

GPS 导航数据提取的设计与研究*

牛立, 王景中

(北方工业大学 信息工程学院多媒体实验室, 北京 100144)

摘要: 为解决盲用定位模块在嵌入式平台下实现的问题, 根据串口通信同步方式实现 Windows CE 5.0 操作系统下 GPS 导航数据的提取, 并对获得的数据提出一种格式转换的方法。以 PXA270 嵌入式系统平台实现系统原型, 在有限硬件条件下验证了系统的功能及可行性。实验结果表明, 该系统运行稳定, 实验数据可靠有效, 可以有效结合电子地图数据提取出周边的位置信息。

关键词: Windows CE; GPS; 串口通信; 同步方式

中图分类号: TP311.1

文献标识码: A

文章编号: 1674-7720(2010)20-0001-03

Abstracting position data from GPS navigation information based on EVC

NIU Li, WANG Jing Zhong

(North China University of Technology, Beijing 100144, China)

Abstract: To resolve the problem of realizing the GPS model for blind man in the embedded system, abstracting position data output from GPS by synchronous mode of a serial port at Windows CE 5.0 flat roof in circumstance of EVC is introduced, and the format transformation to position data is proposed. The prototype system is implemented by PXA270 embedded system, so the feasibility and its function is validated in the limited hardware environment. Experiments show that the system run stably, the data from the experience is effective, and using the data can abstract the circular position information to combine with the digital map data.

Key words: Windows CE; GPS; serial port communication; synchronous mode

Windows CE 是一个开放的、可裁剪的、32 位实时嵌入式窗口操作系统, 具有可靠性好、实时性高、内核体积小等特点, 广泛应用于各种智能设备的开发。系统通过微软提供的 Platform Builder 定制需要的 Windows CE5.0 系统, 运行在硬件平台上。硬件平台采用博创科技 PXA270 实验箱, 该实验箱嵌入式处理器是基于 ARMV5E 的 Xscale 核心 PXA270, 并支持串口通信。

GPS 导航芯片采用天宝 iQ 46240, 将接收到的数据通过串口发送给处理器。串口是计算机系统与外部串行设备之间的数据传输通道, 是嵌入式通信最可靠、最通用的通信方式。程序员利用 Windows API 函数可以编写出高效、可移植性的应用程序。Windows CE 不支持 Windows 下常用的串行通信异步 I/O 方式(Overlapped, 非阻塞), 因此在嵌入式环境下采用了同步 I/O 方式的通信程序设计方法。

实验设计根据 GPS 导航数据有效性确认的标准, 对提取的数据进行处理, 把缓存中接收到的 GPS 数据格式

转化为电子地图上常用的浮点型格式。此设计已应用于智能阅读器盲用定位模块中。

1 串口通信同步 I/O 方式的程序设计

串口通信是串行通信的一种, 串行通信的模式一般分为上位机和下位机通信。上位机可以读取下位机的状态数据, 也可以设置下位机的状态。一般串行通信协议可分为两类, 即读和写。读写协议的描述如图 1 所示, 常用的效验码有异或、累加和、CRC 等^[1]。

| | | | | |
|-----|------|------|-----|--------------|
| 地址码 | 读功能码 | 参数 | 校验码 | 结束标志 (可选) |
| 地址码 | 写功能码 | 写入数据 | 校验码 | 结束标志 (可选) |

图 1 通信协议的一般结构

在实验设计中, GPS 接收装置作为下位机只负责提供固定格式的数据, 实验箱作为上位机不必发送指令, 只负责定时读取 GPS 接收装置发送的数据。即可以简化

* 基金项目: 十一五国家科技支撑平台重点项目课题(2009BAI71B02)

通信协议,提高工作效率。

1.1 设计开发环境

在 Windows NT/ME 环境下安装 eVC4.0 编程环境,设置顺序如下:

- (1)安装同步软件 Microsoft ActiveSync 4.0;
- (2)安装 eVC4.0;
- (3)利用 PB(Platform Builder5.0)定制 Wince 系统对应的 SDK 并安装;
- (4)利用 PB 将定制的 wince 系统下载到实验箱上,并与 PC 机同步^[2]。

