

Design and Implementation of Robot Distributed Control Communication Based on UART

Pengxin Shi, Xiaocan Su, Qingzhen Wang, Jie Shao

Robot and Microsystem Laboratory, Department of Computer Science, Zhengzhou College of Science & Technology, Zhengzhou Henan
Email: sjsyxx@163.com

Received: Sep. 26th, 2018; accepted: Oct. 12th, 2018; published: Oct. 19th, 2018

Abstract

In order to realize the coordination problem of the robot application system and realize the communication between multiple control modules, a distributed control communication protocol based on UART is designed and implemented. On the hardware, the RS-485 type electrical interface is used to analyze the information model of the control network. The real-time information scheduling algorithm of the token ring control field bus is designed. The communication protocol uses token ring principle to schedule each control module in real time, so as to realize the distributed field control based on UART.

Keywords

UART Communication, RS-485 Interface, Information Scheduling Algorithm, Token Ring Principle

基于UART的机器人分布式控制通信设计与实现

师鹏鑫, 苏小灿, 王清珍, 邵杰

郑州科技学院, 信息工程学院, 机器人与微系统实验室, 河南 郑州
Email: sjsyxx@163.com

收稿日期: 2018年9月26日; 录用日期: 2018年10月12日; 发布日期: 2018年10月19日

摘要

为了实现机器人应用系统的协调问题,实现多个控制模块之间的通信,现设计并实现了一种基于UART的机器人分布式控制通信协议。硬件上采用RS-485型电气接口,分析了控制网络的信息模型,实现了令牌环控制现场总线的实时信息调度算法设计,通信协议采用令牌环原理调度每一个控制模块,从而实现基于UART的无主机分布式现场控制。

关键词

UART通信, RS-485接口, 信息调度算法, 令牌环原理

Copyright © 2018 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

机器人智能化程度越来越高,那么对机器人控制应用复杂性也相应需要提高。每丰富一个功能,就需添加一个由微控制器连接相应的传感器、执行器等组成的模块来实现。机器人的功能越丰富,机器人控制系统的微控制模块就越多,因此以UART组成的现场总线网络[1]为基础的分布式控制系统在机器人控制中取得了广泛的应用。通过UART总线实现系统中多个控制模块的互联,它们之间进行信息交换,然而这些信息的传送有时间要求,在此现场总线网络中对信息传送实时性有关键性影响的是控制模块访问控制协议,即信息调度算法。另外,由于每一个控制模块的CPU字长不同,通信方式有多种,为了能够实现通信,需要采用一种通用的通信方式,而每一个微控制器都有的通信方式是UART。但仅靠微控制器自带电平的UART接口传输距离较近,且只能实现主机、从机之间的通信,不能实现任意两个设备之间的通信,不完全符合现场总线网络控制模块之间的要求。

为实现现场总线网络功能,需要设计一套通信协议,实现机器人分布式控制通信。硬件上采用RS-485型电气接口,提高通信距离,用于机器人复杂应用的控制模块之间通信。

2. 硬件结构设计

由于机器人功能的丰富,在机器人内部嵌入的控制模块就会越多,且这些控制模块之间需要协调一致的工作,在机器人的这个可编程控制系统中,模块与模块之间实现无主机的多机通信[2],每一个模块本身在系统中具有双重功能,既是一个主控模块,又是从控模块。模块与模块之间采用RS-485总线通信,由MAX487担任串口的电平转换器,MAX487是一种RS-485标准接口的电平转换芯片。RS-485采用差分半双工通信方式,真正实现多点总线连接,具有传输距离远可靠性高的特点。

每个控制模块是采用字长不同、种类不同的微控制器构成的嵌入式控制模块,模块与模块之间通过串口进行信息通信,系统的总线硬件结构如图1所示,每个控制模块通过电平转换芯片MAX487连接在RS-485总线上,模块与模块之间借助总线进行数据传输,数据传输稳定,方便扩展。各个模块采用一字节地址编码,对应的编码范围为0~255,因此接收总线的控制模块最多可以达到255个,满足机器人应用系统的需求,灵活性比较大。

