PROJECT REPORT

Team members:

Nikhil Venkat Atkuru, Sai Abhinav Nagamalla

Team ID: 4

November 30th, 2024

Contents

1)Problem Statement

- Make a classic arcade game “Snake Game” using Arduino and a display module.
- The traditional Snake Game is limited to graphical interfaces and cannot be played on hardware-based setups like LCDs, which makes it less accessible for interactive learning and embedded systems enthusiasts.
- Currently, there is no simple way to play a Snake Game on an LCD controlled by an Arduino. The challenge lies in translating graphical gameplay into hardware interactions while ensuring simplicity, efficiency, and user engagement

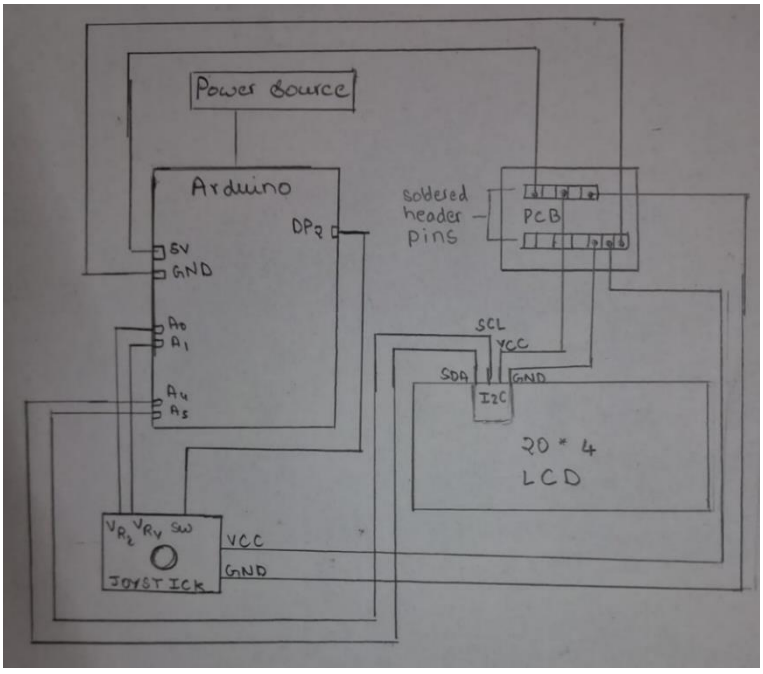
2)Aim of the project

- To design and implement a hardware-based Snake Game using an Arduino, joystick, and an LCD, providing an engaging gaming experience with a simple embedded system.
- This project aims to create a functional and engaging Snake Game, where users control a snake to collect food and avoid bombs using a joystick, with all actions visually represented on an LCD.

