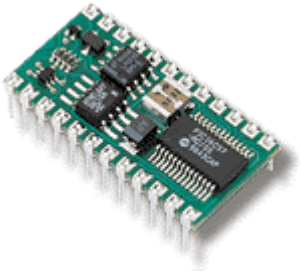
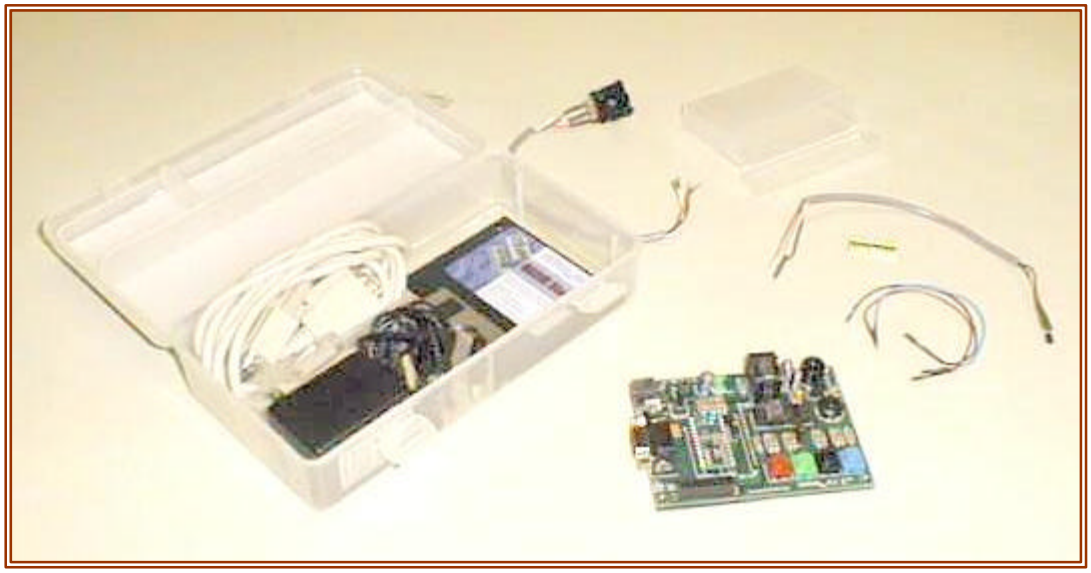
Application of Microcontrollers
Labs
Part I - Principles & The BASIC Stamp®
Version 0.9p





Electronics Management

Department of Information Management Systems
Office of Off-Campus Academic Programs
College of Applied Sciences and Arts
Southern Illinois University, Carbondale

This document is the accompanying labs for the Application of Microcontrollers Manual. If you do not have the manual, please visit the Parallax Stamp-in-Class web site.

Note to instructors:
There is an accompanying version with answers. These labs are currently the same being used in the SIUC ELM program. Distribution of the Instructor version will be very limited at this time. Contact Martin Hebel if you require help or see about an instructor's version.

Application of Microcontrollers

Contact Information

E-mail:

Primary developer of the manual and labs:
Martin Hebel ..... mhebel@siu.edu

Contributing developer and editor:
Will Devenport ..... willd@siu.edu

Director, Off-Campus Academic Programs:
Dr. Terry Bowman..... tbowman@siu.edu

Chair, Department of Information Management Systems:
Dr. Jan Schoen Henry ..... jshenry@siu.edu

Mailing:

Electronics Management
MC: 6614
College of Applied Sciences and Arts
Southern Illinois University, Carbondale
Carbondale, IL 62901-6614

A product such as this is not done alone. The following people are thanked for their contributions: Ken Gracey and the gang at Parallax for their work on making this possible for our students; Philippe Techer at Virtual Micro Design for designing a great simulation package and working with us; Myke Predko for his feedback and recommendation; I. Scott MacKenzie for a concise text on the 8051; our student evaluators John Almy and Kirk Larsen for catching many mistakes and for contextual recommendations; and finally Terry Bowman and Jan Henry for budgeting the endeavor and wanting the best education for our students.

Key Web Sites:
Electronics Management Home Page: ..... www.siu.edu/~imsasa/elm

Off-Campus Programs Home Page: http://131.230.64.6/

Parallax Incorporated Home Page: ..... www.parallaxinc.com
..... www.stampsinclass.com

Virtual Micro Design Home Page (UMPS): ..... www.vmdesign.com

Distributors & Additional Information:
Digi-Key Electronics - Stamps, components ..... www.digikey.com

Jameco Electronics - Stamps, components ..... www.jameco.com

JDR Electronics - Stamps, components ..... www.jdr.com

Wirz Electronics - UMPS U.S. Sales..... www.wirz.com

Peter H. Anderson - General microcontroller information ... www.phanderson.com

SelmaWare Solutions - Specialized interfacing software ..... www.selmaware.com

Texts:
The 8051 Microcontroller, 3rd ed. 1999, Scott MacKenzie. Prentice-Hall
ISBN: 0-13-780008-8

Handbook of Microcontrollers. 1999, Myke Predko. McGraw-Hill
ISBN: 0-07-913716-4

Programming and Customizing the 8051 Microcontroller. 1999, Myke Predko. McGraw-Hill.
ISBN: 0-07-134192-7

The Microcontroller Idea Book. 1994, Jan Axelson. Lakeview Research.
ISBN: 096508190-7

Table of Contents

Introduction: Programming the BS2

Using the Stampw.exe program under Windows is fairly simple. Programs are written and downloaded to the BS2 on the Activity Board via a serial cable connected to COM 1 or COM 2 serial port of the computer. This manual will assume a degree of familiarity using Windows application programs.

