Neeraj Basu

2/16/15

EC 450

Professor Giles

Homework #1: SOS Morse Code

**Overall strategy**

The idea behind this code involved breaking the SOS sequence into 6 manageable states: The first S, the first short pause, the O, the second short pause, the second S, and a long pause. All of these states where part of a switch statement that essentially looked to make sure the last state was finished, before moving onto the next one. This entire process was inside of an if statement that checked to see that the blink counter for each step was equal to zero before moving onto the next state and assigning it the proper blink interval.

**Variables Used**

I started by declaring an arbitrary unit of time that would account for the time management of the controller. I then defined each state scaled in reference to the defined unit time.

**#define** UNIT 30

blink\_counter= 0;

short\_blink= UNIT;

long\_blink = 3 \* UNIT;

pause = 7 \* UNIT;

blink\_interval=short\_blink;

state = 1; //State set to 1 so the sequence will always starts on the first S.

The global variables that were declared were the blink\_counter, the blink\_interval and the state. The state was managed by a switch statement that checked to see when the blink\_counter had executed the proper amount of XOR’s for each state. Once this was true, the state would change and the blink interval would be set to either short\_blink, long\_blink, short pause(same length as a short\_blink) or a pause (7 unit pause).

One of the challenges was making sure that the long blink still had short pauses in between each flash. This meant that the blink\_interval had to toggle between long\_blink and short\_blink. This was done by putting an if statement within case 3(the O) that looked for when the mod of the blink\_counter was equal to zero, or rather every even number. It would then make the next tick a short\_blink before toggling back to a long\_blink.

**Hardware Used**

I simply used pin 0x01 as the output LED on the board which was connected via USB to my PC.