

Primer to Using StampPlot® Pro Standard User Licensed

StampPlot Pro is a serial computer interface program for data acquisition and control for the BASIC Stamp®. StampPlot is a very versatile program with many great features. With the large number of choices, the program can be a little intimidating to the new user. This guide will discuss many of the most popular features of StampPlot, though only brief examples are provided to help you get started.

Here is a partial list of StampPlot features covered in this document:

1. Basic control of the plot through the button bar.
2. Plot analog or digital data easily.
3. Control virtually all StampPlot settings with code.
4. Accept data as strings or as binary values.
5. Save data and images of the plots.
6. Use image, drawing and sound files for effects.
7. Place controls on the interface for monitoring and control.
8. Perform math operations.
9. Configure from macros (text scripts).
10. Use an interface for interactive control of your BASIC Stamp.
11. Perform acquisition and control over the Internet.

Registering Your Software

StampPlot Pro has two types of registration licenses:

- Developer's License for users who wish to use the development features of StampPlot to create configurations, which include drag-and-drop design using the Object Editor and Macro editor to quickly design Graphical User Interfaces (GUIs).
- Standard Users License for those not interested in development. As discussed in this documents, development work may still be done, but not as simply as with the Developer's License. StampPlot may be used with the Standard license for free by educational institutions and home user for personal use.

StampPlot will run for 10 minutes without a registration license for testing. After you have obtained registration, or qualify for the free standard license, you may register by selecting the appropriate license from the Register Menu and entering your provided User Name and Code.

Section 1: Installation and Basic Control

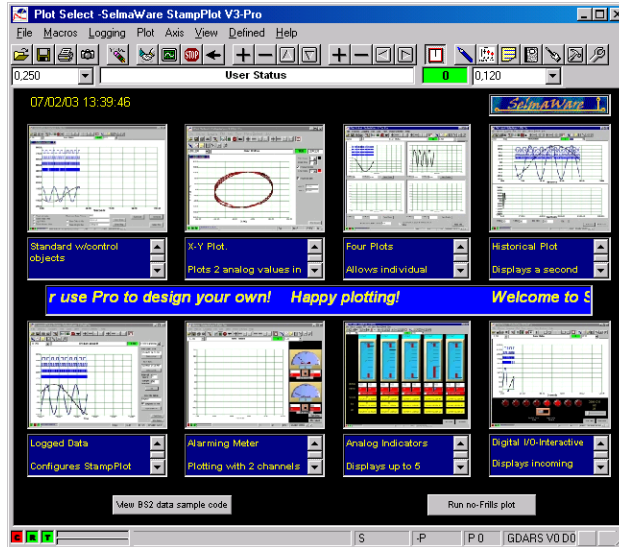
Installation





- Install StampPlot as you would any program by running the setup.exe program.
- Once installed, the program is available from:
Start menu → Program Files → StampPlot V3 Rel2 → StampPlotV3.
- The first time StampPlot is ran it will finish installing the required files and asks you to close and restart StampPlot. The distribution notes will also be displayed.

Configuring and Testing

Before going any further lets make sure you can plot some simple data using StampPlot.

- Open StampPlot.
- The default screen provides a variety of ready to run plots and a BASIC Stamp test program.



- Click on the button "View BS2 Data Code Sample" and answer yes to the security message.
 - A text window will open with Basic Stamp code.
 - Copy and paste the code into your BASIC Stamp Editor.
 - In the editor select your style of BASIC Stamp.
 - Download the program to your BASIC Stamp.
 - View the BASIC Stamp DEBUG Window briefly. You will see data being sent from the BASIC Stamp.
 - ONLY one program can access the communications port receiving this data at any one time. **Before StampPlot can accept and use this data the DEBUG window must be closed.**
- On StampPlot, click the "Run No-Frills Plot" button. This will give you the simple plotting configuration.
- Click the Connect button on the StampPlot button bar. 
- If you get an error message stating there was a problem opening the port:
 - Ensure the BASIC Stamp Debug Window is closed.
 - Open the Configuration Window. 
 - Select the Port Tab.
 - Select the correct COM port then click "Set as Default" to lock it in.
 - Try connecting again.
- If you do not get a connection error message, but do not see an indication of data arriving (R flashing red in lower left corner and no data being plotted):
 - Try another COM port under the Configuration Window, Port Tab.
 - Un-check "DTR Enabled". For certain BASIC Stamp hardware configurations having this checked will 'lock-up' your BASIC Stamp. All boards from Parallax will work with this enabled.
 - Reconnect.
- Click the Plot button to enable plotting on StampPlot. 
- Click the Reset button to reset the plot back to time 0. 

- If all has gone well you should see 4 analog channels and 2 digital values being plotted!
The data is plotted with value for the Y-axis and time of plotting on the X-axis.

Data Points

To be able to redraw and save the plot the incoming digital and analog data is stored in what are termed 'Data Points'. By default the number of Data Points stored is 500. This may be adjusted under the Configuration Window, Data Tab. Changing the number of points will reset your plot. The amount of data points used is shown graphically in the lower left corner.



Once the data points are full the plot will reset. Very large amounts of data points are NOT recommended (over 5000) as they will considerably slow the plot when it is refreshed.

To perform continuous plotting the option to 'Flush Old Data' may be enabled pressing F11. This will remove a percentage of the oldest data from the Data Points storage each time it fills to maximum.

Data Queue

As data arrives it is placed in a queue to wait for processing. If data arrives too quickly the queue will begin to fill and there will be a delay from the time data was sent by the controller to the time it is plotted. The bottom bar indicates the amount of data awaiting processing. The speed with which StampPlot can process data is dependent on the speed of your computer, the nature of the data and other factors, but a rule of thumb is one piece of data every 10 milliseconds (.01 seconds).

Other Plot Control Buttons

Stop and Shift:  

- Using Stop: When the plot reaches the maximum time on the plot the 'Plot' button will go-up stopping the plotting of new data.
- Using Shift: When the plot reaches the maximum time the plot will shift to the left and display the new data being plotted.

