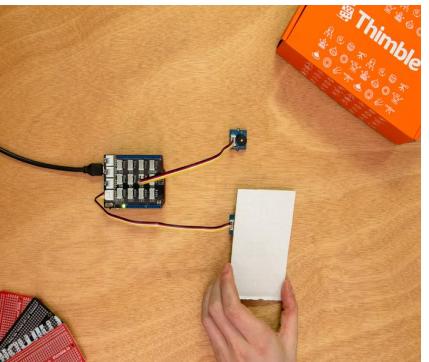
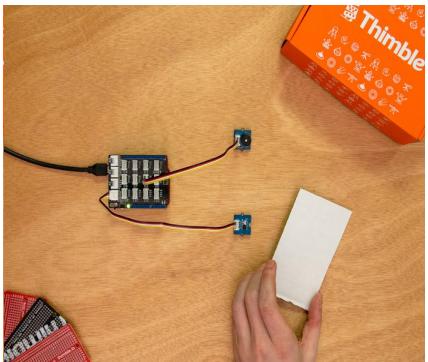


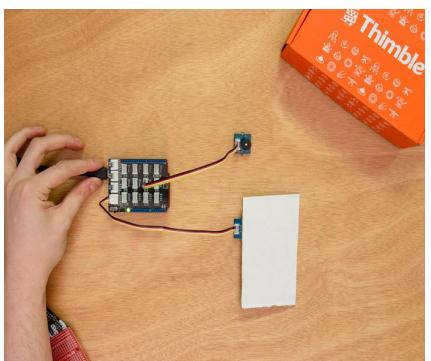
Covered alarm



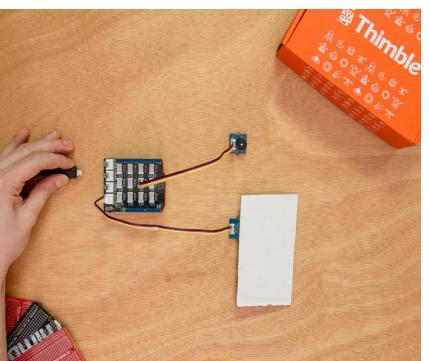
Let's check it



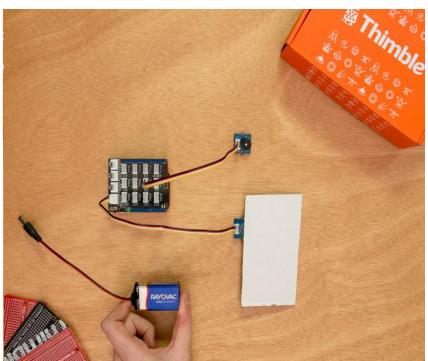
Yeah it works



Grab the USB cable



Unplug the USB cable



A battery with the included cable



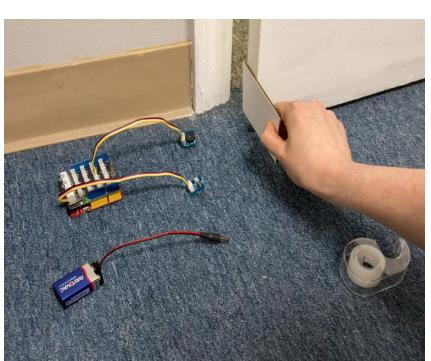
Let's go install



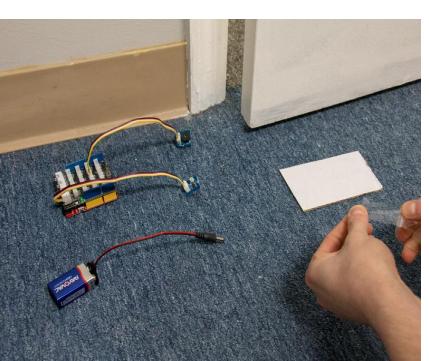
I've found a door



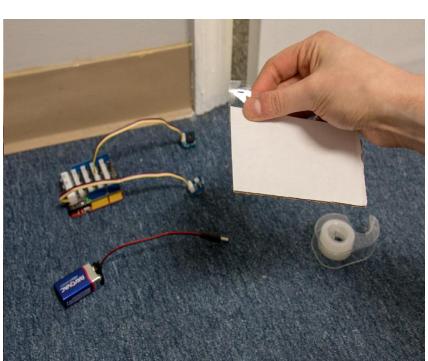
Cardboard? Check



Line it up with the bottom of the door

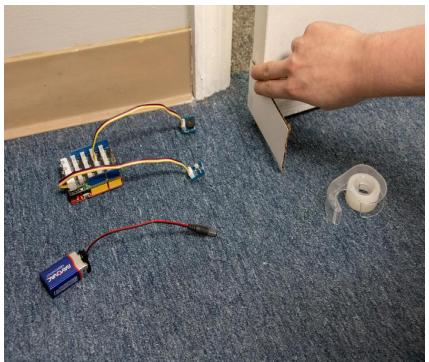


Take a piece of tape

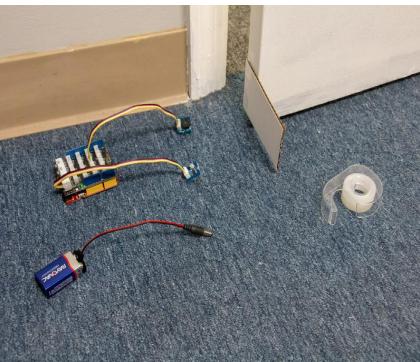


Attach it to the cardboard

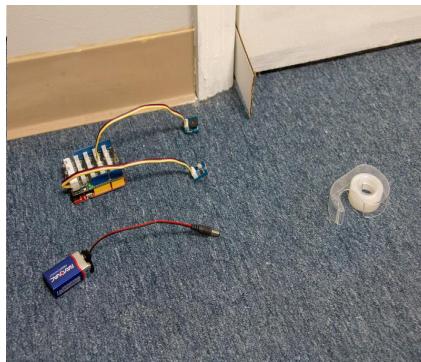
## Thermometer



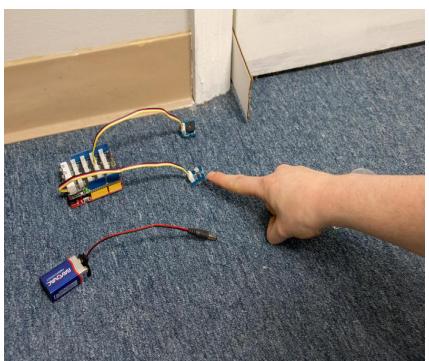
Tape it to the door



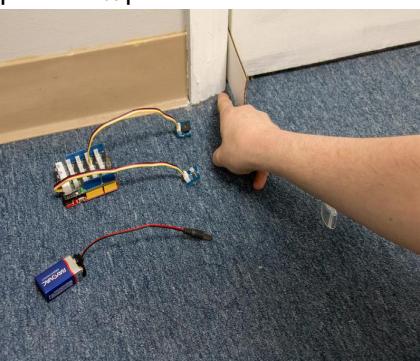
You might need more than one piece of tape



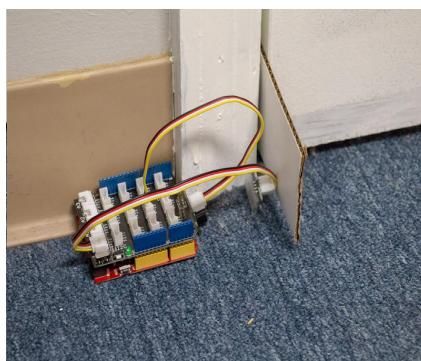
Make sure the door closes



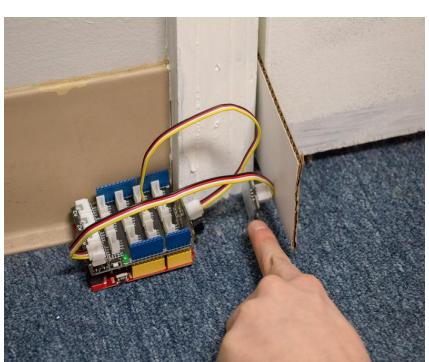
The light sensor...



