

## **Methodology**

Implementation began with the LED matrix on the LPC 2148 microcontroller. The LPC 2148 microcontroller has two on-chip SPI serial communication channels. The shift registers are accessed via the SPI bus, channel #0, and signal P0.15 is used as device select pin. In order to initialize the LED matrix the following was done: bits per transfer was set up, the CPHA mode was enabled, the SCK was set to active high, SPI was set as master mode, and the transfer bits were defined to be 16. Then, the functions of the matrix row number were set as well as the display for the LED matrix. After this was set up, when one specific point of the matrix needs to be lit, the row number and value to the setMatrixRow function needs to be passed. Now when the Matrix is displayed, the point will light up.

The Joystick switch has five internal switches, one for the four directions and one center, push-down switch. All of the input pins (P0.16 – P0.20) can be programmed as interrupt inputs. In the program, the five states of the switch are initialized and the initJoystick function and updateJoystick function are defined to set up and control the Joystick in any direction.

The main parts of the program are to control the paddle and the ball. The paddle contains four continuous points in a line. The row number of the paddle is defined as zero since this is the first line of the matrix. By tilting the joystick left or right, the paddle can be moved left or right. The ball is generated randomly for every cycle. It will first come out from a random position of the bottom row, and then move upward in a random direction. The path of the ball is defined by a random number generator function, however it can only move to neighboring LEDs. When the ball hits the paddle, it is programmed to bounce back and move downward once again in a randomly determined direction. At the point the ball bounces off the paddle, one cycle is counted and the score is increased by one.

The LCD code used in this project is the code developed in a previous lab. It has the LCD\_Init, LCD\_CommandWrite, LCD\_DataWrite functions. Once the ball hits the paddle, one point is added to the score. By using an integer to character conversion the score is displayed on the LCD screen. The “catch” and “miss” phases are classified using an if-else statement. The use of an if-else statement to determine the two phases will be easier to implement because the value of the paddle’s location could only be 240, 120, 60, 30, and 15, and the value of the ball’s location could be 128, 64, 32, ..., 1. If the ball misses the paddle, the while loop breaks, the game is over, and ‘YOU LOSE’ is displayed on the LCD screen. If the score count reaches five (defined by the programmer), the ‘CONGRATULATION’ will be displayed on the screen.

The signals P0.12 and P0.21 are used to control the movement of the step motor. If the score is added to five, besides displaying the message ‘CONGRATULATION’, the step motor is engaged to sweep back and forth at a constant speed to celebrate with the winner. This is done using the forward and reverse functions.