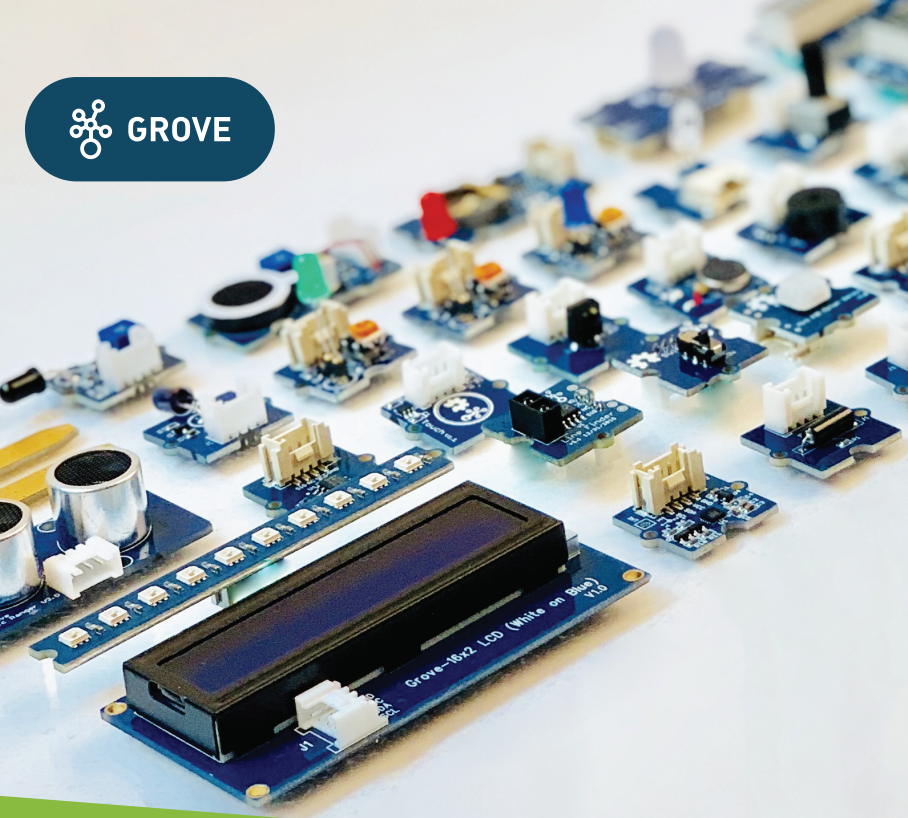
GROVE



Ultimate Guide to

Grove Creator Kit - α/β/γ

Learn, Build and Explore!

www.seeed.cc

Grove - Creator Kit

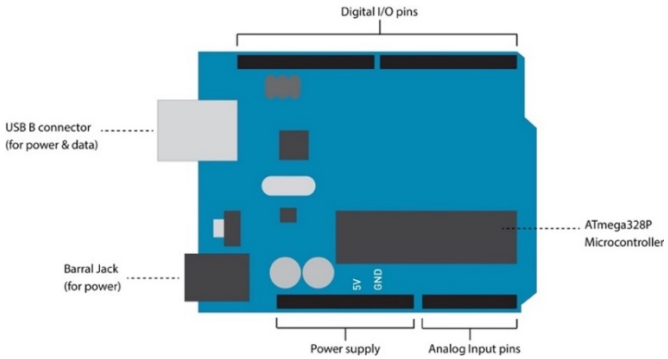
The Grove - Creator Kit is one of the best kits for creators of any age, whether you are complete beginners or have prior experience in electronics. This kit will help you get rid of traditional soldering and complicated wiring and thereby opening doors into the Grove Ecosystem. It supports both Arduino and Arduino compatible boards. Arduino is a development board and it can act as the brain to control all the sensors in your projects. In this book, we have used Seeeduino Lotus (Arduino compatible board) for the example codes to demonstrate the connections while building your projects. Also, this kit includes 40 different Grove modules for a never-ending session of creativity. All you have to do is follow the tutorials, plug the modules into the development board and start your journey!

Contents

What is Arduino?

Arduino is an open source electronics platform that suits well for students, hobbyists and makers to build prototypes very quickly and easily. So, it's basically like a small computer you can program to get things done and interacts through electronic sensors, lights and motors. This makes very complex electronics projects accessible to everyone and therefore will allow them to turn their ideas into a reality.

Arduino has a number of different parts and interfaces combined together on a single board. It comes with different sizes with different chips on-board for different functionalities. The most basic Arduino board is the "Arduino UNO".



Arduino UNO Board

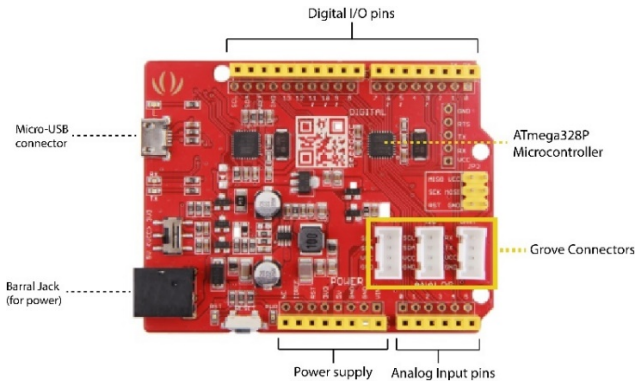
This board has several pins which are used to connect to various components and build your projects. Basically, these pins can be divided into two categories.

- Digital Pins – Used to read and write a single state. 1 or 0, on or off. Arduino UNO has 14 digital I/O pins (input/output)
- Analog Pins – Used to read a range of values which are used for precise control. Arduino UNO has six Analog pins

This board has a main controller chip which is called as a "microcontroller" and it allows you to program the Arduino by executing commands and make decisions based on various inputs. The microcontroller chip used in Arduino UNO is ATmega328P.

Seeeduino

Seeeduino is also a development board just like an Arduino and there are different kinds of Seeeduino boards. The most basic Seeeduino is the Seeeduino V4.2 which is like the Arduino UNO and it is also based on the ATmega328P microcontroller. Seeeduino V4.2 stands out from Arduino UNO because it uses a micro-USB connector for power and data. Also, there are three Grove connectors on-board (2 x I2C and 1 X UART) which allows you to connect all our Grove modules to this board easily (Grove will be introduced later in this book)



Seeeduino V4.2 Board

About Grove

Before we had Grove, at least three wires were needed every time a module had to be connected to an Arduino, including power, signal, and ground.

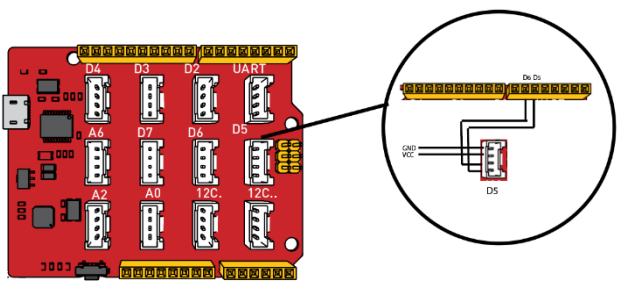
So, Arduino was quite hard to manipulate among excessive wires. How could we simplify this building process? With this goal, we designed and created the Grove system. Grove system includes a number of different modules that has functions such as sensing light, sensing motion etc.

When using Grove modules, you only need one Grove cable to connect between the Grove modules and the Seeeduino/ Base Shield.

Grove cable

Grove Ports and Headers

Also, one thing to note that, the Grove Ports on the Seeeduino Lotus are connected to the corresponding headers internally. So, you will not be able to use a Grove Port and its corresponding header at the same time. However, there is an exception. That is I2C. Multiple I2C devices can be connected to the I2C port and its corresponding header at the same time because I2C devices share the same bus.

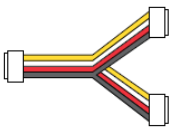


Grove Cables

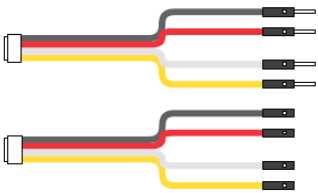
There are four types of Grove cables for different needs. For general projects using Seeeduino or Base Shield, you can simply use the standard cables. For projects where you need to connect two I2C devices by connecting the Grove cable to the I2C port of the main board, you can use the branch cables. If your project uses two servos, you can choose Grove – Branch Cable for Servo. If you don't want to use a Base Shield or a Seeeduino in your projects, you can choose Grove to 4pin Female/Male Jumpers, with which you can connect Grove module to your breadboard directly. Each different kind of cable have five different sizes of length, 5cm, 20cm, 30cm, 40cm, and 50cm.

Standard Cable

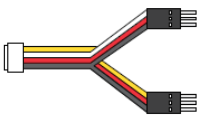
Branch Cable



Grove to 4pin Female/Male Jumper

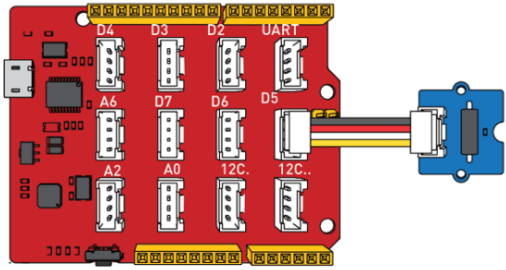


Branch cable for Servo

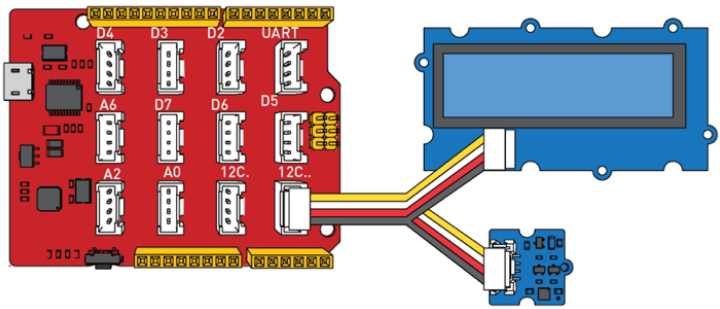


Connections

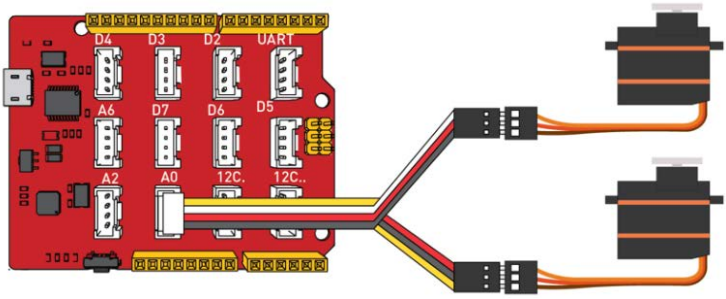
Standard Cable



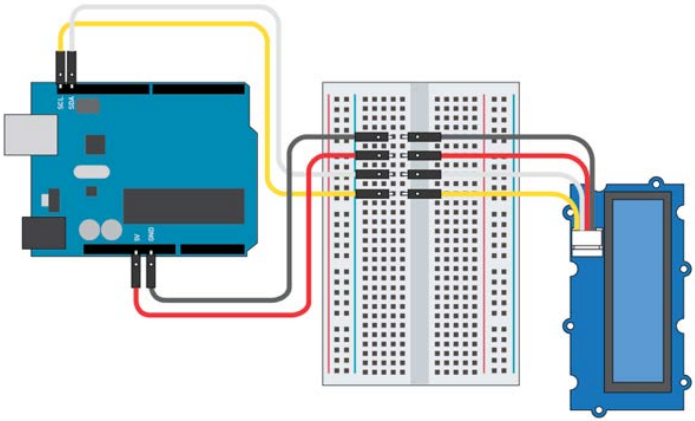
Branch Cable



Branch Cable for Servo



Grove to 4pin Female/Male Jumper



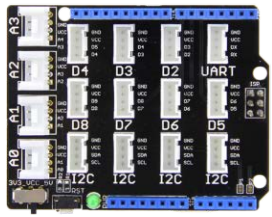
Note

It is recommended to use external power supply for driving servos

Seeeduino Lotus

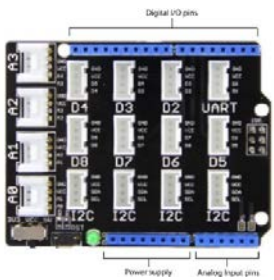
Seeeduino Lotus is a development board with an ATMEGA328 AVR microcontroller and It is the combination of Seeeduino and Grove Base Shield from Seeed Studio.

Seeeduino



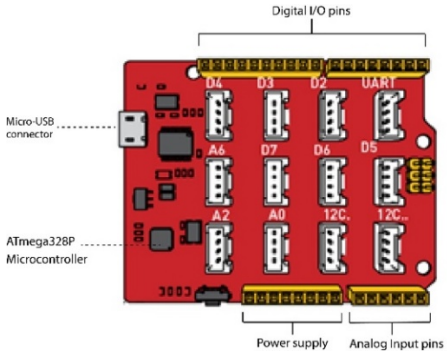
Base Shield

Grove Base Shield is an addon shield to connect on top of the Arduino/Seeeduino boards to expand the amount of Grove ports that can be used for more advanced projects. It has 7 digital ports, 4 Analog ports, 4 I2C ports and 1 UART port for many Grove modules to be connected at once.



Base Shield

Seeeduino Lotus uses an Atmel ATmege328-MU microcontroller and a CP2102N chip. ATmege328-MU is a high performance, low power AVR 8-bit Microcontroller. On the other hand, CP2102N is a USB to Serial converter chip which allows Seeeduino Lotus to communicate with a computer by using a micro-USB cable.



Seeeduino Lotus

Seeeduino Lotus vs Arduino UNO

Seeeduino Lotus has 14 Digital input/output (6 can be used as PWM outputs) and 7 Analog input/output, a micro USB connection, an ICSP header, 12 Grove connectors and a reset button. The Grove connectors on the Seeeduino Lotus are Digital ports (6), Analog ports (3), I2C ports (2), UART (1).

If you know Arduino, Seeeduino Lotus is just like an Arduino, but with added features. Check the chart below for their differences.

