Platform: Final Project Report

**Course:**  CEE 345 – Microprocessor System Design

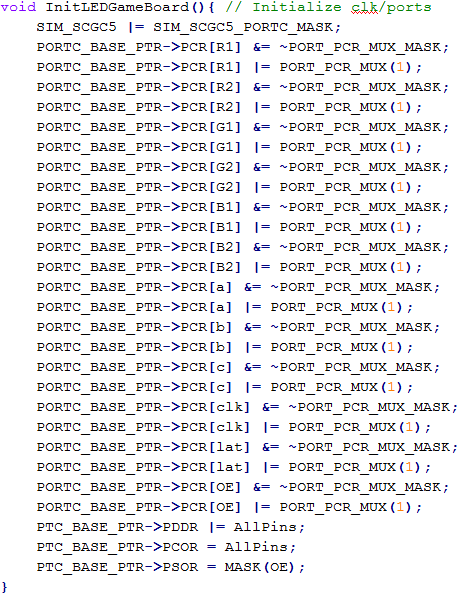
**Project:**  Platform (vertically scrolling LED game)

**Introduction:**

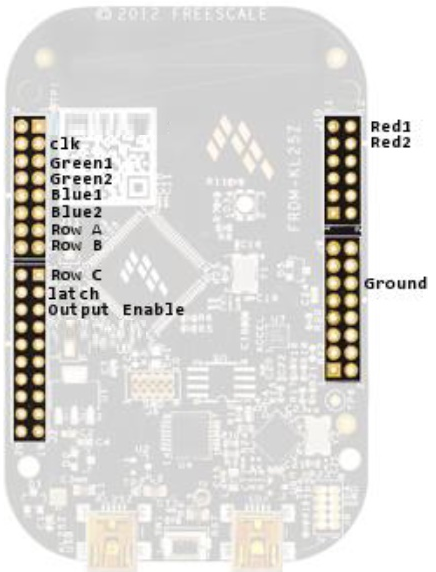
There are many applications that a microcontroller could be used for. These applications are often useful, by either speeding up a task or making it more efficient, but sometimes they can be just for fun: in this case, a simple game. Creating a simple game is a good way to learn more about microcontroller development while also creating something that can be played for fun by the developer and other classmates. We had labs that dealt with interfacing with the GPIO pins, the on-board LED, and the accelerometer, but there was no lab that utilized the on-board touch sensor. I wanted to create a game that could be played with side-to-side scrolling that was already pretty well-known, and put a small twist onto it.

**Design:**

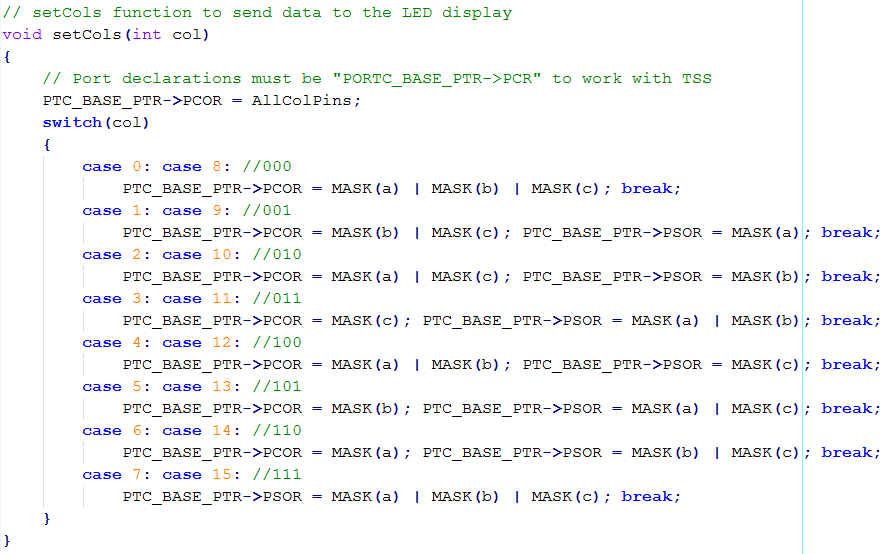
The name of the game I have made is called “Platform”, and it is a simple game that runs on the KL25Z Freedom Board. It uses a 16x32 LED matrix as a display, and the on-board touch-sensor as an input device. When the Freedom Board is turned on, there is a starting screen that displays “Platform | By Cory”. When touching the touch-slider, the main game begins. The player’s “cursor” begins in the middle of the board, and begins to get carried upward. They must scroll their “cursor” from side-to-side to fall through the gaps of the platforms. Being carried to the top of the display is a loss, but reaching the bottom of it is a win. After the game, another screen is shown that depends on a win or a loss.

