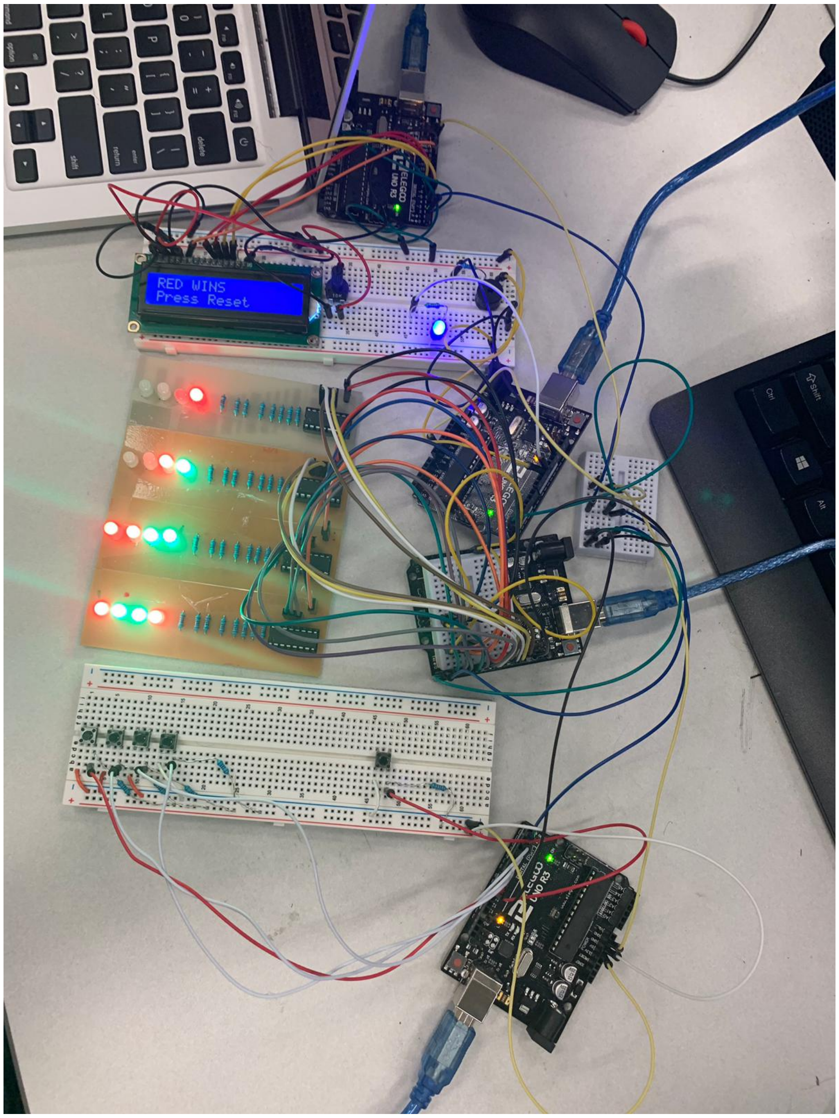
Connect Four with Arduino

Computer Design CS-362 Final Project



Group Members

Fatima Kahack
Saema Ansari
Michelle Baginski
Kirun Haque

Project Name: Connect Four

Abstract

The original project design was to implement a game which all group members have previously coded in Program Design II (CS-141). Newton's Game is a two-player game in which there 4 rows and 8 columns. Each player chooses a column to place their game piece, or in our case, light up the corresponding LED for the board position. Players can choose to rotate the pieces of any column instead of placing a new piece on the board. Due to cross-wiring when creation of the printable circuit board for our game, we chose to modify it so that the board is 4x4, and there is no column rotation. Any player to get four in a row vertically, horizontally, or diagonally wins.

Original Work Description

The development and production of a 4x4 LED matrix PCB that we designed for our game grid. We have added features like displaying messages to the players using an LCD and communicate errors using a buzzer. When a player wins, a separate LED light flashes to show that there's a winner and the game is over.

Detailed Project Idea

Implementation of interactive hardware to function with the game code. In our game, two Arduinos were used with 2 external devices per Arduino, 1 Arduino had 16 external devices, and the final Arduino with 5 external devices to complete our design.

