



高职高专物联网应用技术专业“十三五”规划教材

嵌入式系统基础项目化教程

QIAN RU SHI XI TONG JI CHU XIANG MU HUA JIAO CHENG

主 编 徐明亮

副主编 张爱良 任靖福 廖忠智

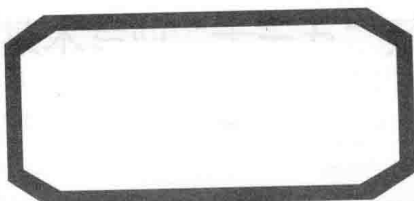


西安电子科技大学出版社
<http://www.xduph.com>



高职高专物联网应用

系列教材



嵌入式系统基础项目化教程

主 编 徐明亮

副主编 张爱良 任靖福 廖忠智

西安电子科技大学出版社

内 容 简 介

本书以飞思卡尔半导体公司 8 位 S08 系列中的 MC9S08AW60 微控制器为背景, 选择 MC9S08AW60 最小系统、GPIO 模块、KBI 模块、SCI 模块、TPM 模块、A/D 转换模块、Flash 模块, 以及嵌入式系统开发所需的软硬件平台和常用人机接口模块为主要教学内容, 并按照基于工作过程项目化教学思路将其组织为初识嵌入式系统、嵌入式系统 C 语言知识要点、AW60 及其最小系统、AW60 软件开发平台、流水灯设计、多位数码管显示、液晶显示、键盘输入、AW60 与 PC 串行通信、简易秒表设计、按键抖动捕捉、LED 呼吸灯、AW60 工作温度监测、Flash 与 RAM 存储特性演示共 14 个项目。

对于本书中 MC9S08AW60 未涉及的模块在附录中都给出了简要介绍。

本书所有项目源程序可从西安电子科技大学出版社网站(www.xduph.com)下载或联系 QQ: 1658283439 索取。

本书可作为高职高专院校机电设备类、自动化类、电子信息类及计算机应用类专业的嵌入式技术类课程教材, 也可以作为职工大学、函授大学、中职学校的参考教材及嵌入式系统应用开发人员的参考工具书和爱好者的辅助读物。

图书在版编目(CIP)数据

嵌入式系统基础项目化教程/徐明亮主编. —西安: 西安电子科技大学出版社, 2017.5

高职高专物联网应用技术专业“十三五”系列教材

ISBN 978-7-5606-4465-3

I. ① 嵌… II. ① 徐… III. ① 微型计算机—系列设计—教材 IV. ① TP360.21

中国版本图书馆 CIP 数据核字(2017)第 081306 号

策 划 陈 婷

责任编辑 韩伟娜 陈 婷

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfb001@163.com

经 销 新华书店

印刷单位 陕西利达印务有限责任公司

版 次 2017 年 5 月第 1 版 2017 年 5 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 14

字 数 328 千字

印 数 1~3000 册

定 价 27.00 元

ISBN 978-7-5606-4465-3/TP

XDUP 4757001-1

如有印装问题可调换

本社图书封面为激光防伪覆膜, 谨防盗版。

前 言

作为计算机系统一个重要的分支,嵌入式系统已在工业控制、汽车电子、测控系统、数据采集、家用电器、个人消费电子等领域获得广泛使用;嵌入式系统的相关课程在专科、本科机电设备类、自动化类、电子信息类及计算机应用类专业课程中也占有重要地位。

在嵌入式系统中,作为核心的微控制器种类繁多,目前主流微控制器有8位、16位、32位。此外,随着芯片集成度的不断提高,微控制器内部资源越来越丰富,功能也越来越多。为了更好地利用这些内部资源,充分使用微控制器的众多功能模块,嵌入式系统的软件也由单任务处理发展到多任务处理,随之而来的是嵌入式操作系统的引入和使用,因此嵌入式系统的开发和学习所需的背景知识也愈加广泛,难度大幅度提高。

