

イーサネット I/F リレーユニット

RLT-2116ENCC
RLT-2132ENCC 2
RLT-2132ENCC
RLT-2116ENCC
RLT-2132ENCC

コマンド説明書

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789-9
TEL 042-705-8312 FAX 042-794-8317



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

目次

【Ⅰ】 概要	
[Ⅰ-1] 概略動作	2
[Ⅰ-2] フォーマット	2
[Ⅰ-3] コマンド	2
[Ⅰ-4] パラメータ	2
[Ⅰ-5] デリミタ (ターミネータ)	3
[Ⅰ-6] エラー処理	3
【Ⅱ】 共通コマンド	
[Ⅱ-1] システムデータ・コマンド	4
[Ⅱ-2] 内部操作・コマンド	4
[Ⅱ-3] 同期・コマンド	5
[Ⅱ-4] ステータス/イベント・コマンド	6
[Ⅱ-5] デバイストリガ・コマンド	7
【Ⅲ】 ステータス報告システム	
[Ⅲ-1] ステータス・ビット・レジスタ	8
[Ⅲ-2] スタンダード・イベント・ステータス・レジスタ	9
[Ⅲ-3] ステータス・レジスタの初期値	10
【Ⅳ】 S C M C コマンド	
[Ⅳ-1] 出力端への出力コマンド	12
[Ⅳ-2] バッファメモリ・コマンド	16
[Ⅳ-3] バッファリングされたデータの出力・コマンド	20
[Ⅳ-4] アボート・コマンド	25
【Ⅴ】 イーサネットアクセスDLLの使用方法	
[Ⅴ-1] 概要	26
[Ⅴ-2] 動作環境	26
[Ⅴ-3] 関数	26
[Ⅴ-4] エラーコード	27

改版履歴	改版日付	改 版 内 容
第1版	2009年06月26日	
第2版	2010年06月24日	
第3版	2010年07月30日	
第4版	2011年09月05日	
第5版	2014年06月09日	R L T - 2 1 3 2 E N C 2 について記載。

【I】概要

本「コマンド説明書」ではRLT-2116ENCやRLT-2132ENC2やRLT-2132ENCを使用する場合の操作コマンドについて説明します。ハード的な仕様については、「取扱説明書」を参照して下さい。

RLT-2116ENCやRLT-2132ENC2やRLT-2132ENCの操作コマンドやその機能は、IEEE-Std488.2-1992に準拠すべく構成、構築されています。(488.2規格は488.1規格の上に成り立っています)

なお、本書ではRLT-2116ENCやRLT-2132ENC2やRLT-2132ENCのことを「本機」または「リレーユニット」と記述しています。

[I-1] 概略動作

本機はイーサネット端末のリレーユニット機器です。本機を使用する場合は、通常はパソコンなどからコマンドを送信して制御します。

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、本機のリレーがON/OFFされます。また、本機の端末側に接続された回路の応答スピードにパソコンの処理速度が影響しないよう、本機内のメモリを使用した、バッファリング機能を利用することができます。(本書 [IV-2] [IV-3] をご参照ください)

488.2規格の共通コマンドを使用すると、本機のソフトウェアリセットや内部状態(ステータス)の読み取りなど、きめ細かな状況把握が可能です。(本書【II】【III】をご参照ください)

以上のような操作がすべてASCII文字列のやりとりで行われ、大変操作しやすくなっています。一般的に、大量のデータのやりとりにASCII文字を使用するとデータ転送の時間が大きくなりますが、それを補うべく、バイナリデータ転送方式もサポートしたコマンドも装備した、大変高機能なモードになっています。

[I-2] フォーマット

コントローラ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。

本機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 [I-5] を参照)

[I-3] コマンド

本機のコマンド体系は各種の計測機器において多く採用され、スタンダードとなっているIEEE488.2体系をベースにしています。(IEEE488.2の採用ではSCPIなどが有名です) 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(本書【IV】)を使うことができます。

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。

[I-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。

10進数値のフォーマットは	数値	(数値は 0, 1, 2, ..., 9 の組み合わせで基数ヘッダがありません)
16進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。

[I - 5] デリミタ (ターミネータ)

本機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 [II - 1] 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)

本機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。

[I - 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 b i t 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 b i t 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 b i t 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。)

【Ⅱ】 共通コマンド

[Ⅱ-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
 <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, RLT-2116EN, 000000, REV1.00	(R L T - 2 1 1 6 E N C の場合)
MCI-ENG, RLT-2132EN, 000000, REV1.00	(R L T - 2 1 3 2 E N C 2 の場合)
MCI-ENG, RLT-2132EN, 000000, REV1.00	(R L T - 2 1 3 2 E N C の場合)

(REV1.00 は改良により変更されることがあります)

例 (Visual Basic 6)

```

Dim ret as Long
Dim IpAddress as String
Dim SendStr as String
Dim SendSize as Long
Dim RecvStr as String * 100 ' 応答データが入るために十分なサイズ
Dim Size as Long
Dim Delim as Byte
Dim Buff as String ' 応答データを格納するバッファ

IpAddress = "192.168.16.100"
SendStr = "*IDN?" & vbCrLf ' デリミタを LF に設定している場合
SendSize = Len(SendStr)
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 には受信した応答データの真のサイズが入っている
End If

```

[Ⅱ-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。

- * 全てのリレーをOFF (復旧) にする
リレーの接点はオープン状態になる (取扱説明書 [Ⅲ-2] を参照)
- * ホストからの受信バッファをクリアする
- * PLAYコマンドシステム、MEMORYコマンドシステムを初期状態にする
- * 前に受け取った*OPCまたは*OPC?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。

- * 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
- * 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。
テストの実行を行った場合はMEMORYコマンドシステム、PLAYコマンドシステムは初期化されます。
初期化の結果、メモリに書き込まれていたデータは破棄されます。
リレーの動作状態、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。)

[II - 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?

