

# 基于 CoOS 和 CAN 总线的多级联电机控制系统<sup>①</sup>

李鹏飞, 张红梅, 张向利, 黄 剑

(桂林电子科技大学 信息与通信学院, 桂林 541004)

**摘 要:** 设计了一种基于 CooCox CoOS 微操作系统和 CAN 总线的高实时性、高精度的多电机控制系统。以恩智浦公司 LPC1114 芯片为处理器, 通过移植 CoOS 操作系统, 实现多任务运行, 有效的解决了多电机协调工作的问题; 利用 CAN 总线转发命令, 解决了多个控制器级联控制的问题, 扩大了控制范围。采用 VS2008 的 MFC 设计上位机控制软件, 实现了整个控制系统的可视化。应用结果表明, 该系统运行稳定可靠, 实时性好, 具有一定的实用和推广价值。

**关键词:** 步进电机; CooCox CoOS; CAN 总线; 多级联控制; MFC

## Multi-cascade Motor Control System Based on CoOS and CAN Bus

LI Peng-Fei, ZHANG Hong-Mei, ZHANG Xiang-Li, HUANG Jian

(School of Information and Communication Engineering, Guilin University of Electronic Technology, Guilin 541004, China)

**Abstract:** In the paper, a high real-time, high-precision multi-motor control system based on micro operating system and CAN bus is introduced. The system is proposed based on NXP LPC1114 micro-processors. The transplanting CoOS operating system allows the execution of the program in order to run the task, which effectively solves the multi-motor synchronous coordination problems. By forwarding commands by CAN bus, the problem of cascading multiple controllers work is solved. It also expands the control range. The VS2008 MFC PC control software design allows to achieve the control of a graphical display. The application results show that the system is stable and reliable, real-time, it also has some practical and promotional value.

**Key words:** step motor; CooCox CoOS; CAN Bus; multiple cascade control; MFC

运动控制在工业生产中的作用举足轻重, 多电机协调运动控制更是尤为重要。现如今, 多电机协调运动控制在工业领域中的应用越来越广泛, 控制精度与实时性的要求也越来越高。只有保证多个电机之间按某种关系同步协调运转, 才能确保生产的正常运行; 系统同步性能的好坏, 直接影响到产品的品质和性能。因此, 深入研究多电机的协调运动控制, 提高其同步精度具有重要的意义<sup>[1]</sup>。

CooCox CoOS 是一款针对 ARM Cortex-M 系列芯片而设计的实时系统内核。具有高度可裁剪性, 支持软件定时器和无中断延时时间。CAN 总线是一种有效的串行通信网络, 可靠性高, 实时性好, 抗干扰能力强<sup>[2]</sup>。CAN 总线的通信距离长, 速度快, 是一种价格

低廉, 适用于多种工业环境下进行通信的现场总线<sup>[3]</sup>。基于这两种技术的优点, 我们结合实际应用中多任务、高可靠性的需求, 设计出一种基于 CoOS 操作系统和 CAN 总线的多级联电机控制系统。根据实际应用场景, 系统可广泛应用在温室通风口控制、监控摄像头角度调节等电机控制领域。

## 1 系统总体设计方案

本系统以恩智浦公司的 LPC1114 微处理器为核心, 采取多控制器协同控制的办法, 控制器间通过 CAN 总线进行通信。上位机用 MFC 开发的控制软件, 通过串口与下位机主板相连接。主板可串联多个从板, 以 CAN 总线连接成直线型(或者接为环形), 主从板上

<sup>①</sup>基金项目:国家自然科学基金(61363031);广西科技开发项目(桂科攻 12118017-2C);桂林市科技开发项目(20120104-13)

收稿时间:2013-12-01;收到修改稿时间:2014-01-17

均可挂接两个 28BYJ-48 电机. 如图 1 所示, 整个系统主要包括上位机控制台、UART 传输线、下位机控制器、CAN 总线、步进电机等. 上位机通过串口发控制命令, 主控制器负责接收、转发、执行上位机命令, 从控制器通过 CAN 总线负责命令转发、执行命令. 各个板中分别移植 CoOS 操作系统, 有效提高可靠性和实时控制的灵活性.

