

Interactive Art with Arduino: From Zero to One

Borui Wang

What do you want to build?

A canvas that glows warmer when you whisper secrets to it...

A sculpture that 'shivers' or vibrates when the room gets too cold...

Inspiration: Let's see what others made!



Image courtesy: Instructables



Fiber Optic Dress

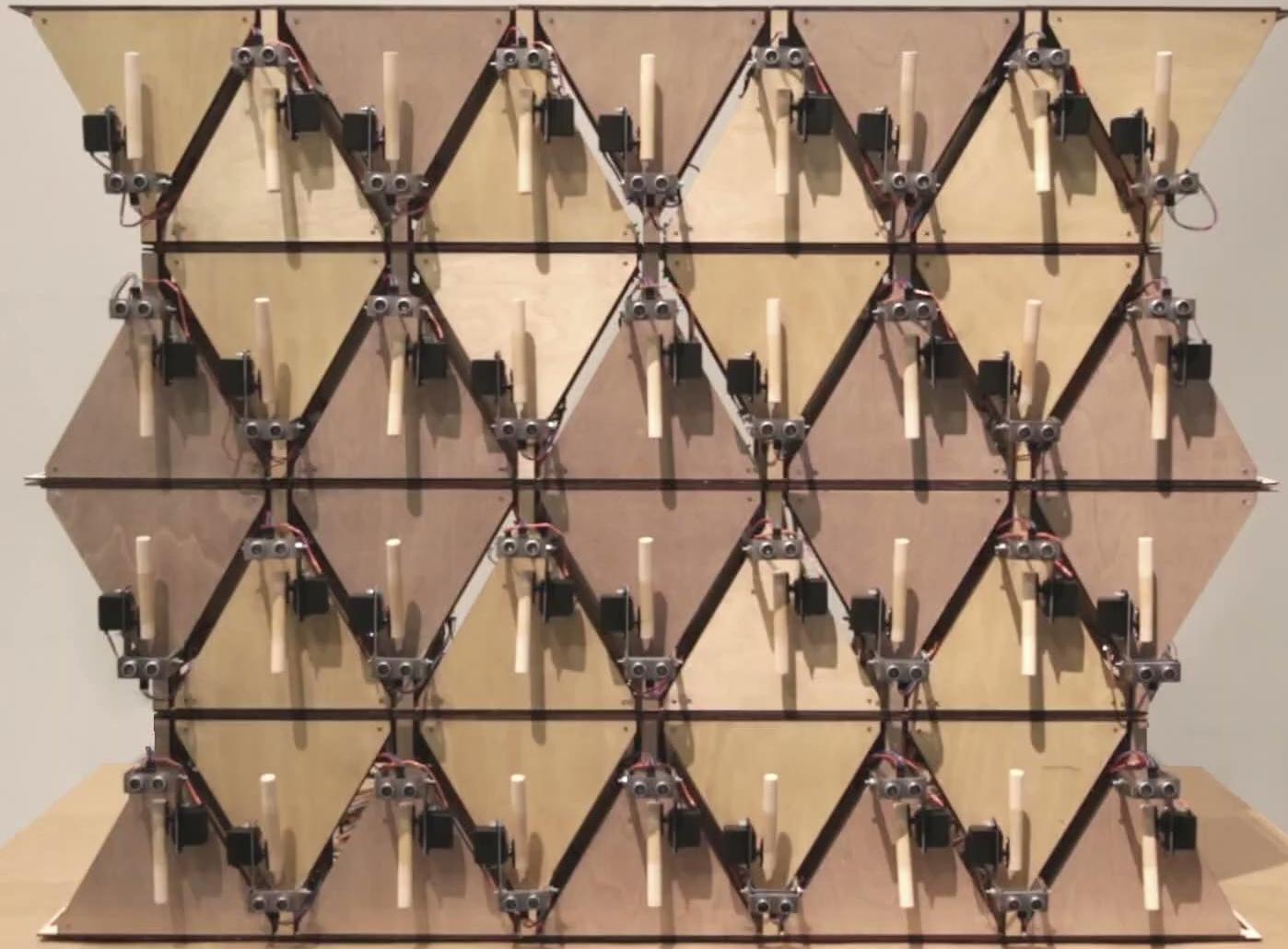
Image courtesy: Natalina from Instructables



Modular Rhythm Machine



Image courtesy: Nicolás Kisic Aguirre



Your Tool: Grove Beginner Kit

Arduino + 10 pre-connected modules

The center piece is the Arduino

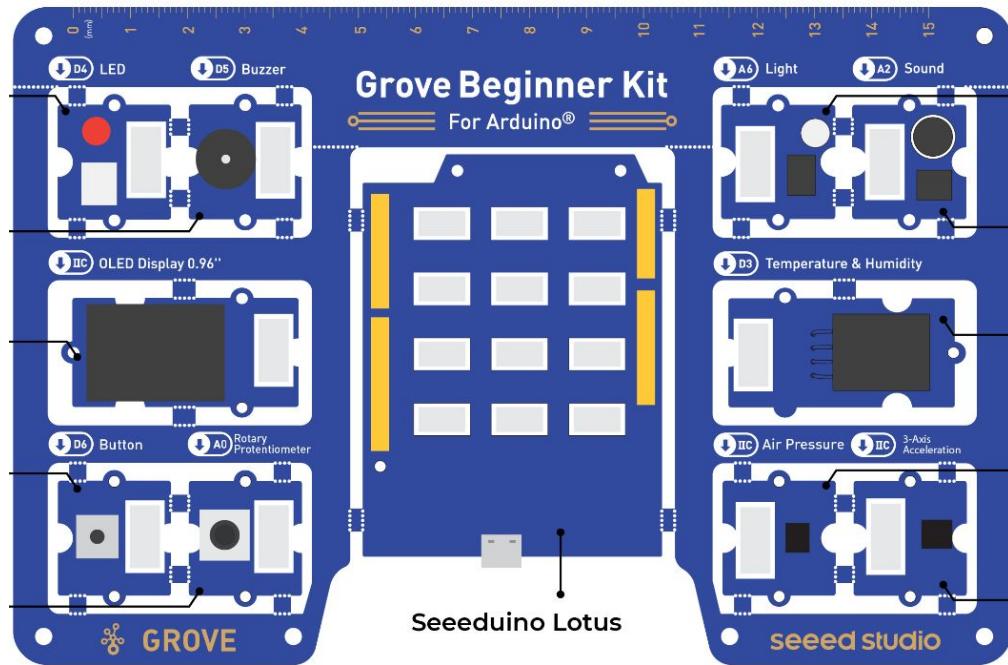


Image courtesy: Seeed Studio

Left Side Modules— Direct Interaction

- 1 LED
- 2 Buzzer: Make beeping sound
- 3 A (very small) screen
- 4 Button!
- 5 Potentiometer: Rotate ~ 360 deg, detects the **angle of your rotation**

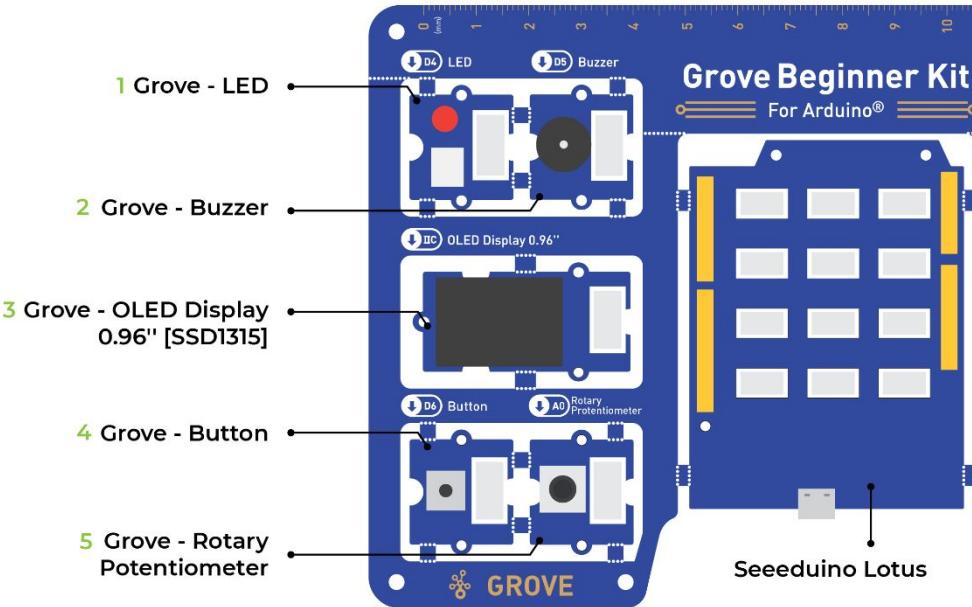
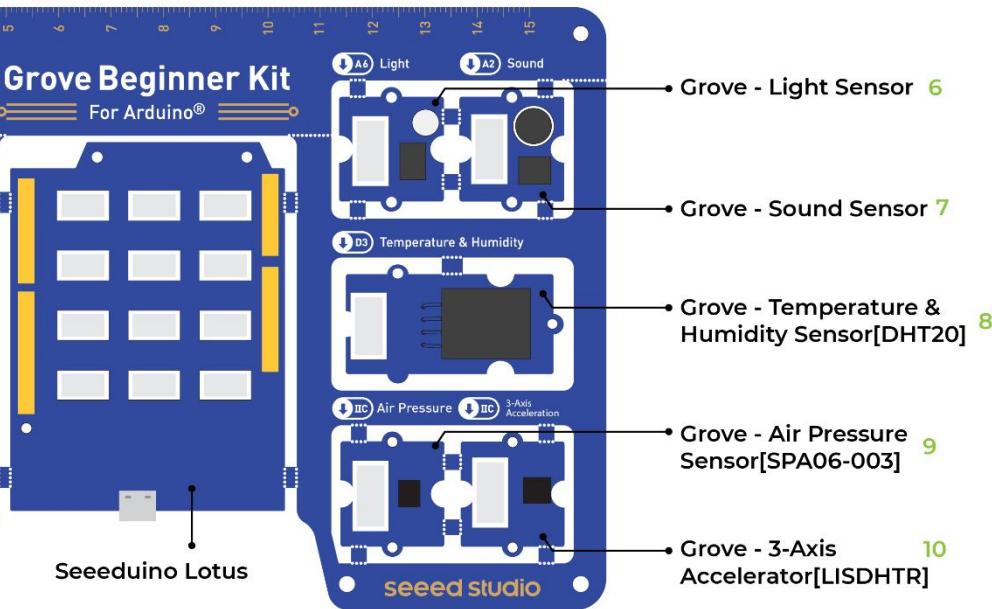


