**Introduction.**

For my 6.115 final project, I will build an augmented reality cursor for my computer. I will build a combined hardware and software system, using both the 8051 and PSoC, that will use photo resistors to track the position of a visual target (i.e. a red square of paper). The system will interpret the location of the target, and determine commands concerning how the cursor should move. I will send serialized commands (of my own design) over the serial port of the R31JP. I will write software for my Mac to interpret and implement these commands. In the end, a user will be able to move the target around physically, and witness the cursor on the computer moving around in a corresponding fashion. I also hope to implement a keyboard interface using the Amulet module, but I consider this an auxiliary part of the project.

I find this idea interesting because the core concept is so massively extensible. With the same hardware, new software could be written to implement a very complicated set of commands. I’m imagining a wide array of different hand motions that the system could detect and interpret. Eventually, this could be more powerful than existing user interfaces. For example, a DJ could have a specialized system from which he can control his music with hand motions; he would be a truly modern conductor!

**Hardware Description.**

There will be a bank of color-sensitive photo resistors. The outputs from these photo resistors will be scaled appropriately using some simple op-amp circuitry, so that the analog values range between 0V and 5V. Next, I will send these analog values into the PSoC as inputs. In the PSoC, I will use an ADC (or possibly multiple ADCs for parallel processing) to convert these analog inputs to 8-bit digital outputs. Furthermore, the PSoC will include some sort of multiplexer, allowing me to select which input is being converted. The output from the PSoC will go to an 8055 (or perhaps multiple 8055s for parallel processing), from which the 8051 will be able to read. See figure 1.

If possible, I would like to implement a keyboard. I will write an html file for the Amulet module that displays a keyboard. I will read input from the Amulet module using the 16C450 chip, just as we did in lab 3. See figure 2.

I’m hoping that I will be able to implement a “clicking” action in software. If I am not able to do so, I will implement a simple hardware system. I will probably just use the keypad as we did in labs 1 and 4. See figure 3.

**Software Description:**

The 8051 microcontroller will scan over the digital values from the 8055. These values represent the readings from photo resistors. After reading in a complete set of values, the program will process the values to determine the (x,y)-coordinates of the location of the target. Depending on the difference between this location and the most recently measured one, the software will determine if the cursor needs to be moved. If it does, a command will be encoded in binary and sent to the computer over the serial port. I’m also hoping the software will be able to interpret a “click”. If so, commands will be sent over the serial port in the same fashion. If I cannot implement an augmented reality click, I will have the 8051 take input from the keypad using an external interrupt, and send click commands that way. See figure 4 for the 8051 software diagram.

If I implement the keyboard, the 8051 will read from the 16C450 using the external interrupt. It will interpret the input bytes and determine the character being typed. It will then send a command over the serial port to reflect that.

I will also need to implement software on my computer to read the input from the serial port. It will read this input as a stream, interpret commands, and use system calls to implement these commands on my computer screen. See figure 5 for the computer-side software diagram.

**Project scope and management:**

My goal, in order to achieve at least a B, is to be able to track coordinate locations of a colored target. I am making this my goal because it is the most fundamental component of the project. Interfacing with my own computer might be difficult, but I’m honestly not sure since I’ve never tried using the serial port or system calls to move the cursor around and click. Furthermore, I’m anticipating that it might be difficult to achieve readings with high enough fidelity to control a cursor smoothly. My goal, in order to achieve an A, is to be able to functionally use my computer via my system. This includes a fully working cursor and mouse, and hopefully a keyboard as well. Note the optional keyboard in figures 2 and 4. The keyboard is not strictly necessary, since I can just use a click-based keyboard application on my computer. Lastly, my “ideal-world” goal is to build a set of physical motions that can be interpreted reliably by my system.

**Special component needs:**

I will need a bank of color-sensitive photo resistors. I will also (hopefully) need the Amulet module. Other than that, I think all my hardware needs are in the lab kit, namely op-amps, the PSoC, 8055, 16C450, the keypad chip, and other basic parts.

**Timetable:**

During the week starting April 15, I will complete sufficient hardware to get a high-fidelity reading from a color-sensitive photo resistor.

During the week starting April 22, I will create sufficient hardware and software to be able to scan over a full set of photo resistor values and store these values. I will also try to use the 16C450 chip with the Amulet module to suit my needs.

During the week starting April 29, I will interpret the photo resistor data as locations, and build a command set to send over the serial port. I will also try to get the Amulet module functioning as a keyboard in its entirety.

During the week starting May 6, I will write a Mac OS X application to perform to take serial input and perform the necessary functions.

During the week starting May 13, I will try to build a software interpreter for “clicking”, and if that fails I will implement a simple hardware interface, thus completing the project.