



(12) 发明专利申请

(10) 申请公布号 CN 105224362 A

(43) 申请公布日 2016. 01. 06

(21) 申请号 201510601043. X

(22) 申请日 2015. 09. 18

(71) 申请人 深圳市新产业生物医学工程股份有限公司

地址 518052 广东省深圳市坪山新区金沙社区金辉路 16 号

(72) 发明人 周刚 饶捷 尹力 陈晓涛 汤俊辉 邱小明 陈淑娟

(74) 专利代理机构 北京聿宏知识产权代理有限公司 11372

代理人 钟日红 张文娟

(51) Int. Cl.

G06F 9/445(2006. 01)

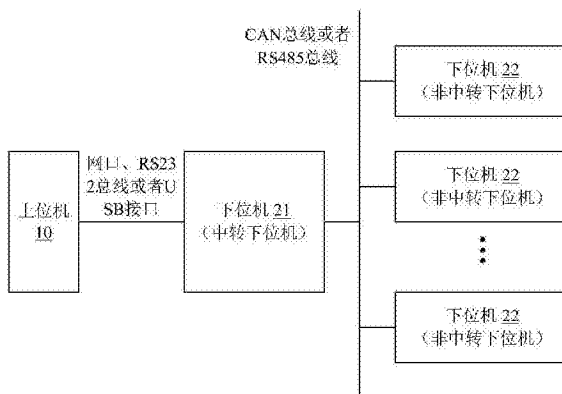
权利要求书2页 说明书8页 附图3页

(54) 发明名称

上位机对下位机进行程序升级的方法及系统

(57) 摘要

本发明公开了一种上位机对下位机进行程序升级的方法及系统,克服现有技术中在线升级效率较低以及成本较高等不足。该方法包括:上位机将升级程序打包成多个数据帧并通过第一通信网络发送给中转下位机;中转下位机按顺序接收多个数据帧,获取每个数据帧中的下位机标识信息;根据下位机标识信息判断出数据帧属于中转下位机的升级程序时,利用升级程序的所有数据帧进行本地的应用程序的升级;根据下位机标识信息判断出数据帧属于非中转下位机的升级程序时,将数据帧转发给非中转下位机;非中转下位机利用升级程序的所有数据帧进行本地的应用程序的升级。本发明提高了下位机程序升级的效率,显著降低了程序升级作业的工作量。



1. 一种上位机对下位机进行程序升级的方法,其中,所述下位机包括与所述上位机通过第一通信网络直接相连的中转下位机以及通过第二通信网络与所述中转下位机相连的非中转下位机,所述第一通信网络与所述第二通信网络不同;该方法包括:

所述上位机将升级程序打包成多个数据帧,通过所述第一通信网络将多个所述数据帧按顺序发送给中转下位机;

所述中转下位机按顺序接收多个所述数据帧,获取每个所述数据帧中的下位机标识信息;

根据所述下位机标识信息判断出所述数据帧属于所述中转下位机的升级程序时,利用所述升级程序的所有数据帧进行本地的应用程序的升级;

根据所述下位机标识信息判断出所述数据帧属于所述非中转下位机的升级程序时,将所述数据帧转发给所述非中转下位机;

所述非中转下位机利用所述升级程序的所有数据帧进行本地的应用程序的升级。

2. 根据权利要求 1 所述的方法,其中:

所述第一通信网络包括通过 RS232 总线、网口或者 USB 接口连接而成的网络,所述第二通信网络包括通过 CAN 总线或者 RS485 总线连接而成的网络。

3. 根据权利要求 1 所述的方法,其中:

所述上位机在向所述中转下位机发送每个所述数据帧之后,均向所述中转下位机查询数据帧的接收情况,并在所述中转下位机未成功接收时再次发送所述数据帧。

4. 根据权利要求 1 所述的方法,其中,所述中转下位机按顺序接收多个所述数据帧,包括:

所述中转下位机根据每个数据帧中的数据帧标识信息,按照所述数据帧标识信息的顺序接收多个所述数据帧。

5. 根据权利要求 1 所述的方法,其中:

所述中转下位机或者所述非中转下位机的主芯片中的程序存储区域包括 Bootloader 区域、原应用程序存储区和新应用程序存储区,所述 Bootloader 区域和原应用程序存储区均设置有各自的升级标志位;其中,所述原应用程序存储区用于存储升级之前的应用程序,所述新应用程序存储区用于存储升级之后的应用程序;

所述中转下位机或者所述非中转下位机的主芯片利用所述新应用程序存储区以及所述原应用存储区和所述 Bootloader 区域的升级标志位,完成升级程序的升级。

6. 根据权利要求 5 所述的方法,其中,所述中转下位机或者所述非中转下位机的主芯片利用所述新应用程序存储区以及所述原应用存储区的和所述 Bootloader 区域的升级标志位,完成升级程序的升级,包括:

所述中转下位机或者所述非中转下位机的主芯片将所述升级程序存储到所述新应用程序存储区,将所述原应用存储区的升级标志位设置为 1,将所述 Bootloader 区域的升级标志位设置为 1,并将所述应用程序的入口修改为所述新应用程序存储区的入口;在将所述新应用程序存储区中的内容复制到所述原应用程序存储区中后,将所述 Bootloader 区域的升级标志位清零。

7. 一种上位机对下位机进行程序升级的系统,其中,包括上位机、与所述上位机通过第一通信网络直接相连的中转下位机以及通过第二通信网络与所述中转下位机相连的非中

转下位机,所述第一通信网络与所述第二通信网络不同;该系统中:

所述上位机,用于将升级程序打包成多个数据帧,通过所述第一通信网络将多个所述数据帧按顺序发送给中转下位机;

所述中转下位机,用于按顺序接收多个所述数据帧,获取每个所述数据帧中的下位机标识信息;并用于根据所述下位机标识信息判断出所述数据帧属于所述中转下位机的升级程序时,利用所述升级程序的所有数据帧进行本地的应用程序的升级;还用于在根据所述下位机标识信息判断出所述数据帧属于所述非中转下位机的升级程序时,将所述数据帧转发给所述非中转下位机;

