Alternate Text Editor for BasicX BX24 Source Code

Date: March 16, 2005

Author: Harry Stoner (tedstoner@1930s.com)

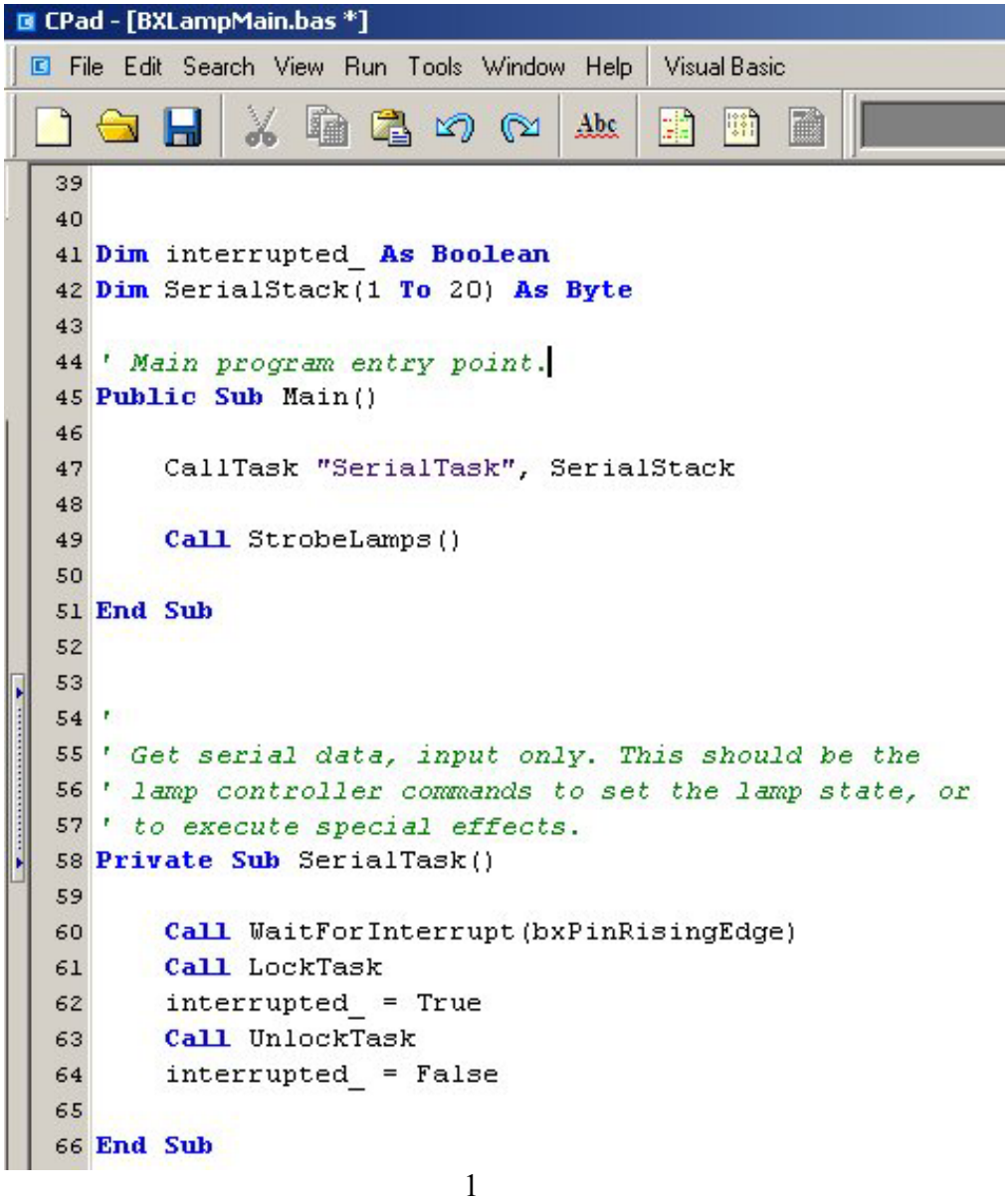
This document describes the setup and use of an alternated text editor for creating and editing the Visual Basic source code to be downloaded to a BX24 microprocessor.