File Installation

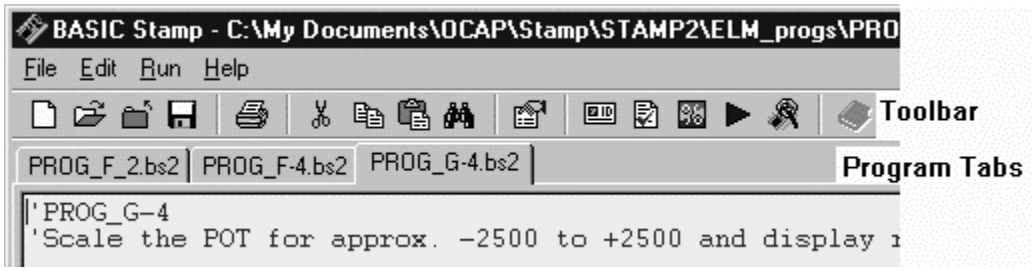
Unzip the amprog.zip file. It will contain the entire sample BS2 program files for the manual and labs.

Using Stampw.exe

1) Connect the Basic Stamp 2 (BS2) to the computer with the provided serial cable. If the only open COM port you have is a DB25 (25 pin) vice a DB9 (9 pin) you will need to use an adapter available at most computer stores or Radio Shack.

2) Provide power to the Activity Board with the provided power supply wall transformer. *Damage may occur to the Activity Board and Stamp if you use a power supply not provided with the kit*.

3) Locate and run the **stampw.exe** program using Windows Explorer. The following explains the toolbar buttons and basic operation of the program.



Standard File and Printing Tools: 

New Document | Open File | Close File | Save | Print

Standard Editing Tools: 

Cut | Copy | Paste | Find

Preferences: 
This tool can be used to set up preferences such as editor background color, font color, etc. It can also define the COM port on which the BS2 is connected under 'Editor Operation'. Normally stampw.exe will auto detect the BS2 on Com 1 or 2. If your computer is configured to another port for the BS2, it may be defined here.

Identify: 
This allows the programmer to verify the BS2 is found and recognized by stampw.exe.

4) Click the Identify button. A message will be returned indicating the BS2 was found, or that there was a problem communicating with the BS2. If a problem occurred, try to resolve prior to continuing.

Syntax Check: 
Prior to the entered program being downloaded to the BS2, a syntax check is performed of the program. The programmer may use this tool to force a syntax check. The BS2 does NOT need to be connected to syntax check programs.

Memory Map: 
This tool will open the memory map window. It displays memory (EEPROM and RAM) usage of the program in the editing area will utilize.

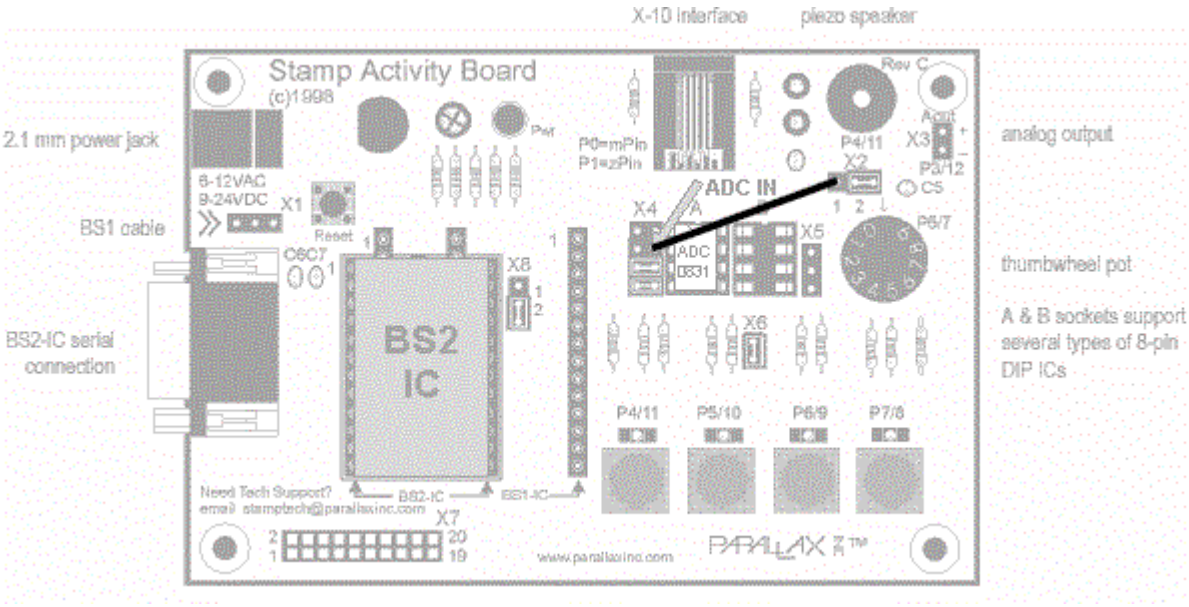
Run: 
This button will perform the following:
- Perform a syntax check on the program in the editor window.
- Tokenize the program.
- Transfer the program to the BS2 memory.
- Execute the program on the BS2.

