6.LED_Bar_Graph

Introduction

In this lesson, we sequentially illuminate the lights on the LED Bar Graph.

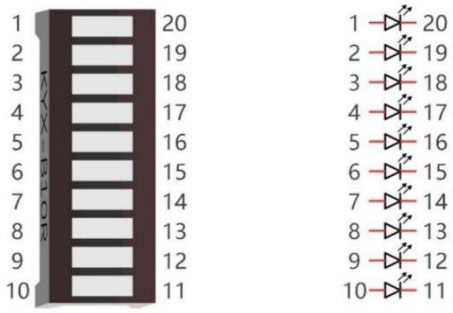
Hardware Required

- ✓ 1 * Raspberry Pi
- ✓ 1 * T-Extension Board
- ✓ 1 * LED Bar Graph
- ✓ 1 * 40-pin Cable
- ✓ Several Jumper Wires
- ✓ 1 * Breadboard
- ✓ 10 * Resistor(220Ω)

Principle

LED Bar Graph

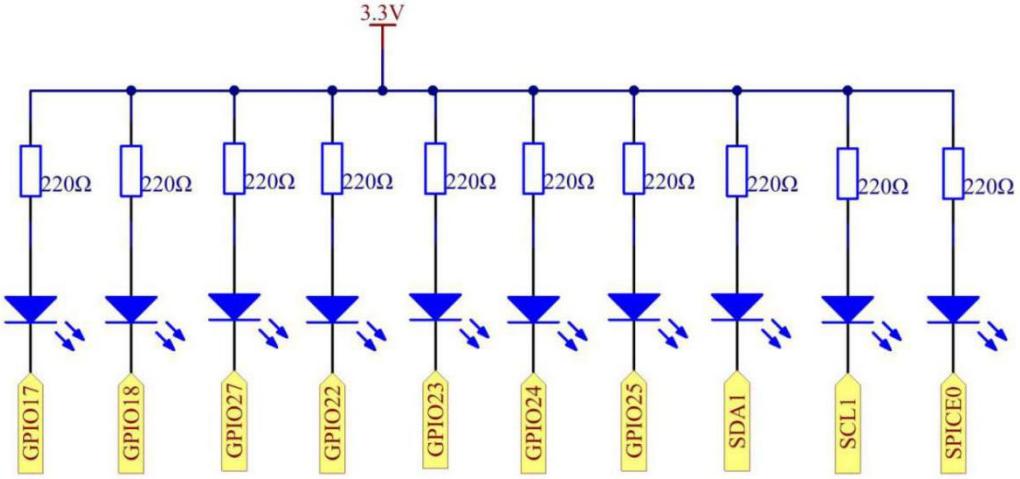
LED Bar Graph is an LED array, which is used to connect with electronic circuit or microcontroller. It’s easy to connect LED bar graph with the circuit like as connecting 10 individual LEDs with 10 output pins. Generally we can use the LED bar graph as a Battery level Indicator, Audio equipments, and Industrial Control panels. There are many other applications of LED bar graphs.



Schematic Diagram

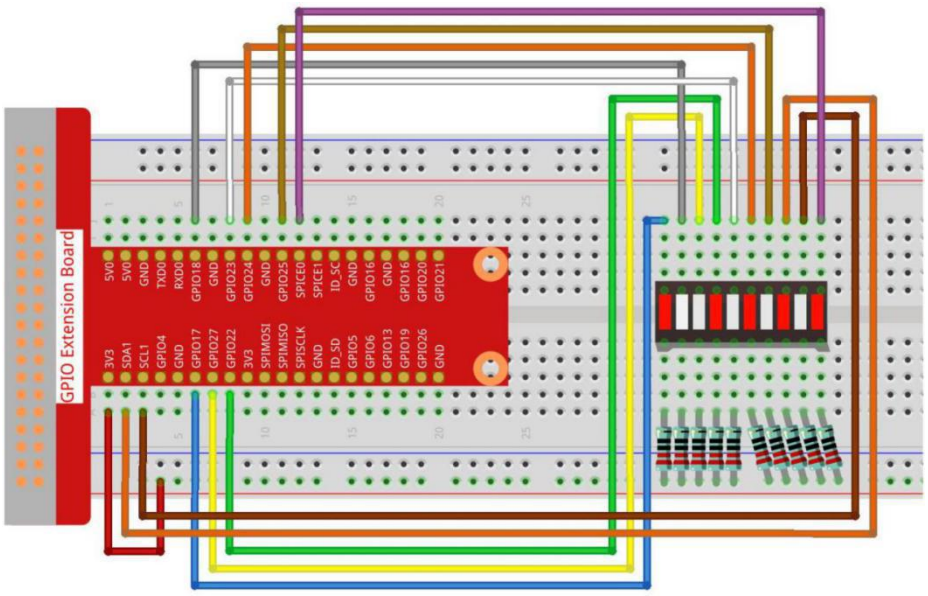
T-Board Name	physical	wiringPi	BCM
GPIO17	Pin 11	0	17

