Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)

ESCAPE PUZZLE BOX

The domain of the Project:

Embedded Systems and IoT

Mentor:

MEHAK MAJEED

(Junior Engineer, ATFAAL Innovations)

By

Ms. KURUKUTI LEHA PRIYA

(B.Tech 4th year pursuing)

Period of the project

November 2025 to December 2025

SURE ProEd

Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)

Declaration

We, the undersigned, hereby declare that the project entitled "**ESCAPE PUZZLE BOX**" has been carried out by us under the mentorship of MEHAK MAJEED and with the support of SURE ProEd during the period June 2025 to December 2025.

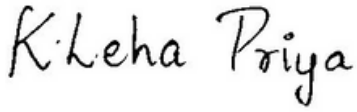
This project has been undertaken for the benefit of gaining hands-on experience in industry-relevant technologies, enhancing our practical knowledge in IoT, embedded systems, and access control solutions, and preparing us for prospective employment opportunities.

We further declare that this work is a result of our genuine effort and has not been copied or reproduced from any other source.

Name:

Kurukuti Leha Priya

Signature:



Name:

Mehak Majeed

Signature:

Seal & Signature:
Prof. Radhakumari
Executive Director & Founder
SURE ProEd

Table of Contents

1. Executive Summary
2. Introduction
3. System Design
4. Project Objectives
5. Methodology & Results
6. Social / Industry Relevance of the Project
7. Learning & Reflection
8. Main Code and File Structure
9. Future Scope & Conclusion

Executive Summary

The Escape Puzzle Box project implements a secure, interactive escape-room mechanism by integrating an ESP32 and an Arduino through UART communication, combining real-time multiplayer game logic with physical actuation. The system is designed to control a solenoid lock and a servo-driven box lid, providing a realistic and automated unlocking experience upon successful puzzle completion.

In this design, the ESP32 acts as the central controller responsible for game logic, player management, Wi-Fi connectivity, and the web-based multiplayer interface. A secret access code is set locally using a keypad and displayed on an I2C LCD. Players connect to the ESP32 via Wi-Fi and authenticate using the correct code through a WebSocket-based interface, ensuring real-time communication and turn-based gameplay for two to four players.

Once authenticated, players participate in a riddle-solving challenge managed entirely by the ESP32. The system enforces turn order, validates answers, and prevents unauthorized actions. When a player submits the correct answer, the ESP32 declares the winner on the LCD, activates a status LED, and transmits an "**OPEN**" command via UART to the Arduino.

The Arduino functions as the actuator controller, receiving commands over the serial interface. Upon receiving the "OPEN" command, it energizes a relay-controlled solenoid to unlock the box and actuates a servo motor to open the lid after a predefined delay. This clear separation of responsibilities ensures reliable hardware control while keeping complex networking and game logic on the ESP32.

Introduction

Background and Context

Escape rooms and puzzle-based games have gained popularity as interactive learning and entertainment platforms that encourage logical thinking, teamwork, and problem-solving. With advancements in embedded systems and IoT, traditional mechanical puzzle boxes can be enhanced using microcontrollers, wireless communication, and real-time control to create smarter and more engaging experiences. The Escape Puzzle Box project is developed in this context to demonstrate the integration of software-driven game logic with physical locking mechanisms.

Problem Statement

The primary goal of this project is to design a secure and interactive puzzle box that unlocks only when players successfully solve a predefined challenge. The system aims to support multiplayer participation, enforce fair turn-based gameplay, and ensure reliable physical actuation of the lock and lid. Another key objective is to establish robust communication between multiple users and embedded hardware while maintaining simplicity and reliability in design.

