

基于 Windows 操作系统的 Code Composer Studio 基础使用教程

软件版本: v8.3

文档版本: v1.2

二零一九 年 八 月

申明

目前，TI 公司生产的可编程器件以广泛应用，作为 TI 官方的 Code Composer Studio 开发软件，越来越多的开发人员与学习者选择使用，但由于该软件语言为英语，对英语不好的用户而言有一定的门槛，特撰写本文档，望对各位有所帮助。本文仅做学习使用，不得作为他用。本文著作权归作者所有。

如有侵权，请联系作者删除。

通江县大晨科技有限公司

2019 年 8 月

前言

随着 TI 公司的不断发展，我国相关行业、各大高校使用该公司的可编程器件越来越多，其中 Code Composer Studio 作为官方的集成开发软件，已有越来越多的用户选择使用，但目前没有较为准确的中文使用方法，以至于该软件中有许多非常实用的工具不被所知，特总结该软件一些基础的使用方法，对于未涉及到的操作方法也可在软件帮助中获取您需要的帮助。文中部分内容无法用文字详细描述，如在使用过程中遇到问题，可通过左下方 QQ 群联系，将竭尽全力为您解答所遇到的问题。

本文是根据作者本人日常使用总结而成，由于水平有限，文中难免有纰漏或不妥之处，如发现文中有误的地方，敬请指出，联系方式见本页左下。

编者

2019 年 8 月

目录

1 功能简介	1
1.1 File（文件）	1
1.2 Edit（编辑）	3
1.3 View（视图）	3
1.4 Navigate（导航）	5
1.5 Project（工程）	6
1.6 Run（运行）	6
1.7 Scripts（脚本）	7
1.8 Window（窗口）	7
1.9 Help（帮助）	7
1.10 Tools（工具）	8
1.11 工具栏图标	8
1.11.1 编程界面工具栏图标	8
1.11.2 调试界面工具栏图标	9
1.12 本章小结	10
2 工作区与工程相关操作	11
2.1 创建/打开工作区	11
2.2 创建工程	13
2.3 构建工程	15
2.4 导入现有工程	15
2.5 导入例程	17
2.6 导出工程	18
2.7 生成 Hex、TXT	20
2.8 添加包含选项	22
2.9 本章小结	23

3 常用工具使用方法	24
3.1 Resource Explorer classic.....	24
3.2 Ccs app center	24
3.3 Terminal.....	25
3.4 Outline	26
3.5 Optimizer assistant	26
3.6 Memory browser	26
3.7 Expressions.....	27
3.8 Registers	28
3.9 Variables	28
3.10 Disassembly.....	28
3.11 Breakpoints.....	29
3.12 添加书签/待办事项	29
3.13 sdk 的启用	30
3.14 编译器的更新	31
3.15 New window.....	32
3.16 本章小结	32
4 个性化设置及操作	34
4.1 字体大小设置	34
4.2 代码颜色设置	35
4.3 背景颜色设置	36
4.4 快捷键	37
4.5 本章小结	38
5 调试/仿真	39
5.1 调试/仿真准备	39
5.2 调试/仿真操作	39
5.3 调试/仿真注意事项	40
5.4 本章小结	40

6 常见问题	41
问题 1 工作区、项目打开/导入类错误	41
问题 2 下载程序完成后自动跳转到仿真界面，但没有仿真	41
问题 3 下载程序时提示错误，但仿真器连接正常	41

淘宝店铺: DC大晨科技

1 功能简介

Code Composer Studio 是一种集成开发环境 (IDE)，支持 TI 的微控制器和嵌入式处理器产品系列。Code Composer Studio 包含一整套用于开发和调试嵌入式应用的工具。它包含了用于优化的 C/C++ 编译器、源码编辑器、工作区构建环境、调试器、描述器以及多种其他功能¹。

ccs 编程软件的主页面如图 1-1 所示，主菜单按照从左至右的排序依次是：File（文件）、Edit（编辑）、View（视图）、Navigate（导航）、Project（工程）、Run（运行）、Scripts（脚本）、Windows（窗口）、Help（帮助），下面简单介绍其功能与作用。

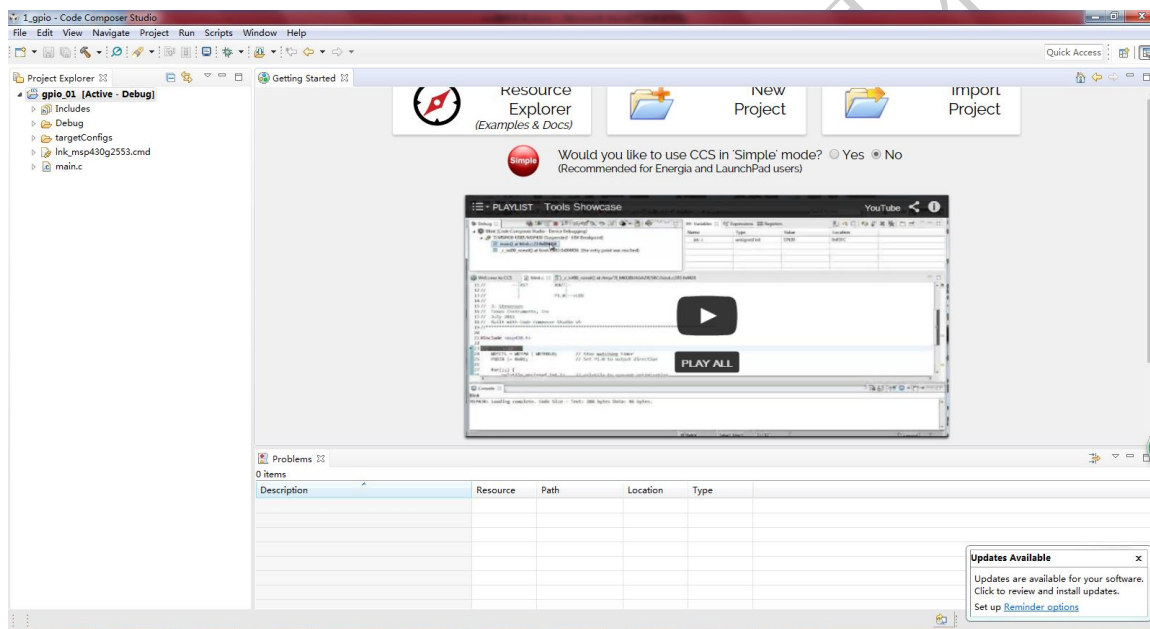


图 1-1 编程界面

1.1 File（文件）

File 路径下的命令主要用于文件的操作，如图 1-2 所示，该路径下用于对工程文件的添加、删除、打开、保存、导入、导出等基本文件操作。

Open file 用于打开外部文件；open project.. 用于导入工程；单击 properties 可以配置当前选中的工程相关设置。

¹ 引用自：<http://www.ti.com.cn/tool/cn/CCSTUDIO>

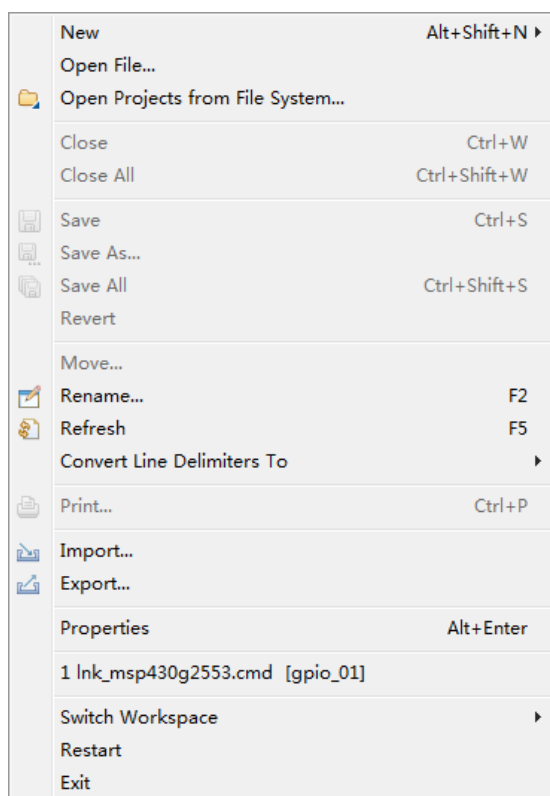


图 1-2 File 路径下相关

如图 1-3 所示，在 File/new 下可以新建工程、.c & .h 文件、文件夹、工程配置文件等。

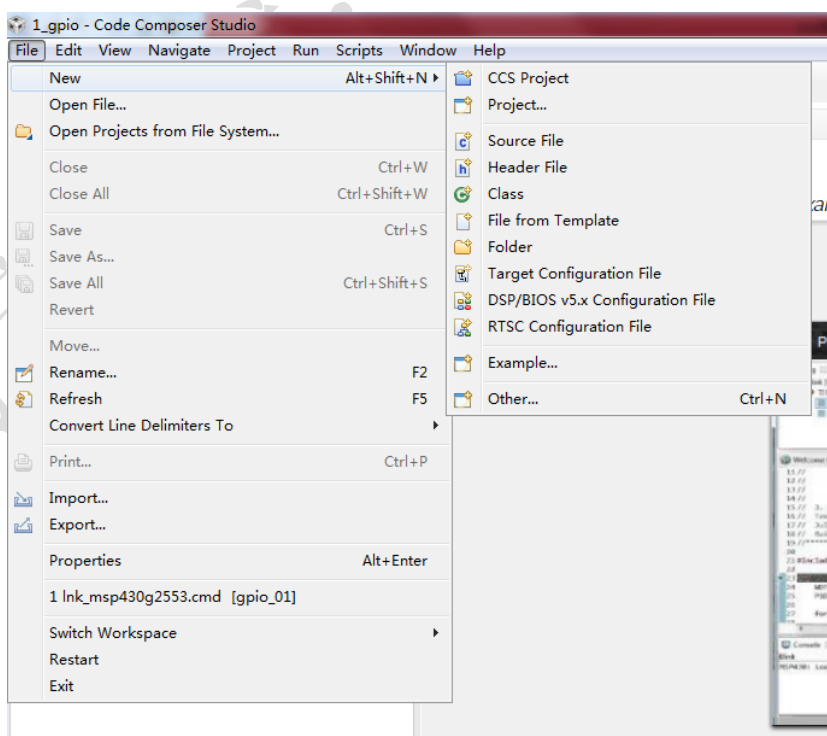


图 1-3 new 路径下的功能

1.2 Edit（编辑）

如图 1-4 所示，在该命令下主要功能是对文本的编辑，例如撤销（undo typing）、复制（copy）、粘贴（paste）、剪切（cut）、查找/替换（Find），添加书签（add bookmark）、任务（add task）等。

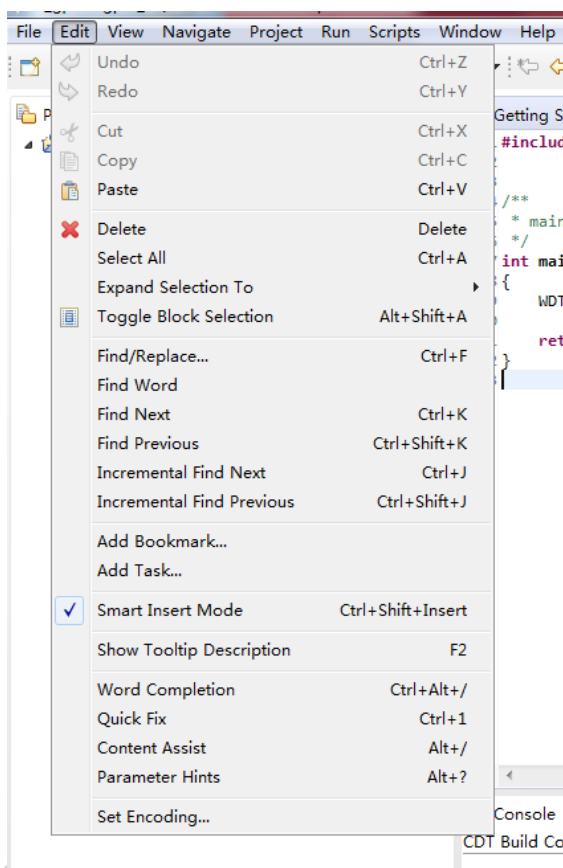


图 1-4 Edit 路径下的功能

1.3 View（视图）

如图 1-5 所示，在该命令下可以管理、打开程序的各类窗口以及窗口程序。

Resource Explorer classic: 管理和查看本地的资源，例如 [MSPware](#)、[Tiva ware](#) 等。

Getting started: 打开初始页面，如图 1-1 所示。

Ccs app center: 打开应用中心，可以下载常用的编译器以及工具等。

Console: 打开控制台，控制台用于显示编译器输出的信息等。

Problems: 当编译工程存在问题时，通过该窗口汇总问题。

Advice: 在编译时生成的建议信息，同 Problems 类似。

Debug: 调试窗口，在仿真时，用于显示调试相关的信息。

Memory browser: 内存浏览器，输入一个地址即可查看该地址的数据。

Registers: 芯片寄存器，可以通过该窗口可在仿真时查看各个模块寄存器的配置情况。

Expressions: 在该窗口中可以添加程序中定义的变量，在仿真时使用可直观的看到数值变化。

Variables: 主要用于显示函数中定义的变量，仿真时使用，退出当前函数则无法查看，可配合 Expressions 窗口使用。

Disassembly: 反汇编窗口，在仿真时可查看相关信息，例如地址、寄存器等。

Breakpoints: 断点窗口，仿真时可在该窗口管理所有的断点，例如断点的名称、类型等信息。

Terminal: 本地终端窗口，可在本窗口中选择命令行功能（cmd.exe）、SSH、串口调试工具、远程登录终端，其中第二和第四普通的开发很少用，可以忽略。

Scripting console: 脚本控制台，通过输入脚本命令，实现相应功能。

Target configurations: Target 窗口，显示本工作区中所有.ccxml 文件，便于管理。

Outline: 显示当前打开的文件中定义的变量、函数、数组等。

Memory allocation: 内存分配，可以查看工程当前内存分配情况。

Optimizer assistant: 程序优化助手，可实现自动优化、并可选择速度权衡或最佳优化，可在芯片内较少或内存不足时使用该功能，在优化完成后可选择应用到本工程。

Other: 其他，各类工具的汇总。在该窗口下可根据分类查找相关的工具。

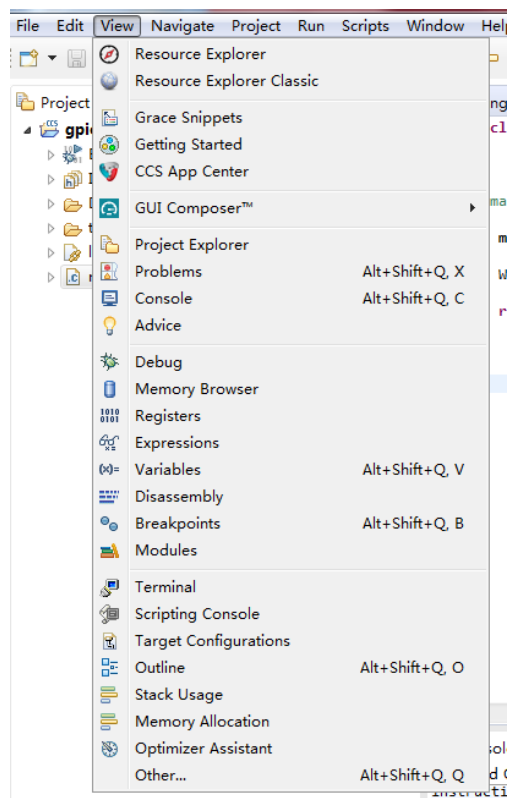


图 1-5 View 路径下的功能

1.4 Navigate（导航）

如图 1-6 所示，在该命令下可快速打开对应功能，例如函数的声明、打开函数，新建注释等功能。

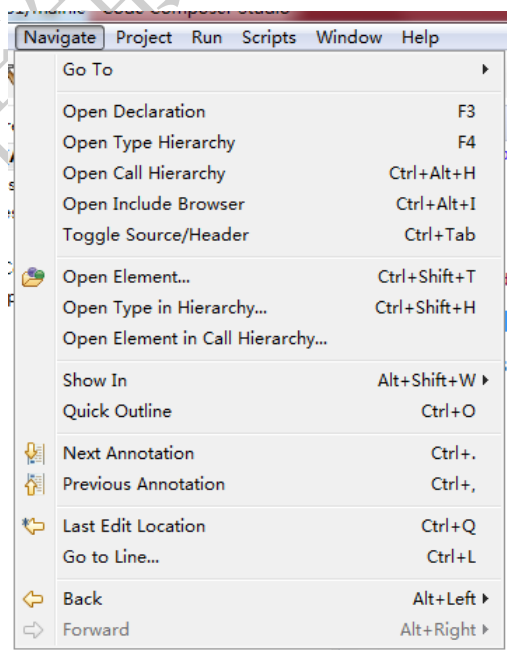


图 1-6 navigate 路径下的功能

1.5 Project（工程）

如图 1-7 所示，该命令下主要用于对工程的管理及其他操作，例如新建、编译、导入、配置工程等。

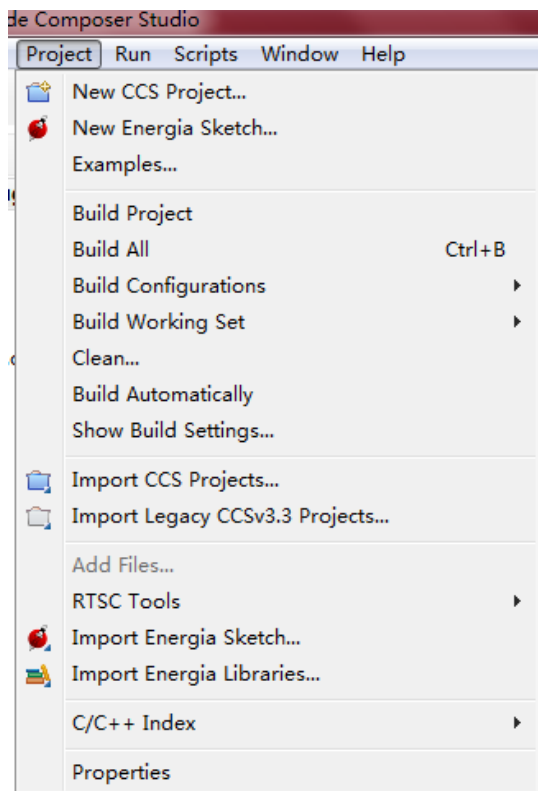


图 1-7 project 路径下的功能

1.6 Run（运行）

如图 1-8 所示，该命令下可以实现工程的下载、仿真等，默认下载当前选择的工程，在 load 中还可对其他外部工程的输出文件进行下载，但必须使用 ti 提供的仿真器调试。

