**1.LED INTERFACING**

Ex1.1 Blinking LED

This example shows the simplest thing you can do with an INDUINO to see physical output: it blinks an LED.

Turns on an LED on for one second, then off for one second, repeatedly.

**Code:**

void setup()

{

pinMode(11,OUTPUT);

}

void loop()

{

digitalWrite(11,HIGH);

delay(1000);

digitalWrite(11,LOW);

delay(1000);

}

Ex1.2 Five times only ON

Ex 1.3 Binary counter

This sketch automatically increases a 3 bit number every second and shows the output on 3 LEDs

the LSB is toggling every cycle, the Middle Bit is toggling every 2 cycles and the MSB is toggling every cycle, To achieve this effect, we can use the MOD operator (%)

Counter % 2 will produce an alternating output of '1' & '0' that changes every cycle

(Counter/2) % 2 will produce an alternating output of '1' & '0' that changes every 2 cycles(whatever you divide by!)

(Counter/4) % 2 will produce an alternating output of '1' & '0' that changes every 4 cycles

**Code:**

void setup()

{

pinMode(11,OUTPUT); // declare LED pins as output pins

pinMode(12,OUTPUT);

pinMode(13,OUTPUT);

}

void loop()

{

for(int i=0;i<8;i++) // increment automatically from 0 to 7 , i is the counter variable

{

int a=i%2; // calculate LSB

int b=i/2 %2; // calculate middle bit

int c=i/4 %2; // calculate MSB

digitalWrite(11,a); // write MSB

digitalWrite(12,b); // write middle bit

digitalWrite(13,c); // write LSB

delay(1000); // wait for a second

}

}

**2.RGB LED**

Ex2.1 ANALOG OUTPUT USING RGB LED

This example shows how we can generate some random color using RGB LED.

**Code**

void setup()

{ // Leave the setup() function empty

}

void loop()

{

analogWrite(3,153);// Setting the voltage for Blue to around 3 Volts

analogWrite(5,51);// Setting the voltage for Green to around 1 Volt

analogWrite(6,51);// Setting the voltage for Red to around 1 Volt

delay(500);

analogWrite(3,51);

analogWrite(5,153);

analogWrite(6,51);

delay(500);

analogWrite(3,51);

analogWrite(5,51);

analogWrite(6,153);

delay(500);

}

Ex2.2 RGB LED INDENSITY CONTROL

Let's try and fade in / fade out the Red Color of the RGB LED.  We use 2 push buttons 7 & 9 for this.

**Code**

byte intensity = 0;

void setup()

{

pinMode(7,INPUT\_PULLUP);

pinMode(9,INPUT\_PULLUP);

}

void loop()

{

if(digitalRead(7)==0)

{

if(intensity<255)

intensity++;

else

intensity = 0;

delay(20);

}

if(digitalRead(9)==0)

{

if(intensity>0)

intensity--;

else

intensity = 255;

delay(20);

}

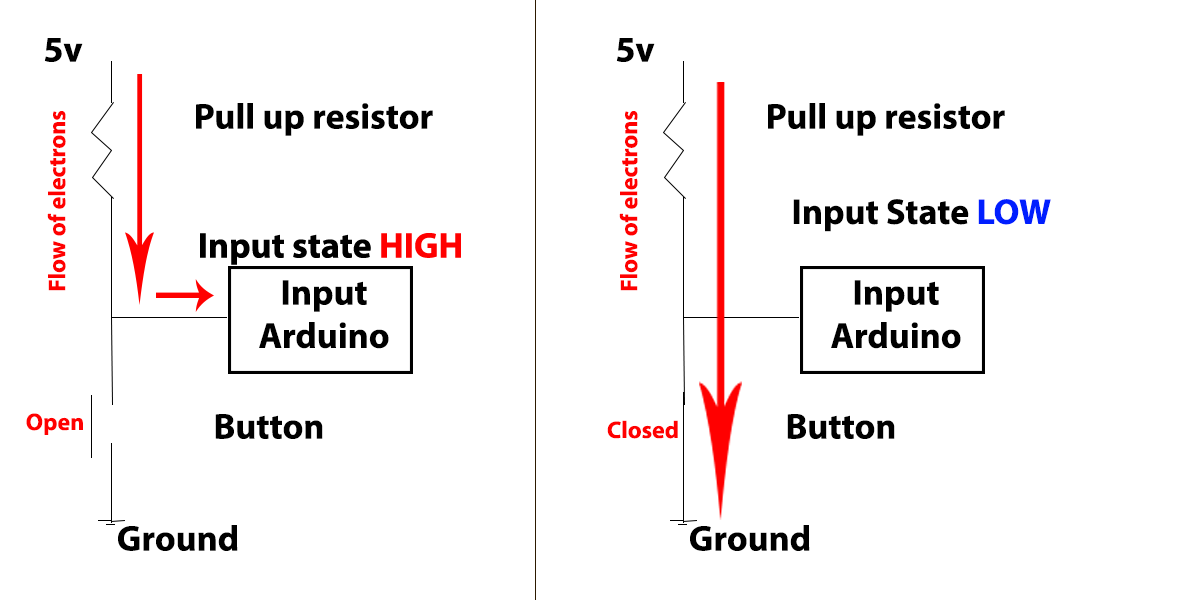
analogWrite(6,intensity);

