The BASIC Stamp II® and LabVIEW®

Prof. Clark J. Radcliffe, PhD
Department of Mechanical Engineering
Michigan State University
East Lansing, MI 48824-1226
radcliffe@me.msu.edu

Overview

LabVIEW® is a general-purpose data acquisition, analysis and graphics software package developed by National Instruments, Inc. of Austin, Texas. It is in common use in government, academia and industry to interface with control and monitoring systems. It uses a novel graphical interface language "G" for programming. Because of its common use, BASIC Stamp® programmers and application developers may find it useful to have the capability to interface BASIC Stamp® devices to a personal workstation (PC, Mac, Unix). This document will present a prototype BSII-to-LabVIEW interface. The interface relies on a Parallax Board-of-Education for cable connections but can be used with any BSII with the addition of a suitable serial interface cable.

The document is divided into (3) sections: 1) an Introduction explaining the structure of the software system, 2) the BSII software and hardware interface and 3) the LabVIEW interface software. The software will be developed in each section as a multi-step process step incrementally developing the final interface so that users of the system can debug and verify all components of the two interfaces.

Introduction

Both LabVIEW® (Laboratory Virtual Instrument Engineering Workbench) and the BASIC Stamp II® (BSII) have well developed serial interface support. The software described here will use that serial interface support to allow the transfer of commands and data between a LabVIEW® workstation host and a BASIC Stamp II® microcontroller. The easiest development hardware to use for interface software development is the serial cable used by the standard BASIC Stamp® software development packages. For the prototype developed here, the Parallax Board of Education (Figure 1) will be used and a low-cost analog and digital interface to the LabVIEW® workstation software generated.

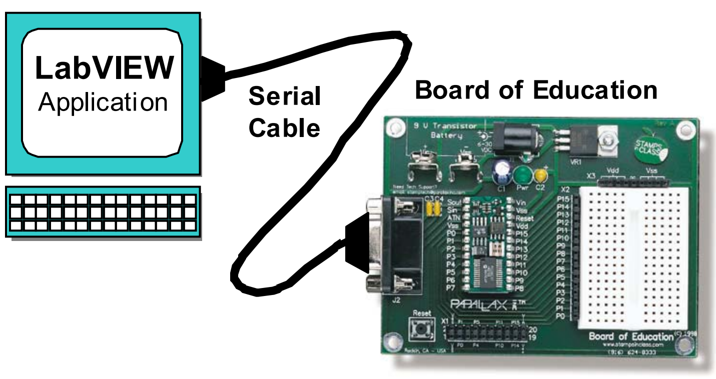


Figure 1: BASIC Stamp® to LabVIEW® interface cable is the Serial Cable used for BASIC Stamp® Software development.

Commands and data will be transferred across the serial cable as variable length strings of 8 bit ASCII encoded characters. Commands will be issued to the serial interface from LabVIEW® to the BSII and responses from the BSII returned via the serial interface to LabVIEW® where it is displayed. The development of software for the BSII can be conducted independently through the use of the "debug" window which forms a part of the Parallax BASIC Stamp® Editor for windows. Before starting, you should have the software and hardware components shown in Table 1 in-hand and installed.

Item	Description	Preferred
1.	BASIC Stamp® Development System	Board of Education + PC
2.	Serial Cable	PC to BOE
3.	Workstation (PC, Mac, UNIX)	LabVIEW Compatible
4.	LabVIEW Software	Version 5.1 or later
5.	Analog Interface Components	See Table 2

Table 1: Required Software and Hardware to Generate a LabVIEW to BSII Interface

The BSII Software And Hardware Interface

The BSII system will be configured to perform as a simple analog and digital interface for LabVIEW. The software system to support this hardware will be developed in stages. First, simple serial interface software will be generated and tested on the Board of Education and then software functions added to support the analog and digital hardware for input and output.

