

# 基于模型的设计及其 嵌入式实现

(第2版)

刘杰 编著



北京航空航天大学出版社  
BEIHANG UNIVERSITY PRESS

# 基于模型的设计及其嵌入式实现(第2版)

刘杰 编著

北京航空航天大学出版社

## 内 容 简 介

本书以基于模型的设计为主线,讲述了M代码的快速编写与调试,用户界面的创建,浮点Simulink/Stateflow模型的建立、调试与验证,用户自定义模块的生成;详细介绍了基于模型设计的全过程,主要有需求分析与跟踪、模型的检查与设计验证、浮点模型到定点模型的转换、模型嵌入式C代码的自动生成、软件/处理器/硬件在环测试,整个过程满足DO-178B航空电子规范,可显著提高工作效率、降低开发成本,并且增强了代码的安全性与鲁棒性,避免了产品开发的潜在市场风险。

本书可作为汽车电子、航天军工、通信与电子信息、电力等领域的工程师从事嵌入式开发的技术手册,也可作为高校电类专业嵌入式系统开发与基于模型设计的教材,还可作为学习MATLAB Simulink/Stateflow的参考书。

### 图书在版编目(CIP)数据

基于模型的设计及其嵌入式实现 / 刘杰编著. -- 2  
版. -- 北京:北京航空航天大学出版社,2017.1

ISBN 978-7-5124-2310-7

I. ①基… II. ①刘… III. ①微控制器—程序设计  
IV. ①TP332.3

中国版本图书馆CIP数据核字(2016)第278811号

版权所有,侵权必究。

### 基于模型的设计及其嵌入式实现(第2版)

刘 杰 编著

责任编辑 孙兴芳

\*

北京航空航天大学出版社出版发行

北京市海淀区学院路37号(邮编100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱:emsbook@buaacm.com.cn 邮购电话:(010)82316936

涿州市新华印刷有限公司印装 各地书店经销

\*

开本:710×1000 1/16 印张:31 字数:661千字

2017年3月第2版 2017年3月第1次印刷 印数:3000册

ISBN 978-7-5124-2310-7 定价:79.00元

若本书有倒页、脱页、缺页等印装质量问题,请与本社发行部联系调换。联系电话:(010)82317024

# 前 言

目前,市场上已经很难找到一款不带有嵌入式控制器件的电子、机电产品了,含有嵌入式系统的产品已深入到我们工作与生活的方方面面。同时,人们对产品的安全性、实时性、可操作性及特定功能等的要求也越来越高,这大大增加了嵌入式系统的复杂性和开发难度。

要在这日趋激烈的市场竞争中占据有利地位,开发出高质量的产品,传统的项目开发方法已很难满足这些需求。这是因为传统项目开发的方法一般分为 4 个步骤:

## 1. 需求分析与技术规范阶段

一般用纸质文档或电子 Word 文档写成,系统工程师团队以此为依据进行概念和算法研究,评估技术规范的可行性。对于厚厚的技术文档,每个系统工程师对需求和技术规范的理解难免会存在偏差。NASA 的研究报告指出:“在需求分析阶段产生的错误占整个开发错误的 50% 以上”,这就给后期的项目开发带来了诸多隐患。

## 2. 设计阶段

硬件开发工程师团队根据系统工程师给出的评估报告,设计制作原型样机(如汽车、航空航天器、电路板等),项目的前期投入巨大、开发周期长,并且不能保证所制作的原型样机一定能满足技术规范。

## 3. 实现阶段

软件开发工程师团队根据需求与技术规范,在原型样机上手工编写 C 代码或汇编代码来实现技术规范所要求的技术指标。这一阶段需要精通软件的编程人员花费大量的时间编制程序、查错、调试及验证,明显增加了工作量,延长了研制周期。此外,手工编写的代码良莠不齐,降低了软件运行的可靠度,增加了代码错误的可能性,给新产品上市带来了风险。

