Bio-Sonification Arduino Shield

Electricity for Progress 2017
updated by Nanotopia 2020

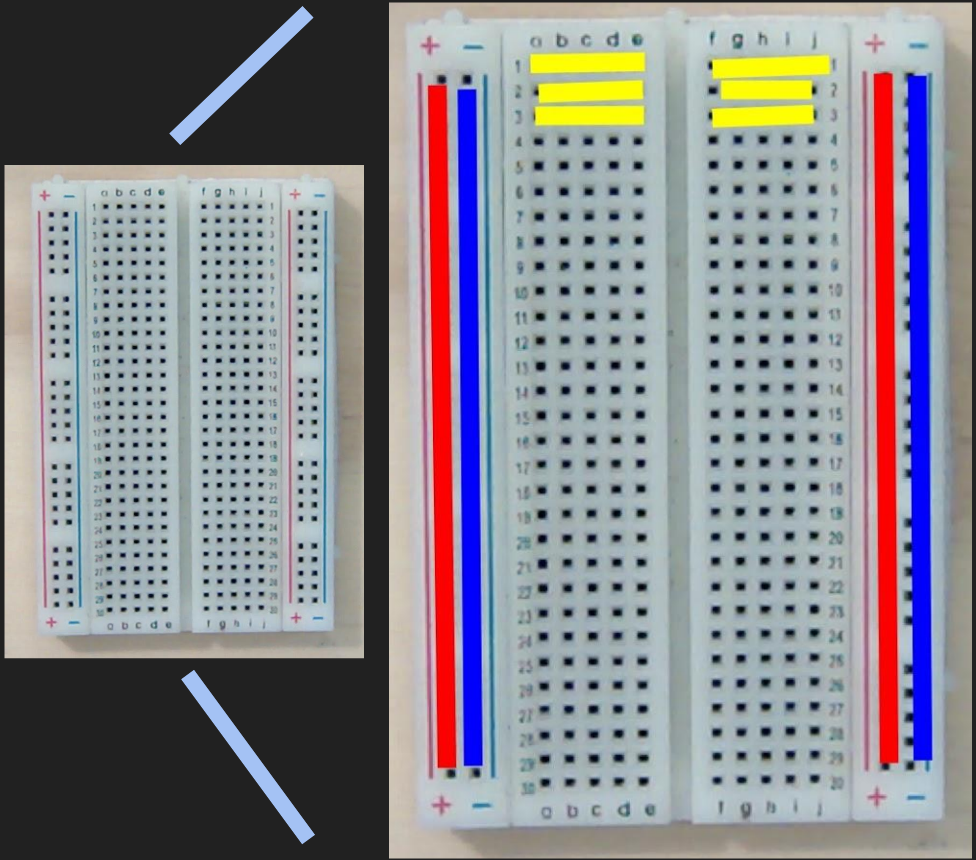
Translating changes in conductivity across two probes using MIDI data (Musical Instrument Digital Interface) for the generation of sound

Organize Kit Parts and Tools

1. Arduino - Uno compatible w/ USB cable and the Biodata Sonification Code
2. Computer with Arduino IDE, Sound Synthesis program (Ableton Live, etc), SerialMIDI (arduino to MIDI bus), MIDI Yoke (windows only MIDI bus)
3. Wire cutters (optional)
4. Electrode Leads 3.5mm jack to dual snap, Electrode Pad pair (snap)
5. Solderless Breadboard
6. Jumper Wires (22) - 6 black (1 long), 6 red, 2 orange (1 long), 2 brown, 2 blue, 2 white, 1 yellow, 1 green
7. LEDs - 2 red, 1 yellow, 1 green, 1 blue, 1 white
8. 555 Timer IC, Potentiometer, Button, 3.5mm Jack, 5 pin MIDI Jack
9. 220 ohm resistor x 7, 100 ohm resistor x 1, 0.1uf capacitor x 1, 4700 pF capacitor x 1, 47 uF capacitor x 1

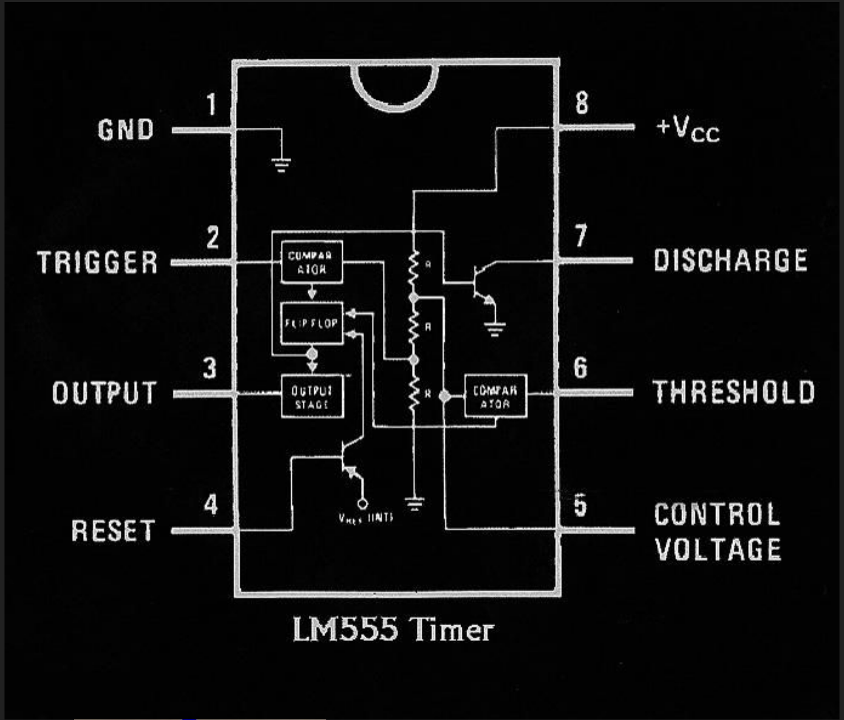
Understanding the Solderless Breadboard

Breadboards contain a series of holes in which the leads of electronic components can be easily placed. Rows are labeled with Numbers and each of the holes in a row is electrically connected (yellow lines). Columns are labeled with Letters and are isolated. Two pairs of 'Bus' are on each side of the board for positive Voltage (VCC) and Ground, the holes in the bus lines are connected vertically (red and blue lines)

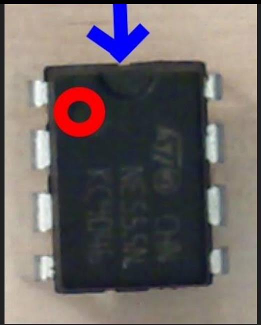


Understanding the 555 Timer

Integrated microchips wrap multiple electronic components into a single package. This chip has 8 pins organized in two rows of 4, called a Dual Inline Package (DIP). On the 555 Timer IC, each pin has a different function, see the diagram.



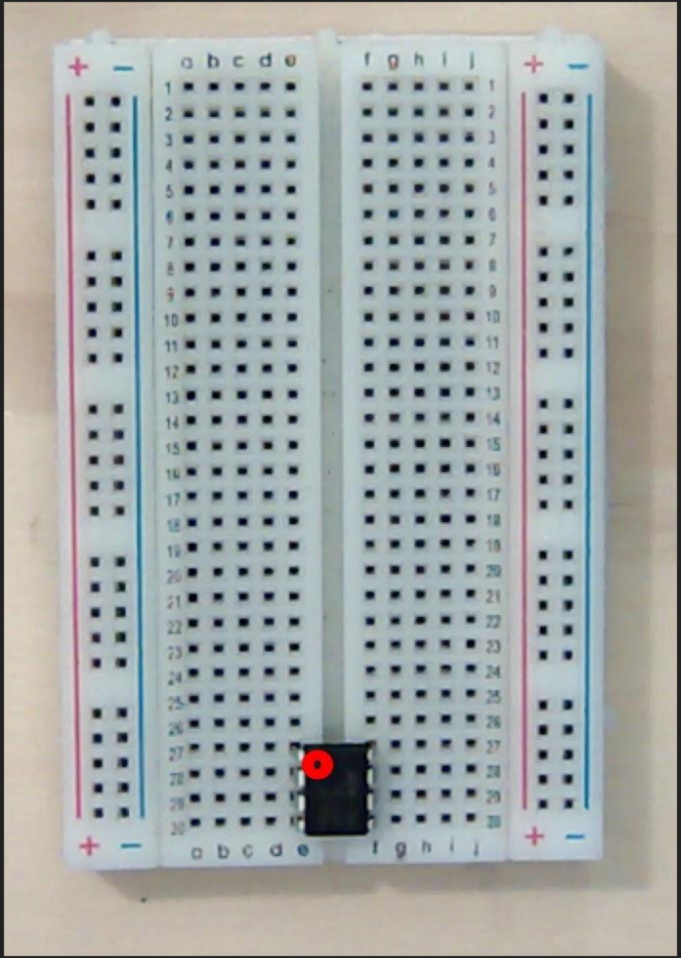
Identify the correct ‘orientation’ of the chip by finding the half-circle cut out at one of the ends this marks the ‘top’ of the chip. You will also see a small circle at the top left corner, marking Pin 1 of the IC.



Insert 555 Timer

With Pin 1 of the 555 identified, insert the chip into the breadboard at the bottom, spanning the central divide of the breadboard in rows 27-30 columns E and F, with pin 1 located at row 27 column E.

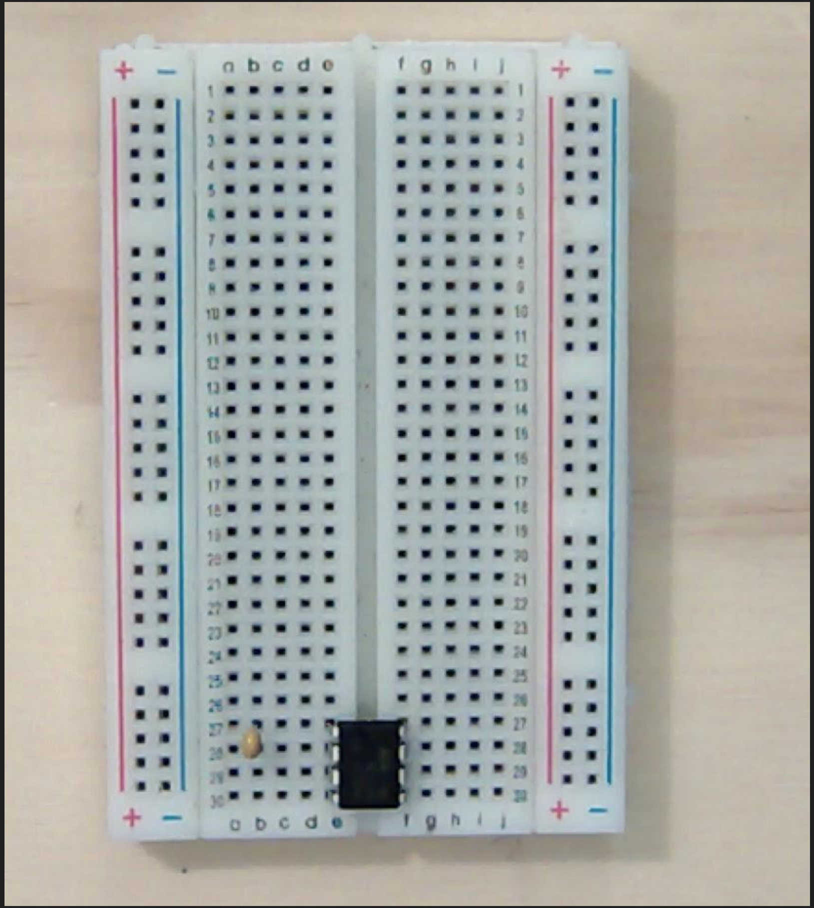
Placing the microchip on the breadboard in this way, allows us to use jumper wires and make connections with each of the pins through the breadboard holes across each row.



4700pF Capacitor

Insert the 4700pF capacitor, labeled 472, into row 27 and 28 in column b, connecting the Pin 2 (Trigger) and Pin 1 (Ground) of the 555 Timer

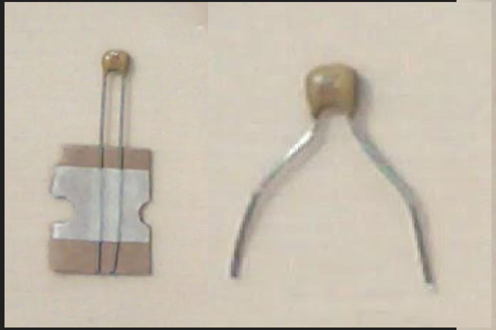
The timing capacitor is a very tiny component with a very small value, which allows the 555 Timer to run at a fast rate in astable mode giving the galvanic conductance meter a high resolution.

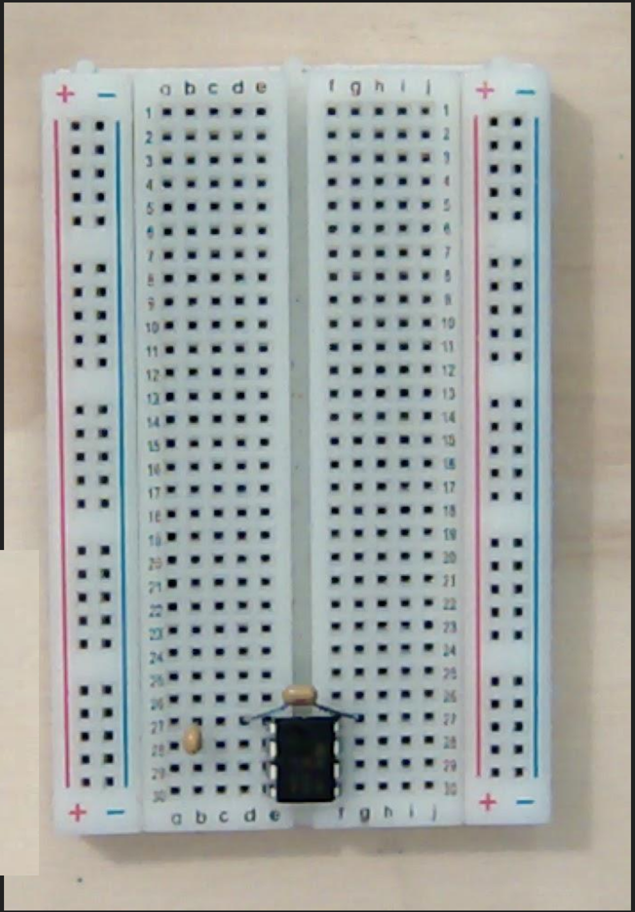


0.1uF Capacitor

Insert the 0.1uF capacitor, labeled 104, into row 27 columns d and g, spanning across the 555 Timer and the two sides of the breadboard connecting Pin 1 (Ground) and Pin 8 (VCC). This capacitor is performing ‘high frequency’ decoupling, which cleans up the noise in the voltage lines.

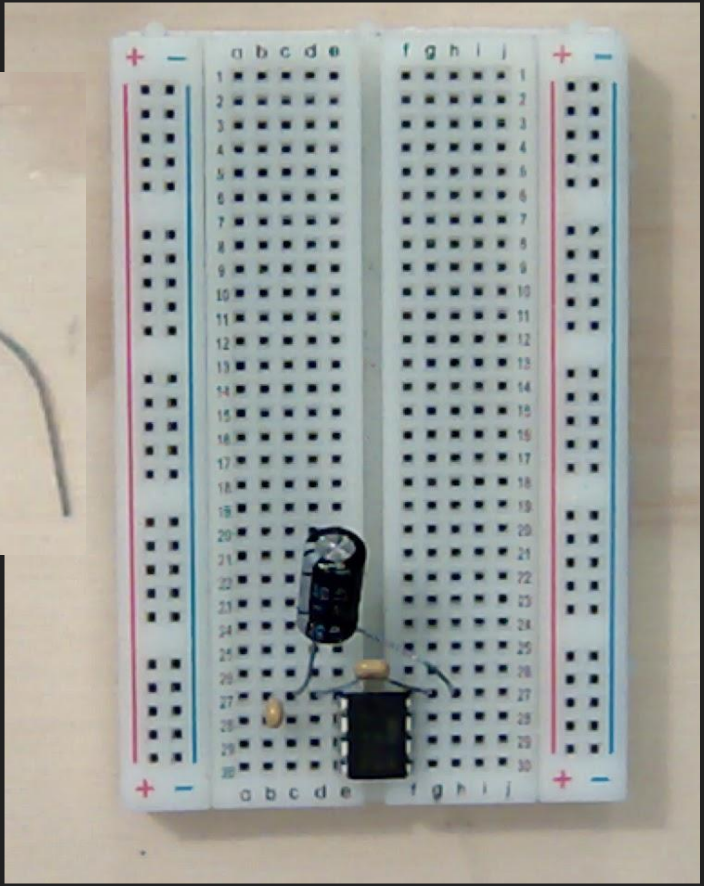
Remove the paper tape and trim down the capacitor leads (optional). Bend the leads outward to span the 555 Timer and insert into the breadboard.





47uF Electrolytic Capacitor

Insert the 47uF electrolytic capacitor, labeled 47uF, into row 27 columns c and h spanning across the 555 Timer and the two sides of the breadboard, connecting Pin 1 (Ground) and Pin 8 (VCC). This capacitor is performing ‘low frequency’ decoupling, cleans out ripples and sags in the voltage line for the 555 Timer.



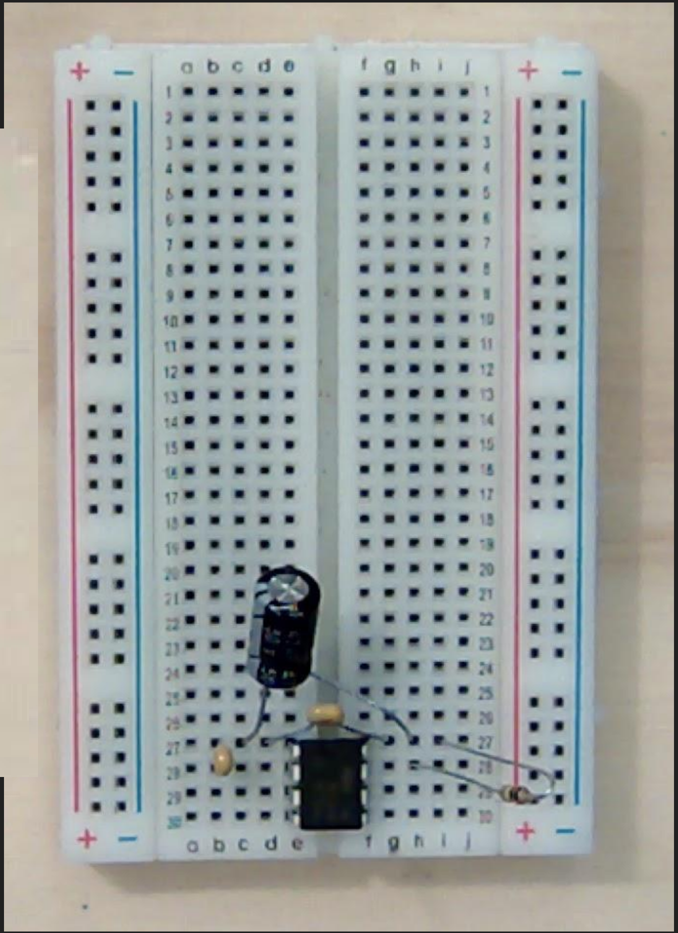
The electrolytic capacitor is polarized, and must be connected in the correct orientation. The longer lead of the capacitor is positive, while the negative lead is shorter and also is labeled (-) on the body of the cap. Attach the negative lead to row 27 column c, Pin 1 (Ground) of the 555 Timer, and the positive lead to row 27 column h, Pin 8 (VCC) of the 555 Timer.

