

应用于 Freescale S12 MCUs 可重定目标 集成开发环境的设计

The Design for IDE of Reorient Object For Freescale S12 MCUs

王宜怀 沈安东 (苏州大学 计算机科学与技术学院 苏州 215006)

摘要:给出了以 Freescale S12 系列 MCU 为应用目标的集成开发环境(SdIDE)的设计,并成功的将其移植到其他 MCU 上去。该系统集工程管理、编辑、编译、程序下载、调试等功能于一体,是一种新型的 MCU 实验开发工具。本文着重阐述了该系统的功能特点、软件(包括程序下载和 IDE)设计思想、不同编译器的调用方法、不同平台的移植方法,对一些技术难点进行了深入的分析说明。

关键词:Freescale S12 SdIDE TBDML 在线编程 Flash 存储器 系统移植 嵌入式系统

1 引言

在早期的 MCU 开发中,程序固化于 EEPROM 中,擦除时间较长,操作烦琐,程序调试困难。为解决此问题,仿真器技术得以较快发展,用软件来模拟目标系统中 MCU 的运行情况,但是使用仿真器往往把目标硬件与软件独立开来,难以实现 100% 的模拟硬件。随着 Flash 技术的成熟,目前,许多 MCU 内部都集成了 Flash 存储器。在线编程(In-Circuit Program)允许单片机内部运行的程序去改写 Flash 存储器的内容,利用这个特点,不仅可以在运行过程中修改某些参数,也为研制新型嵌入式应用开发工具提供了技术基础。Freescale 公司在 Flash 存储器技术相当成熟的时候,推出了新一代 16 位嵌入式单片机 HCS12 系列,其片内集成的 Flash 存储器具有单一电源电压供电、支持在线编程等特点。存储器是 MCU 的重要组成部分,存储器技术的发展对单片机的发展起到了极大的推动作用。

我们通过分析 Freescale S12 系列 MCU 的功能特点和体系结构,在 Freescale 公司提供的样片及技术支持基础上,开发了可重定目标集成开发环境 SdIDE 供用户学习开发使用。该 SdIDE 具有工程管理、编辑、编译、程序下载、调试等功能,由于涉及的技

术内容较多,限于篇幅,本文仅从该 SdIDE 的功能特点、软件(包括程序下载和 IDE)设计思想、不同编译器的调用方法和不同平台的移植方法这几个方面进行阐述,有关技术原理和方法以及相应芯片的评估开发板将在后续文章中讨论。

2 SdIDE 集成开发环境的功能及软件设计

为了方便用户开发和调试基于该系统的不同微控制器的软件,本系统设计了可重定目标的集成开发系统,该系统的在线编程集成开发环境 SdIDE 是一个高度集成的标准的 Win32 界面,包含编辑器、编译汇编链接器、调试器、工程管理、Flash 在线编程及程序下载等工具,该系统已成功用于 Freescale S12 系列微控制器的开发、同时我们也将其成功的移植到 M*Core 核心的 MMC2107 微控制器和 ATMEL 公司的 ARM920T 核心的 AT91RM9200 微控制器的开发上,下文将会介绍基于该开发环境进行平台移植的方法。下面对其功能进行一个概述,其中各模块的实现细节只作一些简要说明,因此也不给出界面。系统软件功能的主要结构如图 1 所示。

该软件采用 Visual C++ + MFC 编写,图 2 显示了它的主界面,其中左方为工程管理窗口,右方为代码

编辑,下方为编译输出及调试窗口。

此集成开发系统的主要功能及其软件设计思想如下:

(1) 标准 Win32 视窗界面,充分利用到 Windows 平台的特性,具有功能强大的菜单和工具栏,软件安装方便,占用系统资源少等特点,而且操作简便。内容丰富的帮助系统和示例程序,可以帮助用户快速的掌握 M68HC12 系列单片机的编程,同时也可以方便的将该系统移植到其他微控制器的平台上去。