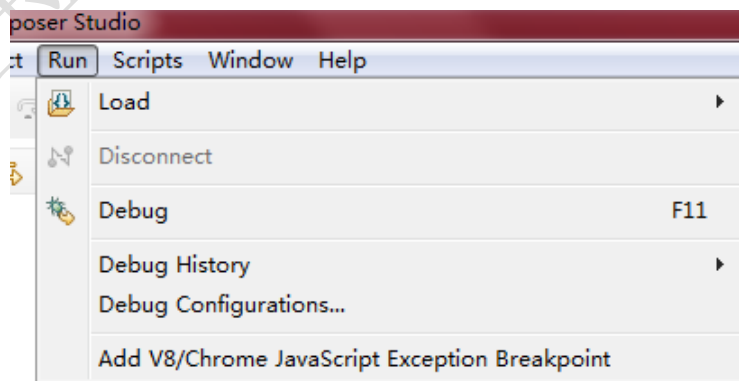


图 1-8 run 路径下的功能

1.7 Scripts（脚本）

Scripts 命令下为可用的脚本工具，如果没有安装则无法使用。

1.8 Window（窗口）

如图 1-9 所示，该命令下主要实现一些与窗口相关的操作，在使用中，一些窗口被关闭可通过：Window\show view\.. 该路径下的命令打开。调试界面通过：Window\perspective\open perspective \ccs Debug 打开。Preferences：在该命令下设置各个组件的基本选项。

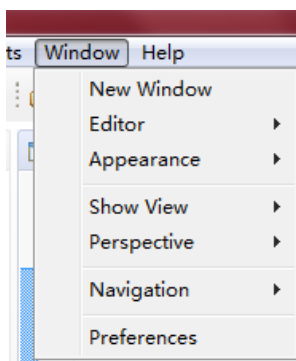


图 1-9 Windows 路径下的功能

1.9 Help（帮助）

如图 1-10 所示，该命令下主要有软件、组件的升级、帮助，编译器、软件的使用说明等。在 Help\ Help Contents 中集中管理编译器、插件的使用说明文件。

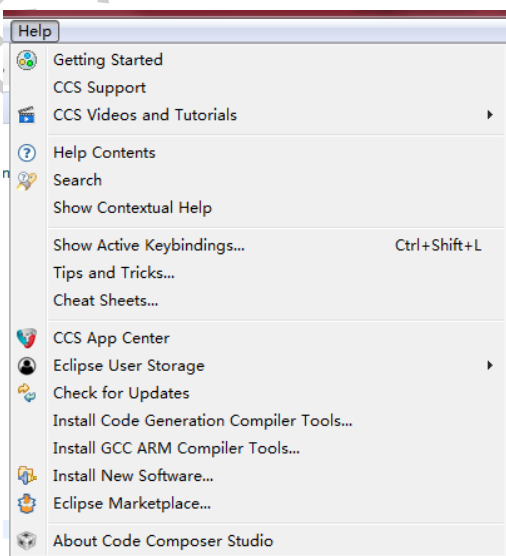


图 1-10 help 路径下的功能

1.10 Tools（工具）

该命令位于调试界面的主菜单中，如图 1-11 所示，该命令下有多种调试工具，主要适配 ti 官方的嵌入式操作系统 TI-RTOS，如不使用 TI-RTOS 实时操作系统可忽略，需要使用的可参阅相关文件。

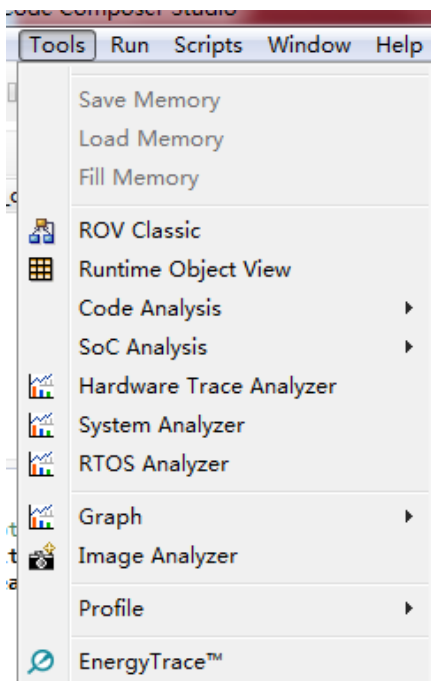


图 1-11 help 路径下的功能

1.11 工具栏图标

1.11.1 编程界面工具栏图标

如图 1-12 所示为编程界面的工具栏图标，下面分别介绍各个功能

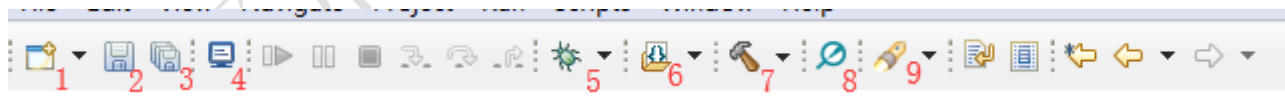


图 1-12 工具栏图标的功能

1: 新建工程、文件、文件夹等。

2: 保存当前编辑的文件。

3: 另存为当前编辑的文件。

4: 终端，常用该功能中的第三个工具：串口终端。

5: debug，单击构建工程并进入调试模式。打开调试界面并开始调试。 它的下拉可用于访问其他启动选项

6: 下载程序。在执行之前, 将程序构建到实际的或模拟的目标板上, 从而生成 COFF 文件(*.out)。

7: 编译。

8: 可以测量和显示应用程序的能源配置文件, 并帮助优化它以实现超低功耗。

9: 搜索。

1.11.2 调试界面工具栏图标

如图 1-13 所示为编程界面的工具栏图标, 下面分别介绍各个功能



图 1-13 工具栏图标的功能

1: 恢复。从当前 PC 位置继续执行当前加载的程序。执行将继续, 直到遇到断点为止。

2: 暂停。调试视图的其余部分将使用最新的目标数据自动更新。

3: 停止。终止所有活动的调试会话。

4: 单步执行。步骤进入突出显示的语句。

5: 跨步执行。跳过突出显示的语句。执行将在同一方法的下一行继续, 或者(如果您在方法的末尾)在调用当前方法的方法中继续。光标跳转到方法的声明并选择这一行。

6: 退出当前执行方法。

7: ROV 工具。

8: 断开/连接。连接到硬件目标。

9: 恢复调试状态。

10: 启用 silicon 实时模式。

11: 启用 polite 实时模式。

12: 软复位、硬复位。重置选定的目标。根据所选设备的不同, 下拉菜单有各种高级重置选项。

13: 重新开始。将 PC 还原到当前加载程序的入口点。如果调试器选项“Run to main on target load or restart”被设置为目标将运行到指定的符号, 否则不会更改目标的执行状态。

14: 添加断点。

15: 不管源代码是否可用，调试器都会执行下一条汇编指令。

16: 调试器将逐步执行单个程序集指令。如果指令是程序集子例程，调试器将执行程序集子例程，然后在程序集函数返回后停止。

17: 刷新调试视图。

1.12 本章小结

本章主要介绍工具栏中的各个命令及其功能，开发人员应当掌握常用的命令打开方法。熟悉常用工具的使用方法。

2 工作区与工程相关操作

Code Composer Studio V8.3 的工程创建大致可分为两部分：工作区 & 工程文件，工作区指的是存放工程的工作区配置文件、工程文件的地方。

如图 2-1 所示，该路径下方框中的文件夹保存的是工作区配置文件，文件夹名称基本固定不变。圆框中则是工程文件，文件夹名称为创建工程时输入的名称，工程中的文件均保存在该路径下。

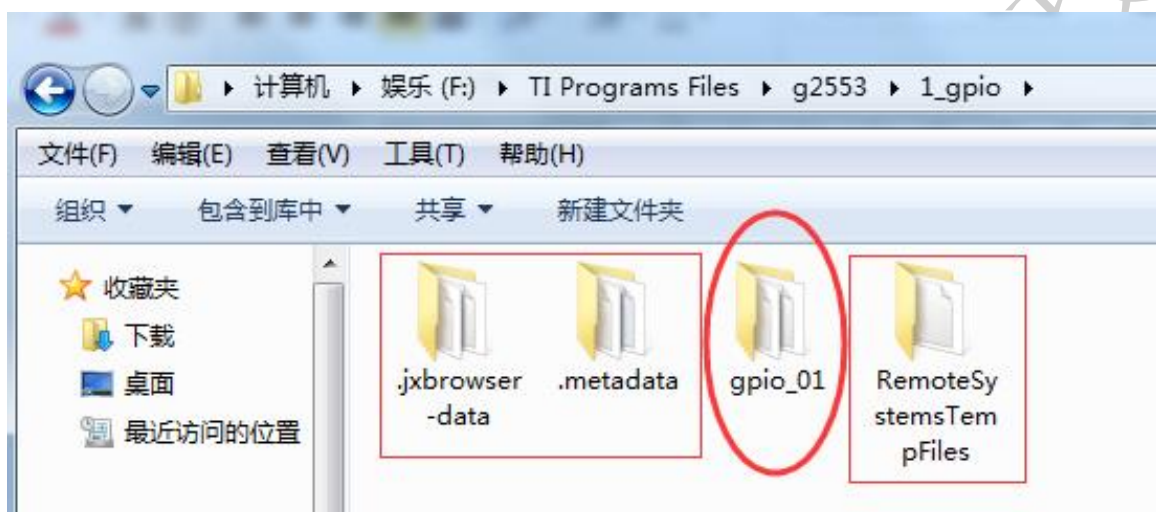


图 2-1 工作区文件夹

如图 2-2 所示，该文件夹下保存着工程的配置文件、调试文件、.C/.H 文件等。

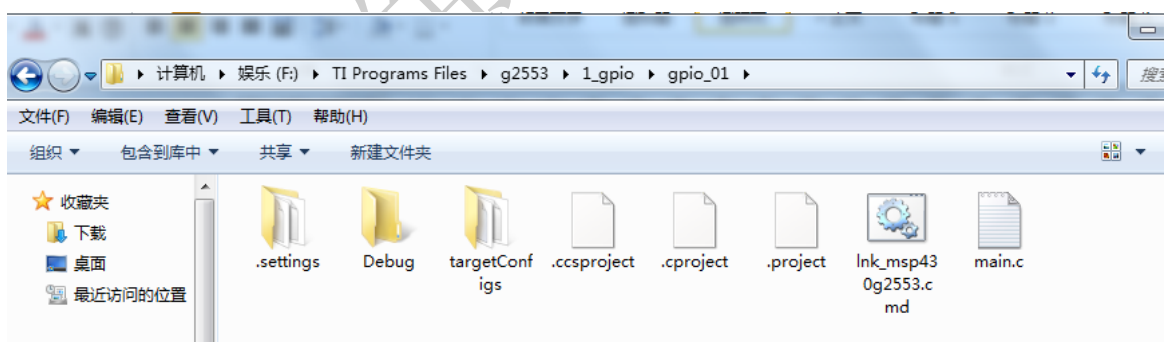


图 2-2 工程文件夹

2.1 创建/打开工作区

打开 Code Composer Studio 8.3 软件，弹出如图 2-3 所示的弹窗，在 workspack 中输入工程路径，单击 launch 进入编程界面。单击弹框的左下角的 **Recent Workspases**，可以看到最近打开的工程，单击直接进入编程界面。

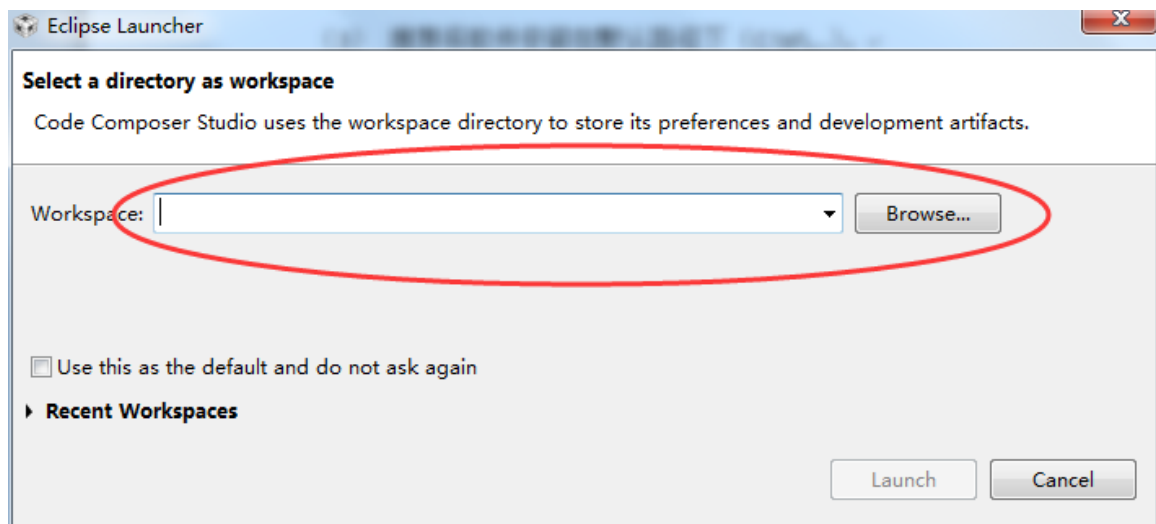


图 2-3 软件启动页面

注意：

1. 检查系统用户名称、系统的用户账号名称中不能包含中文字符，否则会出现软件无法打开、闪退等情况。
2. 路径中不能包含中文，否则软件无法识别，导致工程无法创建、使用。
3. 一个工作区中可以添加多个工程，在工作区文件夹下通过不同的工程文件夹区分。
4. 如图 2-3 中是打开工作区的路径，而不是工程的路径。以图 2-1 为例，在输入路径时输入的应是 (F:\TI Programs Files\g2553\l_gpio), 如果填写成图 2-2 的路径 (F:\TI Programs Files\g2553\l_gpio\gpio_01), 相当于创建了一个空工作区, 软件会在该文件目录下重新添加配置文件。
5. 在图 2-3 中输入的路径下如果没有以有相关的配置文件，则会新建，反之则不会新建。单击 launch 进入编程界面，如图 2-4 所示：
6. 如果需要新建一个工作区，可在图 2-3 中输入存放新的工作区的路径即可快速新建。

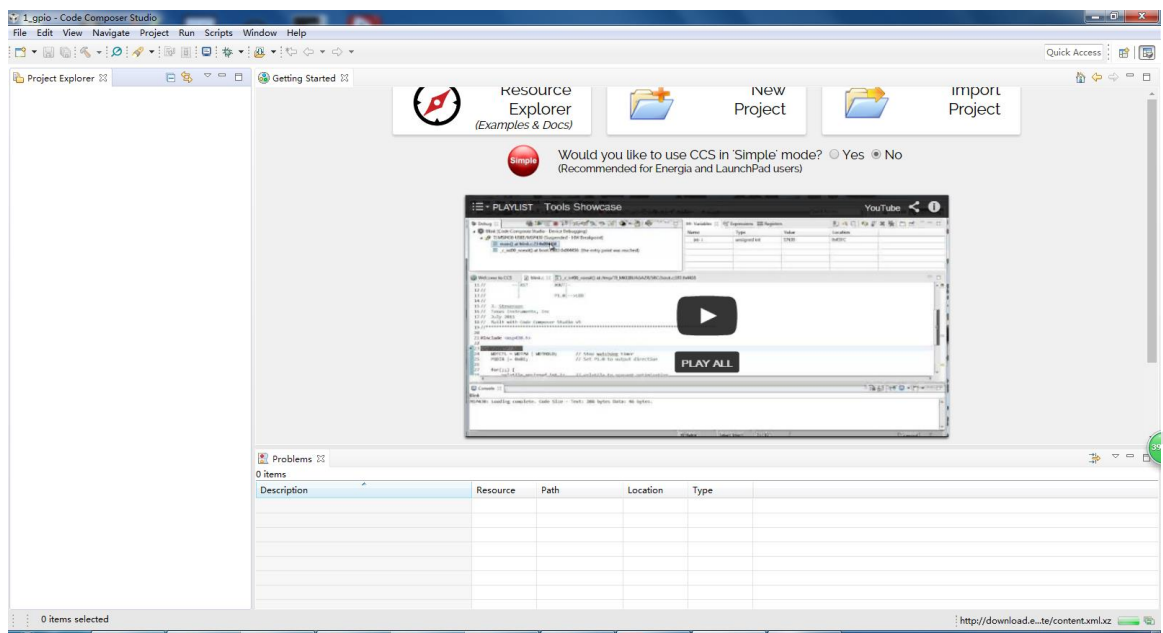


图 2-4 软件编程界面

2.2 创建工程

如图 2-5 所示,可从 Workbench 主菜单中选择 **Project> New CCS Project** 或 **File> New> CCS Project action**。创建工程时推荐将功能相似的工程存放在一个工作区中,不建议将自己编写的所有程序都放在一个工作区中,不利于分类,养成一个良好的编程习惯。

