

## GPS 位置信息解析

### 1. 学习目标

本次课程我们主要学习使用 arduino 和 GPS 模块实现位置信息解析并打印功能。

### 2. 课前准备

GPS 模块采用的是 UART 和 USB 通讯，这里使用 arduino UNO 的 UART 口读取信息，将模块的 TX 连接 arduino UNO 板子的 D0 引脚。VCC 和 GND 分别连接 5V 和 GND。

### 3. 程序

初始化串口。

```
void setup() //初始化内容
{
    GpsSerial.begin(9600);    //定义波特率9600
    DebugSerial.begin(9600);

    Save_Data.isGetData = false;
    Save_Data.isParseData = false;
    Save_Data.isUsefull = false;
}
```

读取串口数据。

```
void gpsRead() {
    while (GpsSerial.available())
    {
        gpsRxBuffer[ii++] = GpsSerial.read();
        if (ii == gpsRxBufferLength) clrGpsRxBuffer();
    }

    char* GPS_BufferHead;
    char* GPS_BufferTail;
    if ((GPS_BufferHead = strstr(gpsRxBuffer, "$GPRMC,") != NULL || (GPS_BufferHead = strstr(gpsRxBuffer, "$GNRMC,") != NULL )
    {
        if (((GPS_BufferTail = strstr(GPS_BufferHead, "\r\n")) != NULL) && (GPS_BufferTail > GPS_BufferHead))
        {
            memcpy(Save_Data.GPS_Buffer, GPS_BufferHead, GPS_BufferTail - GPS_BufferHead);
            Save_Data.isGetData = true;

            clrGpsRxBuffer();
        }
    }
}
```

解析串口数据。

```

void parseGpsBuffer()
{
    char *subString;
    char *subStringNext;
    if (Save_Data.isGetData)
    {
        Save_Data.isGetData = false;
        DebugSerial.println("*****");
        DebugSerial.println(Save_Data.GPS_Buffer);

        for (int i = 0 ; i <= 6 ; i++)
        {
            if (i == 0)
            {
                if ((subString = strstr(Save_Data.GPS_Buffer, ",")) == NULL)
                    errorLog(1); //解析错误
            }
            else
            {
                subString++;
                if ((subStringNext = strstr(subString, ",")) != NULL)
                {
                    char usefullBuffer[2];
                    switch(i)
                    {
                        case 1:memcpy(Save_Data.UTCTime, subString, subStringNext - subString);break; //获取UTC时间
                        case 2:memcpy(usefullBuffer, subString, subStringNext - subString);break; //获取UTC时间
                        case 3:memcpy(Save_Data.latitude, subString, subStringNext - subString);break; //获取纬度信息
                        case 4:memcpy(Save_Data.N_S, subString, subStringNext - subString);break; //获取N/S
                        case 5:memcpy(Save_Data.longitude, subString, subStringNext - subString);break; //获取纬度信息
                        case 6:memcpy(Save_Data.E_W, subString, subStringNext - subString);break; //获取E/W

                        default:break;
                    }

                    subString = subStringNext;
                    Save_Data.isParseData = true;
                    if(usefullBuffer[0] == 'A')
                        Save_Data.isUsefull = true;
                    else if(usefullBuffer[0] == 'V')
                        Save_Data.isUsefull = false;
                }
            }
        }
    }
}