Simple Serial Interface Software for the BSII

Serial I/O on the BASIC Stamp® is done using the SERIN and SEROUT commands. These commands are documented fully for the BASIC Stamp II® in the *BASIC Stamp® Programming* Manual (Version 1.9), pp 307-329. Similar commands are present in the BASIC Stamp® I firmware. The software used here has been tested only on the BSII.

Serin

SERIN *rpin{\fpin},baudmode,{plabel,}{timeout,tlabel,}[inputData]*

Receive asynchronous (e.g., RS-232) data.

Serout

SEROUT *tpin,baudmode,{pace,}[outputData]*
SEROUT *tpin\fpin,baudmode,{timeout,tlabel,}[outputData]*

Transmit asynchronous (e.g., RS-232) data.

```
'LabVIEWSerial.BS2       C.J. Radcliffe   01/01/01
'Establishes serial I/O to and from a BASIC Stamp®

baudmode con 84       'serial code to set baudrate (pg.321, BASIC Stamp® Manual)
                      'For pin 16 (noninverted): 9600=>84, 19200=>32, 38400=>6
                      'For pins 0-15 (inverted): 9600=>16468, 19200=>16416, 38400=>16390

'Initialize
indata  var byte(6)    'data string
i              var byte
indata(6) = 0
i = 0

debug cls, "starting...", CR

'Basic I/O loop
top:
  SERIN 16,baudmode,[str indata\5\CR] 'get command
  SEROUT 16,baudmode,[str indata, ", index= ",dec i, CR]    'Send response
  i = i+1
goto top
```

The BASIC Stamp® firmware is capable of transmitting serial data over any stamp digital pin, however, use of the standard serial interface pins (Sout and Sin) makes the interface easier and will be presented here. The hardware connection is simply then normal BSII programming cable. The first software establishes BSII capability to receive a character string and transmit a response to the host computer connected via a serial cable.

The "LabVIEWSerial.BS2" program initializes a string array then enters a loop where command strings are alternately received and then echoed via the BSII's serial interface.

To use the software, first download it into your BSII then access the debug screen. To transmit characters simulating a command, click on the division between the upper white pane and the lower blue pane of your debug window to expand the white "transmit" upper half of your debug window until it has both a "transmit" and "receive" pane are approximately the same size (Figure 2).

