

EPICS 下基于 ARM 处理器的电机控制系统

黄 辉 黄跃峰 李勇平 尹聪聪 贾文红

(中国科学院上海应用物理研究所 上海 201800)

摘要 提出一种在 EPICS 下开发的电机控制系统,由运行 Linux 操作系统的 S3C2440 计算机和 MAXnet 电机控制器组合而成,详细分析系统设计、软件编写等工作,证明了基于 ARM 处理器的电机控制系统的可行性。该系统成本低体积小,测试表明其可满足位置控制使用要求,但时间响应性能与现有系统还有差距。

关键词 电机控制, ARM, EPICS, 上海光源

中图分类号 TP273.5

上海光源(SSRF)线站电机控制系统在 EPICS 环境下开发而成,由运行 VxWorks 操作系统的 MVME5500 单板机加上 MAXv8000 电机控制器完成对电机的控制。其性能优良,但 VxWorks 价格昂贵、VME 机箱占用空间较大。本文提出一种由运行 Linux 的 S3C2440 计算机(ARM9 内核)和 MAXnet 电机控制器组成的全新电机控制系统,该系统目前已能正常工作。

1 总体结构

EPICS 为美国 Argonne 和 Los Alamos 等实验室为大型物理和工业装置联合开发的分布式控制软件工具,基于标准模型的 Server/Client 模式,具有结构灵活、易于扩展、完全开放源代码等优点,是国际大型光源和对撞机控制系统的主流软件开发工具^[1]。基于 ARM(Advanced RISC Machine)处理器的电机控制系统分为三个部分:(1) OPI(Operator Interface):运行 Linux 或 Windows 操作系统的中控电脑或 Unix 工作站,通过使用 EPICS 提供的 MEDM 或 EDM 设计的人机界面监控各种控制过程,可在同一台 PC 机上对多个 IOC 的运行状况实行监控;(2) IOC:由基于 ARM 架构并安装 Linux 操作系统的 S3C2440 计算机和运行在其中的 EPICS base 核心程序 iocCore 构成,核心任务是通过设备驱动程序控制 MAXnet 电机控制器,并把相关信息以分布式数据库的方式存储;(3) 设备控制层:选用美国 Oregon Micro Systems 公司的 MAXnet 电机控制器,可实现 5 路步进/伺服电机的异步或同步驱动,不依赖 VME 总线,且提供网口和串口进行通信,适用范围较大并可降低成本。电机驱动器选用 SLS20** 系列。

系统的软件设计分为 OPI 层、IOC 层和设备控制层(图 1,标示黑色部分由本组研制)。通过 EPICS 系统的 Channel Access 协议,IOC 响应来自 OPI、IOC 本身和网络上其他 IOC 的数据请求与控制命令^[2]。

2 系统设计

系统的软件开发环境为 Red Hat Linux 9, base3-14-9, Extension, motor, asyn。Extension 提供了多种开发应用软件,包括人机图形界面开发工具 EDM。Motor 软件包提供了 motor 记录,相关的记录支持、设备支持和一些硬件驱动支持等。asyn 用于建立网口设备支持模块,实现 IOC 中的数据库与电机控制器 MAXnet 间的通信。

2.1 OPI 控制台软件

OPI 由 EPICS 提供的 EDM 进行开发,可显示任意 record 中任何感兴趣的域,只需在界面上选定一小块,输入域的信息即可实现,图 2 为操作界面的一个示例。

2.2 IOC 系统

IOC 系统的软件包括 Channel Access、Database Access、IOC Database、Motor Record Support Module、MAXnet device & driver Module 和 IP Port Support Module,其中 IOC Database 须在现有记录的基础上根据系统需要改动,MAXnet device & driver Module 须自行编写,IP Port Support Module 借用 asyn 软件包来实现,其他部分则基本无需改动。

