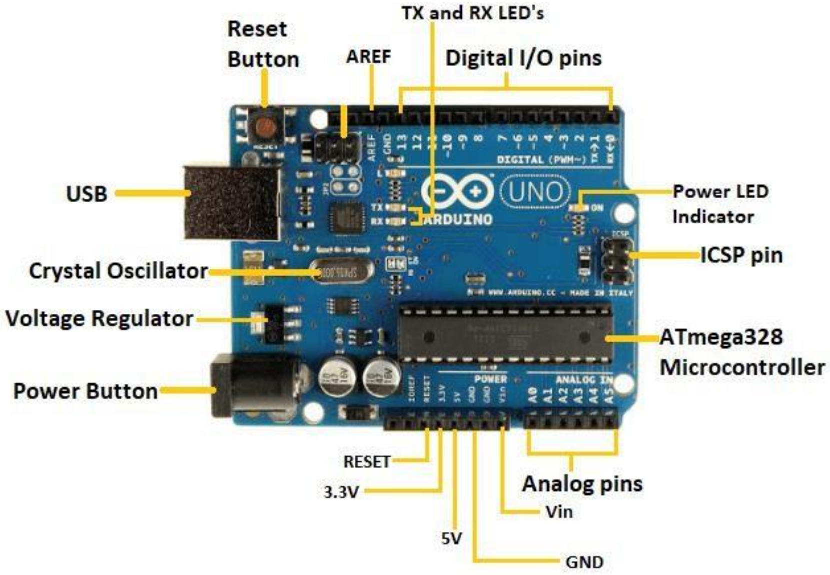
Arduino

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduinoboards 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 Arduinoprogramminglanguage(basedon Wiring),and theArduinoSoftware(IDE),basedon Processing.



ArduinoUnoBoard

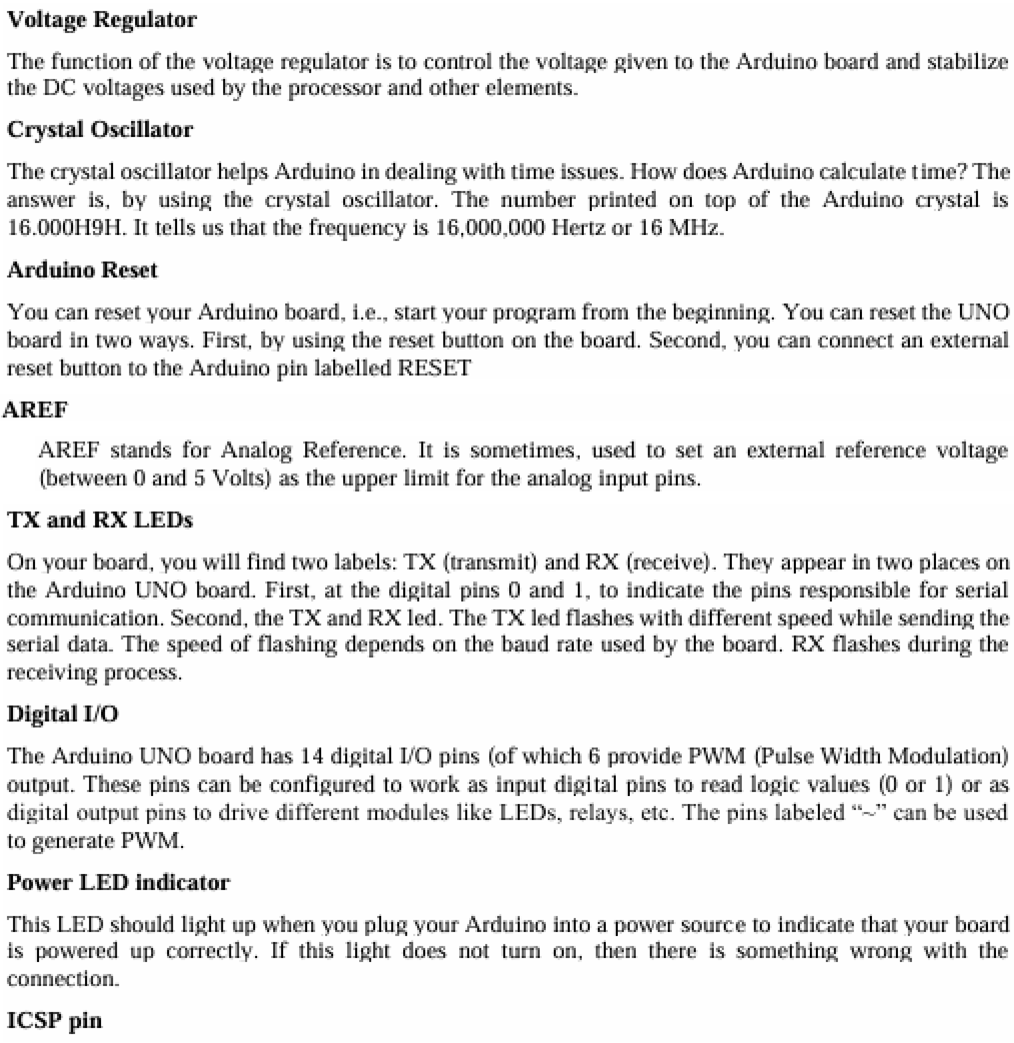
The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac,Windows,andLinux.Teachersandstudentsuseittobuildlowcost scientificinstruments,toprove chemistry and physics principles, or to get started with programming and robotics. Arduino also simplifies the process of working with microcontrollers.

PowerUSB

ArduinoboardcanbepoweredbyusingtheUSBcablefromyourcomputer.Allyouneedtodois connect the USB cable to the USB connection

Powerbutton

Arduino boards can be powered directly from the AC mains power supply byconnecting it to the Jack of Power button.



Voltage Regulator

The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.

Crystal Oscillator

The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.

Arduino Reset

You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET

AREF

AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

TX and RX LEDs

On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led. The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.

Digital I/O

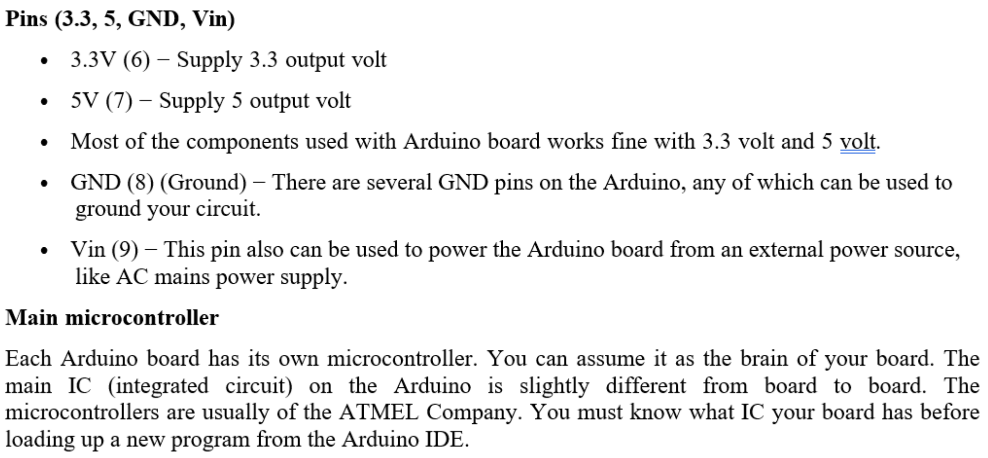
The Arduino UNO board has 14 digital I/O pins (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled "~" can be used to generate PWM.

Power LED indicator

This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.

ICSP pin

It is used to program the microcontroller.



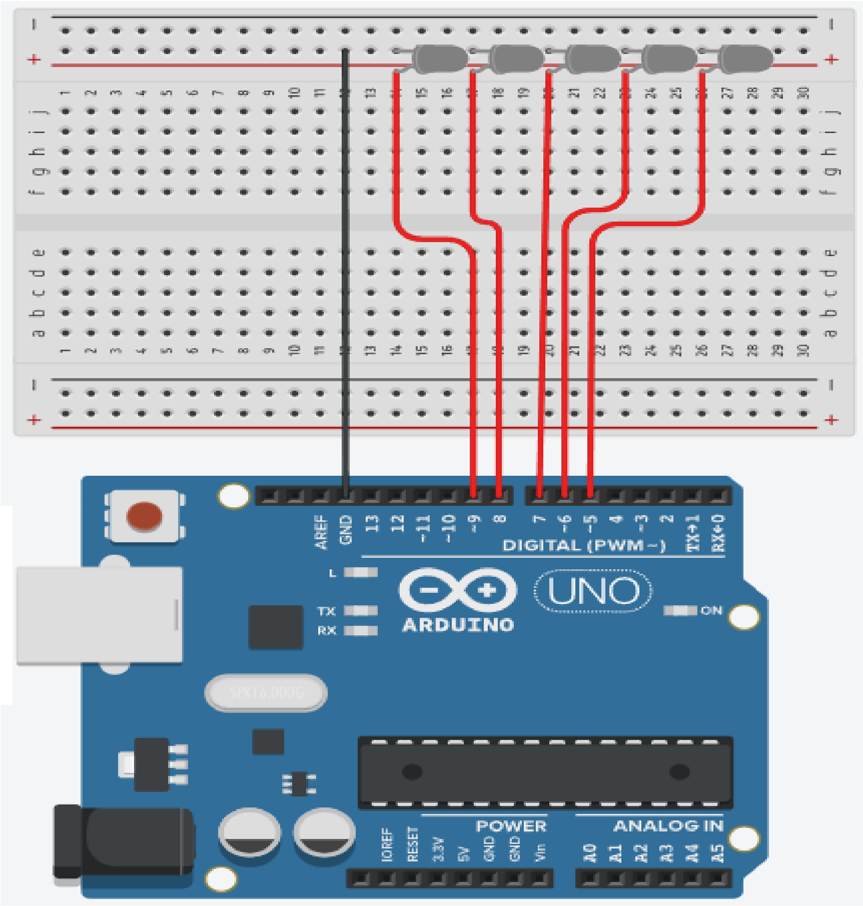
Pins (3.3, 5, GND, Vin)

- 3.3V (6) − Supply 3.3 output volt
- 5V (7) − Supply 5 output volt
- Most of the components used with Arduino board works fine with 3.3 volt and 5 volt.
- GND (8) (Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit.
- Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.