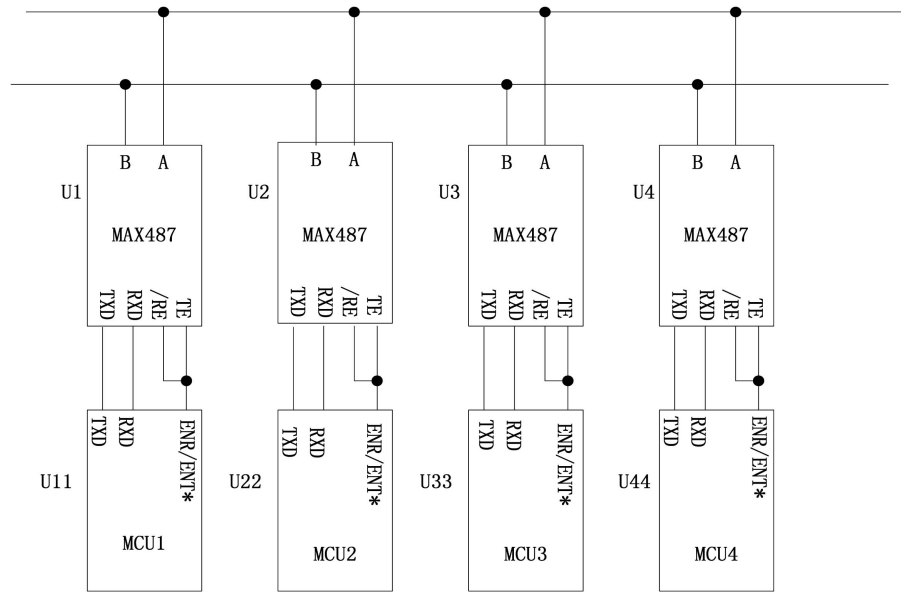


Figure 1. Bus hardware structure
图 1. 总线硬件结构

3. 令牌环[3]控制现场总线的实时信息调度算法

3.1. 控制网络的信息模型

以串行通信总线控制系统为基础的机器人控制系统的结构如图 2 所示，图 2 中的模块包括语音控制模块、灰度与避障等环境信息采集模块、上肢控制模块、定位与导航控制模块、下肢控制模块等，为了协调系统中各个模块工作，本应用系统各个模块之间进行数据的通信，构成此数据交换系统。在本数据交换系统中，模块之间进行交换的有对时间要求比较高的实时信息和对时间要求不高的非实时信息两种。在本应用系统中的实时信息又有两种：非周期性和周期信息，对非周期性信息进行转化，按照信息连续两次产生的时间间隔大于等于某个临界值时变换为周期信息，因此本应用系统只考虑周期性实时信息。考虑机器人控制系统中一个控制器上的所有控制回路的采样周期一般相同，故此采用信息任务驱动机制，将控制系统中一个模块上的周期性信息看作一个信息任务。设控制系统中模块的个数为 n ，由此可得本数据交换系统中实时信息模型为：

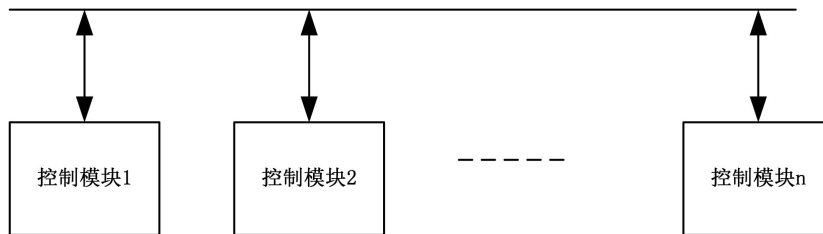


Figure 2. Communication structure chart of control system
图 2. 控制系统的通信结构图

$$M_{pi} = \langle C_{pi}, T_{pi}, D_{pi} \rangle, \quad i = 1, 2, 3, \dots, n \tag{1}$$

