

Project 11

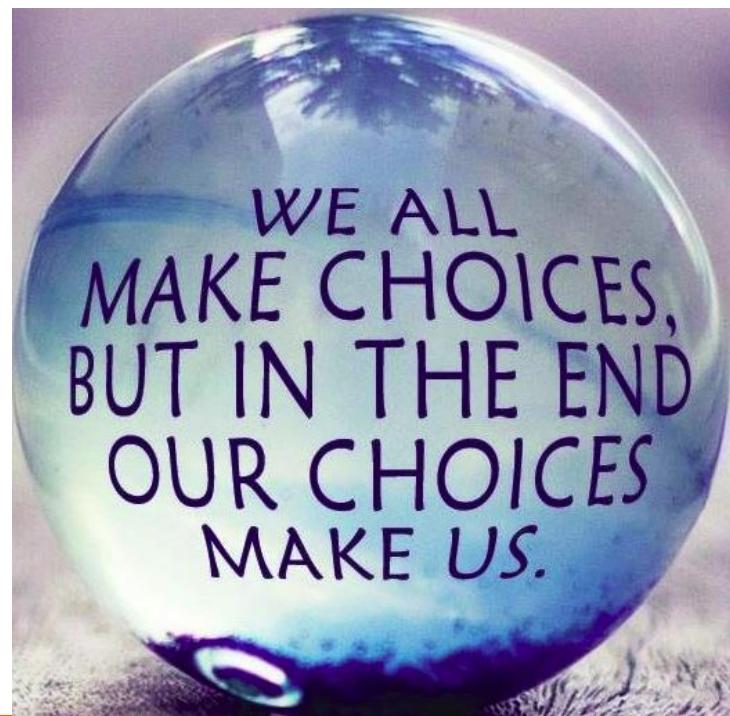
Crystal Ball

Discover: LCD displays, switch/case statements,
random()



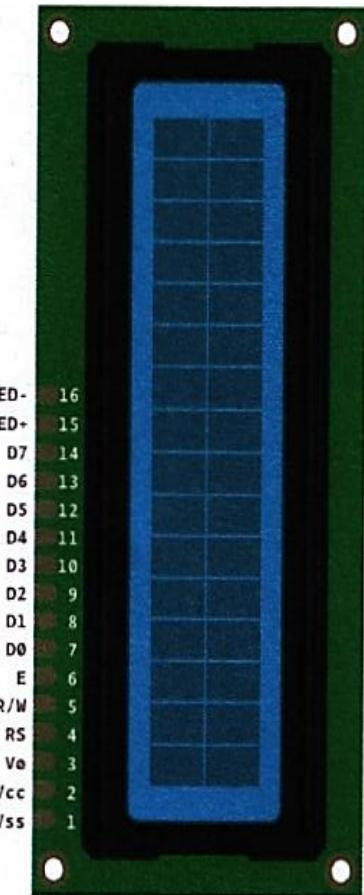
Introduction

- Crystal balls can help "predict" the future.
 - You ask a question to the all-knowing ball, and turn it over to reveal an answer.
 - The answers will be predetermined, but you can write anything you like.
- You'll use your Arduino to choose from a total of 8 responses.
 - The tilt switch help replicates the motion of shaking the ball.



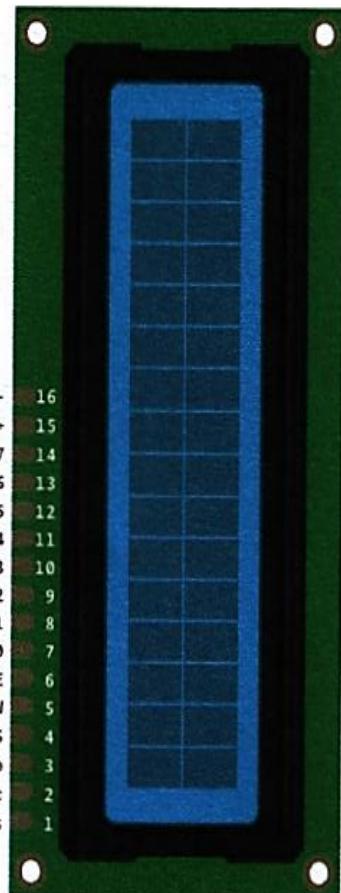
LCD (Liquid-crystal display)

- The LCD can be used to display alphanumeric characters.
 - The one in your kit has 16 columns and 2 rows, for a total of 32 characters.