## 4. 测试与验证阶段

原型样机制造完成后,才能对产品进行测试与验证,只要上述任何一个环节出现偏差都会导致产品开发的失败,这也是传统项目开发最大的弊病。因此,传统的项目开发方法往往重复多次才能成功,开发风险巨大。

为了解决上述问题,工程师必须找到一种以更快速度、更有效率的方法来开发产品,而基于模型的设计就是解决上述问题的一种方法。基于模型的设计始于 20 世纪

90年代初的汽车制造和航空航天工业,这些行业需要使用大量的微处理器单元,因此工程师最先发现了采用建模与仿真的方法来开发嵌入式系统的巨大优势。到了20世纪90年代中期,控制算法仿真技术的发展催生了自动代码生成技术。模型仿真和自动代码生成技术在这些行业得到了成功的应用,人们清楚地认识到它在嵌入式系统开发中的经济性和高效率。这样,基于模型的设计为工程师提供了一种通用的开发与测试平台,使得具有不同工程背景的工程师之间建立起更好的联系,使得开发高集成度的复杂系统成为可能。

目前,国际上流行的基于模型设计的软件主要有SCADE和MATLAB,它们都被成功地应用于大型项目的开发上,例如,欧洲的空客380、美国的GM混合动力车、诺马公司的联合攻击机等项目。由于MATLAB已经成为一种近乎完美的高度集成化的开放式开发平台,在科学计算与建模方面处于不可替代的领先地位,加之拥有国内众多的用户,因此,本书以MATLAB软件为例来讲述基于模型设计的方法。

MathWorks公司的Simulink/Stateflow/Embedded MATLAB等工具使得工程师可以在一个可视化的交互开发测试平台上进行基于模型的设计,工程师还可以利用直观的模块图对系统模型和子系统设计进行可视化处理。

基于模型的设计对应传统的设计方法同样分为4个步骤:

### 1. 可执行、可跟踪的技术规范

在基于模型的设计方法中,系统工程师首先要建立一个系统模型,即通过数学模型来精确、无歧义地描述用户的需求,创建一个可执行、可跟踪的技术规范。工程师可以通过该系统模型动态地确认系统性能。这相对于传统的纸质和电子Word文档描述的需求与技术规范有明显的优势,它使得开发团队中的每个成员都能够无歧义地理解并运行该模型,从而可以更加专注于开发主要模型的各个部分,不会因理解的不同而造成需求的丢失、冗余或冲突。

### 2. 生成定点模型

系统模型与需求之间可建立双向链接,在整个开发过程中,软件工程师可以对模型进行需求追踪和测试,从而将产品的缺点暴露在产品开发的初期。根据具体的嵌入式器件和实现条件,对系统模型进行细化与功能重分区,此后重新进行系统测试、设计测试和模型助手测试,验证是否满足需求与技术规范,判断是否还存在缺失的需求;验证是否符合特殊的行业标准(如DO-178B、IEC-61508、MAAB等),之后再对模型作定点转换,形成简洁、高效的定点模型。

### 3. 嵌入式代码的自动生成

MathWorks公司的Real-Time Workshop Embedded Coder可以将Simulink/Stateflow中的模型自动转换为嵌入式C代码,大大降低了嵌入式系统的开发门槛,因为毕竟不是每个工程师都是编写代码的高手。开发人员可以在Simulink/Stateflow、Embedded MATLAB中建立系统模型,构思解决方案,然后使用RTW-EC自动生成优化的、可移植的、自定义的产品级C代码,并根据特定的目标配置,自动生

成嵌入式系统实时应用程序,这就缩短了开发周期,同时避免了人为引入的错误。

#### 4. 连续的测试和验证

基于模型的设计在整个设计过程中都在不断地进行测试和验证,工程师利用测试用例追踪系统级模型和需求,检测设计变更导致的系统输出变化,并快速追踪到变更的来源,通过测试用例还能够了解系统模型的功能覆盖度。

