

DOI:10.3969/j.issn.1001-4551.2014.05.014

模块化手套机控制系统设计研究

赵子轩,袁嫣红*,张建业

(浙江理工大学机械与自动控制学院,浙江 杭州 310018)

摘要:针对手套机机械结构的变动所产生的手套机控制系统移植性问题,对现有手套机机械结构、硬件电路、电磁铁、电磁阀、Linux 驱动、应用层主控程序等方面进行了研究,对手套机控制系统硬件系统及软件系统进行了归纳,提出了一种基于 Linux 系统的模块化手套机控制系统。首先对现有手套机不同子系统进行分析总结,将手套机的主干部分划分为 6 个模块,依次为选针模块、软轴控制模块、纱线模块、信号检测模块、显示界面模块、按键模块;然后对内核部分的驱动、系统启动脚本、应用层的主控程序软件部分进行编写;最后实现了手套机正常的编制功能。研究表明,通过对手套机的模块化,使应用层手套机主程序与底层分离,减少因手套机子控制系统的变动带来软件上的修改,使手套机控制系统具有较高的移植性。

关键词:Linux;模块化;驱动;启动脚本

中图分类号:TH39;TP273

文献标志码:A

文章编号:1001-4551(2014)05-0607-05

Modular glove machine control system design

ZHAO Zi-xuan, YUAN Yan-hong, ZHANG Jian-yi

(School of Mechanical Engineering & Automation, Zhejiang Sci-Tech University, Hangzhou 310018, China)

Abstract: Aiming at the transplantaion problem of glove machine control system with mechanical structure change of glove machine, it was studied, including the existing glove machine mechanical structure, hardware circuit, solenoid, solenoid valve, Linux driver, application layer of main control program and so on. The control system and software system of glove machine were summarized. A control system for modular glove machine based on Linux system was put forward. Firstly, the existing glove mazine different subsystems were analyzed and summarized. the glove machine's main part was divided into 6 modules, with needle selection module, control module, the soft yarn module, signal detection module, display module, keyboard module. The kernel part of the drive, system startup scripts, application layer control procedure of the software part weve wrote. Finally, the system function glove machine normal was realized. Research results show that, through the modular of glove knitting machine, the main program and the bottom of application layer in glove machine is separated, it reduces the change of software modifications by the glove machine control system brought, the glove machine control system has higher transplantaion.

Key words: Linux; modularization; driver; startup script

0 引 言

通常手套机由信号检测模块、选针模块、软轴控制模块、纱线模块、显示界面模块、按键模块构成。随着生活提高,人们对手套的需求量也越来越大,从而推动了手套机技术的革新^[1]。国内外许多同类全自动电

脑手套机,其选针模块、软轴控制模块、纱线模块都出现了多种结构。这些模块在不同的设备中机械结构或控制方式并不完全相同。

对于针选针模块,有些机型采用电磁铁选针模式,利用多路电磁铁实现电控选针,从而达到动态选针的目的^[2];大部分机械式选针模式则采用固定滚筒选针,在

收稿日期:2013-12-30

作者简介:赵子轩(1987-),男,河南安阳人,主要从事机电控制方面的研究. E-mail:g5912667@163.com

通信联系人:袁嫣红,女,教授,硕士生导师. E-mail:yyh@zstu.edu.cn

小滚筒上的不同部位镶有不同的滚筒销,通过机械传动设计保证选针实现。其选针时序固定,仅能更换滚筒以改变选针时序,但其自身有着选针动作稳定有点。

对于软轴控制模块,有些机型采用大滚筒式软轴控制系统,软轴之间的关系通过机械结构的变换实现。通过在大滚筒上的不同部位镶有不同的传动脚,使 5 路软轴实现固定动作,系统中电磁铁对 5 路软轴整体控制;气缸式软轴控制系统则通过电磁阀控制特制气缸动作,可实现 5 路软轴单独控制,5 路软轴动作互不影响^[3]。

对于纱线模块,简单的机型由张力器和导纱器将纱线送入编织机构,纱线单一且无电控参与;先进的机型则采用双色换纱结构,由主导纱器、副导纱器控制不同的纱线,通过电磁铁控制实现导纱器的切换,从而达到换纱目的^[4]。

