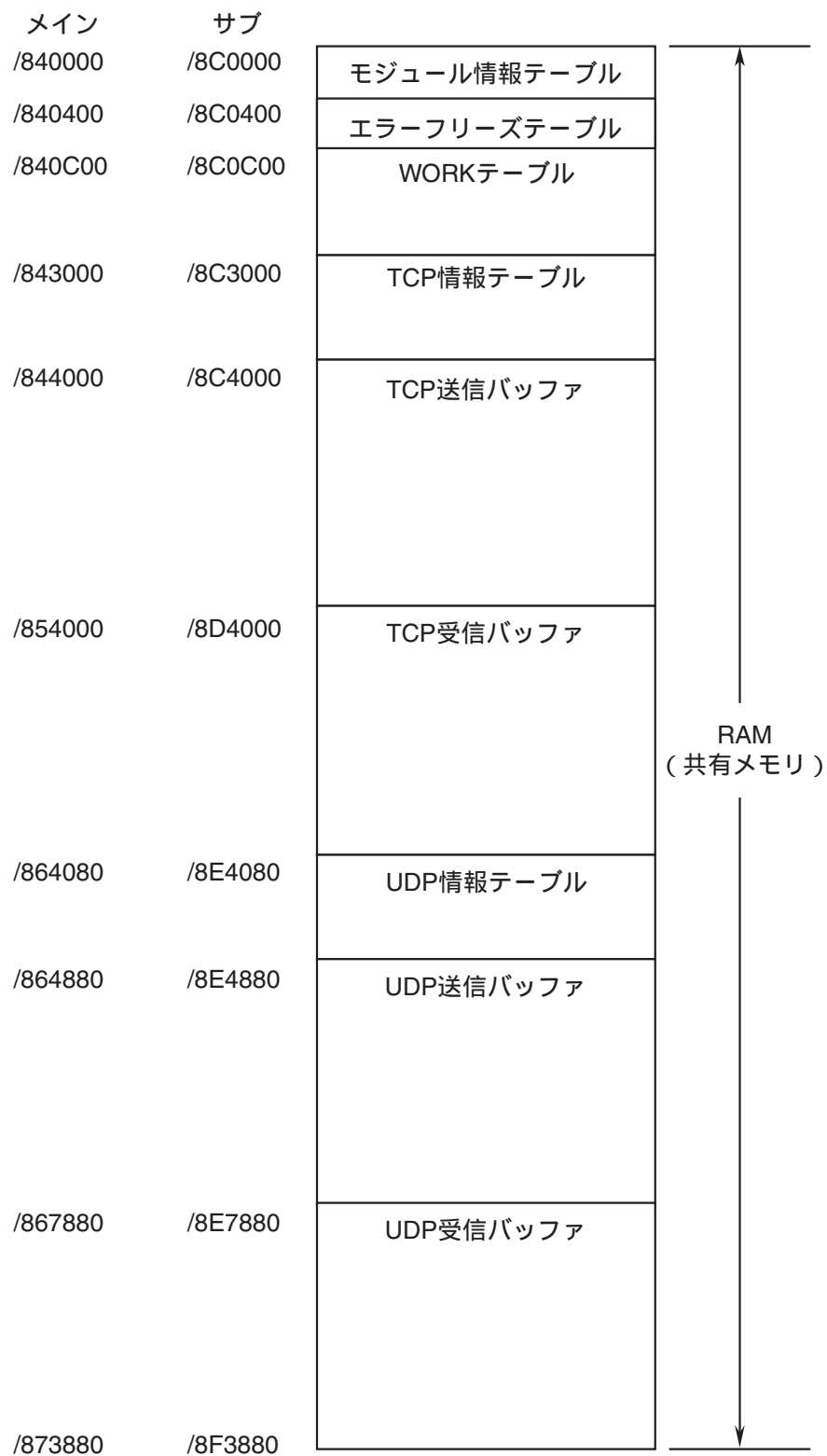


図 6-19 メモリマップ

このページは白紙です。

7 保 守

7. 1 保守点検

モジュールを最適な状態で使用するには、以下のような点検をしてください。点検は、日常または定期的(2回/年以上)に行ってください。

表 7-1 保守点検項目

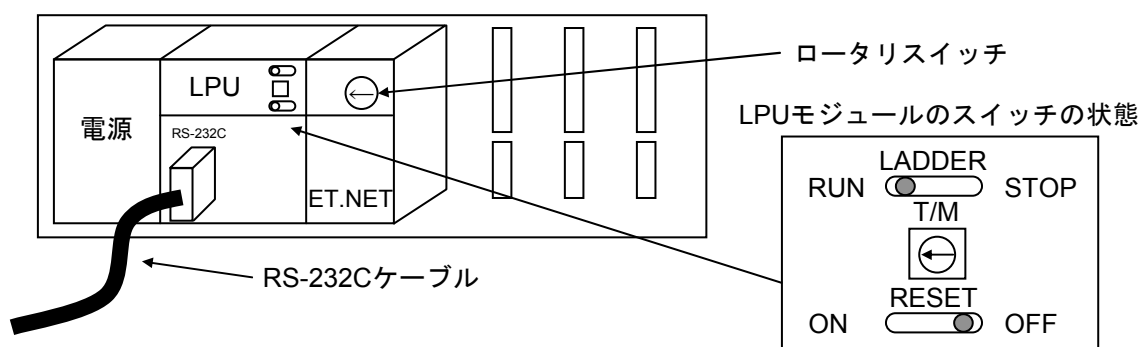
項 目	チェックポイント
モジュールの外観	モジュールケースにひび、割れなどが点検してください。ケース類に異常があると内部回路に破損が生じている場合があります、システム誤動作の原因になります。
LED	モジュールのERR LEDが点灯していないか点検してください。
取り付けねじの緩み	モジュールおよび通信ケーブルの取り付けねじなどに緩みがないか点検してください。緩みがある場合には、増し締めをしてください。ねじに緩みがあるとシステムの誤動作、さらには加熱による焼損の原因になります。
ケーブルの被覆の状態	ケーブルの被覆に異常がないか点検してください。被覆が剥がれているとシステムの誤動作、感電、さらにはショートによる焼損の原因になります。
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、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、焼損の原因になります。
モジュールの交換	通電時のモジュール交換は、ハードウェア、ソフトウェアの破壊につながりますので、必ず電源を切ってから交換してください。
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になります。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。作業する前に、人体の静電気を放電してください。
- モジュールは、必ず電源を切ってから交換してください。感電、モジュールの破損および誤動作の恐れがあります。

7. 1. 1 モジュールの交換、増設

- 交換前準備品
 - ① パソコン (Hitachi S10V ET.NETシステムツール組み込み済み)
 - ② RS-232Cケーブル
 - ③ ET.NETモジュール (LQE520)
 - 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。)
- 交換手順