... should face the cardboard here



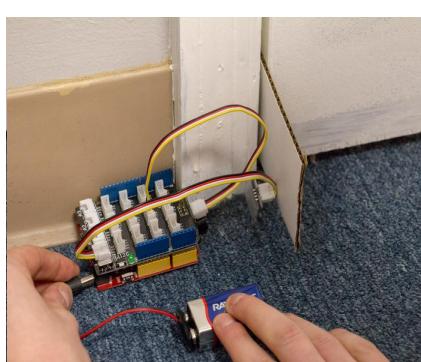
Just like this



Light sensor is right up against the cardboard trigger



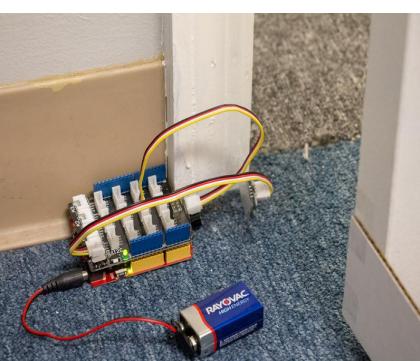
Take the battery...



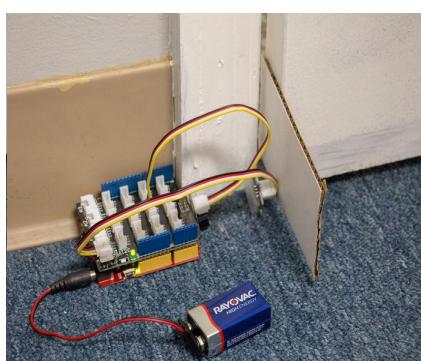
... and plug it in



Green led means the power is on



Open the door and you should hear the chirps



Closing the door resets the alarm

## Start %

Let's put together a thermometer to sense the temperature and hook that up to a display.

# Modules %

Gather the following parts to complete this project.

## Parts



All Parts x Qty



Temperature x 1



Button x 1



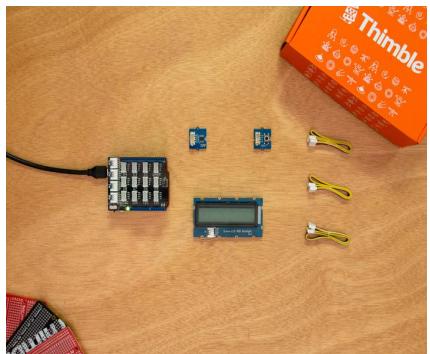
LCD Display x 1



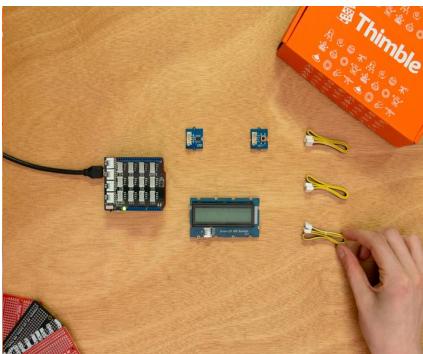
Cable x 3

## Temperature %

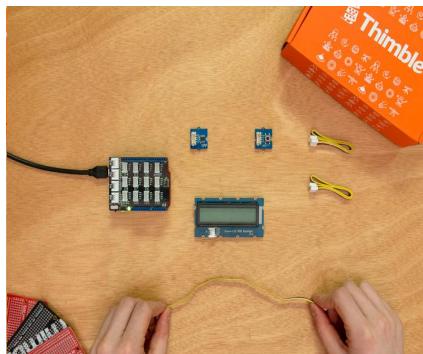
Take a cable and unwrap it. Plug one side into the temperature sensor socket and the other into **Analog** socket A0.



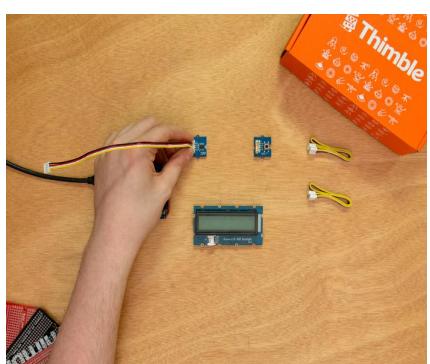
All the parts you'll need



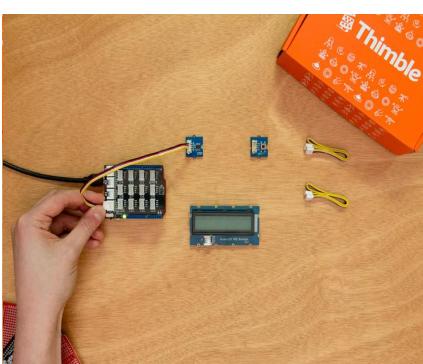
Take a cable...



... and unwrap it



Plug one side into the temperature sensor socket



... and the other into Analog socket A0

## Upload

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

 Copy to clipboard

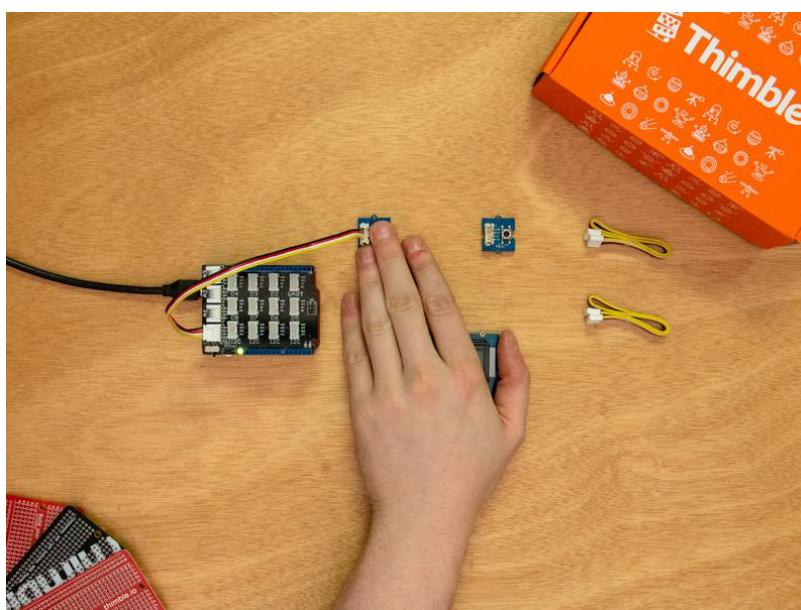
```
1 #include <math.h>
2
3 //Change here if you're using a different socket
4 #define tempSensorSocket A0
5
6 int temp;
7
8 const int B = 4275; // B value of the thermistor;
9 const int R0 = 100000; // R0 = 100k;
10
11 void setup()
12 {
13     Serial.begin(9600);
14 }
15
16 void loop()
17 {
18     temp = 1.0/(log(1023.0/analogRead(tempSensorSocket)-1.0)/B+1/298.15)-273.15;
19     Serial.println(temp);
20 }
```

## Observe

Open up the Serial Monitor and check out the current temperature in Celsius.

## Modify

Move the temperature sensor around and check out the different temperatures. Try putting it in a hot place and a cold place.



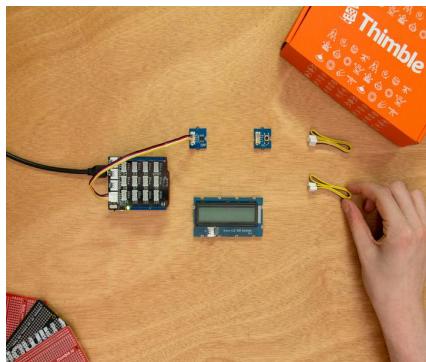
## Body temperature will bring it up

### Experiment

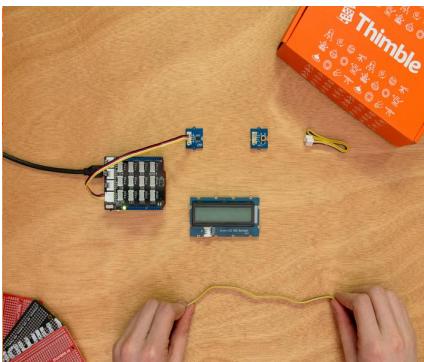
See if you can detect the small changes in temperature within a room. The higher you place the sensor in the room, usually the higher the temperature because heat rises.

### Button %

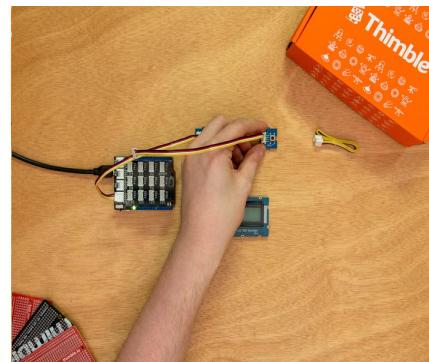
Take a cable and unwrap it. Plug one side into the button and the other into **Digital** socket D4.



