

# 嵌入式可编程逻辑控制器中加挂 惠普图形语言的设计

孙丹枫, 严 义

(杭州电子科技大学 计算机学院, 杭州 310018)

**摘要:** 为解决在 PLC 运动控制开发中难以融入 CAD 技术的问题, 提出在 ePLC 中加挂 HPGL 的设计; 通过 HPGL 指令到功能块的归一化, 目标文件的压缩存储, 层次化的内存分配, 不同下载模式设置实现加挂的接口设计; 采用三级线程抢占式调度算法, 完成 HPGL 程序执行控制算法、功能块程序和紧急任务的线程分配; 对直线、圆、矩形、螺旋线和非规则五种轨迹进行试验, 试验结果均良好拟合 CAD 中轨迹, 验证了该设计在直角坐标机器人控制中的可行性。

**关键词:** ePLC; HPGL; 运动控制; 直角坐标机器人

## Design of Integrating HPGL into ePLC

Sun Danfeng, Yan Yi

(College of Computer, Hangzhou Dianzi University, Hangzhou 310018, China)

**Abstract:** The design solved the problem of difficult to bring CAD technology into PLC motion control. Interface design, as the focus of the implementation, included HPGL instruction normalization, compression storage of the target file, hierarchical memory allocation, different download mode set. HPGL program execution control algorithm, function block and urgent task allocation were allocated in different levels. Tertiary preemptive scheduling algorithm was adopted to complete the task scheduling. After testing straight line, circle, rectangle, helix and the rules track, the results well fit trajectory in CAD. Verify the feasibility of the design in rectangular coordinate robot control.

**Keywords:** ePLC; HPGL; motion control; Cartesian robot

## 0 引言

CAD 软件广泛应用于建筑工程、装饰设计、工业制图、服装加工及电子工业等领域。CAD 软件如 AutoCAD、Corel-Draw 等, 通过 DWG, DWF, PLT (HPGL 文件格式) 等格式文件进行通用设计。在加工行业, 如雕刻、服装加工、玻璃切割等, 控制器采用基于 PC 的运动控制卡或专用控制器, 软件则根据控制器配置专用软件<sup>[1]</sup>。由于 PC 的运动控制卡价格贵、易死机, 而专用控制器不能进行二次开发, 不具有通用性, 存在功能单一, 界面不友好, 开发周期长, 软件适应差等缺点。基于 PLC 形式的运动控制模块具有高效、稳定、通用强的特点, 但由于通用 PLC 的硬件与其编程软件无法对接 CAD 的 HPGL 语言。本文提出 ePLC 上加挂 HPGL 的方法, 实现通用 ePLC 对 CAD 软件的支持, 可被广泛应用于各种数控加工设备中。

## 1 系统概述

ePLC 指嵌入式 PLC (embedded & extensible PLC)。ePLC 可任意扩展功能模块, 包括运动控制模块、RFID 模块、AD/DA 模块、视觉模块及网络模块等, 以适应不同应用领域

的需求。CASS 平台支持 ePLC 开发。CASS 平台开发方式符合 PLCopen 组织制定 IEC61131-3<sup>[2]</sup> 标准, 可支持跨硬件平台设计, 可内嵌智能算法。CASS 平台和 ePLC 的开放性设计为实现 CAD 软件的 HPGL 语言引入打下基础。

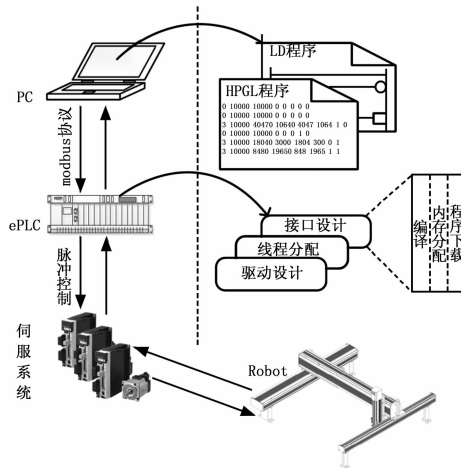


图1 系统结构图

ePLC 上加挂 HPGL 设计的系统结构如图 1 所示。通过 CAD 软件设计获取运动轨迹的 HPGL 语言指令, 采用惠普公司设计的 PLT 文件作为轨迹信息的载体, 通过串口与 ePLC 通信。PC 端的 CASS 平台实现 LD 程序设计, 完成基本运动指令。ePLC 中通过接口设计、线程分配和驱动设计 3 部分实