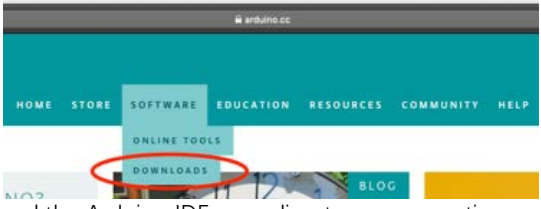
	Seeeduino Lotus V1.1	Arduino UNO R3
Release Date	2018/03	2016/02
Microcontroller	ATMega328P	ATMega328P
Operating Voltage	5V	5V
Flash	32KB	32KB
SRAM	2KB	2KB
EEPROM	1KB	1KB
Power supply interface	Micro USB	USB, DC Port
Grove Connectors	12	None

Software Configuration

After familiarizing with the hardware, let's now setup the software needed to build exciting projects. When you are using an Arduino/ Seeeduino, you need use an IDE to write all your codes and then transfer that code to the boards. For this, you will have to use the "Arduino IDE".

Installing Arduino

1. You will need to download the Arduino IDE in order to write and upload your codes into the development boards. It is open-source, and runs on Windows, Mac OS X and Linux depending on your needs. To download, please visit www.arduino.cc and click **Downloads** under Software section.



2. Download the Arduino IDE according to your operating system and Install. Another website will be re-directed, and simply click "**Just Download**".

3. Install the Arduino IDE following the on-screen installing instructions.

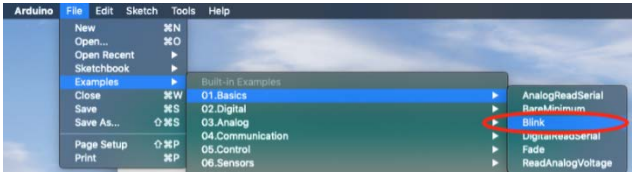
Tips:

For Mac OS X users, there is one more step to do, which is to download and install the CP2102 USB driver so that the Arduino compatible boards can be recognized when connected. To download, please visit: https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers

Programming in Arduino IDE

Now, you can start uploading your very first codes into the development boards. First, you are able to choose from some of the demo codes provided by Arduino IDE to test the correct connections and functionalities of your boards.

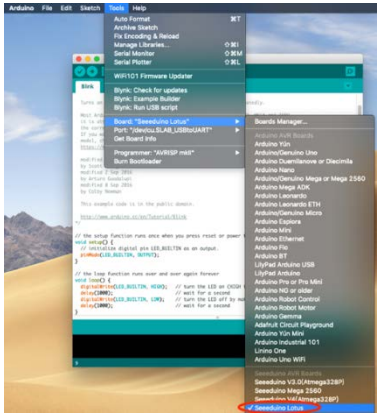
1. To test your first example code, simply click

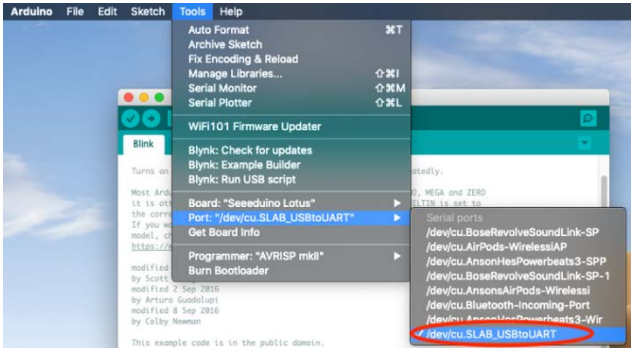


File->Examples->01.Basics->Blink

The first code that you will upload to your development boards will be "Blink" which blinks the on-board LED ON and OFF.

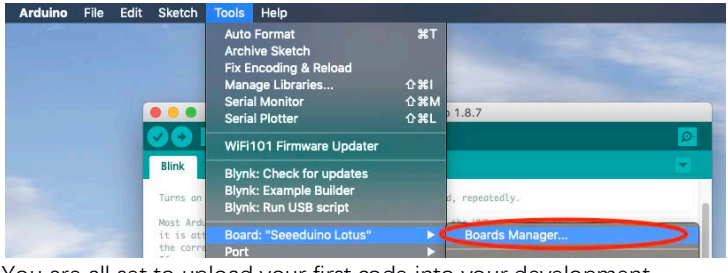
2. To upload, you will need to select the correct board and the correct port. To select the port, click on "Tools->Port:" and choose the one that is connected to your computer. To select the correct board, click on "Tools->Board:" and choose the development boards that you are using. In this case, Seeeduino Lotus is used.





Tips:

To be able to select Seeeduino Lotus as board, you will need to add the Board libraries. To do this, please first go to settings and type in https://raw.githubusercontent.com/Seeed-Studio/Seeed_Platform/master/package_seeeduino_boards_index.json in the **"Additional Boards Managers URLs"** box and click OK. Then, click on **"Tools->Board:->Board Manager"** to open the Boards Manager. Now you can install the Seeeduino board libraries by searching **"Seeeduino AVR"**.



You are all set to upload your first code into your development boards! If everything goes well you should see something like this, and your on-board LED should start blinking!

Basics of Arduino Programming

The structure of Arduino programming is very simple. Arduino programs have a minimum of two blocks. Each block has a set of statements enclosed by curly brackets.

```
void setup()
{
//statement 1;
}
void loop()
{
//statement 2;
}
```

setup():

This function is the preparation block and is called when a sketch starts. It is used to initialize variables, pin modes, start serial communications and etc.

loop():

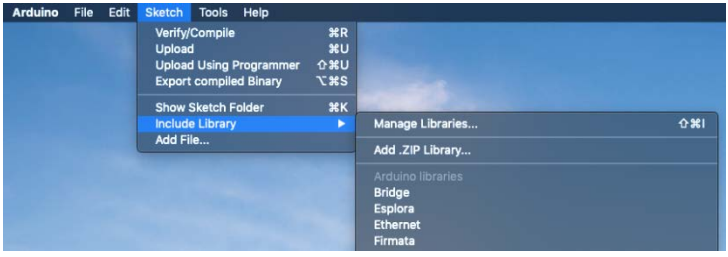
After the setup() function is executed, the execution block **loop()** function runs next. The loop() function executes statements such as reading inputs, trigger outputs and etc. This function will be the base of your code which will run over and over.

Tips:

The void keyword is used only in function declarations. It indicates that the function is expected to return no information when being called.

What are Arduino Libraries and why?

The Arduino environment can be extended through the use of libraries, just like most other programming platforms. Libraries provide extra functionalities for use in sketches, i.e. working with specific hardware or manipulating data. To use a library in a sketch, select it from **Sketch -> Include Library**.



Once this is done, a line of code in the format of #include <xxx.h> will appear on the top of the code to include that library, as shown below. With libraries included, you are able to call pre-defined functions to execute certain commands.

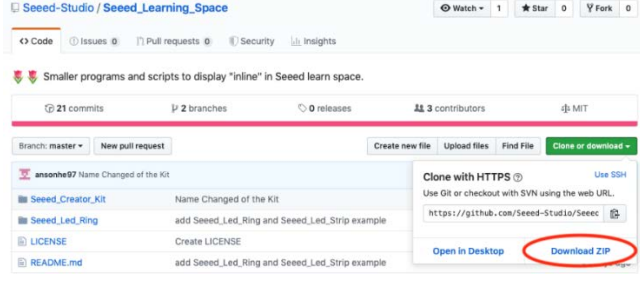
```
#include <rgb_lcd.h>
```

Sketchbook Initialization

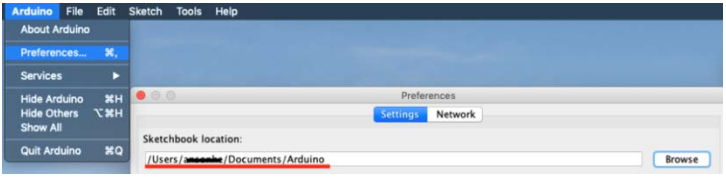
To simplify the coding tasks for you, we packed a few demos of Grove Creator's Kit into a sketchbook file and uploaded it into GitHub. Here is the link to download it.

https://github.com/Seeed-Studio/Seeed_Learning_Space

1. Click on "**Clone or download**" then "**Download ZIP**" to download the packed file.



2. Once the file is downloaded, unzip the file, and check the Sketchbook location of Arduino by clicking "**Arduino->Preferences** (in mac) or "**File->Preferences** (in windows)



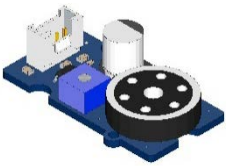
3. Install the sketchbook by copy and pasting the downloaded file into the Sketchbook location. Now, you can access all the demo codes by clicking "**File->Sketch book->Seeed_Creator _Kit**" to explore your creativity!

Modules Introduction

Actuators

Grove - Speaker

Use this module to code your music into the Arduino and DIY your own music box.



Example

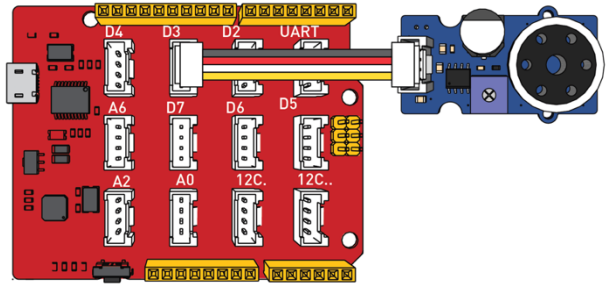
1. Type the following code into the Arduino IDE.

Grove_Speaker

```
/*macro definition of Speaker pin*/
#define SPEAKER 3

int BassTab[]={1911,1702,1516,1431,1275,1136,1012};//bass 1~7

void setup()
{
    pinInit();
}
void loop()
{
        /*sound bass 1~7*/
    for(int note_index=0;note_index<7;note_index++)
    {
        sound(note_index);
        delay(500);
    }
}
void pinInit()
{
    pinMode(SPEAKER,OUTPUT);
    digitalWrite(SPEAKER,LOW);
}
```

```
void sound(uint8_t note_index)
{
    for(int i=0;i<100;i++)
    {
        digitalWrite(SPEAKER,HIGH);
        delayMicroseconds(BassTab[note_index]);
        digitalWrite(SPEAKER,LOW);
        delayMicroseconds(BassTab[note_index]);
    }
}
```

2. Connect the Speaker to **D3** header of Arduino or **D3** Port of Seeeduino Lotus.



3. Hit the **Upload** button

This will output an interesting melody! Also, you can let it output your favorite melody by tinkering with the codes.

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:
File > Sketchbook > Seeed_Creator_Kit > 1.Speaker

Tips

The loudness of this module can be adjusted by the onboard potentiometer. With different input frequencies, the loudspeaker generates different tones.

Grove - Buzzer

The Buzzer can give you interesting sound effects and you will find it fun to play with.

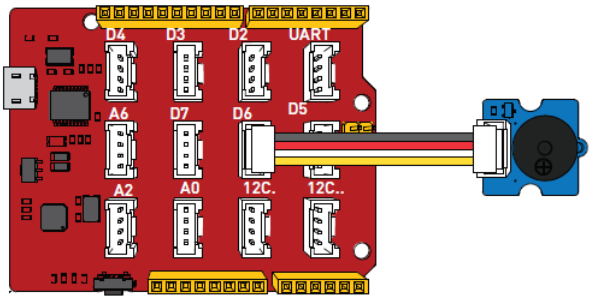
Example

1. Type the following code into the Arduino IDE.

1_digital_buzzer

```
//assign buzzer as pin 6
#define buzzer 6

void setup()
{
  //set buzzer as output
  pinMode(buzzer, OUTPUT);
}

void loop()
{ //turn on buzzer(set logic level high)
  digitalWrite(buzzer, HIGH);
  //wait 1s
  delay(1000);
  //turn off buzzer(set logic level low)
  digitalWrite(buzzer, LOW);
  //wait 1s
  delay(1000);
}
```

2. Connect the Speaker to **D6** header of Arduino or **D6** Port of Seeeduino Lotus.



3. Hit the **Upload** button

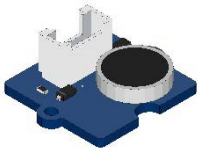
This will make the buzzer beep with a one second interval. However, Grove-Buzzer can be much more fun. It can also play melodies!

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:
File > Sketchbook > Seeed_Creator_Kit > 2. Buzzer

Tips
The sound frequency of this Buzzer will change according to the change in shaking frequency set from the Arduino IDE.

Grove - Vibration Motor

This a mini vibration motor suitable as a non-audible indicator. When the input is HIGH, the motor will vibrate just like your cell phone on silent mode!



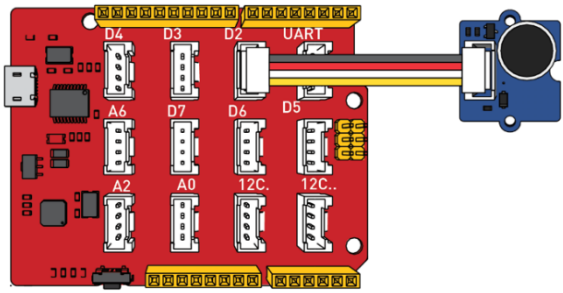
Example

1. Type the following code into the Arduino IDE.

Grove_Vibration_Motor

```
int MoPin = 2;    // vibrator Grove connected to digital pin 9

void setup()  {
    pinMode( MoPin, OUTPUT );
}

void loop()  {

    digitalWrite(MoPin, HIGH);
    delay(1000);

    digitalWrite(MoPin, LOW);
    delay(1000);
}
```

2. Connect the Vibration Motor to **D2** header of Arduino or **D2** Port of Seeeduino Lotus.



3. Hit the **Upload** button

This code will vibrate this module according to the given code. Also, you can tinker with the code and create your own vibration patterns!

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:

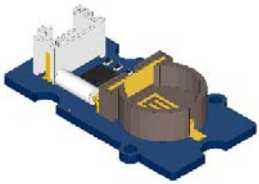
File > Sketchbook > Seeed_Creator_Kit > 3.Vibration Motor

Tips

This motor vibrates when the input is HIGH and stops vibrating when the input is LOW.

