

# 树莓派开发

## 19 树莓派与arduino通讯

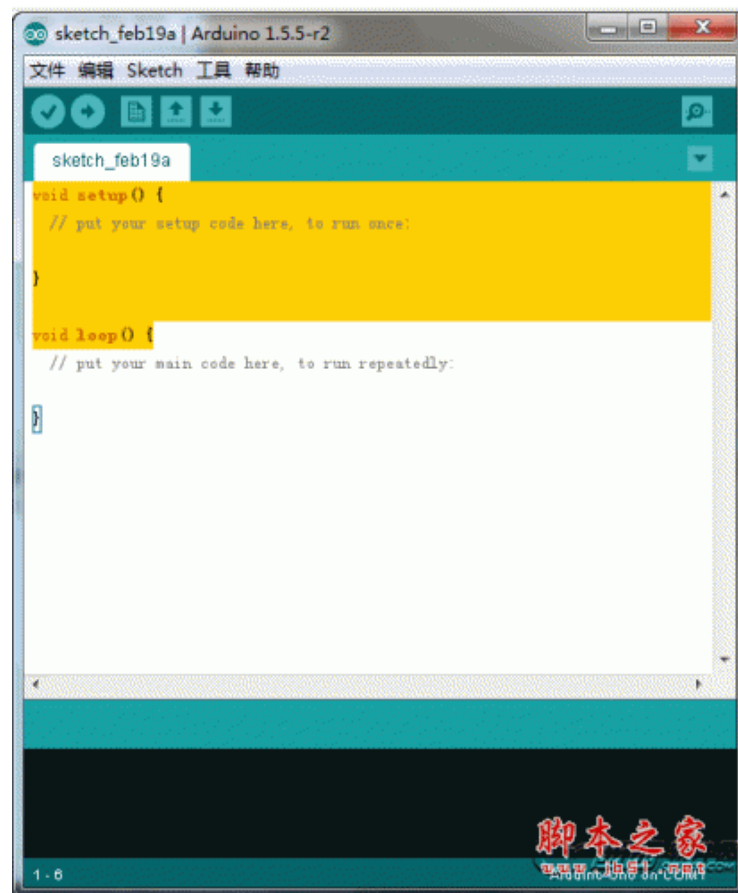
# 1、arduino介绍

❑ Arduino，是一个基于开放源码的软硬件平台，构建于开放源码simple I/O介面版，并且具有使用类似Java，C语言的Processing/Wiring开发环境。

❑ 包含：

❑ 硬件（有多种型号），左图：

❑ 软件（Arduino IDE), 右图：



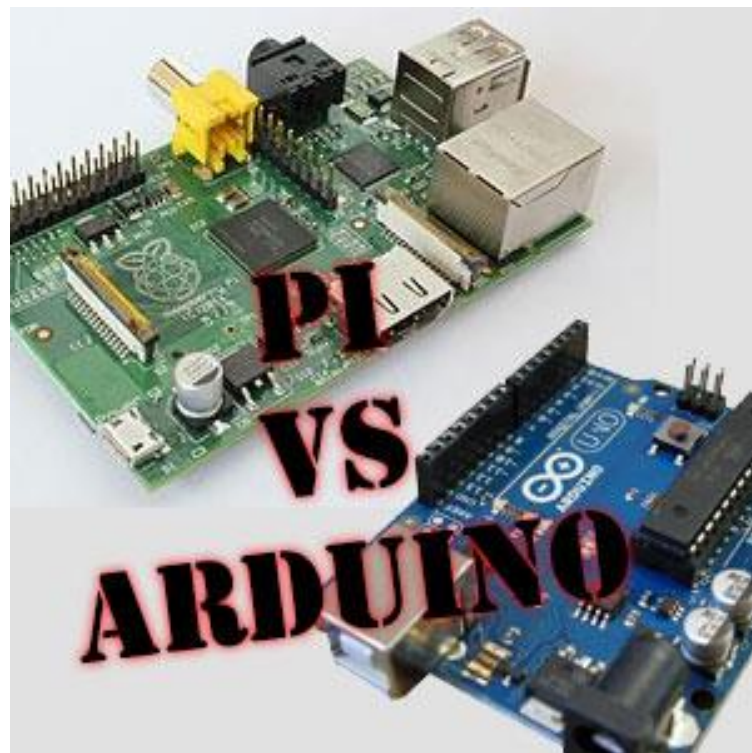
## 2、 arduino介绍——功能

- ❑ Arduino的硬件部分可以用来做电路连接；
- ❑ Arduino的软件部分则是 IDE——你的程序开发环境。
- ❑ Arduino能通过各种各样的传感器来感知环境，通过控制灯光、马达和其他的装置来反馈、影响环境。
- ❑ 板子上的微控制器可以通过Arduino的编程语言来编写程序，编译成二进制文件，写进微控制器（类似传统的51单片机）。
- ❑ 对Arduino的编程是利用 Arduino编程语言 (基于 Wiring)和 Arduino开发环境(基于 Processing)来实现的。
- ❑ 基于**Arduino**的项目，可以只包含**Arduino**，也可以包含**Arduino**和其他一些在**PC**（我们现在是树莓派）上运行的软件，他们之间进行通信 (比如 **Flash, Processing, MaxMSP**)来实现。

### 3、 arduino与树莓派PK

1、 RPi 和 Arduino 是两个完全不同运算水平的平台。毕竟 RPi 使用的是一颗运行在 700MHZ 的 ARM11 CPU；而常用的 Arduino UNO 则是一颗在工业控制领域常见的 8-bit ATmega328，最高运行频率仅 20MHz。

2、 RPi 可以运行完整的操作系统，如 Debian 等常见 Linux 发行版。这意味着你可以使用你熟练的语言（如 Python、Java）和熟悉的库来进行开发，同时后台运行多个进程也毫无压力。而 Arduino 作为一款真正传统意义上的单片机系统，一次只能运行一个你烧进去的程序，功能也相对单一。



### 3、 arduino与树莓派PK

3、RPI 自带的接口比较全面，USB-host、RJ45、HDMI、SD读卡器等常用接口都有；  
而 Arduino 比较单纯，与外部设备打交道需要另外采购接口板。

4、RPI 因为是一个相对完整的「电脑」，其成本肯定比 Arduino 高，目前 B 版在淘宝普遍 200 元左右；而常用的 Arduino UNO只需要 40 元。

5、RPI 拥有更完整的操作系统，这也意味着每次复电后所需的启动时间较长，包括需要有些相关的服务也能正常启动；  
但 Arduino 因为用途单一，重启速度超快，而且重启结束后立即就直接运行你的程序（没有操作系统）。



## 4、 arduino与树莓派的结合

□如果让RPI 作为中央控制服务器，负责与互联网的通信、采样存储 Arduino上报的状态数据、处理数据量大的工作（如音频、视频、图片相关）、提供 API 给 iOS 及 Android 以方便用手机控制家居。RPI 与 Arduino 间通过以太网或 Zigbee 进行数据传输。

□而Arduino 负责传感器采样，如光线强度、温度、人体红外感应等。有的模块比较独立，比如人体感应的夜间走廊灯，单个 Arduino 可以自己实现监控加控制，就不需要上报数据给 RPI 了。需要上报和接收数据的 Arduino ，通过 Zigbee 及以太网和 RPI 进行通信。

□两者的结合发挥了各自的优点，规避了缺点，可谓扬长避短、各得其所，算是物尽其用。

□还有一个更大的便利就是，简化了树莓派对外设的控制编程。

## 4、 arduino与树莓派的结合

### □ Zigbee

□ **ZigBee**是基于IEEE802.15.4标准的低功耗局域网协议。

□ 根据国际标准规定，**ZigBee**技术是一种短距离、低功耗的无线通信技术。

□ 这一名称（又称紫蜂协议）来源于蜜蜂的八字舞，由于蜜蜂(bee)是靠飞翔和“嗡嗡”(zig)地抖动翅膀的“舞蹈”来与同伴传递花粉所在方位信息，也就是说蜜蜂依靠这样的方式构成了群体中的通信网络。

□ 其特点是近距离、低复杂度、自组织、低功耗、低数据速率。

□ 主要适合用于自动控制和远程控制领域，可以嵌入各种设备。

□ 简而言之，**ZigBee**就是一种便宜的，低功耗的近距离无线组网通讯技术。

## 5、 arduino IDE

- ❑ Arduino IDE是Arduino的开放源代码的集成开发环境，其界面友好，语法简单以及能方便的下载程序，使得Arduino的程序开发变得非常便捷。
- ❑ 作为一款开放源代码的软件，Arduino IDE也是由Java、Processing、avr-gcc等开放源码的软件写成
- ❑ 其另一个最大特点是跨平台的兼容性，适用于Windows、Max OS X以及Linux。
- ❑ 2011年11月30号Arduino官方正式发布了Arduino1.0版本，可以下载不同系统下的压缩包，也可以在github（开源代码库）上下载源码重新编译自己的IDE。



## 6、 Windows系统下的arduino IDE

- 一种方法是在Windows下安装Linux环境，然后再安装Arduino IDE
- 这种模式必须先安装开发工具Cygwin、Java JDK、Ant。
  - 安装Cygwin（一个windows系统下的linux仿真环境）
    - 在[Cygwin官网](http://www.cygwin.cn/site/install/) <http://www.cygwin.cn/site/install/> 下载setup文件。
    - 安装选择需要的开发包
      - git—— 版本控制
      - make, gcc-mingw, and g++ —— 编译 arduino.exe
      - perl ——
      - unzip, zip ——
  - 安装Java JDK
    - 在[官网](#)下载安装软件，安装上Java JDK。
    - 配置Wingdows系统环境变量
      - JAVA\_HOME：选择安装路径，例如  
G:\Program\Java\jdk1.7.0\_01
      - Path：%JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin
      - CLASSPATH：  
.;%JAVA\_HOME%\lib\dt.jar;%JAVA\_HOME%\lib\tools.jar

