Project Description

- The project starts off with initializing the IT_Keyboard Interrupt
- Then continues down to initializing the LEDs ports for sending LED signals to the IO_LED
    - Also getting the correct beginning signal ready of all green LED's and moving to the right at base speed
- Then starts the main loop

- Check to see if any LEDs are on – if not wait for the start
- If they are, run through update_LEDs
    - Update_LEDs shifts the LEDs left or right depending on the Direction variable
        - If Direction is #$00 – which is the beginning case – go right
        - If Direction is #$01 – go left
- Then run through the Timer for LEDs
    - Initialize timer
    - Check variable delay to see if the button 1,2, or 3 was pressed
        - If 1 was pressed – set register a to equal NTIMES_x1 or known as base speed
        - If 2 was pressed -- set register a to equal NTIMES_x2 or known as base speed * 2
        - If 3 was pressed -- set register a to equal NTIMES_x3 or known as base speed * 3
    - Clear the TFLG2 bit and spin the timer
    - Continue to spin the timer until register a is equal to 0
    - After a equals 0 – exit the timer and continue back to main loop
- Check if any button was pressed
    - If yes -- Find which button it was and go to the subroutine linked to that number
        - If that number is not 0,1,2,3,4,5 do nothing
        - If 0
            - Go to start - This starts the movement of the LEDs
            - First make sure the LEDs are current not moving already
                - If they are not moving already – continue to start
                    - Set the LEDs to #$F0 and send to the IO_LED
                - If they are moving already – skip start and do nothing
                    - Continue back to main loop
        - If 1
            - Go to play – This changes the direction of the LEDs and the speed

        - Check to see if play was pressed at the time LEDs equal #$F0 and check is the Direction was going left
            - If yes – change the direction of the LEDs shift
                - Also change the speed to base speed * 1
            - If not #$F0 – skip to next test in play
            - If yes #$F0 but Direction is not going left – skip to next test in play
        - Check to see if play was pressed at the time LEDs equal #$0F and check is the Direction was going right
            - If yes – change the direction of the LEDs shift
                - Also change the speed to base speed * 1
            - If not #$0F – leave the play subroutine and go back to main loop
            - If yes #$0F but Direction is not going right – leave the play subroutine and go back to main loop
- If 2
    - Go to play – This changes the direction of the LEDs and the speed
        - Check to see if play was pressed at the time LEDs equal #$F0 and check is the Direction was going left
            - If yes – change the direction of the LEDs shift
                - Also change the speed to base speed * 2
            - If not #$F0 – skip to next test in play
            - If yes #$F0 but Direction is not going left – skip to next test in play
        - Check to see if play was pressed at the time LEDs equal #$0F and check is the Direction was going right
            - If yes – change the direction of the LEDs shift
                - Also change the speed to base speed * 2
            - If not #$0F – leave the play subroutine and go back to main loop
            - If yes #$0F but Direction is not going right – leave the play subroutine and go back to main loop
- If 3
    - Go to play – This changes the direction of the LEDs and the speed
        - Check to see if play was pressed at the time LEDs equal #$F0 and check is the Direction was going left
            - If yes – change the direction of the LEDs shift
                - Also change the speed to base speed * 3
            - If not #$F0 – skip to next test in play

                - If yes #$F0 but Direction is not going left – skip to next test in play
            - Check to see if play was pressed at the time LEDs equal #$0F and check is the Direction was going right
                - If yes – change the direction of the LEDs shift
                    - Also change the speed to base speed * 3
                - If not #$0F – leave the play subroutine and go back to main loop
                - If yes #$0F but Direction is not going right – leave the play subroutine and go back to main loop
        - If 4
            - Go to Stop – This iterates through an infinite loop until
                - Check the Key_value
                    - If the Key_value equals #$06 (The Resume Button) then leave the loop and go right back to the main loop
                    - If they Key_value equals anything else – continue to loop through the infinite loop until Key_value equals #$06
        - If 5
            - Go to Reset – this completely resets the program to its beginning values
                - Set LEDs back to #$00
                - Set Direction back to the right
                - Set Delay back to base speed
                - After setting everything back to start values, exit and go back to main loop
        - If any other number on the IT_Keyboard
            - DO NOTHING
    - If no button was pressed
        - Continue to do the same thing as the previous iteration

- Clear Pressed
- Run through the IT_delay
    - Set value to a and x registers and after each iteration of the loop, decrement the two registers until they are both equal to 0
        - If both registers are equal to 0 then exit the delay and continue back to the main loop

        - If one or both registers are not equal to 0 then continue to run through the delay loop

- Main interrupt for reading IT_Keyboard
    - Load address of masks and ground each column
    - Read data from Row port and see if all are high
        - If not go to next row
        - If yes – begin to check all columns
            - If found a column and row that are set high – compare the two to find the number being pressed
                - Store that Number into Key_value
                - Set Flag and exit the interrupt
            - If could not find a Column set high, a row set high, or neither – then no button was pressed
                - Set Flag and exit the interrupt