

Stefan Crigler  
Andrew Cizas  
ECE 153B  
Tuesday 7:00-10:00 pm

## Project Proposal: LED Minesweeper

### **Overview:**

We plan to design the game Minesweeper on an 8x8 LED matrix display. The player will use a Wii nunchuk controller to control a cursor which will allow the user to traverse the 8x8 matrix and select which locations they wish to unveil. The goal of the game is to unveil all locations that are not bombs, using the help of special flags to mark suspected bomb sites. We will use an RGB LED display, and designate colors to the cursor, bomb selections, flags, and colors on a spectrum representing how many bombs are bordering that spot. When a safe location is unveiled, all bordering safe locations are also unveiled. The player can use the information provided by unveiled locations to flag potential bomb locations using a button on the nunchuk. A display on the computer screen will alert players to when they win/ lose. At the beginning of every game, the bomb locations will be randomly assigned and the number values for each safe location denoting the number of neighboring bombs will be stored in each cell.

### **Peripherals:**

We plan on using the following peripherals:

8x8 RGB LED matrix  
MAX7219 Driver  
Wii Nunchuk  
Nunchucky (Wii nunchuk breakout adapter)

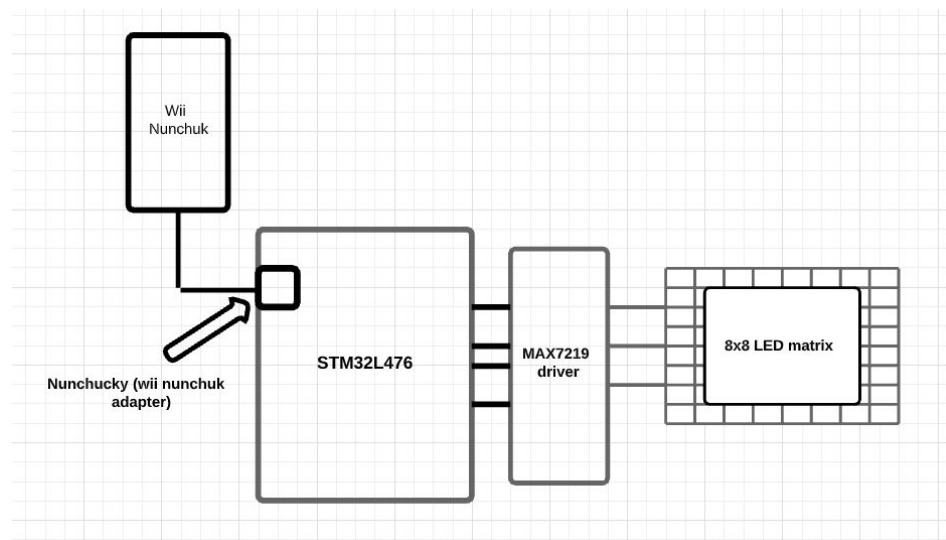
### **Serial Interface Protocols:**

UART  
I2C

### **Group Responsibilities:**

Stefan will handle much of the game state setup, such as building the matrix under the hood and creating the logic and algorithms to build the bomb locations and the numbers for safe spaces. Andrew will handle pin initialization and setup as well as wiring. Stefan and Andrew will split the logic to connect the max driver and the wii nunchuk, as well as website duties and documenting the process.

### **Block diagram:**



### **General Software Structure:**

STM pin configuration: Need 4 pins to control the MAX LED driver, which will in turn control the 8x8 LED matrix. This can be done with 3 GPIO pins and a clock pin, one of which will be initialized as input, one as output, one will be a pin which loads values, and finally the clock. The nunchuk controller, using a I2C configuration, is connected using a 4-pin adapter that requires 4 GPIO pins. These pins are pins for ground, data, clock, and voltage input. Code will be written in the following different areas:

### **Minesweeper game initialization:**

An 8x8 grid will be instantiated, and an appropriate number of bomb cells (approx  $\frac{1}{3}$  of the total cells) will be randomly decided. Then, a for loop will run through each cell, and assign either a number 0-8 or bomb value to each cell determining if it is a bomb or how many bombs border it. The cursor starts out at the top-left LED. It lights up flashing with a unique color while all other LEDs are not lit up. This is then the initial state of the game.

### **Minesweeper game and cursor:**

The nunchuk will be programmed to act as a joystick for the cursor, which when the joystick is used will move the cursor throughout the 8x8 LED grid. Interrupts will be used when the cursor selects a cell, and this interrupt will tell the game 1. Whether the location was a bomb or not, and 2. If it was not a bomb how many neighboring safe spaces to unveil. Interrupts will also be used when the game is won or lost. An interrupt will trigger if a bomb is hit or when there are no more safe spaces to unveil, meaning the player has won. The interrupt handler then changes the state to display the number or the bomb beneath the chosen LED. Players can also plant flags on LEDs by selecting the flag button, which also triggers an interrupt and causes the interrupt handler to change the game state. Additionally, interrupts are also used to flash the LED of the cursor.

**Website:** <https://sites.google.com/view/ledminesweepergame/home?authuser=1>