INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 04, Issue 11, November 2024, pp : 1638-1642
www.ijprems.com
editor@ijprems.com
e-ISSN : 2583-1062
Impact Factor : 7.001

QUIZ BUZZER SYSTEM USING ARDUINO UNO

Shirgaonkar Muhammed[1], Afrin Shaikh[2], Khwaja Ayaan[3], Affan Shaikh[4], Aatif Khan[5], Adil Khan[6], Adnan Shaikh[7], Zahiruddin Khateeb[8]

[1,2,3,4,5]Student, Mechatronics, AIARKP, Navi Mumbai, Maharashtra, India.

[6,7]Lecturer, Mechatronics, AIARKP, Navi Mumbai, Maharashtra, India.

[8]H.O.D, Mechatronics, AIARKP, Navi Mumbai, Maharashtra, India.

ABSTRACT

This project outlines a **Quiz Buzzer System** using **Arduino UNO**, designed to identify the first responder in quizevents. Each participant has a button and LED indicator, and pressing the button lights the LED and sounds a buzzer for the first response, locking out others until reset. This setup is ideal for fair and efficient response tracking in educational and entertainment quizzes, showcasing Arduino's capabilities in real-time systems. The project is simple, scalable, and provides a hands-on introduction to embedded electronics. The Quiz Buzzer System using Arduino is an interactive electronic system designed to facilitate quiz competitions, enabling participants to quickly respond to questions. The system consists of a central controller, typically an Arduino microcontroller, which is interfaced with multiple buzzers for each contestant, and an LCD display or LED indicators to signal the winner. The system works by allowing participants to press a button on their individual buzzers when they know the answer, with the first button press being registered by the Arduino and locking out the other buzzers. The system incorporates visual and audible signals to clearly indicate which contestant has pressed their buzzer first. The project is implemented using simple components such as push buttons, LEDs, and an Arduino board, providing an affordable and reliable solution for quiz-style competitions in educational or recreational settings. This project demonstrates the integration of hardware and software to create a responsive and user-friendly system, enhancing the quiz experience by ensuring fairness and accuracy in timing.

Keywords: Arduino, Quiz Buzzer, Contestant Buzzer, Timing System, LCD Display, LED Indicators, Interactive System.

1. INTRODUCTION

Quiz competitions are a popular and interactive way to test knowledge, often used in educational, corporate, and social events. The need to quickly identify the first participant to respond makes an automated buzzer system highly valuable in quiz setups. This project involves building a quiz buzzer system using an Arduino UNO, a microcontroller platform known for its simplicity and versatility. In this system, each participant is assigned a button and an LED indicator. When a participant presses their button, their LED lights up and a buzzer sounds, identifying them as the first to respond. Subsequent button presses are ignored until the system is reset, allowing for fair and transparent quiz rounds. The project showcases Arduino's capabilities in controlling electronics and provides a hands-on experience in real-time response systems.

2. METHODOLOGY

2.1 System Architecture

a. Arduino Microcontroller (Central Controller)

- **Function**: Acts as the central unit to control the system's operations, process input from contestant buzzers, manage the timing, and trigger signals to display or sound indicators (like LEDs or sound alarms).
- **Components**: An Arduino board (e.g., Arduino Uno) is used as the central microcontroller, which is programmed to handle input from the buzzers and provide output to indicate the first contestant to buzz in.

b. Contestant Buzzers

- Function: Each contestant has a physical button or buzzer to press when they want to answer. The button press sends an electrical signal to the Arduino, indicating that the contestant is ready to answer.
- **Components:**
- Push buttons or touch sensors connected to the Arduino.
- Optional buzzer or LED indicator to signal the contestant that their input has been registered.

c. Input Mechanism

- Function: The input mechanism consists of the buttons pressed by the contestants. When a contestant presses their button, the signal is sent to the Arduino, which registers the press. The system locks out other contestants once a

www.ijprems.com
editor@ijprems.com
INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 04, Issue 11, November 2024, pp : 1638-1642
e-ISSN : 2583-1062
Impact Factor : 7.001

button has been pressed, ensuring only the first contestant gets to answer.

- **Components:**
- Push buttons (one per contestant).
- Each button is connected to a digital input pin on the Arduino.
- The system uses pull-down or pull-up resistors to ensure a stable digital signal when the button is pressed.

d. Timing Control

- **Function**: The Arduino keeps track of the time when each contestant presses their buzzer. This ensures that only the first contestant's buzzer is acknowledged, and others are locked out. Timing control also helps prevent issues like "double buzzing."
- **Components**:
- Software logic implemented in the Arduino code to track the exact time of the buzzer press.
  - Optional: A real-time clock (RTC) module for accurate time tracking, but in most cases, the Arduino's internal timer is sufficient.

e. Output Indicators (LEDs or LCD Display)

