

用 UML 建模开发嵌入式软件

<http://tech.embedev.net/>

<http://book.embedev.net/>

<http://linux.embedev.net/>

摘要 针对面向对象开发与 C 语言的优缺点，提出一种新的方案，用面向对象的思想与开发工具进行软件的设计，用 C 语言进行编码，给出新方案应用实例，并在此基础上提出几种进一步优化程序性能的方法。新的方案增强了用 C 语言开发系统软件的可理解性、可移植性和稳定性，同时又保持了 C 语言高性能与低二进制代码长度的优点。

关键词 面向对象嵌入式 LIML 统一建模语言

引言

面向对象开发方法无疑是当前最流行的软件开发方法。这归功于面向对象开发的众多优点：可靠性高，所开发的程序更健壮；由于面向对象编程的可重用性，可以在应用程序中大量采用成熟的类库，从而缩短了开发时间；继承和封装使得应用程序的修改带来的影响更加局部化，应用程序更易于维护、更新和升级。另外，UML 建模语言和 Rosc 等 CASE 工具为面向对象的流行也起了很大作用，这些工具允许应用规范的面向对象分析和设计的方法与理论，远离纠缠不清的源代码，使得构建和设计变得更直观、更容易理解与修改，从而大大提高开发效率。

嵌入式系统的应用越来越广泛，嵌入式计算机在数量上远远超过了各种通用计算机。嵌入式软件的开发与 PC 软件的开发相比存在一些特别的地方：①嵌入式软件代码往往要求高质量、高可靠性；②嵌入式软件要求尽可能短的二进制代码长度和数据长度，尽管半导体技术的发展使处理器速度不断提高、片上存储器容量不断增加，但在大多数应用中，存储空间仍然是宝贵的；③嵌入式系统往往存在实时性的要求。这些特别要求使得面向对象开发不太适合嵌入式系统。汇编语言是一种非结构化的语言，对于大型的结构化程序设计已经完全不能胜任了，而 C 语言相比其他高级语言具有明显的性能优势，这些原因使得 C 语言成为嵌入式系统开发的最佳选择。

随着后 PC 时代的到来，嵌入式应用迅速增长，应用的复杂性也急剧增加，C 语言对应

的传统结构化设计方法已不能满足嵌入式软件设计和开发的需要。能不能把面向对象开发与 C 语言的优点结合起来?对这个问题,已经有人提出过一些方法,主要集中在用 C 语言来实现 C++的语法,如用结构来模拟类,用函数指针来表示成员函数。这些方法使得本来语法就已经很复杂的 C 语言更加麻烦,同时没有了 C 语言快速、高效的优点。这里提出一种新的方法:用面向对象方法进行分析和设计,最后直接用 C 语言进行编码。

1 应用实例

绕线机控制系统是一个控制两个电机的单片机控制系统,我们使用本文提出的新开发方法进行这个系统的开发。系统采用瑞萨公司 H8 / 300H 系列的 16 位单片机 H8 / 3062,整个系统硬件结构如图 1 所示。单片机通过 I / O 口与 Flash 芯片、控制面板、扬声器以及电机驱动电路相连,并通过光电编码盘检测直流电机转动的圈数。

