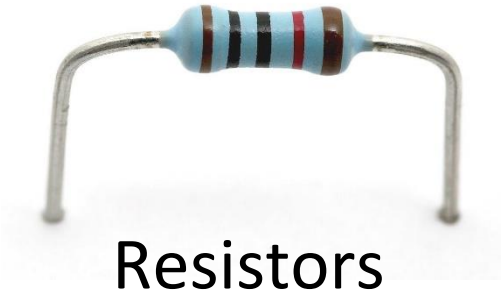
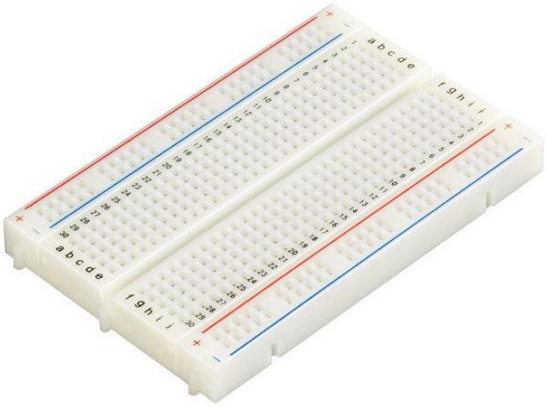
Components



Resistors

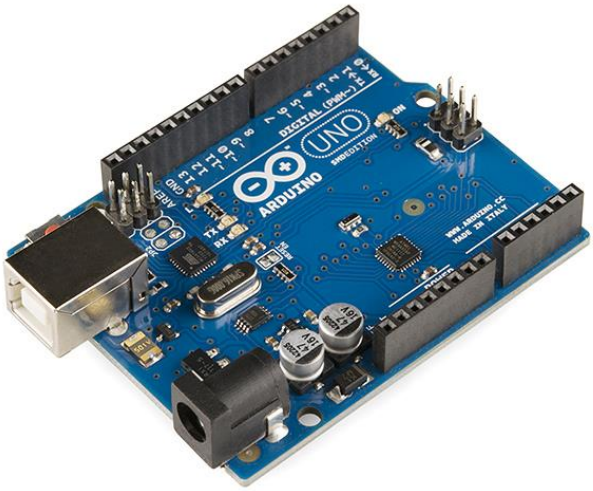
LED

Buzzer



Breadboard

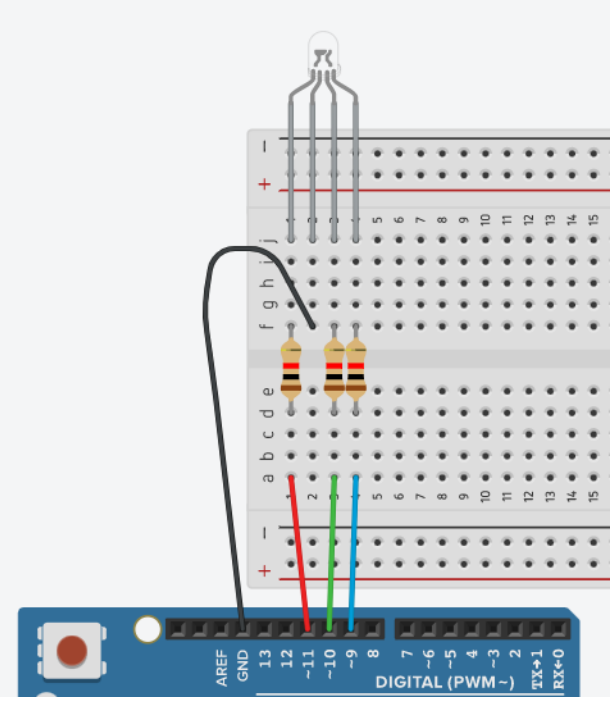
Ultrasonic range sensor



Arduino

Exercise 1: Light an RGB

Clear your board of all the other circuits, and now build the circuit below with an RGB (Red Green Blue) LED. This is a single component with three LEDs inside.



Upload the code below to the Arduino. The LED should light up red.

What happens if you change the line to RGB_color(0,255,0)?