# 6、 Windows系统下的arduino IDE

## □ 安装Ant

- 在[ant官网](#)下载安装包，解压。

- 配置Windows系统环境变量

ANT\_HOME: 选择解压包路径，例如G:\Program\Ant

Path: %ANT\_HOME%\bin

## □ 从Github上下载源代码

- 命令提示符下运行 `git clone git://github.com/arduino/Arduino.git`

## □ 编译

- 进入下载的arduino目录，`/path/arduino/build`

- 命令提示符下运行 `ant`，编译成功。

- 如果对源码进行了修改，运行 `ant run`会重新编译并且运行新的Arduino IDE开发环境。

## □ 更新版本

- Github上会发布Arduino的最新更改，可以运行以下命令更新版本

- 进入Arduino目录 `/path/arduino`

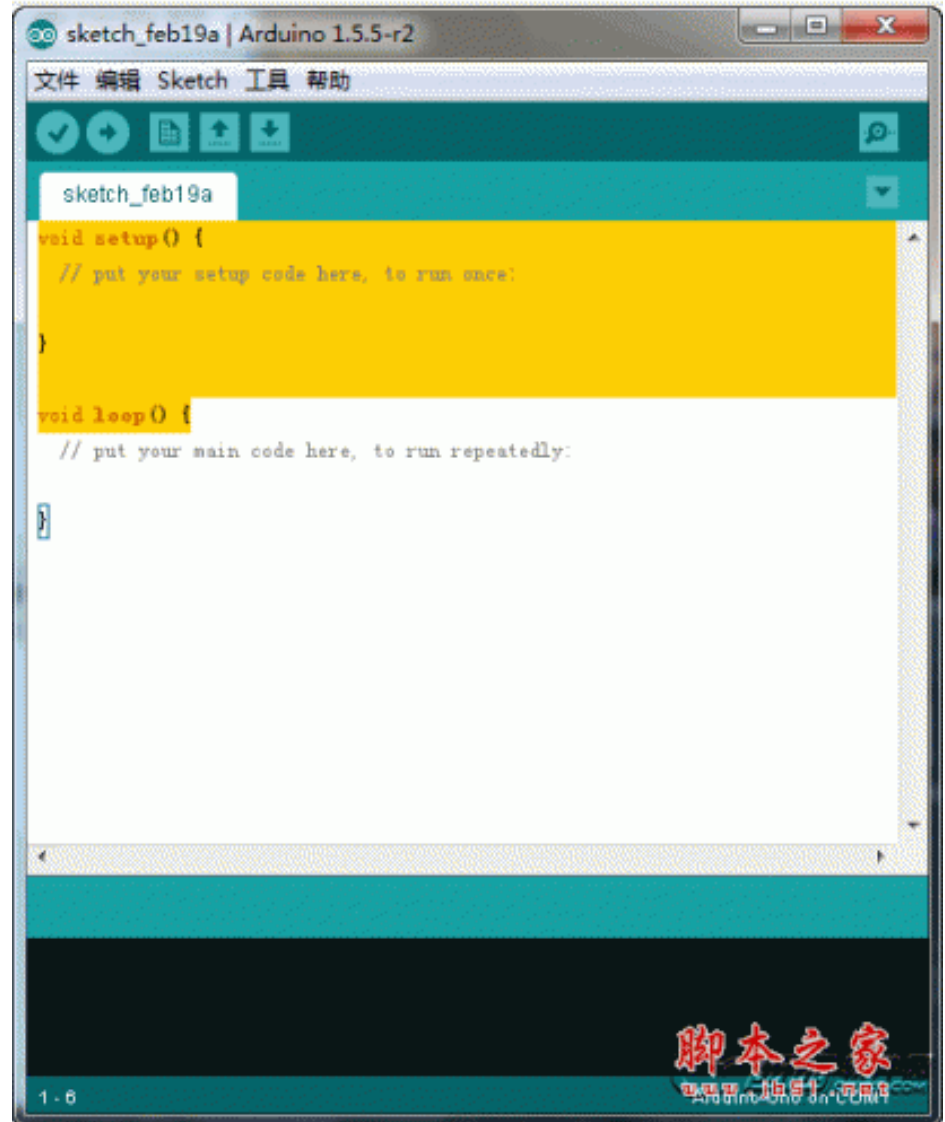
- `git pull`

- `git update`



## 6、 Windows系统下的arduino IDE

□启动IDE，得到如图的界面



## 6、 Windows系统下的arduino IDE

□ 另一种模式是在Windows下直接装Arduino

□ 1、 下载Windows版IDE

IDE <http://arduino.googlecode.com/files/arduino-1.0.1->

windows.zip（可以直接找最新中文版，但有的下载不带驱动）

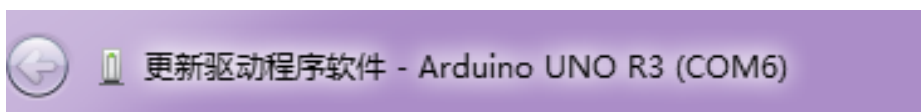
□ **Windows:** XP、WIN7 32位需要手工更新驱动，驱动在IDE目录下drivers\FTDI USB Drivers目录中，指定目录为之后自动搜索即可。WIN7 64位，WIN8自动在线更新驱动即可。

□ 将买回来的arduino拆了包装，使用方头usb数据线连接到电脑，本文以win7 32位为例，会出现识别到新硬件显示。

□ 然后，给出下载的驱动目录即可。

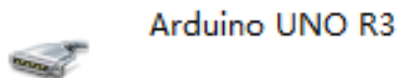
## 6、 Windows系统下的arduino IDE

- 如果没有自动识别新硬件，则双击“计算机|属性|设备管理器|其他设备|USB Serial Port”，选择“更新驱动程序软件”。
- 在弹出的窗口中，给出下载的驱动目录即可。



Windows 已经成功地更新驱动程序文件

Windows 已经完成安装此设备的驱动程序软件:





