ROBOTICS
THE NEXT GENERATION

ROBOTROID
by Olatus Systems Private Limited

Layout & Schematics

Online Code Editor

Getting started guides

Language Reference

CREATE

Single Sign On

Forum and Support

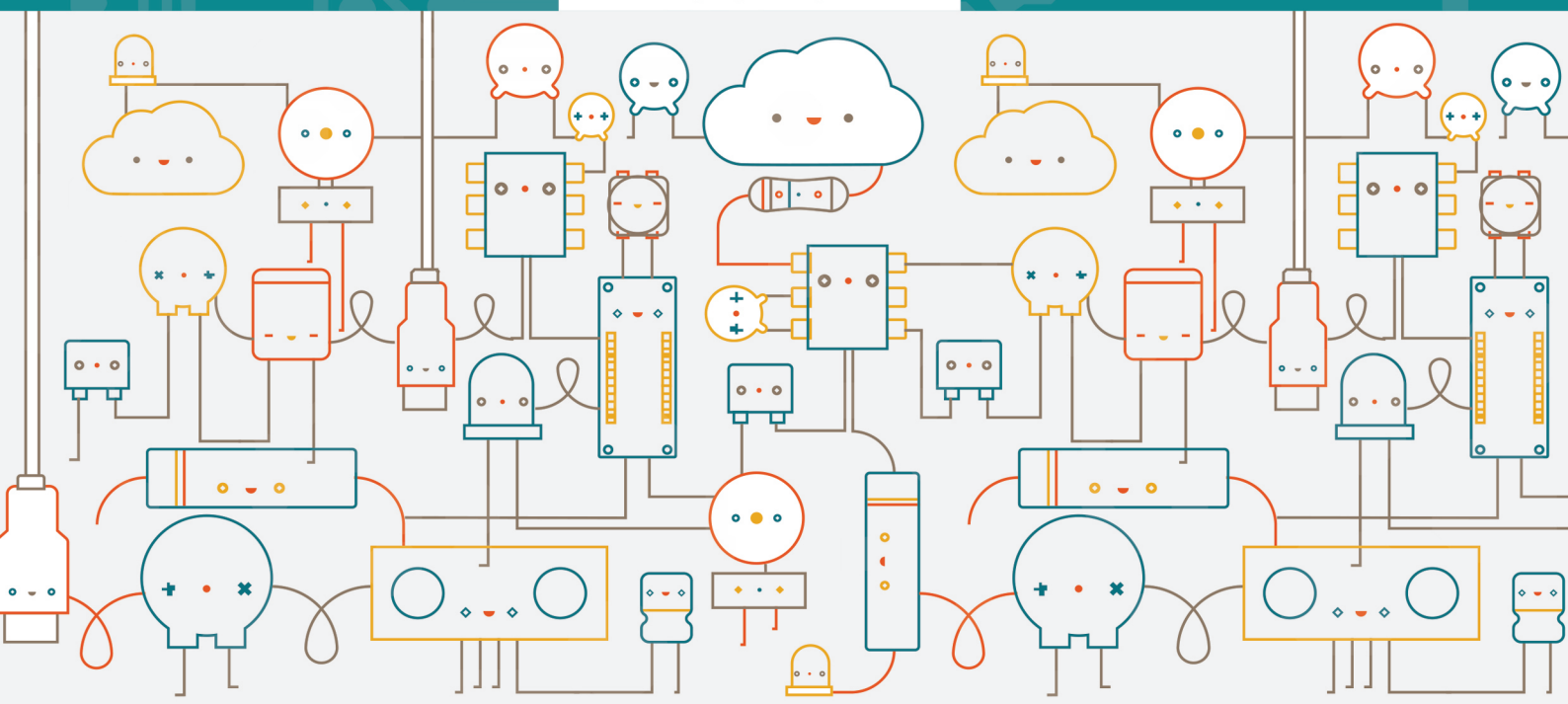
Sketchbook on Cloud

Project & Tutorials

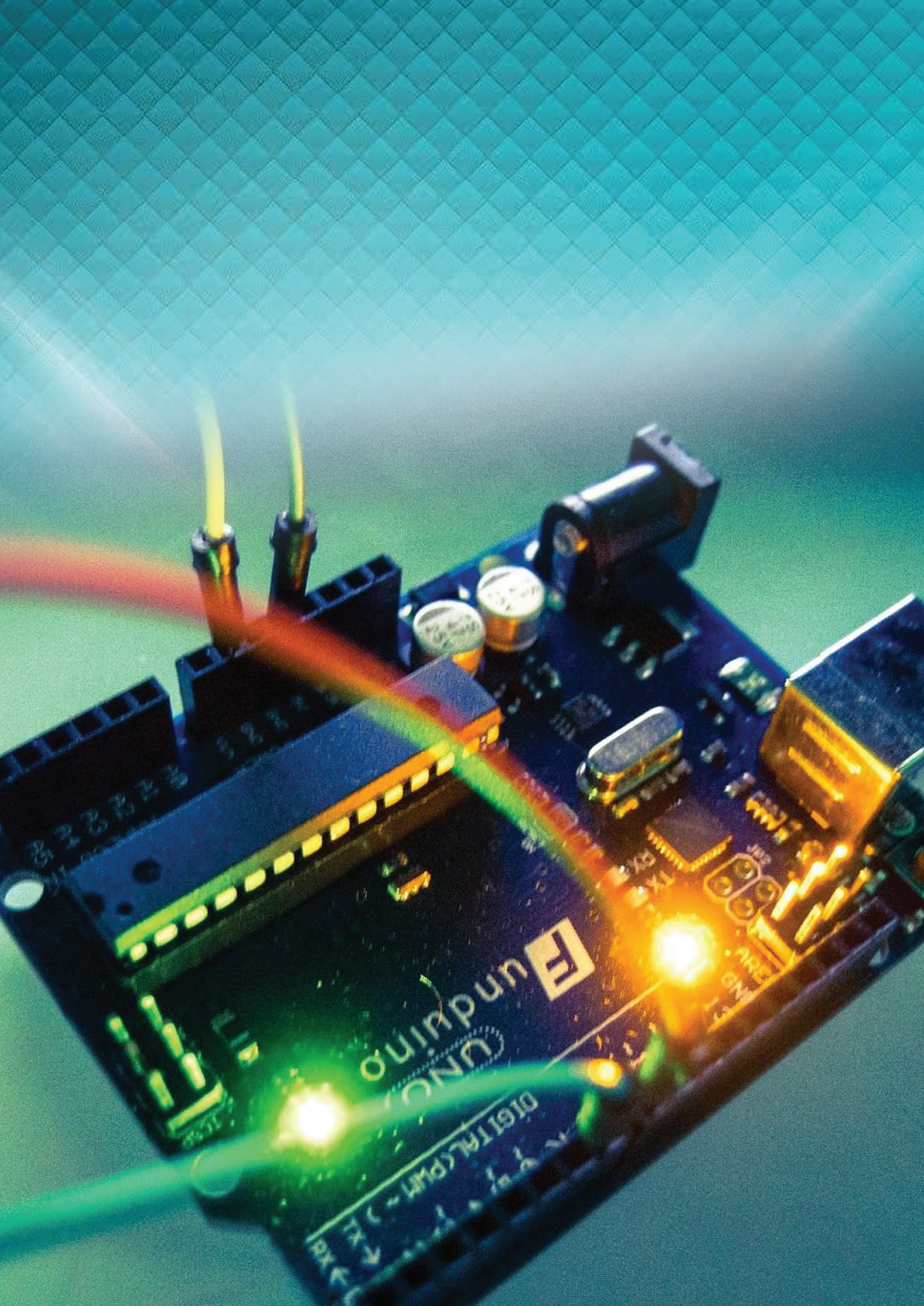
Arduino Store

Sharing features

Volume 2



ADD ON ROBOTICS ACTIVITIES

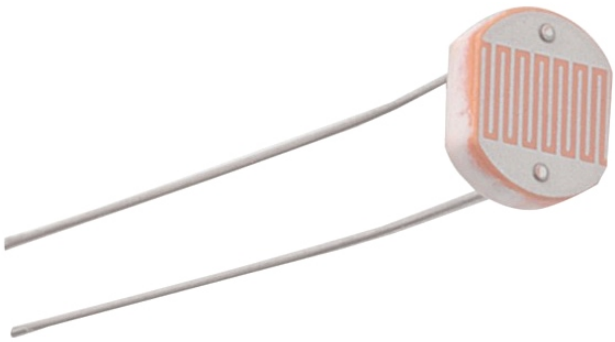

UNO
Arduino

Concept

In this project, we will learn to use a Light Dependent Resistor and a LASER pointer to create a LASER security fence. When the laser diode shines on the photo resistor (LDR), the green LED will light up to signify that the circuit is ready. When the laser beam is broken, the LED turns off and the buzzer sounds. The sensor that can be used to detect light is an LDR.

LDR

The LDR gives out an analog voltage when connected to VCC (5V), which varies in magnitude in direct proportion to the input light intensity on it. That is, the greater the intensity of light, the greater the corresponding voltage from the LDR will be. Since the LDR gives out an analog voltage, it is connected to the analog input pin on the Arduino.



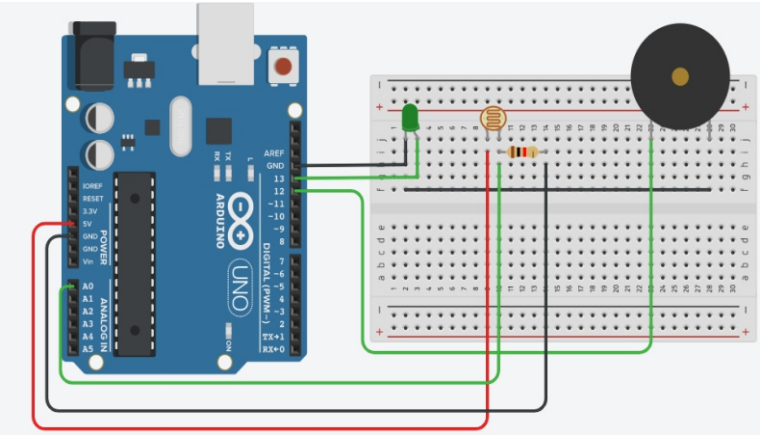
LASER DIODE

The term laser originated as an acronym: **Light Amplification by Stimulated Emission of Radiation**. **Laser is a narrow beam of light particles** emitted by specially made laser diodes. Laser diode is similar to an ordinary LED, but it generates a beam of high intensity light. A laser is a device in which atoms vibrate to produce consistent beam of a single wavelength.

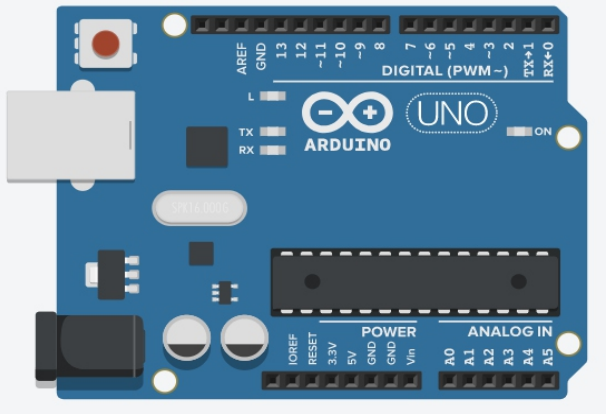
COMPONENTS REQUIRED

- Arduino Uno - 1
- Breadboard mini - 1
- Potentiometer - 1
- Popscicle Stick - 2
- Buzzer - 1 and LED - 1
- Laser Diode - 1 & LDR - 1
- Jumper Cable - 10
- Glue Drops - 2

