

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 13/38 (2006.01)



[12] 发明专利申请公布说明书

[21] 申请号 200610016557.X

[43] 公开日 2007 年 8 月 1 日

[11] 公开号 CN 101008930A

[22] 申请日 2006.1.24

[21] 申请号 200610016557.X

[71] 申请人 中国科学院长春光学精密机械与物理
研究所

地址 130031 吉林省长春市东南湖大路 16 号

[72] 发明人 吴能伟

[74] 专利代理机构 长春科宇专利代理有限责任公司
代理人 李恩庆

权利要求书 3 页 说明书 4 页 附图 2 页

[54] 发明名称

一种自动识别与主控计算机通讯设备的方法

[57] 摘要

本发明属于数据通讯技术领域，涉及一种自动识别与主控计算机通讯设备的方法，通过引入与多路串行通信卡串行端口相连设备的标识号，使主控计算机能够根据通讯数据携带的标识号快速识别与其进行数据通讯的设备，根据接收到的设备输出的自身运行参数和状态了解设备的运行情况，从而方便地进行设备测试和维修，对于有多个设备的自动控制系统，大大减小了测试和维修的难度。

1、一种自动识别与主控计算机通讯设备的方法，其特征在于采用下列步骤：

a、将多路串行通信卡安装在主控计算机的 PCI 总线扩展槽内；

b、在主控计算机上安装 Windows 2000 操作系统，参照多路串行通信卡的操作手册对其进行驱动，在事件管理器里查看该卡是否加载成功，若加载成功则可在主控计算机的设备管理器里查看该卡相应的信息；

c、在主控计算机的设备管理器里设置多路串行通信卡的各端口序号；

d、给主控计算机及将要与其相连的每台设备分配唯一的标识号；

e、确定主控计算机与每台设备间的通讯协议，依据选定协议为控制每台设备的微处理器引出信号线及插头与多路串行通信卡的各端口相连，使主控计算机与每台设备能按通讯协议进行通讯；

f、先为每台设备分配一个串行端口号，在主控计算机中保存与设备相连的端口号及其标识号；

g、在主控计算机与控制每台设备的微处理器中进行程序编制，实现二者按设定的通讯格式和波特率进行数据通讯；

所述主控程序包括下列步骤：

①、对多路串行通信卡的串行端口进行初始化设置：

②、对多路串行通信卡的各个端口进行扫描，接收带有设备标识码的数据信息；

③、判断接收数据的波特率是否为设置的波特率，是则转步骤④，否则转步骤⑥；

④、判断接收的数据信息所携带的标识码是否为已经分配给已知设备的标识码，是则接收的数据信息为已知设备的数据信息，转步骤⑤；否则接收的数据为未知设备的数据信息，转步骤⑥；

⑤、识别接收的数据信息属于哪一个已知设备；

⑥、判断是否结束对多路串行通信卡串行端口的扫描，是则转步骤⑦，否则转步骤③；

⑦、在所有串行端口号中剔除已检测出来的串行端口号，余下的串行端口号顺序分配给未开机的设备；

⑧、关闭所有串行端口。

2、根据权利要求1所述的自动识别与主控计算机通讯设备的方法，其特征在于多路串行通信卡（6）采用美国 MOXA 公司 C320Turbo/PCI 的八路高速串行通信卡，八路高速端口序号为 Com5~Com12，主控计算机（7）标识号为 0xd0，设备（1、2、3、4、5）标识号分别为 0xd1、0xd2、0xd3、0xd4、0xd5；主控计算机（7）与设备（1、2、3、4、5）的通信格式采用串行通信协议，波特率选择 9.6kbps 或 19.2kbps。

3、根据权利要求1所述的自动识别与主控计算机通讯设备的方法，其特征在于软件编程中，包含串行设备 MOXA 公司提供的 Pcomm.h 头文件，编译时链接串行设备 MOXA 公司提供的 Pcomm.lib 库文件；对各台设备的微处理器用 C 语言进行程序编制，主控计算机程序用 Visual C++语言编制；串行口初始化：应用 sio_open(int port)函数打开要访问的串行端口，用 sio_ioctl(int port, int baud, int mode)设置串行端口的串行通信协议，其中 nPort 为 5~12；定义串行端口接收与发送数据的缓冲区分别为 BYTE Asyn_RBuf[24]、Asyn_Sbuf[24]；主控计算机（7）用 sio_write(int port, char *buf, int len)

