BASIC Stamp Windows Interface (v1.1)

FEATURES:

GENERAL:

- **Win95/98/Me/NT 4.0/2000** compatible. This software runs under Windows 95 and 98 (both new and upgrade versions), as well as Windows NT 4.0 and Windows 2000.
- **Supports BS2, BS2e, BS2sx and BS2p modules.** Special features allow for programming entire BS2e, BS2sx or BS2p projects (up to eight files) at once.

EDITOR:

- **Multiple Document Interface** in tabbed-page format allows up to 16 BASIC Stamp source code files to be open at once. Code may be cut and pasted between files easily and efficiently with this interface. Combined with the $STAMP directive (explained later), the editor can manipulate single BS2, BS2e, BS2sx or BS2p source code files or entire BS2e, BS2sx or BS2p projects (up to eight files per project).
- **File History** list appears under file menu allowing quick access to the last 0 to 10 (configurable) files accessed.
- **Color and Font Size** used in the editor windows are configurable.
- **Default COM port** may be set to AUTO, allowing automatic detection of BASIC Stamp 2, 2e, 2sx or 2p, or may be set directly to a specific port. A com port edit button is provided to modify the list of ports available for scanning (for example, it is desirable to remove a modem port from the list).
- **Default directory** where BS2, BS2e, BS2sx or BS2p source code files are stored can be configured.
- **Syntax Check** feature allows proper code format verification without actually downloading the code or engaging the Memory Map window.
- **Memory Map** window provides view of RAM and EEPROM allocation for BS2 files and BS2e, BS2sx or BS2p projects (up to eight source code files).

DEBUG WINDOW:

- **Terminal-like** features allow receiving and sending data. This single interface can provide easy use of the most commonly implemented features of the BASIC Stamp, DEBUG and SERIN/SEROUT. Transmitted and received data appear in two separate panes that are user sizable.
- **Multi-Threaded** features allow up to four Debug windows to be open at one time (on different ports) for debugging complex BASIC Stamp networks.
- **Open any time** feature allows Debug Terminal to be opened at any time, not just after a program download, and may remain open while user switches back to Editor window.
- **Color and Font Size** used in the Transmit and Receive panes are configurable.
- **COM port settings** such as Port, Baud Rate, Parity, Data Bits and Flow Control may be changed in real-time (even while data is being received), allowing easier debugging and configuration at times when the proper settings may not be obvious.
- **Port status indicators** show state of many serial port pins (TX, RX, DSR and CTS) and allow setting the state of DTR and RTS with the click of the mouse button. This is a great way to simulate the effects of the DTR line going high when using a standard terminal program.
- **Buffer Size** for the Receive pane is configurable to allow scrolling back of up to 8192 lines. Pause button may be used along with scrolling feature if data is arriving continually; data will still be stored in buffer, but screen will not update until Resume is selected.
- **Macro Keys** may be defined which contain text/data which are needed to be output from the Debug Terminal. Up to 26 macro keys may be defined in one Macro Key file. Macro Key files may be saved or loaded allowing definition of different macro key sets for different projects.
- **Special Control Character** set has been enhanced to allow more control over Debug Terminal formatting to ease development. Standard (downward compatible) control characters include: Clear Screen, Home, Bell, Tab and Carriage Return. Enhanced control characters include: MoveTo (x,y), Cursor Left, Right, Up and Down, Backspace, Line Feed, Clear Right and Clear Down. Each of the control characters may be disabled through preference settings.

PRODUCT INTRODUCTION

The current version of the Stamp Windows Interface (1.1 and above) provides support for BASIC Stamp 2, BASIC Stamp 2e, BASIC Stamp 2sx and BASIC Stamp 2p in one convenient interface.

INTERFACE INTRODUCTION

The BASIC Stamp Windows Interface was designed to be easy to use and mostly intuitive. Those that are familiar with the DOS-only version of the interface as well as standard Windows software should feel comfortable using the BASIC Stamp Windows Interface.