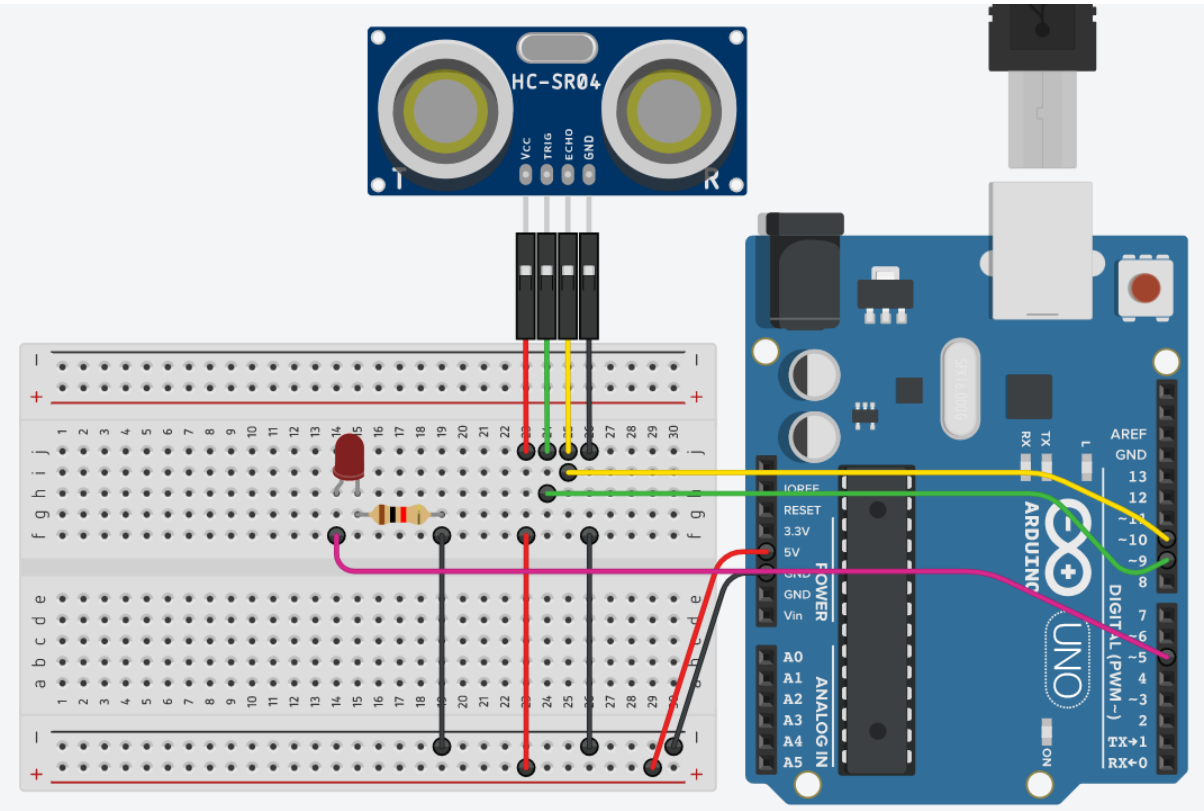
Try different combinations of numbers and see which colours you can make.

```
int red_pin= 11;
int green_pin = 10;
int blue_pin = 9;
void setup() {
  pinMode(red_pin, OUTPUT);
  pinMode(green_pin, OUTPUT);
  pinMode(blue_pin, OUTPUT);
  RGB_color(255, 0, 0);
}
void loop() {
}
void RGB_color(int red, int green, int blue) {
  analogWrite(red_pin, red);
  analogWrite(green_pin, green);
  analogWrite(blue_pin, blue);
}
```

Exercise 2: Ultrasonic sensor

Build the circuit below to use the ultrasonic sensor.

Use a 220 Ohm resistor (red, red, black, black, brown)



- The sensor sends a pulse of ultrasonic sound (sound at a frequency higher than we can hear).

- This sound bounces off any object in front of the sensor, and then the sensor listens for this echo.

