

# Introduction

The following instructions aim to show you how to build, using electrical components, your own version of the 80's classic game of Simon, where one puts their memory to the test in matching sequences of colorful lights and sound. In addition it will have not just four buttons, but five! The steps involved will assume you have some experience with electronics, whether you've taken a course on the topic or a just simply have it as a hobby.

## Materials

- Computer (Windows, Mac OS X, or Linux) with the Arduino software
  - <http://arduino.cc/en/Main/Software>
- Arduino Uno R3 (1x)
  - <http://www.radioshack.com/product/index.jsp?productId=3897297>
- USB 2.0 A/B cable (1x)
  - <http://www.radioshack.com/product/index.jsp?productId=2261723>
- 5mm LED (5x)
  - <http://www.adafruit.com/products/754>
- Tactile switch (5x)
  - <http://www.adafruit.com/products/1010>
- Piezo buzzer (1x)
  - <https://www.adafruit.com/products/160>

The following materials are available in the EE215 Lab Kit, which can be purchased at the EE store (University of Washington, Seattle campus) <https://www.ee.washington.edu/stores/>. Or, if you prefer, they can be purchased separately.

- Breadboard (1x)
  - <https://www.adafruit.com/products/239>
- Toggle/slide switch (1x)
  - <https://www.adafruit.com/products/805>
- Assortment of jumper wire
  - You will need about 30 strands of jumper wire of various lengths, ranging from 10mm to 150mm
  - <http://www.radioshack.com/product/index.jsp?productId=12165717>
- 9V battery connector (1x)
  - <http://www.radioshack.com/product/index.jsp?productId=2062218>
- 9V battery (1x)
  - <http://www.radioshack.com/product/index.jsp?productId=3897297>
- 1.5K ohm resistor (1x)
  - <http://www.radioshack.com/product/index.jsp?productId=2062324>
- 100 ohm resistor (1x)
  - <http://www.radioshack.com/product/index.jsp?productId=2062315>

### Figure 1 - Wiring

- ### 1.1 Matching Figure 1, use the jumper wire to connect the various locations on the breadboard itself as well as between the breadboard and Arduino.



The wiring is setup as it is for future placement of components. Note that your placement doesn't necessarily have to be exactly the same because of the way breadboards themselves are wired up. If you're interested, more information on breadboards can be found at the Wikipedia page, <http://en.wikipedia.org/wiki/Breadboard>.

