



普通高等教育“十一五”国家级规划教材
软件工程专业核心课程系列教材

嵌入式技术基础与实践实验指导 (第二版)

曹金华 王宜怀 沈安东 著

- ◆ 果《中国软件工程学科教程》及专业规范组织编写
- ◆ 根据教育部“软件工程课程体系研究”项目成果《中国软件工程学科教程》及专业规范组织编写
- ◆ 与最新ACM和IEEE CCSE同步
- ◆ 汇集示范性软件工程专业教学成果

清华大学出版社



普通高等教育“十一五”国家级规划教材
软件工程专业核心课程系列教材

嵌入式技术基础与实践实验指导 (第二版)

曹金华 王宜怀 沈安东 著

清华大学出版社
北京

内 容 简 介

全书共分10章。第1~2章为嵌入式实验系统开发软件和硬件设备,介绍了学习嵌入式系统的软件工具和硬件工具,如何自制嵌入式系统的软件开发和硬件构建;第3章为普通输入输出(I/O)入门实验,引导学生学习软件开发工具和构建简单的实验,以便掌握I/O口的入门知识;第4章为串行通信入门实验,引导学生结合PC高端做串行通信SCI的实验,以便掌握简单的通信入门知识;第5章为I/O口的基础实验,通过实验可以掌握嵌入式系统的I/O口基本知识;第6章为模拟量的基础实验,通过实验掌握嵌入式系统采集模拟量、处理模拟量的基本知识;第7章为定时器的基础实验,通过实验掌握嵌入式系统定时器的基本知识;第8章为高级通信实验,是具有较深知识的通信实验,比如SPI、IIC、USB、CAN;第9章为其他模块实验,是嵌入式系统的其他功能实验,比如Flash、IRQ;第10章是毕业设计范例,选取了一个题目,是从苏州大学本科毕业设计的选题中精选出来的,可供学生在课程设计和毕业设计时参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

嵌入式技术基础与实践实验指导/曹金华,王宜怀,沈安东著.—2版.—北京:清华大学出版社,2011.6

(软件工程专业核心课程系列教材)

ISBN 978-7-302-24828-6

I. ①嵌… II. ①曹… ②王… ③沈… III. ①微处理器—系统设计—教学参考资料
IV. ①TP332

中国版本图书馆CIP数据核字(2011)第033027号

责任编辑:魏江江 薛 阳

责任校对:时翠兰

责任印制:杨 艳

出版发行:清华大学出版社

地 址:北京清华大学学研大厦A座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954,jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015,zhiliang@tup.tsinghua.edu.cn

印 装 者:北京嘉实印刷有限公司

经 销:全国新华书店

开 本:185×260 印 张:9 字 数:219千字

版 次:2011年6月第2版 印 次:2011年6月第1次印刷

印 数:1~3000

定 价:18.00元

产品编号:041217-01

前 言

本书是《嵌入式技术基础与实践(第二版)》的配套实验指导书,书中所选实验及毕业设计范例都是从教学、科研的实际程序中提取的,均经过调试,保证具有可验证性,以便学校完成嵌入式系统的实训教学。

全书共分 10 章。第 1~2 章为嵌入式实验系统开发软件和硬件设备,介绍了学习嵌入式系统的软件工具和硬件工具、如何自制嵌入式系统的软件开发和硬件构建;第 3 章为普通输入输出(I/O)入门实验,引导学生学习软件开发工具和构建简单的实验,以便掌握 I/O 口的入门知识;第 4 章为串行通信入门实验,引导学生结合 PC 高端做串行通信 SCI 的实验,以便掌握简单的通信入门知识;第 5 章为 I/O 口的基础实验,本章安排了 3 个实验,学生通过亲自动手完成这些基础实验,可以掌握嵌入式系统的 I/O 口基本知识;第 6 章为模拟量的基础实验,学生通过动手完成这个基础实验,可以掌握嵌入式系统采集模拟量、处理模拟量的基本知识;第 7 章为定时器的基础实验,本章安排了 4 个实验,学生通过动手完成这些基础实验,可以掌握嵌入式系统定时器的基本知识;第 8 章为高级通信实验,本章安排了 4 个实验,是具有较深知识的通信实验,比如 SPI、IIC、USB、CAN;第 9 章为其他模块实验,本章安排了两个实验,是嵌入式系统的其他功能实验,比如 Flash、IRQ;第 10 章是毕业设计范例,选取了一个题目,是从苏州大学本科毕业设计的选题中精选出来的,可供学生课程设计和毕业设计参考。附录中还提供了实验报告的样例,学生可以参考其完成实验报告,同时提供了一个综合实验报告,本实验是综合了嵌入式系统各个模块设计的,以便进一步学习使用。

本书所有实验的源程序见主教材《嵌入式技术基础与实践》的配套资料(下载地址: www.tup.com.cn),该资料包含了课堂教材的所有实例程序、本实验指导教材所涉及的源程序、相关芯片的技术手册及硬件评估板的技术资料等。

本书可作为高等院校自动化、电气工程、应用电子技术、机电一体化、微机应用、数控技术、计算机科学与技术、仪表及控制、信息通信技术、汽车工程等专业的实践教学环节的指导教师,同时也可作为相关教师及工程技术人员的参考书。

