

Color Pursuit

GROUP - 6

CSE-101 Project

Group Members

- Muhammed Bilal Türk
- Burak Kuruçay
- Hakan Ata Yılmaz
- Ömer Faruk Gürsel
- Bedri Kutay Karaman
- Sedef Köroğlu
- Barış Eren Gezici
- Muhammed Şerif Naşit
- Arda Yasan

Contents

What is our project? _____	1
Circuit and Parts _____	Error! Bookmark not defined.
Code and Algorithm _____	4
Problems and Solutions _____	10
Group Members and Task Distribution _____	11

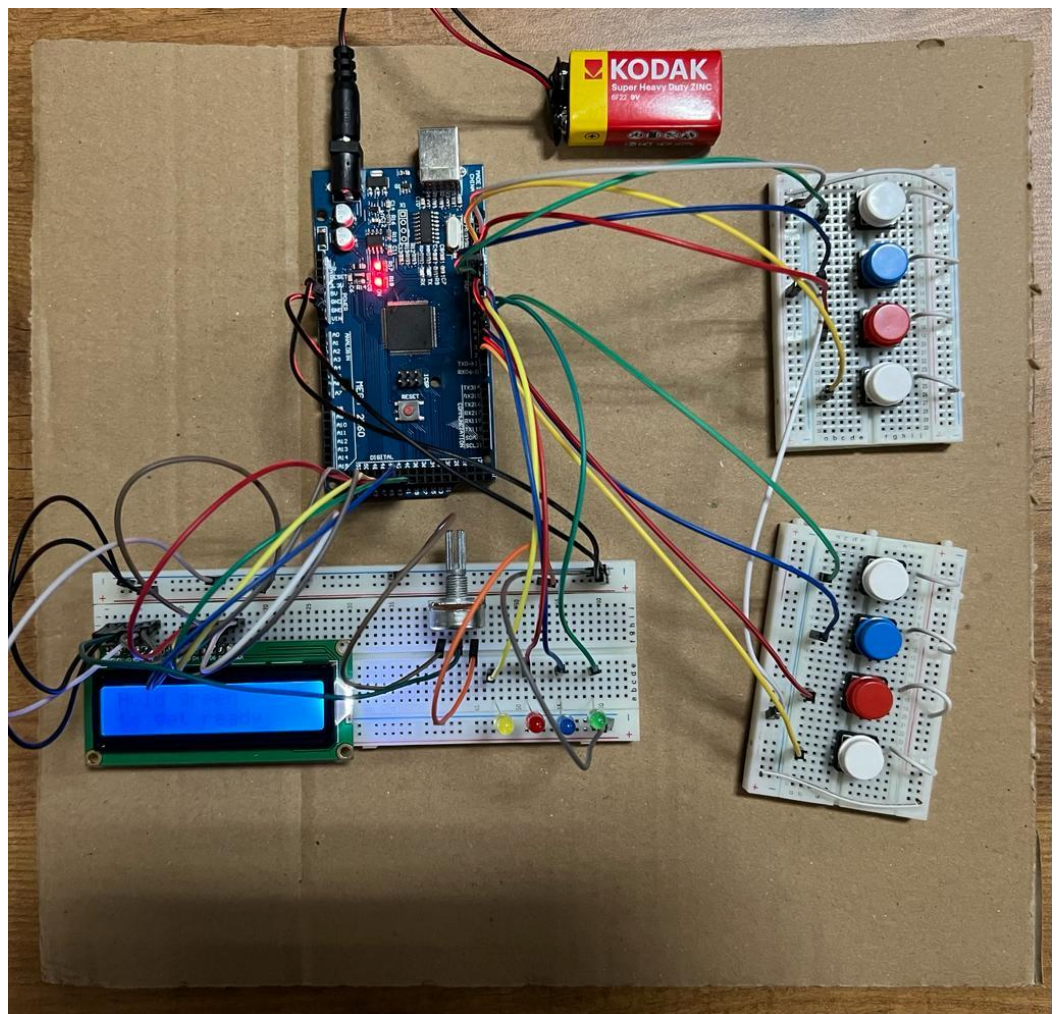
What is our Project?

