

公告

你好，这里是Vamei，一名编程爱好者。
我在博客里写了Python/Linux/网络协议/算法/Java/数据科学系列文章，[从这里开始阅读](#)。非常期待和你的交流。

编程相关出版作品：

树莓派开始，玩转Linux
从Python开始学编程
协议森林

其他作品：

现代小城的考古学家
天气与历史的相爱相杀
随手拍光影

我的豆瓣

昵称： Vamei
园龄： 9年5个月
荣誉： 推荐博客
粉丝： 9940
关注： 27
+加关注

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

Python(62)
Java(42)
Linux(31)
大数据(24)
网络(18)
文青(17)
算法(15)
技普(9)
系列索引(6)
平台(4)
更多

积分与排名

积分 - 760172
排名 - 446

系列文章

数据科学
树莓派与Linux
Python快速教程

树莓派的GPIO编程

作者:Vamei 出处:http://www.cnblogs.com/vamei 严禁转载。

树莓派除了提供常见的网口和USB接口，还提供了一组GPIO(General Purpose Input/Output)接口。这组GPIO接口大大拓展了树莓派的能力。GPIO不仅能实现通信，还能直接控制电子元器件，从而让用户体验到硬件编程的乐趣。

GPIO简介

在树莓派3上，GPIO接口由40个引脚(PIN)组成。每个引脚都可以用导线和外部设备相连。你可以通过焊接的方式来把导线固定在PIN上，也可以用母型的跳线套接在PIN上。



跳线

40个PIN中，有固定输出的5V(2、4号PIN)、3.3V(1、17号PIN)和地线(Ground, 6、9、14、20、25、30、34、39)。如果一个电路两端接在，5V和地线之间，该电路就会获得5V的电压输入。27和28号PIN标着ID_SD和ID_SC。它们是两个特殊的PIN。它们属于ID EEPROM (Electrically Erasable Programmable Read-Only Memory) 接口，用于和拓展树莓派功能的附加电路板通信。其他的PIN大多编程GPIOX的编号，如GPIO14。树莓派的操作系统中，会用GPIO的编号14来指代这个PIN，而不是位置编号的8。有一些PIN除了GPIO功能外，还提供了高级端口功能。比如说，GPIO14和GPIO15就同时可以充当UART端口。此外，GPIO上还能找到I2C和SPI端口。



Pin#	NAME		NAME	Pin#
01	3.3v DC Power	■	DC Power 5v	02
03	GPIO02 (SDA1 , I²C)	●	DC Power 5v	04
05	GPIO03 (SCL1 , I²C)	●	Ground	06
07	GPIO04 (GPIO_GCLK)	●	(TXD0) GPIO14	08
09	Ground	●	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	●	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	●	Ground	14
15	GPIO22 (GPIO_GEN3)	●	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	●	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	●	Ground	20
21	GPIO09 (SPI_MISO)	●	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	●	(SPI_CE0_N) GPIO08	24
25	Ground	●	(SPI_CE1_N) GPIO07	26
27	ID_SD (I²C ID EEPROM)	●	(I²C ID EEPROM) ID_SC	28
29	GPIO05	●	Ground	30
31	GPIO06	●	GPIO12	32
33	GPIO13	●	Ground	34
35	GPIO19	●	GPIO16	36
37	GPIO26	●	GPIO20	38

推荐排行榜

- 1. “不给力啊，老湿！”：RSA加密与破解(233)
- 2. Python快速教程(202)
- 3. Java快速教程(122)
- 4. 野蛮生长又五年(112)
- 5. 协议森林01 邮差与邮局 (网络协议概观)(109)

最新评论

- 1. Re:概率论03 条件概率

原来大神很早之前就在文章里透露了很多信息啊，辛苦了，R.I.P

--ghost_yl

- 2. Re:概率论03 条件概率

感恩 R.I.P

--ghost_yl

- 3. Re:Java基础08 继承

谁有自己做的笔记吗？分享一下

--hsaww

- 4. Re:WordPress快速建站

写的很详细，赞一个

--cloudSer

- 5. Re:从写博到出书：过程全记录

@Revolinux 现在怎么样啦...