由于编者水平有限,书中难免有错误与不妥之处,恳请广大读者批评指正。

编 者

2011 年 3 月

目 录

第 1 章 CodeWarrior for Freescale HC08/S08/RS08 集成开发环境	1
1.1 CW 环境安装与设置	1
1.2 CodeWarrior 的使用	2
1.3 工程文件组织	5
1.4 样例	6
1.5 程序的写入	7
1.6 编译调试程序	8
1.7 程序调试方法	10
第 2 章 HC08/S08 硬件实验系统	12
2.1 SD-ExtBoard-D 型扩展板	12
2.1.1 与核心板接口	12
2.1.2 电源模块	13
2.1.3 复位电路	15
2.1.4 调试小灯模块	16
2.1.5 拨码开关	16
2.1.6 数码管(LED)	17
2.1.7 继电器	19
2.1.8 蜂鸣器	20
2.1.9 电位器	20
2.1.10 键盘	21
2.1.11 液晶(LCD)	22
2.1.12 传感器接口	23
2.1.13 PWM 控制接口	26
2.1.14 串行通信	26
2.1.15 串口转 USB	27
2.1.16 AT45DB041D 芯片的 SPI 接口	28
2.1.17 备用 SPI 接口	28
2.1.18 PCF8563 芯片的 IIC 接口	29
2.1.19 备用 IIC 接口	29
2.1.20 CAN 驱动电路	30
2.1.21 CAN 接口	30
2.1.22 网络接口	31

2.1.23	USB 接口	31
2.1.24	RFID	32
2.1.25	Zigbee 模块接口	33
2.2	核心板	33
2.2.1	实物图	34
2.2.2	原理图	34
2.2.3	元件布局图	35
2.2.4	独立测试程序	35
2.3	S08/S12/Coldfire 三合一写入器(USBDM)	38
2.3.1	USBDM 简介	38
2.3.2	驱动安装	39
2.3.3	调试库安装	39
2.3.4	USBDM 使用	39
2.3.5	USBDM 配套独立写入软件	41
第 3 章	程序框架和 I/O 口的实验	42
第 4 章	RS-232 串口通信实验及高级语言	45
第 5 章	I/O 应用的实验	52
第 6 章	模拟量数字量转换	64
第 7 章	定时器应用	68
第 8 章	嵌入式通信	79
第 9 章	其他模块	96
第 10 章	嵌入式应用技术毕业设计范例	101
附录 A	苏州大学 Freescale 嵌入式系统实验室已有核心板及评估板	117
附录 B	实验报告样例(实验二)	118
附录 C	综合实验报告	123

第 1 章 CodeWarrior for Freescale HC08/S08/RS08 集成开发环境

嵌入式软件开发系统有别于桌面软件开发系统的一个显著的特点是,它一般需要一个交叉编译和调试环境,即编辑和编译软件在通常的 PC 上进行,而编译好的软件需要通过写入工具下载到目标机上执行,如 AW60 目标机。由于主机和目标机处理器的体系结构彼此不同,从而增加了嵌入式软件开发系统的开发难度。所以选择一些好的开发套件有助于对目标机的学习与开发。下面介绍 Freescale 公司的 CodeWarrior for HC08/S08/RS08 V6.2 集成开发环境(简称 CW 环境)的安装和使用。

1.1 CW 环境安装与设置

CW 环境安装没有什么特别之处,在 Windows 操作系统上,只要按照安装向导单击鼠标就可以自动完成。

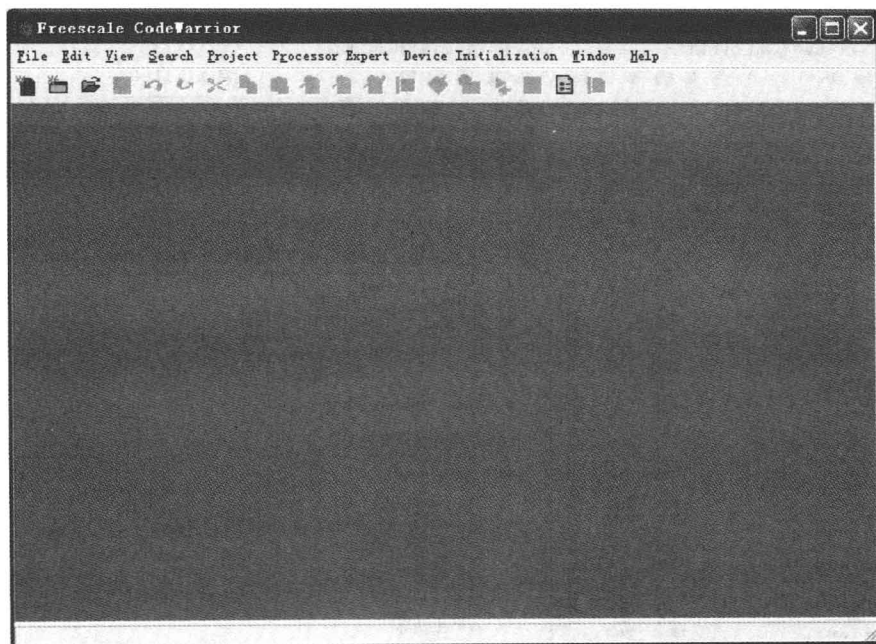


图 1-1 CW 环境运行界面

需要说明的是,安装完毕以后要上网注册以申请使用许可(License Key)。无论是下载的软件还是申请到的免费光盘,安装后都要通过因特网注册,以申请使用许可。这里可通过登录其网站,单击 Request a Key 实现。由于这一注册过程是在网上自动实现的,故只要网

