

# Arduino Light Theremin

Step by step introduction!

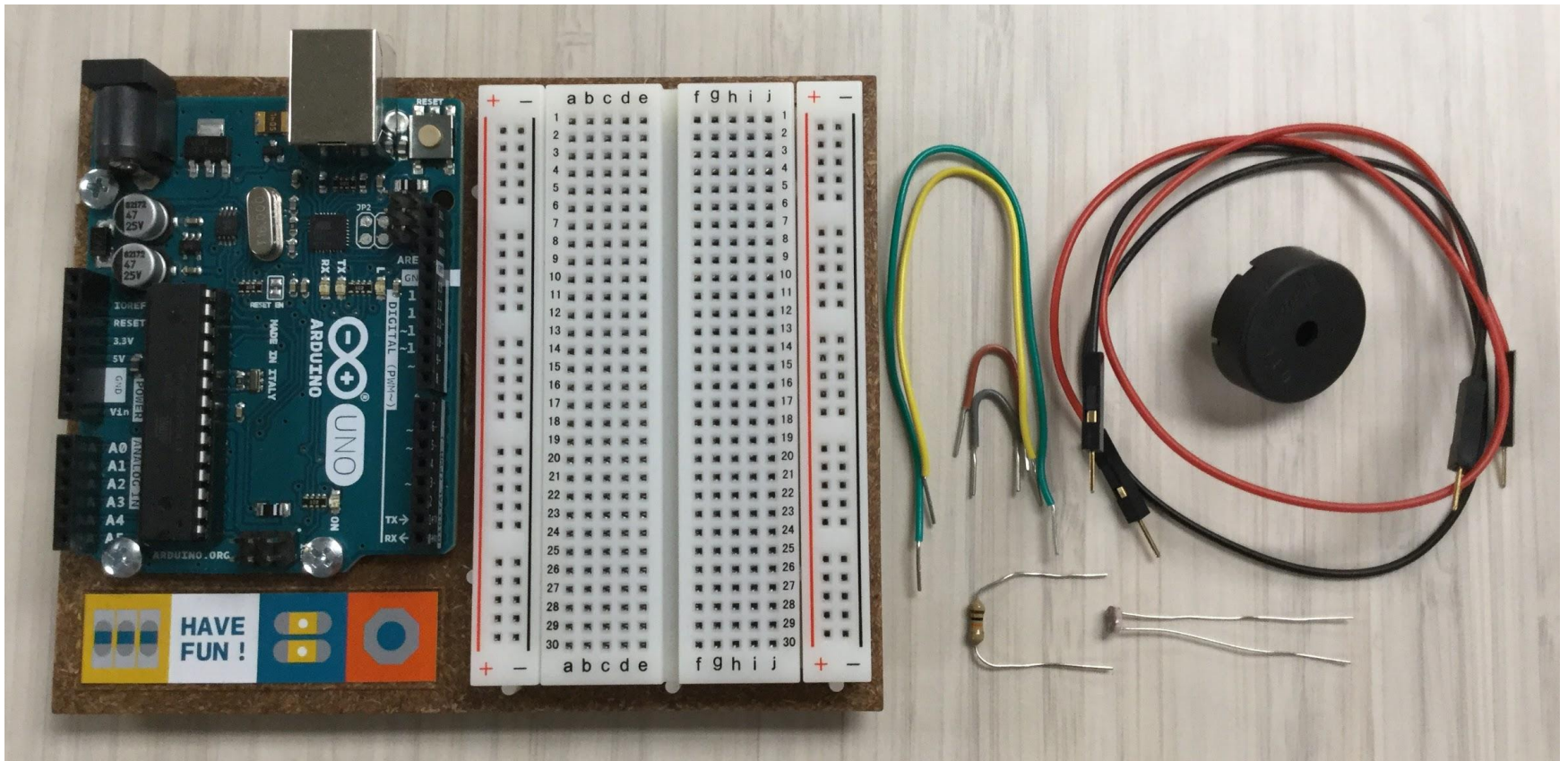
based on Arduino Starter Kit, Project 06  
<http://www.arduino.cc/starterKit>  
workshop for The MILL @ UIdaho Library  
cc-by-sa 2016

For this workshop you will need:

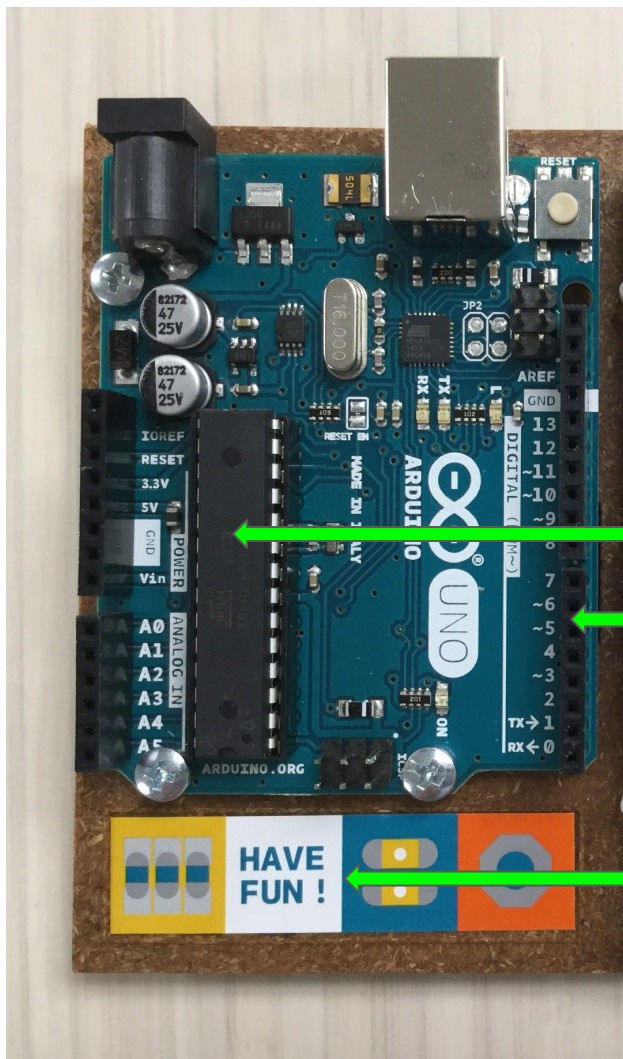
- Arduino UNO and usb cable
- a few basic electronic components, as found in most Arduino starter kits: a breadboard, some jumper wires, a piezo, a 10k resistor, and a photoresistor.
- Arduino Software IDE installed on a computer, <https://www.arduino.cc/en/Main/Software> or use Codebender on Chromebook, <https://codebender.cc/>

Plus the *sketches* to program the theremin, which can be downloaded from Github:

1. download the ZIP package from the repository, <https://github.com/evanwill/arduinoTeaching/archive/master.zip>
2. unzip the file ("arduinoTeaching-master.zip")
3. move the contents of the unzipped folder to your Arduino directory (found in Documents).



Start with these parts from an Arduino Starter Kit:  
UNO, breadboard, jumper wires, piezo, 10k resistor, and photoresistor.



# Arduino UNO

This is your *Microcontroller!*

Arduino is an easy to use prototyping board based on **open-source** hardware and software.

Microchip: ATmega328P (8-bit, 16MHz)

Pins: 14 digital output/input, 6 analog input

Simple programming language,  
<https://www.arduino.cc/en/Reference/HomePage>

Remember, Have Fun!