Responsibility per Arduino

1. LCD and potentiometer
2. Buzzer and 1 Blue LED
3. 5 pushbuttons
4. 16 RGB LED lights

I/O Devices

- 16 RGB LED lights representing a spot on the game board (row, column)
- 1 blue LED light that will illuminate when there is a final winner
- 16x2 Liquid Crystal Display (LCD) that displays a winner and displays the current player's turn
- 4 push buttons representing each column. When one is pressed, the lower most LED light in that column will illuminate when a player makes their choice
- 1 pushbutton to RESET the game board
- Buzzer that will signal a tone when a column that is already full has been chosen

The 16 RGB LED lights will represent the game board and 16 possible game moves. The lights will be alternating between the colors red and green, representing each player's turn.

The LCD is used for a way to communicate to the players in the game. It will display whose turn it is (red or green), if the column is full, and which player won the game if there's a winner.
The blue LED light will be used to signal to the players that a winner has been found, and the game is over.

4 of the 5 push buttons for this project will be used to represent the 4 columns in the game so the players can choose which column they would like to place their pieces.

The last push buttons will be used to reset the game after there's a winner or during a game in case one of the players are a sore loser and wants to start the game all over. This turns off all LED lights on the board and resets the game code, emptying out each row and column.

Lastly, the buzzer will be used for when a player tries to play their turn in a full column. The buzzer will sound, and the LCD will display that the column is full.

Communication Details

We used serial communication. TX (digital pin 1) is the sender and RX (digital pin 0) is the receiver. Below details all communication functions per the 4 microcontrollers.

Arduino 1 controls the push-buttons. Four are used to represent gameboard columns, and the fifth button allows to reset the game to start a new round. It sends player input to the LED-board and does not receive anything. For every button that is pressed, data is sent to the Arduino 2, controlling the LED matrix. The code parses the received data, which is a string that is either the column number or "RS" which stands for reset.

Arduno 2 controls the LED matrix. It sends information about the game state to Arduinos 3 and 4 which update the buzzer, winning LED light, and the LCD display. It receives input from the pushbuttons from Arduino 1, which is which will in turn light up an LED in one of the columns where the player chose to place their piece.

Arduino 3 controls the LCD display and the potentiometer. It does not send data to any other Arduino. From the LED board, it receives the state of the game as a string, which is parsed.
The string indicates:

1. "R" or "G" for red player or green player's turn
2. "F" indicates a column is full
3. "GW" (Green Winner) or "RW" (Red Winner)

This Arduino appropriately updates the LCD screen with a message to the user for one of the above states.

Arduino 4 controls the buzzer and the winning LED; it will send nothing. Its job is to receive signals from the LED board that give information about the current game state. The code for this Arduino will parse data that received from the board Arduino. The data will be a string which can be "W" indicating there's a winner and to flash the light, or "F" indicating the column is full so that the buzzer can go off.

