

# HITACHI

ユーザーズマニュアル

オプション

# ET.NET

(LQE720)

---

# SIOV

Programmable Controller

SVJ-1-128 (E)

ユーザーズマニュアル

オプション

**ET.NET**

(LQE720)

---

**SIOV**

Programmable Controller

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。

2005年 2月 (第1版) SVJ-1-128 (A) (廃版)  
2008年 4月 (第2版) SVJ-1-128 (B) (廃版)  
2010年10月 (第3版) SVJ-1-128 (C) (廃版)  
2013年 4月 (第4版) SVJ-1-128 (D) (廃版)  
2017年 6月 (第5版) SVJ-1-128 (E)

- 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



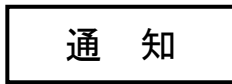
：この表示を無視して誤った取り扱いをすると、死亡または重大な傷害を引き起こす危険の存在を示す。



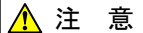

：この表示を無視して誤った取り扱いをすると、死亡または重大な傷害を引き起こすおそれのある危険の存在を示す。



：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。

## 1. 安全に関する共通的な注意事項

取り付け、運転、保守・点検の前に、以下に述べられている安全上の説明をよく読み、十分理解して従ってください。

- 操作は、このマニュアル内のすべての指示、手順に従って行ってください。
- 装置やマニュアルに表示されているすべての注意事項は特に注意を払い、必ず守ってください。これを怠ると人身上の傷害や装置の破損を引き起こすおそれがあります。
- マニュアルに記載されている以外のどんな操作も動作も行わないでください。装置について何か問題がある場合は、弊社保守員に連絡ください。  
装置やマニュアルに表示されている注意事項は、十分に検討されたものでありますが、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。  
操作にあたって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。
- 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。

以下は安全に保守作業を行うための共通的な注意事項です。よく読みそれに従ってください。

### 作業を始める前の注意

- 保守作業を行うのは、訓練を受け、資格を認められている人に限ります。
- このマニュアルおよび関連するマニュアルに記載されている安全上の指示、手順をよく読み、それに従ってください。
- 装置やマニュアルには作業に伴って発生するおそれのある傷害または製品の損傷を防ぐために必要な注意事項が表示されています。これらを十分に理解し、守ってください。
- 装置やマニュアルに表示されている注意事項は、十分に検討されたものでありますが、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。  
操作にあたって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。

### 作業中の注意

- 作業は指示された方法と順序を守って行ってください。
- 専用の工具や機材が指定されている場合は、必ずそれを使用してください。指定がない場合は、一般のもので作業目的に合致したものを使用してください。
- 測定器や電動工具などは定期点検または校正されていることを確認して使用してください。
- 作業場所とその周辺は、よく整理整頓をしてください。
- 保守用部品や資材または取り外した部品などは、つまずいたり誤って倒したりしないように通路を避けて置いてください。
- 部品がはね飛んだりするおそれのある場合は、保護眼鏡をつけてください。
- 刃のある道具を使用するときは、刃の動く方向には指などの体のいかなる部分も、絶対に近づけないでください。
- 作業完了前に装置が完全に元の状態に戻されていることを確認してください（取り外した部品がすべて元の状態に取り付けられており、余分な部品や工具、端材などが装置内に残されていないようにしてください）。

## 感電事故を防ぐための注意

- 作業場所に、感電事故の要因となりうるもの、例えば不完全な接地線やぬれた床などがなく、作業開始前に確認してください。
- 作業開始前に、非常用の電源遮断スイッチの場所と操作方法を確認してください。
- 特に別の指示がない限り、保守作業を始める前に装置への供給電源をすべて遮断してください。装置の電源スイッチを切断するだけでなく、分電盤のスイッチを切断するか、電源コードを抜いてください。  
分電盤のスイッチを切断した場合は、そこに「このスイッチをいれないこと」という貼り紙をしてください。電源にロックアウト装置がある場合は、電源切断後、鍵をかけキーを持ってください。作業を引き継いだ場合などは、推量で電源断になっていると判断してはいけません。スイッチの状態などを確認し、状況によっては計器でチェックしてください。
- 供給電源を遮断しても、装置内のある部分には一定時間電荷が残留していることがあり、注意が必要です（表示されている指示に従ってください）。
- 接地端子つきの装置を取り扱う場合は、接地線が接続されていることを確認してください。
- 活電部分の近くで作業する場合は、電源をいつでも遮断できるよう、別の人がそばで待機してください。
- 感電を防止するために、金属製の腕時計や装身具などは、作業時には身につけないでください。金属枠の眼鏡をかけている場合は、その枠が露出した活電部に触れないよう十分に注意してください。
- 手や腕は、乾いた状態にして作業してください。
- 露出した活電部の近くで作業する場合は、片手で行ってください。誤って活電部に触れた場合でも、心臓に電流が流れるのを防ぐことができます。
- 露出した活電部の近くでは歯科用の手鏡を使用しないでください。たとえプラスチック製であっても、鏡の面は導電製の金属でコーティングされており危険です。
- 特に別の指示がない限り、電源、モータなどを、装置から取り外した状態で通電してはいけません。

## 非常時の処置

### 感電事故発生の場合

- あわてないこと。感電した人に触れて第2の被害者にならないようにしてください。
- まず、被害者への電流源を遮断してください。非常用の電源切断スイッチまたはそれがいない場合は、常用の電源スイッチを切断してください。  
これができない場合は、乾いた木の棒など非導伝性のものを使って、被害者を電流源から引き離してください。
- 救急車を呼んでください。
- 被害者が意識不明の場合は、人工呼吸をしてください。  
このような場合に備えて、人工呼吸のやり方を前もって練習しておいてください。  
被害者の心臓が停止している場合は、心臓マッサージを行う必要がありますが、この処置は訓練を受け、資格のある人以外は行ってはいけません。

### 火災発生の場合

- まず、装置への電源を遮断してください。非常用の電源切断スイッチまたはそれがいない場合は、常用の電源スイッチを切断してください。
- 電源を遮断しても火災が収まらない場合は、状況に応じ、消火作業や、消防署への電話などをしてください。




## 2. このマニュアル内の警告表示

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。


### 2. 1 「 警告」と表示されているもの


(3章、3-4ページ)

 警 告
<ul style="list-style-type: none"><li>● 発煙、異臭などがあつた場合は、ただちに電源を切って原因を調査してください。</li><li>● 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。</li><li>● 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。</li></ul>

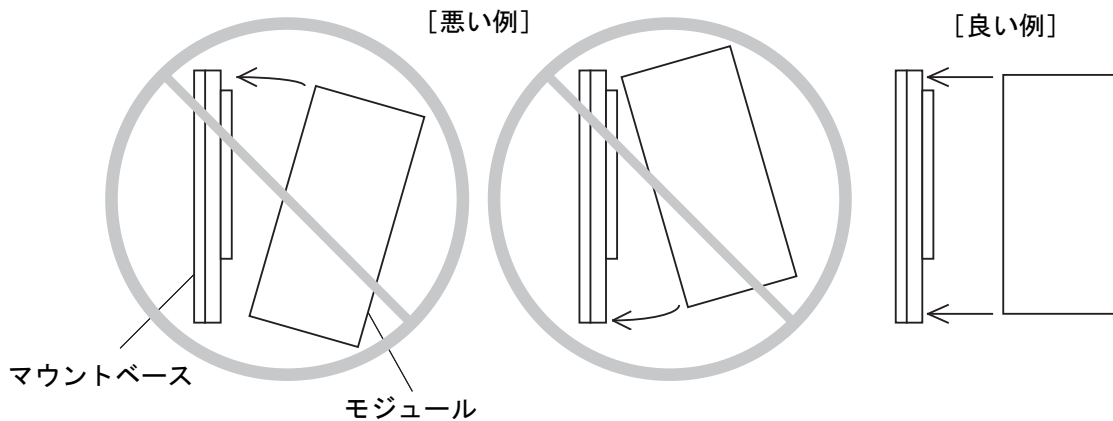
### 2. 2 「 注意」と表示されているもの


(3章、3-2ページ)

 注 意
<ul style="list-style-type: none"><li>● 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。</li><li>● 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。</li></ul>

 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。

## 2. 3 「通知」と表示されているもの

(1章、1-3ページ)

### 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。

(1章、1-4ページ)

### 通 知

この製品を使用するユーザは、Windows®環境およびユーザインタフェースについての知識が必要です。このシステムは、Windows®標準に従っています。このマニュアルは、基本的なWindows®の使用法を習得しているユーザを対象にして記述しています。

(2章、2-3ページ)

### 通 知

MAIN/SUB設定スイッチ、ST.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章、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章、3-7ページ)

### 通 知

- 接触不良または断線により誤作動する恐れがあります。コネクタを接続したときは、必ずロックされているか確認してください。
- 通電中は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作動する可能性があります。

(4章、4-5ページ)

### 通 知

- S10V 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 ET.NETシステムをインストールできません。
- S10V 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した場合、インストールで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」で一度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 ET.NETシステムをインストールしてください。

(4章、4-7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、 いいえ 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章、4-16ページ)

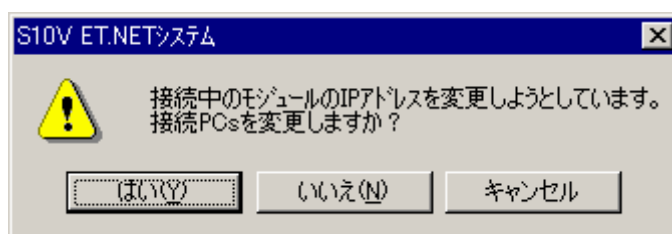
## 通 知

IPアドレスの先頭に「\*」が表示されているET.NETモジュールは、IPアドレスの設定を変更したが、PCsがリセットされていないため、変更前のIPアドレスで動作していることを表していません。変更したIPアドレスを有効にするには、PCsをリセットしてください。

(4章、4-19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章、4-22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。

(5章、5-5ページ)

## 通 知

- 1つのモジュールで、同時に使用できるソケット数は、TCPとUDPで24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（ただし、TCPの60015, 60016およびUDPの60013, 60020はシステムが占有しているので使用できません）。
- 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、LPU内のOSテーブルに設定されます。LPUの交換をした場合は、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、モジュールに対してリセットまたは停復電をして、浮いたソケットを開放してください。

(5章、5-20ページ)

## 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。

(6章、6-4ページ)

## 通 知

このET.NETモジュール（LQE720）はオートネゴシエーション設定のみのサポートとなっております。

スイッチングハブのポート設定は100Mbpsフルデュプレクス設定はしないでください。

回線負荷が上がると正常に通信できないことがあります。

(7章、7-2ページ)

## 通 知

- 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。

## 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 (LQE720)

<プログラムプロダクト>

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	2005.2	
B	モジュールの交換、増設を追加	2008.4	
C	オフライン機能追加、安全上のご注意を変更	2010.10	
D	サポートOSにWindows® 7（32bit）追加	2013.4	
E	サポートOSにWindows® 10（32bit）追加	2017.6	

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、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。

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 : LQE720

耐環境仕様品型式 : LQE720-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。

ラダー図システムやHI-FLOWシステムのイーサネット通信命令、およびソケットハンドラ機能は、LPUモジュールとCMUモジュールのモジュールレビジョンが下記の場合のみ使用できます。下記以外の組み合わせの場合、イーサネット通信命令やソケットハンドラ機能は使用しても実行されません。

モジュール名称	モジュール型式	モジュールレビジョン
LPU (基本モジュール)	LQP510	H以降
	LQP51X (Xが1以降)	A以降
CMU	LQP520	F以降
	LQP52X (Xが5以降)	A以降

<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>

- $2^n$ 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。

- $10^n$ 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）=1,000<sup>2</sup>バイトの計算値です。

1GB（ギガバイト）=1,000<sup>3</sup>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
1.3	システムソフトウェア仕様	1-4
1.3.1	システムの概要	1-4
1.3.2	必要なハードウェアとソフトウェア	1-4
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のご使用にあたり	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
4.1.1	ET.NETシステム立ち上げ手順	4-2
4.1.2	ET.NETシステム機能体系	4-3
4.2	システムインストールと立ち上げ	4-5
4.2.1	インストール	4-5
4.2.2	アンインストール	4-7
4.2.3	システム立ち上げ	4-8
4.2.4	接続PCsの変更	4-8
4.2.5	システム終了	4-10
4.3	コマンド	4-11
4.3.1	モジュール設定	4-11
4.3.2	モジュール検索設定	4-15
4.3.3	IPアドレス設定	4-18
4.3.4	経路情報	4-22
4.3.5	イーサ通信エラーログ情報（ラダー、HI-FLOW）	4-23

4.3.6	イーサ通信エラーログ情報（ソケットハンドラ）	4-25
4.3.7	DHP情報	4-27
4.3.8	DHPトレース情報	4-28
4.3.9	ネットワーク情報	4-30
4.3.10	IPアドレス設定情報ファイルの読み込み	4-32
4.3.11	IPアドレス設定情報のファイル保存	4-34
4.3.12	IPアドレス設定情報の印刷	4-36
4.3.13	IPアドレス設定情報のCSV出力	4-38
4.3.14	IPアドレス設定情報の比較	4-40
<b>5</b>	<b>プログラミング</b>	<b>5-1</b>
5.1	ET.NETのソフトウェア構成	5-2
5.2	ユーザプログラム	5-3
5.3	ソケットハンドラ	5-3
5.3.1	ソケットハンドラー一覧	5-4
5.4	ソケットハンドラ発行手順例	5-30
5.4.1	TCP/IPプログラム使用例	5-30
5.4.2	UDP/IPプログラム使用例	5-31
5.5	CPU間通信プログラム例	5-34
5.5.1	システム構成およびプログラム構成	5-34
5.5.2	CPU01側プログラムのフローチャートとプログラム例	5-35
5.5.3	CPU02側プログラムのフローチャートとプログラム例	5-39
5.6	CPU間連続通信プログラム例	5-42
5.6.1	システム構成およびプログラム構成	5-42
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	5-43
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	5-48
<b>6</b>	<b>利用の手引き</b>	<b>6-1</b>
6.1	推奨するネットワーク構成部品	6-2
6.2	システム構成	6-3
6.3	システム定義情報	6-5
6.3.1	物理アドレス	6-5
6.3.2	IPアドレス	6-5
6.3.3	サブネットマスク	6-7
6.3.4	経路情報	6-7
<b>7</b>	<b>保守</b>	<b>7-1</b>
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	エラーログ情報の見方	7-7
7.3.2	DHPトレース情報の見方	7-14
7.3.3	ネットワーク情報の見方	7-18
7.3.4	ソケットハンドラ検出のエラーコード	7-39
7.4	トラブル調査書	7-42



## 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-T/100BASE-TX通信ケーブルの配線	3-7
図 4-1	立ち上げ手順	4-2
図 4-2	ET.NETシステム機能体系 (LQE720実装時)	4-3
図 4-3	ET.NETシステム機能体系 (LQE520実装時)	4-4
図 4-4	ET.NETシステム機能体系 (オフラインモード)	4-4
図 4-5	[ [S10V] ET.NET] 画面	4-8
図 4-6	[ [S10V] ET.NET] 画面 (RS-232C選択)	4-8
図 4-7	[ [S10V] ET.NET] 画面 (イーサネット選択)	4-9
図 4-8	[ [S10V] ET.NET] 画面 (ステーションNo.検索選択)	4-9
図 4-9	[ [オンライン] モジュール設定] 画面	4-10
図 4-10	[モジュール検索設定] 画面	4-10
図 4-11	[ [オンライン] モジュール設定] 画面	4-11
図 4-12	[ [S10V] ET.NET] 画面 ( [オフライン] ラジオボタンをクリック)	4-12
図 4-13	[ファイルを開く] 画面	4-13
図 4-14	「新規作成確認」ダイアログボックス	4-13
図 4-15	[ [オフライン] モジュール設定] 画面	4-14
図 4-16	[モジュール検索設定] 画面	4-15
図 4-17	[ [オンライン] IPアドレス設定] 画面	4-18
図 4-18	「リセット確認」メッセージ	4-19
図 4-19	「ファイル保存確認」メッセージ	4-20
図 4-20	「保存しました」メッセージ	4-20
図 4-21	「ファイルに保存しないで終了します。」メッセージ	4-21
図 4-22	[ [オンライン] 経路情報] 画面	4-22
図 4-23	[イーサ通信エラーログ情報 (ラダー、HI-FLOW)] 画面	4-23
図 4-24	[名前を付けて保存] 画面 (イーサ通信エラーログ情報 (ラダー、HI-FLOW) 保存)	4-24
図 4-25	[イーサ通信エラーログ情報 (ソケットハンドラ)] 画面	4-25
図 4-26	[名前を付けて保存] 画面 (イーサ通信エラーログ情報 (ソケットハンドラ) 保存)	4-26
図 4-27	[DHP情報] 画面	4-27
図 4-28	[DHPトレース情報] 画面	4-28
図 4-29	[名前を付けて保存] 画面 (DHPトレース情報保存)	4-29
図 4-30	[ネットワーク情報] 画面	4-30
図 4-31	[名前を付けて保存] 画面 (ネットワーク情報保存)	4-31
図 4-32	[ [オンライン] モジュール設定] 画面 ( <span style="border: 1px solid black; padding: 2px;">IPアドレス設定</span> ボタンをクリック)	4-32

図 4-33	[ [オンライン] IPアドレス設定] 画面 ( <b>読み込み</b> ボタンをクリック)	4-32
図 4-34	[ファイルを開く] 画面	4-33
図 4-35	[ [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)	4-33
図 4-36	[ [オンライン] モジュール設定] 画面 ( <b>IPアドレス設定</b> ボタンをクリック)	4-34
図 4-37	[ [オンライン] IPアドレス設定] 画面 ( <b>保存</b> ボタンをクリック)	4-34
図 4-38	[名前を付けて保存] 画面	4-35
図 4-39	[ [オンライン] モジュール設定] 画面 ( <b>IPアドレス設定</b> ボタンをクリック)	4-36
図 4-40	[ [オンライン] IPアドレス設定] 画面 ( <b>印刷</b> ボタンをクリック)	4-36
図 4-41	[印刷] ダイアログボックス	4-37
図 4-42	IPアドレス設定情報印刷例	4-37
図 4-43	[ [オンライン] モジュール設定] 画面 ( <b>IPアドレス設定</b> ボタンをクリック)	4-38
図 4-44	[ [オンライン] IPアドレス設定] 画面 ( <b>CSV出力</b> ボタンをクリック)	4-38
図 4-45	[名前を付けて保存] 画面 (CSV出力)	4-39
図 4-46	[ [オンライン] モジュール設定] 画面	4-40
図 4-47	[ファイルを開く] 画面	4-41
図 4-48	「不一致はありません。」メッセージ	4-41
図 4-49	「比較不一致」メッセージ例	4-42
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	5-3
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	CPU間通信プログラムのシステム構成例	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	CPU間連続通信プログラムのシステム構成例	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	ハブの接続構成	6-3
図 6-2	10Mbps専用ハブのみによる構成例	6-3
図 6-3	100Mbpsハブ クラス1による構成例	6-4
図 6-4	100Mbpsハブ クラス2による構成例	6-4
図 6-5	100Mbpsスイッチングハブによる構成例	6-4
図 7-1	トラブルシューティング手順	7-5
図 7-2	接続ステータス状態遷移	7-20

# 表 目 次

表 1-1	システムソフトウェア（ツール）の種類 .....	1-4
表 5-1	ソケットハンドラー一覧 .....	5-4
表 6-1	ネットワーク構成部品一覧 .....	6-2
表 7-1	保守点検項目 .....	7-2
表 7-2	パニックログエラーメッセージフォーマット一覧 .....	7-7
表 7-3	パニックログデフォルトエラーメッセージ一覧 .....	7-7
表 7-4	パニックログ以外エラーメッセージフォーマット一覧 .....	7-8
表 7-5	パニックログ以外デフォルトエラーメッセージ一覧 .....	7-9
表 7-6	エラーメッセージ .....	7-11
表 7-7	DHPコード一覧 .....	7-15
表 7-8	ソケットハンドラー検出のエラーコード .....	7-39

# 1 仕 様

## 1 仕 様

### 1.1 用 途

ET.NETモジュール（型式：LQE720）は、S10V LPUモジュールと組み合わせ、IEEE802.3i仕様（10BASE-T）、またはIEEE802.3u仕様（10BASE-TX）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。なお、このモジュール（LQE720）をS10miniのCPUユニットに実装して使用することはできません。また、LQE520と混在させて使用することもできません。

### 1.2 仕 様

#### 1.2.1 一般仕様

項 目	仕 様
型 式	LQE720
マウントベース（*）への最大実装枚数	2モジュール/LPU（S10V専用）
質 量	180g

（\*）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。

LQE520とLQE720のET.NETは混在実装できません。  
必ず実装するマウントベースを分けてください。

#### 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	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	ツイストペアケーブル：最長100m/セグメント （カテゴリ5以上）
データ転送速度	10Mbps, 100Mbps

## 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。

## 1. 3 システムソフトウェア仕様

### 1. 3. 1 システムの概要

ET.NETモジュールを使用するためには、様々な情報をモジュールに登録してください。以下のシステムソフトウェア（ツール）により、一般的なWindows®上で作動するアプリケーションと等価なオペレーションで、モジュール情報を登録できます。

表 1-1 システムソフトウェア（ツール）の種類

パッケージ名称	型式	提供形態
	S10V用	
ET.NETシステム	S-7895-29	別売り

### 1. 3. 2 必要なハードウェアとソフトウェア

ET.NETモジュールのシステムソフトウェアを使用するためには以下のハードウェアおよびソフトウェアが必要です。

- ・ Pentium 300MHz以上のCPUを搭載したパーソナルコンピュータ（以下パソコンと略します）本体
- ・ 1GHz以上のCPUを搭載したパソコン本体（Windows® 7（32bit）、Windows® 10（32bit）使用時）
- ・ 800×600ドット（SVGA）以上の解像度を持つディスプレイ
- ・ Microsoft® Windows® 2000 operating system、Microsoft® Windows® XP operating system、Microsoft® Windows® 7（32bit） operating system、またはMicrosoft® Windows® 10（32bit） operating system
- ・ 64MB以上のRAM（Windows® 2000使用時）
- ・ 128MB以上のRAM（Windows® XP使用時）
- ・ 1GB以上のRAM（Windows® 7（32bit）、Windows® 10（32bit）使用時）
- ・ 10MB以上の空きハードディスク容量
- ・ パソコンとLPUユニット間の接続ケーブル（D-sub9ピンコネクタを持つRS-232Cクロスケーブル）またはパソコンとCMUモジュール、またはET.NETモジュール間の接続ケーブル（RJ-45モジュラコネクタを持つ10BASE-Tまたは100BASE-Tツイストペアクロスケーブル）

## 通 知

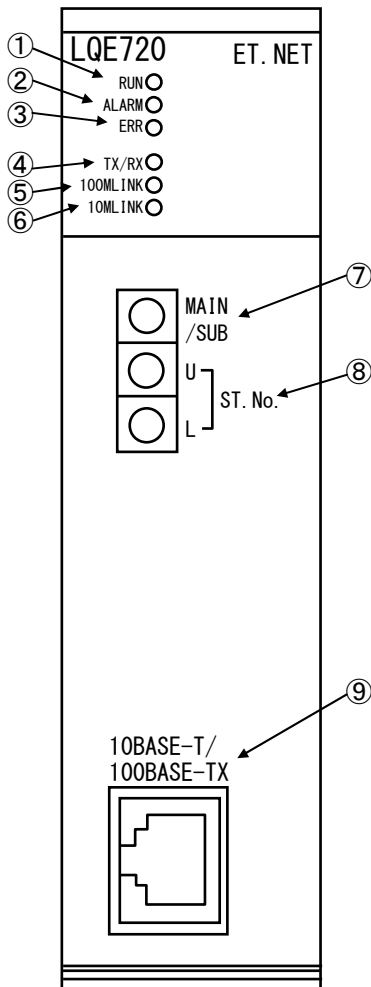
この製品を使用するユーザは、Windows®環境およびユーザインタフェースについての知識が必要です。このシステムは、Windows®標準に従っています。このマニュアルは、基本的なWindows®の使用法を習得しているユーザを対象にして記述しています。

## 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	RUN LED	モジュールが正常に立ち上がったとき点灯します。														
②	ALARM LED	IPアドレス未設定等の誤設定時点灯します。														
③	ERR LED	ハードウェア異常時に点灯します。														
④	TX/RX LED	送受信しているとき点灯します。														
⑤	100MLINK LED	通信回線が100Mbpsで接続されているとき点灯します。														
⑥	10MLINK LED	通信回線が10Mbpsで接続されているとき点灯します。														
⑦	MAIN/SUB 設定スイッチ	このスイッチでメイン/サブ設定を行います。 設定値は下表を参照してください。 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>SW設定</th> <th>内 容</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>メイン設定</td> </tr> <tr> <td>1</td> <td>サブ設定</td> </tr> <tr> <td>2~B</td> <td>設定禁止</td> </tr> <tr> <td>C~F</td> <td>保守設定用につき設定禁止</td> </tr> </tbody> </table>	SW設定	内 容	0	メイン設定	1	サブ設定	2~B	設定禁止	C~F	保守設定用につき設定禁止				
SW設定	内 容															
0	メイン設定															
1	サブ設定															
2~B	設定禁止															
C~F	保守設定用につき設定禁止															
⑧	ST.No. 設定スイッチ	このスイッチでIPアドレスの設定を行います。 設定値は下表を参照してください。 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2">SW設定</th> <th rowspan="2">内 容</th> </tr> <tr> <th>U</th> <th>L</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>設定済みのIPアドレスで動作させるとき使用</td> </tr> <tr> <td>0 \ F</td> <td>1 \ E</td> <td>ネットワークよりIPアドレスを設定して動作させるとき使用</td> </tr> <tr> <td>F</td> <td>F</td> <td>IPアドレス固定(192.192.192.1)としToolと接続して使用</td> </tr> </tbody> </table> <p>*詳細使用方法は「4 オペレーション」を参照してください。</p>	SW設定		内 容	U	L	0	0	設定済みのIPアドレスで動作させるとき使用	0 \ F	1 \ E	ネットワークよりIPアドレスを設定して動作させるとき使用	F	F	IPアドレス固定(192.192.192.1)としToolと接続して使用
SW設定		内 容														
U	L															
0	0	設定済みのIPアドレスで動作させるとき使用														
0 \ F	1 \ E	ネットワークよりIPアドレスを設定して動作させるとき使用														
F	F	IPアドレス固定(192.192.192.1)としToolと接続して使用														
⑨	RJ45コネクタ	10BASE-T/100BASE-TXで通信するためのコネクタです。														

