T.C.
BAHÇEŞEHİR UNIVERSITY

FACULTY OF ENGINEERING AND NATURAL SCIENCES

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

PROJECT REPORT

MEMORY GAME

EEE3205 –Microcontrollers
Course Project

Enes Olgun - 1902653

Hande Coşkun - 1904326

TABLE OF CONTENTS

1 OVERVIEW

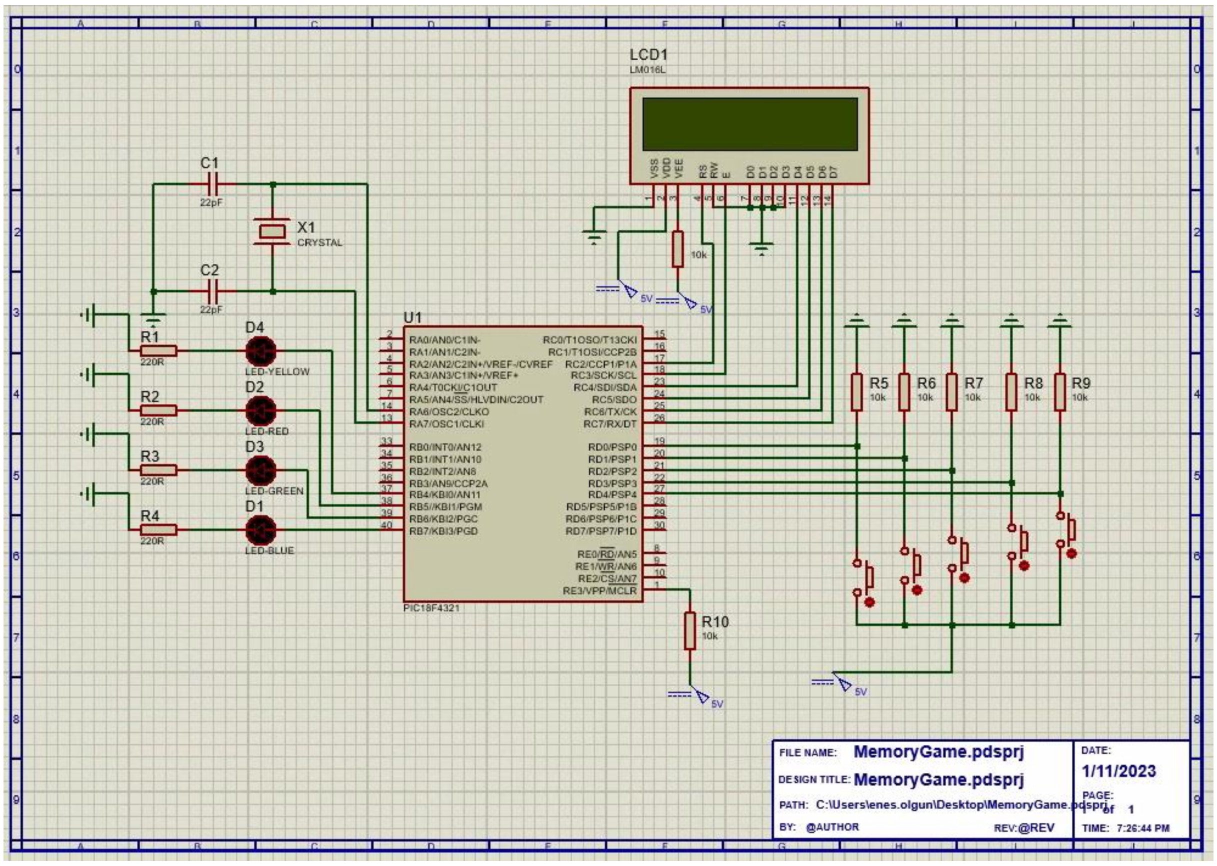
1.1 Problem Statement and Objectives

The Project is a simple game that tests the short-term memory of a human. After the button is pressed, game starts and four different colored LED lights on randomly. Player's goal is to remember the order of the LEDs that light on.