Image courtesy: Seeed Studio

Right Side Modules— Environmental Sensors



- 6 Detect **light** intensity
- 7 Detect **sound** intensity
- 8 Detect **temperature** and **humidity**
- 9 Detect **atmospheric pressure**
- 10 Detects object's **acceleration**

Right Side— Environmental Sensors

- 6 Detect **light** intensity
- 7 Detect **sound** intensity
- 8 Detect **temperature** and **humidity**
- 9 Detect **atmospheric pressure**
- 10 Detects object's **acceleration**

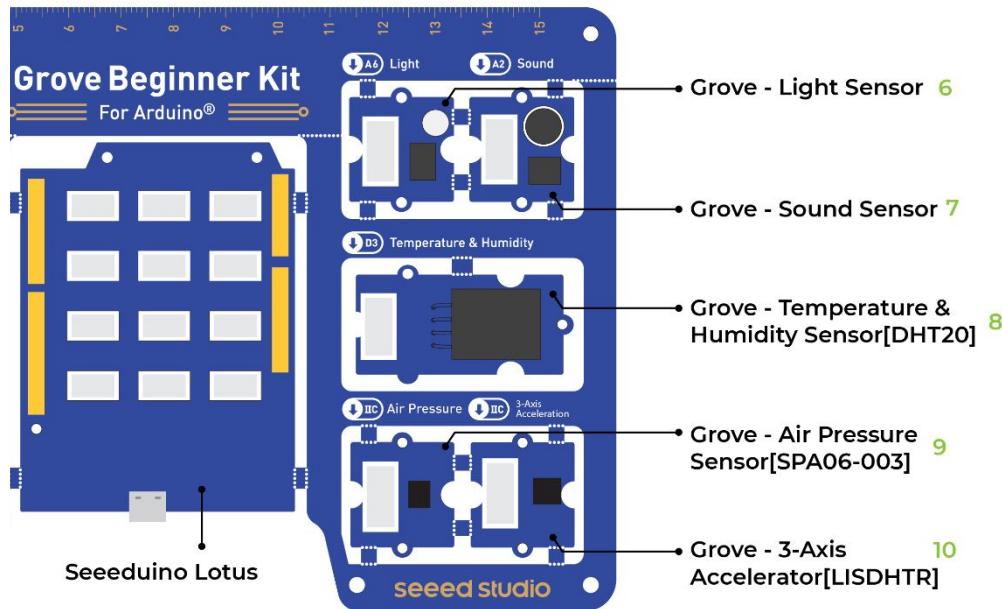
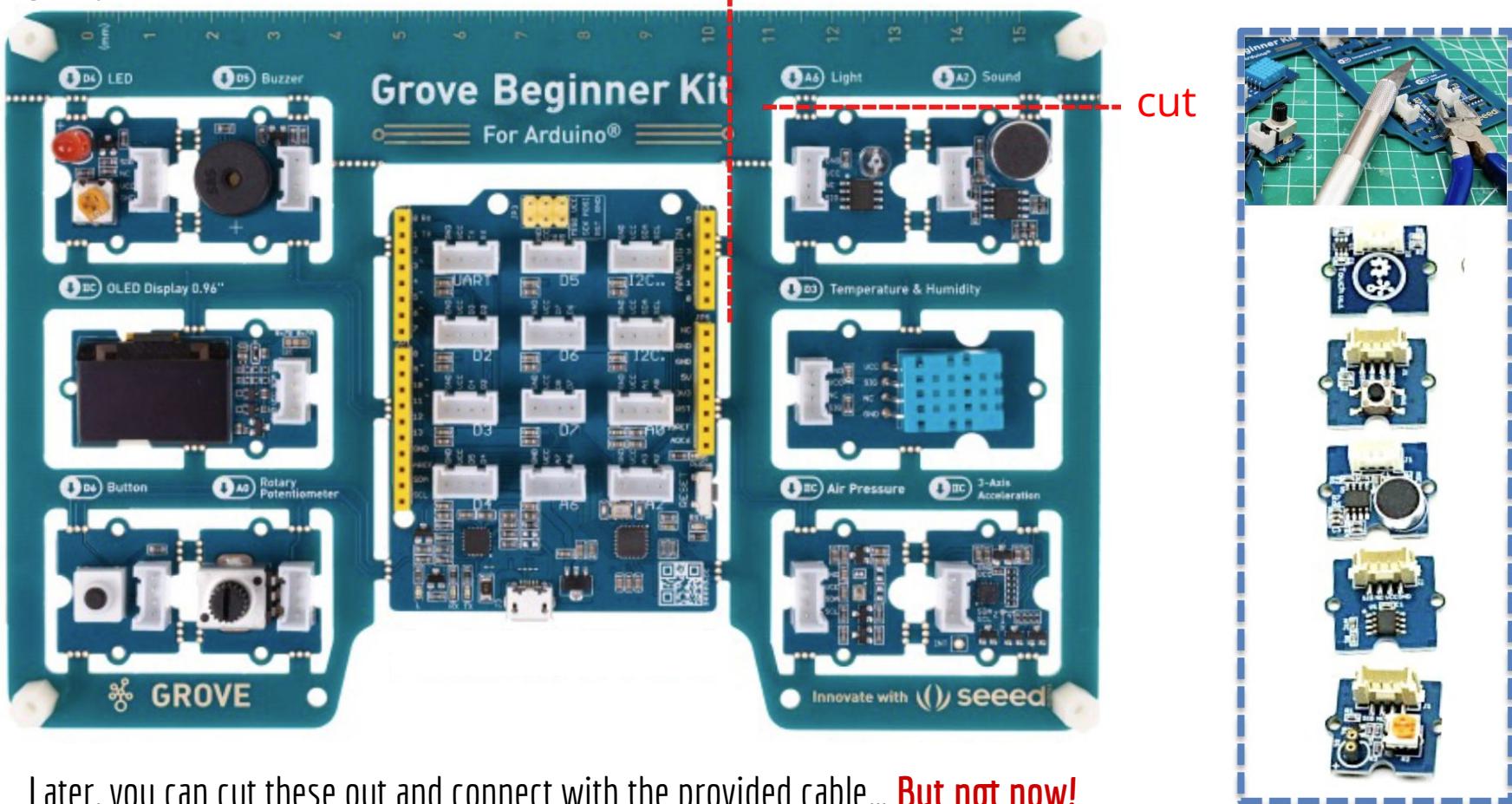
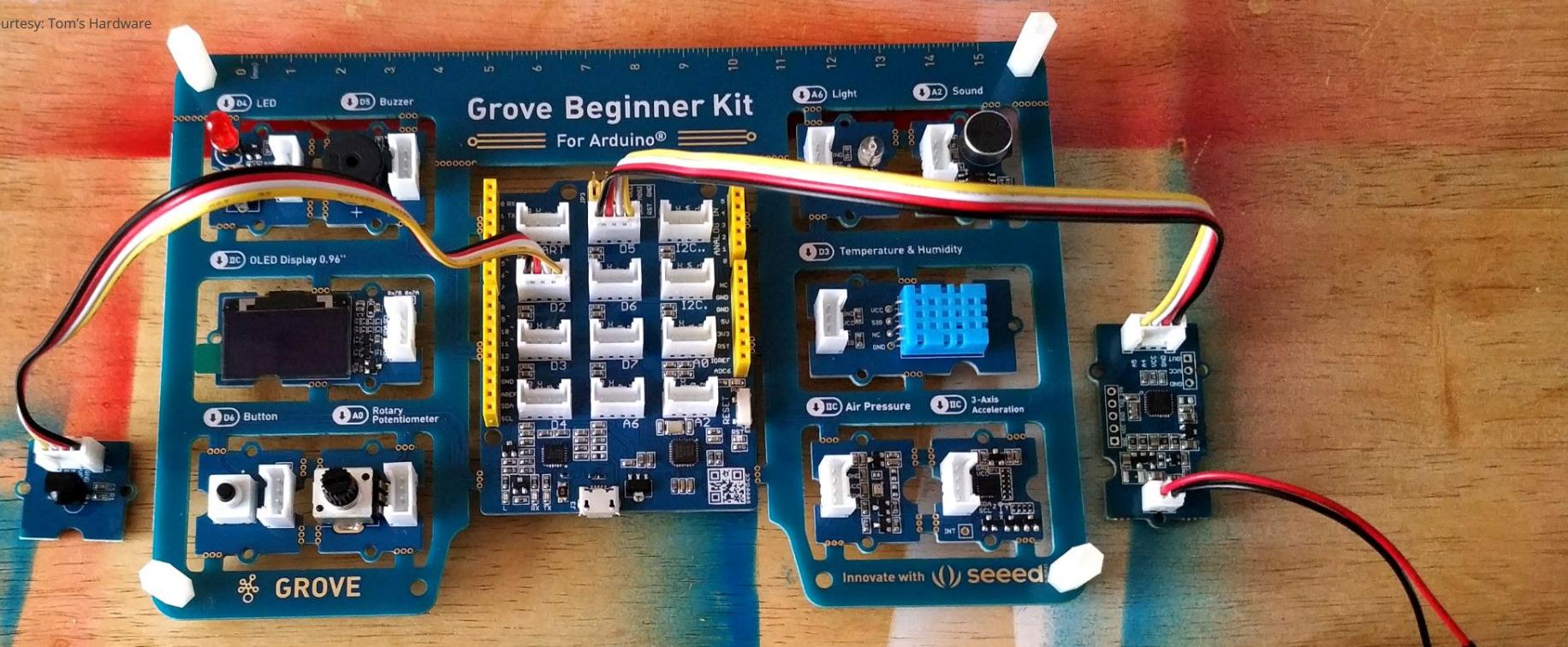


Image courtesy: Seeed Studio



Later, you can cut these out and connect with the provided cable... **But not now!**



Also you can add more modules you need / like!



Power On & Demo

No Coding Needed (Yet!)

1. Plug in: Connect the USB cable to your laptop or power bank.
2. Explore: **Twist the dial** to scroll through module demos.
3. Select: **Press the button** to try one.
4. Exit: **Long-press the button** to go back to the menu.