[II-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
外部・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”255”までの値を10進数または16、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”255”の範囲の10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”255”の範囲の10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”255”までの値を10進または16、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”63”、“128”から”191”の範囲の10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”255”の範囲の10進数整数値です。

[II - 5] デバイストリガ・コマンド

□*TRG トリガ (Trigger)

書式 *TRG

説明 プレイ動作を起動させます。(本書 [IV-3] をご参照ください。)

応答 このコマンドに対する応答はありません。

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 = "*TRG" & 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
```


【Ⅲ】ステータス報告システム

[Ⅲ-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 : : 本機においては常に 0 です。

bit 2 : : 本機においては常に 0 です。

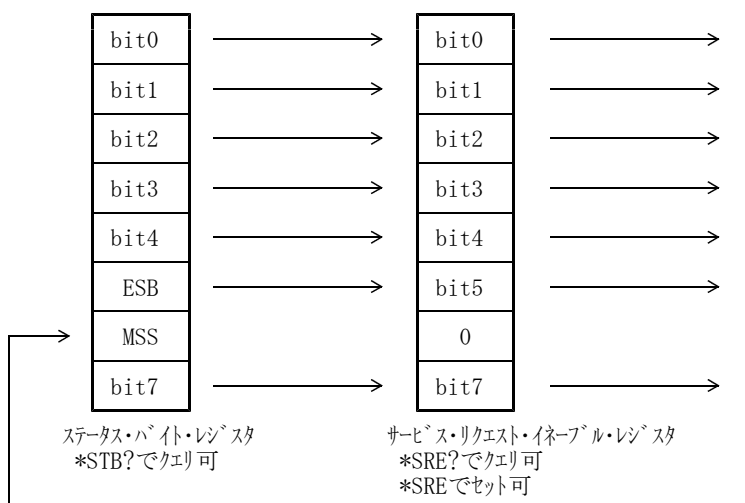
bit 3 : : 本機においては常に 0 です。

bit 4 : : 本機においては常に 0 です。

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。

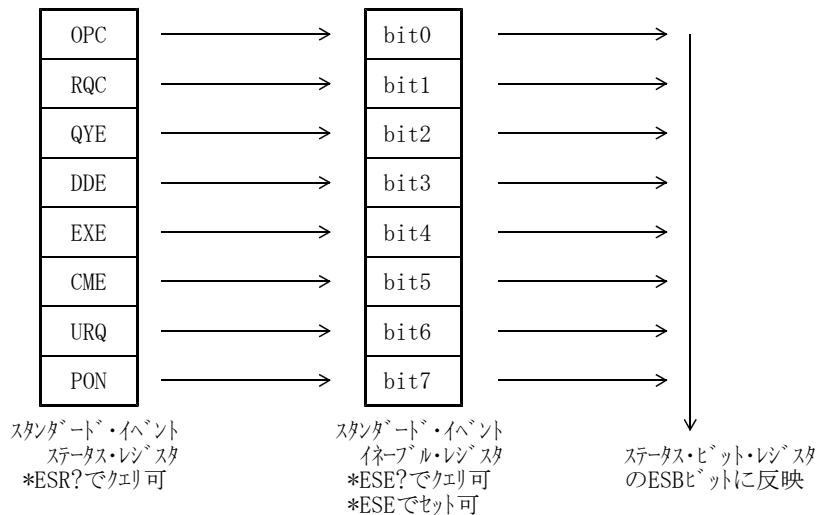
bit 6 : MSS : マスター・ステータス・サマリ
ステータス・ビット・レジスタの他の 7 ビットの代表。

bit 7 : : 本機においては常に 0 です。



[III-2] スタンド・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。



[Ⅲ-3] 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	RQS/MSS	bit5	bit4	bit3	bit2	bit1	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

【IV】SCMCコマンド

◎ コマンド

当SCMCコマンドはI-EEE488.2-1992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。

◎ 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 [I-5] 参照）

[IV-1] 出力端への出力コマンド

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 (BITn), 出力データ バイト名称 (BYTEn), 出力データ ワード名称 (WORDn), 出力データ	
:OUTput?	ビット名称 (BITn), データ形式 バイト名称 (BYTEn), データ形式 ワード名称 (WORDn), データ形式	

ビット名称 (BITn) : n は 0 ~ 31 までの数字で、本機の正面パネルの LDn で示す端子台およびモニタ LED に対応しています。(対応表を下記に示します)
 なお、ビット名称として BITn の代わりに LDn を使用することができます。

BITn	BIT0	BIT1	BIT2	BIT3	BIT4	BIT5	BIT6	BIT7
LDn	LD11	LD12	LD13	LD14	LD15	LD16	LD17	LD18

BITn	BIT8	BIT9	BIT10	BIT11	BIT12	BIT13	BIT14	BIT15
LDn	LD21	LD22	LD23	LD24	LD25	LD26	LD27	LD28

BITn	BIT16	BIT17	BIT18	BIT19	BIT20	BIT21	BIT22	BIT23
LDn	LD31	LD32	LD33	LD34	LD35	LD36	LD37	LD38

BITn	BIT24	BIT25	BIT26	BIT27	BIT28	BIT29	BIT30	BIT31
LDn	LD41	LD42	LD43	LD44	LD45	LD46	LD47	LD48

バイト名称 (BYTEn) : n は 0 ~ 3 までの数字で、
 BYTE0 は BIT0~BIT7 (LD11~LD18) の 8 ビットを総称する名称です。
 BYTE1 は BIT8~BIT15 (LD21~LD28) の 8 ビットを総称する名称です。
 BYTE2 は BIT16~BIT23 (LD31~LD38) の 8 ビットを総称する名称です。
 BYTE3 は BIT24~BIT31 (LD41~LD48) の 8 ビットを総称する名称です。

ワード名称 (WORDn) : n は 0 ~ 1 までの数字で、
 WORD0 は BIT0~BIT15 (LD11~LD18, LD21~LD28) の 16 ビットを総称する名称です。
 WORD1 は BIT16~BIT31 (LD31~LD38, LD41~LD48) の 16 ビットを総称する名称です。

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。