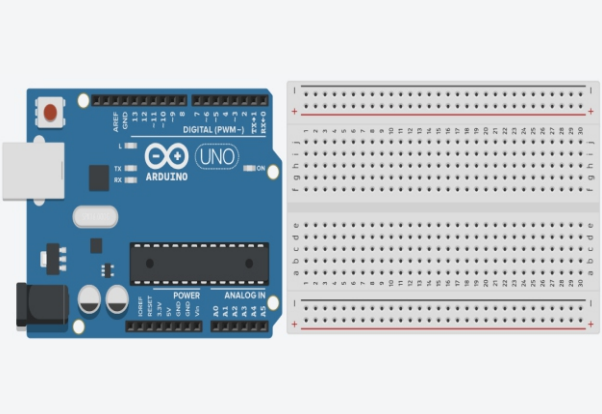
Activity:- Laser Fence



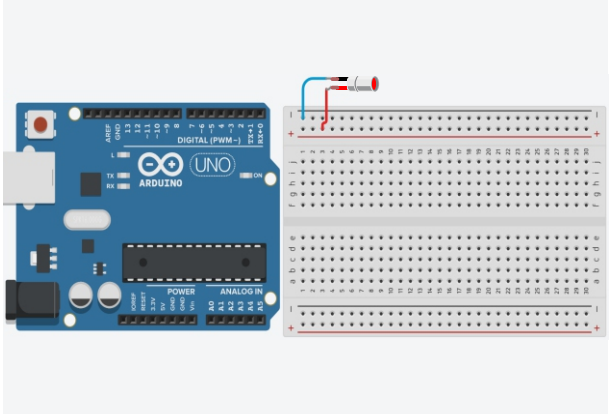
1 Take an Arduino Board



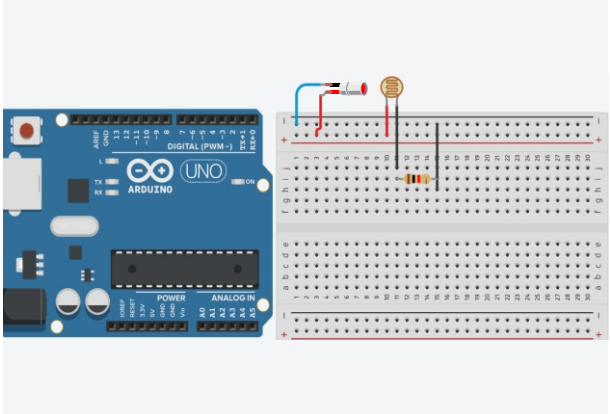
2 Place a breadboard along the Arduino board



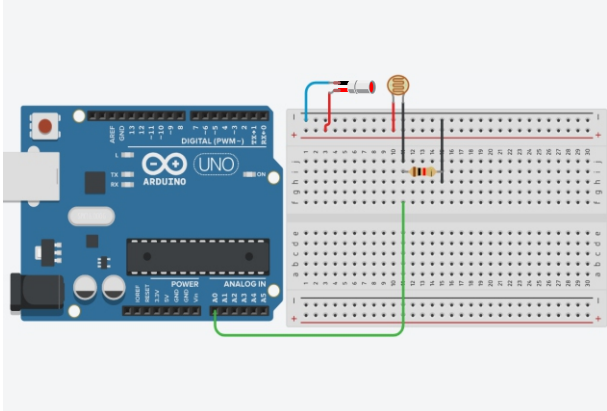
3 Place your LASER diode on the breadboard and connect the negative wire (blue) of the diode to the negative rail of the breadboard and positive wire (red) to the positive rail of the breadboard.



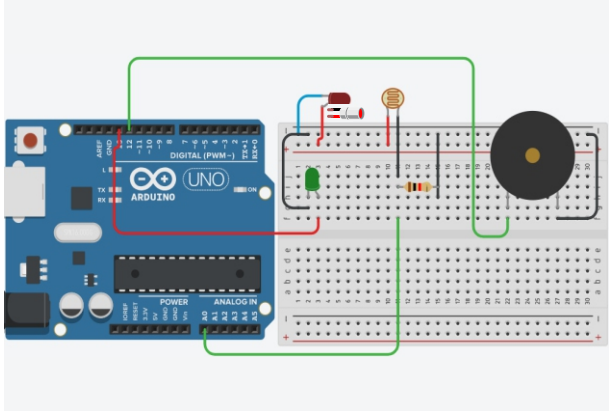
4 Connect a LDR to the breadboard and connect a 1 kiloOhm resistor in series with the LDR as shown. Then connect the components to the power rails as shown.



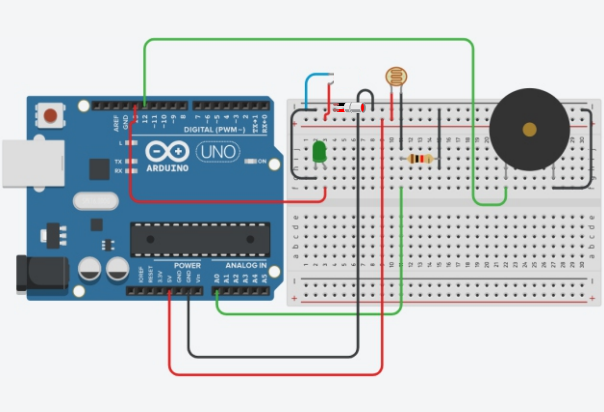
5 Connect one end of the resistor to the A0 pin of the Arduino as shown. This is the correct configu -ration of using a LDR and the A0 pin will now act as the input from the LDR.



6 Place a buzzer on the breadboard. Connect the negative leg(short) of the buzzer to the negative rail of the Arduino and the positive leg(long) to pin no. 12 of the Arduino board. Also Connect an LED at pin no. 13 of the Arduino.



7 Finally, connect the negative rail of the breadboard to the GND pin of the Arduino and the positive rail of the breadboard to the 5V of the Arduino.



8 Open the Arduino IDE Software and write the code given with the activity.

```
int LDR = A0;    //Analog 0 to Photoresistor
int Loud = 12;   //Pin 12 to Loudspeaker
void setup() {
  pinMode(LDR, INPUT);   //Photoresistor is set as an input
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
  Serial.begin(9600);   //set serial monitor at 9600 baud
}
void loop()
{
int Read = analogRead(LDR);  // "Read" reads analog0 data
Serial.println(Read);   // Print that data
```

Note: Upload the code and make sure that the LASER beam is falling on the LDR. Use your finger to break the beam, the buzzer should beep and the LED should glow. If it does not, change the value of Read in the above marked line to suit your needs (increase or decrease the values, by looking at the values on the Serial monitor.)

CODE

```
int LDR = A0;                          //Analog 0 to Photoresistor
int Loud = 12;                         //Pin 12 to Loudspeaker
void setup()
{
 pinMode(LDR, INPUT);                  //Photoresistor is set as an input
 pinMode(12, OUTPUT);
 pinMode(13, OUTPUT);
 Serial.begin(9600);                   //set serial monitor at 9600 baud
}
void loop()
{
 int Read = analogRead(LDR);           // "Read" reads analog0 data
 Serial.println(Read);                 // Print that data
 if (Read < 120)                       //if the value is less than 120 (this can be modified
                                       based on your lighting condition)
 {
  digitalWrite(Loud, HIGH);            //speaker turns on
  digitalWrite(13, LOW);
 }
 else                                  //if the value is greater than 120
 {
  digitalWrite(Loud, LOW);             // speaker will turn it off
  digitalWrite(13, HIGH);
 }
 delay(1000);                          //run every second
```

Assessment

Tick The Correct One:

Q1. Which sensor can be used to detect light?

- ☐ LDR
- ☐ Piezo Buzzer
- ☐ Ultrasonic
- ☐ IR Sensor

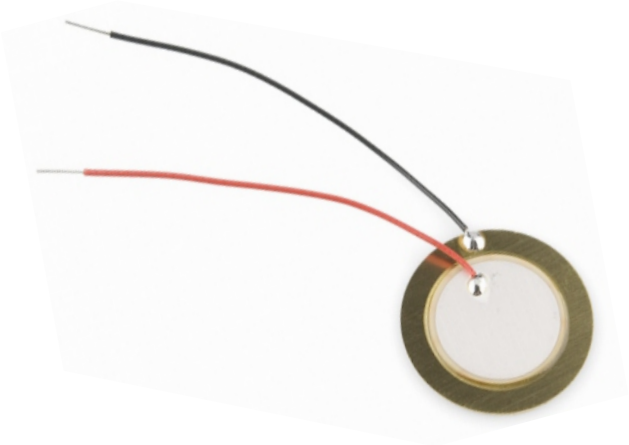
Q2. What is the full form of LASER_____?

LDR analogRead LASER Diode

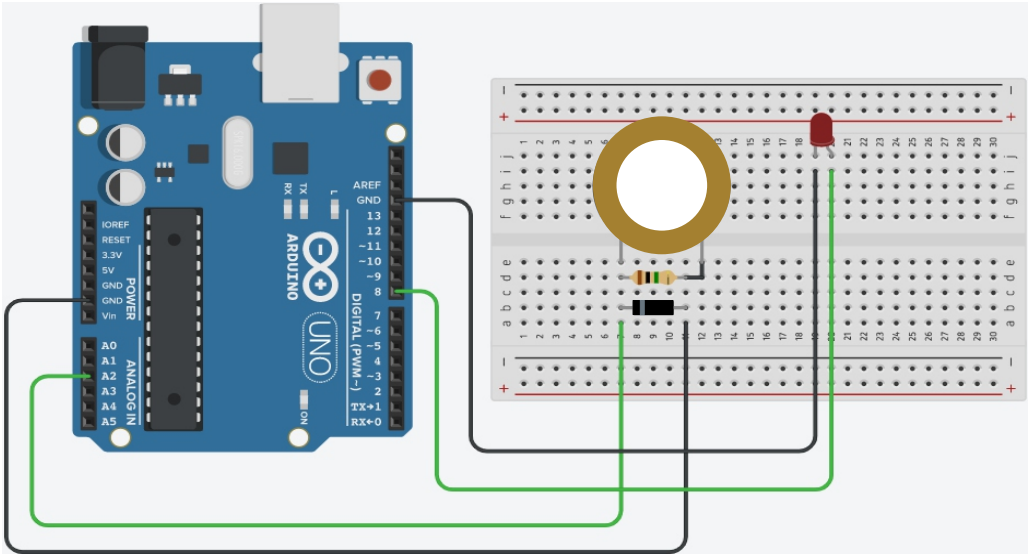
Concept

In this activity we will show you how to use a Piezo element to detect vibration, in this case, a knock on a door, table, or other solid surface.

A **piezo** is an electronic device that generates a voltage when it's physically deformed by a vibration, sound wave, or mechanical strain. Similarly, when you put a voltage across a piezo, it vibrates and creates a tone. Piezos can be used both to play tones and to detect tones. The sketch reads the piezo output using the analogRead () command, encoding the voltage range from 0 to 5 volts to a numerical range from 0 to 1023 in a process referred to as analog-to-digital conversion, or ADC. If the sensors output is stronger than a certain threshold, your board will send the string "Knock!" to the computer over the serial port.



Activity:- Secret Knock Detector



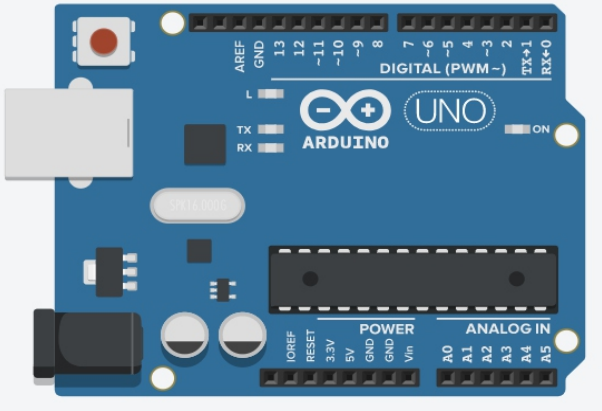
CHAPTER - 2
SECRET KNOCK DETECTOR

*In the code below, the incoming piezo data is compared to a threshold value set by the user. Try raising or lowering this value to increase your sensor's overall sensitivity.

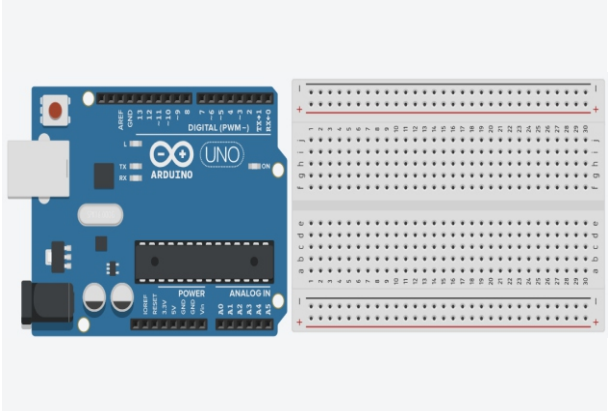
COMPONENTS REQUIRED

- Arduino Uno - 1
- Breadboard mini - 1
- Diode (4001) - 1
- LED - 2
- Piezo Plate with Solder
- 1MΩ Resistor - 1
- Jumper wire - 11

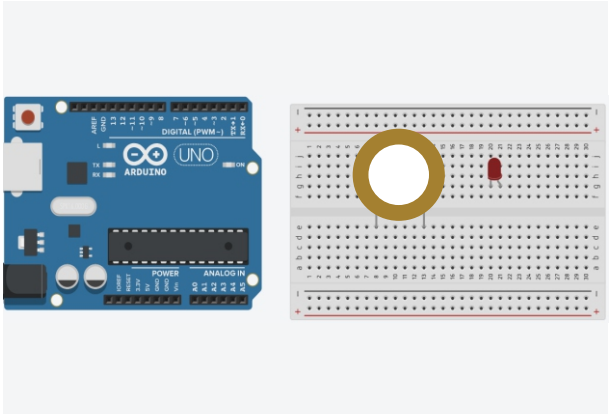
1 Take an Arduino Board.



