

---

# 树莓派 3 风扇调速模块使用手册

适用树莓派 3 系列，

- 可调温控风扇
- 12 个 LED 全彩灯
- DS1302 实时时钟

## 可调温控风扇

### 编程

在/home/pi 目录下新建 fan.py 文件，将下面代码复制并保存到 fan.py 文件中。（fan-project 文件夹下有 fan.py 双击运行即可）

```
#!/usr/bin/env python
# encoding: utf-8

import RPi.GPIO
import time

RPi.GPIO.setwarnings(False)
RPi.GPIO.setmode(RPi.GPIO.BCM)
#我用的 gpio7 对应的 bcm 是 4，这里根据自己的接法修改
RPi.GPIO.setup(4, RPi.GPIO.OUT)
pwm = RPi.GPIO.PWM(4, 100)
RPi.GPIO.setwarnings(False)

speed = 0
prv_temp = 0
```

---

try:

while True:

```
    tmpFile = open( '/sys/class/thermal/thermal_zone0/temp' )
```

```
    cpu_temp = int(tmpFile.read())
```

```
    tmpFile.close()
```

```
    if cpu_temp >= 34500:
```

```
        if prv_temp < 34500 :
```

```
            #启动时防止风扇卡死先全功率转 0.1 秒
```

```
                pwm.start(0)
```

```
                pwm.ChangeDutyCycle(100)
```

```
                time.sleep(.1)
```

```
                speed = min( cpu_temp/125-257 , 100 )
```

```
                pwm.ChangeDutyCycle(speed)
```

```
            else :
```

```
                pwm.stop()
```

```
            prv_temp = cpu_tem
```

```
            time.sleep(5)
```

```
except KeyboardInterrupt:
```

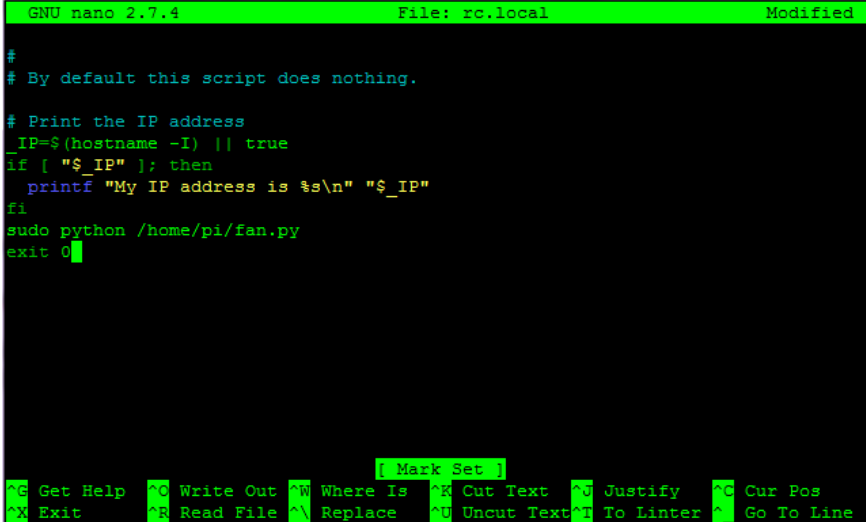
```
    pass
```

```
pwm.stop()
```

---

## 设置温控风扇开机自启。

把这个 python 脚本添加进 /etc/rc.local 中，保存退出。



```
GNU nano 2.7.4 File: rc.local Modified
#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
  printf "My IP address is %s\n" "$_IP"
fi
sudo python /home/pi/fan.py
exit 0
```

## LED

将库文件复制到 python 目录下

```
sudo cp -r blinkt.py/usr/lib/python3.5/blinkt.py
```

运行 fan-project 文件夹下的程序文件。（包括 :rainbow.py、random\_blink\_colours.py、solid\_colours.py）

## 实时时钟

1. 在系统中安装 wiringPi（fan-project 文件夹下有压缩包，解压，在解压好的目录里执行./build 安装）
2. 修改程序 ds1302.c 的 mian 函数里有一个 ds1302setup()函数，用来设置引脚，修改如图：

```

198 int main (int argc, char *argv [])
199 {
200     int i ;
201     int clock [8] ;
202
203     wiringPiSetup () ;
204     //ds1302setup (0, 1, 2) ;
205     ds1302setup (14, 30, 10) ;
206
207     if (argc == 2)
208     {
209         /**/ if (strcmp (argv [1], "-slc") == 0)
210             return setLinuxClock () ;
211         else if (strcmp (argv [1], "-sdsc") == 0)
212             return setDSclock () ;
213         else if (strcmp (argv [1], "-rtest") == 0)
214             return ramTest () ;
215         else
216         {
217             printf ("Usage: ds1302 [-slc | -sdsc | -rtest]\n") ;
218             return EXIT_FAILURE ;
219         }
220     }
}
ds1302.c

```

在 wiringPi/examples 目录下执行 `sudo nano ds1302.c` 修改 ds1302.c 文件 在 wiringPi/examples 目录下执行 `make ds1302` 即可生产可执行文件 ds1302

3. 执行 `sudo ./ds1302 -rtest`  
显示:

```

mm@mm-desktop: ~/gpio/wiringPi/examples$ sudo ./ds1302 -rtest
[sudo] password for mm:
DS1302 RAM TEST
-- DS1302 RAM TEST: OK

```

说明一切正常，否者请检查连接是否正确  
`sudo ./ds1302 -sdsc` 即可将系统时间写进 ds1302 模块中去

4. 从 DS 模块读取时间  
执行 `sudo ./ds1302`

```

mm@mm-desktop: ~/gpio/wiringPi/examples$ sudo ./ds1302
0:    2:09:02  8/10/2016
1:    2:09:03  8/10/2016
2:    2:09:03  8/10/2016
3:    2:09:03  8/10/2016
4:    2:09:03  8/10/2016
5:    2:09:03  8/10/2016
6:    2:09:04  8/10/2016
7:    2:09:04  8/10/2016
8:    2:09:04  8/10/2016
9:    2:09:04  8/10/2016

```

---

## 5. 从 DS 模块读取时间来设置系统的时间

```
sudo ./ds1302 -slc
```

### 开机同步时间

将树莓派的时间与网络同步后，就可以写入到 DS1302 中，这样 DS1302 具有准确的当前时间，就算掉电也不会丢失，在树莓派离线时，可以在每次开机时将 DS1302 的时间写入到树莓派中，从而使得树莓派实现了实时时钟的功能。

获得可执行文件 ds1302 的目录

```
pi@raspberrypi:~/wiringPi/examples $ pwd
```

```
/home/pi/wiringPi/examples
```

然后修改 `/etc/rc.local` 文件，在 `exit 0` 前面添加时间同步命令，这样在树莓派开机时就会自动将 DS1302 的时间同步到树莓派上。添加以下命令：

```
sudo /home/pi/wiringPi/examples/ds1302 -slc
```

保存后退出。我们将树莓派断网，然后树莓派关机，过一会开机，使用 `date` 命令读取时间，可以看到树莓派的走时和当前时间一致，说明时间自动同步成功！