**MAHARASHTRA (INDIA)**



**Project Title**

###### REACTION TIMER GAME

###### FY B.Tech. DAS Report

|  |  |  |
| --- | --- | --- |
|  | **SUBMITTED BY** |  |
| **Student Name:** Sanskruti Shinde | **PRN No:** 202301060044 |
| **Student Name:** Prem Sonkar | **PRN No:** 202301060049 |
| **Student Name:** Prem Late | **PRN No:** 202301060031 |
|  |  |

**GUIDED BY**

Dr. V.G.PURANIK

**SCHOOL OF ENTC ENGINEERING AND TECHNOLOGY MIT ACADEMY OF ENGINEERING, ALANDI (D), PUNE-412105**

### CONTENTS

|  |  |
| --- | --- |
| 1. | OBJECTIVES |
| 2. | INTRODUCTION |
| 3. | METHODOLOGY-block diagram |
| 4. | THEORY-working |
| 5. | SIMULATION |
| 6. | CONSTRUCTION |
| 7. | RESULT AND ANALYSIS |
| 8. | CONCLUSION |
| 9. | REFERENCES- literature review |
| 10. | VIDEO LINK |
| 11. | ANNEXTURE- code |

# 

**OBJECTIVES:**

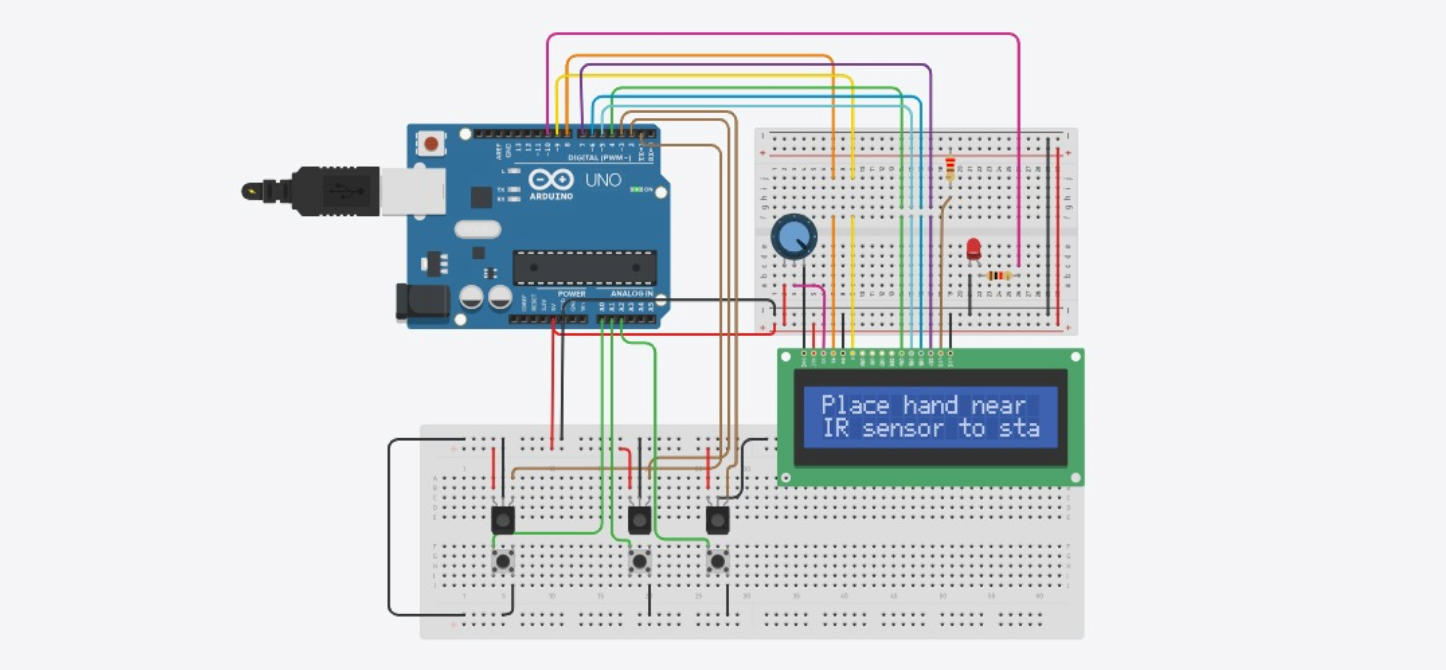
1. To Develop a user interface.
2. To Implement stimulus presentation.
3. To Integrate response detection.
4. To Calculate reaction time.
5. To Provide feedback and scoring.

