

イーサネット I / F 汎用入出力ユニット

U I O - 2 1 4 4 E N B

コマンド説明書

for サーバーモード

エムシーアイエンジニアリング株式会社
〒194-0212 東京都 町田市 小山町 7 8 9 - 9
TEL 042-705-8312 FAX 042-794-8317



URL : <http://www.mci-eng.co.jp>

目次

[1] 概要		
[1-1] 概略動作	_____	2
[1-2] フォーマット	_____	2
[1-3] コマンド	_____	2
[1-4] パラメータ	_____	2
[1-5] デリミタ (ターミネータ)	_____	3
[1-6] エラー処理	_____	3
[2] 共通コマンド		
[2-1] システムデータ・コマンド	_____	4
[2-2] 内部操作・コマンド	_____	4
[2-3] 同期・コマンド	_____	5
[2-4] ステータス/イベント・コマンド	_____	6
[3] ステータス報告システム		
[3-1] ステータス・ビット・レジスタ	_____	7
[3-2] スタンダード・イベント・ステータス・レジスタ	_____	8
[3-3] ポート・ステータス・レジスタ	_____	9
[3-4] ステータス・レジスタの初期値	_____	10
[4] S C M C コマンド		
[4-1] 入力ポートからの入力コマンド	_____	12
[4-2] 出力ポートへの出力コマンド	_____	14
[4-3] ポート・ステータス操作コマンド	_____	17
[5] イーサネットアクセスDLLの使用法		
[5-1] 概要	_____	20
[5-2] 動作環境	_____	20
[5-3] 関数	_____	20
[5-4] エラーコード	_____	21

[1] 概要

本「コマンド説明書」ではUIO-2144ENBをサーバーモードで使用する場合の操作コマンドについて説明します。ハード的な仕様や対向モードでの使用方法については、「取扱説明書」を参照して下さい。

UIO-2144ENBのサーバーモードの操作コマンドやその機能は、IEEE-Std 488.2-1992を基本として構成、構築されています。

[1-1] 概略動作

本機「UIO-2144ENB」はイーサネット端末の汎用デジタル入出力ユニットです。従って本機をサーバーモードで使用する場合は、パソコンなどのホスト機が必要です。

本機にコマンド（メッセージ）を送信するとコマンドの内容により、本機の端末側出力ポートをON/OFFの制御を行ったり、入力ポートに入力されたデータを読み取ったりすることができます。また、ステータス操作コマンドを使って、端末側入力ポートのデータの変化を知ることができます。

本機には全部で五つのポートがあり、どのポートを入力にするか、出力にするかをユーザーが決めることができます。（取扱説明書 [2-1] を参照して下さい）

[1-2] フォーマット

パソコンなどのホスト機からのメッセージのフォーマットは下記の二つのタイプがあります。

- 1：コマンド デリミタ
コマンドのみで、パラメータを必要としないメッセージです。
- 2：コマンド パラメータ デリミタ
パラメータを必要とするコマンドのメッセージです。

UIO-2144ENBからの応答メッセージは、無い場合と、パラメータのみを返送する場合との二つのタイプがあります。どちらの場合でもディップスイッチで選択されたデリミタで終了します。（本書 [1-5] を参照）

[1-3] コマンド

488.2で規定されている共通コマンド、および、488.2で規定されているフォーマットに基づいたSCMC (Standard Commands for Measurement and Control) コマンドを使用することができます。

SCMCコマンドのニーモニックは
[]の部分は省略可能です。コマンド文字列の小文字の部分は省略してもかまいません。
省略しない場合はすべて大文字で表記して下さい。

[1-4] パラメータ

数値パラメータとして、10進数、16進数、8進数、2進数が使用できます。

- | | | |
|----------------|------|--|
| 16進数数値のフォーマットは | #H数値 | (数値は 0, 1, 2, ..., 9, A, B, C, D, E, F の組み合わせ) |
| 8進数数値のフォーマットは | #Q数値 | (数値は 0, 1, 2, 3, 4, 5, 6, 7 の組み合わせ) |
| 2進数数値のフォーマットは | #B数値 | (数値は 0, 1 の組み合わせ) |

数値で表現しないパラメータは英大文字（アルファベット）で表現します。
入出力ポートの名称など、本機に内蔵される信号名や機能名を指定する場合に使用します。
各コマンドの解説で具体的な名称が列記されています。

[1-5] デリミタ (ターミネータ)

UIO-2144ENBが応答メッセージの最後に付加するデリミタ (ターミネータ) はディップスイッチで下記の4種類の中から選択することができます。(取扱説明書 [2-2] を参照)

SW7	SW8	デリミタ (ターミネータ)
OFF	OFF	CR
OFF	ON	CR+LF (NL)
ON	OFF	EOT
ON	ON	LF (NL)

UIO-2144ENBがデリミタとして認識して受け取れるデリミタは下記の2種類です。

- 1 : ニューライン (NL)
 - 2 : ディップスイッチで選択されているデリミタ
- この2種類を選択する方法はありません。コマンドやパラメータの組み合わせで自動的に認識します。

[1-6] エラー処理

文法エラー : 本機が受け取ったコマンドがフォーマットに適合していない場合や未定義コマンドの場合、文法エラーになります。
このエラーが発生するとスタンダード・イベント・ステータス・レジスタのbit 5 (CME) がON (1) になります。

対処 : 正しいコマンドを再度送って下さい。

実行エラー : コマンドがフォーマットに適合していても、範囲外パラメータの場合、実行エラーになります。
また、事前のコマンドにより、本機が実行中の作業と排他しなければならない場合も実行エラーになります。(排他関係は各コマンドの説明を参照)
このエラーが発生するとスタンダード・イベント・ステータス・レジスタのbit 4 (EXE) がON (1) になります。

対処 : 正しいパラメータに修正して、再度送って下さい。
または、排他関係を確認し、実行可能な時に送って下さい。

機器エラー : 本機は電源投入直後、プログラムROMとシステムワークRAMをチェックします。
チェックの結果、異常を発見するとスタンダード・イベント・ステータス・レジスタのbit 3 (DDE) をON (1) にします。

対処 : 一度電源を断にし、再度電源を投入してもこのエラーが発生する場合は修理に出して下さい。
(なお、*TST? によるセルフテストでの異常の場合も同様に修理が必要です。)

[2] 共通コマンド

[2 - 1] システムデータ・コマンド

□ *IDN? 識別クエリ (Identification Query)

書式 *IDN?