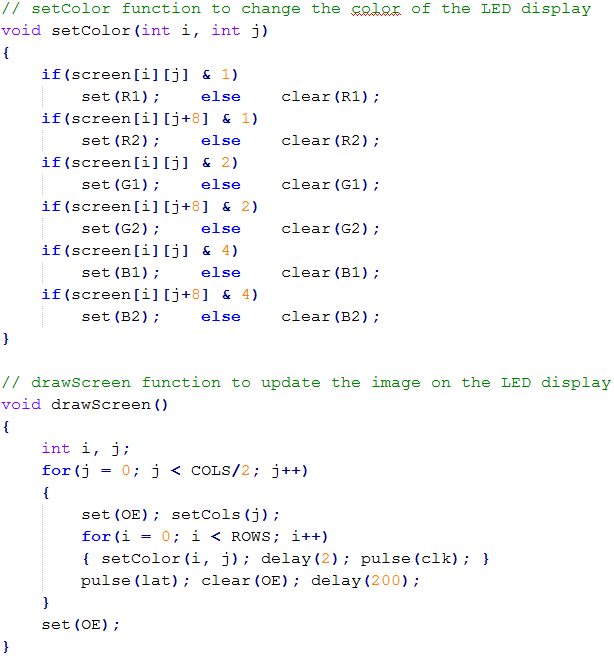
**Schematic:**

* Enable PORTC clock
* Two pins for each LED color
  + Red (R1/R2)
  + Green (G1/G2)
  + Blue (B1/B2)
* Pin for LED display “Latch”
* Pin for LED display “Enable”

