

# REX シリーズ操作説明書

# 目次

目次	2
使用上の注意	3
基本操作方法	4
ファイルを先頭から読み込む	5
ファイルを一部分だけ読み込む	6
ファイルを新規作成する	7
既存ファイルへデータを追加する	8
既存ファイルの一部を書き換える	9
開いているファイルの読み書き位置を得る	10
開いているファイルのファイルサイズを得る	11
ファイルを削除する	12
ファイル名を変更する	13
ディレクトリを作成する	14
ディレクトリを削除する	15
ファイルの一覧を表示する(テキストモードのみ)	16
ファイルの一覧を取得する(バイナリモードのみ)	17
メディアをフォーマットする	18
RTC バックアップ電源について (REX2,REX3,REX4,REX45,REX50,REX60)	19
MOUNT/UNMONT について	20
ファイル名に使用できる文字について	20
OPEN モードの組み合わせについて	21
USB マウス、USB キーボードへの対応について(REX70)	22
テキストモード	23
テキストモードコマンドフォーマット	24
バイナリモード	77
新バイナリモード	79
バイナリモード/新バイナリモードコマンドフォーマット	81
エラーコード	114

汎用ポートについて .....	115
REX50 .....	115
REX60 .....	116
USB キーボードコード表 .....	117
キーコード .....	117
キャラクタコード .....	118
ジョイスティックボタン割り当て .....	119

## 使用上の注意

- 本製品の仕様範囲を超える条件において使用された場合の動作の保証はいたしません。
- 本製品を改造した場合、一切の保証はいたしません。
- 本製品や付属品のリバースエンジニアリングはお断りします。
- 腐食性ガス、可燃性ガス等の環境中でのご使用はご遠慮ください。
- 人命、事故に関わる特別な品質や信頼性が要求される用途でのご使用はご遠慮ください。
- 本製品またはマニュアルには誤りがある可能性があります。製品等に組み込む場合は、十分に評価してからご使用ください。
- 本製品を使用した結果、事故や損失が発生した場合、弊社では一切その責を負いません。

# 基本操作方法

ここでは、主にテキストモードでの操作についての説明です。

網掛けで表記されている部分はテキストモード時に REX へ送るデータを示しています。

四角で囲まれて表記されている部分は、キャラクタコードを 16 進数で示したものです。

バイナリモードのコマンドとデータの最後の 1 バイトの太字はチェックサムを示しています。

## ファイル先頭から読み込む

TEST.TXT というファイルの先頭からのデータを読み込む例を示します。  
あらかじめ TEST.TXT を作成してあり、TEST.TXT の内容は  
abcdefg  
です。

メディアをソケットに挿入します。

MOUNT コマンドを実行して、ファイルへアクセスする準備を行います。MOUNT コマンドは、メディアが挿入された後に 1 回だけ実行してください。REX80 をご利用の場合は MOUNTD コマンドをご使用ください。

### テキストモード

```
MOUNT 0Dh  
OK 0Dh
```

### バイナリモード

```
コマンド 82h 7Dh  
データ なし
```

OPEN コマンドを読み込みモード (OPEN\_EXISTING + READ = 17) でオープンします。  
ファイル名の指定には、フルパス (ディレクトリ名をすべて記述) で指定してください。  
ディレクトリ名を省略すると、ルートディレクトリのファイルを対象にします。  
OPEN コマンドに成功すると、ファイル識別子が返ってきます。

### テキストモード

```
OPEN ¥TEST.TXT 17 0Dh  
0 0Dh
```

### バイナリモード

```
コマンド 84h 11h 09h 5Ch 54h 45h 53h 54h  
2Eh 54h 58h 54h 97h  
データ 00h FFh
```

READ コマンドを実行します。READ コマンドで一度に読み込めるデータは 512 バイトまでです。それ以上の大きさのファイルを読み込みたい場合は、READ コマンドを繰り返して実行してください。  
READ コマンド実行後に返される値は、読み込んだサイズ、読み込んだデータのチェックサムです。次の行からファイルのデータが返されます。

### テキストモード

```
READ 0 512 0Dh  
7 188 0Dh  
abcdefg
```

### バイナリモード

```
コマンド 86h 00h 02h 00h 77h  
データ 00h 07h 61h 62h 63h 64h 65h 66h  
67h 3Ch
```

ファイルの読み込みが終わったら、CLOSE コマンドを実行します。

### テキストモード

```
CLOSE 0 0Dh  
OK 0Dh
```

### バイナリモード

```
コマンド 85h 00h 7Ah  
データ なし
```

## ファイルを一部分だけ読み込む

TEST.TXT というファイルの 4 バイト目からのデータを読み込む例を示します。

あらかじめ TEST.TXT を作成してあり、TEST.TXT の内容は

abcdefg

です。

メディアをソケットに挿入します。

MOUNT コマンドを実行して、ファイルへアクセスする準備を行います。MOUNT コマンドは、メディアが挿入された後に 1 回だけ実行してください。REX80 をご利用の場合は MOUNTD コマンドをご使用ください。

**テキストモード**

MOUNT 0Dh

OK 0Dh

**バイナリモード**

コマンド 82h 7Dh

データ なし

OPEN コマンドを読み込みモード (OPEN\_EXISTING + READ = 17) でオープンします。

ファイル名の指定には、フルパス (ディレクトリ名をすべて記述) で指定してください。

ディレクトリ名を省略すると、ルートディレクトリのファイルを対象にします。

OPEN コマンドに成功すると、ファイル識別子が返ってきます。

**テキストモード**

OPEN ¥TEST.TXT 17 0Dh

0 0Dh

**バイナリモード**

コマンド 84h 11h 09h 5Ch 54h 45h 53h 54h

2Eh 54h 58h 54h 97h

データ 00h FFh

SEEK コマンドで読み込みたい位置を指定します。ここでは、先頭から +3 バイト移動します。SEEK コマンドが成功すると、実際に設定されたファイルの位置を返します。

**テキストモード**

SEEK 0 3 0 0Dh

3 0Dh

**バイナリモード**

コマンド 88h 00h 00h 00h 00h 03h 00h 74h

データ 00h 00h 00h 03h FCh

READ コマンドを実行します。READ コマンドで一度に読み込めるデータは 512 バイトまでです。それ以上の大きさのファイルを読み込みたい場合は、READ コマンドを繰り返して実行してください。

READ コマンド実行後に返される値は、読み込んだサイズ、読み込んだデータのチェックサムです。次の行からファイルのデータが返されます。

**テキストモード**

READ 0 4 0Dh

4 150 0Dh

defg

**バイナリモード**

コマンド 86h 00h 00h 04h 75h

データ 00h 04h 64h 65h 66h 67h 65h

ファイルの読み込みが終わったら、CLOSE コマンドを実行します。

**テキストモード**

CLOSE 0 0Dh

OK 0Dh

**バイナリモード**

コマンド 85h 00h 7Ah

データ なし

## ファイルを新規作成する

TEST.TXT というファイルを新規作成する例を示します。  
この処理により TEST.TXT の内容は  
abcdefg  
となります。

メディアをソケットに挿入します。

MOUNT コマンドを実行して、ファイルへアクセスする準備を行います。MOUNT コマンドは、メディアが挿入された後に 1 回だけ実行してください。REX80 をご利用の場合は MOUNTD コマンドをご使用ください。

### テキストモード

```
MOUNT 0Dh  
OK 0Dh
```

### バイナリモード

```
コマンド 82h 7Dh  
データ なし
```

OPEN コマンドを読み込みモード (CREATE\_NEW + WRITE = 2) でオープンします。  
ファイル名の指定には、フルパス (ディレクトリ名をすべて記述) で指定してください。  
ディレクトリ名を省略すると、ルートディレクトリのファイルを対象にします。  
OPEN コマンドに成功すると、ファイル識別子が返ってきます。

### テキストモード

```
OPEN ¥TEST.TXT 2 0Dh  
0 0Dh
```

### バイナリモード

```
コマンド 84h 02h 09h 5Ch 54h 45h 53h 54h  
2Eh 54h 58h 54h A6h  
データ 00h FFh
```

WRITE コマンドを実行します。WRITE コマンドで一度に書き込めるデータは 512 バイトまでです。それ以上の大きさのファイルを読み込みたい場合は、WRITE コマンドを繰り返して実行してください。

### テキストモード

```
WRITE 0 0Dh abcdefg 03h  
OK 0Dh
```

### バイナリモード

```
コマンド 87h 00h 00h 07h 61h 62h 63h 64h  
65h 66h 67h B5h  
データ なし
```

ファイルの書き込みが終わったら、CLOSE コマンドを実行します。

### テキストモード

```
CLOSE 0 0Dh  
OK 0Dh
```

### バイナリモード

```
コマンド 85h 00h 7Ah  
データ なし
```

## 既存ファイルヘデータを追加する

TEST.TXT というファイルへ 12345 を追記する例を示します。

あらかじめ TEST.TXT を作成してあり、TEXT.TXT の内容は

abcdefg

です。

この処理により TEST.TXT の内容は

abcdefg → abcdefg12345

となります。

REX80 でご使用時、マウント先の FTP サーバの設定で追記が禁止されている事があります。

メディアをソケットに挿入します。

MOUNT コマンドを実行して、ファイルへアクセスする準備を行います。MOUNT コマンドは、メディアが挿入された後に 1 回だけ実行してください。REX80 をご利用の場合は MOUNTD コマンドをご使用ください。

テキストモード

MOUNT 0Dh

OK 0Dh

バイナリモード

コマンド 82h 7Dh

データ なし

OPEN コマンドを読み込みモード (OPEN\_EXISTING + WRITE = 18) でオープンします。

ファイル名の指定には、フルパス (ディレクトリ名をすべて記述) で指定してください。

ディレクトリ名を省略すると、ルートディレクトリのファイルを対象にします。

OPEN コマンドに成功すると、ファイル識別子が返ってきます。

テキストモード

OPEN ¥TEST.TXT 18 0Dh

0 0Dh

バイナリモード

コマンド 84h 12h 09h 5Ch 54h 45h 53h 54h

2Eh 54h 58h 54h 96h

データ 00h FFh

SEEK コマンドを実行して、ファイルの終端へ移動します。

テキストモード

SEEK 0 0 2 0Dh

7 0Dh

バイナリモード

コマンド 88h 00h 00h 00h 00h 00h 02h 75h

データ 00h 00h 00h 07h F8h

WRITE コマンドを実行します。WRITE コマンドで一度に書き込めるデータは 512 バイトまでです。それ以上の大きさのファイルを読み込みたい場合は、WRITE コマンドを繰り返して実行してください。

テキストモード

WRITE 0 0Dh 12345 03h

OK 0Dh

バイナリモード

コマンド 87h 00h 00h 05h 31h 32h 33h 34h

35h 74h

データ なし

ファイルの書き込みが終わったら、CLOSE コマンドを実行します。

テキストモード

CLOSE 0 0Dh

OK 0Dh

バイナリモード

コマンド 85h 00h 7Ah

データ なし

## 既存ファイルの一部を書き換える

TEST.TXT のファイルの先頭から 4 バイト目～を 12 へ書き換える例を示します。

あらかじめ TEST.TXT を作成してあり、TEST.TXT の内容は

```
abcdefg
```

です。

この処理により TEST.TXT の内容は

```
abcdefg → abc12fg
```

となります。

REX80 (FTP サーバマウント) ではファイルの一部書き換えが使用できません。サーバによってはファイルの内容が消える場合がございますので、ご注意ください。

メディアをソケットに挿入します。

MOUNT コマンドを実行して、ファイルへアクセスする準備を行います。MOUNT コマンドは、メディアが挿入された後に 1 回だけ実行してください。

### テキストモード

```
MOUNT 0Dh
```

```
OK 0Dh
```

### バイナリモード

```
コマンド 82h 7Dh
```

```
データ なし
```

OPEN コマンドを読み込みモード (OPEN\_EXISTING + WRITE = 18) でオープンします。

ファイル名の指定には、フルパス (ディレクトリ名をすべて記述) で指定してください。

ディレクトリ名を省略すると、ルートディレクトリのファイルを対象にします。

OPEN コマンドに成功すると、ファイル識別子が返ってきます。

### テキストモード

```
OPEN ¥TEST.TXT 18 0Dh
```

```
0 0Dh
```

### バイナリモード

```
コマンド 84h 12h 09h 5Ch 54h 45h 53h 54h
```

```
2Eh 54h 58h 54h 96h
```

```
データ 00h FFh
```

SEEK コマンドを実行して、変更したい位置を指定します。ここでは先頭から +3 バイト移動します。

### テキストモード

```
SEEK 0 3 0 0Dh
```

```
3 0Dh
```

### バイナリモード

```
コマンド 88h 00h 00h 00h 00h 00h 02h 75h
```

```
データ 00h 00h 00h 07h F8h
```

WRITE コマンドを実行します。WRITE コマンドで一度に書き込めるデータは 512 バイトまでです。それ以上の大きさのファイルを読み込みたい場合は、WRITE コマンドを繰り返して実行してください。