本研究针对可能的机型研究一种通用型的手套机控制系统,可以通过简单的软件配置和硬件模块增减,实现特定机型的控制要求。通过引入 Linux 系统,并将手套机控制系统划分为 6 个模块,每个模块基于总线—设备—驱动结构思想分开编写设备文件、驱动文件,从而使手套机控制系统不仅可实现手套编织等功能,简化整个手套机的软件,而且可有效地解决因机械结构的变动带来的软件的变动。

1 总体方案

手套机样机如图 1 所示。如前所述,不同手套机型上,同一功能模块可能在机械结构与控制电路上并不相同,这样针对不同手套机需要开发不同的控制系统。为此本研究针对市场已有机械结构,将手套机控制系统的硬件、软件模块化,同时引入 Linux 操作系统,从而实现快速组合的通用型控制系统开发要求。

手套机的硬件模块划分如图 2 所示。

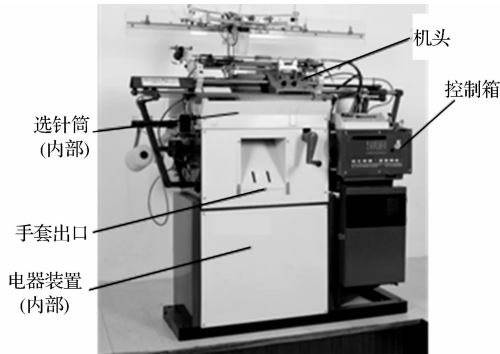


图 1 手套机样图

嵌入式 Linux 一般由引导程序、Linux 内核、文件系统和应用程序组成^[5]。基于 Linux 的手套机嵌入式

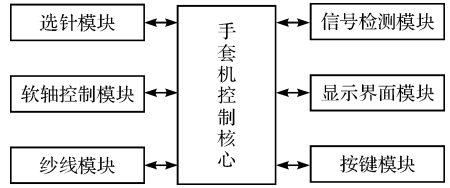


图 2 手套机硬件模块组成

控制系统,应对 Linux 内核、文件系统和应用程序进行相应调整。在 Linux 内核中,加入了 6 个模块对应的驱动。在文件系统中,编写启动脚/etc/init. d/rcS,通过菜单选择,完成对手套机设备的启动检测。最后加载应用程序,即手套机主程序。

对每个模块,将采用 Linux 驱动经典结构:驱动—总线—设备。驱动、设备分别向总线注册,而且驱动(或设备)注册时会调用相应设备(或驱动)注册,其实质为总线依据设备结构体成员 name,驱动结构体成员 id_table 是否匹配来决定是否调用驱动结构体成员 probe 来完成注册^[6]。

将手套机模块化后—即在底层对驱动和设备进行分离,对手套机启动脚本编写。对于各种手套机控制结构,仅需修改或添加设备文件,而不会对应用层主程序产生影响。

2 手套机模块化驱动程序

手套机模块化驱动程序是模块化手套机控制系统设计中重要组成部分。正是因为同种模块的多种设备实现了同样控制动作,使得基于 Linux 手套机控制系统每个模块驱动文件与设备的分离,不仅驱动文件对现有多种设备文件具有通用性,而且对将来产生的新设备起到了兼容,同时底层驱动与应用层主控程序分离,减少了软件上的修改,也是手套机启动脚本可以对设备进行选择的基础。

Linux 驱动一般分为 3 类:字符设备、块设备、网络设备^[7-8]。手套机控制系统中系统环境的网络设备网卡(DM9000AEP),块设备 nand(K9F2G08)芯片厂商出厂时已有驱动,仅做移植修改就可,无需编写。对于 6 个模块,均为字符型设备,独立编写。

以驱动对应设备多寡为标准,现将 6 个模块分 2 部分,手机套各模块与抽象后 Linux 驱动设备对应关系如表 1 所示。

表 1 手套机各模块与抽象后 Linux 驱动及设备对应关系 1

模块名称	Linux 驱动名	Linux 设备名一	Linux 设备名二
选针模块	needle_drv	needle_electromagnet_dev	needle_machine_dev
软轴控制模块	flexible_drv	flexible_electromagnet_dev	flexible_radiotube_dev
纱线模块	yarn_drv	yarn_electromagnet_dev	yarn_machine_dev

