

HI-NET

ユーザーズマニュアル

2023年 10月 第5版

目次

	頁
第1章 概要	1
1. 1 HI-NET の構成	1
1. 2 HI-NET の特徴	1
1. 3 通信仕様	2
1. 4 伝送制御文字	5
第2章 各手順の概要	6
2. 1 BSC手順の概要	6
2. 2 レベル2 A / 2 Bクラス2の概要 < レベル2 Aクラス1を使用する方法 >	8 9
2. 3 レベル2 A / 2 Bクラス3の概要	10
2. 4 誤り制御	12
第3章 言語インターフェイス	13
3. 1 各手順の機能一覧表	13
3. 2 各関数の説明	16
1) ボードの初期化	[initial] 16
2) 回線のオープン	[bscopen] 20
・ "	[nec2open] 23
・ "	[nec3open] 26
3) 電文の送信	[bscsend] 29
・ "	[nec2send] 33
・ "	[nec3send] 37
4) 送信のチェック	[sendck] 41
5) 電文の受信	[hrecv] 42
6) "	[hrecvx] 44
7) 受信データの件数をチェック	[rdtck] 46
8) 受信バッファのクリア	[rbfcclr] 47
9) 送信バッファのクリア	[sbfcclr] 48
10) 回線クローズ	[chclose] 49
11) 自動ダイヤル	[telon] 50
12) 手動電話接続	[mtelon] 52
13) 電話の切断	[teloff] 53
14) バージョンの取得	[verget] 54
15) タイマー値の変更	[timerset] 55
16) リトライ値の変更	[retryset] 58
17) パッドの変更	[padset] 60
18) バッファサイズの変更	[bsizeset] 62
19) SYNCキャラクタの変更	[syncset] 63
20) IDの返却 (BSC手順のみ)	[ridget] 64
21) ポーリングの許可 (レベル2 A / 2 Bクラス3のみ)	[pollon] 66
22) ポーリングの禁止 (レベル2 A / 2 Bクラス3のみ)	[polloff] 67
23) ポーリング状態の取得 (レベル2 A / 2 Bクラス3のみ)	[pollstat] 68
第4章 ステータス一覧表	69
付 録	
1) HI-NET / USB の設定	73
2) ケーブル配線図	74
3) ドライバーのインストール方法	75
4) コード変換表	85
5) 安全にお使いいただくために必ずお守りください	87

第1章 概要

HI-NETはパーソナルコンピュータとホストコンピュータでオンライン通信する場合に、通信手順を特に意識することなく高度なオンラインプログラムが作成できる通信手順ライブラリです。

1. 1 HI-NETの構成

HI-NETは次の内容で構成されています。

- (1) 手順ライブラリ (HI-NET上のROM内)
- (2) 言語インターフェイス (添付のCD-ROM内)

1. 2 HI-NETの特徴

- ・CPUを搭載したインテリジェントボードです。
- ・RS-232Cを2チャンネル持っています。
(HI-NET/USBは1チャンネル)
- ・同時に8回線までのオンライン通信が可能です。
- ・送信バッファ、受信バッファとして各々64キロバイトを持っています。
(各バッファは、標準2048+10バイトのブロックで管理されます。)
- ・インターフェイスの形式で下記の2種類があります。

インターフェイス	製品名	サイズ (mm)
USB	HI-NET/USB	154 × 95 × 34 (ケース)
PCIバス	HI-NET/PCI	175 × 107 (ハイプロファイル・ショートサイズ) (ブラケットを含まず)

1. 3 通信仕様

1. 3. 1 BSC手順

適 応 回 線	専用回線／公衆回線
同 期 方 式	調歩同期式またはSYN同期式
回 線 速 度	(調歩同期式) 150～ 9600 bps (SYN同期式) 150～ 9600 bps
通 信 方 式	半二重通信 (2線式または4線式)
接 続 方 式	コンテンション方式
応 答 方 式	(調歩同期式) 絶対ACK, NAK 応答 (SYN同期式) ACK0, ACK1の交互応答 絶対NAK 応答
伝 送 制 御 コード	EBCDIC／ASCIIの選択 標準 (SYN同期式) 8ビットパリティ無し (調歩同期式) 8ビットパリティ無し
伝 送 コード	EBCDIC／JIS
透 過 方 式	透過／非透過
誤 り 制 御	CRC16／CRC-ITU-T／LRCの選択可 標準 (調歩同期式) LRC (SYN同期式) CRC16
自 動 ダイヤル	V25.bis 対応

1. 3. 2 レベル2 A / 2 Bクラス2手順

適 応 回 線	専用回線／公衆回線
同 期 方 式	(レベル2 Aクラス2) 調歩同期式 (レベル2 Bクラス2) SYN同期式
回 線 速 度	(レベル2 Aクラス2) 150～ 9600 bps (レベル2 Bクラス2) 150～ 9600 bps
通 信 方 式	半二重通信 (2線式または4線式)
接 続 方 式	コンテンション方式
応 答 方 式	(レベル2 Aクラス2) 絶対ACK, NAK 応答 (レベル2 Bクラス2) ACK0, ACK1 の交互応答 絶対NAK 応答
伝 送 制 御 コ ー ド	ASCII 7ビットパリティ有り (レベル2 Aクラス2) 7ビット偶数パリティ (レベル2 Bクラス2) 7ビット奇数パリティ
伝 送 コ ー ド	EBCDIC / JIS
透 過 方 式	透過 / 非透過
誤 り 制 御	(レベル2 Aクラス2) LRC (レベル2 Bクラス2) CRC-ITU-T
自 動 ダ イ ヤ ル	V25.bis 対応

1. 3. 3 レベル2 A / 2 B クラス3 手順

適 応 回 線	専用回線 / 公衆回線
同 期 方 式	(レベル2 A クラス3) 調歩同期式 (レベル2 B クラス3) SYN同期式
回 線 速 度	(レベル2 A クラス3) 150 ~ 9600 bps (レベル2 B クラス3) 150 ~ 9600 bps
通 信 方 式	半二重通信 (2線式または4線式)
接 続 方 式	ポーリング / セレクティング方式
応 答 方 式	(レベル2 A クラス3) 絶対ACK, NAK 応答 (レベル2 B クラス3) ACK0, ACK1 の交互応答 絶対NAK 応答
伝 送 制 御 コード	ASCII 7ビットパリティ有り (レベル2 A クラス3) 7ビット偶数パリティ (レベル2 B クラス3) 7ビット奇数パリティ
伝 送 コード	EBCDIC / JIS
透 過 方 式	透過 / 非透過
誤 り 制 御	(レベル2 A クラス3) LRC (レベル2 B クラス3) CRC-ITU-T
自 動 ダイヤル	V25.bis 対応

1. 4 伝送制御文字

符 号	S Y N同期式		調歩同期式		意 味
	EBCDIC	ASCII	EBCDIC	ASCII	
STX	0 2	0 2	0 2	0 2	電文開始 (透過モードではDLE・STX)
ETX	0 3	0 3	0 3	0 3	電文終了 (透過モードではDLE・ETX)
ETB	2 6	1 7	2 6	1 7	ブロック終了 (透過モードではDLE・ETB)
EOT	3 7	0 4	3 7	0 4	伝送終了
SYN	3 2	1 6	—	—	同期文字 (S Y N同期式)
ENQ	2 D	0 5	2 D	0 5	問い合わせ
ACK	—	—	2 E	0 6	肯定応答 (調歩同期式)
ACK0	10・70	10・30	—	—	肯定応答0 (S Y N同期式)
ACK1	10・61	10・31	—	—	肯定応答1 (S Y N同期式)
SOH	0 1	0 1	0 1	0 1	ヘディング開始
WACK	10・6B	10・3B	10・6B	10・3B	送信待機要求肯定応答 (B S C手順のみ)
WABT	—	10・3F	—	10・3F	受信の一時中断要求 (レベル2 A / 2 Bのみ)
RVI	10・7C	10・3C	10・7C	10・3C	送信局の反転
中断	—	10・7C	—	10・7C	中断 (レベル2 A / 2 Bのみ)
NAK	3 D	1 5	3 D	1 5	否定応答
DLE	1 0	1 0	1 0	1 0	伝送制御拡張
PADL	A A	A A	A A	A A	リーディングパッド ※padset 関数で変更可能
PADT	F F	F F	F F	F F	トレーリングパッド ※padset 関数で変更可能

※レベル2 Aは調歩同期式、レベル2 BはS Y N同期式です。

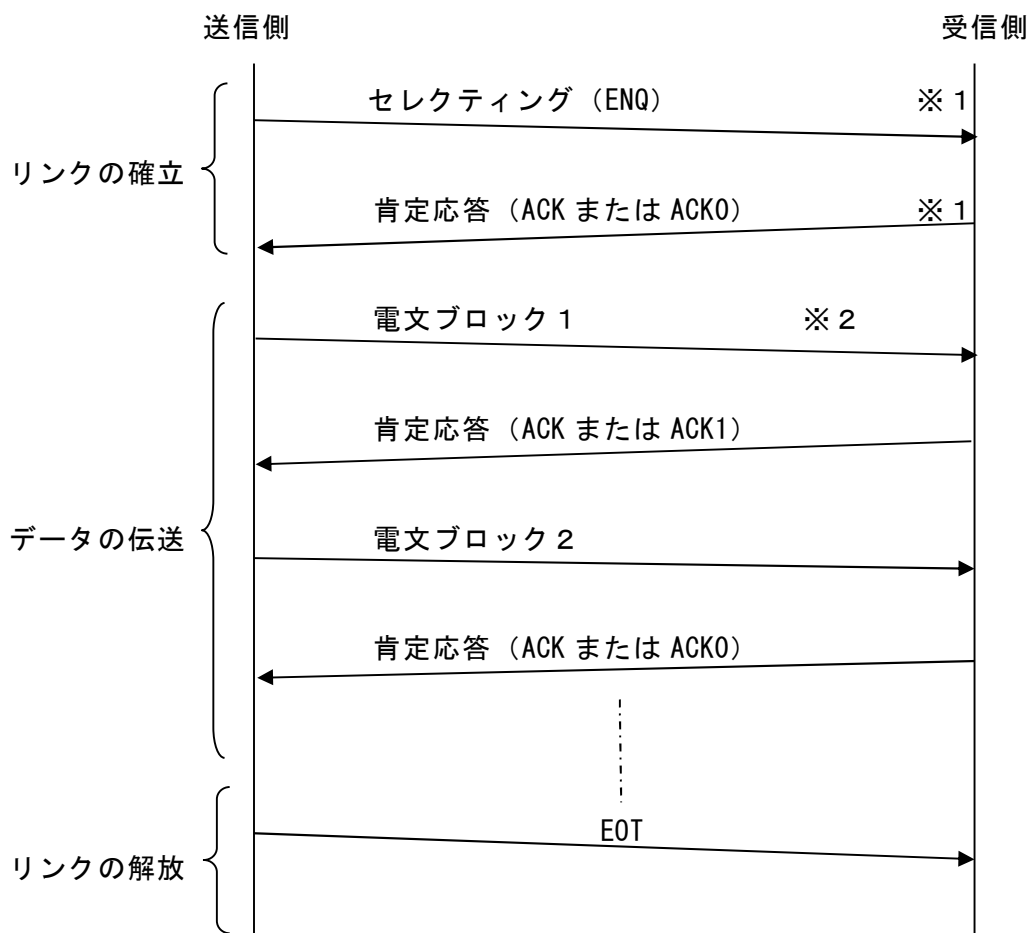
※レベル2 A / 2 Bクラス2、レベル2 A / 2 Bクラス3の伝送制御文字はA S C I Iです。

※伝送上では、レベル2 Aでは偶数パリティ、レベル2 Bでは奇数パリティが付加されます。

第2章 各手順の概要

2.1 BSC手順の概要

2.1.1 データ通信のシーケンス



※1 ID交換をした場合は、セレクトディングのENQの前に自局ID、セレクトディングに対するACKの前に相手局IDが付加されます。

※2 STX+電文+ETB または ETX+BCC

2.1.2 リンクの確立

データを送信する側が、相手局に対してセレクトディング (ENQ) を送出し、今からデータを送る意思を伝えます。

セレクトディングを受信した側は、受信が可能であれば、肯定応答 (調歩同期式: ACK、SYN同期式: ACK0) を返します。

2. 1. 3 データの伝送

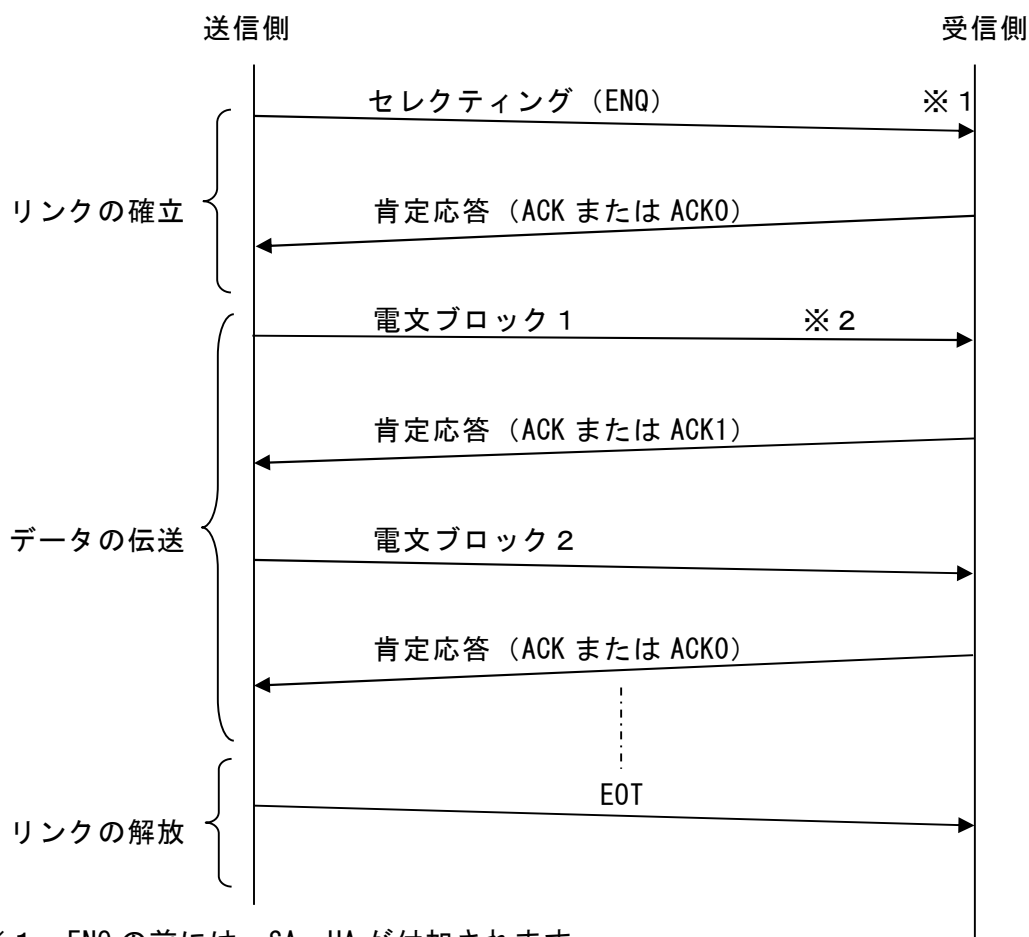
- (1) 送信は、送信データに伝送上の制御符号を付加し相手局に送ります。
- (2) 受信側は、正しくデータを受信すれば送信側に肯定応答を返します。送信側は、次に送るデータがあれば続いてデータを送り、無ければリンクの解放を行います。

2. 1. 4 リンクの解放

送信側は、相手局に対して送信が終了したことを知らせます。(EOTの送出)

2. 2 レベル 2 A / 2 B クラス 2 手順の概要

2. 2. 1 データ通信のシーケンス



※ 1 ENQ の前には、SA、UA が付加されます。

※ 2 STX+SA+UA+電文+ETB または ETX+BCC

2. 2. 2 リンクの確立

データを送信する側が、相手局に対してセレクトィング (ENQ) を送出し、今からデータを送る意思を伝えます。

セレクトィングを受信した側は、受信が可能であれば、肯定応答 (レベル 2 A クラス 2 : ACK、レベル 2 B クラス 2 : ACK0) を返します。

2. 2. 3 データの伝送

- (1) 送信側は、送信データに伝送上の制御符号を付加し相手局に送ります。
- (2) 受信側は、正しくデータを受信すれば送信側に肯定応答を返します。送信側は、次に送るデータがあれば続いてデータを送り、無ければリンクの解放を行います。

2. 2. 4 リンクの解放

送信側は、相手局に対して送信が終了したことを知らせます。(EOTの送出)

※ <レベル2 Aクラス1を使用する方法>

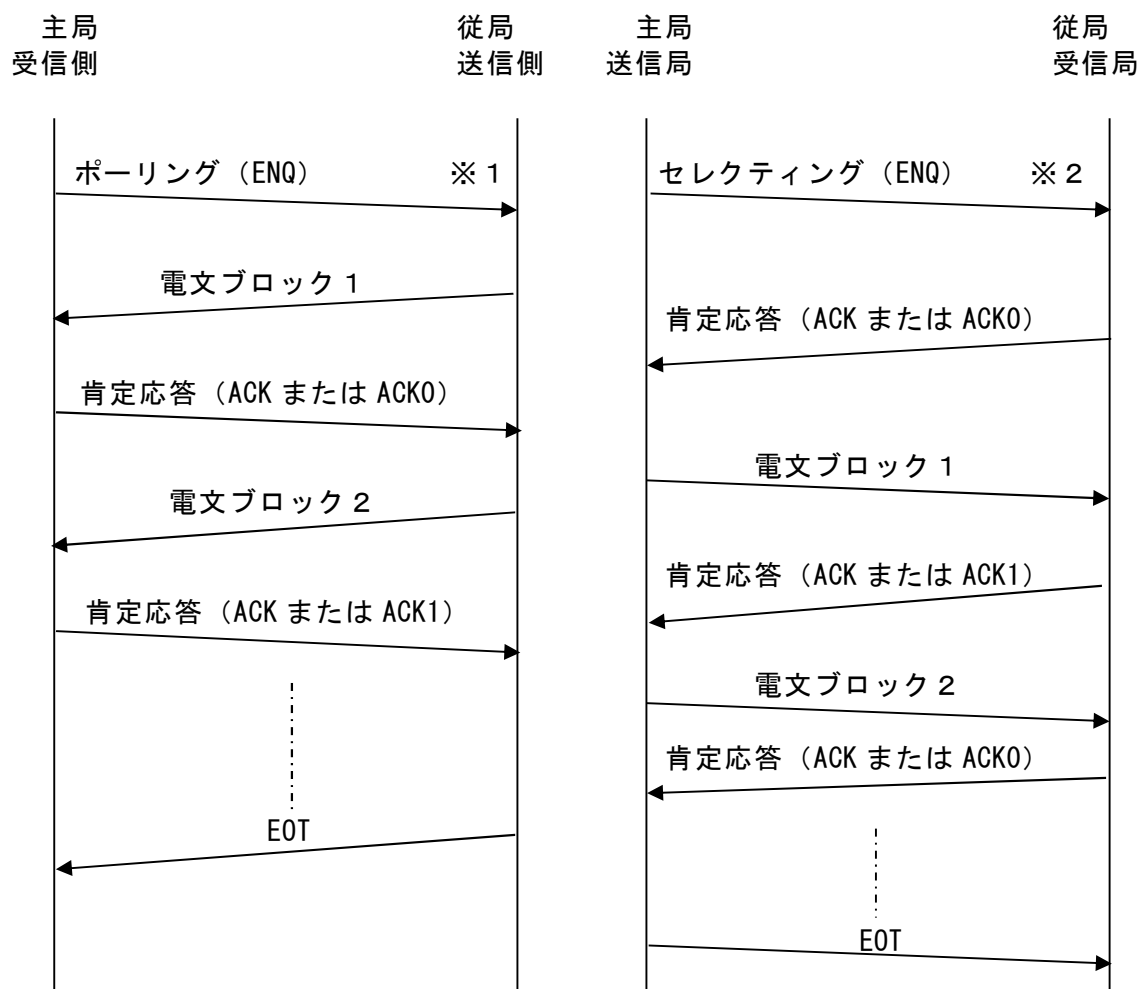
レベル2 Aクラス1は、セレクトイングのみにSA、UAが付き、電文ブロック前にはSA、UAが付かない手順ですので、関数 `nec2open` 以外はレベル2 Aクラス2と同様に使用してください。