通 知

MAIN/SUB設定スイッチ、ST.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。

このページは白紙です。

## 3 実装と配線

### 3 実装と配線

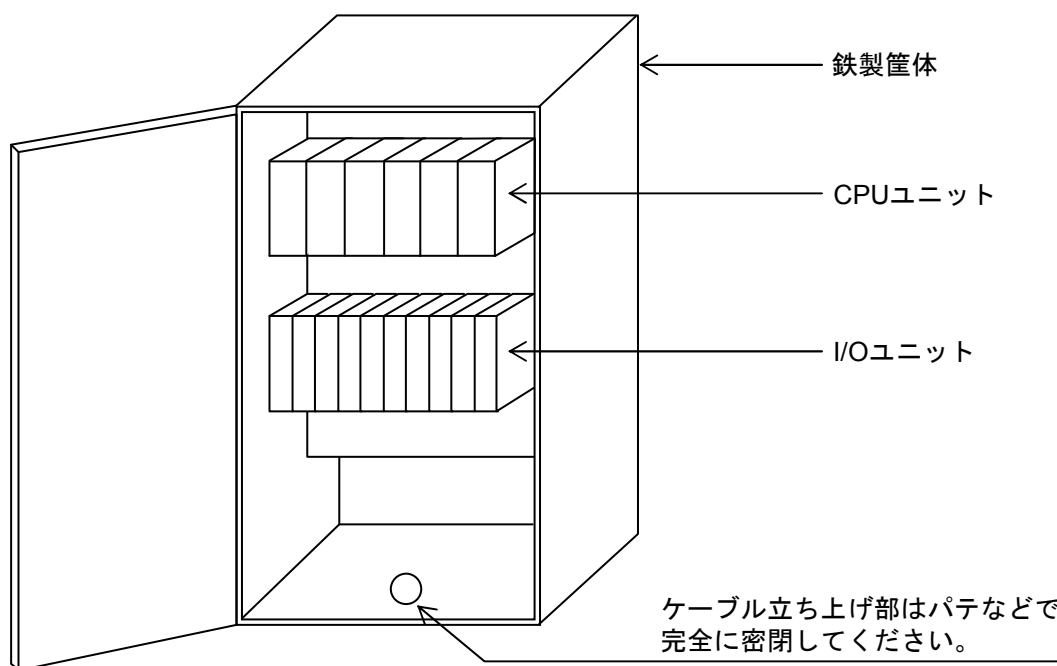
#### 3.1 PCsのご使用にあたり

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、防水筐体の実装してください。



#### ⚠ 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

## 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。

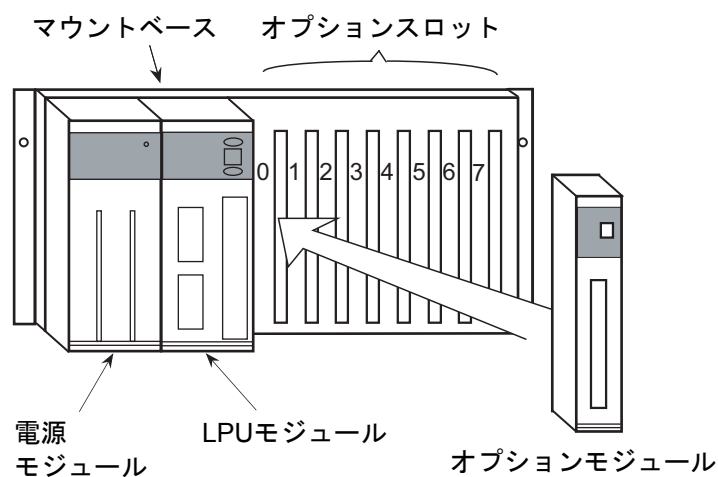


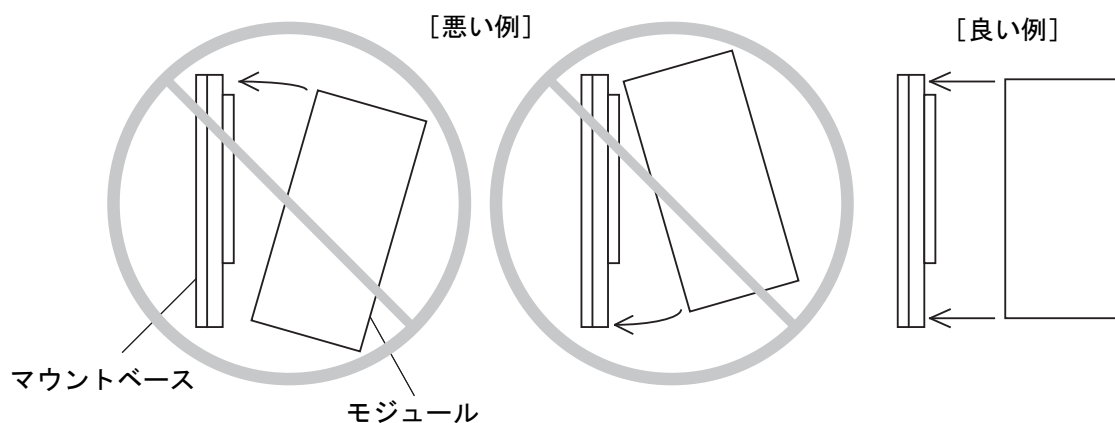
図 3-1 オプションモジュールの実装

#### 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

ET.NETモジュールへのアース配線は不要です。

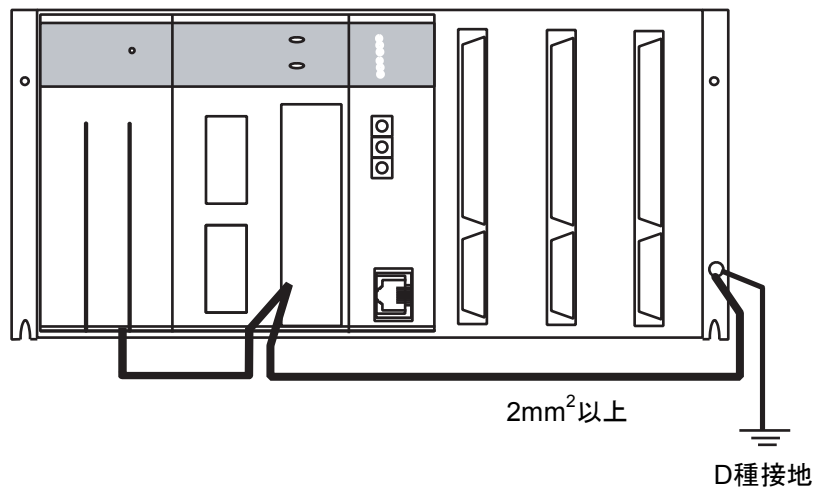


図 3-2 アース配線

### 3.5 通信ケーブルの配線

#### (1) 10BASE-T, 100BASE-TX通信ケーブルの配線

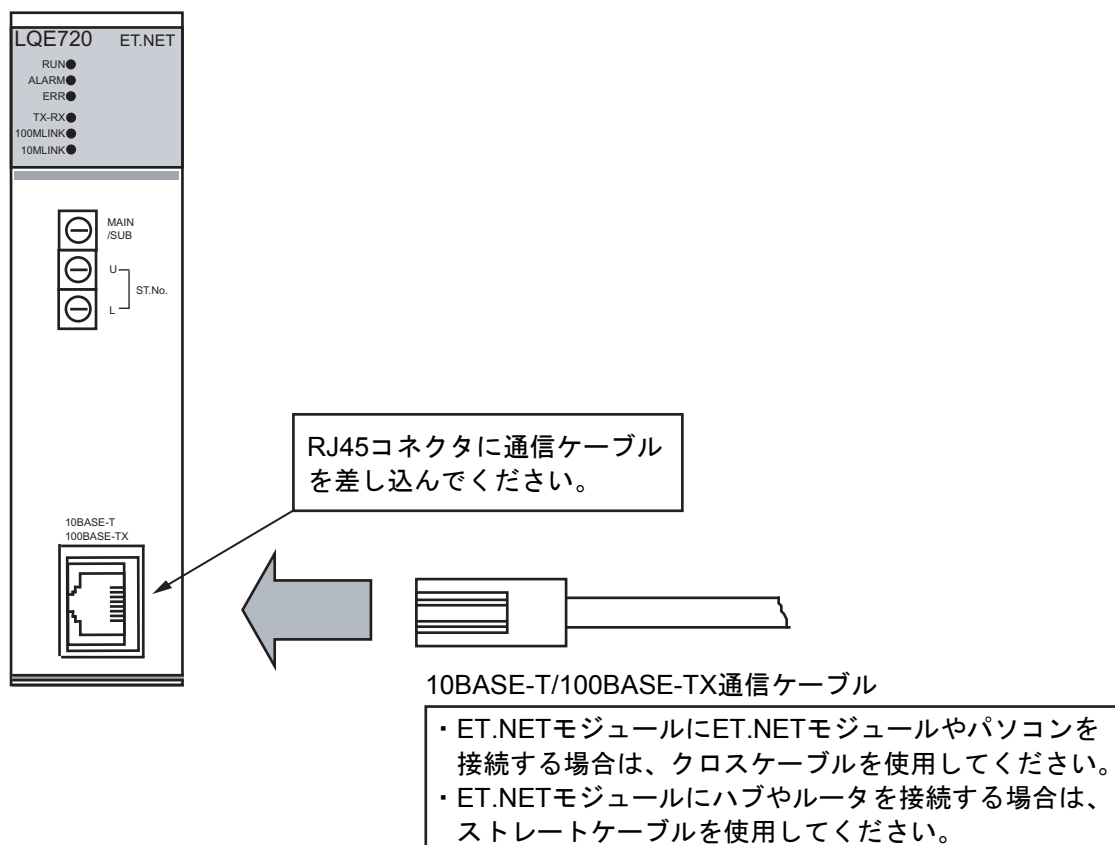


図 3 - 3 10BASE-T/100BASE-TX通信ケーブルの配線

#### 通 知

- 接触不良または断線により誤作動する恐れがあります。コネクタを接続したときは、必ずロックされているか確認してください。
- 通電中は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作動する可能性があります。

このページは白紙です。

## 4 オペレーション

## 4 オペレーション

### 4.1 立ち上げ手順

#### 4.1.1 ET.NETシステム立ち上げ手順

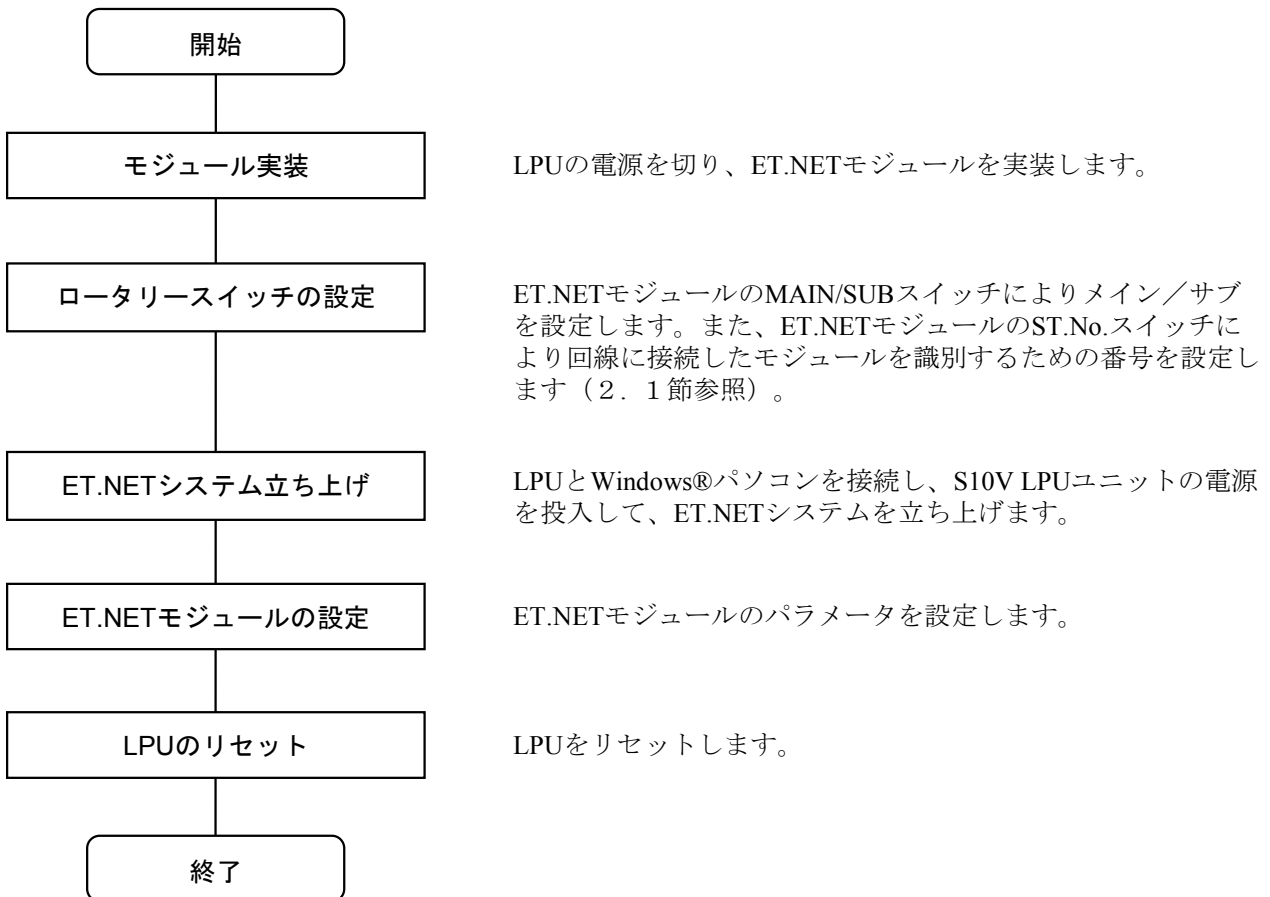


図4-1 立ち上げ手順

## 4. 1. 2 ET.NETシステム機能体系

(1) LQE720を実装しているとき (オンラインモード)

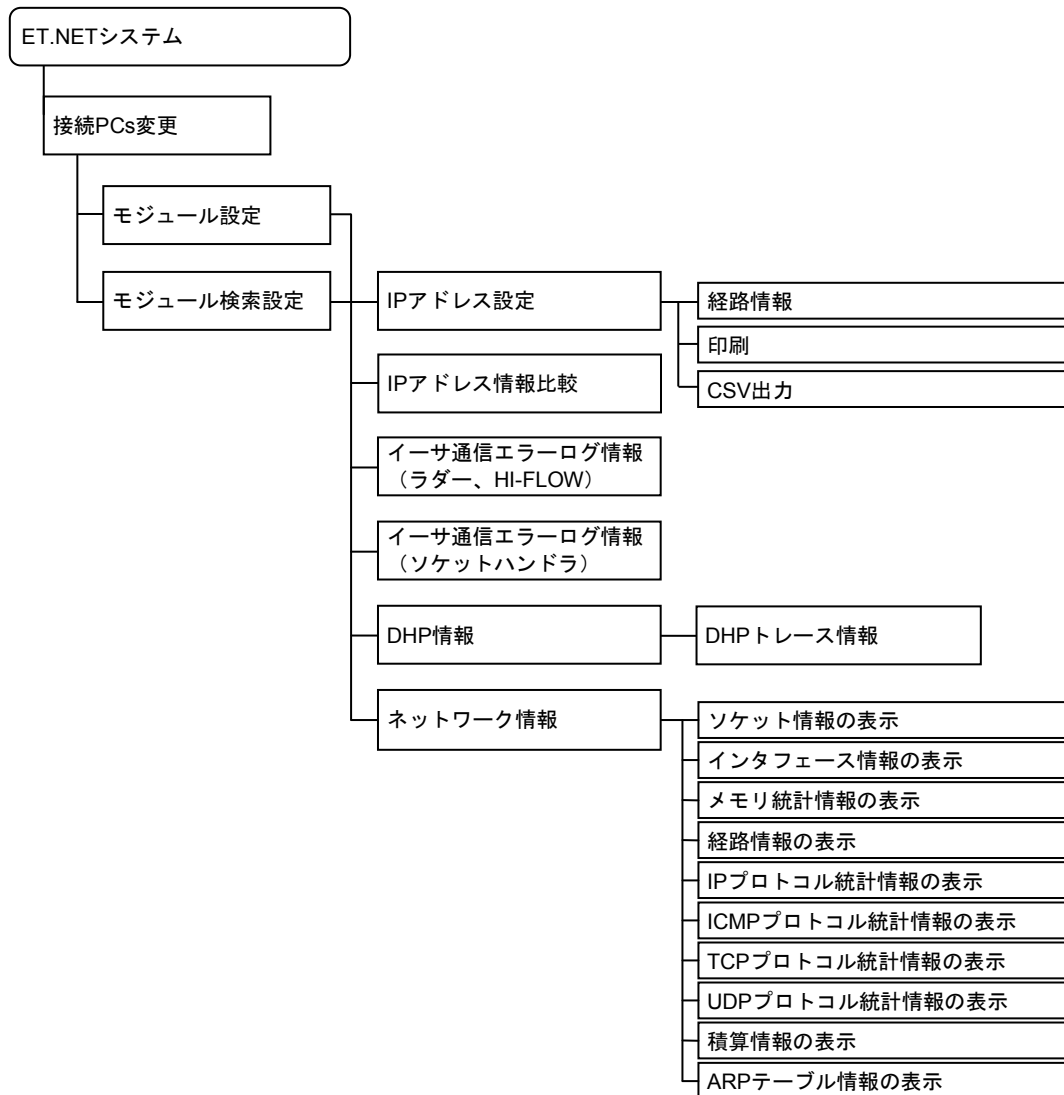


図 4-2 ET.NETシステム機能体系 (LQE720実装時)

## 4 オペレーション

### (2) LQE520を実装しているとき (オンラインモード)

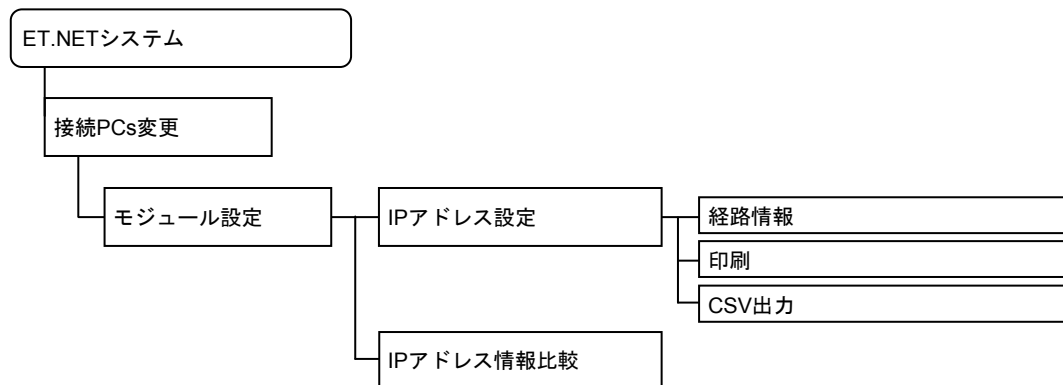


図 4-3 ET.NETシステム機能体系 (LQE520実装時)

### (3) オフラインモード

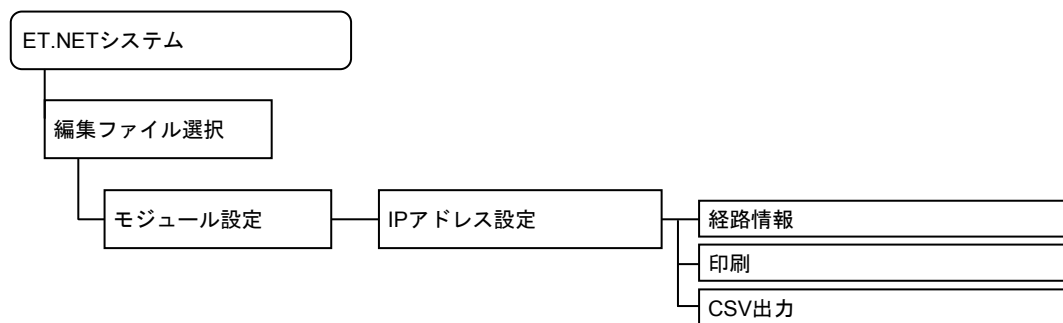


図 4-4 ET.NETシステム機能体系 (オフラインモード)

## 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

S10V ET.NETシステムをインストールするには、S10V 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てください。

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

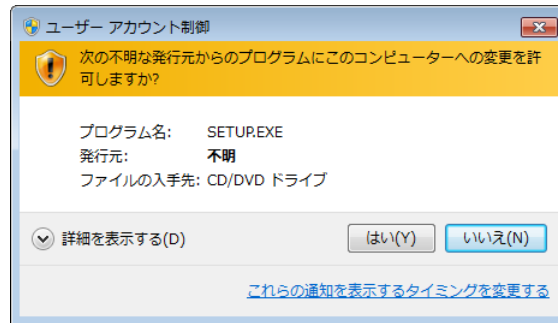
- S10V ET.NETシステムを動作させるためには、S10V基本システムが必要です。インストールされていない場合は、S10V ET.NETシステムをインストールできません。
- S10V ET.NETシステムをインストールする前に、すべてのWindows®上で作動するプログラムを必ず終了してください。ウイルス監視ソフトウェアなどメモリに常駐しているプログラムも必ず終了してください。終了せずにインストールした場合、インストールでエラーが発生する場合があります。その場合は、「4.2.2 アンインストール」で一度アンインストールし、すべてのWindows®上で作動するプログラムを終了してから再度S10V ET.NETシステムをインストールしてください。



## 4 オペレーション

### <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、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. 2. 2 アンインストール

S10V 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 “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 “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

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 オペレーション

### 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 **スタート** ボタンから [Hitachi S10V] - [S10V ET.NETシステム] を選択し、ET.NETシステムを起動すると、[ [S10V] ET.NET ] 画面が表示されます。

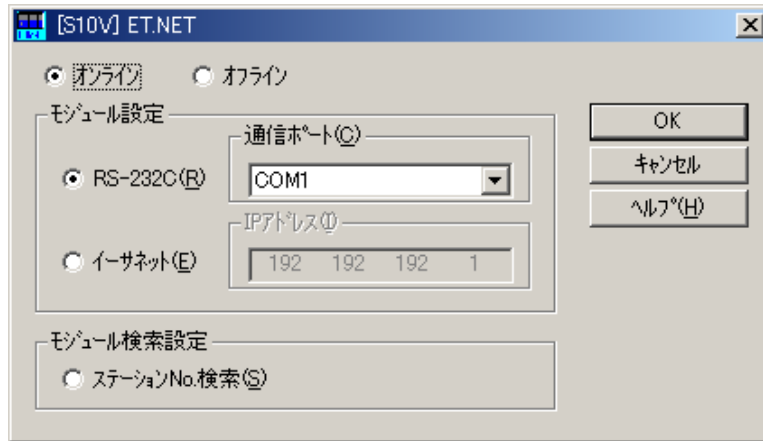


図 4-5 [ [S10V] ET.NET ] 画面

### 4.2.4 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 [ [S10V] ET.NET ] 画面からPCsとパソコンを接続する通信種類を選択します。

- RS-232C接続のとき

[RS-232C] ラジオボタンをチェックし、「通信ポート」をプルダウンメニューから選択してください。“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。

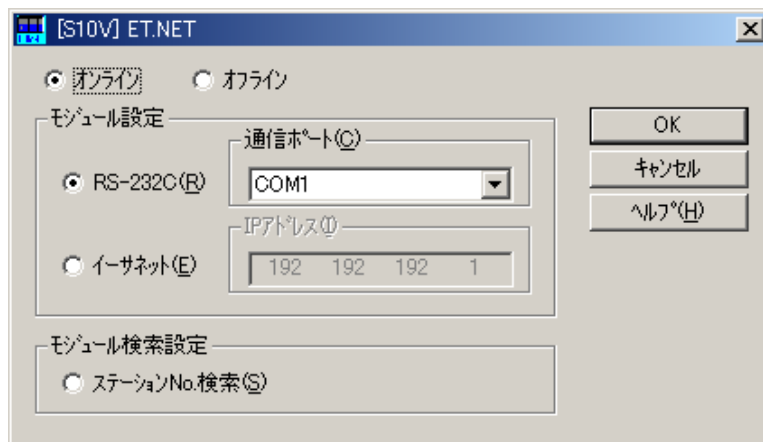


図 4-6 [ [S10V] ET.NET ] 画面 (RS-232C選択)

- ・イーサネット接続のとき

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



図 4-7 [ [S10V] ET.NET ] 画面 (イーサネット選択)

接続するPCsのIPアドレスが不明な場合は、[ステーションNo.検索] ラジオボタンをチェックしてください。

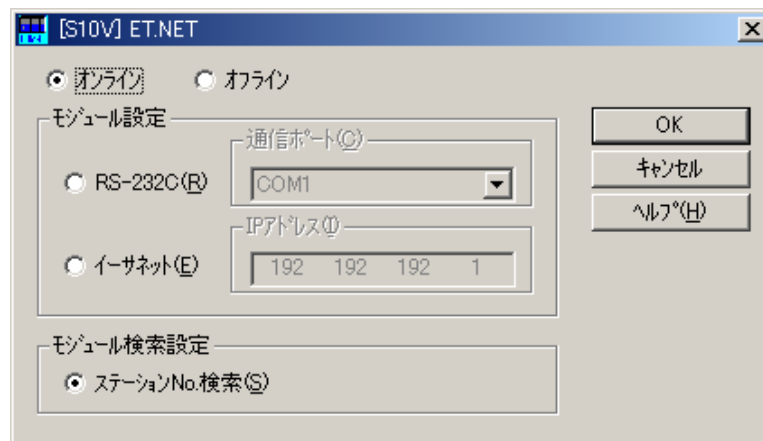


図 4-8 [ [S10V] ET.NET ] 画面 (ステーションNo.検索選択)

## 4 オペレーション

- ② 設定が終了したら **OK** ボタンをクリックしてください。[RS-232C] ラジオボタンまたは [イーサネット] ラジオボタンをチェックしている場合 [ [オンライン] モジュール設定 ] 画面が、[ステーションNo.検索] ラジオボタンをチェックしている場合は [モジュール検索設定] 画面が表示されます。この後は、目的のコマンドのボタンをクリックしてください。

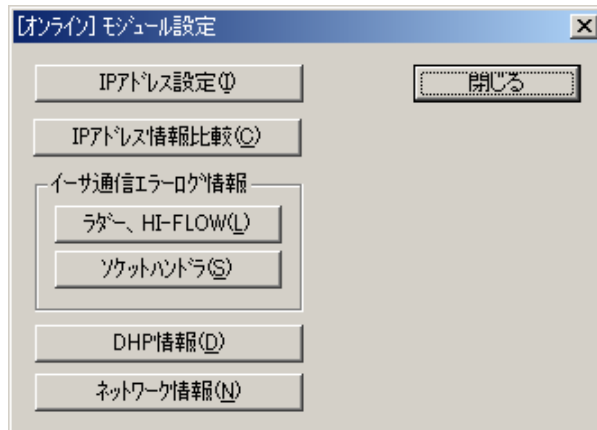


図 4-9 [ [オンライン] モジュール設定 ] 画面

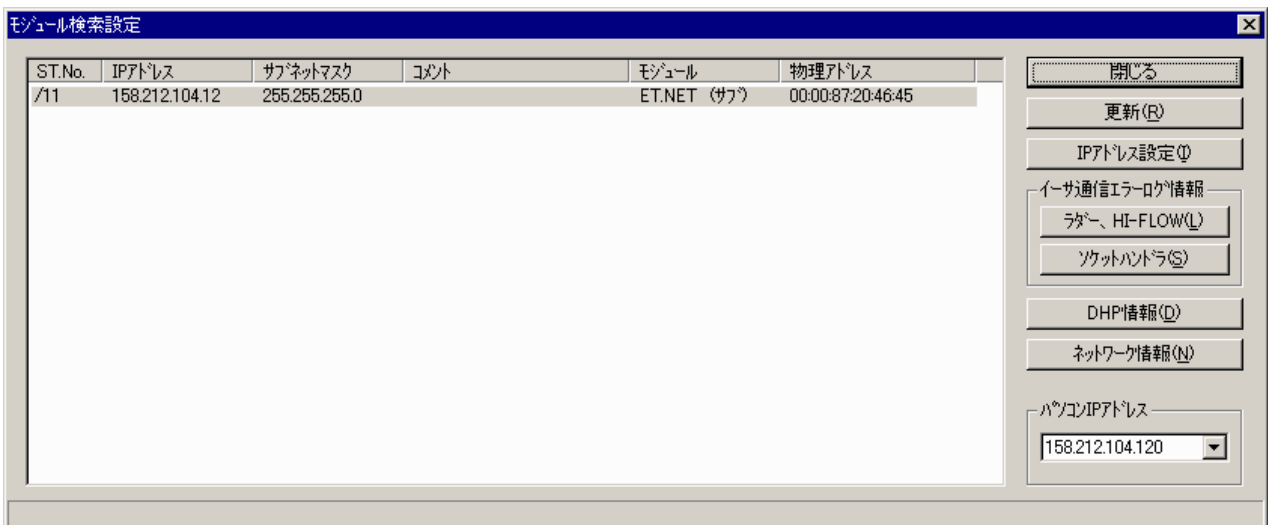


図 4-10 [モジュール検索設定] 画面

### 4.2.5 システム終了

[ [S10V] ET.NET ] 画面 (図 4-5 参照) において、**X** または **キャンセル** ボタンをクリックしてください。

## 4.3 コマンド

### 4.3.1 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：以下にオンラインとオフラインの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  
チェックし、 **OK** ボタンをクリックしてください。
- ② [ [オンライン] モジュール設定 ] 画面が表示されます。

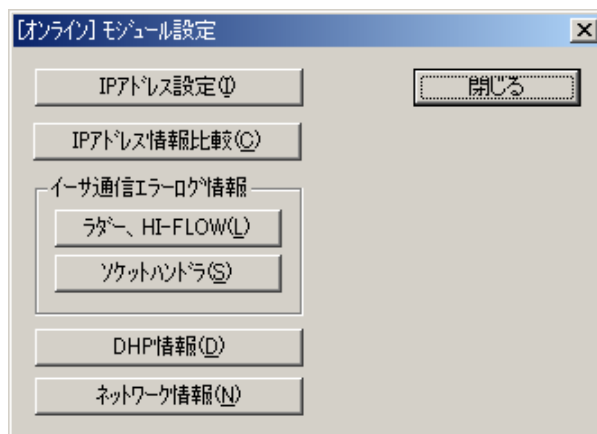


図4-11 [ [オンライン] モジュール設定 ] 画面

- 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。 [ [オンライン] IPアドレス設定 ] 画面が表示されます。  
詳細は、「4.3.3 IPアドレス設定」を参照してください。  
ET.NETモジュールのIPアドレス情報を比較したい場合は、 **IPアドレス情報比較** 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。比較したいIPアドレス設定情報ファイルを選択してください。  
詳細は、「4.3.14 IPアドレス設定情報の比較」を参照してください。
- ④ ラダーやHI-FLOWのイーサネット通信で発生したエラーログを表示したい場合は、 **ラダー、HI-FLOW** ボタンをクリックしてください。 [イーサ通信エラーログ情報 (ラダー、HI-FLOW)] 画面が表示されます。詳細は、「4.3.5 イーサ通信エラーログ情報 (ラダー、HI-FLOW)」を参照してください。また、ソケットハンドラのイーサネット通信で発生したエラーログを表示したい場合は、 **ソケットハンドラ** ボタンをクリックしてください。 [イーサ通信エラーログ情報 (ソケットハンドラ)] 画面が表示されます。詳細は、「4.3.6 イーサ通信エラーログ情報 (ソケットハンドラ)」を参照してください。

## 4 オペレーション

- ⑤ 現在のDHP記録モードやDHPトレース情報を表示したい場合は、**DHP情報** ボタンをクリックしてください。[DHP情報] 画面が表示されます。詳細は、「4. 3. 7 DHP情報」を参照してください。
- ⑥ CMUモジュールおよびET.NETモジュールのネットワーク状態や積算情報を表示したい場合は、**ネットワーク情報** ボタンをクリックしてください。[ネットワーク情報] 画面が表示されます。詳細は、「4. 3. 9 ネットワーク情報」を参照してください。
- ⑦ [[オンライン] 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。

