Jesse Belanger

EC 330

Homework 3

2/27/15

My design utilized a state machine with four major states to handle the four combinations of modes and button activity (recording or playback mode, and buttons pressed or not pressed). The machine starts in play mode, and is initialized to do nothing until it detects two button pushes. When the count of button pushes reaches two, the mode flag is changed to reflect that record mode is now on, and the red LED is toggled to show the user that they have entered record mode. Other, more logistical variables are also reset to clean up for new recording sessions if one or more have already been done.

While in recording mode, one state handles idle periods and tracks elapsed time. When the button is not pressed, a counter named *idle* counts down from 10 seconds, while another counter named *interval* counts up. The first is used to let the system know when it can switch to play mode (e.g. after 10 seconds of inactivity). When it gets to zero, the mode is switched back to zero and both the red and green LED are turned off. The second counter is used to track the time between consecutive button pushes, and can increment up until the system has been idle for too long.

Another state handles active periods in the recording mode. When it senses that the button has been pressed, it toggles the green LED, resets *idle* to be 10 seconds, and stores the *interval* count into an array to be used during playback. This array is 100 integers long, and thus can handle 100 toggles. If the user tries to exceed that number, the system will automatically force the switch to playback mode to prevent errors.

In playback mode, the values are read from the stored array of intervals. An interval pulled from the array is used as a counter, and gets decremented to 0 over successive calls of the interrupt handler. When the counter reaches zero, the LED is toggled and the index into the array is incremented. In this way, the LED is toggled in the same order and with the same timing as it was during the record mode. A few clean-up things also happen here to make sure that the lights return to being off after the pattern has been completely played back.

My system is limited in that it can only run for 100 toggles, as it is constrained by the maximum size of the array in memory. It is also somewhat inelegant in that, when you are initializing the record mode, the timing between button pushes is irrelevant. You could do the first press and then wait a week to do the second press, and it would still go into record mode after the second press.