Adjusting the Plot Scale: 

- This first set of arrows is used to adjust the Y-axis (analog value) scales:
 - Double scale.
 - Half scale.
 - Shift scale up.
 - Shift scale down.
 - Pressing **CTRL-A** will auto adjust the scales for the plotted minimum and maximum scales.
 - Pressing **CTRL-L** will shift into/out-of logarithmic Y scales.
- The second set of arrows is used to adjust the X-axis (time) in a similar manner.
 - Pressing **CTRL-R** will change the X-axis to show time in Real-Time – time of day and date.

Testing out the Installed Macro Configurations

While the data sent is only 4 analog and 2 digital values, StampPlot can perform many functions without changing any of the code in the BASIC Stamp.

- Press CTRL-F1 or select menu option Macro → Run Startup Macro to return to the initial loading screen.

- Select the 1st plot style choice by clicking the image. You may read about it in the scrolling text box below it prior to clicking the image.
- Connect to the BASIC Stamp and observe how the data is used.
- Some of the control choices may be a little confusing at first, but feel free to experiment! This document will help clarify much of what you see.

Section 2: Plotting Analog, Digital and Messages

StampPlot analyzes the format of the arriving data and uses it. General rules for data are:

- Analog data begins with a value, and can have up to 10 values separated by commas.
- Digital data begins with % and contains only 1's or 0's.
- Messages are those not meeting the above (more rules on this later).
- ALL lines must end in a carriage return (CR).

Plotting Analog Data- ASCII Format

Analog data can be sent from the controller to be plotted by sending the values as text or ASCII strings. For example, the following lines of code will send data as text. The DEBUG instruction is used to send data from the controller back to the computer.

Analog data is formatted as a text value followed by a carriage-return (CR).

- Enter the program provided in your BASIC Stamp Editor.
- Monitor the BASIC Stamp Editor's DEBUG Window to observe the format of data being sent before closing it.
- Use the No-Frills choice of StampPlot for testing: Connect, plot, enable shifting and data flushing (F11).

```
\ Plotting 1 analog value
x VAR Byte
Start:
  FOR x = 0 to 255
    DEBUG DEC x, CR
    PAUSE 100
  NEXT
GOTO Start
```

As can be seen in your DEBUG Window the data arrives as a 0, then 1 and so on. Each value is represented by 1, 2 or 3 characters (bytes) forming each number. This is due to the DEC modifier formatting the values as text.

When StampPlot is connected and plotting you will see the values plotted. The PAUSE is used to prevent the data queue from filling from high-speed data.

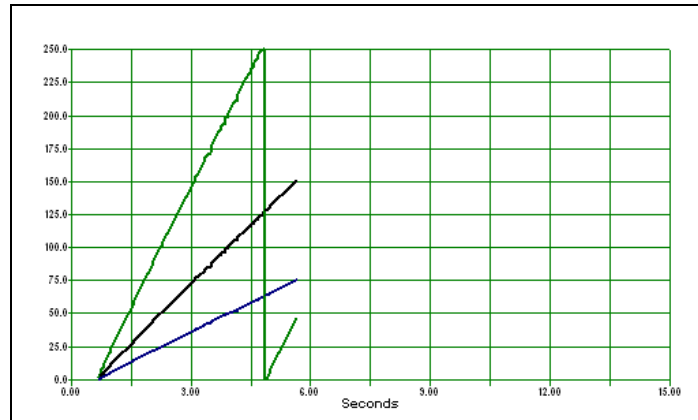
Multiple analog values may be plotted by separating each value with a comma. Notice the form is a DEC value, a comma, a comma in quotes, a comma and another value. The BASIC Stamp requires a comma between each value or string sent, and StampPlot requires commas as part of the text.

```
\ Plotting 3 analog values
Val  VAR Byte
Val2 VAR Byte
Val3 VAR Byte
Start:
  FOR Val = 0 to 255
    VAL2 = VAL / 2
    VAL3 = VAL * 2
    DEBUG DEC VAL, "\", DEC VAL2, "\", DEC VAL3, CR
```

```

PAUSE 100
NEXT
GOTO Start

```



The best way to ensure your data is properly formatted is by viewing it in the DEBUG Window. An example of the data seen in the DEBUG Window is:

100,50,200

Up to 10 values may be plotted simultaneously by separating them with commas.

Plotting Digital Values

By beginning a line with % StampPlot will plot the binary digits – bits (1's and 0's) as digital traces. The IBIN DEBUG modifier is typically used since this will send the values as binary values starting with %. For example **DEBUG IBIN8 129, CR** will display **%10000001**

The IBIN8 will force the BASIC Stamp to send the data with 8 positions. It is very important that the same number of bits is sent consistently and ends with a carriage return.

```

\ Plotting 8 binary values
Val VAR Byte
Start:
  FOR Val = 0 to 255
    DEBUG IBIN8 Val, CR
    PAUSE 100
  NEXT
GOTO Start

```

Sending Messages

When a string is sent that does not start with a value or a percent sign (and a few other restrictions later on), StampPlot will treat it as a message and list it in the message window. As always, the line must end with a carriage return.

```


\ Sending messages
Val VAR Byte
Start:
  FOR Val = 0 to 255
    DEBUG "The value is", DEC Val, CR
    PAUSE 100
  NEXT
GOTO Start

```

Of course analog, binary and messages may be mixed in the same program, but each requires their own line ending with a CR.

Section 3: Debug/Immediate Window and Control Instructions

The Debug/Immediate Window is used to monitor data as it is processed or to directly enter data and instructions.

