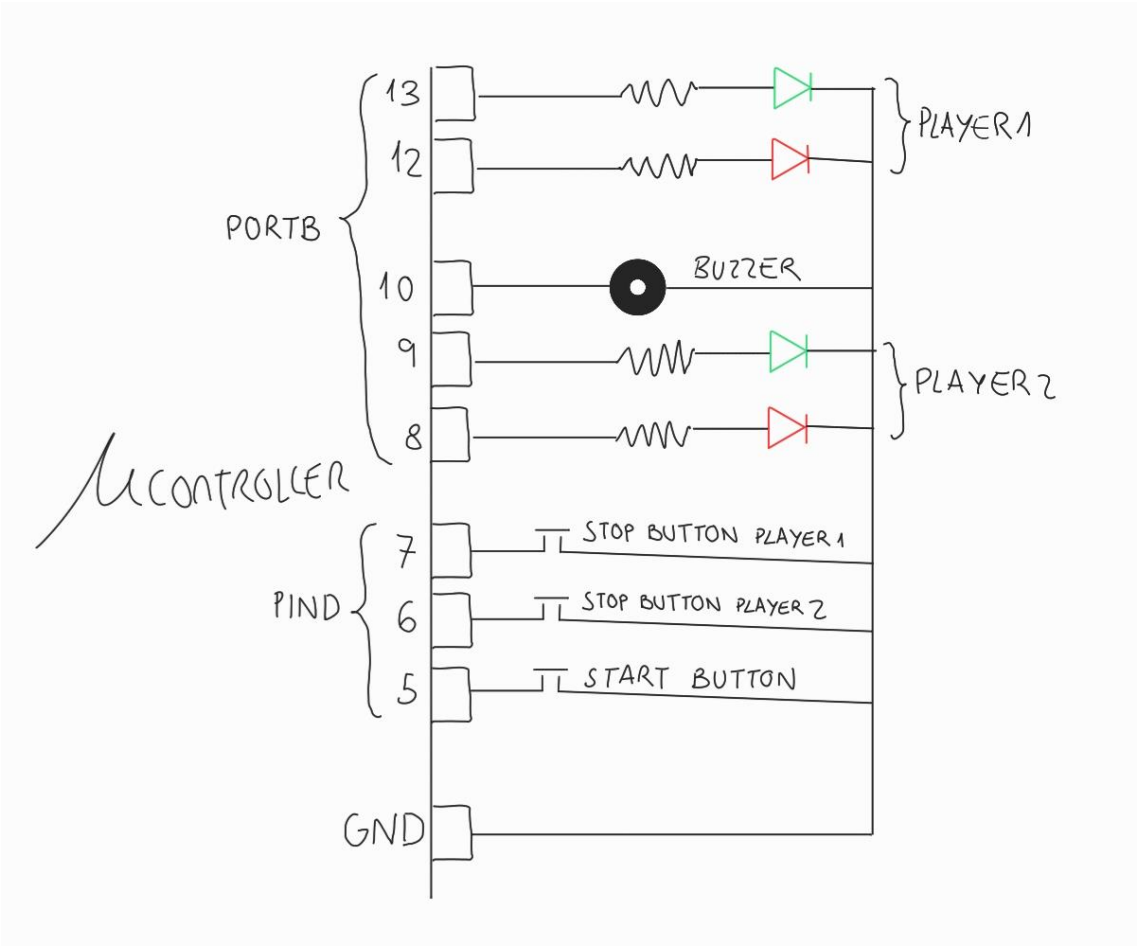
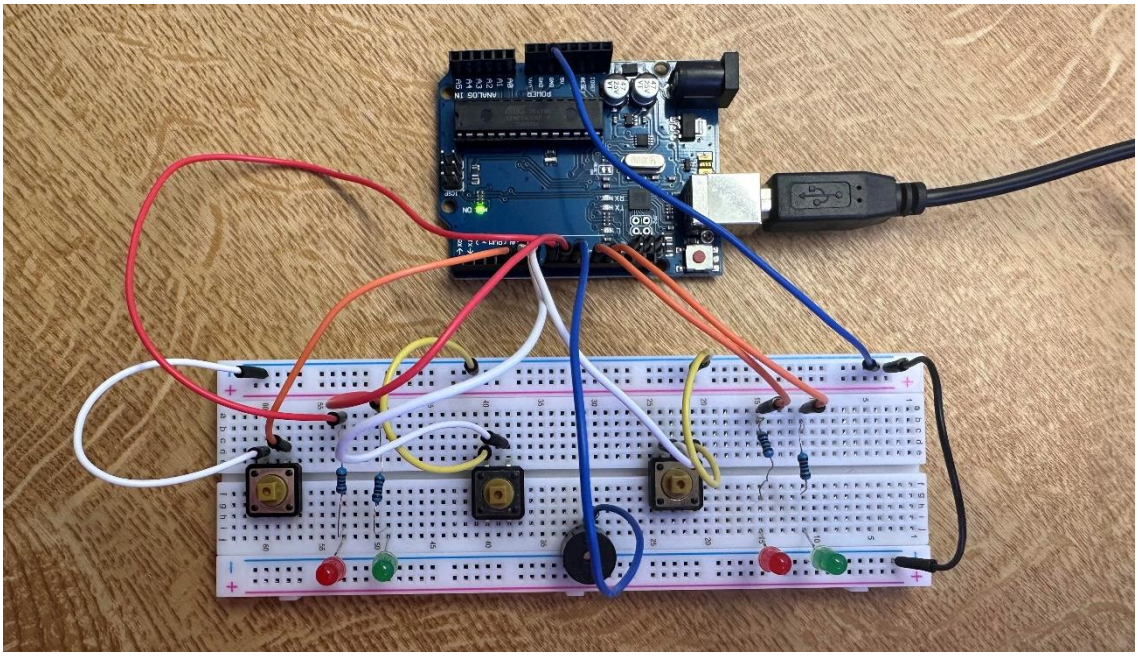
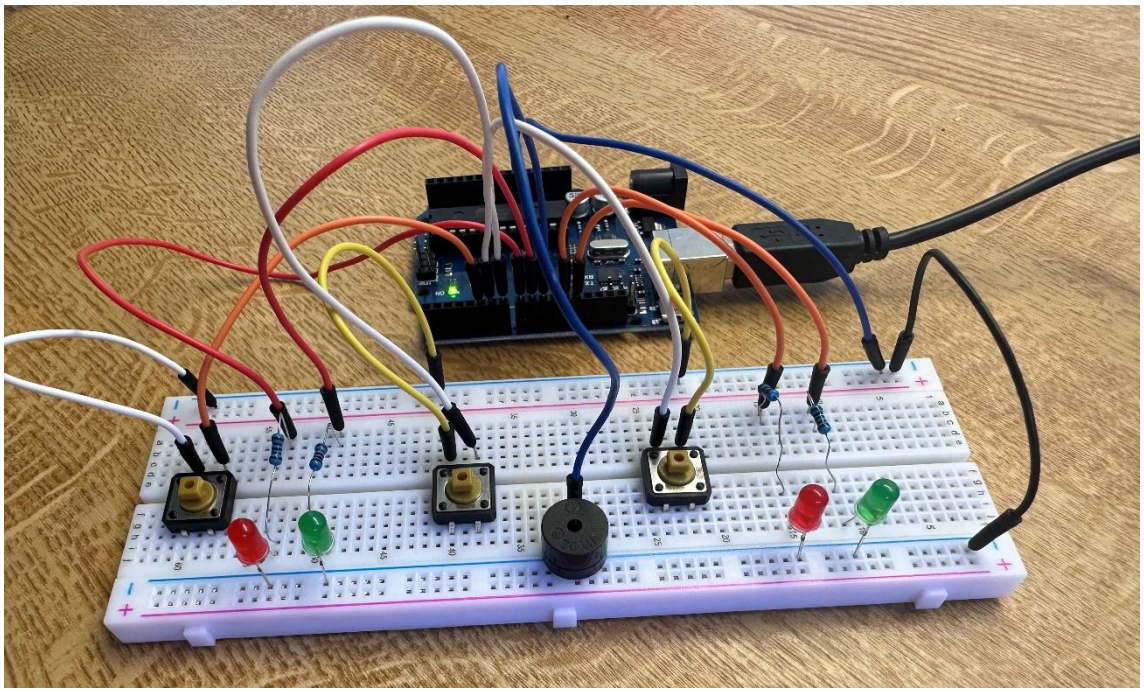
CIRCUIT DIAGRAM



PHOTOS





SOURCE CODE

```
#define GAMES 3

void setup() {
  // Set pin configurations
  DDRB = B00110111; // Configure PORTB pins for output
  DDRD = B00000000; // Configure PORTD pins for input
  PORTD = B11111111; // Enable pull-up resistors on PORTD pins
  Serial.begin(9600); // Initialize the serial communication
  randomSeed(analogRead(0)); // Seed the random number generator
}

void loop() {
  byte res;
  int delayTime, i, t, player1 = 0, player2 = 0;
  int player1Time[GAMES], player2Time[GAMES];

  /* START OF ROUND  */
  Serial.println("PRESS START");
  // Wait for the "Start" button press
  while((PIND & 0x20) != 0x00);
  Serial.println("THE ROUND BEGINS");

  /* GAMES */
  for(i = 0; i < GAMES; i++) {
    Serial.println("\nGAME " + String(i + 1));
```

```
    // Generate a random delay between 1 and 10 seconds
    delayTime = random(1000, 10000);
    delay(delayTime);

    if(((res = PIND) & 0xC0) != 0xC0) { // Check if a player pressed
before the start of the game
      if((res & 0xC0) == 0x40) {
        PORTB = B00010010;
        Serial.println("Player 1 pressed too early.");
        player2++;
        player1Time[i] = 0;