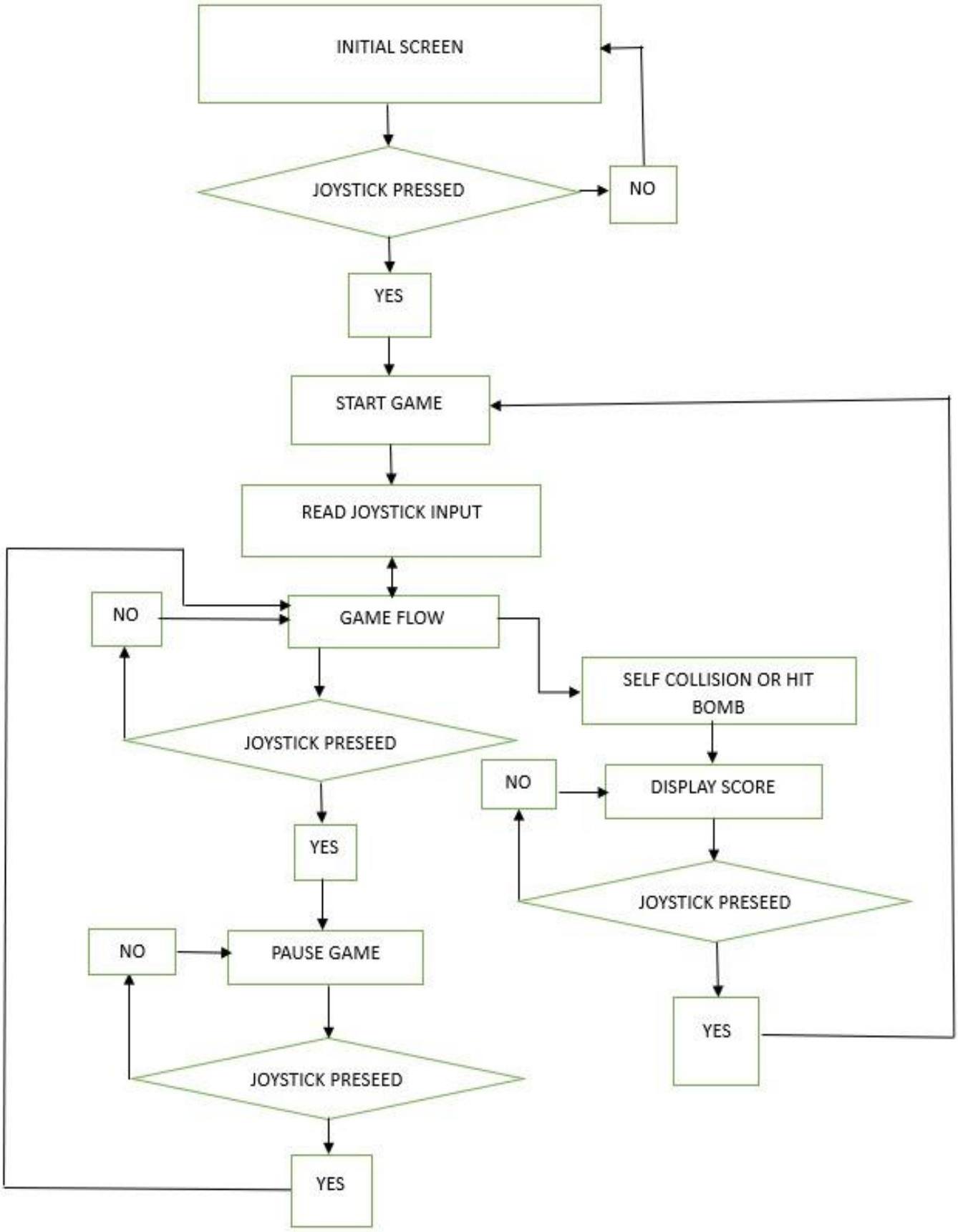
3)Components Used

- Arduino Uno
- PCB-0
- USB-CABLE
- 20X4 LCD screen
- Joystick module
- I2C module
- Jumper wires
- Arduino IDE on Laptop
- Power Source
- Carboard Box (as a game box)