络通畅,这个往返过程在数分钟之内即可可完成。申请后会通过 E-mail 得到一个 License.dat 文件。将该文件复制到相应目录下即可,例如 C:\Program Files\Freescale\CodeWarrior for Microcontrollers V6.2。对于免费的特别版本,安装好后用 License.dat 覆盖安装目录下的 License.dat。CW 环境的运行界面如图 1-1 所示。

由于 CodeWarrior IDE 安装后的默认字体是 Courier New,对中文的支持不完善,因此建议修改字体。选择 Edit → Preferences 选项,弹出 IDE Preferences 对话框。在 Font& Tabs 选项设置字体为 Fixedsys,Script 为 CHINESE_GB2312,由于 Tab 在不同文本编辑器解释不同,建议在 Tab Inserts Spaces 前打钩,使 Tab 键插入的是多个空格。

网上配套资料中有相关使用方法的介绍,本章也有如何进行编辑、编译、程序运行、调试等方面的介绍。

1.2 CodeWarrior 的使用

创建工程有两种方法,一种是向导新建法,另一种是使用已存在的工程来建立另外一个工程,即拷贝新建法。

第一种方法的操作步骤如下:

(1) 创建新项目。打开 CodeWarrior IDE 后,出现如图 1-2 所示的界面。

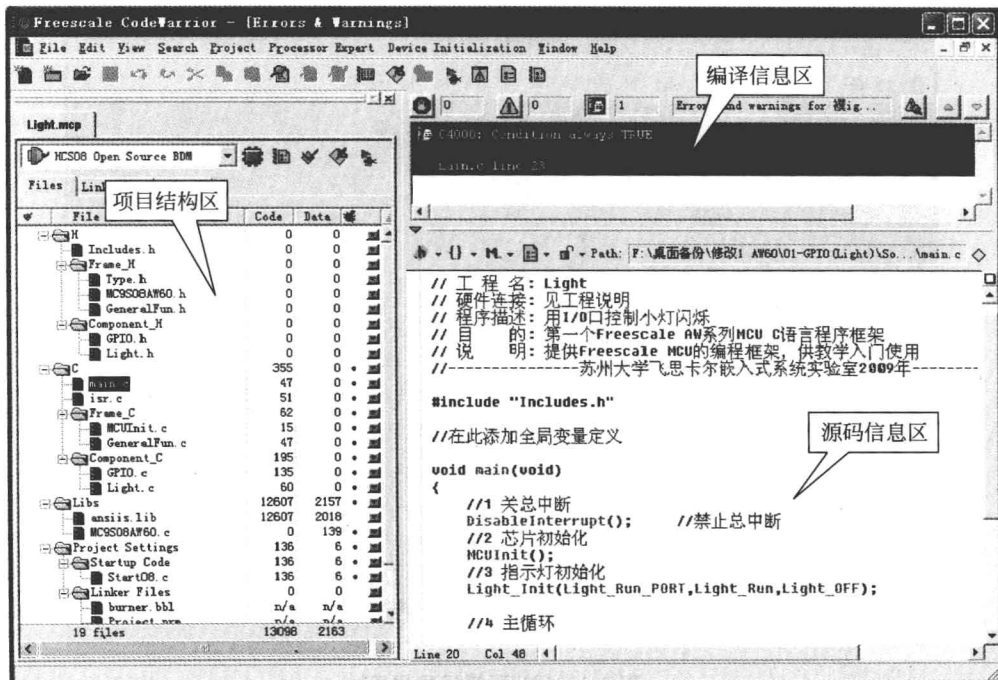


图 1-2 打开 CodeWarrior IDE 后的运行界面

(2) 在 File 菜单中选择 New Project 选项。在弹出的新窗口中,在左侧选择 MCU 芯片(如 MC9S08AW60),在右侧选择 HCS08 Open Source BDM(之前要安装相应调试库,具体

见第 2.3.3 节的调试库安装),单击“下一步”按钮,如图 1-3 所示。

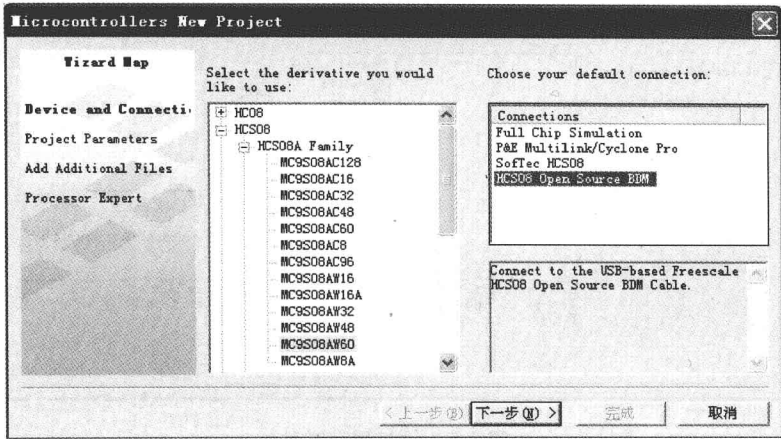


图 1-3 处理器芯片类型选择

(3) 选择语言支持,在 C 前打钩,则支持 C 语言,若程序为汇编代码,则应该选中 Relocatable assembly,若为 C++ 语言,则选中 C++; 在右侧 Project name 中输入工程名,在 Location 中选择工程所在目录。单击“下一步”按钮,如图 1-4 所示。