Debug: 
The Debug window is primarily used for DEBUG commands in a BS2 program to send information back to the host computer. This window will open automatically anytime a program downloaded to the BS2 contains a Debug instruction. Clicking the tool will also open it. The COM port the BS2 is connected to will need to be set following a manual opening
NOTE: *Due to an application quirk, the debug window can be 'lost' by having it open and clicking on another open window, such as the editor. The window is not able to seen again until the stamp2.exe program closes and re-opens. It is HIGHLY RECOMMENDED this window is maximized the first time it is used to minimize the chance of this problem arising.*

Help: 
The help files will eventually bring up Help files for stampw.exe. It is not implemented on the current version(s). Newer releases may be download from Parallax at *www.parallaxinc.com*.

Program Tabs:
These tabs indicate all open BS2 programs. A maximum of 16 programs can be open for editing at any one time. A new tab is created any time a new program is started or a program is opened. All tools work on the active program.

5) Connect one of the provided jumpers between the Analog Input and Pin 1 (the left-most pin) of X2 as shown:



6) Open (File → Open, or use the toolbar button) the program AB-Test.bs2.

7) Run this program using the button on the toolbar. Stampw.exe will tokenize the program and show indication it is transferring the program to the BS2.

8) The debug window will appear. Follow the directions in the debug window. AB-Test will perform a test of several features of the board to ensure proper operation of inputs and outputs. If any of the tests do not work correctly, attempt to resolve the problem and test again. Contact your instructor or coordinator if it cannot be resolved.

9) Close the debug window.

10) Disconnect the BS2 Activity Board from the computer leaving power connected. Press the RESET button the Activity board. The CHARGE! tune should play. The program is still in BS2 memory though it cannot communicate to the host computer.

11) Disconnect the Stamp from power. Wait approximately one minute. Reconnect power. Press RESET again. The program should still be in memory.

12) Reconnect the BS2 to the computer.

13) Remove the ADC in jumper.

14) Open a NEW editing window by clicking the File button. Enter the following code:

```
X var byte
FOR X = 1 to 20
        FREQOUT 11, 100, 2500
        PAUSE 20 - X * 20
NEXT
FREQOUT 11, 500, 3000
END
```

15) Run the program using the toolbar button.

It's gonna blow!!!!!!! Kidding. But see how easy it is to program? Now on to your labs!

Lab A - Introduction to the BASIC Stamp

References:
A. Application of Microcontrollers Manual. 1999, Southern Illinois University.
B. BASIC Stamp Manual, Version 1.9. 1998. Parallax, Inc.

Objectives:
1) Discuss the primary differences between microcontrollers and microprocessors
2) Discuss the difference between processor machine languages and high level languages.
3) Discuss the differences between interpreted and compiled languages.
4) List the ROM, RAM and I/O resources available on the BASIC Stamp II.
5) Develop flowcharts for operations.

- Read Section A of Reference A.
- Refer to Reference B to clarify or expand on Reference A material as needed.

---

1. (1 pt) A _____ has designed into it RAM, ROM and I/O lines.

2. (2 pts) The PIC16C57 has _____ bytes of ROM and _____ bytes of RAM available in it.

3. (2 pts) The PIC16C57 has _____ Input-Output (I/O) pins or lines, of which ____ of these are available to the BS2 programmer.

4. (2 pts) Why are high-level languages independent of the processor being written for?

5. (1 pt) Symbols which are representative of high-level language code and decoded by an interpreter are known as _____.

6. (1 pt) The Language that is used to program the BS2 is called _____.

7. (1 pt) The _____ on the PIC16C57 is used to hold the PBSASIC2 interpreter on the BS2.

8. (1 pt) Where are programs downloaded stored on the BS2?

9. (2 pt) Four I/O pins or lines of the PIC16C57 are not usable to the BS2 programmer. What are these four lines used for?

10. (7 pt) Draw a flowchart for cooking spaghetti noodles. Some of the steps you will want to include are:

- Heating water
- Waiting for water to boil
- Adding noodles
- Testing firmness of noodles
- Draining noodles

Lab B - The Stamp Activity Board & Basic I/O

References:
A. Application of Microcontrollers Manual. 1999, Southern Illinois University.
B. BASIC Stamp Manual, Version 1.9. 1998. Parallax, Inc.

Objectives:
1) List the I/O devices available on the Activity Board.
2) Identify the BS2 pin numbers associated with each I/O.
3) Discuss terminology and respective voltages associated with digital I/O.
4) Write PBASIC2 code to read and write to simple I/O.
5) Use PBASIC2 commands to control the Activity Board speaker.
6) Discuss the need for debouncing input devices.
7) Write PBASIC2 code for debouncing buttons.

- Read Section B of Reference A.
- Run programs and sample code.
- Refer to Reference B to clarify or expand on Reference A material as needed.