100k Resistor

Insert the 100k Resistor (labeled brown, black, yellow) into row 28 column h and row 27 column i

Bend one lead of the resistor all the way around to be parallel with the other lead, trim the leads with wire cutters (optional)

This connects Pin 7 (Discharge) to Pin 8 (VCC), and acts a a ‘pull up’ on the electrode which we will connect to Pin 7 in a later step

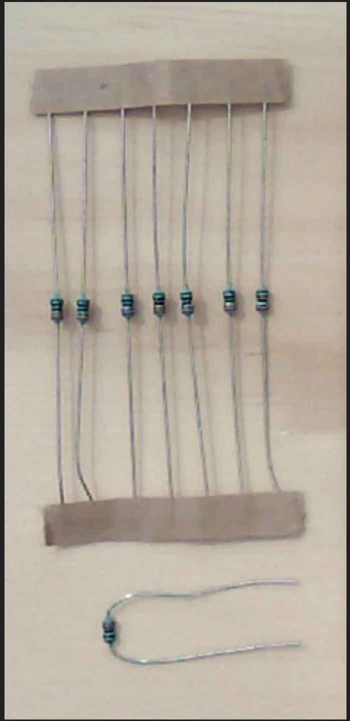


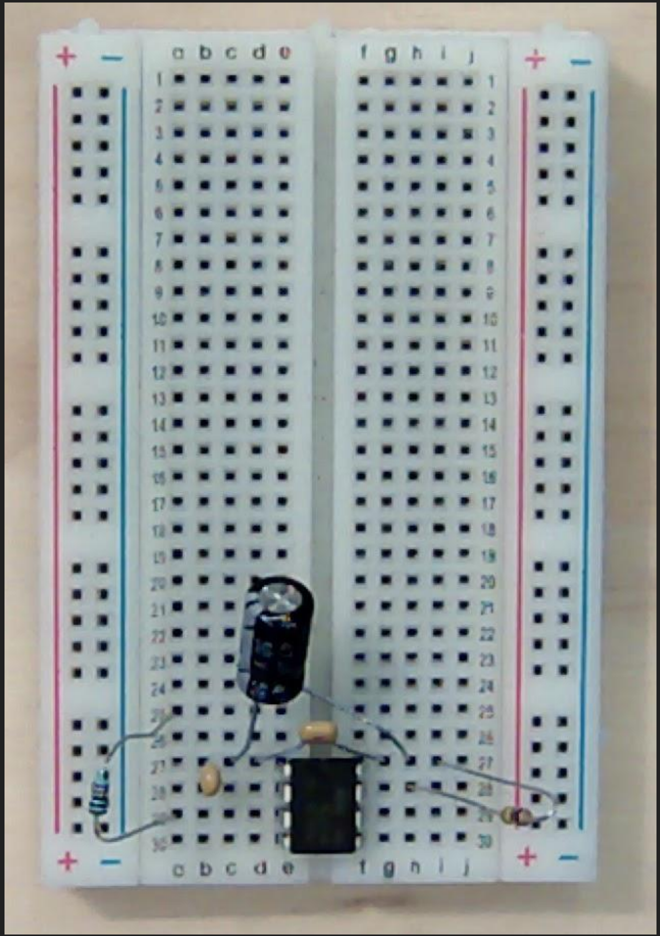
220 Ohm Resistor

Insert one of the 220 Ohm resistors (labeled red, red, black) into row 29 column a and row 25 column a

The resistor should be removed from the paper tape, and bent into a U shape, with each lead of the resistor pointing downward

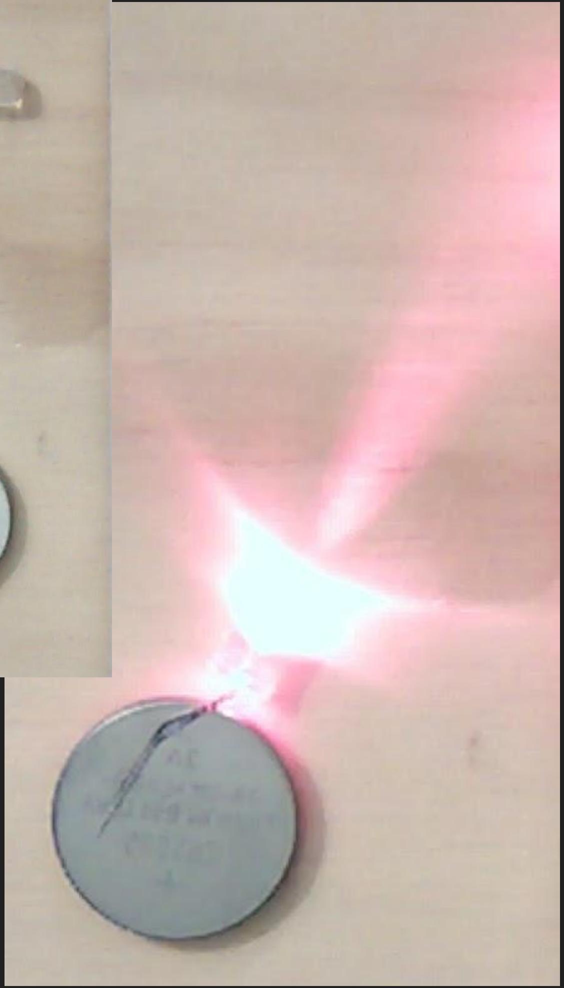
This resistor connects to Pin3 (Output) of the 555 and will limit the current driving the red output LED in the next step.





LEDs - Colour and Polarity

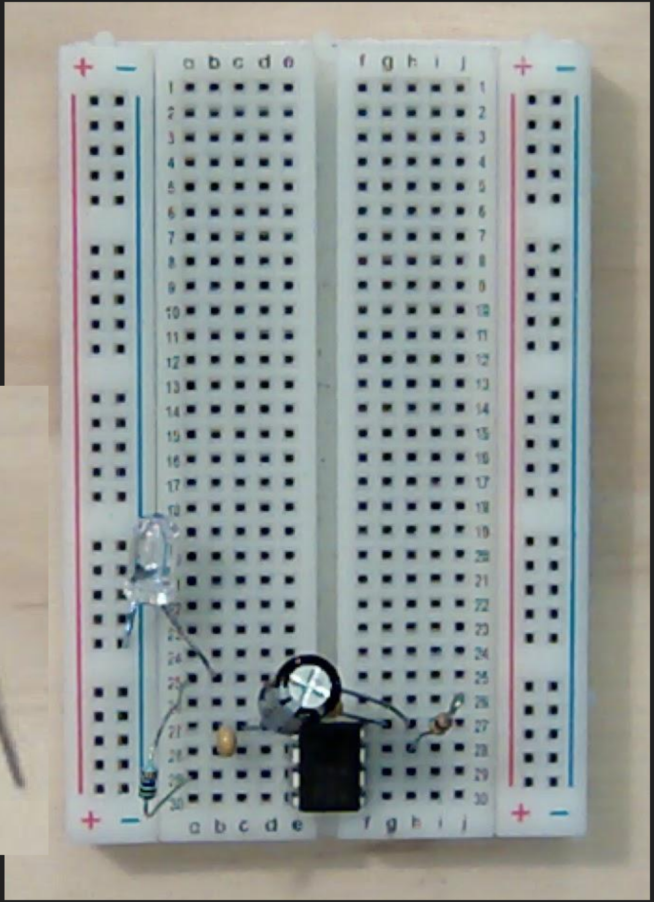
Identify the colour and polarity of your LED by testing it using the CR2032 3 volt battery. Place the battery between the leads of the LED, if it does not illuminate turn the battery around and test again, this will help determine the colour of the LED and will identify the positive cathode of the LED (usually a longer lead) on the positive (+) side of the battery and the negative anode of the LED (shorter lead) on the negative side of the battery.



LED - 555 Timer Output

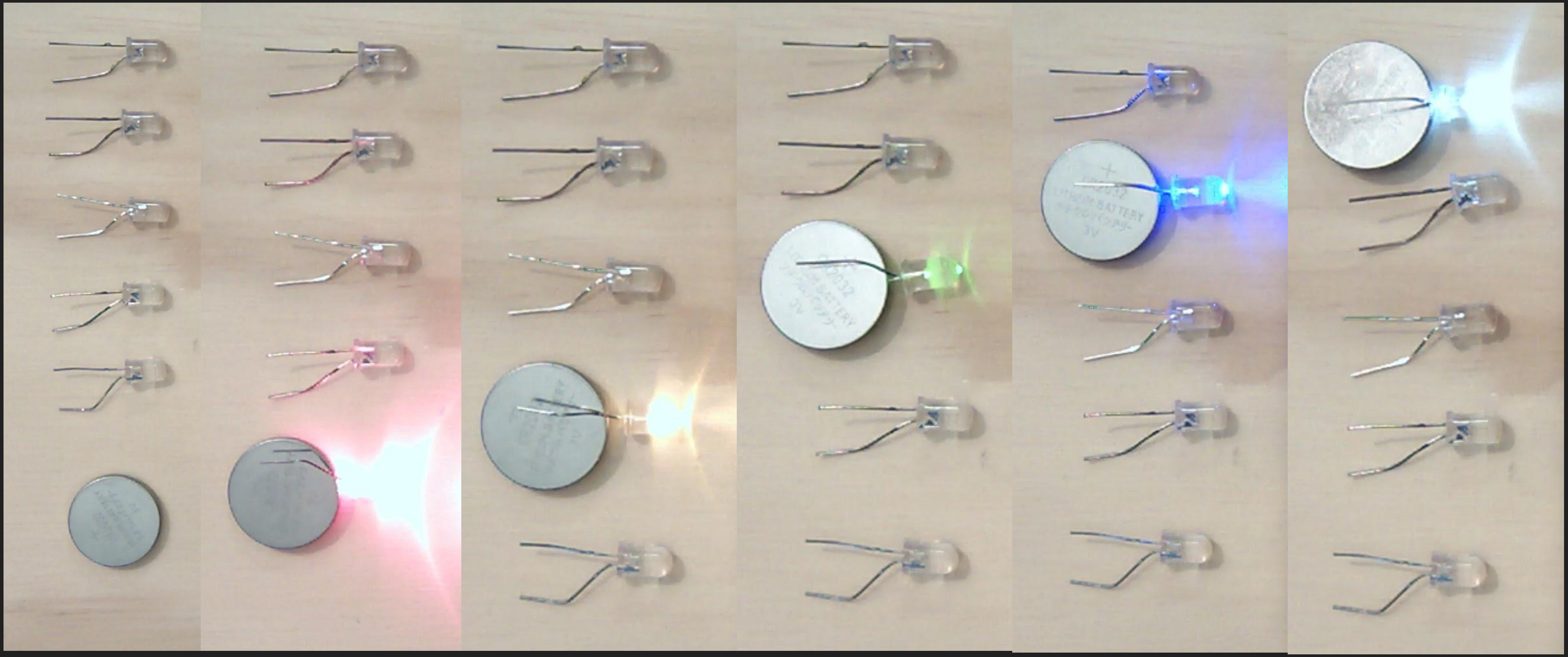
Insert the cathode (positive pin) of a red LED into row 25 column b and the anode (negative pin) of the LED into the Ground Bus (vertical blue line marked ‘-’) on the left side of the breadboard

Bend the cathode of the LED, usually the longer of the two leads, outward in order to reach between the breadboard holes. Bending an LED in this way helps accommodate for the different lead lengths and fits better into a breadboard



LEDs - Five Colour Light Show

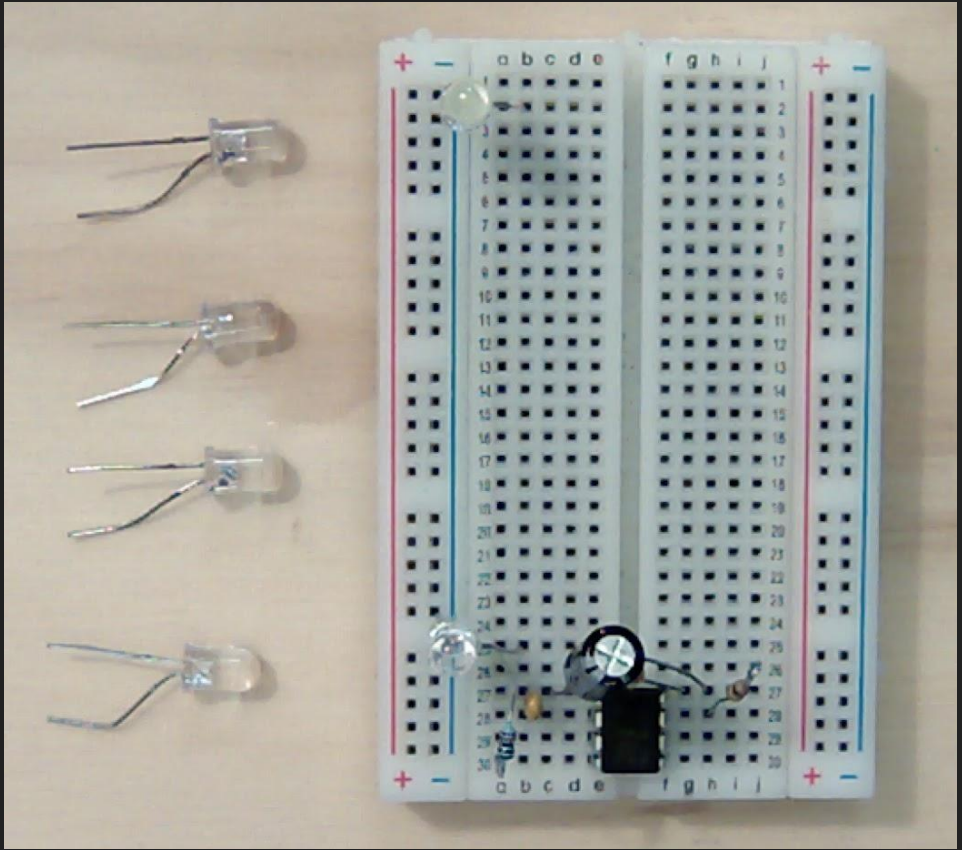
Use the CR2032 battery to identify each LED and bend the leads into shape



LEDs White - Insert into Breadboard

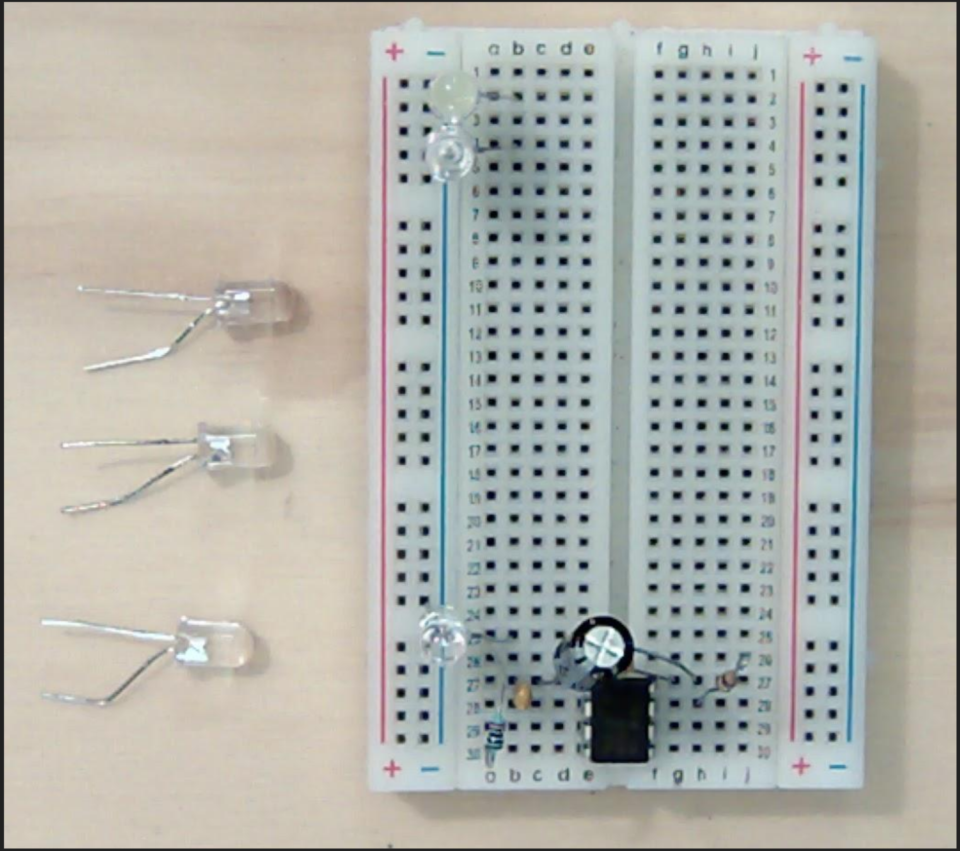
Insert the cathode (positive pin) of a white LED into row 2 column b, and the anode (negative pin) into the Ground Bus on the left side of the breadboard.

Each of the LEDs will be inserted in a similar manner, into rows 2, 4, 6, 8, and 10



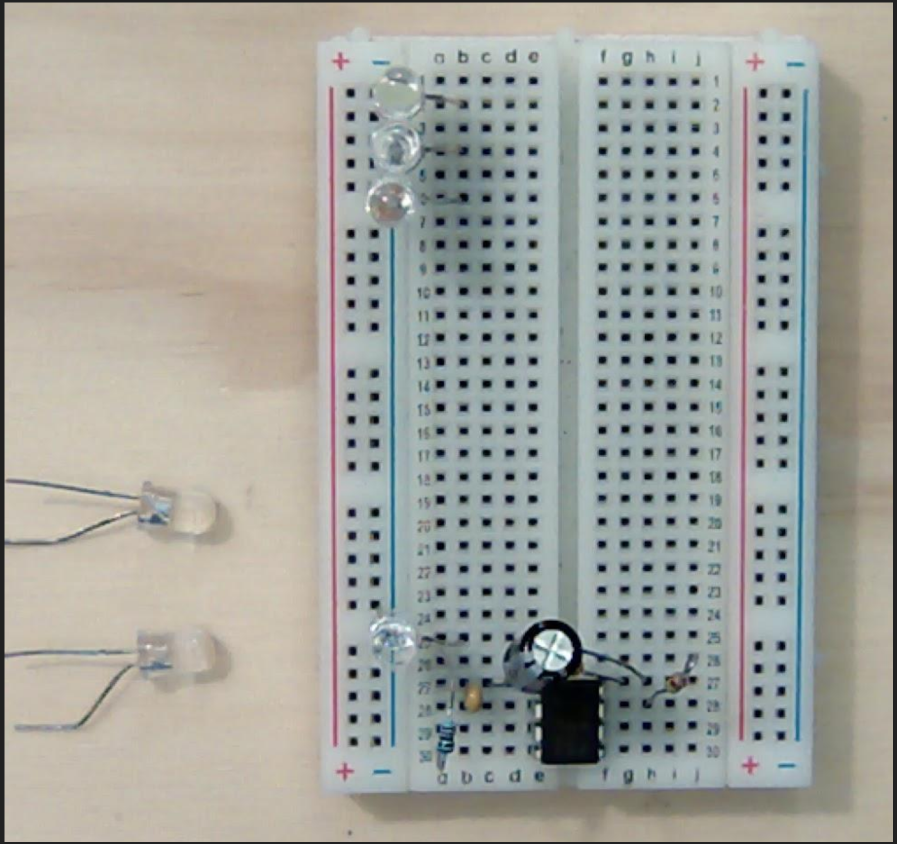
LEDs Blue - Insert into Breadboard

Insert the cathode (positive pin) of a blue LED into row 4 column b, and the anode (negative pin) into the Ground Bus on the left side of the breadboard.