2. 3 レベル 2 A / 2 B クラス 3 手順の概要

2. 3. 1 データ通信のシーケンス

(ポーリング側)

(セレクティング側)



※1 ポーリングの ENQ の前、電文ブロック 1 の STX の後ろには、SA 及び主局側の入力 UA (従局側の出力 UA) が付加されます。

※2 セレクティングの ENQ の前、電文ブロック 1 の STX の後ろには、SA 及び主局側の出力 UA (従局側の入力 UA) が付加されます。

2. 3. 2 リンクの確立

(1) ポーリングの送出 (主局)

主局は従局に対して1局ずつ送信要求の有無を確かめていきます。

S A + 主局側の入力 U A + E N Q
(従局側の出力 U A)

(2) ポーリングに対する応答 (従局)

- ・従局は送信データがあれば、STX から始まる電文を送出します。
- ・従局は送信すべきデータが無い場合、EOT を送じます。

※従局が無応答の場合は、主局はポーリングの送信または EOT を送信してデータリンクの解放を行います。

(3) セレクティングの送出 (主局)

主局は従局に対して送信データがあるときは、その局に対し受信要求を行います。

S A + 主局側の出力 U A + E N Q
(従局側の入力 U A)

(4) セレクティングに対する応答 (従局)

従局はデータを受信できるなら肯定応答 (ACK) を返します。

2. 3. 3 データの伝送

- (1) 送信側は、送信データに伝送上の制御符号を付加し相手局に送ります。
- (2) 受信側は、正しくデータを受信すれば送信側に肯定応答を返します。
送信側は、次に送るデータがあれば続いてデータを送り、無ければリンクの解放を行います。

2. 3. 4 リンクの解放

送信側は、相手局に対して送信が終了したことを知らせます。(EOT の送出)

2. 4 誤り制御

2. 4. 1 誤り制御の種類

通信手順では、データの信頼性を高めるために電文ブロックの後に BCC を付加します。H I - N E T では以下の 3 つの BCC が使用できますが、B S C 手順のみ選択できます。

レベル 2 A クラス 2、レベル 2 A クラス 3 は【L R C 固定】です。

レベル 2 B クラス 2、レベル 2 B クラス 3 は【C R C - I T U - T 固定】です。

(1) C R C - 1 6

BCC は下記の生成多項式で計算された 16 ビットの値です。

$$X^{16} + X^{15} + X^2 + 1$$

(2) C R C - I T U - T

BCC は下記の生成多項式で計算された 16 ビットの値です。

$$X^{16} + X^{12} + X^5 + 1$$

(3) L R C (水平パリティ)

電文ブロックに含まれる文字のビットごとにパリティを取り、BCC は 1 バイトです。

2. 4. 2 B C C の計算対象範囲

(1) 非透過モードの場合

- ・SOH または STX により計算が開始され、これらの文字は計算の対象となりません。但し、SOH に続く STX は計算の対象となります。
- ・計算は ETX または ETB で終了し、これらの文字は計算の対象となります。
- ・電文途中に挿入されたタイムフィル (SYN) は、計算の対象となりません。

(2) 透過モードの場合

- ・SOH、DLE・STX により計算が開始され、これらの文字は計算の対象となりません。但し、SOH に続く DLE・STX は計算の対象となります。
- ・計算は DLE・ETX または DLE・ETB で終了し、これらの文字の内、DLE は計算の対象とならず、ETX、ETB は計算の対象となります。
- ・電文の途中に挿入されたタイムフィル (DLE・SYN) は計算の対象となりません。

第3章 言語インターフェイス

3.1 機能一覧

3.1.1 BSC手順

H I - N E T (B S C 手順) には下記の関数が用意されています。

initial	H I - N E T を使用可能にします。
bscopen	B S C 手順の回線をオープンし、通信可能の状態にします。
bscsend	送信データを送信バッファに入れます。 送信バッファに入ったデータは自動的に送信されます。
sendck	送信終了のチェックを行います。
hrecv	受信バッファから受信データを取り出します。
hrecvx	受信バッファから受信データを取り出します。 (受信データ長が0の受信データも取り出します。)
chclose	回線のクローズを行います。
rdtck	受信データの件数を確認します。
rbfclr	受信バッファをクリアします。
sbfclr	送信バッファをクリアします。
telon	指定した電話番号に自動ダイヤルします。
mtelon	手動発信及び自動着信で電話回線の接続を行います。
teloff	接続されている電話回線を切断します。
timerset	各タイマー値を既定値から変更します。
retryset	各リトライ値を既定値から変更します。
padset	リーディングパッド、トレーリングパッドを既定値から変更します。
bsizeset	送信、受信データバッファの1ブロックのサイズを変更します。
syncset	S Y N C キャラクタとその数を変更します。
ridget	B S C 手順で受信した ID を取得します。
verget	H I - N E T のバージョンを取得します。

3. 1. 2 レベル2 A / 2 Bクラス2手順

H I - N E T (レベル2 A / 2 Bクラス2手順)には下記の関数が用意されています。

initial	H I - N E Tを使用可能にします。
nec2open	レベル2 A / 2 Bクラス2の回線をオープンし、通信可能の状態にします。
nec2send	送信データを送信バッファに入れます。 送信バッファに入ったデータは自動的に送信されます。
sendck	送信終了のチェックを行います。
hrecv	受信バッファから受信データを取り出します。
hrecvx	受信バッファから受信データを取り出します。 (受信データ長が0の受信データも取り出します。)
chclose	回線のクローズを行います。
rdtck	受信データの件数を確認します。
rbfclr	受信バッファをクリアします。
sbfclr	送信バッファをクリアします。
telon	指定した電話番号に自動ダイヤルします。
mtelon	手動発信及び自動着信で電話回線の接続を行います。
teloff	接続されている電話回線を切断します。
timerset	各タイマー値を既定値から変更します。
retryset	各リトライ値を既定値から変更します。
padset	リーディングパッド、トレーリングパッドを既定値から変更します。
bsizeset	送信、受信データバッファの1ブロックのサイズを変更します。
syncset	S Y N Cキャラクタとその数を変更します。
verget	H I - N E Tのバージョンを取得します。

3. 1. 3 レベル2 A / 2 Bクラス3手順

H I - N E T (レベル2 A / 2 Bクラス3手順)には下記の関数が用意されています。

initial	H I - N E Tを使用可能にします。
nec3open	レベル2 A / 2 Bクラス3の回線をオープンし、通信可能の状態にします。
nec3send	送信データを送信バッファに入れます。 送信バッファに入ったデータは自動的に送信されます。
sendck	送信終了のチェックを行います。
hrecv	受信バッファから受信データを取り出します。
hrecvx	受信バッファから受信データを取り出します。 (受信データ長が0の受信データも取り出します。)
chclose	回線のクローズを行います。
rdtck	受信データの件数を確認します。
rbfclr	受信バッファをクリアします。
sbfclr	送信バッファをクリアします。
telon	指定した電話番号に自動ダイヤルします。
mtelon	手動発信及び自動着信で電話回線の接続を行います。
teloff	接続されている電話回線を切断します。
timerset	各タイマー値を既定値から変更します。
retryset	各リトライ値を既定値から変更します。
padset	リーディングパッド、トレーリングパッドを既定値から変更します。
bsizeset	送信、受信データバッファの1ブロックのサイズを変更します。
syncset	S Y N Cキャラクタとその数を変更します。
pollon	指定したユニットアドレスへのポーリングを開始します。
polloff	指定したユニットアドレスのポーリングを停止します。
pollstat	指定した入力ユニットアドレスのポーリング状態を取得します。
verget	H I - N E Tのバージョンを取得します。

3. 2 各関数の説明

initial (HI-NET/PCI)

機 能

HI-NETを使用可能にします。

書 式

ステータス = initial (I/Oアドレスの組) ;

解 説

(1) I/Oアドレスの組

- ・ HI-NET/PCIの場合、プラグ&プレイに対応していますので、自動的にI/Oアドレスが設定されます。
0以外の値を指定してください。
(ボード4枚分までの設定が可能です。)

(2) ステータス

- ・ ボードNo. 0~3 上位8ビット (0 : 1枚目, 1 : 2枚目…)
- ・ ステータス 0~2 下位8ビット

0 : 正常終了

1 : HI-NETのリセットに失敗した

原因① I/Oアドレスの設定が間違っている

② I/Oアドレスの設定が他のボードの
I/Oアドレスとぶつかっている。

2 : HI-NETの起動に失敗した

原因① HI-NETの故障

※複数枚のHI-NETを使用して、複数でエラーがあった場合は、エラーのあった最も小さいボードNo.のエラーが返ります。

※I/Oアドレス(4組分)によって、以降のチャンネル番号が決まります。
1枚目はチャンネル0と1になり、2枚目はチャンネル2と3…となります。

文 例 H I - N E T / P C I を 2 枚 入 れ た 場 合

```

(C言語)  int sts;                               /*ステータス*/
          int bdno;                             /*ボードNo*/
          int erst;                            /*エラー番号*/
          WORD inport_address[4];             /*I/Oアドレス(4組分)*/
          inport_address[0]=0x1e0;           /*1枚目のI/Oアドレスを1E0Hに
                                           設定*/
          inport_address[1]=0x2e0;           /*2枚目のI/Oアドレスを2E0Hに
                                           設定*/

          inport_address[2]=0x0000;          /*3枚目は使用しない*/
          inport_address[3]=0x0000;          /*4枚目は使用しない*/
          sts = initial(inport_address);
          if ( sts != 0x0000 ) {
              bdno = sts/256 ;                /*ボード番号*/
              erst = sts & 0x0f              /*エラーステータス*/
              printf( "Initial Error Board %d st = %x\n" , bdno, erst);
          }

```

```

(BASIC)  TYPE IOTYP                                'I/Oアドレス
          ADRS1      AS INTEGER
          ADRS2      AS INTEGER
          ADRS3      AS INTEGER
          ADRS4      AS INTEGER
          END TYPE

          DIM IO      AS IOTYP

          IO.ADRS1 = &H1E0                        '1枚目のI/Oアドレスを1E0Hに設定
          IO.ADRS2 = &H2E0                        '2枚目のI/Oアドレスを2E0Hに設定
          IO.ADRS3 = &H0000                       '3枚目は使用しない
          IO.ADRS4 = &H0000                       '4枚目は使用しない

          STS% = initial(IO.ADRS1)
          IF STS% <> 0 THEN
              BDNO% = STS% / 256                  ' ボード番号
              ERST% = STS% AND &H0F              ' エラーステータス
              PRINT " Initial Error Board =" ; BDNO% ; "St=" ; ERST%
          END IF

```

initial (HI-NET/USB)

機能

HI-NETを使用可能にします。

書式

ステータス = initial (チャンネル番号)

解説

(1) チャンネル番号

HI-NET/USBのチャンネル番号(0~7)を指定してください。

※出荷時のチャンネル番号は0で設定されています。

※同じアプリケーションで複数のHI-NETを使用する場合、
各チャンネル毎に initial 関数を呼んでください。

※チャンネル番号はHI-NETのディップスイッチで設定できます。
詳しくは「HI-NET/USBの設定」の項を参照してください。

(2) ステータス

0 : 正常終了

- 1 : 指定したチャンネル番号が誤っています。

原因 接続していないHI-NETのチャンネル番号を指定した場合

8057 : イニシャライズを既に行っているにもかかわらず、再度
イニシャライズを実行した

以外 : 初期化に失敗しました。

原因 HI-NETの故障

文 例

```
(C言語) int sts;                               /*ステータス*/  
char chn;                                       /*チャンネル番号*/  
chn = 0x00;  
sts = initial(chn);  
if ( sts != 0x0000 ) {  
    printf( "Initial Error st = %x¥n" , sts);  
}
```

```
(BASIC) CHN% = 0  
STS% = initial(CHN%)  
IF STS% <> 0 THEN  
    PRINT " Initial Error St=" ; HEX$(STS%)  
END IF
```

bscopen

機 能

B S C手順の回線をオープンし、通信可能な状態にします。

書 式

ステータス = bscopen(チャンネル番号, 自局 ID, 相手局 ID, 回線モード, 同期方式, 通信速度)

解 説

(1) チャンネル番号

H I - N E Tのチャンネル番号(0~7)を指定してください。

(2) 自局 ID

相手局に送信する自局 ID を設定します。
ID 交換しない場合は、NULL を入れてください。

(3) 相手局 ID

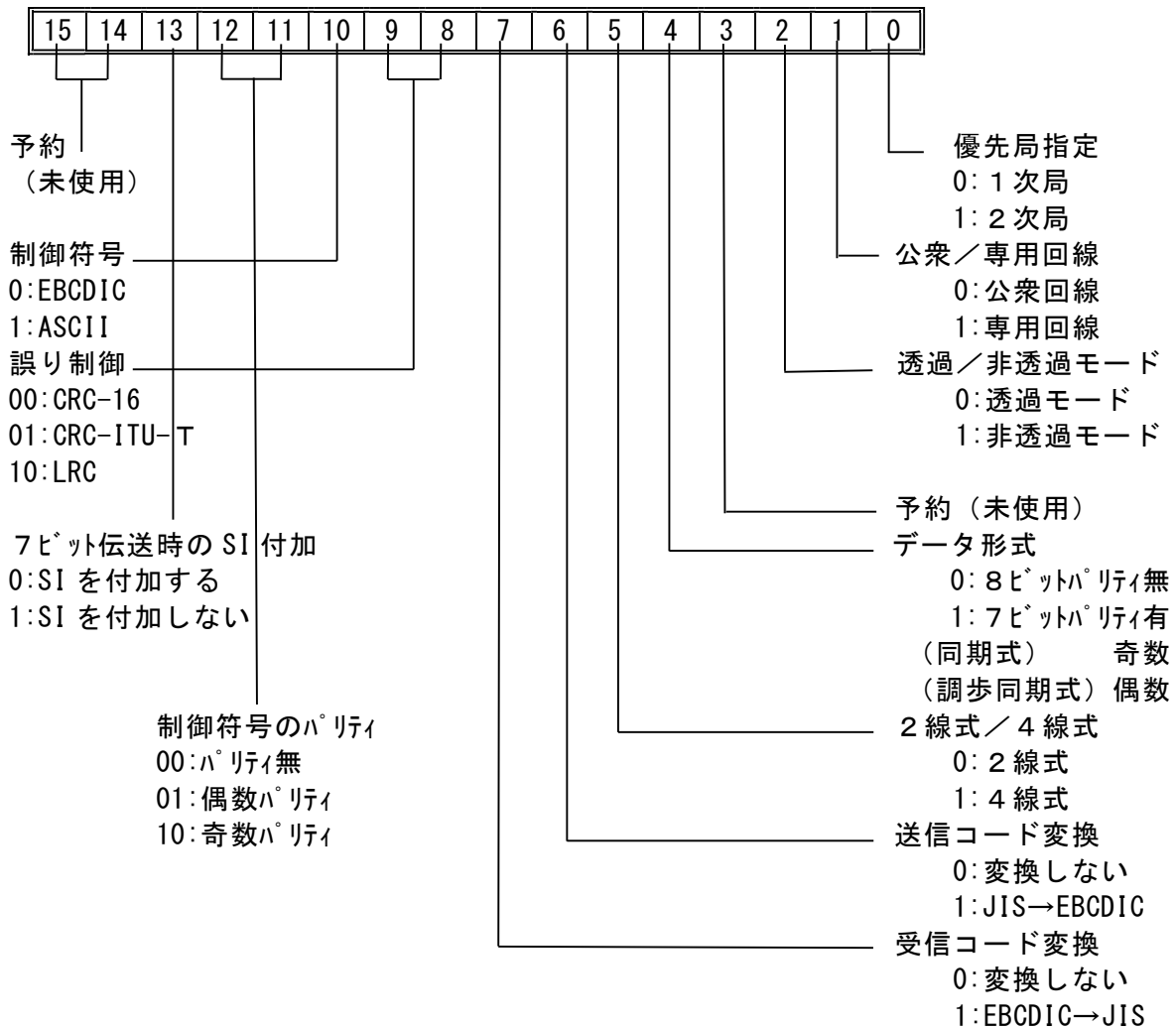
相手局の ID を設定します。
ID 交換しない場合は、NULL を入れてください。

※ID 交換は、公衆回線モードのみ行え、自局 ID は送信時の ENQ の前、相手局からの ENQ に対する ACK の前に付加されます。
また、相手局 ID は、相手局からの ENQ の前に付加されている ID との照合に使用されます。

※設定された ID は、回線モードのコード変換の指定に関わらず、コード変換はされず、設定された通り送付されます。

(4) 回線モード

回線モードは、2バイトで下記のビットで指定してください。



(5) 同期方式

- 0 : 調歩同期式
- 1 : ST1同期式 (内部同期)
(SYN同期式でモデムを使用しない場合)
- 2 : ST2同期式 (外部同期)
(SYN同期式でモデムを使用する場合)

(6) 通信速度

- | | |
|----------|----------|
| 0 : 150 | 4 : 2400 |
| 1 : 300 | 5 : 4800 |
| 2 : 600 | 6 : 9600 |
| 3 : 1200 | |