### (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。

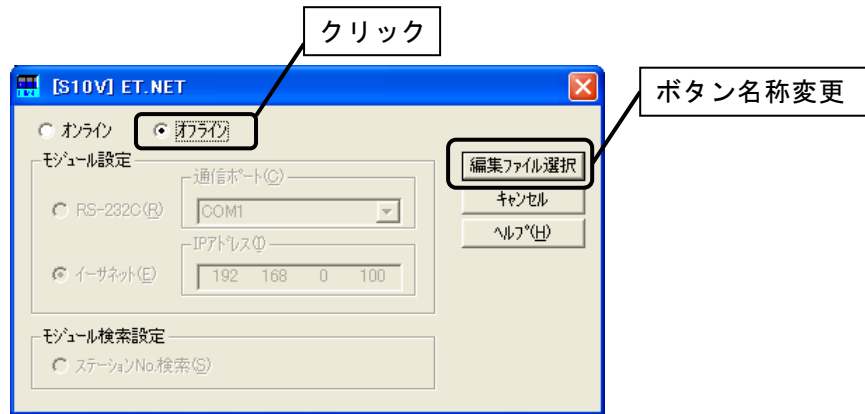


図4-12 [[S10V] ET.NET] 画面（[オフライン] ラジオボタンをクリック）

- 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。

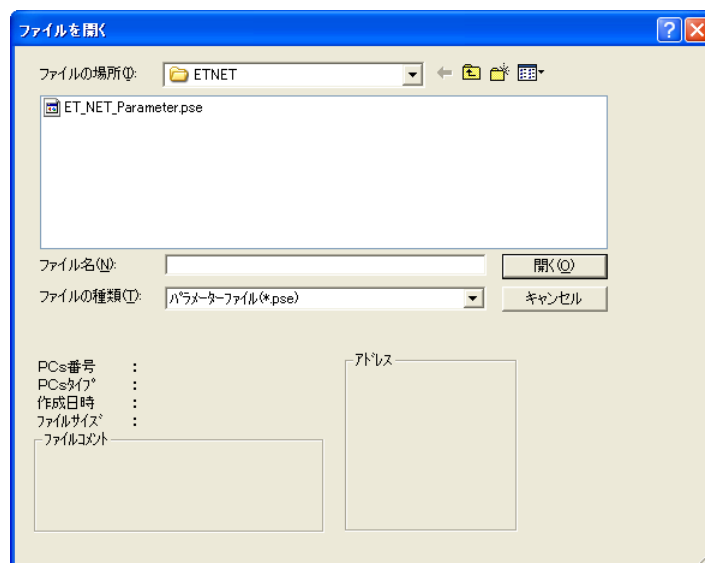


図 4-13 [ファイルを開く] 画面

既存のET.NETパラメータ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。

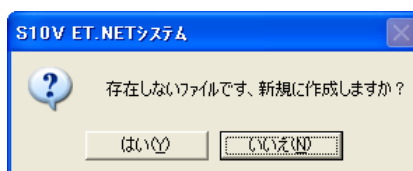


図 4-14 「新規作成確認」ダイアログボックス



## 4 オペレーション

**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。

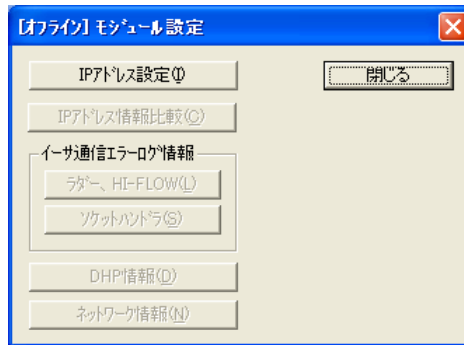


図 4-15 [[オフライン] モジュール設定] 画面

- 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、「4.3.3 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. 3. 2 モジュール検索設定

機能：パソコンと同じネットワークに接続されているET.NETモジュールを検索して一覧を表示します。

操作：以下に操作手順を示します。

- ① [ [S10V] ET.NET ] 画面から [ステーションNo.検索] ラジオボタンをチェックし、 **OK** ボタンをクリックしてください。
- ② [モジュール検索設定] 画面が表示されます。

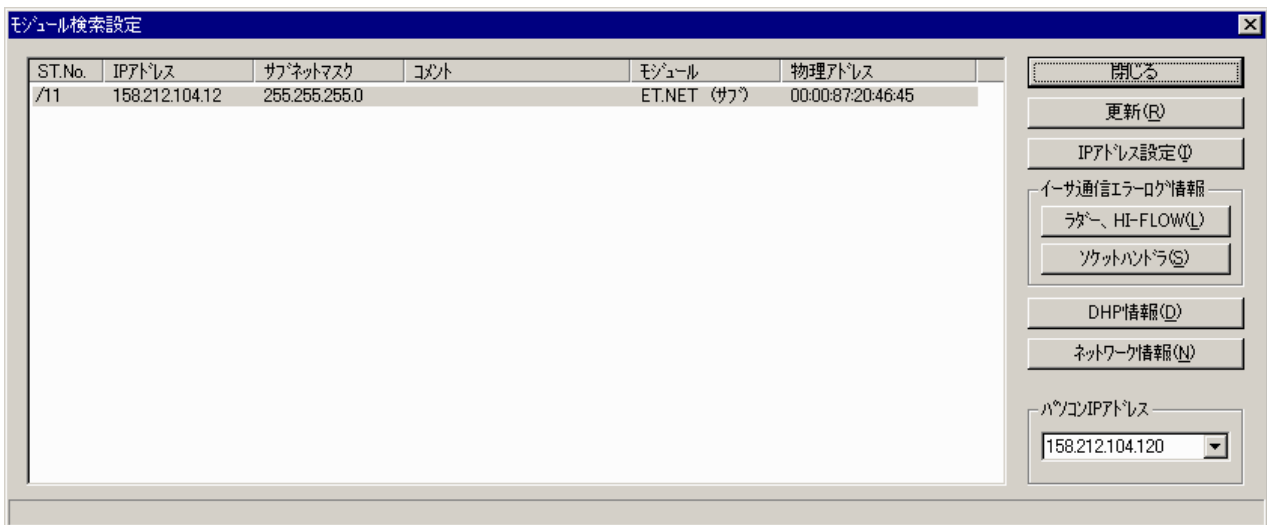


図 4-16 [モジュール検索設定] 画面

リストにパソコンと同じネットワークに接続されているET.NETモジュールの一覧が表示されます。このリストから接続したいET.NETモジュールを選択してください。

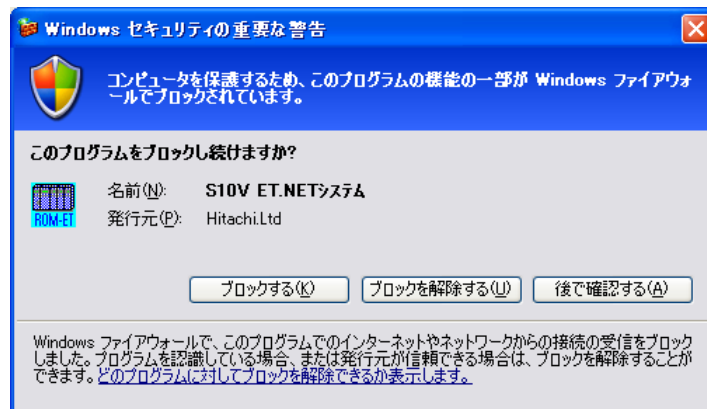
- ③ 複数LANカード実装のパソコンで接続IPアドレスを変更したい場合は、 [パソコンIPアドレス] ボックスから接続したいパソコンのIPアドレスを選択してください。IPアドレス選択後、 **更新** ボタンをクリックすることでリストの表示内容が更新されます。
- ④ ET.NETモジュールのIPアドレス情報を設定したい場合は、 **IPアドレス設定** ボタンをクリックしてください。
- ⑤ ラダーやHI-FLOWのイーサネット通信で発生したエラーログを表示したい場合は **ラダー、HI-FLOW** ボタンをクリックしてください。また、ソケットハンドラのイーサネット通信で発生したエラーログを表示したい場合は **ソケットハンドラ** ボタンをクリックしてください。
- ⑥ 現在のDHP記録モードやDHPトレース情報を表示したい場合は **DHP情報** ボタンをクリックしてください。
- ⑦ CMUモジュールおよびET.NETモジュールのネットワーク状態や積算情報を表示したい場合は **ネットワーク情報** ボタンをクリックしてください。
- ⑧ [モジュール検索設定] 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。

### 通知

IPアドレスの先頭に「\*」が表示されているET.NETモジュールは、IPアドレスの設定を変更したが、PCsがリセットされていないため、変更前のIPアドレスで動作していることを表しています。変更したIPアドレスを有効にするには、PCsをリセットしてください。

#### <注意>

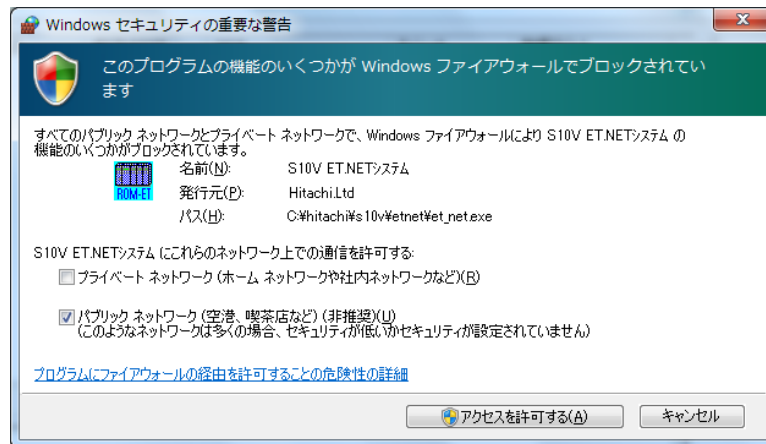
ご使用のパソコンのOSがWindows® XPで、SP2（サービスパック2）、またはSP3（サービスパック3）がインストールしてある場合、S10V ET.NETシステムの初期画面でモジュール検索設定機能を選択すると、以下に示す「Windows セキュリティの重要な警告」ダイアログボックスが表示される場合があります。



以降、S10V ET.NETシステムを継続して使用する場合は、**ブロックを解除する(U)** ボタンをクリックして、Windowsファイアウォールによるブロックを解除してください。

## &lt;注意&gt;

ご使用のパソコンのOSがWindows® 7 (32bit) 、Windows® 10 (32bit) の場合、S10V ET.NETシステムの初期画面でモジュール検索設定機能を選択すると、以下に示す「Windows セキュリティの重要な警告」ダイアログボックスが表示される場合があります。



以降、S10V ET.NETシステムを継続して使用する場合は、「プライベートネットワーク」をチェックし、**アクセスを許可する(A)** ボタンをクリックして、Windowsファイアウォールによるブロックを解除してください。Windowsファイアウォールの解除の際は、「パブリックネットワーク」のチェックは外してのご使用を推奨します。

## 4 オペレーション

### 4.3.3 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 ] 画面または [ モジュール検索設定 ] 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 4-17 [ [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/ブロードキャストアドレス

ET.NETモジュールのIPアドレス、サブネットマスク、およびブロードキャストアドレスを設定してください。設定の詳細は「6.3 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。

・コメント

ユーザのコメントを半角英数字またはハイフンで最大16文字まで設定できます。必要に応じて設定してください。

- ③ ルーティングテーブルを設定する場合は、 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アドレスを設定してください。
- ④ 設定が終了したら、 ボタンをクリックしてください。  
設定しない場合は、 ボタンをクリックしてください。
- ⑤  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。

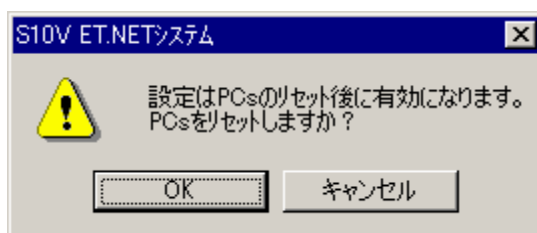
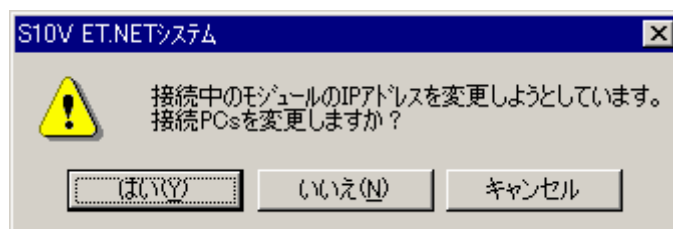


図 4-18 「リセット確認」メッセージ

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。 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  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と接続しない場合は  ボタンを、接続しない場合は  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 オペレーション

### (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 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。

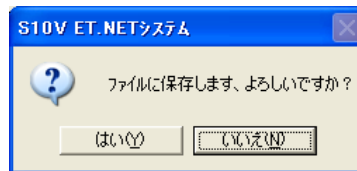


図 4-19 「ファイル保存確認」メッセージ

**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。

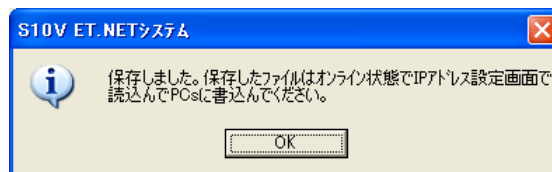


図 4-20 「保存しました」メッセージ

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。

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。

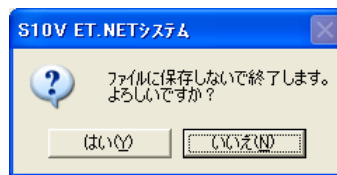


図4-21 「ファイルに保存しないで終了します。」メッセージ

**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 オペレーション

### 4.3.4 経路情報

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。オンラインモードとオフラインモードで同様の手順となります。

- 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 4-22 [[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. 3. 5 イーサ通信エラーログ情報（ラダー、HI-FLOW）

機能：ラダーやHI-FLOWのイーサネット通信で発生したエラーのトレースログを表示します。

操作：以下に操作手順を示します。

- ① [モジュール設定] 画面または [モジュール検索設定] 画面の **ラダー、HI-FLOW** ボタンをクリックしてください。
- ② [イーサ通信エラーログ情報（ラダー、HI-FLOW）] 画面が表示されます。

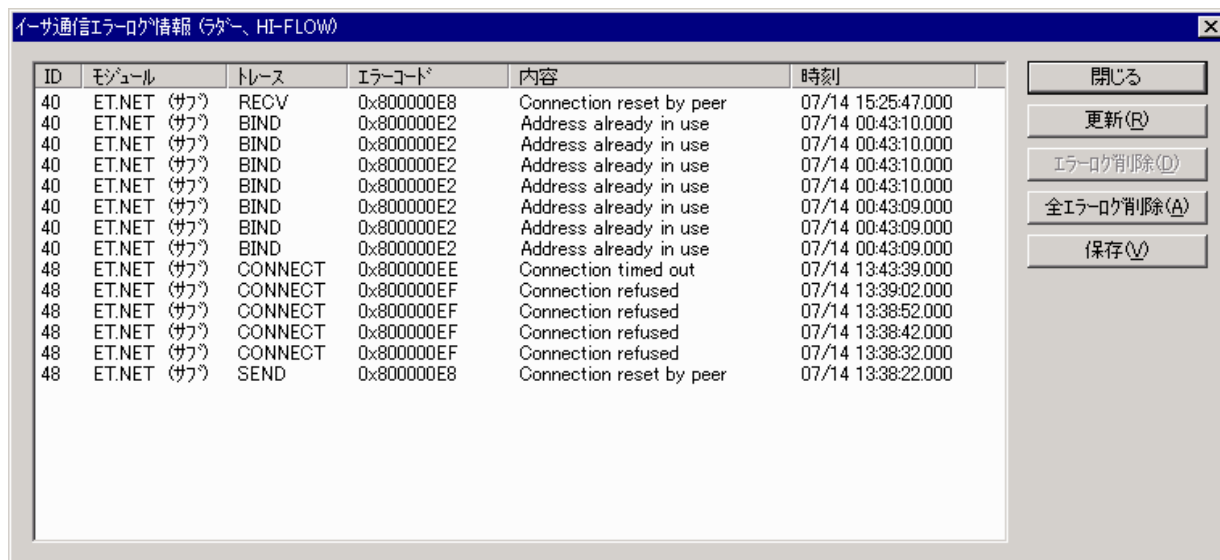


図 4-23 [イーサ通信エラーログ情報（ラダー、HI-FLOW）] 画面

- ③ エラーログ情報の表示内容を以下に示します。

項目	表示内容
ID	ラダー、HI-FLOWイーサネット通信の管理テーブル番号
モジュール	モジュール名称
トレース	トレース情報のトレースコード内容
エラーコード	発生したエラーのエラーコード
内容	発生したエラーのエラーコード内容
時刻	エラー発生時刻

- ④ 最新のエラーログ情報を表示する場合は、 **更新** ボタンをクリックしてください。
- ⑤ 指定したIDのエラーログ情報を削除する場合は、 **エラーログ削除** ボタンをクリックしてください。また、すべてのエラーログ情報を削除する場合は、 **全エラーログ削除** ボタンをクリックしてください。

## 4 オペレーション

- ⑥ 表示しているエラーログ情報の内容をテキストファイルに保存する場合は、**保存** ボタンをクリックして、[名前を付けて保存] 画面から保存するフォルダ、ファイル名を指定してください。

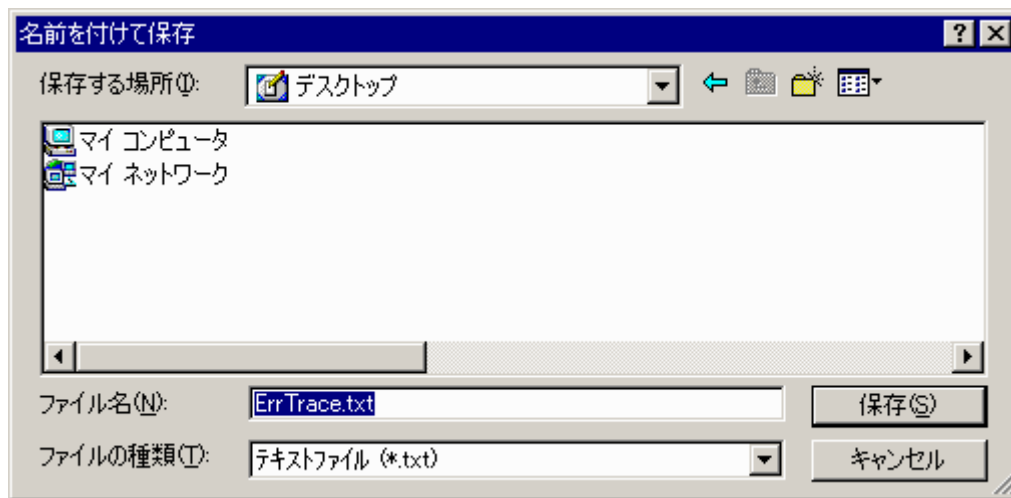


図 4-24 [名前を付けて保存] 画面 (イーサ通信エラーログ情報 (ラダー、HI-FLOW) 保存)

フォルダ、ファイル名を指定し、**保存** ボタンをクリックすることで、エラーログ情報がファイルに保存されます。

- ⑦ [イーサ通信エラーログ情報 (ラダー、HI-FLOW)] 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。

## 4. 3. 6 イーサ通信エラーログ情報（ソケットハンドラ）

機能：ソケットハンドラのイーサネット通信で発生したエラーのトレースログを表示します。

操作：以下に操作手順を示します。

- ① [モジュール設定] 画面または [モジュール検索設定] 画面の ソケットハンドラ ボタンをクリックしてください。
- ② [イーサ通信エラーログ情報（ソケットハンドラ）] 画面が表示されます。

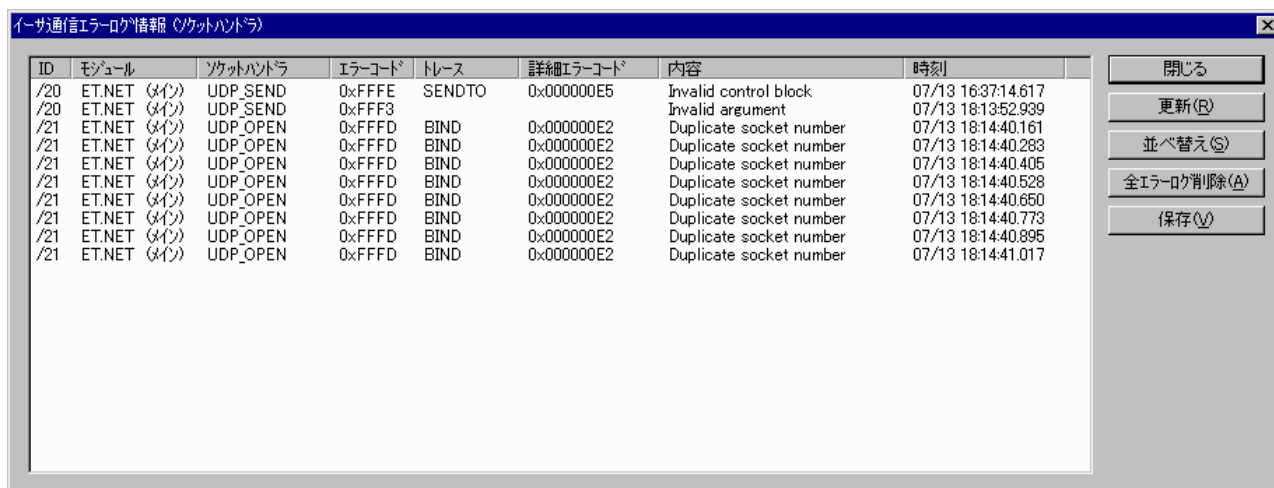


図 4-25 [イーサ通信エラーログ情報（ソケットハンドラ）] 画面

- ③ エラーログ情報の表示内容を以下に示します。

項目	表示内容
ID	ソケットハンドラのソケットID
モジュール	モジュール名称
ソケットハンドラ	ソケットハンドラの名称
エラーコード	ソケットハンドラのエラーコード
トレース	エラー検出箇所
詳細エラーコード	エラー検出時の詳細エラーコード
内容	発生したエラーのエラーコード内容
時刻	エラー発生時刻

- ④ 最新のエラーログ情報を表示する場合は、 更新 ボタンをクリックしてください。
- ⑤ エラーログ情報をエラー発生時刻で並び替える場合は、 並び替え ボタンをクリックしてください。ボタンをクリックするたびに発生時刻の昇順、降順に入れ替わります。

## 4 オペレーション

- ⑥ すべてのエラーログ情報を削除する場合は、**全エラーログ削除** ボタンをクリックしてください。
- ⑦ 表示しているエラーログ情報の内容をテキストファイルに保存する場合は、**保存** ボタンをクリックして、[名前を付けて保存] 画面から保存するフォルダ、ファイル名を指定してください。

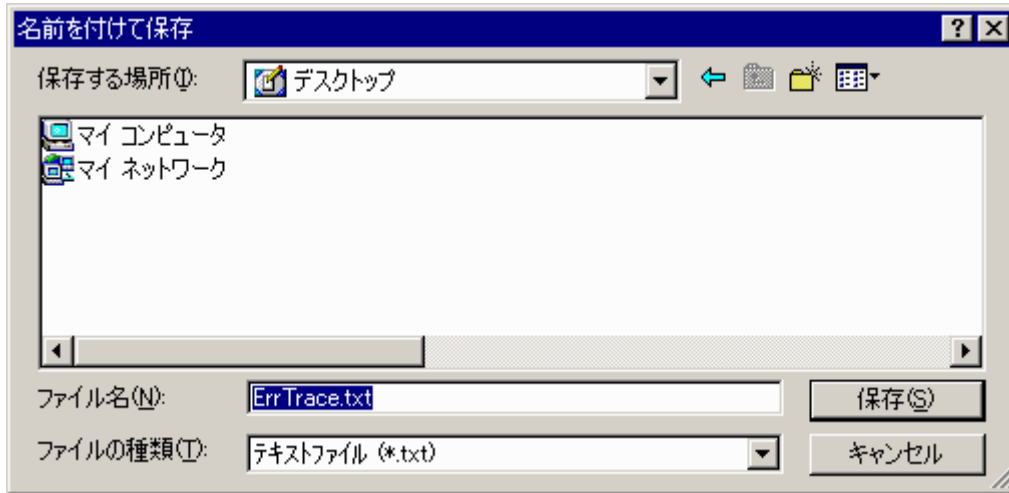


図 4-26 「名前を付けて保存」画面（イーサ通信エラーログ情報（ソケットハンドラ）保存）

フォルダ、ファイル名を指定し、**保存** ボタンをクリックすることで、エラーログ情報がファイルに保存されます。

- ⑧ [イーサ通信エラーログ情報（ソケットハンドラ）] 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。

### 4.3.7 DHP情報

機能：DHPの記録モードを設定します。

操作：以下に操作手順を示します。

- ① [モジュール設定] 画面または [モジュール検索設定] 画面の **DHP情報** ボタンをクリックしてください。
- ② [DHP情報] 画面が表示されます。

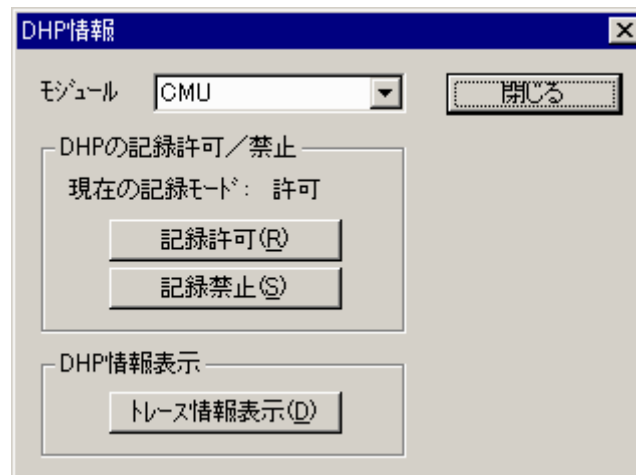


図 4-27 [DHP情報] 画面

- ③ [モジュール] ボックスに、PCsに実装されているCMUモジュールおよびET.NETモジュールの名称が表示されます。[モジュール] ボックスから、DHP情報の表示や設定を行うモジュールを選択してください。
- ④ [現在の記録モード] に、DHPの記録モードが表示されます。記録モードを「許可」に変更する場合は、**記録許可** ボタンをクリックしてください。また、記録モードを「禁止」に変更する場合は、**記録禁止** ボタンをクリックしてください。
- ⑤ DHPのトレース情報を表示する場合は、**トレース情報表示** ボタンをクリックしてください。
- ⑥ [DHP情報] 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。

## 4 オペレーション

### 4.3.8 DHPトレース情報

機能：DHPのトレース情報を表示します。

操作：以下に操作手順を示します。

- ① [DHP情報] 画面の トレース情報表示 ボタンをクリックしてください。
- ② [DHPトレース情報] 画面が表示されます。表示内容については、「7 保守」を参照してください。

DHP	TIME	EVENT	TN	LV	DATA1	DATA2	DATA3	DATA4	DATA5
1	59.075730	DHPREAD	287	03	7C0D0000	11D87028			
2	59.075689	GFACT	287	03	00000001				
3	59.075671	DISPATCH_E	287	03	0000011F	0000002B	846AA000	00000001	
4	59.075629	RUNQ	287	03	0000011F				
5	59.075617	QUEUE	287	03	0000011F	00000001			
6	59.075607	RLEAS	287	03	0000011F				
7	57.633121	IDLE	287	03					
8	57.633101	DISPATCH	287	03	0000011F	0000002B	846AA000		
9	57.633084	EXIT	287	03					
10	57.633055	DHPCTL	287	03	00000003	00000000	11D87080		
11	57.633004	GFACT	287	03	00000001				
12	57.632984	DISPATCH_E	287	03	0000011F	0000002B	846AA000	00000003	
13	57.632930	RUNQ	280	03	0000011F				
14	57.632919	QUEUE	280	03	0000011F	00000001			
15	57.632913	TASK_PRI	280	03	0000011F	0000002B			
16	57.632900	RLEAS	280	03	0000011F				
17	57.603330	IDLE	280	03					
18	57.603311	DISPATCH	280	03	00000118	0000002B	846A3000		
19	57.603292	EXIT	280	03					
20	57.603193	GFACT	280	03	00000001				

図 4-28 [DHPトレース情報] 画面

- ③ DHPトレース情報の表示内容を以下に示します。

項目	表示内容
DHP	DHPトレース番号の表示番号
TIME	トレース時刻 tt.tttttt ↑↑ 秒 1マイクロ秒まで出力
EVENT	トレースポイント種別
TN	タスク番号
LV	優先レベル
DATA1~DATA5	トレースデータ (16進数で出力)

- ④ 最新のDHPトレース情報を表示する場合は、 更新 ボタンをクリックしてください。

- ⑤ 表示しているDHPトレース情報の内容をテキストファイルに保存する場合は、**保存** ボタンをクリックして、[名前を付けて保存] 画面から保存するフォルダ、ファイル名を指定してください。

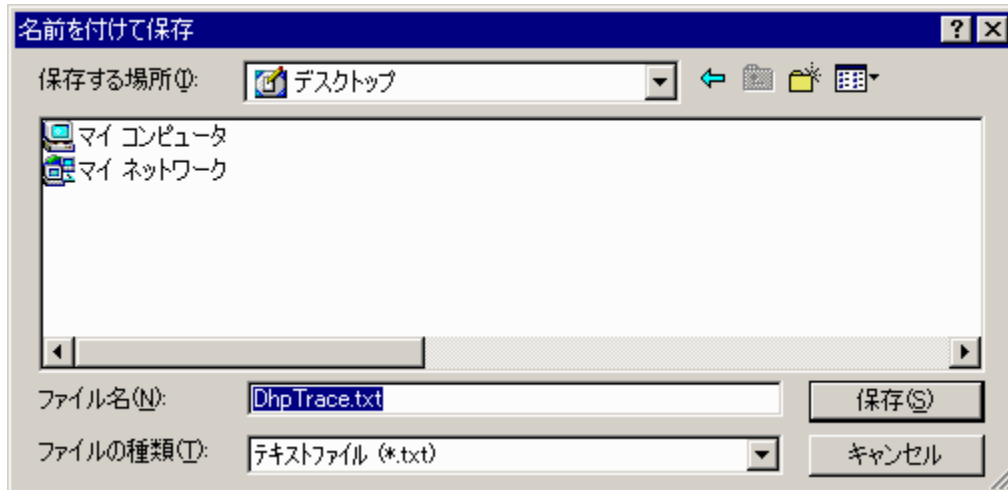


図 4-29 [名前を付けて保存] 画面 (DHPトレース情報保存)

フォルダ、ファイル名を指定し、**保存** ボタンをクリックすることで、DHPトレース情報がファイルに保存されます。

- ⑥ [DHPトレース情報] 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。



## 4 オペレーション

### 4.3.9 ネットワーク情報

機能：CMUモジュールおよびET.NETモジュールのネットワーク情報を表示します。

操作：以下に操作手順を示します。

- ① [モジュール設定] 画面または [モジュール検索設定] 画面の **ネットワーク情報** ボタンをクリックしてください。
- ② [ネットワーク情報] 画面が表示されます。