- Run the previous "Plotting 3 Analog Values" program.
- Connect and Plot on StampPlot.
- Open the Debug/Immediate Window. 
- Check the Anlg (analog) check box to view analog data as it is processed.
- Note that the analog values are displayed.
- Disconnect on StampPlot, enable plotting if not enabled and reset.
- In the text box at the bottom type in 3 values and press return (50,100,150).
- Use the keyboard UP arrow, your last entry is displayed, press return again.
- Enter 3 other values.
- Note in the plot window these values were plotted.
 - NOTE: The plot will not shift automatically when the connection is not open.
- Try some binary values such as %1001 and a message such as Hello World!

The Command Line Interface (CLI) text box at the bottom is used to enter test values or instructions.

Control Instructions

Virtually every facet of StampPlot can be controlled using control instructions. These instructions may come from:

- The user manually entering them in the Debug/Immediate CLI.
- Serially from the BASIC Stamp as strings.
- From a PC based text file (macro).
- Over the Internet using the TCP-Serial Gateway program.

All control instructions are 4-lettered mnemonics starting with an exclamation point (!), and of course, must end with a carriage return.

Try the following instructions using the CLI and watch there effect. Explanations to the right are not to be entered or used. Instructions to enter will be *italics* for emphasis.

<i>!POBJ Clear</i>	Removes all controls form the plot screen
<i>!NEWP</i>	Starts a new plot – default configuration
<i>!SPAN -100,100</i>	Sets the analog (Y-axis) scales
<i>!TMAX 600</i>	Sets maximum time for 600 minutes
<i>!RTIM ON</i>	Enables Real-Time on X-axis
<i>!SHFT ON</i>	Enables plot shifting at maximum
<i>!FLSH ON</i>	Enables data flushing
<i>!TITL Practice</i>	Titles the plot window
<i>!PLOT ON</i>	Enables plotting
<i>!RSET</i>	Resets the plot
<i>!STAT This is a message</i>	Places a message in the User Status text box in the plot.
<i>!DEBUG Hello!</i>	Displays data in the Debug Window.

Instructions that use ON/OFF may also use 1/0:

<i>!CONN 1</i>	Connect on COM port
----------------	---------------------

These control instructions may also be part of the BASIC Stamp code as a DEBUG instruction.

```
` Plotting 1 analog value & using control instructions
PAUSE 100
DEBUG CR,"!POBJ Clear", CR
DEBUG "!NEWP", CR
DEBUG "!PLOT ON", CR
DEBUG "!SHFT ON", CR
DEBUG "!RSET", CR
Val    VAR Byte
Start:
  FOR Val = 0 to 255
    ` Plot Value
    DEBUG DEC VAL, CR
    ` Show value in Status box
    DEBUG "!STAT Value = ", DEC Val, CR
    PAUSE 1000
  NEXT
GOTO Start
```

As always, watch the BASIC Stamp DEBUG Windows to verify the strings look well formed.

NOTE: When StampPlot connects it will cycle the DTR line causing the BASIC Stamp to reset. This is important to ensure the configuration information at the beginning of the program is sent. If the DTR Enabled option is unchecked you may need to manually reset your BASIC Stamp.

To insure there is not a portion of a previous line (from resetting your stamp in the middle of sending) in the StampPlot queue, always start your 1st DEBUG with a CR to end the sting portion.

The StampPlot Help files, Summaries, provides a full listing of available control instructions and use.

Section 4: Plotting Analog Values with Binary Values

Data is sent as 1's and 0's represented by different voltage levels from the BASIC Stamp to the computer. Occasionally a glitch may occur causing perhaps the number 100 to be 300 just because 1 bit (1 or 0) out of 24 was in error (each character in the number 100 is represented by 1 byte which is 8 bits, giving 24 bits for 3 characters). This doesn't happen often (hopefully!) but when it does it can cause problems if you are performing important data collection.

StampPlot provides for the means to use each byte as a unique value and a process called Checksum to verify the integrity of the data. One byte may represent a value between 0 and 255.

We will test this with 3 analog values to be plotted with checksum verification. The first step is to configure StampPlot to use data in this format.

- Under the Configuration Window, Data Tab, select:
 - Use Binary Data
 - Number of bytes per data set: 3
 - Use Checksum must be checked.
- OR, use StampPlot control instructions in the CLI.
 - !USEB ON

- !NUMB 3
- !CSUM ON

This configures StampPlot to expect 3 byte values plus a 4th for the checksum value. To understand what is occurring consider the example if our data were 10, 20 and 30 for the 3 values. Instead of sending a character for 1 then a character for 0 (2 bytes) for the value 10 it sends a single byte of the value 10 and similarly for 20 and 30. The largest value that may be sent is 255 since that is the maximum value for a byte.

Checksum means that the values of the individual bytes in each data set (or packet) are added up and that is sent as a byte. What would the checksum value be in this case? If you said 60 you'd be correct (10+20+30). When StampPlot sees that it has received 4 bytes it adds up the 1st 3 and compares what it calculated to the 4th. If there is a difference StampPlot will not use the data set and issue an error message in the Immediate/Debug window.

You may often get checksum error messages when first connecting due to connecting in the middle of a packet but StampPlot will quickly recover and find good packets.

Let's write a program to send data for this configuration:

```

` Plotting 3 analog values from binary data
Val  VAR Byte
Val2  VAR BYTE
Val3  VAR BYTE
Start:
  FOR Val = 0 to 255
    Val2 = Val / 2
    Val3 = Val * 2
    DEBUG Val, Val2, Val3, Val+Val2+Val3
    PAUSE 100
  NEXT
GOTO Start

```

Note the difference from the previous line to plot 3 values:
 DEBUG DEC VAL, ",", DEC VAL2, ",", DEC VAL3, CR

- It does not use the DEC modifier.
- It does not use the quoted comma-separators (the BASIC Stamp editor requires a comma between each value sent)
- The last value sent is the sum of the 3 bytes.
- No CR is sent at the end. This would be a 5th character sent (a value of 13).

Since a byte can only hold values up to 255, what happens if we send the values of 255,2,2? The checksum value will 'roll over' and start back at 0, so 255+2 = 1 +2 = 3. Don't worry though, the BASIC Stamp and StampPlot know this well and take care of it for you.

