

微雪课堂

WAVESHARE STUDY

请登录

用户名

密码

自动登录

请输入搜索内容

微雪课堂

树莓派

查看内容

树莓派系列教程8：如何点亮一个LED灯（下）

2015-8-27 10:22 | 发布者: MyMX1213 | 查看: 6539 | 评论: 1 | 原作者: MyMX1213

摘要: 本章讲解如果通过库函数编程操作树莓派GPIO

上一章我们讲解了在linux系统下如何通过读写设备文件的方式控制GPIO点亮LED灯，本章我们继续讲解如果通过使用中间层库函数编程控制GPIO。

一、wiringPi

WiringPi是应用于树莓派平台的GPIO控制库函数，WiringPi中的函数类似于Arduino的wiring系统。官网：<http://wiringpi.com/>

1、wiringPi安装

(1) 方案1-使用GIT工具

通过GIT获得wiringPi的源码

1 | git clone git://git.drogon.net/wiringPi

2 | cd wiringPi

3 | ./build

build脚本会帮助你编译和安装wiringPi

(2) 方案1-使用GIT工具

我们可以在网站上直接下载最新版本编译使用, <https://git.drogon.net/?p=wiringPi;a=summary>

Gordons Projects

→ Projects Top-Level GIT

wiringPi / summary

summary | shortlog | log | commit | compare | tree

description wiringPi - A "wiring" like library for the Raspberry Pi
owner: GIT
last change Mon, 9 Mar 2015 00:59:33 +0800 (16:59 +0800)

shortlog

2015-03-09 Gordon Henderson Updated the board types to cope with an 0014 version... master | commit | compare | tree | snapshot

2015-02-03 Gordon Henderson OK. So the Pi v2 I have had older firmware and it wasn... commit | commitdiff | tree | snapshot

2015-01-30 Gordon Henderson OK, as an easier way to manage versions. commit | commitdiff | tree | snapshot

2015-01-30 Gordon Henderson Updates to the build process commit | commitdiff | tree | snapshot

2015-01-29 Gordon Henderson Updated a technicality in softPwm, and added a suggeste... commit | commitdiff | tree | snapshot

2015-01-07 Gordon Henderson Updated versions for new maker - abest commit | commitdiff | tree | snapshot

2014-11-10 Gordon Henderson Minor changes to the PiCflow code - get the orange ... commit | commitdiff | tree | snapshot

2014-07-17 Gordon Henderson Fixed a bug in the gpio readall command on model B... commit | commitdiff | tree | snapshot

2014-07-14 Gordon Henderson Updated mostly to the gpio readall command to support... commit | commitdiff | tree | snapshot

2014-06-27 Gordon Henderson Fixed a small bug in the ISR code where it was looking... commit | commitdiff | tree | snapshot

2014-06-24 Gordon Henderson Bumped version to 2.15 commit | commitdiff | tree | snapshot

2014-06-24 Gordon Henderson Updates for the Raspberry Pi Compute Module - changes... commit | commitdiff | tree | snapshot

2014-05-20 Gordon Henderson changed to pin mode to support softPwm. commit | commitdiff | tree | snapshot

2013-09-03 Gordon Henderson Added some tweaks to gpio to set all modes on pins... commit | commitdiff | tree | snapshot

2013-07-20 Gordon Henderson Bumped version commit | commitdiff | tree | snapshot

2013-07-20 Gordon Henderson It helps if you add the files into GIT... commit | commitdiff | tree | snapshot

heads

4 months ago master shortlog | log | tree

wiringPi - A "wiring" like library for the Raspberry Pi

view raw

在windows上下载wiringPi库，并复制到树莓派中，运行如下命令解压安装。(xxx代表版本号)