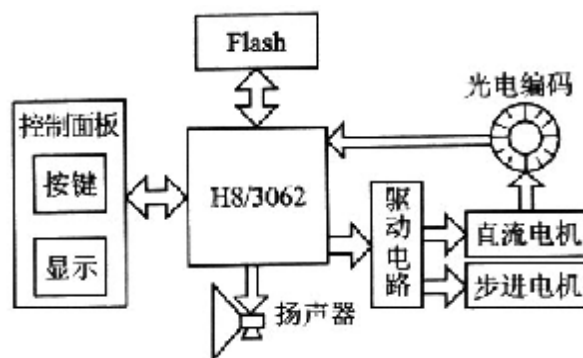


图 1 系统硬件框图

作为面向对象建模的工业标准, UML 几乎被所有面向对象的软件项目所使用。这个项目也使用 UML 来建模,用到的 CASE 工具是 Rose。UML 提供了 5 类图进行不同阶段的建模:用例图、静态图、行为图、交互图、实现图。在开发过程中,可根据不同阶段的具体要求,选择不同的图形来描述系统的静态结构模型或者动态行为模型。一般过程是:①用 UML 的用例模型确定目标系统的主要功能和行为,以便准确、完整地识别系统的需求;②通过对用例模型的分析,从用例的描述中识别反映问题域本质的类和对象,并利用 UML 的类图以及类之间的关系揭示系统的结构和组成;③利用 UML 动态模型中的顺序图、协作图、状态图和活动图描述系统结构元素的动态特性和行为。

1.1 需求过程

系统主要运行过程:①开机时,步进电机自动对原点;②修改参数设置时,单片机把修

改值存入 Flash；③启动键按下时，直流电机开始转动绕线，单片机通过光电编码盘检测转动圈数，并控制步进电机按照设置的参数排线，绕线完成后自动停止；④暂停键按下时，直流电机停止绕线，步进电机排完已转的圈数后停止；⑤复归键按下时，系统重启。

系统的参与者只有用户 1 个，通过对系统需求的分析，可以识别 3 个主要用例：开机、机器设置和绕线控制。通过对用例的进一步分解，得出系统的用例图，如图 2 所示。

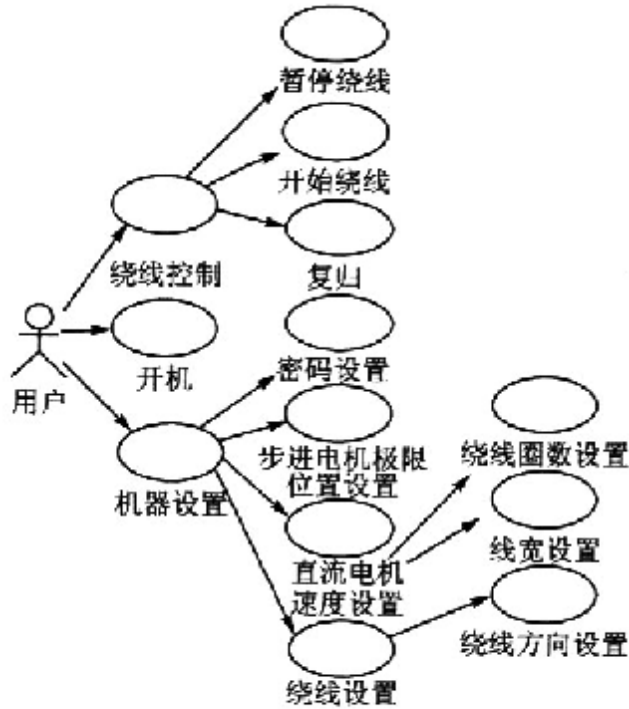


图 2 系统用例图

1.2 设计过程

完成需求分析之后，进入设计阶段。这个阶段中，以需求分析结果为基础，找出系统所需要的类和对象以及其相互关系。针对嵌入式系统的特点，面向对象的一些高级特性不能使用，动态创建对象、派生和多态这样的特性会大大降低系统速度并增大代码空间和数据空间，对瑞萨 H8 / 3062，这款单片机只有 128KB 的 ROM 和 4KB 的 RAM，节省每一个字节都显得非常重要。