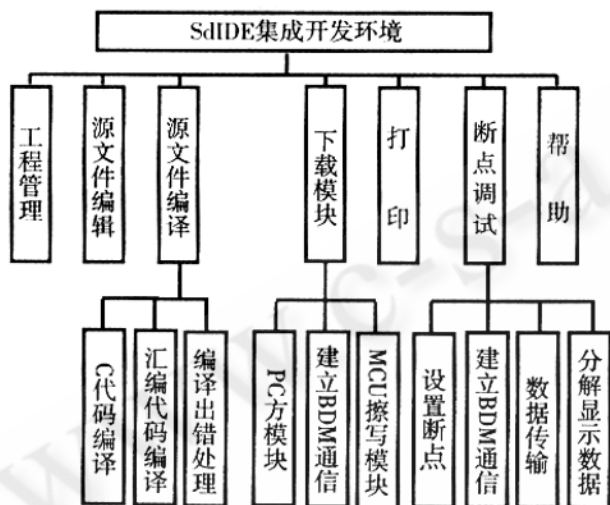


图 1 SdIDE 软件功能的主要结构图

(2) 工程管理。在 SdIDE 集成开发环境中,工程是一个非常重要的概念,它是用户组织一个应用的所有源文件、设置编译链接选项、生成调试信息文件和最终的目标文件的一个基本结构。一个工程管理一个应用的所有源文件、库文件、其他输入文件,并根据实际情况进行相应的编译、链接设置,一个工程须生成一个相对应的目录,以进行文件管理。工程管理分为两部分,可以对 C 源代码和头文件进行管理,同时支持了汇编工程的管理,在主界面上以文件树的形式进行管理。在工程属性中可以设置编译器的编译参数,编译器的路径以及处理器设置等等。该模块的主要功能就是管理源文件和生成编译器需要的文件。在保存工程的时候后台同时生成了该工程的 makefile 文件(包含编译选项和工程属性)、src 文件(包含工程中使用到的文件和资源)。

(3) 源文件编辑。可以方便的编辑源文件,支持剪

切、复制、查找、替换、注释,完全支持中文和语法高亮显示(Syntax Highlight)。此部分主要的技术难题是语法高亮显示,已有的一些控件虽然语法高亮功能比较全,但是对中文支持不是很好,最突出的就是半个汉字问题(在选择和删除字符的时候汉字当两个字符处理,一次只能删除半个,会出现乱码)。使用 MFC 的 RichEdit 控件可以解决中文问题,由于没有语法高亮功能,我们在 RichEdit 的基础上编写了语法高亮类,得到了比较满意的结果。主要编程思维是:捕获用户键盘输入的事件,判断用户输入的单词是否为 C 语言中的关键字,如果是,则改变其颜色。

(4) 编译模块。编译模块使用了 GNU 的 M68HC12 编译器,点击工具栏上的编译按钮就可以开始编译,编译成功后生成标准的 Motorola S-record 文件,而且该编译器会对 C 代码进行优化,如果要使该系统适用于其他的微控制器,则要选用 GNU 提供的不同 MCU 编译器,具体细节将在更换编译器部分介绍。

(5) 下载程序。下载程序包括 S19 文件分析模块、TBDML 通信模块和 Flash 存储器的擦除和写入模块。TBDML 通信模块负责通过 USB 接口将 PC 方的 S-record 代码写入到空白的 Flash 存储器的指定区域。S19 文件分析模块则负责将 S-Record 标准的 S19 文件进行分析,将文件的内容转换成方便传输的格式,以及判断文件中的程序的起始地址、页数、是否越界等等。用户通过点击工具栏上的下载按钮就可以弹出下载程序的界面。该下载程序工作流程如图 3 所示,下载程序的主要功能是先 will Flash 擦写程序数据和用户程序数据写入到 RAM 的指定区域(注意要事先为他们分配好固定的 RAM 区,以免出现相互重叠,从而造成数据丢失情况),当一页的用户程序数据写入到事先分配的 RAM 区后,就调用函数 tbdml_target_go()去执行 RAM 区存放的写入程序代码,若 tbdml_target_go()返回值为"0",则程序就能正确执行,从而将事先存放在 RAM 区的一页数据写入到指定的 Flash 区。

(6) 调试。为了方便用户开发和使用,系统中还集成了程序调试功能,BDM 是 16 位单片机的编程接口,除完成 FLASH 写入、擦除功能外,还可用于应用程序的调试,甚至可以在应用程序运行时,动态地获取

CPU、存储器等瞬态信息。BDM 调试工具与单片机的通信通过双向的 BKGD 引脚实现。点击工具栏上的调试按钮,就可以进入调试模式,调试模式有两种:单步调试和断点调试。两种模式都不需要用户输入调试指令。Flash 查看器和寄存器查看器分别可以查看指定 Flash 区的内容和寄存器的内容。调试器的工作流程与下载程序相似。

