What we want from a Resistor Device is its **“Resistance Value”**

Resistance is used to slow down the intensity of “Current” flow, preventing device from damage (all devices have Current exposure limits). Resistors are also often used for “splitting” Voltage out from the Original Voltage to get a specific Voltage for a purpose. They can come in many shapes and sizes.

There are two groups of Resistor

“Fixed Resistance” and **“Variable Resistance”**

Fixed Resistance Resistor

Resistance Value in each of this Resistor device is already fixed by the manufacturer, the Resistance Value in each device cannot be changed

These device always have some “markings” on the physical device (such as color codes, numbers, alphabets and etc), so that we know how much Resistance Value it can provide

Variable Resistance Resistor

Resistance Value in each of this Resistor device ussually have an initial “pre-fixed” controlled value by the manufacturer. The Resistance Value in this device can be changed later on. These device sometimes have some markings and sometimes does not have no markings at all. Normally, we have to find out the actual Resistance Value on our own (because the resistance value is changing most of the time)

- Resistance Value can be changed by manually adjusting a “knob” or some other kind of “mechanical component” on the Resistor device to get different resistance value

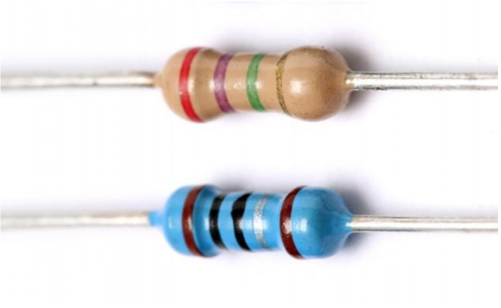
or

- Resistance Value can be changed according to what the Resistor Device is exposed to, for example: temperature, moisture, light, and etc... These kind of resistors are also known as **Analog Sensors**

The easiest way to find out the Resistance Value of a Resistor Device is to use a Multimeter, otherwise you we have to learn all the different markings on the physical Resistor device

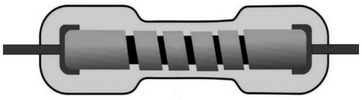
Introducing: The 0.25W Fixed Value Resistor

I am sure all of us have seen these when working with Arduino Uno and LED

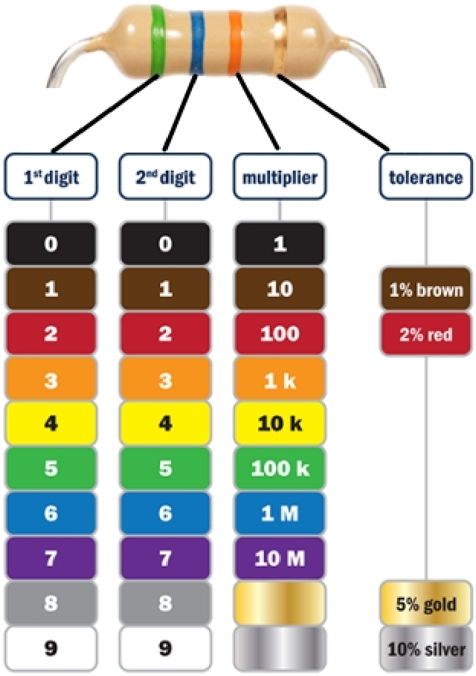


0.25W **Carbon Film Resistor**
has carbon film inside,
cheaper but less accurate

0.25W **Metal Film Resistor**
Has metal film inside,
more expensive but more
accurate



0.25W Resistor Device are color coded for their Resistance Value. You need to have a good eye-sight for it because these colors sometimes are hard to see



1st digit	2nd digit	multiplier	tolerance
0	0	1	
1	1	10	1% brown
2	2	100	2% red
3	3	1 k	
4	4	10 k	
5	5	100 k	
6	6	1 M	
7	7	10 M	
8	8		5% gold
9	9		10% silver

Example: 3 Color Band Resistor

Green = 5
Blue = 6
Orange = 1000(multiplier)
56 x 1000 = 56,000

56kilo-ohm

5% tolerance (deviations)

However, if you have a Multimeter, you can ignore all the colors or markings on any Resistor Device. The Multimeter can tell us the Resistance Value

Another use of Multimeter is, **it can detect faulty Resistors** while the color coding or markings cannot do that for us.