 - ① 実装されているET.NET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 (MODU No.)。
 - ② LPUモジュール前面のスイッチの状態を記録します (LADDER, T/M)。



- ③ パソコンとLPUモジュールをRS-232Cケーブルで接続します。
- ④ Hitachi S10V ET.NETシステムツールを立ち上げ、設定されているIPアドレスを記録します (読み出せない場合は、交換前準備品の④を使用してください)。
- ⑤ LPU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 ET.NETモジュールに接続されているケーブルを外します。
- 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 S10V ET.NETシステムツールからIPアドレスを設定します。
- ⑨ 記録したIPアドレスと一致しているかを確認してください。
- ⑩ LPUモジュール前面のRESETスイッチを一度ONからOFFにし、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ケーブルを外します。
- ⑬ ⑥で外したケーブルを元に戻します。
- ⑭ LPUモジュールのスイッチを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 LPU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、LPU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 「3.3 モジュールの実装」を参照のうえ、ET.NETモジュールを実装します。
- ④ メイン側のモジュールと重複しないようにロータリスイッチをサブ側のNo.に設定してください。
- ⑤ パソコンとLPU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 S10V ET.NETシステムツールから増設したET.NETモジュールにIPアドレスを設定します。
- ⑥ ユニットの電源をOFFにし、増設したET.NETモジュールにケーブルを接続します。
- ⑦ LPU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。
- ⑧ ⑤で接続したRS-232Cケーブルを外します。
- 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。