Main microcontroller

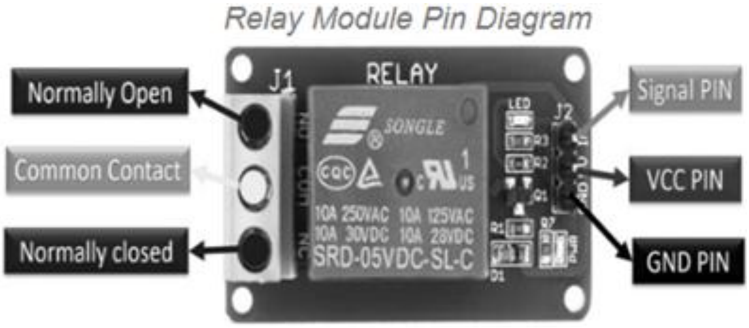
Each Arduino board has its own microcontroller. You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE.

1: Develop a program to blink 5 LEDs back and forth.



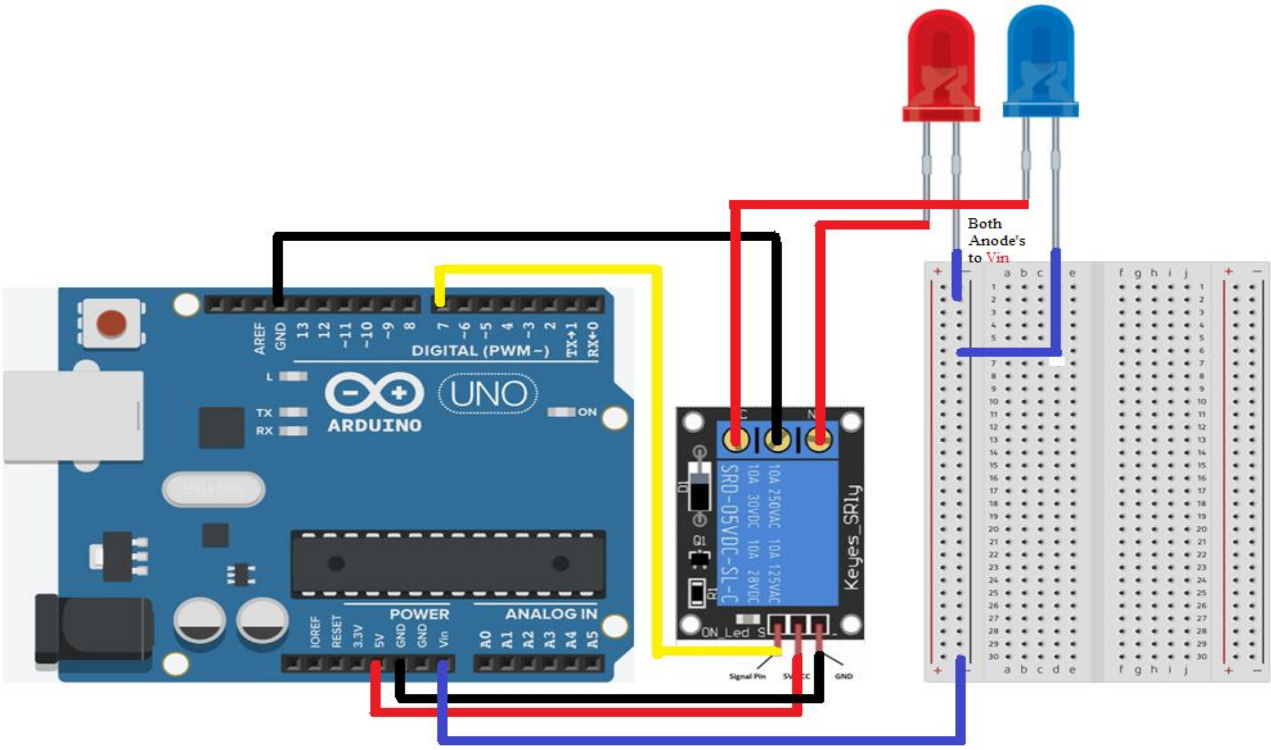
```
int LED1 = 9;
int LED2 = 8;
int LED3 = 7;
int LED4 = 6;
int LED5 = 5;

void setup(){

pinMode(LED1 , OUTPUT); //PIN#9
pinMode(LED2, OUTPUT); //PIN#8
pinMode(LED3,  OUTPUT); //PIN#7
pinMode(LED4,  OUTPUT); //PIN#6
pinMode(LED5,  OUTPUT); //PIN#5
}
// CYCLE TO COMPLETE LED-1 to LED-5 BACK
void loop(){
digitalWrite(LED1 , HIGH);
 delay(3000);
digitalWrite(LED1 ,  LOW);
delay(1000);

digitalWrite(LED2, HIGH);
 delay(1000);
digitalWrite(LED2,  LOW);
 delay(1000);

digitalWrite(LED3, HIGH);
 delay(1000);
digitalWrite(LED3,  LOW);
delay(1000);

digitalWrite(LED4, HIGH);
 delay(1000);
digitalWrite(LED4,  LOW);
delay(1000);

digitalWrite(LED5, HIGH);
 delay(5000);
digitalWrite(LED5,  LOW);
delay(5000);
 // CYCLE TO COMPLETE LED-5 to LED-1 FORTH
digitalWrite(LED5 , HIGH);
 delay(3000);
digitalWrite(LED5 ,  LOW);
delay(1000);

digitalWrite(LED4, HIGH);
 delay(1000);
digitalWrite(LED4,  LOW);
 delay(1000);

digitalWrite(LED3, HIGH);
 delay(1000);
digitalWrite(LED3,  LOW);
delay(1000);

digitalWrite(LED2, HIGH);
 delay(1000);
digitalWrite(LED2,  LOW);
delay(1000);

digitalWrite(LED1, HIGH);
 delay(5000);
digitalWrite(LED1,  LOW);
delay(5000);
}
```

2: Develop a program to interface a relay with Arduino board.

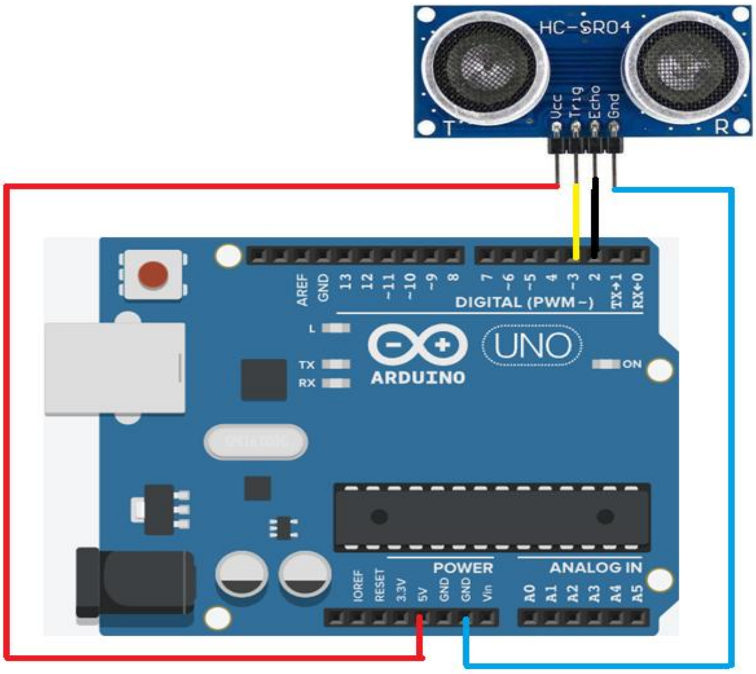


```
const int relayPin=7;
void setup() {
pinMode(relayPin,OUTPUT);
digitalWrite(relayPin,LOW);
}

void loop() {
digitalWrite(relayPin,HIGH);
delay(3000);
digitalWrite(relayPin,LOW);
delay(3000);
}
```

How the connections to be made from relay to Arduino Board and LED's		
Relay: VCC, GND, IN VCC-5V GND-GND IN-7 (as per code)	**Opposite portion of the Relay** (printed in Chinese language) COM (Centre) GND Either side NO, NC NO, NC Cathode of both the LEDS	**LED'S connecton to Arduino** Both Anode's of LED's - Vin (Above the Ao pin) Cathode of individual LED's to NO & NC of Relay



