

# LAB 4: TALKING TO YOUR COMPUTER

You are going to make 2 complete working prototypes of computer interaction systems that take input from sensors, translate it to some communications protocol and tell something more capable what to do (this will be either a computer or a MIDI synth). Your groups are going to demonstrate what you made. This part is informal, chances are you folks made something pretty neat and we want to see it and you some questions about what you did and why.

## PART 1 ETCH-A-SKETCH

Remember Etch-A-Sketch? It had 2 knobs one for each axis and it was really hard to draw stuff? You erased it by shaking it? We are going to make a cooler one! This has three components: the serial and sensor hardware, the basic stamp software and then computer software. You will hook up some sensors to the basic stamp then read them and send the data to a computer, which will then draw for you. The minimum is just to connect the x and y pots and get the computer drawing. You should add some additional functionality; maybe you can change the pen color. Maybe a button clears the screen. Maybe you shake something to clear the screen like the original! Be creative try several things, decide which things work best and which don't and tell us why.

### PART 1A HARDWARE

- 1 Connect the serial cable from the basic stamp to the computer
- 2 Connect 2 potentiometers to the A2D
- 3 Maybe connect some other sensors, for example an FSR, a switch, and accelerometer anything you like, if you have something exotic in mind talk to us (email) and we will figure it out.

### PART 1B EMBEDDED SOFTWARE

Write the basic stamp software that reads input from your sensors. (Have a look at the basic stamp manual or any other resource you find useful for this.) You have already seen some code that reads from one channel of A2D and prints it to the screen using DEBUG, now we want to send both channels over the serial port. Start simple with getting the samples from both channels and turning them into 8 bits each and alternating sending them over the serial port. Pick a baud rate, any baud rate you will be able to receive at any rate on the other side too. You can start with serial parameters I used on the computer end. Once you get that working you can make some improvements to your protocol and how you acquire, possibly condition and send the data.

### PART 1C COMPUTER SOFTWARE

On the computer side you will write software that translates the x y values into pretty lines on the screen. You can start with the program I have provided and then modify it to accommodate your additional sensors and possibly it draw prettier. You cannot use the mouse or keyboard to interact with your system. **YOU MAY ONLY INTERACT THROUGH THE INTERFACE YOU HAVE BUILT.**

Look at <http://processing.org>. It is filled with example code and reference material to do all sorts of things. Download it on your computer and play around. I will put some USB to serial adapters down there and I recommend you get the project working on your own computers if you have a laptop.

Here's my code:

```
import processing.serial.*;

Serial s;

void setup()
{
  size( 400, 400 );
  background( 255 );
  smooth();
  ellipseMode( CENTER );
  noStroke();
  fill( 0 );
  framerate( 50 );

  s = new Serial( this, "COM6", 2400, 'N', 8, 1.0 );
  framerate( 24 );
}

void draw()
{
  while ( s.available() > 1 )
  {
    int x = width*s.read()/255;
    int y = height*s.read()/255;
    println( x + " " + y );
    ellipse( x, y, 10, 10 );
  }
}
```

## THINGS YOU SHOULD DO TO MAKE THIS BETTER

### 1. Fix the protocol

a. Synchronize the protocol (i.e. make sure the x knob is always x and y knob is always y). I wrote it so it just takes the samples one after the other without regard to which is which. This is bad.

b. Use the full precision of the A2D instead of the much simpler 8 bits. How much of a difference does different precision make?

c. Try different baud rates. Does it make a difference?

d. How often is the basic stamp sampling the knobs and your sensors? Do we have to worry about aliasing?

### 2. Try mapping the knobs to other things, things you might try:

a. x y velocity

b. x y acceleration if you are really adventurous

c. polar coordinates

d. Drawing a curve is very difficult; can you come up with a better way of drawing a curve with your knobs?

3. If you look at the drawing it is a little noisy (the lines are never straight). Try some hardware and software filters to fix it and decide what you like best and why.

a. There should be some capacitors down there, try a few values to make a simple low pass filter on the pots. (Put them on the stamp breadboard. You can figure out where by looking at the A2D data sheets and/or the stamp manual or you can just be clever and look at how it's wired)

b. Construct a simple digital filter on the stamp and try that.

c. Try filtering the input on the computer side. Fix my code so instead drawing a bunch of dots it always draws continuous lines.

4. Add awesome features, some things you might try:

a. Clearing the screen. Maybe try a button or erasing by shaking just like the original! (be careful about how you decide if it's shaking or not)

b. Pen color

c. 3D?

This is a good start. Try other things too. This is excellent practice for how to make a fully functional final project.

## PART 2 Make a NEW INSTRUMENT FOR MUSICAL EXPRESSION

Now we are going to make a musical instrument. You will hook up some sensors and make MIDI and subsequently music (or possible just sound) from them. The field is wide open here. The connection between what the user does and then what happens is called a MAPPING. You should experiment with a few mappings and report on which ones worked better and why. Use some of the techniques you used in making your etch-a-sketch. Some possibilities are pots for pitch and velocity, or maybe buttons, or maybe FSRs or photoresistors. Try a bunch and try different combinations. See if you can come up was something that is more than the sum of its parts.

### PART 2A HARDWARE

1 Make sure the MIDI cable is wired up properly. Connect the cable to MIDI synth.

2 Hook up some sensors.

### PART 2B EMBEDDED SOFTWARE

Write a basic stamp program to read data from you sensors and translate it into appropriate MIDI messages. You might use the analog- to-digital converter or just digital IO on the pins. If you found something you want to use but don't know how let me know and we will figure it out. Feel free to tear apart surplus items that belong to you and harvest any parts you think look cool and are suited to your expertise and/or ambition.