}

**3. SWITCH INTERFACING**

Ex3.1 SWITCH WITH LED

Lets Build a Simple Button Based Toggle Control for a LED. When the Button on pin 7 is pressed, the LED on pin 13 will toggle state.



**Code**

constintbuttonPin = 7;

constintledPin = 13;

intbuttonState = 0;

void setup(){

pinMode(7, INPUT\_PULLUP);

pinMode(13, OUTPUT);

}

void loop(){

buttonState=digitalRead(7);

if(buttonState==HIGH){

digitalWrite(13,LOW);

}

else{

digitalWrite(13,HIGH);

}

while(digitalRead(7)==0); // wait for the switch to be released - Part of Debounce

delay(100);

}

Ex3.2 SWITCH WITH BINARY COUNTER

Now Lets try and add the Button to increment our Binary Counter.

**Code:**

intbuttonState = 0;

int i=0;

void setup(){

pinMode(7, INPUT\_PULLUP);

pinMode(11, OUTPUT);

pinMode(12, OUTPUT);

pinMode(13, OUTPUT);

}

void loop(){

buttonState=digitalRead(7);

if(buttonState==LOW)

{

if(i<7) // if counter value is less than 7 or 3 bits

i++; // increment counter value

else

i=0; // reset counter to 0

int a=i%2; // calculate LSB

int b=i/2 %2; // calculate middle bit

int c=i/4 %2; // calculate MSB

digitalWrite(11,a); // write LSB

digitalWrite(12,b); // write middle bit

digitalWrite(13,c); // write MSB

while(digitalRead(7)==0); // wait till button is released to avoid incrementing the counter again

delay(100); // small delay to avoid debounce

}

}

**4. SERIAL COMMUNICATION**

Ex4.1 SERIAL WRITE

Lets see how we can print some data from our Board to the Computer.

**Code**

char txt[]="The current value is:";

intbuttonState = 0;

int i=0;

void setup(){

pinMode(7, INPUT\_PULLUP);

pinMode(11, OUTPUT);

pinMode(12, OUTPUT);

pinMode(13, OUTPUT);

Serial.begin(9600);

Serial.println("Starting the Program");

}

void loop(){

buttonState=digitalRead(7);

if(buttonState==LOW)

{

if(i<7) // if counter value is less than 7 or 3 bits

i++; // increment counter value

else

i=0; // reset counter to 0

Serial.print(txt); // Print Descriptive test from the character array

Serial.println(i); // print the current value

int a=i%2; // calculate LSB

int b=i/2 %2; // calculate middle bit

int c=i/4 %2; // calculate MSB

digitalWrite(11,a); // write LSB

digitalWrite(12,b); // write middle bit

digitalWrite(13,c); // write MSB

while(digitalRead(7)==0); // wait till button is released to avoid incrementing the counter again

delay(100); // small delay to avoid debounce

}

}

Ex4.2 SERIAL READ

Lets see how we can read some data from our Computer to the Board.