按照循序渐进的原则,学习嵌入式系统相关课程应以8位微控制器、单任务处理为起点。在8位微控制器中,英特尔公司的MCS-51系列微控制器进入我国的时间早,应用最为广泛,众多高职高专院校嵌入式系统课程都以51系列微控制器为基础。随着嵌入式系统应用范围的不断拓展,众多公司也相继推出了自己的产品。这其中就包括飞思卡尔S08系列8位单片机,由于可靠性高,抗干扰性能强,因而在工业控制中得到了大量应用。现在越来越多的学校选择S08系列8位单片机作为嵌入式系统教学的入门首选芯片。为适应这一需要,编者结合本校和兄弟院校近几年的教学实践,针对学生学习实际,根据工作过程的项目化教学方法和软硬件模块化的设计思想编写了本书。按照项目内容与要求→项目背景知识→项目硬件设计→项目软件设计这一思路,本书介绍了飞思卡尔MC9S08AW60部分常用功能模块的特性、功能和软件设计方法,力争让读者在应用中学习,在学习中的应用。

本书具有以下特色:

(1) 根据教学实践经验将MC9S08AW60大部分功能模块和嵌入式系统软件开发必备知识组织为14个项目,将知识和技能融于项目。通过项目中软硬件的设计和实现,以循序渐进的方式进行学习,使学生的能力在项目的实施中逐步得到提高,达到“学以致用”的目的。

(2) 项目硬件设计尽可能简化,突出项目所涉及的功能模块相关知识和技能的学习及应用,同时降低学习难度和学习成本。

(3) 以硬件模块为基础进行软件模块化设计,便于代码重用。后续项目尽可能地包含前面设计中所使用的硬件模块和软件模块,进而加快开发进度。

(4) 有些项目设计用于验证相关知识点,通过这些项目的学习,学生不仅可以掌握芯片相关模块功能的使用方法,还能加深对相关知识点的理解。

(5) 提供完整的参考代码,这些代码都经过实际测试,可在项目开发中使用。

全书共14个项目:项目1为初识嵌入式系统,让学生理解和掌握嵌入式系统相关概念与开发特点;项目2为嵌入式系统C语言知识要点,介绍C语言程序设计的基本知识;项目3为AW60及其最小系统,介绍最小系统概念及其结构,为后续项目硬件设计做准备;项目4为AW60软件开发平台,介绍CodeWarrior中项目建立流程和基本调试方法;项目5

为流水灯设计，除介绍 GPIO 口的配置和使用外，还给出开发平台 CodeWarrior 创建实际软件项目过程，引出最小系统模块所对应的芯片初始化软件模块的使用及软件重用方法；项目 6 为多位数码管显示，介绍动态显示软件设计方法；项目 7 为液晶显示，介绍外接带有寄存器功能模块的使用方法，初步建立起时序的概念；项目 8 为键盘输入，让学生掌握 AW60 键盘输入模块和中断编程特点；项目 9 为 AW60 与 PC 串行通信，让学生掌握串行通信模块的使用，能综合应用查询法和中断法进行软件设计；项目 10 为简易秒表设计，介绍定时器模块应用；项目 11 为按键抖动捕捉，介绍输入捕捉并回顾键盘消抖处理的必要性；项目 12 为 LED 呼吸灯，介绍 PWM 模块；项目 13 为 AW60 工作温度监测，涉及 A/D 转换模块及线性插值方法；项目 14 为 Flash 与 RAM 存储特性演示，介绍 Flash 在线写入方法并比较其与 RAM 的存储特性。

附录 A 中给出了常用 ASCII 表。附录 B 中对 AW60 中其他未涉及的模块给出了简要介绍。附录 C 给出了 S08 的 C 语言函数库，以备学习时查找。