※ST2同期式の場合は、モデムの通信速度設定によって決まります。

(7) ステータス

0000H 正常終了

0000H以外は、エラーが発生しました。
「ステータス一覧表」を参照してください。

文 例

```
(C言語) int sts, mode, syn, bps;
char id1[256], id2[256];
char chn;
chn = 0x00; /*チャンネル0を指定*/
strcpy(id1, "ABC"); /*自局IDにABCを指定*/
strcpy(id2, "XYZ"); /*相手局IDにXYZを指定*/
mode = 0x0e3; /*回線モードの指定*/
syn = 2; /*ST2同期を指定*/
bps = 6; /*9600bpsを指定*/

sts = bscopen(chn, id1, id2, mode, syn, bps);
if ( sts != 0x0000 ) {
    printf( "Open Error st = %x¥n" , sts);
}
```

```
(BASIC) CHN% = 0 'チャンネル0を指定
ID1$ = "ABC" '自局IDにABCを指定
ID2$ = "XYZ" '相手局IDにXYZを指定
MODE% = &HE3 '回線モード指定
SYN% = 2 'ST2同期を指定
BPS% = 6 '9600bpsを指定

STS% = bscopen(CHN%, ID1$, ID2$, MODE%, SYN%, BPS%)
IF STS% <> 0 THEN
    PRINT " Open Error St =" ; HEX$(STS%)
END IF
```


nec2open

機能

レベル 2 A / 2 B クラス 2 の回線をオープンし、通信可能の状態にします。

書式

ステータス = nec2open (チャンネル番号, 通信手順, 通信速度, 入力 SA・UA 組, 回線モード)

解説

(1) チャンネル番号

H I - N E T のチャンネル番号 (0~7) を指定してください。

(2) 通信手順

0 : レベル 2 A クラス 2, レベル 2 A クラス 1 (調歩同期式)
1 : レベル 2 B クラス 2 (S T 1 同期式)
(S Y N 同期式でモデムを使用しない場合)
2 : レベル 2 B クラス 2 (S T 2 同期式)
(S Y N 同期式でモデムを使用する場合)

(3) 通信速度

0 : 150	4 : 2400
1 : 300	5 : 4800
2 : 600	6 : 9600
3 : 1200	

※ S T 2 同期式の場合は、モデムの通信速度設定によって決まります。

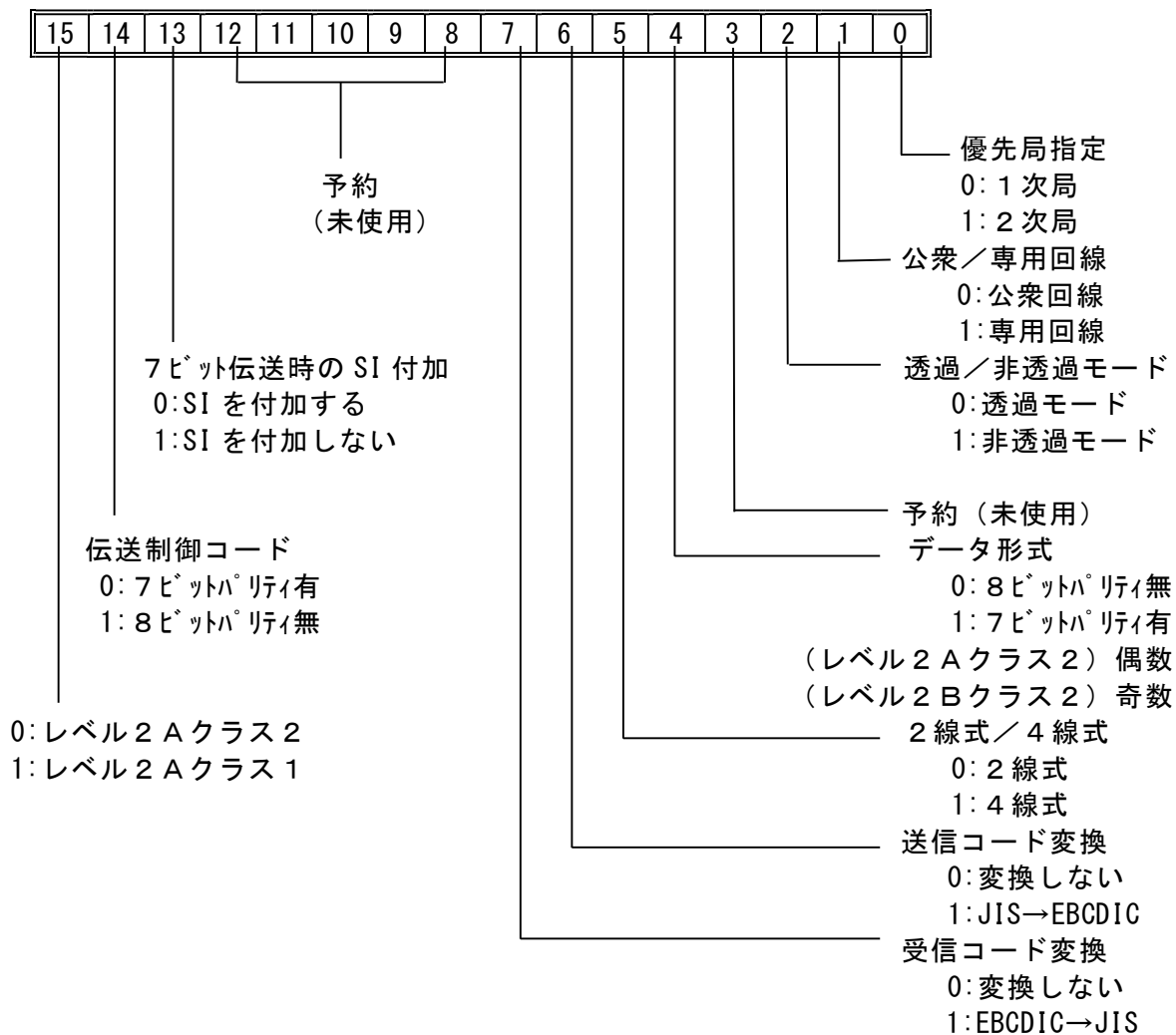
(4) 入力 SA・UA 組

入カステーションアドレス (SA)、ユニットアドレス (UA) を指定してください。

※相手局からのセレクトィングに付加される SA・UA を指定してください。
※複数組の指定が可能です。

(5) 回線モード

回線モードは、2バイトで下記のビットで指定してください。



(6) ステータス

0000H 正常終了

0000H以外は、エラーが発生しました。
「ステータス一覧表」を参照してください。

文 例

```

(C言語) int sts; /*ステータス*/
        int mode; /*回線モード*/
        char chn; /*チャンネル番号*/
        int prot; /*通信手順*/
        int bps; /*通信速度*/
        char isaua[41]; /*入力 SA・UA 組*/
        chn = 0x00; /*チャンネル 0 を指定*/
        prot = 0; /*レベル 2A クラス 2 を指定*/
        bps = 4; /*2400bps を指定*/
        isaua[0] = 0x30; /*入力 SA (1) を 30H*/
        isaua[1] = 0x20; /*入力 UA (1) を 20H*/
        isaua[2] = 0x30; /*入力 SA (2) を 30H*/
        isaua[3] = 0x21; /*入力 UA (2) を 21H*/
        isaua[4] = 0x00; /*指定の終了*/
        mode = 0xe3; /*回線モード の指定*/
        /*コード を変換する(回線上 EBCDIC),
        4 線式, データ 8 ビット, 透過モード,
        専用回線, 2 次局*/

```

```

sts = nec2open(chn, prot, bps, isaua, mode);
if( sts != 0x0000 ) {
    printf( "Open Error ch = %x St = %x\n" , 0, chn, sts);
}

```

```

(BASIC) CHN% = 0 'チャンネル 0 を指定
        PROT% = 0 'レベル 2A クラス 2 を指定
        BPS% = 4 '2400bps を指定
        ISAUAS$ = CHR$(&H30) + CHR$(&H20)
        ' 1 組目の入力 SA, UA を 30H, 20H と
        ' する
        ISAUAS$ = ISAUAS$ + CHR$(&H30) + CHR$(&H21)
        ' 2 組目の入力 SA, UA を 30H, 21H と
        ' する
        ISAUAS$ = ISAUAS$ +CHR$(0) '指定の終了
        MODE% = &HE3 'コード を変換する(回線上 EBCDIC),
        ' 4 線式, データ 8 ビット, 透過モード,
        ' 専用回線, 2 次局*/

```

```

STS% = nec2open(CHN%, PROT%, BPS%, ISAUAS$, MODE%)
IF STS% <> 0 THEN
    PRINT " Open Error Ch =" ;CHN%: "St=" ;HEX$ (STS%)
END IF

```

nec3open

機能

レベル 2 A / 2 B クラス 3 の回線をオープンし、通信可能の状態にします。
主局の場合はポーリングの送が始まります。

書式

ステータス = nec3open (チャンネル番号, 通信手順, 通信速度, SA, 入力 UA 組, 出力 UA 組,
回線モード)

解説

(1) チャンネル番号

H I - N E T のチャンネル番号 (0~7) を指定してください。

(2) 通信手順

- 0 : レベル 2 A クラス 3 (調歩同期式)
- 1 : レベル 2 B クラス 3 (S T 1 同期式)
(S Y N 同期式でモデムを使用しない場合)
- 2 : レベル 2 B クラス 3 (S T 2 同期式)
(S Y N 同期式でモデムを使用する場合)

(3) 通信速度

- | | |
|----------|----------|
| 0 : 150 | 4 : 2400 |
| 1 : 300 | 5 : 4800 |
| 2 : 600 | 6 : 9600 |
| 3 : 1200 | |

※ S T 2 同期式の場合は、モデムの通信速度によって決まります。

(4) SA

ステーションアドレス (入出力共通)

(5) 入力 UA 組

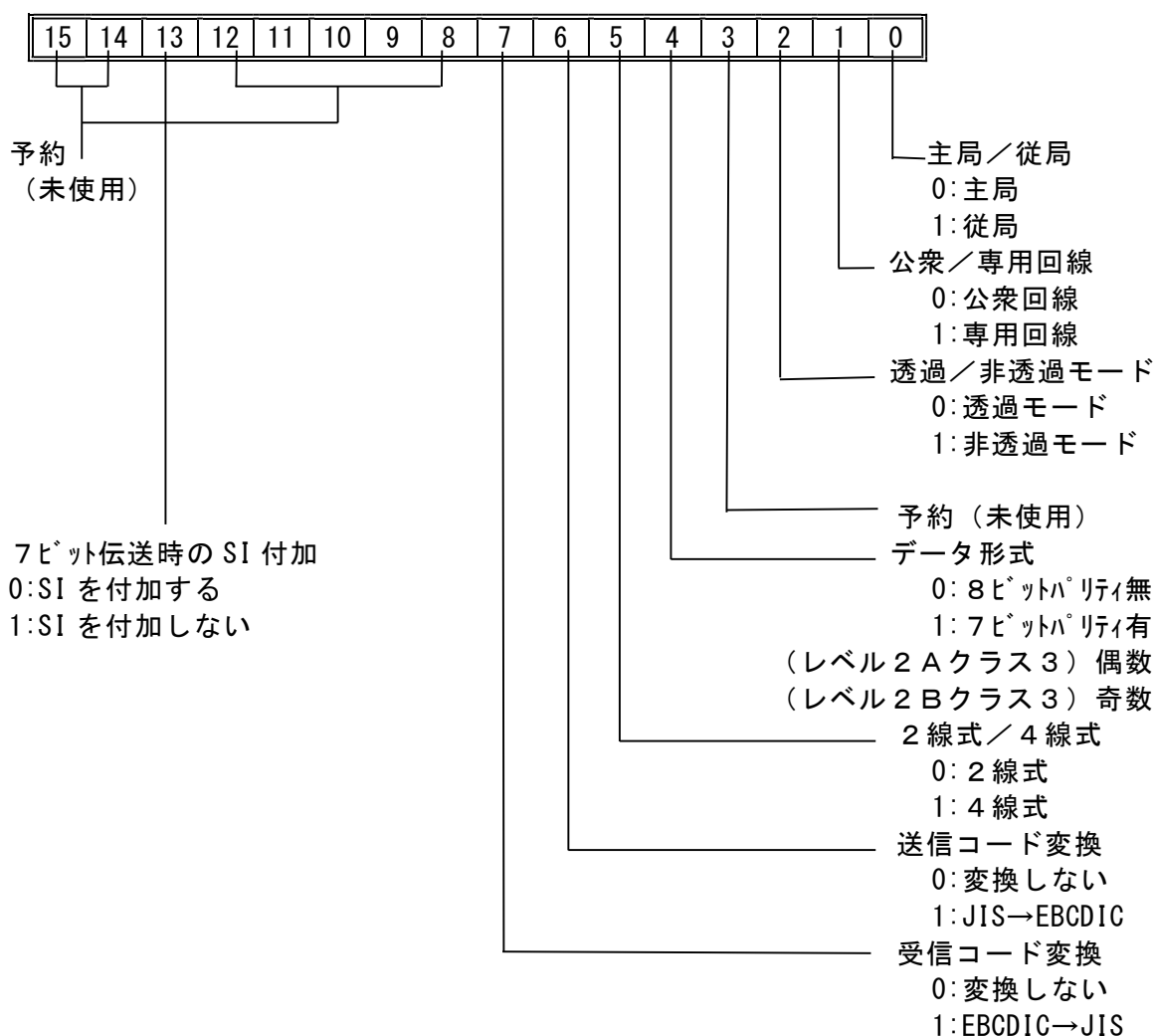
主局の場合、ポーリングする UA の組を指定してください。
従局の場合、セレクトイングされる UA の値を指定してください。

(6) 出力 UA 組

主局の場合、セレクトイングする UA の組を指定してください。
従局の場合、ポーリングされる UA の値を指定してください。

(7) 回線モード

回線モードは、2 バイトで下記のビットで指定してください。



(8) ステータス

0000H 正常終了

0000H以外は、エラーが発生しました。
「ステータス一覧表」を参照してください。

文 例

```

(C言語) int sts;           /*ステータス*/
        int mode;         /*回線モード*/
        char chn;        /*チャンネル番号*/
        int prot;        /*通信手順*/
        int bps;         /*通信速度*/
        int sa;          /*SA*/
        char iua[20];    /*入力 UA 組*/
        char oua[20];    /*出力 UA 組*/
        sa = 0x30;       /*SA*/
        iua[0] = 0x20;   /*入力 UA[0]*/
        iua[1] = 0;      /*入力 UA の指定終了*/
        oua[0] = 0x30;   /*出力 UA[0]*/
        oua[1] = 0;      /*出力 UA の指定終了*/
        chn = 0x00;      /*チャンネル 0 を指定*/
        prot = 0;        /*レベル 2A クラス 3 を指定*/
        bps = 4;         /*2400bps を指定*/
        mode = 0xe3;     /*回線モード の指定*/
                          /*コード を変換する(回線上 EBCDIC),
                          4 線式, テータ 8 ビット, 透過モード,
                          専用回線, 従局*/

```

```

sts = nec3open(chn, prot, bps, sa, iua, oua, mode);
if ( sts != 0x0000 ) {
    printf( "Open Error ch = %x St = %x¥n" , 0, chn, sts);
}

```

```

(BASIC) Dim IUA (0 TO 5) As Byte
        Dim OUA (0 TO 5) As Byte
        CHN% = 0           'チャンネル 0 を指定
        PROT% = 0         'レベル 2A クラス 3 を指定
        BPS% = 4          '2400bps を指定
        SA% = &H30        'SA
        IUA(0) = &H20     '入力 UA
        IUA(1) = 0        '指定の終了
        OUA(0) = &H30     '出力 UA
        OUA(1) = 0        '指定の終了
        MODE% = &HE3      'コード を変換する(回線上 EBCDIC),
                          4 線式, テータ 8 ビット, 透過モード,
                          専用回線, 従局*/

```

```

STS% = nec3open(CHN%, PROT%, BPS%, SA%, IUA(0), OUA(0), MODE%)
IF STS% <> 0 THEN
    PRINT " Open Error Ch =" ;CHN%; "St=" ;HEX$(STS%)
END IF

```

bscsend

機 能

送信データを送信バッファに入れます。
送信バッファに入ったデータは自動的に送信されます。

書 式

ステータス = bscsend (チャンネル番号, 送信データ, 送信データ長, 送信モード)

解 説

(1) チャンネル番号

H I - N E T のチャンネル番号 (0~7) を指定してください。

(2) 送信データ

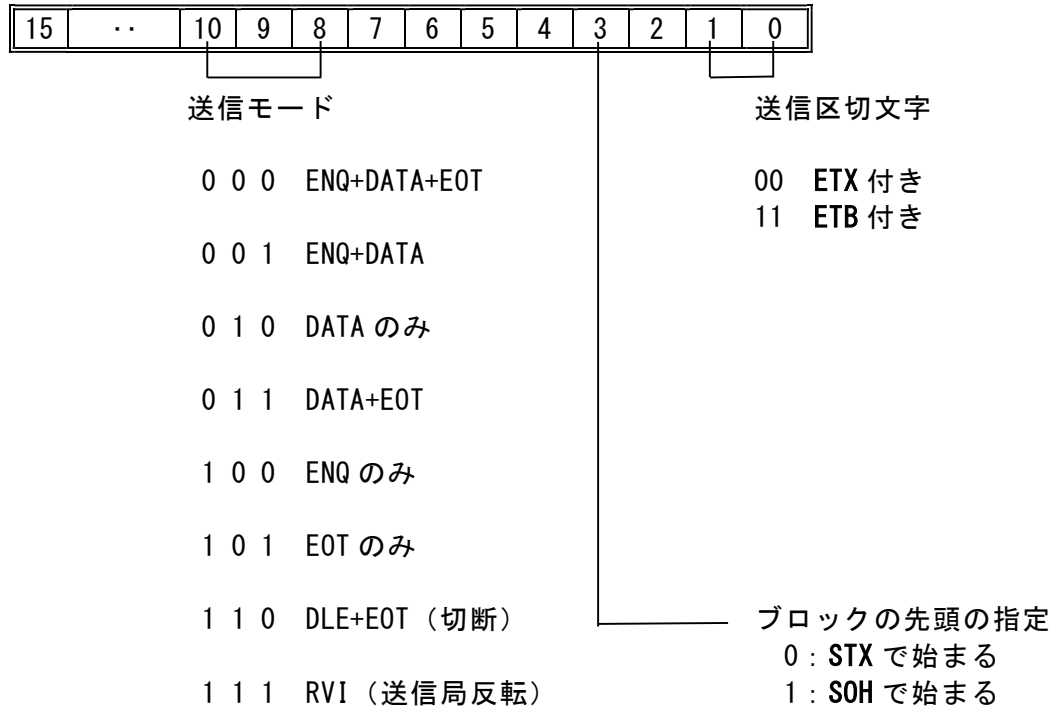
送信するデータを入れてください。
STX、ETB、ETX 等を含める必要はありません。

(3) 送信データ長

送信するデータの長さを指定してください。
ENQ、EOT のみの送信で送信データが存在しない場合は、0 を指定してください。

(4) 送信モード

送信モードおよび区切文字を指定してください。



(送信モードの指定例)

a) 1 電文を送信する場合の例

0 0 0 H E N Q + D A T A (E T X 付き) + E O T

1 つの送信関数で下記の一連の動作を実行します。

- ・ E N Q (セレクトィング) 送信
- ・ A C K 受信
- ・ D A T A (E T X 付き) 送信
- ・ A C K 受信
- ・ E O T 送信

b) 複数ブロック電文を送信する場合の例

4 0 0 H	E N Qのみ
2 0 3 H	D A T Aのみ (E T B付き)
⋮	
2 0 0 H	D A T Aのみ (E T X付き)
5 0 0 H	E O Tのみ