## 6、Windows系统下的arduino IDE

- 进入下载软件解压后的目录。
- 运行arduino.exe文件，打开IDE。

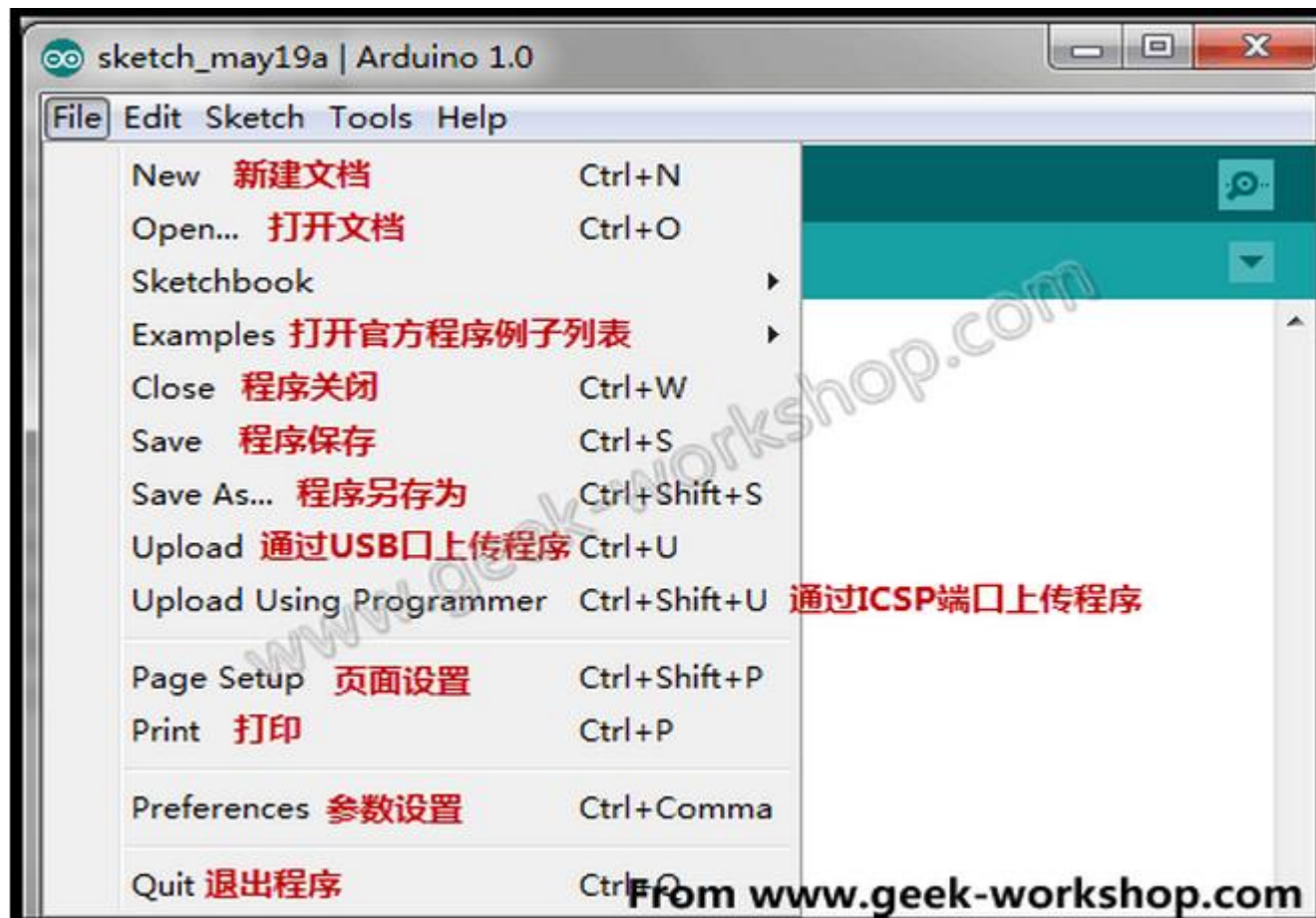


□IDE的几个快捷图标工具是：

- 编译
- 上传
- 新建程序
- 打开程序
- 保存程序
- 串口监视器

## 6、Windows系统下的arduino IDE

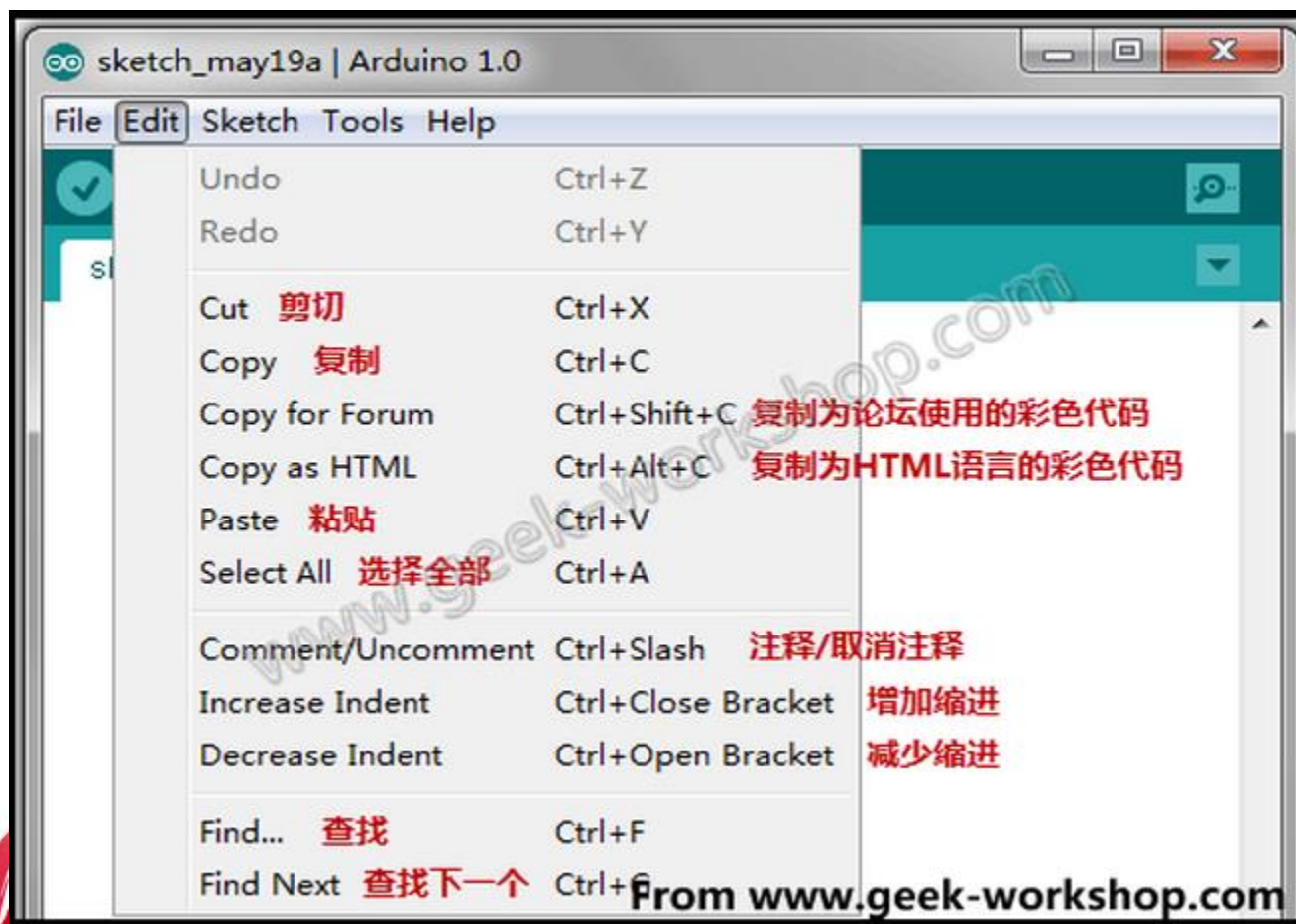
□菜单功能说明如下——文件：



□在IDE上编译完成的程序，就是通过Upload上传到arduino uno上的。

## 6、 Windows系统下的arduino IDE

□菜单功能说明如下——编辑：



## 6、Windows系统下的arduino IDE

□ 菜单功能说明如下——Sketch（草稿）：

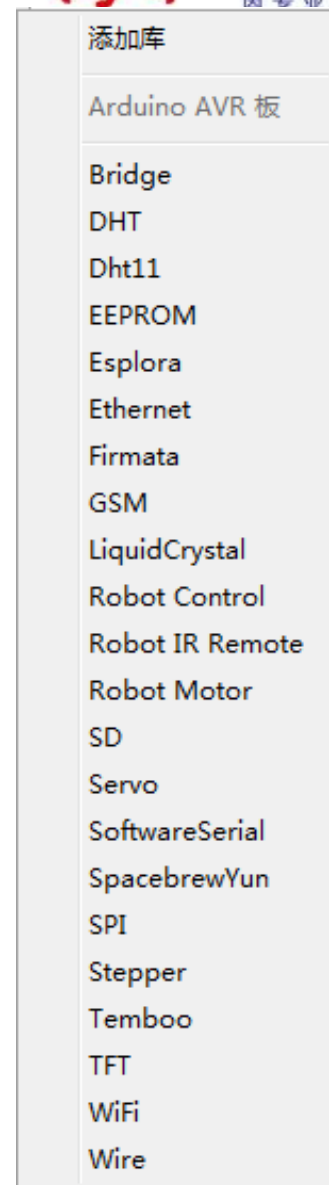


# 6、 Windows系统下的arduino IDE

□菜单功能——Improt Library（导入库）：

□最基本的导入库：

- EEPROM - EEPROM 读写程序库
- Ethernet - 以太网控制器程序库
- LiquidCrystal - LCD 控制程序库
- Servo - 舵机控制程序库
- SoftwareSerial - 任何数字 IO 口模拟串口程序库
- Stepper - 步进电机控制程序库
- Wire - TWI/I2C 总线程序库
- Matrix - LED 矩阵控制程序库
- Sprite - LED 矩阵图象处理控制程序库





## 6、Windows系统下的arduino IDE

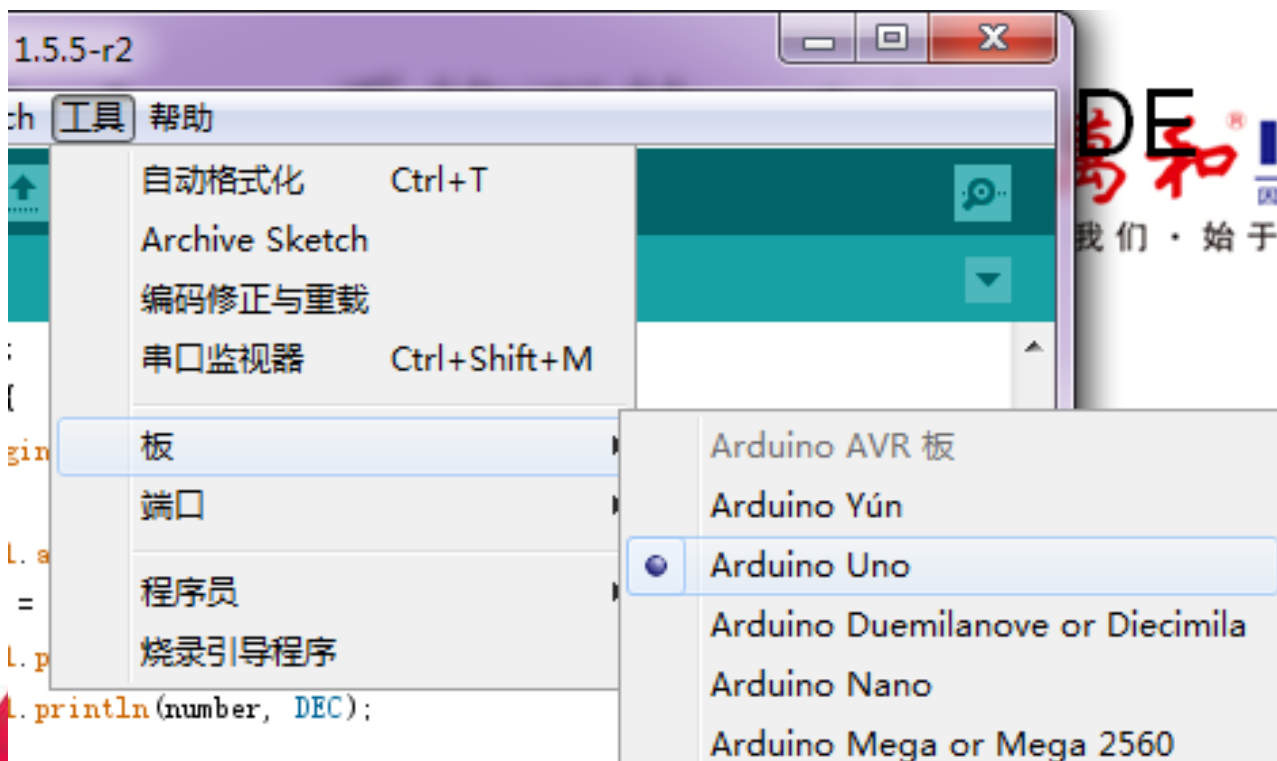
□菜单功能说明如下——Tools（工具）：



From [www.geek-workshop.com](http://www.geek-workshop.com)

