IoT
Internet of Things

STEM
SCIENCE TECHNOLOGY
ENGINEERING MATHEMATICS

OHIO UNIVERSITY
1804
Voinovich School of Leadership and Public Affairs

AMERICAN ELECTRIC POWER FOUNDATION

PORTSFUTURE
IMAGINING THE OPPORTUNITIES, GATHERING YOUR IDEAS
THE FACILITY AT PIKETON, OHIO

Internet of Things Student STEM Project
Jackson High School

Lesson 2: Arduino and LED

Lesson 2: Arduino and LED

Time to complete Lesson

60-minute class period

Learning objectives

- Students learn about Arduino and other microcontrollers and the functionalities they provide.
- Students learn about the Arduino IDE installation, configuration, and basic programming.
- Students get hands on experience with the microcontroller to further examine how they interface with various sensors and modules
- Students learn about how to use and control LEDs
- Students learn how to sound reactive led strips working with Arduino Uno and sound sensor for music

21st century technical skills gained through this activity

- Electronic Circuit Design - Microcontrollers
- Computer Programming
- LED sensor
- Sound sensor

Credits!

Code and tutorial based on the great open source lib & example code at:
https://github.com/hansjny/Natural-Nerd/tree/master/arduino/soundsread2
http://fastled.io/

Introduction

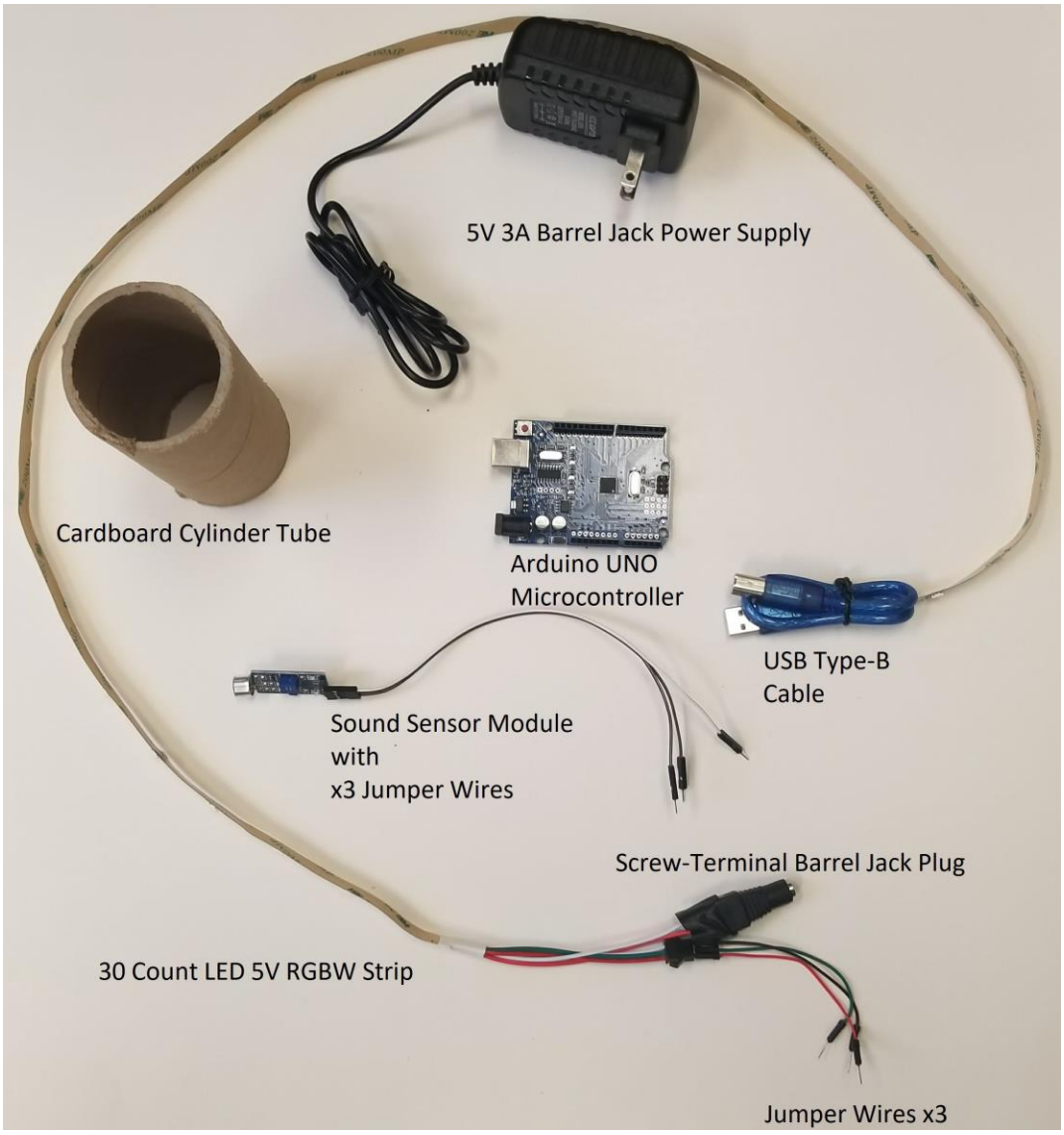
In this lesson, learn how to flash an LED strip to the beat of music using a small sound sensor, transistor and an external power source. This lesson will show you how to use open source libraries for Arduino. We will use the FastLED library for programming addressable LED strips and pixels such as WS2810, WS2811, and WS2812B. It will be reacting on the surrounding sound and music and will be playing like a visualizer.