NOTE: Because data is sent in predefined packet sizes and the bytes represent values, you CANNOT send any other forms of data directly, such as data to be plotted as binary or control instructions once StampPlot is in binary mode.

Section 5: Logging and Saving

Logging Data and Messages

StampPlot allows data and messages to be saved to text files. The data is saved as comma-separated values and optionally time stamped.

The choices can be found under the Logging menu. Choices include saving data and messages to files and opening data and message files. Files are saved to the StampPlot Data directory. StampPlot control instructions for these are:

Files may be named with the instructions:

<i>!SAVD ON</i>	Enable saving of data file.
<i>!SAVM ON</i>	Enable saving of data file.
<i>!NAMD filename.txt</i>	Name of data file to save to.
<i>!NAMM filename.txt</i>	Name of message file to save to.

As a reminder these instructions may be sent by the BASIC Stamp to ensure your data collection in on.

Saving Plots and Snapshots.

Saving a plot is saving the configuration of the plot and the current data points and messages in the message window. A plot may then be reopened and analyzed. If data flushing is on only current data in the data points will be saved for the plot. By clicking the File save button a directory and file save choice will open. By default saved files are to the data directory.

!SAVP filename

StampPlot can also save a jpg image of the plot, a snapshot, by clicking the camera. The image is saved to the data directory and appended with the date and time.

!SNAP filename

Plot files and snapshot saves can be configured to be automatically saved also.

!ASAV ON

!ASNP ON

There are choices to perform these when the data points are at maximum (*!MAXP*) or when the plot reaches the maximum time before shifting (*!MAXT*).

The date and time are automatically appended to the file names unless disabled

!APDT OFF

By default only the plot area information is saved as a file or a snapshot. The entire form, including controls, may be saved by enabling the choice under the File menu or using the instruction.

!FORM ON

Section 6: Use drawing, image and sound files for effects.

Drawing

StampPlot allows drawing of shapes and placing text on the plot. In general, the structure is: The drawing type + instruction, coordinates, parameters.

For example:

@TEXT 35a, 102a, 2, (BLUE), StampPlot Rules!

Drawing type

StampPlot supports 4 different types of drawing depending on what you want to produce.

- Starts with @: Drawing is 'constant' and will survive a reset. Good if you want something on the plot permanently even if you reset.
- Starts with ~: Drawing is 'temporary' and will be erased with resets, or anytime the plot shifts or is refreshed.
- Starts with ^: Drawing is treated as a data points and will be plotted just as data is when the plot is shifted. As data, it is erased when the plot is reset. Plotting must be enabled.
- Starts with !: Drawing is the same as using ^ but the drawing is restricted to the data plotting area. Plotting must be enabled.

Test the following in the Immediate/DEBUG window's CLI:

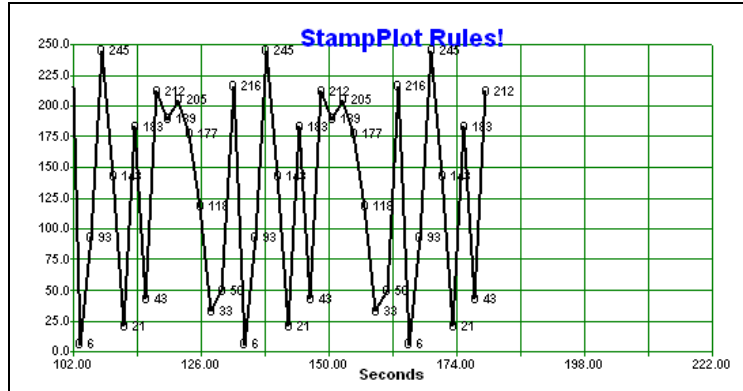
```
!POBJ Clear  
!NEWP  
!PLOT ON  
@TEXT 35a, 102a, 2, (BLUE), StampPlot Rules!  
~FREC 10,25,30,150,(RED)  
^FCIR 96,125,5,(GREEN)  
!RECT 48,50,72,175,(BLACK)
```

Now shift around the plot and compare what occurs as the plot is moved. Why didn't the text move? The objects change position because their coordinates are based on the plot coordinates. Using absolute coordinates, you can plot using points independent of the plot scales, where lower left is 0a, 0a, and upper right is 100a, 100a.

Of course, these instructions can come from the BASIC Stamp using DEBUG. Try this program from your BASIC Stamp:

```
'Plot and mark values of points  
PAUSE 1000  
DEBUG CR,"!POBJ Clear",CR  
DEBUG "!NEWP",CR  
DEBUG "!PLOT ON",CR  
DEBUG "@TEXT 35a, 102a, 2, (BLUE), StampPlot Rules!", CR  
  
x VAR Byte  
x = 100  
DO  
  Random x  
  ' Plot value  
  DEBUG DEC x,CR  
  ' Mark data with value using text  
  DEBUG "!TEXT (PTIME),", DEC x,"1,(black),O ", DEC x,CR  
  PAUSE 2000  
Loop
```

(PTIME) in the text instruction substitutes the current plot time in seconds for the X coordinate. As always, monitor in the Stamp Editor's debug window to ensure the format is correct.



Sound Files

StampPlot can play .WAV files to add sound effects to your program. There is an assortment in the StampPlot/Media directory. The sound files are treated just like the drawing instructions. By default StampPlot looks in the media directory for the sound file unless otherwise specified.

~PWAV clap

IWAV may also be used. This instruction will stop the prior wave file to play this one immediately. Compare the following two sets using the CLI:

~PWAV clap(CR)~PWAV boing

~PWAV clap(CR)~IWAV boing

Can you modify the last program to 'boing' every time a point is plotted?