所述非中转下位机,用于利用所述升级程序的所有数据帧进行本地的应用程序的升级。

8. 根据权利要求 7 所述的系统,其中:

所述第一通信网络包括通过 RS232 总线、网口或者 USB 接口连接而成的网络,所述第二通信网络包括通过 CAN 总线或者 RS485 总线连接而成的网络。

9. 根据权利要求 7 所述的系统,其中:

所述上位机用于在向所述中转下位机发送每个所述数据帧之后,均向所述中转下位机查询数据帧的接收情况,并在所述中转下位机未成功接收时再次发送所述数据帧。

10. 根据权利要求 7 所述的系统,其中:

所述中转下位机用于根据每个数据帧中的数据帧标识信息,按照所述数据帧标识信息的顺序接收多个所述数据帧。

11. 根据权利要求 7 所述的系统,其中:

所述中转下位机或者所述非中转下位机的主芯片中的程序存储区域包括 Bootloader 区域、原应用程序存储区和新应用程序存储区,所述 Bootloader 区域和原应用程序存储区均设置有各自的升级标志位;其中,所述原应用程序存储区用于存储升级之前的应用程序,所述新应用程序存储区用于存储升级之后的应用程序;

所述中转下位机或者所述非中转下位机的主芯片用于利用所述新应用程序存储区以及所述原应用存储区的和所述 Bootloader 区域的升级标志位,完成升级程序的升级。

12. 根据权利要求 11 所述的系统,其中:

所述中转下位机或者所述非中转下位机的主芯片用于将所述升级程序存储到所述新应用程序存储区,将所述原应用存储区的升级标志位设置为 1,将所述 Bootloader 区域的升级标志位设置为 1,并将所述应用程序的入口修改为所述新应用程序存储区的入口;在将所述新应用程序存储区中的内容复制到所述原应用程序存储区中后,将所述 Bootloader 区域的升级标志位清零。

上位机对下位机进行程序升级的方法及系统

技术领域

[0001] 本发明涉及通信技术领域,具体地说,涉及一种上位机对下位机进行程序升级的方法及系统。

背景技术

[0002] 为了使控制系统的软件、数据等资讯信息处于最新状态,都需要实时或及时地对软件进行升级更新。但在对控制系统的多个模块的软件进行升级时,如果一个模块一个模块地逐步升级,则显然存在着升级速度慢、工作量大等缺陷,不仅会影响软件更新的及时性,也会降低仪器运行的速度。

[0003] 目前使用的仪器控制系统常见有嵌入式系统,其在线升级方式多为基于一种通信网络由上位机对所有下位机一一发送升级命令并进行升级,升级步骤繁琐,速度慢,不能满足及时更新的需求。

[0004] 中国专利申请 CN 104102508 A 公开了一种基于 FlexRay 总线的多机程序在线升级方法,其是通过一根 USB 转 FlexRay 调试器将 MCU 单元(下位机)与 PC 机(上位机)相连来进行多级程序的在线升级。

[0005] 但是,当上位机与直接相连的下位机之间的通信网络和所有不与上位机直接相连的下位机之间所采用的通信网络不同时,将导致与上位机无直接相连的下位机不能随上位机发送的升级指令直接进行在线升级,因此,在通信方式日益多元化的环境下极大限制了上位机对下位机的升级方式。

[0006] 生化免疫仪器是基于生化免疫分析法的一种医疗诊断设备。目前的生化免疫仪器的控制系统包括电路控制部分,其主要由多块主控板组成。要实现仪器的正常工作和升级,通常是通过上位机单独对每一块主控板(下位机)进行程序烧写。

[0007] 现有生化免疫仪器的上位机与主控板之间在线升级方法为:首先将所有要升级的代码存放在上位机中;然后将生化免疫仪器的外壳拆开,通过串口线连接上位机和主控板;然后在上位机上操作升级软件,以推荐标准(Recommended Standard,简称 RS)-232 的通信方式将代码写入主控板中,完成一块主控板的升级,而需要升级的主控板均要重复上述操作。

[0008] 综合以上所述,现有技术存在着以下的不足:

[0009] 1) 基于一种通信网络的程序升级,需要对每一主控板(下位机)一一进行升级,升级步骤复杂且速度慢,难以满足实时更新的需求;

[0010] 2) 通信网络单一,不能满足目前通信方式日益多元化的应用环境和发展趋势;

[0011] 3) 针对网络单一的生化免疫仪器的程序升级,当前还需要拆卸生化免疫仪器的外壳并在升级完毕之后再安装好外壳,耗时耗力且浪费时间,效率低下。

发明内容

[0012] 本发明所要解决的技术问题是克服现有技术中上位机与直接相连的下位机所使

用的通信网路和不与上位机直接相连的各下位机之间所使用的通信网络不同时,对整个系统的在线升级效率较低以及成本较高等不足。

[0013] 本发明首先提供了一种上位机对下位机进行程序升级的方法,其中,所述下位机包括与所述上位机通过第一通信网络直接相连的中转下位机以及通过第二通信网络与所述中转下位机相连的非中转下位机,所述第一通信网络与所述第二通信网络不同;该方法包括:所述上位机将升级程序打包成多个数据帧,通过所述第一通信网络将多个所述数据帧按顺序发送给中转下位机;所述中转下位机按顺序接收多个所述数据帧,获取每个所述数据帧中的下位机标识信息;根据所述下位机标识信息判断出所述数据帧属于所述中转下位机的升级程序时,利用所述升级程序的所有数据帧进行本地的应用程序的升级;根据所述下位机标识信息判断出所述数据帧属于所述非中转下位机的升级程序时,将所述数据帧转发给所述非中转下位机;所述非中转下位机利用所述升级程序的所有数据帧进行本地的应用程序的升级。

[0014] 优选地,所述第一通信网络包括通过 RS232 总线、网口或者 USB 接口连接而成的网络,所述第二通信网络包括通过 CAN 总线或者 RS485 总线连接而成的网络。