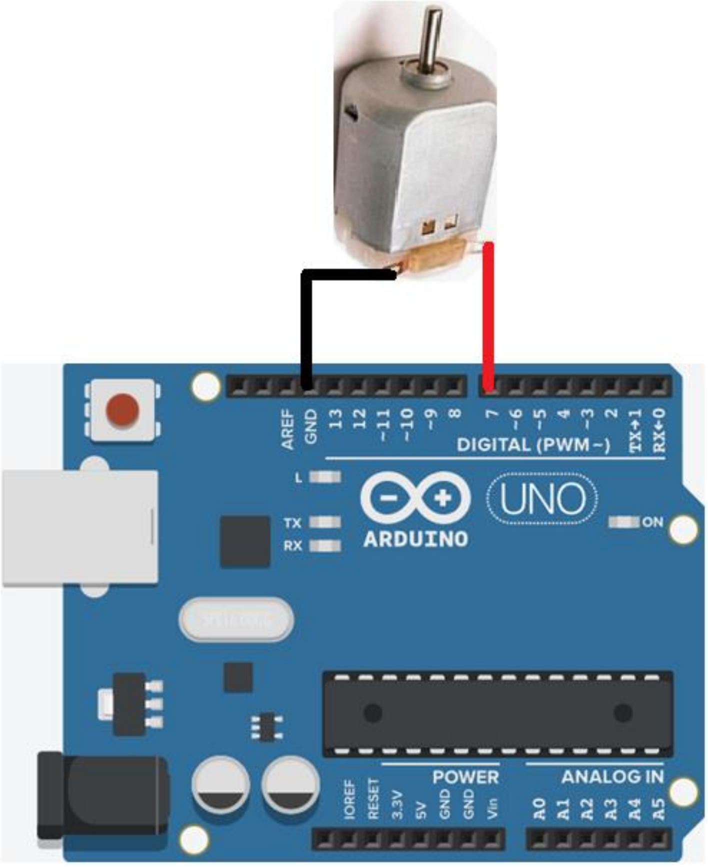
3: Develop a program to deploy an intrusion detection system using Ultrasonic and sound sensors.



```
#define echoPin 2
#define trigPin 3
#define max 50 // maximum sensing range is 50 to 70 cms
long duration;
int distance;
void setup()
{
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 * 0.0344 / 2;
    if(distance < max)
   {
Serial.print(" Instrusion detected, the distance of the instrusion object is : ");
Serial.print(distance);
Serial.println(" cm");
    delay(100);
   }
   else{
Serial.println(" Instrusion not detected. Beyond the sensing range  ");
```

```
    delay(100);
   }
}
```

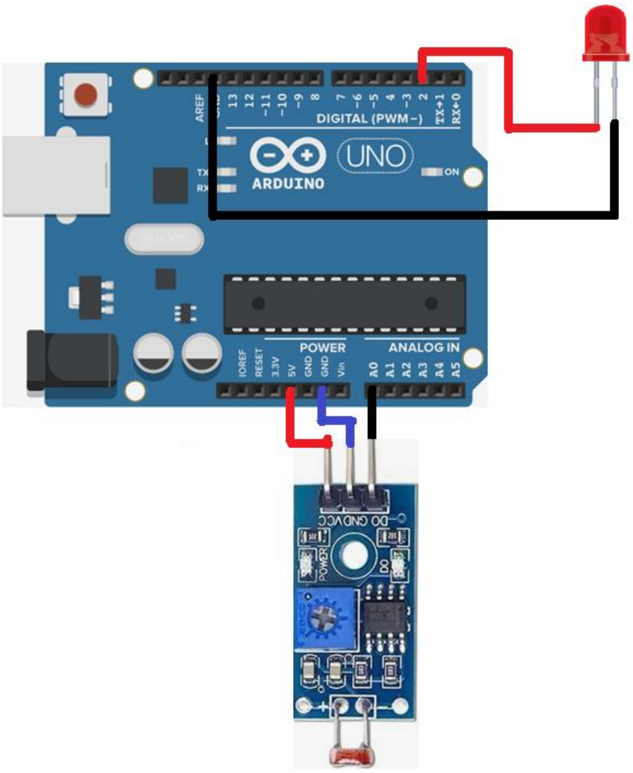
4 Develop a program to control a DC motor with Arduino board.



```
const int DCmotor=7;
void setup() {
  // put your setup code here, to run once:
pinMode(DCmotor,OUTPUT);
digitalWrite(DCmotor,LOW);
}

void loop() {
  // put your main code here, to run repeatedly:
digitalWrite(DCmotor,HIGH);
delay(10000);
digitalWrite(DCmotor,LOW);
delay(3000);
}
```

<table>
<tr><th colspan="2">OUTPUT: NOTE to be taken care of the Motor & Fan before dumping the code to the Arduino</th></tr>
<tr><td><b>Note</b> before running hold the motor in your hand. Please do not keep it on the table as it may damage the fan.</td><td>To change the direction of rotation of the fan, you need to reverse the pins .</td></tr>
</table>

5: Develop a program to deploy smart street light system using LDR sensor.



```
// Used a NTC(Negative Temperature Coefficient) in this experiment.
// Connect DO of LDR to A0.
const int ledPin = 2;
const int ldrPin = A0;
#define threshold 200
void setup()
{
Serial.begin(9600);
pinMode(ledPin, OUTPUT);
pinMode(ldrPin, INPUT);
}
void loop()
{
int ldrStatus = analogRead(ldrPin);
 if (ldrStatus>= threshold)
{
digitalWrite(ledPin, HIGH);
Serial.print("Its DARK, Turn on the Street light : ");
Serial.println(ldrStatus);
}
else
{
digitalWrite(ledPin, LOW);
Serial.print("Its BRIGHT, Turn off the Street light : ");
Serial.println(ldrStatus);
}
}
```

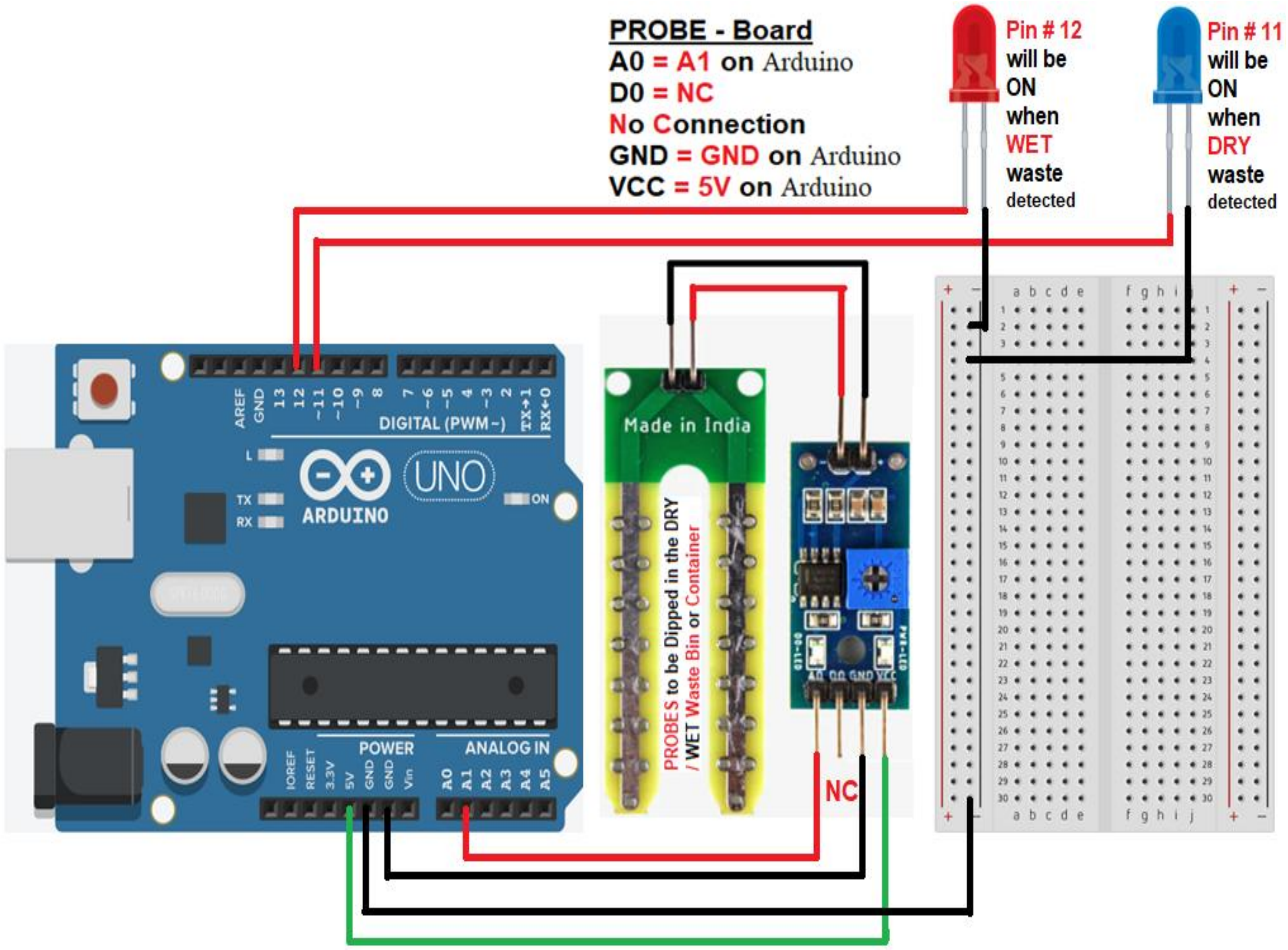
6. Develop a program to classify dry and wet waste with the Moisture sensor (DHT22).

```
const int sensor_pin = A1;
int dry_waste=11;
int wet_waste=12;
int threshold=10;
void setup() {
 Serial.begin(9600);
}

void loop() {
 float moisture_percentage;
 int sensor_analog;
 sensor_analog = analogRead(sensor_pin);
 moisture_percentage = ( 100 - ( (sensor_analog/1023.00) * 100 ) );
 Serial.print("Moisture Percentage = ");
 Serial.print(moisture_percentage);
 Serial.print("%\n\n");
 if (moisture_percentage>threshold)
 {
  Serial.println("Wet waste detected");
  digitalWrite(dry_waste,LOW);
  digitalWrite(wet_waste,HIGH);
  delay(1000);
 }
 else {
    Serial.println("Dry waste detected");
     digitalWrite(wet_waste,LOW);
     digitalWrite(dry_waste,HIGH);
     delay(1000);

 }
 delay(1000);
}
```