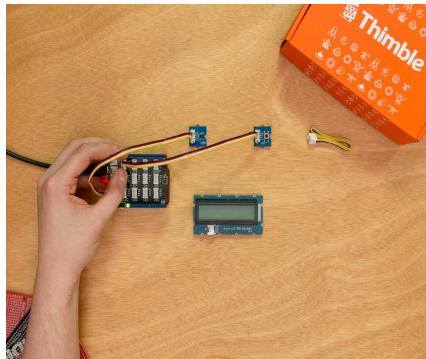
Take a cable...



... and unwrap it



Plug one side into the button socket



... and the other into Digital socket D4

### Upload

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

Copy to clipboard

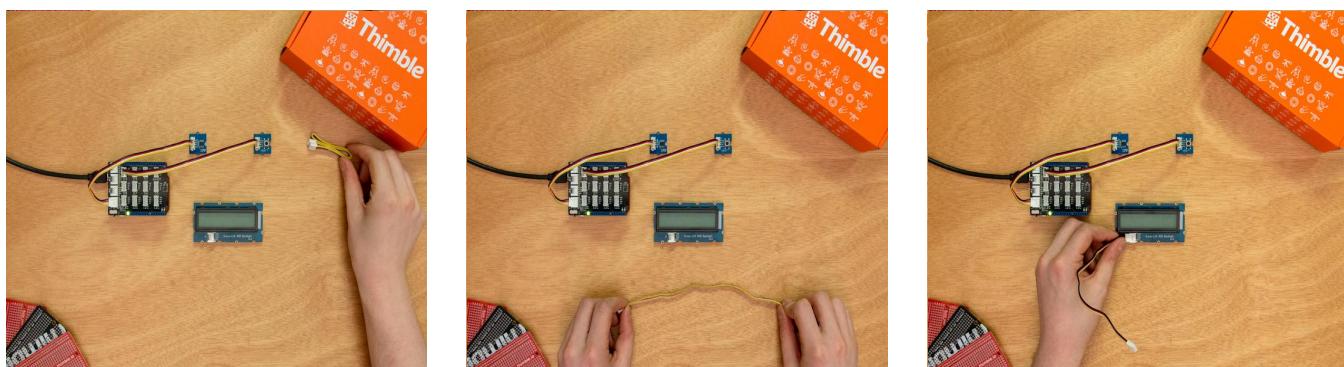
```
1 #include <math.h>
2
3 //If you use different sockets change them below
4 #define buttonSocket 4
5 #define tempSensorSocket A0
6 int temp;
7 int tempC;
8
9 const int B = 4275; // B value of the thermistor;
10 const int R0 = 100000; // R0 = 100k;
11
12 void setup()
13 {
14     pinMode(buttonSocket, INPUT);
15     Serial.begin(9600);
16 }
17
18 void loop()
19 {
20     tempC = 1.0/(log(1023.0/analogRead(tempSensorSocket)-1.0)/B+1/298.15)-273.15;
21     if (digitalRead(buttonSocket)) {
22         Serial.println((far(tempC)));
23     } else {
24         Serial.println(tempC);
25     }
26 }
27
28 int far(int temp) {
29     return temp * 1.8 + 32;
30 }
```

## Observe

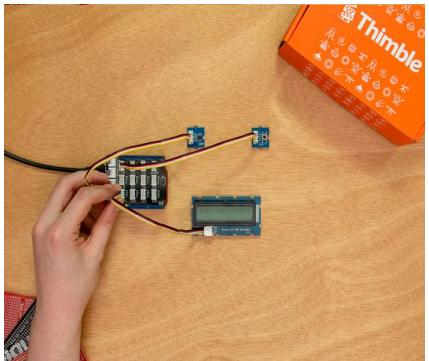
Open up the Serial Monitor and press the button. Watch the temperature reading change from Celsius to Fahrenheit. With higher temperatures the difference between the two units is larger.

## LCD Display %

Take a cable and unwrap it. Plug one side into the LCD Display and the other into any **I<sub>2</sub>C** socket.



Take a cable...



... and unwrap it

Plug one side into the LCD socket

... and the other into any I2C  
socket

## Upload

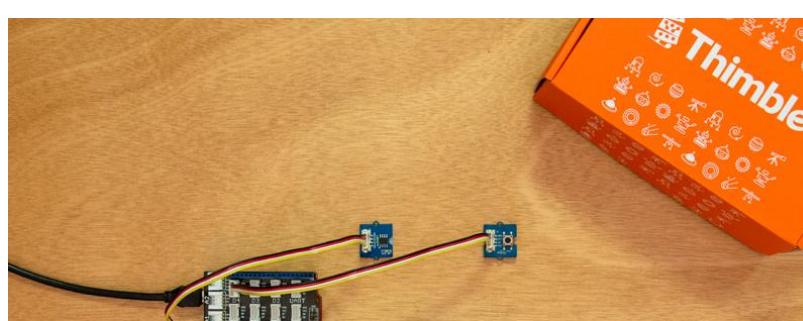
Upload the code shown below.

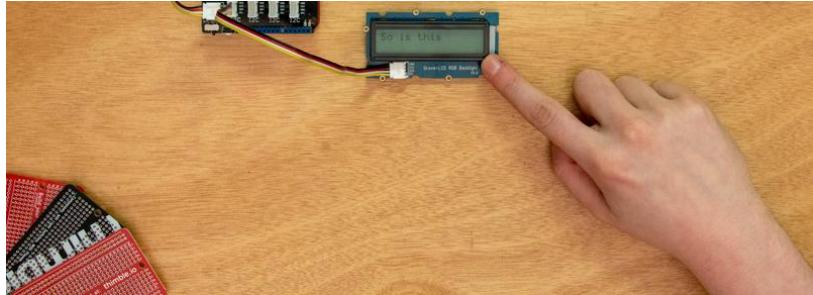
Copy to clipboard

```
1 #include <Wire.h>
2 #include "rgb_lcd.h"
3
4 rgb_lcd lcd;
5 const int colorR = 0;
6 const int colorG = 0;
7 const int colorB = 0;
8
9 void setup()
10 {
11 lcd.begin(16, 2);
12 lcd.setRGB(colorR, colorG, colorB);
13 }
14
15 void loop()
16 {
17 lcd.setRGB(colorR, colorG, colorB);
18 lcd.setCursor(0,0);
19 lcd.print("This is a");
20 lcd.setCursor(0,1);
21 lcd.print("message");
22 delay(1000);
23 clearScreen();
24 lcd.setRGB(colorR, colorG, colorB);
25 lcd.setCursor(0,0);
26 lcd.print("So is this");
27 lcd.setCursor(0,1);
28 lcd.print("");
29 delay(1000);
30 clearScreen();
31 }
32
33 void clearScreen(){
34 lcd.setCursor(0,0);
35 lcd.print("          ");
36 lcd.setCursor(0,1);
37 lcd.print("          ");
38 }
```

## Observe

Take a look at the two messages.

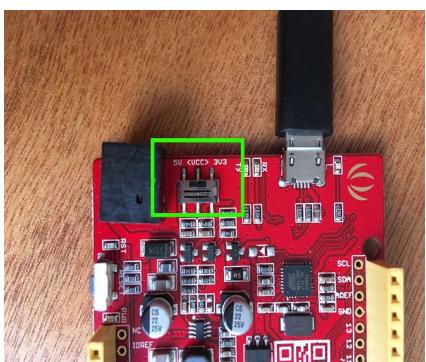




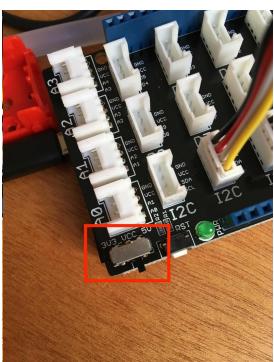
**One of the messages**

## Is The Screen Not Working?

Double check that the switches found on the Seeeduino and base shield are set to **5V** and not 3V3.



Switch on the Seeeduino



Switch on the Expansion Board

## Modify

Change the messages and the delays associated with them.

## All together %

By combining all these modules we can achieve our goal of a digital display thermometer.