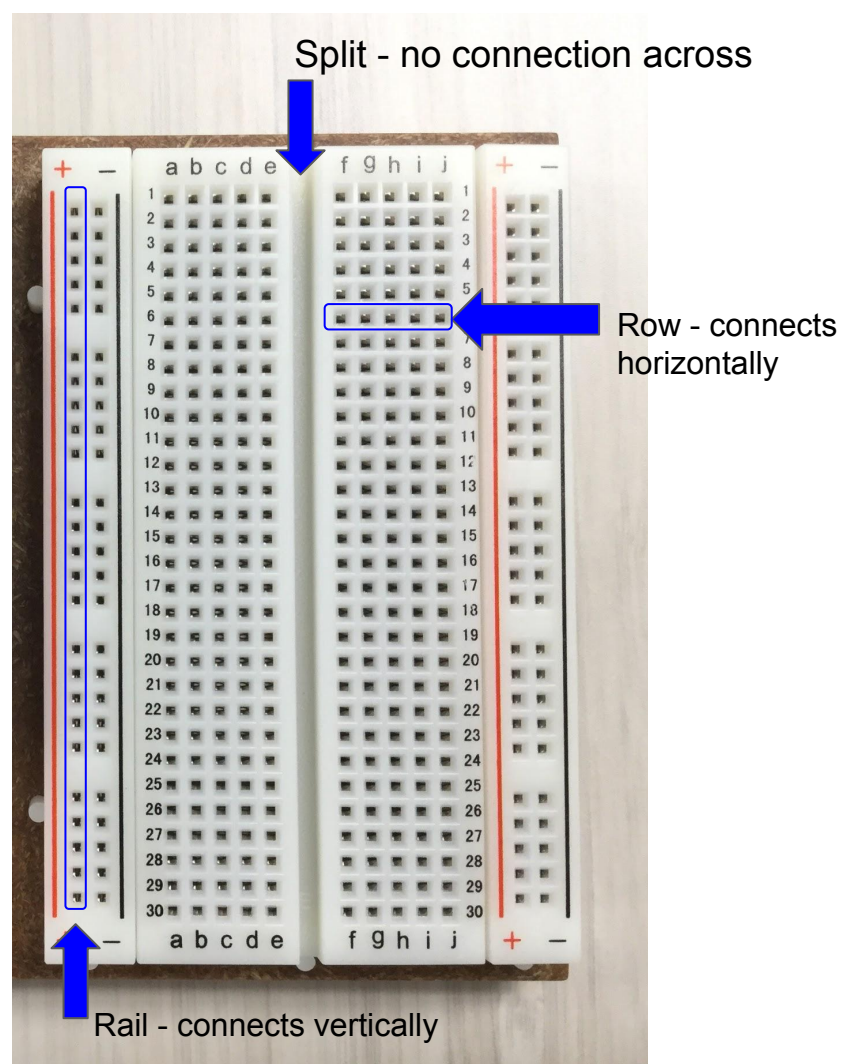
# Solderless Breadboard

Makes it easy to create circuits without soldering!

Each pin hole on the *Rails* is connected vertically.

Each hole in the *Rows* is connected horizontally.

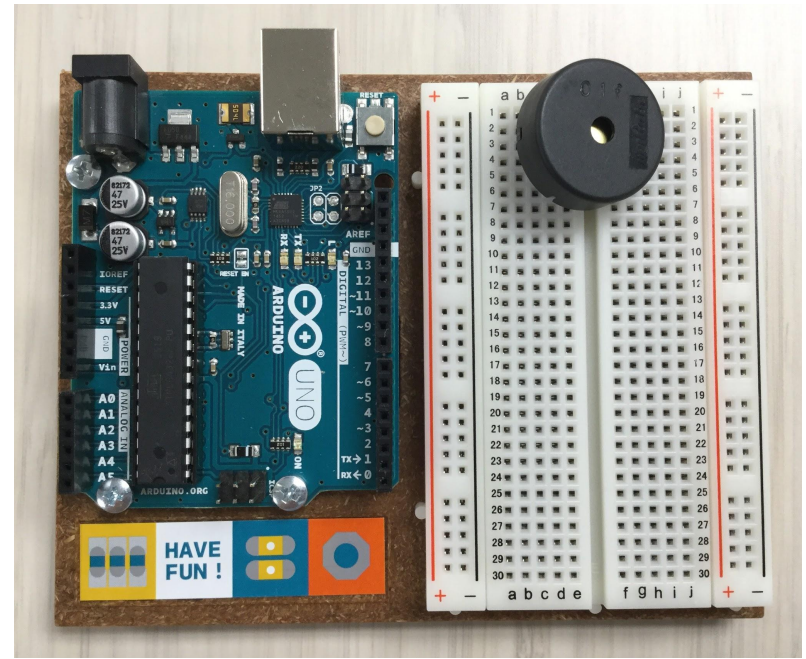
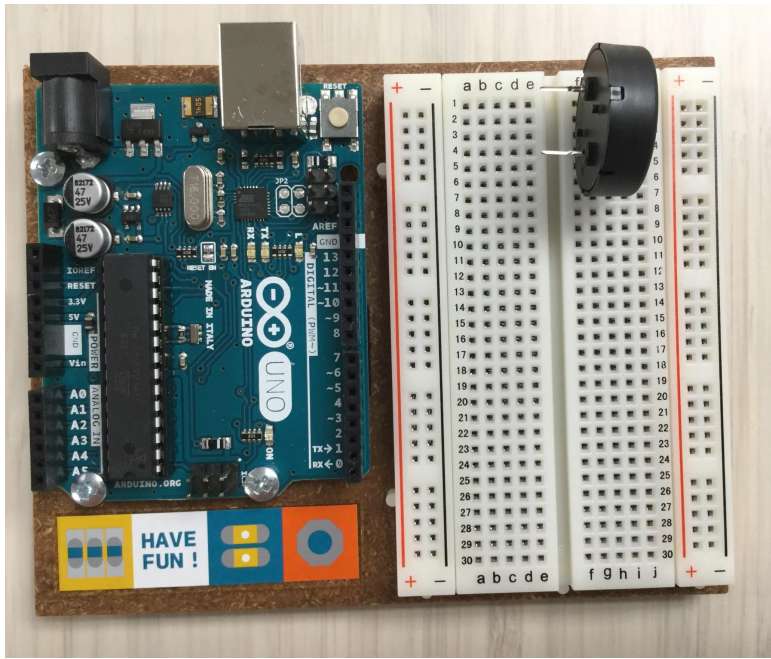
A *Split* runs down the middle dividing the board in two.