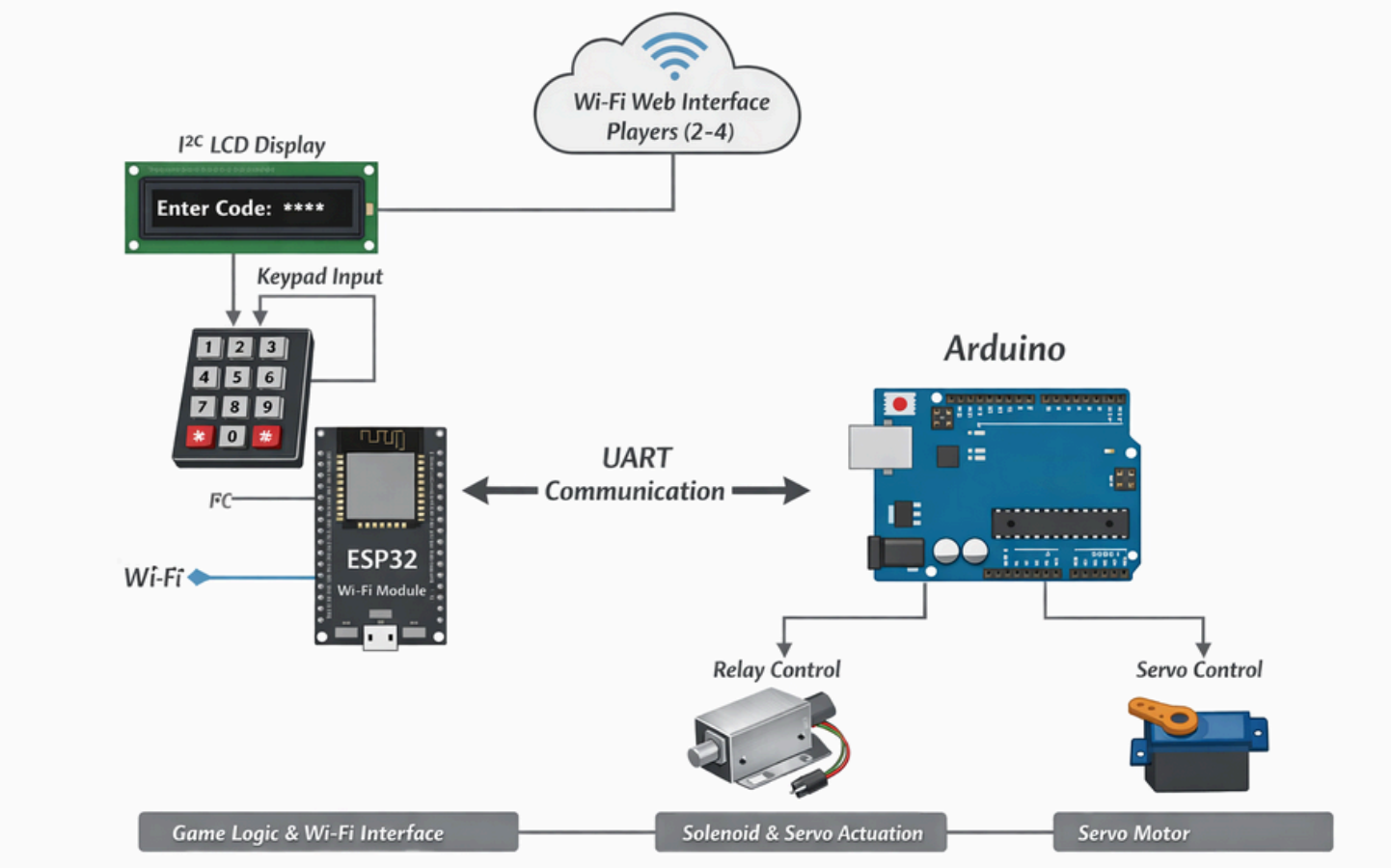
Scope of the Project

- The project supports a multiplayer escape puzzle system where 2 to 4 players can participate simultaneously using a Wi-Fi–based web interface.
- It includes secure code entry and validation using a keypad and LCD, ensuring controlled access to the game.
- The system demonstrates UART-based communication between ESP32 and Arduino for reliable control of a solenoid lock and servo motor.
- It integrates software-driven game logic with physical actuation, providing real-time visual and mechanical feedback upon puzzle completion.

Innovation Component

The innovation of this project lies in the seamless integration of real-time multiplayer game logic, UART-based inter-microcontroller communication, and electromechanical actuation within a compact system. By separating game control (ESP32) and hardware actuation (Arduino), the design improves reliability and modularity. The use of WebSocket-based interaction combined with physical feedback through solenoid and servo mechanisms provides a novel blend of digital gaming and tangible real-world interaction.

System Design



Project Objectives

Defined Objectives and Goals of the Project

1. Design and Development Objectives
   - To design a smart escape puzzle box that integrates both software intelligence and physical actuation.
   - To implement a modular architecture using ESP32 for game logic and Arduino for hardware control.
   - To ensure reliable UART communication between the two microcontrollers.
2. Functional Objectives
   - To enable secure access control through keypad-based code entry and validation.
   - To support multiplayer participation (2–4 players) using a Wi-Fi–based web interface.
   - To implement turn-based riddle gameplay, allowing only the active player to answer.
3. Hardware Control Objectives
   - To control a solenoid lock for secure locking and unlocking of the box.
   - To operate a servo motor for smooth and controlled lid opening.
   - To provide visual feedback using LCD messages and LED indicators.
4. Communication and Networking Objectives
   - To establish a local Wi-Fi network using the ESP32 for player connectivity.
   - To enable real-time interaction between players and the system using WebSocket communication.
   - To synchronize game state updates across all connected players.
5. User Experience Objectives
   - To create an interactive and engaging gameplay experience combining digital and physical elements.
   - To provide clear on-screen instructions and feedback through the LCD and web interface.

6. Learning and Application Goals

- To demonstrate practical application of embedded systems, IoT, and real-time control concepts.
- To enhance understanding of microcontroller integration, serial communication, and actuator control.
- To develop a prototype suitable for academic demonstration and further expansion into advanced escape-room systems.

Expected Outcomes and Deliverables

1. Functional Prototype
   - A working escape puzzle box integrating ESP32, Arduino, keypad, LCD, solenoid, servo, and LEDs.
   - Full interaction between hardware and web interface.
2. Game Logic Implementation
   - Multiplayer riddle-solving mechanism with code authentication and turn-based answer submission.
   - Real-time status updates for players on the LCD and via WebSocket messages.
3. Hardware Actuation
   - Solenoid lock reliably triggered via UART command from ESP32.
   - Servo motor opens the box lid automatically upon correct answer.
4. Documentation and Reporting
   - Complete project report including block diagrams, system diagrams, code snippets, and user manual.
   - Executive summary, objectives, scope, limitations, and expected outcomes for academic evaluation.
5. User Experience Features
   - Web-based interface for multiple players.
   - Clear visual and mechanical feedback signaling game progress and winner.

Methods and Technology Used

1. Hardware Components

- **ESP32 Microcontroller:** Acts as the central controller for game logic, Wi-Fi connectivity, and WebSocket communication.
- **Arduino Board:** Controls physical actuators (solenoid lock and servo motor) via UART commands from the ESP32.
- **Solenoid Lock:** Provides secure mechanical locking and unlocking of the puzzle box.
- **Servo Motor:** Opens the box lid smoothly upon correct answer.
- **I2C LCD Display:** Shows prompts, player status, and winner information.
- **4x4 Keypad:** Used for code entry and local game control.
- **LED Indicators:** Visual feedback for game status.

2. Communication Protocols

- **UART (Serial Communication):** Connects ESP32 and Arduino for sending actuation commands.
- **WebSocket over Wi-Fi:** Enables real-time, bidirectional multiplayer communication through a browser interface.