## Upload

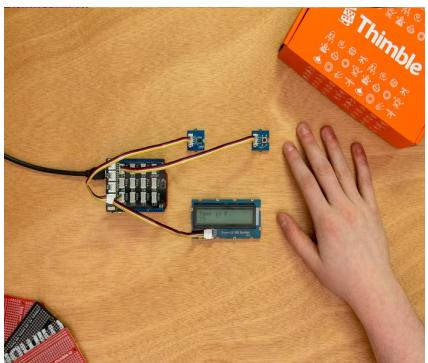
Upload the code shown below.

Copy to clipboard

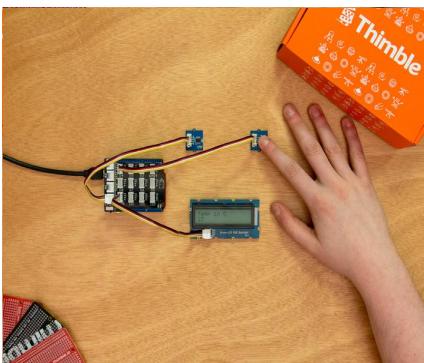
```
1 #include <math.h>
2 #include <Wire.h>
3 #include "rgb_lcd.h"
4
5 //If you use different sockets change them below
6 #define buttonSocket 4
7 #define tempSensorSocket A0
8
9 int temp;
10
11 int tempC;
12
13 int far(int temp) {
14     return temp * 1.8 + 32;
15 }
16
17 const int B = 4275; // B value of the thermistor;
18 const int R0 = 100000; // R0 = 100k;
19
20 rgb_lcd lcd;
21 const int colorR = 0;
22 const int colorG = 0;
23 const int colorB = 0;
24
25 void setup()
26 {
27     pinMode(buttonSocket, INPUT);
28     lcd.begin(16, 2);
29     lcd.setRGB(colorR, colorG, colorB);
30 }
31
32 void loop()
33 {
34     tempC = 1.0 / (log(1023.0 / analogRead(tempSensorSocket) - 1.0) / B + 1 / 298.15) - 2
35     if (digitalRead(buttonSocket)) {
36         lcd.setRGB(colorR, colorG, colorB);
37         lcd.setCursor(0, 0);
38         lcd.print("Temp in C");
39         lcd.setCursor(0, 1);
40         lcd.print(tempC);
41         delay(100);
42     } else {
43         lcd.setRGB(colorR, colorG, colorB);
44         lcd.setCursor(0, 0);
45         lcd.print("Temp in F");
46         lcd.setCursor(0, 1);
47         lcd.print((far(tempC)));
48         delay(100);
49     }
50 }
```

## Observe

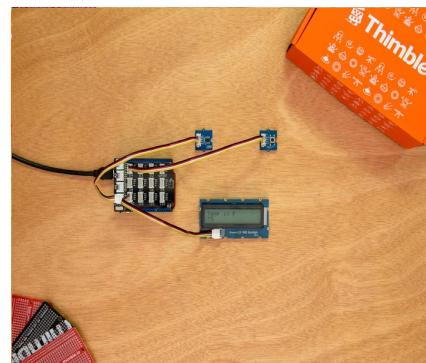
With the display working you can unplug your thermometer from your computer and bring it anywhere on battery power. Pressing the button still switches the units.



Ambient Temperature in Fahrenheit



Ambient Temperature in Celsius



Is it getting hot in here?

### ⚡ Challenge

Map the current temperature to color values displayed by the LCD Display. `lcd.setRGB(colorR, colorG, colorB);`

## Timer

### Start ⏱

This project yields a programmable timer.

### Modules ⚙️

Gather the following parts to complete this project.

#### Parts



All Parts x Qty



LCD Display x 1



Button x 1



Rotary potentiometer x 1

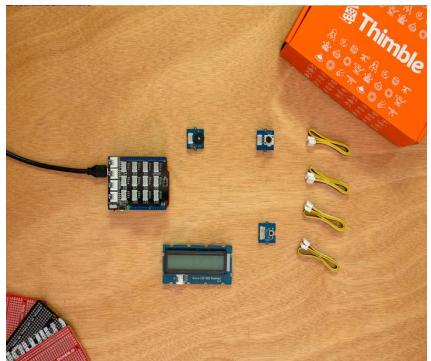


Piezo Buzzer x 1

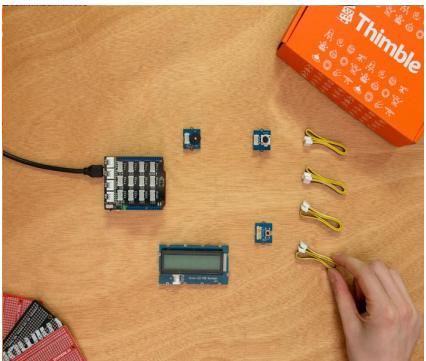
Cable x 4

## A Countdown %

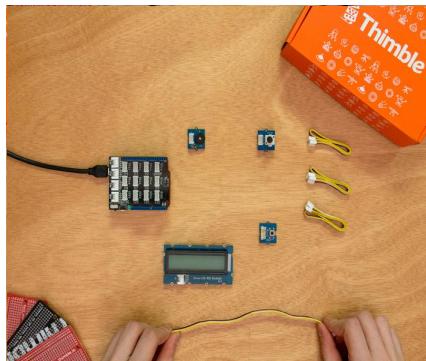
Let's start with the most simple timer. Take a cable and unwrap it. Plug one side into the LCD Display and the other into any **I2C** socket. Take another cable and unwrap it. Plug one side into the button socket and the other into **Digital** socket D6.



All the parts you'll need



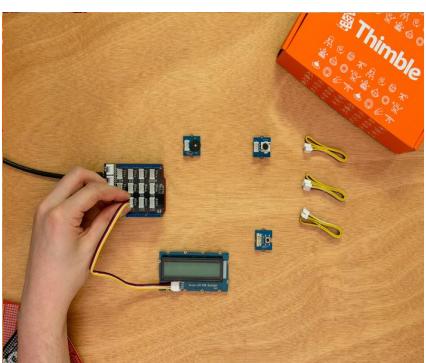
Take a cable...



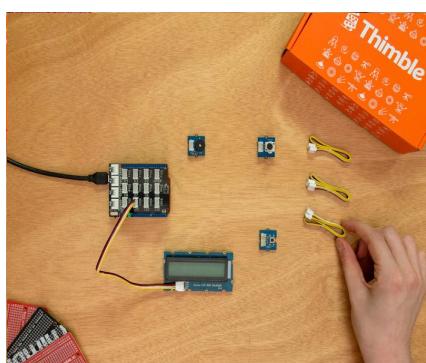
... and unwrap it



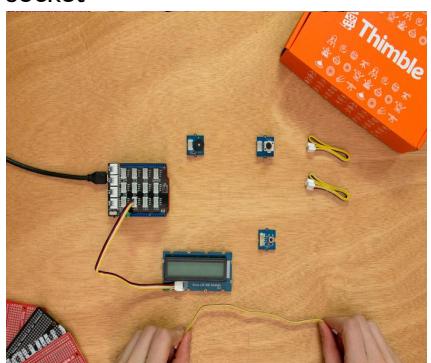
Plug one side into the LCD Display socket



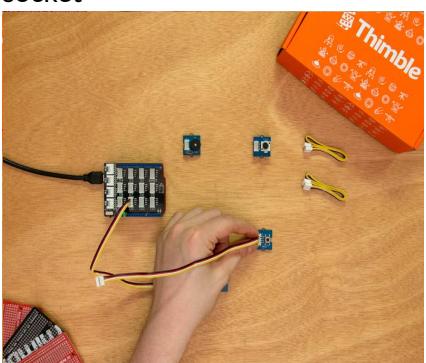
... and the other into any I2C socket



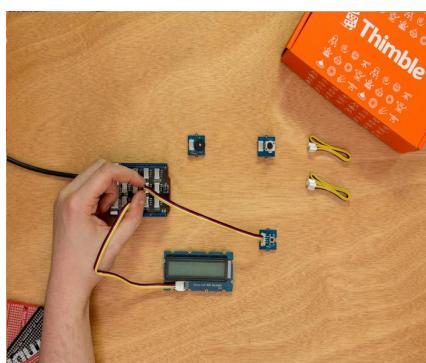
Take a cable...