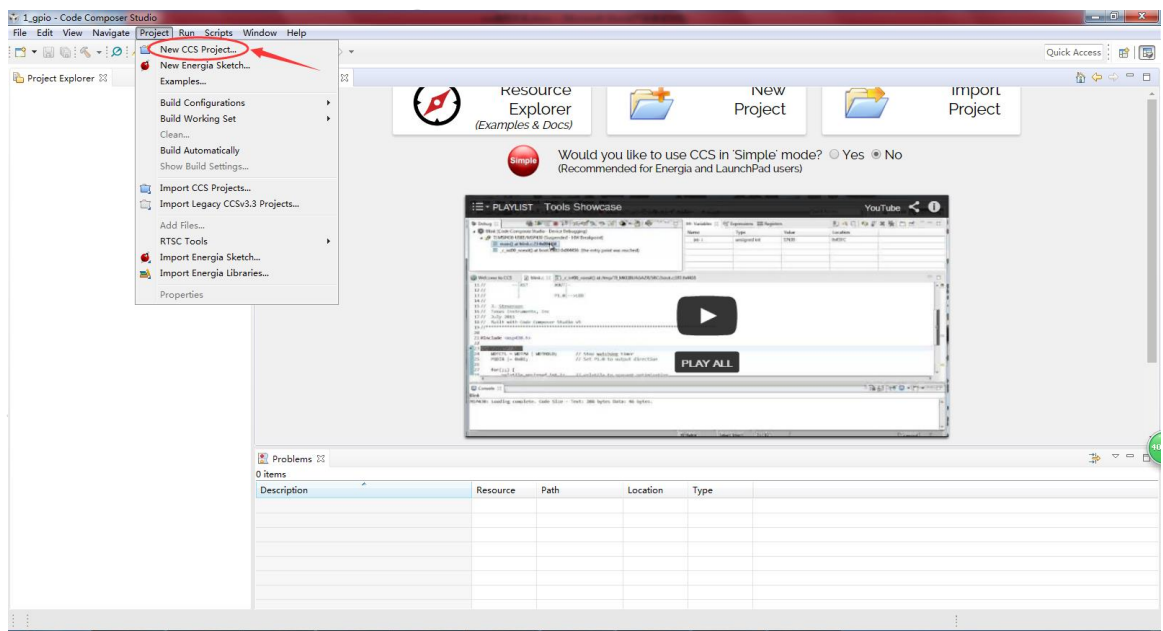


图 2-5 新建工程示意图

如图 2-6 所示,在“New CCS Project”向导中,可以选择取消选择“Use default location”默认位置复选框并设置项目存储位置。默认情况下,项目将在内部创建工作区目录。输

出类型允许您在创建可执行项目或库之间切换。

第一步，选择需要的设备系列和特定设备。 如果未列出您的特定设备，请选择其中一个通用设备。

第二步，根据设备选择相应的调试器。

第三步：输入工程的名称。

第四步：选择编译器。

第五步：选择项目模板或示例。

第六步：单击“Finish”，创建完成后自动跳转至如图 2-7 所示。

同时，也可以使用 TI 资源浏览器（Project > CCS Example Projects）用于导入其他示例项目。

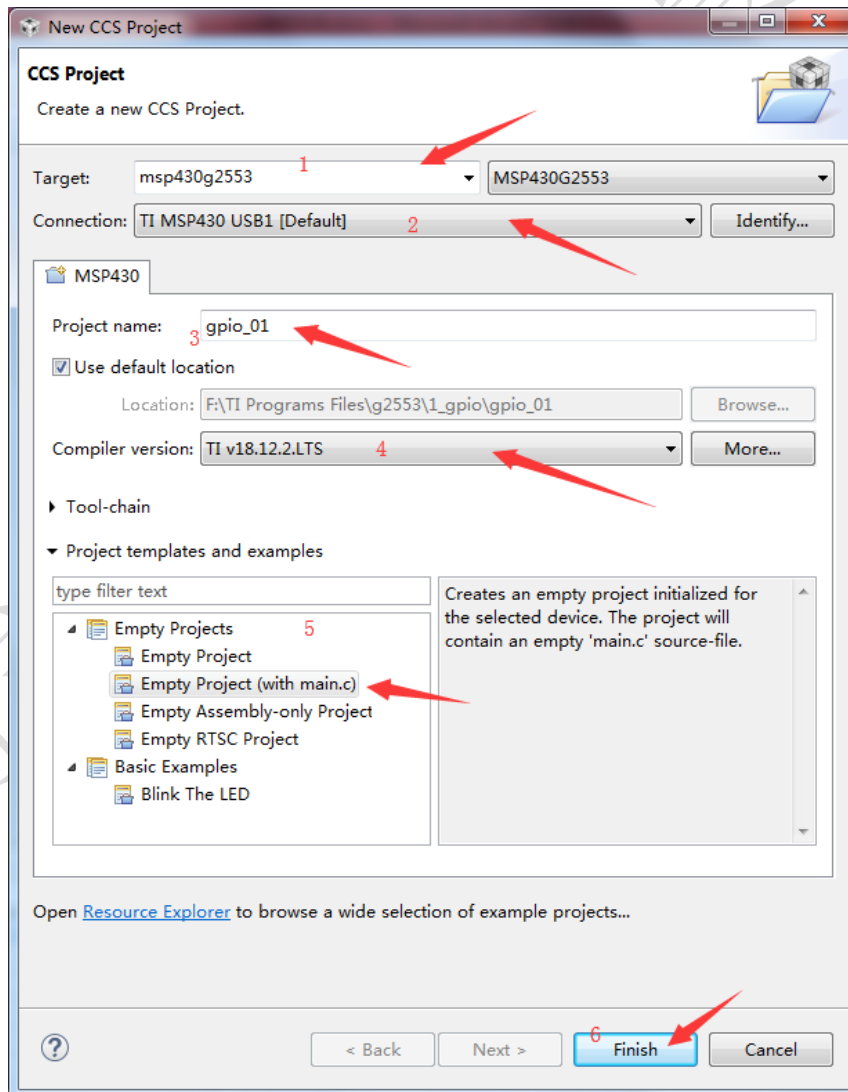


图 2-6 工程创建窗口

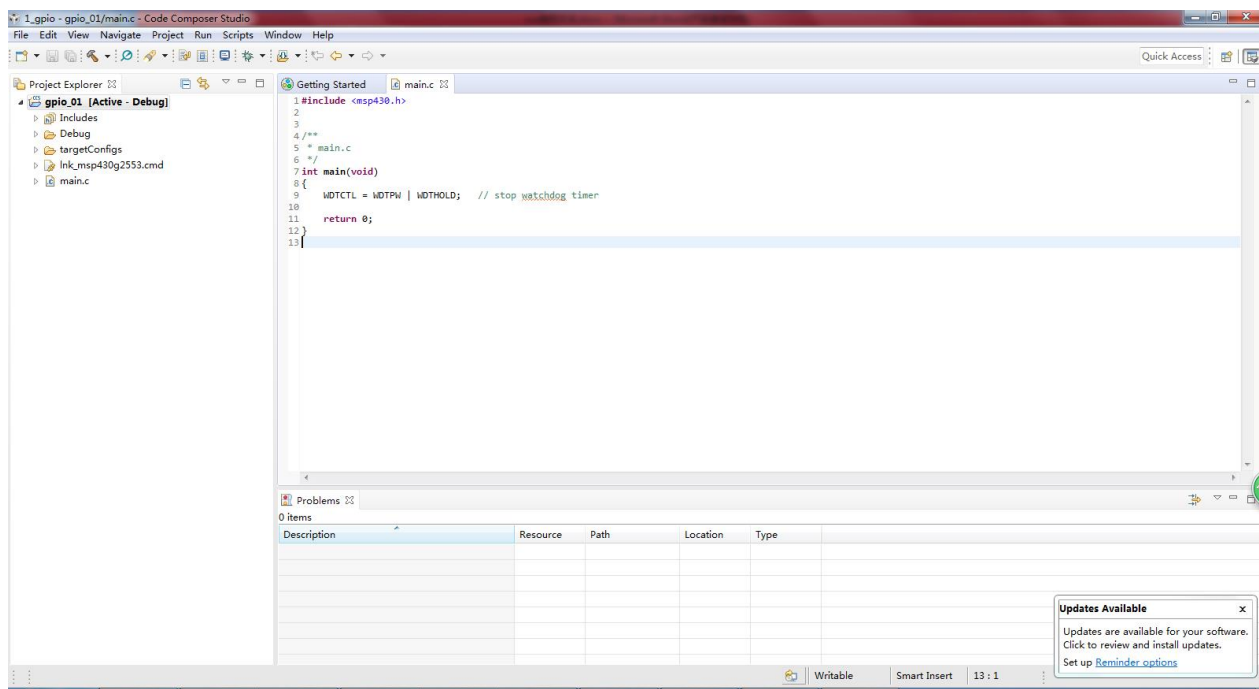


图 2-7 编程窗口

2.3 构建工程

1. 首先选择要构建的工程，然后在工具栏中选择“锤子”图标，或者选择 **Project > Build Project**。同时也可以构建所有项目，通过选择“构建所有”菜单操作来构建工作区中的所有项目。

2. 构建完成后，“Problems View”窗口将显示任何错误或警告。

3. 项目成功构建后，CCS 将为当前活动构建配置创建子目录，并将 .out 文件存储在 其中。可以启动调试会话并将程序加载到目标上。如果需要手动加载程序，在主工具栏上选择 **Run > Load Program** 或“Load Program”图标。

4. 如果需要调试，可以点击调试图标，将自动切换到 Debug 窗口。要手动切换到 Debug 窗口，请选择 CCS 窗口最右侧的 Debug 图标，或者选择 **Windows\perspective\open perspective \ccs Debug**。

2.4 导入现有工程

当需要将现有的工程导入到现有的工作区时，或将现有工作区目录下没有加载的项目重新导入到工作区时，如图 2-8 所示，在工程资源管理器中没有名称为“flash”的工程，但该项目位于工作区下的根目录中，如图 2-9 所示如果需要将其添加则执行 **File / open project..** 打开如图 2-10 所示的窗口，在第 1 步中输入需要导入“项目”的路径(F:\TI

Programs Files\g2553\1_gpio\flash), 在第 2 步勾选需要导入的项目, 第 3 步可根据自己需要选择, 如没有要求选择默认即可, 然后执行第 4 步, 软件将自动导入项目到本工作区。

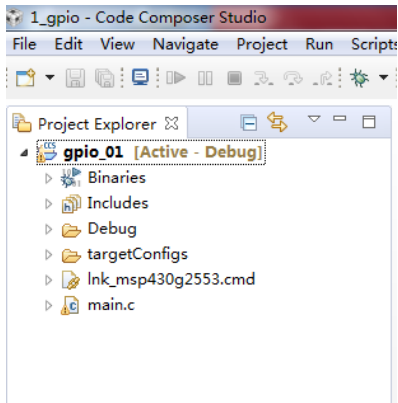


图 2-8 项目资源管理器

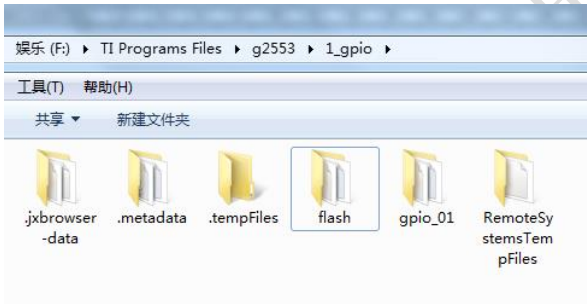


图 2-9 工作区根目录

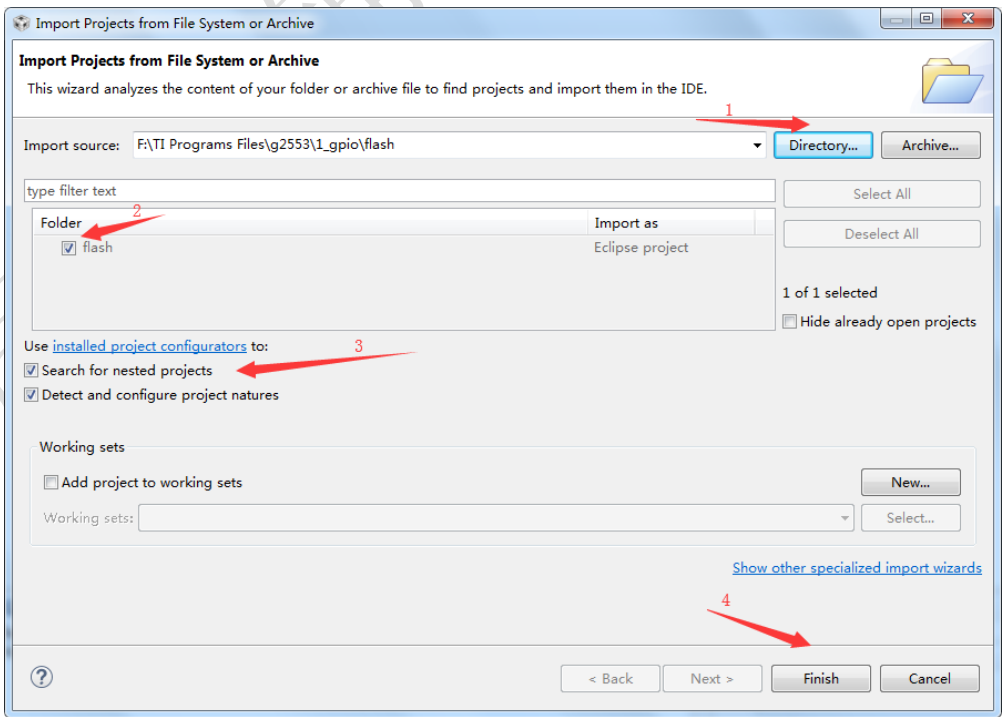


图 2-10 项目导入窗口

2.5 导入例程

Ti 官方提供了许多 SDK，其中包含了许多驱动文件、示例程序等，通过这些文件可快速学习并掌握其使用方法，下面介绍如何通过资源管理器添加示例程序，并使用。通过 view / Resource Explorer classic 打开，如图 2-11 所示，如果已安装 SDK 但该目录下没有，可通过图中箭头处添加 sdk 的安装路径，添加后即可在资源管理器中打开。