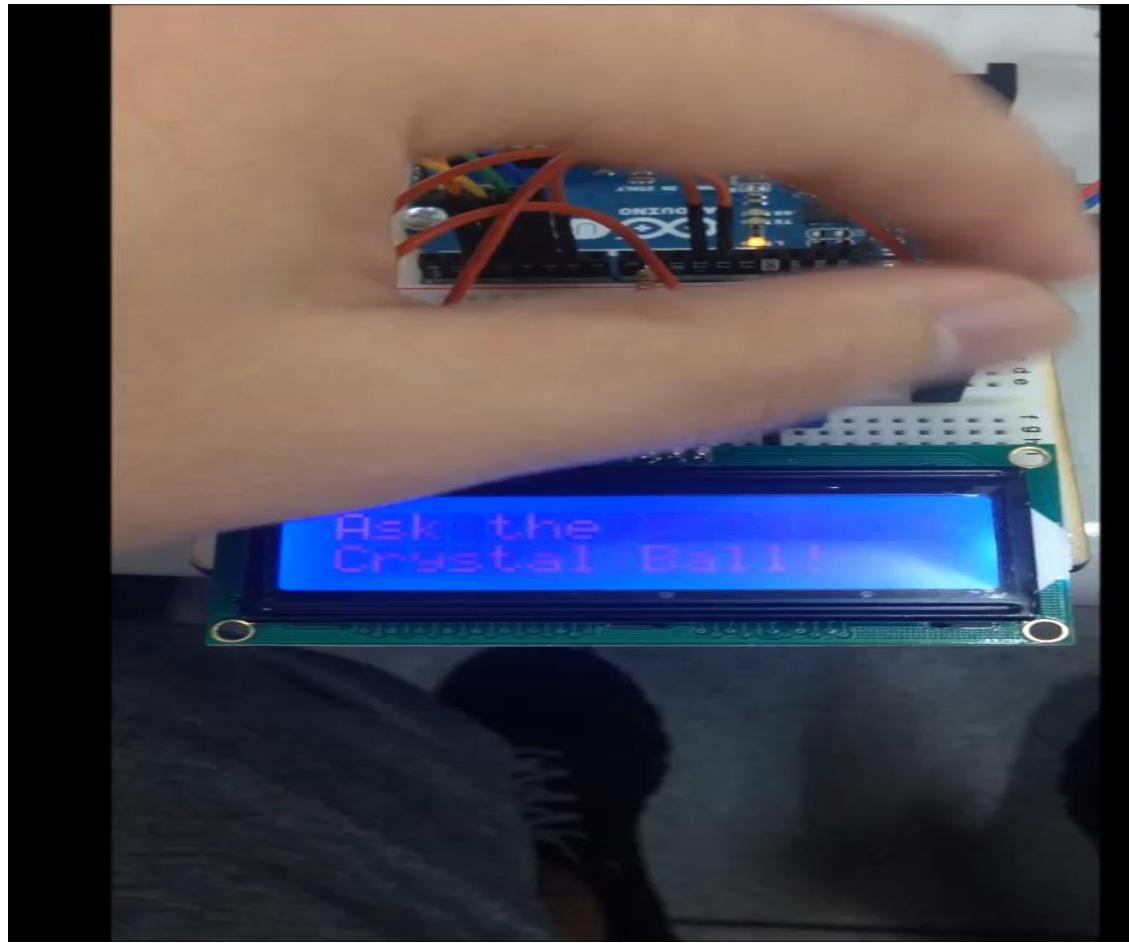


LCD (cont.)

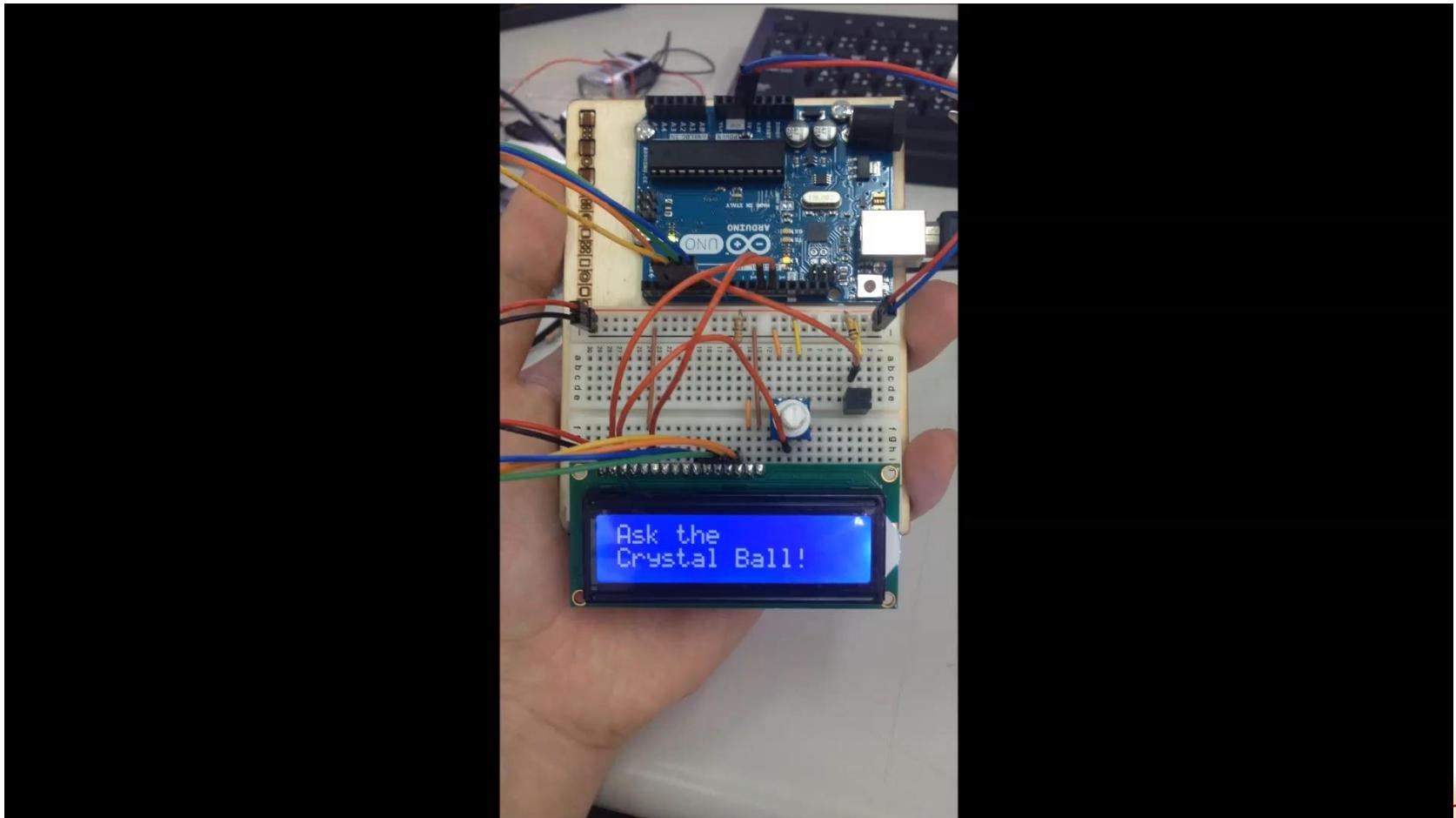
- Pin definition:
 - RS(Register Select):controls where the characters will appear
 - R/W(read/write): puts the screen in read or write mode
 - You'll be using the write mode in this project (connect the R/W pin to ground)
 - EN (enable): tells the LCD that it will be receiving a command.
 - 00-07 (data pins): sends character data to the screen
 - You'll only be using 4 of these (04-07).
 - Vss & LED-: connect to ground.
 - Vcc: connects directly to 5V.
 - LED+: connects to power through a 220 ohm resistor.



Demo (adjust the contrast)

