



# Voice Controlled Digital Scoreboard

This project delivers a hands-free scoreboard controlled by claps.

Team A gets a point for one clap, and Team B for two. This presentation outlines the system's design, components, and performance.

We will also discuss future enhancements, like voice command recognition.

**Made by:**

~ Lakshit Trivedi (BT23EEE025)

~ Sanyam Gupta (BT23EEE034)

~ Hardik Pandey (BT23EEE045)

# System Architecture Overview

## Core Components

- KY 038 Sound sensor
- ESP32 microcontroller
- 4-bit 7 segment Display

## Functionality

Clap detection triggers score updates.

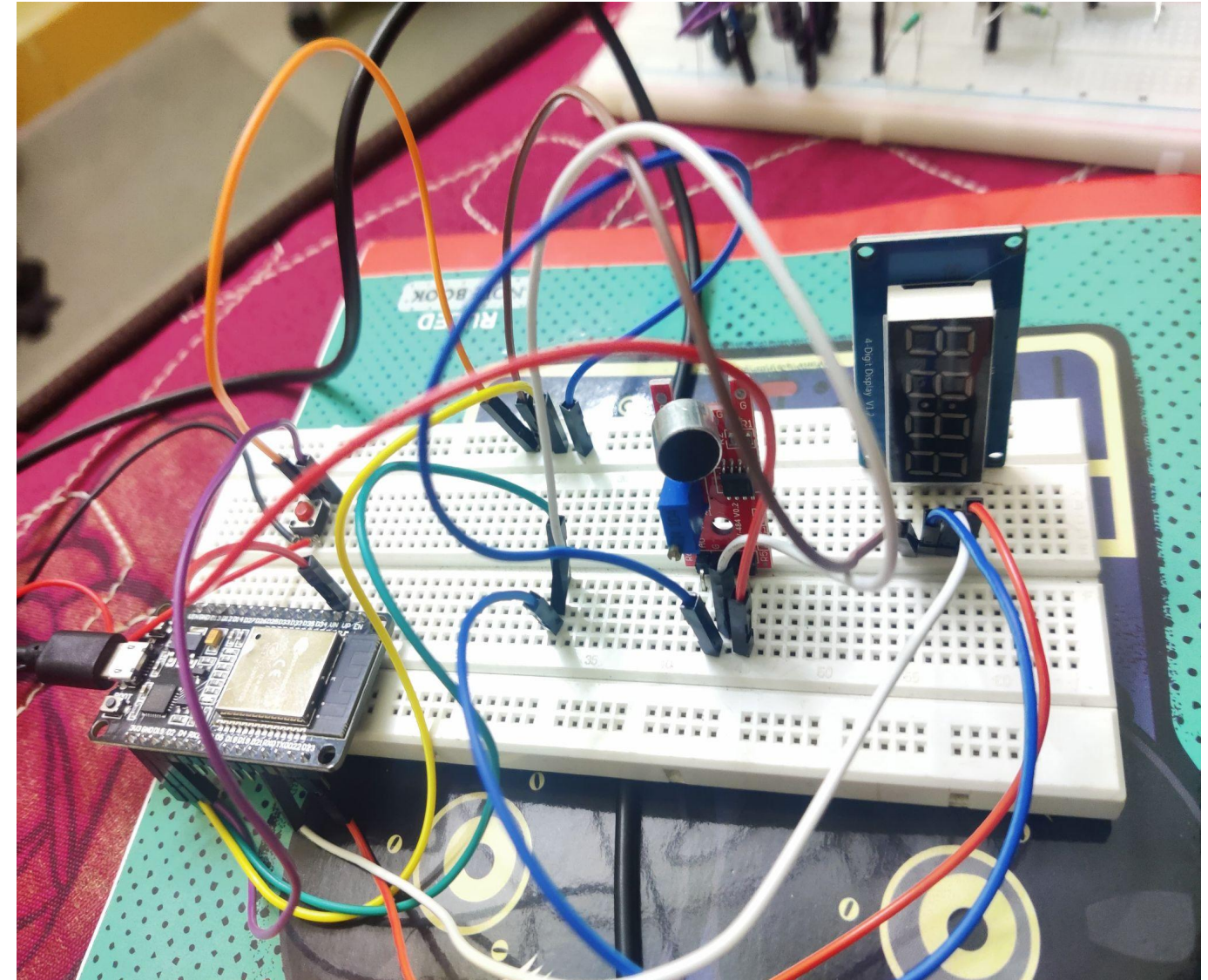
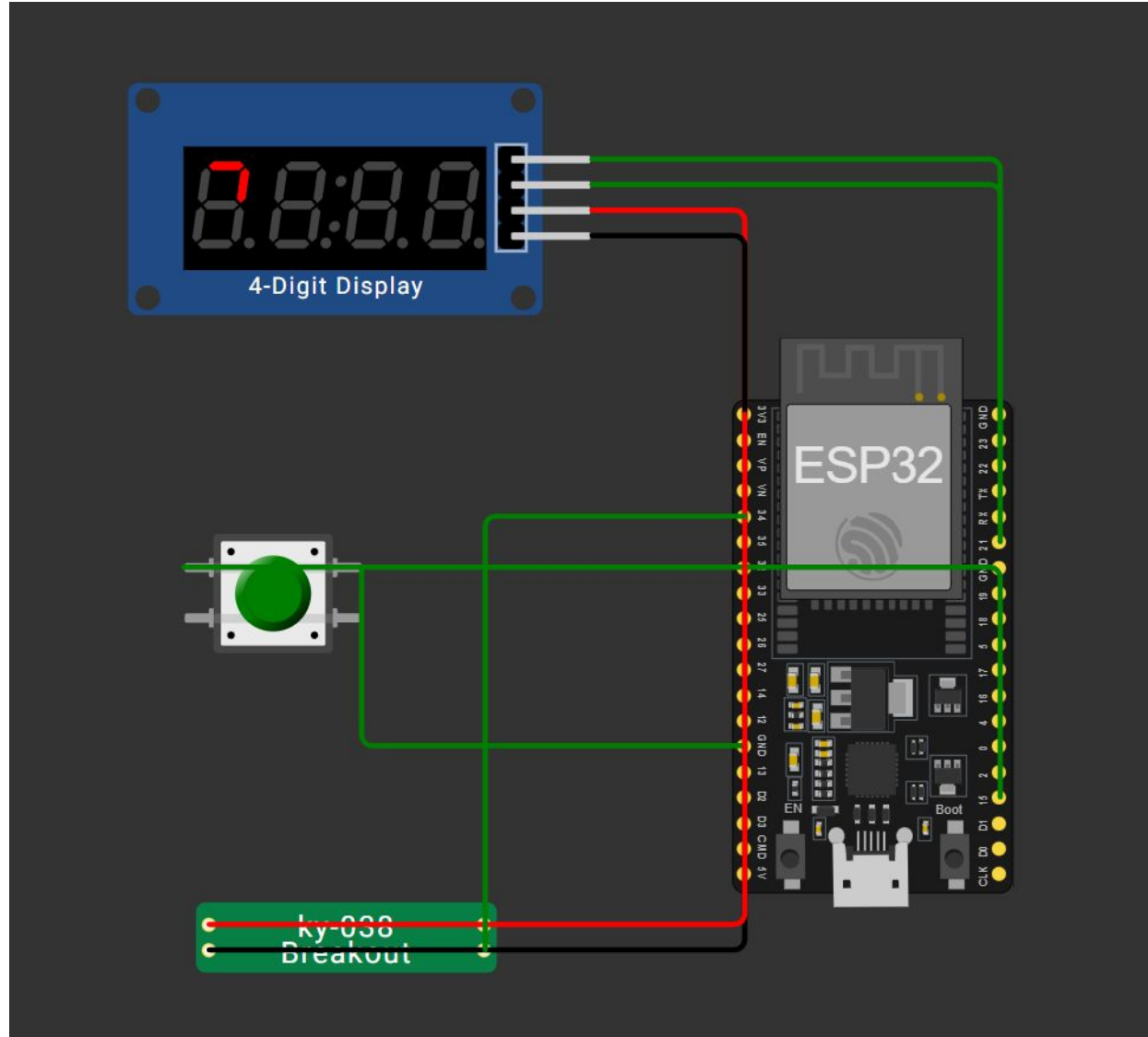
Scores are displayed on the 7 segment Display.

The system is USB powered for portability.





# CIRCUIT DIAGRAM





# Signal Processing Flow



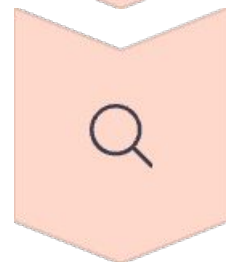
## Audio Capture

Sound sensor captures audio and sends an analog signal.



