

# HANA-BASIC 簡易マニュアル

## Ver. 1.0

03-Nov.-2011 (C)花岡ちゃんに花束を

### 1: 概要

パソコンとUSBで接続し、シリアル通信ソフト(Tera Term 推奨)で操作する Tiny-BASIC ボードマイコン。BASIC プログラムはパソコンのテキストエディタで作成し、通信ソフトでダウンロードする。

ダウンロードしたプログラムは不揮発性のEEPメモリに格納され、パソコンを接続しなくとも自動実行させることができるので、簡単な組み込みボードとしても使える。I/Oとして汎用のデジタル出力 8bit、分解能 10bit のA/D入力 5本がある。

電源はUSBから5Vが供給されるが、スイッチを切り替えると外部電源の使用も可能。電圧は3から5V、ボードだけ(I/Oになにもつながらない状態)なら電流は最大30mA程度。

### 2: TeraTerm の設定

設定→端末

- ・改行コード
  - 受信 CR+LF
  - 送信 CR
- ・ローカルエコー なし

設定→シリアルポート

- ・ボーレート 38400
- ・データ 8bit
- ・パリティ none
- ・ストップ 1bit
- ・フロー制御 none

<重要>

送信遅延を「0 ミリ秒/字 100 ミリ秒/行」に設定

インタプリタは1行ごとに中間言語に変換、EEPに書き込む処理時間が必要。そのためファイルをペーストしたときのように連続的に送信する場合、1行送信後次の行の送信開始までに時間をおく。

### 3: 動作テストとインディケータの説明

・動作テスト

PcとUSB接続し、TeraTermを起動し前節の設定を行う。

HANA-BASIC 基板のリセットボタンを押すとスタートアップメッセージとプロンプトの表示

```
*** HANA-BASIC ***  
Ver 1.0
```

```
?
```

print 1+2 と入力後リターン

```
print 1+2
3
?
```

以上のように動作すればまあ大丈夫

#### ・インディケータの説明

動作状態を表示。実行中ランプ (RUNNING) とエラー (ERROR) ランプがある。実装しなくとも問題ない。

両ランプともリセット時にテストのため一瞬点灯する。実行中ランプはパイロットランプとして点灯し、エラーは通常は消灯。リセット後、エラーが点灯したままになっている場合は、EEP の配線が間違っているか、PIC の 16 ピンのプルダウンがうまくいっていない可能性がある。

RUNNING LED

通電中点灯、BASIC プログラムを実行中点滅

ERROR LED

エラーメッセージを表示する時に点灯。コマンドモードで CR が入力されると消灯

消すためにはコマンドモードで CR をたたか、そのまま気にせずに作業を続けられよ。

## 4:トラブルシューティング

- ・スタートアップメッセージが出ない
  - リセットボタンを押してみる
  - 電源の確認
  - シリアルの確認 (ポート、ボーレートなど)
  - 未接続の /CTS が H になっていないか (配線チェック)
- ・スタートアップはするがプロンプト ? が出ない (ERROR LED が点灯したまま)
  - EEP まわりの配線不良
- ・プログラムをターミナルにペーストして送ると変だ
  - teraTerm の設定 → シリアルポート → 送信遅延 100 ミリ秒 / 行を確認。
  - そうなら、これを 200 ミリ秒 / 行にしてみる。

## 5:操作方法

### ●BASIC コマンドのダイレクト実行

コマンドモード (プロンプトが?) で BASIC コマンドを入力することで実行

```
?print 1+1
2
?a=10
?print a*a
100
```

?

GOTO SOSUB FOR などは実行できない

### 重要！

@0-@29 のスクラッチパッドはインタプリタの作業領域と重なるのでダイレクト実行するとおかしな値になる。あしからず。プログラム実行では問題ない。

## ●プログラム入力モード

コマンドモード (プロンプトが?) で load コマンド入力でこのモードに。プロンプトが>になる。

プログラムは常に最初から書き込まれる状態になる。NEW の状態。

間違えて入力モードにしたら、リターンを押す前に eof コマンドか CTRL-C がリセットボタンで抜けること。こうすれば元のプログラムに何の影響も与えない。

```
?load
```

```
>
```

この状態でプログラムを手打ちすると順に入力できる。編集機能は BS キーでカーソルを戻すことのみ。リターンキーを打つともう手直しは出来ない。

そのため、テキストエディタでプログラムを作っておき、コピペか「ファイル送信」でダウンロードする。

ファイルの転送が終わったら eof (End Of File) コマンドで抜ける。

```
>print "SAYONARA!"  
>end  
>eof
```

```
...
```

```
?
```

重要！ eof 入力後、ラベルの処理を行うので、数秒から数十秒待たされる。(ピリオド)がプログレスバーのように作業の進行状態に応じて表示される。(上の実行例参照)

プロンプト? が再表示されるまで、いかなるコマンドも入力できない。

### <プログラム送信のヒント>

送信する BASIC プログラムの前後に、下の例のように"load"と"eof"を入れておくと、load や eof を手打ちしなくてよい。

```
load  
for i=1 to 10  
print i  
next I  
end  
eof
```

## ●プログラムの手動実行

コマンドモード (プロンプトが?) で run コマンドを入力すると実行

実行中プログラムを停止したいときは CTRL+C を入力。

但し、INPUT 命令で入力待ち、DELAY 命令を実行中は反応しない。その場合は他の命令の実行中のタイミングでキー入力するか、リセットボタンで止める。

プログラム実行中は RUNNIG LED がせわしなく点滅して昔のコンピュータらしさを演出！

プログラムは EEPROM に書きこまれ、電源を切っても消えない。使いたいプログラムをあらかじめロードしておけば、次からは起動後すぐに run するだけで実行出来る。