In this build we'll make a good looking light that dances to all sounds and music, using simple components and some basic Arduino programming.

Project 1: LED Strip

Step 1: Parts List

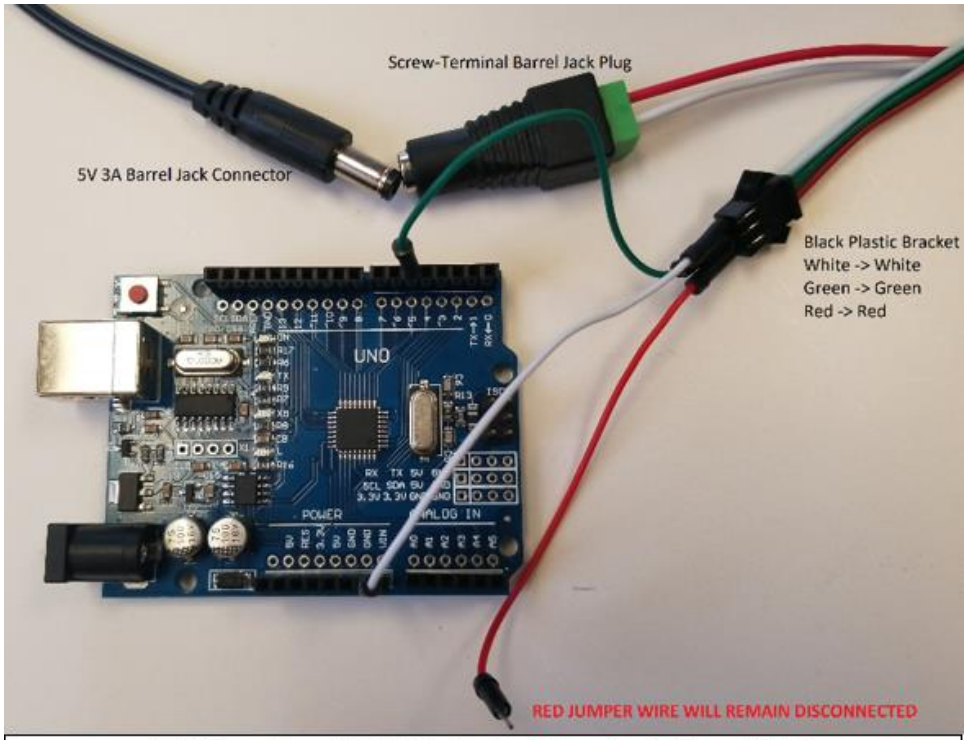
Below you will see the parts that are included in your project kit. Lay these parts out in your workspace and ensure that all pieces are present!