本书所有实例源程序可向西安电子科技大学出版社或作者(QQ:1658283439)索取，相关硬件的了解可与苏州大学飞思卡尔嵌入式系统研发中心联系。

本书除封面署名作者外，李晓妮老师还协助进行了项目 1 至项目 4 的编辑整理工作，本书编写过程中得到苏州大学王宜怀教授的帮助和指导，在此表示诚挚的谢意。

由于作者水平有限，加上时间有限，书中难免存在不足之处，恳请读者提出宝贵意见和建议，以便改进。

作者

2017 年 1 月于江苏无锡

目 录

项目 1 初识嵌入式系统	1
1.1 项目内容与要求	1
1.2 项目背景知识	1
1.2.1 嵌入式系统定义	1
1.2.2 嵌入式系统的发展与现状	1
1.2.3 嵌入式系统的特点	2
1.2.4 嵌入式系统的应用领域	3
1.2.5 嵌入式系统的开发流程	3
习题	4
项目 2 嵌入式系统 C 语言知识要点	5
2.1 项目内容与要求	5
2.2 嵌入式系统 C 语言数据类型	5
2.2.1 基本数据类型	5
2.2.2 构造类型	5
2.2.3 运算符	13
2.2.4 流程控制	14
2.2.5 函数	16
2.2.6 编译预处理	17
习题	19
项目 3 AW60 及其最小系统	20
3.1 项目内容与要求	20
3.2 项目背景知识	20
3.2.1 飞思卡尔 S08 系列微控制器	20
3.2.2 MC9S08AW60 芯片引脚分配	22
3.2.3 AW60 存储器映像	25
3.2.4 S08CPU 的内部寄存器	26
3.2.5 AW60 最小系统	29
项目 4 AW60 软件开发平台	31
4.1 项目内容与要求	31
4.2 AW60 硬件开发平台	31
4.3 AW60 软件开发平台	32
4.3.1 CodeWarrior for S08 V6.2	32
4.3.2 CodeWarrior 工程项目建立与调试	32
4.3.3 prn 文件	41
4.3.4 start08.c 文件及启动过程	50
4.3.5 寄存器头文件 MC9S08AW60.h	51

项目 5 流水灯设计	53
5.1 项目内容与要求	53
5.2 项目背景知识	53
5.2.1 AW60 的 GPIO 概述	53
5.2.2 AW60 的 GPIO 相关寄存器	54
5.3 项目硬件设计	56
5.4 项目软件设计	57
5.4.1 软件结构与流程设计	57
5.4.2 软件代码设计	58
5.5 项目建立与调试	62
设计小结	63
习题	63
项目 6 多位数码管显示	64
6.1 项目内容与要求	64
6.2 项目背景知识	64
6.2.1 LED 数码管结构和显示原理	64
6.2.2 LED 数码管驱动方式	66
6.3 项目硬件设计	66
6.4 项目软件设计	67
6.4.1 软件结构与流程设计	67
6.4.2 工程建立与代码编辑及调试	68
设计小结	74
习题	74
项目 7 液晶显示	75
7.1 项目内容与要求	75
7.2 项目背景知识	75
7.2.1 液晶显示器概述	75
7.2.2 LCD1602 使用说明	76
7.3 项目硬件设计	82
7.4 项目软件设计	82
7.4.1 软件结构与流程设计	82
7.4.2 程序代码设计	83
设计小结	88
习题	89
项目 8 键盘输入	90
8.1 项目内容与要求	90
8.2 项目背景知识	90
8.2.1 键盘	90
8.2.2 中断及其处理过程	93
8.2.3 AW60 中断资源及中断处理	94

