Experiment No. #1

Objective:-

LED blinking with the help of Arduino.

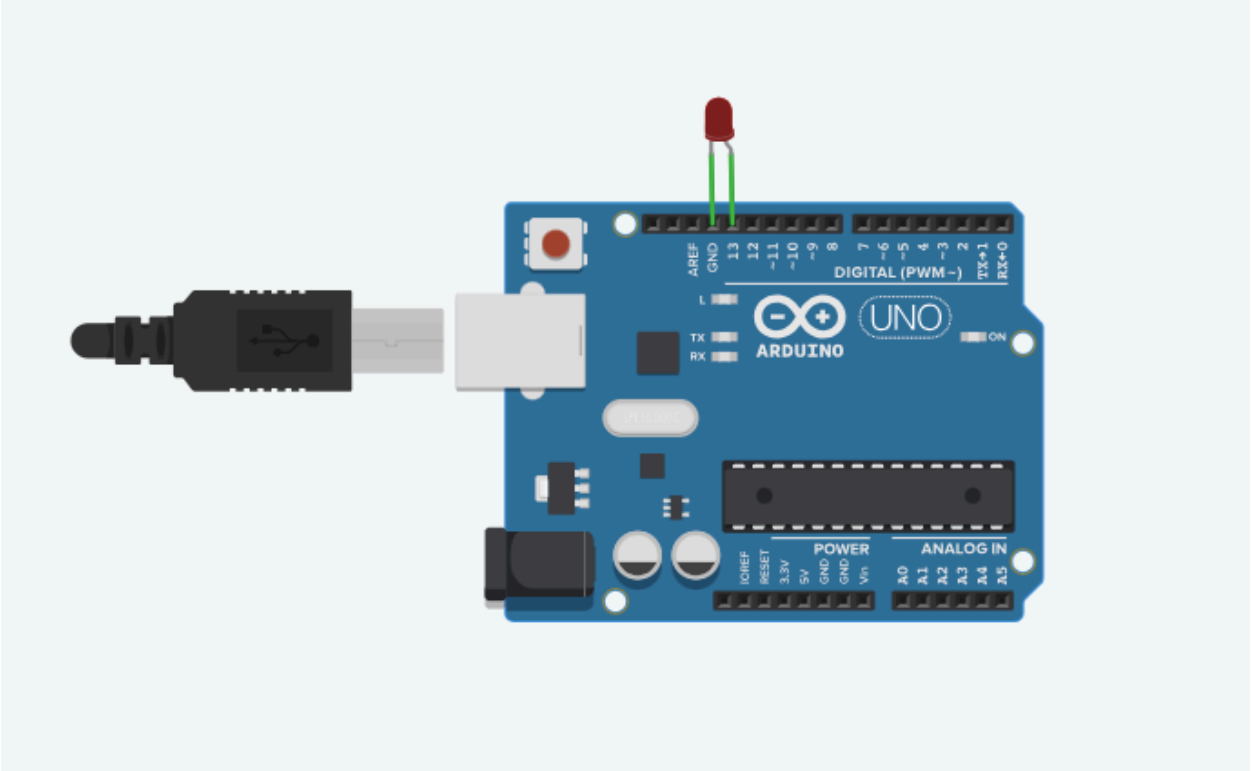
Components Required:-

Arduino UNO
LED

Theory:-

In this project we control the LED using an Arduino UNO microcontroller. First of all we connect the +ive pin of the led with the digital 13 of the Arduino pin and -ive pin of the led connect to the ground pin of the arduino.

BlockDiagram:-



Programming:-

```
void setup() {
  // put your setup code here, to run once:
  pinMode(13, OUTPUT);

}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);

}
```

Program Explain:-

void setup() { … } :- The setup function is called once when the Arduino starts up. Here, we use pinmode to configure digital pin 13 as an OUTPUT, indicating that it will be used to control an output device (the LED).

void loop() { … } :- The loop function is where the main code execution happens, and it runs in a continuous loop.

digitalWrite(13, HIGH) :- This turns the LED on by setting the digital pin 13 to a HIGH voltage(5 volt) level.

delay(1000):- creates a 1 second delay (1000ms = 1s).

digitalWrite(13, LOW):- This turns the LED off by setting the digital pin 13 to a LOW voltage (0 volt) level.

Another delay(1000):- create a 1 second delay (1000ms = 1s).

Experiment No. #2

Objective:-

A digital pin (PWD ~ pin) is used like an analog pin using conditional statement.

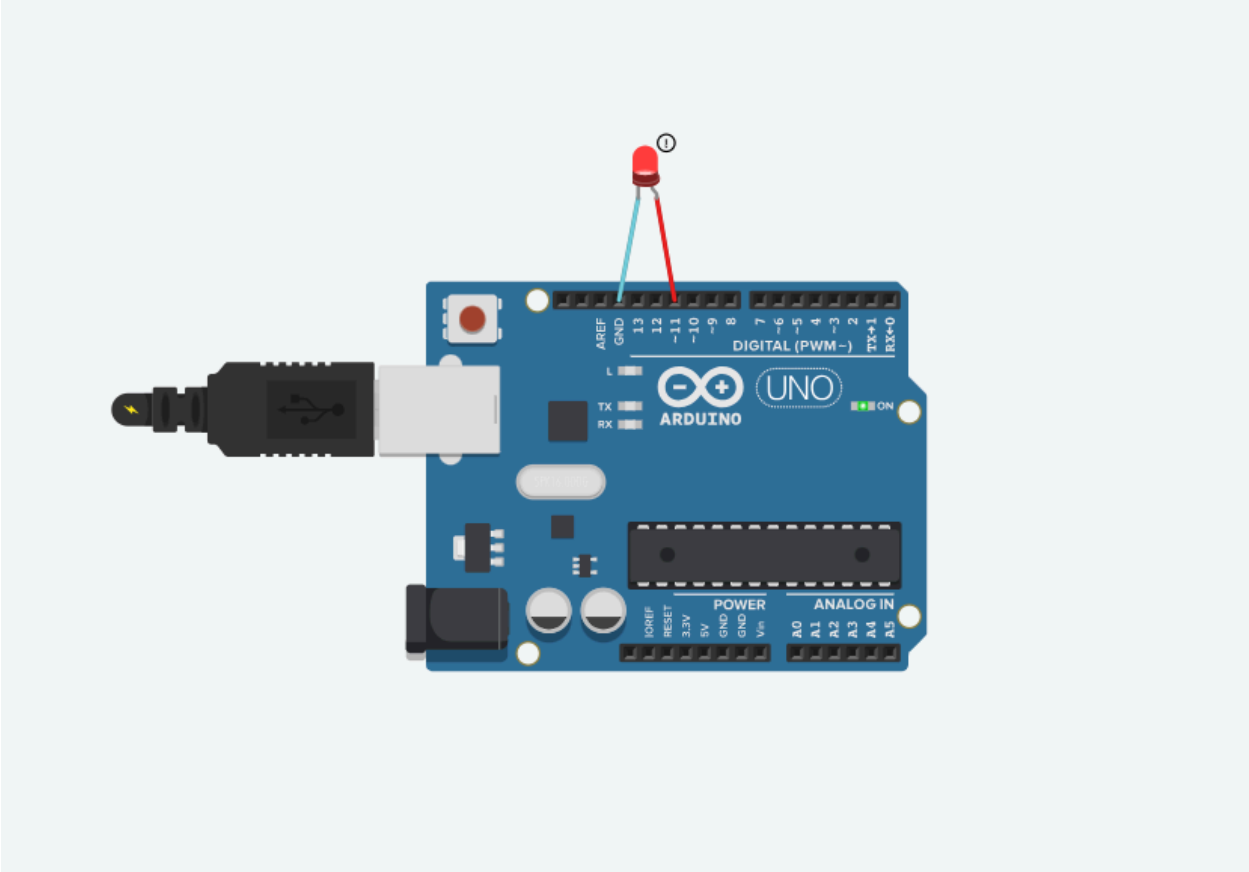
Components Required:-

Arduino UNO
LED

Theory:-

In this project we control the LED-like analog (varying voltage 0v to 5v) using the digital pin (~ PWD pins). First of all we connect the +ive pin of the LED to any pin of the digital pin (~ PWD pins) and -ive pin of the LED pin to the ground of the Arduino.

BlockDiagram:-



Programming:-

```
int ledPin = 11;
int brightness = 0;
int fadeAmount = 5;
void setup() {
  // put your setup code here, to run once:
  pinMode(ledPin, OUTPUT);

}

void loop() {
  // put your main code here, to run repeatedly:
  analogWrite(ledPin, brightness);
  brightness = brightness + fadeAmount;
  if(brightness <= 0 || brightness >= 255){
    fadeAmount = -fadeAmount;
  }
  delay(10);

}
```

Program Explain:-

void setup() { … } :- The setup function is called once when the Arduino starts up. Here, we use pinmode to configure digital pin 13 as an OUTPUT, indicating that it will be used to control an output device (the LED).

void loop() { … } :- The loop function is where the main code execution happens, and it runs in a continuous loop.

analogWrite(ledPin, brightness) :- This turns the LED on the LED depending on the value of the variable brightness. Initially the brightness of the led is 0.And we add the fadeAmount to the brightness the LED glow LOW to HIGH means (0 - 255) range.

Conditional Statement:- In the conditional statement when the brightness is less than and equal to the 0 or greater than equal to the 255 then the variable fadeAmount will be - fadeAmount. Then the LED glow HIGH to LOW means (255 - 0) range and vice versa.

Another delay(10):- create a 1 second delay (10ms = 0.01s).