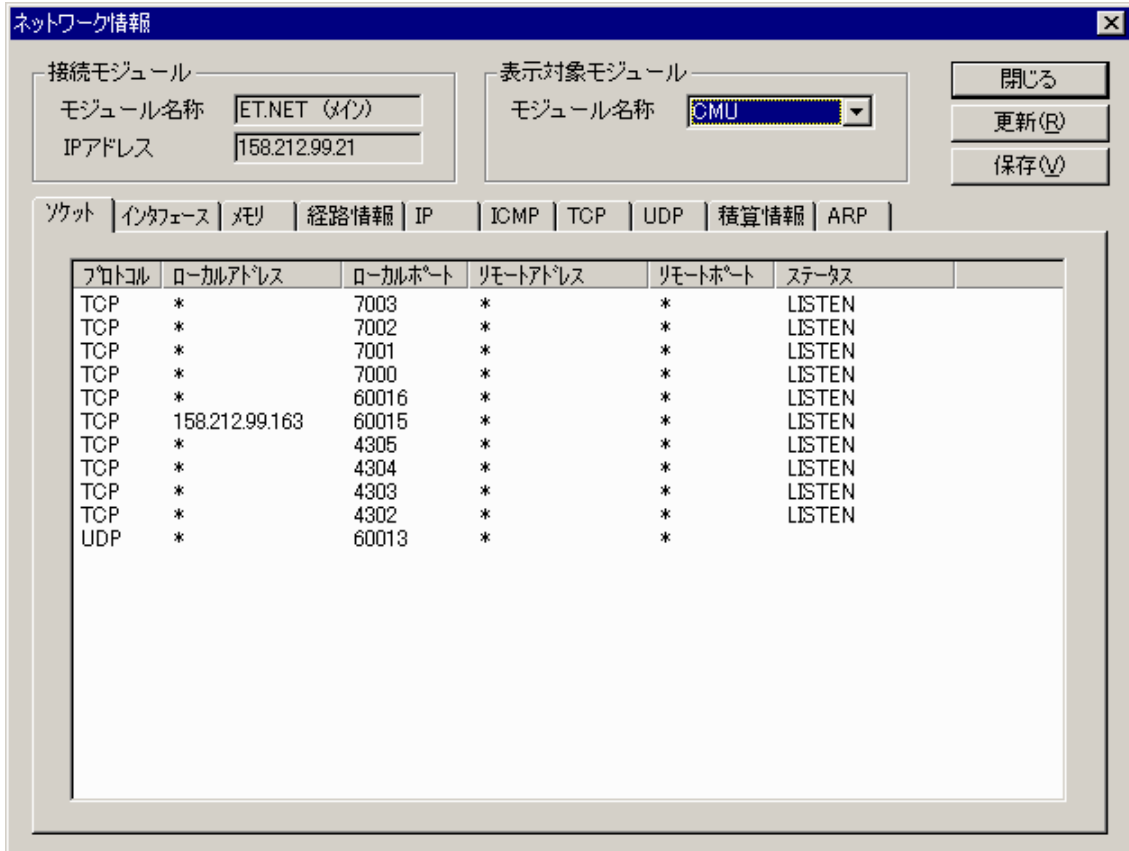


図4-30 [ネットワーク情報] 画面

- ③ [表示対象モジュール] の [モジュール名称] ボックスから、ネットワーク情報を表示するモジュールを選択してください。

- ④ タブを切り替えて表示したいネットワーク情報の種類を選択してください。  
ネットワーク情報は、以下の種類があります。

項目	内容
ソケット	ソケット情報を表示
インタフェース	動作中のネットワークインタフェース情報を表示
メモリ	送受信バッファ管理情報を表示
経路情報	経路情報を表示
IP	IPプロトコルの統計情報を表示
ICMP	ICMPプロトコルの統計情報を表示
TCP	TCPプロトコルの統計情報を表示
UDP	UDPプロトコルの統計情報を表示
積算情報	インタフェースの積算情報を表示
ARP	ARPテーブル情報を表示

- ⑤ **更新** ボタンをクリックすることで、指定した内容のネットワーク情報が表示されます。表示内容については、「7. 3. 3 ネットワーク情報の見方」を参照してください。
- ⑥ 表示しているネットワーク情報の内容をテキストファイルに保存する場合は、**保存** ボタンをクリックして、[名前を付けて保存] 画面から保存するフォルダ、ファイル名を指定してください。

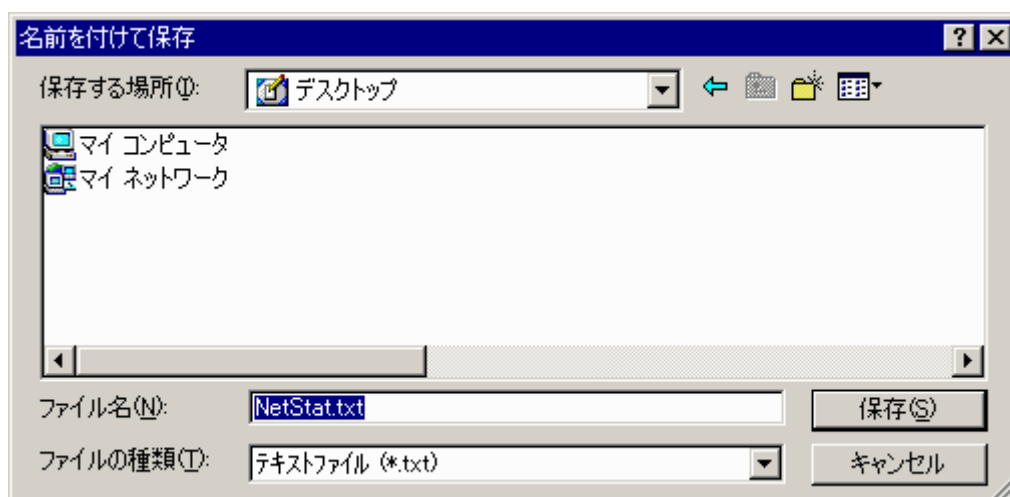


図 4-31 [名前を付けて保存] 画面 (ネットワーク情報保存)

フォルダ、ファイル名を指定し、**保存** ボタンをクリックすることで、ネットワーク情報がファイルに保存されます。

- ⑦ [ネットワーク情報] 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。

## 4 オペレーション

### 4. 3. 10 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。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. 2. 4 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。

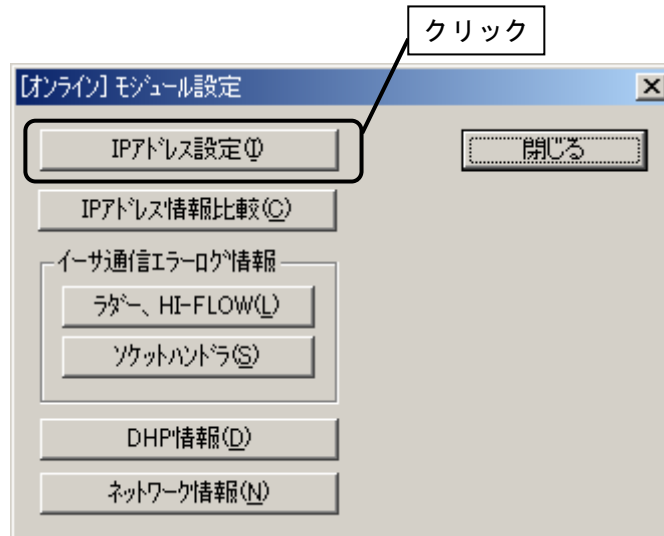


図 4-32 [[オンライン] モジュール設定] 画面 ( **IPアドレス設定** ボタンをクリック)

- 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 4-33 [[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 [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。

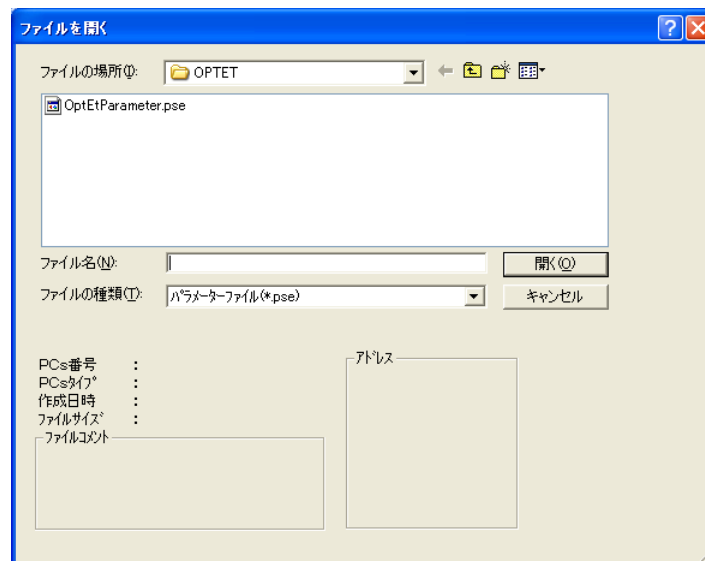


図 4-34 [ファイルを開く] 画面

- 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。

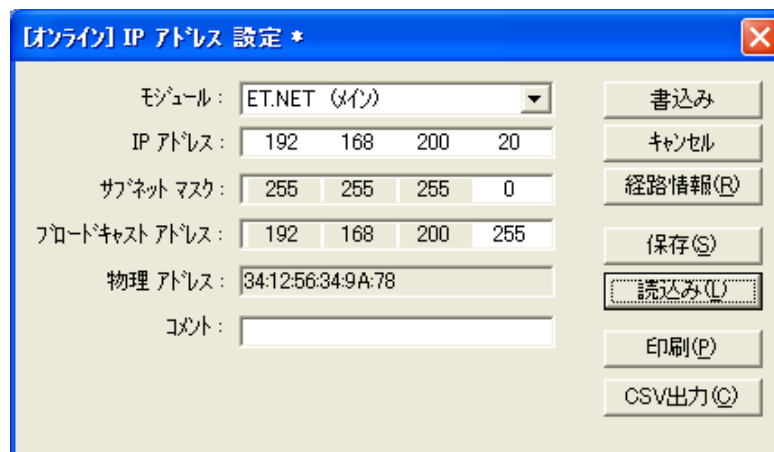


図 4-35 [[オンライン] IPアドレス設定] 画面 (選択したIPアドレス設定情報ファイルの内容表示)

[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. 3. 11 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. 2. 4 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。

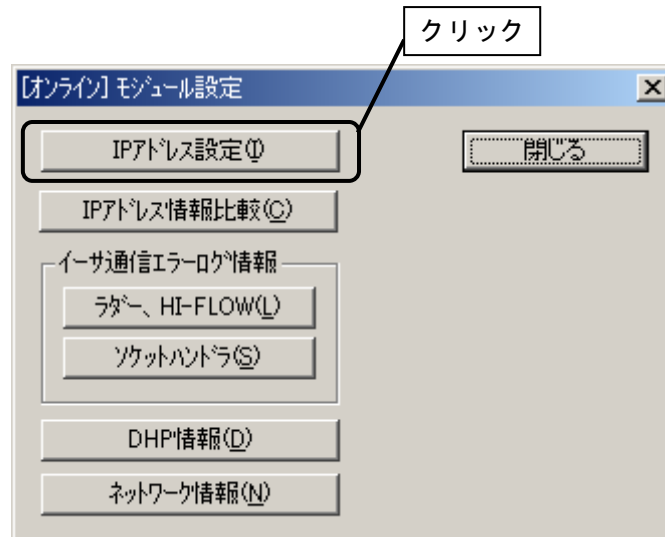


図 4-36 [[オンライン] モジュール設定] 画面（ **IPアドレス設定** ボタンをクリック）

- 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 4-37 [[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。

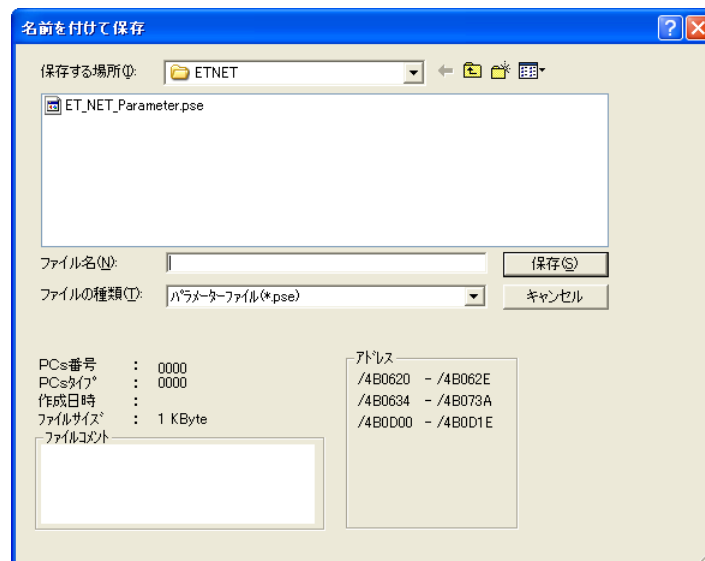


図 4-38 [名前を付けて保存] 画面

- 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. 3. 12 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. 2. 4 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。

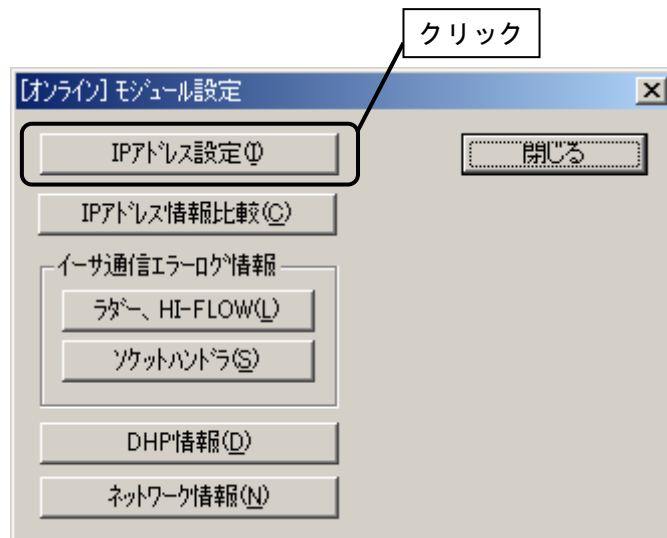


図 4-39 [[オンライン] モジュール設定] 画面（ **IPアドレス設定** ボタンをクリック）

- 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。

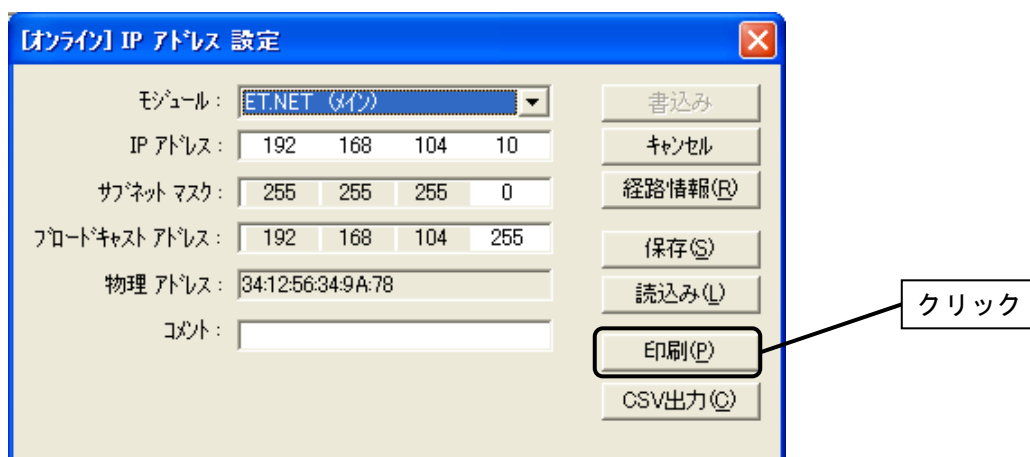


図 4-40 [[オンライン] IPアドレス設定] 画面（ **印刷** ボタンをクリック）

- 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。

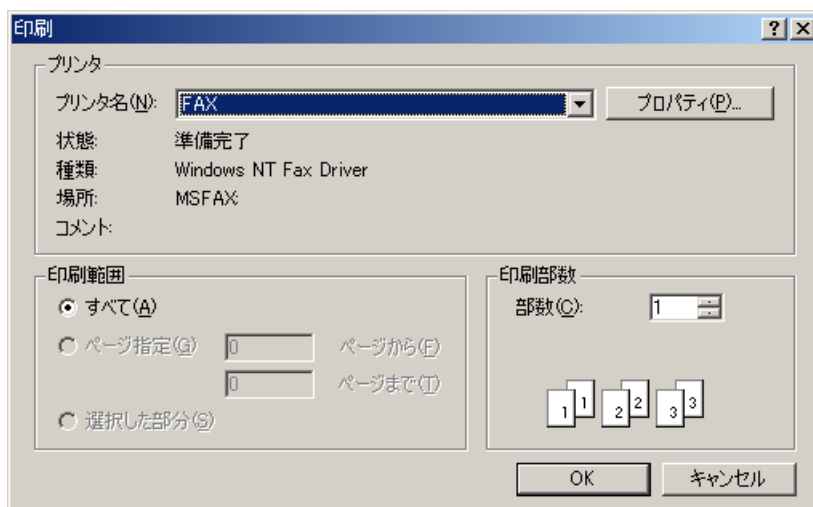


図 4-41 [印刷] ダイアログボックス

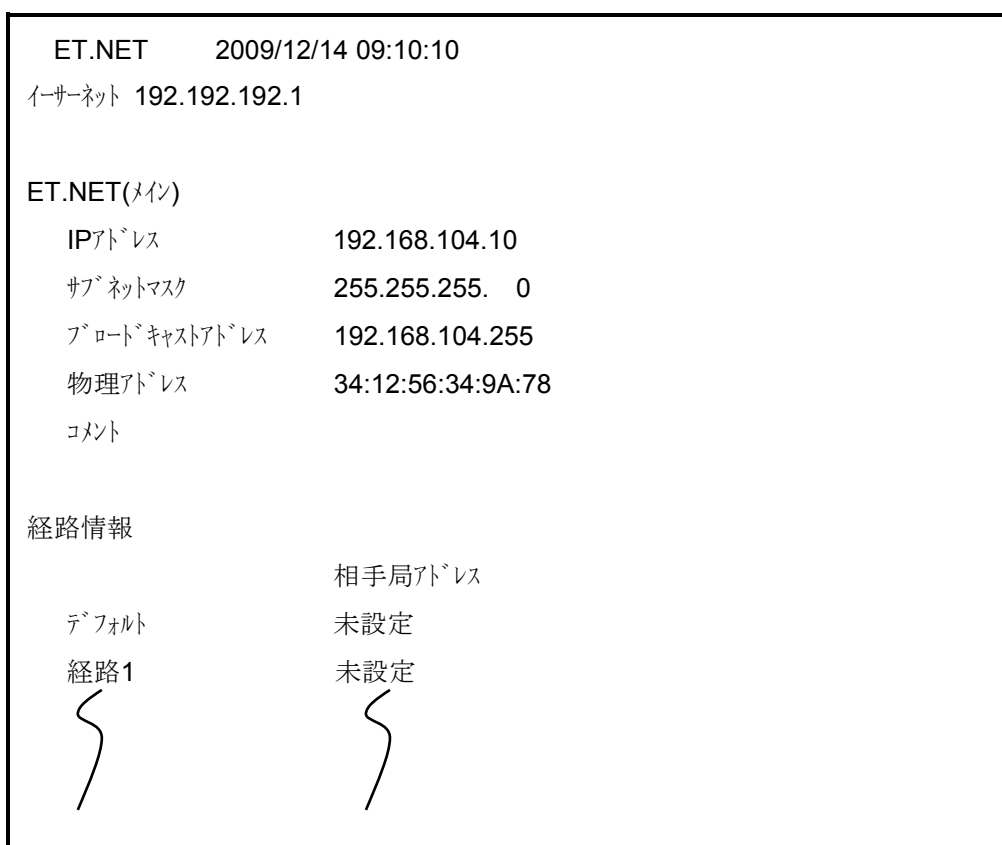


図 4-42 IPアドレス設定情報印刷例



## 4 オペレーション

### 4. 3. 13 IPアドレス設定情報のCSV出力

機能：表示中の設定情報を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. 2. 4 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。

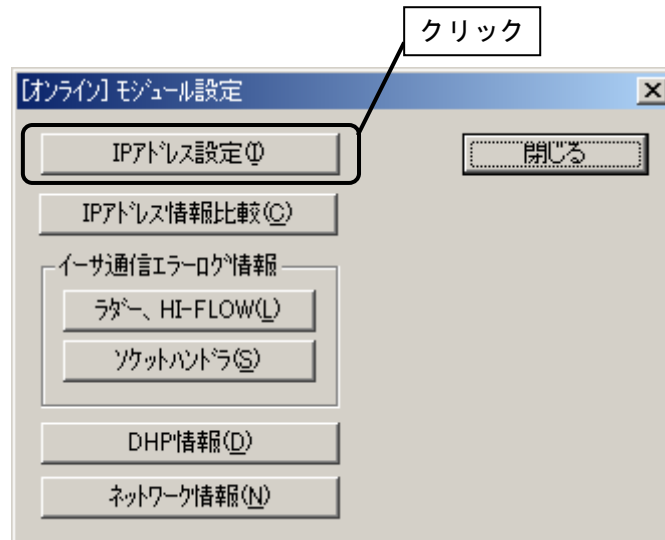


図 4-43 [[オンライン] モジュール設定] 画面（ **IPアドレス設定** ボタンをクリック）

- 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 4-44 [[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。

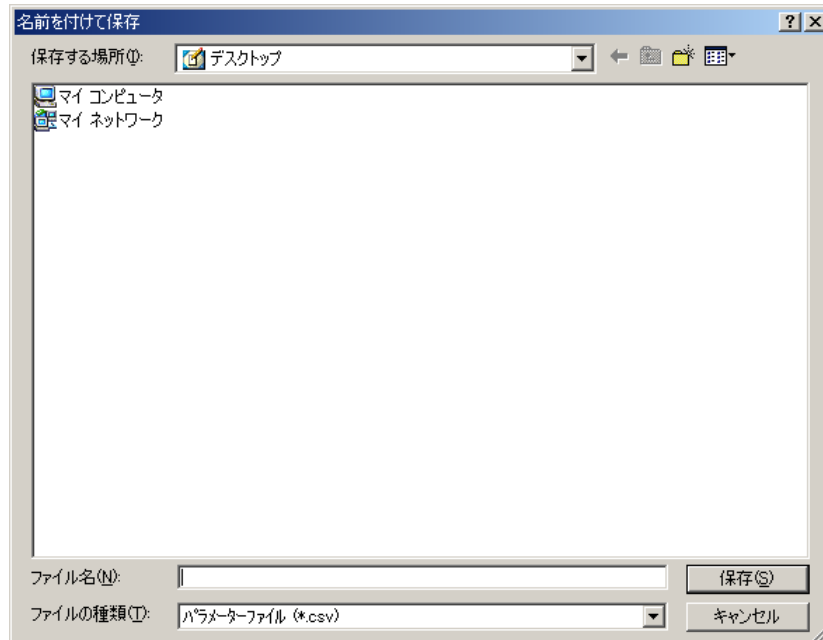


図 4-45 [名前を付けて保存] 画面 (CSV出力)

## 4 オペレーション

### 4. 3. 14 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. 2. 4 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。

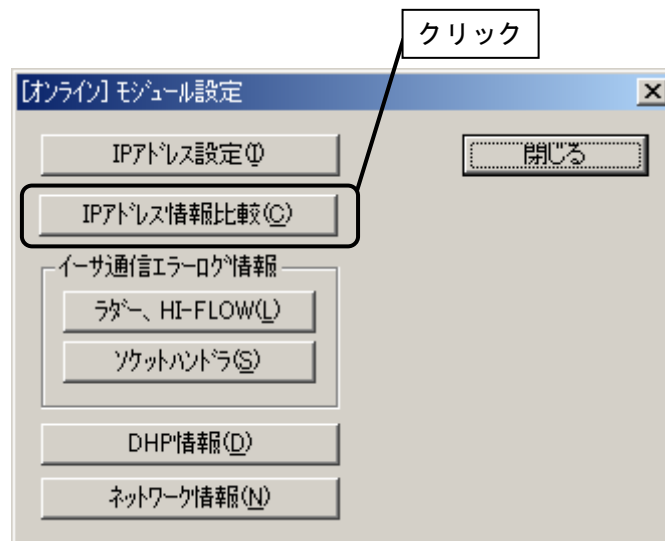


図 4-46 [[オンライン] モジュール設定] 画面

- 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。

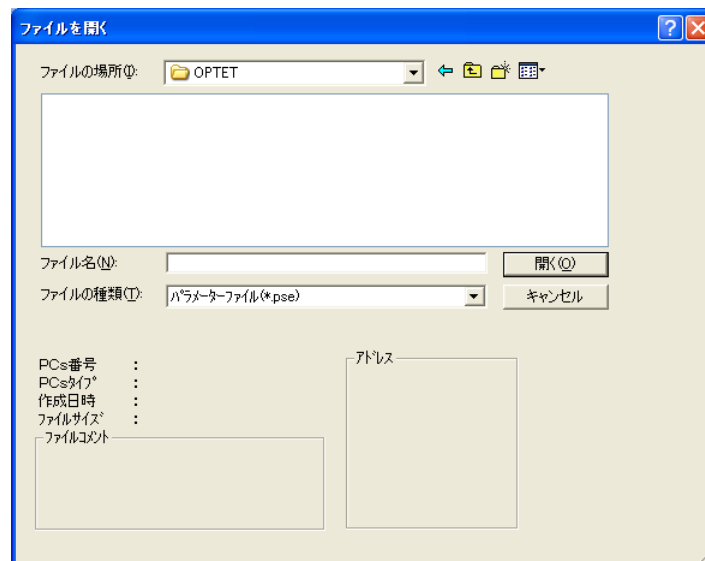


図 4-47 [ファイルを開く] 画面

- 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 4-48 「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 4-49 「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 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。

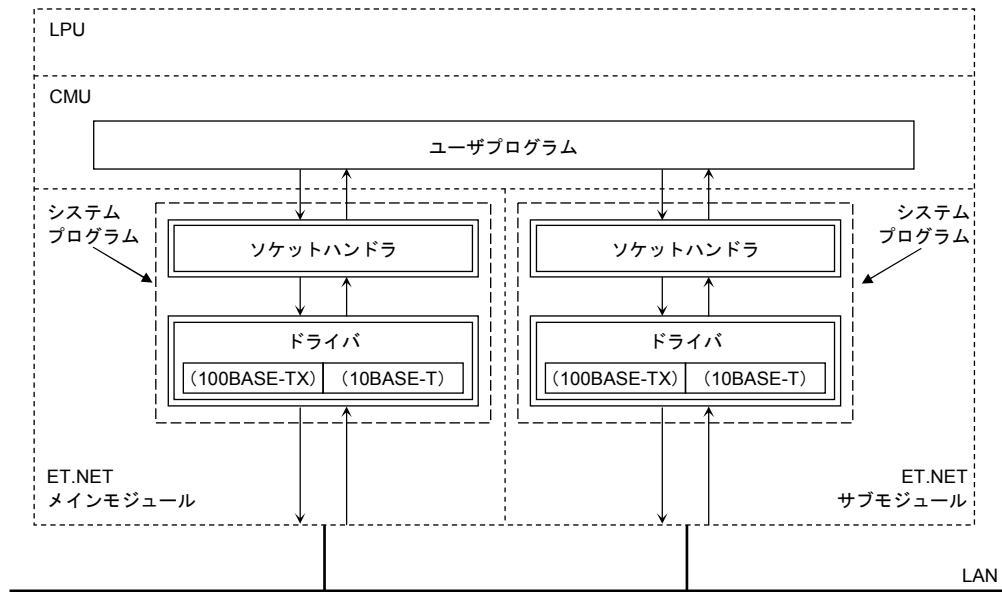


図 5-1 ET.NETのソフトウェア構成

## 5.2 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。

## 5.3 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。

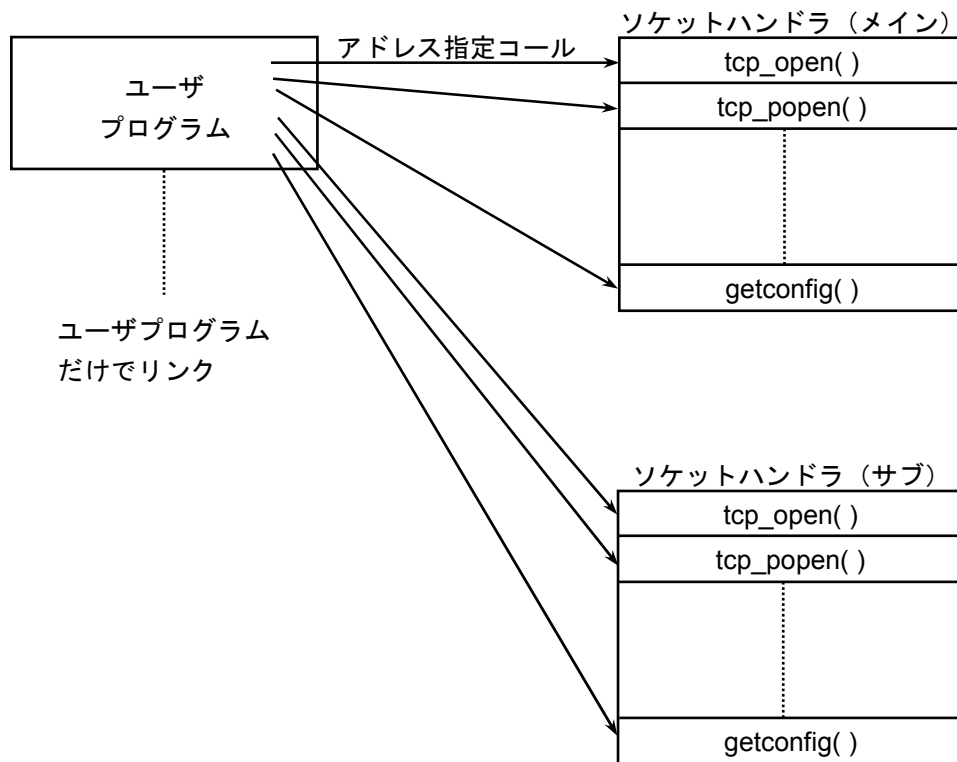


図5-2 ユーザプログラムからのソケットハンドラ呼び出し方法



## 5 プログラミング

### 5.3.1 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、LQE520、LQE720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		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