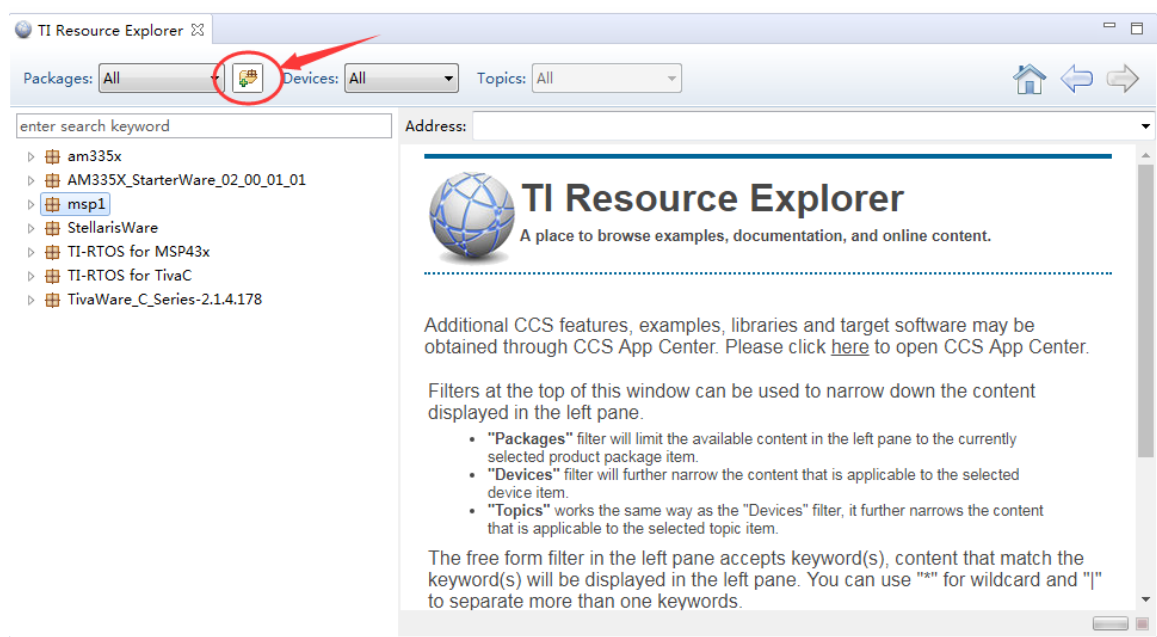
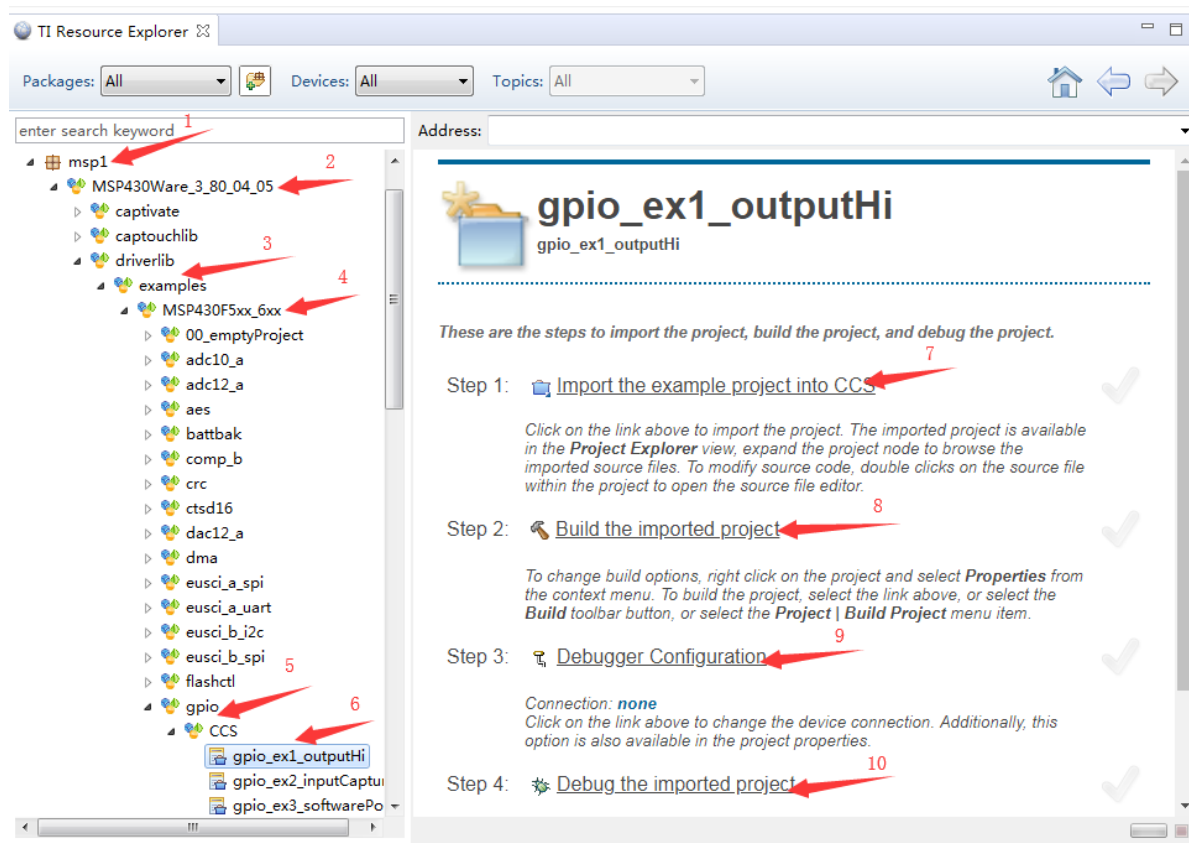


图 2-11 资源管理窗口

本节以 MSP430F5529 单片机为例，添加 MSP430Ware 中的示例工程：

1. 首先创建一个工作区，然后打开资源管理器。
 2. 选择示例工程，如图 2-12 所示，第 1~5 步为选择工程的路径，第 6 步中选择需要添加的示例工程，单击示例工程则会出现图中右侧的工程操作选项，
 3. 导入工程，首先单击第 7 处，导入工程至当前工作区，如果导入成功，则后面的“√”亮起，表示导入成功。
 4. 开始构建工程（编译），导入成功后执行第 8 步，开始构建工程，如果构建成功，则会在控制台输出如图 2-13 所示的提示。如果编译错误，或提示警告，则可尝试修改编译器版本，如图 2-6 中第 4 步，修改为软件中以安装的编译器版本。
 5. 仿真，如果构建工程通过并需要仿真，则执行第 10 步，将自动跳转至调试窗口。
- 自此，导入完成。



2-12 导入示例项目

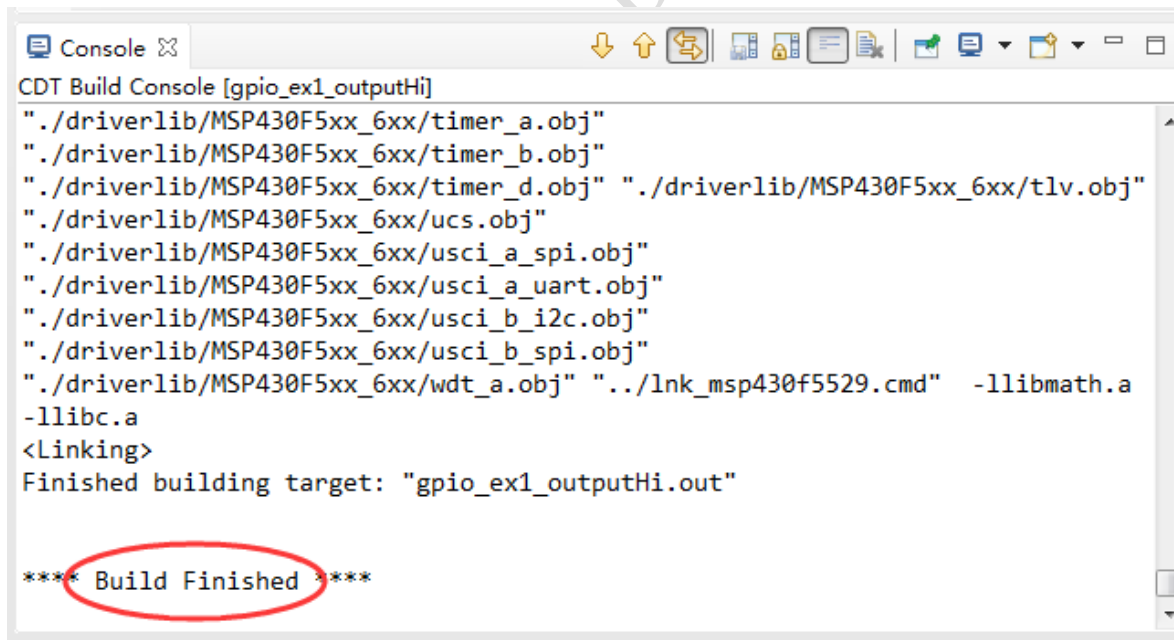


图 2-13 项目构建成功

2.6 导出工程

一个工程由工作区和项目两部分组成，在需要导出工程时，建议将工作区和项目一起打包，防止出现配置文件丢失等情况，导致项目无法正常编译。例如，在“F:\TI Programs

Files\g2553\1_gpio\gpio_01” 路径中，红色为工作区目录，蓝色为项目目录，项目目录位于工作区下。所以打包时应为如图 2-14 所示，打包“1_gpio”文件夹，鼠标右键单击，利用打包软件（压缩软件）进行打包。



图 2-14 工作区目录

当程序结构简单时，也可直接将项目通过 ccs 导出，在项目资源管理器中单击需要导出项目的名称，选择“Export”，会弹出如图 2-15 所示窗口，按图中步骤进行操作，然后在图 2-16 中选择需要导出的项目及文件，在第 2 步中选择导出文件的存储位置，最后执行第 3 步，完成导出。

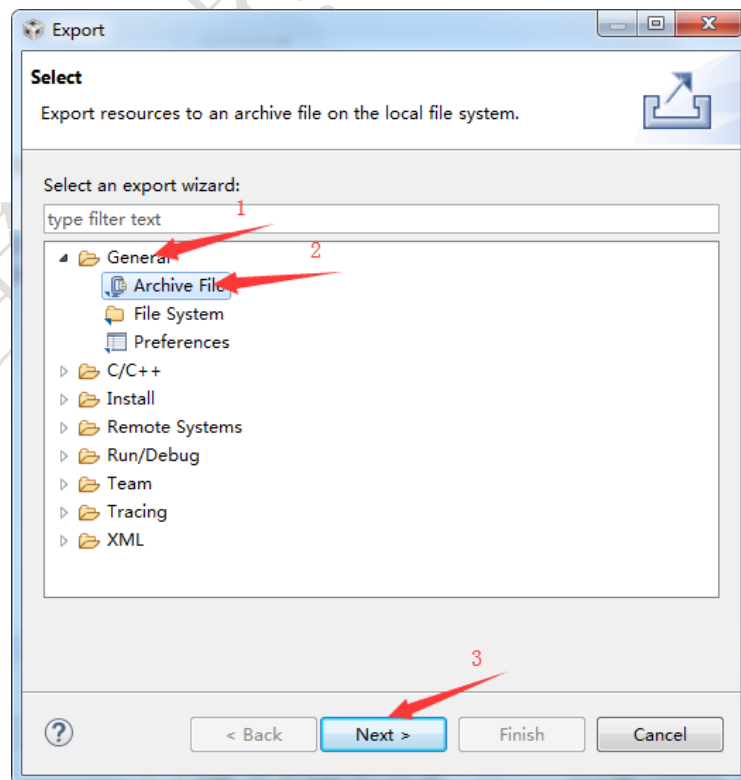


图 2-15 导出项目 1

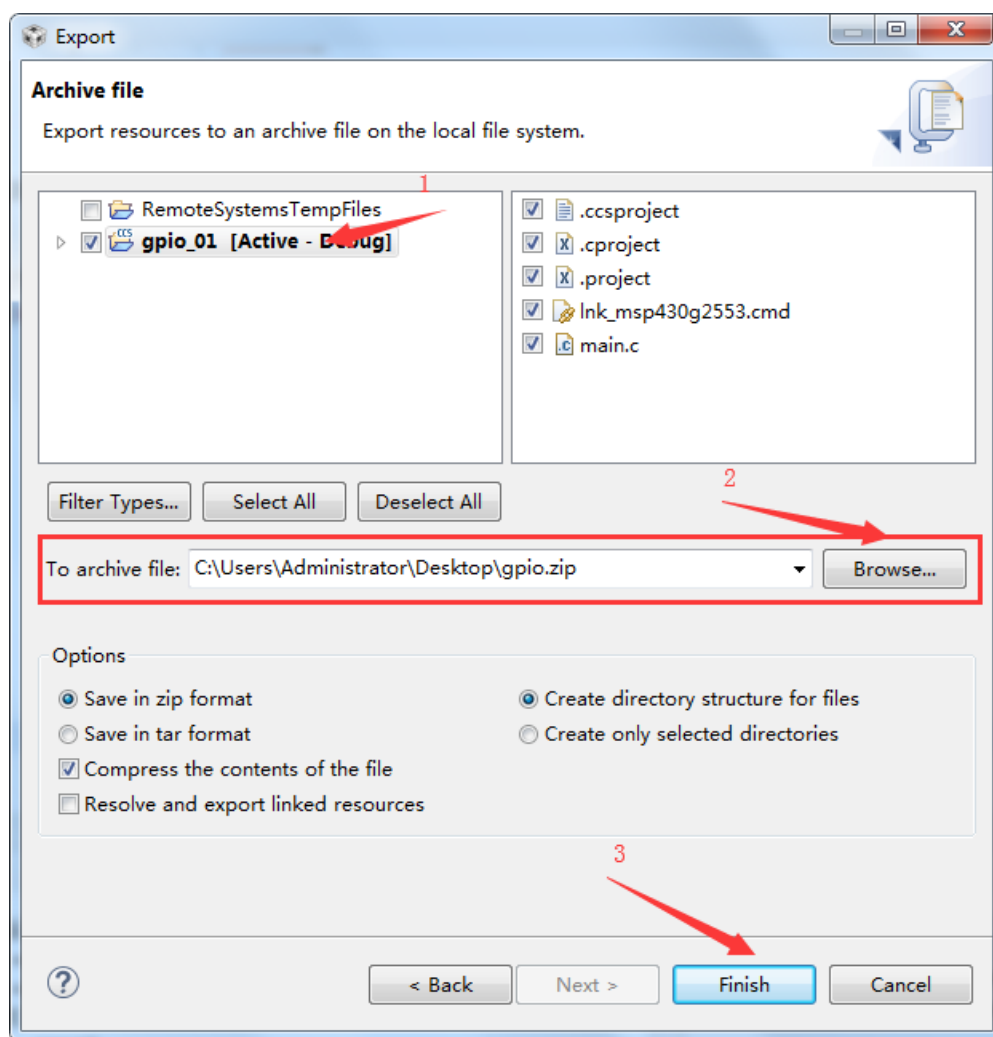


图 2-16 导出项目 2

2.7 生成 Hex、TXT

当需要输出 hex 或 txt 文件时，可通过**项目配置**进行设置，如图 2-17 所示，在第 3 步中勾选启用，即可在构建工程时生成 hex 文件。当需要生成 txt 文件时，选择第 4 步，跳转至如图 2-18 所示，在第 2 步中选择如图所示的选项，此时就能够生成 txt 文件。

