LAB 3

Target Bowling

Haritha Gnanasegar

Professor Leyla Nazhand-Ali

ECE 2534-82953

Time Frame – March 9

Report Summary

This report represents the basic concepts and ideas implemented in the Target Bowling project. It briefly describes the functionality of the game on the microcontroller and how the game is to be played. The report describes the generation of sound and use of joystick in the project and how it is implemented.

Project Description

The project has two main FSM's that run the game. The first FSM is for the user to choose whether to play the game, display the high score or look at how to play. If the user enters any of the modes, then the joystick is to be pressed to come to the main menu(Figure 1). Once the user enters the play game mode, the second FSM comes to play. The second FSM has five modes: game display (the initial game set-up), throw mode, roll mode, move and direction.

The first time the user enters the game display mode and after the initial display the games goes into the throw mode. The game functionality is executed in the throw mode. When in the throw mode and the Joystick button is pushed the game goes into the roll mode and the ball rolled. After the roll is over the game returns to the throw mode.

S1 is pushed: one can change the position of the ball from left to right and back. If S1 is pressed again the games goes into the throw mode.

S2 is pushed: one can change the direction in which the ball is bowled that is 5,10,10 degrees to the left or the right. If the S2 button is pushed again the ball is set the angle and is bowled in the same. After one throw the ball is reset to the original bowling angle which is zero.

The user has 10 bowling trials and then a game over screen appears with the final score scored. From the game over screen to go back to the main menu the user is to press the Joystick button.

The user can check the high scores from the menu which updates after each round (10 bowls) are over and the game can be played again by the user (Figure 3).

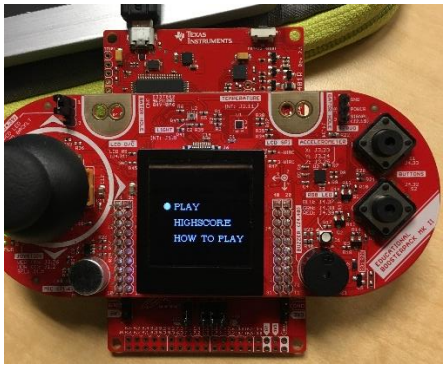


Figure 1: Menu

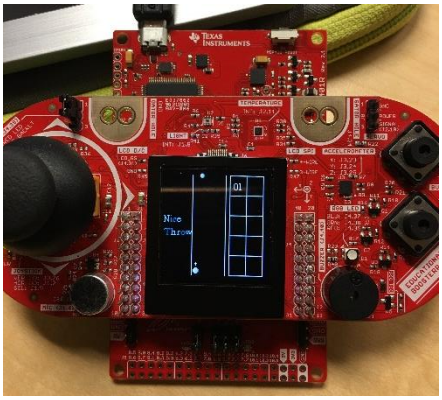


Figure 2:Hit Pin

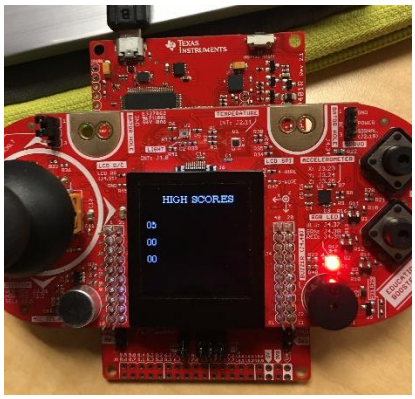


Figure 3: High Score

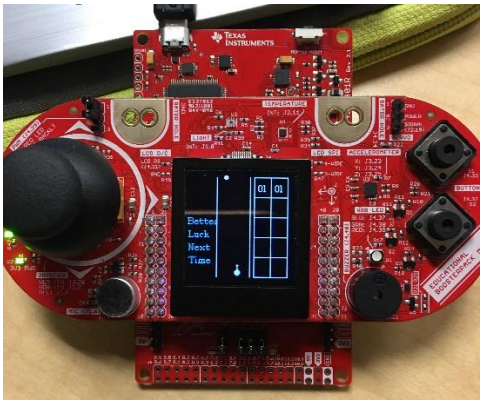


Figure 4: Missed Pin

Extra Credit Implementation:

Gutter implementation: When the player bowls the ball at different angles once the ball hit the end points the ball continues in the straight path inside going past the boundaries.