上記分の送信関数で下記の一連の動作を実行します。

- ・ E N Q (セレクトイング)送信
- ・ A C K受信
- ・ D A T A (E T B付き)送信
- ・ A C K受信
- ⋮
- ・ D A T A (E T X付き)送信
- ・ A C K受信
- ・ E O T送信

(5) ステータス

0 0 0 0 H 正常終了 (送信バッファに貯められた)

0 0 0 0 H以外は、エラーが発生しました。
「ステータス一覧表」を参照してください。

文 例

```
(C言語) int sts, ln, em;
char dt[2000];
char chn;
chn = 0x00;
strcpy(dt, "TEST DATA");
ln = strlen(dt);
em = 0;
sts = bscsend(chn, st, ln, em);
if ( sts != 0x0000 ) {
    printf ( "Send Error st = %x¥n" , sts);
}
```

```
(BASIC) ※注) 送信データは Unicode から ANSI に変換してください。
Dim Chn, Ln, STS%, I, Em As Integer
Dim Dt1, Dt2 As String
Dim bytData(1 to 1000) As Byte          ‘送信データ (Byte 配列)
Chn = 0
Dt1 = "TEST DATA"
Dt2 = StrConv(Dt1, vbFromUnicode)      ‘Unicode → ANSI
Ln = LenB(Dt2)
For I = 1 To Ln
    bytData(I) = AscB(MidB(Dt2, I, 1))  ‘Byte 配列に代入
Next I
Em = 0
STS% = bscsend(Chn, bytData(1), Ln, Em)
If STS% <> 0 Then
    Print "Send Error St =" ; Hex(STS%)
End If
```

nec2send

機 能

送信データを送信バッファに入れます。
送信バッファに入ったデータは自動的に送信されます。

書 式

ステータス = nec2send (チャンネル番号, 送信データ, 送信データ長, 出力 SA, 出力 UA,
送信モード)

解 説

(1) チャンネル番号

H I - N E T のチャンネル番号 (0~7) を指定してください。

(2) 送信データ

送信するデータを入れてください。
STX、ETB、ETX 等を含める必要はありません。

(3) 送信データ長

送信するデータの長さを指定してください。
ENQ、EOT のみの送信で送信データが存在しない場合は、0 を指定してください。

(4) 出力 SA

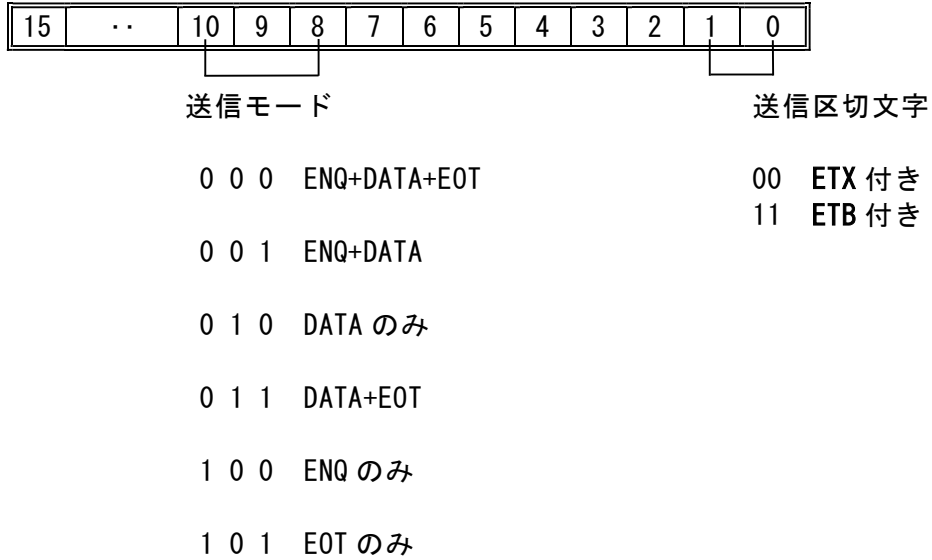
出力 SA を指定してください。(相手局の入力 SA)

(5) 出力 UA

出力 UA を指定してください。(相手局の入力 UA)

(6) 送信モード

送信モードを選択します。



(送信モードの指定例)

a) 1 電文を送信する場合の例

0 0 0 H E N Q + D A T A (E T X 付き) + E O T

1 つの送信関数で下記の一連の動作を実行します。

- ・ E N Q (セレクトイング)送信
- ・ A C K 受信
- ・ D A T A (E T X 付き)送信
- ・ A C K 受信
- ・ E O T 送信

b) 複数ブロック電文を送信する場合の例

400H	ENQのみ
203H	DATA(ETB付き)のみ
⋮	
200H	DATA(ETX付き)のみ
500H	EOTのみ

上記分の送信関数で下記の一連の動作を実行します。

- ・ ENQ(セレクトイング)送信
- ・ ACK受信
- ・ DATA(ETB付き)送信
- ・ ACK受信
- ⋮
- ・ DATA(ETX付き)送信
- ・ ACK受信
- ・ EOT送信

(7) ステータス

0000H 正常終了(送信バッファに貯められた)

0000H以外は、エラーが発生しました。
「ステータス一覧表」を参照してください。

文 例

```
(C言語) int sts;
        int ln;
        int em;
        char osa, oua;
        char dt[2000];
        char chn;
        chn = 0x00;
        osa = 0x30;
        oua = 0x20;
        strcpy(dt, " TEST DATA" );
        ln = strlen(dt);
        em = 0;
        sts = nec2send(chn, dt, ln, osa, oua, em);
        if ( sts != 0x0000 ) {
            printf( "Send Error st = %x¥n" , sts);
        }
```

```
(BASIC) ※注) 送信データは Unicode から ANSI に変換してください。
        Dim bytData(1 to 1000) as Byte          '送信データ (Byte 配列)
        CHN% = 0
        Dt1$ = " TEST DATA"
        Dt2$ = StrConv(Dt1$, vbFromUnicode)    ' Unicode → ANSI
        LN% = LenB(Dt2$)
        For I% = 1 To LN%
            bytData(I%) = AscB(MidB(Dt2$, I%, 1)) ' Byte 配列に代入
        Next I%
        EM% = 0
        OSA% = &H30
        OUA% = &H20
        STS% = nec2send(CHN%, bytData(1), LN%, OSA%, OUA%, EM%)
        IF STS% <> 0 THEN
            PRINT " Send Error St =" ; HEX$(STS%)
        END IF
```

nec3send

機 能

送信データを送信バッファに入れます。
送信バッファに入ったデータは自動的に送信されます。

書 式

ステータス = nec3send (チャンネル番号, 送信データ, 送信データ長, 出力 UA, 送信モード)

解 説

(1) チャンネル番号

H I - N E T のチャンネル番号 (0~7) を指定してください。

(2) 送信データ

送信するデータを入れてください。
STX、ETB、ETX 等を含める必要はありません。

(3) 送信データ長

送信するデータの長さを指定してください。
ENQ、EOT のみの送信で送信データが存在しない場合は、0 を指定してください。

(4) 出力 UA

出力 UA を入れてください。

(5) 送信モード

送信モードを選択します。

15	..	10	9	8	7	6	5	4	3	2	1	0
----	----	----	---	---	---	---	---	---	---	---	---	---

送信モード

0 0 0	ENQ+DATA+EOT	00	ETX 付き
0 0 1	ENQ+DATA	11	ETB 付き
0 1 0	DATAのみ		
0 1 1	DATA+EOT		
1 0 0	ENQのみ		
1 0 1	EOTのみ		

送信区切文字

(送信モードの指定例)

a) 1 電文を送信する場合

0 0 0 H ENQ+DATA (ETX 付き)+EOT

1 つの送信関数で下記の一連の動作を実行します。

- ・ ENQ (セレクトイング) 送信
- ・ ACK 受信
- ・ DATA (ETX 付き) 送信
- ・ ACK 受信
- ・ EOT 送信

b) 複数ブロック電文を送信する場合の例

400H	ENQのみ
203H	DATA(ETB付き)のみ
⋮	
200H	DATA(ETX付き)のみ
500H	EOTのみ

上記分の送信関数で下記の一連の動作を実行します。

- ・ ENQ(セレクトイング)送信
- ・ ACK受信
- ・ DATA(ETB付き)送信
- ・ ACK受信
- ⋮
- ・ DATA(ETX付き)送信
- ・ ACK受信
- ・ EOT送信

(6) ステータス

0000H 正常終了(送信バッファに貯められた)

0000H以外は、エラーが発生しました。
「ステータス一覧表」を参照してください。

文 例

```
(C言語) int sts;
        int ln;
        int em;
        char oua;
        char dt[2000];
        char chn;
        chn = 0x00;
        oua = 0x20;
        strcpy(dt, "TEST DATA");
        ln = strlen(dt);
        em = 0;
        sts = nec3send(chn, dt, ln, oua, em);
        if ( sts != 0x0000 ) {
            printf ( "Send Error st = %x¥n" , sts);
        }
```

```
(BASIC) ※注) 送信データは Unicode から ANSI に変換してください。
        Dim bytData(1 to 1000) as Byte          '送信データ (Byte 配列)
        CHN% = 0
        Dt1$ = "TEST DATA"
        Dt2$ = StrConv(Dt1$, vbFromUnicode)    ' Unicode → ANSI
        LN% = LenB(Dt2$)
        For I% = 1 To LN%
            bytData(I%) = AscB(MidB(Dt2$, I%, 1)) ' Byte 配列に代入
        Next I%
        EM% = 0
        OUA% = &H20
        STS% = nec3send(CHN%, bytData(1), LN%, OUA%, EM%)
        IF STS% <> 0 THEN
            PRINT " Send Error St =" ; HEX$(STS%)
        END IF
```

sendck

機能

送信終了のチェックを行います。

書式

ステータス = sendck (チャンネル番号)

解説

(1) チャンネル番号

H I - N E T のチャンネル番号 (0~7) を指定してください。

(2) ステータス

0 0 0 0 H	正常終了
0 0 0 1 H	送信バッファ内に送信データが残っています。

上記以外は、エラーが発生しました。
「ステータス一覧表」を参照してください。

文例

```
(C言語) int sts;
char chn;
chn = 0x00;
do {
    sts = sendck(chn);
} while ( sts == 0x0001 )

if ( sts != 0x0000 ) {
    printf ( "Send Error st = %x¥n" , sts);
}
```

```
(BASIC) CHN% = 0
DO
    STS% = sendck(CHN%)
LOOP WHILE STS% = 1

IF STS% <> 0 THEN
    PRINT " Send Error St =" ; HEX$(STS%)
END IF
```

hrecv

機能

受信バッファから受信データを取り出します。

書式

ステータス = hrecv (チャンネル番号, 受信データ)

解説

(1) チャンネル番号

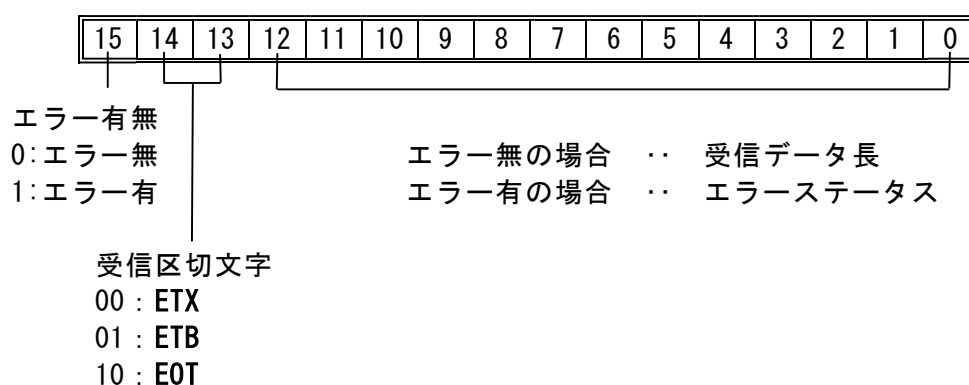
H I - N E T のチャンネル番号 (0~7) を指定してください。

(2) 受信データ

受信したデータが入ります。
EOT を受信した場合は、受信データには何も入りません。

(3) ステータス

0000H	受信データなし
8XXXH	エラー発生
	「ステータス一覧表」を参照してください。
上記以外	受信データあり



注意事項

通信手順がレベル 2 A、レベル 2 B の場合は受信データの先頭に SA、UA の計 2 バイトが含まれています。

文 例

```
(C言語) int sts, ln, em;
char dt[5000];
char chn;
chn = 0x00;
do {
    sts = hrecv(chn, dt);
} while ( sts == 0x0000 )

if (( sts & 0x8000 ) == 0x8000 ) {
    printf ( "Receive Error st = %x¥n" , sts);
} else{
    em = ( sts / 0x2000 ) & 3;          /*受信区切文字*/
    ln = sts & 0x1fff;                /*受信データ長*/
    if ( em != 2 ) {                  /*EOT 以外*/
        printf ( "Receive Data = %x¥n" , dt );
    }
}
}
```

```
(BASIC) ※注) 受信データは ANSI から Unicode に変換してください。
Dim bytData(1 to 1000) as Byte          '受信データ (Byte 配列)
CHN% = 0
DO
    STS% = hrecv(CHN%, bytData)        ' Byte 型で受信
LOOP WHILE STS% = 0

IF ( STS% AND &H8000 ) = &H8000 THEN
    PRINT " Receive Error St =" ; HEX$(STS%)
ELSE
    EM% = ( STS% /&H2000 ) AND 3        '受信区切文字
    LN% = STS% AND &H1FFF              '受信データ長
    IF EM% <> 2 THEN                    'EOT 以外
        Dt1$ = ""
        For I% = 1 To LN%
            Dt1$ = Dt1$ + ChrB(bytData(I%))
        NEXT I%
        Dt2$ = StrConv(Dt1$, vbUnicode) ' ANSI → Unicode
        PRINT "Receive Data=" ; Dt2$
    END IF
END IF
```

hrecvx

機能

受信バッファから受信データを取り出します。
(受信データ長が0の受信データも取出します。)

書式

ステータス = hrecvx (チャンネル番号, 受信データ)

解説

(1) チャンネル番号

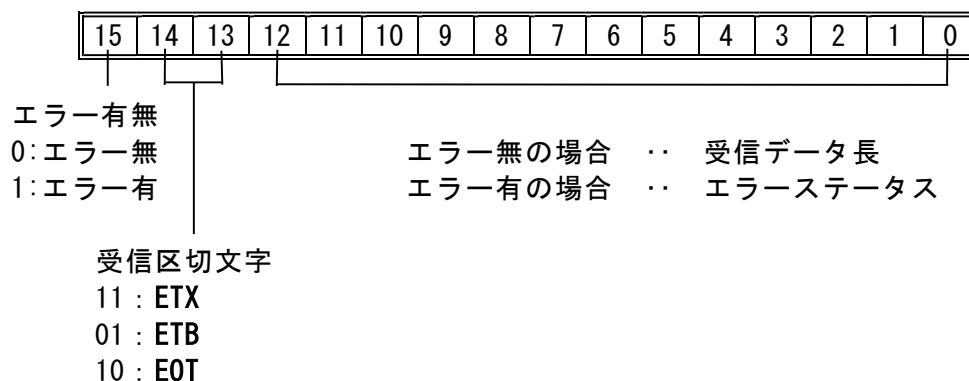
H I - N E T のチャンネル番号 (0~7) を指定してください。

(2) 受信データ

受信したデータが入ります。
EOT を受信した場合の受信データ長は 0 です。

(3) ステータス

0000H	受信データなし
8XXXXH	エラー発生 「ステータス一覧表」を参照してください。
上記以外	受信データあり



注意事項

通信手順がレベル 2 A、レベル 2 B の場合は受信データの先頭に SA、UA の計 2 バイトが含まれています。

文 例

```
(C言語) int sts, ln, em;
char dt[5000];
char chn;
chn = 0x00;
do {
    sts = hrecvx(chn, dt);
} while ( sts == 0x0000 )

if (( sts & 0x8000 ) == 0x8000 ) {
    printf ( "Receive Error st = %x¥n" , sts);
} else{
    em = ( sts / 0x2000 ) & 3;          /*受信区切文字*/
    ln = sts & 0x1fff;                 /*受信データ長*/
    if ( em != 2 ) {                   /*EOT 以外*/
        printf ( "Receive Data = %x¥n" , dt );
    }
}
}
```

```
(BASIC) ※注) 受信データは ANSI から Unicode に変換してください。
Dim bytData(1 to 1000) as Byte          '受信データ(Byte 配列)
CHN% = 0
DO
    STS% = hrecvx(CHN%, bytData)      ' Byte 型で受信
LOOP WHILE STS% = 0

IF ( STS% AND &H8000 ) = &H8000 THEN
    PRINT " Receive Error St =" ; HEX$(STS%)
ELSE
    EM% = ( STS% /&H2000 ) AND 3        '受信区切文字
    LN% = STS% AND &H1FFF               '受信データ長
    IF EM% <> 2 THEN                     'EOT 以外
        Dt1$ = ""
        For I% = 1 To LN%
            Dt1$ = Dt1$ + ChrB(bytData(I%))
        NEXT I%
        Dt2$ = StrConv(Dt1$, vbUnicode) ' ANSI → Unicode
        PRINT "Receive Data=" ; Dt2$
    END IF
END IF
```

rdtck

機能

受信データの件数を確認します。

書式

ステータス = rdtck (チャンネル番号)

解説

(1) チャンネル番号

H I - N E T のチャンネル番号 (0~7) を指定してください。

(2) ステータス

受信したデータ件数を返します。

値が 0 の場合、受信データはありません。

受信データの件数には、EOT や受信エラーも 1 件として数えられます。

0 0 0 0 H 受信データなし

0 X X X H 受信データ件数

文例

```
(C言語) int sts;
         char chn;
         chn = 0x00;
         sts = rdtck(chn);
```

```
(BASIC) CHN% = 0
         STS% = rdtck(CHN%)
```


rbfclr

機能

受信バッファをクリアします。

書式

ステータス = rbfclr (チャンネル番号)

解説

(1) チャンネル番号

H I - N E T のチャンネル番号 (0~7) を指定してください。

(2) ステータス

0000H	正常終了
8XXXXH	異常終了

注意事項

受信関数 (**hrecv**) で受信データを取り出すと、受信バッファからは除かれています。**rbfclr** は意図的に受信バッファをクリアする場合に使用してください。

文例

```
(C言語) int sts;
char chn;
chn = 0x00;
sts = rbfclr(chn);

if ((sts & 0x8000) == 0x8000) {
    printf("バッファクリア Error St = %x\n", sts);
}
```

```
(BASIC) CHN% = 0
STS% = rbfclr(CHN%)

IF (STS% AND &H8000) = &H8000 THEN
    PRINT "バッファクリア Error St=" ; HEX$(STS%)
END IF
```

sbfclr

機能

送信バッファをクリアします。

書式

ステータス = sbfclr (チャンネル番号)