Grove - RTC

This module lets you add a real-time clock and a calendar in your projects!



Example

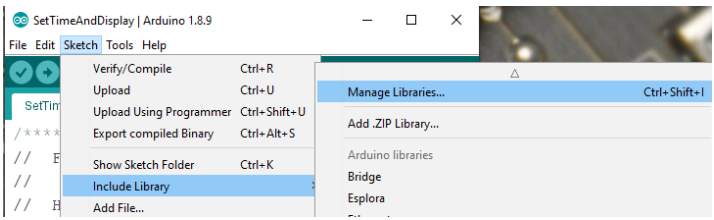
1. Type the following code into the Arduino IDE.

SetTimeAndDisplay §

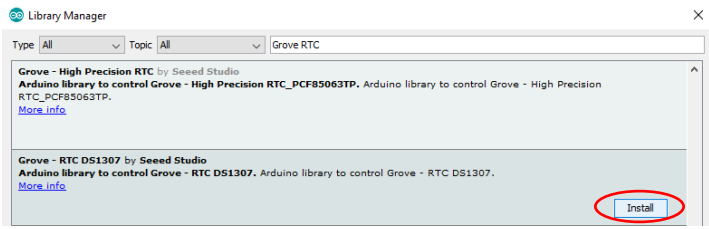
```
#include <Wire.h>
#include "DS1307.h"
DS1307 clock;//define a object of DS1307 class
void setup()
{
  Serial.begin(9600);
  clock.begin();
  clock.fillByYMD(2013, 1, 19); //Jan 19,2013
  clock.fillByHMS(15, 28, 30); //15:28 30"
  clock.fillDayOfWeek(SAT);//Saturday
  clock.setTime();//write time to the RTC chip
}
void loop()
{
  printTime();
}
void printTime()
{
  clock.getTime();
  Serial.print(clock.hour, DEC);
  Serial.print(":");
  Serial.print(clock.minute, DEC);
  Serial.print(":");
  Serial.print(clock.second, DEC);
  Serial.print("  ");
  Serial.print(clock.month, DEC);
  Serial.print("/");
  Serial.print(clock.dayOfMonth, DEC);
  Serial.print("/");
  Serial.print(clock.year + 2000, DEC);
  Serial.print(" ");
  Serial.print(clock.dayOfMonth);
  Serial.print("*");
```

```
switch (clock.dayOfWeek) {
  case MON:
    Serial.print("MON");
    break;
  case TUE:
    Serial.print("TUE");
    break;
  case WED:
    Serial.print("WED");
    break;
  case THU:
    Serial.print("THU");
    break;
  case FRI:
    Serial.print("FRI");
    break;
  case SAT:
    Serial.print("SAT");
    break;
  case SUN:
    Serial.print("SUN");
    break;
}
```

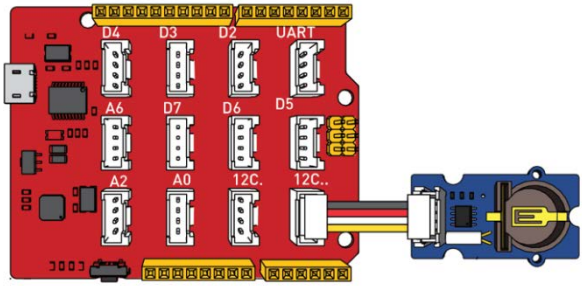
2. Then add the library of "**DS1307**" by following the path below.



3. Type "**Grove RTC**" in the text box and click install.



4. Connect RTC to **SCL, SDA** headers of Arduino or **I2C** Port of Seeeduino Lotus.



5. Hit the **Upload** button

This code will display the time and date on the serial monitor.

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:

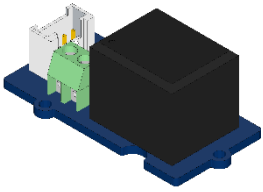
File > Sketchbook > Seeed_Creator_Kit > 4. RTC

Tips

Even though the power to the circuit gets accidentally cut off, you don't need to worry. This RTC retains the date and calendar information if you add a battery.

Grove - Relay

This is a mechanical switch controlled by electrical signals.



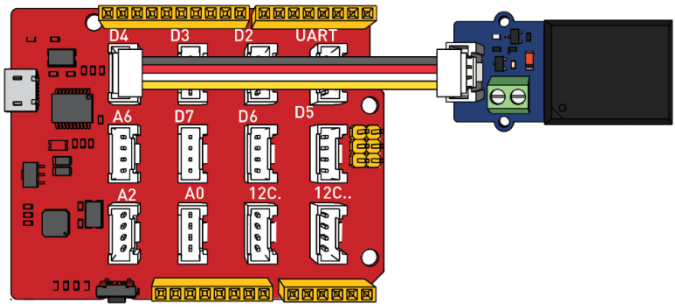
Example

1. Type the following code into the Arduino IDE.

1_relay

```
void setup()
{
  pinMode(4, OUTPUT);
}

void loop()
{
    digitalWrite(4, HIGH);
    delay(1000);
    digitalWrite(4, LOW);
    delay(1000);

}
```

2. Connect the relay to **D4** header of Arduino or **D4** Port of Seeeduino Lotus.



3. Hit the **Upload** button

This code can be used to turn the Relay ON and OFF automatically based on the time interval specified in the code.

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:

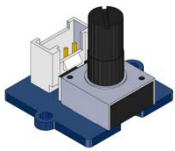
File > Sketchbook > Seeed_Creator_Kit > 5. Relay

Tips

This Relay can be used to control a higher voltage circuit with a much lower voltage.

Grove - Rotary Angle Sensor

This is an input device controlled via turning a knob

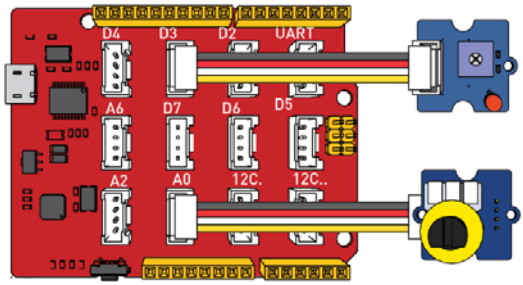


Example

1. Type the following code into the Arduino IDE.

```
1_rotary_angle_sensor_LED §

#define ROTARY_ANGLE_SENSOR A0
#define LED 3  //the Grove - LED is connected to PWM pin D3 of Arduino
#define ADC_REF 5 //reference voltage of ADC is 5v.If the Vcc switch on the seeeduino
                  //board switches to 3V3, the ADC_REF should be 3.3
#define GROVE_VCC 5 //VCC of the grove interface is normally 5v
#define FULL_ANGLE 300 //full value of the rotary angle is 300 degrees

void setup()
{
    Serial.begin(9600);
    pinMode(ROTARY_ANGLE_SENSOR, INPUT);
    pinMode(LED,OUTPUT);
}

void loop()
{
    float voltage;
    int sensor_value = analogRead(ROTARY_ANGLE_SENSOR);
    voltage = (float)sensor_value*ADC_REF/1023;
    float degrees = (voltage*FULL_ANGLE)/GROVE_VCC;
    Serial.println("The angle between the mark and the starting position:");
    Serial.println(degrees);

    int brightness;
    brightness = map(degrees, 0, FULL_ANGLE, 0, 255);
    analogWrite(LED,brightness);
    delay(500);
}
```

2. Connect this sensor to **A0** header of Arduino or **A0** port of Seeeduino Lotus, and connect a Grove – LED to **D3** header of Arduino or **D3** port of Seeeduino Lotus.



3. Hit the **Upload** button

You can use the codes to simply use the rotary angle sensor with an LED module. The brightness of the LED will change according to the angle of rotation.

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:

File > Sketchbook > Seeed_Creator_Kit > ^ Rotary Angle Sensor

Tips

This module is a 10K ohm linear rotary potentiometer with a turning radius range of 300 degrees. It is essentially a slide potentiometer and reflects the position in an analog way.

Grove - Thumb Joystick

Use this module in your next gaming projects to use as a gaming controller.

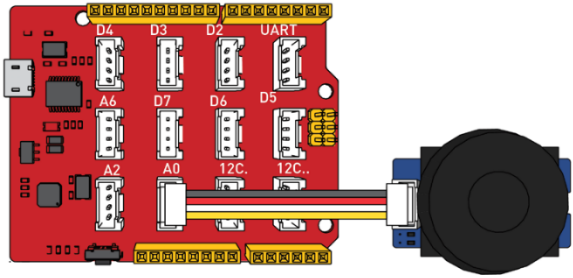
Example

1. Type the following code into the Arduino IDE.

1_thumb_joystick_serial §

```
void setup()
{
    Serial.begin(9600);
}

void loop()
{
    int sensorValue1 = analogRead(A0);
    int sensorValue2 = analogRead(A1);

    Serial.print("The X and Y coordinate is:");
    Serial.print(sensorValue1, DEC);
    Serial.print(",");
    Serial.println(sensorValue2, DEC);
    Serial.println(" ");
    delay(200);
}
```

2. Connect this module to **A0** and **A1** headers of Arduino or **A0** port of Seeeduino Lotus.



3. Hit the **Upload** button

This can be used to output the x & y coordinates and then, later on, translate them to various other applications.

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:

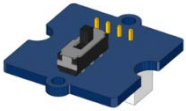
File > Sketchbook > Seeed_Creator_Kit > 7. Thumb Joysitck

Tips

The X and Y axes are two ~10k potentiometers which control 2D movement by generating analog signals. The joystick also has a push button that could be used for special applications.

Grove - Switch(P)

This is a mini SPDT (Single-Pole Single-Throw) slide, also ideal for "ON/OFF" situations, another 'must have' for your Grove prototyping system.



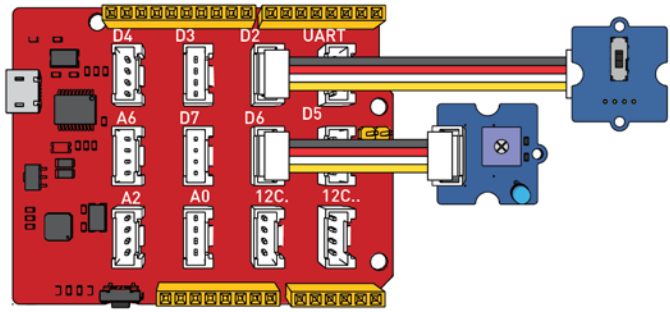
Example

1. Type the following codes into the Arduino IDE.

1_switch_LED

```
const int switchPin = 2;     // the number of the pushbutton pin
const int ledPin =  6;      // the number of the LED pin

int switchState = 0;         // variable for reading the pushbutton status

void setup() {
    // initialize the LED pin as an output:
    pinMode(ledPin, OUTPUT);
    // initialize the switch pin as an input:
    pinMode(switchPin, INPUT);
    Serial.begin(9600);
}

void loop(){
    // read the state of the switch value:
    switchState = digitalRead(switchPin);

    if (switchState == HIGH) {
        //turn LED on:
        digitalWrite(ledPin, HIGH);
        Serial.println("switch high!");
    }
    else {
        //turn LED off:
        digitalWrite(ledPin, LOW);
        Serial.println("switch low");
    }
}
```

2. Connect this module to **D2** header of Arduino or **D2** port of Seeeduino Lotus and connect a Grove-LED to **D6** header of Arduino or **D6** port of Seeeduino Lotus.



3. Hit the **Upload** button

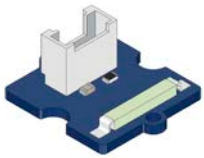
This code can be used to switch ON and OFF an LED. Later on, you can implement this switch into your own design to switch between HIGH and LOW levels.

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:

File > Sketchbook > Seeed_Creator_Kit > 8. Switch

Grove - Magnetic Switch

This is a Grove interface compatible Magnetic switch module (SPDT type), having normally open ruthenium contact. It is a useful module for makers who would like to switch between HIGH and LOW based on the magnetic proximity.



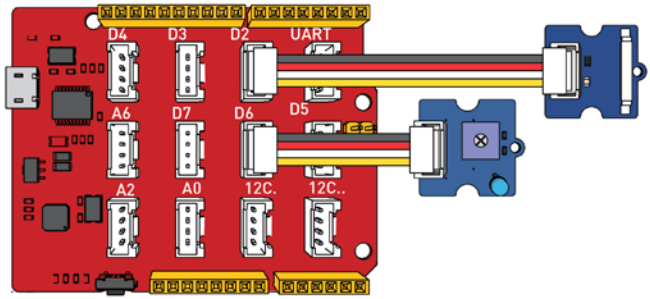
Example

1. Type the following codes into the Arduino IDE (It's the same as the Switch)

1_switch_LED

```
const int switchPin = 2;     // the number of the pushbutton pin
const int ledPin =  6;      // the number of the LED pin

int switchState = 0;         // variable for reading the pushbutton status

void setup() {
    // initialize the LED pin as an output:
    pinMode(ledPin, OUTPUT);
    // initialize the switch pin as an input:
    pinMode(switchPin, INPUT);
    Serial.begin(9600);
}

void loop(){
    // read the state of the switch value:
    switchState = digitalRead(switchPin);

    if (switchState == HIGH) {
        //turn LED on:
        digitalWrite(ledPin, HIGH);
        Serial.println("switch high!");
    }
    else {
        //turn LED off:
        digitalWrite(ledPin, LOW);
        Serial.println("switch low");
    }
}
```

2. Connect this module to **D2** header of Arduino or **D2** port of Seeeduino Lotus and connect a Grove-LED to **D6** header of Arduino or **D6** port of Seeeduino Lotus.



3. Hit the **Upload** button

The above codes can be used to switch ON and OFF an LED when there is a magnet approaching the switch.

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:

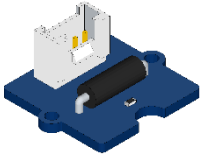
File > Sketchbook > Seeed_Creator_Kit > 9. Magnetic Switch

Tips

This module has wide application areas such as it can be used to build Security alarm sensors, Level sensing systems and plenty more for you to explore!