Extra comments: When the user hits the pin a message is printed “Nice Throw” and when the user misses the throw then the game prints the message “Better Luck Next Time” (Figure 2 and Figure 4)

For the game over an image is printed with the game and the final score of the game.

Sound Generation

In the project, the buzzer is used for sound generation. A tap sound is generated when the bowling ball hits one the pin. For this the buzzer on the MSP432 was configured. By reading the connectivity sheet, it was seen that the buzzer is connected to Port 2 and Pin 7 and is attached to the TimerA 0 with channel 4. Hence in the configuration it is configured to register 4. The output mode for the buzzer was set so that the given input is first set and the tone if played then, the timer resets it. This process continues until the tone is completed.

To calculate the time period the following math was followed:

Pulse period = $\frac{1}{Pulse\ frequency}$

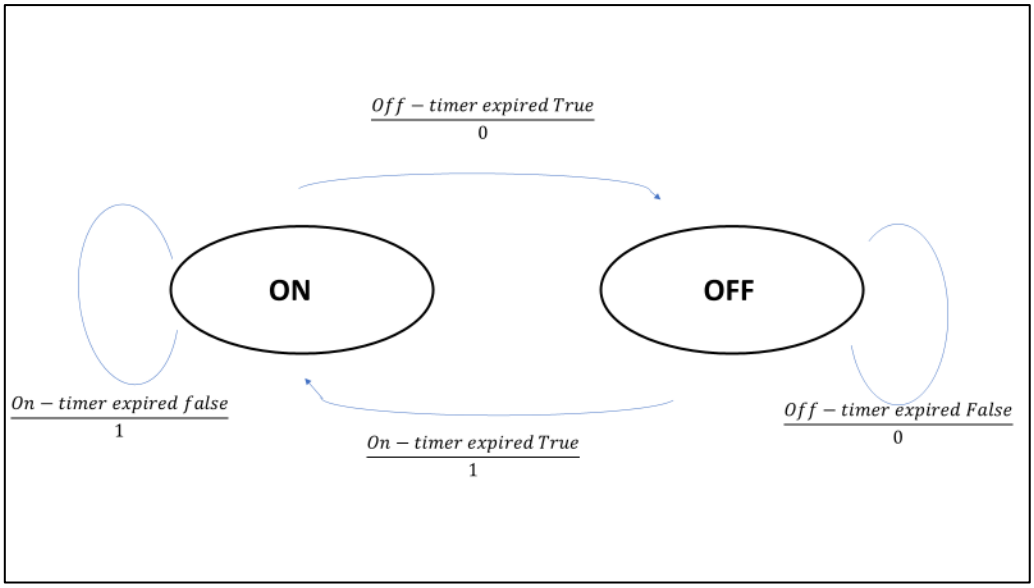
Period of the Counter Clock = Clock Divider * Period of the Source Clock

Period of the Source Clock = 1/48M

Period of the note * Clock Divider/48M = 1/Pulse Frequency

Period of the Note = 48M/(Pulse Frequency * Clock Divider)