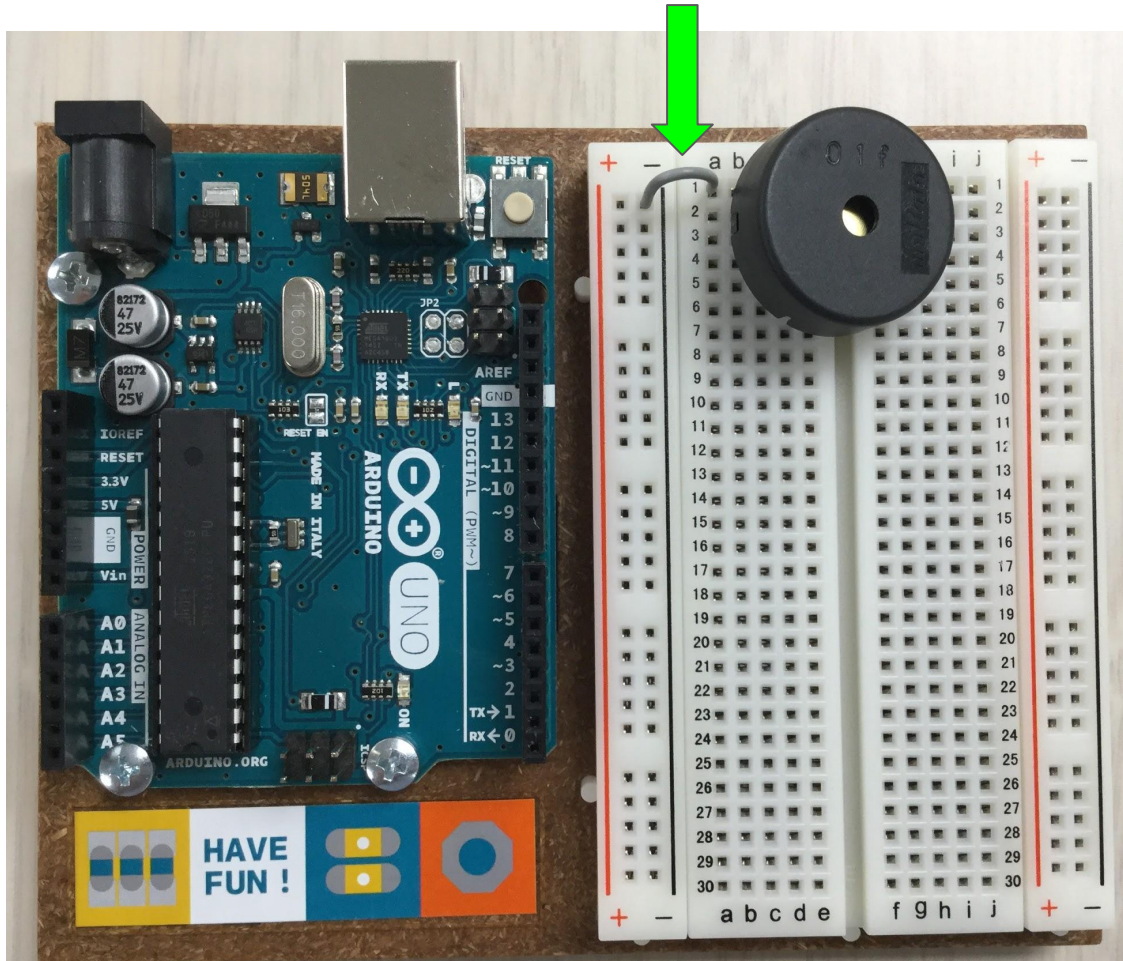
# Connect the Piezo

piezo is the annoying little buzzer that will give us sound!



