

基于 Labwindows/CVI 的北斗短报文上位机设计

张 晓 荷, 苏 新 彦, 姚 金 杰

(中北大学 信息探测与处理技术重点实验室, 太原 030051)

摘要: 为了能够使北斗通信得以广泛的应用, 对北斗短报文的上位机软件进行了设计, 在 Labwindows/CVI 软件开发平台下, 设计了一套进行人机的交互界面, 通过代码的实现以及串口通信函数库, 成功实现了北斗短报文的通信; 重点介绍了北斗短报文的通信功能采用了多线程, 串口通信等关键技术, 给出了功能测试的结果, 将北斗通信的内容得到了完好的接收和发送, 并且该上位机软件性能稳定, 满足实际中对北斗短报文通信的实际要求。

关键词: Labwindows/CVI; 北斗短报文; 串口通信

Design of Beidou Short Message Upper Computer Based on Labwindows/CVI

Zhang Xiaohe, Su Xinyan, Yao Jinjie

(State Key Laboratory of Electronic Testing and Proceeding Technology, North University of China, Taiyuan 030051, China)

Abstract: In order to make Beidou communication widely used, the upper computer software of Beidou short message is designed. Under the development platform of Labwindows/CVI software, a set of interactive interface for man-machine was designed, and the communication of Beidou short message was successfully realized through code realization and serial communication function library. This paper mainly introduces the communication function of Beidou short message using multi-thread, serial communication and other key technologies, and gives the result of functional test. The content of Beidou communication is well received and sent, and the software performance of the upper computer is stable, which meets the actual requirements of Beidou short message communication in practice.

Keywords: Labwindows/CVI; Beidou short message; serial port communication

0 引言

目前, 国外大多数采用 GPS 通信相对比北斗通信精度更加准确, 为了能够提高北斗的大大推广与使用, 北斗短报文通信在国内展开一系列的应用。北斗在国内的优势主要体现在方方面面。例如, 北斗短报文通信在生活中, 军用方面等起着不可替代的作用。通信抄表中, 海上船员健康监测中, 泥石流监测系统中。^[1]根据北斗短报文的广泛应用, 分析北斗短报文通信的基本性能, 设计了基于 Labwindows/CVI 的上位机北斗短报文通信软件, 软件与系统硬件的传输由 Visual C++ 开发的串口通信调用 RS232 接口驱动程序完成。进行完成北斗短报文的通信, 能够完成自检信息和 IC 地址的检测, 根据用户的操作可以每间隔 60 秒进行一次接收与发送通信。最终通过测试顺利完成了基于 Labwindows/CVI 的上位机北斗短报文通信。

1 系统软件设计

1.1 总体设计

北斗短报文主要由上位机软件和北斗信通 BDSC-01 型数传一体机硬件部分组成。上位机软件主要采用模块化设计, 首先利用主函数, 串口通信等框架把上位机主要实

现的功能和程序的基本实现过程进行搭建, 并定义好各个模块之间的通信关系, 得到以各个模块为单位的算法实现, 针对各个模块进行开发, 通过串口将不同的功能模块联系在一起。

上位机的软件开发在架构上有 7 个功能模块: 参数初始化模块, 串口通信模块, 系统自检模块, 定时接收模块, 定时发送模块, 清空信息模块, 生成报表模块, 数据显示模块。如图 1 所示。

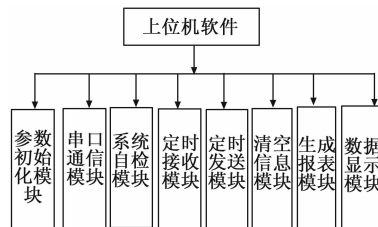


图 1 上位机软件模块组成结构图

参数初始化模块将串口通信的 COM 口清零置位操作, 以便串口通信模块的顺利进行, 系统自检模块通过与硬件天线相结合等待自检完成, 获取天线上的 IC 地址。串口通信连接完成, 自检结束用户可以开始进行消息的定时发送和传输, 位置的获取, 将接收到的消息, 以及位置信息生成报表保存到 Word 文档中进行保存。最终各个模块的信息测试得到的数据都将显示在 Labwindows/CVI 界面上, 在

