Chip Limeburner

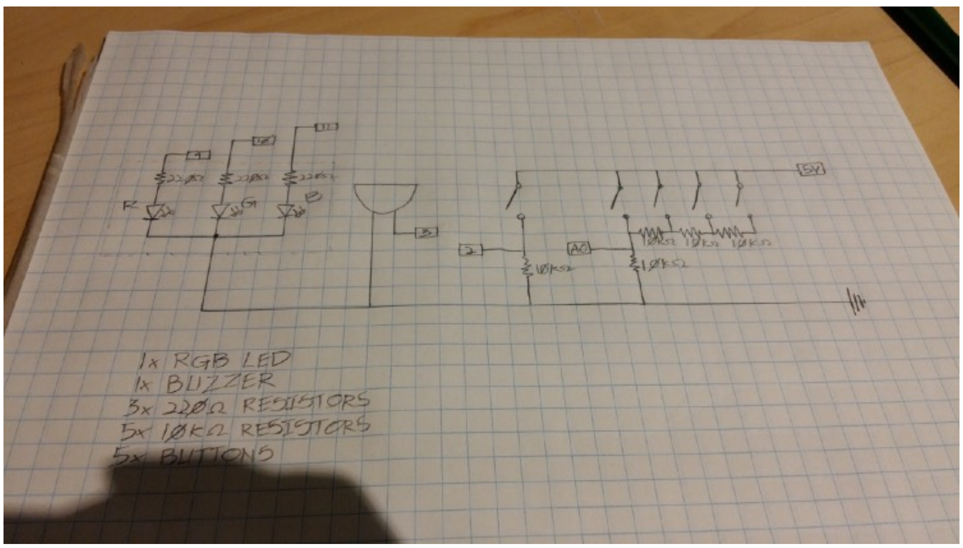
Student ID: 40177255

CART 360

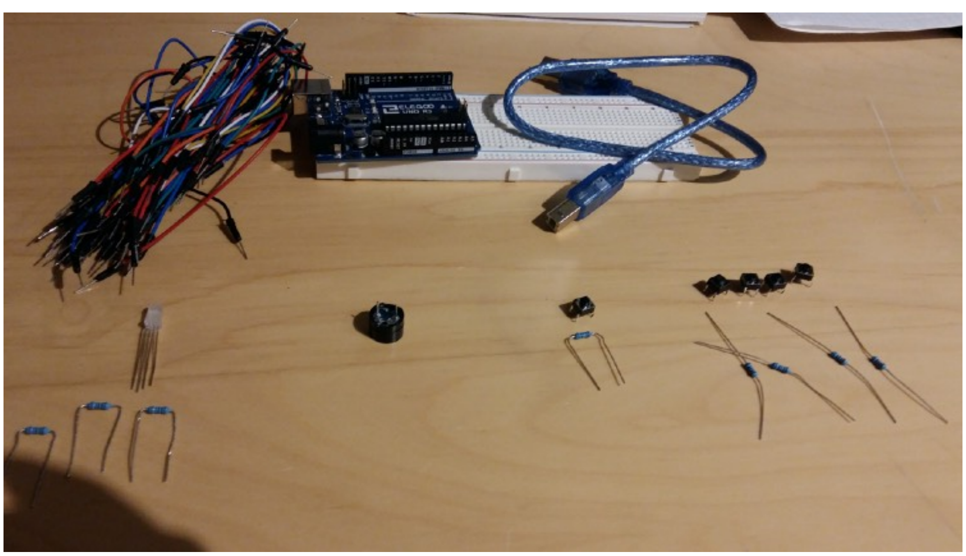
Oct. 15, 2021

<u>Etude 2</u>

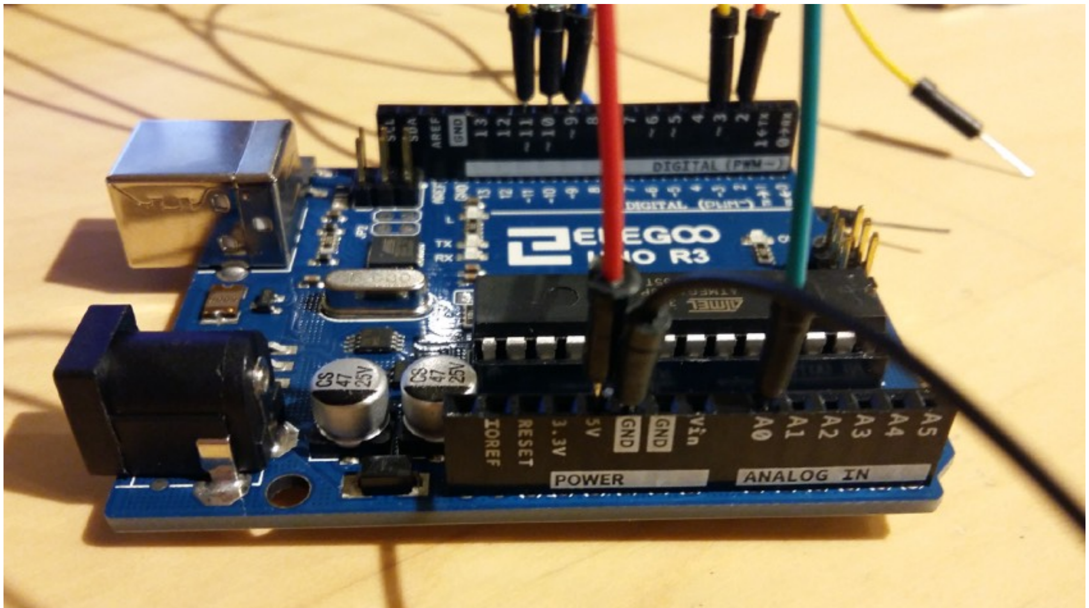
1) The first step I took in assembling the circuit was to draw a quick wiring diagram. This allowed me to think through how the various components were connected