Course Title: CEE-345 Microprocessor System Design

LED control using Capacitive Touch Sensor

Dylan Christopherson

Dr. Cheng Liu

University of Wisconsin-Stout

Menomonie, WI 54751

May 10th, 2019

# TABLE OF CONTENTS

TABLE OF CONTENTS 2

1.0. PROJECT DESCRIPTION 3

2.0. BILL OF MATERIALS 4

3.0. DEsIGN PROCESS 5

4.0. STUmBLING BLOCKS AND CONCLUSION 10

# 

# PROJECT DESCRIPTION

This project was designed to use a capacitive touch sensor to control an on board LED on the Freedom Development Board. The project scope changed to include three tactile push buttons that would change the color of the onboard LED.

**1.1. *Introduction***

For this project, I worked on taking small steps to achieve my larger goal. My first goal was to receive input from the capacitive touch sensor. Initially, I was hoping to develop this myself. However, that eventually proved to be unrealistic. I began to look into pre packaged touch software. Dr. Liu then guided me toward Touch Sensing Software (TSS), which is a software pack put out by NXP Semiconductors. I was able to download and install the software relatively easily. Within the project, there were several examples showing off the TSS. There was an example using the capacitive sensor which was exactly where I wanted to go with my project. The project didn’t immediately build right away as there were issues with the RAM for Algorithm values in the target driver setup dialogue box. Eventually, I was able to build the project.

After I had successfully set up the hardware for the touch sensor, I began to implement code for the tactile push buttons. I relied on a previous lab, and with some work, was able to implement tactile bush buttons to change the onboard LED different colors.

# BILL OF MATERIALS

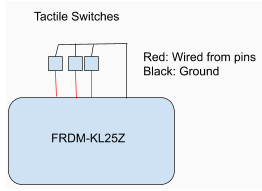
|  |  |
| --- | --- |
| Part Name | Cost |
| Freedom Board | $20 |
| Wires | $2 |
| Breadboard | $3 |
| Tactile Switches | $2 |
| Total | $27 |

# 3.0. DEIGN PROCESS

# 3.1. HARDWARE DEVELOPMENT

Hardware development was relatively easy for this project. For the first phase of the project, just the onboard LED and onboard capacitive sensor were required. For the second phase, I just needed three tactile switches, a breadboard, and a couple of wires. One pin of each of the tactile switches were wired to ground while the others were wired to their respective pins on the Freedom board.

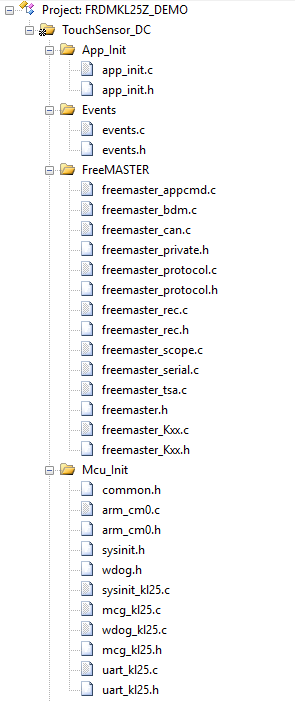
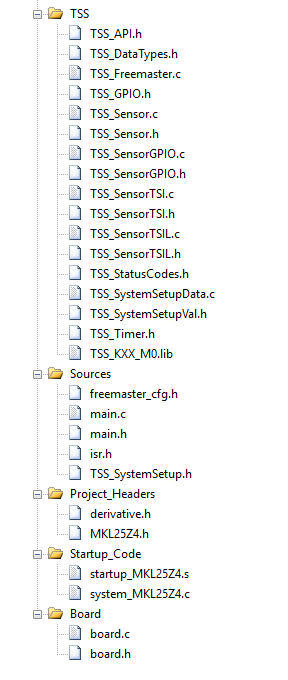
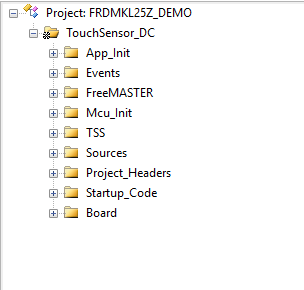
## 3.2. WIRING DISGRAM