## 6、Windows系统下的arduino IDE

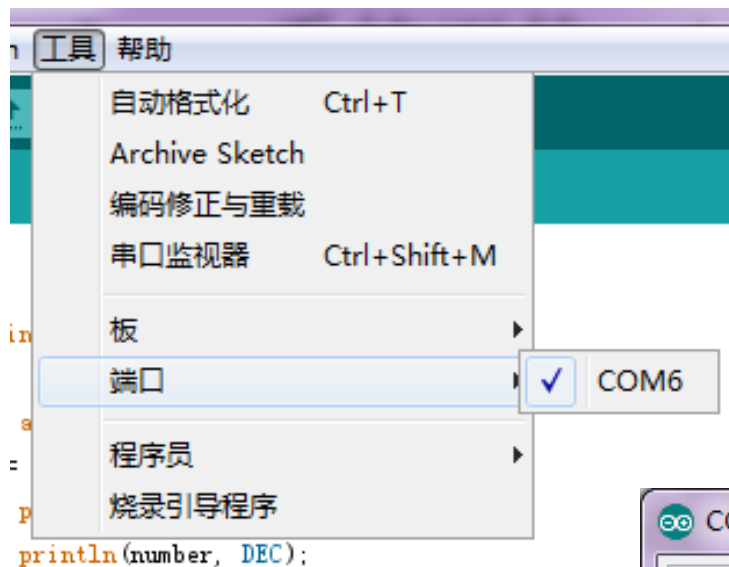
- 工具——选择开发版：
- 点击“工具|板”右键，选择与所使用的arduino一致的板子——我们用的是arduino uno



## 6、Windows系统下的arduino IDE

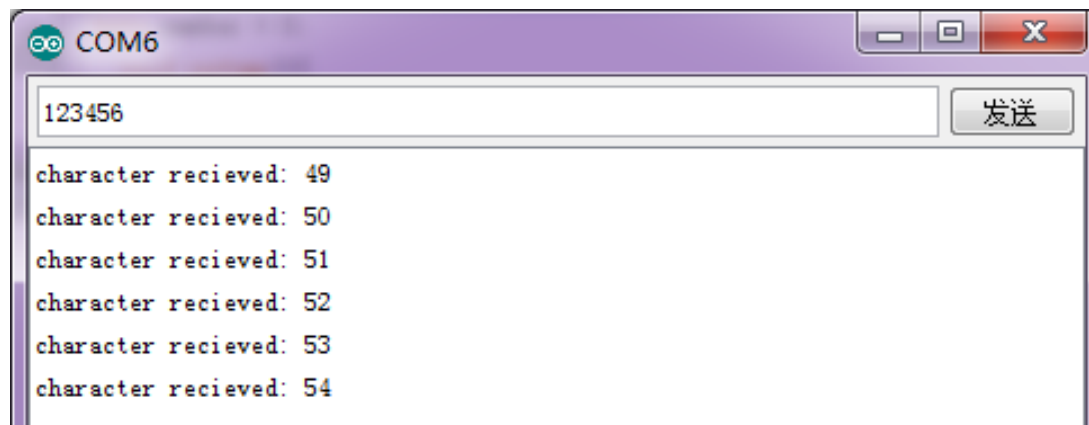
□工具——选择端口：

□点击“工具|端口”右键，看到我们用的是COM6口



□点击最右边的窗口监视器，在输入栏输入123456，得到回应如下图（以在arduino中下载了处理程序，后面介绍）

□说明arduino uno 工作正常。



## 6、 Windows系统下的arduino IDE

□国外某大侠开发出来VS2008 2010使用的[arduino插件](#)，不仅可以使使用VS进行代码的编写（不需要arduino IDE了），而且也可以直接在VS中，下载编译好的代码到Arduino上，非常之方便，教程如下：

□<http://www.geek-workshop.com/forum.php?mod=viewthread&tid=321>

□内有插件下载地址。

□使用方法与Arduino IDE基本一样，包括对各种arduino板子、端口的选择等，是否还可以进行串口监控？

## 7、RPI下的arduino IDE

❑第1步：下载Linux版的Arduino IDE（目前系统已自带）

❑在终端输入以下命令：

wget <http://arduino.googlecode.com/files/arduino-1.0-linux64.tgz>（如果这个地址无法下载，可以直接找合适的地址下载后，传到树莓派上。只要是linux版本就可以）

❑第2步：解压下载的安装包

❑在终端输入以下命令：

tar zxvf arduino-1.0-linux64.tgz

❑第3步：安装avr-gcc

❑在终端输入以下命令：

sudo apt-get install avr-libc

❑第4步：安装FTDI设备驱动

❑在终端输入以下命令：

sudo apt-get install libftdi1 安装FTDI设备驱动



## 7、 RPi下的arduino IDE

❑第5步：安装avrdude

❑在终端输入以下命令：

```
sudo apt-get install avrdude
```

❑第6步：安装java

❑在终端输入以下命令：

```
sudo apt-get install openjdk-7-jre
```

❑第7步：安装java的串口支持

❑在终端输入以下命令：

```
sudo apt-get install librx-tx-java
```

安装串口支持

## 7、 RPi下的arduino IDE

❑第8步：替换一些文件

❑在终端依次输入以下命令：

```
cd arduino-1.0
```

```
for i in $(find . -name "librxtxSerial.so") ;
```

```
do cp /usr/lib/jni/librxtxSerial.so $i ; done
```

```
for i in $(find . -name "RXTXcomm.jar") ;
```

```
do cp /usr/share/java/RXTXcomm.jar $i ; done
```

```
cp /usr/bin/avrdude /home/pi/arduino-
```

```
1.0/hardware/tools/avrdude
```

```
cp /etc/avrdude.conf /home/pi/arduino-
```

```
1.0/hardware/tools/avrdude.conf
```

注意：/home/pi是你安装arduino的目录

## 7、 RPi下的arduino IDE

- ❑ 第9步：加载arduino的USB调试接口
- ❑ `sudo insmod drivers/usb/class/cdc-acm`
- ❑ `sudo insmod drivers/usb/serial/usbserial`
- ❑ `sudo insmod drivers/usb/serial/ftdi_sio`
- ❑ 第10步：通过usb连接（/dev/ttyACMO）你的arduino设备，做你想做的吧。