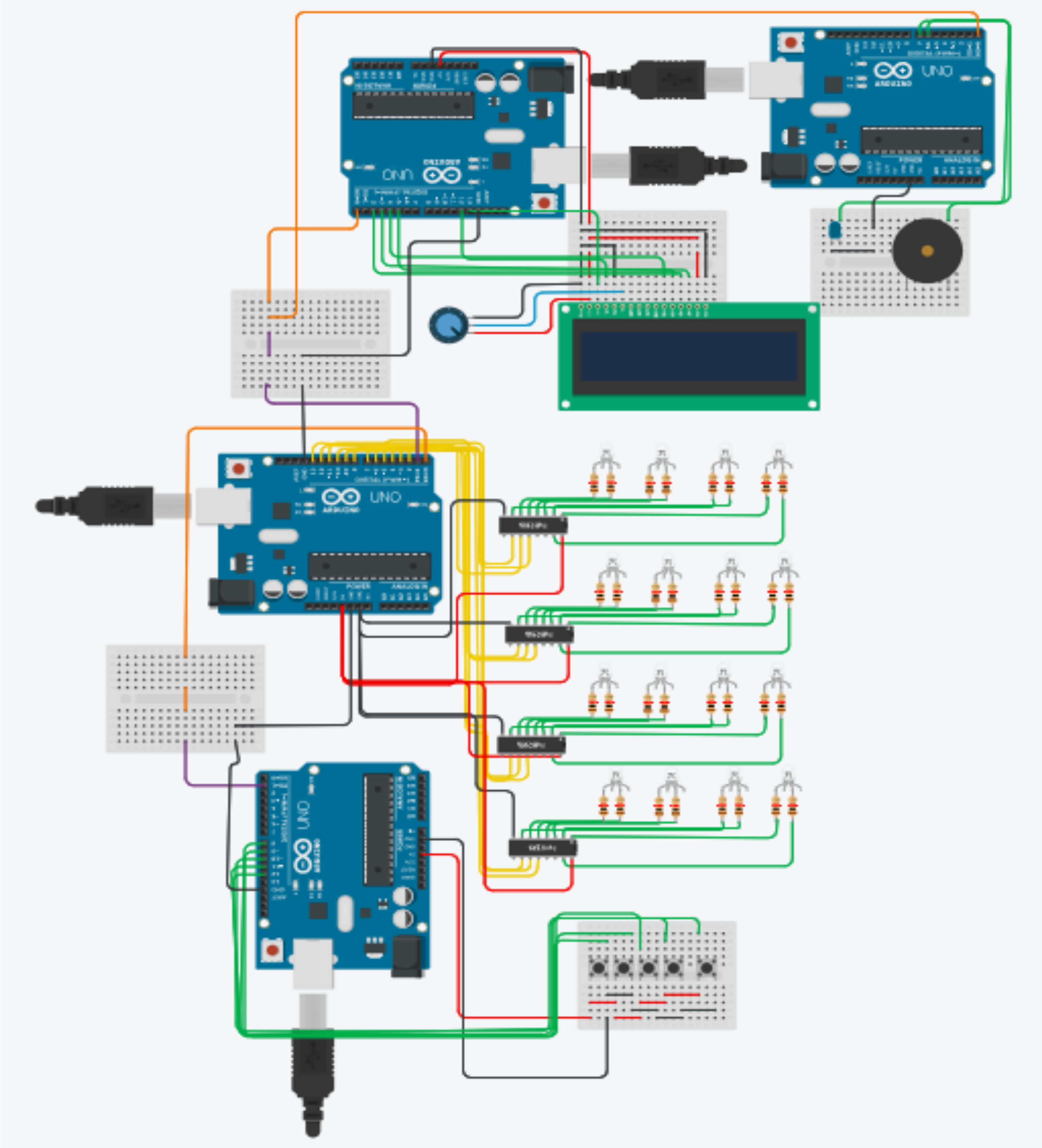
List of Materials Used

1. 16 RGB LED lights
2. 1 Blue LED
3. 4 shift registers
4. 4 16-pin sockets
5. 20 pinheads
6. 40 resistors
   6 10k resistors for the LCD and 5 push buttons
   34 220 ohm resistors for 16 RGB LEDS, 1 standard LED, and 1 buzzer
7. 1 buzzer
8. 5 pushbuttons
9. 1 16x2 LCD display
10. Potentiometer