### テキストモード

```
WRITE 0 0Dh 12 03h
```

```
OK 0Dh
```

### バイナリモード

```
コマンド 87h 00h 00h 02h 31h 32h 13h
```

```
データ なし
```

ファイルの書き込みが終わったら、CLOSE コマンドを実行します。

### テキストモード

```
CLOSE 0 0Dh
```

```
OK 0Dh
```

### バイナリモード

```
コマンド 85h 00h 7Ah
```

```
データ なし
```

## 開いているファイルの読み書き位置を得る

現在の位置を知りたい時は、SEEK コマンドを実行してください。現在の位置から+0 バイト移動すると、位置だけを取得できます。

### テキストモード

```
SEEK 0 0 1 0Dh  
50Dh
```

### バイナリモード

```
コマンド 88h 00h 00h 00h 00h 00h 01h 76h  
データ 00h 00h 00h 05h FAh
```

## 開いているファイルのファイルサイズを得る

開いているファイルのサイズを知りたい場合は、SEEK コマンドでファイル終端へ移動してください。SEEK コマンドの戻り値が、ファイルサイズになります。

### テキストモード

```
SEEK 0 0 20Dh
```

```
32430Dh
```

### バイナリモード

```
コマンド 88h 00h 00h 00h 00h 00h 02h 75h
```

```
データ 00h 00h 0Ch ABh 48h
```

## ファイルを削除する

メディアをソケットに挿入します。

MOUNT コマンドを実行して、ファイルへアクセスする準備を行います。MOUNT コマンドは、メディアが挿入された後に 1 回だけ実行してください。REX80 をご利用の場合は MOUNTD コマンドをご使用ください。

### テキストモード

```
MOUNT0Dh  
OK0Dh
```

### バイナリモード

```
コマンド 82h 7Dh  
データ なし
```

DELETE コマンドを実行します。

### テキストモード

```
DELETE ¥TEST.TXT0Dh  
OK0Dh
```

### バイナリモード

```
コマンド 89h 09h 5Ch 54h 45h 53h 54h 2Eh  
54h 58h 54h A3h  
データ なし
```

## ファイル名を変更する

メディアをソケットに挿入します。

MOUNT コマンドを実行して、ファイルへアクセスする準備を行います。MOUNT コマンドは、メディアが挿入された後に1回だけ実行してください。REX80 をご利用の場合は MOUNTD コマンドをご使用ください。

### テキストモード

```
MOUNT 0Dh  
OK 0Dh
```

### バイナリモード

```
コマンド 82h 7Dh  
データ なし
```

MOVE コマンドを実行します。

移動元ディレクトリと、移動先ディレクトリへ同じディレクトリを設定すると、ファイル名の変更を行うことができます。

ファイル名の指定には、フルパス(ディレクトリ名をすべて記述)で指定してください。

ディレクトリ名を省略すると、ルートディレクトリのファイルを対象にします。

### テキストモード

```
MOVE ¥TEST. TXT ¥ABC. TXT 0Dh  
OK 0Dh
```

### バイナリモード

```
コマンド 8Ah 09h 5Ch 54h 45h 53h 54h 2Eh  
54h 58h 54h 08h 5Ch 41h 42h 43h  
2Eh 54h 58h 54h 4Ah  
データ なし
```

## ディレクトリを作成する

メディアをソケットに挿入します。

MOUNT コマンドを実行して、ファイルへアクセスする準備を行います。MOUNT コマンドは、メディアが挿入された後に 1 回だけ実行してください。REX80 をご利用の場合は MOUNTD コマンドをご使用ください。

### テキストモード

```
MOUNT 0Dh
OK 0Dh
```

### バイナリモード

```
コマンド 82h 7Dh
データ なし
```

MKDIR コマンドを実行します。

### テキストモード

```
MKDIR ¥TESTDIR 0Dh
OK 0Dh
```

### バイナリモード

```
コマンド 8Bh 08h 5Ch 54h 45h 53h 54h 44h
          54h 49h 52h 9Dh
データ なし
```

複数の階層のディレクトリを一度に作成することができませんので、一階層毎に MKDIR コマンドを実行して、ディレクトリを作成してください。

### テキストモード

```
MKDIR ¥DIR1¥DIR2¥DIR3 0Dh
ERROR 0Dh
```

### バイナリモード

```
コマンド 8Bh 0Fh 5Ch 44h 49h 52h 31h 5Ch
          44h 49h 52h 32h 5Ch 44h 49h 52h
          33h 1Eh
データ なし(エラー)
```

↓

### テキストモード

```
MKDIR ¥DIR1 0Dh
OK 0Dh
```

### バイナリモード

```
コマンド 8Bh 05h 5Ch 44h 49h 52h 31h
          03h
データ なし
```

```
MKDIR ¥DIR1¥DIR2 0Dh
```

```
OK 0Dh
```

```
コマンド 8Bh 0Ah 5Ch 44h 49h 52h 31h 5Ch
```

```
44h 49h 52h 32h 91h
```

データ なし

```
MKDIR ¥DIR1¥DIR2¥DIR3 0Dh
```

```
OK 0Dh
```

```
コマンド 8Bh 0Fh 5Ch 44h 49h 52h 31h 5Ch
```

```
44h 49h 52h 32h 5Ch 44h 49h 52h
```

```
33h 1Eh
```

データ なし

## ディレクトリを削除する

メディアをソケットに挿入します。

MOUNT コマンドを実行して、ファイルへアクセスする準備を行います。MOUNT コマンドは、メディアが挿入された後に1回だけ実行してください。REX80 をご利用の場合は MOUNTD コマンドをご使用ください。

### テキストモード

MOUNT 0Dh  
OK 0Dh

### バイナリモード

コマンド 82h 7Dh  
データ なし

RMDIR コマンドを実行します。

### テキストモード

RMDIR ¥TESTDIR 0Dh  
OK 0Dh

### バイナリモード

コマンド 8Ch 08h 5Ch 54h 45h 53h 54h 44h  
54h 49h 52h 9Ch  
データ なし

削除しようとするディレクトリの中にファイルがある場合は、あらかじめファイルを削除しておいてください。

## ファイルの一覧を表示する (REX2, REX3, REX4, REX45, REX50, REX60, REX70/テキストモードのみ)

メディアをソケットに挿入します。

MOUNT コマンドを実行して、ファイルへアクセスする準備を行います。MOUNT コマンドは、メディアが挿入された後に 1 回だけ実行してください。

```
MOUNT␣␣␣  
OK␣␣␣
```

DIR コマンドを実行します。ルートディレクトリのすべてのファイルを表示する場合は¥\*. \*を指定します。

```
DIR ¥*. *␣␣␣  
TEST. TXT                8 2002/11/26 12:34:56␣␣␣  
1. BIN                   124 2002/11/26 13:00:14␣␣␣  
OK␣␣␣
```

拡張子が TXT のファイルだけを表示するには¥\*. TXT を指定します。

```
DIR ¥*. TXT␣␣␣  
TEST. TXT                8 2002/11/26 12:34:56␣␣␣  
OK␣␣␣
```

## ファイルの一覧を取得する (REX2, REX3, REX4, REX45, REX50, REX60, REX70/バイナリモードのみ)

メディアをソケットに挿入します。

MOUNT コマンドを実行して、ファイルへアクセスする準備を行います。MOUNT コマンドは、メディアが挿入された後に 1 回だけ実行してください。

OPEN コマンドをディレクトリ読み込みモード (DIR = 4) で実行します。ファイル名には取得したいファイルのあるディレクトリ名を指定してください。

OPEN コマンドに成功すると、ファイル識別子が返ってきます。

READ コマンドで 22 バイトのデータを読み込みます。

+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11	+12	+13
ファイル名													属性

+14	+15	+16	+17	+18	+19	+20	+21
更新時刻		更新日付		ファイルサイズ			

### 属性

ビット	内容
bit7	
bit6	
bit5	アーカイブ属性
bit4	サブディレクトリ属性
bit3	ボリュームラベル属性
bit2	システムファイル属性
bit1	不可視属性
bit0	読み出し専用属性

### 更新時刻

ビット	内容
bit15-11	時間 (0~23)
bit10-5	分 (0~59)
bit4-0	秒を 2 で割った値 (0~29)

### 更新日付

ビット	内容
bit15-9	1980 年からの経過年数 (0~99)
bit8-5	月 (1~12)
bit4-0	日 (1~31)

READ コマンドでデータが読み込めなくなるまで繰り返すことで、指定されたディレクトリのファイルをすべて列挙します。

列挙が終わったら、CLOSE コマンドを実行します。

## メディアをフォーマットする (REX2, REX3, REX4, REX45, REX50, REX60, REX70)

メディアをソケットに挿入します。

FORMAT コマンドを実行します。

### テキストモード

FORMAT 0Dh

OK 0Dh

### バイナリモード

コマンド 8Dh 72h

データ なし

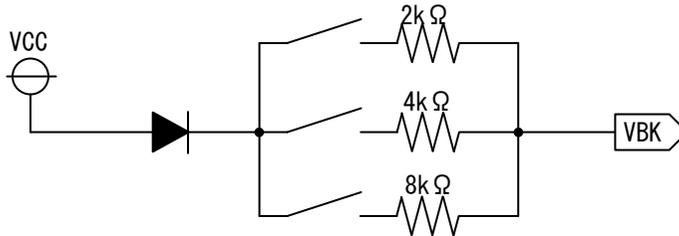
FORMAT コマンドを実行すると、UNMOUNT 状態になりますので、ファイルへアクセスする前に MOUNT コマンドを実行する必要があります。

## RTC バックアップ電源について (REX2, REX3, REX4, REX45, REX50, REX60)

RTC バックアップ電源に充電池を接続する場合は、CHARGE コマンドで充電するように設定してください。初期値は 0 (充電禁止) です。

バックアップ電源電圧が 2.0V 未満になり、設定が保持できなくなった場合は 0 (充電禁止) に初期化されます。

バックアップ電源充電回路は



のようになっています。

REX2, 4, 45, 50 の VCC は外部から供給された電源になります。REX3, 60 は 3.3V になります。

## MOUNT/UNMOUNT について (REX2, REX3, REX4, REX45, REX50, REX60, REX70)

MOUNT コマンドを実行すると、ファイルへアクセスするための準備を行います。ファイルの読み書きや、ディレクトリの操作を行う前には、必ず MOUNT コマンドが実行された状態にする必要があります。

UNMOUNT コマンドを実行すると、閉じ忘れたファイルを閉じ、ファイルアクセスを禁止します。

MOUNT コマンド実行後にメディアがソケットから抜かれると、自動的に UNMOUNT 状態になります。この状態でファイルへのアクセスを行った場合、「MOUNT されていない」エラーを返します。

MOUNT コマンド実行後、UNMOUNT せずにメディアを抜き取ると、キャッシュ内蔵タイプのメディア (高速タイプの CF や USB メモリ) の場合は、メディアの内容が破壊される可能性がありますので、ご注意ください。

## ファイル名に使用できる文字について (REX2, REX3, REX4, REX45, REX50, REX60, REX70)

次の文字はファイル名に使用することができません。

文字	キャラクタコード
制御コード	00h~1Fh
空白文字	20h
“	22h
,	2Ch
/	2Fh
:	3Ah
;	3Bh
=	3Dh
¥	5Ch
	7Ch

小文字の a~z は大文字の A~Z に置換されます。

「.」 (キャラクタコード 2Eh) は、ファイル名と拡張子の区切り文字として、ファイル名に 1 つまで含むことができます。

「¥」 (キャラクタコード 5Ch) 「/」 (キャラクタコード 2Fh) はファイルパスのディレクトリ区切り文字として使用される為、ファイル名の中に使用することはできません。

キャラクタコード 80h~FFh を使用することは可能ですが、REX 以外の機器で使用した場合、OS 等によって文字の取り扱いが異なりますのでご注意ください。

## OPEN モードの組み合わせについて

### CREATE\_NEW + WRITE (2)

書き込みモードで、ファイルを新規作成します。  
ファイルが既に存在している場合は、エラーになります。

### CREATE\_NEW + READWRITE (3)

読み書きモードで、ファイルを新規作成します。  
ファイルが既に存在している場合は、エラーになります。

### CREATE\_ALWAYS + WRITE (10)

書き込みモードで、ファイルを新規作成します。  
ファイルが既に存在している場合は、ファイルの内容はすべて消去されます。

### CREATE\_ALWAYS + READWRITE (11)

読み書きモードで、ファイルを新規作成します。  
ファイルが既に存在している場合は、ファイルの内容はすべて消去されます。

### OPEN\_EXISTING + READ (17)

読み込みモードで、ファイルを開きます。  
ファイルが存在しない場合は、エラーになります。

### OPEN\_EXISTING + WRITE (18)

書き込みモードで、ファイルを開きます。  
ファイルが存在しない場合は、エラーになります。

### OPEN\_EXISTING + READWRITE (19)

読み書きモードで、ファイルを開きます。  
ファイルが存在しない場合は、エラーになります。

### OPEN\_ALWAYS + READ (25)

読み込みモードで、ファイルを開きます。  
ファイルが存在しない場合は、新規作成されます。

### OPEN\_ALWAYS + WRITE (26)

書き込みモードで、ファイルを開きます。  
ファイルが存在しない場合は、新規作成されます。

### OPEN\_ALWAYS + READWRITE (27)

読み書きモードで、ファイルを開きます。  
ファイルが存在しない場合は、新規作成されます。

## USB マウス、USB キーボードへの対応について (REX70)

REX70 では、USB マウス、USB キーボードを接続することができます。

マウスを接続すると、マウスの動きに応じて X 座標、Y 座標のカウンタが増減します。GETMOUSE コマンドを使用すると、X 座標、Y 座標とその時のマウスのボタン状態を得ることが出来ます。