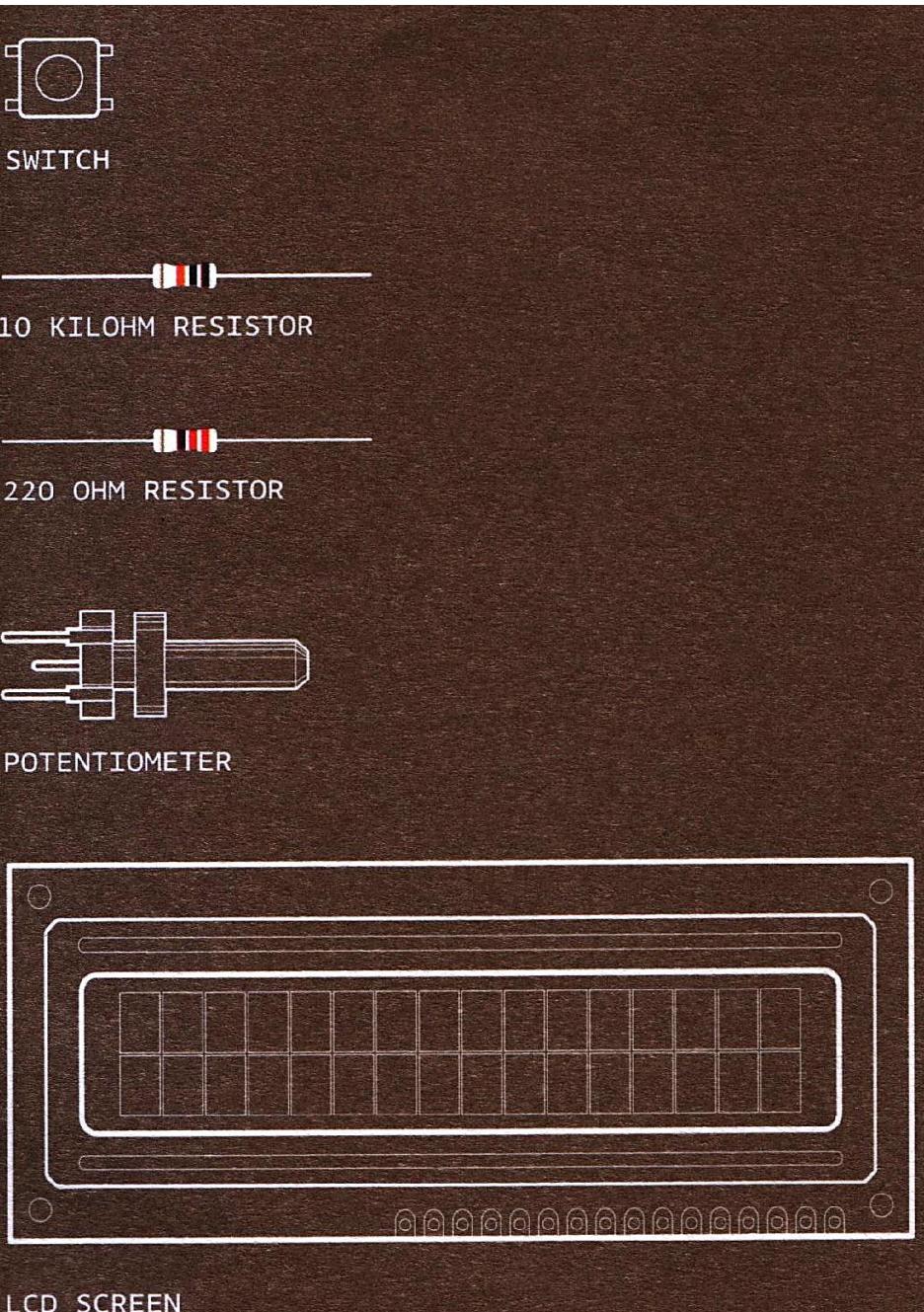


Demo (ask the crystal ball)

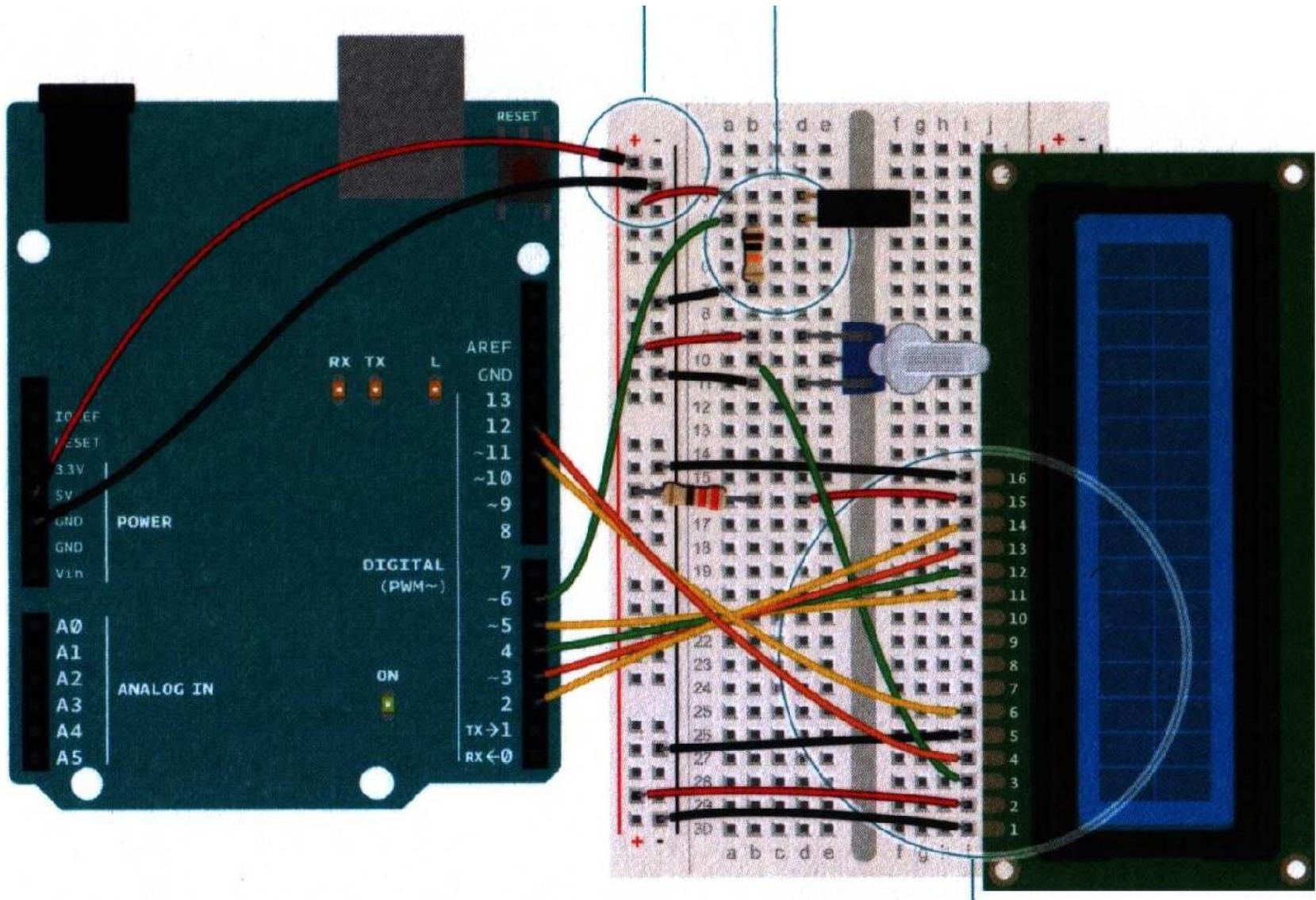


Ingredients

- Tilt sensor
- 1個 $10K\Omega$ 電阻(RESISTOR)
- 1個 220Ω 電阻(RESISTOR)
- 電位器(POTENTIOMETER)
- 液晶螢幕(LCD SCREEN)