## Step 2: Connecting the LEDs

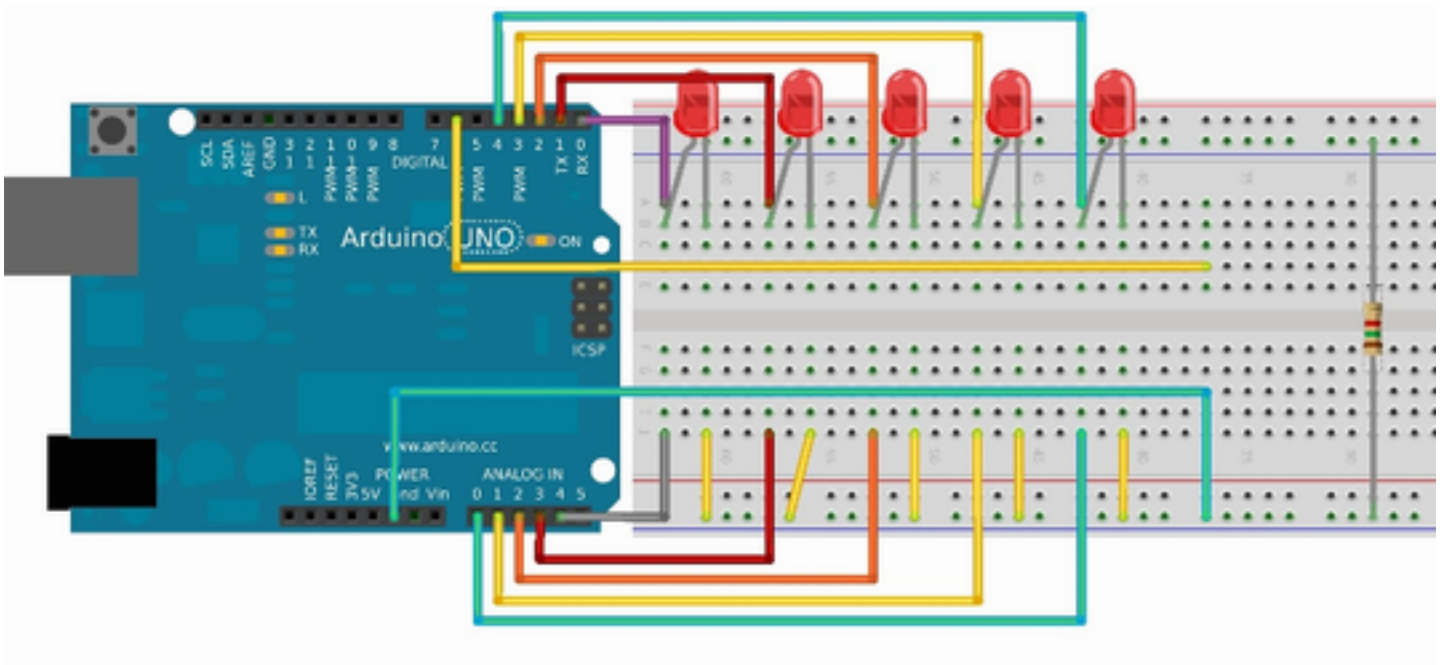


Figure 2 - LED positioning

- 2.1** Plug the five LEDs into the breadboard as shown in Figure 2. Be sure to get each LED polarity correct: positive end of the LED paired next to the longer wire running to the Arduino and negative end paired next to the shorter wire. If you are unsure as to which end is positive or negative, the positive end is typically longer than the other.
- 2.2** Plug in the 1.5k ohm resistor; also shown in Figure 2. It is the one that, if held up with color bands vertical, has the color order from left to right *brown, green, red, gold*.



The purpose of this resistor is to limit the amount of current running through your LEDs, otherwise, plugging them directly into a voltage source without such protection could easily destroy them.



If you're interested, more information on LEDs can be found at the Wikipedia page, <http://en.wikipedia.org/wiki/Led>. Also, a very helpful website for calculating what size of resistors to use with various LEDs, <http://ledcalculator.net/>.