7. 2 トラブルシューティング

7. 2. 1 手 順

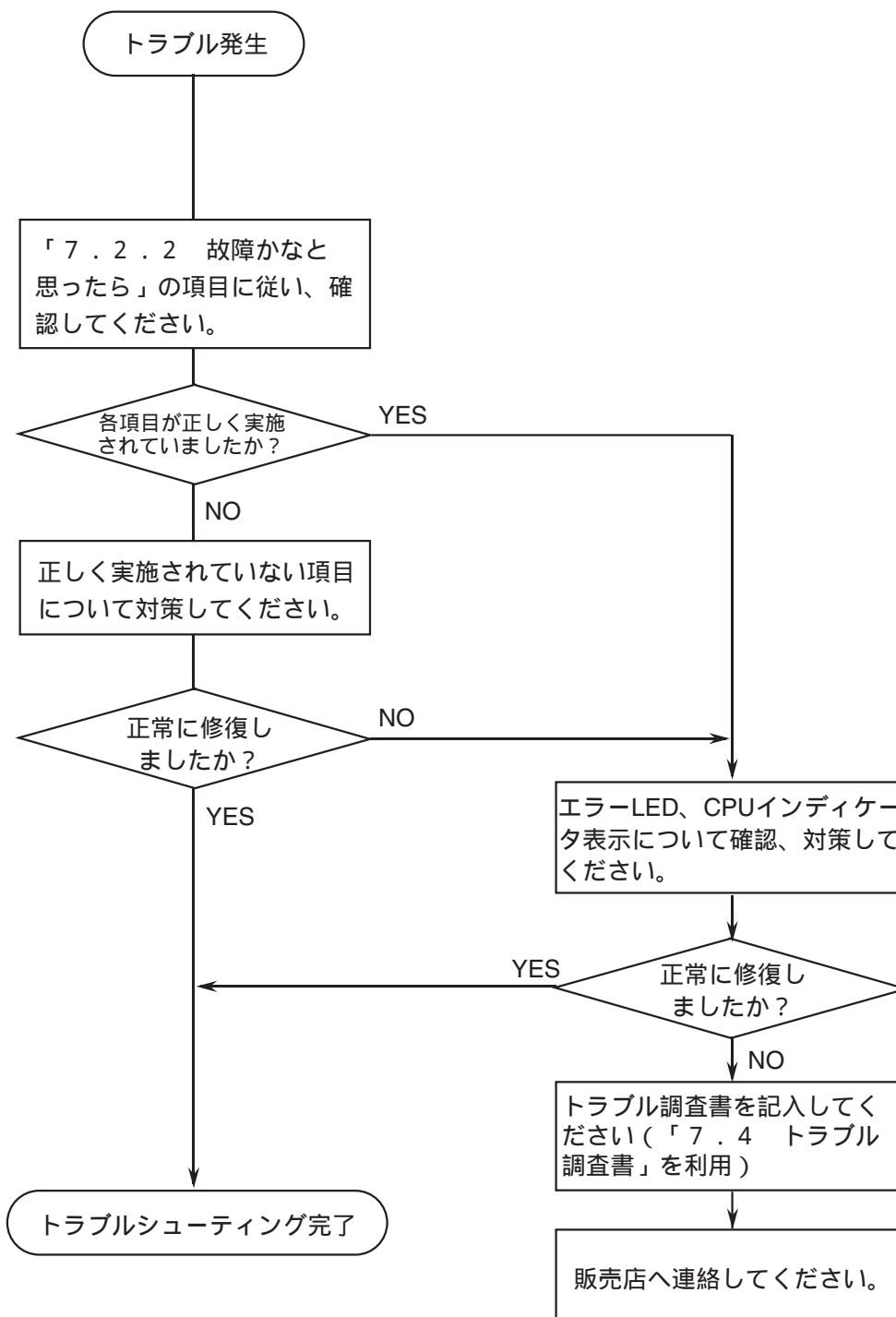
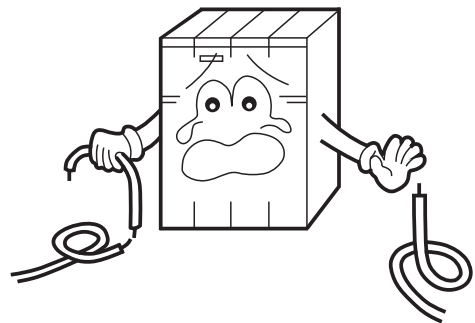