- 1つのモジュールで、同時に使用できるソケット数は、TCPとUDPで24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（ただし、TCPの60015, 60016およびUDPの60013, 60020はシステムが占有しているので使用できません）。
- 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、LPU内のOSテーブルに設定されます。LPUの交換をした場合は、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、モジュールに対してリセットまたは停復電をして、浮いたソケットを開放してください。

## 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、45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（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

(0とした場合、1024~2047までの任意のポート番号となります。)

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。

(/0001~/0018) : 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. 4 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]

## &lt;入力パラメータ詳細&gt;

padr : 入力パラメータの先頭アドレス (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。)

## &lt;出力パラメータ詳細&gt;

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。

(/0001～/0018) : 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. 4 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（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。

(/0001~/0018) : 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. 4 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]

## &lt;入力パラメータ詳細&gt;

padr : 入力パラメータの先頭アドレス（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

## &lt;出力パラメータ詳細&gt;

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. 4 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に対応した接続を強制終了（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（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. 4 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]

## &lt;入力パラメータ詳細&gt;

padr : 入力パラメータの先頭アドレス (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

## &lt;出力パラメータ詳細&gt;

outinf : 出力パラメータの先頭アドレス (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. 4 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に対応した接続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  recvwin; };          { 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  recvwin; };          { 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 (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 (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. 4 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に対応した接続にパラメータの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()で接続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 (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 (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 (全データ送信ウィンドウに格納)

(/F000~/FFFF) : エラー発生

エラーコードは、「7. 3. 4 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]

## &lt;入力パラメータ詳細&gt;

padr : 入力パラメータの先頭アドレス (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

(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))

## &lt;出力パラメータ詳細&gt;

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. 4 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、1024～2047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 (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～/0037) : 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. 4 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 (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. 4 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 (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

(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. 4 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 (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  
(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. 4 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17）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、登録されているエントリ数を返します。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]

## &lt;入力パラメータ詳細&gt;

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

## &lt;出力パラメータ詳細&gt;

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

---

### ■ データ長 (len) の指定について

指定したデータ長 (len) が、登録されているデータ長 (登録エントリ数×1エントリのサイズ (16バイト)) より小さい場合は、指定したデータ長分のみの経路情報が取得できません (リターン値も取得したエントリ数が返ります)。1エントリのサイズより小さい値を指定した場合は、リターン値に0が返ります。

登録されている全経路情報を取得したい場合は、データ長 (len) に0を指定してroute\_list()を発行し、登録されているエントリ数を取得した後、登録エントリ数×1エントリのサイズ (16バイト) をデータ長に指定してroute\_list()を発行してください。

または、最大登録エントリ数 (17エントリ) ×1エントリのサイズ (16バイト) をデータ長に指定してroute\_list()を発行してください。

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 (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. 4 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

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。経路情報テーブルが満杯で登録できない場合は、内部バッファ不足エラー（エラーコード：/FFFF）が返りますので、不要な経路情報をroute\_del()にて削除後に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 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（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. 4 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。

## 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、登録されているエントリ数を返します。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 lstarp_p{   short  len;   short  notuse;   void   *buf; };       { short  (*arp_list)(); short  rtn; struct  lstarp_p  *padr;       {       arp_list = ( short(*) ( ) )0x87418A;       {       rtn = ( *route_arp )( padr );       { </pre>	<pre> struct lstarp_p{   short  len;   short  notuse;   void   *buf; };       { short  (*arp_list)(); short  rtn; struct  lstarp_p  *padr;       {       arp_list = ( short(*) ( ) )0x8F418A;       {       rtn = ( *route_arp )( padr );       { </pre>

## [パラメータ]

## &lt;入力パラメータ詳細&gt;

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

## &lt;出力パラメータ詳細&gt;

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

```

## 5 プログラミング

---

### ■ データ長 (len) の指定について

指定したデータ長 (len) が、登録されているデータ長 (登録エントリ数×1エントリのサイズ (12バイト)) より小さい場合は、指定したデータ長分のみのARP情報が取得できます (リターン値も取得したエントリ数が返ります)。1エントリのサイズより小さい値を指定した場合は、リターン値に0が返ります。

登録されている全ARP情報を取得したい場合は、データ長 (len) に0を指定してarp\_list()を発行し、登録されているエントリ数を取得した後、登録エントリ数×1エントリのサイズ (12バイト) をデータ長に指定してarp\_list()を発行してください。

または、最大登録エントリ数 (32エントリ) ×1エントリのサイズ (12バイト) をデータ長に指定してarp\_list()を発行してください。

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_p{     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_p{     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 (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. 4 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。ARP情報テーブルが満杯で登録できない場合は、内部バッファ不足エラー（エラーコード：/FFFF）でリターンしますので、不要なARP情報をarp\_del( )にて削除後に、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 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（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. 4 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 (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、下記のデータ構造となります。

```

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                         (24)
  char    udp_num   : 最大UDPソケット数                         (24)
  char    rt_num    : 経路情報テーブルサイズ                   (17)
  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             (4096)
};

```

5.4 ソケットハンドラ発行手順例

5.4.1 TCP/IPプログラム使用例

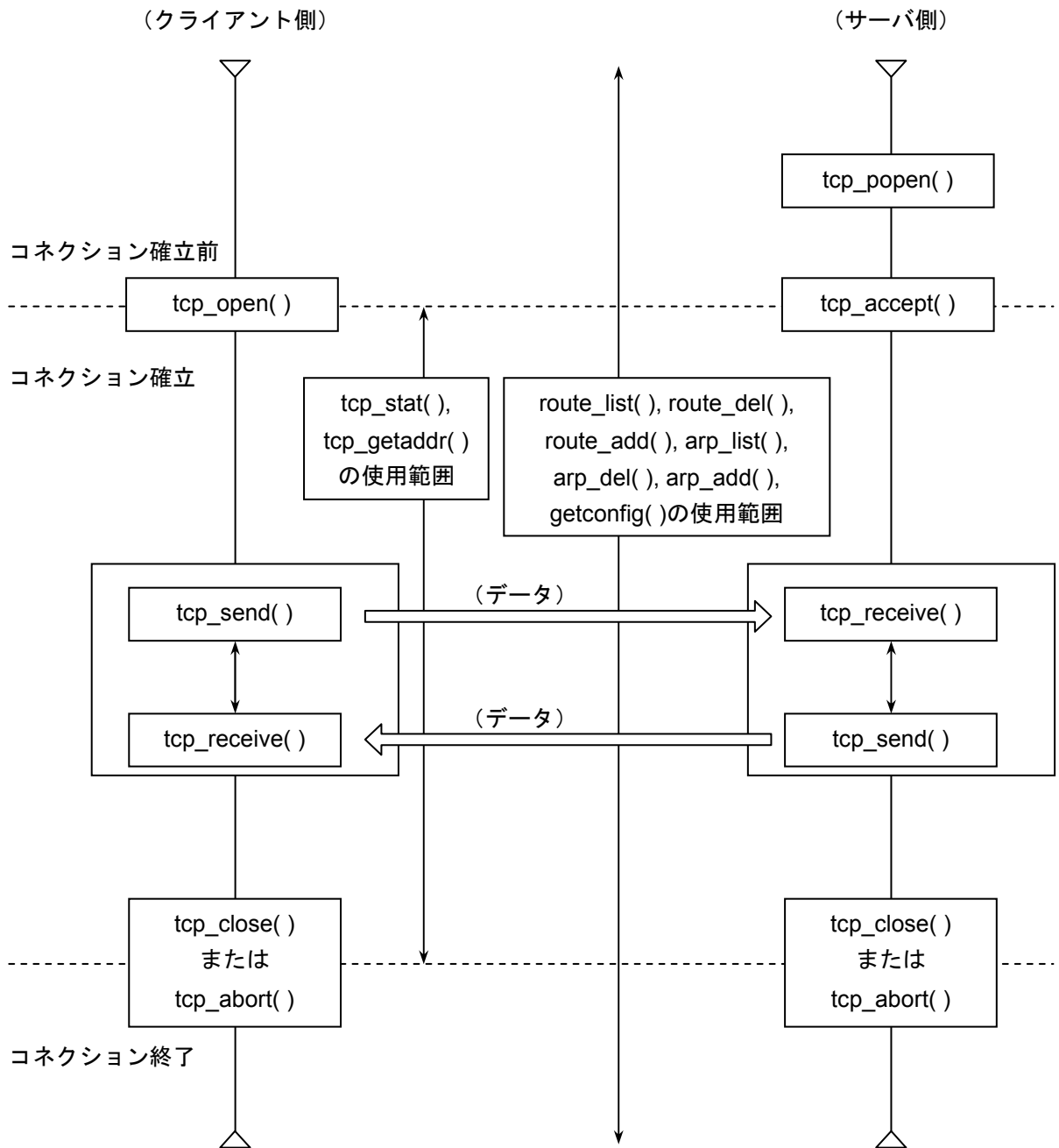


図5-3 TCP/IPプログラム使用時のソケットハンドラ発行手順例

## 5.4.2 UDP/IPプログラム使用例

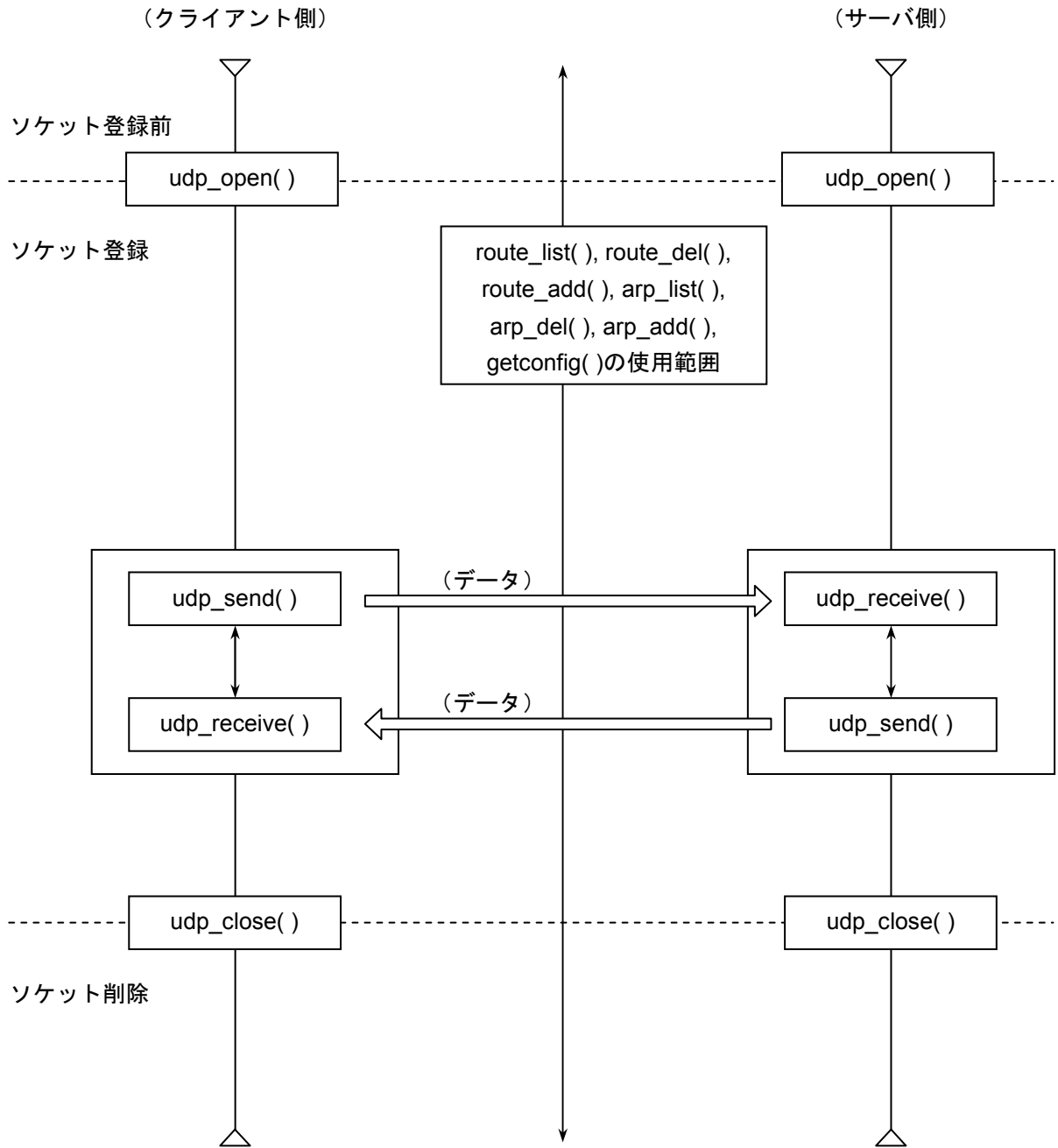
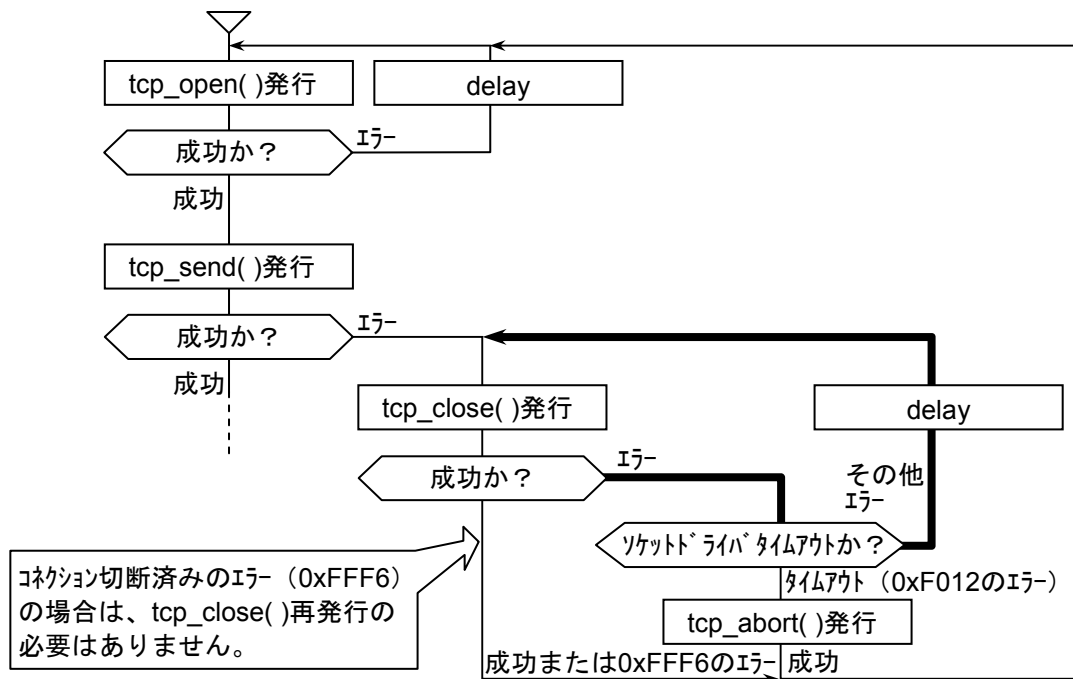


図5-4 UDP/IPプログラム使用時のソケットハンドラ発行手順例

## 5 プログラミング

### ・ 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( )のリターン値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。 LQE720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	

## 5.5 CPU間通信プログラム例

### 5.5.1 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。

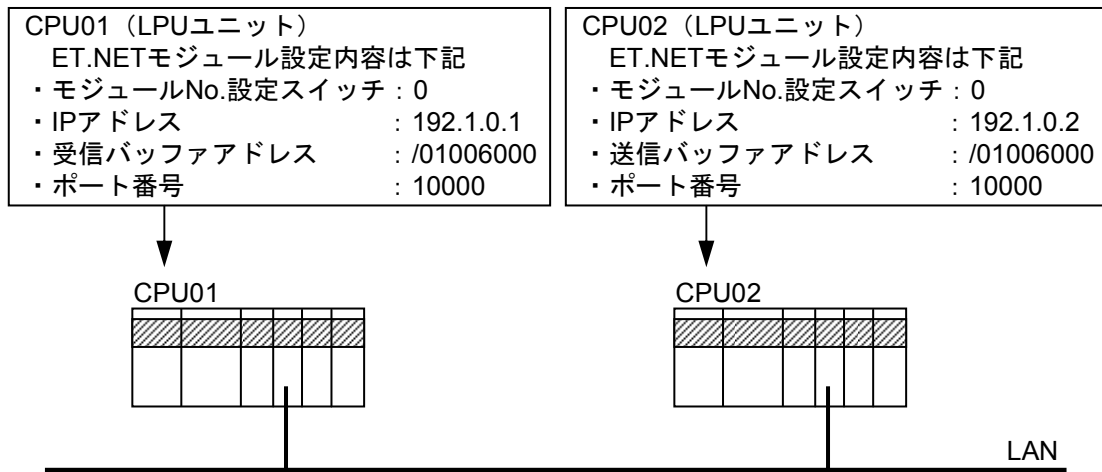


図5-5 CPU間通信プログラムのシステム構成例

## 5.5.2 CPU01側プログラムのフローチャートとプログラム例

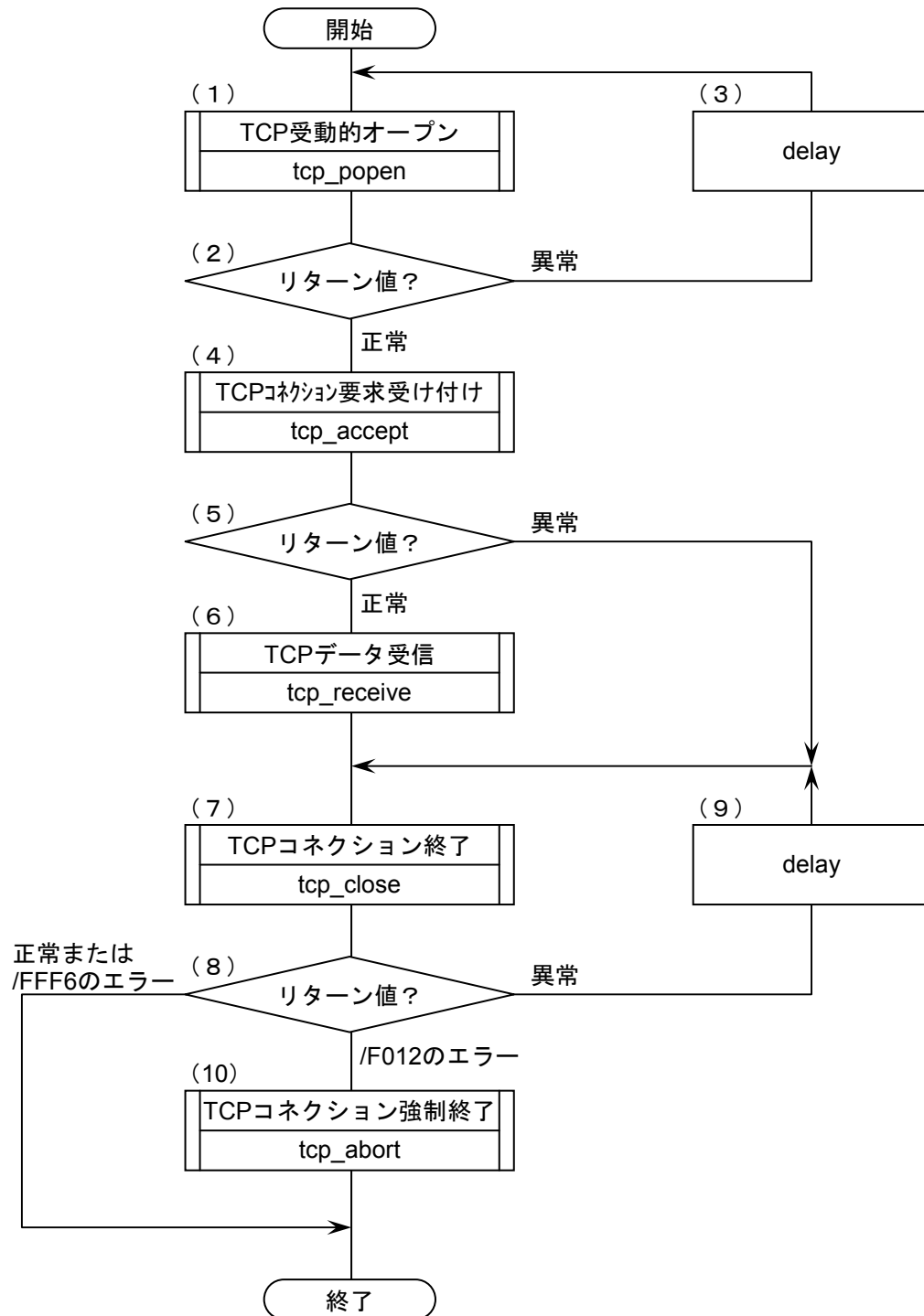


図5-6 CPU01側プログラムのフローチャート



## 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 (メイン) */
#define IPADDR        0xC0010002L   /* 相手局IPアドレス */
#define RBUFADDR      0x01006000L   /* 受信バッファ先頭アドレス */

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    listennum;     /* 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:対向機 (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;
    struct  popen_p    popen;
    struct  accept_p   accpt;
    struct  receive_p  recv;
    struct  close_p    close;

```

## 5 プログラミング

```
struct abort_p abort;
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.listennum   = 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);
}
accept.s_id = rtn;                      /* ソケットID */
tcp_accept  = ( short (*) ( ) ) TCP_ACCEPT;
rtn         =(tcp_accept) (&accept);   /* 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 = 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. 5. 3 CPU02側プログラムのフローチャートとプログラム例

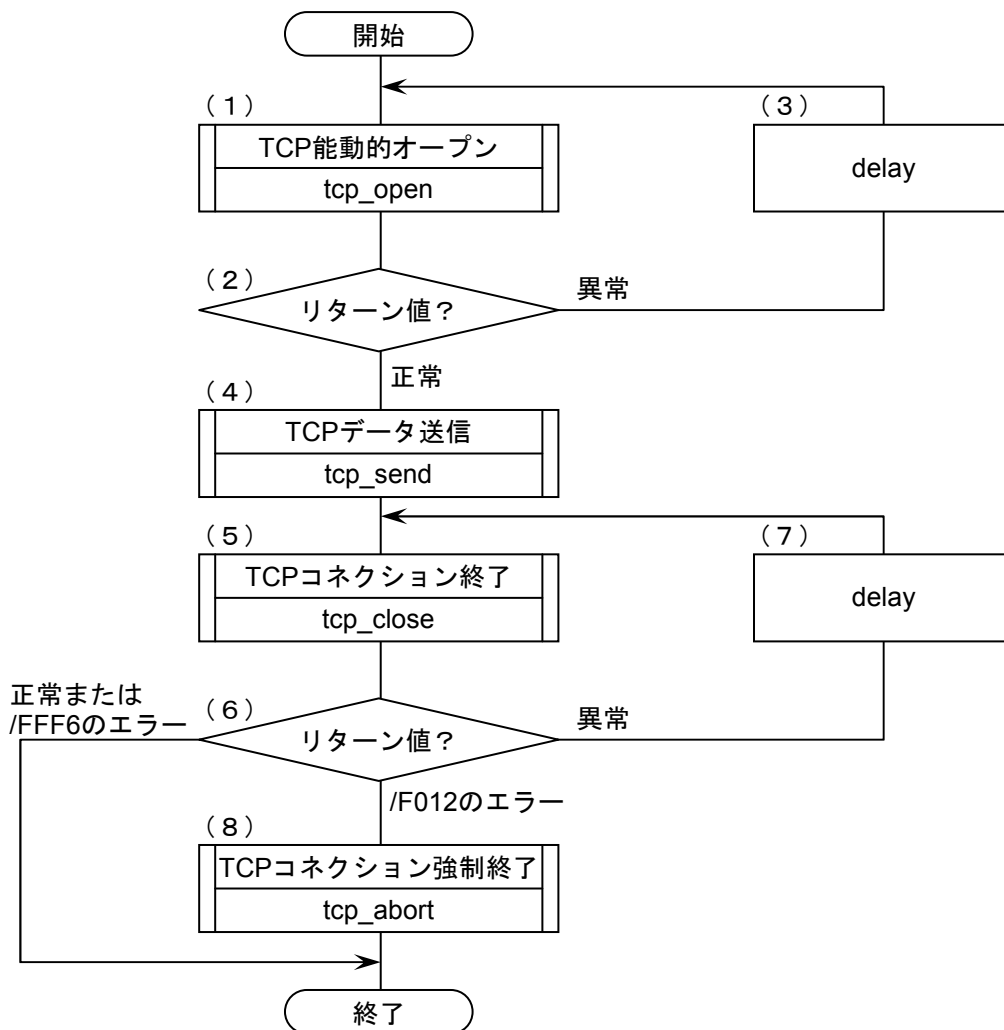


図5-7 CPU02側プログラムのフローチャート

## ■ 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 (メイン) */
#define IPADDR       0xC0010001L   /* 相手局IPアドレス */
#define SBUFADDR     0x01006000L   /* 送信バッファ先頭アドレス */

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;

    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.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 */
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 プログラミング

### 5.6 CPU間連続通信プログラム例

#### 5.6.1 システム構成およびプログラム構成

図5-8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。

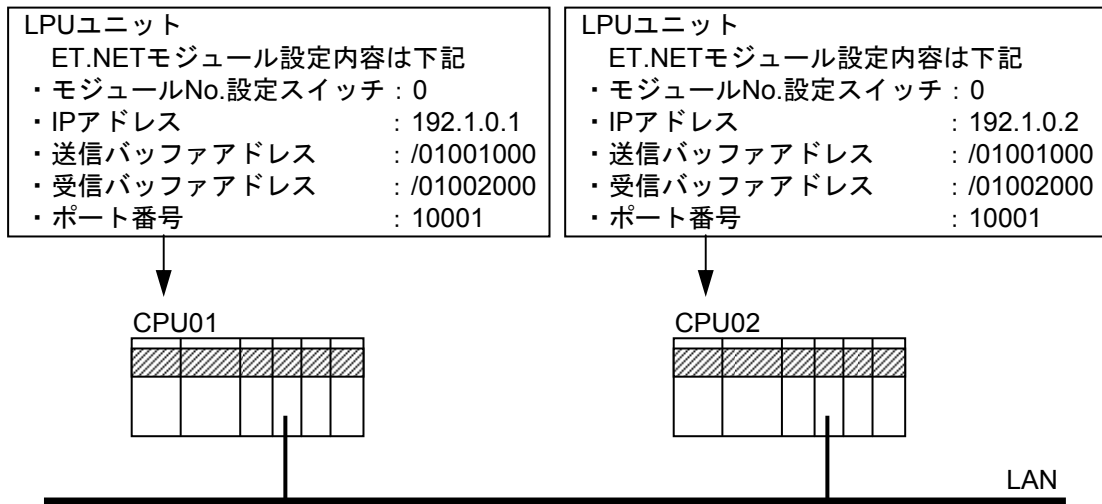


図5-8 CPU間連続通信プログラムのシステム構成例

## 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

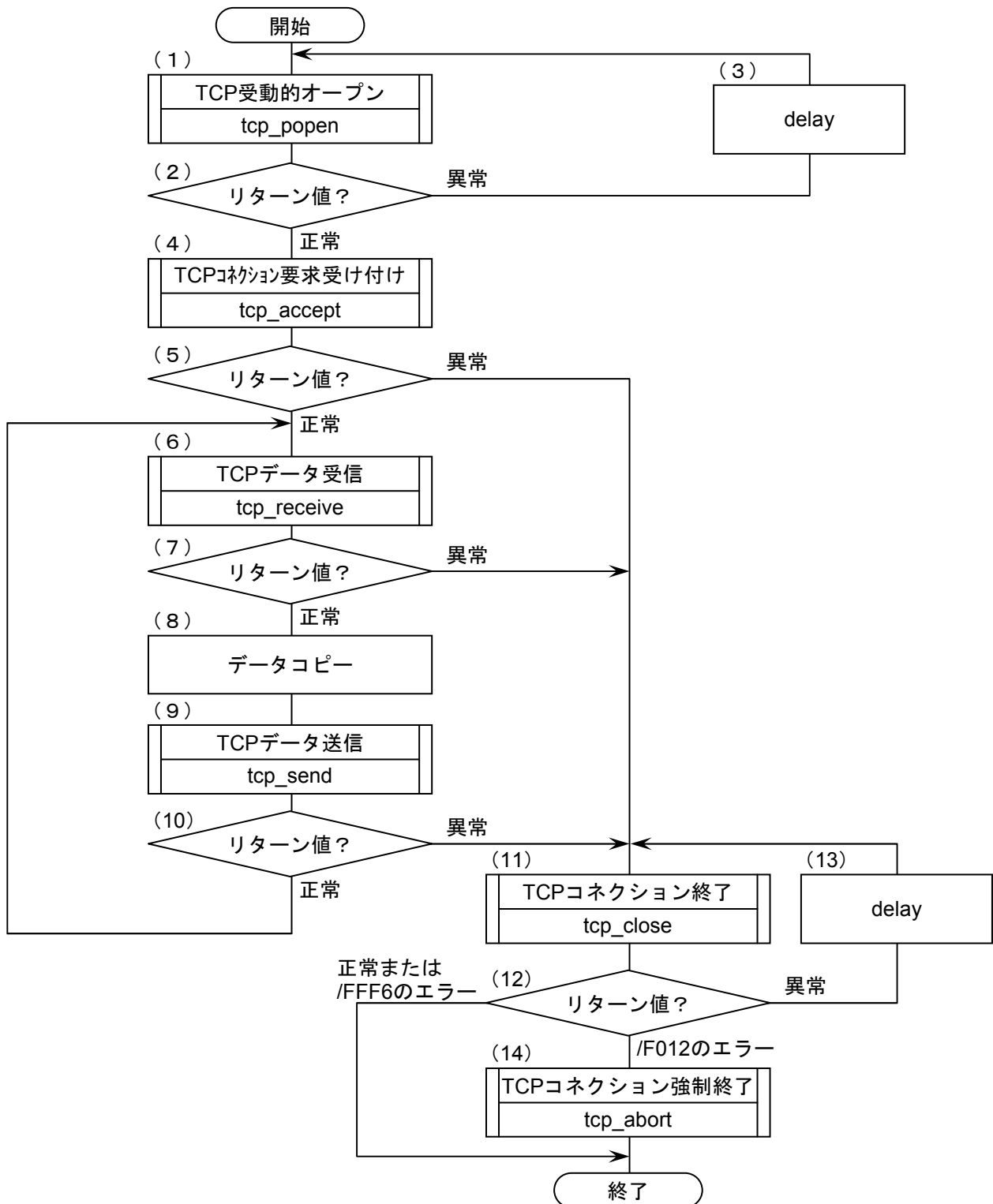


図5-9 CPU01側プログラムのフローチャート



## 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 (メイン) */
#define IPADDR         0xC0010002L    /* 相手局IPアドレス */
#define SBUFADDR       0x01001000L    /* 送信バッファ先頭アドレス */
#define RBUFADDR       0x01002000L    /* 受信バッファ先頭アドレス */

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    listennum;      /* 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:サハ(CPU01) */
/*****/
main()
{
    register short (*tcp_popen) ();
    register short (*tcp_accept) ();
    register short (*tcp_receive) ();
    register short (*tcp_send) ();
    register short (*tcp_close) ();
    register short (*tcp_abort) ();

