Led

Overview

This course will use Raspberry Pi to control the blinking of led lights

Experimental Materials:

Raspberry Pi *1

5mm red LED light *1

T-type expansion board *1

220 ohm resistor *1

Breadboard*1

Some DuPont lines

Product description：

- ● Function: led is a light emitting device
- ● Application: LED advertising signboard lights, LED street lights, etc.

Technical Parameters:

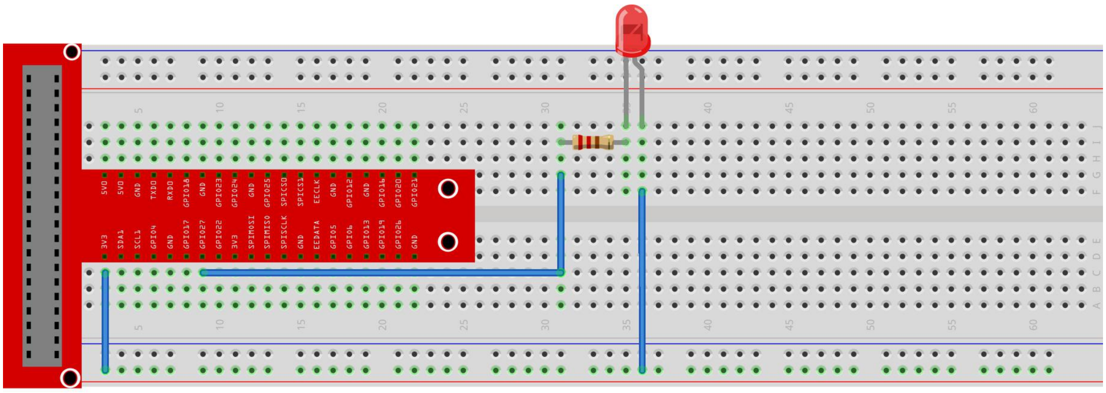
Lens: 5mm Diameter / Frosted / Round

Emitting Color: Red

Viewing Angle: 120 Degree

Polarity: Anode (Longer Part) | Cathode (Shorter Part)

Wiring diagram:



C code:

```
#include <wiringPi.h>
#include <stdio.h>

#define LedPin     2

int main(void)
{
    // When initialize wiring failed, print messageto screen
    if(wiringPiSetup() == -1){
        printf(”setup wiringPi failed !”);
        return 1;
    }

    pinMode(LedPin, OUTPUT);

    printf(”\n”);
    printf(”\n”);
    printf(”========================================\n”);
    printf(”|              Blink LED               |\n”);
    printf(”|    ------------------------------    |\n”);
    printf(”|         LED connect to GPIO2         |\n”);
    printf(”|                                      |\n”);
    printf(”|        LED will Blink at 500ms       |\n”);
    printf(”|                                      |\n”);
    printf(”========================================”);
    printf(”\n”);
    printf(”\n”);

    while(1){
        // LED on
        digitalWrite(LedPin, LOW);
        printf(”...LED on\n”);
        delay(500);
        // LED off
        digitalWrite(LedPin, HIGH);
        printf(”LED off...\n”);
        delay(500);
    }

    return 0;
}
```

Python code：

```
#!/usr/bin/env python

import RPi.GPIO as GPIO
import time

# Set #17 as LED pin
```

```
LedPin = 27

# Define a function to print message at the beginning
def print_message():
    print ("========================================")
    print ("|              Blink LED               |")
    print ("|    ------------------------------    |")
    print ("|         LED connect to GPIO27        |")
    print ("|                                      |")
    print ("|        LED will Blink at 500ms       |")
    print ("|                                      |")
    print ("========================================\n")
    print 'Program is running...'
    print 'Please press Ctrl+C to end the program...'
    raw_input ("Press Enter to begin\n")

# Define a setup function for some setup
def setup():
    # Set the GPIO modes to BCM Numbering
    GPIO.setmode(GPIO.BCM)
    # Set LedPin's mode to output,
    # and initial level to High(3.3v)
    GPIO.setup(LedPin, GPIO.OUT, initial=GPIO.HIGH)

# Define a main function for main process
def main():
    # Print messages
    print_message()
    while True:
        print 'LED ON'
        # Turn on LED
        GPIO.output(LedPin, GPIO.LOW)
        time.sleep(0.5)
        print 'LED OFF'
        # Turn off LED
        GPIO.output(LedPin, GPIO.HIGH)
        time.sleep(0.5)

# Define a destroy function for clean up everything after
# the script finished
def destroy():
    # Turn off LED
    GPIO.output(LedPin, GPIO.HIGH)
    # Release resource
    GPIO.cleanup()

# If run this script directly，do:
if __name__ == ’__main__’:
    setup()
    try:
        main()
    # When ’Ctrl+C’ is pressed，the child program
    # destroy() will be  executed.
    except KeyboardInterrupt:
        destroy()
```

Experimental results:

In the directory where the code file is located, execute the following command

C：

```
gcc -Wall -o blinkLed blinkLed.c -lwiringPi
sudo ./blinkLed
```

Python:

```
python blinkLed.py
```

After the instruction is executed, the led light will blink