図7-1 トラブルシューティング手順

7.2.2 故障かなと思ったら

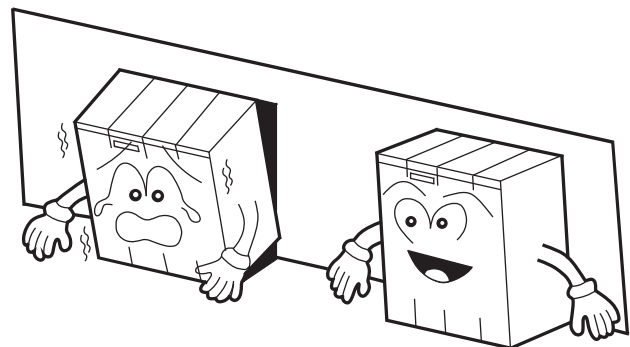
(1) 正しく配線されていますか？

- ・ケーブルの断線、接続誤りがないか調べてください。
- ・トランシーバケーブルは、シールドアース線付きのケーブルを使用しているか調べてください。



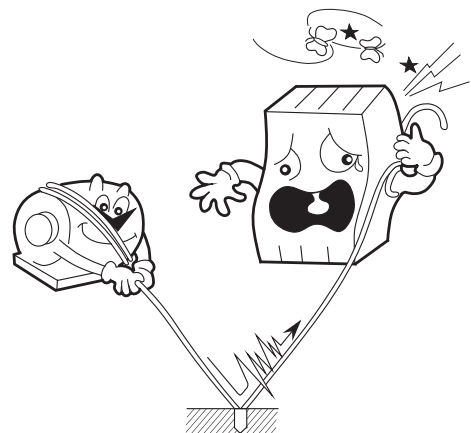
(2) モジュールは正しく実装されていますか？

- ・S10mini CPUマウントベースに実装している場合、ET.NETモジュールは、実装してあるモジュールとの間に空きスロットがないように左詰めで実装されているか調べてください。
- ・取り付けねじに緩みがないか調べてください。



