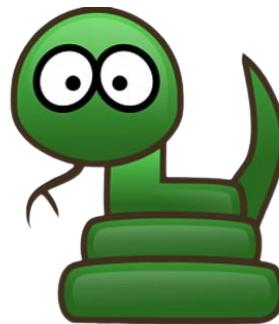


# Embedded Snake Game



By  
Ben Theurich  
Carter Rath  
Cody McKinney

CS435 - Real-Time Embedded Systems  
Professor Najjar  
13 May 2023

<https://github.com/BenTheurich/EmbeddedSnakeGame>

# Table of Contents

<u>1. Abstract</u>	<u>3</u>
<u>2. Introduction</u>	<u>4</u>
<u>2.1 Aim, Scope, and Relevance</u>	<u>4</u>
<u>2.2 Similar Work and Reference Material</u>	<u>4</u>
<u>2.3 Conceptual Design of the System</u>	<u>5</u>
<u>2.3.1 Components Needed</u>	<u>5</u>
<u>2.3.2 Circuit Design</u>	<u>6</u>
<u>2.3.3 PDL</u>	<u>7</u>
<u>2.3.4 Flowchart of Main</u>	<u>8</u>
<u>3. Implementation</u>	<u>9</u>
<u>4. Results</u>	<u>10</u>
<u>5. Lessons Learned</u>	<u>10</u>
<u>5.1 Project Management, Task Allocation, and Team Communication</u>	<u>10</u>
<u>5.2 External learning resources</u>	<u>12</u>
<u>5.3 Challenges faced</u>	<u>12</u>
<u>6. Summary</u>	<u>12</u>
<u>7. Bibliography</u>	<u>13</u>
<u>8. Code Listings</u>	<u>14</u>
<u>9. Demonstration</u>	<u>15</u>

## 1. Abstract

Our project will focus on using the Nucleo microcontroller along with an 8x8 LED matrix[4] with an integrated Max7219 LED display driver[2] and an omnidirectional joystick[3]. We will make use of the Mbed API for the Max7219 driver[1] for controlling our display and receiving input from the Joystick[5]. Our project will be a SnakeGame implementation controlled by the joystick and displayed on the 8x8 LED matrix. The LED matrix starts by loading all of the lights of the matrix in a pattern until the user is ready to start the game. Once the user starts the game by pressing select on the joystick, the lights will fade out row by row and the Snake will start at the top left with a length of two, ready to be controlled by the user. The aim of the game is for the user to guide the Snake to the fruit, avoiding collisions with itself and the edge of the game board. As the Snake consumes fruit it will add an additional part to the body, increasing the length and raising the difficulty of avoiding a collision with itself. Each time the Snake consumes fruit, the score will increase as well. The score will be updated in the terminal which will be cleared each time to show the new score. Additionally, each time a fruit is consumed, the system will play a noise using the piezo speaker.

## 2. Introduction

### 2.1 Aim, Scope, and Relevance

Our aim is to use our knowledge of embedded systems, such as writing binary digits to a display, receiving analog input, handling interrupts, and using the Mbed API in order to replicate a game that can be played in real-time with user interaction. This project gives us the opportunity to combine several different concepts we have learned throughout the class into a functioning project. Our project provides users who are classic video game enthusiasts to build themselves their own custom SnakeGame system using five purchasable parts; a Nucleo-F401RE STM32, a breadboard, an analog joystick with dual axis (XY) controls, an 8x8 LED matrix with a driver attached for use with Serial Peripheral Interface (SPI), and a piezo speaker for game noises.

### 2.2 Similar Work and Reference Material

The inspiration for this project came from a video of a pong game[6] created using two potentiometers and an 8x8 matrix LED, found on an Instagram page. The idea made us consider what other games we could implement on a matrix display with a different input from the user.

After doing some research, we were able to identify various implementations of the SnakeGame using an 8x8 Matrix, including a YouTube video[8] controlling the game with buttons, and a German Github repository[7] that published a very similar approach using a joystick with Mbed OS on an STM-32 Nucleo board.