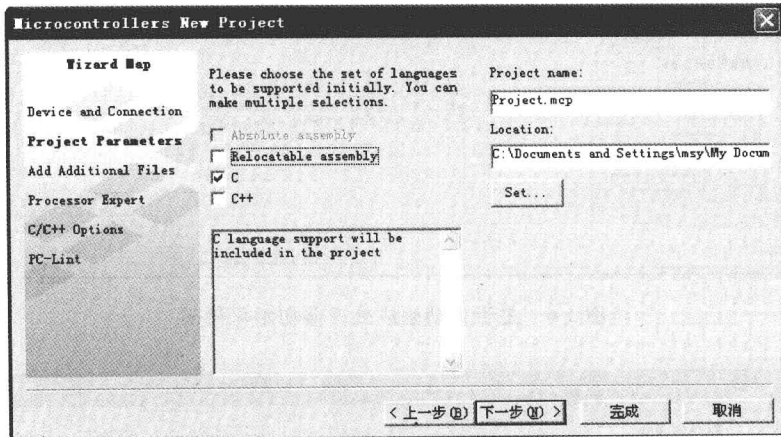


图 1-4 编程语言选择

(4) 选择要添加到当前项目中的文件,单击“下一步”按钮,如图 1-5 所示。

(5) 如图 1-6 所示,选择是否采用 Processor Expert 自动生成初始化代码,该项功能可以方便程序的寄存器设置,若选择 Device Initialization,CodeWarrior 会包含初始化代码、中断向量表、中断服务程序模板;若不想开启这些功能,则可以选择 None,然后单击“下一步”按钮。

(6) 如图 1-7 所示,选择 ANSI startup code 选项,或者选择 minimal startup code 选项,后者可以使得 startup 代码更加紧凑。第二项选择存储模型,一般选“small”。第三项选择浮点数,若工程中需用浮点数,选择 float is IEEE32,double is IEEE32,float is IEEE32,double is IEEE64 中的任意一个,若工程中不用浮点数计算,选择 None 并单击“下一步”按钮。