説明 バスに接続されている機器を識別する文字列を読み出します。

応答 当コマンドを受信した後、本機はトーカに指定されると
 <製造業者>, <モデル番号>, <シリアル番号>, <ファームウェアのバージョン>を表す、
 下記の文字列を返します。
 MCI-ENG, UIO-2144EN, 000000, REV1.00

[2 - 2] 内部操作コマンド

□ *RST リセット (Reset)

書式 *RST

説明 機器をリセットします。

下記の内容のリセットを行います。

- * 出力ポートをリセットする (出力ポートは負論理のため High になる)
- * ホストからの受信バッファをクリアする
- * INPUT コマンドシステムを初期状態にする
- * 前に受け取った *0PC または *0PC? コマンドをクリアする

下記の内容はリセットされません。

- * IP アドレスまたはそのアドレス内容
- * 出力待ち行列の中のデータ・バイト
- * ステータス・バイト・レジスタ
- * サービス・リクエスト・イネーブル・レジスタ
- * スタンダード・イベント・ステータス・レジスタ
- * スタンダード・イベント・イネーブル・レジスタ
- * ポート・ステータス・条件・レジスタ
- * ポート・ステータス・トランジション・レジスタ
- * ポート・ステータス・イベント・レジスタ
- * ポート・ステータス・イネーブル・レジスタ
- * 電源オン・フラグ

応答 当コマンドに対する応答メッセージはありません。

□ *TST? セルフテストクエリ (Self-Test Query)

書式 *TST?

説明 機器に内部セルフテストを実行させ、テストの結果を報告させます。
 テストの内容は下記の2点です。

- ◎ プログラムROMのサムチェック
- ◎ ユーザワークRAMのリードライトチェック

現在実行中の作業がある場合はテストの実行はできません。
 出力データなどの端末側への出力信号の状態、
 ステータス報告システムの各レジスタ、は初期化されません。

応答 当コマンドを受信すると本機はセルフテストを実行し、結果を報告します。
 結果の内容は下記の数値 (10進数の整数) のいずれかで、エラーがあった場合の数値は負です。

- 0 テストはすべて正常
- 1 プログラムROMのチェックサムエラー
- 2 ユーザワークRAMのリードライトエラー
- 90 実行中の作業があったため、テストを実行しなかった。

複数のエラーが発生した場合の数値は各エラーの数値の和を報告します。
 (例えば、-1 と -2 のエラーが発生すると -3 を報告します。)

[3 - 3] 同期コマンド

 *OPC 動作完了 (Operation Complete)

書式 *OPC

説明 実行待ち動作がすべて完了したら、スタンダード・イベント・ステータス・レジスタのビット0をセットするように機器に命令します。

応答 当コマンドを受信すると本機は現在実行中の作業がすべて終了したらスタンダード・イベント・ステータス・レジスタのビット0をセットします。

 *OPC? 動作完了 (Operation Complete Query)

書式 *OPC?

説明 実行待ち動作がすべて完了したら、機器の出力待ち行列（ホストへの送信バッファ）にASCII「1」を入れるように機器に命令します。

応答 当コマンドを受信すると本機は現在実行中の作業がすべて終了したら出力待ち行列にASCII「1」を入れます。その後、それを送信します。

 *WAI 続行待ち (Wait-to-Continue)

書式 *WAI

説明 前に受け取ったコマンドやクエリがすべて終了するまで、新たなコマンドの実行を保留させます。

応答 当コマンドを受信すると本機は現在実行中の作業がすべて終了するまで新たなコマンドを実行しません。現在実行中の作業がすべて終了するとあらたなコマンドを実行します。

関連 *OPC, *OPC?

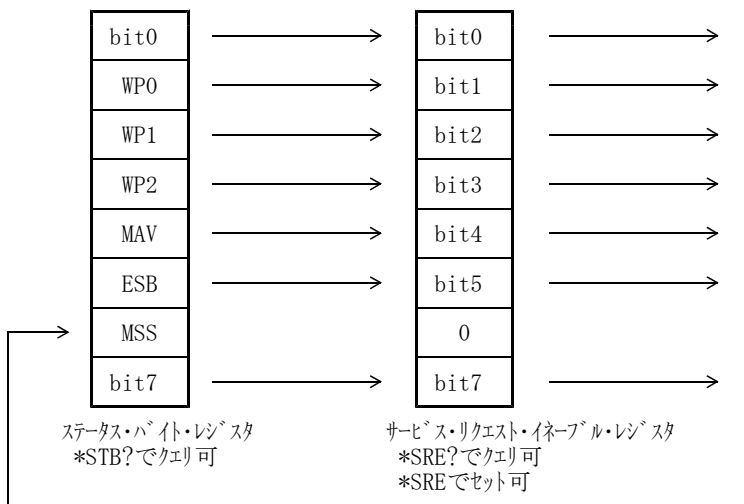
[2 - 4] ステータス/イベント・コマンド

- *CLS ステータス・クリア (Clear Status)
- 書式 *CLS
- 説明 ステータスに関する下記のレジスタをクリアします。
スタンダード・イベント・ステータス・レジスタのすべてのビット
ポート・ステータス・イベント・レジスタのすべてのビット
- 応答 このコマンドに対する応答はありません。
- *ESE スタンダード・イベント・ステータス・イネーブル (Standard Event Status Enable)
- 書式 *ESE 設定値
- 説明 スタンダード・イベント・イネーブル・レジスタに設定値をセットします。
設定値は” 0 ” から” 2 5 5 ” までの値を 1 0 進数または 1 6、8、2 進数で表します。
- 応答 このコマンドに対する応答はありません。
- *ESE? スタンダード・イベント・ステータス・イネーブル・クエリ (Event Status Enable Query)
- 書式 *ESE?
- 説明 スタンダード・イベント・イネーブル・レジスタの内容を読み出します。
- 応答 戻り値は” 0 ” から” 2 5 5 ” の範囲の 1 0 進数整数値です。
- *ESR? イベント・ステータス・レジスタ・クエリ (Event Status Register Query)
- 書式 *ESR?
- 説明 スタンダード・イベント・ステータス・レジスタの内容を読み出します。
読み出されたスタンダード・イベント・ステータス・レジスタはクリアされます。
- 応答 戻り値は” 0 ” から” 2 5 5 ” の範囲の 1 0 進数整数値です。
- *SRE サービス・リクエスト・イネーブル (Service Request Enable)
- 書式 *SRE 設定値
- 説明 サービス・リクエスト・イネーブル・レジスタに設定値をセットします。
設定値は” 0 ” から” 2 5 5 ” までの値を 1 0 進または 1 6、8、2 進数で表します。
- 応答 このコマンドに対する応答はありません。
- *SRE? サービス・リクエスト・イネーブル・クエリ (Service Request Enable Query)
- 書式 *SRE?
- 説明 サービス・リクエスト・イネーブル・レジスタの内容を読み出します。
- 応答 値は” 0 ” から” 6 3 ”、” 1 2 8 ” から” 1 9 1 ” の範囲の 1 0 進数整数値です。
- *STB? ステータス・バイト・クエリ (Read Status Byte Query)
- 書式 *STB?
- 説明 ステータス・バイトを読み出します。
- 応答 戻り値は” 0 ” から” 2 5 5 ” の範囲の 1 0 進数整数値です。