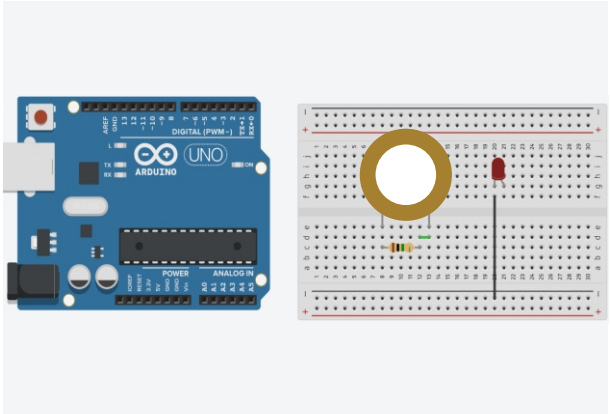
2 Place a breadboard along the Arduino board.



3 Place a piezo plate and a LED on the breadboard as shown.

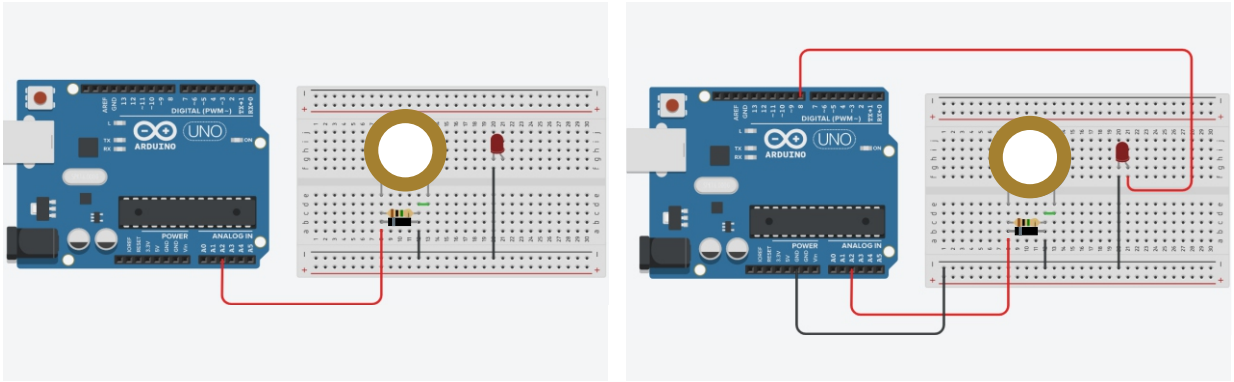


4 Connect a 1 Mega Ohm resistor in parallel to the piezo plate as shown. Connect the negative leg of the LED to the negative rail of the breadboard.



5 Now place a diode parallel to the resistor placed in the previous step as shown. Connect the nega -tive leg of the diode (the side with a silver ring) to the A2 pin of the Arduino and the other pin of the diode to the negative rail of the breadboard.

6 Connect the positive leg of the LED to pin no. 8 of the Arduino. Connect the negative rail of the breadboard to the GND pin of the Arduino.



7 Open up the Arduino IDE and write the code given with the activity.

File Edit Sketch Tools Help

sketch_aug02a§

```
if (senso
  {
 //Check to see if a Hard Knock matches the Secret Knock in the correct sequence.
   if (secretKnock[secretCounter] == 1)
   {
                                        //The Knock was correct, iterate the counter.
     secretCounter++;
Serial.println("Correct");
   }
else
   {
//The Knock was incorrect, reset the counter
secretCounter = 0;
    Serial.println("Fail");
    digitalWrite (outputPin, LOW);
   }
  //close if
  //Allow some time to pass before sampling again to ensure a clear signal.
 delay (100);
                                        //Gentle knock (LOW) is detected
 }
 else if (sensorReading >= thresholdLOW)
 {
```

CODE

```
const int outputPin = 8;                  // led indicator connected to digital pin
const int knockSensor = A2;               // the piezo is connected to an analog pin
const int thresholdHIGH = 120;
                                          // threshold value to decide when the detected knock
                                            is hard (HIGH)
const int thresholdLOW = 70;
                                          // threshold value to decide when the detected knock is
                                            gentle (LOW)
const int secretKnockLength = 3;          // How many knocks are in your secret knock
/* This is the secret knock sequence * 0 represents a LOW or quiet knock *
1 represents a HIGH or loud knock *
  The sequence can be as long as you like, but longer codes increase the difficulty of matching */
const int secretKnock[secretKnockLength] = {0, 0, 1};
int secretCounter = 0;
// this tracks the correct knocks and allows you to move through the sequence

int sensorReading = 0;                    // variable to store the value read from the sensor pin
void setup ()
{
  pinMode (outputPin, OUTPUT);            // Set the output pin as an OUTPUT
  Serial.begin(9600);                     // Begin Serial Communication.
}
void loop ()
{
  // read the piezo sensor and store the value in the variable sensorReading:
  sensorReading = analogRead(knockSensor);
  // First determine is knock if Hard (HIGH) or Gentle (LOW)
  //Hard knock (HIGH) is detected
  if (sensorReading >= thresholdHIGH)
  {
  //Check to see if a Hard Knock matches the Secret Knock in the correct sequence.
    if (secretKnock[secretCounter] == 1)
    {
                                          //The Knock was correct, iterate the counter.
      secretCounter++;
      Serial.println("Correct");
    }
else
    {
//The Knock was incorrect, reset the counter
```

```
 secretCounter = 0;
   Serial.println("Fail");
   digitalWrite (outputPin, LOW);
  }
  //close if
 //Allow some time to pass before sampling again to ensure a clear signal.
delay (100);
                                        //Gentle knock (LOW) is detected
}
else if (sensorReading >= thresholdLOW)
{
 //Check to see if a Gentle Knock matches the Secret Knock in the correct sequence.
  if (secretKnock[secretCounter] == 0)
  {
//The Knock was correct, iterate the counter.
    secretCounter++;
   Serial.println("Correct");
  }
  else
  {
                                        //The Knock was incorrect, reset the counter.
   secretCounter = 0;
   Serial.println("Fail");
  }
 //close if
 //Allow some time to pass before sampling again to ensure a clear signal.
  delay (100);
}
//Check for successful entry of the code, by seeing if the entire array has been walked through.
if (secretCounter == (secretKnockLength))
{Serial.println("Welcome");
 //if the secret knock is correct, illuminate the LED for a couple seconds
 digitalWrite (outputPin, HIGH);
 //Reset the secret counter to 0.
 secretCounter = 0;
} }
```

Assessment

Tick The Correct One:

Q1. What is the maximum and minimum threshold you can give?

- ☐ 0/1023
- ☐ 0/255
- ☐ 50/120
- ☐ None of the above

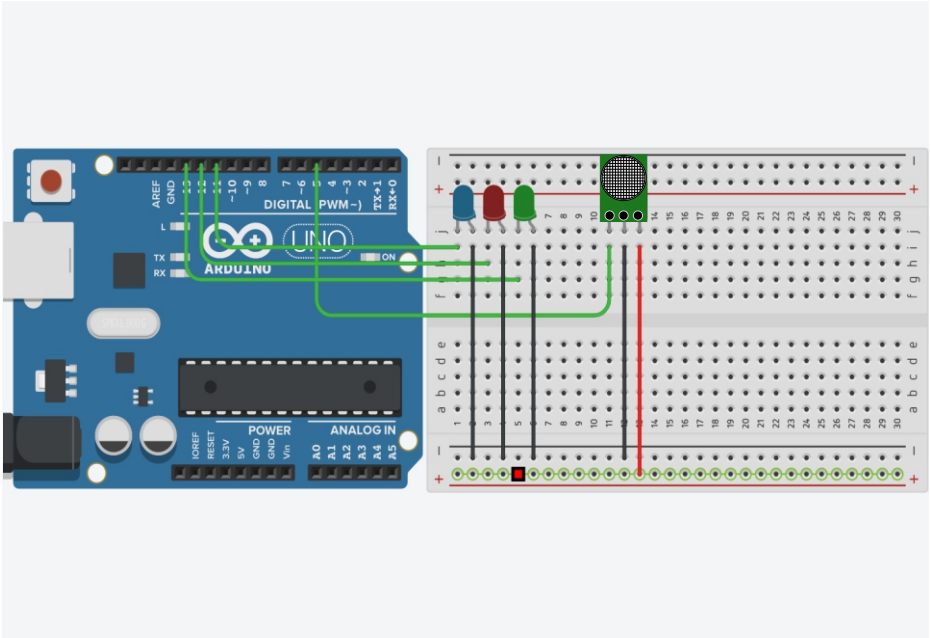
Counter | Threshold | Piezo electric Disk | analogRead

Concept

In this activity we will use a sound sensor as an input for sound from our surroundings and LED's to react to that sounds like a DISCO Ball.

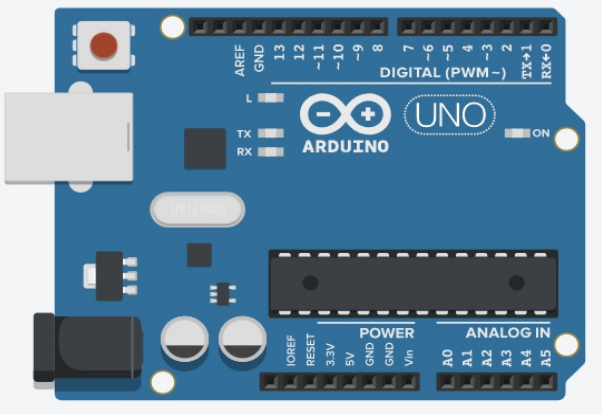
Sound Sensor
A Sound Sensor is a simple device that detects sound. It is simply put a Microphone with some processing circuit. Using a Sound Sensor, you can measure the intensity of sound from different sources like knocks, claps, loud voices, etc.

Activity:- Sound Reactive Lightning

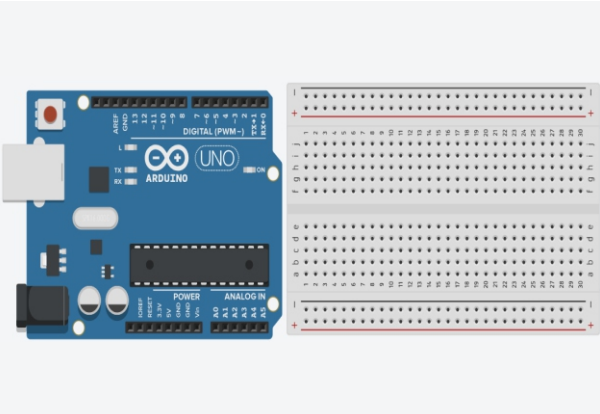


CHAPTER - 3
SOUND SENSOR

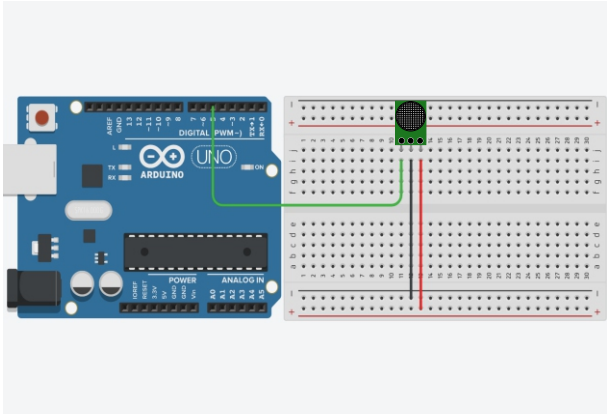
1 Take an Arduino Board



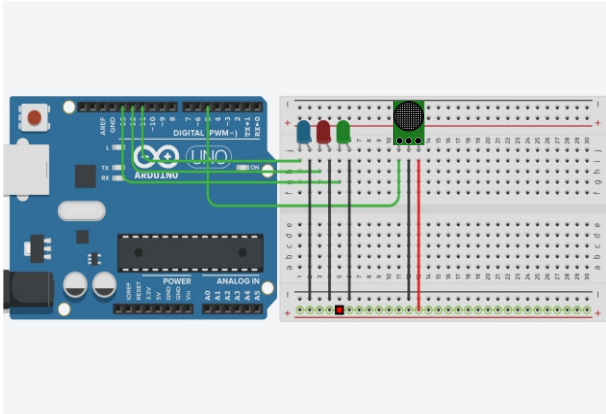
2 Place a breadboard along the Arduino board



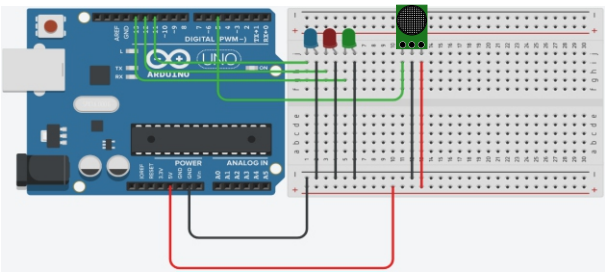
3 Place a Sound sensor on the breadboard. Connect the Vcc pin to the positive rail of the breadboard and the GND pin to the negative rail of the bread board. Connect the OUT/SIG pin to the digital pin 5.