```

打印解析之后的位置信息。

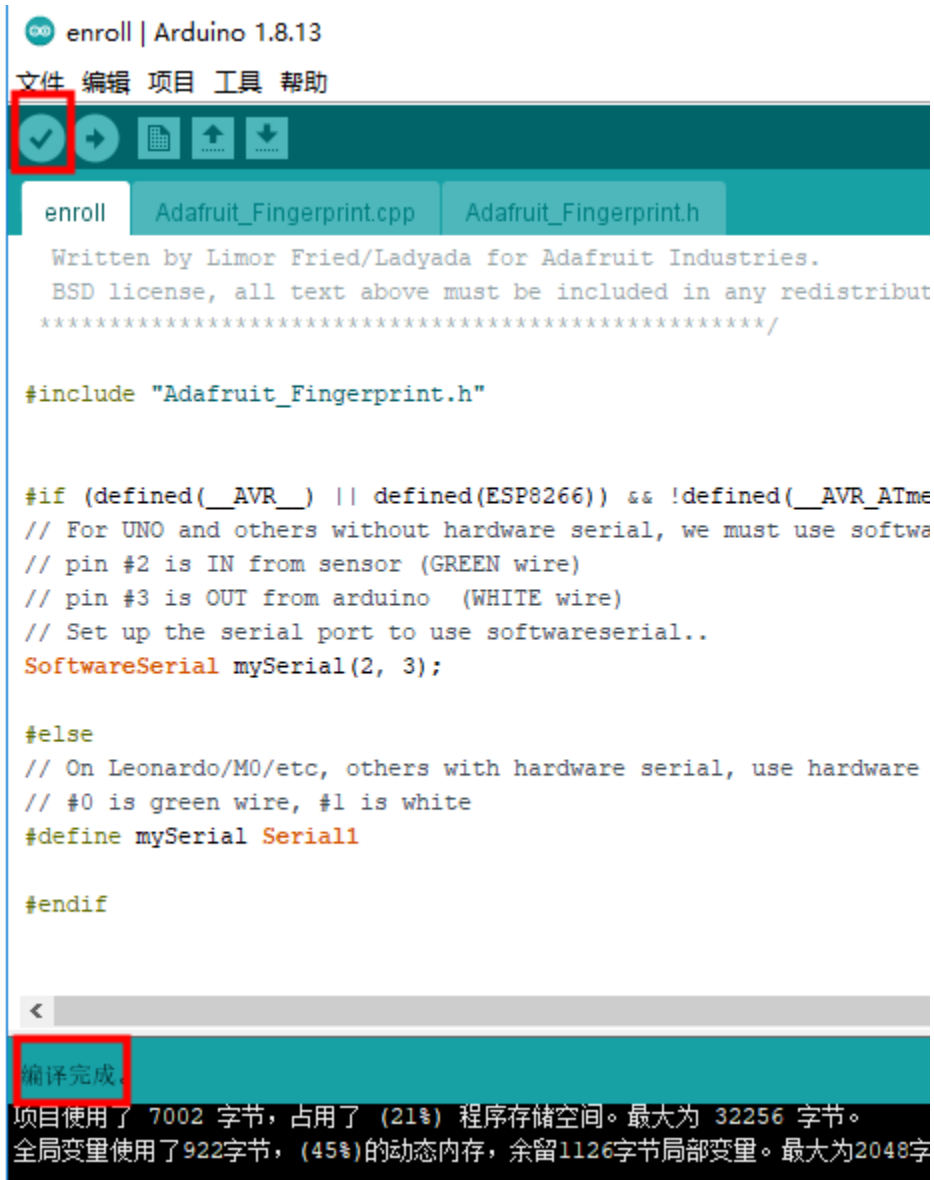
```
void printGpsBuffer()
{
  if (Save_Data.isParseData)
  {
    Save_Data.isParseData = false;

    DebugSerial.print("Save_Data.UTCTime = ");
    DebugSerial.println(Save_Data.UTCTime);

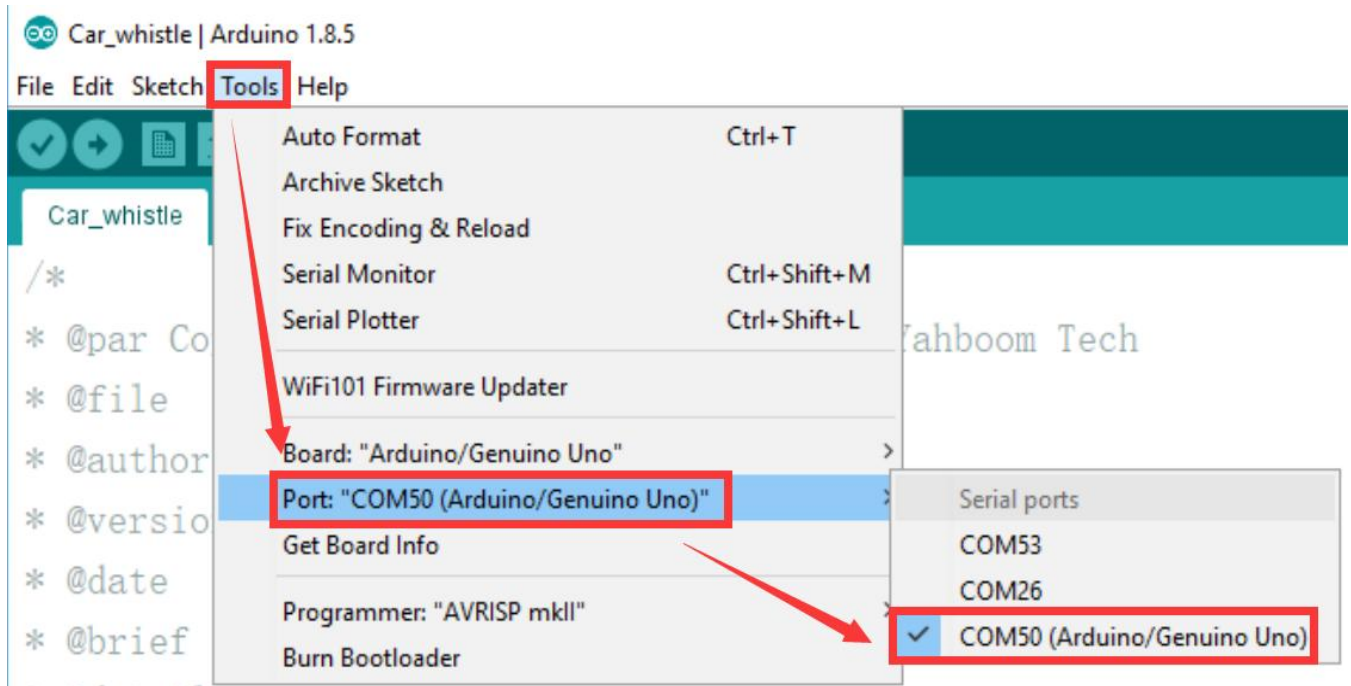
    if(Save_Data.isUsefull)
    {
      Save_Data.isUsefull = false;
      DebugSerial.print("Save_Data.latitude = ");
      DebugSerial.println(Save_Data.latitude);
      DebugSerial.print("Save_Data.N_S = ");
      DebugSerial.println(Save_Data.N_S);
      DebugSerial.print("Save_Data.longitude = ");
      DebugSerial.println(Save_Data.longitude);
      DebugSerial.print("Save_Data.E_W = ");
      DebugSerial.println(Save_Data.E_W);
    }
    else
    {
      DebugSerial.println("GPS DATA is not usefull!");
    }
  }
}
```