Experiment No. #3

Objective:-

Control LED using loop in Arduino

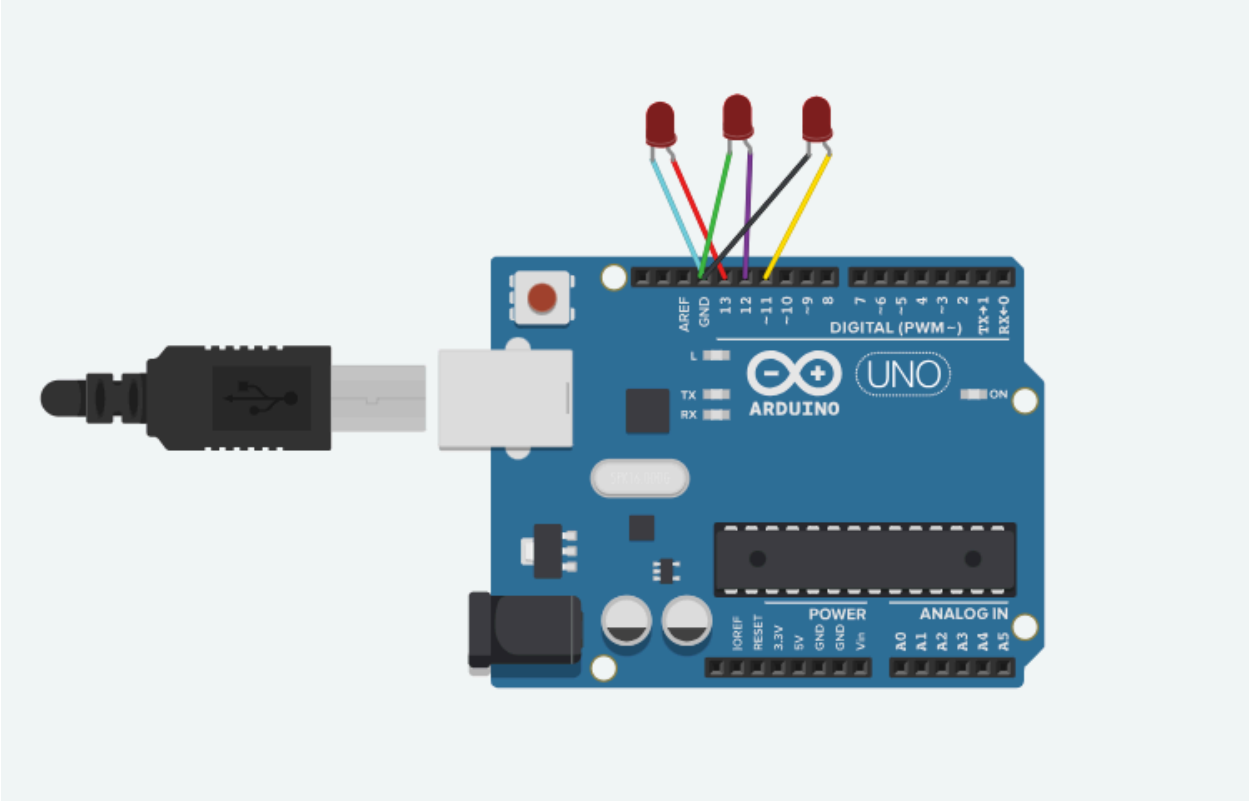
Components Required:-

Arduino UNO
LED

Theory:-

In this project we learn How we control the multiple LEDs using a for loop in Arduino. The main purpose of using the for loop in Arduino is to reduce the time complexity and code complexity and our code runs fast.

BlockDiagram:-



Programming:-

```
void setup() {
  // put your setup code here, to run once:
  for(int thisPin = 11; thisPin <= 13; thisPin++){
    pinMode(thisPin, OUTPUT);
  }

}

void loop() {
  // put your main code here, to run repeatedly:
  for(int thisPin = 11; thisPin <= 13; thisPin++){
    digitalWrite(thisPin , HIGH);
    delay(100);
    digitalWrite(thisPin, LOW);
    delay(100);
  }

}
```

Program Explain:-

void setup() { … } :- The setup function is called once when the Arduino starts up. Here, we use a for loop, the for loop containing pin 11 to 13 and pinmode to configure digital pin 11, 12 and 13 as an OUTPUT, indicating that it will be used to control an output device (the LED).

void loop() { … } :- The loop function is where the main code execution happens, and it runs in a continuous loop.

For loop:- The for loop starts with digital pin 11 and goes to digital pin 13, first digital pin 11 goes to high by digitalWrite function and hang 100ms by delay function and after that the digital pin 11 goes low by digitalWrite function and hang 100ms and after the delay

function the for loop condition the digital pin 12 will start and same process start again whose apply to the digital pin 11 and after the digital pin 13 will start and same process start again whose apply to the digital pin 11 and 12. And this process continuously starts infinite time while we give power to the arduino.

Experiment No. #4

Objective:-

Control Ultrasonic Sensor to measure distance with the help of Arduino UNO.

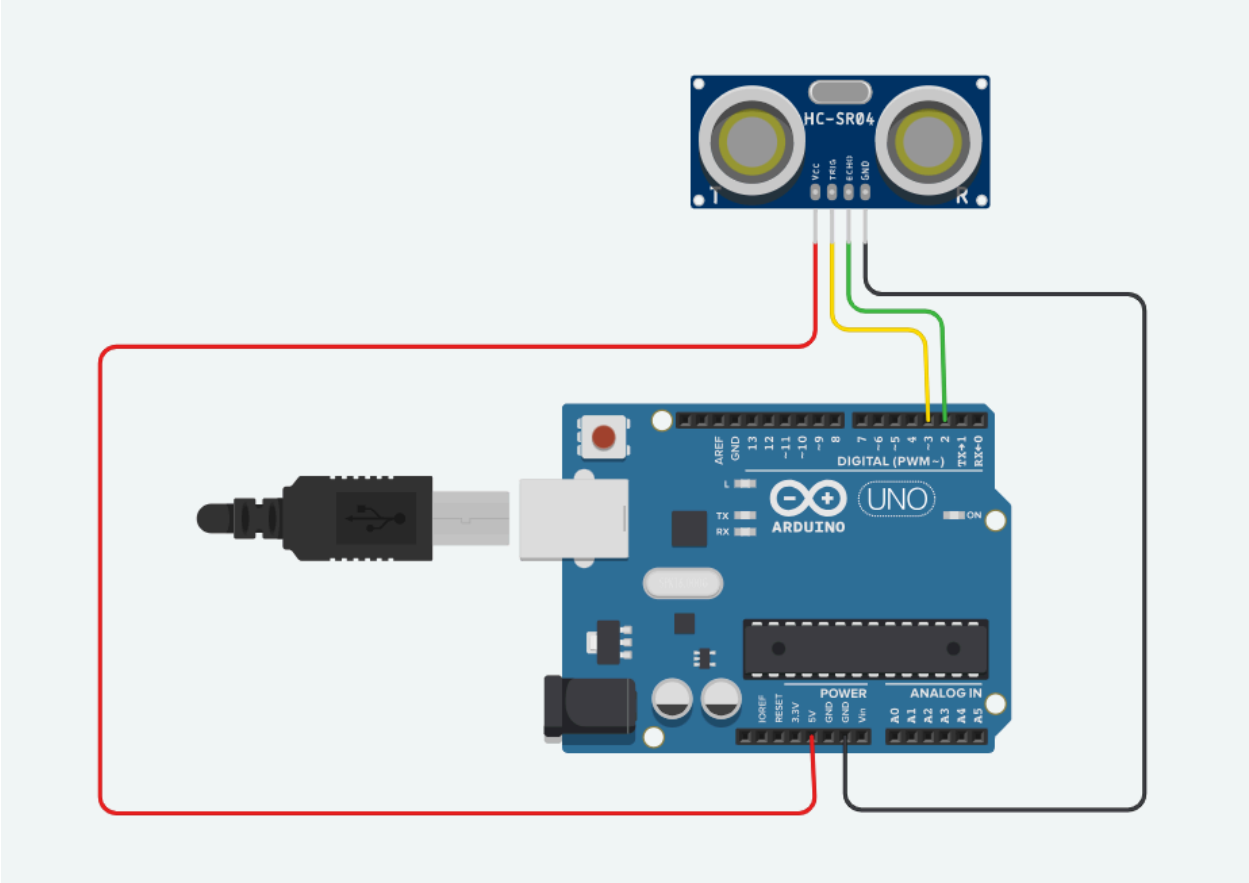
Components Required:-

Arduino UNO
Ultrasonic Sensor
Jumping wires

Theory:-

In this project we use an Ultrasonic Sensor with Arduino. The Ultrasonic Sensor is used to measure the distance or detect the object or find out the object's distance.
The Ultrasonic sensor has 4 pin vcc pin, trig pin, echo pin and ground pin. The vcc pin connect to the 5v pin of the arduino and the ground pin of the ultrasonic sensor connect to the ground pin of the arduino and the echo pin connect to the digital pin 2 of the arduino and the trig pin connect to the digital pin 3 of the arduino.

BlockDiagram:-



Programming:-

```
int trigpin= 3;
int echopin= 2;
long duration;
int distance;
void setup()

{

pinMode(trigpin,OUTPUT);
pinMode(echopin,INPUT);
Serial.begin(9600);

}

void loop()

{

digitalWrite(trigpin,HIGH);
delay(1000);
digitalWrite(trigpin,LOW);
duration=pulseIn(echopin,HIGH);
distance = duration*0.034/2;
Serial.println(distance);
delay(500);

}
```

Program Explain:-

int trigpin = 3; and int echopin = 2; define two constants to represent the trigger pin and echo pin of the ultrasonic sensor, respectively. You need to connect the trigger pin of the sensor to pin 3 and the echo pin to pin 2 on the Arduino or adjust these values if you've connected them to different pins.
long duration; and int distance; declare variables to store the duration of the echo pulse and the calculated distance in centimeters.

In the setup() function:- setup function is used to execute the code for a single time and the variable inside the setup function is fixed for one time. Like Trigpin as a OUTPUT and the echopin as a INPUT

pinMode(trigpin, OUTPUT); sets the trigger pin as an output to send pulses to the ultrasonic sensor.
pinMode(echopin, INPUT); sets the echo pin as an input to read the return pulse from the sensor.
Serial.begin(9600); initializes serial communication at a baud rate of 9600 bits per second, allowing you to output data to the serial monitor.

In the loop() function:-

digitalWrite(trigpin, HIGH); sends a HIGH(ON) pulse to the trigger pin, which triggers the ultrasonic sensor to send out an ultrasonic pulse.
delayMicroseconds(10); waits for 10 microseconds, which is a recommended delay for the HC-SR04 sensor to allow it to settle.
digitalWrite(trigpin, LOW); turns off the trigger pulse.
duration = pulseIn(echopin, HIGH); measures the duration of the echo pulse by timing how long it takes for the echo pin to go HIGH after the trigger pulse. This duration is proportional to the distance between the sensor and the object it bounced the pulse off.
distance = duration * 0.034 / 2; calculates the distance in centimeters based on the duration of the echo pulse. The 0.034 constant is used because the speed of sound in air is approximately 343 meters per second, or 0.034 centimeters per millisecond. Dividing by 2 accounts for the round trip of the sound pulse.
Serial.println(distance); prints the calculate distance to the serial monitor.

delay(500); introduces a 500-millisecond delay before taking the next distance measurement to prevent rapid, continuous measurements and allow for a more stable reading.