4 Place 3 LEDS on the breadboard and connect the positive leg of those LEDs to pin nos. 13, 12 and 11 on the Arduino respectively.



5 Connect the positive rail of the breadboard to the 5v and negative rail of the breadboard to the GND pin of the Arduino board



4 Open the Arduino IDE and copy the code given with the activity.

File Edit Sketch Tools Help

```
int led1 = 13;
int led2 = 12;
int led3 = 11;
int sound = 5;
int value;
void setup()
{
pinMode(sound, INPUT);
pinMode(led1, OUTPUT);
pinMode(led2, OUTPUT);
pinMode(led3, OUTPUT);
Serial.begin(9600);
value = 0;
}
void loop()
{
```

Note: Upload the code and snap your fingers next to the sound sensor. The LEDs should start flickering with the sound. Now play some music near the sound sensor. The LEDs should start responding according to the music

CODE

```
int led1 = 13;
int led2 = 12;
int led3 = 11;
int sound = 5;
int value;
void setup()
{
pinMode(sound, INPUT);
pinMode(led1, OUTPUT);
pinMode(led2, OUTPUT);
pinMode(led3, OUTPUT);
Serial.begin(9600);
value = 0;
}
void loop()
{
value = digitalRead(sound);
Serial.println(value);
if (value == 1) //
{
digitalWrite(led1, HIGH);
digitalWrite(led2, HIGH);
digitalWrite(led3, HIGH);
}
else
{
digitalWrite(led1, LOW);
digitalWrite(led2, LOW);
digitalWrite(led3, LOW);
}
}
```

COMPONENTS REQUIRED

- Arduino Uno - 1
- Breadboard mini - 1
- Sound Sensor - 1
- LEDs - 4 of Different Colors
- Jumper Cable - 11

Assessment

Tick The Correct One:

Q1. What value is the maximum for sound sensor?

- ☐ 1024
- ☐ 1023
- ☐ 255
- ☐ None of the above

analogRead Serial.print sound sensor

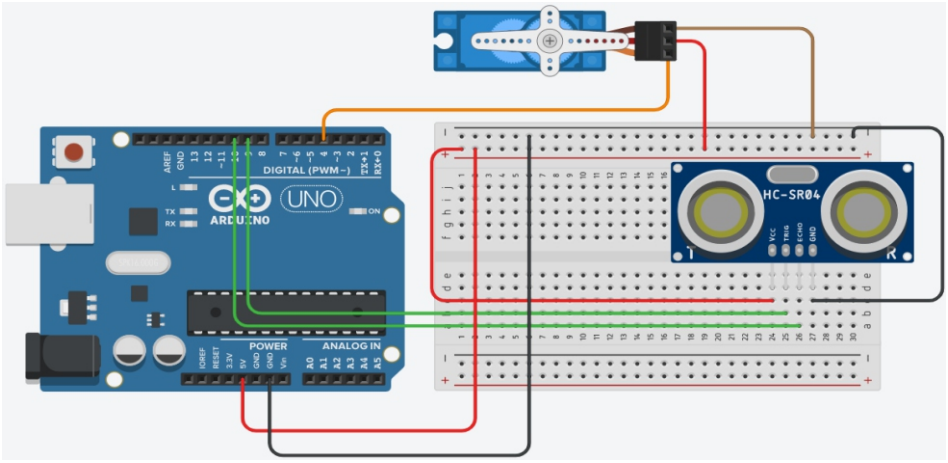
Concept

In this activity, we will learn how to use an ultrasonic sensor with an Arduino board and control a servo motor with it to create a batting mechanism that hits a ball every time it comes in front of the ultrasonic sensor

Ultrasonic Sensor
The HC-SR04 ultrasonic sensor uses Sonar to determine the distance of an object just like the bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package from 2 cm to 400 cm or 1” to 13 feet.

Servo Library
This library allows an Arduino board to control RC (hobby) servo motors. Servos have integrated gears and a shaft that can be precisely controlled. Standard servos allow the shaft to be positioned at various angles, usually between 0 and 180 degrees. Continuous rotation servos allow the rotation of the shaft to be set to various speeds.

Activity:- 1 - Servo - Baseball Batter



Definitions & Syntax:

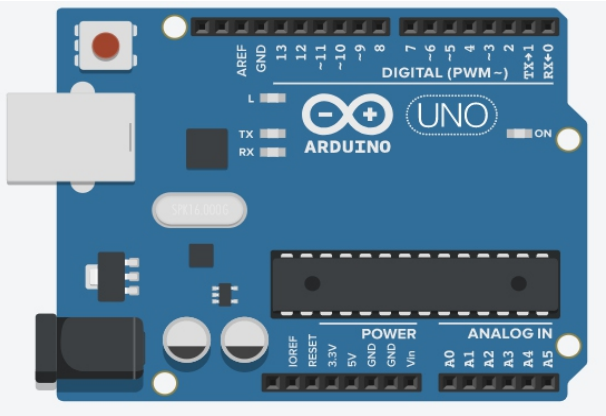
Servo.write- Writes a value to the servo, controlling the shaft accordingly. On a standard servo, this will set the angle of the shaft (in degrees), moving the shaft to that orientation. On a continuous rotation servo, this will set the speed of the servo (with 0 being full-speed in one direction, 180 being full speed in the other, and a value near 90 being no movement).

Syntax: *servo*.write(angle)
Parameters: servo: a variable of type Servo
angle: the value to write to the servo, from 0 to 180
Delay: pinMode(pin,mode)

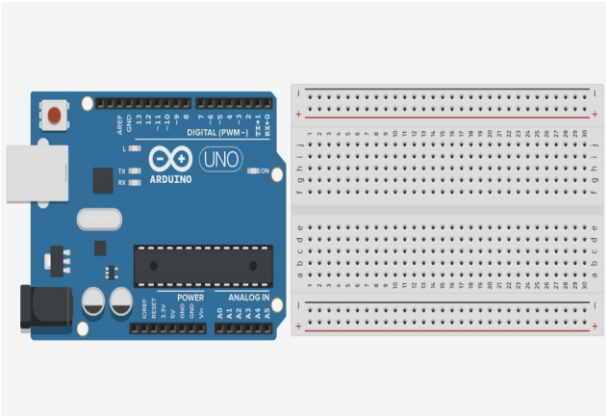
COMPONENTS REQUIRED

- Arduino Uno - 1
- Breadboard mini - 1
- Servo Motor - 1
- Ultrasonic Sensor - 1
- TT Ball - 1, Glue Drops - 7
- Popscicle Stick - 7
- Jumper Cable - 7

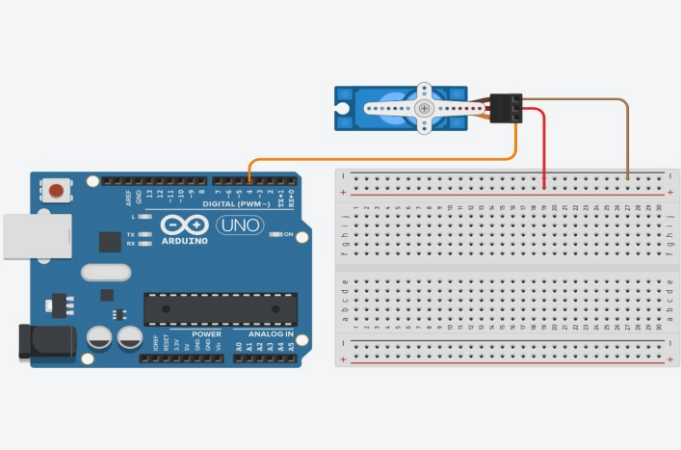
1 Take an Arduino Board



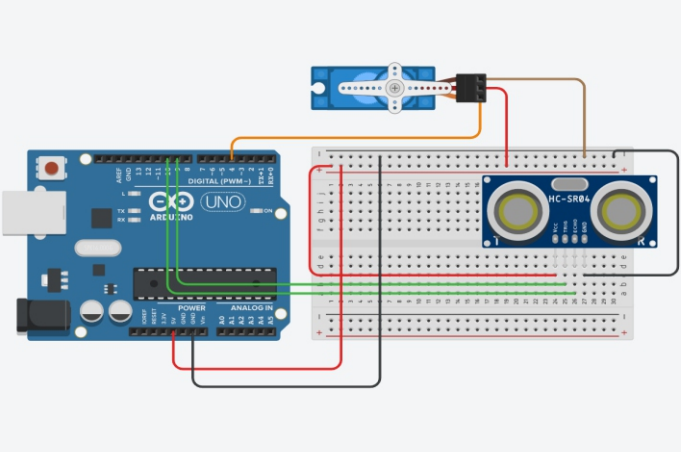
2 Place a breadboard along the Arduino board



3 Take a servo motor and place it next to the Arduino. Connect the Red Wire (5v) to the positive rail and the brown wire(GND) to the negative rail of the breadboard. The Orange wire(Signal) pin is connected to the pin number 4 on the Arduino board.



4 Take an Ultrasonic Sensor (HC-SR04) and place it on the breadboard. Connect the Vcc pin of the sensor to the positive rail of the breadboard. The GND pin is connected to the negative rail of the breadboard. Connect the Echo and Trigger pin of the sensor to the pin 10 and 9 of the Arduino board respectively. Connect the Negative and the Positive rail to the GND and 5V of the Arduino.



5 Upload the code given with this activity in the Arduino IDE.

File Edit Sketch Tools Help

sketch_aug02a§

```
const int trigPin = 9;
const int echoPin = 10;
#include <Servo.h>
long duration;
int distance;
Servo myservo;
void setup() {
myservo.attach(4);
pinMode(13, OUTPUT);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
Serial.begin(9600);
 //myservo.write(0);
}
void loop()
{
```

Note: Now paste a Popsicle stick on the servo motor's horn and throw a ball towards the ultrasonic sensor. If you place it right, the servo motor should hit the ball moving towards the ultrasonic sensor.

CODE

```
const int trigPin = 9;
const int echoPin = 10;
#include <Servo.h>
long duration;
int distance;
Servo myservo;
void setup()
 {
myservo.attach(4);
pinMode(13, OUTPUT);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
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: ");
Serial.println(distance);
if (distance < 20 )
 {
digitalWrite(13, HIGH);
myservo.write(90);
 }
else
 {
digitalWrite(13, LOW);
myservo.write(0);
 }
}
```

Assessment

Tick The Correct One:

Q1. What is the maximum angle to which the servo can rotate?

- ☐ 0
- ☐ 180
- ☐ 90
- ☐ 360

servo.h | trigger pin | myservo.write | echo pin

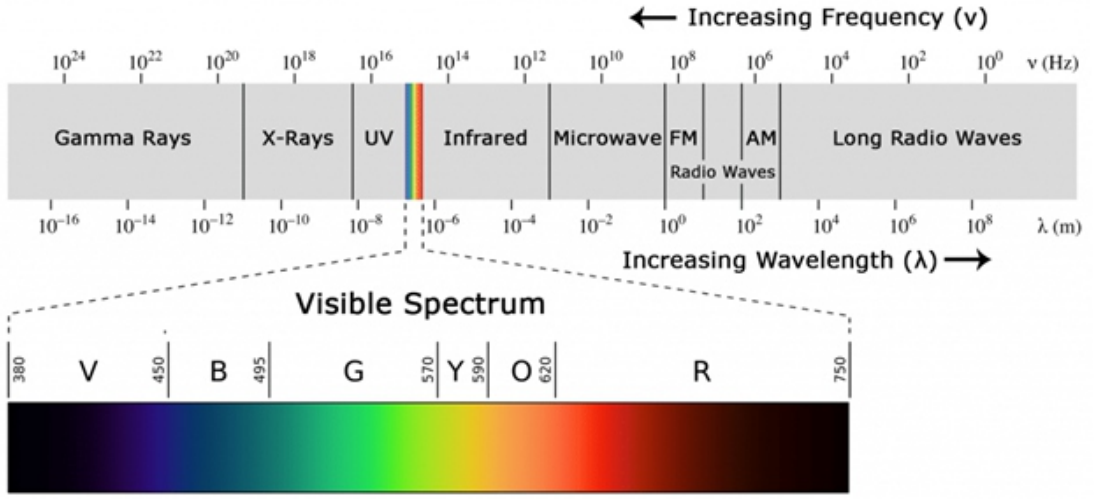
Concept

In this tutorial we'll first explain what infrared is and how it works. Then we'll show you how to set up an IR receiver and remote on an Arduino. You'll also learn how to make a “smart home”, controlling various devices like motors, lights and buzzers with the help of an IR Remote and Arduino

Definitions & Syntax:

What Is Infrared?