List of References

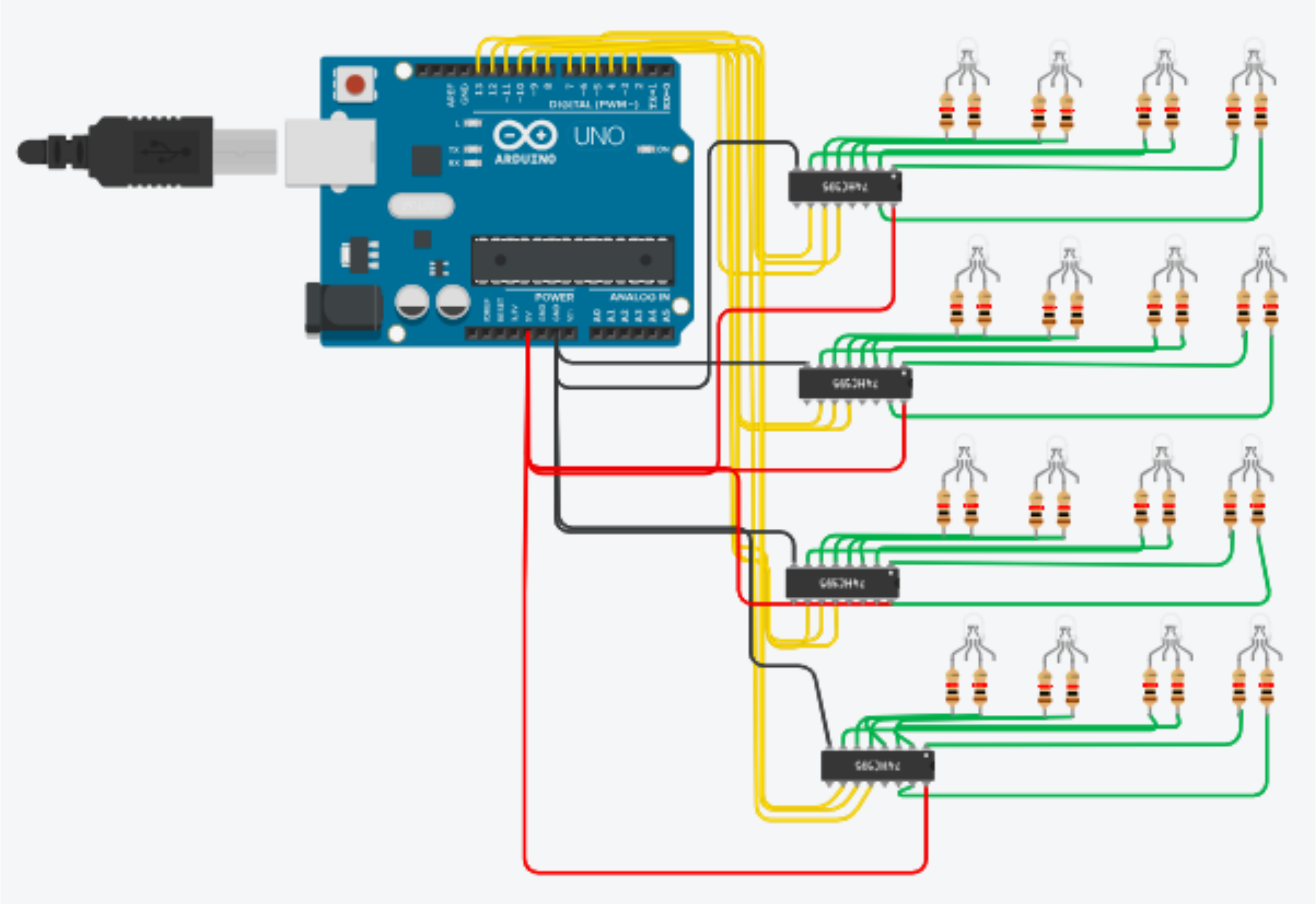
1. Using a 74HC595 to control a LED Matrix – Arduino Playground – Medium
2. RGB LED Matrix: 5 Steps
3. 3x 74HC595 registers driving RGB LEDs with common anode---
4. How to Set Up an IR Remote and Receiver on an Arduino - Circuit Basics
5. LCD_Base_bb_Fritz.png (1845×1020)

