

海洋定点垂直剖面监测控制系统研制

翁 杰 刘敬彪 于海滨 (杭州电子科技大学 电子信息学院 浙江 杭州 310018)

摘要: 介绍了该系统中控制子系统的设计与实现。控制子系统可分为水下监测数据采集平台和水上通讯平台两部分, 分别实现水下垂直剖面监测数据的获取以及监测数据的就地存储与卫星通讯。水上、水下平台均采用接口资源丰富的 AT91SAM9260 作为核心芯片, 以满足多种剖面数据的采集、存储以及通讯需求。实验表明, 控制子系统功能完善, 运行稳定, 达到设计要求。

关键词: Linux; SD 存储; 水上通讯平台; 监测控制; 数据采集

Research and Design of an Ocean Fixed-Pointed Vertical Profiler Monitor and Control System

WENG Jie, LIU Jing-Biao, YU Hai-Bin

(Electronic Information Institution, Hangzhou Dianzi University, Hangzhou 310018, China)

Abstract: This paper introduces the design and implementation of the control subsystem. The control subsystem can be divided into monitoring data collection platform underwater and communication platform on water which respectively realize collecting and storing monitoring data of vertical profiler and satellite communication. Water platform and underwater platform use AT91SAM9260 as core chip which has abundant port resources, to satisfy various profiler of the data collection, storage and communication needs. Experiments show that the control subsystem functions perfectly, operates stably, which meets the design requirements.

Keywords: Linux; SD storage; water communication platform; monitor and control; data collection

1 引言

1.1 文章安排

本文第 2 节介绍系统组成。第 3 节给出控制系统整体设计。第 4 节给出软件设计。第 5 节给出结论。

1.1.1 基本介绍

随着卫星通讯、卫星遥感、水声遥测以及数据同化等技术的不断发展与成熟, 海洋环境监测已进入从空间、沿岸、水面及水下对海洋环境进行立体监测的时代。然而, 目前我国大部分的海洋环境监测信息资料仍然依靠海洋监测站周期性的采集海洋表层或有限的一些水层的要素数据的方式获取, 对于海洋次表层及以下水体的资料十分稀缺, 特别是除了投放 ARGO 浮标外, 几乎没有什么有效的技术手段来获取海洋垂直剖面的包含任意水层要素数据的连续、实时序列观

测数据, 而仅仅依靠表层或有限水层的要素数据难以对我国广阔海域的物理、生物、化学环境等多方面状况进行深入了解, 无法满足全方位的立体监测要求, 不能适应我国海洋经济发展和防灾减灾的需要。因此, 在当前的经济与技术发展水平条件下, 如何实时、自动获取海面以下水体垂直剖面的连续序列海洋环境要素数据已成为我国海洋环境立体监测中关键的一环。海洋定点垂直剖面监测控制系统正是为了满足这个需求而设计的, 可以获取近海范围内不同工作水深垂直剖面的海洋环境参数。

2 系统组成

海洋定点垂直剖面监控系统主要包括水下监测数据采集子系统和水上通讯平台子系统。水下子系统的核心部分是基于 ARM 嵌入式处理器和 Linux 操作系

收稿时间:2009-08-11;收到修改稿时间:2009-09-05

统的水下数据处理和控制系统。水上通讯平台根据已定的协议,接收到电磁耦合模块传过来的数据,并保存在大容量存储器中。考虑到如果用有线通信方式,则由于通信距离不断变化会导致传输线路设计困难、通信的可靠性都难以保障。而电磁耦合通信正是利用电磁感应原理来传输数据的,它的设计电路简单可靠,体积小,成本低,可以实现在水下的近距离无线传输。图 1 是系统整体结构图,图中的非接触式信号传输部分就是电磁耦合模块。

波浪能通过锚缆传递转化为垂直剖面测量系统水下主体部分(小浮力浮球)下行的动能。系统将充分利用大、小浮球的浮力惯性实现抽缆(相当于水下主体部分的下行)功能,当需上行时,只需打开锁阀。在依靠小浮球浮力上行过程中采集各环境参数变量。

整个一次数据采集、存储,送到水上通讯平台,最后海洋环境数据送到监测平台周期为 24 小时,也就是每隔 24 小时采集各剖面环境量。垂直剖面观测深度为 300m。整个系统维护周期约为三个月一次。

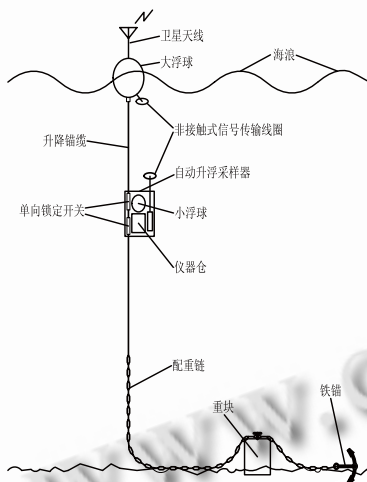


图 1 系统整体结构图

在整个测量系统中,控制系统是核心单元。水下监测数据采集平台的控制块在仪器仓内。它负责指挥和协调系统各部分工作状态。包括控制海洋各剖面数据的采集、存储、数据处理、传输等。水上通讯平台的控制部分主要接收电磁耦合模块传过来的各个采集量,以及获取浮标电压值和浮标姿态等信息,然后通过铱星卫星发送给监控中心。整个监测控制系统的工作流程图如图 2。