```

## 5 プログラミング

```
long    time;
short   rtn, i;
char    *sbuf, *rbuf;
struct  popen_p    popen;
struct  accept_p   accept;
struct  receive_p  recv;
struct  send_p     send;
struct  close_p    close;
struct  abort_p    abort;

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.listennum   = 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);
}
accept.s_id        = rtn;                /* ソケットID              */
tcp_accept         = ( short (*) ( ) ) TCP_ACCEPT;
rtn                = (tcp_accept) (&accept); /* 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

### 5.6.3 CPU02側プログラムのフローチャートとプログラム例

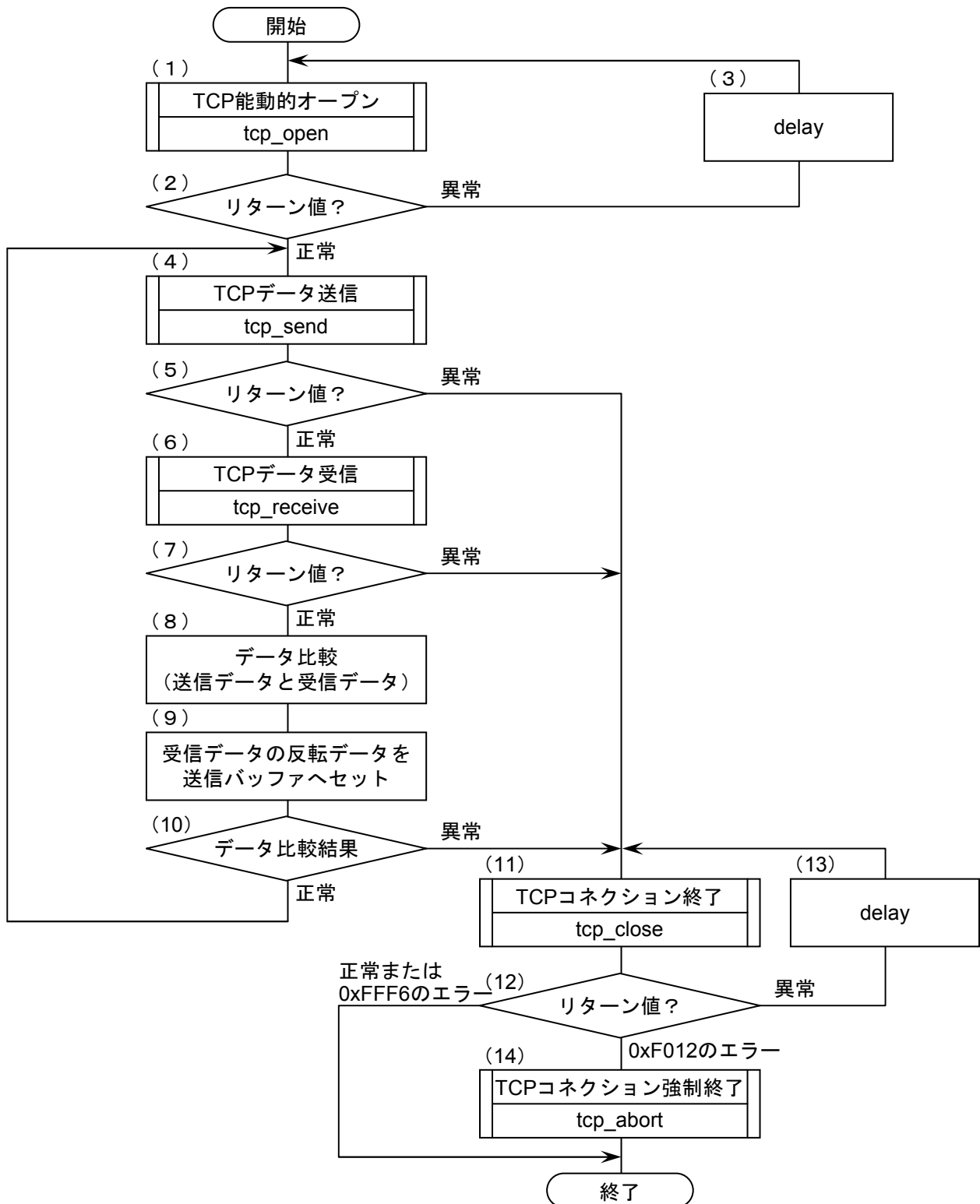


図5-10 CPU02側プログラムのフローチャート

## ■ 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 (メイン) */
#define IPADDR          0xC0010001L   /* 相手局IPアドレス */
#define SBUFADDR        0x01001000L   /* 送信バッファ先頭アドレス */
#define RBUFADDR        0x01002000L   /* 受信バッファ先頭アドレス */

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;
    char *sbuf, *rbuf;
    struct open_p open;
    struct send_p send;
```

```

struct      receive_p      recv;
struct      close_p        close;
struct      abort_p        abort;

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データ送信 */
    if( rtn < 0 ) {                 /* リターン値異常? */
        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) */
    tcp_receive = ( short (*) ( ) ) TCP_RECEIVE;
    rtn         = (tcp_receive) (&recv); /* TCPデータ受信 */
    if( rtn < 0 ) {                 /* リターン値異常? */
        break;
    }
    cerr_flg = 0;                   /* コンパ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;           /* コンパエラーフラグセット */
            break;
        }
        sbuf[ i ] = ~rbuf[ i ];     /* 反転データセット */
    }
    if( cerr_flg == 1 ) {           /* コンパエラー? */
        break;
    }
}
}

```



## 5 プログラミング

---

```
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 )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;
}
```

## 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

LQE720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、LQE720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。LQE720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	LQE720	
②	ハブ	(株)日立製作所	H-7612-90	スイッチングハブ
③	ツイストペアケーブル	日立金属(株)	HUTP-CAT5E-4P***	***はケーブル長

## 6.2 システム構成

図6-1のように、ハブに複数のイーサネット機器を接続することによりイーサネット機器間で通信することができます。ハブにイーサネット機器を接続するには、ツイストペアケーブルを使用します。

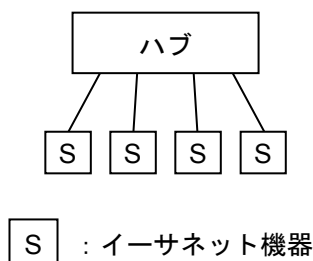


図6-1 ハブの接続構成

### ■ 10Mbps専用ハブ使用時の構成

- ・10Mbps専用ハブで多段接続して使用する場合は、任意のイーサネット機器間の経路において、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。

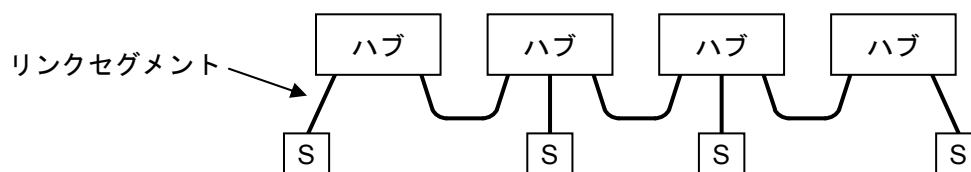


図6-2 10Mbps専用ハブのみによる構成例

## 6 利用の手引き

### ■ 100Mbpsハブ使用時の構成

- 100Mbpsハブ クラス1のハブ使用時は、ハブの多段接続はできません。  
下図のようにハブに接続してください。

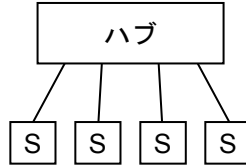


図 6-3 100Mbpsハブ クラス1による構成例

- 100Mbpsハブ クラス2のハブ使用時は、ハブ2段まで接続できます。  
下図のようにハブを接続して使用ください。  
(ハブ間の接続ケーブルの長さは5m以下としてください。)

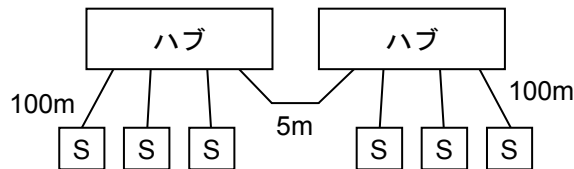


図 6-4 100Mbpsハブ クラス2による構成例

- 100Mbpsスイッチングハブ使用時は、基本的にカスケード接続の段数制限はありませんが、スイッチングハブのマニュアルを確認のうえ接続してください。

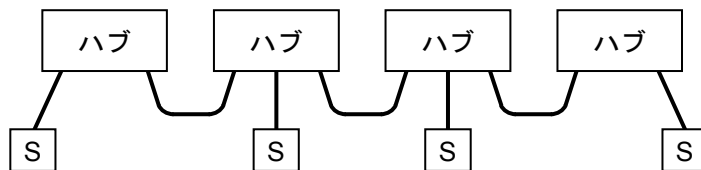


図 6-5 100Mbpsスイッチングハブによる構成例

### 通 知

このET.NETモジュール (LQE720) はオートネゴシエーション設定のみのサポートとなっております。

スイッチングハブのポート設定は100Mbpsフルデュプレクス設定はしないでください。

回線負荷が上がると正常に通信できないことがあります。

## 6.3 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.NET1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.NET1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.NET1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。

### 6.3.1 物理アドレス

ET.NET1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。

### 6.3.2 IPアドレス

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

- (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。)

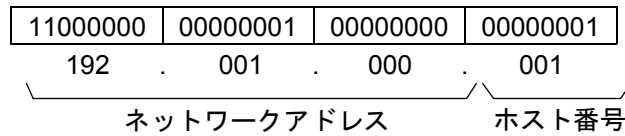
ネットワーク番号 (16ビット)	ホスト番号 (16ビット)
---------------------	---------------

- (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)
---------------------	--------------

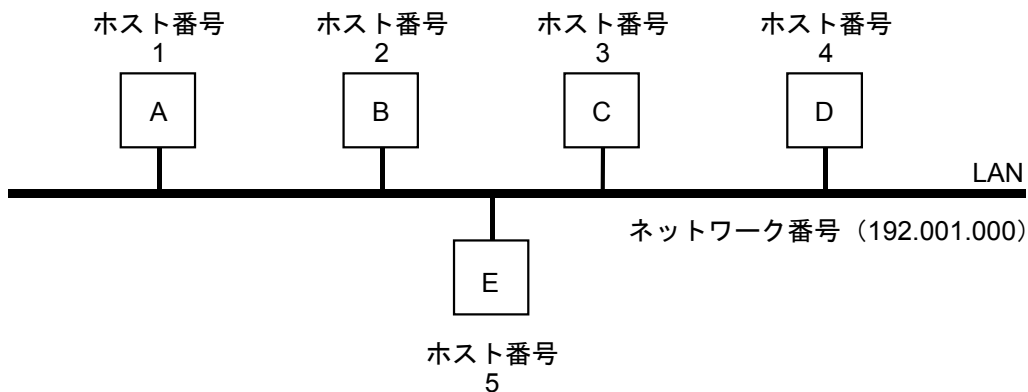
## 6 利用の手引き

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通信より行ってください）。

### 6.3.3 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

### 6.3.4 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モジュール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。

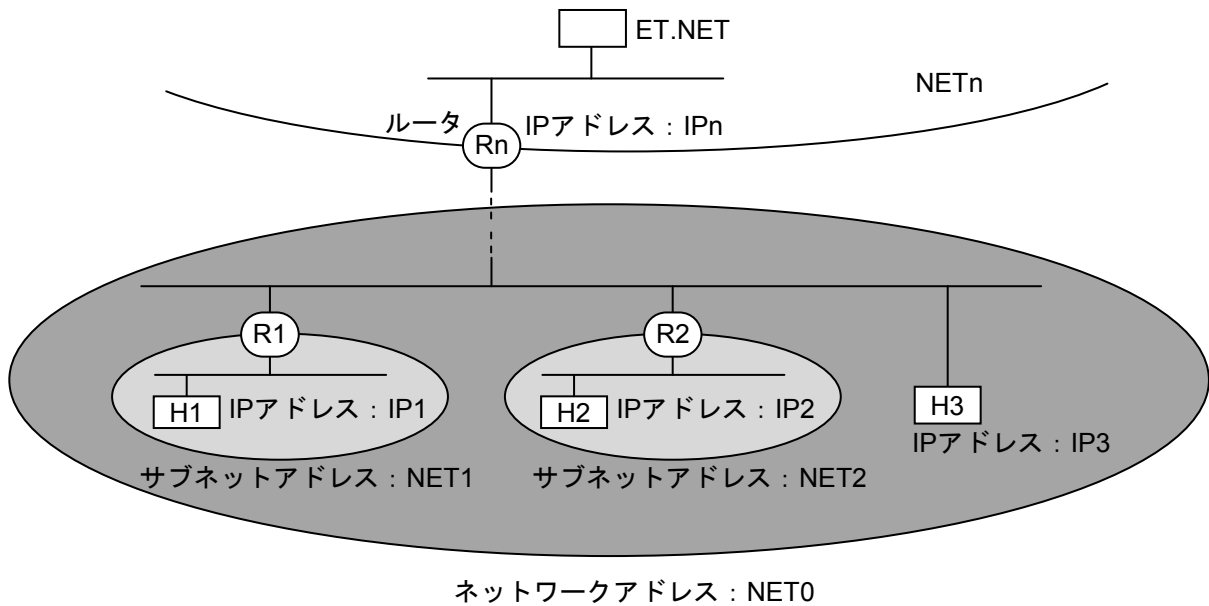
- C言語プログラムでソケットハンドラroute\_add()にて設定する方法  
「5.3.1 ソケットハンドラー一覧」を参照してください。
- ET.NETシステムまたはS10V ET.NETシステムを使用して設定する方法  
「4.3.4 経路情報」を参照してください。



## 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。

# 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。

通 知

- 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。

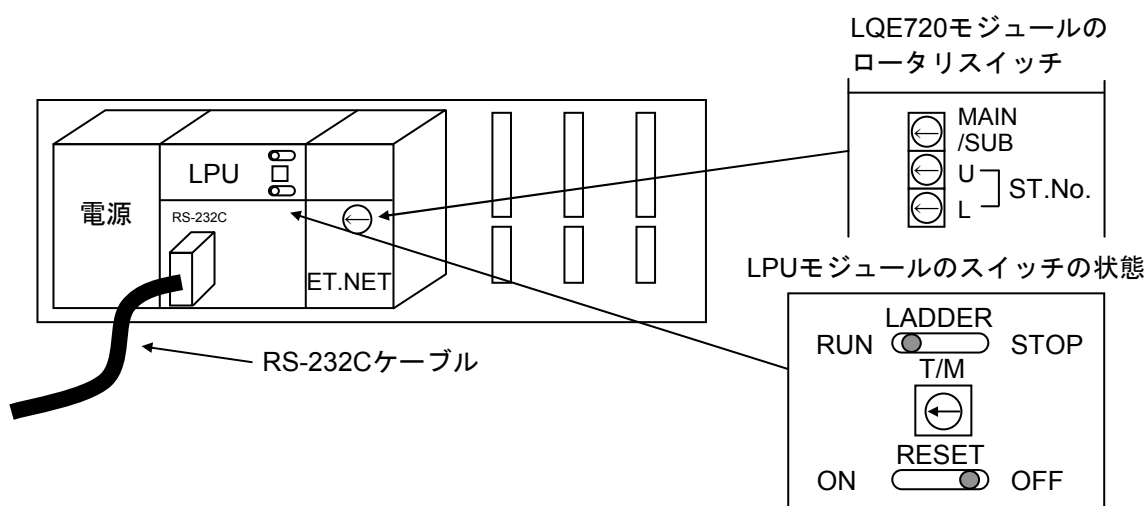
## 7. 1. 1 モジュールの交換、増設

## ● 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 (LQE720)
- 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 (MAIN/SUB, ST.No.U, L)。
- 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 ユニットの電源をOFFにします。
- ⑪ ③で接続したRS-232Cケーブルを外します。
- ⑫ ⑥で外したケーブルを元に戻します。
- ⑬ 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。
- ⑭ 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

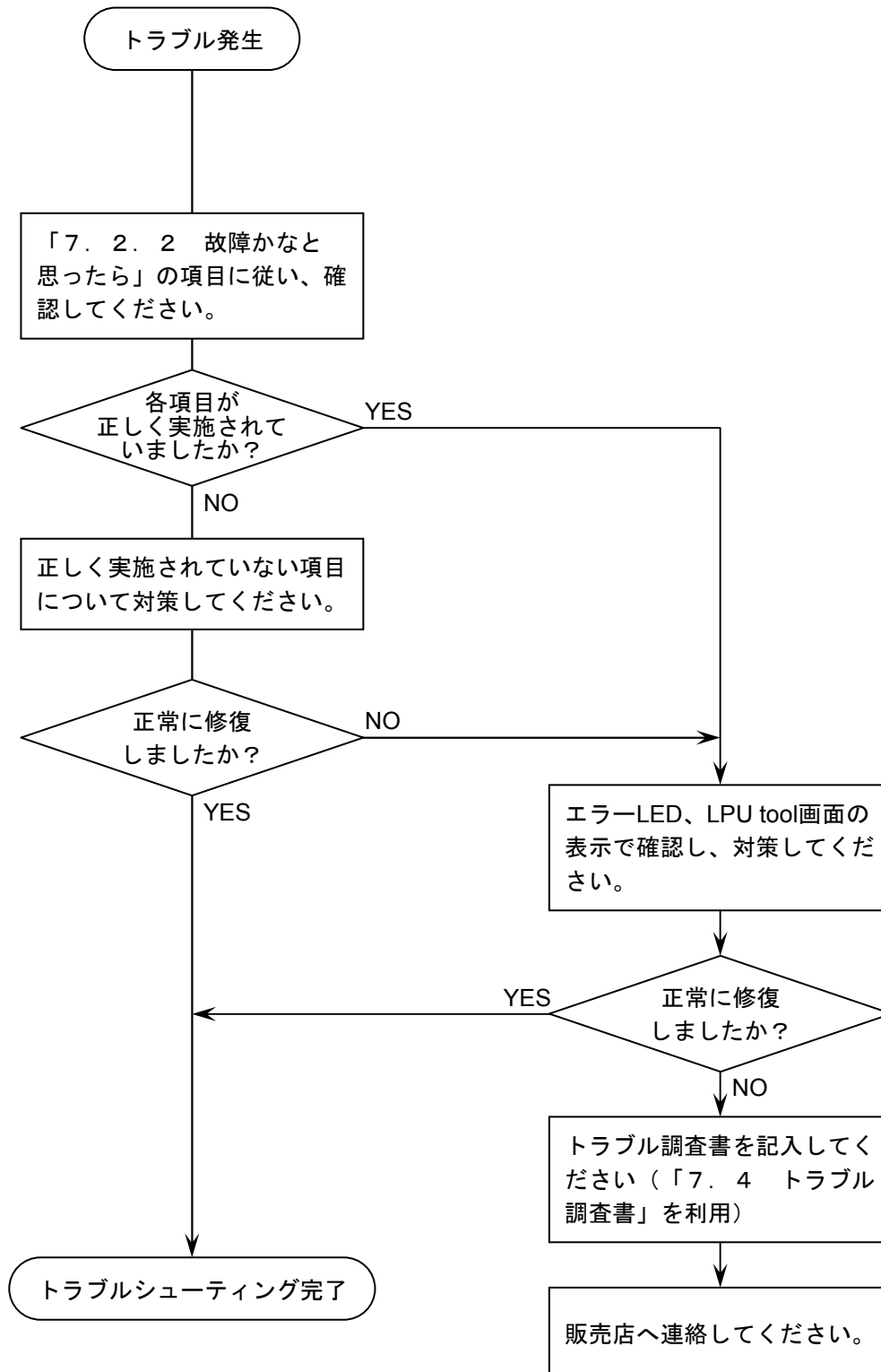


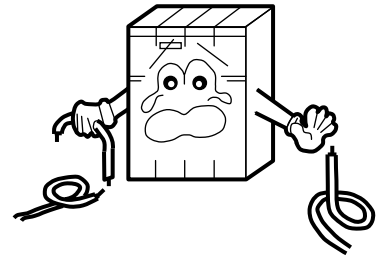
図 7-1 トラブルシューティング手順

## 7 保 守

### 7. 2. 2 故障かなと思ったら

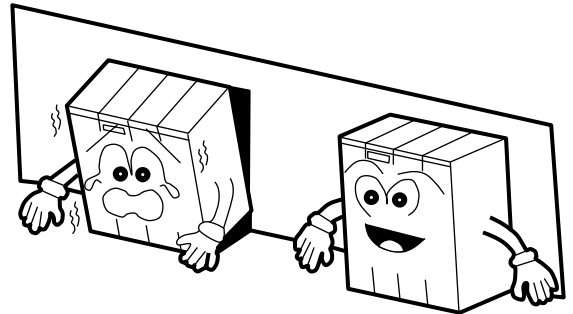
(1) 正しく配線されていますか？

- ・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。



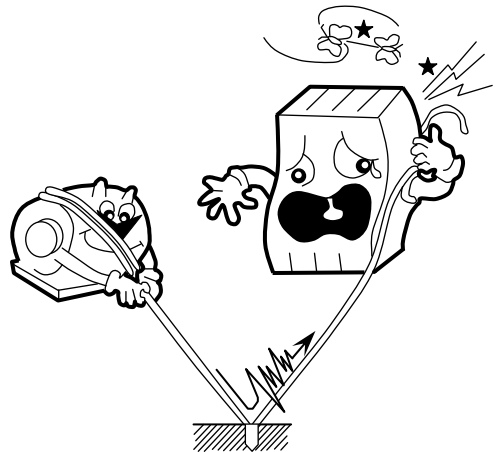
(2) モジュールは正しく実装されていますか？

- ・取り付けねじに緩みがないか調べてください。
- ・LQE520とLQE720が混在して実装されていませんか？LQE520とLQE720の混在実装はできません。



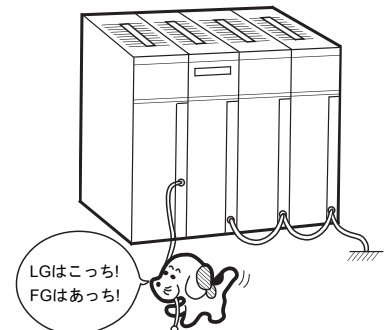
(3) 正しく接地されていますか？

- ・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 エラーログ情報の見方

ET.NETモジュールが検出したエラーは、S10V基本システムの [エラーログ情報] 画面から参照できます。このエラーログ情報は、以下のフォーマットで表示されます。

なお、S10V基本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。

<パニックログ>

[*] ***** (PC=0x*****,FADR=0x*****)			
①	②	③	④

表7-2 パニックログエラーメッセージフォーマット一覧

フォーマットタイプ	エラーメッセージフォーマット
システムダウン（システムエラー）	①+②+③+④

① 障害重要度タイプ

[F]：致命的なエラー

② エラーメッセージ

表7-6を参照してください。なお、エラーメッセージ一覧にないエラーコードの場合、以下のデフォルトエラーメッセージを表示します。

表7-3 パニックログデフォルトエラーメッセージ一覧

フォーマットタイプ	エラーメッセージ
システムダウン（システムエラー）	System down

③ プログラムカウンタ

④ Fault Address



## 7 保 守

<パニックログ以外>

(パターン1)

[*] ***** (UNO=**,DEV=0x******) (TN=***) (SLOT=**)				
①	②	③	④	⑤

表 7-4 パニックログ以外エラーメッセージフォーマット一覧

フォーマットタイプ	エラーメッセージフォーマット
プログラムエラー	①+②+④
マクロパラメータチェックエラー	①+②+④
I/Oエラー	①+②+③
WDTタイムアウトエラー	①+②
モジュールエラー	①+②+⑤
カーネルワーニング	①+②+④
カーネルインフォメーション	①+②+④
システムダウン (カーネルトラップ)	①+②
メモリエラー	①+②+④
システムバスエラー	①+②+⑤
メッセージフレームエラー	①+②
バッファ状態報告	①+②
ソケットエラー	①+②

① 障害重要度タイプ

[F] : 致命的なエラー [W] : 警告

[E] : エラー [I] : 情報

② エラーメッセージ

表 7-6 および「ソフトウェアマニュアル オペレーション NXACP For Windows® (マニュアル番号 SVJ-3-134)」を参照してください。なお、エラーメッセージ一覧にないエラーコードの場合、表 7-5 に示すデフォルトエラーメッセージを表示します。

表7-5 パニックログ以外デフォルトエラーメッセージ一覧

フォーマットタイプ	エラーメッセージ
プログラムエラー	Program error
マクロパラメータチェックエラー	Macro parameter error
I/Oエラー	I/O error
WDTタイムアウトエラー	WDT timeout error
モジュールエラー	Module Error
カーネルワーニング	Kernel Warning
カーネルインフォメーション	Kernel Information
システムダウン (カーネルトラップ)	System down
メモリエラー	Memory error
システムバスエラー	System Bus Error
メッセージフレームエラー	Message frame error
バッファ状態報告	Buffer status
ソケットエラー	Socket error

- ③ ユニット番号、デバイス番号  
 ユニット範囲：1～24  
 デバイス範囲：0x00000000～0xFFFFFFFF
- ④ タスク番号  
 タスク範囲：1～255
- ⑤ スロット番号  
 スロット範囲：0～7

## 7 保 守

---

(パターン2)

パニックログ、パニックログ以外パターン1のフォーマットタイプ以外のエラーは、以下のフォーマットで表示します。

```
%****_*_****_****
```

① ② ③ ④

① エラーを検出したシステム

CPMS : CPMS (基本OS)

LNET : ネットワークドライバ

② 障害重要度タイプ

F : 致命的なエラー      E : エラー

W : 警告                      I : 情報

? : その他の障害

③ 故障種別

HARD : ハードウェア

CPMS : CPMS

SOFT : CPMS以外のソフトウェア

???? : その他

④ コード

フォーマットタイプを表すコードで16進数4桁で表示します。

表 7-6 エラーメッセージ (1/2)