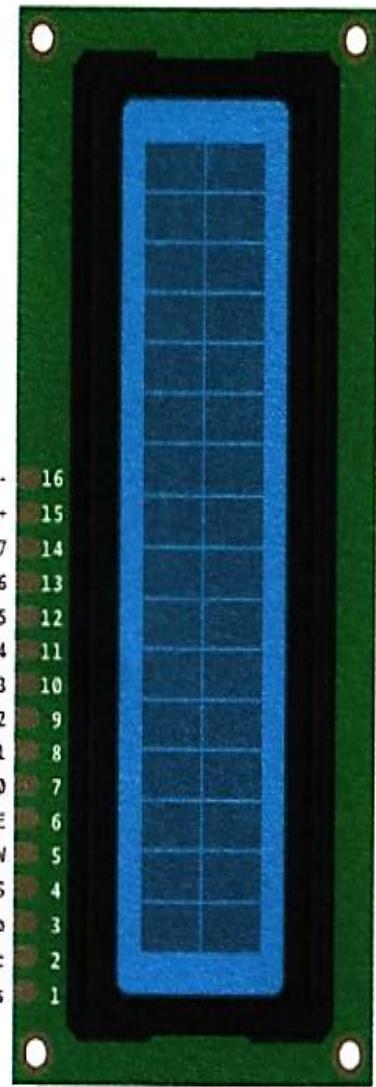
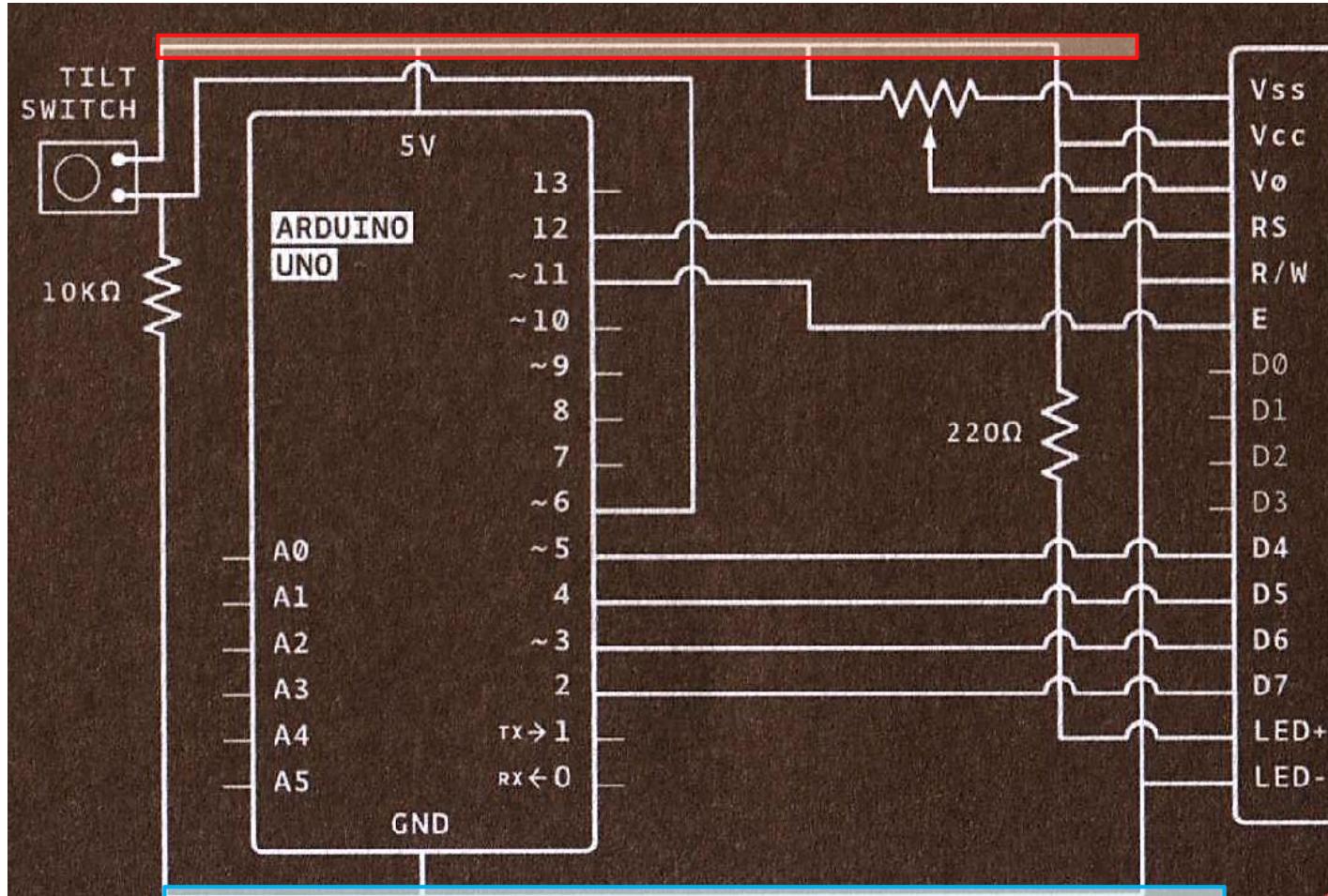


The Top View of the Circuit



Schematic Diagram

LCD pins not
follow the
physical order



後面接線圖！

The Code

```
1 #include <LiquidCrystal.h>
2 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
3 //lcd (RS, E, D4, D5, D6, D7)
4 const int switchPin = 6; Tilt switch
5 int switchState = 0;
6 int prevSwitchState = 0;
```

The Code (cont.)

```
7 void setup() {  
8     lcd.begin(16, 2); 16*2 characters  
9     pinMode(switchPin, INPUT);  
  
10    lcd.print("Ask the");  
  
11    lcd.setCursor(0, 1);  
12    lcd.print("Crystal Ball!");  
13 }
```

The Code (cont.)

```
14 void loop() {  
15     switchState = digitalRead(switchPin);  
  
16     if (switchState != prevSwitchState) {  
17         if (switchState == LOW) {  
18             reply = random(8);  
  
19             lcd.clear();  
20             lcd.setCursor(0, 0);  
21             lcd.print("The ball says:");  
22             lcd.setCursor(0, 1);
```



```
23     switch(reply){  
24         case 0:  
25             lcd.print("Yes");  
26             break;  
27         case 1:  
28             lcd.print("Most likely");  
29             break;  
30         case 2:  
31             lcd.print("Certainly");  
32             break;  
33         case 3:
```



The Code (cont.)

```
34     lcd.print("Outlook good");  
35     break;  
36     case 4:  
37     lcd.print("Unsure");  
38     break;  
39     case 5:  
40     lcd.print("Ask again");  
41     break;
```

