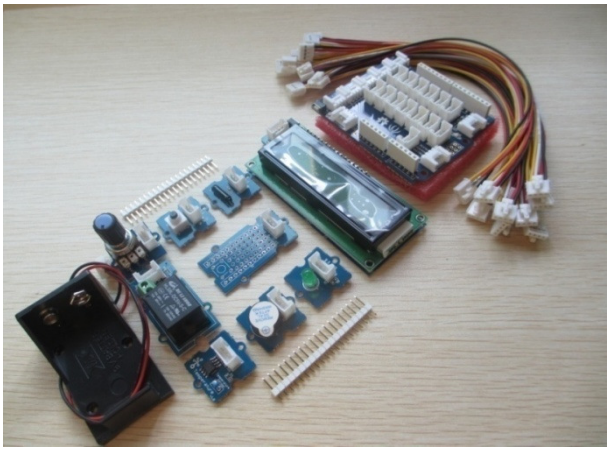
Arduino Starter Kit—Grove-Starter Kit

About Grove-Starter Kit

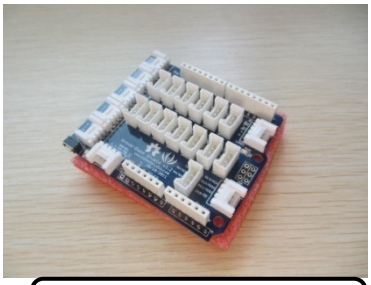
For someone first dabbling in the world of Arduino, the Grove-Starter Kit is an excellent choice in the journey of learning. This kit includes a variety of basic input and output modules and sensors. Using Grove as a unified interface to connect each module, this kit can help you create interesting circuits without soldering.



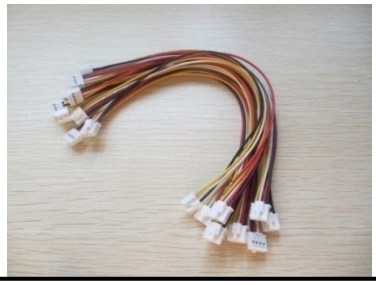
About this Tutorial

To jazz the tutorial up a bit and make something fun in the process, we created a cute farmhouse out of small wooden dowels, added some adorable clay creatures, and strung some EL wire for flare. Then we integrated the modules from the Grove Starter Kit to animate the environment and illustrate their practical applications.

Packing List for the Grove-Starter Kit

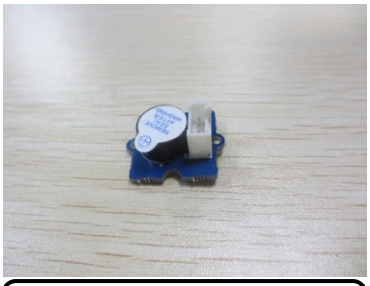


1 Grove-Base Shield



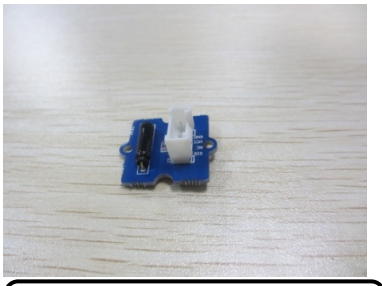
Connectors - 10 Grove Cables

Functional modules:

