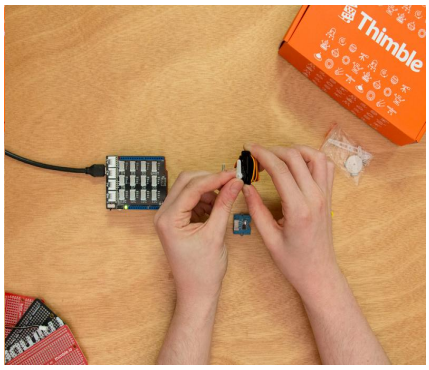
Now we can get started

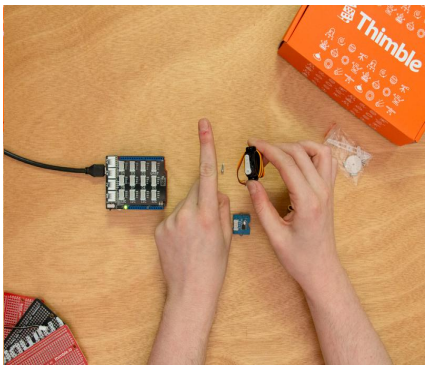
Line up the circles

Take the servo...

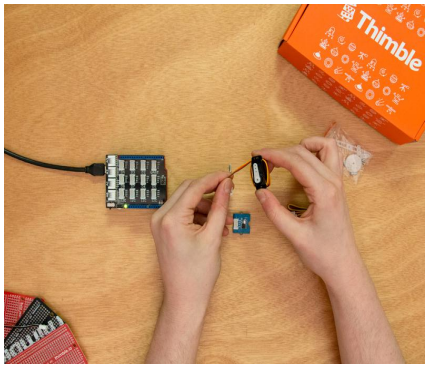


Place them together with the 'Servo Arm' pointing up on the

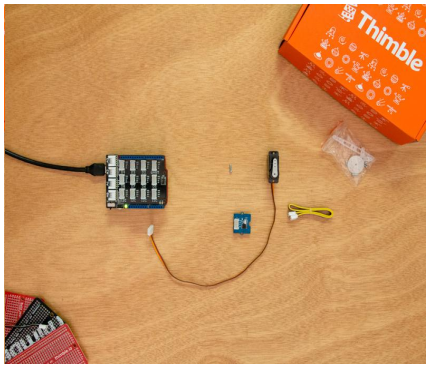
... and the 'Servo Arm'



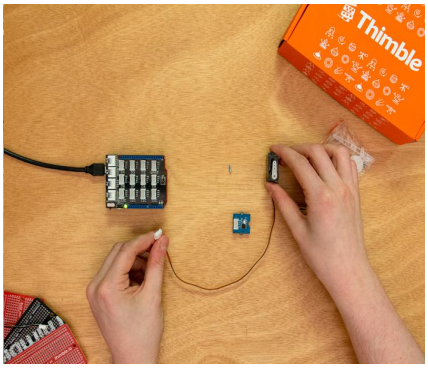
The 'Servo Arm' shouldn't be hanging off the servo body



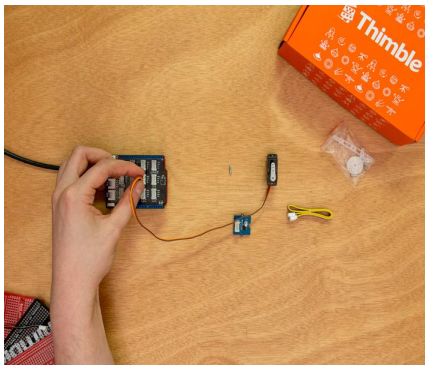
Take the wire around the servo...



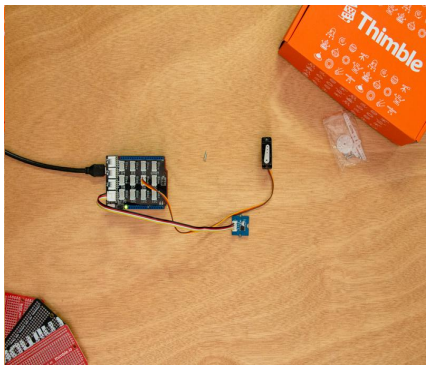
... and unwrap it



Take the cable ...



... plug into Digital socket D6



Plug in the light sensor to Analog socket A0

Upload

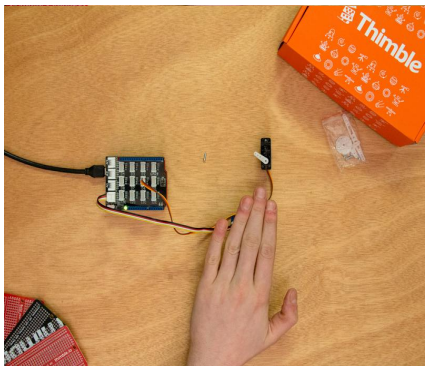
Upload the code shown below. This tutorial uses **Analog** socket A0. If you are using a different socket update the code after copying it.

Copy to clipboard

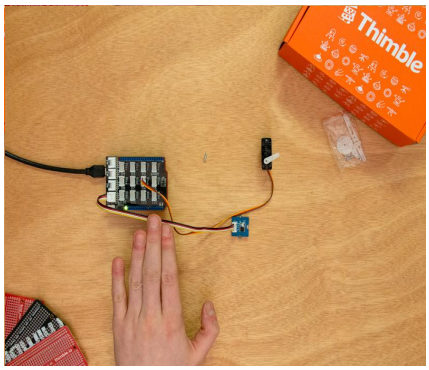
```
//Change here if you're using a different socket
#define sensorSocket A0

int val;

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

void loop() {
  val = analogRead(sensorSocket);
  Serial.println(val);
}
```

Observe

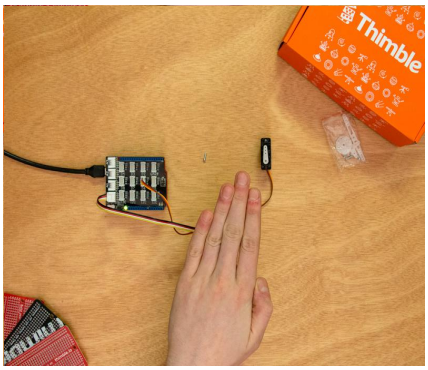
Open the Serial Plotter and take note of what is the highest and lowest values. One will be when light is hitting it and the other is when the sensor is covered.



Minimum



Middle



Maximum

Modify

Upload the code below. Change the trigger variables in `lowTrigger` and `highTrigger`. for the values you recorded.

