1.Project Title:

RSF Gaming Console.

2.Introduction:

The RSF Gaming Console is a microcontroller-based DIY gaming system designed to offer a simple yet engaging gaming experience using minimal electronic components. With the rise of retro-style gaming and educational electronics, this project serves both as an entertaining console and a hands-on learning tool for embedded systems. At its heart is the ATmega328P microcontroller, which handles the core game logic, display control, sound generation, and user input. The console features a SH1106 OLED screen for graphical output, push buttons for user interaction, LEDs for visual effects, and a passive piezo speaker for basic audio feedback. Power is supplied via a battery holder, and a switch allows for manual power control. To ensure accurate timing and stable operation, a 16MHz crystal oscillator and 22pF ceramic capacitors are used.

This project is built entirely with readily available components such as jumper wires, buttons**,** capacitors, and a breadboard, making it a flexible and low-cost prototype. It is programmed using the Arduino IDE, allowing easy development of simple games like Pong, Dino Runner, or Reaction Time challenges. The RSf Gaming Console not only demonstrates how microcontrollers can drive interactive applications but also provides valuable insights into hardware-software integration. The combination of portability, affordability, and functionality makes this an ideal project for students and hobbyists exploring the world of embedded systems and digital design.

3.Methodology:

3.1.Components Overview:

The RSf Gaming Console was constructed using basic yet essential hardware components:

- **Microcontroller**: ATmega328P, responsible for controlling all I/O operations, game logic, and interfacing with peripherals.
- **Display**: SH1106 OLED (128x64), connected via I2C (SCL, SDA) for graphical output.
- **User Input**: 6 push buttons for UP, DOWN, LEFT, RIGHT, START, BTN A, and BTN B.
- **Audio Output**: A passive piezo buzzer connected to a PWM pin for tone generation.
- **LED Indicator**: Red LED for status or power indication.
- **Timing Circuit**: 16 MHz crystal oscillator with two 22pF ceramic capacitors connected to the XTAL pins of the microcontroller.
- **Power Supply**: Battery holder connected via a switch (S1) for on/off control.
- **Connections**: All wiring is completed with male-to-male jumper wires on a breadboard.

- **Programming Header**: ICSP (In-Circuit Serial Programming) header for flashing firmware to ATmega328P.

3.2 Circuit Design:

Below is the full circuit diagram of the RSf Gaming Console:

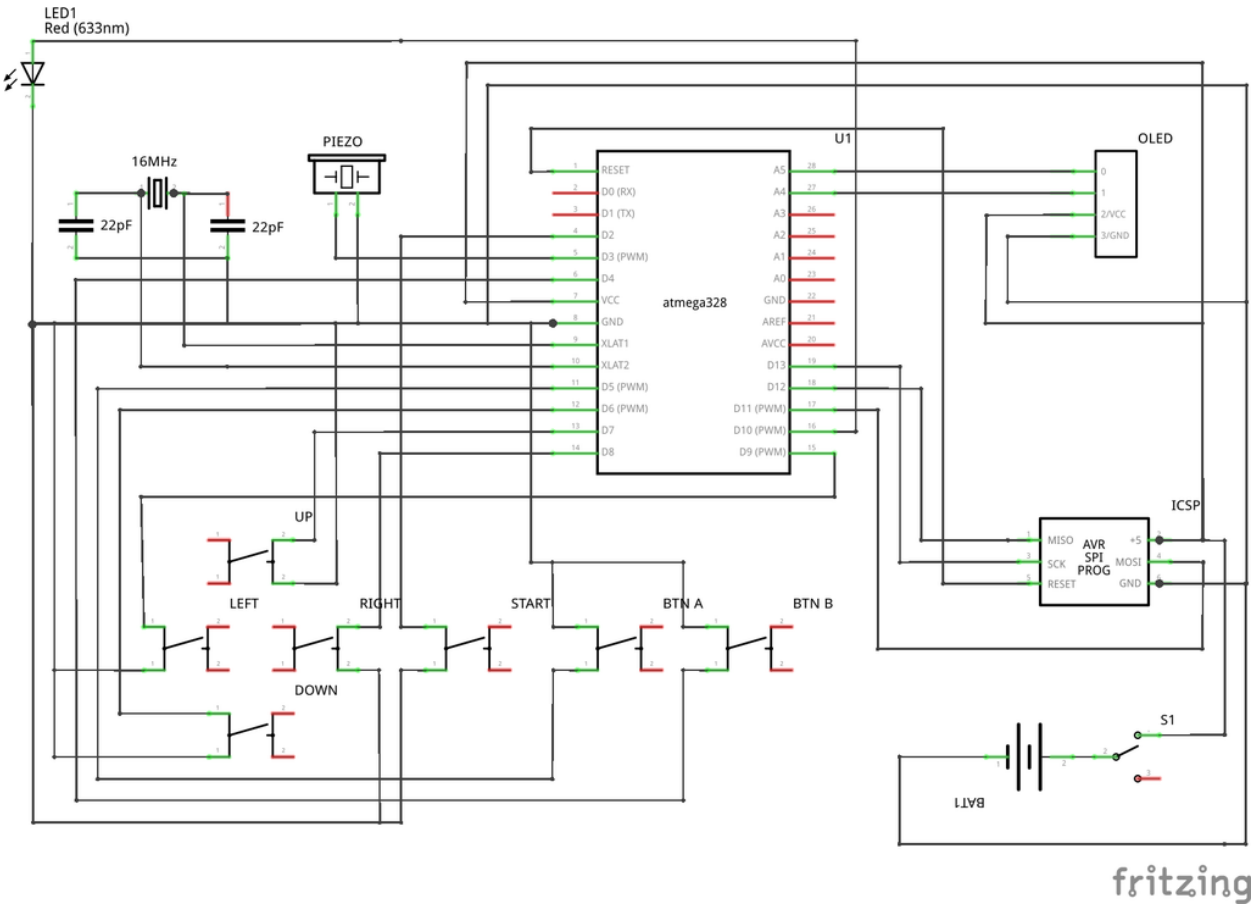


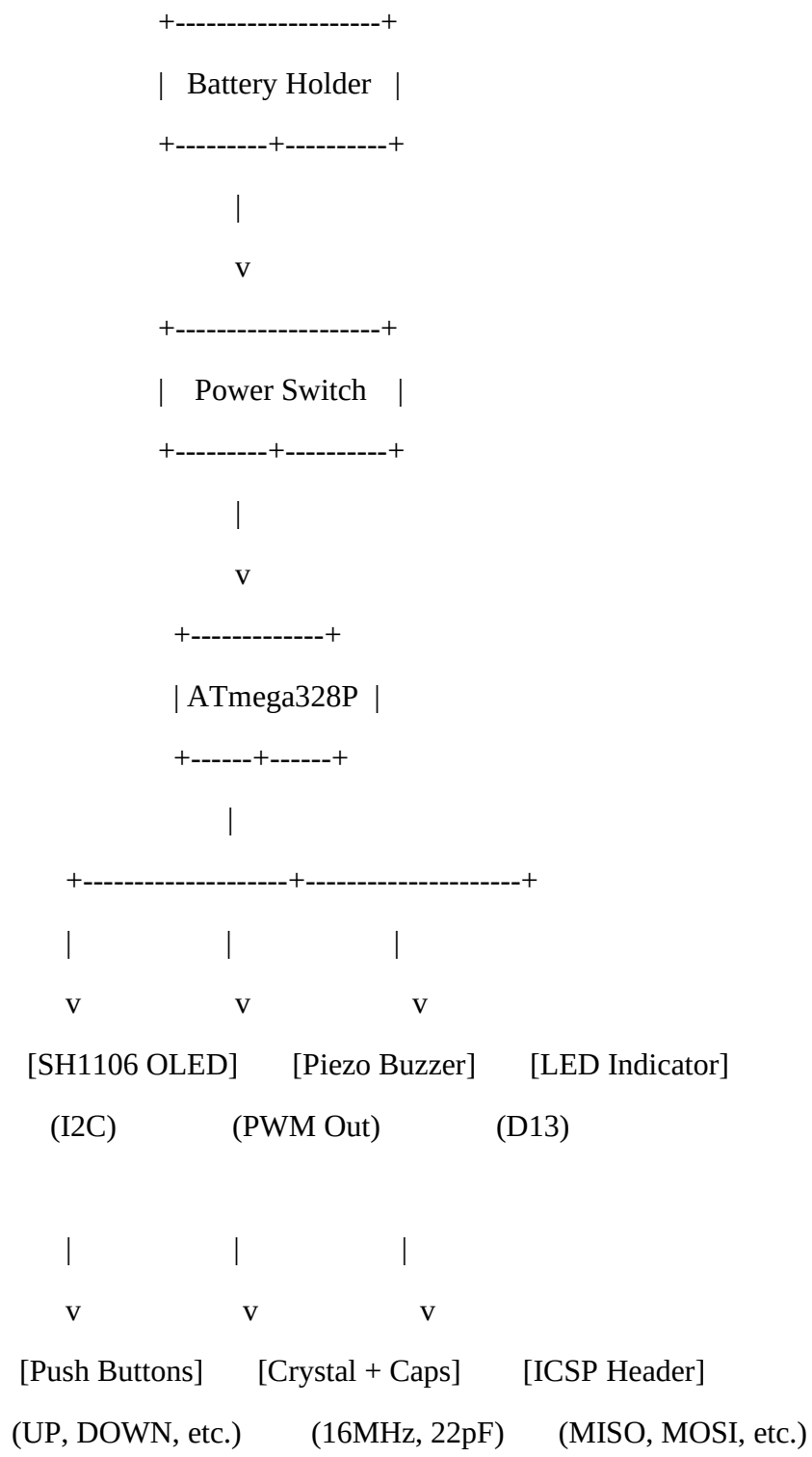
Fig.1: Circuit diagram of gaming console