Grove - Tilt Switch

This switch will turn ON and OFF according to the orientation you hold it.



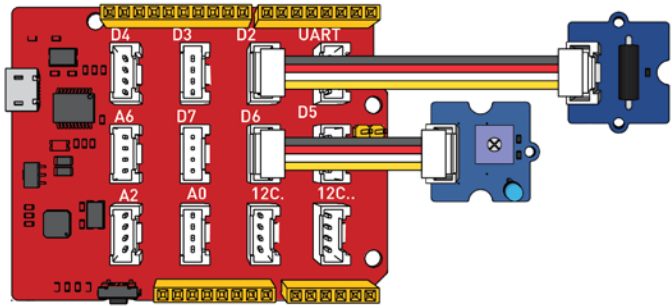
Example

1. Type the following codes into the Arduino IDE

1_switch_LED

```
const int switchPin = 2;     // the number of the pushbutton pin
const int ledPin =  6;      // the number of the LED pin

int switchState = 0;         // variable for reading the pushbutton status

void setup() {
    // initialize the LED pin as an output:
    pinMode(ledPin, OUTPUT);
    // initialize the switch pin as an input:
    pinMode(switchPin, INPUT);
    Serial.begin(9600);
}

void loop(){
    // read the state of the switch value:
    switchState = digitalRead(switchPin);

    if (switchState == HIGH) {
        //turn LED on:
        digitalWrite(ledPin, HIGH);
        Serial.println("switch high!");
    }
    else {
        //turn LED off:
        digitalWrite(ledPin, LOW);
        Serial.println("switch low");
    }
}
```

2. Connect this module to **D2** header of Arduino or **D2** port of Seeeduino Lotus and connect a Grove-LED to **D6** header of Arduino or **D6** port of Seeeduino Lotus.



3. Hit the Upload button

The above codes can be used to switch ON or OFF the LED on the Seeeduino Lotus just by tilting this up or down.

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:

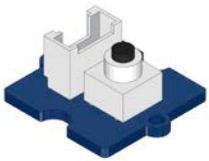
File > Sketchbook > Seeed_Creator_Kit >10. Tilt Switch

Tips

Inside this tilt switch is a pair of balls that make contact with the pins when the case is upright and thus making a connection.

Grove - Button

This is a momentary push button, meaning that it rebounds on its own after it is released. The button itself outputs a HIGH signal when pressed, and LOW when released.



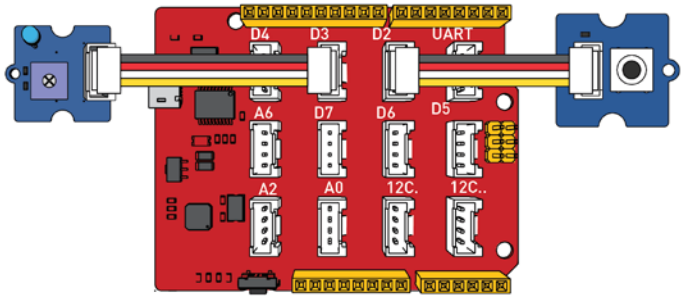
Example

1. Type the following codes into the Arduino IDE

1_button_led

```
const int buttonPin = 2;     // the number of the pushbutton pin
const int ledPin =  3;      // the number of the LED pin

// variables will change:
int buttonState = 0;         // variable for reading the pushbutton status

void setup() {
    // initialize the LED pin as an output:
    pinMode(ledPin, OUTPUT);
    // initialize the pushbutton pin as an input:
    pinMode(buttonPin, INPUT);
}

void loop(){
    // read the state of the pushbutton value:
    buttonState = digitalRead(buttonPin);

    // check if the pushbutton is pressed.
    // if it is, the buttonState is HIGH:
    if (buttonState == HIGH) {
        // turn LED on:
        digitalWrite(ledPin, HIGH);
    }
    else {
        // turn LED off:
        digitalWrite(ledPin, LOW);
    }
}
```

2. Connect this module to **D2** header of Arduino or **D2** port of Seeeduino Lotus and connect a Grove LED to **D3** header of Arduino or **D3** port of Seeeduino lotus.



3. Hit the **Upload** button

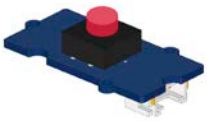
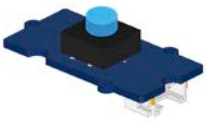
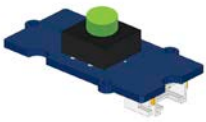
The above codes can be used to turn on the on-board LED on Seeeduino/Arduino.

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:

File > Sketchbook > Seeed_Creator_Kit > 11. Button

Grove - Red/Blue/Green LED Buttons

These modules greatly combine the best of the Grove LED and Grove button. It is very useful to use the LED to indicate the status of the button.

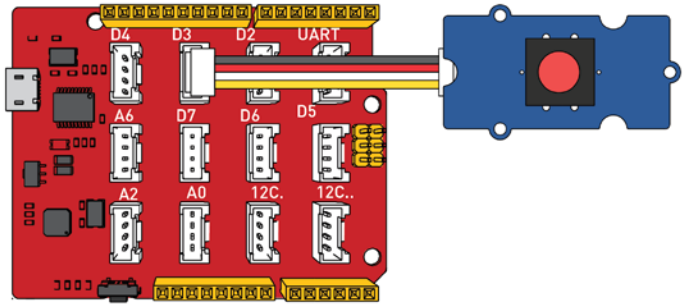
Example

1. Type the following codes into the Arduino IDE

1_led_button §

```
//1: toggle mode, 2: follow mode
#define LED_MODE   1

const int ledPin = 3;      // the number of the LED pin, D3
const int buttonPin = 4;    // the number of the pushbutton pin, D4

void setup() {
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  int reading = digitalRead(buttonPin);
  if (reading != true){
    digitalWrite(ledPin, HIGH);
  }
  else
  {
    digitalWrite(ledPin, LOW);
  }
}
```

2. Connect this module to **D3** and **D4** headers of Arduino or **D3** port of Seeeduino Lotus



3. Hit the **Upload** button

This can demonstrate that the LED lights ON with a fade in/out effect when the button is being pressed.

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:

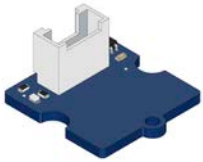
File > Sketchbook > Seeed_Creator_Kit > 12. LED Button

Tips

In this demo, mode 1 which is the toggle mode is chosen. You can change the LED mode to follow mode by simply defining LED_MODE 2.

Grove - Touch Sensor

The Grove Touch Sensor enables you to replace press with simply touch. It can detect the change in capacitance when a finger is nearby, meaning that your finger doesn't need to touch the actual pad but just bring close to the pad and will still output HIGH.



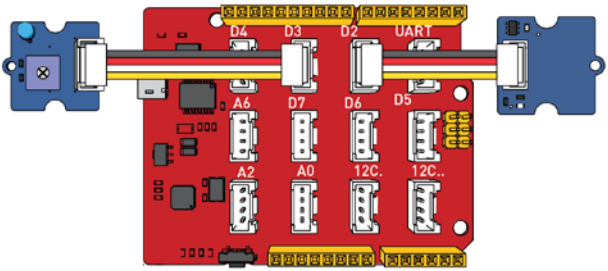
Example

1. Type the following codes into the Arduino IDE

Grove_Touch_Sensor

```
const int TouchPin=2;
const int ledPin=3;

void setup() {
    pinMode(TouchPin, INPUT);
    pinMode(ledPin,OUTPUT);
}

void loop() {
    int sensorValue = digitalRead(TouchPin);
    if(sensorValue==1)
    {
        digitalWrite(ledPin,HIGH);
    }
    else
    {
        digitalWrite(ledPin,LOW);
    }
}
```

2. Connect this module to **D2** header of Arduino or **D2** port of Seeeduino Lotus and connect a Grove LED to **D3** header of Arduino or **D3** port of Seeeduino Lotus.



3. Hit the **Upload** button

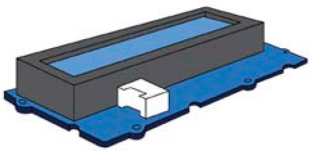
The above codes included to simply light up an LED when the finger is touched on the pad.

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:
File > Sketchbook > Seeed_Creator_Kit > 32. Touch Sensor

Displays

Grove - 16 x 2 LCD (White on Blue)

Use this module to output all your data from sensors connected to your Seeeduino Lotus.



Example

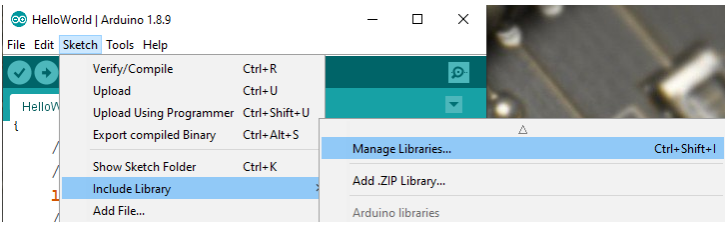
1. Type the following codes into the Arduino IDE

HelloWorld §

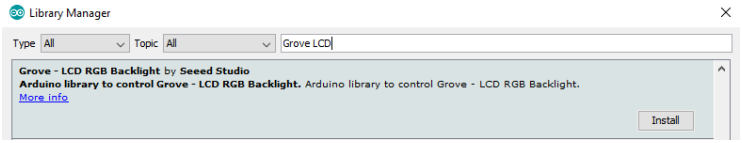
```
#include <Wire.h>
#include "rgb_lcd.h"

rgb_lcd lcd;

const int colorR = 255;
const int colorG = 0;
const int colorB = 0;

void setup()
{
    // set up the LCD's number of columns and rows:
    lcd.begin(16, 2);

    lcd.setRGB(colorR, colorG, colorB);

    // Print a message to the LCD.
    lcd.print("hello, world!");

    delay(1000);
}
```

```
void loop()
{
    // set the cursor to column 0, line 1
    // (note: line 1 is the second row, since counting begins with 0):
    lcd.setCursor(0, 1);
    // print the number of seconds since reset:
    lcd.print(millis()/1000);

    delay(100);
}
```

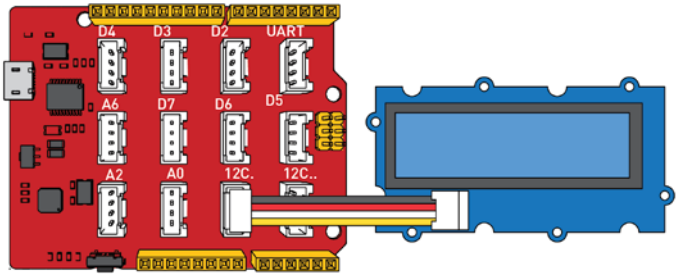
2. Then add the library of "**Grove LCD**" by following the path below.



3. Type "**Grove LCD**" in the text box and click install.



4. Connect this module to **SCL, SDA** headers of Arduino or **I2C** port of Seeeduino Lotus



5. Hit the **Upload** button

You will see that the words “Hello World” will be displayed on the screen.

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:

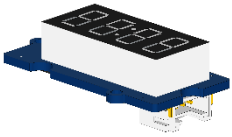
File > Sketchbook > Seeed_Creator_Kit > 13. LCD Display

Tips

Grove - 16 x 2 LCD (White on Blue) display stands out because it takes I2C as communication method with your microcontroller and therefore the number of pins required for data exchange and backlight control shrinks from 10 to 2, leaving I/O’s for other challenging tasks.

Grove - 4-Digit Display

This module can be used in all your projects that require an alpha-numeric display to output information numerically.



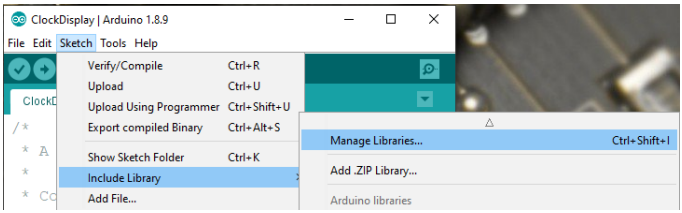
Example

1. Type the following codes into the Arduino IDE

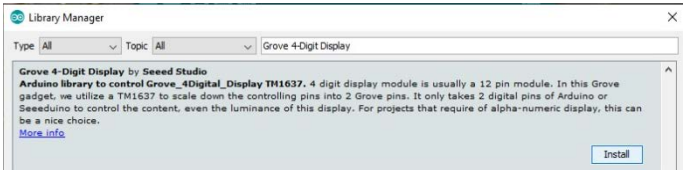
NumberFlow §

```
#include "TM1637.h"
//Include Grove-4 digit display library
#define CLK 2//pins definitions for TM1637 and can be changed to other ports
#define DIO 3
TM1637 tm1637(CLK,DIO);
void setup()
{
  tm1637.init();
  tm1637.set(BRIGHT_TYPICAL);//BRIGHT_TYPICAL = 2,BRIGHT_DARKEST = 0,BRIGHTEST = 7;
}
void loop()
{
  int8_t NumTab[] = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};//0~9,A,b,C,d,E,F
  int8_t ListDisp[4];
  unsigned char i = 0;
  unsigned char count = 0;
  delay(150);
  while(1)
  {
    i = count;
    count ++;
    if(count == sizeof(NumTab)) count = 0;
    for(unsigned char BitSelect = 0;BitSelect < 4;BitSelect ++)
    {
      ListDisp[BitSelect] = NumTab[i];
      i ++;
      if(i == sizeof(NumTab)) i = 0;
    }
    tm1637.display(0,ListDisp[0]);
    tm1637.display(1,ListDisp[1]);
    tm1637.display(2,ListDisp[2]);
    tm1637.display(3,ListDisp[3]);
    delay(300);
  }
}
```

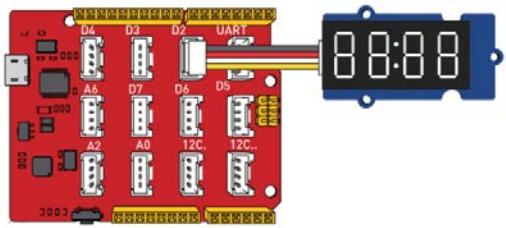
2. Then add the library of "**Grove 4-Digit Display**" by following the path below.