1 | tar -zxvf wiringPi-xxx.tar.gz

2 | cd wiringPi-xxx

3 | ./build

详细安装教程请参考wiringPi官网：<http://wiringpi.com/download-and-install/>

2、测试

wiringPi包括一套gpio命令，使用gpio命令可以控制树莓派上的各种接口，通过以下指令可以测试wiringPi是否安装成功。

1 | gpio -v

2 | gpio readall

微雪课堂

树莓派

Arduino

C8051

PIC

STM8

FPGA

树莓派

01 Alphabot树莓派教程

lede是openwrt的一个分有支持pi3

02 Alphabot树莓派教程

03 树莓派系列教程18：：

04 树莓派系列教程17：：

05 树莓派系列教程16：：

06 树莓派系列教程15：：

07 树莓派系列教程14：：

08 树莓派系列教程13：：

09 树莓派系列教程12：：

010 树莓派系列教程11

011 树莓派系列教程10

012 树莓派系列教程9：

013 树莓派系列教程8：：

014 树莓派系列教程8：

015 树莓派教程系列7：

016 树莓派教程系列6：

017 树莓派系列教程5：

018 树莓派系列教程4：

019 树莓派系列教程3：

020 树莓派系列教程3：

021 树莓派系列教程3：

022 树莓派系列教程2：

023 树莓派系列教程1：

http://www.waveshare.net/study/article-601-1.html

1/4

```

pi pi@raspberrypi ~$ xshell 3
pi@raspberrypi ~$ gpio version: 2.26
Copyright (c) 2012-2015 Gordon Henderson
This is free software with ABSOLUTELY NO WARRANTY.
For details type: gpio -warranty

Raspberry Pi Details:
Type: Model 2, Revision: 1.1, Memory: 1024MB, Maker: Sony
pi@raspberrypi ~$ gpio readall
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi | Name | Mode | V | Physical | V | Mode | Name | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 2 | 8 | SDA.1 | ALTO | 1 | 3 | 4 | | 5v | 15 | 14 |
| 3 | 9 | SCL.1 | ALTO | 1 | 5 | 6 | | 5V | 16 | 15 |
| 4 | 7 | GPIO. 7 | IN | 0 | 7 | 8 | 1 | 0v | 1 | 18 |
| | | | | | | | | TxD | | |
| 17 | 0 | GPIO. 0 | OUT | 0 | 11 | 13 | 1 | 0v | 16 | 15 |
| 27 | 2 | GPIO. 2 | IN | 0 | 13 | 14 | 1 | 0v | 1 | 18 |
| 22 | 3 | GPIO. 3 | IN | 0 | 15 | 16 | 0 | 0v | 4 | 23 |
| | | | | | | | | GPIO. 4 | 5 | 24 |
| 10 | 12 | MOSI | ALTO | 0 | 17 | 18 | 0 | 0v | 6 | 25 |
| 9 | 13 | MISO | ALTO | 0 | 19 | 20 | 0 | 0v | 10 | 8 |
| 11 | 14 | SCLK | ALTO | 0 | 21 | 22 | 1 | 0v | 11 | 7 |
| | | | | | | | | GPIO. 6 | 31 | 1 |
| 0 | 30 | SDA.0 | IN | 1 | 23 | 24 | 1 | 0v | 26 | 12 |
| 5 | 21 | GPIO. 21 | OUT | 1 | 25 | 26 | 1 | 0v | 27 | 16 |
| 6 | 22 | GPIO. 22 | IN | 1 | 27 | 28 | 1 | 0v | 28 | 20 |
| 13 | 23 | GPIO. 23 | IN | 0 | 29 | 30 | 0 | 0v | 29 | 21 |
| 19 | 24 | GPIO. 24 | IN | 0 | 31 | 32 | 0 | 0v | | |
| 26 | 25 | GPIO. 25 | IN | 0 | 33 | 34 | 0 | 0v | | |
| | | | | | | | | GPIO. 26 | 27 | 16 |
| | | | | | | | | GPIO. 27 | 28 | 20 |
| | | | | | | | | GPIO. 28 | 29 | 21 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi | Name | Mode | V | Physical | V | Mode | Name | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
pi@raspberrypi ~$

```

从上图可以知道树莓派管脚有三种编号方式，下面以Pioneer600扩展板的LED1为例，第一种为板上编号（37），即中间两列，表示第几号插针。第二中为wiringPi编码（25），使用wiringPi库编程时是用wiringPi编号，第三种为BCM编号，这个是更加BCM管脚来编号的，使用bcm2835库或python的RPi.GPIO编程时使用BCM编号。

我们可以使用gpio命令操作树莓派GPIO，下面我们可以gpio命令控制Pioneer600扩展的LED1。

```
1 | gpio -g mode 26 out
```

设置管脚为输出模式，-g参数表示是以BCM编号方式，如果去掉这个参数测以wiringPi编号方式，即为25。

```
1 | gpio -g write 26 1
```

设置管脚为高电平，点亮LED。

```
1 | gpio -g write 26 0
```

设置管脚为低电平，熄灭LED，

```
1 | gpio -g read 26
```

读取管脚当前状态

更多gpio命令请查看:<https://projects.drogon.net/raspberry-pi/wiringpi/the-gpio-utility/>

3、wiringPi程序：

```

01 | #include <wiringpi.h>
02 | char LED = 25;
03 |
04 | int main(void)
05 | {
06 |     if(wiringPiSetup() < 0) return 1;
07 |     pinMode (LED,OUTPUT) ;
08 |
09 |     while(1)
10 |     {
11 |         digitalWrite(LED, 1) ;
12 |         delay (200);
13 |         digitalWrite(LED, 0) ;
14 |         delay (200);
15 |     }
16 | }

```

使用vi将代码添加到led.c文件中，运行如下命令编译并执行程序。按Ctrl+C终止程序。

```

1 | gcc -Wall led.c -o led -lwiringPi
2 | sudo ./led