Our approach will be different from the video because instead of using buttons, we are using a joystick to control the Snake. Additionally, our system outputs the score of the SnakeGame in the Mbed Studio terminal. Our game will also differ in terms of the flow of gameplay, we will include a pause and resume feature using interrupts, as well as a launch animation, a losing screen animation, and a game start animation. Additionally, we have added a piezo speaker to play noises in the game when a fruit is consumed.

## **2.3 Conceptual Design of the System**

### **2.3.1 Components Needed**

Nucleo-F401RE Microcontroller

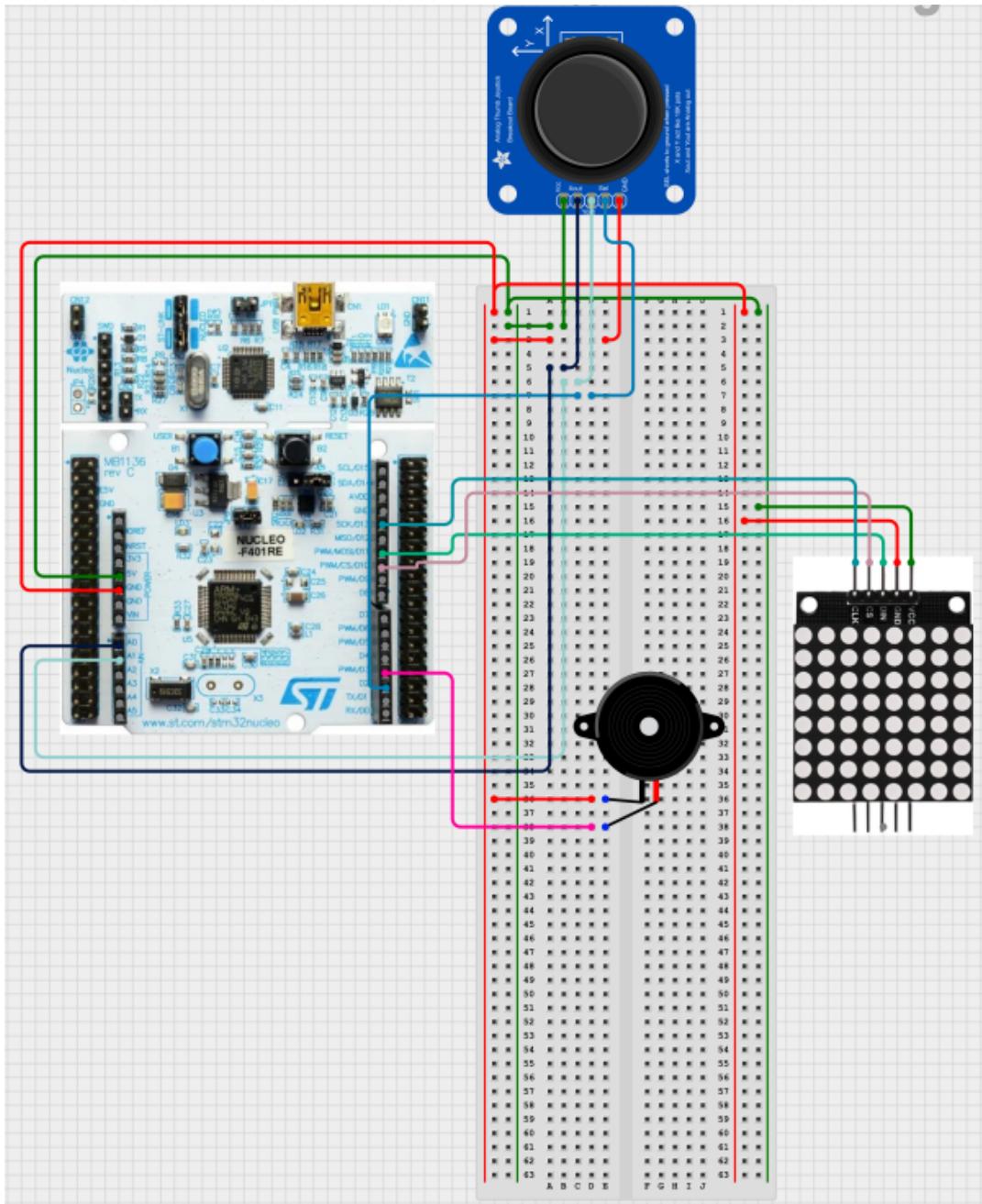
Breadboard

8x8 LED matrix for displaying the game

