Grove Arduino Prototyping

ELECTRONICS

PHYSICAL COMPUTING

ARDUINO UNO COMPATIBLE

10 PRE-WIRED GROVE MODULES

STEP-BY-STEP TUTORIALS
Grove Beginner for Arduino
Grove Kit: What’s Here?

1. Grove - LED
2. Grove - Buzzer
3. Grove - OLED Display 0.96" (SSD1315)
   → 128×64 dot resolution
4. Grove - Button
   → short press to SELECT
   → long press to EXIT/END
5. Grove - Rotary Potentiometer
   → a rotary dial (e.g. a volume control)
   → use to scroll on OLED display

Seeeduino Lotus
Grove Kit: What’s Here?

- **Grove - Sound Sensor** 7
  - detects sound intensity
- **Grove - Light Sensor** 6
  - detects light intensity
- **Grove - Temperature & Humidity Sensor (DHT11)** 8
- **Grove - 3-Axis Accelerator (LIS3DHTR)** 10
  - detects object’s acceleration
- **Grove - Air Pressure Sensor (BMP280)** 9
  - detects atmospheric pressure
Grove Beginner for Arduino

USING GROVE & ARDUINO CODE

GETTING STARTED
Follow my slides to install Arduino and the USB Driver on your computer, to begin working with Grove and Arduino coding, to fix common errors, and to use ChatGPT for solving coding and hardware issues and modifying and generating Arduino code.
- View videos 1 & 2 on Grove and review the Grove Beginners Kit for Arduino Wiki.
- More in an article and a video on how to use ChatGPT to code Arduino for you!
- If you want to know: my notes on what Arduino code means, line-by-line.

ARDUINO CODE YOU CAN COPY & PASTE
Paste these codes in Arduino! Change values in the code! Modify with ChatGPT!
- A Sound Sensor controls a single LED [code].
- A Button controls a single LED and a Buzzer [code].
- A Potentiometer (i.e., Rotary Angle Sensor) controls a Servo Motor [code].
- An Ultrasonic Sensor controls a Servo Motor and an RGB LED Stick [code].
- A Gesture Sensor controls a Servo Motor and a single LED [code].
- A Light Sensor controls a Servo Motor and an RGB LED Stick; a Switch turns the whole system on/off [code].
BEFORE our next class:

Prototype with Grove!

Show us that you can make work, using Grove, any example code that I provided on our course webpage (or any other one).

What to upload: a link to a video (Vimeo or YouTube) of the working Grove system. Pass or Fail [5 pts] uploaded to our shared folder by the due date on our course webpage.

---

5 points | GROVE DEMO | Example 1
---
Show us that you can make work, using Grove, any example code that I provided on this course webpage under the heading, "Arduino Code You Can Copy & Paste."
What to upload: a Word .doc with a link to a video (Vimeo or YouTube) of the working system, uploaded to our shared folder by the due date shown on this webpage. Name your Word .doc [My Name]-Grove-Demo.doc. Here is an example video: 1. This is Pass or Fail [5 pts]: you receive 5 pts if: [a] we can access your video from the link you provided, [b] your video shows a working system, [c] you named your file correctly, and [d] you submitted your .doc by the deadline. If you fail to meet any of these criteria, you receive a zero.

* Example 1
REMEMBER: we are designing an interactive device: your design will have at least one input and one output using the Grove Kit!

Example 1 from our course webpage

https://arl.human.cornell.edu/images/Input-Output.mp4
1. Connect your new Grove Kit into your computer using the provided USB cord.

2. Use the potentiometer (twist dial) to scroll to the different modules; press the button to select one you want to explore.

3. Interact with the module and see what happens.

4. Press the button longer to exit (with a “beep”); then scroll to the next module to explore.

The Grove Beginner Kit has a plug and plays unboxing demo, where you first plug in the power to the board, you get the chance to experience all the sensors in one go! Use the button and rotary potentiometer to experience each sensor demo!
Install Arduino and the required USB Driver for your OS.

