ARDUINO

Robotic Programming

What is Arduino?

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

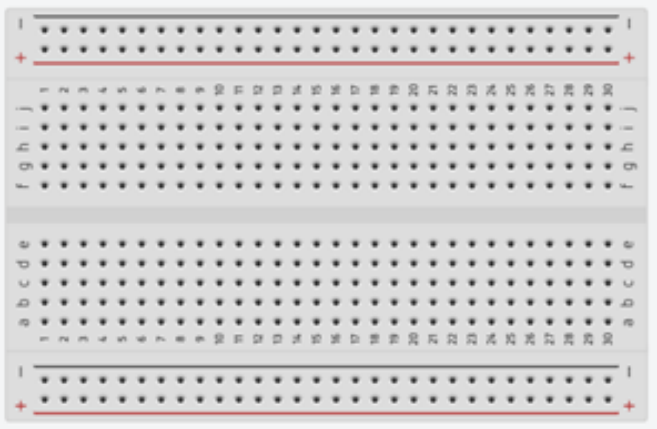
Why Arduino?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community.

Circuit Elements

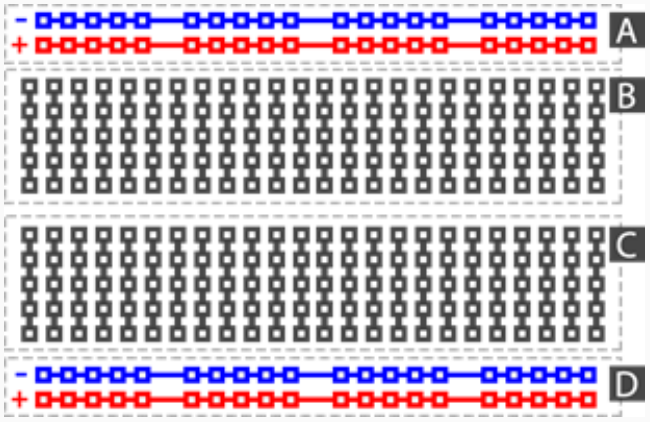
1. BreadBoard

It is the plastic board on which the circuits are built for testing purposes. By attaching the cables and circuit elements to the holes on it, it is ensured that they are combined with each other. In this way, it allows us to set up circuits quickly and practically without soldering or making additions.

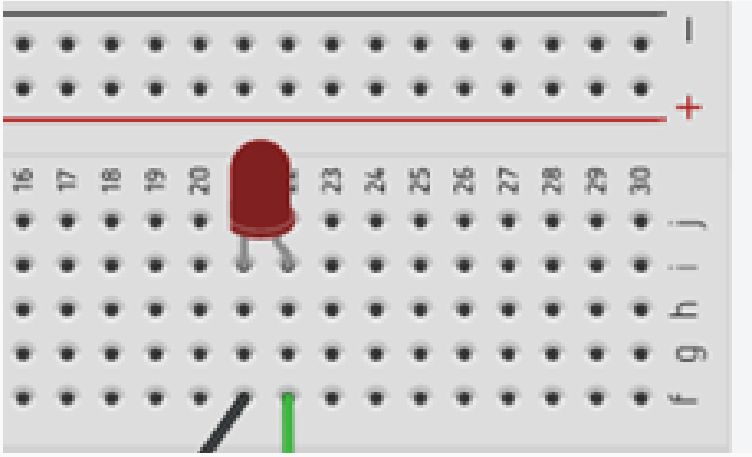


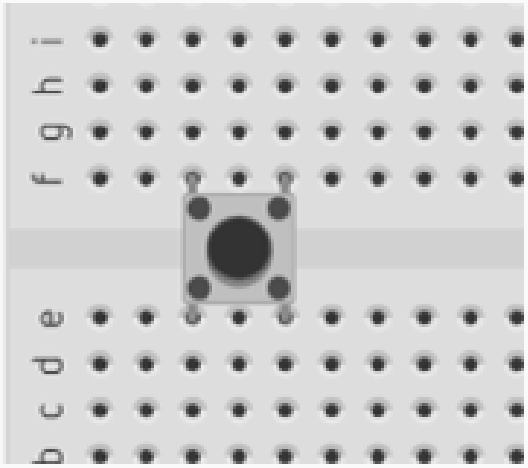
BreadBoard

The internal structure of the breadboard consists of metal clamps positioned vertically and horizontally connected to each other. When viewed from the outside, the red and blue parts are the rows of the breadboard. These parts are in conduction in the form of a line from one end to the other. The areas in the middle, named with letters, are the column parts of the breadboard. The parts of the column are horizontally connected to each other. The point to be noted is that the spaces with spaces in between are in groups within themselves. That is, groups a, b, c, d, e are connected to each other and groups f, g, h, i and j are connected to each other.



Lines indicated as + and – on the breadboard are used to provide voltage connections.





Circuit Elements

2. Jumper Cable

Jumper cables are used to make connections between Arduino, breadboard and circuit elements while designing circuits. The reason why it is produced in different colors is to ensure that cable tracking is easy when installing and controlling the circuit. Generally, black wires are used for GND (ground, -) and red wires are used for power.

Jumper cables are named as male or female according to their ends. There are three types with male and female inputs at the ends.

Jumper Wires

3. Resistor

Resistors of different values

In electrical circuits, resistance is the strain faced by an electric current flowing through a conductor. They serve to keep the current at a certain value by limiting the current in the circuits. It is used to prevent excessive voltage coming to the circuit element. Value information is read through the colors on it.

The resistor circuit element has two legs. The circuit element has no direction. Therefore, it can be connected to the pin or GND port without paying attention to the reverse in the circuit connections.

Circuit Elements

4. LED

5mm LED in different colors

LED is an acronym for Light Emitting Diode. Diodes are two-legged semiconductor circuit elements that allow current to flow in only one direction. It is the circuit element that gives out light when electricity is passed over the LED diode.

The LED circuit element has two legs. One of these legs is short and the other is long. The long leg should be connected to the Arduino pin and the short leg should be connected to the GND.

5. Buzzer

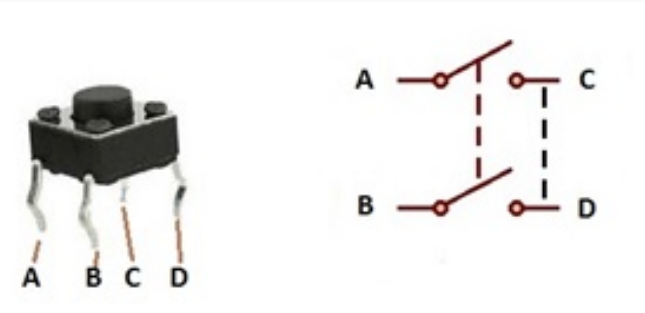
Buzzer is a circuit element used to produce sound in Arduino circuits. We can see the buzzer in many places that give us warning in daily life. For example, parking systems in vehicles, alarms, etc.

Buzzer circuit element has two legs. One of these legs is short and the other is long. The long leg should be connected to the Arduino pin and the short leg should be connected to the GND port.

Buzzer

Circuit Elements

6. Push Buton



Push buttons are also called Push-Pull Button. It works in a way that passes an electric current when pressed, and cuts the current when it is released. There are two types of push buttons. The first is 2 pins (with ends) and the other is 4 pins. They basically have the same working principle. One end is connected to the circuit as input and one end as output.

Push Buttons

7. Ultrasonic Sensor

Ultrasonic sensors are a type of sensor that measures distance using sound waves. The logic of working with radar is the same. It sends ultrasonic waves to the target object and converts the reflection back into an electrical signal. The sent ultrasonic waves travel at a frequency faster than the frequency of sound that humans can hear. This sound frequency is between 20 kHz and 500 kHz. Ultrasonic wave frequency of HC SR04 is 40kHz.

Ultrasonic Sensor - (HC-SR04)

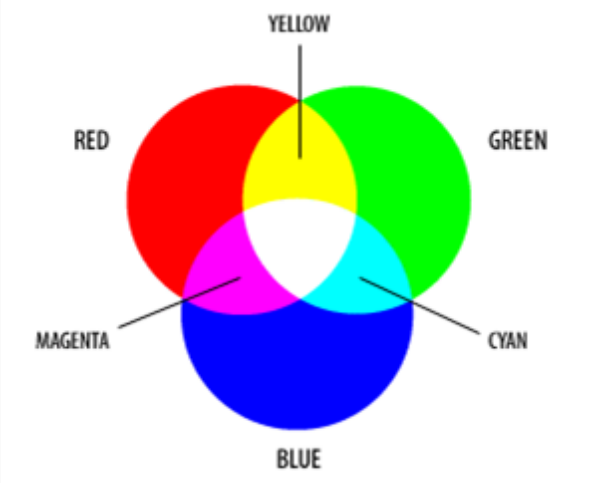
Circuit Elements

8. RGB LED

RGB LED

The RGB LED name consists of the initials of the words Red – Green – Blue. Unlike normal LEDs, it contains 3 different colors (red, green and blue) LEDs in a single package. 16,581,375 colors can be obtained with different values.