1.2 同步 I/O 方式读取的设计方法

为完成串口通信同步 I/O 方式程序设计,分为三个部分,每个部分有一个函数完成其对应的功能^[3]。函数原型为:

```
OnOpenCom(); //打开并设置串口
ReadThreadFunc(LPVOID lparam); //串口接收线程
OnSeriesRead(CWnd *pWnd, BYTE *buf, int bufLen);
//串口接收数据成功回调函数
```

串口接收的具体流程图如图 2 所示,图中对应了串口设置的三个功能函数。

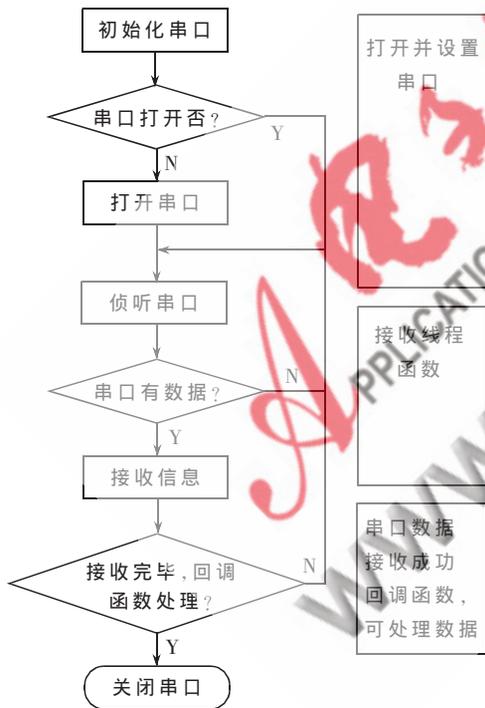


图 2 串口接收的具体程序流程图

1.3 程序设计的核心代码实现

1.3.1 打开并设置通信串口参数

以同步读取方式打开串口 COM1

```
m_hComm = CreateFile (_T ("COM1:"), GENERIC_READ |
GENERIC_READ, 0, NULL, OPEN_EXISTING, 0, NULL);
```

//配置串口,得到打开串口的当前属性参数,修改后再重新设置串口。

```
DCB portDCB;
portDCB.DCBlength=sizeof(DCB); //DCB 结构大小
portDCB.BaudRate=CBR_4800; //波特率
portDCB.ByteSize=8; //字符位
portDCB.Parity=NOPARITY; //奇偶校验位
portDCB.StopBits=ONESTOPBIT; //停止位
//设置串口读写时间,配置超时
COMMTIMEOUTS CommTimeouts;
GetCommTimeouts(m_hComm,&CommTimeouts);
CommTimeouts.ReadIntervalTimeout= MAXDWORD;
CommTimeouts.ReadTotalTimeoutMultiplier=0;
CommTimeouts.ReadTotalTimeoutConstant=0;
//指定端口监测的事件集
SetCommMask (m_hComm, EV_RXCHAR);
//分配设备缓冲区
SetupComm(m_hComm,512,512);
//初始化缓冲区中的信息
PurgeComm(m_hComm,PURGE_TXCLEAR|PURGE_RXCLEAR);
m_hReadCloseEvent=CreateEvent(NULL,TRUE,FALSE,
NULL);
```

1.3.2 GPS 定位信息的接收

在成功打开并设置通信口后,在主程序中创建线程函数 ReadThreadFunc(LPVOID lparam):

```
//创建串口接收线程
hRecvThread=CreateThread(0, 0, CommRecvTread, this, 0,
&IDThread);
```

然后在线程函数中采取事件触发方式进行接收处理,通过等待 EV_RXCHAR 事件的发生来启动 ReadFile 函数完成对 GPS 定位信息的接收:

```
while (TRUE){
if (evtMask & EV_RXCHAR){
ClearCommError(m_hComm,&dwReadErrors,&cmState);
willReadLen = cmState.cbInQue ;
//接收缓冲区中存储的待读取的字符数
readBuf = new BYTE[willReadLen+1];
ReadFile (m_hComm,readBuf,willReadLen,&actualReadLen,
0);
```

```
readBuf[willReadLen]=0; //如果读取的数据大于 0,
if (actualReadLen>0){ //触发读取回调函数
m_OnSeriesRead (ceSeries ->m_pPortOwner,readBuf,actual-
ReadLen); }}
```

如果收到读线程退出信号,则退出线程

```
if (WaitForSingleObject (ceSeries ->m_hReadCloseEvent,500)
== WAIT_OBJECT_0) break;
```