Copy to clipboard

```
#include <Servo.h>

//Change here if you're using a different socket
#define sensorSocket A0
#define servoSocket 6

Servo indicatorServo;

int val;
int lowTrigger = 16;   //<- Change to YOUR measured value
int highTrigger = 560; //<- Change to YOUR measured value

void setup() {
  indicatorServo.attach(servoSocket);
}

void loop() {
  val = analogRead(sensorSocket);
  val = map(val,lowTrigger,highTrigger,179,1);
  indicatorServo.write(val);
  delay(1000);
}
```

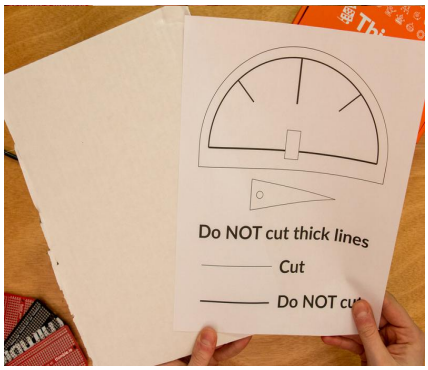
Construction

Recommended - PDF Template (https://d2j2m4p6r3pg95.cloudfront.net/module_files/creator-set/assets/templates/IndicatorTemplate.pdf)

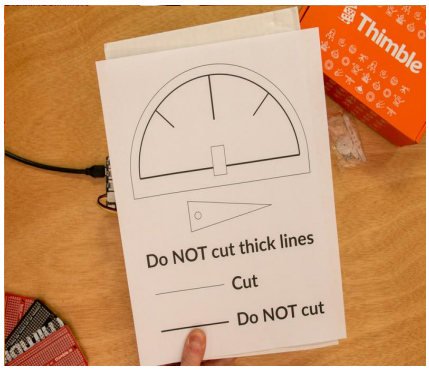
SVG Template (https://d2j2m4p6r3pg95.cloudfront.net/module_files/creator-set/assets/templates/IndicatorTemplate.svg)

AI Template (https://d2j2m4p6r3pg95.cloudfront.net/module_files/creator-set/assets/templates/IndicatorTemplate.ai)

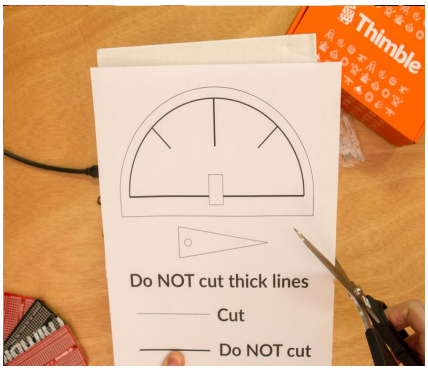
Print out the template and assemble your old fashioned indicator. Find some cardboard and cut out the template. Disconnect the servo from its socket. Place it through the rectangle hole in the middle. Seat the servo and use some tape to keep it in place. Place another piece of tape behind the servo arm then attach the pointer. Use the screw from the beginning and screw it into the hole in the pointer and into the servo. With the construction complete, plug the servo back into the same socket. Check out your data streamed in real time.



Printed out template and some

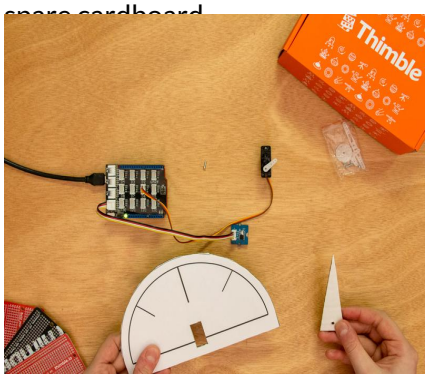


Line it up

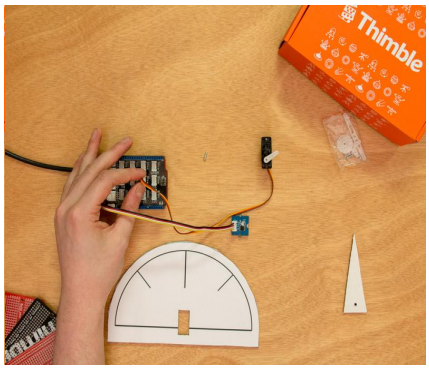


Cut it out

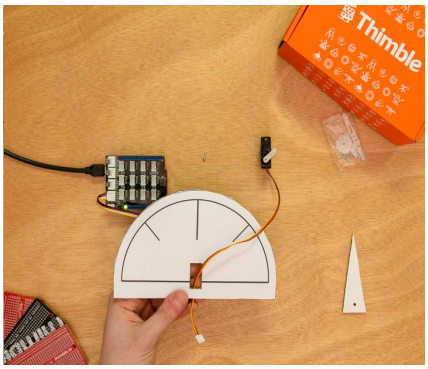
spare cardboard



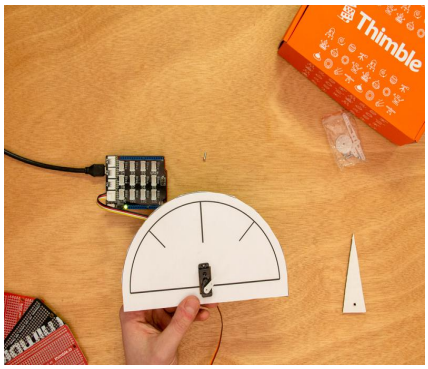
All cut out



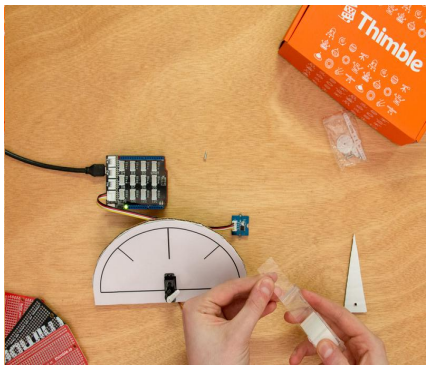
Disconnect the servo from its socket



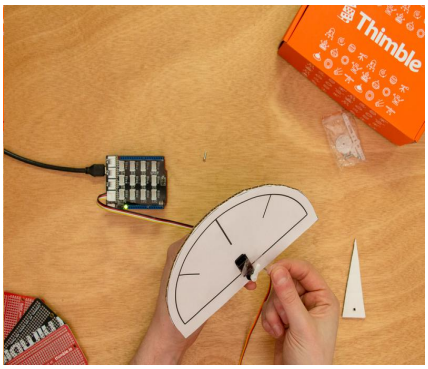
Place it through the rectangle hole in the middle



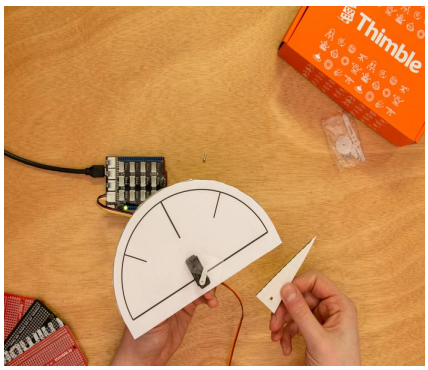
Seat the servo



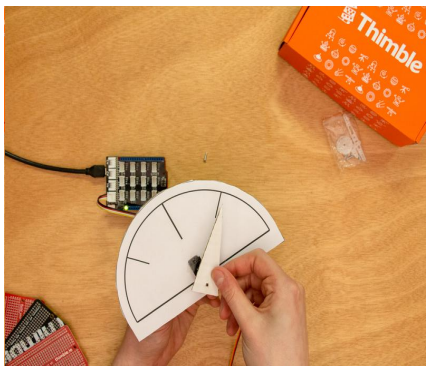
Place a piece of tape...