- **Function**: Provides visual feedback to the participants and audience about which contestant has pressed their buzzer first.
- **Components**:
- **LED Indicators**: Each contestant has an LED light that turns on when they press their buzzer. A unique LED is associated with each contestant.
- **LCD Display**: Optionally, an LCD can be used to display the contestant number or their status (e.g., "Ready", "Pressed", or "Locked Out").
- **Buzzer Sound**: A simple piezoelectric buzzer or speaker can be used to make an audible beep when a contestant presses the buzzer.

f. Lock-out Mechanism

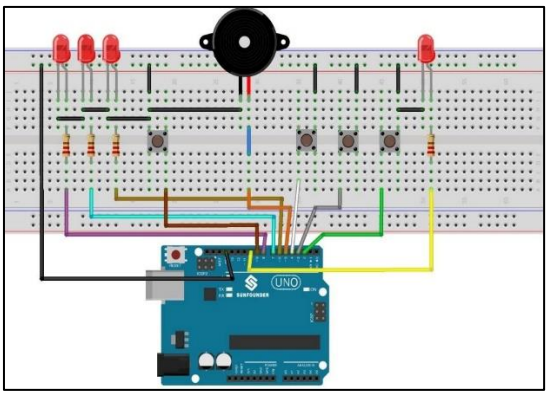
- **Function**: Prevents multiple contestants from buzzing in simultaneously or after the first contestant has pressed their buzzer. Once a contestant's buzzer is pressed, the system "locks out" the remaining contestants to ensure fairness.
- **Components**:
- **The** Arduino code checks which button is pressed first and locks the remaining buzzers until the question is answered or reset.

g. Reset Mechanism

- **Function**: After a contestant answers a question, the system must be reset to allow for the next round. This includes unlocking the buzzers and resetting all indicators.
- **Components**:
- A reset button or an automated process controlled by the Arduino.

3. MODELING AND ANALYSIS

Circuit Diagram



www.ijprems.com
editor@ijprems.com
INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 04, Issue 11, November 2024, pp : 1638-1642
e-ISSN : 2583-1062
Impact Factor : 7.001

Figure 1: Circuit diagram.

Circuit Architecture

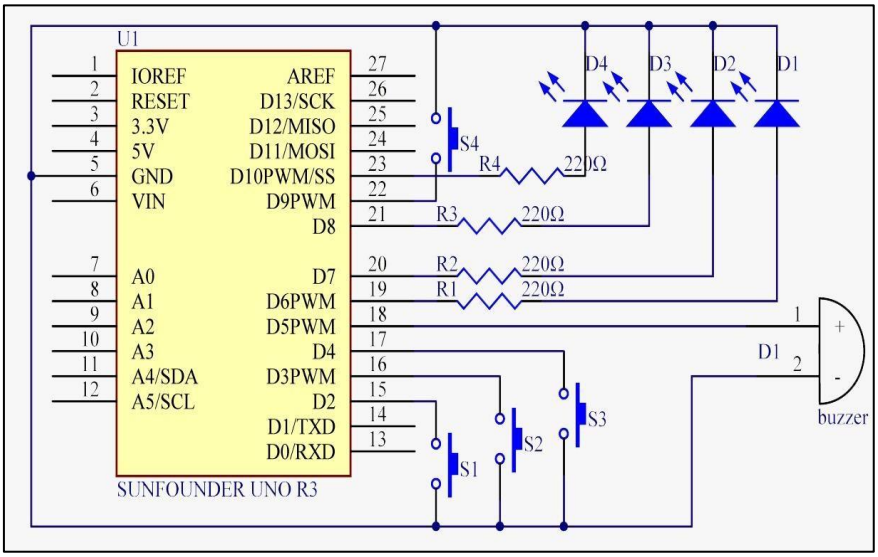


Figure 2: Circuit Architecture.

1. Arduino Code

```
#define button1 2
#define button2 3
#define button3 4
#define button4 9
#define buzzerPin 5
#define LED1 6
#define LED2 7
#define LED3 8
#define LED4 10
#define uint8 unsigned char
uint8 flag =0;
uint8 b1State,b2State,b3State,b4State =0;
void setup() {
pinMode(buzzerPin, OUTPUT);
pinMode(LED1,OUTPUT);
pinMode(LED2,OUTPUT);
pinMode(LED3,OUTPUT);
pinMode(LED4,OUTPUT);
pinMode(button1,INPUT_PULLUP);
pinMode(button2,INPUT_PULLUP);
pinMode(button3,INPUT_PULLUP);
pinMode(button4,INPUT_PULLUP);
digitalWrite(LED1,LOW);
digitalWrite(LED2,LOW);
digitalWrite(LED3,LOW);
digitalWrite(LED4,LOW);}
void loop() {
digitalWrite(LED1,LOW);
digitalWrite(LED2,LOW);
digitalWrite(LED3,LOW);
digitalWrite(LED4,LOW);
b4State =digitalRead(button4);
if(b4State==0)
{ if(b4State ==0)
```