EDITOR WINDOW

The Interface consists of one main editor window that can be used to view and modify up to 16 different source code files at once. Each source code file that is loaded into the editor will have its own tab at the top of the page labeled with the name of the file. Source code that has never been saved to disk will default to "Untitled#"; where # is an automatically generated number. A user can switch between source code files by simply pointing and clicking on a file's tab.

The status of the active source code page is indicated in a status bar below it and the full path to the source code (if it has been loaded from or saved to disk) will appear in the title bar of the BASIC Stamp interface. The status bar contains information such as cursor position, file save status, download status and syntax error/download messages.

BASIC STAMP 2, 2e 2sx & 2p PROGRAMS:

For BASIC Stamp 2 programs, each editor page is a separate project, any of which can be downloaded to a BASIC Stamp 2 module individually. For BASIC Stamp 2e, 2sx and 2p programs, each editor page can be a separate project, or part of a single project. Any "project" (consisting of up to eight files) can be downloaded to the BASIC Stamp 2e, 2sx or 2p module.

A preference setting in the editor can be used to set the default Stamp mode (BS2, BS2e, BS2sx or BS2p) depending on which model of BASIC Stamp used most often. This setting, found on the Editor Operation tab of the Preferences window, can even be used to indicate what model of BASIC Stamp is being programmed at the moment. A better, and more recommended method, is to use the new directive $STAMP.

STAMP 2 DIRECTIVE:

The $STAMP directive is a special command which can be included (usually near the top) in a program to indicate the model of BASIC Stamp targeted. This directive overrides the default Stamp mode preference setting for the currently active program and makes developing on different Stamp models, within the same editing session, more convenient. For BASIC Stamp 2 programs, the directive should look similar to:

```
' {$STAMP BS2}
```

Note that the directive appears on a comment line (the apostrophe (') indicates the remaining text to the right is a comment). This is for compatibility with the DOS Stamp editors. The DOS Stamp editors will ignore the line as if it were just a comment. The entire directive must be enclosed in brackets, {…} and may contain additional spaces in certain areas. For example:
' { $STAMP BS2 } , '{$STAMP BS2} and '{$STAMP BS2 } are all acceptable variations, however:
'{$ STAMP BS2} and '{$STAMPBS2} are not acceptable and will be ignored.

This directive is read and acted upon by the Stamp Windows editor any time a source code file containing it is loaded, tokenized, run (downloaded) or viewed in the Memory Map.

STAMP 2e DIRECTIVE:

For BASIC Stamp 2e programs, the directive is similar to the above syntax but with the additional option of filenames. Here are two of examples:

```
' {$STAMP BS2E}
        - or -
' {$STAMP BS2E, file1, file2, …, file7}
```

Place the first version near the top of your source code to indicate that it is a BS2e program. Use the second version if a BS2e project, consisting of multiple files, is desired. The *file1*, *file2*, etc. items should be the actual name (and optionally the path) of the other files in the project. *file1* refers to BS2e program number 1, *file2* is BS2e program number 2, etc. If no path is given, the path of program 0 (the program in which the $Stamp directive is entered) is used. Up to seven filenames can be included, bringing the total to eight files in the project all together. Upon tokenizing, running or viewing program 0 in the Memory Map, the editor will read the directive, determine if the indicated files exist, will load them if necessary and change their captions to indicate the project they belong to and their associated program number. After the directive is tokenized properly, and all associated files are labeled properly, tokenizing, running or viewing any program in the Memory Map will result in that program's entire project being tokenized, downloaded or viewed.

When a file that is part of a BS2e project is closed, the entire project (all the associated files) will be closed as well. When program #0 of a project is opened from diskette, the entire project will be loaded as well.