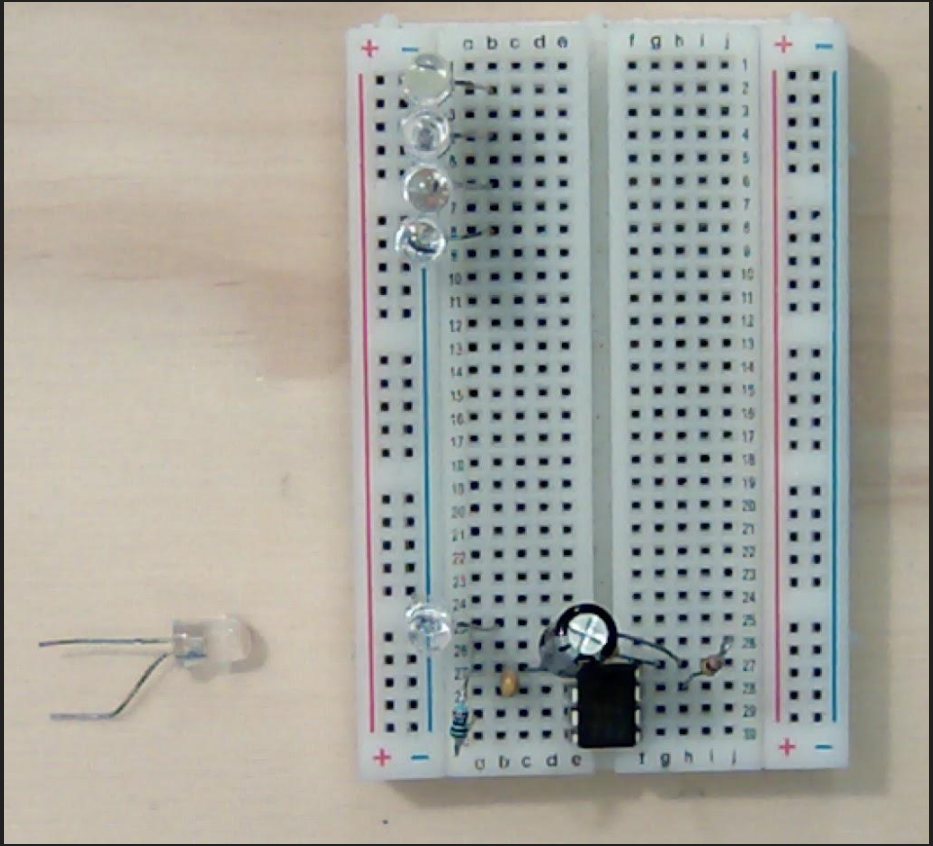
LEDs Green - Insert into Breadboard

Insert the cathode (positive pin) of a green LED into row 6 column b, and the anode (negative pin) into the Ground Bus on the left side of the breadboard.



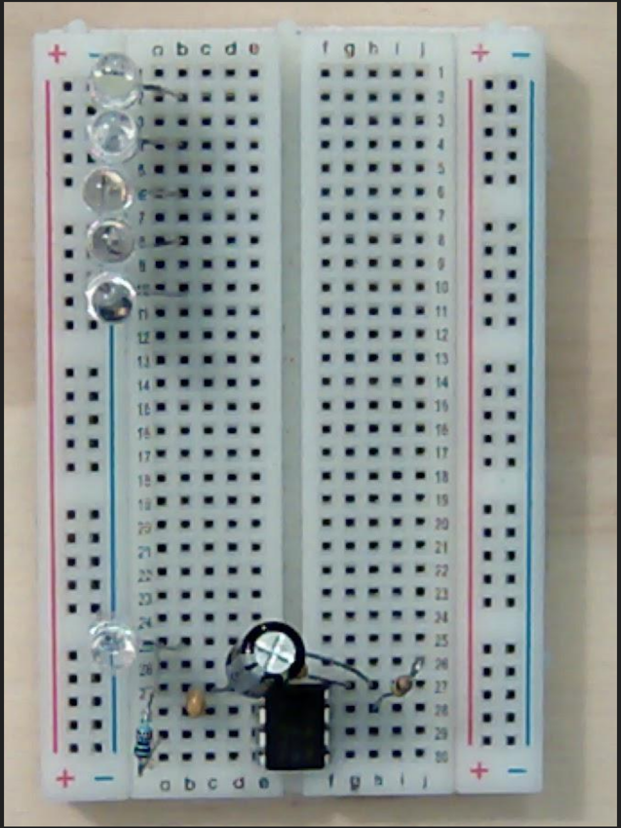
LEDs Yellow - Insert into Breadboard

Insert the cathode (positive pin) of a yellow LED into row 8 column b, and the anode (negative pin) into the Ground Bus on the left side of the breadboard.



LEDs Red - Insert into Breadboard

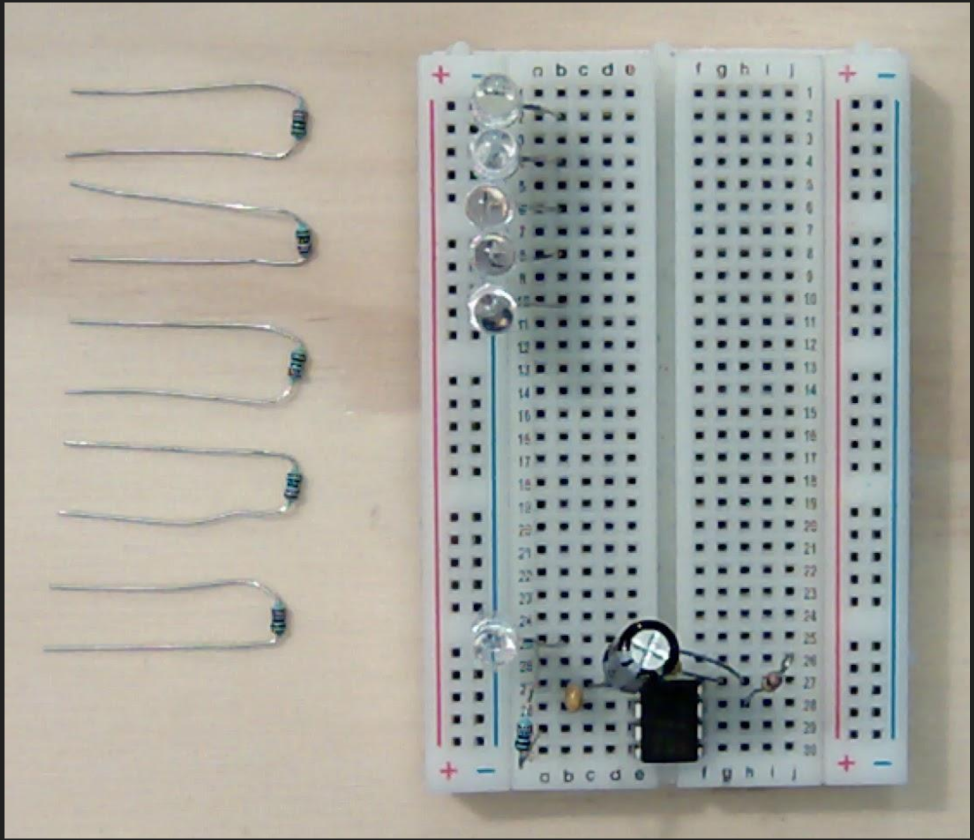
Insert the cathode (positive pin) of a red LED into row 10 column b, and the anode (negative pin) into the Ground Bus on the left side of the breadboard.



220 Ohm Resistor - LED current limiting

Bend five of the 220 ohm resistors (labeled red, red, black) into a ‘U’ shape. These resistors will limit the current running through each of the coloured LEDs and prevents burnout.

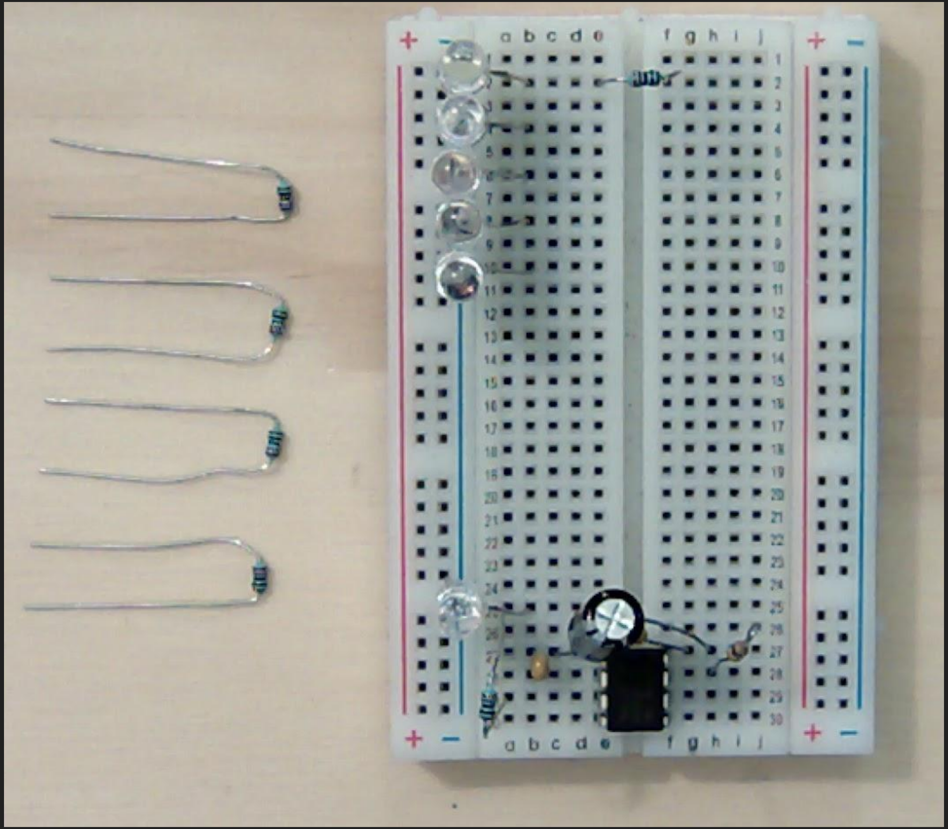
Each resistor will attach to one of the LEDs in rows 2,4,6,8, and 10. The resistors will jump over the divide in the middle of the breadboard, where they will be connected to the Arduino using jumper wires in a later step.



220 Ohm Resistor - White LED current limiting

Insert a 220 Ohm resistor into row 2 column e and row 2 column f

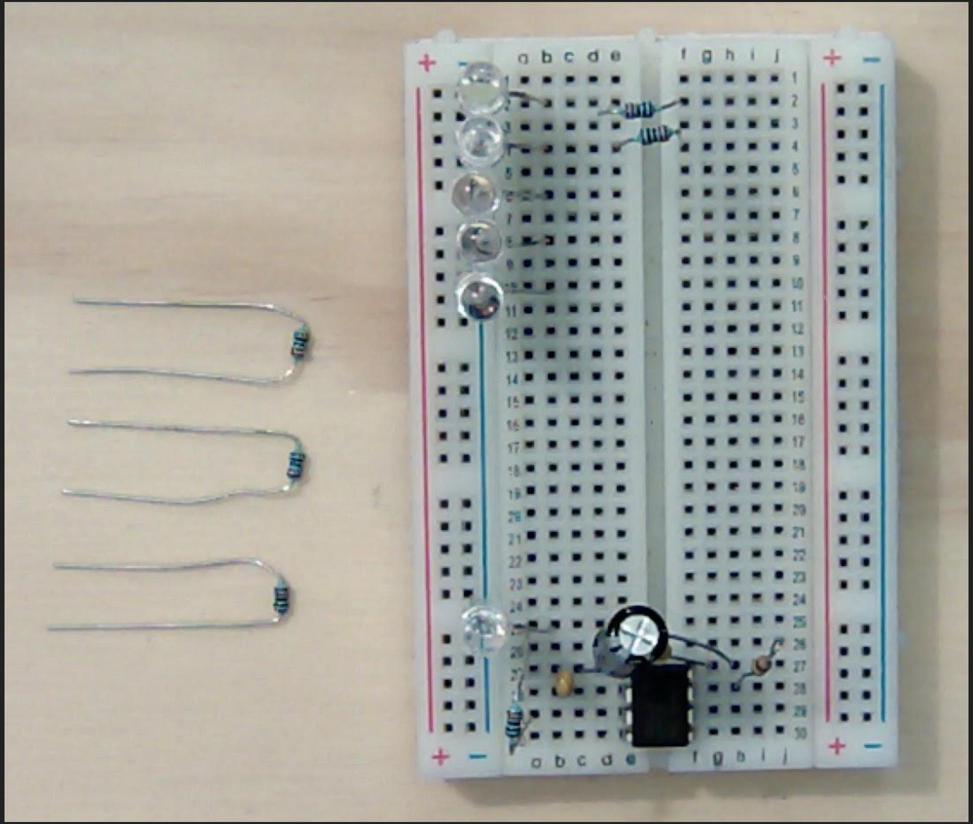
The resistor will span across the divide down the center of the breadboard.



220 Ohm Resistor - Blue LED current limiting

Insert a 220 Ohm resistor into row 4 column e and row 4 column f

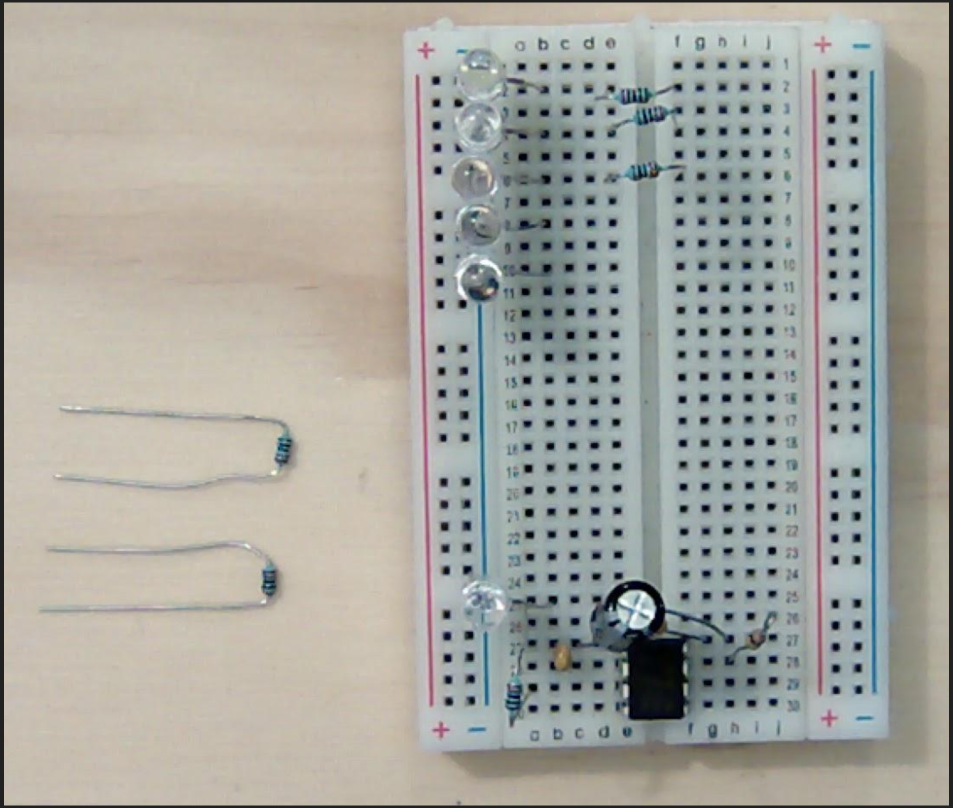
The resistor will span across the divide down the center of the breadboard.



220 Ohm Resistor - Green LED current limiting

Insert a 220 Ohm resistor into row 6 column e and row 6 column f

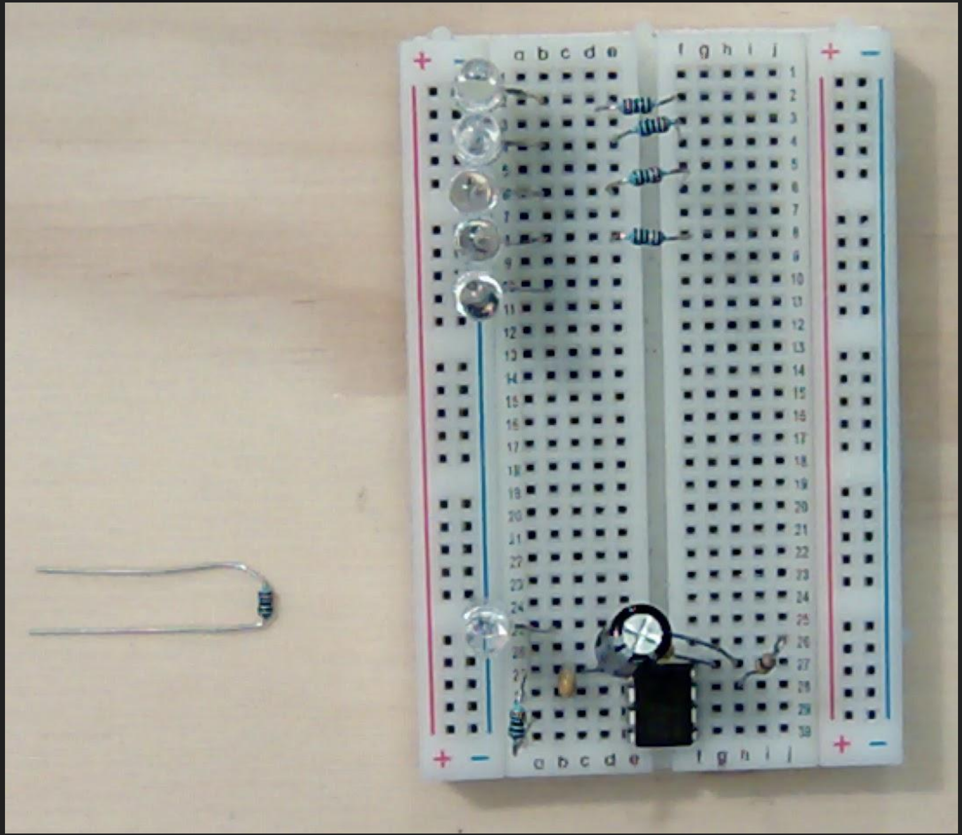
The resistor will span across the divide down the center of the breadboard.



220 Ohm Resistor - Yellow LED current limiting

Insert a 220 Ohm resistor into row 8 column e and row 8 column f

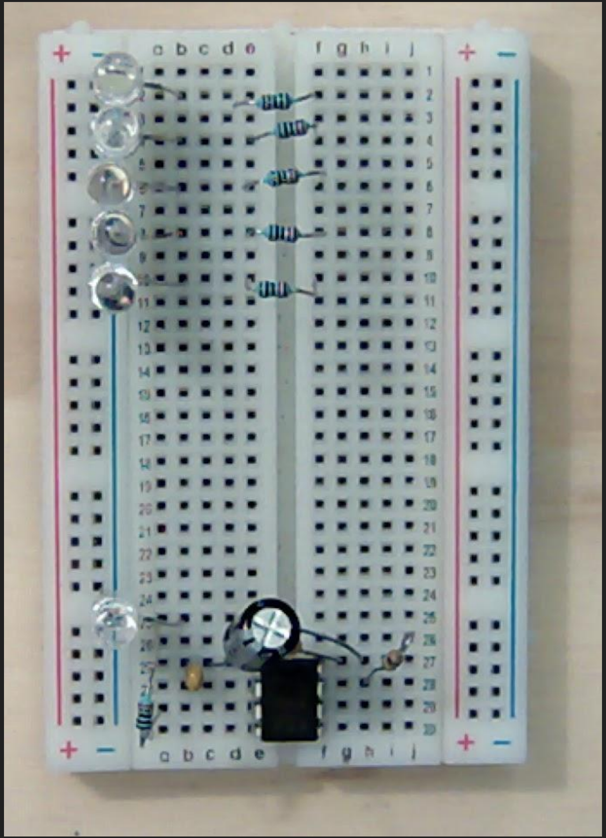
The resistor will span across the divide down the center of the breadboard.