收稿日期: 2014-01-02; 修回日期: 2014-03-21。

**基金项目:** 国家自然科学基金 (61272189); 国家 863 计划 (2013AA040301); 浙江省重点科技创新团队 (2010R50008)。

**作者简介:** 孙丹枫 (1988-), 男, 浙江省临安人, 工学硕士, 主要从事运动控制技术、机器人技术等方向的研究。

现 HPGL 加挂。

## 2 接口设计

### 2.1 编译

编译过程即将 HPGL 程序转化为与功能块程序一一对应的目标文件。主要分为两部分：HPGL 指令到功能块的归一化处理和 HPGL 语句到标准帧格式的转化。

PLCopen 组织制定 Motion Control 编程标准<sup>[3-4]</sup>中定义的主要运动控制功能块如表 1 所示。功能块主要分为运动类和管理类。表中为运动类功能块，每个功能块唯一对应运动控制算法。功能块的功能和名字相符。如 MC\_Home 表示单轴回原点。

表 1 功能块

SingleAxis	Coordinated
MC_Home	MC_GroupHome
MC_Stop	MC_GroupStop
MC_Halt	MC_GroupHalt
MC_MoveAbsolute	MC_MoveLinearAbsolute
MC_MoveRelative	MC_MoveLinearRelative
MC_MoveAdditive	MC_MoveCircularRelative
MC_MoveSuperimposed	MC_MoveCircularAbsolute

惠普公司制定的 HPGL 指令标准可分为 5 个功能组<sup>[5]</sup>：配置及状态组、向量组、多边形组、直线及填充属性组以及字符组等。研究用到的主要是其中的 16 条，分为运动控制指令（如 PA 等）和配置指令（如 VS 等），如表 2 所示。

表 2 HPGL 指令

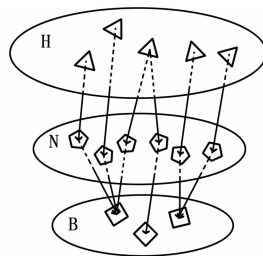
指令	意义	指令	意义
PA	绝对绘图	AR	相对画弧
PR	相对绘图	CI	画圆
PE	折线编码	AT	绝对三点画弧
PU	提笔	RT	相对三点画弧
PD	落笔	EP	多边形描边
EA	绝对矩形描边	IN	初始化
ER	相对矩形描边	VS	速度选择
AA	绝对画弧	SP	选择画笔

指令格式采用指令名与参数的结合方式，如 PA 100, 100 表示从当前位置绝对直线绘图到 (100, 100) 位置。

在实际应用中，功能块和 HPGL 指令之间存在多对多的映射关系。例如 PA 指令，只沿着 X 或者 Y 轴方向绘图时，对应于功能块的绝对单轴运动 MC\_MoveAbsolute，否则对应多轴联动模块 MC\_MoveLinearAbsolute；另一方面 PU, PD 指令都对应 MC\_MoveAbsolute 功能块。为消除两者之间的多对多映射，在 HPGL 指令编译之前需要做归一化处理。实现过程如图 2 所示。

归一化方法定义如下：

定义：设 HPGL 指令集合为  $H$ ,  $Card(H) = \alpha$ , 功能块集合为  $B$ ,  $Card(B) = \beta$ , 在两个集合之间增加一个归一化集合  $N$ . 集合  $N$  的元素产生遵循以下定理：遍历集合  $H = \{h_0, h_1, \dots, h_\alpha\}$ , 如果  $\exists \{b_0, b_1, \dots, b_i\} \in B, 0 \leq i \leq \beta$ , 使得  $\forall 0 \leq j \leq i, \exists (h_k = f_1(b_j)) \in H$ , 则  $N = N \cup \{f_2(h_k,$



△HPGL指令 ◻归一化映像 ◻功能块

图 2 归一化

$b_1), f_2(h_k, b_2) \dots f_2(h_k, b_i)\}$

其中： $f_1$  为  $B$  到  $H$  映射函数， $f_2$  是一个依据变量  $h$  和  $b$  组帧过程。

编译后的目标文件是以标准帧格式保存，需要经过 HPGL 语句到标准帧格式的转化。考虑到嵌入式芯片内存资源有限的问题，目标代码采用压缩存储的方式，即目标代码中每条指令数据按实际大小紧密存储，不考虑不同指令，长度不同的问题，每条指令内部参数顺序按长度降序排列。为实现程序的执行，引擎中有每条指令的指令结构表，在程序执行时，引擎根据表中指令格式，对目标文件进行再解析执行。