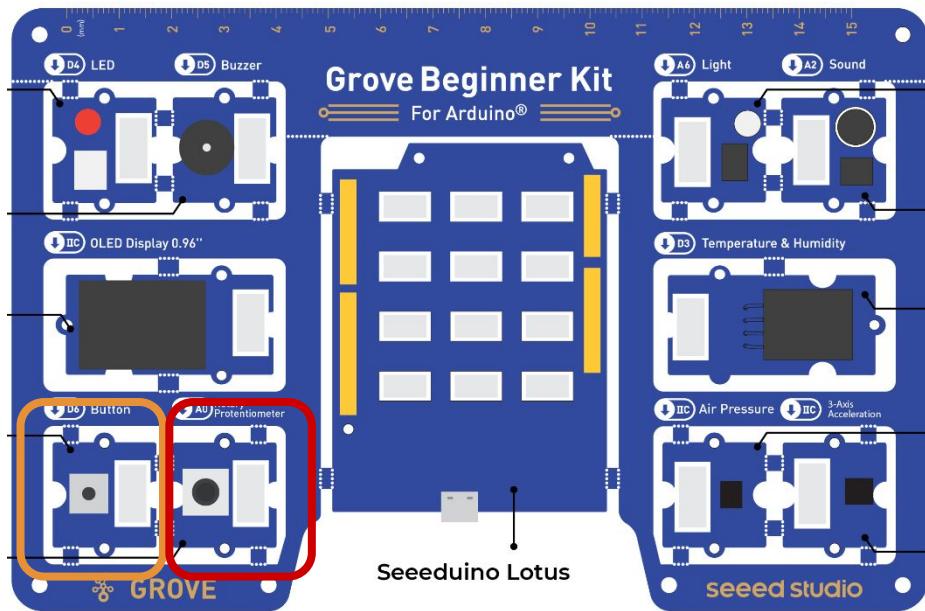


Image courtesy: Seeed Studio

Creating Your Code—Set up IDE

Bring Your Projects to Life with Arduino Software

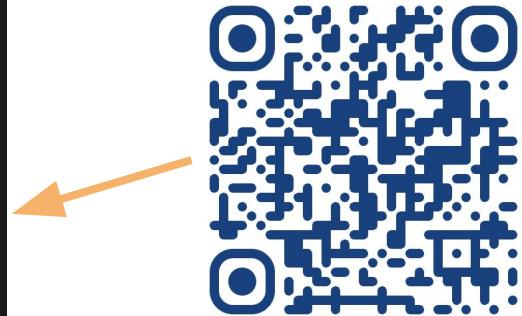


The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocomplete, code navigation, and even a live debugger. For more details, check the [Arduino IDE 2.0 documentation](#).

macOS Intel 10.15 Catalina or newer (64-bit) [DOWNLOAD](#)

Nightly Builds
Download a preview of the incoming release with the most updated features and bugfixes.

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

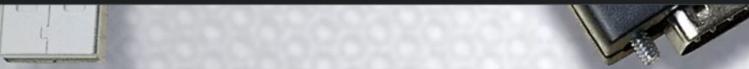


<https://www.arduino.cc/en/software/>

Mac Users: choose *Apple Silicone* if you have M series chip

macOS Intel 10.15 Catalina or newer (64-bit)

macOS Apple Silicon 11 Big Sur or newer (64-bit)



OVERVIEW DOWNLOADS TECH DOCS COMMUNITY & SUPPORT

Download and Install VCP Drivers

Downloads for Windows, Macintosh, Linux and Android below.

*Note: The Linux 3.x.x and 4.x.x version of the driver is maintained in the current Linux 3.x.x and 4.x.x tree at www.kernel.org.

Software Downloads

Software (11)

Software · 11

CP210x Universal Windows Driver	v11.5.0	12/30/2025
CP210x VCP Mac OSX Driver	v6.0.3	5/30/2025
CP210x VCP Windows	v6.7	9/3/2020
CP210x Windows Drivers	v6.7.6	9/3/2020
CP210x Windows Drivers with Serial Enumerator	v6.7.6	9/3/2020

Show 6 more Software

Download USB Driver

Select the Port

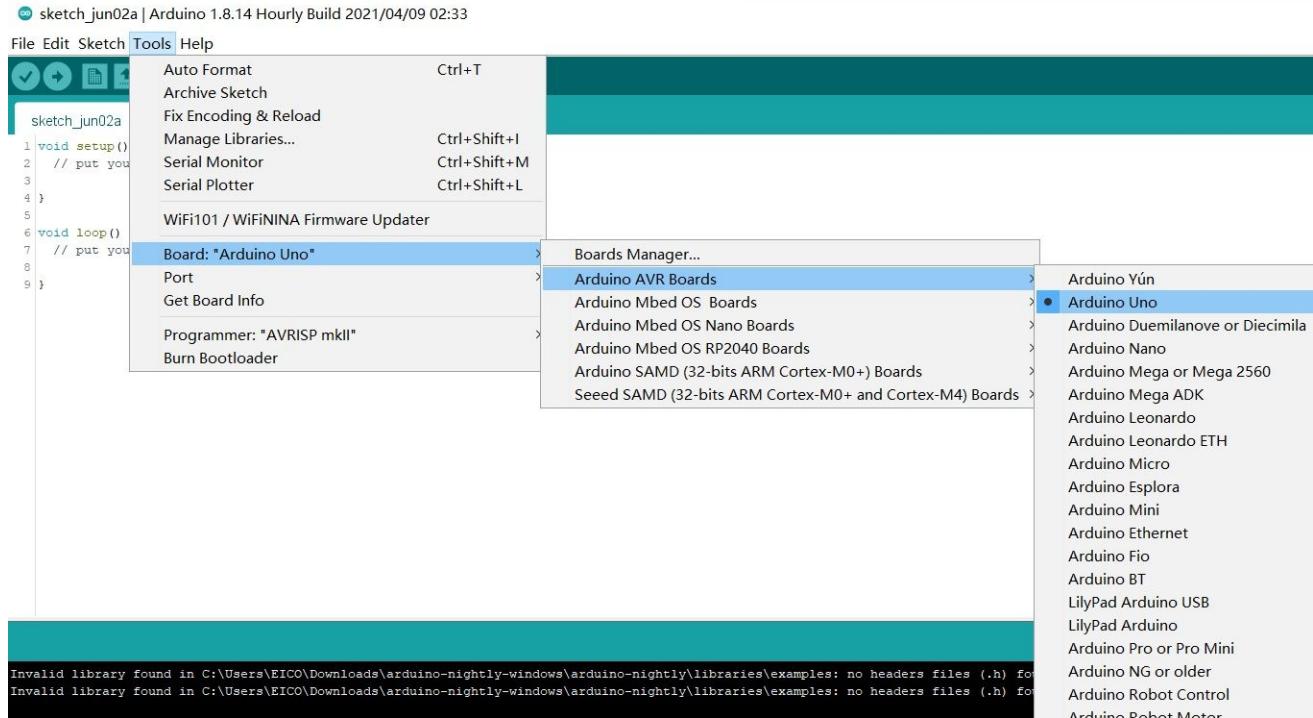


Image courtesy: Seeed Studio

Select the Board

Board: Tools → Board → Arduino Uno

Port: Tools → Port → **COMX**
(Arduino Uno) (Windows) or
xx.usbserial-xxxx (Mac)

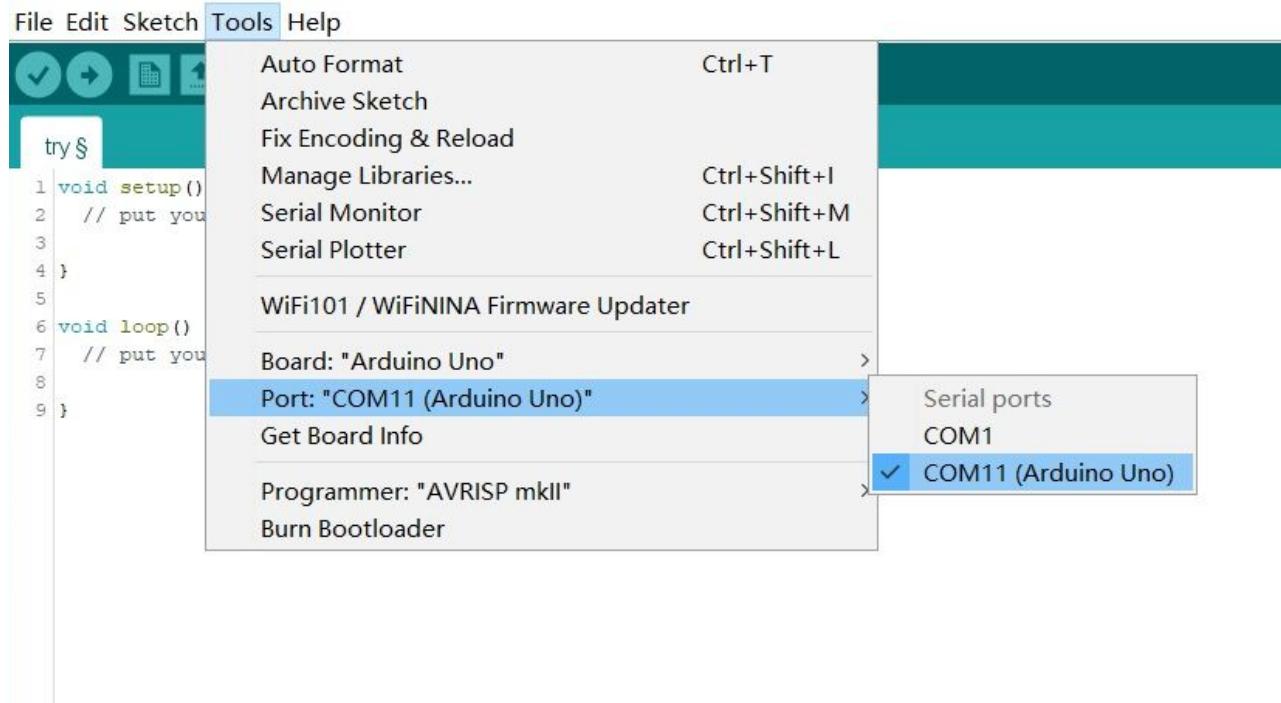
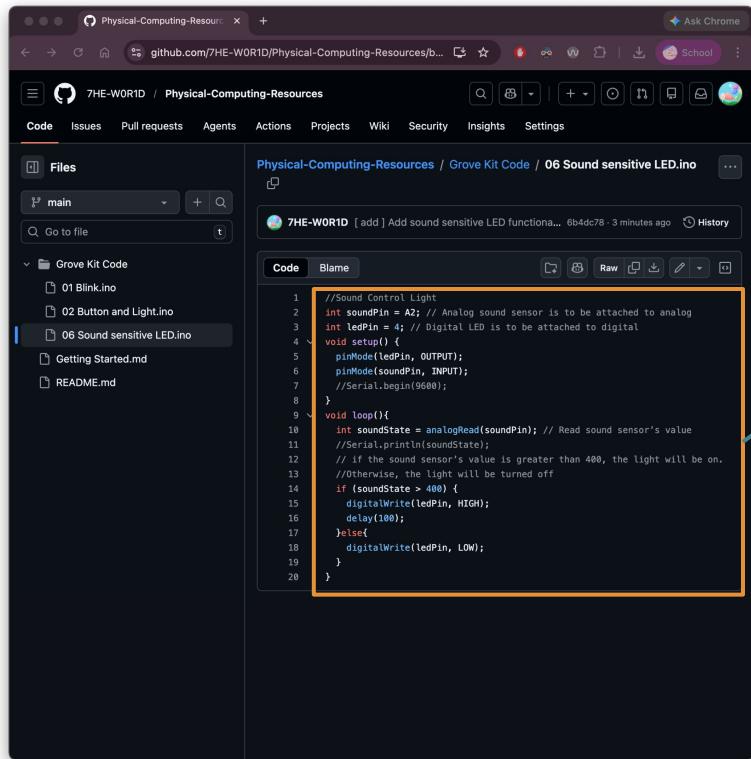