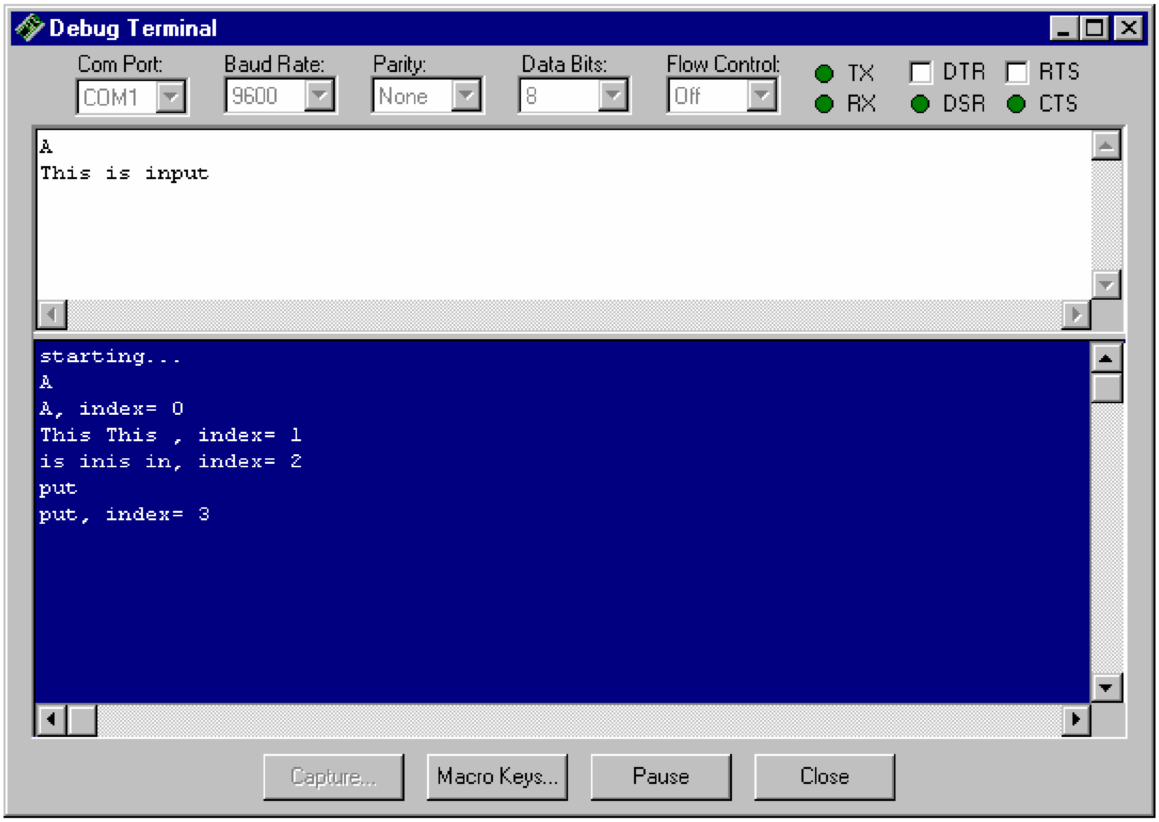


Figure 2: Debug Screen with LabVIEWSerial.bs2 running

Click on the upper, white, transmit pane and type a single "A" character. The character "A" will be echoed by the BSII's serial interface on the lower, blue, receive pane of the debug window as it is received by the SERIN command in the program. The BSII echoes all received characters. Now push the enter key. The SERIN command is waiting for either a string of 5 characters or a terminating "CR" character to complete the input string. The BSII then echoes the "indata" characters as received and a "CR" carriage return character ending the line on the blue receive pane. After SERIN receives a string, the BSII stores the received string in "indata". SEROUT retransmits the string received along with the loop index to demonstrate that the BSII has actually processed the input string data. You should experiment with the properties of this simple I/O program and also note the effect of trying to enter more than (5) characters.

In Figure 2, the string "This is input<CR>" was typed with the result shown. The BSII received the string in (5) character groups and output each group through the SEROUT command in the program with the result shown. Remember the "space" character counts as one character in a group. The third line in the lower blue "received" window is "This This , index= 1" that reflects the echo of "This<sp>" as the characters are sent to the BSII followed by the result of the SEROUT command "This<sp>, index= 1". The fourth fifth and sixth lines continue echoing 5 character groups including the <CR> character that terminates the (3) remaining characters "put" in the input string. The LabVIEW Serial.bs2 software demonstrates serial communication and will be used below as the basis for a simple analog/digital I/O system.

Simple Analog/Digital I/O System for the BSII

Construct the simple Analog/Digital I/O system (Fig. 3). A convenient place is the prototyping area on the Parallax Board of Education (BOE). The potentiometer (pin 1) is an analog input, the capacitor-resistor circuit (pin 2) will set an analog output, the switch (pin 3) is a digital input and finally, the LED (pin 4) forms a digital output. In the *BASIC Stamp® Manual (BSM),* Version 1.9, see BASIC Stamp II® RCTIME documentation, pg. 298-301 for information on the analog input circuit. For information on the analog output circuit, see the BASIC Stamp® I PWM Application Note in *BSM*, pg 143. Digital inputs are discussed on pg. 276-277 and digital output on pg. 284. You may also find the information in the Parallax book *Introduction to Microcontrollers* useful. All documents are available at the Parallax Stamps-in-Class website http://www.stampsinclass.com/.