3. Type "**Grove 4-Digit Display**" in the text box and click install.



4. Connect this module to **D2** and **D3** headers of Arduino or **D2** port of Seeeduino Lotus



This code can be used to display the current time.

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:

File > Sketchbook > Seeed_Creator_Kit > 24. Four-Digit Display

LEDs

Grove - Red/Green/Blue/Multi Color Flash LEDs

These Grove LEDs will light up when enough voltage passes through them! These modules can be easily implemented into your prototypes. For example, they can be used as a pilot lamp for indicating power or signal presence.



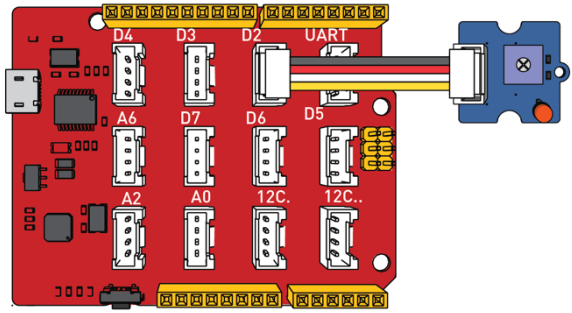
Example

1. Type the following codes into the Arduino IDE.

1_LED

```
void setup() {
  // initialize digital pin2  as an output.
  pinMode(2, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(2, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(2, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

2. Connect this module to **D2** header of Arduino or **D2** port of Seeeduino Lotus



3. Hit the **Upload** button

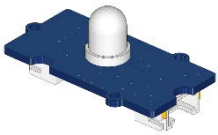
The above codes can be used to turn the LED ON and OFF based on the time interval specified in the code.

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:
File > Sketchbook > Seeed_Creator_Kit > 15. LED

Tips
It has an on-board potentiometer to change the brightness of the LED

Grove – Chainable RGB LED

This RGB LED will allow you to light up your projects by chaining 1024 RGB LEDs at most!



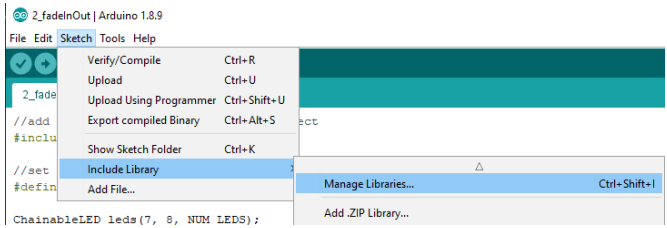
Example

1. Type the following codes into the Arduino IDE

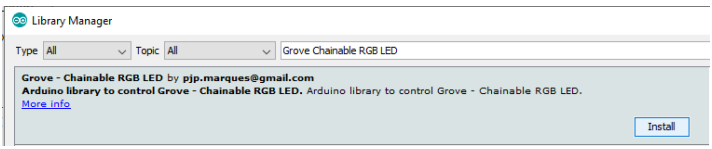
2_fadeInOut §

```
//add ChainableLED library to this project
#include <ChainableLED.h>

//set the number of leds linked to the chain
#define NUM_LEDS  1

ChainableLED leds(7, 8, NUM_LEDS);

void setup()
{
  //initialise ChainableLED leds
  leds.init();
}
//initialise power as byte with value of 0
byte power = 0;

void loop()
{
  /*for loop is used for loop through
    each LED connected to the chain
    in this case there is only one LED
  */
  for (byte i = 0; i < NUM_LEDS; i++)
  {

    if (i % 2 == 0)
      //brighter red color from 0 to full power
      leds.setColorRGB(i, power, 0, 0);
    else
      //dimmer green color from full power to 0
      leds.setColorRGB(i, 0, 255 - power, 0);
  }
  //set power increment as 10
  power += 10;
  //light 0.5s for each brightness
  delay(500);
}
```

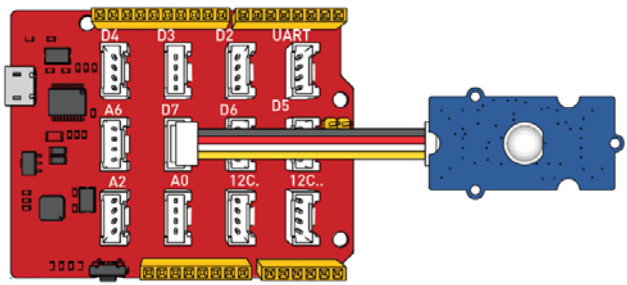
2. Then add the library of "**Grove Chainable RGB LED**" by following the path below.



3. Type "**Grove Chainable RGB LED**" in the text box and click install.



4. Connect this module to **D7** and **D8** headers of Arduino or **D7** port of Seeeduino Lotus



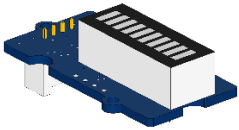
5. Hit the **Upload** button

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:

File > Sketchbook > Seeed_Creator_Kit > 16. Chainable RBG LED

Grove - LED Bar

This is composed of a 10-segment gauge bar and can be used as an indicator for remaining battery life, voltage, water level, volume or other values that require a gradient display. The 10 LED bars in the module include one red, one yellow, one light green and seven green bars.



Example

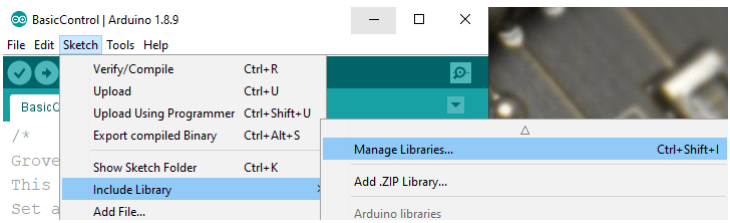
1. Type the following codes into the Arduino IDE.

BasicControl §

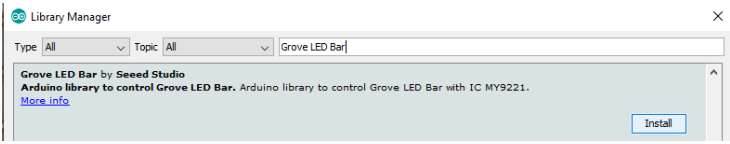
```
#include <Grove_LED_Bar.h>
Grove_LED_Bar bar(7, 6, 0);  // Clock pin, Data pin, Orientation

void setup()
{
  // nothing to initialize
  bar.begin();
}

void loop()
{
  // Turn on all LEDs
  bar.setBits(0x3ff);
  delay(1000);

  // Turn off all LEDs
  bar.setBits(0x0);
  delay(1000);

  // Turn on LED 1
  // 0b0000000000000001 can also be written as 0x1:
  bar.setBits(0b0000000000000001);
  delay(1000);

  // Turn on LEDs 1 and 3
  // 0b0000000000000101 can also be written as 0x5:
  bar.setBits(0b0000000000000101);
  delay(1000);
```

```
  // Turn on LEDs 1, 3, 5, 7, 9
  bar.setBits(0x155);
  delay(1000);

  // Turn on LEDs 2, 4, 6, 8, 10
  bar.setBits(0x2AA);
  delay(1000);

  // Turn on LEDs 1, 2, 3, 4, 5
  // 0b0000000000011111 == 0x1F
  bar.setBits(0b0000000000011111);
  delay(1000);

  // Turn on LEDs 6, 7, 8, 9, 10
  // 0b0000001111100000 == 0x3E0
  bar.setBits(0b0000001111100000);
  delay(1000);
}
```

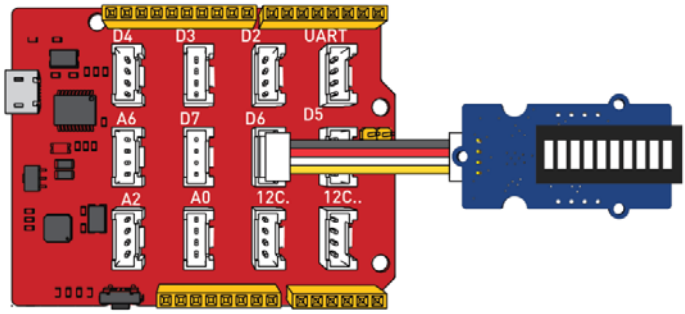
2. Then add the library of "**Grove LED Bar**" by following the path below.



3. Type "**Grove LED Bar**" in the text box and click install.



4. Connect this module to **D6** and **D7** headers of Arduino or **D6** port of Seeeduino Lotus



5. Hit the **Upload** button

The above codes can be used to light up the LEDs sequentially from red to green (bottom to top). For this module, we have packed several demo codes and they can be easily manipulated into your own design for personal needs.

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:

File > Sketchbook > Seeed_Creator_Kit > 17. LED Bar

Grove - RGB LED Stick (10 WS2813 Mini)

This module integrates 10 full-color RGBs on a stick, and with only one signal pin, you can control all 10 LEDs with ease.

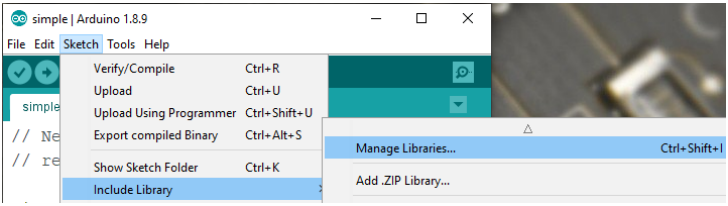
Example

1. Type the following codes into the Arduino IDE.

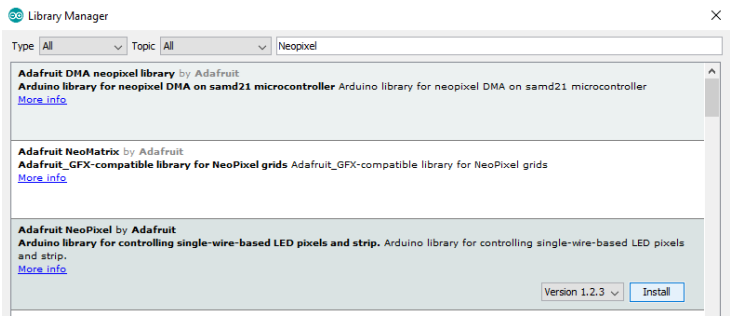
simple §

```
#include "Adafruit_NeoPixel.h"
#ifdef __AVR__
  #include <avr/power.h>
#endif

#define PIN            6
#define NUMPIXELS      10


Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

int delayval = 500; // delay for half a second

void setup() {
#if defined (__AVR_ATtiny85__)
  if (F_CPU == 16000000) clock_prescale_set(clock_div_1);
#endif
  // End of trinket special code
  pixels.setBrightness(255);
  pixels.begin(); // This initializes the NeoPixel library.
}

void loop() {

  for(int i=0;i<NUMPIXELS;i++){

    pixels.setPixelColor(i, pixels.Color(0,150,0));
    pixels.show(); // This sends the updated pixel color to the hardware.

    delay(delayval); // Delay for a period of time (in milliseconds).

  }
}
```

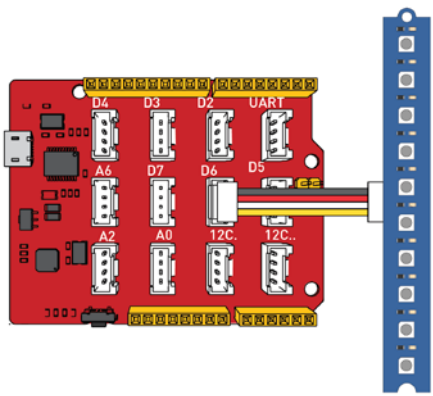
2. Then add the library of "**Grove RGB LED Stick**" by following the path below.



3. Type "**Neopixel**" in the text box and click install.



4. Connect this module to **D6** header of Arduino or **D6** port of Seeeduino Lotus



5. Hit the **Upload** button

You can use the codes above to light up your LED stick and implement this with other Grove input modules for more fun applications!

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:

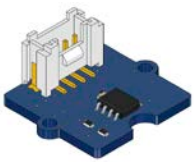
File > Sketchbook > Seeed_Creator_Kit > 18. LED Strip

Tips

All LEDs are WS2813 Mini, and WS2813 support signal break-point continuous transmission, meaning that you can continue to use other LEDs with one LED being broken.

Grove - Temperature Sensor

This module uses a thermistor to detect the surrounding temperature. The resistance of the thermistor increases as the surrounding temperature decreases.



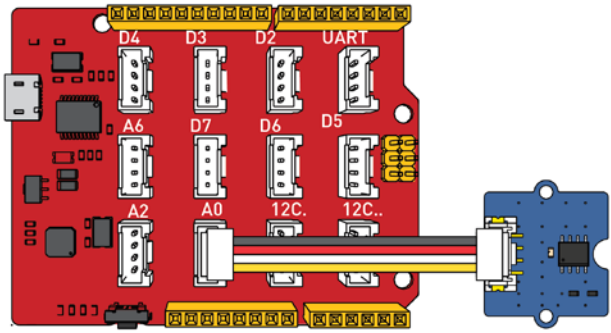
Example

1. Type the following codes into the Arduino IDE.

1_temperature_sensor_serial §