注 : R L T - 2 1 1 6 E N C にはリレーおよびリレーモニタは 16 個が実装されています。
 コマンドでそれを越える指定をしてもエラーにはなりません。

「IV-1-1」

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 ビット名称またはバイト名称,ワード名称で指定する出力端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。

出力先がビットの場合は、データの値は0～1の範囲でなければなりません

出力先がバイトの場合は、データの値は0～255の範囲でなければなりません。

出力先がワードの場合は、データの値は0～65535の範囲でなければなりません。

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。

本機は、このコマンドを受信すると、受信データの値を指定出力端に出力することにより、リレーを動作/復旧させます。

☆出力先がバイトの場合、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。

その結果のデータは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、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。

その結果のデータは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、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。

その結果のデータは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 BIT0,1① BIT0のリレーをONにします。(:OUTPUT LD11,1①としても同じです)
 LD11のモニタLEDが点灯し、端子台LD11のXとYが導通状態になります。

:OUTPUT BYTE1,255① BIT8～BIT15の全てのリレーをONにします。
 LD21～28のモニタLEDが点灯し、端子台LD21～28のXとYが導通状態になります。

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 BIT0,1" & VbLf '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

「IV-1-2」

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 ビット名称またはバイト名称, ワード名称で指定するリレーの動作状態をモニタし、基数で指定する表現で、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:

モニタする出力端がビットの場合、
 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。

モニタする出力端がバイトまたはワードの場合、
 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。

応答 このコマンドを受信した後、指定されたリレーの動作状態をモニタし、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。

モニタ先がビットの場合、データは0か1の正の整数値です。
 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。

モニタ先がバイトの場合、データは0から255の範囲の正の整数値です。
 データ形式が2進数の場合は、例えば#B1000001となっています。
 10進数の場合は、例えば65となっています。
 16進数の場合は、例えば#H41となっています。
 8進数の場合は、例えば#Q141となっています。

モニタ先がワードの場合、データは0から65535の範囲の正の整数値です。
 データ形式が2進数の場合は、例えば#B1000001となっています。
 10進数の場合は、例えば165となっています。
 16進数の場合は、例えば#H2A1となっています。
 8進数の場合は、例えば#Q741となっています。

例

:OUTPUT? BIT0① BIT0のリレーの状態を問い合わせます。(:OUTPUT? LD11① と同じです)
 その後、本機は状態を返送します。
 1① LD11のリレーがONしていると1が返ります。OFFの場合は0です。

:OUTPUT? BYTE1① BIT8～BIT15の全てのリレーの状態を問い合わせます。
 その後、本機は状態を返送します。
 255① LD21～28のリレーが全てONしていると255が返ります。

:OUT? BYTE2, HEX① BIT16～BIT23の全てのリレーの状態を問い合わせます。
 (コマンド「OUTput」の小文字部分は省略することができます)
 その後、本機は状態を返送します。
 #HFF① LD31～38のリレーが全てONしているとFF(255の16進数)が返ります。

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 SendSize as Long
Dim RecvStr as String * 100      ' 応答データが入るために十分なサイズ
Dim Size as Long
Dim Delim as Byte
Dim Buff as String              ' 応答データを格納するバッファ

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)
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 には受信した応答データの真のサイズが入っている
    'LD 2 1 ~ 2 8 のリレーが全てONしていると Buff には 255 が入ります。
End If
```


