

# Conway's Game of Life Displayed on 32x32 LED matrix using Minimal Drivers

Mason Legere, Joshua Muir

April 3, 2019

# What is Conway's Game of Life?

In the field of chaos theory, systems are developed to see how small changes in input can lead to unpredictable changes in the output.

- ① There should be no explosive growth.

# What is Conway's Game of Life?

In the field of chaos theory, systems are developed to see how small changes in input can lead to unpredictable changes in the output.

- ① There should be no explosive growth.
- ② There should exist small initial patterns with chaotic, unpredictable outcomes.

# What is Conway's Game of Life?

In the field of chaos theory, systems are developed to see how small changes in input can lead to unpredictable changes in the output.

- ① There should be no explosive growth.
- ② There should exist small initial patterns with chaotic, unpredictable outcomes.
- ③ There should be no potential self replication.

# What is Conway's Game of Life?

In the field of chaos theory, systems are developed to see how small changes in input can lead to unpredictable changes in the output.

- ① There should be no explosive growth.
- ② There should exist small initial patterns with chaotic, unpredictable outcomes.
- ③ There should be no potential self replication.
- ④ The rules should be as simple as possible.

## Rules of the Game:

- ① Any live cell with fewer than two live neighbours dies, as if by underpopulation.

## Rules of the Game:

- ① Any live cell with fewer than two live neighbours dies, as if by underpopulation.
- ② Any live cell with two or three live neighbours lives on to the next generation.

## Rules of the Game:

- ① Any live cell with fewer than two live neighbours dies, as if by underpopulation.
- ② Any live cell with two or three live neighbours lives on to the next generation.
- ③ Any live cell with more than three live neighbours dies, as if by overpopulation.

## Rules of the Game:

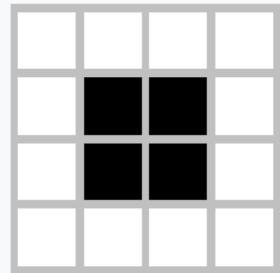
- ① Any live cell with fewer than two live neighbours dies, as if by underpopulation.
- ② Any live cell with two or three live neighbours lives on to the next generation.
- ③ Any live cell with more than three live neighbours dies, as if by overpopulation.
- ④ Any dead cell with exactly three live neighbours becomes a live cell, as if by reproduction.

# Examples (Still Lifes)

## Still Lifes

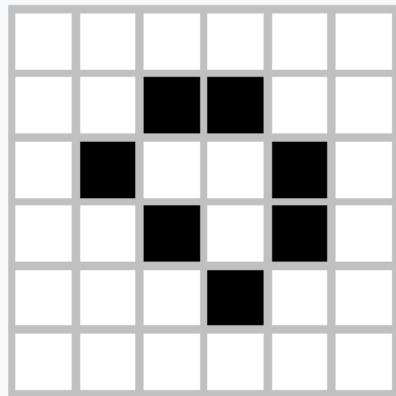
Clusters that do not change from one generation to the next

Block

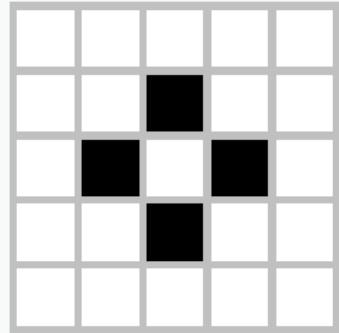


## Examples (Still Lifes)

Loaf



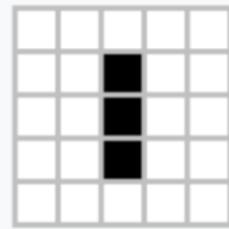
Tub



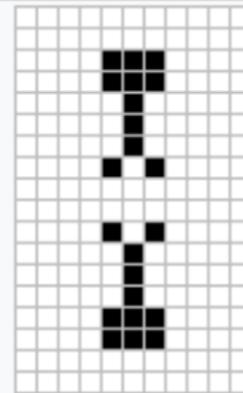
## Examples (Oscillators)

Periodic clusters that have a finite period before returning to their initial state.

