Arduino Usart-GPU serial LCD screen

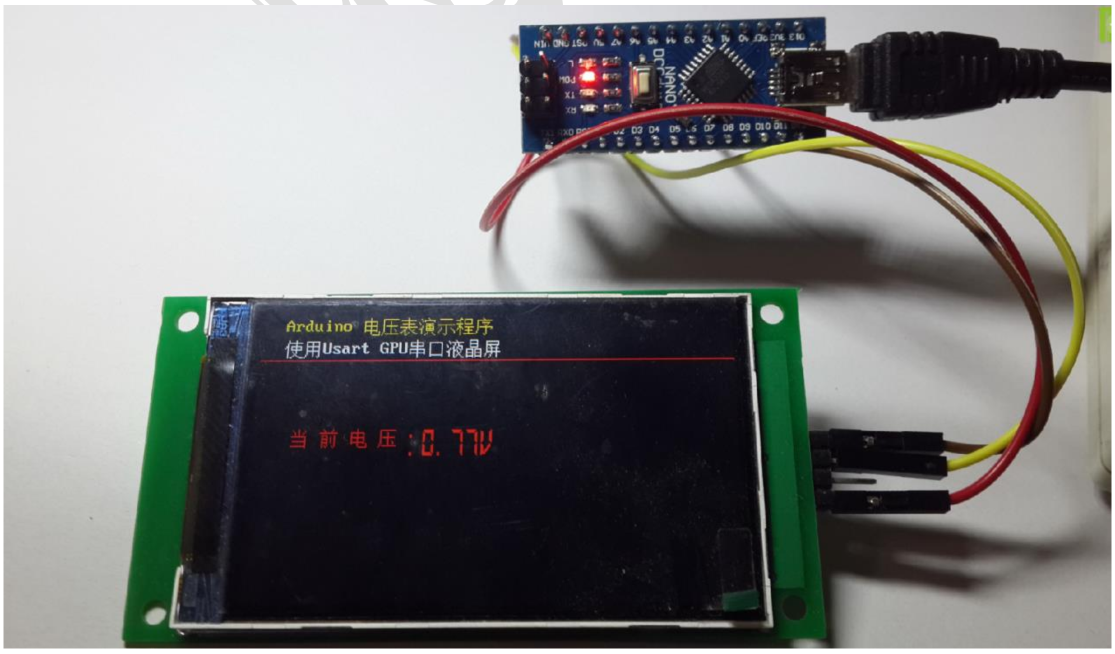
For the original Usart-GPU instructions, see:
Http://pan.baidu.com/share/link?shareid=3358573337&uk=3204894695
Here only talk about Arduino:

Preface:

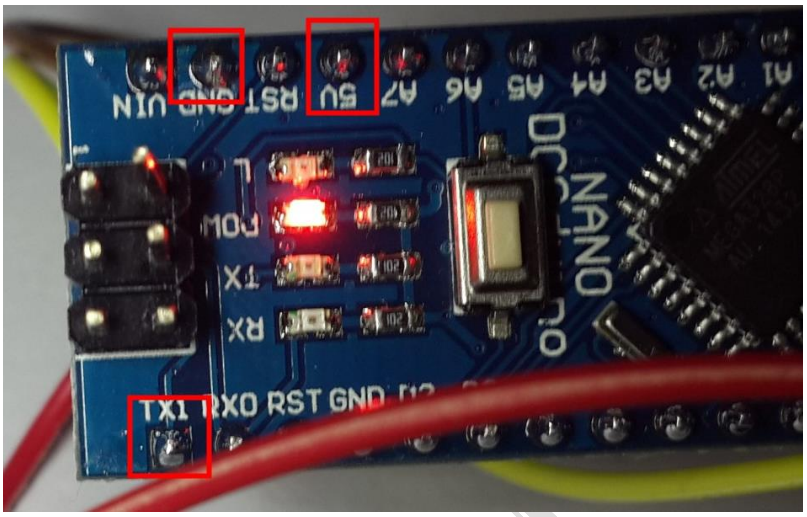
Usart-GPU serial LCD screen since its launch, due to low price, quickly get the majority of MCU enthusiasts of all ages, STM32, STC, and even the old 51 series no problem; but more and more users recently asked Arduino how to use the serial screen , Arduino wanted to serial programming is very convenient, it should be very simple to drive serial port screen, but the actual is not the case, many users are stuck here, so immediately TB a few Arduino development board, began to study ..... .
Found the main difficulty:
1, the only serial port is occupied by the development environment;
2, IDE development environment can not input Chinese, paste the use of the clipboard, the Chinese is UTF8, non-serial screen requirements GB2312 code
3, how to output floating-point to the serial screen