通过对用例和系统结构分析，识别出来的类和关系如图 3 所示。

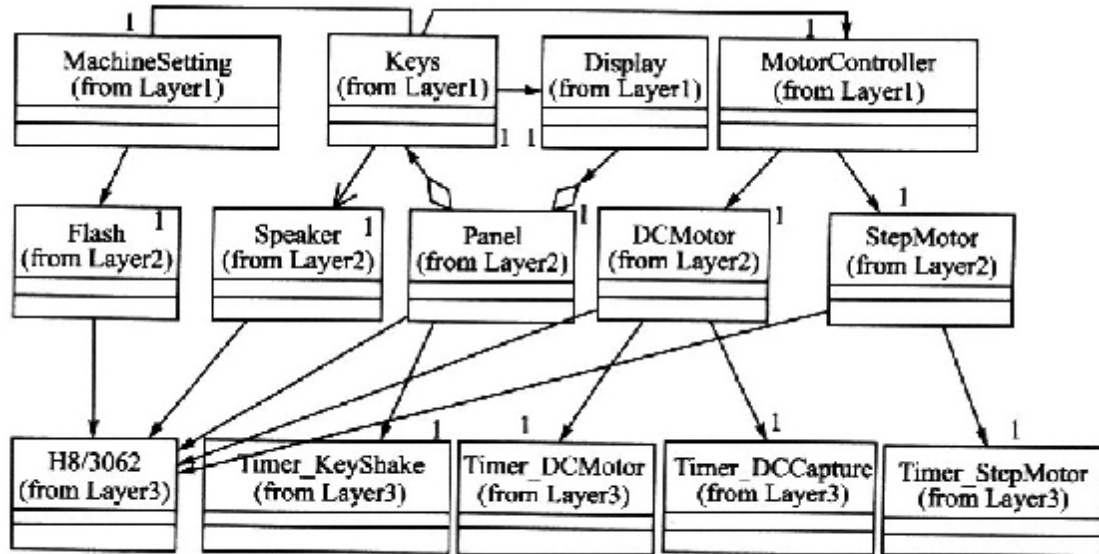


图 3 系统类图

出于可移植性的考虑，所有类被分成 3 个层次：

最顶层是功能逻辑类的抽象，MachineSetting 为机器设置类；Keys 为按键类，Display 为显示类，MotorController 为电机控制器类。这些逻辑上的类与硬件无关，便于以后硬件修改，同时使得开发绕线机的不同版本变得非常容易，如从平绕机到开发环行机，从二轴绕线机到开发三轴绕线机。

中间层是外围硬件类的抽象，Flash 为 Flash 存储类，用于存储设置的参数；Speaker 为扬声器类；Panel 为面板类；DCMoto / 为直流电机类；StepMotor 为步进电机类。

最底层是单片机及其内部功能模块类的抽象，H8 / 3062 为单片机类，用于封装输入 / 输出口；Timer_KeyShake 为按键防抖定时器；Timer_DCMotor 为直流电机 PWM 脉冲产生定时器；Timer_DCCapture 为直流电机圈数捕获定时器；Timer_StepMotor 为步进电机 PWM 脉冲产生定时器。分这一层可以使得变换单片机类型非常容易，瑞萨 H8 / 300H 系列单片机在国内应用非常罕见，之所以选用这款是为了与国外产品的竞争，这样可以更快、更好地打开国际销路，但在选择这款单片机的同时已经做好了移植到 ARM 内核单片机的准备。

