

Homework 5

This project was to complete the tutorial “Bootable Embedded Systems for the DE0-Nano Board” and to alter it to produce a system that has a different display and lets the user press a button to change the speed of the scrolling LEDs. This homework was not too difficult, but the setup proved to be a challenge. These challenges are detailed in homework 6.

Video of working project: <https://www.youtube.com/watch?v=BlbqIXt6fqU>

I made a few changes to the C code to complete the homework. I changed the display by changing the line:

```
int LEDG_bits = 1; // pattern for the green lights  
to  
int LEDG_bits = 0x00000011; // pattern for the green lights
```

The original setting had a single light, the new line had 2 spaced out lights.

To change the speed of the scrolling lights with a button press I had to declare a KEY variable.

```
volatile int *KEY = (int *) 0x10000050;
```

Inside the while 1, I added an if statement that will add to the counter on each press, making it slower and slower.

```
if (*KEY & 1)  
counter = counter+0x190000;  
*(timer_ptr + 2) = (counter & 0xFFFF);  
*(timer_ptr + 3) = (counter >> 16) & 0xFFFF;  
*(timer_ptr + 1) = 0x6; // START = 1, CONT = 1, ITO = 0
```

The altered C code is in Appendix A. I did not know how to find the hardware address of the KEY, and eventually found it in the system.h file in the bsp directory.

PowerPlay Power Analyzer Summary	
PowerPlay Power Analyzer Status	Successful - Mon Mar 02 23:22:37 2015
Quartus II 64-Bit Version	14.1.0 Build 186 12/03/2014 SJ Web Edition
Revision Name	DE0_Nano_Basic_Computer
Top-level Entity Name	DE0_Nano_Basic_Computer
Family	Cyclone IV E
Device	EP4CE22F17C6
Power Models	Final
Total Thermal Power Dissipation	150.86 mW
Core Dynamic Thermal Power Dissipation	1.15 mW
Core Static Thermal Power Dissipation	83.94 mW
I/O Thermal Power Dissipation	65.76 mW
Power Estimation Confidence	Low: user provided insufficient toggle rate data

Appendix A: C Code

```
/* This program sweeps a green light back and forth on the LEDG lights */
enum DIR {LEFT, RIGHT};
void sweep(int *,enum DIR *);

int main(void)
{
    volatile int *LEDG_ptr = (int *) 0x10000010; // green LED address
    volatile int *timer_ptr = (int *) 0x10002000; // interval timer address
    volatile int *KEY = (int *) 0x10000050;

    int LEDG_bits = 0x00000011; // pattern for the green lights
    enum DIR shift_dir = LEFT; // pattern shifting direction

    /* set the interval timer period */
    int counter = 0x190000; // 1/(50 MHz) x 0x190000 = 33 msec
    *(timer_ptr + 2) = (counter & 0xFFFF);
    *(timer_ptr + 3) = (counter >> 16) & 0xFFFF;
    *(timer_ptr + 1) = 0x6; // START = 1, CONT = 1, ITO = 0

    while (1)
    {
        *LEDG_ptr = LEDG_bits; // write to the green lights
        sweep (&LEDG_bits, &shift_dir); // shift the pattern left or right

        if (*KEY & 1)
            counter = counter+0x190000;
        *(timer_ptr + 2) = (counter & 0xFFFF);
        *(timer_ptr + 3) = (counter >> 16) & 0xFFFF;
        *(timer_ptr + 1) = 0x6; // START = 1, CONT = 1, ITO = 0
        //else
        // *LEDG_ptr = 0x00;

        while ( (*timer_ptr & 0x1) == 0 ) // wait for timeout
            ;
        *timer_ptr = 0; // reset the timeout bit
    }
}

/* shift the pattern shown on the LEDs */
void sweep (int *pattern,enum DIR *dir)
{
    if (*dir == LEFT)
        if (*pattern & 0x80)
            *dir = RIGHT;
        else
            *pattern = *pattern << 1;
    else
        if (*pattern & 0x01)
            *dir = LEFT;
        else
            *pattern = *pattern >> 1;
}
```