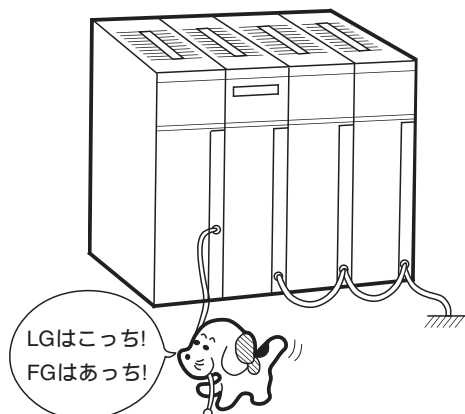
(3) 正しく接地されていますか？

- ・強電機器と同一地点での接地は避け、分離してください。
- ・D種接地以上の接地工事をしてください。



(4) LGとFGは分離されていますか？

- ・電源からのノイズが、LGを介してFGへ入り込み、誤動作の原因になるため、必ず分離してください。
- ・LGは電源供給側で接地してください。



7.3 エラーと対策

7.3.1 インディケータ表示メッセージ

S10miniでは、ET.NETモジュールにイベントやエラーが発生した場合、CPUモジュールのインディケータに、表7-2に示すメッセージを表示します。表示内容は、ET.NETモジュールのメイン設定、サブ設定で区別します。

S10Vでは、エラー情報を収集しますが、LPUモジュールへのエラー表示はしません。収集したエラー情報は、S10V基本システムから参照できます。詳細は、「7.3.2 エラー対処方法」を参照してください。

表7-2 S10mini CPUモジュール表示メッセージ

モジュール	表 示	内容および説明	対 策
メイン	ETM @@	ET.NETモジュールが正常に動作しています。	エラーではありませんので、対策は不要です。
	ETM □□□□	ET.NETモジュールにエラーが発生しています。	「7.3.2 エラー対処方法」を参照してください。
	EXD2 PTY	CPUがET.NETモジュール内部のメモリを読み込んだ際、パリティエラーが発生しました。	CPUのリセットスイッチを用いてリセットしてください。表示が消えない場合は、ET.NETモジュールを交換してください。
サブ	ETM @@	ET.NETモジュールが正常に動作しています。	エラーではありませんので、対策は不要です。
	ETM □□□□	ET.NETモジュールにエラーが発生しています。	「7.3.2 エラー対処方法」を参照してください。
	EXD2 PTY	CPUがET.NETモジュール内部のメモリを読み込んだ際、パリティエラーが発生しました。	CPUのリセットスイッチを用いてリセットしてください。表示が消えない場合は、ET.NETモジュールを交換してください。

- @@は、ET.NETモジュールのバージョンおよびレビジョンを表します。
- □□□□は、「7.3.2 エラー対処方法」のエラー表示メッセージを表します。

7 保 守

7. 3. 2 エラー対処方法

ET.NETモジュールがエラーを検出した場合、S10miniではCPUインディケータに表7-3の「CPU表示」の内容を表示し、S10VではS10V基本システムからエラーログを選択することで表7-3の「エラーコード」を表示します。また、ET.NET上のERR LEDを点灯するとともに、エラーフリーズ情報を収集します。エラーフリーズ情報の詳細は、図7-2に示します。なお、ET.NETモジュールの動作は停止します。

S10Vにおけるツールシステムの起動方法、およびエラーログ情報の表示方法は、「S10V ユーザーズマニュアル 基本モジュール（マニュアル番号 SVJ-1-100）」を参照してください。

表7-3 エラーメッセージ内容

CPU表示 (S10mini)	エラーコード (S10V)	内 容	対 策
BUS	/0010	バスエラー	ET.NETモジュールが故障しています。 モジュールを交換してください。
ADDR	/0011	アドレスエラー	
ILLG	/0012	不当命令	
ZERO	/0013	0除算	
PRIV	/0014	特権違反	
FMAT	/0016	フォーマットエラー	
SINT	/0017	スプリアス割り込み	
EXCP	/0018	未使用例外	
PTY	/0019	パリティエラー	
ROM1	/0102	ROM1チェックサムエラー	
RAM1	/0103	RAM1コンペアエラー	
RAM2	/0105	RAM2コンペアエラー	
ROM3	/010B	ROM3チェックサムエラー	
MAC	/0114	物理アドレス未登録	
PRG	/0112	マイクロプログラムエラー	
MDSW	/0100	モジュールNo.設定スイッチの設定誤り	モジュールNo.設定スイッチに誤りがあります。「2. 1 各部の名称と機能」を参照し、モジュールNo.設定スイッチの設定内容を修正してください。
IPNG	/0113	IPアドレス未登録	IPアドレスを登録してください。
R_NG	/0200	経路情報設定エラー	経路情報設定に誤りがあります。「7. 3. 4 経路情報設定エラーテーブル」を参照し、経路情報設定を修正してください。