The Problem

The text editor supplied with the BasicX IDE (version 2.10 at time of writing) is very crude and does not even support an "undo" operation. To me that makes it unusable. I would expect to have an editor that at a minimum supports one or more levels of undo (hopefully multiple or unlimited). It would also be nice if the editor understood Visual Basic (VB) syntax or structure, and could automatically indent text lines, etc.

In other words, it would be nice to have an editor with capabilities found in many other standalone editor or imbedded IDE editor products available on the market today.

Figure 1 - sample editing screen for a VB program.




```
39
40
41 Dim interrupted_ As Boolean
42 Dim SerialStack(1 To 20) As Byte
43
44 ' Main program entry point.
45 Public Sub Main()
46
47     CallTask "SerialTask", SerialStack
48
49     Call StrobeLamps()
50
51 End Sub
52
53
54 '
55 ' Get serial data, input only. This should be the
56 ' lamp controller commands to set the lamp state, or
57 ' to execute special effects.
58 Private Sub SerialTask()
59
60     Call WaitForInterrupt(bxPinRisingEdge)
61     Call LockTask
62     interrupted_ = True
63     Call UnlockTask
64     interrupted_ = False
65
66 End Sub
```

One Solution – CPad

I found a freeware editor called CPad, and so far it seems acceptable. It can be downloaded from http://zantii.net/ . The version of the tool is 0.26A.

This editor understands VB syntax, at least to the extent that it recognizes comments, reserved words (e.g. "Sub", "Function", "as Byte"), text strings, etc. It supports multiple undo operations and other features standard with many editors. See Figure 1.

Installation and Setup

Go to the "Downloads" page at the site and download the CPad.zip file (approximately 3.9mb). Unzip to a temporary location and run the included .exe setup file.

Once installed, start it up and set up your basic preferences from the "Edit/Preferences" menu items. Every one has their favorite settings and different ways of working, so the preferences I suggest here are just that – suggestions.

On the default display (for the "Editor- Text settings" - see Figure 2) I checked the following options:

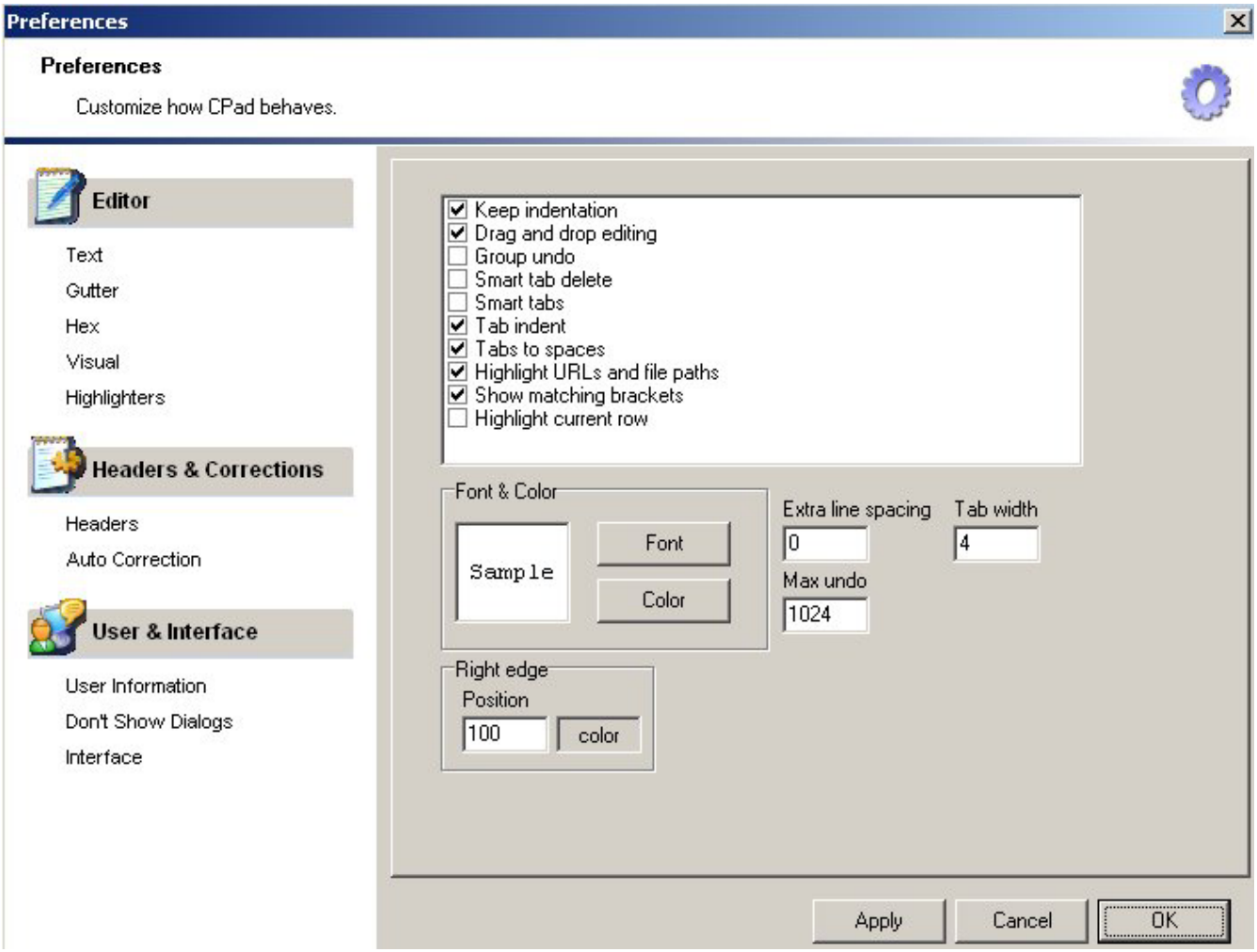
- "Tab indent"
- "Tabs to spaces"

