

文章编号: 1001-9081(2001)11-0094-02

## 多层螺旋调强放疗装置控制系统的动态连接

朱 华, 蔡 淮, 杨 斌, 蒋 晔, 黄惠萍

(西南交通大学 计算机与通信工程学院, 四川 成都 610031)

**摘 要:**介绍了多层螺旋调强放疗装置控制系统中,上层应用程序与底层控制节点的动态连接过程,主要采用动态链接库与 Lonworks 现场总线技术。

**关键词:**多层螺旋调强放疗装置;动态链接库;Lonworks 现场总线

**中图分类号:** TP311.52 **文献标识码:** A

多层螺旋调强放疗装置是一种用于治疗肿瘤的放射性医疗设备。通过治疗床把患者送入装有放射源的旋转体中,经过治疗床 x、y、z 三个方向的定位,打开旋转体中的放射源,对患者进行放射性治疗。该装置集中体现了近年放射肿瘤治疗技术的发展,对肿瘤防治事业有巨大和深远的影响。

该装置控制系统的主要目标是完成对整个装置的定位及治疗过程的控制。控制系统采用 Lonworks 现场总线技术,结构简单、可靠性高,能满足治疗设备的控制和安全要求。

### 1 控制系统的组成和特点

该系统的控制对象包括治疗床、回转体、各种准直器,控制节点采用细分步步进电机和数字交流伺服系统。为了优化设计,对控制功能进行分解,将该控制系统分为顶层控制子系统和底层控制子系统,如图 1 所示。

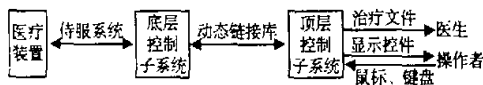


图 1 控制系统的组成

顶层控制子系统即人机界面,它是一个基于 Windows 的应用程序,该程序用 VC++ 编写,通过动态链接库与底层控制部分进行数据交换。

底层控制子系统采用 Lonworks 现场总线技术。控制网络采用总线型,用双绞线作为传输媒质,节点芯片为 3150 神经元芯片,节点为细分步步进电机,通信方式采用显示报文方式。结构简单、可靠性强,有很好的系统集成性,便于安装与维护。

### 2 顶层控制子系统的结构和功能

顶层控制子系统是一个基于 Windows 的应用程序,该应用程序的界面由两部分组成:菜单和主界面。菜单部分主要用于对设备进行调试和维护,主界面部分主要用于操作者对整个系统进行调控和监视。

菜单部分由文件操作、调试、参数设置、故障处理、帮助四部分组成,主界面部分由状态区、控制区、监视区组成。

顶层控制子系统的主要功能是充当操作者与底层控制子

系统之间的接口,进行数据显示和数据传递,从而实现操作者对治疗过程的监视与控制。

数据显示通过控制区的几组图形控件及监视区的几组数字显示控件来实现。根据从底层传来的数据,把回转体、治疗床和准直器的运动用图形形象的表现出来。数字显示控件则把这些数据通过数字准确地显示出来,反馈给操作者。

数据传递通过动态链接库,把治疗方案用命令加参数的方式,传给底层的 Lonworks 网络,对各个节点进行控制。通过函数调用实现控制与监视功能。

### 3 动态链接库的组成

动态链接库是连接顶层控制子系统和底层控制子系统的纽带,也是数据传递的关键,由四个函数和一个数据结构组成。

#### 3.1 动态链接库的函数原形及基本功能

动态链接库的四个函数如下所示:

```

int InitLonAdapter( );
void OnStopmotor( BYTE moterid, BYTE direction,
    BYTE maxfreq, UINT step);
void OnStartmotor( BYTE moterid);
BYTE CheckDataBuffer( LPBYTE revbuf);
  
```

函数 InitLonAdapter 用于初始化 Lonworks 适配卡,该函数返回整型值,0 为失败,1 为成功。函数 OnStartmotor 用于把数据下传给步进电机,参数 moterid 为电机编号,参数 direction 为电机运动方向,1 为正向,0 为反向;参数 maxfreq 为电机运动速度(电机脉冲的最大频率,以 100Hz 为单位);参数 step 为电机运行步数。函数 OnStopmotor 用于停止电机,参数 moterid 表示电机编号。函数 CheckDataBuffer 用于把数据上传给顶层控制系统,参数 revbuf 为存放数据的缓冲区。

#### 3.2 动态链接库的数据结构

动态链接库的数据结构是对从底层上传来的数据格式的约定。通过这种约定,顶层控制系统可从结构体的不同字节中获取每个节点的位置、速度、限位、故障等信息。

数据结构的定义如下:

```

typedef struct
{
  
```

收稿日期:2001-05-28

作者简介:朱华(1973-),男,湖南人,硕士研究生,主要研究方向:控制网络、软件工程;蔡淮(1942-),男,北京人,教授,主要研究方向:计算机控制、软件工程。

```

unsigned char Command;           //命令
unsigned char DestNode;         //目的节点号
unsigned char SourceNode;       //源节点号
unsigned char ID[6];            //源节点 ID 号
unsigned char DataLen;          //实际数据长度
unsigned char Reserve[4];       //保留区
unsigned char Data[16];         //数据区
: LON_DATA_STRUCT;

```

命令为 1 字节十六进制数,表示数据的目的是和作用,具体定义见表 1。

表 1

命令值	命令功能	说明
0x13	传位置信息	数据区为位置信息
0x14	传操作方式	数据区为操作方式信息,1 为手动,2 为程控
0x15	已到限位	数据区为限位信息,1 为最高限位,2 为最低限位
0x16	故障	数据区为各类故障的代码