Step 2: Assembly

First, take the brown film off the back of the LED strip and attach it to the cylinder in a spiral shape as shown. Do this by sticking the end of the strip with the connections to the bottom of the cylinder, then rolling the cylinder and sticking the strip higher and higher until you reach the top of the cylinder and run out of LED strip.

The large blue board in our kit is called the Arduino UNO. The UNO is one of many Arduino microcontrollers, all of which can carry out numerous functionalities. Now, we will connect our Arduino with the LED strip and apply power!



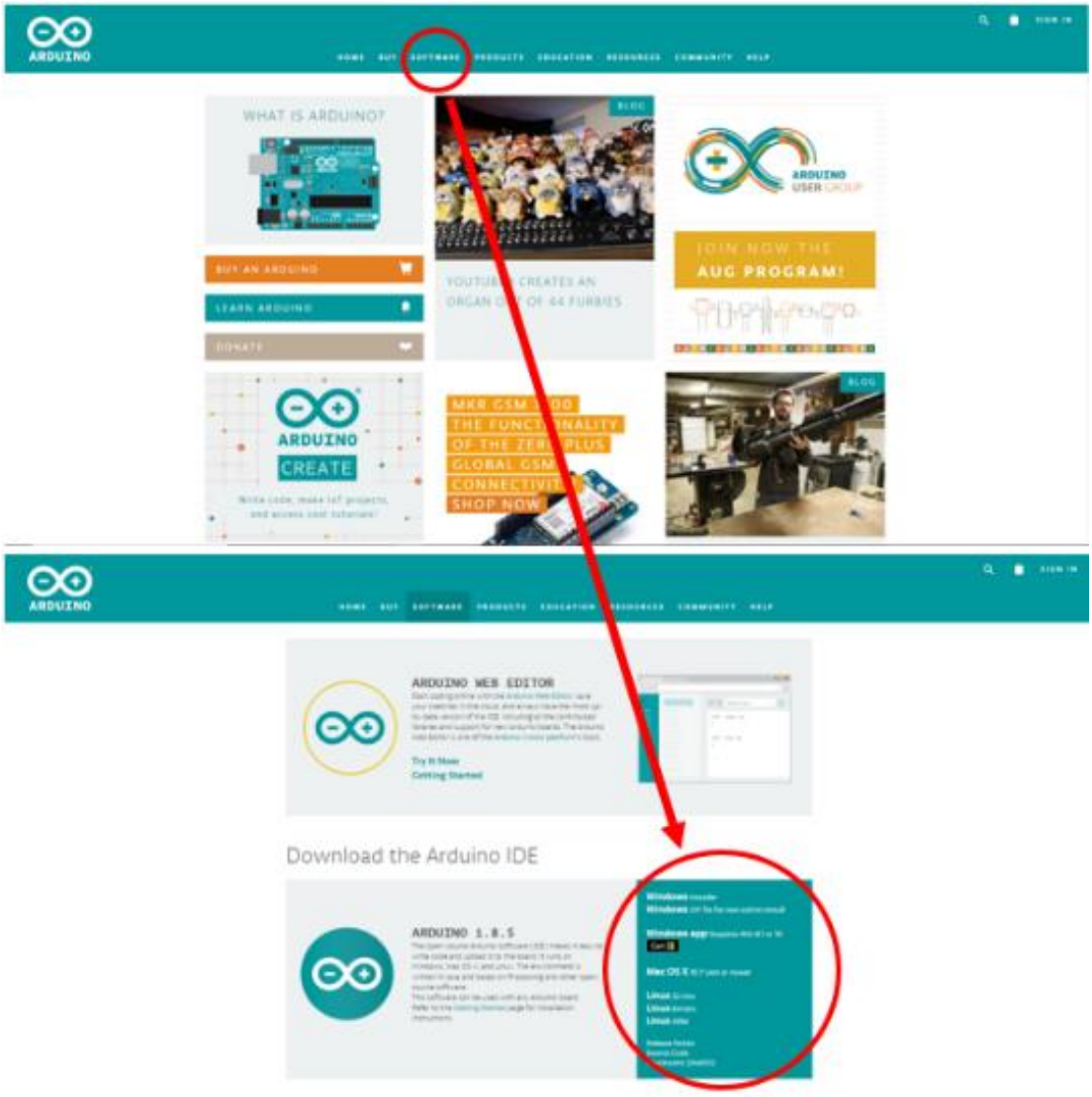
Wiring the LED Strip to the Arduino UNO	
Red (Disconnect for Now)	VIN
Green	DIGITAL PWM (~) 5
White	GND (Ground)