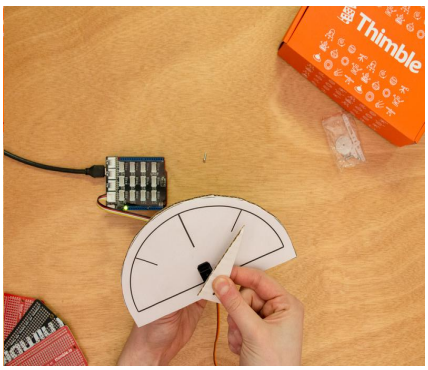
... behind the servo arm



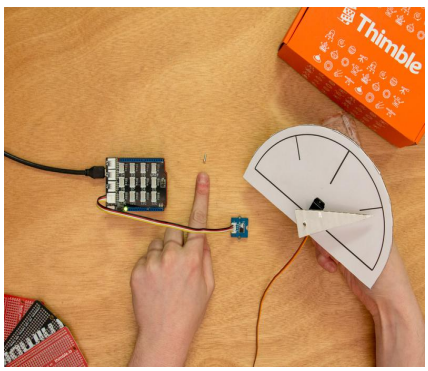
Take the pointer



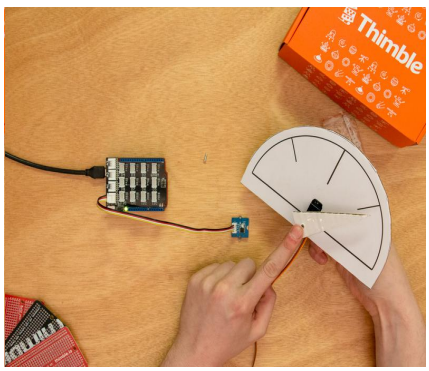
Place it on top of the servo arm



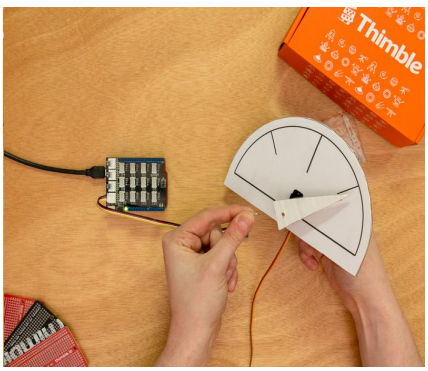
Tape it down



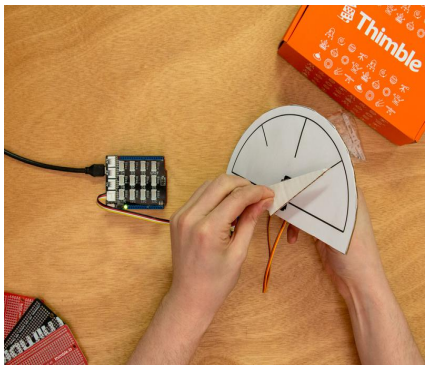
Use the screw from the beginning



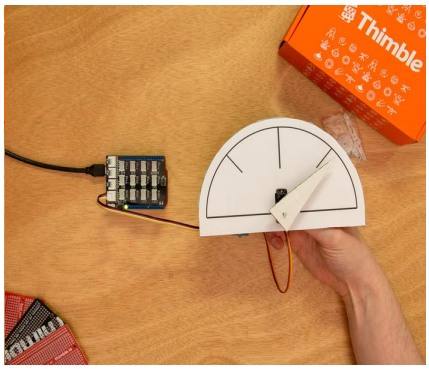
Screw it into the hole in the pointer



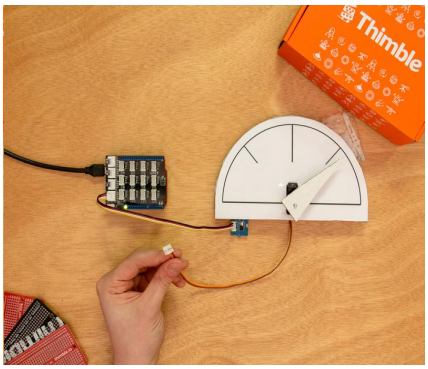
Into the servo



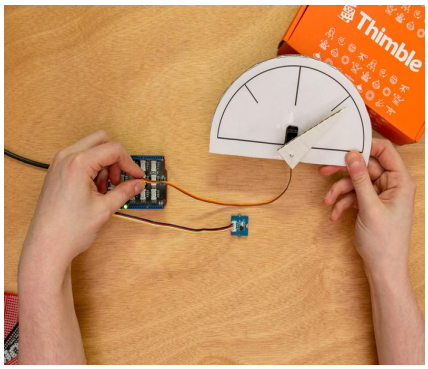
Hand tighten is fine. A screwdriver is easier



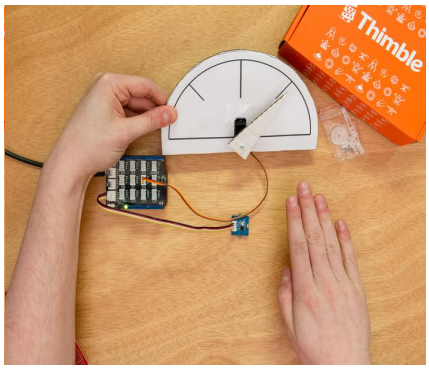
Almost done



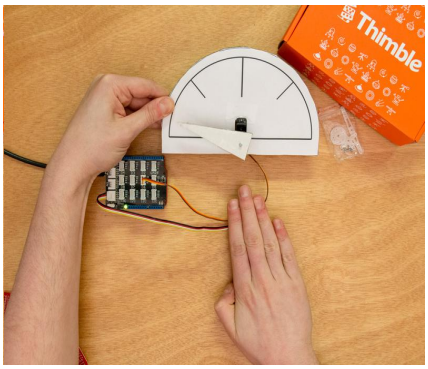
Take the servo cable



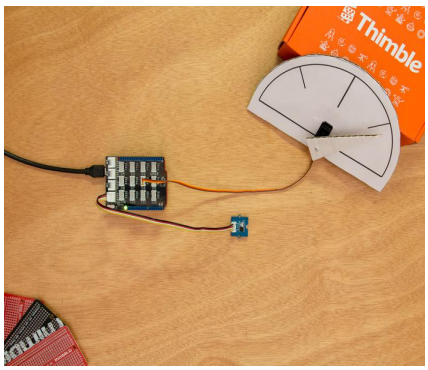
Plug the servo back into the same socket



A lot of light



A little light



Finished

Challenge
Change the sensor from light to sound or temperature.

Music Machine

Start

Using a couple of sensors, this project with change the pitch of a piezo buzzer to create a simple musical instrument.

Modules

Gather the following parts to complete this project.

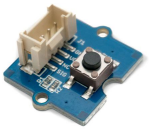
Parts

All Parts x Qty

Piezo Buzzer x 1

Rotary Potentiometer x 1

Light x 1



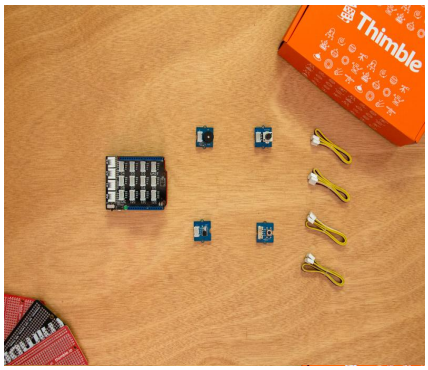
Button x 1

Cable x 4

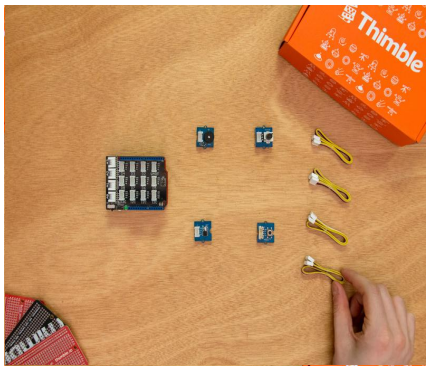
Light based Theremin

A theremin is an electronic musical instrument that is controlled by moving your hands near the machine without actually touching it. Here we'll make something like that with a light sensor.

