DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,
COLLEGE OF ENGINEERING, GUINDY, ANNA UNIVERSITY.

CS6105 DIGITAL FUNDAMENTALS AND COMPUTER ORGANISATION

MINI PROJECT

FASTEST FINGER FIRST CIRCUIT (QUIZ BUZZERS) & QUIZ GAME USING ARDUINO

KIRUTHIGA P M

2022103042

B.E Computer Science and Engineering

3rd Semester

P-BATCH

JOTHI SRI S

2022103049

B.E Computer Science and Engineering

3rd Semester

P-BATCH

SUBMITTED ON : 19.12.2023

FASTEST FINGER FIRST CIRCUIT (QUIZ BUZZERS) & QUIZ GAME USING ARDUINO

OBJECTIVE:

Our project aims to develop a quiz game using **Arduino** and a reliable **fastest finger first circuit** which would be able to indicate the fastest person to hit the button first using **555 Timer IC**, displaying the team number, using **BCD to seven segment display** and combination of OR gates.

DESCRIPTION:

Fastest Finger First Circuit

This circuit consists of multiple replicas of a module, each of which can be assigned to a single quiz taker. Each module contains a trigger and an LED indicating whether the team corresponding to this module is the first one to press the trigger. A monostable arrangement can be coupled to the output of this circuit so that a light or buzzer beeps for a second after the first trigger from any quiz taker.

The basic behind the circuit is, If Pin-2 of 555 IC sees any voltage less than 1/3rds of the supply voltage, it turns the output ON. If the reset Pin of 555 IC sees 0V, it resets the output.

Apart from positive and negative rails, we used two other rails, Reset rail which is pulled up to positive voltage by default using a 1K resistor. Status or Feedback rail which is pulled down to 0V by default using a 1K resistor. This rail is connected to the output of all the modules via diodes. So, this rail will be at 0V by default. But as soon as any module output is ON, the voltage at this rail reaches positive voltage via the diode.

So, when the outputs of all the modules are in the OFF state, the voltage at the status rail will be at 0V (default). When any team presses the button, this 0V from the status rail is applied at Pin-2. Because 0V is less than 1/3rds of the supply voltage, the output of 555 IC corresponding to the team that pressed the trigger first turns ON. Immediately after this happens, the voltage at the status rail changes to positive voltage because of the feedback via the PN diode. So even if other teams press the trigger now, the voltage at Pin-2 of the respective modules will be at the positive voltage and the output doesn't turn ON. Resetting of states of all the modules is done by applying 0V at the reset pin of all the 555 ICs using a reset rail and a dedicated push button.

Inorder to display the output in seven segment display the quiz output is converted into BCD and this BCD is decoded to seven segment display using a BCD decoder

What is BCD to Seven Segment Decoder?

A seven-segment display is an electrical device that displays Hexadecimal digits using seven Light Emitting Diodes (LEDs) arranged in a specific pattern (common cathode or common anode type) (in this case, decimal numbers, as input, are BCD, i.e., 0-9).

7-Segment LED displays are Common Cathode Type (CCT) and Common Anode Type (CAT). Here, we are using **CCT (IC7448)**.

The separate LED segments of a seven-segment display are not directly powered. A BCD to seven-segment decoder converts that signal into the form supplied to the seven-segment display. **The seven-segment decoder, which has four input lines and seven output lines (a, b, c, d, e, f, and g), receives this BCD (A, B, C, and D) input.** The output is provided to a seven-segment LED display that shows the decimal number depending on the inputs.

Arduino Simple Quiz Game

The Arduino Simple Quiz Game is an interactive and educational project that utilizes an Arduino uno controller to create a basic quiz game. It aims to engage users in a fun learning experience by presenting multiple-choice questions and allowing them to select the correct answers using buttons. For button debouncing and ensuring proper current flow, 10K resistors are used.

