

## **ECE153B Final Project Proposal:** <https://sites.google.com/view/drawingarm/home>

by Tiffany Cowan, Lucas Relic

### **Overview**

We are going to build a drawing arm that is controlled by a joystick and several motors. To move around the paper, the arm will consist of two stepper motors, one for motion in the x-direction and one for the y-direction. The joystick will also have a select button, which will control whether or not the arm is drawing while it is in motion, which will result in the lifting and lowering of the pencil. **As a reach goal**, we will also implement an SD card that saves previous drawing directions so that the arm can automatically redraw previous drawings. Multiple drawings can be saved at once, and a selection can be made using the onboard joystick and LCD screen.

### **Peripherals**

- Joystick with Select Button: Controls x-y position of the arm and whether or not arm is drawing
- Stepper Motors: X-Y movement of the arm/up and down movement of the pencil
- Microstepping Controller: Controls motors
- SD Card\*: Saves previous drawings (\*reach goal)

### **Serial Interface Protocols**

- SPI: powerSTEP01 microstepping motor controller, store drawing data in SD card
- USB: joystick connection, controlling arm motion

**Block Diagram** (on website under “Project Proposal” tab)

### **Responsibility List**

- Tiffany: gather materials, update website weekly, construct arm/motor mechanism
- Lucas: program joystick-motor response, program button press-motor response

### **Software Structure**

Our project will be coded in C using the Keil  $\mu$ Vision IDE. We will communicate with the joystick through USB, which will provide the positional data to guide the drawing arm. There is a possibility we will need to design our own joystick driver, however we will use preexisting drivers if available. The microcontroller will use polling to read the positional data from the joystick, while the button press to control writing will be handled with interrupts. To configure the stepper motor controller, we will use SPI to set configuration registers. The controller will then interface with the stepper motors to move the arm in the desired direction. If we meet our reach goal, SPI will be used to interface with the SD card and store drawing data. Interrupts will handle the joystick presses that select the drawing and display its information to the LCD screen.