キーボードを接続しキーを操作すると、「押されたキャラクタコード」や「押された/離されたイベント」が REX70 内部の FIFO バッファへ格納されます。「押されたキャラクタコード」と「押された/離されたイベント」は別の FIFO バッファになっており、それぞれ 32 個のデータまで保持できます。バッファが満杯になった時は、古いデータが捨てられます。

FIFO バッファのデータを取り出すには、GETKEYC コマンドまたは GETKEYE コマンドを実行してください。GETKEYC コマンドで「押されたキャラクタコード」、GETKEYE コマンドで「押された/離されたイベント」を取得できます。

FIFO バッファをクリアする場合は、CLRKEYC コマンドまたは CLRKEYE コマンドを実行してください。CLRKEYC コマンドで「押されたキャラクタコード」バッファをクリアし、CLRKEYE コマンドで「押された/離されたイベント」バッファをクリアします。

# テキストモード

文字列でコマンドを送信するモードです。起動時にはこのモードになります。

エコーバックはありません。行端のコードは送信、受信ともに `0DH` です。

パラメータの区切りには `20H` (スペース) 使用してください。

コマンド名は半角文字で指定してください。

コマンド名には大文字/小文字どちらでも使用できます。

コマンド中の数値指定は 10 進数の文字列で表記してください。

## 調歩同期シリアル通信タイプ (REX2, REX3U, REX4, REX50, REX60, REX70, REX80)

データ送受信中にブレイク信号を検出すると、データの送受信を中断し、コマンド受信待機状態に移行します。ただし、コマンド実行中はブレイク信号を検出してもコマンド動作自体は中断されません。コマンド実行中にブレイク信号を検出した場合は、コマンド動作が完了後にレスポンスを返さずコマンド受信待機状態に移行します。

## 同期シリアル通信タイプ (REX3S, REX50)

データ送受信中に `-CS` 信号を High レベルに変化させると、データ送受信を中断し、コマンド受信待機状態に移行します。コマンド文字列の最終バイトを送信後すぐに `-CS` 信号を High レベルに変化させると、コマンドが実行されない場合がありますので注意してください。確実にコマンドを実行させたい場合は、レスポンスを受信した後、`-CS` 信号を High レベルに変化させてください。

# テキストモードコマンドフォーマット

## GETTIME(時刻の取得)

### 動作

---

RTC の日時を取得します。

### コマンド

---

GETTIME<sup>ODH</sup>

パラメータはありません。

### レスポンス

---

年/月/日<sup>20H</sup>時:分:秒<sup>ODH</sup>

が返ってきます。

年	現在の年を 4 桁の数値で返します (1980~2079)
月	現在の月を 2 桁の数値で返します (01~12)
日	現在の日を 2 桁の数値で返します (01~31)
時	現在の時を 2 桁の数値で返します (00~23)
分	現在の分を 2 桁の数値で返します (00~59)
秒	現在の秒を 2 桁の数値で返します (00~59)

日時を設定していない場合は 1980/01/01 00:00:00 を返します。

### 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

# SETTIME(時刻の設定)

## 動作

---

RTC の日時を設定します。

ファイルのタイムスタンプは RTC の時刻を使用します。RTC が初期値の場合は、ファイルのタイムスタンプは設定されません。

## コマンド

---

SETTIME[20H]年/月/日[20H]時:分:秒[0DH]

年	設定する年を 4 桁の数値で指定します (1980~2079)
月	設定する月を 2 桁の数値で指定します (01~12)
日	設定する日を 2 桁の数値で指定します (01~31)
時	設定する時を 2 桁の数値で指定します (00~23)
分	設定する分を 2 桁の数値で指定します (00~59)
秒	設定する秒を 2 桁の数値で指定します (00~59)

## レスポンス

---

OK[0DH]

が返ってきます。

## 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

# MOUNT(メディアのマウント)

## 動作

---

メディアのフォーマットをチェックし、メディアへのアクセスの準備をします。

## コマンド

---

MOUNT`ODH`

パラメータはありません。

## レスポンス

---

マウント処理が成功した場合は、

OK`ODH`

が返ってきます。

失敗した場合は

ERROR`ODH`

が返ってきます。

## 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70

REX80 では MOUNTD コマンドを使用してください。

# UNMOUNT(メディアのアンマウント)

## 動作

---

メディアのアクセス処理を終了します。

## コマンド

---

UNMOUNT`[ODH]`

パラメータはありません。

## レスポンス

---

アンマウント処理が成功した場合は、

OK`[ODH]`

が返ってきます。

失敗した場合は

ERROR`[ODH]`

が返ってきます。

## 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70

REX80 では UNMOUNTD コマンドを使用してください。

# OPEN(ファイルを開く)

## 動作

メディア上にあるファイルをオープンします。

## コマンド

OPEN<sup>[20H]</sup>ファイルパス<sup>[20H]</sup>モード<sup>[0DH]</sup>

ファイルパス	オープンするファイルをフルパスで指定します(最大 255 文字)		
モード	アクセスモードと作成モードを論理和した値を指定します。		
	bit7-3	作成モードを指定します。	
		CREATE_NEW (0)	ファイルを新規作成します。ファイルが既に存在する場合は、エラーになります。
		CREATE_ALWAYS (1)	ファイルを常に新規作成します。ファイルが既に存在する場合は、上書きされます。
		OPEN_EXISTING (2)	ファイルを開きます。ファイルが存在しない場合はエラーになります。
		OPEN_ALWAYS (3)	ファイルを開きます。ファイルが存在しない場合は新規作成されます。
	bit2-0	アクセスモードを指定します。	
		READ (1)	読込専用モードでオープンします。
		WRITE (2)	書込専用モードでオープンします。
READWRITE (3)		読み書きモードでオープンします。	

## レスポンス

オープン処理が成功した場合は、  
ファイル識別子<sup>[0DH]</sup>  
が返ってきます。  
ファイル識別子はファイル操作時に指定する値になります。

失敗した場合は  
ERROR<sup>[0DH]</sup>  
が返ってきます。

## 備考

既に2つのファイルが開かれている場合は、新しくファイルを開くことはできません。

## 対応機種

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

# CLOSE(ファイルを閉じる)

## 動作

---

オープンされたファイルをクローズします。

## コマンド

---

CLOSE`20H`ファイル識別子`0DH`

ファイル識別子	クローズしたいファイルのファイル識別子を指定します。
---------	----------------------------

## レスポンス

---

オープン処理が成功した場合は、

OK`0DH`

が返ってきます。

失敗した場合は

ERROR`0DH`

が返ってきます。

## 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

# READ(ファイルから読み込む)

## 動作

---

オープンしたファイルのデータを読み込みます。

## コマンド

---

READ[20H]ファイル識別子[20H]サイズ[0DH]

ファイル識別子	読み込みたいファイルの識別子を指定します。
サイズ	バイト単位で読み込むサイズを指定します(1~512)

## レスポンス

---

読み込み処理が成功した場合は、

読み込んだファイルサイズ[20H]チェックサム[0DH]  
読み込んだデータ

が返ってきます。  
読み込んだデータの最後に [0DH] は付加されません。

失敗した場合は

ERROR[0DH]

が返ってきます。

## 備考

---

読み込みオープンしたファイル以外を READ コマンドで読み込もうとするとエラーになります。  
一度に 513 バイト以上のデータを読み込むことはできません。

読み込み後は、読み込んだサイズだけ読み書き位置が移動されます。

## 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

# WRITE(ファイルへ書き込む)

## 動作

---

オープンしたファイルへデータを書き込みます。

## コマンド

---

WRITE 20Hファイル識別子 0DH  
データ 03H

ファイル識別子	書き込みたいファイルの識別子を指定します。
データ	書き込むデータを指定します。 <span style="border: 1px solid black; padding: 0 2px;">0DH</span> の後から <span style="border: 1px solid black; padding: 0 2px;">03H</span> の前までのデータがファイルへ書き込まれます。 一度に書き込めるデータサイズは 512 バイトまでです。

## レスポンス

---

書き込み処理が成功した場合は、

OK0DH

が返ってきます。

失敗した場合は

ERROR0DH

が返ってきます。

## 備考

---

書き込みオープンしたファイル以外を WRITE コマンドで書き込もうとするとエラーになります。  
一度に 513 バイト以上のデータを書き込むことはできません。

書き込み後は、書き込んだサイズだけ読み書き位置が移動されます。

## 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

## SEEK(読み書き位置の移動)

### 動作

---

オープンしたファイルのデータ読み書き位置を移動します

### コマンド

---

SEEK[20H]ファイル識別子[20H]移動量[20H]基準位置[0DH]

ファイル識別子	ファイルの識別子を指定します。
移動量	基準位置からどれだけ移動するかを数値で指定します。
基準位置	移動基準位置を指定します。 0 ファイルの先頭 1 現在の読み書き位置 2 ファイルの終端

### レスポンス

---

移動処理が成功した場合は、

移動後の位置[0DH]

が返ってきます。

失敗した場合は

ERROR[0DH]

が返ってきます。

### 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

# DELETE(ファイルの削除)

---

## 動作

---

指定されたファイルを削除します。

## コマンド

---

DELETE[20H]ファイルパス[0DH]

ファイルパス	削除したいファイルのフルパスを指定します(最大 255 文字)
--------	---------------------------------

## レスポンス

---

削除処理が成功した場合は、

OK[0DH]

が返ってきます。

失敗した場合は

ERROR[0DH]

が返ってきます。

## 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

# MOVE(ファイルの移動)

---

## 動作

---

指定されたファイルを移動(ファイル名の変更)します。

## コマンド

---

MOVE[20H]移動元ファイルパス[20H]移動先ファイルパス[0DH]

移動元ファイルパス	移動したいファイルのフルパスを指定します(最大 255 文字)
移動先ファイルパス	移動先のフルパスを指定します(最大 255 文字)

## レスポンス

---

移動処理が成功した場合は、

OK[0DH]

が返ってきます。

失敗した場合は

ERROR[0DH]

が返ってきます。

## 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

# MKDIR(ディレクトリの作成)

## 動作

---

指定されたディレクトリを作成します。

## コマンド

---

MKDIR

ディレクトリ名	作成したいディレクトリのフルパスを指定します(最大 255 文字)
---------	-----------------------------------

## レスポンス

---

作成処理が成功した場合は、

OK

が返ってきます。

失敗した場合は

ERROR

が返ってきます。

## 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

## RMDIR(ディレクトリの削除)

---

### 動作

---

指定されたディレクトリを削除します。

### コマンド

---

RMDIR  ディレクトリ名

ディレクトリ名	削除したいディレクトリのフルパスを指定します(最大 255 文字)
---------	-----------------------------------

### レスポンス

---

削除処理が成功した場合は、

OK

が返ってきます。

失敗した場合は

ERROR

が返ってきます。

### 備考

---

削除しようとするディレクトリの中にファイルやディレクトリが存在するとエラーになります。

### 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

# DIR(ファイル一覧表示)

## 動作

---

指定された検索パスのファイルを表示します。

## コマンド

---

DIR[20H]検索パス[0DH]