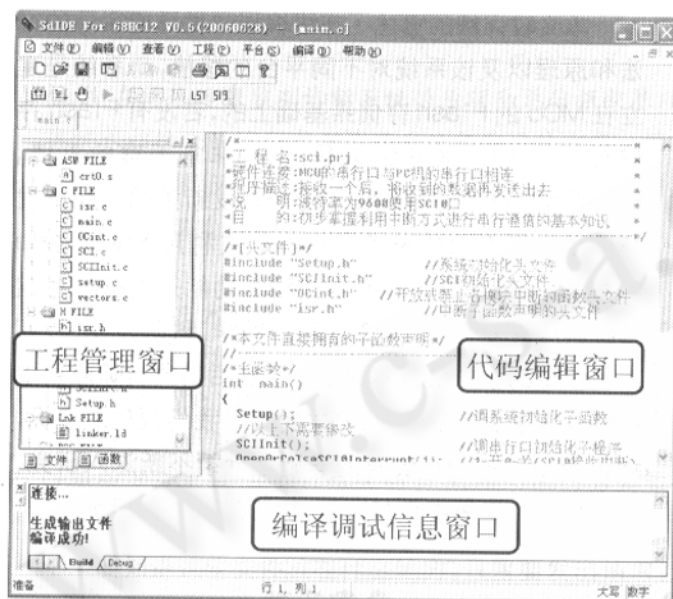


图 2 可重定目标集成开发环境(SdIDE)主界面

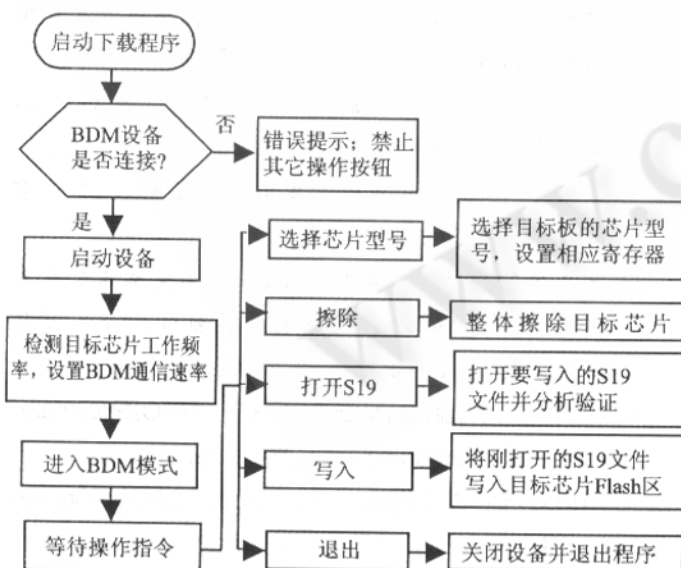


图 3 下载程序流程图

(7) 帮助支持系统和示例程序。考虑到用户能快速的掌握该软件的操作和 M68HC12 C 语言和汇编编程,我们还专门制作了软件的操作手册放在帮助菜单中,里面包含了集成开发系统的所有功能的介绍和使用以及一份教程。除此之外,附带的示例程序可以帮助用户快速的掌握 M68HC12 C 语言和汇编编程,示例程序包括有:普通 I/O 口、SCI、TIMER、A/D、PWM 等模块的基本程序代码。

3 GCC 编译器的基本功能及其调用的方法

本系统使用的是由 GNU 移植的 GCC for M68HC12 的编译器,该软件是开源并且免费的,可以针对 MC68HC11 和 MC68HC12 以及 HCS12 进行 C 和 C++ 源码的编译,而且执行效率高,功能强。调用编译器的方法:只要将该编译器目录放在该系统的安装目录下,采用线程管道的方法调用该编译器目录下的 bin 文件夹中的 make.exe - f 命令来对系统工程文件中的 makefile 脚本文件(是我们针对 M68HC12 编译器编写的)进行编译。该编译器是 win32 控制台程序,输出的编译信息都是在控制台下的。编译器的输出信息都是非标准的输出流,无法使用 ShellExecute 函数捕获,也无法输出到管道文件中。我们采用 CreateThread 的方法,也就是创建子线程的办法,通过线程管道,将编译输出信息捕获,并可显示出来。如果捕获的编译输出信息出现任何错误,包括源文件语法错误和其他错误,编译、链接操作立刻终止,并在输出窗的 Build 子窗口中提示错误。如果是语法错误,那么用户可以通过双击错误提示行,来定位引起错误的源文件行,而且用语法高亮显示该行。