2.1 多设备模块

该部分驱动与设备关系是一对二,故可采用 platform 总线。platform 总线是 Linux 驱动经典结构的代表,即驱动—总线—设备模型。简而言之,该结构可将同类型设备的共性,即实现功能编写于驱动文件,而将各设备的向异性,及硬件属性,编写于设备文件中。platform 总线可依据设备 platform_device 成员 name, 驱动 platform_driver 成员 id_table 是否匹配来决定是否调用驱动 platform_driver 成员 probe 来完成注册^[9]。

2.2.1 设备文件的注册

在 platform 设备文件 xxx.c 中(xxx 即为表中 Linux 设备名一与 Linux 设备名二),platform 通过设备注册函数 platform_device_register() 完成 platform_device 数据结构 xxx_dev, resource 数据结构 xxx_resource []注册,其中主要成员为:

start = Physical_address, end = SIZ: 依手套机控制电路原理图进行设置,主要是电磁铁,电磁阀物理的地址,以及微动开关的中断号设置。

name = "xxx": 设备与驱动匹配标志,其将决定调用哪个驱动。

platform_data = is_xxx: 同模块不同设备区分标志,为便于驱动的编写。

对于虚拟的设备 needle_machine_dev.c (即单色纱线导纱机构),仅赋值 name, platform_data 完成格式统一。

2.2.2 驱动文件的注册

在 platform 驱动文 yyy.c 中(yyy 即为表中 linux 驱动名)中,其核心为完成数组 yyy__table[], 函数 yyy

_probe()注册。以纱线模块为例的分析,数组 yyy__table[] 原型为:

```
static struct platform_device_id yyy_table[] = {
    {
        .name = "xxx",
    }, {} ,
};
是驱动所支持设备 id 数组,对于纱线模块,其驱动文件 yarn_drv.c 将含有
static struct platform_device_id needle_drv_table[] = {
    {
        .name = " needle_electromagnet_dev ",
    }, {
        .name = " needle_machine_dev ",
    }, {} ,
};
```

从而匹配 yarn_electromagnet_dev.c, yarn_machine_dev.c 设备文件中 platform_device 成员 name。

yarn_drv.c 驱动文件中 probe() 将用 platform_get_resource() 获得所注册设备硬件信息(即 yarn_electromagnet_dev_resource[], yarn_machine_dev_resource[] 之一),而在 file_operations 成员 ioctl() 函数将依据 probe() 获得的硬件情况,编写统一控制的函数。从而 yarn_drv.c 驱动使不同纱线模块设备的实现了相同的功能。

ioctl 设备控制函数原型为:

```
static int ioctl(struct inode *inodep, struct file * filp, unsigned, int cmd, unsigned long arg)
```

设备文件注册中 platform_data = is_xxx 在 ioctl() 函数中区分了不同设备,手套机驱动中 Ioctl() 实现的命令 1、命令 2 如表 2、表 3 所示。

表 2 手套机驱动中 Ioctl() 实现的命令 1

模块名称	Linux 驱动名	命令 cmd	命令参数 arg	命令概述
选针模块	needle_drv	NEEDLE	时序	依时序,完成多路选针
软轴控制模块	flexible_drv	FLEXIBLE	时序	依时序,完成 5 路软轴控制
纱线模块	yarn_drv	YARN	无	纱线颜色是否改变

表 3 手套机驱动中 Ioctl() 实现的命令 2

模块名称	Linux 设备名	控制内容
选针模块	needle_electromagnet_dev	大电磁铁,多路小电磁铁
	needle_machine_dev	大电磁铁
软轴控	flexible_electromagnet_dev	电磁铁
制模块	flexible_radiotube_dev	5 路电磁阀
纱线模块	yarn_electromagnet_dev	电磁铁
	yarn_machine_dev	空操作

2.2 单设备模块

对于这 3 个模块,由于驱动与设备一一对应(如表 4 所示),可采用传统字符驱动编写流程^[10],单个文

件将设备和驱动一次性注册,带来程序的简洁性,但并不合适手套机的模块化的思想,例:信号检测模块中接近开关采用新的结构,有可能导致信号检测模块的软件部分改变,以及手套机主程序的改变,即导致手套机整个系统的改变。故考虑对新结构的兼容性,依然采用如第一部分所述设备—总线—驱动模型。

表 4 手套机各模块与抽象后 Linux 驱动及设备对应关系 2

模块名称	Linux 驱动名	Linux 设备名
信号检测模块	signal_drv	signal_dev
显示界面模块	lcd_drv	lcd_dev
按键模块	key_drv	key_dev

