**1.Make your Arduino board blink an LED.**

**Aim:**

To make a LED Blink operation using Arduino

**Procedure:**

1. Start the Arduino IDE and enter the code
2. Declare the variable for input pin
3. Initialize digital pin 7 as an output in void setup function
4. The loop function runs over and over again forever
5. Turn the LED on by making the voltage HIGH
6. Turn the LED on by making the voltage LOW

**Program**

int led = 7;

void setup() {

pinMode(led, OUTPUT);

}

void loop() {

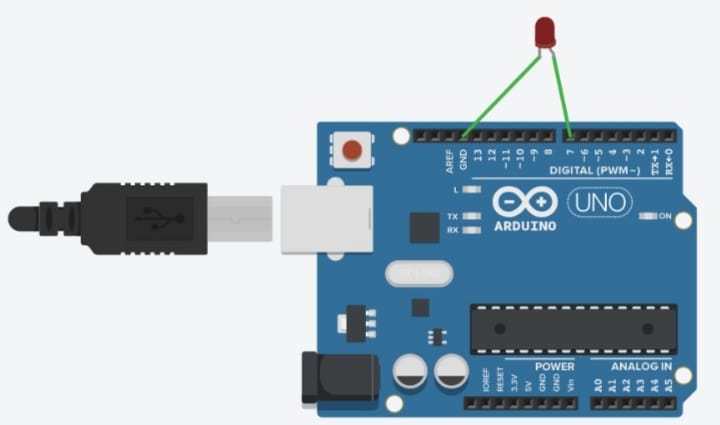
digitalWrite(led, HIGH);

delay(1000);

digitalWrite(led, LOW);

delay(1000);

}



**2.Create an application to find Motion detection using arduino board.**

**Aim:**

To find Motion detection using Arduino board.

**Procedure:**

1. Start the Arduino IDE and enter the code
2. Declare the PIR sensor output variable
3. Define the baud rate for serial transmission
4. The loop function runs over and over again forever
5. Based on the sensor output show the movement.
6. If the PIR sensor value is 1, then print the statement of movement detection
7. If the PIR sensor value is 0, then print the statement of no movement detection.

**Program:**

int led = 13; // the pin that the LED is atteched to

int sensor = 2; // the pin that the sensor is atteched to

int state = LOW; // by default, no motion detected

int val = 0; // variable to store the sensor status (value)

void setup() {

pinMode(led, OUTPUT); // initialize LED as an output

pinMode(sensor, INPUT); // initialize sensor as an input

Serial.begin(9600); // initialize serial

}

void loop(){

val = digitalRead(sensor); // read sensor value

if (val == HIGH) { // check if the sensor is HIGH

digitalWrite(led, HIGH); // turn LED ON

delay(500); // delay 100 milliseconds

if (state == LOW) {

Serial.println("Motion detected!");

state = HIGH; // update variable state to HIGH

}

}

else {

digitalWrite(led, LOW); // turn LED OFF

delay(500); // delay 200 milliseconds

if (state == HIGH){

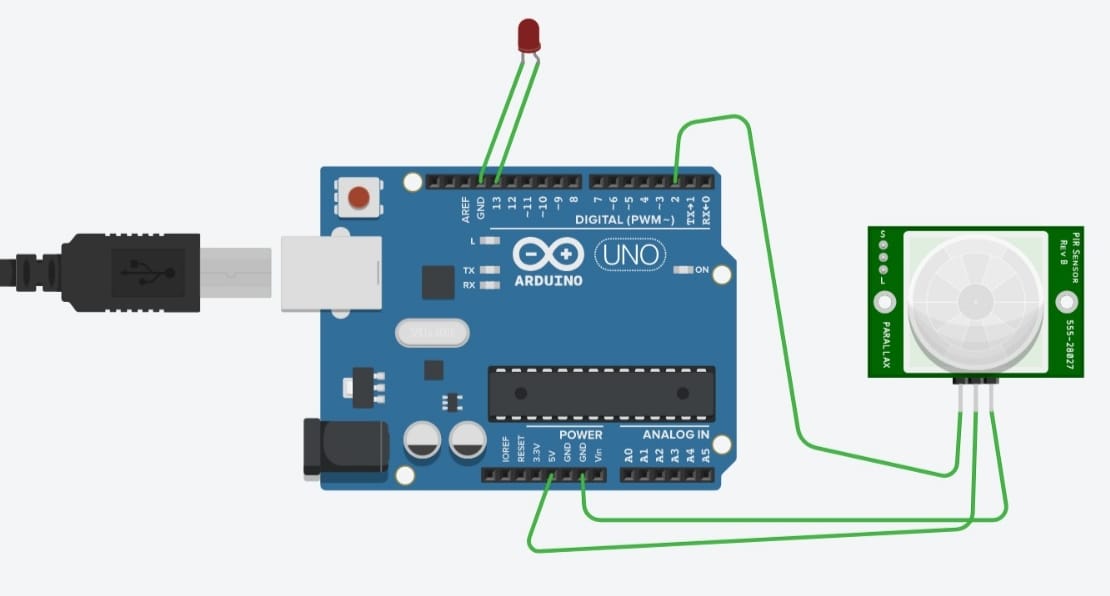
Serial.println("Motion stopped!");

state = LOW; // update variable state to LOW

}

}

}



**3.Demonstrate Ultrasonic distance sensor in Arduino board.**

**Aim:**

To find object distance detection using Arduino board.