注意：如果在项目配置中没有图 2-17 中第 2 步选项，则需要重新创建一个工作区，然后新建项目，再移植原先的程序。

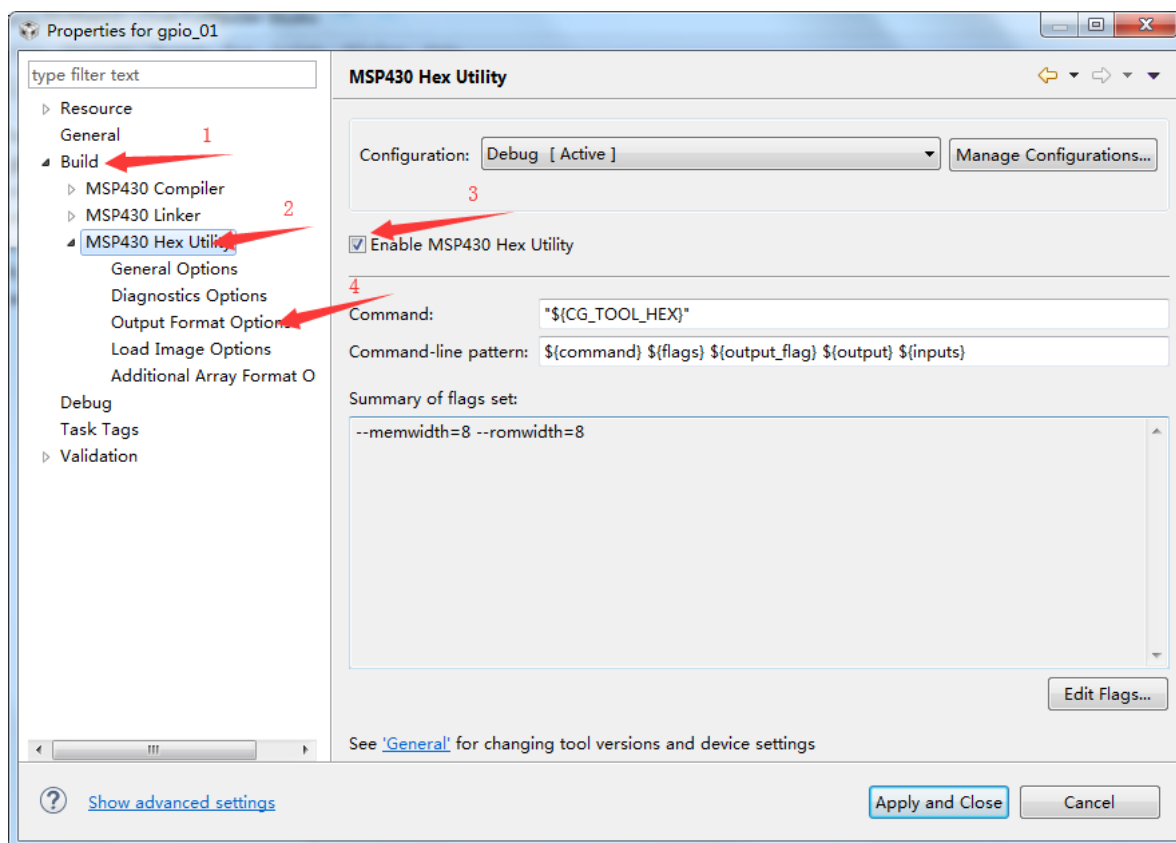


图 2-17 生成 HEX 文件

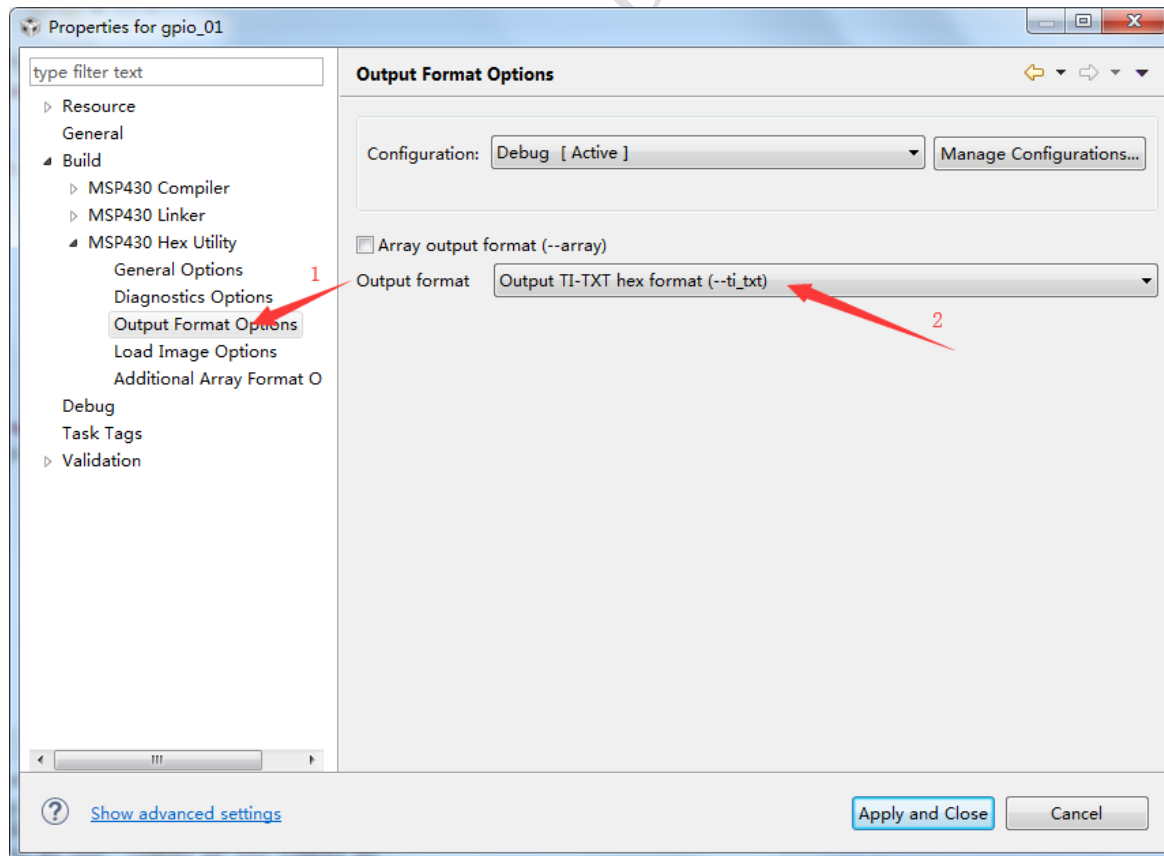


图 2-18 生成 TXT 文件

2.8 添加包含选项

新建项目时只生成一个“main.c”C 文件，**包含选项**指的是在模块化编程中将一些实现不同功能的 C 程序文件或文件夹包含在项目中（引用外部文件），如果在“main.c”中引用了这些外部文件，但没有将他们包含在项目中，编译器在构建项目时则会报错。

Code Composer Studio 中提供了两种选项，一种是添加具体的文件，一种是添加文件夹。如图 2-6 所示中新建的项目中一般只包含一个 main.c 文件，不包含其他的 C 语言文件。当需要外部文件时会出现将外部文件直接保存在项目下、或者保存在项目下新建的文件夹下，方便文件的分类管理。

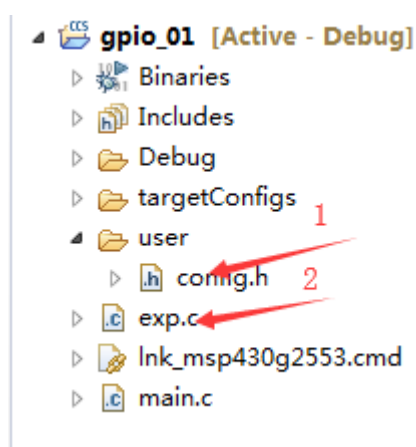


图 2-19 项目资源管理器示意图

如图 2-20 所示，在项目配置窗口中打开，在第 4 步中添加文件夹，在弹出的窗口中选择文件夹所在的位置。在第 5 步中添加文件，添加完成后执行第 6 步，然后重新编译即可。

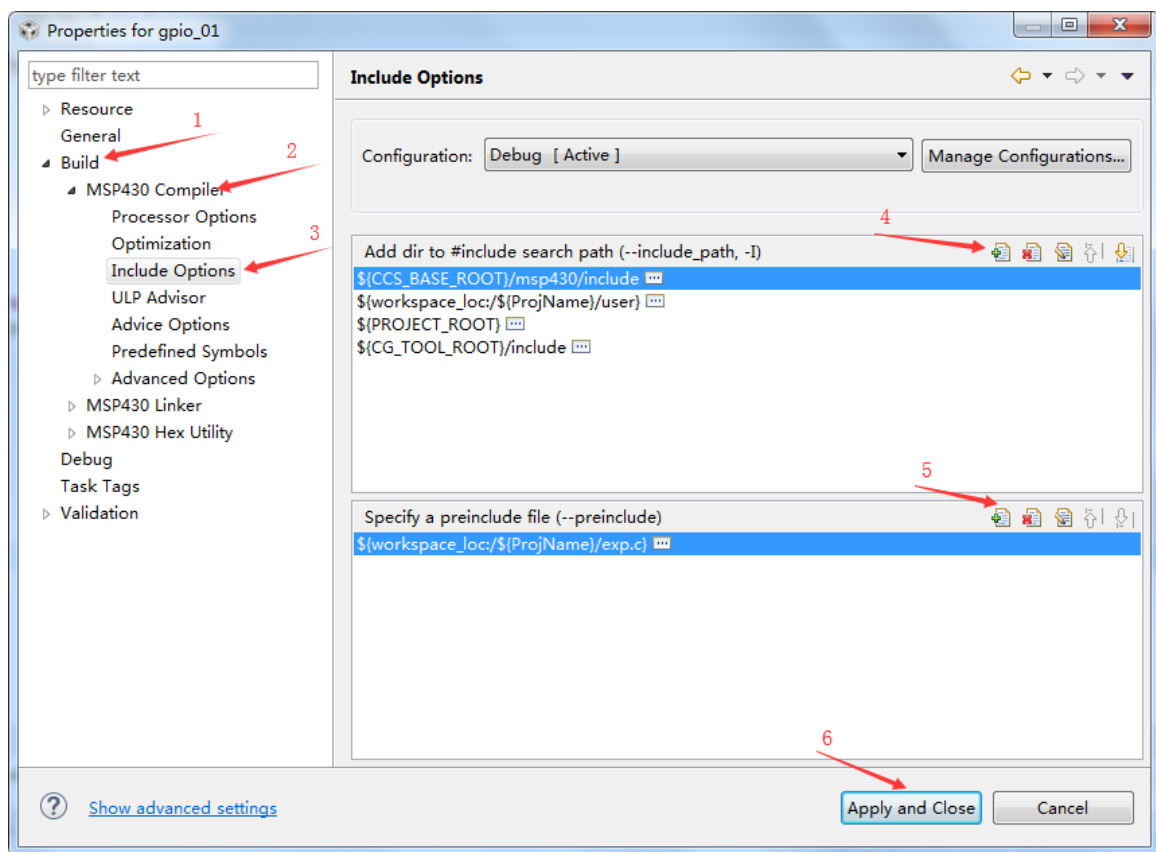


图 2-20 添加包含文件

2.9 本章小结

本章主要介绍了工程相关的操作，需要掌握项目的打开和导入，特别需要注意的两个概念是：“工作区”和“项目”，项目保存在工作区中。一个工作区中可以保存多个工程，在工作区文件夹下通过不同的工程文件夹区分。在软件开始界面打开项目时，应当填写工作区的路径。

3 常用工具使用方法

在编程及仿真时，CCS 提供了许多工具供我们使用，熟练的掌握并使用可以极大的增加编程、调试效率，下面逐个介绍常用的一些工具及其使用方法。

3.1 Resource Explorer classic

打开方法：view/ Resource Explorer classic

功能：管理查看云/本地资源，一般为 ti 为各个产品提供的驱动或示例。

使用方法：如果安装了例如 MSP Ware、Tiva ware 等资源，则会在如图 3-1 所示的窗口左侧看到目录，通过目录可以添加所提供的示例工程，帮助快速学习使用。

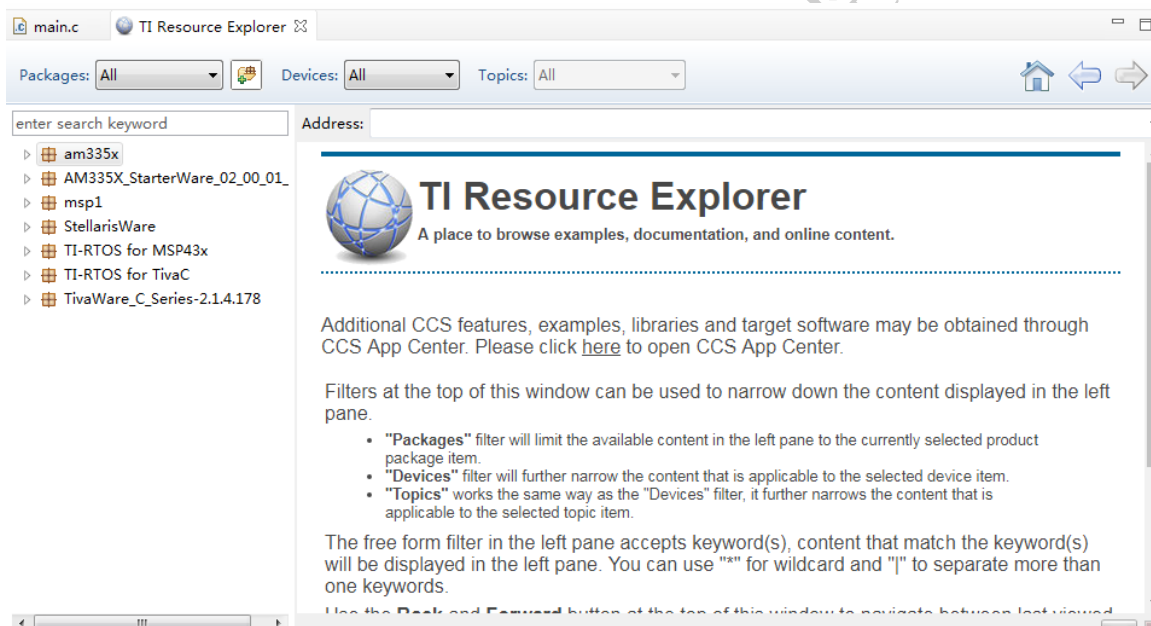


图 3-1 Resource Explorer classic 工具

3.2 Ccs app center

打开方法：view/ Ccs app center

功能：应用中心，提供了诸如 MSP Ware、Tiva ware、编译器、仿真等拓展工具。

使用方法：在如图 3-2 所示中选择自己需要的工具，然后点击“install software”进行安装即可。

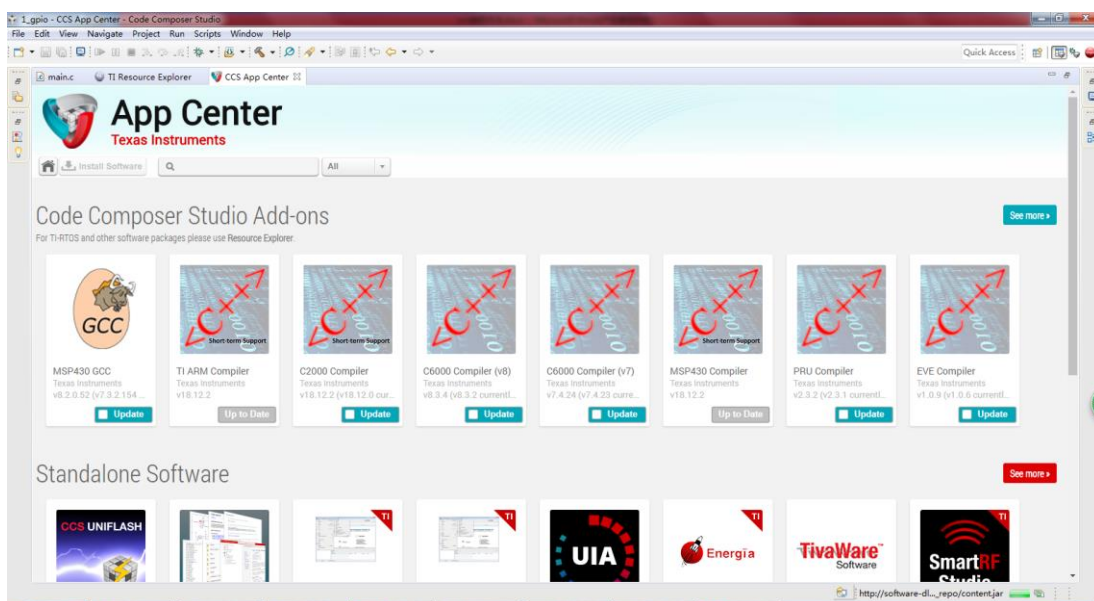


图 3-2 CCS APP Center 工具

3.3 Terminal

打开方法：view/ Terminal； 工具栏图标

功能：终端，提供了诸如命令行功能（cmd.exe）、SSH、串口调试工具、远程登录终端等拓展工具。

使用方法：最常用的是该窗口下的串口调试工具，在如图 3-3 所示中选择相应的、比特率等，该串口工具缺点是默认的编码不支持 2 进制、16 进制的显示，可通过 Echoding 更改。

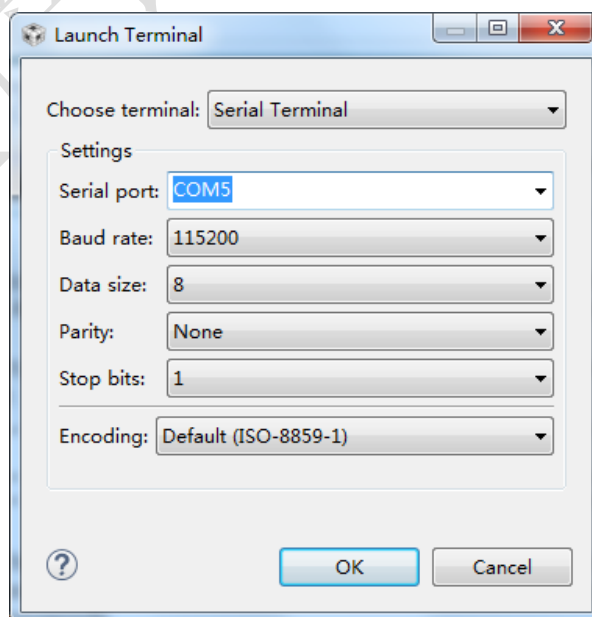


图 3-3 Terminal 串口工具

3.4 Outline

打开方法：view/ Outline

功能：显示当前打开的文件中定义的变量、函数、数组等。

使用方法：如图 3-4 所示，可在该窗口中查看定义的变量、函数、数组等，单击可快速跳转到对应位置。

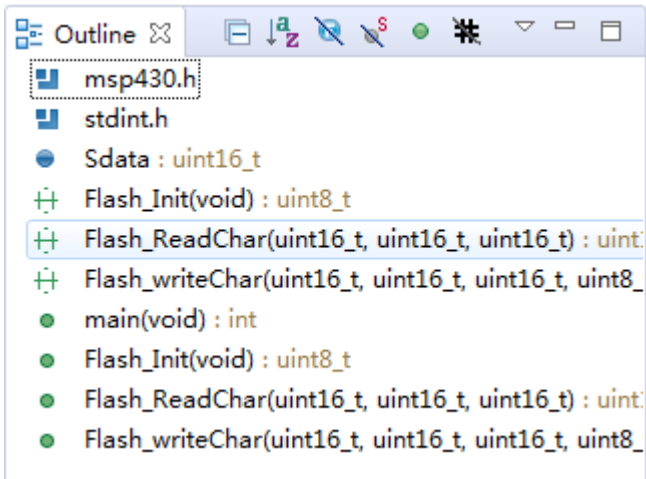


图 3-4 Outline 工具

3.5 Optimizer assistant

打开方法：view/ Optimizer assistant

功能：优化当前的程序。

使用方法：如图 3-5 所示，选择需要的优化模式，单击 start 开始优化，优化完成后可根据需求选择对应优化等级的程序并应用，如果需要重新优化则单击图 3-5 右上的暂停图标。

