# Arduino 使用Usart-GPU串口液晶屏

原Usart-GPU的说明书，请参见：

http://pan.baidu.com/share/link?shareid=3358573337&uk=3204894695

这里只讲和Arduino有关的：

## 前言：

Usart-GPU串口液晶屏自从推出后，由于低廉的价格，迅速得到广大单片机爱好者的青睐，STM32,STC,甚至古老的51系列都没问题；但是最近越来越多用户问Arduino如何使用串口屏，本想Arduino对串口编程非常方便，应该非常简单就可以驱动串口屏，但是实际并非如此，很多用户都卡在这里，于是立即TB下单了几块Arduino开发板，开始研究......

发现主要难点：

1、唯一的串口被开发环境占用；

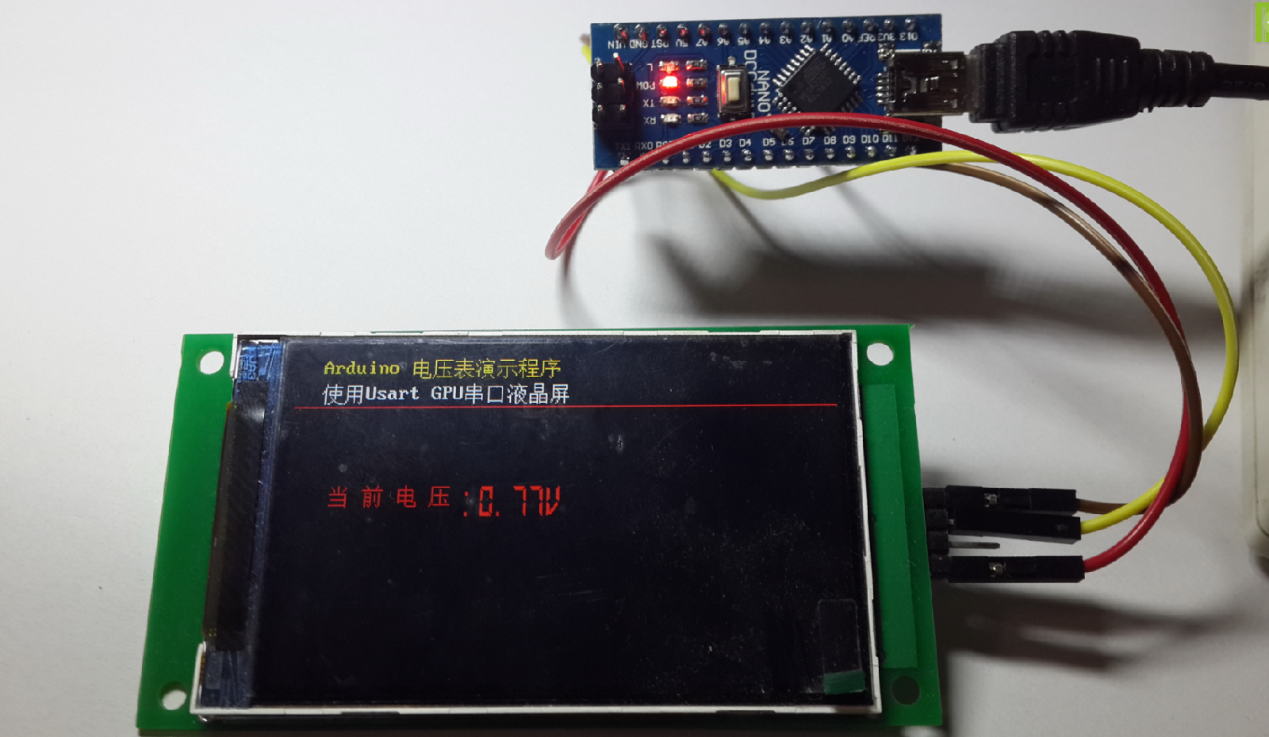
2、IDE开发环境无法输入中文，使用剪贴板贴入，中文也是UTF8，非串口屏要求的GB2312内码