Motor 软件包提供了 motor 记录,并定义了多个域,涵盖了电机控制中的主要信息。最常见的为

第一作者:黄 辉,男,1986 年出生,2009 年毕业于上海交通大学,专业:信号与信息处理

收稿日期:2012-03-02,修回日期:2012-04-05

VELO(运行速度)、VMAX(最大速度)、VBAS(起跳速度)、ACCL(加速度)、DIR(转动方向)等^[3]。选择

所需的域,并赋予特定值,启动 IOC 时载入记录,便完成了 IOC Database 的工作。

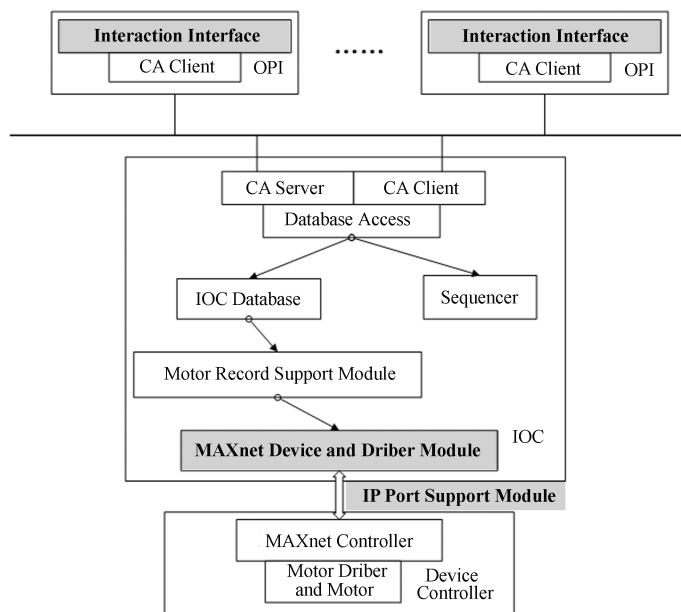


图1 系统软件设计图

Fig.1 Overall design of system software.

Motor Test				
SLIT				
Motor1	Motor2	Motor3	Motor4	Motor5
EGU: Motor1.EGU	EGU: Motor2.EGU	EGU: Motor3.EGU	EGU: Motor4.EGU	EGU: Motor5.EGU
feedback: Motor1.RBV	feedback: Motor2.RBV	feedback: Motor3.RBV	feedback: Motor4.RBV	feedback: Motor5.RBV
Position Motor1.VAL	Position Motor2.VAL	Position Motor3.VAL	Position Motor4.VAL	Position Motor5.VAL
Tweak Motor1.TW	Tweak Motor2.TW	Tweak Motor3.TW	Tweak Motor4.TW	Tweak Motor5.TW
Homing HomR HomF	Homing HomR HomF	Homing HomR HomF	Homing HomR HomF	Homing HomR HomF
Hi limit alarm	Hi limit alarm	Hi limit alarm	Hi limit alarm	Hi limit alarm
HM switch alarm	HM switch alarm	HM switch alarm	HM switch alarm	HM switch alarm
Low limit alarm	Low limit alarm	Low limit alarm	Low limit alarm	Low limit alarm
Moving More	Moving More	Moving More	Moving More	Moving More

图2 OPI 界面示例

Fig.2 Example of OPI.

MAXnet device & driver Module 部分是本设计的重点和难点,其结构图见图3。

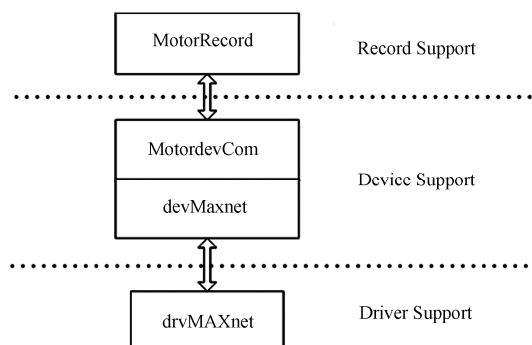


图3 MAXnet 模块结构图

Fig.3 Structure of MAXnet Module.

MotordevCom 是 MotorRecord 调用设备层的通

用接口,根据实际需要调用具体设备的设备支持程序 devMAXnet。devMAXnet 中定义的 motor_dset 类型的数据结构——devOmsMAXnet 是数据库记录和驱动程序之间的接口。

drvMAXnet 的主要工作如下: (1) 在系统中为 MAXnet 的各种状态和运行信息申请存储空间; (2) 根据具体需求设定 MAXnet 的工作模式; (3) 把记录中关于方位、速度等控制信息转换为 MAXnet 的控制指令发送给设备; (4) 监控 MAXnet 运行并返回控制信息到记录,处理各种异常情况。

当状态标记被设置时,MAXnet 会异步发送标记以通知主机。标记通知字符串的格式为“%000 SSSSSSSS”,其中“%”意味着该串为数字流,“000”表示该数字流包含状态标记信息,八位 ASCII 码代

表状态信息。由于此特征,需在 drvMAXnet 中的接收函数中增加对命令解析的处理。

drvMAXnet 的程序架构参照与 MAXnet 同为 OMS(Oregon Micro Systems)公司生产的 MAXv 和 PC68 电机控制器的驱动代码。由于 PC68 使用串口通信,MAXv 与 MAXnet 的指令最为接近,因此,PC68 的数据处理方式与 MAXnet 的指令翻译过程对 drvMAXnet 的开发有很大的参考价值。