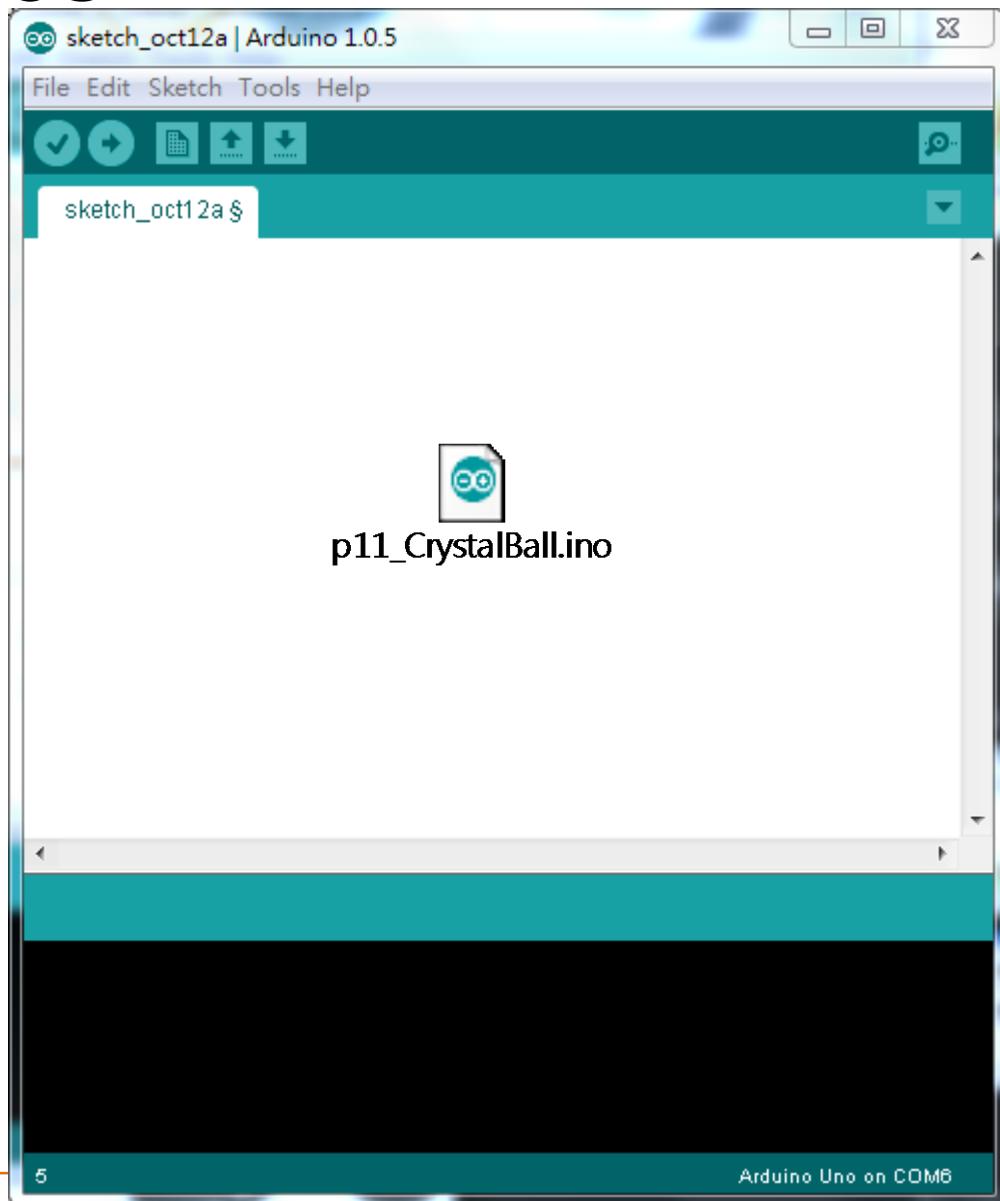


The Code (cont.)

```
42     case 6:  
43         lcd.print("Doubtful");  
44     break;  
45     case 7:  
46         lcd.print("No");  
47     break;  
48 }  
49 }  
50 }  
51 prevSwitchState = switchState;  
52 }
```



Codes



15/30

Project 12

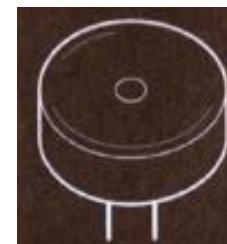
Knock Lock

Discover: input with a piezo, writing your own
functions



Introduction

- The piezo you used for playing back sounds in the projects can also be used as a sensor.
 - When plugged into 5V, the sensor can detect vibrations that can be read by the Arduino's analog inputs.
- You'll need to plug in a high value resistor (like 1-megohm) as the reference to ground for this to work well.
 - Since it is a high-resistance device