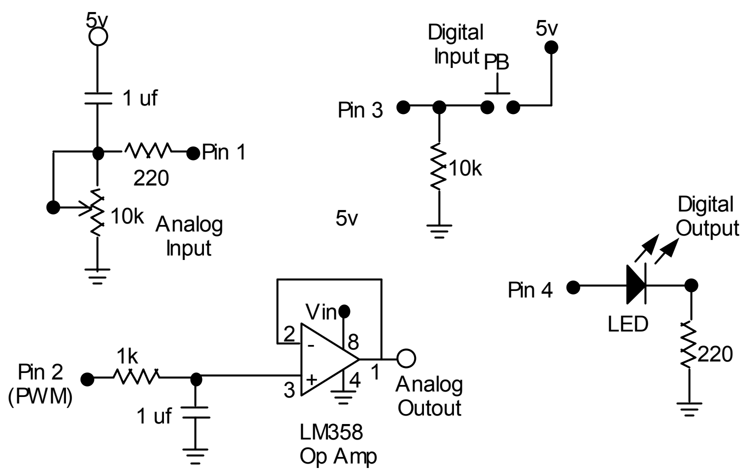


Figure 3a: Simple Analog-Digital I/O System Schematics

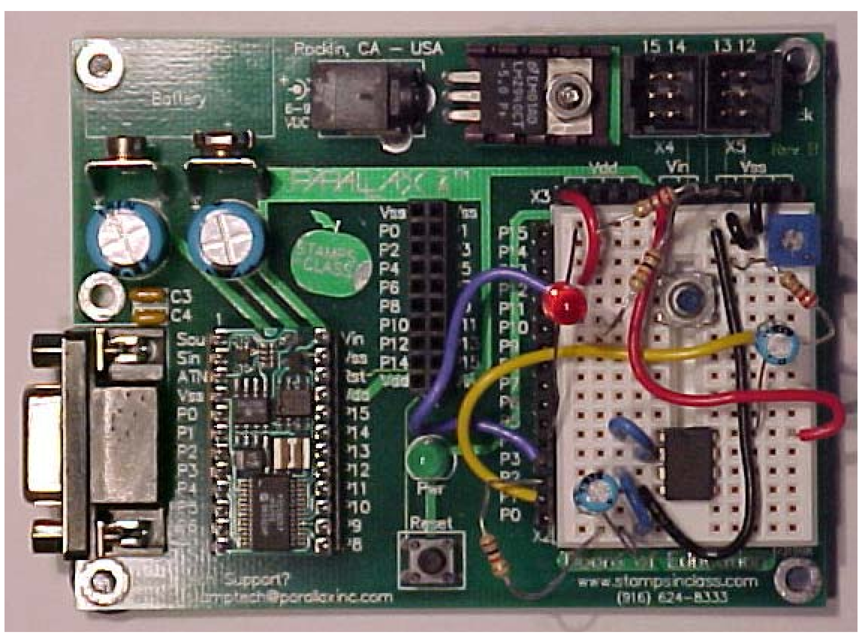


Figure 3b: Simple Analog-Digital I/O System Wiring on Board of Education

The BASIC Stamp® program "LabVIEWIO.bs2" is designed to use the above hardware to form a simple analog/digital I/O system. It uses the BSII RCTIME function to measure the potentiometer position, the BSII PWM function to set an analog output voltage on the 1uf capacitor that is then output by the op amp in a voltage follower mode. The pushbutton switch provides digital input. The LED circuit monitors digital output.

A note on the analog output circuit. The PWM function in the code's "AO:" section generates a varying duty cycle pulse train for 100 ms. The duty cycle is set by "val" over the range of 0 to 255 (8 bits). The varying duty cycle pulse train sets the DC voltage on the capacitor proportion to the duty cycle between 0 and 5 volts (Figure 4). Pin 2 is set to high impedance by the code whenever pin 2 is not actually outputting the PWM pulse train. Because both the op-amp and pin 2 have very high impedance, they discharge the 1 uf capacitor very slowly. With the LM358 op-amp, the discharge time constant is over 2.5 minutes. There are higher impedance op-amps available if you need a slower discharge rate. For accurate output, the analog output should be updated regularly by the stamp application. This is the normal condition for a repeating control application. Finally, because the LM358 op-amp cannot supply output voltages as high as its supply voltage, it is operated off the "Vin" supply rather than the regulated 5 volt supply. Operating the LM358 off the regulated 5 volt supply limits the output to about 3.75 volts.