Blinker  
(period 2)



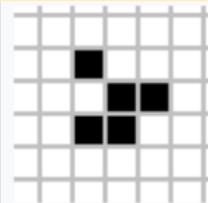
Penta-decathlon  
(period 15)



## Examples (Spaceships)

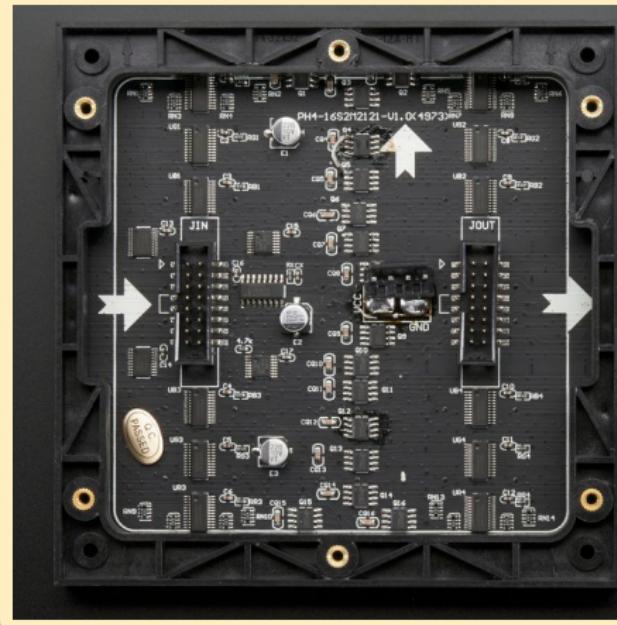
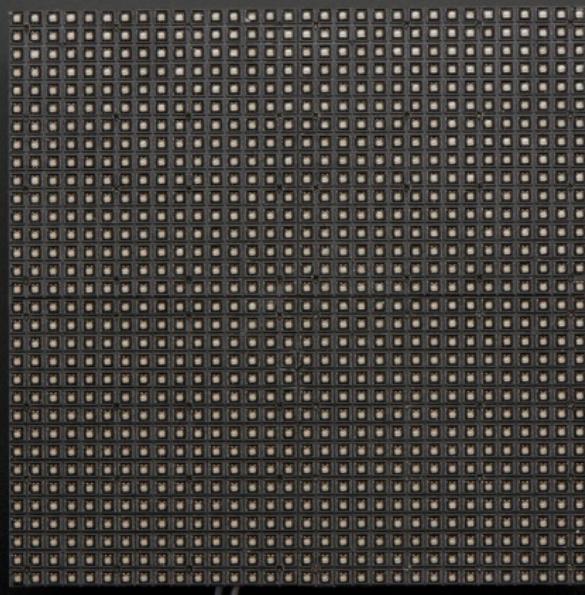
Clusters that translate themselves across the grid.

Glider



# Implementation

Here is the matrix we used for the project.



# How the Matrix Works

- ① There is no proper documentation for the matrix written in English.
- ② Just a big shifter with three separate clks each controlling nested layers of shifters.
- ③ These must be controlled in a precise order after completing exactly the right steps or else nothing the matrix will not turn on.

# Implementation

With no documentation or help online, I emailed this guy: (Bob Davis)



Bob Davis

to me ▾

Mason,

Are you resetting the display by writing to registers 11 and 12 first?

I did not see that in your code.

<http://bobdavis321.blogspot.com/2019/02/p3-64x32-hub75e-led-matrix-panels-with.html>

The reset code is at the bottom.

\*\*\*

# Implementation

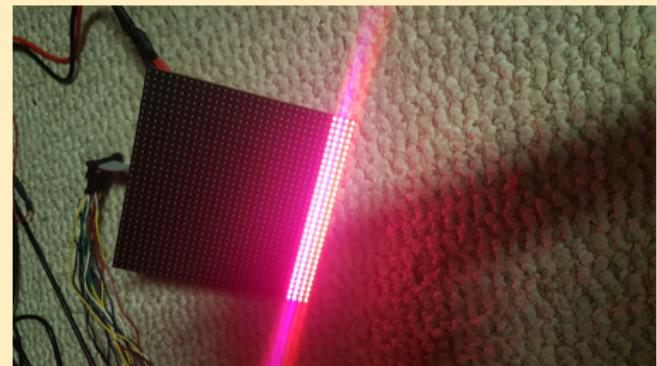
He referenced me to one of his projects where he had the code:

```
PORTB=0;
// Send Data to control register 11
for (int l=0; l<MaxLed; l++){
    int y=l%16;
    PORTD = 0x00;
    if (C12[y]==1) PORTD=0xFC;
    if (l>MaxLed-12){ PORTB=7; PORTB=6; }
    else{ PORTB=1; PORTB=0; }
}
PORTB=0;
```