220 Ohm Resistor - Red LED current limiting

Insert a 220 Ohm resistor into row 10 column e and row 10 column f

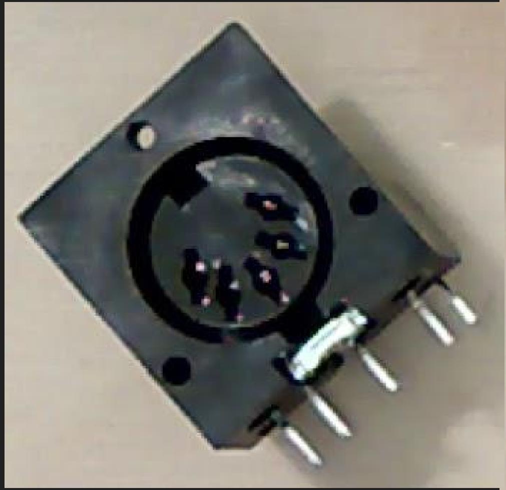
The resistor will span across the divide down the center of the breadboard.

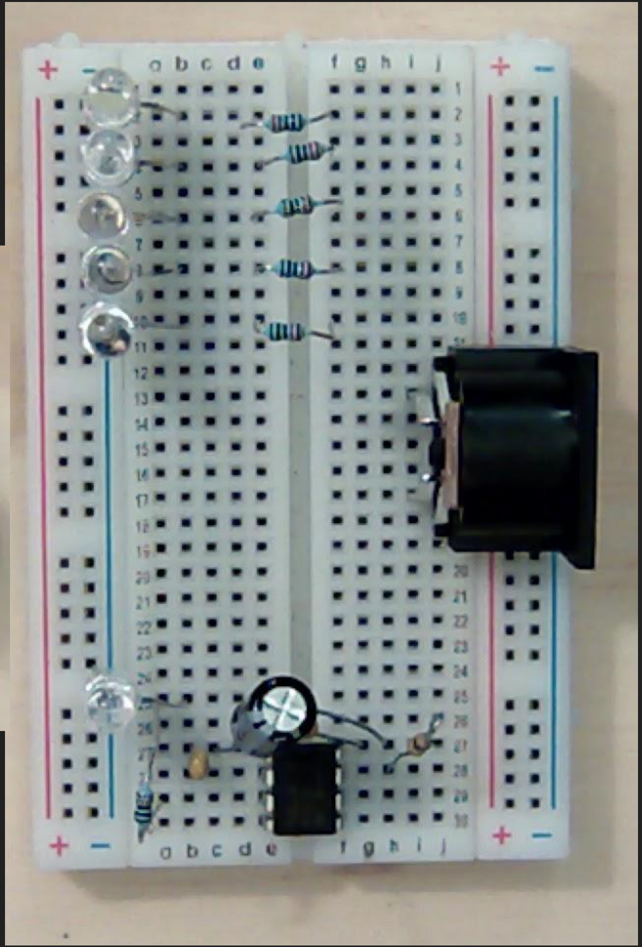


MIDI 5 Pin DIN Connector

Insert the 5 Pin MIDI DIN connector into the right side of the breadboard, the socket pointing towards the right, and with the ‘top’ pin in row 12 column j and the ‘bottom’ pin in row 18 column j.

Each pin should easily fit into the breadboard. Bend up the pointy ‘stress relief’ pins at the front of the connector.

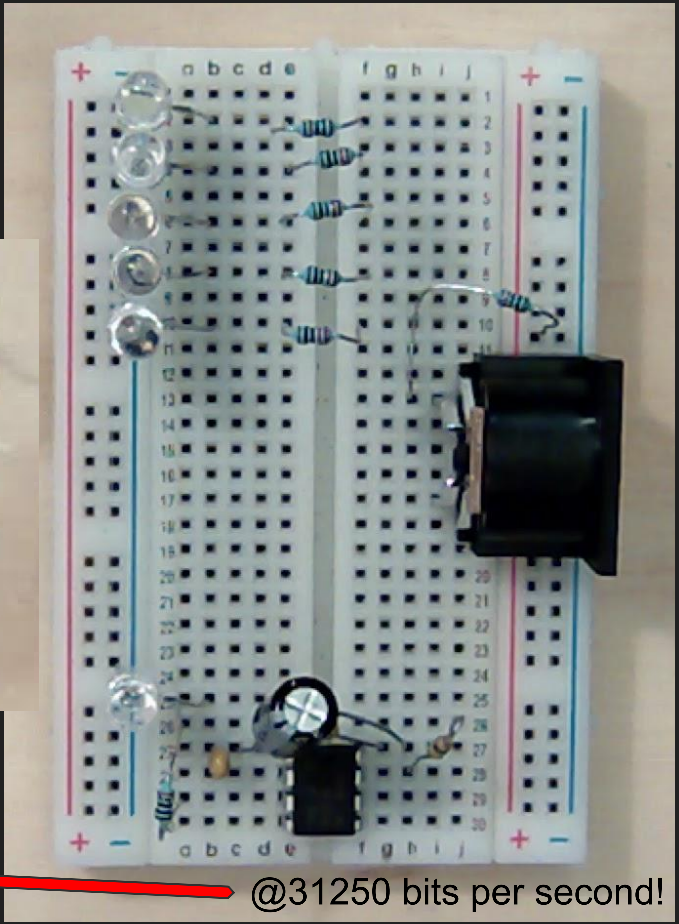




220 Ohm Resistor - MIDI Pull Up

Insert a 220 Ohm resistor into row 13 column h and the positive Bus on the right hand side of the breadboard (marked with a red line and +). Bend the resistor into shape

This pull up resistor powers the transmission signal of the MIDI serial data. MIDI data is ‘optically isolated’, and just like an LED this is the current limiting resistor for the MIDI data output. MIDI data flashes tiny leds inside a synth!

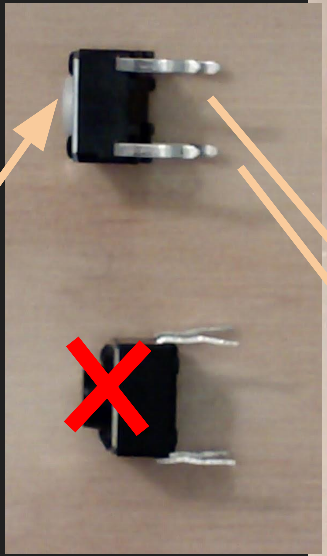


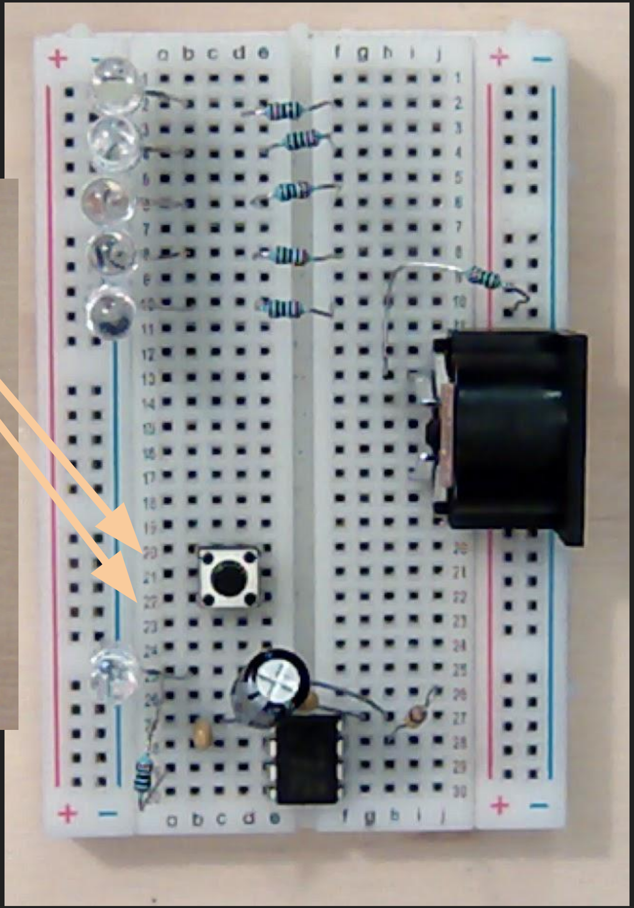
@31250 bits per second!

Tactile Button

The Button as four leads, two of which are normally connected, and two others which are normally open and connect when the button is pressed.

Position the button as in the top image, such that the metal pins are on the 'same face' and connect to row 20 and row 22 columns b/e, the button should stay on the left side of the breadboard.



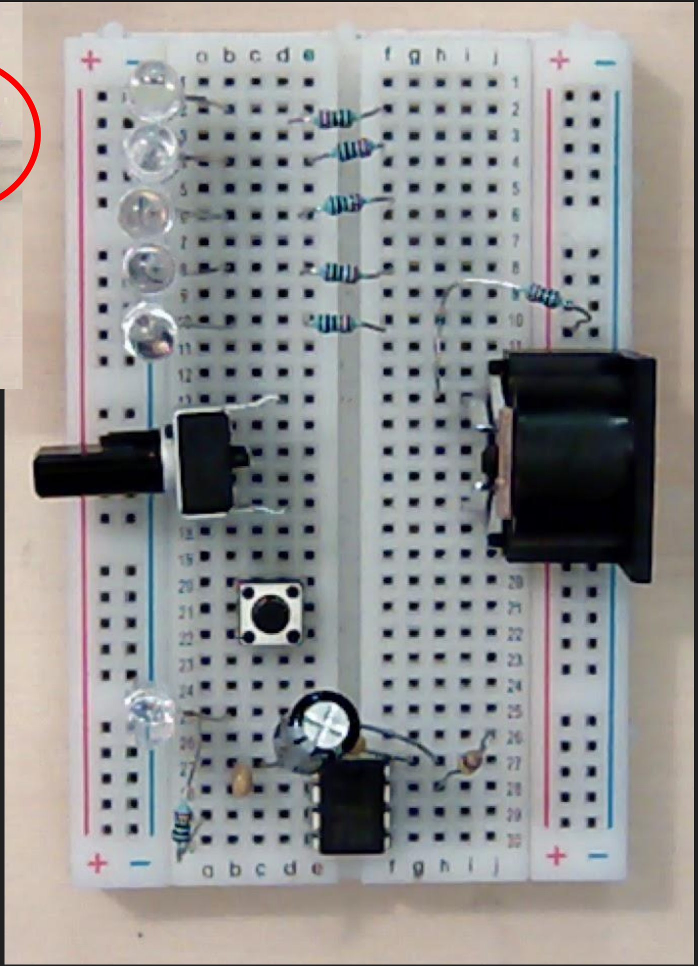


(optionally) Use a multimeter to check resistance/conductivity to determine the pins on the button when pressed.

Potentiometer - 10k

Hold the knob of the potentiometer and carefully bend the pins so they are facing outward as shown.

Insert the 10k Potentiometer into rows 14, 15, and 16 column a, facing outward to the left.

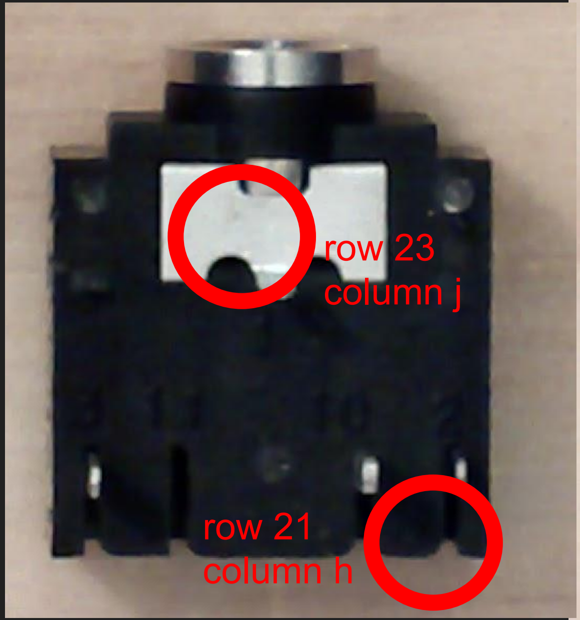


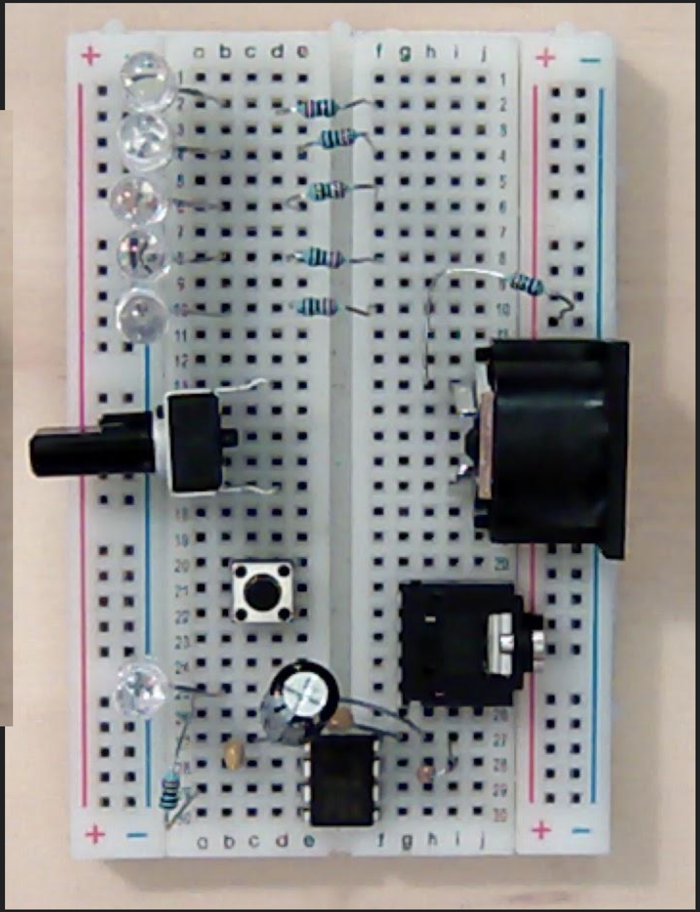
The potentiometer can be seen as a ‘variable’ resistor, with the center pin acting as a ‘wiper’ across the full range resistance. Here we use it as a voltage divider, with one side pulled high to VCC, the other Ground, and the measuring the center pin on Arduino analog input A0.

3.5mm Jack - Electrode Input

Insert the 3.5mm jack such that the center pin is in row 23 column j and the rear pin is in row 21 column h.

The jack should pop easily into the board. Use row 20 as a guide, and keep column f open to connect jumper wires in a later step.

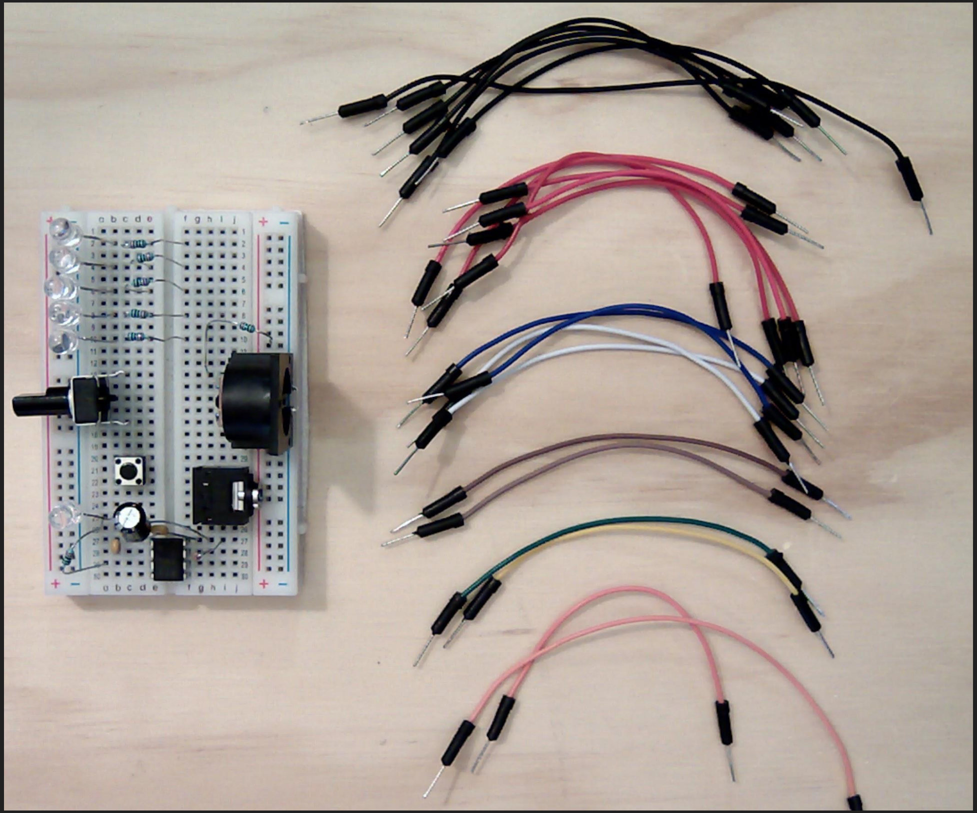




Jumper Wires - Electrical Connections over Distance

Jumper Wires allow connections between components on different rows of the breadboard. Here we will also use Jumper Wires to connect the solderless Breadboard to an Arduino Uno board.

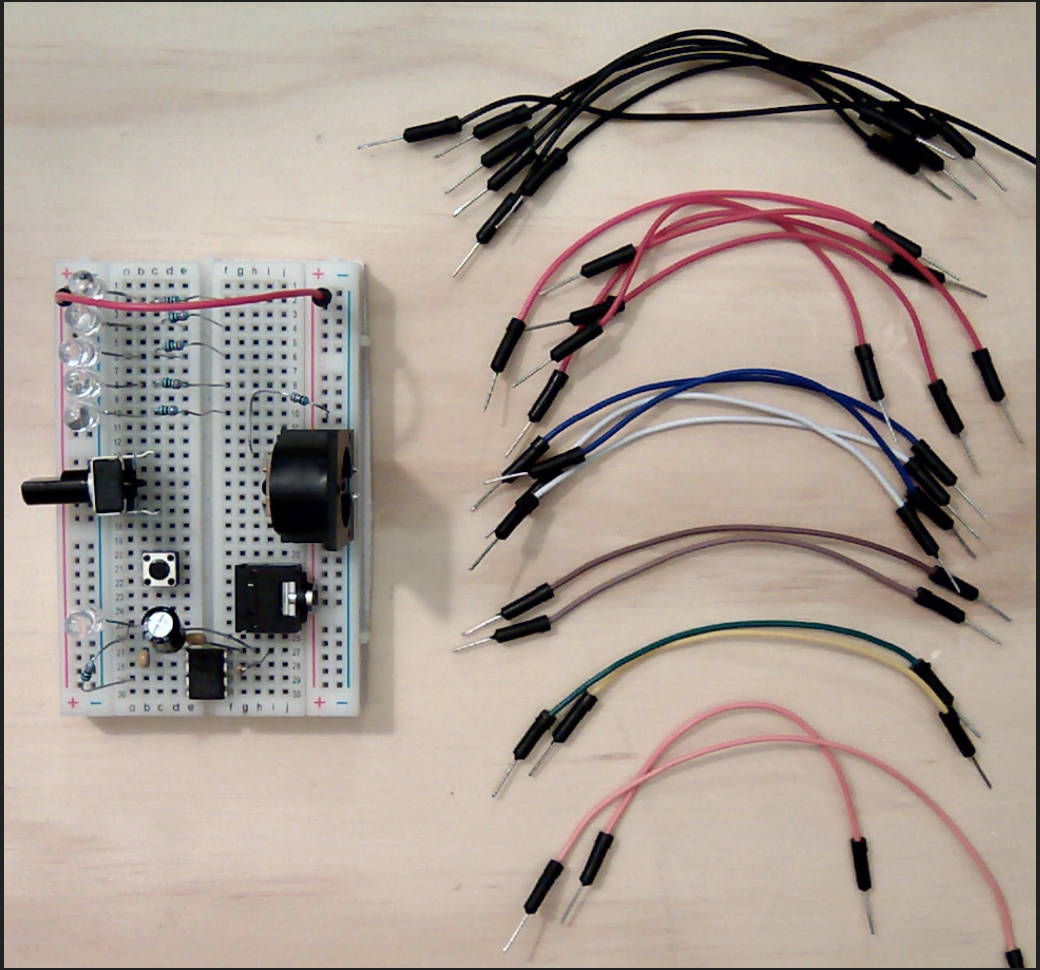
22 Jumper Wires are required for this project, colour doesn't matter but it is helpful to visually understand the circuit. We will use 6 black (1 long), 6 red, 2 orange (1 long), 2 brown, 2 blue, 2 white, 1 yellow, 1 green



VCC Bus Jumper

With a red Jumper Wire connect the left positive VCC Bus on the breadboard to the right VCC Bus, in the top row of each side. The positive Bus is identified by a vertical red line and a + symbol.