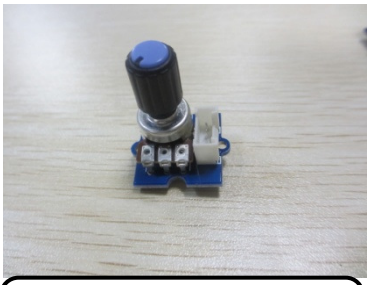


1 Grove-Buzzer

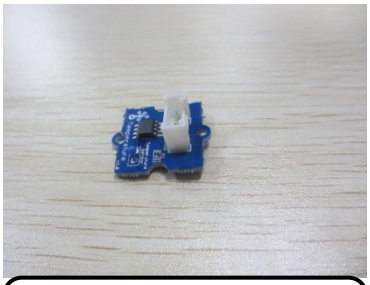
1Grove-LED



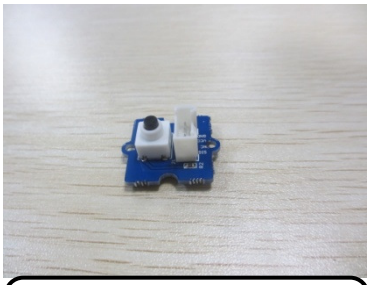
1 Grove-Tilt Switch



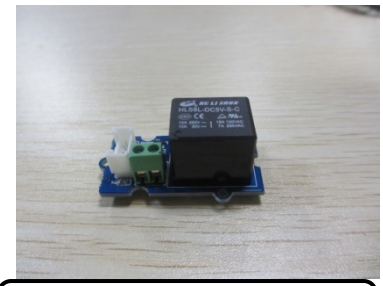
1 Grove-Rotary Angle Sensor



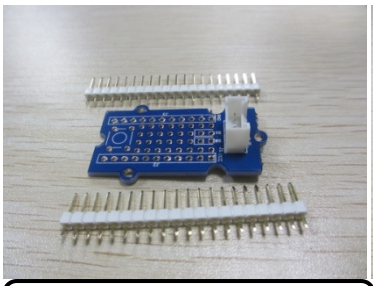
1 Grove-Temperature Sensor



1 Grove-Button



1 Grove-Smart Relay



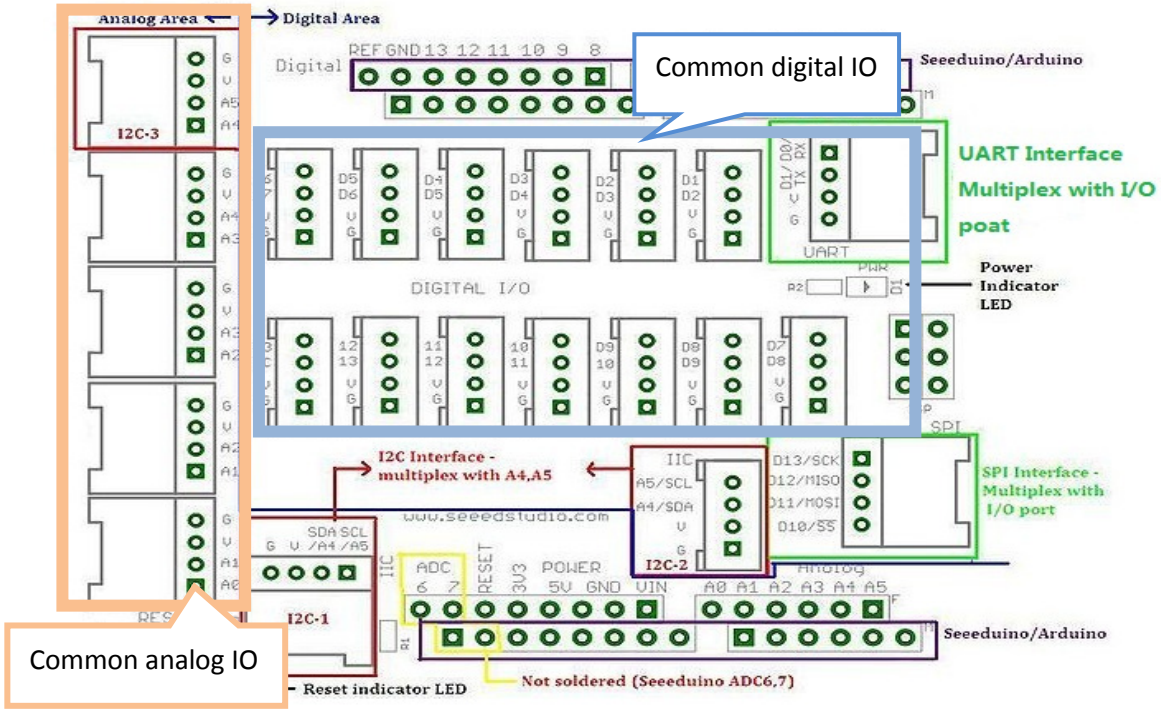
1 Grove-Protoshield

1 Grove-Serial LCD

Preparation

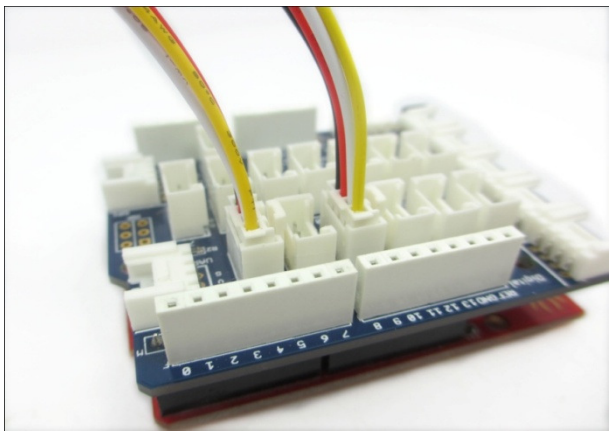
- **Base Shield Introduction**

The purpose of the Base Shield is to allow easy connection of any microprocessor input and output pins to Grove units. Each socket is clearly labeled with its matching I/O pin and contains 5 V, GND, and two I/O pin connections. For a more detailed examination of the Base Shield please refer to the diagram below.



Hardware Installation

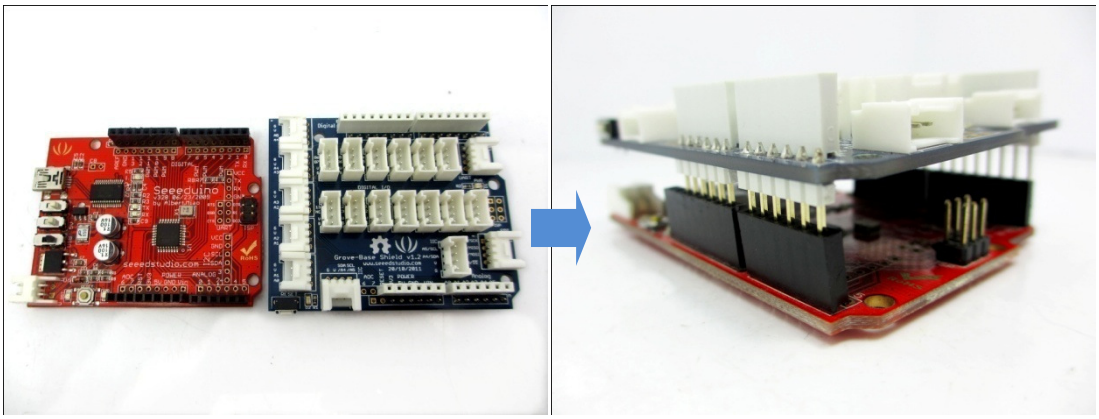
When using the digital I/O, note the staggered alignment of the pins – that is, one socket handles D1 and D2, the next D2 and D3, and so on. If you are going to use a Grove input module and a Grove output module which have two signal pins, like the LED module and the button module, separate your Grove cables so that a socket is between them as shown below.



Grove connectors for two-signal Grove modules cannot sit side-by-side on the Base Shield, because one pin, such as D2, will be utilized by both modules simultaneously. On the other hand, if you have two Grove units that use only one digital pin each, such as the tilt switch or the buzzer, they can sit alongside each

other on the Base Shield as they only use one of the digital lines in the connecting cable and therefore will not interfere with each other. These rules apply to the Analog I/O sockets, as well. Make sure you know the layout of each socket before you hook up your connectors.