标准帧分为指令码和参数两部分。指令码使用归一化方法确定，它唯一对应运动控制功能块。参数部分为 4 字节参数、2 字节参数及 1 字节参数，依次排列。参数内容为控制参数，如速度、距离、加减速等。

### 2.2 内存分配

整个软件系统由 3 部分构成：HPGL 程序、LD 程序以及引擎。为完成相互之间的协调以及确保数据间的安全性，需要对内存进行特别设计。如图 3 所示。内存分配包括 FLASH 和 RAM 的分配，采用层次化的内存分配方式<sup>[6]</sup>。

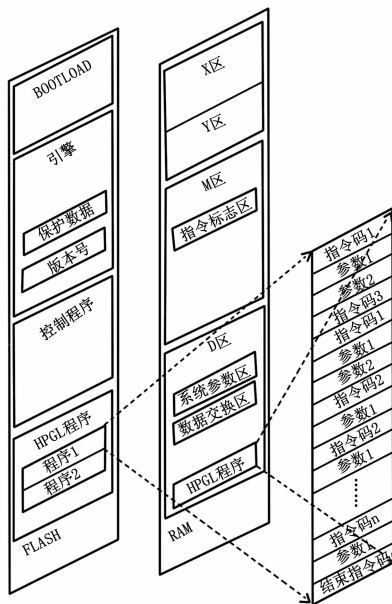


图 3 ePLC 中内存分配图

FLASH 部分存储永久保存的数据, 各部分介绍如下:

1) BOOTLOAD: 引导程序。

2) 引擎: ePLC 完成控制的主体, 包括线程调度算法、逻辑控制、HPGL 程序执行控制算法以及运动控制算法(梯形 S 形速度曲线、直线插补、圆弧插补、电子凸轮等)。引擎中还包含版本号和保护数据区域。版本号用于维护引擎版本, 保护数据区用于保护 RAM 中用户自定义的重要数据。

3) 控制程序: 为梯形图功能块程序。引擎通过线程调度选择合理线程对该区域进行定时扫描, 以执行梯形图程序。

4) HPGL 程序: 存放 HPGL 程序, 可以同时存放多个 HPGL 程序。程序的存放地址和执行程序的选择可在 HPGL 程序编译器中完成。

RAM 区以软元件的形式划分, 包括输入(X区)继电器、输出(Y区)继电器、辅助继电器(M区)、数据寄存器(D区)等。软元件还可根据需求进行可定制划分。为满足课题在 ePLC 中加挂 HPGL 的需要, 在 M 区和 D 区开辟系统专用区: 指令标志区、系统参数区、数据交换区和 HPGL 程序存储区。

1) 指令标志区: 存放 HPGL 指令在梯形图中对应功能块的执行标志。

2) 系统参数区: 存放引擎中运动控制算法所需参数, 如速度, 距离, 加速度, 减速度, 起始速度等。

3) 数据交换区: 为实现 HPGL 程序、梯形图功能块程序以及引擎之间的交互, 并且不破坏各自程序之间的封装性, 系统引入了数据交换区。在 HPGL 程序运行中, 该区域主要存放从 HPGL 程序区中取出的当前需要执行的指令。

4) HPGL 程序存储区: 临时存放 HPGL 程序。

### 2.3 程序下载

为保护 FLASH, 减少其读写次数, 将 HPGL 程序的下载方式分为调试模式和发布模式。调试模式下, HPGL 程序编译器将程序直接下载到 RAM 中, 不对 FLASH 进行写入。发布模式下, HPGL 程序根据编译器中的下载地址下载到 FLASH 中, ePLC 每次重启时会从 FLASH 中读取默认的 HPGL 程序(默认值可在 HPGL 程序编译器中设置)到 RAM 执行。

## 3 线程分配

引擎采用三级线程抢占式调度算法。线程根据优先级从高到低分为中断线程、快速线程和慢速线程, 如图 4 所示。中断线程执行即时响应任务, 快速线程执行快速响应任务, 慢速线程执行低速响应任务。抢占事件发生时, 保存寄存器数据和当前程序指针到系统参数区, 完成线程切换。由于 ePLC 的 RAM 区为所有线程共享, 需要实现数据互斥。为读写数据设置读写锁。当锁打开时, 关闭中断响应。当锁关闭时, 打开中断响应。