式中 M_{pi} 为第 i 个模块传送的实时信息， C_{pi}, T_{pi}, D_{pi} 分别为信息通信时间、通信周期和通信时限。

非实时信息是随机产生的,对于模块 i 是随机产生的。对于模块 i , 假设信息产生符合泊松分布过程, 则其模型为:

$$M_{pi} = \langle C_{pi}, \lambda_{pi} \rangle, i = 1, 2, \dots, n \quad (2)$$

式中, M_{pi} 为第 i 个模块传送的非实时信息, C_{pi} 和 λ_{pi} 分别表示非实时信息 M_{pi} 的临界值(即最大通信时间)和平均达到率, 则有 $T_{pi} = 1/\lambda_{pi}$ 那么平均到达间隔时间表示, 即通信周期 T_{pi} 通过平均达到率的倒数求出。特此说明: 1) 数据的发送时间是指在数据信息在发送过程未被中断的情况下, 从发送模块开始发送计时起, 到接收模块完全接受所发送数据信息为止, 所用的时间。通信时间由三个时间段: 数据信息的发送时间、接收模块的应答时间和令牌的传送时间组成。2) 模块之间交换的信息会有很多种, 但不是每个控制模块拥有所有种类的信息, 另外也不是每一种信息存在于任何一个控制模块上, 设置相应的常量和变量进行存储。

对实时信息是否能被调度的判断条件是在它们的传输时限前传送完成, 否则称为信息不可调度; 对非实时信息是否能被调度的判断条件是在一定时间(即临界时限)内调度完成, 否则称为信息不可调度。

3.2. 信息的可调度算法

信息调度是确定各控制模块上信息的传送时刻。在由多个模块构成的机器人应用系统中信息调度应采用分布式调度算法, 但随着微控制器技术的发展, 各控制模块的资源越来越充分, 可以为每个控制模块设计一个调度器, 采用单处理器实时任务调度算法。根据信息模型, 系统中存在非实时和实时两种信息, 采用最早截止时间优先算法 EDF 或单调速率调度算法 RMS 不适用, 故设计了双优先级队列调度算法[4]用于机器人控制系统中信息的调度, 算法如下:

Step 1: 把到达后的所有实例信息按实时信息和非实时信息组织在两个队列中存放, 等待执行, 然后根据信息不同采用不同的调度算法, 即采用非抢占式单调速率调度算法 RMS 调度实时信息; 采用先进先出 FIFO 算法调度非实时信息。

Step 2: 对实时信息和非实时信息的调度有先后次序。系统首先调度的是实时信息, 只有在实时信息队列中没有信息等待传送时, 才转到非实时信息队列传送非实时信息, 否则要等到实时信息都传送完毕后再转到非实时信息队列传送非实时信息。

Step 3: 在非实时信息处理过程中又把实时信息是否可以抢占非实时信息的调度分为两种: 1) 抢占算法: 实时信息可抢占非实时信息的调度; 2) 非抢占算法: 实时信息不抢占非实时信息的调度。

Step 4: 在把信息实例都按实时信息对待的情况下, 进行判断, 如果一个实时任务的实例未能在其时限(或临界值内)前完成, 则它被处理器丢弃。

4. 通信协议软件设计

模块之间的信息交换采用令牌环原理, 组织成实时任务的形式调度每一个控制模块传送的信息。挂载在总线上的每一个控制模块, 都有唯一的一字节地址码, 并存储有一字节的下位控制模块地址码。总线上的控制模块之间是通过无主机的形式进行通信, 规定拥有令牌标志的控制模块处于主控状态, 才可以向需要传送数据的目标控制模块发送数据, 完毕之后将令牌标志传送给下位控制模块。总线上没有令牌标志的控制模块常态下处于从控状态, 只可以接收。此规定的实现可以保证在总线上的所有控制模块中, 任一时刻只有一个控制模块可以作为主控制模块, 具有数据发送权; 而且可以使每一个控制模块都有向目标控制模块发送数据的机会, 因为令牌标志是在控制模块之间循环的, 从而让每一个控制模块都有数据发送权。