## Step 3: Connecting the Buttons

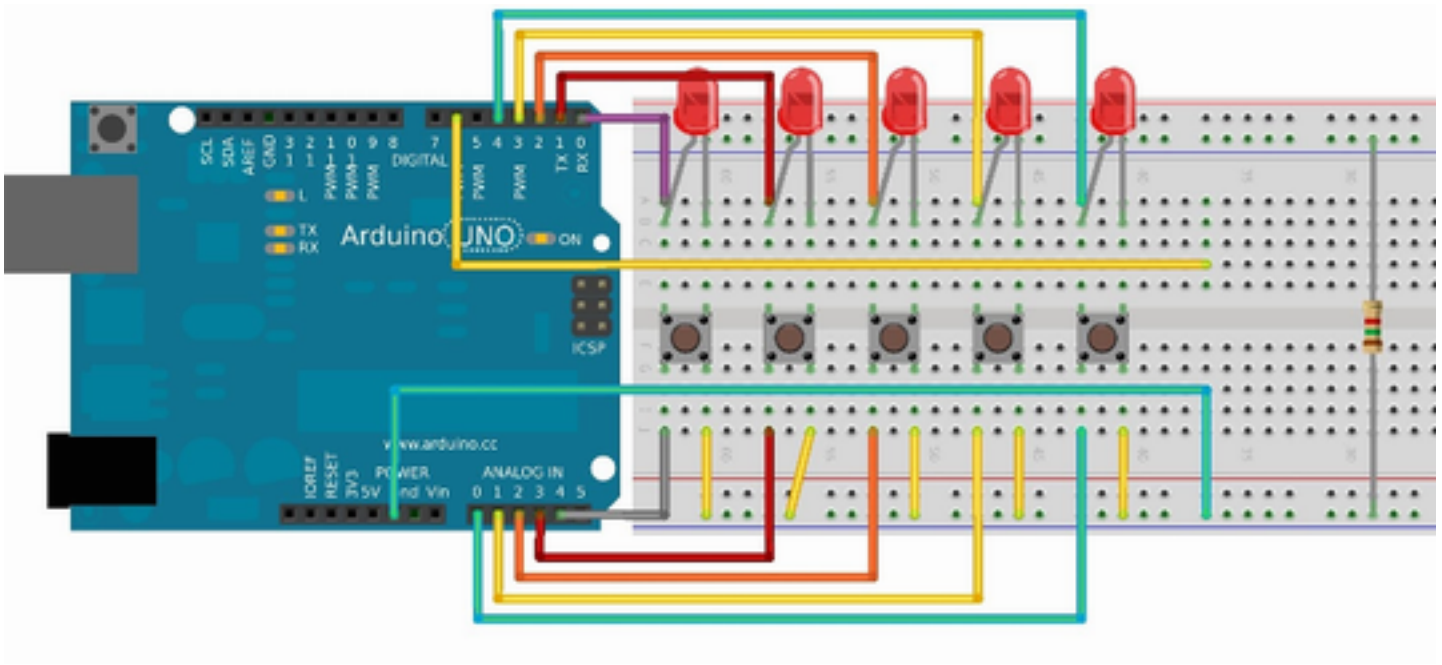


Figure 3 - Push button positioning

- 3.1** Plug the five tactile push buttons into the breadboard as shown in Figure 3. Be sure to match the orientation to that shown.
- 3.2** Note that the push button pins do not span to the top half of the breadboard. If necessary, you can fold the two top pins up so that they cannot fit into the holes of the breadboard.
- 3.3** The pins on your push buttons will likely be a bit oddly shaped and will not fit well into the breadboard as they are often designed for use with printed circuit boards. This can be easily fixed by applying pressure to them with a set of pliers in order to get them straighter and fit better.

?

The following website provides a very nice explanation of push buttons and an example of straightening the pins: <http://www.ladyada.net/learn/arduino/lesson5.html>.

- 3.4** If your push buttons came with colorful caps, now is a good time to attach them and in any color pattern you wish.