3、如何输出浮点数的到串口屏上

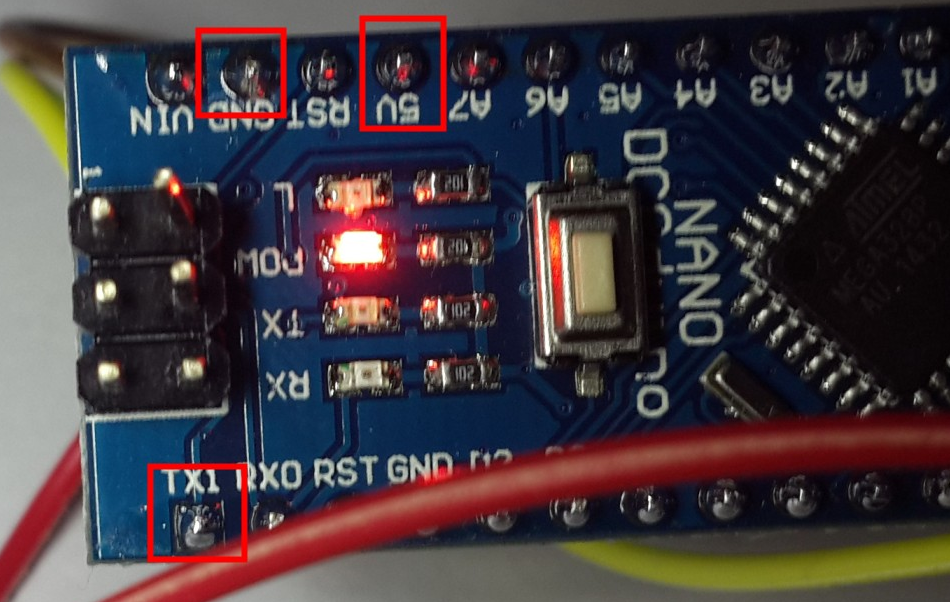
## 硬件接法：

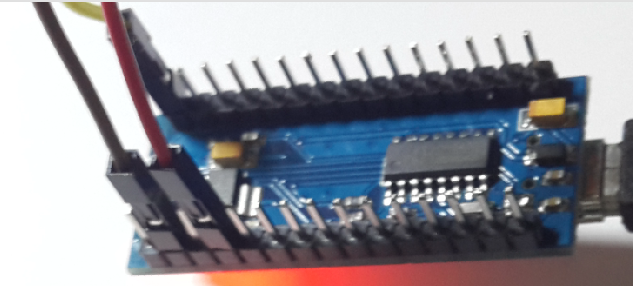
Arduino 就一个串口，而且用于和PC上的IDE环境相通信，编译的文件通过串口上传到MCU上，就是说该串口已经被占用了，那如何接串口屏呢？

经过分析，PC一段主要使用PC(TX)-MCU(RX)这条上行通道；而下行MCU(TX)-PC(RX)这条PC只是接受而已，而串口液晶屏在下行通道的情况下与PC形式一样，因此在接法上出现了前所未有的**一根信号线驱动一块液晶屏**的现象!



从图上可以看到，单片机板和串口屏只接了3根线，分别是+5V线，GND线，和TX线；



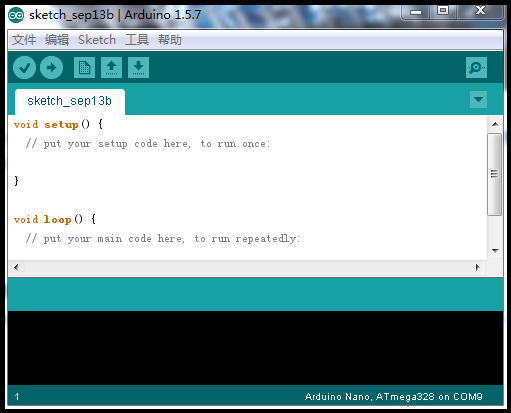


注意：MCU(TX)-串口屏的TX还是RX需要视串口屏的型号而定，由于串口接反只是不起作用而已，因此当不起作用时尝试一下接另外一根线；

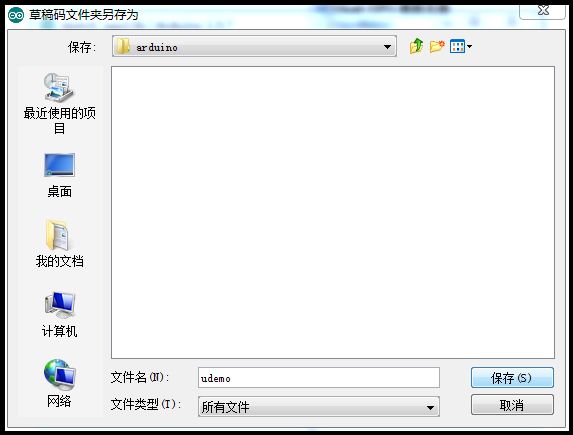
## 制作一个电压表示例：

###### 第一步：建立一个新程序：

打开Arduino的开发环境（关于板子的选择和串口的选择，略），选择：文件->新建



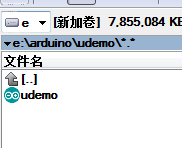
在特定的目录建立一个存放源程序的目录，我建立的是E:\arduino 的目录，此时选择：文件->另存为：