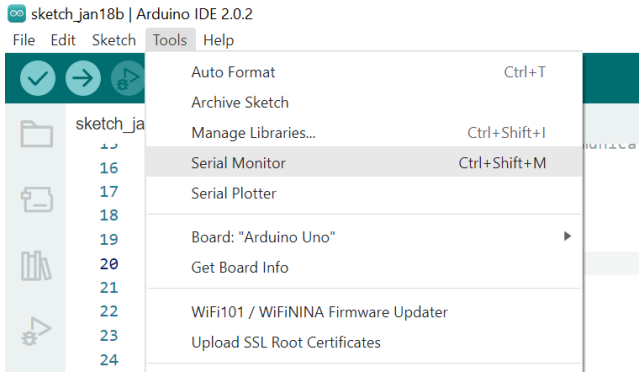
- By measuring the time between sending out the pulse of sound, and hearing the echo, we can calculate the distance from the sensor to the object.

$$\textit{Distance} = \textit{speed} \times \textit{time}$$

Use the code below to measure the distance to the sensor.

```
const int trigPin = 9;
const int echoPin = 10;
const int ledPin = 5;
long duration;
int distance;
int brightness;

void setup() {
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
pinMode(ledPin, OUTPUT);
Serial.begin(9600);
}

void loop() {
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);
distance= (duration*0.034)/2;

Serial.print("Distance (cm): ");
Serial.println(distance);

brightness = map(distance,0,100,0,255);
analogWrite(ledPin,brightness);
}
```

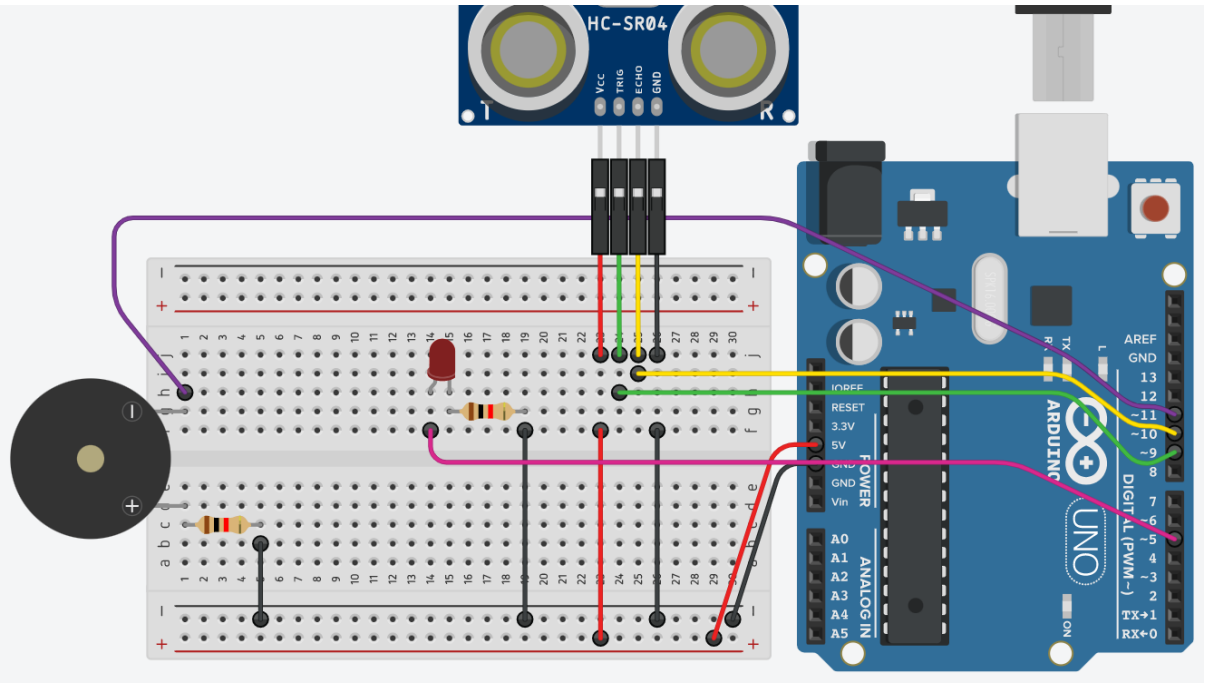
Use the serial monitor to view the distance



Exercise 3: Use a buzzer

Add a buzzer to your circuit like as shown below.