[IV-2] バッファメモリ・コマンド

MEMORYコマンドセット

コマンド	パラメータ	備考	初期値
:MEMory :ASSign :ASSign?	ブロック番号(0~1),ワード数 ブロック番号(0~1)	メモリ領域容量を指定確保する メモリ領域の情報の問い合わせ 領域容量, 使用容量, 空容量を得る	確保されていない
:WRITe :INITialize [:NEXT]	ブロック番号(0~1) ブロック番号(0~1),データ列	指定領域の書込ポインタを初期化 書込ポインタから書込み、 書込ポインタを次へ移す。	
:READ :INITialize [:NEXT]?	ブロック番号(0~1) ブロック番号(0~1),ワード数	指定領域の読出ポインタを初期化 読出ポインタから読出し、 読出ポインタを次へ移す。	
:FORMat :FORMat? :MEMory?	ブロック番号(0~1),データ形式 ブロック番号(0~1)	読出データ形式を指定する 読出データ形式の問い合わせ メモリの情報の問い合わせ 総領域容量, 総空容量を得る	DECIMAL

プレイ動作が、STANDBY状態にある場合は同一ブロック番号のメモリ領域に対して以下のことを行うことができません。

:MEMORY:ASSIGN ブロック番号,ワード数

プレイ動作が、RUNNING状態にある場合は同一ブロック番号のメモリ領域に対して以下のことを行うことができません。

:MEMORY:ASSIGN ブロック番号,ワード数
:MEMORY:WRITE:INITIALIZE ブロック番号
:MEMORY:WRITE:NEXT ブロック番号,データ列
:MEMORY:READ:INITIALIZE ブロック番号
:MEMORY:READ:NEXT? ブロック番号,ワード数

ブロック番号: 0, 1

確保した領域はプレイ動作で使用します。

[IV-2-1]

書式 :MEMORY:ASSIGN ブロック番号,ワード数

説明 ブロック番号で指定する領域の容量をワード単位の数で指定確保します。
読出ポインタと書込ポインタは、この領域の先頭に初期化されます。

ワード数: 0または、1以上、メモリの総空容量以内
ブロック番号で指定する領域のメモリ領域容量をワード単位で指定します。
0を指定した場合は、指定ブロック番号の領域を解放します。

領域の容量を変えたい場合は、一度、「:MEMORY:ASSIGN ブロック番号,0」で領域を開放してから新たな容量で確保します。この時、この領域のデータは消失します。また、「:PLAY:ASSIGN」で割り当てていた同じブロック番号のメモリ領域の割り当ても解放されます。
確保可能なメモリの総空容量は「:MEMORY?»コマンドで調べることができます。

応答 このコマンドに対する応答はありません。

プレイ動作が、STANDBY状態やRUNNING状態にある場合は同一ブロック番号のメモリ領域に対してのこのコマンドを受信すると「実行エラー」になります。

このコマンドで指定するブロック番号でメモリの領域がすでに定義確保されている場合は、データ数が0なら領域の解放を行います。0でない場合は「実行エラー」になります。

例

:MEMORY:ASSIGN 0,100① ブロック0のメモリサイズを100ワード確保します。

:MEM:ASS 1,#HFF② ブロック1のメモリサイズを255ワード確保します。
(コマンドの小文字部分は省略することができます)

注: この例で「①」はデリミタを意味します。

「IV-2-2」

書式 :MEMORY:ASSIGN? ブロック番号

説明 ブロック番号で指定する領域の情報を問い合わせます。

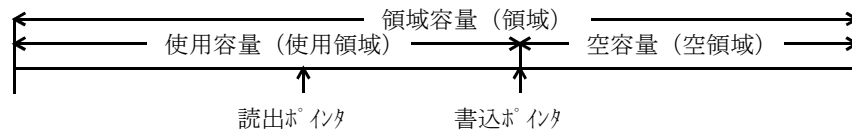
応答 このコマンド受信後、下記の応答メッセージを返送します。

領域容量, 使用容量, 空容量

領域容量: 「:MEMORY:ASSIGN ブロック番号,ワード数」で確保されている領域の容量をワード単位で表しています。

使用容量: 「:MEMORY:WRITE:NEXT ブロック番号,データ列」で書き込まれたデータの数をワード単位で表しています。

空容量: 領域容量から使用容量を差し引いた数をワード単位で表しています。



「IV-2-3」

書式 :MEMORY:WRITE:INITIALIZE ブロック番号

説明 ブロック番号で指定する領域への書込ポインタを初期化します。また、今までに書かれていたデータがあればこれを破棄し、読出ポインタも初期化します。

応答 このコマンドに対する応答はありません。プレイ動作が、RUNNING状態にある場合は、同一ブロック番号のメモリ領域に対してこのコマンドを受信すると「実行エラー」になります。

「IV-2-4」

書式 :MEMORY:WRITE:NEXT ブロック番号,データ列

説明 ブロック番号で指定する領域へデータを連続的に書き込みます。この動作の後、書込ポインタは最終書込データの格納された次をポイントします。読出ポインタは変化しません。データ列のフォーマットは下記のどちらの場合でも使用できます。

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

<DAS0>: 書込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>: 書き込むべきデータです。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、で区切ってください。

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、偶数でなければなりません。基数の場合はエラーになります。<DABm>: 端末側へ出力させるデータで、バイナリコードで入れて下さい。

後でプレイ出力しようとする出力先がビットの場合は、(0x0000) か (0x0001) のワードデータを、後でプレイ出力しようとする出力先がバイトの場合は、(0x0000) ~ (0x00FF) の範囲のワードデータを、後でプレイ出力しようとする出力先がワードの場合は、(0x0000) ~ (0xFFFF) の範囲のワードデータを上位バイト、下位バイトに分けて、上位バイトを先に入れて下さい。

例: (0x0034) と (0x5678) の2ワードを書き込む場合は下記のようになります。
#14<0x00><0x34><0x56><0x78>

どちらのフォーマットの場合でも、確保されたメモリ領域の領域容量を越えたら途中までで強制的に終了します。

応答 このコマンドに対する応答はありません。プレイ動作が、RUNNING状態にある場合は、同一ブロック番号のメモリ領域に対してこのコマンドを受信すると「実行エラー」になります。

「IV-2-5」

書式 :MEMORY:READ:INITIALIZE ブロック番号

説明 ブロック番号で指定する領域からの読出ポインタを初期化します。
書込ポインタは変化しません。

応答 このコマンドに対する応答はありません。
プレイ動作が、RUNNING状態にある場合は、同一ブロック番号のメモリ領域に対してのこのコマンドを受信すると「実行エラー」になります。

「IV-2-6」

書式 :MEMORY:READ:NEXT? ブロック番号,ワード数

説明 ブロック番号で指定する領域からデータを連続的に読み出します。
この動作の後、読出ポインタは最後に読み出したデータの格納されていた次をポイントします。
書込ポインタは変化しません。