as well as verify values for resistors and identify components that would need to be retrieved for the project.

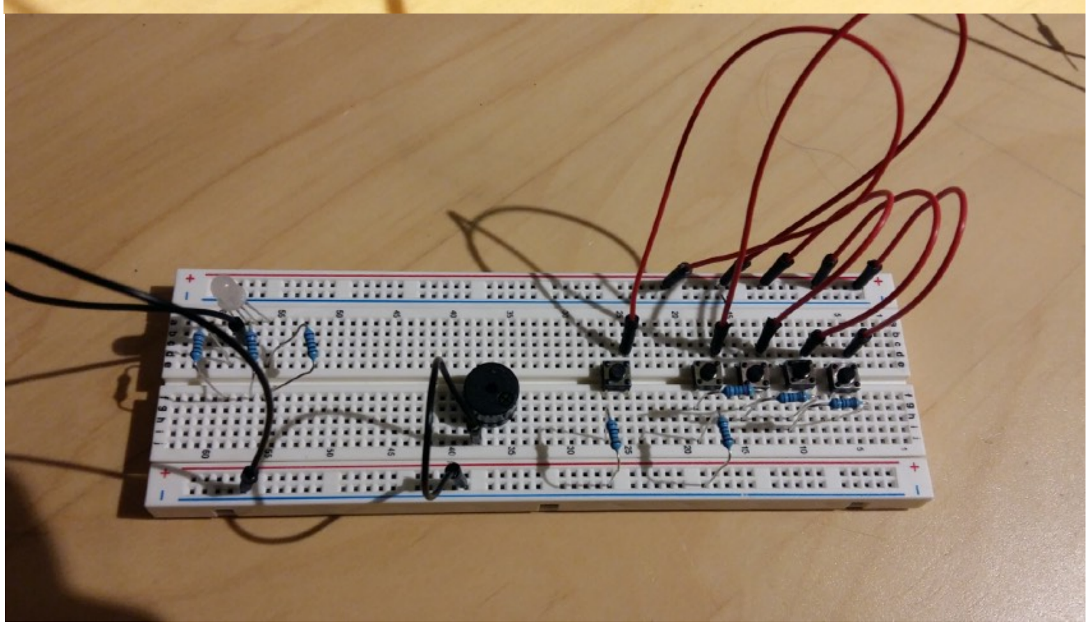


The next step was the retrieve the components that would be necessary for the project, grouping them loosely by the sub-circuit to which they would belong (mode button, resistor ladder, buzzer, and RGB LED).



