Programming for Embedded Systems
Lab 1: Swinging LED Images

February 2, 2014

Introduction

If you have ever watched a bright light in a dark room you may have noticed that when the light moves quickly it appears to leave a trail of light behind it. If the light is turning on and off rapidly then you will see a dashed trail rather than an unbroken one. This phenomenon has been taken advantage to build "swing clocks" that seem to write the time in the air. These clocks have a thin swinging arm with a column of LEDs on them. The LEDs light up as the arm swings and our eyes see a trail of light that spells out a message or displays the time.

To keep this project simple we will concentrate on just the LEDs and the timing and you will provide motion with your own arm. We have already used the watchdog timer for precision timing and know how to output a voltage from a pin. Using a breadboard, LEDs, resistors, and a few jumper wires we will write a few simple messages in the air.

Hardware Setup

Figure 1 shows how to run several LEDs in series with resistors and the MSP430 output pins to create a column of 5 LEDs. An MSP430 attached to the Launchpad can only output 3 volts, so the resistance of the resistors is the only way to tune the brightness of the LEDs. You can use LEDs without a resistor, but running too high a voltage through an LED for too long will burn it out.

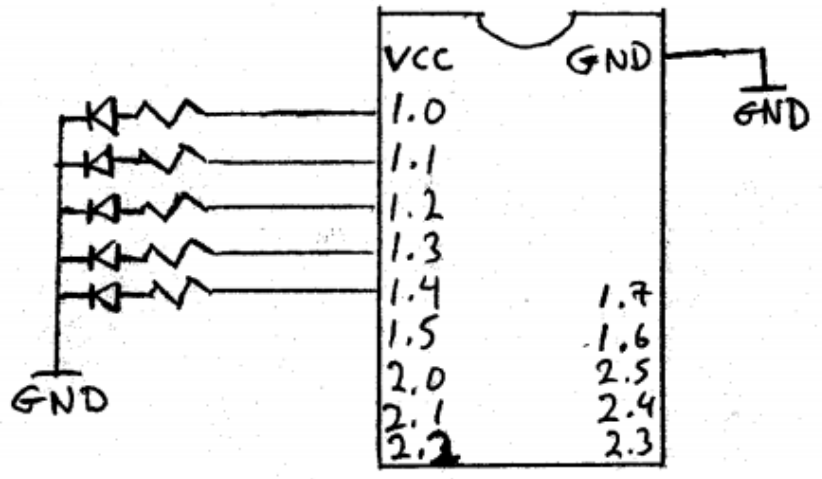

Figure 1: Diagram of the LED, resistor, and pin connections.

Software Setup

We will use a column of 5 LEDs to draw images. The column of LEDs is equivalent to a column in an image. An image has multiple columns, so we move the LEDs over slightly to make the second column in the image. The trick in doing this is to keep the LEDs moving while the going from column to column. You should switch columns at around 150Hz.

Each pin will be responsible for a row of output, and each column represents the state at a particular moment in time. As an example, imagine we want to draw a smiley face that looks like this (please forgive my drawing):

	0	1	2	3	4	5
P1.0		*			*	
P1.1						
P1.2	*					*
P1.3		*			*	
P1.4			*	*		

In our code, we could represent the columns like this:

```
const unsigned char face[] = {BIT2, BIT0|BIT3, BIT4, BIT4, BIT0|BIT3, BIT2,
                0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ,0 ,0, 0, 0, 0, 0, 0};
```

We pad the smiley face with 0s to leave a space between one smiley face and the next. To display a column we simply set the output pin to that state. For example, to switch to the first column we would do this:

```
P1OUT = face[0];
```

We will keep a counter to remember which column we are on. Once the column count reaches sizeof(face) we will wrap back around to 0.

In Class Lab Assignment: Draw a Smiley Face)

Your first task is to make the LEDs flash in a smiling face pattern so that when the LEDs are moved rapidly sideways a smiley face appears in the air. Use the watchdog timer to switch columns at a rate of about 150Hz. It will take six cycles to display the entire face. You may draw the face differently if you think something else looks better.

Your code should take the following steps:

1. Stop the watchdog timer
2. Set up pins 1.0 to 1.4 as output pins
3. Set up the watchdog timer interrupt to switch columns at about 150Hz.
4. Each time the interrupt is called, shift to the next column. Go back to the first column upon reaching the end.

Take Home Project

Work outside of class to expand the in-lab assignment so that you can display text. Write a program that displays any three letter word. Since we are displaying text, leave an empty column between each character in your word.

When you are figuring out the columns to use for a character, try drawing the character as ascii art first. For example, you might lay out a "T" like this:

P1.0	X	X	X	X	X
P1.1			X		
P1.2			X		
P1.3			X		
P1.4			X		

Your code should be submitted to sakai by midnight on Wednesday, February 4. There will also be a quiz on Wednesday covering the project material: interrupts and interrupt vectors, the watchdog timer and its interrupt, and basic digital output in the MSP430.

Grading

30% Repeatedly flash one character.

25% Repeatedly flash two characters.

20% Repeatedly flash three characters.

10% Timing makes the word visible. There should be a delay between characters and a longer delay after the word.

15% Code quality. Your program should be commented, well formatted, and variable names should be reasonable.