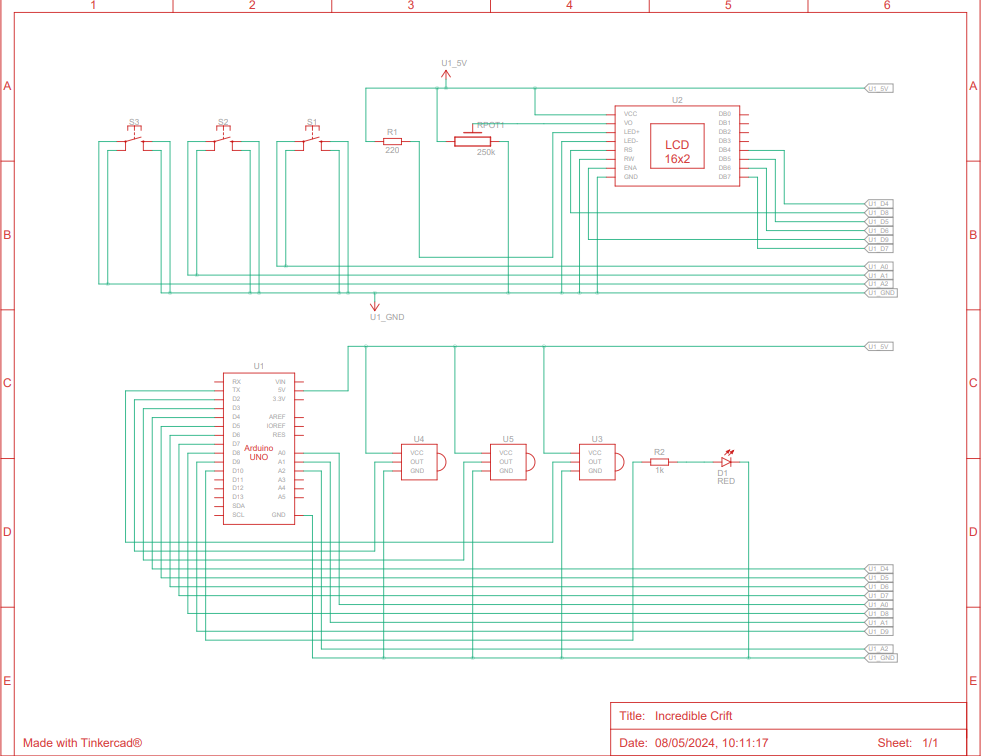
**INTRODUCTION:**

# We’ve designed a data acquisition in which, once the user starts the game, the Arduino chooses random delay. after that delay, led turns off then has to press the button as fast as possible. the Arduino measures the time between turning of the led and use.

# Step into the world of rapid reactions with our Arduino-powered Reaction Timer Game! As you embark on this adrenaline-pumping journey, prepare to engage your senses and hone your reflexes. With every moment counting, can you beat the clock and emerge victorious? It's time to find out.

**CIRCUIT DIAGRAM :-**





**WORKING :**

Connect an LED, a push button, and a few resistors to your Arduino board. The LED will act as the indicator, and the push button will serve as the trigger.

Write the Arduino code to control the game logic. You'll need to use functions like random() to generate a random delay before turning off the LED. Then, you'll wait for the user to press the button and measure the reaction time.

Use millis() function to keep track of time accurately. This function returns the number of milliseconds since the Arduino board began running the current program.

