Light blocking sensor

Overview

This course will use the Raspberry Pi to capture the light-blocking sensor signal, which control the LED light on and off.

Experimental Materials

RaspberryPi \*1

Breadboard \*1

Light block module \*1

Led \*1

Dupont Line

Ready to work

1. Install python interpreter in your Raspberry Pi system

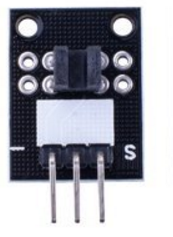
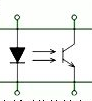
2. Install the RPi.GPIO library in your Raspberry Pi system

3. Install the wiringPi library in your Raspberry Pi system

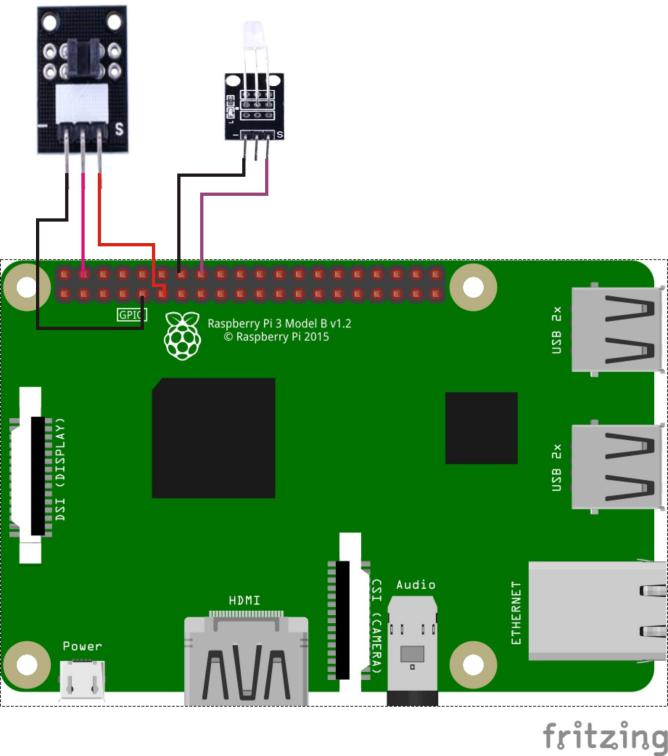
Refer to the attached "Installing a Python Interpreter and Corresponding Libraries in a Raspberry Pi System" for details.

product description

The light blocking sensor is consist of parallel light-emitting slice and light-receiving slice. There is a PNP phototransistor inside. The base electrode of the phototransistor is connected to the light-receiving slice , the collector is connected to the S pin, and the emitter electrode is connected to the GND pin. The S-pin of the light blocking sensor module is connected to VCC via a pull-up resistor, so when there is no blockage, the level of S-pin is at a high level. When the light blocking sensor is blocked, the light-receiving film cannot receive light, then the base electrode of the PNP phototransistor becomes a low level, the collector and the emitter electrode are turned on, which is equivalent to grounding the S pin, so the level of the S pin becomes low level. Therefore, we can do the corresponding actions by the level change of the S pin of the Raspberry Pi. Light blocking sensors are widely used in many areas such as city security, community, factories, companies, schools, families, villas, warehouses, resources, petroleum, chemicals, gas transmission and so on.

Wiring diagram



Sample code

1. python code

#!/usr/bin/env python

import RPi.GPIO as GPIO

import time

LightBreakPin = 11

LedPin = 16

def setup():

GPIO.setmode(GPIO.BOARD) # Numbers GPIOs by physical location

GPIO.setup(LedPin, GPIO.OUT) # Set LedPin's mode is output

GPIO.setup(LightBreakPin, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP)

GPIO.output(LedPin, GPIO.LOW) # Set LedPin low to off led

def loop():

while True:

if(GPIO.input(LightBreakPin) == 0):

print 'Be covered....'

GPIO.output(LedPin, GPIO.HIGH)

else:

GPIO.output(LedPin, GPIO.LOW)

time.sleep(0.2)

def destroy():

GPIO.output(LedPin, GPIO.LOW) # led off

GPIO.cleanup() # Release resource

if \_\_name\_\_ == '\_\_main\_\_': # Program start from here

setup()

try:

loop()

except KeyboardInterrupt: # When 'Ctrl+C' is pressed, the child program destroy() will be executed.

destroy()

1. C code

#include <wiringPi.h>

#include <stdio.h>

#define LightBreakPin 0

#define LedPin 4

int main(void)

{

if(wiringPiSetup() == -1)

{

printf("setup wiringPi failed !");

return -1;

}

pinMode(LightBreakPin, INPUT);

pullUpDnControl(LightBreakPin, PUD\_UP);

pinMode(LedPin, OUTPUT);

while(1)

{

if(digitalRead(LightBreakPin) == LOW)

{

printf("Be covered....\n");

digitalWrite(LedPin, HIGH); //led on

}

else

{

digitalWrite(LedPin, LOW); //led off

}

}

return 0;

}

Experimental phenomena

When the light blocking sensor between the light-emitting film and light-receiving film is blocked, the LED lamp is lit and vice versa.