

# Lesson 3 Flowing LED Lights

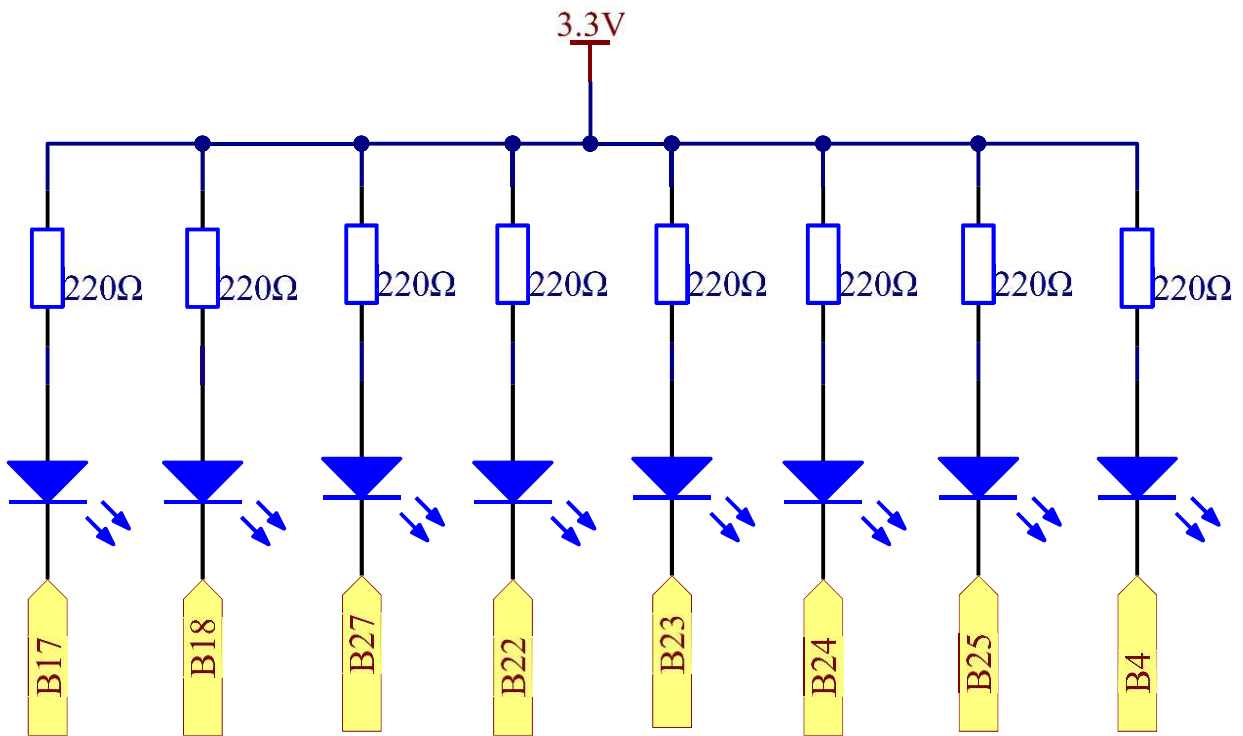
## Introduction

In this lesson, we will learn how to make eight LEDs blink in various effects as you want based on Raspberry Pi.

## Components

- 1 \* Raspberry Pi
- 1 \* Breadboard
- 8 \* LED
- 8 \* Resistor (220Ω)
- Jumper wires
- 1 \* T-Extension Board
- 1 \* 40-Pin Cable