Our objective in this Project is to exercise our brain. However, the main goal is simply to be able to make a game.

1.2 Background Information

This product, as it is understood from the name of the Project, is mainly about memory. There are lots of applications that are beneficial in terms of memory, such as crosswords, puzzle, chess and sudoku. A study [1] showed that frequent participation in activities, specifically those related to playing games and puzzles, is beneficial to brain health among middle-aged adults at increased risk for Alzheimer's disease which is a well-known and frequently encountered disease. Our memory game can contribute to the development of visual short-term memory. In this way, it has the same function as puzzles.



FILE NAME: MemoryGame.pdsprj
DESIGN TITLE: MemoryGame.pdsprj
PATH: C:\Users\enes.olgun\Desktop\MemoryGame.pdsprj
BY: @AUTHOR
REV:@REV
DATE: 1/11/2023
PAGE: 1 of 1
TIME: 7:26:44 PM

Figure 1. Circuit schematic in Proteus

2 METHODOLOGY

2.1 Project Design

In the Project, PIC18F4321 microcontroller chip is used. The chip is powered up with 5V through RE3/VPP/MCLR pin. Crystal oscillator is connected to RA6/OSC2 pin and RA7/OSC1 pin with two external 22pF load capacitor. Five buttons are used and they are connected to RB0, RB1, RB2, RB3 and RB4 pins with 10k ohm pull-down resistors. Input signal is given to microcontroller using buttons.

LCD connections:

Vss, RW, D0, D1, D2, D3→ grounded (0V)

Vdd → 5V

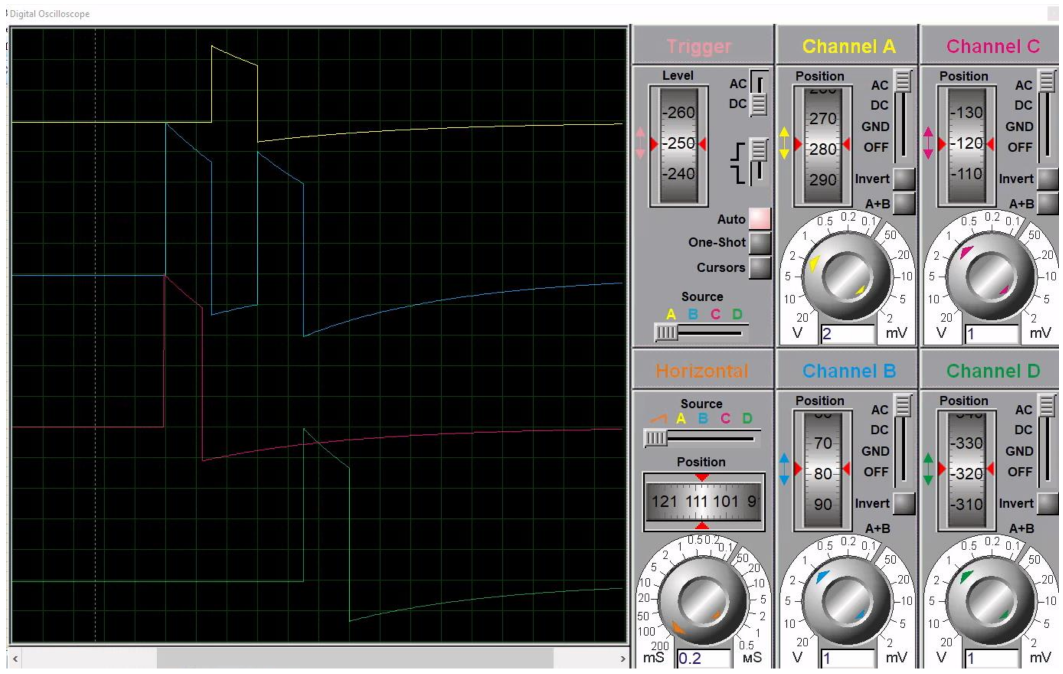
Vee → 5V through the 10k ohm resistor

RS, E, D4, D5, D6, D7 → RC2, RC3, RC4, RC5, RC6, RC7 respectively

Yellow LED, red LED, green LED, blue LED are connected to RB4, RB5, RB6, RB7 pins respectively. LEDs are grounded through the 220 ohm resistors. LEDs are the output of the circuit.

