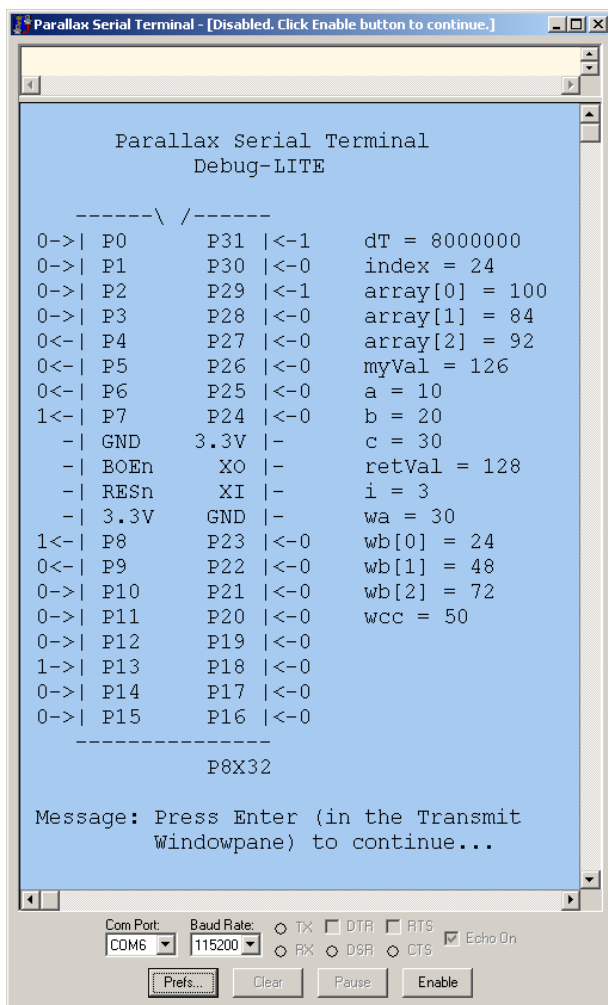




## PE Kit Tools – Debug LITE for the Parallax Serial Terminal

The PST Debug LITE object is a simple and convenient tool for debugging Spin code with the Parallax Serial Terminal. Figure 1 shows an example debug display, paused at a breakpoint during a debugging session. The display features I/O pin directions and states next to a list of variables and their values. There's also a message at the bottom prompting to press Enter to continue from the breakpoint. (Make sure to click the Parallax Serial Terminal's transmit windowpane before pressing Enter.)



**Figure 1:** Parallax Serial Terminal debugging session

“Test Debug.spin” is a program that changes a number of variable values along with a few I/O pin manipulations for the sake of viewing with PST Debug LITE. The calls that resulted in the Figure 1 display are highlighted in Figure 2 along with the typical setup (clock settings, PST Debug LITE object declaration, and call its Start method). The ListHome method sends the Parallax Serial Terminal's cursor to the beginning of the list, then three Vars calls populate the list with three groups of contiguous variables. After the Vars method calls, the Break method call stops program execution

until the PST Debug LITE receives a character (typically Enter) from the Parallax Serial Terminal's transmit windowpane.

**Figure 2:** Parallax Serial Terminal debugging session

```
''' TestDebug.spin
CON

    _clkmode = xtal1 + pll16x          ' System clock Settings:
    _xinfreq = 5_000_000              ' PE Kit's 5 MHz crystal → 80 MHz

OBJ

    debug : "PST Debug LITE"          ' Nickname PST Debug Lite → debug

VAR

    word wa, wb[3], wcc              ' Declare some global variables

PUB go | dT, index, array[3], myVal  ' Local variables

    debug.start(115200)              ' Start PST Debug Lite

    dira[9..4]~~                    ' I/O pins 9..4 set to output
    dt := clkfreq/10                 ' dt stores time increment

    repeat                          ' Main loop
        array[index // 3] += index++ ' Store values in array variables
        outa[9..4] := index          ' Index bits 5..0 to outa bits 9..4
        waitcnt(dT + cnt)            ' Wait 1/10 of a second

    ' Go to start of variable list and display variables + values
    debug.ListHome
    debug.Vars(edT, String("! dT, index, array[3], myVal"))

    myVal := moreVars(10, 20, 30)    ' Call the MoreVars method

PUB MoreVars(a, b, c) : retVal | i   ' More variable operations

    wa := a + b                      ' Store values in wa & wcc
    wcc := b + c
    repeat i from 0 to 2              ' Store values in another array
        wb[i] += (i+1)
    retVal := wa + wcc + wb[1]        ' Store something in retVal

    ' Append variable list with local & global variables, then stop at breakpoint.
    debug.Vars(ea, String("PUB moreVars(a, b, c) : retVal | i"))
    debug.Vars(ea, String("word wa, wb[3], wcc"))
    debug.Break
```