**Procedure:**

1. Start the Arduino IDE and enter the code
2. Declare the pin for echo and trigger
3. Define the baud rate for serial transmission
4. The loop function runs over and over again forever
5. Based on the sensor output show the distance between the object and sensor
6. Display the distance value in serial monitor.

**Program:**

const int trigPin = 9;

const int echoPin = 10; // defines variables

long duration;

int distance;

void setup() {

pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output

pinMode(echoPin, INPUT); // Sets the echoPin as an Input

Serial.begin(9600); // put your setup code here, to run once:

}

void loop() {

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

digitalWrite(trigPin, LOW);

delayMicroseconds(2000);

// Sets the trigPin on HIGH state for 10 micro seconds

digitalWrite(trigPin, HIGH);

delayMicroseconds(2000);

digitalWrite(trigPin, LOW);

// Reads the echoPin, returns the sound wave travel time in microseconds

duration = pulseIn(echoPin, HIGH);

// Calculating the distance

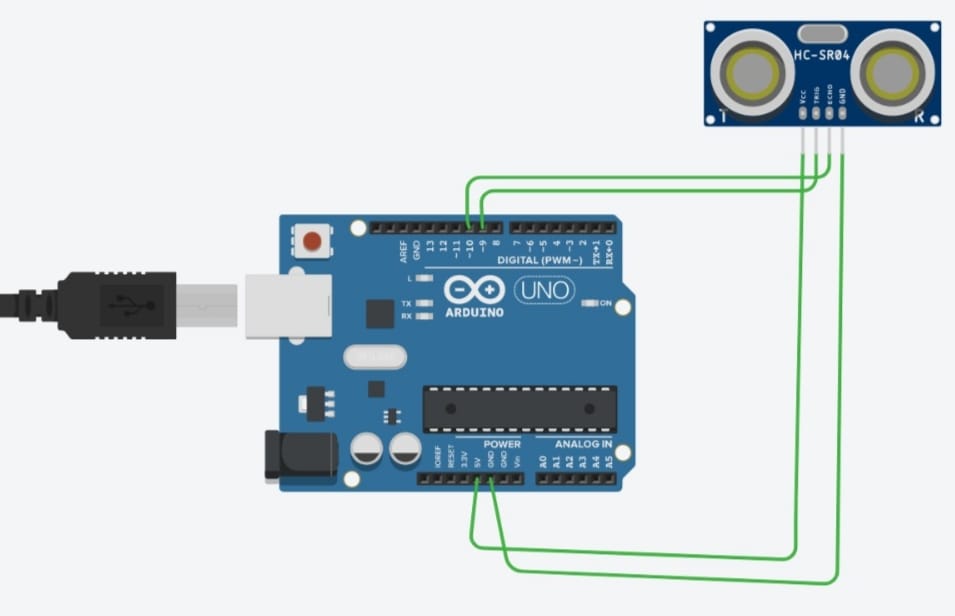
distance= duration\*0.034/2;

// Prints the distance on the Serial Monitor

Serial.print("Distance: ");

Serial.println(distance);

}



**4.Gas Sensor Working.**

**Aim:** Develop an application to measure the gas present inside the working environment.

**Procedure:**

1. Declare the input variables for LED and Gas Sensor.
2. Make the connections between Gas Sensor and Arduino.
3. The loop function runs over and over again forever.
4. Based on the Sensor output, it shows the current Gas value.
5. Display the gas value in the Serial Monitor.

**Program:**

// C++ code

int LED=A1;

const int gas=0;

int MQ2pin=A0;

void setup(){

Serial.begin(9600);

}

void loop(){

float i,MQ2pin;

i=analogRead(MQ2pin);

if(i<=100){

digitalWrite(LED,LOW);

Serial.print(i);

Serial.println(" SMOKE DETECTED");

}

else{

digitalWrite(LED,HIGH);

Serial.println(" Sensor value: ");

Serial.println(i);

}

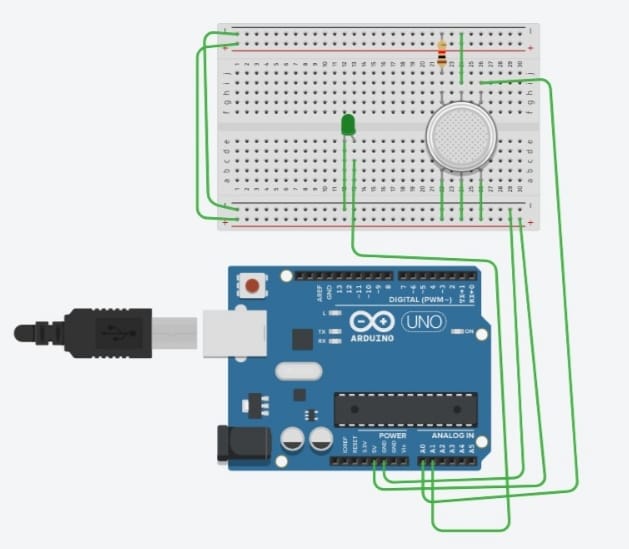
delay(1000);