8.2.4 AW60 键盘模块相关寄存器	98
8.3 项目硬件设计	102
8.4 项目软件设计	103
8.4.1 软件结构与流程设计	103
8.4.2 软件代码设计	104
设计小结	112
习题	112
项目 9 AW60 与 PC 串行通信	113
9.1 项目内容与要求	113
9.2 项目背景知识	113
9.2.1 串行通信	113
9.2.2 AW60 的 SCI 模块	118
9.2.3 串口调试器	128
9.3 项目硬件设计	128
9.4 项目软件设计	129
9.4.1 软件流程设计	129
9.4.2 软件结构与代码设计	130
9.4.3 系统测试	134
设计小结	134
习题	135
项目 10 简易秒表设计	136
10.1 项目内容与要求	136
10.2 项目背景知识	136
10.2.1 AW60 定时器模块概况	136
10.2.2 定时/计数相关寄存器	136
10.3 项目硬件设计	142
10.4 项目软件设计	142
10.4.1 软件结构与流程设计	142
10.4.2 软件结构与代码设计	142
设计小结	146
习题	146
项目 11 按键抖动捕捉	147
11.1 项目内容与要求	147
11.2 项目背景知识	147
11.2.1 输入捕捉	147
11.2.2 AW60 输入捕捉引脚及寄存器	148
11.3 项目硬件设计	151
11.4 项目软件设计	152
11.4.1 软件结构与流程设计	152
11.4.2 代码设计	152

设计小结	158
习题	158
项目 12 LED 呼吸灯	159
12.1 项目内容与要求	159
12.2 项目背景知识	159
12.2.1 呼吸灯及其设计原理	159
12.2.2 PWM	159
12.2.3 AW60 输出比较	160
12.3 项目硬件设计	162
12.4 项目软件设计	163
12.4.1 软件结构与流程设计	163
12.4.2 代码设计	163
设计小结	165
习题	166
项目 13 AW60 工作温度监测	167
13.1 项目内容与要求	167
13.2 项目背景知识	167
13.2.1 A/D 转换相关概念	167
13.2.2 AW60 的 A/D 转换模块	168
13.2.3 AW60 的内置温度传感器	183
13.3 项目硬件设计	183
13.4 项目软件设计	184
13.4.1 软件结构与流程设计	184
13.4.2 主要代码分析	184
设计小结	188
习题	188
项目 14 Flash 与 RAM 存储特性演示	189
14.1 项目内容与要求	189
14.2 项目背景知识	189
14.2.1 RAM 与 Flash 概述	189
14.2.2 S08 系列 MCU 的 Flash 存储器特点	190
14.2.3 AW60 Flash 存储器的编程寄存器	191
14.3 项目硬件设计	199
14.4 项目软件设计	199
设计小结	205
习题	205
附录 A 常用 ASCII 表	206
附录 B AW60 其他模块简要介绍	207
附录 C S08 的 C 语言函数库	212
参考文献	216

项目1 初识嵌入式系统

1.1 项目内容要求

- (1) 了解嵌入式系统定义、发展进程。
- (2) 了解嵌入式系统微处理器特点。
- (3) 了解嵌入式系统相关术语。

1.2 项目背景知识

1.2.1 嵌入式系统定义

嵌入式系统(Embedded System)的定义众多。目前国内普遍认同的定义是：嵌入式系统是以应用为中心，以计算机技术为基础，并且软硬件可裁减，适用于应用系统，对功能、可靠性、成本、体积、功耗有严格要求的专用计算机系统。它一般由嵌入式微处理器、外围硬件设备、嵌入式操作系统以及用户的应用程序四个部分组成，用于实现对其他设备的控制、监视或管理等目标。

1.2.2 嵌入式系统的发展与现状

1960年，在通信系统上，首次采用存储式程序控制系统(Stored Program Control)对电子机械电话交换进行控制，这种控制系统就是嵌入式系统的雏形。嵌入式系统的概念是在1970年左右出现的，当时大部分软件设计都是由汇编语言完成的，而且这些汇编程序只能用于某一种固定的微处理器，因此嵌入式系统的发展和应用受到了一定的约束。

