

unofficial programming guide to Bean's Embedded BASIC language for the Propeller Chip

EMBEDDED BASIC by Bean

February 24, 2011

Humanoido's comment: For personal use, this is a collection of material posted on the Parallax Forums combined with the only found text document about Beans Embedded BASIC in an effort to provide additional reference information and programming sources for this remarkable language. Note: Information may apply to various versions of Embedded BASIC.

Compiling

You need to use BST to compile it. The PropTool doesn't support LMM code generation.

<http://forums.parallax.com/showthread.php?118611-Download-PropBASIC-here...-00.01.04>

Go here <http://www.fnarfbargle.com/bst/Latest/> and get the latest version of BST. The .exe version is for windows, Linux for linux and osx for macintosh.

Applications

If you want to try Embedded BASIC I now have it running on the spinneret as a telnet application.

<http://forums.parallax.com/showthread.php?129444-Contest-Entry-Spinneret-Embedded-BASIC>

Overview

This is truly a stand-alone programmable system. It runs on a demo board, PS/2 keyboard and a TV.

Display

It has a driver for NTSC display (48x19 characters). It has a function CHARS that returns the address of the character bitmap table. From that you can POKEB the values

Compatibility

Run as-is on the demo board.

Compared to PropBASIC

PropBasic only allow one expression per line, and ONLY during assignment to a variable.

In this case Embedded Basic is a vast improvement as expressions can be used most anywhere a value is needed.

Keyboard & Text

It uses a PS/2 Keyboard. Use of lowercase characters is permitted. Commands get converted to uppercase.

Speed

The goal of EB is to be at LEAST as fast as spin. A "FOR NEXT" loop of 10,000 takes only 230mSec.

```
10 LET A=CNT
20 FOR B=1 TO 10000
30 NEXT B
40 PRINT CNT-A/80000; " MILLISECONDS."
```

This short program show that a simple assignment is 1872 clocks (23.4uSec @ 80MHz)

```
10 LET A=CNT
20 LET B=CNT
30 PRINT B-A
```

Connect a PS/2 keyboard and NTSC monitor to a demo board and try this demo program to see the speed of "Embedded Basic". The "1 CONT" line makes the program auto-run at startup if you SAVE it. Press the ESC key to stop the program.

```
1 CONT
10 CLS
20 PLOT 0,0,1
30 FOR a=1 TO 2000:LINE RND 255, RND 191, 1:NEXT a
40 FOR a=1 TO 2000:LINE RND 255, RND 191, 0:NEXT a
50 GOTO 30
```

Here's another cool speed test program.

```
10 start=CNT
20 FOR a=1 TO 10000
30 NEXT a
40 PRINT CNT-start/80000;" milliseconds"
50 PRINT "for 10,000 FOR..NEXT loops."
```

LMM

The program uses the LMM model

The System Counter CNT

Use CNT to time things in embedded BASIC. This program takes 28 milliseconds for 1000 for...next loops. Or 28 microseconds per loop.

```
10 start=CNT
20 FOR a=1 TO 1000:NEXT a
30 PRINT (CNT-start)/80000;" milliseconds."
```

Program

All program lines must start with a line number. It is customary to use increments of 10 in case you want to add some lines in-between afterwards.

Comments

You can use ' to put comments after commands.

Variables

Variable names must start with a letter, may contain letters and numbers, may be up to 8 characters long. FOR..NEXT variables must be a single letter. Variables are 32-bit signed integers.

Registers

DIRA	Pin direction 0=INPUT; 1=OUTPUT	- write-only
OUTA	Pin outputs 0=LOW; 1=HIGH	- write-only
INA	Pin inputs 0=LOW; 1=HIGH	- read-only
CNT	System counter	- read-only
CTRA, CTRB	Counter mode	- write-only
FRQA, FRQB	Counter frequency	- write-only
PHSA, PHSB	Counter phase	- write-only
VCFG, VSCL	Sets video generator	- write-only
INKEY	Returns value of keypress	- read-only
VARS	Address of variables	- read-only
CHARS	Address of character bitmaps	- read-only

Pin I/O

INPUT, OUTPUT, HIGH, LOW, PIN

For all commands that operate on hardware pins you can specify a range of pins by using MSB..LSB. For example to make pin 23 high use: HIGH 23

To make pins 24 thru 26 high use: HIGH 24..26

!!! NOTE if the MSB value is less than the LSB value, the bits will be reversed, this is the same as the spin language !!!

Operators (in order of precedence)

Parenthesis ()

UNARY +, UNARY -, !, ABS, RND, PEEKB, PEEKW, PEEKL, PIN, ..

SHL, SHR, ROL, ROR, SAR, REV

&

|, ^

*, /, //

+, -

=, <, >, <=, >=, <>

NOT

AND

OR

&, |, ^ are bitwise (AND,OR,XOR); "AND" and "OR" are logical AND and OR.

4 | 1 = 5 ; 4 OR 1 = -1

Math Evaluation Math is evaluated strictly left-to-right.