検索パス	検索したいファイルのパスをワイルドカードを使用して指定します。
------	---------------------------------

## レスポンス

---

処理が成功した場合は、

```
検索結果[0DH]  
:  
:  
検索結果[0DH]  
OK[0DH]
```

が返ってきます。

失敗した場合は

```
ERROR[0DH]
```

が返ってきます。

## 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

# GETFREE(ディスク容量の取得)

## 動作

---

メディアの総容量と、空き容量を取得します。

## コマンド

---

GETFREE`ODH`

パラメータはありません

## レスポンス

---

処理が成功した場合は、

空き容量/総容量`ODH`

が返ってきます。

空き容量、総容量が 2G バイト以上の時は 2147483136 バイトが返ってきます (ver5.13 以降の場合)。

失敗した場合は

ERROR`ODH`

が返ってきます。

## 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70

## GETFREESEC(ディスク容量の取得)

---

### 動作

---

メディアの総容量と、空き容量を 512 バイト単位で取得します。

### コマンド

---

GETFREESEC`ODH`

パラメータはありません

### レスポンス

---

処理が成功した場合は、

空き容量/総容量`ODH`

が返ってきます。

空き容量、総容量は 512 倍した値が実際のバイト単位の容量になります。

失敗した場合は

ERROR`ODH`

が返ってきます。

### 対応機種

---

REX50(ver5.13以降), REX60(ver5.13以降), REX70

# FORMAT(フォーマット)

## 動作

---

メディアをフォーマットします。

## コマンド

---

FORMAT`ODH`

パラメータはありません

## レスポンス

---

フォーマットが成功した場合は、

OK`ODH`

が返ってきます。

失敗した場合は

ERROR`ODH`

が返ってきます。

## 備考

---

MOUNT コマンドを実行しなくてもフォーマットすることができます。  
フォーマット後は UNMOUNT 状態になります。

## 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70

# GETERR(エラーコードの取得)

## 動作

---

最後に発生したエラーのエラーコードを取得します。

## コマンド

---

```
GETERR[ODH]
```

パラメータはありません

## レスポンス

---

```
エラーコード[ODH]
```

が返ってきます。

## 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

# VERSION(バージョン取得)

## 動作

---

ファームウェアのバージョンを取得します。

## コマンド

---

VERSION`0DH`

パラメータはありません

## レスポンス

---

REX3`20H`Version1.0`0DH`

が返ってきます。

## 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

## CHARGE(RTC バックアップ電源充電設定)

---

### 動作

RTCバックアップ電源の充電モードを設定します。

### コマンド

---

CHARGE[20H]モード[0DH]

モード	充電モードを設定します。 0 充電禁止 1 最大 1.3mA で充電 2 最大 650uA で充電 3 最大 325uA で充電
-----	--

### レスポンス

---

OK[0DH]

が返ってきます。

### 備考

---

初期値は0(充電禁止)です。

バックアップ電源電圧が2.0V未満になり、設定が保持できなくなった場合は0(充電禁止)に初期化されます。

### 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60

# STATUS(メディア状態を得る)

## 動作

---

現在のメディアの状態を取得します。

## コマンド

---

STATUS<sup>ODH</sup>

## レスポンス

---

メディアステータス<sup>ODH</sup>

が返ってきます。  
メディアステータスは 10 進数の値です。

## 備考

---

メディアステータスの bit0 (LSB) が 0 の時は、REX のソケットにメディアが挿入されていません。  
メディアステータスの bit0 (LSB) が 1 の時は、REX のソケットにメディアが挿入されています。

メディアステータスの bit1 が 0 の時は、MOUNT されていない状態です。  
メディアステータスの bit1 が 1 の時は、MOUNT された状態です。

REX60 ver5.11 以降、REX70 では、bit7 が USB メモリの挿抜状態を意味します。bit7 が 1 の時は、REX に USB メモリが挿入されており、bit7 が 0 の時は USB メモリは挿入されていません。bit0 は USB メモリが使用可能になった時に 1 になります。  
ステータスの bit2 はライトプロテクトの状態を示します。0 の時は書き込み許可、1 の時は書き込み禁止状態です。

REX80 では bit7 が USB モデムの挿抜状態を示しています。bit7 が 1 の時は、REX に USB モデムが挿入されており、bit7 が 0 の時、USB モデムは挿入されていません。Bit0 は PPP 接続状態を示しており、0 のときは未接続、1 のときは接続中を示しています。bit1 はドライブ 0 の MOUNT 状態、bit2 はドライブ 1 の MOUNT 状態のように bit1~6 はドライブ 0~5 の MOUNT 状態を示します。

## 対応機種

---

REX2 (ver1.08 以降), REX3U (ver2.08 以降), REX3S (ver2.08 以降), REX4 (ver3.08 以降), REX45 (ver3.08 以降), REX50, REX60, REX70, REX80

# AUTOMOUNT(自動マウントモードの設定)

## 動作

---

オートマウント機能の有効/無効を設定します。

## コマンド

---

AUTOMOUNT  設定値

設定値	オートマウントの有効/無効を指定します。
-----	----------------------

## レスポンス

---

OK

が返ってきます。

## 備考

---

設定値に 1 を指定すると、オートマウントが有効になります。  
オートマウントが有効になると、メディアの挿入時に自動で MOUNT コマンドを実行します。

設定値に 0 を指定すると、オートマウントは無効になります。

電源投入直後は、オートマウント無効状態になります。

## 対応機種

---

REX50 (ver5.03 以降), REX60, REX70

# PORTOUT(汎用ポート出力設定)

## 動作

---

基板上の P10, P9 ポートの出力信号を制御します。

## コマンド

---

PORTOUT[20H]ポート番号[20H]出力値[0DH]

ポート番号	設定を行うポートの番号を指定します。
出力値	ポートに出力する値を指定します。 0 = Low レベル 1 = High レベル 2 = Hi-Z 3 = ステータスの bit0 4 = ステータスの bit1

## レスポンス

---

OK[0DH]

が返ってきます。

## 備考

---

出力値に 0 または 1 を指定すると、High または Low レベルがポートに出力されます。  
出力値に 2 を指定すると、ポートはハイインピーダンスになり、入力ポートとして使用できます。  
出力値に 3 を指定すると、STATUS コマンドで得られる値の bit0 の値がポートへ出力されます。  
出力値に 4 を指定すると、STATUS コマンドで得られる値の bit1 の値がポートへ出力されます。  
出力値に 3 または 4 を指定した場合は、最新の STATUS 値が常にポートへ出力されます。

電源投入後、全ポートはハイインピーダンスになります。

P10 の 9 番ピンから High レベルを出力したい場合は、

PORTOUT[20H][9][20H][1][0DH]

とコマンドを送ります。

## 対応機種

---

REX50 (ver5.03 以降), REX60

## PORTIN(汎用ポートの状態を得る)

---

### 動作

---

基板上の P10, P9 ポートの信号を読み取ります。

### コマンド

---

PORTIN $\square$ 20H $\square$ ポート番号 $\square$ 0DH

ポート番号	取得するポート番号を指定します。
-------	------------------

### レスポンス

---

取得値 $\square$ 0DH

が返ってきます。  
指定されたポートが High レベルの場合は 1 が、Low レベルの時は 0 が値になります。

### 備考

---

PORTOUT コマンドで Hi-Z 以外を指定しているポートも読み込みは可能です。

### 対応機種

---

REX50 (ver5.03 以降), REX60

# SETFLOW(シリアル通信のフロー設定)

## 動作

---

シリアル通信のフロー制御の設定を行います。

## コマンド

---

SETFLOW $\overline{20H}$ 受信制御 $\overline{20H}$ 送信制御 $\overline{0DH}$

受信制御	受信フロー制御信号(P2の4ピン)の動作を指定します。 0 = 制御しない 1 = 受信が出来ない時に High レベルを出力します。 2 = 受信が出来ない時に Low レベルを出力します。
送信制御	送信フロー制御信号(P2の6ピン)の動作を指定します。 0 = 信号を無視します。 1 = High レベルの時は、データの送信を行いません。 2 = Low レベルの時は、データの送信を行いません。

## レスポンス

---

OK $\overline{0DH}$

が返ってきます。

## 備考

---

電源投入直後は、受信フロー=2、送信フロー=0 の状態です。

## 対応機種

---

REX50(ver5.06以降), REX60, REX70, REX80

# WRITEN(ファイルへ書き込む)

## 動作

---

オープンしたファイルヘデータを書き込みます。

## コマンド

---

WRITE 20Hファイル識別子 0DH  
データ 03H

ファイル識別子	書き込みたいファイルの識別子を指定します。
データ	書き込むデータを指定します。 <span style="border: 1px solid black; padding: 0 2px;">0DH</span> の後から <span style="border: 1px solid black; padding: 0 2px;">03H</span> の前までのデータがファイルへ書き込まれます。 一度に書き込めるデータサイズは 512 バイトまでです。

## レスポンス

---

書き込み処理が成功した場合は、

ファイルへ書き込んだデータの長さ 0DH

が返ってきます。

失敗した場合は

ERROR 0DH

が返ってきます。

## 備考

---

書き込みオープンしたファイル以外を WRITE コマンドで書き込もうとするとエラーになります。  
一度に 513 バイト以上のデータを書き込むことはできません。

書き込み後は、書き込んだサイズだけ読み書き位置が移動されます。

コマンドの実行が成功すると、ファイルへ書き込んだデータの長さが返ってきます。

## 対応機種

---

REX50 (ver5.06 以降), REX60, REX70, REX80

# SLEEP(スリープ)

## 動作

---

REX をスリープさせます。

## コマンド

---

SLEEP`0DH`

## レスポンス

---

OK`0DH`

が返ってきます。

## 備考

---

レスポンスで OK を返した後に、REX はスリープモードに入ります。  
P9-15 番ピンに LOW レベルを 5msec 以上入力すると、スリープモードから復帰します。

スリープから復帰して 100msec 経過後にシリアルポートが有効になりますので、  
スリープ復帰後すぐにコマンドを送られると、正常に通信できない場合があります。

## 対応機種

---

REX50 (ver5.08 以降)

# GETMOUSE(マウス状態を得る)

## 動作

---

USB ポートに接続されたマウスの座標とボタン状態を得ます。

## コマンド

---

GETMOUSE`[ODH]`

パラメータはありません

## レスポンス

---

処理が成功した場合は、

X座標(16進数8桁), Y座標(16進数8桁), 予約(16進数8桁), ボタン状態(16進数8桁)`[ODH]`

が返ってきます。

X 座標、Y 座標は起動時からの積算値になります。

ボタン状態は各ビットが各ボタンに対応しており、一般的なマウスの場合は、

Bit0 左ボタン

Bit1 右ボタン

Bit2 ホイールボタン

となっています。

## 対応機種

---

REX70

# CLRKEYC(キー入力バッファのクリア)

---

## 動作

---

USB ポートに接続されたキーボード用の入力キャラクタバッファをクリアします。

## コマンド

---

CLRKEYC`ODH`

パラメータはありません

## レスポンス

---

OK`ODH`

が返ってきます。

## 対応機種

---

REX70

# GETKEYC(キー入力データの取得)

## 動作

---

USBポートに接続されたキーボード用の入力キャラクタバッファから1文字取得します。

## コマンド

---

GETKEYC`ODH`

パラメータはありません

## レスポンス

---

処理が成功した場合は、

キャラクタコード`ODH`

が返ってきます。

入力データが無い場合は、0が返ってきます。

## 対応機種

---

REX70

# CLRKEYE(キー入カイベントバッファのクリア)

---

## 動作

---

USB ポートに接続されたキーボード用の入カイベントバッファをクリアします。

## コマンド

---

CLRKEYE $\square$ ODH

パラメータはありません

## レスポンス

---

OK $\square$ ODH

が返ってきます。

## 対応機種

---

REX70

# GETKEYE(キー入カイベントデータの取得)

## 動作

---

USB ポートに接続されたキーボード用の入カイベントバッファから 1 個のデータを取得します。

## コマンド

---

GETKEYE $\boxed{0DH}$

パラメータはありません

## レスポンス

---

処理が成功した場合は、

イベントコード(16進数4桁) $\boxed{0DH}$

が返ってきます。データが無い場合は、0 が返ってきます。

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
予約 (常に 0)							押下状態 0=Up 1=Down	キーコード (4~255)							

下位 8bit はキーコードを示しています。

bit8 はキーの状態で、Bit8 が 0 の時はキーが離された、1 の時はキーが押されたことを示します。

## 対応機種

---

REX70

# KEYCMODE(キー入力データ出力モード)

## 動作

---

USB ポートに接続されたキーボード用の入力キャラクタをそのまま出力するモードへ移行します。

## コマンド

---

KEYCMODE`0DH`

パラメータはありません

## レスポンス

---

OK`0DH`

が返ってきます。

OK 応答後に、キーデータ出力モードになり、キーボードから入力されたキャラクタデータがそのままシリアル通信で出力されます。キーデータ出力モードから抜ける場合は、`03h`を REX へ送信してください。

## 対応機種

---

REX70

# KEYEMODE(キー入カイベント出力モード)

## 動作

---

USB ポートに接続されたキーボード用の入カイベントを出力するモードへ移行します。

## コマンド

---

KEYEMODE`0DH`

パラメータはありません

## レスポンス

---

OK`0DH`

が返ってきます。

OK 応答後に、キーイベント出力モードになり、キーボードのイベントデータを 16 進数 4 桁でシリアル通信で出力されます。キーイベント出力モードから抜ける場合は、`03h`を REX へ送信してください。