## ●プログラムの自動実行

AUTO モードに設定しておく、リセット後ロードされているプログラムを自動的に実行。パソコンを接続する必要がないので、組み込みマイコンのように使える。

### ・自動実行モードに設定する

CTRL+A キーで自動実行モードになる。

?

```
AUTORUN MODE
```

### ・自動実行

リセットボタンを押すと、約5秒後にプログラムを開始する。別電源が供給されていれば USB を接続しなくても問題ない。

### ・自動実行モードを解除する

パソコンに接続し Tera Term を起動、リセットボタンを押すと、下記のような表示が出る。

```
*** HANA-BASIC ***
```

```
Ver 1.0
```

```
AUTORUN..
```

AUTORUN の後にピリオドが追加されている間に何かキーを押すと自動実行モードが解除され、プロンプト?が表示され待機状態になる。ピリオドが5個になるとモードは解除されず BASIC プログラムを自動実行する

## 6: 言語仕様

### ●モニタ

ターミナルからのコマンドやプログラム入力を受け付ける。BS キーで入力文字の訂正も出来る。

#### ■コマンドモード(プロンプトが"?")

- ・入力した BASIC コマンドをダイレクト実行
- ・run 入力モードで保存したプログラムを実行
- ・load プログラム入力モードに移行
- ・CTRL+C システムリセット
- ・CTRL+A AUTO モード(自動実行モード)セット

#### ■プログラム入力モード(プロンプトが">")

- ・入力したプログラムを一行ずつ保存(一行最大 80 文字)
- ・eof コマンドモードに移行

### ●変数、定数、式

変数 A-Z の 26 個。符号付き 16bit

定数 10 進定数の他、0x8000 のような 16 進定数、"...文字列..."は PRINT 文でのみ使用可。

算術演算 左から順に実行(数式どおりではない)。四則演算と、C 同様の論理演算(&, |, ~)

条件演算 左から順に実行(数式どおりではない)。C 同様(==, !=, <, >, =<, =>)

カッコは使えない。

## ●メモリ参照子@

@+定数または変数で配列(メモリ)などに16bitでアクセス

@0-@29 スクラッチパッドメモリ(1次元の配列のように使用できる)

@30-@34 ANALOG IN 読み込み(10bit 0-1023)\*回路図参照  
AD0<-@30, AD1<-@31, AD2<-@32, AD3<-@33, AD4<-@34

@35 キーコード読み込み(シリアル入力1文字読み込み、1バイト、叩いた文字のASCIIコードが入る)

@36 1バイトの乱数読み込み(0-255)

@40 DIGITAL OUT(D0-D7\*回路図参照)へのデータ出力(1バイト)

@100-@1000 EEP へのデータ読み書き。電源を切っても消えないのでデータファイルとして利用可。

## ●PRINT

ターミナルに出力。

```
PRINT "ANSWER=" ;
```

```
PRINT A
```

行末が ; ならば改行しない。

## ●INPUT

ターミナルから数値を入力。

```
INPUT A
```

数値(10進のみ)を入力しリターンキーを押すと変数に取り込む。

## ●IF～THEN

使用例:

```
IF A>10 THEN PRINT "OVER!"
```

条件成立(式の値が1以上なら)THEN以降を実行

## ●GOTO,GOSUB,RETURN,:(ラベル識別子)

使用例:

```
A=0
```

```
:L0
```

```
PRINT A
```

```
A=A+1
```

```
IF A<10 THEN GOTO L0
```

このように使う。ラベルは何文字でもよいが、最初の2文字のみで区別する。

GOSUBスタックは8レベル

## ●FOR～NEXT

使用例:

```
FOR I=1 TO 100
```

```
PRINT I
```

```
NEXT I
```

NEXTのあとの変数名は見えないので省略可。また、入れ子は可能だがスタックはGOSUBと共用。

## ●DELAY

```
DELAY 100
```

のように使う。プログラムを指定時間待たせる。単位はms。

## ●END

プログラムを終了、モニタに戻ってコマンドモードになる。

## ●コメント

REM 文だが REM とは書けない。 # と略す。

```
#
#     このように使います
#
```

## 7:エラーコード

エラーチェックは不完全。該当するエラーがあってもエラーにならない場合が多い。実行させるとおかしな結果になる場合は、そのようなエラーが隠れている可能性が高い。

### ●ERR 1XX IN ???

1 行入力ごとにチェックされるインラインエラー。エラーコードとその行番号を表示。

入力時に表示されるので、ペースト終了後にスクロールして確認

エラーがあっても実行するが、エラーの出た行は入力されていないので注意！！ 実行させると結果がおかしくなる。

101	キーワードが見つからない
102	文字定数が""で囲まれていない
103	16 進定数が 4 桁以上
104	THEN か TO のつづりが間違っている
105	GOTO の構文エラー
106	GOSUB の構文エラー
107	GOTO か GOSUB のつづりが間違っている
108	IF か INPUT のつづりが間違っている
109	不明なキーワード

### ●ERR 2XX

ラベル割り当て時にチェックされるラベル参照エラー

実行は出来るが、おそらく暴走する

重要！！1.0 では同じラベルが二つ以上あってもエラーにならない。最初に見つかったラベルを参照する。

200	対応するラベルが見つからない
-----	----------------

### ●ERR X

実行時エラー

エラー発生でプログラムは停止する。

1	代入文で変数の次に=がない
2	代入文でアドレス参照子の次に=がない
3	FOR 文で変数名がない
4	FOR 文で=がない
5	FOR 文で TO がない
6	INPUT 文で入力数字が大きすぎる
10	ゼロでの除算が発生

以上