进行信息交换的两个控制模块之间所传送的信息有指令和数据两种情况，二者都按字节进行传输，51 控制器通过所传输的是指令则第 9 位为 1，是数据则第 9 位为 0 进行区分，双字长和四字长的控制器是指令则低字的高字节为非零，是数据低字的高字节为零进行区分，通过传输的数据是否为令牌标志作为主机与从机的区别。表 1、表 2 给出了传输指令和令牌标志位的内容。

Table 1. Control word and answer word allocation table

表 1. 控制字与应答字分配表

控制字	1~8 位	第九位或高字节	应答字	1~8 位	第九位或高字节
寻址指令	0xxxxxxx	1 或非零	令牌接收应答	10110000	1 或非零
数据传送起始指令	10100001	1 或非零	地址匹配应答	10110001	1 或非零
数据传送终止指令	10100010	1 或非零	接收准备完毕应答	10110010	1 或非零
授令牌指令	10101111	1 或非零	数据接收完毕应答	10110011	1 或非零

Table 2. Sign bit faith resolution table

表 2. 标志位信义解析表

标志位
第九位数据或高字节数据 1 或非零: 指令 0: 数据
数据接收允许位 1: 仅指令接收 0: 数据、指令接收
地址匹配标志 addr_matching 1: 有效
数据待发标志 data_tx_ready 1: 有效
数据待收标志 data_rx_ready 1: 有效
485 控制引脚 etr485 1: 发送状态 0: 接收状态
令牌标志 token 1: 有效

信息传输过程中需要设置一系列的变量和常数。每个控制模块通信实现的软件包括从控模式和主控模式两个控制流程。

4.1. 从控模式控制流程

从控模式下控制模块处于接收状态，可以通过 UART 中断运行程序，对接收到的数据做出一系列判断。接收到的信息有指令和数据两种情况，要么是指令，要么是数据。51 控制器通过所传输的是指令则第 9 位为 1，是数据则第 9 位为 0 进行区分；双字长和四字长的控制器是指令则低字的高字节为非零，是数据低字的高字节为零进行区分。通过传输的数据是否拥有令牌标志作为主机模式与从机模式的区别。接收到的信息若为指令，通过识别判断指令的内容，结合相应标志位做出相应的操作，其程序控制流程图如图 3 所示。

进入 UART 中断服务程序执行，首先判断是否为接收中断，不是发送中断请求，ERT485 清 0，转为接收状态，若是进行信息接收；其次对接收到的信息进行判定是指令还是数据，若是寻址指令，进行地址匹配判定，并置位地址匹配标志位，置位的取值是在初始状态下，对于数据接收的控制模块来说，若是从控模式控制模块接收到寻址指令；若接收到的是数据开始发送指令，且地址已经匹配，则将主从模块标志位清 0，初始化数据接收缓存区，做好数据接收准备；若接收到数据传送结束指令，则将主从模块标志位置 1，接收缓冲区初始化，为下一次数据接收作准备，发送接收准备完毕应答，同时将接收地址匹配标志位复位，退出接收状态。