Infrared radiation is a form of light similar to the light we see all around us. The only difference between IR light and visible light is the frequency and wavelength. Infrared radiation lies outside the range of visible light, so humans can't see it:



Because IR is a type of light, IR communication requires a direct line of sight from the receiver to the transmitter. It can't transmit through walls or other materials like WiFi or Bluetooth.

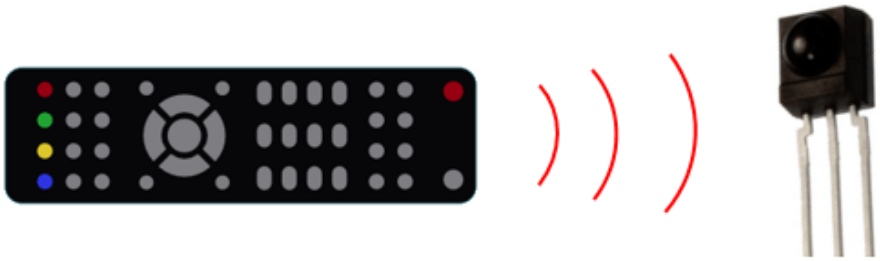
How Ir Remotes And Receivers Work

A typical infrared communication system requires an IR transmitter and an IR receiver. The transmitter looks just like a standard LED, except it produces light in the IR spectrum instead of the visible spectrum. If you have a look at the front of a TV remote, you'll see the IR transmitter LED:

The infrared receiver is the component shown in the figure below. This is the TSOP4838. **Infrared**

Receiver pins:
First pin: Vout
Second pin: GND
Third pin: Vcc

When you press your remote control, it sends infrared modulated signals. These signals contain information that your receiver collects. Each button sends specific information. So, we can assign that information to a specific button.

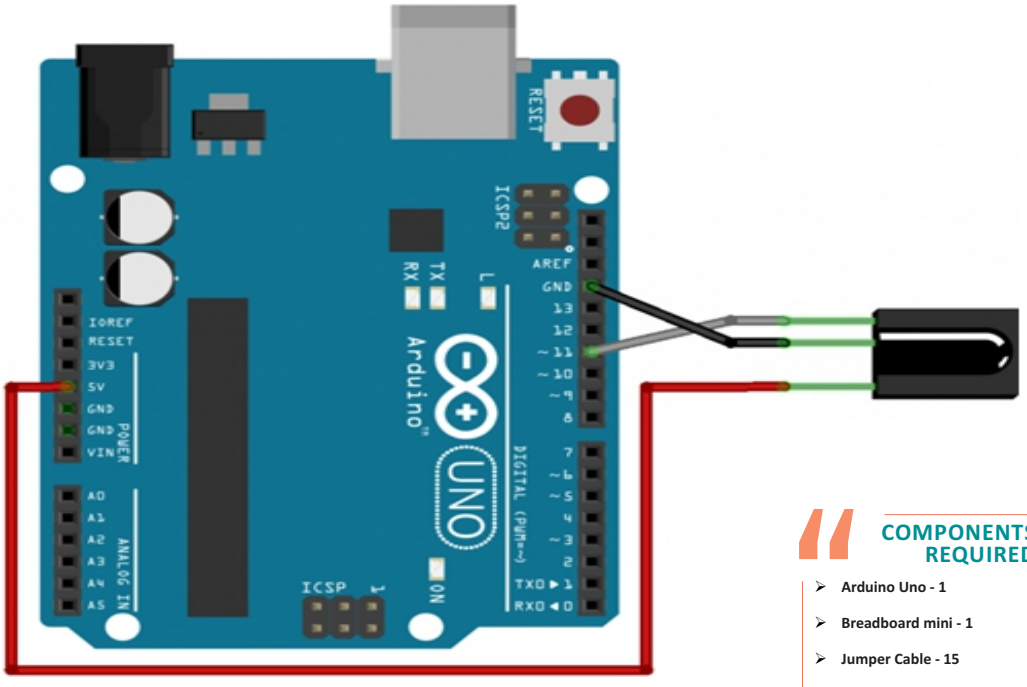


Decode the IR signals

In this part of the project you need to decode the IR signals associated with each button.

Schematics

Connect the IR receiver accordingly to the schematics below.

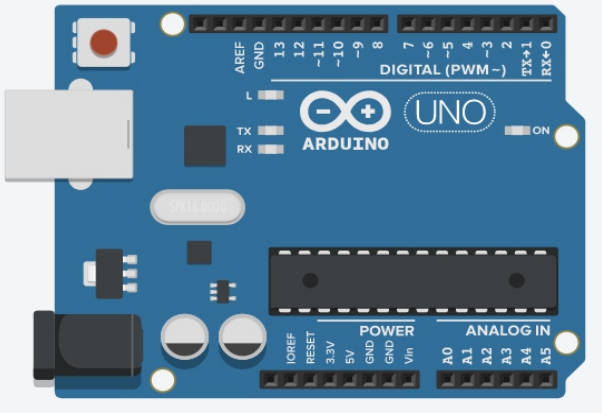


COMPONENTS REQUIRED

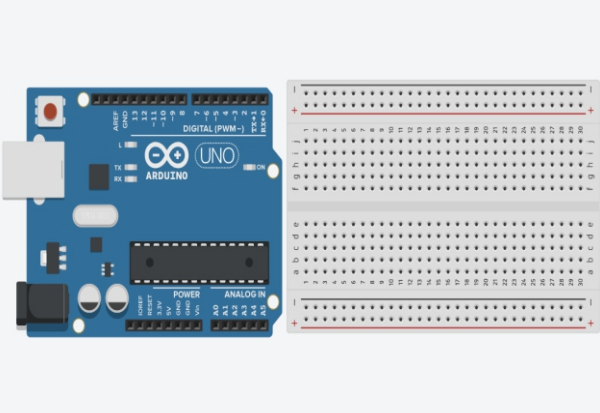
- Arduino Uno - 1
- Breadboard mini - 1
- Jumper Cable - 15
- LED - 1 and Buzzer - 1
- IR Remote -1 & TSOP - 1
- Servo Motor - 1

CHAPTER - 5
IR REMOTE CONTROL

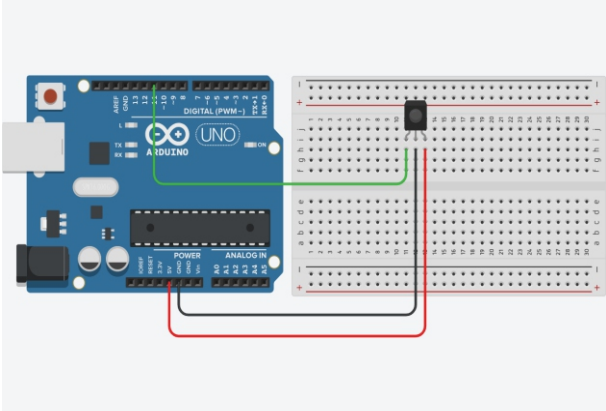
1 Take an Arduino Board



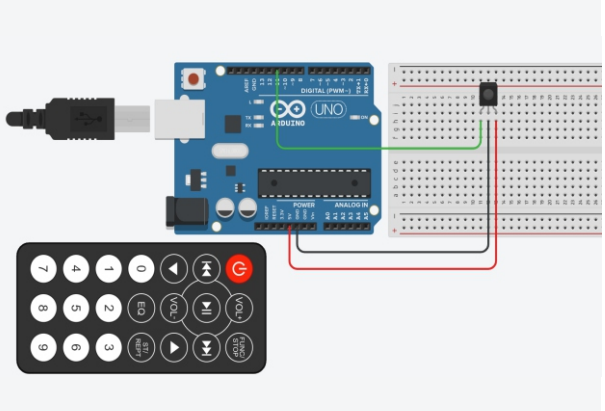
2 Place a breadboard along the Arduino board



3 Place a TSOP IR receiver on the breadboard and connect it to the Arduino as shown.



4 Place the remote close to the IR receiver.



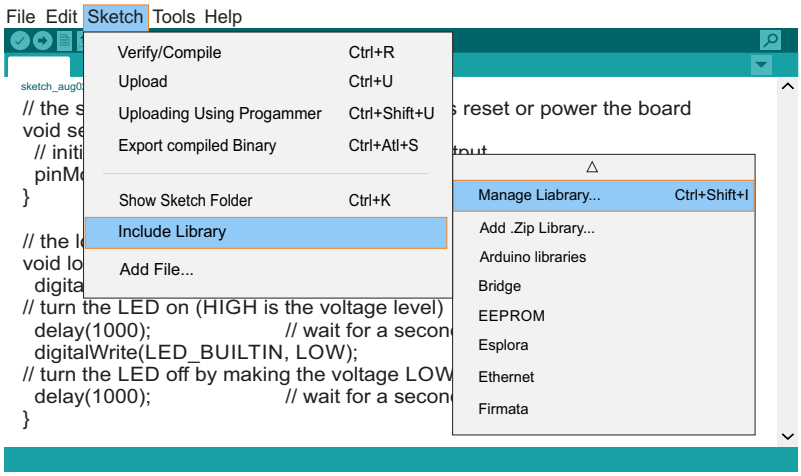
5 Open the Arduino IDE software and write the code given with the activity.

File Edit Sketch Tools Help

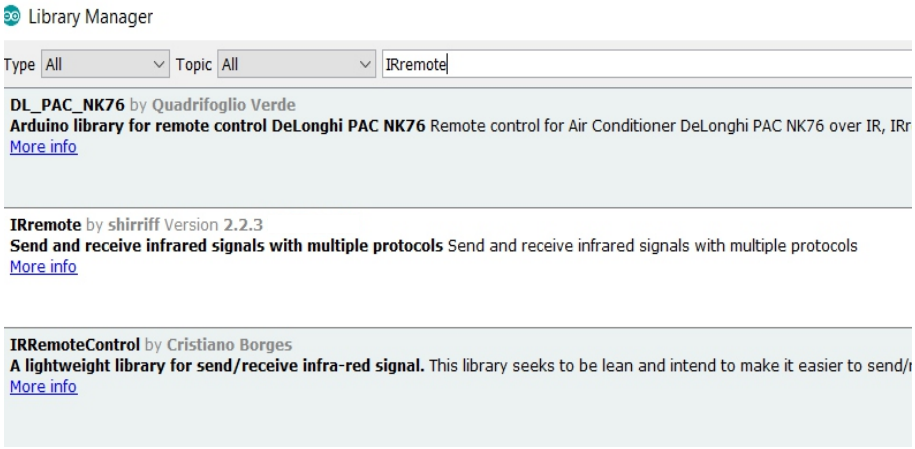
```
#include <IRremote.h>
int RECV_PIN = 11;
IRrecv irrecv(RECV_PIN);
decode_results results;
void setup() {
  Serial.begin(9600);
  irrecv.enableIRIn();                    // Start the receiver
}
void loop() {
  if (irrecv.decode(&results)) {
    Serial.println(results.value, HEX);
    irrecv.resume();                      // Receive the next value
  }
  delay(100);
}
```

Note: Before Uploading the code, you would need to download the IRremote libray. Click on “Sketch” option on the IDE.