3. Software Technologies

- **Arduino IDE Programming:** For programming both ESP32 and Arduino.
- **Embedded C/C++:** Core programming language for microcontroller firmware.
- **AsyncWebServer and AsyncTCP Libraries:** For handling WebSocket connections and real-time communication on ESP32.
- **Keypad and LiquidCrystal_I2C Libraries:** To manage keypad input and LCD display interactions.

4. Methods and Design Approach

- **Modular Design:** Separates game logic (ESP32) from actuator control (Arduino) for reliability and scalability.
- **Real-Time Multiplayer Game Logic:** Handles multiple players, turn-based riddles, and code validation.
- **User Interaction Flow:** Players set the code, join via web interface, receive riddles, and actuate hardware upon correct answers.
- **Testing and Iteration:** Components individually tested (solenoid, servo, keypad, Wi-Fi connection) and integrated sequentially to ensure robust system performance.

This combination of hardware, software, and communication protocols ensures a reliable, interactive, and engaging escape puzzle box experience.

Project Architecture

The architecture of the Escape Puzzle Box follows a layered and modular design, ensuring clear separation of functionality and reliable system operation.

1. Hardware Architecture

- The system uses two microcontrollers:
  - ESP32 as the main controller.
  - Arduino as the actuator controller.
- Peripheral components such as keypad, LCD, LED indicators, solenoid lock, and servo motor are connected based on their functional roles.
- The solenoid and servo are isolated from the ESP32 and handled by the Arduino to avoid timing and power-related issues.

2. Control and Processing Layer

- ESP32 (Game Logic Controller):
  - Handles code entry and storage.
  - Manages player connections (2–4 players).
  - Implements riddle logic and turn-based gameplay.
  - Updates LCD and LED status.
  - Sends actuation commands to Arduino via UART.
- Arduino (Actuator Controller):
  - Continuously listens for serial commands from ESP32.
  - Activates the solenoid lock to unlock the box.
  - Rotates the servo motor to open the box lid.

3. Communication Architecture

- Wi-Fi (SoftAP Mode):
  - ESP32 creates a local Wi-Fi network.
  - Players connect using a web browser.
- WebSocket Protocol:
  - Enables real-time, bidirectional communication.
  - Used for sending codes, riddles, answers, and game status.
- UART Serial Communication:
  - Used for reliable command transfer between ESP32 and Arduino (e.g., “OPEN”).

4. User Interaction Layer

- Local Interaction:
  - Keypad for entering the secret game code.
  - LCD for displaying instructions, game status, and winner details.
- Remote Interaction:
  - Web interface for players to join, receive riddles, and submit answers.

- Feedback Mechanisms:
  - LED indicators for game events.
  - Physical unlocking and lid opening via solenoid and servo.

5. Overall Architectural Flow

1. Code is set via keypad → displayed on LCD.
2. Players join through web interface → authenticated by ESP32.
3. ESP32 controls game flow and turn order.
4. Correct answer → ESP32 sends UART command to Arduino.
5. Arduino unlocks solenoid and opens lid using servo.
6. Winner is displayed → game concludes.

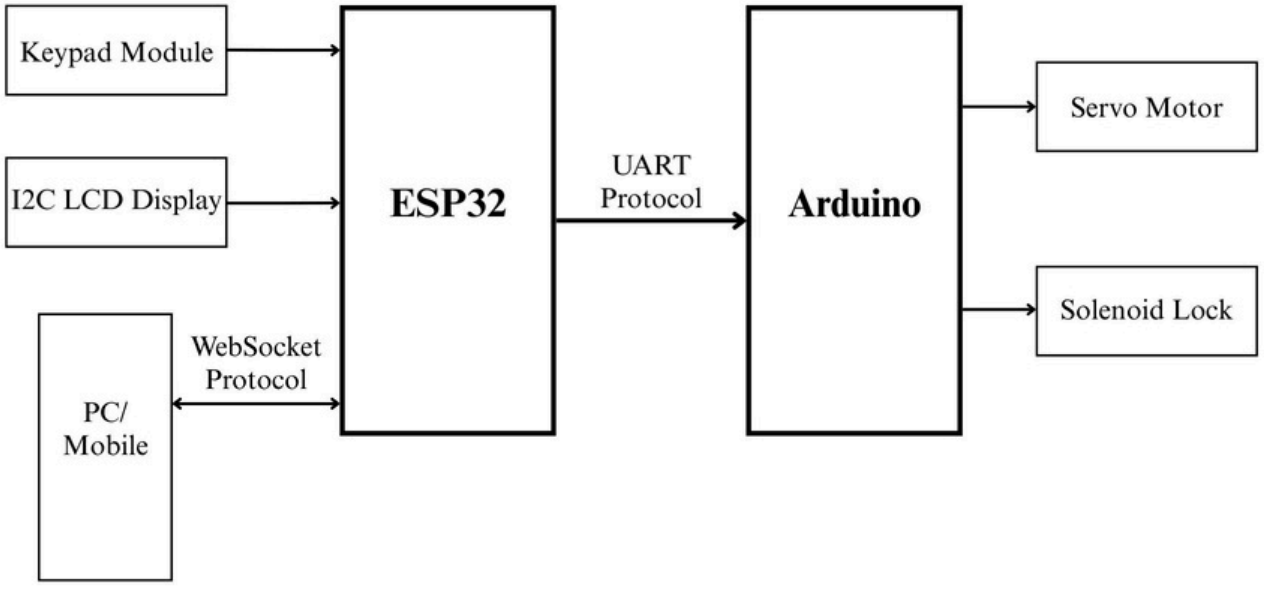
Results

The successful execution of the Escape Puzzle Box project resulted in the following outcomes:

1. **Functional Working Prototype**
   - A fully operational puzzle box capable of controlled locking and unlocking.
   - Seamless integration of ESP32, Arduino, and peripheral components.
2. **Reliable Multiplayer Interaction**
   - Successful implementation of real-time WebSocket communication.
   - Support for 2 to 4 players with proper authentication and turn-based gameplay.
3. **Accurate Hardware Actuation**
   - Solenoid lock activated correctly upon successful puzzle completion.
   - Servo motor reliably opened the box lid with controlled timing.
4. **Secure and Controlled Access**
   - Code-based authentication prevented unauthorized access.
   - Game logic ensured fairness by restricting actions to the active player only.