解説

(1) チャンネル番号

H I - N E T のチャンネル番号 (0~7) を指定してください。

(2) ステータス

0000H	正常終了
8XXXXH	異常終了

注意事項

送信が終了すると送信データは送信バッファから除かれていきます。
sbfclr は意図的に送信バッファをクリアする場合に使用してください。

文例

```
(C言語) int sts;
char chn;
chn = 0x00;
sts = sbfclr(chn);

if ((sts & 0x8000) == 0x8000) {
    printf("バッファクリア Error St = %x\n", sts);
}
```

```
(BASIC) CHN% = 0
STS% = sbfclr(CHN%)

IF (STS% AND &H8000) = &H8000 THEN
    PRINT "バッファクリア Error St=" ; HEX$(STS%)
END IF
```

chclose

機能

回線のクローズを行います。

書式

ステータス = chclose (チャンネル番号)

解説

(1) チャンネル番号

H I - N E T のチャンネル番号 (0~7) を指定してください。

(2) ステータス

0 0 0 0 H 正常終了

0 0 0 0 H 以外ならエラーが発生しました。
「ステータス一覧表」を参照してください。

文例

```
(C言語) int sts;
char chn;
chn = 0x00;                               /*チャンネル0を指定*/
sts = chclose(chn);

if (sts != 0x0000) {
    printf("Close Error St = %x¥n", sts);
}
```

```
(BASIC) CHN% = 0                               'チャンネル0を指定
STS% = chclose(CHN%)

IF STS% <> 0 THEN
    PRINT "Close Error St=" ; HEX$(STS%)
END IF
```

telon

機能

指定した電話番号に自動ダイヤルします。

書式

ステータス = telon (チャンネル番号, 電話番号, 接続タイムアウト値)

解説

(1) チャンネル番号

H I - N E T のチャンネル番号 (0~7) を指定してください。

(2) 電話番号

自動ダイヤルする電話番号を文字列で指定してください。
電話番号に許される文字は使用するモデムの説明書を参考にしてください。

(3) 接続タイムアウト値

電話接続のタイムアウト値を 100msec 単位で指定してください。
0 を指定した時は、既定値の 60 秒が使用されます。
指定した時間内に相手局とつながらなければエラーとなります。

(4) ステータス

0 0 0 0 H 正常終了

0 0 0 0 H 以外ならエラーが発生しました。
「ステータス一覧表」を参照してください。

注意事項

自動ダイヤル機能は、V25.bis に対応したモデムでしか使用できません。

文 例

```
(C言語) int sts, tm;
char telno[20];
char chn;
chn = 0x00;
strcpy(telno, "XX-XXXX-XXXX"); /*電話番号*/
tm = 300; /*接続タイムアウトを 30 秒*/
sts = telon(chn, telno, tm);

if (sts != 0x0000) {
    printf ("Telon Error St = %x¥n", sts);
}
```

```
(BASIC) CHN% = 0
TELNO$ = "XX-XXXX-XXXX" '電話番号
TM% = 300 '接続タイムアウトを 30 秒
STS% = telon(CHN%, TELNO$, TM%)
IF STS% <> 0 THEN
    PRINT "Telon Error St =" ; HEX$(STS%)
END IF
```

mtelon

機能

自局が発信局の場合は、手動で電話接続が完了したかのチェックを行い、自局が着信局の場合は、自動着信を行います。

書式

ステータス = mtelon (チャンネル番号)

解説

(1) チャンネル番号

H I - N E T のチャンネル番号(0~7)を指定してください。

(2) ステータス

0000H	正常終了
8039H	電話接続が完了していませんので、 mtelon をリトライしてください。

上記以外ならエラーが発生しました。
「ステータス一覧表」を参照してください。

文例

```
(C言語) int sts;
char chn;
chn = 0x00;
do {
    sts = mtelon(chn);
} while (sts == 0x8039)
if (sts != 0x0000) {
    printf("Mtelon Error St = %x¥n", sts);
}
```

```
(BASIC) CHN% = 0
DO
    STS% = mtelon(CHN%)
LOOP WHILE STS% = &H8039
IF STS% <> 0 THEN
    PRINT "Mtelon Error St=" ; HEX$(STS%)
END IF
```

teloff

機能

接続されている電話回線を切断します。

書式

ステータス = teloff (チャンネル番号)

解説

(1) チャンネル番号

H I - N E T のチャンネル番号(0~7)を指定してください。

(2) ステータス

0 0 0 0 H 正常終了

0 0 0 0 H 以外ならエラーが発生しました。
「ステータス一覧表」を参照してください。

※切断時には、切断信号(DLE・EOT)が相手局に送られます。

文例

```
(C言語) int sts;
         char chn;
         chn = 0x00;
         sts = teloff(chn);

         if (sts != 0x0000) {
             printf("Teloff Error St = %x¥n", sts);
         }
```

```
(BASIC) CHN% = 0
         STS% = teloff(CHN%)
         IF STS% <> 0 THEN
             PRINT "Teloff Error St=" ; HEX$(STS%)
         END IF
```

verget

機能

H I - N E T のバージョンを取得します。

書式

ステータス = verget (チャンネル番号, バージョン取得文字列)

解説

(1) チャンネル番号

H I - N E T のチャンネル番号 (0~7) を指定してください。

(2) バージョン取得文字列

バージョン取得文字列として 20 バイトのエリアを確保してください。
バージョン取得文字列に「YYMMDD (X.XX)」の形式でバージョンが返ってきます。

YYMMDD は、「年月日」を表します。

X.XX は、「バージョン」を表します。

文例

```
(C言語) int sts, tm;
        char chn;
        char ver[20];
        sts = verget(ver);
        if (sts != 0x0000) {
            printf ("Version = %x¥n", ver);
        }
```

```
(BASIC) CHN% = 0
        VER$ = SPACE$(20)
        STS% = verget(VER$)
        IF STS% <> 0 THEN
            PRINT "Version =" ; VER$
        END IF
```


timerset

機能

各タイマー値を既定値から変更します。

書式

timerset (チャンネル番号, タイマー1, タイマー2, タイマー3, タイマー4, タイマー5)

解説

(1) チャンネル番号

H I - N E T のチャンネル番号 (0~7) を指定してください。

(2) タイマー1 (無通信監視タイマー)

公衆回線モードにおいて、送受信が共に行われない監視時間を指定します。

タイムアウト時は、切断 (DLE・EOT) を送信して回線を切断します。

100msec 単位で指定してください。

既定値は 20 秒 (200) となっています。

※タイマーの起動条件

- ・ EOT 送信時
- ・ EOT 受信時
- ・ 電話接続 (telon) にて相手と接続された場合
- ・ 受信時にタイマー3 (受信監視タイマー) がタイムアウトした時
- ・ 受信時に異常終了した場合

※タイマーのリセット条件

- ・ セレクティングシーケンスを送信した場合
- ・ セレクティングシーケンスを受信した場合
- ・ 切断 (DLE・EOT) を受信した場合

(3) タイマー 2 (送信応答監視タイマー)

セレクトイングまたは電文を送信した後、相手局からの応答を待つ時間を指定します。
タイムアウト時にセレクトイングまたは督促 ENQ を送出します。
100msec 単位で指定してください。
既定値は 2 秒 (20) となっています。

※タイマーの起動条件

- ・セレクトイングシーケンス送信時
- ・電文送信時
- ・ENQ 送信時

※タイマーのリセット条件

- ・有効な応答を受信した時
- ・タイムアウト時

(4) タイマー 3 (受信監視タイマー)

セレクトイングに対して ACK を返した後、または送信データに対して ACK を返した後、次の電文または EOT を受信するまでの監視時間を指定します。
100msec 単位で指定してください。
既定値は 3 秒 (30) となっています。

※タイマーの起動条件

- ・受信応答を送信した場合
(セレクトイングに対する ACK を返した場合、電文ブロックを受信して ACK を返した場合)

※タイマーのリセット条件

- ・次の電文ブロックを受信した場合
- ・EOT を受信した場合
- ・タイムアウト時

(5) タイマー 4 (WACK 応答 ENQ 送出タイマー)

WACK を受信した場合、次に ENQ を送出するまでの時間を指定します。
100msec 単位で指定してください。
既定値は 1 秒 (10) となっています。
BSC手順のみ有効です。

(6) タイマー 5 (主局のポーリング間隔)

100msec 単位で指定してください。
既定値は 1 秒 (10) となっています。
レベル 2 A / 2 B クラス 3 の主局のみ有効です。

※既定値のまま通信を行う場合は、この関数を呼ぶ必要がありません。

※各タイマー値に 0 を指定した場合は既定値が使用されます。

注意事項

回線オープンの前に実行してください。

文例

```
(C言語) char chn;
int tm1, tm2, tm3, tm4, tm5;
chn = 0x00;
tm1 = 300; /* 無通信監視タイマー 30秒*/
tm2 = 30; /* 送信応答監視タイマー 3秒*/
tm3 = 40; /* 受信監視タイマー 4秒*/
tm4 = 20; /* WACK 応答 ENQ 送出タイマー 2秒*/
tm5 = 0; /* 主局のポーリング間隔 既定値*/
timerset(chn, tm1, tm2, tm3, tm4, tm5);
```

```
(BASIC) CHN% = 0
TM1% = 300 '無通信監視タイマー 30秒
TM2% = 30 '送信応答監視タイマー 3秒
TM3% = 40 '受信監視タイマー 4秒
TM4% = 20 'WACK 応答 ENQ 送出タイマー 2秒
TM5% = 0 '主局のポーリング間隔 既定値
CALL timerset(CHN%, TM1%, TM2%, TM3%, TM4%, TM5%)
```

retryset

機 能

各リトライ値を既定値から変更します。

書 式

retryset (チャンネル番号, リトライ 1, リトライ 2, リトライ 3, リトライ 4, リトライ 5)

解 説

(1) チャンネル番号

H I - N E T のチャンネル番号 (0~7) を指定してください。

(2) リトライ 1

セレクトイング送出後、相手局が無応答の場合にセレクトイングを再送する回数を指定します。
既定値は 7 回となっています。

(3) リトライ 2

セレクトイング送出後、相手局から NAK が返ってきた場合にセレクトイングを再送する回数を指定します。
既定値は 7 回となっています。

(4) リトライ 3

データ送信後、相手局が無応答の場合に督促 ENQ を送出する回数を指定します。
既定値は 7 回となっています。

(5) リトライ 4

データ送信後、相手局から NAK が返ってきた場合にデータを再送信する回数を指定します。
既定値は 7 回となっています。

(6) リトライ 5

データ送信後、相手局から WACK が返ってきた場合に ENQ を送出する回数を指定します。
既定値は 15 回となっています。

※既定値のまま通信を行う場合は、この関数を呼ぶ必要はありません。

※各リトライ値を 0 にした場合は、リトライは行われません。

※リトライ値の最大は **65535** です。
(大きな数値に変更すると、エラーの発生に時間が長くなってしまいます。)

注意事項

回線オープンの前に実行してください。

文 例

```
(C言語) char chn;
int rt1, rt2, rt3, rt4, rt5;
chn = 0x00;
rt1 = 5; /* リトライ値 1 を 5 にする */
rt2 = 3; /* リトライ値 2 を 3 にする */
rt3 = 5; /* リトライ値 3 を 5 にする */
rt4 = 3; /* リトライ値 4 を 3 にする */
rt5 = 10; /* リトライ値 5 を 10 にする */
retryset(chn, rt1, rt2, rt3, rt4, rt5);
```

```
(BASIC) CHN% = 0
RT1% = 5 'リトライ値 1 を 5 にする
RT2% = 3 'リトライ値 2 を 3 にする
RT3% = 5 'リトライ値 3 を 5 にする
RT4% = 3 'リトライ値 4 を 3 にする
RT5% = 10 'リトライ値 5 を 10 にする
CALL retryset(CHN%, RT1%, RT2%, RT3%, RT4%, RT5%)
```

padset

機能

リーディングパッド、トレーリングパッドを既定値から変更します。

書式

ステータス = padset (チャンネル番号, リーディングパッド, リーディングパッド数,
トレーリングパッド, トレーリングパッド数)

解説

(1) チャンネル番号

H I - N E T のチャンネル番号 (0~7) を指定してください。

(2) リーディングパッド

リーディングパッドキャラクタを指定してください。
初期値は AAH となっています。

(3) リーディングパッド数

リーディングパッドキャラクタの個数を指定してください。(最大 4 個)
初期値は 2 個となっています。

(4) トレーリングパッド

トレーリングパッドキャラクタを指定してください。
初期値は FFH となっています。

(5) トレーリングパッド数

トレーリングパッドキャラクタの個数を指定してください。(最大 4 個)
初期値は 2 個となっています。

注意事項

回線オープンの前に実行してください。

文 例

```
(C言語) int sts;
char chn;
char lpad;
int lpadct;
char tpad;
int tpadct;
chn = 0;
lpad = 0x55;          /*リーディングパッドを 55H に変更*/
lpadct = 3;          /*リーディングパッドの数を 3 に変更*/
tpad = 0xff;         /*トレーリングパッドを FFH に変更*/
tpadct = 3;         /*トレーリングパッドの数を 3 に変更*/
sts = padset(chn, lpad, lpadct, tpad, tpadct);
```

```
(BASIC) CHN% = 0
LPAD% = &H55          'リーディングパッドを 55H に変更
LPADCT% = 3          'リーディングパッドの数を 3 に変更
TPAD% = &HFF         'トレーリングパッドを FFH に変更
TPADCT% = 3         'トレーリングパッドの数を 3 に変更
STS% = padset(CHN%, LPAD%, LPADCT%, TPAD%, TPADCT%)
```

bsizeset

機能

送信、受信データバッファの1ブロックのサイズを変更します。

書式

ステータス = bsizeset (チャンネル番号, バッファサイズ)

解説

(1) チャンネル番号

H I - N E T のチャンネル番号 (0~7) を指定してください。

(2) バッファサイズ

1 ブロックのサイズを指定してください。

初期値は 2048 バイトとなっています。

(最小 100 バイト・最大 4096 バイト)

注意事項

回線オープンの前に実行してください。

実際のデータブロック長に 10 バイトを加えた値を設定してください。

文例

```
(C言語) int sts;
char chn;
int buflen;
chn = 0;
buflen = 4096; /*データ1ブロックのサイズを4096バイトに変更*/
sts = bsizeset(chn, buflen);
```

```
(BASIC) CHN% = 0
BUFLN% = 4096 'データ1ブロックのサイズを4096バイトに変更
STS% = bsizeset(CHN%, BUFLN%)
```


syncset

機能

SYNCキャラクタとその数を変更します。

書式

ステータス = syncset (チャンネル番号, SYNCキャラクタ, SYNC数)

解説

(1) チャンネル番号

HI-NETのチャンネル番号(0~7)を指定してください。

(2) SYNCキャラクタ

SYNCキャラクタコードを指定してください。

初期値は B S C 32H
 レベル2 B 16H となっています。

(3) SYNC数

SYNCキャラクタの出力個数を指定してください。(最大4個)

初期値は4個です。

値を0にすると、4個を出力します。

注意事項

回線オープンの前に実行してください。

SYN同期式通信の場合のみ有効です。

文例

```
(C言語)  int sts;
          char chn;
          int sync, synclen;
          chn = 0;
          sync = 0x32;            /* SYNC キャラクタを 32H に指定 */
          synclen = 3;           /* SYNC キャラクタを 3 個出力に変更 */
          sts = syncset(chn, sync, synclen);
```

```
(BASIC)  CHN% = 0
          SYNC% = &H32           'SYNC キャラクタを 32H に指定
          SYNCLEN% = 3           'SYNC キャラクタを 3 個出力に変更
          STS% = SYNCSET%(CHN%, SYNC%, SYNCLEN%)
```

ridget

機能

B S C手順で受信した ID を取得します。

書式

ステータス = ridget (チャンネル番号, ID データ)

解説

(1) チャンネル番号

H I - N E T のチャンネル番号 (0~7) を指定してください。

(2) ID データ

受信した ID が格納されます。

(3) ステータス

0 0 0 0 H	ID データなし
0 X X X H	ID データ長

注意事項

受信後にこの関数を呼んでください。
bscopen で相手局 ID を NULL に指定した場合は、全ての ID の受信が可能です。

文 例

```
(C言語) int sts;
char iddt[100];
char chn;
chn = 0x00;
sts = ridget(chn, iddt);
if (( sts!= 0x0000) {
    printf( "Receive ID = %x¥n" , iddt);
}
```

```
(BASIC) CHN%= 0
IDDT$ = SPACE$(500)
STS% = ridget(CHN%, IDDT$)
IF ( STS% <> 0000 ) THEN
    PRINT "Receive ID =" ; IDDT$
END IF
```

pollon

機能

指定したユニットアドレスへのポーリングを開始します。
(ポーリング再開)

書式

ステータス = pollon (チャンネル番号, 入力ユニットアドレス)

解説

(1) チャンネル番号

H I - N E T のチャンネル番号 (0~7) を指定してください。

(2) 入力ユニットアドレス

ポーリングを開始する入力ユニットアドレスを指定してください。

(3) ステータス

0 0 0 0 H	正常
0 0 0 3 H	指定した入力ユニットアドレスが回線オープン時に指定されていない。

注意事項

レベル 2 A / 2 B クラス 3 の主局側のみで有効です。

文例

```
(C言語) int sts;
         char chn;
         char iua;
         chn = 0;
         iua = 0x21;
         sts = pollon(chn, iua);
```

```
(BASIC) CHN% = 0
         IUA% = &H21
         STS% = pollon(CHN%, IUA%)
```

polloff

機能

指定したユニットアドレスへのポーリングを停止します。

書式

ステータス = polloff (チャンネル番号, 入力ユニットアドレス)

解説

(1) チャンネル番号

H I - N E T のチャンネル番号 (0~7) を指定してください。

(2) 入力ユニットアドレス

ポーリングを停止する入力ユニットアドレスを指定してください。

(3) ステータス

0 0 0 0 H	正常
0 0 0 3 H	指定した入力ユニットアドレスが回線オープン時に指定されていない。

注意事項

レベル 2 A / 2 B クラス 3 の主局側のみで有効です。

文例

```
(C言語) int sts;
         char chn;
         char iua;
         chn = 0;
         iua = 0x21;
         sts = polloff(chn, iua);
```

```
(BASIC) CHN% = 0
         IUA% = &H21
         STS% = polloff(CHN%, IUA%)
```

pollstat

機能

指定したユニットアドレスのポーリング状態を取得します。

書式

ステータス = pollstat (チャンネル番号, 入力ユニットアドレス)

解説

(1) チャンネル番号

H I - N E T のチャンネル番号 (0~7) を指定してください。

(2) 入力ユニットアドレス

ポーリング状態を取得する入力ユニットアドレスを指定してください。

(3) ステータス

0 0 0 0 H	ポーリング応答なし (またはエラー状態)
0 0 0 1 H	ポーリング応答あり
0 0 0 2 H	ポーリングを開始していない
0 0 0 3 H	指定した入力ユニットアドレスが回線オープン時に指定されていない。