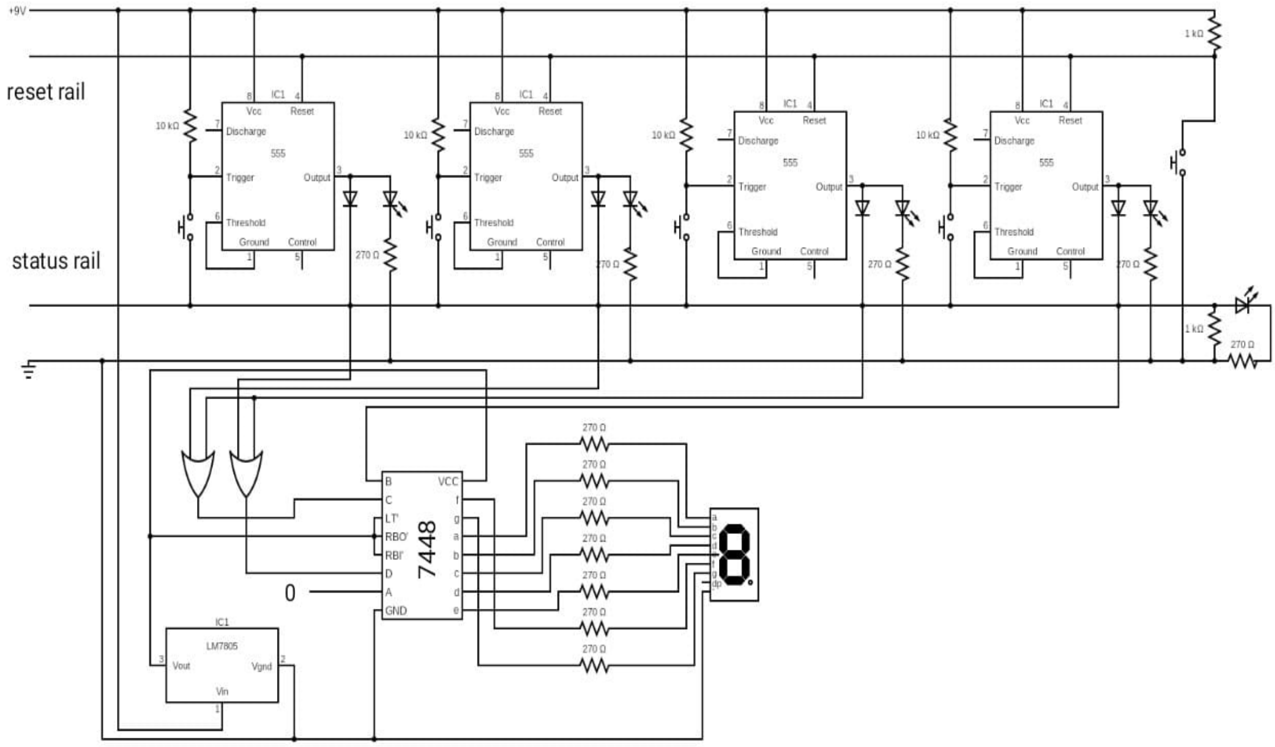
Question display is done by the Arduino which controls the LCD to display questions, one at a time, along with multiple-choice options. Users can select their answers using a push button. The system is designed to handle multiple-choice questions. Instant feedback is provided to the user after each question, indicating whether their answer was correct or incorrect. This can be done through the LCD. The Arduino keeps track of the user's score throughout the quiz. The program manages the flow of the quiz, progressing to the next question after the user has answered the current one. At the end of the quiz, the Arduino displays the user's final score and a message based on their performance.

COMPONENTS NEEDED:

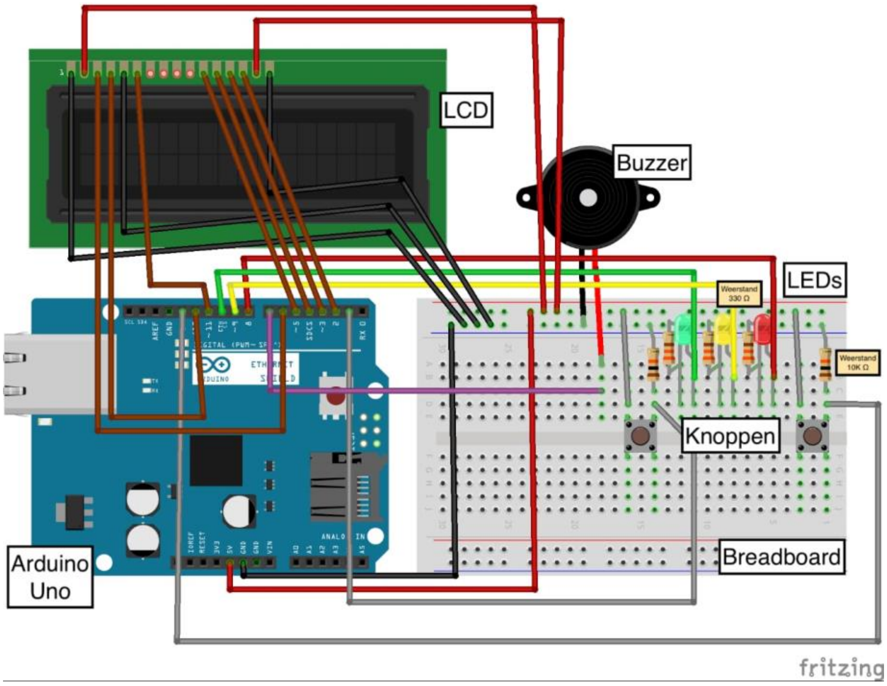
S.no	Components	Quantity
1.	9V Battery and battery cap	1
2.	Breadboard	3
3.	Push Button	7
4.	Buzzer	2
5.	LED’s	8
6.	Seven Segment Display (cc)	1
7.	Voltage Regulator (5V)	1
8.	IC 7432	1
9.	IC 7448	1
10.	555 Timer IC	4
11.	PN Diode (1N 4148)	4
12.	Arduino Uno	1
13.	16X2 LCD Display	1
14.	10KΩ Potentiometer	1
15.	Jump wires	10
16.	1KΩ Resistor 10KΩ Resistor 270Ω Resistor 330Ω Resistor	16 6 7 3
17.	Connecting wires	As required

CIRCUIT DIAGRAM:

Fastest fingers first circuit with seven segment display:



Quiz game:



TRUTH TABLE:

Buzzer outputs				BCD Code			
W	**X**	**Y**	**Z**	**A**	**B**	**C**	**D**
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1
0	1	0	0	0	0	1	0
0	0	1	0	0	0	1	1
0	0	0	1	0	1	0	0

Truth Table for Decoding the buzzer output to BCD

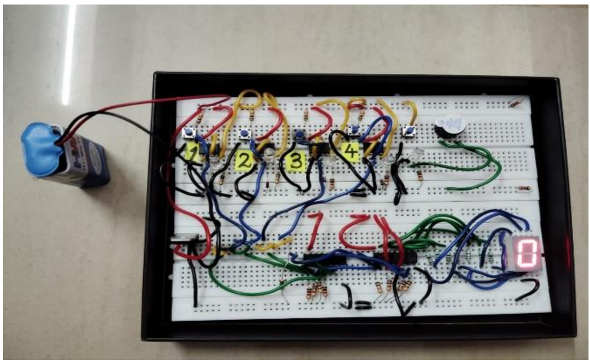
EQUATIONS:

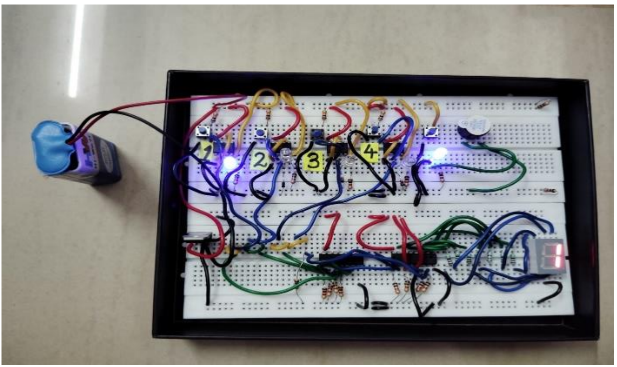
A=0 B=Z C=X+Y D=W+Y

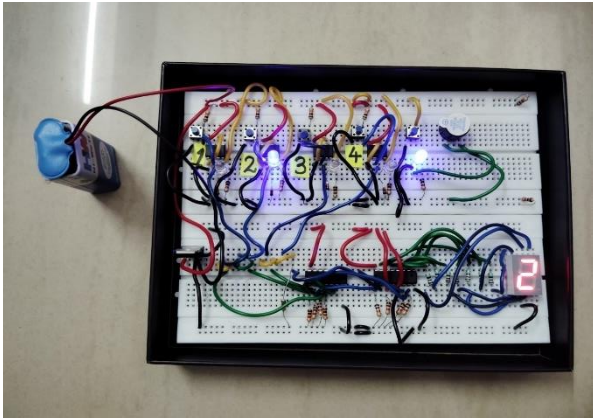
BCD Inputs				Output Logic Levels from IC 7447 to 7-segments							Decimal number display
D	**C**	**B**	**A**	**a**	**b**	**c**	**d**	**e**	**f**	**g**	
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	1	0	0	1	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0	2
0	0	1	1	0	0	0	0	1	1	0	3
0	1	0	0	1	0	0	1	1	0	0	4
0	1	0	1	0	1	0	0	1	0	0	5
0	1	1	0	1	1	0	0	0	0	0	6
0	1	1	1	0	0	0	1	1	1	1	7
1	0	0	0	0	0	0	0	0	0	0	8
1	0	0	1	0	0	0	1	1	0	0	9

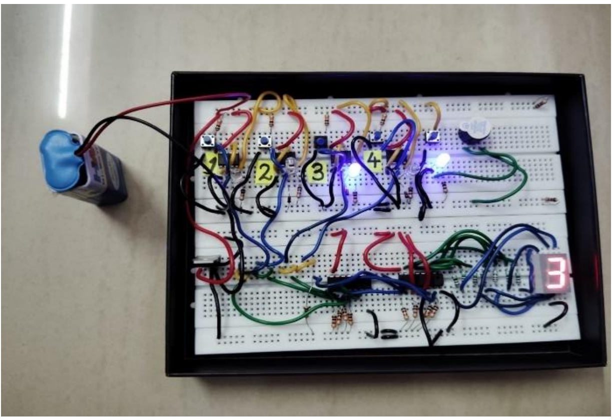
Truth Table for BCD to Seven segment decoder

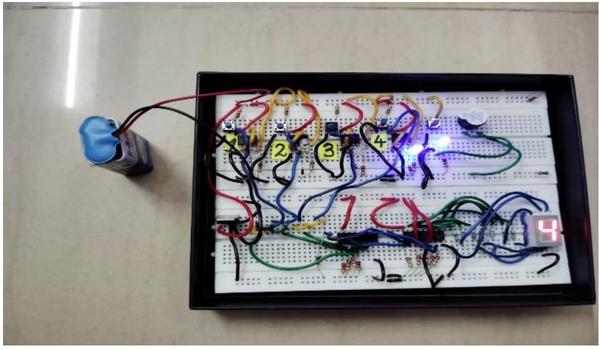
OUTPUTS OF THE PROJECT:

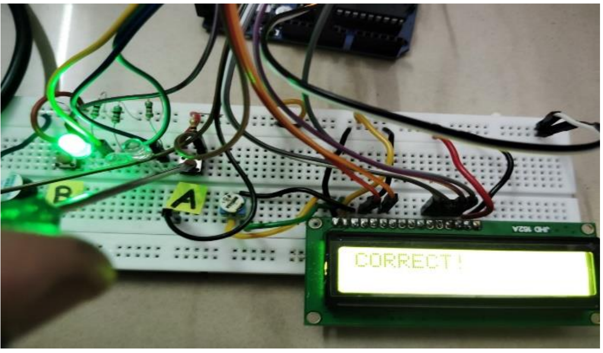


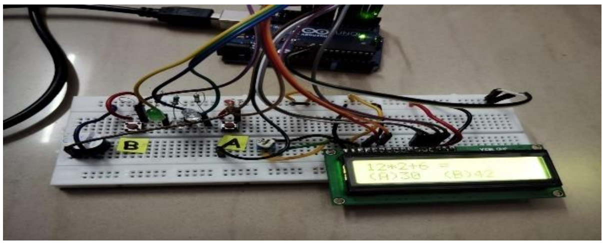


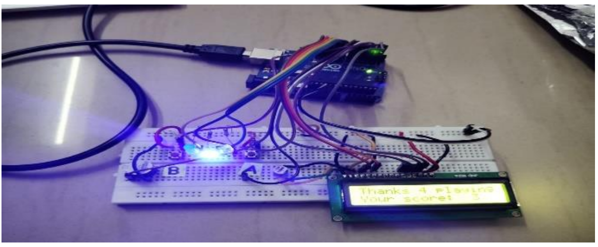


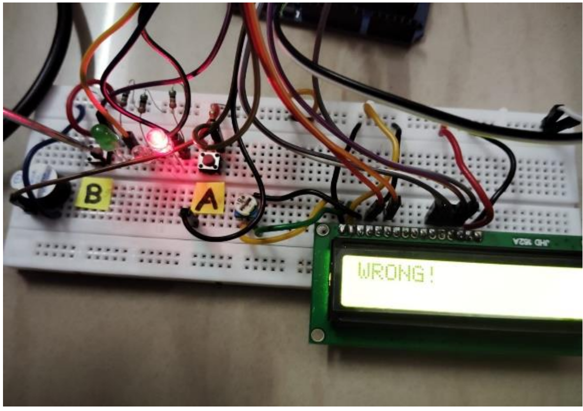


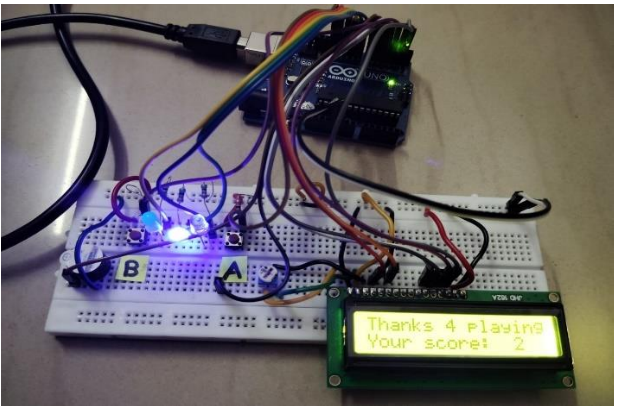












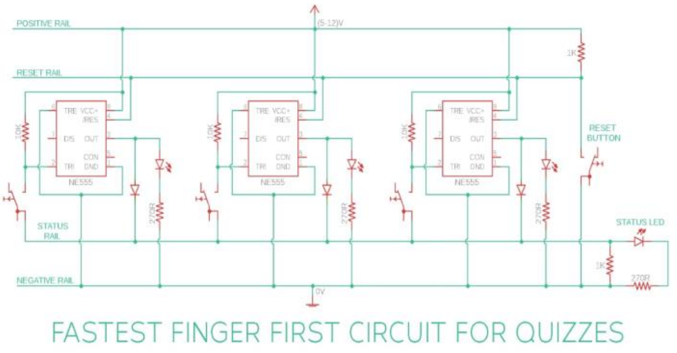
REFERENCE LINKS:

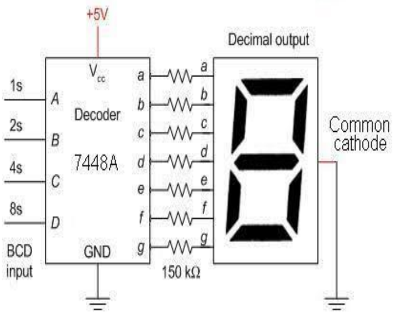
https://youtu.be/V2MLKod1G9U?si=UXWp1oUYuN7XqHhe

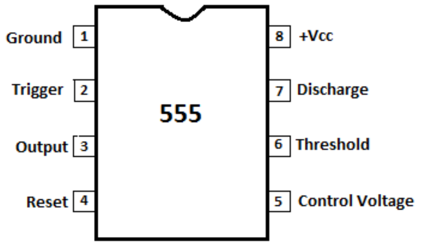
https://youtu.be/OqT0pV7JWu8?si=pvZP6wWtY7gu0yUx

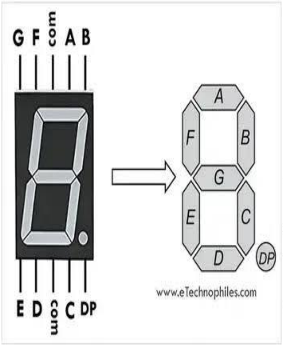
https://youtu.be/HU-3VD1_Pgg?si=kOFMeHAsndUwuU3H

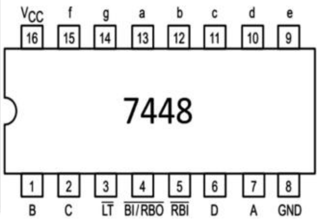
REFERENCE IMAGES AND PIN CONFIGURATION:











CONCLUSION:

In conclusion, the "Fastest Finger First" circuit quiz game project is an engaging and competitive way to test participants' knowledge. With its responsive design, accurate timing mechanism, and well-implemented scoring system, it offers an exciting experience for players. Additionally, the incorporation of user-friendly controls enhances the overall usability of the game. Overall, this project successfully combines electronics and programming to create an entertaining and educational quiz game.