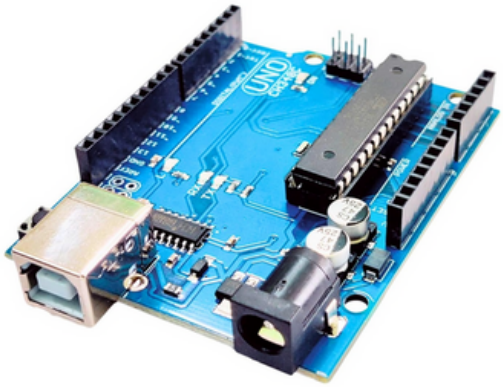
5. **Clear User Feedback**
   - LCD displayed real-time messages such as code status, player turns, and winner information.
   - LEDs provided instant visual confirmation of game success.
6. **Demonstration of Industry-Relevant Skills**
   - Practical application of embedded C/C++, UART communication, IoT networking, and hardware integration.
   - Experience gained in designing modular, scalable embedded systems.

Block Diagram

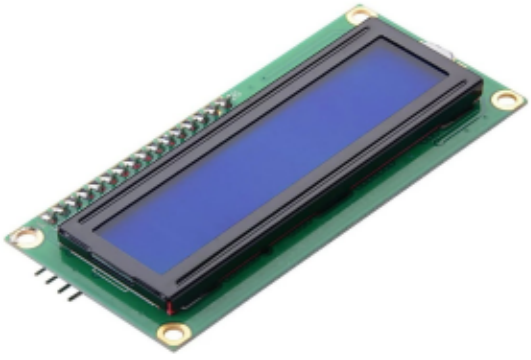


Components

ESP32



Arduino



I2C LCD Display

Keypad Module

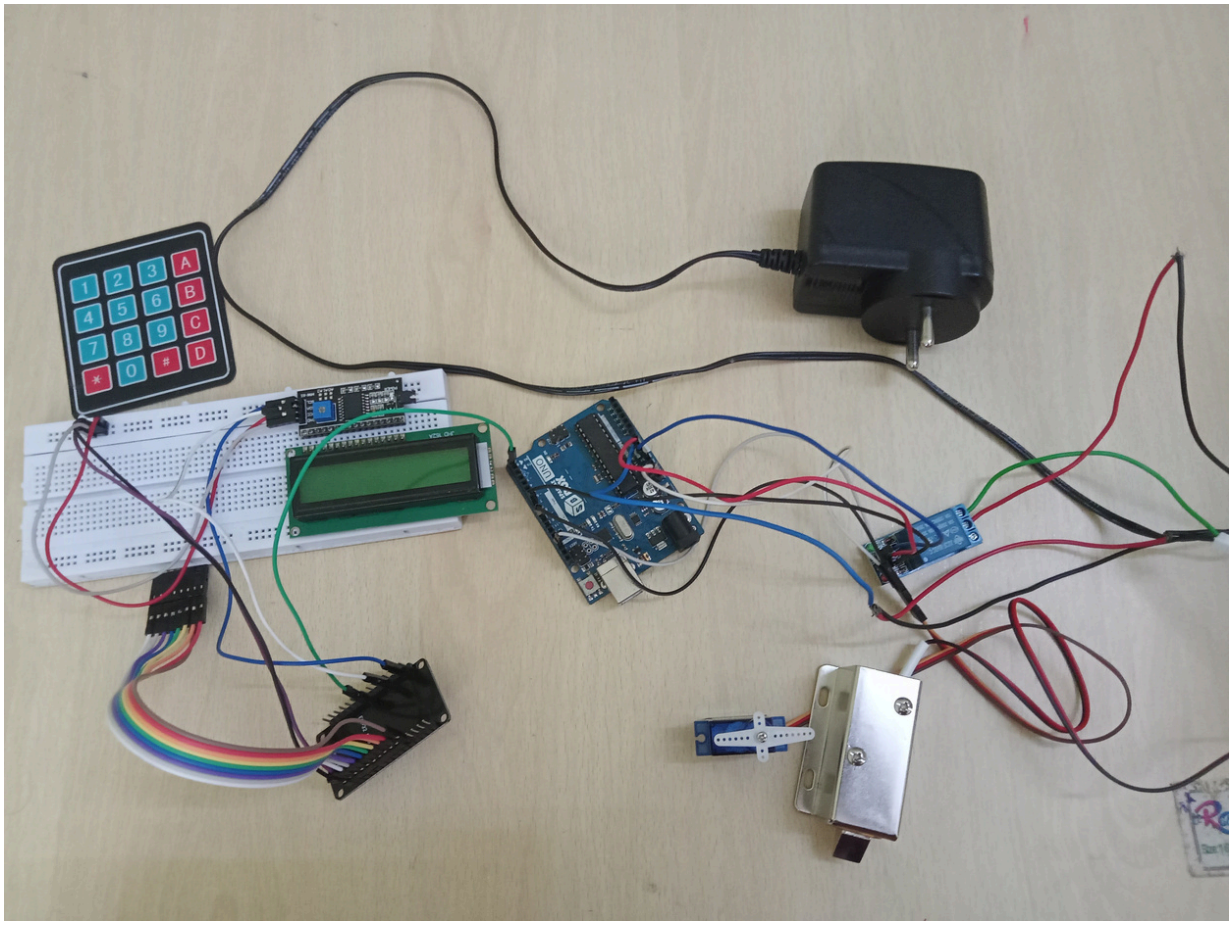
Solenoid Lock

Relay Module

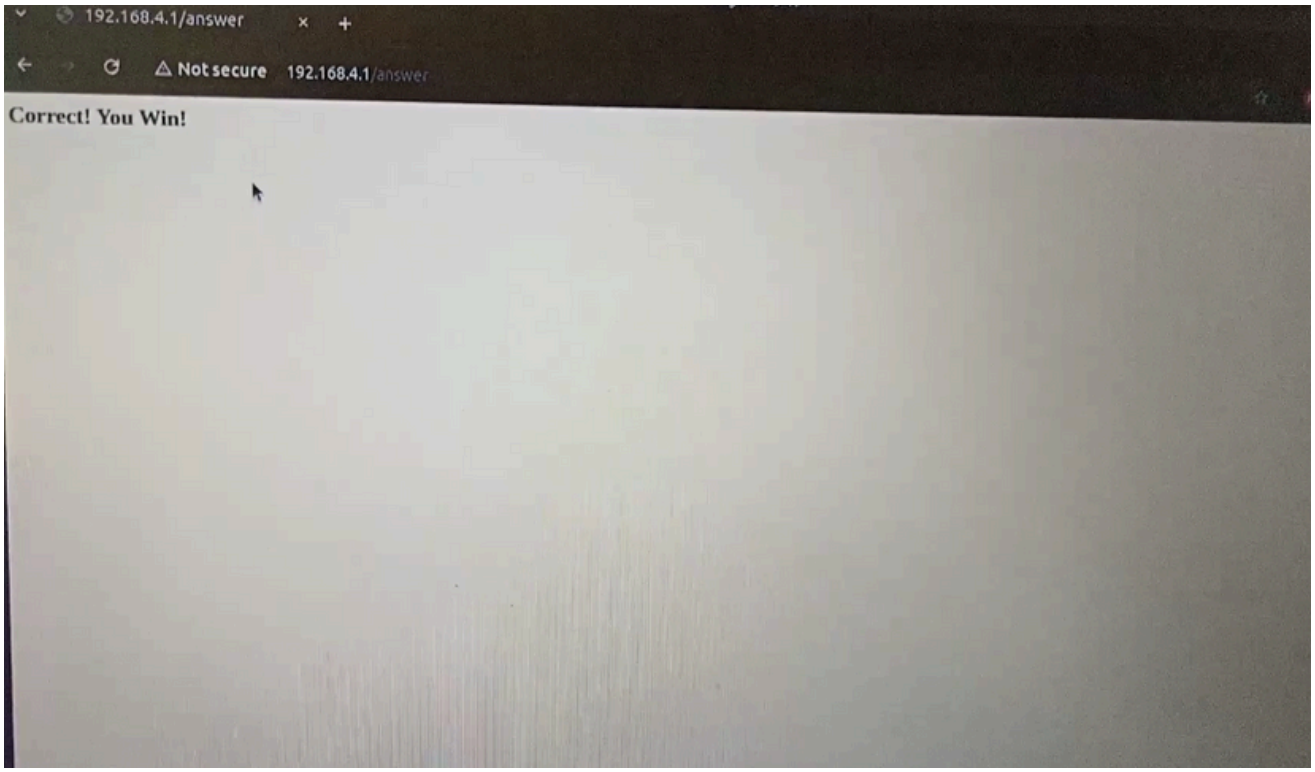
Servo Motor

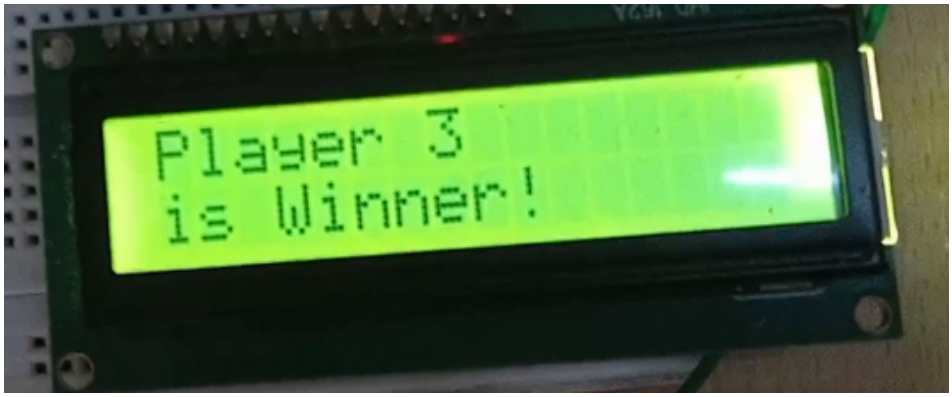
12V Adapter

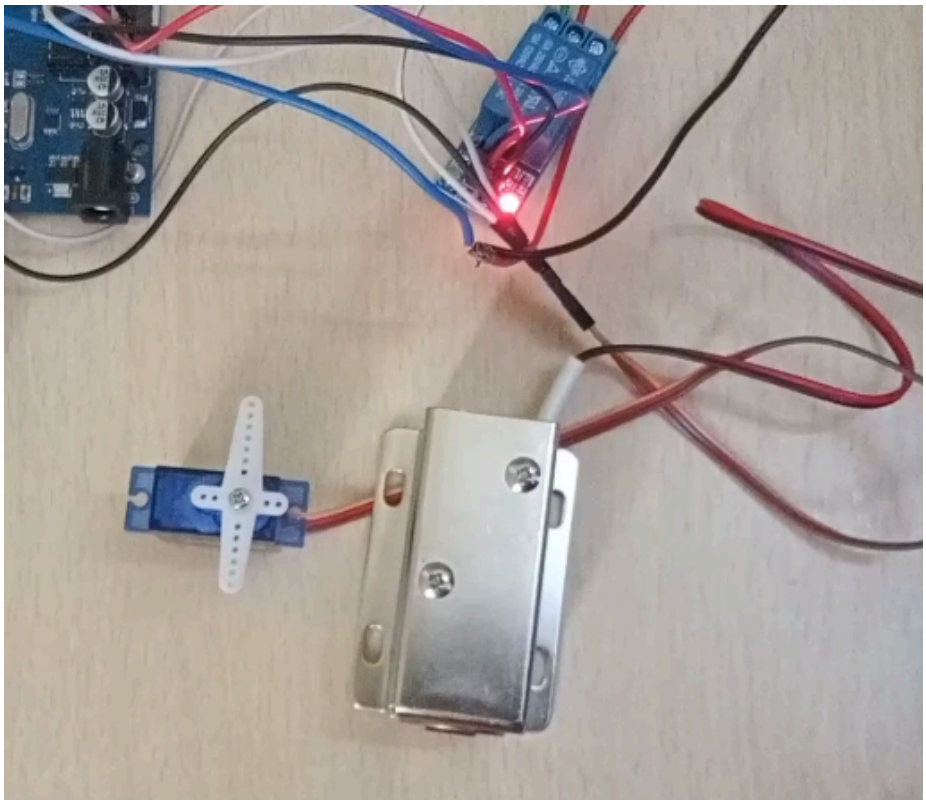
Circuit Diagram



Output


192.168.4.1/answer
Not secure 192.168.4.1/answer
Correct! You Win!


Player 3
is Winner!



Social and Industry Relevance of the Project

The Escape Puzzle Box project holds significant relevance both socially and within the industry due to its combination of interactive gameplay, embedded systems, and IoT technologies. Its impact spans education, technology innovation, and commercial applications.

1. Educational and Skill Development Applications

- **Enhancing Cognitive Skills:** The turn-based riddle-solving gameplay stimulates logical reasoning, critical thinking, and problem-solving abilities. Players must analyze clues, plan their moves, and anticipate outcomes, making it an effective cognitive training tool.
- **Hands-On STEM Learning:** By combining microcontrollers (ESP32, Arduino), sensors (keypad, LCD), actuators (servo and solenoid), and wireless communication, the project offers a practical, tangible example of STEM concepts in action. Students can learn about embedded system design, real-time control, and human–machine interaction.