向设备（1~5）发送带有主控计算机标识码（0xd0）的数据信息，用 `sio_flush(int port, int func)` 清空串行端口数据；主控计算机（7）用 `sio_read(int port, char *buf, int len)` 循环接收带有设备（1、2、3、4、5）标识码 0xd1、0xd2、0xd3、0xd4、0xd5 的数据信息，根据标识码达到确定与串行端口相连设备的名称；如果设备（1、2、3、4、5）中的某个或几个未开机，则在所有串行端口号中剔除已检测出来的串行端口号，余下的串行端口号顺序分配给未开机的设备；在分配完所有的串行端口后，用 `sio_close(int port)` 关闭所有串行端口。

一种自动识别与主控计算机通讯设备的方法

技术领域

本发明属于数据通讯技术领域，涉及自动控制系统中的一种识别与主控计算机进行数据通讯分系统设备的方法。

背景技术

随着计算机与电子技术的快速发展和新型传感器的不断涌现与应用，采用单个计算机控制方式已越来越难以满足自动控制系统设备控制的要求。一方面，所控制的每个分系统设备既需要外部输入一些必要的信息，也需要向外部输出自身的运行参数和状态；另一方面，各个分系统设备间所需传输的控制信号也日益增加，所有这些，都要求我们采用分布式控制网络技术，将众多设备通过高速 RS-485/RS-422 串行端口与主控计算机有机地连成一体，以保证整个系统安全可靠地运行。如何从众多的 RS-485/RS-422 串行端口中自动识别与主控计算机进行数据通讯的分系统设备，传统方法是采用在串行端口处粘贴标签及记载串行端口说明的配套资料借助人的智能来完成，大大增加了控制系统的维修和测试难度。

发明内容

为了解决现有技术存在的自动控制系统维修和测试难度大的问题，本发明提供一种自动识别与主控计算机通讯设备的方法，通过引入与主控计算机相连设备的标识号，使主机能够快速识别与其进行通讯的设备，了解设备的运行情况，从而对设备及时、方便地进行测试和维修。

本发明特征在于采用下列步骤：

- a、将多路串行通信卡安装在主控计算机的 PCI 总线扩展槽内；
- b、在主控计算机上安装 Windows 2000 操作系统，参照多路串行通信卡的操作手册对其进行驱动，在事件管理器里查看该卡是否加载成功，若加载成功则可在主控计算机的设备管理器里查看该卡相应的信息；
- c、在主控计算机的设备管理器里设置多路串行通信卡的各端口序号；
- d、给主控计算机及将要与其相连的每台设备分配唯一的标识号；
- e、确定主控计算机与每台设备间的通讯协议，依据选定协议为控制每台设备的微处理器引出信号线及插头与多路串行通信卡的各端口相连，使主控计算机与每台设备能按通讯协议进行通讯；
- f、先为每台设备分配一个串行端口号，在主控计算机中保存与设备相连的端口号及其标识号；
- g、在主控计算机与控制每台设备的微处理器中进行程序编制，实现二者按设定的通讯格式和波特率进行数据通讯，主控计算机可根据通讯数据所携带的标识号来识别与串行端口相连的各台设备。

所述主控程序包括下列步骤：

- ①、对多路串行通信卡的串行端口进行初始化设置；
- ②、对多路串行通信卡的各个端口进行扫描，接收带有设备标识码的数据信息；
- ③、判断接收数据的波特率是否为设置的波特率，是则转步骤④，否则转步骤⑥；
- ④、判断接收的数据信息所携带的标识码是否为已经分配给已知设备的标识码，是则接收的数据信息为已知设备的数据信息，转步骤⑤；否则接收的数据为未知设备的数据信息，转步骤⑥；