Experiment No. #5

Objective:-

LED Controlled by user using Serial Monitor.

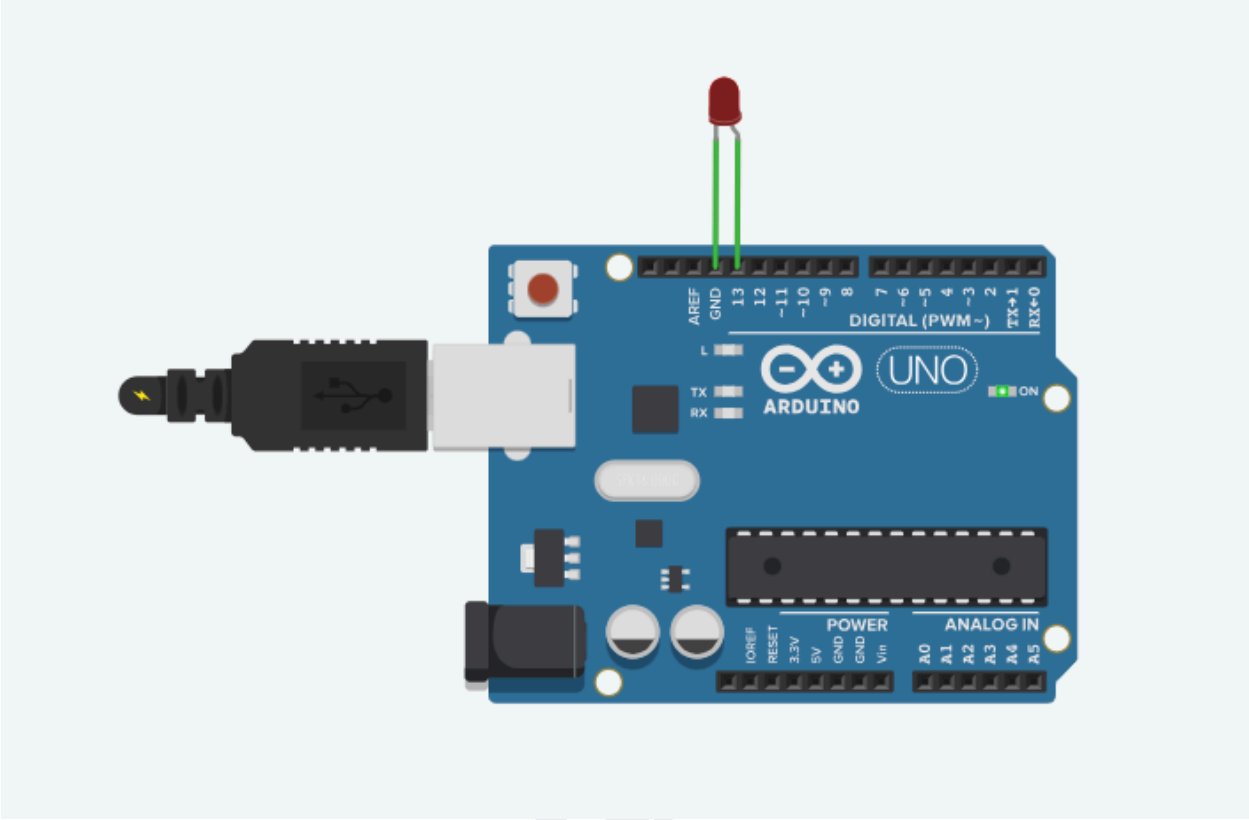
Components Required:-

Arduino UNO
LED

Theory:-

In this project, we control the LED by the user using a serial monitor. First of all we connect the +ive led pin with any digital pin of the arduino and -ive pin of the led pin to the ground pin of the arduino.

BlockDiagram:-



Programming:-

```
int data;
void setup() {
  // put your setup code here, to run once:
  pinMode(13, OUTPUT);
  Serial.begin(9600);

}

void loop() {
  // put your main code here, to run repeatedly:
  if(Serial.available()!=0){
     data = Serial.read();
  }
  if(data == '1'){
    digitalWrite(13, HIGH);
    Serial.println("ON");
  }else if(data == '2'){
    digitalWrite(13, LOW);
    Serial.println("OFF");
  }
  delay(100);

}
```

Program Explain:-

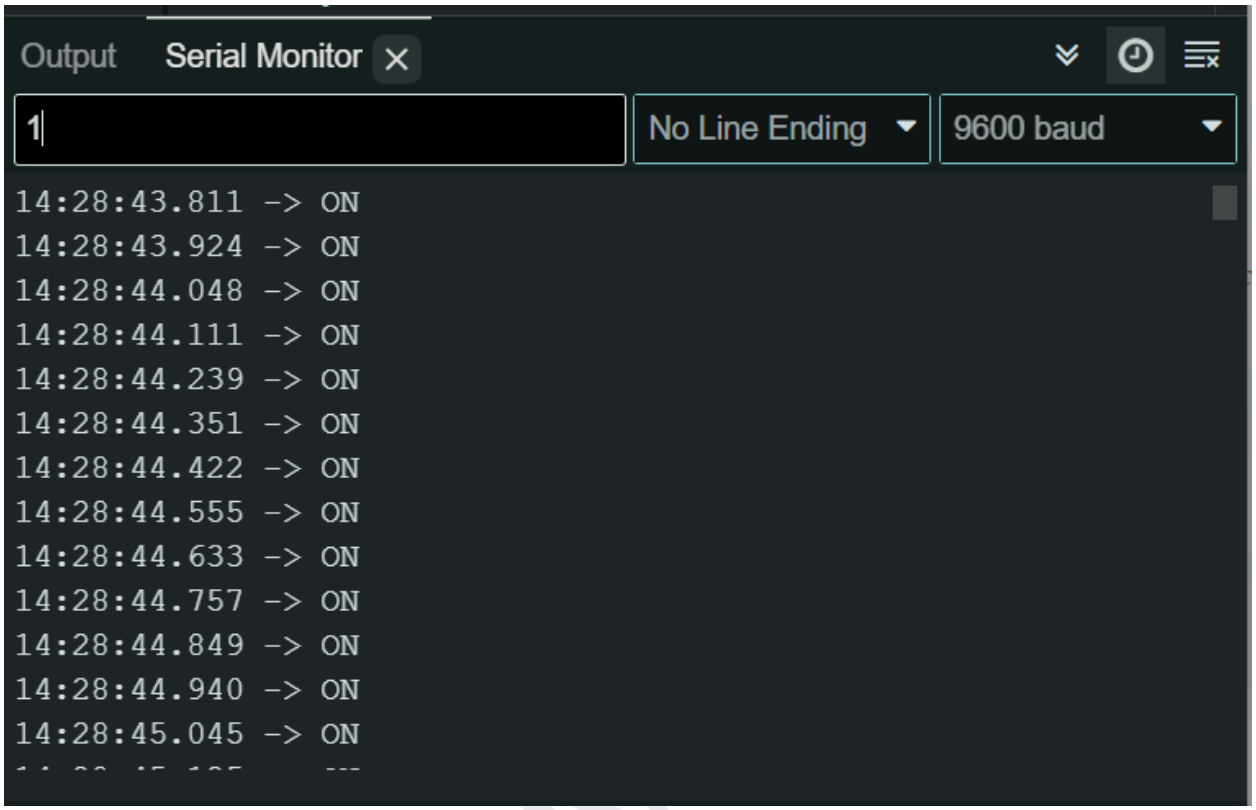
int data; This integer variable data is used to store the data received from the serial communication. It will be used to determine whether to turn the LED on or off.

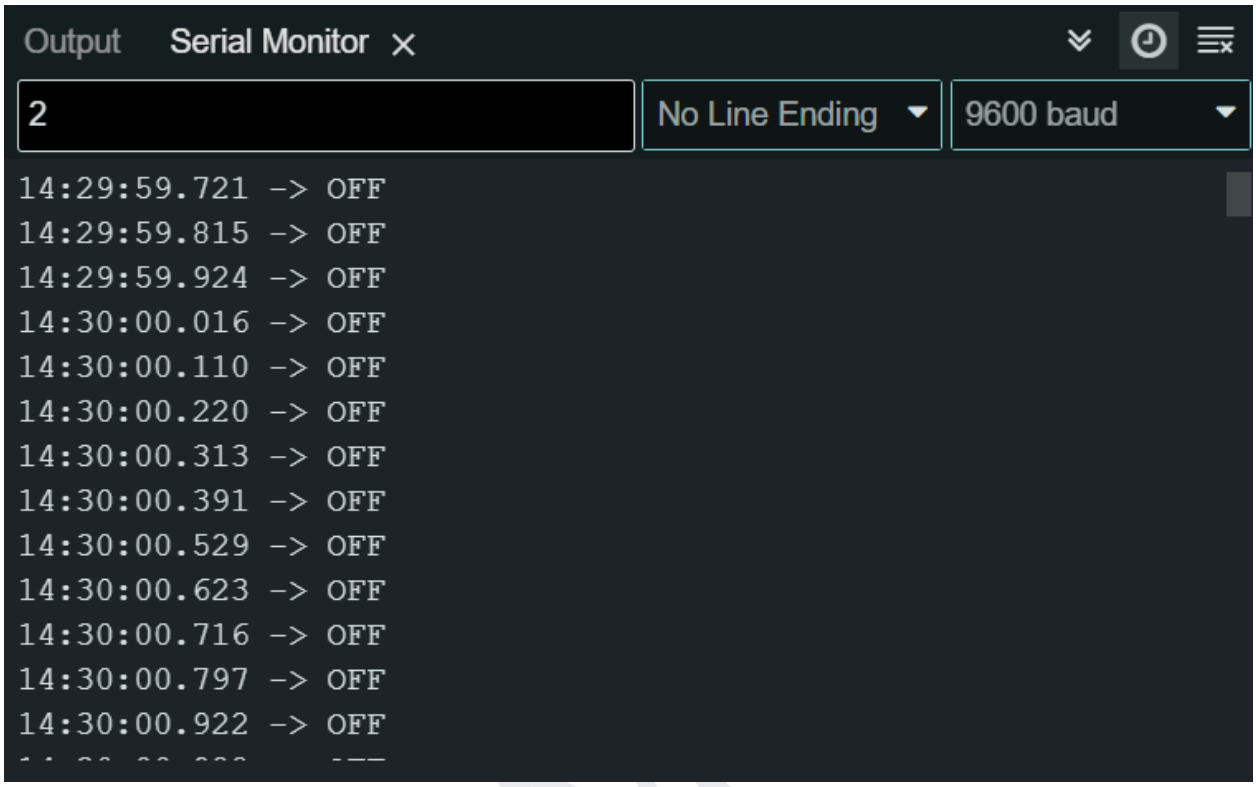
In the setup() function:- setup function is used to execute the code for a single time and the variable inside the setup function is fixed for one time. Like 13 pin of the arduino as a OUTPUT

pinMode(13, OUTPUT); configures digital pin 13 as an OUTPUT, indicating that it will be used to control an external device, in this case, an LED.
Serial.begin(9600); initializes serial communication with a baud rate of 9600 bits per second, which should match the baud rate used in the software or device sending data to the Arduino.