Using DISCRETE / PHYSICAL COMPONENTS

Sl #	NAME OF THE COMPONENTS	SPECIFICATIONS	QTY	REMARKS
1	Arduino UNO With USB-C cable	-	1-SET	
2	Moisture / Soil Sensor	With Probe	1-SET	
3	LED’s	Different Colours	2	
4	Empty Containers for WASTE	Dry WASTE & Wet WASTE	1+1	
5	Jumper wires	-	-	

Rig-Up circuit FOR dry and wet waste



PROBE - Board
A0 = A1 on Arduino
D0 = NC
No Connection
GND = GND on Arduino
VCC = 5V on Arduino

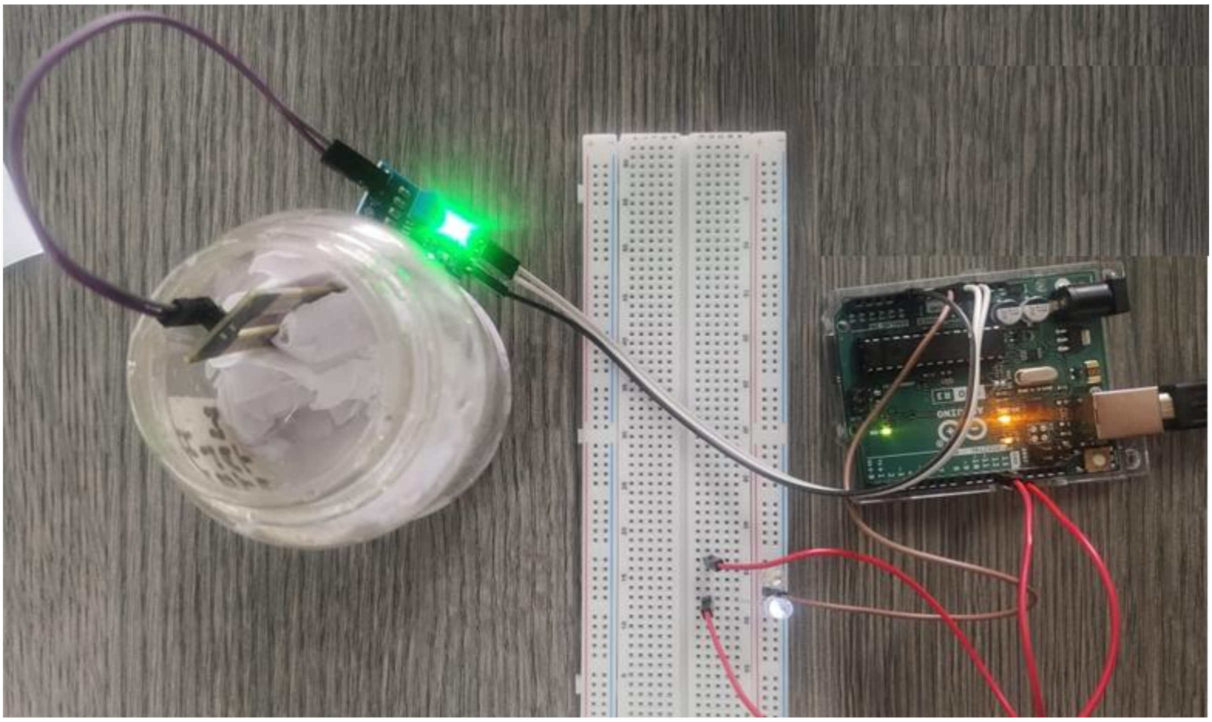
Pin # 12 will be ON when WET waste detected

Pin # 11 will be ON when DRY waste detected

PROBES to be Dipped in the DRY / WET Waste Bin or Container

NC

For Reference Only: **Rig-Up circuit verified at our IoT-Lab**

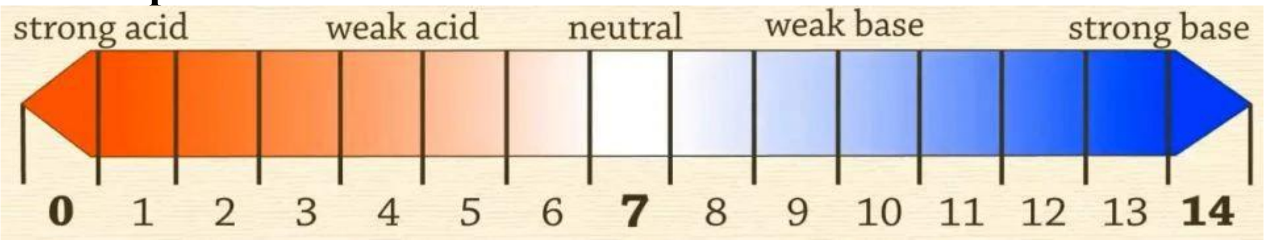


OUTPUT: To be observed on Serial Monitor (For VISUAL indication observe the Relevant LED’s)

7. Develop a program to read the pH value of a various substances like milk, lime and water.

```
float calibration_value = 31.5;
int phval = 0;
unsigned long int avgval;
int buffer_arr[10],temp;
void setup()
{
Serial.begin(9600);
}
void loop() {
for(int i=0;i<10;i++)
 {
buffer_arr[i]=analogRead(A0);
delay(30);
 }
for(int i=0;i<9;i++)
 {
for(int j=i+1;j<10;j++)
 {
 if(buffer_arr[i]>buffer_arr[j])
 {
 temp=buffer_arr[i];
buffer_arr[i]=buffer_arr[j];
buffer_arr[j]=temp;
 }
 }
 }
avgval=0;
for(int i=2;i<8;i++)
avgval+=buffer_arr[i];
 float volt=(float)avgval*5.0/1024/6;
 float ph_act = -5.70 * volt + calibration_value;
Serial.print("pH Value: ");
Serial.println(ph_act);
}
```

What is pH Value?



- pH is a scale that is used to measure the acidity or basicity of an aqueous solution. The pH scale ranges from 0 to 14. 0 being the most acidic and 14 being the most basic. Pure water is neutral and thus has a PH of 7.

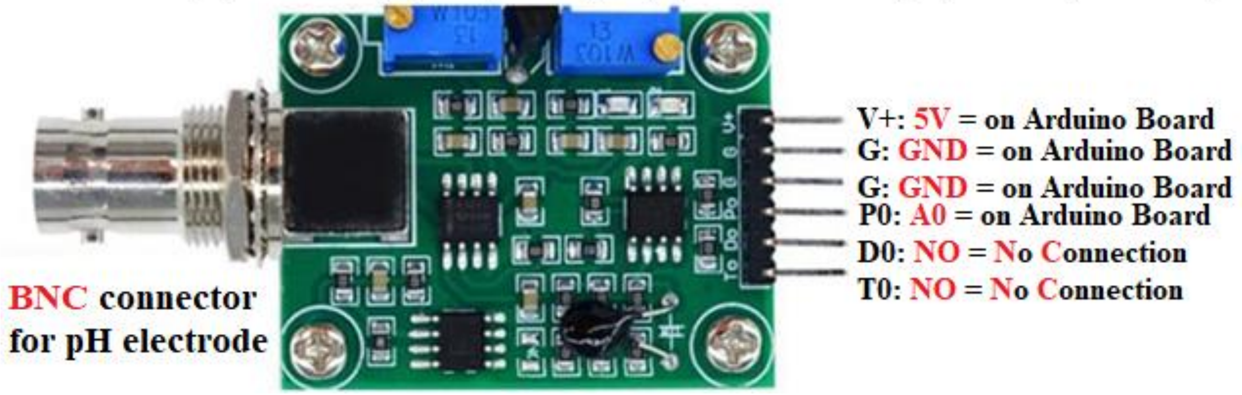
.