CODE:

```
#include <LiquidCrystal.h>
int Contrast = 150; //depends on your screen
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
boolean buttonLeft;
boolean buttonRight;
String topQuestions[] = {"INDIA CAP IS", "4+5 =",
"X+X'=", "X+0 =", "12*2+6 ="};
String bottomQuestions[] = {"(A)DELHI (B)TN", "(A)8
(B)9", "(A)0  (B)1", "(A)X  (B)0", "(A)30  (B)42"};
int maxNumQuestions = 5;
int NumQuestions = 0;
int answers[] = {1, 0, 0, 1, 1};    //1 is correct - 0 is wrong
int score = 0;
void setup()
{
    pinMode(13, INPUT);  //Rechter knop
    pinMode(1, INPUT);   //Linker knop
    pinMode(10, OUTPUT); //LED groen
    pinMode(9, OUTPUT);  //LED geel
    pinMode(8, OUTPUT);  //LED rood
    analogWrite(6, Contrast);
    lcd.begin(16, 2);
    lcd.setCursor(0, 0);            //Introduction welcome
    lcd.print("Welcome");
    lcd.setCursor(0, 1);
    lcd.print("ARE YOU READY");
    delay(3000);
    lcd.clear();
    lcd.print("Quiz game!");
    delay(3000);
    lcd.clear();
    lcd.print("Starting...")
    lcd.clear();
}
void correctAnswer()
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("CORRECT!");
    digitalWrite(10, HIGH);
    tone(7, 2500, 100);
    delay(100);
    tone(7, 3500, 100);
    delay(250);
    digitalWrite(10, LOW);
    noTone(7);
    delay(2000);
    lcd.clear();
}
void wrongAnswer()
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("WRONG!");
    digitalWrite(8, HIGH);
    tone(7, 1350, 100);
    delay(100);
    tone(7, 1000, 100);
    delay(250);
    digitalWrite(8, LOW);
    noTone(7);
    delay(2000);
    lcd.clear();
```

```
}
void marioIntro()
{
    tone(7, 660, 100);
    delay(150);
    tone(7, 660, 100);
    delay(300);
    tone(7, 660, 100);
    delay(300);
    tone(7, 510, 100);
    delay(100);
    tone(7, 660, 100);
    delay(300);
    tone(7, 770, 100);
    delay(550);
    tone(7, 380, 100);
    delay(575);
}
void showScore()
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Thanks 4 playing");
    lcd.setCursor(0, 1);
    lcd.print("Your score:");
    lcd.setCursor(13, 1);
    lcd.print(score);
    if (score > 3)
    {
        digitalWrite(10, HIGH);
    }
    {
        digitalWrite(9, HIGH);
    }
    else if (score < 3)
    {
        digitalWrite(8, HIGH);
    }
    delay(30000);                       //end of game
}
void loop()
{
    buttonLeft = digitalRead(1);         //False button
    buttonRight = digitalRead(13);       //True button
    if (NumQuestions != maxNumQuestions)
    {
        lcd.setCursor(0, 0);                        //display question
        lcd.print(topQuestions[NumQuestions]);
        lcd.setCursor(0, 1);                        //display possible answer
        lcd.print(bottomQuestions[NumQuestions]);
        if (buttonRight == HIGH)                    //if True button is clicked
        {
            if ((answers[NumQuestions]) == 1)
            {
                correctAnswer();
                NumQuestions++;
                score++;
            }
            else if ((answers[NumQuestions]) == 0)
            {
                wrongAnswer();
                NumQuestions++;
            }
        }
        if (buttonLeft == HIGH)                    //if False
button is clicked
        {
            if ((answers[NumQuestions]) == 0)
            {
                correctAnswer();
                NumQuestions++;
                score++;
            }
            else if ((answers[NumQuestions]) == 1)
            {
                wrongAnswer();
                NumQuestions++;
            }
        }
    }
    if (NumQuestions == maxNumQuestions)
//5 questions solved? --> score
    {
        showScore();
    }
}
```