Commands

BCOLOR BCOLOR {expression}

Sets the background color (see COLOR, FCOLOR)

CLS CLS
Clears the screen

COLOR COLOR {expression}
Sets both background and foreground colors with one value (see BCOLOR, FCOLOR)

CONT CONT {expression}
Continue program after ESC is pressed

DATA DATA expression, expression, expression
Define data to be read with READ (see READ, RESTORE)
DATA elements can be full expressions, for example, DATA 1, 2,3, 2+2, A+6

DEBUG DEBUG
Shows line #'s as program runs

DISPLAY DISPLAY 42 ' prints a "*"
Prints ascii character. May use multiple parameters.
Value 10 moves to next line and moves back to starting position (for multi line displays)

DUMP DUMP
Shows program bytes, press a key to stop

END END
Stops program and returns to command prompt

FCOLOR FCOLOR {expression}
Sets the foreground color (see COLOR, BCOLOR)

FOR/TO/STEP FOR A = 1 TO 10 {STEP 1} (only single letter variable names)
Creates a program loop

GOSUB GOSUB 1000
Go to subroutine (see RETURN)

GOTO GOTO 1000
Jumps to line

HIGH HIGH 23
HIGH 23..26
Make pin(s) an output and high

IF/THEN IF A = B THEN GOTO 1000
IF A <> B THEN c=1000:d=1000
If the condition is true, execute commands following THEN
now allowed: "IF...THEN commands" instead of just "IF...THEN line#"

```
5 CLS
10 LET A=1
20 LET B=CNT
```

```
30 PRINT A;"",";
40 LET A=A+1
50 IF A <= 1000 THEN 30
60 LET C=CNT
65 PRINT
70 PRINT C-B/80000;" MILLISECONDS."
```

INKEY

INPUT INPUT 23

INPUT 23..26

Make pin(s) an input

LET LET A=A*10

LET A=PIN 27..24

Assigns a value to a variable. (LET is optional)

LIST LIST

LIST 1000

Show program listing (Press a key to stop)

LOAD LOAD

LOAD 1

Retrieves program from EEPROM, if 64K eeprom can use LOAD [1-4]

LOCATE LOCATE 5, 10

Sets print location

LOW LOW 23

LOW 23..26

Make pin(s) an output and low

NEW NEW

Clears program and displays version info

NEXT NEXT A

Adjusts value and loops back to FOR line

NODEBUG NODEBUG

Does NOT show line #'s as it runs (see DEBUG)

OUTPUT OUTPUT 23

OUTPUT 23..26

Makes pin(S) an output

PAUSE PAUSE 1000

Pauses for value milliseconds

PIN PIN 23,1

PIN 27..24,15

Sets pin output state. NOTE: DOES NOT SET PIN TO OUTPUT

Here is an explanation of the PIN command for the demo board. You can also do:

varname = PIN 23..16

```
10 OUTPUT 23..16
20 FOR a=0 TO 255
30 PIN 23..16,a
40 PAUSE 100
50 NEXT a
60 GOTO 20
```

PEEK

POKE does the same thing as WRBYTE, it writes to RAM, not EEPROM.
Among other things it will be useful to change the character bitmaps.

POKEB POKEB a,100
Changes a byte of program memory (RAM, not EEPROM)

POKEW POKEW a,1000
Changes a word of program memory (RAM, not EEPROM)

POKEL POKEL a,100000
Changes a long of program memory (RAM, not EEPROM)

PRINT PRINT
PRINT a
PRINT "The value is ";a
Prints to the screen.

READ READ A,B,C
Reads data from the DATA lines

REM REM This is a comment
dirx = 1 ' set direction to 1
Comment

RESTORE RESTORE
RESTORE 1000
Set line that READ will start reading data from

RETURN RETURN
Return from subroutine

RUN RUN
RUN 1000
Runs program

SAVE SAVE

SAVE 1

SAVES program to EEPROM, if 64K eeprom can use SAVE [1-4]

If you have a 64K EEPROM you can save 5 programs total. SAVE or SAVE 0 will save to the default location (loaded at power-up). SAVE 1 thru SAVE 4 will save to extra 32K EEPROM space.

VARs

NOTES

- Single letter variable names are faster than multi-letter variable names
- FOR...NEXT is faster than GOTO
- GOTO needs to scan from the beginning to find the line # requested
- FOR does NOT have to be the first command on a line.
10 CLS: FOR a=1 TO 10:PRINT a:NEXT a
- x.y returns $(x + y*256 + 11141120)$ $11141120 = \$AA0000$ and is just a unique number which means (this is a .. result) $x.y - 0..0 = x + y*256$

Auto Run

Any program where the first line is "1 CONT" will auto-run when loaded (or at powerup for default location).

Multiple commands per line

Create multiple commands per line by separating with a colon ":" For example LET A=5:LET B=6

References

<http://forums.parallax.com/showthread.php?123678-Embedded-BASIC-interpretter&p=919764>

<http://forums.parallax.com/showthread.php?129658-re-ULTIMATE-List-of-Propeller-Languages-amp-speed>

<http://forums.parallax.com/showthread.php?123678-Embedded-BASIC-interpretter>

<http://forums.parallax.com/showthread.php?129576-Potatotext-being-revised-amp-Bean-Embedded-Basic-WIP>