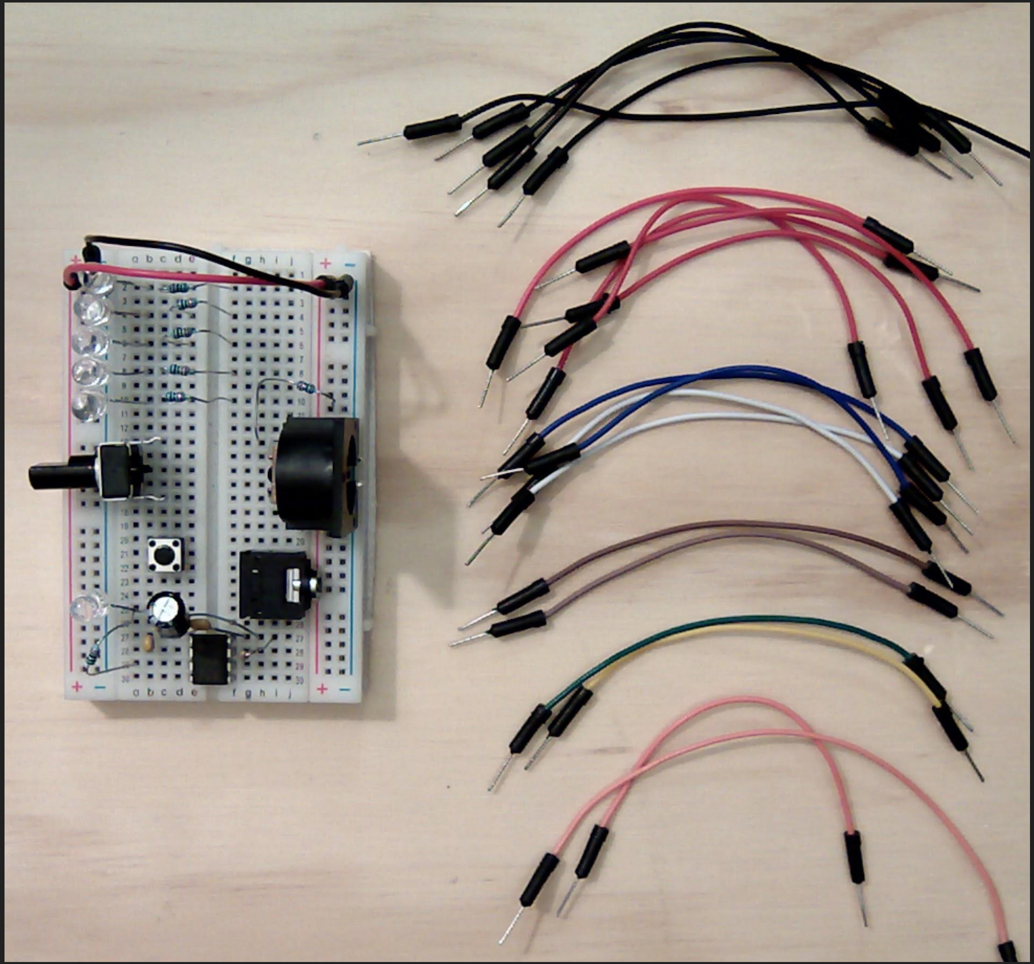
Each of the holes in the VCC column are electrically connected, allowing power to pass up and down both sides.



Ground Bus Jumper

With a black Jumper Wire connect the left negative Ground Bus on the breadboard to the right Ground Bus, in the top row of each side. The negative Bus is identified by a vertical blue line and a - symbol.

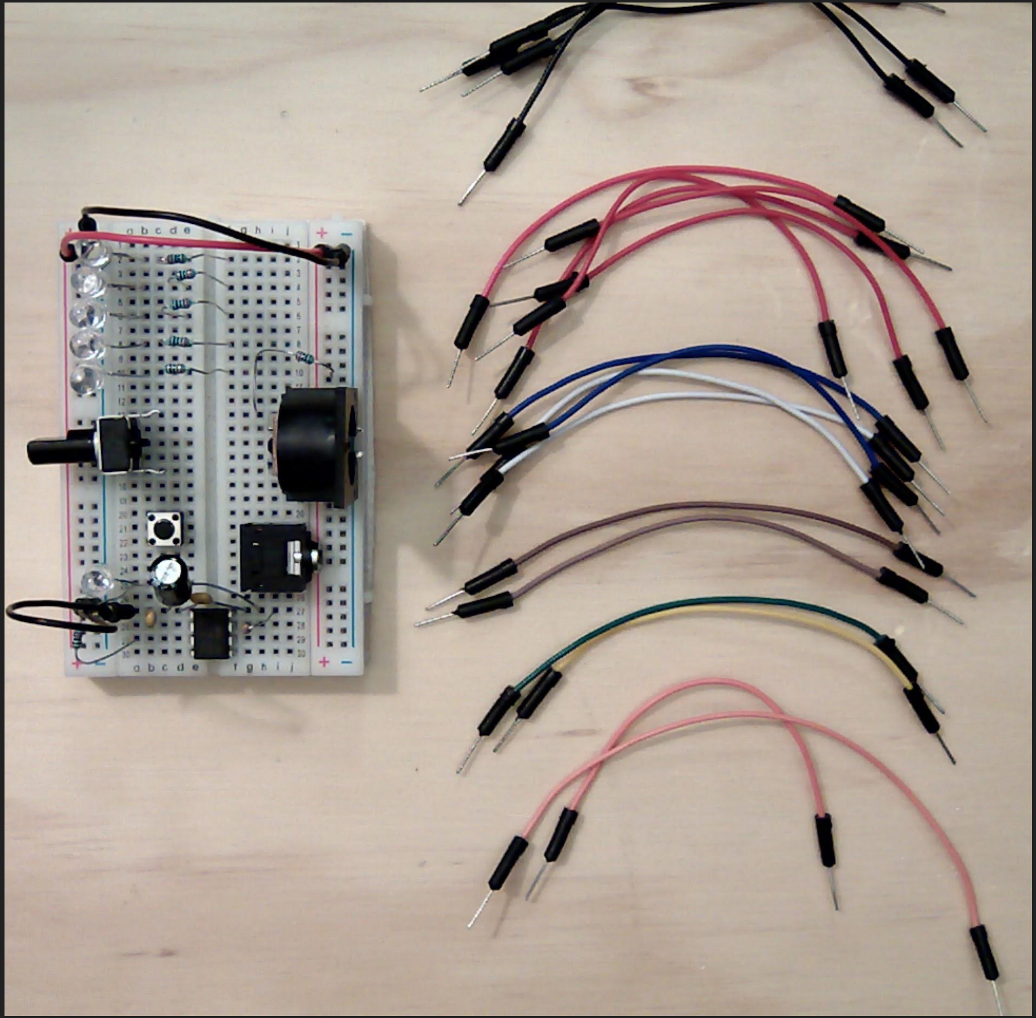
Each of the holes in the Ground column are electrically connected, allowing Ground access up and down both sides.



555 Ground Jumper

With a black Jumper Wire connect 555 pin 1 (Ground) row 27 column a to the left Ground Bus.

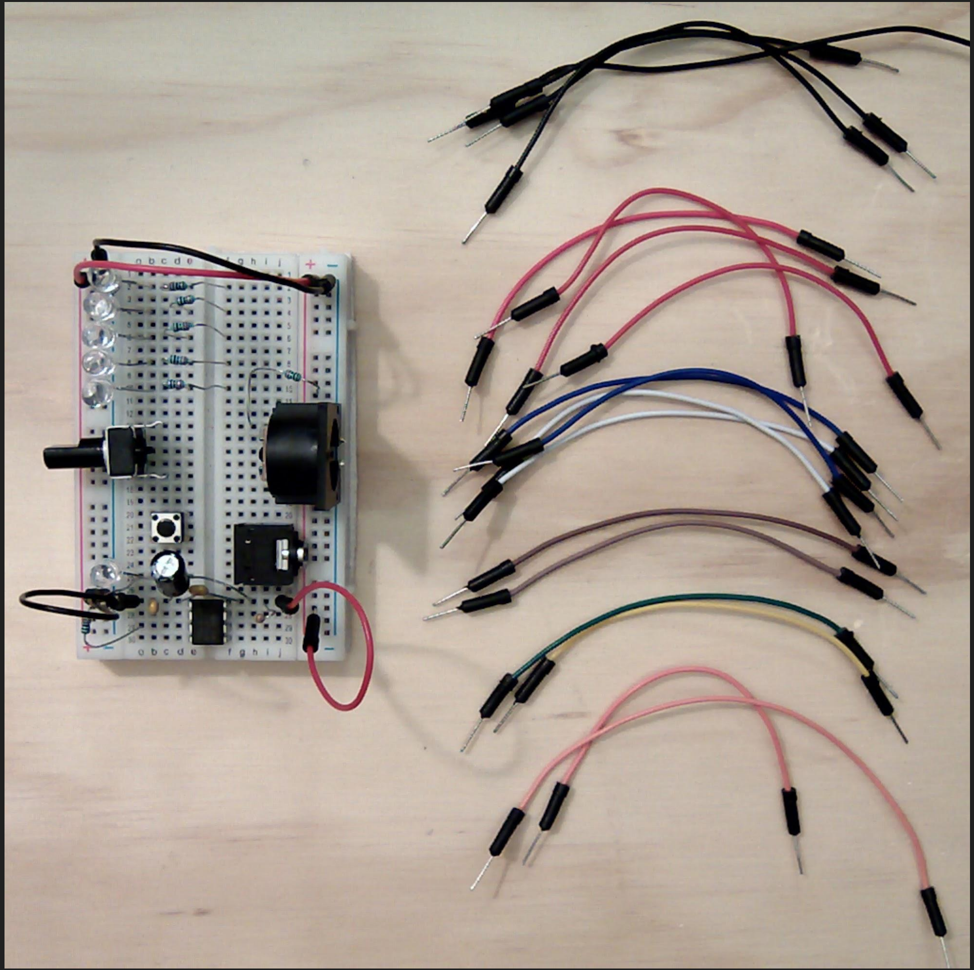
This will provide Ground connection for the 555 Timer IC.



555 VCC Jumper

With a red Jumper Wire connect 555 pin 8 (VCC) row 27 column j to the right positive VCC Bus.

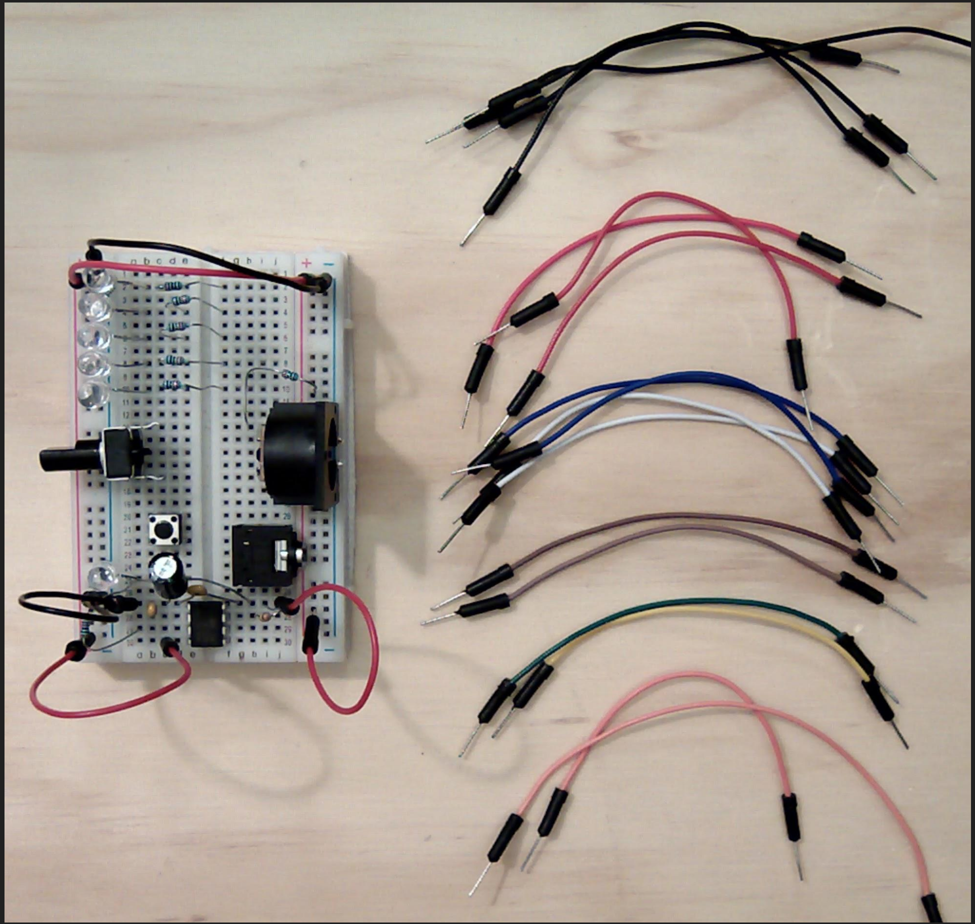
This will provide positive voltage connection for the 555 Timer IC.



555 Pin4 (Rest) to VCC

With a red Jumper Wire connect 555 pin 4 (Reset) row 30 column c to the left positive VCC Bus.

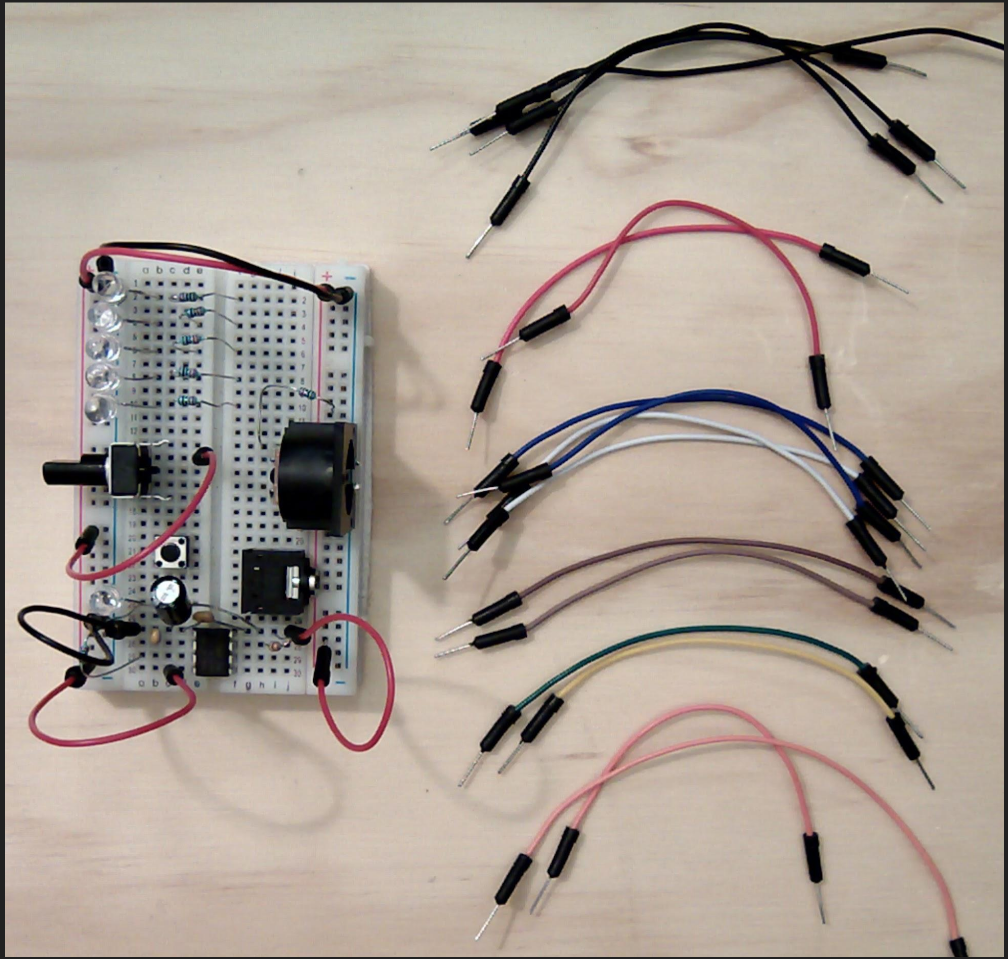
This will provide pull up voltage to the reset connection in the 555 Timer IC for 'astable' mode.



Potentiometer to VCC

With a red Jumper Wire connect Potentiometer ‘top’ Pin row 14 column e to the left positive VCC Bus.

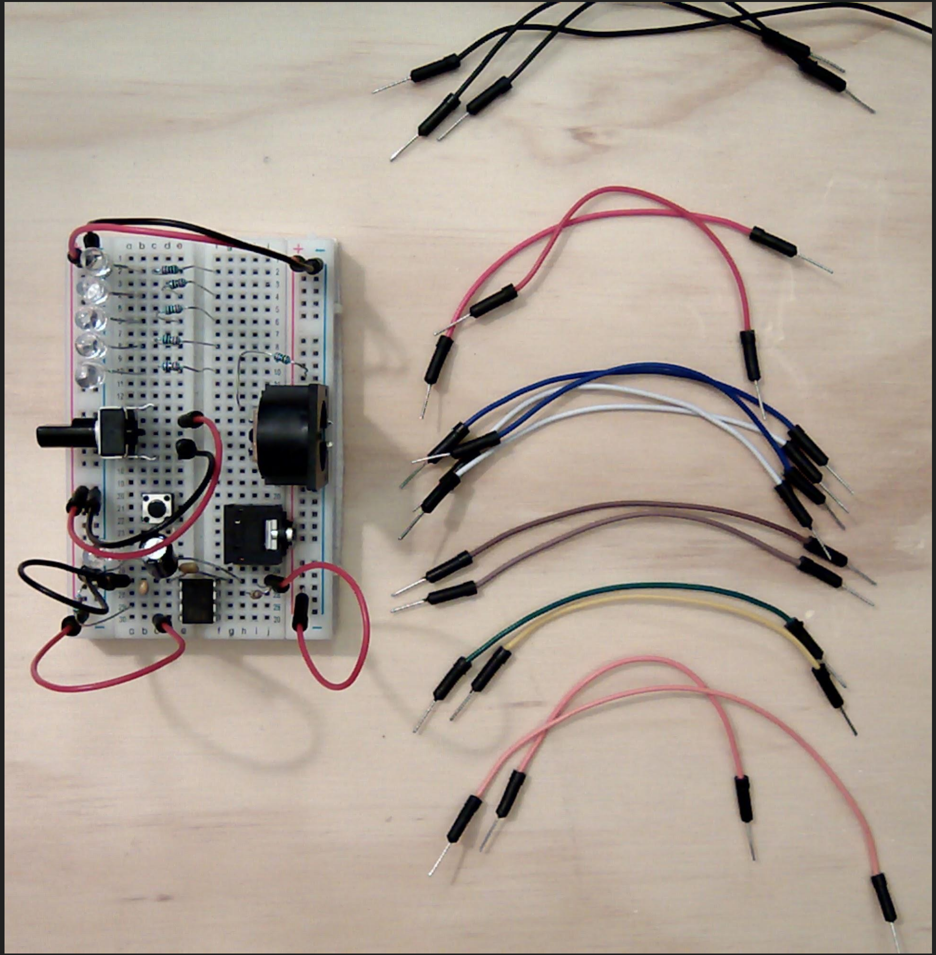
This will provide positive voltage connection for the potentiometer acting as a voltage divider.