Analog joystick for user control

Piezo speaker for game sounds

### 2.3.2 Circuit Design



### Figure 2.3.2.1 Circuit Design

### **2.3.3 PDL**

#### **BEGIN**

Declare pins for the MAX7219 LED display driver

Declare pins for the Joystick

Declare the game board object

Configure the LED display

Enable the LED display

Start Launch Game animation

#### **DO FOREVER**

Click joystick to start game

Create new game board

Play Start Game animation

While game is not over and game is not paused

    Get direction of Snake from joystick

    Move Snake

    Check game status and end if game is lost

    Update game board

    If Snake consumes fruit

        Grow Snake

        Update score

    Loop through rows of game board

        Convert each row vector from binary to decimal

        Write 1s to the row of LED display according to decimal value

    Play end game animation when user loses

#### **ENDDO**

#### **END**

#### 2.3.4 Flowchart of Main

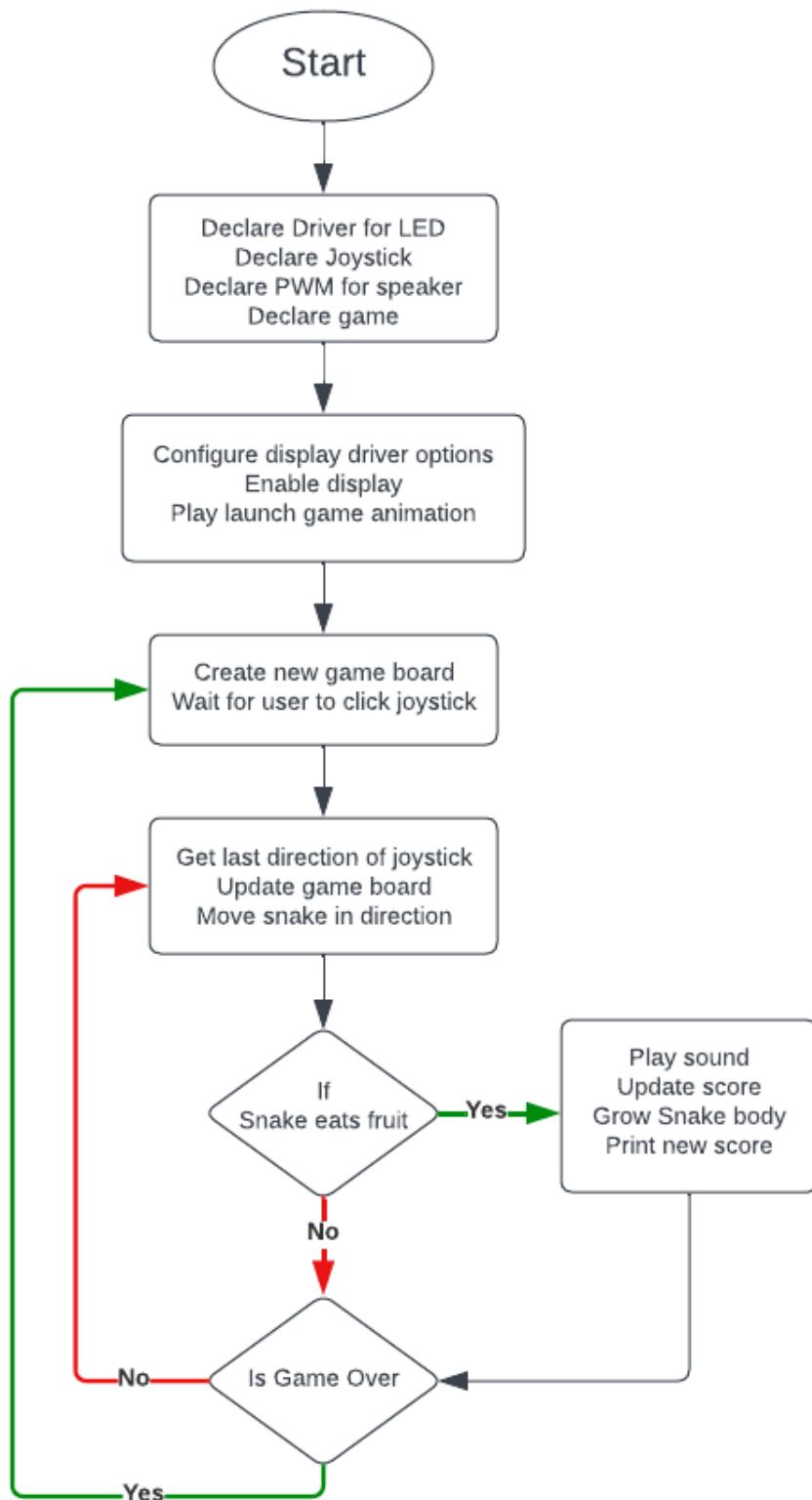


Figure 2.3.3.1 Flowchart of main.cpp

### 3. Implementation

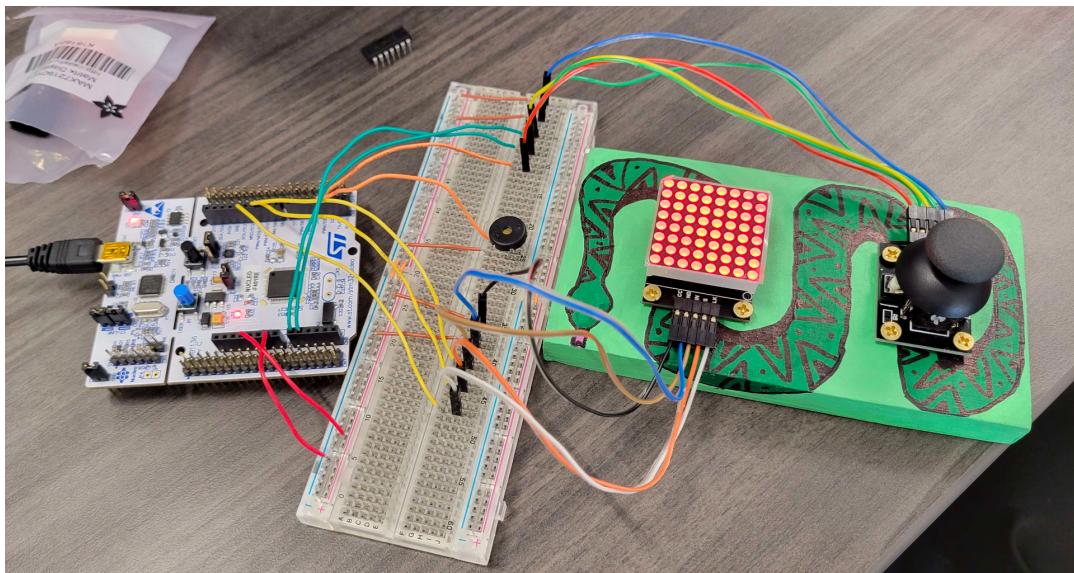
For the Snake object, we are using a linked list to track each body part. When the Snake moves, we add a node to the head and remove the tail. When the Snake eats a fruit, we move the snake forward by one without removing the tail. We found that this was the easiest and best way to keep track of where the snake is at all times.