---一只小猪叫嘟嘟

- 6. Re:一个程序员被骗去养猪

R.I.P

--DustyPosa

- 7. Re:Python基础07 函数

```
def isleapyear(year, month, day): if
    year % 4 == 0 and year % 100 !=
        0: return True else: return Fal...
```

--Jason_Wayne

- 8. Re:纸上谈兵: 算法与数据结构

逝者安息

--滚去学习！！

- 9. Re:Python简史

新手报到

--conlin_lc

- 10. Re:Java快速教程

很好，简单易懂

--惠明

- 11. Re:纸上谈兵: 算法与数据结构

R.I.P

--DD斩首

- 12. Re:Python标准库12 数学与随机数

```
(math包，random包)
`import random seq=[] for i in
    range(8):
        j=random.choice(range(1,7))
        seq.append(j) #list包含数字，不能
        直接转化成...
```

---一季314

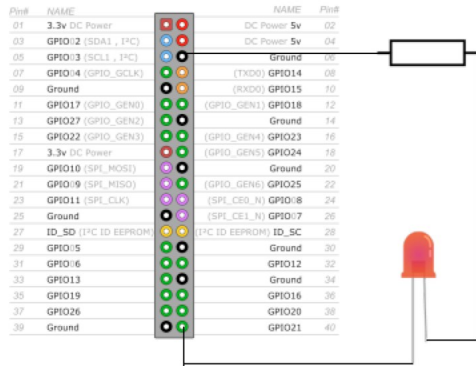
- 13. Re:Python简史

树莓派3的GPIO引脚

在计算机中，通常用高、低两个电压来表示二进制的1和0。树莓派也是如此。GPIO用相同的方式来表示数据。每个GPIO的PIN都能处于输入或输出状态。当处于输出状态时，系统可以把1或0传给该PIN。如果是1，那么对应的物理PIN向外输出3.3V的高电压，否则输出0V的低电压。相应的，处于输入状态的PIN可以探测物理PIN上的电压。如果是高电压，那么该PIN将向系统返回1，否则返回0。就是利用上述简单机制，GPIO实现了和物理电路的互动。

控制LED灯

我们先来看GPIO输出的一个例子。我们在GPIO21和地线之间接了一个串联电路。电路上有一个LED灯，还有一个用于防止短路的330欧姆电阻。当GPIO21位于高电平时，将有电流通过电路，从而点亮LED灯。



我们用bash命令来控制GPIO21。在Linux中，外部设备经常被表示成文件。向文件写入或读取字符，就相当于向设备输出或者从设备输入字符。树莓派上的GPIO端口也是如此，其代表文件位于/sys/class/gpio/下。首先，激活GPIO21：

```
echo 21 > /sys/class/gpio/export
```

这个命令的意思，是把字符"21"输入到/sys/class/gpio/export。可以看到，命令执行后，/sys/class/gpio/下面增加了代表GPIO21的一个目录，目录名就是gpio21。下一步，我们把GPIO21置于输出状态：

```
echo out > /sys/class/gpio/gpio21/direction
```

文件/sys/class/gpio/gpio21/direction用于控制GPIO21的方向。我们向里面写入了代表输出的字符"out"。最后，向GPIO21写入1，从而让PIN处于高电压：

```
echo 1 > /sys/class/gpio/gpio21/value
```

可以看到，LED灯亮了起来。如果想关掉LED灯，只需要向GPIO21写入0：

```
echo 0 > /sys/class/gpio/gpio21/value
```

使用完毕GPIO21，可以删除该端口：

```
echo 21 > /sys/class/gpio/unexport
```

/sys/class/gpio/gpio21随即消失。

两个树莓派之间的GPIO

我们可以用GPIO的方式连接两个树莓派。一个树莓派的GPIO输出，将成为另一个树莓派的GPIO输入。连接方式很简单，只需要2根导线。一个导线连接两个树莓派的地线，另一根导线连接树莓派的两个PIN：