        player2Time[i] = 0;
      } else if((res & 0xC0) == 0x80) {
        PORTB = B00100001;
        Serial.println("Player 2 pressed too early.");
        player1++;
        player1Time[i] = 0;
        player2Time[i] = 0;
      } else if((res & 0xC0) == 0x00) {
        PORTB = B00010001;
        Serial.println("Both players pressed too early.");
        player1Time[i] = 0;
        player2Time[i] = 0;
      }
    } else {
      PORTB = 0x04; // Turn on the buzzer
      t = millis();

      // Wait for a player to press their button
      while(((res = PIND) & 0xC0) == 0xC0);

      if((res & 0xC0) == 0x40) {
        player1Time[i] = millis() - t;

        // Wait for the other player to press their button
        while((PIND & 0x40) != 0x00);

        player2Time[i] = millis() - t;
        PORTB = B00100001;
        Serial.println("Player 1 wins the game " + String(i+1) + ".");
        player1++;
      } else if((res & 0xC0) == 0x80) {
        player2Time[i] = millis() - t;

        // Wait for the other player to press their button
        while((PIND & 0x80) != 0x00);

        player1Time[i] = millis() - t;
```

```
        PORTB = B00010010;
        Serial.println("Player 2 wins the game " + String(i+1) + ".");
        player2++;
      } else if((res & 0xC0) == 0x00) {
