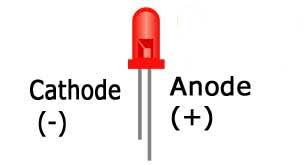
**HOW TO BLINK A LED USING NODE MCU-ESP8266**

**LED**

LED in full **light-emitting diode**, in [electronics](https://www.britannica.com/technology/electronics), a [semiconductor device](https://www.britannica.com/technology/semiconductor-device) that emits infrared or visible light when charged with an [electric current](https://www.britannica.com/science/electric-current). Visible LEDs are used in many electronic devices as indicator lamps, in automobiles as rear-window and brake lights, and on billboards and signs as alphanumeric displays or even full-colour posters. Infrared LEDs are employed in autofocus cameras and television remote controls and also as light sources in fibre-optic [telecommunication](https://www.britannica.com/technology/telecommunication) systems.



The important thing to be remembered is that the long end of the LED is the anode and the shorter end is the cathode.

if someone's trimmed the legs, try finding the flat edge on the LED's outer casing. The pin nearest the **flat edge** will be the negative, cathode pin.

Negative terminal (ie.,cathode ) is given to the ground.

Here we are trying to blink a led using node mcu-ESP8266.

To proceed with this we need to have the following:

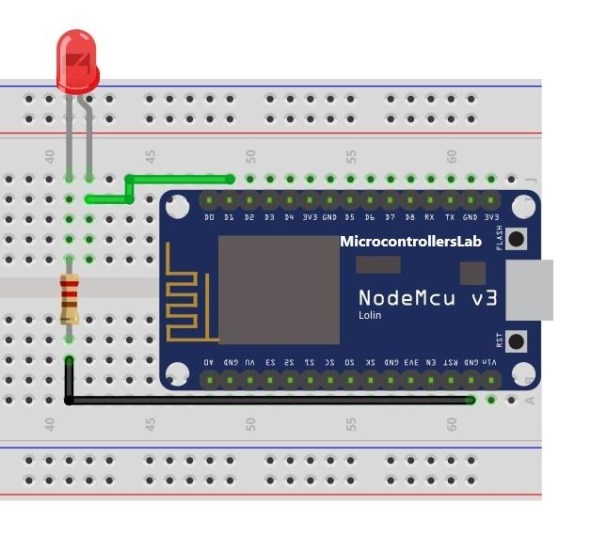
* Node mcu-ESP8266
* A mini bread board
* A resistor of 330 ohms
* LED of favourable color
* Connecting wires

Breadboard Setup

Push your node MCU pins into the mini breadboard at one end, so that the micro-USB is hanging over the edge (allowing you to plug it in).

Then run a resistor from the GND (G) pin to an empty lane on the breadboard. Next, connect the shorter (cathode) leg of your LED to the same lane, and the longer leg (anode) to the opposite lane.

Finally, run your jumper wire from the anode LED leg lane to the D7 pin on your node MCU.



1. **int** LED = D7;
2. **void** setup() {
3. pinMode(LED, OUTPUT); // Make LED pin D7 an output pin
4. }
5. **void** loop() {
6. digitalWrite(LED, LOW); // LED off
7. delay(3000); // Wait 3 seconds
8. digitalWrite(LED, HIGH); // LED on
9. delay(1000); // Wait 1 second
10. }

Understanding the Sketch

Let’s quickly explain what each part is doing here:

1. **int** LED = D7;

This is giving the D7 pin a name – ‘LED’.

1. **void** setup() {
2. pinMode(LED, OUTPUT); // Make LED pin D7 an output pin
3. }

This is a one-time command, setting the ‘LED’ pin as an output (to ‘output’ power when we tell it to)

1. **void** loop() {
2. digitalWrite(LED, LOW); // LED off
3. delay(3000); // Wait 3 seconds
4. digitalWrite(LED, HIGH); // LED on
5. delay(1000); // Wait 1 second
6. }

This is the loop – the main part of our program. This will keep looping over and over forever.

We tell the LED to set LOW then wait 3000 milliseconds (3 seconds) i.e. the LED should be off for 3 seconds.

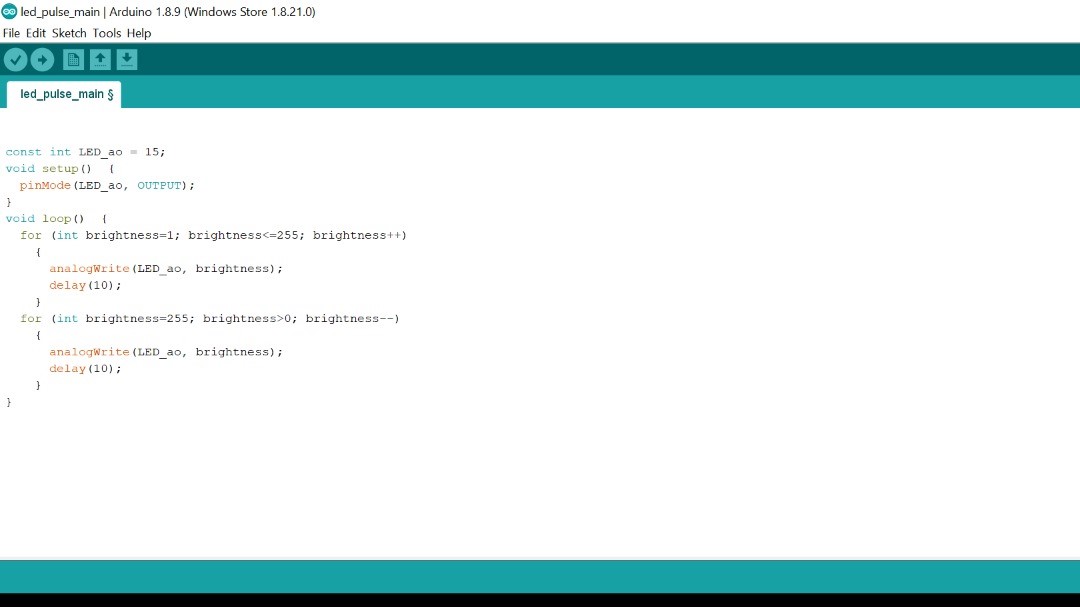
It then sets the LED HIGH and waits for 1000 milliseconds (1 second) i.e. the LED should light for 1 second.

Blinking an LED is just a basic step now let us tune it (increase or decrease the brightness)

TUNING THE LED

You can increase or decrease the brightness of the led by making a few modifications in the code.

Code:



REGARDS:

NARSIMHA REDDY

DHEERAJ T

RAGASRI YARLAGADDA