Image courtesy: Seeed Studio

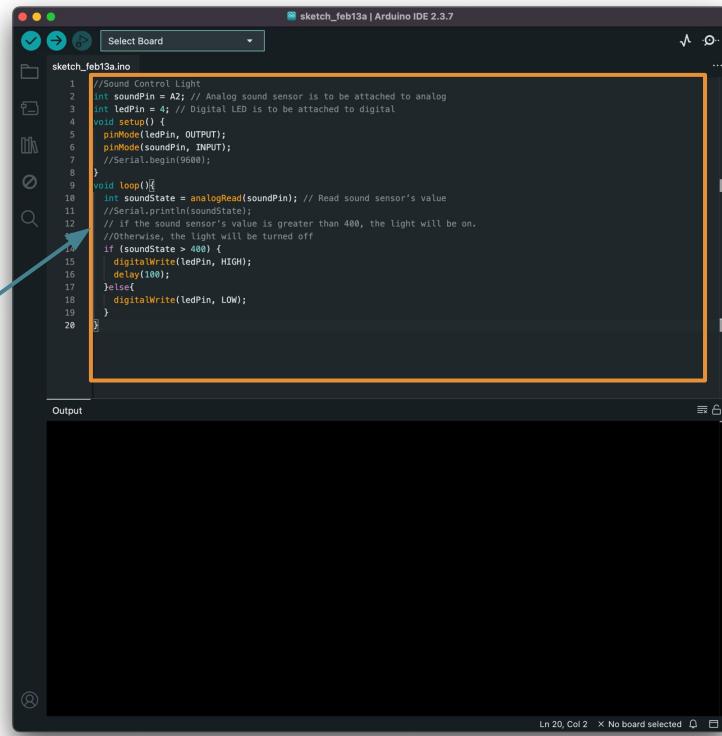
Make the LED blink

<https://tinyurl.com/INFO4940-soundblink>



The screenshot shows a GitHub repository for "Physical-Computing-Resources". The repository contains several files, including "main", "01 Blink.ino", "02 Button and Light.ino", and "06 Sound sensitive LED.ino". The "06 Sound sensitive LED.ino" file is selected and shown in the code editor. The code is as follows:

```
//Sound Control Light
int soundPin = A2; // Analog sound sensor is to be attached to analog
int ledPin = 4; // Digital LED is to be attached to digital
void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(soundPin, INPUT);
  //Serial.begin(9600);
}
void loop(){
  int soundState = analogRead(soundPin); // Read sound sensor's value
  //Serial.println(soundState);
  // if the sound sensor's value is greater than 400, the light will be on.
  //Otherwise, the light will be turned off
  if (soundState > 400) {
    digitalWrite(ledPin, HIGH);
    delay(100);
  }else{
    digitalWrite(ledPin, LOW);
  }
}
```

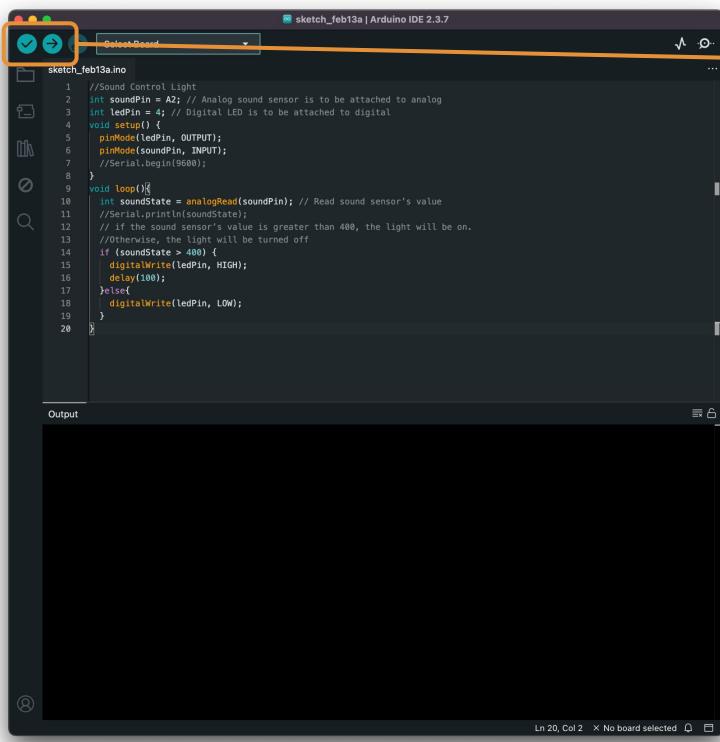


The screenshot shows the Arduino IDE with the sketch "sketch_feb13a.ino" open. The code is identical to the one in the GitHub repository, with the first few lines commented out. The code is as follows:

```
//Sound Control Light
int soundPin = A2; // Analog sound sensor is to be attached to analog
int ledPin = 4; // Digital LED is to be attached to digital
void setup() {
  //pinMode(ledPin, OUTPUT);
  //pinMode(soundPin, INPUT);
  //Serial.begin(9600);
}
void loop(){
  int soundState = analogRead(soundPin); // Read sound sensor's value
  //Serial.println(soundState);
  // if the sound sensor's value is greater than 400, the light will be on.
  //Otherwise, the light will be turned off
  if (soundState > 400) {
    digitalWrite(ledPin, HIGH);
    delay(100);
  }else{
    digitalWrite(ledPin, LOW);
  }
}
```

Make the LED blink

<https://tinyurl.com/INFO4940-soundblink>



The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** sketch_feb13a | Arduino IDE 2.3.7
- Sketch Name:** sketch_feb13a.ino
- Code Area:** Contains C++ code for a sound-controlled LED. The code reads the analog sound sensor value and toggles a digital LED based on whether it's greater than 400.

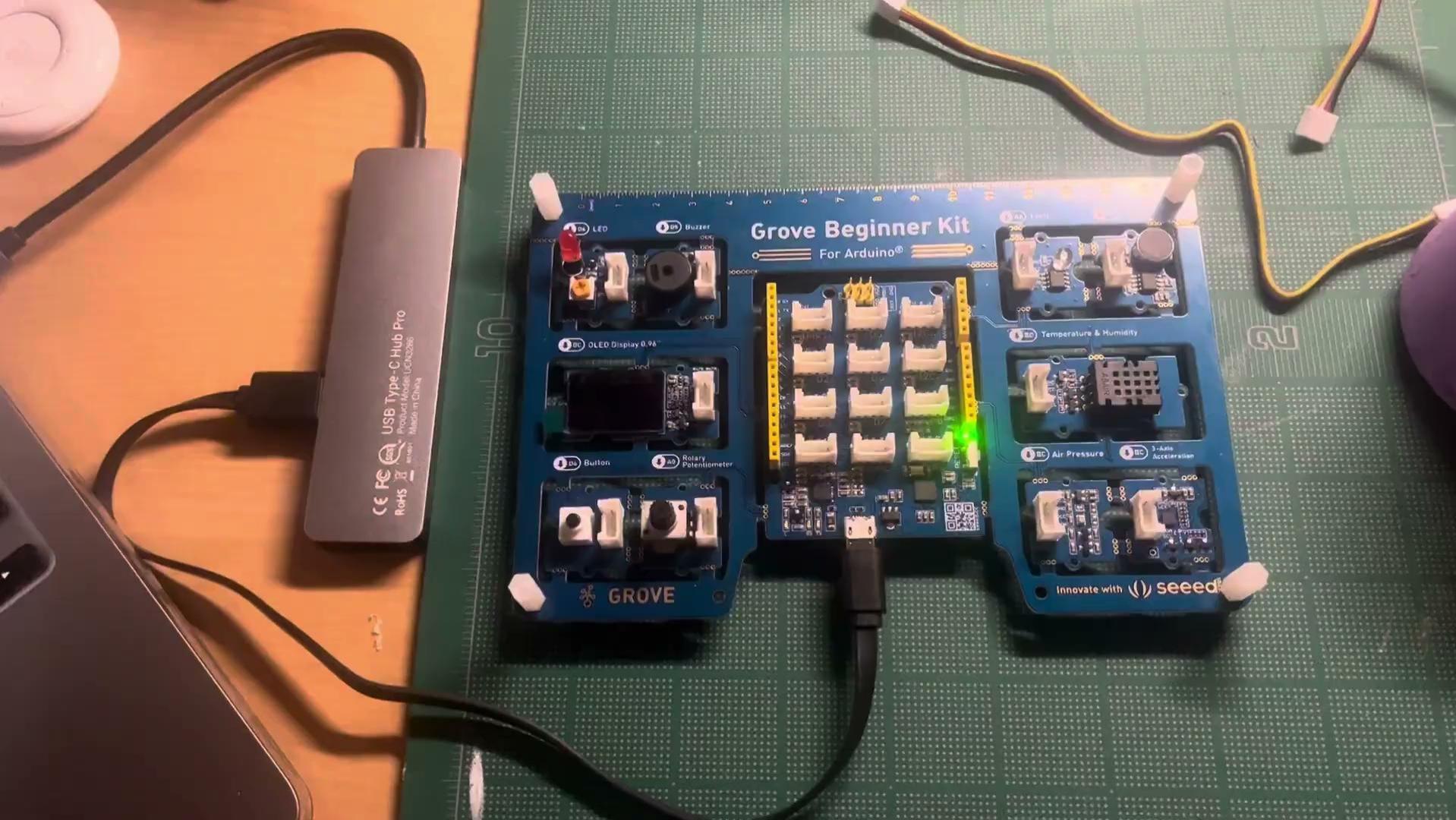
```
1 //Sound Control Light
2 int soundPin = A2; // Analog sound sensor is to be attached to analog
3 int ledPin = 4; // Digital LED is to be attached to digital
4
5 void setup() {
6     pinMode(ledPin, OUTPUT);
7     pinMode(soundPin, INPUT);
8     //Serial.begin(9600);
9 }
10 void loop(){
11     int soundState = analogRead(soundPin); // Read sound sensor's value
12     //Serial.println(soundState);
13     // if the sound sensor's value is greater than 400, the light will be on.
14     //Otherwise, the light will be turned off.
15     if (soundState > 400) {
16         digitalWrite(ledPin, HIGH);
17         delay(100);
18     }else{
19         digitalWrite(ledPin, LOW);
20     }
}
```

- Output Area:** Empty, showing "Ln 20, Col 2 X No board selected".