## 7、RPI下的arduino IDE

□这样Arduino IDE在你的树莓派上就安装完成了

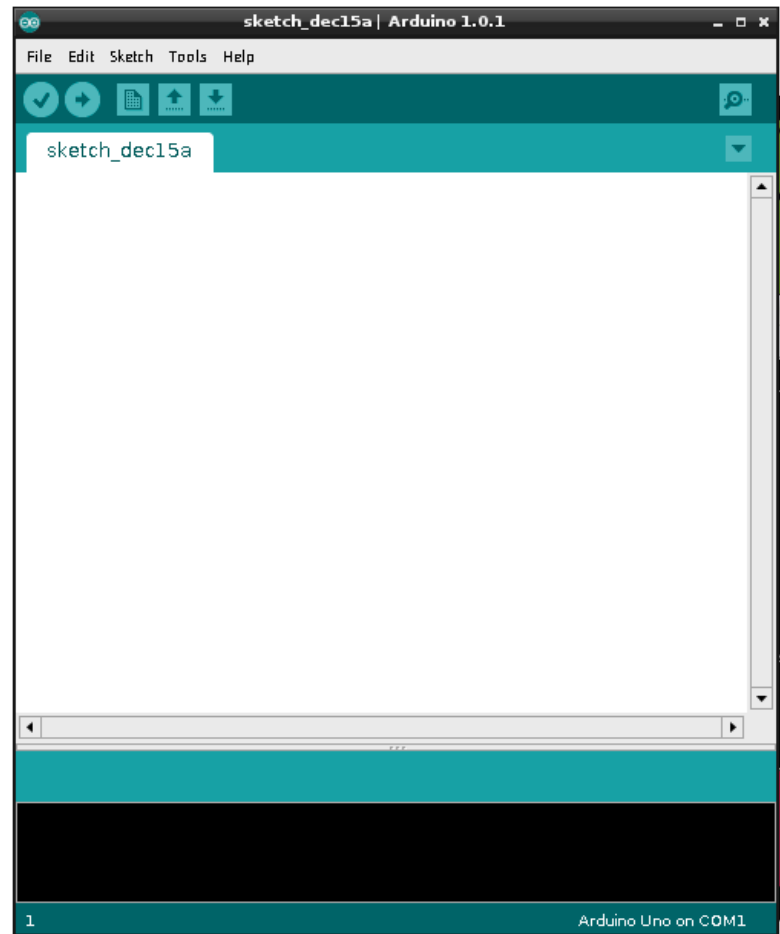
□运行方法如下：

□在终端依次输入以下命令：

```
cd arduino-1.0/
```

```
./arduino
```

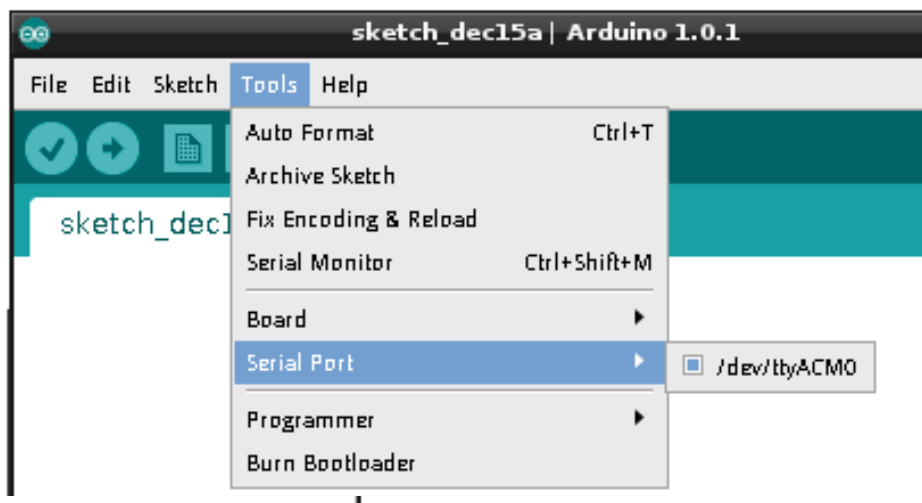
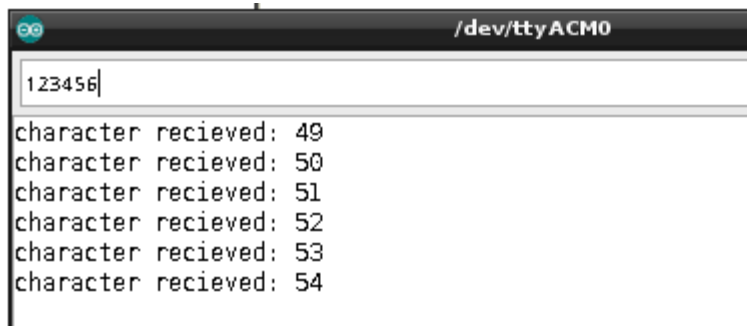
□此时，你就会发现  
Arduino IDE已经成  
功运行，与在  
Windows下的界面是  
完全一样的。



## 7、RPI下的arduino IDE

- ❑ 在arduino IDE菜单上选择Tools|Board, 选择Arduino Uno
- ❑ 与Windows下的通信端口不同, 在树莓派上, 我们使用的是Tools|SerialPort勾选/dev/ttyACM0

❑ 点击窗口监视器, 输入123456, 同样可以得到来自arduino uno的回应





## 8、 arduino 连接树莓派

□树莓派连接Arduino Uno的方式有很多种，主要有三种：

□USB方式

□GPIO方式

□I2C方式

□其他

## 9、 arduino 连接PC的驱动

□ Arduino与PC（Windows）连接

□ 下载Arduino与PC连接的USB驱动地址:

PL2303\_Prolific\_DriverInstaller\_v1.8.0.zip:

[http://www.prolific.com.tw/US/ShowProduct.aspx?p\\_id=225  
&pcid=41](http://www.prolific.com.tw/US/ShowProduct.aspx?p_id=225&pcid=41)

PL2303\_Prolific\_DriverInstaller\_v1.5.0.zip:

[ftp://ftp.omega.com/public/DASGroup/products/OM-PL-  
USBS/PL2303\\_Prolific\\_DriverInstaller\\_v1.5.0.zip](ftp://ftp.omega.com/public/DASGroup/products/OM-PL-USBS/PL2303_Prolific_DriverInstaller_v1.5.0.zip)

XP与WIN7都没问题,在WIN8 64下有可能遇到驱动无法使用问题, win8亲测可用,注意安装的时候以管理员身份win7兼容性运行, 安装完成之后点击“更新驱动程序软件”才可用。

# 10、 arduino 树莓派库

□1.安装python（安装过的请跳到下一步）：

```
sudo aptitude install python-dev
```

□2.安装python的GPIO模块,用于控制LED、电机等（安装过的请跳到下一步）

下载GPIO库

```
wget http://raspberrypi-gpio-  
python.googlecode.com/files/RPi.GPIO-0.3.1a.tar.gz
```

tar解压

```
tar xvzf RPi.GPIO-0.3.1a.tar.gz
```

进入解压后的文件夹

```
cd RPi.GPIO-0.3.1a
```

安装GPIO库文件

```
sudo python setup.py install
```

# 10、arduino 树莓派库

❑3.安装serial，用于串口通信及USB通信：

`sudo apt-get install python-serial`

❑4.当然你要是想在树莓派装串口调试工具就装：

`sudo apt-get install minicom`

❑配置minicom：

`sudo minicom -s`

❑命令minicom是进入串口超级终端画面，而minicom -s为配置minicom。

❑/dev/ttyAMA0 对应为串口0，  
为你连接开发板的端口

```
+-----[configuration]-----+
| Filenames and paths          |
| File transfer protocols      |
| Serial port setup            |
| Modem and dialing            |
| Screen and keyboard         |
| Save setup as dfl            |
| Save setup as..             |
| Exit                         |
| Exit from Minicom           |
+-----+-----+-----+-----+
```

# 11、配置树莓派串口

- 启动出现配置菜单：选serial port setup，进入串口配置：
  - 输入A：配置串口驱动为/dev/ttyAMA0
  - 输入E：配置速率为9600 8N1
  - 输入F：将Hardware Flow Control设为：NO
  - 回车：退出

```
+-----+
| A -   Serial Device       : /dev/ttyAMA0
| B - Lockfile Location    : /var/lock
| C -   Callin Program     :
| D -   Callout Program    :
| E -   Bps/Par/Bits       : 9600 8N1
| F - Hardware Flow Control : No
| G - Software Flow Control : No
|
| Change which setting? █
+-----+
```

# 11、配置树莓派串口

□ 由于我们使用minicom作为超级终端控制路由器等设备，而不是控制modem，所以需要修改Modem and dialing，将Init string, Reset string, Hang-up string设置为空。

```
+-----[Modem and dialing parameter setup]-----+
A - Init string .....
B - Reset string .....
C - Dialing prefix #1... ATDT
D - Dialing suffix #1... ^M
E - Dialing prefix #2... ATDP
F - Dialing suffix #2... ^M
G - Dialing prefix #3... ATX1DT
H - Dialing suffix #3... ;X4D^M
I - Connect string ..... CONNECT
J - No connect strings .. NO CARRIER          BUSY
                        NO DIALTONE             VOICE
K - Hang-up string .....
L - Dial cancel string .. ^M

M - Dial time ..... 45      Q - Auto bps detect ..... No
N - Delay before redial . 2  R - Modem has DCD line .. Yes
O - Number of tries ..... 10 S - Status line shows ... DTE speed
P - DTR drop time (0=no). 1  T - Multi-line untag .... No

Change which setting? █ Return or Esc to exit. Edit A+B to get defaults.
```