**Code**

char txt[]="The current value is:";

intbuttonState = 0;

int i=0;

void setup(){

pinMode(7, INPUT\_PULLUP);

pinMode(11, OUTPUT);

pinMode(12, OUTPUT);

pinMode(13, OUTPUT);

Serial.begin(9600);

Serial.println("Starting the Program");

}

void loop(){

buttonState=digitalRead(7);

if(buttonState==LOW || Serial.available()>0){

if(buttonState==LOW)

{

if(i<7)

i++;

else

i=0;

while(digitalRead(7)==0);

delay(100);

}

if(Serial.available()>0)

{

intval = Serial.read();

val = val - 48;

if(val>=0 &&val<=7)

{

i = val;

}

}

Serial.print(txt);

Serial.println(i);

int a=i%2;

int b=i/2 %2;

int c=i/4 %2;

digitalWrite(11,a);

digitalWrite(12,b);

digitalWrite(13,c);

while(digitalRead(7)==0);

delay(100);

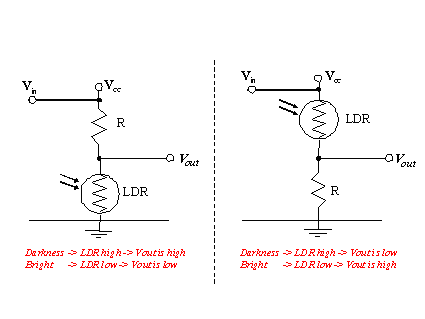
}

}

**5. LDR(LIGHT DEPENDENT RESISTOR)**

Ex5.1 READ ANALOG VALUE USING LDR

This example is how we can read Analog value using LDR.



**Code**

intldrValue = 0;

void setup()

{

Serial.begin(9600);

}

void loop(){

ldrValue = analogRead(A3);

Serial.print("Current LDR Value : ");

Serial.println(ldrValue);

delay(1000);

}

Ex5.2 LED INDENSITY VARIATION USING LDR

This example we can see the intensity variation of LED based on LDR value.

**Code**

intldrValue = 0

intledValue=0;

void setup()

{

Serial.begin(9600); }

void loop()

{

ldrValue = analogRead(3);

ledValue=map(ldrValue, 0, 1023, 0, 255);

analogWrite(11, ledValue);

Serial.print("Current LDR Value : ");

Serial.println(ldrValue);

delay(1000);

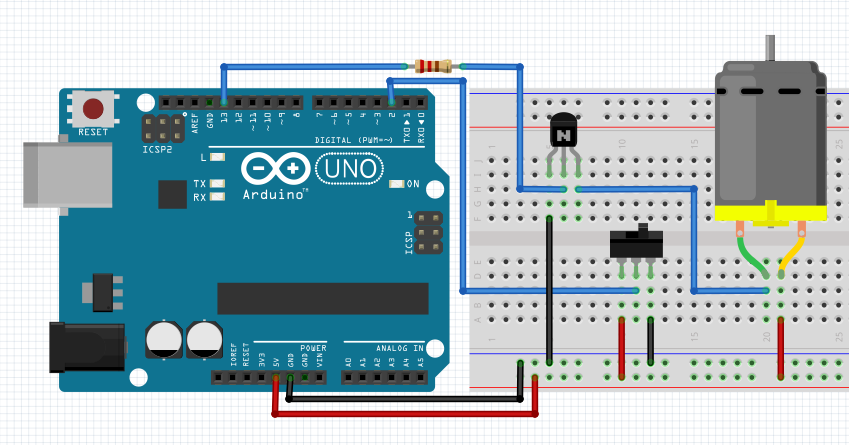
}

**6.DC MOTOR INTERFACING**

Ex6.1 SWITCH WITH DC MOTOR

Pushbuttons or switches connect two points in a circuit when you press them. This example turns on the MOTOR on pin 13 when you press the button.

**Circuit diagram**

**Code**

// constants won't change. They're used here to

// set pin numbers:

constintbuttonPin = 2; // the number of the pushbutton pin

constintledPin = 13; // the number of the LED pin

// variables will change:

intbuttonState = 0; // variable for reading the pushbutton status

void setup() {

// initialize the LED pin as an output:

pinMode(ledPin, OUTPUT);

// initialize the pusbutton pin as an input:

pinMode(buttonPin, INPUT);

}

void loop() {

// read the state of the pushbutton value:

buttonState = digitalRead(buttonPin);

// check if the pushbutton is pressed.

// if it is, the buttonState is HIGH:

if (buttonState == HIGH) {

// turn LED on:

digitalWrite(ledPin, HIGH);

}

else {

// turn LED off:

digitalWrite(ledPin, LOW);

}

}

Ex.6.2 SWITCH WITH DC MOTOR & LED

Aim of the mini project switch ON the DC Motor and LED, when you ON the switch.