Notice the three jumper wires coming off the LED strip. Follow the wires back behind the black plastic joint, and you will notice red, green, and white cables. Using those colors, follow the above diagram and table to wire the LED strip to the Arduino.

Do not connect the red jumper wire to VIN until Steps 3 and 4 are complete!

Step 3: The Code

Your next task is done on the computer. First, grab the file that contains our code! You will be instructed on where to go to get this file. Next, open up your web browser and go to https://www.arduino.cc. From here, click on the “Software tab” at the top of the screen

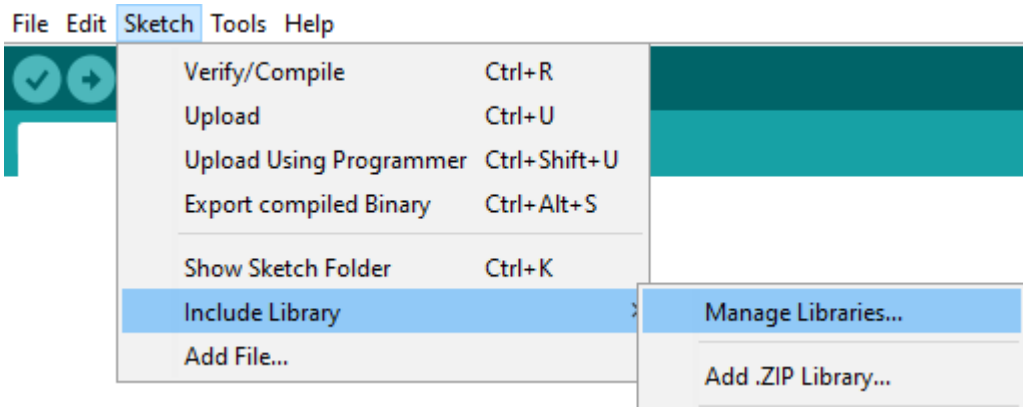


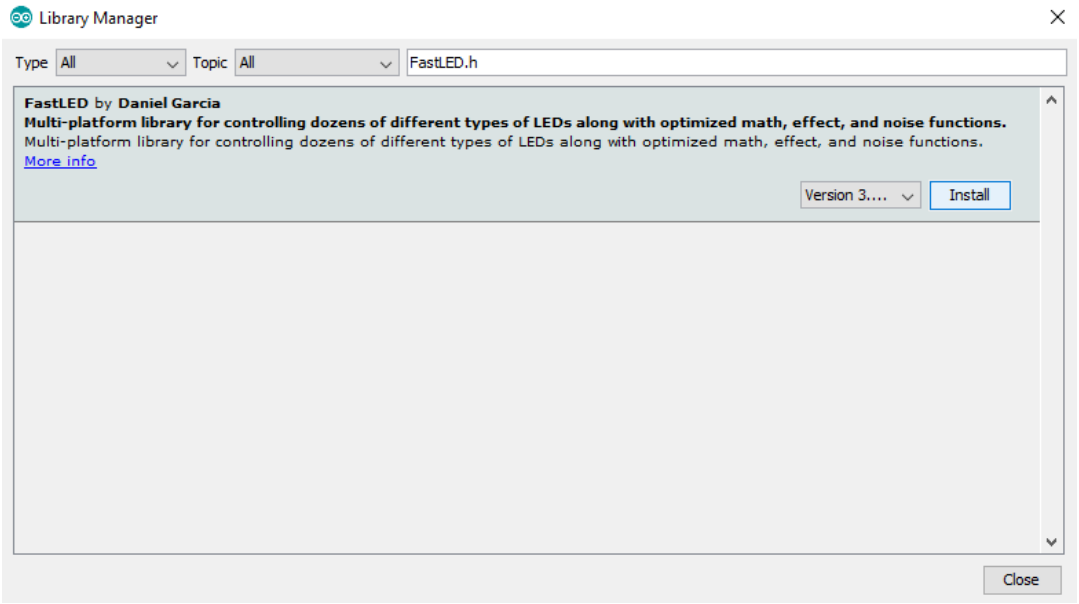
You will be brought to the download screen, where you will pick your preferred download for your system and install! Once the Arduino IDE (Integrated Developmental Environment) is installed on your system, load it up!

