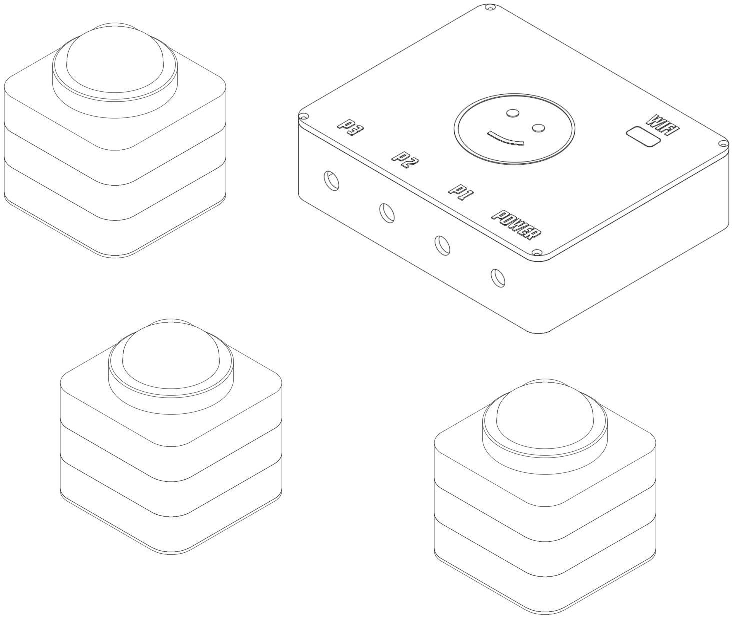
Buzzer System Manual

Table of Contents



1. Device Overview

The Buzzer System consists of three buzzers and a central control unit (the "brain"). The brain must be connected to power and the buzzers wired to it for the game setup.

2. Features

- Detect Player Buzzes
- Communicate with Game Controller
- Indicate Player Lock and Win States
- Flash LEDs for Winning Player
- Display WiFi Connection Status

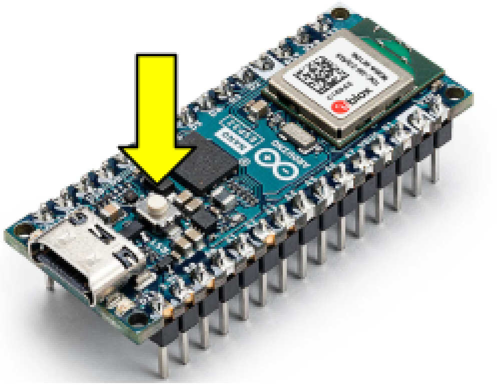
3. Setup Instructions

Initial Setup

1. Open the front cover of the brain to gain access to the internals.
2. Plug in USB-C to the Microcontroller in the brain.
3. Prepare/Open Code from Arduino IDE.
4. Open the Serial Monitor in the upper right corner.
5. Wait for power to connect to WiFi and record the IP Address of the Buzzer System. This will be important for connecting the devices wirelessly.
6. You will also need the IP Address from the device running Companion.

Update Software

1. Open the backplate of the device.
2. Plug in USB-C to the Microcontroller.
3. Prepare/Open Code from Arduino IDE.

4. Click the **Tools** setting, check **Pin Numbering**, ensure it is set to **By GPIO numbering (legacy).** *If code suddenly stops working check this first.*
5. Prepare Microcontroller for upload.
   - Press the white button on the microcontroller twice. The LED built on the board should begin flashing rainbow colors.
   - Press the white button once more; the microcontroller should pulse green.
   - 
6. Upload the code to the microcontroller.
   - Watch the output box on the bottom of the IDE wait until you get the DONE! Statement.
     -

4. Operation Instructions

Setting Up for Game

- Connect all three buzzers to the brain.
- Connect the brain to a power source.
- If the devices have powered down for a period of time, they may get a new IP Address when they reconnect to WiFi.
- When preparing for the show, if the buzzers and controller both are connected to WiFi but are not communicating, check that the IP Addresses match.
- The same applies if the messages are not getting to the companion app.

Player Wins

When a player wins:

1. The winning player's button press is detected by the Buzzer System.
2. An OSC (Open Sound Control) message is sent to the game controller to indicate the player's win.
3. An OSC message is also sent to the companion app.
4. The corresponding LED on the Buzzer System will flash with the team color of the winning player.