目的节点号为 1 字节十六进制数,表示数据的接收对象,供接收方判断是否需要进行处理,0xFF 为广播地址,0x00 为主机节点号。

源节点和源节点 ID 号分别为 1 字节和 6 字节十六进制数,表示数据的来源,供接收方进行比较、判断和某些信息的更新(如节点硬件是否已更换等)。

数据长度为 1 字节十六进制数,表示本数据包中实际数据的长度(以字节为单位)。

数据区为 16 字节,用于存放实际数据。另外保留 4 字节的区域备用。

### 4 控制系统的实现

顶层与底层控制子系统以动态链接库为纽带,通过初始化、数据下传、数据上传三个过程来完成数据交换,实现控制功能。

#### 4.1 初始化过程

当系统首次运行或更换网络节点时应执行初始化过程。首先,应将节点数总和及各节点号、各节点 ID 值保存于 Windows 下的初始化文件 Lonmotor.ini 中。当系统开始运行时,初始化程序先将数据从 Lonmotor.ini 中读出,然后通过 InitLonAdapter 函数初始化 Lonworks 适配卡,该函数返回值为 1,表明初始化成功;返回值为 0,表示初始化失败。

Lonmotor.ini 中定义了系统的配置参数 number 与 IDm, number 表示系统中的节点数量, IDm 表示第 m 个节点的 ID 号,当 m=0 时表示主机节点。此系统中共有 1 个主机节点和 26 个分节点,故 number=27。

#### 4.2 数据下传

数据下传指把操作者输入的控制参数经主机节点,通过动态链接库函数送到 Lonworks 网络,最后由分节点完成具体的控制功能。

下传的参数主要为三个,分别为各节点细分步步进电机的转向、速度和步数(进给量)。实现下传的动态链接库函数为 OnStartmotor 与 OnStopmotor。

下面以治疗床的运动控制为例,来说明控制过程的实现。治疗床有 x、y、z 三个方向的运动,分别由三个细分步步

进电机控制,这三个电机所对应的节点号分别为 2、3、4。现在希望治疗床 x 向电机以 1000 步/秒的进给频率正向旋转 5000 步,y 向电机以 3000 步/秒的进给频率反向旋转 4000 步,z 向电机以 2000 步/秒的进给频率正向旋转 6000 步。则在程序中调用 OnStartmotor 三次,如下所示:

```

OnStartmotor(2,1,10,5000);
OnStartmotor(3,0,30,4000);
OnStartmotor(4,1,20,6000);

```

在治疗床各向运动中,随时可让运动停止,此时调用动态链接库函数 OnStopmotor。如在治疗床 x 向运动中,要想运动立即停止下来,只须执行如下语句:

```
OnStopmotor(2);
```

#### 4.3 数据上传

数据上传指把各节点的信息从 Lonworks 网络经动态链接库传给应用程序,并把把这些信息通过主界面上的图形控件和数字显示控件反馈给操作者。

上传的数据有各节点的位置、操作方式、限位、故障等信息。应用程序通过动态链接库函数 CheckDataBuffer 来获得这些信息,并从结构体的数据区提取这些信息。

在应用程序中设定一个定时器,定时执行 CheckDataBuffer 函数,该函数把各分节点的变化参数读到 revbuf 缓冲区。因为 CheckDataBuffer 每次最多返回 30 个数据包,而每个数据包的最大长度为 30 个字节(数据包定义如前所述的结构体),故缓冲区定义为 900 个字节,各节点数据包按节点顺序从小到大在 900 个字节中排列。然后从 900 个字节(30 个包中)提取具体的数据。仍以前面所举的治疗床的运动为例,在某个时刻,治疗床 x 向电机已运动了 4000 步,y 向电机已运动了 3000 步,z 向电机已运动了 5000 步,则缓冲区中的前三个数据包如表 2 所示。

表 2

1 号数据包	2 号数据包	3 号数据包
0x13 (1byte)	0x13 (1byte)	0x13 (1byte)
0x00 (1byte)	0x00 (1byte)	0x00 (1byte)
2 (1byte)	3 (1byte)	4 (1byte)
2 号节点 ID(6byte)	3 号节点 ID(6byte)	4 号节点 ID(6byte)
18 (1byte)	18 (1byte)	18 (1byte)
保留区 (4byte)	保留区 (4byte)	保留区 (4byte)
4000 (4byte)	3000 (4byte)	5000 (4byte)
剩余区 (12byte)	剩余区 (12byte)	剩余区 (12byte)

先从缓冲区中的第 1 个字节 0x13 得知数据区为位置信息,从第 3 个字节的值 2 中得知这是 2 号节点的数据,从第 10 个字节中得知此数据包有效长度为 18 个字节。根据数据结构的定义,实际数据从第 15 个字节开始,故从第 15~18 个字节中提取 2 号节点的位置信息 4000。在设计时,把每个数据包的长度固定为 30 个字节,有效长度由结构体的数据长度字节给出。故第 2 个数据包为第 31~60 字节,第三个数据包为第 61~90 字节,依此类推。

#### 参考文献

[1] 阳宪惠. 现场总线技术及其应用[M]. 北京:清华大学出版社, 1999.  
 [2] David J. Kruglinski, Scot Wingo, George Shepherd. Visual C++ 6.0 技术内幕[M]. 北京:北京希望电子出版社, 1999.