So, if you can, get a multimeter, any cheap ones will do. However, the more expensive ones (auto-range type) are more accurate and easier to use

What is 0.25W ?

0.25W is the specification given by manufacturer, the maximum amount of Power that the Resistor Device can withstand.

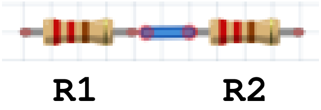
Power(Watt) = Voltage(Volt) x Current(Ampere)

In a 5V circuit, 0.25W / 5V = 0.05A of current. Means you should not have more than 50mA Current going into it (for Arduino use, that is fine)

The common standard power ratings for Fixed Value Resistors are 0.25W, 0.5W, 1W, 2W, 5W, and 25W. Choose the Resistor Power ratings accordingly.

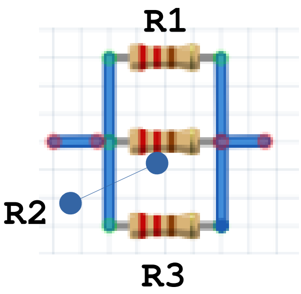
Sometimes a single Fixed Value Resistor cannot give us the Resistance Value we wanted, we can combine a few of them to get our Resistance Value

When more than one Fixed Value Resistors are used combined, they will result in a different Resistance Value. The Resistance value will depend on how the Fixed Value Resistors are connected (in Series or in Parallel and the number of Resistors combined)



More than one Resistors in Series:

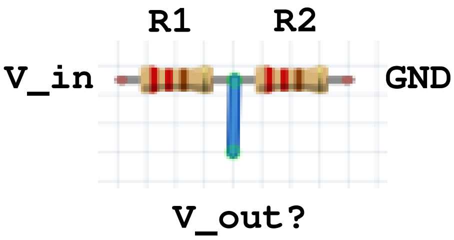
Total= R1 + R2



More than one Resistors in Parallel:

1/Total R = (1/R1)+(1/R2)+(1/R3)+...

Resistors can also be used as a "Voltage Divider". We will be using a lot of this when working with Arduino and micro-controller in our future lessons

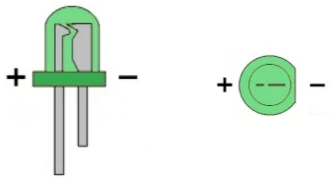


Voltage Divider:

V_out = V_in x (R2 / R1+R2)

LED (Light Emitting Diode) is special Diode, it can glow when electric current flow through it. Because of this glowing property, LED is often an Lighting Output Device (apart from its function as a Diode - to block current from flowing at the wrong direction)

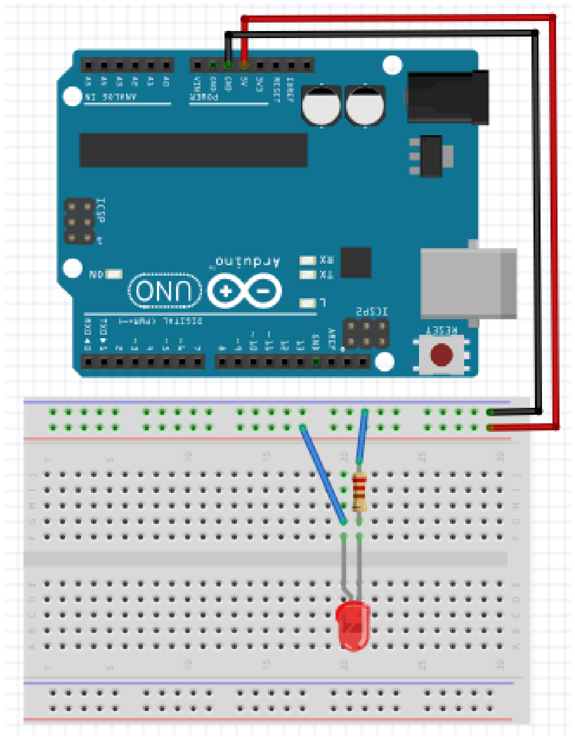
LED comes with many shape, sizes and brightness, all of them are Diode and they have Negative and Positive terminals.