- **Integratation with Arduino/Seeeduino**



Take out your Arduino or Seeeduino, and insert the pins of the Base Shield into the corresponding Arduino/Seeeduino ones, as depicted above.

➢ Arduino Analog & Digital Pins

Analog Pins:

Arduino reads analog signals through analog pins, and outputs analog signals through pulse-width modulation (PWM). Arduino has 6 analog input ports (A0-A5) and 6 analog output ports (D3 D5 D6 D9 D10 D11). These pins can be used to supply variable output voltages. Pins (A0-A5) can also be used as digital input and output pins, but by default, they are analog input pins.

Digital Pins:

Arduino reads digital signals from digital pins. These pins can be used as both input and output pins. Arduino has 14 digital I/O pins (D0-D13), of which D13 is usually used as an output pin because it is connected to an LED on PCB.

NB: All signals read from the pins are voltages instead of current values. Because of the high internal resistance in the pins, only a little current passes to the ground.

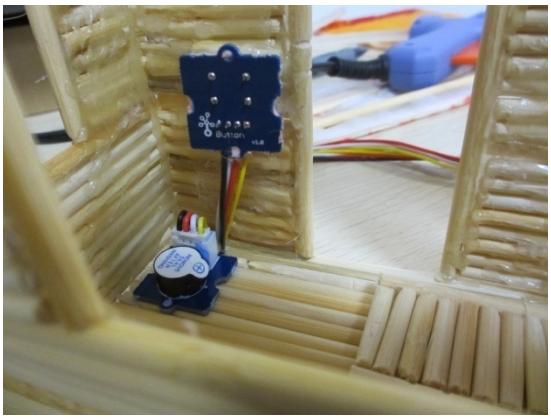
IDE Installation

We won't give an introduction of this part here. For more detailed information, please refer to the website, http://arduino.cc/en/Guide/HomePage, and then install step-by-step according to your operating system.

Welcome to the Grove-Starter Kit Farmhouse Project

1. Doorbell — Button + Buzzer

Goal: When the doorbell/button is pushed, the buzzer will buzz and announce the visitor.




➢ Module Introduction

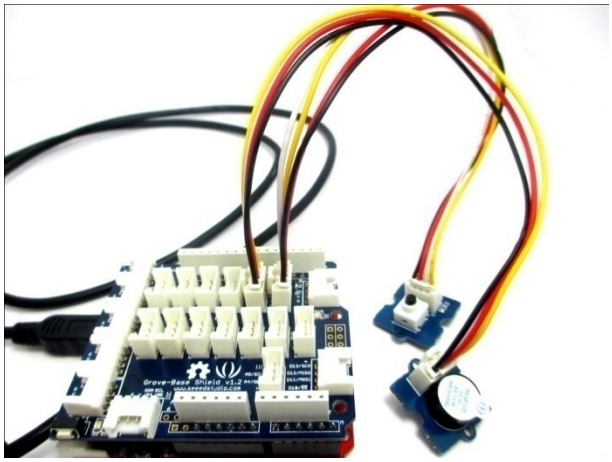
Button: Simple momentary button for input.

Buzzer: It can be connected to digital outputs, and will emit a tone when the output is high. Alternatively it can be connected to an analog pulse-width modulation output to generate various tones and effects.

➢ Hardware Setup

Materials Needed: 1 Button + 1 Buzzer + 2 Grove Cables

Hardware Connection Method: Plug the button module into the D1 connector on the Base Shield. The button is now connected to digital pin 1 on the Arduino. Plug the buzzer into the D2 connector on the Base Shield. Now the buzzer is connected to digital pin 2 on the Arduino.



➢ Software Design

Code 1: Simple Doorbell

```
int buttonPin = 1;
```

```
int buzzerPin = 2;

void setup()
{
   pinMode(buttonPin,INPUT);//set button as digital input
   pinMode(buzzerPin,OUTPUT);//as buzzer as digital output
}

void loop()
{
   if(digitalRead(buttonPin))//check button is pressed or not
   {
         digitalWrite(buzzerPin,HIGH);//pressed，then buzzer buzzes
   }
   else
   {
         digitalWrite(buzzerPin, LOW);//not pressed，then buzzer remains silent
   }
}
```

Ultimate Result：Press the doorbell, and the buzzer will Buzz “B”

Code 2: Musical Doorbell

```
int buttonPin = 1;
int buzzerPin = 2;

int length = 40; // the number of notes
char notes[] = "ccggaagffeeddc "; // a space represents a rest
int beats[] = { 1,1,1,1,1,1,2,1,1,1,1,1,1,2,4 };
int tempo = 300;

void playTone(int tone, int duration) {
   for (long i = 0; i < duration * 1000L; i += tone * 2) {
      digitalWrite(buzzerPin, HIGH);
      delayMicroseconds(tone);
      digitalWrite(buzzerPin, LOW);
      delayMicroseconds(tone);
   }
}

void playNote(char note, int duration) {
   char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' };
   int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014, 956 };

   // play the tone corresponding to the note name
```

```
    for (int i = 0; i < 8; i++) {
      if (names[i] == note) {
        playTone(tones[i], duration);
      }
    }
}

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

void loop() {
    if(digitalRead(buttonPin))
    {
    for (int i = 0; i < length; i++) {
      if (notes[i] == ' ') {
        delay(beats[i] * tempo); // rest
      } else {
        playNote(notes[i], beats[i] * tempo);
      }

      // pause between notes
      delay(tempo / 20);
    }
    }
}
```

End Result：Press the doorbell, and the buzzer will play beautiful music.

