

用 VC + + 6.0 实现计算机与 GPS 通信

陈静 邱国廷 许劼

(武汉理工大学自动化学院, 湖北武汉, 430070)

[摘要] 本文首先介绍遵循 NMEA0183 协议的 GPS 数据, 然后通过实例详细地介绍基于 VC + + 6.0 的计算机与 GPS 串口通信应用程序。

[关键词] GPS; NMEA0183 协议; 串口通信; MSComm 控件

[中图分类号] P228.4 [文献标识码] A [文章编号] 1001-8379(2004)02-0062-02

USING VC + + 6.0 TO DESIGN THE GPS-COMPUTER COMMUNICATION APPLICATION

CHEN Jing QIU Guo-ting XU Jie

(Wuhan University of Technology, Wuhan 430070, China)

Abstract: The paper introduces the GPS data conform to the NMEA0183 protocol and an example which uses VC + + 6.0 to design the GPS-computer communication application is provided.

Key words: GPS; NMEA0183 protocol; Serial communication; MSComm controls

GPS (Global Position System) 即全球定位系统, 利用 24 颗 GPS 卫星的测距和测时功能进行全球定位。在许多系统中, 如机场导航系统、出租车辆管理和调度系统、江河流域的灾害信息管理和预测系统中, GPS 得到了广泛的应用。

对于用户来说, 关键的设备就是用户接收机 (GPS 接收机)。很多情况下, 用户都是用计算机和 GPS 接收机通信, 将 GPS 信息导入计算机然后再处理。而 VC 由于其在和 GPS 接收机的通信及后面的数据处理应用方面的强大功能, 成为许多用户开发 GPS 应用程序的首选开发语言。本文介绍的 GPS 数据采集程序采用 GPS 的异步串行传送方式, 通过 D 型 RS232C 串行口采集遵循 NMEA0183 协议的 GPS 数据。

1 NMEA0183 协议

表 1 描述了 NMEA0183 通信协议部分程序所需 GPRMC 定位语句的结构, 其各字段含义如下:

\$ GPRMC: 其中 \$ 字符为所有 GPS 语句的起始标志, GPRMC 为地址域值, 代表 GPS 推荐的最短数据。

编号为 1 至 8 的字段都是数据字段, 每个字段

的起始都为逗号, 表示分隔。

编号为 1 的字段代表 UTC_TIME24h (24 小时) 制的标准时间, 格式为 “, HHMMSS”, HH 表示小时值, MM 表示分钟值, SS 表示秒钟值。

编号为 2 的字段表示通信状态, 值为 A 或者 V, A 表示数据 “OK”, V 表示一个警告。

编号为 3、4 的字段表示纬度值, 格式为 “, XXYY. YYYY, Z”, XX 表示度; YY. YYYY 表示分; Z 区分南纬或北纬, “N” 表示北纬, “S” 表示南纬。

编号为 5、6 的字段表示经度值, 格式为 “, XXXYY. YYYY, Z”, XXX 表示度; YY. YYYY 表示分; Z 区分东经或西经, “W” 表示西经, “E” 表示东经。

编号为 7 的字段表示 UTC_DATE 标准日期, 格式为 “, DDMMYYXX”, DD 表示天, MM 表示月份, YY 表示年度。年份应该在 1997 ~ 2040 之间。

编号为 8 的字段为校验和域, 格式为 “.. XX”, XX 表示校验和。

GPRMC 定位语句的最末是回车换行符 (CR LF, 即 0x0d 0x0a), 为语句结束标志。

表 1 GPRMC 定位语句结构

\$ GPRMC	123456	A	3444, 1234, N	13521, 4567, E	220199	* 10	CR LF
Field #	1	2	3 4	5 6	7	8	

如果与卫星的通信正常，可以从串口读出以下形式的字符串：

```
$ GPRMC , 201100 , A , 4147. 2316 , N , 12324. 7801 , E , 180503 , * 16
```

根据字符串可以得到依次以下测量数据：格林威治时间 20：11：00，数据接收成功，北纬 4147.2316 度，东经 12324.7801 度，2003 年 5 月 18 日，校验位。

2 用 MSComm 控件进行串口编程

Microsoft Communications Control (MSComm) 是 Microsoft 公司提供的简化 Windows 下串行通信编程的 ActiveX 控件，通过串行口传输与接收数据，为应用程序提供串行通信功能，该控件在串口编程时非常方便，程序员不必去花时间了解较为复杂的 API 函数，而且在 VC++、VB、Delphi 等语言中均可使用。

1) 在工程中插入 MSComm 控件。MSComm 不是工具箱控件，在 VC 中如果要使用 MSComm 控件，首先应将该控件插入到工程中。

选择“Project→Add To Project→Components and Controls”菜单命令，在弹出的 Components and Controls Gallery 对话框的 Registered ActiveX 文件夹中选中“Microsoft Communications Controls, Version6.0”选项，单击“Insert”按钮就插入了 MSComm 控件。如果工程中的视类是基于 CFormView 的，那么创建 CMSComm 类非常简单，只需插入 MSComm 控件，再为该控件声明一个变量即可，本程序中为 m_Com。如果视类不是基于 CFormView 的，那么就需要建立一个 MSComm 控件，创建方法与普通 ActiveX 控件的创建类似，本文不再详述。