HPGL 相关线程分配为功能块程序、HPGL 程序执行控制算法以及紧急任务(包括错误以及紧急停止发生时的处理等)。HPGL 程序较大, 且其响应时间较长, 将 HPGL 程序控制算法放到慢速线程中执行。功能块程序有响应速度要求, 在快速线程中执行。而紧急任务具有即时性, 在中断线程中执行。

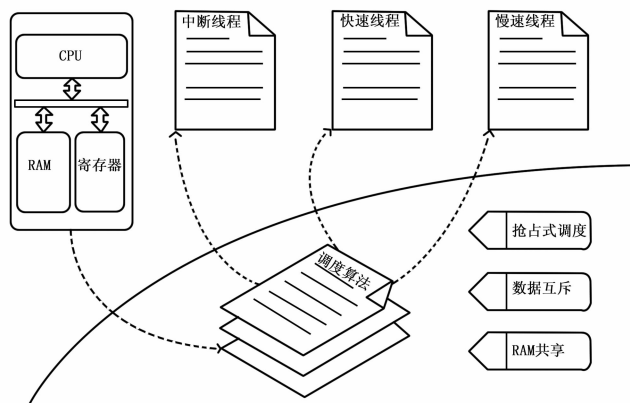


图 4 三级线程抢占式调度

## 4 驱动设计

### 4.1 HPGL 指令结构表的设计

RAM 中可执行程序为压缩存储数据, 为识别 HPGL 指令, 在引擎中增加了 HPGL 指令结构表。

HPGL 指令结构表是一种专用结构。表中的每一条数据对应一种 HPGL 指令。指令结构表的内容为指令码, 4 字节参数数, 2 字节参数数, 1 字节参数数。

HPGL 指令数量不多, 且每次对表的访问是随机的, 故采用二维数组的存储方式, 假设指令条数为  $n$ , 则指令结构表的 C 语言描述为:

```
-----
Unsigned short
InstructionTable[n][4]={
    {0,i04,i02,i01}
    {1,i14,i12,i11},
    ...,
    {n,in4,in2,in1}
};
-----
```

### 4.2 功能块程序

功能块程序将数据交换区数据传递到系统参数区, 调用运动控制算法完成运动控制。

功能块程序中, 由 M 软元件控制每段功能块程序。M 软元件和 HPGL 指令码一一对应。例如让 M2000 对应指令码 00H, M2001 对应指令码 01H, 由于 M 软元件后的地址为 8 进制表示, 故数量超过 8 个时 M 软元件地址要进位, 即 M2010 对应指令码 08H, 以此类推直到实现所有指令。

### 4.3 HPGL 程序执行控制算法

HPGL 程序执行控制算法主要分为两步:

1) 取出一条 HPGL 指令。从 RAM 中 HPGL 程序区读取一条指令存入数据交换区, 取出指令为结束指令则 HPGL 程序执行完毕, 否则将 HPGL 程序指针指向下一条。取指令需要知道指令开始的起始地址以及指令长度, 指令开始地址为 HPGL 程序的起始地址(运行后为执行完的指令的偏移)。长度的计算需要查找指令结构表。假设依据当前指令码查得该指令 4 字节参数数为  $N_4$ , 2 字节参数数为  $N_2$ , 1 字节参数数为

$N1$ ，则指令总长度  $InstructionLen$  为：

$$InstructionLen = N4 \times 4 + N2 \times 2 + N1$$

2) 开启该条指令。开启指令的过程即根据指令码将功能块控制位打开。例如当前指令码是 00H，则将 M2000 置成 1。

### 4.4 HPGL 程序执行过程

HPGL 程序的执行过程由引擎控制，其执行过程如图 5 所示。ePLC 上电，引擎先做初始化操作，参数初始化，RAM 区清零，恢复掉电保护数据。初始化成功后，进入主循环执行三级线程调度模块。一旦检测到开始执行 HPGL 程序信号（外部输入，或者其他内部条件满足），则执行 HPGL 程序执行控制算法，将一条指令存入数据交换区，开启该条指令。梯形图程序等到下一个扫描周期到时，执行相应运动控制功能块，从数据交互区读取数据，将该数据交给引擎算法参数区并开始执行相应引擎算法，如此循环进行，直到 HPGL 程序执行到结束指令。