Use a 10 kOhm resistor for the buzzer, (brown, black, black, red, brown)



The “pitch” of a musical note is how high or low it sounds, this depends on the frequency of the sound wave.

We can produce different frequencies of sound using the buzzer.

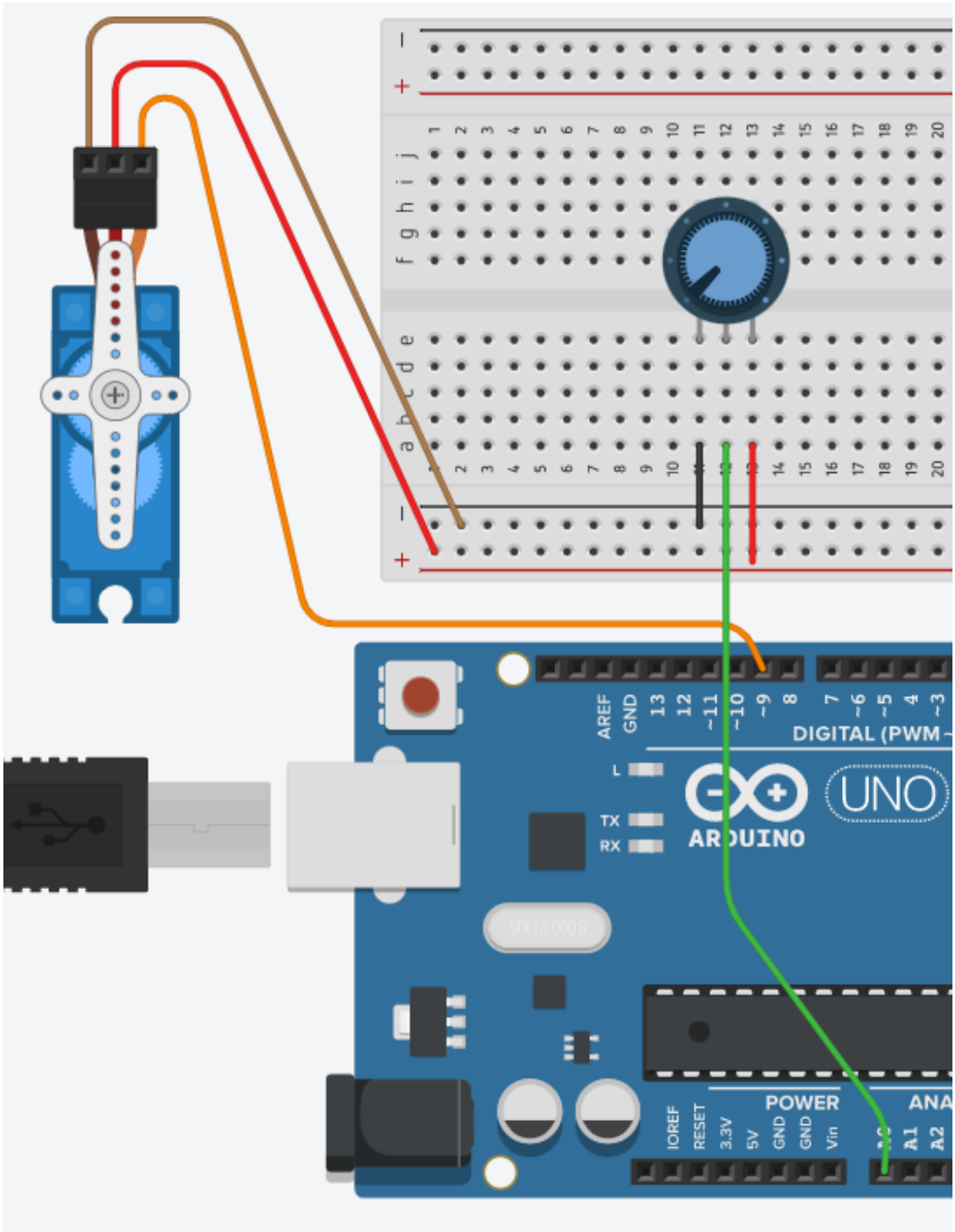
Add the lines of code which are highlighted on the next page. Can you play a tune using the range sensor and buzzer?