Circuit works as follows:

Until the button connected to RD0 pin is pressed, pin's logic level is 0. After pressed, logic level changes to logic 1 state and this triggers the output signal. LEDs will be on and off. Then, user will give the input signal by pressing buttons and again, this will trigger the output signal.



2.2 Project components

Proteus and mikroC for PRO programs are used in this Project for design procedure and simulation. Design part of the Project is done by using Proteus program.

1- Resistors are used to limit the current.
2- Crystal oscillator is used to provide stable output.
3- Capacitors are used with crystal oscillator to make them resonate in their natural parallel mode.
4- LCD is used to provide instructions for the player and feedback on winning. LCD's main function is to display intended output.
5- Buttons are used to control the game and provide input data to circuit by pressing the button. Button's main function is to provide input to an application.
6- LEDs are used to indicate the outputs and provide inputs indirectly (e.g. if red LED is on, player should press the button that lights on the red LED). [2] LED's main function is to deliver efficient light generation with little-wasted electricity.

2.3 Final product and results

When program is runned, LCD will be on. " WELCOME TO" will be displayed on the first row of the LCD and " MEMORY GAME!" will be displayed on the second row of the LCD. After 2 second delay, the screen of the LCD will be cleared and "WATCH THE LEDS" will be displayed on the first row, " CAREFULLY!" will be displayed on the second row. After 2 seconds delay " GOOD LUCK" will appear on LCD screen and after another 2 seconds LCD screen will be cleared.

A button is connected to RD0 pin as input. Until this button is pressed, there will be no action. All LEDs are configured as output. Yellow LED is connected to RB4 pin, red LED is connected to RB5 pin, green LED is connected to RB6 pin and blue LED is connected to RB7 pin.

Once the button that connected to RD0 pin is pressed, 10 ms (10 ms for checking the status of the button) after, the game will start and four LED will light on for 500 ms randomly then will be off. In total, six LED will be on and off. After all LEDs off, "TRY TO GUESS" will appear on the first row and "THE ORDER!" will appear on the second row of the LCD for 1 second.

Buttons connected to PORTD configured as input. To light on :

Yellow LED → button connected to RD4 pin must be pressed

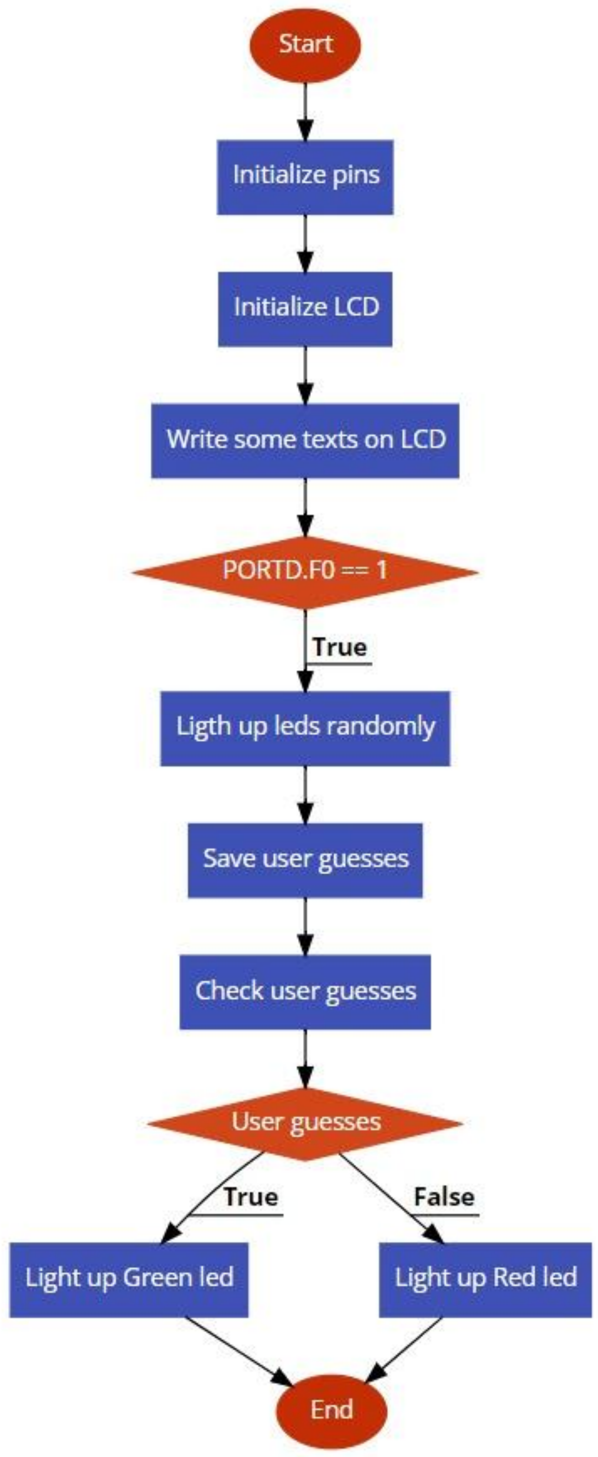
Red LED → button connected to RD3 pin must be pressed