[0015] 优选地,所述上位机在向所述中转下位机发送每个所述数据帧之后,均向所述中转下位机查询数据帧的接收情况,并在所述中转下位机未成功接收时再次发送所述数据帧。

[0016] 优选地,所述中转下位机按顺序接收多个所述数据帧,包括:所述中转下位机根据每个数据帧中的数据帧标识信息,按照所述数据帧标识信息的顺序接收多个所述数据帧。

[0017] 优选地,所述中转下位机或者所述非中转下位机的主芯片中的程序存储区域包括 Bootloader 区域、原应用程序存储区和新应用程序存储区,所述 Bootloader 区域和原应用程序存储区均设置有各自的升级标志位;其中,所述原应用程序存储区用于存储升级之前的应用程序,所述新应用程序存储区用于存储升级之后的应用程序;所述中转下位机或者所述非中转下位机的主芯片利用所述新应用程序存储区以及所述原应用存储区和所述 Bootloader 区域的升级标志位,完成升级程序的升级。

[0018] 优选地,所述中转下位机或者所述非中转下位机的主芯片利用所述新应用程序存储区以及所述原应用存储区的和所述 Bootloader 区域的升级标志位,完成升级程序的升级,包括:所述中转下位机或者所述非中转下位机的主芯片将所述升级程序存储到所述新应用程序存储区,将所述原应用存储区的升级标志位设置为 1,将所述 Bootloader 区域的升级标志位设置为 1,并将所述应用程序的入口修改为所述新应用程序存储区的入口;在将所述新应用程序存储区中的内容复制到所述原应用程序存储区中后,将所述 Bootloader 区域的升级标志位清零。

[0019] 本发明还提供了一种上位机对下位机进行程序升级的系统,其中,包括上位机、与所述上位机通过第一通信网络直接相连的中转下位机以及通过第二通信网络与所述中转下位机相连的非中转下位机,所述第一通信网络与所述第二通信网络不同;该系统中:所述上位机,用于将升级程序打包成多个数据帧,通过所述第一通信网络将多个所述数据帧按顺序发送给中转下位机;所述中转下位机,用于按顺序接收多个所述数据帧,获取每个所述数据帧中的下位机标识信息;并用于根据所述下位机标识信息判断出所述数据帧属于所述中转下位机的升级程序时,利用所述升级程序的所有数据帧进行本地的应用程序的升

级；还用于在根据所述下位机标识信息判断出所述数据帧属于所述非中转下位机的升级程序时，将所述数据帧转发给所述非中转下位机；所述非中转下位机，用于利用所述升级程序的所有数据帧进行本地的应用程序的升级。

[0020] 优选地，所述第一通信网络包括通过 RS232 总线、网口或者 USB 接口连接而成的网络，所述第二通信网络包括通过 CAN 总线或者 RS485 总线连接而成的网络。

[0021] 优选地，所述上位机用于在向所述中转下位机发送每个所述数据帧之后，均向所述中转下位机查询数据帧的接收情况，并在所述中转下位机未成功接收时再次发送所述数据帧。

[0022] 优选地，所述中转下位机用于根据每个数据帧中的数据帧标识信息，按照所述数据帧标识信息的顺序接收多个所述数据帧。

[0023] 优选地，所述中转下位机或者所述非中转下位机的主芯片中的程序存储区域包括 Bootloader 区域、原应用程序存储区和新应用程序存储区，所述 Bootloader 区域和原应用程序存储区均设置有各自的升级标志位；其中，所述原应用程序存储区用于存储升级之前的应用程序，所述新应用程序存储区用于存储升级之后的应用程序；所述中转下位机或者所述非中转下位机的主芯片用于利用所述新应用程序存储区以及所述原应用存储区的和所述 Bootloader 区域的升级标志位，完成升级程序的升级。

[0024] 优选地，所述中转下位机或者所述非中转下位机的主芯片用于将所述升级程序存储到所述新应用程序存储区，将所述原应用存储区的升级标志位设置为 1，将所述 Bootloader 区域的升级标志位设置为 1，并将所述应用程序的入口修改为所述新应用程序存储区的入口；在将所述新应用程序存储区中的内容复制到所述原应用程序存储区中后，将所述 Bootloader 区域的升级标志位清零。

[0025] 与现有技术相比，本发明对于各下位机之间的通信网络与上位机和其中一个下位机之间的通信网络不同时，仅通过上位机的操作就可以对所有下位机进行升级，从而提高了下位机程序升级的效率，显著降低了程序升级作业的工作量。

[0026] 本发明的其它特征和优点将在随后的说明书中阐述，并且，部分地从说明书中变得显而易见，或者通过实施本发明而了解。本发明的目的和其他优点可通过在说明书、权利要求书以及附图中所特别指出的结构来实现和获得。

附图说明

[0027] 为了更清楚地说明本发明实施例中的技术方案，下面将对实施例描述中所需要的附图做简单的介绍：

[0028] 图 1 是本发明实施例的上位机对下位机进行程序升级的系统的结构示意图。

[0029] 图 2 是本发明实施例的上位机向中转下位机发送升级程序的方法的流程示意图。

[0030] 图 3 是本发明实施例的中转下位机的程序升级的方法的流程示意图。

[0031] 图 4 是本发明实施例的非中转下位机的程序升级的方法的流程示意图。

[0032] 图 5 是本发明实施例的下位机在运行应用程序时的流程示意图。

具体实施方式

[0033] 以下将结合附图及实施例来详细说明本发明的实施方式，借此对本发明如何应用

技术手段来解决技术问题,并达成技术效果的实现过程能充分理解并据以实施。需要说明的是,只要不构成冲突,本发明中的各个实施例以及各实施例中的各个特征可以相互结合,所形成的技术方案均在本发明的保护范围之内。

[0034] 本发明实施例的上位机对下位机进行程序升级的方法,适用于上位机对多台下位机进行程序升级。如图 1 所示,本发明实施例的上位机 10 对下位机进行程序升级的系统示意图中,该上位机 10 与该多台下位机中的其中一台通过第一通信网络直接相连,所有下位机之间通过第二通信网络连接在一起。该上位机 10 和与其直接相连的下位机 21 之间采用该第一通信网络进行通信,下位机之间均通过该第二通信网络进行通信。其中,该第一通信网络和第二通信网络并不相同。