... and unwrap it



Plug one side into the button socket



... and the other into Digital socket D6

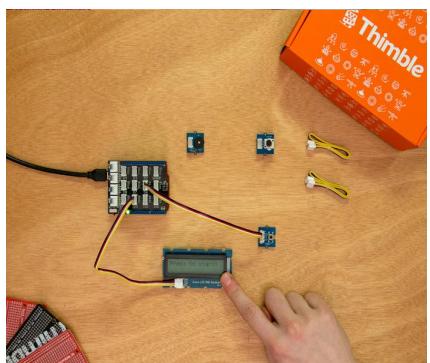
Upload

```
1 #include <Wire.h>
2 #include "rgb_lcd.h"
3
4 //Change here if you're using a different socket
5 #define buttonSocket 6
6
7 int milliseconds;
8 int Time;
9 int seconds;
10
11 int Seconds(int milliseconds) {
12     seconds = milliseconds / 1000;
13     return seconds;
14 }
15
16 rgb_lcd lcd;
17 const int colorR = 0;
18 const int colorG = 0;
19 const int colorB = 0;
20
21 void setup()
22 {
23     pinMode(buttonSocket, INPUT);
24     lcd.begin(16, 2);
25     lcd.setRGB(colorR, colorG, colorB);
26     Serial.begin(9600);
27
28 }
29
30 void loop()
31 {
32     Serial.println(Time);
33     lcd.setRGB(colorR, colorG, colorB);
34     lcd.setCursor(0, 0);
35     lcd.print("Press to start:");
36     lcd.setCursor(0, 1);
37     lcd.print(" ");
38     Time = 10;
39     if (digitalRead(buttonSocket)) {
40         while (Time > 0) {
41             lcd.setRGB(colorR, colorG, colorB);
42             lcd.setCursor(0, 0);
43             lcd.print("Time remaining:");
44             lcd.setCursor(0, 1);
45             lcd.print("      ");
46             lcd.setCursor(0, 1);
47             lcd.print(Time);
48             delay(1000);
49             Time = Time - 1;
50             Serial.println(Time);
51         }
52     }
53 }
```

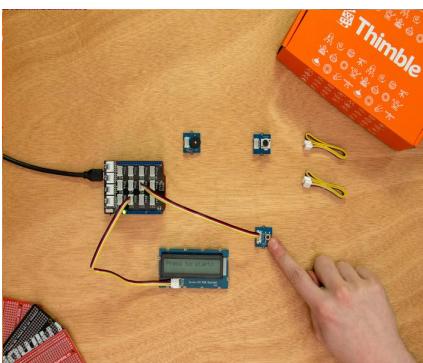
```
52     while (!digitalRead(buttonSocket)) {  
53         lcd.setRGB(colorR, colorG, colorB);  
54         lcd.setCursor(0, 0);  
55         lcd.print("Time is up!      ");  
56         lcd.setCursor(0, 1);  
57         lcd.print("      ");  
58     }  
59     delay(100);  
60 }  
61 }  
62 }
```

## Observe

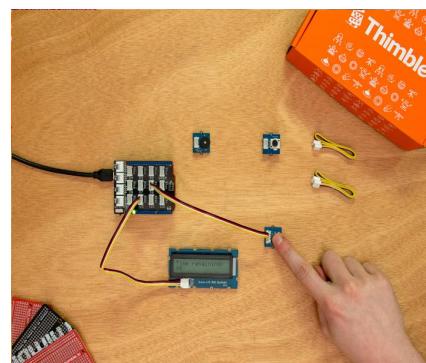
After pressing the button, the countdown will begin. The clock ticks down in seconds.



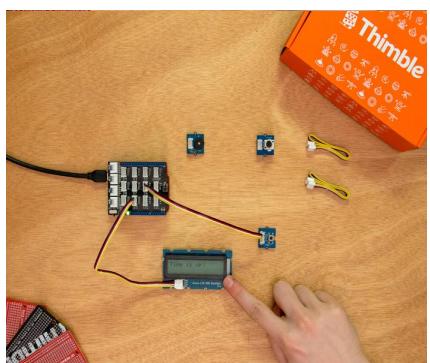
The screen shows the starting text



And with a press of the button...



... the countdown starts



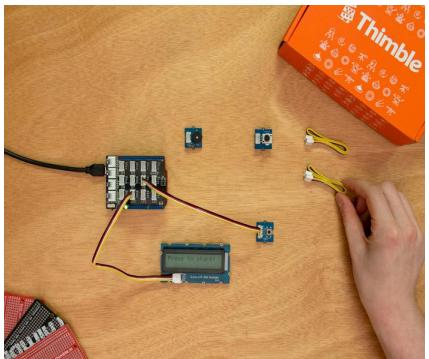
A message when the time is up

## Modify

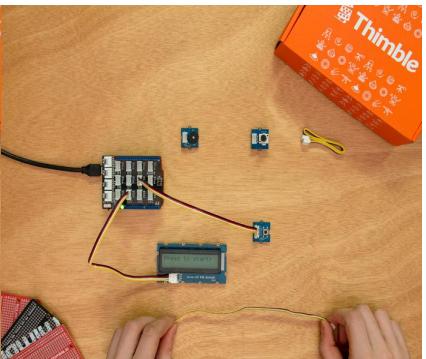
Changing the time value in the code will change the countdown.

## Adding Some Sound ☺

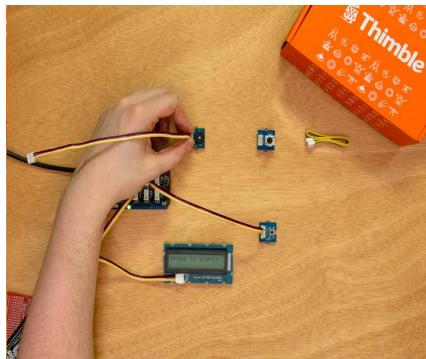
Take a cable and unwrap it. Plug one side into the buzzer socket and the other into **Digital** socket D8.



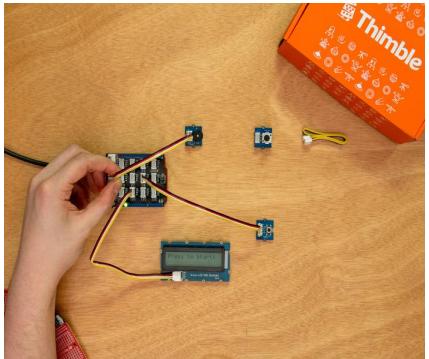
Take a cable...



... and unwrap it



Plug one side into the buzzer socket



... and the other into Digital socket  
D8

## Upload

Upload the code shown below. This tutorial uses **Digital** socket 8. If you are using a different socket update the code after copying it.

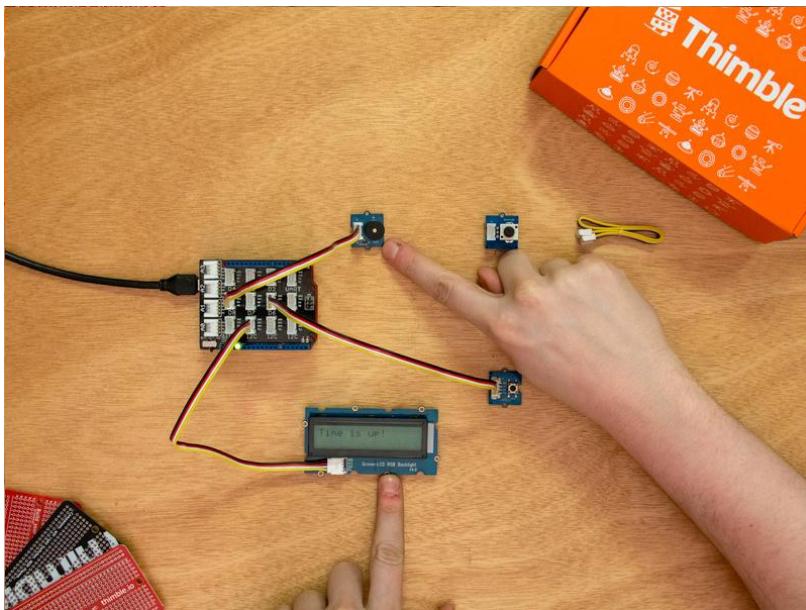
Copy to clipboard

```
1 #include <Wire.h>
2 #include "rgb_lcd.h"
3
4 //If you use different sockets change them below
5 #define buttonSocket 6
6 #define buzzerSocket 8
7
8 int Time;
9
10 rgb_lcd lcd;
11 const int colorR = 0;
12 const int colorG = 0;
13 const int colorB = 0;
14
15 void setup()
16 {
17     pinMode(buttonSocket, INPUT);
18     pinMode(buzzerSocket, OUTPUT);
19     lcd.begin(16, 2);
20     lcd.setRGB(colorR, colorG, colorB);
21
22 }
23
24 void loop()
25 {
26     lcd.setRGB(colorR, colorG, colorB);
27     lcd.setCursor(0, 0);
28     lcd.print("Press to start:");
29     lcd.setCursor(0, 1);
30     lcd.print(" ");
31     Time = 10;
32     digitalWrite(buzzerSocket, LOW);
33     if (digitalRead(buttonSocket)) {
34         while (Time > 0) {
35             lcd.setRGB(colorR, colorG, colorB);
36             lcd.setCursor(0, 0);
37             lcd.print("Time remaining:");
38             lcd.setCursor(0, 1);
39             lcd.print("      ");
40             lcd.setCursor(0, 1);
41             lcd.print(Time);
42             delay(1000);
43             Time = Time - 1;
44         }
45         while (!digitalRead(buttonSocket)) {
46             lcd.setRGB(colorR, colorG, colorB);
47             lcd.setCursor(0, 0);
48             lcd.print("Time is up!      ");
49             lcd.setCursor(0, 1);
50             lcd.print("      ");
51             digitalWrite(buzzerSocket, HIGH);
```

```
52     }
53     delay(100);
54 }
55 }
```

