

sphinxcompiler

mpark@sphinxcompiler.com

Installation

The Sphinx distribution consists of three directories: sphinx0, sphinx1, and sphinx2.

- sphinx0 contains files that you compile on your PC.
- sphinx1 contains the core Sphinx system.
- sphinx2 contains the source code for Sphinx.

The following installation instructions also serve as an introductory tutorial to Sphinx. Please read and follow them carefully.

Part 1

1. Open sphinx0\sphinx.spin in the Propeller Tool (or equivalent).
2. Edit the clock settings and pin assignments at the top of the file to match your Propeller system.
3. Compile and save to your Propeller's EEPROM (press F11 in PropTool).

Part 2

1. Insert an SD card into your PC. If necessary, format it as FAT16 with clusters at least 32k bytes in size. If you are running Windows, you can format the card by opening a command prompt and entering the following command (change f: as appropriate):

```
format f: /fs:fat /a:32k
```
2. Copy the *contents* of the sphinx1 directory to the SD card. There should be two dozen or so files in the root directory of the SD card.
3. If you are interested in compiling Sphinx with Sphinx, copy the contents of the sphinx2 directory to the SD card. This step is optional.
4. Move the SD card to your Propeller.

Part 3

1. Boot up your Propeller. On the screen you should see something like this:

```
===== Sphinx xxxxxx =====  
video @ pin 24  
i 1 1 0 1 0 1 0
```

```
keyboard @ pin 13
SD @ pin 16
>_
```

The underscore is the cursor.

2. Now it's time to build the editor. Most of the editor's sub-objects are already compiled, but you have to compile a couple of files to get some practice. Enter the following commands:

```
c tvtexted (this will compile the editor's video driver)
cl ed (this compiles and links the editor, producing the file ed.bin)
```

3. Now use the program you just built to edit a simple Spin program:

```
ed hello1.spn
```

Note that source files end with ".spn" (Sphinx only supports short (8.3) filenames).

4. Ed is a simple full-screen editor. Use the arrow keys to move the cursor. Edit the clock settings and video pin as appropriate for your Propeller and change "<your name>" to your name.
5. Hit control-Q to quit. Ed will give you the choice to save the file as hello1.spn. Hit Enter to save your changes.
6. Enter the following commands:

```
cl hello1
run hello1
```

The run command effectively performs a reset before running hello1. You should see a blue screen with your name on it. After a couple of seconds your Propeller will reboot back to Sphinx.

Part 4

1. Edit hello2.spn, changing "<your name>" to your name, and save. hello2 is "Sphinx-aware" and uses Sphinx I/O.
2. Compile and link hello2 using the following commands:

```
c hello2
link hello2
```

Here we compiled and linked in two steps. If you ever mistype c when you meant cl, don't panic—you can always run the linker separately as you did here.

3. Run hello2 by simply entering its name (do not use the run command):
hello2

Note that the output from hello2 appears on the Sphinx screen and the Sphinx prompt appears immediately after hello2 terminates.

Part 5

1. Compile and link the rest of the [utility](#) programs.

```
cl cogcheck
```

```
cl copy
cl del
cl dir
cl echo
```

2. Try running the utilities by entering their names at the command prompt. These are Sphinx-aware programs like hello2, so do not use the run command.

Part 6 (optional)

At this point you already have a functional Sphinx system. This next section is completely optional. It shows you how to build Sphinx with Sphinx.

1. Copy the contents of the sphinx2 directory to the SD card if you did not do so in Part 2.

2. Enter the following commands to build codegen.bin:

```
c stringx
c bintree
c kwdefs
c isxfs
c tokenrdr
c symbols
c eval
c methods (this will take about 40 seconds)
c sxfile
c isxtv
cl codegen (this is another big one)
```

3. Enter the following commands to build lex.bin and link.bin:

```
c fltmath
c fltstr
c keywords
c tokenizr
cl lex
cl link
```

4. Enter the following commands to build sphinx.bin:

```
c sxkb
c sxtv
c sxsdspiq
c sxfs2
c sxfs
cl sphinx
```