【3】ステータス報告システム

[3-1] ステータス・バイト・レジスタ

- bit 0 : : 本機においては常に0です。
- bit 1 : WP 0 : WP 0・ポート・ステータス・レジスタを代表するサマリ・ビット
- bit 2 : WP 1 : WP 1・ポート・ステータス・レジスタを代表するサマリ・ビット
- bit 3 : WP 2 : WP 2・ポート・ステータス・レジスタを代表するサマリ・ビット
- bit 4 : MAV : メッセージ・アベイラブル・ビット
機器のデータ出力の待ち行列が空であるかどうかを示します。
本機のホストへの送信バッファに送信データが有る場合、1にセットされます。
- bit 5 : ESB : イベント・ステータス・ビット
あらかじめ許可された「スタンダード・イベント」が発生した場合、1にセットされます。
- bit 6 : MSS : マスター・ステータス・サマリ
ステータス・ビット・レジスタの他の7ビットの代表。
過去に本機がサービス・リクエストが発生したかどうかを示します。
- bit 7 : : 本機においては常に0です。



例 (Visual Basic 6)

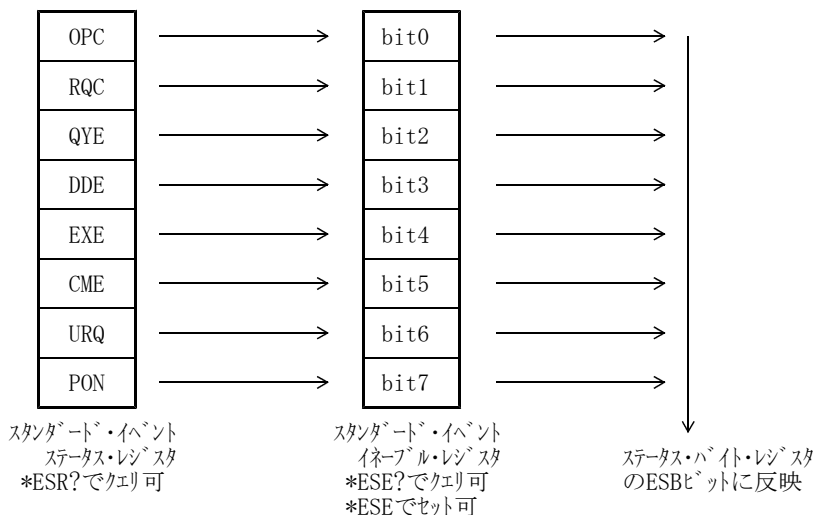
```

Dim ret as Long
Dim IPAddress as String
Dim SendStr as String
Dim SendSize as Long
Dim Size as Long
Dim RecvStr as String
Dim Buff as String

IPAddress = "192.168.16.100"
SendStr = "*STB?" & vbCrLf 'デリミタをLFに設定している場合
SendSize = Len(SendStr)
RecvStr = " " '想定される応答文が入るために十分なサイズを確保する
Size = Len(RecvStr)
Delim = &H0A 'デリミタをLFに設定している場合
ret = En_SendRecvStr(IPAddress, SendStr, SendSize, RecvStr, Size, Delim)
If ret < 0 Then
    エラー処理を記述する
Else
    Buff = Mid(RecvStr, 1, Size) 'Size には受信した応答文の真のサイズが入っている
End If
    
```


[3-2] スタンダード・イベント・ステータス・レジスタ (SESR)

- bit 0 : OPC : 動作完了
本機が処理を完了し、新しいコマンドを受け入れる状態であることを示します。
このビットは動作完了コマンド (*OPC) の応答として発生します。
- bit 1 : RQC : リクエスト・コントロール
本機においては常に0です。
- bit 2 : QYE : クエリ・エラー
本機においては常に0です。
- bit 3 : DDE : 機器定義エラー
本機が電源投入された場合、プログラムROMのサムチェックとシステムワークRAMのリードライトチェックを行い、エラーが発生した場合1になります。
- bit 4 : EXE : 実行エラー
本機がコマンド実行時にエラーが発生したことを示します。
原因は、本機がサポートしていないコマンドを受け取ったか、
現在の本機の状態では実行不可能なコマンドを受け取ったことによります。
- bit 5 : CME : コマンド・エラー
本機が受け取ったコマンドがフォーマットに適合していない場合に発生します。
- bit 6 : URQ : ユーザ・リクエスト
本機においては常に0です。
- bit 7 : PON : パワー・オン
スタンダード・イベント・ステータス・レジスタを最後にクエリして以降、
本機の電源を入れなおしたことを示します。



例 (Visual Basic 6)

```

Dim ret as Long
Dim IpAddress as String
Dim SendStr as String
Dim SendSize as Long
Dim Size as Long
Dim RecvStr as String
Dim Buff as String

IpAddress = "192.168.16.100"
SendStr = "*ESR?" & vbCrLf 'デリミタをLFに設定している場合
SendSize = Len(SendStr)
RecvStr = " " '想定される応答文が入るために十分なサイズを確保する
Size = Len(RecvStr)
Delim = &H0A 'デリミタをLFに設定している場合
ret = En_SendRecvStr(IpAddress, SendStr, SendSize, RecvStr, Size, Delim)
If ret < 0 Then
    エラー処理を記述する
Else
    Buff = Mid(RecvStr, 1, Size) 'Size には受信した応答文の真のサイズが入っている
End If
    
```

[3 - 3] ポート・ステータス・レジスタ

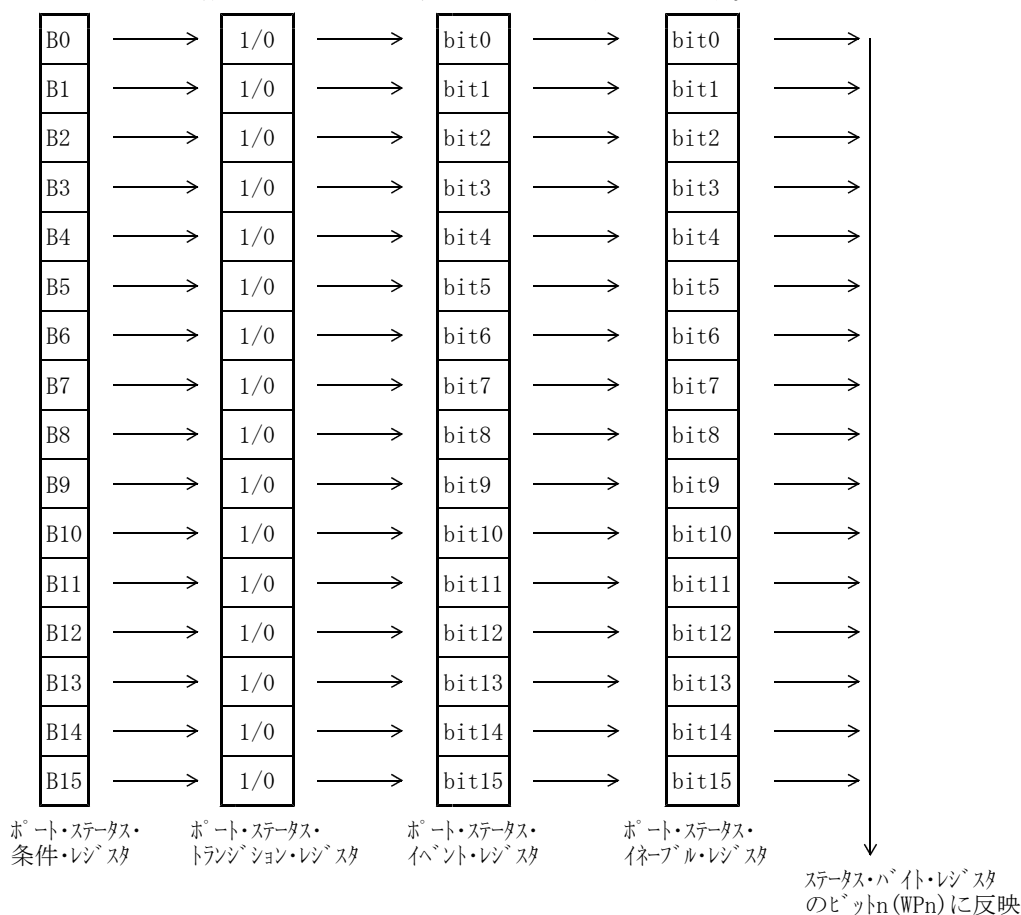
ポート・ステータス・レジスタ群はUIO-2144ENBの入出力ポート（0～4）の変化状況を補足する機能があります。
 五つのポートはWPORT0（ポート0、1）、WPORT1（ポート2、3）、WPORT2（ポート4）の三つのグループに分けて管理されます。