- **Teamwork and Collaboration:** The multiplayer setup encourages collaboration and communication, teaching participants the value of coordinated problem-solving. This is especially relevant in educational settings and workshops where team-based learning is emphasized.
- **Inspiring DIY and Maker Culture:** The project motivates students and hobbyists to explore electronics, programming, and IoT design in a creative and enjoyable manner.

2. Industry and Commercial Relevance

- **Smart Lock and Security Systems:** The integration of a solenoid lock and servo motor controlled via microcontrollers showcases principles applicable in automated security, access control systems, and smart home technology. This can serve as a prototype for commercial smart locks, secure storage, and IoT-based access management systems.
- **IoT and Real-Time Embedded Systems:** The project demonstrates practical skills in UART communication between microcontrollers and WebSocket-based real-time multiplayer interaction, technologies widely used in industrial automation, robotics, and connected devices.
- **Entertainment and Edutainment Markets:** The escape box can be adapted into commercial escape-room kits or interactive educational toys, targeting hobbyists, schools, museums, and gaming centers. Its modular and scalable design allows for customization with multiple riddles or puzzles, creating engaging experiences for users.
- **Prototype for Interactive Products:** The modular architecture—separating game logic (ESP32), actuator control (Arduino), and user interface (web browser, keypad, LCD)—demonstrates best practices in product prototyping, providing a foundation for industries developing hybrid physical-digital products.

3. Social Engagement and Innovation