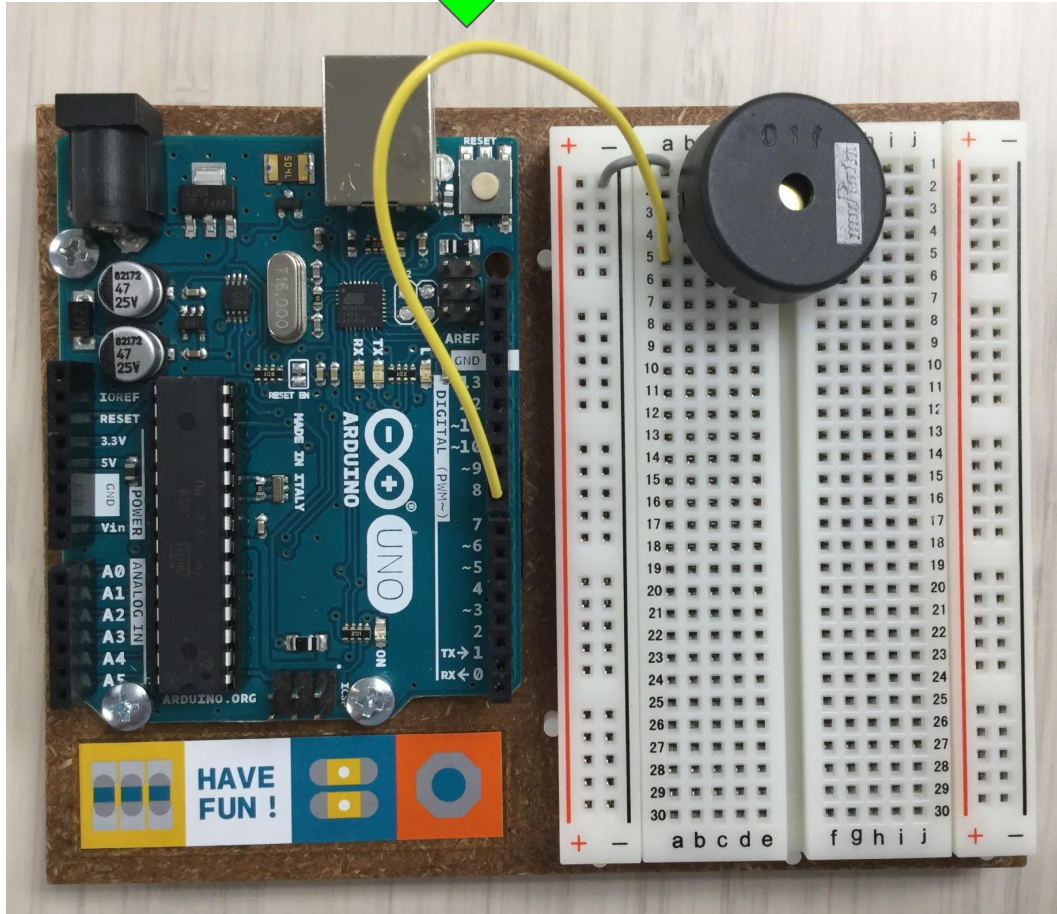


1. Gently insert the pins of your **Piezo** into row one and five.



2. Connect row one to the **Ground (-)** rail with a wire jumper. (i.e. piezo to GND)



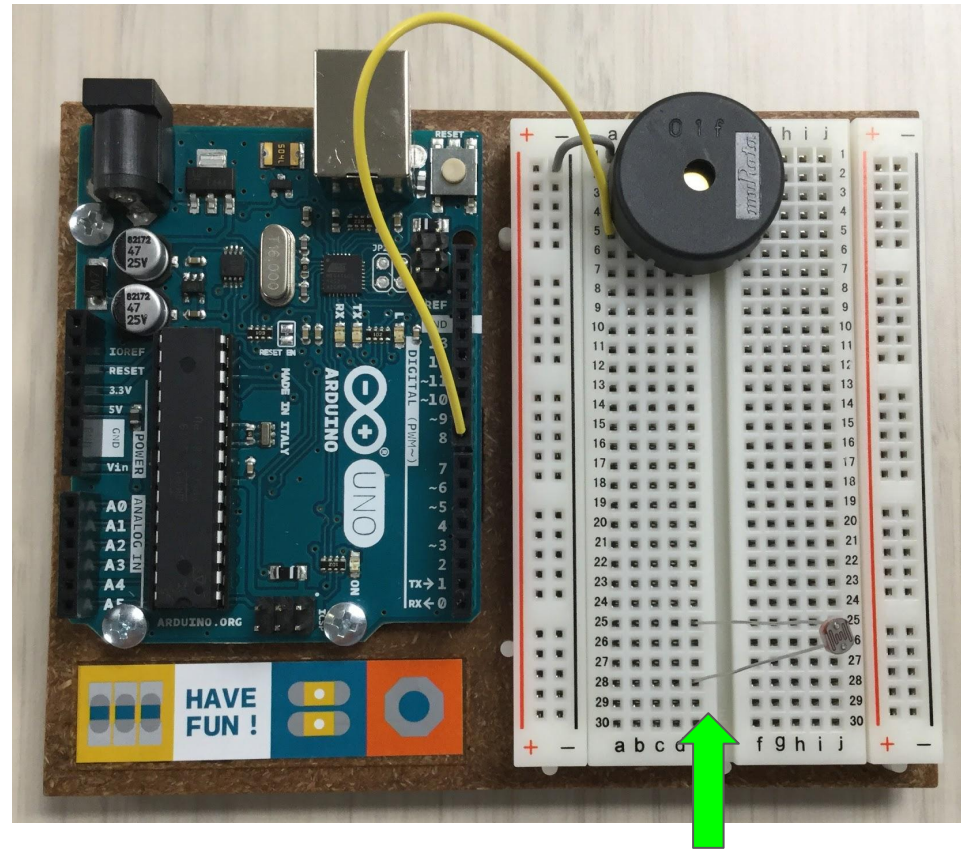


3. Connect row five to **Pin 8** on the UNO using a wire jumper.  
(i.e piezo to digital pin)

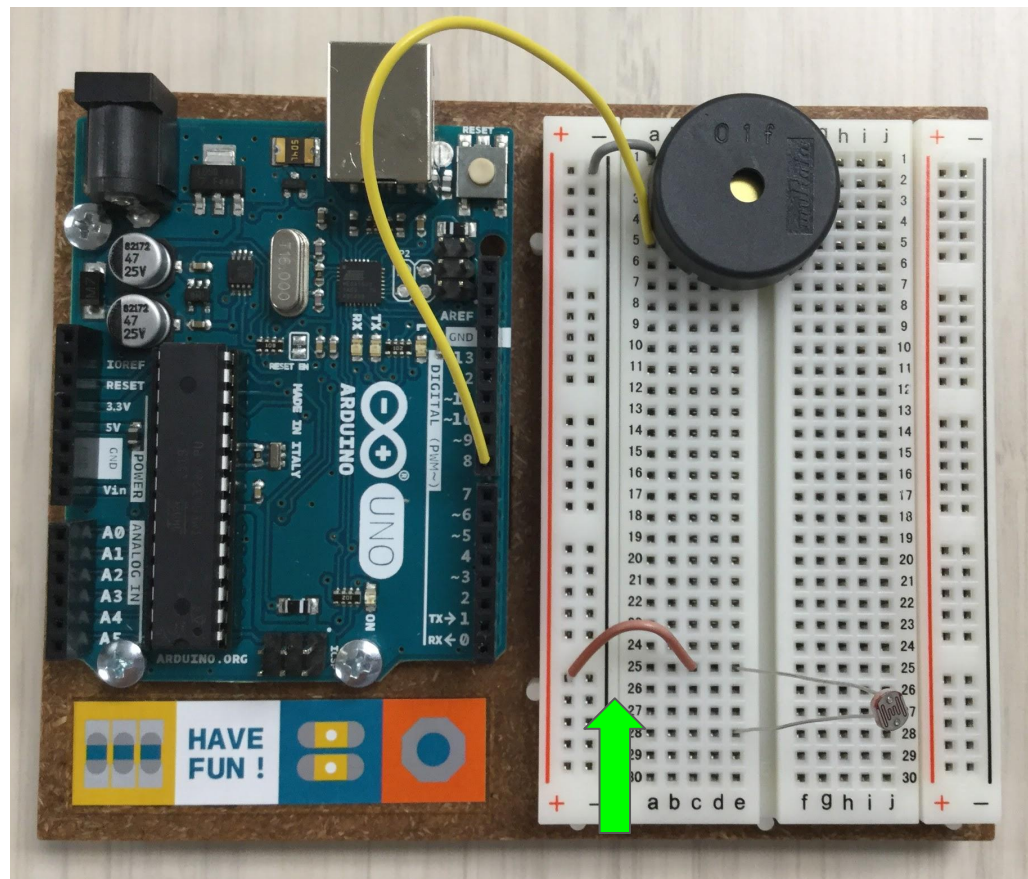
# Connect the Photoresistor

the photoresistor is the sensor that will control our musical instrument

4. Insert the pins from the photoresistor in rows twentyfive and twentyeight.

