 New type normally closed vibration sensor module, alarm module SW-420

**Usage:**

The utility model is used for various vibration triggering functions, reporting theft alarms, smart cars, earthquake alarm, motorcycle alarm, etc..

Compared with the normally open vibration sensor module, the vibration triggering time of the module is longer, and the relay module can be driven

**Module features:**

1、Used the SW-420 normal closed vibration sensor which is produced by our company

2、Comparator output, clean signal, good waveform, driving ability, more than 15mA

3、Operating voltage 3.3V-5V

4、Output form: digital switch output (0 and 1)

5、Bolt holes are fixed to facilitate installation

6、Small size: 3.2cm x 1.4cm PCB

7、Use wide voltage LM393 comparator

**Module instructions:**

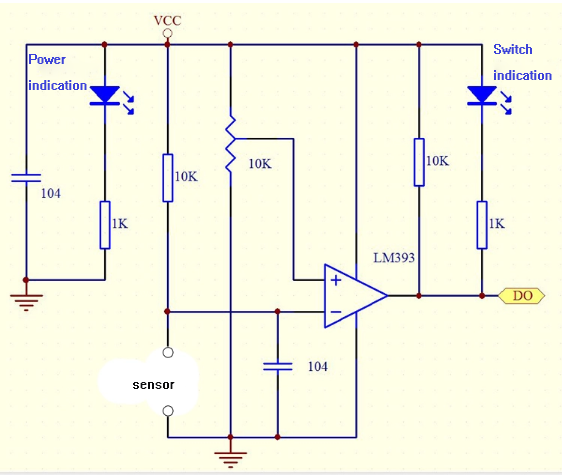
1, when the product does not vibrate, the vibration switch is in a closed conduction state, the output end is low, and the green indicator light is on;

2, when the product vibrates, the vibration switch is broken at once, the output end is high, and the green indicator light is not bright;

3, the output can be directly connected with the microcontroller, through the microcontroller to detect high and low levels, thus to detect whether the environment has a shock, play a warning role

**the reverse and positive surface of this moduleProduct structure and wiring diagram**

**Open source circuit diagram, friends can refer to the following picture**



**Singlechip test program**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
Sensor trigger test  
singlechip：STC89C52  
bit rate：9600  
Product uses: vibration induction trigger, alarm trigger.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
#include   
unsigned char date;  
#define uchar unsigned char  
#define uint unsigned int  
sbit key1=P0^1;

/\* Function declaration -----------------------------------------------\*/  
void delay(uint z);  
void Initial\_com(void);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*\* name ： delay(uint z)  
\*\* function ： delay function  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*/  
void delay(uint z)  
{  
uint i,j;  
for(i=z;i>0;i--)  
for(j=110;j>0;j--);  
}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\* Serial initialization function \*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
void Initial\_com(void)  
{  
EA=1; //enable interrupt

ES=1; //enable serial interrupt  
ET1=1; // enable timer T1 interrupt  
TMOD=0x20; // timer T1 interrupts the baud rate in mode 2  
PCON=0x00; //SMOD=0  
SCON=0x50; // Mode 1 is controlled by timer  
TH1=0xfd; //bit rate set to 9600  
TL1=0xfd;  
TR1=1; // Turn on timer T1 to run control bit

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
//\*\*\*\*\*\*\*\*\*\*main function\*\*\*\*\*\*\*\*\*  
//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
main()  
{  
Initial\_com();  
while(1)  
{  
  
if(key1==1)  
{  
delay(); // Jitter elimination  
if(key1==1) // Confirm trigger{  
SBUF=0X01;  
delay(200);  
  
  
}  
  
}  
  
if(RI)  
{  
date=SBUF; // SCM accept

SBUF=date; // SCM send

RI=0;  
}  
  
  
}  
}