# Introduction to Embedded Systems

Practical no. 1

Aim: Introduction to Arduino circuits and breadboarding. Blinking of LEDs

Hardware requirements: Arduino UNO R3, Breadboard, Resistor, LED.

Code:

```
void setup()
{
 pinMode(LED_BUILTIN, OUTPUT);
 digitalWrite(LED_BUILTIN, HIGH);
}
void loop()
{
 digitalWrite(LED_BUILTIN, HIGH);
 delay(5000); // Wait for 1000 millisecond(s)
 digitalWrite(LED_BUILTIN, LOW);
 delay(5000); // Wait for 1000 millisecond(s)
}
```

Output:

**
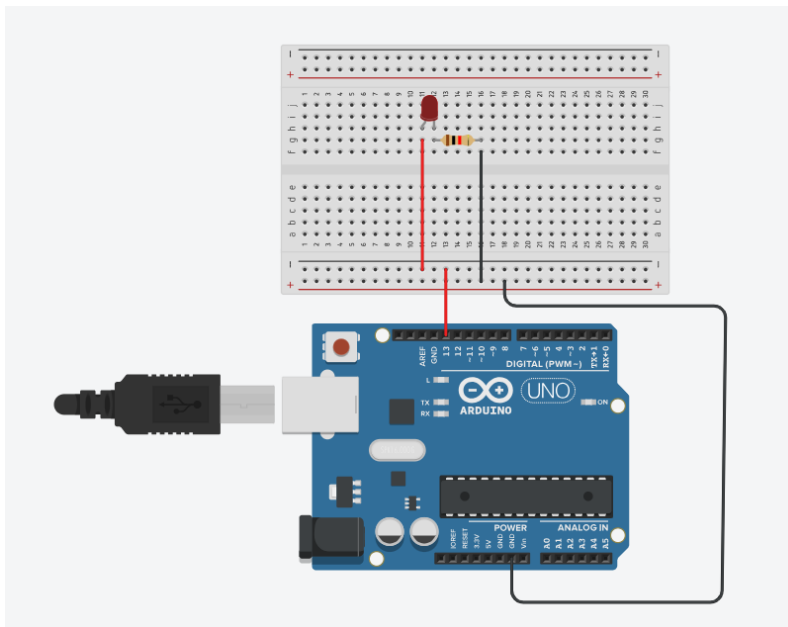
Practical no. 2

Aim: Program using Light Sensitive Sensors

Hardware requirements: Arduino UNO R3, Photoresistor,Resistor, LED.

Code:
```
int sensorValue=0;
void setup()
{
 pinMode(A0, INPUT);
 pinMode(9, OUTPUT);
 Serial.begin(9600);
}
void loop()
{
 sensorValue= analogRead(A0);
 Serial.println(sensorValue);
 analogWrite(9,map(sensorValue,0,1023,255,0));
 delay(100); // Wait for 100 millisecond(s)
}
```

Output:

****
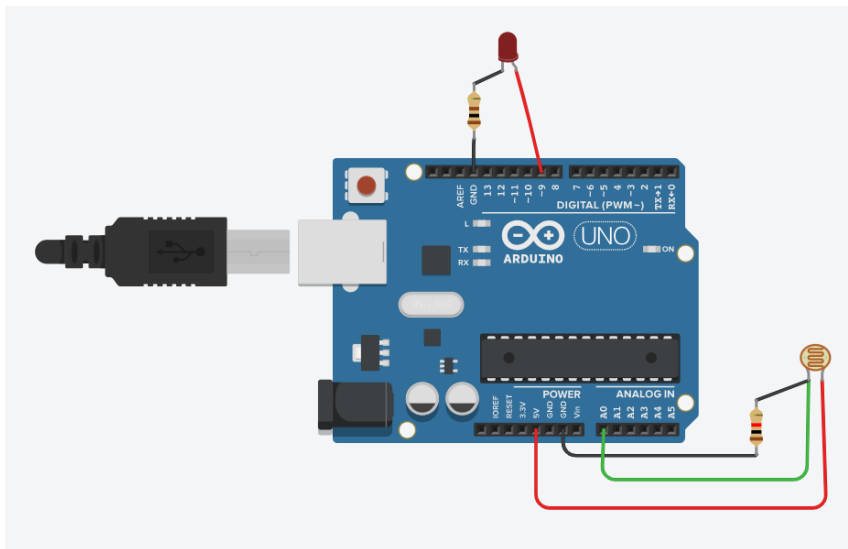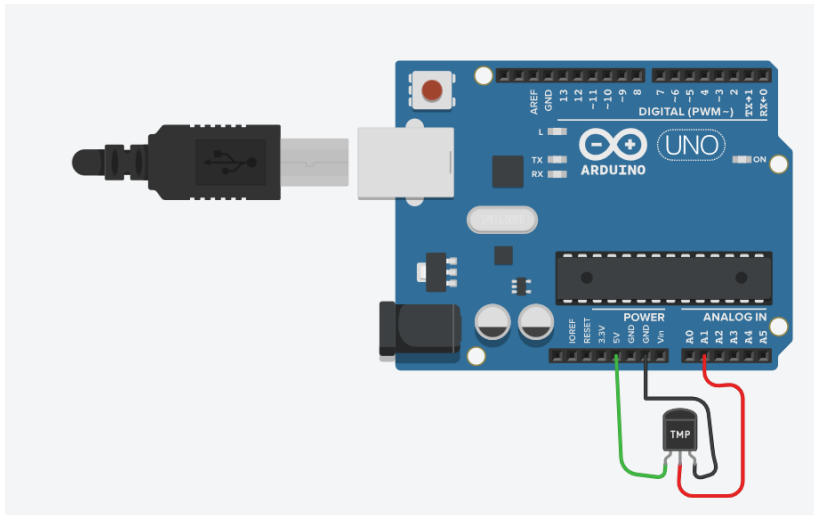
Practical no. 3

Aim: Program using temperature sensors

Hardware requirements: Arduino UNO R3, Temperature Sensor.

Code:

```
char degree=176; //ASCII Value of Degree
const int sensor=A1;
void setup()
{
 pinMode(sensor, INPUT);
 Serial.begin(9600);
}
void loop()
{
 int tmp=analogRead(sensor);
 float voltage=(tmp*5.0)/1024;
 float tmpCel=(voltage-0.5)*100.0;
 Serial.print("Celsius:");
 Serial.print(tmpCel);
 Serial.println(degree);
 delay(1000);
}
```

Output:

****
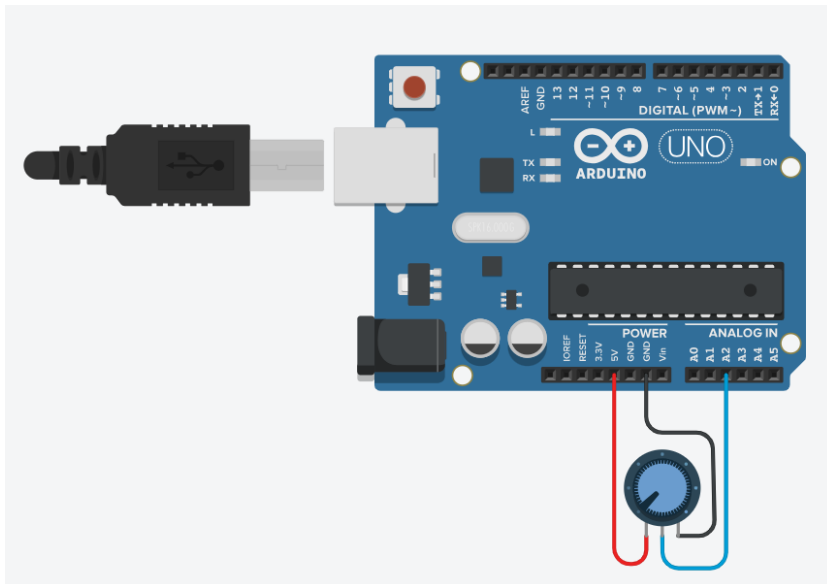
Practical no. 4

Aim: Programs using humidity sensors

Hardware requirements: Arduino UNO R3, Potentiometer.