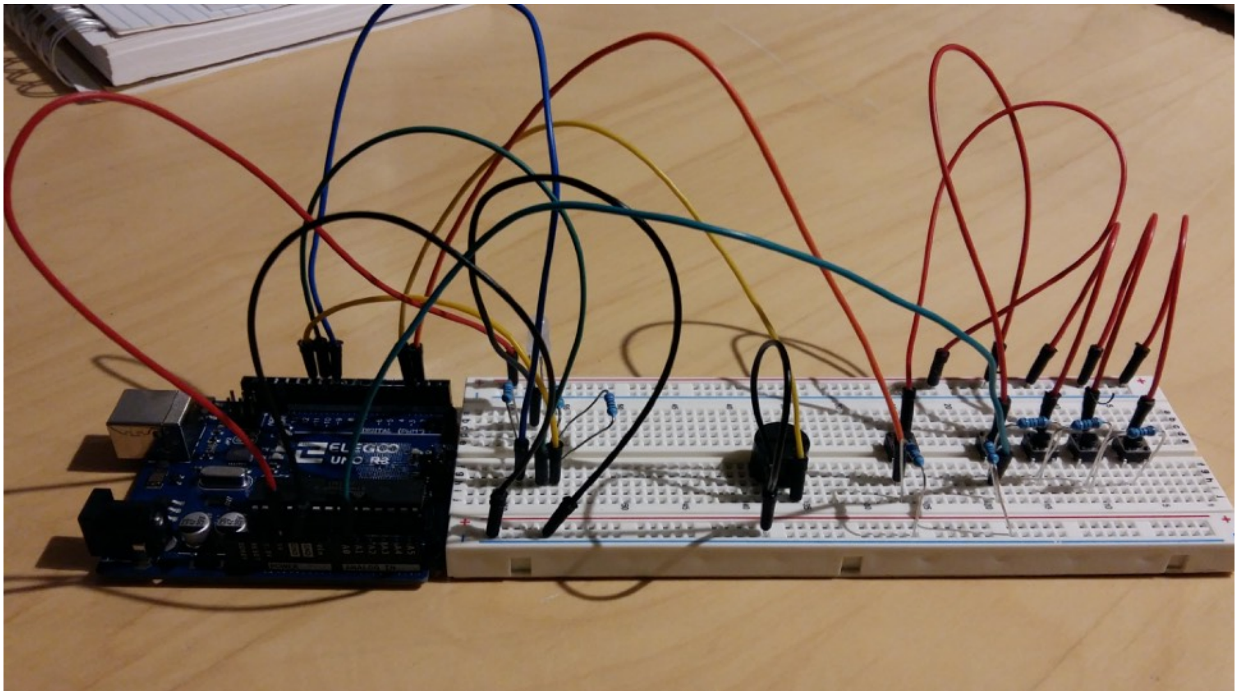
Following the retrieval of parts, I began assembling the circuit, separately combining the components appropriately on the breadboard, as well as inserting jumper wires into the pins of the Arduino that would be used.