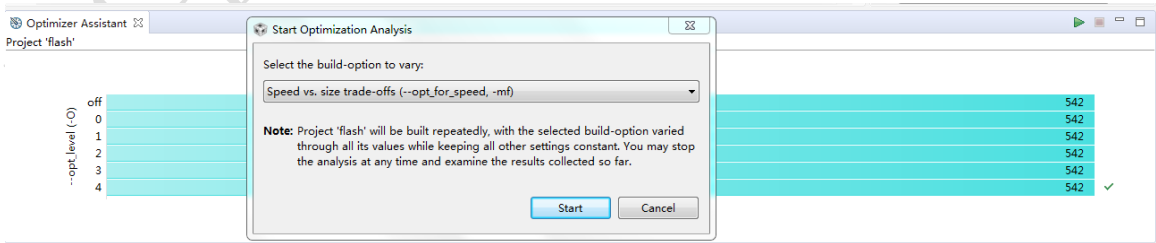


图 3-5 Optimizer assistant 工具

3.6 Memory browser

打开方法：view/ Memory browser

功能：该工具只能在进行仿真时使用；可查看输入的地址中存放的数据。

使用方法：如图 3-6 所示，在“1”处输入一个地址然后单击键盘“回车”即可查看当前地址的数据。在图中标记的“2”处保存了最近打开的地址，可方便操作。在图中“3”处可以选择导入、导出当前代码段。在“4”处可以选择数据显示的方式，常用有 8 位、16 位、32 位，根据调试的芯片或自己需求选择。

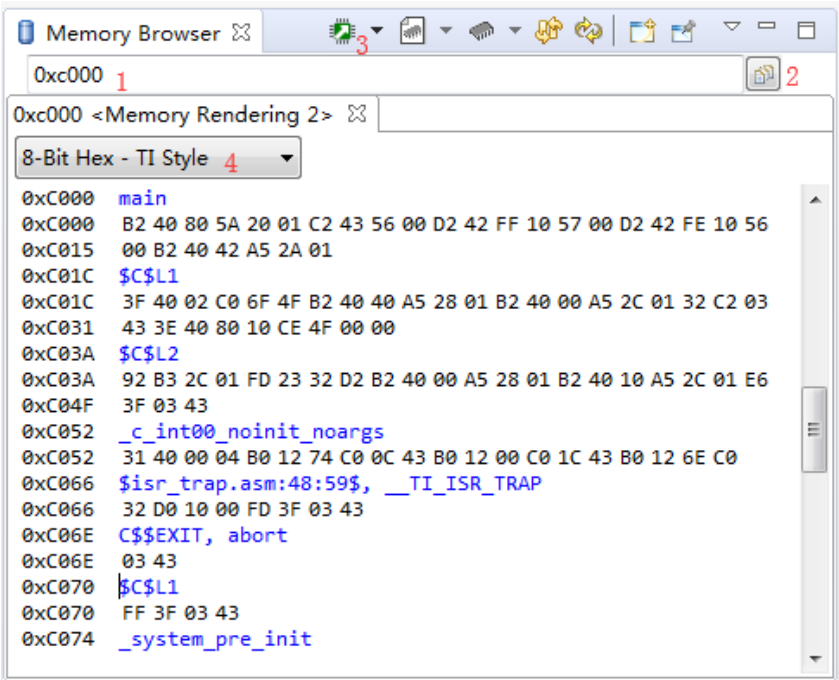


图 3-6 Memory browser 工具

3.7 Expressions

打开方法：view/ Expressions

功能：在该窗口中可以添加程序中定义的变量，在仿真时使用可直观的看到数值变化。

使用方法：如图 3-7 所示，单击“add new exp...”在弹出的窗口中输入要添加的变量名即可，调试过程中当变量值有变化时会自动更新。

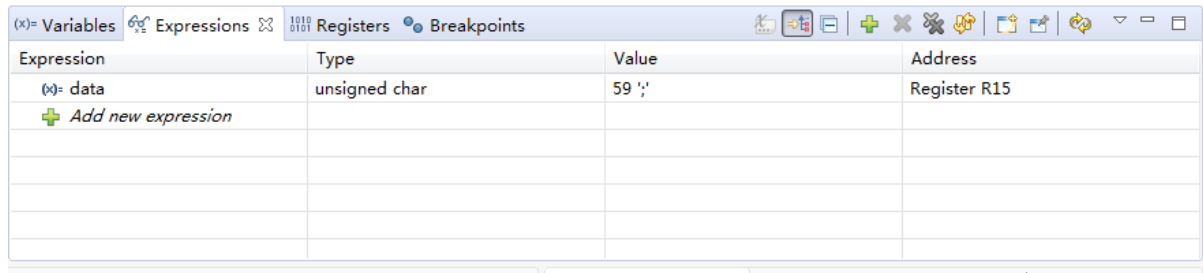


图 3-7 Expressions 工具

3.8 Registers

打开方法：view/ Registers

功能：可以通过该窗口在仿真时查看各个模块的寄存器配置情况。

使用方法：如图 3-8 所示，在该工具下可以实时查看寄存器的值，并可对其值修改。

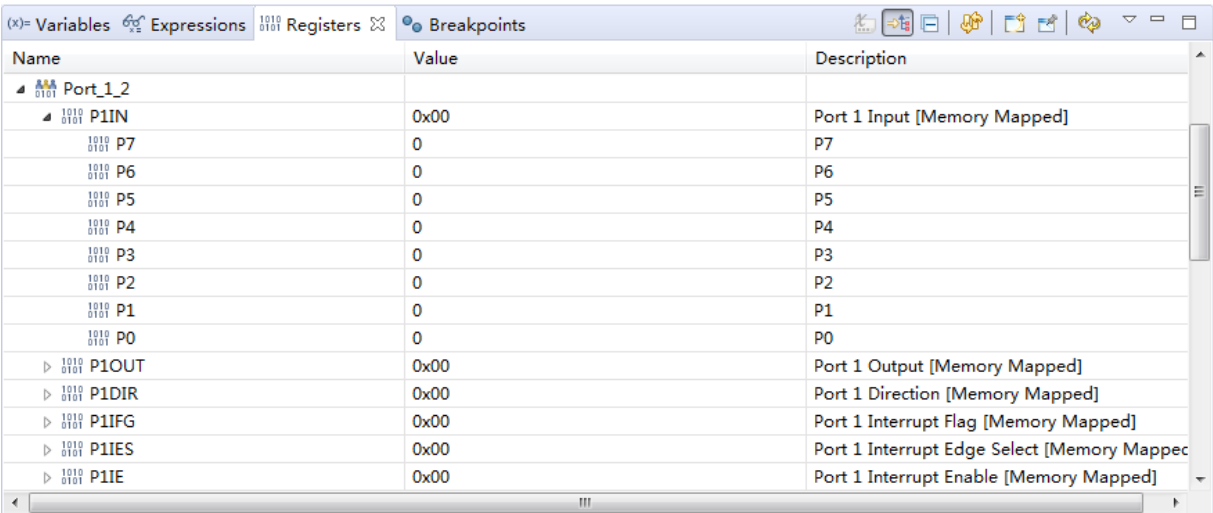


图 3-8 Registers 工具

3.9 Variables

打开方法：view/ Variables

功能：显示函数中定义的变量。

使用方法：如图 3-9 所示，在仿真时会自动识别当前函数中所定义的变量并显示，当跳出当前执行的函数后则不再显示，可配合 Expressions 工具使用。

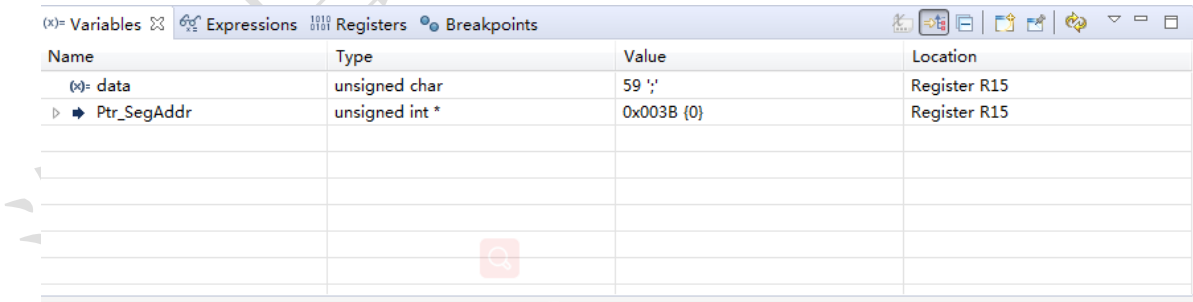


图 3-9 Variables 工具

3.10 Disassembly

打开方法：view/ Disassembly

功能：机器码(汇编语言) → 高级编程语言。

使用方法：如图 3-10 所示，可在该工具窗口直观的读取反编译代码。

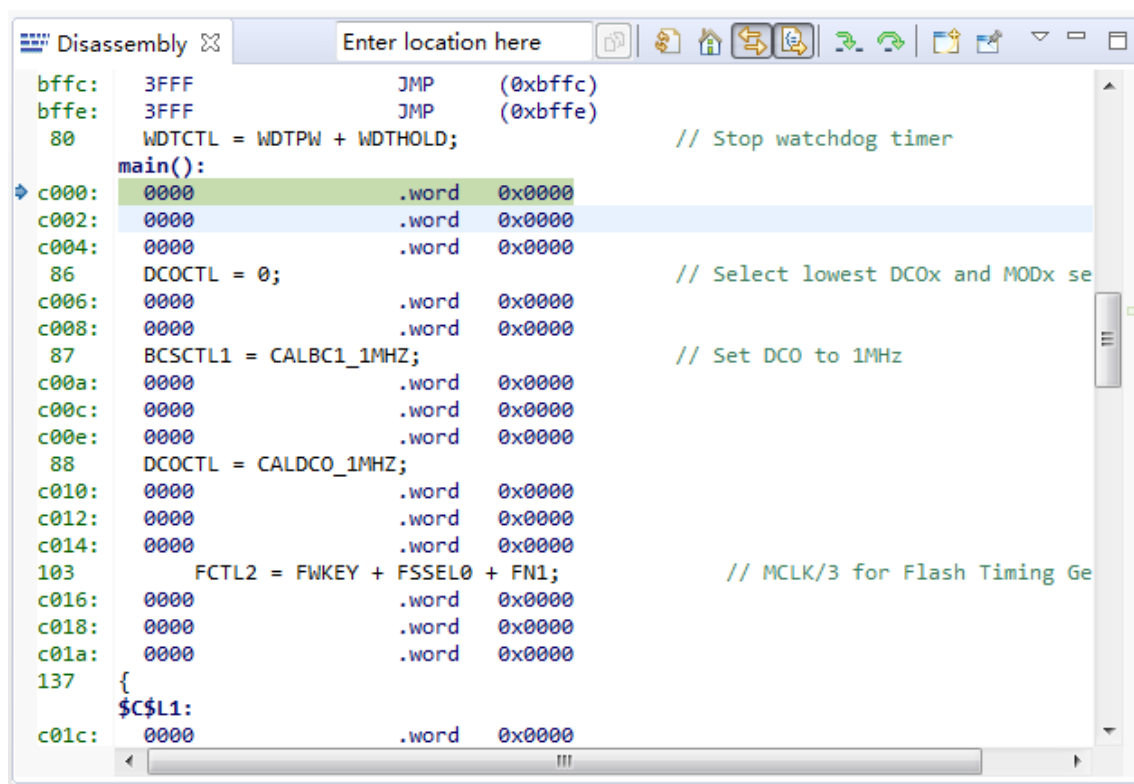


图 3-10 Disassembly 工具

3.11 Breakpoints

打开方法：view/ Breakpoints

功能：可在查看所有的断点。

使用方法：如图 3-11 所示，仿真时可在该窗口管理所有的断点，可查看例如断点的名称、类型等信息。

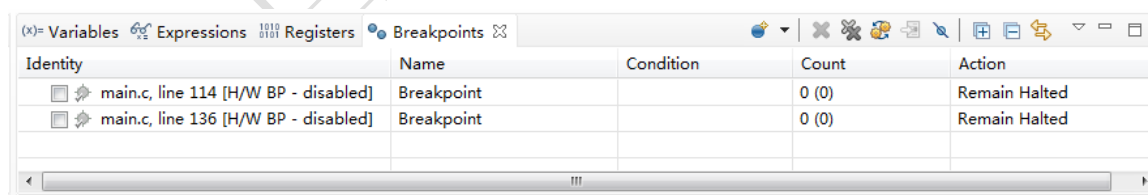


图 3-11 Breakpoints 工具

3.12 添加书签/待办事项

打开方法：在代码编辑窗口最左侧行数前单击鼠标右键；在代码行输入“TODO”，

功能：添加书签/待办事项。

使用方法：在代码编辑窗口中选择需要添加书签/待办的代码前端显示（行数）的位置单击鼠标右键即可弹出图 3-12 的窗口，选择“add book..”为添加书签。，选择“add task..”

为添加待办事项。也可使用双斜杠注释一行，然后输入“TODO”，空一个字符、然后添加相关信息，如此也可添加一个待办事项。

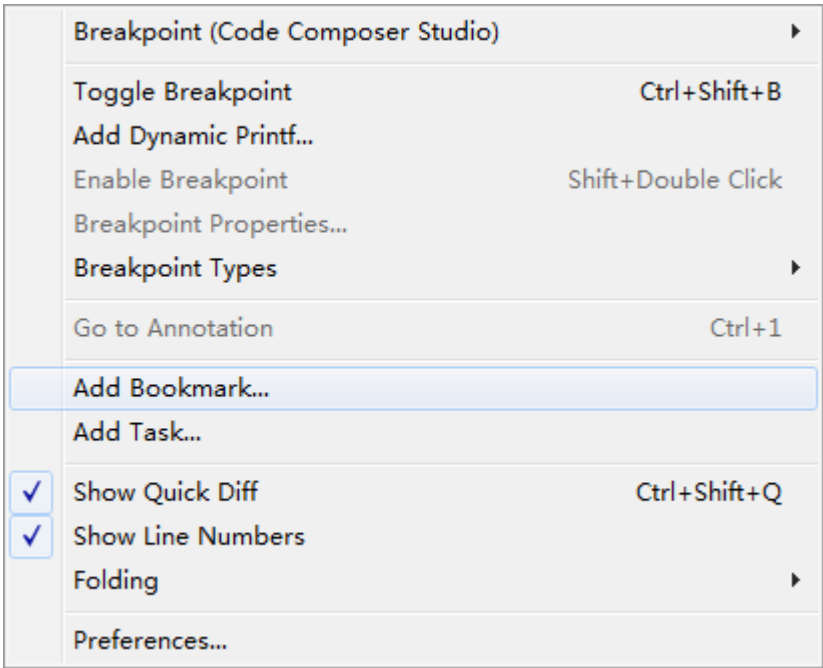


图 3-12 添加书签/待办事项工具

如图 3-13 所示，可将 add tasks 用在提醒需要修改程序参数的地方，可通过 view/other/General/Tasks 查看当前的待办事项，双击可自动跳转至待办事项。同样可通过 view/other/General/Bookmarks 查看当前的书签。

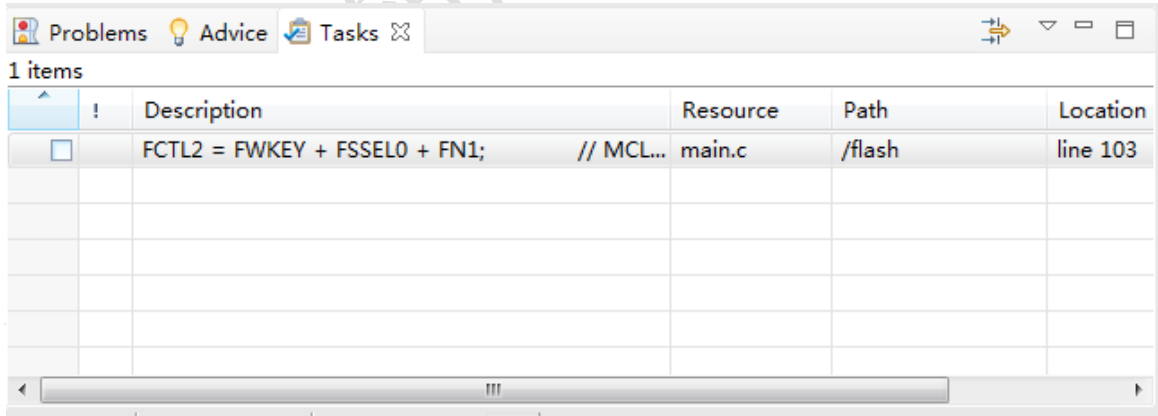


图 3-13 待办事项管理工具

3.13 SDK 的启用

有时候，当我们使用 msp ware、tiva ware 及其他 ti 官方提供的集成 SDK 报错时，可能仅仅是安装了，但 CCS 软件并没有启用这些插件，解决方法是在 Windows/Preferences/ Code Composer Studio /products/RTSC 启用相关的 SDK，如图 3-14