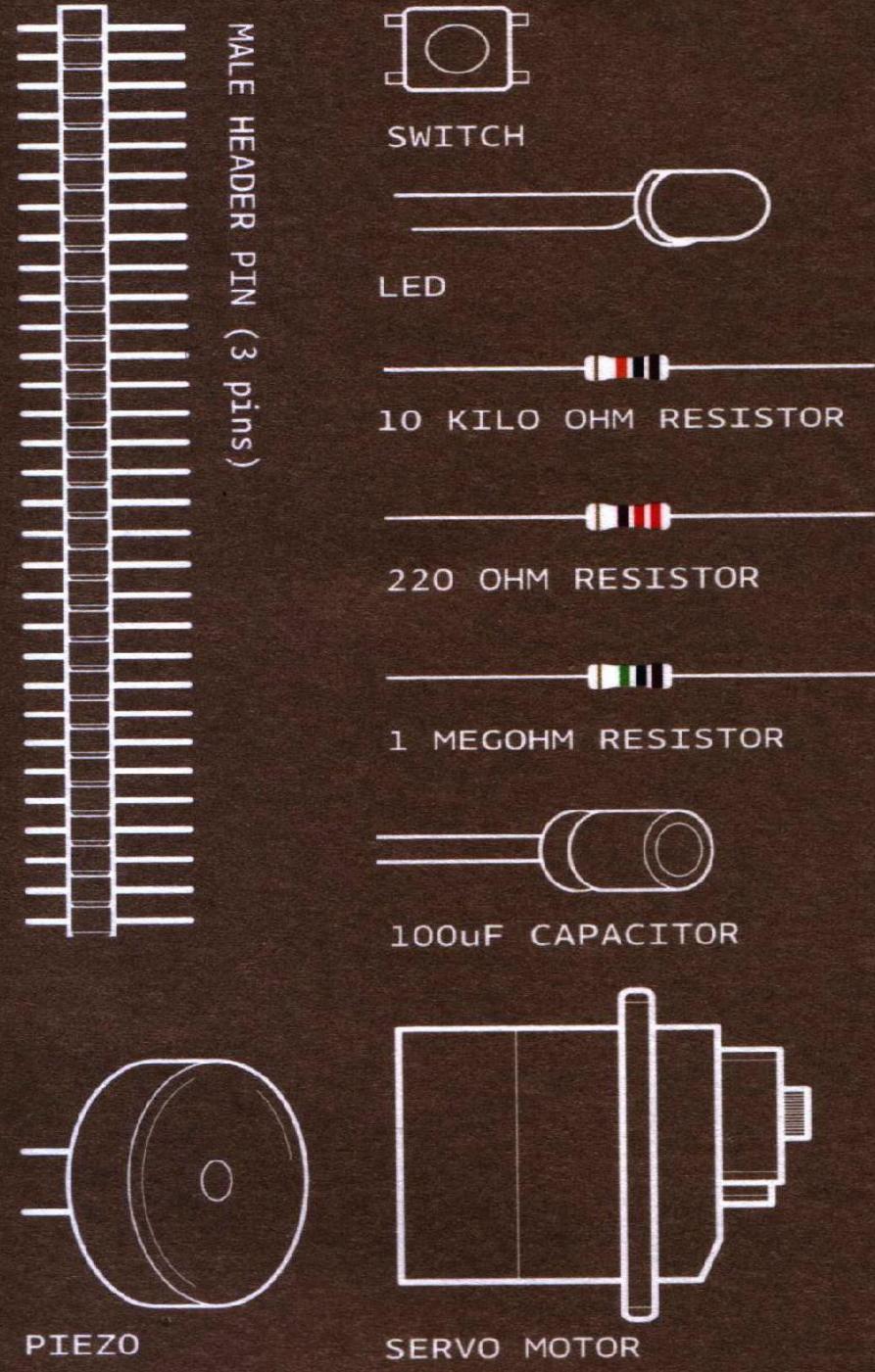
Demo

You'll probably hear the piezo make a small "click" when it first gets power.



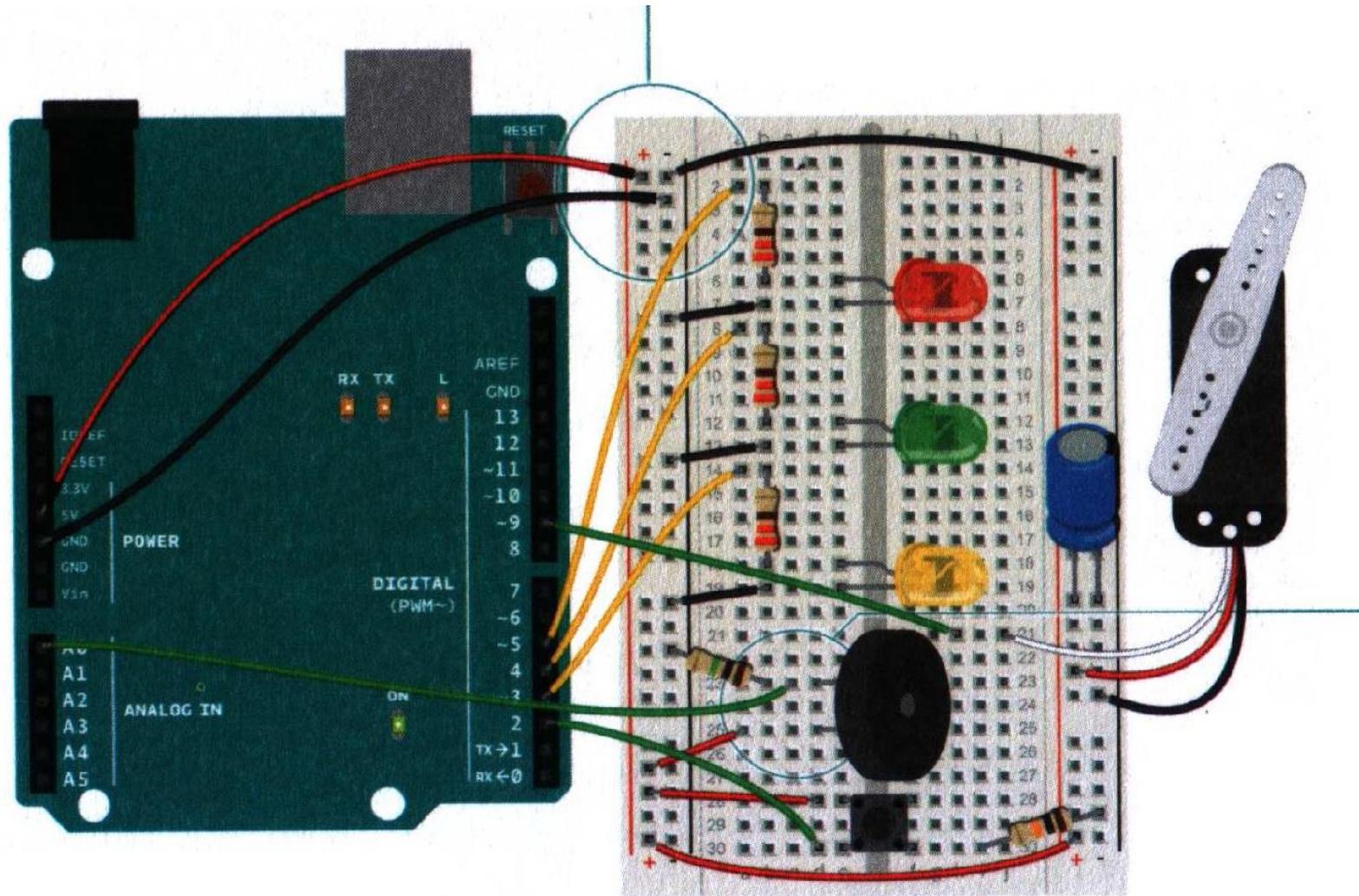
Ingredients

- 1個開關(SWITCH)
- LED燈(紅黃綠各1個)
- 1個 $10K\Omega$ 電阻(RESISTOR)
- 3個 220Ω 電阻(RESISTOR)
- 1個 $1M\Omega$ 電阻(RESISTOR)
- 1個 $100\mu F$ 電容(CAPACITOR)
- 伺服馬達(SERVO MOTOR)
- 蜂鳴器(PIEZO)