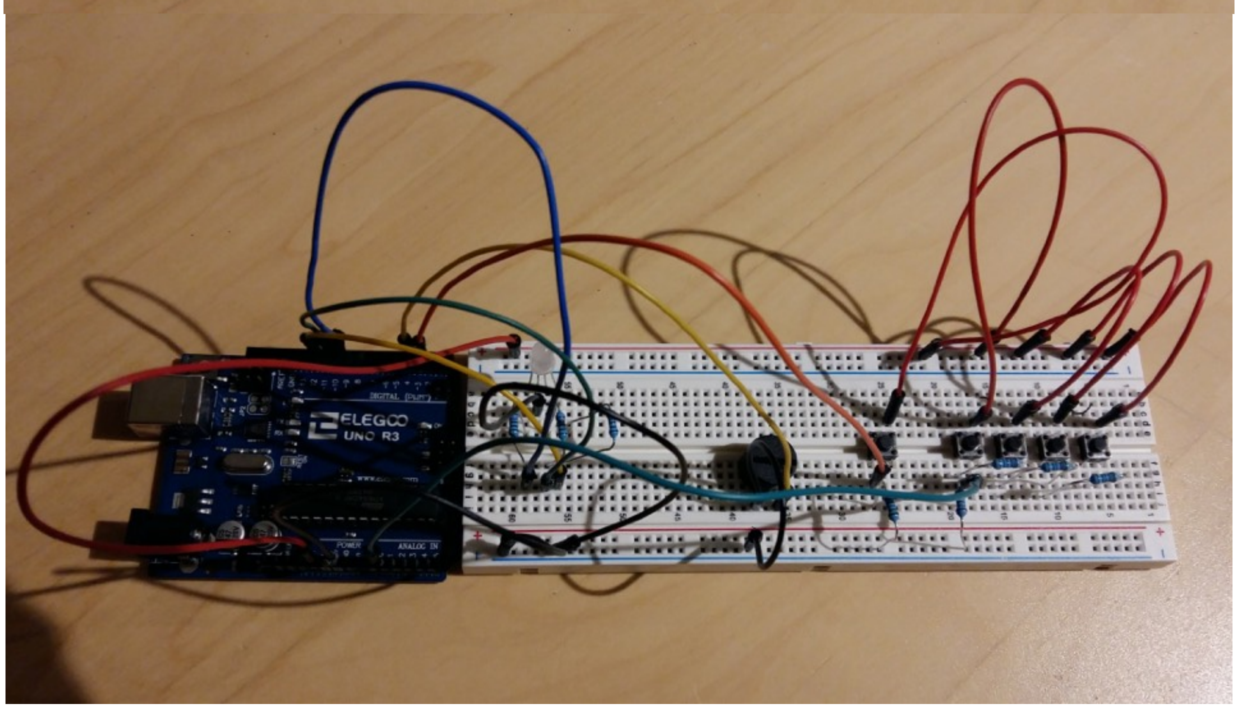




After the Arduino and breadboard were assembled separately, I took a moment to verify all connections were well-fitted and in the right order, before finally plugging the wires from the Arduino into the breadboard and proceeding to the stage of programming the Arduino. One change was made to the circuit during the coding process, specifically that it was observed that the pins for the RGB LED had been defined incorrectly in the provided code.

Consequently, the leads from pins 9 and 11 were reversed in where they made contact with the LED legs to resolve the issue.





2) In the resistor ladder provided, the buttons allow for a closed circuit to be formed at different "levels" of the ladder, passing the current through a variable number of resistors. For instance, pushing the button closest to wire leading to the Arduino circumvents all 10KΩ resistors of the ladder, while pushing the next button over passes the current through one resistor, then the next through two resistors, and so on. When pushing two buttons, the one passing the current through the fewest resistors prevails as this effectively creates a parallel circuit where one branch of the parallel loop contains no resistors. Since the rule for calculating the total resistance of two resistors in parallel is:

$$\frac{1}{R_{total}} = \frac{1}{R_1} + \frac{1}{R_2} \tag{1}$$

Or, written another way:

$$\frac{R_1 R_2}{R_1 + R_2} = R_{total} \tag{2}$$

Then if one of the branches has effectively zero resistance, say R_2, then the entire left side of the equation goes to zero, and thus the effective resistance of two buttons in our ladder being pushed is only as resistive as those resistors that come after the parallel branching (i.e. after the button closest to the wire running to the Arduino).

Bearing this all in mind, since the wire running to the analog pin of the Arduino has a pull-down resistor, voltage at the pin is clamped to zero until a a circuit is closed to 5V via pushing one of the buttons. The voltage supplied to the pin then becomes proportional to how many resistors the current must pass through, in accordance with Ohm's law and the discussion above. The voltage on the analog pin is then converted to a digital signal with integer value between 0 and 1023, which within the code is passed to the tone() function. The tone() function interprets the number as a frequency and sends a pulsed digital signal from pin 3 with a duty cycle corresponding to the given frequency. This signal then reaches the buzzer, where the width of the pulses causes the diaphragm of the buzzer to vibrate at the specified frequency, creating an audible tone.

The mode button behaves quite simply as a digital pin with a pull-down resistor clamping it to zero. When the mode button is pushed, a circuit is closed to 5V and the digital pin reads HIGH, allowing us to detect a signal.