```

注：（1）-Wall 表示编译时显示所有警告，-lwiringPi表示编译时动态加载wiringPi库

（2）终止程序后，LED的状态为不确定，这和python程序相比显得有点不足。

关于更多wiringPi的库函数，可参看wiringPi官网，也可参考wiringPi用户手册。

http://wenku.baidu.com/link?url=U_APbVE_ga5pSSwPwWABIGJymLVvyC-0W9AEOT2cjhLZzoLywa0-QpEIYNT2yHvNV0P7BbqTZCgG0ctaQZLi_ovkAGXREBOE6h68eTt-Q3y

bcm2835库是树莓派cpu芯片的库函数，相当于stm32的固件库一样，底层是直接操作寄存器。而wiringPi库和python的RPi.GPIO库其底层都是通过读写linux系统的设备文件操作设备。

1、安装bcm2835库

从bcm2835官网 (<http://www.airspayce.com/mikem/bcm2835/>) 下载最新版本的库，然后解压安装。

```
1 tar -zxvf bcm2835-1.xx.tar.gz
2 cd bcm2835-1.xx
3 ./configure
4 make
5 sudo make check
6 sudo make install
```

2、示例程序

新建名为led.c的文件，添加如下程序。

```
01 #include <bcm2835.h>
02
03 #define PIN 26
04 int main(int argc, char **argv)
05 {
06     if (!bcm2835_init())return 1;
07     bcm2835_gpio_fsel(PIN, BCM2835_GPIO_FSEL_OUTP);
08
09     while (1)
10     {
11         bcm2835_gpio_write(PIN, HIGH);
12         bcm2835_delay(500);
13         bcm2835_gpio_write(PIN, LOW);
14         bcm2835_delay(500);
15     }
16     bcm2835_close();
17     return 0;
18 }
```

编译并执行程序，按Ctrl+C可结束程序。

```
1 gcc -Wall led.c -o led -lbcm2835
2 sudo ./led
```

注：（1）-lbcm2835表示动态加载bcm2835库

（2）注意bcm2835程序管脚使用bcm编号，和wiringPi编号不一样。

（3）和wiringPi一样，程序结束时GPIO的状态不确定。

三、python

1、安装RPi.GPIO

（1）先安装python-dev,输入以下指令。

```
1 sudo apt-get install python-dev
```

（2）安装RPi.GPIO

```
1 <span style="font-size: 9pt; line-height: 1.8em;">#下载
2 wget https://pypi.python.org/packages/source/R/RPi.GPIO/RPi.GPIO-0.5.11.tar.gz
3 #解压：
4 tar -zxvf RPi.GPIO-0.5.11.tar.gz
5 #进入解压之后的目录：
6 cd RPi.GPIO-0.5.3a
7 #启动安装：
8 sudo python setup.py install</span>
```

2.示例程序

新建led.py文件，添加如下代码并保存。

```
01 #!/usr/bin/python
02 # -*- coding:utf-8 -*-
03 import RPi.GPIO as GPIO
04 import time
05
06 LED = 26
07
08 GPIO.setmode(GPIO.BCM)
09 GPIO.setup(LED,GPIO.OUT)
10 try:
11     while True:
12         GPIO.output(LED,GPIO.HIGH)
13         time.sleep(1)
14         GPIO.output(LED,GPIO.LOW)
15         time.sleep(1)
16 except:
17     print("except")
18 GPIO.cleanup()
```

执行程序，按Ctrl+C结束程序

```
1 sudo python led.py
```

注：（1）#!/usr/bin/env python，定义python解析脚本的绝对路径。

（2）# -*- coding: utf-8 -*-，python文件为utf-8格式，否则无法写入中文注释。

（3）GPIO.setmode(GPIO.BCM)，采用bcm编号方式。

(4) python程序使用try except语言，当按下Ctrl+C结束程序是会触发

异常，程序执行gpio.cleanup() 语句清楚GPIO管脚状态。

对python语言不是很熟悉的孩童，可以查看Python基础教

程：<http://www.runoob.com/python/python-tutorial.html>

这里有一个python语言的wiringPi库，有兴趣的可以参考：<https://github.com/WiringPi/WiringPi2-Python>

56



顶一下



刚表态过的朋友 (56 人)

邀请 收藏

上一篇：树莓派系列教程8：如何点亮一个LED灯(上)

下一篇：树莓派系列教程9：按键

相关阅读

- 树莓派系列教程2：树莓派烧写镜像
- 树莓派教程系列6：文件共享(samba)
- 树莓派系列教程3：访问树莓派之SSH篇
- 树莓派教程系列7：wiringPi、bcm2835、python库安装
- 树莓派系列教程3：访问树莓派之WiFi篇

最新评论

发表评论

Ezio_神仙醋 2017-1-19 15:15

[引用](#)

stydy

[查看全部评论\(1\)](#)

验证码 [换一个](#)

评论