WPORT0と1に関するレジスタは16ビット、WPORT2は8ビットで構成されています。
 それぞれのグループに下記の管理用レジスタが存在します。

- (A) ポート・ステータス・条件・レジスタ : 入力ポートのコピーです。
 「:STATUS:WPORTn:CONDITION?」で内容を読み出せます。
- (B) ポート・ステータス・トランジション・レジスタ : イベントレジスタに記録する変化の種類（立ち上がり／立ち下がり）を選択する情報を設定します。
 「:STATUS:WPORTn:TRANSITION?」で内容を読み出せます。
 「:STATUS:WPORTn:TRANSITION 数値」で設定できます。
- (C) ポート・ステータス・イベント・レジスタ : トランジションレジスタで選択した変化があった事を記録します。
 「:STATUS:WPORTn:EVENT?」で内容を読み出せます。
- (D) ポート・ステータス・イネーブル・レジスタ : 発生したイベントにより ステータス・バイト・レジスタ のビットn(WPn)に反映するかどうかを設定します。

(以上のコマンドの説明は、本書 [4 - 3] をご参照ください)

各レジスタのビット構成およびレジスタの関係は下図のようになります。



[3 - 4] ステータス・レジスタの初期値

本機の電源を投入した場合、背面のディップスイッチでサーバーモード/対向モードの状態を変更した場合、ステータス報告システムの各レジスタの初期値は下記のように設定されます。

ステータス・バイト・レジスタ	bit7	MSS	ESB	MAV	WP2	WP1	WP0	bit0
	0	0	0	0	0	0	0	0
サービス・リクエスト・イネーブル・レジスタ	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	0	0	0	0	0	0	0	0
スタンダード・イベント・ステータス・レジスタ	PON	URQ	CME	EXE	DDE	QYE	RQC	OPC
	1	0	0	0	0	0	0	0
スタンダード・イベント・イネーブル・レジスタ	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	0	0	0	0	0	0	0	0

W P O R T 0 (ポート0、ポート1) のレジスタ群

ポート・ステータス・条件・レジスタ

B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ポート・ステータス・トランジション・レジスタ

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ポート・ステータス・イベント・レジスタ

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ポート・ステータス・イネーブル・レジスタ

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

W P O R T 1 (ポート2、ポート3) のレジスタ群

ポート・ステータス・条件・レジスタ

B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ポート・ステータス・トランジション・レジスタ

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ポート・ステータス・イベント・レジスタ

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ポート・ステータス・イネーブル・レジスタ

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

W P O R T 2 (ポート4) のレジスタ群

ポート・ステータス・条件・レジスタ

B7	B6	B5	B4	B3	B2	B1	B0
0	0	0	0	0	0	0	0

ポート・ステータス・トランジション・レジスタ

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	0	0	0	0	0	0	0

ポート・ステータス・イベント・レジスタ

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	0	0	0	0	0	0	0

ポート・ステータス・イネーブル・レジスタ

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	0	0	0	0	0	0	0

[4] SCMCコマンド for UIO-2144ENB

◎ コマンド

当SCMCコマンドはIEEE488.2-1992規格を基に階層構造になっています。
設定データのほとんどはクエリ（設定値の確認読み出し）する事ができます。

◎ 数値パラメータ

数値パラメータはASCII文字による10進表記を基本として、16進、8進、2進表記も使用できます。
10進表記では、符号、小数点、指数部付き表記を使用できますが、
16、18、2進表記では整数のみを使用します。
また、2進数の特別な扱いとして論理値（LON, LOFF）を使用することができます。

◎ ディスクリットパラメータ

数値では表現できない設定データ、または未知の数値データを表すパラメータです。
例えば、トリガ源として外部トリガ入力を指定（選択）する場合は、EXTERNAL
例えば、信号の立ち上がりを指定（選択）する場合は、POSITIVE
例えば、アンプのゲインを最大に取りたい場合は、MAX
の様に使います。

◎ ブロックパラメータ

大量のデータを送受するための特別なフォーマットです。
この中でも、データ個数があらかじめ特定できる場合と、できない場合があります。

☆ 確定長・データ・ストリング・フォーマット <DAS0>,<DAS1>,<DAS2>,< >,<DASm>

<DAS0>: 後に続くデータの個数を表します。
数値の表現は10進、2進、8進、16進のいずれも使用できます。
<DAS1>~<DASm>: データです。10進、2進、8進、16進のいずれの表現も使用できます。
各<DASm>は、で区切られています。

☆ 確定長・データ・バイナリ・フォーマット #nm<DAB1><DAB2>< ><DABm>

n: 1桁のASCII数値、データ・バイトのバイト数mの桁数を表します。
このnは、10進数で表現します。
m: n桁のASCII数値、データ・バイトのバイト数を表します。
この後に続く、<DAB1>から<DABm>までの個数をバイト単位で表します。
このmは、10進数で表現します。
<DAB1>~<DABm>: データのバイナリ・コードです。
各<DABm>はで区切られていません。

☆ 不確定長・データ・ストリング・フォーマット 0,<DAS1>,<DAS2>,< >,<DASm>

0: 不確定長ストリングを表す、ASCII文字です。
<DAS1>~<DASm>: データです。
数値の表現は10進、2進、8進、16進のいずれも使用できます。
各<DASm>は、で区切られています。