2. Little Chandelier — Rotary Angle Sensor

Goal: Rotate the Rotary Angle Sensor and the luminance of the chandelier inside the house will increase or decrease depending on the rotary angles.

➢ Module Introduction

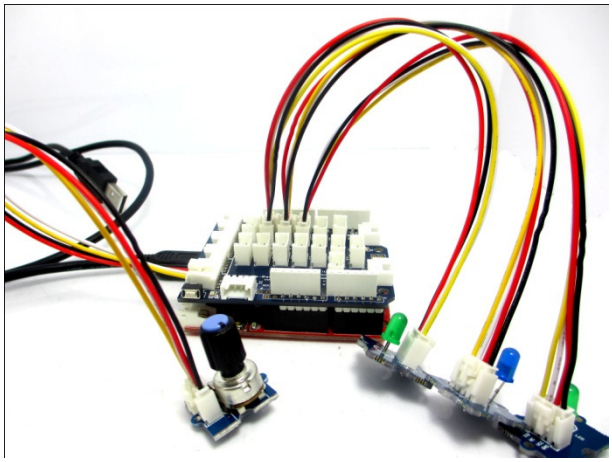
Rotary Angle Sensor: It can be used as an Arduino/Seeeduino analog input. Its output voltage changes according to the rotated angles.

LED: It can be used as an Arduino/Seeeduino digital output, or it can also be controlled using PWM.

➢ Hardware Setup

Materials Needed: 1 Rotary Angle Sensor + 3 LEDs + 4 Grove Cables

Hardware Connection Method: Plug the rotary angle sensor into the A0 connector on the Base Shield. The rotary angle sensor is now connected to analog pin 0 on the Arduino. Plug 3 LEDs into theD3, D4, D5 connectors on the Base Shield. The 3 LEDs are now connected to digital pins 3, 4, and 5 on the Arduino.



➢ Software Design

Code: Little Chandelier

```
int sensorPin = 0;
int ledPin1 = 3;
int ledPin2 = 4;
int ledPin3 = 5;
int sensorValue = 0;
void setup()
{
   pinMode(ledPin1,OUTPUT);//set 3 LEDs as digital output
   pinMode(ledPin2,OUTPUT);
   pinMode(ledPin3,OUTPUT);
}
void loop(){
   sensorValue = analogRead(sensorPin);//read the value from the sensor
   //adjust the value 0 to 1024 to 0 to 255
   sensorValue = map(sensorValue,0,1024,0,255);
   analogWrite(ledPin1,sensorValue);//write the value
```

```
    analogWrite(ledPin2,sensorValue);
    analogWrite(ledPin3,sensorValue);
}
```

3. Swing

Goal: When the bear swings, the green EL wire will light up.


➢ Module Introduction：

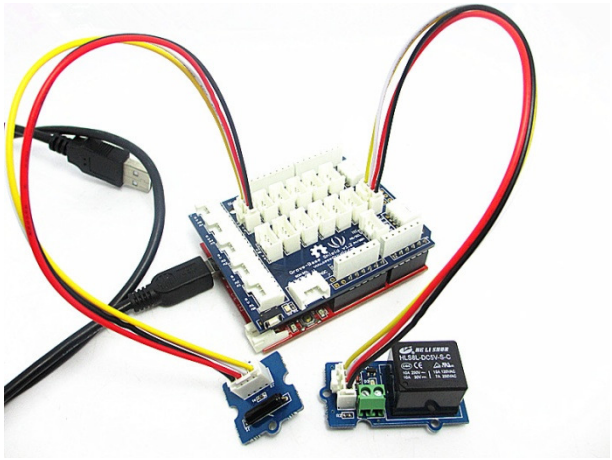
Tilt Switch: This is used as an ordinary switch. When it tilts, the switch is off; when it's balanced, the switch is on.

Smart Relay: Control high voltage circuit by 5 V logic digital output. Relay is equal to an electronic switch. When the signal from the Arduino/Seeeduino is at a high level, the switch is off and vice versa. This switch can also be used in other circuits. In this case, relay is a switch in the luminescent tube circuit.

➢ Hardware Setup

Materials Needed: 1 Tilt Switch + 1 Smart Relay + 2 Grove Cables + 1 EL wire + 1 EL wire 2xAA pocket inverter

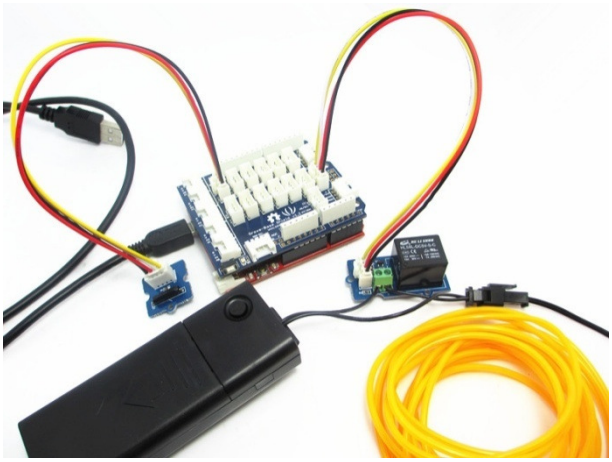
Hardware Connection Method: Plug the Grove button module into the D6 connector on the Base Shield. The button is now connected to digital pin 6 on the Arduino. Then plug the buzzer into the D7 connector on the Base Shield. Now the LED is connected to digital pin 7 on the Arduino.