```
#include <math.h>
int a;
float temperature;
int B=3975;                  //B value of the thermistor
float resistance;

void setup()
{
    Serial.begin(9600);
}

void loop()
{
    a=analogRead(0);
    resistance=(float)(1023-a)*10000/a; //get the resistance of the sensor;
    temperature=1/(log(resistance/10000)/B+1/298.15)-273.15;
    delay(1000);
    Serial.print("Current temperature is ");
    Serial.println(temperature);
}
```

2. Connect this module to **A0** header of Arduino or **A0** port of Seeeduino Lotus



3. Hit the **Upload** button

You can use the above codes to simply measure the surrounding temperature and print out onto the serial monitor!

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:

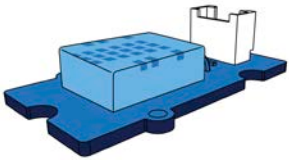
File > Sketchbook > Seeed_Creator_Kit > 19. Temperature Sensor

Tips

Temperature sensors are widely used in different scenarios, for example, it can be used in environmental-related applications!

Grove - Temperature & Humidity Sensor (DHT11)

This module provides a pre-calibrated digital output for both temperature and humidity. A unique capacitive sensor element measures relative humidity and the temperature is measured by a negative temperature coefficient thermistor (NTC).



Example

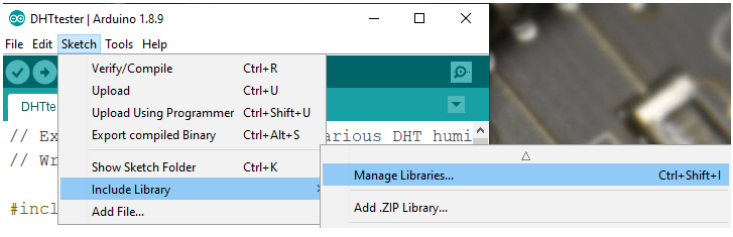
1. Type the following codes into the Arduino IDE.

DHTtester §

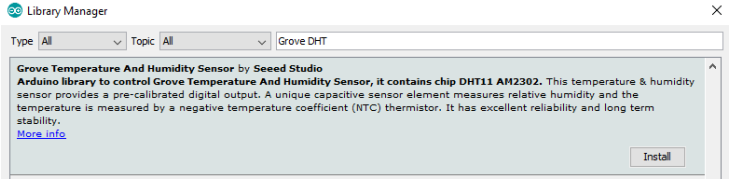
```
#include "DHT.h"
#define DHTPIN 2     // what pin we're connected to

// Uncomment whatever type you're using!
//#define DHTTYPE DHT11   // DHT 11
#define DHTTYPE DHT22   // DHT 22  (AM2302)
//#define DHTTYPE DHT21   // DHT 21 (AM2301)

// Connect pin 1 (on the left) of the sensor to +5V
// Connect pin 2 of the sensor to whatever your DHTPIN is
// Connect pin 4 (on the right) of the sensor to GROUND
// Connect a 10K resistor from pin 2 (data) to pin 1 (power) of the sensor

DHT dht(DHTPIN, DHTTYPE);

void setup()
{

    Serial.begin(9600);
    Serial.println("DHTxx test!");

    /*if using WIO link,must pull up the power pin.*/
    // pinMode(PIN_GROVE_POWER, OUTPUT);
    // digitalWrite(PIN_GROVE_POWER, 1);

    dht.begin();
}
```

```
void loop()
{
    // Reading temperature or humidity takes about 250 milliseconds!
    // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
    float h = dht.readHumidity();
    float t = dht.readTemperature();

    // check if returns are valid, if they are NaN (not a number) then something went wrong!
    if (isnan(t) || isnan(h))
    {
        Serial.println("Failed to read from DHT");
    }
    else
    {
        Serial.print("Humidity: ");
        Serial.print(h);
        Serial.print(" %\t");
        Serial.print("Temperature: ");
        Serial.print(t);
        Serial.println(" *C");
    }
}
```

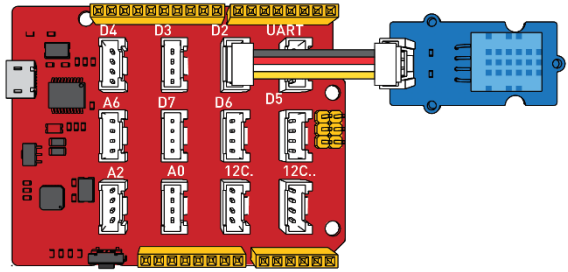
2. Then add the library of "**Grove - Temperature & Humidity Sensor**" by following the path below.



3. Type "**Grove DHT11**" in the text box and click install.



4. Connect this module to **D2** header of Arduino or **D2** port of Seeeduino Lotus.



5. Hit the **Upload** button

The above codes can be used to simply measure the surrounding temperature and humidity and print out onto the serial monitor! This can easily manipulate into your own environmental sensing project.

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:
File > Sketchbook > Seeed_Creator_Kit > 20. Temperature&Humidity Sensor(DHT11)

Tips

Please note that this sensor will not work for temperatures below 0 degrees with its temperature range between 0~50 ºC with ± 2ºC tolerance and humidity range between 20-90%RH with ±5%RH tolerance.

Grove - Moisture Sensor

The Grove Moisture Sensor can be used for detecting the moisture of soil or judge if there is dampness around the sensor. It can be used to decide if the plants in the garden need watering. This can be implemented easily by just inserting the sensor into the soil.

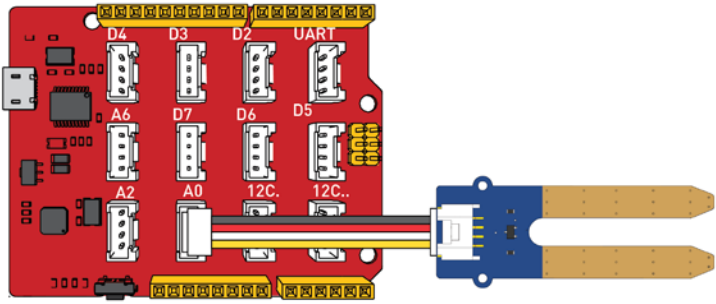
Example

1. Type the following codes into the Arduino IDE

1_moisture_serial

```
int sensorPin = A0;
int sensorValue = 0;

void setup() {
    Serial.begin(9600);
}
void loop() {
    // read the value from the sensor:
    sensorValue = analogRead(sensorPin);
    Serial.print("Moisture = " );
    Serial.println(sensorValue);
    delay(1000);
}
```

2. Connect this module to **A0** header of Arduino or **A0** port of Seeeduino Lotus.



3. Hit the **Upload** button

The above codes can be used to measure the dampness of the soil and this can be easily implemented with a Wi-Fi module for smart gardening solutions.

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:

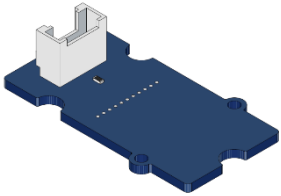
File > Sketchbook > Seeed_Creator_Kit > 21. Moisture Sensor

Tips

This sensor is not hardened against contamination or exposure of the control circuitry to water and may be prone to electrolytic corrosion across the probes. Please use it for prototyping only

Grove - Water Sensor

This module indicates whether the sensor is dry, damp or completely immersed in water by measuring conductivity.

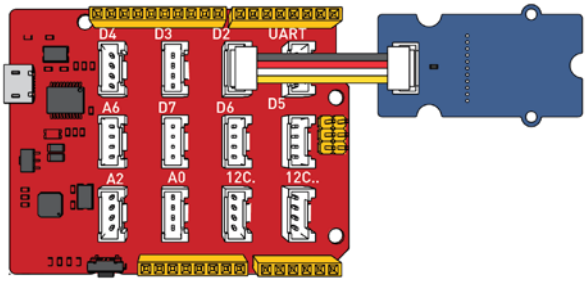


Example

1. Type the following codes into the Arduino IDE

```
1_water_sensor_serial
#define WATER_SENSOR 2

void setup()
{
  Serial.begin (9600);
  pinMode(WATER_SENSOR, INPUT);
}
void loop()
{
  Serial.println(digitalRead(WATER_SENSOR));
  delay(500);
}
```

2. Connect this module to **D2** header of Arduino or **D2** port of Seeeduino Lotus.



3. Hit the **Upload** button

The above codes can be used to simply detect if there is water present. When there is water on the bare conducting wires, the value is LOW. Otherwise, it will be HIGH.

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:

File > Sketchbook > Seeed_Creator_Kit > 22.Water Sensor

Tips

Believe it or not, this circuit can work with the digital I/O pins of your Arduino or you can use it with the analog pins to detect the amount of water induced contact between the grounded and sensor traces.

Grove - Sound Sensor

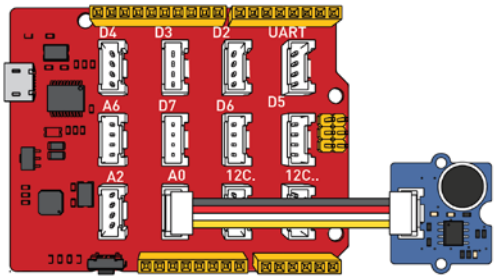
The Grove – Sound Sensor can detect the sound intensity of the environment. This module's output is analog and can be easily sampled and tested using a Seeeduino/ Arduino.

Example

1. Type the following codes into the Arduino IDE.

```
1_sound_sensor_serial §
const int pinAdc = A0;

void setup()
{
    Serial.begin(115200);
    //Serial.println("Grove - Sound Sensor Test...");
}

void loop()
{
    long sum = 0;
    for(int i=0; i<32; i++)
    {
        sum += analogRead(pinAdc);
    }

    sum >>= 5;

    Serial.println(sum);
    delay(10);
}
```

2. Connect this module to **A0** header of Arduino or **A0** port of Seeeduino Lotus.



3. Hit the **Upload** button

The above code can be used to analog output the sound intensity of the surrounding, simple as that!

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:

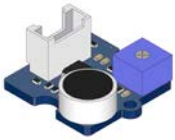
File > Sketchbook > Seeed_Creator_Kit > 23. Sound Sensor

Tips

This module brings a lot of creativity, for example, it can be used with LED strips to design LED lighting effects according to the sound!

Grove - Loudness Sensor

The Grove Loudness Sensor is used to detect the sound of the environment. It amplifies and filters the high frequency signals that was received from the microphone, and outputs a positive envelope.



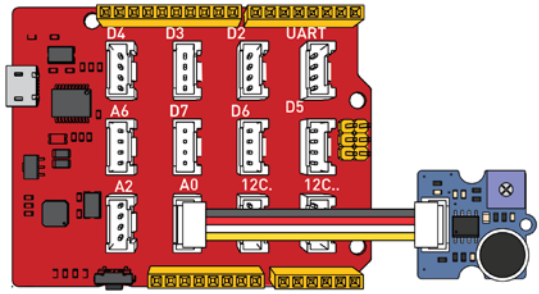
Example

1. Type the following codes into the Arduino IDE

1_loudness_serial

```
int loudness;

void setup()
{
    Serial.begin(9600);
}

void loop()
{
    loudness = analogRead(0);
    Serial.println(loudness);
    delay(200);
}
```

2. Connect this module to **A0** header of Arduino or **A0** port of Seeeduino Lotus.



3. Hit the **Upload** button

The codes above can be used to analog output the loudness of the surrounding!

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:

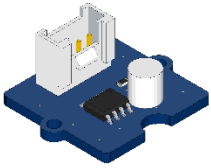
File > Sketchbook > Seeed_Creator_Kit > 24. Loudness Sensor

Tips

To avoid unnecessary signal disturbances, input signal will go through two times, filtering inside the module. There is a screw potentiometer that enables manual adjustment of the output gain.

Grove - Light Sensor v1.2

The Grove Light Sensor integrates a photo-resistor (light dependent resistor) to detect the intensity of light. The resistance of photo-resistor decreases when the intensity of light increases.



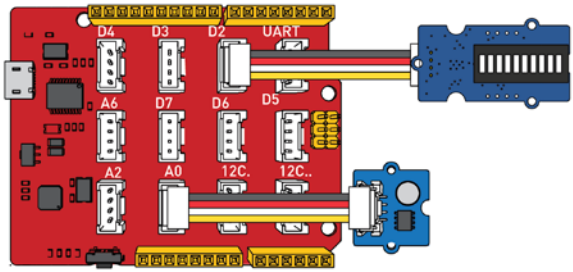
Example

1. Type the following codes into the Arduino IDE.

1_lightsensor §

```
#include <Grove_LED_Bar.h>

Grove_LED_Bar bar(3, 2, 0);  // Clock pin, Data pin, Orientation

void setup()
{
  // nothing to initialize
  bar.begin();
  bar.setGreenToRed(true);
}

void loop()
{

  int value = analogRead(A0);
  value = map(value, 0, 800, 0, 10);

  bar.setLevel(value);
  delay(100);
}
```

2. Connect this module to **A0** header of Arduino or **A0** port of Seeeduino Lotus and connect a Grove – LED Bar to **D2** and **D3** headers of Arduino or **D2** port of Seeeduino.



3. Hit the **Upload** button

The above codes are used to demonstrate how this module work with other modules. In this demo, we used an LED Bar, thus the LED Bar will change based on the surrounding light. Find out more fun applications yourself!

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:

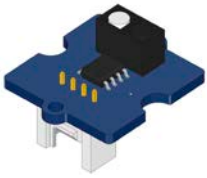
File > Sketchbook > Seeed_Creator_Kit > 25. Light Sensor v1.2

Tips

This module can be used to build a light-controlled switch i.e. switch off lights during day time and switch on lights during night time.

Grove - Line Finder v1.1

The Grove Line Finder is designed for line-following robots! It has an IR emitting LED and an IR sensitive phototransistor. It outputs a digital signal to the microcontroller and the robot can follow a black line on white background.



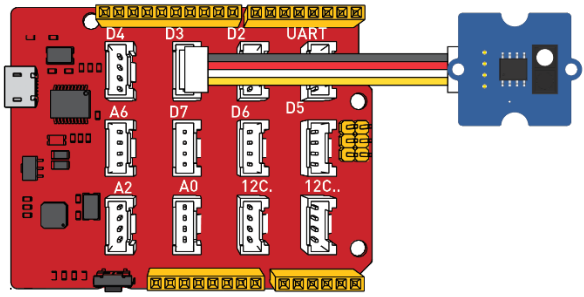
Example

1. Type the following codes into the Arduino IDE.

1_LineDetectorSerial

```
//naming pin3 as singalPin
#define signalPin 3

void setup() {
  // initialize the digital pin as an input:
  pinMode(signalPin, INPUT);
  // opens serial port, sets data rate to 9600 bps
  Serial.begin(9600);
}

void loop() {
  //read the line detector input
  int val = digitalRead(signalPin);