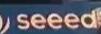
Click on  to verify code (app checks for obvious errors)

Then, if no errors, click on  to upload code to Arduino



CE FCC RoHS
Product Model: UCH20268
Made in China

Grove Beginner Kit
For Arduino®

Innovate with  seeed

What if I have errors?

I see "No board selected" or "Missing FOBN".

Fix: Go back to Tools > Board and Tools > Port to make sure "Arduino Uno" is checked.

No error shown, but my LED is not lighting up

Fix: double-check your pin numbers in the code, is it 4 for LED (not 13)?

The computer doesn't show any "COM" or "usbserial" ports.

Fix: Install the CP210x Driver.

...it's not any of those!

Use AI to help you!

Copy error messages

Copy **all** of your code

The screenshot shows the Arduino IDE interface. The top bar indicates "sketch_feb13a | Arduino IDE 2.3.7". The main area displays the code for "sketch_feb13a.ino". The code reads sound sensor data and controls an LED based on the sound level. A specific line of code is highlighted in red:

```
1 //Sound Control Light
2 int soundPin = A2; // Analog sound sensor is to be attached to analog
3 int ledPin = 4; // Digital LED is to be attached to digital
4 void setup() {
5   pinMode(ledPin, OUTPUT);
6   pinMode(soundPin, INPUT);
7   //Serial.begin(9600);
8 }
9 void loop(){
10   int soundState = analogRead(soundPin); // Read sound sensor's value
11   //Serial.println(soundState);
12   // if the sound sensor's value is greater than 400, the light will be on.
13   //Otherwise, the light will be turned off
14   if (soundState > 400) {
15     digitalWrite(ledPin, HIGH);
16     delay(100);
17   }else{
18     digitalWrite(ledPin, LOW)
19   }
20 }
```

The "Output" window at the bottom shows the compilation results:

```
/private/var/folders/q4/7s_566js4h5_k9lmhwrgq_c0000gn/T/.arduinoIDE-unsaved20261113-3723-shc5xe.8txg/sketch_feb13a/sketch_feb13a.ino
/private/var/folders/q4/7s_566js4h5_k9lmhwrgq_c0000gn/T/.arduinoIDE-unsaved20261113-3723-shc5xe.8txg/sketch_feb13a/sketch_feb13a.ino
}
^
exit status 1

Compilation error: expected ';' before ')' token
```

A callout box highlights the error message "Compilation error: expected ';' before ')' token" in the output window. A blue button labeled "COPY ERROR MESSAGES" is located at the bottom right of the callout.

...it's not any of those!

Paste it using the following syntax
(or edit to fit your style)

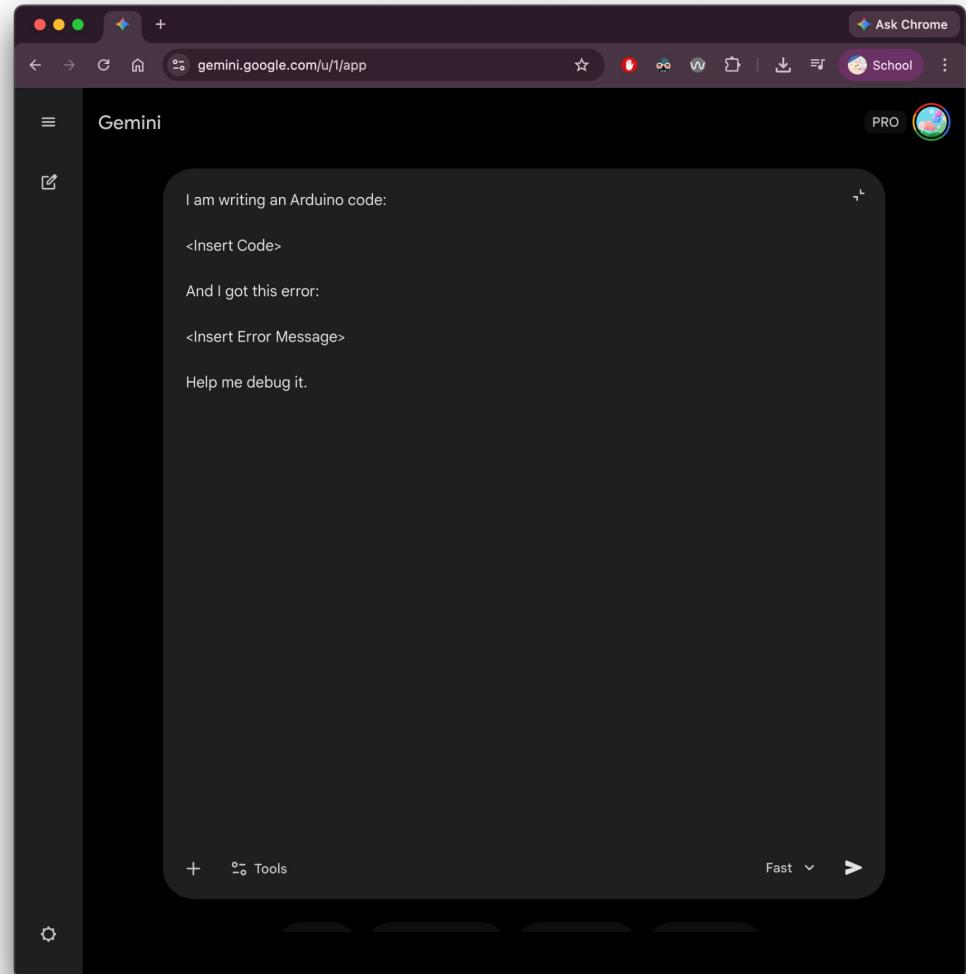
I am writing an Arduino
code:

<Insert Code>

And I got this error:

<Insert Error Message>

Help me debug it.



I want the light to be more/less sensitive

But how much more / less sensitive?

Arduino can give you the answer!

Delete the **//** at the beginning of these lines

```
//Serial.begin(9600);  
  
//Serial.println(soundState);  
  
//delay(100);
```

Now they should be **colored**

Verify code and **upload**



```
06_Sound_sensitive_LED | Arduino IDE 2.3.7  
Select Board  
06_Sound_sensitive_LED.ino  
1 //Sound Control Light  
2 int soundPin = A2; // Analog sound sensor is to be attached to analog  
3 int ledPin = 4; // Digital LED is to be attached to digital  
4 void setup() {  
5     pinMode(ledPin, OUTPUT);  
6     pinMode(soundPin, INPUT);  
7     //Serial.begin(9600); ←  
8 }  
9 void loop(){  
10    int soundState = analogRead(soundPin); // Read sound sensor's value  
11    //Serial.println(soundState); ←  
12    // if the sound sensor's value is greater than 400, the light will be on.  
13    //Otherwise, the light will be turned off  
14    if (soundState > 400) {  
15        digitalWrite(ledPin, HIGH);  
16        delay(100);  
17    }else{  
18        digitalWrite(ledPin, LOW);  
19    }  
20    //delay(100); ←  
21 }  
22 }
```

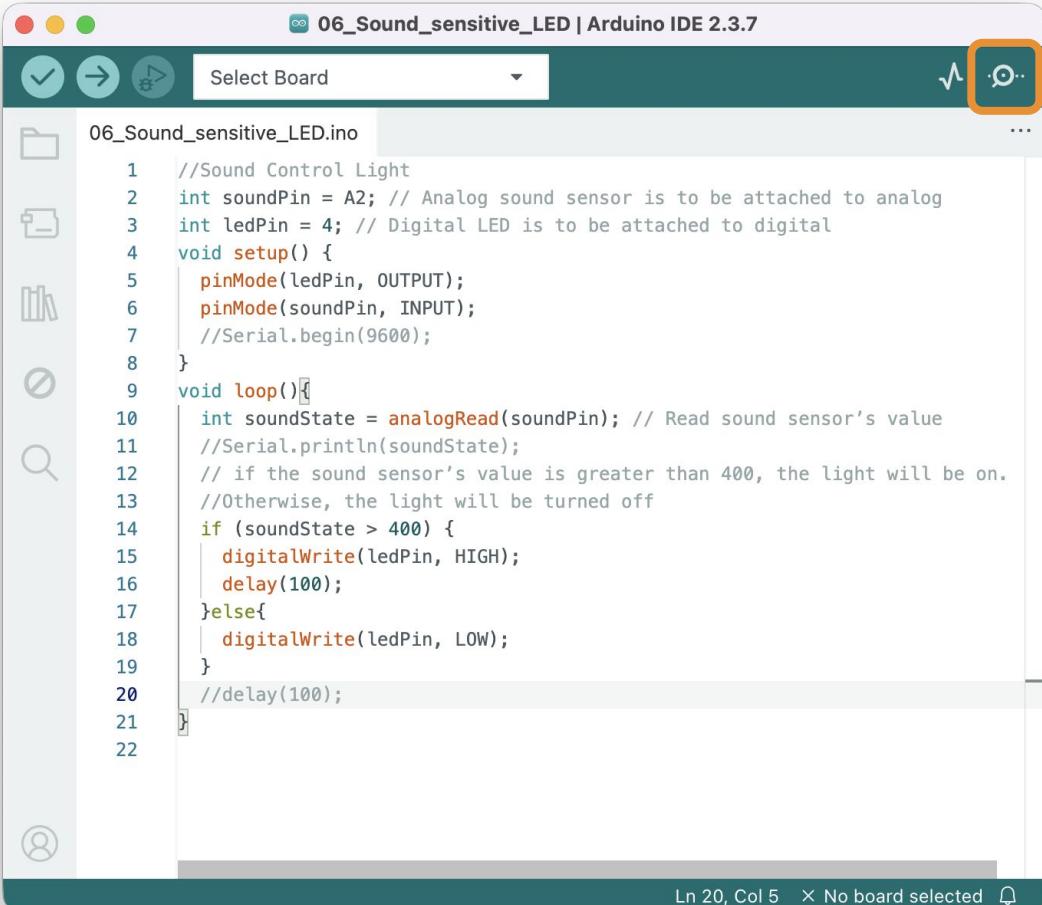
So data won't scroll too fast!