⑤、识别接收的数据信息属于哪一个已知设备；

⑥、判断是否结束对多路串行通信卡串行端口的扫描，是则转步骤⑦，否则转步骤③；

⑦、在所有串行端口号中剔除已检测出来的串行端口号，余下的串行端口号顺序分配给未开机的设备；

⑧、关闭所有串行端口。

本发明的工作过程：打开主控计算机要访问的多路串行通信卡串行端口并进行扫描，循环接收带有设备标识码的数据信息。如果接收的数据信息的波特率是预先设置的主控计算机和设备通信波特率，则根据接收数据信息携带的识别码判断接收的数据信息属于哪一个设备。如果已知设备没有开机，则在所有串行端口号中剔除已检测出来的串行端口号，余下的串行端口号顺序分配给未开机的设备，在分配完所有的串行端口后关闭所有串行端口。

有益效果：本发明通过引入与多路串行通信卡串行端口相连设备的标识号，使主控计算机能够根据标识号快速识别与其进行数据通讯的设备，根据接收到的设备输出的自身运行参数和状态了解设备的运行情况，从而及时、方便地进行设备测试和维修，对于有多个设备的自动控制系统，大大减小了测试和维修的难度。

附图说明

图1为本发明硬件结构示意图。图中1、2、3、4、5为自动控制系统中与主控计算机通讯的已知设备，6多路串行通信卡，7主控计算机。

图2为本发明计算机主控程序流程图。

具体实施方式

如图1所示，本发明多路串行通信卡6采用美国MOXA公司C320Turbo/PCI

的八路高速串行通信卡，主控计算机 7 选用 PC 机；该多路串行通信卡 6 安装在主控计算机 7 的 PCI 总线扩展槽上。通过主控计算机 7 设置八路高速串行通信卡端口序号为 Com5~Com12，主控计算机 7 标识号为 0xd0，设备 1、2、3、4、5 标识号分别为 0xd1、0xd2、0xd3、0xd4、0xd5。主控计算机 7 与设备 1、2、3、4、5 的通信格式采用串行通信协议（设置起始位、停止位、数据位、奇偶校验等），对于少量数据非实时传输的设备波特率选择 9.6kbps，对于大量数据实时传输的设备选择波特率 19.2kbps。

软件编程中，首先包含串行设备 MOXA 公司提供的 Pcomm.h 头文件（以下调用的函数均在其中有说明），编译时链接串行设备 MOXA 公司提供的 Pcomm.lib 库文件。对各台设备的微处理器用 C 语言进行程序编制，主控计算机程序用 Visual C++ 语言编制。串行口初始化：应用 sio_open(int port) 函数打开要访问的串行端口，用 sio_ioctl(int port, int baud, int mode) 设置串行端口的串行通信协议（包括串行端口的 Baud 率、起始位、数据位、停止位、奇偶校验位），其中 nPort 为 5~12。定义串行端口接收与发送数据的缓冲区分别为 BYTE Asyn_RBuf[24]、Asyn_Sbuf[24]；主控计算机 7 用 sio_write(int port, char *buf, int len) 向设备（1~5）发送带有主控计算机标识码（0xd0）的数据信息，用 sio_flush(int port, int func) 清空串行端口数据；主控计算机 7 用 sio_read(int port, char *buf, int len) 循环接收带有设备 1、2、3、4、5 标识码 0xd1、0xd2、0xd3、0xd4、0xd5 的数据信息，根据标识码达到确定与串行端口相连设备的名称；如果设备 1、2、3、4、5 中的某个或几个未开机，则在所有串行端口号中剔除已检测出来的串行端口号，余下的串行端口号顺序分配给未开机的设备；在分配完所有的串行端口后，用 sio_close(int port) 关闭所有串行端口。