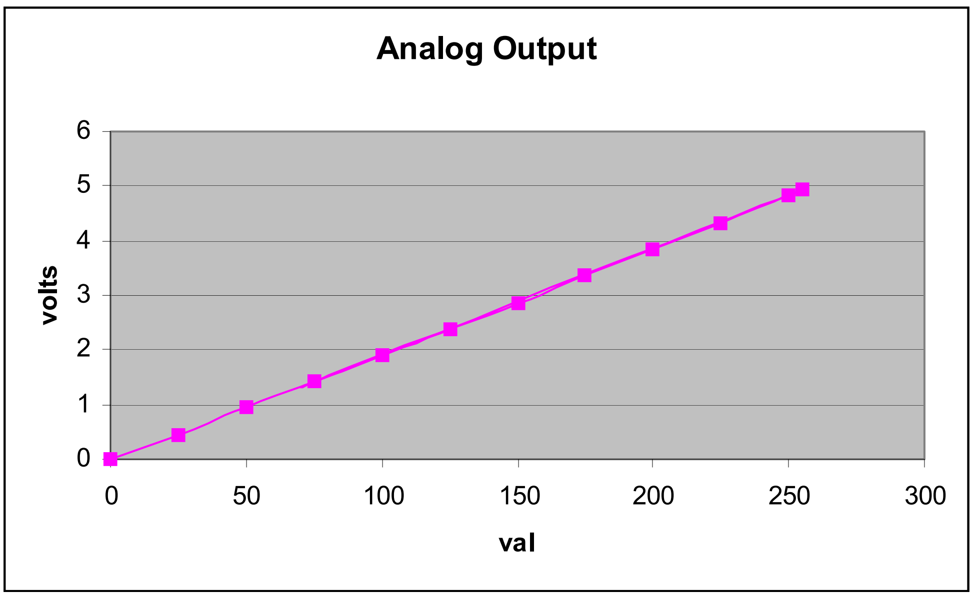


Figure 3: Analog Voltage Output Vs. The Analog Input Code Value "Val"

```
'LabVIEWIO.BS2       C. Radcliffe   03/15/01
'  Establishes serial between a Basic Stamp and LabVIEW with
'  Analog and Digital I/O capability
'        input        output (# is a decimal digit 0-9)
'       "AI<CR>"      "AO###<CR>"
'       "DI<CR>"      "DO#<CR>"

baudmode con 84        'serial code to set baudrate (pg.321, Basic Stamp Manual)
                       'For pin 16 (noninverted): 9600=>84, 19200=>32, 38400=>6
                       'For pins 0-15 (inverted): 9600=>16468, 19200=>16416, 38400=>16390

AIport   con  1        'port for Analog Input
AOport   con  2        'port for Analog Output
DIport   con  3        'port for Digital Input
Din        var IN3     'Digital input bit
DOport   con 4 'port for Digital Output
Dout       var OUT4    'Digital output bit

'Initialize
comm           var byte(3)    'command data char
val            var word       'command value
comm(2) = 0
input AOport                 'Make AOport high impedence when not PWM
Output DOport                'Declare Digital Output port

'LabVIEW command I/O loop
top:
  SERIN 16,baudmode,[str comm\2, dec3 val]      'get command and value
'Signal Data Received...always a 2 char command plus 3 digits
  comm(0)=comm(0)&%01011111  'force 1st char to uppercase
  comm(1)=comm(1)&%01011111  'force 2nd char to uppercase
  IF comm(0)="A" then A            'test first character
  IF comm(0)="D" then D
 goto top                    'on no match, return

A:          'Analog section
  IF comm(1)="I" then AI
  IF comm(1)="O" then AO
  goto top                   'on no match, return

D:          'Digital section
  IF comm(1)="I" then DI
  IF comm(1)="O" then DO
  goto top                   'on no match, return

'Get analog input
AI:
  high AIport
  pause 1
  rctime AIport,1,val               'Measure Analog input
  val=val/8                         'scale to fit in 3 characters
  SEROUT 16,baudmode,["AI ", dec3 val, CR]      'Send acknowledgement
  goto top

'Set analog output
AO:
  PWM AOport,val,100
  SEROUT 16,baudmode,["AO ", dec3 val, CR]      'Send response
  goto top

'Get Digital input
DI:
    SEROUT 16,baudmode,["DI ", dec3 Din, cr]    'Send response
  goto top

'Set Digital output
DO:
    Dout=val                  'Set digital output
    SEROUT 16,baudmode,["DO ", dec3 Dout, cr]   'Send response
  goto top
```