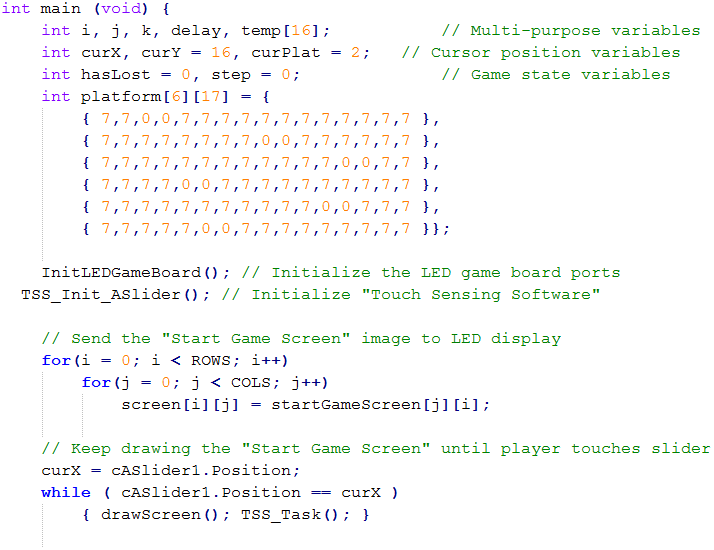
****

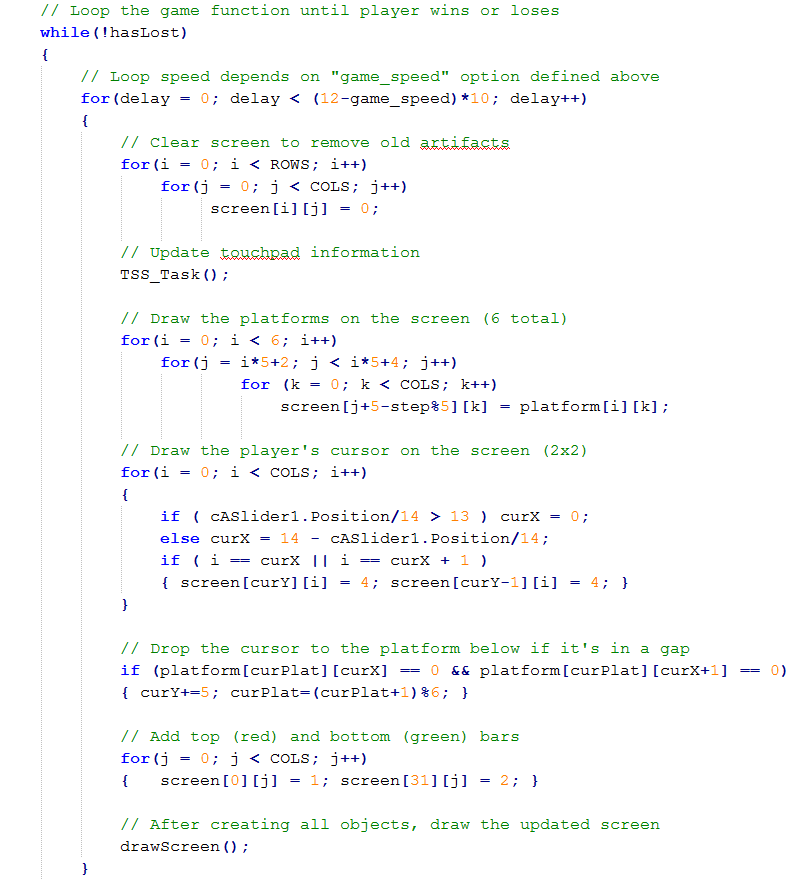
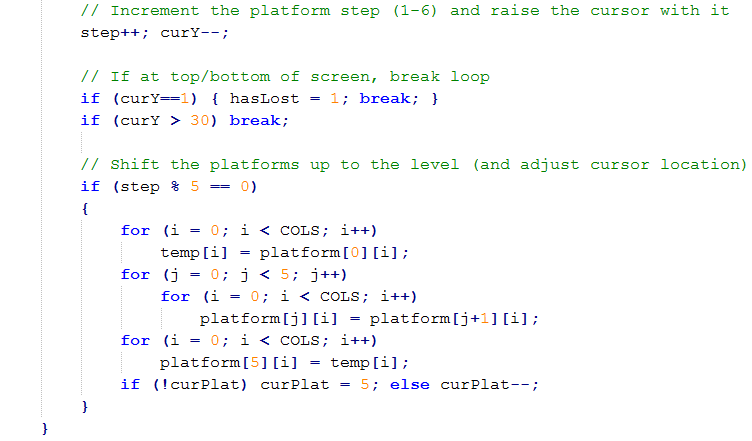
**Code to manage LED Display:**