It has four legs as R, G, B and GND.



Renk Görünümü	Renk Adı	HEX değeri	RGB değeri
	White - Beyaz	#FFFFFF	255,255,255
	Yellow - Sarı	#FFFF00	255,255,0
	Fuchsia - Fuşya	#FF00FF	255,0,255
	Red - Kırmızı	#FF0000	255,0,0
	Silver - Gümüş	#C0C0C0	192,192,192
	Gray - Gri	#808080	128,128,128
	Olive - Zeytin yeşili	#808000	128,128,0
	Purple - Mor	#800080	128,0,128
	Maroon - Vişne Çürüğü	#800000	128,0,0
	Aqua - Turkuaz	#00FFFF	0,255,255
	Lime - Parlak yeşil	#00FF00	0,255,0
	Teal - Koyu Yeşil	#008080	0,128,128
	Green - Yeşil	#008000	0,128,0
	Blue - Mavi	#0000FF	0,0,255
	Navy - Yeşil	#000080	0,0,128
	Black - Siyah	#000000	0,0,0

Circuit Elements

9. Rotary potentiometer

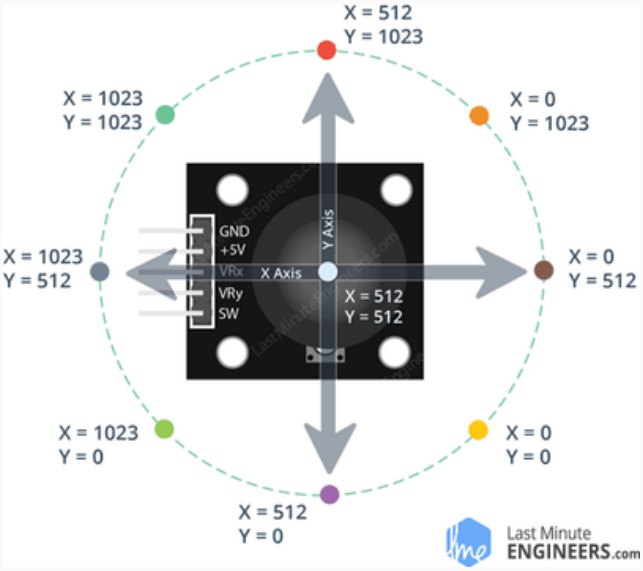
Rotary potentiometer with caps

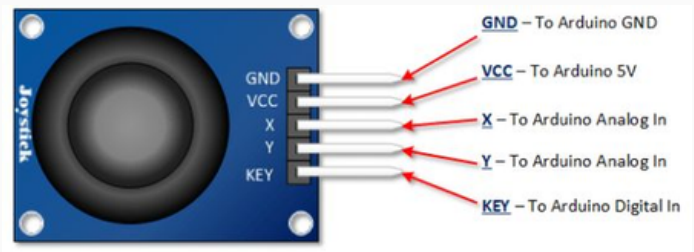
A potentiometer is an analog circuit element that produces values between 0 and 1023. The potentiometer has 3 terminals. The middle part of the potentiometer is the common terminal and is connected to the analog input of the microcontroller board. The left and right terminals of the potentiometer are connected to the 5V and GND terminals.

10. XY-axis Joystick Module

Joystick is a circuit element that produces values between 0 and 1023 according to the movement in the x and y axis. The middle part also acts as a pullup button. When the button is pressed, the value 1 is read.

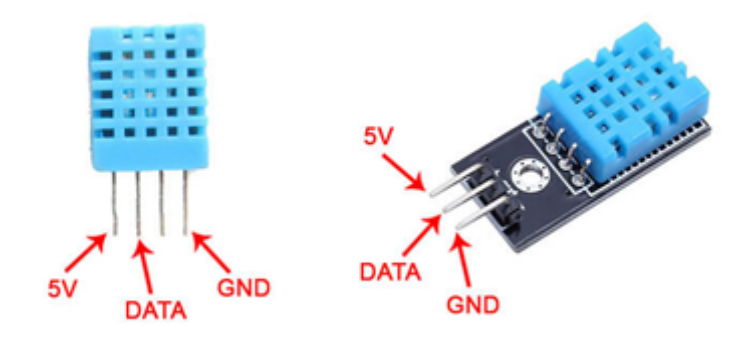
Joystick





Circuit Elements

11. DHT11 Basic Temperature - Humidity Sensor



DHT11 Sensor

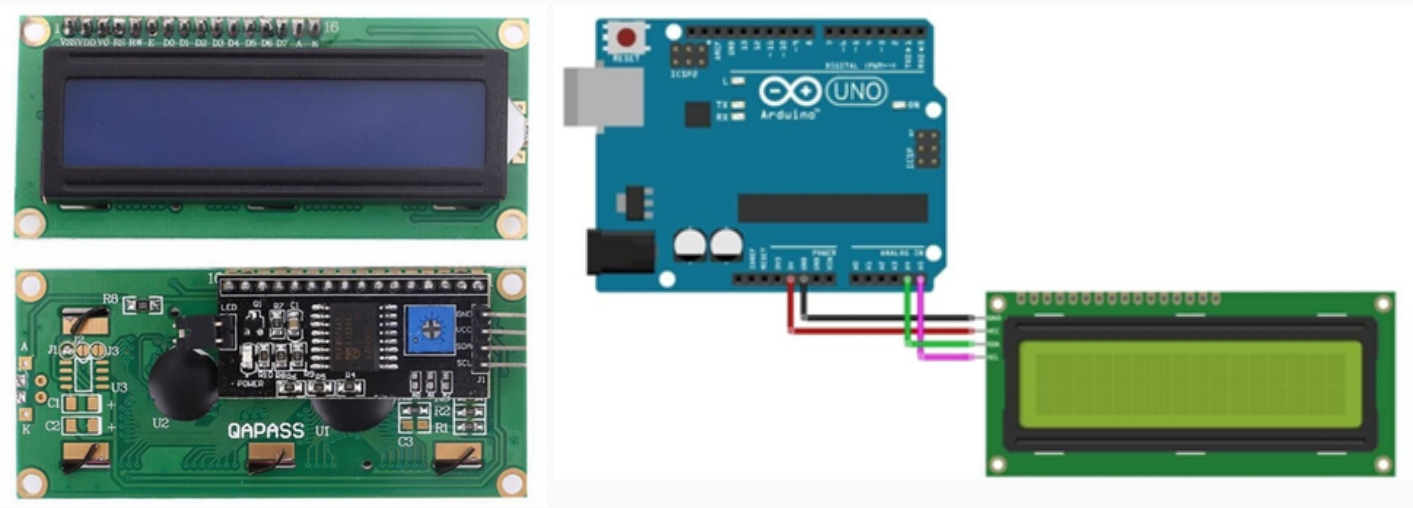
DHT11 is a sensor used to measure temperature and humidity in the environment. It is used to obtain the ambient temperature in various heat units (celcius, farhenayt, kelvin) and to measure the humidity in the environment in %.

12. 2X16 LCD Display

LCD displays are used to show textual information to the user. The number 2 indicates the amount of rows and the number 16 indicates the number of columns. There are different sizes in the market such as 2x8, 2x16, 4x16, 2x20, 2x40 and 4x40 according to the number of rows and columns. It has a white text color on a blue or green background. It is illuminated by a back LED and works with 5V. No extra power supply is required, which can be connected directly to the Arduino.

Connection Pins

GND = Arduino GND
VCC = Arduino 5V
SDA = Arduino A4
SCL = Arduino A5



2X16 LCD Display

Circuit Elements

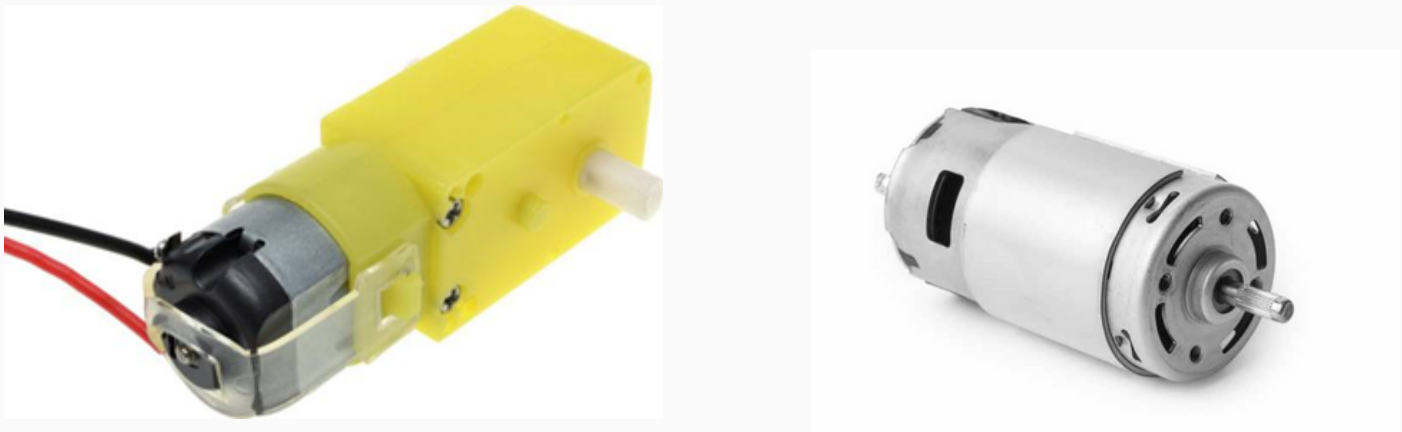
13. Servo Motor

It is a type of motor that can move angularly. Servo motors are produced in the form of 90, 180, 360 degrees and continuous rotation. It has 3 connectors.