ワード数: 0 または、1 ~ 1000000

ブロック番号で指定する領域から読み出したいデータの数をワード単位で指定します。
指定したブロック番号の領域に存在する未読み出しデータよりおおきな数を指定してもエラーにはならず、未読み出しデータ全部を正常に読み出す事ができます。
0を指定した場合は、指定したブロック番号の領域の残りデータ全部を読み出す事になります。

応答 このコマンドを受信後、「:MEMORY:READ:FORMAT ブロック番号,データ形式」で指定されているフォーマットに従って下記のいずれかで返送します。
読み出すべきデータが無い場合はデータの個数を0として返送します。
また、指定されたメモリ領域が「:MEMORY:ASSIGN」コマンドで定義確保されていない場合も同様です。

データ形式を BINARY、OCTAL、DECIMAL、HEX と指定してある場合は以下のようにになります。
確定長・データ・ストリング・フォーマット <DAS0>,<DAS1>,<DAS2>,< >,<DASm>

<DAS0>:読み出すデータの個数を表します。数値の表現は10進整数です。
「:MEMORY:READ:NEXT ブロック番号,ワード数」で指定したワード数、
またはメモリ領域に入っていたデータの数のどちらか小さい方です。
<DAS1>~<DASm>:読み出したデータです。10進整数で表現しています。
各<DASm>は、で区切られています。

データ形式を CODE と指定してある場合は以下のようにになります。
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。
「:MEMORY:READ:NEXT ブロック番号,ワード数」で指定したワード数、
またはメモリ領域に入っていたデータの数のどちらか小さい方の2倍です。
このmは、10進数で表現します。
<DAB1>~<DABm>:各<DABm>は、で区切られていません。
ワード単位のデータがバイト単位に分割され、上位バイトが先に、
下位バイトが後に入っています。

プレイ動作が、RUNNING状態にある場合は同一ブロック番号のメモリ領域に対してのこのコマンドを受信すると「実行エラー」になります。

「IV-2-7」

書式 :MEMORY:READ:FORMAT ブロック番号,データ形式

説明 「:MEMORY:READ:NEXT? ブロック番号,ワード数」コマンドに対する応答メッセージのデータ形式を指定します。

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。
バイナリーコードを指定する場合は、CODE と記述します。
「論理」は指定できません。

応答 このコマンドに対する応答はありません。

「IV-2-8」

書式 :MEMORY:READ:FORMAT? ブロック番号

説明 「:MEMORY:READ:NEXT ブロック番号,ワード数」コマンドに対する応答メッセージのフォーマットの
設定選択状況を問い合わせます。

応答 このコマンドを受信した後、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
CODE

「IV-2-9」

書式 :MEMORY?

説明 メモリの情報を問い合わせます。

応答 このコマンドを受信した後、下記の応答メッセージを返送します。

総領域容量, 総空容量

総領域容量 : 「:MEMORY:ASSIGN ブロック番号,バイト数」で確保されているメモリ領域の合計を
ワードの単位で表しています。

総空容量 : MEMORYコマンドシステムで使用できる残りの容量をワードの単位で表しています。
「:MEMORY:ASSIGN ブロック番号,バイト数」コマンドで確保されているメモリ領域が
無い(総領域容量=0ワードの場合、512ワードです)。

例

:MEMORY?① メモリの情報を問い合わせます。

0,512① 未使用であれば、使用容量0、空き容量512ワードの返事があります。

注 : この例で「①」はデリミタを意味します。

本機のメモリ管理方法

本機において、メモリ領域は16ワード単位で管理しています。

「:MEMORY:ASSIGN」コマンドで領域を確保すると総空容量は16ワード単位で減ります。

例えば、「:MEMORY:ASSIGN 0,10」、「:MEMORY:ASSIGN 1,20」とするとブロック0に10ワード、
ブロック1に20ワードが確保され、総空容量は48ワード減ります。

[IV-3] バッファリングされたデータの出力コマンド

PLAYコマンドセット

コマンド	パラメータ	備考	初期値
:PLAY			
:CLOCK			
:LEVel	バイト名称, クロックソースのレベル値 ビット名称, クロックソースのレベル値 ワード名称, クロックソースのレベル値	レベル値の設定 設定値は10~10000000 (m秒) 精度は±100μ秒	10 (m秒)
:LEVel?	バイト名称 ビット名称 ワード名称	レベル値の問い合わせ 応答は10~10000000 (m秒)	
:REPeat	バイト名称, 回数 (0, 1~1000000) ビット名称, 回数 (0, 1~1000000) ワード名称, 回数 (0, 1~1000000)	繰り返し回数の設定 0を指定すると無限	1
:REPeat?	バイト名称 ビット名称 ワード名称	繰り返し回数の問い合わせ 応答は0~1000000	
:ASSign	バイト名称, ブロック番号, データ数 ビット名称, ブロック番号, データ数 ワード名称, ブロック番号, データ数	プレイ入出力の割り当て	割り当てなし
:ASSign?	バイト名称 ビット名称 ワード名称	プレイ入出力の問い合わせ	
[:STARt]	バイト名称, 指令 ビット名称, 指令 ワード名称, 指令	指令は下記のいずれか。 ENable, DISable	
:STATe?	バイト名称 ビット名称 ワード名称	PLAY動作の状態を返す。 下記のいずれか。 IDLE, STANDBY, RUNNING	IDLE