Ln 20, Col 5 X No board selected

I want the light to be more/less sensitive

Open Serial Monitor  at the top right corner

Serial monitor is the way you “listen” and “talk” to the Arduino.



The screenshot shows the Arduino IDE interface with the title bar "06_Sound_sensitive_LED | Arduino IDE 2.3.7". A red box highlights the "Serial Monitor" icon in the top right corner. The left sidebar shows a file tree with "06_Sound_sensitive_LED.ino" selected. The main code editor contains the following sketch:

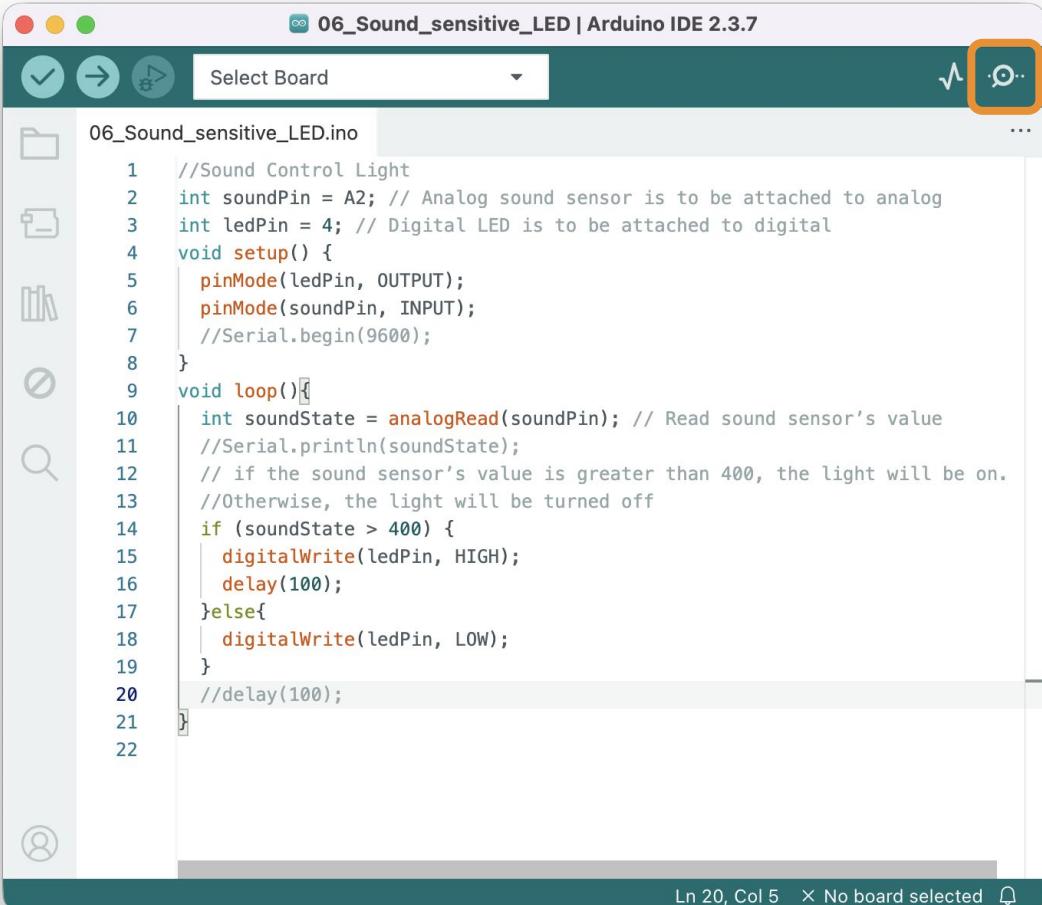
```
06_Sound_sensitive_LED.ino
1 //Sound Control Light
2 int soundPin = A2; // Analog sound sensor is to be attached to analog
3 int ledPin = 4; // Digital LED is to be attached to digital
4 void setup() {
5     pinMode(ledPin, OUTPUT);
6     pinMode(soundPin, INPUT);
7     //Serial.begin(9600);
8 }
9 void loop(){
10    int soundState = analogRead(soundPin); // Read sound sensor's value
11    //Serial.println(soundState);
12    // if the sound sensor's value is greater than 400, the light will be on.
13    //Otherwise, the light will be turned off
14    if (soundState > 400) {
15        digitalWrite(ledPin, HIGH);
16        delay(100);
17    }else{
18        digitalWrite(ledPin, LOW);
19    }
20    //delay(100);
21 }
22 }
```

The status bar at the bottom right shows "Ln 20, Col 5" and "No board selected".

I want the light to be more/less sensitive

Open Serial Monitor  at the top right corner

Serial monitor is the way you “listen” and “talk” to the Arduino.



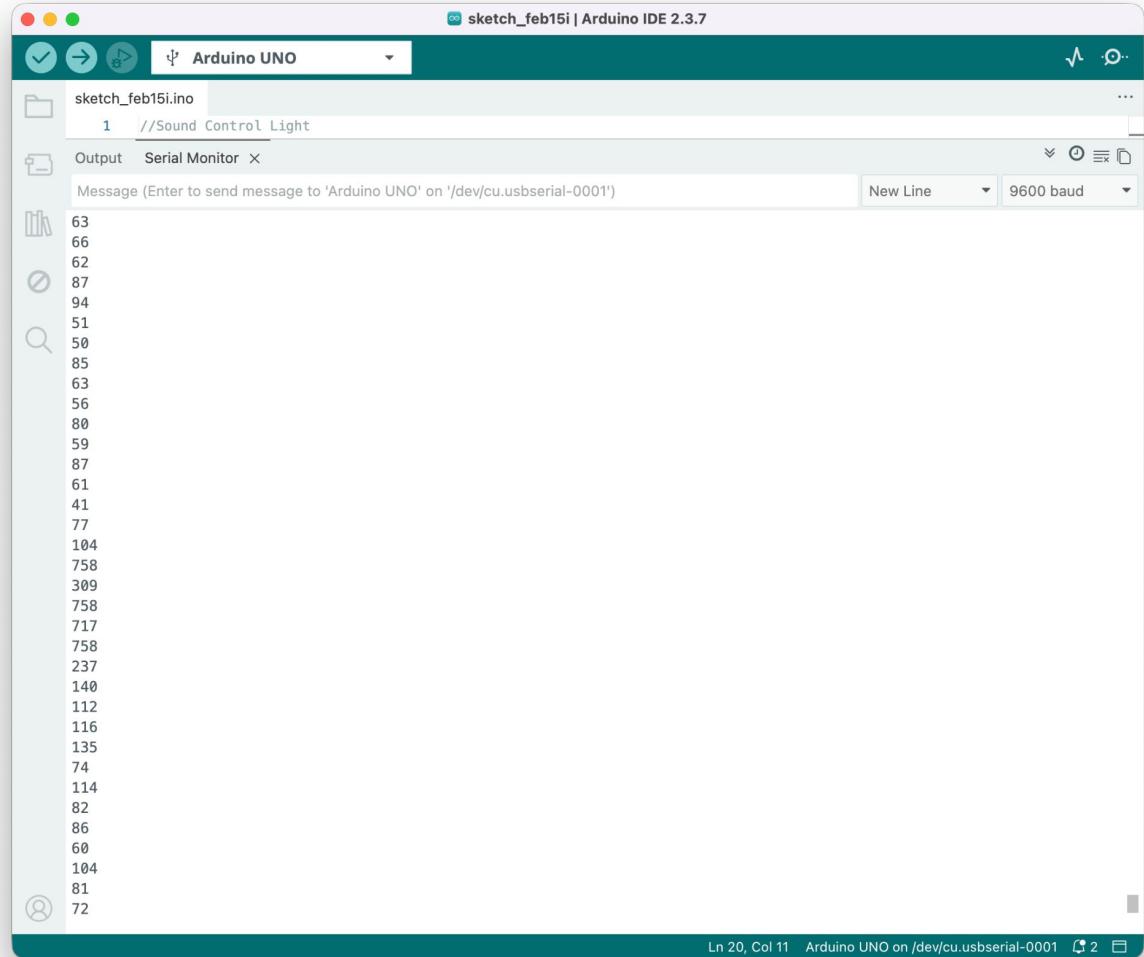
The screenshot shows the Arduino IDE interface with the title bar "06_Sound_sensitive_LED | Arduino IDE 2.3.7". A red box highlights the "Serial Monitor" icon in the top right corner. The left sidebar shows a file tree with "06_Sound_sensitive_LED.ino" selected. The main code editor contains the following sketch:

```
06_Sound_sensitive_LED.ino
1 //Sound Control Light
2 int soundPin = A2; // Analog sound sensor is to be attached to analog
3 int ledPin = 4; // Digital LED is to be attached to digital
4 void setup() {
5     pinMode(ledPin, OUTPUT);
6     pinMode(soundPin, INPUT);
7     //Serial.begin(9600);
8 }
9 void loop(){
10    int soundState = analogRead(soundPin); // Read sound sensor's value
11    //Serial.println(soundState);
12    // if the sound sensor's value is greater than 400, the light will be on.
13    //Otherwise, the light will be turned off
14    if (soundState > 400) {
15        digitalWrite(ledPin, HIGH);
16        delay(100);
17    }else{
18        digitalWrite(ledPin, LOW);
19    }
20    //delay(100);
21 }
22 }
```

The status bar at the bottom right shows "Ln 20, Col 5" and "No board selected".

I want the light to be more/less sensitive

Do your desired action and see the change in numbers!