Servo Motors

14. DC Motor

DC motors are electrical machines that convert electrical energy into motion energy. It works between 6V and 48V. Since it needs more energy than 5V, it needs a motor herd module in its use.



DC Motor

Circuit Elements

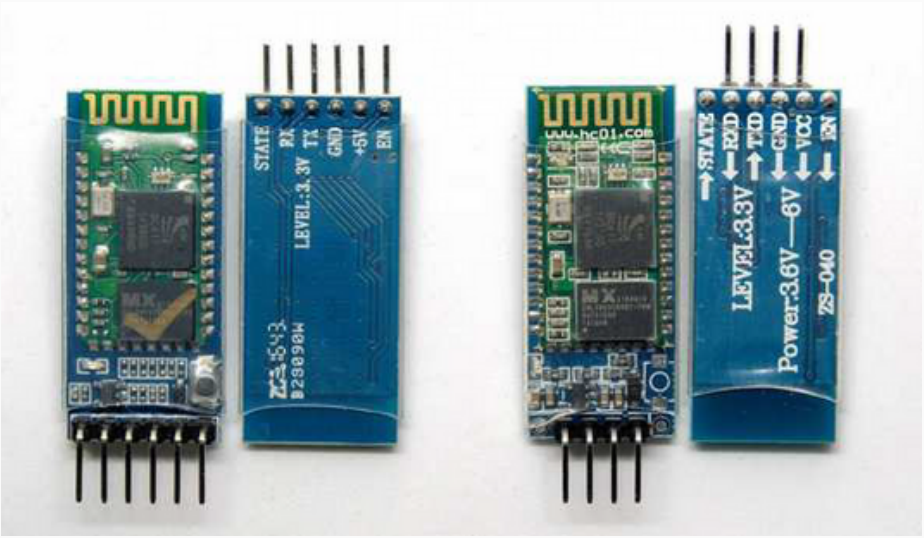
15. IR REMOTE CONTROL

IR Remote Control

The infrared controller has one transmitter controller and one receiver module. It sends an infrared (light) signal to the receiver circuit with the help of the keys on the transmitter. This signal is invisible to the human eye.

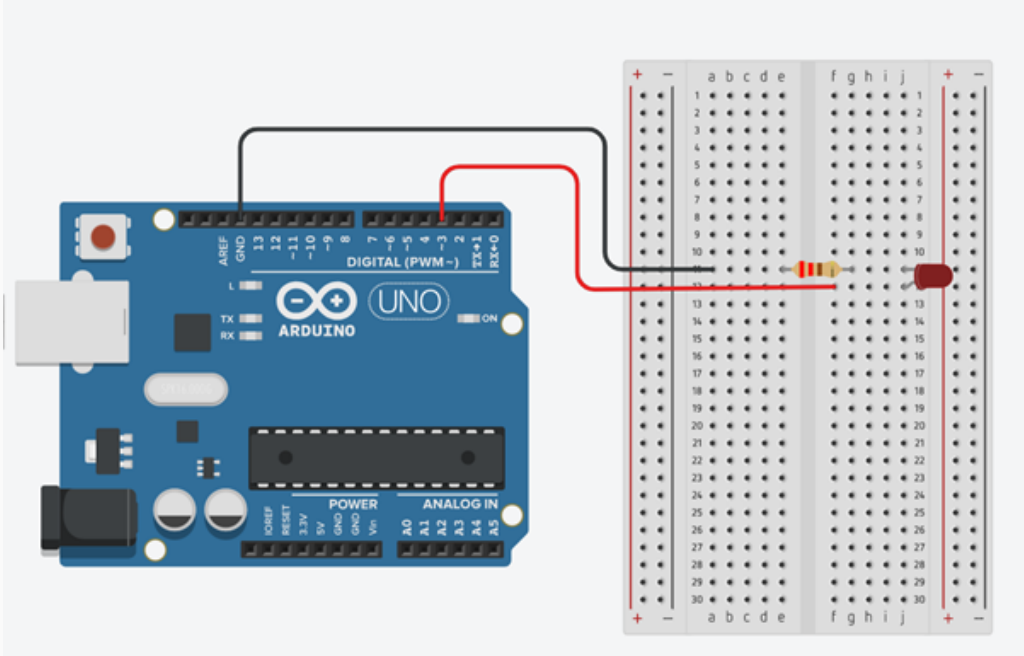
16. Bluetooth Modul

Bluetooth is wireless communication technology. When the Bluetooth module is connected to any microcontroller card circuit, it is possible to send data to the microcontroller card from different environments such as mobile phones, tablets, computers with bluetooth connection.



Bluetooth Modul

1-TURNING AN LED ON AND OFF WITH ARDUINO



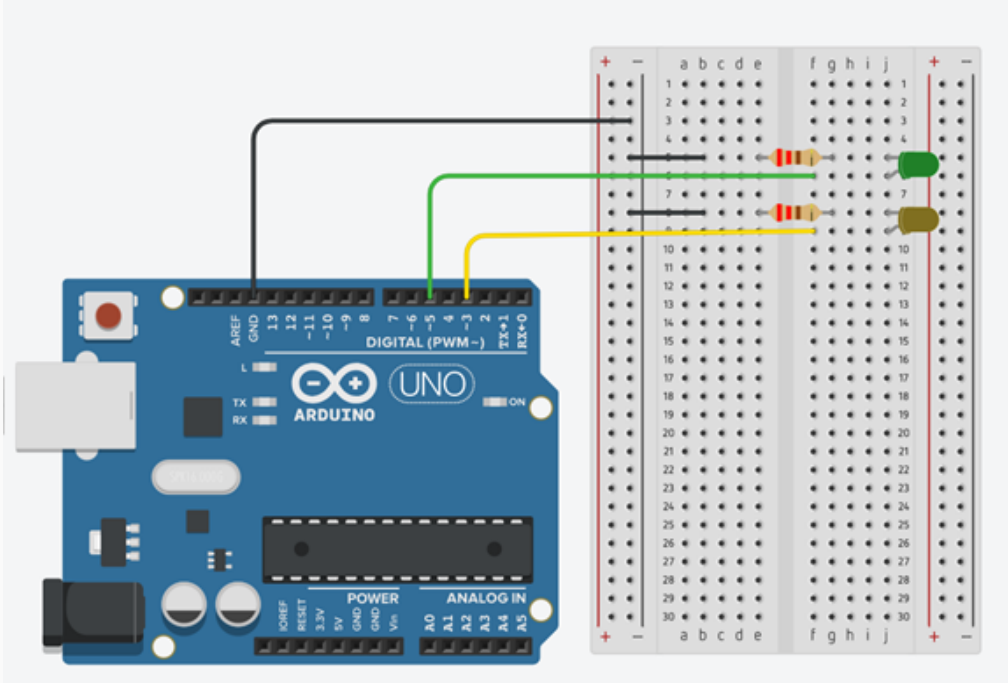
Codes

```
int led=3;

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

void loop()
{
digitalWrite(led, 1);
delay(1000);
digitalWrite(led, 0);
delay(1000);
}
```