In the loop() function:-

if(Serial.available() !=0){ … } : this checks if there is data available in the serial buffer. If data is available, it is read and stored in the data variable using Serial.read(). This allows the Arduino to receive commands from an external source via serial communication (e.g., from a computer over a USB connection).
if (data == '1'){ … } else if (data == '2') { … } : These conditional statements check the value of data. If data is equal to '1' , the code sets digital pin 13 (connected to an LED) to HIGH, turning the LED on, and it sends "ON" to the serial monitor. If data is equal to '2', the code sets pin 13 to LOW, turning the LED off, and it sends "OFF" to the serial monitor. These actions are based on the commands received from the serial input.
delay(100); This delay is added to prevent rapid processing of data. It pauses for 100 milliseconds between iterations of the loop() to ensure that the Arduino doesn't act on the same data too quickly.


Output
Serial Monitor
1
No Line Ending
9600 baud
14:28:43.811 -> ON
14:28:43.924 -> ON
14:28:44.048 -> ON
14:28:44.111 -> ON
14:28:44.239 -> ON
14:28:44.351 -> ON
14:28:44.422 -> ON
14:28:44.555 -> ON
14:28:44.633 -> ON
14:28:44.757 -> ON
14:28:44.849 -> ON
14:28:44.940 -> ON
14:28:45.045 -> ON



Experiment No. #6

Objective:-

Measure Temperature and Humidity using DHT-11 Sensor using Arduino.

Components Required:-

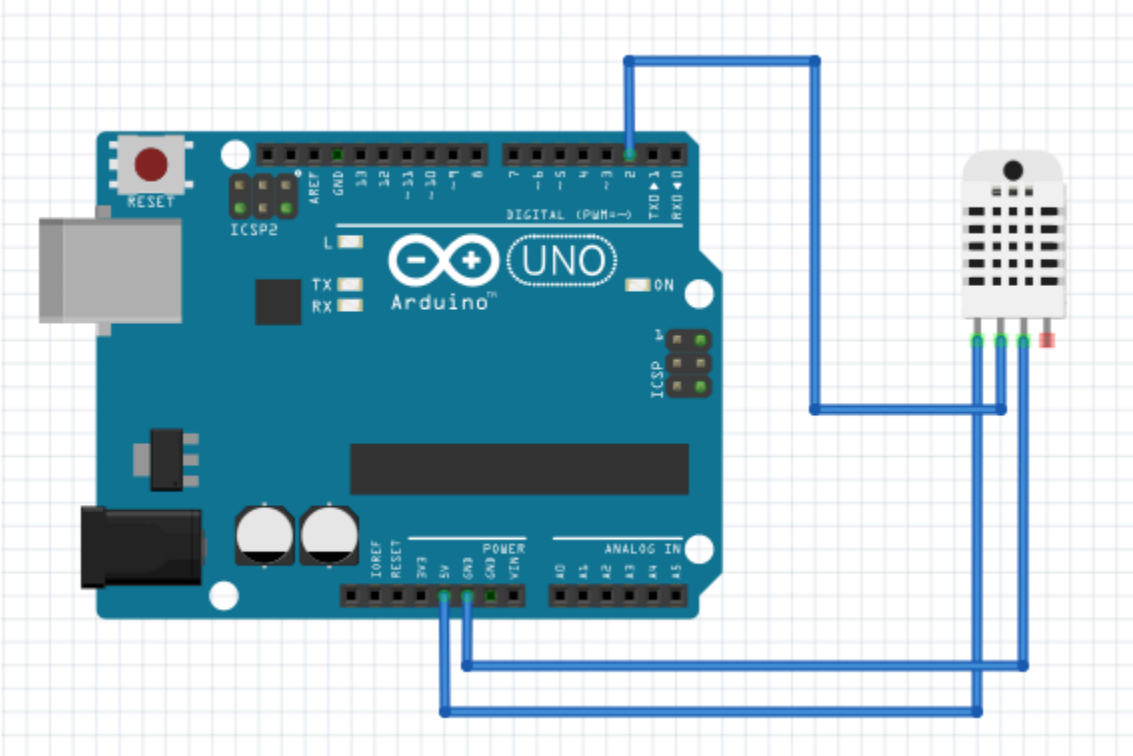
Arduino UNO

DHT-11 Sensor

Theory:-

In this project, we measure the temperature and humidity of the environment using a temperature and humidity sensor (DHT-11 or DHT-22) using Arduino. First we connect the 5v pin of the sensor to the Arduino 5v pin and ground pin of the sensor to the ground pin of the arduino and the and the data pin of the sensor connect to the digital pin(2 or whatever you want) of the arduino.

BlockDiagram:-



Programming:-

```
#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT11


float tempF;
float tempC;
int humidity;

DHT TH(DHTPIN, DHTTYPE);

int setTime = 500;
int del = 1000;

void setup() {
  // put your setup code here, to run once:
Serial.begin(9600);
TH.begin();
delay(setTime);
}

void loop() {
  // put your main code here, to run repeatedly:
tempC = TH.readTemperature();
tempF = TH.readTemperature(true);
humidity = TH.readHumidity();

Serial.print(tempC);
Serial.print(" , ");
Serial.println(humidity);

delay(del);
}
```

Program Explain:-

First of all we add the DHT-11 Sensor library to our arduino code and then we define the pin in the arduino we connect the sensor data pin to the define pin and then we define our sensor type which means which sensor we used like DHT-11 or DHT-22. And after that we define some variable like temp and humidity and after that we make object of the DHT-11 sensor

In the setup() function:- setup function is used to execute the code for a single time and the variable inside the setup function is fixed for one time.

In the loop() function:- In loop() function the all code executes infinite time while we give the power to the arduino. The readTemperature() function reads the data using DHT-11 sensor and stores a value in variable tempC. Same for readHumidity() function and store the value in variable in humidity. And then we print these values on the serial monitor.

Output:-

```
12:31:20.432 -> Temperature is: 30.90 , Humidity is: 27
12:31:21.468 -> Temperature is: 30.90 , Humidity is: 27
12:31:22.462 -> Temperature is: 30.90 , Humidity is: 27
12:31:23.515 -> Temperature is: 30.90 , Humidity is: 27
12:31:24.519 -> Temperature is: 30.90 , Humidity is: 27
12:31:25.534 -> Temperature is: 30.90 , Humidity is: 27
12:31:26.546 -> Temperature is: 30.90 , Humidity is: 27
12:31:27.575 -> Temperature is: 30.90 , Humidity is: 27
12:31:28.558 -> Temperature is: 30.90 , Humidity is: 27
12:31:29.595 -> Temperature is: 30.90 , Humidity is: 27
12:31:30.588 -> Temperature is: 30.90 , Humidity is: 27
```

Experiment No. #7

Objective:-

Motion Detector Sensor or PIR Sensor using Arduino

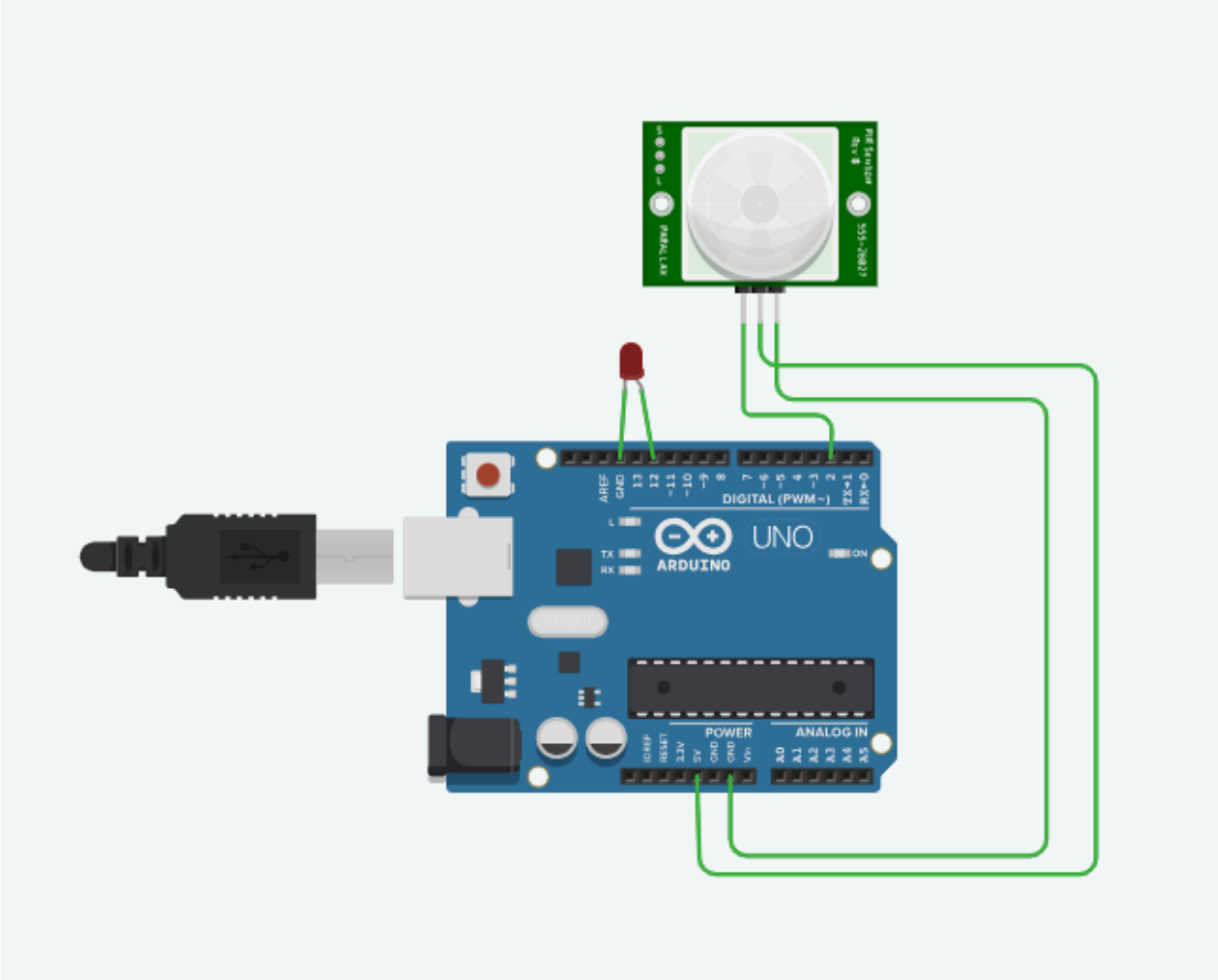
Components Required:-

Arduino UNO
PIR Sensor
LED

Theory:-

In this project, we detect the motion using Motion Sensor or PIR Sensor with arduino. First we connect the positive pin of the sensor to the 5v pin of the arduino and the negative pin of the sensor connects to the ground pin of the arduino and the signal pin of the sensor to any digital pin of the arduino.

