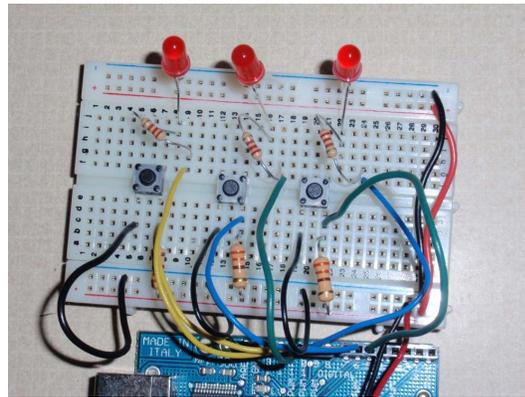
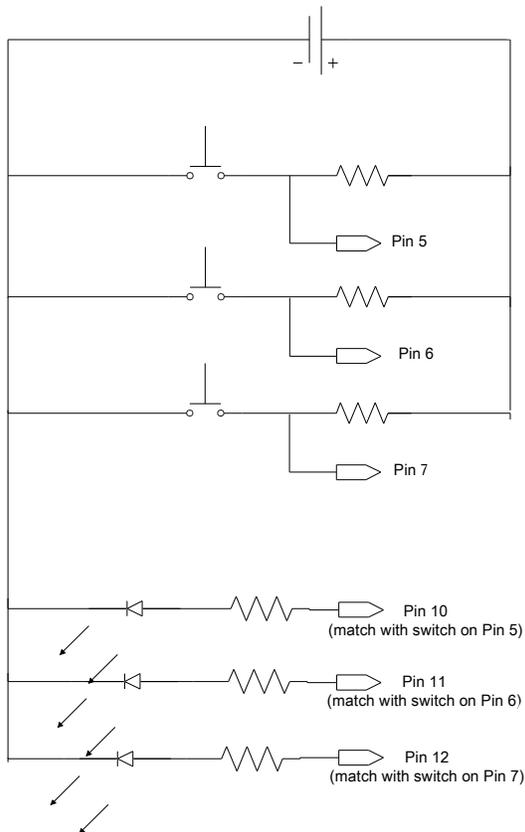


## LEDs And Sensors Part 1: Play Simon

Do you remember the electronic game *Simon*. You press buttons on Simon and they light up and a musical note sounds. To play, you first observe a sequence of sounds and lights and then try to re-create that same sequence by manually pressing the buttons. The sequence would get longer and harder to reproduce after each successful try. In this exercise, we are going to make a somewhat simplified version of Simon with the aid of an on-line tutorial by TuxRadar entitled, [Arduino hardware hacking: Part 2](#).

Follow the instructions in the [tutorial](#). You will need to create a breadboard layout similar to that depicted in the photo and schematic below.



You can download the code from the tutorial, but you'll need to make a few obvious changes before it will compile. Try it out.

### The Hard Part

Once you've gotten the code to run, play the game. You'll note that there is a problem: when you depress the correct button sequence, it doesn't recognize it as correct. Your job is to figure out why and fix the problem. Here's a hint. You may want to look at the input and output values as the program runs, and you can do that by writing information

to the serial port just as we did in [Sound Part 2](#). The first half of [Ladyada Lesson 4](#) provides more extensive information on using the serial port.

One more hint. The Serial library is a bit “C like” in the construct of the print() and println() methods. You can not append arguments using the “+” operator; check the [on-line reference](#).

### **Extra Credit---Try it, it's easy!**

Modify the program so that each button also plays a certain note and that note is included in the description of the button sequence you try to mimic when playing the game. To make this an interesting part of the game, modify the program so that depressing a button causes the corresponding LED to flash and the associated note to play. This modification will enable the game player to see and hear the sequence they are creating.