## Observe

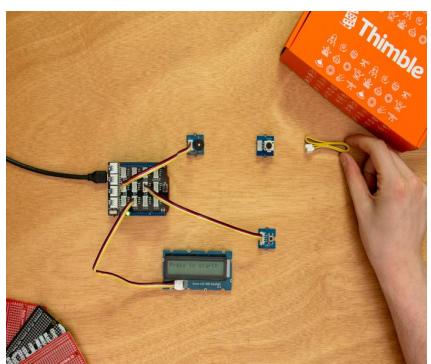
When the timer has reached the end of its countdown, the buzzer turns on. A button press will reset it.



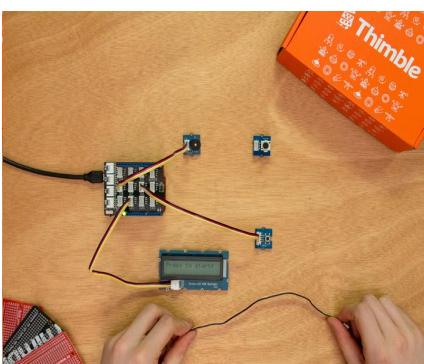
**Alarming!**

## Adding some control ☰

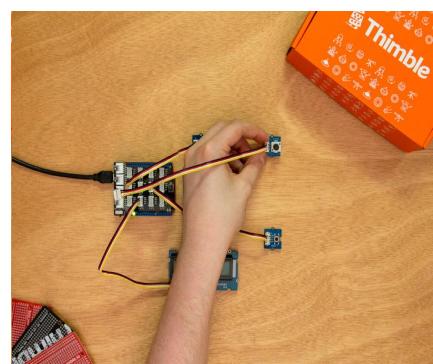
Now we'll add a rotary potentiometer so we can set the timer amount without needing a computer. Take a cable and unwrap it. Plug one side into the rotary potentiometer and the other into **Analog** socket A3.



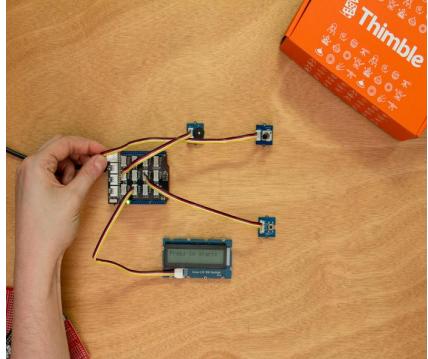
Take a cable...



... and unwrap it



Plug one side into the rotary potentiometer socket



... and the other into Analog socket  
A3

## Upload

Upload the code below. This tutorial uses **Analog** socket A3. If you are using a different socket update the code after copying it.

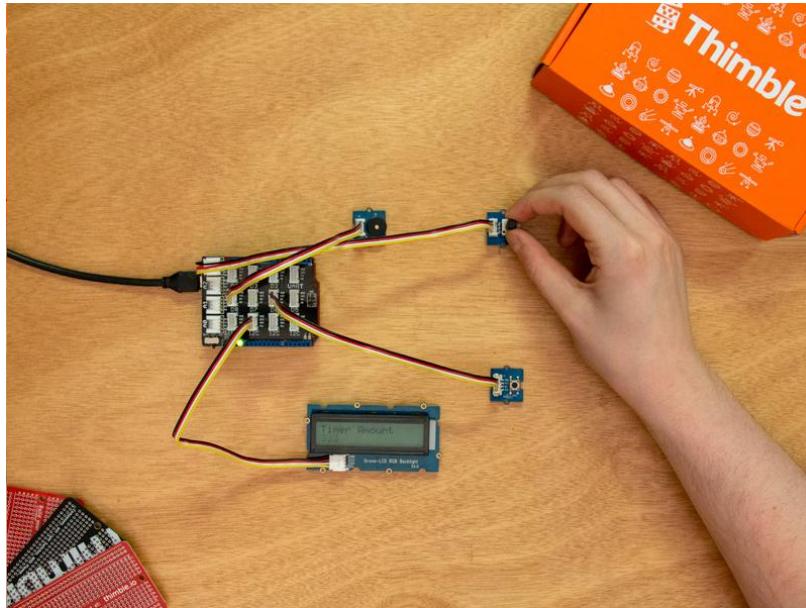
Copy to clipboard

```
1 #include <Wire.h>
2 #include "rgb_lcd.h"
3
4 //Change here if you're using a different socket
5 #define buttonSocket 6
6 #define buzzerSocket 8
7 #define dialSocket A3
8
9 int Time;
10
11 rgb_lcd lcd;
12 const int colorR = 0;
13 const int colorG = 0;
14 const int colorB = 0;
15
16 void setup()
17 {
18     lcd.begin(16, 2);
19     lcd.setRGB(colorR, colorG, colorB);
20
21     pinMode(buttonSocket, INPUT);
22     pinMode(buzzerSocket, OUTPUT);
23 }
24
25 void loop()
26 {
27     digitalWrite(buzzerSocket, LOW);
28     lcd.setRGB(colorR, colorG, colorB);
29     lcd.setCursor(0, 0);
30     lcd.print("Timer Amount:    ");
31     lcd.setCursor(0, 1);
32     lcd.print("    ");
33     lcd.setCursor(0, 1);
34     lcd.print((map(analogRead(dialSocket), 0, 1024, 0, 3600)));
35     delay(100);
36     if (digitalRead(buttonSocket)) {
37         Time = (map(analogRead(dialSocket), 0, 1024, 0, 3600));
38         while (Time > 0) {
39             lcd.setRGB(colorR, colorG, colorB);
40             lcd.setCursor(0, 0);
41             lcd.print("Time remaining:");
42             lcd.setCursor(0, 1);
43             lcd.print("    ");
44             lcd.setCursor(0, 1);
45             lcd.print(Time);
46             delay(1000);
47             Time = Time - 1;
48         }
49         while (!digitalRead(buttonSocket)) {
50             lcd.setRGB(colorR, colorG, colorB);
51             lcd.setCursor(0, 0);
```

```
52     lcd.print("Time is up!");
53     lcd.setCursor(0, 1);
54     lcd.print("      ");
55     digitalWrite(buzzerSocket, HIGH);
56   }
57 }
58 }
59 }
```

## Observe

Now the timer amount can be set to any amount between 0 and 3600 seconds (one hour).



**Changeable!**

## Modify

Changing the `maxTime` variable will change what time range can be chosen.

### Challenge

Modify the code to print the time in hours, minutes, and seconds.

## LCD Backlight Disco

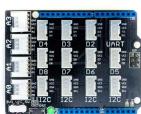
### Start

This project only uses the backlight from the LCD Display to create a disco party.

### Modules

Gather the following parts to complete this project.