选择E:\arduino\ 并把文件名起名为udemo

此步的目的是为下一步建立中文字符串文件做准备；

保存之后：



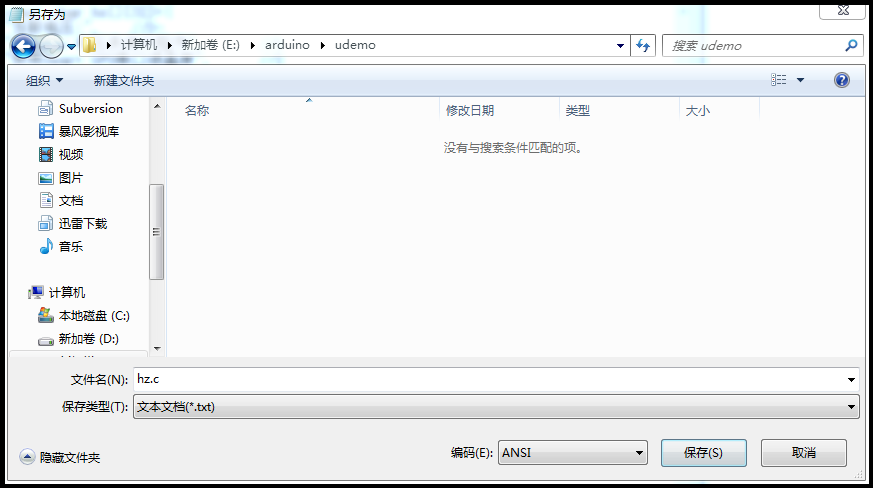
目录就建立好了，

###### 第二步：建立中文字符串文件

使用记事本（或其他类似文本编辑器），填写如下C代码：

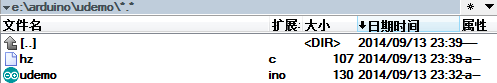
|  |
| --- |
| const char hz[][32]={  "当前电压:", //0-  "Arduino 电压表演示程序", //1  "使用Usart GPU串口液晶屏", //2  }; |

使用记事本的“文件”->“另存为：”



选择E:\arduino\udemo\ 目录，并将文件起名为hz.c,注意编码一定要选择“ANSI”，千万不要选择UTF8之类；

点击保存，文件存好；



备注：

const char hz[][32] 实际上是声明了一个字符串数组，每个字符串不能超过32字节（GB2312编码的汉字16个),如果你要显示的汉字特别长，请修改32这个数字；