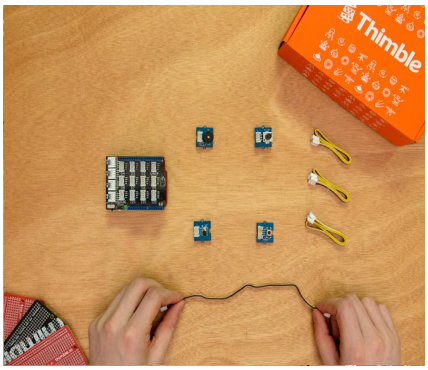
Take a cable and unwrap it. Plug one side into the buzzer socket and the other into **Digital** socket D6 to start making music. Take another cable and attach one end to the light sensor and the other to **Analog** socket A0.



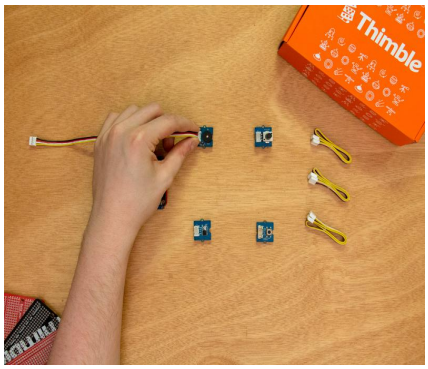
All the parts you'll need



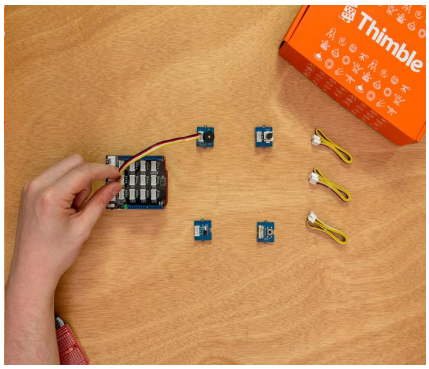
Take a cable...



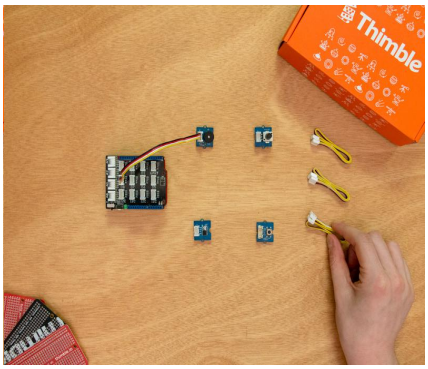
... and unwrap it



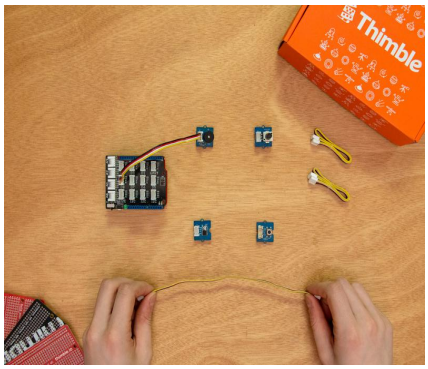
Plug one side into the buzzer socket



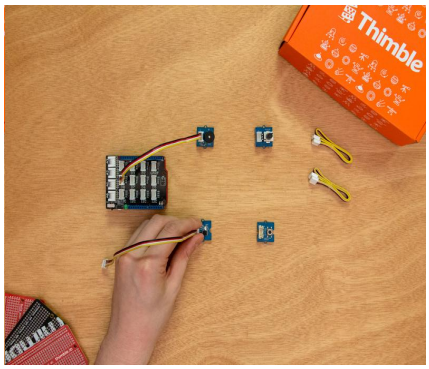
The other into any Digital socket



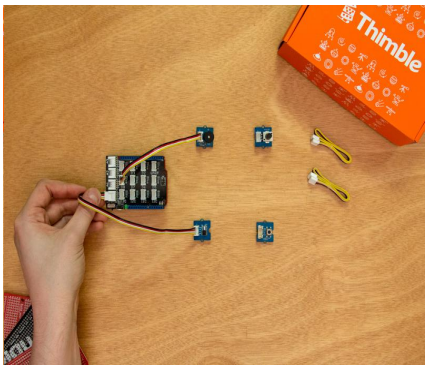
Take a cable...



... and unwrap it



Plug one side into the light sensor socket



The other into any Analog socket

Upload

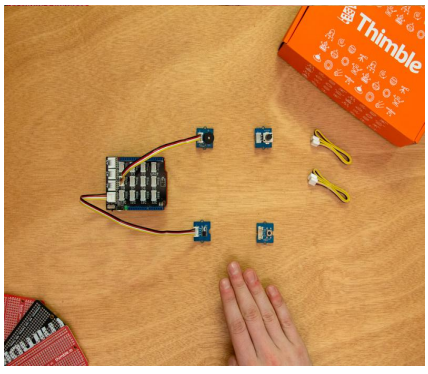
Upload the code the below. This tutorial uses **Analog** socket A0 and **Digital** socket 6. If you are using a different socket update the code after copying it.

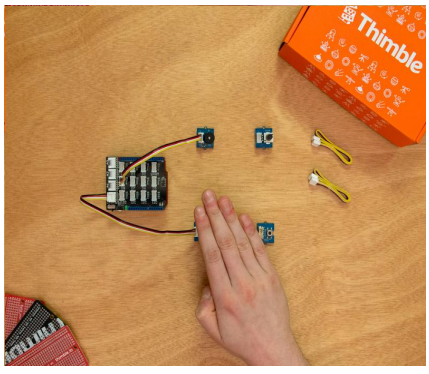
Copy to clipboard

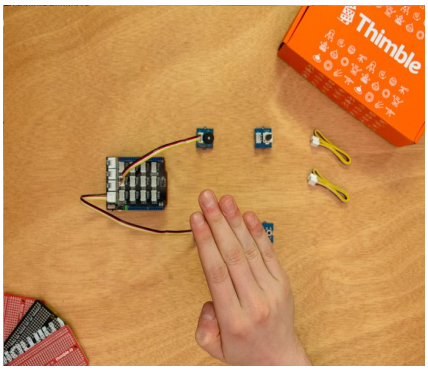
```
#define NOTE_C3  131
#define NOTE_CS3 139
#define NOTE_D3  147
#define NOTE_DS3 156
#define NOTE_E3  165
#define NOTE_F3  175
#define NOTE_FS3 185
#define NOTE_G3  196
#define NOTE_GS3 208
#define NOTE_A3  220
#define NOTE_AS3 233
#define NOTE_B3  247

int notes[7] = {NOTE_C3, NOTE_D3, NOTE_E3,
                NOTE_F3, NOTE_G3, NOTE_A3,
                NOTE_B3
               };

#define buzzerSocket 6
#define lightSensorSocket A0

int lightMin = 350;  // <-- This should be the lowest value that you get from your ligh
int lightMax = 750;  // <-- This should be the highest value that you get from your lig
int freq = 0;

void setup() {

}

void loop() {
  freq = map(freq, lightMin, lightMax, 0, 6);  // Convert reading to note number betwee
  freq = constrain(freq, 0, 6);  // Make sure that the value stays between 0 and 6

  tone(buzzerSocket, notes[freq]);
}
```

Observe

There isn't a lot of control here. It plays notes all the time. So let's add some.