Key Observations:

- **Microcontroller Setup**: The ATmega328P is configured with an external 16MHz crystal and two 22pF capacitors connected to the XTAL1 and XTAL2 pins for stable operation.
- **OLED Interface**: The SH1106 display is connected via I2C (SDA, SCL) to the ATmega328P.
- **Button Inputs**: Push buttons are connected to digital pins with internal pull-up resistors.
- **Sound Output**: The passive piezo buzzer is connected to a PWM-capable digital pin for sound generation.
- **LEDs**: Connected with current-limiting resistors to GPIO pins for visual feedback.
- **Power Supply**: A battery holder provides 3V to 5V, with a slide switch used to turn the console on/off.

3.3 Architectural Diagram (Block View):

Here's a simplified Architectural Diagram (block representation):



3.4. Software Development:

- Developed using Arduino IDE with ATmega328P bootloader.
- Libraries:
  - **U8g2 Library**: For interfacing with the SH1106 OLED.
  - **Tone Library**: To generate sound on the passive buzzer.
- Games Developed:

Simple games such as DINO**,** Maze Runner were implemented to test graphics, buttons, and sound.

4. Output:

The RSf Gaming Console successfully boots up upon switching on the battery-powered system, displaying a welcome or menu screen on the SH1106 OLED display. From there, the user can start either the Dino or Maze game using the push buttons. In the Dino Runner game, the screen shows a character on the left that must jump over approaching obstacles. The gameplay runs smoothly with responsive controls, and the jumping action is rendered clearly despite the limited resolution. If the player hits an obstacle, the game displays a game-over screen, and the piezo buzzer emits a distinct tone to indicate failure. The LED also lights up to show power or game state.

In the Maze game, the OLED renders a maze structure, and the player navigates a pixel-based character through the paths using directional buttons. Collision logic ensures that the player can't pass through walls, and reaching the end of the maze triggers a success message along with a simple tone from the buzzer. Both games demonstrate accurate button input, consistent display performance, and reliable game logic processing. The buzzer provides real-time audio feedback, while the LED gives visual cues, enhancing the interactive experience. Despite the system being built on a breadboard with jumper wires, the console operates reliably, showcasing the ATmega328P's capability in delivering responsive and engaging game output using minimal resourc