This code was nonsensical and did nothing. Bob was running a newer model of the LED board that was from exactly the same company but ran completely different.

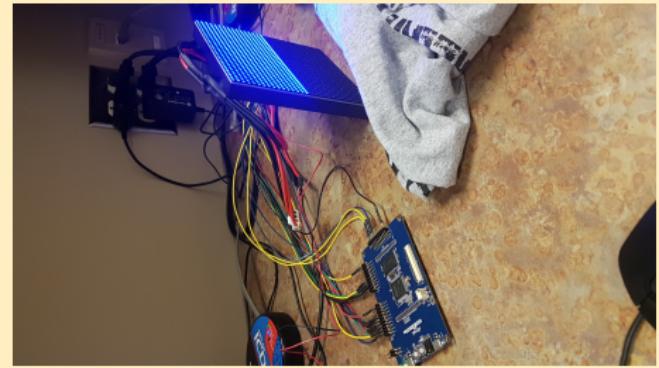
# Implementation

After two weeks of nothing the only thing that made the Matrix light up was accidentally plugging in the OE pin (one of the latches) into VCC.



# Implementation

Following this we dealt with endless broken drivers leading to:



## Additional Problems

- ① Flaky wires leading to sporadic display (my fault)
- ② One broken SAM4S (possibly my fault??)
- ③ {r1} should have been an {r3}

dead\_eq3:

```
mov r8, r3
mov r9, r0
mov r10, 1
bl set_pixel
b end_of_loop
```

# Programming the Game of Life

- ① Every line of code is written in assembly
- ② Originally we had a buffer from 0x2000000 to 0x2000200 that holds all the information for the matrix
- ③ Each byte in this buffer has the form  $[0, 0, R_1, G_1, B_1, R_0, G_0, B_0]$
- ④ The first two bits were then used to save alternating values in between generation updates.

```
dead_eq3:  
    mov r8, r3  
    mov r9, r8  
    mov r10, 1  
    bl set_pixel  
    b end_of_loop  
  
square_is_alive:  
    cmp r2, #2  
    blt alive_lt2  
    cmp r2, #3  
    bgt    alive_gt3  
  
    // keep the pixel alive  
    mov r8, r3  
    mov r9, r8  
    mov r10, 1  
    bl set_pixel  
    b end_of_loop  
  
alive_lt2:  
    mov r8, r3  
    mov r9, r8  
    mov r10, 0  
    bl set_pixel  
    b end_of_loop  
  
alive_gt3:  
    mov r8, r3
```

# Programming the Game of Life

- ① We realized additional colours are not needed.
- ② The buffer format changed to  $[0, 0, 0, R_{1_1}, R_{1_0}, 0, R_{0_1}, R_{0_0}]$
- ③ This was after the quick realization that next generation has to be based solely on the previous generation and not a combination of its own development.
- ④ Therefore the system works modulo 2, alternating between read and writes at respective entries.

## Relates to CPSC231

- The project is written entirely in assembly language.
- This matrix is an I/O device which we had to create all drivers for.
- This project was accomplished with the Atmel SAM4S being the controller for the matrix.

## References

- All images for Conway's Game of Life are from  
[https://en.wikipedia.org/wiki/Conway%27s\\_Game\\_of\\_Life](https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life)
- The images of the board are from  
<https://www.adafruit.com/product/2026>
- And the picture of Bob is from his YouTube channel,  
[youtube.com/user/bobdavis321](https://youtube.com/user/bobdavis321), in his video titled *Projectors For Africa*