

摘要: 本文介绍了智能牵引仪系统的总体结构和功能, 详细介绍了系统设计方案, 以及用 VC++ 实现上位机管理与通信的主要代码。

关键词: 牵引仪 监控 通信规程 接口类 通信程序

1 前言

牵引仪是在研究传统牵引术和中医推拿正骨手法的基础上, 根据腰椎间盘的生理解剖和周围神经组织的特点, 采用机电、液压控制和计算机控制技术研制出的新产品。该仪器在计算机系统的控制下模拟中医正骨手法, 通过牵引、倾角和旋转三个基本动作, 完成对病人的定量三维治疗。

2 技术指标

牵引距离: 0~70mm
 转角角度: 左 8~25 度, 右 8~25 度
 倾斜角度: 上 0~15 度, 下 0~15 度
 牵引力: 3500N~4800N

3 系统结构和功能

整个系统有三部分组成, 如图 1 所示。

3.1 上位机 PC

上位机主要用于管理和监控, 将病人的姓名、性别、年龄、身高、体重、病史、病变斜度等资料保存在数据库中, 治疗时根据病人的身体情况以及以往病史和治疗记录, 设定牵引距离、旋转方向和角度、倾角方向和角度等参数,

智能牵引仪微机 监控系统的设计与实现

鲍蓉 (徐州彭城职业大学计算机系 221008)

通过 RS232 接口将参数传给下位机。治疗过程中实时监测下位机的工作情况, 治疗结束后, 保存本次治疗数据并给出结束信息。

3.2 下位机

下位机从上位机的串行通信口获取治疗参数并保存在下位机的缓存中, 下位机控制步进电机工作设置牵引距离, 控制电磁阀动作实现快牵、旋转和倾斜操作, 并通过传感器采集倾角角度、转角角度、牵引距离等数据进行分析存储, 同时通过 RS232 串口将数据传给上位机。

3.3 牵引床

牵引床用于实现治疗, 由步进电机驱动, 床体可伸缩、倾斜和旋转, 动作的角度可通过距离传感器和转角传感器传送给下位仪, 由下位机进行分析处理。

4 总体设计

4.1 系统配置及开发环境

上位机: 采用 486 以上 PC, 要求 256M 以上的内存; 操作系统为 Windows 2000 Professional, 监控程序的开发工具为 Visual C++ 6.0, 数据库管理系统使用 MS Access。

下位机: 采用 89C51 单片机, 其 I/O 端口为 32 位, 自带一个串口中断和 UART 单元, 可同时



图 1

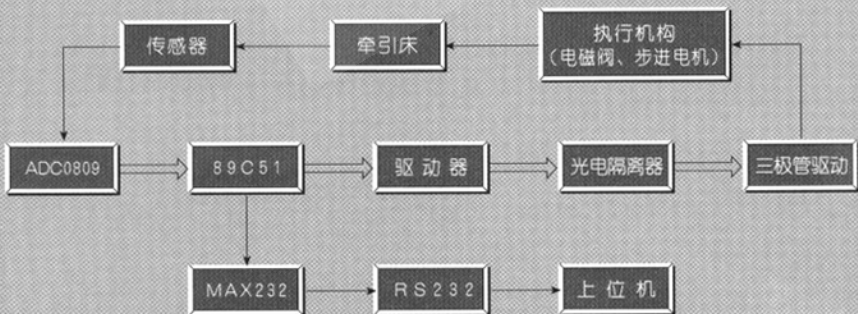


图 2

表 1

字段名称	字段标题	字段类型	字段大小
Num	编号	文本	10
name	姓名	文本	30
Sex	性别	文本	6
Age	年龄	整型	
High	身高	整型	
Weight	体重	单精度型	
pathistory	病史	备注	
Wheresick	发病部位	备注	
Angle	病变斜度数	单精度型	

图 3

图 4

进行数据的发送与接收,也可以做为一个同步移位寄存器使用,下位机的通信处理程序使用汇编语言实现。

4.2 硬件框图

4.3 通信规程

上位机和下位机的通信采用 RS232 串口标准,异步通信方式,信号传送采用同样的波特率 9600bps, 1 位起始位, 1 位停止位, 8 位数据位, 无校验。

4.4 数据库设计

上位机利用数据库管理病人的信息,本系统中建立一个基本表 pathistory, 表结构如下:

5 上位机通信程序设计

本系统使用 Visual C++ 6.0 开发上位机监控程序,系统设计通信接口类 CSerial.cpp, 该类的主要成员函数有: 构造函数 CSerial(), 串口连接函数 OpenConnection(), 设置串口通信方式函数 SetupConnection(),

在 CCureDlg.cpp 中包含 CSerial.h, 系统开始工作时, 运行 PLifeDlg.cpp 文件, 在窗体中输入病人的编号或姓名, 程序连接数据库, 查找病人记录并显示在窗体中, 如图 3 所示,

按下“治疗”按钮后, 调用 CCureDlg.cpp 文件, 显示“设置治疗参数”的窗口, 如图 4 所示,

在窗口中输入牵引距离、倾斜和旋转角度等参数后, 按下“开始治疗”按钮, 程序将打开 RS232 口, 将数据送到串口缓冲区中, 主要程序代码如下:

```
(1) 通信类 CSerial 构造函数 CSerial ()
```

```
.....
```

```
m-bByteSize=8; // 8 位数据位
```