Software Downloads

- **Click on the one for your OS and you’re done.**

- **It will download for you.**

<table>
<thead>
<tr>
<th>Software</th>
<th>Version</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP210x Universal Windows Driver</td>
<td>v11.3.0</td>
<td>9/24/2023</td>
</tr>
<tr>
<td>CP210x VCP Mac OSX Driver</td>
<td>v6.0.2</td>
<td>10/28/2021</td>
</tr>
<tr>
<td>CP210x Windows Drivers</td>
<td>v6.7.6</td>
<td>9/3/2020</td>
</tr>
<tr>
<td>CP210x Windows Drivers with Serial Enumerator</td>
<td>v6.7.6</td>
<td>9/3/2020</td>
</tr>
<tr>
<td>CP210x_5x_AppNote_Archive</td>
<td></td>
<td>9/3/2020</td>
</tr>
</tbody>
</table>

Show 5 more Software
Select your Board & Port

* See next slide if your window doesn’t look like the one below.

- **USB serial**
- **Uno**
- **USB**
Select your Board

1. Open the Arduino IDE on your PC.
2. Click on Tools -> Board -> Arduino AVR Boards -> Arduino Uno to select the correct Development Board Model. Select Arduino Uno as Board.
Select your Port
Try our first example code on your own.

Arduino Code You Can Copy & Paste
Paste these codes in Arduino! Change values in the code! Modify with ChatGPT!

- A Sound Sensor controls a single LED [code].
- A Button controls a single LED and a Buzzer [code].
- A Potentiometer (i.e., Rotary Angle Sensor) controls a Servo Motor [code].
- An Ultrasonic Sensor controls a Servo Motor and an RGB LED Stick [code].
- A Gesture Sensor controls a Servo Motor and a single LED [code].
- A Light Sensor controls a Servo Motor and an RGB LED Stick: a Switch turns the whole system on/off [code].

• A Sound Sensor controls a single LED [code].
Try our first example code on your own.

Sound Sensor to LED

URL to video demo:
https://drive.google.com/file/d/19xwRG4q0Y06yv3syiYYozPFNYY1yet2b/view?usp=sharing

Code:

```c
//The sound sensor detects sounds and lights up an LED

int soundPin = A2; //connect sound sensor to A2
int ledPin = 4; //connect LED to D4
void setup() {
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}
void loop() {
  int soundState = analogRead(soundPin); //read sound sensor’s value
  Serial.println(soundState);
  if (soundState > 200) {
    digitalWrite(ledPin, HIGH); //when a sufficiently loud sound is detected, LED turns on
    delay(10);
  }
  digitalWrite(ledPin, LOW); //when it's quiet (sound ends), LED turns off
}
```

COPY THIS:
Try our first example code on your own.

Sound Sensor to LED

URL to video demo:
https://drive.google.com/file/d/19xwRG4q0Y06yv3syiYYozPFNYY1yet2b/view?usp=sharing

Code:

```cpp
//The sound sensor detects sounds and lights up an LED

int soundPin = A2; //connect sound sensor to A2
int ledPin = 4; //connect LED to D4
void setup() {
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}
void loop() {
  int soundState = analogRead(soundPin); //read sound sensor’s value
  Serial.println(soundState);
  if (soundState > 200) {
    digitalWrite(ledPin, HIGH); //when a sufficiently loud sound is detected, LED turns on
    delay(10);
  }
  digitalWrite(ledPin, LOW); //when it's quiet (sound ends), LED turns off
}
```

COPY THIS:

Tells me which module is connected to which pin.
Paste my code in Arduino window.