What we did

We designed color puzzle memorizing game called “Color Pursuit” by using Arduino environment.

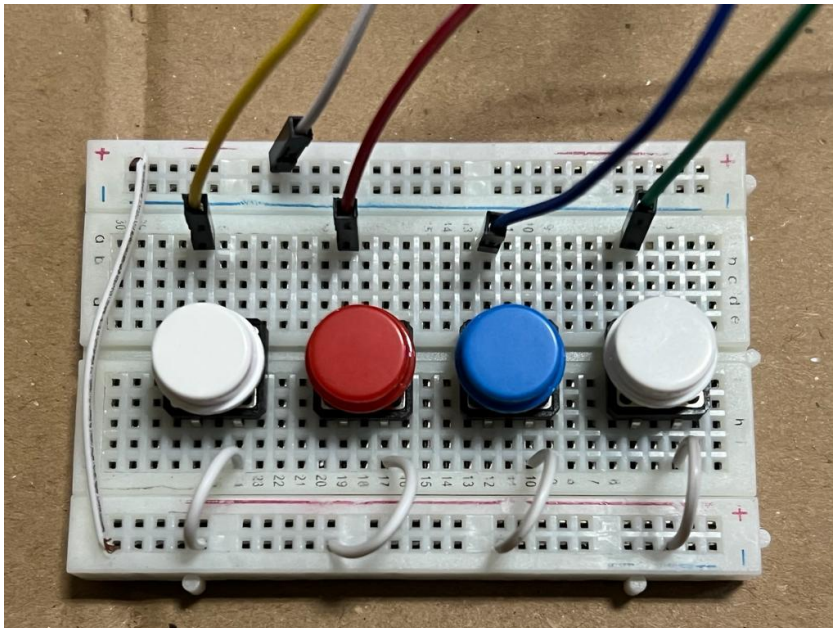
Game: Colour Pursuit

The game depends on generating random puzzles and receiving answers from the players.

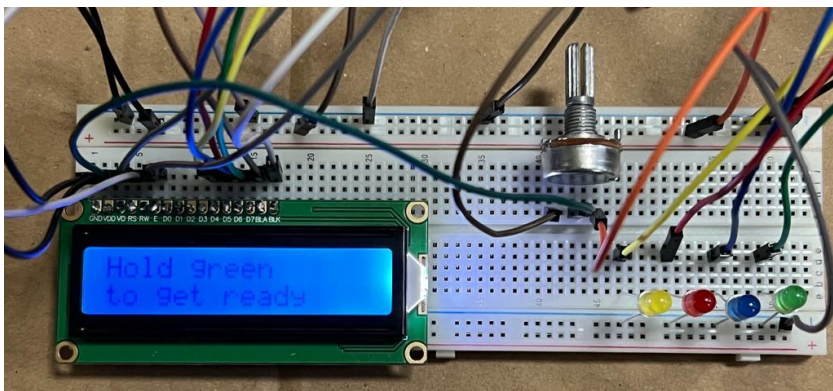


Circuit and Parts

Controllers: Each controller have 4 push buttons for players to give colour inputs.



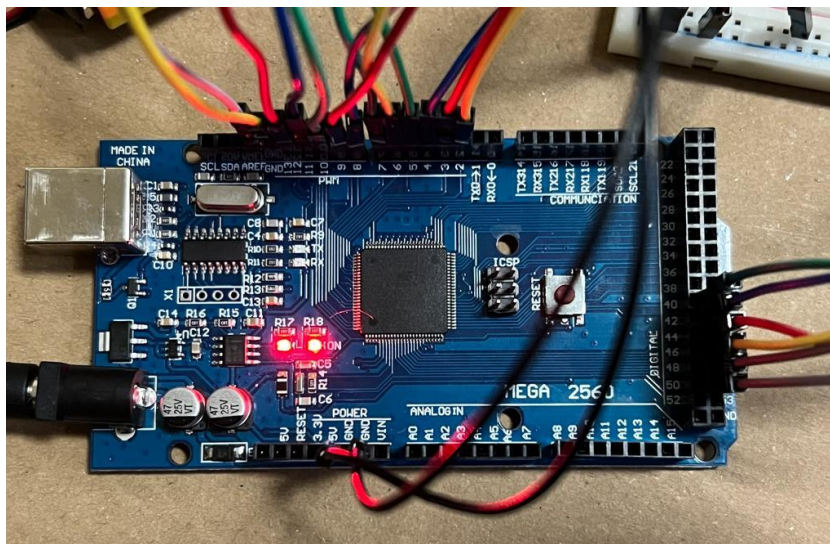
Main Breadboard: It provides communication between Arduino, controllers and LCD screen. There are 4 led and LCD screen on the main breadboard.