◎ デリミタ（ターミネータ）

すべてのコマンドメッセージはデリミタで終了させてください。
本機からの応答メッセージもすべてデリミタで終了します。（本書 [1-5] 参照）

[4 - 1] 入力ポートからの入力コマンド

INPUTコマンドセット

コマンド	パラメータ	備考	初期値
:INPut [:DATA]?	ビット名称 (BIT00～BIT47) バイト名称 (BYTE0～4) ワード名称 (WORD0～2)		
:FORMat	データ形式	データ形式の指定	DECIMAL
:FORMat?		データ形式の問い合わせ	
:IOMode?	データ形式	ポートの入出力設定状況の問い合わせ	

ビット名称 : BIT00～07, BIT10～17, BIT20～27, BIT30～37, BIT40～47

バイト名称 : BYTE0～4

ワード名称 : WORD0～2

データ形式 :

ASCII文字数値の2進数を指定する場合は、BINary と記述します。
 ASCII文字数値の8進数を指定する場合は、OCTal と記述します。
 ASCII文字数値の10進数を指定する場合は、DECimal と記述します。
 ASCII文字数値の16進数を指定する場合は、HEX と記述します。
 ASCII文字数値の論理を指定する場合は、LOGical と記述します。

「 4 - 1 - 1 」

書式 :INPUT[:DATA]? ビット名称
 :INPUT[:DATA]? バイト名称
 :INPUT[:DATA]? ワード名称

説明 ビット名称、バイト名称、ワード名称で指定する入力ポートの信号を入力し、応答メッセージを作成することを指示します。[]の部分は省略可能です。
 応答メッセージのフォーマットは「:INPUT:FORMAT データ形式」で指定されたフォーマットです。
 「:INPUT[:DATA] バイト名称」の場合で、「:INPUT:FORMAT」で「論理」を指定してあった場合、「BINARY」の表現で応答データを返送します。

応答 このコマンドの後、指定された入力ポートの信号を入力し、指定されたフォーマットで応答メッセージを返送します。

応答メッセージのフォーマット

不確定長・データ・ストリング・フォーマット 0,<DAS1>

0 : 不確定長ストリングを表す、ASCII文字です。
 <DAS1> : 指定されたデータ形式で表した数値のデータです。

☆「:INPUT:DATA? バイト名称」コマンドに対する応答の場合、

データの値は0～255の範囲です。
 指定データ形式が2進数の場合は、例えば#B11011となっています。
 10進数の場合は、例えば27となっています。
 16進数の場合は、例えば#H1Bとなっています。
 8進数の場合は、例えば#Q27となっています。
 論理の場合は、例えば#B11011となっています。

☆「:INPUT:DATA? ビット名称」コマンドに対する応答の場合、

データの値の範囲は0または1です。
 指定データ形式が2進数の場合は、#B0または#B1となっています。
 10進数の場合は、0または1となっています。
 16進数の場合は、#H0または#H1となっています。
 8進数の場合は、#Q0または#Q1となっています。
 論理の場合は、LOFFまたはLONとなっています。

例

:INPUT? BIT00① BIT00の状態を読みます。
 応答は「0,0①」または「0,1①」となります。

:INP? BYTE1① ポート1のBIT10～BIT17の全てを読みとります。
 応答は「0,0①」から「0,255①」の範囲の数値となります。

注 : この例で「①」はデリミタを意味します。

「4-1-2」

書式 :INPUT:FORMAT データ形式

説明 「:INPUT:DATA ビット名称/バイト名称/ワード名称」 コマンドに対する応答メッセージのフォーマットを指定します。

データ形式:

ASCII 文字数値の2進数を指定する場合は、BINary と記述します。
 ASCII 文字数値の8進数を指定する場合は、OCTal と記述します。
 ASCII 文字数値の10進数を指定する場合は、DECimal と記述します。
 ASCII 文字数値の16進数を指定する場合は、HEX と記述します。
 ASCII 文字数値の論理を指定する場合は、LOGical と記述します。

応答 このコマンドに対する応答はありません。

「IV-1-3」

書式 :INPUT:FORMAT?

説明 「:INPUT:DATA ビット名称/バイト名称/ワード名称」 コマンドに対する応答メッセージのデータ形式の指定状況を問い合わせます。

応答 このコマンドの後、下記のいずれかの応答メッセージを返送します。

BINARY
 OCTAL
 DECIMAL
 HEX
 LOGICAL

例 (Visual Basic 6)

```
Dim ret as Long
Dim IPAddress as String
Dim SendStr as String
Dim SendSize as Long
Dim Size as Long
Dim RecvStr as String

IPAddress = "192.168.16.100"
SendStr = ":INPUT:FORMAT?" & vbCrLf 'デリミタをLFに設定している場合
SendSize = Len(SendStr)
RecvStr = " " '想定される応答文が入るために十分なサイズを確保する
Size = Len(RecvStr)
Delim = &H0A 'デリミタをLFに設定している場合
ret = En_SendRecvStr(IPAddress, SendStr, SendSize, RecvStr, Size, Delim)
If ret < 0 Then
  エラー処理を記述する
Else
  Buff = Mid(RecvStr, 1, Size) 'Size には受信した応答文の真のサイズが入っている
End If
```

「IV-1-4」

書式 :INPUT:IOMODE? データ形式

説明 入出力ポートの設定状況を問い合わせます。「データ形式」は省略可能です。省略すると10進数で応答します。

データ形式:

ASCII 文字数値の2進数を指定する場合は、BINary と記述します。
 ASCII 文字数値の8進数を指定する場合は、OCTal と記述します。
 ASCII 文字数値の10進数を指定する場合は、DECimal と記述します。
 ASCII 文字数値の16進数を指定する場合は、HEX と記述します。

応答 このコマンドの後、指定された「データ形式」で設定状況を0~15の範囲の数値で返送します。数値は入出力選択信号線 (B0~B3) の個々がビット対応した形になっています。high が1、Low が0に相当します。例えば、全てのポートが入力に、論理が負論理に設定されている場合は「8」が返送されます。(取扱説明書の「[2-1] ポートの入出力設定」の項を参照して下さい)

[4 - 2] 出力ポートへの出力コマンド

OUTPUTコマンドセット

コマンド	パラメータ	備考
:OUTput	ビット名称 (BIT00~47), 出力データ バイト名称 (BYTE0~4), 出力データ ワード名称 (WORD0~2), 出力データ	
:OUTput?	ビット名称 (BIT00~47), データ形式 バイト名称 (BYTE0~4), データ形式 ワード名称 (WORD0~2), データ形式	

ビット名称 : BIT00~07, BIT10~17, BIT20~27, BIT30~37, BIT40~47

バイト名称 : BYTE0~4

ワード名称 : WORD0~2

データ形式 : 2進数を指定する場合は、BINary と記述します。
8進数を指定する場合は、OCTal と記述します。
10進数を指定する場合は、DECimal と記述します。
16進数を指定する場合は、HEX と記述します。
論理を指定する場合は、LOGical と記述します。

「4-2-1」

書式 :OUTPUT ビット名称,出力データ
 :OUTPUT バイト名称,出力データ
 :OUTPUT ワード名称,出力データ

説明 ビット名称、バイト名称、ワード名称で指定する出力ポートへ出力データを出力させます。

出力データ:

出力データの値は10進数、16進数、8進数、2進数のいずれかで表現したASCII文字で指定します。

基数ヘッダが付加されないと10進数とみなされます。

基数を2進数とする場合は、例えば#B101などと記述します

8進数とする場合は、例えば#Q107などと記述します。

10進数とする場合は、例えば245などと記述します。

16進数とする場合は、例えば#HE1と記述します。

出力先がビットの場合に限って、論理表現、LONまたはLOFFと記述してもかまいません。

データが整数でない場合は整数になるよう、四捨五入されます。

☆出力先がバイトの場合、四捨五入の結果のデータは0から255の範囲の正の値でなければなりません。範囲外はエラーになります。出力先には四捨五入した整数値が出力されます。

☆出力先がビットの場合、四捨五入の結果のデータは0または1の範囲の正の値でなければなりません。範囲外はエラーになります。出力先には四捨五入した整数値が出力されます。

☆出力先がワードの場合、四捨五入の結果のデータは0から65535の範囲の正の値でなければなりません。範囲外はエラーになります。出力先には四捨五入した整数値が出力されます。

応答 このコマンドに対する応答はありません。

例

:OUTPUT BIT00,1① BIT00をONにします。(:OUTPUT BIT00,LON①としても同じです)

:OUTPUT BYTE1,255① BIT10～BIT17の全てをONにします。

注: この例で「①」はデリミタを意味します。

例 (Visual Basic 6)

```
Dim ret as Long
```

```
Dim IPAddress as String
```

```
Dim SendStr as String
```

```
Dim Size as Long
```

```
IPAddress = "192.168.16.100"
```

```
SendStr = ":OUTPUT BIT00,1" & vbCrLf 'デリミタをLFに設定している場合
```

```
Size = Len(SendStr)
```

```
ret = En_SendStr(IPAddress, SendStr, Size)
```

```
If ret < 0 Then
```

```
    エラー処理を記述する
```

```
End If
```


「4-2-2」

書式 :OUTPUT? ビット名称[,データ形式]
 :OUTPUT? バイト名称[,データ形式]
 :OUTPUT? ワード名称[,データ形式]

説明 ビット名称、バイト名称、ワード名称で指定する出力ポートのデータを、データ形式で指定する表現で、応答メッセージを作成させます。
 []の部分は省略可能です。データ形式の指定を省略した場合は10進数とみなされます。

データ形式：

- 2進数を指定する場合は、BINary と記述します。
- 8進数を指定する場合は、OCTal と記述します。
- 10進数を指定する場合は、DECimal と記述します。
- 16進数を指定する場合は、HEX と記述します。
- 論理を指定する場合は、LOGical と記述します。（対象がビット名称の場合にのみ有効）

応答 このコマンドの後、指定された出力ポートへ出力しているデータを、指定されたデータ形式の数値で応答メッセージを返送します。

応答メッセージのフォーマットは下記のとおりです。

数値

数値は指定された基数ヘッダが付加されたASCII文字列のデータがひとつです。
 ただし、指定されたデータ形式が10進数の場合は基数ヘッダは省略され、それ以外の基数ヘッダは下記のようにになっています。

- 2進数の場合の基数ヘッダは、「#B」となっています。
- 16進数の場合の基数ヘッダは、「#H」となっています。
- 8進数の場合の基数ヘッダは、「#Q」となっています。

データ形式が論理の場合は、対象がビット名称の時のみ有効で、数値の代わりにLOFFまたはLONとなります。

例 (Visual Basic 6)

```
Dim ret as Long
Dim IPAddress as String
Dim SendStr as String
Dim SendSize as Long
Dim Size as Long
Dim RecvStr as String

IPAddress = "192.168.16.100"
SendStr = ":OUTPUT? BYTE1" & vbCrLf 'デリミタをLFに設定している場合
SendSize = Len(SendStr)
RecvStr = " " '想定される応答文が入るために十分なサイズを確保する
Size = Len(RecvStr)
Delim = &H0A 'デリミタをLFに設定している場合
ret = En_SendRecvStr(IPAddress, SendStr, SendSize, RecvStr, Size, Delim)
If ret < 0 Then
  エラー処理を記述する
Else
  Buff = Mid(RecvStr, 1, Size) 'Size には受信した応答文の真のサイズが入っている
End If
```

[4 - 3] ポート・ステータス操作コマンド

STATUSコマンドセット

コマンド	パラメータ	備考
:STATUS		
:WPort0		
:TRANSition	数値(0~65535)	イベント発生条件を設定する 0 = HighからLowへの変化で発生 1 = LowからHighへの変化で発生
:ENable	数値(0~65535)	イベント発生によるStatus・Byte・Registerへの反映を禁止/許可する 0 = 禁止、1 = 許可
:TRANSition?		イベント発生条件をクエリする
:EVEnt?		イベントの発生状況をクエリする
:ENable?		イベント発生によるStatus・Byte・Registerへの反映の禁止/許可をクエリする
:CONDition?		条件レジスタをクエリする
:WPort1		
:TRANSition	数値(0~65535)	イベント発生条件を設定する 0 = HighからLowへの変化で発生 1 = LowからHighへの変化で発生
:ENable	数値(0~65535)	イベント発生によるStatus・Byte・Registerへの反映を禁止/許可する 0 = 禁止、1 = 許可
:TRANSition?		イベント発生条件をクエリする
:EVEnt?		イベントの発生状況をクエリする
:ENable?		イベント発生によるStatus・Byte・Registerへの反映の禁止/許可をクエリする
:CONDition?		条件レジスタをクエリする
:WPort2		
:TRANSition	数値(0~255)	イベント発生条件を設定する 0 = HighからLowへの変化で発生 1 = LowからHighへの変化で発生
:ENable	数値(0~255)	イベント発生によるStatus・Byte・Registerへの反映を禁止/許可する 0 = 禁止、1 = 許可
:TRANSition?		イベント発生条件をクエリする
:EVEnt?		イベントの発生状況をクエリする
:ENable?		イベント発生によるStatus・Byte・Registerへの反映の禁止/許可をクエリする
:CONDition?		条件レジスタをクエリする

WPort0は、ポート0とポート1を、連続した16ビットとして扱います。
 WPort1は、ポート2とポート3を、連続した16ビットとして扱います。
 WPort2は、ポート4を、8ビットとして扱います。

「4-3-1」

書式 :STATUS:WPORT0:TRANSITION 数値
 :STATUS:WPORT1:TRANSITION 数値
 :STATUS:WPORT2:TRANSITION 数値

説明 入出力ポートの各ビットの変化によるイベント発生条件を設定します。
 設定値は、WPORT0 と WPORT1 の場合は0～65535の範囲の数値で行います。
 WPORT2 の場合は0～255の範囲の数値で行います。
 例えば、ポート0のビット0の Low から High の変化で、他は High から Low の変化でイベント発生とする場合の数値は、バイナリであれば 111111111111110 なので、65534を設定します。
 この数値はポート・ステータス・トランジション・レジスタに設定されます。