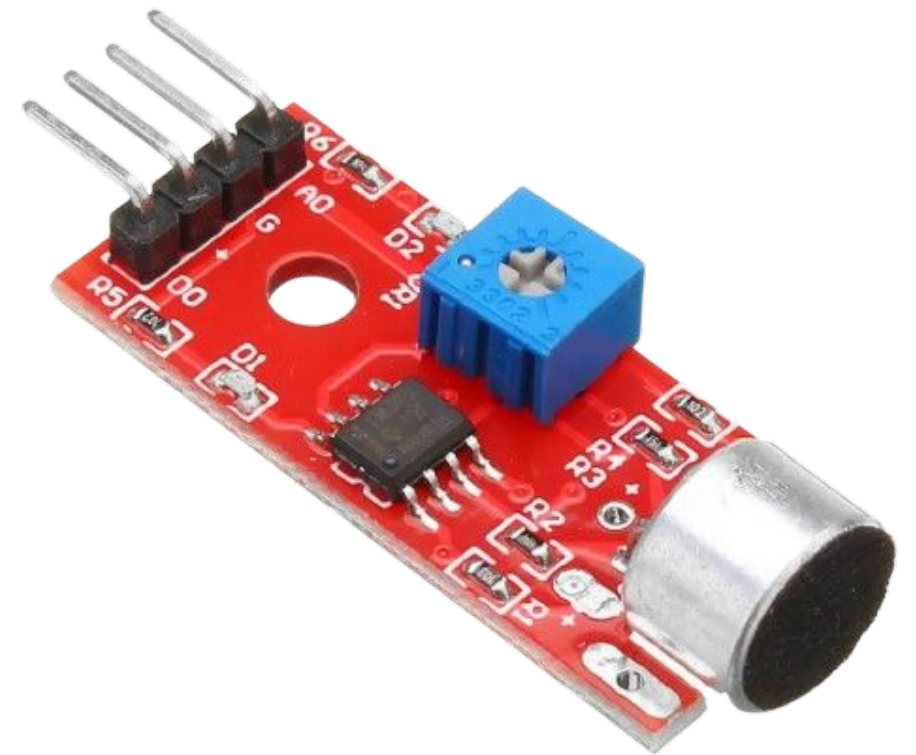
## Analog-to-Digital Conversion

ESP32 ADC converts the signal to digital.



## Clap Detection

Algorithm counts peaks within a time window.