## 対応機種

---

REX70

# GETJOYSTICK(ジョイスティック状態取得)

## 動作

---

USB ポートに接続されたジョイスティックの状態を取得します。

## コマンド

---

GETJOYSTICK<sup>ODH</sup>

パラメータはありません

## レスポンス

---

処理に成功すると、

左X軸量(16進数4桁), 左Y軸量(16進数4桁), 右X軸量(16進数4桁), 右Y軸量(16進数4桁),  
左トリガ量(16進数4桁), 右トリガ量(16進数4桁), ボタン状態(16進数8桁)<sup>ODH</sup>

が返ってきます。

左右スティックの各軸量は符号付き 16bit の値を返します。左右トリガは符号無し 16bit の値を返します。ボタン状態の値は、押されたボタンに対応するビットが 1 になります。

## 対応機種

---

REX70

# MOUNTD(ドライブのマウント)

## 動作

---

指定された URL のサーバをマウントします。

## コマンド

---

MOUNTD `[20H]` ドライブ番号 `[20H]` URL `[0DH]`

ドライブ番号	マウントするドライブ番号を 0~5 で指定します
データ	マウントするサーバの URL を指定します。以下の形式で指定してください。  ftp://ユーザー名:パスワード@ホスト:ポート番号/  ユーザー名 = FTP サーバのユーザー名を指定します パスワード = FTP サーバのパスワードを指定します ホスト = FTP サーバの IP アドレスまたはドメイン名を指定します。 ポート番号 = FTP サーバのポート番号を指定します。  ポート番号を省略する場合は、 ftp://ユーザー名:パスワード@ホスト/ と指定してください。ポート番号が省略された場合は、21 が使用されます。

## レスポンス

---

処理に成功すると、  
`OK[0DH]`  
が返ってきます。  
失敗した場合は  
`ERROR[0DH]`  
が返ってきます。

## 備考

---

マウント後に回線切断や USB モデム抜け等で PPP リンクが切れた場合は、アンマウント状態に移行しません。

## 対応機種

---

REX80

# UNMOUNTD(ドライブのアンマウント)

---

## 動作

---

サーバをログオフし、アンマウント状態に移行します。

## コマンド

---

UNMOUNTD[20H]ドライブ番号[0DH]

ドライブ番号	マウントするドライブ番号を 0~5 で指定します
--------	--------------------------

## レスポンス

---

処理に成功すると、

OK[0DH]

が返ってきます。

失敗した場合は

ERROR[0DH]

が返ってきます。

## 対応機種

---

REX80

# CONNECT(PPP 接続開始)

## 動作

---

PPP 接続を開始します。

## コマンド

---

CONNECT[20H]APN番号[0DH]

APN 番号	使用する APN を指定します。 0=イーモバイル制限なしプラン 1=使用禁止(予約) 2=使用禁止(予約) 3=イーモバイル制限ありプラン
--------	--

APN 番号には 0 を指定してください。

イーモバイル制限ありプランを使用する場合は 3 を指定してください。

## レスポンス

---

処理に成功すると、

OK[0DH]

が返ってきます。

失敗した場合は

ERROR[0DH]

が返ってきます。

## 備考

---

コマンド発行後、接続完了を待たずに応答が返ります。

接続完了は STATUS コマンドを発行して、レスポンスの bit0 をチェックするか、DCD 信号をチェックしてください。

## 対応機種

---

REX80

# DISCONNECT(PPP 切断開始)

## 動作

---

PPP 切断処理を開始します。

## コマンド

---

DISCONNECT`ODH`

パラメータはありません。

## レスポンス

---

処理に成功すると、

`OKODH`

が返ってきます。

失敗した場合は

`ERRORODH`

が返ってきます。

## 備考

---

コマンド発行後、切断完了を待たずに応答が返ります。

切断完了は STATUS コマンドを発行して、レスポンスの bit0 をチェックするか、DCD 信号をチェックしてください。

## 対応機種

---

REX80

# HTOPEN(HTTP 接続)

## 動作

---

HTTP 接続を開始します。

## コマンド

---

HTOPEN[20H]アクション[20H]URL[0DH]

アクション	GET または POST を指定します。
URL	接続先リソースの URL を指定します。

## レスポンス

---

処理に成功すると、  
HTTP識別子[0DH]  
が返ってきます。  
失敗した場合は  
ERROR[0DH]  
が返ってきます。

## 対応機種

---

REX80

# HTWRITE(HTTP POST データ出力)

## 動作

---

HTTP 接続完了後の POST データを出力します。

## コマンド

---

HTWRITE[20H]HTTP識別子[0DH]出力データ[03H]

HTTP 識別子	接続済の HTTP 識別子を指定します。
出力データ	出力するデータを指定します。

## レスポンス

---

処理に成功すると、  
OK[0DH]  
が返ってきます。  
失敗した場合は  
ERROR[0DH]  
が返ってきます。

## 備考

---

HTOPEN コマンド発行直後に HTWRITE コマンドを発行してください。  
HTOPEN 発行後に HTREAD コマンドや HTGETRESP コマンドや HTCLOSE コマンドを発行すると、HTWRITE コマンドはエラーになります。

## 対応機種

---

REX80

# HTGETRESP(HTTP レスポンスコード取得)

---

## 動作

---

HTTP 接続完了後の HTTP レスポンスを取得します。

## コマンド

---

HTGETRESP[20H]HTTP識別子[0DH]

HTTP 識別子	接続済の HTTP 識別子を指定します。
----------	----------------------

## レスポンス

---

処理が成功した場合は、

レスポンスコード[0DH]

が返ってきます。

HTTP 処理が正常の場合は 200、HTTP エラーの場合は 500 などのコードが返ってきます。

失敗した場合は

ERROR[0DH]

が返ってきます。

## 対応機種

---

REX80

# HTREAD(HTTP BODY データ入力)

## 動作

---

HTTP 接続完了後の BODY データを入力します。

## コマンド

---

HTREAD[20H]HTTP識別子[20H]サイズ[0DH]

HTTP 識別子	接続済の HTTP 識別子を指定します。
サイズ	入力するデータサイズを 1~512 で指定します。

## レスポンス

---

処理が成功した場合は、

読み込んだサイズ[20H]チェックサム[0DH]  
読み込んだデータ

が返ってきます。  
読み込んだデータの最後に [0DH] は付加されません。

失敗した場合は

ERROR[0DH]

が返ってきます。

## 対応機種

---

REX80

# HTCLOSE(HTTP 切断)

## 動作

---

HTTP 接続を閉じます。

## コマンド

---

HTCLOSE[20H]HTTP識別子[0DH]

HTTP 識別子	接続済の HTTP 識別子を指定します。
----------	----------------------

## レスポンス

---

処理が成功した場合は、

OK[0DH]

が返ってきます。  
失敗した場合は

ERROR[0DH]

が返ってきます。

## 対応機種

---

REX80

# HTTIMEOUT(HTTP タイムアウト時間設定)

## 動作

---

HTTP 処理のタイムアウト時間を設定します。

## コマンド

---

HTTIMEOUT`[20H]`タイムアウト時間`[0DH]`

タイムアウト時間	HTTP 処理のタイムアウト時間をミリ秒単位で指定します。 REX80 の起動後のタイムアウト時間は 60000 ミリ秒です。
----------	--

## レスポンス

---

処理が成功した場合は、

OK`[0DH]`

が返ってきます。  
失敗した場合は

ERROR`[0DH]`

が返ってきます。

## 対応機種

---

REX80

# SKOPEN(ソケット接続)

## 動作

---

ソケット接続を開始します。

## コマンド

---

SKOPEN`[20H]`モード`[20H]`ホスト:ポート`[0DH]`

モード	CONNECT または LISTEN を指定します。
ホスト	接続先の IP アドレスまたはホスト名を指定します。 モードに LISTEN を指定した場合は、localhost と指定してください。
ポート	接続先のポート番号を指定します。 モードに LISTEN を指定した場合は、待ち受けポート番号を指定してください。

## レスポンス

---

処理に成功すると、  
ソケット識別子`[0DH]`  
が返ってきます。  
失敗した場合は  
ERROR`[0DH]`  
が返ってきます。

## 備考

---

モードに CONNECT を指定すると、相手からの接続許可を受信するまで応答を返しません(タイムアウト時間が経過するとエラー応答を返します)。

モードに LISTEN を指定すると、SKOPEN コマンド発行直後に応答を返します。  
相手からの接続要求を受け、接続が完了すると、SKSTATUS コマンドのレスポンス値の bit0 が 1 になります。SKSTATUS コマンドのレスポンス値の bit0 が 1 の状態で、SKREAD/SKWRITE コマンドを発行してください。

## 対応機種

---

REX80

# SKSTATUS(ソケットステータス取得)

## 動作

---

ソケットの状態を取得します。

## コマンド

---

SKSTATUS[20H]ソケット識別子[0DH]

ソケット識別子	ソケット識別子を指定します。
---------	----------------

## レスポンス

---

ステータス[0DH]

が返ってきます。  
ステータスは10進数の値です。

## 備考

---

ステータス値は下記の意味があります。

Bit0	接続状態	0	接続処理中
		1	接続完了
Bit1	切断状態	0	未切断
		1	切断された
Bit2	受信データ状態	0	受信データなし
		1	受信データあり

## 対応機種

---

REX80

# SKWRITE(ソケットデータ出力)

---

## 動作

---

接続したソケットヘデータを出力します。

## コマンド

---

SKWRITE[20H]ソケット識別子[0DH]出力データ[03H]

ソケット識別子	接続済のソケット識別子を指定します。
出力データ	出力するデータを指定します。

## レスポンス

---

処理に成功すると、  
OK[0DH]  
が返ってきます。  
失敗した場合は  
ERROR[0DH]  
が返ってきます。

## 対応機種

---

REX80

# SKREAD(ソケットデータ入力)

## 動作

---

接続したソケットからデータを入力します。

## コマンド

---

SKREAD[20H]ソケット識別子[20H]サイズ[0DH]

ソケット識別子	接続済のソケット識別子を指定します。
サイズ	入力するデータサイズを 1~512 で指定します。

## レスポンス

---

処理が成功した場合は、

読み込めたサイズ[20H]チェックサム[0DH]  
読み込んだデータ

が返ってきます。

読み込んだデータの最後に [0DH] は付加されません。

失敗した場合は

ERROR[0DH]

が返ってきます。

## 対応機種

---

REX80

# SKCLOSE(ソケット切断)

---

## 動作

---

ソケットじます。

## コマンド

---

SKCLOSE[20H]ソケット識別子[0DH]

ソケット識別子	ソケット識別子を指定します。
---------	----------------

## レスポンス

---

処理が成功した場合は、

OK[0DH]

が返ってきます。  
失敗した場合は

ERROR[0DH]

が返ってきます。

## 対応機種

---

REX80

# SKTIMEOUT(ソケット タイムアウト時間設定)

---

## 動作

---

ソケット処理のタイムアウト時間を設定します。

## コマンド

---

SKTIMEOUT`[20H]`タイムアウト時間`[0DH]`

タイムアウト時間	ソケット処理のタイムアウト時間をミリ秒単位で指定します。 REX80 の起動後のタイムアウト時間は 60000 ミリ秒です。
----------	---