応答 このコマンドに対する応答はありません。

イネーブル・レジスタがON（1）に設定されている該当ビットのトランジション・レジスタの値によって、「High から Low の変化」または「Low から High の変化」を検出し、イベントを発生させます。イベントが発生するとイベント・レジスタの該当ビットがON（1）になります。各ポートの変化検出はソフトウェアでの監視により行っているため、どちらの変化も検出できませんが高速の信号変化（パルス幅500uSec以下）には対応できません。

例 (Visual Basic 6)

```
Dim ret as Long
Dim IpAddress as String
Dim SendStr as String
Dim Size as Long

IpAddress = "192.168.16.100"
SendStr = ":STATUS:WPORT0:TRANSITION 65534" & vbCrLf 'デリミタをLFに設定している場合
Size = Len(SendStr)
ret = En_SendStr(IpAddress, SendStr, Size)
If ret < 0 Then
  エラー処理を記述する
End If
```

「4-3-2」

書式 :STATUS:WPORT0:ENABLE 数値
 :STATUS:WPORT1:ENABLE 数値
 :STATUS:WPORT2:ENABLE 数値

説明 入出力ポートのビット変化によるイベント発生でステータス・バイト・レジスタの該当ビットをON（1）にするかどうかを設定します。

入出力ポート	ステータス・バイト・レジスタの該当ビット
WPORT0（ポート0、ポート1）	bit1: WP0
WPORT1（ポート2、ポート3）	bit2: WP1
WPORT2（ポート4）	bit3: WP2

設定値は、WPORT0 と WPORT1 の場合は0～65535の範囲の数値で行います。
 WPORT2 の場合は0～255の範囲の数値で行います。

例えば、ポート2のビット7のイベント発生でWP1ビットをONにする場合の数値は、128を設定します。
 この数値はポート・ステータス・イネーブル・レジスタに設定されます。

応答 このコマンドに対する応答はありません。

「4-3-3」

書式 :STATUS:WPORT0:TRANSITION?
 :STATUS:WPORT1:TRANSITION?
 :STATUS:WPORT2:TRANSITION?

説明 入出力ポートのビット変化によるイベント発生条件の設定内容を読み出します。

応答 このコマンドの後、応答メッセージとして、ポート・ステータス・トランジション・レジスタの内容を、下記のように10進整数値で返送します。
 数値は、WPORT0 と WPORT1 の場合は0～65535の範囲です。
 WPORT2 の場合は0～255の範囲です。

数値

「4-3-4」

書式 :STATUS:WPORT0:EVENT?
 :STATUS:WPORT1:EVENT?
 :STATUS:WPORT2:EVENT?

説明 入出力ポートのビット変化によるイベント発生条件によるイベントの発生状況を読み出します。
 読み出されたポート・ステータス・イベント・レジスタはクリアされます。

応答 このコマンドの後、応答メッセージとして、ポート・ステータス・イベント・レジスタの内容を、
 下記のように10進整数値で返送します。

数値

「4-3-5」

書式 :STATUS:WPORT0:ENABLE?
 :STATUS:WPORT1:ENABLE?
 :STATUS:WPORT2:ENABLE?

説明 入出力ポートのビット変化によるイベント発生条件によるイベント発生での、ステータス・バイト・
 レジスタの該当ビット (WP n) への反映許可/不許可設定内容を読み出します。

応答 このコマンドの後、応答メッセージとして、ポート・ステータス・イネーブル・レジスタの内容を、
 下記のように10進整数値で返送します。

数値

「4-3-6」

書式 :STATUS:WPORT0:CONDITION?
 :STATUS:WPORT1:CONDITION?
 :STATUS:WPORT2:CONDITION?

説明 入出力ポートの現在の状況を読み出します。

応答 このコマンドの後、応答メッセージとして、ポート・ステータス・条件・レジスタの内容を、
 下記のように10進整数値で返送します。

数値

例 (Visual Basic 6)

```
Dim ret as Long
Dim IpAddress as String
Dim SendStr as String
Dim SendSize as Long
Dim Size as Long
Dim RecvStr as String

IpAddress = "192.168.16.100"
SendStr = ":STATUS:WPORT1:CONDITION?" & vbCrLf 'デリミタをLFに設定している場合
SendSize = Len(SendStr)
RecvStr = " " '想定される応答文が入るために十分なサイズを確保する
Size = Len(RecvStr)
Delim = &H0A 'デリミタをLFに設定している場合
ret = En_SendRecvStr(IpAddress, SendStr, SendSize, RecvStr, Size, Delim)
If ret < 0 Then
  エラー処理を記述する
Else
  Buff = Mid(RecvStr, 1, Size) 'Size には受信した応答文の真のサイズが入っている
End If
```

[5] イーサネットアクセスDLLの使用法

[5-1] 概要

本機をパソコンからアクセスするためのDLLが付属しています。このDLLはユーザーがイーサネットを簡便に使用するための補助ライブラリです。

[5-2] 動作環境

OS : Windows 2000 / XP
イーサネットアダプタ : 上記OSから制御できること

[5-3] 関数

以下にDLLに含まれる関数の機能等について記載します。

変数のサイズは

CHAR : 符号有り 8ビット整数
Short : 符号有り16ビット整数
INT32 : 符号有り32ビット整数
となっています。

[1] DLLの初期化

機能 : DLLの初期化をおこないます。

書式 : (C言語) : int32 En_Open(Void)
(Visual Basic 6) : Declare Function En_Open Lib "21xxEN.dll" () As Long

引数 : 入力 : なし
戻値 : エラーコード

[2] DLLの終了

機能 : DLLの終了処理をおこないます。

書式 : (C言語) : int32 En_Close(Void)
(Visual Basic 6) : Declare Function En_Close Lib "21xxEN.dll" () As Long

引数 : 入力 : なし
戻値 : エラーコード

[3] DLLのバージョン取得

機能 : DLLのバージョン文字列を指定 (Size) 分だけバッファにコピーする。

書式 : (C言語) : int32 En_Get_Dll_Version(char *Ver, int32 *Size)
(Visual Basic 6) : Declare Function En_Get_Dll_Version Lib "21xxEN.dll" (ByVal Vers As _
String, ByRef Size As Long) As Long

引数 : 入力 : Ver : DLLのバージョン文字列を格納するバッファのポインタ
Size : DLLのバージョン文字列を格納するバッファのサイズ
(リターン時は格納した文字数が入る)

戻値 : エラーコード

[4] IPアドレス調査

機能 : 指定されたMACアドレスに割り当てられているIPアドレスを取得する。

書式 : (C言語) : int32 En_Get_IPAddress(char *MacAddress, char *IpAddress, int32 *Size)
(Visual Basic 6) : Declare Function En_Get_IPAddress Lib "21xxEN.dll" (ByVal MacAddress _
As String, ByVal IpAddress As String, ByRef Size As Long) As Long