**Code**

constintbuttonPin = 2; // the number of the pushbutton pin

constintledPin = 13; // the number of the LED pin

constint led1Pin = 7;

// variables will change:

intbuttonState = 0; // variable for reading the pushbutton status

void setup() {

// initialize the LED pin as an output:

pinMode(ledPin, OUTPUT);

pinMode(led1Pin, OUTPUT);

// initialize the pushbutton pin as an input:

pinMode(buttonPin, INPUT);

}

void loop() {

// read the state of the pushbutton value:

buttonState = digitalRead(buttonPin);

// check if the pushbutton is pressed.

// if it is, the buttonState is HIGH:

if (buttonState == HIGH) {

// turn LED on:

digitalWrite(ledPin, HIGH);

digitalWrite(led1Pin, HIGH);

}

else {

// turn LED off:

digitalWrite(ledPin, LOW);

digitalWrite(led1Pin, LOW);

}

}

Ex6.3 SWITCH WITH DC MOTOR & LED

Aim of the mini project when you ON the switch the motor will run at same time the LED will OFF.

**Code**

constintbuttonPin = 2; // the number of the pushbutton pin

constintledPin = 13; // the number of the LED pin

constint led1Pin = 7;

// variables will change:

intbuttonState = 0; // variable for reading the pushbutton status

void setup() {

// initialize the LED pin as an output:

pinMode(ledPin, OUTPUT);

pinMode(led1Pin, OUTPUT);

// initialize the pushbutton pin as an input:

pinMode(buttonPin, INPUT);

}

void loop() {

// read the state of the pushbutton value:

buttonState = digitalRead(buttonPin);

// check if the pushbutton is pressed.

// if it is, the buttonState is HIGH:

if (buttonState == HIGH) {

// turn LED on:

digitalWrite(ledPin, HIGH);

digitalWrite(led1Pin, LOW);

}

else {

// turn LED off:

digitalWrite(ledPin, LOW);

digitalWrite(led1Pin, HIGH);

}

}

Ex6.4 LDR WITH DC MOTOR & LED

Aim of the mini project when the sunset occur at same time garden motor will start and LED also ON for indication purpose.

**Code**

intldrValue = 0; // variable to store the LDR value

intledPin = 13;

intmotorPin = 11;

int limit = 800;

void setup()

{

// No pinMode Statement required for Analog Pins :)

Serial.begin(9600); // Initialise Serial Communication

pinMode(ledPin, OUTPUT);

pinMode(motorPin, OUTPUT);

}

void loop()

{

ldrValue = analogRead(3); // Read the ldr value and store it in the variable

//ledValue=map(ldrValue, 0, 1023, 255, 0);

//analogWrite(11, ledValue);

Serial.print("Current LDR Value : ");

Serial.println(ldrValue); // print the ldr value to serial monitor

delay(1000); // a delay - do not remove this delay (you can change it) as otherwise the serial monitor will get flooded with data and might crash

if(ldrValue>=limit )

{

digitalWrite(motorPin, HIGH);

digitalWrite(ledPin, HIGH);

}

else

{

digitalWrite(motorPin, LOW);

digitalWrite(ledPin, LOW);

}

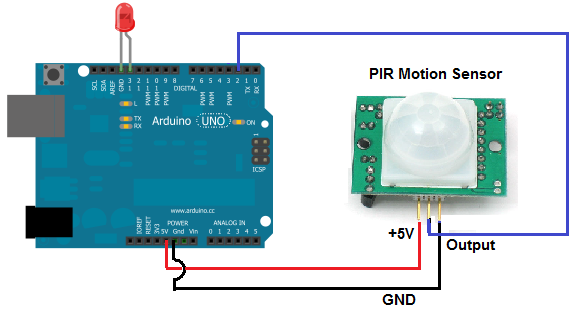
}

**7. PIR SENSOR INTERFACING**

Ex7.1 PIR SENSOR WITH LED

PIR sensor connect with Arduino board and detect obstacle.

**Circuit diagram**

****

**Code**

// constants won't change. They're used here to

// set pin numbers:

constintbuttonPin = 2; // the number of the pushbutton pin

constintledPin = 13; // the number of the LED pin

// variables will change:

intbuttonState = 0; // variable for reading the pushbutton status

void setup() {

// initialize the LED pin as an output:

pinMode(ledPin, OUTPUT);

// initialize the pushbutton pin as an input:

pinMode(buttonPin, INPUT);

}

void loop() {

// read the state of the pushbutton value:

buttonState = digitalRead(buttonPin);

// check if the pushbutton is pressed.