## レスポンス

---

処理が成功した場合は、

OK`[0DH]`

が返ってきます。  
失敗した場合は

ERROR`[0DH]`

が返ってきます。

## 対応機種

---

REX80

## BIN(バイナリモードへ移行)

### 動作

---

バイナリモードへ移行します。

### コマンド

---

BIN`[ODH]`

パラメータはありません

### レスポンス

---

レスポンスはありません。

### 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

## BINNEW(新バイナリモードへ移行)

---

### 動作

---

新バイナリモードへ移行します。

### コマンド

---

`BINNEW``ODH`

パラメータはありません

### レスポンス

---

レスポンスはありません。

### 対応機種

---

REX50 (ver5.10 以降), REX60 (ver5.10 以降), REX70, REX80

# バイナリモード

バイトデータでコマンドを送信するモードです。テキストモードで BIN`ODH` を送信するとこのモードに移行します。

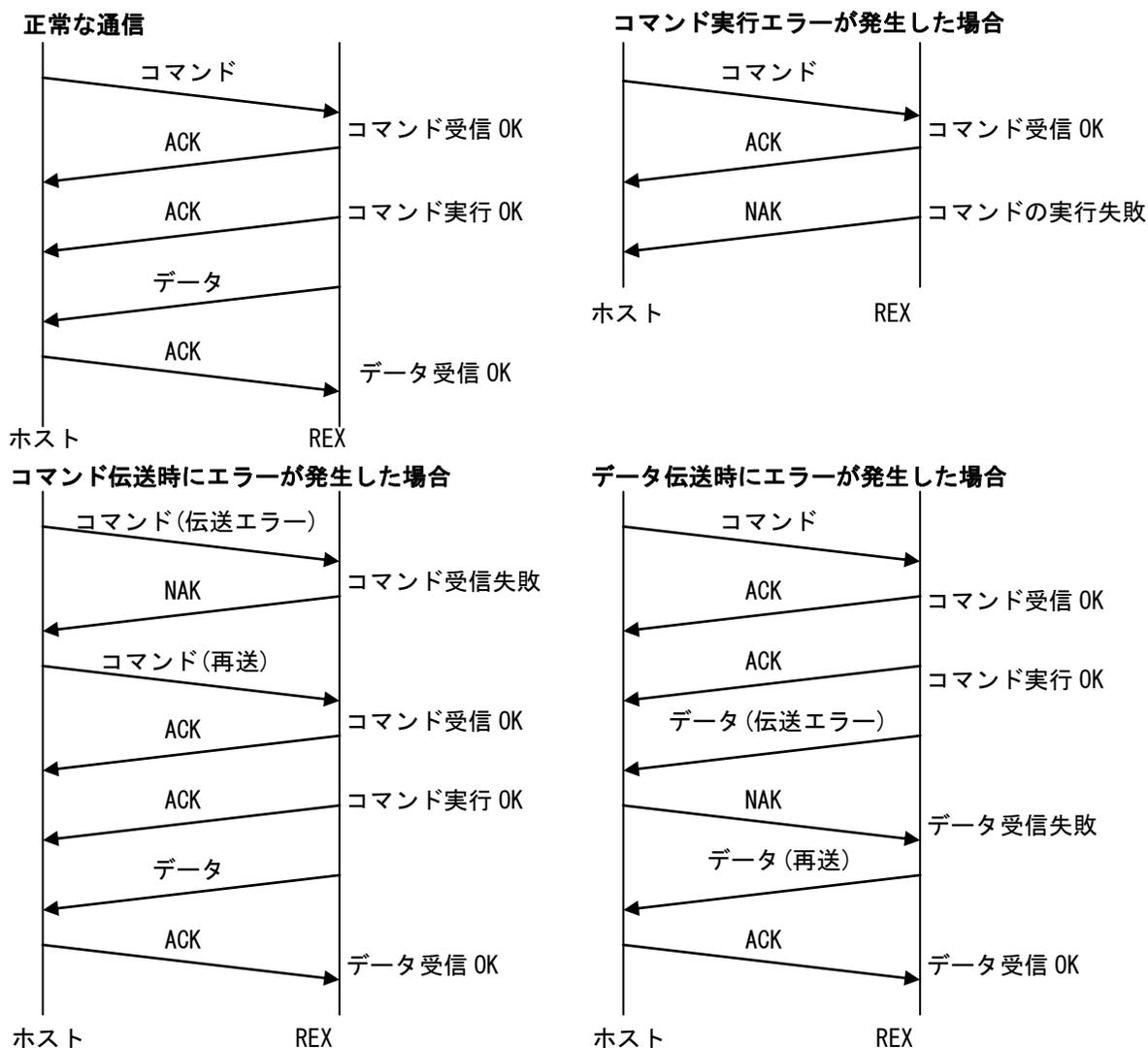
## 調歩同期シリアルタイプ (REX2, REX3U, REX4, REX50, REX60)

データ送受信中にブレーク信号を検出すると、データの送受信を中断し、コマンド受信待機状態に移行します。ただし、コマンド実行中にブレーク信号を検出してもコマンド動作自体は中断されません。コマンド実行中にブレーク信号を検出した場合は、コマンド動作が完了後にレスポンスを返さずコマンド受信待機状態に移行します。

## 同期シリアルタイプ (REX3S, REX50)

データ送受信中に -CS 信号を High レベルに変化させると、データ送受信をを中断し、コマンド受信待機状態に移行します。コマンドの最終バイトを送受信後すぐに -CS 信号を High レベルに変化させると、コマンドが実行されない場合がありますので注意してください。確実にコマンドを実行させたい場合は、レスポンスを受信した後、-CS 信号を High レベルに変化させてください。

## バイナリモードプロトコル



## コマンド

ホストから REX へ送られるコマンドデータです。1 バイトのコマンドと、パラメータ(コマンドによって長さが異なる)、チェックサムの組み合わせです。チェックサムはコマンドとパラメータのバイト単位で合計した値をビット反転したものです。

`コマンド番号(1byte)` `パラメータ(?byte)` `SUM(1byte)`

## コマンド受信応答

REX からホストへ送られるデータです。コマンドを正常に受信できた場合は ACK(70H) が返ります。コマンドが正常に受信できなかった場合は NAK(71H) が返ります。

## コマンド実行応答

REX からホストへ送られるデータです。コマンドを正常に実行できた場合は ACK(70H) が返ります。コマンドの実行に失敗した場合は NAK(71H) が返ります。処理に時間がかかるコマンドを実行した場合は、コマンド実行応答がされるまでに時間がかかる場合があります。

## データ

REX からホストへ送られるデータです。コマンドの実行によって生成されたデータが送られます。送られるデータのサイズはコマンドによって異なります。送るべきデータがない場合はデータとデータ受信応答は省略されます。

`データ(?byte)` `SUM(1byte)`

## データ受信応答

ホストから REX へ送るデータです。データを正常に受信できた場合は、ACK(70H) を返してください。NAK(71H) を返すと REX はデータを再送します。CAN(72H) を送信するとデータフェーズを中止します。

# 新バイナリモード

バイトデータでコマンドを送信するモードです。テキストモードで BINNEW`ODH`を送信するとこのモードに移行します。

## 調歩同期シリアルタイプ (REX50, REX60)

データ送受信中にブレーク信号を検出すると、データの送受信を中断し、受信待機状態に移行します。ただし、コマンド実行中にブレーク信号を検出してもコマンド動作自体は中断されません。コマンド実行中にブレーク信号を検出した場合は、コマンド動作が完了後に受信待機状態に移行します。

## 同期シリアルタイプ (REX50)

データ送受信中に-CS 信号を High レベルに変化させると、データ送受信をを中断し、受信待機状態に移行します。

# 新バイナリモードと旧バイナリモードの違い

新バイナリモードは、コマンド受信、コマンド実行、データ受信の3つのシーケンスに分けられます。コマンドの実行完了をホスト機器が CHECK (76h) コードの応答でチェックするようになります。

REX は最上位ビットが 1 のデータを受信すると、コマンドと識別し、指定されたコマンドのパラメータ長のデータを受信します。コマンドは次のような書式になります。

`コマンド番号 (1byte)` `パラメータ (?byte)` `SUM (1byte)`

コマンドの受信が完了すると、REX はチェックサムを比較して、正常に受信できた場合は ACK (70h) コードを、正常に受信できなかった場合は NAK (71h) コードをホスト機器へ送ります。REX は、コマンドの受信が完了しても、CHECK (76h) コードが送られてくるまでコマンドの実行は行いません。

コマンド受信完了後、CHECK (76h) コードを受信すると、受信したコマンドを実行し、RUN (73h) コードをホスト機器へ送信します。コマンド受信完了後、2 回目以降の CHECK (76h) コードの受信には、現在のコマンド実行状態を送り返します。ホスト機器は CHECK (76h) コードを送信して、コマンドの実行の完了をポーリング (10msec 程度の間隔) でチェックしてください。

NAK (71h) = コマンドを受信していない

RUN (73h) = 実行中

OK (74h) = 実行完了

NG (75h) = 実行失敗

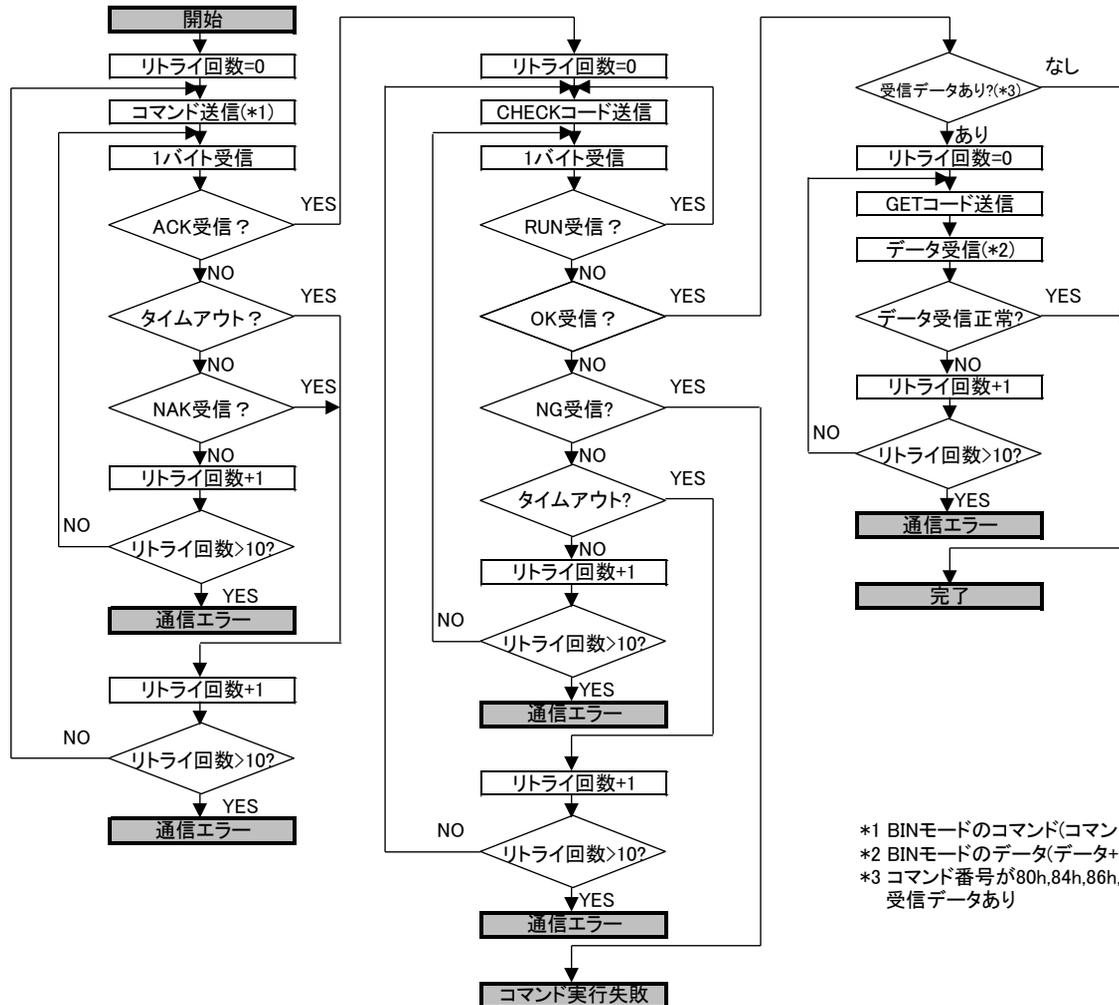
コマンドの実行が完了した後、受信データがある時は、GET (77h) コードを REX へ送信してください。コマンドの実行結果とチェックサムが返ってきます。次のような書式になります。

`データ (?byte)` `SUM (1byte)`

受信したデータのチェックサムを比較し、異常がある場合は、再び GET (77h) コードを REX へ送信してください。

CHECK (76h) コードの応答が NG (75h) コードの時は、コマンドの実行に失敗しています。失敗の原因を知るためには、ERROR (8Fh) コマンドでエラーコードを取得してください。

# 新バイナリモードフロー



ACKコード	70h
NAKコード	71h
RUNコード	73h
OKコード	74h
NGコード	75h
CHECKコード	76h
GETコード	77h

タイムアウト時間100ms

- \*1 BINモードのコマンド(コマンド番号+パラメータ+チェックサム)と同じ形:
- \*2 BINモードのデータ(データ+チェックサム)と同じ形式
- \*3 コマンド番号が80h,84h,86h,88h,8Eh,8Fh,90h,98h,9Bh,9Dhの時、受信データあり

# バイナリモード/新バイナリモードコマンドフォーマット

## 80H GETTIME(時刻の取得)

### 動作

---

RTC の日時を取得します。

### コマンド

---

`80H` `SUM`

パラメータはありません。

### データ

---

`年(2byte)` `月(1byte)` `日(1byte)` `時(1byte)` `分(1byte)` `秒(1byte)` `SUM`

年	現在の年を返します (1980~2079)
月	現在の月を返します (01~12)
日	現在の日を返します (01~31)
時	現在の時を返します (00~23)
分	現在の分を返します (00~59)
秒	現在の秒を返します (00~59)

日時を設定していない場合は 1980/01/01 00:00:00 を返します。

### 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

## 81H SETTIME(時刻の設定)

### 動作

---

RTCの日時を設定します。

### コマンド

---

81H 年(2byte) 月(1byte) 日(1byte) 時(1byte) 分(1byte) 秒(1byte) SUM

年	設定する年を指定します(1980~2079)
月	設定する月を指定します(01~12)
日	設定する日を指定します(01~31)
時	設定する時を指定します(00~23)
分	設定する分を指定します(00~59)
秒	設定する秒を指定します(00~59)

