Knock Switch

Overview  
 This course will use the Raspberry Pi to get the signal of the knock switch and control the LED light on and off.

Experimental Materials

RaspberryPi \*1

breadboard \*1

Knock Switch \*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  
See the attached "Installing a Python Interpreter and Corresponding Libraries in a Raspberry Pi System" for details.

Product description

I. Introduction:

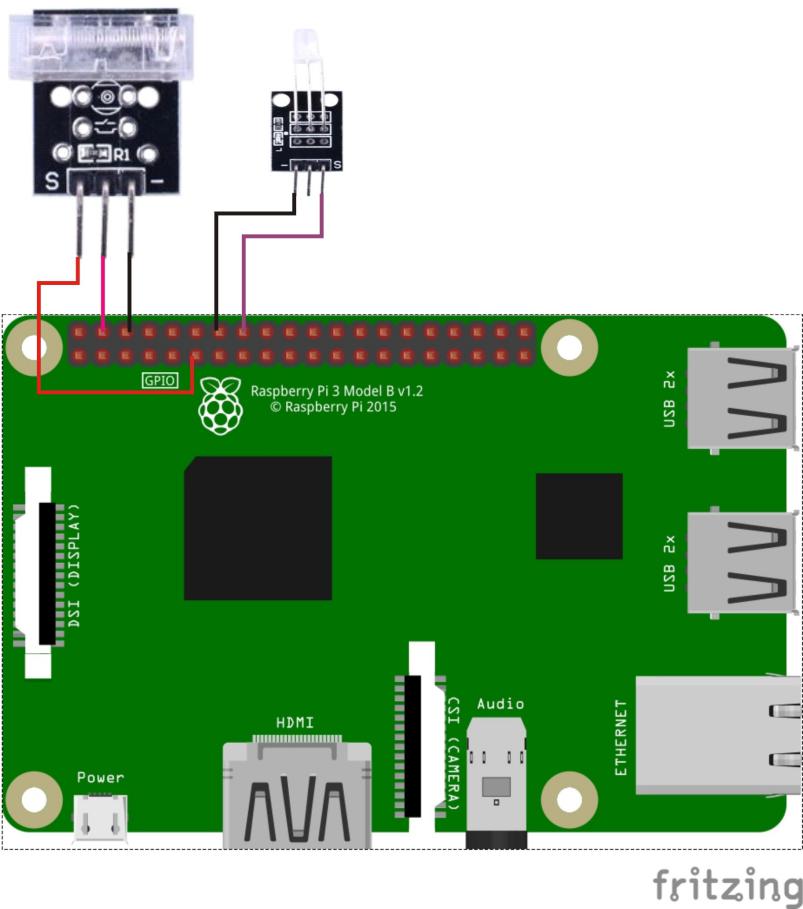
The knock switch is a device that converts the detected tap signal into a corresponding electrical signal.The working principle is that when the knock switch is correctly knocked, the spring in the switch will come into contact with the near metal plate. At this time, the circuit is turned on and the level of the signal pin changes.  
A knock signal is captured by detecting the level change of the signal pin.  
And thus,it is possible to control the corresponding circuit behavior by tapping the switch.



Characteristic parameters

◆ Module soldered 1KΩ pull-up resistor  
◆ Pull-up resistor connects to 5V

Wiring diagram



Sample code

1. Python code

#!/usr/bin/env python

import RPi.GPIO as GPIO

KnockPin = 11

LedPin = 16

Led\_status = 0

def setup():

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

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

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

def swLed(ev=None):

global Led\_status

Led\_status = not Led\_status

GPIO.output(LedPin, Led\_status) # switch led status(on-->off; off-->on)

print "LED: " + ("on" if Led\_status else "off")

def loop():

GPIO.add\_event\_detect(KnockPin, GPIO.FALLING, callback=swLed, bouncetime=200) # wait for falling

while True:

pass # Don't do anything

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 KnockPin 0

#define LedPin 4

int knockPinValue = -1;

int main(void)

{

int knockValue = -1;

if(wiringPiSetup() == -1)

{

printf("setup wiringPi failed !");

return 1;

}

pinMode(KnockPin, INPUT);

pinMode(LedPin, OUTPUT);

while(1)

{

knockValue = digitalRead(knockPin);

knockPinValue = knockValue;

delay(6);

knockValue = digitalRead(knockPin);

if(knockPinValue != knockValue)

{

printf("Detected knocking!\n");

digitalWrite(LedPin, !digitalRead(LedPin));

}

}

return 0;

}

Experimental phenomena

When a knock switch detects a knock, the status of the LED light will change once, so that the LED light can be turned on or off by knock switch.