GROUP 13

PROJECT: A REACTION TIME TESTER

GROUP MEMBERS

TIBOORE SAMPSON KAARA-YIRI -3559718

FORSON STEPHEN EKOW FOSU- 3000820

HONU SERAPHINE ELORM- 3002320

ELISHA ATAKORA-7125621

DON JULIAN YARTEY- 3014720

BOADU-AYEBOAFO KWADWO-7127221

TABLE OF CONTENTS

PROBLEM STATEMENT

In various fields such as sports, medicine, and human-computer interaction, reaction time is a crucial parameter for performance evaluation. Traditional reaction time tests often require expensive equipment or specialized software. This project aims to develop a cost-effective and interactive reaction time tester using an Arduino, LEDs, push buttons, and an LCD display.

- Randomly turn on an LED.
- Measure how quickly a player presses a button in response.
- Display the reaction time (in milliseconds) on an LCD.

METHODOLOGY

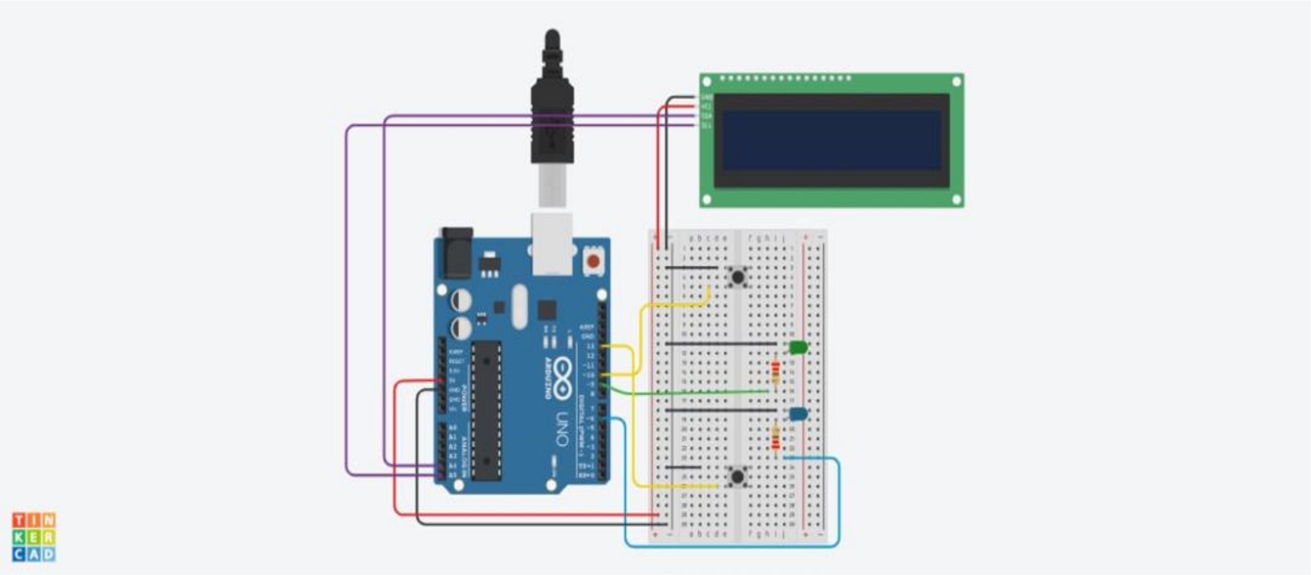
HARDWARE COMPONENTS

- Arduino Uno
- LCD Display (I2C Interface)
- LEDs (Blue, Green)
- Push Buttons (Connected via pull-up resistors)
- Resistors and connecting wires
- Breadboard

CIRCUIT DESIGN

- Two LEDs are connected to digital pins (D6, D9) to indicate the reaction prompt.
- Two push buttons are connected to digital pins (D13, D10) with internal pull-up resistors enabled.
- The LCD (16x2) is connected via the I2C interface at address 0x20.