LCD Screen: LCD screen is the interface that displays the informations about players, countdowns, scores and winner.



Arduino Mega: It includes the code of the game and it provides electricity to circuit.



Code and Algorithm

Void Loop ()

Main function of the algorithm is “loop” function. Loop function includes all steps of the game and algorithm.

```
void loop() {
    while (digitalRead(G1) != LOW || digitalRead(G2) != LOW) // Both players must press to green button
    {
        //at the same time to starting game.
    }

    for (int level=0 ; 5>level ; level++)
    {
        lcd_print2(); // Prints get ready to LCD Screen.
        delay(5000);
        int len = 3+1level; // Lenght of the LED Sequence.
        charpuzzle = rndm(len); // Generates random LED sequences of lenght len.
        lcd_clear(); // Resets LCD Screen.
        turn_lamb(puzzle,len,(level+1)*100); // Turns on LED according to generated puzzle.
        for(int j = 0; j < 2; j++){ // j = 0 for player 1, j = 1 for player 2.
            unsigned long time = millis();
            delay(1000);
            lcd_print3(j+1);
            lcd.setCursor(0,1);
            lcd.print("Time left: ");
            lcd.setCursor(13,1);
            while ((millis()-time) <= 10000)
            {
                {
                    check1(); // Takes input from player 1.
                    lcd.setCursor(13,1);
                    lcd.print(10-((millis()-time)/1000));
                    if (j != 0)
                    {
                        check2(); // Takes input from player 2.
                        lcd.setCursor(13,1);
                        lcd.print(10-((millis()-time)/1000));
                    }
                }
            }
        }
        lcd_print4(puzzle);
        comparer(puzzle,len); // Checks the given input wheter true or not, and increases to scoreboard.
        str1_len = 0;
        str2_len = 0;
        free(puzzle);
    }
    finish(); // Prints who won to the LED Screen.
}
```

Steps of the Void Loop

Step 1: There is a while loop at the start of the void loop. This while loop checks inputs from green buttons. If both green button is pressed while breaks and game starts.

Step 2: Starts for loop after the while loop. This for loop turns 5 times because there are 5 levels at the game.

Step 3: After the for loop starts Arduino prints “Get Ready” 5 seconds at the LCD screen and Arduino generates random color sequence by “rndm ()” function.

rndm() Function: Generates sequences.

```
char * rndm(int len){
    char *str = (char *)malloc(len+1);
    randomSeed(analogRead(A3));
    for (int i=0 ; len>i ; i++)
    {
        int num = random(0,4);
        if (num == 0)
        {
            str[i] = 'R';
        }
        else if(num == 1)
        {
            str[i] = 'B';
        }
        else if(num == 2)
        {
            str[i] = 'G';
        }
        else str[i] = 'Y';
    }
    str[len] = '\0';
    return str;
}
```

Step 4: LEDs blinks in generated order by “turn_lamb ()” function.