GPIO18	Pin 12	1	18
GPIO27	Pin 13	2	27
GPIO22	Pin 15	3	22
GPIO23	Pin 16	4	23
GPIO24	Pin 18	5	24
GPIO25	Pin 22	6	25
SDA1	Pin 3	8	2
SCL1	Pin 5	9	3
SPICE0	Pin 24	10	8



Experimental Procedures

Step 1: Build the circuit.



For C Language Users

Step 2: Go to the folder of the code.

```
cd /home/pi/REXQualis_Raspberry_Pi_Complete_Starter_Kit/C/6.LED_Bar_Graph
```

Step 3: Compile the code.

```
gcc 6.LED_Bar_Graph.c -o LED_Bar_Graph.out -lwiringPi
```

Step 4: Run the executable file.

```
sudo ./LED_Bar_Graph.out
```

After the code runs, you will see the LEDs on the LED bar lit one by one from left to right, and then lit one by one from right to left.

Code

```
#include <wiringPi.h>
#include <stdio.h>
int pins[10] = {0,1,2,3,4,5,6,8,9,10};
int main(void)
{
    int i;
    if(wiringPiSetup() == -1){ //when initialize wiring failed,print message to screen
        printf("setup wiringPi failed !");
        return 1;
    }
    for(i=0;i<=10;i++){            //make led pins' mode is output
        pinMode(pins[i], OUTPUT);
    }
    while(1){
        for(i=0;i<=10;i++){     //make led on from left to right
            if(i==7){continue;} //skip pin 7
            digitalWrite(i, LOW);
            delay(100);
```

```
            digitalWrite(i, HIGH);
        }
        for(i=10;i>-1;i--){     //make led on from right to left
            if(i==7){continue;}
            digitalWrite(i, LOW);
            delay(100);
            digitalWrite(i, HIGH);
        }
    }
    return 0;
}
```

Code Explanation

```
    for(i=0;i<=10;i++){          //make led pins' mode is output
        pinMode(pins[i], OUTPUT);
    }
```

Employ a for loop to set the corresponding pins to OUTPUT, and the pin[] array stores the pin numbers that are connected to the LED bar graph.

```
        for(i=0;i<=10;i++){     //make led on from left to right
            if(i==7){continue;} //skip pin 7
            digitalWrite(i, LOW);
            delay(100);
            digitalWrite(i, HIGH);
        }
```

A for loop is employed to turn the lights on and off once on the LED bar graph, from left to right.

```
            if(i==7){continue;}
```

In this case, we don't use pin7, instead we use continue; let's skip the case where i is 7.

Note: The break statement terminates the loop, whereas continue statement forces the

next iteration of the loop.

For Python Language Users

Step 2: Go to the folder of the code.

```
cd /home/pi/REXQualis_Raspberry_Pi_Complete_Starter_Kit/Python
```

Step 3: Run the executable file.

```
sudo python3 6.LED_Bar_Graph.py
```

After the code runs, you will see the LEDs on the LED bar lit one by one from left to right, and then lit one by one from right to left.