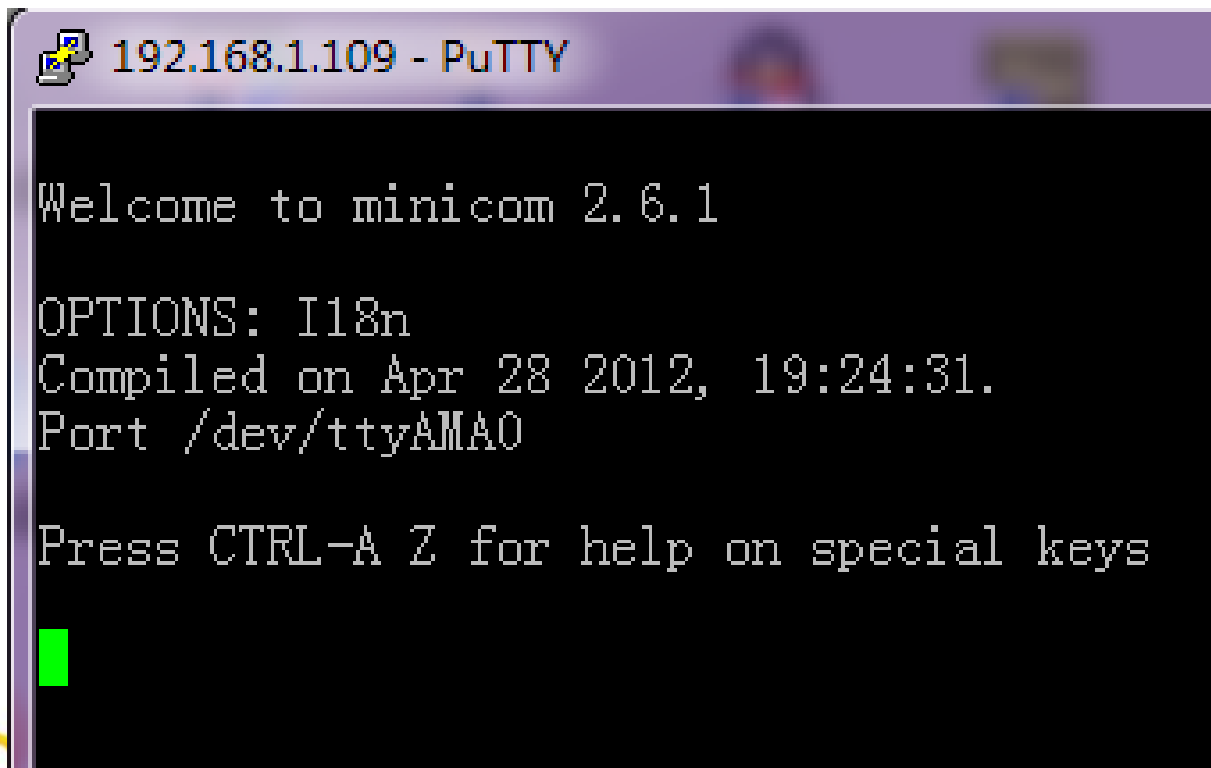
# 11、配置树莓派串口

- 设置完成后选择**Save setup as df1**将当前设置保存为默认设置。
- 在配置菜单，选**Save setup as df1**保存（一定要记得这一步）
- 选**Exit**退出。

```
+-----[configura Configuration saved
|
| Filenames and
| File transfer +-----+
| Serial port setup
| Modem and dialing
| Screen and keyboard
| Save setup as df1
| Save setup as..
| Exit
| Exit from Minicom
+-----+
```

# 11、配置树莓派串口

□下次再输入minicom（没有参数-s），即可直接进入串口调试工具。



```
192.168.1.109 - PuTTY

Welcome to minicom 2.6.1

OPTIONS: I18n
Compiled on Apr 28 2012, 19:24:31.
Port /dev/ttyAMA0

Press CTRL-A Z for help on special keys

█
```

# 11、配置树莓派串口

□测试环境设置是否都OK，在树莓派上：

```
sudo nano test.py
```

□或用xwindow下用python IDE打开一个空的文件

输入一下内容：

```
import serial
```

```
import RPi.GPIO
```

□保存退出，然后运行代码：

```
python test.py
```

□如果没有报错那就说明**RPi.GPIO**与**serial**两个库安装成功

## 12、USB对接方式

□1.用网线将Arduion Uno与树莓派的USB互连上，然后在树莓派输入：`ls /dev/tty*`

□查看有没有**ttyACM0** 这个文件

□只有在两个硬件USB互连的情况下才会有这个设备。如果两者没有连接是不会有的) 最新的系统一般都会自动生成。看到**ttyACM0**就说明二者可以通讯了，接下来运行测试代码



## 12、USB对接方式

□命令

ls /dev/tty\*

□结果:

```
root@raspberrypi:~# ls /dev/tty*
/dev/tty    /dev/tty19  /dev/tty3   /dev/tty40  /dev/tty51  /dev/tty62
/dev/tty0    /dev/tty2   /dev/tty30  /dev/tty41  /dev/tty52  /dev/tty63
/dev/tty1    /dev/tty20  /dev/tty31  /dev/tty42  /dev/tty53  /dev/tty7
/dev/tty10   /dev/tty21  /dev/tty32  /dev/tty43  /dev/tty54  /dev/tty8
/dev/tty11   /dev/tty22  /dev/tty33  /dev/tty44  /dev/tty55  /dev/tty9
/dev/tty12   /dev/tty23  /dev/tty34  /dev/tty45  /dev/tty56  /dev/ttyACM0
/dev/tty13   /dev/tty24  /dev/tty35  /dev/tty46  /dev/tty57  /dev/ttyAMA0
/dev/tty14   /dev/tty25  /dev/tty36  /dev/tty47  /dev/tty58  /dev/ttyprintk
/dev/tty15   /dev/tty26  /dev/tty37  /dev/tty48  /dev/tty59
/dev/tty16   /dev/tty27  /dev/tty38  /dev/tty49  /dev/tty6
/dev/tty17   /dev/tty28  /dev/tty39  /dev/tty5   /dev/tty60
/dev/tty18   /dev/tty29  /dev/tty4   /dev/tty50  /dev/tty61
root@raspberrypi:~#
```

## 12、Arduino代码

□先把下面的代码上传到**Arduino**上，然后再把两者**USB**互接：

```
byte number = 0;
void setup(){
  Serial.begin(9600); //设置串口通讯速率为9600/s
}
void loop(){
  if (Serial.available()) {
    number = Serial.read(); //读取串口（来自树莓派）
    Serial.print("character recieved: "); //送回串口接收到的字符
    Serial.println(number, DEC);
  }
}
```

□在**IDE**上编译完成上述程序后，通过**Upload**上传到**arduino uno**上。



# 13、树莓派python代码

```
import serial
ser = serial.Serial('/dev/ttyACM0', 9600, timeout=1) //设置串口
ser.open() //打开串口
ser.write("testing") //写串口
try:
    while 1:
        response = ser.readline() //读串口
        print response //显示内容
except KeyboardInterrupt: //接收键盘中断
    ser.close() //关闭串口
```

□保存xxx.py 退出

# 14、运行

- 注:树莓派下的**serial**与**GPIO**库都要在**root**帐户下才能运行
- 然后运行树莓派代码 `python xxx.py`（这里的**xxx**就是你保存的树莓派代码）看到：

```
character recieved: 116 -- t  
character recieved: 101 -- e  
character recieved: 115 -- s
```

...

```
character recieved: 103 -- g
```

- 说明两者USB通讯成功了，上述收到的代码是什么意思？
  - 就是**xxx.py**的一行代码**ser.write("testing")**，
  - 以及**Arduino**的两行显示代码：

```
Serial.print("character recieved: ");  
Serial.println(number, DEC);
```

- 还可以在**IDE**的串口监视器输入一些内容，看看能否接收到。

# 15、总结

□从上述程序例子中，可以大致了解树莓派与arduino之间是如何通过串口方式（实际为USB转）进行交互的。

树莓派程序：

```
ser.open() //打开串口
ser.write("testing") //写串口
try:
    while 1:
        response = ser.readline() //读串口
        print response //显示内容
except KeyboardInterrupt: //接收中断
    ser.close() //关闭串口
```

□在本例中：

- 树莓派有发送和接收
- Arduino有接收和发送
- 构成“交互”
- 串口读写、实现很简单

Arduino程序：

```
void loop(){
    if (Serial.available()) {
        number = Serial.read(); //读取串口（来自树莓派）
        Serial.print("character recieved: "); //送回接收到的
        字符
        Serial.println(number, DEC);
```

# 16、Arduino代码

□Arduino 语言是以setup()开头，loop()作为主体的一个程序构架。

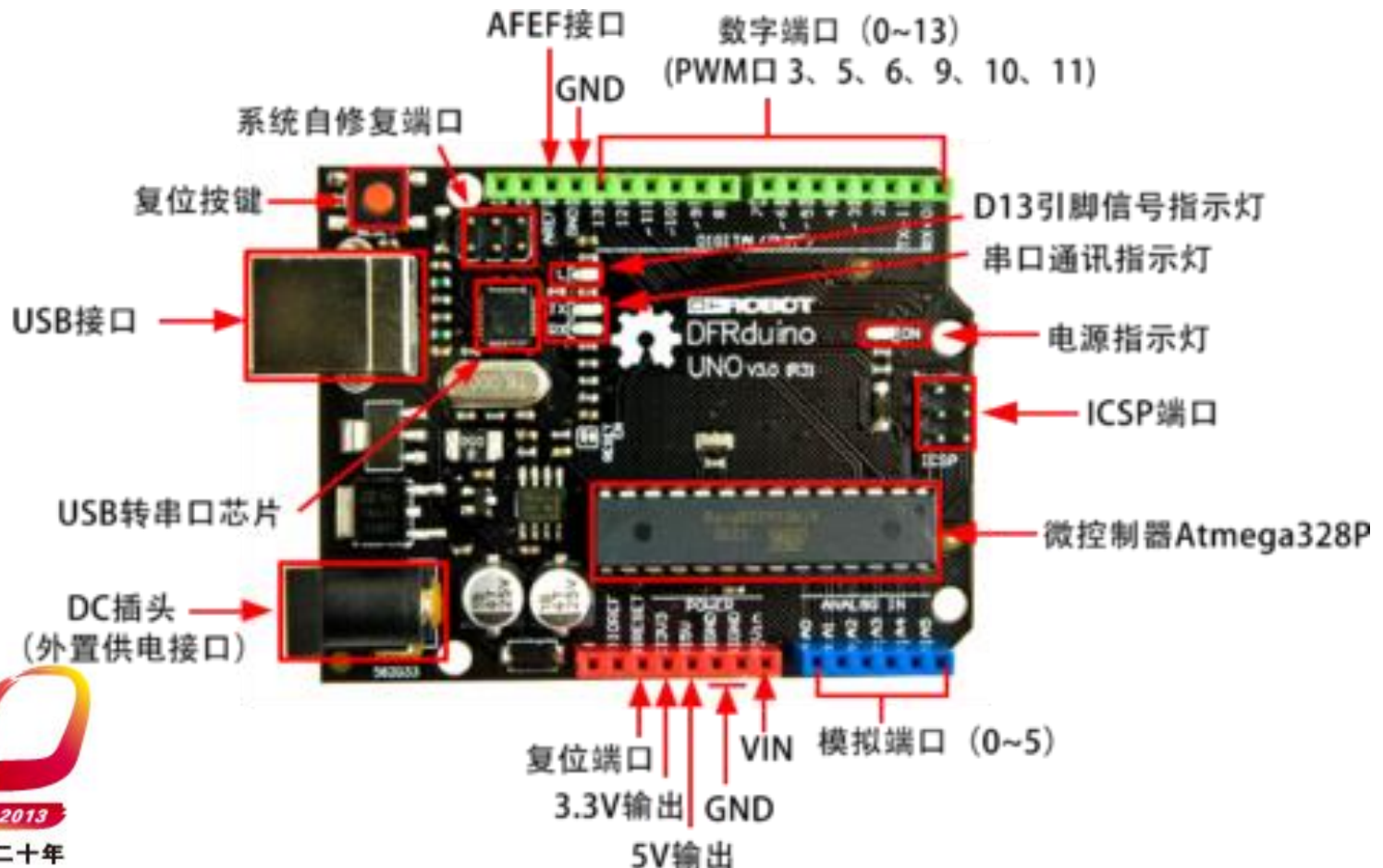
□setup(): 用来初始化变量，管脚模式，调用库函数等等，此函数只运行一次。