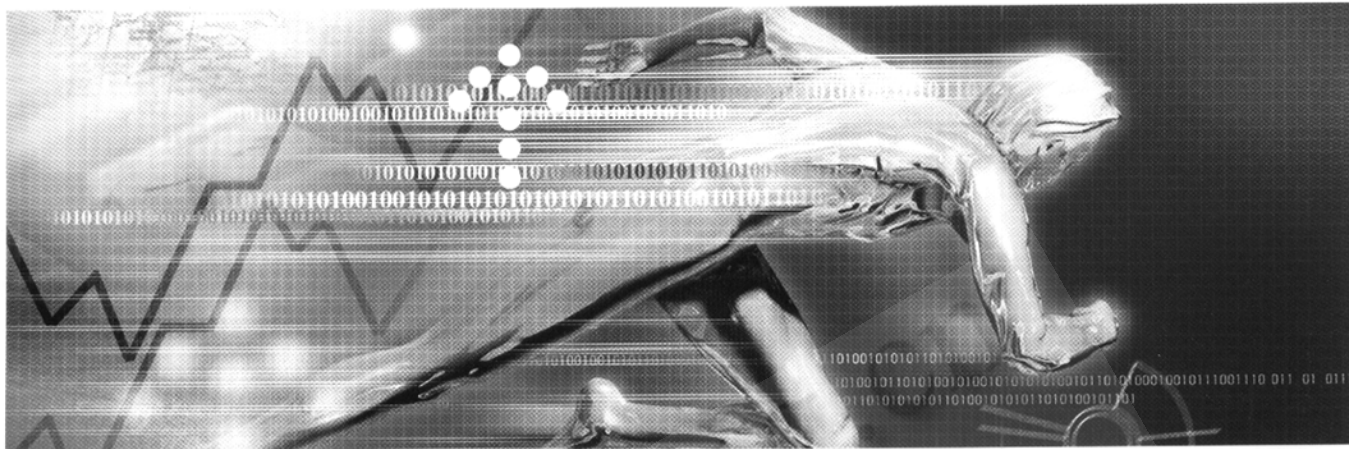
```
m-bStopBits=ONESTOPBIT; // 1 位停止位
```

```
m-bParity=NOPARITY; // 无校验
```

```
m-dwBaudRate=9600; // 传输率为 9600bps
```

```
m-bEvtChar=EVENTCHAR;
```

```
m-fBinary=1;
```



.....

(2) 按下“开始治疗”按钮, 激发CCureDlg::OnBtncure()方法, 本方法将窗口中数据传至COM2端口的缓冲区中。

```
void CCureDlg::OnBtncure()
{
    char buf [30];
    UpdateData(TRUE); // 控件赋值给绑定的成员变量
    m-comDev.OpenConnection(); // m-comDev为串口类CSerial的实例对象, OpenConnection()为串口类CSerial的成员函数
    if(m-comDev.IsConnected){
        sprintf(buf, "%f,%f,%f,%f,%f",
m-fdraw,m-ffront,m-fback,m-fleft,m-fright);
        // 将参数写入 buf
        ::WriteCommBlock(m-comDev,buf,
DWORD dwBytesToWrite); // 将参数写入串口缓冲区
    }
}
```

(3) 在CSerial.OpenConnection()中调用windows API函数CreateFile(), 打开串口

```
m-hCom = CreateFile("COM2",GENERIC-
READ|GENERIC-WRITE, 0, NULL,
OPEN-EXISTING, FILE-ATTRIBUTE-NOR-
MAL|FILE-FLAG-OVERLAPPED, NULL);
```

.....

```
SetupComm( m-hCom, 4096, 4096 ); // 分配设备缓冲区, m-hCom为串口句柄, 是CSerial类的成员变量
```

```
PurgeComm( m-hCom, PURGE-TXABORT|
PURGE-RXABORT |
PURGE-TXCLEAR | PURGE-RXCLEAR );
// 初始化缓冲区中的信息
```

(4) 在CSerial.OpenConnection()中调用CSerial::SetupConnection(), 设置串口通信参数

```
DCB dcb ;
if(m-hCom == NULL)
```

```
return FALSE;
dcb.DCBlength = sizeof( DCB );
GetCommState( m-hCom, &dcb );
dcb.BaudRate = this->m-dwBaudRate;
// 设置波特率
dcb.ByteSize = this->m-bByteSize;
```

```
// 设置数据位
dcb.Parity = this->m-bParity;
dcb.StopBits = this->m-bStopBits ;
dcb.EvtChar = this->m-bEvtChar ;
// 设置硬件流控制
.....
// 设置软件流控制
.....
```

(5) 在WriteCommBlock(m-comDev,buf,

DWORD dwBytesToWrite)中调用WriteFile(), 向串口缓冲区写数据

```
fWriteStat = WriteFile( comDev.m-hCom , buf,
dwBytesToWrite, &dwBytesWritten, &
(comDev.m-wtos) );
```

(6) 上位机读取下位机的数据时, 使用

```
fReadStat = ReadFile( comDev.m-hCom,
lpzBlock,
dwLength, &dwLength, &(comDev.
m-rdos);
```

其余代码类似, 不再详述

6 结束语

·本系统已成功用于牵引床系统, 系统相当稳定可靠, 是目前非手术治疗腰椎间盘突出症的理想设备。 ■

参考文献

- 1 胡汉才 编著, 单片机原理及其接口技术, 清华大学出版社, 2001年。
- 2 田学锋、周予滨, 计算机通信与RS-232接口实用指南, 电子工业出版社, 1995年。
- 3 乔林等 编著, Visual C++6.0程序设计-提高篇, 中国铁道出版社, 2001年。
- 4 陈元琰、邓宗明等 编著, Visual C++ 6.0编程实用技术与案例, 清华大学出版社, 2001年。