MATERIALS

- *Arduino Uno*
- *Breadboard*
- *LED*
- *220 Ω Resistor*
- *Jumper wires*

2-TURNING TWO LEDS ON AND OFF WITH ARDUINO



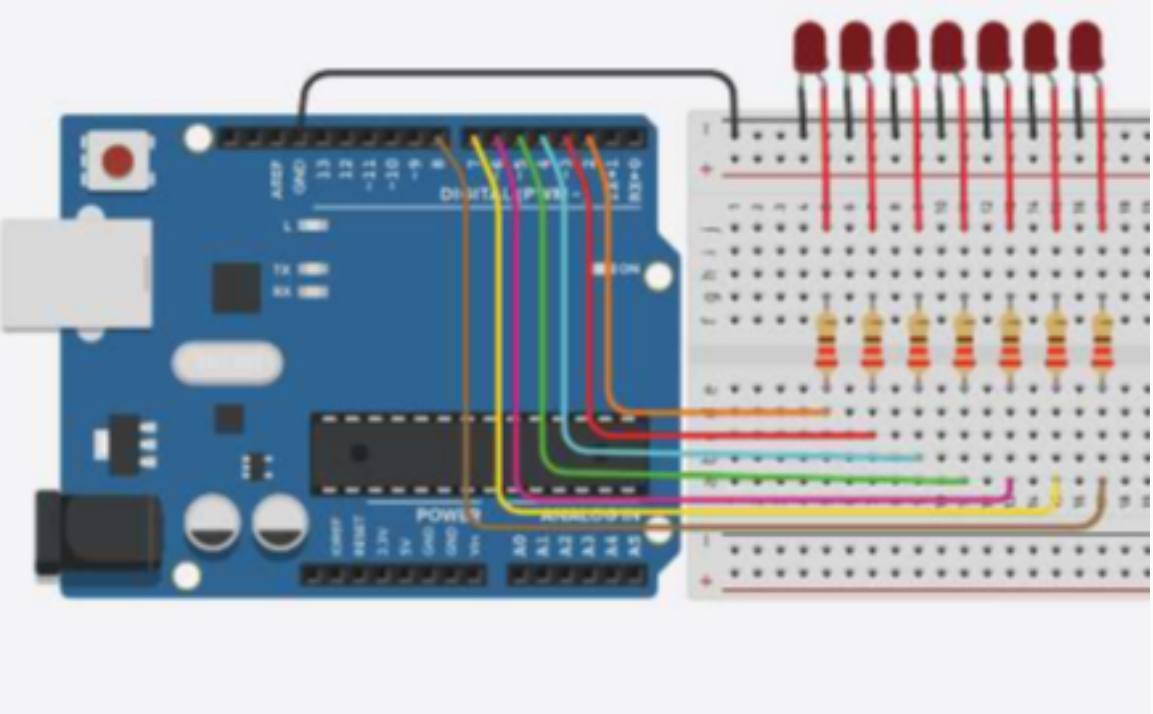
Codes

```
int led1=3;
int led2=5;

void setup()
{
 pinMode(led1, OUTPUT);
 pinMode(led2, OUTPUT);
}

void loop()
{
 digitalWrite(led1, HIGH);
 digitalWrite(led2, LOW);
 delay(1000);
 digitalWrite(led1, LOW);
 digitalWrite(led2, HIGH);
 delay(1000);
}
```

MATERIALS

- *Arduino Uno*
- *Breadboard*
- *2 LEDs*
- *2 220 Ω Resistors*
- *Jumper wires*

3-KNIGHT RIDER CIRCUIT



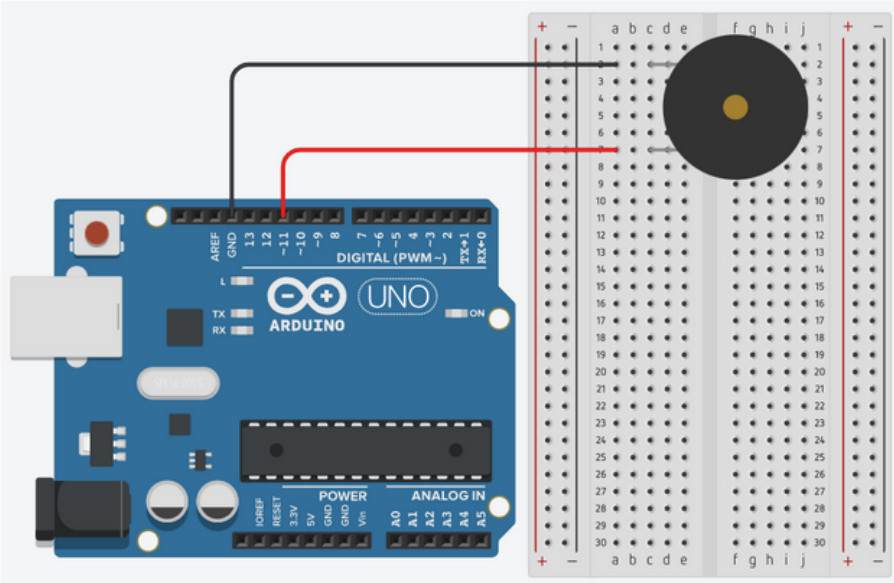
Codes

```
int i;
void setup()
{
 for(i=2;i<=8;i++)
 pinMode(i, OUTPUT);
}
void loop()
{
 for(i=2;i<=8;i++)
 {
  digitalWrite(i, 1);
  delay(1000);
 }
 for(i=2;i<=8;i++)
 {
  digitalWrite(i, 0);
  delay(1000);
  }
}
```

MATERIALS

- *Arduino Uno*
- *Breadboard*
- *7 LEDs*
- *7 220 Ω Resistors*
- *Jumper wires*

4- WE MAKE SOUNDS WITH OUR CARDS



WEB Adress :

https://github.com/robsoncouto/arduino-songs

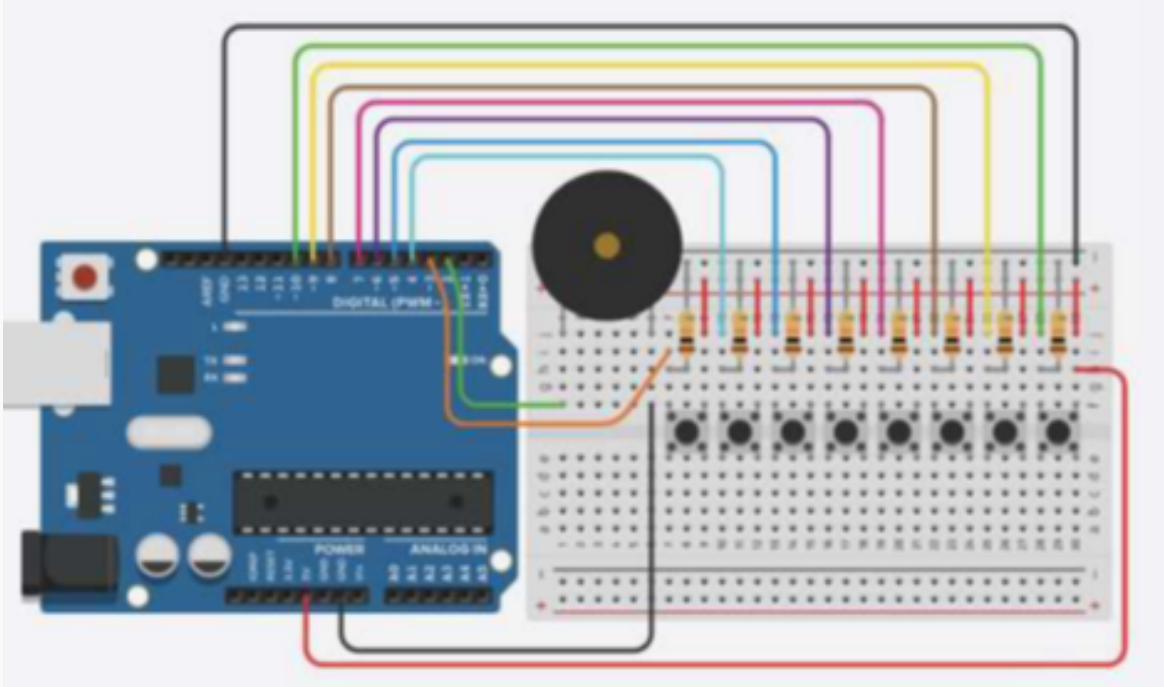
Sound List :

- Happy Birthday
- Game of Thrones
- Star Wars
- Harry Potter
- Asabranca
- Nokia Tune
- PacMan (Tune)
- Super Mario Bros.(Tune)
- Pink Panther
- Tetris (Tune)
- The Godfather
- Merry Christmas
- KeyboardCat
- Never Gonna Give Up
- Silent Night
- and more.....