- Verify (code)
- Upload (code)
- New (code)
- Open (code)
- Save (code)

**Serial Monitor**

```
// put your setup code here, to run once:
void setup()
{

// put your main code here, to run repeatedly:
void loop()
{

```

**Info and errors shown here**
5. In the upper left corner of the Arduino IDE, there are two buttons: Verify and Upload. First, press the Verify button (✓) to compile. After the compilation is successful, press the upload button (→).
You should be able to light up the red LED by making noise…

Example 1 from our course webpage

https://arl.human.cornell.edu/images/Input-Output.mp4
ERROR!

... or not!

```cpp
//LED Blink
int ledPin = 4;
void setup() {
    pinMode(ledPin, OUTPUT);
}
void loop() {
    digitalWrite(ledPin, HIGH);
    delay(1000);
    digitalWrite(ledPin, LOW);
    delay(1000);
}
```

ERROR!
Error: “No board selected.”
Select your board!

Select Other Board and Port

Select both a Board and a Port if you want to upload a sketch. If you only select a Board you will be able to compile, but not to upload your sketch.

BOARDS
- Arduino Robot Control
- Arduino Robot Motor
- Arduino Uno
- Arduino Uno Mini
- Arduino Uno WiFi

PORTS
- /dev/cu.Bluetooth-Incoming-Port Serial Port
- /dev/cu.JabraElite45h Serial Port
- /dev/cu.usbserial-0001 Serial Port (USB)

Please pick a board connected to the port you have selected.

[OK]
Other Common Errors

You didn’t copy & paste the entire code!

You didn’t connect the right components to the right pins!

You didn’t download the driver!

You have a misplaced or extra { or } in your code!
I can’t figure out what the error is!
(ChatGPT may be your answer!)

I unplugged the display and plugged it back in. Nothing displays on the screen with this code. In the previous version of the code, text was displayed.

I apologize for the confusion. It seems like there might be some issues with the communication or initialization of the display. Before proceeding, please ensure the following:

1. Double-check your connections between the display and the Arduino Seeed Lotus board. Make sure the I2C connections (SCL and SDA) are correctly wired.
2. Confirm that you’re using the correct I2C address for your specific OLED display module. If you’re not sure about the I2C address, refer to the documentation provided by Seeed for the Grove OLED Display 0.96” (SSD1315).
3. Ensure that the display module is working correctly. Sometimes, displays might have manufacturing defects or loose connections.

Once you’ve confirmed the above points, let’s try a simpler code snippet to display the eye pattern:

```cpp
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64

// Replace with your display's I2C address
#define OLED_ADDRESS 0x30

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_ADDRESS);
```
Example prompts for correcting Arduino code using ChatGPT:

Correct the following code: [paste the entire code to be corrected here].

(Correct the following code: `shift > return` // allows you to move to line below in ChatGPT [paste the entire code to be corrected here].)

(You can also past your code into ChatGPT, copy & paste the error code from Arduino, and ask ChatGPT to correct the error.)
It works! / Now try changing values…

Sound Sensor to LED

Code:

```cpp
//The sound sensor detects sounds and lights up an LED

int soundPin = A2; //connect sound sensor to A2
int ledPin = 4; //connect LED to D4

void setup() {
pinMode(ledPin, OUTPUT);
Serial.begin(9600);
}

void loop() {
int soundState = analogRead(soundPin); //read sound sensor’s value
Serial.println(soundState);
if (soundState > 200) {
  digitalWrite(ledPin, HIGH); //when a sufficiently loud sound is detected, LED turns on
  delay(10); //Try changing 10 to 100 and then to 1000. Notice anything different?
} else {
  digitalWrite(ledPin, LOW); //when it's quiet (sound ends), LED turns off
}
LOW = “Off!” (i.e., no light!)
```
Try our 5th example code on your own.

### ARDUINO CODE YOU CAN COPY & PASTE

Paste these codes in Arduino! Change values in the code! Modify with ChatGPT!