Locking a Player's Buzzer

To lock a player's buzzer:

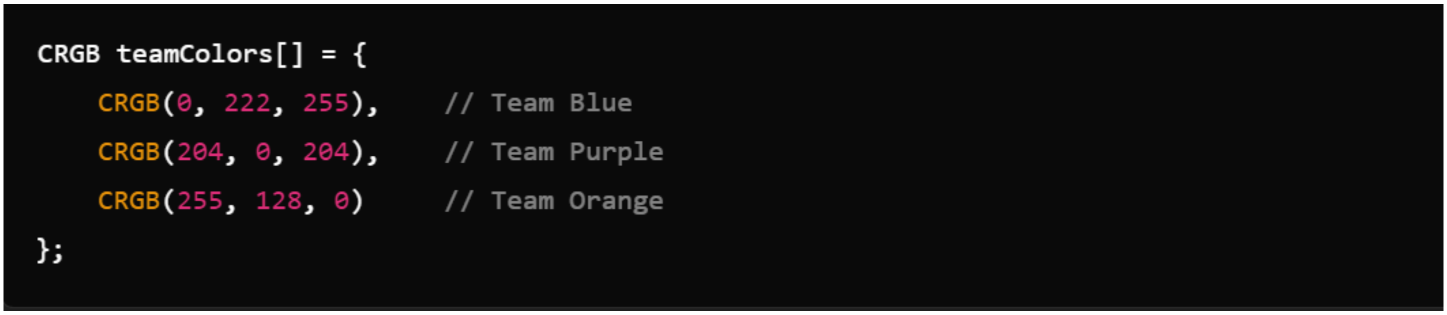
1. An OSC message is received from the game controller indicating the lock status.
2. The Buzzer System will toggle the lock status for that player.
3. The corresponding LED for that player's lock will change color:
   - Red if the buzzer is locked.
   - Team color if the buzzer is unlocked.
4. The lock status is also displayed on the Serial Monitor for verification.

5. Safe Code Modifications

You can safely change the following parts of the code to customize the Game Controller for your needs:

Team Colors

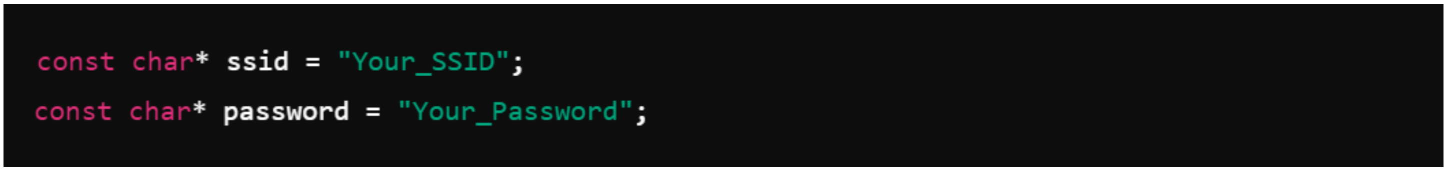
Modify the teamColors array to change the colors associated with each team:



```
CRGB teamColors[] = {
    CRGB(0, 222, 255),    // Team Blue
    CRGB(204, 0, 204),    // Team Purple
    CRGB(255, 128, 0)     // Team Orange
};
```

WiFi Information

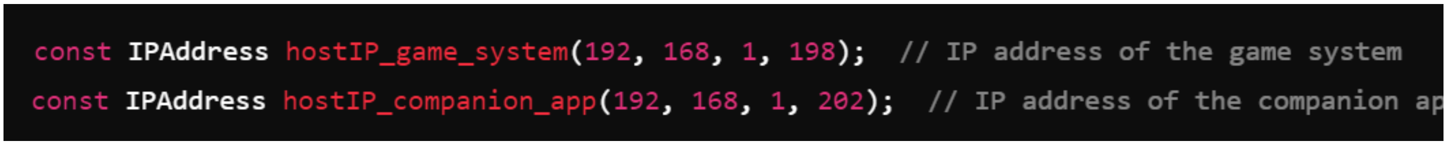
Update the WiFi SSID and password to match your network:



```
const char* ssid = "Your_SSID";
const char* password = "Your_Password";
```