---

1. (5 pts) List the BS2 I/O pin numbers for the listed Activity Board devices. The first one is done for you:

   The blue (right-most) pushbutton *P8*

   The red (left-most) pushbutton..... ___

   The speaker ..... ___

   The A/D converter Dataout line ..... ___

   The potentiometer..... ___

2. (1 pt) A digital I/O which is HIGH is also referred to as _____ as if it were an on-off switch, and _____ for its binary representation.

3. (1 pt) LEDs on the Activity Board require a _____ output level to light them. Pushbuttons which are not depressed are represented by a _____ input level.

4. (2 pts) Write code that would turn on (light) the LED associated with the Black button.

5. (3 pts) Write code that would produce a 2300 Hz frequency from the speaker for 1.5 seconds.

6. Program the following code into the BS2.

```
INPUT 8
DIR10 = 1
Loop:
   IF IN8 = 0 THEN ToggleMe
GOTO Loop

ToggleMe:
TOGGLE 10
GOTO Loop
```

7. (2 pts) With the program from question 6 running, depress the blue pushbutton 10 times. What effect does it have? Is it consistent? Why or why not?

8. (4 pts) Change the program in question 6 to use the Button command. Structure it so that it provides a delay of 250 and repeats at a rate of 150. For the bytevariable, name it BtnPress. Test your program.

9. (1 pt) Connect a voltmeter so that the negative lead is on GND of the I/O pin header and the positive lead is on P10 for the BS2 on the I/O pin header. Record the DC voltage when the LED is OFF: _____Vdc. And when the LED is ON: _____Vdc. (USE CAUTION TO NOT SHORT TOGETHER PINS)

10. (1 pt) Connect a voltmeter so that the negative lead is on GND of the I/O pin header and the positive lead is on P8 for the BS2 on the I/O pin header. Record the DC voltage when the Blue button is not depressed: _____Vdc. Is depressed: _____Vdc. (USE CAUTION TO NOT SHORT TOGETHER PINS)

Lab C - Binary Numbers

References:
A. Application of Microcontrollers Manual. 1999, Southern Illinois University.
B. BASIC Stamp Manual, Version 1.9. 1998. Parallax, Inc.

Objectives:
1) Discuss the relationship between the decimal and binary number systems.
2) Discuss the need to use binary numbers.
3) Convert binary numbers to decimal.
4) List the number of bits and range for nibbles, bytes and words.
5) Write PBASIC2 code to use bit groups for inputs and outputs.
6) Write PBASIC2 code to use the debug window to show data.

- Read Section C of Reference A.
- Run programs and sample code.
- Refer to Reference B to clarify or expand on Reference A material as needed.

---

1. (2 pt) Decimal uses a number base of 10 and is represented by 10 unique digits, 0 - 9. Binary uses a number base of ____ and is represented by ____ unique digits that are: _____.

2. (1 pt) The 3rd place in a binary number has the decimal equivalent weight of 4, the 6th place has the decimal equivalent weight of _____.

3. (1 pt) The binary number 100110_2 converted to decimal is _____.

4. (1 pt) PBASIC2 uses a _____ to represent a binary number instead of a subscripted 2.

5. (3 pt) List the range in decimal numbers for the following binary groups:

   Nibble: _____

   Byte: _____

   Word: _____

6. (1 pts) A binary number with 12 places can hold a maximum decimal count of _____. Show your math.

7. (2 pts) Write PBASIC2 code to display the number 85 in binary in the debug window.

8. (2 pts) Write code that will light all 4 Activity Board LEDs simultaneously as a single nibble.

9. (1 pt) In the following number, identify the LSB and the MSB: 10010010_2.

10. (6 pts) Write code that will read all 4 buttons as a single nibble and sound the speaker based on the nibble read.

- Span it so that 0000_2 = 0 Hz and 1111_2 = 4500 Hz.
  - Use ^ %1111 to invert the value read from the pushbuttons.
  - Multiply the nibble by 300 to span.
- Each tone will last 250mSec
- Display the Frequency in the Debug window
- The shared pin of 11 for the speaker and the red pushbutton may cause problems. If it does in your program, ignore it and use the other 3 pushbuttons to test.
- Loop continuously.

Lab D - Analog Inputs and Outputs

References:

A. Application of Microcontrollers Manual. 1999, Southern Illinois University.
B. BASIC Stamp Manual, Version 1.9. 1998. Parallax, Inc.

Objectives:

1) Discuss how digital values are representative of analog quantities.
2) Discuss how Pulse Width Modulation can be used to vary the output of a device.
3) Write PBASIC2 code to use the PWM command.
4) Discuss analog to digital conversion and quanta levels.
5) Program the BS2 to gather data from the Activity Board 0831ADC
6) Program the BS2 to read the Activity Board potentiometer.

- Read Section D of Reference A.
- Run programs and sample code.
- Refer to Reference B to clarify or expand on Reference A material as needed.