Image Files

StampPlot has a collection of jpeg (.jpg) image files that can be used to add images to your plot. These are again treated as drawing instructions using coordinates. The media directory is the default directory for images:

!POBJ Clear

!NEWP

!PLOT ON

^IMGP 80a,100a,90a,110a,comp\led_red_1.jpg

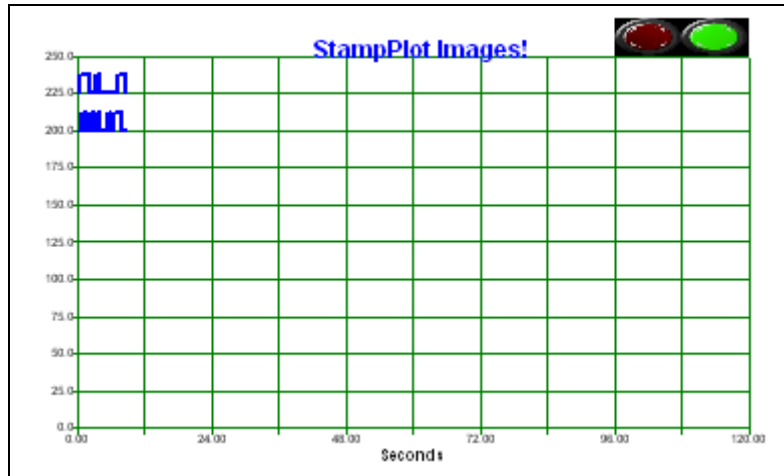
^IMGP 90a,100a,100a,110a,comp\led_grn_0.jpg

Again, the instructions may come from the BASIC Stamp:

```
'Plot digital and place 2 images
PAUSE 1000
DEBUG CR,"!POBJ Clear",CR
DEBUG "!NEWP",CR
DEBUG "!PLOT ON",CR
DEBUG "@TEXT 35a, 102a, 2, (BLUE), StampPlot Images!", CR

x VAR Byte
DO
  RANDOM x
  ' plot digital values
  DEBUG IBIN2 x,CR
  ' Place LEDs on plot
  DEBUG "^IMGP 80a,100a,90a,110a,comp\led_red_(BIT0).jpg", CR
  DEBUG "^IMGP 90a,100a,100a,110a,comp\led_grn_(BIT1).jpg", CR
  PAUSE 500
Loop
```

(BIT0) and (BIT1) are replaced when used by StampPlot with 1 or 0 depending on the status of those bits plotted respectively.



Section 7: Place controls on the interface for monitoring and control.

Creating a Plot Object Control

Controls, such as text boxes, buttons, gauges and so on may be placed on the StampPlot background once the plot is resized.

```
!POBJ Clear  
' Size plot to 70% by 80% of window.  
!PPER 70,80  
!NEWP
```

A new plot object control is created by defining the type, naming it, setting the coordinates (the background is 0,0 to 100,100) and setting parameters.

To create a new meter the format is:

!POBJ oMeter.objName=L,T [,W,H,scale min, scale max,alarm min, alarm max]

Where oMeter means to use a meter control plot object.

ObjName is what you want to name it.

L = Left Coordinate of meter.

T = Top Coordinate of meter.

[] indicates these parameters are optional.

W = Width of the meter.

H = Height of the meter.

Scale min = The minimum value of the meter.

Scale Max = The maximum value of the meter.

Alarm Min = The lower alarm set point.

Alarm Max = The upper alarm set point.

So, to create a meter called meter1 at 75 left and 50 top using default width and height, and range from 0 to 255 with alarm set points at 25 and 200 (ensure you run previous code first):

```
!POBJ oMeter.Meter1=75,50,,,0,255,25,200
```

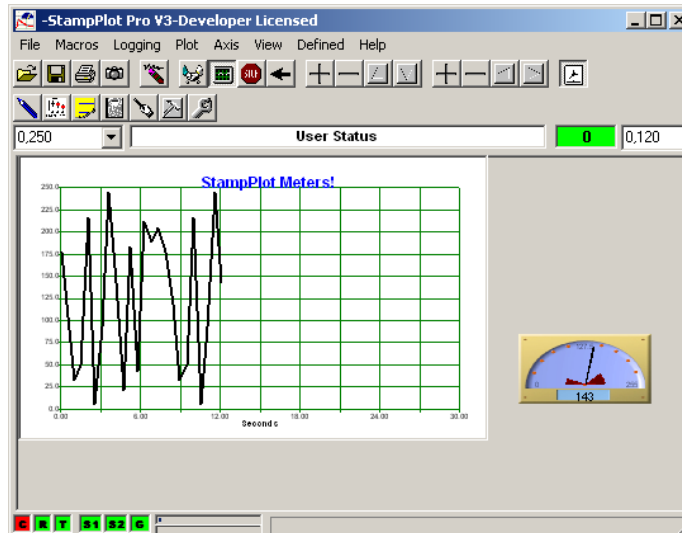
Meter1 may be updated by giving it a new value (notice how its name is used):

```
!POBJ Meter1= 100
```

BASIC Stamp Code to run the meter may be as follows:

```
'Plot and use meter
PAUSE 1000
DEBUG CR,"!POBJ Clear",CR
DEBUG "!NEWP",CR
DEBUG "!PLOT ON",CR
DEBUG "@TEXT 35a, 102a, 2, (BLUE), StampPlot Meters!", CR
' Size the plot
DEBUG "!PPER 70,80",CR
' Create meter
DEBUG "!POBJ oMeter.Meter1=75,50,,,0,255,25,200",CR

x VAR Byte
x = 100
DO
  RANDOM x
  ' Plot value
  DEBUG DEC x,CR
  ' Update meter
  DEBUG "!O Meter1=", DEC x,CR
  PAUSE 500
Loop
```



Notice the use of !O. This is short hand for !POBJ to save typing and code space. Also note the update string must be sent each time the value changes. This can be automated using an Update Value for the control object.

Update Values

StampPlot keeps track of all kinds of different values such as earlier when (PTIME) was used for the current plot time, and (BIT0) was used for the last digital bit 0 received. These are termed Macro Math Values.