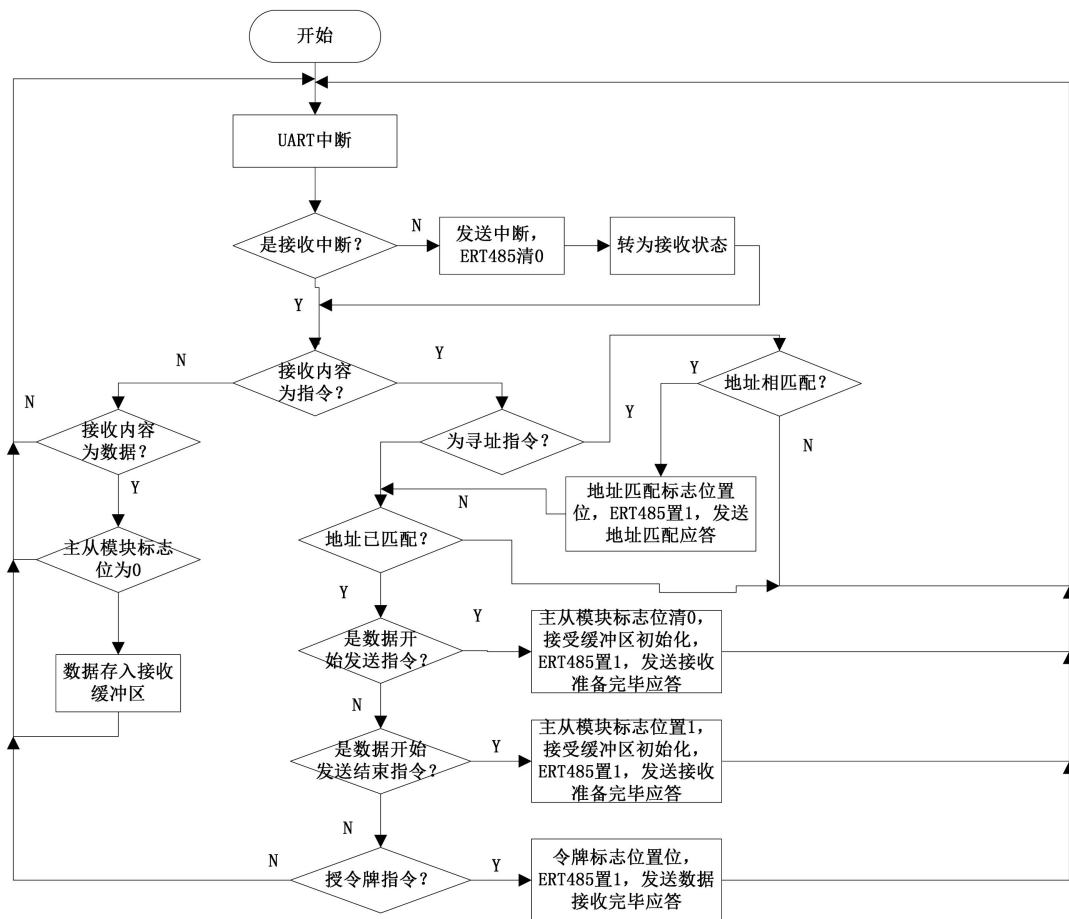


Figure 3. Flowchart of the slave mode program
图 3. 从控模式程序流程图

令牌标志位置位的实现是在 UART 中断程序执行过程中, 首先为接收中断, 且接收内容为指令, 其次进行寻址指令的判定和地址匹配的情况下, 然后判断不是数据开始发送指令也不是数据开始发送结束指令, 而是授令牌指令时, 才置位令牌标志位, 并发送数据接收完毕应答, 最后开始运行主控模式下的数据发送操作。若接收的内容是数据, 且数据允许接收位主从模块标志位为 0, 则将数据存入接收缓存区。

在主从控制模块之间进行指令信息的交换, 是从控模式控制模块每接收到一次有效指令, 都要向主控模式控制模块做出相应应答, 每一个控制模块都有不同的地址码, 保证总线上每次只有唯一的从控模式控制模块, 可以作出应答。在接收和应答相切换时要转换 MAX487 电平转换接口的收发状态。每一次接收都由中断触发, 发送应答后关闭中断, 可以节省 CPU 资源。

4.2. 主控模式控制流程

当控制模块接收到有效的授令牌指令之后, 转为主控模式, 并运行相关主控模块控制程序段, 程序控制流程图如图 4 所示。若主控控制模块有数据需要向目标控制模块发送, 则按主控模式程序控制流程图所示过程执行, 依次发送目标控制模块地址寻址指令, 发送数据开始传送指令, 连续发送数据, 最后发送数据传送完毕指令; 若主控控制模块没有数据需要发送或数据发送完毕, 便需要将令牌标志传给下位机, 依次发送下位机地址寻址指令和授令牌指令, 退出主控模式。

知网检索的两种方式：

1. 打开知网页面 <http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD>
下拉列表框选择：[ISSN]，输入期刊 ISSN：2326-3415，即可查询
2. 打开知网首页 <http://cnki.net/>
左侧“国际文献总库”进入，输入文章标题，即可查询

投稿请点击：<http://www.hanspub.org/Submission.aspx>

期刊邮箱：airr@hanspub.org