---

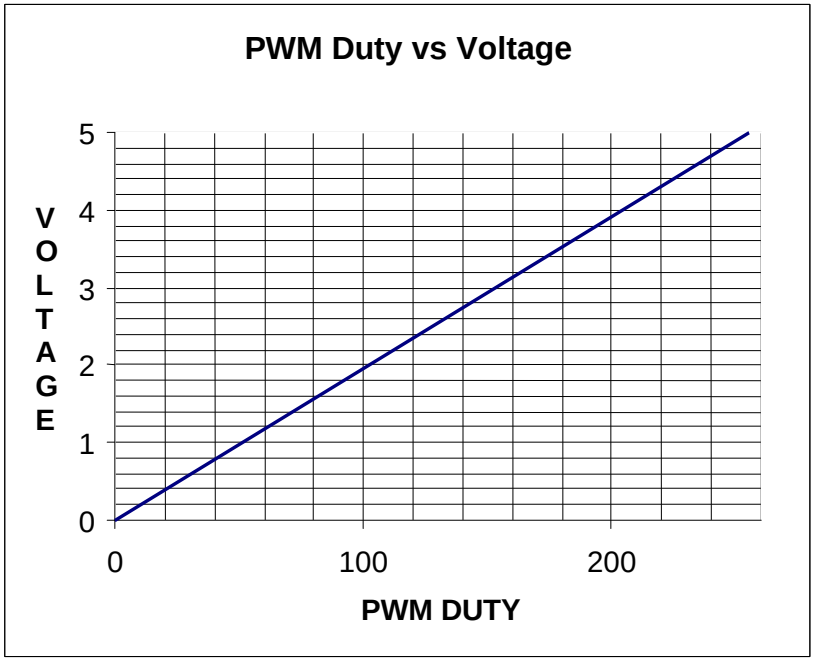
1. (1 pt) What limits the resolution of an analog quantity represented by a digital number?

2. (1 pt) The resolution of the pulse width modulation control for the BS2 is ____ bits or _____ quanta levels.

3. (2 pt) If a heater produces 1000 BTUs of heat at 100% duty cycle, write PBASIC2 code to produce 800 BTUs through pulse width modulation from pin 5 with 500 cycles per execution.

4. (2 pts) Open and run program D-2 from reference A.

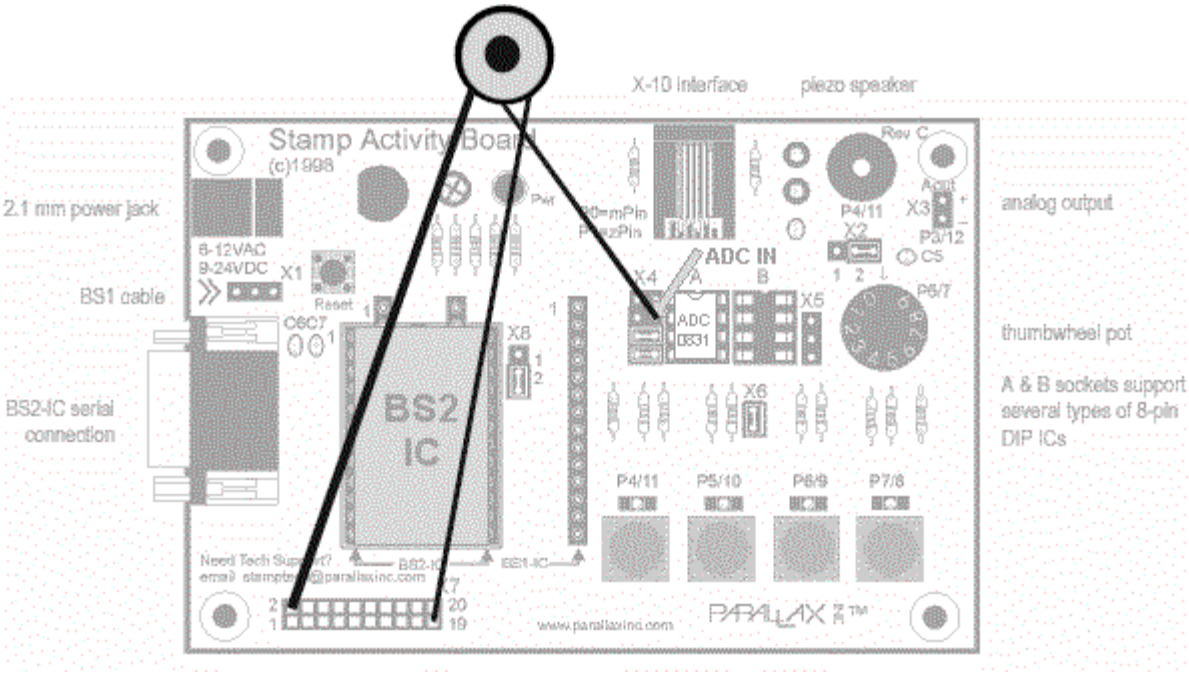
- Using a voltmeter to measure DC voltage, connect the positive (red) lead to the + terminal of Aout and the negative lead to the - terminal of Aout.
- Use the Blue and Red buttons to adjust the voltage.
- For the range of 1- 255 in increments of 20:
  - The PWM Duty vs Displayed voltage (expected) has been graphed.
  - Plot the PWM Duty vs Actual voltage in another color.



5. (1 pt) Using a midpoint value from the data collected in question 4, calculate the %Error (show work):

$$\% \text{ Error} = \frac{\text{Actual-Expected}}{\text{Expected}} \times 100$$

6. (1 pt) What conclusions can you draw from your results?