Once you have the IDE running, go to file on the top tab, and click open (or simply press Ctrl + O.) Open up the code file you downloaded earlier, and you will notice the first line reads

#include "FastLED.h"

FastLED.h is a library, or a file that provides our code with an extra functionality, in this case, more control over our particular set of LED's. While we can see the library in our code, we need to install the library for our code to run properly. Click on Sketch on the toolbar at the top of the IDE, go to Include Library, and then click Manage Libraries.

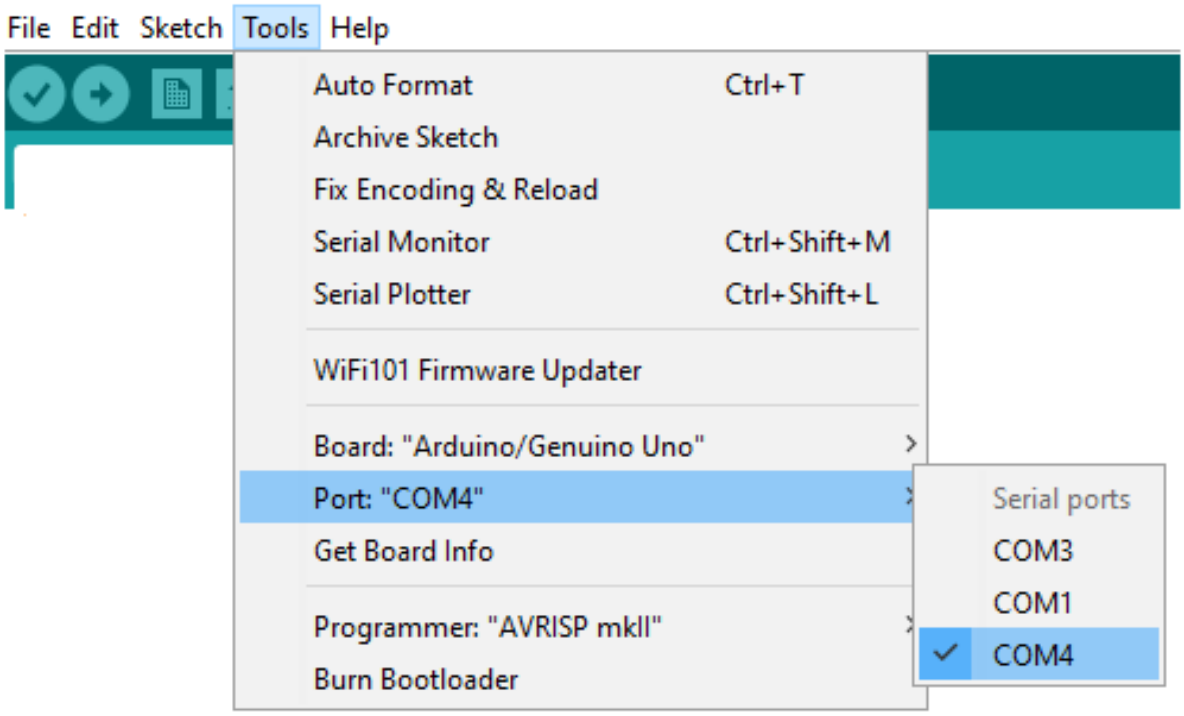




When the Library Manager appears, search for "FastLED.h" in the search bar. **FastLED** by **Daniel Garcia** should appear. Click on the library and you will see the install button. Click the install button and you will install the latest library.