- **Positive Terminal from LED** must be connected to **Positive(or VCC) Terminal of Power Supply**

- **Negative Terminal from LED** must be connected to **Negative(or GND) Terminal of Power Supply**

Each LED can handle a very small amount of Electric Current. If the Current exceeds the its limit, we will risk the LED being damaged. This is where we need to **add a Fixed value Resistor to the LED,** so that the intensity of current at LED is reduced, **protecting the LED from being damaged.**



1x Computer with Arduino IDE Software
1x USB 2.0 Type A/B Data Cable
1x Arduino Uno Board
Jumper Wires

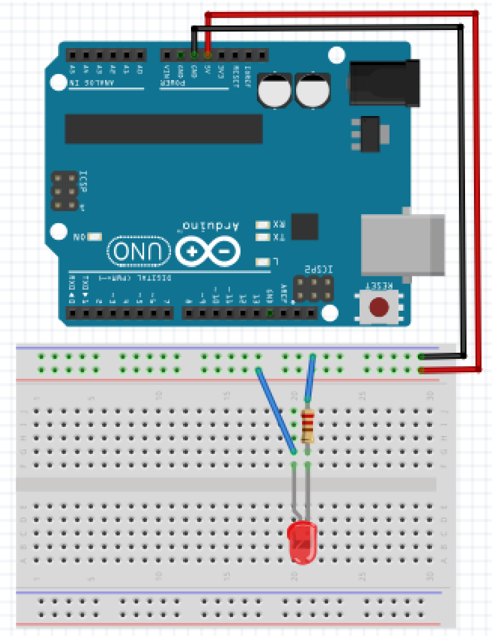
1x Solderless Breadboard
1x 5mm bulb LED
1x 220 Ohm resistor

LED +ve to Arduino 5V
LED -ve to Resistor to Arduino GND

There is no programming required here. The LED in this circuit will just light up after receiving 5V from the Arduino Uno Board

Here we are using a 0.25W Resistor (220 Ohm) with the Arduino Uno 5V to protect the LED from being damaged

Why 220 Ohm ?



1x Computer with Arduino IDE Software
1x USB 2.0 Type A/B Data Cable
1x Arduino Uno Board
Jumper Wires

1x Solderless Breadboard
1x 5mm bulb LED
1x 220 Ohm resistor

LED +ve to Arduino 5V
LED -ve to Resistor to Arduino GND

Why 220 Ohm Resistor ?

We have following information from our circuit and devices:
1. Arduino Uno Board 5V Pin
- Total Supply Voltage = 5V

2. Regular 5mm “bulb” LED (information from LED manufacturer datasheet)
- LED Voltage Drop = 1.8V
- LED Current Limit = 20mA (or 0.02A)

The following two information (3 and 4), we need to find out on our own:
3.Voltage Drop at Resistor(VDR) ?
4.How much Resistor Value, to have 20mA(or 0.02A) on the LED ?

3.Voltage Drop at Resistor(VDR) ?

Kirchhoff’s Voltage Law,
Total Supply Voltage = Total Voltage Drop

Total Supply Voltage 5V(Arduino Uno) = 1.8V(LED) + **VDR?(Resistor)**
VDR = 5V-1.8V **= 3.2V**

4.How much Resistor Value, to have 20mA(or 0.02A) on the LED ?
We know we 20mA is the limit for our LED, how much Resistor to get exactly 20mA limit ?