## Step 4: Connecting the Speaker

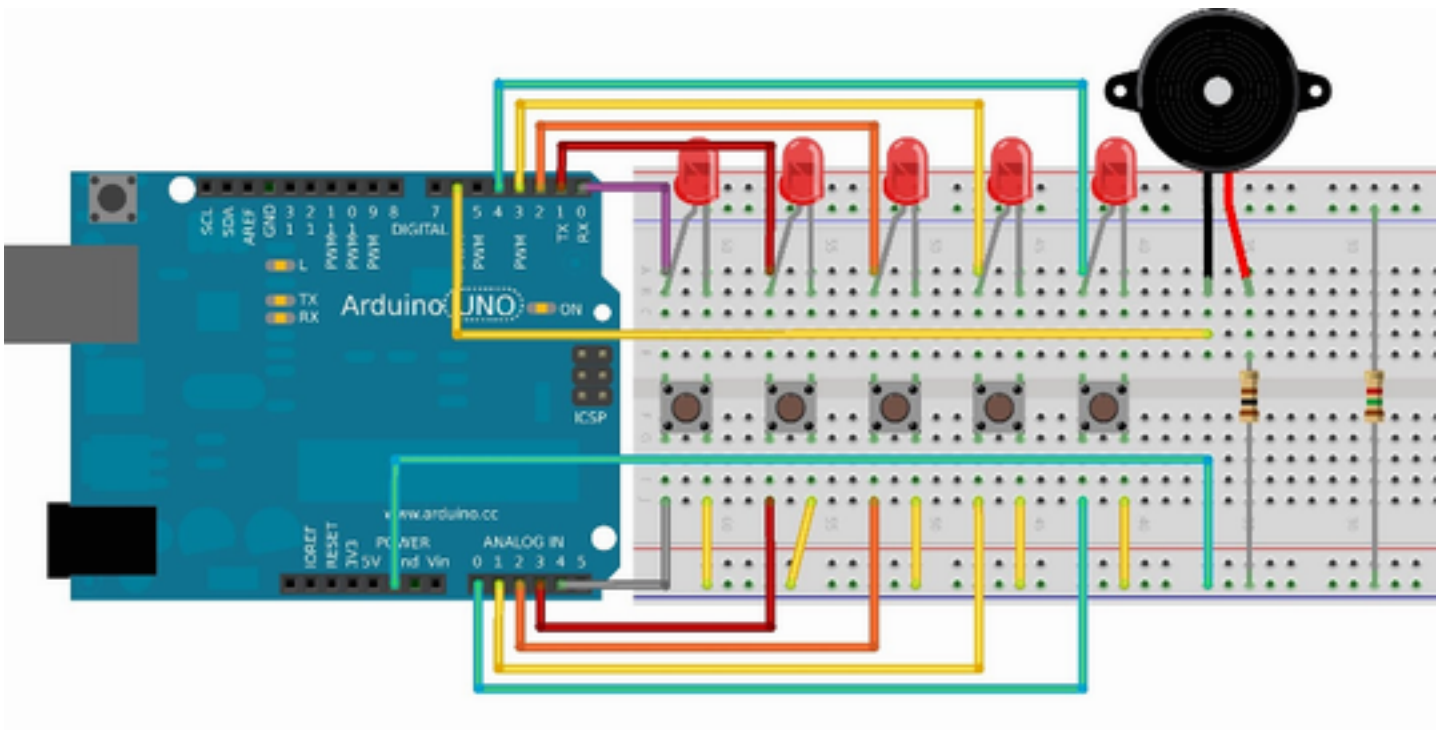


Figure 4 - Speaker positioning

- 4.1** Plug the speaker into the breadboard as shown in Figure 4. Note that its polarity isn't an issue as the figure may suggest.
- 4.2** Plug in the 100 ohm resistor; also shown in Figure 4. It is the one that, if held up with color bands vertical, has the color order from left to right *brown, black, brown, gold*.

## Step 5: Uploading the Software

**5.1** Download and install the latest Arduino software. You can visit their website at <http://arduino.cc/en/Guide/HomePage> to download it and for a good introduction to the product as a whole and getting started with the software.

**5.2** Visit <https://github.com/go4ble/simon5> and download ButtonLightTest. Upload it to the arduino.

This program serves to test that all of your current connections are correct. When you press a button, the corresponding LED should light up and you should hear a particular tone. If something seems to not be working properly, consult the following list for troubleshooting:

- A particular LED not lighting up?
  - It may be plugged in backwards. Try flipping it.
  - Its button maybe oriented wrong. Try rotating the button 90°.
  - It may be damaged. Try using a different LED in that spot.
- Wrong LED lighting up?
  - Your wiring is likely mixed somewhere. Consult Figure 1 and make sure everything matches up.
- Nothing working?
  - Make sure the Arduino is still plugged into your computer. That is how it's being powered (for now).

**5.3** If everything seems to be properly working, go back to <https://github.com/go4ble/simon5> and download Simon5. Upload it to the Arduino. When the upload is complete, you should see a sequence of lights with some tones indicating that the game is starting. If you let it sit for a several seconds, a random light will flash with a tone, wait, and then indicate a losing round.

## Conclusion

Congratulations! You have completed enough to play a game of Simon 5. Just press the *RESET* button on the Arduino, and a new game will start for you. To play, repeat the pattern initially displayed. It will increase by one after each successful guess. The game will run through a sequence of 15, after which you win. At any time, you can edit the Simon5.ino source file and change this value, as well as others, to customize the game as you wish. There is another step after this that is optional where you can set up the game to be powered by a battery and make it portable!