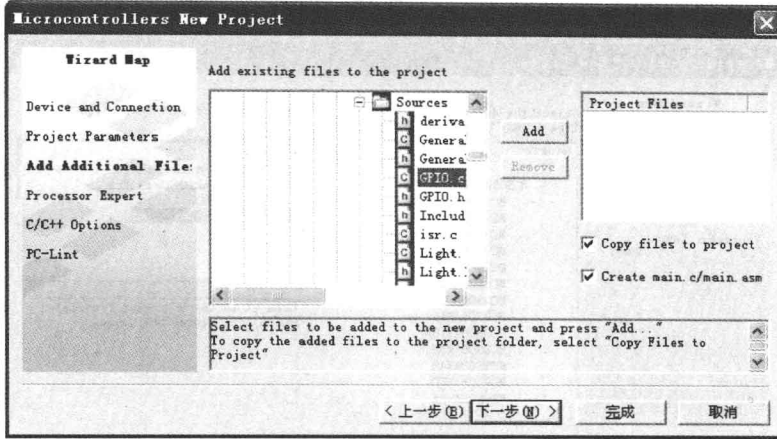


图 1-5 添加现有文件

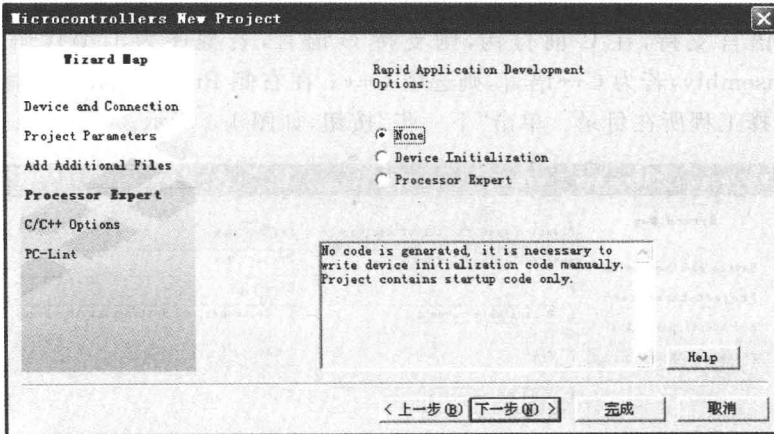


图 1-6 是否自动生成处理器初始化代码

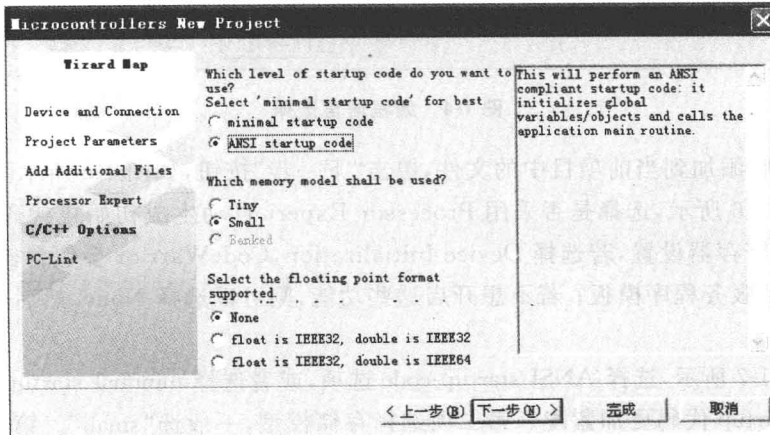


图 1-7 Startup 代码选择

(7) 如图 1-8 所示,选择是否采用 PC-Lint,一般选择 No,单击“完成”按钮,操作结束。

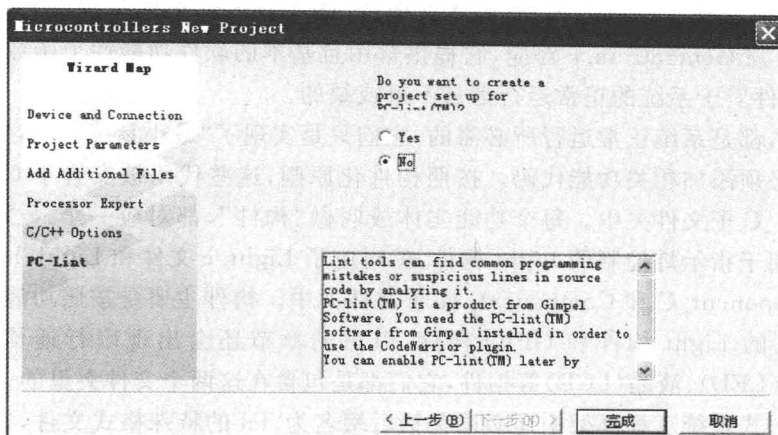


图 1-8 是否采用 PC-Lint

第二种方法是使用已有的工程来建立另一个工程。在已有工程的基础上,修改成另一个项目,比如在 Light 工程的基础上编写 LCD 程序,需要进行如下设置:

- (1) 更改工程文件夹名为 LCD;
- (2) 更改 Light.mcp 为 LCD.mcp;
- (3) 删除文件夹 Light_Data,编译后自动生成 LCD_Data;
- (4) 将 bin 文件夹的所有内容删掉。

1.3 工程文件组织

以控制小灯闪烁工程为例,介绍基于 CW 环境的嵌入式工程文件组织方法。图 1-9 给出了该工程相关源文件的树型结构,可分为头文件、源文件、库文件、工程设置文件、工程说明文件 5 个部分。其中头文件包含总头文件、总体框架头文件和软件构件头文件 3 个部分;源文件分为主函数 main、中断处理函数文件、总体框架程序文件和软件构件文件 4 个部分。

以 C 语言(其他语言也类似)嵌入式程序为例进行阐述,请特别注意 main.c 和 isr.c 这两个文件,从源程序角度来看,一般嵌入式软件的执行流程如下:系统启动并初始化后,程序根据 main.c 中定义的主循环顺序执行;当遇到中断请求时,转而执行 isr.c 中定义的相应中断处理程序;中断处理结束则返回中断处继续顺序执行。由于 main.c 和 isr.c 文件反映了软件系统的整体执行流程,故在工程文件组织时,将它与其余 C 语言程序文件分开管理,同样,与这两个文件对应的总头文件 includes.h 也被单独拿出,放在头文件的根目录下。

此外,与总体框架程序相关的头文件和源文件分别放在了 Frame_H 和 Frame_C 子文件夹中,以归类管理。Frame_H 里包含了 type.h、MC9S08AW60.h、GeneralFun.h 三个头文件。type.h 用于类型别名定义,它将 C 语言中用于变量类型定义的关键字简化定义成比较简短的形式,这样,开发者在定义变量时,可以不必敲入冗长的变量定义关键字,同时,这也为不同编译体系间的代码复用和移植提供了方便。MC9S08AW60.h 是 MC9S08AW60

芯片寄存器及相关位定义头文件,它可被视为芯片的接口文件,没有这个文件,就不可能对该芯片进行任何操作。Frame_C 子文件夹中的 MCUInit.c 文件包含实际初始化代码。GeneralFun.h 与 GeneralFun.c 对应,它提供常用且基本的软件功能性子函数,如延时子函数等。这些文件对于系统的正常运行都是不可或缺的。

以上所说,都是系统正常运行所必需的,它们只是实现了“最小系统”。若要系统能做实际的事情,还必须添加相关功能代码。按照构件化原则,这些代码被安置于 Component_H 与 Component_C 子文件夹中。每个功能实体或叫做“构件”,都对应一个.c 文件和一个.h 文件。例如,用于指示灯控制的 Light 构件,就对应了 Light.c 文件和 Light.h 文件,它们分别放置于 Component_C 和 Component_H 子文件夹中。构件主要是按照功能进行划分的,除了这里定义的 Light 构件和 GPIO 构件,以后的章节还会出现串行通信(SCI)、键盘(KB)、数码管(LED)、液晶(LCD)等构件,它们都是包含在这两个文件夹里的。

另外,嵌入式系统工程框架中还必须包括后缀名为.lcf 的特殊格式文件,它是一个地址链接文件,用于告诉编译器代码是如何安放在具体的地址空间的。了解该文件的格式,有助于全面理解嵌入式系统的运作,本章后续章节会简要介绍它的内容。特别注意,要重视工程说明文件,必要的工程说明对于软件开发是非常必要的。