4)Circuit Schematics and Process Diagrams

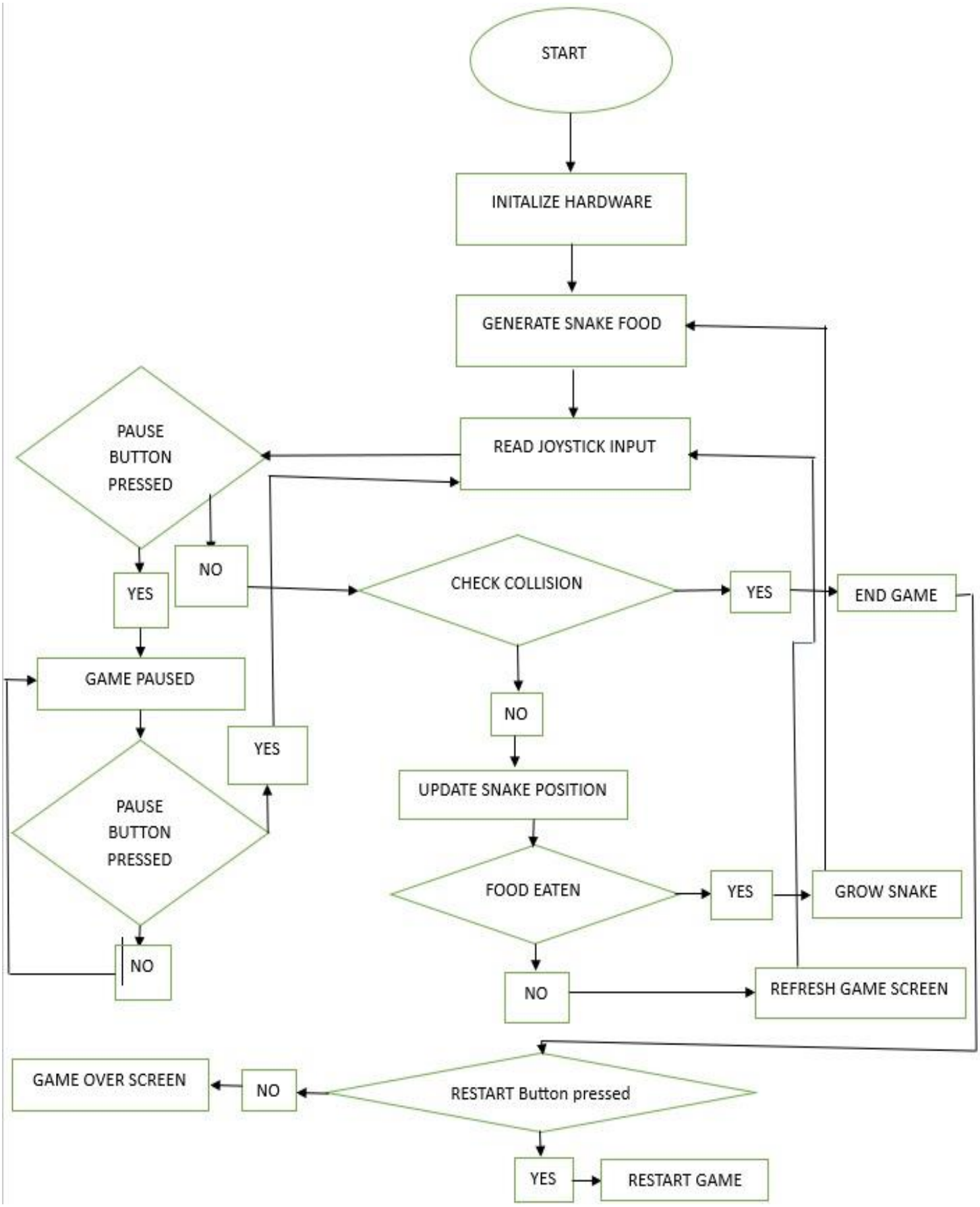


Circuit Diagram



Process Diagram

5)Flowchart of the code



6)Description of the project

The Snake Game Project integrates an interactive gaming experience using an Arduino UNO, joystick, and a 20x4 LCD display. This project combines hardware and software components to simulate the classic snake game while utilizing real-time user inputs. Below is a detailed explanation of the approach, methodology, and steps involved:

Approach

The project is designed to mimic the classic snake game, where players control a snake that moves around the screen to collect food while avoiding collisions with itself and bombs. The game progresses with increasing difficulty as the snake grows in length.

- **User Interaction**: The joystick acts as the primary control for the snake's movement (up, down, left, right).
- **Display**: The game is visually rendered on a 20x4 LCD using the LiquidCrystal_I2C library for efficient communication.
- **Logic**: The Arduino handles game mechanics, including movement, collision detection, and score tracking.