# User Interface and Control

## Score Display

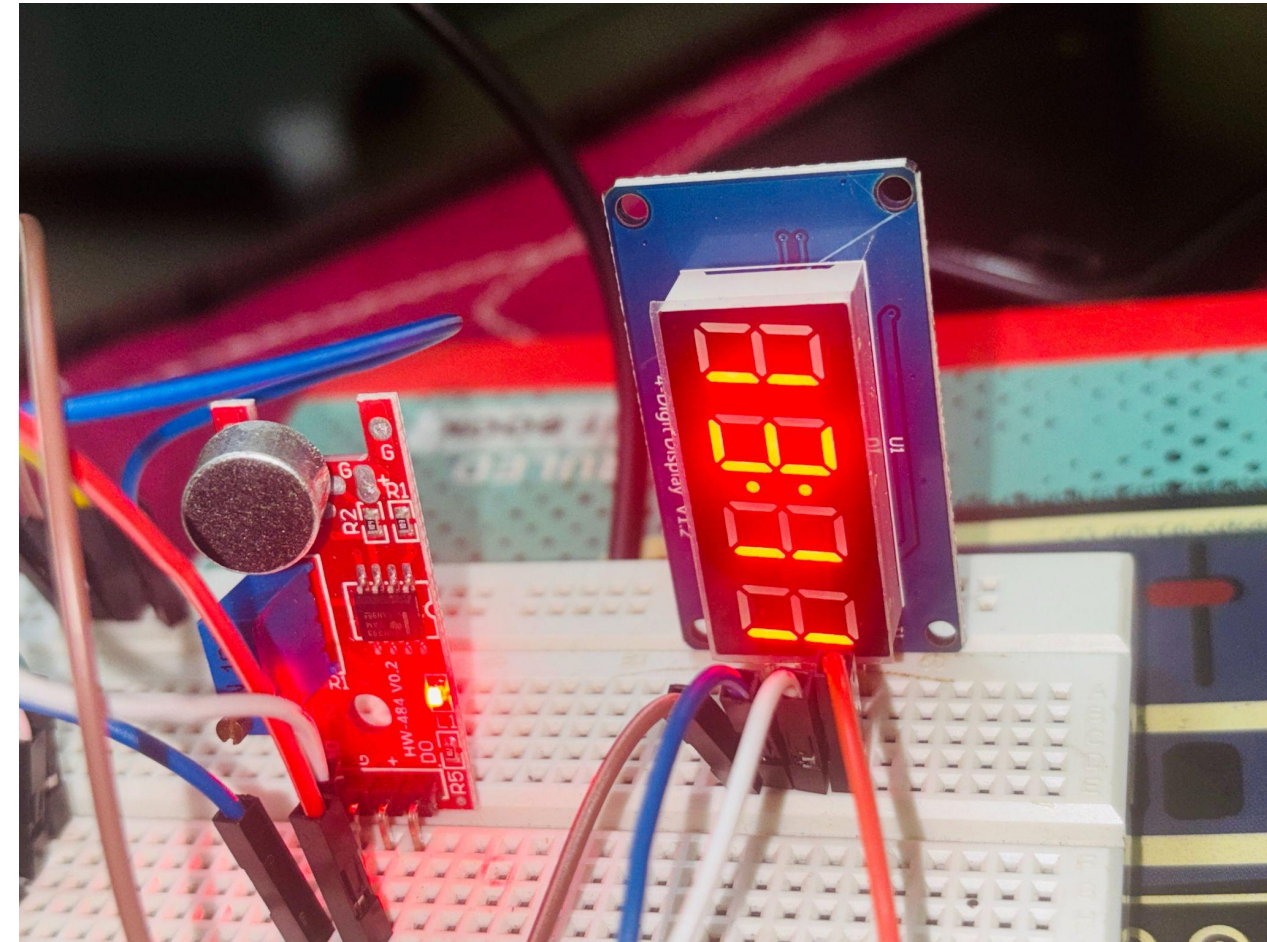
7 segment displays Team A and B scores.

## Clap Input

Clap detection increments scores accordingly.

## Reset

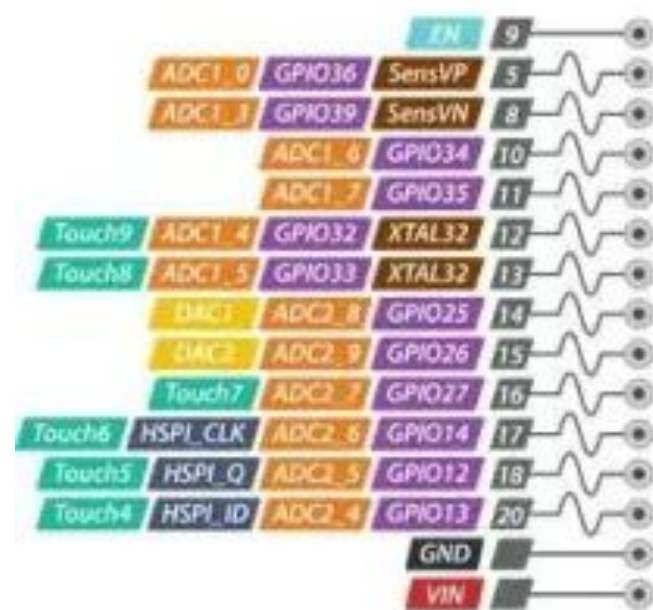
Button or voice command resets the scores.