STAMP 2sx DIRECTIVE:
For BASIC Stamp 2sx programs, the directive is exactly the same as the BASIC Stamp 2e except that it specifies BS2sx instead. Here are two of examples:

```
' {$STAMP BS2SX}
        - or -
' {$STAMP BS2SX, file1, file2, …, file7}
```

STAMP 2p DIRECTIVE:
For BASIC Stamp 2p programs, the directive is exactly the same as the BS2e and BS2sx except that it specifies BS2p instead. Here are two of examples:

```
' {$STAMP BS2P}
        - or -
' {$STAMP BS2P, file1, file2, …, file7}
```

SOURCE CODE TABS:
The tabs at the top of each open source code page will indicate the name of the file and, in the case of BS2e, BS2sx or BS2p files and projects, the program's logical number and project it belongs to. For example, if a tab displays "Test.bs2", it is a BS2 program called Test. If a tab displays "0:Test.bse", it is a BS2e file, called Test, is logical program number 0 and will be downloaded as such. If a tab displays "[Test] 1:Process.bse", it is a BS2e file, called Process, is logical program number 1 and belongs to a BS2e project called Test. It will be downloaded into program slot #1 in the BS2e immediately after 0:Test.bse is downloaded into program slot #0.

BS2e, BS2sx and BS2p DOWNLOAD MODES:
The editor has the ability to treat BS2e, BS2sx or BS2p projects as one logical unit and can download each of the associated source code files to the BS2e, BS2sx or BS2p at once. In order to minimize download time for large projects a BS2e, BS2sx or BS2p Default Download Mode is available in the Preferences window. The available modes are: "Modified" (the default), "All" or "Current" and are explained below. This item only affects download operations for the BS2e, BS2sx and BS2p.

Download Mode	Function
Modified (default)	This mode will cause only the source code files that were modified since the last download to be downloaded next time. If no files have been modified since the last download, or the entire project has just been loaded into the editor, all the files will be downloaded next time. This mode decreases the delay during downloading BS2e, BS2sx or BS2p projects and should help speed development and testing.
All	This mode will cause all the source code files to be downloaded each time.
Current	This mode will cause only the current source code file to be downloaded, ignoring all the others.

Regardless of the download mode selected, the BS2e, BS2sx and BS2p programs will be downloaded into the program slot indicated in their tab.

DEBUG TERMINAL
The Debug Terminal window (simply called the Debug window in the DOS-only software) is available via the Run menu. It has been greatly improved compared to the DOS-only version and may be opened at any time. It will also automatically appear after a program containing a DEBUG command is downloaded to the BASIC Stamp. A new Debug Terminal window can be manually opened by clicking on the New Debug button, selecting Run->Debug->New, pressing Ctrl+D, or by pressing the F11 key. Up to four Debug Terminal windows can be opened at once, each on a different serial port. The Debug Terminal may be left open while editing and downloading new code. If multiple Debug Terminals are open, their title bar indicates their identity. For example, the first will be called "Debug Terminal #1", the second called "Debug Terminal #2", etc. Pressing Ctrl+1, Ctrl+2, etc. will bring terminal #1 or terminal #2 forward (if it was hidden by another window). Pressing the F12 key will iterate through the open windows, including the Editor window. Additionally, each open Debug Terminal has an associated button on the editor window that can be used to bring that terminal to the front.

The main portion of the Debug Terminal is split into two panes consisting of the Transmitter pane (above) and the Receiver pane (below). A movable splitter bar separates the two panes and can be used to resize them in relation to each other. The Receiver pane is similar to the entire Debug window in the DOS-only software. It displays any data received from the Stamp's DEBUG or SEROUT command. Data that recently scrolled off the pane can be viewed again through the use of the Receiver pane's vertical

scroll bar. The Transmitter pane displays data entered at the keyboard and can be used to send data to the BASIC Stamp through the same serial port.

The top of the Debug Terminal contains many serial port configuration items that can be modified unless the Debug Terminal was opened automatically after a download. For example, (if the Debug Terminal was opened manually) the Baud Rate of the port can be changed even while data is being received. To change to another available serial port, simply select one from the COM Port combo box. An error message will be displayed if a port cannot be accessed for any reason.