1.4 样 例

图 1-9 是第一个 MC9S08AW60 样例程序的工程文件组织。

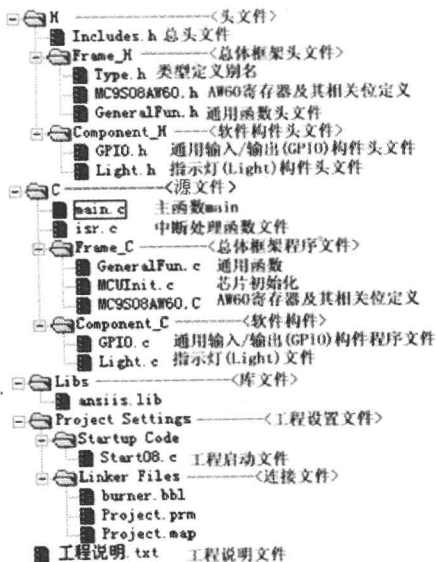



图 1-9 第一个小灯程序工程文件组织

具体程序代码见 www.tup.com.cn 上的资料。

1.5 程序的写入

程序写入 Flash 中,只需在编译无错后,单击  即可。但这里有几点要注意一下,在写入初始,会出现 USBDM Configuration 对话框,如图 1-10 所示。

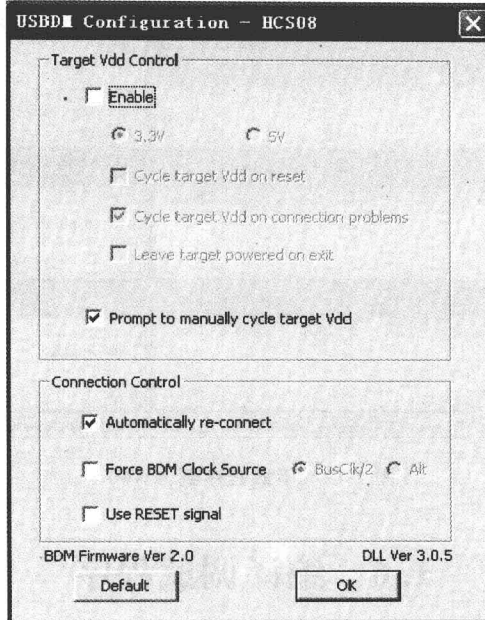


图 1-10 USBDM Configuration 对话框

若此时核心板未接扩展板,则需单击 Enable 复选框,点选 3.3V 单选按钮,再进行写入。若核心板接扩展板且扩展板已接电源,则不用选择电压亦可,否则没有电源供应会出现如图 1-11 所示的错误对话框。请选择“重试”,再选择 Enable 和 3.3V 选项。