[0035] 优选地,上位机 10 与其中一台下位机 21 采用的第一通信网络为 PC 组网时常用的利用 RS232 总线、网口或者 USB 接口组成的网络,下位机之间采用的通信网络为利用 CAN 总线或者 RS485 总线组成的网络。可以理解,第一通信网络和第二通信网络的种类并不限于上述所列举的几种网络,实际运用中可以根据需求合理选择不同的通信网络,在基于本发明思想下的两种通信网络的变化,也应属于本发明的专利保护范围。

[0036] 具体地,在需要为与上位机 10 直接相连的下位机 21 进行程序升级时,上位机 10 通过网口、RS232 总线或者 USB 接口等通信方式将升级程序发送到该下位机 21 中,由该下位机 21 进行程序升级。在需要为不与上位机 10 直接相连的下位机 22 进行程序升级时,上位机 10 先通过网口、RS232 总线或者 USB 接口的通信方式将升级程序发送到与该上位机 10 直接相连的下位机 21 中,再由与上位机 10 直接相连的该下位机 21 通过 CAN 总线或者 RS485 总线通信方式将升级程序发送到不与上位机 10 直接相连的下位机 22 中,由最终接收到该升级程序的下位机(不与上位机 10 直接相连的下位机 22)进行升级。

[0037] 为清晰起见,在下文中,将与上位机 10 直接相连的下位机 21 称之为中转下位机,将不与上位机 10 直接相连的下位机 22 称之为非中转下位机。如果没有特别说明,下文中的下位机通常指的是所有下位机,既包括中转下位机,也包括非中转下位机。

[0038] 在本发明的实施例中,无论是给哪个下位机进行升级,上位机 10 均先将升级程序打包成多个数据帧,然后通过网口、RS232 总线或者 USB 接口发送给中转下位机。这些数据帧中包含有下位机标识信息,数据帧标识信息,升级程序的数据内容,升级程序的指示位、结束位以及校验位等。在本发明的实施例中,下位机标识信息可以是下位机的编号。数据帧标识信息可以是数据帧的编号。

[0039] 上位机 10 通过升级程序的指示位来对有关升级程序的数据帧进行标识。下位机(包括中转下位机和非中转下位机)收到数据帧之后,通过指示位来识别该数据帧是否为有关升级程序的。下位机在接收有关升级程序的数据帧时,通过数据帧标识信息来判断数据帧的接收是否按照预定顺序进行的,并通过结束位来确认有关升级程序的数据帧是否完整接收。校验位用于下位机对接收到的数据帧进行校验,判断接收到的数据帧是否有错。

[0040] 对于中转下位机,根据所接收到的数据帧中的下位机标识信息判断出该数据帧是用来对自己的程序进行升级,则在本地利用所接收到的数据帧进行程序升级。如果根据所接收到的下位机标识信息判断出该数据帧是用来对非中转下位机的程序进行升级,则根据下位机标识信息将所接收到的数据帧通过第二通信网络发送给与下位机标识信息相对应的下位机。

[0041] 如果中转下位机与非中转下位机之间的第二通信网络出现通信出错、程序升级成功或升级失败等,中转下位机都会对上位机 10 的相应查询进行针对性的应答。中转下位机作为两种通信网络的通信中转枢纽,对上位机 10 与非中转下位机之间的通信内容进行转发,并与上位机 10 进行直接通信。

[0042] 本发明的实施例中,采用下位机编号来作为下位机的标识信息。并且,下位机编号可以固定在主控芯片程序中,也可以是确定在下位机中的硬件设置中。比如,不同的下位机,将拨档开关的拨码位置设置成不同,则各自的编号就不同。当然,也可以采用同时在主控芯片程序中固定结合在硬件设置中确定的方法。无论采用哪种编号确定方式,每个下位机的编号必须各不相同,具有唯一性,以相互区分。

[0043] 基于嵌入式芯片上电时,总是先从程序存储区域中的 Bootloader 区域开始执行,然后才进入到应用程序存储区域。而芯片升级程序时,一般升级的是应用程序。本发明的技术方案,通过对应用程序内的升级标志位和 Bootloader 区域内的升级标志位 B 的判断来改变应用程序入口的方式,令下位机在完整收到升级程序后,可实现不断电的升级,并且为升级中出现断电的意外提供了可恢复的保障。

[0044] 本发明的实施例中,将下位机的主芯片中的程序存储区域划分为三个部分,分别为 Bootloader 区域、原应用程序存储区和新应用程序存储区。芯片上电后开始运行的位置为 Bootloader 区域,这部分具有可以指定应用程序入口的功能。原应用程序存储区存储升级前的应用程序。新应用程序存储区存储接收所有数据帧后,重新整合好新的升级程序。中转下位机或者非中转下位机的主芯片,利用该新应用程序存储区以及该原应用存储区和所述 Bootloader 区域的升级标志位,即可完成升级程序的升级处理。

[0045] 其中 Bootloader 区域和原应用程序存储区均设置有各自的升级标志位。当主芯片将升级程序存储到新应用程序存储区后,将原应用存储区的升级标志位 A 设置为 1,此时主芯片强制复位进入 Bootloader 区域,将 Bootloader 区域中的升级标志位 B 设置为 1,将应用程序的入口修改为新应用程序存储区的入口,并将新应用程序存储区的内容复制到原应用程序存储区中。然后,将 Bootloader 区域中的升级标志位 B 清零,完成应用程序的升级。在下次上电时,判断到 Bootloader 区域中的升级标志位 B 为 0,则执行的应用程序入口是原应用程序存储区的入口,此时的原应用程序存储区中存储的是应用程序已经更新之后的版本。

[0046] 本发明的实施例中,断电恢复机制的原理为:若主芯片在程序复制过程中出现断电的意外,在重新上电时,会根据 Bootloader 区域中被设置为 1 的升级标志位 B 来重新进行复制动作,继续进行应用程序的升级,直到复制完成,Bootloader 区域中的升级标志位清零,才完整地实现了整个升级过程。