        player1Time[i] = millis() - t;
        player2Time[i] = player1Time[i];
        PORTB = B00100010;
        Serial.println("Both players win the game " + String(i+1) + ".");
        player1++;
        player2++;
      }
    }
  }

  /* PRINT RESULTS */
  if(player1 > player2) {
    Serial.println("\nWinner is Player 1:");
  } else if(player1 < player2) {
    Serial.println("\nWinner is Player 2:");
  } else {
    Serial.println("Draw:");
  }

  // Print game results and time differences
  for(i = 0; i < GAMES; i++) {
    if(player1Time[i] > player2Time[i]) {
      Serial.println("- Game " + String(i+1) + "(" +
String(player1Time[i]) + "ms), Difference (" + String(player1Time[i]-
player2Time[i]) + "ms).");
    } else {
      Serial.println("- Game " + String(i+1) + "(" +
String(player2Time[i]) + "ms), Difference (" + String(player1Time[i]-
player2Time[i]) + "ms).");
    }
  }

  Serial.println("\n");
  delay(1500);
  PORTB = 0x00; // Turn off all outputs
}
```

NOTES AND OBSERVATIONS

The code seems to be working correctly; I haven't encountered significant difficulties while developing it. The main function is too large, but it seemed like a bad idea to split the code into functions since all the code works on the ports and the 4 same variables.