# Hardware Components

## ESP32 DevKitC

The ESP32 processes audio and controls the 7 segment Display



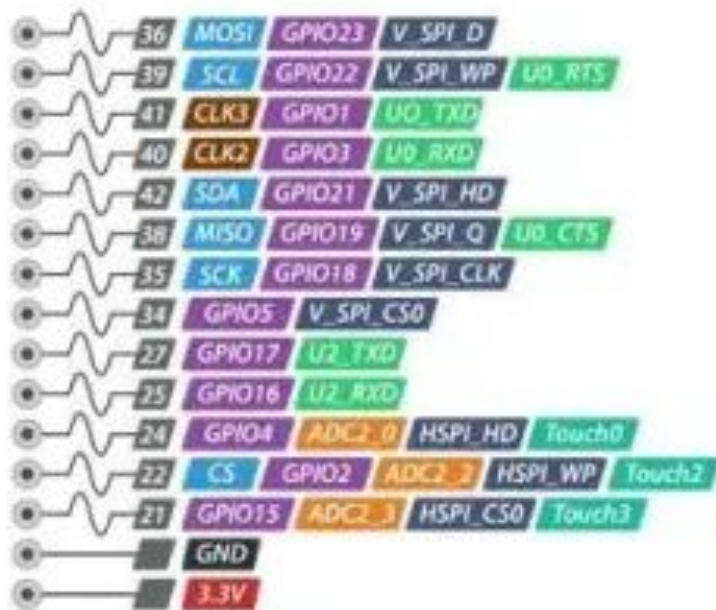
## KY 038 Sound Sensor

The KY 038 sound sensor captures audio input.



## LED Matrix

The 4 bit display visually displays the scores.



# Software Implementation



## Arduino IDE

The code is written in C++ using Arduino libraries.



