Spinneret PE-Basic Project Report

Project ID: SPIN111244 by Terry Hitt (Bean)

PE-Basic:

While working on an embedded BASIC interpreter for the Parallax Propeller microcontroller, Parallax announced the Spinneret contest and I thought it would be the perfect device to run BASIC over the internet.

The embedded BASIC interpreter is called PE-Basic (PE meaning Propeller Embedded). PE-Basic is written in PropBasic (PropBasic is a BASIC compiler that runs on a PC). Using compiler directives PE-Basic can be compiled for different platforms. One platform is using a terminal program (PST) for input and output. One platform is using a keyboard and TV. And one platform is using the spinneret and telnet.

PE-Basic is a basic interpreter similar to the BASIC used in most home computers of the 1980's. If you have ever used a Commodore 64, TI 99/4A or even a Timex Sinclair computer then you will be familiar with PE-Basic.

Telnet:

The telnet protocol was chosen because a telnet client is included Microsoft Windows so no PC software would have to be developed. To get the telnet client in windows go to the control panel and choose “Programs and features”. Then click on “Turn windows features on or off”. In the list you will find “Telnet Client”. Turn this feature on. Then go to a command prompt and type in the following:

TELNET pebasic.com

If my spinneret is turned on, you should get an OK when you press ENTER.

Using telnet, PE-Basic allows anyone to program the spinneret module from anywhere in the world. Internet access and a telnet client are the only requirements.

Spinneret PE-Basic Project Report

Specifications:

Here are the specifications for Spinneret PE-Basic:

- Telnet interface allows programming over the internet.
- Uses ANSI sequences for cursor positioning and text color
- 4K memory for program.
- ASCII escape sequences for color and cursor control.
- Program storage to EEPROM on Spinneret.
- Up to 8 character variable names (single letter for FOR…NEXT)
- Uses the “ESC” key as break.

Spinneret PE-Basic gives you full control of the Propeller’s registers and hardware counters. DIRA, INA, OUTA, CTRA, FRQA, PHSA, CTRB, FRQB, PHSB

Limitations:

No way to easily move programs to/from a PC.
4K of program space
Strings are not implemented
Array are not implemented
Serial, SPI, I2C and 1-wire not implemented

Future Improvements:

SD Card file storage
String Handling
Arrays
Serial, SPI, I2C and 1-wire
Implement a dual propeller design.

- One propeller handles Video, keyboard and SD card
- Second propeller handles PE-Basic and program storage
- This will provide much more memory for video and programs

Spinneret PE-Basic Project Report

Usage:

If you have ever used a home computer in the 1980's (Apple II, C64, TI 99/4A, etc) then you will be familiar with how to program in PE-Basic. Here is how to program PE-Basic:

Programs are entered as Line# Command.

Line #'s are typically in steps of 10. 10, 20, 30 to allow the insertion of additional commands later.

Commands entered without a line# are executed immediately.

If a line# is entered that already exist, the existing line is deleted and the new line added in it's place.

Variable names may be up to 8 characters long, except FOR…NEXT variables must be a single letter (A..Z).

Variable names must start with a letter, and may contain only letters and digits.

All variables are 32-bit integers.

I/O pins are controlled using the INPUT, OUTPUT, HIGH LOW, PIN commands and the PIN() function.

I/O pin groups are accessed much line "spin". As MSB..LSB for example HIGH 23..16

There are different operators for bitwise and logical operation. Logical false is zero (any value NOT zero is considered true) and logical true is -1.
Hence 1 AND 3 = -1, but 1 & 3 = 3. Also 1 AND 2 = - 1, but 1 & 2 = 0

PE-Basic Manual:

Please see the included PE-Basic manual file for more details.

Spinneret PE-Basic Project Report

Example Programs:

```
1 REM ---------------
2 REM Guess my number
3 REM ---------------
10 CLS
20 a=RND(100)+1
30 PRINT "Guess my number (1 to 100):";
40 b=0
50 c=INKEY:IF c=0 THEN 50
60 IF c=13 THEN 120
70 IF c=8 THEN DISPLAY 32,8:b=b/10:GOTO 50
80 c=c-48:IF c<0 OR c>9 THEN 50
100 b=b*10+c
110 GOTO 50
120 PRINT b;
130 IF b>a THEN PRINT " is too high..."
140 IF b<a THEN PRINT " is too low..."
150 IF b<>a THEN 30
160 PRINT " is the correct answer !!!"
```

```
1 REM ----------------------
2 REM Flashing LED on PIN 24
3 REM ----------------------
10 REM Using HIGH, LOW
20 FOR i=1 TO 20
30 HIGH 24
40 PAUSE 100
50 LOW 24
60 PAUSE 100
70 NEXT i
99 REM
100 REM Using PIN
110 OUTPUT 24 ' Make sure pin is an OUTPUT, PIN does NOT make the pin
an output
120 FOR i=1 TO 40
130 PIN 24, 1-PIN(24) ' Toggle pin state
140 PAUSE 50
150 NEXT i
200 REM Using hardware counter
210 LOW 24 ' Make sure cog has pin low
220 CTRA=4 SHL 26 + 24 ' Mode=4; APIN=24
230 FRQA=100 ' Rate = 2^32 / 80MHz / 100 = 0.53687 Hz
240 PAUSE 3000
250 CTRA=0 ' Stop counter
```

Spinneret PE-Basic Project Report

```
1 REM -------------
2 REM Bouncing Ball
3 REM -------------
10 CLS
20 oldx=0:oldy=0:ballx=0:bally=0:dirx=1:diry=1
100 LOCATE oldx,oldy:DISPLAY 32:LOCATE ballx,bally:DISPLAY 42
110 oldx=ballx:oldy=bally:ballx=ballx+dirx:bally=bally+diry
120 IF ballx = 0 OR ballx=79 THEN dirx=-dirx
130 IF bally = 0 OR bally = 24 THEN diry=-diry
140 PAUSE 100
150 GOTO 100
```