For the Game object, we are using a 2D array to represent the board imposed on the 8x8 Matrix. For each row, we represent the LEDs that should be turned off as 0's and the LEDs that should be turned on as 1's. This makes it easy for us to create a binary number based on each row in the array and update the led matrix accordingly.

In terms of implementation of the components, we are using an 8x8 LED matrix that is working through Serial Peripheral Interface (SPI) with the Nucleo board. We are using an omnidirectional analog joystick that takes two analog inputs for X and Y directions, as well as a digital input for the select button. Finally, for the game sounds, we are using a piezo speaker that is using Pulse Width Modulation (PWM).

## 4. Results

This is the final result of our project. We managed to create what we intended to without facing too many challenges. Additionally, we were able to add nice-to-have features, such as animations, game sounds, and a decorative board for our components to attach to.



## 5. Lessons Learned

### 5.1 Project Management, Task Allocation, and Team Communication

For this project, we met several times on campus as a group and worked together on all portions of the project. The three of us have been working together in groups for several semesters and have gotten used to working with each other. For team communication, we regularly met in person to discuss the project, often spending around four hours in the Hunter Design Lab on campus.

In terms of project management, the code we wrote we uploaded to a GitHub repository that we could each pull from to make modifications. Each time we would meet, one of us would pull the repo to their machine, and we would all pair program the task we decided to address for the day.

For task allocation, because we were always in-person when working on the project together, each of us was intimately involved with all portions of the project. Cody invested time into the research of the system-to-be and the different types of hardware we could use to implement. To start the project, all three of us learned how to appropriately wire the hardware using SPI, and then learned how to write to the LED matrix. Once we understood the basics, Carter and Cody worked on implementing the Snake object using a linked list of nodes to represent the Snake's body. They then wrote the code for the game board as an eight-by-eight 2D array to handle each row and column, with the methods for the functionality of the game. Ben began working on the losing screen animation that displays a matrix digital rain on the LED to cover the screen. Carter and Cody then implemented the Joystick API and changed a portion of the default code to only handle four directions instead of eight. After this, they began working on reading the analog input of the joystick to determine the Snake's direction. Carter worked on the launch game animation and Ben worked on the handling of the game logic to check if the Snake runs into itself or the wall. He worked on the random spawning of the fruit, handling the consumption of the fruit as well as the score counter and printing to the console. Near the end, we added the piezo speaker, and Ben wrote the code to make a sound when a fruit is consumed as well as the sounds for the

loading and death screens. Lastly, we bought a piece of wood[11], Cody cut and sanded it, Carter painted it, and we attached our components to it.

## 5.2 External learning resources

We used the Mbed API[9] for a lot of the functionalities of our peripherals, including the Joystick API and MAX7219 driver API. Additionally, we used the datasheets for the joystick[3] and the LED matrix[10].

## 5.3 Challenges faced

We initially started off using an 8x8 LED matrix with the display driver as a separate component which had to be connected manually. Understanding how the MAX7219 driver interacted with the LED matrix was very time-consuming and took a lot of connections. There were over 24 wires that would have had to be connected each time we wanted to set up and work on the project. This would have made it very challenging to make progress on this project while still completing the class labs. Ultimately, we decided to go with a new matrix peripheral that had the driver already integrated. This made the setup much more simple as well as giving us a larger screen to work with.

## 6. Summary

Our project focused on creating a SnakeGame using a Nucleo microcontroller, an 8x8 LED matrix, and an omnidirectional joystick. By leveraging the Mbed API, we successfully implemented the LED display and joystick input which we used to control the Snake. The objective of the game is to guide the Snake around the board to

consume fruits while avoiding collisions with its own body and the edges of the board. Notably, we implemented features like score tracking in the terminal and immersive sound effects through a piezo speaker. However, we were not fully satisfied with the initial results, and created a decorative board to screw our components into, creating a robust and appealing final product. This project allowed us to apply our knowledge of embedded systems and integrate various concepts into a fully functional game system. While we initially faced challenges with the LED matrix setup, we overcame them by utilizing an integrated matrix peripheral. Altogether, working on this project was a really fun and rewarding experience, and we look forward to working on more projects like this in the future.