  //display the line detector status, 1 is black, 0 is white.
  Serial.println(val);
}
```

2. Connect this module to **D3** header of Arduino or **D3** port of Seeeduino Lotus.



3. Hit the **Upload** button

This demo codes simply demonstrate how this can be implemented for different projects. It indicates black when the sensor is on top of black lines and white on white area. With several Line Finder modules and servos, you will be able to build a simple Line Following robot!

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:

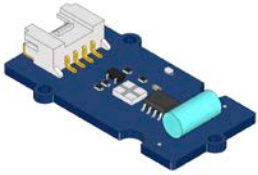
File > Sketchbook > Seeed_Creator_Kit > 26. Line Finder v1.1

Tips

This module is perfect for makers who would like to build their first robot project!

Grove - Vibration Sensor (SW-420)

The Grove Vibration Sensor (SW-420) is a high sensitivity non-directional vibration sensor. When the module is stable, the circuit is turned on and the output is high. When the movement or vibration occurs, the circuit will be briefly disconnected and output LOW.



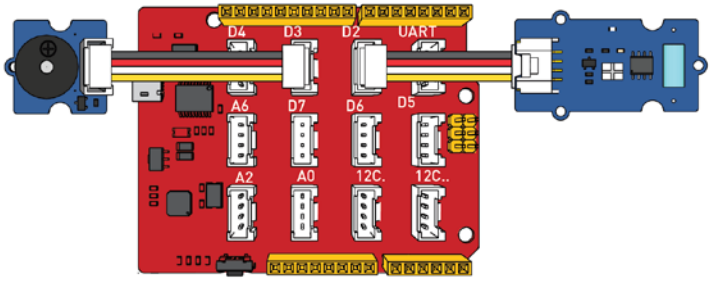
Example

1. Type the following codes into the Arduino IDE.

1_vibration_sensor_buzzer §

```
// constants won't change. They're used here to set pin numbers:
const int vibrationPin = 2;     // the number of the vibration pin
const int buzzer =  3;      // the number of the buzzer pin

// variables will change:
int state = 0;          // variable for reading the viabration sensor status

void setup() {
  // initialize the buzzer  as an output:
  pinMode(buzzer, OUTPUT);
  // initialize the viabraiton as an input:
  pinMode(vibrationPin, INPUT);
}

void loop() {
  // read the state of the viabration value:
  state = digitalRead(vibrationPin);

  if (state == HIGH) {
    // turn Buzzer off:
    digitalWrite(buzzer, LOW);
  } else {
    // turn Buzzer on:
    digitalWrite(buzzer, HIGH);
  }
}
```

2. Connect this module to **D2** header of Arduino or **D2** port of Seeeduino Lotus and connect a Grove-Buzzer to **D3** header of Arduino or **D3** port of Seeeduino Lotus.



3. Hit the **Upload** button

The codes can be used to build a simple security alarm, that produce high frequency buzz when there is a movement.

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:

File > Sketchbook > Seeed_Creator_Kit > 27. Vibration Sensor

Tips

You can use a screwdriver to rotate the potentiometer which controls the sensitivity of the sensor!

Grove – Mini PIR Motion Sensor

The Grove Mini PIR Motion allows to sense human movement with its range. It is widely used for making motion detecting applications such as alarm burglar systems, visitor presence monitoring, light switches and robots.

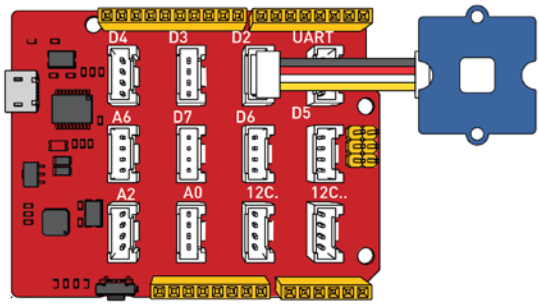
Example

1. Type the following codes into the Arduino IDE.

1_mini_pir_motion

```
/*macro definitions of PIR motion sensor pin and LED pin*/
#define PIR_MOTION_SENSOR 2//Use pin 2 to receive the signal from the module


void setup()
{
    pinMode(PIR_MOTION_SENSOR, INPUT);
    Serial.begin(9600);

}

void loop()
{
    if(digitalRead(PIR_MOTION_SENSOR))//if it detects the moving people?
        Serial.println("Hi,people is coming");
    else
        Serial.println("Watching");

 delay(200);
}
```

2. Connect this module to **D2** header of Arduino or **D2** port of Seeeduino Lotus.



3. Hit the **Upload** button.

The codes can be used to simply to detect whether there is human movement in front of the sensor.

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:

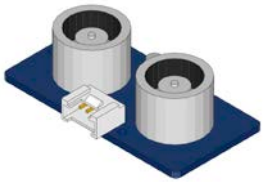
File > Sketchbook > Seeed_Creator_Kit > 28. Mini PRI Motion Sensor

Tips

Please note that the orientation can influence the sensitivity, for more accurate detection, please place the sensor horizontally!

Grove - Ultrasonic Distance Sensor

The Grove Ultrasonic Distance Sensor is a non-contact distance measurement module which works at 40Khz. It has a measuring range of 2-350cm with 1cm resolution.



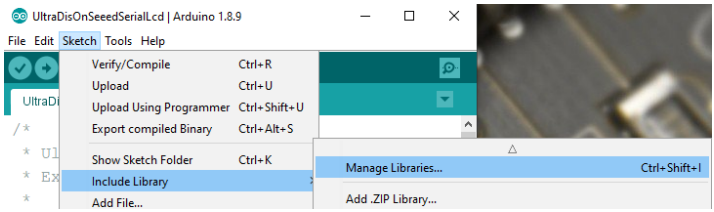
Example

1. Type the following codes into the Arduino IDE.

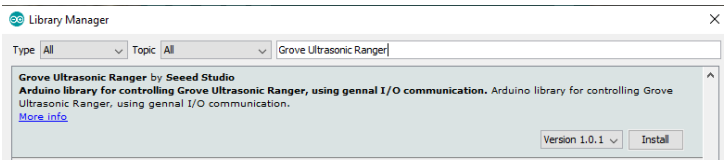
Ultrasonic_ranger_serial

```
#include "Ultrasonic.h"

Ultrasonic ultrasonic(7);
void setup()
{
    Serial.begin(9600);
}
void loop()
{
    long RangeInInches;
    long RangeInCentimeters;

    Serial.println("The distance to obstacles in front is: ");
    RangeInInches = ultrasonic.MeasureInInches();
    Serial.print(RangeInInches);//0~157 inches
    Serial.println(" inch");
    delay(250);

    RangeInCentimeters = ultrasonic.MeasureInCentimeters(); // two measurements
    Serial.print(RangeInCentimeters);//0~400cm
    Serial.println(" cm");
    delay(250);
}
```

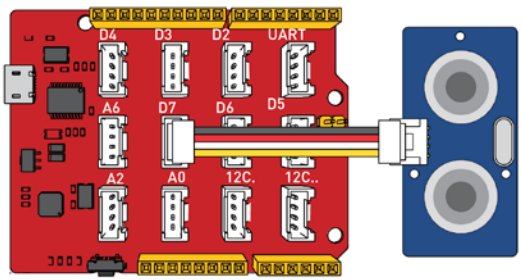
2. Then add the library of "**Grove - Ultrasonic Distance Sensor**" by following the path below.



3. Type "**Grove Ultrasonic Ranger**" in the text box and click install.



4. Connect this module to **D7** header of Arduino or **D7** port of Seeeduino Lotus.



5. Hit the **Upload** button

The codes can be used to measure the distance to the obstacles in front.

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook

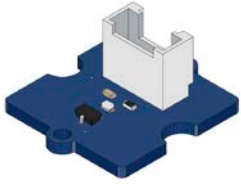
File > Sketchbook > Seeed_Creator_Kit > 29. Ultrasonic Sensor

Tips

This module can work well with actuators such as buzzer and vibration motor for 'Smart solutions'.

Grove - Hall Sensor

The Grove Hall Sensor is based on Hall Effect, which is the production of a potential difference across an electrical conductor, transverse to an electric current in the conductor and a magnetic field perpendicular to the current. This module can be used to build products such as RPM meter, simple DC motor and more!



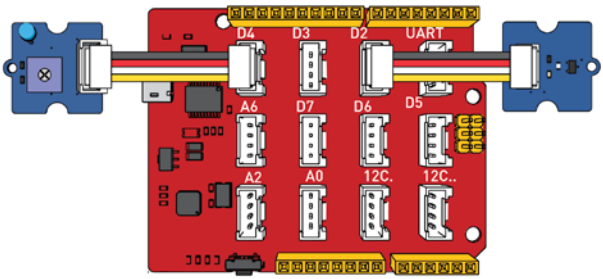
Example

1. Type the following codes into the Arduino IDE.

MagnetControlLED §

```
#define HALL_SENSOR 2
#define LED 4//the Grove - LED is connected to D4 of Arduino

void setup()
{
  pinsInit();
}

void loop()
{
  if(isNearMagnet())//if the hall sensor is near the magnet?
  {
    turnOnLED();
  }
  else
  {
    turnOffLED();
  }
}
void pinsInit()
{
  pinMode(HALL_SENSOR, INPUT);
  pinMode(LED,OUTPUT);
}
```

2. Connect this module to **D2** header of Arduino or **D2** port of Seeeduino Lotus and connect a Grove-LED to **D4** header of Arduino or **D4** port of Seeeduino Lotus.



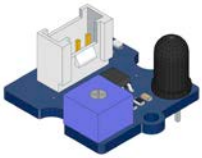
3. Hit the **Upload** button

The codes can be used to demonstrate when a magnet whose south pole is facing up approaching the onboard sensor, the LED will be turned on. Otherwise, the LED will be off.
If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:

File > Sketchbook > Seeed_Creator_Kit > 30. Hall Sensor

Grove - Flame Sensor

The Grove Flame Sensor can be used to detect fire source or other light sources of the wavelength in the range of 760nm - 1100nm.



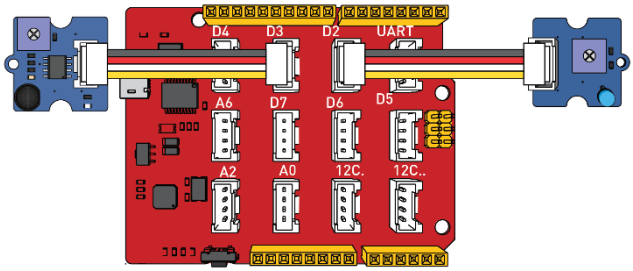
Example

1. Type the following codes into the Arduino IDE

FlameDetector §

```
#define FLAME_SENSOR 3 //connect FLAME_SENSOR to digital pin3
#define LED 2//connect Grove - LED to pin2

void setup()
{
  pinsInit();
}
void loop()
{
  if(isFlameDetected())
    turnOnLED();
  else turnOffLED();
}
/*******************************/
void pinsInit()
{
  pinMode(FLAME_SENSOR, INPUT);
  pinMode(LED,OUTPUT);
  digitalWrite(LED,LOW);
}
void turnOnLED()
{
  digitalWrite(LED,HIGH);
}
void turnOffLED()
{
  digitalWrite(LED,LOW);
}
boolean isFlameDetected()
{
  if(digitalRead(FLAME_SENSOR))
    return false;
  else return true;
}
```

2. Connect this module to **D3** header of Arduino or **D3** port of Seeeduino Lotus and connect a Grove-LED to **D2** header of Arduino or **D2** port of Seeeduino Lotus.



3. Hit the **Upload** button

The above codes can be used to detect if there is visible flame near the module. The module itself detects the infrared light and outputs digital 0 and 1 through a comparator output.

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:

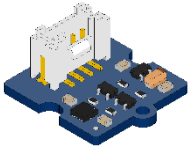
File > Sketchbook > Seeed_Creator_Kit > 31. Flame Sensor

Tips

The sensitivity is adjustable by the precision potentiometer on board.

Grove - 3-Axis Digital Accelerometer(±1.5g)

The Grove - 3-Axis Digital Accelerometer(±1.5g) can be used for sensing data changes, product orientation, and gesture detection through an interrupt pin. It is a very low power, low profile capacitive MEMS sensor.



Example

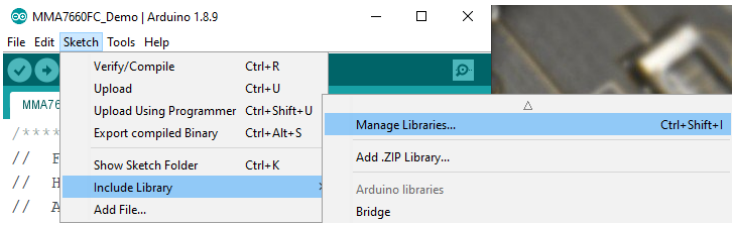
1. Type the following codes into the Arduino IDE

MMA7660FC_Demo §

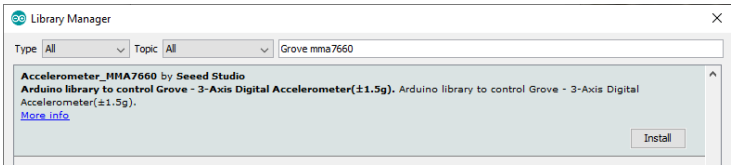
```
#include <Wire.h>
#include "MMA7660.h"
MMA7660 accelemeter;
void setup()
{
  accelemeter.init();
  Serial.begin(9600);
}
void loop()
{
  int8_t x;
  int8_t y;
  int8_t z;
  float ax,ay,az;
  accelemeter.getXYZ(&x,&y,&z);

  Serial.print("x = ");
    Serial.println(x);
    Serial.print("y = ");
    Serial.println(y);
    Serial.print("z = ");
    Serial.println(z);
```