Potentiometer to Ground

With a black Jumper Wire connect Potentiometer ‘bottom’ Pin row 16 column e to the left negative Ground Bus.

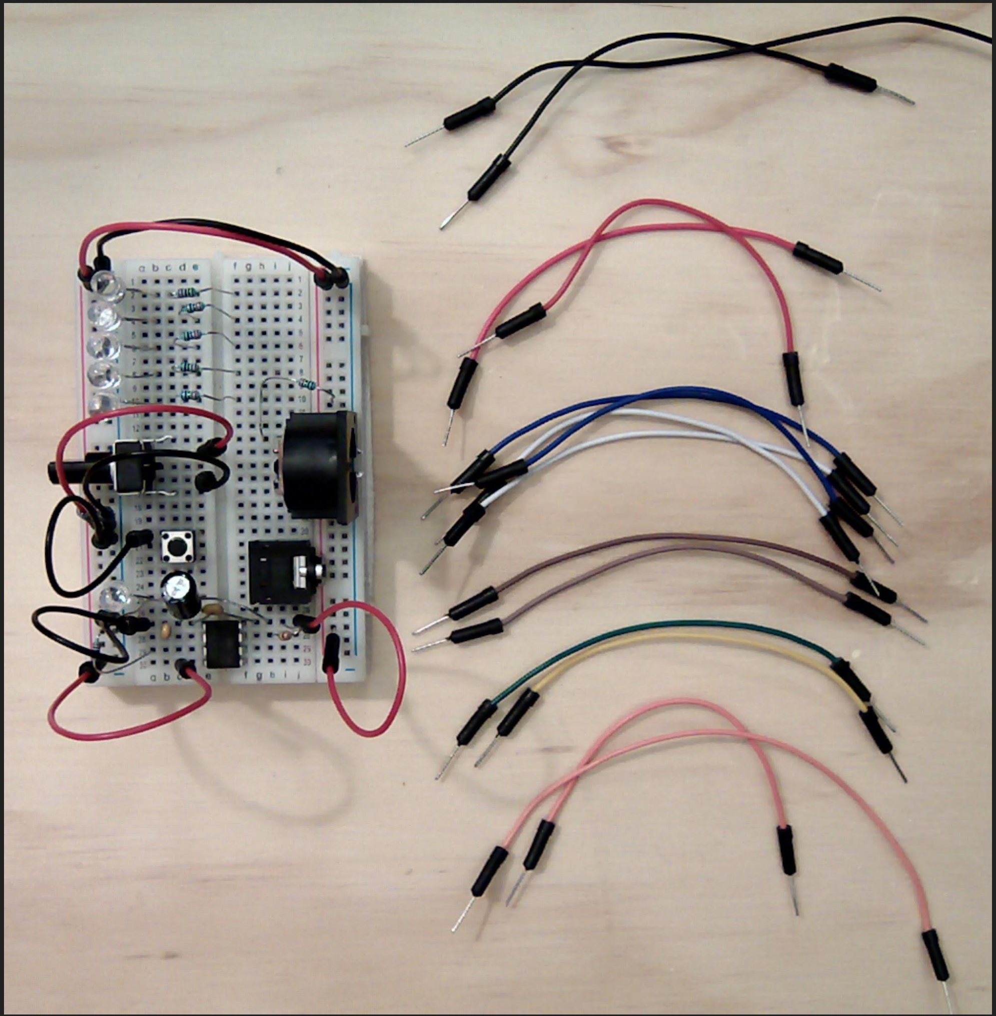
This will provide Ground connection for the potentiometer acting as a voltage divider.



Button to Ground

With a black Jumper Wire connect Button ‘top’ Pin row 20 column a to the left negative Ground Bus.

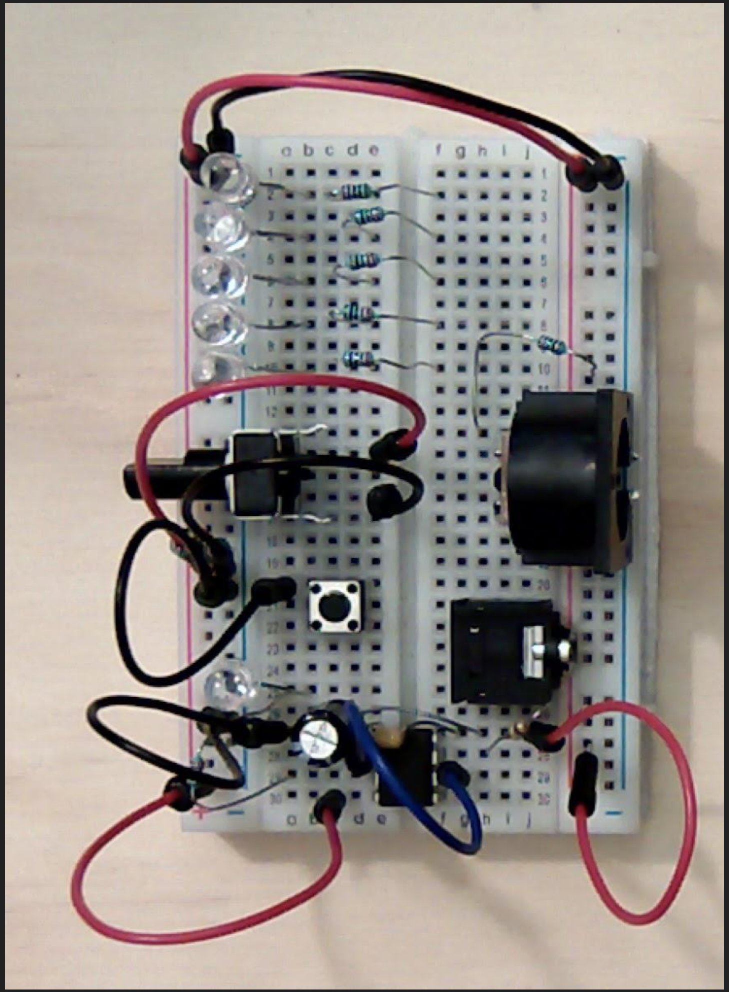
This will provide Ground connection for the button, with a pull-up set on the Arduino input pin.



555 Pin2 (Trigger) to Pin6 (Threshold)

With a blue Jumper Wire connect 555 Pin2 (Trigger) row 28 column d to 555 Pin6 (Threshold) row 29 column g

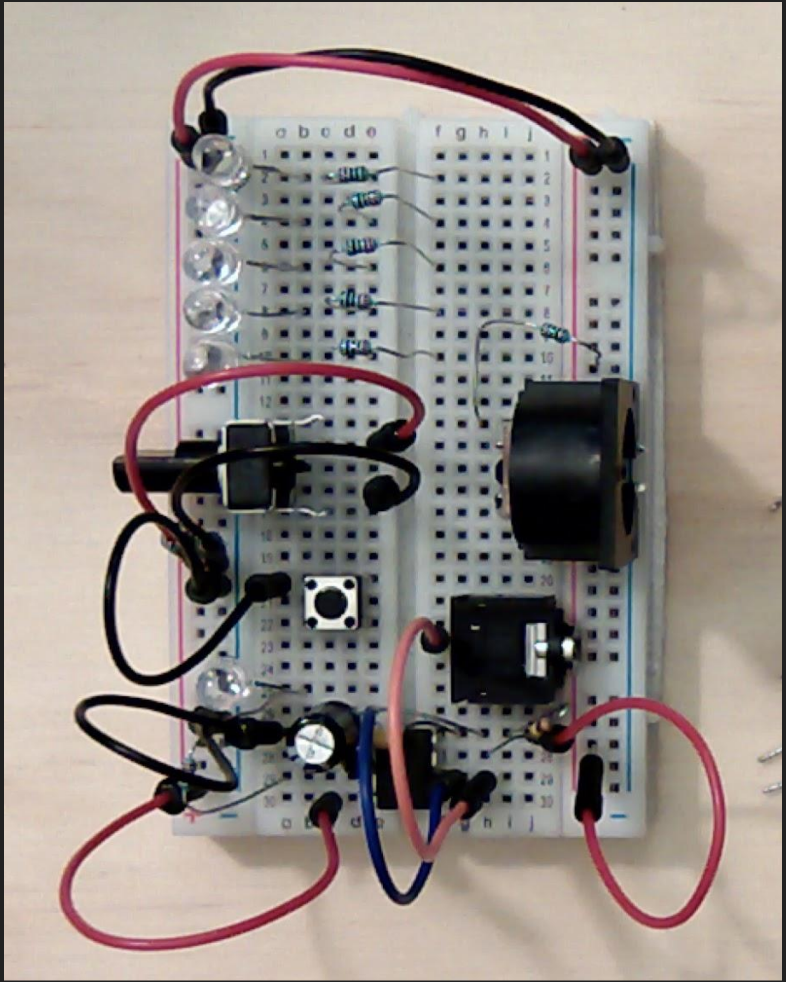
This connects the trigger to the threshold, setting the timer rate. Acting as one of the electrode connections, this is half of a variable resistor created by the 100k resistor and the two electrodes.



555 Pin6 (Threshold) to 3.5mm Jack Sleeve

With an orange Jumper Wire connect
555 Pin6 (Threshold) row 29 column h to the Sleeve pin of the 3.5 mm Jack row 23 column f

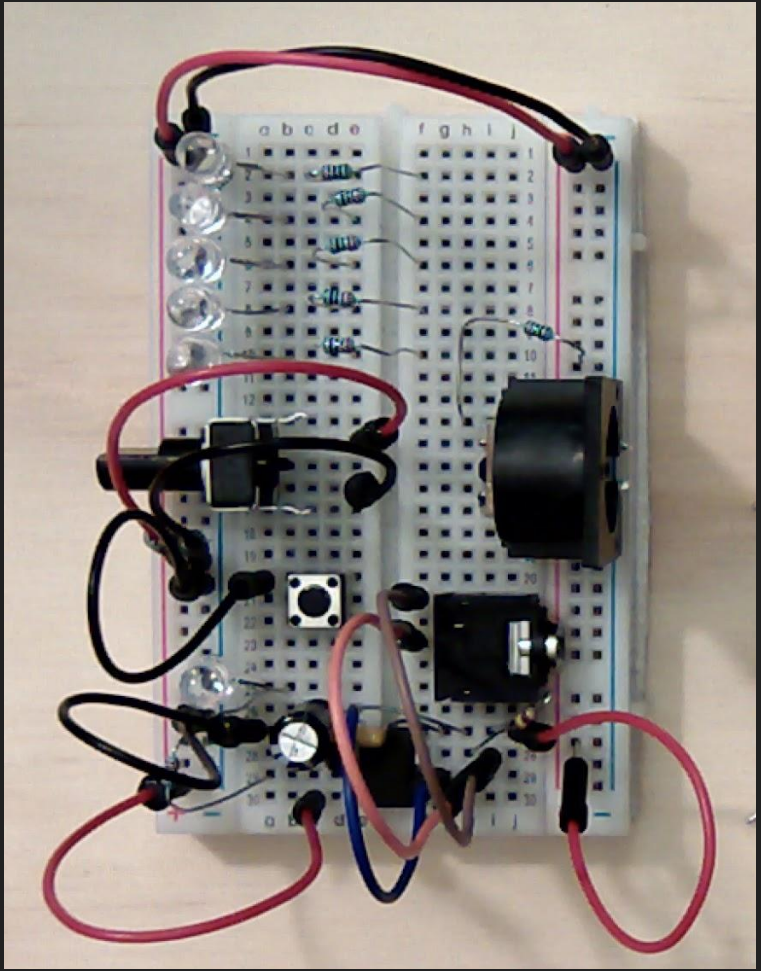
This is the first electrode connection, which will use one of the snap connections on the electrode leads.



555 Pin7 (Discharge) to 3.5mm Jack Tip

With a brown Jumper Wire connect 555 Pin7 (Discharge) row 28 column i to the Tip pin of the 3.5 mm Jack row 21 column f

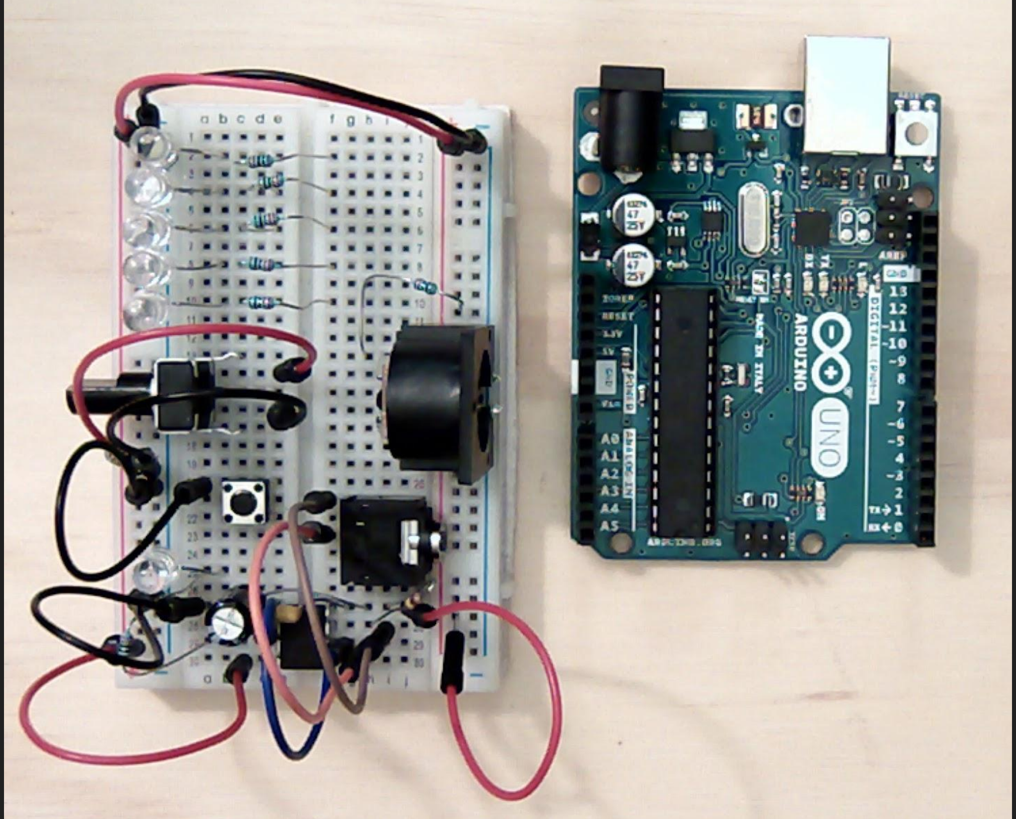
This is the second electrode connection, which will use one of the snap connections on the electrode leads.



Arduino Uno (or compatible board)

An Arduino Uno or compatible board can be used for this project.

There are 5 digital pins being used for PWM (pulse width modulation) to fade the LEDs, two Analog Inputs (A0,A1) used to read the Potentiometer and Button, one Digital Input Interrupt (INT0) on Pin2 to read the 555 timer Output, and Serial TX output for MIDI data.

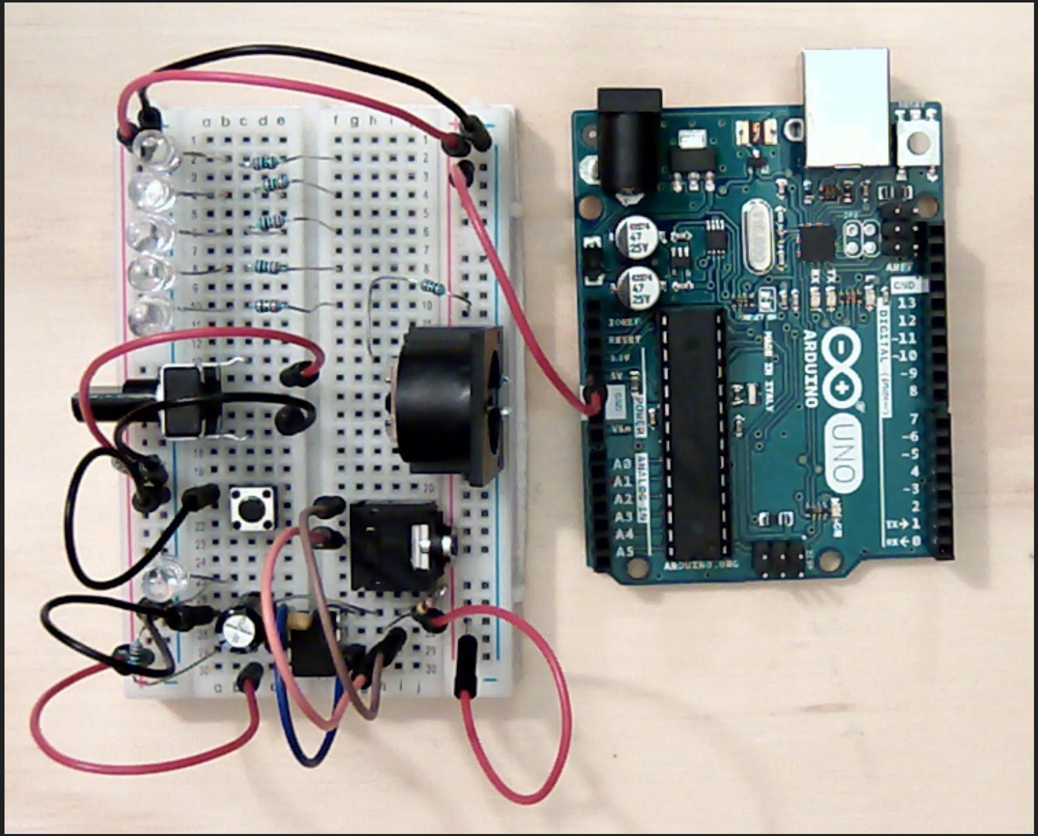


Arduino Uno to VCC

Connect a red Jumper Wire from the right positive VCC Bus to the 5v Arduino Pin.

Be careful, do NOT have your Arduino plugged into USB until a later step!!!