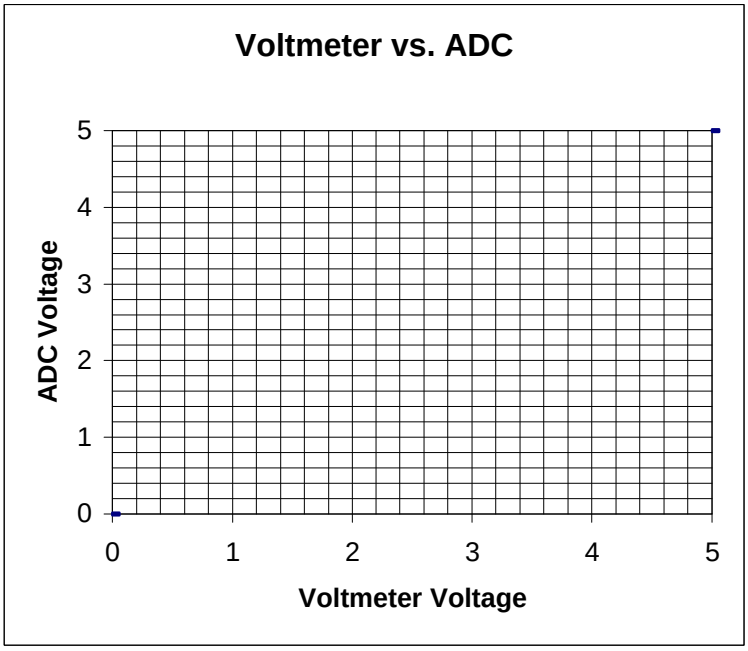
7. Connect the potentiometer provided to the Activity Board as shown:

- Red lead to +5V Vdd (Pin 19)
- White center lead to the analog input on the ADC0831 (ADC IN)
- Black lead to Ground (Pin 2)

8. (2 pts) Open and run Program D-4 from the Reference A. Using a voltmeter measure DC voltage between the center-tap (White wire) of the potentiometer and ground (black wire) for at least 10 potentiometer settings from fully counter clockwise to fully clockwise. Record the voltmeter reading and the mVolt reading from the program D-4 debug window ADC Reading.

Position	Voltmeter	DEBUG ADC Reading		Position	Voltmeter	DEBUG ADC Reading
Fully CCW				6		
2				7		
3				8		
4				9		
5				Fully CW		

9. (1 pt) Plot your results for the ADC voltage on the following page.



10. (1 pt) What conclusions can you draw from your results?

11. Replace the potentiometer in question 7 with the LM-34 temperature sensor using the same color coding for connections!. This sensor measures temperature (up to 300°F) and converts it to a voltage, where 1°F = 0.01 volts (1 tenth of a volt). At 100°F the voltage would be 1.0 volts (100 tenths of a volt).

12. (2 pts) Measure and record the voltage with a voltmeter between the green lead of the LM-34 (voltage output) and ground (black lead): _____ vdc.
Running program D-4 record the voltage displayed in the debug window:_____tenths of a volt. Based on the voltage reading from the voltmeter, what would you expect room temperature to be? _____.

13. (1 pt) Alter Program D-4 to display ADres in degrees Fahrenheit. Write the changed line below.

14. (1 pt) Briefly hold a flame to the LM-34 while the program is running. Monitor the voltage with a voltmeter and compare it to the debug window readings. Do they track?

15. (4 pts) The Activity Board potentiometer knob has the numbers 0 - 10 embossed on it. Write a PBASIC2 program to approximately display the expected number to be under the arrow (located above the knob) as it is adjusted. (The RCTIME function is non-linear, so it will not track perfectly).
    Hint: Use a scaler to convert the full range value of RCTIME to a range of 0-10.

Lab E - Process Control

References:

A. Application of Microcontrollers Manual. 1999, Southern Illinois University.
B. BASIC Stamp Manual, Version 1.9. 1998. Parallax, Inc.

Objectives:

1) Discuss methods of process-control.
2) Discuss advantages and disadvantages of the different process-control methods.
3) Write PBASIC2 code to perform simple process-control.

- Read Section E of Reference A.
- Run programs and sample code.
- Refer to Reference B to clarify or expand on Reference A material as needed.

---

1. (1 pt) A system is programmed so that a pump fills a 200 gallon tank when the level drops below 100 gallons. The pump shuts off when the level is at or above 100 gallons. The type of process control implemented is _____.

2. (2 pts) Given the following program, add code that will control the pump in question 1. The potentiometer acts as the level sensor, the blue button's LED will indicate the pump is running.

```
Gallons var byte
POT var word

Loop:
HIGH 7:Pause 10
RCTIME 7,1,POT

Gallons = POT * 2 /53 MIN 1   'Scale for ~ 0-200 gallons.
Debug "Tank volume = ", DEC Gallons, CR




GOTO Loop
```

3. (1 pt) What is a problem associated with the control of the system in the above manner (and for most systems that use this method of process control)?

4. (3 pts) Modify the program in question 2 so that the pump energizes at 50 gallons and de-energizes at 150 gallons.

```
Gallons var byte
POT var word

Loop:
HIGH 7:Pause 10
RCTIME 7,1,POT

Gallons = POT * 2 /53 MIN 1                'Scale for ~ 0-200 gallons.
Debug "Tank volume = ", DEC Gallons, CR




GOTO Loop
```

5. (1 pt) Question 4 uses a _____ process control method for level control.