Code:

```
const int analogIn=A2;
int humiditySensorOutput=0;
void setup()
{
        Serial.begin(9600);
}
void loop()
{
 humiditySensorOutput=analogRead(analogIn);
 int humidityPercentage =map(humiditySensorOutput, 0,1023,10,70);
 Serial.print("Humity:");
 Serial.print(humidityPercentage);
 Serial.println("%");
 delay(5000);
}
```

Output:

****
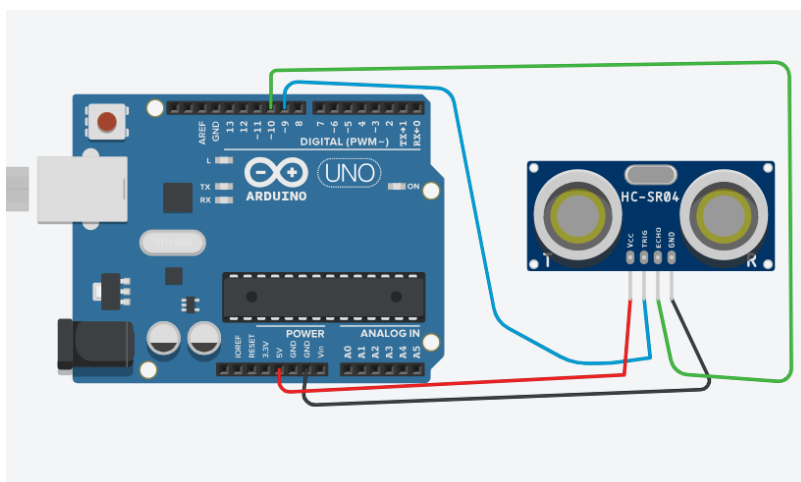
Practical no. 5

Aim: Programs using Ultrasonic Sensors

Hardware requirements: Arduino UNO R3, Ultrasonic Distance Sensor.

Code:

```
const int trigPin=9;
const int echoPin=10;
long duration;
int distance;
void setup()
{
 Serial.begin(9600);
 pinMode(trigPin,OUTPUT);
 pinMode(echoPin,INPUT);
}
void loop()
{
 digitalWrite(trigPin,LOW);
 delayMicroseconds(2);
 digitalWrite(trigPin,HIGH);
 duration=pulseIn(echoPin,HIGH);
 distance=duration*0.034/2;
 Serial.print("Distance:");
 Serial.print(distance);
 Serial.println("cm");
 delay(1000);
}
```

Output:

****
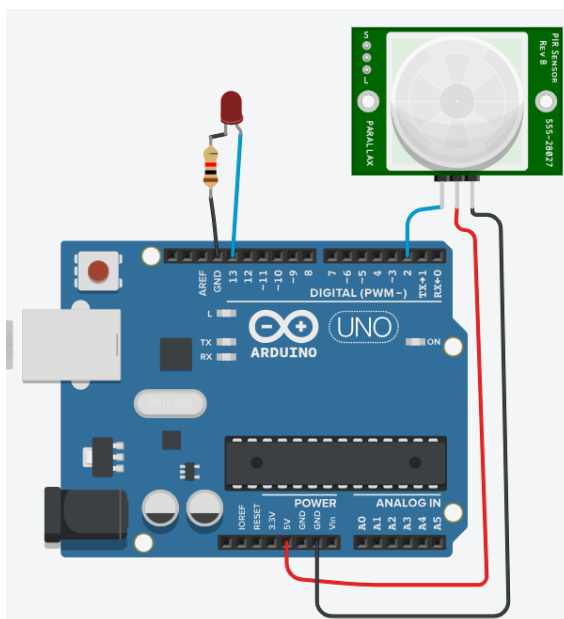
Practical no. 6

Aim: Programs using digital infrared motion sensors.

Hardware requirements: Arduino UNO R3, Resistor, LED, PIR sensor.

