

## GPS 数据解析

### 1. 学习目标

本次课程我们主要学习使用 STC89C52RC 型号的 51 单片机和 GPS 模块实现位置信息解析功能。

### 2. 课前准备

GPS 模块采用的是 UART 和 USB 通讯，这里使用 C51 的 UART 口读取信息，将模块的 TX 连接 51 板子的 P3.0 引脚。VCC 和 GND 分别连接 5V 和 GND。

### 3. 程序

初始化串口和数据数组

```
{  
    unsigned char i = 0;  
    Uart_Init();  
    Delay_ms(10);  
  
    UartPrintf("Welcome to use!");  
  
    clrStruct(); //清空缓存数组
```

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;
        UartPrintf("*****\r\n");
        UartPrintf(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;
                }
            }
        }
    }
}

```

通过串口打印接收到的数据。

```

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

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

        if(Save_Data.isUsefull)
        {
            Save_Data.isUsefull = false;
            UartPrintf("Save_Data.latitude = ");
            UartPrintf(Save_Data.latitude);
            UartPrintf("\r\n");

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

            UartPrintf("Save_Data.longitude = ");
            UartPrintf(Save_Data.longitude);
            UartPrintf("\r\n");

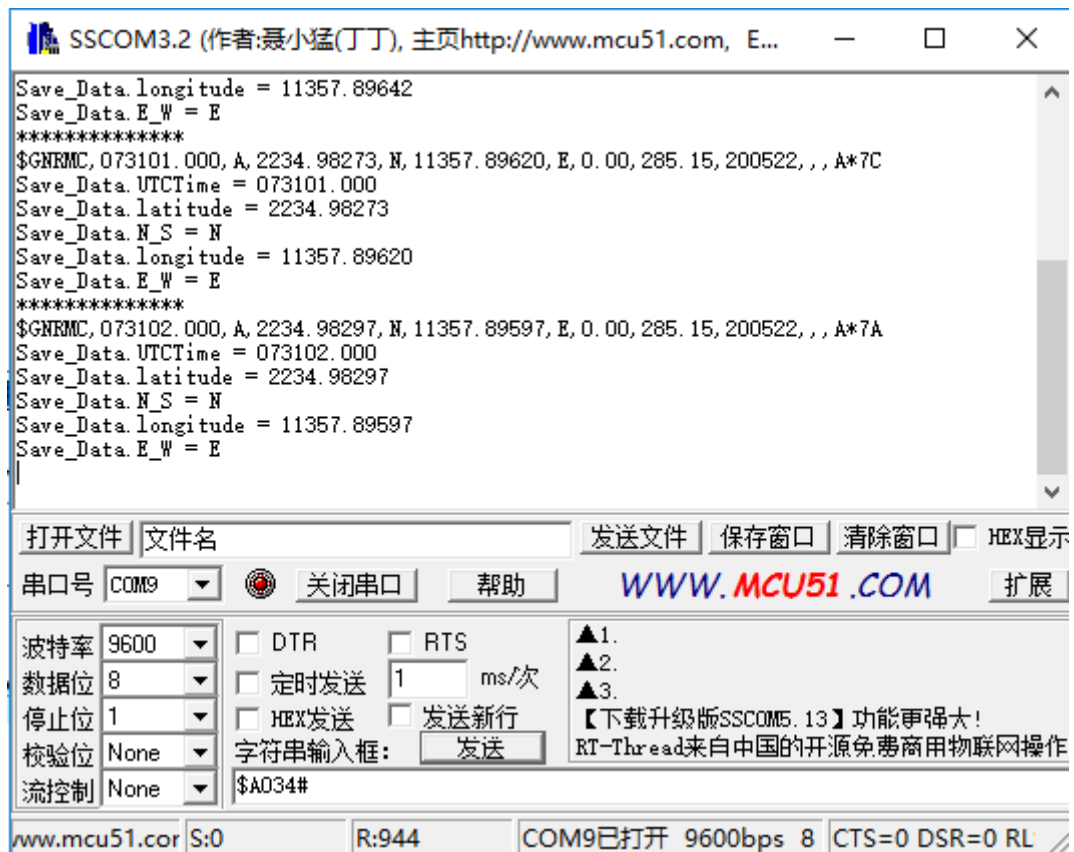
            UartPrintf("Save_Data.E_W = ");
            UartPrintf(Save_Data.E_W);
            UartPrintf("\r\n");
        }
        else
        {
            UartPrintf("GPS DATA is not usefull!\r\n");
        }
    }
}

```

#### 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。