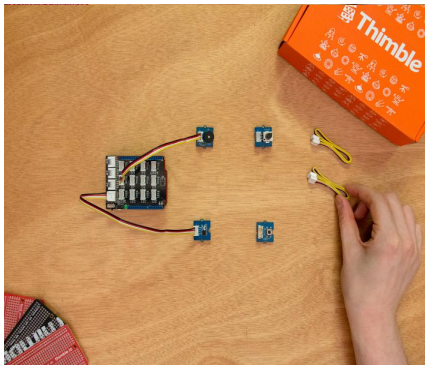




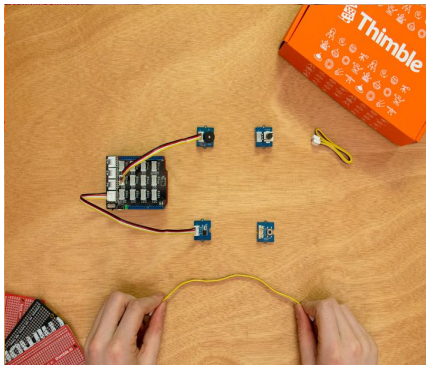


A Better Theremin

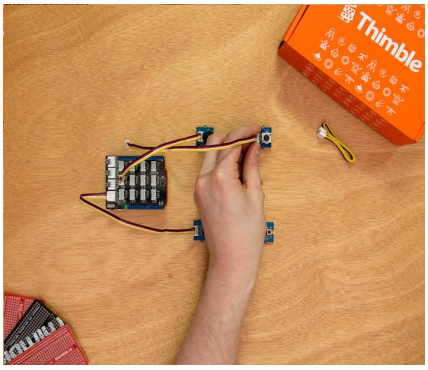
Take a cable and unwrap it. Plug one side into the rotary potentiometer and the other into **Analog** socket A1. Take another cable and attach one end to a button and the other to **Digital** socket D4.



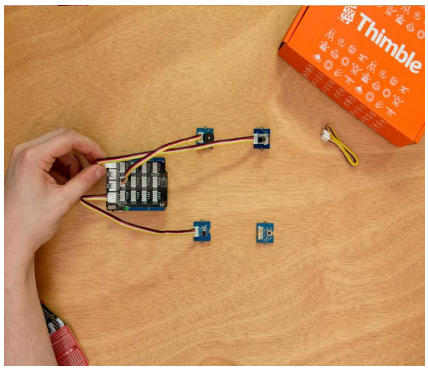
Take a cable...



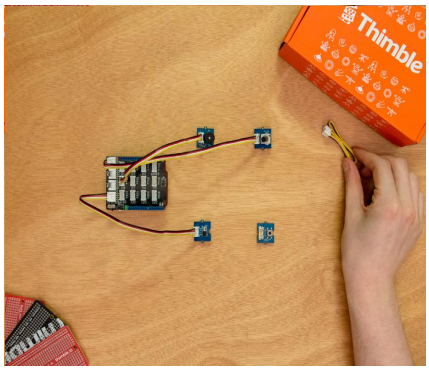
... and unwrap it



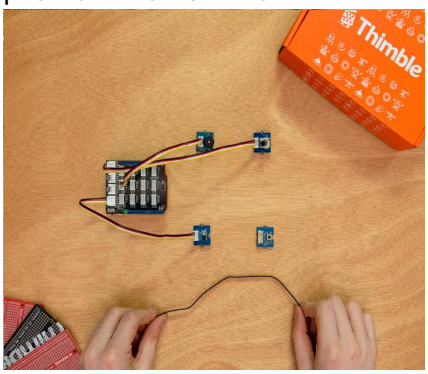
Plug one side into the rotary potentiometer socket



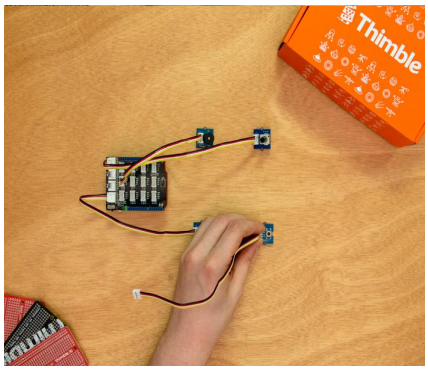
The other into Analog socket A1



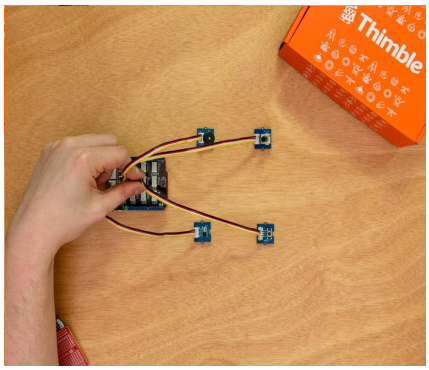
Take a cable...



... and unwrap it



Plug one side into the button socket



The other into any Digital socket D4

Upload

Upload the code below. This tutorial uses **Analog** socket A1 and **Digital** socket 4. If you are using a different socket update the code after copying it.