MATERIALS

- Arduino Uno
- Breadboard
- Buzzer
- Jumper wires

5-SIMPLE PIANO MAKING USING A BUZZER



Notes

Note	Letter	Audio frequency
Do	C	261 Hz
Re	D	294 Hz
Mi	E	329 Hz
Fa	F	349 Hz
Sol	G	392 Hz
La	A	440 Hz
Si	B	493 Hz
Do	c	523 Hz

MATERIALS

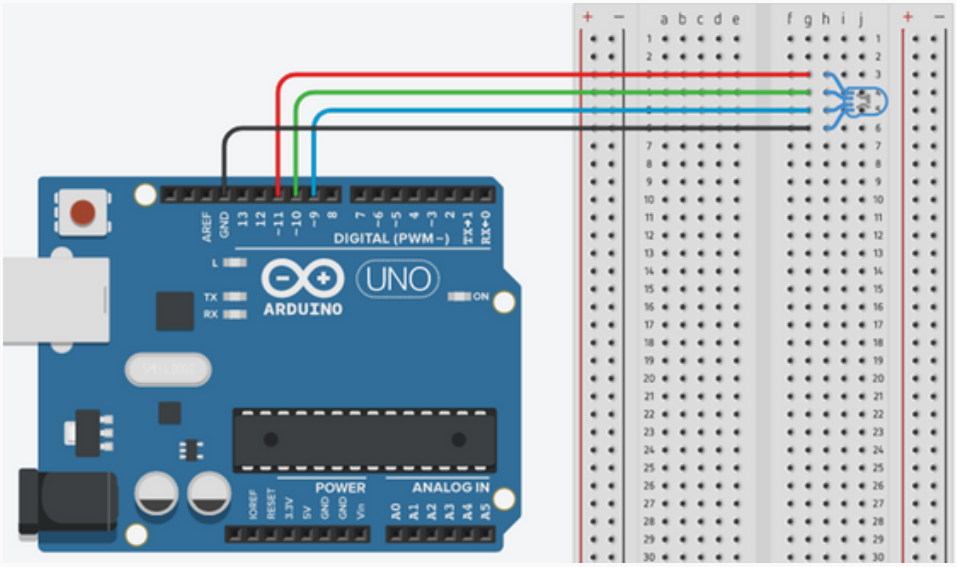
- Arduino Uno
- Breadboard
- Buzzer
- 8 Buttons
- 8 10KΩ Resistors
- Jumper wires

5-SIMPLE PIANO MAKING USING A BUZZER

Codes

```
int buzzer=2;
int C=261;
int D=294;
int E=329;
int F=349;
int G=392;
int A=440;
int B=493;
int c=523;

void setup() {
pinMode(buzzer,OUTPUT);
for (int i=3;i<=10;i++)
  pinMode(i,INPUT);
pinMode(13,OUTPUT);
}

void loop()
{
noTone(buzzer);

if (digitalRead(3)==1){
tone(buzzer,C);
delay(100);
noTone(buzzer);
delay(50);
}
else if (digitalRead(4)==1){
tone(buzzer,D);
delay(100);
noTone(buzzer);
delay(50);
}
else if (digitalRead(5)==1){
tone(buzzer,E);
delay(100);
noTone(buzzer);
delay(50);
}
else if (digitalRead(6)==1){
tone(buzzer,F);
delay(100);
noTone(buzzer);
delay(50);
}
else if (digitalRead(7)==1){
tone(buzzer,G);
delay(100);
noTone(buzzer);
delay(50);
}
else if (digitalRead(8)==1){
tone(buzzer,A);
delay(100);
noTone(buzzer);
delay(50);
}
else if (digitalRead(9)==1){
tone(buzzer,B);
delay(100);
noTone(buzzer);
delay(50);
}
else if (digitalRead(10)==1){
tone(buzzer,c);
delay(100);
noTone(buzzer);
delay(50);
} }
```

6- CREATE A COLOR YOURSELF



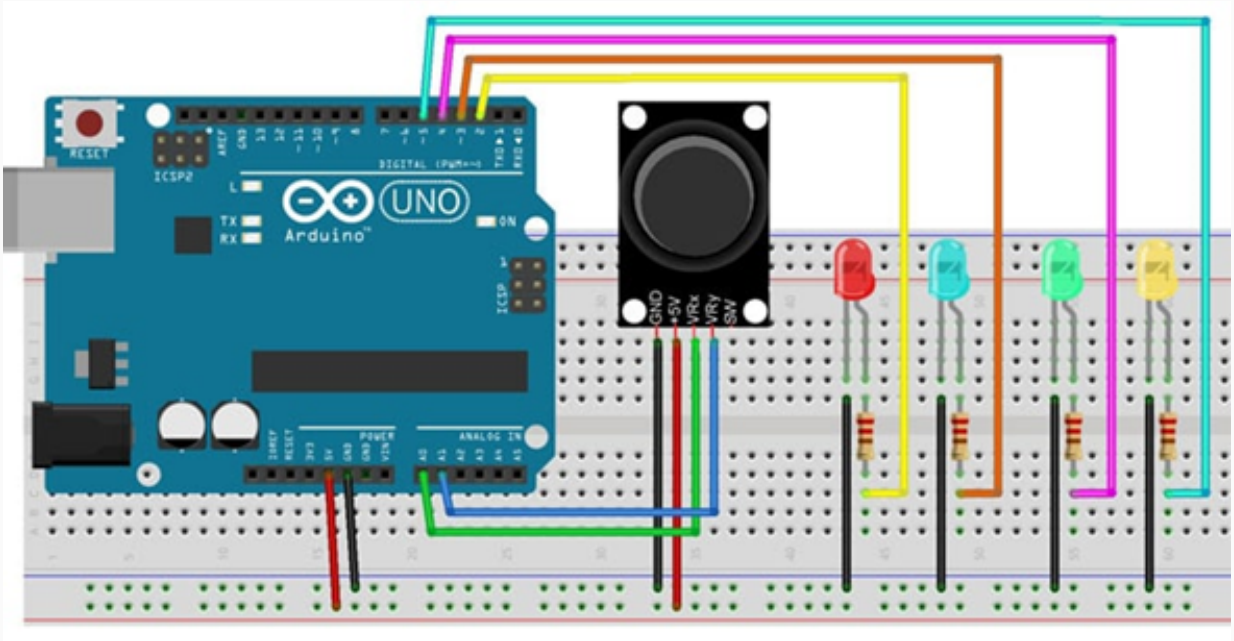
Codes

```
int r=11;
int g=10;
int b=9;
void setup() {
 pinMode(b,OUTPUT);
 pinMode(g,OUTPUT);
pinMode(r,OUTPUT);
}
void loop() {
analogWrite(r,255);
analogWrite(b,0);
analogWrite(g,0);
delay(200);
analogWrite(r,0);
analogWrite(b,0);
analogWrite(g,255);
delay(200);
analogWrite(r,0);
analogWrite(b,255);
analogWrite(g,0);
delay(200);
}
```

MATERIALS

- Arduino Uno
- Breadboard
- RGB Led
- Jumper wires

7-JOYSTICK MAKING



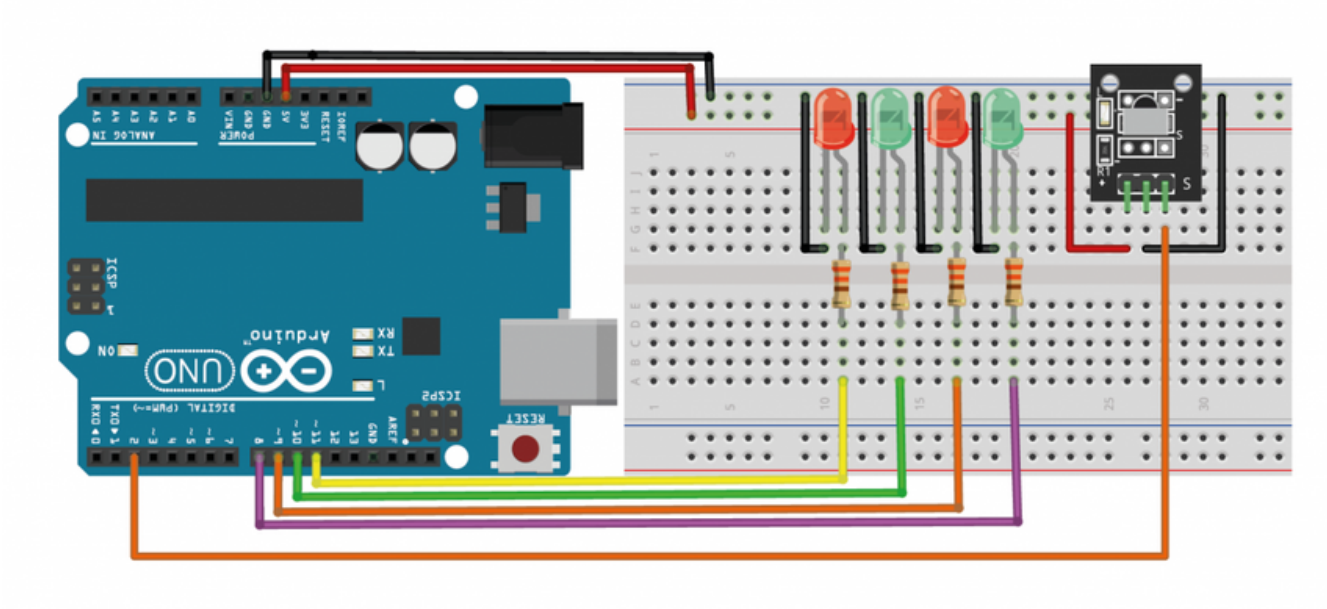
Codes

```
int xPin = A0;
int yPin = A1;
int ledR=2;
int ledB=3;
int ledG=4;
int ledY=5;
int xPosition = 0;
int yPosition = 0;

void setup() {
pinMode(xPin, INPUT);
pinMode(yPin, INPUT);
pinMode(ledR,OUTPUT);
pinMode(ledB,OUTPUT);
pinMode(ledG,OUTPUT);
pinMode(ledY,OUTPUT);
}
void loop() {
xPosition = analogRead(xPin);
yPosition = analogRead(yPin);
if(xPosition<10)
  digitalWrite(ledR,HIGH);
else
  digitalWrite(ledR,LOW);
if(xPosition>1000)
  digitalWrite(ledB,HIGH);
else
 digitalWrite(ledB,LOW);
if(yPosition<10)
 digitalWrite(ledG,HIGH);
else
 digitalWrite(ledG,LOW);
if(yPosition>1000)
 digitalWrite(ledY,HIGH);
else
digitalWrite(ledY,LOW);
}
```

