# 通用运动控制器软件架构设计需求

## 通用运动控制器

目前硬件平台为arm cortex-A这样的微处理器平台，计划采用通用软件的设计方案。未来更高性能，多核处理器将更广泛的运用到工业自动化中来，通用的应用软件设计会在开发和维护上带来更高效率的好处。

在实时控制领域中要做到能够使用通用应用软件开发方式进行开发的前提条件就是分层抽象，屏蔽与隔离。确保应用设计与硬件代码无关。较典型的代表为Codesys 、Linuxcnc、openil、orocos。相对codesys,Linuxcnc和openil的架构比较简单。参考Linuxcnc得到如下的架构图。



软实时

硬实时

通用运动控制器软件框架

与硬件无关的可以分为5个部分，每个部分都是单独编译的，当有模块需要跟新的时候，不需要重新编译整个系统，只需要更新对应的模块即可。5个模块分别为：固件、接口抽象、运动控制、译码层、界面。

### 硬件平台固件

平台固件是针对特定的主板开发底层软件，首先平台固件需要确认硬件能够正常和稳定运行，其次为应用程序提供调试和开发和运行环境。当前主流的RTOS（VxWorks、qnx、xenomai等）都提供诸如GDB、telnet、tftp等较为丰富的远程调式手段，装载了标准固件的硬件可以通过网络的形式连接到服务器，此时设备是服务器上的一个标准节点。上层应用开发人员可以通过服务器的方式连接到硬件进行开发、调试、和部署，很好的屏蔽了硬件的特殊性。使得调试更加方便和高效。标准固件设备远端调试部署如下图所示：



平台固件程序主要由主板的引导、操作系统及相关组件和根文件系统组成，可以结合实际的需求来选择不同的实施方案，主流的固件程序组成部分如下图所示：



### RTAPI

用于隔离操作系统，分离出实时接口。运动控制系统统一使用该映射过的接口进行系统访问。上层的代码与具体的实施操作系统无关，这允许开发者可以根据实际情况来选择或移植到合适的实时操作系统平台。

### 硬件抽象层HAL

HAL 硬件抽象层。经过HAL抽象过的模块具有可被加载和可互连的特性。在顶层，这只是一种将多个构建的代码块装配一个复杂的系统的方式。HAL最初的设计是为了在各种各样的硬件设备下更容易配置cnc系统。对于硬件部分，这些代码块是硬件设备的驱动，然而HAL不仅仅是可用来配置硬件驱动，还可以用来做许多事情。

任何系统（包括CNC机床）都由相互连接的组件组成。对于cnc设备，这些组件可能是主控制器，伺服驱动、步进电机、马达、编码器、限位开关、吊坠按钮、也有可能是主轴驱动的VFD，PLC运行换刀装置等等，机械制造商必须选择，安装这些零件并将它们连接在一起，以构成一个完整的系统。

HAL基于与设计硬件电路和系统相同的原理，每个HAL组件都是一个具有明确定义的输入，输出和行为的软件，可以根据需要进行安装和互连。使得设备软件设计能够像硬件设计一样，通过选择、安装并将它们连接在一起，构成一个完整的系统。使用HAL规则封装的步进电机模块如下图所示：



步进电机的HAL模块

HAL强调的是模块独立，每一个功能可以被单独的编译成可加载模块，这样可以有效的减少模块间的耦合性，上层应用不需要拿到功能模块的具体实现代码整合到自己的代码空间进行联合编译，只需要模块接口的描述文件即可对模块进行访问。如果对应的功能模块需要更新，数据访问者不需要重新编译自己的代码。只需要在系统中重新载入发生更新的模块即可。简化了更新维护过程。调用者与功能模块的开发关系如下图所示：



### 运动控制层

运动控制层分为EMCTask（任务执行器），EMCMOT（运动控制模块）、EMCIO（IO控制模块）和译码四个部分。

基本上，EMCTask是一个协调模块，位于GUI之下以及EMC层次结构中的EMCMOT和EMCIO之上。 它通过GUI从操作员那里接收命令，监视下级模块（EMCMOT和EMCIO）的状态，根据当前情况分析命令，并在EMCTask自身内部调用函数或将命令分派到其下级模块（EMCMOT或EMCIO） ）。 在EMC的四个模块中，只有EMCMOT是实时模块。 其他三个，即 EMCTask，EMCMOT和译码是非实时模块。 非实时模块之间的通信是通过NML通道实现的，实时模块（EMCMOT）和非实时模块（EMCTask）之间的通信是通过共享内存实现的。

EMCMOT部分是硬实时模块，实现对路径规划、正逆姿态算法的计算。这个过程包括各个轴的位置取样，运动插补，运动规划等。算法开发者可以使用HAL接口针对不同应用场景（5axis、scara、puma)灵活的封装出不同的运动学组件。得益于HAL的可加载与互联性，各个不同的运动学模块很方便的集成到系统中来，而不需要算法开发人员额外关注其他软件部分的功能。

EMCIO部分是控制器控制所有的IO事件，不同的设备的IO设计不同，得益于HAL接口的灵活性，可以方便的针对不同的设备配置不同IO控制逻辑，也可以借助译码层，做软plc逻辑控制。

译码部分是实现对当前设备主流的语言进行译码的功能，如cnc设备使用的G代码，以及机器人语言、以及支持当前主流的脚本编程语言（python、lua）。更广泛的支持不同行业的不同业务逻辑开发。

### 机器会话层

当前的工业自动化应用场景中，都要求设备与设备，设备与服务器，设备与操作终端之间能够建立灵活的通讯机制。通过引入机器会话层，可以让控制器支持当前主流的通讯协议，客户可以根据具体的应用场景选择合适的通讯协议进行开发。

### 实施预览

Linuxcnc提供了一套从业务开发层到末端执行机构开发的整体构建方案，下图展示了该方案从业务、控制到末端的连接关系。