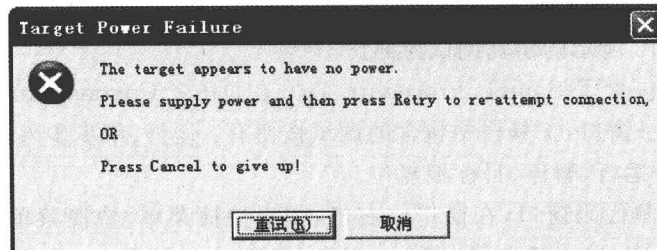


图 1-11 电源失败对话框

写入完成后,即进入调试界面,如图 1-12 所示。

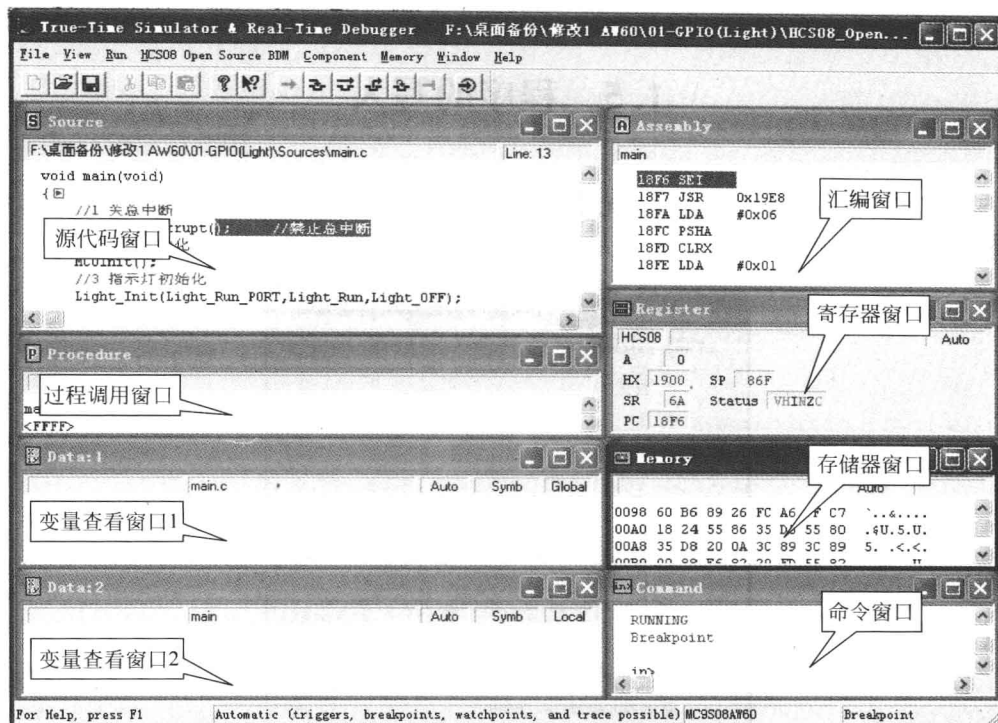




图 1-12 下载程序到 AW60

1.6 编译调试程序

1. Build project

Build project 时,可以用以下几种方法:使用菜单 project→make 命令;或者使用工具栏中的  按钮;或者直接按 F7,如果程序语法有错的话,系统会报错,修改错误继续以下操作。

2. Start debugger

在编译、链接没有错误时,可以进行仿真调试程序。单击工程窗口工具栏的  (Debug 按钮),会弹出图 1-12 所示的仿真调试界面。

仿真调试界面包含了 Source、Procedure、Data1、Data2、Command、Assembly、Register 和 Memory 共 8 个子窗口,工具栏中包含的调试按钮有:运行、单步步入、单步步过、单步步出、汇编语言的单步运行、暂停、目标板复位。

Source 窗口即源代码窗口,在窗口中右击,弹出快捷菜单,快捷菜单中有设置断点、运行到光标行、显示断点、取消断点等功能。

Procedure 窗口即过程调用窗口,实时地显示主函数中各个函数的调用情况。



Assembly 窗口即汇编窗口,其内容是 C 语言所对应的汇编代码。在调试过程中,C 语言的语句往往对应了一条或多条汇编语句,汇编窗口蓝条指示的位置的指令,就是下一条要执行的指令。








Register 窗口即寄存器窗口,该窗口在调试过程中,是一个重要的信息窗口,它反应了


上一条指令执行后寄存器的值。


Memory 窗口即存储器窗口,HC08 和 S08 内部采用统一编址,所以通过存储器窗口可以查看 I/O 寄存器、RAM 区和 Flash 区的内容,单击窗口中的字节,可以输入新值,然后按 Enter 键确认。

Command 窗口即命令窗口,当执行单步调试、连续运行、停止等命令时,在命令窗口中显示相应的命令及执行信息。

Start debugger 时,可以使用以下三种方法:使用 project→debug 菜单命令;或者使用工具栏按钮 ;或者直接按 F5 键,出现如图 1-12 所示的窗体,等待下载完成,单击  完成下载。


在线调试时常用到的几个按钮是       。


 代表开始运行程序,单片机开始运行,在上面的窗口里面可以看见一些变量和常量的值。


 代表单步执行,功能是一行一行地执行程序代码,方便查看每句程序运行后变量的状态。

 代表不进入函数内部执行。

 代表从已进入的函数中跳出。

 代表汇编语句的单步执行。

 代表暂停程序的执行。

 代表重新启动程序。

调试中常常需查看程序运行时的变量和常量的值,如图 1-13 所示,在 Data1 或者 Data2 的窗口空白处双击,出现对话框,输入你想查看的变量或常量名,比如 sensor_state 就可以查看了。最后还有一步要做:在 Data1 或者 Data2 的窗口空白处右击,选择 mode→periodical,在弹出的窗口中将 10 改成 1,这样就可以动态地观察变量的值。

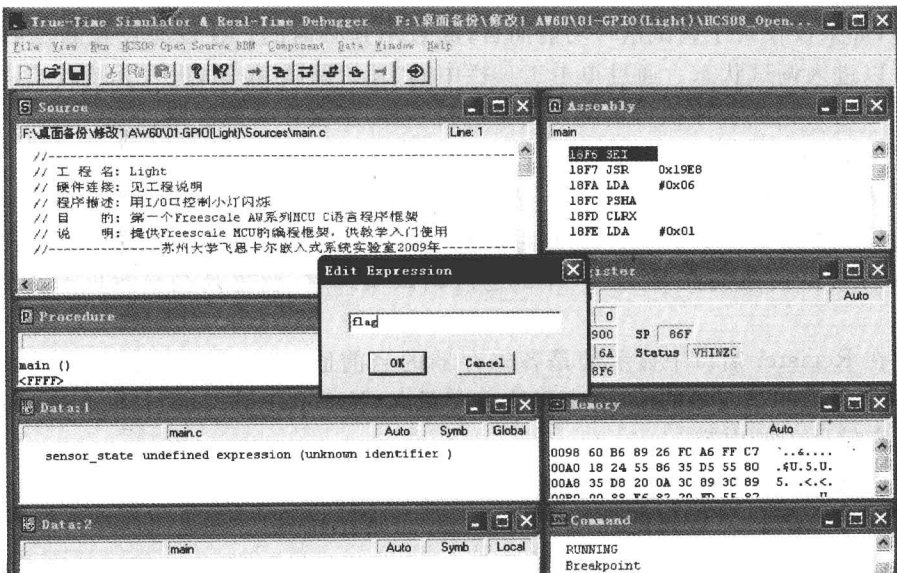



图 1-13 运行时查看变量的值

1.7 程序调试方法

下面以调试小灯程序为例讲述程序的调试过程。

1. 调试前提

在使用调试前,必须确保目标 MCU 芯片内的程序与集成开发环境中的待调试的程序一致。方法有两种,自动方式和手工方式。自动方式是只需单击 CodeWarrior IDE(如图 1-1 所示)的工具栏的  即可。手工方式是在仿真调试界面(如图 1-12 所示)通过菜单中的 Flash 操作完成:首先,“擦除”MCU 芯片内的程序;其次,“写入”待调试的程序。

