

嵌入式系统的多路步进电机控制系统的设计

王宜瑜, 宋树祥, 王 斌, 庞中秋
(广西师范大学 电子工程学院, 广西 桂林 541004)

摘要: 设计并实现了一种基于 FreeRTOS 的多路步进电机控制系统, 运用于小型流水线; 系统控制电路采用 STM32F4 系列的微控制器, 结合小功率步进电机驱动器 A4984 以及嵌入式以太网控制器 W5500, 完成硬件设计, 后对其进行软件开发, 设计出可以实时控制并改变电机脉冲起始频率、最大频率、电机转动方向、电机脉冲总数、加速脉冲数以及减速脉冲数等参数的上位机模块; 在电机驱动中加入改进型 S 型算法, 有效的优化步进电机初始速率增加过快导致的步进电机失步和过冲现象; 在实际项目的验证过程中, 证明该系统具有良好的稳定性。

关键词: FreeRTOS; 步进电机; 嵌入式系统; 软件设计

Design of Multi-Channel Stepper Motor Control System for Embedded System

Wang Yiyu, Song Shuxiang, Wang Bin, Pang Zhongqiu

(College of Electronic Engineering, Guangxi Normal University, Guilin 541004, China)

Abstract: Designs and implements a multi-channel stepper motor control system based on FreeRTOS, which is used in small pipeline. The system control circuit uses STM32F4 series microcontrollers, combined with small power stepper motor driver A4984 and embedded Ethernet controller W5500, completes the hardware design. Design a host computer module that can control and change the motor pulse start frequency, maximum frequency, motor rotation direction, total motor pulse number, acceleration pulse number and deceleration pulse number in real time. An improved S-type algorithm is added to the motor drive to effectively optimize the phenomenon of step motor out-of-step and overshoot caused by the excessive increase of the initial speed of the stepper motor. In the verification process of the actual project, it is proved that the system has good stability.

Keyword: FreeRTOS; stepper motor; embeded system; software design

0 引言

随着社会与经济的飞速发展, 自动化控制代替人为操作越显明显^[1]。步进电机作为自动化控制重要的执行机构来说, 其控制一直以来都是一个重要的研究课题, 其主要研究集中在加减速曲线优化和具体实现上。与此同时随着嵌入式技术的快速发展, 特别是其 ARM 架构芯片的性能有了大幅度的提升以及嵌入式实时控制系统的功能变得更加完善。因此将嵌入式技术应用于多路步进电机控制系统中不仅可以提高了步进电机控制的实时性和可靠性, 而且还降低了控制系统的成本^[2]。本文设计的主控芯片使用了 ST 公司的 STM32F4 系列微处理器, 嵌入式以太网控制器 W5500、电机控制芯片 A4984、光电传感器、RFID 射频识别传感器等多种外设, 操作系统采用嵌入式实时操作系统 FreeRTOS, 上位机控制软件应用程序采用 PyQt, 并进行相应的软件程序开发, 本设计致力于设计一种可靠稳定, 实时性高的多路步进电机控制系统。

1 硬件电路设计

1.1 总体设计

本设计总体架构如图 1 所示。主控芯片通过嵌入式以太网控制器 W5500 与上位机通信, 接收上位机软件发送的命令, 并向上位机返回现在所处的工作状态。W5500 通过 SPI 通信将上位机命令发送给主控制端 STM32, 后由主控进行命令解析, 得到指令中的电机脉冲数、频率以及正反转等参数, 最终得到驱动电机的脉冲数量、脉冲频率以及脉冲信号顺序, 从而控制各路步进电机按照指令所需的频率精确达到目的位置。

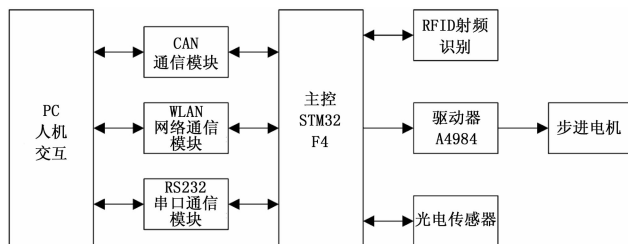


图 1 系统总体结构图

收稿日期: 2019-02-18; 修回日期: 2019-03-29。

作者简介: 王宜瑜(1993-), 男, 硕士, 广西师范大学, 主要从事嵌入式系统研究与应用方向的研究。

通讯作者: 宋树祥(1970-), 男, 教授, 广西师范大学, 主要从事集成电路、嵌入式系统研究与应用方向的研究。

本设计电路中主控芯片采用 ST 公司的 STM32F407 系列微控制器, 电机驱动芯片采用 A4984SLPTR-T 芯片, 步进电机选择俩相四线的混合式步进电机。