メイン	サブ	エラーコード	No.	エラーコード	内 容
/840400	/8C0400	エラーコード	1	/0010	バスエラー
/840404	/8C0404	——	2	/0011	アドレスエラー
/840410	/8C0410	D0レジスタ	3	/0012	不当命令
/840414	/8C0414	D1レジスタ	4	/0013	0除算
/840418	/8C0418	D2レジスタ	5	/0014	特権違反
/84041C	/8C041C	D3レジスタ	6	/0016	フォーマットエラー
/840420	/8C0420	D4レジスタ	7	/0017	スプリアス割り込み
/840424	/8C0424	D5レジスタ	8	/0018	未使用例外 (CHK, TRAPV, L1010など)
/840428	/8C0428	D6レジスタ	9	/0019	パリティエラー
/84042C	/8C042C	D7レジスタ	10	/001A	停電予告
/840430	/8C0430	A0レジスタ	11	/0100	モジュールNo.設定スイッチの設定誤り
/840434	/8C0434	A1レジスタ	12	/0102	ROM1チェックサムエラー
/840438	/8C0438	A2レジスタ	13	/0103	RAM1コンペアエラー
/84043C	/8C043C	A3レジスタ	14	/0105	RAM2コンペアエラー
/840440	/8C0440	A4レジスタ	15	/010B	ROM3チェックサムエラー
/840444	/8C0444	A5レジスタ	16	/0112	マイクロプログラムエラー
/840448	/8C0448	A6レジスタ	17	/0113	IPアドレス未登録
/84044C	/8C044C	A7レジスタ	18	/0114	MACアドレスエラー
/840450	/8C0450	スタックフレーム (図7-3参照)	19	/0200	経路情報設定エラー
/8404FC	/8C04FC				

図7-2 エラーフリーズ情報

メイン	サブ	ステータスレジスタ	R/W	I/N	FC
/840450	/8C0450	ステータスレジスタ			
/840452	/8C0452	プログラム			
/840454	/8C0454	カウンタ			
/840456	/8C0456	命令レジスタ			
/840458	/8C0458	ステータスレジスタ			
/84045A	/8C045A	プログラム			
/84045C	/8C045C	カウンタ			

R/W : 書き込み=0、読み出し=1
 I/N : 命令=0、非命令=1
 FC : ファンクションコード

図7-3 スタックフレームの内容

7 保 守

7.3.3 ソケットハンドラ検出のエラーコード

ソケットハンドラのエラーコードと対策について、以下に示します。

表7-4 ソケットハンドラ検出のエラーコード

(1/3)

エラーコード	内 容	原 因	対 策
F000	コネクション未接続	ハンドラ起動時、未接続またはポートが開放されました。	tcp_open()またはtcp_popen()を発行し、コネクション確立後にハンドラを再発行してください。
F002	FIN受信	ハンドラ起動時、FINを受信しました。	tcp_close()を発行してコネクション切断後、tcp_open()またはtcp_popen()から再度コネクション確立をしてください。
F010	ソケットID不正	<ul style="list-style-type: none"> ソケットIDが範囲外 (TCP : $1 \leq ID \leq 15$、 UDP : $/20 \leq ID \leq /27$) 使用していないソケットID、または開放済みのソケットIDを指定しました。 未接続または接続が確立されていません (tcp_accept()のみ)。 	ユーザプログラムを見直してください。 (tcp_open()またはtcp_popen()のリターン値をソケットIDとしているかなど)
F011	ソケット数オーバ	ソケットを制限数以上登録しています (TCP : 12個、UDP : 8個)。	未使用ソケットをクローズ後 (tcp_close()またはudp_close()発行)、tcp_open()またはtcp_popen()から再度コネクション確立をしてください。
F012	ソケットドライバタイムアウト	<ul style="list-style-type: none"> 一定時間経過してもソケットドライバから応答がありません。 送信ウィンドウ満杯等により送信がタイムアウトしました (tcp_send()のみ)。 	<p>tcp_close()を発行してコネクション切断後、tcp_open()またはtcp_popen()から再度コネクション確立をしてください。</p> <p>再度コネクション確立を繰り返しても通信できない場合は、コネクタ、ケーブル、または相手局に異常がないか確認してください。</p> <p>tcp_close()にて発生した場合は、tcp_abort()を発行してコネクション切断後、tcp_open()またはtcp_popen()から再度コネクション確立をしてください。</p>