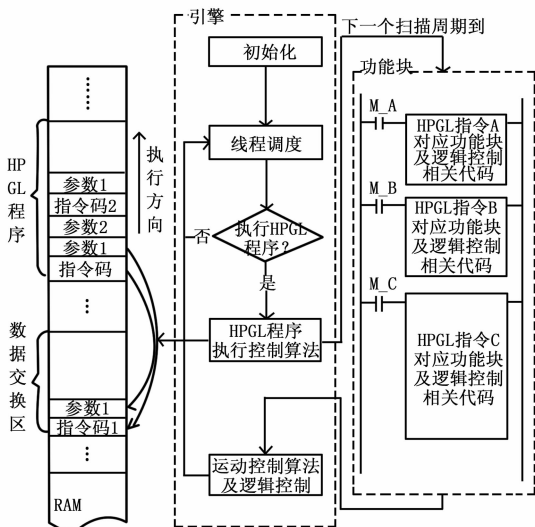


图 5 HPGL 程序执行过程

### 5 功能验证

加挂 HPGL 指令后的 ePLC 对龙门式直角坐标机器人的控制实例如图 6 (a) 所示。



图 6 直角坐标机器人和 HPGL 程序编译器

动力部分使用富士的通用伺服系统。HPGL 的加挂设计在 CASS-ePLC70A 型 ePLC 上完成。该 ePLC 采用 DSP+ARM 双核芯片，具有位置和速度闭环，可同时控制六路电机，实例中用三路。如图 6 (b) 所示为 HPGL 程序编译器。该编译器使用 C# 开发，具有编译、编辑、运行监控及下载等功能。

实验过程中，选取直线、矩形、圆、螺旋线以及随意绘制的非规则轨迹分组试验，将 CAD 中的绘制轨迹与运行轨迹进行对比。

CAD 软件中生成 HPGL 程序前的非规则轨迹和实际画笔绘制的实物对比图如图 7 所示。图 7 (a) 是 CAD 中的绘制轨迹，图 7 (b) 是直角坐标机器人运行轨迹。实验结果显示运行轨迹良好拟合了 CAD 中的绘制轨迹，验证了 PLC 中融入 CAD 技术的可行性。

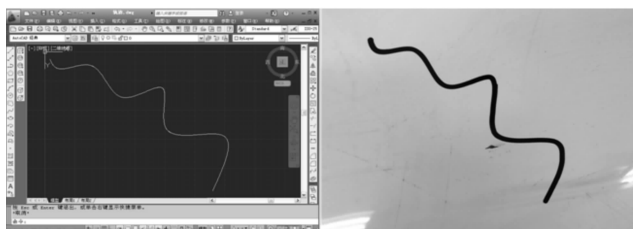


图 7 CAD 中绘制轨迹于实物对比图

### 6 结束语

本文提出在 ePLC 中加挂 HPGL 的方法，实现对直角坐标机器人可视化在线编程；归一化方法简化 HPGL 指令到功能块的映射关系，多层次的内存分配确保了各部分数据的独立性和安全性，HPGL 指令的压缩存储以及引擎中指令结构表的设计降低了内存需求，三级线程抢占调度算法很好地协调了 HPGL 加挂设计中各部分程序执行；这些方法为开发 PLC 更强的功能提供了较好的范例，开发的系统可广泛应用在各种带 CAD 设计软件的数控设备中。

#### 参考文献：

- [1] 李建奇, 肖继国, 贺盛修, 等. 基于 NURBS 插补算法的数控雕刻机控制系统的设计与实现 [J]. 计算机测量与控制, 2012, 20 (12): 3249 - 3252.
- [2] 彭 瑜. 工控编程语言 IEC61131-3 和 PLCopen 组织 [J]. 电气时代, 2006, 3 (5): 32 - 36.
- [3] Technical Specification PLC open - Technical Committee 2, Task Force Function blocks for motion control [M]. 2011.
- [4] PLC open Technical Committee 2, Task Force Function Blocks for motion control: Part 4 - Coordinated Motion [M]. 2008.
- [5] Packard H. The HP-GL/2 and HP RTL Reference Guide [M]. Massachusetts: Addison-Wesley, 1996.
- [6] 董 禄, 严 义, 邬惠峰. 基于 PLC 多层架构构建数控语言的方法与实现 [J]. 机电工程, 2012, 07: 856 - 860.

机器人由三组线性模组相互垂直组成 XYZ 直角坐标系。