Same code as before – just add the highlighted lines

```
const int trigPin = 9;
const int echoPin = 10;
const int buzzer = 11;
const int ledPin = 5;
long duration;
int distance;
int brightness;
int note;

void setup() {
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
pinMode(buzzer, OUTPUT);
pinMode(ledPin, OUTPUT);
Serial.begin(9600);
}

void loop() {
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);
distance= (duration*0.034)/2;

Serial.print("Distance (cm): ");
Serial.println(distance);

brightness = map(distance,5,100,0,255);
analogWrite(ledPin,brightness);

note = map(distance,0,100,120,1200);
tone(buzzer,note);

}
```

Exercise 4: Use a Servo

Build the circuit below to use a servo motor with a potentiometer.

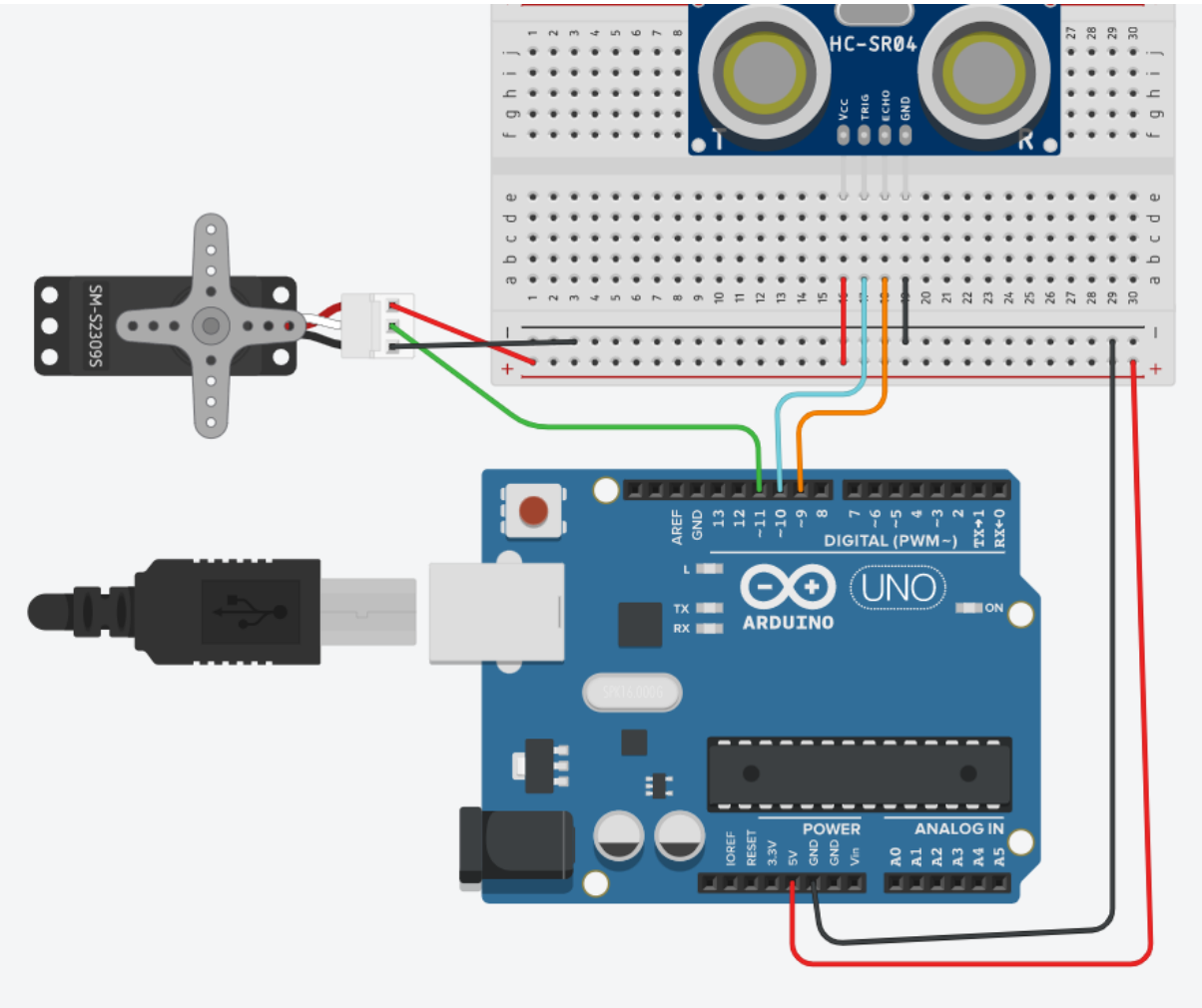


Upload the code below. What happens when you turn the potentiometer?

```
#include <Servo.h>

Servo myservo;
int potpin = 0;
int val;

void setup() {
  myservo.attach(9);
}

void loop() {
  val = analogRead(potpin);
  val = map(val, 0, 1023, 0, 180);
  myservo.write(val);
  delay(15);
}
```

Exercise 5: Servo and Ultrasonic Sensor

Build the circuit below to use a servo motor with the ultrasonic sensor.



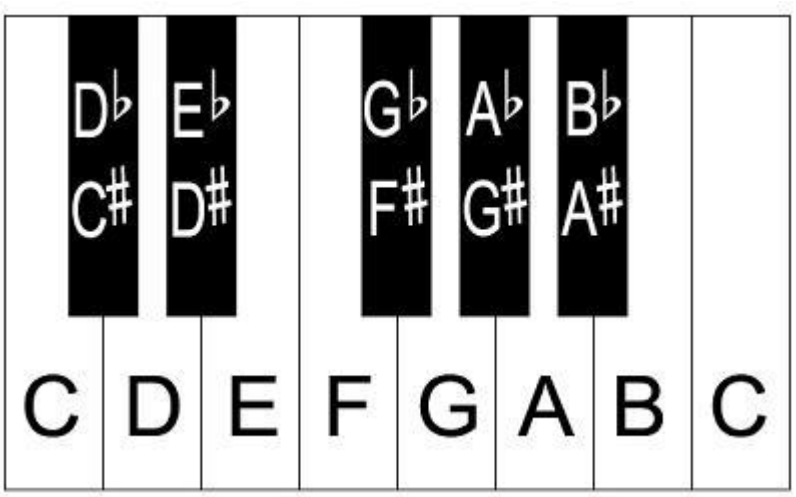
Upload the code on the next page. Can you control the servo to point towards your partner?

```
#include <Servo.h>
Servo myservo;

myservo.attach(11);