プレイ動作が STANDBY状態にある場合は同一ブロック番号のメモリ領域に対して以下のことを行うことができません。

:MEMORY:ASSIGN ブロック番号, ワード数

プレイ動作が RUNNING状態にある場合は同一ブロック番号のメモリ領域に対して以下のことを行うことができません。

:MEMORY:ASSIGN ブロック番号, ワード数
:MEMORY:WRITE:INITIALIZE ブロック番号
:MEMORY:WRITE:NEXT ブロック番号, データ列
:MEMORY:READ:INITIALIZE ブロック番号
:MEMORY:READ:NEXT? ブロック番号, ワード数

ビット名称 (BITn) : nは0~31までの数字で、本機の正面パネルのLDnで示す端子台およびモニタLEDに対応しています。(対応表を下記に示します)
なお、ビット名称として BITn の代わりに LDn を使用することができます。

BITn	BIT0	BIT1	BIT2	BIT3	BIT4	BIT5	BIT6	BIT7
LDn	LD11	LD12	LD13	LD14	LD15	LD16	LD17	LD18

BITn	BIT8	BIT9	BIT10	BIT11	BIT12	BIT13	BIT14	BIT15
LDn	LD21	LD22	LD23	LD24	LD25	LD26	LD27	LD28

BITn	BIT16	BIT17	BIT18	BIT19	BIT20	BIT21	BIT22	BIT23
LDn	LD31	LD32	LD33	LD34	LD35	LD36	LD37	LD38

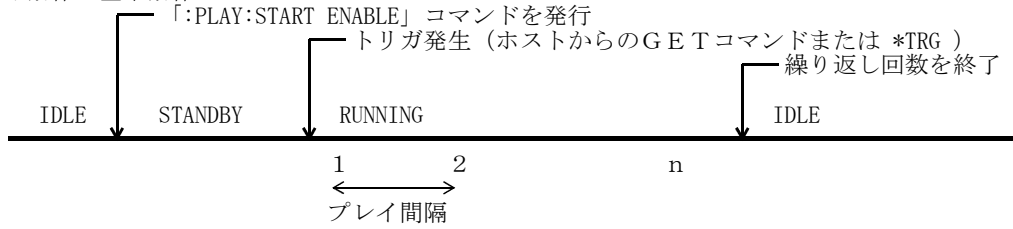
BITn	BIT24	BIT25	BIT26	BIT27	BIT28	BIT29	BIT30	BIT31
LDn	LD41	LD42	LD43	LD44	LD45	LD46	LD47	LD48

バイト名称 (BYTE n) : nは0~3までの数字で、
 BYTE0 は BIT0~BIT7 (LD11~LD18) の8ビットを総称する名称です。
 BYTE1 は BIT8~BIT15 (LD21~LD28) の8ビットを総称する名称です。
 BYTE2 は BIT16~BIT23 (LD31~LD38) の8ビットを総称する名称です。
 BYTE3 は BIT24~BIT31 (LD41~LD48) の8ビットを総称する名称です。

ワード名称 (WORDn) : nは0～1までの数字で、
 WORD0 は BIT0～BIT15 (LD11～LD18, LD21～LD28) の16ビットを総称する名称です。
 WORD1 は BIT16～BIT31 (LD31～LD38, LD41～LD48) の16ビットを総称する名称です。

注 : R L T - 2 1 1 6 E N Cにはリレーおよびリレーモニタは16個が実装されています。
 コマンドでそれを越える指定をしてもエラーにはなりません。

プレイ動作の基本動作



プレイ間隔のタイミングで、メモリ領域から指定出力端へデータを出力します。
 プレイ間隔は「:PLAY:CLOCK:LEVEL」でのレベル値で指定したタイマー時間によります。
 出力するデータの数nは原則として

$$n = \text{「:PLAY:ASSIGN」で指定したデータ数} \times \text{「:PLAY:REPEAT」で指定した回数}$$
 で表されます。

「IV-3-1」

書式 :PLAY:CLOCK:LEVEL ビット名称, クロックソースのレベル値
 :PLAY:CLOCK:LEVEL バイト名称, クロックソースのレベル値
 :PLAY:CLOCK:LEVEL ワード名称, クロックソースのレベル値

説明 ビット名称、またはバイト名称、ワード名称で指定する出力端への信号をプレイするためのクロック源のタイマー値を指定します。

クロックソースのレベル値 : 10～10000000

出力端へ信号を出力する間隔を指定します。

10m秒～10000000m秒の範囲の1m秒の整数倍の値で設定します。

範囲以外はエラーになり、以前の設定値が残ります。

「:PLAY:START ENABLE」コマンドの後、トリガが発生すると、ここで指定された間隔でメモリ領域のデータを出力端へ出力します。

応答 このコマンドに対する応答はありません。
 このコマンドで指定するビット名称/バイト名称/ワード名称のプレイ動作がRUNNING状態の時にこのコマンドを受信すると「実行エラー」になります。

「IV-3-2」

書式 :PLAY:CLOCK:LEVEL? ビット名称
 :PLAY:CLOCK:LEVEL? バイト名称
 :PLAY:CLOCK:LEVEL? ワード名称

説明 ビット名称、またはバイト名称、ワード名称で指定する出力端への信号をプレイするためのクロック源のタイマー値の設定選択状況を問い合わせます。

応答 このコマンドの後、下記の応答メッセージを返送します。
 数値は設定されているクロックソースのレベル値で、プレイ間隔を示しています。
 数値の範囲は10m秒～10000000m秒の範囲の1m秒の整数倍の値です。