The Labview Interface Software

LabVIEW® uses the graphical programming language "G" to generate programs. It is not the intention of this note to provide LabVIEW® programming instruction. As far as possible, I have used examples provided a standard part of the LabVIEW® development package as the core of the LabVIEW® application presented here. This example uses application software for serial input/output that is found in the directory ".../LabVIEW/EXAMPLES/INSTR/smplserl.llb".

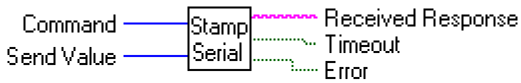
The LabVIEW® library provided as part of this package "StampVI.llb" contains the virtual instrument "VI" to interface with the BSII board. To run this VI, follow the steps below.

Step	**Action**
1	Connect the BSII board to the PC Development system and Run the Stamp editor
2	Load the BASIC Stamp II® software "LabVIEWIO.bs2" into the BASIC Stamp® board.
3	Quit the stamp editor
4	Use a serial (RS232) cable to connect the BSII board to the target LabVIEW® workstation. Remember to power-up your stamp. If you are a PC user, this step repeats step 1.
5	Open the LabVIEW® library "StampVI.llb" and Run the VI "StampSerial". In some LabVIEW systems, it will run automatically when opening the library. For other systems, you may need to select and run it from the contents of the library.
6	Select a command and click on the run arrow. The run arrow is the single full arrow on the upper left corner of the VI's window.
7	The VI will send a command to the BSII, receive a response and display the received response.

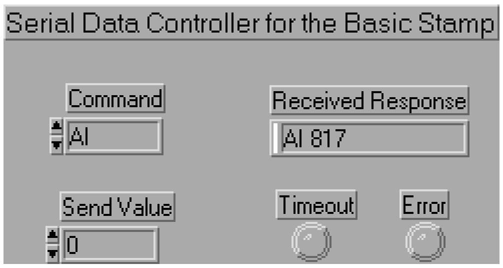
After reading the input from the front panel when run, the LabVIEW® vi sequentially, (Diagram 0, subdiagrams 0 and 1) sends a command string of 2 characters plus 3 numbers, then (Diagram 1) waits a "little"(10 ms), and finally (Diagram 2) receives the response string from the BASIC Stamp® . This VI has been configured so that it can now be used as a sub-vi for more sophisticated applications such as plotting of repeated analog input samples.