### **Incorporating PST Debug Lite into an Application**

Follow these steps to incorporate PST Debug LITE into an application:

- 1) Declare the Object

```
OBJ
  debug : "PST Debug LITE"
```

- 2) Pass the Parallax Serial Terminal's baud rate to its start method

```
debug.start(115200)
```

- 3) Make sure the clock settings support serial communication at the baud rate you chose. If in doubt with the PE Kit, set the system clock to 80 MHz with the 5 MHz crystal.

```
CON
  _clkmode = xtal1 + pll16x
  _xinfreq = 5_000_000
```

- 4) Use the ListHome, Vars methods to control the display and the Break method to halt cog code execution at certain points.

### **ListHome, Vars, and Break Methods**

Calls to Vars append the variable list on the left hand side of Figure 1. A call to the ListHome method places the cursor back at the beginning of the list:

```
debug.ListHome
```

The list of variables in Figure 1 is three separate lists that were transmitted to the Parallax Serial Terminal as a result of the three different Vars method calls. The list includes both global and local variables, arrays and a method return value for good measure.

The Vars method can be used to display global long, word or byte variables as well as local variables. When displaying local variables, the Vars call should be inside the method with the local variables to be displayed.

The Vars method call accepts two parameters: startAddr and stringAddr. The startAddr parameter should be the first variable in a declaration with the variables you want to display, and stringAddr is the address of a string that contains a copy of the variable declaration. Here are three examples from Test Debug.spin:

```
debug.Vars(@wa, String("word wa, wb[3], wcc"))
debug.Vars(@dT, String("| dT, index, array[3], myVal"))
debug.Vars(@a, String("PUB moreVars(a, b, c) : retVal | i"))
```

...and of course, you can set a breakpoint with a call to the Break method:

```
debug.Break
```

To resume program execution from a break method, click the Parallax Serial Terminal's transmit windowpane, and then press the enter key on your keyboard once.

### **Testing PST Debug LITE**

- ✓ Download “PE Kit Tools – PST Debug LITE.zip”.
- ✓ Unzip to a folder.
- ✓ Open Test Debug.spin with the Propeller Tool.
- ✓ Open PST.exe, and set the Baud Rate to 115200 to match the baud rate in the example program’s Start method. Then, set the Com Port to the Propeller chip’s programming port. If you’re not sure what the COM port number is, use F7 in the Propeller Tool software to find out.
- ✓ Make sure the Parallax Serial Terminal is resized to be large enough to fit the display shown in Figure 1.
- ✓ In the Propeller Tool, press F11 to load the program into the Propeller chip’s EEPROM.
- ✓ As soon as the Propeller Communicating window reports “Loading EEPROM...” click the Parallax Serial Terminal’s Enable button. (Don’t wait for the download to complete before clicking Enable.)
- ✓ Click the Parallax Serial Terminal’s transmit windowpane and press/release the Enter key on your computer’s keyboard to advance from one breakpoint to the next.
- ✓ Make sure to take a close look at the Start, ListHome, Vars, and Break method calls.

### **Vars Method Call Tips**

Here is a quick, simple and reliable way to add a Vars method call a program:

- ✓ Start with `debug.Vars(@ , String(“”))`
- ✓ Copy and paste the method block or variable declaration that contains the variables you want to display into the string.
- ✓ Copy/paste the first variable from the declaration as an address (starting with @) in the first argument.

Example:



Make sure that:

- ✓ The String begins with byte, word, long, PUB, PRI or |. (Not case sensitive.)
- ✓ The start address in the first parameter matches the first variable in the variable declaration string.

If the Parallax Serial Terminal’s Echo On checkbox is checked, make sure to only Press/release the enter key once after each breakpoint is reached.

### **Next Steps for the PST Debug LITE Project**

This first step in the PST Debug LITE project involves testing with some simple programs in a single cog to improve the interface and get the bugs out. (Along the way, code space needs to be reduced so that it really qualifies as LITE, string parsing needs to be condensed, user interface methods improved and so on...)

Step two will incorporate semaphores to allow multiple cogs to pass information without scrambling the display. The version of FullDuplexSerialPlus this object relies on for communication with the Parallax Serial Terminal incorporates features in SerialMirror from the [obex.parallax.com](http://obex.parallax.com). This will be crucial for step 3, expanding PST Debug LITE to allow all objects in a given application to pass information to the same serial buffer.

Please feel free to report bugs, ask questions or pitch in with suggestions in this thread. You can also contact me directly using the email button to the left of this post.