我们用左侧的树莓派来输出，右侧树莓派来输入。输出过程和上面控制LED灯的例子相似。在第一个树莓派中的GPIO21准备输出：

新手报到！

--星光不问_赶路人

14. Re: [一个程序员被骗去养猪](#)

一路走好，只要博客园在，大佬就一直在。

--代码如风~~~

15. Re: [C编译: makefile基础](#)

从您的博客中受益匪浅，今天重新来到这里，才发现您已经去世了，愿大神一路走好

--Mars.wang

```
echo 21 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio21/direction
```

在第二个树莓派中，准备好读取GPIO26：

```
echo 26 > /sys/class/gpio/export
echo in > /sys/class/gpio/gpio26/direction
```

当我们往/sys/class/gpio/gpio26中写入"in"时，就把GPIO26置于输入状态。

此后，在第一个树莓派中，就可以更改输出值为1或0：

```
echo 1 > /sys/class/gpio/gpio21/value
echo 0 > /sys/class/gpio/gpio21/value
```

在第二个树莓派中，可以用cat命令来读取文件，获得输入值：

```
cat /sys/class/gpio/gpio26/value
```

由于cat命令读完一次后会返回。为了持续读取，我们可以用bash中的无限循环，来反复调用cat：

```
while true; do cat /sys/class/gpio/gpio26/value; done
```

随着第一个树莓派中输出的改变，第二个树莓派获得的输入也随之改变。我们在两个树莓派之间实现了简单的通信。

最后，在使用完GPIO后，别忘了删除端口。

UART编程

计算机的数据都是许多位的0和1构成的序列。尽管GPIO可以在0和1之间切换，但并不能准确地分割出位。比如说，我们把一个二进制序列11000111输出到GPIO端口，那么在输入端看来，只是输入了一段时间的1，然后变成0，然后又变成1。输入端没法准确说出，一段高电平输入究竟包好了几位1。

一个解决方案是用多个PIN同时通信，每个PIN表示一位。当输入端读取完成后，通知输出端，让输出端送来下面一批的数据。这种通信方式被称为并口传输。和并口对应的是串口传输。传输时依然是用一个PIN，但输入方可以知道一位的数据持续了多长时间。GPIO上的UART、I2C、SPI都是串口通信。

UART与其余两者的区别在于，通信双方通过事先约定的速率来发送或接受数据。这种通信方式称为异步通信。在I2C和SPI这样的同步通信方式，会用额外的连线来保证双方速率相同。UART的连线 and 实现方式很简单，成为最流行的串口通信方式。但UART的缺点在于，如果发送方和接收方的速率不同，那么通信就会发生错误。通信速率就称为“波特率”(baudrate)，单位是每秒通信的位数(bps)。

UART的端口至少有RX、TX和地线三个针脚。RX负责读取，TX负责输出。如果有两个UART端口，它们的连接方式如下：

在树莓派3的情况下，TX和RX就是GPIO14和GPIO15针脚。因此，我们可以把两个树莓派之间按照上图的方式连接起来，然后在两个树莓派之间实现UART通信。

在这里，我们要注意树莓派3发生的一点变化。树莓派1和2中都使用了标准的UART，在操作系统中的对应文件是/dev/ttyAMA0。在树莓派3中，新增的蓝牙模块占用了标准UART端口和树莓派沟通，外部的UART通信采用了简单的Mini UART，在操作系统中的对应文件是/dev/ttyS0。由于mini UART的波特率依赖于CPU时钟频率，而CPU频率可能在运行过程中浮动，因此mini UART经常会带来意向不到的错误。一般有两种解决方案有。一种是关闭蓝牙模块，让外部连接重新使用标准UART端口。另一种是固定CPU时钟频率，以便mini UART能以准确的波特率进行通信。