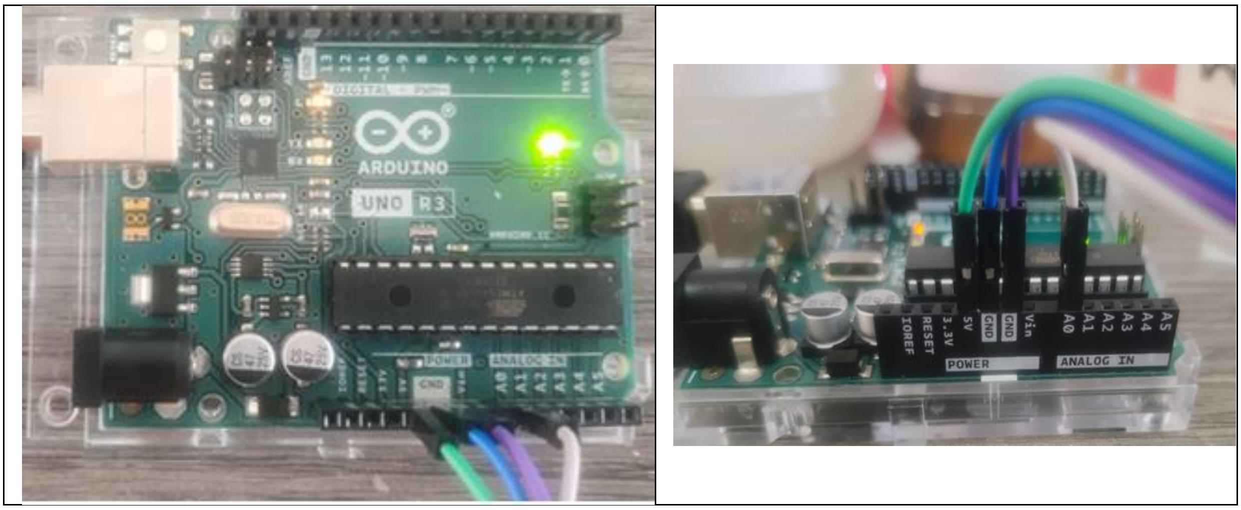
Using DISCRETE / PHYSICAL COMPONENTS

Sl #	NAME OF THE COMPONENTS	SPECIFICATIONS	QTY	REMARKS
1	Arduino UNO With USB-C cable	-	1-SET	
2	Gravity Analog pH sensor	cylindrical glass structure	1	
3	pH Signal Conversion Board	BNC to 6-Pins Board	1	
4	Empty Containers for liquids	Pure water & Other solutions	2-5	
5	Jumper wires	-	-	

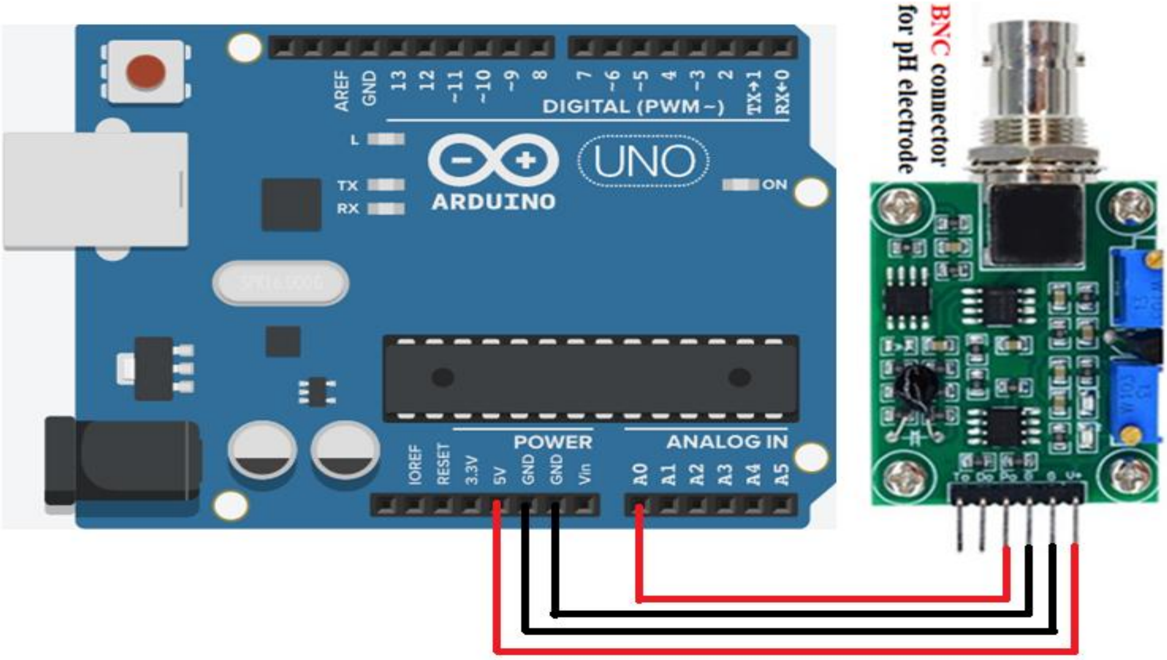
Pin Description of pH Signal Conversion Board

V+: 5V DC input, G: Ground pin, Po: pH analog output, Do: 3.3V DC output, To: Temperature output

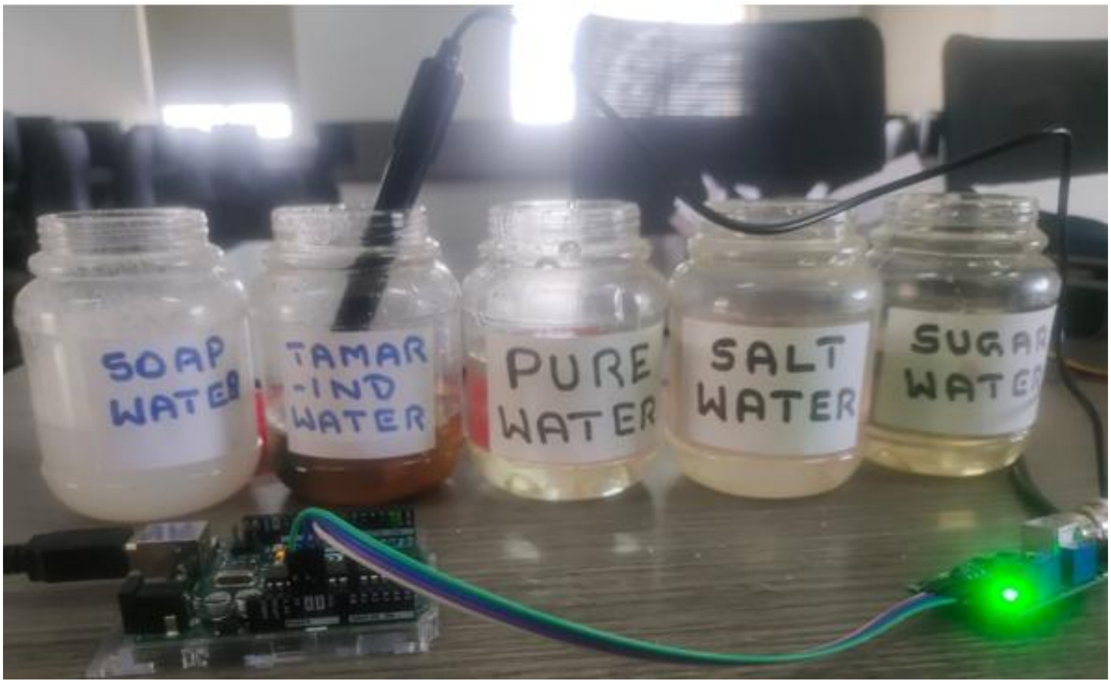

V+: 5V = on Arduino Board
G: GND = on Arduino Board
G: GND = on Arduino Board
P0: A0 = on Arduino Board
D0: NO = No Connection
T0: NO = No Connection
BNC connector for pH electrode



Rig-Up circuit



For Reference Only: Rig-Up circuit verified at our IoT-Lab
We took 5-different solutions and tested the pH meter with them

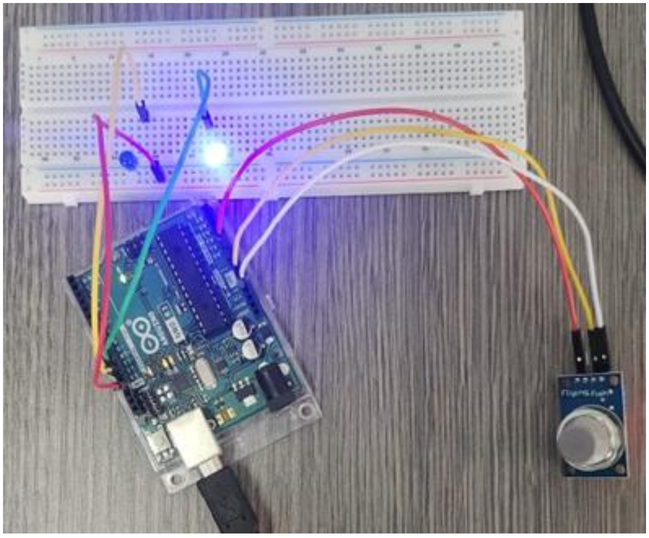


OUTPUT: To be observed on Serial Monitor
(Note down the pH values of Different Solutions)

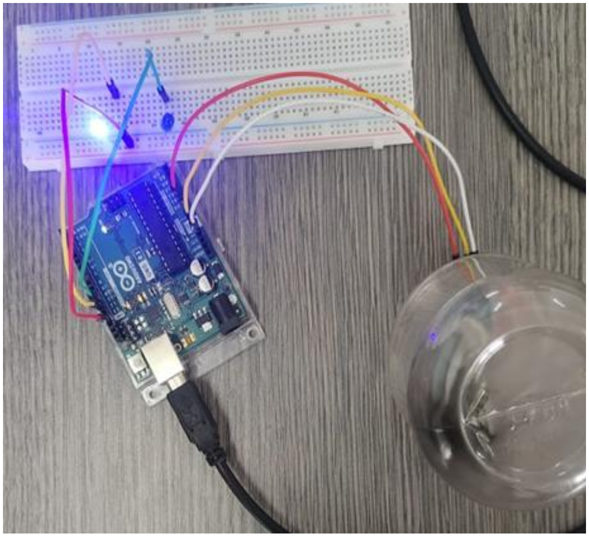
8 Develop a program to detect the gas leakage in the surrounding environment.

```
int GAS_VAL = 0;
int threshold=300;

void setup()
{
  pinMode(A0, INPUT); // MQ-6 A0 Pin
  Serial.begin(9600);

  pinMode(11,OUTPUT); // LED-11: ON no Gas Leakage other time OFF
  pinMode(12,OUTPUT); // LED-12: ON during Gas Leakage Detected other time OFF

}
              void loop()
{
  GAS_VAL = analogRead(A0);
Serial.println(GAS_VAL);
if (GAS_VAL < threshold)

{
    digitalWrite(11,HIGH);
    digitalWrite(12,LOW);
    Serial.println(" No gas leakage ");
  }
  Else
 {
     digitalWrite(11,LOW);
     digitalWrite(12,HIGH);
    Serial.println(" gas leakage detected");

  }
  delay(2000);
}
```