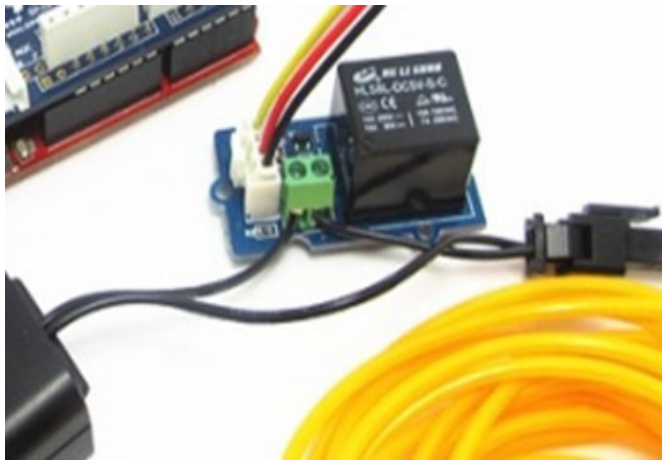


Hardware Connection Method of Smart Relay's output circuit:

1 Cut either one wire of EL wire Pocket Converter and plug each end into the terminals of the Relay

2 Tighten the screws



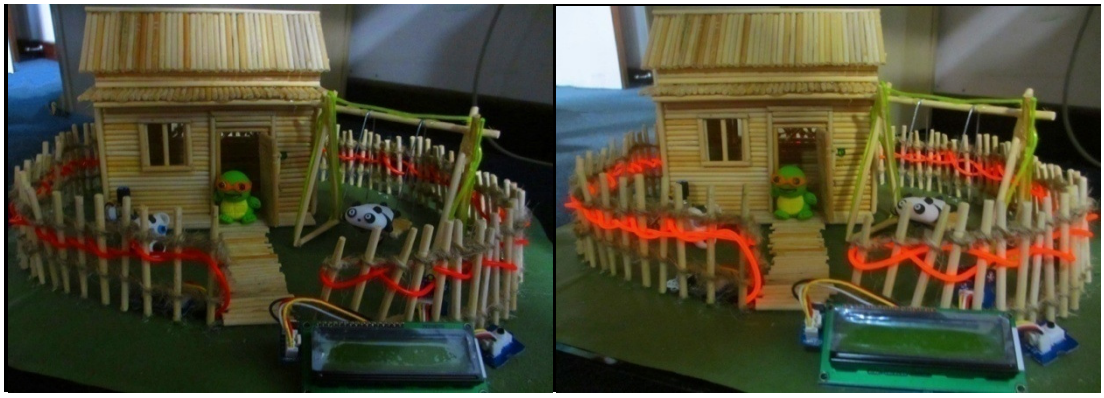


➢ Software Design

```
int tiltPin = 6;
int relayPin = 7;
void setup()
{
   pinMode(tiltPin,INPUT);//set tilt as digital input
   pinMode(relayPin,OUTPUT);//set relay as digital output
}
void loop()
{
   if(digitalRead(tiltPin)) //check whether tilt is balanced or not
   {
         digitalWrite(relayPin,HIGH);// tilt is inbalanced，then relay is off
   }
   else
   {
         digitalWrite(relayPin, LOW);//tilt is balanced，then relay is on
   }
}
```

4. Fence

Goal: Switch on, and the EL wire woven through the fence will light up.



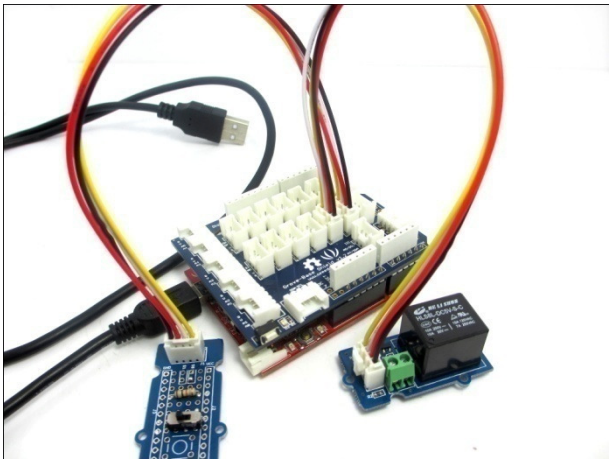
➢ Module Introduction

Protoshield: This allows you to add your own circuitry or components to your Grove system prototypes.

➢ Hardware Set-up

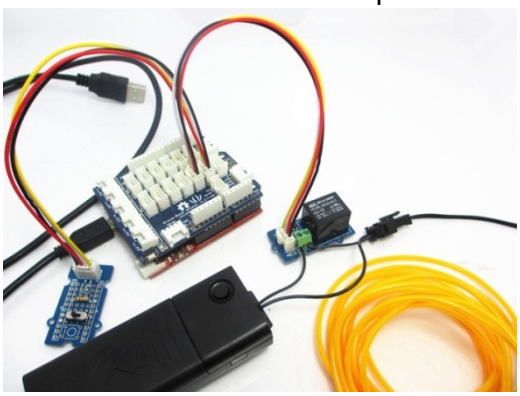
Materials Needed: 1 Protoshield+ 1 Smart Relay + 2 Grove Cables + 1 EL wire + 1 EL wire 2xAA pocket inverter

Hardware Connection Method: Plug the button into the D8 connector on the Base Shield. The button is now connected to digital pin 8 on the Arduino. Then plug the buzzer into the D9 connector on the Base Shield . The LED is now connected to digital pin 9 on the Arduino.

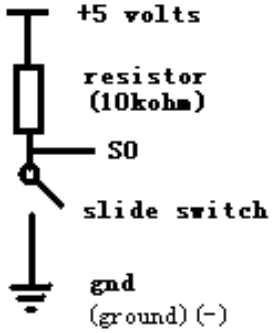


Hardware Connection Method of Smart Relay's output circuit：

The same as the method depicted above in the "Swing" part.