The screenshot shows the Arduino IDE interface. The title bar reads "sketch_feb15i | Arduino IDE 2.3.7". The main window shows a file named "sketch_feb15i.ino" with the code block containing the comment "//Sound Control Light". Below the code editor is the "Serial Monitor" window, which has tabs for "Output" and "Serial Monitor". The "Serial Monitor" tab is active, displaying the message "Message (Enter to send message to 'Arduino UNO' on '/dev/cu.usbserial-0001')". The text area of the monitor shows a series of numerical values: 63, 66, 62, 87, 94, 51, 50, 85, 63, 56, 80, 59, 87, 61, 41, 77, 104, 758, 309, 758, 717, 758, 237, 140, 112, 116, 135, 74, 114, 82, 86, 60, 104, 81, 72. At the bottom of the monitor window, there are buttons for "New Line" and "9600 baud". The status bar at the bottom of the IDE window shows "Ln 20, Col 11 Arduino UNO on /dev/cu.usbserial-0001" and some small icons.

Empty or garbled text?

Check if your `Serial.begin(XXXX)`
matches your **serial monitor baud!**

The screenshot shows the Arduino IDE interface. The top bar displays the title "06_Sound_sensitive_LED | Arduino IDE 2.3.7". The main window shows the code for "06_Sound_sensitive_LED.ino". In the setup() function, the line `Serial.begin(9600);` is highlighted with a green box. A red arrow points from this box to a red rectangular callout containing the word "Wrong!". Another red arrow points from the "Wrong!" callout to the baud rate selection dropdown in the bottom right corner of the Serial Monitor tab, which is set to "19200 baud". The Serial Monitor tab shows a message box with placeholder text "Message (Enter to send message to 'Arduino UNO' on '/dev/cu.usbserial-0001' New Line)". The status bar at the bottom indicates "Ln 11, Col 3 Arduino UNO on /dev/cu.usbserial-0001".

```
06_Sound_sensitive_LED.ino
1 //Sound Control Light
2 int soundPin = A2; // Analog sound sensor is to be attached to analog
3 int ledPin = 4; // Digital LED is to be attached to digital
4 void setup() {
5     pinMode(ledPin, OUTPUT);
6     pinMode(soundPin, INPUT);
7     Serial.begin(9600);
8 }
```

Serial Monitor X

Message (Enter to send message to 'Arduino UNO' on '/dev/cu.usbserial-0001' New Line) 19200 baud

Ln 11, Col 3 Arduino UNO on /dev/cu.usbserial-0001

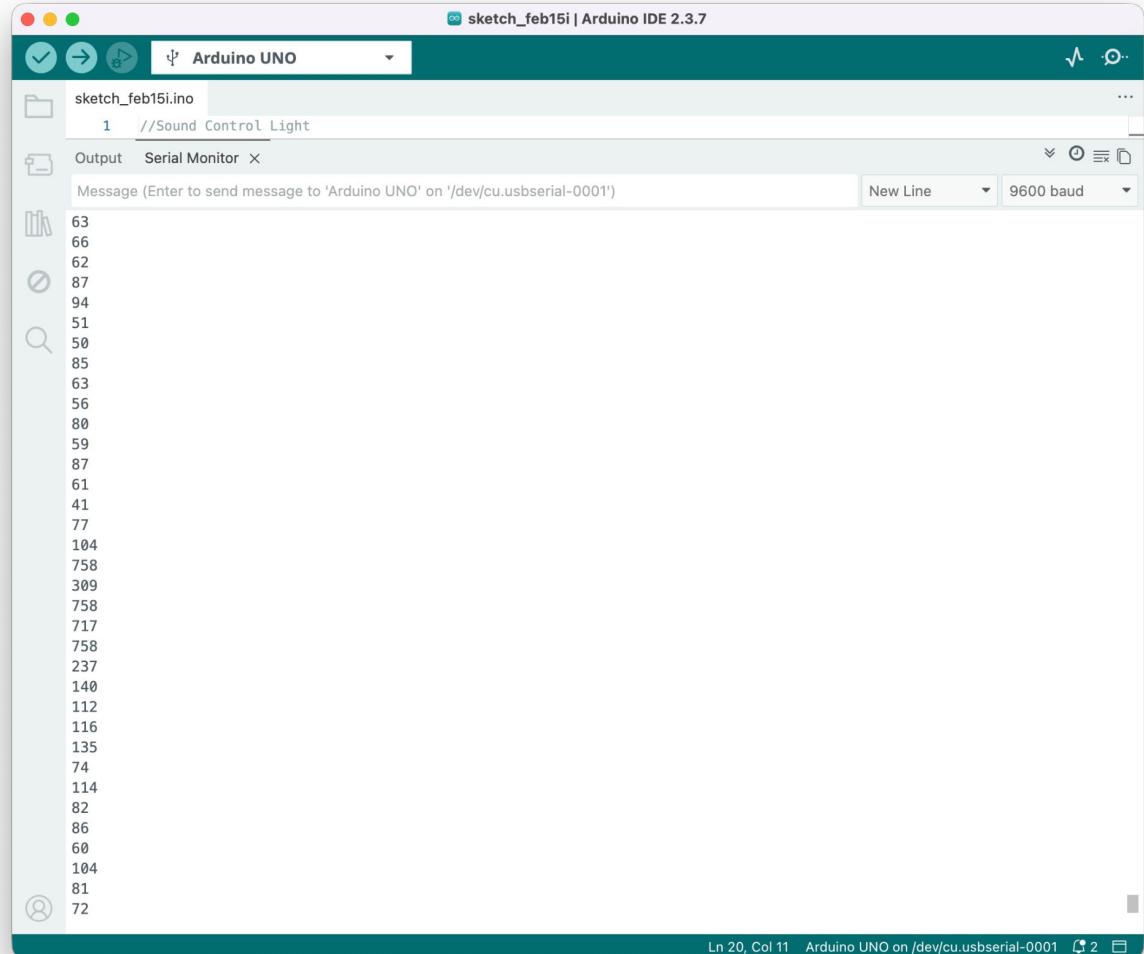
I want the light to be more/less sensitive

```
56
80
59
87
61
41
77
104
758
309
758
717
758
237
140
112
116
135
74
114
82
86
60
104
```

Click

```
Value > 700 = Click
```

104



I want the light to be more/less sensitive

Change 400 in `soundState > 400` to your determined new number

The screenshot shows the Arduino IDE interface with the title bar "06_Sound_sensitive_LED | Arduino IDE 2.3.7". The central area displays the code for "06_Sound_sensitive_LED.ino". The code reads the value from an analog sound sensor (pin A2) and controls a digital LED (pin 4). In the `loop()` function, it checks if the sound state is greater than 400. If true, the LED is turned on (HIGH); otherwise, it is turned off (LOW). An orange callout box highlights the condition `if (soundState > 700)` with the text "700 now!".

```
06_Sound_sensitive_LED.ino
1 //Sound Control Light
2 int soundPin = A2; // Analog sound sensor is to be attached to analog
3 int ledPin = 4; // Digital LED is to be attached to digital
4 void setup() {
5     pinMode(ledPin, OUTPUT);
6     pinMode(soundPin, INPUT);
7     //Serial.begin(9600);
8 }
9 void loop(){
10    int soundState = analogRead(soundPin); // Read sound sensor's value
11    //Serial.println(soundState);
12    // if the sound sensor's value is greater than 400, the light will be on.
13    //Otherwise, the light will be turned off
14    if (soundState > 700) { 700 now!
15        digitalWrite(ledPin, HIGH);
16        delay(100);
17    }else{
18        digitalWrite(ledPin, LOW);
19    }
20    //delay(100);
21 }
22
```

Ln 22, Col 1 X No board selected

I want the light to be more/less sensitive

Change 400 in `soundState > 400` to your determined new number

Add back the `//` at the beginning of these lines

```
//Serial.begin(9600);

//Serial.println(soundState);

//delay(100);
```

Verify code and **upload**



```
06_Sound_sensitive_LED | Arduino IDE 2.3.7
Select Board
06_Sound_sensitive_LED.ino
1 //Sound Control Light
2 int soundPin = A2; // Analog sound sensor is to be attached to analog
3 int ledPin = 4; // Digital LED is to be attached to digital
4 void setup() {
5     pinMode(ledPin, OUTPUT);
6     pinMode(soundPin, INPUT);
7     //Serial.begin(9600); ←
8 }
9 void loop(){
10    int soundState = analogRead(soundPin); // Read sound sensor's value
11    //Serial.println(soundState); ←
12    // if the sound sensor's value is greater than 400, the light will be on.
13    //Otherwise, the light will be turned off
14    if (soundState > 700) {← 700 now!
15        digitalWrite(ledPin, HIGH);
16        delay(100);
17    }else{
18        digitalWrite(ledPin, LOW);
19    }
20    //delay(100); ←
21 }
```

Ln 22, Col 1 X No board selected

Reflections

Recall your answers: "What do you want to build?"

Nobody said, "I'm here to learn circuit theory today."

Recall the "30s Romantic Love" Assignment

We didn't critique your editing software or camera settings.

Our discussion focused on the **emotions and ideas** you conveyed.

The Camera was just a Tool

The **lens** didn't make the art; your **vision** did.

Arduino is no different

It is simply a "**lens**" to help you express a thought.

The Tool Mindset in Action

Don't Reinvent the Wheel

If you want to build something, there is likely already a Library or an AI prompt for that.

Just-in-Time Learning

Only learn the specific modules your project requires.

Be an "Architect," not a Scientist

Your role is to assemble these Grove modules like Lego bricks to tell a story.

"Good Enough" is Perfect

If the interaction works and the message is clear, the technology has done its job.

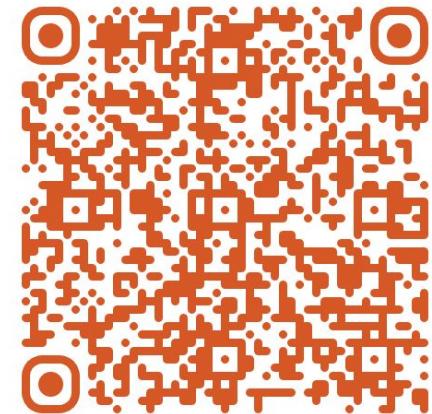
Useful Resources

[ARL Learning Slides](#): Detailed hardware walkthrough from my lab, built for a similar class.

[Instructables.com](#): A massive DIY library with step-by-step guides.

[Arduino Forum and Project Hub](#): The official hub. Use the "Project Guidance" section to discuss ideas or the "Troubleshooting" section for technical bugs.

And don't forget **your Professor and TA!**



Link to Slide

Questions?

Make the LED blink

<https://tinyurl.com/INFO4940-blink>

[https://tinyurl.com/INFO4940-sound
blink](https://tinyurl.com/INFO4940-soundblink)

More Complex Sensors

LIS3DHTR 3 axis accelerometer

Senses *movement* and *orientation*

How does it communicate with Arduino?