注意事項

レベル 2 A / 2 B クラス 3 の主局側のみで有効です。

文例

```
(C言語) int sts;
char chn;
char iua;
chn = 0;
iua = 0x21;
sts = pollstat(chn, iua);
```

```
(BASIC) CHN% = 0
IUA% = &H21
STS% = pollstat(CHN%, IUA%)
```

第4章 ステータス一覧表

ステータス	意味	対象機能	原因	対処方法
8001H	送信バッファにデータが残っている	bcsend nec2send nec3send	1. 送信バッファに未送信のデータが残っている	1. send 関数で送信バッファに格納した直後に返る可能性があります がエラーではありません
8002H	送信バッファオーバーフロー	bcsend nec2send nec3send sendck	1. 送信バッファがオーバーフローした	1. 自局のアプリケーションを調べる
8003H	リンク・オフ	telon	1. 電話接続タイマー時間を経過したが、電話が接続されなかった	1. 相手局が立ち上がっていない
		bcsend nec2send nec3send sendck hrecv	(4線式の場合) 1. 回線オープン以降にDR信号が落ちた (2線式の場合) 2. 電話接続が完了していない	1. 自局のモデムまたはケーブルを調べる
8004H	電話がつながっているにもかかわらず、再度電話をかけようとした	telon	1. 自局アプリケーションの誤り	1. 自局のアプリケーションを調べる
8005H	オープンされているにもかかわらず、再度オープンしようとした	bscopen nec2open nec3open	1. 自局アプリケーションの誤り	1. 自局のアプリケーションを調べる
8006H	オープンされていない	telon mtelon bcsend nec2send nec3send sendck hrecv chclose	1. オープンをしないで他の関数を呼んだ (自局アプリケーションの誤り)	1. 自局のアプリケーションを調べる
8007H	イニシャライズ(initial)関数で初期化を行っていないチャンネルに対してオープンを行った	bscopen nec2open nec3open	1. 自局アプリケーションの誤り	1. 自局のアプリケーションを調べる
8010H	送信処理でセレクトィング中、RVI(送信局の反転)を受信した	sendck	1. 相手局が割り込みで送信しようとしている	1. 自局から送信しようとしたが、相手局が送信しようとしているので受信を行う 2. アプリケーション間の同期をとる

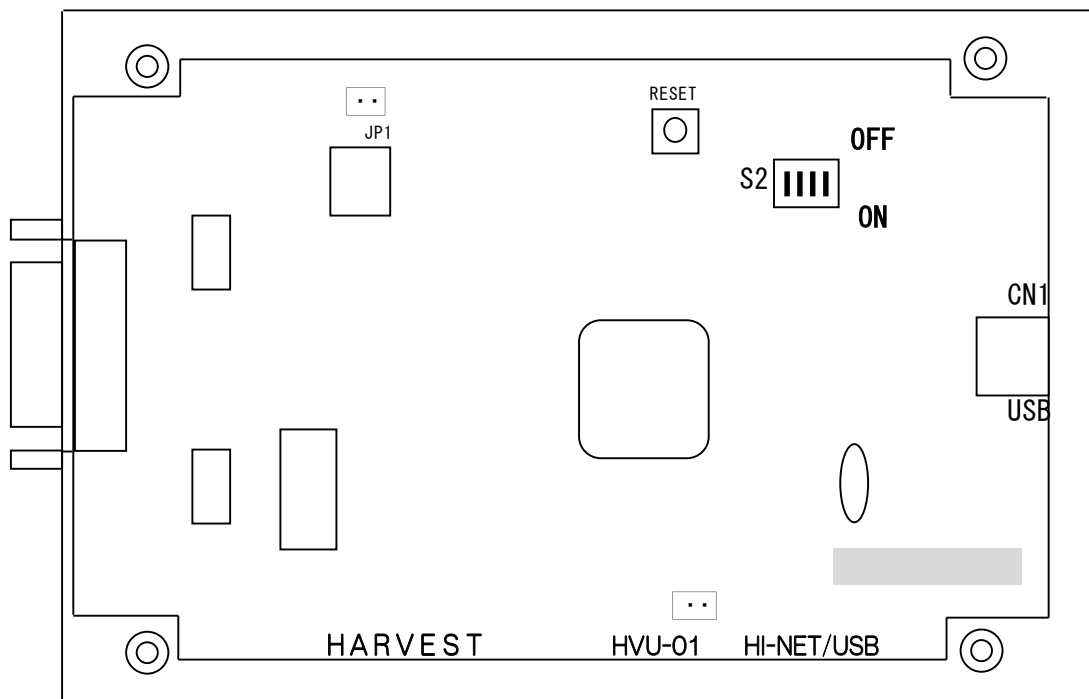
ステータス	意味	対象機能	原因	対処方法
8011H	送信処理でセレクトイング中、回線切断(DLE・EOT)を受信した	sendck	1. 相手局が何らかの原因で回線切断(DLE・EOT)を送出している (タイムアウト等)	1. 相手局を調べる ・ 自局側からのデータの送出手法が遅い
8012H	送信処理でセレクトイング中、無応答リトライを規定回数行ったが応答がなかった	sendck	1. 相手局が立ち上がっていない 2. 通信速度が相手局と異なる 3. 出力 SA・UA が相手局の入力 SA・UA と異なっている (レベル 2 A / 2 B の場合) 4. ID が合っていない (BSC 手順の場合)	1. 相手局を調べる 2. 自局と相手局の通信速度の設定を調べる 3. 自局と相手局の SA、UA の設定を調べる 4. 自局と相手局の ID の設定を調べる
8013H	送信処理でセレクトイング中、無効応答(NAK)を規定回数受け取った	sendck	1. 透過／非透過の設定が相手局と異なる	1. 自局と相手局の透過／非透過の設定を調べる
8014H	送信処理で電文送信中、回線切断(DLE・EOT)を受信した	sendck	1. 相手局が何らかの原因で回線切断(DLE・EOT)を送出している (タイムアウト等)	1. 相手局を調べる ・ 自局側からのデータ送出手法が遅い
8015H	送信処理で電文送信中、無応答リトライを規定回数行ったが、応答がなかった	sendck	1. 相手局が何らかの原因で電文に対する ACK を返していない	1. 相手局を調べる
8016H	送信処理で電文送信中、否定応答(NAK)を規定回数受け取った	sendck	1. 透過／非透過の設定が相手局と異なる 2. 誤り制御方法が相手局と異なる 3. 伝送コード長(7ビット／8ビット)が相手局と異なる 4. 相手局の受信バッファより長いデータを送信した	1. 自局と相手局の透過／非透過の設定を調べる 2. 自局と相手局の誤り制御方法の設定を調べる 3. 自局と相手局の伝送コード長の設定を調べる 4. 自局から送ろうとしている送信データ長を調べる
8017H	送信処理で中断を受け取った	sendck	1. 相手局から何らかの原因で中断を送出した	1. 相手局を調べる
8018H	WACK に対する ENQ 送信を規定回数行ったが、正常応答がこなかった (BSC 手順のみ)	bscsend sendck	1. 相手局の受信処理が遅いため	1. 相手局を調べるまたは自局側の WACK のリトライ回数を増やす

ステータス	意味	対象機能	原因	対処方法
8019H	無通信タイムアウト	bscsend nec2send nec3send sendck hrecv	1. 無通信状態が設定されている無通信監視タイマー値の時間分を過ぎ、回線を切断した (公衆回線の場合)	1. 自局または相手局のアプリケーションで、無通信状態が長く続くようになっていないかを調べるかもしくは、無通信監視タイマー値を大きくする
801AH	送信をしようとしたが、ハード的に送れない状態である	bscsend nec2send nec3send sendck	1. RS 信号を立てたが 10 秒たっても CS 信号が立ち上がってこない 2. クロックが供給されていない (同期式) 3. 送信時にキャリア (CD) が立ち上がったまま (2 線式)	1. モデムまたはケーブルを調べる 2. モデムまたはケーブルを調べる 3. モデムを調べる
801BH	送信中に RVI を受信した (B S C 手順のみ)	bscsend sendck	1. 相手局から何らかの原因で RVI が送出された	1. 相手局を調べる
801CH	送信中に EOT を受信した	bscsend nec2send nec3send sendck	1. 相手局から何らかの原因で EOT が送出された	1. 相手局を調べる
8021H	受信処理で回線切断 (DLE・EOT) を受信した	hrecv	1. 相手局から回線切断 (DLE・EOT) を受信した	1. 相手局を調べる
8022H	受信バッファオーバーフロー	hrecv	1. 受信処理で受信バッファを越えるデータを受け取った	1. 相手局を調べる
8023H	受信監視タイムアウト	hrecv	1. 受信処理で受信監視タイムアウトになった	1. 相手局を調べる
8031H	自動ダイヤルに送ったコマンドが誤っている	telon	1. 電話番号として許されない文字が含まれている	1. モデムの説明書に記述されている電話番号として許される文字を使用してください。
8035H	電話接続にて、CRN コマンドで指定された電話番号がパラメータで示されている時間 (分) 発呼を禁止している	telon	1. モデムのディレードコールの機能による	1. 一定時間あけた後、再度発呼するようにアプリケーションで対応するか、モデムの設定でディレードコールの機能ははずして下さい
8036H	電話接続にて、相手が話し中である	telon	1. 相手局が話し中	1. リダイヤルする
8038H	電話接続にて、ダイヤル開始直前に着呼が検出された	telon		1. 自局のアプリケーションで対応する

ステータス	意味	対象機能	原因	対処方法
8039H	電話接続にて、回線接続後 10 秒を経ても 2100Hz のアンサートーンが検出できない	telon		1. 自局のモデムを調べる
	電話が接続されていない	mtelon		1. mtelon をリトライする
803AH	電話接続にて、相手局からのリングバックトーンは検出されているが、相手局がでないまま 50 秒以上経過した	telon		1. 自局、相手局を調べる
803BH	電話接続にて、ダイヤル終了後モデムで設定されている時間以上経ても回線が接続されていない	telon		1. 自局のモデムを調べる

付録 HI-NET/USBの設定

【 HI-NET/USBの略図 】



1. ディップスイッチの設定

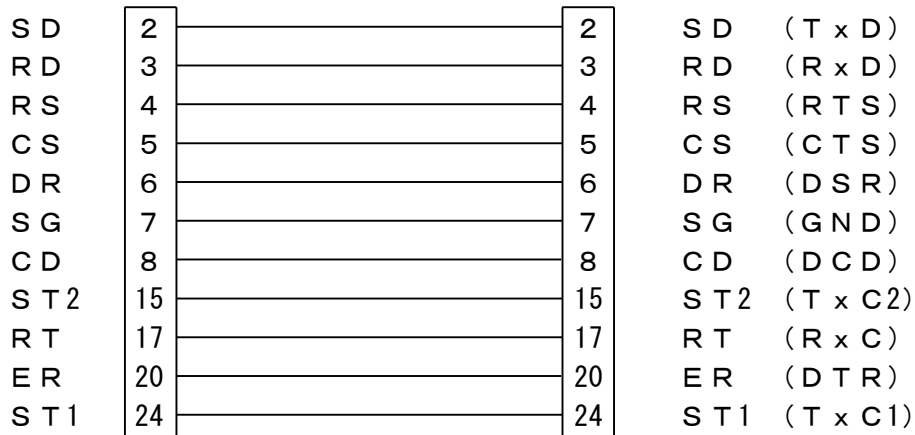
S 2 … チャンネル番号を設定します

	1	2	3	4	
ON					チャンネル番号 0 (出荷時設定)
OFF	■	■	■	■	

	1	2	3	4	
ON	■				チャンネル番号 1
OFF		■	■	■	

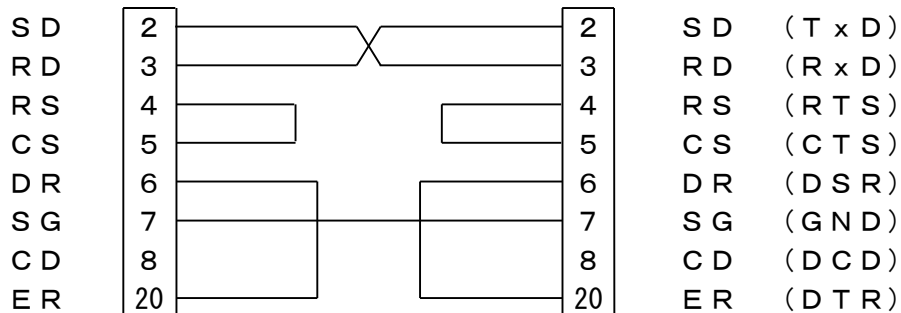
付 録 ケーブル配線図

【 対モデムケーブルの配線 】

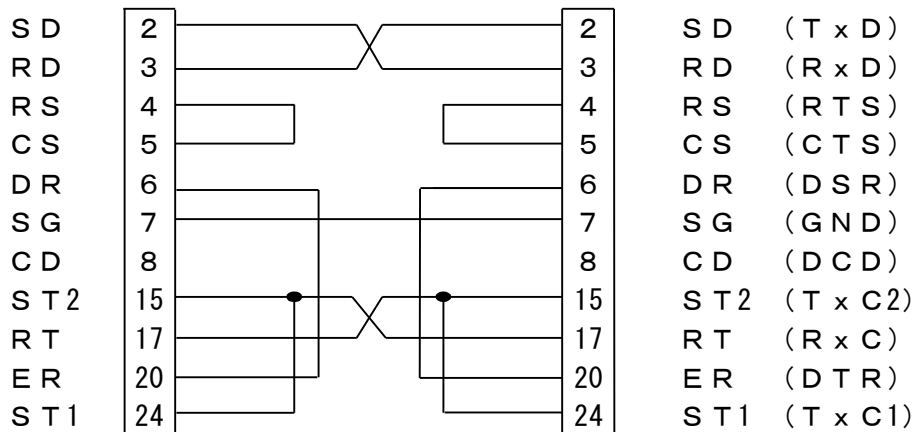


【 直結ケーブルの配線 】

① B S C (調歩同期式) ・ レベル 2 A クラス 2 ・ レベル 2 A クラス 3



② B S C (SYN 同期式) ・ レベル 2 B クラス 2 ・ レベル 2 B クラス 3



付 録 ドライバーのインストール方法

【 H I - N E T / P C I 】

1. 添付CD-ROM内のPCIフォルダには下記のファイルが納められています。

- ① Sample ----- Visual BASIC, C言語の各手順サンプル
- ② Driver (32ビットOS用) Windows2000, XP, 7, Server2008
 - HI32NT.DLL ----- DLL
 - Hi32nt.cpp ----- DLLのソース
 - HI32NT.DEF ----- DLLのソース
 - Hi32nt.lib ----- ライブラリ
 - Hiiop2000.inf ----- ボード検出時ドライバー情報
 - Hiiop2000.sys ----- ボード検出時組み込みドライバー
 - Hiiop2000_1.inf ----- ボード検出時ドライバー情報
 - Hiiop2000_1.sys ----- ボード検出時組み込みドライバー
 - Hiiop2000_2.inf ----- ボード検出時ドライバー情報
 - Hiiop2000_2.sys ----- ボード検出時組み込みドライバー
 - Hiiop2000_3.inf ----- ボード検出時ドライバー情報
 - Hiiop2000_3.sys ----- ボード検出時組み込みドライバー

2. H I - N E T / P C I ボードをコンピュータにセットして起動すると、「新しいハードウェアの検索ウィザードの開始」が表示されます。

3. 「デバイスに最適なドライバーを検索する（推奨）」を選択し、「フロッピーディスクまたは場所を指定」で添付CD-ROM内の ¥PCI¥Driver フォルダ内の Hiiop2000.inf を指定してください。

※ 1台のコンピュータにH I - N E T / P C I ボードを2枚入れて使用する場合は、2枚目のH I - N E T / P C I ボードのドライバー組込み時に Hiiop2000_1.inf を指定してください。

1台のコンピュータにH I - N E T / P C I ボードを3枚入れて使用する場合は、3枚目のH I - N E T / P C I ボードのドライバー組込み時に Hiiop2000_2.inf を指定してください。

1台のコンピュータにH I - N E T / P C I ボードを4枚入れて使用する場合は、4枚目のH I - N E T / P C I ボードのドライバー組込み時に Hiiop2000_3.inf を指定してください。

※ 2枚目以降をインストールする場合は、ドライバーの自動検索を行わずにファイルを指定してください。

《 HI-NET/USBドライバ対応表 》

OS	インストール用ファイル	アプリケーション	使用 DLL
32bit(WindowsXP 以前)	HI-NET_USB_CH0. inf	32bit	HI-NET32USB. dll
32bit(Windows7, Server2008 以降)	HI-NET_USB32(64). inf	32bit	HI-NET32USB(64). dll
64bit	HI-NET_USB32(64). inf	32bit	HI-NET32USB(64). dll
64bit	HI-NET_USB32(64). inf	64bit	HI-NET64USB. dll

HI-NET32USB(64). DLL および HI-NET64USB. DLL は自動でシステム領域にコピーされませんので、アプリケーションの実行ファイルと同じフォルダまたはシステム領域にコピーしてください。

【 HI-NET/USB 】 (Windows2000, XP)

1. 添付CD-ROM内のUSBフォルダには下記のファイルが納められています。

- ① Sample ----- Visual BASIC, C言語の各手順サンプル
- ② Driver
 - HI-NET_USB. sys ----- 組み込みドライブ
 - HI-NET32USB. dll ----- DLL
 - HI-NET_USB_CH0. inf ----- ボード検出時ドライバー情報 (1台目)
 - HI-NET_USB_CH1. inf ----- ボード検出時ドライバー情報 (2台目)
 - HI-NET_USB_CH2. inf ----- ボード検出時ドライバー情報 (3台目)
 - HI-NET_USB_CH3. inf ----- ボード検出時ドライバー情報 (4台目)
 - HI-NET_USB_CH4. inf ----- ボード検出時ドライバー情報 (5台目)
 - HI-NET_USB_CH5. inf ----- ボード検出時ドライバー情報 (6台目)
 - HI-NET_USB_CH6. inf ----- ボード検出時ドライバー情報 (7台目)
 - HI-NET_USB_CH7. inf ----- ボード検出時ドライバー情報 (8台目)

2. HI-NET/USBをコンピュータに接続すると、「新しいハードウェアの検索ウィザードの開始」が表示されます。

3. 「デバイスに最適なドライバーを検索する(推奨)」を選択し、通常は添付CD-ROM内のUSB¥Windows2000_XP¥Driverフォルダ内のHI-NET_USB_CH0. inf (チャンネル番号0の場合)を指定してください。

※1台のコンピュータにHI-NET/USBを複数接続して使用する場合は、HI-NET/USBのディップスイッチを各チャンネル番号に変更して下記のドライバーを指定してください。