Documentation for "StampSerial.vi" Software

Connector Pane



Front Panel



Controls and Indicators

Send Value

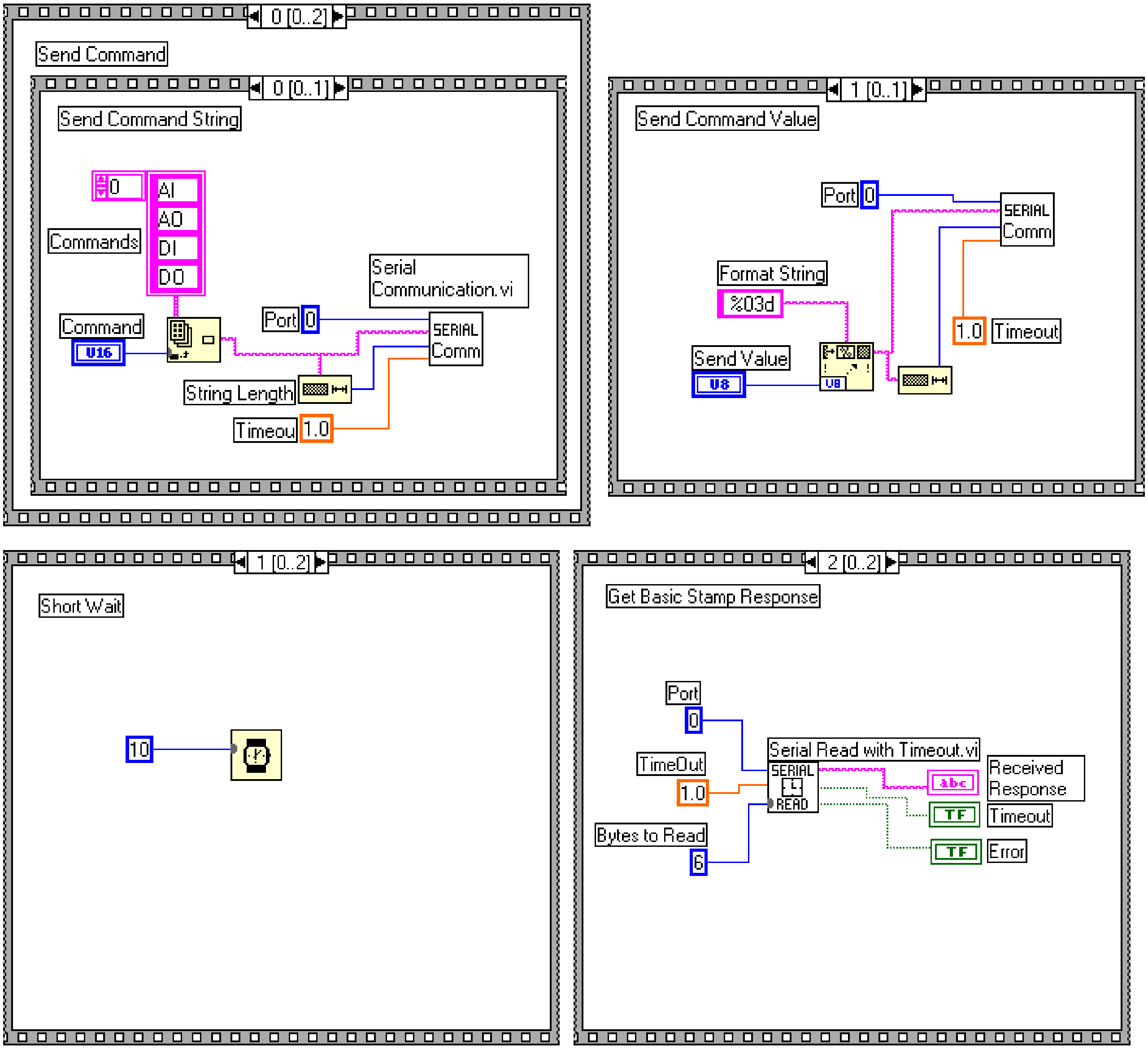
Command

Received Response

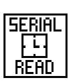
Error

Timeout

Block Diagram



List of SubVIs



Serial Read with Timeout.vi

C:\My Documents\LabVIEW files\USER.LIB\StampVI.llb\Serial Read with Timeout.vi

Serial Communication.vi

C:\My Documents\LabVIEW files\USER.LIB\StampVI.llb\Serial Communication.vi