}

float geti(int pin){

return (analogRead(pin));

}

**5. LED Blinking in Forward and Backward Biases.**

**Aim:**

To make the LED Blink in both Forward and Backward Biases

**Procedure:**

1. Declare the LED variables.
2. Initialize digital pins 2, 3, 4, 5, 6, 7 pins as output in void setup() function.
3. The loop function runs over and over again forever.
4. Execute for loop for performing forward and backward bias.
5. Turn the LED On by making the voltage HIGH
6. Turn the LED Off by making the voltage LOW.

**Program:**

int timer=500;

void setup(){

for (int i=2;i<8;i++){

pinMode(i,OUTPUT);

}

}

void loop(){

for(int i=2;i<8;i++){

digitalWrite(i,HIGH);

delay(timer);

digitalWrite(i,LOW);

}

for(int i=7;i>=2;i--){

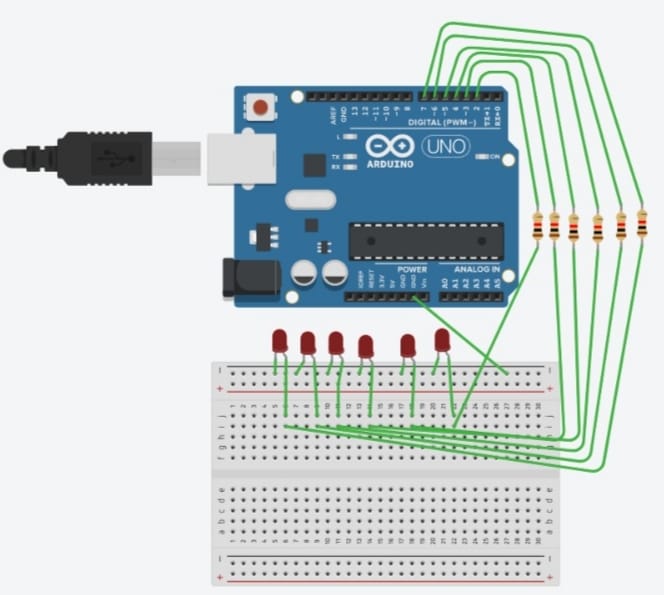
digitalWrite(i,HIGH);

delay(timer);

digitalWrite(i,LOW);

}

}



**6. Temperature Sensor Working.**

**Aim**:

To detect the Temperature and display it using Arduino Board.

**Procedure:**

1. Declare the variables for temperature sensor and LEDs.
2. Define the Baud Rate for Serial Transmission.
3. The loop function runs over and over again forever for getting Sensor data.
4. Based on the Sensor output, its shows the current temperature.
5. Display the Temperature value and category of temperature in LED’s Display.

**Program:**

int baselineTemp = 0;

int celsius = 0;

int fahrenheit = 0;

void setup(){

pinMode(A0, INPUT);

Serial.begin(9600);

pinMode(2, OUTPUT);

pinMode(3, OUTPUT);

pinMode(4, OUTPUT);

}

void loop(){

// set threshold temperature to activate LEDs

baselineTemp = 40;

// measure temperature in Celsius

celsius = map(((analogRead(A0) - 20) \* 3.04), 0, 1023, -40, 125);

// convert to Fahrenheit

fahrenheit = ((celsius \* 9) / 5 + 32);

Serial.print(celsius);

Serial.print(" C, ");

Serial.print(fahrenheit);

Serial.println(" F");

if (celsius < baselineTemp) {

digitalWrite(2, LOW);

digitalWrite(3, LOW);

digitalWrite(4, LOW);

}

if (celsius >= baselineTemp && celsius < baselineTemp + 10) {

digitalWrite(2, HIGH);

digitalWrite(3, LOW);

digitalWrite(4, LOW);

}

if (celsius >= baselineTemp + 10 && celsius < baselineTemp + 20) {

digitalWrite(2, HIGH);

digitalWrite(3, HIGH);

digitalWrite(4, LOW);

}

if (celsius >= baselineTemp + 20 && celsius < baselineTemp + 30) {

digitalWrite(2, HIGH);

digitalWrite(3, HIGH);

digitalWrite(4, HIGH);

}

if (celsius >= baselineTemp + 30) {

digitalWrite(2, HIGH);

digitalWrite(3, HIGH);

digitalWrite(4, HIGH);

}

delay(1000); // Wait for 1000 millisecond(s)

}