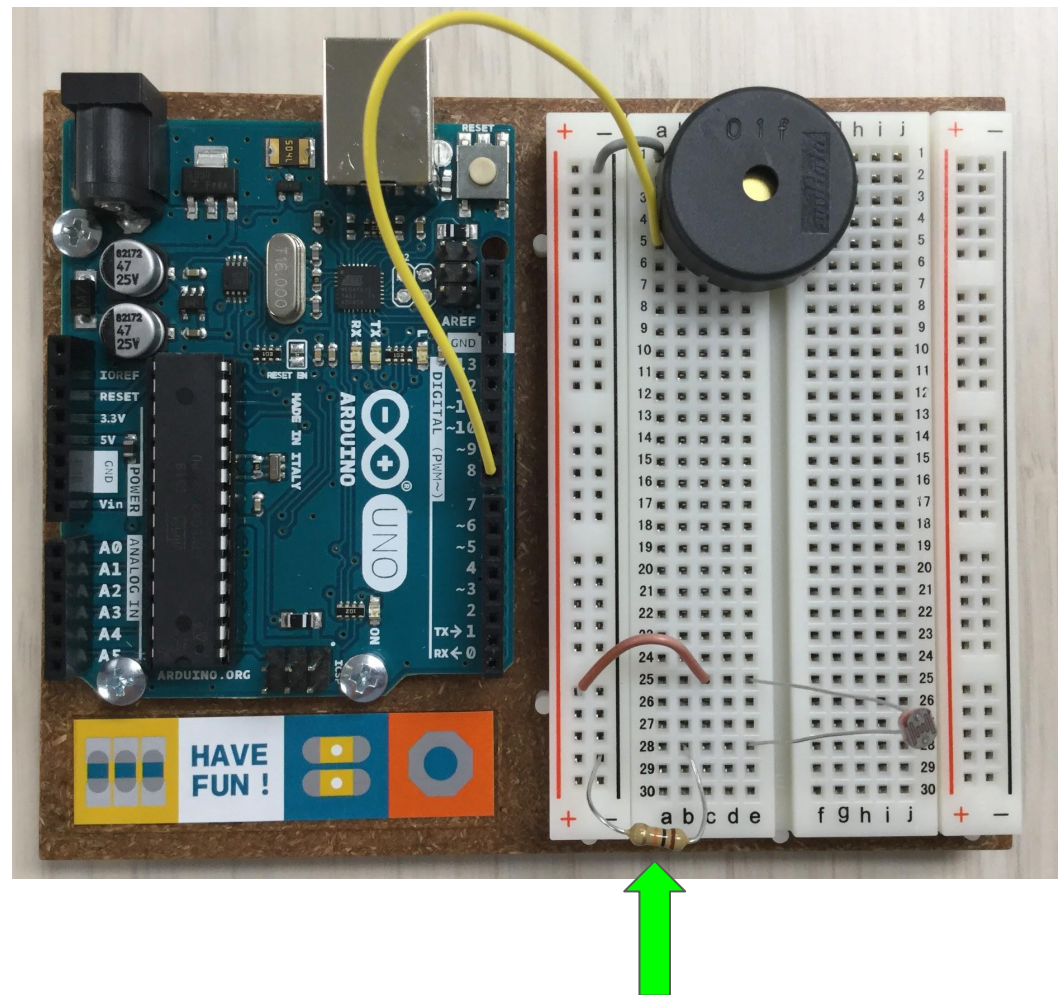


5. Connect row twentyfive to the **Power (+)** rail with a jumper wire.  
(i.e. photoresistor to 5V)

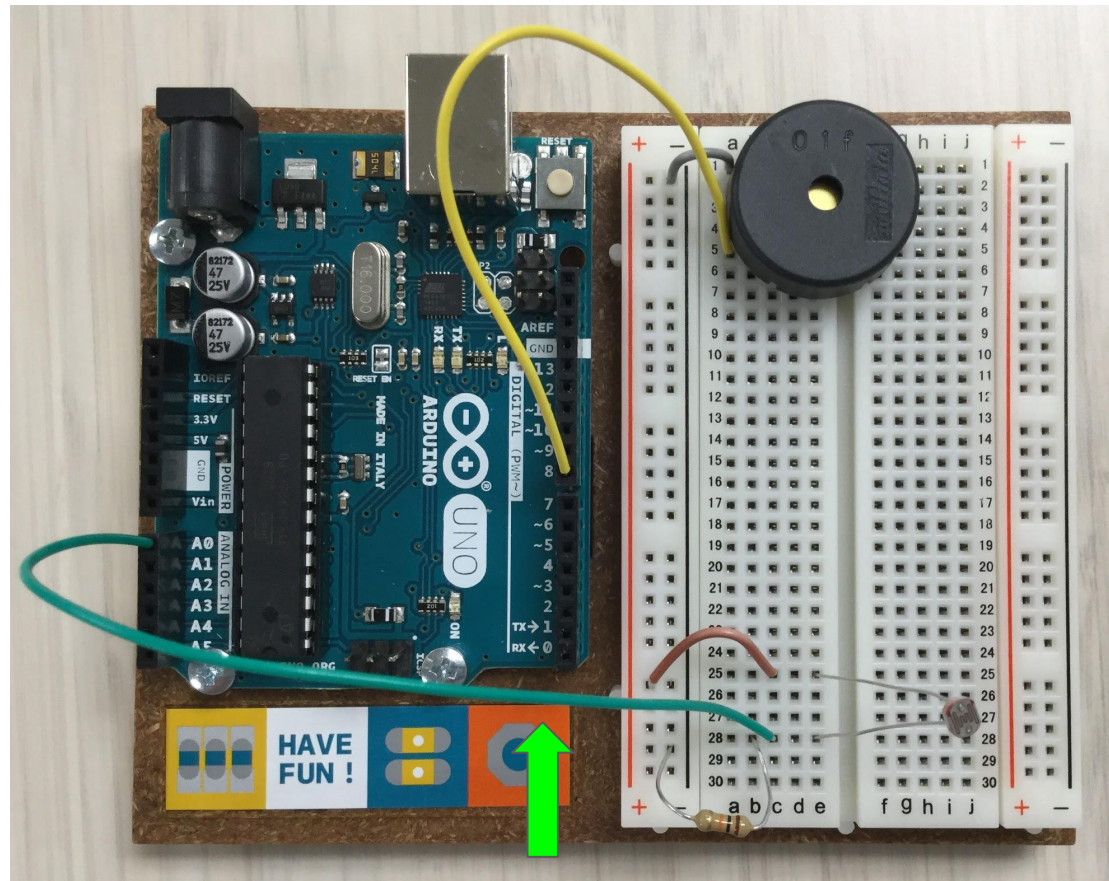




6. Use **10k resistor** to connect row twentyeight to **Ground (-)** rail.  
(i.e. photoresistor to GND)



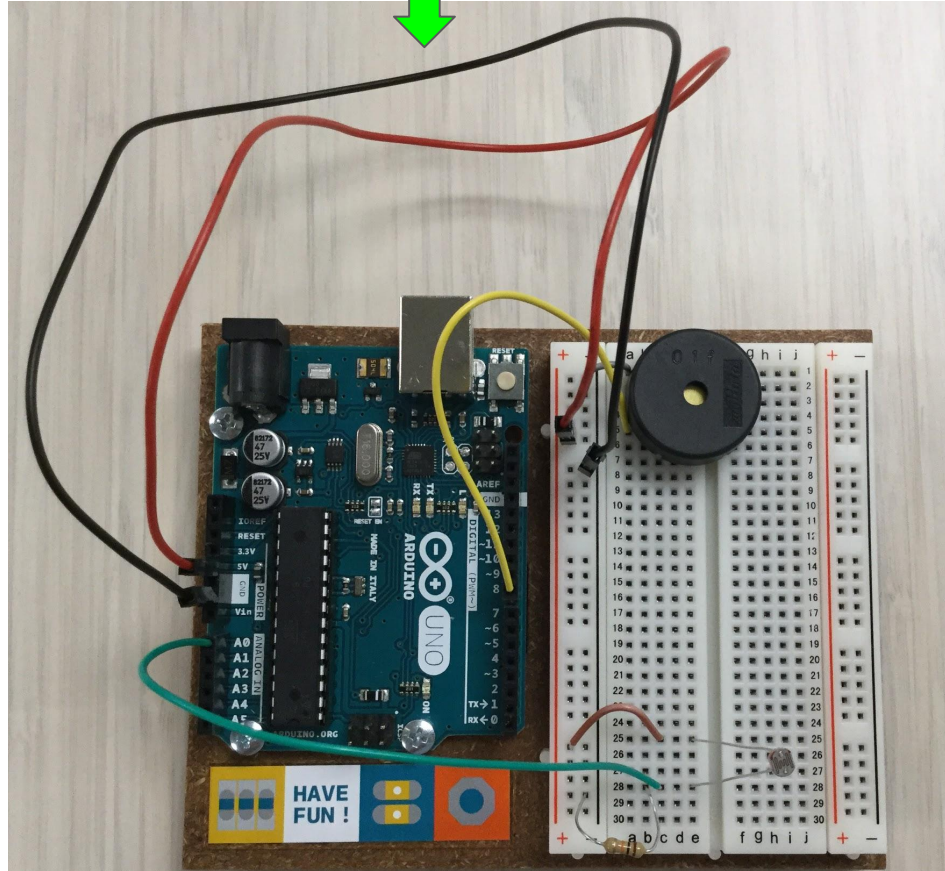
7. Connect row twentyeight to **Pin A0** on the UNO.  
(i.e. photoresistor to analog input)



# Connect the Power

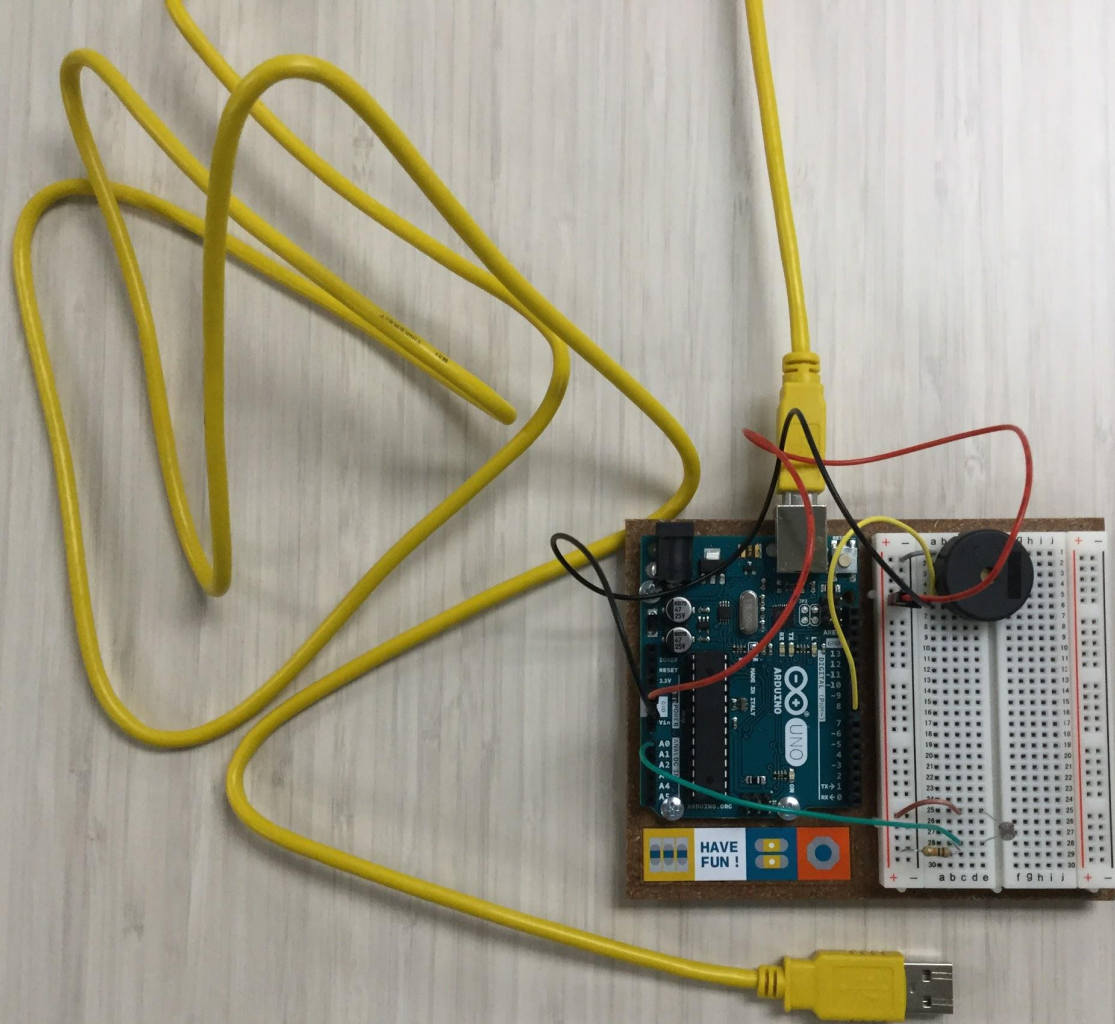
the circuit needs electricity, 5V is provided by the UNO