CIRCUIT DIAGRAM



WORKING PRINCIPLE

1. The system initializes, displaying “Reaction Timer” on the LCD.

2. A random delay (2 to 5 seconds) occurs before one of the LEDs lights up.

3. The user must press the corresponding button as quickly as possible.

4. The Arduino calculates the reaction time based on the delay between the LED turning on and the button press.

5. The reaction time is displayed on the LCD.

6. After a short delay, the game resets for another round.

CODE

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Initialize LCD with I2C address 0x20 (16x2)
LiquidCrystal_I2C lcd(0x20, 16, 2);

// Define LED and Button Pins
const int ledPins[2] = {6, 9};     // Blue (D6), Green (D9)
const int buttonPins[2] = {13, 10}; // Blue Button (D13), Green Button (D10)

int chosenLED;
unsigned long startTime, reactionTime;
bool waitingForReaction = false;

void setup() {
    Wire.begin();
    lcd.init();
    lcd.backlight();

    lcd.setCursor(0, 0);
    lcd.print("Reaction Timer");
    delay(2000);
    lcd.clear();

    // Set LED pins as OUTPUT
    for (int i = 0; i < 2; i++) {
        pinMode(ledPins[i], OUTPUT);
        digitalWrite(ledPins[i], LOW); // Ensure LEDs start OFF
```

```
  }

  // Use INPUT_PULLUP for buttons
  for (int i = 0; i < 2; i++) {
    pinMode(buttonPins[i], INPUT_PULLUP);
  }

  randomSeed(analogRead(A0)); // Better randomization
  startGame();
}

void loop() {
  if (waitingForReaction) {
    for (int i = 0; i < 2; i++) {
      if (digitalRead(buttonPins[i]) == LOW) { // Button Pressed
        delay(50); // Debounce
        if (digitalRead(buttonPins[i]) == LOW) { // Confirm press
          reactionTime = millis() - startTime;
          lcd.clear();

          if (i == chosenLED) { // Correct button
            lcd.setCursor(0, 0);
            lcd.print("Correct!");
            lcd.setCursor(0, 1);
            lcd.print("Time: ");
            lcd.print(reactionTime);
            lcd.print(" ms");
          } else { // Wrong button
```

```
            lcd.setCursor(0, 0);
            lcd.print("Wrong Button!");
          }

          digitalWrite(ledPins[chosenLED], LOW);
          waitingForReaction = false;
          delay(3000);
          startGame();
        }
      }
    }
  }
}

void startGame() {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Wait...");

  // Ensure all LEDs are OFF
  for (int i = 0; i < 2; i++) {
    digitalWrite(ledPins[i], LOW);
  }

  while (anyButtonPressed()) {
    delay(100);
  }
  int randomDelay = random(2000, 5000);  // Random delay between 2 to 5 seconds
  delay(randomDelay);

  chosenLED = random(0, 2); // Select either 0 (Blue) or 1 (Green)

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Press NOW!");
  digitalWrite(ledPins[chosenLED], HIGH);

  startTime = millis();
  waitingForReaction = true;
}

// Helper function to check if any button is already pressed
bool anyButtonPressed() {
  for (int i = 0; i < 2; i++) {
    if (digitalRead(buttonPins[i]) == LOW) return true;
  }
  return false;
}
```

DEMONSTRATION/SIMULATION

The project was simulated on TinkerCAD, where:

- The LEDs correctly light up at random intervals.
- The LCD successfully displays the reaction time when the correct button is pressed.
- Incorrect button presses are detected and displayed as an error message.
- The system resets after each test cycle

KEY TAKEAWAYS

- The randomness ensures fair testing.
- The LCD provides immediate feedback.
- The system is simple but effective for measuring reflexes.

CONCLUSION

CHALLENGES

- Ensuring button debouncing (not implemented here but can be added).
- LCD display formatting for better readability.

IMPROVEMENTS

- Adding sound feedback (buzzer)
- Multiplayer
- Multiple test modes (e.g., average of 5 trials)
- The project can be expanded with additional features such as data logging

This Arduino-based reaction time tester effectively measures human response time in an affordable and interactive manner.