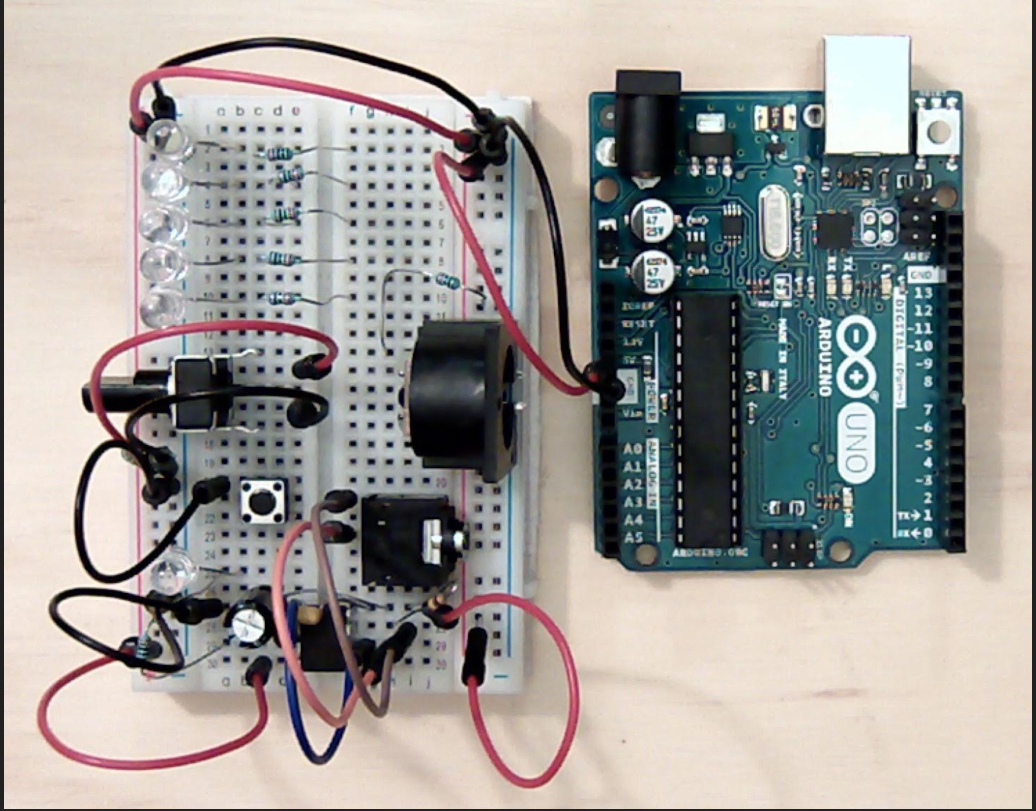
Be sure to use the 5V pin on the Arduino, not the 3V or Vin!! This way we will get a clean 5V supply



Arduino Uno to Ground

Connect a black Jumper Wire from the right negative Ground Bus to the GND Arduino Pin.

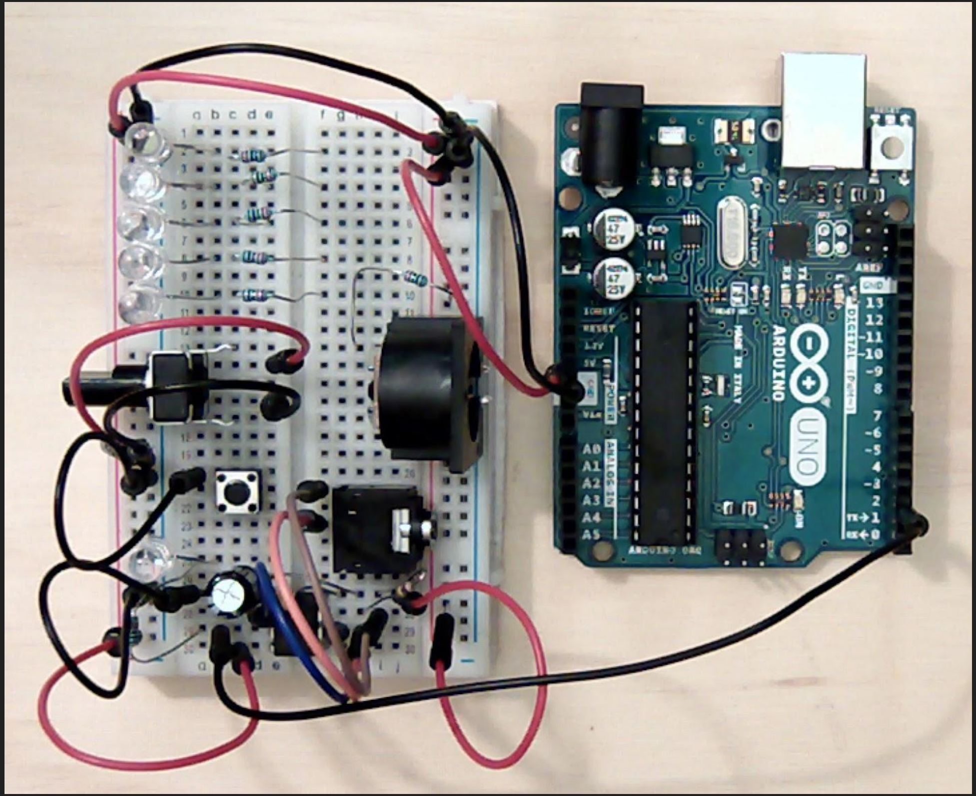
This will provide Ground connection between the breadboard circuits and the Arduino.



Arduino Uno Pin2 to 555 Output Pin3

Connect a black (long) Jumper Wire from 555 Pin3 (Output) row 29 column b to Digital Input Arduino Pin2 (INT0).

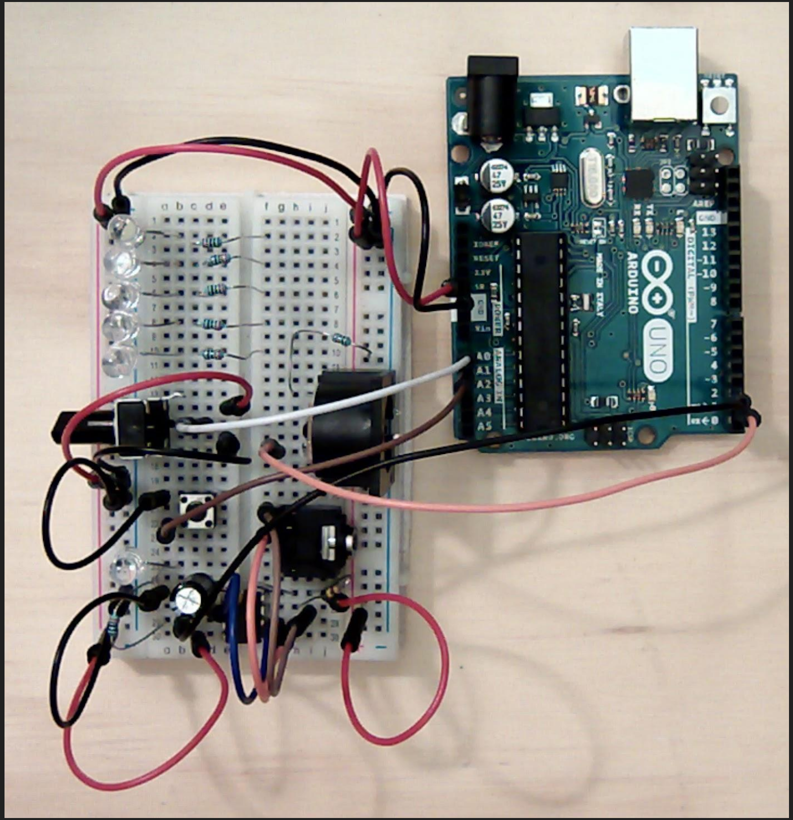
The 555 Timer output is a variable width square pulse, which is measured using an Interrupt INT0. Variations in the width of the pulse shows changes in conductivity measured across the electrodes of the 555 Timer.



Arduino Uno TX to MIDI Output

Connect an orange (long) Jumper Wire from MIDI output Pin row 17 column g to Arduino TX Pin. Connect a white Jumper Wire from row 15 column d to A0 analog input on Arduino. Connect brown/purple Jumper Wire row 22 column a to A2 analog input on Arduino.

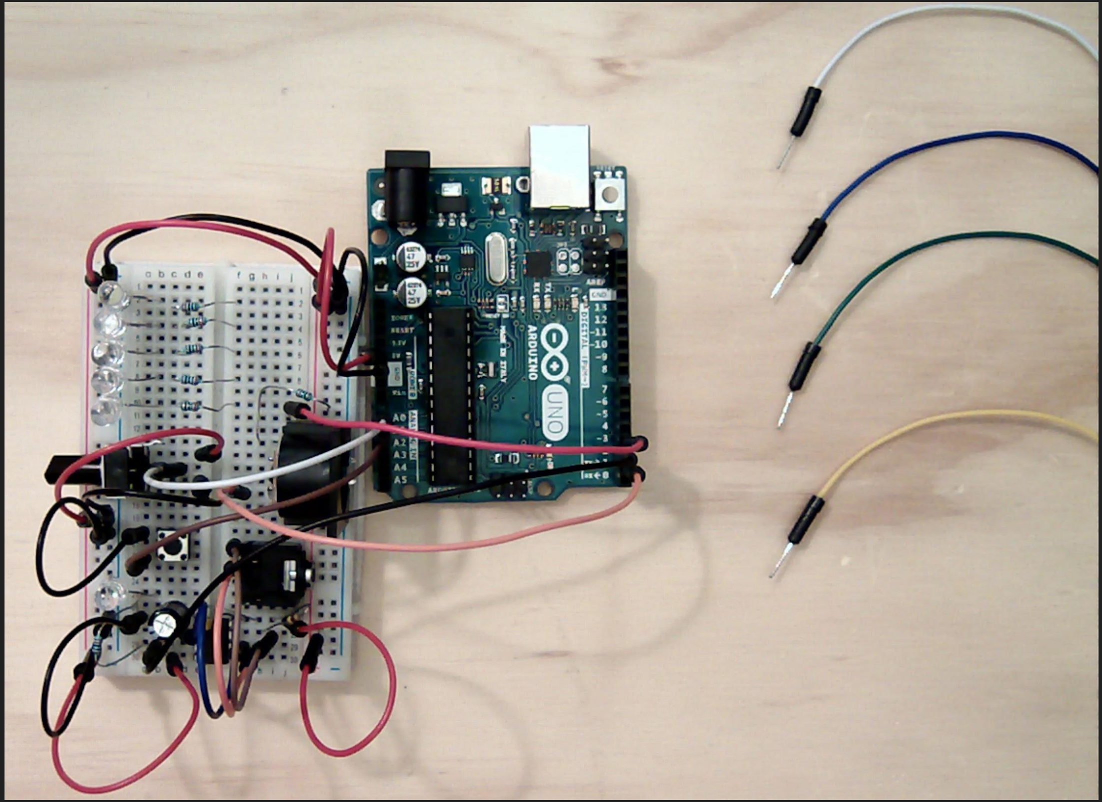
The MIDI Serial data is output at 31250 bits per second from the TX Pin of the Arduino, to the MIDI DIN output Jack on the Breadboard. Synthesizers, Computers, and other equipment can be connected to this port. Also, USB can be used to bring the MIDI data into your computer using SerialMIDI.



Red LED - row 10 to Arduino Pin3~

Connect a red Jumper Wire from the current limiting resistor of the red LED row 10 column j to Arduino Pin3~ (PWM).

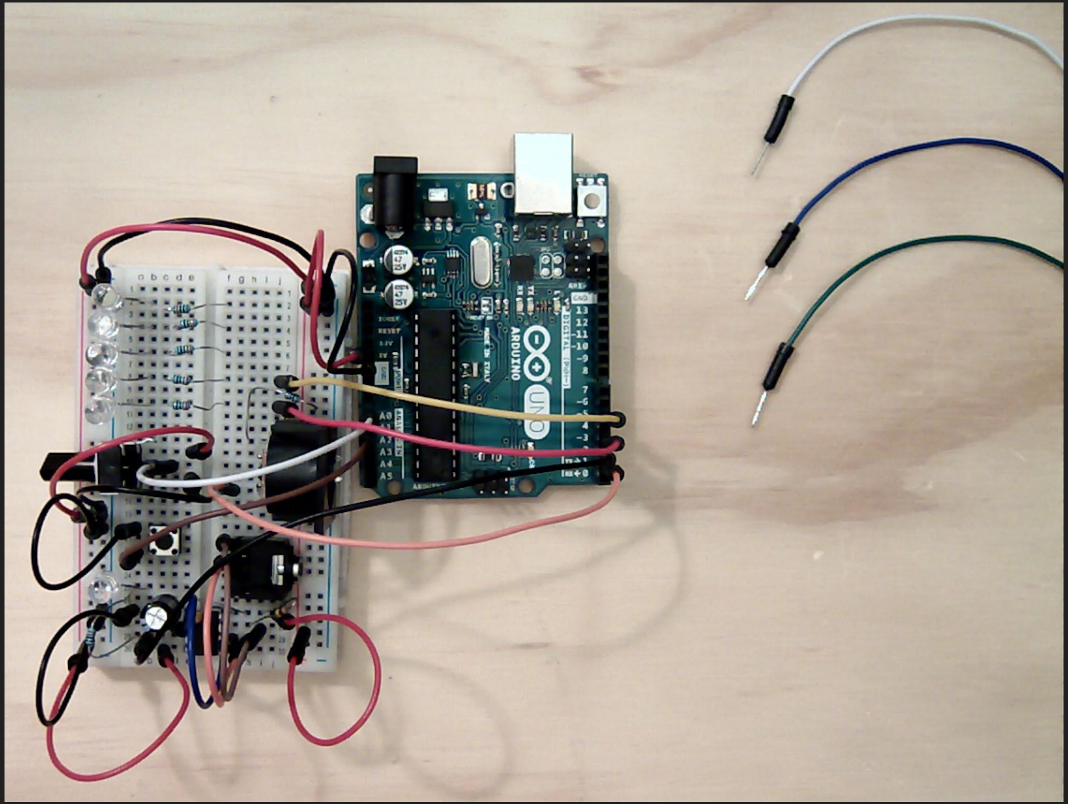
This will allow the LED to fade for the Light Show and Menu Modes.



Yellow LED - row 8 to Arduino Pin5~

Connect a yellow Jumper Wire from the current limiting resistor of the yellow LED row 8 column j to Arduino Pin5~ (PWM).

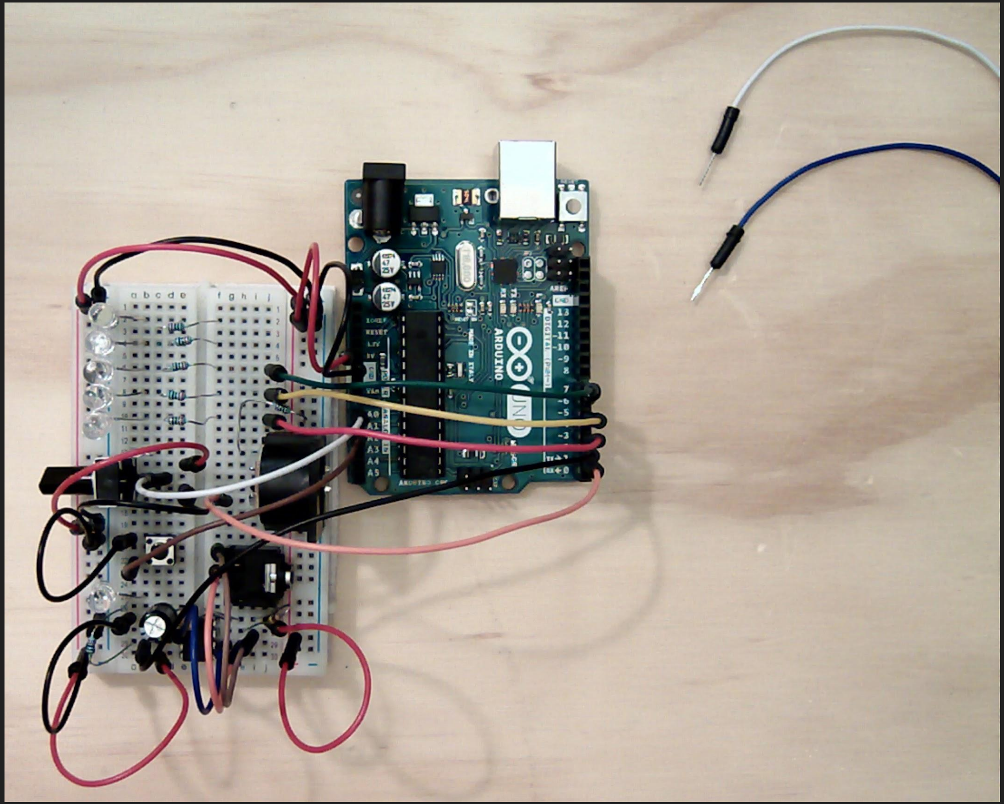
This will allow the LED to fade for the Light Show and Menu Modes.



Green LED - row 6 to Arduino Pin6~

Connect a green Jumper Wire from the current limiting resistor of the green LED row 6 column j to Arduino Pin6~ (PWM).

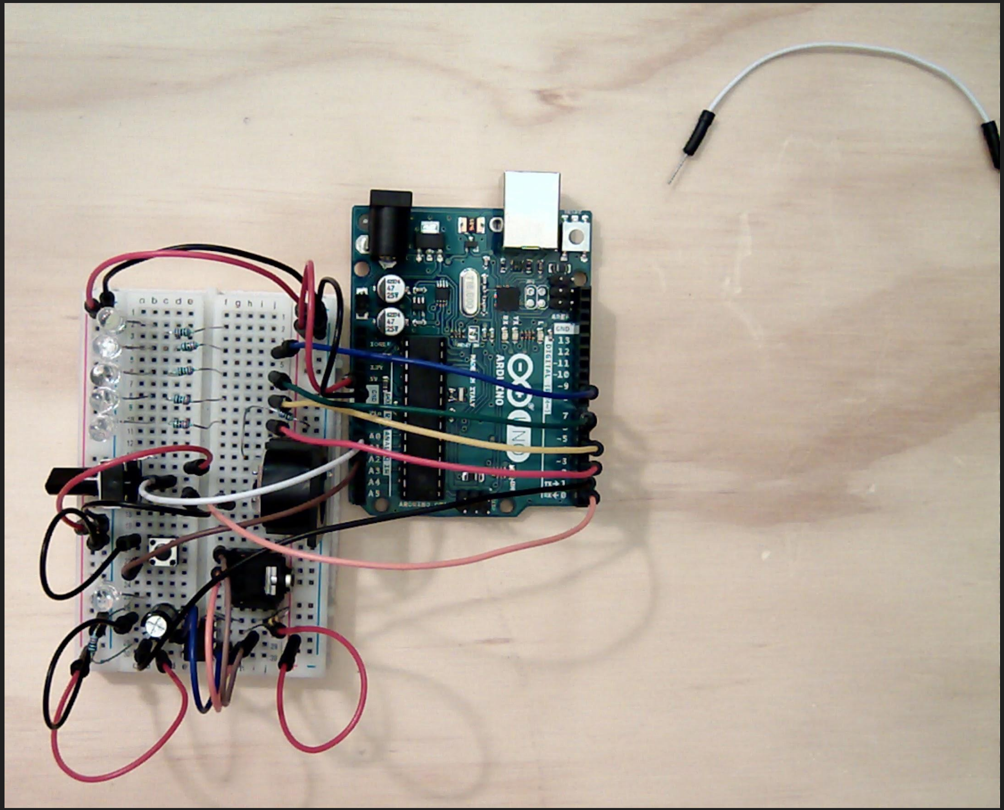
This will allow the LED to fade for the Light Show and Menu Modes.



Blue LED - row 4 to Arduino Pin9~

Connect a blue Jumper Wire from the current limiting resistor of the blue LED row 4 column j to Arduino Pin9~ (PWM).

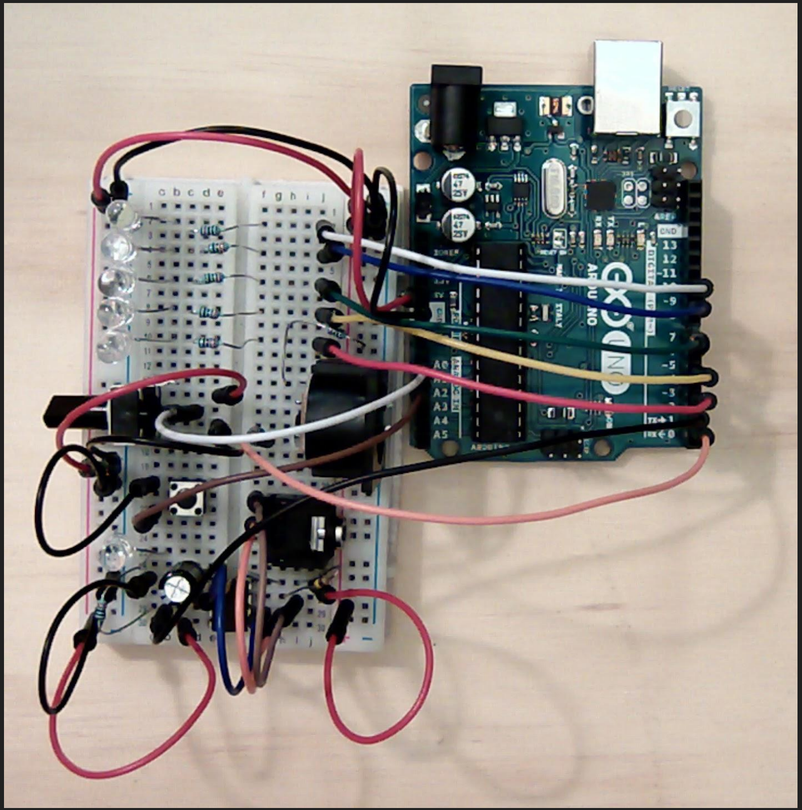
This will allow the LED to fade for the Light Show and Menu Modes.



