Bangladesh University of Engineering and Technology

Department of Computer Science and Engineering

CSE 316 - Project

Retro Snake

Submitted by:

Yeasir Khandaker (*1805105*)
Sk. Sabit Bin Mosaddek (*1805106*)
Md. Redwanul Karim (*1805111*)

Introduction

Retro Snake is a hardware implementation of the game *Snake*, one of the first games available on mobile phones. For this project, we used ATMega32 as the Microcontroller.

Implemented Features

The game is displayed on a grid(16×16) built with 4 8×8 bi-color LED matrices. There are 2 digital IR sensors to control the snake. In the game, we can move the snake head right or left by using these sensors. Whenever the snake eats a food, one point is added to the score and the length of the snake increases.

There are two modes in the game:

- Classic Mode
- Obstacle Mode

In classic mode, the game continues until the head of the snake collides with its own body.

In obstacle mode, there are 4 levels. When the snake eats two foods, the level increases except for the last level. The game ends when the snake's head collides with its body or a obstacle.

The score and the high score is displayed on a 16×2 I2C LCD display. We have used a buzzer to indicate that the snake has eaten a food.

Instruments

Item	Quantity
ATMega32	1
IC 74154(4 to 16 decoder)	2
IC 7404(NOT Gate)	1
Bi-color 8×8 LED Matrix	4
16×2 I2C LCD Display	1
IR Sensor Modulde	2
Buzzer	1

Table 1: List of Instruments

Circuit Diagram

In the circuit, the outputs of one of the decoders are connected to red color pins of the LED matrices. And the outputs of the other decoder are to be connected to the green color pins of the LED matrices. Since there are no bi-color LED matrices available on Proteus software the output connections of the second decoder are omitted in Figure: 1.

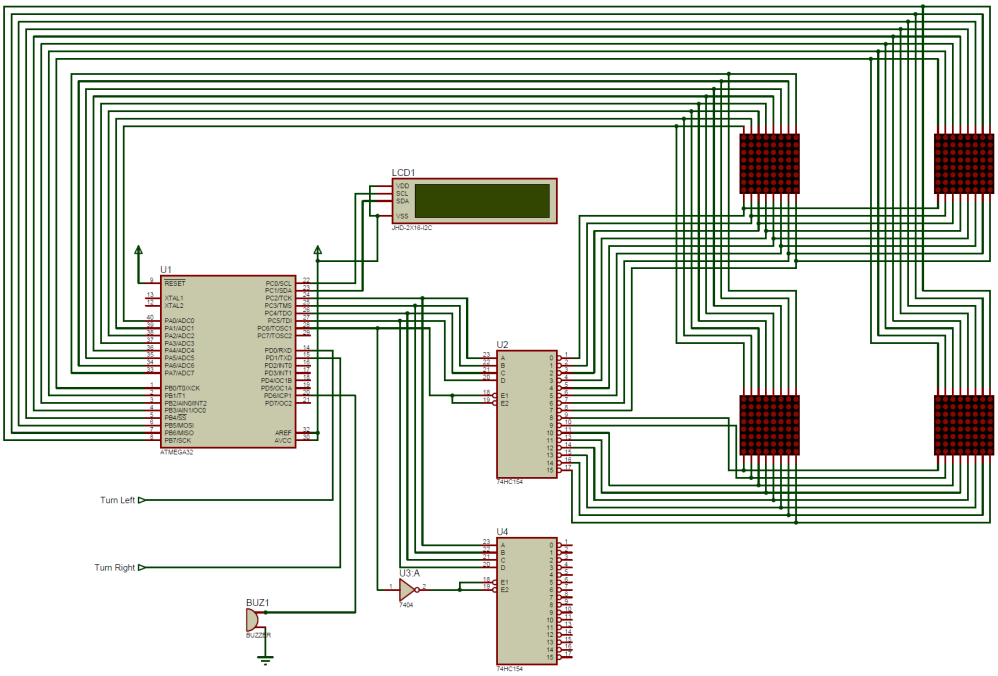

Figure 1: Circuit Diagram

Problems and Challenges

- **Power Supply Issues**
  Driving all the modules together required a significant amount of power. A 9V battery could not supply this much power. Output of LED matrices were very dim when we used a 9V battery as our power source. Sometimes the outputs were unusual.

  The problem was solved by using a power bank as our power source.

- **Low Output Voltage on PORT A**
  The LED matrices that were supplied voltage from PORT A were a bit dim. We found that the output voltage from the pins were low. We went through the documentation of ATMega32 and found that some voltage was supplied to PORT A from AVCC. So we connected the AVCC to power source and the problem was solved.

- **Using LCD Display**
  Configuring LCD Display that uses parallel communication required too many pins of the ATMega32. To multiplex LED matrices we had already used up most of the pins. So we did not have enough pins left to configure the LCD display.

  This problem was solved by using the I2C adapter to enable serial communication between the Microcontroller and the LCD display. We needed only 2 pins from PORT C to connect the I2C adapter. Then finding resources to interface I2C with ATMega32 was another challenge.

- **Configuring Range of IR Sensors**
  The range of the IR sensors could be controlled by the potentiometer attached to each of the sensors. It was difficult to decide at what distance the sensors should detect our gestures.

  We had to do some trial and error to find the perfect range.

- **Debouncing IR Sensors**
  We faced problem of bouncing with the IR sensors. The sensors sometimes registered more than one input for one gesture of the player. This problem was solved by including some delay between registering two inputs.

- **Issue with Wires**
  Due to faulty wires and loose connection between wires and breadboard, at first we did not get expected output. We had to test and reconnect many wires while debugging the circuit. We had to use a lot of wires for LED matrices, so this process was quite tough.