The PWM follows the following FSM



Using Joystick

The output from the A/D convertor was used in this project extensively

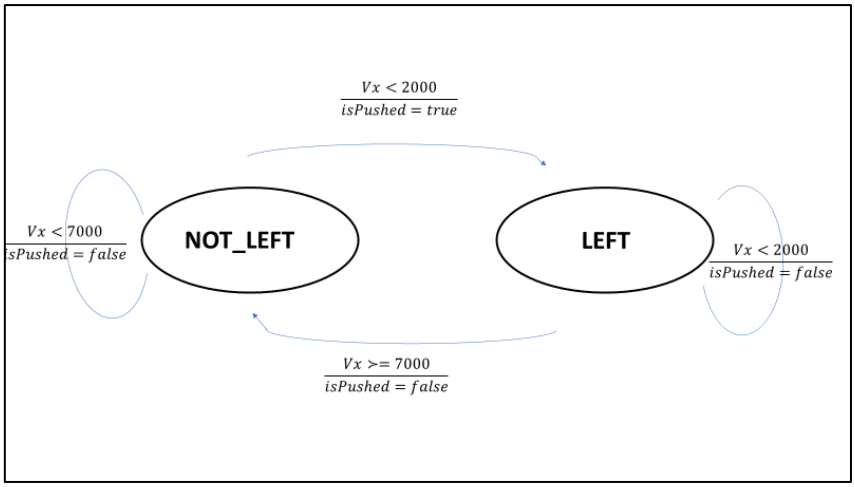
Change of Speed the ball rolls:

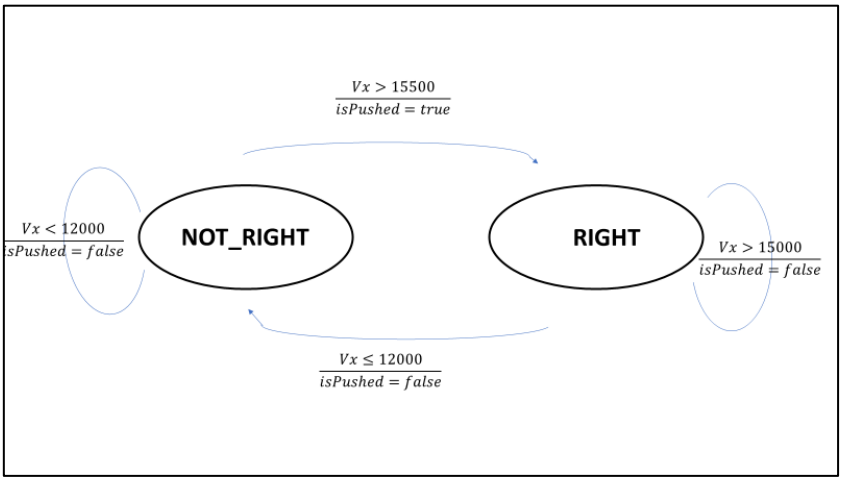
The speed of the ball changes as the load value of the hardware timer is changed. This change occurs when the A/D output on the y has a greater value (Vy) than 7500 this change takes place, where Vy is the threshold this change is to take place. The program is designed so that when the threshold is greater than 7500 and less than 9000 the load value is set to 9600 and the ball moves at its slowest speed. When the user moves the joystick and the output read from the A/D is greater than 9000 and less than 12000 we have the load change to move a little faster speed. Similarly, if the threshold is greater than the 12000 the load changes to have the ball even faster.

Hence, with the output received from the A/D the speed of the ball was changed.

Changing the position of the ball from left to right on the bowling lane:

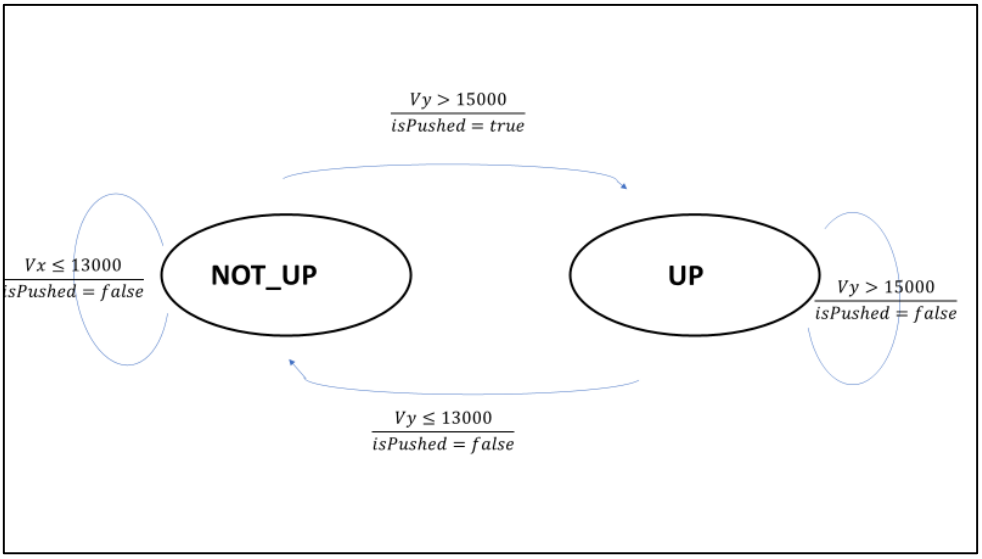
In the game when the S1 is pushed the Joystick can be used to change the position of the bowling ball from left to right. This was achieved with a debounced FSM.

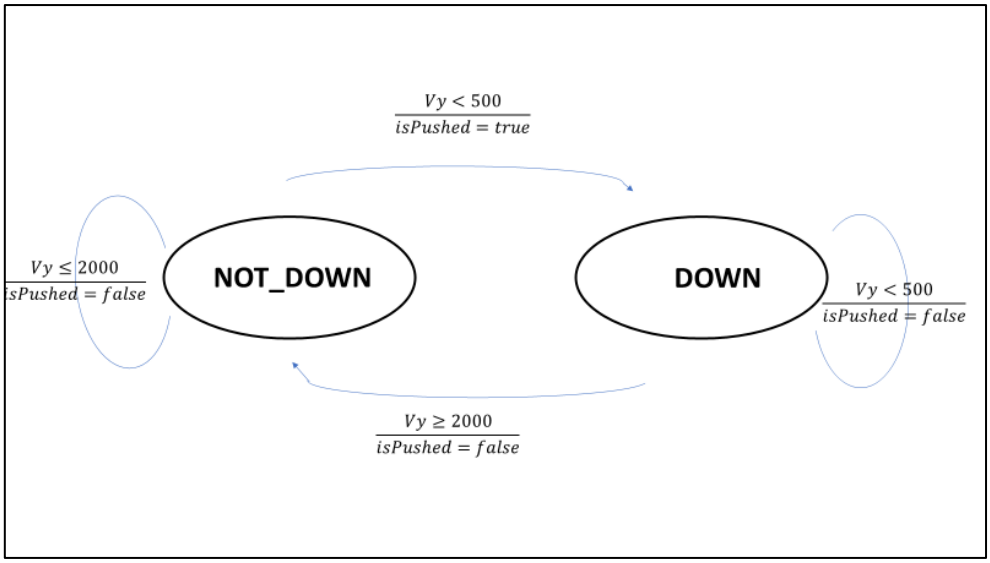




Moving the Cursor on the menu screen:

After three seconds of the Splash screen, the user views a menu and can go back and forth from high score, how to play, and the game with the help of moving the joystick up and down. This is achieved in a similar manner as changing the position of the ball the only difference being this is by changing the threshold in the y direction.





Conclusion

Key take away from the project:

Starting the project, the day it came out helped me organize my time as from project 2 I didn't start until one week in and it was hard to mange to the time with other classes. But starting this project early gave me time to work a little by little each day.

Doing this project, I got more comfortable using the debugger and following the variables as lot of my errors was bad initialization and the debugger helped me look at it better.

Doing this project, I learnt how to break my code into different methods and putting them in different .h and .c files. I am also more comfortable to use enums now as I didn't use any in project 2.