Ressa Reneth Sarreal
CMPE 311 HW 5 Report
Due: 12/13/2016

The assignment is to implement a slot machine game. In the three files, U0_LCD_Driver.h, U0_LCD_Driver.c and main.c, the AVR Butterfly is used by the user to play the slot machine game. The user will have ten tries to earn a predetermined score to win the game. If after ten tries, the user does not reach the score, they will lose. The user will be able to see their score, or reset the game at any time they want.

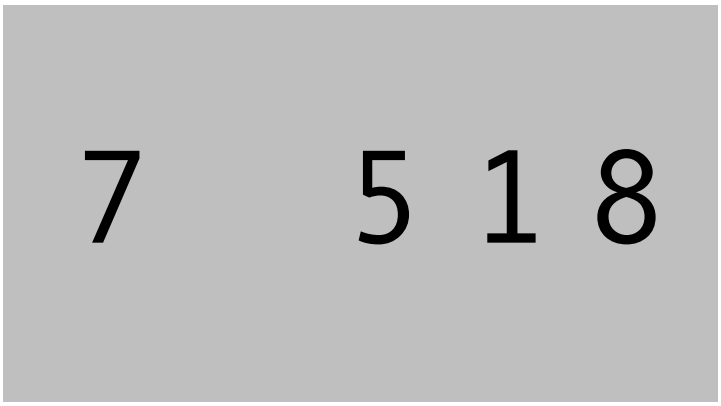

Figure 1: What is displayed when the player is on screen 1. The player has 7 turns remaining and has either stopped when the slot machine was at 518 or has not stopped the slot machine, and the slot machine is still rotating through the numbers. Hitting left on the joystick will show screen 1.
When the player has not started the game the screen shown in Figure 1 will display:

```
0 _  _ 0 0 0
```

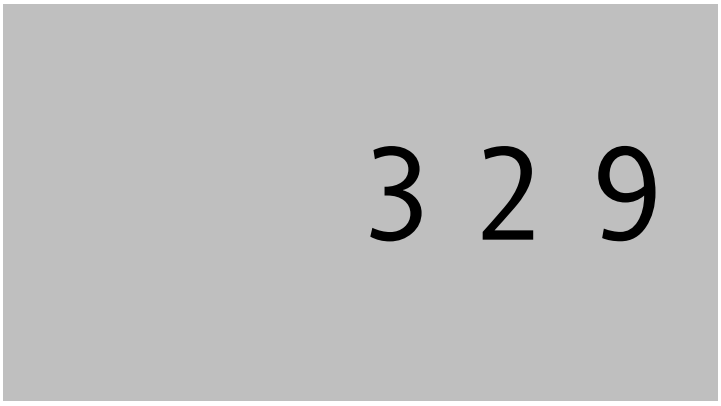

Figure 2: This is what is displayed when the player is on screen 2 and has a total score of 329 points. Hitting right on the joystick will show screen 2.
When the player has not started the game, the screen shown in Figure 2 will display :

```
_  _  _ 0 0 0
```

Ressa Reneth Sarreal
CMPE 311 HW 5 Report
Due: 12/13/2016

A few obstacles I had to overcome were that for some reason my screen 2, where the user views their total score for the game, was constantly refreshing. It was very hard to see the score since the numbers were constantly flashing 000 then to whatever the score actually was. I realized that I had forgotten to put one LCD display chunk of code inside an if statement for 'if the user was currently on screen 1', so it was happening for screen 2 as well, which was not intended. Another issue I had faced was that while I was initially troubleshooting, I had realized that my left and pushbutton on the joystick were not functional. This interfered with testing for a few minutes, but I had just realized that I did not initialize the pins. This was an easy fix.

The main functionality of the code is based around having the joystick work so that up was a reset button, down was the slot machine spin activator, left was to switch to screen 1 which was the actual slot machine and turn counter, and right was to switch to screen 2 to view total score. I used interrupts to make the joystick functional. There was also the buzzer that had to be implemented so that it buzzed twice when the user won or lost, and the LED had to be configured and coded so that it would blink twice when the user won. The LCD display was also implemented using the LCD_Driver code. The basic flow of the code is as follows:

```
#DEFINE WINNING_SCORE  150
bool spinning; // will be T=1/F=0 depending on pressed buttons
int timePassed;
int remainingTurns = 10; // number of turns starts at 10
int totalScore, roundScore;
int hundredsVal, tensVal, onesVal; // digits at hundreds, tens,
                                   // or ones place

// the functions used should be self explanatory
int main() {
     INITIALIZE_ALL();
     while (true) {
          while(spinning) {
               if (timePassed % 10 == 0) {
                    hundredsVal = (hundredsVal % 10) + 1;
                    if(currentScreen == 1) {
                         DisplayHundredsValue();
                    }
               }
               if (timePassed % 15 == 0) {
                    tensVal = (tensVal % 10) + 1;
                    if (currentScreen == 1) {
                         DisplayTensValue();
                    }
               }
               if (timePassed % 20 == 0) {
                    onesVal = (onesVal % 10) + 1;
```

```
                    if (currentScreen == 1) {
                        DisplayOnesValue();
                    }
                }
                _delay_ms(5);
                timePassed += 5;
            } // not spinning anymore

            roundScore = hundredsVal*100 + tensVal*10 + onesVal;
            totalScore += roundScore;
            if (currentScreen == 2) {
                DisplayTotalScore();
            }
            if (totalScore >= WINNING_SCORE) {
                WinBuzzer();
                WinLight();
            }

            if (remainingTurns <= 0 ) {
                LoseBuzzer();
            }
        }
        return 0;
}
```

This project allowed me to sum up all of the knowledge obtained throughout the course into one large project where I implemented interrupts, LCD_driver usage, pin configuration, basic I/O for LEDs and Buzzers, and wrote logical, clean code. This was a really great year and I really think I learned a lot, and enjoyed the projects. All of the TAs were really cool and informative! Have a wonderful winter break.