The Protoshield Circuit Schematic（switch + resistance）:



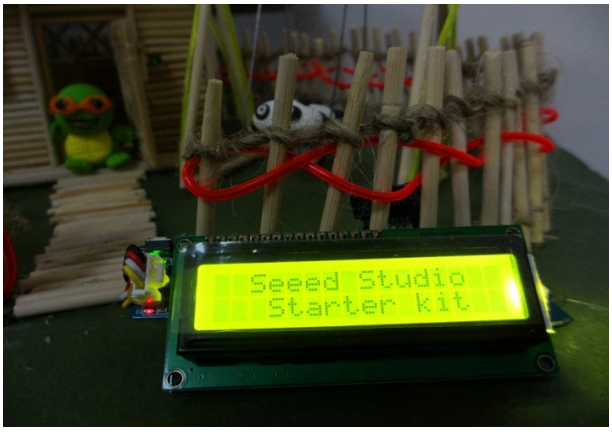
- **Software Design**

```
int protoshieldPin = 8;
int relayPin = 9;
void setup()
{
   pinMode(protoshieldPin,INPUT);
   pinMode(relayPin,OUTPUT);
}
void loop()
{
   if(digitalRead(protoshieldPin))
   {
        digitalWrite(relayPin,HIGH);
   }
   else
   {
        digitalWrite(relayPin, LOW);
   }
}
```

5. Screen

5.1 Alpha-Numeric Character Display

Goal： The static display on the screen is “Seeed Studio” and “Starter Kit”.

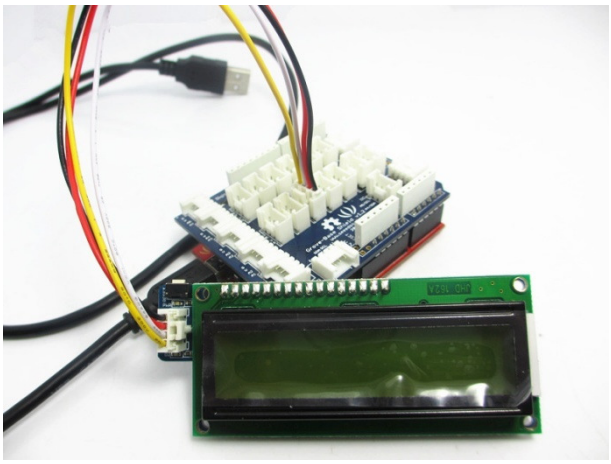


➢ Module Introduction

Serial LCD: Display alpha-numeric characters max 32 in a display.

➢ Hardware Set-up

Materials Needed: 1 Serial LCD + 1 Grove Cable
Hardware-connection Method: Plug the serial LCD into the D11 and D12 connectors on the Base Shield. The Serial LCD is now connected to digital pins 11 and 12 on the Arduino.



➢ Software Design

Attention: We provide you with the Serial LCD library. Before programming, please download the dedicated library and decompress it to the Arduino library.

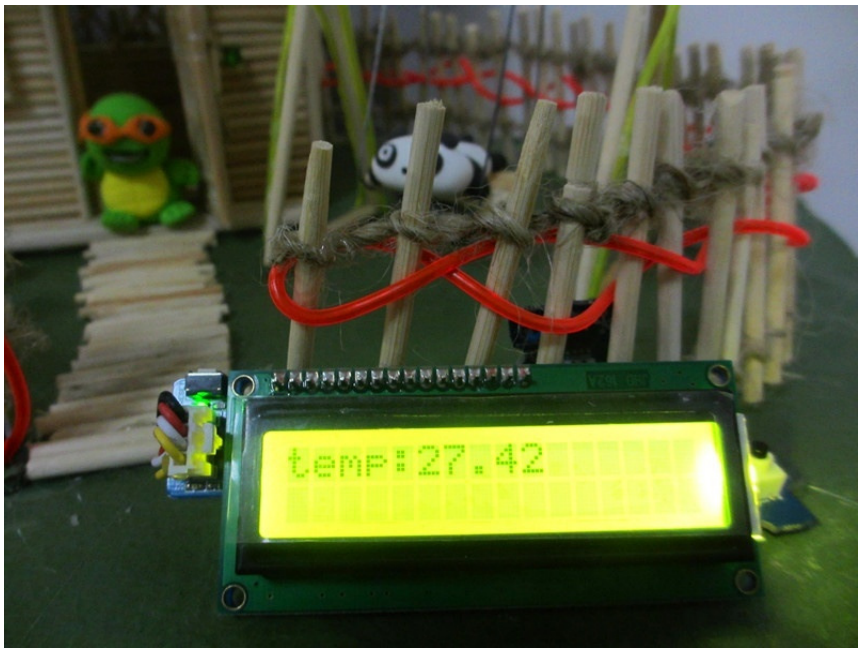
1. Click http://www.seeedstudio.com/wiki/images/a/ac/SerialLCD.zip and download the library.
2. Decompress it to C:\Program Files\arduino-1.0\libraries.

```
#include <SerialLCD.h>
#include <SoftwareSerial.h> //this is a must
SerialLCD slcd(11,12); //this is a must, assign soft serial pins
void setup()
{
```

```
    slcd.begin();// set up :
}
void loop()
{
    slcd.backlight();// Turn on the backlight:
    slcd.setCursor(0,0); // set the cursor to (0,0):
    slcd.print("    Seeed Studio"); // Print a message to the LCD.
    slcd.setCursor(0,1);
    slcd.print("      Starter kit      ");
}
```

5.2 Temperature Display

Goal： Display present temperature.