[0047] 如图 2 所示,本发明实施例的上位机向中转下位机发送升级程序的方法流程中,主要包括如下步骤。

[0048] 步骤 S210,将存储在上位机中待发送的升级程序,根据上位机与中转下位机选择的通信网络不同,打包成对应网络可识别的多个数据帧。这里通信网络可以是 PC 机常用的 RS232 总线、网口或者 USB 接口等形成的网络。不同网络采用的数据帧格式不同,但各数据帧的内容除了包含升级程序的数据内容,还包含有下位机标识信息,数据帧标识信息,升级程序指示位、结束位和校验位等。其中下位机标识信息采用下位机编号,用于识别升级程序

所对应的下位机。数据帧标识信息采用数据帧编号,用于鉴别数据帧在发送过程中是否发送中是否有丢包。升级程序的指示位用于标识该数据帧的内容是否为升级程序,升级程序的结束位用于向下位机指示是否已经完整发送有关该升级程序的所有数据帧,升级程序的校验位用于下位机对所有接收到的数据帧进行校验,以判断所有接收到的数据帧内容是否有误。

[0049] 步骤 S220,上位机按照为数据帧预设的发送顺序,将这些数据帧发送给中转下位机。该预设的发送顺序,是这些数据帧的数据帧标识信息;中转下位机接收时,也按照数据帧标识信息的顺序来接收这些数据帧。每次发完一个数据帧后,就向中转下位机发送查询请求以查询当前所放松的这个数据帧的接收情况。收到下位机已经成功收到当前发送的数据帧的应答后,上位机再继续发送下一个数据帧。本发明的实施例中,为数据帧预设的发送顺序,采用的是数据帧编号。中转下位机接收时,也按照数据帧编号的顺序来接收。如果上位机在发送一个数据帧 Z 之后,向下位机查询是否成功接收该数据帧 Z 时,超过预定时间仍然没有收到下位机的应答,则重新发送刚刚已经发送过的数据帧 Z,直到收到下位机反馈已经成功收到该数据帧 Z 的应答后,再继续发送该数据帧 Z 的下一个数据帧,直至所有数据帧成功地发送到相应的下位机中。在这个过程中,对某一个数据帧,如果重发次数超过预设的数量阈值,则进行报警,向用户提示对某一个数据帧已经重发达到预设次数但可能仍然未成功发送给相应的下位机。前述的预定时间以及预设的数量阈值,根据具体的通信环境、通信速度以及下位机的数量等进行确定。

[0050] 步骤 S230,上位机发送完升级程序的所有数据帧之后,向对应的下位机查询升级状态,并为下位机的升级时间进行计时。上位机判断下位机的升级状态依据是下位机的升级标志位,当下位机的升级标志位为 1 时,表示下位机正在升级中,若升级标志位已清零则表示下位机已完成升级。

[0051] 升级正常完成后,上位机将升级计时清零,本次升级成功。

[0052] 如果超过预设的升级时间,上位机仍未收到下位机升级完成的状态,则生成报警信息进行报警,提示本次升级失败。此处预设的升级时间,主要是根据具体的通信环境、通信速度以及下位机的数量等进行确定。

[0053] 图 3 为本发明实施例的中转下位机的程序升级的流程示意图。

[0054] 步骤 S310,中转下位机接收由上位机发送下来的升级程序的所有数据帧。

[0055] 中转下位机每接收一个数据帧,就对数据帧编号进行对比,判断本次接收的数据帧的数据帧编号是否等于上一次接收的数据帧的数据帧编号 +1,是则合法,并针对上位机的查询反馈接收成功的应答。上位机收到该接收成功的应答后就继续发送下一个数据帧。

[0056] 如果中转下位机在进行数据帧编号的比对时,发现本次接收的数据帧的数据帧编号并不等于上一次接收的数据帧的数据帧编号 +1,此时说明出现了通信错误。针对上位机的查询反馈接收出错的应答。上位机接收到接收出错的应答后,产生报警信息进行报警。

[0057] 中转下位机根据升级程序的结束位判断出从上位机接收到升级程序的最后一个数据帧时,对升级程序的所有数据帧中数据进行校验,利用校验计算所得到的校验值与升级程序的校验位进行对比。

[0058] 若校验值与升级程序的校验位一致,则说明升级程序的数据帧完全正确地接收了。若校验值与升级程序的校验位不一致,则表示升级程序的数据帧接收过程中出现了错

误,此时主动向上位机上报接收出错的报告或者根据上位机的查询向上位机反馈接收出错的应答,上位机收到该接收出错的应答后,产生报警信息进行报警。本发明的实施例中,校验算法可以是奇偶校验或者CRC(循环冗余)校验等。

[0059] 步骤 S320,中转下位机根据所接收到的数据帧中所携带的下位机编号,判断升级程序是否为自身的升级程序。若该升级程序为中转下位机自身的升级程序,则转步骤 S330,若不是中转下位机自身的升级程序,则该升级程序为非中转下位机的升级程序,转步骤 S350。

[0060] 转步骤 S330,中转下位机将升级程序从数据帧中整合出来存储到本地的新应用程序存储区,将升级标志位设置为 1。

[0061] 步骤 S340,中转下位机通过对自身主控芯片中 Bootloader 区域的设计,实现自身的程序升级,并反馈升级状态给上位机。

[0062] 步骤 S350,通过下位机之间的通信网络 CAN 总线或者 RS485 总线,依次将各数据帧转发给对应的非中转下位机,然后转步骤 S360。

[0063] 中转下位机向非中转下位机转发数据帧,还请参考上位机向中转下位机发送数据帧的内容。各非中转下位机都拥有自己的标识信息(本实施例中将下位机的编号作为下位机的标识信息),以确保各下位机之间均能有效区分。

[0064] 步骤 S360,中转下位机监测接收了各数据帧的非中转下位机的升级状态,并将升级成功与否的升级结果反馈给上位机。

[0065] 对于接收中转下位机转发的数据帧的非中转下位机,在接收到自身的升级程序后,通过对自身主控芯片中的 Bootloader 区域,实现自身的程序升级,具体升级处理过程请参考下文图 4 所示的实施例,并将升级程序的标志清零,对上位机反馈下位机升级成功的应答。