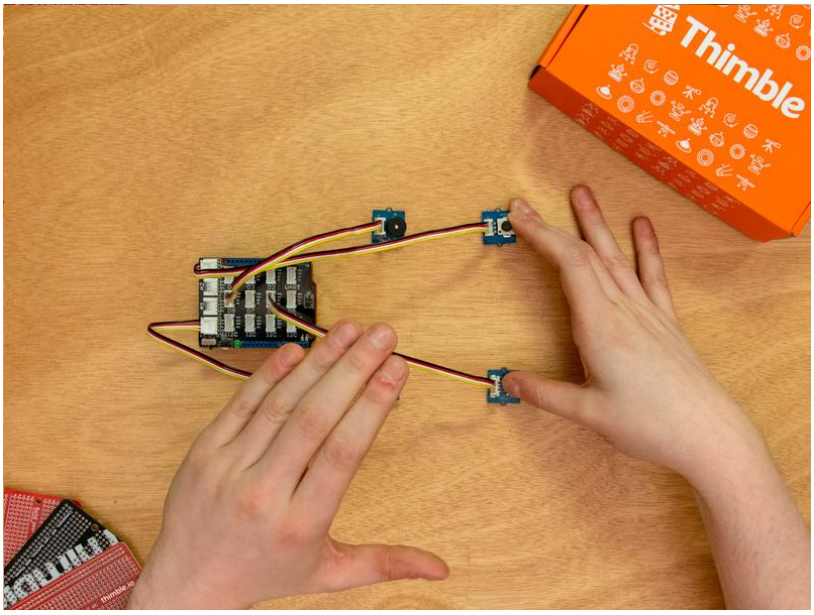
```
#define NOTE_B0  31
#define NOTE_C1  33
#define NOTE_CS1 35
#define NOTE_D1  37
#define NOTE_DS1 39
#define NOTE_E1  41
#define NOTE_F1  44
#define NOTE_FS1 46
#define NOTE_G1  49
#define NOTE_GS1 52
#define NOTE_A1  55
#define NOTE_AS1 58
#define NOTE_B1  62
#define NOTE_C2  65
#define NOTE_CS2 69
#define NOTE_D2  73
#define NOTE_DS2 78
#define NOTE_E2  82
#define NOTE_F2  87
#define NOTE_FS2 93
#define NOTE_G2  98
#define NOTE_GS2 104
#define NOTE_A2  110
#define NOTE_AS2 117
#define NOTE_B2  123
#define NOTE_C3  131
#define NOTE_CS3 139
#define NOTE_D3  147
#define NOTE_DS3 156
#define NOTE_E3  165
#define NOTE_F3  175
#define NOTE_FS3 185
#define NOTE_G3  196
#define NOTE_GS3 208
#define NOTE_A3  220
#define NOTE_AS3 233
#define NOTE_B3  247
#define NOTE_C4  262
#define NOTE_CS4 277
#define NOTE_D4  294
#define NOTE_DS4 311
#define NOTE_E4  330
#define NOTE_F4  349
#define NOTE_FS4 370
#define NOTE_G4  392
#define NOTE_GS4 415
#define NOTE_A4  440
#define NOTE_AS4 466
#define NOTE_B4  494
#define NOTE_C5  523
#define NOTE_CS5 554
```

```
#define NOTE_D5  587
#define NOTE_DS5 622
#define NOTE_E5  659
#define NOTE_F5  698
#define NOTE_FS5 740
#define NOTE_G5  784
#define NOTE_GS5 831
#define NOTE_A5  880
#define NOTE_AS5 932
#define NOTE_B5  988
#define NOTE_C6  1047
#define NOTE_CS6 1109
#define NOTE_D6  1175
#define NOTE_DS6 1245
#define NOTE_E6  1319
#define NOTE_F6  1397
#define NOTE_FS6 1480
#define NOTE_G6  1568
#define NOTE_GS6 1661
#define NOTE_A6  1760
#define NOTE_AS6 1865
#define NOTE_B6  1976
#define NOTE_C7  2093
#define NOTE_CS7 2217
#define NOTE_D7  2349
#define NOTE_DS7 2489
#define NOTE_E7  2637
#define NOTE_F7  2794
#define NOTE_FS7 2960
#define NOTE_G7  3136
#define NOTE_GS7 3322
#define NOTE_A7  3520
#define NOTE_AS7 3729
#define NOTE_B7  3951
#define NOTE_C8  4186
#define NOTE_CS8 4435
#define NOTE_D8  4699
#define NOTE_DS8 4978

#define buttonSocket 4
#define buzzerSocket 6
#define lightSensorSocket A0
#define dialSocket A1

int noteMatrix[5][7] = {
  {NOTE_C2, NOTE_D2, NOTE_E2, NOTE_F2, NOTE_G2, NOTE_A2, NOTE_B2},
  {NOTE_C3, NOTE_D3, NOTE_E3, NOTE_F3, NOTE_G3, NOTE_A3, NOTE_B3},
  {NOTE_C4, NOTE_D4, NOTE_E4, NOTE_F4, NOTE_G4, NOTE_A4, NOTE_B4},
  {NOTE_C5, NOTE_D5, NOTE_E5, NOTE_F5, NOTE_G5, NOTE_A5, NOTE_B5},
  {NOTE_C6, NOTE_D6, NOTE_E6, NOTE_F6, NOTE_G6, NOTE_A6, NOTE_B6}
};
```

```
int freq = 0;
int scale = 0;
int lightMin = 350;
int lightMax = 750;

void setup() {
  // put your setup code here, to run once:
  pinMode(buttonSocket, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  freq = analogRead(lightSensorSocket);
  freq = map(freq, lightMin, lightMax, 0, 6);
  freq = constrain(freq, 0, 6);
  scale = analogRead(dialSocket);
  scale = map(scale, 0, 1024, 0, 4);

  if (digitalRead(buttonSocket)) {
    tone(buzzerSocket, noteMatrix[freq][scale]);
  } else {
    noTone(buzzerSocket);
  }
}

```

Observe

Now by using the one hand over the light sensor and another on the potentiometer and button we have access to more notes. The button plays the note. The potentiometer changes the octave. The light sensor changes the note and you control the music!



Finished

Simon Says

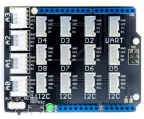
Start

Using many inputs and some random instructions, this project recreates the feel of the classic Simon Says game.

Modules

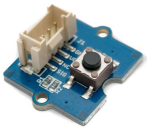
Gather the following parts to complete this project.

Parts



All Parts x Qty

LCD Display x 1



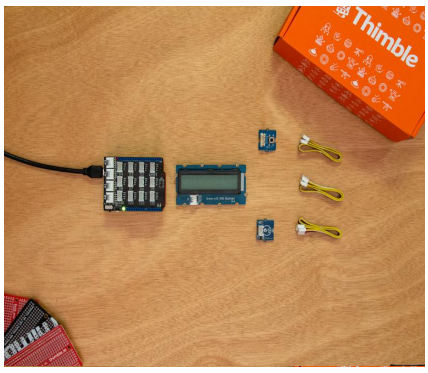
Button x 1

Touch x 1

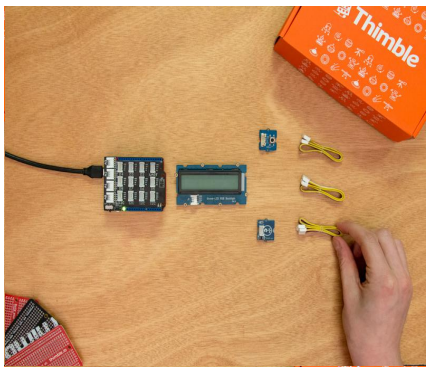
Cable x 4

OneButton

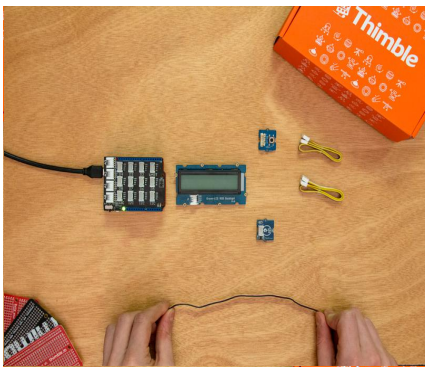
Take a cable and unwrap it. Plug one side into the LCD Display and the other into any **I2C** socket. Take another cable and attach one end to a button and the other to **Digital** socket D6.



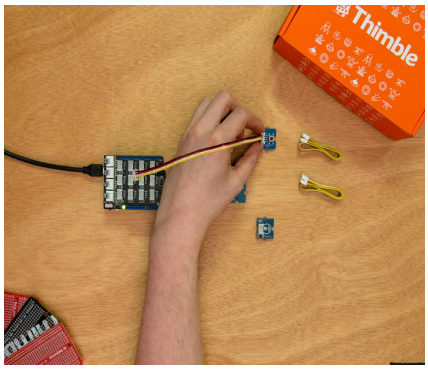
All the parts you'll need



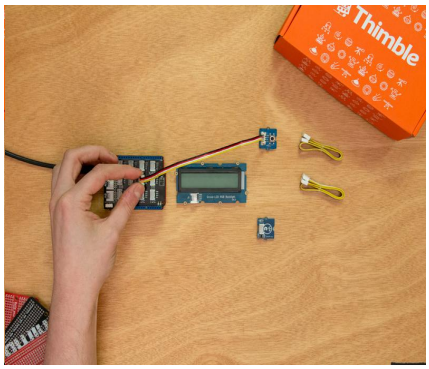
Take a cable...