No.	エラーログタイトル	エラーコード	エラーメッセージ	内容	ALARM LED	ERR LED	復旧処置
1	%CPMS-E-SOFT-0001	EC=03620000	Program error (Invalid Data Access)	データアクセスエラー	— (消灯)	— (消灯)	ハードウェア交換
2	%CPMS-E-SOFT-0001	EC=03660000	Program error (Data Access Protection)	データアクセス保護エラー	—	—	ハードウェア交換
3	%CPMS-E-SOFT-0001	EC=03600000	Program error (Data Page Fault)	データアクセスページフォルト	—	—	ハードウェア交換
4	%CPMS-E-SOFT-0001	EC=03420000	Program error (Invalid Inst. Access)	命令アクセスエラー	—	—	ハードウェア交換
5	%CPMS-E-SOFT-0001	EC=03460000	Program error (Inst. Access Protection)	命令アクセス保護エラー	—	—	ハードウェア交換
6	%CPMS-E-SOFT-0001	EC=03400000	Program error (Instruction Page Fault)	命令アクセスページフォルト	—	—	ハードウェア交換
7	%CPMS-E-SOFT-0001	EC=03030000	Program error (Inst. Alignment Error)	命令アラインメントエラー	—	—	ハードウェア交換
8	%CPMS-E-SOFT-0001	EC=03080000	Program error (Privileged Instruction)	特権命令エラー	—	—	ハードウェア交換
9	%CPMS-E-SOFT-0001	EC=03040000	Program error (Illegal Instruction)	不当命令エラー	—	—	ハードウェア交換
10	%CPMS-E-SOFT-0001	EC=03390000	Program error (FP Program Error)	浮動小数点演算エラー	—	—	ハードウェア交換
11	%CPMS-E-SOFT-0001	EC=03470000	Program error (Data Alignment Error)	データアラインメントエラー	—	—	ハードウェア交換
12	%CPMS-E-SOFT-0002	EC=05130000	Macro parameter error	未定義マクロ発行	—	—	ハードウェア交換
13	%CPMS-E-SOFT-0002	EC=05110000	Macro parameter error	マクロパラメータ異常	—	—	ハードウェア交換
14	%CPMS-E-SOFT-0005	EC=05C70000	WDT timeout error	ウォッチドッグタイマアウト	—	点灯	ハードウェア交換
15	%CPMS-E-HARD-0006	EC=03B70000	Module error (Bus Target Abort)	バスターゲットアボート	—	—	ハードウェア交換
16	%CPMS-E-HARD-0006	EC=05000000	Module error (Invalid Interrupt)	無効割り込み	—	—	ハードウェア交換
17	%CPMS-E-HARD-0006	EC=05000001	Module error (Undefined Invalid Interrupt)	未定義無効割り込み	—	—	ハードウェア交換
18	%CPMS-E-HARD-0006	EC=05000002	Module error (INTEVT Invalid Interrupt)	INTEVT無効割り込み	—	—	ハードウェア交換
19	%CPMS-E-HARD-0006	EC=0500F001	Module error (HERST Invalid Interrupt)	重障害無効割り込み	—	—	ハードウェア交換
20	%CPMS-E-HARD-0006	EC=0500F002	Module error (HERST2 Invalid Interrupt)	重障害無効割り込み2	—	—	ハードウェア交換
21	%CPMS-E-HARD-0006	EC=0500F003	Module error (BUERRSTAT Invalid Interrupt)	バスエラー重障害割り込みステータス無効	—	—	ハードウェア交換
22	%CPMS-E-HARD-0006	EC=0500F006	Module error (NHPMCLG Invalid Interrupt)	メモリ重障害割り込みステータス無効	—	—	ハードウェア交換
23	%CPMS-E-HARD-0006	EC=0500F007	Module error (ECC 2bit Master Invalid Interrupt)	メモリECC2ビットエラー重障害ステータス無効	—	—	ハードウェア交換
24	%CPMS-E-HARD-0006	EC=0500F008	Module error (RERRMST Invalid Interrupt)	RERR割り込みステータス無効	—	—	ハードウェア交換
25	%CPMS-E-HARD-0006	EC=0500C001	Module error (NINTR Invalid Interrupt)	NINTステータス無効	—	—	ハードウェア交換
26	%CPMS-E-HARD-0006	EC=0500B001	Module error (PUINTR Invalid Interrupt)	PUINTステータス無効	—	—	ハードウェア交換
27	%CPMS-E-HARD-0006	EC=05005001	Module error (RINTR Invalid Interrupt)	RINTステータス無効	—	—	ハードウェア交換
28	%CPMS-E-HARD-0006	EC=05003001	Module error (LV3 INTST Invalid Interrupt)	レベル#3割り込みステータス無効	—	—	ハードウェア交換
29	%CPMS-E-HARD-0006	EC=05003002	Module error (RQI6 INF Invalid Interrupt)	RQI6ステータス無効	—	—	ハードウェア交換
30	%CPMS-E-HARD-0006	EC=05001001	Module error (RQI3 INT Invalid Interrupt)	RQI3ステータス無効	—	—	ハードウェア交換
31	%CPMS-E-HARD-0006	EC=05001002	Module error (RQI3 Link Invalid Interrupt)	RQI3リンクステータス無効	—	—	ハードウェア交換
32	%CPMS-E-HARD-0006	EC=05001003	Module error (RQI3 Module Invalid Interrupt)	RQI3モジュールステータス無効	—	—	ハードウェア交換
33	%CPMS-E-HARD-0006	EC=07D00001	Module error (INVALID EXCEPTION)	無効例外発生	—	点灯	ハードウェア交換
34	%CPMS-E-HARD-0006	EC=07D00010	Module error (INVALID MAIN/SUB SWITCH SETTING)	メイン/サブ設定スイッチの設定誤り	—	点灯	メイン/サブスイッチ設定見直し
35	%CPMS-E-HARD-0006	EC=07D00011	Module error (INVALID MAC ADDRESS)	MACアドレス異常	—	点灯	ハードウェア交換
36	%CPMS-E-HARD-0006	EC=07D00012	Module error (MAIN/SUB SWITCH SETTING DUPLICATION)	メイン/サブ設定スイッチの設定重複	点滅	点灯	メイン/サブスイッチ設定見直し
37	%CPMS-E-HARD-0006	EC=07D00013	Module error (ETHERNET LSI CHECK ERROR)	LANCE診断異常	—	点灯	ハードウェア交換
38	%CPMS-E-HARD-0006	EC=07D00014	Module error (SDRAM CHECK ERROR)	SDRAM初期化異常	—	点灯	ハードウェア交換
39	%CPMS-E-HARD-0006	EC=07D00015	Module error (OS-ROM CHECKSUM ERROR)	ROMサムエラー (CPMS部)	—	点灯	ハードウェア交換
40	%CPMS-E-HARD-0006	EC=07D00016	Module error (CAN NOT MOUNTING WITH LQE520 MODULE)	LQE520と混在実装	—	点灯	LQE520とは同じマウントベースに実装できません。LQE520を取り外してください。

表 7-6 エラーメッセージ (2/2)

No.	エラーログタイトル	エラーコード	エラーメッセージ	内容	ALARM LED	ERR LED	復旧処置
41	%CPMS-E-HARD-0006	EC=07D00018	Module error (TASK-ROM CHECKSUM ERROR)	ROMサムエラー (通信タスク)	—	点灯	ハードウェア交換
42	%CPMS-W-HARD-0006	EC=07D01003	Module error (THE VERSION OF CMU MODULE IS OLD)	LQE720未対応のCMUモジュールと同時実装	点灯	—	CMUモジュールをLQE720対応版と交換
43	%CPMS-W-HARD-0006	EC=0D010000	Module error (Memory Alarm)	メモリビットエラー (リリット)	—	—	ハードウェア交換
44	%CPMS-E-HARD-0006	EC=0D330000	Module error (Hardware WDT timeout)	ハードウェアウォッチドッグタイムアウト	—	—	ハードウェア交換
45	%CPMS-E-HARD-0006	EC=0D340000	Module error (Software WDT Timeout)	ソフトウェアウォッチドッグタイムアウト	—	—	ハードウェア交換
46	%CPMS-F-HARD-0009	EC=0D810000	System down (BPU Error)	BPUエラー	—	点灯	ハードウェア交換
47	%CPMS-F-HARD-0009	EC=03820000	System down (Memory Error)	メモリエラー	—	点灯	ハードウェア交換
48	%CPMS-F-HARD-0009	EC=038A0000	System down (Memory Access Error)	メモリアクセスエラー	—	点灯	ハードウェア交換
49	%CPMS-F-HARD-0009	EC=038B0000	System down (Internal Bus Parity)	内部バスパリティエラー	—	点灯	ハードウェア交換
50	%CPMS-F-HARD-0009	EC=038C0000	System down (System Bus Parity)	システムバスパリティエラー	—	点灯	ハードウェア交換
51	%CPMS-F-HARD-0009	EC=038F0000	System down (Undefined Machine Check)	未定義マシンのチェックエラー	—	点灯	ハードウェア交換
52	%CPMS-F-CPMS-0009	EC=03620000	System down (Invalid Data Access)	データアクセスエラー	—	点灯	ハードウェア交換
53	%CPMS-F-CPMS-0009	EC=03660000	System down (Data Access Protection)	データアクセス保護ロケールエラー	—	点灯	ハードウェア交換
54	%CPMS-F-CPMS-0009	EC=03600000	System down (Data Page Fault)	データアクセスページフォールト	—	点灯	ハードウェア交換
55	%CPMS-F-CPMS-0009	EC=03420000	System down (Invalid Inst. Access)	命令アクセスエラー	—	点灯	ハードウェア交換
56	%CPMS-F-CPMS-0009	EC=03460000	System down (Inst. Access Protection)	命令アクセス保護ロケールエラー	—	点灯	ハードウェア交換
57	%CPMS-F-CPMS-0009	EC=03400000	System down (Instruction Page Fault)	命令アクセスページフォールト	—	点灯	ハードウェア交換
58	%CPMS-F-CPMS-0009	EC=03030000	System down (Inst. Alignment Error)	命令アラインメントエラー	—	点灯	ハードウェア交換
59	%CPMS-F-CPMS-0009	EC=03040000	System down (Illegal Instruction)	不当命令エラー	—	点灯	ハードウェア交換
60	%CPMS-F-CPMS-0009	EC=03380000	System down (FP Unavailable)	浮動小数点使用不可例外	—	点灯	ハードウェア交換
61	%CPMS-F-CPMS-0009	EC=03390000	System down (FP System down)	浮動小数点演算エラー	—	点灯	ハードウェア交換
62	%CPMS-F-CPMS-0009	EC=03470000	System down (Data Alignment Error)	データアラインメントエラー	—	点灯	ハードウェア交換
63	%CPMS-F-CPMS-0009	EC=030F0000	System down (Illegal Exception)	不当例外エラー	—	点灯	ハードウェア交換
64	%CPMS-F-CPMS-0009	EC=05700000	System down (System Error)	システムクラッシュ (システムエラー)	—	点灯	ハードウェア交換
65	%CPMS-F-CPMS-000A	EC=05800000	System down (Kernel Trap)	システムクラッシュ (カーネルトラップ)	—	点灯	ハードウェア交換
66	%LNET-W-HARD-004	EC=07801308	I/O error (SEND_TIMEOUT)	送信タイムアウトエラー	—	—	自動復旧
67	%LNET-E-HARD-004	EC=07801308	I/O error (SEND_TIMEOUT)	送信タイムアウトエラー	—	点灯	復電、復旧しない場合ハードウェア交換
68	%LNET-W-HARD-004	EC=0780130A	I/O error (RESET_ERROR)	ハードリセットエラー	—	—	自動復旧
69	%LNET-E-HARD-004	EC=0780130A	I/O error (RESET_ERROR)	ハードリセットエラー	—	点灯	復電、復旧しない場合ハードウェア交換
70	%LNET-W-HARD-004	EC=0780130E	I/O error (MEMORY)	メモリエラー	—	—	自動復旧
71	%LNET-E-HARD-004	EC=0780130E	I/O error (MEMORY)	メモリエラー	—	点灯	復電、復旧しない場合ハードウェア交換
72	%LNET-W-HARD-004	EC=07801310	I/O error (LOSS)	キャリアロスタエラー	—	—	伝送路見直し (*1)
73	%LNET-W-HARD-004	EC=07801311	I/O error (RETRY)	リトライエラー	—	—	伝送路見直し (*3)
74	%LNET-W-HARD-004	EC=07801312	I/O error (LATE)	レイトコリジョンエラー	—	—	伝送路見直し (*5)
75	%LNET-W-HARD-004	EC=07801351	I/O error (TX_ABORT)	送信異常終了	—	—	伝送路見直し
76	%LNET-W-HARD-004	EC=07801353	I/O error (TX_DEFER)	送信遅延による送信エラー	—	—	伝送路見直し
77	%LNET-W-HARD-004	EC=07801370	I/O error (EC_PCI_ERROR)	通信LSIにてPCIエラー検出	—	—	ハードウェア交換 (*6)
78	%LNET-W-HARD-004	EC=07801376	I/O error (TX_DATA_UNDER)	送信データFIFOオーバーラン	—	—	伝送路見直し (*6)
79	%LNET-W-HARD-004	EC=07801375	I/O error (RX_STAT_OVER)	受信ステータスFIFOオーバーラン	—	—	回線負荷見直し (*6)
80	%LNET-W-HARD-004	EC=07801377	I/O error (RX_DATA_OVER)	受信データFIFOオーバーラン	—	—	回線負荷見直し (*7)
81	%LNET-E-HARD-004	EC=07D01001	I/O error (IP_ADDRESSES_NOT_REGISTERED)	IPアドレス未設定	点灯	—	IPアドレスの登録
82	%LNET-E-HARD-004	EC=07801400	I/O error (PCI_BUS_ERR)	PCIバスエラー	—	—	ハードウェア交換
83	%LNET-E-HARD-004	EC=07801505	I/O error (INV_INTR)	回線からの無効割り込み発生	—	—	ハードウェア交換
84	%LNET-E-SOFT-004	EC=07801508	I/O error (BUF_OVF)	OS管理送受信バッファがオーバーフロー	—	—	回線負荷見直し (*2)
85	%LNET-W-SOFT-004	EC=0780150F	I/O error (SOCKET_OVF)	ソケットバッファが満杯	—	—	ユーザプログラム見直し
86	%LNET-W-SOFT-004	EC=07801510	I/O error (IFCONFIG_UP)	ネットワーク/F初期化エラー	—	—	設定内容見直し
87	%LNET-W-SOFT-004	EC=07801512	I/O error (IPADDR_DUPL)	IPアドレス重複エラー	点滅 (*4)	—	設定内容見直し (*4)

- (\*1) LSIキャリアロスを32回検出した場合に、このメッセージを1回出力します。LSIキャリアロスはLINK LEDが消灯時（リンク未確立）にデータ送信した場合に発生します。したがって、CPUを立ち上げた際、LINK LEDが点灯するまでにアプリケーションプログラムから32回以上送信要求があった場合にも発生します。この場合は、LINK LEDが点灯してから送信する等アプリケーションプログラムの改善が必要です。
- (\*2) 通信高負荷時バッファ不足により発生します。
- (\*3) 連続32回検出した場合に、このメッセージを1回出力します。
- (\*4) 他の計算機とIPアドレスが重複しています。重複している相手より後から回線に接続した場合はARARM LEDが点滅します。
- (\*5) 連続8回検出した場合に、このメッセージが1回出力されます。連続16回以上検出した場合は、256回検出に1回出力します。
- (\*6) 連続5回検出した場合に、このメッセージが1回出力されます。
- (\*7) 連続10回検出した場合に、このメッセージが1回出力されます。

7. 3. 2 DHPトレース情報の見方

DHPトレース情報の内容は、以下のフォーマットで表示されます。

- DHPの表示は、最新のデータから古いデータの方向に表示します。
- DHPは、イベントDISPATCH\_Eを区切りにタスク、アイドル、OSに分けられます。
- DISPATCH\_EでDATA1=0x00000001~0x0000012Cがタスク番号で、タスクを示します。
- 時間は、小数点以下6桁までの秒、つまりマイクロ秒まで示します。
- DHPのイベントとデータの対応は、表7-7を参照してください。

<DHPの表示例>

以下のDHPトレース情報の表示例について、動作説明にタスクの動作と動作タスク切り替え時のOSの動作を示します。

						動作説明	
新 ↑	165	40.901912	TASK_PRI	112	10	00000071 00000032	タスク (112) 動作
	166	40.901901	RLEAS	112	10	00000071	
	167	40.901883	DISPATCH_E	112	10	00000070 00000032 84DB2000 00000002	OS (タスク111 abortし、 タスク112に動作切替)
	168	40.901868	DISPATCH	111	10	0000006F 00000032 84DAF000	
	169	40.901832	DISPATCH_E	111	10	0000006F 00000032 84DAF000 00000002	
	170	40.901815	RUNQ	112	10	00000070	
	171	40.901810	DISPATCH	112	10	00000070 00000032 84DB2000	
	172	40.901796	RUNQ	112	10	0000006F	タスク (112) 動作
	173	40.901785	WAKEUP	112	10	F0000000	
	174	40.901771	ABORT	112	10	0000006F	
	175	40.901748	GFACT	112	10	00000003	OS (タスク111 delayし、 タスク112に動作切替)
	176	40.901727	DISPATCH_E	112	10	00000070 00000032 84DB2000 00000002	
	177	40.901703	DISPATCH	111	10	0000006F 0000001C 84DAF000	
	178	40.901691	TASK_PRI	111	10	0000006F 0000001C 00000000	
	179	40.901611	DELAY	111	10	00000BB8	
	180	40.901600	RUNQ	111	10	00000070	タスク (111) 動作
	181	40.901590	QUEUE	111	10	00000070 00000003	
	182	40.901579	TASK_PRI	111	10	00000070 00000032	
	183	40.901568	RLEAS	111	10	00000070	
	184	40.901546	GFACT	111	10	00000002	OS (タスク110 waitし、 タスク111に動作切替)
185	40.901525	DISPATCH_E	111	10	0000006F 00000032 84DAF000 00000002		
186	40.901507	DISPATCH	110	10	0000006E 00000032 84DAC000		
187	40.901493	SLEEP	110	10	841C982C 00000032		
188	40.901483	WAIT	110	10	5004502C		
189	40.901471	RUNQ	110	10	0000006F	タスク (110) 動作	
190	40.901459	QUEUE	110	10	0000006F 00000002		
191	40.901446	TASK_PRI	110	10	0000006F 00000032		
192	40.901434	RLEAS	110	10	0000006F		
193	40.901408	DISPATCH_E	110	10	0000006E 00000032 84DAC000 00000001	OS (タスク119終了し、 タスク110に動作切替)	
194	40.901399	RUNQ	110	10	0000006E		
195	40.901393	DISPATCH	110	10	0000006E 00000032 84DAC000		
196	40.901373	DISPATCH_E	110	10	0000006E 00000032 84DAC000 00000002		
197	40.901348	DISPATCH	119	10	00000077 00000032 84DC7000		
198	40.901323	EXIT	119	10		タスク (119) 動作	
199	40.901311	RUNQ	119	10	0000006E		
200	40.901300	WAKEUP	119	10	841C982C		
201	40.901288	POST	119	10	5004502C 00001234		

表 7-7 DHPコード一覧 (1/3)

CPMS処理 (トレース処理)

コード値	DHP表示名称	トレースイベント	DATA1	DATA2	DATA3	DATA4	DATA5
0x010001	TRACE_ON	トレース開始					
0x010002	TRACE_OFF	トレース停止					
0x010003	TRACE_TBU	時刻記録	old tbu (Time Base Upper)	new tbu (Time Base Upper)			

CPMS処理 (スケジューリング処理)

コード値	DHP表示名称	トレースイベント	DATA1	DATA2	DATA3	DATA4	DATA5
0x020001	WAKEUP	トレースイベント	wchan				
0x020002	SLEEP	SLEEPイベント	wchan	pri (優先レベル)			
0x020003	DISPATCH	thread invoke処理前	tn (タスク番号)	pri (優先レベル)	cont (CPMSタスク情報)		
0x020003	DISPATCH	thread invoke処理後	tn (タスク番号)	pri (優先レベル)	cont (CPMSタスク情報)		
0x020004	RUNQ	RUNQ接続	tn (タスク番号)				
0x020005	IDLE	IDLE処理					
0x020006	TASK_PRI	優先レベル制御	tn (タスク番号)	pri (優先レベル)			

CPMS処理 (エラーログ、組み込みサブルーチン処理)

コード値	DHP表示名称	トレースイベント	DATA1	DATA2	DATA3	DATA4	DATA5
0x030002	ELSETK	else処理	エラータイプ	エラークラス	エラーフォーマット	エラーコード	
0x030003	IOERR	I/Oエラー処理	uno	デバイス番号	デバイスアドレス	詳細エラーコード	
0x030004	PRGERR	プログラマエラー処理	tn (タスク番号)	フォーマットアドレス	PC	expectレジスタ	
0x030005	WDTERR	WDTエラー処理	time				
0x030006	PIOERR	PI/Oエラー処理	slot				
0x030007	MODERR	モジュールエラー処理	エラーコード	slot	HERSTレジスタ	INTSTレジスタ	
0x030008	KERN_PANIC	パニック処理	tn (タスク番号)	フォーマットアドレス	PC	拡張エラーコード	
0x03000a	ASSERT	アサーションパニック処理	エラー発生部位	line	判定条件		
0x03000b	CPUSTOP	CPU停止処理	組み込みサブルーチン名	組み込みサブルーチン番号	組み込みサブルーチン値		

CPMS処理 (立ち上げ/停止処理)

コード値	DHP表示名称	トレースイベント	DATA1	DATA2	DATA3	DATA4	DATA5
0x040001	SETUP_MAIN	立ち上げ処理					
0x040002	HDUTL_STOP	停止処理	1 固定				
0x040003	HDUTL_RSUM	リサマ処理					
0x040004	HDUTL_ERR	ERROR処理					

CPMS処理 (例外処理)

コード値	DHP表示名称	トレースイベント	DATA1	DATA2	DATA3	DATA4	DATA5
0x050001	EXCEPTION	例外処理	例外種別				
0x050002	SLIH_SRES	システムセット例外	NMI要因レジスタ	PC			
0x050005	SLIH_SM	システム管理割り込み例外	MSWレジスタ				
0x050007	SLIH_HERR	重畳割り込み処理	重畳書要因レジスタ				



表 7-7 DHPコード一覧 (2/3)

CPMS処理 (マクロ処理)

コード値	DHP表示名称	トピック	DATA1	DATA2	DATA3	DATA4	DATA5
0x100000	NOSYS	未定義マクロ発行					
0x100001	QUEUE	queue発行	tn (タスク番号)	fact (起動要因)			
0x100002	REAS	rleas発行	tn (タスク番号)				
0x100003	SFACT	sfact発行	tn (タスク番号)	fact (起動要因)			
0x100004	ABORT	abort発行	tn (タスク番号)				
0x100005	SUSP	susp発行	tn (タスク番号)				
0x100006	RSUM	rsum発行	tn (タスク番号)				
0x100007	CTIME	ctime発行	tn (タスク番号)	fact (起動要因)			
0x100008	WAIT	wait発行	ecb (ECBアドレス)				
0x100009	POST	post発行	ecb (ECBアドレス)				
0x10000a	TIMER	timer発行	id (イベントタイプ)	pcode (ポストコード)	fact (起動要因)	t (時間/時刻)	cyt (周期時間)
0x10000b	DELAY	delay発行	t (ミリ秒)				
0x10000c	STIME	stime発行	year (年)	month (月)	day (日)	msec (ミリ秒)	
0x10000d	CHAP	chap発行	tn (タスク番号)	chgp (優先レベル)			
0x10000e	RSERV	rsvr発行	n (共有資源の数)	para1	para2	para3	para4
0x10000f	FREE	free発行	n (共有資源の数)	para1	para2	para3	para4
0x100010	PRSERV	prsvr発行	n (共有資源の数)	para1	para2	para3	para4
0x100011	PFREE	pfree発行	n (共有資源の数)	para1	para2	para3	para4
0x100012	GFACT	gfact発行前後	fact (起動要因)				
0x100013	GTIME	gtime発行	time (time_tアドレス)				
0x100014	EXIT	exit発行					
0x100015	ASUSP	asusp発行					
0x100016	ARSUM	arsum発行					
0x100017	OPEN	open発行	uno	flag			
0x100018	CLOSE	close発行	uno				
0x100019	READ	read発行	uno	論理アドレス	cnt		
0x10001a	WRITE	write発行	uno	論理アドレス	cnt		
0x10001e	DHPCTL	dhpctl発行	cmd (コマンド)	id (Major ID)	info		
0x10001f	DHPREAD	dhpread発行	論理アドレス	size			
0x100023	CHML	chml発行	論理アドレス	para1	para2	para3	para4

CPMS処理

コード値	DHP表示名称	トピック	DATA1	DATA2	DATA3	DATA4	DATA5
0x200004	SETTCB	settc b発行	topbn	cnt	tcbaadr		
0x20000c	REGSET	taskレジスタの設定	reg	レジスタアドレス			

表 7-7 DHPコード一覧 (3/3)

RCTLNET (ネットワークドライバ) 処理

コード値	DHP表示名称	イベント	DATA1	DATA2	DATA3	DATA4	DATA5
0x300001	SOCKET	SOCKET発行	uno (ユニット番号)	タイプ	アドレス	ワーク	ワーク
0x300002	BIND	BIND発行	ポート番号	ポート番号	IPアドレス	ワーク	ワーク
0x300003	LISTEN	LISTEN発行	接続待ち最大数		ワーク	ワーク	ワーク
0x300004	ACCEPT	ACCEPT発行	アドレス情報	アドレス情報	アドレス情報	ワーク	ワーク
0x300005	CONNECT	CONNECT発行	ポート番号	ポート番号	IPアドレス	ワーク	ワーク
0x300006	SEND	SEND発行		上位ワード；データ長 下位ワード；送信フラグ	上位ワード；データ長 下位ワード；送信フラグ	ワーク	ワーク
0x300007	SENDTO	SENDTO発行	ポート番号	上位ワード；データ長 下位ワード；送信フラグ	ポート番号	IPアドレス	内部アドレス情報
0x300008	RECV	RECV発行		上位ワード；データ長 下位ワード；送信フラグ	上位ワード；データ長 下位ワード；送信フラグ	ワーク	ワーク
0x300009	RECVFROM	RECVFROM発行		上位ワード；データ長 下位ワード；送信フラグ	上位ワード；データ長 下位ワード；送信フラグ	アドレス情報	アドレス情報
0x30000a	SETSOCKOPT	SETSOCKOPT発行	レベル	レベル	オプション	アドレス情報	アドレス情報
0x30000b	GETSOCKOPT	GETSOCKOPT発行	レベル	レベル	オプション	アドレス情報	アドレス情報
0x30000c	SHUTDOWN	SHUTDOWN発行	シャットダウン方法		ワーク	ワーク	ワーク
0x30000d	NET_END	マクロ異常終了	エラー番号		ワーク	ワーク	ワーク
0x300010	NET_CTLR	IOCTL発行	ユニット番号 + スロット番号	制御情報	制御情報	制御情報	制御情報
0x300011	NET_START	モジュール制御受け付け	アドレス番号 + コマンド	フレーム長 + 送信No.	対象種別 + データ長	データアドレス	ワーク
0x300011	NET_START	NCP-F I/O起動	アドレス情報	アドレス情報	アドレス情報	起動情報1	起動情報2
0x300011	NET_START	内蔵LANCE, LANC送信	アドレス情報	アドレス情報	アドレス情報	アドレス情報	割り込み情報
0x300012	NET_TERM	NCP-F終了割り込み					
0x300012	NET_TERM	内蔵LANCE, LANC終了割り込み	LANCEアドレス情報 (TMD0, TMD1, TMD2, TMD3)				
0x300013	NET_ATEN	NCP-Fアドレス割り込み	アドレス情報	アドレス情報	アドレス情報	アドレス情報	割り込み情報
0x300013	NET_ATEN	内蔵LANCE, LANC受信	アドレス情報	アドレス情報	アドレス情報	アドレス情報	割り込み情報
0x300014	NET_STO	ソフトウェアリセット	アドレス情報	アドレス情報	アドレス情報	アドレス情報	起動情報
0x300015	NET_SUB	エラー検出	エラー種別	エラー情報	エラー情報	エラー情報	エラー情報

## 7. 3. 3 ネットワーク情報の見方

ET.NETモジュールのネットワーク情報は、以下のフォーマットで表示されます。

## (1) ソケット情報

現在のネットワーク接続の一覧を表示します。



- プロトコル  
コネクションのプロトコル名称が表示されます。
- ローカルアドレス  
ローカル（接続元）ホストのIPアドレスが表示されます。  
ソケットにIPアドレスがバインドされていない場合は「\*」が表示されます。
- ローカルポート  
ローカル（接続元）ホストのポート番号が表示されます。
- リモートアドレス  
リモート（接続先）ホストのIPアドレスが表示されます。  
ソケットにIPアドレスがバインドされていない場合は「\*」が表示されます。

- リモートポート

リモート（接続先）ホストのポート番号が表示されます。

ソケットにIPアドレスがバインドされていない場合は「\*」が表示されます。

- ステータス

TCPプロトコルの接続ステータスが表示されます。

ステータスには以下の種類があります。

表示	内容
CLOSED	未使用状態。
LISTEN	利用可能なポートの待ち受け状態。
SYN_SENT	サーバに接続（SYN）要求を送信したが、応答（ACK）を受けていない状態。
SYN_RECEIVED	クライアントから接続要求を受け取った直後の状態。
ESTABLISHED	TCPでの接続が確立されて、現在通信が行われている状態。
FINWAIT1	サーバからFINが送信された状態。
FINWAIT2	ACK受信状態。
CLOSEWAIT	サーバからFINを受信した状態。
LASTACK	FINに対するACK待ちの状態。
CLOSING	FINを受け取ってコネクションが閉じられる状態。
TIMEWAIT	接続終了待ちの状態。

接続ステータスの状態遷移を以下に示します。

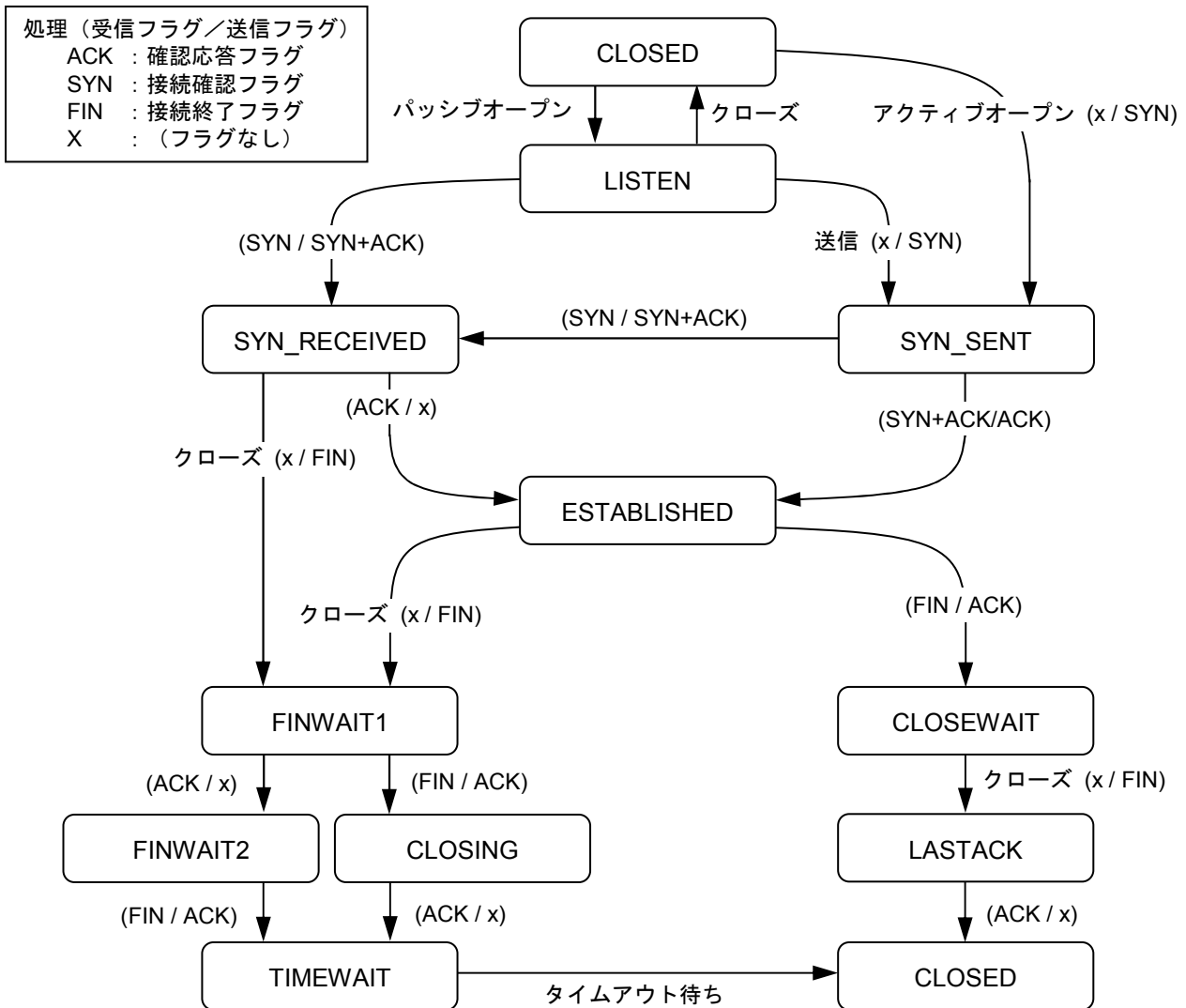


図 7-2 接続ステータス状態遷移

- TCPプロトコルが150ポートを超えた場合、TCPプロトコルの超えた分、およびUDPプロトコルは表示されません。
- TCPプロトコルが80ポートを超えた場合、UDPプロトコルの一部が表示されない場合があります。