## Parts



All Parts x Qty



LCD Display x 1



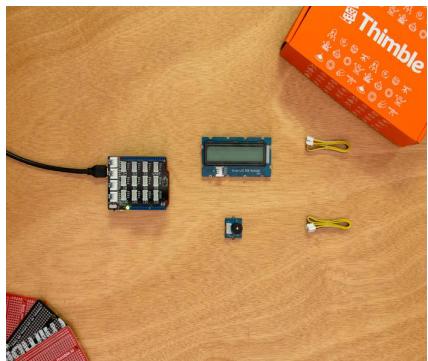
Piezo Buzzer x 1



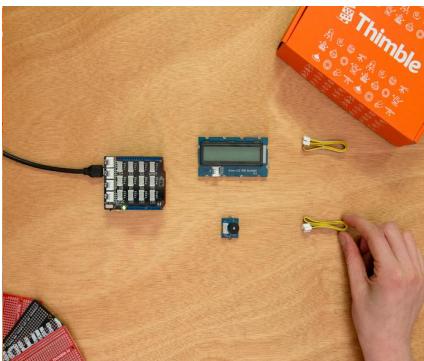
Cable x 2

## Disco isn't

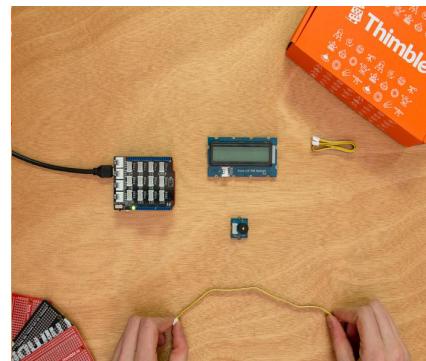
Take a cable and unwrap it. Plug one side into the LCD Display and the other into any **I2C** socket.



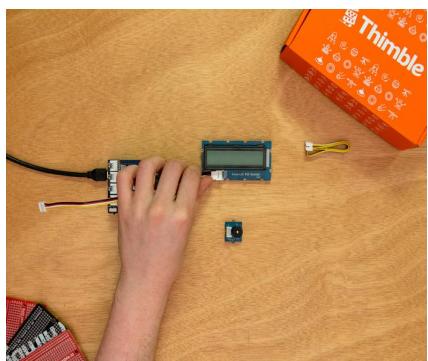
All the parts you'll need



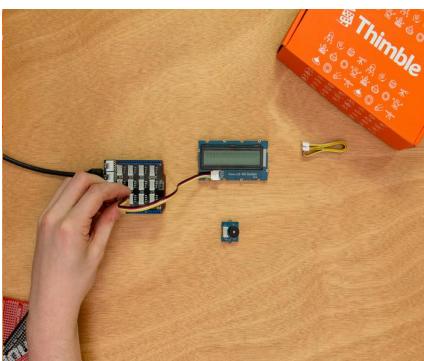
Take a cable...



... and unwrap it



Plug one side into the LCD Display socket



... and the other into any I2C socket

## Upload

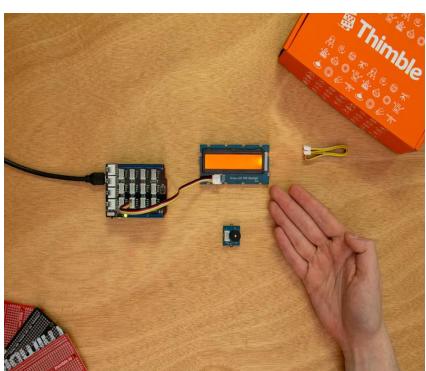
Upload the following code and let's start grooving.

Copy to clipboard

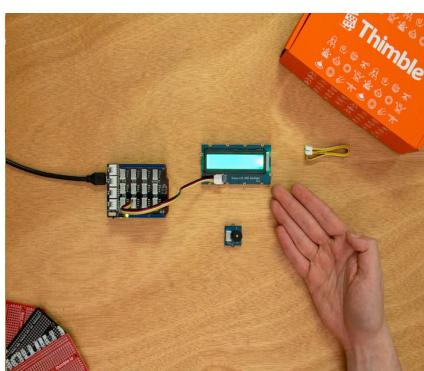
```
1 #include <Wire.h>
2 #include "rgb_lcd.h"
3
4 rgb_lcd lcd;
5 const int colorR = 0;
6 const int colorG = 0;
7 const int colorB = 0;
8
9 int delayTime = 100;
10
11 void setup() {
12     // put your setup code here, to run once:
13     lcd.begin(16, 2);
14     lcd.setRGB(colorR, colorG, colorB);
15     clearScreen();
16 }
17
18 void loop() {
19     // put your main code here, to run repeatedly:
20     lcd.setRGB(255, 0, 0);
21     delay(delayTime);
22     lcd.setRGB(0, 255, 0);
23     delay(delayTime);
24     lcd.setRGB(0, 0, 255);
25     delay(delayTime);
26 }
27
28 void clearScreen() {
29     lcd.setCursor(0, 0);
30     lcd.print("                ");
31     lcd.setCursor(0, 1);
32     lcd.print("                ");
33 }
```

## Observe

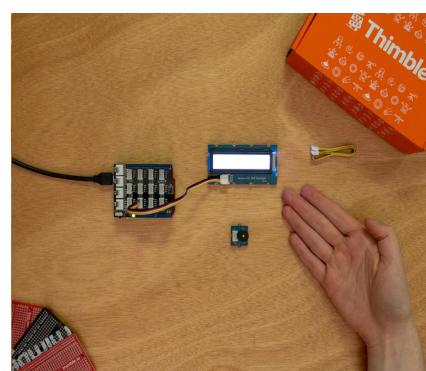
There isn't text being displayed on the display. This is normal for this code. You can see we aren't writing anything to it. Instead, we are just setting the backlight of the LCD Display.



Red



Green



Blue

## Modify

The backlight is following a pattern. With some different code we can make the pattern more random.

Upload the code below to give that a try!

 Copy to clipboard

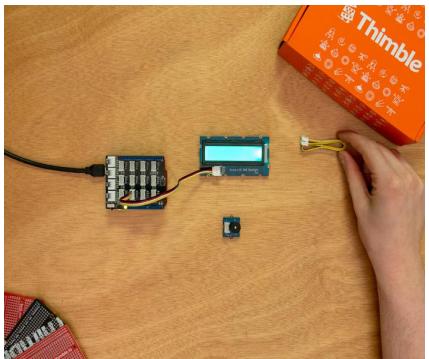
```
1 #include <Wire.h>
2 #include "rgb_lcd.h"
3
4 rgb_lcd lcd;
5 const int colorR = 0;
6 const int colorG = 0;
7 const int colorB = 0;
8
9 int delayTime = 100;
10
11 void setup() {
12     // put your setup code here, to run once:
13     lcd.begin(16, 2);
14     lcd.setRGB(colorR, colorG, colorB);
15     clearScreen();
16 }
17
18 void loop() {
19     // put your main code here, to run repeatedly:
20     lcd.setRGB(random(0, 255) , random(0, 255) , random(0, 255));
21     delay(delayTime);
22     lcd.setRGB(random(0, 255) , random(0, 255) , random(0, 255));
23     delay(delayTime);
24     lcd.setRGB(random(0, 255) , random(0, 255) , random(0, 255));
25     delay(delayTime);
26 }
27
28 void clearScreen() {
29     lcd.setCursor(0, 0);
30     lcd.print("                ");
31     lcd.setCursor(0, 1);
32     lcd.print("                ");
33 }
```

## Experiment

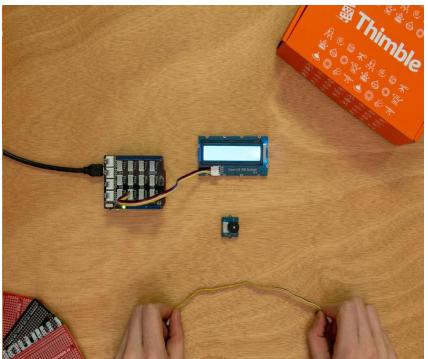
Change around some of the random set variables to see what kind of patterns you can make.

## Adding Some Sound ☺

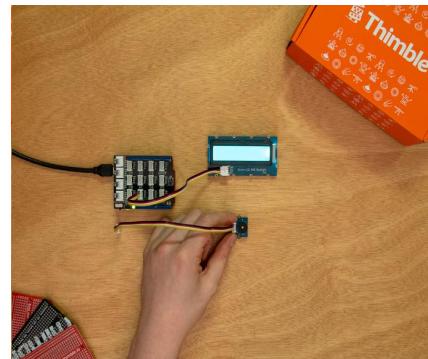
Take a cable and unwrap it. Plug one side into the buzzer socket and the other into **Digital** socket D6.



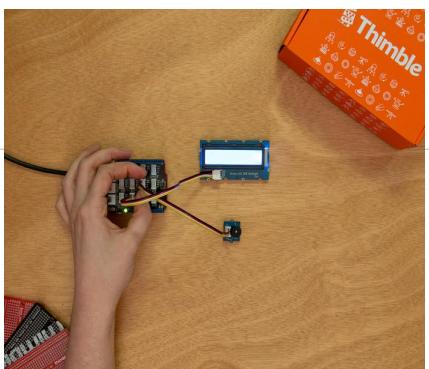
Take a cable...