Methodology

1. **Hardware Setup**:
    - The joystick module is connected to the Arduino for reading directional input.
    - The 20x4 LCD, along with an I2C module, simplifies connections and enables clear graphical representation of the snake, food, and bombs.
2. **Game Logic**:
    - **Snake Movement**: The snake moves in a direction based on joystick input, and its position updates on the LCD.
    - **Food Generation**: Randomly generated food appears on the screen, avoiding locations near the snake's path to allow reaction time.
    - **Bombs**: Bombs occasionally appear, adding a layer of challenge, and are randomly positioned away from the snake.
    - **Collision Detection**: The game checks for collisions with the snake's body or a bomb, triggering a "Game Over" state if detected.
    - **Score Tracking**: Each piece of food eaten increases the score and, at intervals, accelerates the snake's movement to increase difficulty.

3. **Display Management**:
    - The LCD renders the snake, food, and bomb using specific characters, updated every game loop.
4. **Game States**:
    - **Start Screen**: Displays instructions (press to play) and waits for a button press to begin.
    - **Countdown**: A three-second countdown precedes gameplay.
    - **Gameplay**: The snake moves continuously; user input alters direction.
    - **Pause**: A button press pauses/resumes the game.
    - **Game Over**: Displays the player's score and provides the option to restart.

Steps Involved

1. **Hardware Assembly**:
    - Solder connections between components and mount them securely within a game box.
    - Connect the joystick module to the Arduino (analog pins for X/Y, digital pin for button).
    - Attach the I2C module and LCD to the Arduino.

2. Code Development:

- Write functions to handle joystick input, game logic, and display updates.
- Implement algorithms for random food/bomb generation and collision detection.
- Optimize the refresh rate of the LCD for smooth gameplay.

3. Testing and Refinement:

- Test each module independently (joystick, LCD, and collision detection).
- Debug issues such as display glitches or incorrect input handling.
- Refine the logic for smoother gameplay and a responsive user experience.

4. Final Assembly:

- Organize all components within a game box, ensuring clean wiring and accessibility.
- Label controls and display areas for user-friendliness.

7)Results

- The snake game successfully runs on the 20x4 LCD display connected using an I2C module.
- The joystick is used for the movement of the snake throughout the game.
- The game mechanics include food generation, snake movement, and collision detection.
- The movement of the snake is smooth and moves according to the input of the joystick.
- Less to no delay is observed in the movement of the snake.
- The movement of the snake across the edges is wrapped around smoothly.
- The generation of the food is random, making the game unpredictable.
- The score keeps on updating whenever the snake eats the food.
- The game ends when the snake collides with itself.
- The game can be paused whenever liked by pressing the joystick button.
- Once the game ends the game can be restarted by pressing the joystick button.
- Once the snake reaches the size of 10, the speed of the snake keeps on increasing and the size remains constant.
- The bomb appearing on the screen automatically disappears after 5 seconds.

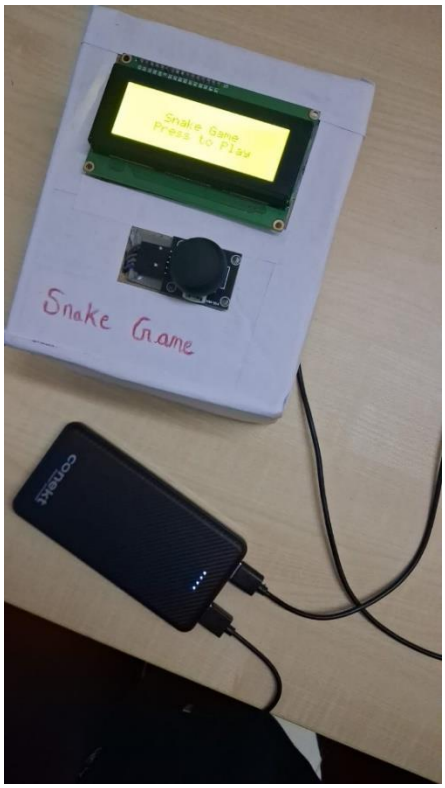
- The food disappears if not eaten in 7 seconds.
- The snake head is generated at random position at the start of the game.
- High score is displayed at the end i.e. in the game over screen.
- When the game is resumed or restarted, a 3 2 1 countdown is initialized, giving heads up to the player.
- The food and the bomb are not being generated on the snake or on the 3 squares in front of the snake.