Step 4: Running the Program

Now we are ready to run our first program! First, use the USB cable to connect the Arduino to your computer. Next, you need to select a COM port so that the IDE knows where to upload the code. Go to the Tools tab on the toolbar, select the option that says Port, then click on the bottom port on the Serial ports list. This can be a port of any number. For this example, the port was COM4.



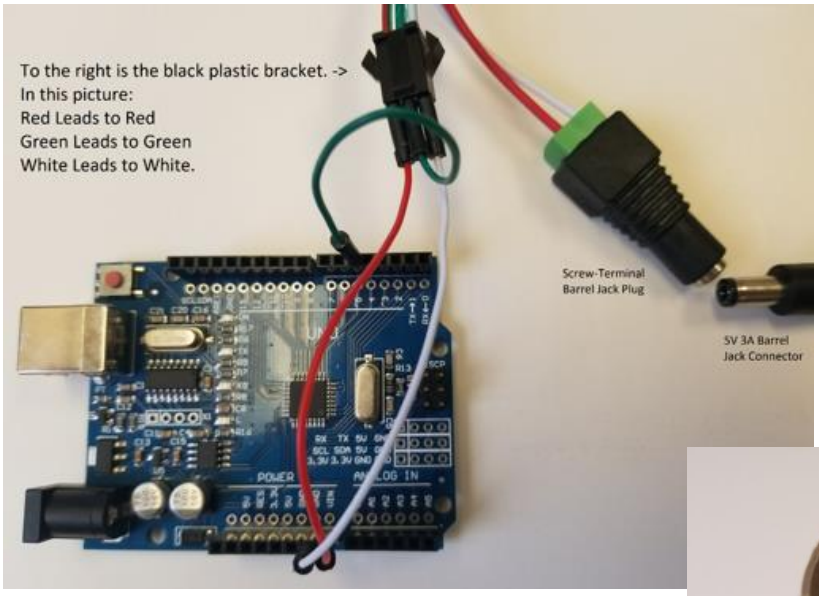
Let's test the upload! Click on the arrow pointing right in the top left of the IDE. This will compile our code and attempt to upload it to the Arduino. If the upload takes a long time and/or fails, try to switch the COM port to one of the others on the list. When the correct COM port is selected, the code will compile and upload to the Arduino board. The bottom green bar of the IDE will say "Done Uploading" once upload is completed. Unplug the USB cable from the Arduino, but keep the cable itself in the computer (we will need it for the next projects.)

Step 5: Putting it all Together

Lastly, we have to apply power so that our Arduino can process the code and our LEDs can light up!

You can now connect the red wire to VIN from Step 2

Now find a power outlet, and connect the 5V 3A Barrel Jack Power Supply to the power outlet. If you were successful, you will see the LEDs change color as they move up and down the cylinder.



Project 2: Sound-Sensor Module

Step 1: The Code | Setup and Run

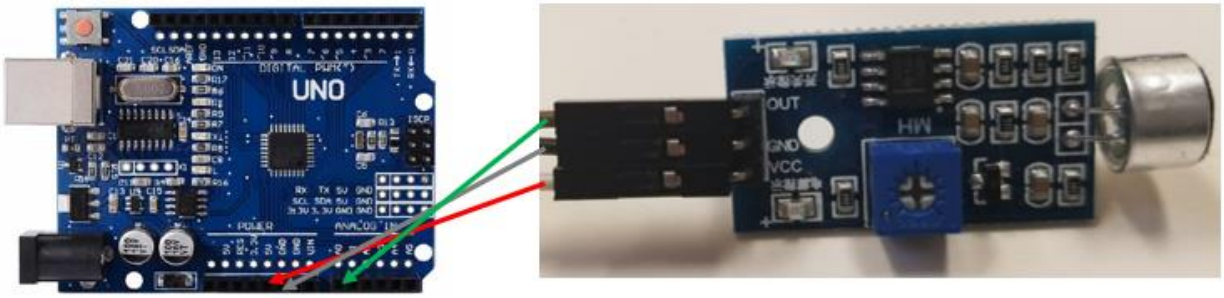
Since we have the Arduino IDE set up and running, our next project will require some altered code to be put into the IDE. You will be instructed once again on where you can get the file containing the code. Once you have the file, open it in the IDE following the same method as in Step 2 of Project 1. Upon opening the code, you will see that FastLED.h is once again included as a library. We already installed this library in the last project, so there is no need to reinstall. Your COM port should remain as it was.