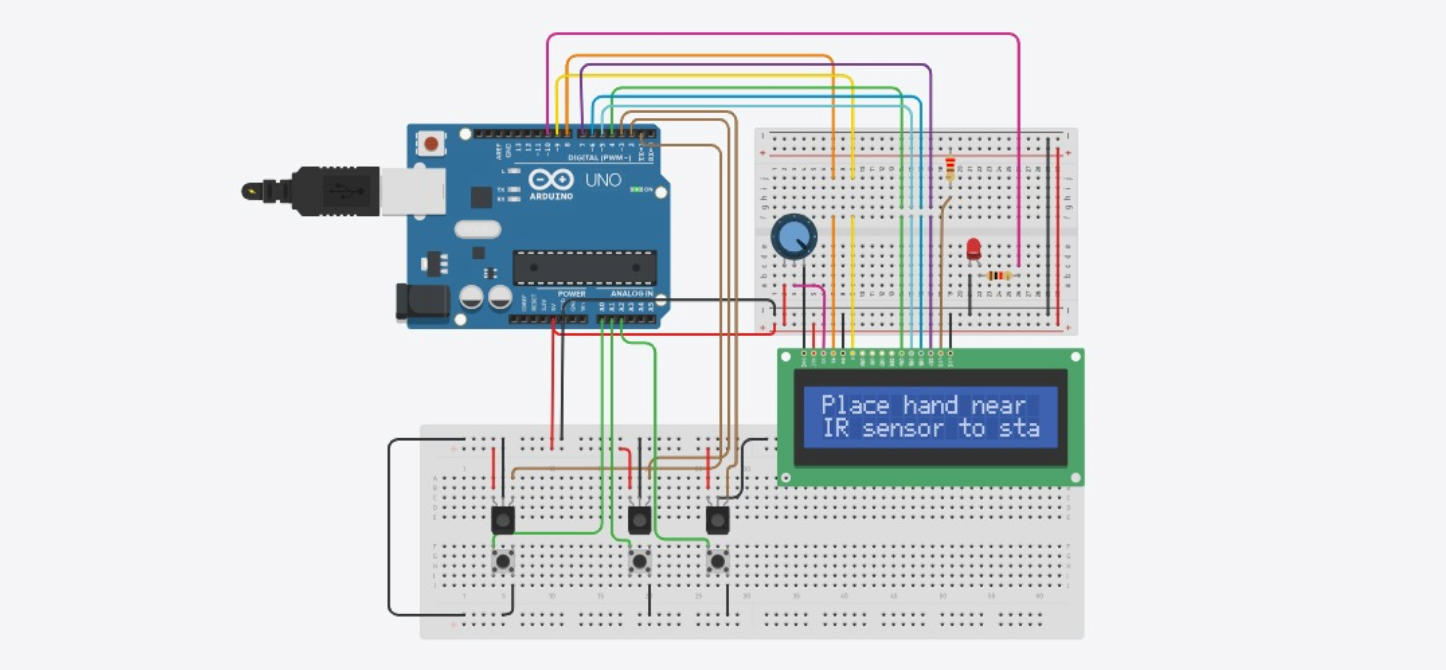
After the user presses the button, you can display the reaction time on a display, like an LCD screen or serial monitor, and provide feedback, like "Your reaction time: X milliseconds."

Keep the game running in a loop so the user can play multiple rounds.

Optionally, you can keep track of the fastest reaction time or implement a scoring system to make the game more competitive.

