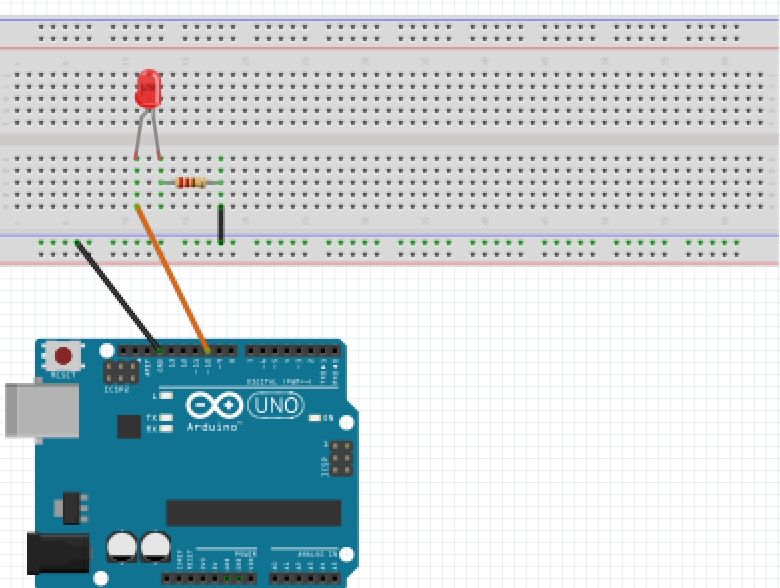
**ARDUINO LED PROJECT**

**Blinking LED light**



For this project we will need:

* Arduino board.
* LED light.
* Breadboard.
* 470 Ohm resistor.
* 2 Male to Male wires.

Circuit Design:

1. First make sure that the Arduino is powered off (no USB cable plugged to power).
2. Check the LED, you will see that one of the leg is shorter than the other one.
3. Plug the longer leg of the LED(anode) to a horizontal line on the breadboard. Connect using an orange wire from this line to pin 12 of the arduino.
4. Plug the shorter leg of the LED(cathode) to the breadboard. Separate horizontal line. Use a 470 ohm resistor from this line to connect to the common ground of the breadboard (marked by a blue (-) line).
5. Plug a 470 ohm resistor to the horizontal line connected to the shorter leg(anode), then to the common ground of the breadboard.
6. Plug the common ground of the breadboard to the ground pin of the Arduino using a black male jumper wire.

|  |
| --- |
| const int ledPin=10; // declare ARDUINO pin connected to LED anode  void setup() // assign the Led to output function.  pinMode(ledPin, OUTPUT);  }  void loop(){  digitalWrite(ledPin,HIGH); //turn on LED  delay(1000);  digitalWrite(ledPin,LOW); //turn off LED`  delay(1000);  } |

const int ledPin = 10;

First, we create a reference for the digital pin we intend to use and define a variable for that pin number. This will enable us to reference the pin by that variable name ledPin instead of the hard-coded number. In subsequent designs, if you need to use a different digital pin (for example pin 11), then you just need to change the number here and it will update it everywhere in your design program.

const shows that the value assigned doesn’t change.

int shows the value is a number.

pinMode(ledPin, OUTPUT);

After the execution of this line, the digital pin 10 will be set as output, and this will enable us to send information to it and control the LED.

digitalWrite(ledPin, HIGH);

Now, to control the LED, it’s very simple. We need to use the digitalWrite() function to control the LED. We will use the below parameters to achieve this goal:

* Pin number: We use the defined variable for the pin number, LED\_PIN.
* State: This has only 2 choices. HIGH to power on the LED, and LOW to power it off.

Note: The mode needs to be set before using the digitalWrite(), otherwise the LED won’t be powered on.

delay(1000);

This determines how long the state will be maintained. The value passed is measured in milliseconds.

void setup (){}

This initializes the arduino and assigns functionality to its pins.

This also provides required resources for monitoring.

void loop(){}

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.