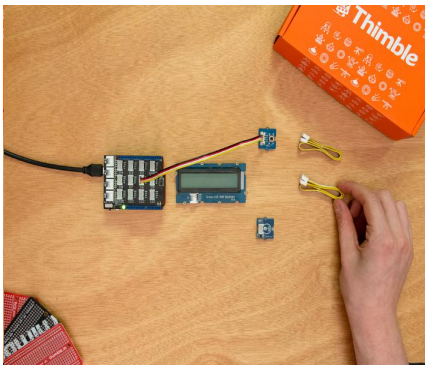
... and unwrap it



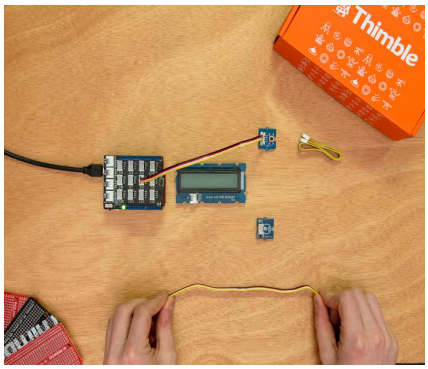
Plug one side into the LCD Display socket



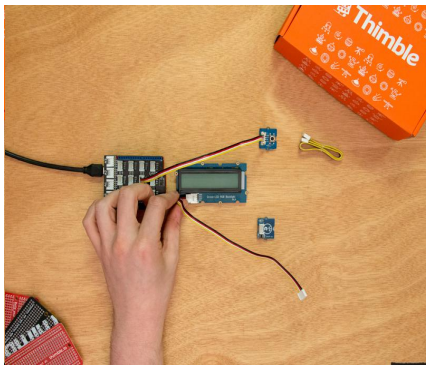
The other into any I2C socket



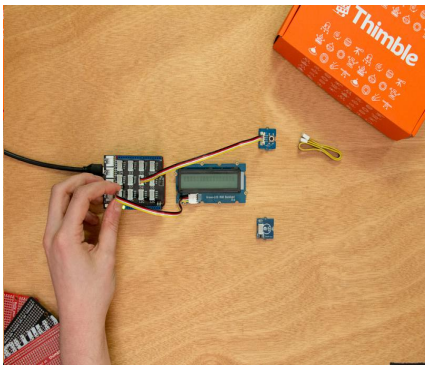
Take a cable...



... and unwrap it



Plug one side into the button socket



The other into any Digital socket

Upload

Upload the code below. This tutorial uses **Digital** socket 6. If you are using a different socket update the code after copying it.

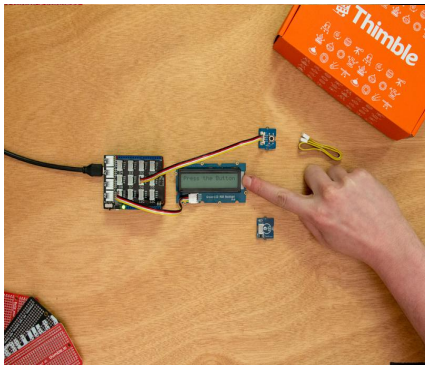
Copy to clipboard

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

#define buttonSocket 6

rgb_lcd lcd;
const int colorR = 0;
const int colorG = 0;
const int colorB = 0;

OneButton button1(buttonSocket, false);

void setup() {
  lcd.begin(16, 2);
  lcd.setRGB(colorR, colorG, colorB);

  button1.attachClick(click1);
  button1.attachDoubleClick(doubleclick1);
  button1.attachLongPressStart(longPressStart1);
  button1.attachLongPressStop(longPressStop1);
  button1.attachDuringLongPress(longPress1);
}

void loop() {
  button1.tick();
  lcd.setCursor(0, 0);
  lcd.print("Press the Button");
}


void click1() {
  lcd.setCursor(0, 0);
  lcd.print("Button click    ");
  delay(1000);
}


void doubleclick1() {
  lcd.setCursor(0, 0);
  lcd.print("Button Doubleclk");
  delay(1000);
}


void longPressStart1() {
  lcd.setCursor(0, 0);
  lcd.print("Longpress Start ");
  delay(1000);
}


```

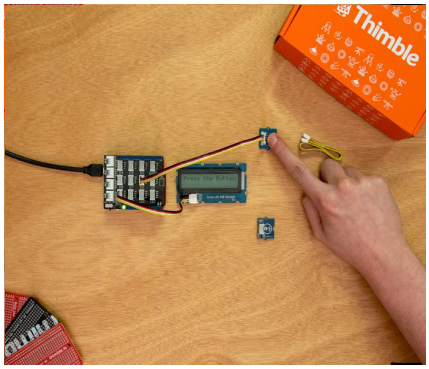
```
52 void longPress1() {
53   lcd.setCursor(0, 0);
54   lcd.print("Longpress Press ");
55   delay(1000);
56 }
57
58
59 void longPressStop1() {
60   lcd.setCursor(0, 0);
61   lcd.print("Longpress Stop  ");
62   delay(1000);
63 }
```

Observe

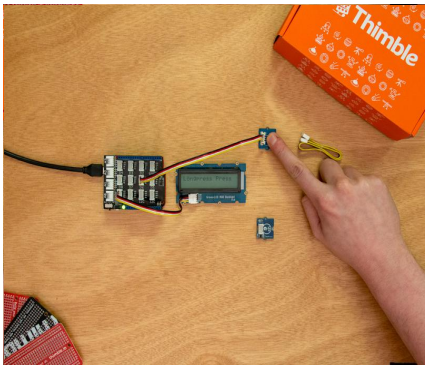
Press the button once. Press the button twice. Hold it down. Check out all the features of the OneButton library. What was once a simple button now has many uses.



Press the button



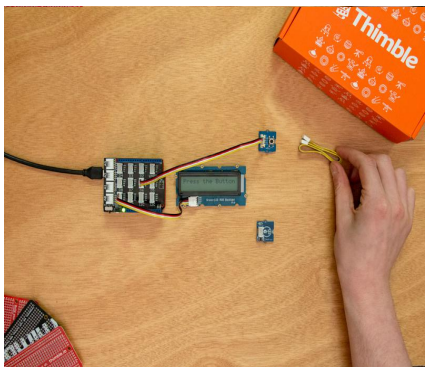
Let's do it



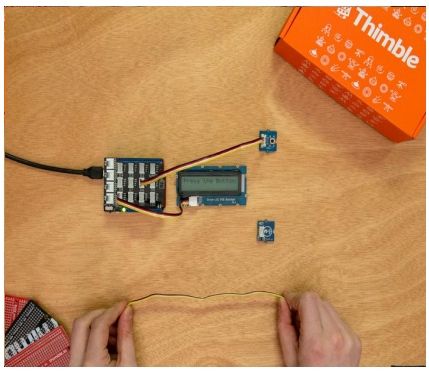
Holding the button down

Simple Simon

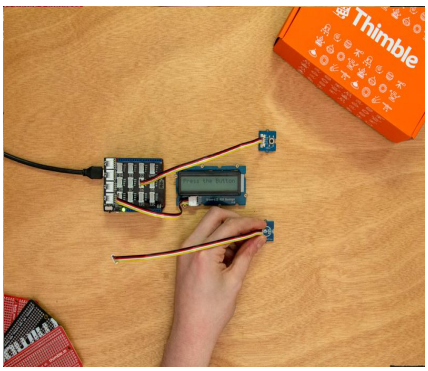
Take a cable and attach one end to a touch sensor and the other to **Digital** socket D7.



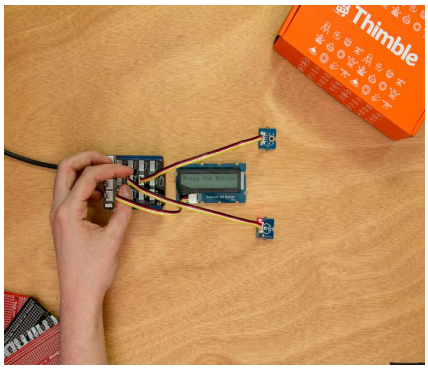
Take a cable...



... and unwrap it



Plug one side into the touch socket



The other into Digital socket D7

Upload

Upload the code below. This tutorial uses **Digital** socket 7. If you are using a different socket update the code after copying it.

Copy to clipboard

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

#define buttonSocket 6
#define touchSocket 7

rgb_lcd lcd;
const int colorR = 0;
const int colorG = 0;
const int colorB = 0;

OneButton button(buttonSocket, false);
OneButton touch(touchSocket, false);

String actions[] = {"buttonSingle", "buttonDouble", "touchSingle", "touchDouble"};
String currentAction = "";
String actionTaken = "";


void setup() {
  lcd.begin(16, 2);
  lcd.setRGB(colorR, colorG, colorB);

  button.attachClick(buttonSingle);
  button.attachDoubleClick(buttonDouble);
  touch.attachClick(touchSingle);
  touch.attachDoubleClick(touchDouble);

  currentAction = actions[random(0, 4)];

  lcd.setCursor(0, 0);
  lcd.print("Action To Do:");
  lcd.setCursor(0, 1);
  lcd.print(currentAction);
}

void loop() {
  button.tick();
  touch.tick();
}


void buttonSingle() {
  actionTaken = "buttonSingle";
  correctCheck();
}

void buttonDouble() {
  actionTaken = "buttonDouble";
  correctCheck();
}
```