Step 2: Wiring

With Project 1 complete, we already have most of the parts needed for Project 2 in place! For now, remove the power supply and the red jumper wire connecting the LED strip power to the microcontroller. Now plug the Arduino back into your computer! We are going to upload our new code to the board, so click the arrow pointing right in the top left of the IDE, and make sure that all of our code compiles and is sent over to the Arduino successfully.

Now unplug the Arduino from the computer and rewire the red jumper wire, ensuring the correct connection of the LED strip (reference Project 1 if you need help placing the cables back in their correct spots.) Now we will make our addition to the first project's setup.

This is the sound-sensor module. In this project, the component that will be able to pick up sound and send it to the Arduino, where this sound will be (in a way) converted to light on the LEDs. You will notice three jumper wires coming off of the module, these will be connected to the Arduino using the labels on the module itself.



Wiring the Sound-Sensor Module to the Arduino UNO	
OUT	Analog In A0
GND	GND
VCC	5V

Using the above diagram and chart, add the sound-sensor module to your Arduino. Once everything is wired up, re-plug the power supply to the Screw-Terminal Barrel Jack Plug on the LED strip. If you were

successful, you should be able to notice that with more sound, the more LEDs light up. To adjust the sensitivity, obtain a small screwdriver and insert it into the cross on the blue box of the sound-sensor module. Now turn clockwise or counter clockwise to change how sensitive the module reacts to sound.

Lesson 2: Modifications!

If you are finished with the projects before the time is up, feel free to browse this section for help modifying the codes to add a bit of personal touch to your projects!

Project 1:

- LEDS.setBrightness(125);
    - This line of code pertains to the brightness of the LEDs. The minimum brightness is 0, and the maximum brightness is 255. It is not recommended to keep the LEDs on 255 for a long period of time. Change the numbers to whatever you would like to see changes in the brightness!

Project 2:

- float fade_scale = 1.2;
    - This line of code refers to the speed that the LEDs fade to black if they are not relit. Make some small changes to this value and you will notice that the LEDs will stay lit longer or fade quicker.

Jackson High School STEM IoT

LESSON 2 – MICROCONTROLLER & LED PROJECT

KIT CONTENTS/CHECKLIST

❑ [x1] **Arduino UNO Microcontroller**

- The Arduino UNO microcontroller is the primary brain of the circuit, it is programmed to tell the LEDs and Sound Sensor what to do.

❑ [x1] **USB Type-B Cable [Blue]**

- This cable is to connect the Arduino UNO microcontroller to the Computer's USB Port.

❑ [x1] **30 COUNT LED 5V RGBW Strip [White]**

- This is the LED strip that we will be lighting up and working with. It is different than cheaper 'SINGLE LED' strips in that it contains all 3 LED colors (R, G, B) so that we can combine them into different colors made from them.

❑ [x1] **Sound-Sensor Module [Small Blue Circuit Board]**

- This sound-sensor module picks up a music source that is placed against it (such as your phone's speaker) and sends that info to the LED strip to light up based on the music.

❑ [x1] **5V 3A Barrel Jack Power Supply**

- This provides the required current for the microcontroller & LED strip.

❑ [x1] **Screw-Terminal Barrel Jack Plug [Black/Green]**

- This plug-piece allows us to connect the barrel-jack power supply to the Arduino circuit to power the Arduino & LED Strip.

❑ [x6] **Breadboard Wires (Colors may vary)**

- These wires are used to connect all of the pieces of our circuit together.

❑ [x1] **Cardboard Cylinder Tube**

- This is the cardboard cylinder that the LED strip will be wrapped around.