MAXnet 的设备应用程序使用了网口,需要 IP 设备的驱动支持模块。EPICS 中的 asyn 软件包提供的 asynDriver(asynchronous driver support)可用于被控设备与低端驱动层的接口连接,使设备支持模块的工作变得更容易^[4]。此外,asyn 软件包中提供的 asyncomRecord 可方便地用来调试网口^[5]。

在一台 Linux 主机上完成了安装 EPICS base 包和 asyn 软件包、编译 MAXnet device & driver 程序和定义 motor 记录后,启动 IOC 还需要编写适当的启动脚本。

经上述步骤,主机上建立 IOC 的工作基本完成,本设计的目标是在嵌入式目标机上运行 IOC 和电机控制程序。首先在一台 Linux 主机上安装 Linux-arm 的交叉编译器,对将要下载到 ARM 开发板上的内核进行裁剪,减小占用空间,编译内核后下载到开发板中。用主机上成功安装的 EPICS base 包以及

motor、asyn 等支持包交叉编译出开发板上可运行的 IOC 程序,在这个过程中,用 NFS 方式 mount 主机的根文件系统会对调试程序提供很大的便利,省去很多下载工作。程序调试成功后通过 ftp 方式把程序、数据库和启动脚本下载到目标机,修改开发板的开机配置文件使之自动运行 IOC,完成了 ARM 开发板上 IOC 的开发。

2.3 电机设备控制部分

电机部分由 MAXnet 电机控制器、电机驱动器和电机组成。MAXnet 电机控制器(美国 Oregon Micro Systems 公司)是高性能电机控制器,与 MAXv 系列功能类似。MAXnet 无 VME 接口,而代之以串口和 Ethernet 接口,本文通过网口连接 ARM 开发板和 MAXnet 电机控制器。MAXnet 可同时控制 5 轴电机的同步或异步运动,对于步进和伺服电机也同样适用。

控制开环的步进电机连接线见图 4,基本控制方式为“方向+转速”。Direction 控制电机的转动方向,Step 产生 50% 占空比的脉冲,脉冲个数决定电机转过的角度。脉冲速度为 0~4,199,303/s,加速度为 0~8,000,000/s²^[6],选择合适的脉冲速度和加速度可控制电机的运行速度、起跳速度和加速度。

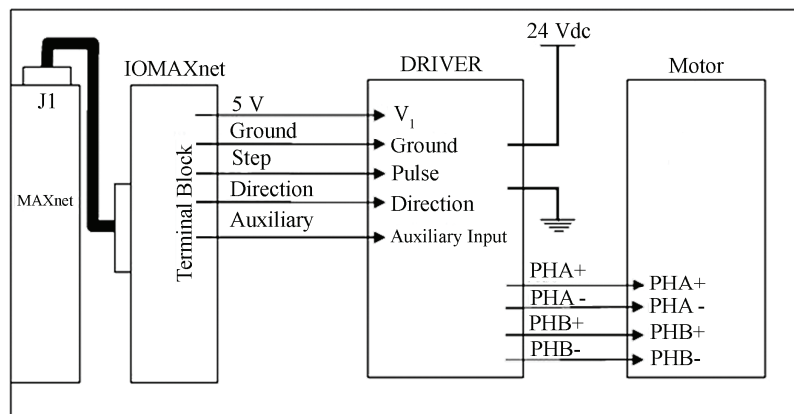


图 4 MAXnet 与步进电机的接线图^[6]
Fig.4 The Wiring Diagram between MAXnet and motor.

3 系统性能测试

目前上海光源电机控制系统的使用方式有单电机运动控制和多轴电机同步运动控制。单电机运动,其运动性能与操作系统的反应时间和 IOC 与控制器之间的通信时间无关,且 MAXnet 控制器与 MAXv 控制器命令集及参数指标基本一致。因此,本设计

可满足简单的单电机运动控制的使用要求。

对于多轴电机同步运动控制,我们用 Soft Motor 与 transform 记录结合触发三路电机同步运动进行了测试,规定延迟时间为第一路与第三路电机开始运动之间的时间差,同步误差为运动过程中第一路与第三路电机之间的最大距离差,光源现有电机系统的延迟时间~23 ms。本设计的延迟时间为~100 ms,当电机速度为 150 $\mu\text{m/s}$ 时,新系统的同

步误差达 15 μm , 为光源现有系统的 4 倍。新系统中的 S3C2440 计算机升级为性能更高的 PC 机时, 系统的延迟时间能控制在 12 ms 左右, 仅为光源现

有电机系统延迟时间的一半, 产生的同步误差也将为光源现有系统同步误差的一半。表 1 为三种电机系统的软硬件及性能比较。

表 1 系统参数及性能
Table 1 System parameters and performance.