- **Bridging Digital and Physical Experiences:** The combination of real-time web-based interaction and physical actuation (solenoid and servo) creates a tangible interactive experience, making technology more relatable and exciting.
- **Fostering Creativity and Innovation:** By allowing users to design new puzzles, riddles, or game logic, the project encourages creative thinking and innovation, which are critical in both educational and industrial contexts.

- **Workshops and Demonstrations:** The project is ideal for educational exhibitions, hackathons, and innovation showcases, helping students, researchers, and hobbyists understand embedded systems and IoT integration.
- **Promoting Awareness of Modern IoT Applications:** Participants gain exposure to practical applications of IoT, real-time communication, and embedded control, which are key skills in today's tech-driven industries.

4. Broader Implications

- **Educational Equity:** The project can be implemented in schools or community workshops to make advanced technology concepts accessible to students from diverse backgrounds.
- **Commercial Potential:** Beyond education, the project can be scaled for interactive escape rooms, gamified learning tools, and secure product prototypes, demonstrating both social and economic relevance.
- **Future Innovation Pathways:** The modular system can evolve into AI-powered riddles, mobile app integration, multi-room escape challenges, or more complex IoT-based gaming solutions, bridging the gap between academic projects and real-world industry applications.

In summary, the Escape Puzzle Box is more than a simple game; it is a convergence of education, entertainment, and industry-relevant IoT technology. It teaches practical skills, fosters teamwork, inspires creativity, and demonstrates concepts directly applicable in the fields of embedded systems, smart automation, and interactive product design.