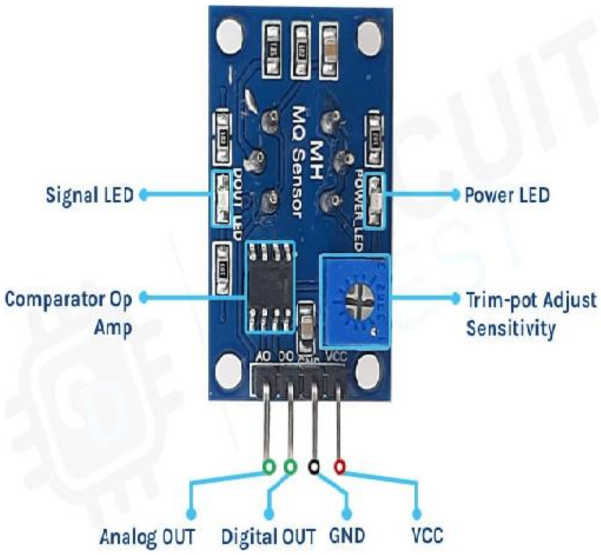
NO gas leakage

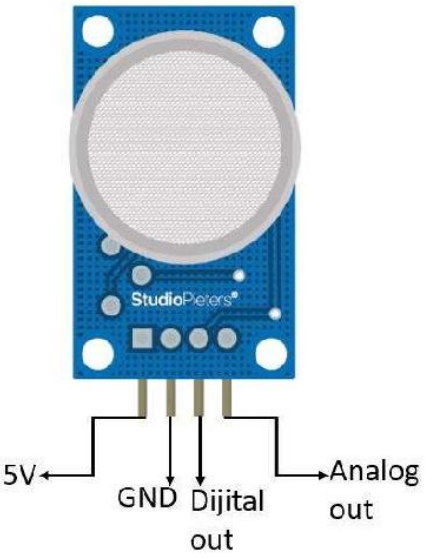


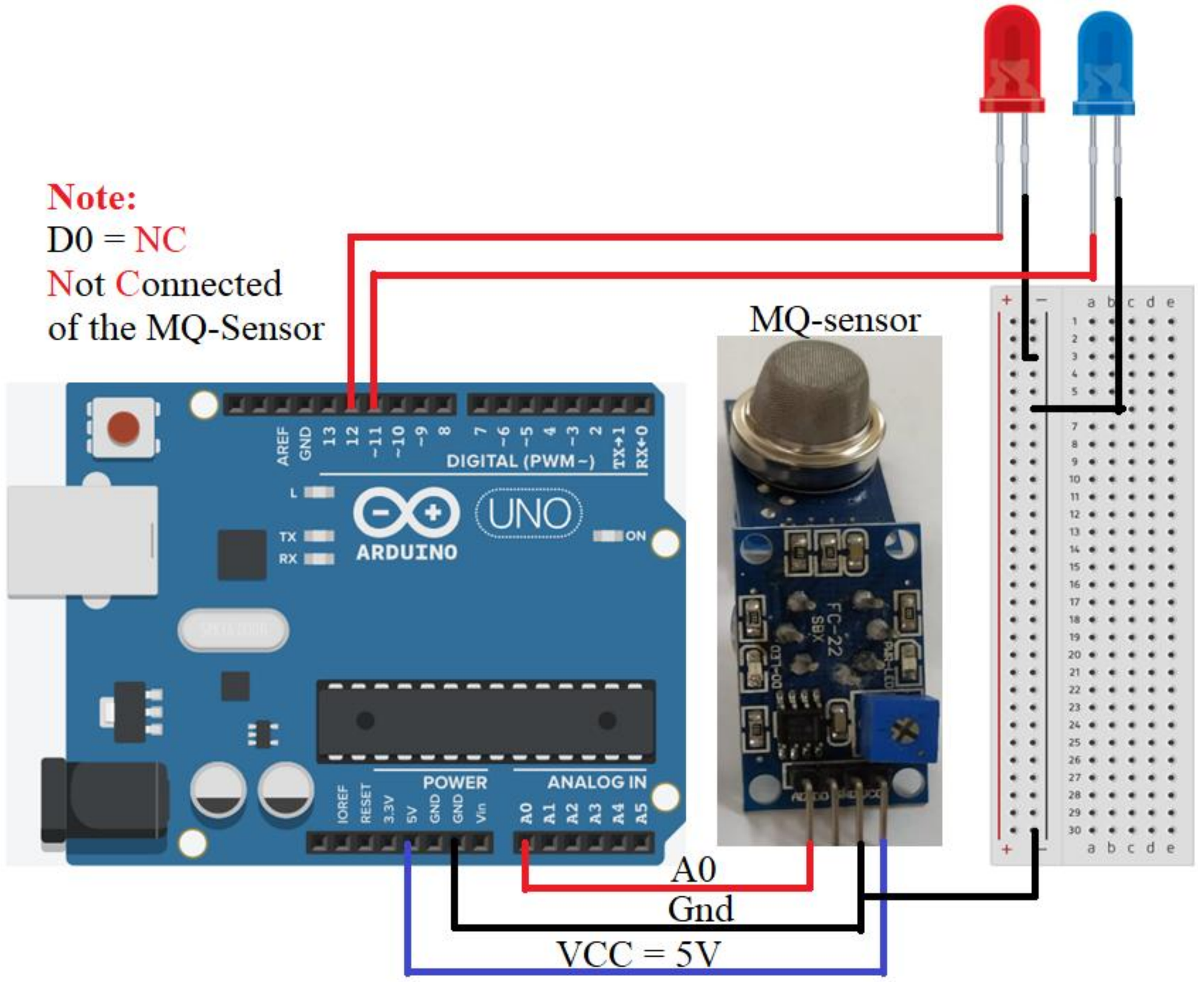
Gas Leakage DETECTED



PINOUT DIAGRAM OF MQ-SENSOR



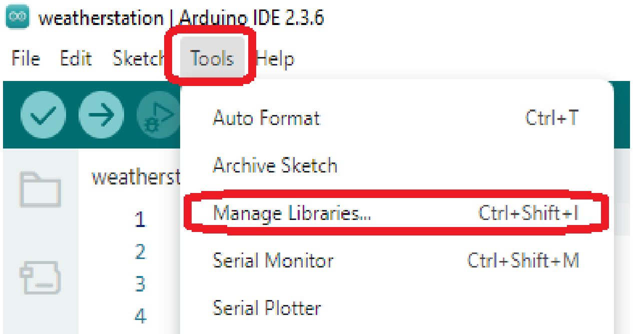



Note:
D0 = NC
Not Connected
of the MQ-Sensor

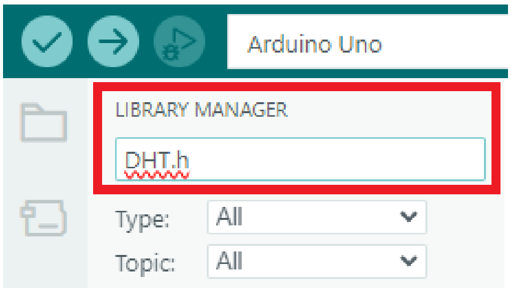
9 Develop a program to demonstrate weather station readings using Arduino.

```
#include "DHT.h"
#define dhtpin 9
#define dhttype DHT11

DHT dht(dhtpin,dhttype);
void setup() {
  // put your setup code here, to run once:
Serial.begin(9600);
dht.begin();

}
void loop() {
  // put your main code here, to run repeatedly:
float humi=dht.readHumidity();
float tempC=dht.readTemperature();
Serial.println("Welcome to the Weather reading Station.  ");
Serial.print("The Humidity is: ");
Serial.print(humi);
Serial.print("%");
Serial.print("    |    ");
Serial.print("The temperature is:");
Serial.print(tempC);
Serial.print(" Degree Celcius ~ \n");
delay(2000);
}
```

<table>
<tr><th colspan="2">Follow below steps to add DHT.h (Header file) before execution and uploading the sketch</th></tr>
<tr>
<td>
Click on Manage Libraries



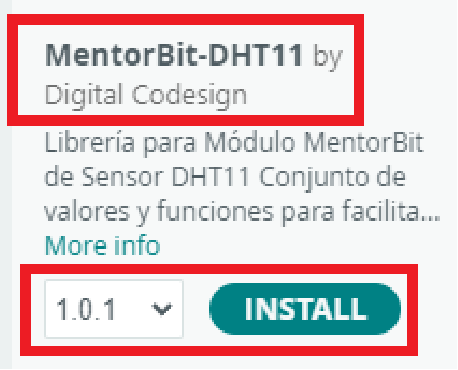
</td>
<td>
Type: DHT.h in the search field



DHT.h

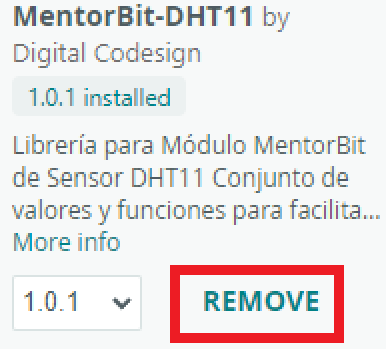
Type: All

Topic: All
</td>
</tr>
<tr>
<td>
Scroll down and click on install as shown below



Librería para Módulo MentorBit de Sensor DHT11 Conjunto de valores y funciones para facilita...