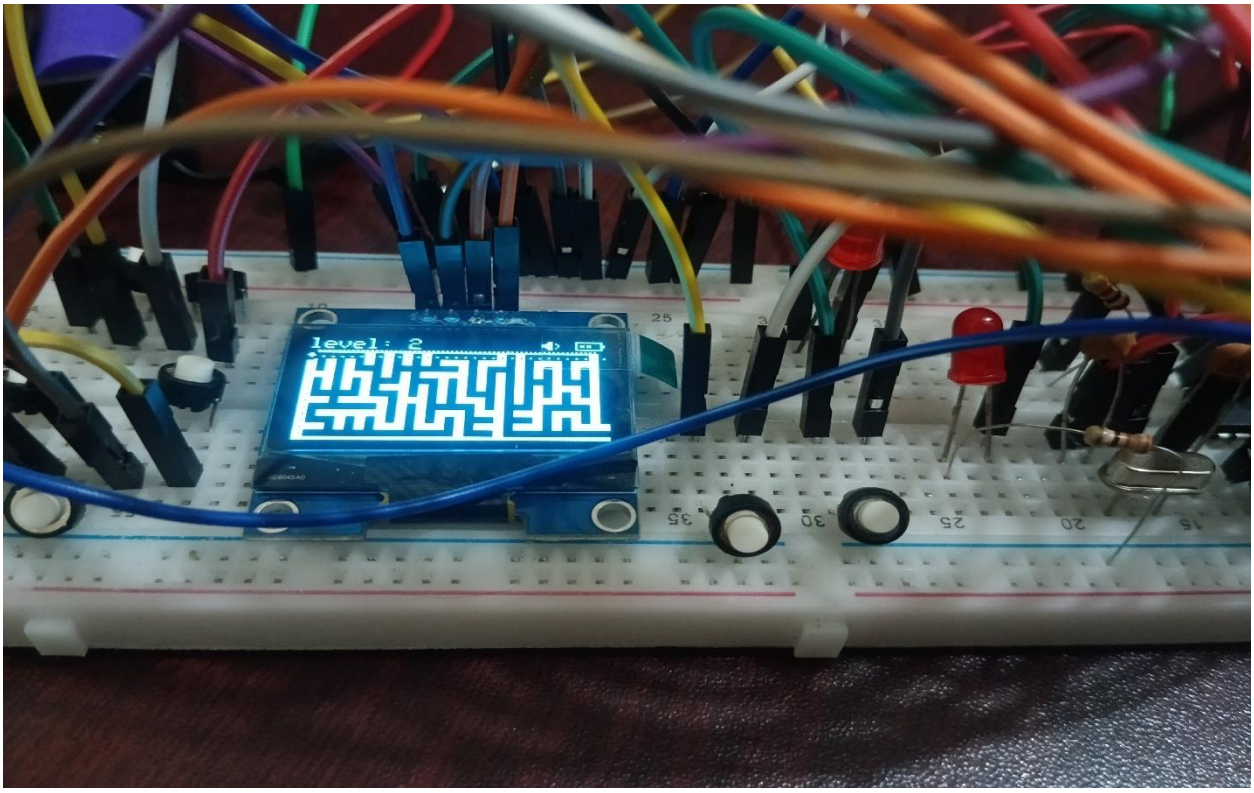


Fig.2:Maze runner as output

5. Discussion:

Integrating two distinct games into a limited-resource microcontroller posed some challenges, particularly in memory optimization and screen rendering speed. The Dino game required efficient redrawing of moving frames to simulate animation, while the Maze game demanded precise input handling and collision detection.[1], [2], [3] The SH1106 OLED provided a suitable resolution for these games, although its refresh rate required careful frame management to avoid flicker. Audio feedback using the passive piezo was successfully implemented, albeit in a minimalistic form. Despite being constructed entirely on a breadboard using jumper wires, the console remained stable during gameplay. This proves that even with basic components, interactive systems like gaming consoles are achievable and educational.

6. Conclusion:

The RSf Gaming Console project demonstrates how simple electronic components and a microcontroller like the ATmega328P can be combined to create a fully functional, interactive gaming device. Through the integration of input controls, an OLED display, a passive buzzer, and LEDs, this console offers a complete user experience with two playable games—Dino Runner and Maze. These games not only showcase fundamental game mechanics like real-time rendering,

collision detection, and input handling but also prove the versatility of embedded systems in entertainment applications. The successful implementation of the project highlights how even limited hardware resources can deliver meaningful and enjoyable outcomes.

Beyond just gameplay, this project serves as a strong foundation for understanding embedded system design, microcontroller programming, and hardware-software integration. It reinforces essential concepts such as timing control using crystal oscillators, interfacing over I2C, managing digital inputs and outputs, and producing sound through PWM. The console's portable and low-cost nature makes it ideal for learning and experimentation. With future upgrades, such as EEPROM-based score saving, menu navigation, or multiplayer support, the RSf Gaming Console could evolve into a more sophisticated platform. Overall, this project successfully meets its goals of creativity, education, and functional design.

7.References:

[1] "A-Maze/A-Maze.ino at master · alojzjakob/A-Maze · GitHub." Accessed: Jul. 14, 2025. [Online]. Available: https://github.com/alojzjakob/A-Maze/blob/master/A-Maze.ino

[2] "Build Your Own Video Game Console — Kids & Tech | by Ryan Boder | Technology Hits | Medium." Accessed: Jul. 14, 2025. [Online]. Available: https://medium.com/technology-hits/build-your-own-video-game-console-kids-tech-68296ab8ad48

[3] "Architecture of Consoles | A Practical Analysis." Accessed: Jul. 14, 2025. [Online]. Available: https://www.copetti.org/writings/consoles/