数値

「IV-3-3」

書式 :PLAY:REPEAT ビット名称,回数
 :PLAY:REPEAT バイト名称,回数
 :PLAY:REPEAT ワード名称,回数

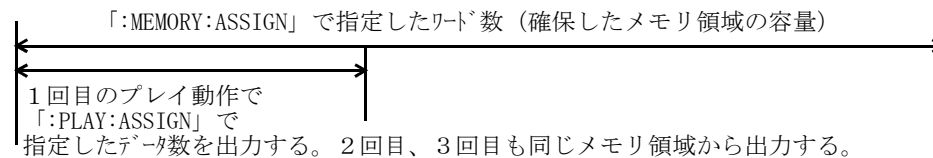
説明 ビット名称、またはバイト名称、ワード名称で指定する出力端へ信号をプレイする繰り返し回数を指定します。

回数 : 0, 1, 2, 3, ..., 1000000

1 ~ 1 0 0 0 0 0 0 を指定すると一回のプレイ動作を指定した回数、繰り返します。

0 を指定すると、:ABORT ([IV-4]を参照)、または *RST を受信するまで繰り返します。

下図に「:PLAY:REPEAT ビット名称/バイト名称/ワード名称, 3」を指定した場合のメモリ領域の使用状況を示します。



この時、「:MEMORY:WRITE:NEXT」で書き込んだデータの数が「:PLAY:ASSIGN」で指定したデータの数より少ない場合、出力したデータの数が「:PLAY:ASSIGN」で指定したデータの数に満たなくてもこの回を終了し、次の回に移ります。

応答 このコマンドに対する応答はありません。
 このコマンドで指定するビット名称/バイト名称/ワード名称のプレイ動作がRUNNING状態の時にこのコマンドを受信すると「実行エラー」になります。

「IV-3-4」

書式 :PLAY:REPEAT? ビット名称
 :PLAY:REPEAT? バイト名称
 :PLAY:REPEAT? ワード名称

説明 ビット名称、またはバイト名称、ワード名称で指定する出力端へ信号をプレイする繰り返し回数の設定値を問い合わせます。

応答 このコマンドを受信した後、設定されている回数を10進整数で返送します。

「IV-3-5」

書式 :PLAY:ASSIGN ビット名称,ブロック番号,データ数
 :PLAY:ASSIGN バイト名称,ブロック番号,データ数
 :PLAY:ASSIGN ワード名称,ブロック番号,データ数

説明 ビット名称、またはバイト名称、ワード名称で指定する出力端へ信号をプレイするデータが格納されているメモリ領域の割り当てを行います。

ブロック番号 : 0, 1

あらかじめ、「MEMORY:ASSIGN ブロック番号,ワード数」コマンドで、メモリ領域とその容量を定義確保しておかなければなりません。

データ数 : 1以上、メモリ領域容量以内

一回のプレイ動作で出力するデータの数を指定します。

0 を指定すると、プレイデータ源とデータ出力先の割り当てを解除 (解放) します。

応答 このコマンドに対する応答はありません。
 このコマンドのブロック番号で指定するメモリ領域の領域容量が「MEMORY:ASSIGN ブロック番号,ワード数」コマンドのワード数で、定義確保されていない場合は「実行エラー」になります。

このコマンドで指定するビット名称/バイト名称/ワード名称のプレイ動作がSTANDBY状態やRUNNING状態の時にこのコマンドを受信すると「実行エラー」になります。

このコマンドで指定するビット名称/バイト名称/ワード名称に、ブロック番号で指定するメモリ領域とその容量をすでに定義確保している場合は、ワード数が0なら割り当ての解除を行います。
 0でない場合は「実行エラー」になります。

このコマンドで指定するビット名称/バイト名称/ワード名称に、ブロック番号で指定する他のメモリ領域とその容量をすでに定義確保している場合は、「実行エラー」になります。

「IV-3-6」

書式 :PLAY:ASSIGN? ビット名称
 :PLAY:ASSIGN? バイト名称
 :PLAY:ASSIGN? ワード名称

説明 ビット名称、またはバイト名称、ワード名称で指定する出力端へ信号をプレイするデータが格納されているメモリ領域の割り当て状況を問い合わせます。

応答 このコマンド受信後、下記の応答メッセージを返送します。

ブロック番号, データ数

応答メッセージのブロック番号が-1、データ数が0の場合は、指定されたビット名称/バイト名称/ワード名称と指定されたブロック番号のメモリ領域が結び付けられていない(割り当てられていない)ことを示します。

「IV-3-7」

書式 :PLAY[:START] ビット名称, 指令
 :PLAY[:START] バイト名称, 指令
 :PLAY[:START] ワード名称, 指令

説明 ビット名称、またはバイト名称、ワード名称で指定する出力端へ信号のプレイ動作を開始、終了させます。

「:PLAY:START ビット名称/バイト名称/ワード名称, ENABLE」の後のトリガ発生でプレイ動作を開始します。メモリ領域からデータを出力し、「:PLAY:CLOCK:LEVEL」で指定した間隔でリレーを動作/復旧させます。書き込まれたデータの数を越えたら終了します。

「:PLAY:START ビット名称/バイト名称/ワード名称, DISABLE」でプレイ動作を終了し、IDLE状態になります。

指令: ENABLE, DISABLE

ENABLEで開始します。しかし、このコマンド実行以前に「:PLAY:ASSIGN」コマンドが実行されている必要があります。
 DISABLEで終了します。

応答 このコマンドに対する応答はありません。

「:PLAY:START ビット名称/バイト名称/ワード名称, ENABLE」を受信したとき、指定と同じビット名称/バイト名称/ワード名称に対するプレイ動作がSTANDBY状態やRUNNING状態にある時は無視します。