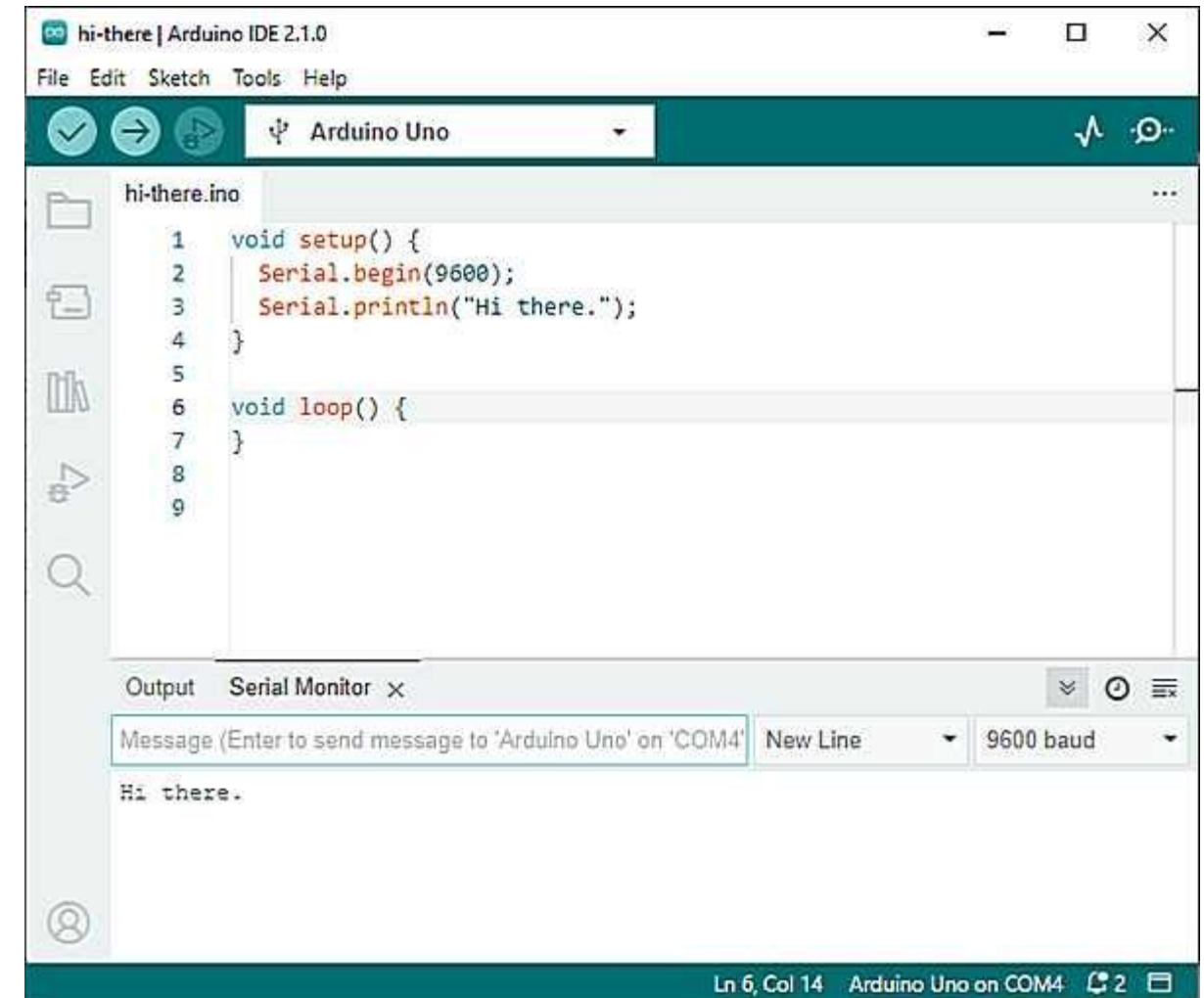
## ESP32 Libraries

Libraries manage ESP32 functions.



## Display Control

Functions update the display.





```

1  #include <TM1637Display.h>
2
3  // TM1637 Pins (CLK, DIO)
4  #define TM1637_CLK 22
5  #define TM1637_DIO 21
6  #define RESET_BUTTON 15    // Push button (connect to GND)
7
8  const int micPin = 34;    // KY-038 analog pin (adjust as needed)
9
10 int teamAScore = 0;
11 int teamBScore = 0;
12
13 int micState = LOW;
14 int lastMicState = LOW;
15
16 int clapCount = 0;
17 unsigned long firstClapTime = 0;
18 const unsigned long maxClapGap = 500;    // Max time between claps (ms)
19 const unsigned long minClapGap = 50;    // Min time to avoid false triggers
20
21 bool waiting = false;
22 bool alreadyProcessed = false;
23
24 TM1637Display display(TM1637_CLK, TM1637_DIO);    // Initialize TM1637
25
26 void setup() {
27     Serial.begin(115200);
28     pinMode(micPin, INPUT);
29     pinMode(RESET_BUTTON, INPUT_PULLUP);
30
31     // TM1637 Setup
32     display.setBrightness(7);    // Brightness (0-7)
33     display.showNumberDecEx(teamAScore * 100 + teamBScore, 0b01000000, true);    // Initial display: "00 00"
34 }
35
36 void loop() {
37     micState = digitalRead(micPin);    // Still using digital for now (adjust if needed)
38     unsigned long now = millis();
39     if (digitalRead(RESET_BUTTON) == LOW) {
40         delay(50);    // Debounce

```

```

        if (digitalRead(RESET_BUTTON) == LOW) {
            resetScores();
            while (digitalRead(RESET_BUTTON) == LOW);    // Wait for release
        }
    }
    // Rising edge detection (clap)
    if (micState == HIGH && lastMicState == LOW && (now - firstClapTime > minClapGap)) {
        clapCount++;
        Serial.println("👏 Clap detected!");
        if (clapCount == 1) {
            firstClapTime = now;
            waiting = true;
            alreadyProcessed = false;
        } else if (clapCount == 2 && waiting && !alreadyProcessed) {
            // Double clap → Team B +1
            teamBScore++;
            updateDisplay();
            Serial.print("Team A: "); Serial.print(teamAScore);
            Serial.print(" | Team B: "); Serial.println(teamBScore);

            alreadyProcessed = true;
            waiting = false;
            clapCount = 0;
        }
    }
    // Timeout for single clap
    if (waiting && (now - firstClapTime > maxClapGap)) {
        if (clapCount == 1 && !alreadyProcessed) {
            // Single clap → Team A +1
            teamAScore++;
            updateDisplay();
            Serial.print("Team A: "); Serial.print(teamAScore);
            Serial.print(" | Team B: "); Serial.println(teamBScore);
        }
        clapCount = 0;
        waiting = false;
        alreadyProcessed = false;
    }
}

```

```

        lastMicState = micState;
    }

    // Update TM1637 display (AA BB format)
    void updateDisplay() {
        int scoreDisplay = (teamAScore * 100) + teamBScore;    // Combines scores (e.g., 5-3 → "0503")
        display.showNumberDecEx(scoreDisplay, 0b01000000, true);    // Colon in middle: "AA:BB"
    }

    void resetScores() {
        teamAScore = 0;
        teamBScore = 0;
        updateDisplay();
        Serial.println("Scores reset to 00:00");
    }
}