收稿日期:2018-09-17; 修回日期:2018-10-18。

作者简介:张 晓 荷(1993-), 女, 河北衡水人, 硕士研究生, 主要从事北斗定位方向和信号处理方向的研究。

上面运行上位机界面时将看不到定时器的相关信息，会出现在界面上两个倒计时的数字，以使用户能够准确的进行信息的发送。

1.2 功能模块

上位机软件和北斗天线通过 RS232 串口总线进行连接，总线串口软件使用编写的串口通信函数动态库与 RS232 通信，并进行波特率，数据位，校验位，停止位的配置^[2]。配置完成打开串口，进行系统自检，判断是否进行进行到系统自检，当判断是否进行自检时，进行了一步步判断依次进行当上位机首先检测到不是系统自检时，进行下一步判断是否进行通信申请当进行通信申请正常时，依次进行下一步的自检信息，当自检信息依次上面流程完成时，便可进入正常的信息的发送与接收。获取北斗天线的 IC 地址值，地址值获取完成即可进行地址的获取以及信息的发送与接收，不管发送数据还是进行接收数据都是识别人机界面发生的事件，从而响应用户的操作，进行各种指令的后台准备，可以对每次发送消息的字符限制进行测试。并将获取来的地址进行加以分析。北斗短报文的上位机可以为消息的获取，以及地理位置的准确定位，同时进行时间上的间隔以免消息发生漏传，漏发^[3]。经过总体的把握和具体实际开发，最后形成的人机交互界面如图 2 所示。

图 2 是上位机主界面，在进行上位机的自检之前，应当先进行串口通信，确保在通信之前所有的串口进行初始化配置。

1.3 硬件组成部分

北斗短报文的硬件部分基于 BDSC-01 型数传一体机，针对北斗电力抄表、水利雨量监测、海洋浮标数据传输设计的一款支持北斗 RDSS、RNSS 以及 GPS 的多功能北斗通信终端。可实现 RDSS 的双向定位、短报文通信、位置报告以及 RNSS 的定位和授时功能。终端采用一体化设计，固定杆安装更为方便，配备串口，便于用户进行二次开发。终端体积小、功耗低，连接简单、操作方便，可广泛应用于电力、水利、海洋等行业。

全天候的双向短报文通信和导航定位功能；支持 BDS/GPS 双模工作，保证导航定位更加稳定；集成化程度高，RDSS 模块、RNSS 模块及天线融于一体；采用防水设计 IP67，可在室外可靠工作。相对于北斗短报文的通信模块用户通过发送通信指令，实现短报文通信功能；短报文通信需要对卫星发射信号并占用服务频度；短报文最大字节数由用户卡所决定；单次发送的电文字节数不能超过用户卡所规定的字节数，否则多余的字节数将会被丢弃。采用插卡模式用户机要正常使用，必须插入北斗专用的 SIM 卡。

SIM 卡主要参数，卡号：用户卡的唯一识别号码；入站频度：用户机在两次发送的最小时间间隔，单位秒；最大发送电文 BIT 数：用户机能够一次发送的最大信息 BIT 数。北斗双向定位用户通过发送定位指令，实现双向定位功能；双向定位需要对卫星发射信号并占用服务频度；卫星收到请求指令后，会反馈给用户机当前所在位置的经度、

纬度和高程；这部分主要是作为北斗短报文上位机的辅助部分，作为参考使用。

北斗短报文的硬件使用时应该注意用户机底座装有安装支架，可以用喉箍将装好安装支架的用户机牢牢地绑定到固定物上；请确保用户机圆形天线端朝上，并安装牢固；确认是否有东西会遮挡用户机，如果用户机被遮挡请及时清理，否则将使用户机性能变差或失效。用户机未连接电缆时，必须给电源接口装上密封套。如图 3 所示。