... and unwrap it



Plug one side into the buzzer socket



The other into Digital socket D6

## Upload

Upload the following code. The example below uses the 6 digital socket. You can use any of them just remember to update the sketch.

[Copy to clipboard](#)

```
1 #include <Wire.h>
2 #include "rgb_lcd.h"
3
4 #define NOTE_C3 131
5 #define NOTE_CS3 139
6 #define NOTE_D3 147
7 #define NOTE_DS3 156
8 #define NOTE_E3 165
9 #define NOTE_F3 175
10 #define NOTE_FS3 185
11 #define NOTE_G3 196
12 #define NOTE_GS3 208
13 #define NOTE_A3 220
14 #define NOTE_AS3 233
15 #define NOTE_B3 247
16
17 rgb_lcd lcd;
18 const int colorR = 0;
19 const int colorG = 0;
20 const int colorB = 0;
21
22 int delayTime = 100;
23
24 //If you use a different socket for the buzzer update the number below
25 #define buzzerSocket 6
26
27 int notes[] = {NOTE_C3, NOTE_CS3, NOTE_D3, NOTE_DS3,
28                 NOTE_E3, NOTE_F3, NOTE_FS3, NOTE_G3,
29                 NOTE_GS3, NOTE_A3, NOTE_AS3, NOTE_B3
30             };
31
32 void setup() {
33     // put your setup code here, to run once:
34     lcd.begin(16, 2);
35     lcd.setRGB(colorR, colorG, colorB);
36     clearScreen();
37 }
38
39 //tone() = tone(pin number, note or frequency, duration of note/frequency);
40
41 void loop() {
42     // put your main code here, to run repeatedly:
43     lcd.setRGB(random(0, 255) , random(0, 255), random(0, 255));
44     tone(buzzerSocket, notes[random(0, 12)], delayTime / 2);
45     delay(delayTime);
46     lcd.setRGB(random(0, 255) , random(0, 255), random(0, 255));
47     tone(buzzerSocket, notes[random(6, 12)], delayTime / 2);
48     delay(delayTime);
49     lcd.setRGB(random(0, 255) , random(0, 255), random(0, 255));
50     tone(buzzerSocket, notes[random(0, 6)], delayTime / 2);
51     delay(delayTime);
```

```
52 }
53
54 void clearScreen() {
55     lcd.setCursor(0, 0);
56     lcd.print("                ");
57     lcd.setCursor(0, 1);
58     lcd.print("                ");
59 }
```

## Observe

Now sounds play when the color switches.

### ⚡ Challenge

Use the melody code from the 8 Bit Music Conductor to play a certain melody.

## Analog Indicator

### Start ⏱

This project is a dial indicator. It takes a signal coming from a sensor and displays using the servo. It's an example of how a device that is normally an **Actuator** can be used as an **Indicator**.

### Modules ⚙️

Gather the following parts to complete this project.

#### Parts



All Parts x Qty



Servo x 1



Light x 1



Temperature x 1



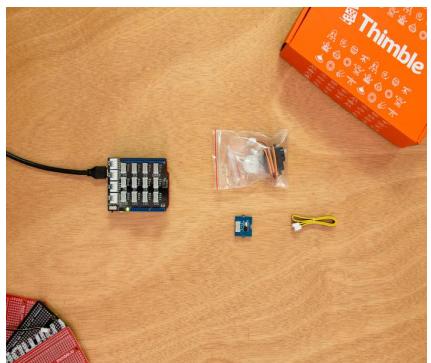
Sound x 1



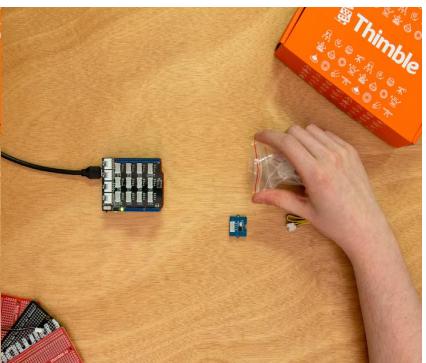
Cable x 3

## Trigger Values ⚙️

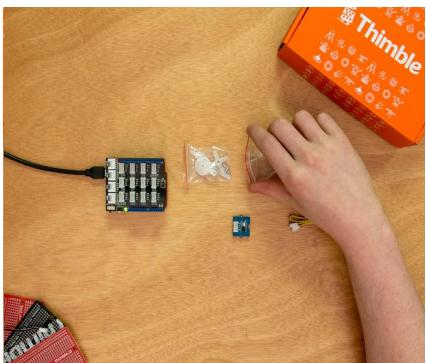
The servo comes in a bag that also has a smaller bag of components in it. You'll need both for this project. Take the servo out as well as the small servo arm and a screw for it. One of the longer screws is better. Take the servo arm and servo and connect them. The arm should be pointing up with none of it hanging off the servo body. After unwrapping the servo cable, plug it in to **Digital** socket D6. For this example we'll be using the light sensor. Take a cable and attach one end to the light sensor and the other to **Analog** socket A0.



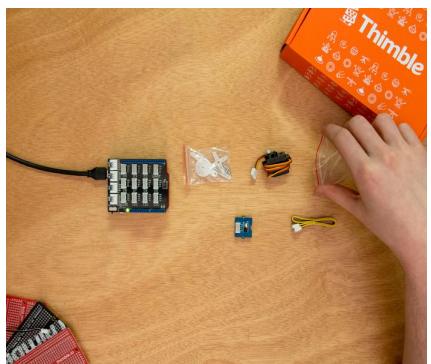
All the parts you'll need



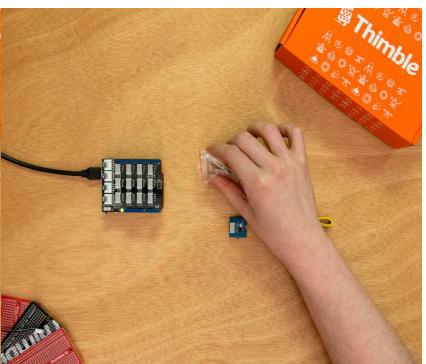
Open the plastic bag



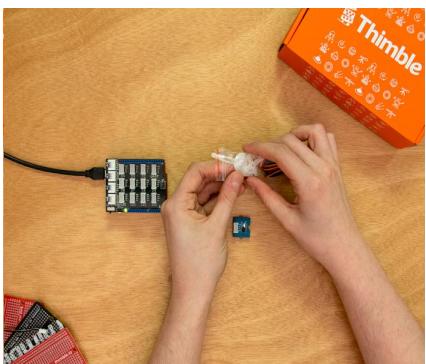
Take out the bag of parts inside...



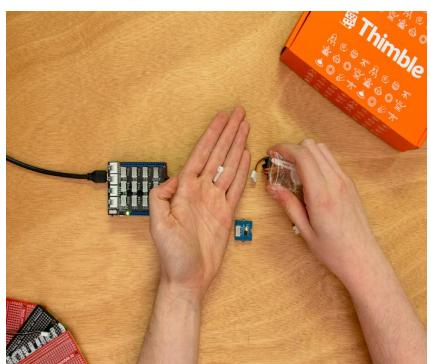
... and the servo



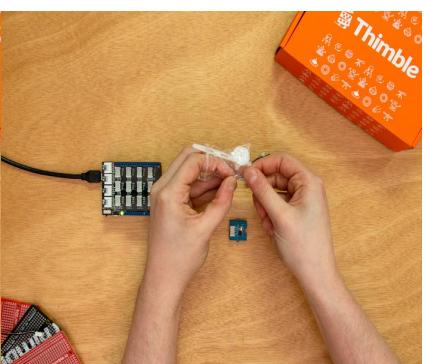
Open the smaller bag...



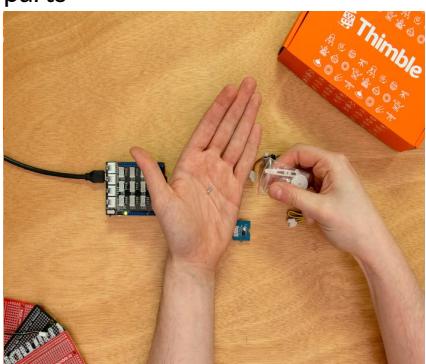
Be careful! There are many little parts



Take out the small 'Servo Arm'...



... and ...



... one longer screw