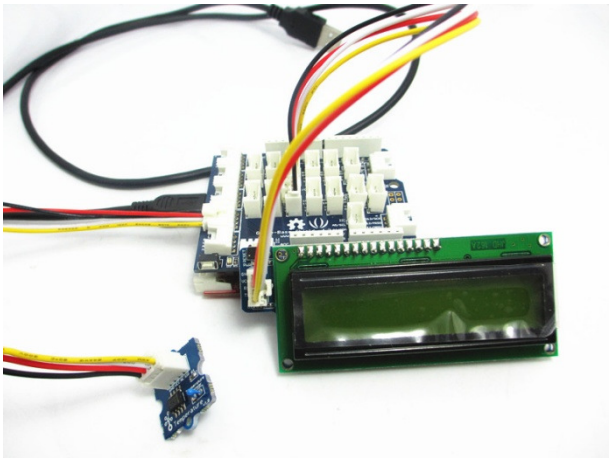
➢ Module Introduction

Temperature Sensor: Perfect for use with Arduino/Seeeduino analog inputs. Its output voltage changes according to temperature.

➢ Hardware Set-up

Materials Needed: 1 Serial LCD + 1 Temperature Sensor + 2 Grove Cables

Hardware-connection Method: Plug the temperature sensor into the A1 connector on the Base Shield. The temperature sensor is now connected to analog pin 1 on the Arduino. Plug the serial LCD into the D11 and D12 connectors on the Base Shield. The Serial LCD is now connected to analog pins 11 and 12 on the Arduino.

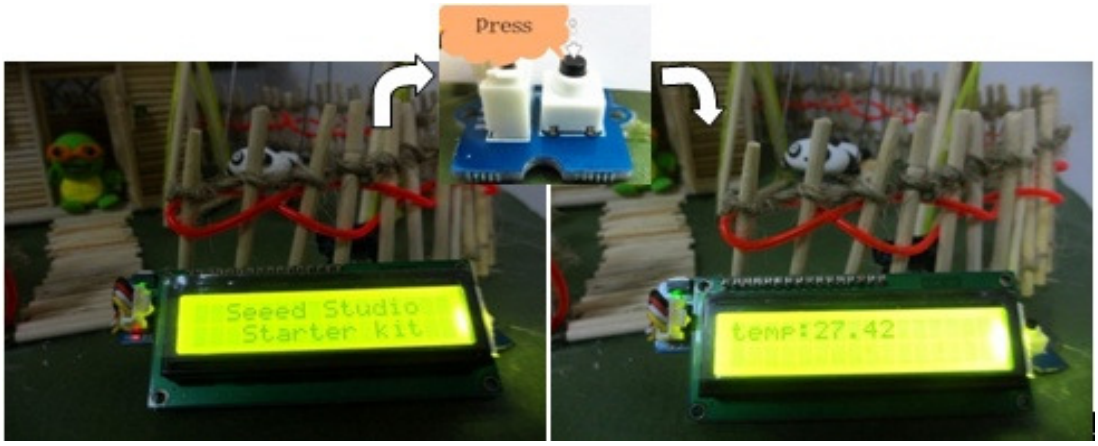


- **Software Design**

```
#include <SerialLCD.h>
#include <SoftwareSerial.h>
int tempPin = 1;
SerialLCD slcd(11,12);

void setup()
{
   slcd.begin();
}
void loop()
{
   slcd.backlight();
   float temp = analogRead(tempPin);//getting the voltage reading from the
                                    //temperature sensor
   temp = (float)(1023 - temp)*10000/temp;
   temp = 1/(log(temp/10000)/3975+1/298.15)-273.15;
   slcd.setCursor(0,0);
   slcd.print("temp:");
   slcd.print(temp,2);

   delay(1000);
   slcd.clear();
}
```

5.3 Switching Between Alpha-numeric Characters and Temperature

Goal： Turn on the power, and the screen will display “Seeed Studio” and “Starter Kit”. Press the button, and the screen will display present temperature. Release the button, the screen will display the previous verbage.



➢ Hardware Set-up

Materials Needed: 1 Serial LCD + 1 Temperature Sensor + 1 button + 3 Grove Cables

Hardware-connection Method: Plug the temperature sensor into the A1 connector on the Base Shield. The temperature sensor is now connected to analog pin 1 on the Arduino. Plug the serial LCD into the D11 and D12 connectors on the Base Shield. The Serial LCD is now connected to digital pins 11 and 12 on the Arduino. Plug the button into the D10 connector on the Base Shield. The button is now connected to digital pin 10 on the Arduino .

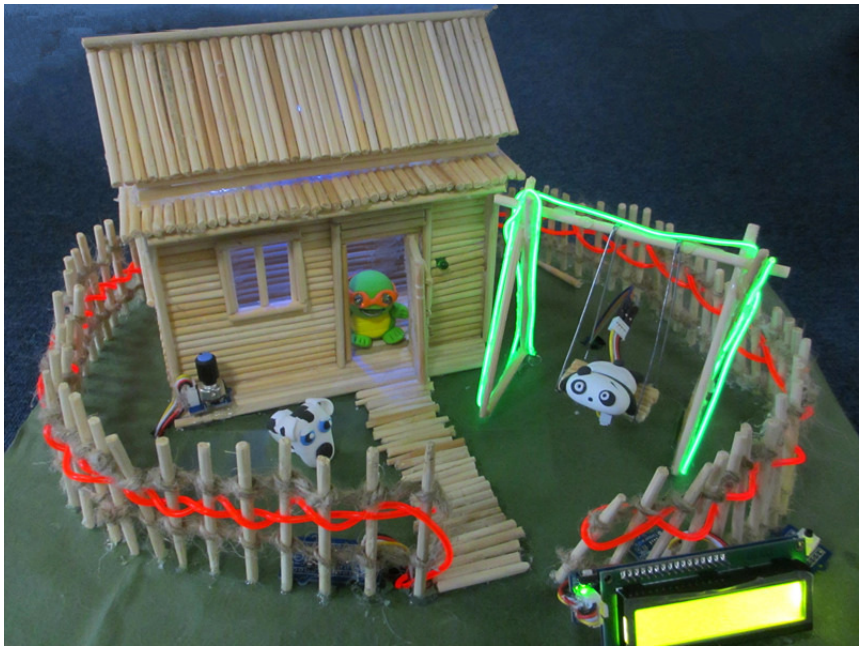
➢ Software Design

```
#include <SerialLCD.h>
#include <SoftwareSerial.h>

int tempPin = 0;
int buttonPin = 10;
SerialLCD slcd(11,12);
void setup()
{
   slcd.begin();
   pinMode(buttonPin,INPUT);
}

void loop()
{
   slcd.backlight();
   if(digitalRead(buttonPin))
   {
      delay(10);
      slcd.clear();
   }
   if(digitalRead(buttonPin))//check whether the button is pressed or not
```

```
    {
        slcd.clear();
            float temp = analogRead(tempPin);// is pressed, temperature is displayed
            temp = (float)(1023 - temp)*10000/temp;
            temp = 1/(log(temp/10000)/3975+1/298.15)-273.15;
            slcd.setCursor(0,0);
            slcd.print("temp:");
            slcd.print(temp,2);
            delay(500);
    }else//Or else, numeric and characters are displayed
    {
       slcd.clear();
            slcd.setCursor(0,0);
            slcd.print("    Seeed Studio");
            slcd.setCursor(0,1);
            slcd.print("    Starter kit      ");
            delay(500);
    }
}
```