I unchecked:

- Highlight current row (I found this feature annoying).

I also set tab width to 4 from the default 2.

Figure 2 - preferences screen (Edit/Preferences).



Next go to the "Editor – Highlighters" settings. Select from the "Language" drop down "Visual Basic". Then click the "Highlighters" tab. Here you can customize the highlight colors the editor uses with the code.

For example, for the "Comments" attribute, I set the foreground color to dark green and the background color to white. For the "Reserved Words" attribute I set the foreground color to blue.

Configuring the BasicX Compiler

It's nice if you can both edit and compile the code from the editor application. CPad of course does not directly know about BasicX or it's flavor of a VB compiler. So you need to run the BasicX compiler.

With some fiddling I managed to figure out how to invoke the BasicX compiler from a command line. I was then able to use this command format to configure an external too for CPad. It's pretty easy to set up.

First, the BasicX command format to do a command line compile. Note that the BasicX IDE gui application is still used, but it will close when it is done.

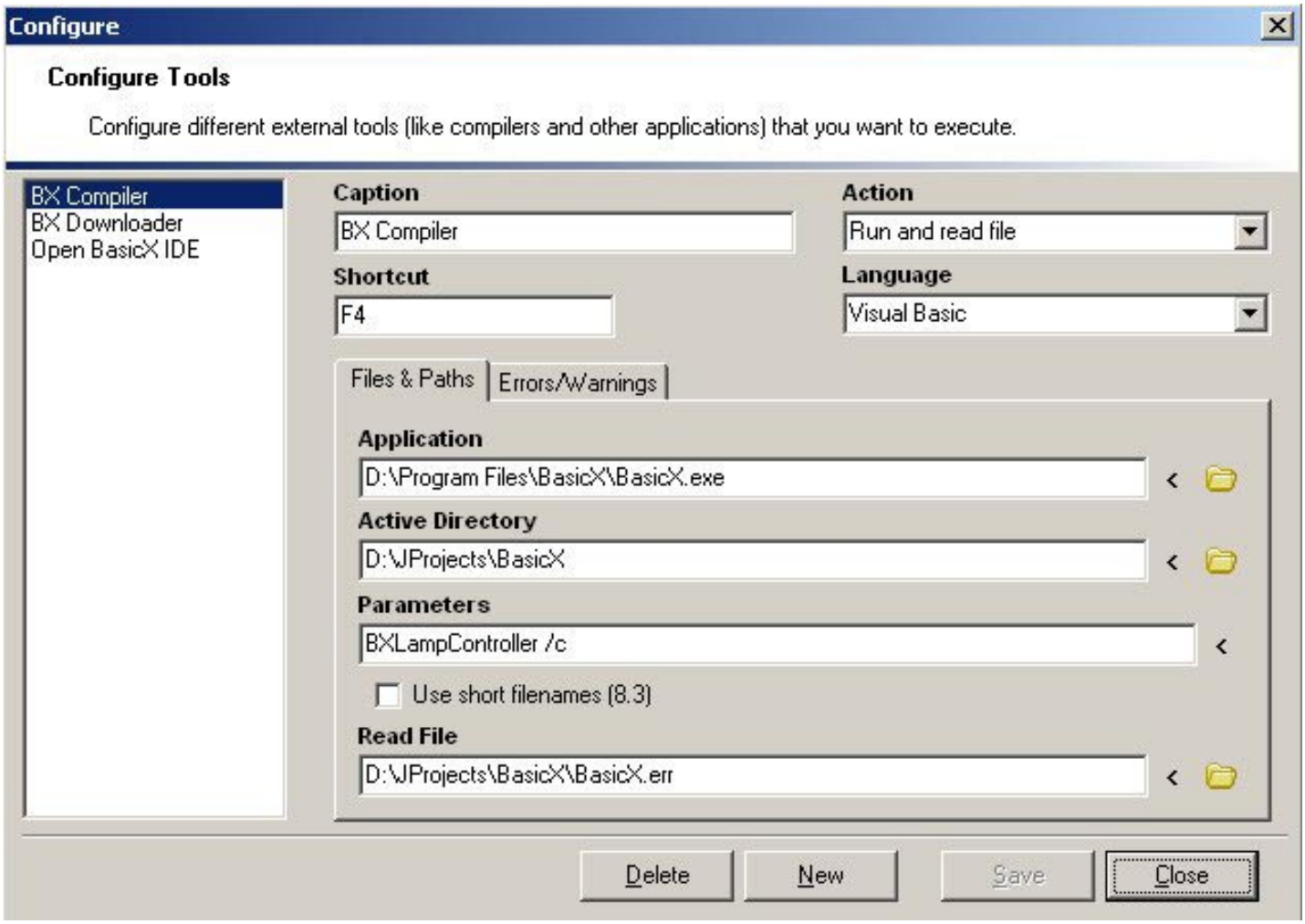
Compile Command

The command format is:

```
basicx (Basicx project name) /c
```

To try it out, execute this from a DOS prompt with the current directory set to the BasicX install directory on your machine. The BasicX project and associated files need to be set up ahead of time from within the BasicX IDE.

Figure 3 - configuring the editor to run theBasicX IDE for compiling.



When you run this command, it will compile the source code in the project then close the BasicX IDE application. Compilation messages are written to file BasicX.err in the project directory.

Now to configure CPad to use this See Figure 3.

Click "Run/Configure External Tools …". From the dialog, click "New" to define a new tool. Fill in the fields as required. Below is some guidance:

- Caption – I called mine "BX Compiler". This is the name as it will appear in the "Run" menu. Call it whatever you want.
- Shortcut – this is the keyboard shortcut. I used "F4". So if I press F4 the external tool (BasicX compiler) will do it's job.
- Application – point to the BasicX.exe executable.
- Active Directory – point to your project directory.
- Parameters – enter the command line parameters as described above (BasicX project name followed by "/c").
- Action – set to "Run and read file". Setting this will cause the "Read file" text box to appear at the bottom.
- Language – "Visual Basic"
- Read file – point to your project directory with file name "BasicX.err".

Click "Save" and then "Close". The tool should now appear in the Run menu.

Edit your VB file, save it, then either press your keyboard shortcut or click "Run/toolname", where "toolname" is the "Caption" set above. See Figure 4.