- A Sound Sensor controls a single LED [code].
- A Button controls a single LED and a Buzzer [code].
- A Potentiometer (i.e., Rotary Angle Sensor) controls a Servo Motor [code].
- An Ultrasonic Sensor controls a Servo Motor and an RGB LED Stick [code].
- A Gesture Sensor controls a Servo Motor and a single LED [code].
- A Light Sensor controls a Servo Motor and an RGB LED Stick; a Switch turns the whole system on/off [code].

• A Gesture Sensor controls a single LED [code].
The Gesture Sensor controls a servo and a singular LED.

The Gesture Sensor module provides simple motion-based gesture recognition and object tracking. Make sure to orient the Sensor in the correct orientation (i.e., the cable extends to the left and the two mounting holes are oriented towards ceiling and floor. The small camera at the front of the Sensor should be looking straight ahead.)

In gesture mode it can recognize 9 hand moving gestures:

- move up
- move down
- move left [USED IN THIS CODE: turns LED on and rotates servo -120 degrees]
- move right [USED IN THIS CODE: turns LED off and rotates servo +120 degrees]
- circle clockwise
- circle anti-clockwise (counter-clockwise / widdershins)
- forwards / towards
- backwards / away
- wave

The device also counts the quantity of hand waves.

*/

#include <Wire.h> // This library is for I2C communication.
#include "paj7620.h" // This is the library for the Gesture Sensor.
#include <Servo.h> // This is the library for the Servo.

[There's more code in between here on the word doc. Don’t copy code from this slide!]

```c
    case GES_RIGHT_FLAG:
        digitalWrite(LED_pin, LOW); // Turn LED off
        servo_1.write(0); // Return servo to 0 degrees
        break;
    default:
        break;
    }
}

delay(100);
```
Paste my code in Arduino window.

Serial Monitor

- Verify (code)
- Upload (code)
- New (code)
- Open (code)
- Save (code)

Info and errors shown here

Paste here (in place of what’s here; nothing in yellow highlighting should remain).
5. In the upper left corner of the Arduino IDE, there are two buttons, **Verify and Upload**. First, press the Verify button (✓) to compile. After the compilation is successful, press the upload button (→).
ERRORS

Adding Libraries

ERROR!

Missing libraries!
E.g., “No [Servo.h] : No such file or directory.”
The Gesture Sensor controls a servo and a singular LED. 

The Gesture Sensor module provides simple motion-based gesture recognition and object tracking. Make sure to orient the Sensor in the correct orientation (i.e. the cable extends to the left and the two mounting holes are oriented towards ceiling and floor). The small camera at the back of the Sensor is the primary sensor for hand gestures. 

In gesture mode it can recognize 9 hand moving gestures:

- move up
- move down
- move left [USED IN THIS CODE: turns LED on and rotates servo -120 degrees]
- move right [USED IN THIS CODE: turns LED off and rotates servo +120 degrees]
- circle clockwise
- circle anti-clockwise (counter-clockwise / widdershins)
- forwards / towards
- backwards / away
- wave

The device also counts the quantity of hand waves.

*/

#include <Wire.h> // This library is for I2C communication.
#include "paj7620.h" // This is the library for the Gesture Sensor.
#include <Servo.h> // This is the library for the Servo.

Try our 5th example code on your own.

You need to add these “libraries” to make your code work, or you will get an ERROR.

```c
    case GES_RIGHT_FLAG:
        digitalWrite(LED_pin, LOW);  // Turn LED off
        servo_1.write(0);             // Return servo to 0 degrees
        break;
    default:
        break;
    }
}
delay(100);
```
Adding Libraries

To correct this ERROR, you need to add the libraries identified in your code (and again in your ERROR message below it) to your Arduino libraries folder.

For this 5th example, we need the 3 libraries indicated after `#include`: `<Wire.h>`, "paj7620.h", and `<Servo.h>.