关闭蓝牙模块，需要修改/boot/config.txt，在文件末尾增加：

```
dtoverlay=pi3-disable-bt
```

修改后重启。此后的UART通信, 就可以通过/dev/ttyAMA0进行。

如果是采取第二种解决方案, 还是要修改/boot/config.txt, 上面的修改变成:

```
core_freq=250
dtoverlay=pi3-miniuart-bt
```

修改后重启。此后的UART通信, 就可以通过/dev/ttyS0进行。

我们以第一种解决方案为例, 进行UART通信。设定波特率:

```
stty -F /dev/ttyAMA0 9600
```

输出文本:

```
echo "hello" > /dev/ttyAMA0
```

读取文本:

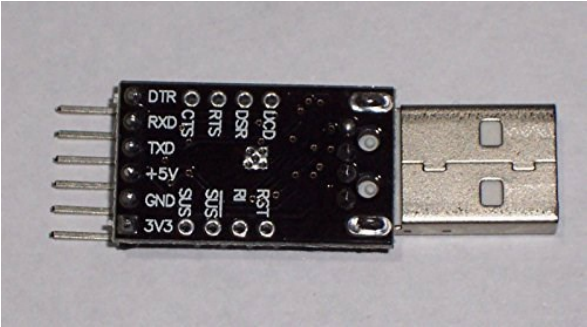
```
cat /dev/ttyAMA0
```

如果使用第二种解决方案, 那么只需要把上面的/dev/ttyAMA0改为/dev/ttyS0。

可以看到, UART可以实现更加复杂的文本通信。

用UART连接PC

一般的PC都没有暴露在外的UART针脚。为了通过UART来连接PC和树莓派, 我们需要一个USB和UART的转换器。这个转换器的一端是USB接口, 另一端是UART的针脚。我们把USB一端插入到PC。另一端按照UART到UART的方式, 连接到树莓派的UART针脚。



连接好之后, 就可以在PC上, 利用串口操作软件来和树莓派通信。在Linux下, USB连接表示为/dev/ttyUSB0。当然, 当计算机上只有1个USB设备时, 最后的编号才会是0。而在我的Mac OSX上, 该USB连接被表示成/dev/cu.SLAB_USBtoUART。此后, 就可以通过操作USB文件来进行UART通信。在Windows下, 也有现成的进行串口通信的图形化软件。

用UART登陆树莓派

我们还可以用UART的方式连接并登陆树莓派。进入树莓派设置:

```
sudo raspi-config
```

在Interfacing Options->Serial中, 允许开机时通过串口登陆。

重启后, 树莓派启动时会自动把开机信息已115200的波特率推到UART端口。在UART另一端的PC上, 如果你使用Mac OSX, 那么你可以用下面命令连接:

```
screen /dev/cu.SLAB_USBtoUART 115200
```

如果PC是Linux系统, 只需要把USB设备文件改为对应的设备文件即可。如果是Windows系统, 还可以用图形化软件。这里不再赘述。

欢迎阅读“[骑着企鹅采树莓](#)”系列文章

如果你喜欢这篇文章, 欢迎[推荐](#)。
技术推动进步, 分享促进社区。

分类: [Linux](#)

标签: [Linux](#)

[好文要顶](#)[关注我](#)[收藏该文](#)



Vamei
关注 - 27
粉丝 - 9940

推荐博客
[+加关注](#)

« [上一篇](#): [树莓派：设置与软件安装](#)


» [下一篇](#): [用树莓派玩转蓝牙](#)

5
[推荐](#)

0
[反对](#)

posted @ 2017-04-23 17:48 [Vamei](#) 阅读(23897) 评论(5) [编辑](#) [收藏](#) [举报](#)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

 登录后才能查看或发表评论, 立即 [登录](#) 或者 [逛逛](#) 博客园首页

编辑推荐:

- 2次心态变化和27个问题：机制落地的部分全貌与节奏控制
- CPU 被挖矿，Redis 竟是内鬼！
- 巧用滤镜实现高级感拉满的文字快闪切换效果
- 记一次 .NET 某公交卡扣费系统 程序卡死分析
- 设计系统，设计和开发之间的“DevOps”