**9.** Connect the **Ground (-)** rail to a **GND pin** on the UNO with a power jumper wire.



**10.** Connect the **Power (+)** rail to the **5V Pin** on the UNO with a power jumper wire.





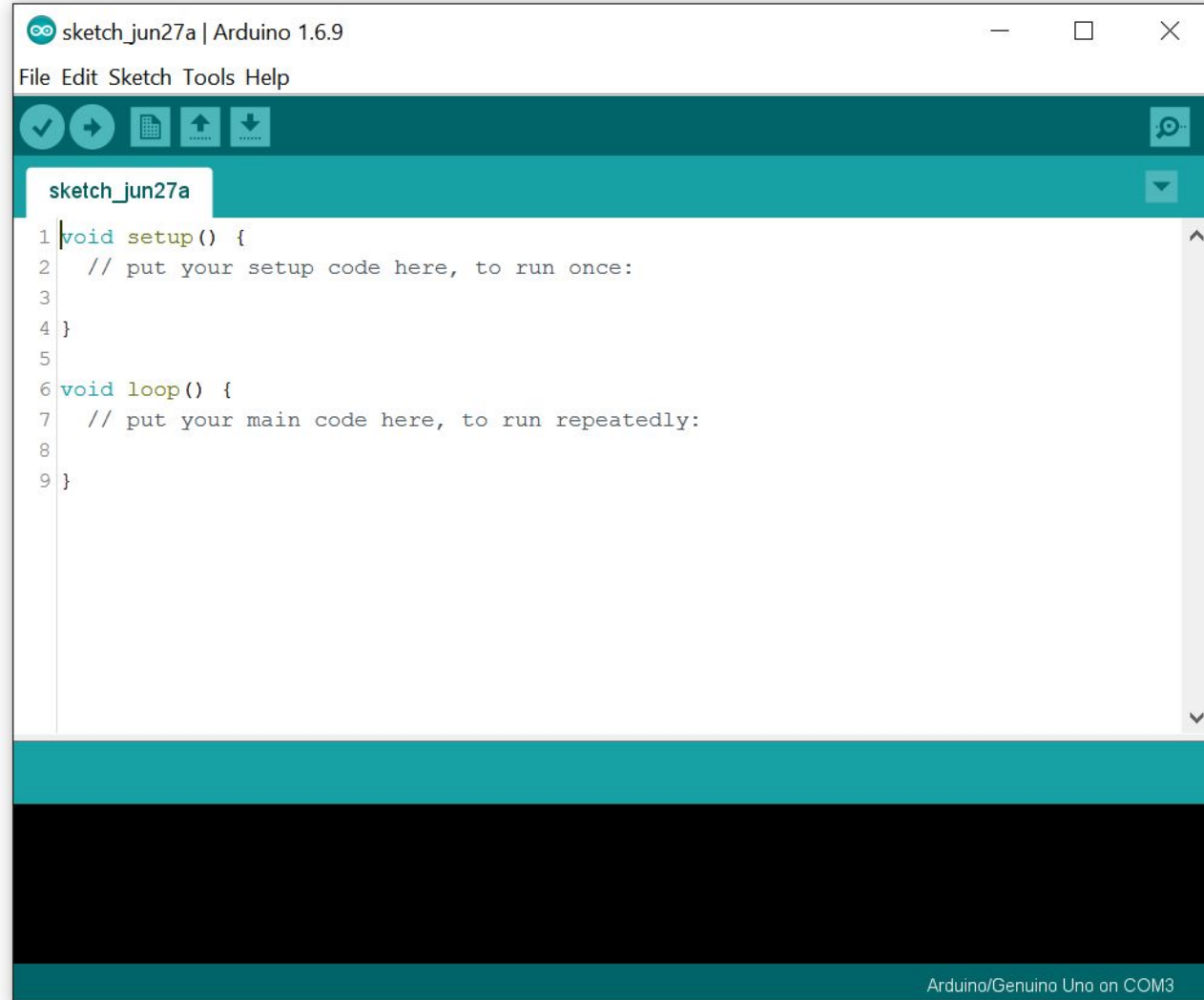
**Circuit  
complete!**

# Load the Sketch

UNO is programmed in the Arduino language,  
<https://www.arduino.cc/en/Reference/HomePage>  
and loaded to the board using the Arduino IDE,  
<https://www.arduino.cc/en/Main/Software>

# 1. Open the Arduino Software (IDE).

IDE = Integrated Development Environment = an application that allows you to edit/write code, compile it, and send it to Arduino devices.

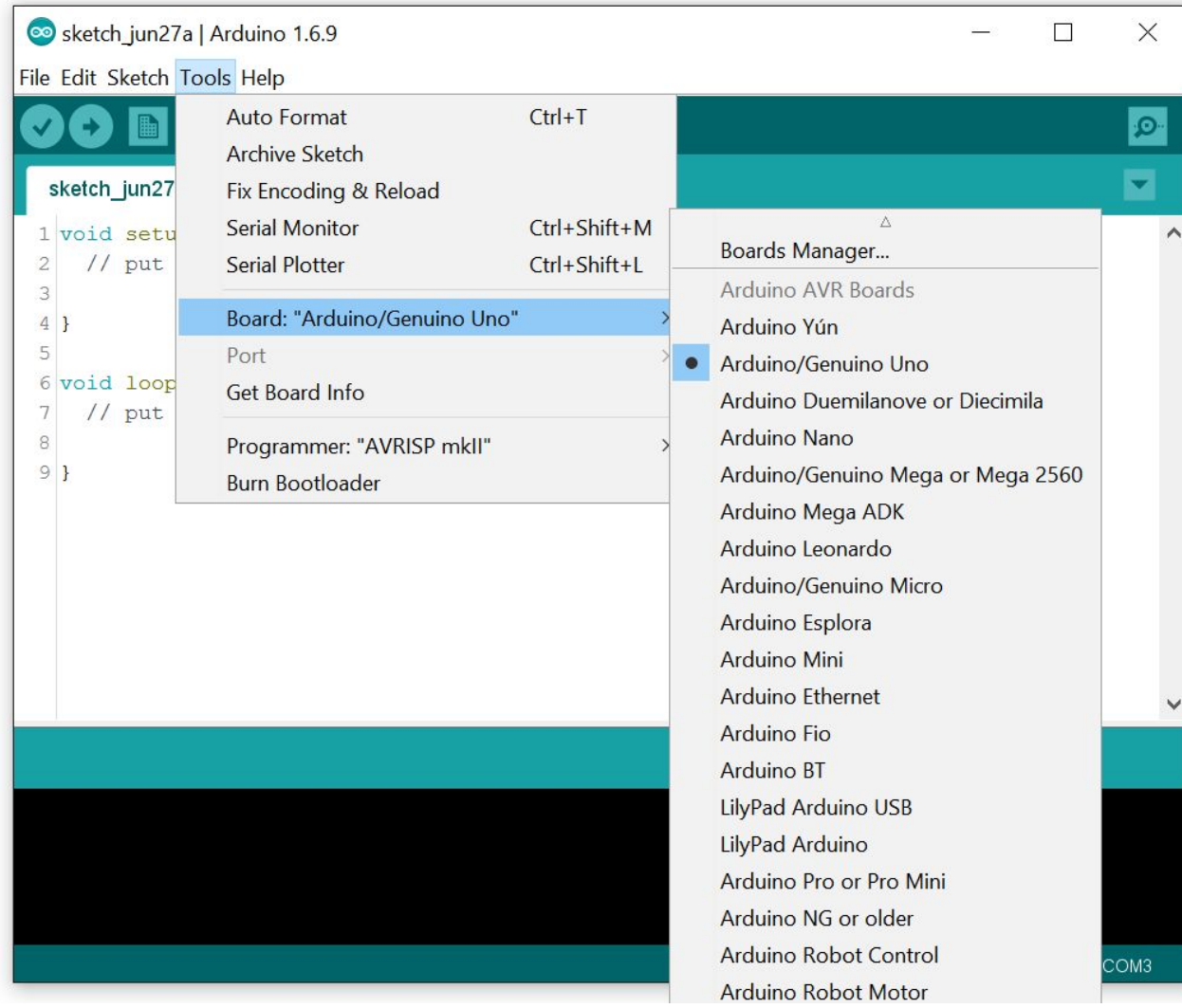


## 2. Connect your UNO.

Connect the UNO to your computer using the USB cable. Some LEDs on the board should light up, and the board should automatically be detected.