系统 System	MVME5500(vxworks)+MAXv	S3C2440(linux)+MAXnet	PC(linux)+MAXnet
总线类型 Bus type	VME	IP	IP
总线速度 Bus speed / $\text{Mb}\cdot\text{s}^{-1}$	160	100	100
CPU 频率 CPU frequency / MHz	10^3	406	2.4×10^3
内存 Memory / MB	512	64	2×10^3
操作系统最大时延	50	无法保证	无法保证
Max delay of OS / μs		Unable to determine	Unable to determine
延迟时间 Delay / ms	23	100	12

由运行 Linux 非实时操作系统的 PC 机组成的电机控制系统性能上优于光源现有电机系统。相比总线速度和操作系统实时性, 计算机的运算能力对延迟时间的影响更大。实时操作系统带来的优势并不明显。

本次设计表明基于 ARM 处理器的电机控制系统的可行性, 运行 Linux 的 S3C2440 计算机与 MAXnet 组成的电机系统在时间响应性能上与光源现有电机系统有一定差距, 原因在于使用 ARM9 内核的 S3C2440 计算机是嵌入式开发的入门级产品, 性能有限。将 S3C2440(linux)+MAXnet 的电机控制系统的计算机升级为 PC 机, 系统的时间响应性能会有较大提升。若使用运算能力较强的 ARM 处理器的计算机, 时间响应性能可得到很大提升。

4 结语

本文介绍了 EPICS 下基于 ARM 处理器的电机控制系统, 软硬件设计与系统调试均已完成, 证明了基于 ARM 处理器的电机控制系统的可行性。本系统在位置控制可满足使用要求, 在时间响应性能上存在一定差距, 若使用高性能处理器, 时间响应性能可得到很大提升。

使用 S3C2440(linux)+MAXnet 方案将电机控制系统的成本降低了一个数量级, 而且省去 VME 机箱, 系统体积大大减小。本次设计是对电机控制系统的一种新尝试, 为上海光源的电机控制系统设计提供了一种可供参考的思路。

参考文献

- 1 闫芬, 李爱国, 杨科, 等. 基于 EPICS 的硬 X 微聚焦试验站数据采集系统[J]. 核技术, 2009, 32(11): 801-805
YAN Fen, LI Aiguo, YANG ke, *et al.* EPICS-based data acquisition system of hard X-ray micro focus station[J]. Nucl Tech, 2009, 32(11): 801-805
- 2 Kraimer M R. EPICS input/output controller(IOC) application developer's guide[OL], Release 3.14.8, 2005, <http://www.aps.anl.gov/epics/base/R3-14/8.php>
- 3 Tim Mooney, Joe Sullivan, Ron Sluiter. Motor record and related software[OL]. 2003, <http://www.Aps.anl.gov/bcda/synApps/motor/R6-2/motorRecord.html>
- 4 Marty Kraimer, Eric Norum, Mark Rivers, *et al.* asynDriver: asynchronous driver support[OL], Release 4.5, 2006, <http://www.aps.anl.gov/epics/modules/soft/asyn/R4-7/asynDriver.html>
- 5 Mark Rivers, Marty Kraimer. Asyn record, release 4-7[OL], 2004, <http://www.aps.anl.gov/epics/modules/soft/asyn/R4-7/asynRecord.html>
- 6 User's manual intelligent motion controller for Ethernet MAXnet[OL]. 2010, <http://www.pro-dex.com/motion-control/product-catalog/motion-controllers/ethernet/maxnet-motion-controller.aspx>

An ARM-based motor control system in EPICS

HUANG Hui HUANG Yuefeng LI Yongping YIN Congcong JIA Wenhong

(Shanghai Institute of Applied Physics, Chinese Academy of Science, Shanghai 201800, China)

Abstract In this paper, a motor control system, formed by a S3C2440 computer, the Linux operating system and a MAXnet motor controller, is developed in EPICS at SSRF (Shanghai Synchrotron Radiation Facility). A detailed analysis of the system design, and software development, are provided. It indicates that an ARM-based motor control system is feasible. Test results show that it has an equivalent performance in position control, with greatly reduced cost and volume, though the time response is not as good as the present motor control system. It provides a reference for designing new motor control systems of SSRF.

Key words Motor control, ARM, EPICS, SSRF

CLC TP273.5