// if it is, the buttonState is HIGH:

if (buttonState == HIGH) {

// turn LED on:

digitalWrite(ledPin, HIGH);

}

else {

// turn LED off:

digitalWrite(ledPin, LOW);

}

}

Ex7.2 PIR SENSOR WITH LED & BUZZER & DC MOTOR

Aim of the project automatic door open at same time indicator also ON(LED & BUZZER), when the PIR sensor will detect human.

Code

constintbuttonPin = 2; // the number of the pushbutton pin

constintledPin = 13; // the number of the LED pin

constintbuzzPin = 12;

constintmotorPin = 11;

// variables will change:

intbuttonState = 0; // variable for reading the pushbutton status

void setup() {

// initialize the LED pin as an output:

pinMode(ledPin, OUTPUT);

pinMode(buzzPin, OUTPUT);

pinMode(motorPin, OUTPUT);

// initialize the pushbutton pin as an input:

pinMode(buttonPin, INPUT);

}

void loop() {

// read the state of the pushbutton value:

buttonState = digitalRead(buttonPin);

// check if the pushbutton is pressed.

// if it is, the buttonState is HIGH:

if (buttonState == HIGH) {

// turn LED on:

digitalWrite(ledPin, HIGH);

digitalWrite(buzzPin, HIGH);

digitalWrite(motorPin, HIGH);

}

else {

// turn LED off:

digitalWrite(ledPin, LOW);

digitalWrite(buzzPin, LOW);

digitalWrite(motorPin, LOW);

}

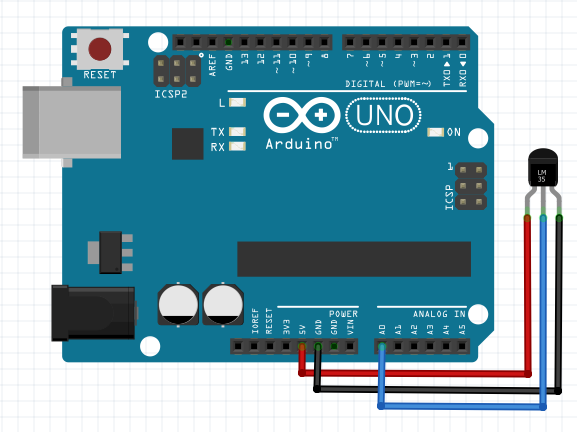
}

**8.TEMPERATURE SENSOR INTERFACING**

Ex8.1 READ TEMPERATURE VALUE

Temperature sensor connect with Arduino board and calculate temperature value.

**Circuit diagram**

****

**Code**

floattempC;

int reading;

inttempPin = 0;

void setup()

{

analogReference(INTERNAL);

Serial.begin(9600);

}

void loop()

{

reading = analogRead(tempPin);

tempC = reading / 9.31;

Serial.print("temp=");

Serial.print(tempC);

delay(1000);

}

Ex8.2 TEMPERATURE SENSOR WITH LED & BUZZER

Aim of the project when the boiler temperature exceed above 33ºC in a industry, the cooling fan will start and indication device also start.

**Code**

floattempC;

int reading;

inttempPin = 0;

int limit = 33;

constintledPin = 13;

constintbuzzPin = 12;

constintmotorPin = 11;

void setup()

{

analogReference(INTERNAL);

Serial.begin(9600);

pinMode(ledPin, OUTPUT);

pinMode(buzzPin, OUTPUT);

pinMode(motorPin, OUTPUT);

}

void loop()

{

reading = analogRead(tempPin);

tempC = reading / 9.31;

Serial.print("temp=");

Serial.print(tempC);

delay(500);

if(tempC>=limit){

digitalWrite(ledPin, HIGH);

digitalWrite(buzzPin, HIGH);

digitalWrite(motorPin, HIGH);

}

else{

digitalWrite(ledPin, LOW);

digitalWrite(buzzPin, LOW);

digitalWrite(motorPin, LOW);