[0066] 如果非中转下位机升级失败,则中转下位机将非中转下位机升级失败的应答信息反馈给上位机,上位机生成报警信息并进行报警。如果中转下位机查询到非中转下位机成功升级,则完成升级,对上位机反馈非中转下位机升级成功的应答。

[0067] 图 4 为本发明实施例的非中转下位机的程序升级的流程示意图。

[0068] 步骤 S410,基于 CAN 或者 RS485 的通信网络,接收中转下位机发下来的数据帧,判断数据帧所携带的下位机编号与自身的编号是否一致。

[0069] 若所接收的数据帧所携带的下位机编号与自身的编号一致,则向中转下位机反馈数据帧接收成功的应答,并根据数据帧的升级标志位设置为 1。非中转下位机从中转下位机接收升级程序的数据帧的接收过程,请参考中转下位机从上位机接收升级程序的数据帧的接收过程。对于接收正常(或者说接收成功)或者接收异常(或者说接收失败)的处理,是非中转下位机通过中转下位机向上位机进行应答。

[0070] 步骤 S420,非中转下位机从中转下位机处接收完升级程序的所有数据帧后,将升级程序从数据帧中整合出来,存储到本地的新应用程序存储区,将升级标志位设置为 1。通过对自身主控芯片中的 Bootloader 区域的设计,实现自身的程序升级。

[0071] 步骤 S430,非中转下位机在完成本地的应用程序升级后,将升级程序标志清零,并通过中转下位机向上位机反馈本地的应用程序升级成功的应答。

[0072] 图 5 是本发明实施例的下位机在运行应用程序时的流程示意图。

[0073] 步骤 S510, 在执行应用程序时, 不间断地对应用程序中的升级标志位进行监控。当监控到升级标志位为 0 时, 不进行任何升级操作。当待升级的下位机已将升级程序存储到新应用程序存储区后, 监测到升级标志位被设置为 1 时, 应用程序跳转到芯片的 Bootloader 处。

[0074] 步骤 S520, 下位机在 Bootloader 处将存储在 Bootloader 内的升级标志位 B 设置为 1, 并且修改应用程序的入口为新应用程序存储区的入口。

[0075] 步骤 S530, 将新应用程序存储区中已经升级的应用程序复制到原应用程序存储区, 复制完成后将 Bootloader 内的升级标志位 B 清零。

[0076] 步骤 S540, 下位机跳转到新的应用程序入口执行升级后的应用程序。

[0077] 在执行完上述处理后, 由于升级后的应用程序中的升级标志位默认为 0, 所以, 上位机通过中转下位机查询时, 查询到的下位机的升级状态为完成, 从而完成整个应用程序的升级过程。

[0078] 本发明的技术方案中, 存在着如下的断电恢复机制: 当下位机上电时, 正常情况下主芯片都会从 Bootloader 区域开始执行。Bootloader 内的程序根据升级标志位 B 选择是否修改应用程序入口。若 Bootloader 中的应用程序是正常的升级之后的应用程序, 则其内部的升级标志位 B 为 0, Bootloader 选择原应用程序存储区作为应用程序的入口, 此时原应用程序存储区中的应用程序即为升级之后的应用程序, 故下位机正常运行升级之后的应用程序即可。若主芯片在程序复制过程中出现了断电, 在重新上电时, 由于应用程序的升级因中途断电而未完成, Bootloader 中的升级标志位仍为 1 状态, 所以芯片会自动继续剩下的升级过程, 直到应用程序完整地升级完成。

[0079] 本发明的上位机对下位机进行程序升级的方法实施例以及上位机对下位机进行程序升级的系统实施例, 均可以广泛应用于上位机与多个下位机组网的领域, 比如电解质分析仪、生化分析仪、化学发光测定仪等, 只要上位机与直接相连的中转下位机所使用的通信网络和不直接与上位机相连但与该中转下位机相连的各下位机 (包括中转下位机) 之间所采用的通信网络不同, 均可应用本发明的技术方案。

[0080] 本发明的实施例在两种通信网络并存的情况下, 由直接发送或者转发的方式令下位机完整接收到上位机发出的升级程序, 实现了不同通信方式下的跨越式升级。本发明的技术方案依靠对下位机主控芯片中 Bootloader 区域的设计, 实现了下位机程序更新和断电自恢复更新, 程序升级过程中出现断电的意外情, 也能够保证再次上电后能继续进行程序更新。

[0081] 虽然本发明所公开的实施方式如上, 但所述的内容只是为了便于理解本发明而采用的实施方式, 并非用以限定本发明。任何本发明所属技术领域的技术人员, 在不脱离本发明所公开的精神和范围的前提下, 可以在实施的形式上及细节上作任何的修改与变化, 但本发明的专利保护范围, 仍须以所附的权利要求书所界定的范围为准。