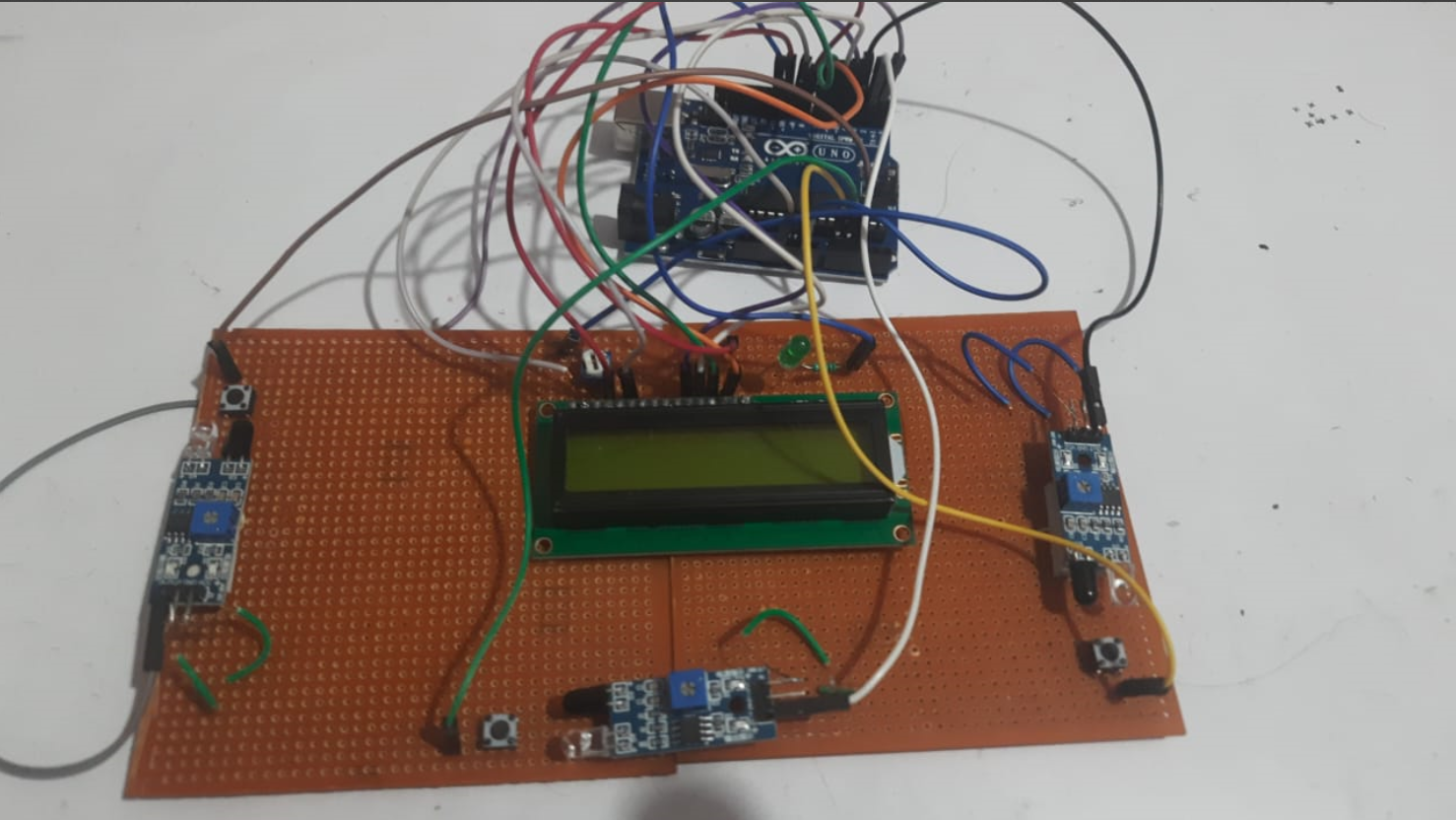
Ensure that the circuit and components are safely connected and that the user cannot accidentally touch any live wires.

**SIMULATION :**

****

**CONSTRUCTION OF PROJECT**

****



**RESULT AND ANALYSIS**

Results for reaction timer game is:

1. Keeping user engaged.
2. Compares reaction times between players.
3. Displays winner and his/her winning time.
4. Increases reaction time and hand-eye co-ordination.

ANALYSIS:

The analysis indicates that the reaction timer game effectively combines engagement, competition, and skill improvement, making it an enjoyable and beneficial experience for users.

**CONCLUSION**

* Analyze how well players reacted overall.
* Evaluate the level of engagement the game generated.
* Discuss the potential for the game to improve reaction times with continued practice.
* Highlight any educational benefits of the game, such as improving cognitive skills like attention and response inhibition

**REFERENCES**

* <https://forum.arduino.cc/t/reaction-time-arduino-program/553630>

AUTHOR: SHUXUNCHI

PUBLISHED ON: OCT 2018

We’ve designed a DIY device to measure the time taken for pressing buttons. the data collected includes timestamps for each button press, as we’re focused solely on recording the time of button presses rather than reaction times to external stimuli.

* [https://www.researchgate.net/publication/363227400\_Design\_of\_a\_set\_of\_interfaces\_to\_estimate\_whether\_computer\_games\_improve\_user%27s\_skills\_and\_abilities#pf8](https://www.researchgate.net/publication/363227400_Design_of_a_set_of_interfaces_to_estimate_whether_computer_games_improve_user%27s_skills_and_abilities)

AUTHOR: CHRISTIAN BAINES AND EMANUELE LINDO SECCO

PUBLISHED ON: SEPT 2022

Evolving with the internet, computer games are accessible on portable systems and online. research suggests they may enhance cognitive skills like memory, reaction time, and hand eye co-ordination, with potential applications in diverse fields.

* <https://www.researchgate.net/publication/361412722_Design_And_Development_Of_Reaction_Timer_Device_Based_Visual_Reaction_Time_Vrt_Scale_Stimulant_For_Silat_Athlete>

AUTHORS : CHUN NG, ZAIN MUHAMAD KEAT, NUR ANIDA ZAIM, AND NUR ANIDA JUMADI.

PUBLISHED ON: MARCH 2022

This study developed a reaction timer device to measure the reaction time of silat athletes during layang kick performances outdoors. nine male participants across different weight categories were recruited. results shows thst 66.7% of participants reacted fastest to the green led stimulus.