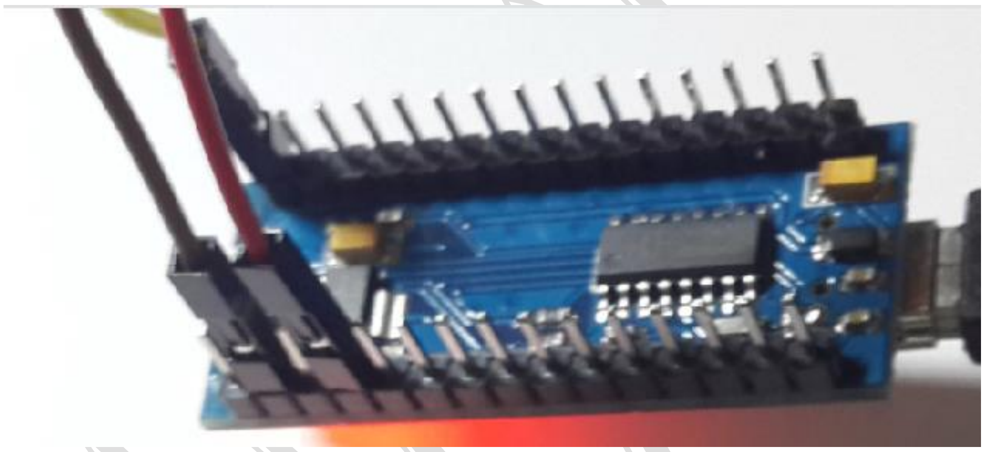
Hardware connection:

Arduino on a serial port, but also for the PC and the IDE environment to communicate, compile the file upload to the MCU through the serial port, that is, the serial port has been occupied, then how to take the serial screen?
PC (TX) -MCU (RX) This uplink channel; and downlink MCU (TX) -PC (RX) This PC is only accepted, and the serial LCD screen in the case of the downlink channel with the PC form, so the connection appeared on an unprecedented signal line to drive a liquid crystal screen phenomenon!



From the figure you can see, single-chip board and serial port screen only received three lines,namely,5Vline,GNDline,andTXline;





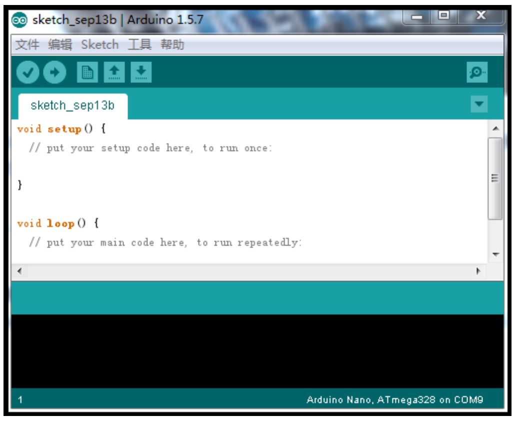
Note: MCU (TX) - Serial port TX or RX need to see the serial port of the screen model, the

serial port is reversed just does not work, so when it does not work, try to take another line;

Example of making a voltmeter:

Step 1: Create a new program:

Open Arduino development environment (on the board of choice and the choice of serial), select: File -> New


sketch_sep13b | Arduino 1.5.7
文件 编辑 Sketch 工具 帮助
sketch_sep13b

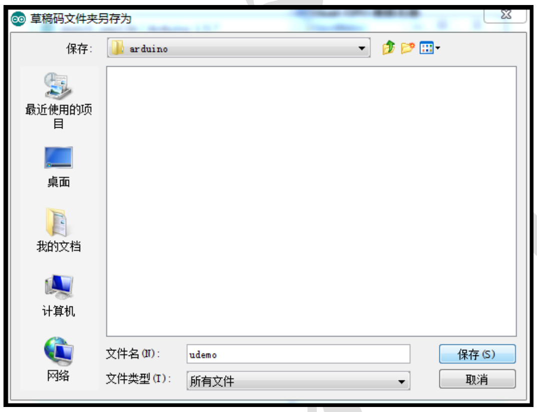
```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:
```