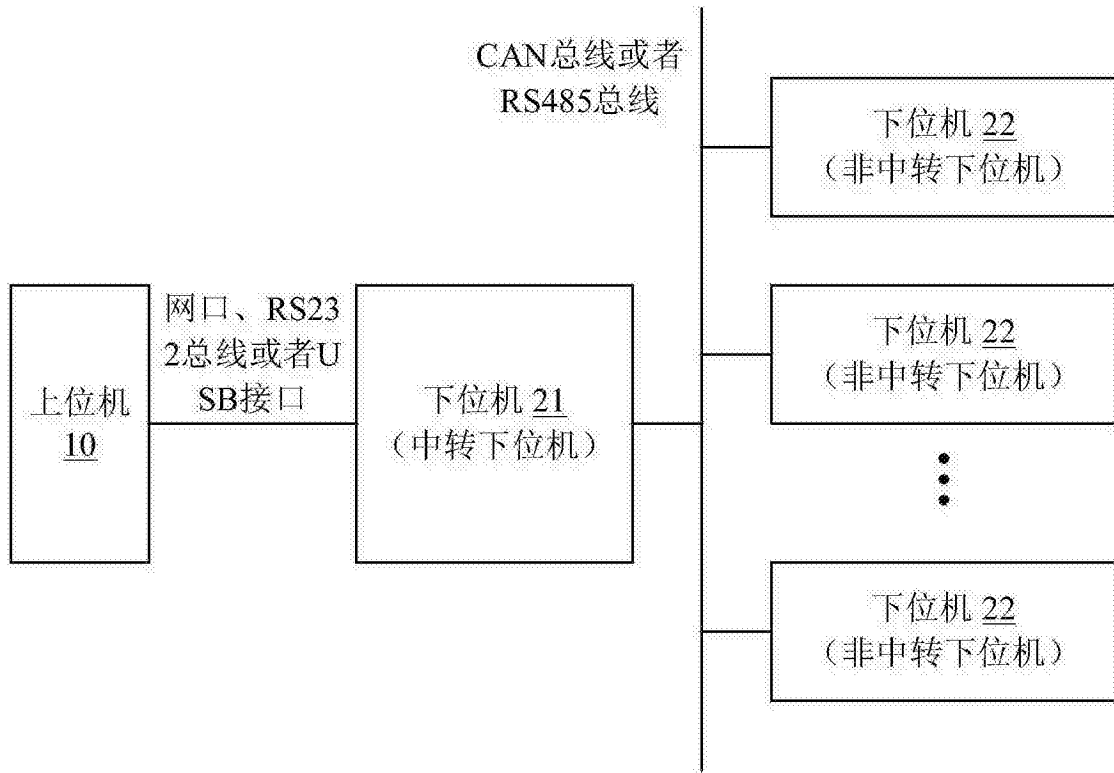


图 1

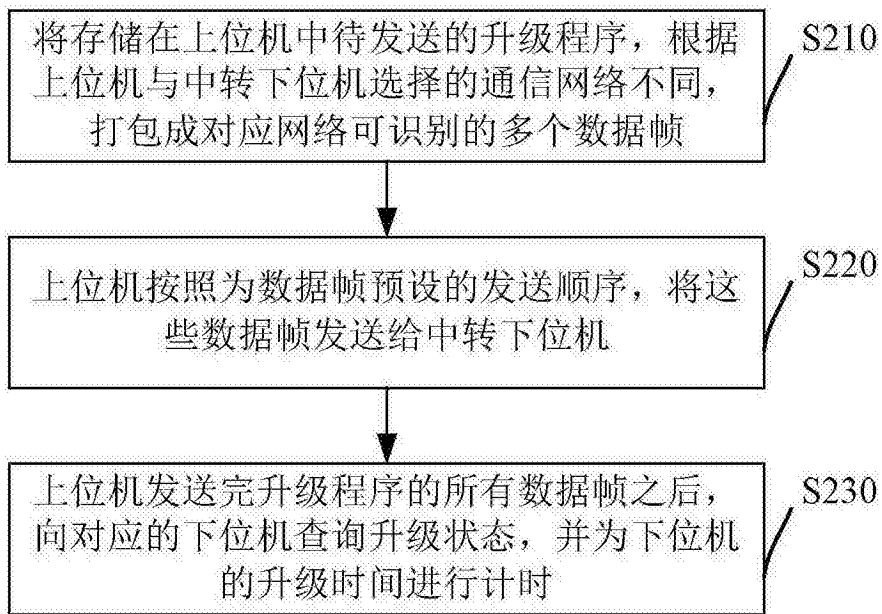


图 2

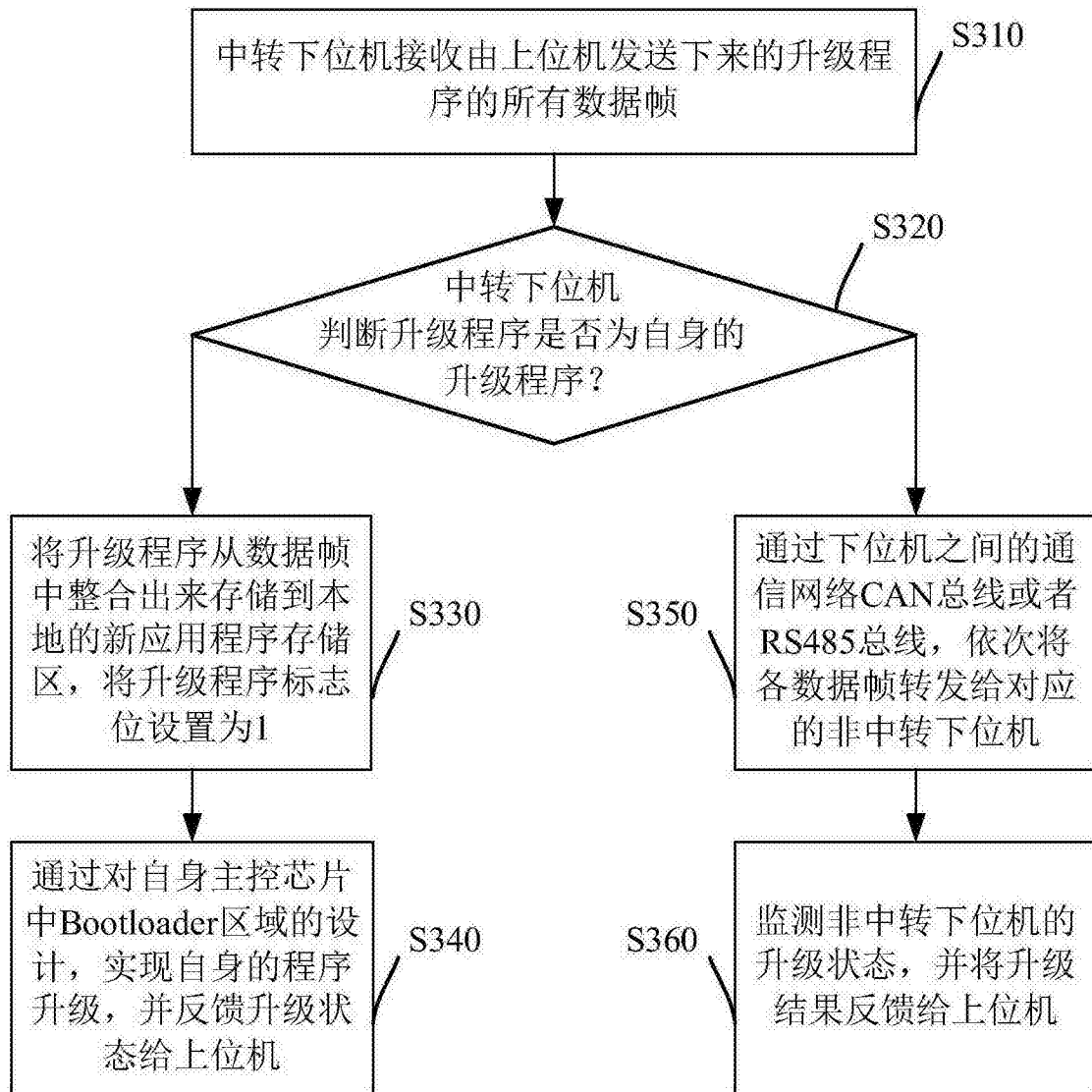