2) 初始化并打开串口。对串口初始化通常要完成以下几个设置：设定通信端口号，即 CommPort 属性；设定通信协议，即 HandShaking 属性；设定传输速率等参数，即 Settings 属性；打开通信端口，即将 PortOpen 属性设为 True。

```
m_Com. SetCommPort ( 1 ); //选择 COM1
m_Com. SetInBufferSize ( 1024 ); //设置输入缓冲区的大小
if ( ! m_Com. GetPortOpen ( ) ) //打开串口
m_Com. SetPortOpen ( TRUE );
m_Com. SetInputMode ( 1 ); //设置输入方式为
```

二进制方式

```
m_Com. SetSettings ( " 9600 , n , 8 , 1 " ); //设置数据传输速率等参数
```

```
m_Com. SetRThreshold ( 1 ); //为 1 表示有一个字符引发一个事件
```

```
m_Com. SetInputLen ( 0 );
```

3) 捕捉串口事项。MSComm 控件提供两种处理通信的方式：事件驱动方式和查询方式。查询方式是一种最直接的读串口方式，但定时查询存在一个致命弱点，即查询是定时发生的，可能发生的过早或过晚。在数据变化较快的情况下（如本例接受 GPS 信号）特别是主控计算机的串口通过扩展板扩展至多个时，需定时地对所有串口轮流查询，此时容易发生数据的丢失。虽然定时间隔越小，数据的实时性越高，但系统的资源也被大量占用。事件驱动方式是一种高效的读串口方式，这种方式的实时性较高，特别是对于扩展了多个串口的情况，并不要求像查询方式那样定时地对所有串口轮流查询，而是像中断方式那样，只有当设定的事件发生时才进行相应处理以避免数据丢失，因此本程序采用事件驱动方式。

在 MainFrm. h 中添加：

```
protected :
afx_msg void OnCommMscomm ( ) ;
DECLARE_EVENTSINK_MAP ( )
```

在 MainFrm. cpp 中添加事件的映射：

```
BEGIN_EVENTSINK_MAP ( CmainFrame , CframeWnd )
ON_EVENT ( CmainFrame , ID_COMMCTRL , 1 ,
OnCommMscomm , VTS_NONE ) //映射 ActiveX 控件事件
END_EVENTSINK_MAP ( )
```

4) 读串口数据。完成读串口的函数很简单，即 GetInput ()，该函数原型是 VARIANTGetInput ()，本程序将从串口接收的数据保存到字符串变量 m_strReceive 中：

```
VARIANT vResponse ;
int k , i = 0 ;
char * str1 , * str2 ;
if ( m_Com. GetCommEvent ( ) == 2 )
{k = m_Com. GetInBufferCount ( ) ; //接收到的字符数目
(下转第 84 页)
```

生成 MDE 文件的过程,实际上是对数据库进行编译、自动删除所有可编辑的原代码并压缩目标数据库的过程,其操作方法为:

关闭数据库,并为数据库建立一个副本,单击 Access 菜单栏的工具,选择“数据库实用工具”中的“生成 MDE 文件”,将“测量设备管理系统”输入到“文件名”框内,然后单击“保存”,即完成了文件的生成操作。

7 结束语

该数据库配置要求不高,界面友好,系统以接收单形式形成页面,全部为人机对话模式,直接简便。可自动生成页码序号,避免了人为错误的发

生,对于仪器的型号、仪器的编号、送修目的、故障原因、送修日期、修检人、修检费用等都一一体现在界面上,并能根据使用者的需要,方便地进行修改和编写。此测量设备管理数据库在本单位使用后,不仅减少了大量的人工劳动,提高了劳动效率,而且易于查询和统计,是管理者进行量化管理和工作总结的良好助手。

[收稿日期] 2004-03-15

[作者简介] 刘志鹏(1971-),男,1994年7月毕业于武汉测绘科技大学光学仪器及电子技术系,同年进入铁道第二勘察设计院仪器修理厂,负责全站仪、测距仪的维修服务工作。

(上接第 63 页)

```
if (k > 0)
{vResponse = m_Com. GetInput (); //读串口
str1 = (char *) (unsigned char *) vResponse.
parry ->pvData ;}
str2 = str1 ;
while (i < k)
{ i ++ ; str2 ++ ;}
* str2 = '\0' ;
m_strReceive = (const char *) str ;
UpdateData (False) ;
```

5) 关闭串口。使用完 MSComm 通信对象后,需要将通信端口关闭。如果窗体注销时, MSComm 控件注销可以完成这一功能,可以用下面的语句来实现该功能:

```
m_Com. SetPortOpen (FALSE)
```

3 结束语

利用 Visual C++ 6.0 提供的 MSComm 控件,采用 CMSComm 类的事件驱动方法在实现串行通信中具有编程简单、可视化强、使用稳定等优点,使计算机和 GPS 接收机之间的数据传输更加快捷准确,在利用 GPS 进行导航和定位系统中得到了广泛的应用。

参考文献

- [1] 谭思亮, 邹超群, 等. Visual C++ 串口通信工程开发实例导航 [M]. 北京: 人民邮电出版社, 2003.
- [2] 刘大杰, 等. 全球定位系统 (GPS) 的原理与数据处理 [M]. 上海: 同济大学出版社, 1996.
- [3] 侯俊杰. 深入浅出 MFC [M]. 武汉: 华中科技大学出版社, 2001.

[收稿日期] 2004-04-07

[作者简介] 陈静(1965-), 汉, 女, 副教授, 研究方向: 计算机控制技术与信息系统集成、智能控制理论与应用。