IP Addresses for Companion and Buzzers

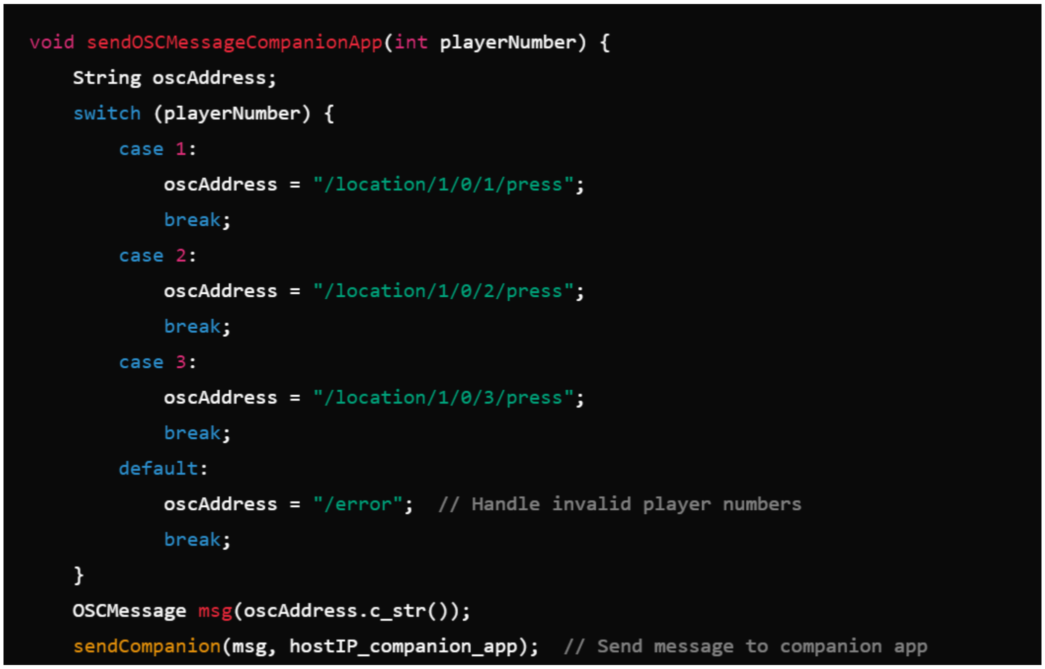
Change the IP addresses to match the devices in your network:



```
const IPAddress hostIP_game_system(192, 168, 1, 198);  // IP address of the game system
const IPAddress hostIP_companion_app(192, 168, 1, 202);  // IP address of the companion ap
```

Companion Button Locations

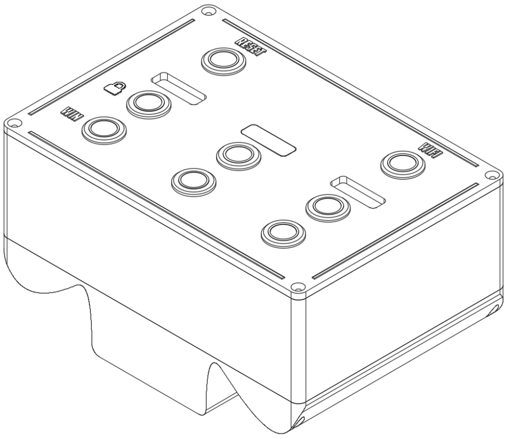
Update the OSC addresses for the companion app buttons:



```
void sendOSCMessageCompanionApp(int playerNumber) {
    String oscAddress;
    switch (playerNumber) {
        case 1:
            oscAddress = "/location/1/0/1/press";
            break;
        case 2:
            oscAddress = "/location/1/0/2/press";
            break;
        case 3:
            oscAddress = "/location/1/0/3/press";
            break;
        default:
            oscAddress = "/error";  // Handle invalid player numbers
            break;
    }
    OSCMessage msg(oscAddress.c_str());
    sendCompanion(msg, hostIP_companion_app);  // Send message to companion app
```

Game Controller Manual

Table of Contents



1. Device Overview

The Game Controller manages the core functions of the overall game system in a hands-free format. It enables the host to control key aspects of the game, such as resetting buzzers, locking players' buzzers, and forcing a player win state.