To make sure the IDE knows which board to use, in the Tools menu:

- under Board, select Arduino/Genuino Uno
- under Port, select the one that says "Uno"

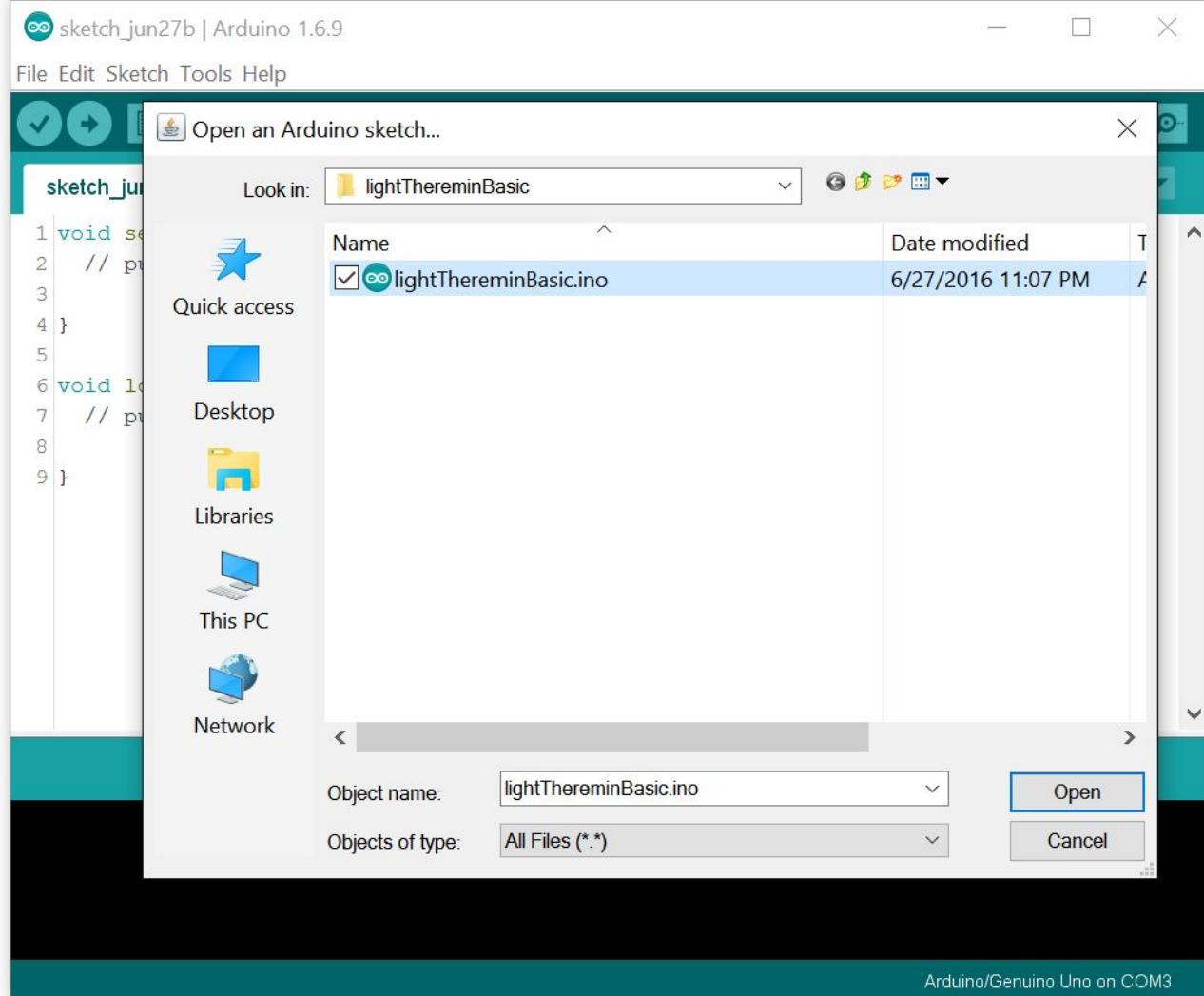




### 3. Open the sketch.

Sketches are programs written using the Arduino IDE. Each is saved in its own folder and has the extension *.ino*.

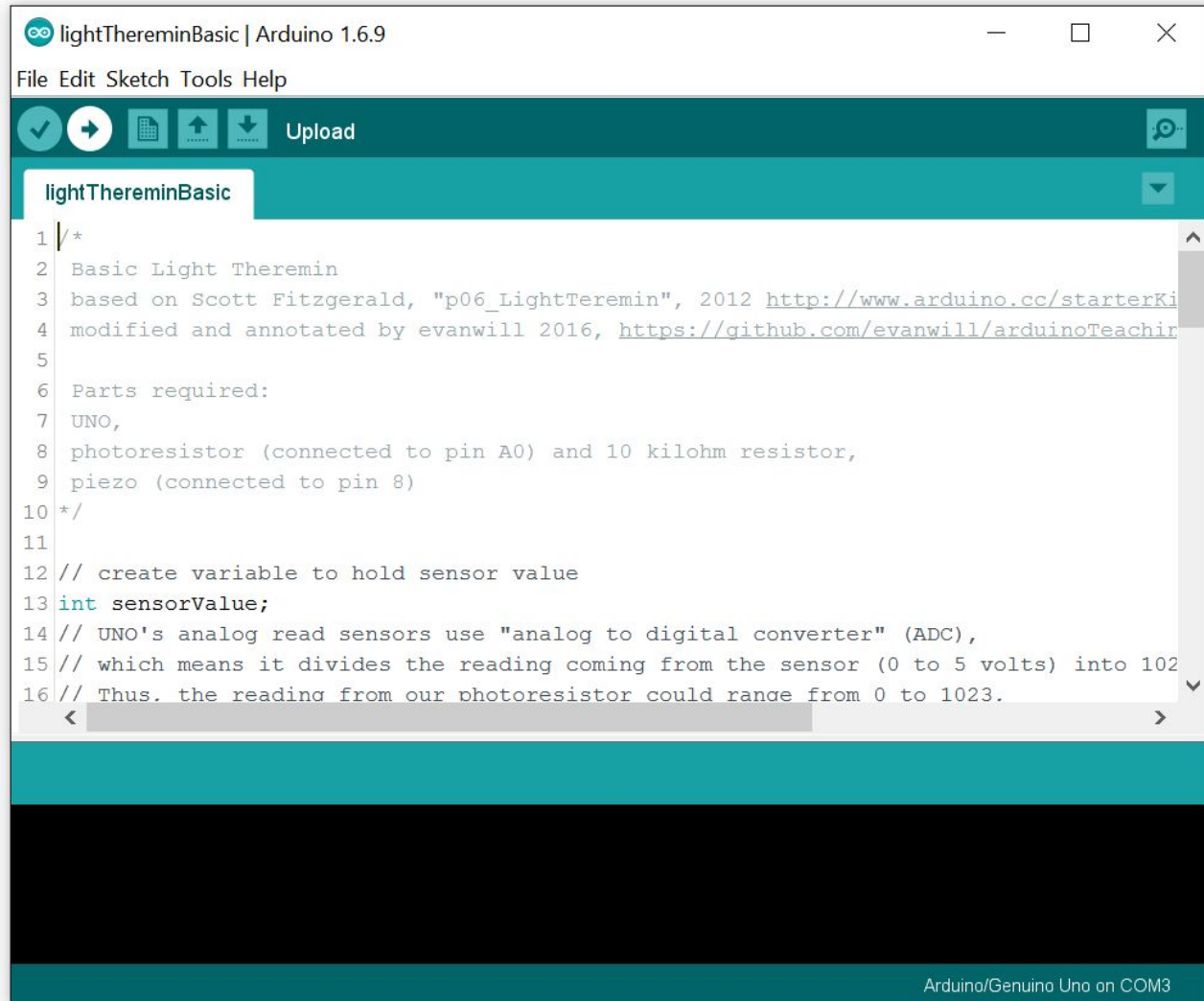
Choose File then Open to navigate to the sketch you want to load, for example "lightThereminBasic.ino".