对于嵌入式系统,还需测试其实时性。工程师可以使用硬件在环测试,检测嵌入式代码的实时性。通过测试,收集实时数据,修改代码参数。硬件在环测试能够确保在开发早期就完成嵌入式软件的测试。当系统整合时,嵌入式软件测试就比传统方法检测得更彻底、更全面,这样可以及早地发现问题,大大降低解决问题的成本。

本书对第1版的内容做了较大幅度的增减(修改率约为50%),并采用 MATLAB R2015b 版和 CCS 5x 作为软件开发平台。

翁公羽、孙瑶瑶全程参加了本书的资料整理和撰写工作,郑仁富、罗兵、周宇博、陈添丁、郑明魁、李涵、胡步发也参加了本书的策划和个别小节的编写工作,在此一并表示感谢。

由于作者水平有限,错误和纰漏在所难免,诚望广大读者批评指正。

作者

2017年1月于大怡园

# 目 录

第 1 章 搭建软件开发环境	1
1.1 下载与安装所需的软件	1
1.1.1 下载开发软件包	1
1.1.2 安装开发软件包	1
1.1.3 安装更新	4
1.2 创建一个包含 DSP/BIOS 的 C6000 DSP 工程	5
1.3 设置 MATLAB R2015b 与 CCS 5.11 数据链配置	14
1.3.1 checkEnvSetup()	14
1.3.2 xmakefilesetup	16
1.4 有关 MATLAB R2015b 与 CCS 3.3 的配置问题	18
第 2 章 MATLAB 高级应用基础	19
2.1 MATLAB 的功能简介	20
2.1.1 函数浏览器	20
2.1.2 函数提示	21
2.1.3 目录浏览器	22
2.1.4 文件交换服务	24
2.2 M 文件	26
2.2.1 M 文件结构	26
2.2.2 清理程序	28
2.2.3 创建 M 文件	29
2.2.4 M 脚本文件	29
2.2.5 M 函数	30
2.2.6 匿名函数	34
2.3 加快 M 文件的编写	36
2.3.1 什么是代码检查器	36
2.3.2 代码检查器的使用方法	36
2.3.3 代码检查器实例	36

2.4	加快 M 文件的调试——cell .....	40
2.4.1	什么是 cell .....	40
2.4.2	cell 的定义与删除 .....	40
2.4.3	cell 调试实例 .....	42
2.4.4	应 用 .....	44
2.5	数据存取 .....	47
2.5.1	生成 MAT 文件 .....	47
2.5.2	加载 MAT 文件 .....	49
2.5.3	读/写音视频文件 .....	50
2.6	代码效率分析 .....	53
2.7	MATLAB Coder 简介 .....	55
2.7.1	MATLAB Coder 支持/不支持生成 C 代码的类型 .....	56
2.7.2	MATLAB Coder 的使用要求 .....	57
2.7.3	Embedded Coder 的常用命令 .....	57
2.7.4	C 编译器的设置 .....	58
2.7.5	应用实例 .....	59
<b>第 3 章</b>	<b>图形用户界面简介 .....</b>	<b>72</b>
3.1	GUIDE 简介 .....	72
3.1.1	GUIDE 界面简介 .....	72
3.1.2	获取当前图形对象句柄的常用函数 .....	75
3.1.3	Callback 函数 .....	76
3.2	基于 GUIDE 工具的实例 .....	76
3.2.1	读取图像的 GUI 实例 .....	76
3.2.2	制作及发布简易计算器 .....	94
<b>第 4 章</b>	<b>Stateflow 原理与建模基础 .....</b>	<b>106</b>
4.1	Stateflow 概述 .....	107
4.1.1	状 态 .....	111
4.1.2	迁 移 .....	114
4.1.3	事 件 .....	118
4.1.4	数据对象 .....	120
4.1.5	条件与动作 .....	122
4.1.6	节 点 .....	122
4.2	流程图 .....	128
4.2.1	手动建立流程图 .....	128