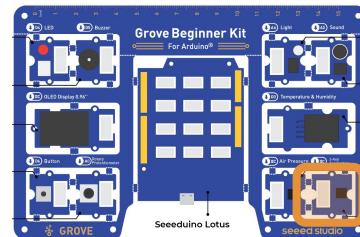
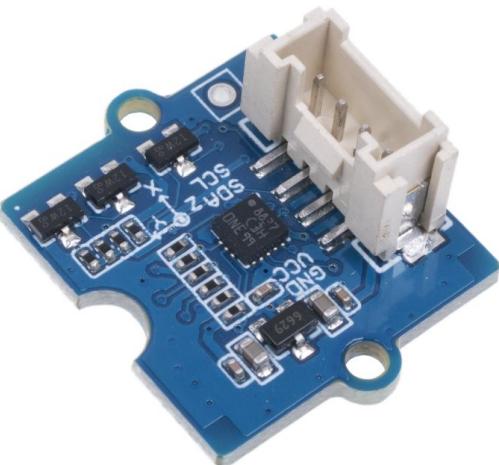


Image courtesy: Seeed Studio

More Complex Sensors

DHT 11/20 Temperature & Humidity Sensor

Senses *movement* and *orientation*

How does it communicate with Arduino?

*You don't need to know! You just need a **library**.*

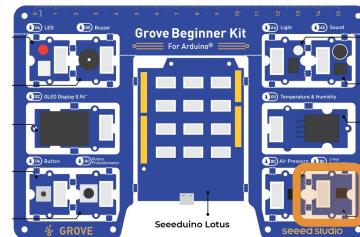
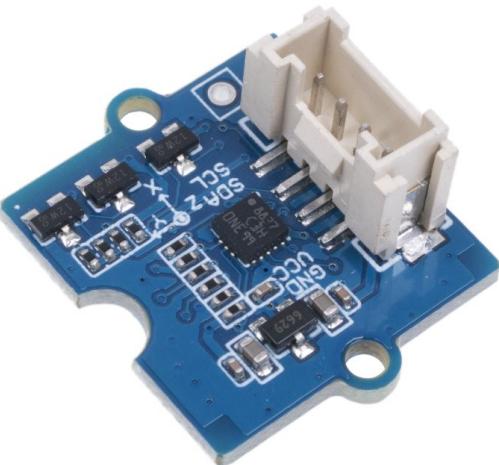


Image courtesy: Seeed Studio

More Complex Sensors

DHT 11/20 Temperature & Humidity Sensor

Senses *movement* and *orientation*

How does it communicate with Arduino?

*You don't need to know! You just need a **library**.*

Don't reinvent the wheel!

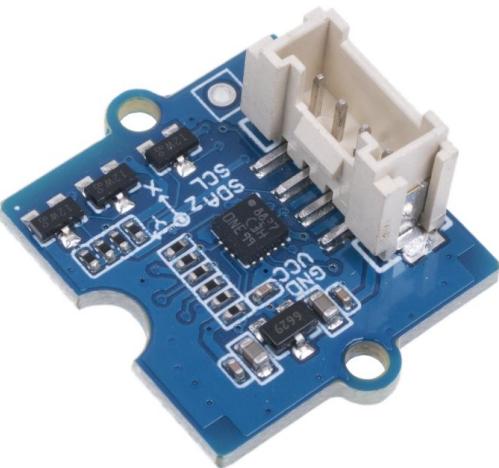
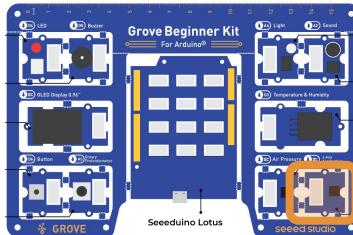


Image courtesy: Seeed Studio

Visit <https://tinyurl.com/INFO4940LIS>

The screenshot shows a GitHub repository page for the Seeed_Arduino_LIS3DHTR library. The repository is public and has 12 watchers, 22 forks, and 27 stars. It contains 2 branches and 5 tags. The repository was last updated by fea8012 last year with 46 commits. The repository page includes sections for About, Releases, and Build Status.

About

This library is for Grove - 3-Axis Digital Accelerometer ±2g to 16g (LIS3DHTR) use I2C or SPI to get acceleration data and temperature.

Releases 5

v1.2.4 (Latest)
on May 5, 2022
+ 4 releases

Seeed_Arduino_LIS3DHTR Build Status

Releases · Seeed-Studio/Seeed_Arduino_LIS3DHTR

github.com/Seeed-Studio/Seeed_Arduino_LIS3DHTR/releases

Seeed-Studio / Seeed_Arduino_LIS3DHTR

Type to search

Code Issues Pull requests Agents Actions Projects Security Insights

Releases Tags Find a release

May 5, 2022 v1.2.4 Latest

lakshanthad v1.2.4 27d7b8f Compare

- Update LIS3DHTR.cpp
- modified SPI connection code to use SPISettings
- Removing Unused variable

▼ Assets 2

Source code (zip) May 5, 2022

Source code (tar.gz) May 5, 2022

Nov 2, 2020 v1.2.3

ansonhex v1.2.3

Only edited library properties for correcting formats for adding to Arduino Library Manager (READY)



Arduino IDE

File

Edit

Sketch

Tools

Help

- Verify/Compile ⌘ R
- Upload ⌘ U
- Configure and Upload
- Upload Using Programmer ⌘ ⌘ U
- Export Compiled Binary ⌘ S
- Optimize for Debugging

- Show Sketch Folder ⌘ K
- Include Library >
- Add File...

- Manage Libraries... ⌘ I

- Add .ZIP Library...

Add the .zip file you just downloaded

New Sketch ⌘ N

New Cloud Sketch ⌘ ⌥ N

Open... ⌘ O

Open Recent >

Sketchbook >

Examples >

Close ⌘ W

Save ⌘ S

Save As... ⌘ ⌥ S

Built-in examples

01.Basics >

02.Digital >

03.Analog >

04.Communication >

05.Control >

06.Sensors >

07.Display >

08.Strings >

09.USB >

10.StarterKit_BasicKit >

11.ArduinoISP >

Examples for Arduino UNO

EEPROM >

Ethernet >

Firmata >

Keyboard >

LiquidCrystal >

SD >

Servo >

SoftwareSerial >

SPI >

Stepper >

TFT >

Wire >

Examples from Custom Libraries

Grove 3-Axis Digital Accelerometer ±2g to 16g (LIS3DHTR) >

Grove Temperature And Humidity Sensor >

U8g2 >

LIS3DHTR_IIC

LIS3DHTR_IIC_AcceleratorReadings

LIS3DHTR_ICC_ADC

LIS3DHTR_ICC_Temperature

LIS3DHTR_SPI

LIS3DHTR_SPI_AcceleratorReadings

LIS3DHTR_SPI_ADC

LIS3DHTR_SPI_Temperature

Check out the example

Change Serial.begin(**115200**) to **9600**

Add **//** at the beginning of

LIS.begin(WIRE)

Delete the **//** at the beginning of

//LIS.begin(WIRE, 0x19)

Your code should look like this **after**
the edits

Verify code and **upload**



```
LIS3DHTR_IIC_AcceleratorReadings.ino
1 // This example shows the 3 axis acceleration.
2 #include "LIS3DHTR.h"
3 #include <Wire.h>
4 LIS3DHTR<TwoWire> LIS; //IIC
5 #define WIRE Wire
6
7 void setup() {
8     //Serial.begin(9600); ←
9     while (!Serial) {};
10    //LIS.begin(WIRE); //IIC init default :0x18 ←
11    LIS.begin(WIRE, 0x19); //IIC init ←
12    delay(100);
13    // LIS.setFullScaleRange(LIS3DHTR_RANGE_2G);
14    // LIS.setFullScaleRange(LIS3DHTR_RANGE_4G);
15    // LIS.setFullScaleRange(LIS3DHTR_RANGE_8G);
16    // LIS.setFullScaleRange(LIS3DHTR_RANGE_16G);
17    // LIS.setOutputDataRate(LIS3DHTR_DATARATE_1HZ);
18    // LIS.setOutputDataRate(LIS3DHTR_DATARATE_10HZ);
19    // LIS.setOutputDataRate(LIS3DHTR_DATARATE_25HZ);
20    LIS.setOutputDataRate(LIS3DHTR_DATARATE_50HZ);
21    // LIS.setOutputDataRate(LIS3DHTR_DATARATE_100HZ);
22    // LIS.setOutputDataRate(LIS3DHTR_DATARATE_200HZ);
23    // LIS.setOutputDataRate(LIS3DHTR_DATARATE_1_6KHZ);
```

Output

Sketch uses 6724 bytes (20%) of program storage space. Maximum is 32256 bytes.

Ln 23, Col 57 Arduino UNO on /dev/cu.usbserial-0001 2

Open Serial Monitor

You should have data printed out

Use the data however you want!

The screenshot shows the Arduino IDE interface. The title bar reads "LIS3DHTR_IIC_AcceleratorReadings | Arduino IDE 2.3.7". The top menu bar includes icons for file operations and a dropdown set to "Arduino UNO". The left sidebar contains project navigation icons. The main code editor window displays the following sketch:

```
1 // This example shows the 3 axis acceleration.
2 #include "LIS3DHTR.h"
3 #include <Wire.h>
4 LIS3DHTR<TwoWire> LIS; //IIC
5 #define WIRE Wire
6
7 void setup() {
8     Serial.begin(9600);
```

Below the code editor is a tab bar with "Output" and "Serial Monitor" selected. The "Serial Monitor" tab shows the following text input field: "Message (Enter to send message to 'Arduino UNO' on '/dev...')". To the right of the input field are buttons for "New Line" and "9600 baud". The main content area of the "Serial Monitor" shows a series of sensor readings:

```
..., -0.01, -0.05, 1.01
..., -0.03, -0.05, 1.02
..., -0.06, -0.04, 1.06
..., -0.06, -0.04, 1.02
..., -0.05, -0.01, 1.02
..., -0.06, -0.05, 0.80
..., -0.04, -0.04, 1.04
..., -0.06, -0.04, 1.01
..., -0.04, -0.04, 1.01
..., -0.08, -0.04, 1.02
..., -0.04, -0.04, 1.01
..., -0.03, -0.03, 1.04
..., -0.06, 0.01
```

At the bottom of the IDE window, status information is displayed: "Ln 23, Col 57" and "Arduino UNO on /dev/cu.usbserial-0001". There are also small icons for volume and window control.