图 3

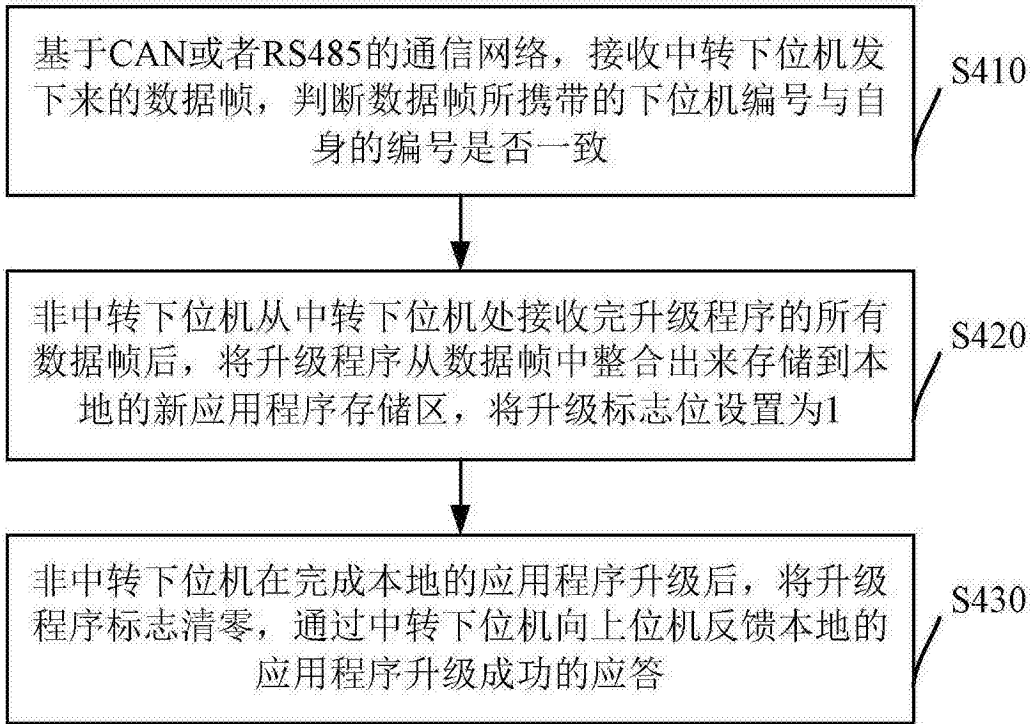


图 4

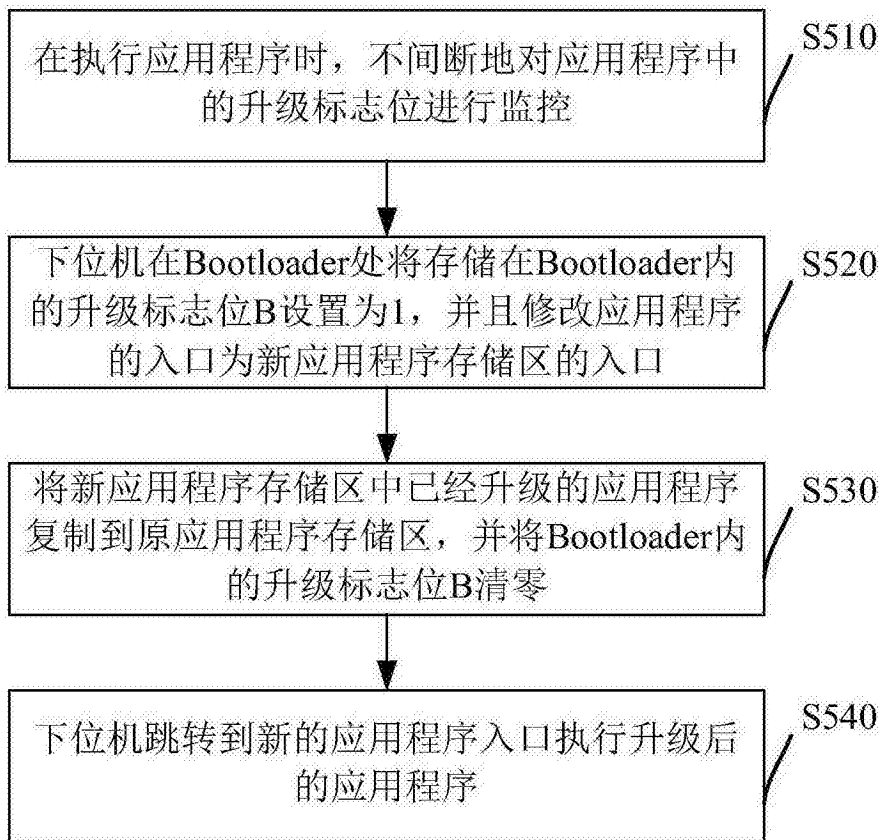


图 5