Ohms Law,
Resistance(Ohm) = Voltage(Volt) / Current(Ampere)

Resistor Value = **3.2V**(from **VDR**) / **0.02A**(our target)
Resistor Value = 160 Ohm

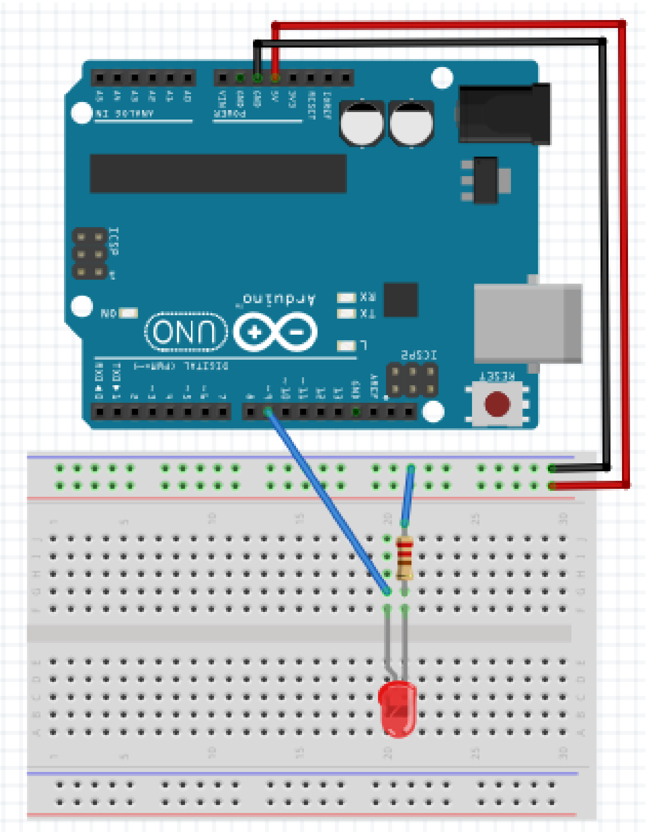
- If we use a **160 Ohm Resistor**, we **have exactly 20mA on the LED**
- If we use a Resistor that is **less than 160 Ohm**, we **have more than 20mA on the LED** (risk LED damaged)
- If we use a Resistor that is **more than 160 Ohm**, we **have less than 20mA on the LED** (LED is safe)

The next higher available “0.25W Resistor Device” from **160 Ohm**, is **220 Ohm**, that is why **we use 220 Ohm Resistor in this circuit.**

If we do not have a 220 Ohm resistor, we can use other values (as long as it is more than 160 Ohm). Some people prefer to use the 330 Ohm Resistor or higher. The higher the resistance, then dimmer LED will become.

ATMEGA328/Arduino Uno – LED / Resistor and I/O Pin

https://github.com/teaksoon/lmaewapm



1x Computer with Arduino IDE Software
1x USB 2.0 Type A/B Data Cable
1x Arduino Uno Board
Jumper Wires

1x Solderless Breadboard
1x 5mm bulb LED
1x 220 Ohm resistor

LED +ve to Arduino Pin 9
LED -ve to Resistor to Arduino GND

Just like the 5V Pin on the Arduino Uno Board, the OUTPUT I/O Pin also supplies 5V. So, we will need to have a resistor (220 Ohm or more) when we want to use the OUTPUT I/O Pin with the 5mm bulb LED

Here is a Program to supply 5V and 0V to the OUTPUT I/O Pin 9 on Arduino Uno
Upload and watch the LED

```
void setup() {
  pinMode(9, OUTPUT);
}

void loop() {
  digitalWrite(9,HIGH);
  delay(250);
  digitalWrite(9,LOW);
  delay(250);
}
```

Our Program starts from “setup” function. Inside the “setup” function, we run the function “pinMode”, which set I/O Pin 9 to be an OUTPUT Pin.

Since there are no more codes inside the “setup” function after the “pinMode(9,OUTPUT);” The “loop” function will start, starting from the code “digitalWrite(9,HIGH);” inside the “loop” function.

Inside the “loop” function,
Step 1. Supply 5V to OUTPUT I/O Pin 9, caused the LED turned ON
Step 2. waits for 250 mili-seconds doing nothing,
Step 3. Supply 0V to OUTPUT I/O Pin 9, caused the LED turned OFF
Step 4. waits for 250 mili-seconds doing nothing,

After the “loop” function ran its last code Step 4. , the “loop” function will run from Step 1. inside the “loop” function again. Means the “loop” function runs forever repeating Step 1. to Step 4. That is how we get a “blink” effect on the LED

You can try use a large Resistor Value, for example 100KOhm and watch the LED, it will still blink but very be much dimmer.