

第六章 实用程序设计

6.1 程序设计方法

程序设计就是用计算机所能接受的语言把解决问题的步骤描述出来，也就是编制计算机的程序。AVR 单片机程序设计语言有：C 编译高级语言和宏汇编汇编语言。

在设计应用系统时，软件的编制是重要环节。软件的质量直接影响整个系统功能的实现。

本章从应用的角度出发，介绍一些实用子程序，读者既可按需要改编调用，也可以吸收其设计方法，以便更好地设计出适合于自己系统的实用软件。

6.1.1 程序设计步骤

应用程序的设计因系统而异，因人而异。尽管如此，程序设计总是有共同特点及其规律的。在编写程序时，设计人员可以采取如下几个步骤：

(1) 分析问题，明确所要解决问题的要求。将软件分成若干个相对独立的部分。根据功能关系和时序关系，设计出合理的软件总体结构。

(2) 建立正确的数学模型。即根据功能要求，描述出各个输入和输出变量之间的数学关系，并确定采用的计算公式和计算方法。

(3) 制定程序框图。根据所选择的计算方法，制定出运算的步骤和顺序，并画出程序框图。这不仅是程序设计的一个重要组成部分，而且是决定成败的关键部分。

(4) 合理分配系统资源，包括程序 Flash、E2PROM、SRAM、定时器/计数器、中断、堆栈等。确定数据格式，分配好工作单元，进一步将程序框图画成详细的操作流程。

(5) 根据程序的流程图和指令系统，编写出程序。注意在程序的有关位置处写上功能注释，提高程序的可读性。

(6) 程序调试。通过编辑软件编辑出的源程序，必须用编译程序汇编后生成目标代码。如果源程序有语法错误，需修改源文件后继续编译，直到无语法错误为止。这之后利用目标码通过仿真器进行程序调试，排除设计和编程中的错误，直到成功。

(7) 程序优化。使各功能程序实行模块化、子程序化，缩短程序的长度，加快运算速度和节省数据存储空间，减少程序执行的时间。

6.1.2 程序设计技术

1. 模块化程序设计

模块化程序设计是单片机应用中常用的一种程序设计技术。它是把有关功能完整的、较长的程序，分解为若干个功能相对独立的、较小的程序模块，各个程序模块分别进行设计、编程和调试，最后把各功能模块集成为所需的程序。

模块化程序设计的优点是，单个功能明确的程序模块的设计和调试比较方便、容易完成。一个模块可以为多个程序所共享，也可利用现成的程序模块。

2. 自上而下的程序设计

自上而下的程序设计时，先从主程序开始设计，从属的程序和子程序用符号来代替。主程序编好后，再编制各个从属程序和子程序，最后完成整个系统软件的设计。调试也按这个次序进行。

自上而下程序设计的优点是，比较习惯人们的日常思维，设计、调试和连接同时按一个线索进行，程序错误可以较早发现。缺点是修改比较麻烦。

3 软件抗干扰设计

用于生产现场的单片机应用系统，易受各种干扰侵袭，直接影响到系统的可靠性。因此，应用系统的抗干扰设计是非常重要的。

在实际情况中，针对不同的干扰后果，采用不同的软件对策。在实时数据采集系统中，为

了消除传感器通道中的干扰信号,可采用软件数据滤波,如算术平均法、比较取舍法、中值法、一阶递推数字滤波法等;在开关量控制系统中,为防止干扰进入系统,造成各种控制条件超差、输出失控,可采取软件冗余、程序自检等措施;为防止程序计数器失控,造成程序盲目运行或“死机”,可设置软件“看门狗”。来监视程序运行状态,也可在非程序区设置软件陷阱,强行使程序拉回复位状态,重新启动。