Learning and Reflection

The Escape Puzzle Box project provided valuable learning experiences in embedded systems, IoT integration, and system design. The reflection can be organized into technical, personal, and professional insights:

1. Technical Learning

- **Embedded System Design:** Gained hands-on experience programming both ESP32 and Arduino, understanding their roles in separating game logic and hardware actuation.
- **UART and WebSocket Communication:** Learned how to implement reliable serial communication between microcontrollers and real-time multiplayer communication over Wi-Fi using WebSockets.
- **Integration of Hardware Components:** Developed skills in connecting keypad, LCD, LEDs, solenoid lock, and servo motors, ensuring synchronized operation with software logic.
- **Software Modularity:** Learned the importance of separating game logic, user interface, and actuator control for easier debugging, scalability, and maintenance.

2. Personal Learning

- **Problem-Solving and Debugging:** Encountered and resolved issues related to timing, communication delays, and hardware-software coordination.
- **Time Management:** Learned to plan tasks systematically, prioritizing component testing, integration, and troubleshooting.
- **Adaptability:** Improved ability to adapt solutions when hardware limitations or unexpected behavior occurred during testing.

3. Professional Skills and Reflection

- **Project Documentation:** Gained experience in documenting technical processes, system architecture, and design decisions clearly for academic or professional audiences.
- **Industry-Relevant Skills:** Developed understanding of IoT protocols, embedded systems communication, and real-time user interface design, which are valuable for careers in IoT, robotics, and automation.

4. Reflection on Project Impact

- The project reinforced the importance of combining hardware and software for real-world solutions.
- Learned how interactive, user-driven systems can be designed to be intuitive, reliable, and engaging.
- Recognized areas for future improvement, such as adding multiple puzzles, AI-based riddles, or mobile app integration to enhance user experience and complexity.

The Escape Puzzle Box project was a comprehensive learning experience that enhanced both technical and professional skills while providing insights into practical IoT-based system design.

Main Code and File Structure

Libraries Used

- **Servo.h**
  - Control servo motor to open the box lid
- **WiFi.h**
  - ESP32 Wi-Fi connectivity
- **ESPAsyncWebServer.h**
  - Asynchronous web server for ESP32
- **AsyncTCP.h**
  - Supports asynchronous TCP connections
  - Internal for WebSocket communication
- **LiquidCrystal_I2C.h**
  - Interface with I2C LCD (16x2)
- **Keypad.h**
  - Interface with 4x4 matrix keypad

Functions from Your Code

- **setup()**
  - Initializes Wi-Fi, WebSocket, LCD, keypad, and hardware pins
- **loop()**
  - Main loop to check keypad inputs and manage game logic
- **onWsEvent()**
  - Handles WebSocket events such as player connection, code verification, and answer submission
- **getPlayerIndex(AsyncWebSocketClient* client)**
  - Returns the index of a connected player in the array

Functions from the Libraries

- **Servo.h**
  - attach(pin): Attaches servo motor to a specified pin
  - write(angle): Rotates servo to the given angle (0–180°)
- **WiFi.h**
  - WiFi.softAP(ssid, password): Sets ESP32 as access point
  - WiFi.softAPIP(): Returns IP address of ESP32 AP
- **ESPAsyncWebServer.h / AsyncTCP.h**
  - server.addHandler(&ws): Registers WebSocket handler
  - server.begin(): Starts the asynchronous server
  - ws.onEvent(onWsEvent): Attaches WebSocket event callback
- **LiquidCrystal_I2C.h**
  - init(): Initializes the LCD
  - backlight(): Turns on LCD backlight
  - setCursor(col,row): Sets cursor position
  - print(text): Prints text on the LCD
  - clear(): Clears the LCD display
- **Keypad.h**
  - getKey(): Reads the pressed key from keypad matrix
  - makeKeymap(keys): Creates key mapping for keypad rows and columns