}

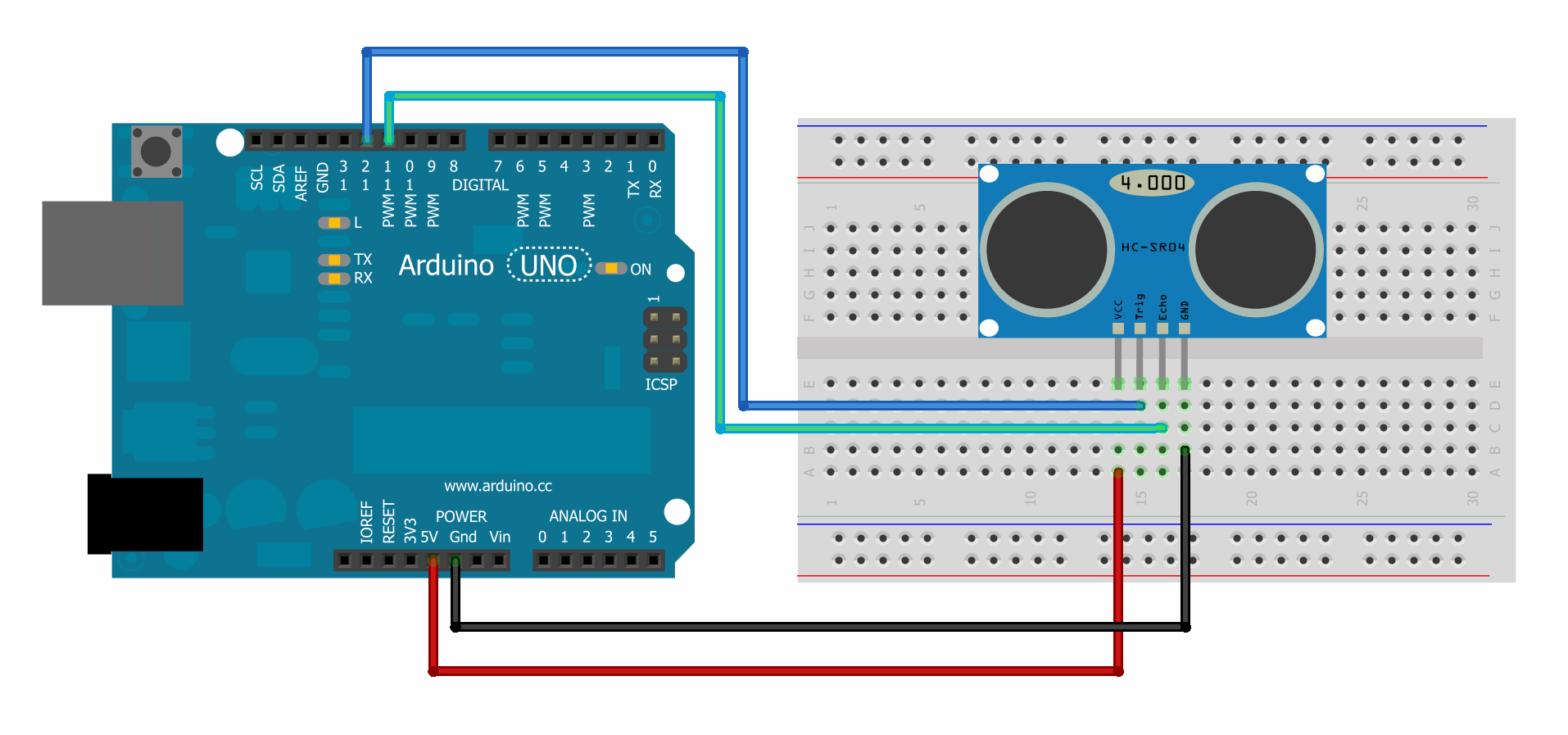
}

**9.ULTRASONIC SENSOR INTERFACING**

Ex9.1 FIND OBSTACLE DISTANCE USING ULTRASONIC

Ultrasonic sensor connect with Arduino board and calculate obstacle distance.

**Circuit diagram**

****

**Code**

#define echoPin 11

#define trigPin 12

void setup() {

Serial.begin(9600);

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

}

void loop() {

int distance,duration;

digitalWrite(trigPin, HIGH);

delay(100);

digitalWrite(trigPin, LOW);

duration=pulseIn(echoPin, HIGH);

distance=(duration/2)/29.1;

Serial.print(distance);

Serial.print("CM");

Serial.println("");

delay(1000);

}

Ex9.2 ULTRASONIC SENSOR WITH DC MOTOR & LED & BUZZER

Aim of the project a robot following a track with carrying some material in a industry. If any obstacle is occur the track, suddenly the robot will stop and LED also OFF at same time buzzer will shout.