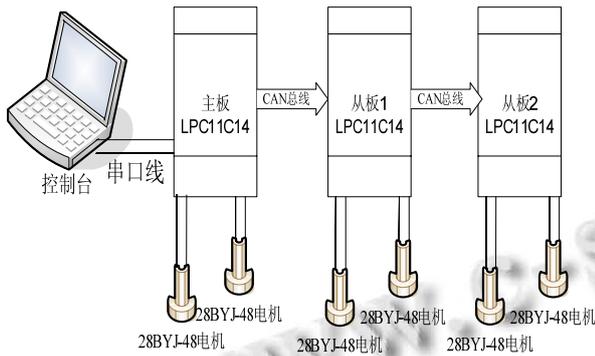


图 1 多级联电机控制系统简图

## 2 嵌入式硬件平台

### 2.1 LPC11C14 微处理器

本系统采用 NXP 公司的 LPC11C14 作为 CAN 通信和电机控制的控制器. LPC11C14 是一款基于 ARM Cortex-M0 的处理器, 具有丰富的外设组件: 32KB 片内 Flash 程序存储器, 8KB 片内 SRAM, 1 个 UART 串行接口, 1 个 CAN 控制器等. 并且提供了初始化和通信的 API 函数<sup>[4]</sup>.

### 2.2 UART 与 CAN 通信的硬件实现

UART 是上位机与下位机通信的接口, 通过接收模块 U0RX 监听串行输入线的有效输入<sup>[4]</sup>. UART 接收移位寄存器通过 RXD 接收上位机应用程序产生的控制命令. 主板负责命令的解析、执行与转发.

CAN 总线通信以及多级联电机控制电路如图 2 所示. CAN 节点是分布在 CAN 网络中进行相互通信的基本单元, 主要由主控制器、CAN 控制器和 CAN 收发器组成.

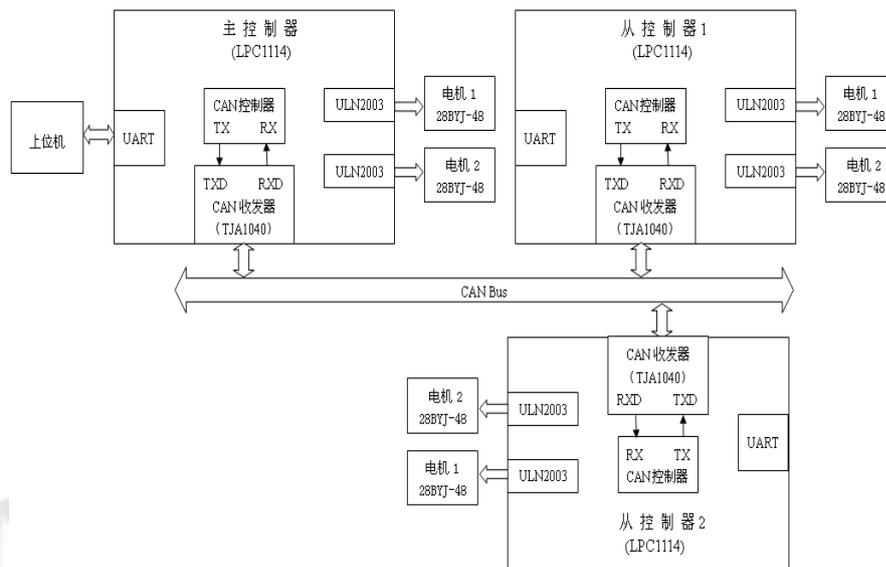


图 2 CAN 总线通信以及多级联电机控制电路

CAN 收发器采用周立功公司生产的 TJA1040 芯片, 它是 CAN 控制器和物理总线之间的接口. CAN 控制器通过一条串行数据输出线(TX)和一条串行数据输入线(RD)分别连接到收发器的发送数据输入端(TXD)和接收数据输出端(RXD). 主板通过 CAN 控制器把从板的命令发送到 CAN 收发器. 而收发器通过 CANH 和 CANL 总线终端向总线提供差动发送功能和

向 CAN 控制器提供差动接收功能<sup>[5]</sup>. 有效地将主板接收到的从板的命令进行转发与接收.

### 2.3 步进电机

步进电机是一种将电脉冲转化为角位移的执行机构. 电机采用 28BYJ-48 步进电机, 是一种四相八拍电机, 工作电压为 DC5V-DC12V. 由于其驱动电压高且功率较大. 故采用 ULN2003 驱动芯片, 以获得大电流

输出能力,从而驱动 28BYJ-48 步进电机。

### 3 嵌入式软件设计

CooCox CoOS 是一款针对 ARM Cortex-M 系列芯片而设计的实时系统内核,提供了信号量、互斥体和事件标志实现任务间的同步<sup>[6]</sup>。本设计引入操作系统的目的是让程序的执行以任务的方式运行,这样可以有效的解决多任务并行问题,在任务的切换过程中减少延时等待。利用 IAR 集成开发环境,在将 CoOS 操作系统成功移植到 LPC1114 处理器的基础上,对 CoOS 进行软件扩充,主要包括编写步进电机驱动程序, CAN 通信程序以及任务划分等。根据系统设计的要求,主要建立了 4 个任务模块:消息处理任务,电机 1 任务,电机 2 任务以及 CAN 通信任务。其总体结构图如图 3 所示。