## 4. Upload the sketch.

Click the Upload button (a right arrow) and the IDE will compile the code, then send it to the connected Arduino board.

If there are any errors, a message will appear in the terminal at the bottom.



The screenshot shows the Arduino IDE interface. The title bar reads "lightThereminBasic | Arduino 1.6.9". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar contains a checkmark, a right-pointing arrow (the "Upload" button), a grid icon, an upload icon, and a download icon. The main editor area displays the sketch code for "lightThereminBasic". The code includes a multi-line comment, a parts list, and the start of a C++ program with variable declarations and comments.

```
1 /*  
2 Basic Light Theremin  
3 based on Scott Fitzgerald, "p06_LightTeremin", 2012 http://www.arduino.cc/starterKi  
4 modified and annotated by evanwill 2016, https://github.com/evanwill/arduinoTeachir  
5  
6 Parts required:  
7 UNO,  
8 photoresistor (connected to pin A0) and 10 kilohm resistor,  
9 piezo (connected to pin 8)  
10 */  
11  
12 // create variable to hold sensor value  
13 int sensorValue;  
14 // UNO's analog read sensors use "analog to digital converter" (ADC),  
15 // which means it divides the reading coming from the sensor (0 to 5 volts) into 102  
16 // Thus. the reading from our photoresistor could range from 0 to 1023.
```

At the bottom of the IDE, the text "Arduino/Genuino Uno on COM3" is visible.

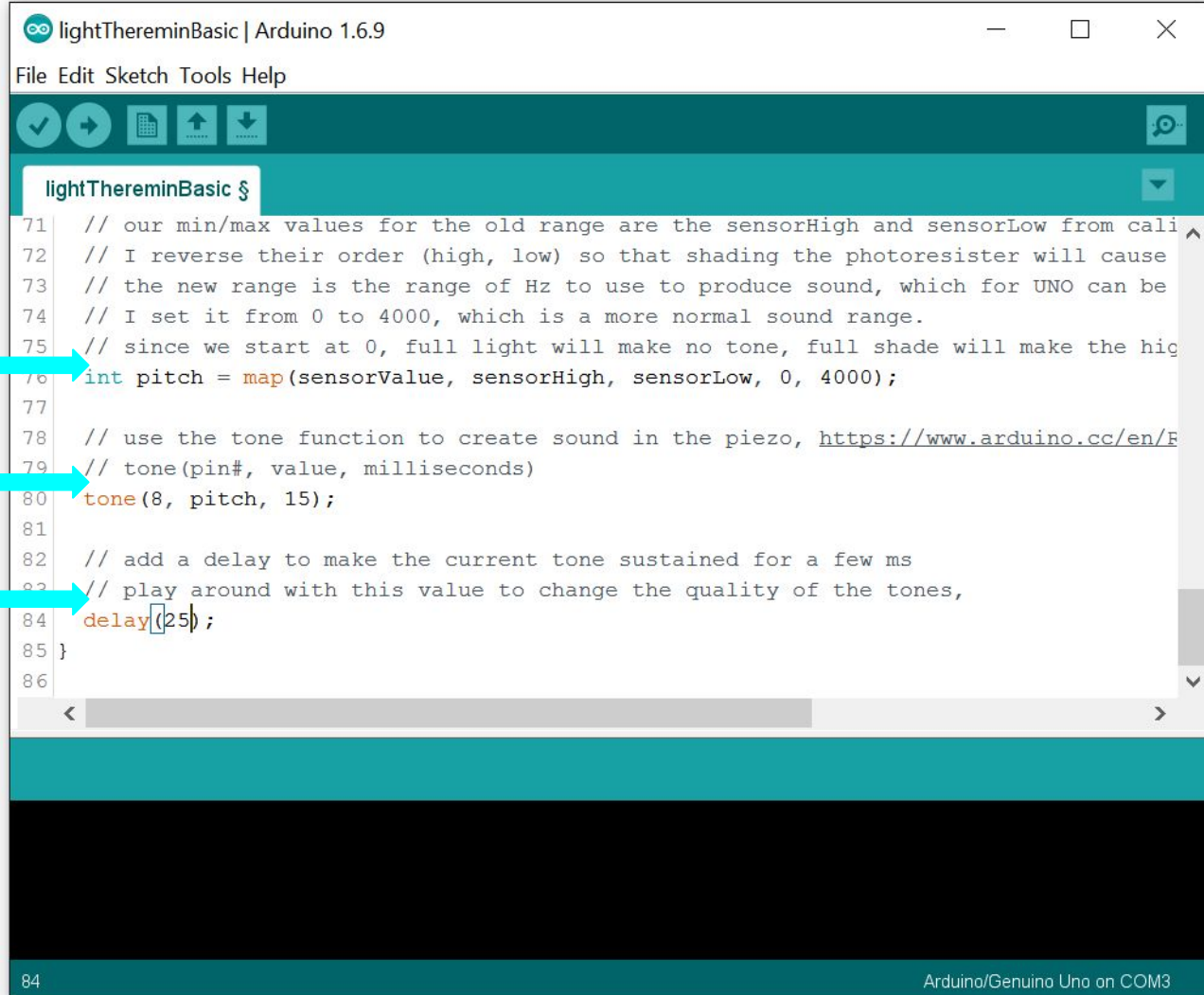
## 5. Edit!

In the text editor window, try changing a few things at the bottom of the loop, such as:

- range of pitch
- tone length
- loop delay

There are lots of comments in the code (marked by //) to explain what each line is doing.

Try loading one of the other lightTheremin examples!  
(they all work with the same circuit setup)



```
lightThereminBasic $
71 // our min/max values for the old range are the sensorHigh and sensorLow from cali
72 // I reverse their order (high, low) so that shading the photoresister will cause
73 // the new range is the range of Hz to use to produce sound, which for UNO can be
74 // I set it from 0 to 4000, which is a more normal sound range.
75 // since we start at 0, full light will make no tone, full shade will make the hig
76 int pitch = map(sensorValue, sensorHigh, sensorLow, 0, 4000);
77
78 // use the tone function to create sound in the piezo, https://www.arduino.cc/en/F
79 // tone(pin#, value, milliseconds)
80 tone(8, pitch, 15);
81
82 // add a delay to make the current tone sustained for a few ms
83 // play around with this value to change the quality of the tones,
84 delay(25);
85 }
86
```

84

Arduino/Genuino Uno on COM3

# Arduino Resources

Official Arduino site, <https://www.arduino.cc/>

- get Arduino IDE.
- purchase official kits.
- learning and reference materials.
- watch Arduino Starter Kit videos, [https://www.youtube.com/playlist?list=PLT6rF\\_I5kknPf2qIVFlvH47qHvqvzkknd](https://www.youtube.com/playlist?list=PLT6rF_I5kknPf2qIVFlvH47qHvqvzkknd)

Codebender, <https://codebender.cc/>

- online Arduino code editor that works with Chromebook / Android

16 Hertz, <http://www.16hertz.com/>

- cheaper non-official kits.
- great graphic novel style intro book, [bit.ly/16hzguide](http://bit.ly/16hzguide)

Adafruit, <https://www.adafruit.com/>

- good source for purchasing electronics, including specialized Arduino based boards and accessories.
- lots of projects and learning resources.

Sketches used in this workshop, <https://github.com/evanwill/arduinoTeaching>