(2/3)

エラーコード	内 容	原 因	対 策
F013	モジュール停止	ハンドラ起動時、100秒経過してもソケットドライバの初期化が終了できません。	アプリケーションの許容範囲内で、tcp_close()を発行後、tcp_open()またはtcp_popen()から再度コネクション確立をしてください。
F020	送信データ長不正	送信データ長が制限値を満たしていません。 (TCP : 1 ≤ データ長 ≤ 4,096、 UDP : 1 ≤ データ長 ≤ 1,472)	ユーザプログラムを見直してください。
F021	受信データ長不正	受信データ長が制限値を満たしていません。 (1 ≤ データ長 ≤ 4,096)	ユーザプログラムを見直してください。
F0FF	ポート開放	<ul style="list-style-type: none"> ハンドラ起動後、ポート開放状態 (RST受信) になりました (tcp_open())。 ハンドラ起動時、ポート開放状態でした (tcp_send()またはtcp_receive())。 	<ul style="list-style-type: none"> tcp_open()またはtcp_popen()から再度コネクション確立をしてください。 tcp_close()を発行後、tcp_open()またはtcp_popen()から再度コネクション確立をしてください。
FFF0	アドレス不正	<ul style="list-style-type: none"> udp_open()、udp_send()ともに相手局のIPアドレス、ポート番号に0を設定しています。 udp_send()でイーサネットレベルのエラー (コリジョンなど) が発生しました。 	<ul style="list-style-type: none"> ユーザプログラムを見直してください。 トラフィックが下がった時点で再度udp_send()を発行してください。
FFF3	引数不正	不正なパラメータを指定しました。	ユーザプログラムを見直してください。
FFF5	接続タイムアウト	相手局からの応答がありません。	tcp_close()を発行してコネクション切断後、tcp_open()またはtcp_popen()から再度コネクション確立をしてください。再度コネクション確立を繰り返しても通信できない場合は、コネクタ、ケーブル、または相手局に異常がないか確認してください。
FFF6	クローズ済み	コネクションが終了した (closeまたはabortされた) ソケットIDに対し、コマンドを発行しました。	tcp_open()またはtcp_popen()から再度コネクション確立をしてください。

エラーコード	内 容	原 因	対 策
FFF8	FIN受信	相手局からFINを受信しました。	tcp_close()を発行し、ソケットをクローズしてください。
FFFA	コネクション強制終了	相手局から強制終了（RST受信）されました（RST受信後にtcp_receive()を発行しました）。	tcp_close()を発行してコネクション切断後、tcp_open()またはtcp_popen()から再度コネクション確立をしてください。
FFFC	ネットハンドル不正	TCPまたはUDPで、オープンしていないハンドル番号を使用して送受信しました。RST受信で発生する可能性があります（tcp_receive()で受信待ちの際にRSTを受信しました）。	tcp_close()を発行してコネクション切断後、tcp_open()またはtcp_popen()から再度コネクション確立をしてください。
FFFD	二重ソケットエラー	同じソケット（相手局のIPアドレス、相手局ポート番号、自局ポート番号）がすでに存在しています。	ユーザプログラムを見直してください。
FFFE	コントロールブロック不正	制限を超えてソケットを使用しています。	未使用ソケットをクローズ後（tcp_close()またはudp_close()発行）、tcp_open()またはtcp_popen()から再度コネクション確立をしてください。