4.2.2	快速建立流程图 .....	131
4.2.3	车速控制 .....	132
4.3	状态图的层次 .....	136
4.3.1	历史节点 .....	138
4.3.2	迁移的层次性 .....	139
4.3.3	内部迁移 .....	140
4.4	并行机制 .....	143
4.4.1	广播 .....	143
4.4.2	隐含事件 .....	149
4.4.3	时间逻辑事件 .....	150
4.5	其他的图形对象 .....	152
4.5.1	真值表 .....	152
4.5.2	图形盒 .....	155
4.5.3	图形函数 .....	156
4.6	MATLAB 函数 .....	157
4.6.1	建立调用 MATLAB 函数的 Simulink 模型 .....	157
4.6.2	编写 MATLAB 函数 .....	159
4.6.3	调试 .....	160
4.7	Simulink 函数 .....	163
4.7.1	Simulink 函数的使用 .....	163
4.7.2	使用 Simulink 函数需遵循的规则 .....	169
4.8	集成自定义代码 .....	170
4.9	Stateflow 建模实例——计时器 .....	174
<b>第 5 章 Simulink 建模与验证</b> .....		<b>184</b>
5.1	Simulink 的基本操作 .....	185
5.1.1	启动 Simulink .....	185
5.1.2	Simulink 模块库简介 .....	186
5.1.3	模块操作 .....	188
5.2	信号采样误差 .....	193
5.2.1	信号源 .....	193
5.2.2	MATLAB 工作空间 .....	198
5.2.3	用户自定义函数 .....	202
5.2.4	非线性系统 .....	204
5.2.5	离散模块 .....	207
5.2.6	采样误差 .....	209

5.2.7	建立子系统 .....	211
5.2.8	封装子系统 .....	212
5.2.9	数据类型匹配 .....	215
5.2.10	模型信息 .....	218
5.2.11	模型元件化 .....	221
5.2.12	自定义模块库 .....	222
5.3	音频信号处理 .....	224
5.3.1	仿真环境 .....	224
5.3.2	基于采样的模型 .....	225
5.3.3	帧结构 .....	228
5.3.4	基于帧结构的模型 .....	228
5.3.5	信号缓冲器 .....	230
5.4	视频监控 .....	232
5.4.1	原 理 .....	233
5.4.2	SAD 子系统 .....	233
5.4.3	阈值比较 .....	234
5.4.4	视频记录子系统 .....	235
5.4.5	源视频帧计数及显示 .....	236
5.4.6	数据读取与显示 .....	237
5.4.7	实验结果 .....	239
5.5	模型调试 .....	241
5.5.1	图形调试模式 .....	241
5.5.2	命令行调试模式 .....	244
5.5.3	调试过程 .....	245
5.5.4	断点设置 .....	249
5.5.5	显示仿真及模型信息 .....	253
5.6	模型检查与验证 .....	260
5.6.1	使用系统检查器——Model Advisor 检查模型 .....	260
5.6.2	建立测试用例 .....	269
5.6.3	模型覆盖度分析 .....	279
5.6.4	模型效率分析 .....	285
<b>第 6 章 用户驱动模块的创建 .....</b>		<b>289</b>
6.1	什么是 S-Function .....	289
6.1.1	S-Function 的工作机制 .....	291
6.1.2	函数回调方法 .....	292