1
Arduino Nano, ATmega328 on COM9

In a specific directory to create a directory to store the source code, I created the E: \ arduino directory, then select: File -> Save As:



保存: arduino

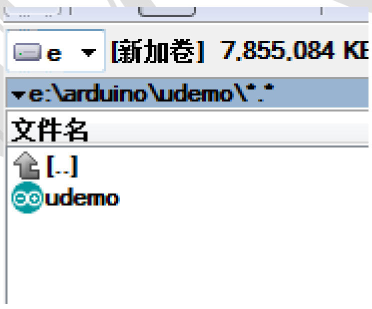
文件名 (N): udemo

文件类型 (T): 所有文件

Select E: \ arduino \ and name the file udemo

The purpose of this step is to prepare for the next step in the establishment of the Chinese string file;

After saving:



e:\arduino\udemo\*.*

[..]

udemo

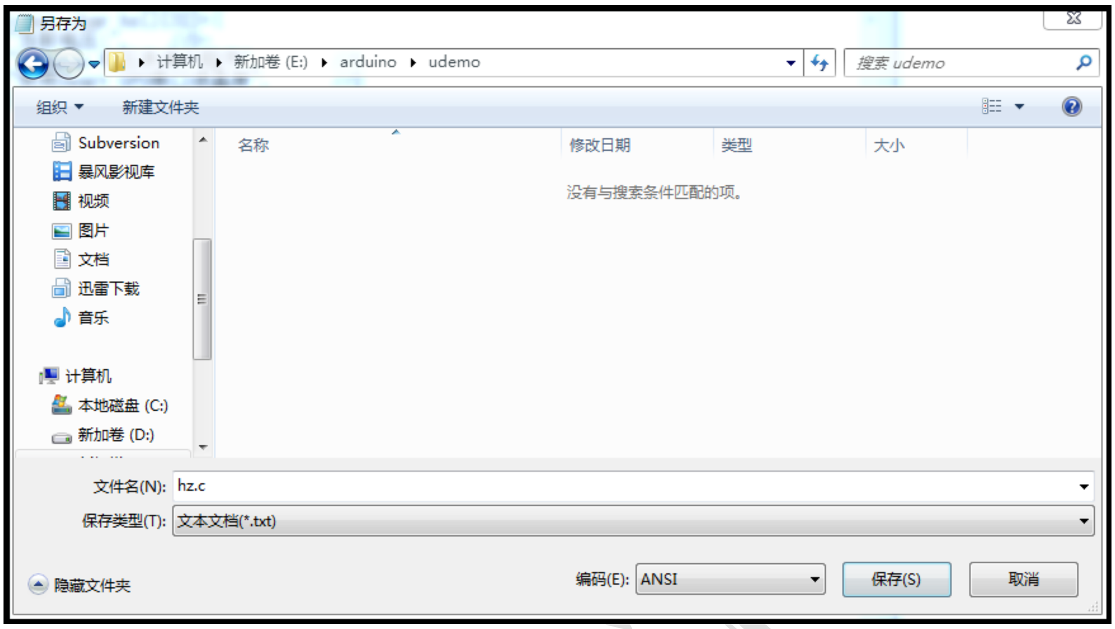
Directory on the establishment of good,

The second step: the establishment of the Chinese string file

Using Notepad (or other similar text editor), fill in the following C code:

```
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
};
```

Use Notepad's "File" -> "Save As:"



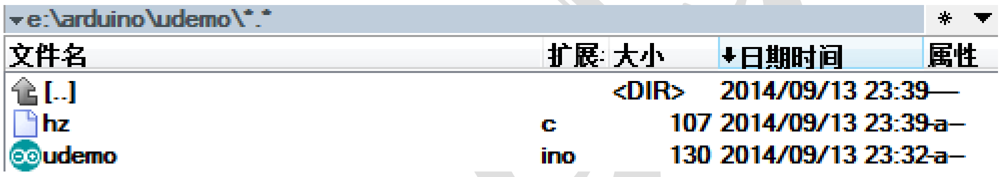
文件名(N): hz.c

保存类型(T): 文本文档(*.txt)

编码(E): ANSI

Select E: \ arduino \ demo \ directory, and the file named hz.c, pay attention to the code must choose "ANSI", do not choose UTF8 and the like;