1.2 网络通信模块

STM32 网络通信主要有两种方式一种采用移植 LWIP

协议栈, 一种是采用外置网络芯片与主控芯片之间进行通信。由于主控芯片已经移植了小型操作系统 FreeRTOS, 为了不增加主控芯片的内存压力且外置网络芯片加以外部晶振, 较 LWIP 协议栈以及内部晶振时钟会更加精确, 故采用外置网络芯片加以外部晶振来进行网络通信。网络通信芯片采用韩国 WIZnet 公司的 W5500 芯片, 该芯片集成了 IP/TCP 协议栈, 方编程模块化, 有助于系统后期的扩展性, 也减少了主控芯片的资源占用。

1.3 电机驱动模块

本系统电路中电机驱动模块采用 Allegro 公司的 A4984SLPTR-T 电机驱动器, 该驱动器是一个完整的带有内置转换器的微电机驱动器。A4984SLPTR-T 包括一个固定停机时间电流稳压器, 该稳压器可在慢或混合衰减模式下工作。该电机驱动器具有高达 35 V 和 ±2 A 的输入驱动能力, 且控制简单。若电机的温度过高, 会导致电机力矩降低乃至失步, 故将电机的电流设定为 1.414 A, 既可以使电机芯片不过载, 又尽可能的使得电机的利用率更高。驱动模块电路如图 2 所示。

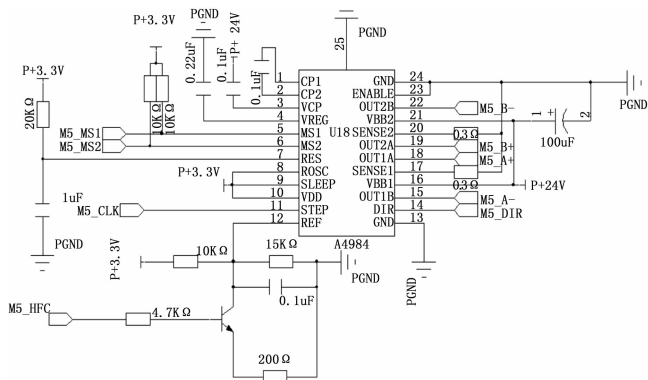


图 2 电机驱动模块原理图

主控制器只需配置好与电机驱动芯片上相连的的 ENABLE、DIR、STEP、MS1、MS2 五个管脚即可完全控制此步进电机。ENABLE 为步进电机控制器的使能端, 通过给与 ENABLE 命令来决定电机是否处于使能状态或者电机锁死状态防止电机自主或者人为滑动; DIR 为电机的方向控制端, 通过接收主控给与的电平高低来控制电机的转动方向; STEP 为电机的脉冲控制端, 通过接收主控发出的脉冲数量来控制电机的转动次数以及脉冲频率来控制电机转速; MS1 以及 MS2 为电机细分控制端, 都置高电平即为该系统的细分模式: 8 细分。

2 下位机软件系统设计

2.1 软件总体设计

根据设计需要实现的目标, 本文把系统分为不同的子功能, 充分利用嵌入式系统多任务的优势, 把各个子功能作为系统中不同的任务进程实现^[3], 包括: 网络通信任务、上位机通讯协议解析任务、电机驱动任务、光电传感器任务等, 各个任务通过操作系统的信号量进行同步运行, 由上位机软件统筹规划。为了保证步进电机系统的实时性,

其中电机驱动进程为核心进程, 设定其任务优先级为次高, 除 W5500 的通信模块, 没有其他进程可以抢占他的 CPU 控制权。

本文主要介绍软件设计网络通信管理和电机控制的实现: 从网口读取运动指令, 转换为实际电机的移动。

1) 上位机软件通过工作人员设定的电机工作时序来发送相对应的电机控制指令, 命令通过网络传输至网络控制芯片中的命令缓存区中, 后由 SPI 通信发送至主控制端 STM32 所建立的循环链表中进行存储, 防止命令丢失, 最后又主控制端进行相对应的命令解析;