## (2) インタフェース情報

動作中のネットワークインタフェース情報、および入出力パケット積算情報を表示します。

内容	ユニットNo.1
スロット番号	4
インタフェース種別	EPORT
最大転送バイト数(MTU)	1500
IPアドレス	192.192.192.1
ネットマスク	255.255.255.0
ブロードキャストアドレス	192.192.192.255
メッセージの送信要求を受け付けた回数	0
メッセージ送信に成功した回数	162
メモリ不足によりメッセージ送信が失敗した回数	0
ハートからメッセージ送信失敗を報告された回数	3
ユーザへ送信メッセージを渡した回数	0
ハートからメッセージの受信を報告された回数	250
メモリ不足によりメッセージの受信に失敗した回数	0
ハートからメッセージ受信失敗を報告された回数	0

- ① スロット番号  
表示対象モジュールの実装スロット番号が表示されます。
- ② インタフェース種別  
常に「EPORT」が表示されます。
- ③ 最大転送バイト数 (MTU)  
一度の転送で送信できるデータ分割ブロックの最大値を表すMTU (Maximum Transmission Unit)が表示されます。
- ④ IPアドレス  
IPアドレスが表示されます。
- ⑤ ネットマスク  
サブネットマスクが表示されます。
- ⑥ ブロードキャストアドレス  
ブロードキャストアドレスが表示されます。
- ⑦ メッセージの送信要求を受け付けた回数  
メッセージの送信要求を受け付けた回数が表示されます。

- ⑧ メッセージ送信に成功した回数  
メッセージ送信に成功した回数が表示されます。
- ⑨ メモリ不足によりメッセージ送信が失敗した回数  
メモリ不足によりメッセージ送信が失敗した回数が表示されます。
- ⑩ ハードからメッセージ送信失敗を報告された回数  
ドライバがハードウェアにメッセージの送信要求をした結果、ハードウェアからメッセージ送信の失敗を報告された回数が表示されます。
- ⑪ ユーザへ送信メッセージを渡した回数  
ユーザへ送信メッセージを渡した回数が表示されます。
- ⑫ ハードからメッセージの受信を報告された回数  
ハードウェアからメッセージの受信を報告された回数が表示されます。
- ⑬ メモリ不足によりメッセージの受信に失敗した回数  
メモリ不足によりメッセージの受信に失敗した回数が表示されます。
- ⑭ ハードからメッセージ受信失敗を報告された回数  
ドライバがハードウェアに受信メッセージの取得要求をした結果、ハードウェアから失敗を報告された回数が表示されます。

## (3) メモリ統計情報

メモリ管理ルーチンによって記録された統計値を表示します。

ネットワーク情報

接続モジュール  
モジュール名称 ET.NET (メイン)  
IPアドレス 192.192.192.1

表示対象モジュール  
モジュール名称 ET.NET (メイン)

閉じる  
更新(R)  
保存(V)

ソケット | インタフェース **メモリ** | 経路情報 | IP | ICMP | TCP | UDP | 積算情報 | ARP

クラスタ先頭アドレス: 0x84934000

内容	CURRENT	MAX	HIGH	DROP
使用中のmbuf数/全mbuf数	46/48	32/48	64	0/0
通信データ	1	1	7	0
パケットヘッダ	2	0	3	0
ソケットテーブル	14	10	14	0
プロトコル制御テーブル	26	18	26	0
ルーティングテーブルエントリ	2	2	2	0
ソケットアドレス	0	0	1	0
ソケットオフショフ	0	0	1	0
ネットワークインタフェースアドレス	1	1	1	0
使用中のクラスタメモリサイズ/総クラスタメモリサイズ	6/992	10/992	10	0/0
mbuf	6	6	8	0
クラスタ	0	4	4	0

- CURRENT : 現在のmbufの状態が表示されます。
- MAX : 最大使用時のmbufの状態が表示されます。
- HIGH : 各項目のピーク値が表示されます。
- DROP : オーバーフロー時のmbufの状態が表示されます。

## ① クラスタ先頭アドレス

クラスタメモリの先頭アドレスが表示されます。



## 7 保 守

### ② 使用中のmbuf数／全mbuf数

現在使用中のmbuf数、および全mbuf数が表示されます。

使用中のmbufの詳細は、以下の項目を参照してください。

項目	内容
通信データ	通信データを格納しているmbuf数
パケットヘッダ	パケットヘッダを格納しているmbuf数
ソケットテーブル	ソケットテーブルを格納しているmbuf数
プロトコル制御テーブル	プロトコル制御テーブルを格納しているmbuf数
ルーティングテーブルエントリ	ルーティングテーブルエントリを格納しているmbuf数
IPリアセンブル待ちデータ	IPリアセンブル待ちデータを格納しているmbuf数
ソケットアドレス	ソケットアドレスを格納しているmbuf数
ソケットオプション	ソケットオプションを格納しているmbuf数
ネットワークインタフェースアドレス	ネットワークインタフェースのアドレスを格納しているmbuf数

### ③ 使用中のクラスタメモリサイズ／総クラスタメモリサイズ

現在使用中のクラスタメモリサイズ／mbuf、およびクラスタに割り当てられている総メモリサイズが表示されます。

使用中のクラスタメモリサイズの詳細は、以下の項目を参照してください。

内容	項目
mbuf	mbufとして使用しているメモリサイズ
クラスタ	クラスタとして使用しているメモリサイズ

### ④ クラスタ満杯のためmbuf、クラスタを確保できなかった回数

クラスタ満杯のためmbuf、クラスタを確保できなかった回数が表示されます。

### ⑤ クラスタの使用数が最大まで達した後、クラスタを要求した回数

クラスタの使用数が最大まで達した後、クラスタを要求した回数が表示されます。

CURRENT, MAX, HIGH, DROPの値がすべて0の項目はリストに表示されません。

## (4) 経路情報

CMUモジュールおよびET.NETモジュールに登録されている経路情報を表示します。



## ① 相手局アドレス

宛先ネットワークアドレス番号が表示されます。

仮想ネットワークアドレスの場合、アドレスの末尾に「\*」が追加されます。

## ② ゲートウェイIPアドレス

相手局アドレスに対応したゲートウェイのIPアドレスが表示されます。

## ③ Flags

経路状態のフラグが表示されます。

フラグには以下の種類があります。

表示	内容
U	経路が作動中
G	ゲートウェイへのルーティング
H	ホストへのルーティング

④ Refcnt

その経路を使用中のユーザ数が表示されます。

⑤ Metric

宛先へ到達するまでに通過するゲートウェイ数が表示されます。

⑥ インタフェース種別

常に「EPORT」が表示されます。

## (5) IP統計情報

IPプロトコルに関する統計値を表示します。



## ① フォワーディング

フォワーディングするゲートウェイとして動作するかが表示されます。

フォワーディング機能は非サポートですので「host」とのみ表示されます。

## ② デフォルトTTL

最大ホップ数を決めるTTL (Time To Live) の初期値が表示されます。

## ③ 受信IPパケット数

全ネットワークインタフェースから受信したIPパケットの総数が表示されます。

## ④ 破棄されたパケット数 (IPヘッダエラー)

IPヘッダ中にチェックサムエラーやバージョンエラーなど、何らかのエラーがあったため破棄されたIPパケットの総数が表示されます。

## ⑤ 破棄されたパケット数 (アドレスエラー)

宛先IPアドレスが不正だったため破棄されたIPパケットの総数が表示されます。

## ⑥ 破棄されたパケット数 (IPフォワーディング)

フォワードされた (別のインタフェースへルーティングされた) IPパケットの総数が表示されます。

- ⑦ 破棄されたパケット数（プロトコル不明）  
IPヘッダ中に記述されている上位プロトコルが未定義であったIPパケットの総数が表示されま  
す。
- ⑧ 破棄された受信パケット数  
バッファ領域不足などにより、受信したが上位プロトコルへは渡されずに破棄されたり、受信が  
拒否されたりしたIPパケットの総数が表示されます。
- ⑨ 上位プロトコルに渡したパケット数  
TCPやUDPなどの上位プロトコルに渡したIPパケットの総数が表示されます。
- ⑩ 上位プロトコルから渡されたパケット数  
上位プロトコルから送信を依頼されたIPパケットの総数が表示されます。
- ⑪ 破棄された送信パケット数  
バッファ不足などにより破棄されたIPパケットの総数が表示されます。
- ⑫ 破棄されたパケット数（経路不明）  
経路情報の設定ミスなどによりルーティングできずに破棄されたIPパケットの総数が表示されま  
す。
- ⑬ フラグメント保持タイムアウト時間  
再構成待ちフラグメントがホールドされている最大秒数が表示されます。
- ⑭ 受信フラグメント数  
受信したフラグメントパケットの総数が表示されます。
- ⑮ 再構成に成功したフラグメント数  
フラグメントの再構成が成功した回数が表示されます。
- ⑯ 再構成に失敗したフラグメント数  
タイムアウトやリソース不足などにより、フラグメントの再構成が失敗した回数が表示されま  
す。
- ⑰ フラグメント化に成功したパケット数  
送信時にMTUサイズを超えたためフラグメント化された送信IPパケットの総数が表示されます。
- ⑱ フラグメント化に失敗したパケット数  
リソース不足などにより、フラグメント化に失敗した送信IPパケットの総数が表示されます。
- ⑲ 作成されたフラグメント数  
送信IPパケットをフラグメント化して作成されたフラグメントパケットの総数が表示されます。
- ⑳ 破棄されたルーティングエントリ数  
破棄されたルーティングエントリ数が表示されます。

## (6) ICMP統計情報

ICMPプロトコルに関する統計値を表示します。

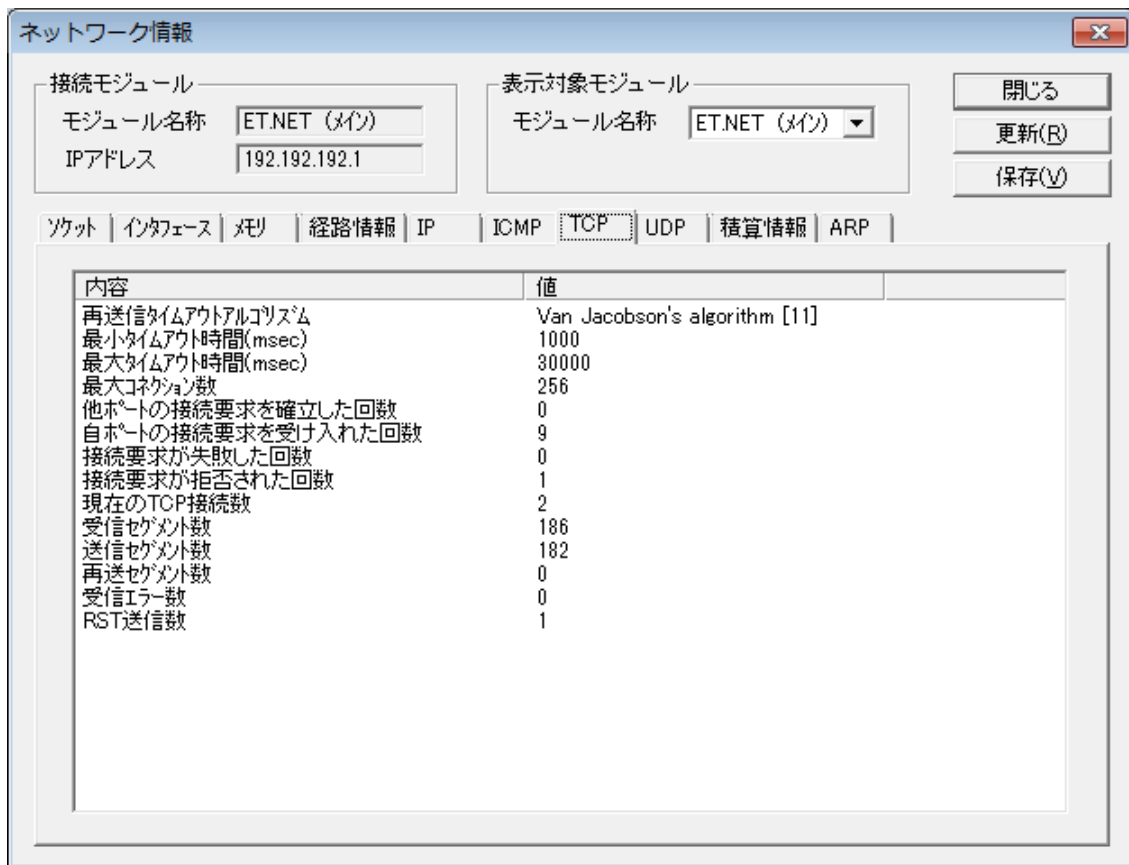


- ① ICMPメッセージ数  
ICMPメッセージの総数が表示されます。
- ② ICMPエラーメッセージ数  
エラーICMPメッセージの総数が表示されます。
- ③ Destination Unreachableメッセージ数  
宛先へ送信できなかったことを表すICMPメッセージの総数が表示されます。
- ④ Time Exceededメッセージ数  
TTL (Time To Live) 不足のため、ルーティング途中で破棄されたことを表すICMPメッセージの総数が表示されます。
- ⑤ Parameter Problemsメッセージ数  
ICMPメッセージのパラメータエラーを表すICMPメッセージの総数が表示されます。
- ⑥ Source Quenchesメッセージ数  
受信側のリソース不足のため送信の抑制を要求するICMPメッセージの総数が表示されます。
- ⑦ Redirectメッセージ数  
より適切な経路があることを表すICMPメッセージの総数が表示されます。

- ⑧ Echo要求メッセージ数  
pingの送信側から送るICMPメッセージの総数が表示されます。
- ⑨ Echo応答メッセージ数  
pingの受信側から返すICMPメッセージの総数が表示されます。
- ⑩ Timestamp要求メッセージ数  
Timestamp要求のICMPメッセージの総数が表示されます。
- ⑪ Timestamp応答メッセージ数  
Timestamp要求に対する応答ICMPメッセージの総数が表示されます。
- ⑫ Address Mask要求メッセージ数  
Address Mask Request要求のICMPメッセージの総数が表示されます。
- ⑬ Address Mask応答メッセージ数  
Address Mask Request要求に対する応答のICMPメッセージの総数が表示されます。

## (7) TCP統計情報

TCPプロトコルに関する統計値を表示します。



- ① 再送信タイムアウトアルゴリズム  
使用している再送信タイムアウト (RTO) アルゴリズムが表示されます。
- ② 最小タイムアウト時間 (msec)  
再送タイムアウト時間の最小値がミリ秒単位で表示されます。
- ③ 最大タイムアウト時間 (msec)  
再送タイムアウト時間の最大値がミリ秒単位で表示されます。
- ④ 最大コネクション数  
同時に接続できる最大コネクション数が表示されます。
- ⑤ 他ポートの接続要求を確立した回数  
外部へ接続を要求して、接続が確立した回数が表示されます。
- ⑥ 自ポートの接続要求を受け入れた回数  
外部からの接続要求を受け入れた回数が表示されます。
- ⑦ 接続要求が失敗した回数  
接続要求が失敗した回数が表示されます。



- ⑧ 接続要求が拒否された回数  
途中で接続が拒否された回数が表示されます。
- ⑨ 現在のTCP接続数  
現在アクティブなTCP接続の総数が表示されます。
- ⑩ 受信セグメント数  
受信したセグメント（TCPにおける1回のデータ送信単位）の総数が表示されます。
- ⑪ 送信セグメント数  
送信したセグメントの総数が表示されます。
- ⑫ 再送セグメント数  
相手から受信確認が戻ってこないため、再送信したセグメントの総数が表示されます。
- ⑬ 受信エラー数  
エラーを受信したフラグメント数が表示されます。
- ⑭ RST送信数  
リセットフラグを受信したフラグメント数が表示されます。

## (8) UDP統計情報

UDPプロトコルに関する統計値を表示します。

ネットワーク情報

接続モジュール  
 モジュール名称 ET.NET (メイン)  
 IPアドレス 192.192.192.1

表示対象モジュール  
 モジュール名称 ET.NET (メイン)

閉じる  
 更新(R)  
 保存(V)

パケット | インタフェース | メモリ | 経路情報 | IP | ICMP | TCP | **UDP** | 積算情報 | ARP

内容	値
UDPパケット受信数	89
送信先ポートがなかったUDPパケット数	89
UDPパケット受信エラー数	0
UDPパケット送信数	3

## ① UDPパケット受信数

受信したUDPパケットの総数が表示されます。

## ② 送信先ポートがなかったUDPパケット数

送信すべき上位アプリケーション（ポート番号）がなかったUDPパケット数が表示されます。

## ③ UDPパケット受信エラー数

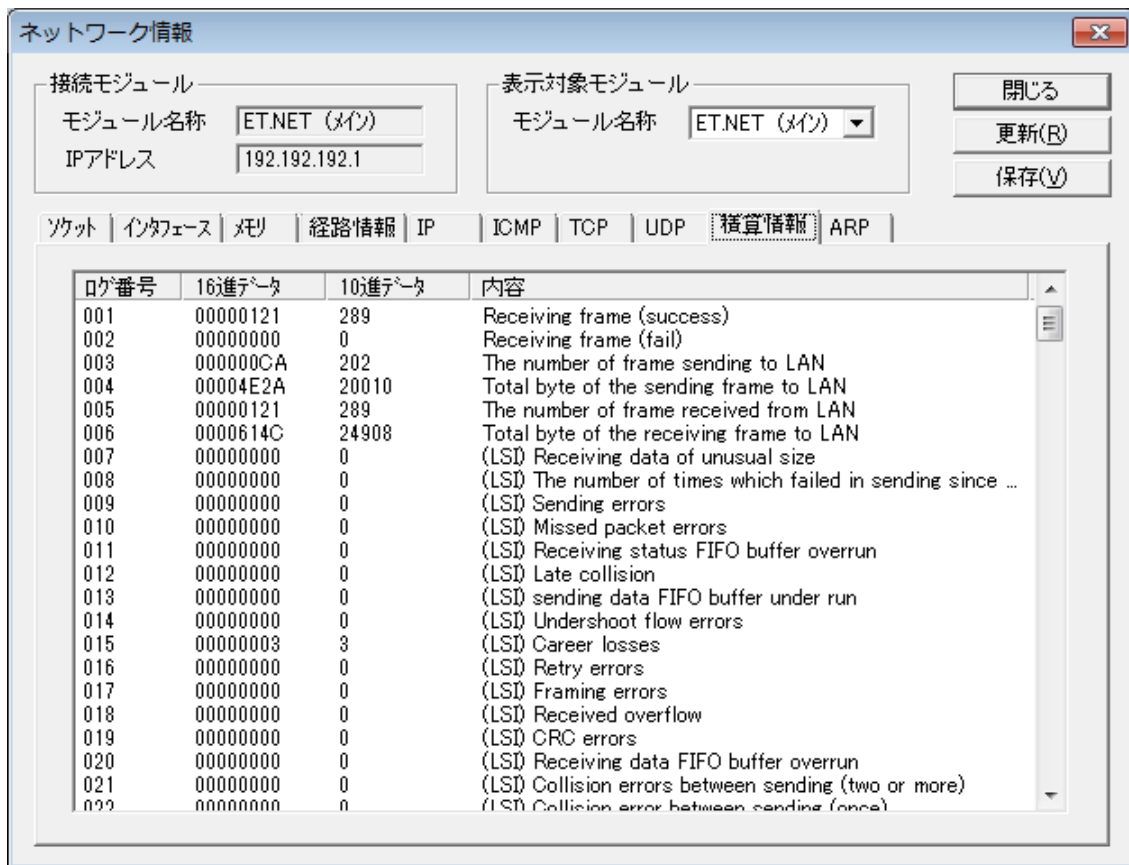
エラーやその他の理由により、上位のサービスにまで届けることができなかったUDPパケットの総数が表示されます。

## ④ UDPパケット送信数

送信したUDPパケットの総数が表示されます。

## (9) 積算情報

インタフェースの積算情報が表示されます。



## &lt;主な積算情報の内容&gt;

以下、ログ番号001～024および129に関する説明を示します。これ以外は保守用の内部情報になります。

- ログ番号001 : Receiving frame (success)  
正常受信フレーム数。
- ログ番号002 : Receiving frame (fail)  
受信時に異常が発生したフレーム数。
- ログ番号003 : The number of frame sending to LAN  
回線へ送信したフレーム数。
- ログ番号004 : Total byte of the sending frame to LAN  
回線へ送信したフレームの合計バイト数。
- ログ番号005 : The number of frame received from LAN  
回線から受信したフレーム数。この値は正常受信フレーム数および異常受信フレーム数も含まれます。
- ログ番号006 : Total byte of the receiving frame to LAN  
回線から受信したフレームの合計バイト数。

- ログ番号007 : (LSI) Receiving data of unusual size  
フレーム長が異常なフレームを受信した回数。
- ログ番号008 : (LSI) The number of times which failed in sending since 3.2msec was exceeded  
送信遅延により送信失敗 (3.2ms以上経過しても送信を開始できなかった) した回数。
- ログ番号009 : (LSI) Sending errors  
送信が途中で打ち切られた回数。
- ログ番号010 : (LSI) Missed packet errors  
通信LSI内部バッファ満杯によりパケットを取り漏らした回数。
- ログ番号011 : (LSI) Receiving status FIFO buffer overrun  
受信時に通信LSI内部の受信ステータスFIFOオーバーランが発生した回数。
- ログ番号012 : (LSI) Late collision  
送信時にレイトコリジョン (プリアンプル送出から64バイト以降でコリジョンを検出) が発生した回数。
- ログ番号013 : (LSI) sending data FIFO buffer under run  
送信時に通信LSI内部の送信データFIFOアンダーランが発生した回数。
- ログ番号014 : (LSI) Undershoot flow errors  
送信時に送信バッファアンダーフローエラーの発生した回数。
- ログ番号015 : (LSI) Career losses  
送信時にケーブル抜け、ハブの電源断などによりキャリア消失を検出した回数。
- ログ番号016 : (LSI) Retry errors  
送信時にコリジョンリトライオーバーが発生した回数。
- ログ番号017 : (LSI) Framing errors  
受信時にフレーミングエラーが発生した回数。
- ログ番号018 : (LSI) Received overflow  
受信時に受信バッファオーバーフローが発生した回数。
- ログ番号019 : (LSI) CRC errors  
受信時したフレームのCRCエラーを検出した回数。
- ログ番号020 : (LSI) Receiving data FIFO buffer overrun  
受信時に通信LSI内部の受信データFIFOオーバーランを検出した回数。
- ログ番号021 : (LSI) Collision errors between sending (two or more)  
送信時に複数回のコリジョンを検出した回数。
- ログ番号022 : (LSI) Collision error between sending (once)  
送信時に1回コリジョンを検出した回数。
- ログ番号023 : (LSI) Delay between sending  
送信時に遅延が発生した回数。送信は正常終了。
- ログ番号024 : Frame-send-timeout  
フレーム送信タイムアウトが発生した回数。

- ログ番号129 : Adapter state (top 2 byte), LINK, 10M/100Mbps, Full duplex / half-double state (bottom 2 byte)

ET.NETモジュールのデータ通信速度および全二重／半二重状態が表示されます。

以下の表を参考に確認してください。

接続種類		表示 (*)	
		16進データ	10進データ
10 Mbps	半二重	00000001	1
	全二重	00000005	5
100 Mbps	半二重	00000003	3
	全二重	00000007	7

(\*) 回線が未接続の場合、表示は0固定となります。

## (10) ARP情報

ARP (Address Resolution Protocol) で使用されているIPアドレスから物理アドレスのへの変換テーブルを表示します。



## ● ARP情報

- ユニット番号  
常に「1」が表示されます。
- インタフェース種別  
常に「EPORT」が表示されます。

## ● インタフェース情報

登録されているARPエントリが表示されます。

- ホスト名称  
IPアドレスに対応したホスト名称が表示されます。  
ホスト名称は動作しているPCのhostsファイルに登録されているもののみ表示されます。ホスト名称が未踏の場合は、「?」が表示されます。
- IPアドレス  
ARPテーブルに登録されている相手IPアドレスが表示されます。

## 7 保 守

---

- 物理アドレス

ARPテーブルに登録されている相手物理アドレスが表示されます。ARPエントリが無効な場合、「(incomplete)」が表示されます。

- ステータス

ARPエントリの状態ステータスが表示されます。

ステータスには以下の種類があります。

表示	内容
permanent	固定エントリ
published	代理応答エントリ

## 7. 3. 4 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7-8 ソケットハンドラ検出のエラーコード

(1/2)

エラーコード	内 容	原 因	対 策
F000	接続 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、接続確立後にハンドラを再発行してください。
F010	ソケットID 不正	<ul style="list-style-type: none"> <li>ソケットIDが範囲外 (TCP : /01 ≤ ID ≤ /18、 UDP : /20 ≤ ID ≤ /37)</li> <li>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。</li> <li>未接続、接続が確立されていません。または既に接続が確立されています (tcp_accept()のみ)。</li> </ul>	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とUDP合計で24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()から再度接続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"> <li>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。</li> <li>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)。</li> </ul>	tcp_close()を発行して接続切断後、tcp_open()またはtcp_popen()から再度接続確立をしてください。再度接続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。 tcp_close()にて発生した場合は、tcp_abort()を発行して接続切断後、tcp_open()またはtcp_popen()から再度接続確立をしてください。
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。	モジュールが停止している可能性があります。モジュールを交換してください。
F020	送信データ長 不正	送信データ長が制限値を満足していません。 (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足していません。 (1 ≤ データ長 ≤ 4,096)	ユーザプログラムを見直してください。



エラーコード	内 容	原 因	対 策
F0FF	ポート開放	<ul style="list-style-type: none"> <li>・ハンドラ起動後、ポート開放状態（RST受信）になりました（<code>tcp_open()</code>）。</li> <li>・ハンドラ起動時、ポート開放状態でした（<code>tcp_send()</code>または<code>tcp_receive()</code>）。</li> </ul>	<ul style="list-style-type: none"> <li>・<code>tcp_open()</code>または<code>tcp_popen()</code>から再度コネクション確立をしてください。</li> <li>・<code>tcp_close()</code>を発行後、<code>tcp_open()</code>または<code>tcp_popen()</code>から再度コネクション確立をしてください。</li> </ul>
FFF0	アドレス不正	<ul style="list-style-type: none"> <li>・<code>udp_open()</code>、<code>udp_send()</code>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。</li> <li>・経路情報が未設定の相手局へ送信しました（<code>udp_send()</code>のみ）。</li> </ul>	<ul style="list-style-type: none"> <li>・ユーザプログラムを見直してください。</li> <li>・経路情報を設定してください。</li> </ul>
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（ソケットハンドラの引数 <code>padr</code> 、 <code>buf</code> 、 <code>outinf</code> 、 <code>tim</code> の設定値）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。	<code>tcp_close()</code> を発行してコネクション切断後、 <code>tcp_open()</code> または <code>tcp_popen()</code> 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。
FFF8	FIN受信	相手局からFINを受信しました。	<code>tcp_close()</code> 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 <code>tcp_receive()</code> を発行しました）。	<code>tcp_close()</code> を発行してコネクション切断後、 <code>tcp_open()</code> または <code>tcp_popen()</code> 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。自局から <code>tcp_close</code> でソケット終了した場合は発生する可能性があります。（*）
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（ <code>tcp_close()</code> または <code>udp_close()</code> 発行）、 <code>tcp_open()</code> または <code>tcp_popen()</code> から再度コネクション確立をしてください。
FFFF	内部バッファ不足	<ul style="list-style-type: none"> <li>・送信バッファが満杯になりました（<code>udp_send()</code>）。</li> <li>・内部登録エリアが満杯になりました（<code>route_add()</code>、<code>arp_list()</code>）。</li> </ul>	<ul style="list-style-type: none"> <li>・一定時間後に再度<code>udp_send()</code>を発行してください。</li> <li>・不要になった登録情報を削除してから再度発行してください。</li> </ul>

- (\*) `tcp_open()`または`tcp_popen()`にて自局のポート番号を指定してオープンしたソケットを、`tcp_close()`にて自局からクローズした場合、ソケットはTIMEWAITという状態になります（「7.3.3 ネットワーク情報の見方」の「図7-2 接続ステータス状態遷移」を参照）。そのソケットは、`tcp_close()`終了後約20秒間クローズされません。ソケットがTIMEWAIT状態で、再度同じポート番号を指定して`tcp_open()`または`tcp_popen()`を発行すると二重ソケットエラー（エラーコード=FFFD）が発生します。その場合は、下記のうちいずれかの方法にて対策してください。
- `tcp_close()`後20以上間隔を空けてから`tcp_open()`または`tcp_popen()`を発行する。
  - `tcp_abort()`にてソケットをクローズする。
  - `tcp_open()`または`tcp_popen()`にて自局のポート番号を指定せずにソケットをオープンする。  
（ポート番号は1024～2047のうち未使用の番号が任意に確保されます）。

## 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	時 分
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						