□loop()函数是一个循环函数，函数内的语句周而复始的循环执行，功能类似c语言中的main()。

```
byte number = 0;
void setup(){
    Serial.begin(9600); //设置串口通讯速率为9600/s
}
void loop(){
    if (Serial.available()) {
        number = Serial.read(); //读取串口（来自树莓派）
        Serial.print("character recieved: "); //送回串口接收到的字符
        Serial.println(number, DEC);
    }
}
```

# 17、GPIO方式

□GPIO方式就是利用树莓派的pin8(GPIO14) pin10(GPIO15)与Arduion pin0(rx) pin1(tx)的进行对接



# 17、GPIO方式

□注：对接时，需要一个硬件逻辑电平转换器（这个淘宝上有，一个5元钱）。要实在没有就用两电阻也行（一个是3.3K，一个是1.6K）线路怎么接后面会讲，千万不要把两个直接用电线对接。两个电压（RPI是3.3V，Arduion是5V）不一样会造烧坏你的RPI。



转载请标明 树莓派论坛  
[www.shumeipai.net](http://www.shumeipai.net)



# 18、GPIO方式的代码

- Anduino 测试代码与上面一样不用做改动
- 树莓派python代码改动一下就是把 **ttyACM0** 改为 **ttyAMA0** 其他的不变。

```
import serial
ser = serial.Serial('/dev/ttyAMA0', 9600, timeout=1)
ser.open()
ser.write("testing")
try:
    while 1:
        response = ser.readline()
        print response
except KeyboardInterrupt:
    ser.close()
```

# 19、修改配置文件

## □ 1. 修改cmdline.txt

```
sudo nano /boot/cmdline.txt
```

## □ 看到

```
dwc_otg.lpm_enable=0  
console=ttyAMA0,115200  
kgdboc=ttyAMA0,115200  
console=tty1  
root=/dev/mmcblk0p2  
rootfstype=ext4  
elevator=deadline  
rootwait
```

## □ 把

```
console=ttyAMA0,115200  
kgdboc=ttyAMA0,115200
```

删除



## 20、修改配置文件

❑ 2. 修改inittab，然后禁止掉从串口登录的功能

❑ 输入

```
sudo nano /etc/inittab
```

❑ 找到

❑ T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100

❑ 在这一行前面加个#注释掉这一行

❑ 然后ctrl+x 按Y保存退出

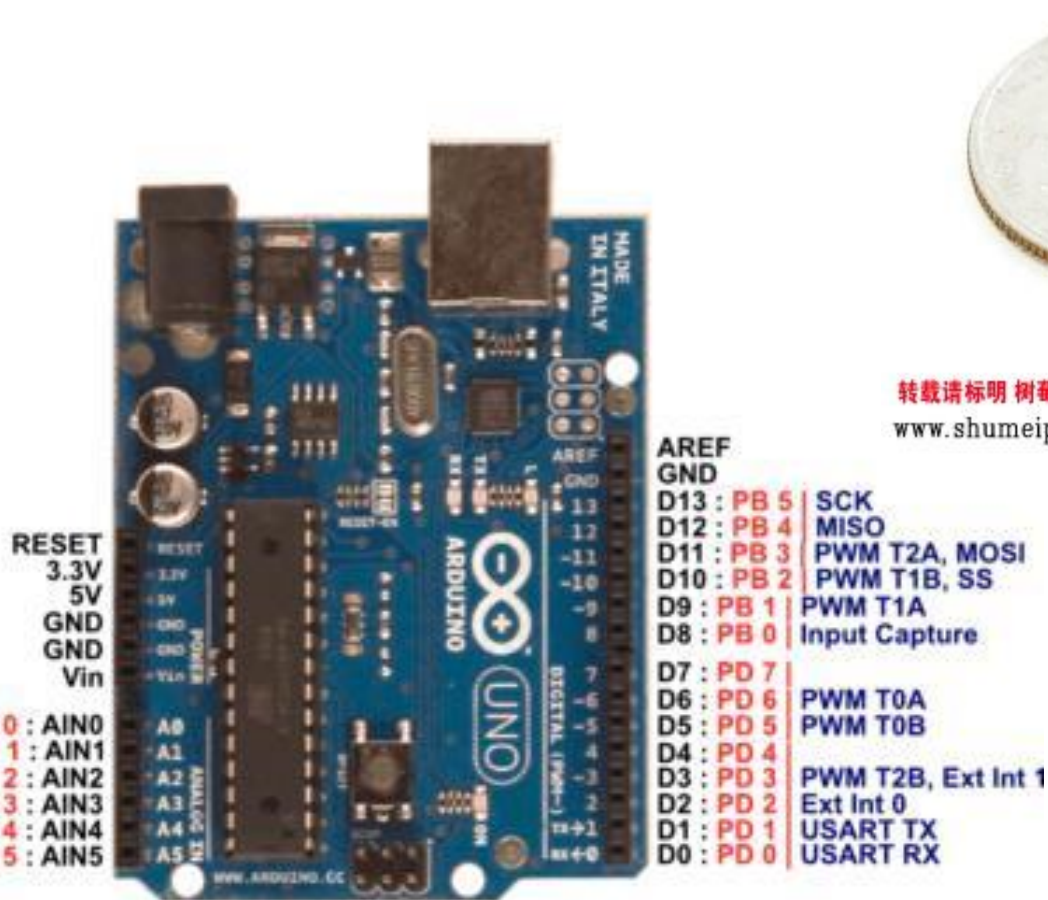
❑ 重启树莓派后，树莓派的pin8(GPIO14) pin10(GPIO15)这两个口就可以当成普通的串口来用了

❑ 现在你可以像使用一个普通串口那样使用/dev/ttyMA0了

# 21、硬件连接

□接下来做硬件的部份的对接

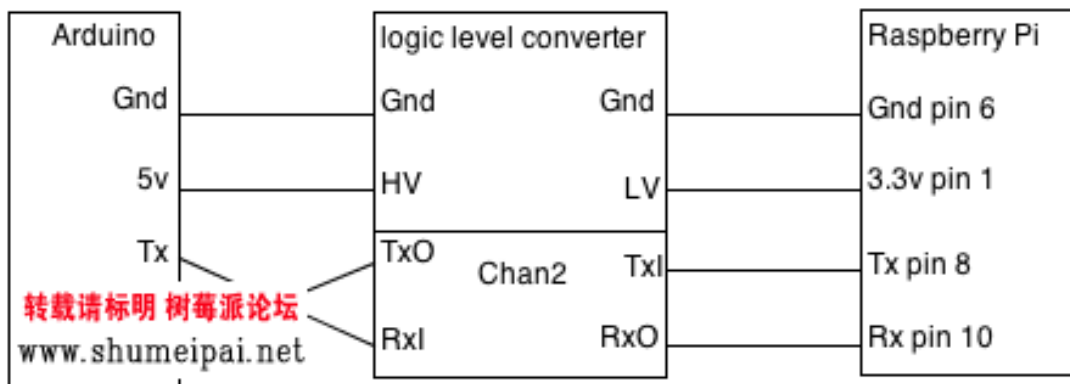
1.逻辑电平转换器方式, 下图是中间件



转载请标明 树莓派论坛  
[www.shumeipai.net](http://www.shumeipai.net)