为了理解系统的动态行为，还应该用顺序图、协作图、状态图和活动图对系统动态行为

进行描述。状态图对嵌入式系统的设计尤其重要，这里用其对 MotorController 的状态转换进行描述，如图 4 所示。

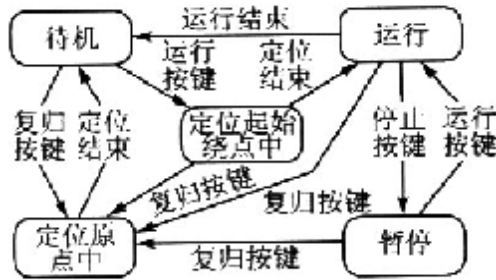


图 4 电机状态图

可以用时序图等对用例进行进一步的分析，“启动”按钮按下后处理的时序图如图 5 所示。

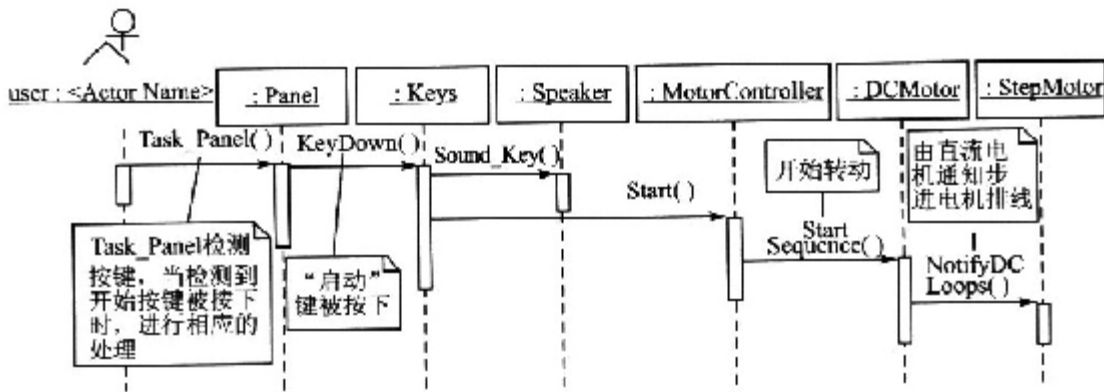


图 5 启动过程的时序图

建模过程不是一个直线过程。它往往是一个演化、迭代的过程，不断地分析、评价、修改、再分析，才可能得到一个高质量的模型，为高质量的软件打下基础。

1.3 编码过程

Rose 具有自动生成 C++ 框架代码的功能。在 Rose 菜单中选择 Tools→ANSI C++ →Generate Code，可以生成 C++ 的框架代码。

要用 C 编码，需要把 C++ 代码稍作修改变成 C 代码。由于没有动态创建实体，所有的对象都是静态的。这使得修改工作极其简单——把“: :”改成“_”即可。修改后的“_”前面的部分可以看成是对象名，后面的部分看成是成员函数名，源文件的修改如图 6 所示。