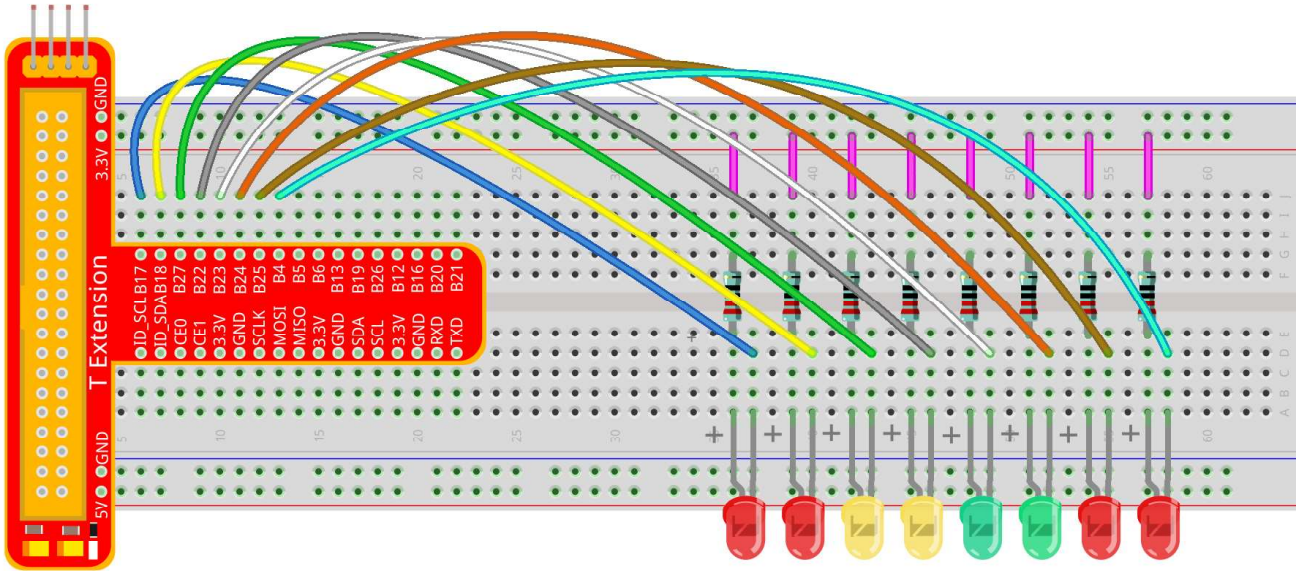
## Principle



**Principle:** Judging from the schematic diagram, we can know that a LED and a current-limiting resistor have been connected to B17, B18, B27, B22, B23, B24, B25, and B4 respectively. The current-limiting resistor has been connected to the 3.3V power supply on other side. Therefore, if we want to light up one LED, we only need to set the GPIO of the LED as low level. So in this experiment, set B17, B18, B27, B22, B23, B24, B25, and B4 to low level in turn by

programming, and then LED0-LED7 will light up in turn. You can make eight LEDs blink in different effects by controlling their delay time and the order of lighting up.

### Experimental Procedures



**Step 1:** Build the circuit

**For C language users:**

**Step 2:** Open the code file

```
cd /home/pi/SunFounder_Super_Kit_V3.0_for_Raspberry_Pi/C
```

**Note:** Use the cd command to switch to the code path of this experiment.

**Step 3:** Compile the Code

```
gcc 03_8Led.c -o 03_8Led -lwiringPi
```

or

```
make 03_8Led
```

**Note:** gcc is a linux command which can realize compiling and generating the C language program file **03\_8Led.c** to the executable file **03\_8Led**.

make is a linux command which can compiling and generating the executable file according to the rule inside the makefile.

**Step 4:** Run the executable file above

```
sudo ./03_8Led
```

**Note:** Here the Raspberry Pi will run the executable file *03\_8Led* compiled previously.

```

pi@raspberrypi: ~/SunFounder_Super_Kit_V3.0_for_Raspberry_Pi/C
pi@raspberrypi:~/SunFounder_Super_Kit_V3.0_for_Raspberry_Pi/C $ make 03_8Led
[CC] 03_8Led.c
[link]
pi@raspberrypi:~/SunFounder_Super_Kit_V3.0_for_Raspberry_Pi/C $ sudo ./03_8Led

=====
                        8 LEDs
-----
LED0 connect to GPIO0
LED1 connect to GPIO1
LED2 connect to GPIO2
LED3 connect to GPIO3
LED4 connect to GPIO4
LED5 connect to GPIO5
LED6 connect to GPIO6
LED7 connect to GPIO7

Flow LED effect

SunFounder|
=====

From left to right.
From right to left.
From left to right.

```

## Code Explanation

`void Led_on(int channel)`: This is a subfunction with a formal parameter `int channel` for importing the numbers of the controlled pins. Its function body is `digitalWrite(channel, LOW)`; Set the I/O port of `channel` as low level(0V), the LED on this port lights up. `void led_off(int channel)` is to turn off the LED. Setting function simplifies the input for the repeated content.

```

for(i=0;i<8;i++){          //make 8 pins' mode is output
    pinMode(i, OUTPUT);
}
//The cathodes of the 8 LEDs connect to B17, B18, B27, B22, B23, B24, B25, and B4 of
the T-shape extension board respectively, corresponding to 0,1,2,3,4,5,6,7. It is to set
the 8 LEDs as output separately. Use for loop to make it more concise and efficient.
for(i=0;i<8;i++){        //make LED on from left to right
    Led_on(i);           // turn the LED i on
    delay(100);          // keep the LED i lighting for 100ms.
    Led_off(i);          // Turn the LED i off
}
// Light up and turn off the LEDs in GPIO0~7 successively. i increases
progressively from 0 to 7, LED0 to LED7 changes accordingly, making it like a flowing
LED light from left to right.
for(i=7;i>=0;i--){       //make LED off from right to left
    led_on(i);           // turn the LED i on
    delay(100);          // keep the LED i lighting for 100ms
    led_off(i);          //turn the LED i off
}
// In this for loop, light up and turn off the LED in GPIO7 to GPIO0 successively,
making a flowing LED light from left to right.

```

**For Python users:**

**Step 2:** Open the code file

```
cd /home/pi/SunFounder_Super_Kit_V3.0_for_Raspberry_Pi/Python
```

**Step 3:** Run

```
sudo python 03_8Led.py
```

### Code Explanation

```
LedPins = [17, 18, 27, 22, 23, 24, 25, 4] # The cathodes of the 8 LEDs connect to B17,
B18, B27, 22, 23, 24, 25, 4 of the T-shape extension board. In BCM, these pins are
corresponding to 17, 18, 27, 22, 23, 24, 25, and 4.
leds = ['- ', '- ', '- ', '- ', '- ', '- ', '- ', '- '] # the array to print out the status of
the 8 LEDs
for pin in LedPins # Assign the element in pins list to pin variable one by one. In
GPIO.setup (pin, GPIO.OUT), set the pins in list as output one by one.
    GPIO.output(pin, GPIO.LOW) # Set each element in the pins list as low level to
light up the LEDs
    leds[LedPins.index(pin)] = 0 # Show which LED is on
    time.sleep(0.1) # wait for 0.1s
    GPIO.output(pin, GPIO.HIGH) # After delaying, set it as high level to light up or
turn off the LED.
    leds[LedPins.index(pin)] = '- ' # Show the led is off
```

You will see the eight LEDs lighten up one by one, and then dim in turn.