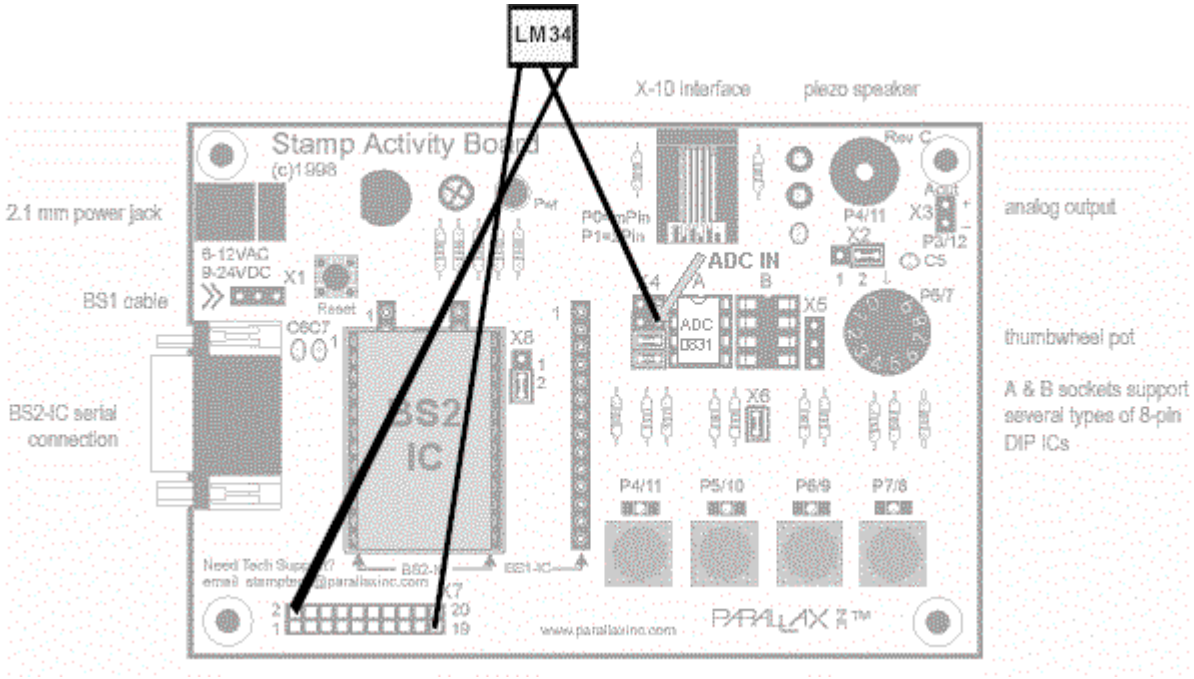
6. (2 pts) Assume the pump can be driven at variable speeds based on duty cycle. Draw a graph showing %Duty cycle vs. the volume of 50 gallons (highest flow rate) to 150 gallons (fully off).

7. (1 pt) The pump would be driven at _____ % Duty cycle if the tank volume was 60 gallons.

8. (2 pts) Write code that will drive P10 (Green button's LED) continuously on the BS2 at 90% duty cycle for 100 cycles.

9. (1 pts) Was the LED in question 8 brightly lit or dimly lit? Why?

10. Connect the LM-34 to the Activity Board.



- Red lead to +5V Vdd (Pin 19)
- White center lead to the analog input on the ADC0831 (ADC IN)
- Black lead to Ground (Pin 2)

11. (6 pts) Write a program that will measure the temperature and perform the following:

- Energize the green PB's LED when temperature is < 80 °F, de-energize at or above 80°F.
- Energize the blue PB's LED when temperature is > 90 °F, de-energize at or below 90°F.
- Sound an alarm from the speaker when temperature is > 100 °F and lock in until black PB is pressed.

Lab F - Hexadecimal & BS2 Memory

References:
A. Application of Microcontrollers Manual. 1999, Southern Illinois University.
B. BASIC Stamp Manual, Version 1.9. 1998. Parallax, Inc.

Objectives:
1) Explain the reason for working in the hexadecimal number system.
2) Convert between binary and hexadecimal number systems.
3) Explain the areas of the BS2 Memory Map and their contents.
4) Discuss the function and use of the BS2 registers.
5) Discuss the use of the BS2 EEPROM.
6) Program in PBASIC2 to read and write to EEPROM memory locations.

- Read Section F of Reference A.
- Run programs and sample code.
- Refer to Reference B to clarify or expand on Reference A material as needed.

---

1. (1 pt) What is the major advantage of using hexadecimal when working with digital systems versus binary?

2. (1pt) Hexadecimal is a base ____ number systems.

3. Enter the following program and open the Memory Map window for the BS2.

```
w var word
x var byte
y var nib
z var nib

w = 512
y = 11
FOR y = 0 TO 10
x = y + 65
WRITE y, x
NEXT
x = 205
z = 10
```

4. (2pts) What is the lowest EEPROM memory address containing program code (per the legend color-coded) in hexadecimal: _____. Converted to decimal: _____.

5. (2pt) The variable x is stored in register _____. Why?

6. (2 pts) Write code at the end of the program in question 3 that will store the contents of the x, y and z variable memory locations as a word in variable w. Display the variable w in hexadecimal in the debug window. (hint: Use the register names vice the variable names).