Click Save, save the file;



```
e:\arduino\udemo\*.*
文件名                    扩展  大小    日期时间           属性
[..]                          <DIR>  2014/09/13 23:39 ——
hz                        c     107  2014/09/13 23:39 a–
udemo                     ino   130  2014/09/13 23:32 a–
```

Note:

Const char hz [] [32] is actually declared a string array, each string can not exceed 32 bytes (GB2312 encoded Chinese characters 16), if you want to display the characters particularly long, please modify the 32 number ;

This method is actually put all the Chinese characters into an array, the application of such methods is very easy to implement the international version of the program, that is, switching the array can achieve the display of different languages;

【Note】: The purpose of this plug-in file is to solve the use of IDE programming environment can not edit the GB2312 code within the Chinese characters, do not use the IDE programming environment to open edit this file, this operation will be rewritten to UTF8 code file .

Step 3: Edit the main program

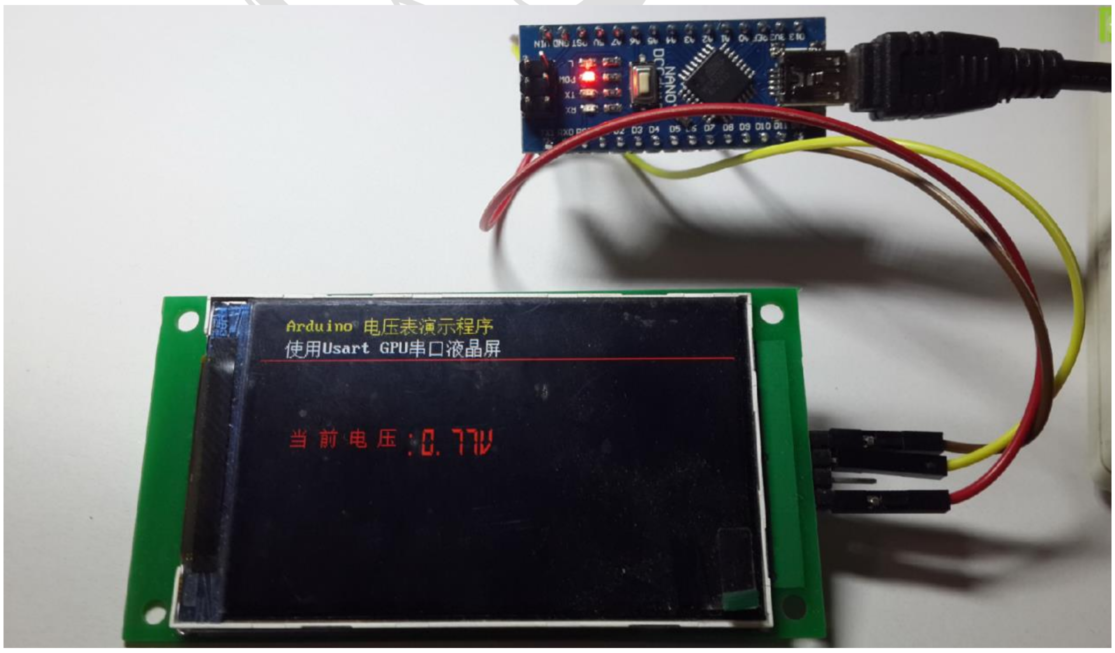
In the development environment, enter the following code:

```
#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
}
```

Click "Upload", the program compiles, the screen immediately shows the program provides the interface:



Code Description:

```
#include"e:\arduino\Udemo\hz.c"
```

This is a document quoted Chinese string, the relative path to how this will not, and therefore had to use the absolute path;

```
Serial.print("DS16(20,2,'");Serial.print(hz[1]);Serial.print("',4);");
```

This is the output of a DS16 statement;

```
Serial.println("PL(0,40,399,40,1);");Serial.flush();delay(200);
```

Note that the difference between print and println is that println will automatically follow 0d 0a, which is consistent with the serial port screen requirements, so the last statement must be println, and need to delay the serial port display is complete, can be issued the following statement; So need delay (200); and: Serial.flush (); In fact,

```
float vol=sensorValue*2.5/1024;
```

Here to do a very simple AD value to the voltage value conversion, purely a demonstration;

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

Output voltage value, of which: Serial.print (vol, 2); that the use of two decimal places after the vol output floating point, see arduino development documents;

With the serial screen batch screen and graphics, I believe we can make a very good product, please press this document a try;
Attached: Usart-GPU firmware upgrade URL:
Http://stm32.sinaapp.com/gpu.html