2. 断点设置



在调试状态下,用户可以利用断点使程序的执行停留在设置了断点的代码行上,从而了解此时程序执行的状况。用户能够了解的信息包括各寄存器的值、程序中定义的变量值、Flash 中相应地址的值。右击某行在右键菜单中选择 Set BreakPoint 选项,即可在该行设置断点。设置了断点的代码行将以  标记(如图 1-14 所示);在右键菜单中选择 Delete BreakPoint 选项即可去除该行的断点。



图 1-14 设置断点

但是,如果仅仅对于调试这一功能而言,断点不是必需的。在没有设置断点的情况下,用户仍可以进入调试状态。通过单击工具栏中的“开始调试”按钮,进入无断点的调试模式。此时,MCU 芯片内的程序执行情况 and 正常运行一样,不会在某处停留。

3. 开始断点调试

在成功地设置断点之后,用户可以单击工具栏中的  按钮(“开始调试”按钮),进入断点调试模式。程序暂停于断点所在行,但所在行尚未执行。在断点调试模式下,当前即将执行的代码行将以蓝色高亮显示(一开始进入调试模式时,蓝色高亮显示的行为设置断点的行)。

同时在 Register 窗口中显示的是各寄存器的当前值(各寄存器的值会自动显示);在 Data 窗口中显示的是变量的值(双击可输入变量名;按 Enter 键即可);在 Memory 窗口中显示的是存储器中的内容。

在断点调试状态下,用户不可以执行“擦除”、“写入”等 Flash 操作命令。用户仅可以执行与调试相关的命令:“设置断点”、“清除断点”、“退出调试”、“单步步入”、“单步步过”和“执行”。通过设置新的断点,或执行“单步步入”、“单步步过”、“执行”,用户可以控制程序的执行流程。

4. “单步步过”和“单步步入”

CodeWarrior 支持单步步过和单步步入功能,使得程序可以按照用户的意愿向下执行。用户每单击一次单步步过或者单步步入,调试程序就会向下执行一行代码。

单步步入和单步步过的最大区别在于:当待执行的代码行为子函数调用时,单步步过会把该子函数当作整体的一行代码执行完毕;而单步步入则会进入该子函数中,执行该子函数的每一行代码,如图 1-15 所示。

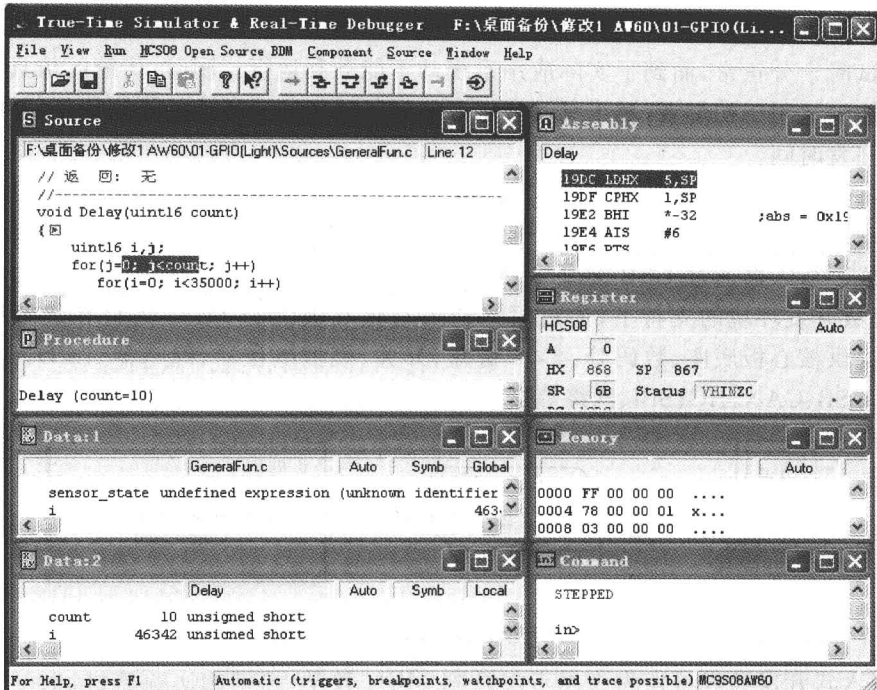


图 1-15 单步步入调用的子函数

如调试 Delay() 函数,单击 Single Step  按钮进入 Delay() 函数,单步执行,此时可同步观察 Data 窗口中的变量值的变化。单击 Step Out  按钮则可跳出 Delay() 函数。

5. 执行

CodeWarrior 集成开发环境在“调试”工具栏中提供“执行”按钮。“执行”可以使程序向下运行到断点处,但此时断点处的代码并不执行。

如果在“执行”时用户没有设置断点,MCU 芯片内的程序执行情况和正常运行一样,不会在某处停留。

6. 退出调试

在调试状态下,用户可以通过“关闭”按钮,返回代码编辑状态。