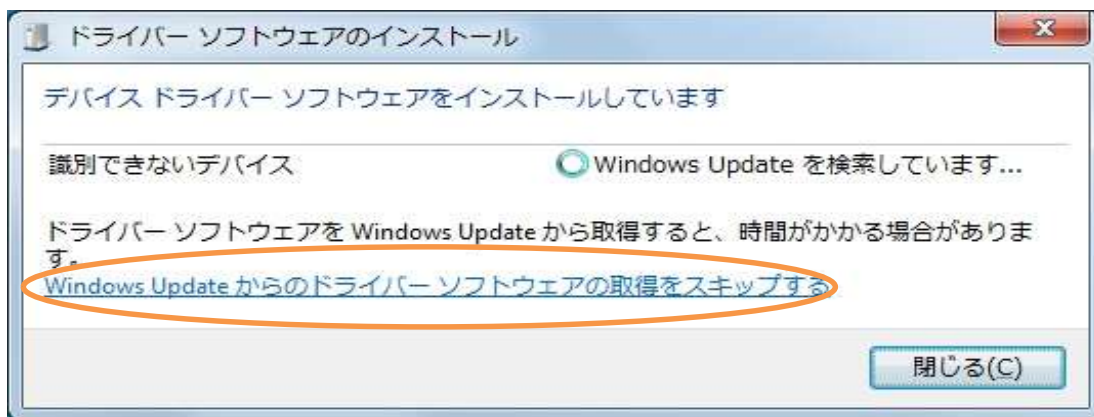
- チャンネル番号1の場合 : HI-NET_USB_CH1. inf (2台目)
- チャンネル番号2の場合 : HI-NET_USB_CH2. inf (3台目)
- チャンネル番号3の場合 : HI-NET_USB_CH3. inf (4台目)
- チャンネル番号4の場合 : HI-NET_USB_CH4. inf (5台目)
- チャンネル番号5の場合 : HI-NET_USB_CH5. inf (6台目)
- チャンネル番号6の場合 : HI-NET_USB_CH6. inf (7台目)
- チャンネル番号7の場合 : HI-NET_USB_CH7. inf (8台目)

※チャンネル番号の変更方法は「HI-NET/USBの設定」を参照してください。

【 H I - N E T / U S B 】 (Windows7, Server2008 以降)

注意: 既に Windows2000、XP 用のドライバーをインストールしている PC には Windwos7, Server2008 以降用のドライバーをインストールしないでください。
もし誤ってインストールした場合は「システムの復元」を使用して Windwos7, Server2008 以降用のドライバーをインストールする前の状態に戻してください。

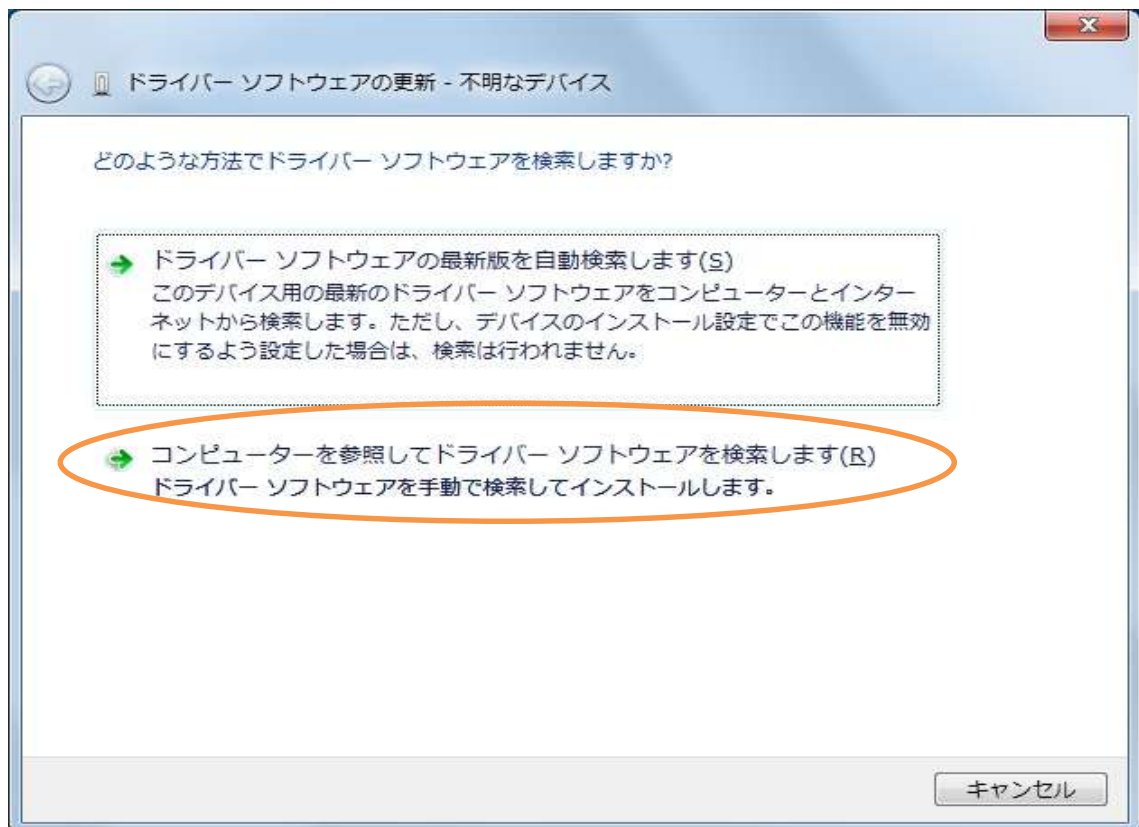
- ① H I - N E T / U S B を初めて接続すると下記のドライバーインストールの画面が表示されますので、「Windows Update からのドライバーソフトウェアの取得をスキップする」をクリックしてください。



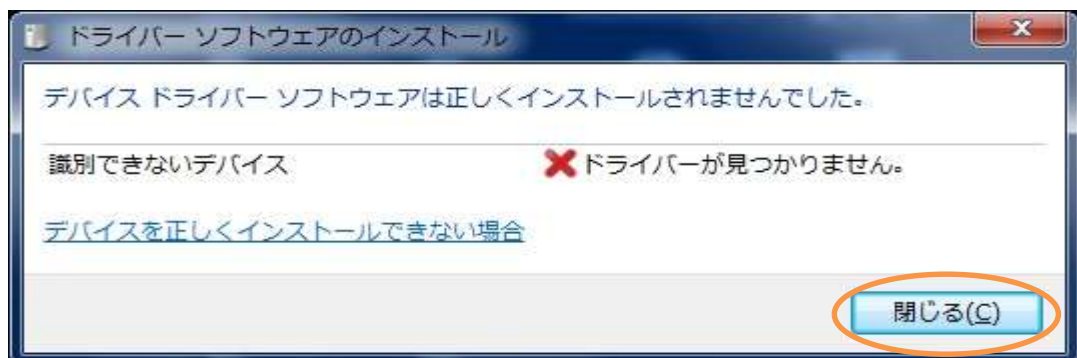
- ② 「Windows Updata からのドライバーソフトウェアの取得をスキップしますか？」のウィンドウ画面が表示されますので、「はい」をクリックしてください。



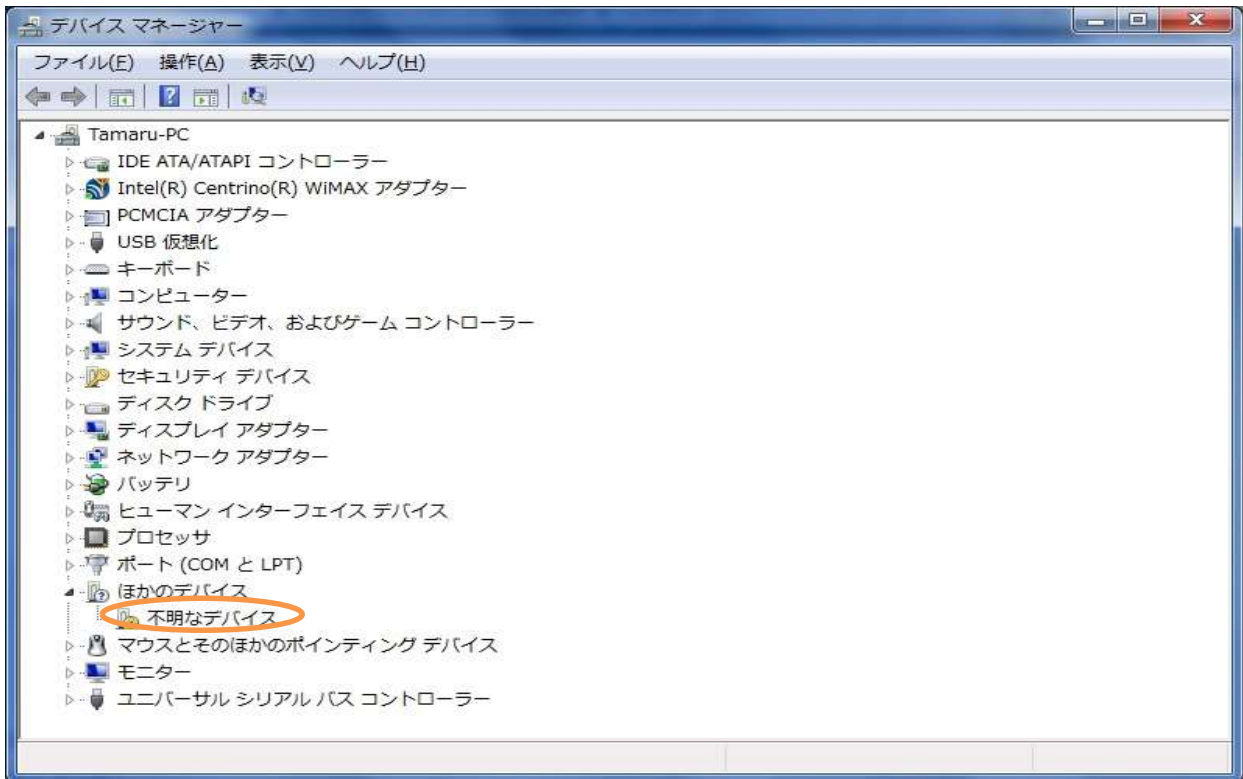
- ③ ドライバーの検索方法の選択ウィンドウが表示されますので、「コンピューターを参照してドライバーソフトウェアを検索します(R)」を選択してください。



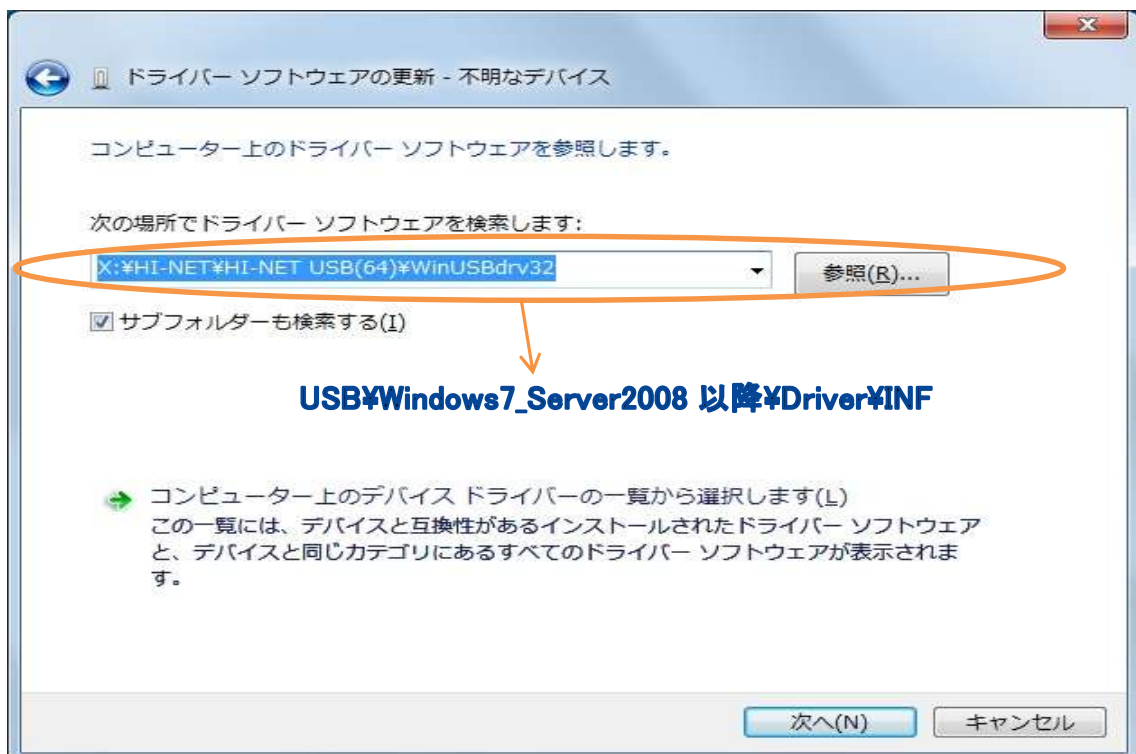
- ④ 下記の画面が表示されますので、「閉じる」ボタンをクリックして終了してください。



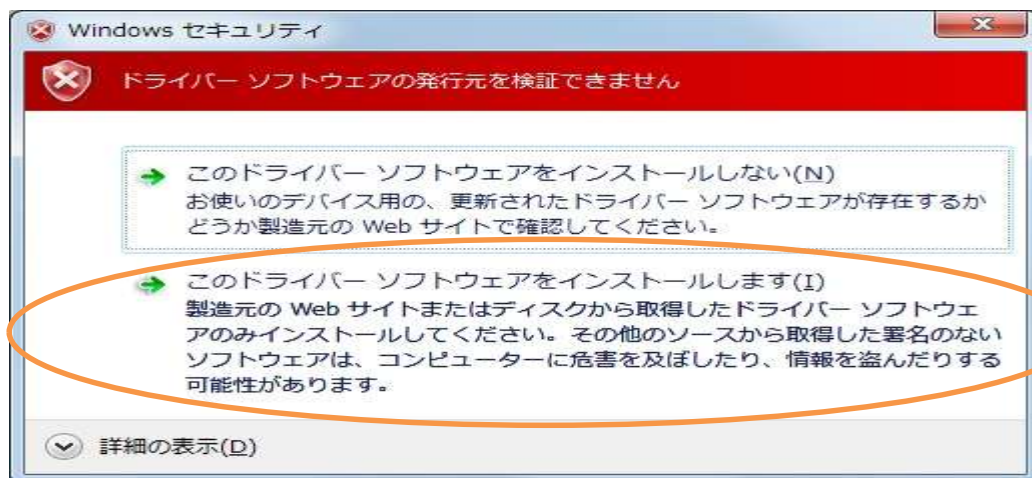
- ⑤ コントロールパネルの「システム」からデバイスマネージャーを表示してください。デバイスマネージャーに「不明なデバイス」と表示されますので、右クリックから「ドライバーの更新」を選択してください。



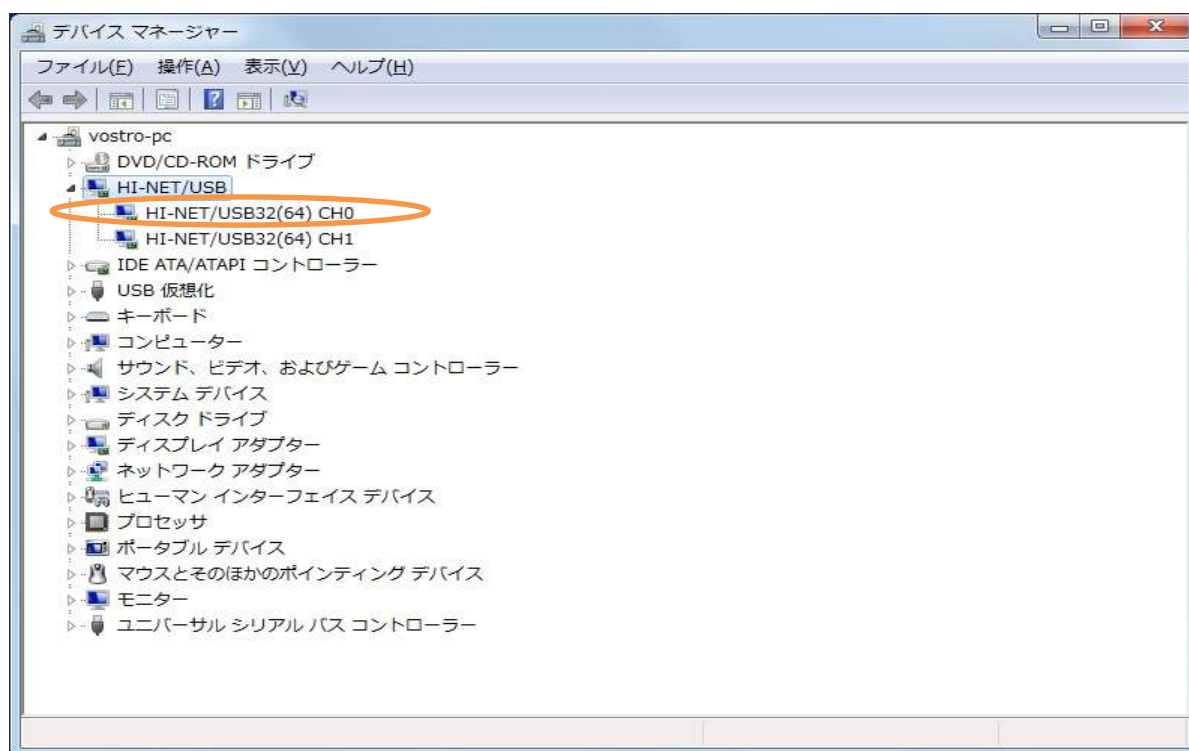
- ⑥ ドライバーの検索場所の指定では Windows7_Server2008 以降用のドライバーのフォルダを指定してください。



- ⑦ Windows セキュリティ画面が表示されますので、「このドライバーソフトウェアをインストールします(I)」を選択してください。



- ⑧ ドライバーが正しくインストールされたら、デバイスマネージャーに「HI-NET/USB 32(64) CHn」が表示されます。(n:0, 1, …)

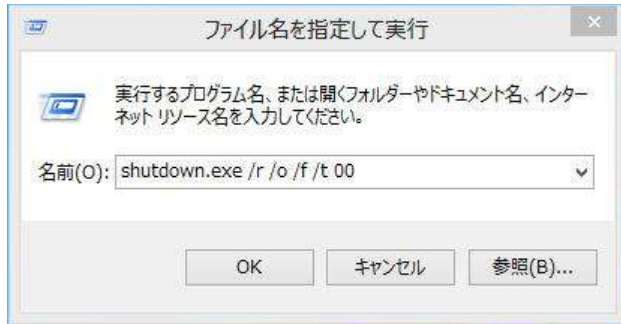


※ HI-NET/USB を複数台使用する場合は、2 台目以降も同じ手順で行ってください。

【 H I - N E T / U S B 】 (Windows8, Server2012)

- ① Windows8 等(64ビット OS)ではドライバーのインストール時に「デバイスのドライバーソフトウェアのインストールに関する問題が発生しました」が表示されインストールできませんので、「ドライバー署名の強制を無効化」を下記の手順で行ってからインストールして下さい。