图 2 上位机主界面



图 3 北斗短报文硬件组成

2 软件设计中的关键技术

2.1 串口通信技术

基于 Labwindows/CVI 的串口通信在基于北斗短报文的通信平台中提供了极大方便。在上图 2 中打开串口，进行串口初始化，设置串口通信。包括端口参数串口号，波特率，数据位，停止位，奇偶校验位等参数设置。^[4-5]在 Labwindows/CVI 中的 RS232 函数库中提供了串口通信的控制函数：打开和关闭函数；串口的读写函数；串口设置函数以及回调函数。其中部分被用到打开关闭函数如下：OpenCom（打开串口），OpenComConfig（打开串口同时设置通讯参数），CloseCom（关闭串口）；串口的读写函数如下：ComRd（从串口读一个字符串），ComWrt（从串口写一个字符串）；串口设置函数如下：SetCTSMODE（设置硬件握手），FlushInQ（清空接收队列）^[6-7]，FlushOutQ（清空接收队列）；SetX-Mode（允许或禁止串口通信的软件握手）。

使用基于 Labwindows/CVI 开发环境串口通信程序不仅仅降低了开发效率，同时大大缩短了测试时间，提高了可靠性并且具有很高的通用性。相对于其它上位机软件的编写而言再该程序软件中主要图形界面应用简单大方得体，使得开发者很容易进行程序的上手，便于参考着的使用。良好的用户界面（GUI）强大库函数和便捷的调试手段，能够更好更优的开发。部分串口通信代码如下：

```
GetCtrlVal(panelHandle, PANEL_COM_B, &portNumber);
//串口号
if(portNumber>0)
{
    GetCtrlVal ( panelHandle, PANEL_OPEN_COM_B,
    &getFlag);
    if(getFlag == 1)
    {
        GetCtrlVal ( panelHandle, PANEL_BAUDRATE_B,
        &baudRate) //波特率
```

```

GetCtrlVal(panelHandle, PANEL_PARITY_B, &parity);
//校验位
GetCtrlVal ( panelHandle,  PANEL _ DATABITS _ B,
&.dataBits); //数据位
GetCtrlVal(panelHandle,PANEL_STOPBITS_B, &.stopBits);
//停止位
sprintf(deviceName,"COM%d",portNumber)
COM_B = portNumber;
ret = OpenComConfig ( portNumber, deviceName, baudRate,
parity, dataBits, stopBits, 512, 512);
if (ret < 0) MessagePopup (" RS232 错误"," 配置串口 B
失败!");
SetComTime (portNumber, 5);
InitCOM_B(1);
SetCtrlAttribute(panelHandle, PANEL_SELFTEST_B, AT-
TR_DIMMED, 0);
}
else if(getFlag == 0)
{
SetCtrlVal(panelHandle, PANEL_XTZJ_LED_B, 0);
Sleep(500);
CloseCom (portNumber);
InitCOM_B(0);
SetCtrlAttribute(panelHandle,PANEL_SELFTEST_B, ATTR
_DIMMED, 1);
SetCtrlAttribute ( panelHandle,  PANEL _ SEND _ B, ATTR _
DIMMED, 1);
}
}
}

```

2.2 北斗协议分析

通过串口精灵监视得到的数据根据北斗通信协议加以分析。打开串口后进行上位机的 XTZJ (系统自检), ICJC (IC 检测), TXSQ (通信申请), ZJXX (自检信息) 依次进行完之后获取得到自检成功同时获得 IC 地址, 监视数据如图 4。

序号	时间	进程(PID)	IO类型	CO	数据	数据
802	09:16:14...	北斗RDS	IRP_MJ_CREATE	COM3	0	
803	09:16:14...	北斗RDS	IOCTL_SET_B	COM3	4	9600
804	09:16:14...	北斗RDS	IOCTL_SET_R	COM3	0	
805	09:16:14...	北斗RDS	IOCTL_SET_D	COM3	0	
806	09:16:14...	北斗RDS	IOCTL_SET_U	COM3	3	WordLength: 8StopBits: 1Parity: 0(None)
807	09:16:14...	北斗RDS	IOCTL_SET_C	COM3	6	Chars EOF: 0, Error: 0, break: 0, Event: 113, Xon: 17, Xoff: 1
808	09:16:14...	北斗RDS	IOCTL_SET_H	COM3	16	DCB fRtsControl=RTS_CONTROL_ENABLEDCCB EDrContr
809	09:16:18...	北斗RDS	IRP_MJ_WRITE	COM3	15	24 58 54 5A 4A 00 00 00 00 00 00 00 35 0D 0A \$XTZJ#0
810	09:16:18...	北斗RDS	IRP_MJ_WRITE	COM3	15	24 58 54 5A 4A 00 00 00 00 00 00 00 35 0D 0A \$XTZJ#0
811	09:16:18...	北斗RDS	IRP_MJ_READ	COM3	1	24 \$
812	09:16:18...	北斗RDS	IRP_MJ_READ	COM3	1	5A Z
813	09:16:18...	北斗RDS	IRP_MJ_READ	COM3	4	4A Z

