Snake Game

Introduction

My project is a copy of the popular game “Snake” displayed on an 8×8 Module LED Matrix using an Arduino UNO R3 board.

My inspiration came from my childhood when I was an avid fan of this and often played it trying to get a higher and higher score.

My aim is to try and recreate the game so that anyone can share in the fun of this old-school classic.

General Description

My project uses an Arduino UNO R3 board as its base, as well as an 8×8 Module LED Matrix to display the game itself as well as the score, a Module Joystick to control the Snake's movements during the game and a buzzer that makes noises whenever you pick up a fruit or lose.

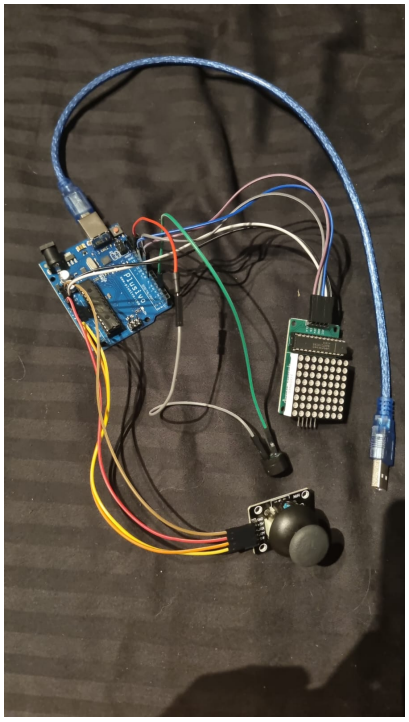
Hardware Design

List of Parts:

- Arduino Uno R3
- 8×8 Module LED Matrix
- Passive Buzzer
- Module Joystick

- Wires

Final Design:



Software Design

1. The necessary libraries are included, specifically the “LedControl” library for controlling the LED matrix.

2. Pin definitions are specified using a struct. Pins for the joystick, LED matrix, and buzzer are defined.

3. Constants for LED matrix brightness, message scrolling speed, and initial snake length are set.

4. The `setup()` function is called when the Arduino board starts. It initializes the serial communication, initializes pins and LED matrix, calibrates the joystick, and shows the “snake” message scrolling on the LED matrix.

5. The `loop()` function is continuously executed after the `setup()` function. It consists of several steps:

```
 a. `generateFood()` checks if there is food on the matrix and generates it if necessary.
 b. `scanJoystick()` watches the joystick movements and blinks the food on the matrix.
 c. `calculateSnake()` calculates the snake movement and updates the LED matrix accordingly.
 d. `handleGameStates()` checks for game over conditions, shows the score and game over messages, and resets the game if needed.
```

6. Supporting variables and structures are declared, including variables for snake and food positions, joystick calibration values, snake parameters, direction constants, joystick threshold, and gameboard storage.

7. The code includes various utility functions, such as `playDingSound()` and `playGameOverSong()`, which generate sounds using the buzzer, and `fixEdge()`, which handles snake movement at the edge of the LED matrix.

8. The code also includes functions for calibrating the joystick, initializing the board and LED matrix, and dumping the gameboard for debugging purposes.

9. The `snakeMessage` array defines the pattern for the “snake” message displayed on the LED matrix.

Overall, this code implements a basic snake game that runs on an Arduino UNO R3 board and displays the game on a LED matrix. The joystick is used to control the snake's movement, and the game ends when the snake collides with its own body.

Results

The result is a pretty vivid recreation of the original Snake Game. I would say it is quite good given the limited technology I used in comparison. Only major flaw I see is that if the snake is exactly the length of the LED matrix there is no way to tell where the head is, the snake just forming a line. It doesn't impede your experience in any way but I find it quite annoying.

Conclusions

In summary, my Arduino Uno Snake game project is a fun and educational demonstration of how programming and hardware can come together to create an engaging gaming experience. It has been an exciting journey that has taught me valuable skills and sparked my curiosity about microcontrollers.

Download

Code for the game : snake_game.txt

Bibliografy/Resources

Youtube Channel GDTitans : https://www.youtube.com/channel/UCUJGE6eXB1OXPXbx4CXzTPA

Snake game sounds mp3

Arduino_datasheet:
https://docs.arduino.cc/static/32b4941b81a2c6a5308e0e9bd348d0e6/A000066-datasheet.pdf

Export to PDF