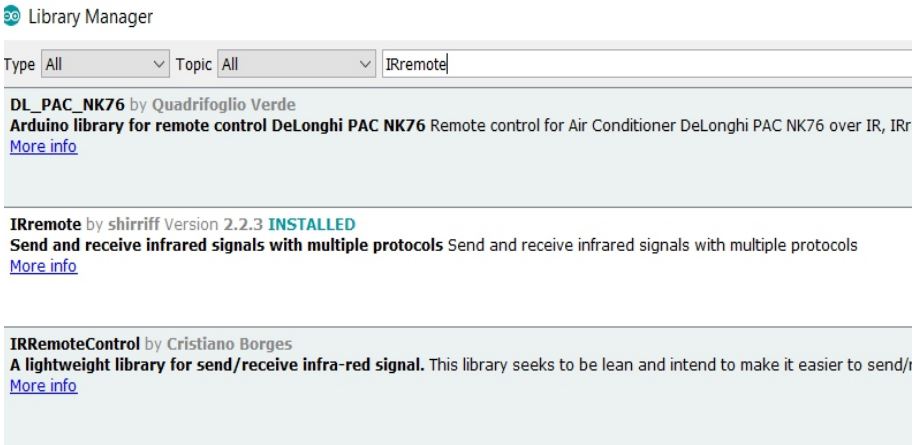
Now scroll down to the “include library” option and keep your mouse pointer there. From the new drop down menu, click on “Manage Libraries” .



Inside the library manager, in the search menu, search for IRremote as shown.



Select the IRremote by shirriff and click on the Install button. When it's done, you will see INSTALLED written next to the library.



Upload the code and then open up the Serial Monitor and press random buttons on the remote. See the code for the buttons being displayed on the screen.

Code library

To control the IR receiver, you need to install the IRremote Library in the Arduino IDE.

Installing the IRremote library

Click here to download the IRremote library

- You should have a .zip folder in your Downloads
- Unzip the .zip folder and you should get **IRremote-master** folder
- Rename your folder from **IRremote-master** to **IRremote**
- Move the **IRremote** folder to your Arduino IDE installation libraries folder
- Finally, re-open your Arduino IDE

Copy the following code to your Arduino IDE, and upload it to your Arduino board.
Make sure that you have the right board and COM port selected.

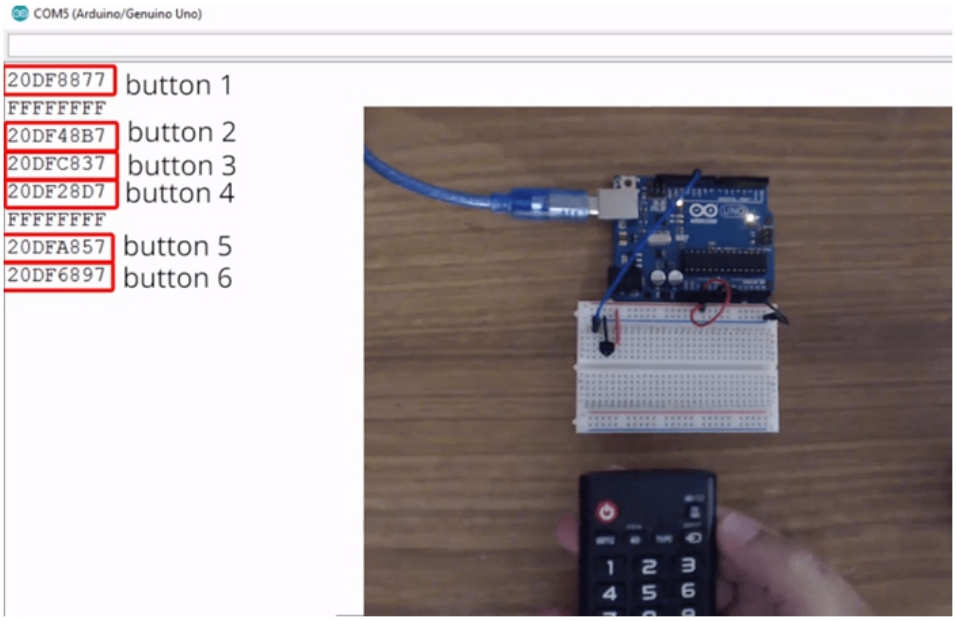
CODE

```
#include <IRremote.h>
int RECV_PIN = 11;
IRrecv irrecv(RECV_PIN);
decode_results results;
void setup() {
  Serial.begin(9600);
  irrecv.enableIRIn();                    // Start the receiver
}
void loop() {
  if (irrecv.decode(&results)) {
    Serial.println(results.value, HEX);
    irrecv.resume();                      // Receive the next value
  }
  delay(100);
}
```

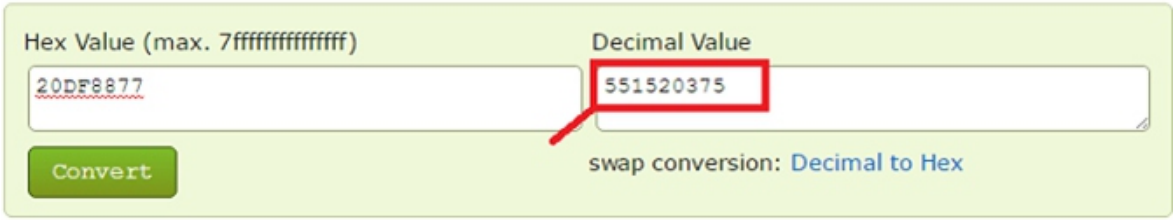
Open the serial monitor at a baud rate of 9600.

In this project you want to control 3 devices. Choose 6 buttons for the following tasks:
1.device1 – ON
2.device1 – OFF
3.device2 – ON
4.device2 – OFF
5.device3 – ON
6.device3 – OFF
Press, for example, the button number 1 of your remote control. You should see a code on the serial monitor. Press the same button several times to make sure you have the right code for that button. If you see something like FFFFFFFF ignore it, it's trash.

Do the same for the other buttons.
Write down the code associated with each button, because you'll need that information later.

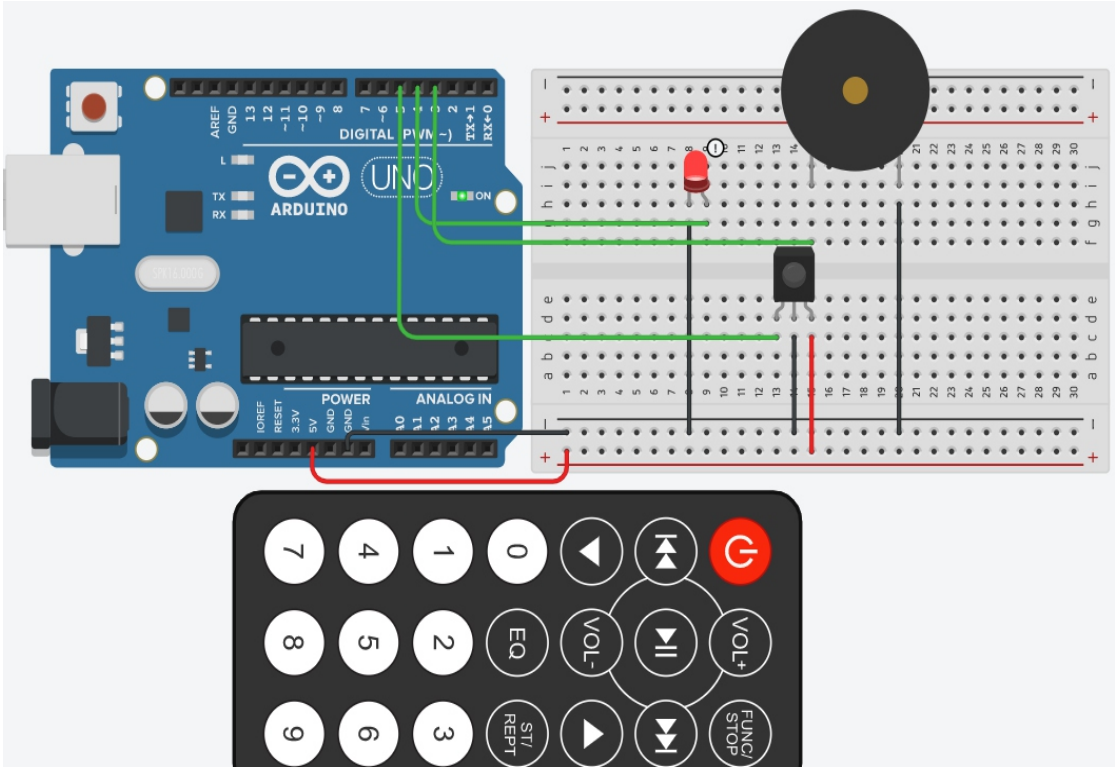


Code
Now, grab the codes you captured in the previous step. You need to convert your codes from hex to decimal.
For that, you can use the following website:
www.binaryhexconverter.com/hex-to-decimal-converter
Here's a conversion example for one of my codes:

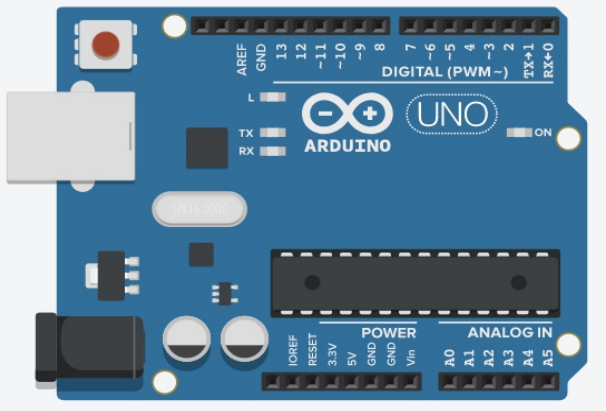


Repeat that process to all your hex values and save the **decimal** values. These are the ones you need to replace in the code below.
For more information you can refer to the following link:
https://randomnerdtutorials.com/arduino-ir-remote-control/

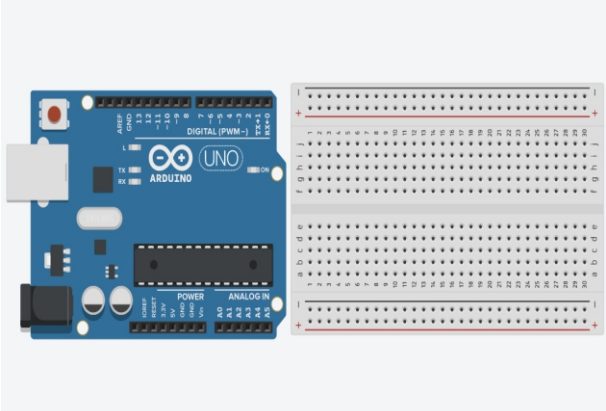
Activity:- Ir Controlled LED, Toy Motor And Buzzer



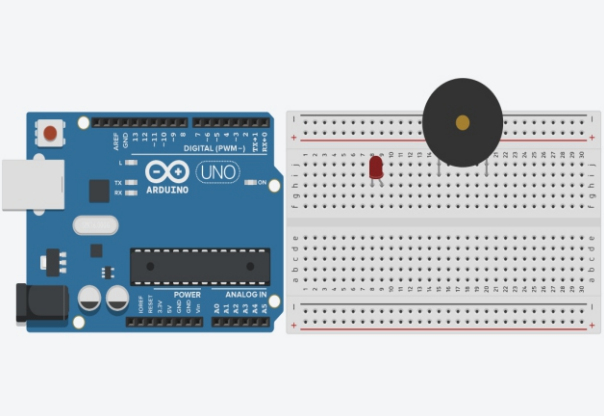
1 Take an Arduino Board



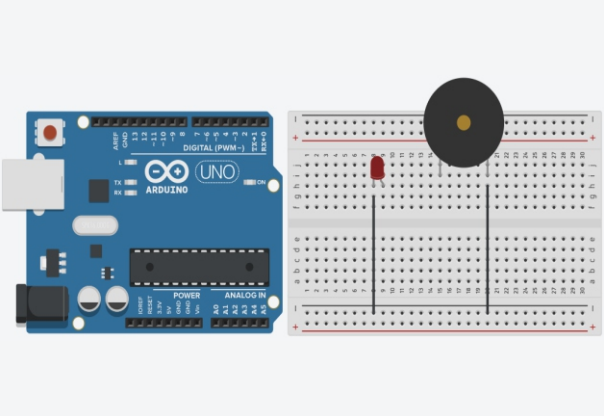
2 Place a breadboard along the Arduino board



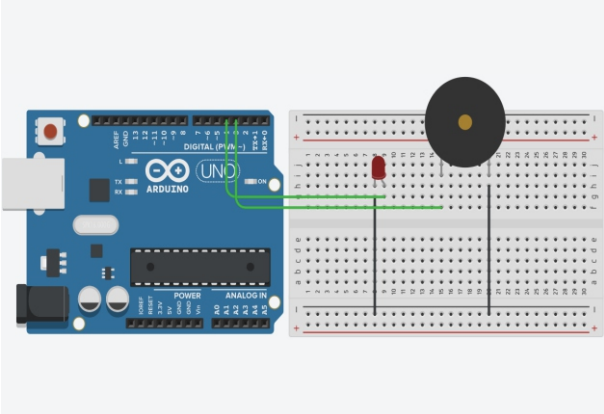
3 Connect the positive leg (longer) of the LED and the buzzer to Arduino's pin number 13 and 12 respectively.



