GPS 解析输出

1. 学习目标

本次课程我们主要学习使用 STM32F103C8T6 和 GPS 模块模块实现位置信息解析输出功能。

2. 课前准备

GPS 模块采用的是 UART 和 USB 通讯，这里使用 STM32 的 UART 口读取信息，将模块的 TXD 连接 STM32F103C8T6 板子的 PA10 引脚。VCC 和 GND 分别连接 STM32F103C8T6 的 5V 和 GND；TTL 模块的 GND 和 RXD 分别与 STM32 的 GND 和 PA9 连接。

3. 程序

模块的波特率为 9600。

```
NVIC_Configuration();    //设置NVIC中断分
uart_init(9600);    //串口初始化为9600
```

读取并解析接收到的数据。

```
void parseGpsBuffer()
{
  char *subString;
  char *subStringNext;
  char i = 0;
  if (Save_Data.isGetData)
  {
    Save_Data.isGetData = false;
    printf("***************\r\n");
    printf(Save_Data.GPS_Buffer);


    for (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; //获取经度信息
            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;

        }
```

将经纬度信息的单位转化为度

```
// GPS数据转化单位为度。
double Convert_to_degrees(char* data)
{
    double temp_data = atof(data);
    int degree = (int)(temp_data / 100);
    double f_degree = (temp_data / 100.0 - degree)*100/60.0;
    double result = degree + f_degree;
    return result;
}
```

通过串口打印接收到的数据。

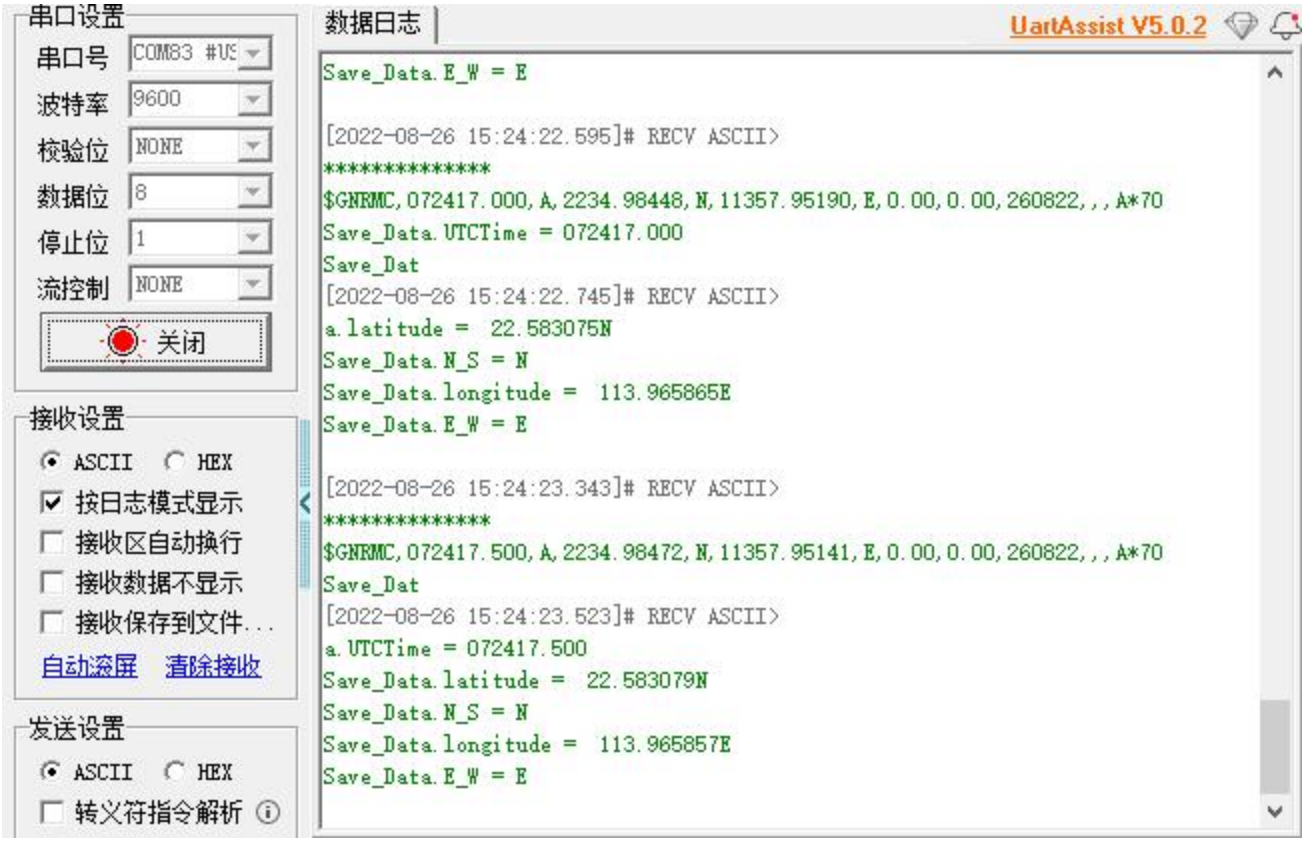
```
void printGpsBuffer()
{
    double f_latitude = 0.0;
    double f_longitude = 0.0;

    if (Save_Data.isParseData)
    {
        Save_Data.isParseData = false;

        printf("Save_Data.UTCTime = ");
        printf(Save_Data.UTCTime);
        printf("\r\n");

        if(Save_Data.isUsefull)
        {
            Save_Data.isUsefull = false;
            printf("Save_Data.latitude = ");
            // printf(Save_Data.latitude);
            // printf("--");
            f_latitude = Convert_to_degrees(Save_Data.latitude);
            printf("%lf%s", f_latitude, Save_Data.N_S);
            printf("\r\n");

            printf("Save_Data.N_S = ");
            printf(Save_Data.N_S);
            printf("\r\n");

            printf("Save_Data.longitude = ");
            // printf(Save_Data.longitude);
            // printf("--");
            f_longitude = Convert_to_degrees(Save_Data.longitude);
            printf("%lf%s", f_longitude, Save_Data.E_W);
            printf("\r\n");

            printf("Save_Data.E_W = ");
            printf(Save_Data.E_W);
            printf("\r\n");
        }
        else
        {
            printf("GPS DATA is not usefull!\r\n");
        }

    }
}
```

注意：其实 GPS/北斗定位的坐标系值并不是简单的 100 倍关系，而是需要做一次度分秒的转换的。则我们获取的 GPS/北斗坐标值，如北纬 2429.53531，东经 11810.78036，需要做如下计算：24+（29.53531/60）≈ 24.49225517 118+（10.78036/60）≈118.17967267。并且不同单片机可能存在数据转化精度的问题而存在一定误差。

4. 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，打开串口软件，波特率设置为 9600，串口会循环打印现在的位置信息。



注意，模块天线需要在室外，否则可能搜索不到 GPS 信号。