

# CS 23N, Winter 2004

## Homework #1: Controlling a servo

### Handout #3

---

**Due in class on Wednesday, April 14.** Materials for this problem can be picked up from the lab (B38). However, you **must** form a team and email [cs23n-instructors@cs.stanford.edu](mailto:cs23n-instructors@cs.stanford.edu) with your list of team members (4-5 students) before getting the materials and starting work.

Using the supplied bread-board, microcontroller, and other materials, construct a circuit to control a single servo, using a Pololu servo controller (PSC). You can download a manual for the PSC at [http://www.pololu.com/products/pololu/0290/ssc02a\\_guide.pdf](http://www.pololu.com/products/pololu/0290/ssc02a_guide.pdf)

Your controller should move a servo to (approximately)  $0^\circ$ , pause,  $90^\circ$ , pause,  $0^\circ$ , pause,  $-90^\circ$ , pause, and repeat the cycle from the beginning.

Your circuit should:

- Use “nice” connectors made with crimped pins and connector housings for connecting the power supplies and serial line to the PSC.
- Have a single power switch which simultaneously turns on/off both the microcontroller and the PSC’s electronics.
- A separate power supply for the servos on the PSC. (No power switch necessary.)

After completing the problem above, you may also wish to experiment with the following optional extensions.

1. (Easy) Add three LEDs to your circuit, and have one of the LEDs lit at each instant to indicate whether the servo is currently commanded to go to  $-90^\circ$ ,  $0^\circ$  or  $90^\circ$ .
2. (Easy) Add a pushbutton to your circuit so that pushing down and releasing the button causes your microcontroller to be reset. The easiest way to do this is by having the pushbutton connect pin 4 of the microcontroller to ground when it is pressed down. However, be careful not to create a short circuit!
3. (Intermediate) Following the instructions on the printout in the lab, modify a servo for continuous rotation. Instead of going to the three pre-set positions, your servo should now rotate continuously in one direction; stop and pause; rotate continuously in the other direction; stop and pause; and repeat.
4. (Challenging) Figure out how to make your microcontroller accept inputs. Add a potential-meter to your circuit, so that by turning the screw on the potentialmeter, we can control the length of the pauses between the different positions ( $-90^\circ$ ,  $0^\circ$ ,  $90^\circ$ ) the servo moves to.
5. (Challenging) You probably used a delay loop, such as a for-loop, to implement the pauses. This is computationally inefficient if the processor can be fruitfully used for other computations in the background. Figure out how to use interrupts on the microcontroller, and rewrite your code using timer-driven interrupts.

In addition to the suggested extensions above, you should feel encouraged to explore other changes/improvements to the circuit, and tell us what you did!