High Level Circuit Sketch

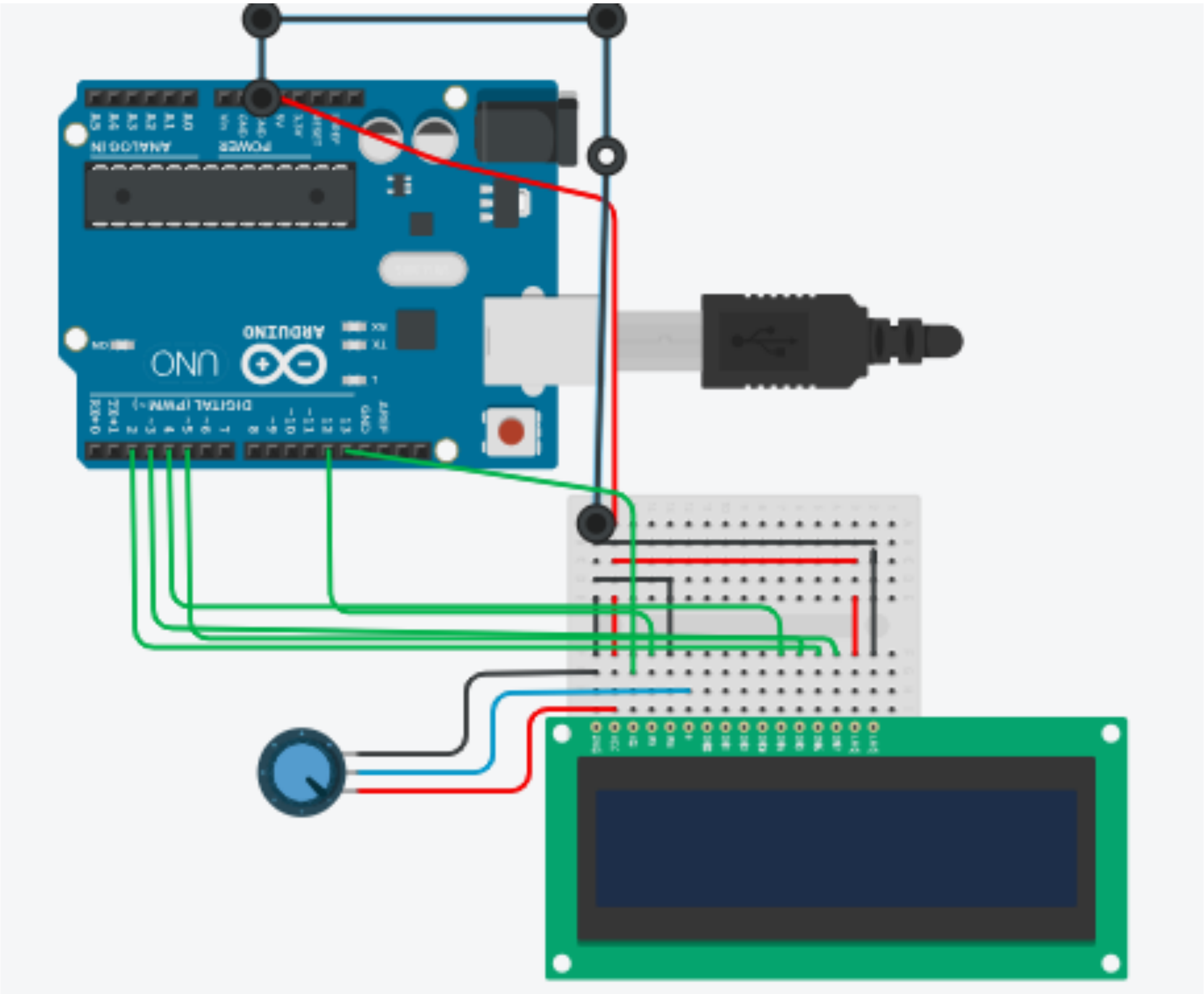


Lower Level Sketches

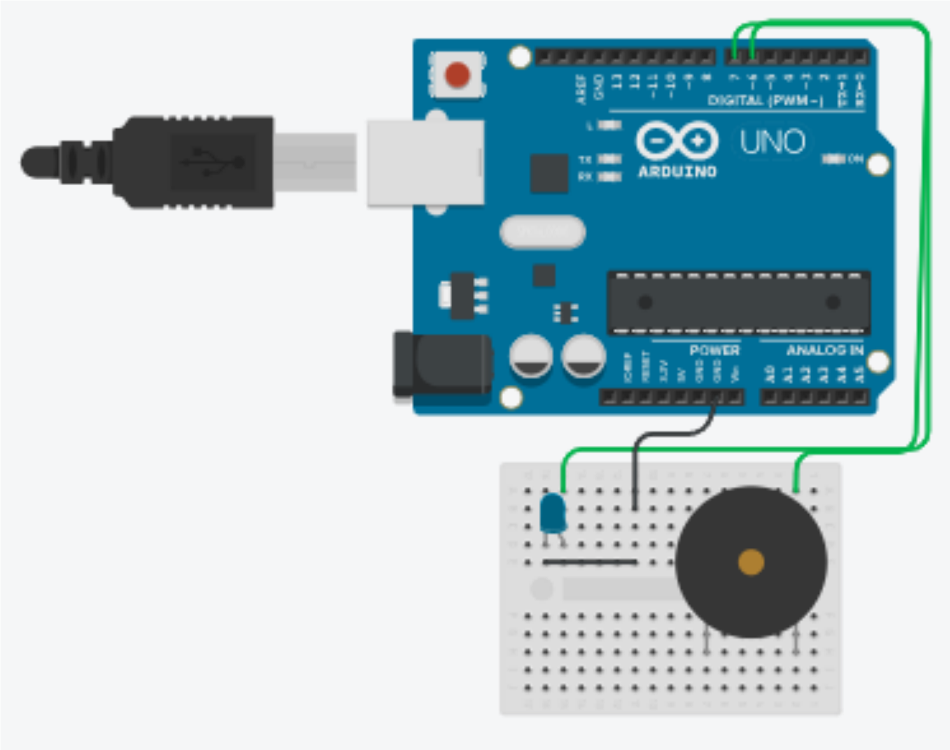
LED Board



