## The BASIC Stamp

This lab is our first using the BASIC Stamp (BS), a programmable chip made by parallax. The Stamp has 16 pins, each of which can be designated as inputs or outputs, as desired. Code is placed into the memory of the Stamp using the BASIC Stamp editor program, which you should have installed on your laptop by now.

While the Stamp is being programmed, it will sit on something called the "Board of Education" (BOE), which connects the Stamp to the computer, via either USB or serial connections. Although the Stamp can be removed from the BOE after it is programmed, we will not be removing it for today's lab. The Stamp is fairly snug in the BOE, and repeated removals might damage it.

To get started, attach the BOE to your computer, and run the BS Editor. The editor allows you to program the stamp using a language called PBASIC, which is syntactically very similar to the classic BASIC programming language. If you press CTRL-I, or click the "Identify" button at the top of the editor, the editor will tell you whether or not it has detected the BOE. However, the Stamp itself is probably not active. To power the Stamp itself, move the power switch on the BOE to the middle ("1") position. While the switch is in this position, the battery will slowly drain, so try to remember to power the Stamp down while you are programming or otherwise not directly using the Stamp. The power switch also has a right ("2") position, which allows some of the other peripherals on the BOE to receive power (such as the servo-motor connectors).

We will continue to use the PAD-234 boxes for peripherals (switches and LEDs). In order for the two systems to work together correctly, you will need to connect the ground of the Stamp/BOE (also called Vss) to the ground of the PAD-234. You should not connect the Vcc wires, just the grounds.

## Programming the Stamp

Every program on the Stamp must begin with two *directives*. To save the repetition of typing them for every program, the editor has a button that can directly add these lines to your code:

```
' {$STAMP BS2}
' {$PBASIC 2.5}
```

The first line is added by clicking the green picture of the Stamp labeled "Stamp Mode BS2".
The second is added by clicking the white "PBASIC Language 2.5" button.

At the end of your program, you should include an END statement. While the program will run fine without it, this command puts the Stamp into low power mode, which will save your battery. You will be given one battery today. It is yours. If your battery runs low, or you forget to bring it, you will not be given a replacement.

Unlike the "program" that runs continuously inside a 7400 NAND chip, the code on the Stamp executes only as many times as you tell it to. To make a program run continuously, you should include the appropriate portions of your code in a DO/LOOP structure. This will be especially important when we want to take the Stamp away from the BOE.

You should probably save the programs you write so you can refer to them later.

**Instructions**

Part A

Write and test a program on the Stamp to accomplish the following:

1. Use the debug command to print your name to the screen using ASCII codes. Be sure to use upper and lower case characters, as appropriate.
2. Create several simultaneous gates on the Stamp: P7 = P1·P2, P8 = P3+P4, and P9 = $\overline{P5 \cdot P6}$. Use switches 1 through 6 switches from the PAD-234 to set the inputs. You might use code similar to this:
   **IF (IN1 = 0) AND (IN2=0) THEN HIGH 3.**
   Other logical comparisons are also available.
3. Use switch 0 with P0 as an active-low "stop" switch, with the LOOP UNTIL command.

Part B

Write and test a program on the Stamp to accomplish the following:

1. Monitor 5 switches from the PAD-234, and create a bar-chart of 5 outputs indicating how many switches are high (the same as lab 3, but with five switches). To count inputs, I used an array of bits, which I added in a FOR loop.
2. Use on of the momentary switches as a stop switch.
3. Use 4 outputs on the Stamp to light up a 7-segment display indicating the number of active switches, from zero to five. Use the OUTL or OUTH command to send all 4 signals DCBA to the 7-segment display at one time. Note that OUTL 7 is the same as OUTL %00000111 (the % symbol means "binary").

Part C

Write and test a program on the Stamp to accomplish the following:

1. Mimic a JK flip flop with an active low trigger. Be careful! You shouldn't actually change Q or $\overline{Q}$ until the end of each loop!
2. Connect your five inputs (CLR, PRE, J, K, and CLK) to a 74112, and compare the resulting Q of your program with the 74112.