Code:

```
int sensorState = 0;
void setup()
{
        pinMode (2, INPUT);
        pinMode (LED_BUILTIN, OUTPUT);
}
void loop()
{
        sensorState = digitalRead(2);
        if (sensorState == HIGH)
  {
                digitalWrite(LED_BUILTIN, HIGH);
        }
        else
        {
                digitalWrite(LED_BUILTIN, LOW);
        }
        delay(10);
}
```

Output:

****
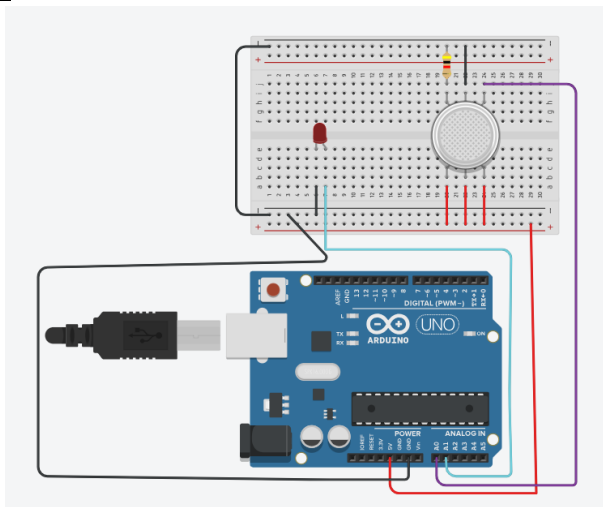
Practical no. 7

Aim: Programs using gas sensors.

Hardware requirements: Arduino UNO R3, Breadboard, Resistor, LED, Gas sensor.

Code:

```
const int LED_PIN = A1;
const int SENSOR_PIN = A0;
const int SMOKE_THRESHOLD=470;
void setup()
{
        Serial.begin(9600);
        pinMode (LED_PIN, OUTPUT);
}
void loop()
{
        int sensorValue = analogRead(SENSOR_PIN);
  {
   if (sensorValue >= SMOKE_THRESHOLD)
   {
                digitalWrite(LED_PIN, LOW);
                Serial.print("Smoke Detected!Sensor Value: ");
        Serial.println(sensorValue);
     }
        else
   {
        digitalWrite(LED_PIN, HIGH);
                Serial.print("No Smoke. Sensor Value: ");
                Serial.println(sensorValue);
   }
  }
delay(1000);
 }
```

Output:

****

Practical no. 8

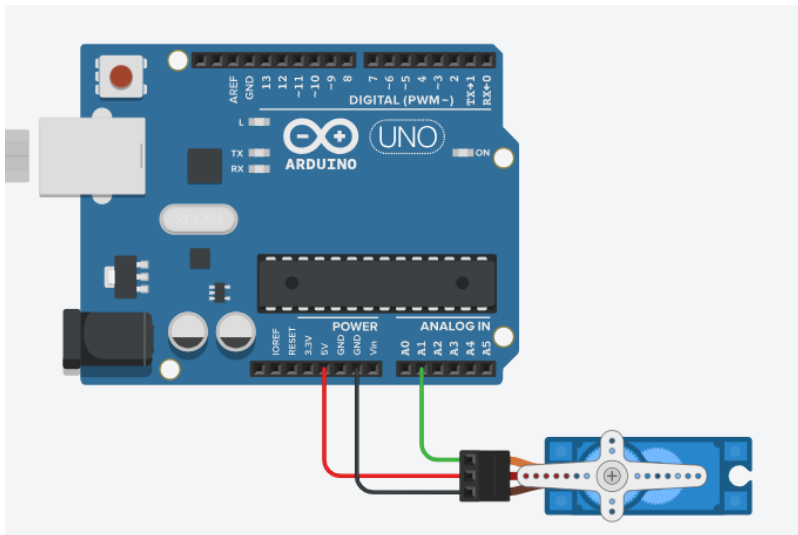**Aim:** Programs using servo motors

Hardware requirements: Arduino UNO R3, Micro Servo Motor.

Code:

```
#include<Servo.h>
Servo servoBase;
void setup()
{
 servoBase.attach(A1);
 servoBase.write(0);
}

void loop()
{
 for(int i=0;i<=180;i+=20)
 {
  servoBase.write(i);
  delay(1000);
 }
}
```

Output:

****