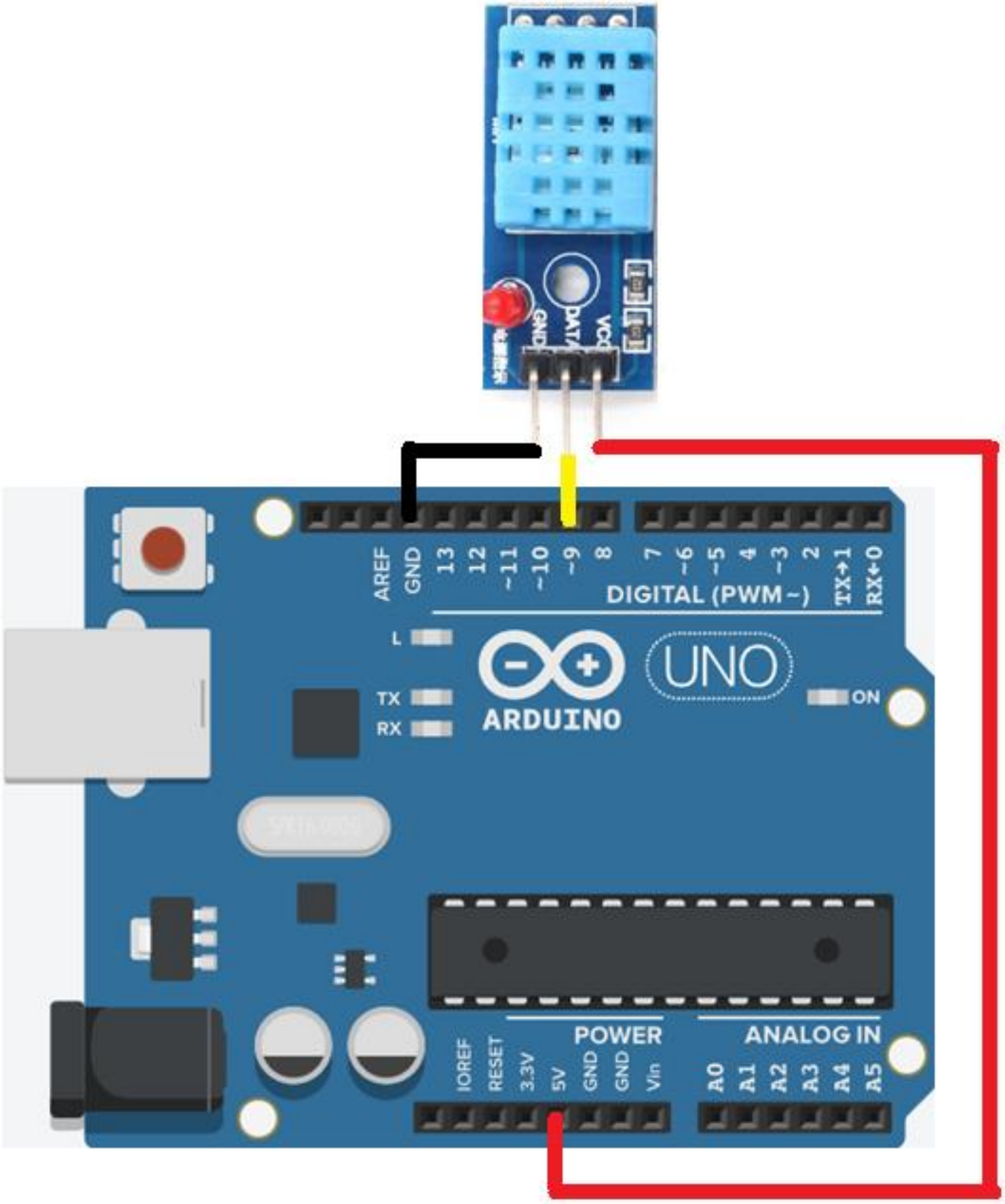
</td>
<td>
Click on install all, REMOVE indicates Installation is successful, verify the sketch



Librería para Módulo MentorBit de Sensor DHT11 Conjunto de valores y funciones para facilita...

</td>
</tr>
</table>

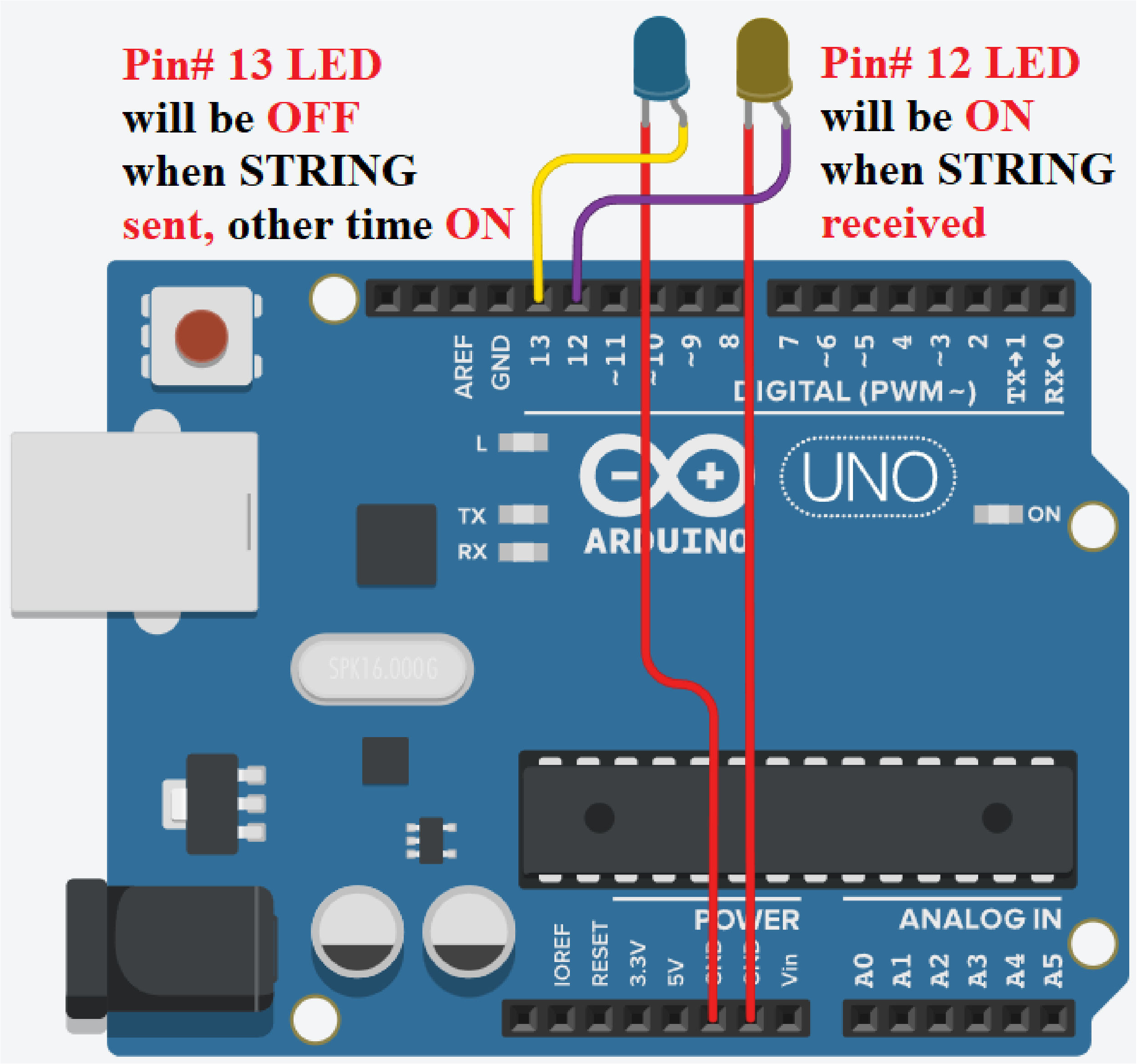
DHT11


GND
DATA
VCC
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
L
TX
RX
ARDUINO
UNO
ON
POWER
ANALOG IN
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5

10. Develop a program to setup a UART protocol and pass a string through the protocol.

```
int S1 = 12;
int S2 = 13;
void setup()
 {
 Serial.begin(9600);
 pinMode(S1, OUTPUT);
 pinMode(S2, OUTPUT);

 }
 void loop()
 {
 if (Serial.available())
 {

 String received = Serial.readString();
 Serial.print("String Received is: ");
 Serial.println(received);
 delay(1000);
 digitalWrite(S2, LOW);
 digitalWrite(S1, HIGH);

 }
 else
 {
  Serial.println("no string received");
  delay(1000);
  digitalWrite(S1, LOW);
  digitalWrite(S2, HIGH);

 }

 }
```