图 4 串口监视数据

根据串口精灵监视得到的数据, 对应北斗协议 2.0 分析得出 IC 检测: 24 49 43 58 58 00 16 04 81 1C 00 1D 81 DD 06 00 3C 03 00 00 00 D9 其中 24 49 43 58 58 代表 IC 检测, 中间部分代表帧号, 04 81 1C 代表用户机的 IC 地址 295196, 04 B3 E1 代表 320433, 两个用户机分别是在该实验中进行试验得到的数据, 00 D9 代表校验位发送中北大学回显与北斗一号的用户级数据接口协议相符显示十六进制

如下: 24 46 4B 58 58 00 10 04 81 1C 04 00 00 00 3C 98。进行多次试验进行一次发送汉字, 英文字母, 数字等按照北斗通信协议一一对应多次进行研究查阅相应的编码表得出相应的编码规范能够应用于广泛的通信中, 得到对应十六进制的解析。其中在通信中涉及到了一些进制的转化, 比如 ASCII 与十六进制的转化进行相关计算最终得出相应的准确信息。如下表 1, 表 2 依次表示用户机接口协议。

表 1 用户机, 信息检测接口协议

IC 信息 \$ ICXX	帧号	通播 ID	用户特征	服务频度	通信等级	加密标志	下属用户总数
		24bit	8bit	16bit	8bit	8bit	16bit
	8bit	下属用户地址					
		24bit * 用户数目 (<100)					

表 2 用户机协议

通信申请 \$ ICXX	信息类别 8bit	用户地址 24bit	电文长度 16bit	是否应答 8bit	电文内容最长 1680bit
--------------	-----------	------------	------------	-----------	----------------

在实验中点击自检, 等待自检完成时获取 IC 地址分别获取两个用户机的地址分别为 295196 和 320433 如图 5~6 所示。



图 5 用户机 IC 地址获取 图 6 用户机 IC 地址获取

2.3 多线程技术

上位机软件在设置定时的过程中涉及多线程以及线程池问题, 当接收信息和发送信息时遇到多任务的处理情况, 比如在进行客户端 A 向客户端 B 发送消息时, 同时客户端 B 向客户端 A 发送数据时可以并行进行。若采用单线程变成方法, 需要等待一方数据传输完时才能进行另一方的信息传输, 大大的降低了程序的执行效率, 系统的实时性也难以得到保证。^[8]而多线程程序是在同一段时间内进行至少有两个独立的线程在执行代码, 其中每一个线程都有自己的堆栈情况进行独立的分配, 并且能够进行独立的运行在其他程序的线程中, 即能进行各种保证线程运行的实时性, 又能够响应客户的其它操作, 能够在最大限度上地提高系统的运行速度以及程序的响应速度。^[9]

北斗短报文上位机设计主要在定时中涉及到了多线程池问题, 应用于客户端操作的 CmtThreadPoolHandle pool-Handle 线程池操作, 具有最高的优先级, 保证了客户端 A 和客户端 B 在同时进行消息发送时及时的响应处理。这样就不用担心信息在传输过程中的丢失。

在 Labwindows/CVI 给用户提供了两种线程机制: 线

程池和异步定时器。线程池就像一个线程容器，里面放的就是运行中的线程数据起着保存数据分析数据的作用，采用函数 `CmtScheeduleThreadPoolFunction` 启动一个新线程，定时线程在线程池中开辟。异步定时器是一种在程序中适时生成的定时器，在一定时间内反复执行代码实现功能，生成异步定时器的函数可以是 `NewAsyncTimer`，在函数中配置其间隔时间，事件发生次数以及事件的响应函数，信息显示线程通过异步定时器实现。^[10]使用多线程机制后，信息的读取无丢失，处理以及分析的时间充足，大大提高了程序的执行能力。