### データ

---

データはありません。

### 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

## 82H MOUNT(メディアのマウント)

### 動作

---

メディアのフォーマットをチェックし、メディアへのアクセスの準備をします。

### コマンド

---

`82H` `SUM`

パラメータはありません。

### データ

---

データはありません。

### 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70

REX80 では MOUNTD (00h) を使用してください。

## 83H UNMOUNT(メディアのアンマウント)

### 動作

---

メディアのアクセス処理を終了します。

### コマンド

---

`83H` `SUM`

パラメータはありません。

### データ

---

データはありません。

### 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70

REX80 では UNMOUNTD(00h) を使用してください。

## 84H OPEN(ファイルを開く)

### 動作

メディア上にあるファイルをオープンします。

### コマンド

**84H** **モード(1byte)** **ファイルパス長さ(1byte)** **ファイルパス** **SUM**

モード	アクセスモードと作成モードを論理和した値を指定します。		
	bit7-3	作成モードを指定します。ディレクトリ読みモード時は無効です。	
		CREATE_NEW(0)	ファイルを新規作成します。ファイルが既に存在する場合は、エラーになります。
		CREATE_ALWAYS(1)	ファイルを常に新規作成します。ファイルが既に存在する場合は、上書きされます。
		OPEN_EXISTING(2)	ファイルを開きます。ファイルが存在しない場合はエラーになります。
		OPEN_ALWAYS(3)	ファイルを開きます。ファイルが存在しない場合は新規作成されます。
	bit2-0	アクセスモードを指定します。	
		READ(1)	読み専用モードでオープンします。
		WRITE(2)	書き専用モードでオープンします。
READWRITE(3)		読み書きモードでオープンします。	
DIR(4)	ディレクトリ読みモードでオープンします。		
ファイルパス長さ	ファイルパスの文字列長さを指定します(1~255)		
ファイルパス	オープンするファイルのフルパスを指定します。 ディレクトリ読み込み込む場合はディレクトリを指定してください。		

### データ

オープン処理が成功した場合は、  
**ファイル識別子(1byte)** **SUM**  
が返ってきます。

### 備考

既に2つのファイルが開かれている場合は、新しくファイルをオープンすることはできません。  
ディレクトリ読み込みを指定すると、指定されたディレクトリのリストを READ コマンドで取得することができます。

### 対応機種

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

## 85H CLOSE(ファイルを閉じる)

### 動作

---

オープンされたファイルをクローズします。

### コマンド

---

`85H` `ファイル識別子(1byte)` `SUM`

ファイル識別子	クローズしたいファイルのファイル識別子を指定します。
---------	----------------------------

### データ

---

データはありません。

### 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

## 86H READ(ファイルから読み込む)

### 動作

---

オープンしたファイルのデータを読み込みます。

### コマンド

---

`86H` `ファイル識別子(1byte)` `サイズ(2byte)` `SUM`

ファイル識別子	読み込みたいファイルの識別子を指定します。
サイズ	バイト単位で読み込むサイズを指定します(1~512)

### データ

---

読み込み処理が成功した場合は、

`読み込んだファイルサイズ(2byte)` `読み込んだデータ` `SUM`

が返ってきます。

### 備考

---

読み込みオープンしたファイル以外を READ コマンドで読み込もうとするとエラーになります。一度に 513 バイト以上のデータを読み込むことはできません。

読み込み後は、読み込んだサイズだけ読み書き位置が移動されます。

ディレクトリ読み込みでオープンした場合は、サイズに 22 を指定してください。返ってくるデータは以下のような形式になります。

オフセット	サイズ(byte)	内容
0	13	ファイル名
13	1	属性
14	2	更新時刻
16	2	更新日付
18	4	サイズ

### 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

## 87H WRITE(ファイルへ書き込む)

### 動作

---

オープンしたファイルへデータを書き込みます。

### コマンド

---

`87H` `ファイル識別子(1byte)` `サイズ(2byte)` `データ` `SUM`

ファイル識別子	書き込みたいファイルの識別子を指定します。
サイズ	書き込むデータの長さを指定します(1~512)
データ	書き込むデータを指定します。 一度に書き込めるデータサイズは 512 バイトまでです。

### データ

---

データはありません。

### 備考

---

書き込みオープンしたファイル以外を WRITE コマンドで書き込もうとするとエラーになります。  
一度に 513 バイト以上のデータを書き込むことはできません。

書き込み後は、書き込んだサイズだけ読み書き位置が移動されます。

### 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

## 88H SEEK(読み書き位置の移動)

### 動作

---

オープンしたファイルのデータ読み込み位置を移動します

### コマンド

---

`88H` `ファイル識別子(1byte)` `移動量(4byte)` `基準位置(1byte)` `SUM`

ファイル識別子	ファイルの識別子を指定します。
移動量	基準位置からどれだけ移動するかを数値で指定します。
基準位置	移動基準位置を指定します。 0 ファイルの先頭 1 現在の読み書き位置 2 ファイルの終端

### データ

---

移動処理が成功した場合は、

`移動後の位置(4byte)` `SUM`

が返ってきます。

### 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

## 89H DELETE(ファイルの削除)

### 動作

---

指定されたファイルを削除します。

### コマンド

---

ファイルパス長さ	ファイルパスの長さ(1~255)
ファイルパス	削除したいファイルのフルパスを指定します(最大 255 文字)

### データ

---

データはありません。

### 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

## 8AH MOVE(ファイルの移動)

### 動作

---

指定されたファイルを移動(ファイル名の変更)します。

### コマンド

---

`8AH` `移動元長さ(1byte)` `移動元ファイルパス` `移動先長さ(1byte)` `移動先ファイルパス` `SUM`

移動元長さ	移動元ファイルパスの長さ(1~255)
移動元ファイルパス	移動したいファイルのフルパスを指定します(最大 255 文字)
移動先長さ	移動先ファイルパスの長さ(1~255)
移動先ファイルパス	移動先のフルパスを指定します(最大 255 文字)

### データ

---

データはありません。

### 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

## 8BH MKDIR(ディレクトリの作成)

### 動作

---

指定されたディレクトリを作成します。

### コマンド

---

8BH  ディレクトリ名長さ(1byte)  ディレクトリ名  SUM

ディレクトリ名長さ	ディレクトリ名の長さ(1~255)
ディレクトリ名	作成したいディレクトリのフルパスを指定します(最大 255 文字)

### データ

---

データはありません。

### 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

## 8CH RMDIR(ディレクトリの削除)

### 動作

---

指定されたディレクトリを削除します。

### コマンド

---

`8CH` `ディレクトリ名長さ(1byte)` `ディレクトリ名` `SUM`

ディレクトリ名長さ	ディレクトリ名の長さ(1~255)
ディレクトリ名	削除したいディレクトリのフルパスを指定します(最大 255 文字)

### データ

---

データはありません

### 備考

---

削除しようとするディレクトリの中にファイルやディレクトリが存在するとエラーになります。

### 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

## 8DH FORMAT(フォーマット)

### 動作

---

メディアをフォーマットします。

### コマンド

---

`8DH` `SUM`

パラメータはありません

### データ

---

データはありません。

### 備考

---

MOUNT コマンドを実行しなくてもフォーマットすることができます。  
フォーマット後は UNMOUNT 状態になります。

### 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70

## 8EH GETFREE(ディスク容量の取得)

### 動作

---

メディアの総容量と、空き容量を取得します。

### コマンド

---

`8EH` `SUM`

パラメータはありません

### レスポンス

---

処理が成功した場合は、

`空き容量(4byte)` `総容量(4byte)` `SUM`

が返ってきます。

空き容量、総容量が 2G バイト以上の時は 7FFFFFF0h が返ってきます (ver5.13 以降の場合)。

### 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70

## 8FH ERROR(エラーコードの取得)

### 動作

---

最後に発生したエラーのエラーコードを取得します。

### コマンド

---

`8FH` `SUM`

パラメータはありません

### データ

---

`エラーコード(1byte)` `SUM`

が返ってきます。

### 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

## 90H VERSION(バージョン取得)

### 動作

---

ファームウェアのバージョンを取得します。

### コマンド

---

`90H` `SUM`

パラメータはありません

### データ

---

`メジャー番号(1byte)` `マイナー番号(1byte)` `SUM`

が返ってきます。

### 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60, REX70, REX80

# 91H CHARGE(RTC バックアップ電源充電設定)

## 動作

---

RTCバックアップ電源の充電モードを設定します。

## コマンド

---

91H モード(1byte) SUM

充電モード	充電モードを設定します。 0 充電禁止 1 最大 1.3mA で充電 2 最大 650uA で充電 3 最大 325uA で充電
-------	--

## データ

---

データはありません。

## 備考

---

初期値は0(充電禁止)です。  
バックアップ電源電圧が2.0V未滿になり、設定が保持できなくなった場合は0(充電禁止)に初期化されます。

## 対応機種

---

REX2, REX3U, REX3S, REX4, REX45, REX50, REX60

## 98H STATUS(メディア状態の取得)

### 動作

---

現在のメディアの状態を取得します。

### コマンド

---

98H SUM

パラメータはありません

### データ

---

メディアステータス(1byte) SUM

が返ってきます。

### 備考

---

メディアステータスの bit0 (LSB) が 0 の時は、REX のソケットにメディアが挿入されていません。  
メディアステータスの bit0 (LSB) が 1 の時は、REX のソケットにメディアが挿入されています。

メディアステータスの bit1 が 0 の時は、MOUNT されていない状態です。  
メディアステータスの bit1 が 1 の時は、MOUNT された状態です。

REX60 ver5.11 以降、REX70 では、bit7 が USB メモリの挿抜状態を示します。bit7 が 1 の時は、REX に USB メモリが挿入されており、bit7 が 0 の時は USB メモリは挿入されていません。bit0 は USB メモリが使用可能になった時に 1 になります。

ステータスの bit2 はライトプロテクトの状態を示します。0 の時は書き込み許可、1 の時は書き込み禁止状態です。

REX80 では bit7 が USB モデムの挿抜状態を示しています。bit7 が 1 の時は、REX に USB モデムが挿入されており、bit7 が 0 の時、USB モデムは挿入されていません。Bit0 は PPP 接続状態を示しており、0 のときは未接続、1 のときは接続中を示しています。bit1 はドライブ 0 の MOUNT 状態、bit2 はドライブ 1 の MOUNT 状態のように bit1~6 はドライブ 0~5 の MOUNT 状態を示します。

### 対応機種

---

REX2 (ver1.08 以降), REX3U (ver2.08 以降), REX3S (ver2.08 以降), REX4 (ver3.08 以降), REX45 (ver3.08 以降), REX50, REX60, REX70, REX80

## 99H AUTOMOUNT(自動マウントモードの設定)

### 動作

---

オートマウント機能の有効/無効を設定します。

### コマンド

---

99H 設定値 SUM

設定値	オートマウントの有効/無効を指定します。
-----	----------------------

### データ

---

データはありません。

### 備考

---

設定値に1を指定すると、オートマウントが有効になります。  
オートマウントが有効になると、メディアの挿入時に自動で MOUNT コマンドを実行します。

設定値に0を指定すると、オートマウントは無効になります。

電源投入直後は、オートマウント無効状態になります。

### 対応機種

---

REX50 (ver5.03 以降), REX60, REX70

## 9AH PORTOUT(汎用ポート出力設定)

### 動作

---

基板上の P10, P9 ポートの出力信号を制御します。

### コマンド

---

`9AH` `ポート番号` `出力値` `SUM`

ポート番号	設定を行うポートの番号を指定します。
出力値	ポートに出力する値を指定します。 0 = Low レベル 1 = High レベル 2 = Hi-Z 3 = ステータスの bit0 4 = ステータスの bit1

### データ

---

データはありません。

### 備考

---

出力値に 0 または 1 を指定すると、High または Low レベルがポートに出力されます。  
出力値に 2 を指定すると、ポートはハイインピーダンスになり、入力ポートとして使用できます。  
出力値に 3 を指定すると、STATUS コマンドで得られる値の bit0 の値がポートへ出力されます。  
出力値に 4 を指定すると、STATUS コマンドで得られる値の bit1 の値がポートへ出力されます。  
出力値に 3 または 4 を指定した場合は、最新の STATUS 値が常にポートへ出力されます。