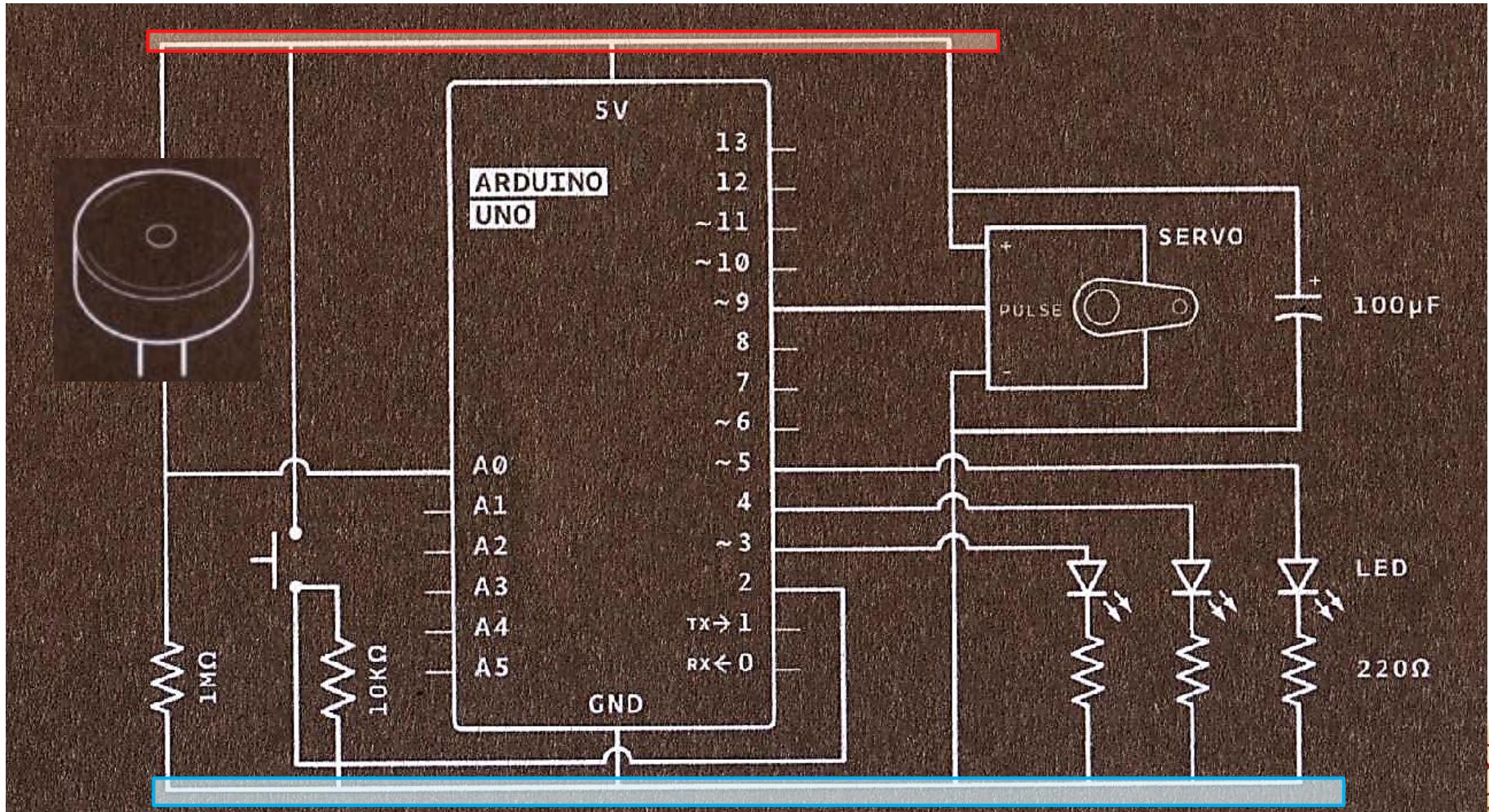


The Top View of the Circuit

If your piezo has a red wire or one marked with a "+", that is the one to connect to power.



Schematic Diagram



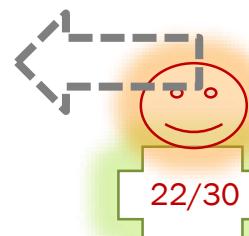
The Codes

```
1 #include <Servo.h>
2
3 Servo myServo;
4
5 const int piezo = A0;
6
7 const int switchPin = 2;
8
9 const int yellowLed = 3;
10
11 const int greenLed = 4;
12
13 const int redLed = 5;
14
15 int knockVal; //Set up some constants to use as thresholds for the knock
16
17 int switchVal; //maximum and minimum levels.
18
19 const int quietKnock = 10;
20
21 const int loudKnock = 100;
```



The Codes (cont.)

```
14 void setup(){
15     myServo.attach(9);
16     pinMode(yellowLed, OUTPUT);
17     pinMode(redLed, OUTPUT);
18     pinMode(greenLed, OUTPUT);
19     pinMode(switchPin, INPUT);
20     Serial.begin(9600);
```



The Codes (cont.)

```
21  digitalWrite(greenLed, HIGH);  
22  myServo.write(0); //degree  
23  Serial.println("The box is unlocked!");  
24 }  
  
25 void loop(){  
26  if(locked == false){  
27    switchVal = digitalRead(switchPin);
```



The Codes (cont.)

```
28 if(switchVal == HIGH){  
29     locked = true;  
30     digitalWrite(greenLed,LOW);  
31     digitalWrite(redLed,HIGH);  
32     myServo.write(90);  
33     Serial.println("The box is locked!");  
34     delay (1000);  
35 }  
36 }
```

The Codes (cont.)

```
37 if(locked == true){  
38     knockVal = analogRead(piezo);  
  
39     if(numberOfKnocks < 3 && knockVal > 0){  
40         if(checkForKnock(knockVal) == true){  
41             numberOfKnocks++;  
42         }  
43         Serial.print(3-numberOfKnocks);  
44         Serial.println(" more knocks to go");  
45     }
```



The Codes (cont.)

```
46 if(numberOfKnocks >= 3){  
47     locked = false;  
48     myServo.write(0);  
49     delay(20);  
50     digitalWrite(greenLed,HIGH);  
51     digitalWrite(redLed,LOW);  
52     Serial.println("The box is unlocked!");  
53 }  
54 }  
55 }
```

The Codes (cont.)

```
56 boolean checkForKnock(int value){  
  
57     if(value > quietKnock && value < loudKnock){  
  
58         digitalWrite(yellowLed, HIGH);  
59         delay(50);  
60         digitalWrite(yellowLed, LOW);  
61         Serial.print("Valid knock of value ");  
62         Serial.println(value);
```