所示，勾选需要添加的插件。

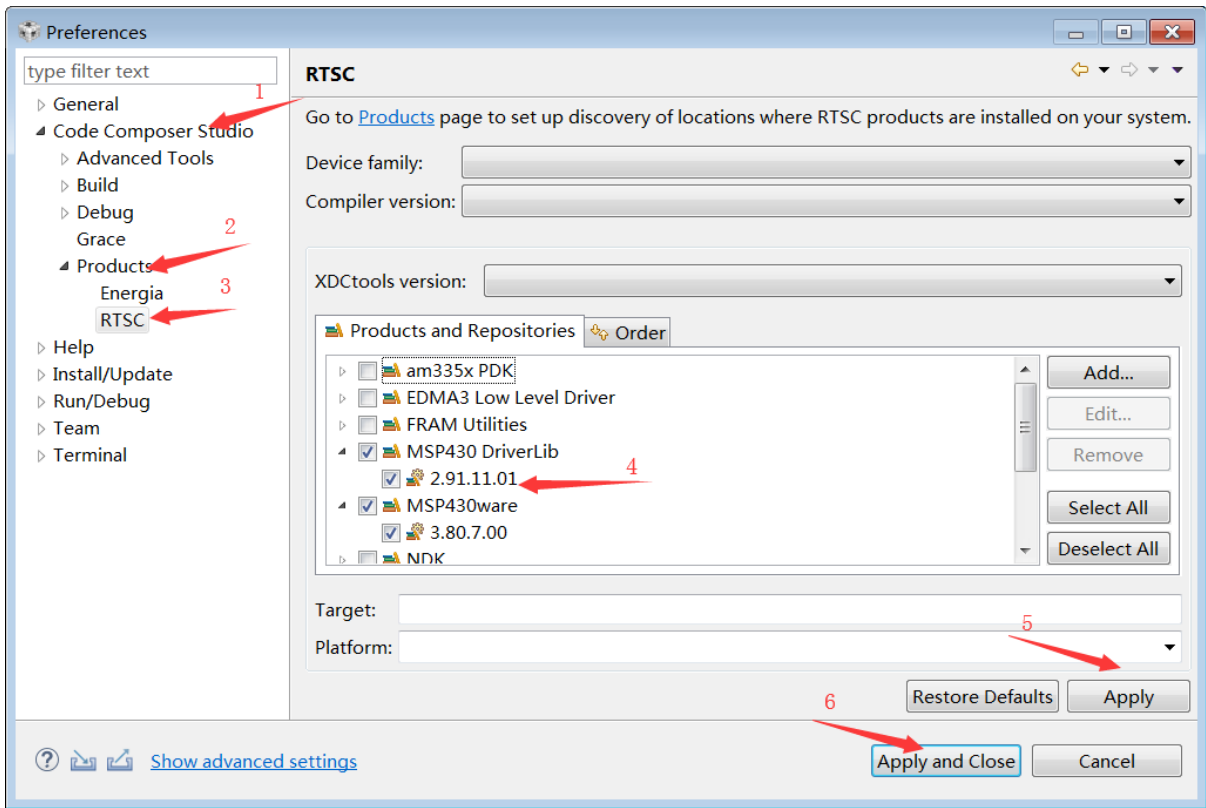


图 3-14 启用 SDK

3.14 编译器的更新

Ti 会不定期更新 Code Composer Studio，可以通过 Help/ Install new software 打开如图 3-15 的窗口，在“work with”中选择需要更新的类别，或选择所有可更新的项目（All Available Sites），然后在需要更新的工具前勾选，再根据提示操作即可。

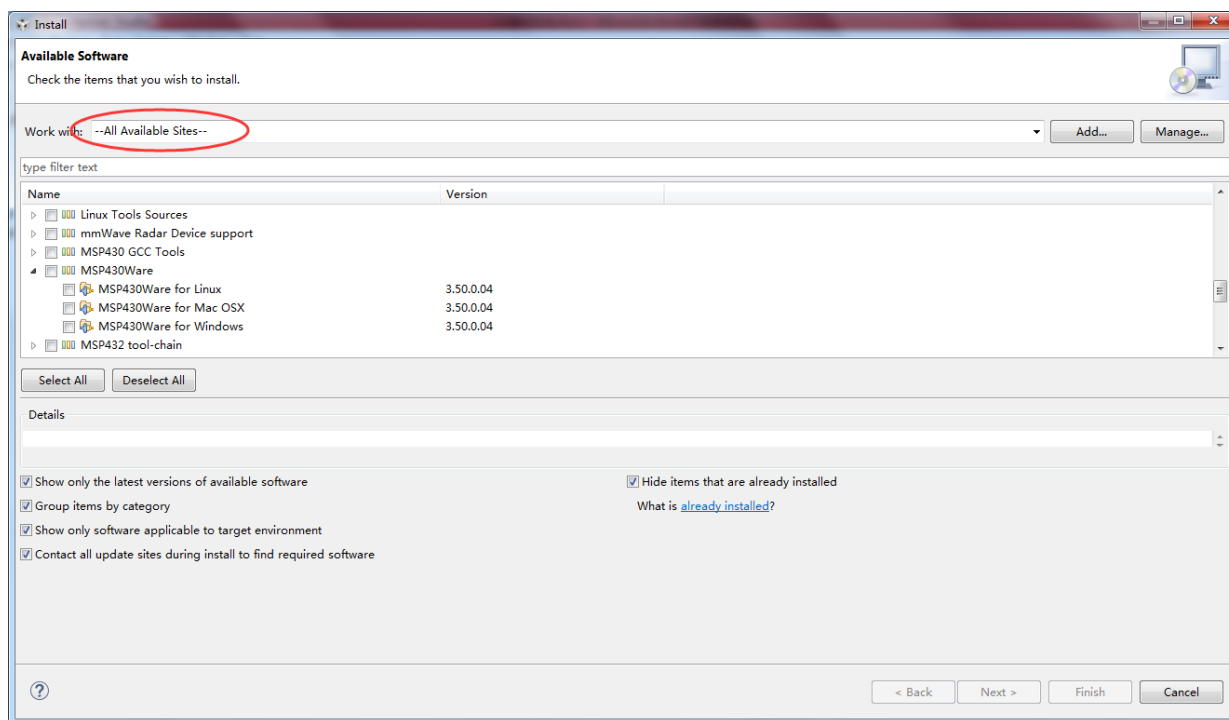


图 3-15 Code Composer Studio 更新界面

3.15 New window

打开方法：Window / New window

功能：克隆一个新的窗口。

使用方法：如图 3-16 所示，在一个窗口显示不便时，可以通过该工具实现窗口克隆，在其中任意一个窗口中的操作会同步到所有窗口中。

3.16 本章小结

本章主要介绍了各个工具的打开、使用方法，其中许多工具可以对编程、调试起到很大的作用。Ccs 中的标签、待办事项可用在程序需要修改的地方，可在移植程序时通过这些信息快速的完成移植，降低开发时间与开发难度。

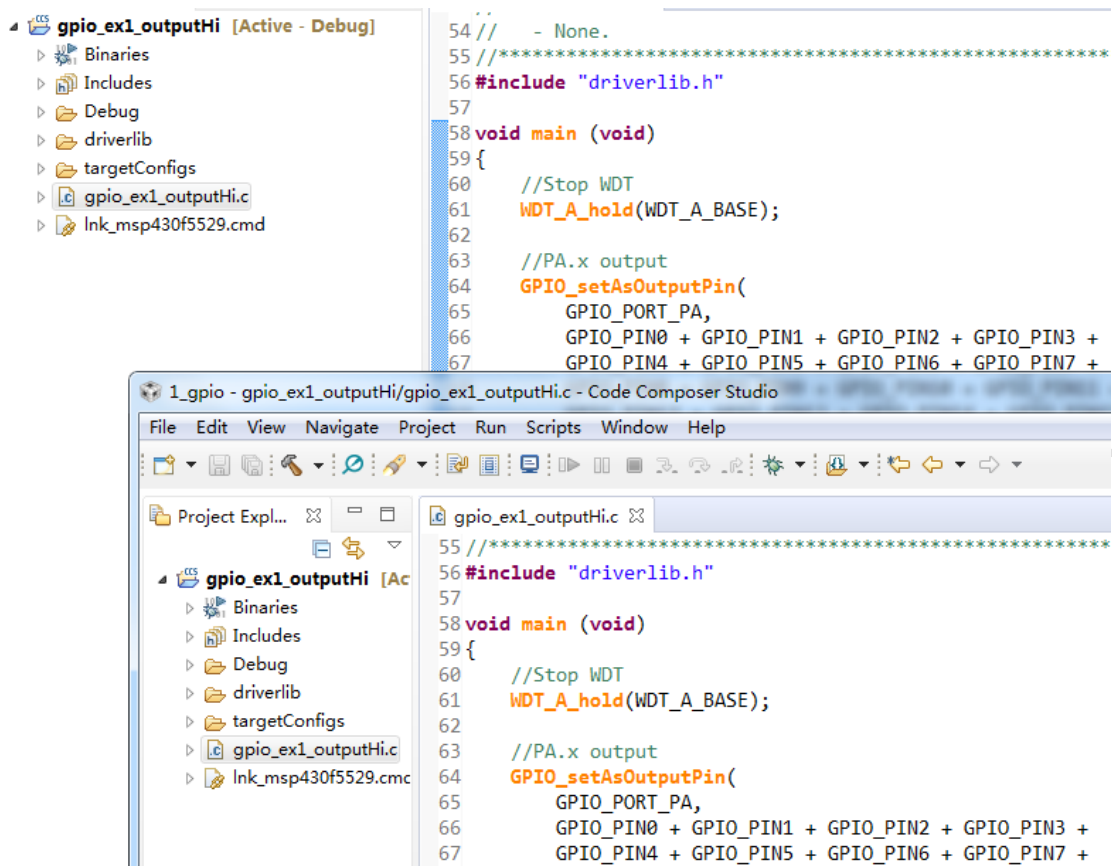


图 3-16 窗口克隆界面

4 个性化设置及操作

4.1 字体大小设置

在代码编辑窗口中单击鼠标右键，选择弹窗底部的“Preferences”命令，按照图 4-1 进行操作，在第六步中将打开图 4-2，在该窗口中设置字体、字体格式、字体大小等。

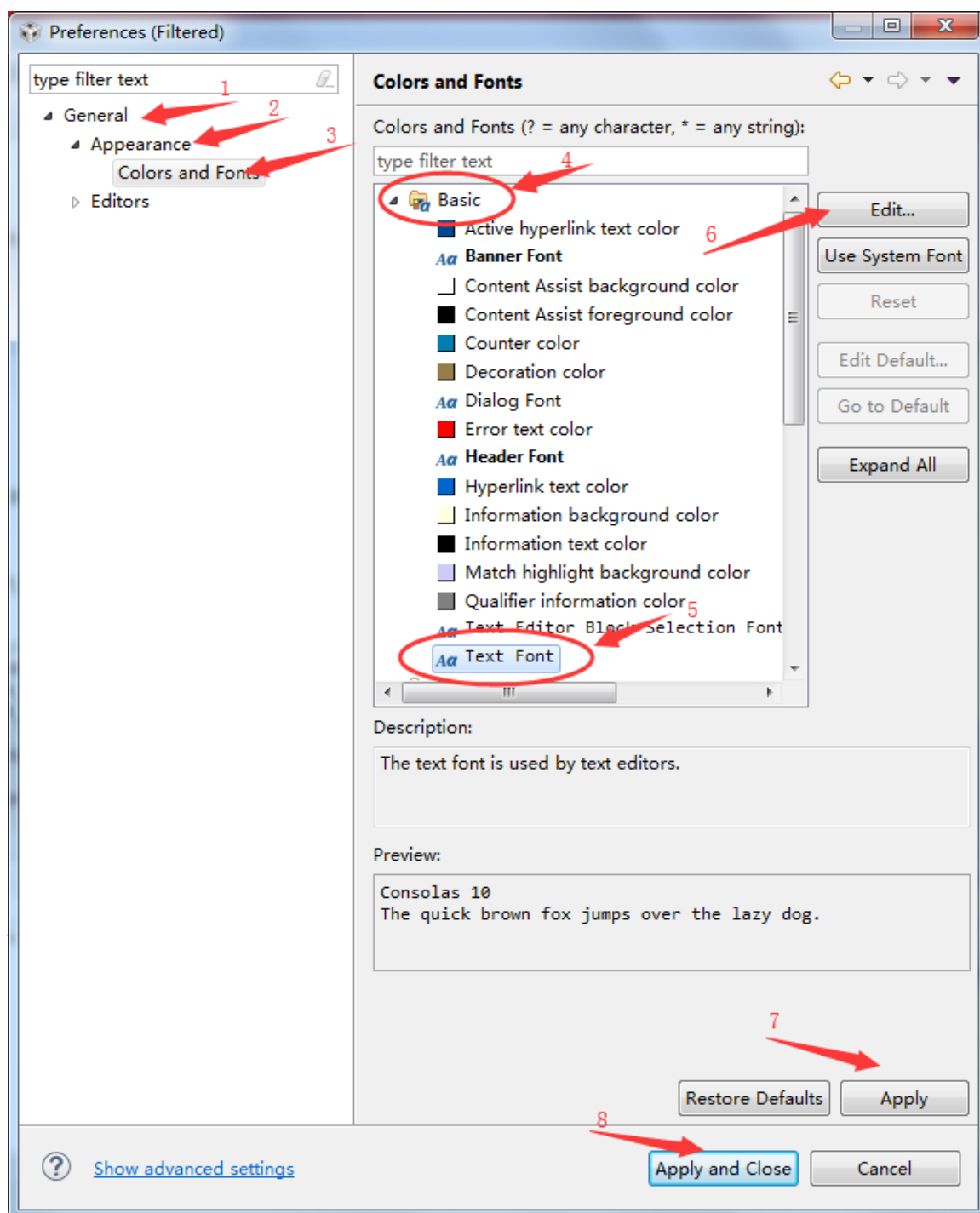


图 4-1 字体大小设置 a

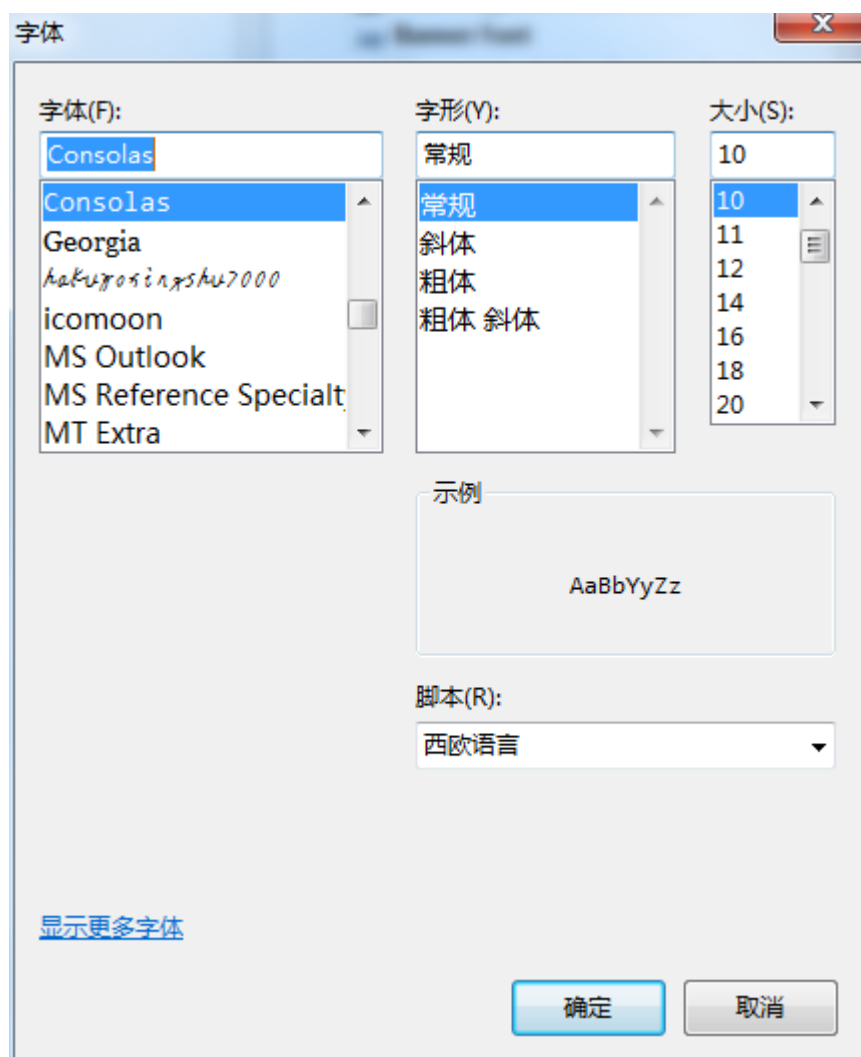


图 4-2 字体大小设置 b

4.2 代码颜色设置

Code Composer Studio 提供了个性化设置功能，可以修改代码颜色，以更好的区分不同的类别。在代码编辑器中单击鼠标右键，选择弹窗底部的“Preferences”命令，可根据图 4-3 进行操作，首先切换为高级设置模式，然后根据提示操作，到第 5 步时，在该选项框中选择不同的类别，然后修改其颜色、字体加粗、斜体、删除线、下划线等。通常会修改数字的颜色和函数名称以方便查看。