Green LED → button connected to RD2 pin must be pressed

Blue LED → button connected to RD1 pin must be pressed

Player must use these buttons to guess the order. When each button is pressed, LED will be on for 700 ms. Six button presses will be expected to continue.

Finally, if player pressed the correct LEDs in correct order, green LED will be on and "WELL DONE!" will appear on the LCD screen. If player guessed wrong, red LED will be on and "GAME OVER!!" will appear on the LCD screen.



2.4 Code

```
#include <stdlib.h> //  inserting header file
#include <stdio.h>  // inserting header file
// LCD connections
sbit LCD_RS at RC2_bit;
sbit LCD_EN at RC3_bit;
sbit LCD_D4 at RC4_bit;
sbit LCD_D5 at RC5_bit;
sbit LCD_D6 at RC6_bit;
sbit LCD_D7 at RC7_bit;

sbit LCD_RS_Direction at TRISC2_bit;
sbit LCD_EN_Direction at TRISC3_bit;
sbit LCD_D4_Direction at TRISC4_bit;
sbit LCD_D5_Direction at TRISC5_bit;
sbit LCD_D6_Direction at TRISC6_bit;
sbit LCD_D7_Direction at TRISC7_bit;

unsigned int led_order[6];   // unsigned integer array to keep the values of LED’s order
int i;
unsigned int guess[6];  // unsigned integer array to keep the values of user guess

// Function that lights up random LED
void light_up_random_led(unsigned int *led_order)
{
   int i;    // Define counter for number of LEDs that light
   for(i=0;i<6;i++){
   unsigned int p = rand(); // Generate a random number
   p = p % 4 + 1;
   led_order[i] = p;  // array to keep random numbers
   switch(p)   // switch function to select to be lighted on
   {
      case 1:

         PORTB.F4 = 1; // Light up LED1
         delay_ms(500); // Wait for half a second
         PORTB.F4 = 0; // Turn off LED1
         break;
      case 2:

         PORTB.F5= 1; // Light up LED2
         delay_ms(500); // Wait for half a second
         PORTB.F5 = 0; // Turn off LED2
         break;
      case 3:

         PORTB.F6 = 1; // Light up LED3
         delay_ms(500); // Wait for half a second
         PORTB.F6 = 0; // Turn off LED3
         break;
      case 4:

         PORTB.F7 = 1; // Light up LED4
         delay_ms(500); // Wait for half a second
         PORTB.F7 = 0; // Turn off LED4
         break;
   }
   }
}
```

```
void save_guess(unsigned int *guess, int i)  // Function that saves user's guesses
{
   for(i = 0; i < 6; i++){   // For loop for six user's presses
      while(1){   //  Infinite loop
         if(PORTD.F4==1){ //Yellow button pressed
            guess[i]=1;
            PORTB.F4 = 1; // Light up LED1
            delay_ms(700); // Wait for 700 ms
            PORTB.F4 = 0; // Turn off LED1
            break;
         }
         if(PORTD.F3==1){ //Red button pressed
            guess[i]=2;
            PORTB.F5= 1; // Light up LED2
            delay_ms(700); // Wait for 700 ms
            PORTB.F5 = 0; // Turn off LED2
            break;
         }
         if(PORTD.F2==1){ //Green button pressed
            guess[i]=3;
            PORTB.F6 = 1; // Light up LED3
            delay_ms(700); // Wait for 700 ms
            PORTB.F6 = 0; // Turn off LED3
