ELEC 327 Final Project Report - Oliver MacGregor

Link to Demo Video:
https://drive.google.com/file/d/1K-JdOMDA46po-demzzSUuGIaSmHMHkWU/view?usp=sharing
(Must be logged into Rice gmail account to view).

Abstract:

For my ELEC327 final project, I populated the provided TicTacToe PCB and wrote code for the MSP430 that allows two users to play repeated games of TicTacToe.

Hardware Design:

LED Display:

The TicTacToe board contains a 3 x 3 RGB LED array that is used as the primary display for the TicTacToe game. Each LED can be set to 3 primary colors (red, green or blue), or some combination of the 3 to create many different colors.

Each RGB led is actually an integrated packaging of 3 distinct LEDs: one red, one green and one blue. The LEDs are configured with a common anode and isolated cathodes. To turn on an LED, current must be run through the anode and out of the cathode of the LED to be turned on. This is done by applying a voltage to the anode, causing current to flow through any of the cathodes that are connected to ground via a series resistor, which acts to limit the current through the LED.

The MSP430 on the TicTacToe board has a unique IO pin routed to each of the common anodes of the 9 LEDs. Also, three more IOs are attached, through a series resistor, to one of the three cathodes on the RGB LED.

To turn an LED on, a logic high voltage is applied to the anode of the desired LED. The color of all of the LEDs that are currently on can be controlled by changing the voltage on the RED, GREEN and BLUE IO pins. By connecting these IO ports to ground, a potential is applied across the associated LED, exceeding the reverse voltage of the LED and causing current to flow, and thus light to be emitted.

Buttons

To allow the user to interface with the TicTacToe game, two buttons are used. Each button has an MSP430 IO (input) port attached to it, which is pulled up internally. By pressing the button, a contacter connect the IO port to ground, and thus creating a logic low reading at the input. Reading these high or low voltages can be used to control the flow of the game.

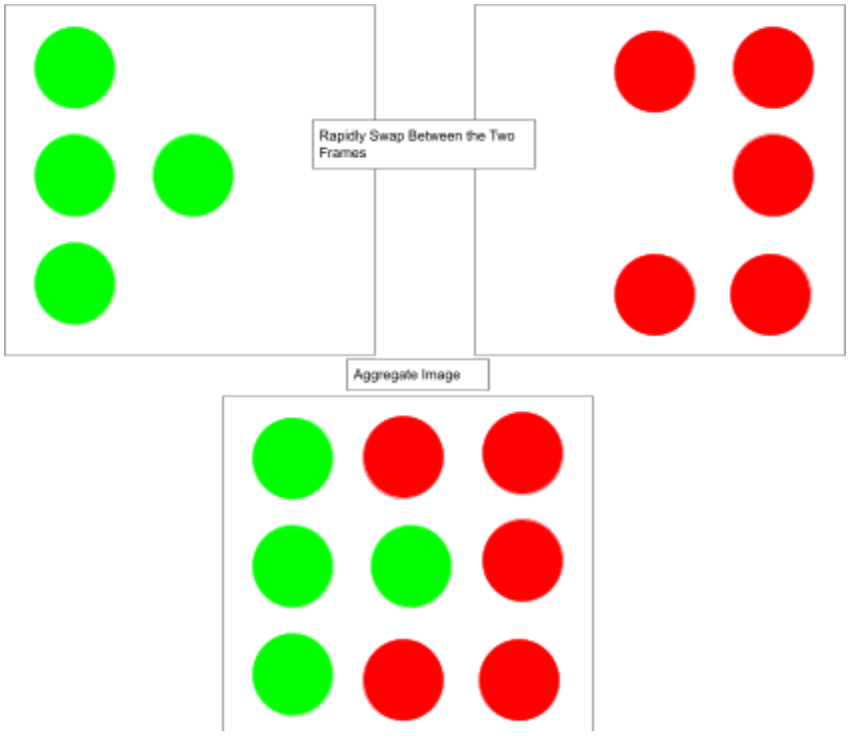
Software Design:

The TicTacToe board uses a number of the hardware modules of the MSP430 that interface with the peripheral hardware such that TicTacToe can be played. This section will describe the software used on the TicTacToe board.

LED Display

As described in the hardware section of this report, for all LEDs that are currently turned on, only a constant color can be displayed. However, a uniformly colored display is not much use for a game of TicTacToe, so we must used software in order to make the display appear to have multiple colors showing at once.

This is achieve by tracking the on/off state of the RGB leds in a bit vector associated with the LED, and sweeping across the colors at a speed high enough to allow the human persistence of vision to combine multiple frames in rapid succession as a single image. Using the TimerA1 timer modules, one at a time the RED, GREEN or BLUE pins are pulled low, and any of the LEDs that should be turned on at that time have their common anode IO pin pulled high. By changing across the colors in quick succession, a persistent image is achieved. An example of this is shown below.



To indicate the current position that the next marker will be placed, the associated LED will be made to flash. This is achieved using the TimerA0 timer module and the CCR0 interrupt service routine. At a fixed interval, the output of the associated position is binary XORd with itself to create the flashing effect.

Buttons

To interact with the players, buttons are used, as described in the hardware section. When a button is pressed, pulling the associated IO pin low, the high-to-low edge transition trigger the Port2 interrupt service routine to execute. The majority of the TicTacToe logic is contained within this interrupt service routine.

Firstly, when a button is pressed and the interrupt service routine begins, a small delay is inserted to allow for the unavoidable small delays that result from a human attempting to press two buttons simultaneously, as well as to account for the inherent bouncing that occurs when using electro-mechanical contactors. If the second button is then found to be pressed also, this indicates that the player has completed their go, and a marker of their color will be placed at the current marked position. If only one button is pressed, the current marked position will jump to the next available position or previous available position depending on which button is pressed.

Once a player's turn is complete, the current board arrangement is checked to see if there is a winning configuration. If the completed go has resulted in a win, then an animation will be played that displays the color of the winning player. If the end of the game is reached (all positions are filled) and no winning configuration has been created, a drawn-game animation will play that contains the colors of both of the players. Following either of these events, the game resets with a blank board immediately.