2) 主控制端解析电机控制命令后, 使用电机控制算法来控制电机来进行相对应的电机动作, 其中主要参数包括电机脉冲频率、电机转动方向、脉冲总数以及加减速脉冲数等信息;

3) 完成动作后, 主控制端再通过 SPI 向网络控制芯片传输完成指令, 请求下一指令; 网络控制芯片向上位机返回相对应的结束命令, 结束动作, 等待下一次协议命令;

2.2 嵌入式系统选择

ARM 系列的微处理器不支持大型操作系统的移植, 但是在单片机模式下的纵向处理程序处理又满足不了任务量多的程序设计。而此时嵌入式实时操作系统既满足 ARM 处理器的内存要求且可以进行多个任务假性同步运行, 一个处理器核心每次仅可以执行一个任务, 但实时操作系统的主要工作则是将任务切换的时间变得非常短, 而这已经满足了本设计对操作系统的需求。

本设计采用了 FreeRTOS 嵌入式实时操作系统。FreeRTOS 实时操作系统内核占用空间小、源码公开、可移植、实时性高, 可免费用于商业用途。其基本功能包括: 通过时间片调度切换任务; 多种信号量应用于各种场景; 多个实时任务且及其优先级的配置; 文件任务的管理等。

在嵌入式领域, FreeRTOS 是不多的同时具有实行性、开源性, 可靠性, 易用性, 多平台支持等特点的嵌入式操作系统^[4]。

2.3 网络通信管理

由于单步执行的程序架构中, 下位机会来不及处理上位机发送的连续运动指令, 上位机向网关发送命令, 对数据进行实时监控, 但是由于接收到的命令要从流水线的控制器交互, 交互需要一个过程, 在这个过程中, 导致上位机命令冗余, 存储在一个数组里, 但是上位机长时间收不到命令, 网络连接会中断, 这种异常在工业控制里是坚决不允许的, 所以, 对处理的命令进行过滤, 即设计了一个环形链表算法, 将先存储在链表中但未处理的命令删除, 把刚接收到的命令, 插入尾部。如图 3 环形链表示意图。

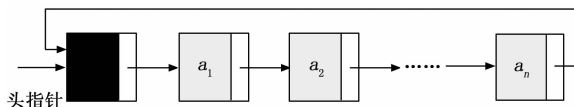


图 3 环形链表示意图

环形链表结构体定义所示:

```
typedef struct rx_net_data
{
char rx_data[CMD_MAX_DATA_LEN];
unsigned char rx_data_len;
SemaphoreHandle_t task_binarySemaphore;
SemaphoreHandle_t ISR_binarySemaphore;
Struct rx_net_data * next;
} RX_NET_DATA_LinkedListDef;
```

结构体中的 rx_data 用来存放上位机发送过来的指令，rx_data_len 表示 rx_data 数据长度，task_binarySemaphore 为任务级的二值信号量，ISR_binarySemaphore 为中断级的信号量，用来同步中断中接收到的数据与数据包解析任务同步。

中断接收指令过程，流程图如图 4 所示。

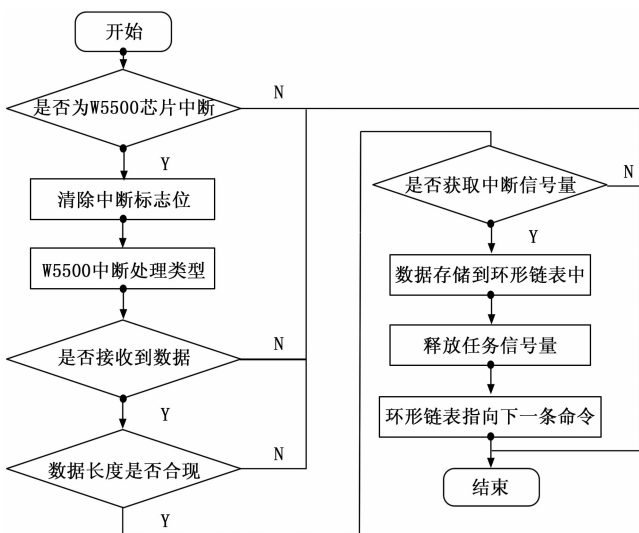


图 4 网络通信流程图

2.4 电机驱动控制

步进电机的控制方式通常分为两种，一种是使用定时器中断来驱动步进电机，另一种则是定时器 PWM 脉宽调制来驱动。本文采用的则是定时器 PWM 脉宽调制，通过设置自动装载值和预分频值来控制占空比，从而控制电机的频率。