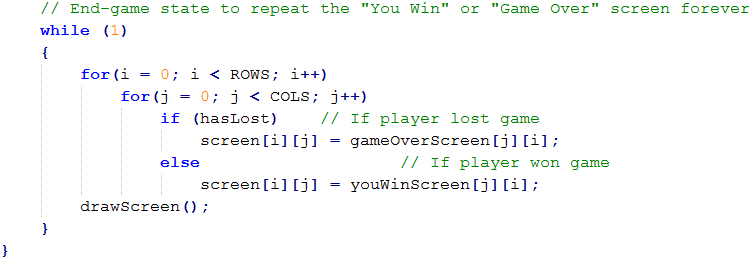




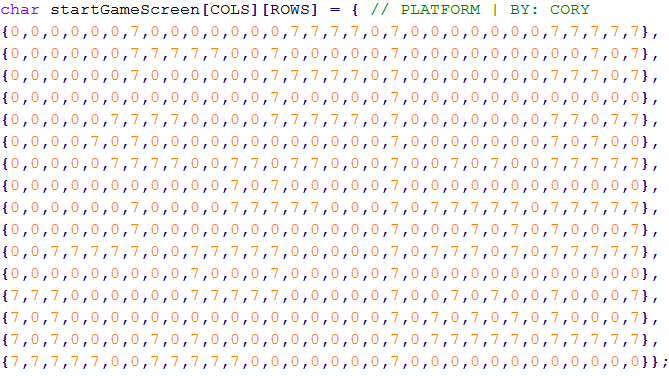
**Code for the game main function:**







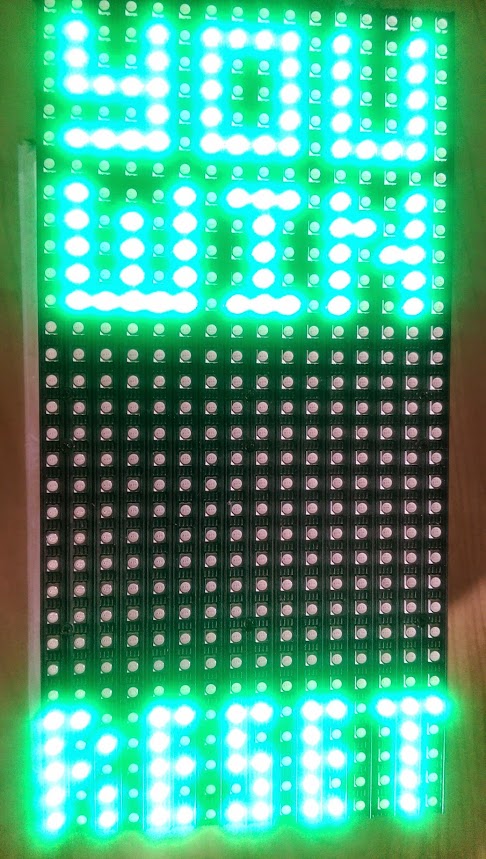
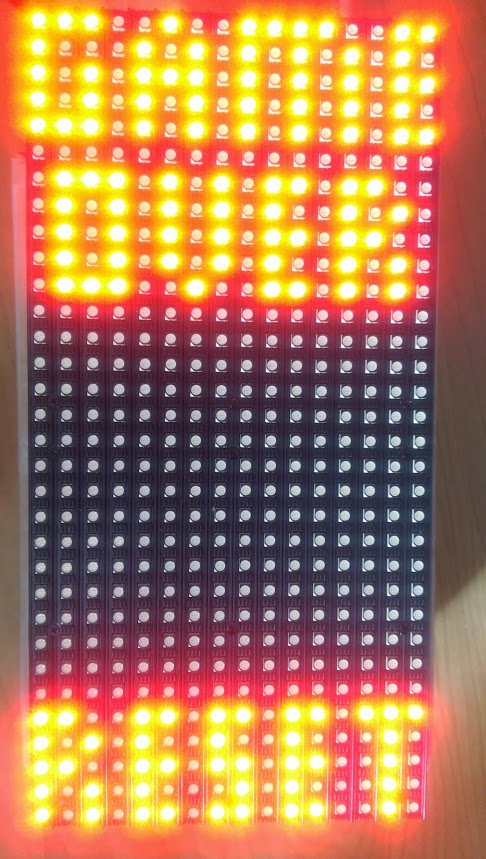
**Code for the static start/win/lose images:**



**Results:**

The design for this project was quite successful, and the specified criteria have all been met successfully. The game can be played on the LED display as it was meant to be, controlled with the touch-slider on the Freedom Board. This can be demonstrated by playing the game and viewing the starting screen, game screen, and win/loss screen (shown below).



**Stumbling Blocks:**

* The TSS library was out of date compared to the MDK being used, so it required some edits to the port declaration portions of code to be compatible.
* The TSS library requires a large amount of overhead that may slow down the program.
* The LED matrix only works properly when plugged in, limiting its portability and ease of use.
* The 16x32 display is enough, but a higher resolution (32x64 or more) would look a lot better (especially with text) and provide a better experience.
* Gameplay can be abused by rapidly swiping side-to-side on the slider, since the touch surface is so small, to fall through all of the gaps very quickly.

**Conclusion:**

Overall, the project was a great learning experience with ARM development. It was useful to understand how to add external libraries to my project (the TSS library) and utilize them in my solution. The game that was created can be fun to play, and it is nice to be able to actually enjoy the project that I’ve created. Because the LED display code is already created, it is very modular and easy to create other games that can work with it. This is also true with the TSS library implementation, as it can be used with other games easily, using the current code as a base model. It would be very interesting to see what other applications can be made with the LED display and TSS slider, and encourage other future students to attempt such a design for their final projects.