White LED - row 2 to Arduino Pin10~

Connect a white Jumper Wire from the current limiting resistor of the white LED row 2 column j to Arduino Pin10~ (PWM).

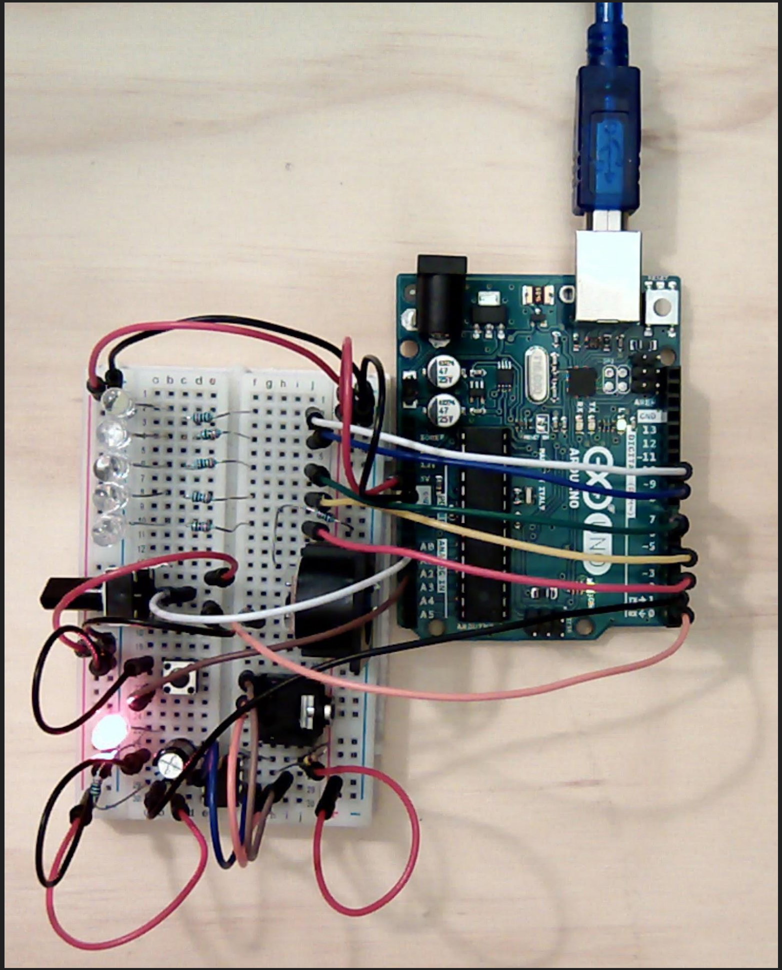
This will allow the LED to fade for the Light Show and Menu Modes.



Arduino - USB Power

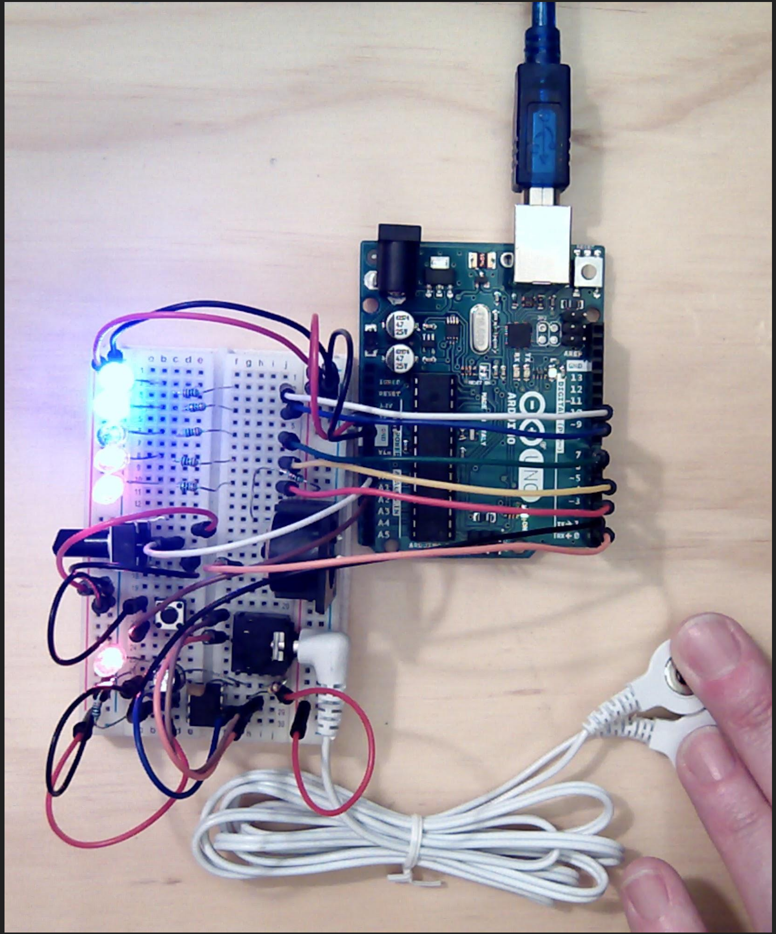
Connect a USB cable to the Arduino and attach to a computer or USB battery pack. The red LED near the 555 Timer should illuminate, and may flash very slowly.

Upload the *Biodata Sonification Arduino Sketch* to your Arduino Uno using the Arduino IDE. Once loaded, the 5 LEDs will light up in sequence showing the program has been written successfully.

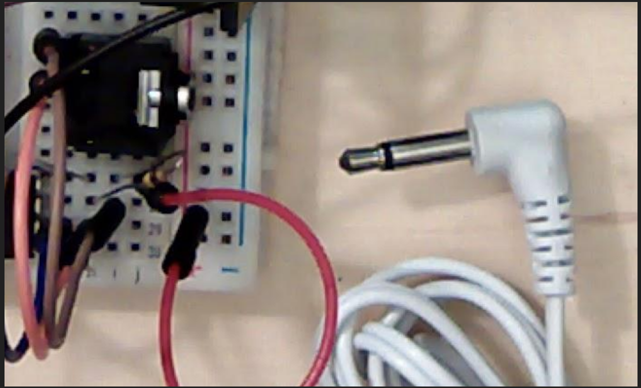


Electrodes - 3.5mm Jack

Plug the Electrode leads into the 3.5mm Jack, and touch the metal ends of the trodes to trigger the Biodata Sonification system to generate MIDI data and LED light show.



Hold the 3.5mm Jack carefully into the breadboard while inserting the electrode plug.



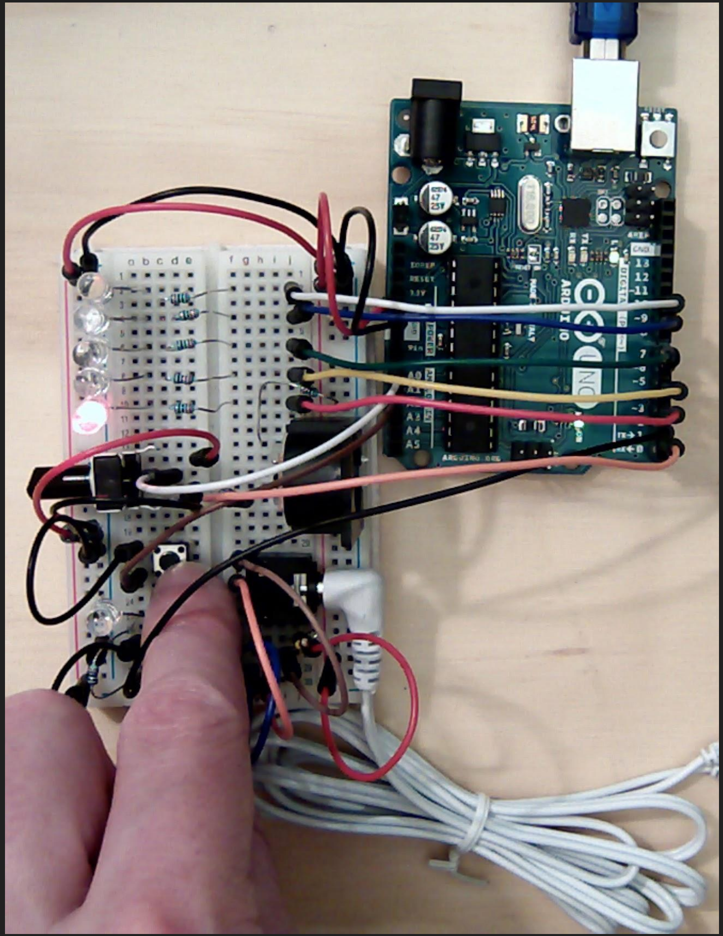
Mode Button

By pressing the Button, Menu Mode is entered. One LED will pulse, indicating the current Menu.

Turning the Potentiometer will change the Menu and indicate using the LEDs.

Pressing the button again will select the Menu (Threshold, Note Scale, MIDI Channel, Brightness) and enter Value Selection Mode

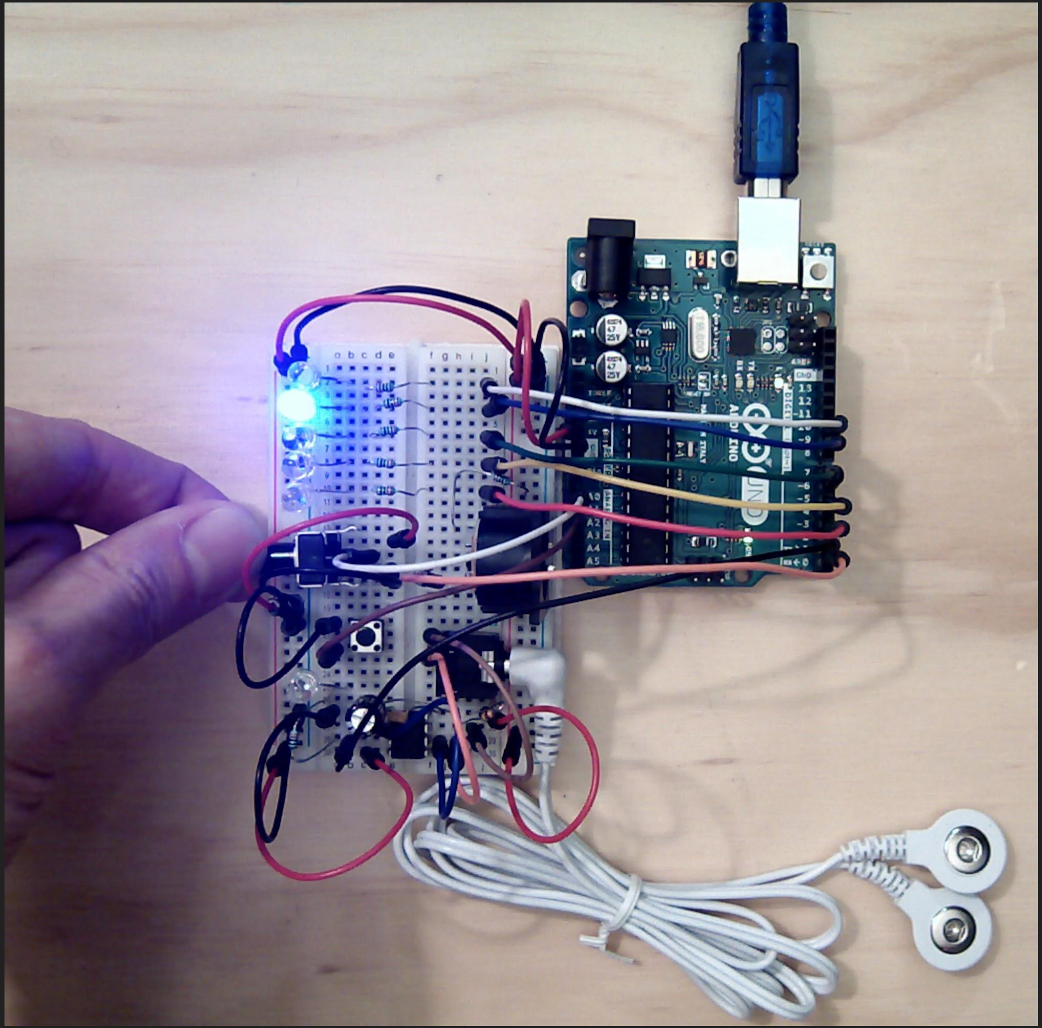
After 15 seconds, if the Potentiometer has not been turned or the button not pressed again, the program will exit Menu Mode and resume the normal Light Show



Value Knob

Depending on the Menu chosen, turning the Potentiometer will change the associated Value and indicate using the LEDs.

(Threshold, Note Scale, MIDI Channel, Brightness)

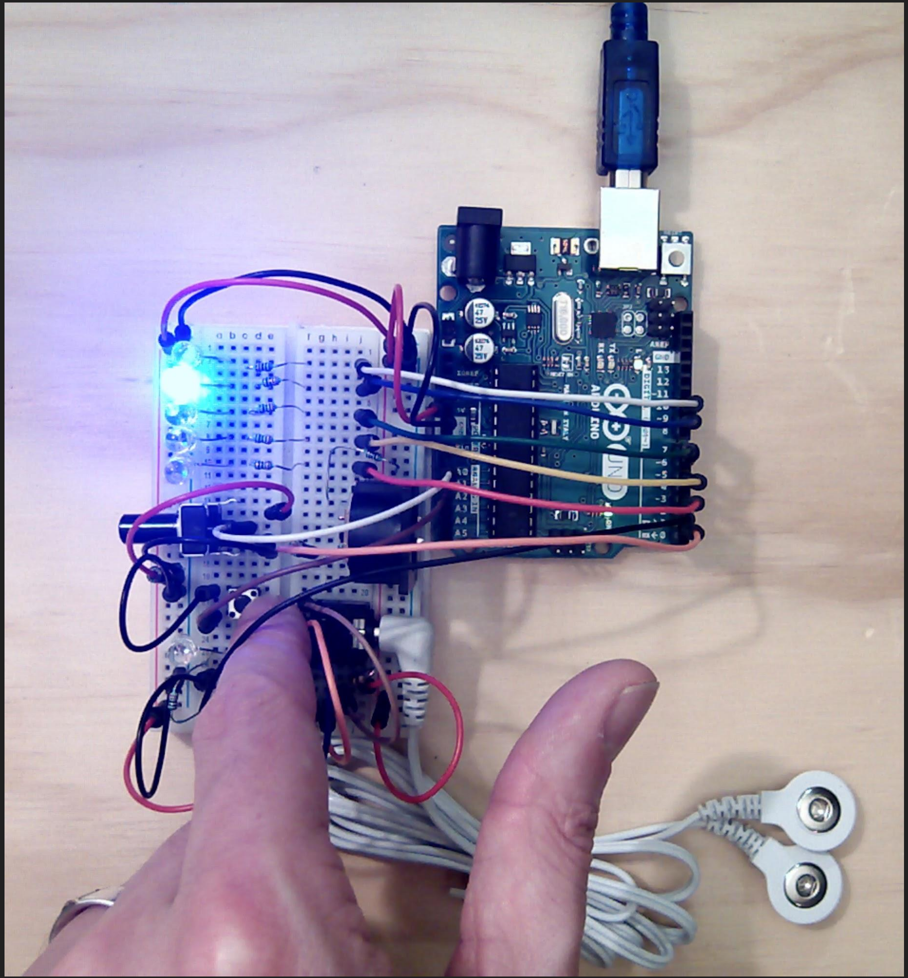


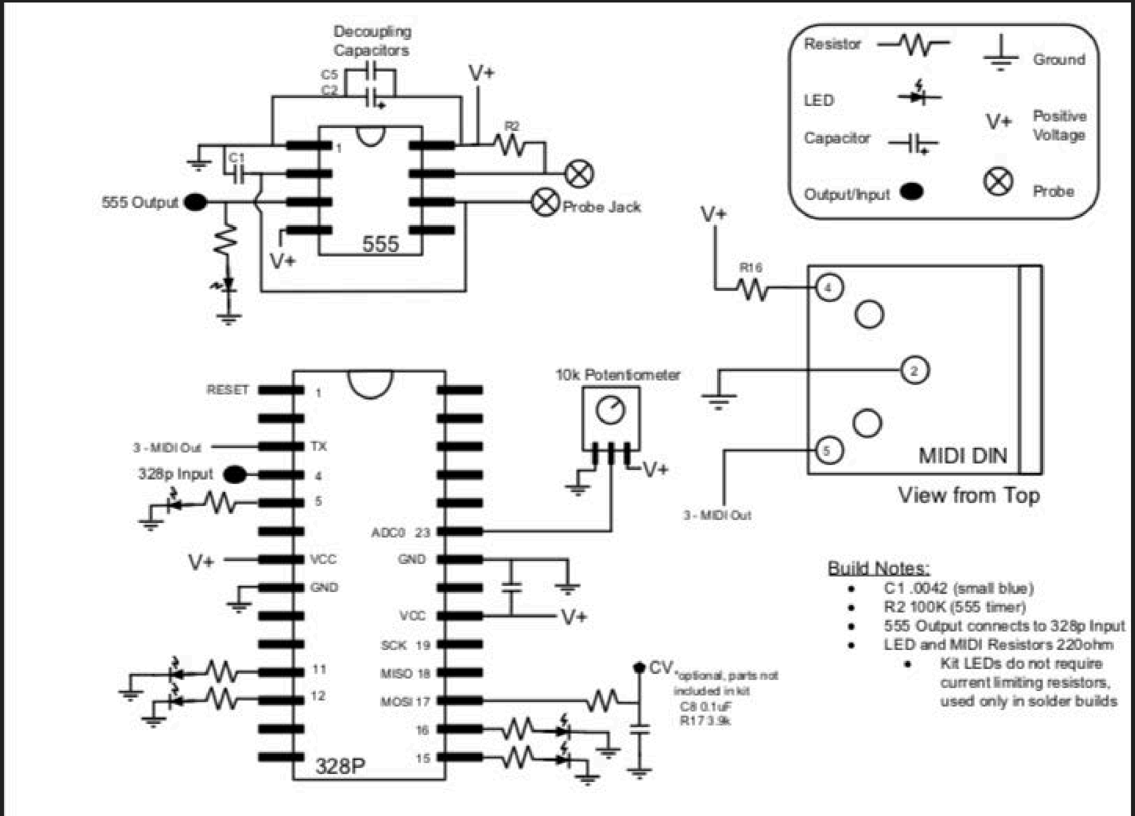
Store Settings Button

Press the button from within a Menu to store the Value.

The Menu Mode will exit, and the program will return to the normal Light Show.

All parameters changed will be active, and will save to the EEPROM memory. This allows the system to remember settings after powering off. (coming soon)





Build Notes:

- C1 .0042 (small blue)
- R2 100K (555 timer)
- 555 Output connects to 328p Input
- LED and MIDI Resistors 220ohm
    - Kit LEDs do not require current limiting resistors, used only in solder builds