由于步进电机本身的局限性，在步进电机起始阶段频率变化过大，则会导致电机失步、过冲现象，所以启动阶段需要一段较为缓慢的预热阶段。为了防止电机出现过冲或失步的现象，在电机控制中加入改进型 S 型算法，则有效的规避了该现象。

由文献 [3] 可得 S 型曲线的速度函数，所得的 S 型曲线，如图 5 所示：

$$V(t) = \begin{cases} \frac{a^2 t^2}{2(V_m - V_s)} + V_s & t \in [0, T_1] \\ \frac{a^2 (t - T_2)^2}{2(V_s - V_m)} + V_m & t \in [T_1, T_2] \\ V_m & t \in [T_2, T_3] \\ \frac{a^2 (t - T_3)^2}{2(V_e - V_m)} + V_m & t \in [T_3, T_4] \\ \frac{a^2 (t - T_5)^2}{2(V_m - V_e)} + V_e & t \in [T_4, T_5] \end{cases} \quad (1)$$

式 (1) 中：a 为设定的系统加速度；V_m 为步进电机运行的最大速度；V_s 为步进电机运行的起始速度。

S 型算法的好处是速度具有很好的平滑性，运动精度也很高^[5-6]。S 型算法又名 Sigmoid 函数，Sigmoid 函数原型为：

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2)$$

对其加以修改，式如下所示：

$$F(i) \begin{cases} F_{\min} + \frac{(F_{\max} - F_{\min})}{1 + e^{-\frac{a \cdot i}{k}}} & i \in [0, T_1] \\ F_{\max} & i \in [T_1, T_2] \\ F_{\max} - \frac{(F_{\max} - F_{\min})}{1 + e^{-\frac{a \cdot i}{k}}} & i \in [0, T_1] \end{cases} \quad (3)$$

在式 (3) 所示：F(i) 为即时频率值；F_{min} 为步进电机运行的起始速度；F_{max} 为步进电机运行的最大速度；k 为代表设定的加速度；i 为加速或者减速的脉冲索引 (i ≥ 2 num)。

给定 K 值为 7 时，电机运作较为良好，值得提醒的是加速阶段或者减速阶段脉冲索引都从 0 开始计数。

通过 MATLAB 对公式 (3) 的加速部分进行仿真，取电机初始频率 F_{min} 为 2 000 Hz，最大频率 F_{max} 为 3 000 Hz，加速脉冲数 num (电机一个脉冲数为 1.8°) 为 4 000，T₁ 取值为 2 000，加速度 k 为 7，所获得加速 S 曲线如图 5 所示：电机转动主要分为 3 个阶段，加加速，匀加速以及加减速阶段^[7-8]，减速和加速阶段类似，故不做多余讲解。

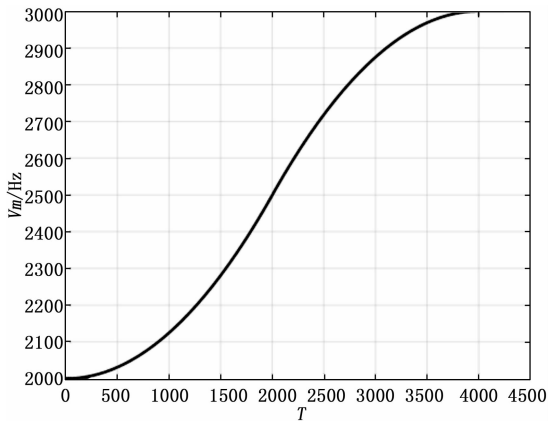


图 5 文献 [3] S 型加速曲线图

通过图 5，图 6 两者比较可以明显发现，本文中的 S 型算法有效的避免了电机初始加速速率变化过大而导致的电机速率突变的情况，避免了电机过冲与丢步的现象；电机也可以在最短的时间内达到所需的频率，也大大的降低系统的功耗^[9-10]。将图 5、图 6 进行对比，通过对 MATLAB 仿真图进行取脉冲值 0，1 000，2 000，3 000，4 000，得到电机的实时频率值，比较 S 型加速减速的脉冲数和频率之间的关系如表 1。