7. 3. 4 経路情報設定エラーテーブル

経路情報の設定エラー時、下記テーブルにエラーコードが設定されます。

メイン	サブ	2^{31} _____ 2^0		
/873880	/8F3880	デフォルト	+0	エラーコード
/873884	/8F3884	ユーザ (1)	+2	重複ユーザ番号
/873888	/8F3888	ユーザ (2)		
/87388C	/8F388C	ユーザ (3)		
/873890	/8F3890	ユーザ (4)		
/873894	/8F3894	ユーザ (5)		
/873898	/8F3898	ユーザ (6)		
/87389C	/8F389C	ユーザ (7)		
/8738A0	/8F38A0	ユーザ (8)		
/8738A4	/8F38A4	ユーザ (9)		
/8738A8	/8F38A8	ユーザ (10)		
/8738AC	/8F38AC	ユーザ (11)		
/8738B0	/8F38B0	ユーザ (12)		
/8738B4	/8F38B4	ユーザ (13)		
/8738B8	/8F38B8	ユーザ (14)		

エラーコード : 次ページ参照

重複ユーザ番号 : 重複しているユーザ番号を格納します。
(デフォルト=0、他ユーザ=1~14)

7 保 守

No.	エラーコード	内 容	重複ユーザ番号 設定の有無
1	/0010	相手局IPアドレスが自局IPアドレスと重複しています。	なし
2	/0011	相手局IPアドレスが他のゲートウェイIPアドレスと重複しています。	あり
3	/0012	相手局IPアドレスが他の相手局IPアドレスと重複しています。	あり
4	/0013	相手局IPアドレスのネットワークアドレスに自局と同じネットワークアドレスを設定しています。	なし
5	/0014	相手局IPアドレスのネットワークアドレスと他の相手局IPアドレスのネットワークアドレスが重複しています。	あり
6	/0016	相手局IPアドレスが255.255.255.255になっています。	なし
7	/0020	ゲートウェイIPアドレスが自局IPアドレスと重複しています。	なし
8	/0022	ゲートウェイIPアドレスが他の相手局IPアドレスと重複しています。	あり
9	/0023	ゲートウェイIPアドレスのネットワークアドレスに自局と同じネットワークアドレスを指定しています。	なし
10	/0024	ゲートウェイIPアドレスのネットワークアドレスと他の相手局IPアドレスのネットワークアドレスが重複しています。	あり
11	/0026	ゲートウェイIPアドレスが255.255.255.255になっています。	なし
12	/0030	ゲートウェイIPアドレスのサブネットが自局のサブネットと一致していません。 (*)	なし

(*) 経路情報登録済みの状態で、ツールとの接続のためにET.NETモジュールのモジュールNo.設定スイッチを4または5に設定すると、経路情報設定エラー（エラーコード/0030）となることがあります（自局のIPアドレスのネットワークアドレスが192.192.192.0の場合はエラーとなりません）。このエラーは、モジュールNo.設定スイッチを元に戻せば回復します。また、このメッセージが表示された状態でも、ET.NETとパソコンとを直接ケーブルで接続する場合はツールとの通信には支障はありません。

7.4 トラブル調査書

この調査書をご記入のうえ、販売店へご提出ください。

貴会社名			担当者		
発生日時	西暦		年	月	日
ご連絡先	ご住所				
	TEL				
	FAX				
	Eメール				
不具合モジュール型式			CPU/LPU型式		
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				
	型式				
	配線状態				
システム構成およびスイッチ設定					
通信欄					

このページは白紙です。