## Step 6: Connecting the Power Source (optional)

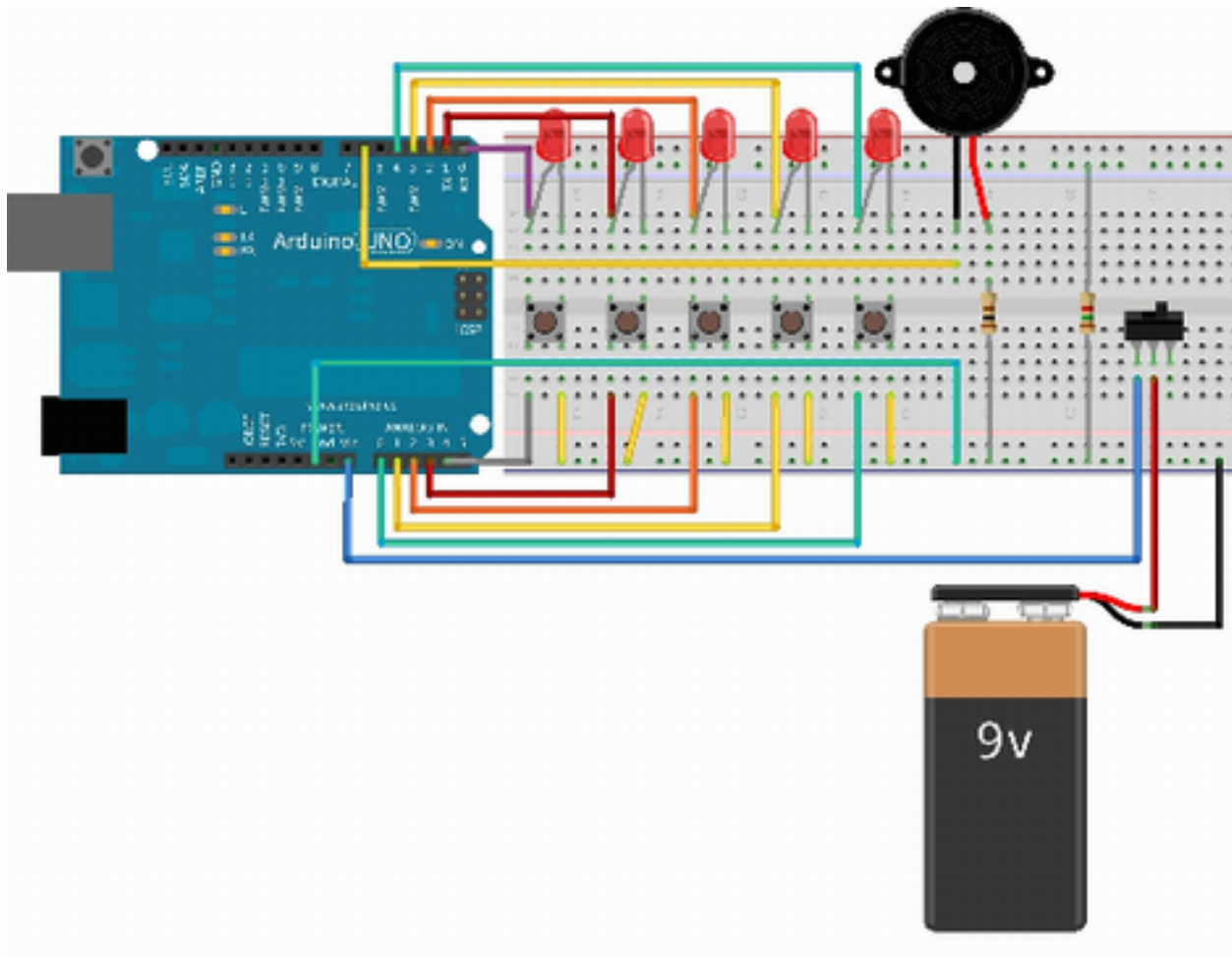


Figure 5 - Battery hookup

- 6.1** Make sure the Arduino is unplugged from the computer.
- 6.2** As shown in Figure 5, plug in the toggle/slide switch such that the switch pairs the positive end of the battery connector with Vin of the Arduino board. Plug the negative end of the battery connector in as shown.
- 6.3** Plug the 9V battery into the battery connector.
- 6.4** Use the switch to turn the unit on and off.