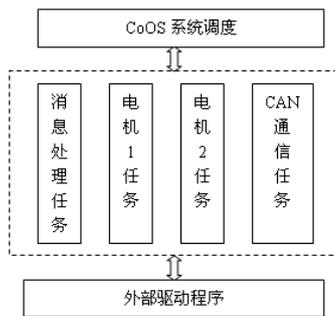


图 3 系统软件结构框图

#### 3.1 消息处理

主控制板主要是接收来自上位机的命令,不需要通过 CAN 转发,而直接自己执行。系统采用中断的方式,上位机发送命令至串口 buff,由下位机循环检测 buff,当有数据进入,并检测到“板子编号”位是 0x01,则在本机进行消息处理,即解析命令。

命令的解析主要是把上位机传递的命令字节按协议进行移位分离,并设置步进电机的参数变量,根据解析出来的值设置步进电机的参数变量,例如:

```
controlNum =(cmdRev&(0x1F<<23))>> 23;//板子编号
motorNum =(cmdRev &(0x1F <<18))>>18;//电机编号
motorDir =(cmdRev &(0x1F <<13))>>13;//解析方向
motorAngle =(cmdRev &(0xFF <<5))>>5;//解析角度
motorSpeed =(cmdRev &(0x03 <<3))>>3;//解析速度
```

由于电机的任务是同步协调执行的,故需要使用事件标志。当等待任务所等待的事件标志处于未就绪

状态时,这些任务处于不可调度状态。但一旦等待对象变成就绪状态,任务将很快恢复运行。因此,当解析命令时,发现是本机的命令,且电机需要启动,就要把相应的电机的标志通过 CoSetFlag()函数触发标志。例如通过 CoSetFlag(g\_MTIStart)触发电机 1 的标志,从而电机 1 的任务从未就绪态转为就绪态,等待系统调度。从控制器的命令解析类似,不同之处是检测到是 0x02 或 0x03 时,通过 CAN 控制器,转发到 1 号或 2 号板,由从板进一步解析。

#### 3.2 CAN 通信任务

当主板检测到不是本机程序时,则需要通过 CAN 转发命令。通信协议遵循 ISO 国际标准化的串行通信协议。需定义一个结构体 CAN\_buff[]数组对象,用来存放需要转发的命令及相关信息,比如数据长度及具体的命令协议。

```
CAN_buff[1].id = controlNum; //板子的编号
CAN_buff[1].dlc = 0x04; // 数据长度为 4 个字节
CAN_buff[1].data[0] = cmdctrl & (0xFFFF);
CAN_buff[1].data[1] = (cmdctrl&(0xFFFF <<16)) >>16;
// 4 字节的命令协议
CoEnterMutexSection(CANSendMutex);//进入互斥区
CAN_Send( 10, FALSE, (uint32_t *)&CAN_buff[1]);
CoLeaveMutexSection(CANSendMutex);//离开互斥区
```

在对调用 CAN 发送和接收函数,都要访问同一个“临界区”,会造成程序执行错误。CoOS 提供互斥区域解决这种的问题。互斥区域禁止多个任务同时进入受保护的代码“临界区”。因此,在任意时刻,只能有一个任务进入这样的代码保护区<sup>[4]</sup>。可以调用 CoCreateMutex()来创建一个互斥区域,调用 CoEnterMutexSection()和 CoLeaveMutexSection()进入和离开互斥区域,以实现临界区代码的保护。

从机的 CAN 中断程序在检测到 buff 中有数据写入时,从中开始读取数据即可:

```
CANRevBuff.id = Buff->id; //读板子编号
CANRevBuff.dlc = Buff->dlc; //读字节长度
CANRevBuff.data[0] = Buff->data[0];//读命令数据
CANRevBuff.data[1] = Buff->data[1];
```

命令解析过程与上述的消息处理过程类似。

#### 3.3 电机任务

为了能够达到两个电机同步协调工作的功能,采用 CoCreateTask()函数创建两个电机任务,并且任务

的优先级设置为同级. 对于同一优先级就绪任务, CoOS 遵循“先入先出(FIFO)”的原则, 将迟来的任务排在同一优先级的最后. 同级任务间的调度, 则根据其自身任务的时间片进行轮转调度, 系统执行完当前任务的时间片长度, 就将控制权交给下一个同一优先级任务. 在创建任务之后, 首先调用 CoCreateFlag(0,0)函数创建一个事件标志 g\_MT1Start, 把任务设置为未就绪状态, 随后执行 CoWaitForSingleFlag(g\_MT1Start,0)等待事件标志被触发. 事件标志是在解析命令是调用 CoSetFlag(g\_MT1Start)触发标志, 任务从非就绪状态转为就绪状态.