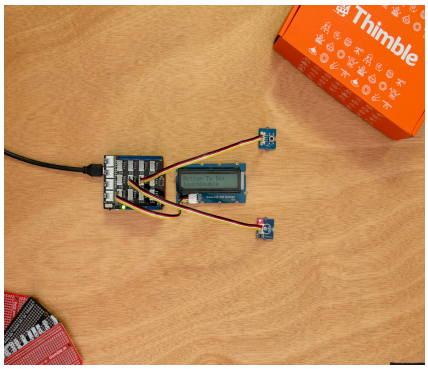
```
52
53 void touchSingle() {
54   actionTaken = "touchSingle";
55   correctCheck();
56 }
57
58 void touchDouble() {
59   actionTaken = "touchDouble";
60   correctCheck();
61 }
62
63 void correctCheck() {
64   if (currentAction == actionTaken) {
65     clearScreen();
66     lcd.setRGB(0, 255, 0);
67     lcd.setCursor(0, 0);
68     lcd.print("Correct!");
69     lcd.setCursor(0, 1);
70     lcd.print("Picking Action");
71     delay(2000);
72     lcd.setRGB(colorR, colorG, colorB);
73     currentAction = actions[random(0, 4)];
74     clearScreen();
75     lcd.setCursor(0, 0);
76     lcd.print("Action To Do:");
77     lcd.setCursor(0, 1);
78     lcd.print(currentAction);
79   } else {
80     lcd.setRGB(255, 0, 0);
81     delay(500);
82     lcd.setRGB(colorR, colorG, colorB);
83   }
84 }
85
86 void clearScreen() {
87   lcd.setCursor(0, 0);
88   lcd.print("                ");
89   lcd.setCursor(0, 1);
90   lcd.print("                ");
91 }
```

Observe

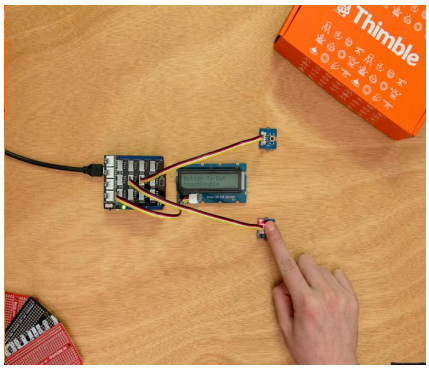
Follow the actions on screen. Green for correct and red for try again.

Modify

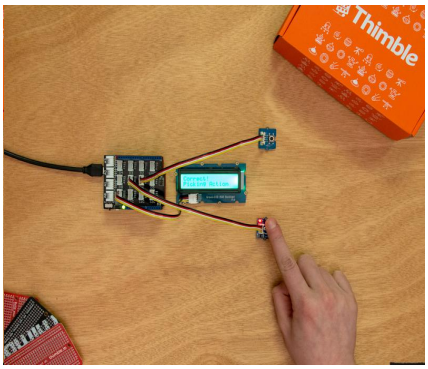
You can change the colors that the LCD Display uses.



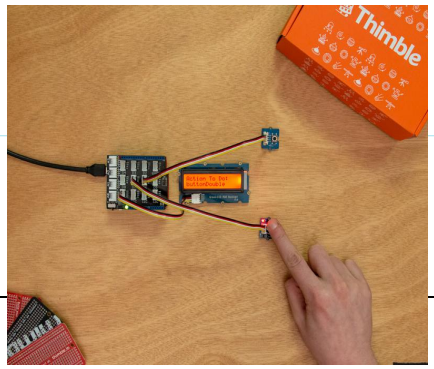
An action is displayed



Let's do it



Correct!



Wrong!

Challenge

Try adding in some of the other uses of the OneButton library as possible actions for *Simple Simon*. Or, try adding a scoring system.

Datasheets

Info for each Modules

Button (https://d2j2m4p6r3pg95.cloudfront.net/module_files/creator-set/assets/datasheets/pdf/button.pdf)

Buzzer (https://d2j2m4p6r3pg95.cloudfront.net/module_files/creator-set/assets/datasheets/pdf/buzzer.pdf)

LCD Display (https://d2j2m4p6r3pg95.cloudfront.net/module_files/creator-set/assets/datasheets/pdf/lcd.pdf)

LED (https://d2j2m4p6r3pg95.cloudfront.net/module_files/creator-set/assets/datasheets/pdf/led.pdf)

Light Sensor (https://d2j2m4p6r3pg95.cloudfront.net/module_files/creator-set/assets/datasheets/pdf/light.pdf)

Relay (https://d2j2m4p6r3pg95.cloudfront.net/module_files/creator-set/assets/datasheets/pdf/relay.pdf)

Rotary Potentiometer (https://d2j2m4p6r3pg95.cloudfront.net/module_files/creator-set/assets/datasheets/pdf/rotaryPotentiometer.pdf)

Servo Motor (https://d2j2m4p6r3pg95.cloudfront.net/module_files/creator-set/assets/datasheets/pdf/servo.pdf)

Sound Sensor (https://d2j2m4p6r3pg95.cloudfront.net/module_files/creator-set/assets/datasheets/pdf/sound.pdf)

Temperature Sensor (https://d2j2m4p6r3pg95.cloudfront.net/module_files/creator-set/assets/datasheets/pdf/temp.pdf)

Touch Sensor (https://d2j2m4p6r3pg95.cloudfront.net/module_files/creator-set/assets/datasheets/pdf/touch.pdf)