



# CLASSIC SNAKE GAME

- Sai Keerthi Doma  
00450887

# INTRODUCTION

Developed a classic snake game on Arduino with an ATmega328P microcontroller and assembly language programming. The game will be played on an 8x8 LED matrix (MAX7219) display and controlled with push button to change directions

# REQUIRED HARDWARE COMPONENTS

- ATmega328P microcontroller
- 8x8 LED matrix display (MAX7219)
- Push button
- Some connecting wires

# MAKE A 1 PIXEL SNAKE

Let the Variable for storing 1 pixel snake are:

- $Y+10 \rightarrow Snake\_x$
- $Y+11 \rightarrow Snake\_y$

Let initialize the default position for the snake is at (2,5)

So, LED should on at position(2,5) on 8\*8 LED Matrix



# MOVE THE SNAKE

Following are the variables that are responsible for changing the direction of Snake.

$Y+0 \rightarrow$  represents  $dx$      $Y+1 \rightarrow$  represents  $dy$

- $dx=1, dy=0 \rightarrow$  moves right
- $dx=-1, dy=0 \rightarrow$  moves left
- $dx=0, dy=1 \rightarrow$  moves down
- $dx=0, dy=-1 \rightarrow$  moves up



# MOVE THE SNAKE

How to create the illusion that the snake is moving?

Turn OFF the LED at the previous position of the snake pixel and turn ON the LED at the current snake pixel after adding  $dx$  to  $x$  and  $dy$  to  $y$ .

- $x(\text{curr\_snake\_x}) = x(\text{prev\_snake\_x}) + dx$
- $y(\text{curr\_snake\_y}) = y(\text{prev\_snake\_y}) + dy$

Delay for about a second for turning LED ON at new snake pixel.



# MOVE THE SNAKE

Let give the default direction for snake

$dx \rightarrow 1$     $dy \rightarrow 0 \Rightarrow$  moves right

So, the snake moves right till the gamer presses the push down button to change the direction of the snake.



# GAME OVER

If the snake touches the edge then game should stop and turn ON LED on entire display.

Following are the four conditions for Game Over:

1. Snake x ( $Y+10$ ) is at 8 or more
2. Snake x is at -1 or less
3. Snake y ( $Y+11$ ) is at 8 or more
4. Snake y is at -1 or less



# BUTTON RESPONSE

When a button is pressed, the direction of movement should turn counter-clockwise.

- Right moving snake to upward ( $dx=1, dy=0 \Rightarrow dx=0, dy=-1$ )
- Upward moving snake to left ( $dx=0, dy=-1 \Rightarrow dx=-1, dy=0$ )
- Left moving snake to downward ( $dx=-1, dy=0 \Rightarrow dx=0, dy=1$ )
- Downward moving snake to right ( $dx=0, dy=1 \Rightarrow dx=1, dy=0$ )

# GENERATE APPLE AT RANDOM LOCATION

The goal of the snake is to eat the apple

The apple is a stationary dot that appears at a random location and is moved when the snake eats it.

Following are the variables that hold the x,y position of the apple:

$Y+2 \Rightarrow apple\_x$        $Y+3 \Rightarrow apple\_y$

# GENERATE APPLE AT RANDOM LOCATION

If snake eats the apple then shift the apple to random location:

Snake eats apple:

- $Y+10(\text{snake\_x}) == Y+2(\text{apple\_x})$
- $Y+11(\text{snake\_y}) == Y+3(\text{apple\_y})$

Update apple:

```
apple_x = apple_x + 23
```

```
apple_x = apple_x AND 7
```

```
apple_y = apple_y + 23
```

```
apple_y = apple_y AND 7
```

# MAKE A 3-PIXEL SNAKE

Let the Variable for storing 3 pixel snake are:

- Y+10 -> Snake's head x- coordinate
- Y+11 -> Snake's head y- coordinate
- Y+12 -> Snake's middle x- coordinate
- Y+13 -> Snake's middle y- coordinate
- Y+14 -> Snake's tail x- coordinate
- Y+15 -> Snake's tail y- coordinate

# MOVE THE 3-PIXEL SNAKE

To create the illusion that the 3-pixel snake is moving?

Turn off pixel at tail ( $Y+14 \Leftarrow OFF, Y+15 \Leftarrow OFF$ )

Make previous middle to current tail ( $Y+14 \Leftarrow Y+12, Y+15 \Leftarrow Y+13$ )

Make previous head to current middle ( $Y+12 \Leftarrow Y+10, Y+13 \Leftarrow Y+11$ )

Turn on pixel at new head ( $Y+10 \Leftarrow Y+10+(Y+0), Y+11 \Leftarrow Y+11+(Y+1)$ )



# MAKE IT FASTER

To make the snake move faster after eating the apple, reduce the iteration value in the delay loop by 20.

Consider Y+5 variable to hold the iteration value.

When snake eat apple  $\Rightarrow Y+5 = (Y+5) - 20$



**THANK YOU**