```



# Display and Control Logic

## Display Control

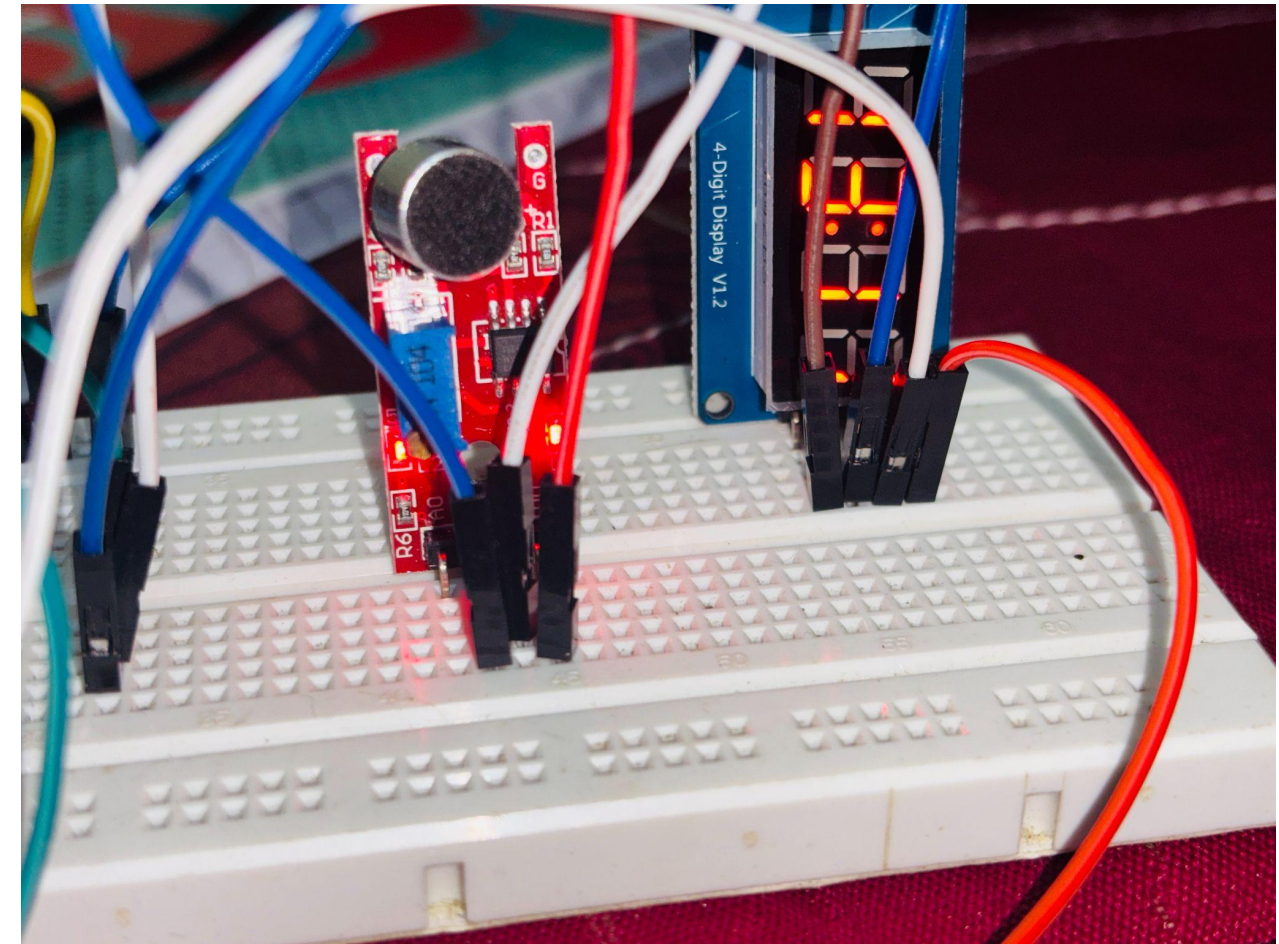
Set pixel colors for score display.

## Score Updating

Increment team scores.

## Resetting Scores

Set the scores to zero.



# Results and Performance

## 90%

Clap detection accuracy

## 0.5s

Average response time

The system achieved 90% accuracy in clap detection. The average response time from clap to display was 0.5 seconds. The system is sensitive to loud ambient noise.







# Conclusion and Future Work

## Future Enhancements

- Voice command recognition
- Wireless connectivity
- Improved noise filtering

The project successfully implemented a voice-controlled scoreboard.

Future enhancements include voice command recognition, wireless connectivity, and improved noise filtering.