引数 : 入力 : MacAddress : MACアドレス文字列を格納するバッファのポインタ
IpAddress : IPアドレス文字列を格納するバッファのポインタ
Size : IPアドレス文字列を格納するバッファのサイズ
(リターン時は格納した文字数が入る)
見つからなかった場合やエラーが起こった場合はNULLポインタが入る。

戻値 : エラーコード

[6] 問い合わせコマンドの送信とその応答の受信

機能：応答のある、問い合わせコマンドを送信し、それに対する応答を受信する。
 書式：（C言語） : int32 En_SendRecvStr(char *IpAddress, char *SendStr, int32 SendSize, char *RecvStr, int32 *Size, char Delim)
 (Visual Basic 6) : Declare En_SendRecvStr Lib "21xxEN.dll" (ByVal IpAddress As String, _
 ByVal SendStr As String, ByVal SendSize As Long, _
 ByVal RecvStr As String, ByRef Size As Long, ByVal Delim As Byte _
) As Long
 引数：入力：IpAddress：端末のIPアドレス文字列を格納するバッファのポインタ
 SendStr : 送信する文字列を格納するバッファのポインタ
 SendSize : 送信する文字列のサイズ（文字列の長さ）
 RecvStr : 受信する応答文字列を格納するバッファのポインタ
 Size : 受信する応答文字列を格納するバッファの大きさ（最大は1,000,000）
 (リターン時は受信した文字列のサイズが入る)
 Delim : 受信する応答文字列のデリミタコード
 戻値：エラーコード

[7] 設定コマンドの送信

機能：応答のない、設定コマンドを送信する。
 書式：（C言語） : int32 En_SendStr(char *IpAddress, char *SendStr, int32 Size)
 (Visual Basic 6) : Declare En_SendStr Lib "21xxEN.dll" (ByVal IpAddress As String, _
 ByVal SendStr As String, ByVal Size As Long) As Long
 引数：入力：IpAddress：端末のIPアドレス文字列を格納するバッファのポインタ
 SendStr : 送信する文字列を格納するバッファのポインタ
 Size : 送信する文字列のサイズ（文字列の長さ）
 戻値：エラーコード

[5-4] エラーコード

戻値/ エラーコード	エラー内容	対処例
0	エラーなし、正常終了	
-10	システム異常 他のアプリがソケットを上限まで使った上でDLLを呼んだ場合などに起こりうる。	ネットワークリソースを多用している他のアプリケーションを終了させるなどの処理を行った後、再度、試してみる。
-11	システム異常 Windows 95 OSR2 以前の古いバージョンのWindowsで動かした時に発生する場合がある。	Windows 9xでの動作を保証できませんのでWindows 2000, Windows XPなどを使って下さい。
-12	システム異常 ネットワークアダプタに関する情報を取得するWSocket関数が失敗した時に発生する。	ネットワークリソースを多用している他のアプリケーションを終了させるなどの処理を行った後、再度、試してみる。
-13	システム異常 ネットワークアダプタに関する情報を取得するWSAIoct1関数が失敗した時に発生する。	
-14	システム異常 ネットワークが使用できない	ネットワークが正しく使用できる環境にしてから再度試してみる。
-15	システム異常 ネットワークインターフェースの数が一つでない場合に発生。（たとえばLANカードが2枚入っている、イーサネット+ダイアルアップ接続している、VPN接続を行っている等）	複数のネットワークインターフェースが存在しても稼働中でなければこのエラーは発生しないので、例えば、使用しないLANカードからLANケーブル抜く、ダイアルアップを切断するなどの処置を行う。
-100	ステート異常（入力異常） Lx_open前、En_open失敗時、En_close後にその他の関数を呼び出した場合に発生する。（但し、En_Get_Dll_Versionを除く）	関数使用前にEn_Openを呼び出してください。
-101	通信異常 指定したIPアドレスのデバイスに接続できない。	本機のMACアドレス・IPアドレスを、又は、PCと本機間のLANケーブル・ハブの状況を確認して下さい。
-102	システム異常	再起動後に試してみる。または他のPCで試してみる。
-103	システム異常	再起動後に試してみる。または他のPCで試してみる。
-104	システム異常	再起動後に試してみる。または他のPCで試してみる。

-106	通信異常 Winsock関数呼び出し後、SOCKET_ERROR が発生した場合のエラー（指定IPアドレスのデバイスに接続できない場合など）	PCから本機までのケーブルルートに不具合が無いか、確認をする。
-107	システム異常	再起動後に試してみる。または他のPCで試してみる。
-108	指定IPアドレスへの接続準備に失敗した。	
-200	指定IPアドレスへの送信準備に失敗した。	
-201	通信異常 Winsock関数呼び出し後、SOCKET_ERROR が発生した場合のエラー。他のユーザがアクセス中のデバイスにアクセスした時に発生する。	他のアプリケーションからのアクセスがない環境で再度試す。 本機の電源を一度 ON/OFF してから試す。
-300	DLL内部の問題	発生状況のご連絡をお願いします。
-301	通信異常 ソフトのバグ、本機側ハードウェア異常、本機以外のデバイスと通信しようとした。	指定のMACアドレスが確かに本機のものか確認して下さい。本機の電源がONになっているか、使用電源が仕様に合っているか、確認して下さい。
-302	送信準備に失敗。受信準備に失敗。	
-303	通信異常 Winsock関数呼び出し後、SOCKET_ERROR が発生した場合のエラー（指定IPアドレスからの受信ができなかった場合）	
-304	指定IPアドレスからの受信に失敗した。	
-400	送信や受信の最終処理に失敗した。	
-500	MACアドレスからIPアドレスを探す場合の準備に失敗した。	
-501	システム異常。 ソケットが生成できない。	再起動後に試してみる。または他のPCで試してみる。
-508	MACアドレスからIPアドレスを探す場合の終了処理に失敗した。	
-507	通信異常 Winsock 関数呼び出し後、SOCKET_ERRORが発生した場合のエラー	
-509	通信異常 Winsock 関数呼び出し後、SOCKET_ERRORが発生した場合のエラー	
-510	MACアドレスからIPアドレスを探す場合に失敗した。	
-511	MACアドレスからIPアドレスを探す場合に失敗した。	
-512	DLL内部の問題	発生状況のご連絡をお願いします。
-513	DLL内部の問題	発生状況のご連絡をお願いします。
-1005	入力異常 入力変数が範囲外だった場合や不正な場合にこのエラーが発生する。	入力変数の数値範囲やフォーマットを確認して下さい。MACアドレスの場合、英文字は大文字を使用して下さい。IPアドレスを受け取る文字バッファは16文字分以上必要です。
-1007	入力異常 入力されたIPアドレスが、自分のIP到達範囲内がない場合に発生する。	設定するPCから接続可能なIPアドレスを指定してください。 PCのネットマスク・IPアドレスと本機のIPアドレスの関係を確認して下さい。
-1008	指定したMACアドレスを持つデバイスが見つからなかった。	デバイスのMACアドレスやIPアドレスを確認して下さい。PCとデバイス間のLANケーブルやハブの状況を確認して下さい。