To the right of the serial port configuration items is a set of status lights and check boxes that indicate the current status of their associated data or control line on the serial port. A bright green light indicates that line is active (high), while a dark green light indicates the line is inactive (low). This can be used to monitor received and transmitted data on the serial port. The DTR and RTS checkboxes also indicate line status but may be directly controlled by clicking on them. A checkmark indicates the line is active, while an empty box indicates the line is inactive.

At the bottom of the Debug Terminal window, is a set of buttons. The Capture button (not implemented in this version) will store the received data (from the Receiver pane) to a file on disk. The Macro Key button will open the Macro Key definition window (see below). The Pause button will halt the updating of the Receiver pane allowing a user to scroll the display backward in cases where data is being received constantly. The Close button will close the Debug Terminal.

The Debug Terminal Receiver pane supports a number of special control characters. Like the DOS-only version, Clear Screen, Home, Bell, Tab and Carriage Return are supported and perform the indicated function. In addition to these standard control characters, MoveTo(x,y), Cursor Left, Right, Up and Down, Backspace, Line Feed, Clear Right and Clear Down are also supported. The Cursor Left, Right, Up and Down control characters are ASCII 3,4,5 and 6, respectively. Sending an ASCII 3 value from the BS2-IC (with the command: SEROUT 16,84+$4000,[3]) would result in the Receiver pane's cursor moving left one position. These special characters give great flexibility in formatting and refreshing data on the Receiver pane. Most should operate in an intuitive manner. Those that may not are listed below:

- MoveTo(x,y) – ASCII 2 – This is a unique control character that allows moving the screen cursor to a specific character location on the screen. It must be followed by two additional characters whose value indicates the X and Y position desired. For example: SEROUT 16,84+$4000,[2,10,15,"hello"] would first move the cursor into column 10 (the x position) and row 15 (the y position) and then display *hello* on the screen starting from that position. NOTE: If the pane is not big enough for the position indicated in a MoveTo command, the cursor will wrap to the other side of the screen.
- Clear Right – ASCII 11 – This control character will cause all characters starting at the cursor position and to the right being cleared from the screen.
- Clear Down – ASCII 12 – This control character will cause all lines starting at the cursor and below being cleared from the screen. NOTE: Only visible lines will be affected.

Any of the special control characters can be disabled from the Debug Function tab of the Preferences window. The chart below lists all the available special control characters and their function:

Decimal Value	Function	Decimal Value	Function
0	Clear Screen	7	Bell
1	Home	8	Backspace
2	MoveTo (X,Y)	9	Tab
3	Cursor Left	10	Line Feed
4	Cursor Right	11	Clear Right
5	Cursor Up	12	Clear Down
6	Cursor Down	13	Carriage Return

MACRO KEY WINDOW

The Macro Key definition window (available by clicking on the Macro Keys button from the Debug Terminal) allows the creation of special hot-keys for predefined text. This can be used to store common data to be transmitted to the BASIC Stamp 2 during run-time. For example, during the development of a project, it might be necessary to manually send data to the Stamp that would normally be sent by some other device, or by another computer program. If this data were long or complicated, it would be tedious to type it in at the keyboard, over and over again, throughout the development. The Macro Key definition window allows this data to be assigned to one of 26 hot-key combinations. A user can define, for example, Ctrl+Shift+A to be the standard output data of a GPS receiver. Simply pressing Ctrl+Shift+A in the Debug Terminal will thereafter result in the data stream being entered into the Transmiter pane and transmitted through the serial port just as if it was typed in from the keyboard.

The Macro Key definition window requires the defined macro keys be stored in a macro key file. Macro Key files can be loaded and saved at any time. To add a new macro key, click on the Add button and enter the Name, Key and Data into the edit controls on the right. The Current Macros list on the left displays all the currently defined macros in the list, sorted by Key.