2 对导航数据的格式进行处理

对于实验中所使用的 iQ46240 接收芯片,其发送到计算机的数据(采用 NEMA0183 语句)主要由帧头、帧尾和帧内数据组成。根据数据帧的不同,帧头也不相同,主

软件天地 Software Technology

要有“\$GPGGA”、“\$GPGSA”、“\$GPRMC”等。这些帧头标识了后续帧内数据的组成结构,各帧均以回车符和换行符作为帧尾识别一帧的结束。本文中,定位数据经纬度、速度、时间等均可以从“\$GPGGA”帧中获取得到。该帧的结构及各字段释义如下^[4]:

```
$GPGGA,<1>,<2>,<3>,<4>,<5>,<6>,<7>,<8>,<
9>,<10>,<11>,<12>,<13>,<14>*hh
<1>当前位置的格林尼治时间
<2>纬度值
<3>纬度标识,N 或者 S(南北)
<4>经度值
<5>经度标识,E 或者 W(东西)
<6>卫星接收信号质量
<7>正在使用卫星的数量
```

2.1 GPS 导航数据有效性确认标准

GPS 定位的基本原理是根据高速运动卫星的瞬间位置作为已知的起算数据,采用空间距离后方交会的方法确定待测点的位置。假设 t 时刻在地面待测点安置 GPS 接收机,可以测定 GPS 信号到达接收机的时间,加上接收机所接收到的卫星星历等其他数据可以确定以下 4 个方程式。卫星定位示意图如图 3 所示。

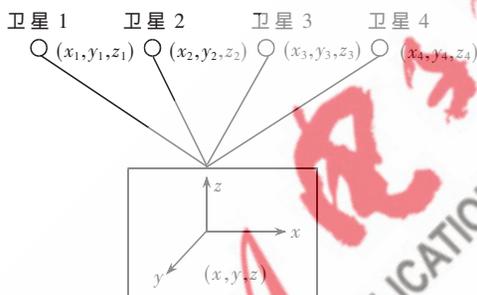


图 3 卫星定位示意图

$$[(x_1-x)^2+(y_1-y)^2+(z_1-z)^2]^{\frac{1}{2}}+c(V_{t_1}-V_{t_0})=d_1$$

$$[(x_2-x)^2+(y_2-y)^2+(z_2-z)^2]^{\frac{1}{2}}+c(V_{t_2}-V_{t_0})=d_2$$

$$[(x_3-x)^2+(y_3-y)^2+(z_3-z)^2]^{\frac{1}{2}}+c(V_{t_3}-V_{t_0})=d_3$$

$$[(x_4-x)^2+(y_4-y)^2+(z_4-z)^2]^{\frac{1}{2}}+c(V_{t_4}-V_{t_0})=d_4$$

上述 4 个方程式中待测点 x 、 y 、 z 和 Δt_0 为未知参数,其中 $d_i=c(\Delta t_i, i=1, 2, 3, 4)$ 分别为卫星 1、2、3、4 到接收机的距离。 $\Delta t_i(i=1, 2, 3, 4)$ 分别为卫星 1、2、3、4 的信号到达接收机所经历的时间。 c 为 GPS 信号的传播速度(c 为光速)。

$x_i, y_i, z_i(i=1, 2, 3, 4)$ 分别代表卫星 1、2、3、4 在 t 时刻的空间直角坐标,可由卫星导航电文求得, V_{t_i} 代表卫星钟差, V_{t_0} 为接收机的钟差。

由以上 4 个方程式可计算出待测点的坐标 x 、 y 、 z 和接收机的钟差 V_{t_0} 。因此导航数据能够有效计算必须保

证接收到 4 个卫星的星历。对固定格式的导航电码中提取卫星符号进行确认,如果满足 4 个卫星的接收状态即可确定当前接收的导航电码可用于数据处理。通常,3 颗卫星可以在二维平面上得到经度纬度坐标,为精确起见,4 颗卫星可以保证获得三维空间坐标。