Module Assembly

The hardware for the Grove-Starter Kit Farmhouse project is ready. Let's take a look at the final product.



The overall software designs are as follows:

// include the library code:

```
#include <SerialLCD.h>
#include <SoftwareSerial.h> //this is a must

int buttonPin = 1;
int buzzerPin = 2;
int slidePin = A0;
int ledPin1 = 3;
int ledPin2 = 4;
int ledPin3 = 5;
int tiltPin = 6;
int relayPin = 7;
int protoshieldPin = 8;
int relayPin2 = 9;
int lcd_buttonPin = 10;
int lcdPin1 = 11;
int lcdPin2 = 12;
int tempPin = A1;

SerialLCD slcd(11,12);

void playTone(int tone, int duration) {
   for (long i = 0; i < duration * 1000L; i += tone * 2) {
      digitalWrite(buzzerPin, HIGH);
      delayMicroseconds(tone);
      digitalWrite(buzzerPin, LOW);
      delayMicroseconds(tone);
   }
}
void playNote(char note, int duration) {
   char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' };
   int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014, 956 };

   // play the tone corresponding to the note name
   for (int i = 0; i < 8; i++) {
      if (names[i] == note) {
         playTone(tones[i], duration);
      }
   }
}
void playMusic()
{
   int length = 40; // the number of notes
   char notes[] = "ccggaagffeeddc "; // a space represents a rest
   int beats[] = { 1,1,1,1,1,1,2,1,1,1,1,1,1,2,4 };
```

```
    int tempo = 300;
    for (int i = 0; i < length; i++) {
        if (notes[i] == ' ') {
            delay(beats[i] * tempo); // rest
        } else {
            playNote(notes[i], beats[i] * tempo);
        }
        // pause between notes
        delay(tempo / 20);
    }
}
void lightAdjust(int value)
{
    value = map(value,0,1023,0,255);
    analogWrite(ledPin1,value);
    analogWrite(ledPin2,value);
    analogWrite(ledPin3,value);
}

void setup() {
    pinMode(buzzerPin, OUTPUT);
    pinMode(buttonPin,INPUT);
    pinMode(ledPin1,OUTPUT);
    pinMode(ledPin2,OUTPUT);
    pinMode(ledPin3,OUTPUT);
    pinMode(tiltPin,INPUT);
    pinMode(relayPin,OUTPUT);
    pinMode(protoshieldPin,INPUT);
    pinMode(relayPin,OUTPUT);
    pinMode(lcd_buttonPin,INPUT);
    slcd.begin();
    slcd.backlight();
}

void loop() {
    int slideValue = 0;
    int saveValue = 0;
    int lcd_buttonFlag = 1;
    while(1)
    {
    //1.doorbell
    if(digitalRead(buttonPin) == true)
    {
    playMusic();
```

```
}
//2.room light
slideValue = analogRead(slidePin);
if(slideValue != saveValue)
{
   lightAdjust(slideValue);
   saveValue = slideValue;
}
//3.swing
if(digitalRead(tiltPin))
{
      digitalWrite(relayPin,HIGH);
      delay(200);
}
else
{
      digitalWrite(relayPin, LOW);
      delay(200);
}
//4.fence
   if(digitalRead(protoshieldPin))
{
      digitalWrite(relayPin2,HIGH);
}
else
{
      digitalWrite(relayPin2, LOW);
}
//5.lcd
  if(!digitalRead(lcd_buttonPin))
{
   delay(10);
   slcd.clear();
}
if(digitalRead(lcd_buttonPin)&&lcd_buttonFlag == 1)
{
   slcd.clear();
   slcd.setCursor(0,0);
   slcd.print("Welcome to Seeed Grove");
   slcd.setCursor(0,1);
   slcd.print("         Starter kit      ");
   delay(500);
   lcd_buttonFlag = 0;
}
```

```
    if(!digitalRead(lcd_buttonPin))
    {
        slcd.clear();
        float temp = analogRead(tempPin);
        temp = (float)(1023 - temp)*10000/temp;
        temp = 1/(log(temp/10000)/3975+1/298.15)-273.15;
        slcd.setCursor(0,0);
        slcd.print("temp:");
        slcd.print(temp,2);
        delay(500);
        lcd_buttonFlag = 1;
    }
    }//end while(1)

}
```