My senior project will be based around designing, building, and documenting a model of a Turing Machine. The Turing Machine is the brainchild of Alan Turing, and serves as a hypothetical device capable of calculating anything that a modern computer could calculate. Thus, a demonstrative model of such a device could serve as an easily accessible mathematical model for the sorts of computations that our society has become used to performing on a day-to-day basis. My advisor for this project will be Wenjun Hu.

The first part of this project would be to design, build, and test an appropriate electronic circuit with which a microcontroller could interact to display the state of the Turing Machine. As I am a Computer Science major, and not one that has formally studied electrical engineering, this would present a considerable academic challenge. The hardware involved in this Turing Machine would likely revolve around a strip of two-color LEDs, which would make up the tape. One color would indicate a “0” value, the other color indicates a “1” value, and an off LED indicates an empty cell. Part of the challenge of this type of implementation would be to have a large enough strip of LEDs to faithfully represent a reasonably sized Turing Machine, though this would likely involve more LEDs than there are outputs on the microcontroller board I plan to use. The design of this circuit would comprise the first deliverable of the project.

The next step would be to write a driver for the microcontroller to take as an input a finite state machine and a starting state of LED states and display the calculations that a Turing Machine would
make. As I have also not formally studied programming for any model of microcontroller, this step will prove an interesting academic hurdle, and I would hope to rely on my knowledge of other programming languages for this step. I anticipate using an Intel Galileo board to run the calculations for this machine. The finite state machine would likely be input as a text file on an SD card, and so would be easy to modify to run different programs. Also on the SD card would be the initial state of the machine. At first, I would also design the machine to accept a second text file as the initial state. The driver would interact with the circuit described previously to show not only the current state of the machine, but also display where the read/write head of the machine is. This driver would be the second deliverable for the project. The aim would be so that this machine could handle calculations at least as large as four-digit addition, and the circuit and driver would be written accordingly.

The third deliverable for this project would be a full writeup documenting my progress, setbacks, design decisions, and final build. I would keep a journal throughout this process to facilitate an accurate final writeup.

There are a number of different stretch goals that could be implemented to improve this project from this point. The first of these would be to add a small display that could show the names of files on the SD card. This would enable the machine to accept SD cards with more than one finite state machine file, and allow a user to switch easily between them. Adding this display would also entail including a number of additional inputs to the microcontroller to allow the user to choose between files and start the machine. This display could also output the current state of the machine in real time as a calculation is being completed.

The second stretch goal would be to allow a user to input the initial state of the machine, so it could be easily modified without connecting the SD card to a computer each time the user wanted to
run a program on a different input. This would involve adding a couple of inputs to the microcontroller, and would interact with the small display added in stretch goal #1.

A final stretch goal would be to configure the device to communicate wirelessly with another computer to accept new initial inputs, and possibly new finite state machines, without physically attaching or detaching an SD card. The Intel Galileo board can be expanded with a WiFi shield, making this step feasible.

The list of deliverables would be as follows:

1. Produce a circuit design that could interact with the Intel Galileo board to display the state of the machine as a program is run. Done by March 7
2. Write the driver that would interpret a finite state machine encoded on a removable SD card and interact with the circuit described in deliverable #1. Done by April 17
3. Produce a writeup of observations, design decisions, setbacks, and final build. This writeup would be accompanied by the full code produced in deliverable #2. Done by April 29

Potential stretch goals:

1. Add a display to allow the user to easily switch between programs
2. Create an interface to allow the user to change the initial state of the machine easily
3. Incorporate a wireless card to allow the device to more easily communicate with a PC to accept new finite state machines and/or initial states.