MATERIALS

- Arduino Uno
- Breadboard
- 4 LEDs
- 4 10KΩ Resistors
- 1 Joystick
- Jumper wires

8- LET'S DO A REMOTE CONTROL



Codes

```
#define CH1 0xFFA25D
#define CH 0xFF629D
#define CH2 0xFFE21D
#define PREV 0xFF22DD
#define NEXT 0xFF02FD
#define PLAYPAUSE 0xFFC23D
#define VOL1 0xFFE01F
#define VOL2 0xFFA857
#define EQ 0xFF906F
#define BUTON0 0xFF6897
#define BUTON100 0xFF9867
#define BUTON200 0xFFB04F
#define BUTON1 0xFF30CF
#define BUTON2 0xFF18E7
#define BUTON3 0xFF7A85
#define BUTON4 0xFF10EF
#define BUTON5 0xFF38C7
#define BUTON6 0xFF5AA5
#define BUTON7 0xFF42BD
#define BUTON8 0xFF4AB5
#define BUTON9 0xFF52AD
```

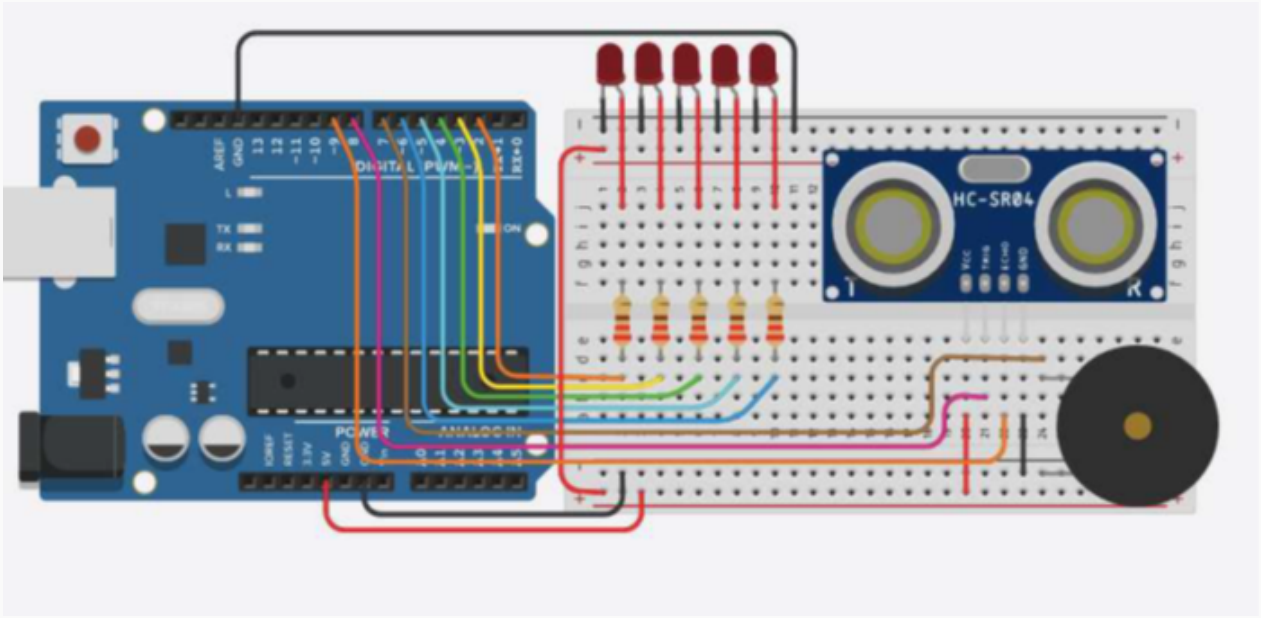
MATERIALS

- Arduino UNO
- Breadboard
- IR Remote Control Set
- 4 LEDs
- 4 220 Ω Resistors
- Jumper wires

Codes

```
#include <IRremote.h>

int RECV_PIN = 2;
IRrecv irrecv(RECV_PIN);
decode_results results;

#define BUTON0 0xFF6897
#define BUTON1 0xFF30CF
#define BUTON2 0xFF18E7
#define BUTON3 0xFF7A85
#define BUTON4 0xFF10EF
#define BUTON5 0xFF38C7

int led1 = 8;
int led2 = 9;
int led3 = 10;
int led4 = 11;

void setup() {
 pinMode(led1, OUTPUT);
 pinMode(led2, OUTPUT);
 pinMode(led3, OUTPUT);
 pinMode(led4, OUTPUT);
 irrecv.enableIRIn();
}

void loop() {

 if (irrecv.decode( &results)) {
  if (results.value == BUTON1) {
   digitalWrite(led1, !digitalRead(led1));
  }
  else if (results.value == BUTON2) {
   digitalWrite(led2, !digitalRead(led2));
  }
  else if (results.value == BUTON3) {
   digitalWrite(led3, !digitalRead(led3));
  }
 else if (results.value == BUTON4) {
   digitalWrite(led4, !digitalRead(led4));
  }
  else if (results.value == BUTON0) {
   digitalWrite(led1, LOW);
   digitalWrite(led2, LOW);
   digitalWrite(led3, LOW);
   digitalWrite(led4, LOW);
  }
 else if (results.value == BUTON5) {
   digitalWrite(led1, HIGH);
   digitalWrite(led2, HIGH);
   digitalWrite(led3, HIGH);
   digitalWrite(led4, HIGH);
  }
  irrecv.resume();
 }
}
```

9- MAKING SIMPLE PARKING SENSOR



Codes

```
int buzzer=13;
void setup() {
Serial.begin(9600);
pinMode(8,OUTPUT);//Trig
pinMode(9,INPUT); //Echo
for (int i=2;i<=7;i++)
pinMode(i,OUTPUT);
pinMode(buzzer,OUTPUT);
}

void loop() {
long duration, distance;
  digitalWrite(8, LOW);
  delayMicroseconds(2);
  digitalWrite(8, HIGH);
  delayMicroseconds(10);
  digitalWrite(8, LOW);
  duration = pulseIn(9, HIGH);
  distance = duration / 58.2;
  delay(1);
Serial.print("Distance=");
Serial.println(distance);
```

MATERIALS

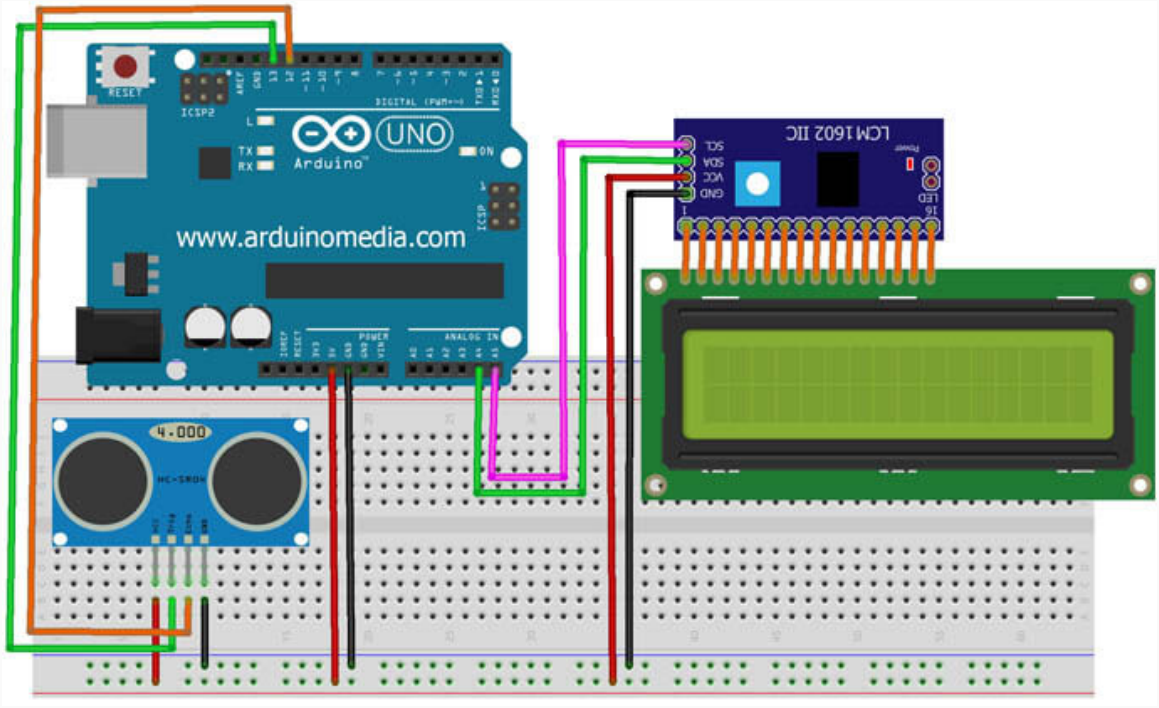
- Arduino Uno
- Breadboard
- 5 LEDs
- 5 Resistors
- HC-SR04 Ultrasonik Sensor
- Buzzer
- Jumper kablo