综上所述,6 个模块均采用 Linux 驱动经典结构:驱动—总线—设备^[11],以达到同一个驱动实现不同设备的操作,亦可对将来新设备结构兼容,从而实现手套机控制系统的模块化。

2.3 手套机选针电磁铁驱动实现

对于电磁铁选针结构的手套机,因其多达 92 路的选针电磁铁,故而其总的功耗需求量比较大。针对这种情况,对相关的源代码进行编写时,应对选针电磁铁的吸合时间进行合理的控制,以达到降低功耗的目的,同时也延长选针电磁铁的使用寿命。选针电磁铁的流程图如图 3 所示。

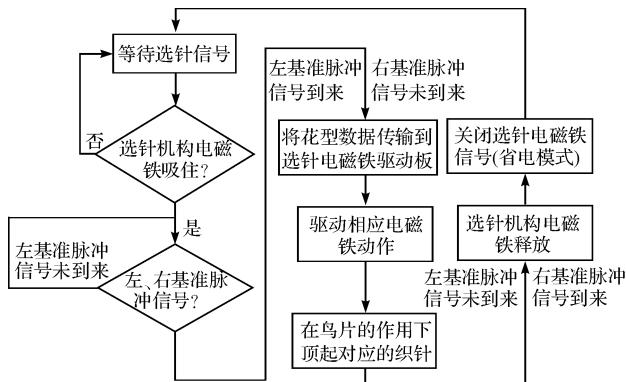


图 3 选针驱动电磁铁程序流程图

3 手套机设备检测启动程序

正是由于手套机控制系统的模块化,使得手套机设备检测启动程序可以在人工干预的情况下,进行对设备文件进行选择性的加载,使手套机控制系统具有智能化的特点。

Linux 内核启动时,如不添加/etc/inittab 文件,默认情况下将会分析/etc/init.d/rcS,其中 rcS 为 shell 文件,在该文件中可以完成对 Linux 应用层的初始化^[12],手套机硬件自检测启动程序流程如图 4 所示。

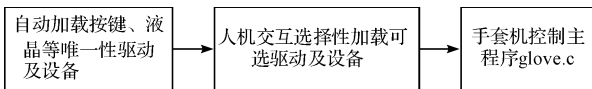


图 4 手套机硬件自检测启动程序流程

除系统环境所移植的驱动采用固态加载(即与内核统一编译,Linux 内启动,其将自动加载)外,6 个模块均采用动态加载方式(命令行模式 insmod xxx.ko),分 3 步加载。

3.1 单设备文件加载

信号检测模块、显示界面模块、按键模块的设备文件及驱动文件加载,其中显示界面模块、按键模块将参

与第二部分设备文件选。

3.2 多设备文件加载

要控制人员根据手套机的结构对选针模块、软轴控制模块、纱线模块设备进行二选一,依次运行 needle_dev_selsect.c, flexible_dev_selsect.c, yarn_dev_selsect.c 进行设备选择,同时在子 shell 脚本中/glove_selsect.sh 记录设备选择。3 个模块的设备选择完毕,以("/glove_selsect.sh)"启动子 shell 脚本,进行设备文件加载。以纱线模块的 yarn_dev_selsect.c 为例,流程图如图 5 所示。



图 5 yarn_dev_selsect.c 流程图

即:

```

Int Main()
{
...
fd = fopen("/glove_selsect.sh", "w+");
...
LCM_Yarn(); //显示纱线模块设备选择
Key = KEY_Read(); //先读按键
if( Key = KEY_INVALIDE) { //有按键
KEY_Wait_Realease(); //等待按键释放
switch( Key) { //判断按键
case LEFT: /* 左键按下选择单色纱线导纱机构 */
fprintf( fd, "insmod yarn_electromagnet_dev. ko/n"); /* 记录所选设备 */
break;
case RIGHT: /* 右键按下选择单色纱线导纱机构 */
fprintf( fd, "insmod yarn_machine_dev. ko/n");
//记录所选设备
break;
...
}
}

```

3.3 主程序 glove.c 调用

在底层中已完成设备与驱动的隔离,启动文件已将所需设备加载,故 glove.c 主程序仅对手套机 6 个模块对应驱动调用,其所做完全与具体设备无关,从而实现手套机的模块化,手套机与主程序流程图如图 6 所示。