            break;
         }
         if(PORTD.F1==1){ //Blue button pressed
            guess[i]=4;
            PORTB.F7 = 1; // Light up LED4
            delay_ms(700); // Wait for 700ms
            PORTB.F7 = 0; // Turn off LED4
            break;
         }
      }
   }

}

// Function to compare guess and  led_order arrays
void check_guess(unsigned int *led_order, unsigned int *guess)

{
   int i;
   for(i=0;i<6;i++){
      if(led_order[i] != guess[i])  // Wrong guess
      {
         PORTB.F5 = 1; // Light up red LED (indicating incorrect guess)
         Lcd_Cmd(_LCD_CLEAR); // Clear the screen initially
         Lcd_Out(1, 4, "GAME OVER!!"); // Print the text
         return;
      }
   }
```

```
    PORTB.F6 = 1; // Light up green LED (indicating correct guess)
    Lcd_Cmd(_LCD_CLEAR); // Clear the screen initially
    Lcd_Out(1, 4, "WELL DONE!"); // Print the text
}



void main() {
    TRISB = 0; // Set all PORTB pins as outputs
    TRISD = 1; // Set all PORTD pins as inputs
    PORTD.F0 = 1; //RD0 is input
    Lcd_Init(); // Initialize LCD module
    Lcd_Cmd(_LCD_CLEAR); // Clear the screen initially
    Lcd_Out(1, 4, "WELCOME TO"); // Print the text
    Lcd_Out(2, 3, "MEMORY GAME!"); // Print the text
    delay_ms(2000);
    Lcd_Cmd(_LCD_CLEAR);  // Clear the screen
    Lcd_Out(1, 2, "WATCH THE LEDS");  // Print the text
    Lcd_Out(2, 4, "CAREFULLY!");  // Print the text
    delay_ms(2000);
    Lcd_Cmd(_LCD_CLEAR);    // Clear the screen
    Lcd_Out(1, 4, "GOOD LUCK!"); // Print the text
    delay_ms(2000);
    Lcd_Cmd(_LCD_CLEAR);  // Clear the screen


    srand(10);
    while(1) {
        if(PORTD.F0 == 1){ // Wait for button press
            delay_ms(10); // debouncing
                light_up_random_led(led_order);
                Lcd_Out(1, 3, "TRY TO GUESS");   // Print the text
                Lcd_Out(2, 4, "THE ORDER!");  // Print the text
                delay_ms(1000);
                save_guess(guess,i);// Calling the function
                check_guess(led_order,guess);        // Calling the function

}
}
}
```

3 CONCLUSION

In conclusion, the code and circuit designed in Proteus for the LED guessing game is a functional and interactive system that allows for a challenging and entertaining user experience. The use of microcontroller, LEDs, and buttons in the circuit, in conjunction with the well-written code, allows for the random generation of a sequence of LED lights, the ability for the user to make a guess by pressing buttons corresponding to the colors of the LEDs, and the comparison of the user's guess to the correct sequence. Overall, the project is a successful demonstration of the capabilities of microcontroller-based systems in creating interactive and engaging applications.

4 REFERENCES

1- [1] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4417099/

2- [2] https://www.rohm.com/electronics-basics/leds/what-are-leds#:~:text=Compared%20with%20conventional%20light%20sources,generation%20with%20little%2Dwasted%20electricity.