6.1.3	编写 C MEX S-Function .....	295
6.1.4	Simulink 引擎与 C S-Function 的相互作用 .....	300
6.1.5	TLC 文件 .....	309
6.1.6	LEVEL-2 M 文件 S-Function 介绍 .....	313
6.1.7	调用仿真模型外部的 C 代码和生成代码 .....	324
6.2	S-Function Builder .....	327
6.2.1	S-Function 名及参数名 .....	328
6.2.2	初始化 .....	329
6.2.3	数据属性 .....	329
6.2.4	库文件 .....	332
6.2.5	输 出 .....	333
6.2.6	连续状态求导 .....	336
6.2.7	离散状态更新 .....	337
6.2.8	编译信息 .....	338
6.2.9	应 用 .....	340
6.3	MATLAB Function 模块 .....	342
6.3.1	MATLAB Function 模块的生成方法 .....	343
6.3.2	集成用户自定义的 C 代码 .....	347
6.4	实 例 .....	348
6.4.1	IIR 滤波器 .....	348
6.4.2	S-Function 的参数设置与封装 .....	351
6.4.3	读取数据文件 .....	357
<b>第 7 章</b>	<b>嵌入式代码的快速生成</b> .....	<b>362</b>
7.1	利用 Embedded Coder 生成 DSP 目标代码 .....	362
7.2	CCS 5/6 与 MATLAB R2015b 的数据链配置 .....	364
7.3	TI DSP 原装板的实时代码生成 .....	368
7.4	代码验证 .....	375
7.5	TI C6416 DSK 目标板应用例程 .....	380
7.6	用户自定义目标板的应用 .....	401
<b>第 8 章</b>	<b>基于模型的设计</b> .....	<b>406</b>
8.1	传统设计过程与基于模型设计过程的对比 .....	407
8.2	DO-178B 标准简介 .....	409
8.2.1	什么是 DO-178B 标准 .....	409
8.2.2	DO-178B 标准验证要求 .....	410

# 目 录

8.2.3 DO-178B 软件生命周期 .....	411
8.3 基于模型设计的工作流程 .....	412
8.4 需求分析及跟踪 .....	417
8.4.1 根据需求建立系统模型 .....	417
8.4.2 建立需求与模块间的关联 .....	418
8.4.3 一致性检查 .....	421
8.5 模型检查及验证 .....	423
8.5.1 Model Advisor 检查 .....	423
8.5.2 SystemTest .....	424
8.5.3 Design Verifier .....	433
8.6 定点模型 .....	439
8.7 软件在环测试 .....	447
8.8 处理器在环测试 .....	448
8.9 代码跟踪 .....	449
8.10 硬件模型 .....	453
8.10.1 建立硬件模型 .....	453
8.10.2 模块设置 .....	454
8.11 代码优化及代码生成 .....	457
8.11.1 子系统原子化 .....	457
8.11.2 优化模块库 .....	460
8.11.3 指定芯片 .....	461
8.11.4 代码检查 .....	462
8.11.5 IDE 环境下的代码优化 .....	464
8.11.6 工程选项及代码生成 .....	465
8.12 代码有效性检查原理 .....	469
8.13 硬件在环测试 .....	472
8.13.1 建立 PC 端模型 .....	472
8.13.2 模块参数设置 .....	473
8.13.3 实施硬件在环测试 .....	476
8.13.4 代码效率剖析 .....	477
8.13.5 内存使用分析 .....	478
参考文献 .....	480

# 第 1 章

## 搭建软件开发环境

本章将简单介绍 CCS 5.11 及其 C64 支持软件包的下载、安装及使用方法。

### 1.1 下载与安装所需的软件

下面将扼要介绍 CCS 5.11 及其 C64 支持软件包的下载与安装。

#### 1.1.1 下载开发软件包

CCS 5.11 软件的下载地址：

[http://processors.wiki.ti.com/index.php/Download\\_CCS](http://processors.wiki.ti.com/index.php/Download_CCS)。