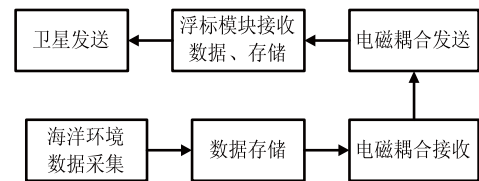


图 2 监测控制系统工作流程图

3 控制系统整体设计

3.1 水下监测数据采集平台设计

考虑到系统水下部分的体积与重量,水下部分携带的电池的容量必然受到较大的限制,而系统又将在很长的一段时期内处于无人值守的全自动工作模式,因此系统中所有的电子线路必须采用低功耗处理技术以降低能耗。普通的单片机集成度不高,许多外设都需要自己扩展,就会增加许多额外功耗,而 ARM 将多种控制器集成在一个芯片内,省去了很多子模块,也大大降低了系统的功耗。对海洋环境的监测需要采集各数据量进行全面的分析,传感器的数据输出类型,大部分采用 RS-232 串口和 A/D 转换来读取数据。基于这些,本系统水下主体部分 CPU 采用 AT91SAM 9260,该芯片包括性能高达 200 MIPS 的 ARM 926EJ-S 内核;7 个通用同步/异步收发器 (USART),多个自带的串口不需要外扩,降低了整体功耗。自带 4 个 10-bit 的 ADC 能支持带有内置 ECC、SD、SDIO 和多媒体卡(MMC)接口的 SDRAM、闪存以及 NAND 闪存。7 个串口最多可以同时接 6 个传感器,剩下一个串口接电磁耦合通信模块(如图 3)。

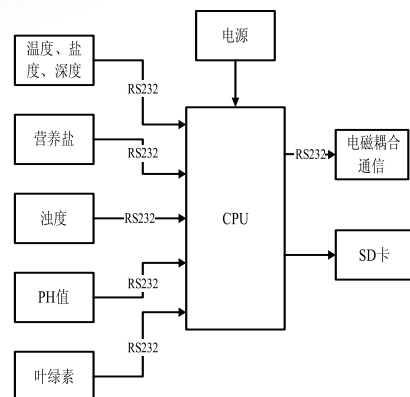


图 3 水下采集平台控制部分结构图

水下监测采集平台的控制电路工作过程是:升降平台在每次上升过程中,控制电路不断的读取深度信

息(CTD), 并按照预定时序, 采集、存贮各个剖面传感器(营养盐、温度、盐度、浊度、叶绿素、PH 值等)送出的数据, 当平台完成一个周期的工作后, 将这个周期的剖面测量数据全部发送给 ARM 并通过大容量的 SD 卡进行存储。

3.2 水上通讯平台设计

通讯平台采用太阳能电池和蓄电池组合供电。平台的主体是浮标部分。为提高可靠性, 有的浮标采用两个独立的供电系统, 每个系统都有蓄电池和太阳能电池板, 都能为整个浮标供电。这种备份尽管不是必需的, 但它提高了浮标的可靠性等级。浮标控制系统的 CPU 同样用 AT91SAM9260, 一路串口按序接收水下电磁耦合模块传过来的各个传感器的采集数据, 并保存在 SD 大容量存储卡中(如图 4)。存于卡中的各路传感器采集的数据通过铱星卫星发送, ARM9 的 COM2 口接 IRIDIUM SBD 9601 模块, 用来与监控中心进行通信。该模块通过 RS232 接口实现 SBD 业务, 默认的波特率是 19200bps, 可以通过 AT+IPR 指令设置。铱星数据通讯的功耗较小, 能够满足海洋浮标设备低功耗的要求。

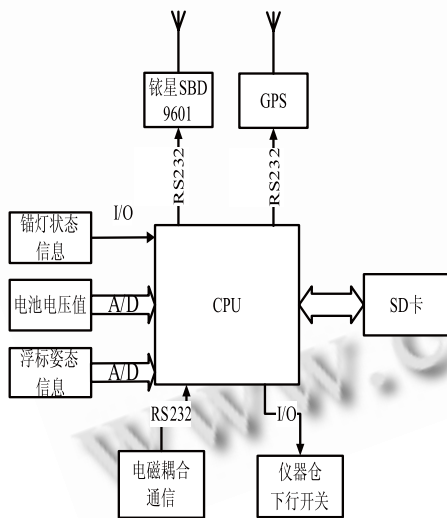


图 4 浮体平台结构图

同时, 浮标上装有锚灯, 内有传感器检测光线, 自动点亮, 给监测带来方便。同时 CPU 通过 I/O 获取锚灯状态信息。

AT91SAM9260 通过自带的 A/D 读取电池的电压值和倾角传感器获得的浮标姿态信息。

浮标上装有 GPS 天线, 用于定位。还装有罗盘、

避雷针等。

4 控制系统软件设计

系统采用了 AT91SAM9260 作为 CPU, 它支持主流的 Linux、Windows XP 等操作系统。系统移植了 Linux 操作系统来进行任务管理和调度, 改变了传统的在类似硬件平台下采用 MSDOS 单用户、单任务操作系统难以完成较为复杂的分布式多任务应用的缺点。在数据采集系统中, Linux 系统可以同时处理多个传感器发送来的较为复杂的控制任务, 从而提高系统的整体响应速度及并发处理能力。