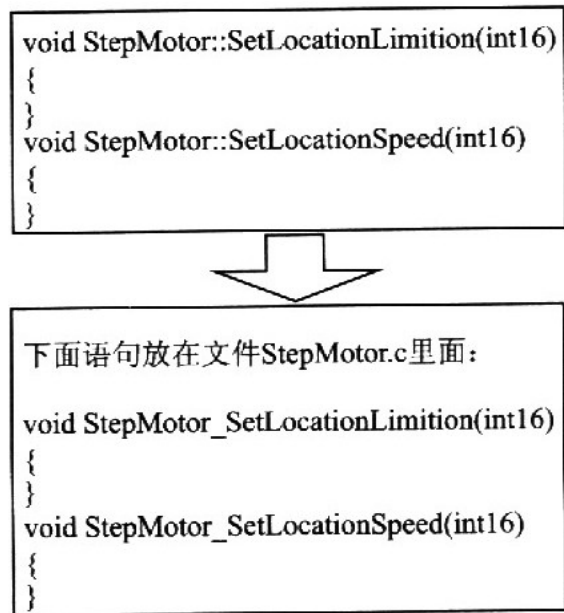


图 6 源文件修改

头文件也需要做相应的修改，对变量的定义需要放到源文件里面，如图 7 所示。

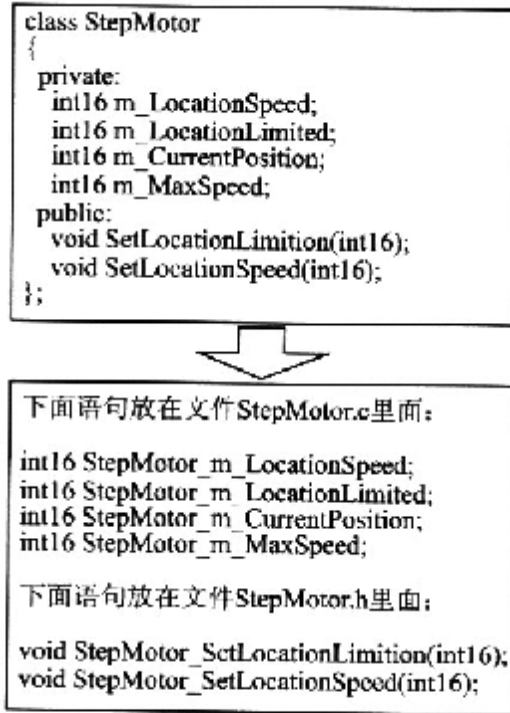


图 7 头文件修改

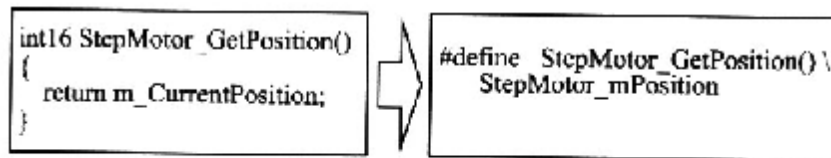


图 8 修改成宏函数

代码修改完成后，剩下的就是 C 语言的编码过程了。编码时，当调用对象成员时，“.” 操作符用“_”代替，则刚好对应上修改后的变量和函数。

2 进一步优化程序的方法

为了进一步提高程序的速度，还应对程序进行专门的优化，例如中断处理程序里面一两条指令引起的差异就非常明显。下面介绍几种优化办法。

①把成员函数定义成宏函数。由于面向对象数据隐藏的特点，对数据的操作需要以函数的形式提供，使得数据的操作效率变低。C++编程时，类的成员函数不能修改成宏，改为 C 函数后，就可以修改了；但这种优化最好在调试结束后进行，因为宏函数不方便错误定位。

修改方法如图 8 所示。

②把成员函数定义成内联函数，如图 9 所示。

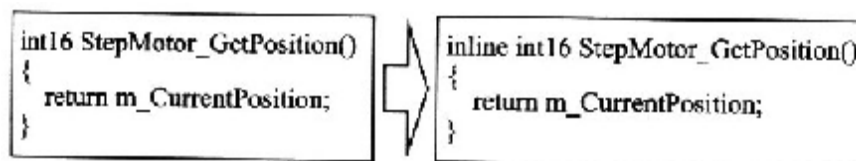


图 9 修改成内联函数

③程序中使用查表。对一些很消耗时间资源的非常复杂的运算，可以使用查表的方式。在步进电机的加减速控制中，加减速曲线一般采用指数曲线或者 S 曲线，运算量较大。可以事先把曲线对应的定时器上界寄存器的值离线计算出来，并片 j 数组保存起来，需要时直接读数组即可。这种方法通过牺牲空间来换取时间，但是牺牲的是相对宽余的 ROM 空间，还是比较划算的。

结语

绕线机项目获得了很好的效果，达到了系统的性能要求。测试时，发现的错误相比以前类似项目大大减少。更让人兴奋的是，测试中没有发现任何大的软件结构上的错误。这归功于 Rose 的前期可视化建模功能。编码之前可以很清楚地掌握系统的结构，这是面向对象设计相比传统方法的优势。

应用结果表明，这种开发方法是很实用的，可以优化程序结构、缩短开发周期，实现快速开发又不降低程序的效率；但这种方法的应用是有局限性的，对效率与空间要求不高的嵌入式系统，完全可以直接用 C++ 编程，因此这种方法只适合于空间紧张、实时性要求比较高的系统。

作者：华中科技大学 郑孝洋 沈安文 陈光东