4 更换编译器的步骤

在该系统中更换编译器的步骤如下:

- (1) 把新的编译器目录放在该系统的安装目录下。
- (2) 将系统源程序中调用编译器中相应命令的路径更改为新编译器中相应命令的路径。如要将 MMC2107 的编译器(编译器的目录名: m6811)更改为 MMC2107 的编译器(编译器的目录名: mcore),则只要将程序中的调用 m6811 中相应命令的路径改写成

调用 mcore 中相应命令的路径即可。

(3) 更换 makefile 脚本文件。主要是对其中的有关编译器命令的路径进行更换。其他的基本上可以不变动。

5 更换芯片的步骤(如从 M68HCS12 平台更换为 ARM 平台)

在该系统中更换平台的步骤如下:

(1) 更换编译器,步骤如上节所示。

(2) 在工程设置中来更换目标 MCU,从而选择相应 MCU 编译器的编译参数(因为不同编译器的编译参数不同),使之对新芯片有效。

(3) 该系统的安装目录下,有个工程模板文件夹 Stationary,在该文件中存放相应芯片运行环境和初始化之类的通用源程序。在新建工程时,系统会自动把 Stationary 中的相应芯片的通用源程序拷到新建的工程目录中去,用户只要在工程中添加其他的所需文件即可。因而要为新的目标平台创建一个相应芯片的通用源程序模板,放在 Stationary 中。如从 M68HCS12 芯片更换为 AT91RM9200 芯片时,在 Stationary 中要加入一个 AT91RM9200 通用源程序模板。这样我在新建工程时可在目标平台中增加一个以该芯片命名的新模板来建该芯片的工程。

6 总结

可重定目标集成开发环境,除了要能实现基本的编辑、工程管理、编译、程序下载、调试等功能外,还要考虑到系统便于移植的功能,所以,我们在编写系统模块时,使模块与模块间尽可能的拥有独立性和可移植性。与编译器有关的模块,尽量把调用编译器命令的路径用全局变量来定义,在工程设置对话框中与编译器有关的参数通过设置下拉框来选定,目标平台的选择也通过设定下拉框或菜单来选择。这样就不会在程序代码中对与编译器和目标平台有关的模块写死,从而方便系统的移植。为了增加可移植性,最主要的是要安排好如何编写 makefile 脚本文件,特别是其中与编译器有关的路径,和 MCU 有关的参数要用全局变量

来定义。makefile 是编程人员和 make 之间的接口,本设计通过写 makefile 文件来说明各模块的依赖关系,makefile 文件要严格按其书定规则来编写。在建立 makefile 之后,就可以使用 make 命令来完成所需的编译工作。

7 结束语

本文较详细的描述了在线编程开发环境实现的方法和原理以及该系统对不同平台的移植,此系统是建立在 MCU 的 Flash 存储器基础上的,若没有 Flash 存储器,实现嵌入式在线编程,则是非常困难的事情。Flash 存储器的在线编程技术为开发新型的集成开发环境提供了有力的技术支持。本文利用 FreeScale 的 MCU 所具有的 Flash 在线编程功能开发的可重定目标集成开发环境,本意是用于教学、科研开发,并可方便移植到新的目标芯片上去。在后期的实际应用中已经发挥了良好的作用,基本实现了我们预期的目标,这也鼓励了我们在现有的基础上继续不断深入研究,使该集成开发环境更加完善,功能更加的多和新。

参考文献

- 1 王宜怀,嵌入式应用在线编程开发系统的研制,《计算机工程》2002 年第 12 期。
- 2 李林功、李继凯、谷金宏,嵌入式系统集成开发环境的构成,《计算机工程》2001 年第 5 期。
- 3 邵贝贝 编著,单片机嵌入式应用的在线开发方法,清华大学出版社,2004。
- 4 田泽 编著,嵌入式系统开发与应用实验教程,北京航空航天大学出版社,2004。
- 5 第三届 Motorola 杯单片机设计应用大奖赛竞赛委员会编,第三届 Motorola 杯单片机设计应用大奖赛论文集,电子产品世界特刊,2000 年 1 月。
- 6 John E. Swanke, Visual C++ MFC Programming by Example,机械工业出版社。
- 7 Turbo BDM Light interface. 2005, Daniel Malík