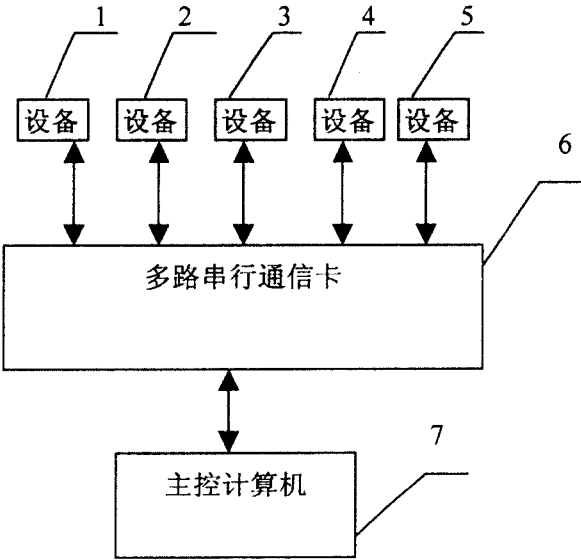


图 1

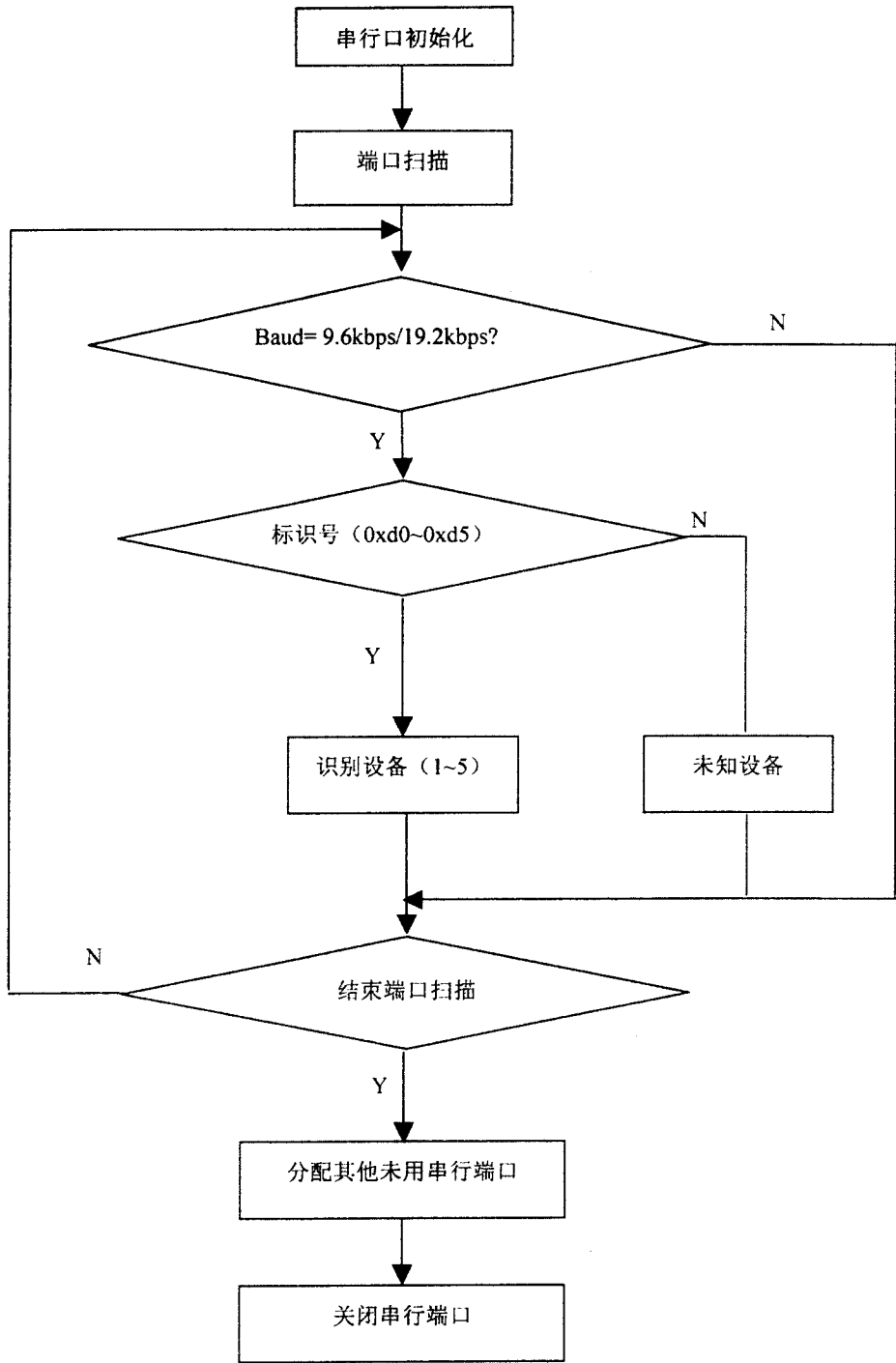


图 2