Download Command

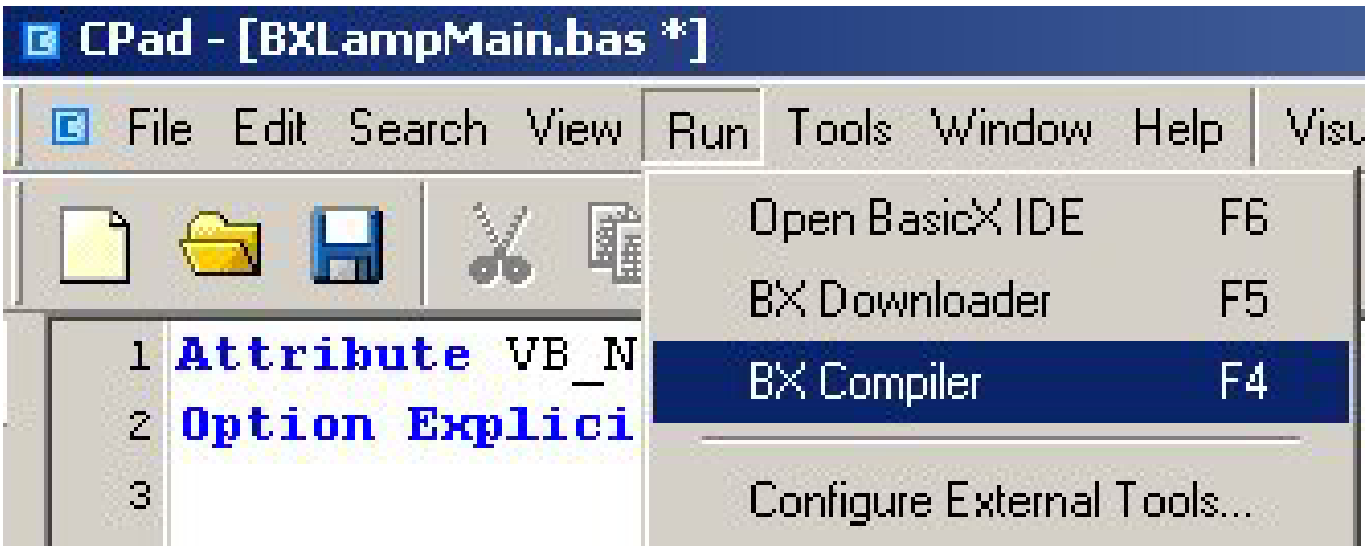
To get the BasicX program to download code, you use the same command format as above, except the command line switch is "/d" instead of "/c".

So repeat the above instructions to define a tool that will be called "BX Downloader" (or whatever), with the same settings except change the Parameters to use "/d" instead of "/c". I also used F5 for the keyboard shortcut.

Open the IDE

You may also want to configure a similar command in the "Run" menu to simply open the BasicX IDE (without compiling or downloading), for example if you were going to do debugging from there.

Figure 4 - doing a compile via the "Run" menu.



Summary

With the above configuration, I can open and edit multiple VB files with a reasonably feature-rich editor. To compile the code I press F4. Results of the compilation will be captured and displayed in the output window at the bottom. Note that you may have to click the "expand/collapse" widget at the bottom to see the output window.

If compilation is successful, then I would press F5 to download the code to the physical device.

I would now want to open the BasicX IDE if I was doing debugging or wanted to make minor code changes. It would be best to close CPad while doing that. However if you have no need for debugging in the IDE console, then you can probably work exclusively in CPad.

Caveats

I have only used CPad for a coupler of days now and I am inexperienced with the use of BasicX, so there may be things I am missing, the tool may turn out to be buggy, etc. But the tool and techniques shown here illustrate how you can reasonably expect to develop and download BasicX code without relying upon the inferior BasicX IDE text editor.

Other text editor tools both free and non-free may work better than CPad. It just happens to the tool I found and initially liked.

Note also that CPad will show a "Visual Basic" menu item on the menu bar at top if you open a VB file. Don't get excited though – you would need to code a CPad plug-in to do anything fancy, and you would have to do that using a couple of specific tools.

I did not locate a VB plug-in for the open-source Eclipse IDE. However, if someone were to create one there, Eclipse would probably be a superior platform to build on than CPad.

A BasicX Yahoo group contriibutor (Tom Handley) recommends http://www.winsite.com/bin/Info?500000017700 (Programmer's File Editor) as another free editor. I have not tried this myself.