www.ijprems.com
editor@ijprems.com
INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 04, Issue 11, November 2024, pp : 1638-1642
e-ISSN : 2583-1062
Impact Factor : 7.001

```
{    flag =1;
digitalWrite(LED4,HIGH);
delay(200);}
}
if(1==flag)
{
b1State =digitalRead(button1);
b2State =digitalRead(button2);
b3State =digitalRead(button3);
if(b1State==0)
{    flag =0;
digitalWrite(LED4,LOW);
Alarm();
digitalWrite(LED1,HIGH);
digitalWrite(LED2,LOW);
digitalWrite(LED3,LOW);
while(digitalRead(button4));}
if(b2State==0)
{    flag =0;
digitalWrite(LED4,LOW);
Alarm();
digitalWrite(LED1,LOW);
digitalWrite(LED2,HIGH);
digitalWrite(LED3,LOW);
while(digitalRead(button4));}
if(b3State ==0)
{    flag=0;
digitalWrite(LED4,LOW);
Alarm();
digitalWrite(LED1,LOW);
digitalWrite(LED2,LOW);
digitalWrite(LED3,HIGH);
while(digitalRead(button4));}
}
}
void Alarm ()
{  for (int i=0; i<100; i++) {
digitalWrite(buzzerPin,HIGH);
delay(2);
digitalWrite(buzzerPin,LOW);
delay(2);}
}
```

2. EXPLANATION OF THE CODE:

2.1 Setup: The pins for each button and LED are defined, along with the buzzer. The pull-up resistor mode is used to simplify button wiring.

2.2 Loop: The code continuously checks if a button is pressed. When a button press is detected, the program identifies the first player who pressed it, activates the LED and buzzer for that player, and locks out furtherresponses until reset.

www.ijprems.com
editor@ijprems.com
INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 04, Issue 11, November 2024, pp : 1638-1642
e-ISSN : 2583-1062
Impact Factor : 7.001

2.3 Reset Function: The system can be reset manually, turning off all LEDs and allowing new button presses.

3. SYSTEM FLOW:

3.1 Initialization: At the start, all contestant buzzers are unlocked, and no buzzer has been pressed. LEDs are off, and the system is waiting for a contestant to press their button.

3.2 Contestant Buzzes: A contestant presses their buzzer. The Arduino detects the button press and locks out all other buzzers.

3.3 Indication: The system triggers the LED corresponding to the first contestant to buzz in, signaling that the buzzer has been pressed.

3.4 Timer Control: The Arduino monitors the time to ensure no other contestant can buzz in once the first buzzer has been pressed.

3.5 Question Answering: The contestant answers the question or is given a short time to respond.

3.6 Reset: After the question is answered or the time expires, the system resets to allow the next round to begin. All buzzers are unlocked, and the system waits for the next contestant to press their button.

4. APPLICATIONS:

4.1 Quiz competitions for identifying the first participant to answer.

4.2 Games and event management require quick response identification.

4. RESULTS AND DISCUSSION

The Quiz Buzzer System using Arduino UNO was tested in a simulated quiz environment to evaluate its performance in identifying the first responder and locking out subsequent inputs. The key observations from the testing phase include:

- Response Time: The system accurately detected the first button press and activated the corresponding LED and buzzer with minimal delay (less than 0.1 seconds).
- Input Lockout: Once the first response was registered, the system successfully prevented other participants' inputs from being processed, ensuring that only the initial responder's input was recorded.
- Reset Functionality: The manual reset mechanism allowed the system to be quickly reinitialized for subsequent questions, demonstrating effective control for quiz continuity.
- LED and Buzzer Indicators: The LED indicators and buzzer provided clear visual and auditory signals of the first response, enhancing the user experience and making the system effective for use in real-time quiz competitions.

The testing results confirm that the Arduino-based Quiz Buzzer System functions as intended, with reliable response tracking and user-friendly features. The immediate activation of the LED and buzzer ensures participants and moderators can quickly identify the first responder, minimizing confusion in competitive settings. The lockout mechanism, critical for fair play, operated seamlessly, preventing errors related to simultaneous inputs.

5. CONCLUSION

This project is a simple and effective implementation of a quiz buzzer system, ideal for school and college events. It showcases the use of Arduino in real-time response systems and can be further expanded to include more players or automated scoring.

6. REFERENCES

[1] Banzi, M., & Shiloh, M. (2014). Getting Started with Arduino. Maker Media, Inc.

[2] Monk, S. (2017). Programming Arduino: Getting Started with Sketches. McGraw-Hill Education.

[3] Purdum, J. (2020). Beginning C for Arduino. Apress.

[4] Arduino Forum (2023). Retrieved from forum.arduino.cc

[5] Bhagat, N. (2019). "Arduino Quiz Buzzer System." Instructible. Retrieved from instructables.com

[6] Wikipedia. (2023). "Arduino." Retrieved from wikipedia.org