What’s an LED?

LED stands for Light-Emitting Diod. In other words a light source. Where might you find some - for starters on your keyboard - press Caps Lock - usually a light turns on - that is an LED.
The usual one coloured LED has two pins - anode(+) and cathode(-).
Usually we connect anode to the power source and the cathode to ground(GND). A resistor is needed to limit the current to the LED so it does not get damaged. The value of the resistor is determined by Ohm's law:

$$R = U / I$$

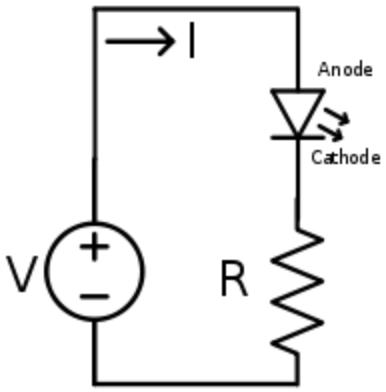
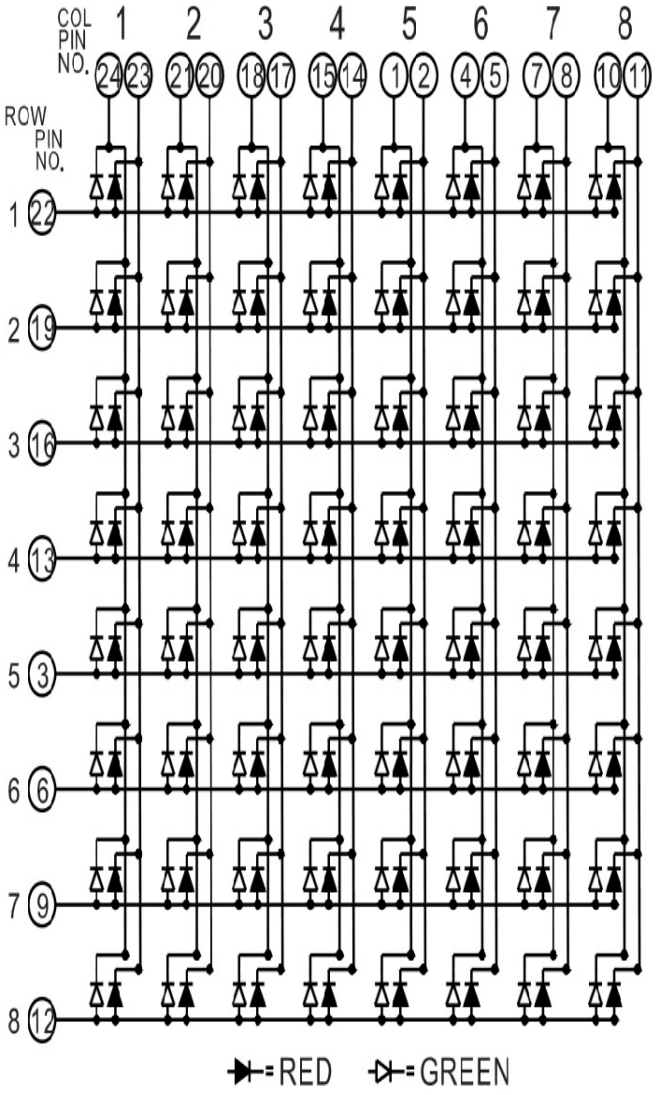


Image source:
https://upload.wikimedia.org/wikipedia/commons/thumb/c/c9/LED_circuit.svg/200px-LED_circuit.svg.png

What’s a LED matrix?

GS23088BXX

PCB:M23088A/BEG



Many LEDs connected together.

The can be two types of matrices - common-row anode and common-row cathode.

What I am using is common-row anode - the anodes of all the LEDs in a row are connected together. The idea is to use 16 pins to control 8 by 8 one-coloured matrix instead of 64 for controlling

each individual LED.
To light all LEDs we need to set all rows(anode) to high and all columns(cathode) to low. In my case the matrix is Bi-colour,which means I need to set the other colour to high. For example, I want to light all Green LEDs - rows - high, columns green - low, but if we leave red to low as well we will get yellow because the red ones would be turned on as well. So we set the red ones high so they do not interfere with the green.
To turn all LEDs off we simply set everything to low.
To turn on a single row only that specific row must be high - everything else is low.
To turn a specific column - all rows are high, all columns are high without the one we want to light up.
To turn on a specific LED we combine the two approaches from above.

Displaying an image

Multiplexing is a common way of displaying images on a matrix. Multiplexing uses the Persistence of Vision - which means that if an object changes its state quickly enough our brain would not have the time to process that information and see it as movement, but would rather see it as a static image. This is what we are trying to achieve here. We scan the matrix row by row and display the pattern on each individual row. If we manage to do this quickly enough our eyes would see it as a perfectly still image.

Problems:

On a Raspberry Pi - without extras this is quite impossible. Raspbian or any other Linux based system are the very reason for that. In order for the Pi to do the multiplexing it makes very frequent calls to the CPU - however these calls' priority is lower than the Raspbian itself. Another issue is the lack of real clock in the hardware, making it even harder to multiplex, because we do not have reliable information whether exactly 1 ms for example has passed.
To observe that directly try and moving the mouse while the code for multiplexing is being executed.

Possible Solutions to multiplexing:
You can use the Adafruit's LED Bi-colour matrix. This matrix has a separate chip that does the multiplexing for you.
Another possibility is to buy an extra chip to manage the multiplexing.
Also the less programs that you have open or running - the better. Images would not be flickering that much.
Other suggestions for dealing with this problem will be appreciated!