# 21、硬件连接

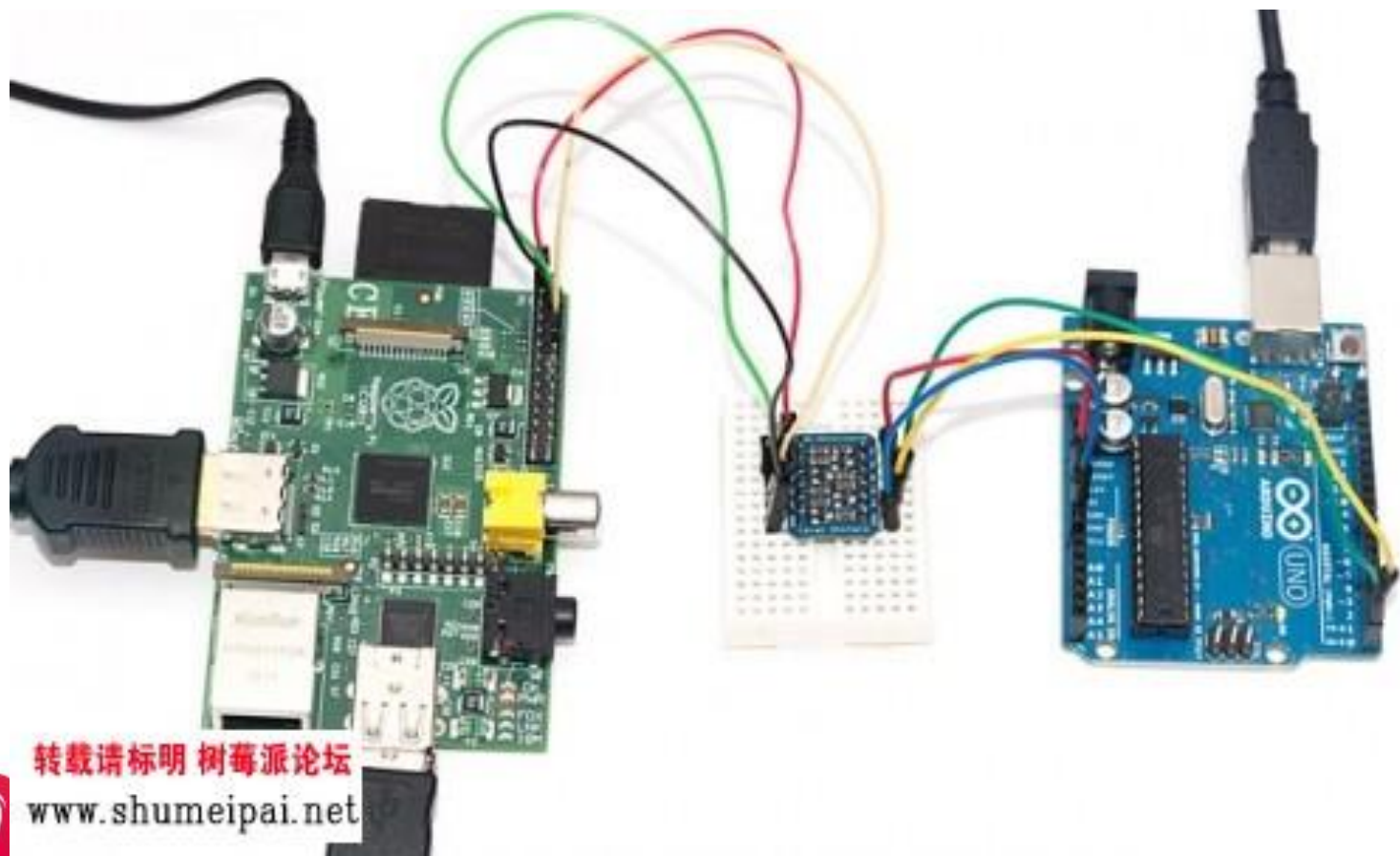
## □接线（逻辑图）





## 21、硬件连接

□接线（实际效果图）

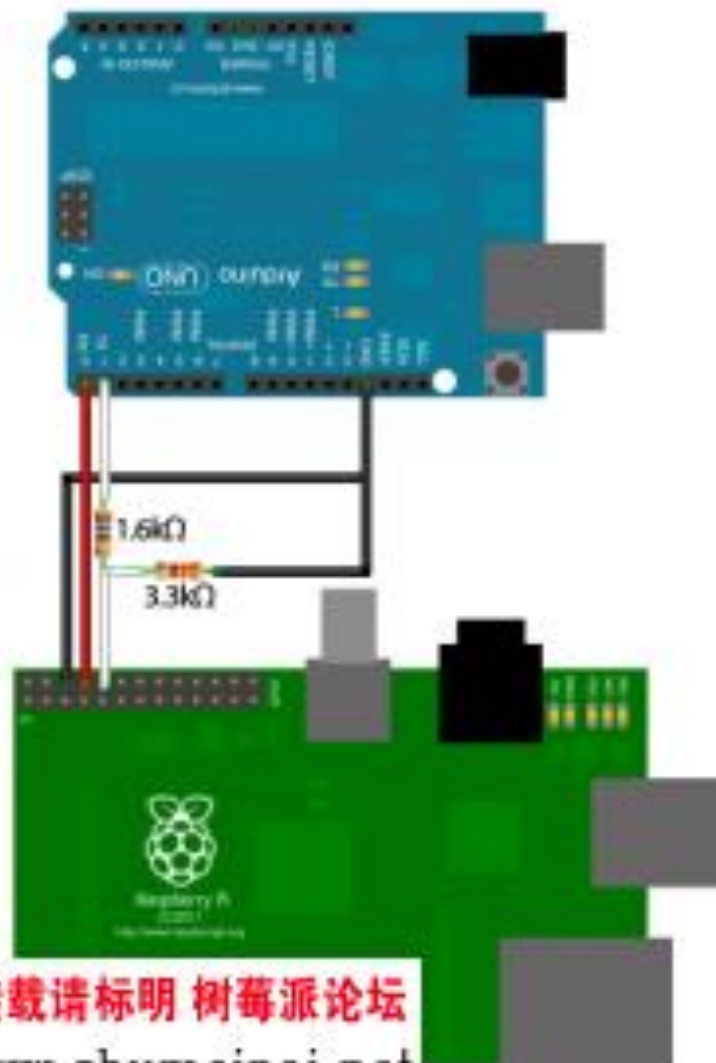


转载请标明 树莓派论坛  
[www.shumeipai.net](http://www.shumeipai.net)



## 21、硬件连接

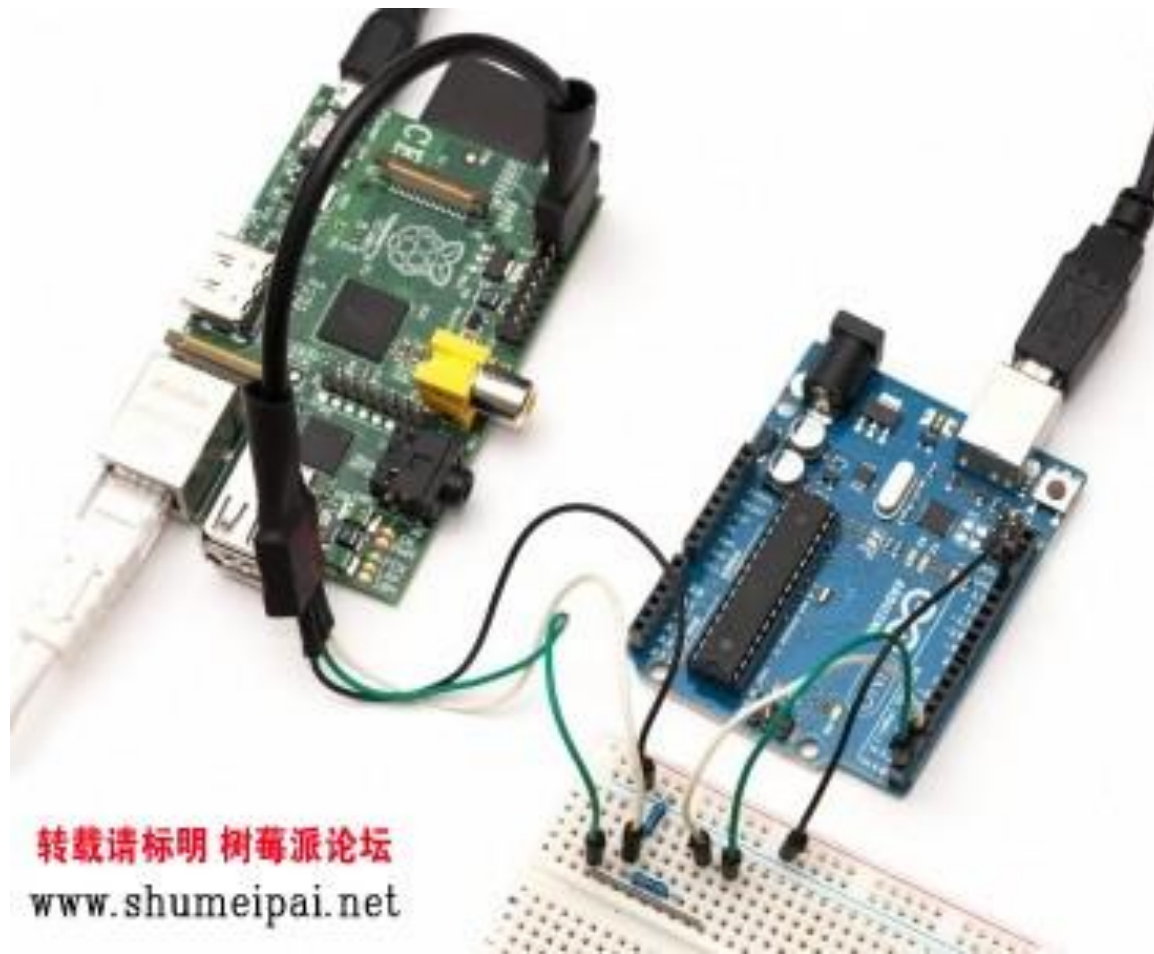
□接线（用电阻连接）



转载请标明 树莓派论坛  
[www.shumeipai.net](http://www.shumeipai.net)

## 21、硬件连接

### □接线（用电阻连接）



转载请标明 树莓派论坛  
[www.shumeipai.net](http://www.shumeipai.net)