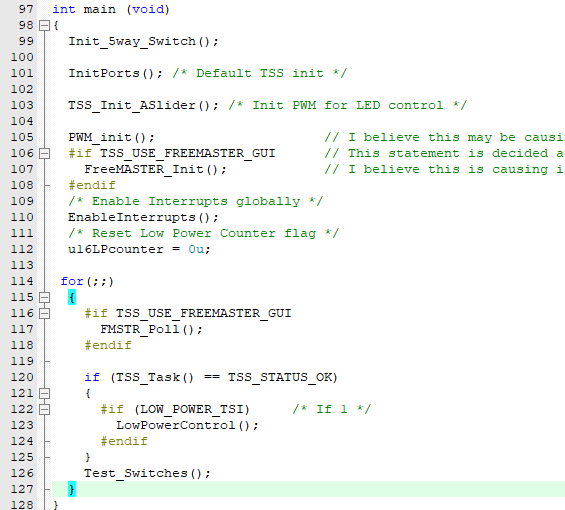
Capacitive Touch Sensor

## 3.3. SOFTWARE DEVELOPMENT

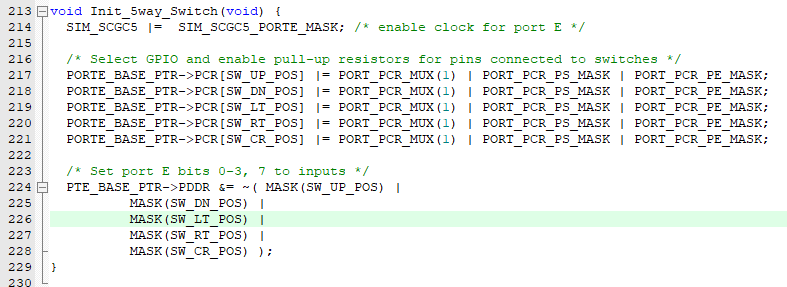
The Touch Sending Software (TSS) that I downloaded from NXP Semiconductors was a hefty package that had the ability to handle many different types of projects. This meant the file structure was quite dense and there were a lot of different files and dependencies. This made it difficult to get a general idea of what different files did and how they related to the project. Below is the file structure for the TSS project.



The most important code exists within the main.c file.

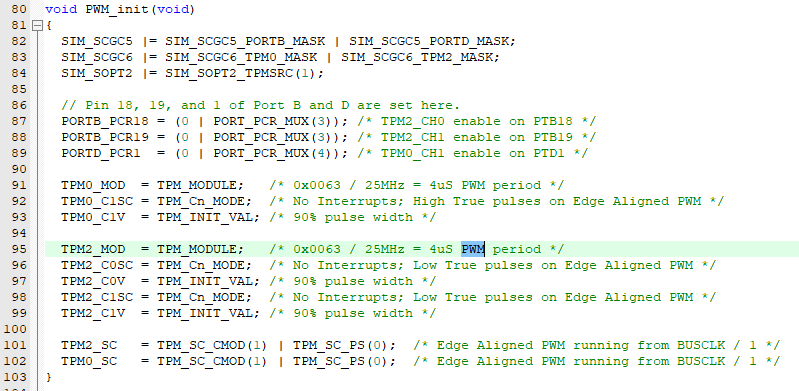


The Init\_5way\_Switch() function initializes the ports on board of the freedom board to accept input from the tactile switches, this allows the board to read in button presses to change the color of the LED.