8)Simulations and video demonstrations

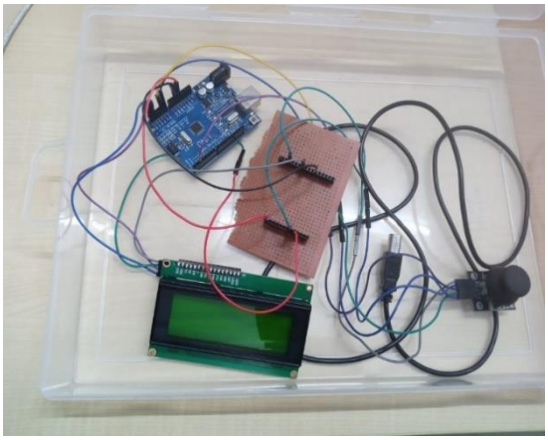
Demo link

Our tinkercad simulation does not work because tinkercad does not support the required header files for the code and it does not have joystick

Pictures of the final Game box



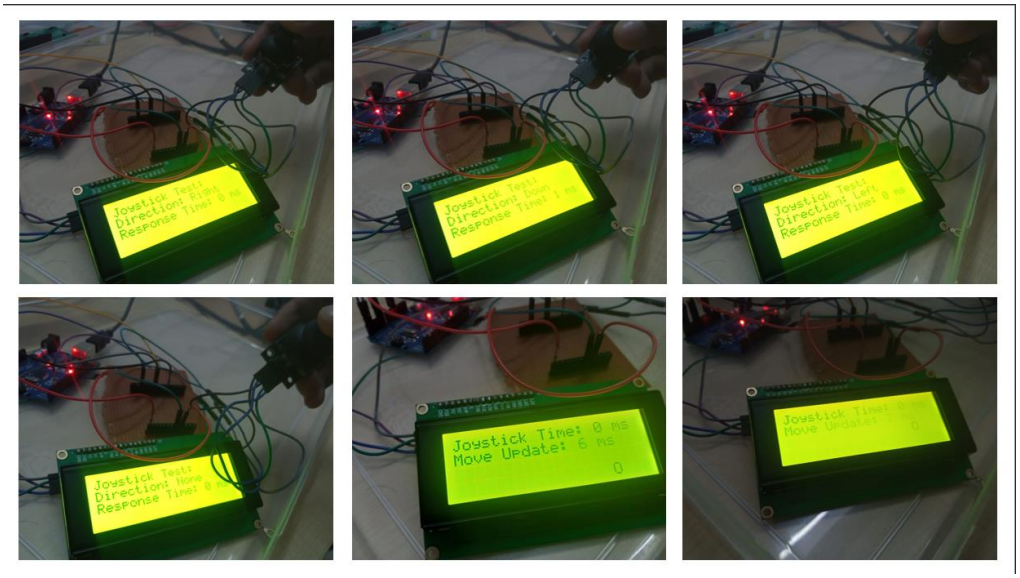
9)Photos



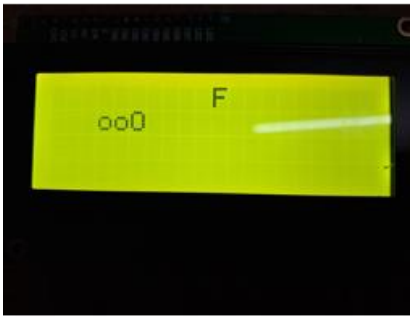
Our Circuit

Current Drawn by circuit

Results of our Key Performance Indicators tests



Few photos of our gameplay



10)Bibliography

- ChatGPT help to rectify our sequences of calling functions and rectify our game logic.
- To check the functioning of the lcd
- YouTube tutorial on how to use I2C Module
- A tutorial on our project done on a different display and different logic.
- Another approach to our project (different algorithm from what we used)

----------THE END----------