BlockDiagram:-



Programming:-

```
int pirPin = 2;
int ledPin = 12;
int data;
void setup() {
  // put your setup code here, to run once:
  pinMode(pirPin, INPUT);
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);

}

void loop() {
  // put your main code here, to run repeatedly:
  data = digitalRead(pirPin);
  if(data == HIGH){
    Serial.println("Motion Detected");
    digitalWrite(ledPin, HIGH);
  }else{
    Serial.println("Not Motion Detected");
    digitalWrite(ledPin, LOW);
  }
  delay(500);

}
```

Program Explain:-

First of all we initialize the data pin of the sensor to the arduino digital pin 2 and the led pin is connected to the digital pin to the arduino 12.

In the setup() function:- setup function is used to execute the code for a single time and the variable inside the setup function is fixed for one time.

In loop() function:- The code execute infinite time while we give power to our arduino and the digitalRead() function inside the loop function read while motion is present in front of the PIR sensor and store the data in the data variable and we apply the condition if data == HIGH then print motion detected and led is high else not motion detected and led is low.

Output:-

```
13:03:31.597 -> Not Motion Detected
13:03:32.243 -> Not Motion Detected
13:03:32.604 -> Not Motion Detected
13:03:33.076 -> Not Motion Detected
13:03:33.580 -> Not Motion Detected
13:03:34.097 -> Not Motion Detected
13:03:34.613 -> Motion Detected
13:03:35.103 -> Motion Detected
13:03:35.591 -> Motion Detected
13:03:36.107 -> Motion Detected
13:03:36.614 -> Motion Detected
13:03:37.100 -> Motion Detected
13:03:37.582 -> Motion Detected
13:03:38.110 -> Motion Detected
```

Experiment No. #8

Objective:-

Servo motor control by Arduino UNO

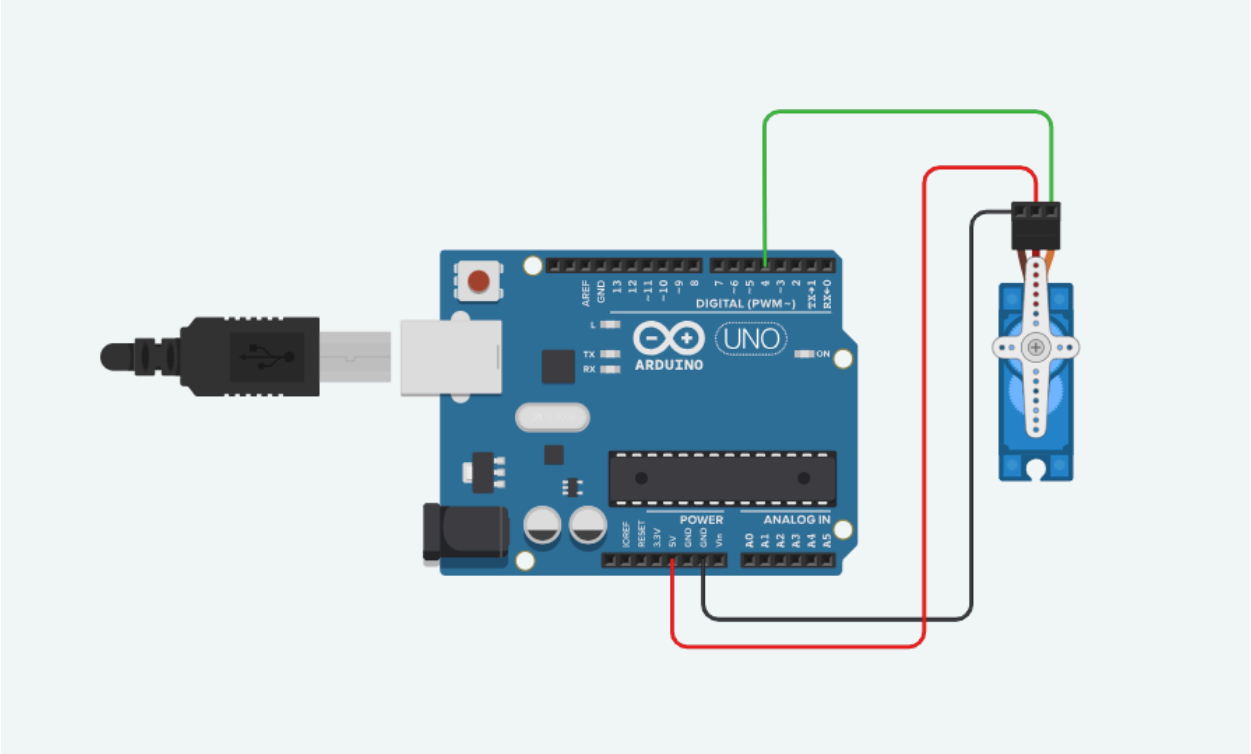
Components Required:-

Arduino UNO
Servo motor

Theory:-

In this project, We control the servo motor sensor using arduino the servo motor have three pins
VCC, GND and SIGNAL. The vcc pin of the servo connects to the 5v pin of the arduino and the ground pin of the servo motor connects to the ground pin of the arduino and the signal pin of the servo motor connects to any digital pin of the arduino.

BlockDiagram:-



Programming:-

```
#include <Servo.h>

Servo myservo;

int angle = 0;

void setup() {
  Serial.begin(9600);
  myservo.attach(4);
}

void loop() {
  myservo.write(180);
  Serial.println("Angle move 180 degree");
  delay(1000);

  myservo.write(0);
  Serial.println("Angle move 0 degree");
  delay(1000);
}
```

Program Explain:-

First of all we define the library of the servo motor to control the servo motor and after that we make the object of the servo library and after that we store a 0 angle in a variable.

In the setup() function:- setup function is used to execute the code for a single time and the variable inside the setup function is fixed for one time. And myservo.attach is used to connect the signal pin to the arduino any digital pin for signal.

In loop() function:- In the loop function first of all we write the angle 180 using the myservo.write function and after that we print the angle using serial.println function and then give one second delay. And after that we write the angle 0 using the myservo.write function and after that we print the angle using serial.println function and then give one second delay.

Output:-

```
10:59:01.431 -> Angle move 180 degree
10:59:02.448 -> Angle move 0 degree
10:59:03.426 -> Angle move 180 degree
10:59:04.457 -> Angle move 0 degree
10:59:05.465 -> Angle move 180 degree
10:59:06.450 -> Angle move 0 degree
10:59:07.467 -> Angle move 180 degree
10:59:08.438 -> Angle move 0 degree
10:59:09.451 -> Angle move 180 degree
10:59:10.462 -> Angle move 0 degree
10:59:11.453 -> Angle move 180 degree
10:59:12.466 -> Angle move 0 degree
10:59:13.435 -> Angle move 180 degree
10:59:14.449 -> Angle move 0 degree
10:59:15.474 -> Angle move 180 degree
10:59:16.474 -> Angle move 0 degree
10:59:17.479 -> Angle move 180 degree
10:59:18.445 -> Angle move 0 degree
```

Experiment No. #9

Objective:-

Measure Air quality using MQ2 gas sensor and Arduino UNO

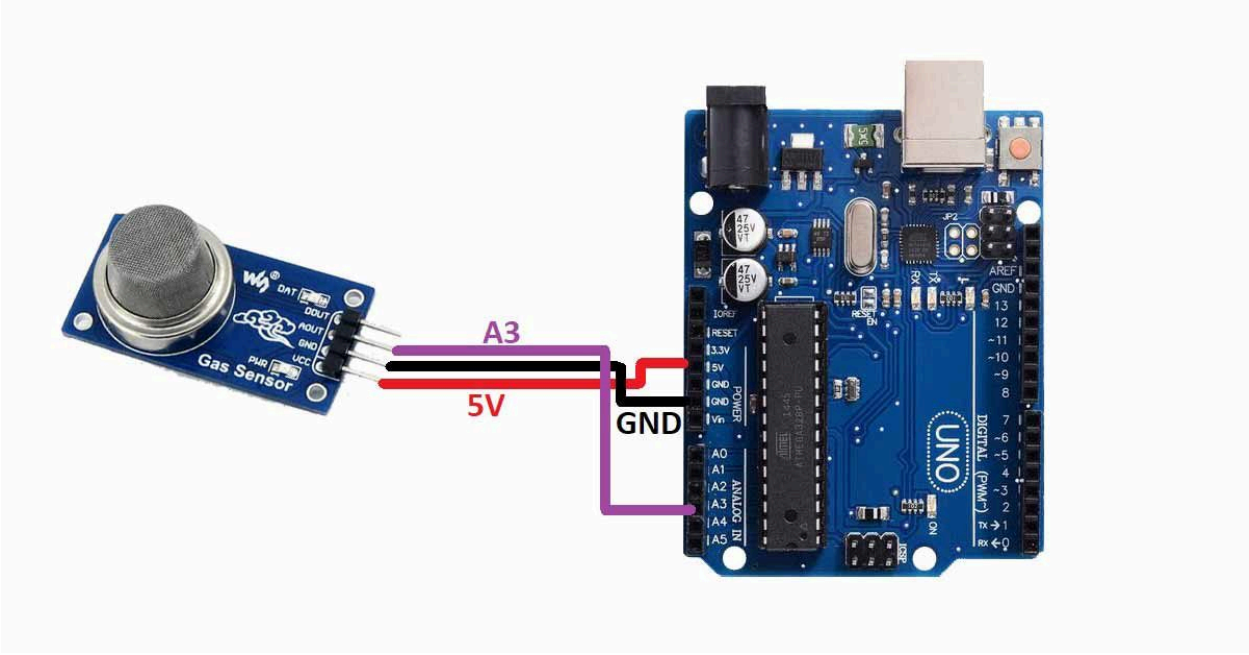
Components Required:-

Arduino UNO
MQ2 gas sensor

Theory:-

In this project, We measure the air quality using MQ2 gas sensor and arduino. The gas sensor has 4 pins. The vcc pin of the gas sensor connect to the 5v pin of the arduino and the ground pin of the gas sensor connect to the ground pin of the arduino and the gas sensor have 2 signal pin first analog pin and second digital pin we can use both pins individually like we can use analog pin or digital pin at a time.