電源投入後、全ポートはハイインピーダンスになります。

### 対応機種

---

REX50 (ver5.03 以降), REX60

## 9BH PORTIN(汎用ポートの状態を得る)

---

### 動作

---

基板上の P10, P9 ポートの信号を読み取ります。

### コマンド

---

ポート番号	取得するポート番号を指定します。
-------	------------------

### データ

---

が返ってきます。  
指定されたポートが High レベルの場合は 1 が、Low レベルの時は 0 が値になります。

### 備考

---

PORTOUT コマンドで Hi-Z 以外を指定しているポートも読み込みは可能です。

### 対応機種

---

REX50 (ver5.03 以降), REX60

## 9CH SETFLOW(シリアル通信のフロー設定)

### 動作

---

シリアル通信のフロー制御の設定を行います。

### コマンド

---

**9CH** **受信制御** **送信制御** **SUM**

受信制御	受信フロー制御信号(P2の4ピン)の動作を指定します。 0 = 制御しない 1 = 受信が出来ない時に High レベルを出力します。 2 = 受信が出来ない時に Low レベルを出力します。
送信制御	送信フロー制御信号(P2の6ピン)の動作を指定します。 0 = 信号を無視します。 1 = High レベルの時は、データの送信を行いません。 2 = Low レベルの時は、データの送信を行いません。

### データ

---

データはありません。

### 備考

---

電源投入直後は、受信フロー=2、送信フロー=0 の状態です。

### 対応機種

---

REX50(ver5.06以降), REX60, REX70, REX80

## 9DH WRITEN(ファイルへ書き込む)

### 動作

---

オープンしたファイルへデータを書き込みます。

### コマンド

---

`9DH` `ファイル識別子(1byte)` `サイズ(2byte)` `データ` `SUM`

ファイル識別子	書き込みたいファイルの識別子を指定します。
サイズ	書き込むデータの長さを指定します(1~512)
データ	書き込むデータを指定します。 一度に書き込めるデータサイズは 512 バイトまでです。

### データ

---

`書き込んだデータサイズ(2byte)` `SUM`

が返ってきます。

### 備考

---

書き込みオープンしたファイル以外を WRITE コマンドで書き込もうとするとエラーになります。  
一度に 513 バイト以上のデータを書き込むことはできません。

書き込み後は、書き込んだサイズだけ読み書き位置が移動されます。

コマンドの実行が成功すると、ファイルへ書き込んだデータの長さが返ってきます。

### 対応機種

---

REX50(ver5.06以降), REX60, REX70, REX80

## 9EH TEXT(テキストモードへ戻る)

### 動作

---

テキストモードへ戻ります。

### コマンド

---

**9EH** **SUM**

### データ

---

データはありません。

### 備考

---

バイナリモード/新バイナリモードからテキストモードへ戻ります。

### 対応機種

---

REX50(ver5.10以降), REX60(ver5.10以降), REX70, REX80

## 9FH GETFREESEC(ディスク容量の取得)

### 動作

---

メディアの総容量と、空き容量を 512 バイト単位で取得します。

### コマンド

---

`9FH` `SUM`

パラメータはありません

### データ

---

処理が成功した場合は、

`空き容量(4byte)` `総容量(4byte)` `SUM`

が返ってきます。

空き容量、総容量は 512 倍した値が実際のバイト単位の容量になります。

### 対応機種

---

REX50 (ver5. 13 以降) , REX60 (ver5. 13 以降) , REX70

## A7H GETMOUSE(マウス状態を得る)

### 動作

---

USB ポートに接続されたマウスの座標とボタン状態を得ます。

### コマンド

---

A7H SUM

パラメータはありません

### データ

---

処理が成功した場合は、

X座標(4byte) Y座標(4byte) 予約(4byte) ボタン状態(4byte) SUM

が返ってきます。

X 座標、Y 座標は起動時からの積算値になります。

ボタン状態は各ビットが各ボタンに対応しており、一般的なマウスの場合は、

Bit0 左ボタン

Bit1 右ボタン

Bit2 ホイールボタン

となっています。

### 対応機種

---

REX70

## A8H CLRKEYC(キー入力バッファのクリア)

### 動作

---

USB ポートに接続されたキーボード用の入力キャラクタバッファをクリアします。

### コマンド

---

**A8H** **SUM**

パラメータはありません

### データ

---

データはありません。

### 対応機種

---

REX70

## A9H GETKEYC(キー入力データの取得)

---

### 動作

---

USB ポートに接続されたキーボード用の入力キャラクタバッファから 1 文字取得します。

### コマンド

---

**A9H** **SUM**

パラメータはありません

### データ

---

処理が成功した場合は、

**キャラクタコード(1byte)** **SUM**

が返ってきます。

入力データが無い場合は、0 が返ってきます。

### 対応機種

---

REX70

## ABH CLRKEYE(キー入カイベントバッファのクリア)

---

### 動作

---

USB ポートに接続されたキーボード用の入カイベントバッファをクリアします。

### コマンド

---

**ABH** **SUM**

パラメータはありません

### データ

---

データはありません。

### 対応機種

---

REX70

# ACH GETKEYE(キー入カイベントデータの取得)

## 動作

---

USB ポートに接続されたキーボード用の入カイベントバッファから 1 個のデータを取得します。

## コマンド

---

ACH SUM

パラメータはありません

## データ

---

処理が成功した場合は、

イベントコード(2byte) SUM

が返ってきます。データが無い場合は、0 が返ってきます。

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
予約 (常に 0)							押下状態 0=Up 1=Down	キーコード (4~255)							

下位 8bit はキーコードを示しています。

bit8 はキーの状態で、Bit8 が 0 の時はキーが離された、1 の時はキーが押されたことを示します。

## 対応機種

---

REX70

# AEH GETJOYSTICK(ジョイスティック状態取得)

## 動作

---

USB ポートに接続されたジョイスティックの状態を取得します。

## コマンド

---

`AEH 00h SUM`

パラメータには 00h を指定してください

## データ

---

処理が成功した場合は、

`左X軸量(2byte) 左Y軸量(2byte) 右X軸量(2byte) 右Y軸量(2byte) 左トリガ量(2byte)`  
`右トリガ量(2byte) ボタン状態(4byte) SUM`

が返ってきます。

左右スティックの各軸量は符号付き 16bit の値を返します。左右トリガは符号無し 16bit の値を返します。ボタン状態の値は、押されたボタンに対応するビットが 1 になります。

## 対応機種

---

REX70

## B9H SLEEP(スリープ)

---

### 動作

---

REX をスリープさせます。

### コマンド

---

**B9H** **SUM**

### データ

---

データはありません。

### 備考

---

コマンド実行応答で ACK を返した後に、REX はスリープモードに入ります。  
P9-15 番ピンに LOW レベルを 5msec 以上入力すると、スリープモードから復帰します。

スリープから復帰して 100msec 経過後にシリアルポートが有効になりますので、  
スリープ復帰後すぐにコマンドを送られると、正常に通信できない場合があります。

### 対応機種

---

REX50 (ver5.08 以降)

## エラーコード

0	エラーなし
1	5V メディアには対応していない
2	カードが挿入されていない
3	読み込みエラー
4	非対応のフォーマット
5	非対応のメディア
6	ファイルが見つからない
7	空がない
8	書き込みエラー
9	マウントされていない
10	書き込み禁止
11	読み込み専用
12	ディレクトリが空ではない
13	ファイルが既に存在する
14	無効なファイル名
15	内部エラー(無効なクラスタ番号)
64	無効なオープンモード
65	同時オープン可能数を超えた
66	無効なファイル識別子
67	アクセスできない
68	RTCの準備ができていない(REX70)
255	未定義のコマンドを実行した

# 汎用ポートについて

## REX50

基板上の汎用ポートの位置と、コマンドで指定する番号の対応は次のようになります。

ポート番号	基板ポート位置	備考
0	P10-1 番ピン	
1	P10-2 番ピン	
2	P10-3 番ピン	
3	P10-4 番ピン	
4	P10-5 番ピン	
5	P10-6 番ピン	
6	P10-7 番ピン	
7	P10-8 番ピン	
8	P10-9 番ピン	
9	P10-10 番ピン	
10	P10-11 番ピン	
11	P10-12 番ピン	
12	P10-13 番ピン	
13	P10-14 番ピン	
14	P10-15 番ピン	
15	P10-16 番ピン	
16	P9-5 番ピン	プルダウンされています
17	P9-6 番ピン	N チャンネルオープンドレイン出力
18	P9-7 番ピン	N チャンネルオープンドレイン出力
19	P9-8 番ピン	
20	P9-9 番ピン	
21	P9-10 番ピン	
22	P9-11 番ピン	
23	P9-12 番ピン	
24	P9-13 番ピン	
25	P9-14 番ピン	

汎用ポートには 0[V]～VCC[V]の電圧を入力できます。  
範囲外の電圧が加えられると、コントローラが破壊されるおそれがあります。

汎用ポートに電流を流しすぎた場合、コントローラが破壊されるおそれがあります。  
1ポート辺り 5mA 以上の電流を流さないようにしてください。

表記していない空きポートには、何も接続しないでください。  
誤動作やコントローラの破壊の原因になります。

P9-6, P9-7 ピンは、オープンドレイン出力になりますので、外部にプルアップ抵抗等が必要です。

## REX60

基板上の汎用ポートの位置と、コマンドで指定する番号の対応は次のようになります。

ポート番号	基板ポート位置	備考
0	P10-1 番ピン	
1	P10-2 番ピン	
2	P10-3 番ピン	
3	P10-4 番ピン	
4	P10-5 番ピン	
5	P10-6 番ピン	
6	P10-7 番ピン	
7	P10-8 番ピン	
8	P10-9 番ピン	
9	P10-10 番ピン	
10	P10-11 番ピン	
11	P10-12 番ピン	
12	P10-13 番ピン	
13	P10-14 番ピン	
14	P10-15 番ピン	
15	P10-16 番ピン	
16	P9-5 番ピン	プルダウンされています
17	P9-6 番ピン	N チャンネルオープンドレイン出力
18	P9-7 番ピン	N チャンネルオープンドレイン出力
19	P9-8 番ピン	
20	P9-9 番ピン	
21	P9-10 番ピン	
22	P9-11 番ピン	
23	P9-12 番ピン	
24	P9-13 番ピン	
25	P9-14 番ピン	

汎用ポートには 0[V]～VCC[V]の電圧を入力できます。  
範囲外の電圧が加えられると、コントローラが破壊されるおそれがあります。

汎用ポートに電流を流しすぎた場合、コントローラが破壊されるおそれがあります。  
1 ポート辺り 5mA 以上の電流を流さないようにしてください。

表記していない空きポートには、何も接続しないでください。  
誤動作やコントローラの破壊の原因になります。

P9-6, P9-7 ピンは、オープンドレイン出力になりますので、外部にプルアップ抵抗等が必要です。

# USB キーボードコード表

## キーコード

上位 下位	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0		M も	#あ 3あ	{「 [°	F7	←	8 ↑									左 Ctrl	
1		N み	\$う 4う		F8	↓	9 PgUp									左 Shift	
2		O ら	%え 5え	]」 ]む	F9	↑	0 Ins									左 Alt	
3		P せ	&お 6お	+」 :れ	F10	NumLock	. Del									左 GUI	
4	A ち	Q た	`や 7や	*ヶ :け	F11	/										右 Ctrl	
5	B こ	R す	(ゆ 8ゆ	半角/ 全角	F12	*	APP									右 Shift	
6	C そ	S と	)よ 9よ	<、 .ね	Print Screen	-										右 Alt	
7	D し	T か	~を 0わ	>。 .る	Scroll Lock	+	=		—  \ろ							右 GUI	
8	E い	U な	Enter	?・ /め	Pause	Enter	F13		カタカナ ひらがな								
9	F は	V ひ	Esc	CapsLock	Insert	1 End	F14		— ¥—								
A	G き	W て	Back space	F1	Home	2 ↓	F15		変換								
B	H く	X さ	TAB	F2	Page Up	3 PgDn			無変換								
C	I に	Y ん	SPACE	F3	Delete	4 ←											
D	J ま	Z つ	=&#x2192; -ぼ	F4	End	5											
E	K の	!ぬ	—々 ^へ	F5	Page Down	6 →											
F	L り	“ 2ふ	`¢ @°	F6	→	7 Home											

## キャラクタコード

上位 下位	0	1	2	3	4	5	6	7
0			SP	0	@	P	`	p
1			!	1	A	Q	a	q
2			“	2	B	R	b	r
3			#	3	C	S	c	s
4			\$	4	D	T	d	t
5			%	5	E	U	e	u
6			&	6	F	V	f	v
7			‘	7	G	W	g	w
8	BS		(	8	H	X	h	x
9	TAB		)	9	I	Y	i	y
A			*	:	J	Z	j	z
B		ESC	+	;	K	[	k	{
C			,	<	L	¥	l	
D	Enter		-	=	M	]	m	}
E			.	>	N	^	n	~
F			/	?	O	_	o	

## ジョイスティックボタン割り当て

ビット	ボタン
0	UP
1	DOWN
2	LEFT
3	RIGHT
4	A, ×
5	B, ○
6	X, □
7	Y, △
8	LB, L1
9	RB, R1
10	LT, L2
11	RT, R2
12	左スティック押下
13	右スティック押下
14	START
15	BACK, SELECT
16	CENTER
30	決定 (A, ○)
31	キャンセル (B, ×)