随着集成电路技术的发展，嵌入式系统的体系结构经历了由CISC(复杂指令集计算机)到RISC(精简指令集计算机)和Compact RISC的转变。在处理字长上由4位发展到8位、16位、32位、64位；寻址范围不断扩大，从64KB发展到16MB，直至4GB；指令执行速度也由最初的0.1MIPS提高到2000MIPS。在巨大需求的推动下，Motorola、ARM、MIPS、TI、Hitachi等公司推出了各类芯片，性能也不断提高。

嵌入式系统的核心是处理器，主要有嵌入式微处理器(EMPU, Embedded MicroProcessor Unit)、嵌入式DSP处理器(EDSP, Embedded Digital Signal Processor)、嵌入式片上系统(ESoC, Embedded System on Chip)、嵌入式微控制器(EMCU, Embedded MicroController Unit)等几种类型。

EDSP 以通用计算机 CPU 为基础。在应用中,将微处理器装配在专门设计的电路板上,只保留和嵌入式应用有关的母板功能,以大幅度减小系统体积和功耗。

EDSP 对系统结构和指令进行了特殊设计,使其适合于执行 DSP 算法,指令执行速度较高。ESoC 是一个硅片上集成的一个更为复杂的系统,除个别无法集成的器件以外,整个系统大部分均集成到一块或几块芯片中,应用系统电路板变得很简洁,对于减小体积和功耗、提高可靠性非常有利。

EMCU 又称单片机,顾名思义,就是将整个计算机系统集成到一块芯片中,一般简称 MCU。MCU 一般以某种微处理器内核为核心,芯片内部集成 ROM/EPROM、RAM、总线、总线逻辑、定时计数器、Watchdog、I/O、串口、脉宽调制输出、A/D、D/A、Flash ROM、EEPROM、LAN 控制器等各种必要的功能模块和外设。嵌入式微控制器目前的品种和数量最多,占嵌入式系统约 70% 的市场份额。为适应不同的应用需求,一般一个系列的单片机具有多种衍生产品,每种衍生产品的处理器内核都是一样的,不同的是存储器和外设的配置及封装。这样可以使单片机在功能不变的基础上最大限度地和应用需求相匹配,从而减少功耗和成本。和嵌入式微处理器相比,微控制器的最大特点是单片化,体积大大减小,功耗和成本下降,可靠性提高。微控制器是目前嵌入式系统工业的主流,微控制器的片上外设资源一般比较丰富,适合控制其他设备,因此称为微控制器。

可以预见的是,嵌入式系统将朝着以下趋势发展:

(1) 系统化。各个厂家在推出相关产品的同时,与之配套的支撑系统也将不断完善和发展,系统开发将会越来越便利。例如,三星推出 ARM7、ARM9 芯片的同时提供了与之配套的开发板和版本支持包(BSP)。微软在不断完善 Window CE 嵌入式操作系统的同时还提供 Embedded VC++ 开发工具。

(2) 微控制器功能越来越强大。随着微电子技术的进步,微控制器将朝着高集成度、多核、运行速度更快的方向发展,例如恩智浦公司推出的 8 核 P4080 微控制器。

(3) 网络化。随着 Internet 的飞速发展,Internet 技术与信息家电、工业控制技术和射频通信等技术日益密切的结合以及嵌入式系统与 Internet 的结合将推动物联网技术的发展。

1.2.3 嵌入式系统的特点

嵌入式系统通常是面向用户、面向产品、面向特定应用的。嵌入式 CPU 与通用型 CPU 的最大不同就是嵌入式 CPU 大多工作在为特定用户群设计的系统中,通常都具有功耗低、体积小、集成度高等特点,并且能够把通用 CPU 中许多由板卡完成的功能集成在芯片内部,从而有利于嵌入式系统小型化。

嵌入式系统是先进的计算机技术、半导体技术和电子技术与各个行业的具体应用相结合的产物。嵌入式系统开发涉及知识领域较广,不仅需要软硬件方面的知识,还需要应用领域的相关知识。在系统规模较大时还需要掌握团队分工和软硬件协同设计、协同测试等方面的知识和开发文档编写的规范。这就决定了它必然是一个技术密集、资金密集、高度分散、不断创新的知识集成系统。