* <https://www.scribd.com/document/510881411/Reaction-Time-Games>

AUTHOR: TAN YI KA

PUBLISHED ON:

Reaction time is fun and meaningful mini game that can help player to test their reaction time. the faster the reaction time can help to avoid any sudden situation such like car accident and other accident in daily life. with this game, the reaction speed can be known and increase if used it day by day. it’s also very simple and easy to use especially when feel boring.

**VIDEO LINK**

[https://drive.google.com/file/d/13\_8pwIXjsTIJREg7ACLlrZtLz86Nmig6/view?usp=drivesdk](https://d.docs.live.net/6005511a5070dbba/Desktop/REPORT1.docx)

**ANNEXTURE**

CODE-

#include <LiquidCrystal.h>

// LCD pins

const int rs = 8, en = 9, d4 = 4, d5 = 5, d6 = 6, d7 = 7;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

// IR sensor pins

const int numOfIR = 3; // Number of IR sensors

const int irSensorPins[numOfIR] = {1, 2, 3}; // IR sensor pins

const int irSensorPowerPins[numOfIR] = {10, 11, 12}; // IR sensor power control pins

// Button pins

const int numOfButtons = 3; // Number of buttons

const int reactionButtonPins[numOfButtons] = {A0, A1, A2}; // Reaction button pins

// LED pins

const int numOfPlayers = 3; // Number of players

const int ledPins[numOfPlayers] = {13, 14, 15}; // LED pins

// Player scores

unsigned long playerScores[numOfPlayers] = {0};

void setup() {

// Initialize LCD

lcd.begin(16, 2);

// Initialize IR sensor pins

for (int i = 0; i < numOfIR; i++) {

pinMode(irSensorPins[i], INPUT);

pinMode(irSensorPowerPins[i], OUTPUT);

digitalWrite(irSensorPowerPins[i], HIGH); // Power on initially

}

// Initialize button pins

for (int i = 0; i < numOfButtons; i++) {

pinMode(reactionButtonPins[i], INPUT\_PULLUP);

}

// Initialize LED pins

for (int i = 0; i < numOfPlayers; i++) {

pinMode(ledPins[i], OUTPUT);

digitalWrite(ledPins[i], LOW);

}

// Display initial message on LCD

lcd.clear();

lcd.print("Place hand near");

lcd.setCursor(0, 1);

lcd.print("IR sensor to start");

}

void loop() {

// Wait for all IR sensors to stop sensing

while (true) {

bool allSensorsStopped = true;

for (int i = 0; i < numOfIR; i++) {

if (digitalRead(irSensorPins[i]) == HIGH) {

allSensorsStopped = false;

break;

}

}

if (allSensorsStopped) {

break;

}

}

// Countdown

lcd.clear();

for (int i = 3; i > 0; i--) {

lcd.print("Starting in ");

lcd.print(i);

delay(1000);

lcd.clear();

}

// Add random delay between 4 to 5 seconds

delay(random(4000, 5001));

// Cut off power to IR sensors

for (int i = 0; i < numOfIR; i++) {

digitalWrite(irSensorPowerPins[i], LOW);

}

// Flash LEDs

for (int i = 0; i < numOfPlayers; i++) {

digitalWrite(ledPins[i], HIGH);

}

unsigned long startTime = millis();

// Wait for button press

int winnerButtonIndex = -1;

while (true) {

for (int i = 0; i < numOfButtons; i++) {

if (digitalRead(reactionButtonPins[i]) == LOW) { // Check for button press

winnerButtonIndex = i;

break;

}

}

if (winnerButtonIndex != -1) {

break;

}

}

// Restore power to IR sensors

for (int i = 0; i < numOfIR; i++) {

digitalWrite(irSensorPowerPins[i], HIGH);

}

// Turn off LEDs

for (int i = 0; i < numOfPlayers; i++) {

digitalWrite(ledPins[i], LOW);

}

// Calculate reaction time

unsigned long reactionTime = millis() - startTime;

// Display reaction time on LCD

lcd.clear();

lcd.print("Reaction Time:");

lcd.setCursor(0, 1);

lcd.print(reactionTime);

// Update player scores

playerScores[winnerButtonIndex] = reactionTime;

// Display winner on LCD

lcd.clear();

lcd.print("Winner: Player ");

lcd.print(winnerButtonIndex + 1);

lcd.setCursor(0, 1);

lcd.print("Score: ");

lcd.print(playerScores[winnerButtonIndex]);

// Wait before restarting game

delay(5000);

}