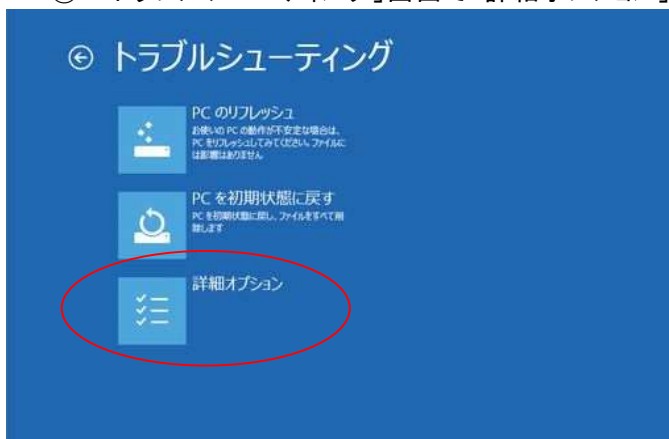
「Windows」キー＋「R」で「ファイル名を指定して実行」を呼び出して「shutdown.exe /r /o /f /t 00」を入力して下さい。



- ② 「オプションの選択」画面で「トラブルシューティング」を選択して下さい。



- ③ 「トラブルシューティング」画面で「詳細オプション」を選択して下さい。



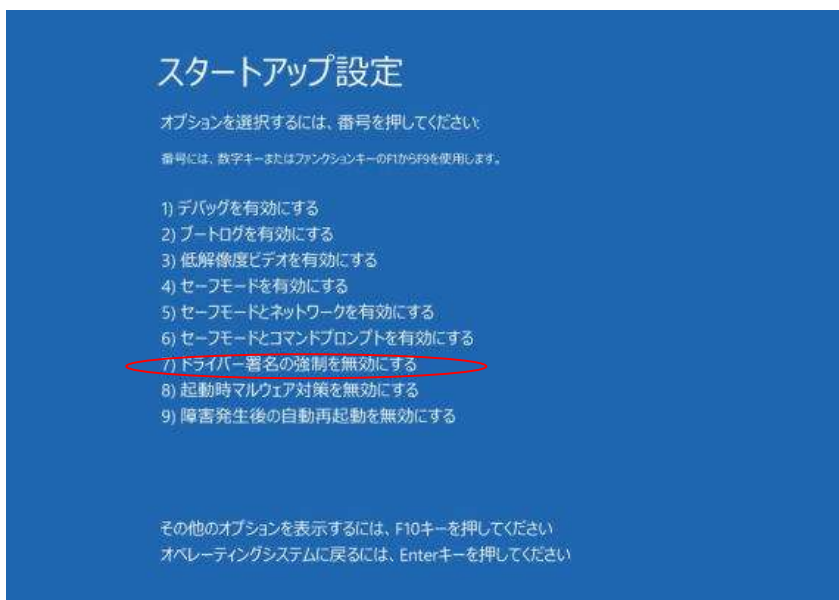
④ 「詳細オプション」画面で「スタートアップ設定」を選択して下さい。



⑤ 「スタートアップ設定」画面で「再起動」を選択して下さい。



⑥ 再起動後、「スタートアップ設定」画面で、「7) ドライバー署名の強制を無効にする」を選択して下さい。

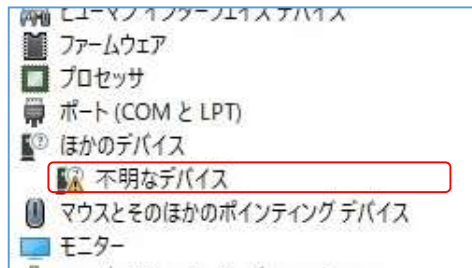


⑦ Windows が起動したら、再度ドライバーをインストールして下さい。
(デバイスマネージャーで「ドライバーの更新」を実行して下さい。)
77 ページからの「ドライバーのインストール方法 HI-NET/USB(Windows7, Server2008)以降」を参照して下さい。

※HI-NET/USB を複数台使用の場合は、2 台目以降も同じ手順で行ってください。

【 HI-NET/USB 】 (Windows10, Server2016)

- ① HI-NET/USB を初めて接続した後にデバイスマネージャーを開くと「ほかのデバイス」内に「不明なデバイス」と表示されます。



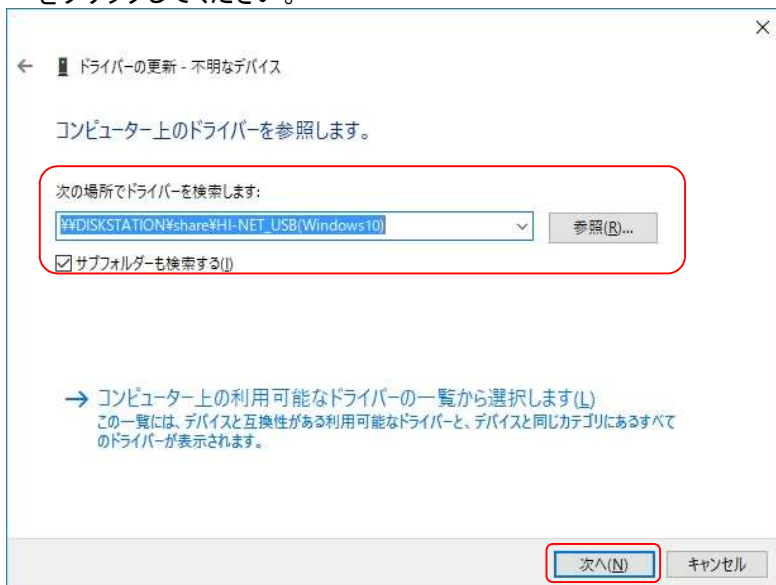
- ② 「不明なデバイス」でプロパティを表示し、「ドライバーの更新」をクリックしてください。



- ③ 「コンピュータを参照してドライバーソフトウェアを検索」をクリックしてください。



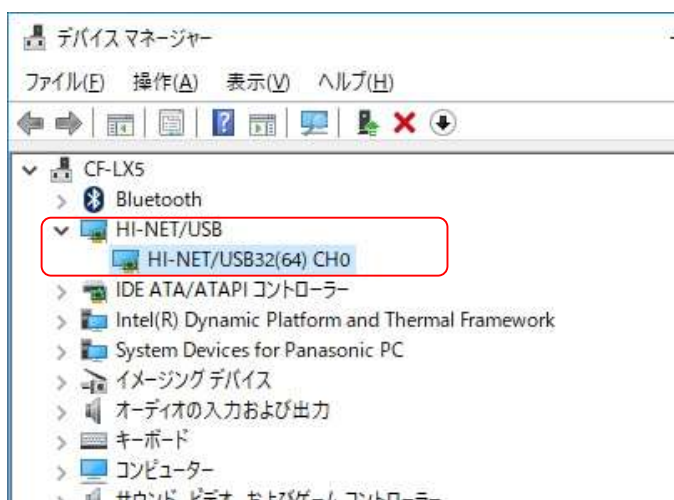
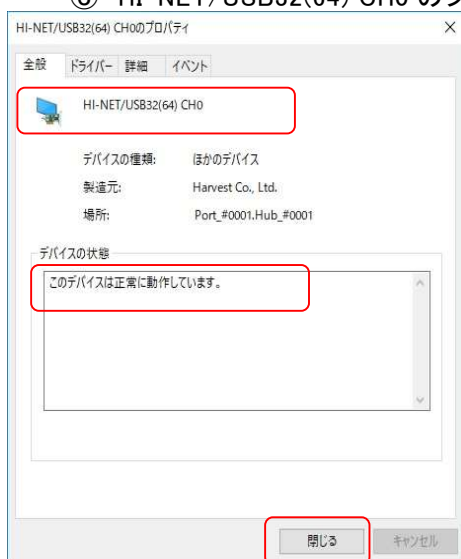
- ④ 「参照」からHI-NET/USB のWindows10用のドライバーが存在するフォルダを指定して「次へ」をクリックしてください。



- ⑤ 「ドライバーが正常に更新されました」が表示されますので「閉じる」をクリックしてください。



- ⑥ HI-NET/USB32(64) CH0のプロパティが正しく表示されます。



※HI-NET/USB を複数台使用する場合、2 台目以降も同じ手順で行ってください。

表 A. 1 JIS 8からEBCDICカタカナコードへのコード変換表

列 行	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL 00	DLE 10	SP 40	0 F0	@ 7C	P D7						- B1	タ 58	ミ 91		
1	SOH 01	DC1 11	! 4F	1 F1	A C1	Q D8					。 41	ア 81	チ 92	ム A6		
2	STX 02	DC2 12	“ 7F	2 F2	B C2	R D9					「 42	イ 82	ツ 93	メ A7		
3	ETX 03	DC3 13	# 7B	3 F3	C C3	S E2					」 43	ウ 83	テ 94	モ A8		
4	EOT 37	DC4 3C	\$ E0	4 F4	D C4	T E3					、 44	エ 84	ト 95	ヤ A9		
5	ENQ 2D	NAK 3D	% 6C	5 F5	E C5	U E4					・ 45	オ 85	ナ 96	ユ AA		
6	ACK 2E	SYN 32	& 50	6 F6	F C6	V E5					ヲ 46	カ 86	ニ 97	ヨ AC		
7	BEL 2F	ETB 26	' 7D	7 F7	G C7	W E6					ア 47	キ 87	ヌ 98	ラ AD		
8	BS 16	CAN 18	(4D	8 F8	H C8	X E7					イ 48	ク 88	ネ 99	リ AE		
9	HT 05	EM 19) 5D	9 F9	I C9	Y E8					ウ 49	ケ 89	ノ 9A	ル AF		
A	LF 15	SUB 3F	* 5C	: 7A	J D1	Z E9					エ 51	コ 8A	ハ 9D	レ BA		
B	VT 0B	ESC 27	+ 4E	; 5E	K D2	[4A					オ 52	サ 8C	ヒ 9E	ロ BB		
C	FF 0C	FS 1C	, 6B	< 4C	L D3	¥ 5B					ヤ 53	シ 8D	フ 9F	ワ BC		
D	CR 0D	GS 1D	- 60	= 7E	M D4] 5A					ユ 54	ス 8E	ヘ A2	ン BD		
E	SO 0E	RS 1E	. 4B	> 6E	N D5	^ 5F					ヨ 55	セ 8F	ホ A3	ン BE		
F	SI 0F	US 1F	/ 61	? 6F	0 D6	- 6D					ッ 56	ソ 90	マ A4	ン BF		

備考

上段
下段

 JIS 8コード
EBCDICカタカナコードの16進表示

表 A. 2 EBCDICカタカナからJIS8コードへのコード変換表

列 行	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL 00	DLE 10	80	90	SP 20	& 26	- 2D	6B	74	7 BF	78	7A	{ 7B	} 7D	\$ 24	0 30
1	SOH 01	DC1 11	81	91	。 A1	ヱ AA	/ 2F	6C	7 B1	夕 C0	- 7E	A A0	J 41			1 31
2	STX 02	DC2 12	82	16	〒 A2	オ AB	63	6D	イ B2	チ C1	^ CD	B E0	K 42	S 4B		2 32
3	ETX 03	DC3 13	83	93	↓ A3	ヤ AC	64	6E	ウ B3	ツ C2	ホ CE		C E1	L 43	T 4C	3 33
4	9C	9D	84	94	、 A4	ユ AD	65	6F	エ B4	テ C3	マ CF		D E2	M 44	U 4D	4 34
5	HT 09	LF 0A	85	95	・ A5	ヨ AE	66	70	オ B5	ト C4	ミ D0		E E3	N 45	V 4E	5 35
6	86	08	BS 08	ETB 17	ヲ A6	ツ AF	67	71	カ B6	ナ C5	ム D1		F E4	O 46	W 4F	6 36
7	DEL 7F	87	ESC 1B	EOT 04	7 A7		68	72	キ B7	ニ C6	メ D2		G E5	P 47	X 50	7 37
8	97	CAN 18	88	98	イ A8	- B0	69	73	ク B8	ヌ C7	モ D3		H E6	Q 48	Y 51	8 38
9	8D	EM 19	89	99	ウ A9		6A	60	ケ B9	ネ C8	ヤ D4		I E7	R 49	Z 52	9 39
A	8E	92	8A	9A	[5B] 5D	7C	3A	コ BA	ノ C9	ユ D5	レ DA				
B	VT 0B	8F	8B	9B	・ 2E	¥ 5C	,	# 2C	23	75	76	79	□ DB			
C	FF 0C	FS 1C	8C	14	< 3C	*	% 2A	@ 25	40	77	7D	7C	ヨ D6	ワ DC		
D	CR 0D	GS 1D	ENQ 05	NAK 15	(28) 29	- 5F	,	シ BC	ハ CA	ラ D7	ソ DD				
E	SO 0E	RS 1E	ACK 06	9E	+ 2B	; 3B	> 3E	= 3D	ス BD	ヒ CB	リ D8	リ DE				
F	SI 0F	US 1F	BEL 07	SUB 1A	! 21	^ 5E	? 3F	“ 22	セ BE	フ CC	ル D9	° DF				

備考

上	段
下	段

 EBCDICカタカナコード
JIS8コードの16進表示

付 録 安全にお使いいただくために必ずお守りください



お客様や他の人々への危害や財産への障害を未然に防ぎ、本製品を安全にお使いいただくために守っていただきたい事項を記載しました。

正しく使用するために、必ずお読みになり、内容をよく理解した上でお使いください。なお、本書には弊社製品だけでなく、弊社製品を組み込んだパソコンシステム運用全般に関する注意事項も記載されています。

パソコンの故障／トラブルや、いかなるデータの消失・破損または、取り扱いを誤ったために生じた本製品の故障／トラブルは、弊社の保証対象には含まれません。あらかじめご了承ください。

■ 使用している表示の意味

警告表示の意味

 警告	絶対に行ってはいけないことを記載しています。この表示の注意事項を守らないと、使用者が死亡または、重傷を負う可能性が想定される内容を示しています。
 注意	この表示の注意事項を守らないと、使用者がけがをしたり、物的損害の発生が考えられる内容を示しています。



- I. **液体や異物が内部に入ったら、電源スイッチをOFFにし、ACコンセントからプラグを抜いてください。**
そのまま使用を続けていると、ショートして火災や感電する恐れがあります。
- II. **風呂場など、水分や湿気が多い場所では、本製品を使用しないでください。**
火災や感電・故障する恐れがあります。
- III. **本製品の分解や改造や修理を自分でしないでください。**
火災や感電する恐れがあります。
- IV. **本製品の取り付け／取り外しをするときは、本製品およびパソコン、周辺機器の電源スイッチをOFFにし、電源プラグをACコンセントから抜いてください。**
電源プラグがコンセントに接続されたまま、取り付け／取り外しを行うと、感電および故障の原因となります。
- V. **小さなお子様が電気製品を使用する場合には、本製品の取り扱い方法を理解した大人の監視、指導のもとで行うようにしてください。**

VI. 電気製品の内部やケーブル、コネクタ類に小さなお子様の手が届かないように機器を配置してください。

触ってけがをする危険があります。

VII. 本製品を落としたり、強い衝撃を与えたりしないでください。与えてしまった場合は、すぐに電源スイッチをOFFにして、電源プラグを抜いてください。

そのまま使用を続けると、ショートして火災や感電する恐れがあります。

VIII. 本製品の取り付け、使用する際は、必ずパソコンメーカーおよび周辺機器メーカーが提示する警告・注意指示に従ってください。

IX. 煙が出たり変な臭いや音がしたら、すぐに電源をOFFにし、ACコンセントから電源プラグを抜いてください。

そのまま使用を続けると、ショートして火災や感電する恐れがあります。



I. 本製品の取り付け／取り外しや、ソフトウェアをインストールするときなど、お使いのパソコン環境を少しでも変更するときは、変更前に必ずパソコン内（ハードディスク等）すべてのデータのバックアップ（CD、MOディスク等）を作成してください。

誤った使い方をしたり、故障などが発生してデータが消失、破損したときなど、バックアップがあれば被害を最小限に抑えることができます。

データが消失、破損したことによる損害については、弊社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。

II. 電源ケーブルがACコンセントに接続されているときには、濡れた手で本製品にさわらないでください。

感電の原因となります。

III. ハードディスク内のデータは、必ず他のメディア（CD、MOディスク等）にバックアップしてください。

とくに、修復・再現できない重要なデータは、オリジナルの更新前・更新後と、常に二重のバックアップを作成されることをお勧めします。

以下のような場合に、データは消失・破損する恐れがあります。

- ・誤った使い方をしたとき
- ・静電気や電氣的ノイズの影響を受けたとき
- ・故障・修理などのときに
- ・パソコンの電源OFF直後に、すぐに電源を入れたとき
- ・長時間使っていなかったために電池が自然放電したとき
- ・天災による被害を受けたとき

上記の場合、またその他のいかなる場合でも、データ消失・破損したことによる損害について弊社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。

IV. 静電気による破損を防ぐため、本製品に触れる前に、身近な金属（ドアノブやアルミサッシ等）に手を触れて、身体の静電気を取り除くようにしてください。

人体などからの静電気は、本製品を破損、または、データを消失・破損させる恐れがあります。

V. ハードディスク・CD・MOドライブなどの、データの格納用機器へのアクセス中は、パソコンや機器の電源をOFFにしたり、リセットしないでください。

データが消失・破損する恐れがあります。データが消失、破損したことによる損害については、弊社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。

VI. アプリケーションソフトの動作中に電源をOFFにしたり、リセットしないでください。

データが消失・破損する恐れがあります。データが消失、破損したことによる損害については、弊社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。

VII. 次の場所には設置しないでください。感電、火災の原因となったり、製品やパソコンに悪影響を及ぼすことがあります。

強い磁界が発生するところ	故障の原因となります。
静電気が発生するところ	故障の原因となります。
振動が発生するところ	けが、故障、破損の原因となります。
平らでないところ	転倒したり、落下してけがや故障の原因となります。
温度、湿度がパソコンのマニュアルが定めた仕様環境を超える、または結露するところ	故障の原因となります。
直射日光があたる場所	故障や変形の原因となります。
火気の周辺、または熱気のかもるところ	故障や変形の原因となります。
漏電の危険があるところ	故障や感電の原因となります。
漏水の危険があるところ	故障や感電の原因となります。

■ 使用上のお願い

H I - N E T ボードは精密機器です。正しいご使用のために、本書を必ずお読みください。
パソコンの故障／トラブルまたは、取り扱いを誤ったために生じたH I - N E T ボードの故障／トラブルは、弊社の保証対象には含まれません。

■ 取付けについて

- ・パソコンおよび周辺機器の電源OFFにし、ケーブル類を取り外してから作業を行ってください。そのまま取り付けや取り外しを行うと、感電するおそれがあります。
- ・パソコンおよび周辺機器の取扱いは、それぞれ付属のマニュアルに記載されている手順で行ってください。
- ・各種コネクタのチリ・ホコリなどは取り除いてください。

< 注意事項 >

この装置は、情報処理装置等電波障害自主規制協議会（VCCI）の基準に基づくクラスA情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよう要求されることがあります。

ハーベスト 株式会社

〒564-0063 大阪府吹田市江坂町1丁目17番14号
江坂吉川ビル5階

TEL 06-4861-0811 (代)
FAX 06-4861-0812