Main Code

ESP32 Code

```
#include <WiFi.h>
#include <ESPAsyncWebServer.h>
#include <AsyncTCP.h>
#include <LiquidCrystal_I2C.h>
#include <Keypad.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

const byte ROWS = 4, COLS = 4;
char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowpins[ROWS] = {13, 12, 14, 27};
byte colpins[COLS] = {26, 25, 33, 32};
Keypad keypad = Keypad(makeKeymap(keys), rowpins, colpins, ROWS, COLS);

const char* ssid = "EscapeBox";
const char* password = "Password";

\AsyncWebServer server(80);
AsyncWebSocket ws("/ws");
```

```
String storedCode = "";
bool codeEntered = false;

String riddle = "I speak without a mouth and hear without ears. What am I?";
String answer = "echo";

AsyncWebSocketClient* players[4];
int playerCount = 0;
int currentPlayer = 0;
bool gameOver = false;

const int ledPin = 2;

int getPlayerIndex(AsyncWebSocketClient* client) {
  for (int i = 0; i < playerCount; i++) {
    if (players[i] == client) return i;
  }
  return -1;
}

void onWsEvent(AsyncWebSocket *server,
               AsyncWebSocketClient *client,
               AwsEventType type,
               void *arg,
               uint8_t *data,
               size_t len) {
```

```
if (type == WS_EVT_CONNECT) {
   client->text("CONNECTED");
  }

  if (type == WS_EVT_DISCONNECT) {
    int idx = getPlayerIndex(client);
    if (idx != -1) {
      for (int i = idx; i < playerCount - 1; i++)
        players[i] = players[i + 1];
      playerCount--;
      if (currentPlayer >= playerCount) currentPlayer = 0;
    }
  }

  if (type == WS_EVT_DATA) {
    String msg = "";
    for (int i = 0; i < len; i++) msg += (char)data[i];

    if (msg.startsWith("CODE:")) {
      String code = msg.substring(5);

      if (!codeEntered) {
        client->text("CODE_NOT_SET");
        return;
      }

      if (code == storedCode) {
        if (getPlayerIndex(client) == -1 && playerCount < 4) {
          players[playerCount++] = client;
        }
```

```
ifif (playerCount < 2) {
          client->text("NEED_MORE_PLAYERS");
          return;
        }

        int myIndex = getPlayerIndex(client);
        if (myIndex == currentPlayer) {
          client->text("RIDDLE:" + riddle);
        } else {
          client->text("WAIT");
        }
      } else {
        client->text("WRONG_CODE");
      }
    }

    if (msg.startsWith("ANSWER:") && !gameOver) {
      int myIndex = getPlayerIndex(client);
      if (myIndex != currentPlayer) {
        client->text("NOT_YOUR_TURN");
        return;
      }

      String ans = msg.substring(7);
      if (ans.equalsIgnoreCase(answer)) {
        gameOver = true;

        digitalWrite(ledPin, HIGH);
        Serial2.println("OPEN");
```

```
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Player ");
        lcd.print(currentPlayer + 1);
        lcd.setCursor(0, 1);
        lcd.print("is Winner!");

        ws.textAll("WINNER");
      } else {
        currentPlayer = (currentPlayer + 1) % playerCount;
        ws.textAll("NEXT_PLAYER:" + String(currentPlayer + 1));
      }
    }
  }
}


void setup() {
  Serial.begin(115200);
  Serial2.begin(9600);

  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);

  lcd.init();
  lcd.backlight();
  lcd.print("Enter Code:");
```

```
  WiFi.softAP(ssid, password);
  Serial.println(WiFi.softAPIP());

  ws.onEvent(onWsEvent);
  server.addHandler(&ws);
  server.begin();
}


void loop() {
  if (!codeEntered) {
    static String input = "";
    char key = keypad.getKey();

    if (key) {
      if (key == '#') {
        storedCode = input;
        input = "";
        codeEntered = true;

        lcd.clear();
        lcd.print("Code Set");
        delay(1000);
        lcd.clear();
        lcd.print("Players Join");
      }
```

```
    else if (key == '*') {
      input = "";
      lcd.clear();
      lcd.print("Enter Code:");
    }
    else if (input.length() < 6) {
      input += key;
      lcd.setCursor(input.length() - 1, 1);
      lcd.print("*");
    }
   }
  }
}
```

Arduino Code:

```
#include <Servo.h>
#define RELAY_PIN 7
#define SERVO_PIN 9
Servo boxlid;
void setup() {
 Serial.begin(9600);
 pinMode(RELAY_PIN, OUTPUT);
 digitalWrite(RELAY_PIN, LOW);
 boxlid.attach(SERVO_PIN);
 boxlid.write(0);
}
```

```
void loop() {
 if (Serial.available()) {
 String cmd = Serial.readStringUntil('\n');
 cmd.trim();
 if (cmd == "OPEN") {
 digitalWrite(RELAY_PIN, HIGH);
 delay(3000);
 boxlid.write(90);
 delay(5000);
 }
 }
}
```

Conclusion and Future Scope

The Escape Puzzle Box project successfully demonstrates a complete and well-integrated embedded and IoT-based system that combines software intelligence with physical actuation to create an engaging, interactive experience. The project effectively bridges the gap between digital gameplay and real-world hardware by integrating an ESP32 for game logic, networking, and multiplayer coordination with an Arduino responsible for controlling electromechanical components such as the solenoid lock and servo motor through UART communication.

Throughout the development of this project, key embedded system concepts such as microcontroller interfacing, serial communication, real-time event handling, and modular system design were implemented and validated. The use of a keypad and LCD for local interaction ensures secure code entry and clear system feedback, while the WebSocket-based Wi-Fi interface enables real-time multiplayer participation through standard web browsers. The turn-based riddle mechanism enforces fairness and structured gameplay, enhancing both usability and reliability.

From a technical perspective, the separation of responsibilities between the ESP32 and Arduino improves system stability and scalability. The ESP32 efficiently manages networking, player authentication, and game state, while the Arduino ensures precise and reliable control of hardware actuators. This modular architecture reflects industry-relevant design practices and allows future upgrades without major changes to the existing system.

In conclusion, the Escape Puzzle Box stands as a successful implementation of a real-time, interactive embedded system, offering valuable technical insights, hands-on experience, and strong potential for future enhancements. The project not only meets its defined objectives but also lays a solid foundation for further innovation in IoT-based interactive and security-focused systems.

Future Scope

The Escape Puzzle Box project has several opportunities for enhancement and expansion:

1. **Multiple Puzzles and Levels**
   - Introduce a series of riddles or puzzles with increasing difficulty.
   - Enable sequential unlocking to create more engaging gameplay.
2. **Mobile App Integration**
   - Develop a mobile application for players to join and interact with the game.
   - Enable notifications, hints, or score tracking via the app.
3. **Advanced Security Features**
   - Implement encryption for Wi-Fi and UART communication to enhance security.
   - Use biometric authentication (fingerprint or facial recognition) for code entry.
4. **AI-Powered Riddles and Dynamic Challenges**
   - Integrate AI to generate dynamic riddles or adaptive challenges.
   - Provide personalized gameplay based on player performance.
5. **Scalability for Larger Multiplayer Games**
   - Support more than four players with networked ESP32 boards.
   - Enable multi-room escape puzzles with interconnected boxes.
6. **Integration with IoT Platforms**
   - Connect to cloud services for data logging, analytics, and remote monitoring.
   - Allow remote access or control for demonstration or educational purposes.
7. **Enhanced Feedback and Interaction**
   - Add additional actuators like motors, lights, or sound modules for immersive experience.
   - Use touchscreens or OLED displays for richer interaction.
8. **Commercial and Educational Applications**
   - Deploy as an interactive educational kit for STEM learning.
   - Adapt for commercial escape rooms or gamified learning platforms.