Another is (AINVAL0) to (AINVAL99) for a set of analog values received. We can set the meter to automatically use the value when updated by setting an Update value:

```
!O Meter1.U=(AINVAL0)
```

Now, when the instruction to update is issued, the meter will be updated with this value.

```
!O Update
```

```
'Plot with meter and update value
PAUSE 1000
DEBUG CR,"!POBJ Clear",CR
DEBUG "!NEWP",CR
DEBUG "!PLOT ON",CR
DEBUG "@TEXT 35a, 102a, 2, (BLUE), StampPlot Meters!", CR
' Size the plot
DEBUG "!PPER 70,80",CR
' Create meter
DEBUG "!POBJ oMeter.meter1=75,50,,,0,255,25,200",CR
' Set an update value
DEBUG "!O Meter1.U=(AINVAL0)",CR

x VAR Byte
x = 100
DO
  RANDOM x
  ' Plot value
  DEBUG DEC x,CR
  ' Update all plot object controls
  DEBUG "!O Update",CR
  PAUSE 500
Loop
```

While this doesn't seem to save a lot, if you had 3 meters, text boxes, and other various controls, you would be able to update them all at once using *!O Update*.

ALARMS – Adding Event Code

Event code is StampPlot code that is ran when an event occurs. An event is a meter hitting an alarm level, a button being clicked, a text box having its text changed, or a variety of other actions that may take place with the plot object controls.

Event code is written by specifying the code to take place for the object, such as our meter:

```
!O Meter1.C=~PWAV boing
```

Multiple instructions can be issued by separating them with (CR). If you are using multiple lines, a (;) may also be used (you'll see this in macro text files).

```
!O Meter1.C=~PWAV boing(CR)!STAT ALARM AT (RTIME)
```

Where (RTIME) is the real time of the plot (the computer's hours:minutes:seconds).

Can you add the event code to the previous program? (hint: It belongs after the meter is created).

Please look through the help files on other plot object controls and their use, and of course try some out!

Section 8: Performing Math Operations.

StampPlot can perform math operations above and beyond what the BASIC Stamp can handle, including full floating point math. Here are some rules for using math:

- Math operations are enclosed in square brackets [].
- Only one operation per bracket.
- Math is performed inner bracket to outer bracket.
- Commas separate values and operators.

Take for example:

```
!STAT [100/,5]
```

This will calculate and display in the status text box the value of $100 / 5$ when entered.

Or, a little more complex:

```
!STAT [[100/,5],-20]
```

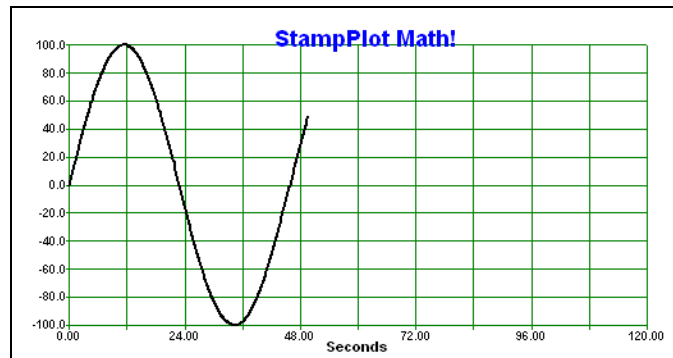
Will calculate $100/5 - 20$.

The BASIC Stamp can send data formatted to have math operated on it prior to being plotted or used in other ways:

```
'Plot sine wave
PAUSE 1000
DEBUG CR,"!POBJ Clear",CR
DEBUG "!NEWP",CR
DEBUG "!PLOT ON",CR
DEBUG "!SPAN -100,100",CR
DEBUG "@TEXT 35a, 102a, 2, (BLUE), StampPlot Math!", CR

x VAR WORD

DO
  FOR X = 0 TO 360
    ' Plot sine value * 100  [[value,SIN],*,100]
    DEBUG "[[" ,SDEC x," ,SIN] ,*,100]",CR
    PAUSE 100
  NEXT
Loop
```



Section 9: Configure from macros (text scripts).

Of course, all this great stuff of drawing, graphics, sounds, controls and math takes up room on your BASIC Stamp, and in some cases you may not be able to modify the controller code to perform such operations.

Macros are simply text files that contain configuration information and can be used for the processing of data. Below is a simple macro.

- Use the Macro Menu →Edit Macro with NotePad
- Provide a name, such as macroTest
- Answer yes to create (twice).
- Place text in file.
- Save file.
- Use the Macro menu→Run Macro, select the macro and open it.

```
'Macro to plot and show value in meter

INIT:
' Define this macro as the default macro for data
!DEFS (ME)

' Clear all plot objects
!POBJ Clear

' Start a new plot
!NEWP

' Span the Y-axis
!SPAN -100,100

' Enable plotting
!PLOT ON

' Place graphic text
@TEXT 35a, 102a, 2, (BLUE), StampPlot Meters!

' Size the plot
!PPER 70,80

' Create meter
!O oMeter.Meter1=75,50,,, -100,100
' set update value, format for 2 decimal places
!O Meter1.U=[(AINVAL0),FORMAT,0.00]

' Use default routine when data arrives
!USED ON
ENDMAC

'Routine ran when analog data arrives
DEFAULT:
' Update plot object controls
!O UPDATE
ENDMAC
```

This macro has two routines in it. INIT and DEFAULT and each ends with ENDMAC. INIT is run when the macro is opened. By specifying !USED ON, the DEFAULT routine is ran when analog data arrives. ALL comments must be on separate lines from the code in macros.

Here is a simple BASIC Stamp program to test it.

```
x VAR WORD

DO
  FOR X = 0 TO 360
    DEBUG DEC x,CR
    PAUSE 100
  NEXT
Loop
```

Now, the combination of the Macro and BASIC Stamp plot the values and display in the meter from 0 to 360.

But what if we want the SIN values plotted instead?