C6xCSL(TMS320C6000 Chip Support Library)(sprc090)的下载地址：

<http://www.ti.com/tool/sprc090>。

c64plus(TMS320C64x Image Library)(sprc094)的下载地址：

<http://www.ti.com/tool/sprc094>。

说明：这里仅以 C6416 和 C6713 DSP 所需的支持软件包为例，其他型号的 C6000 DSP 芯片所需的支持软件包请按提示自行在 TI 网站上下载。

#### 1.1.2 安装开发软件包

##### 1. CCS 5.11 软件的安装

CCS 5.11 软件的安装步骤如下：


① 按默认路径安装，如图 1-1 所示。

② 手动选中与 C6000 DSP 有关的芯片(见图 1-2)，默认条件下该芯片为取消选中状态。

③ 按提示完成其他步骤即可完成 CCS 5.11 的安装。

##### 2. C6xCSL 支持软件包的安装

C6xCSL 支持软件包的安装步骤如下：

① 单击 sprc090 软件包中的图标启动 C6xCSL 安装。

② 在弹出的对话框中单击 Next 按钮，如图 1-3 所示。

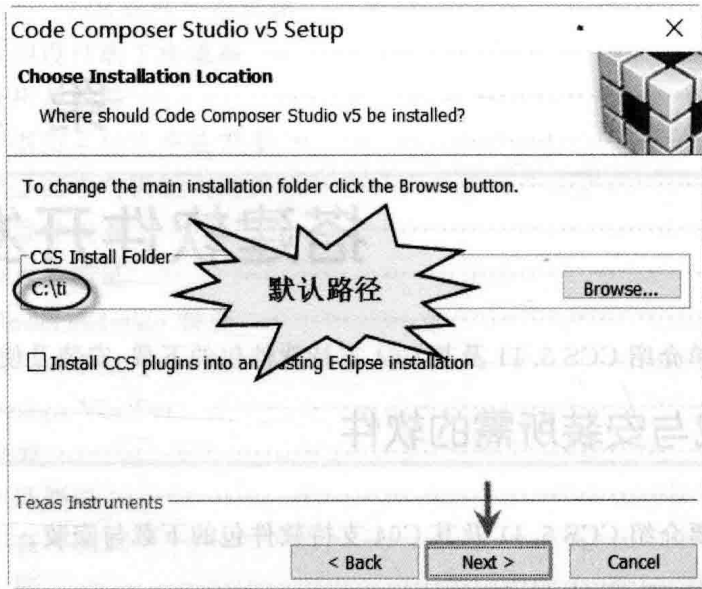


图 1-1 按默认路径安装

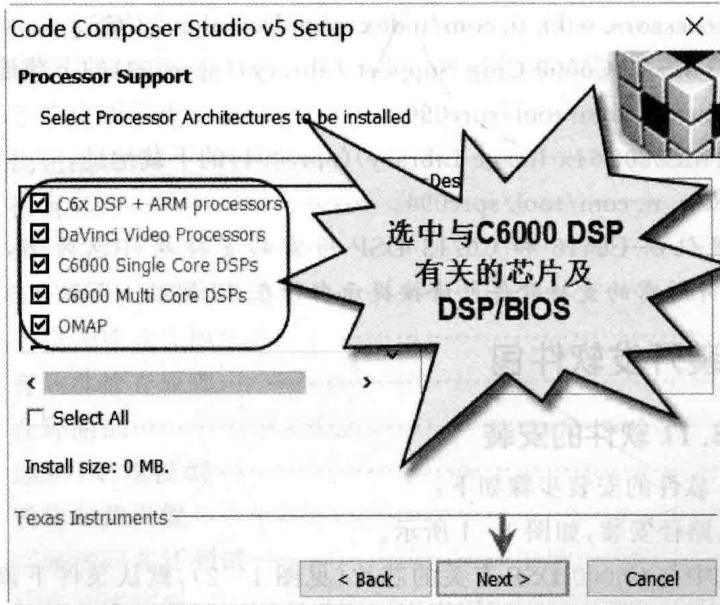


图 1-2 选中与 C6000 DSP 有关的芯片

③ 在弹出的对话框中选中“*Yes, I agree with all the terms of this license agreement*”复选框,然后单击 *Next* 按钮,如图 1-4 所示。