构成嵌入式 Linux 系统至少需要下面 3 个基本元素^[1]: 引导程序、Linux 微内核和初始化进程。如果要想让它有更多的功能, 还可加上文件系统、GUI 和设计精简的应用程序, 并将其放在 diskonchip 中启动。

我们移植好了文件系统。在 Linux 操作系统下, 有三类主要的设备文件类型: 字符设备、块设备和网络设备。字符设备和块设备的主要区别是: 在对字符设备发出读/写请求时, 实际的硬件 I/O 一般紧接着发生; 块设备则不然, 它利用一块系统内存做缓冲区, 当用户进程对设备请求能满足用户的要求时, 就返回请求的数据; 如果不能, 就调用请求函数来进行实际的 I/O 操作。块设备是针对磁盘等慢速设备设计的, 以免耗费过多的 CPU 时间来等待。用户进程通过设备文件来与实际的硬件打交道。每个设备文件都有其文件属性(c/b), 表示是字符设备还是块设备。每个文件都有两个设备号, 第一个是主设备号, 标志驱动程序; 第二个是从设备号, 标志使用同一个设备驱动程序的不同的硬件设备。设备文件的主设备号必须与设备驱动程序在登记时申请的主设备号一致, 否则用户进程将无法访问到驱动程序。

Linux 操作系统将所有设备作为文件来处理^[2], 他们可以使用文件、I/O 相关函数来处理, 这样就方便了对设备的处理。例如: 串口文件位于/dev 目录下, 串口 1 为/dev/ttyS0, 串口 2 为/dev/ttyS1。打开串口用标准的文件打开函数操作。如: `int fd = open(Dev, O_RDWR);` Dev 是设备文件, 返回 -1 则表示打开失败, 成功则返回大于 0 的值。

其他几个主要函数如下:

```
set_speed(fd, 115200) //设置串口波特率函数, fd 对应的串口文件
```

```

set_Parity(fd,8,1,'N') //设置串口接收数据格式,8 位数据位,1 位停止位,无校验
write(fd,buff,nread) //往串口 buff 写 nread 字节数据
nread = read(fd,buff,512) //读串口缓冲区 512 字节数据,返回读取的实际个数
fclose(fp) //关闭文件(包括普通文件和设备文件)
num=fwrite(buff, sizeof(char), nread, fp)
//往 fp 对应的文件写 nread 字节

```

读取一个串口的程序流程图如下:

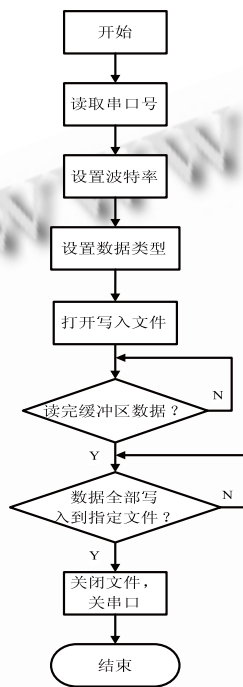


图 5 串口操作程序流程图

对串口的操作^[3]需要用到的头文件有:

```

#include <stdio.h> //标准输入输出定义
#include <fcntl.h> //对文件控制的函数
#include <stdlib.h> // 标准函数库定义
#include <unistd.h> //Unix 标准函数定义
#include <sys/types.h> // 系统函数定义
#include <sys/stat.h> //系统调用的函数头文件
#include <termios.h> //POSIX 终端控制定义
#include <errno.h> //错误号定义
#include <pthread.h> //线程库定义

```

5 结论

系统在实验室 PC 机调试助手模拟数据发送,成功将数据发送到带 Linux 操作系统的目标板上,并可以收到多个串口的数据,把它们保存到一个文件中。实现了多串口数据的采集,在上位机上^[4]动态的显示收到的数据。在水下进行测试过程中,控制电路工作正常,可以实现多串口的数据采集、存储、传输。水上浮体平台数据能正常接收、存储,倾角传感器、电池电压值能通过 A/D 采集。且监测数据通过铱星卫星成功发送给监控中心。

参考文献

- 1 孙天泽,袁文菊,张海峰编著.嵌入式设计及 Linux 驱动开发指南-基于 ARM9 处理器.北京:电子工业出版社, 2006.
- 2 于磊,曾刚.POSIX 多线程程序设计.北京:中国电力出版社, 2003.
- 3 罗苑棠,德.嵌入式 Linux 应用系统开发实例精讲.北京:电子工业出版社, 2007.
- 4 孙鑫,余安萍.VC++深入详解.北京:电子工业出版社, 2007.