The `Wire` and `Servo` libraries can be added by simply following the path shown on the screen shot here:

(The Servo library is added by just clicking on "Servo"; repeat the same by clicking “Wire”.}
Adding Libraries

The “paj7620” library is not on this list -- it’s a less common Arduino module. paj7620 can be added by simply following the path shown on the screenshot here:

Type paj7620 in search

Click
Adding Libraries

Once you’ve added a library for a [Servo], you will always have it loaded for the future. (You only need to add the [Servo] library once.)
ChatGPT

More fun with ChatGPT: add modules, change colors, change speeds, ...

If ChatGPT generated a code for you (or if you pasted one of my example codes into ChatGPT) but now you wanted, in your project, [an OLED display], you (in the same “Chat”) add a prompt such as:

Write code to display the temperature and humidity readings from a DHT22 sensor on a SSD1306 OLED display. DHT is connected to pin D3 of an Arduino Uno. SSD1305 uses I2C.

You could likewise ask ChatGPT to add a pushbutton to your code:

Add a pushbutton connected to pin D4 to the above circuit. Code the pushbutton to toggle the temperature display from Celsius to Fahrenheit.

You could likewise ask ChatGPT to substitute the LED stick for the single [red] LED:

Replace the single LED with a Grove RGB LED Stick 15-WS2813 Mini. If the soundState of the sound sensor is > 200, the LED Stick lights up blue.

This is the way to build-up a complex Arduino code: working from a simple code to a more complex code, step-by-step, by asking for modifications like the above.
ChatGPT

Generate code with ChatGPT:

Generating Arduino code using ChatGPT | Example prompts you type into ChatGPT:

Notes [1]: When ChatGPT has generated the code for you, you can copy the code by simply clicking Copy code in the upper-right of the ChatGPT window.

Note [2]: Note ChatGPT’s “Note” found beneath the window containing the code. This note tells you which pins are used to connect your hardware (e.g., a push button, a light sensor, a servo motor) to your Arduino board, which Arduino libraries need to be installed to run your code, and where to find this library.

Prompt Example 1:
Write code to flash the onboard LED on an Arduino Uno once per second.

Prompt Example 2:
Control the brightness of an LED attached to pin 9 on an Arduino Uno with a potentiometer connected to analog pin A0, using PWM.

Prompt Example 3:
Write code to display the temperature and humidity readings from a DHT22 sensor on the serial monitor. DHT is connected to pin D3 of an Arduino Uno.
What do the pins mean?

D is Digital, A is Analog, I2C is Inter-Integrated Circuit, UART is universal asynchronous receiver/transmitter

D for Digital: 0 or 1 values: e.g., the button

A for Analog: 0 to 1023 values (10 bit); e.g., the potentiometer (aka = rotary angle sensor); the light sensor

I2C for Inter-Integrated Circuit: e.g., the LCD display. NOTE: Unlike with D and A pins, with I2C pins, in your code, you do not have to designate a particular I2C pin for your [LCD display], as in “I2C 3”; simply plug the LCD display into any one of the three I2C pins and the code will find it.

UART for Universal Asynchronous Receiver-Transmitter (i.e., a serial port for connecting the given Arduino board to other Arduino boards or a computer or other computing devices. For more information, see https://docs.arduino.cc/tutorials/nano-every/run-4-uar
What does the code mean?

```cpp
//LED Blink // is the start of a comment (not part of the code; for you!)
//The LED will turn on for one second and then turn off for one second
int ledPin = 4; // the pin # on Arduino board connected to the LED
void setup() { // } setup begins the program
  pinMode(ledPin, OUTPUT); // ; ends the line
}

void loop() { // loop loops (repeats) the function
  digitalWrite(ledPin, HIGH); // ledPIN “HIGH” means 3.3v delivered: “on”
  delay(1000); // delay: how long on or off in ms (milliseconds)
  digitalWrite(ledPin, LOW); // ledPIN “LOW” means 0v delivered: “off”
  delay(1000);
}
```