2. Features

- **Reset the Buzzers**
- **Lock Players' Buzzers**
- **Force a Player Win State**

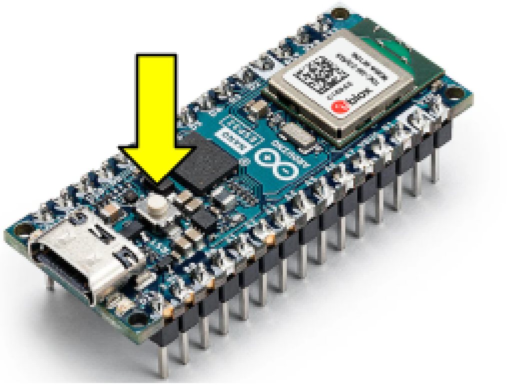
3. Setup Instructions

Initial Setup

1. Open the backplate of the device.
2. Plug in USB-C to the Microcontroller.
3. Prepare/Open Code from Arduino IDE.
4. Open the Serial Monitor in the upper right corner.
5. Wait for power to connect to WiFi and record the IP Address of the Game Controller. This will be important for connecting the devices wirelessly.
6. You will also need the IP Address from the device running Companion.

Update Software

1. Open the backplate of the device.
2. Plug in USB-C to the Microcontroller.
3. Prepare/Open Code from Arduino IDE.

4. Click the **Tools** setting, check **Pin Numbering**, ensure it is set to **By GPIO numbering (legacy).** *If code suddenly stops working check this first.*
5. Prepare Microcontroller for upload.
    - Press the white button on the microcontroller twice. The LED built on the board should begin flashing rainbow colors.
    - Press the white button once more; the microcontroller should pulse green.
    - 
6. Upload the code to the microcontroller.
    - Watch the output box on the bottom of the IDE wait until you get the DONE! Statement.

4. Operation Instructions

Setting Up for Game

- If the devices have powered down for a period of time, they may get a new IP Address when they reconnect to WiFi.
- When preparing for the show, if the buzzers and controller both are connected to WiFi but are not communicating, check that the IP Addresses match.
- The same applies if the messages are not getting to the companion app.

Player Wins

1. The winning player's button press is detected by the Game Controller.
2. An OSC (Open Sound Control) message is sent to the game system to indicate the player's win.
3. An OSC message is also sent to the companion app.
4. The corresponding LED on the Game Controller will light up with the team color of the winning player.

Locking a Player's Buzzer

1. Press the corresponding player lock button on the Game Controller.
2. The Game Controller will toggle the lock status for that player.
3. An OSC message is sent to the game system to indicate the player's lock status.
4. The corresponding LED for that player's lock will change color:
    - Green if the buzzer is unlocked.
    - Red if the buzzer is locked.
5. The lock status is also displayed on the Serial Monitor for verification.

Forcing a Player Win State

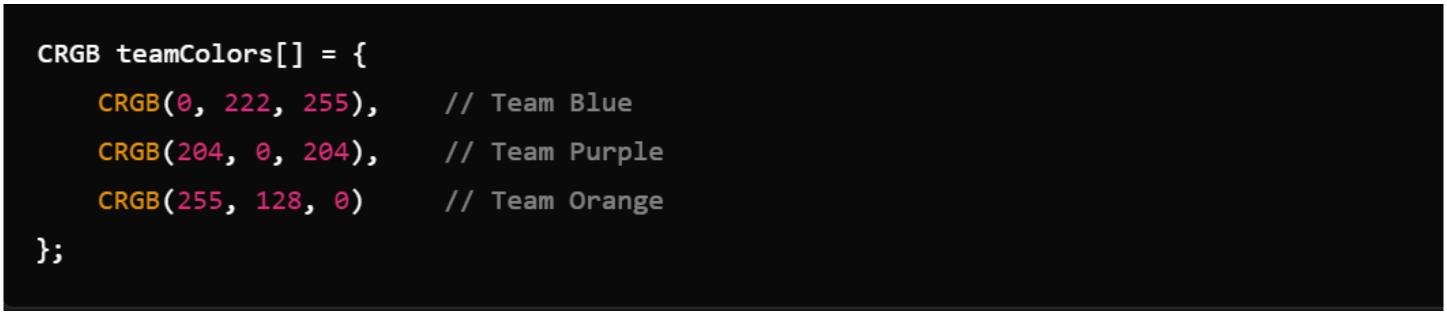
1. Press the corresponding player win button on the Game Controller.
2. The Game Controller sends an OSC message to the game system indicating that the player has won.
3. An OSC message is also sent to the companion app to update the win status.
4. The corresponding LED on the Game Controller will light up with the team color of the winning player.
5. The win state is displayed on the Serial Monitor for verification.