「:PLAY:START ビット名称/バイト名称/ワード名称, DISABLE」を受信したとき、指定と同じビット名称/バイト名称/ワード名称に対するプレイ動作がIDLE状態にある時は無視します。

「:PLAY:START ビット名称, ENABLE」を受信したとき、指定のビット名称が含まれるバイト名称/ワード名称に対するプレイ動作がSTANDBY状態やRUNNING状態にある時は「実行エラー」になります。

「:PLAY:START バイト名称, ENABLE」を受信したとき、指定のバイト名称が含まれるワード名称、または指定のバイト名称に含まれるビット名称に対するプレイ動作がSTANDBY状態やRUNNING状態にある時は「実行エラー」になります。

「:PLAY:START ワード名称, ENABLE」を受信したとき、指定のワード名称に含まれるビット名称/バイト名称に対するプレイ動作がSTANDBY状態やRUNNING状態にある時は「実行エラー」になります。

「:PLAY:ASSIGN」コマンドが実行されていないビット名称/バイト名称/ワード名称に対する「:PLAY:START ビット名称/バイト名称/ワード名称, ENABLE」を受信すると「実行エラー」になります。

「:PLAY:START ビット名称/バイト名称/ワード名称, ENABLE」を受信したとき、指定されたビット名称/バイト名称/ワード名称に割り当てられたメモリ領域に対するプレイ動作がSTANDBY状態やRUNNING状態にある時は「実行エラー」になります。

「IV-3-8」

書式 :PLAY:STATE? ビット名称
 :PLAY:STATE? バイト名称
 :PLAY:STATE? ワード名称

説明 ビット名称、またはバイト名称、ワード名称で指定する出力端への信号のプレイ動作の状態を問い合わせます。

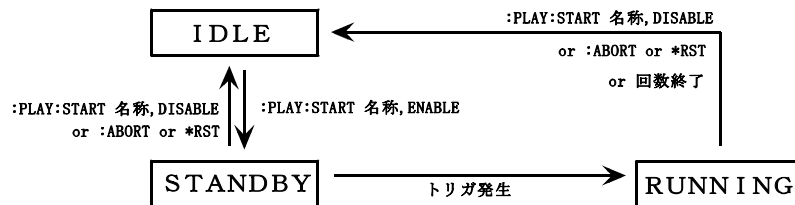
応答 このコマンドの受信後、下記のいずれかの応答メッセージを返送します。

IDEL
 STANDBY
 RUNNING

IDLE状態：「:PLAY:START ビット名称/バイト名称/ワード名称 ENABLE」コマンドを受信していません。
 または、指定された一連のプレイ動作をすべて終了しています。
 または、「:PLAY:START ビット名称/バイト名称/ワード名称 DISABLE」コマンドを受信したか、
 *RST、*ABORTなどの受信により、プレイ動作を強制終了しています。

STANDBY状態：「:PLAY:START ビット名称/バイト名称/ワード名称 ENABLE」コマンドを受信し、
 トリガの発生を待っています。

RUNNING状態：「:PLAY:START ビット名称/バイト名称/ワード名称 ENABLE」コマンドを受信し、
 トリガが発生し、一連のサンプル動作を行っています。



[IV-4] アボート・コマンド

ABORTコマンドセット

コマンド	パラメータ	備考
:ABORt		

トリガ・システムをアイドル・ステートにセットする。

「IV-4-1」

書式 :ABORT

説明 トリガ・システムをアイドル・ステートにし、プレイ動作の状態をIDLEにします。

応答 このコマンドに対する応答はありません。

【V】イーサネットアクセスDLLの使用方法

[V-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。

[V-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

[V-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

[V-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。

- 1 0 6	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 I P アドレスのデバイスに接続できない場合など）	P C 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。
- 1 0 7	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。
- 1 0 8	指定 I P アドレスへの接続準備に失敗した。	
- 2 0 0	指定 I P アドレスへの送信準備に失敗した。	
- 2 0 1	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。
- 3 0 0	D L L 内部の問題	発生状況のご連絡をお願いします。
- 3 0 1	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。
- 3 0 2	送信準備に失敗。受信準備に失敗。	
- 3 0 3	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 I P アドレスからの受信ができなかった場合）	
- 3 0 4	指定 I P アドレスからの受信に失敗した。	
- 4 0 0	送信や受信の最終処理に失敗した。	
- 5 0 0	MAC アドレスから I P アドレスを探す場合の準備に失敗した。	
- 5 0 1	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。
- 5 0 8	MAC アドレスから I P アドレスを探す場合の終了処理に失敗した。	
- 5 0 7	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	
- 5 0 9	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	
- 5 1 0	MAC アドレスから I P アドレスを探す場合に失敗した。	
- 5 1 1	MAC アドレスから I P アドレスを探す場合に失敗した。	
- 5 1 2	D L L 内部の問題	発生状況のご連絡をお願いします。
- 5 1 3	D L L 内部の問題	発生状況のご連絡をお願いします。
- 1 0 0 5	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。I P アドレスを受け取る文字バッファは 1 6 文字分以上必要です。
- 1 0 0 7	入力異常 入力された I P アドレスが、自分の I P 到達範囲内がない場合に発生する。	設定する P C から接続可能な I P アドレスを指定してください。 P C のネットマスク・I P アドレスと本機の I P アドレスの関係を確認して下さい。
- 1 0 0 8	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 I P アドレスを確認して下さい。P C とデバイス間の L A N ケーブルやハブの状況を確認して下さい。