9-MAKING SIMPLE PARKING SENSOR

Codes

```
if( (distance>30) and (distance<=40))
{
led(2);
melody(distance * 10);
}
else
turnThemAllOff();

if( (distance>20) and (distance<=30))
{
led(3);
melody(distance * 10);
}
else
turnThemAllOff();

if( (distance>10) and (distance<=20))
{
led(4);
melody(distance * 10);
}
else
turnThemAllOff();

if( (distance>5) and (distance<=10))
{
led(5);
melody(distance * 10);
}
else
turnThemAllOff();

if(distance<=5)
{
led(6);
melody(distance * 10);
}
else
turnThemAllOff();
}

int melody(int dly)
{
  tone(buzzer, 440);
  delay(dly);
  noTone(buzzer);
  delay(dly);
}

int led(int count)
{
 for(int i=2;i<=count;i++)
  digitalWrite(i,1);
  }

void turnThemAllOff(){
for (int i=2;i<=7;i++)
digitalWrite(i, LOW);
noTone(buzzer);
}
```

10- MAKE THE DIGITAL METER YOURSELF



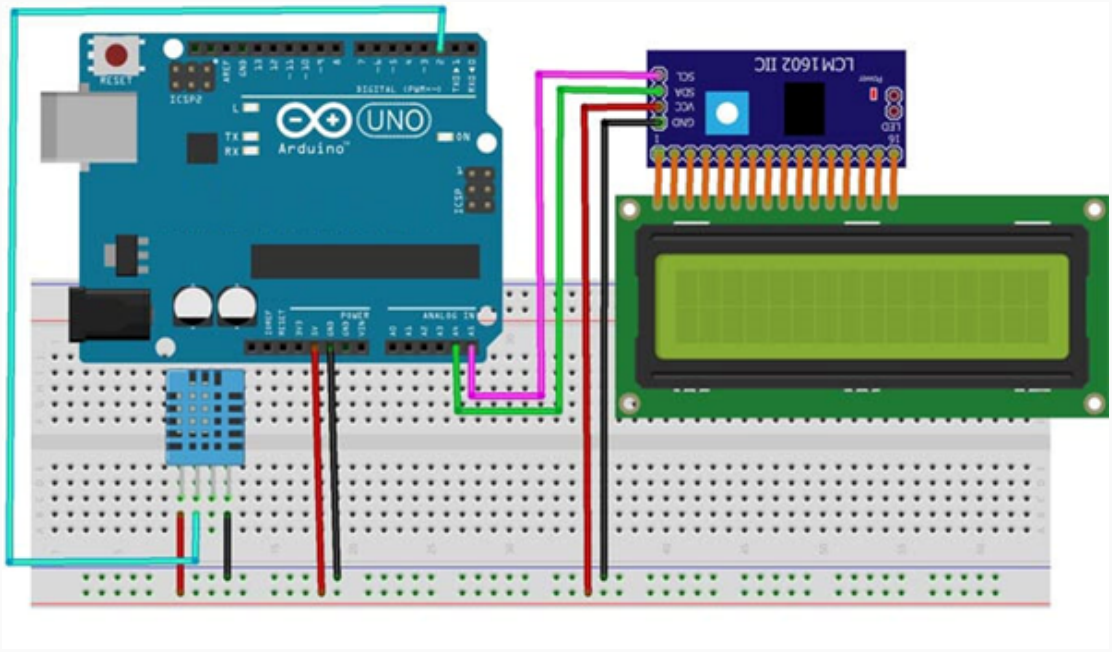
Codes

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
lcd.begin();
lcd.backlight();
Serial.begin(9600);
pinMode(13,OUTPUT);//Trig
pinMode(12,INPUT); //Echo
}
void loop() {
digitalWrite(13,1);
delay(1);
digitalWrite(13,0);
int time=pulseIn(12,1);
int distance=(time/2)/28.97;
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Distance:");
lcd.setCursor(0, 1);
lcd.print(distance);
lcd.print("cm");
delay(100); }
```

MATERIALS

- Arduino Uno
- Breadboard
- 2X16 LCD Display
- HC-SR04 Ultrasonik Sensor
- Jumper wires

11- TELLING THE AIR TEMPERATURE (LCD USING)



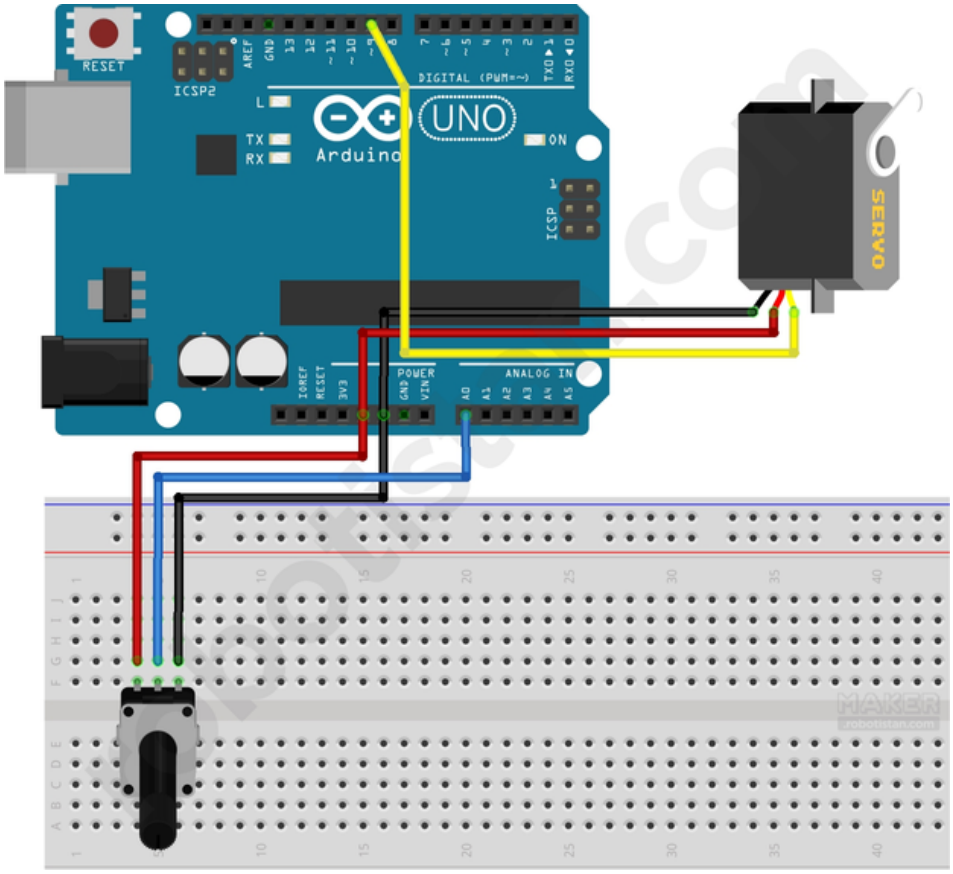
Codes

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
#include <dht11.h>
#define DHT11PIN 2
dht11 DHT11;

void setup()
{
   lcd.begin();
   lcd.backlight();
}

void loop()
{
  int chk = DHT11.read(DHT11PIN);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Temperature; ");
  lcd.setCursor(0,1);
  lcd.print((float)DHT11.temperature, 2);
  delay(1000);
}
```