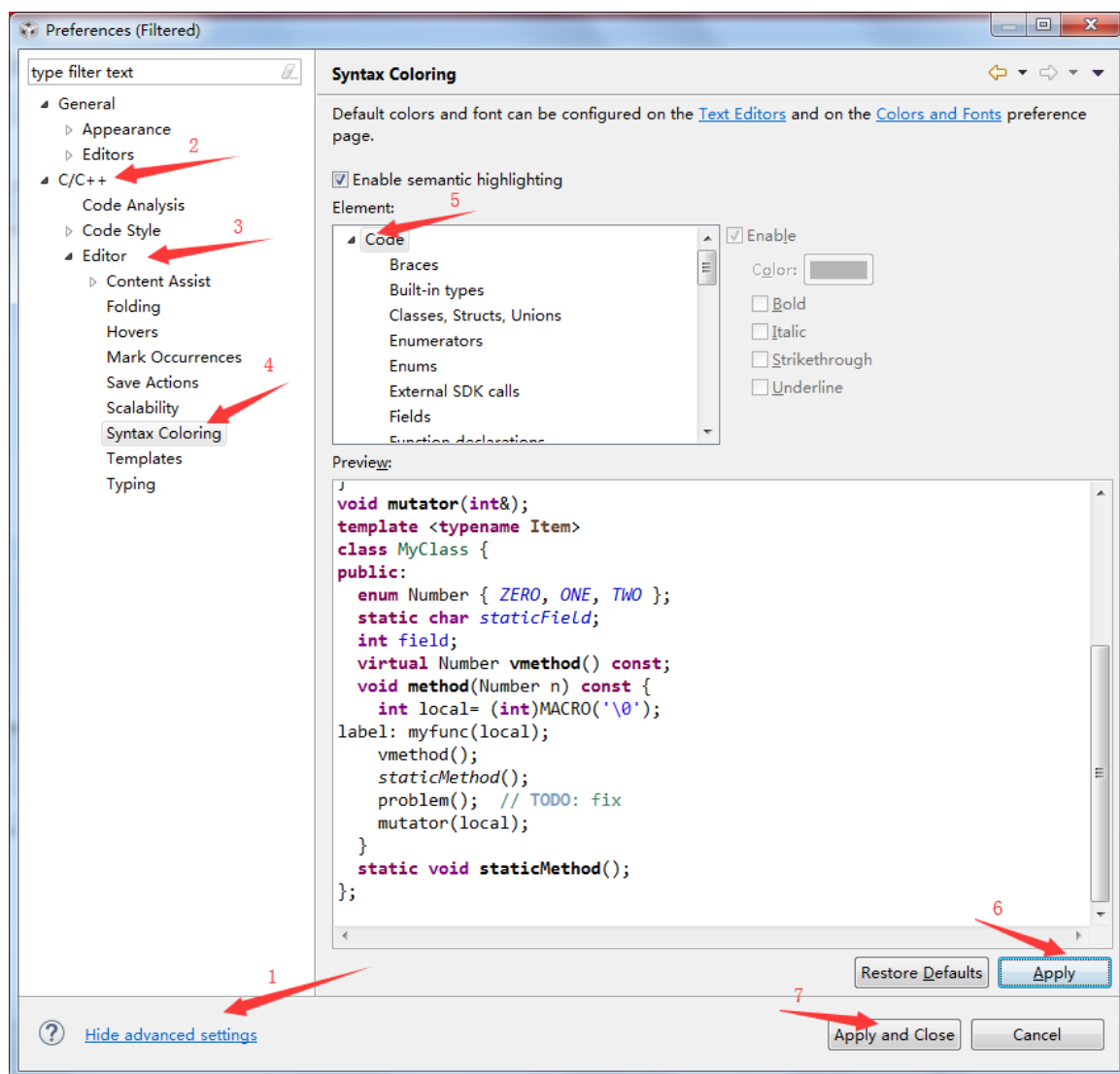


图 4-3 代码颜色设置

4.3 背景颜色设置

在代码编辑器中单击鼠标右键，选择弹窗底部的“Preferences”命令，根据图 4-4 进行操作，取消第 4 步的选项即可在第 5 步中选择自己喜欢的颜色做背景。

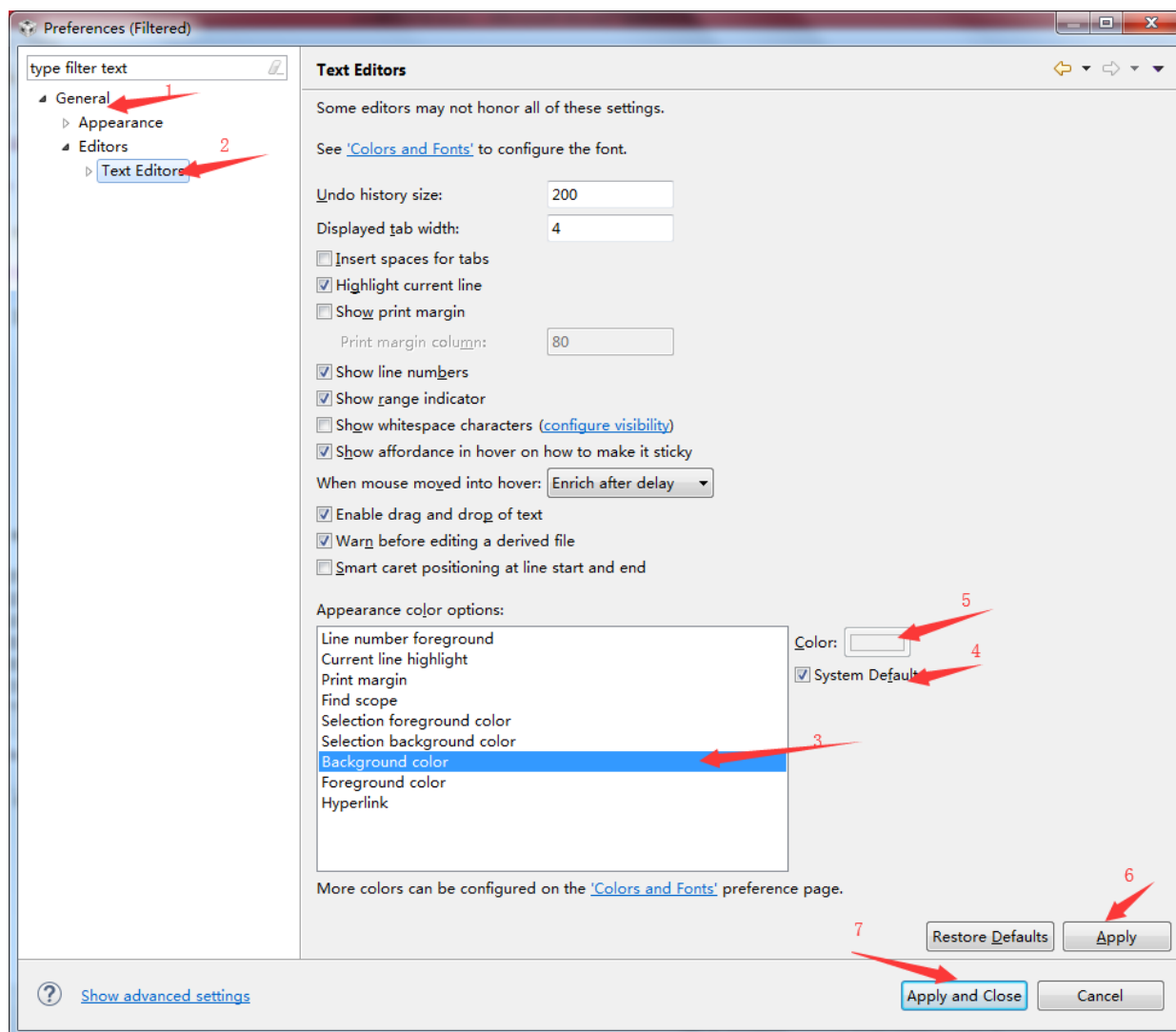


图 4-4 背景颜色设置

4.4 快捷键

快捷键的设置如图 4-5 所示，在 Windows\ Preferences 中打开。常用的快捷键操作有：

Ctrl + Z: 撤销

Ctrl + X: 剪切

Ctrl + C: 复制

Ctrl + V: 粘贴

Ctrl + F: 查找

Ctrl + 7: 注释，已注释掉的再次点击则为取消注释，且可同时进行多行操作。

Alt + Enter: 打开当前项目的配置窗口。

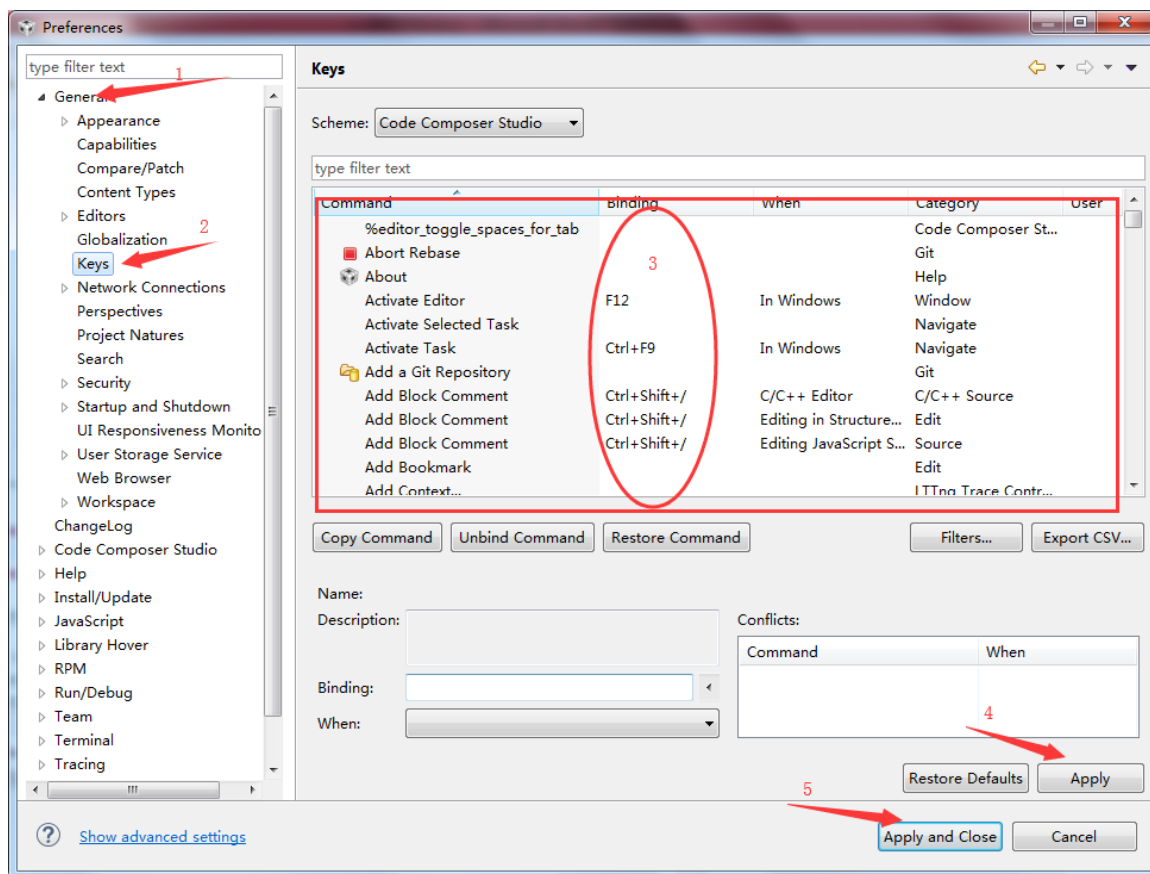


图 4-5 快捷键设置


4.5 本章小结

本章主要介绍了 Code Composer Studio 软件的一些常用的个性化设置，通过这些设置可以提高代码的可读性，通过修改背景色也可以减缓长时间使用电脑造成的眼部疲劳。

5 调试/仿真

当编写的代码中出现 bug 时，最好的方法是通过调试来快速定位问题，如果需要调试则需要使用仿真器连接芯片，常见的仿真器有 MSP4-FET430UIF、eZ-FET lite（TI 官方评估板板载仿真器）等。本例中使用 MSP-EXP430F5529LP 上板载的仿真器调试 MSP430G2553 最小系统，简单的演示调试过程、注意事项以及常见错误。

5.1 调试/仿真准备

首先需要回顾第一章中工具栏图标[工具栏图标](#)部分和第三章的内容，了解、掌握其功能。当硬件连接上电脑后在设备管理器中检查仿真器驱动，然后连接好最小系统，构建好程序后，点击命令行中“虫子”样式的图标  或执行 RUN/Debug 开始调试，软件会自动下载程序至 MSP430G2553 中，并跳转至调试界面。如图 5-1 所示：

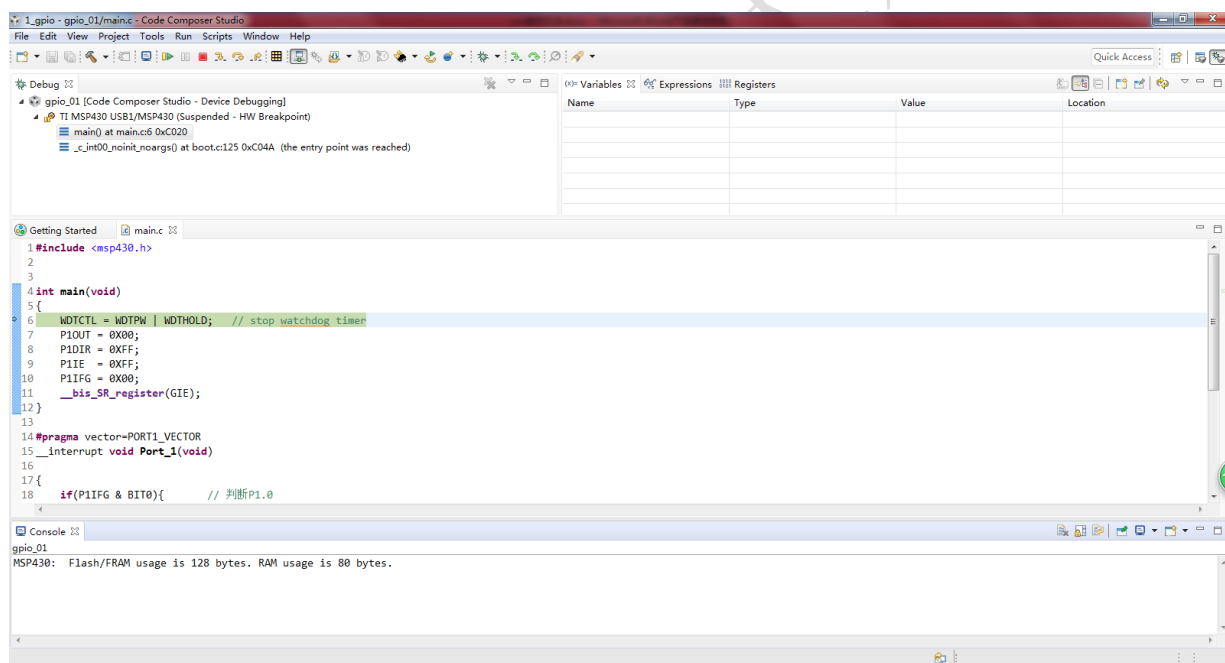


图 5-1 调试界面

5.2 调试/仿真操作

当程序被下载到 MSP430G2553 并跳转至窗口界面后，即可在调试界面执行相关的调试操作。一般常用的调试功能有单步调试和跨步调试，调试时根据自己需要选择。还有诸多在第三章中介绍的许多实用工具。在调试时，第 3.6 ~ 3.11 节中的工具是调试时非常常用的工具，可利用工具实时观察程序中变化的变量值以及查看内存中的数据。

5.3 调试/仿真注意事项

1. 在调试时特别需要注意的是**尽量不要添加延时函数**，在单步执行会导致调试时卡死，可以在调试前先注释掉，仿真结束后取消注释。
2. 在调试 MSP430 系列单片机时**不要启用低功耗模式**，当关闭 CPU 后，调试就无法进行。
3. 使用 MSP4-FET430UIF 的两线 SBW 仿真时，遇到不能下载、仿真，在确保硬件没有问题时，可尝试重新安装驱动。

5.4 本章小结

本章介绍了调试/仿真相关的知识，在调试中最重要的是调试工具的使用，重点需要掌握的工具具有 **Variables**、Expressions、Registers、Disassembly、Memory browser 等，这些常用工具的使用方法在第 3 章中，要认真学习掌握这些工具。

6 常见问题

问题 1 工作区、项目打开/导入类错误

以本文中第 2 章的案例来讲，如果需要打开一个工作区，那么路径应该为：F:\TI Programs Files\g2553\1_gpio，如果需要打开或者导入一个现有的项目，那么路径应该为项目的根目录，例如：F:\TI Programs Files\g2553\1_gpio\gpio_01 或 F:\TI Programs Files\g2553\1_gpio\flash

问题 2 下载程序完成后自动跳转到仿真界面，但没有仿真

该问题多出现于 MSP-EXP430F5529LP，如果没有使用到板载的串口，则可取掉 RXD、TXD 的跳线帽，一般就可解决该问题，或者尝试更新软件版本或编译器版本。

问题 3 下载程序时提示错误，但仿真器连接正常

该问题多数是因为版本不兼容，可通过删除项目下的.ccsml 文件，再新建一个，重新构建工程即可。

其他常见问题可参考：http://processors.wiki.ti.com/index.php/FAQ_-_CCSv6

如果在使用过程中有任何疑问或遇到问题，可通过以下方式联系我们：



QQ: 1419604776



QQ 群: 176242843



[淘宝店铺](#)

版权所有 ▪ 侵权必究