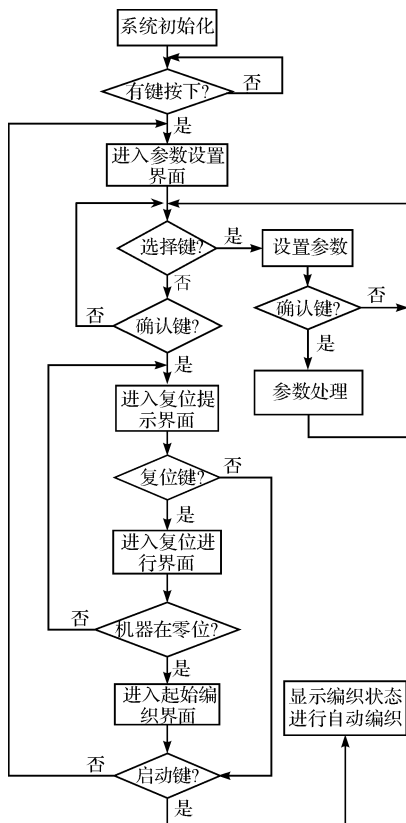


图6 手套机主程序流程图

4 结束语

随着手套机的机电一体化水平的提高,手套机的一些子控制系统由传统的机械型转化为机电结合型,而且电控比重越来越大,甚至有超过机械部分的趋势。虽然电控型有着结构简单、操作灵活等优点,但传统的机械型有着稳定的优点,故两种类型各自占据市场的一定比例。本研究将手套机的主要部分模块化,使现有常见的不同子控制系统的手套机可以使用统一的主控程序。

基于Linux系统,本研究设计了模块化手套机控制系统;基于总线—设备—驱动结构,该系统将传统的

手套机控制部分划归为6个模块,并针对每个模块编写通用的驱动,实现多种同类设备控制。通过对手套机的模块化,使应用层手套机主程序与底层分离,减少因手套机机械结构的变动带来软件上的修改。该设计已实现模块化手套机控制系统的基本功能。

利用该设计所提出的模块化思想,可对手套机新、旧子系统在软件上的兼容性问题提供良好的解决办法;使手套机不再因为部分子系统的改变而带来整个软件系统的改变。

参考文献(References):

- [1] 吕渭贤. 我国纺织机械行业自动化技术应用现状与发展趋势[J]. 江苏纺织, 2004(5): 8-10.
- [2] 郭得权, 袁嫣红, 张建业. 基于ARM的手套机控制器设计[J]. 机电工程, 2012, 29(1): 75-78.
- [3] 鲁红良. 全自动电脑手套机新型软轴控制装置[J]. 针织工业, 2011, 23(3): 11-13.
- [4] 梁志华. SGM_F7型手套机改造为魔术手套机的实践[J]. 上海纺织科技, 2003, 31(6): 33.
- [5] 王红凯. 基于Linux嵌入式全自动横机软件系统设计[J]. 纺织学报, 2008(29): 101-105.
- [6] 宋宝华. 设备驱动开发详解[M]. 北京: 人民邮电出版社, 2012.
- [7] 毛德操, 胡希明. Linux内核源代码情景分析[M]. 杭州: 浙江大学出版社, 2001.
- [8] 高洁, 罗凌红. 基于Linux 2.6内核开发Mini GUI的触摸屏驱动[J]. 兵工自动化, 2013, 32(2): 94-96.
- [9] 冯国进. Linux驱动程序开发实例[M]. 北京: 机械工业出版社, 2011.
- [10] 韦东山. 嵌入式Linux应用开发完全手册[M]. 北京: 人民邮电出版社, 2012.
- [11] 刘峥嵘. 嵌入式Linux应用开发详解[M]. 北京: 机械工业出版社, 2004.
- [12] 刘森. 嵌入式系统接口设计与Linux驱动程序开发[M]. 北京: 北京航空航天大学出版社, 2006.

[编辑: 洪炜娜]

本文引用格式:

赵子轩, 袁嫣红, 张建业. 模块化手套机控制系统设计研究[J]. 机电工程, 2014, 31(5): 607-611.

ZHAO Zi-xuan, YUAN Yan-hong, ZHANG Jian-yi. Modular glove machine control system design[J]. Journal of Mechanical & Electrical Engineering, 2014, 31(5): 607-611.

《机电工程》杂志: <http://www.meem.com.cn>