## 7. Bibliography

- [1] “MAX7219 | Mbed,” os.mbed.com.  
<https://os.mbed.com/teams/Maxim-Integrated/code/MAX7219/docs/tip/classMax7219.html>  
(accessed Apr. 21, 2023).
- [2] “Serially Interfaced, 8-Digit LED Display Drivers” [Online]. Available:  
<https://cdn-shop.adafruit.com/datasheets/MAX7219.pdf> (accessed Apr. 21, 2023).
- [3] ArduinoModules, “KY-023 Dual Axis Joystick Module,” ArduinoModulesInfo, May 21, 2018. <https://arduinomodules.info/ky-023-joystick-dual-axis-module/> (accessed Apr. 21, 2023).
- [4] S. Lagishetty, “8x8 LED Matrix MAX7219 with Arduino Circuit & Code,” How To Electronics, Jan. 06, 2020. <https://how2electronics.com/8x8-led-matrix-max7219-arduino/>  
(accessed Apr. 21, 2023).

- [5] “Joystick | Mbed,” os.mbed.com, May 05, 2016.  
<https://os.mbed.com/teams/ELEC2645-201516/code/Joystick/docs/tip/> (accessed Apr. 22, 2023).
- [6] max.imagination, “Arduino Project - 8x8 LED Pong Game,” Instagram, Jan. 27, 2023.  
<https://www.instagram.com/reel/Cn61hkIDx5c/?igshid=YmMyMTA2M2Y=> (accessed Mar. 23, 2023).
- [7] Kevger, “GitHub - Kevger/MBed-STM32-Snake: Snake für den STM32 Nucleo-L031K6 mit MBed OS. Visualisierung durch ein 8x8 LED Matrix Display.,” GitHub.  
<https://github.com/Kevger/MBed-STM32-Snake> (accessed Apr. 22, 2023).
- [8] Carlos Miranda, “Embedded Systems: Snake Game,” YouTube. Jun. 07, 2021. [Online]. Available: <https://www.youtube.com/watch?v=vg8U1FO0HE0> (accessed Apr. 22, 2023).
- [9] “Full API list - APIs | Mbed OS 5 Documentation.”  
<https://os.mbed.com/docs/mbed-os/v5.15/apis/index.html> (accessed May 11, 2023).
- [10] html.alldatasheet.com, “1088AS Datasheet(1/1 Pages) ETC | LED 8x8 matrix 3mm 1088AS.” <https://html.alldatasheet.com/html-pdf/1179268/ETC/1088AS/61/1/1088AS.html> (accessed May 11, 2023).
- [11] “19/32 in. D x 5-1/2 in. W x 6 ft. H Cedar Dog-Ear Fence Picket”  
[Outdoor Essentials 19/32 in. D x 5-1/2 in. W x 6 ft. H Cedar Dog-Ear Fence Picket 327357 - The Home Depot](#) (accessed May 11, 2023).

## 8. Code Listings

Main.cpp

<https://github.com/BenTheurich/EmbeddedSnakeGame/blob/main/main.cpp>

Game.h

<https://github.com/BenTheurich/EmbeddedSnakeGame/blob/main/Game.h>

Game.cpp

<https://github.com/BenTheurich/EmbeddedSnakeGame/blob/main/Game.cpp>

Snake.h

<https://github.com/BenTheurich/EmbeddedSnakeGame/blob/main/Snake.h>

Snake.cpp

<https://github.com/BenTheurich/EmbeddedSnakeGame/blob/main/Snake.cpp>

Joystick.h

<https://github.com/BenTheurich/EmbeddedSnakeGame/blob/main/Joystick.h>

Joystick.cpp

<https://github.com/BenTheurich/EmbeddedSnakeGame/blob/main/Joystick.cpp>

## 9. Demonstration

Click this [link](#) to view our demonstration.