## 4. 编译下载程序

4.1 我们需要通用 Arduino IDE 软件打开文件，然后点击菜单栏中的“√”编译程序，并且等待左下角出现“编译成功”的字样。



4.2 在 Arduino IDE 的菜单栏中, 我们需要选择【工具】---【端口】---选择设备管理器中刚刚显示端口号, 如下图所示。



4.3 选择完成后，点击菜单栏下的“→”将代码上传到 UNO 板。当左下角出现“上传完成”字样时，表示程序已成功上传到 UNO 板，如下图所示。

```

文件 编辑 项目 工具 帮助

Written by Limor Fried/Ladyada for Adafruit Industries.
BSD license, all text above must be included in any redistributi
*****/

#include "Adafruit_Fingerprint.h"

#if (defined(__AVR__) || defined(ESP8266)) && !defined(__AVR_ATmeg
// For UNO and others without hardware serial, we must use softwar
// pin #2 is IN from sensor (GREEN wire)
// pin #3 is OUT from arduino (WHITE wire)
// Set up the serial port to use softwareserial..
SoftwareSerial mySerial(2, 3);

#else
// On Leonardo/M0/etc, others with hardware serial, use hardware s
// #0 is green wire, #1 is white
#define mySerial Serial1

#endif

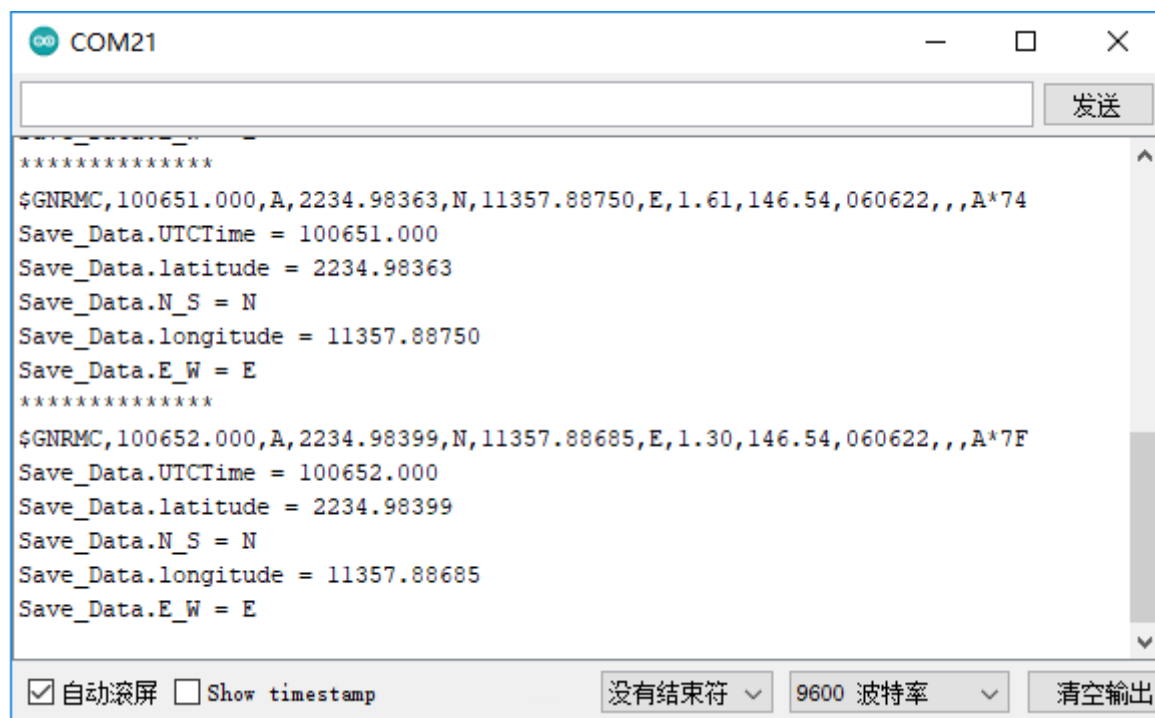
<
上传成功。
项目使用了 7002 字节, 占用了 (21%) 程序存储空间。最大为 32256 字节。
全局变量使用了 922 字节, (45%) 的动态内存, 余留 1126 字节局部变量。最大为 2048 字节。

```

## 5. 实验现象

模块通电后，需要 32s 左右的时间启动，之后模块上的串口打印状态灯会持续闪烁，此时可以正常接收数据。

程序下载后运行，打开串口监视窗口，打开串口软件，波特率设置为 9600，串口会循环打印解析过的实时位置信息。



注意，模块天线需要在室外，否则可能搜索不到 GPS 信号。