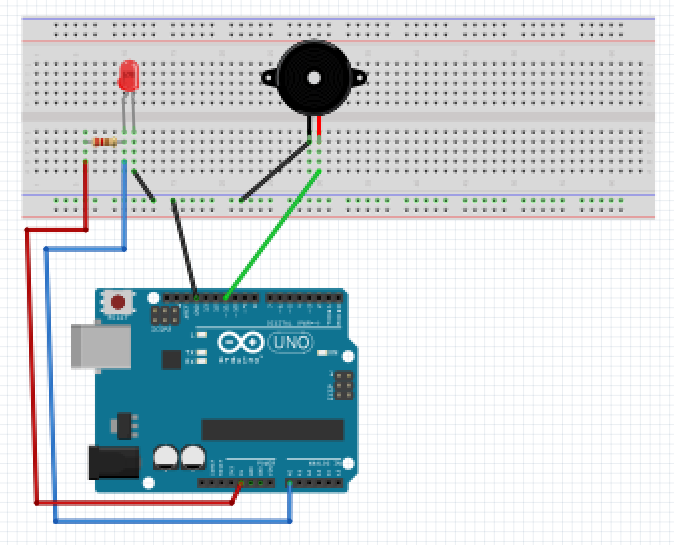
**ARDUINO LED PROJECT**

**Flame sensor with Buzzer**



For this project we will need:

* Arduino board.
* Buzzer.
* Flame Sensor.
* Breadboard.
* 470 Ohm resistor.
* 2 Male to Male jumper wires.

Circuit Design:

1. First make sure that the Arduino is powered off (no USB cable plugged to power).
2. Check the Flame Sensor, you will see that one of the leg is shorter than the other one.
3. Plug the longer leg (anode) to the breadboard to a separate horizontal line.
4. Attach a 470 Ohm resistor from the anode line to connect to pin 5V on the arduino..
5. On the same anode and resistor line, plug a blue jumper wire to connect to port A0 on arduino.
6. Plug the shorter leg (cathode) to the breadboard to a separate horizontal line..
7. Plug a black jumper wire from the cathode line to the common ground of the breadboard.
8. Check the buzzer .Note the + sign attached to the longer leg.
9. Plug the longer leg (anode) of the buzzer to the breadboard to a separate horizontal line..
10. Using a green jumper wire, connect the anode line to pin 11 on Arduino.
11. Plug the shorter leg (cathode) of the buzzer to the breadboard to a separate horizontal line..
12. Using a black jumper wire, connect the cathode line to common ground of the breadboard..
13. Using a black male jumper wire, connect the common ground of the breadboard to the GND arduino pin.

|  |
| --- |
| const int flameSensor =A0;  const int buzzer= 11;  void setup() {  Serial.begin(9600);  pinMode(flameSensor, INPUT);  pinMode(buzzer, OUTPUT);  }  /void loop(){  int flameVal=analogRead(flameSensor);  if (flameVal<1023){  tone(buzzer,800,800);  delay(200);  tone(buzzer,600,800);  delay(200);  }  else{  digitalWrite(buzzer,LOW);  }  Serial.println(flameVal);  delay(1000);  } |

const int flameSensor =A0;

const int buzzer= 11;

We instruct the Arduino to use these values as the pins attached to the flame sensor and the buzzer.

const the value cannot be changed during program execution or reassigned a new value.

int the value is a number.

pinMode(flameSensor, INPUT);

pinMode(buzzer, OUTPUT);

This sets the declared pins , flame sensor and buzzer , functionality .

This allow as to receive INPUT and send OUTPUT to the Arduino.

int flameVal=analogRead(flameSensor);

This reads the values received from the pin attached to sensor and stores it in number form.

if (flameVal<1023){\

This sets a condition to be checked.

With every new value received as input, this condition will be checked.

tone(buzzer,800,800);

delay(200);

tone This is a predefined function that takes in 3 values.

The first value represents the device we wish to use to play the sound.

The second value represents the frequency of the tone.

The third value represents the duration to play the tone.

else{

digitalWrite(buzzer,LOW);

if the input value did not meet the condition the program will send a signal LOW to the buzzer.

This will turn it off.

voidloop(){}

After executing the void setup() function, we enter the void loop() and this function is executed continuously and repeatedly, until you Arduino is powered off.