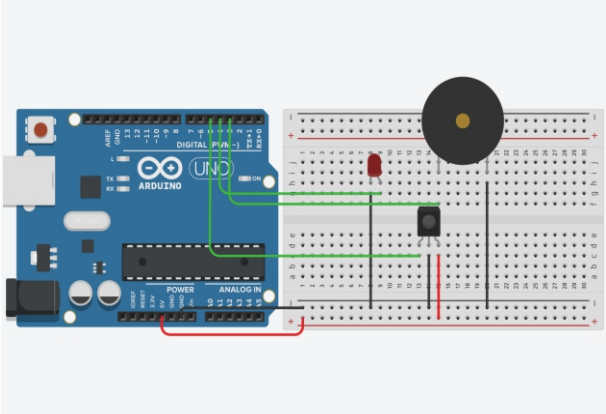
4 Place a keypad near the breadboard.



5 Connect the keypad to the Arduino as shown.



6 Open the Arduino IDE and write the code given with the activity.

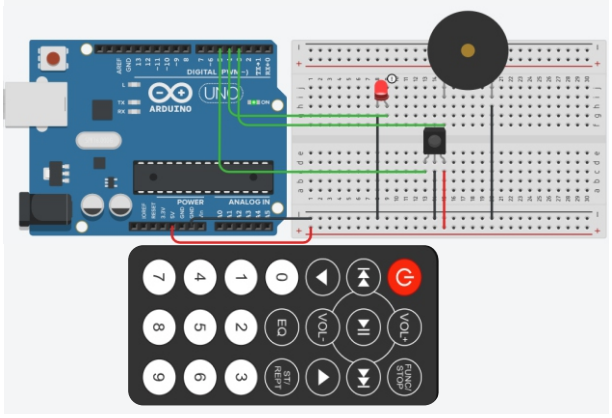


7 Open the Arduino IDE software and write the code given with the activity.

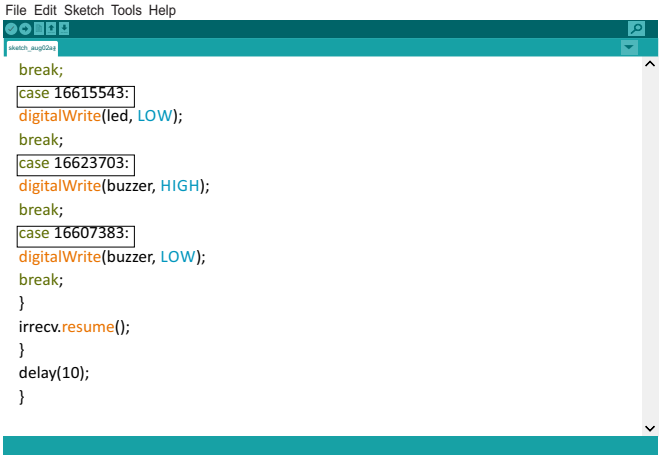
File Edit Sketch Tools Help

```
#include<Servo.h>
#include <IRremote.h>
int IR_Recv = 5;
int led = 4;
int buzzer = 2;
IRrecv irrecv(IR_Recv);
decode_results results;
void setup() {
  Serial.begin(9600);
  irrecv.enableIRIn();
  pinMode(led, OUTPUT);
  pinMode(buzzer, OUTPUT);
 }
void loop() {
  if (irrecv.decode(&results))
 {
```

8 Now upload the code and see the values of the button pressed on the remote. Press the buttons to see which buttons controls the devices.



Note: If you are using a different remote, you can always change the values of the button codes in the Arduino code at the marked place.



CODE

```
#include<Servo.h>
#include <IRremote.h>
Servo myservo;
int IR_Recv = 5;
int led = 4;
int buzzer = 2;
int pos = 0;
IRrecv irrecv(IR_Recv);
decode_results results;
void setup()
 {
Serial.begin(9600);
irrecv.enableIRIn();
pinMode(led, OUTPUT);
pinMode(buzzer, OUTPUT);
}
void loop()
 {
if (irrecv.decode(&results))
{
long int decCode = results.value;
Serial.println(results.value);
switch (results.value)
```

```
{
case 16582903:
digitalWrite(led, HIGH);
break;
case 16615543:
digitalWrite(led, LOW);
break;
case 16623703:
digitalWrite(buzzer, HIGH);
break;
case 16607383:
digitalWrite(buzzer, LOW);
break;
}
irrecv.resume();
}
delay(10);
}
```

Assessment

Tick The Correct One:

Q1. Why IR isn't visible to human eyes?

- ☐ Because lies outside the range of visible light
- ☐ Because its frequency is more
- ☐ Because its wavelength is more
- ☐ None of the above

frequency wavelength IR rays IRemote.h

Concept

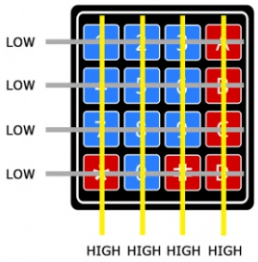
In this activity, you will learn how to setup a keypad on the Arduino. Then you will make a Keypad Security System with password. If the correct pin is entered the LED will glow up otherwise the Buzzer will sound.

What is a Keypad?

A keypad is a set of buttons or keys bearing digits, symbols and/or alphabetical letters placed in order on a pad, which can be used as an efficient input device. A keypad may be purely numeric, as that found on a calculator or a digital door lock, or alphanumeric as those used on cellular phones. The Arduino detects which button is pressed by detecting the row and column pin that's connected to the button.

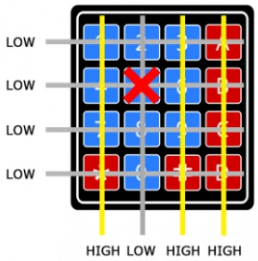
Step:- 1

First, when no buttons are pressed, all of the column pins are held HIGH, and all of the row pins are held LOW:



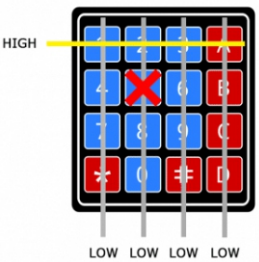
Step:- 2

When a button is pressed, the column pin is pulled LOW since the current from the HIGH column flows to the LOW row pin:



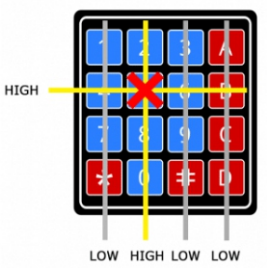
Step:- 3

The Arduino now knows which column the button is in, so now it just needs to find the row the button is in. It does this by switching each one of the row pins HIGH, and at the same time reading all of the column pins to detect which column pin returns to HIGH:



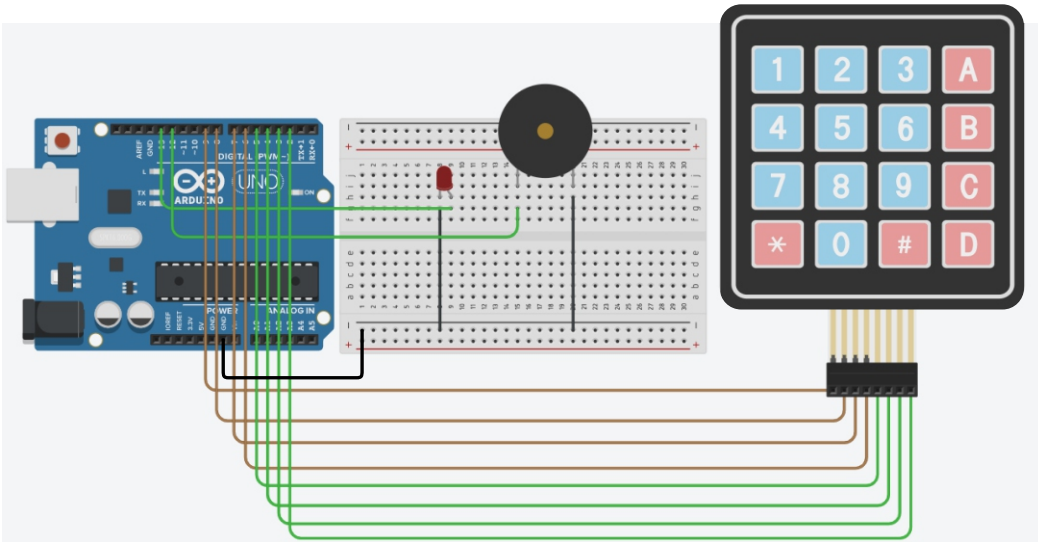
Step:- 4

When the column pin goes HIGH again, the Arduino has found the row pin that is connected to the button:

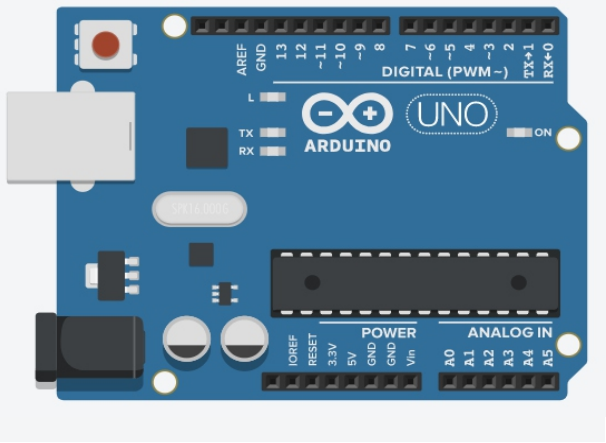


From the diagram above, you can see that the combination of row 2 and column 2 could only mean that the number 5 button was pressed.

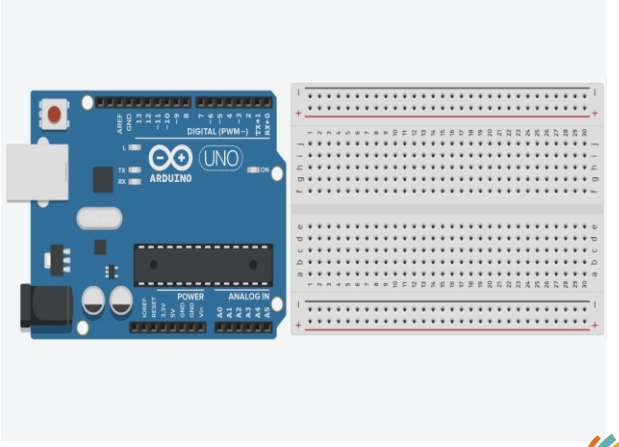
Activity:- Keypad Security System



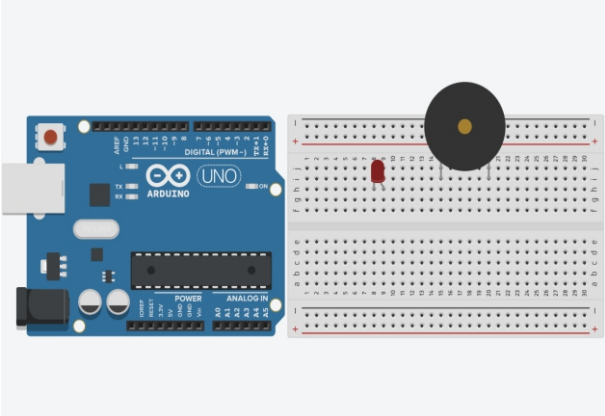
1 Take an Arduino Board



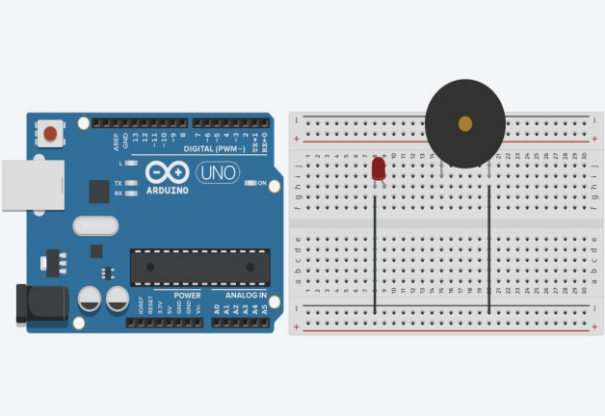
2 Place a breadboard along the Arduino board



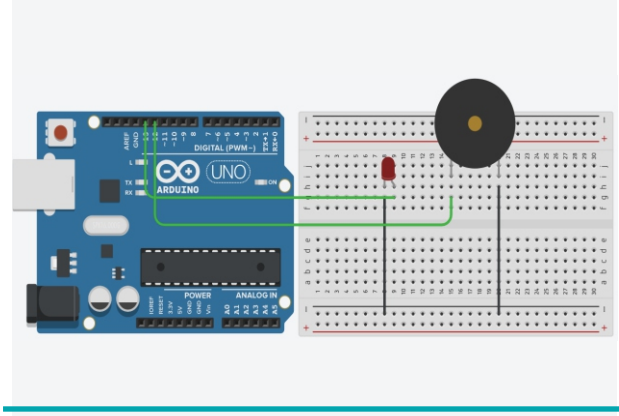
3 Place a LED and a buzzer on the breadboard.