3 上位机软件设计

PC 上位机使用 python3. 5 和 pyqt5 进行研发编写。

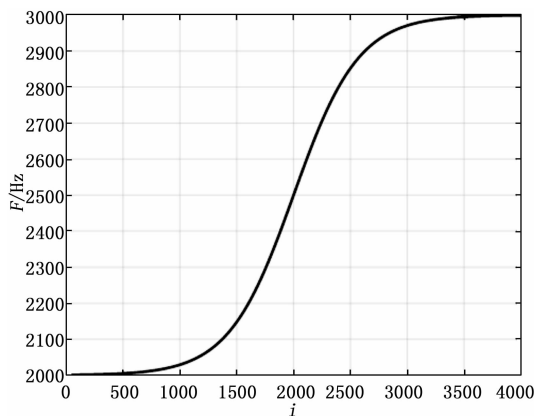


图 6 本文 S 型加速曲线图

表 1 步进脉冲时间间隔

	脉冲=0	脉冲=1 000	脉冲=2 000	脉冲=3 000	脉冲=4 000
本文(实时频率)	2 000	2 030	2 500	2 971	3 000
文献 3(实时频率)	2 000	2 125	2 500	2 875	3 000

Python 是功能很强大的跨平台解释性脚本语言, 而 PyQt 是 python 针对 Qt 的一门功能性拓展^[11]。由于 Qt 本身功能的强大性加上 Python 语言的良好扩展性, 该设计选择 Python 和 PyQt 作为开发语言和开发环境。

界面如图 7 所示, 电机开始侦听则开始与下位机进行连接, 连接成功则显示连接成功, 如果失败则返回连接失败。界面主要参数包括: 脉冲总数、电机转向、起始频率值、最大频率值、脉冲加速脉冲数、减速脉冲数以及电机选择等。通讯协议数据帧以图 7 为例为:

0231 | CT | MA | 500 | 10 000 | CW | 20 000 | 5 000 | 5 000 | AABB03。其中起始位 0231, 控制位 CT; 电机选择为 MA; 起始频率为 500; 最大频率 10 000; 电机转向, 正向为 CW, 反向为 CCW; 电机总脉冲数为 20 000; 加速脉冲数为 5 000; 减速脉冲数为 5 000; 校验位为 AABB 以及截止位为 03。当电机动作完成之后, 下位机需返回信息给上位机告知动作完成。返回信息的通讯协议仅将控制位和电机选择进行互换, 表示是下位机向上位机返回的信息。电机动作完全结束后, 需将电机进行复位, 等待下一次的动作命令。复位是通过触碰光电传感器从而使电机返回初



图 7 上位机软件示意图

始位置, 后进入待机状态, 等待上位机的协议命令。电机的误差就不会进行叠加。

4 实验结果与分析

在实际应用中有 8 台步进电机同时工作, 所以在 FreeRTOS 中创建 8 个实时电机控制任务。给电机相同起始频率 2 000 以及最大频率 5 000, 不同脉冲数, 比较丢失步的比例。由于电机转动需要主控制端给脉冲, 而每给一个脉冲数就要进入一次定时器中断, 故在定时器中断中给一个叠加参数, 当脉冲数完成后由串口通信打印至串口调试助手, 即可得到其电机实际接受到的脉冲数。

表 2 电机丢失步比例情况

电机	给定脉冲数	实际脉冲数	丢失步比例
1	1 000	993	0.7%
2	2 000	1 985	0.75%
3	5 000	4 988	0.24%
4	8 000	7 993	0.08%
5	10 000	9 976	0.24%
6	15 000	14 982	0.12%
7	20 000	19 966	0.17%
8	30 000	29 973	0.09%

由上述数据以及通过若干次测试可以显示该电机控制系统的丢失步比例均在 1% 以内, 电机一个脉冲为 1.8°, 加上电机细分数为 8 细分, 所以电机丢失步所造成的转动距离非常的小。因此将此控制系统运用至实际工程中去, 实际项目工程图如图 8 所示。



图 8 项目工程图

5 结论

用嵌入式实时控制系统和步进电机控制技术, 设计了一种基于嵌入式系统的多路步进电机控制系统, 并且阐述了系统的硬件设计、步进电机控制技术以及上位机控制软件应用程序。通过测试表明该系统提高了电机控制的稳定性以及可靠性, 有效的防止步进电机失步和过冲现象, 有效的提高了试管架的检测效率; 另一方面使用 ARM7 微处理器的嵌入式控制技术, 使得控制成本大大降低, 更适合中小型控制系统的低成本要求。该系统现已在多进样流水线中运用, 且系统运作良好。