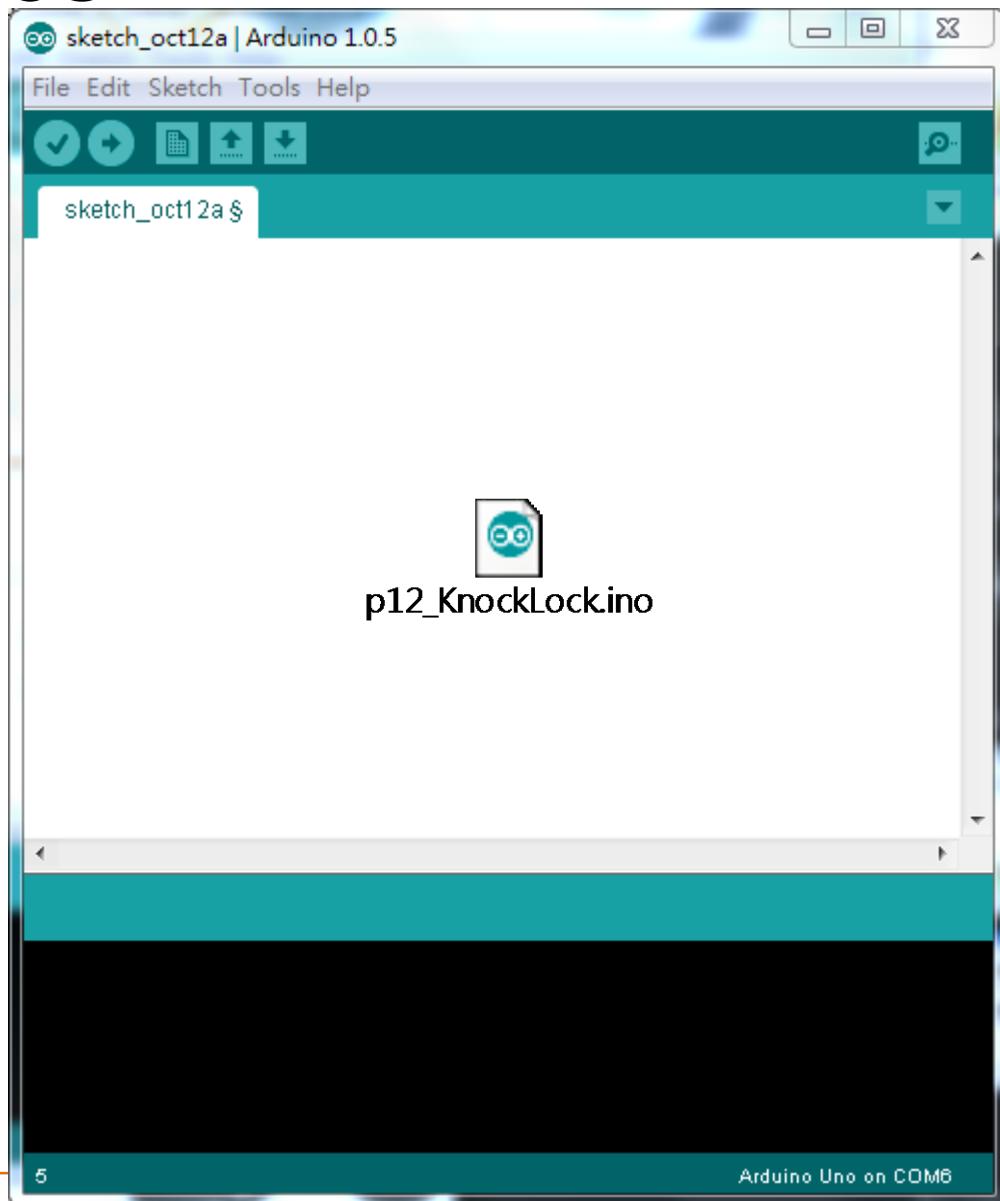


The Codes (cont.)

```
63     return true;  
64 }  
  
65 else {  
66     Serial.print("Bad knock value ");  
67     Serial.println(value);  
68     return false;  
69 }  
70 }
```

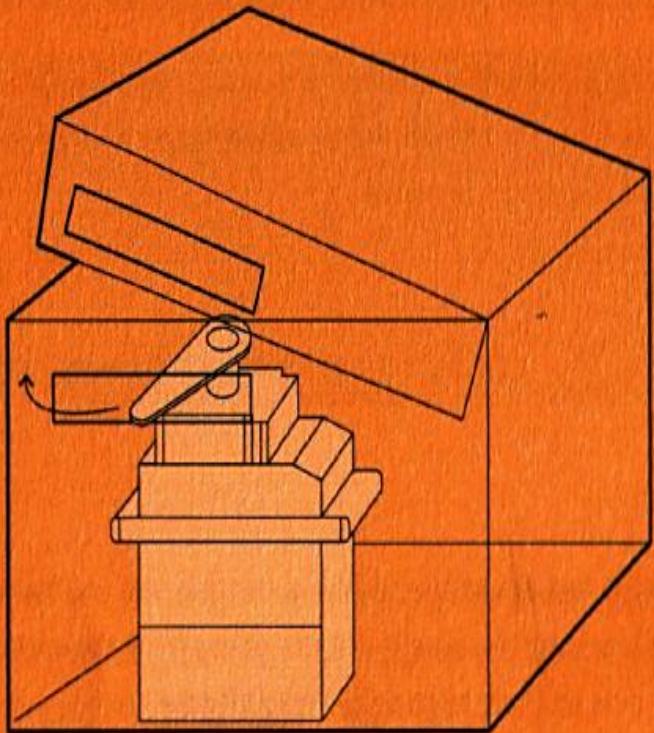


Codes



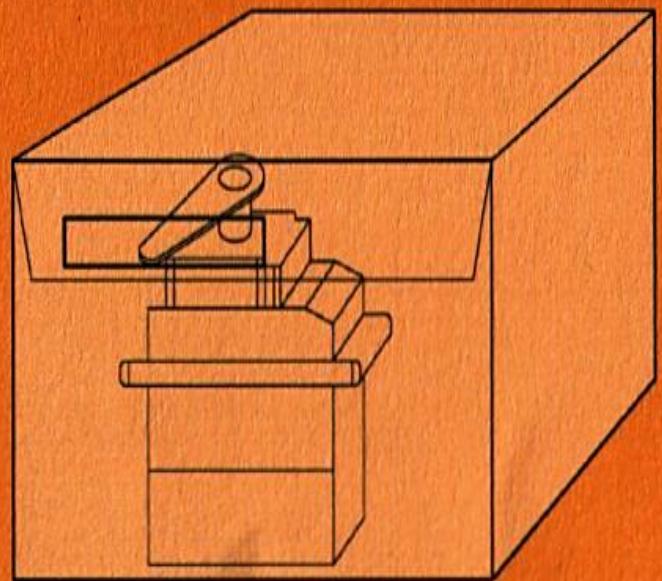
29/30

Finish Up



1

Cut 2 holes in your box: one on the side, and a second through the cover flap. Place the servo in the box so that the arm can move in and out of the holes when closed.



2

Secure the servo in place with some tape, again making sure the arm can easily rotate through the slot you made.