④ 在弹出的对话框中指定 C6xCSL 的安装目录,然后单击 *Next* 按钮完成安装,如图 1-5 所示。



图 1-3 单击 Next 按钮

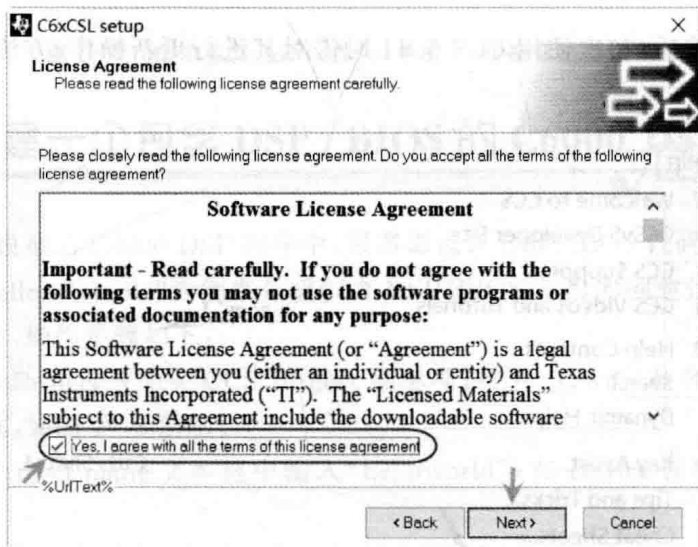


图 1-4 选中“Yes, I agree with all the terms of this license agreement”复选框

### 3. c64plus 支持软件包的安装

c64plus 支持软件包的安装步骤如下：

- ① 单击 sprc094 软件包中的 C64xIMGLIB\_v104b.exe 启动 c64plus 安装。
- ② 安装目录选择 C:\ti, 其他步骤参见 C6xCSL 支持软件包的安装步骤。