turn_lamb () Function: Blinks LEDs in order.

```
void turn_lamb(char *puzzle,int len,int x)
{
    //ledleri yakan kod
    for (int i=0;i<len;i++){
        if (puzzle[i]=='G' ){
            digitalWrite(green_lamb,HIGH);
            delay(500);
            digitalWrite(green_lamb,LOW);
            delay(500);
        }
        else if (puzzle[i]=='B' ){
            digitalWrite(blue_lamb,HIGH);
            delay(500);
            digitalWrite(blue_lamb,LOW);
            delay(500);
        }
        else if (puzzle[i]=='R' ){
            digitalWrite(red_lamb,HIGH);
            delay(500);
            digitalWrite(red_lamb,LOW);
            delay(500);
        }
        else
        {
            digitalWrite(yellow_lamb,HIGH);
            delay(500);
            digitalWrite(yellow_lamb,LOW);
            delay(500);
        }
    }
}
```


Step 5: Second for loop starts for take the inputs of the Player 1 and Player 2 separately. This for loop includes “check1()” and “check2 ()” functions to take the inputs.

check1 () function: Takes inputs from Player 1.

```
void check1()
{
    if(digitalRead(R1) == LOW)
    {
        str1[str1_len] = 'R';
        str1_len++;
        delay(300);
    }
    else if(digitalRead(B1) == LOW)
    {
        str1[str1_len] = 'B';
        str1_len++;
        delay(300);
    }
    else if(digitalRead(Y1) == LOW)
    {
        str1[str1_len] = 'Y';
        str1_len++;
        delay(300);
    }
    else if(digitalRead(G1) == LOW)
    {
        str1[str1_len] = 'G';
        str1_len++;
        delay(300);
    }
}
```

Step 6: “comparer()” function works and compares inputs and generated sequence, If input and generated sequence same it adds point to players.

comparer() function: Compares inputs and gives points.

```
void comparer (char *puzzle,int len){
    if (str1_len == len)
    {
        int flag = 1;
        for (int i=0 ; len>i ; i++)
        {
            if (puzzle[i] != str1[i])
            {
                flag = 0;
                break;
            }
        }
        if (flag) score1++;
    }
    if (str2_len == len)
    {
        int flag = 1;
        for (int i=0 ; len>i ; i++)
        {
            if (puzzle[i] != str2[i])
            {
                flag = 0;
                break;
            }
        }
        if (flag) score2++;
    }
    lcd_print();
    delay(5000);
}
```

- Second, third, fourth, fifth and sixth steps repeats five times for 5 levels under favour of the first for loop.

Step 7: First for loop ends, winner determines and prints to LCD by the “finish ()” function.

finish() function:

```
void finish()
{
    lcd_clear();
    lcd.setCursor(0,0);
    if (score1 == score2)
    {
        lcd.print("Draw.");
    }
    else if (score1 > score2)
    {
        lcd.print("Player1 Won!!!!");
    }
    else
    {
        lcd.print("Player2 Won!!!!");
    }
}
```

Problems and Solutions

We had some problems while doing our project.

Problem: In the beginning, we designed a game where two players can send input at the same time but our code didn't work.

Solution: We changed our game so that two players can send inputs in turn.

Problem: LCD screen did not show what we sent.

Solution: We added a resistor to the circuit to adjust the contrast of the LCD.

Problem: The game was getting extra input from players,

Solution: We realized that this problem was caused by the short delay time. We increased the waiting time.

Group Members and Task Distribution

1- Muhammed Şerif Naşit – 210104004052

Tasks: Presentation, circuit building and debugging.

2- Hakan Ata Yılmaz – 210104004048

Tasks: Coding, algorithm, circuit building and debugging.

3- Arda Yasan – 210104004043

Tasks: Presentation, coding and debugging.

4- Barış Eren Gezici – 210104004041

Tasks: Coding, algorithm, coordinatorship and debugging.

5- Ömer Faruk Gürsel – 210104004053

Tasks: Preparing demo, circuit building and debugging.

6- Bedri Kutay Karaman – 210104004045

Tasks: Circuit building, preparing demo and debugging.

7- Sedef Köroğlu – 210104004051

Tasks: Coding, circuit building and debugging.

8- Muhammed Bilal Türk – 210104004047

Tasks: Circuit building, coding, algorithm and debugging.

9- Burak Kuruçay – 210104004049

Tasks: Circuit building, debugging and preparing report.

Bahadır Dikici – 210104004054 (He did not participate in group.)