此种方法实际上是将汉字全部提出到一个数组中，应用此类方式很容易实现国际版程序,即切换数组就可以实现不同语种的显示；

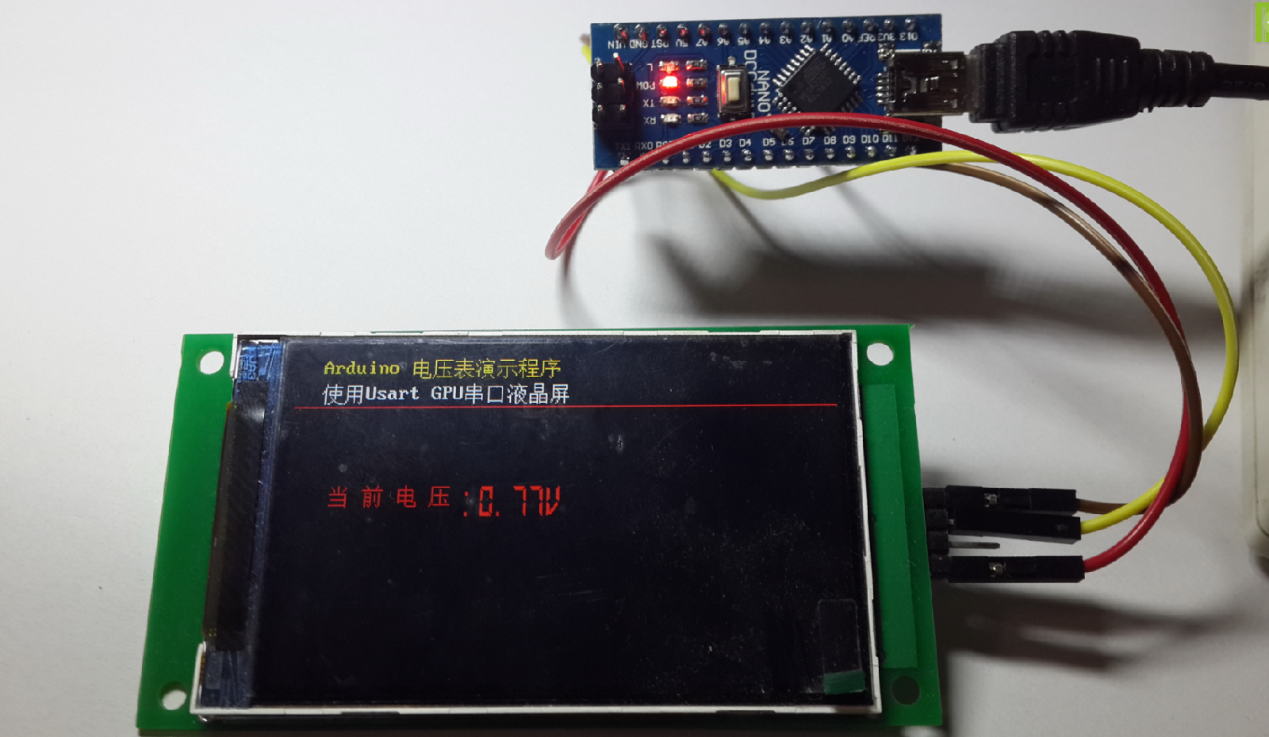
**【注意】：建立这个外挂文件的目的就是为了解决使用IDE编程环境无法编辑GB2312内码汉字的问题，请勿使用IDE编程环境打开编辑此文件，这样操作会将此文件改写为UTF8内码的文件！**

###### 第三步：编辑主程序

在开发环境中输入以下代码：

|  |
| --- |
| #include"e:\arduino\Udemo\hz.c"  void setup() {  Serial.begin(115200);  while (!Serial) {  // wait for serial line to be ready  }  Serial.print("CLS(0);");  Serial.print("DS16(20,2,'");Serial.print(hz[1]);Serial.print("',4);");  Serial.print("DS16(20,22,'");Serial.print(hz[2]);Serial.print("',15);");  Serial.println("PL(0,40,399,40,1);");Serial.flush();delay(200);  }  // the loop routine runs over and over again forever:  void loop() {  // read the input on analog pin 0:  int sensorValue = analogRead(A0);  float vol=sensorValue\*2.5/1024;  Serial.print("DS24(30,100,'");Serial.print(hz[0]);Serial.print(vol,2); Serial.println("V',1);");  delay(150); // delay in between reads for stability  } |

点击“上传”，则程序编译，屏上立即显示程序规定的界面：



**代码说明：**

#include"e:\arduino\Udemo\hz.c" 这句是引用中文字符串文件，如何做到相对路径这个不会，因此只好使用绝对路径了；

Serial.print("DS16(20,2,'");Serial.print(hz[1]);Serial.print("',4);"); 这是输出一个DS16语句；

Serial.println("PL(0,40,399,40,1);");Serial.flush();delay(200); 注意，print 和println的差异就在于println后面会自动跟0d 0a，这和串口屏要求一致，因此最后一条语句必须是println,且需要延时让串口屏显示完成后，才能下发下面的语句；因此需要delay(200); 而：Serial.flush();其实可省；

float vol=sensorValue\*2.5/1024; 这里做了一个非常简单的AD值到电压值的转换，纯属演示；

Serial.print("DS24(30,100,'");Serial.print(hz[0]);Serial.print(vol,2); Serial.println("V',1);");

输出电压值，其中：Serial.print(vol,2); 表示使用小数点后2位输出vol这个浮点数，详细参见arduino的开发文档；

配合串口屏的批页面以及图形，相信大家可以做出非常优秀的产品，请大家按本文档一试；

附：Usart-GPU固件升级网址：

http://stm32.sinaapp.com/gpu.html