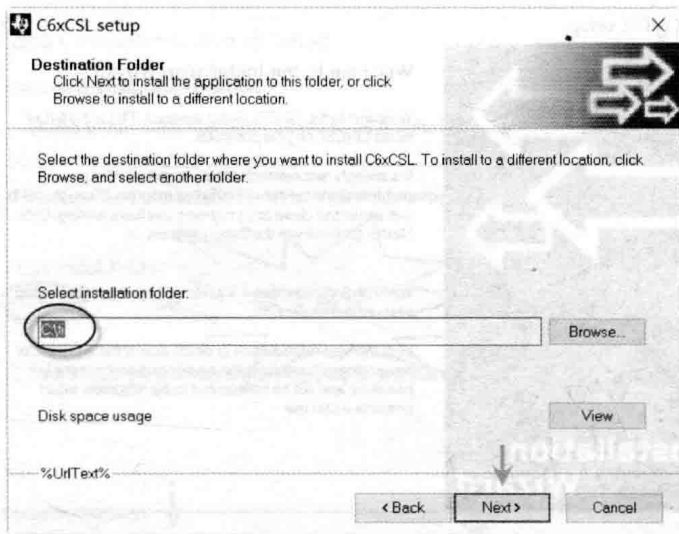


图 1-5 指定 C6xCSL 的安装目录

## 1.1.3 安装更新

在安装完成后,初次使用 CCS 5.11 时需对其进行更新操作,方法如图 1-6 和图 1-7 所示。

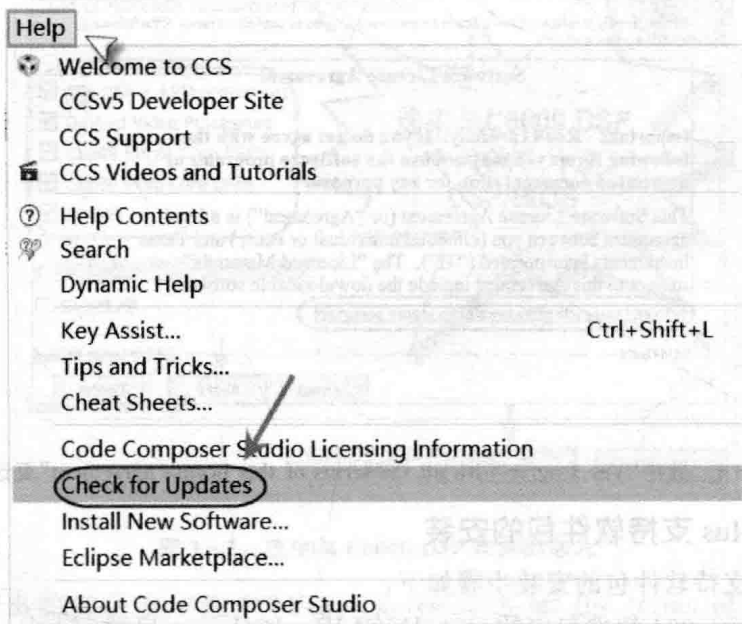


图 1-6 更新操作



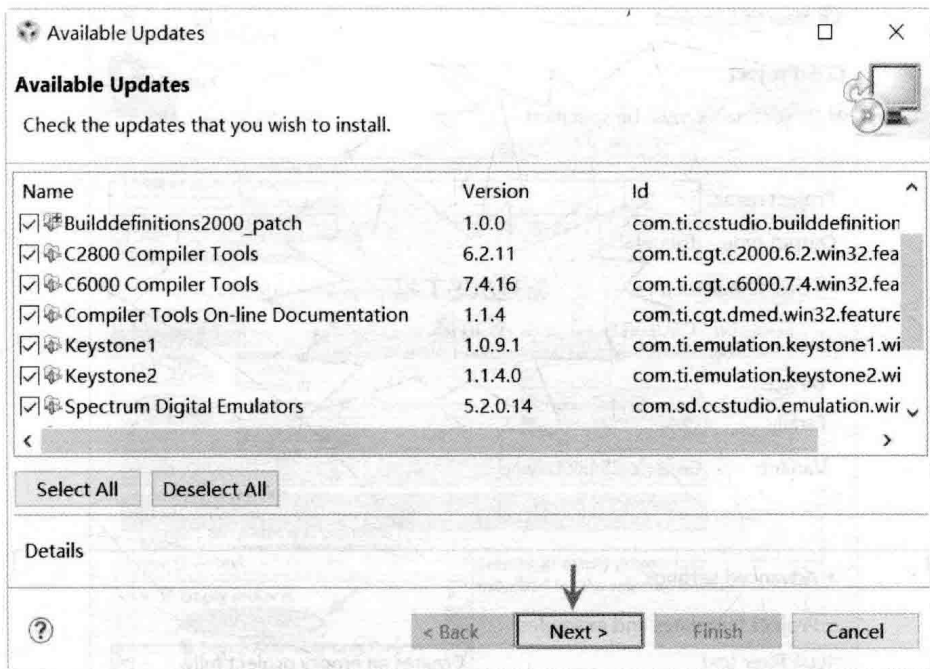


图 1-7 选中需要更新的部件,单击 Next 按钮进行更新操作

## 1.2 创建一个包含 DSP/BIOS 的 C6000 DSP 工程

在基于模型的 C6000 DSP 程序中,很多都包含 DSP/BIOS 代码。这里将以一个最简单的“hello example”为例来介绍包含 DSP/BIOS 工程的创建过程(实验平台为 C6455 DSK),操作步骤如下:

① 选择 Project→New CCS Project 菜单项,打开 New CCS Project(新建 CCS 工程)对话框,如图 1-8 所示。

② 在 Project name 文本框中输入“helloworld”;在 Device 选项组中进行如下操作:

第一,在 Family 下拉列表框中选择 C6000。

第二,在 Variant 下拉列表框中选择 Generic C64x+Device。

③ 单击 Project templates and examples,在下拉列表框中选择 Empty Project,创建一个空工程,如图 1-9 所示。

④ 单击图 1-9 中的 Finish 按钮完成新的 CCS 工程的创建。

⑤ 创建.tcf 文件;步骤如下:

第一,选择 File→New→DSP/BIOS v5. x Configuration File 菜单项,创建.tcf 文件,如图 1-10 所示。