Modeling Lab

Embedded and Real Time Software, CSCI1600, Fall 2017, Prof. Reiss

Background
The purpose of this lab is to have you demonstrate what you have learned about the various ways that systems can be modeled. You should think about the problem and the various different types of models and choose the appropriate ones for the problem at hand.

The basic idea of the lab is to provide a model (as complete as possible given the time frame) of what the software for the game of Simon should look like. You can work alone or in teams. You should produce a model which can be turned in at the end of the lab.

Pre-Lab Work
Simon is a simple computerized game that was popular in the 80’s. It had four different colored lights that also served as (large) push buttons. It had a speaker to produce tones. There were also a couple of other switches to control the game. The original game is shown below:

The original game instructions can be found at [http://www.hasbro.com/common/instruct/Simon.PDF](http://www.hasbro.com/common/instruct/Simon.PDF).

A demonstration of Simon can be seen at [https://www.youtube.com/watch?v=4YhVyt4q5HI](https://www.youtube.com/watch?v=4YhVyt4q5HI).

The circuit for the game is relatively simple as can be seen:
Here you can think of U1 as an Arduino where the ports K1, K2, K4 and K8 are digital input lines (pulled high) and the ports R0-R7 are the output lines. The Chip U2 is just a driver used to power the non-LED lights and a 1980’s speaker (i.e. the inputs and outputs there are directly related). K1-K8 are used to sense the switches which are chosen by R0, R6 and R7; R1-R4 drive the lights; R5 drives the speaker. (For the latter, the Arduino has a tone() function which outputs a square wave of a given frequency.)

You are expected to understand both the game itself and this circuit before coming to the lab. Come prepared with some ideas as to how the program driving the hardware might work.

**Lab Work**

You are responsible for creating a complete model of the Simon game. This model should be in hard copy (either done on-line or just hand-drawn on paper) and should be handed in at the end of the lab. Be sure all names in your group are on the hand-in so you get credit for the lab.

You should start by enumerating the tasks that are involved in the game. Think about everything the program is responsible for and make sure your tasks cover it all.

Next you should create models for each of the tasks. Most of these should be relatively simple. Where tasks need to cooperate, note how that cooperation is done. You should make sure that you handle all illegal inputs (i.e. pushing a button at the wrong time) correctly in the models. You can assume the longest sequence of notes is 32. Note that there are 3 starting states (start/last/longest). If you have time, create a model of the overall system using the individual task models.