嵌入式系统的硬件和软件都必须高效率地设计,去除冗余,力争在同样的硅片面积上实现更高的性能。这样才能在具体应用中更具有竞争力。

嵌入式系统和具体应用有机地结合在一起,它的升级换代也和具体产品同步进行,因

此嵌入式系统产品一旦进入市场,就具有较长的生命周期。

为了提高执行速度和系统可靠性,嵌入式系统中的软件一般都固化在存储器芯片或单片机中,而不是存储于磁盘等载体中。由于嵌入式系统的运算速度和存储容量仍然存在一定程度的限制,同时大部分嵌入式系统必须具有较高的实时性,因此对程序的质量,特别是可靠性,有着较高的要求。

嵌入式系统本身不具备自主开发能力,即使设计完成以后用户通常也不能对其中的程序功能进行修改,必须有一套开发工具和环境才能对其进行开发。嵌入式系统开发包括硬件设计和软件设计。其中软件较为特殊,由于目标机资源空间限制无法运行软件设计时需要的开发平台,所以软件设计是以通用计算机为平台进行的。一般将软件开发用的通用计算机称为主机,主机上运行嵌入式系统软件开发平台,开发出的嵌入式软件一般无法在主机上直接运行,而是在嵌入式系统硬件(一般将其称为目标机)上运行。由于需要将主机上的代码下载到目标机,因此还需要相关的软硬件支持,有的还需要代码在线调试功能,这些也需要额外的软硬件支持。目前的开发工具平台主要分为以下几类:

- (1) 实时在线仿真系统 ICE。
- (2) 高级语言编译器源。
- (3) 程序模拟器(Simulator)。
- (4) 实时多任务操作系统 RTOS。

其中,RTOS 是嵌入式系统的软件开发平台,它的引入解决了随着嵌入式系统中软件比重不断上升、应用程序越来越大而带来的嵌入式软件开发标准化的难题。

1.2.4 嵌入式系统的应用领域

嵌入式系统技术具有非常广阔的应用前景,其应用领域如下:

- (1) 工业控制。
- (2) 交通运输。
- (3) 信息家电。
- (4) POS 网络和电子商务。
- (5) 环境监测。
- (6) 机器人。

1.2.5 嵌入式系统的开发流程

嵌入式系统开发的一般流程主要包括系统需求分析(要求有严格规范的技术要求)、体系结构设计、软硬件协同设计、系统集成和系统测试,最后得到最终产品。

1. 系统需求分析

系统需求分析是指确定设计任务和设计目标,提炼出设计规格说明书,作为正式设计的指导和验收的标准。系统的需求一般分功能性需求和非功能性需求两方面。功能性需求是系统的基本功能,如输入/输出信号、操作方式等;非功能性需求则是对系统性能、成本、功耗、体积、重量等的要求。

2. 体系结构设计

体系结构设计是指描述系统如何实现所述的功能和非功能的需求，包括对硬件、软件和执行装置的功能划分，以及系统的软件、硬件选型等。一个好的体系结构是设计成功与否的关键。

3. 软硬件协同设计

软硬件协同设计是指基于体系结构，对系统的软件、硬件进行详细设计。为了缩短产品开发周期，设计往往是并行的。嵌入式系统的设计工作大部分都集中在软件设计上，面向对象技术、软件组件技术、模块化设计是现代软件工程经常采用的方法。

