

PH317 Lab 8 – Introduction to the STAMP microcontroller

(one week)

Objectives:

1. Become familiar with the Stampw windows interface and basic programming
2. Compose alternative programs using the Stamp to measure resistance or capacitance.
3. Design and implement a circuit similar to one from previous labs using the Stamp.

Parts Needed:

Stamp II - BS2-IC

RS-232 serial cable (9pin D-sub female conn. To protoboard pins)

Computer with National Instruments data acquisition card and SC-2075 extension board

22 gauge insulated jumper wires

References:

Previous lab reports

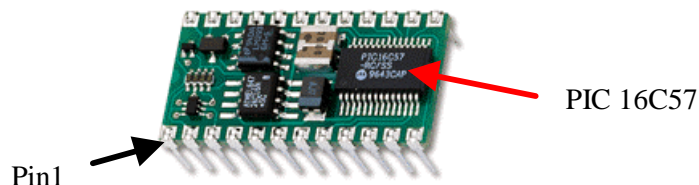
CD from Parallax (files are installed on the computer)

1. ..\Documentation\Basic_Stamps\BASIC_Stamp_Manual_v1_9.pdf
2. ..\Documentation\Basic_Stamps\BASIC_Stamp_Windows_Interface_Manual_v1_2.pdf
3. ..\Documentation\Educational_Curriculum\What's_A_Microcontroller_v1_5.pdf
4. Example programs in ..\Source_Code\BASIC_Stamp_Activity_Board

Procedure:

1. Connecting the Stamp controller to the SC-2075 extension board and computer.

It is important to connect the Stamp correctly to prevent damage to this CMOS chip. Before applying power to the circuit have the lab personnel check the wiring. The small DIP board (called BS2-IC) holding the PIC16C57 is compatible with the 0.1" spaced pins on the SC-2075 extension board. Pin 1 is located as shown.



DO NOT connect the 9 pin D-sub connector to the computer at this time. Connect pins to the following Stamp controller pins:

pin1 (white)	TX Serial output	Connected to pin 2 of DB9 connector
pin2 (red)	RX Serial input	Connected to pin 3 of DB9 connector
pin3 (green)	ATN Active-hi reset	Connected to pin 4 of DB9 connector (DTR) with two 0.1 F capacitors, one in series with the input and another from the input to ground.
pin4 (shield)	GND Data Ground	Connected to pin 5 of DB9 connector (GND)

In addition the Stamp COM port is detected by the software if RTS and DSR (pins 6 and 7 of the COM port) are shorted together. This is internally connected in the 9 pin D-sub connector.

Finally we need to add +5 volts and GND to the circuit. Set the +5V select switch on the SC-2075 to off. **DO NOT TURN ON THE POWER until the connections have been checked.**

pin21 +5V supply 5-volt input (or regulated output).
pin22 RES Active-low reset Pull low to reset; no connection
pin23 GND System ground
pin24 PWR ** Regulator input; no connection

At this time check your circuit yourself (the IC is \$40 and we have limited quantities), then have lab personnel check it before turning on the +5 volts. We are ready to start! Once this has been connected any power supplies can be switched on or off without any risk of damage.

2. Output to the Pins - LED test circuit.

Construct an 8 line data indicator like the one you built in Lab 1 on the proto board area of the SC-2075 using a 74240, 8 LEDs and 150 Ω current limiting resistors. Connect the 8 data I/O lines to the 8 inputs of the data indicator. Write a short program to blink one of the LEDs and test to see if your program can successfully control (blink) an LED. There is a simple description in the first experiment of "What is a Microcontroller?".

```
output 8
reblink:
  out8 = 0
  pause 1000
  out8 = 1
  pause 1000
  debug ?out8      'this prints out the value to PC
  goto reblink
```

The first statement is the equivalent of DIR8 = 1, that is the direction register bit is set to 1, which is an output. Now use the reserved register names DIRS to set all the pins to outputs and drive all the LEDs using the reserve words OUTS or OUTH or OUTL. Note that you would not have been able to drive all 8 LEDs without the open collector logic (74240 IC) because the Stamp could not provide enough current – it is limited to a total of 50 mA per 8 pins.

3. Digital Outputs.

Construct a circuit (draw this for the write up) and write code to test the following commands:

PWM pin, duty, cycles

Convert a digital value to analog output via pulse-width modulation.

- **Pin** is a variable/constant (0-15) that specifies the I/O pin to use. This pin will be placed into output mode during pulse generation then switched to input mode when the instruction finishes.
- **Duty** is a variable/constant (0-255) that specifies the analog output level (0 to 5V).
- **Cycles** is a variable/constant (0-65535) specifying an approximate number of milliseconds of PWM output.

Measure the way that this is generated. What are the lengths of time intervals for a typical waveform? Is it periodic?

PULSOUT pin, time

Output a pulse of 2 μ s to 131 ms in duration.

- **Pin** is a variable/constant (0-15) that specifies the I/O pin to use. This pin will be placed into output mode immediately before the pulse and left in that state after the instruction finishes.
- **Time** is a variable/constant (0-65535) that specifies the duration of the pulse in 2 μ s units.

Note that the pulsout command can produce different pulses depending in the initial state of that pin. Test this.

FREQOUT pin, duration, freq1{,freq2}

Generate one or two sine-wave tones for a specified duration.

- **Pin** is a variable/constant (0–15) that specifies the I/O pin to use. This pin will be put into output mode during generation of tones and left in that state after the instruction finishes.
- **Duration** is a variable/constant specifying the length in milliseconds (1 to 65535) of the tone(s).
- **Freq1** is a variable/constant specifying frequency in hertz (Hz, 0 to 32767) of the first tone.
- **Freq2** is a variable/constant specifying frequency (0 to 32767 Hz) of the optional second tone

Look on an oscilloscope to see what the output looks like. How are tones generated?

4. Capacitance meter.

Use the following function to make a capacitance meter. The description of this is in the Stamp manual. Remember that this is based on the threshold voltage for the HCT type logic.

RCTIME pin, state, resultVariable

Count time while pin remains in state—usually to measure the charge/discharge time of resistor/capacitor (RC) circuit.

- **Pin** is a variable/constant (0–15) that specifies the I/O pin to use. This pin will be placed into input mode and left in that state when the instruction finishes.
- **State** is a variable or constant (1 or 0) that will end the Rctime period.
- **ResultVariable** is a variable in which the time measurement (0 to 65535 in $2\mu\text{s}$ units) will be stored.