const int trigPin = 10;
const int echoPin = 9;
const int ledPin = 5;
long duration;
int distance;
int angle;

void setup() {
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
pinMode(ledPin, OUTPUT);
Serial.begin(9600);
}

void loop() {
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);
distance= (duration*0.034)/2;

Serial.print("Distance (cm): ");
Serial.println(distance);

angle = map(distance,0,100,0,180);
myservo.write(angle);
delay(15);
}
```

Exercise 6: Write some music

Build the circuit from exercise 3 (we'll just be using the buzzer) and upload the code on the next page.

We can now play musical notes (C, C#, D, D#...) on the buzzer using the Arduino.



We can play any note by writing play(note, length);

Where note is the name (C,D,E,etc), and length is how long to play the note for, in milliseconds.

Try modifying the code to play "Twinkle twinkle little star", (C, C, G, G, A, A, G…) or make up your own tune.

```
#define C 262
#define CS 277
#define D 294
#define DS 311
#define E 330
#define F 349
#define FS 370
#define G 392
#define GS 415
#define A 440
#define AS 466
#define B 494
#define C5 524

int buzzer = 11;

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

void loop(){
  play(C,500);
  play(D,500);
  play(E,500);
}

void play(int note,int length){
  tone(buzzer,note);
  delay(length);
  noTone(buzzer);
  delay(100);
}
```