The debug window displayed a value of w = _____$_{16}$.

7. (4 pts) Add to the end of the program in Question 3 the code needed to read and display the contents of EEPROM memory locations 000_{16} - 010_{16}. Display the results to the debug window as ASCII (do not specify DEC, IHEX, IBIN or any other modifier). Write your code here.

The debug window of your code displayed: _____.

8. Open a new editing window and enter the following code:

```
ReadMe  DATA $45, $4C, $4D, $20, $2D, $2D, $20, $53, $49, $55, $43
```

9. (1 pt) Using the Memory Map window, the data from the program in ASCII is _____.

10. (4 pts) Write the code required to read the data in question 8 and display it in ASCII.

11. (2 pts) Write code that will store 'HELLO WORLD!' in EEPROM memory beginning at memory location 50_{16}. Use 'Greeting' as the data name.

12. (1 pt) The letter 'D' from question 11 is located in memory location ____$_{16}$.

Lab G - Logic Operators and Signed Numbers

References:

A. Application of Microcontrollers Manual. 1999, Southern Illinois University.
B. BASIC Stamp Manual, Version 1.9. 1998. Parallax, Inc.

Objectives:

1) Discuss how logic is used to make true/false decisions.
2) Write the truth tables for ANDs, ORs, NOTs and XORs.
3) Write PBASIC2 code utilizing logic operators.
4) Apply masks to perform bit operations.
5) Write PBASIC2 code to mask individual bits in binary numbers.
7) Convert number to 2's compliment signed numbers.
8) Subtract binary numbers using 2's compliment.

- Read Section G of Reference A.
- Run programs and sample code.
- Refer to Reference B to clarify or expand on Reference A material as needed.

---

1. (1 pt) A binary 1 would be representative of a _____ logic statement.

2. (3 pts) Complete the following logic statements:

   1 AND 1 = _____ NOT 0 = _____

   0 OR 1 = _____ 1 XOR 1 = _____

   1 AND 0 = _____ 1 XOR 0 = _____

3. (1 pt) Complete the following PBASIC2 code that will sound the speaker if the Green button is pressed or the Blue button is pressed, but not both at the same time.

```
INPUT 8
INPUT 10

Loop
PAUSE 1000
_____ THEN SOUND
GOTO LOOP

SOUND
FREQOUT 11, 250, 2000
GOTO LOOP
```

4. (3 pts) Complete the following masking operations:

1010	1010	1010
OR 1100	AND 1100	XOR 1100

5. (2 pts) Given any binary number, such as 11011101 write a mask and logic operation that would clear (set to 0) every bit except the first and the last.

6. (3 pts) Write PBASIC2 code that will read the Activity Board inputs P8 - P11 as a nibble and will invert the P8 and P9 bits only. Display it in binary in the debug window.

7. (1 pt) Add the following binary numbers: $1001_2 + 0111_2$. Show your work.

8. (2 pt) Convert the number - 0110_2 to 2's compliment.

9. (2 pts) Use 2's compliment to subtract the following numbers: $1100_2 - 0101_2$. Show your work.

10. (2 pts) Write PBASIC2 code to display -5 as a signed decimal number and as a 2's compliment binary number.

Lab H - Digital Communications

References:

A. Application of Microcontrollers Manual. 1999, Southern Illinois University.
B. BASIC Stamp Manual, Version 1.9. 1998. Parallax, Inc.

Objectives:

1) Discuss advantages of parallel and serial communications.
2) Discuss the need for address lines in digital systems.
3) Discuss the need for synchronizing serial data.
4) Discuss how asynchronous data is decoded.
5) Write PBASIC2 code to communicate between BS2s.

- Read Section H of Reference A.
- Run programs and sample code.
- Refer to Reference B to clarify or expand on Reference A material as needed.

---

1. (2 pts) The transfer of data of all bits simultaneously is known as _____ communications. The transfer of data one bit at a time is _____ communications.

2. (1 pt) _____ lines are used to the same parallel bus to send data to several devices, or to even control the latching of a single receiver

3. (2 pts) Why does serial data communication need some form of synchronization?

4. (1 pt) _____ communications, such as RS-232, depends upon transmitter and receiver timing to separate individual bits from a data stream.

5. (1 pt) If the transmitter is sending asynchronous data at 19200 baud, the length of time to transmit the entire word (including start & stop bits) is _____ μS?

6. (2 pts) Complete the PBASIC2 code to transmit the variable 'Bytedata' non-inverted in an 8-bit transmission at 19200 baud on P3 using the SEROUT command.

```
Bytedata var byte
Bytedata = 70
```

7. (2 pts) Complete the PBASIC2 code to accept a serial byte on P6 using the SERIN command. The data will be a non-inverted 8-bit transmission at 19200 baud and saved to the variable ByteIn. Use a timeout of 1 second.

```
Loop:
ByteIn var byte



Timeout:
Debug "Timed out",CR
Goto Loop
```

8. (9 pts) Write code for a serial transmitter/receiver. The transmitter will send the setting of the Activity Board potentiometer scaled to fit in a byte at 9600 baud on P2 every 1 second. The receiver will read the byte on P3, re-scale it back to the approximate original value, and sound the speaker at the specified frequency.