MEMORY MAP WINDOW

The Memory Map window displays detailed and condensed views of how the current source code will fill the Stamp's EEPROM memory and how the RAM memory will be allocated. To view this information, simply press Ctrl+M, press the F8 key, or select the Run->Memory Map menu item while viewing the desired source code. The source code must tokenize properly in order to display the Memory Map window.

It is important to note that the display only shows the compile-time state of the memory, rather than run-time or current state. The Memory Map does not read the Stamp's memory, it simply generates a display of how the current program will allocate it.

The title bar of the Memory Map window displays the program's name as well as the total percentage of EEPROM memory that will be occupied by the program.

The Detailed EEPROM Map displays a close-up view of EEPROM allocation. The EEPROM data is displayed in 24 rows of 16 columns and can be scrolled up or down with the provided scroll bar. The data is color coded according to the EEPROM Legend.

The Condensed EEPROM Map displays a zoomed-out view of EEPROM allocation. A yellow box indicates the window of data current being displayed in the Detailed EEPROM Map. The Detailed EEPROM Map can also be scrolled by clicking within the yellow box and dragging up or down.

The RAM Map displays all 16 word registers available in the module's RAM memory. Each box within the registers represents one bit and is numbered from 15 to 0 across the top of the map. The registers are color coded according to the RAM Legend. NOTE: The additional Scratch Pad RAM memory (in the BS2e, BS2sx and BS2p) is not shown on the Memory Map windows since it is allocated and used during run-time only.

The Source Code field displays the name of the source code that is being viewed in the window. For BS2 programs and single BS2e, BS2sx or BS2p programs, this field only displays the name of the selected source code. For BS2e, BS2sx or BS2p projects, this field can be set to any of the programs within the project. If this field is selected (highlighted) the up and down cursor keys can be used to iterate through all the programs in the BS2e, BS2sx or BS2p project.

COMMAND-LINE OPTIONS

The Stamp Windows Interface supports the following command-line options:

Command-line Switch	Function
/NOREGISTRY	Keeps the software from writing to the Windows registry. This can be helpful for those running Windows NT under a non-administrator user profile. It is best used by first logging on as an adiministrator, running the software without this switch, setting the desired preferences, exiting the software and running the software with this switch from then on (under any user profile).
/NONOTES	Keeps the software from reading and writing to the Notes.dat file (in the Windows directory) where the last BASIC Stamp filenames accessed are stored.

The command-line options should be entered to the right of the software's program name, separated by spaces. For example:

```
C:\stamp\stampsw.exe  /noregistry  /nonotes
```

will start the software without registry writing or notes.dat reading and writing.

SHORTCUT KEYS
The following table lists the available keyboard shortcuts within the BASIC Stamp Windows Interface.

File Functions	
Shortcut Key	**Function**
Ctrl+O	Open a source code file into the Editor window.
Ctrl+S	Save current source code file to disk.
Ctrl+P	Print current source code.
Editing Functions	
Shortcut Key	**Function**
Ctrl+Z	Undo last action.
Ctrl+X	Cut selected text to the clipboard.
Ctrl+C	Copy selected text to the clipboard.
Ctrl+V	Paste text from clipboard to selected area.
Ctrl+A	Select all text in current source code.
Ctrl+F	Find or Replace text.
F3	Find text again.
F5	Open Preferences window.
Coding Functions	
Shortcut Key(s)	**Function**
F6 or Ctrl+I	Identify BASIC Stamp firmware.
F7 or Ctrl+T	Perform a syntax check on the code and display any error messages.
F8 or Ctrl+M	Open Memory Map window.
F9 or Ctrl+R	Tokenize code, download to the BASIC Stamp and open Debug window if necessary.
F11 or Ctrl+D	Open a new Debug window.
F12	Switch to next window (Editor, Debug #1, Debug #2, Debug #3 or Debug #4)
Ctrl+1, Ctrl+2, Ctrl+3, Ctrl+4	Switch to Debug Terminal #1, Debug Terminal #2, etc. if that Terminal window is open.
Ctrl+`	Switch to Editor window.
ESC	Close current window.