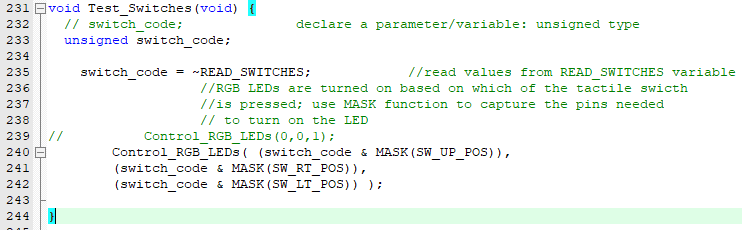
Within the main function, the InitPorts() function is the default TSS init function, which was provided by the TSS software. The TSS\_Init\_ASlider() function, also provided by the TSS software, is to Init PWM for LED control.

The PWM\_Init() function is important because it initializes the timers and output pins for the RGB LED. I initially had issues when integrating color LEDs because I was initializing the clock and output pins for the RGB LED twice in two different functions causing conflicts. I eventually fixed it but just using the PWM\_Init() function.

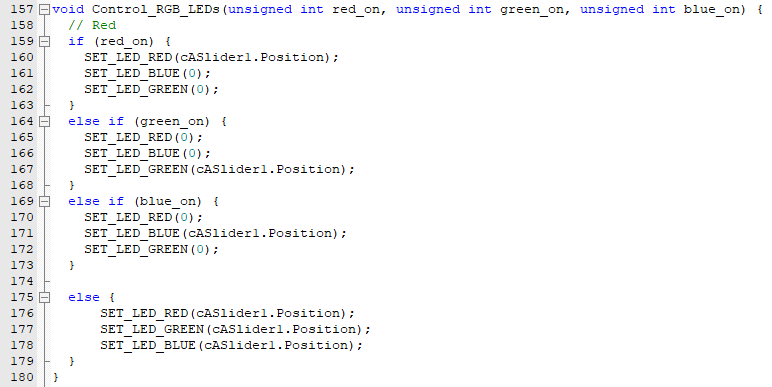


Generally, the other functions are provided by the TSS software and I’m unsure what they do as I didn’t do anything in them.

The other important function is called Test\_Switches(). Test\_Switches() reads in if a tactile button is pressed, if it is, the value of that tactile switch is passed into the Control\_RGB\_LEDs function.



The Control\_RGB\_LEDs function is important because it sets the color of the RGB LED based on the value cASlider1.Position. Both SET\_LED\_COLOR() and cASlider1.Position are provided by TSS. The issue was that it took me a very long time to find them in the project because of its scope. cASlider1.Position represents the location on the capacitive touch sensor and brightens and lowers the LED based on it’s location. If the red tactile switch is on, the LED is set to the cASlider1.Position while the other colors are set to zero. The same goes for blue and green. When no buttons are pressed, all values are set to the cASlider1.Position which makes the LED white. After eight seconds, if the capacitive sensor is not touches, it begins blinking a green color to indicate nothing is being sensed.



# 4.0. PROJECT ILLUSTRATION

# 

As you can see, the project is relatively simple. Just three tactile switches wired up on the bread board and then connected to the Freedom board.

# 4.0. STUMBLING BLOCKS AND CONCLUSION

In summary, this project was challenging, but once you had the necessary information, there were was only a couple of key items to know. Some of the biggest issues were figuring out the different settings necessary to build the Touch Sensing Software based project. Then, integrating two projects together to allow for the use of different color LEDs with the capacitive touch slider. Finally, figuring out the most important functions and variables needed to effectively use the capacitive touch slider with the onboard LED on the Freedom board. A couple things I learned from this project is that sometimes, there just isn’t going to be a lot of coding. This project required very little software development and mostly consisted of integrating multiple sources of code effectively. I also learned that I won’t necessarily always like projects. I don’t really like the Freedom board, and I think that severely affected my willingness to put time into the project. Often times it was on the backburned and I’d push it off till later. Overall, even though I didn’t necessarily enjoy it that much, I did learn useful skills that I’ll be able to use both in my internship this summer, and in future classes.