2.2 有效数据提取和数据格式转化

有效数据的提取和数据格式转化都是在回调函数中进行的^[5]。帧内各数据段由逗号分割,因此在处理缓存数据时,可以通过搜索 ASCII 码“\$”来判断是否是帧头。对帧头类别进行识别后,再通过对所经历逗号的个数计数来判断当前处理的是哪一种定位导航参数,并做相应的处理。eVC 支持 CString 类型格式,由于定位信息格式固定,本文先利用 mbstowcs 函数将缓存中的字符型数据转换为宽字符型,然后强制转化为字符串类型。

```
WCHAR wszbuf[512];
```

```
mbstowcs(wszbuf, (char*)buf, strlen((char*)buf)); 字符串类型进行处理,然后利用 Find 函数,搜索"$GPGGA",
```

```
strRecv.Find(_T("$GPGGA"), 1);
```

```
state=strRecv.Mid(pos+37, 1);
```

```
得到 GPS 质量指示指标
```

```
strSatelliteNum=strRecv.Mid(pos+39, 1);
```

得到接收到的卫星数量字符,将卫星数量字符型转化为整型判断卫星数量是否大于 4,作为判断是否为有效数据的标准。

```
int iSatelliteNum = atoi ((LPCTSTR) (LPCTSTR)strSatelliteNum);
```

当 $iSatelliteNum > 3 \&\& \text{state} == '1'$ 时说明接收到的是有效数据,可对 strRecv 中的数据进行提取,并赋给经纬度和时间变量。

```
strLatitude =strRecv.Mid(pos+16, 8);
```

```
strLongitude=strRecv.Mid(pos+27, 9);
```

将提取到得经度纬度字符型数据转化为浮点型数据,通过 atof 函数实现。

```
double Longitude_new = (atof(strLongitude))/100;
```

```
double Latitude_new = (atof(strLatitude))/100;
```

这样将经度纬度信息提取到 GPS 结构数组中,后续的处理和高层决策可根据该结构中存储的数据作出相应的处理。

3 程序运行结果分析

程序运行后,在实验平台上收集的部分数据如表 1 所示。

对以上结果分析,可知通过串口在不同时段接收的数据是比较稳定的,能够以此为基础提取到有效的数据和定位信息,本系统结合超图格式(pwr, pmw)的北京市公交站点地图数据,在编写程序时调用超图接口函数打开电子地图数据,将串口接收到的数据读入,得到附近的公交站点并以文本方式输出到界面。

表 1 提取到的部分数据

| 定位时间 (年月日时分秒) | 纬度 | 经度 |
|------------------|----------|-----------|
| 20100405112628 | 39.925 0 | 116.125 9 |
| 20100405112630 | 39.924 5 | 116.126 0 |
| 20100405112632 | 39.925 8 | 116.127 3 |
| 20100406085608 | 39.923 8 | 116.128 3 |
| 20100406085633 | 39.923 3 | 116.128 5 |
| 20100406085639 | 39.922 7 | 116.128 8 |
| 20100407100651 | 39.926 1 | 116.123 7 |
| 20100407100708 | 39.927 1 | 116.124 9 |
| 20100407100721 | 39.928 3 | 116.124 5 |

本文结合相关程序代码陈述了基于 eVC 环境下串口通信程序的设计,对 GPS 全球定位系统定位信息的接收和数据提取进行了详细的分析和讨论。主要解决了 EVC 编程环境下实现串口通信功能、对 GPS 定位信息的提取和处理的问题。实验对 wince5.0 自定义平台下开发 GPS 接收装置给出了代码样例供参考,并已应用于智能

阅读器项目中盲用定位模块。

参考文献

- [1] 张文军.GPS 与嵌入式系统软硬件接口及导航信息提取软件[J].计算机工程,2005,31(18):210-212.
- [2] 张冬泉,谭南林,苏树强.WindowsCE 实用开发技术[M].北京:电子工业出版社,2009.
- [3] 钱燕,张继锋.基于 EVC 环境的串口通信程序设计[J].科学技术与工程,2007,7(10):2361-2364.
- [4] 翟羽佳,张晓林,李宏伟.基于 Wince 的 GPS 导航信息处理软件的实现[J].电子测量技术,2007,30(10):93-95.
- [5] 李现勇.Visual c++串口通信技术与工程实践[M].北京:人民邮电出版社,2004.

(收稿日期:2010-05-26)

作者简介:

牛立,男,1982年生,在读硕士,主要研究方向:串口通信,GIS 组件开发,图像处理。

王景中,男,1962年生,教授,主要研究方向:计算机应用,信号与信息处理,信息安全。