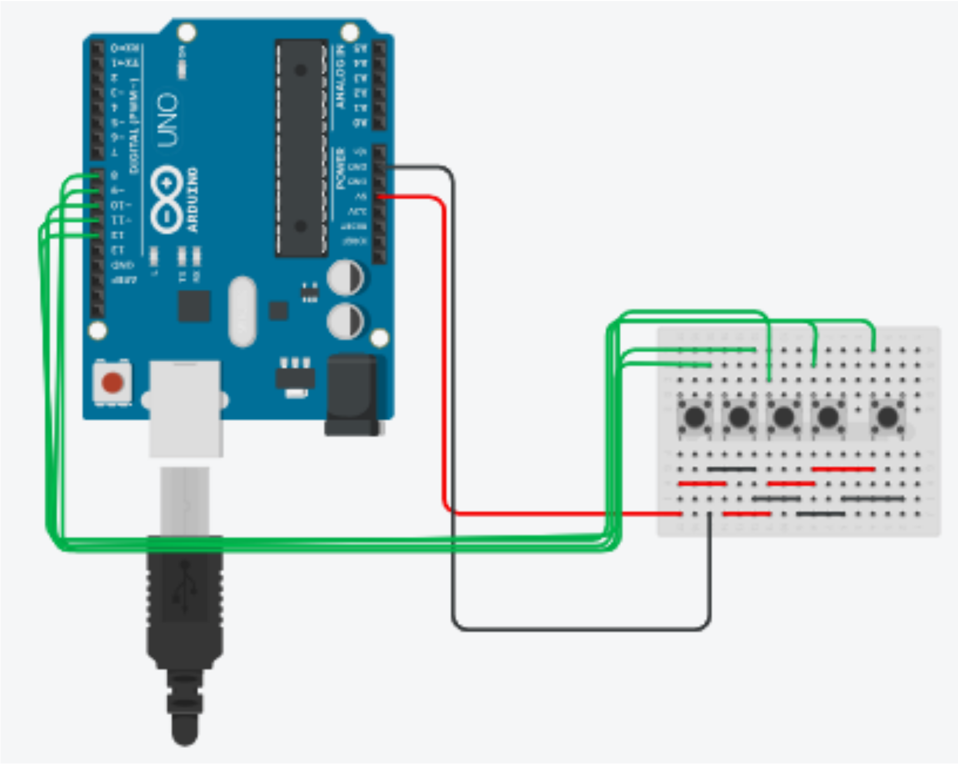
LCD and Potentiometer



LED and Buzzer



Pushbuttons



DIGITAL (PWM ~)
UNO
ARDUINO
POWER
ANALOG IN