BlockDiagram:-



Programming:-

```
int gasPin = A3;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);

}

void loop() {
  // put your main code here, to run repeatedly:
  int data = analogRead(gasPin);
  Serial.print("Gas Detected: ");
  Serial.println(data);
  delay(1000);

}
```

Program Explain:-

First of all we define the gas sensor pin to a variable. You can use an analog pin or digital pin but in my case I am using the analog pin of the gas sensor.

In the setup() function:- setup function is used to execute the code for a single time and the variable inside the setup function is fixed for one time.

In loop() function:- In the loop function we define a local variable and store the gasPin data using analogRead() function and after that we print a message and then we print the data of the sensor using Serial.println() function and then we give one second delay using delay() function.

Output:-

```
11:37:29.171 -> Gas Detected: 330
11:37:30.193 -> Gas Detected: 324
11:37:31.185 -> Gas Detected: 318
11:37:32.205 -> Gas Detected: 317
11:37:33.216 -> Gas Detected: 316
11:37:34.176 -> Gas Detected: 310
11:37:35.224 -> Gas Detected: 306
11:37:36.212 -> Gas Detected: 303
11:37:37.190 -> Gas Detected: 299
11:37:38.190 -> Gas Detected: 296
11:37:39.222 -> Gas Detected: 293
11:37:40.184 -> Gas Detected: 290
11:37:41.230 -> Gas Detected: 288
11:37:42.187 -> Gas Detected: 285
11:37:43.202 -> Gas Detected: 282
11:37:44.229 -> Gas Detected: 280
11:37:45.232 -> Gas Detected: 278
```

Experiment No. #10

Objective:-

Print Date and Time using RTC Sensor with Arduino UNO

Components Required:-

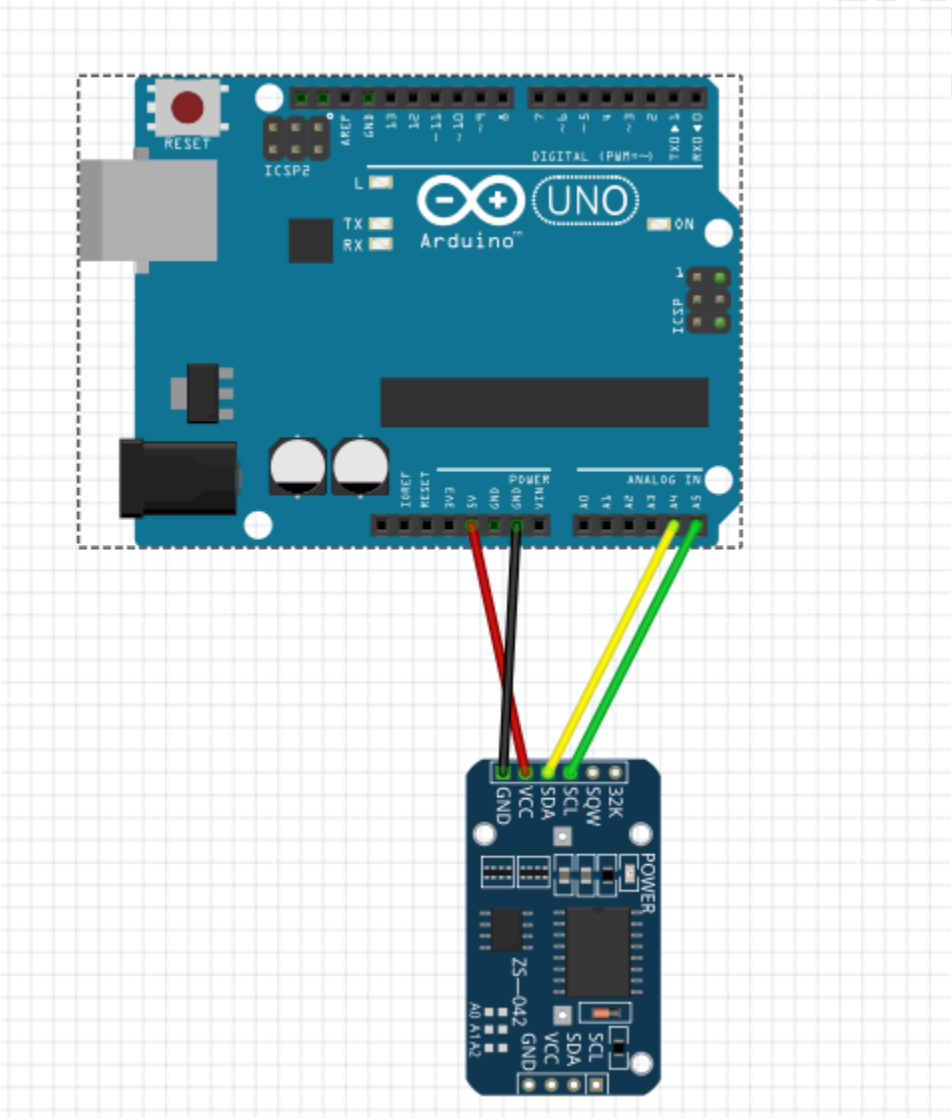
Arduino UNO

RTC Sensor

Theory:-

In this project, we print the data and time in the serial monitor. The RTC sensor vcc pin connect ot the 5v pin of the arduino and the ground pin of the sensor connect to the ground pin of the arduino, the SDA pin of the sensor connect to the A4 pin of the arduino and the SCL pin of the sensor is connect to the A5 pin of the arduino.

BlockDiagram:-



Programming:-

```
#include <RTClib.h>
RTC_DS3231 rtc;
char t[32];
void setup()
{
  Serial.begin(9600);
  rtc.begin();
}

void loop() {
  // put your main code here, to run repeatedly:
  DateTime now = rtc.now();

  Serial.print("Date/Time: ");
  Serial.print(now.year(), DEC); // DEC is used to print date and time in decimal number
  Serial.print('/');
  Serial.print(now.month(), DEC);
  Serial.print('/');
  Serial.print(now.day(), DEC);
  Serial.print(' ');
  Serial.print(now.hour(), DEC);
  Serial.print(':');
  Serial.print(now.minute(), DEC);
  Serial.print(':');
  Serial.print(now.second(), DEC);
  Serial.println();

  delay(1000);

}
```

Program Explain:-

First of all we define the TRClib library and then create object of this library

In the setup() function:- setup function is used to execute the code for a single time and the variable inside the setup function is fixed for one time.

In loop() function:- In the loop function we define a variable, in this variable we store the data of the rtc.now() function and we use this variable value to print the year, month, day, hour, minute and second.

Output:-

```
2023/10/31 14:25:30
2023/10/31 14:25:31
2023/10/31 14:25:32
2023/10/31 14:25:33
2023/10/31 14:25:34
2023/10/31 14:25:35
2023/10/31 14:25:36
2023/10/31 14:25:37
2023/10/31 14:25:38
2023/10/31 14:25:39
2023/10/31 14:25:40
2023/10/31 14:25:41
2023/10/31 14:25:42
2023/10/31 14:25:43
```


Experiment No. #11

Objective:-

Control Sound Sensor using Arduino

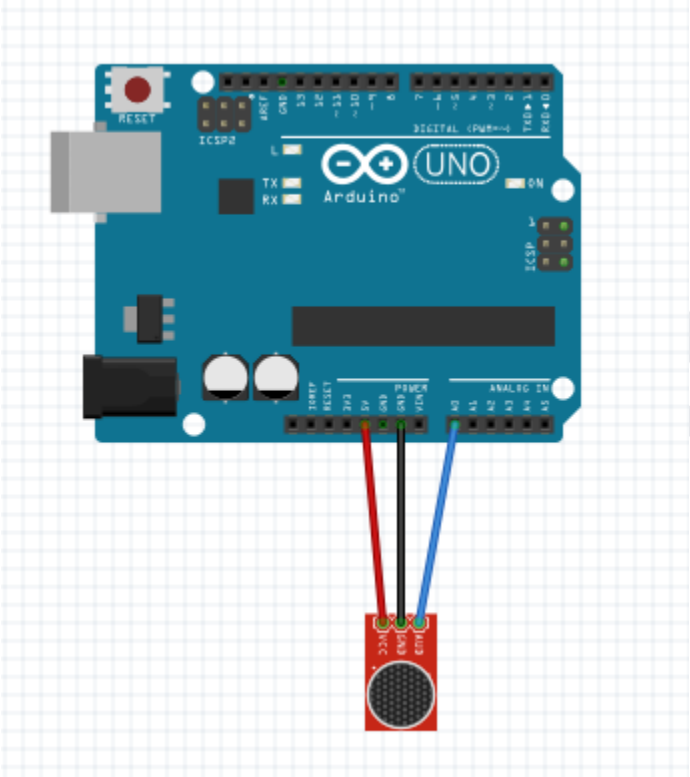
Components Required:-

Arduino UNO
Sound Sensor

Theory:-

In this project, We control the sound sensor with the help of an Arduino UNO. The Power pin of the sensor connects to the 5v of the arduino and the ground pin of the sensor connects to the ground pin of the arduino and the signal pin of the sensor connects to any Analog pin of the arduino.