Manipulating Analog Data Before Plotting

We can stop StampPlot from automatically plotting the incoming analog data and manipulate the data before plotting it.

At the end of the INIT: routine, before ENDMAC, add this:

```
` Use analog data for macro only - do not plot
!USEA ON
```

Change the DEFAULT Routine to this:

```
'Routine ran when analog data arrives
DEFAULT:
` Plot analog channel 0 in red
!ACHN 0, [(AINVAL0),SIN],*,100], (RED)
` Update plot object controls
!O UPDATE
ENDMAC
```

!ACHN tells StampPlot to plot the value on a channel (0-9) in color specified.

!ACHN 3,100,(BLUE)

Other Analog Data Processing Methods

Besides use of the Default macro, object controls can be used to be trigger event code when analog data, digital data or message data is received. This is performed using a specially structured name for these object controls.

Name beings with:

DA_ Event code will be processed when analog data arrives.

DB_ Event code will be processed when digital data arrives.

DM_ Event code will be processed when messages for the message list are received.

An example for analog data:

```
' DA_Hidden -- OBUTTON *****
!POBJ oButton.DA_Hidden=72.,73.,10.,5.,Obj10,8
!POBJ DA_Hidden.V=0
'-- Event Code
!POBJ DA_Hidden.C='Local Channels(;)
(;)
'Plot on channel 0 1st byte(;)
^AchN 0, [(AINVAL0),*,0.02], (ORANGE) (;)
(;)
'plot on channel 1 2nd byte(;)
^AchN 1, [(AINVAL1),*,0.02], (RED) (;)
(;)
'plot on channel 2 3rd byte(;)
^AchN 2, [(AINVAL2),*,0.02], (BLUE) (;)
!IFTH [(AINVAL1),==,(AINVAL2)],==,1,~pwav beep(;)
!POBJ UPDATE
```

- A button is placed on the plot and named DA_Hidden.
- The button is set to be hidden – not visible (DA_Hidden.V =0)
- When analog data arrives the event code will be ran and 3 channels of analog data will be scaled by multiplying each value by 0.2 and plotted in Orange, Red and Blue.
- If analog channel 1 is equal to analog channel 2 (the logical result of comparing them, 0 or 1, is equal to 1 the beep wav is played.

- The plot objects are updated with current values. This example is used by the Stamps In Class text – “Experiments with Renewable Energy” by John Gavlik to convert binary data (0-255) to 0 to 5 Volts.

One other way to trigger code on arrival of data is the use of the oAnalog object introduced in StampPlot Version 3 Release 2.

```
oAnalog.C = ~PWAV beep (:)
!STAT Value is (AINVAL0)
```

Macros (.spm files) may be opened directly by double-clicking them from Windows Explorer or from an Internet link. This will load StampPlot and bring up the macro.

Section 10: Use an interface for interactive control of your BASIC Stamp.

The !READ instruction may be used to send data BACK to the BASIC Stamp where the BASIC Stamp would accept and use the value. Treating their name like a macro math value references object values. For example, a slider control named 'Setpoint' could have its value sent to the BASIC Stamp using:

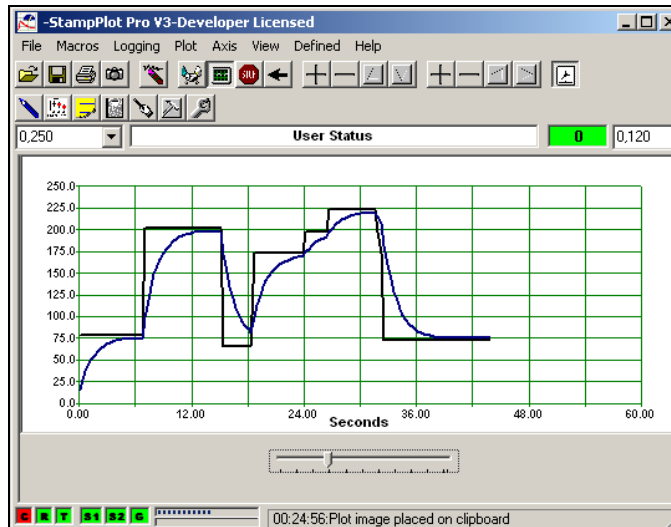
```
!READ (Setpoint)
```

We will have the BASIC Stamp create and read this interface with a slider with a range from 0 to 255 and adjust a value to match the slider's.

```
' Interactive control with a StampPlot slider
PAUSE 1000
DEBUG CR,"!POBJ Clear",CR
DEBUG "!NEWP",CR
DEBUG "!PPER 100,80",CR
' create slider on StampPlot named Setpoint
DEBUG "!POBJ oHslider.Setpoint=38,15,29,7,0,255,78",CR

Actual   VAR Byte
Setp     VAR Byte

Do
  ' Request value from StampPlot
  DEBUG "!READ (Setpoint)",CR
  ' Accept value from StampPlot
  DEBUGIN DEC Setp
  ' Compare and adjust actual
  IF Actual < Setp THEN Actual = Actual + ((Setp-Actual)/5)
  IF Actual > Setp THEN Actual = Actual - ((Actual - Setp)/5)
  ' Plot setpoint and actual
  DEBUG DEC Setp, ",", DEC Actual,CR
  PAUSE 250
Loop
```