Code

```
import RPi.GPIO as GPIO
import time
ledPins = [11, 12, 13, 15, 16, 18, 22, 3, 5, 24]
def setup():
    GPIO.setmode(GPIO.BOARD)        # Numbers GPIOs by physical location
    for pin in ledPins:
    GPIO.setup(pin, GPIO.OUT)    # Set all ledPins' mode is output
    GPIO.output(pin, GPIO.HIGH) # Set all ledPins to high(+3.3V) to off led
def loop():
    while True:
        for pin in ledPins:        #make led on from left to right
            GPIO.output(pin, GPIO.LOW)
            time.sleep(0.1)
            GPIO.output(pin, GPIO.HIGH)
        for pin in ledPins[10:0:-1]:            #make led on from right to left
            GPIO.output(pin, GPIO.LOW)
            time.sleep(0.1)
            GPIO.output(pin, GPIO.HIGH)
def destroy():
```

```
    for pin in ledPins:
        GPIO.output(pin, GPIO.HIGH)      # turn off all leds
        GPIO.cleanup()                   # Release resource
if __name__ == '__main__':       # Program start from here
    setup()
    try:
        loop()
    except KeyboardInterrupt:  # When 'Ctrl+C' is pressed, the program destroy() will be executed.
        destroy()
```

Code Explanation

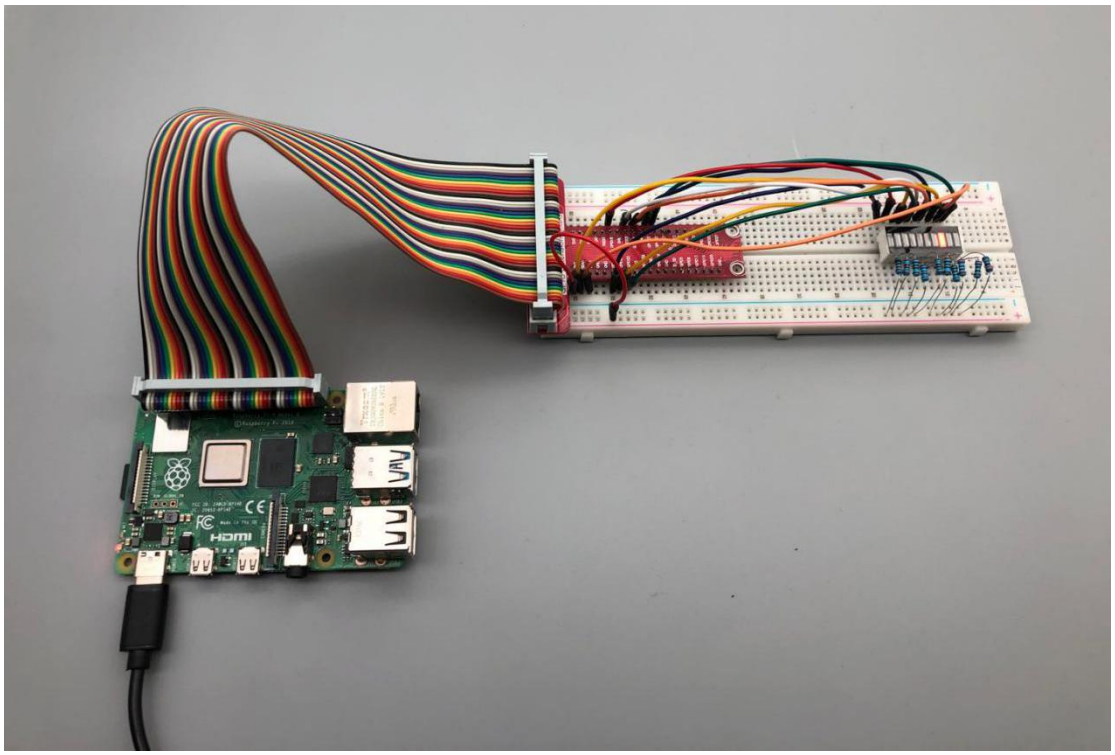
```
def setup():
    GPIO.setmode(GPIO.BOARD)        # Numbers GPIOs by physical location
    for pin in ledPins:
        GPIO.setup(pin, GPIO.OUT)   # Set all ledPins' mode is output
        GPIO.output(pin, GPIO.HIGH) # Set all ledPins to high(+3.3V) to off led
```

Set the numbering mode of GPIO to BOARD, and employ a for loop to set the corresponding pins to “output”. By the way, the array ledPins stores the numbering that is connected to the pins of LED bar graph.

```
def destroy():
    for pin in ledPins:
        GPIO.output(pin, GPIO.HIGH)      # turn off all leds
        GPIO.cleanup()                   # Release resource
```

A for loop is used to turn the lights on and off once on the LED bar graph, from left to right.

Phenomenon Picture


HDMI