BlockDiagram:-



Programming:-

```ino
int sensor = A0; // Connected to digital output of KY-038 sound sensor module
int led = 8; // Connected to postive of led
boolean is_on = false; //To determine/track if led is on or off

void setup() {
  pinMode(sensor, INPUT); //Setting the pin to input for reading data
  pinMode(led, OUTPUT); //Setting the pin to output for turning the led on/off

}

void loop() {

  int data = digitalRead(sensor); //Reading data from sensor and storing in variable

  if (data == 1) { // 1 is sent from sensor when loud noise is detected
    if (is_on == true) { // If led is on then turn it off
      digitalWrite(led, LOW);
      is_on = false;
    }
    else { // else if led is off then turn it on
      digitalWrite(led, HIGH);
      is_on = true;
    }
  }
}
```

Program Explain:-

First of all we connect the signal pin of the sensor to the arduino A0 analog pin and after that we take a led when the sensor detects sound the led is glow, and we take a variable boolean is_on by default false.

In the setup() function:- setup function is used to execute the code for a single time and the variable inside the setup function is fixed for one time.

In loop() function:- In the loop function, we take a variable the name of the variable is data when the sound sensor detect the sound then the sensor store the data in the data variable and then we apply the condition like if data is double equal 1 and the flag is true then led glow else the led not glow.

Experiment No. #12

Objective:-

Ultrasonic Sensor with buzzer control by Arduio

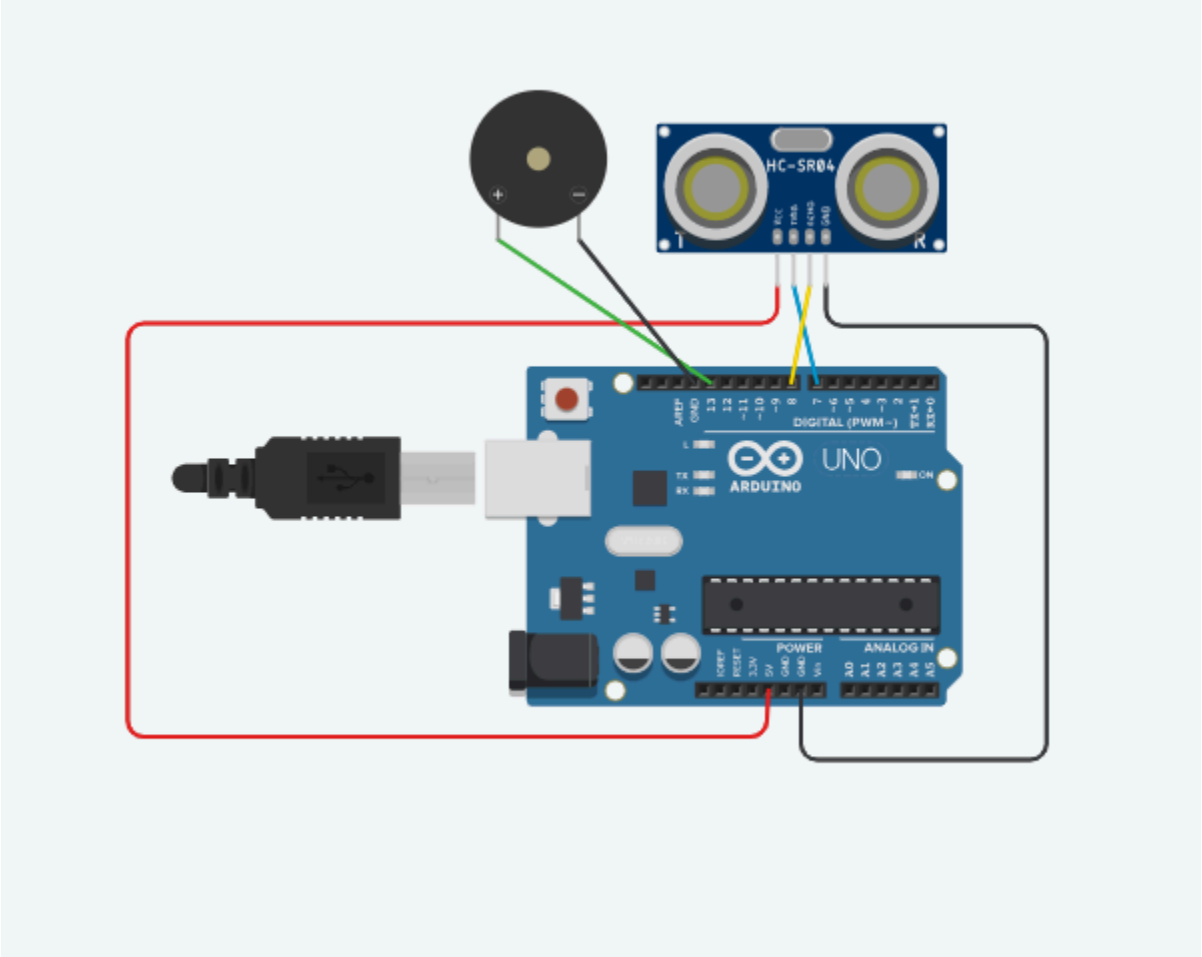
Components Required:-

Arduino UNO
Ultrasonic Sensor
buzzer

Theory:-

In this project, we make a security sensor using ultrasonic sensor and buzzer control by arduino, first of all we connect the trigpin of the sensor to the digital pin 7 of the arduino and the echopin of the sensor connect to the digital pin 8 of the sensor and the ground pin of the sensor to connect to the ground pin of the arduino and the vcc pin of the sensor connect to the 5v pin of the arduino and the buzzer +ive pin is connected to the digital pin 13 of the arduino and -ive pin of the buzzer is connected to the ground.

BlockDiagram:-



Programming:-

```
const int trigpin= 7;
const int echopin= 8;
buzzerPin = 13;
long duration;
int distance;
void setup()
{
pinMode(trigpin,OUTPUT);
pinMode(echopin,INPUT);
pinMode(buzzerPin, OUTPUT);
Serial.begin(9600);

}
void loop()
{

digitalWrite(trigpin,HIGH);
delay(1000);
digitalWrite(trigpin,LOW);
duration=pulseIn(echopin,HIGH);
distance = duration*0.034/2;
Serial.println(distance);
if(distance <= 50){
  digitalWrite(buzzerPin, HIGH);
}else{
  digitalWrite(buzzerPin, LOW);
}
}
```

Program Explain:-

First of all we define all pins of the sensor and the buzzer pin in our IDE.

In the setup() function:- setup function is used to execute the code for a single time and the variable inside the setup function is fixed for one time. Like Trigpin as a OUTPUT and the echopin as a INPUT

pinMode(trigpin, OUTPUT); sets the trigger pin as an output to send pulses to the ultrasonic sensor.
pinMode(echopin, INPUT); sets the echo pin as an input to read the return pulse from the sensor.
Serial.begin(9600); initializes serial communication at a baud rate of 9600 bits per second, allowing you to output data to the serial monitor.

In the loop() function:-

digitalWrite(trigpin, HIGH); sends a HIGH(ON) pulse to the trigger pin, which triggers the ultrasonic sensor to send out an ultrasonic pulse.
delayMicroseconds(10); waits for 10 microseconds, which is a recommended delay for the HC-SR04 sensor to allow it to settle.
digitalWrite(trigpin, LOW); turns off the trigger pulse.
duration = pulseIn(echopin, HIGH); measures the duration of the echo pulse by timing how long it takes for the echo pin to go HIGH after the trigger pulse. This duration is proportional to the distance between the sensor and the object it bounced the pulse off.
distance = duration * 0.034 / 2; calculates the distance in centimeters based on the duration of the echo pulse. The 0.034 constant is used because the speed of sound in air is approximately 343 meters per second, or 0.034 centimeters per millisecond. Dividing by 2 accounts for the round trip of the sound pulse.
Serial.println(distance); prints the calculated distance to the serial monitor.
delay(500); introduces a 500-millisecond delay before taking the next distance measurement to prevent rapid, continuous measurements and allow for a more stable reading.

When the distance is low and equal to the 50cm then the buzzer is beep and if the distance is more than 50cm then the buzzer does not beep.