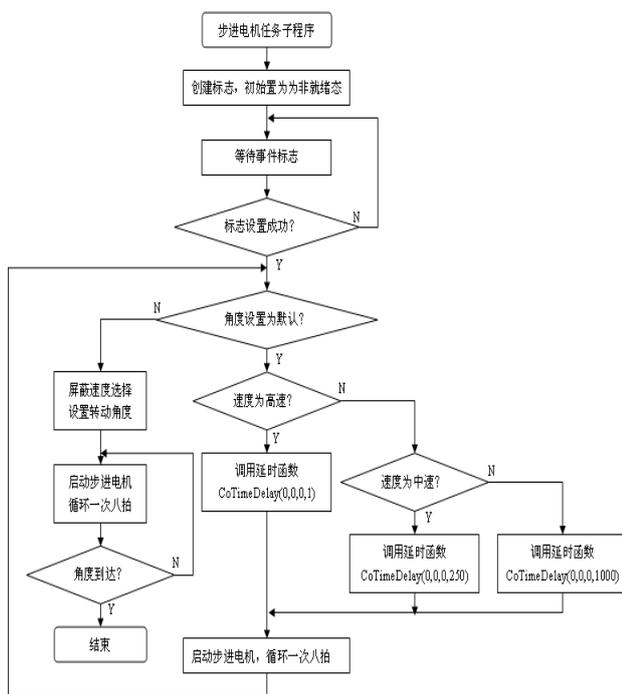


图 4 步进电机任务流程图

电机速度的选择是以电机节拍间的延时作为控制依据的. 步进电机运行方式为设置为四相八拍, 每八拍为一个循环, 即自定义 StepmotorMovePitch()函数为循环函数. 每调用一次循环函数, 执行一次八拍, 电机转过一个齿距角. 因此电机转速的控制可以通过调用 StepmotorMovePitch()函数的快慢来实现. 本系统中用 CoTimeDelay()系统延时函数实现. 对于高、中、低速, 分别调用 CoTimeDelay(0,0,0,1), CoTimeDelay(0,0,0,250), CoTimeDelay(0,0,0,1000)延时 1ms、250ms 与 1000ms.

在电机的角度控制上, 是通过计算指定角度内步

进的步数来实现的. 在选择角度时, 默认屏蔽掉对速度的选择, 其速度标志位为 0x03. 角度的控制通过调用 motorAngleMove(unsigned int angle)函数, 其中参数 angle 为协议字段中对应的角度设置值.

```
unsigned int motorAngleMove(unsigned int angle)
{
    int i=0;
    for(i;i<=angle;i++)
        StepmotorMovePitch(StepMotor_1,mt1Dir);
    return 0;}

```

#### 4 上位机软件设计及系统实现

上位机作为指令的发出者, 需要与下位机达成一致的通信协议, 并提供良好的人机交互界面, 本系统采用 MFC 开发出上位机控制软件, 根据用户选择不同的操作方式, 在上位机软件中根据通信协议组合成相应的控制命令串, 在点击启动按钮时, 下发到下位端 buff, 并进行解析.

##### 4.1 通信协议

定义统一的通信协议是上下位机间通信的关键. 在本系统中, 自定义了 4 个字节 32 位的命令, 作为两者的通信协议. 命令 flag 作为命令起始的标志位; 命令编号作为命令种类标志, 为我们后期扩展其他类型的命令打下基础, 最多支持 4 种类型命令; 板子编号是下位端区分主板和从板的标志, 最多下挂 31 个从板; 电机编号主要区分同一块板子上的不同电机, 如果板上电机驱动芯片支持, 最多可挂接 32 个电机任务; 回执编号用来标识板间发送和接收命令, 为后期扩展回馈信息做好准备. 通信协议具体格式如表 1 所示:

表 1 通信协议格式 单位: bit

命令 Flag	命令编号	板子编号	电机编号	方向	角度	速度	回执编号
2	2	5	5	5	8	2	3

- (1)命令 Flag: 命令起始标志位, 2 位, 规定为 01;
- (2)命令编号: 标识命令种类, 2 位, 规定为 01;
- (3)板子编号: 标识下挂板子, 5 位; 目前有三个: 0x01: 主板; 0x02: 从板 1; 0x03: 从板 2.
- (4)电机编号: 标识每个板子下挂的电机, 5 位; 目前下挂两个电机: 0x01: 1 号电机; 0x02: 2 号电机.
- (5)方向: 标识电机运行方向、状态; 四种状态: 0x00: 顺时针; 0x01: 逆时针; 0x02: 暂停; 0x03 恢复.
- (6)角度: 标识电机运行角度, 实现微调功能, 8 位;

当选择调节速度时,默认填写 0xff; 定义 8 种角度调制: 0x08:45 度; 0x10:90 度; 0x18:135 度; 0x20:180 度; 0x28:225 度; 0x30:270 度; 0x38:315 度; 0x40:360.