```
  accelemeter.getAcceleration(&ax,&ay,&az);
    Serial.println("accleration of X/Y/Z: ");
  Serial.print(ax);
  Serial.println(" g");
  Serial.print(ay);
  Serial.println(" g");
  Serial.print(az);
  Serial.println(" g");
  Serial.println("*************");
  delay(500);
}
```

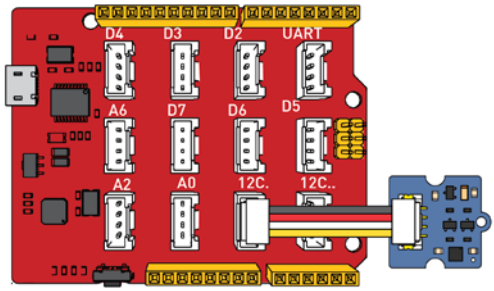
2. Then add the library of "**Grove - 3-Axis Digital Accelerometer**" by following the path below.



3. Type "**Grove mma7660**" in the text box and click install.



4. Connect this module to **SCL, SDA** headers of Arduino or **I2C** Port of Seeeduino Lotus.



5. Hit the **Upload** button

The above code codes can be used determine the orientation of the module, as the outputs provide the raw data as well as the 3-axis acceleration information converted into the unit of gravity, "g".

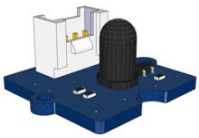
If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:
File > Sketchbook > Seeed_Creator_Kit > 3-Axis Digital Accelerometer

Tips
This module can be used in applications such as Motion detection of Robot, mobile phone and etc.

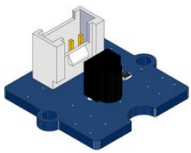
Grove - Infrared Emitter/Infrared Receiver

The Grove - Infrared Emitter is used to transmit infrared signals through an infrared LED, whereas the Grove – Infrared Receiver is used to receive the signals on the other side. The reliable range for this module can be up to 10 meters.

Infrared Receiver



Infrared Emitter



Example

1. Type the following codes into the Arduino IDE.

recv §

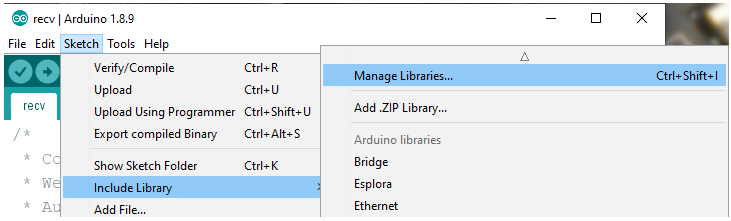
```
#include <IRSendRev.h>
#define BIT_LEN         0
#define BIT_START_H     1
#define BIT_START_L     2
#define BIT_DATA_H      3
#define BIT_DATA_L      4
#define BIT_DATA_LEN    5
#define BIT_DATA        6
const int pinRecv = 2;              // ir receiver connect to D2
void setup()
{
  Serial.begin(115200);
  IR.Init(pinRecv);
  Serial.println("init over");
}
unsigned char dta[20];
void loop()
{
  if (IR.IsDta())                 // get IR data
  {
    IR.Recv(dta);                 // receive data to dta
```

```
    Serial.println("+-----------------------------------------------------+");
    Serial.print("LEN = ");
    Serial.println(dta[BIT_LEN]);
    Serial.print("START_H: ");
    Serial.print(dta[BIT_START_H]);
    Serial.print("\tSTART_L: ");
    Serial.println(dta[BIT_START_L]);

    Serial.print("DATA_H: ");
    Serial.print(dta[BIT_DATA_H]);
    Serial.print("\tDATA_L: ");
    Serial.println(dta[BIT_DATA_L]);

    Serial.print("\r\nDATA_LEN = ");
    Serial.println(dta[BIT_DATA_LEN]);

    Serial.print("DATA: ");
    for (int i = 0; i < dta[BIT_DATA_LEN]; i++)
    { Serial.print("0x");
      Serial.print(dta[i + BIT_DATA], HEX);
      Serial.print("\t");
    }
    Serial.println();
    Serial.print("DATA: ");
    for (int i = 0; i < dta[BIT_DATA_LEN]; i++)
    {
      Serial.print(dta[i + BIT_DATA], DEC);
      Serial.print("\t");
    }
    Serial.println();
    Serial.println("+-----------------------------------------------------+\r\n\r\n");
  }
}
```

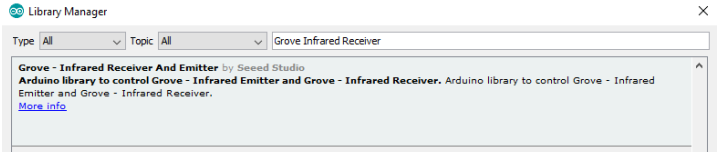
send §

```
#include <IRSendRev.h>

#define BIT_LEN         0
#define BIT_START_H     1
#define BIT_START_L     2
#define BIT_DATA_H      3
#define BIT_DATA_L      4
#define BIT_DATA_LEN    5
#define BIT_DATA        6

const int ir_freq = 38;                 // 38k

unsigned char dtaSend[20];

void dtaInit()
{
    dtaSend[BIT_LEN]        = 11;     // all data that needs to be sent
    dtaSend[BIT_START_H]    = 179;      // the logic high duration of "Start"
    dtaSend[BIT_START_L]    = 90;     // the logic low duration of "Start"
    dtaSend[BIT_DATA_H]     = 11;     // the logic "long" duration in the communication
    dtaSend[BIT_DATA_L]     = 33;     // the logic "short" duration in the communication

    dtaSend[BIT_DATA_LEN]   = 6;

    dtaSend[BIT_DATA+0]     = 128;      // data that will sent
    dtaSend[BIT_DATA+1]     = 127;
    dtaSend[BIT_DATA+2]     = 192;
    dtaSend[BIT_DATA+3]     = 63;
  dtaSend[BIT_DATA+4]     = 192;
    dtaSend[BIT_DATA+5]     = 63;
}

void setup()
{
    dtaInit();
}

void loop()
{
    IR.Send(dtaSend, 38);

    delay(2000);
}
```

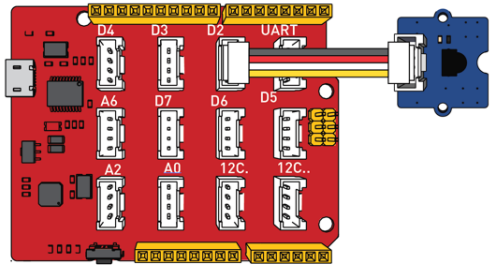
2. Then add the library of "**Grove – Infrared Emitter/Receiver**" by following the path below.

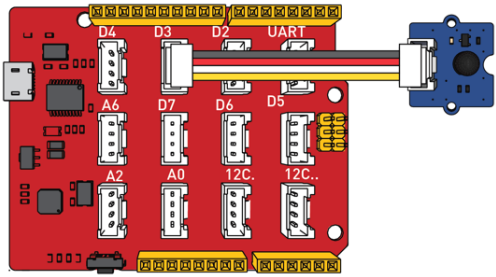


3. Type "**Grove Infrared Receiver**" in the text box and click install.



4. Connect the Infrared Emitter to **D3** header of the one Arduino or **D3** port of Seeeduino Lotus, and connect the Infrared Receiver to **D2** header of the other Arduino or **D2** port of Seeeduino Lotus.





5. Hit the **Upload** button

The code above will give a quick run through of these modules. The output results will be printed out on the serial monitor.

If you want to find this demo and more demos, browse to this location inside the Creators Kit sketchbook:

File > Sketchbook > Seeed_Creator_Kit > 34. Infrared Emitter&Recevier

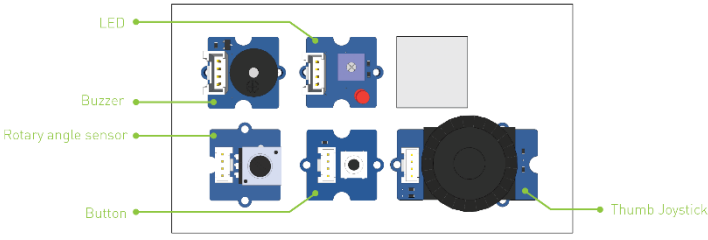
Tips

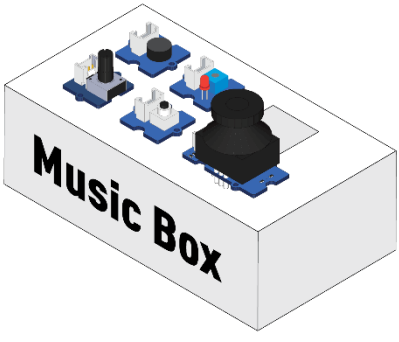
We can use the emitter not only to transmit data or commands, but also to emulate remotes to control your home appliance using an Arduino!

Demo Projects

This demo allows you to build your own music box, using just Grove modules and an Arduino compatible board! You can play musical notes through buzzer by rotating the rotary angle sensor and pressing the button! What's more exacting, it's that we have also packed five pre-defined melodies in this demo for you to explore. To trigger melodies, simply move joystick in to one of four directions or even hold down the joystick.

Top view





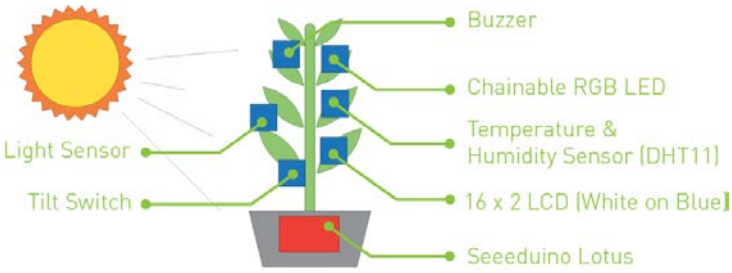
Materials List:

- Seeeduino Lotus v1.1
- Grove – Buzzer
- Grove – Button
- Grove – Rotary Angle Sensor
- Grove – Joystick
- Grove – LED
- Grove cables

Find the complete recipe here:

File > Sketchbook > Seeed_Creator_Kit >Smart_Music_Box

This is a Smart Garden using the modules from the Grove Beginner Kit to have a sensing and reminder system.



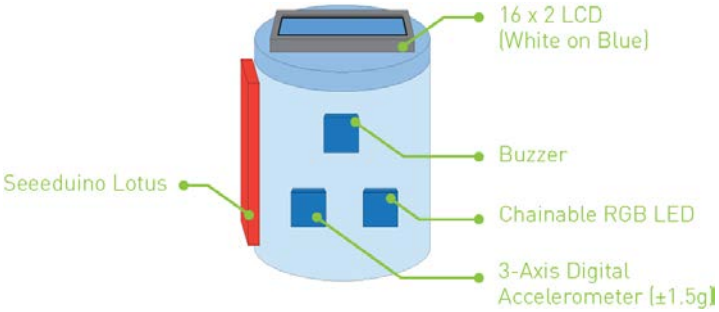
Materials List:

- Seeeduino Lotus v1.1
- Grove – Buzzer
- Grove – Chainable RGB LED V2.0 Grove – Light Sensor v1.2
- Grove – 16 x 2 LCD (White on Blue)
- Chainable RGB LED
- Grove – Temperature &Humidity Sensor (DHT11)
- Grove – Tilt Switch
- Grove cables

Find the complete recipe here:

File > Sketchbook > Seeed_Creator_Kit > Smart_Garden

This is a smart cup that reminds you to drink water at a certain period of time.



Materials List:

- Seeeduino Lotus v1.1
- Grove – Buzzer
- Grove – Chainable RGB LED V2.0
- Grove – 16 X 2 LCD (White on Blue)
- Grove – 3-Axis Digital Accelerometer (±1.5g)
- Grove cables

Find the complete recipe here:

File > Sketchbook > Seeed_Creator_Kit > Smart_Cup

Communication Methods

The Grove modules communicate via different protocols, and you can quickly figure out how to use them by familiarizing yourself with the communication methods of each module.

1. Digital Ports

There are six digital Grove ports. They are equivalent to digital pins 0 through 7 on the Seeeduino Lotus. Normally, they are used when reading a digital sensor that only outputs 0 or 1 or turning on or off an actuator. Some of these ports are multi-purpose and can function as PWM (pulse width modulation) outputs. They are port 3, port 5 and port 6. You will need these ports when driving a servo or fading an LED.

Digital ports are a must for serial communication too. There is one built-in hardwired serial port, which is UART, on port 1. This is the Seeeduino's default port for serial communication with the PC.

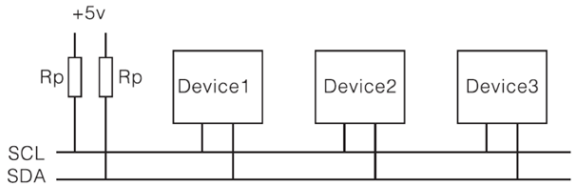
In cases where you need at least two serial devices or you need an available serial port for debugging purposes, other digital ports, software serial ports, can be used as well. We will encounter them a lot in our Grove system.

2. Analog Input Ports

There are three Grove ports for taking analog readings. Analog sensors can return readings ranging from 0 - 1024. Compared with digital sensors that only return 0 or 1, analog readings are more detailed and precise.

3. I2C Ports

There are two I2C Grove ports. I2C is a low-speed bus protocol that transfers data via two wires: SCL and SDA. SCL is the clock line that synchronizes data transfer over the I2C bus, and SDA is the data line. The following diagram illustrates the framework of an I2C bus.



You can connect up to 128 devices on the I2C bus; however, only one of them can work in master mode, while all of the others work in slave mode. For Grove, the master is the Arduino.

It generates the clock signals and sends commands to and/or receives data from all of the devices. In theory, each slave device has a unique hardware address and the master device can find slave devices via their addresses.

I2C ports are generally used when the amount of data is overwhelming for simple digital and analog ports. For example, when we want to obtain complex information such as angular acceleration, or read the current time from an RTC module, we should use the I2C ports.

4. UART Port

There is one UART port on this board. UART enables to have a serial communication with the connected devices. Therefore, you can send commands from a computer to the board and vice versa.

Resources

Arduino website: *www.arduino.cc*

Seeed bazaar: *www.seeedstudio.com*

Seeed wiki main page: *www.wiki.seeedstudio.com*

For technical inquiries: *techsupport@seeed.cc*

For more insights: *forum.seeedstudio.com*

Scan here to:

Stay tuned with our newsletter for all the latest product releases

www.seeedstudio.com
www.seeed.cc

@seeedstudio