**Code**

#define echoPin 11

#define trigPin 12

constintmotorPin = 9;

constintbuzzPin = 10;

constintledPin = 13;

int limit = 20;

void setup() {

// put your setup code here, to run once:

Serial.begin(9600);

pinMode(trigPin, OUTPUT);

pinMode(motorPin, OUTPUT);

pinMode(buzzPin, OUTPUT);

pinMode(ledPin, OUTPUT);

pinMode(echoPin, INPUT);

}

void loop() {

// put your main code here, to run repeatedly:

intdistance,duration;

digitalWrite(trigPin, HIGH);

delay(100);

digitalWrite(trigPin, LOW);

duration=pulseIn(echoPin, HIGH);

distance=(duration/2)/29.1;

Serial.print(distance);

Serial.print("CM");

Serial.println("");

delay(1000);

if(distance>=limit)

{

digitalWrite(motorPin, HIGH);

digitalWrite(ledPin, HIGH);

digitalWrite(buzzPin, LOW);

}

else

{

digitalWrite(motorPin, LOW);

digitalWrite(ledPin, LOW);

digitalWrite(buzzPin, HIGH);

}

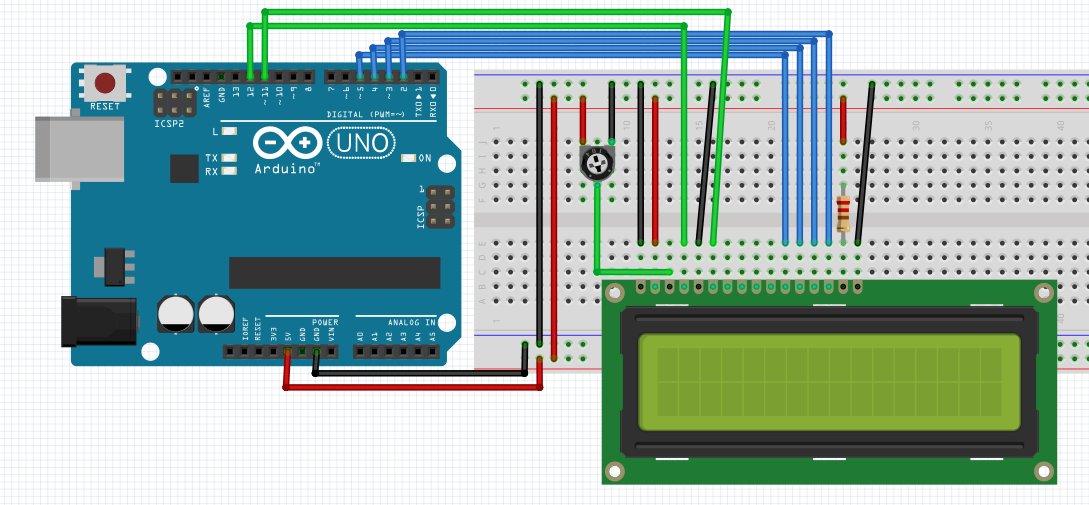
}

**10. STEPPER MOTOR INTERFACING**

**11.LCD INTERFACING**

LCD display connect with Arduino board.

**Circuit diagram**

****

**Code**

\* LCD RS pin to digital pin 12

\* LCD Enable pin to digital pin 11

\* LCD D4 pin to digital pin 5

\* LCD D5 pin to digital pin 4

\* LCD D6 pin to digital pin 3

\* LCD D7 pin to digital pin 2

\* LCD R/W pin to ground

\* LCD VSS pin to ground

\* LCD VCC pin to 5V

\* 10K resistor:

\* ends to +5V and ground

\* wiper to LCD VO pin (pin 3)

// include the library code:

#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins

LiquidCrystallcd(12, 11, 5, 4, 3, 2);

void setup() {

// set up the LCD's number of columns and rows:

lcd.begin(16, 2);

// Print a message to the LCD.

lcd.print("hello, world!");

}

void loop() {

// set the cursor to column 0, line 1

// (note: line 1 is the second row, since counting begins with 0):

lcd.setCursor(0, 1);

// print the number of seconds since reset:

lcd.print(millis() / 1000);

}