(7)速度: 标识电机运行速度, 实现速度调节功能, 2 位; 当选择调节角度时, 此处默认 0x03; 目前定义 3 种速度: 0x00: 高速; 0x01:中速; 0x02: 低速.

(8)回执编号: 标识板间发送和接收命令, 主要用于 CAN 总线上标识命令; 目前定义为 0x01.

## 4.2 上位机软件设计

利用 VS2008 中 MFC 开发环境, 调用串口通信控件 MSComm, 实现上位机和下位机之间的通信控制软件. 在 VS2008 中, 对控件属性的操作都是通过特定的函数来实现的, 这些函数都是 CMSComm 类的成员函数<sup>[9]</sup>. 根据要求, 设计了如图 5 所示的步进电机控制系统的软件.

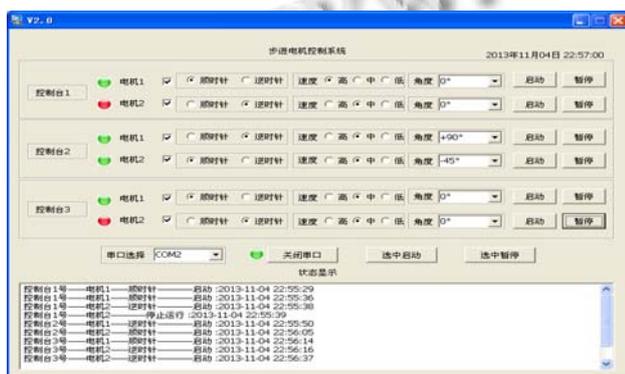


图 5 步进电机控制系统

软件界面简单直观, 实现了电机运行的速度、角度、方向控制, 由红、绿、灰三色灯指示当前运行状态. 实时显示操作日志并记录当前操作时间. 并且支持电机的多选操作, 为实际操作带来方便. 在实际应用中本软件可以根据实际需求扩展控制台数量, 具有

良好的扩展性和适应性. 在实际测试中, 开启运行一周无异常, 表现出良好的可靠性和稳定性.

## 5 结语

本文设计的基于 CoCoX CoOS 操作系统和 CAN 总线的多级联电机控制系统, 采用了移植 CoOS 微控制器操作系统和 CAN 总线技术, 以任务调度和命令转发的方式, 实现了多级联步进电机的协同工作. 结果表明该系统的实时性好, 抗干扰能力强, 并且充分利用了 CAN 总线技术的优势, 提高了电机的控制率. 与 VS2008 中 MFC 的结合进一步扩大了其应用范围.

## 参考文献

- 1 刘涛,王宗义,孔庆磊,武光田.基于 CAN 总线的多电机协调运动控制系统研究,机床与液压,2010,38(3):75-77.
- 2 饶运涛,邹继军,郑勇芸.现场总线 CAN 原理与应用技术.北京:北京航空航天大学出版,2007.
- 3 郭宽明.现场总线技术应用选编.北京:北京航空航天大学出版社,2003.
- 4 NXP Semiconductor Corp.LPC111x/LPC11C1x User Manual. <http://www.NXP.com> .2011.
- 5 桂电-丰宝联合实验室.LPC1100 系列处理器原理及应用.北京:电子工业出版社,2011.
- 6 CoCoX CoOS 用户手册. [http://www.cocox.org/downloadfile/CoOS/PDF/CoOS\\_User\\_Guide\\_CH.pdf](http://www.cocox.org/downloadfile/CoOS/PDF/CoOS_User_Guide_CH.pdf)
- 7 魏衡华,陈根杰,张玉斌,李隆.基于 CAN 总线的步进电机控制系统的设计.自动化与仪表,2010(1):29-32.
- 8 刘文英,武津城,刘艳彬.基于 u c /OS -II 的步进电机控制,科技信息,2009,5:79.
- 9 刘书智,李琳娜.Visual C++实践与提高—串口通信与工程应用.北京:中国铁道出版社,2009.