Rig-Up Circuit

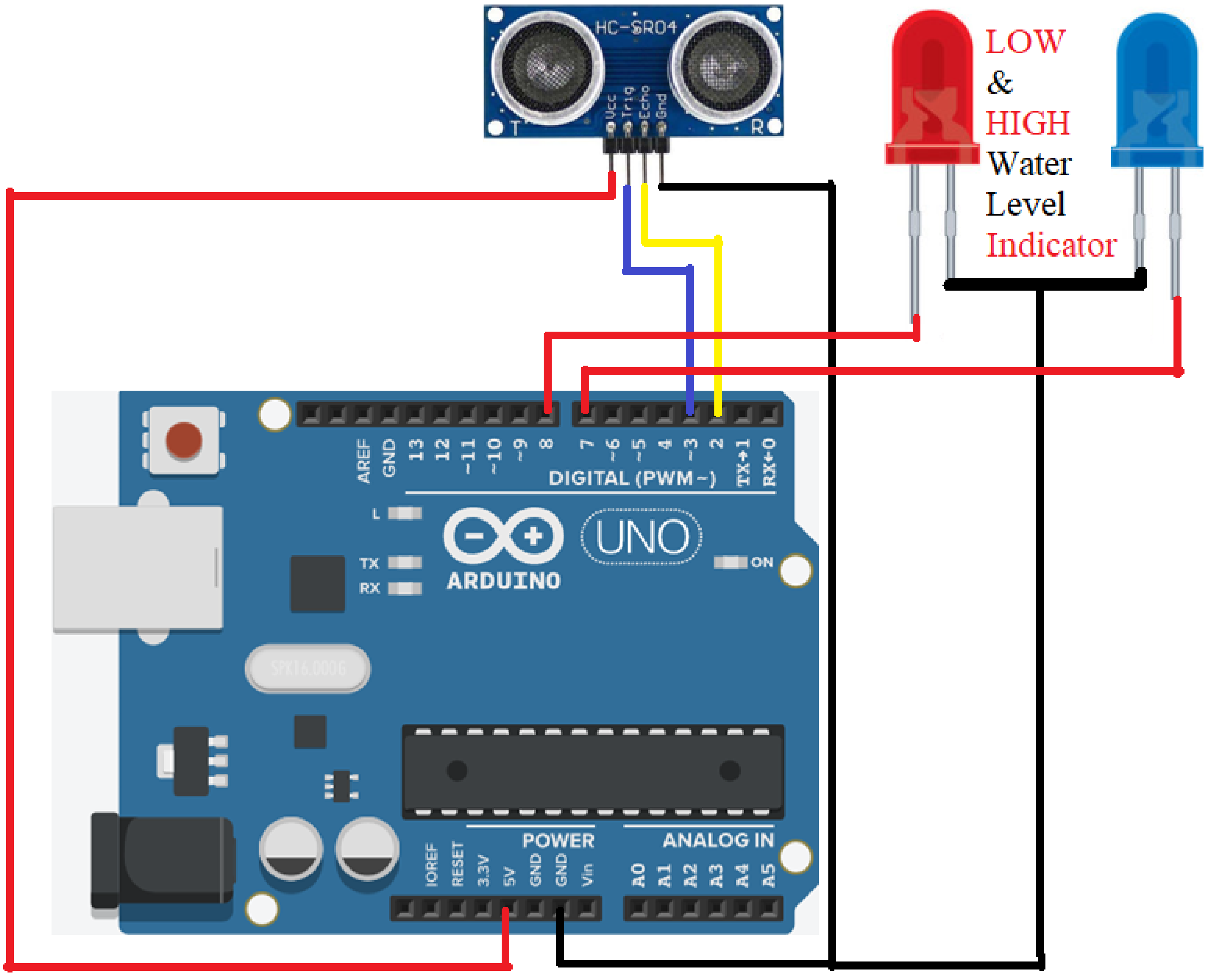


OUTPUT:

Open serial monitor to send the **STRING**

11: Develop a water level depth detection system using Ultrasonic sensor.

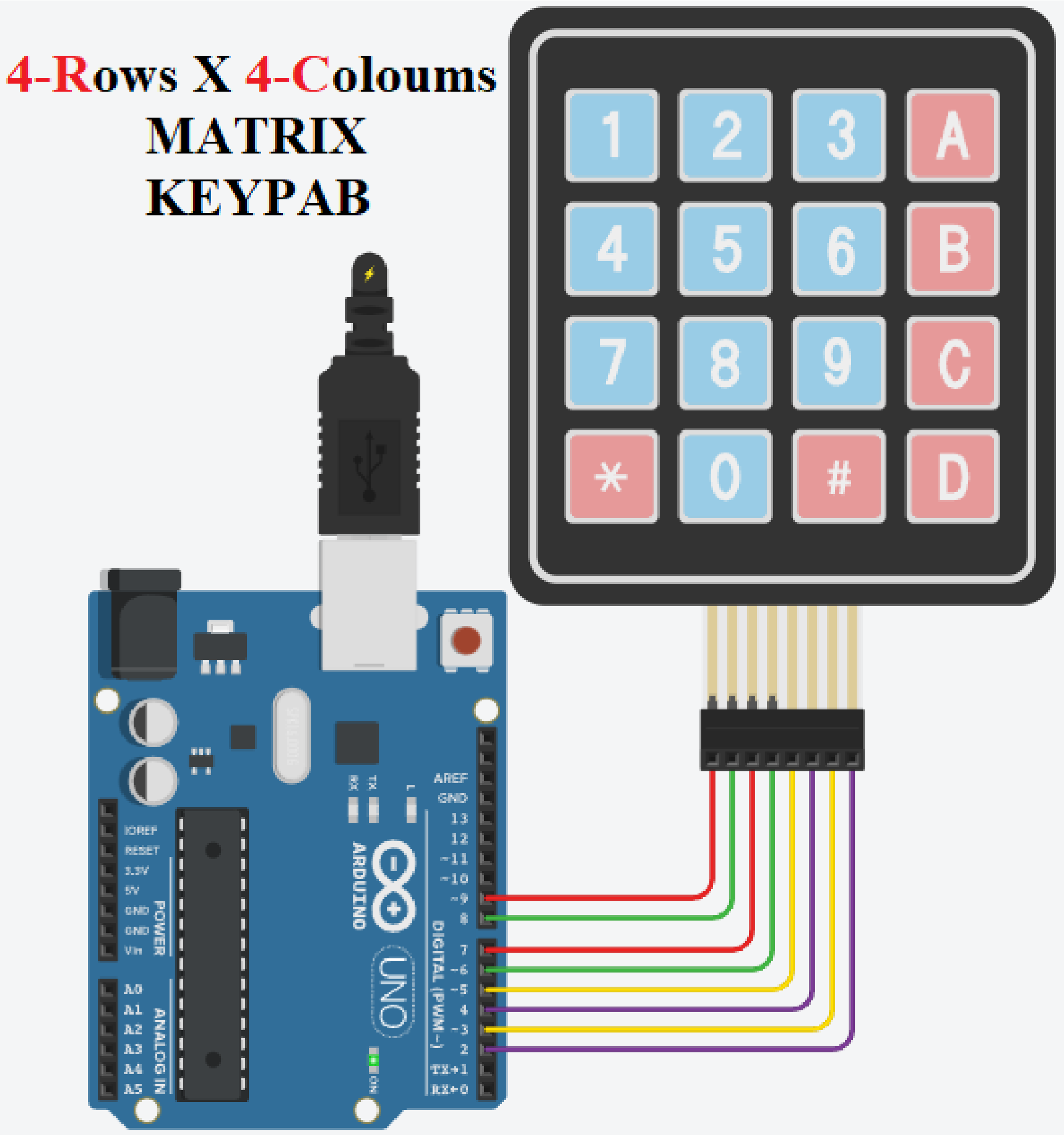
```
#define echoPin 2
#define trigPin 3
#define max 5 // length of the tank(bottle used)
long duration;
int distance;
void setup()
{
    pinMode(trigPin,OUTPUT);
    pinMode(echoPin, INPUT);
    Serial.begin(9600);
}
void loop()
{
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin,HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin,LOW);
    duration = pulseIn(echoPin, HIGH);
    distance = duration * 0.0344 / 2;
    if(distance > max)
   {
    Serial.println("Tank is empty");
    digitalWrite(8,LOW);
    digitalWrite(7,HIGH);
    //Serial.print(distance);
    //Serial.println(" cm");
    delay(1000);
   }
   else{
     Serial.println("There is water in the tank");
     digitalWrite(7,LOW);
    digitalWrite(8,HIGH);
   // Serial.print(distance);
    //Serial.println(" cm");

    delay(1000);
   }
}
```


HC-SR04
T
R
Vcc
Trig
Echo
Gnd
LOW
&
HIGH
Water
Level
Indicator
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM ~)
L
TX
RX
ARDUINO
UNO
ON
POWER
ANALOG IN
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5

12. Develop a program to simulate interfacing with the keypad module to record the keystrokes.

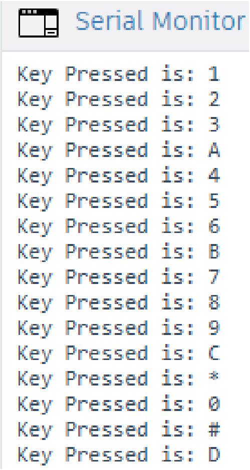
```
#include <Keypad.h>
const byte ROWS = 4;
const byte COLS = 4;

char keys[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 keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

void setup(){
 Serial.begin(9600);
}
void loop(){
 char key = keypad.getKey();
 if (key){
   Serial.print("Key Pressed is: ");
   Serial.println(key);
 }
}
```

Rig-Up circuit



OUTPUT: To be observed on Serial Monitor (Press any the KEY on the KEYPAD MODULE & verify)

Sample OUTPUT



```
Key Pressed is: 1
Key Pressed is: 2
Key Pressed is: 3
Key Pressed is: A
Key Pressed is: 4
Key Pressed is: 5
Key Pressed is: 6
Key Pressed is: B
Key Pressed is: 7
Key Pressed is: 8
Key Pressed is: 9
Key Pressed is: C
Key Pressed is: *
Key Pressed is: 0
Key Pressed is: #
Key Pressed is: D
```