4. 系统集成

系统集成是指把系统的软件、硬件和执行装置集成在一起，进行调试，发现并改进单元设计过程中的错误。

5. 系统测试

系统测试是指对设计好的系统进行测试，看其是否满足规格说明书中给定的功能要求。

习 题

1. 简述嵌入式系统中的 MCU 与通用计算机的 CPU 的异同。
2. 嵌入式系统开发有何特点？
3. 简述嵌入式系统开发的主要流程。

项目2 嵌入式系统 C 语言知识要点

2.1 项目内容要求

- (1) 了解嵌入式系统定义、发展进程及特点。
- (2) 了解嵌入式系统相关术语。
- (3) 熟悉嵌入式系统开发常用 C 语言语法概要。

2.2 嵌入式系统 C 语言数据类型

数据类型本质上是规定数据在内存中占据多少个字节。嵌入式系统 C 语言的数据类型有基本类型和构造类型两大类。

2.2.1 基本数据类型

嵌入式系统 C 语言的基本数据类型有：

- (1) signed char, 有符号字节型, 位长为 8 位, 字节数为 1, 数据范围为 $-128 \sim +127$ 。
- (2) unsigned char, 无符号字节型, 位长为 8 位, 字节数为 1, 数据范围为 $0 \sim 255$ 。
- (3) signed short, 有符号短整型, 位长为 16 位, 字节数为 2, 数据范围为 $-32768 \sim +32767$ 。
- (4) unsigned short, 无符号短整型, 位长为 16 位, 字节数为 2, 数据范围为 $0 \sim 65535$ 。
- (5) signed int, 有符号短整型, 位长为 16 位, 字节数为 2, 数据范围为 $-32768 \sim +32767$ 。
- (6) unsigned int, 无符号短整型, 位长为 16 位, 字节数为 2, 数据范围为 $0 \sim 65535$ 。
- (7) signed long, 有符号长整型, 位长为 32 位, 字节数为 4, 数据范围为 $-2147483648 \sim +2147483647$ 。
- (8) unsigned long, 无符号长整型, 位长为 32 位, 字节数为 4, 数据范围为 $0 \sim 4294967295$ 。
- (9) float, 浮点型, 位长为 32 位, 字节数为 4, 数据范围为约为 $-3.4 \times 10^{-38} \sim +3.4 \times 10^{+38}$ 。
- (10) double, 双精度型, 位长为 64 位, 字节数为 8, 数据范围约为 $-1.7 \times 10^{-308} \sim +1.7 \times 10^{+308}$ 。

2.2.2 构造类型

1. 数组

1) 数组的定义

数组的定义格式如下：

类型说明符 数组名 [常量表达式 1][常量表达式 2]…[常量表达式 n];

其中，类型说明符是任一种基本数据类型或构造数据类型。数组名是用户定义的数组标识符。方括号中的常量表达式表示每一维中数据元素的个数，一维数组中也称为数组的长度。

2) 数组的初始化

数组可以在定义的时候初始化。

```
int array[5]={1,2,3,4,5};
```

在定义数组时，可以用放在一对大括号中的初始化表对其进行初始化。初始化值的个数可以和数组元素个数一样多。如果初始化的个数多于元素个数，将产生编译错误；如果少于元素个数，后续的元素被初始化为 0。

如果维数表达式为空，那么将用初始化值的个数来隐式地指定数组元素的个数。

[例]

```
int array[]={1,2,3,4,5};
```

不过在编程时，一般不建议这么编写。

3) 数组元素访问

数组元素的访问一般都是通过下标。

【例】

```
int a;a=array[0];
```

则 a 的值为数组 array 的第一个元素。

需要注意的是，数组的第一个元素下标从 0 开始，在赋值的时候，还可以使用变量作为数组下标。

4) 字符数组

用来存放字符量的数组称为字符数组。其定义的一般形式是：

```
char 数组名[字符长度]
```

【例】

```
char array[5]='H','E','L','L','O';
```

对于单个字符，必须要用单引号括起来。由于字符和整型是等价的，所以上面的字符型数组也可以这样表示：`char array[5]={72,69,76,76,79};` /*用对应的 ASCII 码*/