Experiment No. #13

Objective:-

PIR Sensor with DC Motor control by Arduino

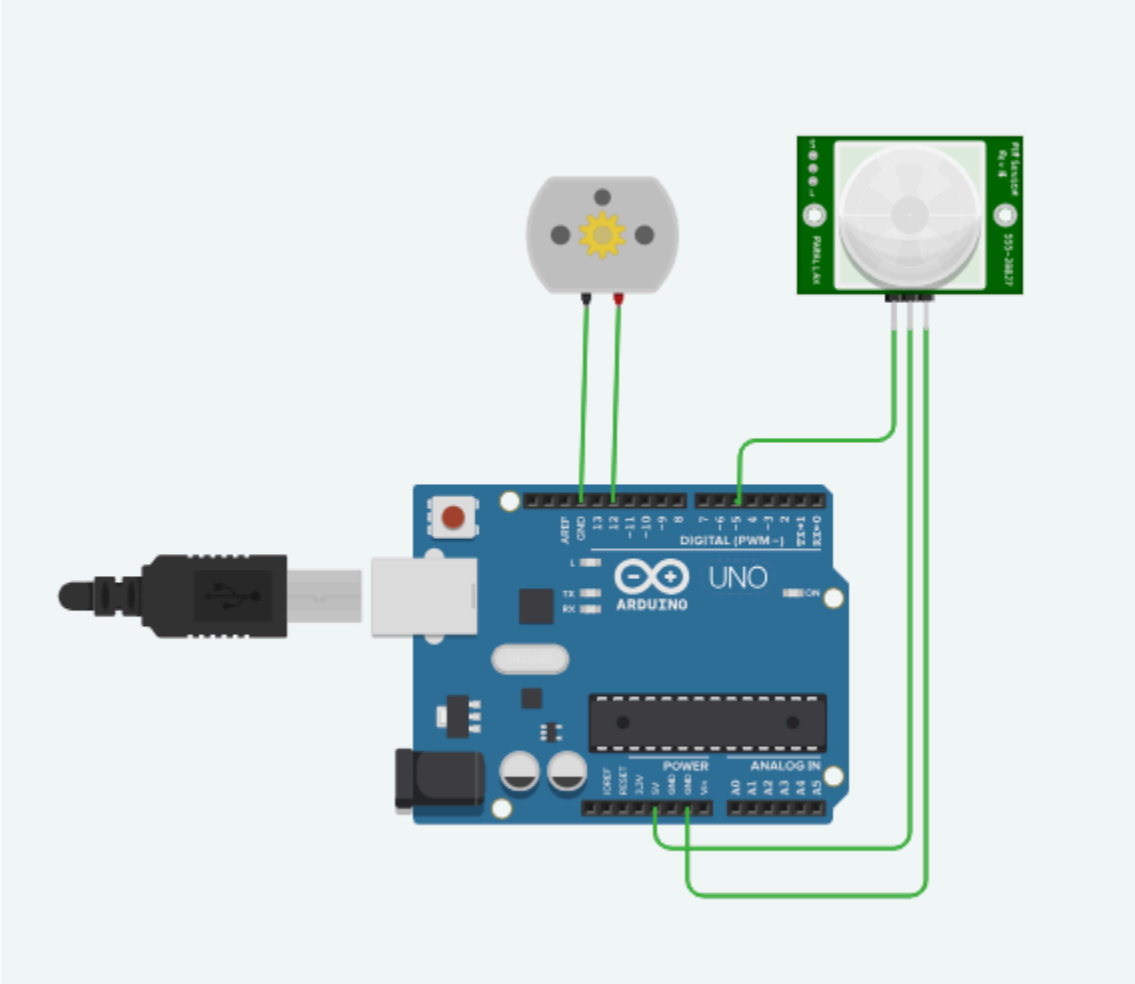
Components Required:-

Arduino UNO
PIR Sensor
Motor Sensor

Theory:-

In this project, we control when we get motion then the led is glow else the led off so we need to connect the vcc pin of the PIR Sensor to the 5v Arduino's pin and the ground pin of the sensor connect to the ground pin of the arduino and the signal pin of the sensor connect to the any digital pin of the Arduino and the +ive pin of the Motor connect to any digital pin of arduino and the -ive pin of the Motor is connected to the ground.

BlockDiagram:-



Programming:-

```
int pirPin = 2;
int motorPin = 12;
int data;
void setup() {
  // put your setup code here, to run once:
  pinMode(pirPin, INPUT);
  pinMode(motorPin, OUTPUT);
  Serial.begin(9600);

}

void loop() {
  // put your main code here, to run repeatedly:
  data = digitalRead(pirPin);
  if(data == HIGH){
    Serial.println("Motion Detected");
    digitalWrite(motorPin, HIGH);
  }else{
    Serial.println("Not Motion Detected");
    digitalWrite(motorPin, LOW);
  }
  delay(500);

}
```

Program Explain:-

First of all we initialize the data pin of the sensor to the arduino digital pin 2 and the motorPin is connected to the digital pin to the arduino 12.

In the setup() function:- setup function is used to execute the code for a single time and the variable inside the setup function is fixed for one time.

In loop() function:- The code execute infinite time while we give power to our arduino and the digitalRead() function inside the loop function read while motion is present in front of the PIR sensor and store the data in the data variable and we apply the condition if data == HIGH then print motion detected and the motor rotate else not motion detected and the motor is not rotate.

Experiment No. #14

Objective:-

RFID control by Arduino UNO

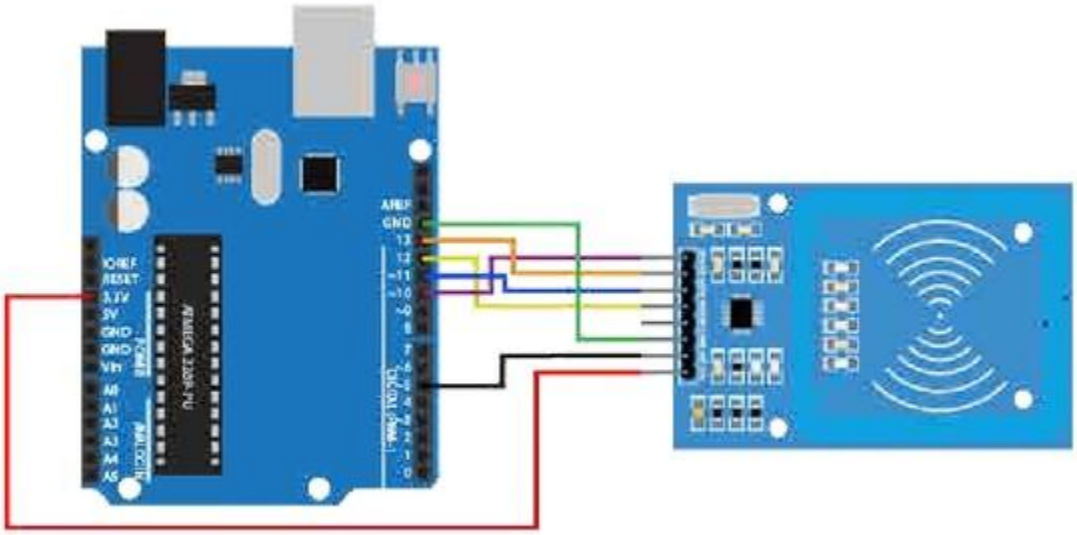
Components Required:-

Arduino UNO
RFID Sensor

Theory:-

In this project, we control RFID Sensor with Arduino UNO, first of all we connect the vcc pin of the sensor to the 3.3v pin of the arduino and the ground pin of the sensor connect to the ground pin of the Arduino and connect all the pins as below the diagram.

BlockDiagram:-



Program:-

```
#include <SPI.h>
#include <MFRC522.h>

#define RST_PIN 9
#define SS_PIN 10

MFRC522 mfrc522(SS_PIN, RST_PIN);

void setup() {
  Serial.begin(9600);
  SPI.begin();
  mfrc522.PCD_Init();
}

void loop() {
  if (mfrc522.PICC_IsNewCardPresent()) {
    Serial.println("RFID Card Detected");
    delay(1000);
  }
}
```

Program Explain:-

First of all we define the library of the RFID sensor as well as we define the pin of the sensor.

In the setup() function:- setup function is used to execute the code for a single time and the variable inside the setup function is fixed for one time.

In loop() function:- The code executes infinite time while we give power to our arduino. In the loop function we apply a condition when an RFID card is present then we see a message that RFID Card Detected else nothing to see.

Experiment No. #15

Objective:-

NFC control by Arduino UNO

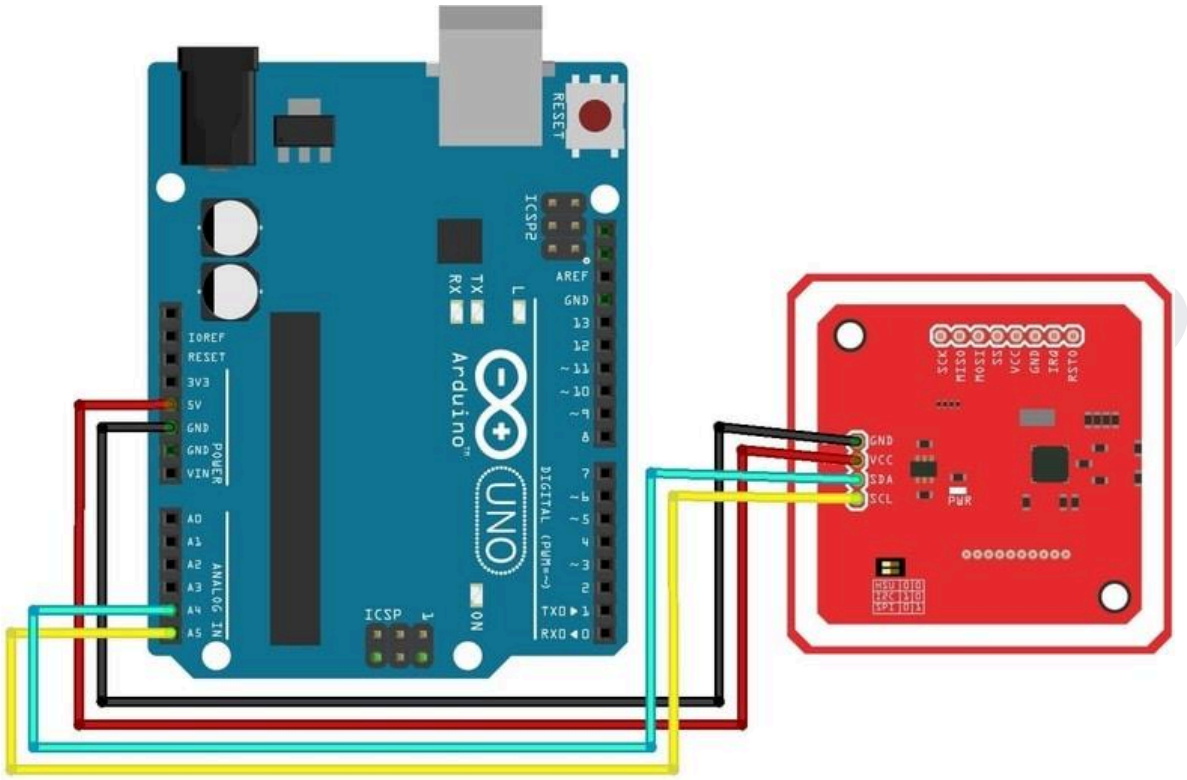
Components Required:-

Arduino UNO
NFC Sensor

Theory:-

In this project, we control the NFC sensor with the help of the Arduino, first of all we connect the vcc pin of the sensor connect to the arduino 5v pin and the ground pin of the sensor connect to the ground pin of the arduino and the SDA pin of the sensor connect to the A4 pin of the arduino and the SCL pin of the sensor connect to the A5 pin of the arduino.

BlockDiagram:-



Program:-

```
#include <Wire.h>
#include <PN532_I2C.h>
#include <NfcAdapter.h>

PN532_I2C pn532_i2c(Wire);
NfcAdapter nfc = NfcAdapter(pn532_i2c);

void setup(void) {
  Serial.begin(115200);
  Serial.println("System initialized");
  nfc.begin();
}

void loop() {
  if (nfc.tagPresent()) {
    NfcTag tag = nfc.read();
    tag.print();
    Serial.print("UID: ");
    Serial.println(tag.getUidString());
  }
  delay(1000);
}
```

Program Explain:-

First of all we define the library of the NFCsensor. And we create a object of the NFC sensor

In the setup() function:- setup function is used to execute the code for a single time and the variable inside the setup function is fixed for one time.

In loop() function:- The code executes infinite time while we give power to our arduino. In the loop function we apply a condition when an NFC card is present then we see a message that NFC Card Detected and print the card data into the SerialMonitor Screen.