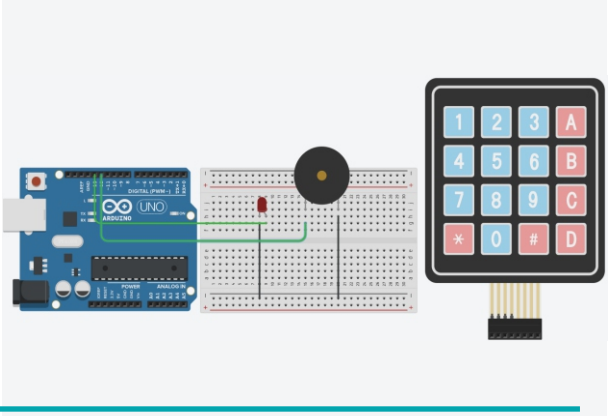
4 Connect the shorter leg of the LED and the shorter leg of the buzzer to the negative rail of the bread board.



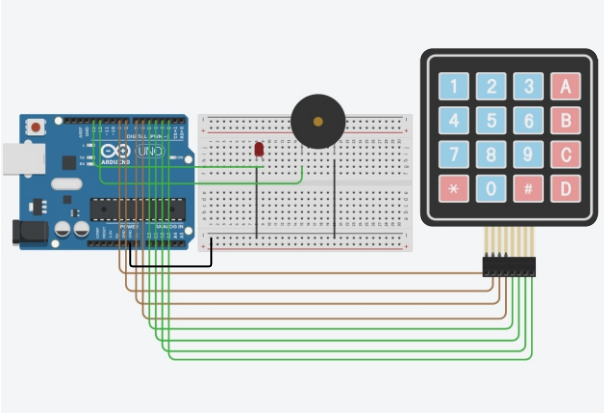
5 Connect the positive leg (longer) of the LED and the buzzer to Arduino's pin number 13 and 12 respectively.



6 Place a keypad near the breadboard.



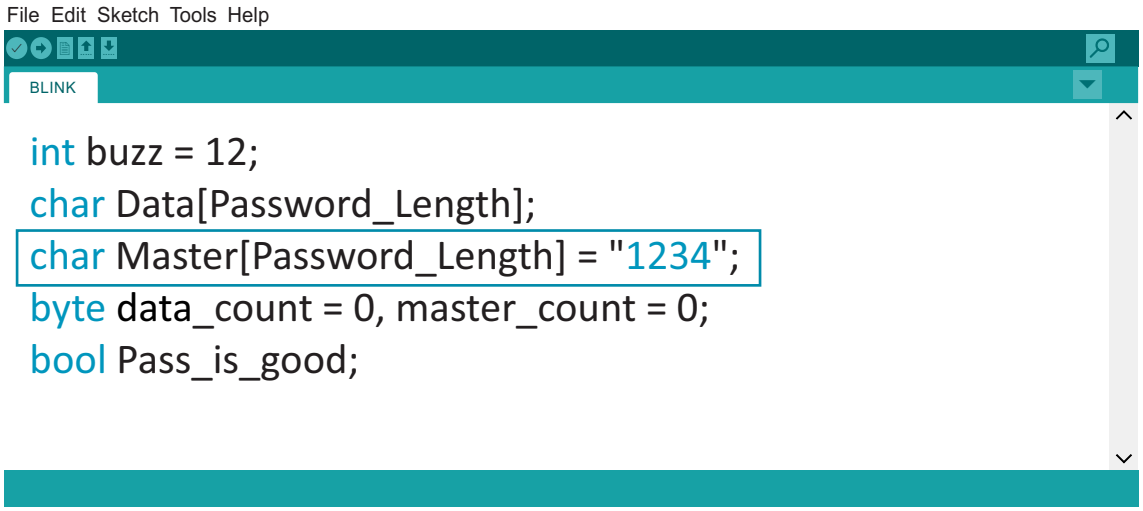
7 Connect the keypad to the Arduino as shown.



8 Open the Arduino IDE and write the code given with the activity.

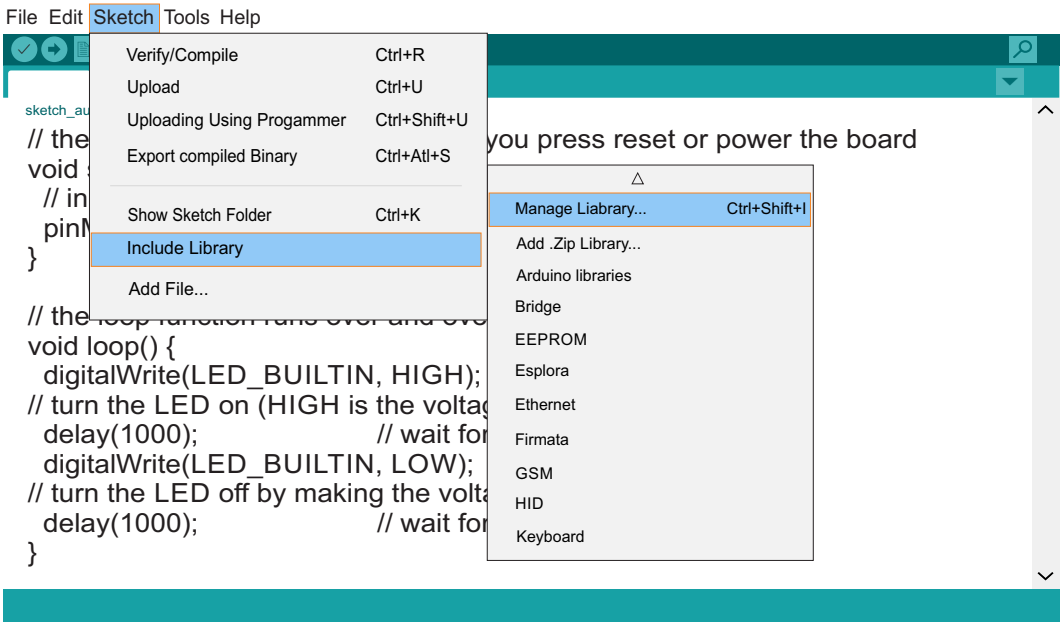
```
#include <Keypad.h>
#define Password_Length 5
int signalPin = 13;
int buzz = 12;
char Data[Password_Length];
char Master[Password_Length] = "1234";
byte data_count = 0, master_count = 0;
bool Pass_is_good;
char customKey;

const byte ROWS = 4;
const byte COLS = 4;
char hexaKeys[ROWS][COLS] = {
```

9 Look for a variable called “master”. This is the variable which stores the password. Right now it says “1234”, you can change it to a password of your own choosing.

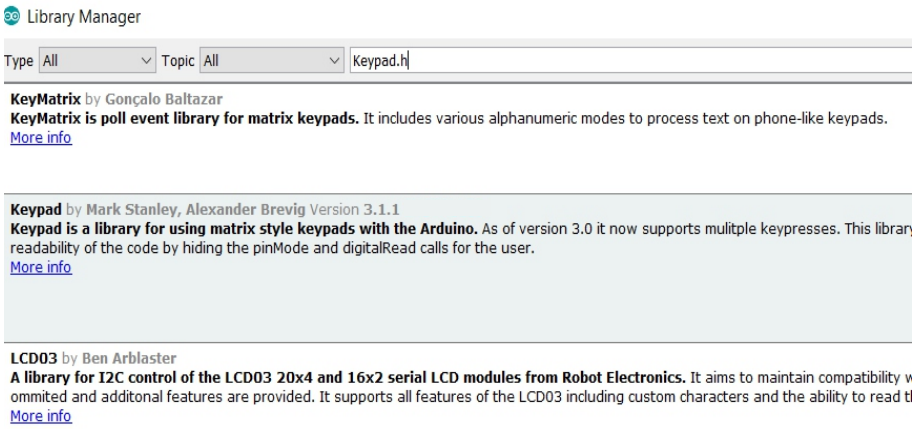


Note: Enter the code using the keypad. If the password is right, the led would glow up, else a buzzer would sound.

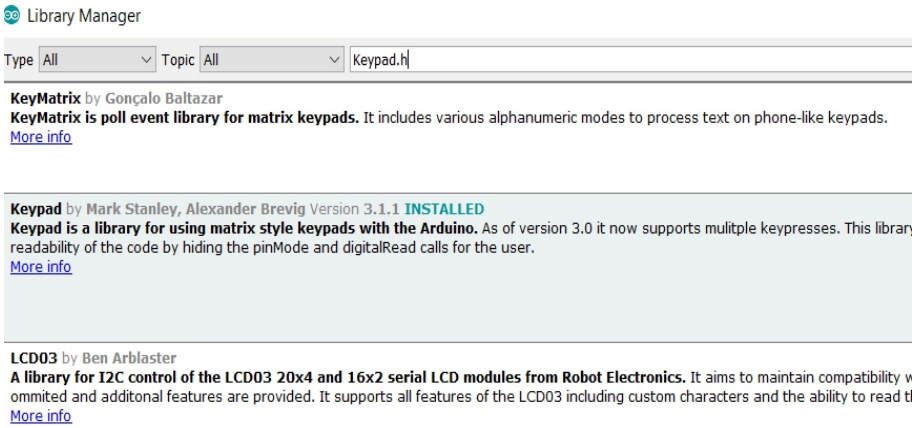
Now scroll down to the “include library” option and keep your mouse pointer there. From the new drop down menu, click on “Manage Libraries” .



Inside the library manager, in the search menu, search for IRremote as shown.



Select the IRremote by shirriff and click on the Install button. When it's done, you will see INSTALLED written next to the library.



Upload the code and then open up the Serial Monitor and press random buttons on the remote. See the code for the buttons being displayed on the screen.

COMPONENTS REQUIRED

- Arduino Uno - 1
- Breadboard mini - 1
- Servo Motor - 1
- Buzzer - 1
- Keypad - 1
- Plastic Box - 1
- Jumper Cable - 8

CODE

```
#include <Keypad.h>
#define Password_Length 5
int signalPin = 13;
int buzz = 12;
char Data[Password_Length];
char Master[Password_Length] = "1234";
byte data_count = 0, master_count = 0;
bool Pass_is_good;
char customKey;
const byte ROWS = 4;
const byte COLS = 4;
char hexaKeys[ROWS][COLS] =
{
  {'1', '2', '3', 'A'},
 {'4', '5', '6', 'B'},
 {'7', '8', '9', 'C'},
 {'*', '0', '#', 'D'}
};
byte rowPins[ROWS] = {9, 8, 7, 6};
byte colPins[COLS] = {5, 4, 3, 2};
Keypad customKeypad = Keypad(makeKeymap(hexaKeys),
rowPins, colPins, ROWS, COLS);
void setup()
{
 Serial.begin(9600);
pinMode(signalPin, OUTPUT);
pinMode(buzz, OUTPUT);
Serial.println("Enter Password: ");
}
void loop() {
customKey = customKeypad.getKey();
if (customKey)
{
Data[data_count] = customKey;
Serial.println(Data[data_count]);
data_count++;   }
if (data_count == Password_Length - 1)
{
if (!strcmp(Data, Master))
{
```

```
 Serial.println("ACCESS GRANTED");
digitalWrite(signalPin, HIGH);
delay(5000);
digitalWrite(signalPin, LOW);
}
else
{
Serial.println("ACCESS DENIED");
digitalWrite(buzz, HIGH);
delay(5000);
digitalWrite(buzz,LOW);
 }
clearData();
 }
 }
voidclearData() {
while (data_count != 0)
{
Data[data_count--] = 0;
}
return;
}
```

Assessment

Tick The Correct One:

Q1. What is the maximum length of the password you can insert?

- ☐ 10
- ☐ 255
- ☐ 1023
- ☐ None of the above

Keypad Makekeymap customkeypad


ARDUINO