MATERIALS

- Arduino Uno
- Breadboard
- 2X16 LCD Display
- DHT11 Sensor
- Jumper wires

12-ENGINE CONTROL VIA POTENTIOMETER



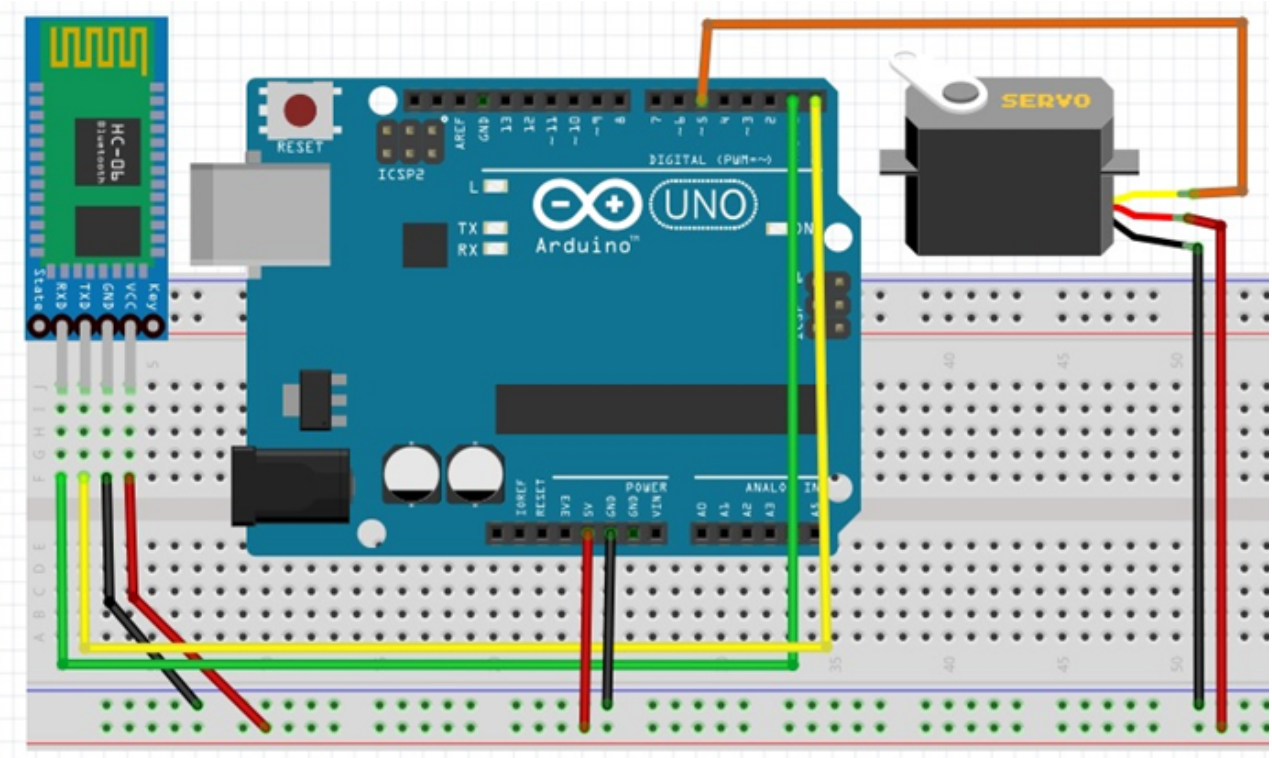
Codes

```
#include <Servo.h>

Servo myservo;
int potpin = A0;

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

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

MATERIALS

- Arduino UNO
- Breadboard
- Potentiometer
- Servo Motor (SG90)
- Jumper Wires

13-ENGINE CONTROL VIA BLUETOOTH



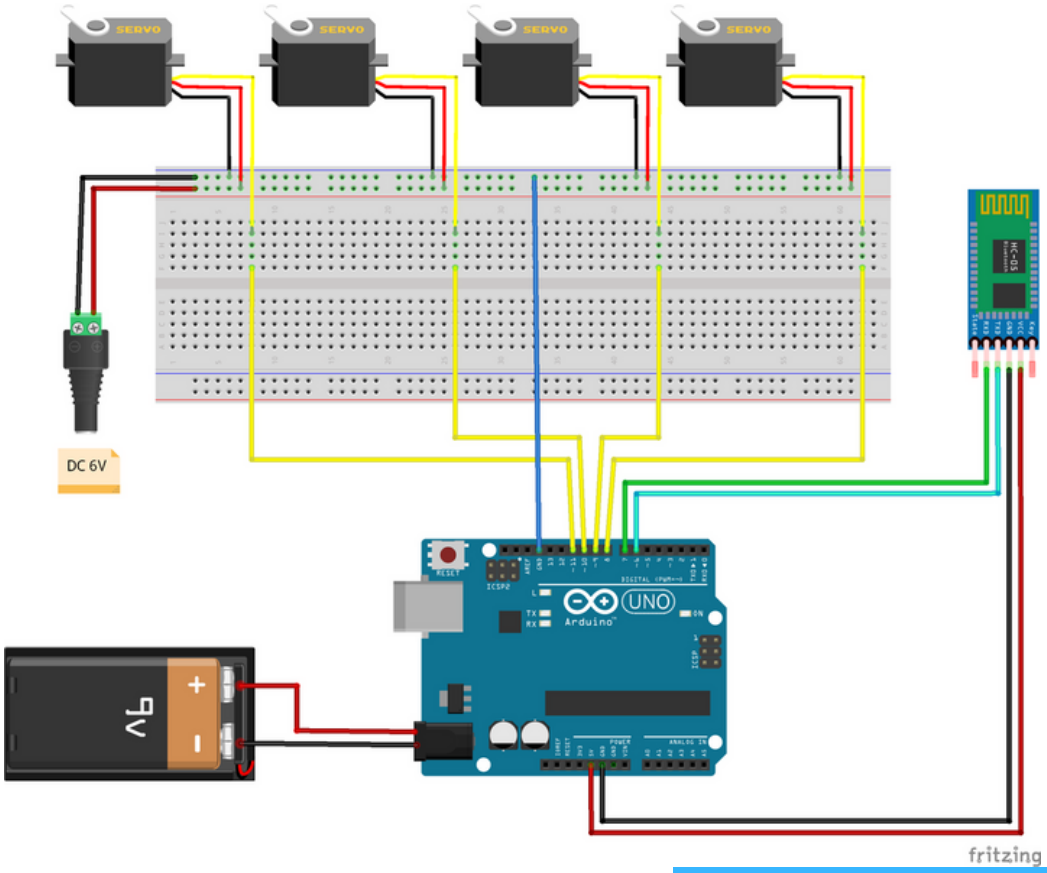
Codes

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

void setup()
{
Serial.begin(9600);
motor.attach(9);
}
void loop()
{
if(Serial.available() > 0)
{
int data = Serial.read();
motor.write(data);
 }
}
```

MATERIALS

- Arduino UNO
- Breadboard
- Bluetooth Module (HC-06)
- Servo Motor (SG90)
- Jumper Wires

14- MAKING OF ARDUINO ROBOTIC ARM



MATERIALS

- Arduino UNO
- Breadboard
- Bluetooth Module (HC-06)
- 4 Servo Motor (SG90)
- Jumper Wires

Codes

```
#include <SoftwareSerial.h>

#include <Servo.h>
Servo myservo1, myservo2, myservo3, myservo4;

int bluetoothTx = 6;
int bluetoothRx = 7;

SoftwareSerial bluetooth(bluetoothTx, bluetoothRx);

void setup()
{
  myservo1.attach(8);
  myservo2.attach(9);
  myservo3.attach(10);
  myservo4.attach(11);
  Serial.begin(9600);

  bluetooth.begin(9600);
}

void loop()
{
  if(bluetooth.available()>= 2 )
  {
    unsigned int servopos = bluetooth.read();
    unsigned int servopos1 = bluetooth.read();
    unsigned int realservo = (servopos1 *256) + servopos;
    Serial.println(realservo);

    if (realservo >= 1000 && realservo <1180)
{
    int servo1 = realservo;
    servo1 = map(servo1, 1000,1180,0,180);
    myservo1.write(servo1);
    Serial.println("servo 1 ON");
    delay(10);
 }
    if (realservo >=2000 && realservo <2180)
{
      int servo2 = realservo;
      servo2 = map(servo2,2000,2180,0,180);
      myservo2.write(servo2);
      Serial.println("servo 2 On");
      delay(10);
    }
  if (realservo >=2000 && realservo <2180)
{
    int servo2 = realservo;
    servo2 = map(servo2,2000,2180,0,180);
    myservo2.write(servo2);
    Serial.println("servo 2 On");
    delay(10);
 }

   if (realservo >=3000 && realservo < 3180)
{
    int servo3 = realservo;
    servo3 = map(servo3, 3000, 3180,0,180);
    myservo3.write(servo3);
    Serial.println("servo 3 On");
    delay(10);
 }

   if (realservo >=4000 && realservo < 4180)
{
    int servo4 = realservo;
    servo4 = map(servo4, 4000, 4180,0,180);
    myservo4.write(servo4);
    Serial.println("servo 4 On");
    delay(10);
  }

   }
}
```