5. Safe Code Modifications

You can safely change the following parts of the code to customize the Game Controller for your needs:

Team Colors

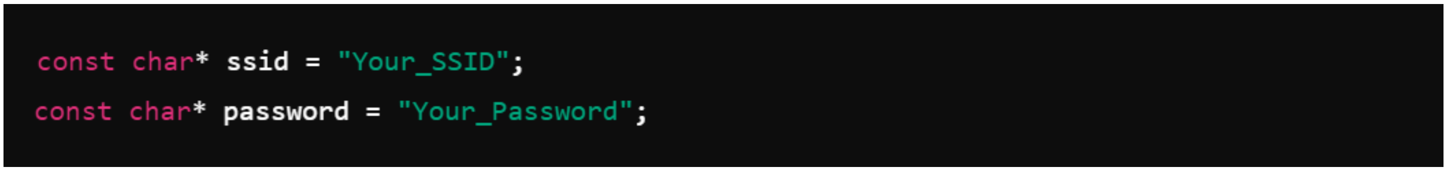
Modify the teamColors array to change the colors associated with each team:



```
CRGB teamColors[] = {
    CRGB(0, 222, 255),    // Team Blue
    CRGB(204, 0, 204),    // Team Purple
    CRGB(255, 128, 0)     // Team Orange
};
```

WiFi Information

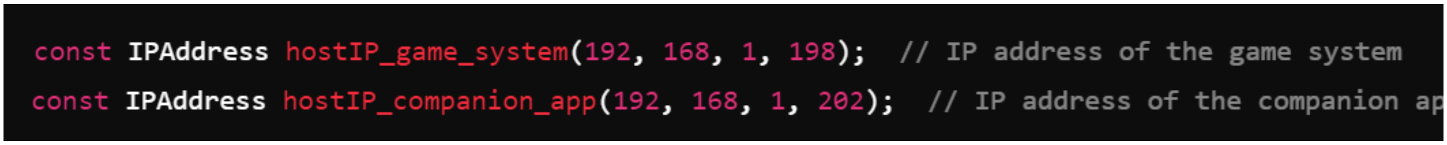
Update the WiFi SSID and password to match your network:



```
const char* ssid = "Your_SSID";
const char* password = "Your_Password";
```

IP Addresses for Companion and Buzzers

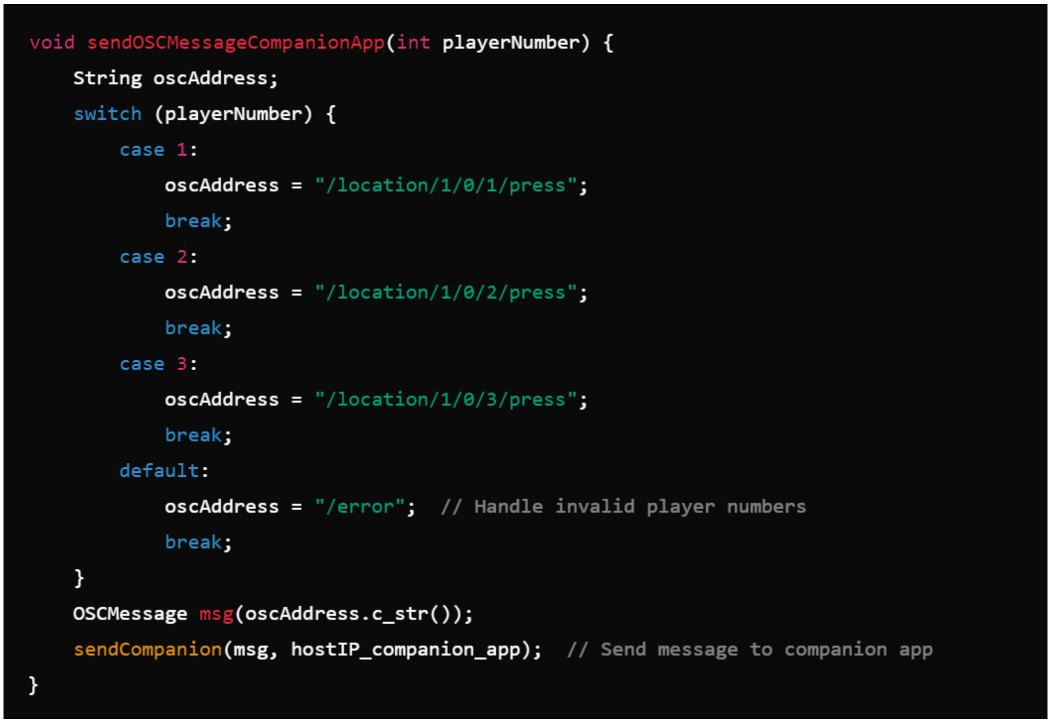
Change the IP addresses to match the devices in your network:



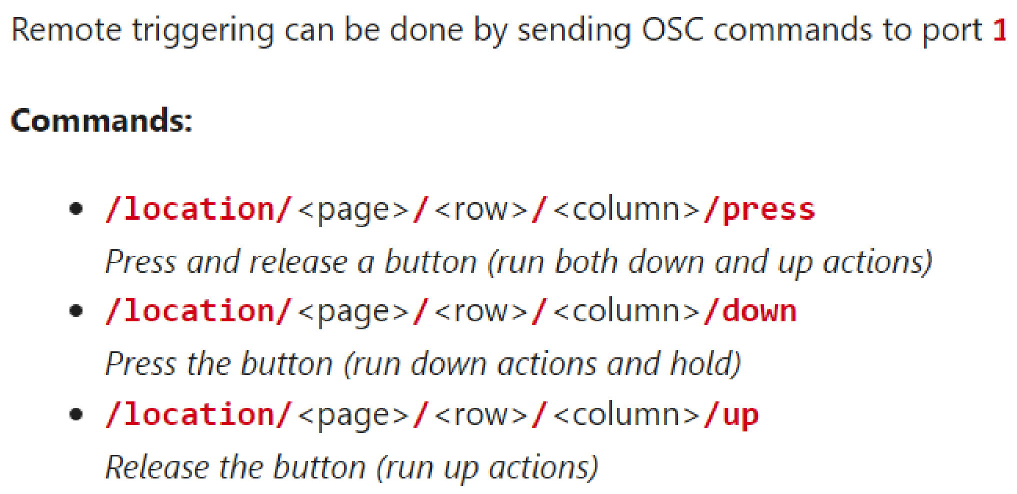
```
const IPAddress hostIP_game_system(192, 168, 1, 198);  // IP address of the game system
const IPAddress hostIP_companion_app(192, 168, 1, 202);  // IP address of the companion ap
```

Companion Button Locations

Update the OSC addresses for the companion app buttons:



```
void sendOSCMessageCompanionApp(int playerNumber) {
    String oscAddress;
    switch (playerNumber) {
        case 1:
            oscAddress = "/location/1/0/1/press";
            break;
        case 2:
            oscAddress = "/location/1/0/2/press";
            break;
        case 3:
            oscAddress = "/location/1/0/3/press";
            break;
        default:
            oscAddress = "/error";  // Handle invalid player numbers
            break;
    }
    OSCMessage msg(oscAddress.c_str());
    sendCompanion(msg, hostIP_companion_app);  // Send message to companion app
}
```



Remote triggering can be done by sending OSC commands to port `1`

Commands:

- `/location/`<page>`/`<row>`/`<column>`/press`
  Press and release a button (run both down and up actions)
- `/location/`<page>`/`<row>`/`<column>`/down`
  Press the button (run down actions and hold)
- `/location/`<page>`/`<row>`/`<column>`/up`
  Release the button (run up actions)

6. Troubleshooting

Common Issues and Solutions

- **Controller is not Communicating to Buzzer System:**
  - Check the IP Address of the Buzzer System is correct in the Game Controller Software.

LED Indicators

- **WiFi Indicator:**
  - Green - WiFi Connected
  - Red - Disconnected