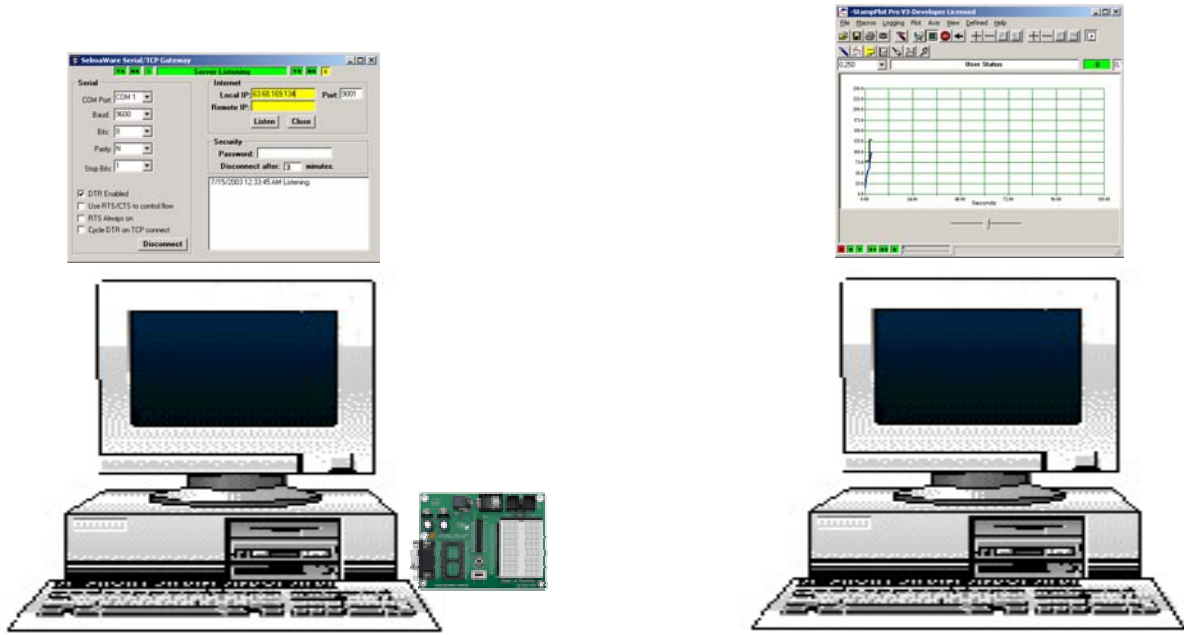
Note: It is recommended to use the SERIN BASIC Stamp command with a timeout instead of DEBUGIN instruction. With DEBUGIN, if a response is not received (StampPlot not connected) the controller will 'hang' waiting for a return.

```
SERIN 16, 84,500,Timeout, [DEC Setp]
Timeout:
```

Another fun use is image buttons that display one of 2 graphics when clicked and return a 1 or 0 when read and can be used as virtual switches. See the help files!

Section 11: Perform acquisition and control over the Internet.

StampPlot also installs a program called the TCP-Serial Gateway. This program allows monitoring and control over the Internet by creating a link (gateway) between your serial port and your Internet connection. The typical configuration is shown:



Local - Stamp/Gateway

Remote - StampPlot

Local Configuration:

1. On the local computer, the BASIC Stamp is connected and programmed for use with StampPlot (most any program will do).
2. Open the TCP-IP Gateway program.
3. Connect to the serial port (select cycle DTR if you want your BASIC Stamp to reset on a connection).
4. Connect Listen on the Internet side.
5. Record the Local IP address shown.

Remote Configuration:

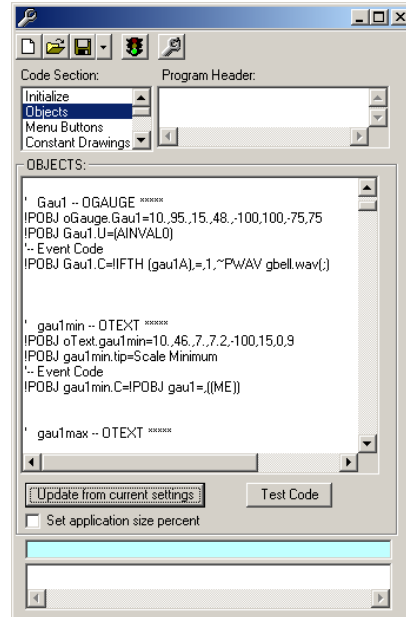
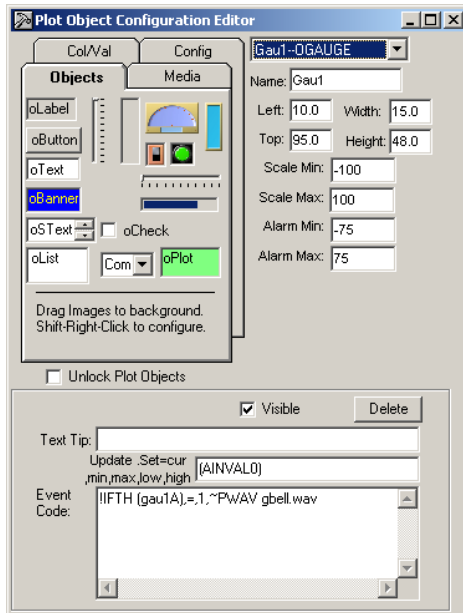
1. Open StampPlot
2. Open the Configuration Window.
3. On the Port Tab, select TCP.
4. Enter the IP address from above.
5. Click the Connect and Plot buttons.
6. Watch it plot and say wowwwwwwwwww.
7. Interactive control works also!

If you have problems connecting, check with your network administrator, there may be a firewall. StampPlot TCP gateway does not support proxy-servers currently.

One hint is if you are trying to perform interactive control by sending data back to the BASIC Stamp is to increase the timeout values of SERIN since the connection will not be as fast as straight serial. A good value is 500 to 1000 for timeout.

Developers License

A Developers license allows the use of the Plot Object and Macro Editor to Drag-and-Drop objects, program parameters and event code and automatically build macros from the current configuration and is available through Parallax.



This guide touched briefly on many aspects of StampPlot. The help files have plenty of more great information on StampPlot Pro.

Licenses available at:

www.parallax.com

www.stampplot.com

and many other BASIC Stamp distributors.

For support needs concerning StampPlot please contact:

support@selmaware.com

Or join our Yahoo Group at:

<http://groups.yahoo.com/group/selmaware/>

Happy Plotting!

Martin Hebel

SelmaWare Solutions