2.4 Labwindows/CVI 的定时技术

上位机软件中进行定时接收和定时发送中，使用到定时器间隔一定的时间间隔来触发事件。由定时器控制这些事件进行的速度，用户只能在界面上看到该控件，但是在程序的运行过程中是不可见的。正是因为如此，在上位机程序的运行过程中，用过操作时不能使用鼠标或是键盘去进行操控。定时器会按照用户设定好的时间间隔，产生 `E-VENT_TIMER_TICK` 事件，随后会执行相应的回调函数。其中里面涉及到定时器的两个重要属性分别为：`ATTR_ENABLED` 为定时器是否可以相应控件的回调函数；`ATTR_INTERVAL` 为设置定时器执行回调函数的时间间隔。^[11-12]其中部分代码如下：

```
CopyBytes (len, 0, RecvBuf_B, 4, 2);
ComRd (COM_B, RecvBuf_B, HexConvertDecimal (len, 2) - 7);
CopyBytes (temp, 7, RecvBuf_B, 0, HexConvertDecimal (len, 2) - 7);
if (Checksum (temp, HexConvertDecimal (len, 2) - 1) != temp
[HexConvertDecimal (len, 2) - 1])
{
TimerRecvB = WAIT_RECV_B;
return 0;
}
CopyBytes (ADD_B, 0, RecvBuf_B, 0, 3);
sprintf (addStr, "%ld", HexConvertDecimal (ADD_B, 3));
SetCtrlVal (panelHandle, PANEL_IC_ADD_B, addStr);
SetCtrlAttribute (panelHandle, PANEL_SEND_B, ATTR_DIMMED, 0);
TimerRecvB = WAIT_RECV_B;
break;
case 0x0d: //FKXX
//InsertTextBoxLine (panelHandle, PANEL_RECEIVEBOX_B, -1, "FKXX");
break;
case 0x0c: //TXXX
CopyBytes (len, 0, RecvBuf_B, 4, 2);
ComRd (COM_B, RecvBuf_B, HexConvertDecimal (len, 2) - 7);
CopyBytes (temp, 7, RecvBuf_B, 0, HexConvertDecimal (len, 2) - 7);
if (Checksum (temp, HexConvertDecimal (len, 2) - 1) != temp
[HexConvertDecimal (len, 2) - 1])
```

```
{//TimerRecvB = WAIT_RECV_B;
return 0;
CopyBytes (len, 0, RecvBuf_B, 9, 2);
CopyBytes (temp, 0, RecvBuf_B, 12, HexConvertDecimal
(len, 2));
sprintf (disStr, "%s", temp);
InsertTextBoxLine (panelHandle, PANEL_RECEIVEBOX_B,
-1, disStr);
TimerRecvB = WAIT_RECV_B;
break;
}
}
```

3 测试结果与分析

打开软件客户端 A 串口与客户端 B 串口，根据电脑系统检测到的串口选择好串口号，波特率，校验位，数据位以及停止位，初始准备工作已完成，进行系统自检指示框变成绿色表示已经自检成功，点击系统自检获取 IC 地址如图所示的客户端 A 与客户端 B 的地址，此时表明可以进行正常客户端 A 与客户端 B 的通信，每隔 60 秒定时由客户端 A 向客户端 B 发送数据成功显示在客户端 B 上，同时也测试客户端 B 向客户端 A 发送数据客户端 A 成功显示，如下图 7 所示。



图 7 实验测试结果

4 结论

通过介绍北斗短报文通信的上位机软件设计与实现。设计主要采用以模块化为思路，贯穿整个上位机的设计过程，将系统的所有功能按照模块加以划分，使得程序的结构清晰明了，便于测试，大大的提高了编程效率，增强了软件的可维护性与可扩展性。软件开发过程中使用线程池技术以及定时器技术，得以使信息正确地进行传输；根据北斗通信协议的正确解析与分析使得通信的正常运行；串口通信技术，实现了软件与硬件的连接，顺利完成了多次测试任务，性能稳定，界面简单大方，操作简单，坚实的消息的正常传输更简单化，智能化。

(下转第 278 页)