但是字符型数组和整型数组也有不同的地方。

【例】

```
char array[]="HELLO";
```

该变量实际上在内存中的存储情况为：

```
char array[]={'H','E','L','L','O','\0'};
```

上面最后一个字符'\0'是一个字符常量，字符型数组的最后自动加上一个\0，作为字符

的结束标志。所以虽然“HELLO”只有5个字符，而存入到数组的字符个数却是6个，但是，数组的长度仍然是5。

2. 结构体

结构体是一种构造类型，它是由若干“成员”组成。每一个成员可以是一个基本数据类型或者是一个构造类型。既然结构体是一种“构造”而成的数据类型，那么在说明和使用之前就必须先定义。

1) 结构的定义

定义一个结构的一般形式为：

```
struct 结构名
{
    类型说明符 变量名1;
    类型说明符 变量名2;
    ...
};
```

如：

```
struct student
{
    char name[8];
    int age;
    char sex[2];
    char depart[20];
    float wage1, wage2;
} wang;
```

上面定义了一个结构名为 student 的结构变量 wang，如果省略变量名 wang，则该定义是对结构的说明。用已说明的结构名也可定义结构变量，如 struct student ding;

2) 结构体访问

依据结构体变量类型的不同，对于结构体成员一般有两种访问方式，一种为直接访问，一种为间接访问。直接访问应用于普通的结构体变量，间接访问应用于指向结构体变量的指针。直接访问使用“结构体变量名.成员名”，间接访问使用“(*结构体指针名).成员名”或者使用“结构体指针名->成员名”。

3. 联合

共用体(Union)也叫“联合”，表示几个变量共用一个内存位置，在不同的时间保存不同的数据类型和不同长度的变量。在 union 中，所有的共用体成员共用一个空间，并且同一时间只能储存其中一个成员变量的值。换句话说，这个内存内容可以按照不同的形式进行访问。

共用体的声明和共用体变量定义都使用关键字 union，其形式与结构体十分相似。具体为：

```
union 共用体名
```

```
{
    数据类型 成员名;
    数据类型 成员名;
    ...
} 变量名;
```

访问共用体成员的方法与结构体相同，即采用“.”号和“->”访问。同样普通共用体变量用“.”访问成员变量，指向共用体变量的指针用“->”访问成员变量。

4. 枚举类型

在程序设计中，有时会用到由若干个有限数据元素组成的集合，如三种颜色红、黄、绿组成的集合，反映到程序中即为某个变量取值为有限可列。此时，可将这些有限可列的数据集合定义为枚举类型。

1) 枚举类型变量定义

定义枚举类型变量有三种方法，即：先定义类型后定义变量，定义类型的同时定义变量，直接定义变量 C 结构与联合体也有类似的定义方法，现介绍如下：

(1) 先定义类型后定义变量

```
enum <枚举类型名>{<枚举元素表>;
```

格式：<枚举类型名> <变量 1> {, <变量 2>, ..., <变量 n>} ;

例如：enum colors

```
{red=5,blue=1,green,black,white,yellow};
```

```
colors c1,c2;
```

c1、c2 为colors类型的枚举变量。

(2) 定义类型的同时定义变量。

格式：enum <枚举类型名>

```
{<枚举元素表> } <变量1> {, <变量2>, ..., <变量n>} ;
```

例如：enum colors

```
{red=5,blue=1,green,black,white,yellow} c1,c2;
```

(3) 直接定义枚举变量。

格式：enum

```
{<枚举元素表> } <变量1> {, <变量2>, ..., <变量n>} ;
```

例如：enum

```
{red=5,blue=1,green,black,white,yellow} c1=red,c2=blue;
```

由上例可以看出，定义枚举变量时，可对变量进行初始化赋值，c1的初始值为red，c2的初始值为blue。

```
enum <枚举类型名>
```

```
{<枚举元素表>};
```

其中，关键词enum表示定义的是枚举类型，枚举类型名由标识符组成，而枚举元素表由枚举元素或枚举常量组成。例如：

```
enum weekdays
```